ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

TECHNIQUES TO INTEGRATE PATIENT-SPECIFIC SIMULATION OF ANEURYSMAL BLOOD FLOW INTO THE CLINICAL WORKFLOW

Gábor Závodszky[1], Roland Joó-Kovács[1], György Paál[1] and István Szikora[2]

[1]Budapest University of Technology and Economics
Műegyetem rkp. 3., Budapest, 1111
e-mail: {gzavodszky,rjoo-kovacs, gypaal}@hds.bme.hu

[2] National Institute of Clinical Neurosciences
Amerikai str. 57, Budapest, 1145
e-mail: szikoraistvan@me.com

Keywords: CFD, GPU, Blood flow, Aneurysm, Clinical workflow

Abstract. *The CFD simulation of the emergent flow field inside vessel malformations is a generally employed technique during research processes. The results of an accurate numeric calculation might be used for risk assessment or for predictive purposes, such as the analysis of the possible pathogenesis. Even though several generally accepted numerical methods exist which are applied frequently during research, the current mainstream methods carry specific properties that significantly limit their application in everyday medical practice. We propose a well-selected set of methodologies to provide a complete solution that is able overcome several of these shortcomings by making the preparation of the numerical mesh more automated while reducing the average computational time to a clinically relevant time-scale on a single desktop workstation. A study involving 10 patients with intracranial aneurysm was carried out using the presented fast simulation method. The sole human interaction required for the process, that is, the selection of the boundary conditions might add a few minutes at most to the total investigation time required for one patient. The possibility of modelling the effects of a virtually implemented stent without losing the performance benefits is also discussed.*

1 INTRODUCTION

Intracranial aneurysms are focal dilatations of branches of the Circle of Willis, most frequently appearing around bifurcation points. The pathogenesis, growth, and rupture of this malformation is still not fully understood. The analysis of the flow parameters (e.g., wall shear stress, pressure, oscillatory shear index) may play a critical role in understanding these processes. The application of computational fluid dynamics (CFD) in vascular research is often considered to be the best method to non-invasively extract information about the flow field. Amongst others, the research of cerebral aneurysms relies heavily on such techniques [1, 2, 3]. A further improvement would be to incorporate these methods not only in research activities but also during clinical practice. However, there are two major difficulties preventing an effective solution. The first one is the setup of the patient-specific numerical model, including the proper description of the lumen surface geometry, the creation of the numerical mesh, and finally the selection and initialisation of the appropriate boundary conditions. Although the steps to perform such operations accurately are well-established [4, 5], they certainly demand the interaction of a specialist at several points. The second problem that follows is the significant amount of required computational time to carry out an appropriately detailed transient simulation. A possible solution to the latter one is the application of super-computers, however, that can in return further augment the first problem. While these are quite feasible in research processes, they usually do not fit the requirements of an everyday clinical protocol. In the current work a feasible solution is presented based on the combination of a carefully selected set of existing technologies.

2 METHODS

In this study the triangulated lumen surface geometry is considered as first input. Most of the recent clinical workstations are capable of exporting vessel sections in this format. Note that this segmentation step usually requires human interaction and presents the scientifically most questionable part of the pipeline since the quality of the resulting surface depends strongly on subjective decisions. The automation of this step is the subject of several current researches. The vessel geometries for our study were obtained from DSA-Angiography records.

2.1 The CFD method

Since both performance and automation are high priorities, the lattice Boltzmann method was chosen as a CFD numerical scheme, which is a highly parallel numerical scheme known to be able to simulate aneurysmal flows accurately [6]. Its regular numerical grid allows for an automatic setup of the simulation domain without the need for a specialist to verify the quality of the generated numerical mesh. The LBM method also carries a few properties that makes it very appealing in today's computational environments: it is inherently parallel due to being explicit in time and local in space. It was shown to be capable of simulating large-scale cardiovascular systems [7] while exhibiting linear scaling behaviour over several GPUs [8]. To take full advantage of the parallelism, the applied solver was based on an open source LBM code [9] implemented for graphics processing units (GPUs) which excel at high performance parallel code execution. This solver relies heavily on meta-programming through the usage of a templating language to generate the GPU specific code part at run-time. This approach helps to keep the implementation of the physical description of the dynamics closer to the mathematical definition and hide some of the complications of the actual hardware specific implementation. An indirect addressing for lattice cells was used, which increased the computational time mainly

due to the overhead added by pointer dereferencing, however, it was necessary to reduce the memory requirements of storing the LBM distribution functions. The MRT relaxation model [10] was chosen to represent the collision dynamics which is known to be able to produce accurate results in the case of highly curved geometries [6]. The performance and the scaling of the code was first tested using a single computational node equipped with four Tesla C2070 cards, using double floating point precision, two Intel Xeon 5650 CPUs and 48GB or RAM. Figure 1 shows the performance results in 10^6 lattice updates per second (MLUPS). The visible drop of performance using three cards is the result of the simple domain decomposition algorithm which caused some imbalance in the computational distribution.

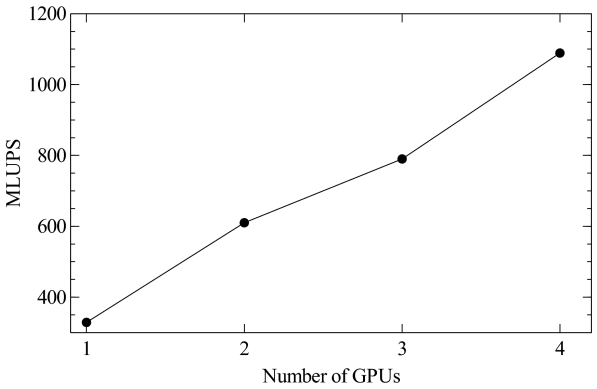

Figure 1: The strong scaling of the GPU implementation over four Tesla cards in a single computing node.

2.2 Model preparation

The surface description is turned into a regular grid by means of voxelisation. In this work the algorithms of the open source CVMLCPP library [1] have been used for this purpose. During the process one can prescribe the required resolution of the numerical grid. A higher resolution naturally leads to higher accuracy although care should be taken not to choose orders of magnitude smaller spatial steps than the original resolution of the angiographic record. In our cases the spatial steps were between 3 - 6×10^{-5}m. The solid wall and fluid cell array can be efficiently generated from the voxelised geometry using circular shifts and basic Boolean algebra.

The Newtonian rheology model was used since it was shown [11] that in most cases it does not lead to significant differences compared to the non-Newtonian rheology described by the Casson model. However, the utilisation of non-Newtonian rheology introduces only a small computational overhead, and as such, the findings of this works remain valid in this case as well.

2.3 Boundary conditions

A rigid wall model was imposed during the simulations. From one hand, the main motivation of this decision was the nature of the available medical records as they represent a time-averaged information on the vessel lumen. On the other hand, a simulation with flexible walls would significantly increase the computational costs. Currently, this options seems to be out of bounds

[1] http://tech.unige.ch/cvmlcpp/ - version 2012-04-22

for a single-workstation computational environment when a clinically relevant time-scale is necessary (that is, the time-scale of minutes rather than hours). For representing the walls in the CFD solver, the half-way bounce back algorithm was used but it is possible to change it to Guo's off-lattice boundaries [12] for higher accuracy.

The prescribed shape of the velocity profile on the inlet can also have significant influence on the resulting flow field. In the current work the same artificial volumetric inflow function was used as in [6]. The Reynolds number was set to the value of 450 at the inlet in all the cases. To recover the correct inlet shape in vessel geometries it is common to create an elongated artificial lead-in vessel section in which the accurate velocity profile can develop. This, however, leads to increased computational costs. To circumvent this, the inlet velocity vector profiles can be calculated as follows: the direction of the inlet velocity vector is chosen so that it is parallel with the beginning section of the centreline of the vessel; the length of the vector is calculated using the sum of the Womersley components. The inlet function is usually given as a periodic volume flow function representing the flow generated by the cardiac system, therefore it can be expanded using the Fourier series:

$$Q(t) = \sum_{n=0}^{N} q_n e^{in\omega t}. \tag{1}$$

Combining these, the Womersley velocity profile of every Fourier mode can be calculated and their linear sum produces the answer of the fluid system for the inlet excitation [13]:

$$u(r,t) = \frac{2q_0}{\pi R^2}\left[1-(\frac{r}{R})^2\right] + \sum_{n=1}^{N} \frac{q_n}{\pi R^2}\left[\frac{1-\frac{J_0(\beta_n \frac{r}{R})}{J_0(\beta_n)}}{1-\frac{2J_1(\beta_n)}{\beta_n J_0(\beta_n)}}\right] e^{in\omega t}, \tag{2}$$

where

$$\beta_n = i^{\frac{3}{2}}\alpha_n = i^{\frac{3}{2}} R\sqrt{\frac{n\omega}{\nu}}. \tag{3}$$

In this expression u is the length of the axial velocity vector at a given r radius, R is the radius of the pipe, J_0 and J_1 denotes the zeroth and first order Bessel functions, ν is the kinematic viscosity, ω is the angular frequency, α is the Womersley number, and finally n denotes the ordinal number of the harmonic component from the Fourier expansion.

On the outlets constant pressure is imposed on a surface perpendicular to the outlet centerline section. This might be improved by imposing a resistance model [14] based on the idea that the capillary level can be regarded as a common equipressure point and the vessel network leading to it can be represented as hydrodynamic resistance.

To model the effect of an implanted stent is generally a computationally intensive task. A numerically effective solution is to represent it from a macroscopic viewpoint as a porous layer representing some resistance on the flow [15]. The Darcy equation describes such porous materials. Since the intra-aneurysmal flow can be regarded as a high velocity flow, the equation should be extended with the Forcheimer term [16] to account for the nonlinear behaviour:

$$\bigtriangledown p = -\frac{\mu}{\kappa}\vec{v} - \frac{\rho}{\kappa_1}|\vec{v}|\vec{v}, \tag{4}$$

where κ denotes the permeability, κ_1 is the inertial permeability and μ is the dynamic viscosity. Naturally, to acquire a proper value for κ and κ_1 for a real implanted stent is not an easy task. The final resistance is dependent on many factors, such as the structure of the stent, the implantation technique, and the local curvature. The authors have an ongoing research in this direction.

3 RESULTS

The workflow being investigated consists of several steps. Firstly, an automatic voxelisation of the input lumen surface geometry is carried out. The run-time of this step represents a few seconds time at most, thus it can be regarded as negligible. The following stage is to select the appropriate inlet and outlet boundary conditions. This can be done in any 3D editor supporting voxel geometries (such as 3DSlicer[2]) by labelling the openings. This is the only step requiring human interaction and based on the number of openings might take up to a few minutes. Finally, the CFD simulation of the model takes place. Generally, this is the most time-consuming part, therefore, the run-times of several cases were evaluated. Depending on the use-case, either a stationary velocity snapshot or the full transient cardiac cycle might be required. Both cases were tested for the expected run-times. Since the LBM method is inherently transient, the stationary simulations were actual transient computations executed until the flow inside the whole domain reached its equilibrium. Every computation starts with a "warm-up" phase during which the fluid is slowly accelerated from the initial zero velocity to the stationary velocity that corresponds to the beginning of the inlet volume flow function. This ensures a good initial approximation of the velocity profiles within the domain without having initial transients. In our case, the run-times for both stationary and transient situations were evaluated on a high-end portable notebook (Intel i7-4980HQ, 16GB RAM, nVidia Geforce 980M) and on a desktop workstation (Intel i7-3770K, 32GB RAM, nVidia Geforce Titan Black) for ten intracranial aneurysmal geometries corresponding to different patients. Two different numerical grid resolutions were generated for every geometry. For the lower resolution one the target number of numerical mesh elements were 1 million while for the high resolution it was 9 million (this is the number of useful cells, on average accounting for around 20% of the total number of cells residing in the encompassing prismatic domain). To access the GPUs the CUDA 7.5 API was used. Table 1 shows the average execution times required by the CFD computations.

	Notebook [mm:ss]	Workstation [mm:ss]	
		Single	Double
low resolution (1M)	1:17 ($\sigma = 0.27$)	1:01 ($\sigma = 0.25$)	4:58 ($\sigma = 0.31$)
high resolution (9M)	10:05 ($\sigma = 0.61$)	8:13 ($\sigma = 0.63$)	23:05 ($\sigma = 0.75$)

Table 1: Run-time averages of the transient computations of one full cardiac cycle for the ten different geometries with the sample standard deviations in minutes. The hardware of the notebook allowed for single precision only while the computations on the workstation were run using both single (32 bit) and double (64 bit) floating point representation precisions.

The stationary computations typically required about 60 % of the transient run-times, since the "warm-up" phase is necessary here as well. Finally, the user interface of the solver is implemented in Python language[3]. For visualisation purposes, it embeds an instance of a scripted MayaVi [17] application. During the computation of a transient simulation, the already calculated time steps are moved to the host memory from the GPU. This makes them immediately available for vector or streamline presentation, thus, it is not necessary to wait for the full computation to begin evaluating the flow field properties. Figure 2 shows a snapshot of the working application.

[2]https://www.slicer.org/

[3]https://www.python.org/

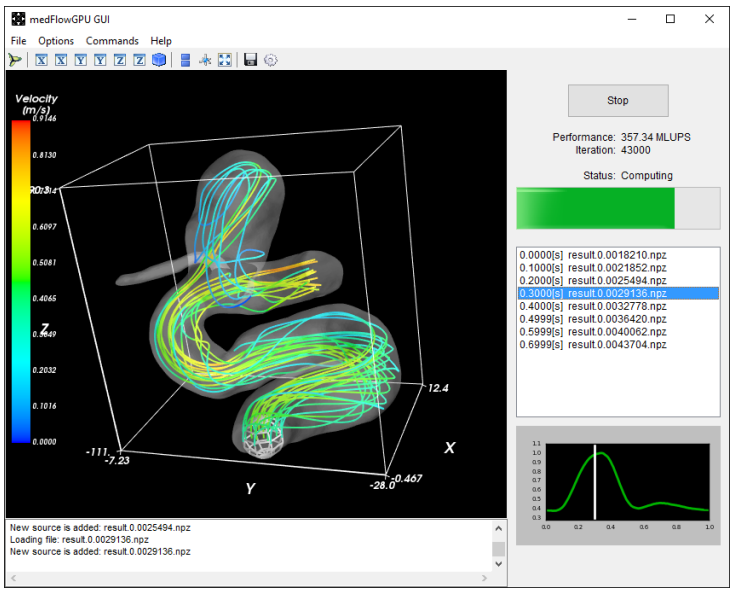

Figure 2: The software in work on a notebook with the already computed time segments available for evaluation.

4 CONCLUSIONS

The resulting run-time values of around 10 minutes for the high-resolution models using single precision are well within the clinically relevant time-scale. We should point out here that even though the presented set of techniques are evaluated in the case of aneurysmal blood flows, they do not build on any information specific to these cases, henceforth, they are applicable to a wider range of fluid flows (e.g., carotid and coronary arterial flows and other vascular flows where the same modelling approximations are valid). Thus, it can be concluded that with the adequate set of methods the fluid dynamics computation of a real aneurysm can be carried out as part of a regular clinical examination workflow even on a portable notebook. The only phase depending on human interaction is confined to the selection of the appropriate boundary conditions for the given case. This step, however, can be carried out using general clinical 3D image processing tools and does not build on the knowledge of any specialised software.

5 ACKNOWLEDGEMENT

The project was supported by the KTIA_NAP_13-1-2013-0001 Hungarian Brain Research Program.

REFERENCES

[1] J. R. Cebral, M. A. Castro, J. E. Burgess, R. S. Pergolizzi, M. J. Sheridan, and C. M. Putman, "Characterization of cerebral aneurysms for assessing risk of rupture by using patient-specific computational hemodynamics models," *American Journal of Neuroradiology*, vol. 26, no. 10, pp. 2550–2559, 2005.

[2] G. Paál, A. Ugron, I. Szikora, and I. Bojtár, "Flow in simplified and real models of intracranial aneurysms," *International Journal of Heat and Fluid Flow*, vol. 28, no. 4, pp. 653–664, 2007.

[3] I. Szikora, G. Paal, A. Ugron, F. Nasztanovics, M. Marosfoi, Z. Berentei, Z. Kulcsar, W. Lee, I. Bojtar, and I. Nyary, "Impact of aneurysmal geometry on intraaneurysmal flow: a computerized flow simulation study," *Neuroradiology*, vol. 50, no. 5, pp. 411–421, 2008.

[4] D. M. Sforza, C. M. Putman, and J. R. Cebral, "Hemodynamics of cerebral aneurysms," *Annual review of fluid mechanics*, vol. 41, p. 91, 2009.

[5] L. Campo-Deaño, M. S. Oliveira, and F. T. Pinho, "A review of computational hemodynamics in middle cerebral aneurysms and rheological models for blood flow," *Applied Mechanics Reviews*, vol. 67, no. 3, p. 030801, 2015.

[6] G. Závodszky and G. Paál, "Validation of a lattice Boltzmann method implementation for a 3D transient fluid flow in an intracranial aneurysm geometry," *International Journal of Heat and Fluid Flow*, vol. 44, no. 0, pp. 276 – 283, 2013.

[7] S. Melchionna, M. Bernaschi, S. Succi, E. Kaxiras, F. J. Rybicki, D. Mitsouras, A. U. Coskun, and C. L. Feldman, "Hydrokinetic approach to large-scale cardiovascular blood flow," *Computer Physics Communications*, vol. 181, no. 3, pp. 462–472, 2010.

[8] M. Bisson, M. Bernaschi, S. Melchionna, S. Succi, and E. Kaxiras, "Multiscale Hemodynamics Using GPU Clusters," *Communications in Computer Physics*, vol. 11, pp. 48–64, 2012.

[9] M. Januszewski and M. Kostur, "Sailfish: a flexible multi-gpu implementation of the lattice boltzmann method," *Computer Physics Communications*, vol. 185, no. 9, pp. 2350–2368, 2014.

[10] D. D'Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo, "Multiple-relaxation-time lattice Boltzmann models in three dimensions.," *hilosophical Transactions of the Royal Society - Series A: Mathematical, Physical and Engineering Sciences*, vol. 360, no. 1792, pp. 437–451, 2002.

[11] M. A. Castro, M. C. A. Olivares, C. M. Putman, and J. R. Cebral, "Unsteady wall shear stress analysis from image-based computational fluid dynamic aneurysm models under newtonian and casson rheological models," *Medical & biological engineering & computing*, vol. 52, no. 10, pp. 827–839, 2014.

[12] Z. Guo, C. Zheng, and B. Shi, "An extrapolation method for boundary conditions in lattice boltzmann method," *Physics of Fluids (1994-present)*, vol. 14, no. 6, pp. 2007–2010, 2002.

[13] J. R. Cebral, M. A. Castro, S. Appanaboyina, C. M. Putman, D. Millan, and A. F. Frangi, "Efficient pipeline for image-based patient-specific analysis of cerebral aneurysm hemodynamics: technique and sensitivity," *Medical Imaging, IEEE Transactions on*, vol. 24, no. 4, pp. 457–467, 2005.

[14] G. Janiga, P. Berg, O. Beuing, M. Neugebauer, R. Gasteiger, B. Preim, G. Rose, M. Skalej, and D. Thévenin, "Recommendations for accurate numerical blood flow simulations of stented intracranial aneurysms," *Biomedizinische Technik/Biomedical Engineering*, vol. 58, no. 3, pp. 303–314, 2013.

[15] L. Augsburger, P. Reymond, D. Rufenacht, and N. Stergiopulos, "Intracranial stents being modeled as a porous medium: flow simulation in stented cerebral aneurysms," *Annals of biomedical engineering*, vol. 39, no. 2, pp. 850–863, 2011.

[16] D. Ruth and H. Ma, "On the derivation of the forchheimer equation by means of the averaging theorem," *Transport in Porous Media*, vol. 7, no. 3, pp. 255–264, 1992.

[17] P. Ramachandran and G. Varoquaux, "Mayavi: 3d visualization of scientific data," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 40–51, 2011.