VI International Conference on Particle-based Methods – Fundamentals and Applications
PARTICLES 2019
E. Oñate, M. Bischoff, D.R.J. Owen, P. Wriggers & T. Zohdi (Eds)

COMPARISON OF THE FINITE VOLUME METHOD WITH LAGRANGIAN VORTEX METHOD FOR 2D FLOW SIMULATION AROUND AIRFOILS AT INTERMEDIATE REYNOLDS NUMBER

KSENIIA S. KUZMINA[1,2], ILIA K. MARCHEVSKY[1,2] AND EVGENIYA P. RYATINA[1,2]

[1] Bauman Moscow State Technical University
105005, 2-nd Baumanskaya st., 5, Moscow, Russia

[2] Ivannikov Institute for System Programming of the Russian Academy of Sciences
109004, Alexander Solzhenitsyn st., 25, Moscow, Russia

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

Key words: viscous vortex domain method, 2D flow, incompressible flow, open-source code, OpenFOAM, VM2D, backward-facing step

Abstract. The paper is dedicated to the numerical simulation of two-dimensional viscous incompressible flow. The Viscous Vortex Domains method is considered, which is one of the modifications of Lagrangian vortex methods. This method is implemented in the open-source VM2D code, which developes by the authors. Model problems of external and internal flow simulation are considered for VM2D testing. For the problem of the flow simulation around two closely spaced circular cylinders, VM2D is compared to OpenFOAM in terms of computational efficiency. As the internal flow problem the flow in a channel with a backward-facing step is considered. For both problems, the results obtained in VM2D are in good agreement with the results of other researchers.

1 INTRODUCTION

Today in computational hydrodynamics, different mesh methods are widely used, such as the finite difference method, the finite volume method and, rarely, the finite element method. There are a large number of their modifications; many of them are implemented in widely used software packages, both commercial and free: `Fluent`, `OpenFOAM`, `STAR-CCM`, `Flow Vision` and others. As a result, researchers and engineers have efficient tools for numerical simulation both in fundamental and industrial applications.

However, in recent decades meshless CFD methods are also gaining popularity. This class includes vortex methods, where vorticity is the primary calculated variable. The range of application of vortex methods is limited by incompressible flows; however, for such problems they can significantly exceed the efficiency of mesh methods. They are especially efficient for external flows simulation, since in this case the flow region with non-zero vorticity is relatively small in comparison to the computational domain usually considered in mesh methods. The advantages of meshless methods are even more pronounced when simulating flows around movable/deformable bodies, due to no need of mesh deformation or reconstruction.

There are many modifications of vortex methods. The key difference between them is the approach to the viscous forces modeling. From this point of view, stochastic and deterministic algorithms can be distinguished. The stochastic approach, for example, the random walk method (called also Random Vortex Method) [1] was one of the first methods of taking viscosity effects into account. According to it, the diffusion of vorticity is simulated by a stochastic process. There are two deterministic approaches: the methods of circulations redistribution and the diffusive velocity methods. Among the circulations redistribution method, the Particle Strength Exchange method [2] and Vorticity Redistribution Method [3] can be pointed out. PSE is a quite popular method, and there are various two- and three-dimensional implementations [4, 5, 6, 7, 8].

In the present paper, only two dimensional flows are considered, and deterministic approach is used, called the Viscous Vortex Domains method (VVD) [9, 10], which is pure Lagrangian and belongs to a class of diffusive velocity methods. According to this approach, the vortex particles retain their circulations and move according to the velocity field, which is a superposition of the convective and diffusive velocities.

Despite the fact that there are number of scientific groups involved in vortex methods development, today there is a very limited number of available codes implementing vortex methods. Mostly, such implementations are in-house codes, which are used by a narrow circle of specialists. It is easy to found the `vvflow` code [11], implemented by scientists led by prof. G.Ya. Dynnikova in Moscow State University. The `vvflow` can be downloaded for free as executable (binary) application; the source code is not available, so it is impossible to study it and introduce any modifications.

This paper discusses the authors implementation of the VVD method — the `VM2D` code, which is open source and available on the `GitHub` platform [12]. The `VM2D` code is cross-platform and it has a modular structure. Parallel algorithms in `VM2D` are implemented using OpenMP, MPI and Nvidia CUDA technologies, that allows performing computations on multiprocessor systems with classical (CPU) architecture and using GPU accelerators. In `VM2D` it is possible to simulate incompressible flows around airfoil or system of airfoils, including transient regimes, to calculate hydrodynamic loads acting the airfoils, to solve fluid-structure interaction (FSI) problems when the airfoils move under the hydrodynamic loads. It is also possible to simulate internal flows.

The aim of this paper is to verify the `VM2D` code on number of model problems with intermediate Reynolds numbers ($\mathrm{Re} \sim 10^2 \dots 10^3$) and to compare its efficiency with the `OpenFOAM` code, which implements the finite volume method. The structure of the paper is as follows. The second section briefly describes the main ideas of the VVD method and some specific aspects of its numerical implementation in the `VM2D` code. In the third section, a test problem of the flow simulation around two closely spaced circular airfoils of different diameters is considered. In [13] the results are given for hydrodynamic loads and the Strouhal number, obtained experimentally and numerically using the finite element method. In the last section, the verification of the VM2D code for the simulation of internal flows is performed. For this purpose, the problem of flow simulation in a channel with a backward-facing step [14] is considered.

2 VISCOUS VORTEX DOMAINS METHOD AND VM2D CODE

In the VVD method, as in other vortex methods, the vorticity $\vec{\Omega}(\vec{r},t) = \mathrm{curl}\vec{V}(\vec{r},t)$ is the primary calculated variable. Its distribution in the flow domain is represented by a set of vortex elements, each of them is characterized by its circulation Γ_i and position $\vec{r}_i$. New vorticity, i.e., new vortex elements is generated only on the airfoil surface line (or on the outer boundaries in the case of internal flows). Further in the description of the algorithm only external flows and immovable airfoils are discussed, but all ideas and numerical schemes can be easily transferred to more general cases without significant modifications.

A time-step of the VVD algorithm implemented in the VM2D code can be divided into 4 blocks:

- vorticity generation on the airfoil surface line,
- vorticity transfer from the airfoil surface line to the flow area,
- vortex wake evolution simulation,
- pressure reconstruction and hydrodynamic loads calculation (if necessary).

1. *Vorticity generation.* The vorticity which is generated during a time-step period is simulated by a thin vortex sheet at the airfoil surface line K. Its intensity $\gamma(\vec{r})$ can be found from the no-slip boundary condition, which can be written down in the form of a boundary integral equation [16]

$$\oint_K \frac{\vec{k} \times (\vec{r} - \vec{\xi})}{2\pi|\vec{r} - \vec{\xi}|^2}\gamma(\vec{\xi})dl_\xi - \frac{\gamma(\vec{r})}{2}\vec{\tau}(\vec{r}) = \vec{f}(\vec{r}), \quad \vec{r} \in K, \tag{1}$$

where $\vec{k}$ is unit vector orthogonal to the flow plane, $\vec{\tau}(\vec{r})$ is tangent unit vector, $\vec{f}(\vec{r})$ is a known function depending on the incident flow velocity, the airfoil surface line velocity and vorticity distribution in the flow. There are two possible ways to satisfy vector integral equation (1): by projecting it onto the normal or tangential direction on the airfoil surface line.

In the original VVD method, the projection onto the normal direction is used [9]; however it is possible to achieve higher accuracy by projecting (1) onto tangent direction [17, 18]. In the VM2D code “tangent” approach is implemented and the following boundary integral equation is solved:

$$\oint_K \frac{(\vec{r} - \vec{\xi}) \cdot \vec{n}(\vec{\xi})}{2\pi|\vec{r} - \vec{\xi}|^2}\gamma(\vec{\xi})dl_\xi - \frac{\gamma(\vec{r})}{2} = \vec{f}(\vec{r}) \cdot \tau(\vec{r}), \quad \vec{r} \in K.$$

There are several numerical schemes for the numerical solution of this equation based on the Galerkin method and various approaches to airfoil surface line discretization [17, 18, 19]. In this paper, one of the simplest schemes is used, based on discretization of the airfoil surface line by a polygon and a piecewise-constant representation of the solution [17].

2. To *transfer the vorticity from the airfoil surface line to the flow domain* the distributed vorticity, which forms the vortex sheet at the airfoil surface line, is transformed into separate vortex elements (VE). They become part of the vortex wake.

3. *Vortex wake evolution simulation.* According to the VVD method, vortex elements in the flow with circulations Γ_i and positions $\vec{r}_i$, $i = 1, \dots, N$, move along the velocity field $(\vec{V} + \vec{W})$:

$$\frac{d\vec{r}_i}{dt} = \vec{V}(\vec{r}_i) + \vec{W}(\vec{r}_i), \quad i = 1, \dots, N,$$

where $\vec{V}$ is convective velocity and $\vec{W}$ is so called diffusive velocity. Convective velocity can be calculated from the vorticity distribution using the Biot — Savart law

$$\vec{V}(\vec{r}) = \vec{V}_\infty + \int_S \frac{\vec{\Omega}(\vec{r}) \times (\vec{r} - \vec{\xi})}{2\pi|\vec{r} - \vec{\xi}|^2} dS_\xi + \oint_K \frac{\vec{k} \times (\vec{r} - \vec{\xi})}{2\pi|\vec{r} - \vec{\xi}|^2} \gamma(\vec{\xi}) dl_\xi =$$

$$= \vec{V}_\infty + \sum_{i=1}^{N} \frac{\vec{k} \times (\vec{r} - \vec{r}_i)}{2\pi|\vec{r} - \vec{r}_i|^2} \Gamma_i dS_\xi + \oint_K \frac{\vec{k} \times (\vec{r} - \vec{\xi})}{2\pi|\vec{r} - \vec{\xi}|^2} \gamma(\vec{\xi}) dl_\xi, \quad \vec{r} \in S,$$

where $\vec{V}_\infty$ is incident flow velocity. Note, that direct calculation of the vortex elements velocities through the Biot — Savart law is time-consuming procedure for large number of vortices, so in practice fast approximate methods are used, which have logarithmic ($\sim N \log N$) complexity against quadratic one ($\sim N^2$).

The diffusive velocity [20]

$$\vec{W}(\vec{r}) = -\nu \frac{\nabla \vec{\Omega}(\vec{r})}{\Omega(\vec{r})}$$

is proportional to the flow viscosity ν and depends both on the vorticity distribution in the flow domain in a neighborhood to the point r and on the shape of the flow region boundary (if there is such in a neighborhood to r).

4. *Hydrodynamic loads calculation.* In order to reconstruct the pressure distribution in the flow domain, an analogue of the Cauchy — Lagrange integral can be used [21]. However, in practice, as a rule it is necessary to determine hydrodynamic loads (forces and torque) acting the airfoil in the flow. It is possible to use for this purpose the integral formulae derived by prof. G.Ya. Dynnikova and adapted to several types of problems being solved by means of vortex methods [10, 15, 22]:

- flow around an immovable airfoil;
- flow around a rigid airfoil in translational motion;
- flow around a rigid airfoil in rotational motion;

- flow around a rigid airfoil in arbitrary motion.

These integral formulae have been obtained by analytical integration of the pressure distribution over the airfoil surface line. There is also approximate formula for viscous stresses computation.

The `VM2D` source code is written in C++ and has a modular structure. It is a cross-platform code and can be compiled under Windows, Linux and MacOS by using MSVC, GCC, Intel C++ Compiler, Clang compilers (as well as other ones supporting the C++11 standard). The `Eigen` external library is used in `VM2D` for the numerical solution of linear equations systems. The OpenMP, MPI and Nvidia CUDA technologies are used for computation acceleration on multi-core and multiprocessor cluster systems, including hybrid architectures with graphic accelerators [23].

A detailed description of the code structure and instructions for compiling and running can be found in [12, 24]. There is also doxygen-documentation for the VM2D [12], to date only in Russian.

3 FLOW SIMULATION AROUND TWO CLOSELY SPACED CIRCULAR AIRFOILS

A series of model problems of the external flow simulation around two circular airfoils with different mutual positions is considered (fig. 1). The angle α varies from $0°$ to $180°$, as in [23], where numerical and experimental results for such problems are presented. The simulations were carried out for the regime with intermediate Reynolds number, namely $\mathrm{Re} = 10^3$. The Reynolds number is calculated with respect to the diameter of a large cylinder: $\mathrm{Re} = \frac{\rho D V_\infty}{\nu}$.

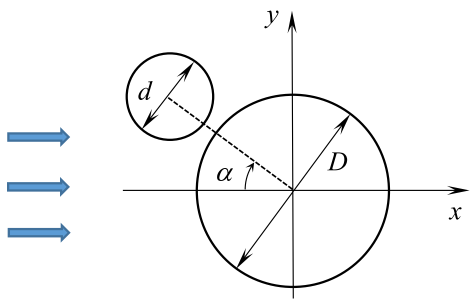

Figure 1: Mutual position of two circular cylinders with different diameters

For simulation in `OpenFOAM`, meshes with different number of cells were used: 50 000 cells, 150 000 cells, 450 000 cells. Computations in `VM2D` were performed with different discretization: the large circle was discretized into 250, 500 and 1000 elements (panels), which corresponds to approximately 15 000, 29 000 and 100 000 vortex elements in the vortex wake in steady-state mode, respectively.

The results obtained with different discretization show that for both codes the most coarse discretization is sufficient to obtain results that are in acceptable agreement with the results of 2D simulation in [13].

The following characteristics were investigated: the average values and the root mean square (RMS) amplitudes for the drag coefficient C_D and the lift coefficient C_L and the dimensionless vortex shedding frequency St. The drag and lift coefficients for the large cylinder are calculated as $C_{D1} = \frac{2F_{D1}}{\rho D V_\infty^2}$ and $C_{L1} = \frac{2F_{L1}}{\rho D V_\infty^2}$, respectively, and those on the small cylinder by $C_{D2} = \frac{2F_{D2}}{\rho d V_\infty^2}$ and $C_{L2} = \frac{2F_{L2}}{\rho d V_\infty^2}$, respectively, where F_D and F_L are the drag and lift forces acting the cylinder in the x- and y-direction, respectively, the subscripts 1 and 2 represent the large and small cylinders, respectively.

In the figures 2–3, average values of drag and lift coefficients are shown for different values of α (for the most coarse mesh/discretization) in comparison to the data given in [13].

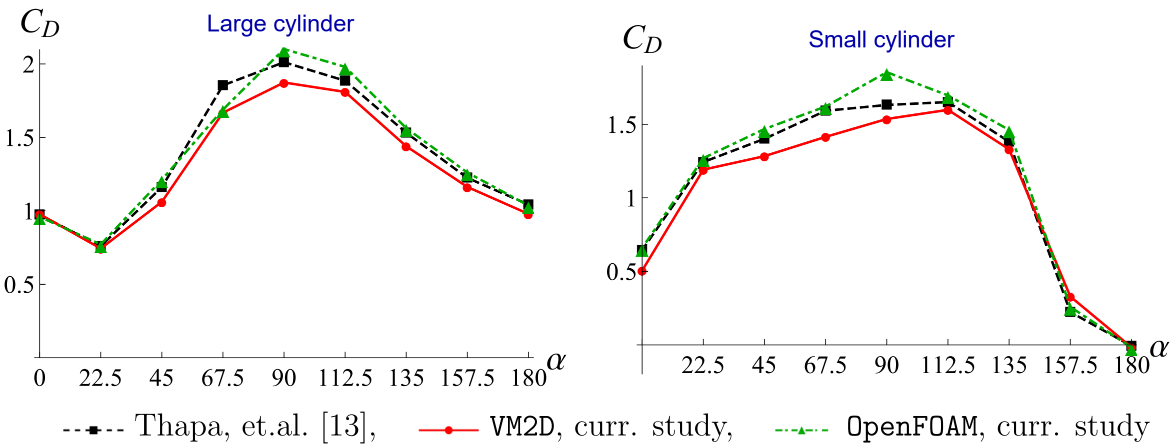

Figure 2: Average values of drag coefficients for large and small cylinders

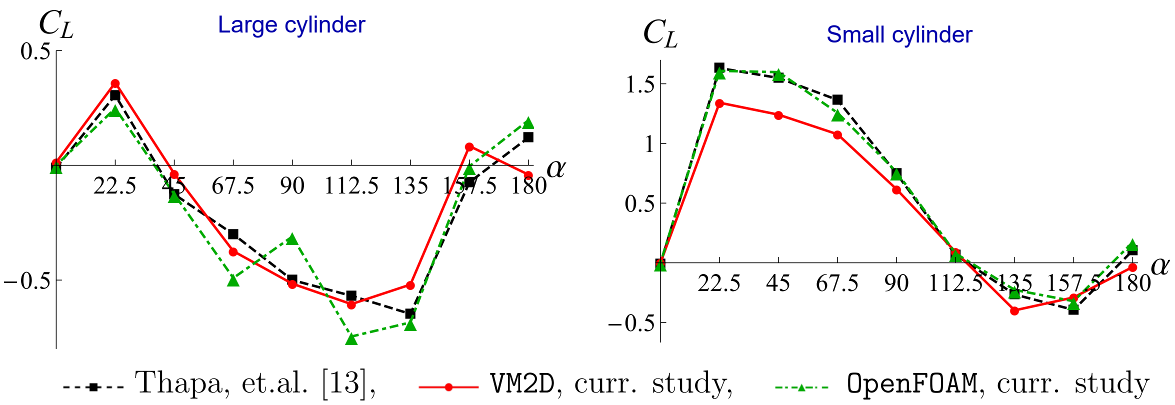

Figure 3: Average values of lift coefficients for large and small cylinders:

It is seen that the results obtained in OpenFOAM and VM2D are in acceptable agreement with the data given in [13]. The highest difference between VM2D and [13] results is observed for RMS of C_L for the small cylinder, while the graph for OpenFOAM is nearly the same as the results [13]. However, for a large cylinder, the results for RMS of C_L obtained in `VM2D` and [13] correlate well enough, while `OpenFOAM` gives a notable error and incorrect tendency of dependence on the angle.

The vortex shedding frequency is determined by applying the Fast Fourier Transform to the lift coefficients of the large cylinder and choosing the dominant frequency from the spectra. The dependency of the Strouhal number, calculated as $\mathrm{St} = fD/V_\infty$, where f is the dominant frequency of the oscillation of the lift coefficients, is shown in the figure 4.

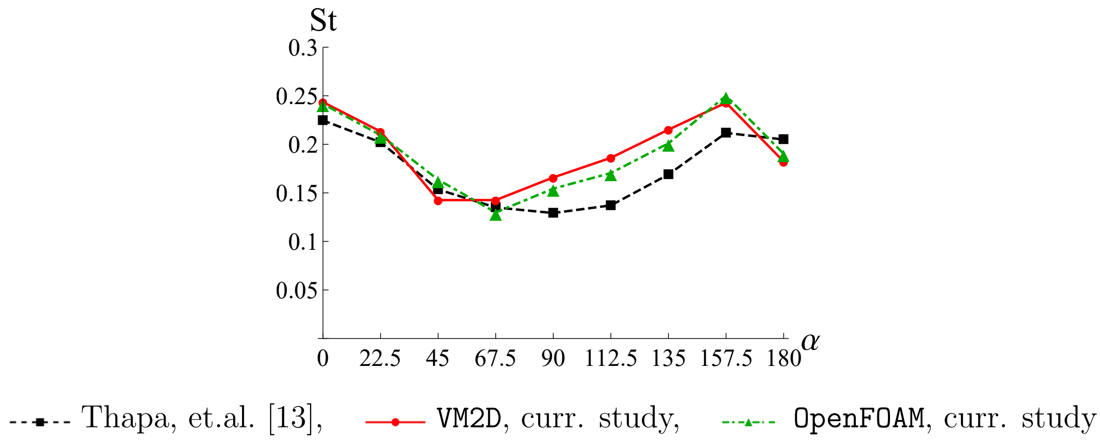

Figure 4: The dimensionless vortex shedding frequency

It is seen that the results obtained in `VM2D` and `OpenFOAM` agree with each other quite well, but slightly differ from the results [13].

The `VM2D` allows performing computations using GPU accelerators. Such technology is highly efficient due to the fact that vortex methods are particle methods. For the discussed model problem, the computations were carried out using two GPU accelerators: Nvidia GeForce GTX970 (with rather small peak performance, approx. 100 Gflops) and Nvidia Tesla V100 (flagship GPU accelerator for today, approx. 7 Tflops in peak). Simulations in `OpenFOAM` were performed using 12 CPU Intel Xeon X5670 (2.93 GHz).

The table 1 shows times of computations for simulations in `OpenFOAM` and `VM2D` using various discretization. Each simulation was performed for time period $T = 0 \dots 150$. In `VM2D`, the time step was set manually, and in `OpenFOAM` it was selected automatically. For `VM2D`, number N of elements discretizing the airfoil surface line, the approximate number N_{wake} of vortex elements in the wake, and the number of time steps are specified. For `OpenFOAM`, the number N of mesh cells is specified.

It can be seen that simulations in `VM2D` using 1 GPU accelerator GeForce GTX970 with rather low performance require less computational time than simulations in `OpenFOAM` using 12 CPUs. When using a video card Tesla V100, simulations in `VM2D` require 5–10 times less computational time than simulations in `OpenFOAM`.

Table 1: The computation times for simulations in `VM2D` and `OpenFOAM`

VM2D					OpenFOAM		
N	N_{wake}	steps	Time GTX970	Time V100	N	steps	Time
250	13 500	18750	50 min	11 min	45 000	60 000	58 min
500	29 000	37500	300 min	41 min	125 000	120 000	340 min

4 FLOW SIMULATION IN A CHANNEL WITH A BACKWARD-FACING STEP

In this section, `VM2D` will be tested for the case of internal flow simulation. To this purpose, the flow inside the channel with a backward-facing step is considered. In [14], the results of experiments and numerical simulations are shown for this problem. The form of the channel is shown in the figure 5.

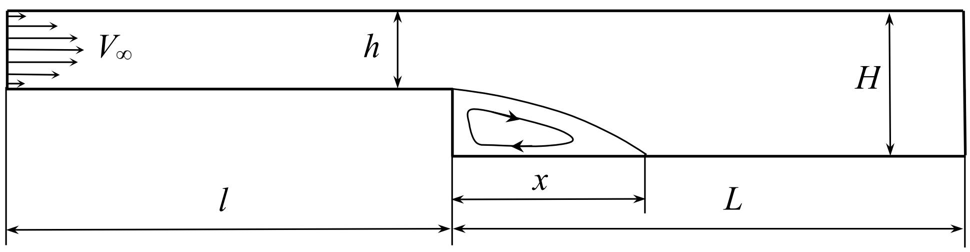

Figure 5: Scheme of the channel with backward-facing step

The channel with a backward-facing step has the following parameters: the height of the input part of the channel $h = 1$, the height of the output part $H = 1.94$. The figure 5 schematically shows the point of flow reattachment, the distance from step to this point is noted by x. An example of the flow for the $\mathrm{Re} = 100$ is shown in the figure 6. The Reynolds number is calculated as

$$\mathrm{Re} = \frac{VD}{\nu},$$

where V is two-thirds of the measured maximum inlet velocity, D is the hydraulic diameter of the inlet (small) channel and is equivalent to twice its height, $D = 2h$, and ν is the kinematic viscosity.

We will compare the positions of the flow reattachment point obtained from the simulations in `VM2D` and from experiment in [14] for different Reynolds $(50 \ldots 400)$ numbers. In order to eliminate the influence of the front and back boundaries of the region, simulations were performed for channels with different lengths of the input and output parts l and L, respectively: $l = 7$, $L = 13$; $l = 10$, $L = 19$; and $l = 15$, $L = 28$. The simulations show that in all cases the results for the position of the point of flow reattachment are approximately the same.

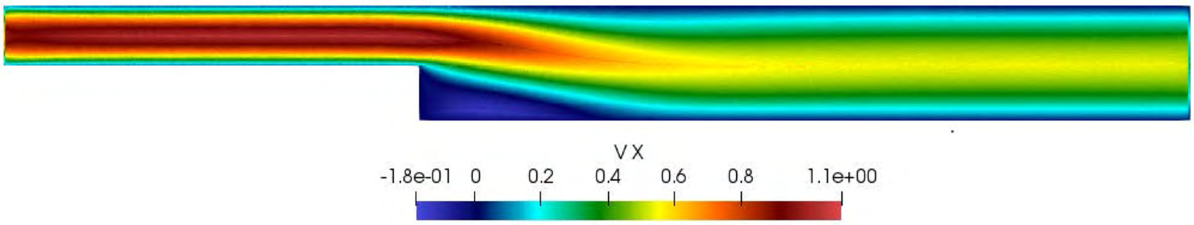

Figure 6: x-component of the flow velocity in the channel with backward-facing step

The figure 7 shows the dependency of the value $s = \frac{x}{H-h}$ on the Reynolds number. It can be seen that the results are in a very good agreement in the interval $\mathrm{Re} = 50 \ldots 300$. There is a significant difference between the results when $\mathrm{Re} > 300$, due to the fact that, as also noted by the authors of [14], the flow becomes three-dimensional and two-dimensional simulation is incorrect.

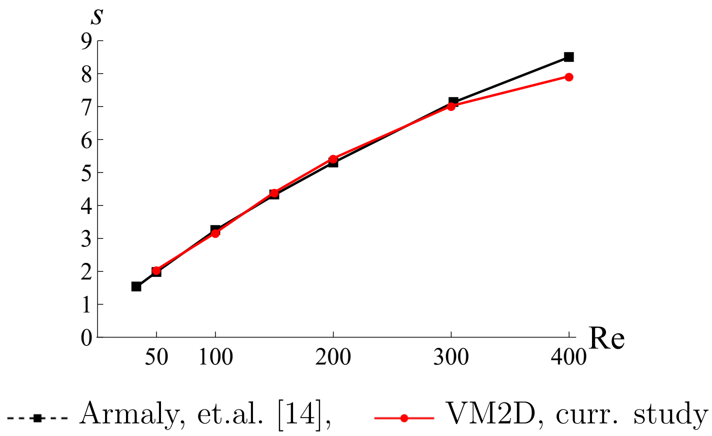

Figure 7: Positions of the flow reattachment point s for different Reynolds numbers

5 CONCLUSIONS

In this paper, using the model problems, the author's software package `VM2D` is verified for two-dimensional incompressible flows simulation. In `VM2D`, the Viscous Vortex Domains method is implemented, which is pure Lagrangian vortex method. For the problem of flow simulation around two circular closely spaced airfoils, the computational efficiency of `VM2D` was compared with the `OpenFOAM` code, where the Finite Volume Method is implemented. It is shown that for such problem, simulations in `VM2D` using one `GPU` accelerator take the same or even less time than simulations in `OpenFOAM` using dozens of CPU cores.

`VM2D` was also verified for internal flows. The flow in a channel with a backward-facing step was considered. The comparison of the numerical and experimental data for the flow reattachment point behind the step show a very good agreement between the results.

The research is supported by Russian Science Foundation (Project No. 17-79-20445).

REFERENCES

[1] Chorin, A.J. Numerical study of slightly viscous flow. *J. Fluid Mech.* (1973) **57(4)**:785–796.

[2] Raviart, P-A. *Méthodes particulaires.* Lecutre Notes, Ecole d'été d'analyse numérique, Centre d'étude du Bréau-sans-nappe, France, (1987).

[3] Shankar, S. and VanDommelen, L. A new diffusion procedure for vortex methods. *J.Comput.Phys.* (1996) **127(1)**:88–109.

[4] Cottet, G.-H., Michaux, B., Ossia, S. and VanderLinden, G. A comaprison of spectral and vortex methods in three-dimensional incompressible flows. *J. Comput. Phys.* (2002) **175**:702–712.

[5] Najm, H.N. A Hybrid Vortex Method with Deterministic Diffusion. In: *Vortex Flows and Related Numerical Methods.* Eds: Beale, J.T., Cottet, GH., Huberson, S. NATO ASI Series (Series C: Mathematical and Physical Sciences), Springer, Dordrecht. (1993) **395**:207–222.

[6] Ploumhans, P., Winckelmans, G.S. and Salmon, J.K. Vortex particles and tree codes: I. Flows with arbitrary crossing between solid boundaries and particle redistribution lattice; II. Vortex ring encountering a plane at an angle, *ESAIM Proc.* (1999) **7**:335–348.

[7] Eldredge, J.D., Leonard, A. and Colonius, T. A general deterministic treatment of derivatives in particle methods. *J. Comput. Phys.* (2002) **180**:686–709.

[8] Kotsur, O.S. and Shcheglov, G.A. Implementation of the particle strength exchange method for fragmentons to account for viscosity in vortex element method. *Herald of the Bauman Moscow State Technical University, Series Natural Sciences.* (2018) **3**:48–67.

[9] Dynnikova, G.Ya. The lagrangian approach to solving the time-dependent navier-stokes equations. *Doklady Physics.* (2004) **49(11)**:648–652.

[10] Andronov, P.R., Grigorenko, D.A., Guvernyuk, S.V. and Dynnikova, G.Ya. Numerical simulation of plate autorotation in a viscous fluid flow. *Fluid Dynamics.* (2007) **42(5)**:719–731.

[11] *Vvflow CFD Suite (stable).* URL:https://packagecloud.io/vvflow/stable.

[12] *VM2D: Vortex method for 2D flow simulation.* URL: https://github.com/vortexmethods/VM2D.

[13] Thapa, J., Zhao, M., Cheng, L. and Zhou, T. Three-dimensional flow around two circular cylinders of different diameters in a close proximity. *Physics of Fluids.* (2015) **27**:085106.

[14] Armaly, B.F., Durst, F., Pereira, J.C.F. and Schonung, B. Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech.* (1983) **127**:473–496.

[15] Guvernyuk, S.V. and Dynnikova, G.Ya. Modeling the flow past an oscillating airfoil by the method of viscous vortex domains. *Fluid Dynamics.* (2007) **42(1)**:1–11.

[16] Kempka, S.N., Glass, M.W., Peery, J.S., Strickland, J.H. and Ingber, M.S. Accuracy considerations for implementing velocity boundaryconditions in vorticity formulations. *SANDIA report.* (1996) **SAND96-0583, UC-700**.

[17] Kuzmina, K.S., Marchevskii, I.K. and Moreva, V.S. Vortex Sheet Intensity Computation in Incompressible Flow Simulation Around an Airfoil by Using Vortex Methods. *Mathematical Models and Computer Simulations.* (2018) **10(3)**:276–287.

[18] Kuzmina, K.S., Marchevskii, I.K., Moreva, V.S. and Ryatina, E.P. Numerical scheme of the second order of accuracy for vortex methods for incompressible flow simulation around airfoils. *Russian Aeronautics.* (2017) **60(3)**:398–405.

[19] Marchevsky, I., Kuzmina, K. and Soldatova, I. Improved algorithm of boundary integral equation approximation in 2D vortex method for flow simulation around curvilinear airfoil. *Mathematics and Mathematical Modeling.* (2018) **6**:22–51.

[20] Dynnikova, G.Ya. Vortex motion in two-dimensional viscous fluid flows. *Fluid Dynamics.* (2003) **38(5)**:670–678.

[21] Dynnikova, G.Y. An analog of the Bernoulli and Cauchy-Lagrange integrals for a time-dependent vortex flow of an ideal incompressible fluid. *Fluid Dynamics.* (2000) **35(1)**:24–32.

[22] Dynnikova, G.Y. and Andronov, P.R. Expressions of force and moment exerted on a body in a viscous flow via the flux of vorticity generated on its surface. *European Journal of Mechanics, B/Fluids.* (2018) **72**:293–300.

[23] Kuzmina, K.S., Marchevsky, I.K. and Ryatina, E.P. On CPU and GPU parallelization of VM2D code for 2D flows simulation using vortex method. *6th European Conference on Computational Mechanics (ECCM 6) 7th European Conference on Computational Fluid Dynamics (ECFD 7)*, 11–15 June 2018, Glasgow, UK. 2390–2401.

[24] Kuzmina, K.S., Marchevsky, I.K. and Ryatina, E.P. Open source code for 2D incompressible flow simulation by using meshless lagrangian vortex methods. *Proceedings of Ivannikov ISPRAS Open Conference.* (2017) 97–103.