On the efficient numerical simulation of heterogeneous anisotropic diffusion models for tumor invasion using GPUs

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

ON THE EFFICIENT NUMERICAL SIMULATION OF HETEROGENEOUS ANISOTROPIC DIFFUSION MODELS FOR TUMOR INVASION USING GPUS

DONATO PERA*, CARLOS MÁLAGA, CHIARA SIMEONI, AND RAMÓN G. PLAZA

Abstract. The aim of this article is to show how continuous mathematical models for tumor dynamics can be solved efficiently using commodity Graphics Processing Units (GPUs) found in personal and portable computers. The test set of equations models haptotaxis and heterogeneous anisotropic diffusion of the cancer cells population. The numerical solution is obtained by using a second order finite difference Euler scheme. It is proven that, as the space resolution improves, the GPU implementation of the numerical scheme shows an increasingly better performance than that of the Central Processing Units (CPUs).

1. Introduction

Mathematical models for tumor invasion are often used to predict the behavior of cancer evolution and can produce strikingly nontrivial patterns. Therefore, their numerical solution demands high spatial resolution to capture the detailed biophysical phenomena. As a consequence, long computational times are often required when using a serial implementation of the numerical schemes. Parallel computation can improve drastically the time efficiency of some numerical methods such as finite differences algorithms, which are relatively simple to implement and are easily applicable to continuous tumor invasion models. For clinical practitioners 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 Graphics Processing Units (GPUs) excel.

Parallel computing based on modern GPUs has the advantage of high performance at a relatively low energy and monetary costs. In 2002, commodity graphics cards started to outperform Central Processing Units (CPUs). As GPUs grew faster and cheaper, the interest to harvest their power for applications others than graphical display originated, around 2006, what is known as General Purpose GPU computation (GPGPU) – `https://hgpu.org`. By the year 2009, GPUs that

2010 *Mathematics Subject Classification.* 35Q92, 65M06, 65Y05, 65Y20, 92C17, 92C50.

Key words and phrases. tumor invasion models, haptotaxis, anisotropic diffusion, finite difference schemes, Graphics Processing Units, computational efficiency.

*Author to whom correspondence should be addressed.

could be bought out the shelf had a theoretical peak performance of more than a thousand single precision GFLOPs (10^9 floating point operations per second), almost ten times more than their multi-core CPU counterpart. Nowadays, GPUs found in personal computers and laptops can perform double precision computations with a ratio of speed over cost larger than any other parallel computing architecture. Additionally, GPUs are also energetically efficient making them an affordable and portable option for parallel computation.

The codes used to study the performance of GPUs presented in this article were programmed using CUDA (Compute Unified Device Architecture). The CUDA platform, introduced by NVIDIA© in 2007, was designed to support GPU execution of programs and focuses on data parallelism [13]. With CUDA, graphics cards can be programmed with a medium-level language, that can be seen as an extension to C/C++, without requiring a great deal of hardware expertise. We refer to Kirk and Hwu [11], as well as Sanders and Kandrot [16] for a comprehensive introduction to GPU-based parallel computing, including details about the CUDA programming model and the architecture of current generation NVIDIA© GPUs.

In order to exemplify overall GPU performance in the context of efficient simulation of continuous tumor invasion models, in this article we consider a system of partial differential equations that models anisotropic and heterogeneous diffusion of tumor cells. The equations are based on the mathematical system proposed by Enderling *et al.* [6] to account for solid breast tumor growth (see also [2, 3]). The system under consideration is a set of time-evolution parabolic equations of reaction-diffusion type, which includes haptotaxis of tumor cells or directed movement up the gradient of fixed chemicals, as well as the effect of degradation enzymes. We compute their numerical solutions considering spatial discretization by centered finite differences and time integration through an explicit Euler method. The choice of time-explicit algorithms is motivated by their greater ease of parallel implementation and performance on GPU devices, despite the limitations related to their reduced stability properties.

Plan of the article. In Section 2 we present the mathematical model of breast tumor growth of Enderling *et al.* [6] with the incorporation of an anisotropic and heterogeneous diffusion tensor. The numerical scheme and a set of numerical experiments are illustrated in Section 3. Section 4 contains the comparative performance evaluation between GPU and CPU implementations of the numerical scheme. Finally, in Section 5 we propose a brief discussion of our results.

2. Modeling haptotaxis and anisotropic solid tumor growth

The mathematical model considered in this article mainly accounts for interactions between the cancer cells population and the surrounding tissue, and neglects the interaction between the tumor and the vascular system. It is based on three dynamical variables: the tumor cells density represented by n, the Matrix Degradation Enzymes (MDEs) represented by m, and the fibronectin molecules density

bound to the Extracellular Matrix (ECM) represented by f. The ECM is the tissue surrounding the tumor and typically degraded by the cancer cells. The ECM is a complex mixture of macromolecules, some of which are believed to play structural roles while others are involved in cells adhesion, spreading and motility. The MDEs are produced by the cancer cells and are able to digest the ECM, so that enabling the migration of cancer cells through the tissue. In summary, the invasive process is determined by the ability of the tumor to degrade the surrounding ECM by production of the MDEs. Another preeminent mechanism of tumor invasion is known as *haptotaxis*, which is the movement towards gradients of chemicals which are fixed, that is, they do not diffuse (in contrast to chemotaxis where there is diffusion of the chemo-attractant). The present model assumes that the tumor cells move haptotactically towards the gradients of fibronectin.

Finally, the other contribution to the flux of cancer cells is due to *diffusion*. In general, the motility flux takes the form $-\mathbb{D}\,\nabla n$, where $\mathbb{D}$ is a (positive definite) diffusion tensor which, in general, may depend on f and m, and vary through space. Nevertheless, our study focuses on the effects of anisotropic and heterogeneous diffusion on tumor invasion. This is reflected in the fact that tumor cells exhibit altered random motility depending on their localization and on preferred ECM directions. The motivation for considering such a situation is precisely to understand the qualitative behavior of tumor movement when migration between different types of tissue occurs. We are not interested in how random motility increases or decreases when the concentrations of ECM and MDEs change from region to region, but on the intrinsic diffusion properties imposed by the geometry and/or the structure of the environment. The heterogeneous rate of invasion of certain tumors on healthy tissue is a widely reported phenomenon (see [14] and the references therein). The anisotropy and heterogeneity of glioma invasion along brain structures is a good example of the latter [7]. Aligned migration of tumor cells is thus determined by the properties of the tissue itself (among other factors). Anisotropic diffusion tensors for tumor cells density have been proposed to account for complex tissue structures, which predict the rate and the directional movement of malign cells as they infiltrate an healthy tissue. One of the first models of this kind is that of Swanson *et al.* [19].

We consider the following model of tumor invasion with non-constant diffusion,

$$\begin{cases} n_t = \nabla\cdot\big(\mathbb{D}\,\nabla n\big) - \Gamma\nabla\cdot\big(n\,\nabla f\big) + \Lambda\, n\Big(1 - \dfrac{n}{n_0} - \dfrac{f}{f_0}\Big) \\ f_t = -K m f \\ m_t = D_m\Delta m + \Theta\, n\Big(1 - \dfrac{m}{m_0}\Big) - B\, m \end{cases} . \tag{2.1}$$

These equations are defined on a two-dimensional domain $\Omega \subset \mathbb{R}^2$, and $t > 0$. The typical domain Ω is a square with sides of length L corresponding, for example, to a *Petri dish* or any support where in vitro experiments are usually performed

(see Figure 1). Besides, zero flux boundary conditions are imposed, namely

$$n \cdot \hat{\nu} = 0\,, \quad f \cdot \hat{\nu} = 0\,, \quad m \cdot \hat{\nu} = 0\,, \qquad \text{on } \partial\Omega\,, \tag{2.2}$$

where $\hat{\nu}$ is the outer unit normal at each point of the boundary $\partial\Omega$. Since all densities n, m and f are scalar functions, the requirement (2.2) actually turns out to be Dirichlet boundary conditions, and consistent with experimental justifications.

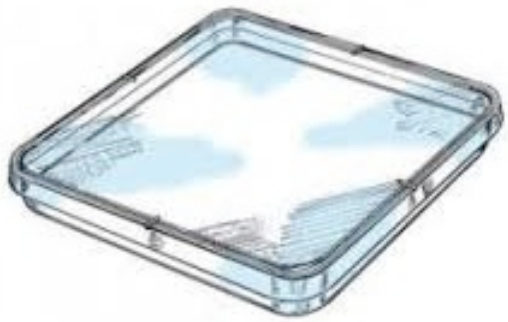


FIGURE 1. A squared Petri dish for in vitro experiments.

The difference with the model proposed by Enderling *et al.* [6] lies on the consideration of an anisotropic and heterogeneous (positive definite) diffusion tensor of the form

$$\mathbb{D}(x,y) = \begin{pmatrix} a(x,y) & b(x,y) \\ b(x,y) & c(x,y) \end{pmatrix}, \tag{2.3}$$

where a, b and c are known functions of space. The constants $\Gamma > 0$ and $\Lambda > 0$ are the tumor cells haptotaxis and proliferation coefficients, respectively. The fibronectin degradation coefficient is $K > 0$, while D_m, Θ and B are the (positive) coefficients of diffusion, production and decay of enzymes, respectively. The constants n_0, f_0 and m_0 are characteristic concentrations of tumor cells, fibronectin and enzymes, which are related to the initial experimental concentrations.
It is worthwhile remarking that a constant diffusion coefficient D_m in the MDEs equation is biologically justified because of the ECM degradation carried out by the enzymes, which can subsequently move over an unconstrained domain. Moreover, various dependencies of the parameters can easily be incorporated into the model for reproducing more realistic experiments.

For the non-dimensionalization of the system, according to Enderling *et al.* [6] we rescale distance with L, tumor cells density with n_0, ECM density with f_0, MDEs concentration with m_0 and the characteristic time with $\tau = 1$ year, to

obtain the following non-dimensional system,

$$\begin{cases} n_t = \nabla\cdot\big(\mathbb{D}\,\nabla n\big) - \gamma\nabla\cdot\big(n\,\nabla f\big) + \lambda\, n\big(1-n-f\big) \\ f_t = -\kappa\, m f \\ m_t = d_m\Delta m + \delta\, n\big(1-m\big) - \beta\, m \end{cases}, \tag{2.4}$$

further endowed with no flux boundary conditions (2.2). All variables are dimensionless, including the diffusion tensor $\mathbb{D}$ rescaled with τ/L^2. The other dimensionless parameters are $d_m = \tau D_m/L^2$, $\delta = \tau\Theta\, n_0/m_0$, $\beta = \tau B$, $\kappa = \tau K m_0$, $\gamma = \tau f_0\,\Gamma/L^2$ and $\lambda = \tau\Lambda$, and all take positive values. In the next section, we show a series of numerical solutions to (2.4) on the rescaled unit square Ω.

3. Numerical scheme and simulations

The set of equations (2.4) is solved numerically on the unit square $\Omega = (0,1)\times(0,1)$ discretized by a Cartesian grid $\big(i\Delta x, j\Delta y\big)$, $i = 0,1,...,L_1$, $j = 0,1,...,L_2$, $L_1, L_2 \in \mathbb{N}$, where Δx and Δy represent the space steps in each coordinate direction. The spatial derivatives are approximated by centered finite differences and the time derivatives by an explicit Euler method. The time variable takes discrete values $t_k = k\Delta t$, where Δt is the constant time step and $k \in \mathbb{N}$.

We denote the approximated values of the dynamical variables at the grid points and discrete times as

$$\begin{aligned} N^k_{i,j} &\approx n(i\Delta x, j\Delta y, k\Delta t)\,, \\ F^k_{i,j} &\approx f(i\Delta x, j\Delta y, k\Delta t)\,, \\ M^k_{i,j} &\approx m(i\Delta x, j\Delta y, k\Delta t)\,, \end{aligned}$$

and the numerical values of the entries of diffusion tensor (2.3) at the grid points are likewise defined as

$$\begin{aligned} a_{i,j} &= a(i\Delta x, j\Delta y)\,, \\ b_{i,j} &= b(i\Delta x, j\Delta y)\,, \\ c_{i,j} &= c(i\Delta x, j\Delta y)\,, \end{aligned}$$

for each (i,j) and k. Then, the evolution of the system is computed through the following numerical scheme,

$$\begin{aligned} N^{k+1}_{i,j} = N^k_{i,j} &+ a_{i,j}\frac{\Delta t}{\Delta x^2}\big(N^k_{i+1,j} - 2N^k_{i,j} + N^k_{i-1,j}\big) \\ &+ c_{i,j}\frac{\Delta t}{\Delta y^2}\big(N^k_{i,j+1} - 2N^k_{i,j} + N^k_{i,j-1}\big) \\ &+ b_{i,j}\frac{\Delta t}{2\Delta x\Delta y}\big(N^k_{i+1,j+1} - N^k_{i+1,j-1} - N^k_{i-1,j+1} + N^k_{i-1,j-1}\big) \\ &+ \frac{\Delta t}{4\Delta x^2}\big(a_{i+1,j} - a_{i-1,j}\big)\big(N^k_{i+1,j} - N^k_{i-1,j}\big) \\ &+ \frac{\Delta t}{4\Delta y^2}\big(c_{i,j+1} - c_{i,j-1}\big)\big(N^k_{i,j+1} - N^k_{i,j-1}\big) + ... \end{aligned} \tag{3.1}$$

$$
\begin{aligned}
&+ \frac{\Delta t}{4\Delta x \Delta y}\Big[\big(b_{i+1,j} - b_{i-1,j}\big)\big(N^k_{i,j+1} - N^k_{i,j-1}\big) \\
&\qquad\qquad + \big(b_{i,j+1} - b_{i,j-1}\big)\big(N^k_{i+1,j} - N^k_{i-1,j}\big)\Big] \\
&+ \gamma \frac{\Delta t}{4\Delta x^2}\big(N^k_{i+1,j} - N^k_{i-1,j}\big)\big(F^k_{i+1,j} - F^k_{i-1,j}\big) \\
&+ \gamma \frac{\Delta t}{4\Delta y^2}\big(N^k_{i,j+1} - N^k_{i,j-1}\big)\big(F^k_{i,j+1} - F^k_{i,j-1}\big) \\
&+ \gamma \frac{\Delta t}{4\Delta x^2} N^k_{i,j}\big(F^k_{i+1,j} - 2F^k_{i,j} + F^k_{i-1,j}\big) \\
&+ \gamma \frac{\Delta t}{4\Delta y^2} N^k_{i,j}\big(F^k_{i,j+1} - 2F^k_{i,j} + F^k_{i,j-1}\big) \\
&+ \Delta t\, \lambda\, N^k_{i,j}\big(1 - N^k_{i,j} - F^k_{i,j}\big)\,,
\end{aligned}
$$

$$
F^{k+1}_{i,j} = F^k_{i,j}\big(1 - \Delta t\, \kappa\, M^k_{i,j}\big)\,, \tag{3.2}
$$

$$
\begin{aligned}
M^{k+1}_{i,j} = M^k_{i,j} &+ d_m \frac{\Delta t}{\Delta x^2}\big(M^k_{i+1,j} - 2M^k_{i,j} + M^k_{i-1,j}\big) \\
&+ d_m \frac{\Delta t}{\Delta y^2}\big(M^k_{i,j+1} - 2M^k_{i,j} + M^k_{i,j-1}\big) \\
&+ \Delta t\, \delta\, N^k_{i,j}\big(1 - M^k_{i,j}\big) - \Delta t\, \beta M^k_{i,j}\,.
\end{aligned} \tag{3.3}
$$

The following proposition establishes an upper bound for the time step to ensure numerical stability of the scheme (3.1)-(3.3). The validity of such bound is related to the discrete maximum principle for the *Beltrami color flow*, which is typically used in image processing (refer to [5, 4]).

Proposition 3.1. *The numerical scheme given by* (3.1)-(3.3) *is stable provided that*

$$
\Delta t \,\leq\, \min\Big\{\frac{1}{8}\frac{\max(\Delta x^2, \Delta y^2)}{\max_{i,j}\big(a_{ij}, c_{ij}\big)}\,,\; \frac{1}{2}\frac{\max(\Delta x^2, \Delta y^2)}{d_m}\Big\}\,. \tag{3.4}
$$

Proof. The proof follows from Theorem 5.1 in [5]. Indeed, the central explicit difference scheme in (3.1) exhibits the general form

$$
N^{k+1}_{i,j} = N^k_{i,j} + r\Delta t\, O_{i,j}(N^k, F^k, M^k)\,,
$$

as the finite difference numerical operator of the *Beltrami color flow* considered in [5] for equations of the form

$$
N_t = \frac{1}{\sqrt{g}}\nabla\cdot\big(D\,\nabla N\big)\,,
$$

with diffusion matrix given by the general expression (2.3), and where g is the metric associated to the Laplace-Beltrami operator. In the present case of a flat Euclidean space, the metric reduces to the identity with $r = 1$ (the ratio of color

and spatial distances, see [4]). Substitution of $g^k_{i,j} \equiv 1$ and $r = 1$ into the formula (5.5) in [5] yields the first term of the right hand side of (3.4). The second term follows from the well-known stability estimate for standard reaction-diffusion equations with diffusion coefficient d_m (see [8], for instance). □

Remark 3.2. As we mentioned in the introduction, explicit numerical schemes usually require very small time steps in order to ensure their stability. The GPU architecture, however, is very well-suited to execute finite difference calculations for many data elements (grid points) simultaneously [13]. As a consequence, the inconvenience of small time steps is overcome through parallelization techniques which accomplish thousands of iterations in relatively small wall-clock running times of simulation (a couple of hours). Nevertheless, the overall benefits of parallel processing are slightly more complicated than just decreasing the running time of some programs, as we shall discuss in Section 4.

In this section, we present the results of the numerical approximation of system (2.4) obtained by applying the scheme (3.1)-(3.3). All simulations have been performed using a NVIDIA© GTX 670 graphics card with 1344 CUDA cores and 4 Gb of RAM, installed on a processor HP DL585G7 4 AMD Opteron 6128 with 8 cores, clock frequency 2.0 GHz, 64 Gb of RAM, operating system Linux centOS 5.5 amd64, compiler GNU gcc 4.4 and NVIDIA© CUDA 6.5 linux 64 bit toolbox. For comparison (see Section 4), the same numerical scheme and parameter values have been implemented serially on one single processor in the HP DL585G7 4 AMD Opteron 6128 CPU with 8 cores, but the graphical results are naturally omitted. All numerical simulations in this article have been realized on the Linux HPC cluster Caliban – `http://caliban.dm.univaq.it` – located in the Laboratory of High Performance Parallel Computing at the University of L'Aquila [15].

For consistency with the previous work by Enderling *et al.* [6], we consider the parameter values used in their simulations, which are based on tumor invasion in breast tissue. These parameter values can be found in Table 1.

Moreover, for the numerical computations we adopt a Cartesian grid of progressively increasing sizes of $128, 256, 512, 1024$ and 2048 points, for which the time steps are correspondingly defined to satisfy the stability estimate (3.4). In the sequel, we only reproduce the graphical results with a grid size of 128 points, since all simulations performed with finer grids are qualitatively the same.

3.1. **Isotropic, homogeneous diffusion.** As a first step, we consider the case of isotropic and homogeneous diffusion tensors of the form

$$\mathbb{D} = d_n \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

in order to set a numerical experiment which is tantamount to reproducing the results in [6]. The initial condition for n is taken as

$$n_0(x, y) = N_0 \exp\big\{-\omega_0((x - 0.5)^2 + (y - 0.5)^2)\big\}, \tag{3.5}$$

TABLE 1. Non-dimensional parameters used for the simulations [6]

Description	Symbol	Value
Diffusion coefficient of tumor cells	d_n	0.0001
Haptotactic sensitivity	γ	$\{0, 0.00005\}$
Proliferation rate of tumor cells	λ	0.75
Degradation rate of ECM	κ	10.0
Diffusion coefficient of MDEs	d_m	0.0005
Production rate of MDEs	δ	0.1
Decay rate of MDEs	β	0

with $N_0 = 0.75$ and $\omega_0 = 0.005$, thus simulating an initial radial tumor located at the center of the domain. The initial conditions for MDEs and ECM densities are taken, respectively, as

$$m_0(x,y) = \tfrac{1}{2} n_0(x,y)\,, \qquad f_0(x,y) = 1 - \tfrac{1}{2} n_0(x,y)\,. \tag{3.6}$$

In order to isolate the diffusion from the haptotactic effects, we first consider the case when $\gamma = 0$, and next we allow directed movement with haptotactic sensitivity set as $\gamma = 0.00005$.

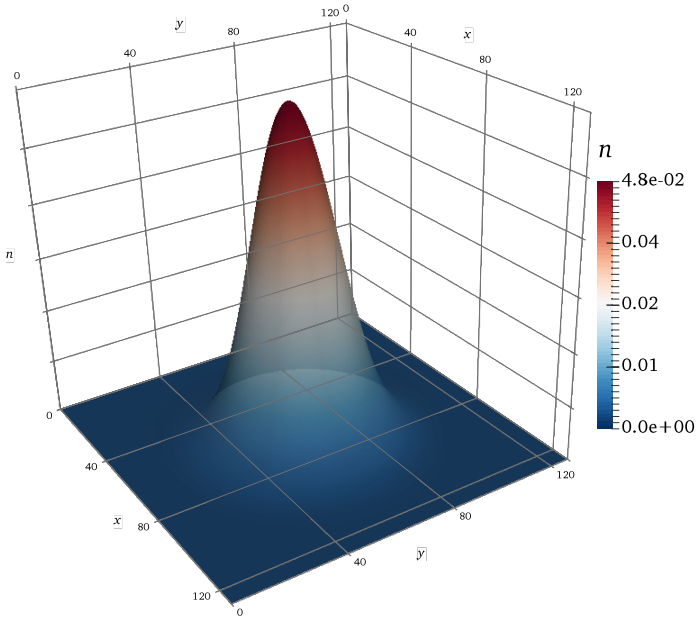
Figure 2 shows the results of the first simulation with solely isotropic and homogeneous diffusion, in the absence of haptotaxis ($\gamma = 0$). As expected from previous numerical observations in [6], the initial circular tumor diffuses symmetrically in all radial directions.

Figure 3 shows the results of the second simulation with also haptotactic effects ($\gamma = 0.00005$), again with isotropic and homogeneous diffusion. Because of the directed movement toward the fibronectin molecules, tumor cells form a depression in the center of the domain due to degradation of the former: this is sometimes quoted as the *volcano effect*, which is well-known in chemotaxis (refer to [17]).

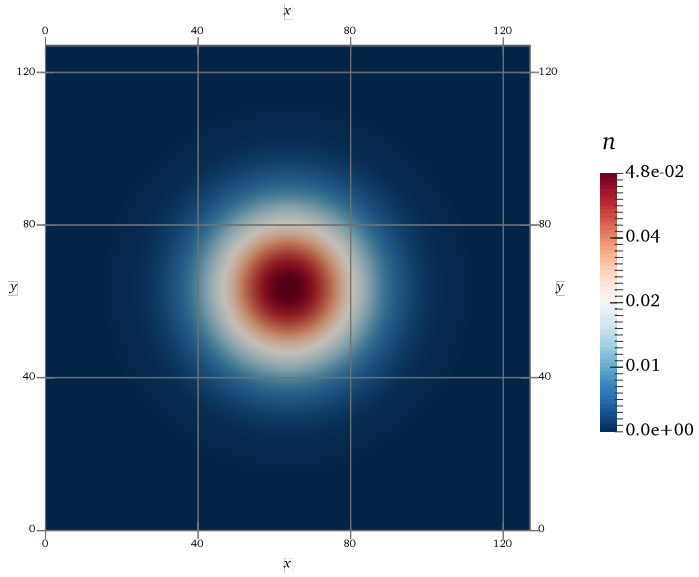
3.2. **Anisotropic, homogeneous diffusion.** Next, we focus on the effects of diffusion anisotropy on the tumor invasion. For that purpose, we consider anisotropic and homogeneous diffusion tensors of the form

$$\mathbb{D} = d_n \begin{pmatrix} a & 0 \\ 0 & 1 \end{pmatrix},$$

where $a \in \{0.1, 0.01\}$, namely we allow the diffusion on the y direction to be 10 and 100 times greater than in the x direction, with $|\mathbb{D}| = O(d_n)$ for consistency with the measurements in [6]. Once again the constant diffusion coefficient for cancer cells is $d_n = 0.0001$ and the initial conditions are given by (3.5)-(3.6). We analyze both the effects of the anisotropic diffusion alone, first by setting $\gamma = 0$,

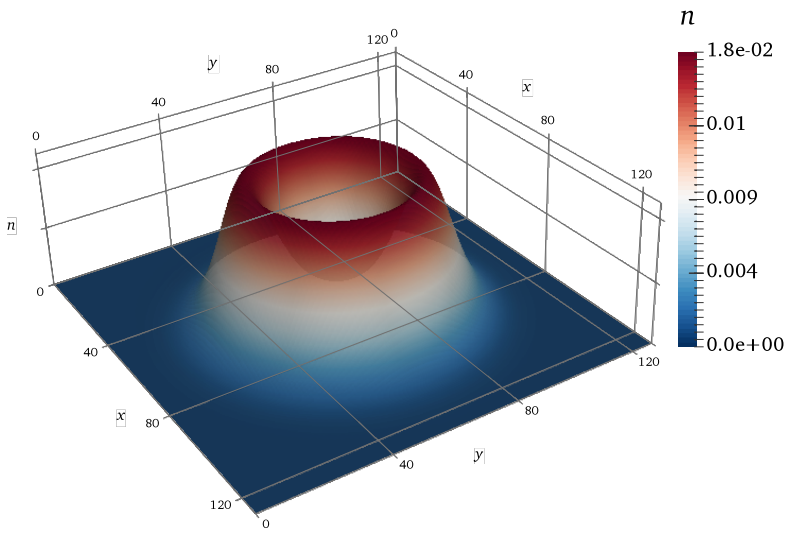


(a) tumor cells density

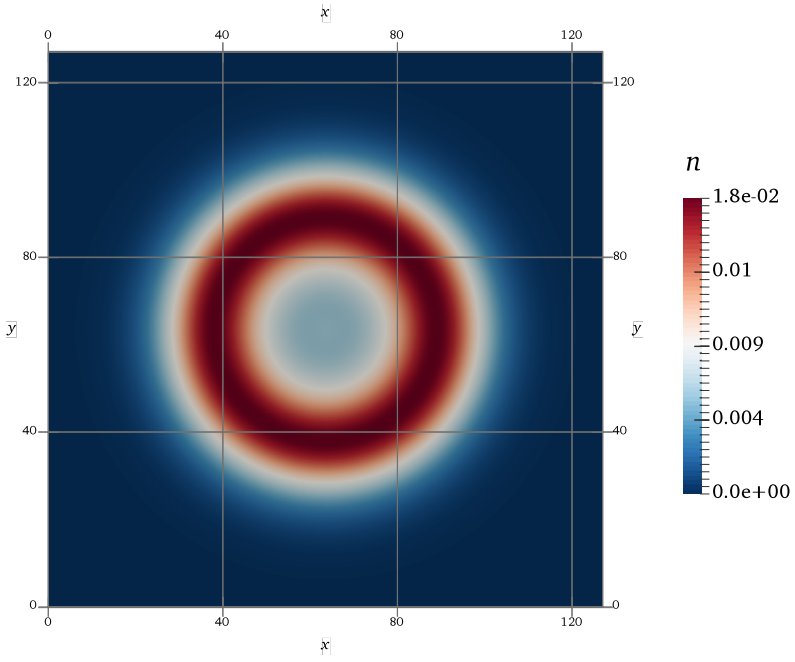


(b) contour plot

FIGURE 2. The left panel (a) shows the numerical solution of tumor cells density n evaluated at $t = 1$ unit of simulation time in the isotropic, homogeneous diffusion case ($a = c = 1.0\,,\, b = 0.0$) without haptotaxis ($\gamma = 0.0$). The right panel (b) shows the matching contour plot (color online).



(a) tumor cells density



(b) contour plot

FIGURE 3. The left panel (a) shows the numerical solution of tumor cells density n evaluated at $t = 1$ unit of simulation time in the isotropic, homogeneous diffusion case ($a = c = 1.0\,,\, b = 0.0$) with weak haptotactic signal ($\gamma = 0.00005$). The right panel (b) shows the matching contour plot (color online).

and then the combined diffusion with directed movement due to haptotaxis, by taking $\gamma = 0.00005$.

Figure 4 shows the results of the simulation in the case of anisotropic and homogeneous diffusion with $a = 0.01$ and no haptotaxis. The dynamics clearly shows enhanced diffusion in the direction y without formation of the depression in the center of the domain, in view of the absence of haptotactic effects.

Figure 5 shows the results of the simulation in the case of anisotropic and homogeneous diffusion with $a = 0.01$ and haptotactic sensitivity value $\gamma = 0.00005$, so that we notice the combined effects of enhanced diffusion in the direction y plus the *volcano effect* already mentioned above.

3.3. **Anisotropic, inhomogeneous diffusion.** Finally, we consider the case of an anisotropic and heterogeneous diffusion tensor (2.3). The heterogeneity is modeled by splitting the spatial domain Ω into two regions: more precisely, we set

$$\mathbb{D}(x,y) = d_n \begin{pmatrix} a(x,y) & 0 \\ 0 & 1 \end{pmatrix}, \tag{3.7}$$

where

$$a(x,y) = \begin{cases} 0.001\,, & x < 0.5\,, \\ 0.005\,, & x \geq 0.5\,. \end{cases}$$

These simulations are cartoon-like simplifications of the behavior of cancer dynamics related to tissues with heterogeneous biological properties in different regions of space (see [18] for a recent study in the case of brain tumors).

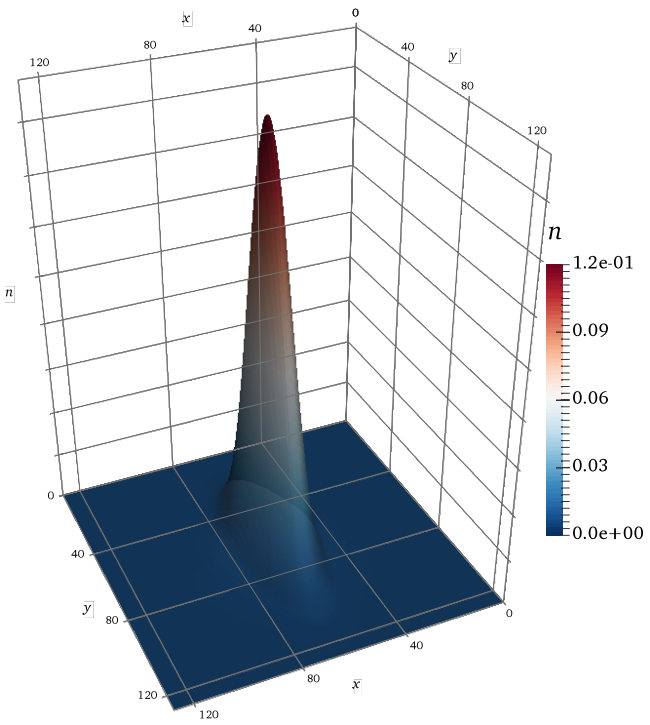
Figure 6 shows the results of the simulation in the case of anisotropic and heterogeneous diffusion tensor (3.7) without haptotactic effects ($\gamma = 0$). We clearly notice the effect of anisotropy and heterogeneity in the spatial distribution of the tumor cells density.

Figure 7 shows the results of the simulation in the case of anisotropic and heterogeneous diffusion tensor (3.7) combined with haptotactic effects ($\gamma = 0.00005$). We now notice the effect of anisotropy and heterogeneity in the spatial distribution of the tumor cells density as well as the structural formation of the *volcano effect* due to haptotaxis.

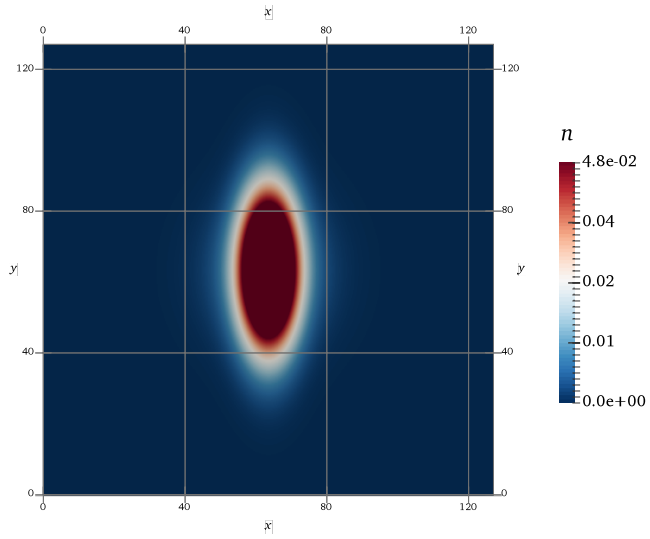
Remark 3.3. It is worthwhile commenting on the numerical oscillations which are clearly noticeable in Figure 5 and Figure 7. In both cases, the simulations concern system (2.4) with anisotropic diffusion tensor $\mathbb{D}$ and nonzero haptotactic signal (for $\gamma \neq 0$) occurring simultaneously within the dynamics of the cancer cells population. Therefore, the mathematical model under consideration exhibits increased hyperbolic-type characteristics and numerical oscillations may develop when using a second order finite difference Euler scheme (3.1)-(3.3), unless suppressed by applying suitable *slope-limiter* techniques (see [9, 10], for instance).

4. Performance evaluation

In this section, we analyze the performance of the parallel implementation of the numerical scheme introduced in Section 3. For that purpose, we recall the

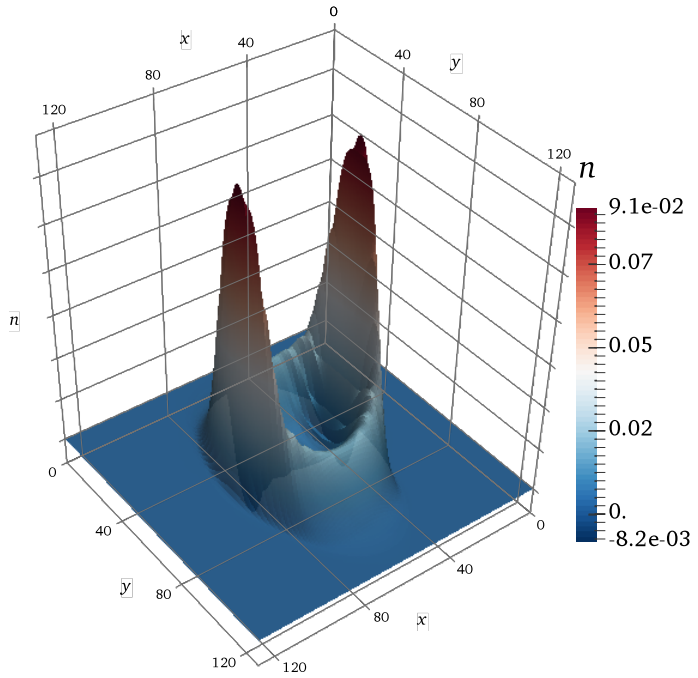


(a) tumor cells density

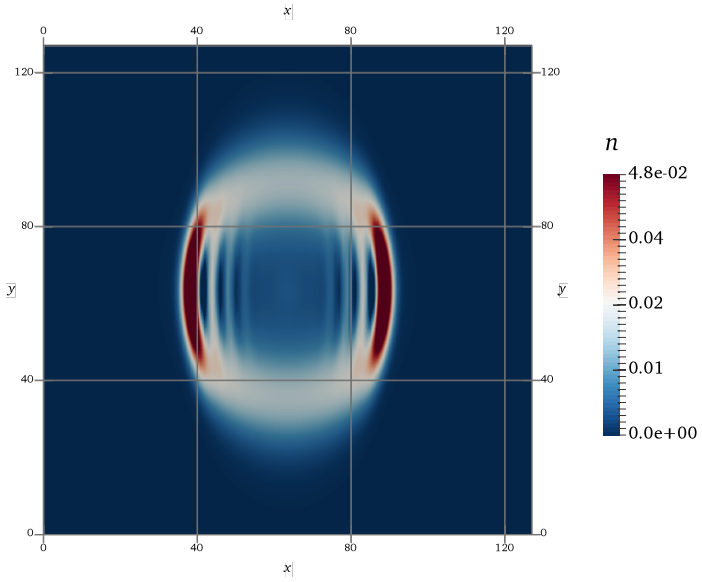


(b) contour plot

FIGURE 4. The left panel (a) shows the numerical solution of tumor cells density n evaluated at $t = 1$ unit of simulation time in the anisotropic, homogeneous diffusion case ($a = 0.01$, $c = 1.0$, $b = 0.0$) without haptotaxis ($\gamma = 0.0$). The right panel (b) shows the matching contour plot (color online).

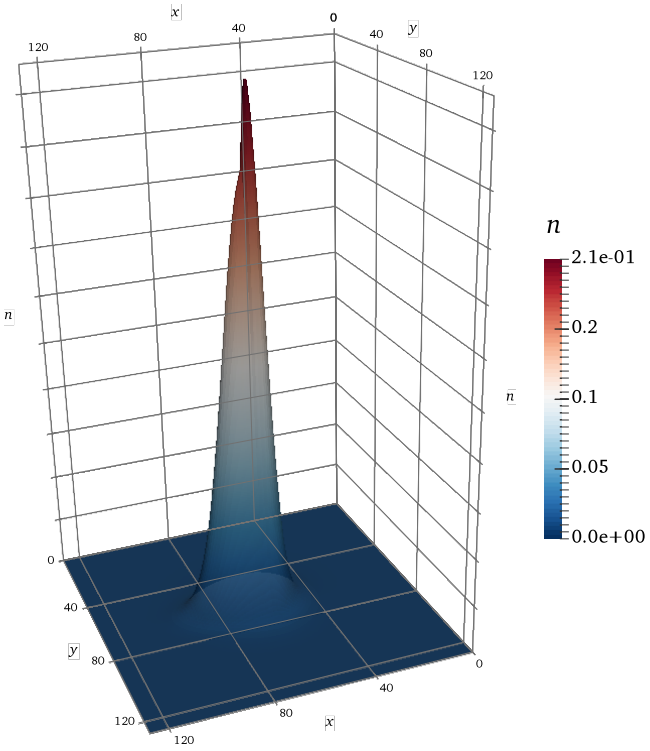


(a) tumor cells density

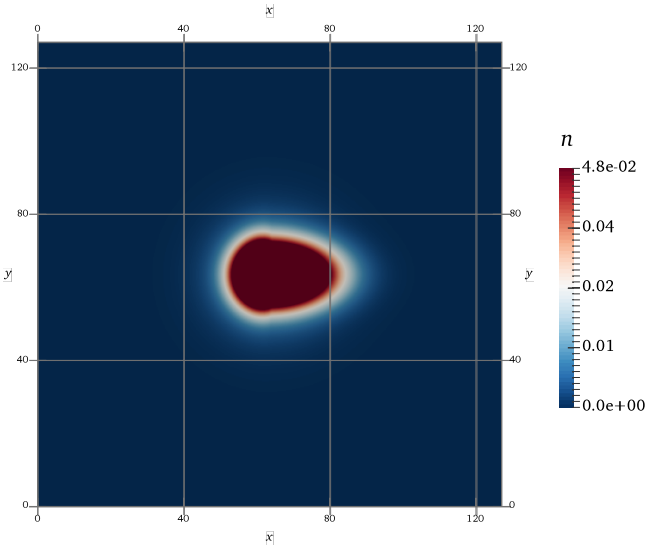


(b) contour plot

FIGURE 5. The left panel (a) shows the numerical solution of tumor cells density cells n evaluated at $t = 1$ unit of simulation time in the anisotropic, homogeneous diffusion case ($a = 0.01$, $c = 1.0$, $b = 0.0$) with weak haptotactic signal ($\gamma = 0.00005$). The right panel (b) shows the matching contour plot (color online).

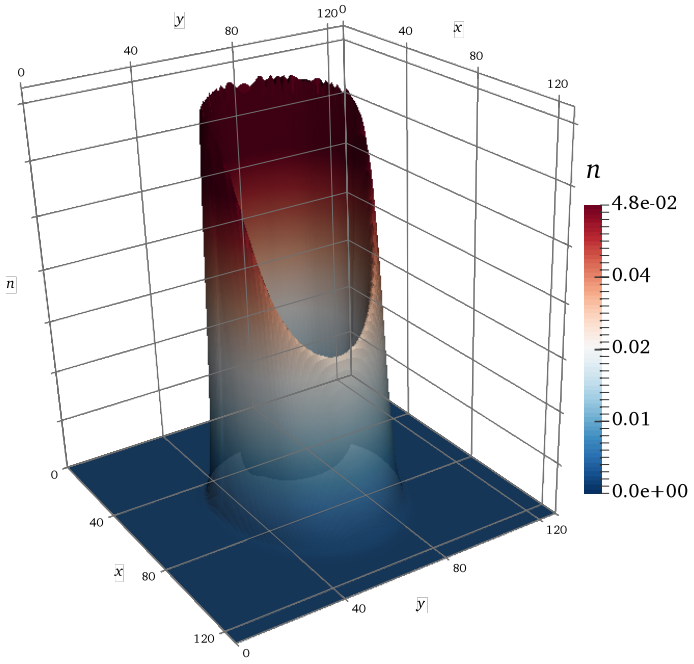


(a) tumor cells density

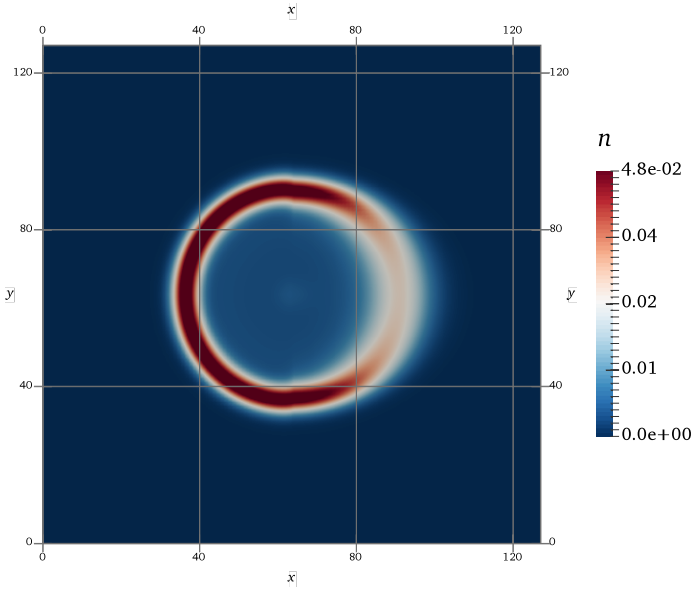


(b) Contour plot

FIGURE 6. The left panel (a) shows the numerical solution of tumor cells density n evaluated at $t = 1$ unit of simulation time in the case of anisotropic, inhomogeneous diffusion tensor (3.7) and without haptotaxis ($\gamma = 0.0$). The right panel (b) shows the matching contour plot (color online).



(a) tumor cells density



(b) Contour plot

FIGURE 7. The left panel (a) shows the numerical solution of tumor cells density n evaluated at $t = 1$ unit of simulation time in the case of anisotropic, inhomogeneous diffusion tensor (3.7) and with weak haptotactic signal ($\gamma = 0.00005$). The right panel (b) shows the matching contour plot (color online).

definition of two indices often used in parallel processing, the *speed-up* and the *efficiency* [12]. The benefits of parallel processing can be affected by all parts of the numerical code that cannot be actually parallelized. Indeed, every program necessarily performs some sequential calculations such as Input/Output data management, for example, and one must always take into account such interactions. Assuming that, when the program is compiled to produce a serial machine code, its wall-clock running time is T_s seconds, and that the fraction of this work that can be parallelized is $P \in (0,1)$. Thus, the parallelizable part accounts for PT_s seconds, whereas the remaining serial part is $(1-P)T_s$ seconds. Theoretically, once the program has been parallelized, the corresponding reduction in running time is PT_s/η, where η is the ratio of serial to parallel running time for the parallelizable code parts, namely η is proportional to the number of processors (or cores). In conclusion, the total running time gets modified to $T_p = (1-P)T_s + PT_s/\eta$.

The speed-up is a very simple index that allows to assess how much the problem solved in parallel mode is faster compared to the same problem solved purely in serial mode. It is defined as

$$S_p = \frac{T_s}{T_p} = \frac{1}{(1-P) + P/\eta} \geq 1\,, \tag{4.1}$$

and its expression is independent of T_s, according to the *Amdahl's law* [1, 12]. The theoretical upper bound for the speed-up is always equal to the number of parallel processes to which a given problem is distributed: if the program is fully parallelized, then $P = 1$ and the Amdahl's law (4.1) yields $S_p = \eta$. However, it is difficult to reach such a limit due to the various hardware bottlenecks associated with the computing architectures. In particular, the memory access time and data traffic usually degrade the performance compared to the ideal speed-up limit. The lower bound for the speed-up, for instance, is reached when no part of the code can be parallelized ($P = 0$ and $S_p = 1$).

In order to obtain an estimate of the effectiveness of a given computing architecture, it is useful to define the efficiency parameter as the ratio between the speed-up and the number of used processors, that is

$$E_p = \frac{S_p}{\eta}\,, \tag{4.2}$$

where S_p is the speed-up as defined in (4.1) and η is the number of usable processors (or cores). It is to be noticed that there exists a theoretical upper limit for the efficiency (4.2) given by $E_p \leq 1$. In other words, we have the highest efficiency when all available resources are actually employed. However, as mentioned above, the difficulty to reach the theoretical maximum speed-up implies that, therefore, the maximum efficiency is almost surely never reached.

Remark 4.1. In principle, it is not formally correct to effectuate performance evaluations by directly comparing CPU and GPU execution times, because that means to compare results related to two intrinsically different computing architectures (refer to [11, 13, 16]). For this reason, with the aim of homogenizing the

numerical results, we have introduced supplementary dedicated metrics to significantly juxtapose serial and parallel computational times related to our problem.

To evaluate the effectiveness of the numerical simulations carried out to compute approximated solutions to system (2.4), we define the *program speed-up* as

$$S_{pt} = \frac{T_{\mathrm{CPUtot}}}{T_{\mathrm{CPUser}} + T_{\mathrm{GPU}}}\,, \tag{4.3}$$

where T_{CPUtot} is the total CPU execution time of the program running in serial mode, T_{CPUser} is the CPU execution time of the irreducible serial part of the program and T_{GPU} is the GPU execution time of the parallel part of the program (equal to the CUDA kernel(s) execution time(s) in the distributed environment). Likewise, we define the *kernel speed-up* as

$$S_{pk} = \frac{T_{\mathrm{CPUpar}}}{T_{\mathrm{GPU}}}\,, \tag{4.4}$$

where T_{CPUpar} is the CPU execution time of the parallel part of the program running in serial mode. The corresponding efficiencies are defined as

$$E_{pt} = \frac{S_{pt}}{\eta}\,, \qquad E_{pk} = \frac{S_{pk}}{\eta}\,, \tag{4.5}$$

with $\eta = 1344$ the number of CUDA cores used for our simulation (see Section 3). We notice that $T_{\mathrm{CPUtot}} = T_{\mathrm{CPUser}} + T_{\mathrm{CPUpar}}$ and elementary algebra allows to infer that $S_{pt} < S_{pk}$, inasmuch as $T_{\mathrm{GPU}} < T_{\mathrm{CPUpar}}$ (the GPU execution time for the parallelizable part of the code is obviously smaller than the analogous CPU execution time). Consequently, it holds that $E_{pk} > E_{pt}$ as well.

Table 2 shows the performance expressed as execution time (in seconds) of the global program with its parallel components implemented in serial or alternatively parallel mode, in terms of the grid size, for the numerical simulations with strong haptotaxis ($\gamma = 0.005$). It also contains the computed values of the program speed-up S_{pt} and its efficiency E_{pt}. The results in the case without haptotactic effects ($\gamma = 0$) are very similar (see Table 3), thus confirming the optimality of the code implementation even in presence of spatially staggered coupling nonlinear terms (for $\gamma \neq 0$) inside the numerical algorithm (3.1).

Table 4 shows the performance expressed as execution time (in seconds) of the program parallel components executed by the CPU serial kernel or alternatively the CUDA parallel kernel, in terms of the grid size, for the numerical simulations with strong haptotaxis ($\gamma = 0.005$). It also contains the computed values of the kernel speed-up S_{pk} and its efficiency E_{pk}. The results in the case without haptotactic effects ($\gamma = 0$) are shown in Table 5.

Finally, Figure 8 shows the comparison between the global program speed-up S_{pt} in (4.3) and the kernel speed-up S_{pk} in (4.4) as functions of the grid size. Analogously, Figure 9 shows the program efficiency E_{pt} versus the efficiency of the parallel kernel E_{pk} defined in (4.5) as functions of the grid size. It is to be observed that $S_{pk} > S_{pt}$ and $E_{pk} > E_{pt}$, as we have pointed out above. We also notice

TABLE 2. **Performance of the global program (with haptotaxis)**. Second and third columns (from left to right) contain the computation time (in seconds) when employing serial and CUDA kernels, respectively. Fourth and fifth columns contain the relative program speed-up (4.3) and efficiency (4.5).

Grid size	Serial kernel	CUDA kernel	Speed-up	Efficiency
128×128	37.06	0.42	89.56	0.066
256×256	247.89	1.38	180.01	0.134
512×512	2909.48	5.39	539.60	0.402
1024×1024	6395.32	21.30	300.30	0.223
2048×2048	17847.28	84.24	211.87	0.158

TABLE 3. **Performance of the global program (without haptotaxis)**. Second and third columns (from left to right) contain the computation time (in seconds) when employing serial and CUDA kernels, respectively. Fourth and fifth columns contain the relative program speed-up (4.3) and efficiency (4.5).

Grid size	Serial kernel	CUDA kernel	Speed-up	Efficiency
128×128	36.97	0.42	89.19	0.066
256×256	247.67	1.38	189.80	0.134
512×512	2909.48	5.39	539.47	0.401
1024×1024	6298.98	21.30	295.79	0.220
2048×2048	16628.33	84.20	197.50	0.147

that there is a threshold value around a grid size of 512 points from which both speed-up and efficiency of the global program become significantly smaller than the speed-up and efficiency of the CUDA parallel kernel, suggesting that the GPU implementation of the numerical scheme enjoys an increasingly better performance than that of the CPU implementation as the space resolution improves.

5. Discussion and conclusions

In this article, we have reported the results of the numerical simulation of a system of partial differential equations modeling tumor growth. The system is based on a previous reaction-diffusion model by Enderling *et al.* [6], with the incorporation of an anisotropic and heterogeneous diffusion tensor for the dynamics of the tumor cells over the spatial domain. The model is further endowed with

TABLE 4. **Performance of the program kernels (with haptotaxis)**. Second and third columns (from left to right) contain the computation time (in seconds) when employing serial and CUDA kernels, respectively. Fourth and fifth columns contain the relative kernel speed-up (4.4) and efficiency (4.5).

Grid size	Serial kernel	CUDA kernel	Speed-up	Efficiency
128×128	40.78	0.42	97.10	0.072
256×256	272.62	1.40	194.73	0.145
512×512	3199.90	5.41	591.48	0.440
1024×1024	11358.36	21.31	533.01	0.400
2048×2048	57109.48	84.23	678.02	0.504

TABLE 5. **Performance of the program kernels (without haptotaxis)**. Second and third columns (from left to right) contain the computation time (in seconds) when employing serial and CUDA kernels, respectively. Fourth and fifth columns contain the relative kernel speed-up (4.4) and efficiency (4.5).

Grid size	Serial kernel	CUDA kernel	Speed-up	Efficiency
128×128	40.68	0.43	94.61	0.0704
256×256	272.39	1.40	194.56	0.145
512×512	3006.40	5.41	555.71	0.413
1024×1024	13415.00	21.31	629.51	0.468
2048×2048	52165.00	84.20	619.54	0.461

haptotaxis towards fibronectin molecules and extracellular matrix degradation by means of enzymes. The purpose of our numerical experiments has been to compare a parallel implementation on a GPU architecture of the numerical scheme with its serial CPU counterpart.

The simulations convey some qualitative observations. For example, upon inspection of Figures 2 to 6, it emerges that cases with haptotaxis exhibit greater directional movement of the tumor cells within the same time window. This behavior suggests that the presence of haptotaxis may lead to faster metastasis growth inside the tissue surrounding the tumor. Regarding the comparative performances of the GPU and CPU implementations (see Figure 8 and Figure 9), we observe a strong improvement on the execution time for each test, with a maximum speed-up of almost 700 and an efficiency around 0.5 for increasing grid sizes. Moreover,

it is to be noticed that, as the space resolution improves, the GPU parallel implementation of the numerical scheme shows an increasingly better performance than that of its CPU serial counterpart. We believe that our results confirm that GPU architectures provide the user with high computational performance (at low cost).

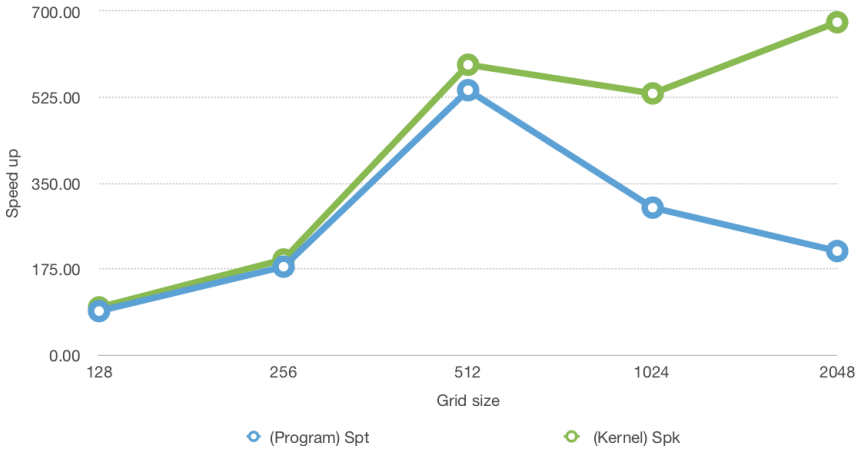


FIGURE 8. Speed-up of the global program S_{pt} (blue line) and of the CUDA parallel kernel S_{pk} (green line) as functions of the grid size, for the scheme (3.1)-(3.3) with haptotaxis ($\gamma \neq 0$).

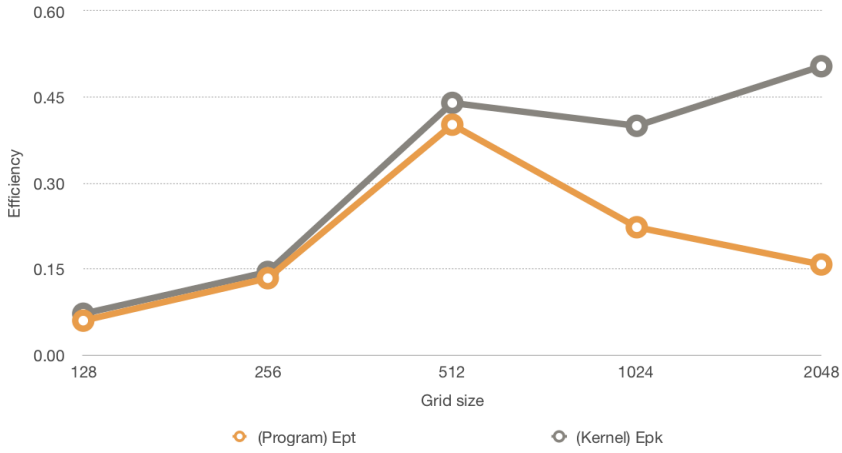


FIGURE 9. Efficiency of the global program E_{pt} (orange line) and of the CUDA parallel kernel E_{pk} (grey line) as functions of the grid size, for the scheme (3.1)-(3.3) with haptotaxis ($\gamma \neq 0$).

Acknowledgements

All numerical simulations in this article have been realized on the Linux HPC cluster Caliban – `http://caliban.dm.univaq.it` – located in the Laboratory of High Performance Parallel Computing at the University of L'Aquila [15].
This research has been supported by CONACyT (Mexico) and MIUR (Italy) through the MAE Program for Bilateral Research, grant no.146529. The research of R.G. Plaza has been partially supported by DGAPA-UNAM, program PAPIIT, grant IN100318.

References

[1] Amdahl, G. M., Validity of the single processor approach to achieving large scale computing capabilities, Proceedings of the Spring Joint Computer Conference AFIPS'67, April 18-20, 1967, New York, NY, USA, 1967, ACM, pp. 483–485 (1967)
[2] Anderson, A.R.A., A hybrid mathematical model of solid tumour invasion: the importance of cell adhesion, Math. Med. Biol. **22**, no. 2, 163–186 (2005)
[3] Anderson, A.R.A., Chaplain, M.A.J., Newman, E.L., Steele, R.J.C., Thompson, A.M., Mathematical modelling of tumour invasion and metastasis, J. Theor. Med. **2**, no. 2, 129–154 (2000)
[4] Dascal, L., Ditkowski, A., Sochen, N.A., On the discrete maximum principle for the Beltrami color flow, J. Math. Imaging Vision **29**, no. 1, 63–77 (2007)
[5] Dascal, L., Sochen, N.A., A maximum principle for Beltrami color flow, SIAM J. Appl. Math. **65**, no. 5, 1615–1632 (2005)
[6] Enderling, H., Anderson, A.R.A., Chaplain, M.A.J., Munro, A.J., Vaidya, J.S., Mathematical modelling of radiotherapy strategies for early breast cancer, J. Theor. Biol. **241**, no. 1, 158–171 (2006)
[7] Giese, A., Kluwe, L., Laube, B., Meissner, H., Berens, M. E., Westphal, M., Migration of human glioma cells on myelin, Neurosurgery **38**, 755–764 (2006)
[8] Goudon, Th., Mathematics for modeling and scientific computing, Mathematics and Statistics Series, ISTE London, John Wiley & Sons, Inc., Hoboken, NJ, USA (2016)
[9] Harten, A., Osher, S., Uniformly high-order accurate nonoscillatory schemes, SIAM J. Numer. Anal. **24**, no. 2, 279–309 (1987)
[10] Hubbard, M.E., Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids, J. Comput. Phys. **155**, no. 1, 54–74 (1999)
[11] Kirk, D.B., Hwu, W.-M.W., Programming Massively Parallel Processors: A Hands-on Approach, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2010)
[12] Kupferschmid, M., Classical Fortran: Programming for Engineering and Scientific Applications, CRC Press, Inc., Boca Raton, FL, USA, 2nd ed. (2009)
[13] NVIDIA CUDA C Programming Guide, Developer Manual, NVIDIA Corporation (2010)
[14] Painter, K.J., Hillen, T., Mathematical modelling of glioma growth: the use of diffusion tensor imaging (DTI) data to predict the anisotropic pathways of cancer invasion, J. Theoret. Biol. **323**, 25–39 (2013)
[15] Pera, D., May, J.B., Simeoni, C., The Linux High Performance Computing cluster Caliban at Univaq, preprint
[16] Sanders, J., Kandrot, E., CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 1st ed. (2010)
[17] Simons, J.E., Milewski, P.A., The volcano effect in bacterial chemotaxis, Math. Comput. Model. **53**, no. 7-8, 1374–1388 (2011)
[18] Swan, A., Hillen, T., Bowman, J.C., Murtha, A.D., A patient-specific anisotropic diffusion model for brain tumour spread, Bull. Math. Biol. **80**, no. 5, 1259–1291 (2018)

[19] Swanson, K.R., Alvord Jr., E.C., Murray, J.D., A quantitative model for differential motility of gliomas in grey and white matter, Cell Prolif. **33**, 317–329 (2000)

(D. Pera) Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica, Università degli Studi dell'Aquila, Via Vetoio (snc), Località Coppito, L'Aquila 67010 (Italy)

E-mail address: donato.pera@univaq.it

(C. Málaga) Departamento de Física, Facultad de Ciencias, Universidad Nacional Autónoma de México, Circuito Exterior s/n, Ciudad Universitaria, Ciudad de México 04510 (Mexico)

E-mail address: cmi.ciencias@ciencias.unam.mx

(C. Simeoni) Laboratoire de Mathématiques J.A. Dieudonné UMR CNRS 7351, Université de Nice Sophia-Antipolis, Parc Valrose 06108 Nice Cedex 02 (France)

E-mail address: simeoni@unice.fr

(R.G. Plaza) Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Circuito Escolar s/n, Ciudad de México 04510 (Mexico)

E-mail address: plaza@mym.iimas.unam.mx