



**HAL**  
open science

# On the efficient numerical simulation of heterogenous anisotropic diffusion models of tumor invasion using GPUs

Donato Pera, Carlos Málaga, Ramón G. Plaza, Chiara Simeoni

► **To cite this version:**

Donato Pera, Carlos Málaga, Ramón G. Plaza, Chiara Simeoni. On the efficient numerical simulation of heterogenous anisotropic diffusion models of tumor invasion using GPUs. PUMPS+AI2018 – Summer School on Programming and Tuning Massively Parallel Systems+Artificial Intelligence, Jul 2018, Barcelona, Spain. . hal-02525342

**HAL Id: hal-02525342**

**<https://hal.science/hal-02525342>**

Submitted on 30 Mar 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On the efficient numerical simulation of heterogenous anisotropic diffusion models of tumor invasion using GPUs

Donato Pera\*(donato.pera@univaq.it), Carlos Málaga\*\*(camalaga@yahoo.com.uk), Ramón G. Plaza @(plaza@mym.iimas.unam.mx), Chiara Simeoni @@(chiara.simeoni@unice.fr)

\* Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica Università degli Studi dell'Aquila (Italy)

\*\* Departamento de Física, Facultad de Ciencias, Universidad Nacional Autónoma de México, Circuito Exterior s/n, Ciudad Universitaria, Cd. Mx. C.P. 04510 (Mexico)  
@ Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Circuito Escolar s/n, Ciudad de México C.P. 04510 (Mexico)  
@@ Laboratoire de Mathématiques J. A. Dieudonné UMR CNRS 7351, Université de Nice Sophia-Antipolis, Parc Valrose 06108 Nice Cedex 02 (France)

## Introduction

Mathematical models for tumor invasion are often used to predict the behaviour of cancer evolution and can produce strikingly nontrivial patterns. Therefore, their numerical solution often requires high spatial resolution to capture detailed biophysical phenomena. As a consequence, long computational times are required when using a serial implementation of numerical schemes. Parallel computing can improve dramatically the time efficiency of some numerical methods such as finite differences algorithms, which are relatively simple to implement and apply to tumor invasion models. For clinical operators and applied scientists involved in setting up realistic experiments, the possibility of running fast comparative simulations using simple algorithms implemented into affordable processors is of primary interest, and that is where Graphical Processing Units (GPUs) excel.

In this work we focus on a mathematical model of anisotropic and heterogeneous diffusion of tumor cells, a set of time-evolution parabolic equations. The model includes chemotaxis and haptotaxis effects, that appear in other biophysical phenomena [1]. We compute numerical solutions considering spatial discretization by centered finite differences and time integration through an explicit Euler method. The choice of time-explicit algorithms is due to their greater ease of implementation, and performance, on GPU devices, despite the limitations related to their reduced stability properties.

The codes used in this work are designed using CUDA. The CUDA platform (Compute Unified Device Architecture NVIDIA 2007), was designed to support GPU execution of programs and focuses on data parallelism. With CUDA, graphics cards can be programmed with a medium-level language, that can be seen as an extension to C/C++/Fortran, without requiring a great deal of hardware expertise. We refer to [6] and [5] for a comprehensive introduction to GPU-based parallel computing.

## Mathematical Model

In our cancer dynamic model we considered :

- Tumour cells density (n)
- Matrix Degradative Enzymes (MDE) concentration (m)
- Extracellular Matrix (ECM) density (f)

The ECM consists of a complex mixture of macromolecules, some of which are believed to play structural roles and others are very important for cell adhesion, spreading and motility. One of the key mechanism allowing the tumour diffusion is its capability to degraded progressively the ECM. The acronym MDE indicates a family of enzymes, produced by cancer and cells and digest the ECM, which enables the migration of cancer cells through the tissue. [\*] Roughly speaking invasive process is mostly determined by the ability of the tumour to degrade extracellular matrix. The tumour segregates matrix degradation enzymes which degrade the tissue locally.

The tumour cell density evolves according to

$$n_t = \nabla \cdot (D(x, y) \nabla n) - \gamma \nabla \cdot (n \nabla f) + \lambda n \left(1 - \frac{n}{n_0} - \frac{f}{f_0}\right), \quad (1)$$

Here  $(x, y) \in \Omega \subset R^2$ ,  $t > 0$ .

The evolution of  $f$  is described by the following simple equation:

$$f_t = -\kappa m f, \quad (2)$$

Finally the evolution of the MDE concentration is

$$m_t = D_m \nabla^2 m + \delta n \left(1 - \frac{m}{m_0}\right) - \beta m, \quad (3)$$

where the first term on the right side is a diffusion term, the second one models the production and the decay respectively. Let us to summarize as follow the system will be studied numerically in the following section

$$\begin{aligned} n_t &= \nabla \cdot (D(x, y) \nabla n) - \gamma \nabla \cdot (n \nabla f) + \lambda n \left(1 - \frac{n}{n_0} - \frac{f}{f_0}\right) \\ f_t &= -\kappa m f, \\ m_t &= D_m \nabla^2 m + \delta n \left(1 - \frac{m}{m_0}\right) - \beta m. \end{aligned} \quad (4)$$

Here, as usual,  $(x, y) \in \Omega \subset R^2$ ,  $t > 0$ . This system of equations is based on the model by Enderling *et al.* [4], with the new feature of an anisotropic diffusion tensor of the form

$$D(x, y) = \begin{pmatrix} a(x, y) & b(x, y) \\ b(x, y) & c(x, y) \end{pmatrix}. \quad (5)$$

Compared to [2], [1] and [4], we introduce a spatially varying and anisotropic diffusion tensor  $D$ . For that purpose we set

$$\tilde{D}(x, y) = \frac{\tau}{L^2} D(x, y),$$

to arrive at the non-dimensional system

$$\begin{aligned} n_t &= \nabla \cdot (\tilde{D}(x, y) \nabla n) - \gamma \nabla \cdot (n \nabla f) + \lambda n (1 - n - f) \\ f_t &= -\kappa m f, \\ m_t &= d_m \nabla^2 m + \delta n (1 - m) - \beta m. \end{aligned} \quad (6)$$

The free parameters are  $d_m, \delta, \beta, \kappa$  and  $\lambda > 0$  (see Table 1). We can write this numerical scheme as [3]:

$$\begin{aligned} N_{i,j}^{k+1} &= N_{i,j}^k + a_{i,j} \frac{\Delta t}{4\Delta x^2} (N_{i+1,j}^k - 2N_{i,j}^k + N_{i-1,j}^k) + c_{i,j} \frac{\Delta t}{4\Delta y^2} (N_{i,j+1}^k - 2N_{i,j}^k + N_{i,j-1}^k) + \\ &+ b_{i,j} \frac{\Delta t}{2\Delta x \Delta y} (N_{i+1,j+1}^k - N_{i+1,j-1}^k - N_{i-1,j+1}^k + N_{i-1,j-1}^k) + \\ &+ \frac{\Delta t}{4\Delta x^2} (a_{i+1,j} - a_{i-1,j}) (N_{i+1,j}^k - N_{i-1,j}^k) + \\ &+ \frac{\Delta t}{4\Delta x^2} (F_{i+1,j}^k - F_{i-1,j}^k) (N_{i+1,j}^k - N_{i-1,j}^k) + \frac{\Delta t}{4\Delta y^2} (F_{i,j+1}^k - F_{i,j-1}^k) (N_{i,j+1}^k - N_{i,j-1}^k) + \\ &+ \frac{\Delta t}{4\Delta x \Delta y} ((b_{i+1,j} - b_{i-1,j}) (N_{i+1,j}^k - N_{i-1,j}^k) + (b_{i,j+1} - b_{i,j-1}) (N_{i,j+1}^k - N_{i,j-1}^k)) + \\ &+ \frac{\Delta t}{4\Delta y^2} (c_{i,j+1} - c_{i,j-1}) (N_{i,j+1}^k - N_{i,j-1}^k) + \\ &+ \frac{\Delta t}{4\Delta x^2} \gamma N_{i,j}^k (F_{i+1,j}^k - 2F_{i,j}^k + F_{i-1,j}^k) + \\ &+ \frac{\Delta t}{4\Delta y^2} \gamma N_{i,j}^k (F_{i,j+1}^k - 2F_{i,j}^k + F_{i,j-1}^k) + \lambda N_{i,j}^k (1 - F_{i,j}^k - N_{i,j}^k). \end{aligned} \quad (7)$$

$$F_{i,j}^{k+1} = F_{i,j}^k (1 - \Delta t \kappa M_{i,j}^k). \quad (8)$$

$$\begin{aligned} M_{i,j}^{k+1} &= M_{i,j}^k + \delta N_{i,j}^k (1 - M_{i,j}^k) - \beta M_{i,j}^k + \\ &+ d_m \frac{\Delta t}{\Delta x^2} (M_{i+1,j}^k - 2M_{i,j}^k + M_{i-1,j}^k) + d_m \frac{\Delta t}{\Delta y^2} (M_{i,j+1}^k - 2M_{i,j}^k + M_{i,j-1}^k). \end{aligned} \quad (9)$$

The numerical scheme (7) - (9) is stable provided that

$$\Delta t < \min \left( \frac{1 \max(\Delta x^2, \Delta y^2)}{8 \max_{i,j} (a_{ij}, c_{ij})}, \frac{1 \max(\Delta x^2, \Delta y^2)}{2 d_m} \right).$$

For consistency with the previous work by Enderling *et al.* [4] we consider the parameter values used in their simulations, which were based on tumor invasion in breast tissue. The values can be found in Table 1.

Description	Symbol	Value
Diffusion coefficient of MDE	$d_m$	0.0005
Rate of production of MDE	$\delta$	0.1
Degradation rate of ECM	$\kappa$	10.0
Proliferation rate of tumor cells	$\lambda$	0.3
Haptotactic sensitivity	$\gamma$	0, 0.0005
Decay rate of MDE	$\beta$	0

## Performance analysis comparisons and conclusions

We run our tests on 6 square domains, from  $64 \times 64$  to  $2048 \times 2048$  elements over 10000 time-steps. We define boundary conditions equal to 0 and

$$n_0(x, y) = N_0 \exp(-\omega_0((x - 0.5)^2 + (y - 0.5)^2)) \quad (10)$$

, with  $N_0 = 0.75$  and  $\omega_0 = 0.005$ ,  $m_0(x, y) = 0.5n_0(x, y)$  and  $f_0(x, y) = 1 - 0.5n_0(x, y)$ .

$$D(x, y) = d_n \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix}$$

where  $a = 0.1, 0.01$ . All numerical simulations were performed using these GPUs NVIDIA GTX 670/1344 CUDA cores/4Gbyte of RAM installed on HP DL585G7 4 AMD Opteron 6128 8 cores clock frequency 2.0 GHz 64 Gbytes of RAM, OS Linux centOS 5.5 amd64, compiler GNU gcc 4.4 and NVIDIA CUDA 6.5 linux 64 bit toolbox and NVIDIA GTX 1080/2560 CUDA cores/4Gbyte of RAM installed on HP DL585G7 4 AMD Opteron 6128 8 cores clock frequency 2.0 GHz 64 Gbytes of RAM, OS Linux centOS 6.5 amd64, compiler GNU gcc 4.4.7 and NVIDIA CUDA 8.0 linux 64 bit toolbox

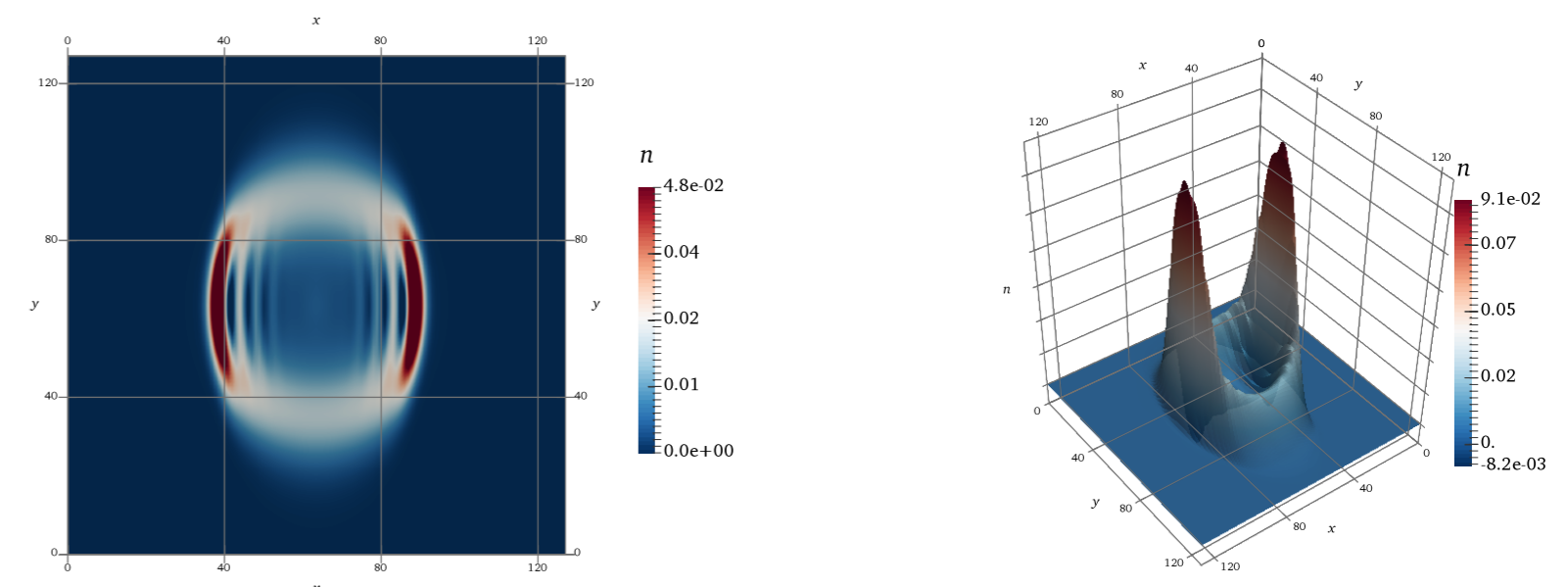


Figure 1: Numerical solution of tumor density cells  $n$  after 1 second for simulation anisotropic case. With haptotaxis in ( $\gamma = 0.005$ ) contour(left), 3D results (right)

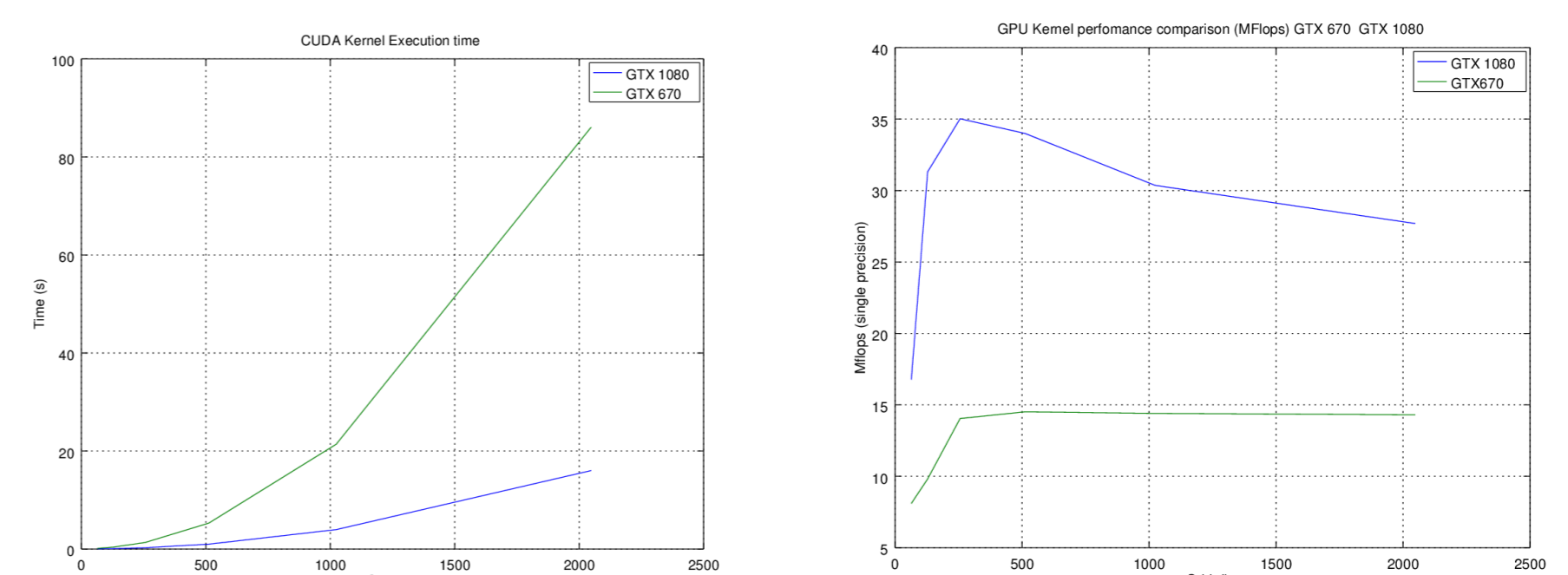


Figure 2: GPU CUDA Kernel performance comparison Mflops between GTX 670 and GTX 1080

In the above graphs we report the CUDA kernel performance (MFlops single precision) and the related execution time (in seconds) for the two GPUs used.

## Main References

- [1] A. R. A. ANDERSON, M. A. J. CHAPLAIN, E. L. NEWMAN, R. J. C. STEELE, AND A. M. THOMPSON, *Mathematical modelling of tumour invasion and metastasis*, J. Theor. Med. **2** (2000), no. 2, pp. 129–154.
- [2] A. R. A. ANDERSON, *A hybrid mathematical model of solid tumour invasion: the importance of cell adhesion*, Math. Med. Biol. **22** (2005), no. 2, pp. 163–186.
- [3] L. DASCAL AND N. A. SOCHEN, *A maximum principle for Beltrami color flow*, SIAM J. Appl. Math. **65** (2005), no. 5, pp. 1615–1632.
- [4] H. ENDERLING, A. R. A. ANDERSON, M. A. J. CHAPLAIN, A. J. MUNRO, AND J. S. VAIDYA, *Mathematical modelling of radiotherapy strategies for early breast cancer*, J. Theoret. Biol. **241** (2006), no. 1, pp. 158–171.
- [5] Jason Sanders Edward Kandrot, *CUDA by example An Introduction to General Purpose GPU Programming*, Addison-Wesley, (2010)
- [6] D. B. KIRK AND W.-M. W. HWU, *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st ed., 2010.