Parallel Computations on GPU in 3D Vortex Particle Method

Andrzej Kosior, Henryk Kudela

Institute of Aerospace, Process and Power Machines Engineering,
Wroclaw University of Technology, 50-370 Wroclaw, Wybrzeze Wyspianskiego 27, Poland,
andrzej.kosior@pwr.wroc.pl, henryk.kudela@pwr.wroc.pl

Introduction

MANY problems in fluid mechanics can be analyzed using the vorticity and its evolution in time and space. From this fact results a great importance of methods that are based on the vortex particle method. In this method particles that carry information about vorticity are used. Tracing positions of these particles allows one to study evolution of the vorticity. It is well known that distribution of the vorticity permits on calculation of the fluid velocity. In vortex particle method the particles after several steps have a tendency to concentrate in areas where velocity gradient is very high. It may lead to spurious vortex structures. To avoid this situation after arbitrary number of time steps the redistribution of particles to regular positions is done. In 2D case we noted [4, 5, 6] that it is useful to do the remeshing at every time step. At the beginning the vortex particles are put on the grid nodes. After displacement in every time step the intensities of the particles are redistributed again to the initial grid nodes. This strategy has several advantages like shortening of computational time and accurate simulation of the viscosity. In present paper we implemented this idea to the case of 3D flow. Due to a very long time of computations we found that speed-up given by parallel computing is necessary. The VIC method is very well suited for parallel computation. The Poisson equation for each component of the vector potential can be solved independently. The remeshing process which has to be done in each time step has a local character and the computations for each particle can be done independently. Also the displacement of the vortex particle has a local character. To speed-up calculations we decided to use multicore architecture of the graphics card (GPU). To find out how large speed-up can be obtained we decided to conclude some tests.

Equations of motion

Equations of incompressible viscous fluid motion have the following form:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{u} \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{2}$$

where $\mathbf{u} = (u, v, w)$ is velocity vector, ρ – fluid density, p – pressure, ν – kinematic coefficient of viscosity. The equation (1) can be transformed to the Helmholtz equation for the vorticity evolution:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u}\cdot\nabla)\boldsymbol{\omega} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{u} + \nu\Delta\omega \tag{3}$$

where $\boldsymbol{\omega} = \nabla\times\mathbf{u}$. From incompressibility (2) stems the existence of the vector potential $\mathbf{A}$

$$\mathbf{u} = \nabla\times\mathbf{A} \tag{4}$$

Assuming additionally that vector $\mathbf{A}$ is incompressible ($\nabla\cdot\mathbf{A} = 0$) its components can be obtained by solution of the Poisson equation

$$\Delta A_i = -\omega_i, \qquad i = 1, 2, 3 \tag{5}$$

Description of VIC method for three-dimensional case

In this work we used so called viscous splitting algorithm. In this algorithm the solution is found in two substeps. First we solve motion equations for inviscid flow. To do so we have to discretize our computational domain. We set up a regular 3D grid $(j_1\Delta x, j_2\Delta y, j_3\Delta z)$ $(j_1, j_2, j_3 = 1, 2, \ldots, N)$, where $\Delta x = \Delta y = \Delta z = h$. The same mesh will be used for solving Poisson equation. The continuous field of vorticity is replaced by a discrete distribution of Dirac delta measures [2, 7]

$$\boldsymbol{\omega}(\mathbf{x}) = \sum_{p=1}^{N} \boldsymbol{\Gamma}_p(\mathbf{x}_p)\delta(\mathbf{x}-\mathbf{x}_p) \tag{6}$$

where $\boldsymbol{\Gamma}_p$ means vorticity particle $\boldsymbol{\Gamma}_p = (\Gamma_{p1}, \Gamma_{p2}, \Gamma_{p3})$ at position $\mathbf{x}_p = (x_{p1}, x_{p2}, x_{p3})$. The domain of the flow is covered by numerical mesh $(N_x \times N_y \times N_z)$ with equidistant spacing h, and the i-th component of the vector particle $\boldsymbol{\alpha}$ is defined by expression:

$$\Gamma_i = \int_{V_p} \omega_i(x_1, x_2, x_3), d\mathbf{x} \approx h^3\omega_i(\mathbf{x}_p), \quad \mathbf{x}_p \in V_p, \quad |V_p| = h^3 \tag{7}$$

From the Helmholtz theorems [11] we know that the vorticity is carried on by the fluid:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}(\mathbf{x}_p, t), \tag{8}$$

We must take into account also the fact that due to three dimensionality of the vorticity field the intensity of the particles are changed by the stretching effect

$$\frac{d\boldsymbol{\Gamma}_p}{dt} = [\nabla\mathbf{u}(\mathbf{x}_p, t)]\cdot\boldsymbol{\Gamma}_p \tag{9}$$

Velocity at the grid nodes was obtained by the solution of the Poisson equation (5) by the finite difference method and using (4). The system of equations (8), (10) was solved by Runge–Kutta method of the 4-th order.

In the second substep of viscous splitting algorithm we simulate viscosity effect on particles by solving the equation:

$$\frac{d\boldsymbol{\omega}}{dt} = \nu\nabla\boldsymbol{\omega} \tag{10}$$

Remeshing

In Vortex in Cell method particles have a tendency to gather in regions with high velocity gradients. This can lead to inaccuracies as particles coming too close to one another. To overcome this problem particles have to be remeshed that is distributed back to the nodes of the rectangular mesh. It is done by an interpolation:

$$\omega_j = \frac{1}{h^3}\sum_p \tilde{\Gamma}_{pn}\varphi\left(\frac{\mathbf{x_j} - \tilde{\mathbf{x}}_p}{h}\right) \tag{11}$$

where j is index of grid node and p is index of a particle. The quality of that interpolation strongly depend on the properties of the kernel φ. In this work we used following interpolation kernel:

$$\varphi(x) = \begin{cases} (2 - 5x^2 + 3|x|^3)/2 & \text{if } 0 \le |x| \le 1 \\ (2-|x|)^2(1-|x|)/2 & \text{if } 1 \le |x| \le 2 \\ 0 & \text{if } 2 \le |x| \end{cases} \tag{12}$$

Kernel (12) used in this work is of order $m = 3$.

Multigrid Poisson Equation Solver

For solution of the Poisson equation a Multigrid solver was created.
The 3D Poisson equation for the l-component of the vector potential in Cartesian coordinates has the form:

$$\frac{\partial^2 A_l}{\partial x^2} + \frac{\partial^2 A_l}{\partial y^2} + \frac{\partial^2 A_l}{\partial z^2} = -\omega_l \tag{13}$$

A seven point stencil is used for discretization of the Poisson equation on a rectangular grid $(ih_x, jh_y kh_z)$. If we assume additionally that grid steps in each direction are equal $(h_x = h_y = h_z = h)$ we can rewrite equation (13) in the form:

$$\psi_{i,j+1,k} + \psi_{i+1,j,k} + \psi_{i,j,k+1} - 6\psi_{i,j} + \psi_{i-1,j,k} + \psi_{i,j-1,k} + \psi_{i,j,k-1} = -\omega_{i,j,k}h^2 \tag{14}$$

where

$$i = 0, 1, 2, \ldots, N_x \qquad j = 0, 1, 2, \ldots, N_y \qquad k = 0, 1, 2, \ldots, N_z$$

This way we obtain set of algebraic equations with unknown vector of stream function $\psi_{i,j}$ at grid nodes. This set of equations was solved by Gauss-Seidel, SOR and Multigrid iterative methods.
The Gauss-Seidel method is easy to parallelize but cannot compete with the Fast Poisson Solvers (used for calculation on a single core). In the multigrid method one uses the fact that Gauss-Seidel method is smoothing the error. The best smoothing relate to the highest frequency error components. A smooth error is well approximated on a coarse grid. Gauss-Seidel method on different grid levels gives rapid reduction of the corresponding high frequency components and as this process covers all frequencies, a rapid reduction of the overall error can be achieved. It is very efficient linear solver [10]. Computations can be done in so called V-cycle which can be seen in Figure 1.

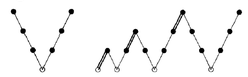

Figure: *Different types of computational cycles used in multigrid method. Left - V-cycle, right - full multigrid cycle*

An initial smoothing of the error for iterative solvers like multigrid, can be obtained by nested iteration. The general idea behind the nested iteration is to provide an initial approximation to the finest grid from the coarser grids. It simply means that a lower (coarser) discretization level is used in order to provide a good initial approximation for the iteration on the next higher (finer) level. This combination is called the Full Multigrid (FMG). Typically, the FMG scheme is the most efficient multigrid version [10]. FMG cycle can be seen in Figure 1.
To test the presented algorithms we solved three-dimensional Poisson equation with known solution:

$$\psi(x, y, z) = sin(2\pi x)\cdot sin(2\pi y)\cdot sin(2\pi z), x, y, z \in [0, 1] \tag{15}$$

with Dirichlet boundary conditions.
We carried out the computation on different grids and the time of computation we compared with the Fast Poisson Solver executed on single CPU core. A stop criteria for multigrid method was that the maximal residual was lower then $1.0E-8$. Final error of calculations is presented in the Table 1 and the speed-up is presented the Figure 2. Computations were performed on: CPU (Intel Core i7 960), and GPU (NVIDIA GTX 480).

Table: *Final error of different methods with Dirichlet boundary condition.*

Number of nodes	FES(CPU)	MGD(GPU)	FMG(GPU)
65×65×65	8.0358e-4	8.0358e-4	8.1833e-4
129×129×129	2.0082e-4	2.0082e-4	2.0374e-4
257×257×257	5.0201e-5	5.0201e-5	5.0757e-5

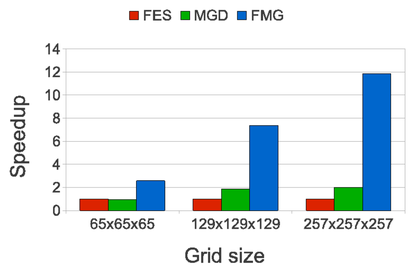

Figure: *Speed-up of Multigrid method and Full multigrid method comparing to Fast Poisson Elliptic with Dirichlet boundary conditions*

Abbreviations used in Table 1 and Figure 2: FES - Fast Elliptic Solver, MGD - Multigrid method, FMG - Full multigrid method.

Tests

We carried out numerical experiments to validate the code. The test was the evolution of the vortex ring with constant vorticity distribution in its core in inviscid fluid. The parameters of the vortex ring $R_0 = 1.5$ and $\epsilon_0 = 0.3$ were constant but $\Gamma \in [0.23, 0.94]$ was changed.
Computational domain was $2\pi \times 2\pi \times 2\pi$ and a numerical grid had 129 nodes in each direction. It was assumed the periodic boundary conditions in all directions.
Initial condition was given in a form that every node which fulfils equation:

$$r_p = \left(\sqrt{x^2+y^2} - R_0\right)^2 + z^2 < \epsilon_0^2 \tag{16}$$

obtain initial vorticity. The initial number of particles used was 179780.
Time step for iteration of equation (8) was equal to $\triangle t = 0.01$.
The results can be seen in the Figure 3.
In the second test we put two vortex rings of the same dimensions as in the first test in the viscous fluid. The rings were given opposite circulation in order to approach one another.
The results of the second test can be seen in the Figure 4.

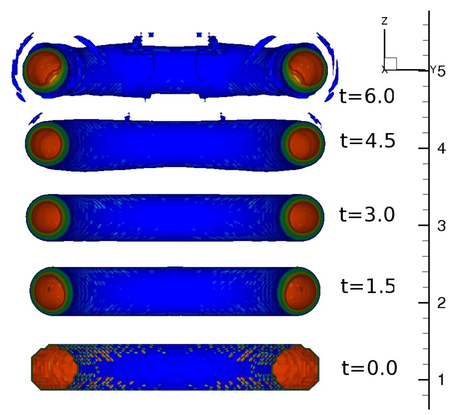

Figure: *Vortex ring evolution in inviscid fluid for the case $\Gamma = 4.23$. Sequence of iso–vorticity surface for $|\omega| \in [6, 14]$ at different time t*

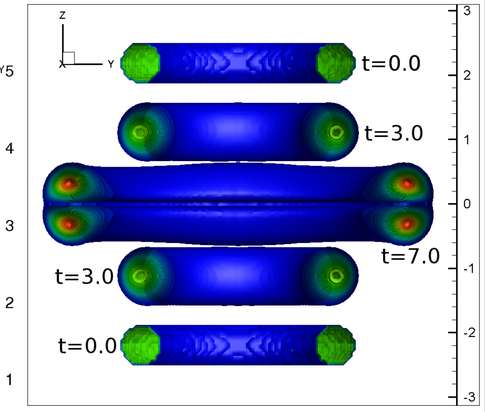

Figure: *Two vortex ring evolution in viscous fluid for the case $\Gamma = 2.5$ and $\nu = 0.001$. Sequence of iso–vorticity surface for $|\omega| \in [0.5, 14]$ at different time t*

Conclusions

From presented results we can conclude that Vortex Particle Methods are very well suited for numerical studies of 3D vortex structures. In our work we use the multicore architecture for parallel computing. It speeds up the calculations about 46 times with respect to a single processor. Distribution of vorticity inside the core is important element of the behaviour of the vortex ring. The initial uniform vorticity distribution evolve in time approaching non-uniform distribution resembling the Gaussian one.

Bibliography

[1] Braide B. *A Friendly Introduction to Numerical Analysis*, Pearson Prentice Hall, 2006,

[2] Cottet G. H., Koumoutsakos P. D., *Vortex Methods: Theory and Practice*, Cambridge University Press, 2000,

[3] Green S. I. *Fluid Vortices* , Springer, 1995,

[4] Kudela H., Malecha Z. M. *Viscous Flow Modeling Using the Vortex Particles Method*, Task Quarterly 13 No 1-2, 15-32, 2008,

[5] Kudela H., Malecha Z. M. *Eruption of a boundary layer induced by a 2D vortex patch*, Fluid Dyn. Res. 41, 2009,

[6] Kudela H., Kozlowski T. *Vortex in Cell Method For Exterior Problems*, Journal of Theoretical and Applied Mechanics 47, 4, pp. 779-796, 2009,

[7] Kudela H., Regucki P. *The Vortex-in-Cell Method for the Study of Three-Dimensional Flows by Vortex Methods. In: Tubes, Sheets and Singularities in Fluid Dynamics*, Fluid Mechanics and Its Applications, vol. 7, pp. 49–54, Kluwer Academic Publisher, 2009, .

[8] Saffman P. G. *Vortex Dynamics*, Cambridge University Press, 1992,

[9] Thomas J.W. *Numerical Partial Differential Equations: Conservation Laws and Elliptic Equations*, Springer-Verlag, New York, 1999,

[10] Trottenberg U., Oosterlee C.W., Schuller A. *Multigrid*, Academic Press, London, 2001,

[11] Wu J. Z., Ma H. Y., Zhou M. D. *Vorticity and Vortex Dynamics*, Springer, 2006.

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

DOLNY ŚLĄSK

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Zadanie współfinansowane ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego