Parallel Implementations of Numerical Simulation of the Vascular Solid Tumour Growth Model under the Action of Therapeutic Agents (Chemo- and Antyangiotherapy).

Damian Borys, Krzysztof Psiuk-Maksymowicz, Sebastian Student and Andrzej Świerniak
Institute of Automatic Control
Silesian University of Technology
Gliwice, Poland
damian.borys@polsl.pl,
krzysztof.psiuk-maksymowicz@polsl.pl,
sebastian.student@polsl.pl,
andrzej.swierniak@polsl.pl

***Abstract*—The authors' main interest was to develop vascular solid tumour growth model, implement efficient numerical methods for simulations towards finding a solution of the model and trying to optimise the influence of different types of therapies. A system of partial differential equations was introduced in order to simulate the growth of tumour and normal cells as well as the dynamics of the diffusing nutrient and anti-angiogenic or chemotherapeutic factors within the tissue. We have implemented finite difference time-domain (FDTD) method, which was formerly shown to produce numerical stable solutions. In order to make calculations in larger space, which includes a complex three-dimensional structure of capillaries, a single processing unit is not sufficient. Hence, there is the need for using high computing power in order to obtain the results at reasonable time. Furthermore, over some computing space limit, the amount of memory required to compute the solution extends the capacity of single computing machine, making computer cluster is the only choice. We are comparing the implementation of the numerical method for multi-computer system (cluster) using the message passing programming (MPI) paradigm with massively parallel computing implementation using graphic computing accelerators. The code was written in C++ and compared with Matlab implementation with appropriate toolboxes (Parallel Computing Toolbox and Distributed Computing Server). In all cases, the use of parallel implementation speedups the simulation time in comparison to the standard implementation on a single processor computer. Our results showed that we can reduce the simulation time significantly, when we use parallel computing written in C++. The speedup depends on the size of the computation domain, available memory size, the type of processors used and realization accuracy. Parallelisation of the code allows to perform optimisation of therapeutic protocols included in the model.**

Keywords - *tumour growth model*; *parallel computations*; *message passing interface*; *CUDA*

I. Introduction

Solid tumour progression is inseparably connected with vascular network, surrounding its volume [1], [2]. Tumour needs to grow oxygen and nutrition factors, which will be delivered through the vascular network. That is why, considering the network as well, as its dynamics is crucial in more realistic models. In literature, we can find many aspects of solid tumour models, based on cellular automation [3], structured models [4], single cell-based models [5] and models based on physical mass and momentum equations [6]. It is possible to distinguish different phases of the tumour growth. There are many models which focus on one particular phase, for example on hiperplastic growth phase [7], tumour growth *in situ* [8], invasion [9], angiogenesis [10], or process of metastasis [11]. The microvascular network plays crucial role in development of the solid tumours. It constitutes a source of the nutrient for the tumour and enables its continuous growth. However, due to fast metabolism of the tumour cells hypoxic regions may occur causing creation of tumour necrosis sites. The phenomenon of hypoxia is important because it may lead to the process of angiogenesis and additionally is a reason of lower efficiency of different therapies. The model taking into consideration processes mentioned above was developed and its numerical solution has been performed. Independently of the type of mathematical model, calculation of its solution is always time and resources demanding (computations time or computer memory) [12]. Presented here, the model of vascular tumour growth is described by set of partial differential equations. We have implemented FDTD method which was already shown to produce numerical stable solutions. In order to make calculations in larger space, which include complex three-dimensional structure of capillaries, single processor computers are not sufficient. Hence, there is need to use more computing power to obtain the results in a reasonable time. We are comparing the implementation of the numerical method for multi-computer system (cluster) with the message passing programming paradigm [13] with massively parallel computing implementation using graphic computing accelerators (Nvidia CUDA) [14].

The structure of this article is divided into this introduction section, next the description of materials and methods used for simulations, mathematical model section and sections for results presentation and, at the end, for discussion of presented results.

II. MATERIALS AND METHODS

In order to find a solution for the mathematical model, appropriate numerical methods have had to be used. Among the explicit numerical methods, one-step Lax-Wendroff method [15] was chosen for transport equations, and the standard forward time centered space was chosen for the diffusion equations. Numerical simulations have been done on the basis of synthetic micro environment created to reflect real environment in the tissue. Except for the normal cells fraction, tumour cells fraction and ECM (see the model description in the next section), syntetic vascular network has been included in the environment. It creates the distribution pathways for nutritients, oxygen and therapeutic agents. Parameters of the model has been based on a literature.

Computations have been performed in Mathworks Matlab for testing purposes (finding optimal and stable numerical method, non-parallel implementation) and in C++ language for parallel version using MPI (Message Passing Interface) libraries and C language for CUDA with thurst, CUBLAS and STL libraries. Each implementation details are presented in Fig. 1. For CUDA entire computational domain is processed by graphic accelerator processing units. For MPI, domain is divided for some subdomains and sent to workers (slaves) to compute new subdomain. After that results are sent to master node and new domain (for next time step) is merged. The code is based on dynamic task allocation, so the number of workers (S) is lower than the number of subdomains. This technique keeps the balance of workers load. For both implementations parallelisation is done only in one time step. Next time step is dependent from the previous one, that is reason for which it has to be calculated sequentially.

Calculations were carried out using the computer cluster Ziemowit [16] funded by the Silesian BIO-FARMA project No. POIG.02.01.00-00-166/08 in the Computational Biology and Bioinformatics Laboratory of the Biotechnology Centre in the Silesian University of Technology. Every node used for MPI calculations has 2 six-cores Intel Xeon CPUs and 36GB RAM. Computer for CUDA computing was equipped with Nvidia Tesla C2075 graphic accelerator and Intel Xeon processor.

III. MATHEMATICAL MODEL

A set of partial differential equations was introduced in order to simulate growth of tumour and normal cells as well as the dynamics of the nutrient, anti-angiogenic and chemotherapeutic particles diffusing within the tissue. For modelling the tumour growth, different approaches are used. Unlike in [17], [18] we do not distinguish proliferative, quiescent and apoptotic cells. Cell behaviour is determined by the oxygen concentration in the tissue. The equations for the cell dynamics originate from the multiphase theory [19], [20], [21]. The main constituents of the multiphase part of the model are normal cells, tumour cells and extracellular matrix (ECM), thus variable n denotes volume fraction of normal cells, a denotes volume fraction of tumour cells, and m denotes volume fraction of the ECM. For simplicity volume fraction of ECM is assumed to be homogeneous and constant. The models in which the dynamics of the ECM is investigated can be found in works by Chaplain et al. [22] or by Psiuk-Maksymowicz [21]. The overall volume fraction occupied by the cells spread on the ECM must satisfy the inequality $\psi = n + a + m \leq 1$. In order to close the model, the porous media assumption is applied [23]. In order to provide physiological picture, the heterogeneity of concentration of the nutrient and xenobiotics is ensured. Mathematical model consists of five partial differential equations (PDE) :

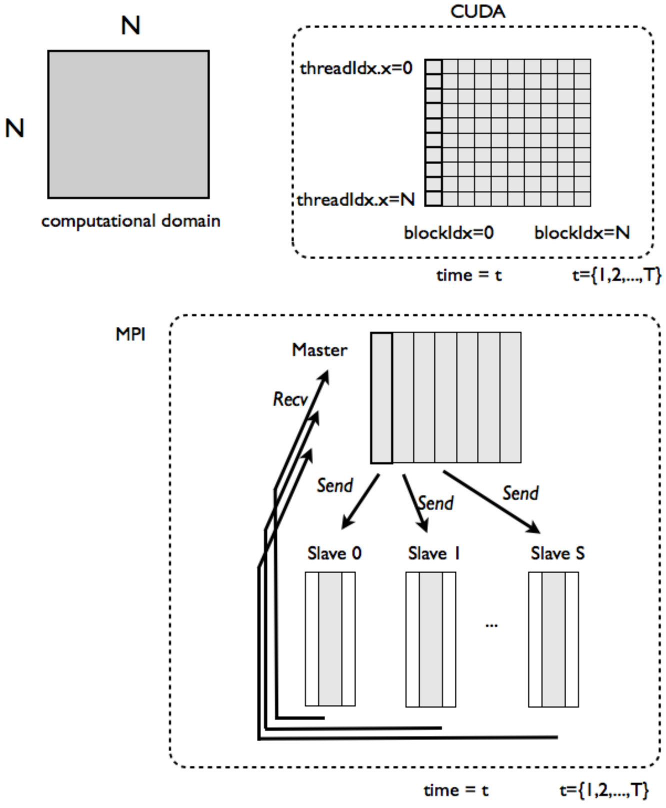

Figure 1. Diagrams of CUDA and MPI implementations for 2D computational domain. For CUDA whole computational domain is processed by graphic accelerator processing units. For MPI domain is divided for some subdomains and is sent to workers (slaves) to compute new domain.

$$
\begin{cases}
\dfrac{\partial n}{\partial t} = \nabla \cdot (n\, K\, \Sigma'\, \nabla\psi) + nF(c - c_P)[\alpha_n(1-\psi) - k_n d_{ch}] - \gamma_n n F(c_A - c), \\
\dfrac{\partial a}{\partial t} = \nabla \cdot (a\, K\, \Sigma'\, \nabla\psi) + aF(c - c_P)[\alpha_a(1-\psi) - k_a d_{ch}] - \gamma_a a F(c_A - c), \\
\dfrac{\partial c}{\partial t} = D_c \nabla^2 c - (k_{n_P} n + k_{a_P} a)F(c - c_P) - (k_{n_Q} n + k_{a_Q} a)F(c_P - c)F(c - c_A) + S_1(e), \\
\dfrac{\partial d_a}{\partial t} = D_{da} \nabla^2 d_a + S_2(e) - k_{da} d_a e - \lambda_{da} d_a, \\
\dfrac{\partial d_{ch}}{\partial t} = D_{dch} \nabla^2 d_{ch} + S_3(e) - k_{dch}(n+a)F(c - c_P) - \lambda_{dch} d_{ch}.
\end{cases}
$$

where K is a coefficient related to the permeability of the medium, Σ is a stress function, c stands for oxygen concentration, d_a stands for the concentration of anti-angiogenic treatment agent, d_{ch} stands for the concentration of chemical treatment agent, e stands for the binary function denoting occurrence of blood capillaries. Growth of the cells is of logistic type, where α_n and α_a stands for growth rate for normal and tumour cells, respectively. Normal and tumour cells undergo apoptosis with γ_n and γ_a rates, respectively. Growth and degradation of the cells is dependent on the oxygen availability, therefore in both terms sigmoid function $F(\cdot)$ is present. In growth terms it is dependent on the proliferation oxygen concentration c_P, and in degradation terms it is dependent on the apoptotic oxygen concentration c_A. In reaction-diffusion equations D_c, D_{da}, D_{dch} are present

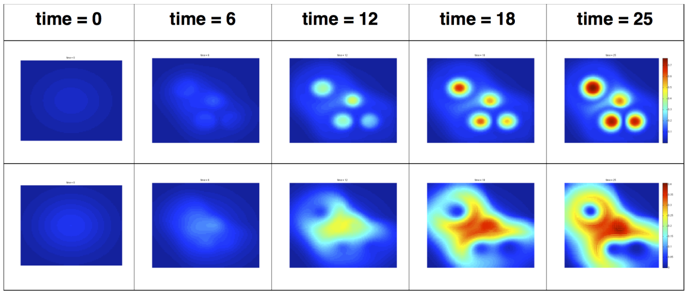

Figure 2. Spatial changes of the cellular density due to anti-angiogenic therapy (top row) and chemotherapy

denoting oxygen, anti-angiogenic agent and chemotherapeutic agent diffusion coefficients, respectively. Source terms are denoted by $S_i(\cdot)$, $i \in \{1, 2, 3\}$ functions dependent on the position of the blood vessels. Different types of stress-volume ratio relations can be taken into account. The simplest feature characterising stress is that below the value ψ_0 it vanishes, and increases for $\psi > \psi_0$ and tends to infinity as $\psi - 1$, e.g.

$$\Sigma(\psi) = E(1 - \psi_0) \left(\frac{\psi - \psi_0}{1 - \psi} \right)_+ , \qquad (1)$$

where $(f)_+$ denotes the positive part of f and E is the value of the derivative in$\psi = \psi_0$, a sort of Young's modulus for moderate compressions.

IV. RESULTS

Example of model result with the drugs acting on healthy and tumour cells are presented in pictures collected in Fig. 2. The colors correspond to density of the cells after anti-angiogenic therapy (top row) and after chemotherapy (bottom row). For the top row, the higher density of the cells corresponds also with localisation of the vessels network.

The main results of our work present a comparison of the speedup of parallel implementation with the basic Matlab computations (Fig. 3). We have compared the speedups of MPI implementations with different domain sizes (100x100 and 400x400 points). The speedup is increasing up to about 12 cores then is slightly lower when the number of processing units increases (Fig. 3). This is caused by the architecture - single computing machine has 12 physical cores and, when increasing this number, we are causing that processes needs to communicate through the computer network which is always slower than shared memory architecture (even for Infiniband QDR connection). When spatial computational domain was increased 16 times the speedup increased up to 8 and the absolute computation times ratio increased maximum to about 10 times (Fig. 4). Comparing MPI (with 11cores) and CUDA implementation (Fig. 5) we can see that the speedup is higher for smaller domains but when increased the performance is significantly lower.

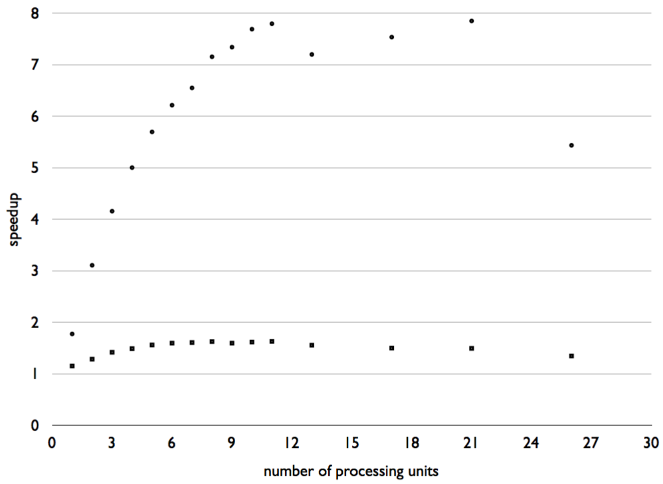

Figure 3. Speedup of the calculations time in dependence with the number of processing units. Two series are compared - with smaller spatial domain (100x100, boxes) and with large domain (400x400, circles).

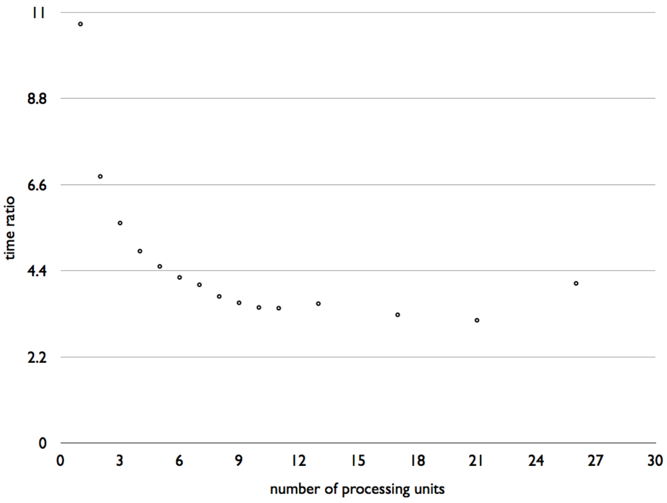

Figure 4. Time ratio after 16 times increase of the spatial domain (T400/T100) depending of the number of processing units.

V. DISCUSSION

The presented results show that when the computing problem is relatively small, parallelisation using MPI technique and the usage of big cluster architecture is not the best choice, as long as the speedup is figurative. However, using CUDA architecture we can obtain very interesting results. With the growth of the size of the problem CUDA application meets its limitations related to memory bandwidth limits and hence MPI implementation seems to be a reasonable choice. We could also observe that, if the problem can be solved using single multi-core machine it will give us slightly better performance than using more machines.

Single simulation, having relatively small data domain (as in our case 400x400), is possible to compute using single computing machine and it takes about an hour (or less) to compute. However, switching the space domain to the third dimension only MPI implementation should be considered.

Parallelisation of presented numerical simulations serves us not only to study different methods of parallelisation performance, but it is a crucial step toward trying to find optimal therapeutic protocols of implemented chemo- and antyangiotherapy. To complete any optimization algorithm it is required to perform

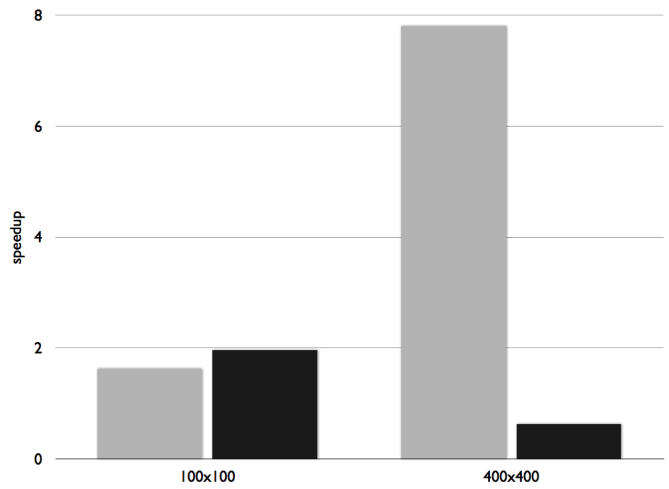

Figure 5. Speedup of MPI with 11cores (light grey colour) and CUDA (black) implementation for different spatial domain sizes (100x100 and 400x400).

thousands of repetitions of model simulation. Even for relatively small 2D domains and using a single computing machine the computation time is unacceptably long without using the parallelisation.

Further works will include implementation of meta-heuristic methods as simulated annealing, genetic algorithms, ant colony optimisation and others to find the optimal solution. However, this methods are inherently connected with multiple model simulation so even apparently small speedup of execution time, multiplicated during optimisation step, will contribute significanlty to the overall execution time.

Acknowledgment

This work was supported by the National Science Centre (NCN) in Poland under Grant No. N- N519-647840.

References

[1] R. P. Araujo and D. L. S. McElwain, A history of the study of solid tumour growth: the contribution of mathematical modelling. Bull Math Biol, 66, 2004, pp:1039-1091.

[2] H. P. Greenspan. Models for the growth of a solid tumor by diffusion. Studies in Applied Mathematics, 4, 1972, pp:317-340.

[3] S. Dormann and A. Deutsch, Modeling of self-organized avascular tumor growth with a hybrid cellular automaton. In Silico Biology, 2, 2002, pp:393-406.

[4] F. Billy, J. Clairambault, F. Delaunay, C. Feillet and N. Robert, Age-structured cell population model to study the influence of growth factors on cell cycle dynamics. Math Biosci Eng, 10, 2013, pp:1-17.

[5] K. A. Rejniak, A single-cell approach in modeling the dynamics of tumor microregions. Math Biosci Eng, 2, 2005, pp:643-55.

[6] J. S. Lowengrub, et al., Nonlinear modelling of cancer: bridging the gap between cells and tumours. Nonlinearity, 23, 2010, pp:R1-R9.

[7] E. Mamontov, A. Koptioug and K. Psiuk-Maksymowicz, The minimal, phase-transition model for the cell-number maintenance by the hyperplasia-extended homeorhesis. Acta Biotheor, 54, 2006, pp:61-101.

[8] A.3rd Bankhead, N. S. Magnuson and R. B. Heckendorn, Cellular automaton simulation examining progenitor hierarchy structure effects on mammary ductal carcinoma in situ. J Theor Biol, 246, 2007, pp:491-498.

[9] Y. Kam, K. A. Rejniak, and A. R. Anderson, Cellular modeling of cancer invasion: integration of in silico and in vitro approaches. J Cell Physiol, 227, 2012, pp:431-438.

[10] M. A. Chaplain and A. R. Anderson, Mathematical modelling, simulation and prediction of tumour-induced angiogenesis. Invasion Metastasis, 16, 1996, pp:222-234.

[11] J. A. Sherratt, Predictive mathematical modeling in metastasis. Methods Mol Med, 57, 2001, pp:309-315.

[12] G. Evans, J. Blackledge, and P. Yardley, Numerical Methods for Partial Differential Equations. Springer London, 1999.

[13] The Message Passing Interface (MPI) standard, http://www.mcs.anl.gov/research/projects/mpi/, [retrieved 03.2014]

[14] Nvidia CUDA Toolkit, https://developer.nvidia.com/cuda-toolkit, [retrieved 03.2014]

[15] S. Larsson and V. Thomee, Finite Difference Methods for Hyperbolic Equations. Springer Berlin Heidelberg, 2003.

[16] Ziemowit cluster web page, http://www.ziemowit.hpc.polsl.pl, [retrieved 03.2014]

[17] F. Billy, et al., A pharmacologically based multiscale mathematical model of angiogenesis and its use in investigating the efficacy of a new cancer treatment strategy. J Theor Biol, 260, 2009, pp:545-62.

[18] P. Macklin and J. Lowengrub, Nonlinear simulation of the effect of microenvironment on tumor growth. J Theor Biol, 245, 2007, pp:677-704.

[19] H. M. Byrne, J. R. King, D.L.S. McElwain, and L. Preziosi, A two-phase model of solid tumour growth. Appl Math Lett, 16, 2003, pp:567-573.

[20] H. Byrne and L. Preziosi, Modelling solid tumour growth using the theory of mixtures. Math Med Biol, 20, 2003, pp:341-366.

[21] K. Psiuk-Maksymowicz, Multiphase modelling of desmoplastic tumour growth. J Theor Biol, 329, 2013, pp:52-63.

[22] M.A.J. Chaplain, L. Graziano, and L. Preziosi, Mathematical modelling of the loss of tissue compression responsiveness and its role in solid tumour development. Math Med Biol, 23, 2006, pp:197-229.

[23] D. Ambrosi and L. Preziosi, On the closure of mass balance models for tumor growth. Mathematical Models and Methods in Applied Sciences, 12, 2002, pp:737-754.