Simulations of Cardiac Electrophysiology Combining GPU and Adaptive Mesh Refinement Algorithms

Rafael S. Oliveira[1(✉)], Bernardo M. Rocha[2], Denise Burgarelli[4],
Wagner Meira Jr.[3], and Rodrigo W. dos Santos[2]

[1] Departamento de Ciência da Computação,
Universidade Federal de São João de Rei, São João del Rei, Brazil
Sachetto@ufsj.edu.br

[2] Departamento de Ciência da Computação e Programa em Modelagem
Computacional, Universidade Federal de Juiz de Fora, Juiz de Fora, Brazil

[3] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais,
Belo Horizonte, Brazil

[4] Departamento de Matemática, Universidade Federal de Minas Gerais,
Belo Horizonte, Brazil

Abstract. Computer models have become valuable tools for the study and comprehension of the complex phenomena of cardiac electrophysiology. However, the high complexity of the biophysical processes translates into complex mathematical and computational models. In this paper we evaluate a hybrid multicore and graphics processing unit numerical algorithm based on mesh adaptivity and on the finite volume method to cope with the complexity and to accelerate these simulations. This is a very attractive approach since the electrical wavefront corresponds to only a small fraction of the cardiac tissue. Usually, the numerical solution of the partial differential equations that model the phenomenon requires very fine spatial discretization to follow the wavefront, which is approximately 0.2 mm. The use of uniform meshes leads to high computational cost as it requires a large number of mesh points. In this sense, the tests reported in this work show that simulations of three-dimensional models of cardiac tissue have been accelerated by more than 626 times using the adaptive mesh algorithm together with its parallelization, with no significant loss in accuracy.

1 Introduction

The phenomenon of electric propagation in the heart comprises a set of complex non-linear biophysical processes. Its multi-scale nature spans from nanometre processes such as ionic movements and protein dynamic conformation, to centimetre phenomena such as whole heart structure and contraction.

Computer models [4] have become valuable tools for the study and comprehension of such complex phenomena, as they allow different information acquired from different physical scales and experiments to be combined to generate

F. Ortuño and I. Rojas (Eds.): IWBBIO 2016, LNBI 9656, pp. 322–334, 2016.
DOI: 10.1007/978-3-319-31744-1_29

a better picture of the whole system functionality. Not surprisingly, the high complexity of the biophysical processes translates into complex mathematical and computational models. The modern cardiac models are described by non-linear systems of partial differential equations (PDE) coupled to a non-linear set of ordinary differential equations (ODE) resulting in a problem with millions of variables and hundreds of parameters.

The bidomain model [14] is considered to be the most complete description of the electrical activity in cardiac tissue. Under suitable assumptions the bidomain equations (a non-linear system of PDEs) may be reduced to a simpler model, called monodomain, which is less computationally demanding. Unfortunately, large scale simulations, such as those resulting from the discretization of an entire heart, still a computational challenge. In spite of the difficulties and the complexity associated with the implementation and use of these models, the benefits and applications justify their use. Computer models have been used during the tests of new drugs, development of new medical devices, new techniques of non-invasive diagnosis for several cardiac disease, cardiac arrhythmia, reentry, fibrillation or defibrillation and have been the research topic of many studies [2,7,11].

In the simulations of cardiac electrophysiology, the electrical wave front that travels through the heart is very sharp. Due to this sharp spatial variation the numerical methods need fine spatial discretizations to follow the wavefront, which is approximately 0.2 mm [5], to ensure sufficiently accurate results. The execution of cardiac simulations on meshes with a large number of nodes is computationally expensive as it requires repeated solutions of linear systems with millions degrees of freedom. In addition, the memory requirements of such simulations become increasingly large. The use of adaptive mesh methods provides a solution to these problems. By maintaining the extremely fine resolution only where it is needed (i.e., near the wavefront) the number of degrees of freedom is significantly reduced, resulting in faster computations, lower memory usage and reducing the need for disk space for the recording of the output files.

In this paper we extend the parallel accelerated adaptive mesh algorithm, presented in [9] by making the following improvements: 1. extension of the mathematical formulation to be able to make simulations using three-dimensional meshes; 2. inclusion of a graphics processing unit (GPU) parallel implementation to solve the system of ODEs; 3. inclusion of a pre-conditioner for solving the linear system associated to the PDE.

2 Monodomain Model

The wave of excitation propagates through the cardiac tissue because the cardiac cells are electrically coupled via special proteins called gap junctions. This phenomenon can be described mathematically by a reaction-diffusion equation called monodomain model, given by

$$\beta C_m \frac{\partial V(x,y,t)}{\partial t} + \beta I_{ion}(V(x,y,t), \boldsymbol{\eta}(x,y,t)) = \nabla \cdot (\boldsymbol{\sigma}(x,y)\nabla V(x,y,t)) + I_{stim}(x,y,t) \tag{1}$$

$$\frac{\partial \boldsymbol{\eta}(x,y,t)}{\partial t} = \boldsymbol{f}(V(x,y,t), \boldsymbol{\eta}(x,y,t)), \tag{2}$$

where V is the variable of interest and represents the transmembrane potential, i.e. the difference between intracellular to extracellular potential; $\boldsymbol{\eta}$ is a vector of state variables that also influence the generation and propagation of the electric wave, and usually includes the intracellular concentration of different ions (K^+, Na^+, Ca^{2+}) and the permeability of different membrane ion channels; β is the surface-volume ratio of heart cells; C_m is the membrane capacitance, I_{ion} the total ionic current, which is a function of V and $\boldsymbol{\eta}$, I_{stim} is the current due to an external stimulus and $\boldsymbol{\sigma}$ is the monodomain conductivity tensor. We assume that the boundary of the tissue is isolated, i.e., no-flux boundary conditions ($\mathbf{n} \cdot \sigma \nabla V = 0$ on $\partial \Omega$) are imposed.

In this work, two cell models from distinct species with different levels of complexity were considered to simulate the kinetics of the reaction term I_{ion} in Eq. 2. The Bondarenko et al. model [3] that describes the electrical activity of left ventricular cells of mice and the ten Tusscher-Panfilov model for human ventricular tissue [16]. The Bondarenko et al. model (BDK) model consists of the sum of 15 transmembrane currents. In short, Bondarenko's model is based on a system of ODEs with 41 differential variables that control ionic currents and cellular homeostasis. In this model most of the ion channels are represented by Markov chains (MCs). The ten Tusscher-Panfilov (TT2) model has 19 state variables which are described by ODEs, and it also describes the intracellular calcium concentration and conductances of the ionic channels. A complete description of the currents, the equations and parameters of the BDK and TT2 can be found in [3,16] respectively.

2.1 Finite Volume Model Applied to Monodomain

In this section we will make a brief description of the Finite Volume Method (FVM) applied to the monodomain equations. Details about the FVM applied to monodomain for two-dimensional problems can be found in [9,10].

The reaction and diffusion part of the monodomain equations can be split by employing the Godunov operator splitting. Each time step involves the solution of two different problems: a nonlinear system of ODEs

$$\frac{\partial V}{\partial t} = \frac{1}{C_m}\left[-I_{ion}(V, \eta_i) + I_{stim}\right] \tag{3}$$

$$\frac{\partial \eta_i}{\partial t} = f(V, \eta_i) \tag{4}$$

and a parabolic linear PDE

$$\frac{\partial V}{\partial t} = \frac{1}{\beta C_m}\left[\nabla \cdot (\boldsymbol{\sigma} \nabla V)\right] \tag{5}$$

The spatial discretization of the parabolic PDE results in a linear system of equations that has to be solved at each time step.

Time Discretization. The time derivative present in Eq. (5), which operates on V is approximated by an implicit first-order Euler scheme:

$$\frac{\partial V}{\partial t} = \frac{V^{n+1} - V^{n}}{\Delta t}, \tag{6}$$

where V^n represents the transmembrane potential at time t_n and Δt the time step.

Space Discretization. The diffusion term of Eq. (5) needs to be spatially discretized. To do this we will consider the following relations:

$$J = -\sigma \nabla V \tag{7}$$

where J (μA/cm^2) represents the density of the intracellular current flow and

$$\nabla \cdot J = -I_v. \tag{8}$$

In this expression, I_v(μA/cm^3) is a volumetric current and corresponds to the left side of Eq. (5).

For simplicity, we will consider a tri-dimensional uniform mesh, consisting of cubes (called "Volumes"). Situated in the center of each volume is a node and the transmembrane potential V is associated with each node of the mesh.

After defining the geometry of the mesh and the partitioning of the domain in control volumes, the FVM-specific equations can be presented. Equation (8) can be integrated spatially over a specific cube, leading to:

$$\int_{\Omega} \nabla \cdot J da = - \int_{\Omega} I_v \, da. \tag{9}$$

applying the divergence theorem, we find that

$$\int_{\Omega} \nabla \cdot J da = \int_{\partial\Omega} J \cdot \mathbf{n}, \tag{10}$$

where $\mathbf{n}$ is the vector normal to the surface.

Finally, assuming that I_v represents an average value in each particular cube, and using Eq. (5), we have the following relationship:

$$\beta \left(C_m \frac{\partial V}{\partial t} \right) \Bigg|_{(i,j,k)} = \frac{- \int_{\partial\Omega} J \cdot \mathbf{n}}{h^3}, \tag{11}$$

where h^3 is the volume of the control cell and $\mathbf{n}$ represents the vector normal to the surface.

For the three-dimensional problem, formed by a uniform grid of cubes with face area h^2, the calculation of J can be split as the sum of the flows on the six faces:

$$\int_{\partial\Omega} J \cdot \mathbf{n} = h^2 \cdot \sum_{l=1}^{6} J_l \tag{12}$$

where,

$$\sum_{l=1}^{6} J_l = J_{x_{i+1/2,j,k}} - J_{x_{i-1/2,j,k}} + J_{y_{i,j+1/2,k}} - J_{y_{i,j-1/2,k}} + J_{z_{i,j,k+1/2}} - J_{z_{i,j,k-1/2}},$$

The tensor $\sigma = \mathrm{diag}[\sigma_x, \sigma_y, \sigma_z]$ must be determined at the interfaces of the volume. For this, we use the harmonic mean:

$$\sigma_{x_{i+1/2,j,k}} = \frac{2\sigma_{x_{i,j,k}}\sigma_{x_{i+1,j,k}}}{\sigma_{x_{i+1,j,k}} + \sigma_{x_{i,j,k}}} \tag{13}$$

A similar reasoning can be used to calculate $\sigma_{x_{i-1/2,j,k}}$, $\sigma_{y_{i,j+1/2,k}}$, $\sigma_{y_{i,j-1/2}}$, $\sigma_{z_{i,j,k+1/2}}$ and $\sigma_{z_{i,j,k-1/2}}$.

The flows $J_{x_{m,n,o}}$, $J_{y_{m,n,o}}$ and $J_{y_{m,n,o}}$ are calculated at the faces $((m,n,o) = (i+1/2,j,k)$, $(i-1/2,j,k)$, $(i,j+1/2,k)$, $(i,j-1/2,k)$, $(i,j,k+1/2)$ or $(i,j,k-1/2))$ as follows:

$$J_{x_{m,n,o}} = \sigma_x(m,n,o)\left.\frac{\partial V}{\partial x}\right|_{(m,n,o)}, \tag{14}$$

$$J_{y_{m,n,o}} = \sigma_y(m,n,o)\left.\frac{\partial V}{\partial y}\right|_{(m,n,o)}, \tag{15}$$

$$J_{z_{m,n,o}} = \sigma_z(m,n,o)\left.\frac{\partial V}{\partial z}\right|_{(m,n,o)}. \tag{16}$$

Adaptive Non-uniform Mesh (ALG). When the electrical wave is propagating through the heart, only a fraction of the excitable medium is occupied by wavefronts. In these regions, the solution or its derivatives change rapidly. Therefore, the numerical solution of the differential equations in these regions requires the use of an extremely fine mesh. Thus, the use of uniform meshes leads to high computational costs. Therefore, adaptive procedures that take into account the scale differences in the phenomena present reliable and efficient solutions.

Recently, the use of adaptive refinement to obtain meshes suitable for the representation of the cardiac electrophysiology equations has been investigated, see, for example [1,15]. The application of ALG in the simulations of cardiac electrophysiology for 2-dimensional meshes can be found in [9,10]. In this work we are proposing the application of ALG in 3-dimensional cardiac meshes.

In order to apply FVM in ALG, we will approximate the partial derivatives of V on the interfaces using the following finite difference scheme, considering uniform discretizations in space ($\Delta x = \Delta y = \Delta z = h$). For sake of simplicity, we only show the equations for direction x since the equations for y and z can be obtained similarly.

$$\left.\frac{\partial V}{\partial x}\right|_{(i+1/2,j,k)} = \sum_{c=1}^{m_1} \frac{V_{r,c} - V_{i,j,k}}{h_1}, \tag{17}$$

$$\left.\frac{\partial V}{\partial x}\right|_{(i-1/2,j,k)} = \sum_{c=1}^{m_2} \frac{V_{i,j,k} - V_{l,c}}{h_2}, \tag{18}$$

where m_1 is the number of neighbors at right of the cell centered at (i, j, k) and m_2 is the number of neighbors at left; $V_{r,k}$ are neighbors at right, and $V_{l,k}$ are the neighbors at left. The discretizations are defined by:

$$\begin{aligned} h_1 &= h_{i,j} \text{ if } \mathcal{L}_{i,j} > \mathcal{L}_{r,k} \text{ and } h_1 = h_{r,k} \text{ otherwise,} \\ h_2 &= h_{i,j} \text{ if } \mathcal{L}_{i,j} > \mathcal{L}_{l,k} \text{ and } h_1 = h_{l,k} \text{ otherwise,} \end{aligned} \tag{19}$$

where $\mathcal{L}$ is the refinement level of the cell. Rearranging and substituting the discretizations in (11) and decomposing the operators as described by Eqs. (3), (4) and (5) yields:

$$\begin{aligned} &C_m \frac{V^*_{i,j,k} - V^n_{i,j,k}}{\Delta t} = \\ &- \frac{(S_1 J^*_{x_{i+1/2,j,k}} - S_2 J^* x_{i-1/2,j,k} + S_3 J^*_{y_{i,j+1/2,k}} - S_4 J^*_{y_{i,j-1/2,k}} + S_5 J^*_{z_{i,j,k+1/2}} - S_6 J^*_{z_{i,j,k-1/2}})}{\beta h^3_{i,j,k}} \end{aligned} \tag{20}$$

$$C_m \frac{V^{n+1}_{i,j,k} - V^*_{i,j,k}}{\Delta t} = -I_{ion}(V^*_{i,j,k}, \boldsymbol{\eta}^n) \tag{21}$$

$$\frac{\partial \boldsymbol{\eta}^{n+1}}{\partial t} = f(\boldsymbol{\eta}^n, V^*, t) \tag{22}$$

where:

$$S_1 J_{x_{i+1/2,j,k}} = -\sigma_{x_{i+1/2,j,k}} \sum_{c=1}^{m_1} \frac{V_{r,c} - V_{i,j,k}}{h_1} S_1 \tag{23}$$

$$S_2 J_{x_{i-1/2,j,k}} = -\sigma_{x_{i-1/2,j,k}} \sum_{c=1}^{m_2} \frac{V_{i,j,k} - V_{l,c}}{h_2} S_2 \tag{24}$$

For a regular grid we have $S_1 = h_1^2$ and $S_2 = h_2^2$, i.e., the area of the volume face. Therefore, we can simplify the above equations, obtaining:

$$J_{x_{i+1/2,j,k}} = -\sum_{c=1}^{m_1} \sigma_{x_{r',c}} (V_{r,c} - V_{i,j,k}) h_1 \tag{25}$$

$$J_{x_{i-1/2,j,k}} = -\sum_{c=1}^{m_2} \sigma_{x_{l',c}} (V_{i,j,k} - V_{l,c}) h_2 \tag{26}$$

where $\sigma_{x_{r',c}}$, $\sigma_{x_{l',c}}$ $\sigma_{y_{b',k}}$ are the conductivity values calculated using Eq. 13.

Developing all the equations, we can now define the formula for each volume:

$$\begin{aligned} \alpha V^{*}_{i,j,k} &- \sum_{c=1}^{m_1} \sigma_{x_{r'},c}(V_{r,c} - V_{i,j,k}) + \sum_{c=1}^{m_2} \sigma_{x_{l'},c}(V_{i,j,k} - V_{l,c}) \\ &- \sum_{c=1}^{m_3} \sigma_{y_{t'},c}(V_{t,c} - V_{i,j,k}) + \sum_{c=1}^{m_4} \sigma_{y_{b'},c}(V_{i,j,k} - V_{b,c}) \\ &- \sum_{c=1}^{m_5} \sigma_{z_{f'},c}(V_{f,c} - V_{i,j,k}) + \sum_{c=1}^{m_6} \sigma_{z_{bk'},c}(V_{i,j,k} - V_{bk,c}) \\ &= V^{n}_{i,j,k}\alpha \end{aligned} \tag{27}$$

where $\alpha = (\beta C_m h^3_{i,j})/\Delta t$.

3 Methods

In this section, we discuss the parallel numerical implementation and experimental setup of the various cardiac simulations we performed.

3.1 Parallel Numerical Implementations

Algorithm 1 describes the steps used for the numerical resolution of monodomain model. As can be seen, we have to reassemble the monodomain matrix at each time step if a refinement or derefinement operation has been performed in that step. In this paper, the criteria used for refinement and derefinement are based on the flux across the interface of neighboring cells, as described in [9].

Algorithm 1. Steps used for the numerical resolution of monodomain model

```
set cell model initial conditions;
assemble the monodomain matrix (Linear system form PDE);
while $t < final\_t$ do
    update cell Model state vector;
    solve cell model;
    solve linear system (PDE) via conjugate gradient method;
    refine-derefine
    reassemble the monodomain matrix if needed;
    $t = t + dt$
end while
```

Computer simulations, such as those resulting from fine spatial discretization of a tissue, are computationally expensive. For example, when a 100 µm discretization is used in a 5 cm × 5 cm × 5 cm 3D tissue, and the Bondarenko model, which has 41 differential equations, is used as cardiac cell model a total of $500 \times 500 \times 500 \times 41 = 5,125,000,000$ unknowns must be computed at each time step. In addition, to simulate 150 ms of cardiac electrical activity 5 billions of unknowns of the nonlinear systems of ODEs and the PDE with 125,000,000 of unknowns must be computed 15,000 times (with Δt = 0.01 ms). To deal with this high computational cost we parallelized, using OpenMP, the functions described in line 8 (assembly of the monodomin matrix) and 6 (conjugate gradient method); and using CUDA the function in line 5 (solution of odes) of Algorithm 1. The full description of the OpenMP implementation can be found in [10].

Differently from [10], in this work we use a Jacobi preconditioner [14] to accelerated the convergence of the conjugate gradient method. Despite being simple, the Jacobi preconditioner suites very well for the adaptive mesh algorithm, as we do not need to rebuild the preconditioning matrix every refine/derefine step, as this method uses only the diagonal of the linear system matrix as the preconditioning matrix.

To solve the non-linear systems of ODEs present in the BDK model, the explicit Euler (EE) method was used. Although it is well known that explicit numerical methods have strong limitations because of stability and accuracy restrictions, they are widely used due to their simplicity of implementation [6].

For the TT2 model, the numerical solution of ODEs at each volume was performed using the Rush-Larsen (RL) method [12]. The RL method is an explicit method, easy to implement and has better stability properties than the explicit Euler method. Thus, it allows the use of larger time steps resulting in an efficient method for the numerical solution of cell models of cardiac electrophysiology. Unfortunately, this method is not suitable for every model, like BDK due to the use of Markov chains [6].

The solution of these ODEs is a embarrassingly parallel problem regardless of the numerical method. No dependency exists between the solutions of the different systems of ODEs at each finite volume $Vol_{i,j,k}$. Therefore, it is quite simple to implement a parallel version of the code: each thread is responsible to solve a fraction of the non-linear systems of ODEs.

In order to accelerate even further our simulations, we also used GPU implementations, using CUDA, for the solutions of ODEs using both numerical methods (RL and EE). A description of these implementations can be found in [13].

3.2 Computational Simulations

In this section we present the numerical experiments and the computing environment used to perform them. We also report the results of the experiments and compare the performance of the parallel adaptive mesh approach with the fixed mesh implementation.

Computing Environment. All the numerical experiments were performed using a GNU/Linux 4.1.13 machine, with 32 GB of memory and a Intel Core i7-4930K 3.40 GHz processor with 6 cores. Our monodomain solver was implemented in C++ and compiled using GNU GCC 5.2.0 with $-O3$ optimization flag enabled. For the GPU tests we used a GeForce GTX 760, with 1152 CUDA cores organized in 6 multiprocessors. This GPU has a total of 4 GB of memory. The CUDA code was compiled with NVIDIA nvcc compiler version 7.5.17 with the same optimization flags of the CPU code.

Test Problems. In order to evaluate the acceleration and to validate our adaptive mesh implementation we used 2 different test problems, using simplified geometries. A brief description of these problems is presented here.

Benchmark Problem. In [8], a benchmark problem was proposed to help the validation of implementations of the monodomain model. In this problem, the domain is a rectangular region of size $2 \times 0.7 \times 0.3\,\mathrm{cm}^3$ and the fibers are parallel to the longitudinal direction and the conductivity tensor is considered transversely isotropic. The conductivity in the fiber direction is 1.334 mS/cm and the conductivity in the cross-fiber direction is 0.176 mS/cm. Other parameters are defined as: $\chi = 1400\,\mathrm{cm}^{-1}$ and $\mathrm{Cm} = 1\,\mu\mathrm{F/cm}^2$. The TT2 is used in this test case. An external stimulus of $-50\ 000\,\mu\mathrm{A/cm}^3$ is applied during 2 ms in a small cubic region of $0.15\,\mathrm{cm}^3$ at one corner of the tissue to trigger the electrical activity. This test problem will be referenced as *Test 1.*

Modified Benchmark. In order to better evaluate the performance of our implementation, we modified the Benchmark problem described above, by using the BDK model as the cellular model. This test case was develop as we wanted to evaluate the performance of our approach when using a more complex and complicated model to solve. This test problem will be referenced as *Test 2.*

As we interested in the impact of spatial discretisation and the parallel implementation on the execution times of the simulations, we solved the test problems using the following configurations:

- Test 1:
  - 150 ms of cardiac activity simulation.
  - $\Delta t = 0.05$ for both EDOs and EDP.
  - Fixed mesh with $\Delta x = 100\,\mu\mathrm{m}$. Adaptive mesh with minimum $\Delta x = 100\,\mu\mathrm{m}$ and maximum $\Delta x = 400\,\mu\mathrm{m}$ (referred to as 100–400). Adaptive mesh with minimum $\Delta x = 125\,\mu\mathrm{m}$ and maximum $\Delta x = 500\,\mu\mathrm{m}$ (referred to as 125–500).
- Test 2:
  - 150 ms of cardiac activity simulation.
  - $\Delta t = 0.05$ for EDP and $\Delta t = 0.0001$ for the EDOs.
  - Fixed mesh with $\Delta x = 100\,\mu\mathrm{m}$. Adaptive mesh with minimum $\Delta x = 100\,\mu\mathrm{m}$ and maximum $\Delta x = 400\,\mu\mathrm{m}$. Adaptive mesh with minimum $\Delta x = 125\,\mu\mathrm{m}$ and maximum $\Delta x = 500\,\mu\mathrm{m}$.

Figure 1(a)–(d) shows the propagation of the electrical wave in the benchmark problem mesh from the initial stimulus until its complete activation.

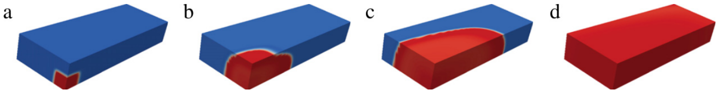


Fig. 1. Propagation of the electrical wave in the benchmark problem mesh.

4 Results

In this section, we present the results with respect to execution time and speedup associated with the solution of our test problems using our parallel adaptive mesh implementation.

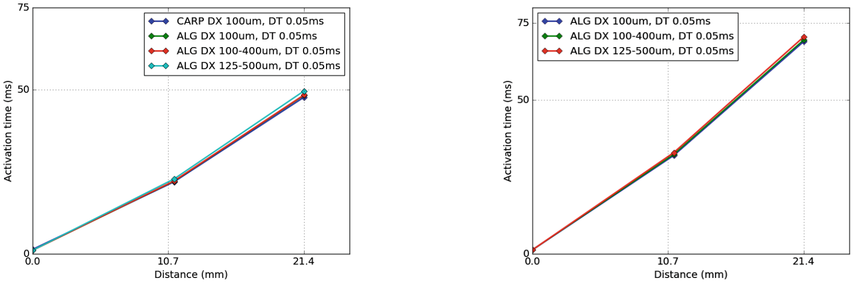
4.1 Test 1

The benchmark was solved using the configurations described in Sect. 3.2. All the parallel executions were performed using 6 threads. To measure the accuracy of our implementation we considered the activation time of a node, which was defined as the time at which the transmembrane potential v reaches the value of 0 mV. We used same metric that [8] used to compare the results of several codes.

Figure 2a shows the activation times of CARP [17] compared with our implementations using different mesh configurations. The activation times resulting for ALG simulations are almost identical to the one presented by CARP, either using fixed or adaptive meshes. This result shows that our implementation can be used to perform simulations of the electrical activity of cardiac tissue. Is worth noting that the OpenMP and the GPU code resulted in the same activation times.

Table 1 shows the achieved speedups for different mesh and code configurations when compared to the execution with a serial code using 100 μm fixed mesh ($\approx$ 175.2 min). By only using an adaptive mesh our implementation became 50.02× faster for the 125–500 configuration and 16.4× faster for the 100–400.

We also tested our implementations using the hybrid OpenMP and GPU version of the code and compared the resolution time with a serial code using 100 μm fixed mesh. As can be seen in Table 1, our code using multicore and GPU was 118.17× faster for the 125–500 configuration and 51.02× faster for the 100–400.



(a) Activation times of CARP, as reported in [8] compared to our implementation using three configurations.

(b) Comparison of the activation times for the modified benchmark.

Fig. 2. Comparison of the activation times for Test 1 and Test 2.

Table 1. Speedups over a 100 µm fixed mesh for Test 1.

Mesh	Code	Speedup
125–500	serial	50.02
125–500	OpenMP+GPU	118.17
100–400	serial	16.4
100–400	OpenMP+GPU	51.02

Table 2. Speedups over a 100 µm fixed mesh for Test 2.

Mesh	Code	Speedup
125–500	serial	255.65
125–500	OpenMP+GPU	626.75
100–400	serial	120.28
100–400	OpenMP+GPU	292.70

4.2 Test 2

After the validation of our implementation using the benchmark problem, we solved the modified benchmark in order to investigate the behaviour of our code when using a more complex cellular model as the BDK model. In Fig. 2b we have the activation times for the three configurations of Test problem 2. As can be seen, the activation time are very close for all mesh configurations.

In Table 2 we show the speedups by using parallel computing and adaptive meshes over the serial fixed mesh code (≈ 392.5 h). For this problem, our parallel implementation achieved speedups of 626.75× and 292.70× for the 125–500 and the 100–400 configurations.

The difference of the achieved speedups between the two test problems can be explained by one main reason: the BDK model used in Test 2 is more complex to solve than the TT2, used in Test 1. Because of this, if we solve less BDK EDOs (by using adaptive meshes) we are saving more time than with we solve less TT2 EDOs. Furthermore, for the same complexity reason, the GPU solver of the BDK model is more efficient as we have more computation over memory accesses than the TT2 GPU solver.

5 Conclusions

In this paper we developed, implemented, parallelized and validated an adaptive mesh strategy in order to speed up cardiac electrophysiology simulations for 3D domains. The achieved results are very promising, indicating that the use of ALG and parallel computing is able to reduce the execution time of a simulation by more than 626× (from 16 days to less then 38 min) for a complex cellular model, compared to the use of fixed meshes and serial executions using only a single node with 6 cores and a inexpensive GPU.

Acknowledgments. This work was partially funded by CNPq, Capes, Fapemig, UFJF and Finep.

References

1. Bendahmane, M., Bürger, R., Ruiz-Baier, R.: A multiresolution space-time adaptive scheme for the bidomain model in electrocardiology. Numer. Met. Partial. Diff. Equ. **26**(6), 1377–1404 (2010)
2. Biktashev, V., Holden, A.: Re-entrant activity and its control in a model of mammalian ventricular tissue. Proc. Royal Soc. of London. Ser. B Biol. Sci. **263**(1375), 1373–1382 (1996)
3. Bondarenko, V.E., Szigeti, G.P., Bett, G.C., Kim, S.J., Rasmusson, R.L.: Computer model of action potential of mouse ventricular myocytes. Am. J. Physiol. Heart Circulatory Physiol. **287**(3), H1378–H1403 (2004)
4. Hodgkin, A., Huxley, A.: A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. **117**, 500–544 (1952)
5. Hunter, P., Borg, T.: Integration from proteins to organs: the physiome project. Nat. Rev. Mol. Cell Biol. **4**(3), 237–243 (2003)
6. Moreira Gomes, J., Alvarenga, A., Silva Campos, R., Rocha, B., Couto da Silva, A., Weber dos Santos, R.: Uniformization method for solving cardiac electrophysiology models based on the markov-chain formulation. IEEE Trans. Biomed. Eng. **62**(2), 600–608 (2015)
7. Morgan, S., Plank, G., Biktasheva, I., Biktashev, V.: Low energy defibrillation in human cardiac tissue: a simulation study. Biophys. J. **96**(4), 1364–1373 (2009)
8. Niederer, S.A., Kerfoot, E., Benson, A.P., Bernabeu, M.O., Bernus, O., Bradley, C., Cherry, E.M., Clayton, R., Fenton, F.H., Garny, A., Heidenreich, E., Land, S., Maleckar, M., Pathmanathan, P., Plank, G., Rodríguez, J.F., Roy, I., Sachse, F.B., Seemann, G., Skavhaug, O., Smith, N.P.: Verification of cardiac tissue electrophysiology simulators using an n-version benchmark. Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci. **369**(1954), 4331–4351 (2011)
9. Oliveira, R.S., Rocha, B.M., Burgarelli, D., Meira Jr., W., dos Santos, R.W.: An adaptive mesh algorithm for the numerical solution of electrical models of the heart. In: Murgante, B., Gervasi, O., Misra, S., Nedjah, N., Rocha, A.M.A.C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2012, Part I. LNCS, vol. 7333, pp. 649–664. Springer, Heidelberg (2012)
10. Oliveira, R.S., Rocha, B.M., Burgarelli, D., Meira Jr., W., Santos, R.W.D.: A parallel accelerated adaptive mesh algorithm for the solution of electrical models of the heart. Inter. J. High Perform. Syst. Archit. **4**(2), 89–100 (2012)
11. Panfilov, A., Müller, S., Zykov, V., Keener, J.: Elimination of spiral waves in cardiac tissue by multiple electrical shocks. Phys. Rev. E **61**(4), 4644 (2000)
12. Rush, S., Larsen, H.: A practical algorithm for solving dynamic membrane equations. IEEE Trans. Biomed. Eng. BME **25**(4), 389–392 (1978)
13. Sachetto Oliveira, R., Rocha, B.M., Amorim, R.M., Campos, F.O., Meira Jr., W., Toledo, E.M., dos Santos, R.W.: Comparing CUDA, OpenCL and OpenGL implementations of the cardiac monodomain equations. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Waśniewski, J. (eds.) PPAM 2011, Part II. LNCS, vol. 7204, pp. 111–120. Springer, Heidelberg (2012)
14. Weber dos Santos, R., Plank, G., Bauer, S., Vigmond, E.J.: Preconditioning techniques for the bidomain equations. Lecture Notes in Computational Science and Engineering, vol. 40, pp. 571–580. Springer, Heidelberg (2004)

15. Southern, J., Gorman, G., Piggott, M., Farrell, P.: Parallel anisotropic mesh adaptivity with dynamic load balancing for cardiac electrophysiology. J. Comp. Sci. **3**, 8–16 (2012)
16. ten Tusscher, K.H.W.J., Panfilov, A.V.: Alternans and spiral breakup in a human ventricular tissue model. Am. J. Physiol.: Heart Circulatory Physiol. **291**(3), H1088–H1100 (2006)
17. Vigmond, E.J., Hughes, M., Plank, G., Leon, L.J.: Computational tools for modeling electrical activity in cardiac tissue. J. Electrocardiol. **36**, 69–74 (2003)