3D Heart Modeling With Cellular Automata, Mass-Spring System and CUDA

Ricardo Silva Campos[1], Ronan Mendonça Amorim[2], Bernardo Lino de Oliveira[3], Bernardo Martins Rocha[1], Joakim Sundnes[3], Luis Paulo da Silva Barra[1], Marcelo Lobosco[1], and Rodrigo Weber dos Santos[1]

[1] Universidade Federal de Juiz de Fora, Programa de Pós-Graduação em Modelagem Computacional, Juiz de Fora MG, Brazil,
rodrigo.weber@ufjf.edu.br,
WWW home page: http://www.fisiocomp.ufjf.br
[2] University of Calgary, Calgary, Canada
[3] Simula Research Laboratory, Norway

Abstract. The mechanical behavior of the heart is guided by the propagation of an electrical wave, called action potential. Many diseases have multiple effects on both electrical and mechanical cardiac physiology. To support a better understanding of the multi-scale and multi-physics processes involved in physiological and pathological cardiac conditions, a lot of work has been done in developing computational tools to simulate the electro-mechanical behavior of the heart. In this work, we implemented an aplication to mimic the heart tissue behavior, based on cellular automaton, mass-spring system and parallel computing with CUDA. Our application performed 3D simulations in a very short time. In order to assess the simulation results, we compared them with another synthetic model based on well-known partial differential equations(PDE). Preliminary results suggest our application was able to reproduce the PDE results with much less computational effort.

1 Introduction

Cardiac disorders are the major cause of death worldwide. Therefore the scientific community has done a lot studies in order to find the causes of heart diseases. The heart's role is pumping blood to all organs, allowing the body to change CO_2 by O_2 with lungs. The mechanical contraction of the cardiac tissue that ejects blood is preceded and triggered by a fast electrical wave, i.e. the propagation of the so called action potential (AP). Abnormal changes in the electrical properties of cardiac cells as well as in the structure of the heart tissue can lead to life-threatening arrhythmias and fibrillation.

A widely-used technique to observe the heart behavior is in silico experiments. They comprise in mathematical models that can reproduce the heart's tissue function through computational tools. Generally these models are described by differential equations, representing the cell's electrical and mechanical activity by ordinary differential equations (ODEs), and the electrical wave propagation on the tissue and cardiac mechanics via partial differential equations (PDEs). Cardiac cells are connected to each other by gap junctions creating a channel between neighboring cells and allowing the flux of electrical current in the form of ions. An electrically stimulated cell transmits an electrical signal to the neighboring cells allowing the propagation of an electrical wave to the whole heart which triggers contraction.

Although PDEs are able to perform realistic tissue simulations, they involve the simulation of thousands of cells, which make its numerical solution quite challenging. This is an issue for clinical softwares, that may demand accurate results and real-time simulations. In this manner, some effort has been done in speeding up the solving process of PDEs by parallel computing, as well different techniques to emulate PDEs with less computational cost.

This work implements a cellular automata model to represent electrical excitation of cells propagating according to simple rules in a regular, discrete and finite network. It uses precomputed profiles of cell AP and force-development that mimics those obtained by complex models based on ODEs. Although it is less accurate than the models based on ODEs, it is much faster than PDE based-simulators, making possible real time simulations of heart behavior. We present a 3D cellular automata (CA) simulator of the electrical activity of the heart coupled with a mass-spring system to simulate the cardiac mechanical behavior. Our mechanical model is governed by Hooke's Law, plus a damping and a volume preserving equation.

With our simulator we were able to evaluate interesting cases such as the influence of ischemic cells on the generation of spiral waves and the mechanical behavior under this pathological condition. In order to simulate thousands of cells we have parellelized our application with CUDA, so that the simulation runs on Graphic Processing Units. Because of the embarrassingly parallel nature of the CA and the mass-spring system, the simulator was able to allow real-time simulation for relatively large setups, i.e. for cardiac tissues composed by large number of cells.

Some electro-mechanical heart models based on cellular automaton can be found in the literature, in 2D and 3D [7, 8]. The main contribution of this work is adding a preserving volume restriction to the model. This is important to get accurate simulations, since the cardiac tissue is mostly composed of water, therefore it does not have big changes in its volume. Despite its incompressibility, the tissue contraction can cause a big change in its thickness. This feature can not be simulated by simple mass-spring systems with damping[9]. Futhermore, this work is an evolution of our previous cellular automaton-based model of the heart[6], that was able only to perform 2D simulations of compressible model.

2 Methodology

The implementation of this work is related with a cellular automaton to represent the action potential propagation through the tissue, another cellular automaton to model the force development coupled with the electrical activity and the

Newton's Law to model a mass-spring system. Futhermore we have added a damping coefficient and volume preserving equation. And finally, in order to allow simulations with a lot of elements, we proposed a parallel implementation with CUDA.

2.1 Cellular automata

A cellular automaton is the model of a spatially distributed process that can be used to simulate various real-world processes. A 2-dimensional cellular automaton consist of a two-dimensional lattice of cells where all cells are updated synchronously for each discrete time step. The cells are updated according to some updating rules that will depend on the local neighborhood of a particular cell.

The Cellular Automata (CA) can be used to simulate macroscopically the excitation-propagation in the cardiac tissue. If the electrical potential of a cell exceeds a threshold the cell gets excited and this can trigger the excitation of the neighboring cells. Therefore, an electrical stimulus will propagate through the CA grid as the time steps are computed.

The following sections will describe the CA approach for simulating the anisotropic electrical propagation in the cardiac tissue and the force-development in each cell. In this work, the CA states are related to the action potential (AP) and force development in a cell. To make CAs more efficient they are usually parametrized using simulated data from accurate models. This means that the states related to the AP in the cell will be related to a specific portion of the cardiac cell potential. Figure 1 presents the AP computed by ODEs, the AP divided into five different states and an interpolation of the discrete values of AP computed by CA. The interpolation granted smooth propagations of the action potential with low computational costs.

In state S0 the cell is in its resting potential where it can be stimulated, in S1 the cell was stimulated and can stimulate the neighbors. In S2 the cell is still able to stimulate the neighbors. In S3 the cell is in its absolute-refractory state where it cannot be stimulated and does not stimulate its neighbors. In S4 the cell is in its relative refractory state where it can be stimulated by more than one neighbor but it does not stimulate its neighbors. As described, the states of a cell generate rules for when a cell can stimulate a neighbor and when it can be stimulated. These rules are an important aspect which will allow the stimulus to propagate. Another important point is how the cells change their states. The AP has a predetermined period so that the states will be spontaneously changed when an AP starts, according to the timing presented in Figure 1.

Some fatty build-ups in arteries can totally or partially block the blood irrigation in parts of cardiac tissue causing it to die (not propagating stimulus, dead cells) or to behave differently (ischemic cells). The ischemic cells of the tissue present an AP with different properties than a healthy cell. Usually it has a shorter duration of the AP with different potential values. Table 1 presents the CA states, potential and times for both healthy and ischemic tissue. The contraction of a cardiac cell is coupled with the electrical potential of the cell.

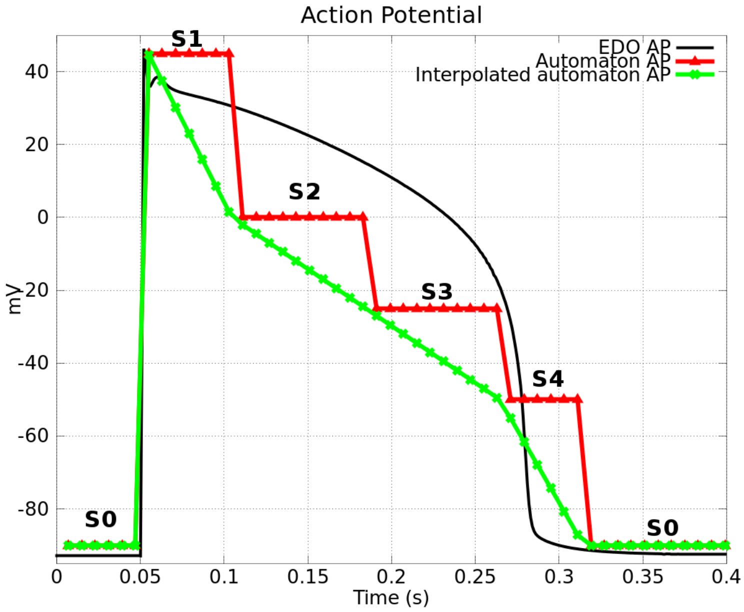

Fig. 1. Action potential of a cardiac cell:The black line is the AP computed by ordinary differential equations. The red line is the AP separated into five different states (S0, S1, S2, S3, S4) for the cellular automata, and the green line is the interpolation of the discrete values of the CA-computed AP.

When the cell is stimulated, there is an increase in the concentration of calcium ions inside the cell, which trigger the cell contraction. The force development has a delay after the cell is stimulated because of its dependence on the calcium ions. The force development of a cell can be represented in states that change over time like the electrical potential states. Figure 2 presents the force development states and its relation with the electrical states. The force-development states will only pass from state F0 (no contraction force) to state F1 when the electrical state of the cell goes from state S1 to S2. After this change, force development will be time dependent but will not depend on the electrical state of the cell.

Propagation Rules The electrical propagation velocity in the cardiac tissue is dictated by the fiber direction and tissue type, and it has three axis of anisotropy. Each one has a different electro-mechanical behaviour. Figure 3 shows a ventricular tissue segment, where the dark-gray lines are the three axes. The first direction is the fiber axis, where the wave propagation velocity is faster. The sheet direction is orthogonal to the fiber, and the third direction is called sheet-normal, that is normal to the plane formed by the fiber and sheet axis.

Since the anisotropy must influence the AP propagation of the CA, we set up a velocity of propagation to each direction. So we find the direction of the neighbor element, in order to compare its direction with the fibers and then

	Healthy Cell		Ischemic Cell	
CA State	Duration	Potential	Duration	Potential
S0	in rest	-90mV	in rest	-70mV
S1	50ms	+20mV	50ms	0mV
S2	80ms	0mV	40ms	-40mV
S3	80ms	-25mV	25ms	-60mV
S4	50ms	-50mV	10ms	-65mV

Table 1. Healthy and ischemic states of the cellular automata and the respective duration and potentials.

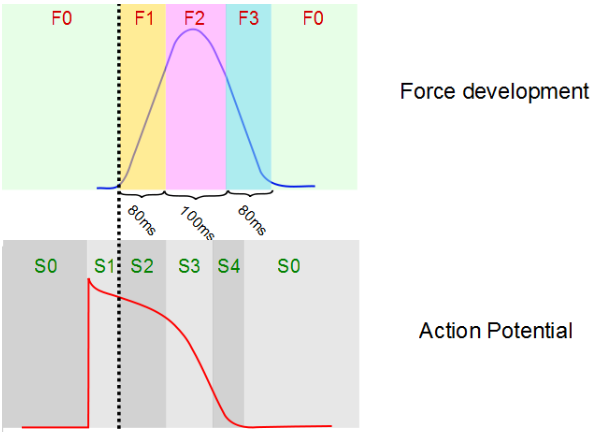

Fig. 2. Force development states in relation with the cell electrical states (Adapted from [6]).

determine the resultant velocity. Therefore, the traveling time of a stimulus from a cell to another is found, and this depends on the directions of the propation.

The activation of a cell will depend on the time of the activation of its neighbors. For each time step each cell checks if the neighboring cells are activated. If an activated cell is found the time of the stimulus to travel from that cell is computed and compared with the activation time of the same cell to check if there was enough time to the stimulus to travel. For the activation to take place the neighboring cells must also be in state S1 or S2, i.e. states that allow one cell to stimulate another. After a cell is stimulated it will, independently of the neighboring cells, dynamically change its state until it finally goes to the states S4 and S0, when it may be stimulated again. With this set of rules the stimulus is able to propagate through the CA simulating the electrical propagation of APs on the cardiac tissue.

The CA is discretized in space and time. The 3D CA is composed by small cubes with edges $dx \times dx \times dx$, where dx is the space discretization. The vertices of the cube are the masses. CA states are updated at every discretized time, dt.

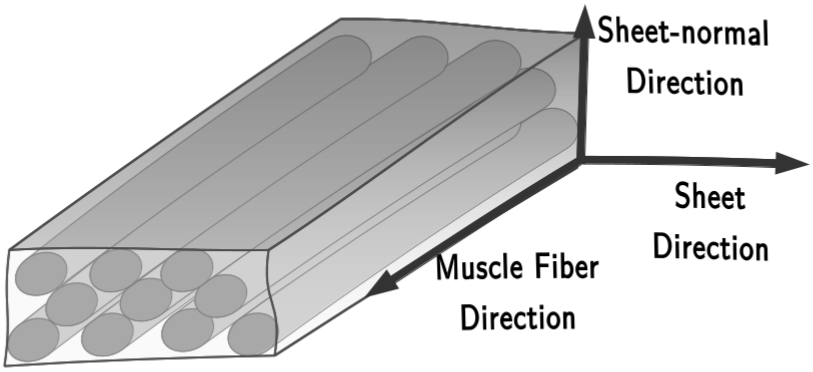

Fig. 3. Fiber directions in cardiac tissue

Based on this information and the velocities we can calculate the time that a stimulus takes to travel from one CA cell to another. For simplification, imagine that the propagation has the same velocity v in all directions (isotropic tissue). To find the time t for a stimulus travel from the center of one cell to another, first we have to find the directions of the neighbor:

$$\overrightarrow{Dir}_{c\_n} = \frac{\overrightarrow{X}_n - \overrightarrow{X}_c}{\|\overrightarrow{X}_n - \overrightarrow{X}_c\|} \tag{1}$$

Where $\overrightarrow{Dir}_{c\_n}$ is the neighbor direction relative to the current element, $\overrightarrow{X}_n$ and $\overrightarrow{X}_c$ are the coordinates of the neighbor and current element.

We have the three vectors of the anisotropy axes, fiber, sheet and normal-sheet: $\overrightarrow{F}$, $\overrightarrow{S}$ and $\overrightarrow{NS}$, as well are given three scalars representing the velocities on each one of axis, respectivelly: vel_f, vel_s and vel_{ns}. So, to calculate the velocity of the wave propation between two elements, we use the dot product:

$$V_f = vel_f(\overrightarrow{Dir}_{c\_n} \cdot \overrightarrow{F}) \tag{2}$$

$$V_s = vel_s(\overrightarrow{Dir}_{c\_n} \cdot \overrightarrow{S}) \tag{3}$$

$$V_{ns} = vel_{ns}(\overrightarrow{Dir}_{c\_n} \cdot \overrightarrow{NS}) \tag{4}$$

$$\tag{5}$$

Once we have the vector $\overrightarrow{V} = [V_f \ V_s \ V_{ns}]$ we find the final velocity by applying a norm to $\overrightarrow{V}$:

$$v = \|\overrightarrow{V}\| \tag{6}$$

To find the distance between the elements, we also apply a norm:

$$d = \|\overrightarrow{X}_n - \overrightarrow{X}_c\| \tag{7}$$

Finally, the traveling time from one element to another is found:

$$t = \frac{d}{v} \tag{8}$$

Assuming a Moore neighborhood for 2D with radius 1, the immediate top, bottom, left, right and all diagonals are considered neighbors. For the diagonals, a different distance of $dx\sqrt{2}$ is assumed, as presented in Figure 4 A. For a 3D CA, the definition of Moore neighborhood is analogous. In Figure 4 C, we have a central element (in grey) and its 27 neighbors (in black).

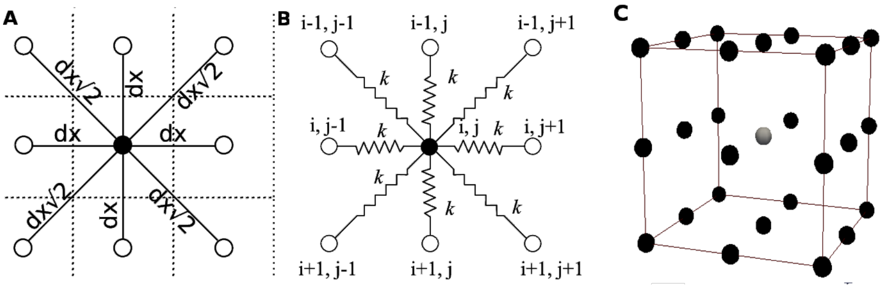

Fig. 4. A) The distances in Moore neighborhood of a cell with radius 1. B) Interconnected set of masses by springs. C) 3D Moore Neighborhood.

2.2 Mechanical Model

The modeling of cardiac tissue deformation can be simplified by the use of mass-spring systems. In such systems, masses are connected with the neighboring masses by springs and forces can be applied to the system deforming its spatial distribution. The springs of the system will try to bring the system to its initial configuration again. The cardiac tissue does not have a linear stress-strain relation. However the linear model of the Hooke's law can be used as a simplification. For each mass, the total force is the summation of the Hooke's law between each one of its connecting neighbors:

$$\overrightarrow{F}^i_s = \sum_{j=0}^{n} (-k\overrightarrow{\Delta^j}) \tag{9}$$

where k is the spring constant that determines the stiffness of a spring, n is the number of neighbors and $\overrightarrow{\Delta^j}$ the displacement between mass i and its neighbor j. But such systems will oscillate forever. In practice there will be forces in the environment that will resist to the movement. Such forces are called damping forces $\overrightarrow{F}_d$ and are proportional to the velocity of the mass $\overrightarrow{V}^i(t)$:

$$\overrightarrow{F}^i_d = -\beta\overrightarrow{V}^i(t) \tag{10}$$

where β is the damping coefficient. The interconnected set of masses (lattices) are depicted in 4 B. We also included in the system the contraction force of the cell $\overrightarrow{F}_c$, and another force in order to preserve the volume.

The preserving force is important for accurate simulations. Since the cardiac tissue is mainly composed by water, it does not have significant changes in its volume. To add this feature in the mass spring system, we applied a force to each mass in order to keep the same volume during the simulation, although the tissue contracts. This force depends on each cube surroundding the mass. It emanates from the cube's baricenter $\overrightarrow{X}_b$, and its module depends on the cube's volume variation. The resulting formula is given below:

$$\overrightarrow{F}^i_v = \sum_{\forall j \in \Omega i} k_v \frac{(V^j_t - V^j_0)}{V^j_0} \frac{\overrightarrow{X}_i - \overrightarrow{X}^j_b}{\|\overrightarrow{X}_i - \overrightarrow{X}^j_b\|} \tag{11}$$

where V^j_t is the volume of cube j sorrounding mass i with coordinates X_i. V^j_0 is the initial volume. The cubes on the neighborhood of mass i are represented by Ωi. The total force on each mass is:

$$\overrightarrow{F}^i_{total} = \overrightarrow{F}^i_s + \overrightarrow{F}^i_d + \overrightarrow{F}^i_c + \overrightarrow{F}^i_v \tag{12}$$

From Newton's second law we have that $F = ma$ and equating this with the total force we have:

$$\overrightarrow{a}^i(t) = \frac{F^i_{total}}{m} \tag{13}$$

Finally it is necessary to integrate the system of equations for each cell in the CA to simulate the mechanical deformation of the tissue. Using the Forward Euler's method we have

$$\overrightarrow{V}^i(t + dt) = \overrightarrow{V}^i(t) + \overrightarrow{a}^i(t)dt \tag{14}$$

$$\overrightarrow{P}^i(t + dt) = \overrightarrow{P}^i(t) + \overrightarrow{V}^i(t)dt \tag{15}$$

where $\overrightarrow{V}^i(t)$ and $\overrightarrow{P}^i(t)$ are the velocity and position of a mass at time t. In this way, CA and mass-spring models are coupled by F_c, i.e by the force generated by the cell during the dynamically change of its AP.

There are different approaches for modeling the electro-mechanical behavior of the heart. In [2], a method based on CA is presented that can simulate electrical propagation in 3D cardiac tissue with arbitrary local fiber orientations. In [3], a CA is used to simulate cardiac electrical propagation and a model based on the finite element method is used to simulate cardiac mechanics. In [4] the CA approach is used for the electro-mechanical simulation of the cardiac tissue. A comparison of models for cardiac tissue based on differential equations and on CA is presented in [5].

2.3 Parallel Computing

Our CA coupled with mass-spring system is not computational expensive. A mesh with a thousand elements can be simulated in a few minutes. However, our intentions for future work is to perform simulations for the whole heart, which may demand millions of elements. Futhermore, we also need to estimate some parameters using genetic algorithms (GA), in order to find accurate simulations. For this GA, it is necessary to evaluate a few hundreds of the CA, with different parameters. According to the comparison to real data, the parameters that resulted in the best outcome are chosen for the next iteration, and the other parameters are adapted. This process is repeated until the best solution reachs an acceptable error. Undoubtedly, this technique is computational expensive and may take some hours to get finished. So, in order to speedup the simulation, we proposed a parallel implementation using Graphic Processing Units (GPU) with CUDA (Compute Unified Device Architecture). In this architecture, it is necessary to copy the elements' matrix that represents the CA from the CPU (host) to the GPU (device). This is done in the beginning of the computation as follows:

```
//copy data from CPU to GPU
copyHostToDevice();
//set up the number of threads to each block
dim3 threads(3, 3, 3);
dim3 grid_size; //set up the grid
grid_size.x = (width + threads.x)/ threads.x;
grid_size.y = (height + threads.y)/ threads.y;
grid_size.z = (depth + threads.z)/ threads.z;
while(t<finalTime){
   time += dt; //increments simulations time
   simulate<<<grid_size, threads>>>();
}
```

In this code, the CA matrix is divided into blocks containing $3\times3\times3$ threads. The distinct blocks form the matrix grid that is set up according to its dimensions, *height*, *width* and *depth*. This pseudo code is run in the CPU. The functions called with the triple angle bracket are invoked by the CPU but they actually are run in the GPU.

3 Results

Different tests with the implementation were performed. The simulations used a cubic tissue with $9\times9\times9$ cells with a spatial discretization of $0.001m$ and a time discretization of $1ms$. Conduction velocity is assumed to be $0.5m/s$, $0.17m/s$ and $0.17m/s$ along the fiber, sheet and normal sheet axis, respectively. A stimulated cell is placed in the left-bottom of the tissue and the propagation will be studied. The first test assumed horizontal fiber orientation for all cells of the tissue, and all cells in the system are healthy. The tests' results are presented in Figure 5.

The second test also assumed a horizontal fiber, but it was included ischemic cells. At rest, healthy cells have $-90mV$, identified in the figures by the grey color. Ischemic cells have $-70mV$ and at their rest and they are colored with

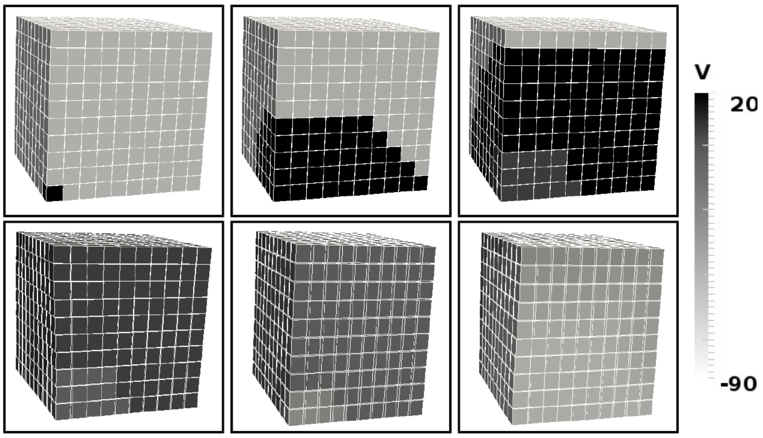

Fig. 5. Simulation of cube with $10mm$ edge, all cells are healthy and the stimulation starts in left-bottom corner, and fiber is horizontal.

a darker grey shade. This configuration is presented in Figure 6. Because of its shorter AP duration the ischemic tissue is able to be stimulated again earlier than the healthy tissue. This leads to the creation of a reentry circuit where the tissue keeps stimulating itself forever via the propagation of spiral wave. This causes arrhythmic contractions of the cardiac tissue.

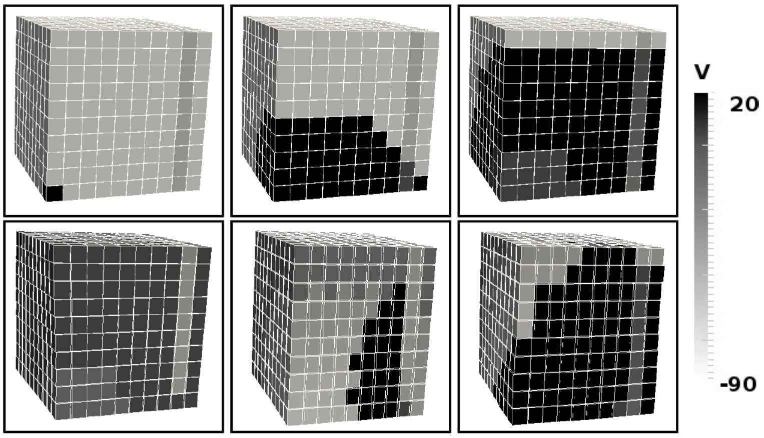

Fig. 6. Simulation with ischemic cells.

The third and fourth tests compared the influence of the fiber's direction in the AP propagation. In the first row in Figure 7, the AP propagates through a tissue with all tissue with in horizontal direction. In the second row, the fiber is at 45°. It can be observed that the stimulus propagates faster through the tissue in the preferential direction of the fiber, and the mechanical contraction is also fiber-prefered. Another test was performed to evaluate the fiber influence on the AP propagation, where the same tissue was mapped with two different fibers. In Figure 8, the upper half of the tissue has horizontal fibers and the bottom half has a 45° inclined fiber. The stimuling cell is central, and the wave propagates according to fiber direction on each cell.

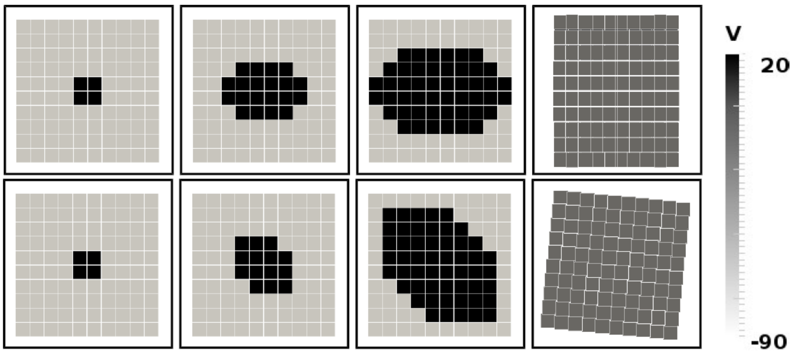

Fig. 7. Simulating with different fiber directions. First row with fibers at 0° and second row at 45°.

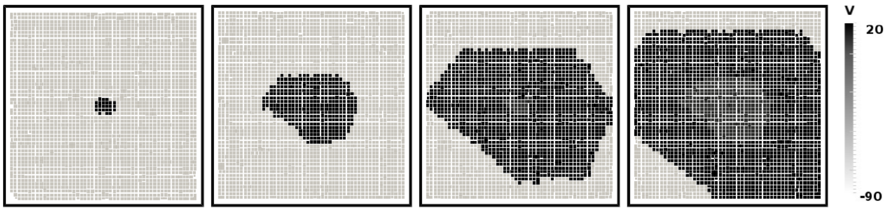

Fig. 8. Same tissue with different fibers directions.

In order to compare the CA results with differential equations (DE), we have evaluated the wall thickness (WT) of the tissue during simulation and the execution time. Here, we calculate the WT as the average thickening of the tissue wall in the transmural direction. Both simulations produced a maximum of 35% WT. The implementation of DE solver is described in [10]. Briefly, it contains an eletromechanical cell model that computes electrophysiology and active force generation, coupled via calcium concentration, and the action potential propagation through the tissue is modeled with the bidomain equations.

;Comparing the execution times, we perform a test with the same configuration for both CA and DE. With a $10mm^3$ mesh, the CPU execution time for DE is 13h30min(48600s). On the other hand, the CA execution in CPU took 38s and the GPU version 7s, that respectivelly are 1200 and 6800 times faster than the simulation based on DE.

The volume preserving force (F_v) was able to decrease the volume variation. In a compressible tissue, the volume could change until 40% of its initial value. With the incompressible force, it changed at the most 10% of its initial volume. Figure 9 presents how F_v acts for volume preserving. There are two tissues configurations: compressible (part A) and incompressible (part B). F_v acts from the baricenter of each component cube, and resultant force is best visible at the mesh's corners. After a contraction in a direction, the tissue "growns" in the other directions in an attempt to keep the volume constant.

In order to our mass-spring system to reproduce a DE simulation is necessary determine parameters, which is tiresome trial-and-error process, altough it was possible to get similiar results. Nevertheless, it is still necessary to find a better technique to estimate parameters.

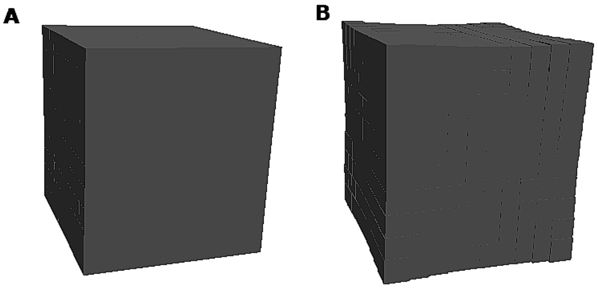

Fig. 9. Comparison between compressible (A) and incompressible (B) tissues.

Finally, we studied the impact of parallelizing the application. All tests were performed in an Intel i7 machine with 8 GB of RAM, Nvidia GeForce GTX 480, running Ubuntu/Linux 4.6.1 with gcc 4.6.1. We run both implementations (sequential CPU and parallel GPU) with different meshe sizes. We simulated $1s$ of the tissue activity, all tests were run four times. We present the average execution time, in seconds and in all cases the standard deviation observed in the execution times was less than 1%. The speedup was computed considering the average execution time for each mesh size:

$$Speedup = \frac{T_{CPU}}{T_{GPU}} \tag{16}$$

Table 2 presents the performance of the sequential and CUDA codes. It is possible to observe that the speedup is bigger with bigger meshes. This happens because there is an overhead for managing threads and GPU calls. There is also an implicit barrier at end of iteration, since it is necessary to synchronize the Euler integration method where one iteration is dependent on the previous one. When small meshes are run, the overhead is not negligible comparing with their small execution time.

Mesh size	$10mm^3$	$30mm^3$	$50mm^3$
GPU avg. time(s)	7	148	631
CPU avg. time(s)	38	1190	5697
Speedup	5.4	8.0	9.0

Table 2. Performances

4 Conclusion

This work presented a 3D simulator of the electro-mechanical activity of cardiac tissue via the coupling of CA and mass-spring models. Although models based on PDEs are more accurate and detailed, they are very computationally expensive. CA has shown to be an alternative for real-time simulation because of its fast performance, that resulted in a 1200 to 6800-fold improvement on computational time. The pattern of propagation obtained with CA has shown to be very similar to the patterns obtained with PDE-based models, including WT and volume, although a more detailed comparison is necessary. The tissue deformation obtained with the mass-spring system has shown to be very responsive to the force-development providing a qualitative demonstration of cardiac contraction. The volume preserving force was able to avoid big changes in the tissue without changing its WT, although manual determing parameters for reproducing PDE simulations is a hard task. The parallel implementation speeds up the simulations by 9 times, but it is still necessary to investigate improvements for the code to decrease overheads for faster simulations.

As future work, some improvements can also be achieved in terms of determing parameters of both CA and mass-spring systems, adjusting them to data obtained in experiments or from more realistic heart models.

References

1. Campos, R.S., Amorim R.M., Costa, C.M., Oliveira, B.L., Barbosa, C.B., Sundnes, J., Santos, R.W.: Approaching cardiac modeling challenges to computer science with CellML-based web tools. Future Generation Computer Systems, vol 26, 462–470 (2010)
2. Feldman, A.B., Murphy, S.P., Coolahan, J.E.: A method for rapid simulation of propagating wave fronts in three-dimensional cardiac muscle with spatially-varying fibre orientations. Engineering in Medicine and Biology(2002)

3. Cimrman, R., Kroc, J., Rohan, E., Rosenberg, J., Tonar, Z.: On coupling cellular automata based activation and finite element muscle model applied to heart ventricle modelling. 5th International Conference on Simulations in Biomedicine - Advances in Computational Bioengineering (2003)
4. Hurmusiadis, V.: Virtual Heart: Cardiac Simulation for Surgical Training & Education. Workshop & Conference on Virtual Reality and Virtual Environments (2007)
5. Sachse, F. B., Blmcke, L. G.,Mohr, M., Glnzel, K., Hfner, J., Riedel, C., Seemann, G., Skipa, O., Werner, C. D., Dssel, O.: Comparison of macroscopic models of excitation and force propagation in the heart. Biomed Tech (Berl), vol 47, 217–220 (2002)
6. Amorim, R. M., Campos, R. S., Lobosco, M., Jacob, C., Santos R. W. An Electro-Mechanical Cardiac Simulator Based on Cellular Automata and Mass-Spring Models Lecture Notes in Computer Science(Cellular Automata), 434–443 (2012)
7. Bora, C. and Serinagaoglu, Y. and Tonuk, E. Electromechanical heart tissue model using cellular automaton Biomedical Engineering Meeting (BIYOMUT), 2010 15th National,1–4, 2010
8. Bora, C. Cellular Automaton Based Electromechanical Model Of The Heart (Master Thesis), Middle East Technical University, 2010
9. Mollemans, W., Schutyser, F., Van Cleynenbreugel, J., Suetens, P. Tetrahedral mass spring model for fast soft tissue deformation Lecture Notes in Computer Science (IS4TM) 2673, 145–154, 2003
10. de Oliveira, B. L., Rocha, B. M., Barra, L. P. S., Toledo, E. M., Sundnes, J. and Weber dos Santos, R. Effects of deformation on transmural dispersion of repolarization using in silico models of human left ventricular wedge Int. J. Numer. Meth. Biomed. Engng.. doi: 10.1002/cnm.2570, 2013