XX Polish Fluid Mechanics Conference,
Gliwice, 17-20 September 2012

PARALLEL COMPUTATIONS ON GPU OF VORTEX TUBES RECONNECTION USING THE VORTEX PARTICLE METHOD

Andrzej Kosior[1], Henryk Kudela[1]
[1]*Institute of Aerospace, Process and Power Machines Engineering,*
Wroclaw University of Technology, 50-370 Wroclaw, Wybrzeze Wyspianskiego 27, Poland
E-mail: henryk.kudela@pwr.wroc.pl, andrzej.kosior@pwr.wroc.pl

Abstract

The paper presented the Vortex in Cell (VIC) method for solving the fluid motion equations in 3D. Due to the long time of computation on single processor the parallel implementation of the VIC method was presented. The speed-up for the entire VIC method implementation on the GPU was 46 times. Two test cases were presented. First one, concerning parallel vortex tube reconnection and second concerning two vortex rings reconnection. Results were compared with ones received by spectral methods. The agreement was very good.

Key words: Vortex in cell, reconnection, GPU, parallelization

INTRODUCTION

Vorticity dynamics is one of the most fundamental means of understanding fluid motion, especially at high Reynolds number and in turbulent flows. For an incompressible flow, the vorticity field contain full informTION bout the flow. Although vorticity is more difficult to measure in experiments, particulary in three-dimensional turbulent flows, there are several reasons why the vorticity field is a more fundamental quantity (Kida et al., 1991).

First, in high-Reynolds-number flows, high vorticity regions are more localized in space than velocity. Vorticity, unlike velocity, is a Galilean invariant. A velocity field induced by vorticity in an incompressible flow is obtainable from the Biot-Savart induction equation. Thus it is easier to understand fluid phenomena and to build a model for them, if necessary, in terms of vorticity than velocity. The vortex method, which is one of the most powerful numerical schemes to solve high-Reynolds-number flows, is based upon idealization of concentrated vorticity regions.

Second, turbulent motion is more clearly evident in the vorticity field than in the velocity field. The large-scale organized structures, the so-called coherent structures, which persist for a relatively long times in turbulent flows, are characterized by domains of coherent vorticity and their interactions and evolution can be explained in terms of vortex dynamics. Further, energy dispersion can be expressed in terms of vorticity.

In this work we showed interaction and evolution of vortex tubes and vortex rings.

For simulation we used the Vortex Particle Method. In the 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 an arbitrary number of time steps, the redistribution of particles to regular positions is done. In 2D we noted (Kudela and Malecha, 2008, 2009; Kudela and Kozlowski, 2009) that it is useful to remesh at every time step. At the beginning the vortex particles are put on numerical mesh nodes. After displacement at every time step the intensities of the particles are redistributed again onto the initial mesh nodes. This strategy has several advantages such as the shortening of computational time and the accurate simulation of viscosity. In the present paper we implemented this idea in 3D flow. Since the computations took very long time, we found that speed-up rendered by parallel computing was necessary.

The VIC method is very well suited for parallel computation. The remeshing process, which has to be done at each time step, has a local character and the computations for each particle can be done independently. Also each displacement of the vortex particle has a local character. To speed-up calculations we decided to use the multicore architecture of the graphics card (GPU). To find out just how much speed-up would be obtained, we decided to conclude some tests.

EQUATIONS OF MOTION

Equations of incompressible and inviscid fluid motion have the following form:

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

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

where $\mathbf{u} = (u, v, w)$ is velocity vector, ρ is fluid density, p is pressure. The equation (1) can be transformed into the Helmholtz vorticity transport equation:

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

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

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

Solving (5) one is able to calculate the velocity by formula (4).

In vortex particle methods the viscous splitting algorithm is used. The solution is obtained in two steps: first, the inviscid - Euler equation is solved.

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} \tag{6}$$

In the second step the viscosity effect is simulated by solving the diffusion equation.

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \nu \Delta \boldsymbol{\omega} \tag{7}$$

For the solution of the diffusive equation one can use any suitable method like the Particle Strength Exchange (PSE) method or the Finite Difference method.

DESCRIPTION OF THE VIC METHOD FOR A THREE-DIMENSIONAL CASE

First we have to discretize our computational domain. To do this, we set up a regular 3D numerical mesh $(j_1\Delta x, j_2\Delta y, j_3\Delta z)$ $(j_1, j_2, j_3 = 1, 2, \ldots, N)$, where $\Delta x = \Delta y = \Delta z = h$. The same mesh will be used for solving the Poisson equation. The continuous field of vorticity is replaced by a discrete distribution of the Dirac delta measures (Cottet and Koumoutsakos, 2000; Kudela and Regucki, 2009)

$$\boldsymbol{\omega}(\mathbf{x}) = \sum_{p=1}^{N} \boldsymbol{\alpha}_p(\mathbf{x}_p)\delta(\mathbf{x} - \mathbf{x}_p) \tag{8}$$

where $\boldsymbol{\alpha}_p$ means vorticity particle $\boldsymbol{\alpha}_p = (\alpha_{p1}, \alpha_{p2}, \alpha_{p3})$ at position $\mathbf{x}_p = (x_{p1}, x_{p2}, x_{p3})$. The i-th component of the vector particle $\boldsymbol{\alpha}_i$ is defined by the expression:

$$\alpha_i = \int_{V_p} \omega_i(x_1, x_2, x_3)\,\mathrm{d}\mathbf{x} \approx h^3 \omega_i(\mathbf{x}_p), \quad \mathbf{x}_p \in V_p, \quad |V_p| = h^3 \tag{9}$$

where V_p is the cell volume with index p.

From the Helmholtz theorems (Wu et al., 2006) we know that vorticity in inviscid flow is carried out by the fluid.

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}(\mathbf{x}_p, t) \tag{10}$$

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

$$\frac{d\boldsymbol{\alpha}_p}{dt} = [\nabla \mathbf{u}(\mathbf{x}_p, t)] \cdot \boldsymbol{\alpha}_p \tag{11}$$

The velocity at the numerical mesh nodes was obtained by solving the Poisson equation (5) by the finite difference method and (4). In the sequential implementation, this system of equations was solved with the Fast Poisson Solver (Schwarztrauber, 1984). (In this work we used an implementation from the FISHPACK numerical library). The system of equations (10), (11) was solved by the Runge–Kutta method of 4th order.

Remeshing

In the Vortex-in-Cell method, particles have a tendency to gather in regions with high velocity gradients. This can lead to inaccuracies, as the particles are coming too close to one another. To overcome this problem particles have to be remeshed; that is, they have to be distributed back to the nodes of the rectangular mesh. In practice, remeshing is done after several time steps. Usually the simulation of viscosity is done by solving the diffusion equation using the numerical mesh. For this reason it is better to remesh at every time step. It is done using an interpolation:

$$\omega_j = \sum_p \tilde{\alpha}_{p_n} \varphi\left(\frac{\mathbf{x_j} - \tilde{\mathbf{x}}_p}{h}\right) h^{-3} \tag{12}$$

where j is the index of the numerical mesh node, p is the index of a particle.

Let us assume that $x \in \mathbb{R}$. In this work, we used the following interpolation kernel (Cottet and Koumoutsakos, 2000; Cottet et al., 2002)

$$\varphi(x) = \begin{cases} (2 - 5x^2 + 3|x|^3)/2 & \text{if } 0 \le |x| \le 1 \\ (2 - |x|)^2(1 - |x|)/2 & \text{if } 1 \le |x| \le 2 \\ 0 & \text{if } 2 \le |x| \end{cases} \tag{13}$$

For the 3D case, $\varphi = \varphi(x)\varphi(y)\varphi(z)$.

We require our interpolation kernel to satisfy

$$\sum_p \varphi\left(\frac{\mathbf{x} - \mathbf{x_p}}{h}\right) \equiv 1 \tag{14}$$

The discrepancy between the old (distorted) and new (regular) particle distribution, can be measured by the difference

$$\sum_p \tilde{\alpha}_{p_n} \delta(\mathbf{x} - \tilde{\mathbf{x}}_p) - \sum_p \alpha_{p_n} \delta(\mathbf{x} - \mathbf{x}_p) \tag{15}$$

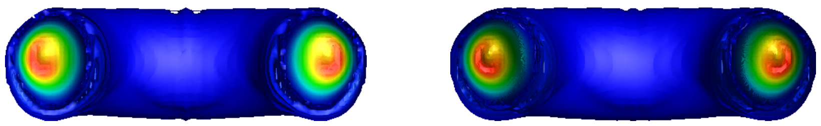

Fig. 1. *Results from the VIC method after 600 time steps ($t = 6.0$) for the case $\Gamma = 1.00$ and 129 nodes in each direction running on different hardware. Left - CPU, right - GPU*

In multiplying (15) by a test function ϕ one can get (Cottet and Koumoutsakos, 2000):

$$E = \sum_p \tilde{\alpha}_{p_n} \phi(\tilde{\mathbf{x}}_p) - \sum_p \alpha_{p_n} \phi(\mathbf{x}_p) \tag{16}$$

where $\tilde{\alpha}_{p_n}$ and $\tilde{\mathbf{x}}_p$ are values from old distribution.

Using (12) we can write:

$$E = \sum_p \tilde{\alpha}_{p_n} \left[\phi(\tilde{\mathbf{x}}_p) - \sum_j \phi(\mathbf{x}_j) \varphi\left(\frac{\mathbf{x}_j - \tilde{\mathbf{x}}_p}{h}\right)\right] \tag{17}$$

To evaluate error E in (17) we have to evaluate the function

$$f(\mathbf{x}) = \phi(\mathbf{x}) - \sum_j \phi(\mathbf{x}_j) \varphi\left(\frac{\mathbf{x}_j - \mathbf{x}}{h}\right) \tag{18}$$

Using (14) the equation (18) can be rewritten as

$$f(\mathbf{x}) = \sum_j [\phi(\mathbf{x}) - \phi(\mathbf{x}_j)] \varphi\left(\frac{\mathbf{x}_j - \mathbf{x}}{h}\right) \tag{19}$$

The Taylor expansion of ϕ will yield:

$$f(\mathbf{x}) = \sum_j \sum_k \left[\frac{1}{k!}(\mathbf{x}_j - x)^k \frac{d^k\phi}{dx^k}\right]^k \varphi\left(\frac{\mathbf{x} - \mathbf{x}_j}{h}\right) \tag{20}$$

We may conclude, that if φ satisfies the following condition:

$$\sum_j (\mathbf{x} - \mathbf{x}_j)^k \varphi\left(\frac{\mathbf{x} - \mathbf{x}_j}{h}\right) \qquad 1 \le |k| \le m - 1 \tag{21}$$

then

$$f(\mathbf{x}) = O\left(h^m\right) \tag{22}$$

and the remeshing will be of the order m. It means that the polynomial functions up to the order m will be exactly represented by this interpolation.

Kernel (13) used in this work is of order $m = 3$.

IMPLEMENTATION ON GPU

Graphics Processing Units that were developed for video games provide cheap and easily accessible hardware for scientific calculations.

An implementation of the Vortex Particle Method on GPU was done. Test concerning movement of a vortex ring in inviscid fluid was carried out. The calculations were done on a single processor (CPU – Intel Core i7 960) and on a graphics card (GPU – NVIDIA GTX 480). The results for computations are shown in the Figures 1 and 2. The discrepancy may stem from

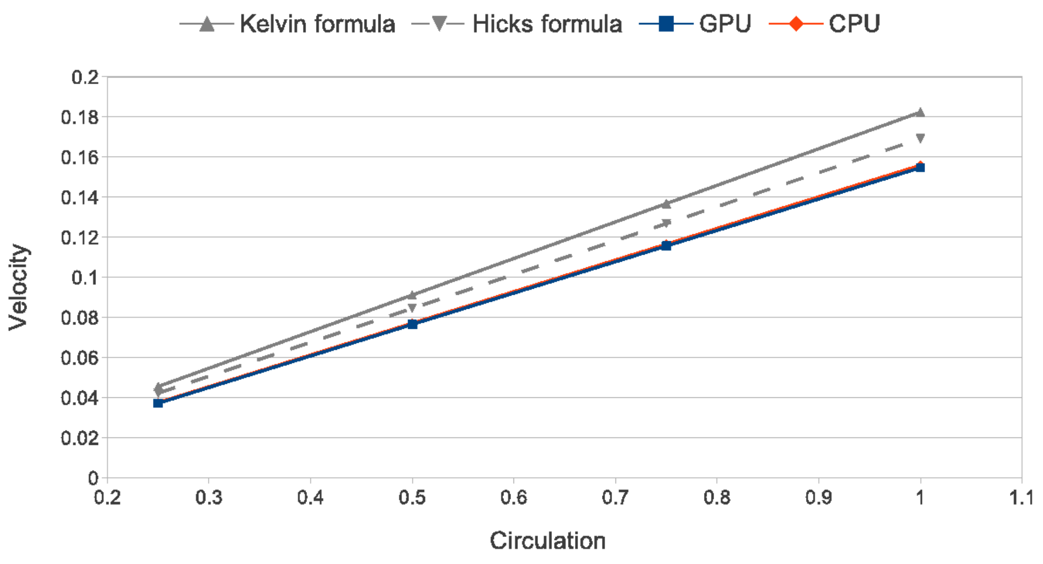

Fig. 2. *Velocity of the vortex ring as a function of circulation for calculation on CPU and GPU*

the fact that the Fast Poisson Solver that was used on CPU solved the algebraic equations exactly. On GPU the iterative Multigrid method was used. This shows the correctness of our implementation on a graphics card.

Execution time for the application running on the GPU is 46 times shorter then the one using CPU. This is a very significant speed up.

More details about this implementation can be found in (Kosior and Kudela, 2012).

VORTEX TUBES RECONNECTION CASE

Understanding the dynamics and the mutual interaction among various types of vortical motions, including vortex reconnection, is a key ingredient in clarifying and controlling fluid motions (Kida and Takaoka, 1994). There is much experimental evidence that tube-like vortex regions evolve and interact at high Reynolds number in 3D turbulent flow. One can imagine that most of the physical space is filled with irrotational or very weakly rotational fluid and that the flow is driven by "small" diameter vortex tubes. However, when tube segments approach walls or other tube segments, then short term rapidly evolving processes may be expected (Zabusky and Melander, 1989). The breaking and rejoining of vortex lines may be a fundamental process in the evolution of three-dimensional vortices and the mechanics of turbulence (Saffman, 1990).

There are some common mechanisms in the reconnection process of two vortex tubes with nearly the same intensity. A typical sequence of physical events is as follows. First, the tubes tend to approach each other in antiparallel fashion advected by the mutual- and self-induction velocity. As the two vortex tubes get closer, the shape of the vortex core is deformed typically in to so-called head-tail structure. Then viscous cancellation of opposite signed vorticity in the interaction zone initiates vorticity reconnection. Advected by a complicated three-dimensional velocity field, the vorticity lines now experience a cross-linking, or bridging (Kida and Takaoka, 1994). (Zabusky and Melander, 1989) also found secondary structures called "hairpins" and "fingers".

In viscous fluid a test with reconnection of two vortex tubes was carried out. Test case was the same as one used in (Zabusky and Melander, 1989).

The vorticity field is assumed to be periodic with a period 2π in all the three orthogonal directions. We consider the motion of vortex rings in a cyclic cube of side 2π. The test was with straight offset tubes of Gaussian cross section perpendicular to each other. Each vortex tube had the form

$$\omega(r) = \omega_0 \exp\left(-\frac{r^2}{l^2}\right) \tag{23}$$

where r is the distance from the core centerline, ω_0 is the maximum vorticity at the core center and l is the e^{-1}-fold radius of the core. In this case $l = 3^{-1/3}$ and $\omega_0 = 20$.

In the Figure 3 one can see comparison between results obtained in (Zabusky and Melander, 1989) and current work. The evolution of the vortex rings is conveniently represented by iso-surfaces of vorticity norm $|\omega|$. They are plotted in the Figure 3 at several representative stages of evolution. The level of the iso-surface plotted is $|\omega| = 12$.

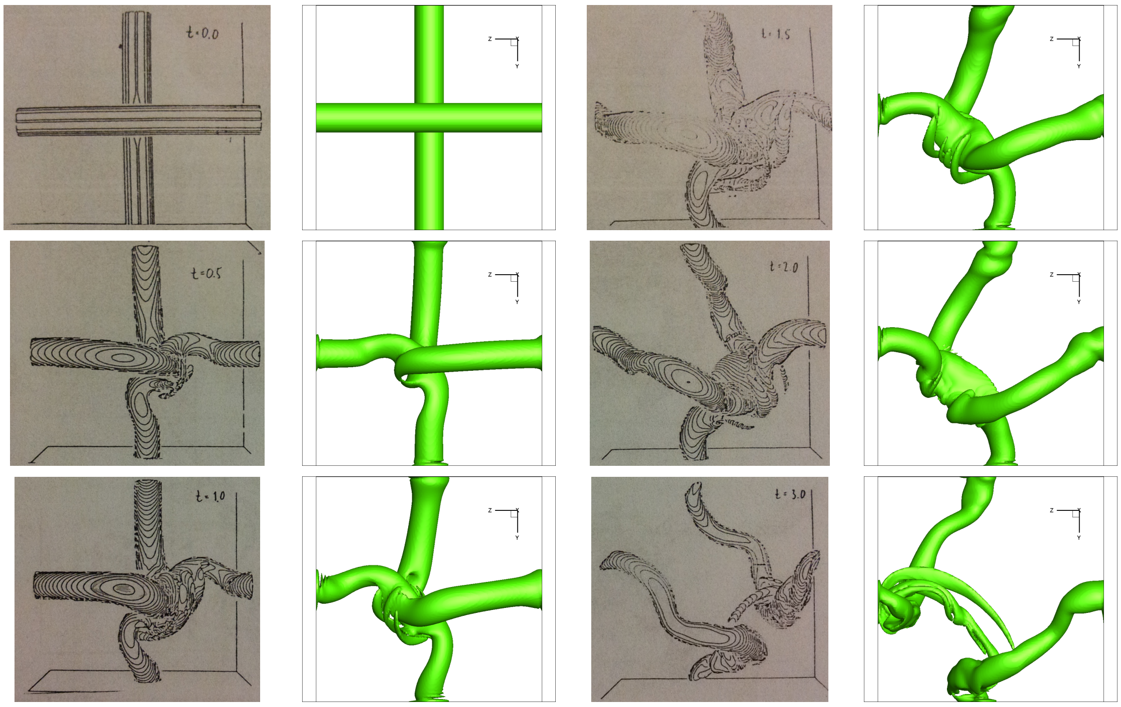

Fig. 3. Comparison of vortex tubes reconnection between (Zabusky and Melander, 1989) - left and current work - right

COLLISION OF TWO VORTEX RINGS

Second test was the vortex ring reconnection. The interaction of two identical circular viscous vortex rings starting in a side-by-side configuration was investigated. The initial data was the same as the one in (Kida et al., 1991).

The vorticity field is assumed to be periodic with a period 2π in all the three orthogonal directions. We consider the motion of vortex rings in a cyclic cube of side 2π.

Initially, two identical circular vortex rings are set up as shown in the top left corner of Figure 4. The centers of vortex rings are on y-axis.The distance between the centers of the two vortex rings is 3.65. The radius of the circular vortex rings is 0.982 The vortex rings are parallel to the xy-plane. We used a Gaussian vorticity distribution in the core:

$$\omega(r) = \omega_0 \exp\left[-\left(\frac{r}{a}\right)^2\right] \tag{24}$$

where r is the distance from the core centerline, ω_0 is the maximum vorticity at the core center and a is the e^{-1}-fold radius of the core. In this case $a = 0.393$ and $\omega_0 = 23.8$.

The evolution of the vortex rings is conveniently represented by iso-surfaces of vorticity norm $|\omega|$. They are plotted in the Figure 4 at several representative stages of evolution. The level of the iso-surface plotted is $|\omega| = 1.7$.

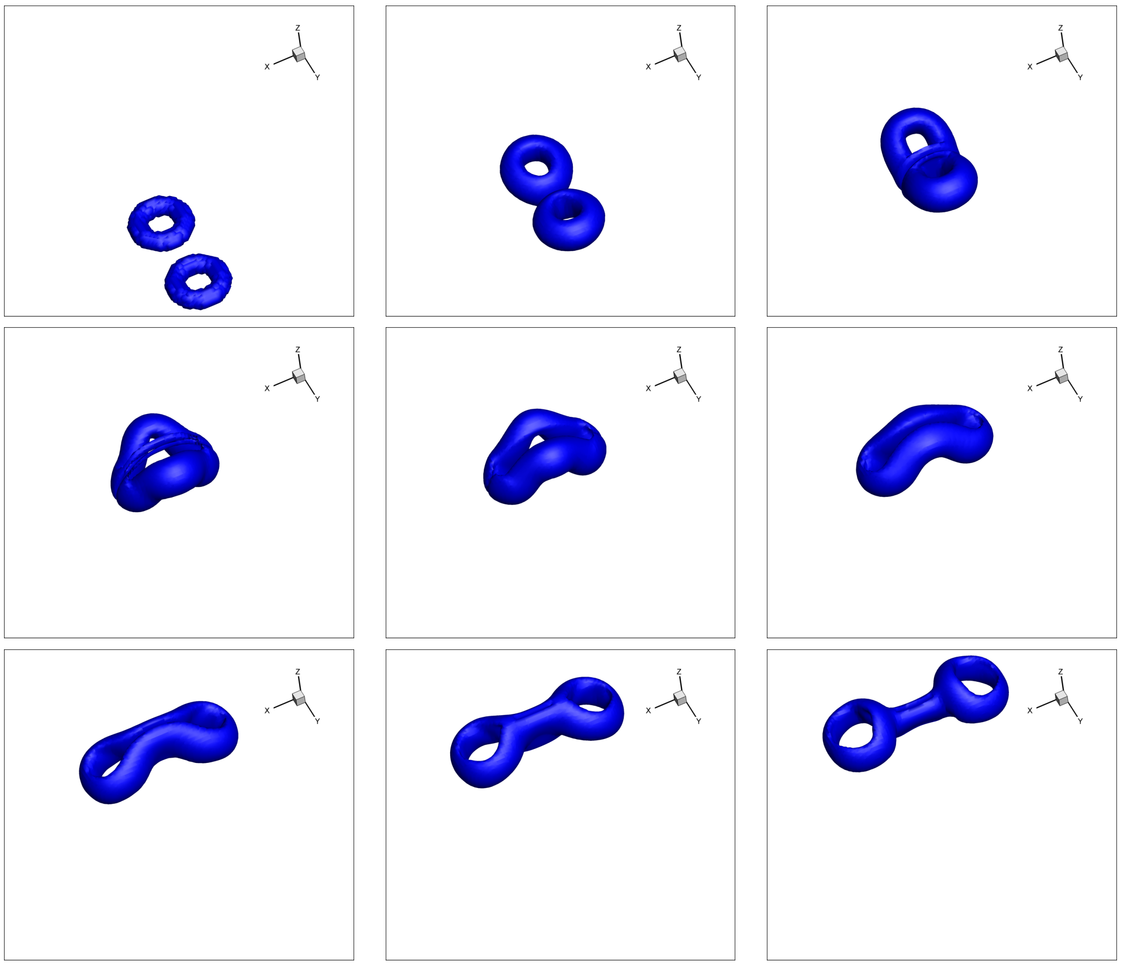

Fig. 4. *Collision of two vortex rings*

The results can be seen in the Figure 4. The vortices undergo two successive reconnections, fusion and fission, as in (Kida et al., 1991).

CONCLUSIONS

Nowadays it is not difficult to notice that the computational power of a single processor has stopped rising. Parallel architectures need to be used to deliver the means to speed up computation. Developing programs on GPU's is an interesting alternative to using the CPU. Thanks to hundreds of streaming processors working in parallel we can get the results faster. The processors also quite cheap and easily accessible.

An important element of parallel computations is choosing the right computational method to allow for the effective use of computer architecture. Moving a sequential program to this hardware may not be the best solution. As we have shown in this article, graphics cards are capable of conducting scientific computations.

Shown test cases proves that Vortex Particle Method is capable of reconstructing reconnection phenomena of vortex structures.

It is obvious that if one wants to have a good resolution of the physical phenomena, one has to use a fine numerical mesh in computations. That requires greater memory and computational time. To overcome this problems, one can use many graphics cards. Introducing of multi GPU computation is our aim in the nearest future. Properly used GPU's (memory management, parallel algorithms, etc.) allows programs to be executed much faster at relatively low cost.

ACKNOWLEDGEMENTS

Fellowship co-financed by European Union within European Social Fund.

REFERENCES

Cottet, G. H. and Koumoutsakos, P. D. (2000). *Vortex Methods: Theory and Practice*. Cambridge University Press.

Cottet, G. H., Michaux, B., Ossia, S., and VanderLinden, G. (2002). A comparison of spectral and vortex methods in three-dimensional incompressible flows. *Journal of Computational Physics*, 175:1–11.

Kida, S. and Takaoka, M. (1994). Vortex reconnection. *Annual Review of Fluid Mechanics*, 26:169–189.

Kida, S., Takaoka, M., and Hussain, F. (1991). Collision of two vortex rings. *Journal of Fluid Mechanics*, 230:583–646.

Kosior, A. and Kudela, H. (2012). Parallel computations on gpu in 3d using the vortex particle method. *Computers & Fluids*.

Kudela, H. and Kozlowski, T. (2009). Vortex in cell method for exterior problems. *Journal of Theoretical and Applied Mechanics*, 47, 4:779–796.

Kudela, H. and Malecha, Z. M. (2008). Viscous flow modelling using the vortex particles method. *Task Quarterly*, 13 No 1-2:15–32.

Kudela, H. and Malecha, Z. M. (2009). Eruption of a boundary layer induced by a 2d vortex patch. *Fluid Dynamics Research*, 41.

Kudela, H. and Regucki, P. (2009). 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*, 7:49–54.

Saffman, P. G. (1990). A model of vortex reconnection. *Journal of Fluid Mechanics*, 212:395–402.

Schwarztrauber, P. N. (1984). Fast poisson solvers. *Studies in Numerical Analysis*, 24:319–370.

Wu, J. Z., Ma, H. Y., and Zhou, M. D. (2006). *Vorticity and Vortex Dynamics*. Springer.

Zabusky, N. J. and Melander, M. V. (1989). Three-dimensional vortex tube reconnection: Morphology for orthogonally – offset tubes. *Physica D*, 37:555–562.