Towards Real-Time Cardiac Electrophysiology Computations Using GP-GPU Lattice-Boltzmann Method

Bogdan Georgescu, Saikiran Rapaka, Tommaso Mansi, Oliver Zettinig, Ali Kamen, and Dorin Comaniciu

Siemens Corporation, Corporate Technology, Imaging and Computer Vision, Princeton, NJ, USA

Abstract. With recent advances in numerical methods and experimental validation, cardiac electrophysiology models can become surrogate tools for improved diagnostics and therapy planning. However, day-to-day clinical applications require models that are accurate and detailed enough to capture the main pathological patterns, but at the same time fast, with near real-time computation time. In particular, the models should be computed in a reasonable amount of time to enable personalization and on-line therapy guidance. Towards this goal, we present in this manuscript a novel algorithm adapted to graphics processing units (GPU) that enables near real-time cardiac electrophysiology computation with state-of-the-art cellular models. Our method relies on LBM-EP, a Lattice-Boltzmann method, which is naturally scalable to massively parallel architectures. Tested on a synthetic case and on a patient geometry, our experiments demonstrate the high scalability of the algorithm, reaching 10× speed up with respect to the CPU implementation of the algorithm.

1 Introduction

In the last decade, intense efforts have been put to apply complex computational models of cardiac electrophysiology to clinical problems. As models mature (see [1] for a comprehensive review), their application for planning and guidance of electrophysiology therapies is being investigated. For instance, in [4, 7], models were employed to plan radio-frequency ablation of ventricular tachycardia, whereas in [6] cardiac resynchronization therapy was investigated, all studies showing promising results.

A crucial requirement for clinical use is accuracy and computational efficiency. Because the models are applied to complex pathologies, simplified methods like those based on the Eikonal equation may not be suited. Phenomenological or even ionic models would be required [1]. Yet, the later models are often computationally demanding, in particular when solved using the finite element method. Recently, implementations on massively parallel architectures, like graphics processing units (GPU) have been investigated. In [8], the authors parallelized a finite element solver and implemented it on GPU, achieving a speed-up

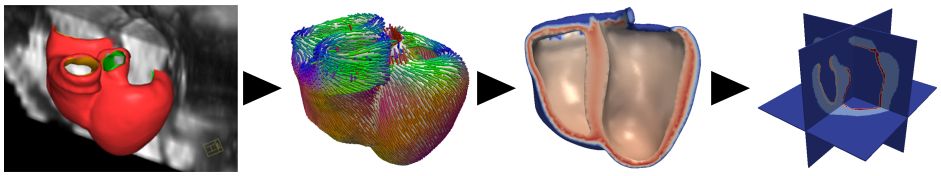

Fig. 1: Workflow for building the computational domain. From left to right: Automatic segmentation of the bi-ventricular myocardium, Rule-based model of fiber architecture, Rule-based model of cellular parameter heterogeneity (here τ_{close}), Rasterization to Cartesian domain

factor of 9 to 17 times with respect to a central processing unit (CPU) implementation. However, the overall computation time was still 160 times slower than real-time. Finite element methods have also been optimized in [7]. There, the system is solved using an explicit scheme, especially optimized for GPU processing, yielding a computation time of 3-7s. However, it is not clear how scalable is the approach when the number of tetrahedra increases and complex phenomena arise, as a smaller time step and conjugate gradient tolerance would be required to achieve accuracy.

Inspired from computational fluid dynamics approaches, Rapaka et al. [5] proposed a different solver to compute cardiac electrophysiology mono-domain models based on the Lattice-Boltzmann method, called LBM-EP. The key features of LBM-EP are: 1) the equations are solved on a Cartesian grid, which is of particular interest when dealing with patient-specific problems where the geometry is obtained from imaging data; 2) the algorithm is node-wise, and therefore suited for parallel architectures without significant modifications to the solver; 3) the method is accurate: results are invariant with respect to the preferred diffusion direction and, by use of a level-set representation of domain boundaries, is second-order accurate in space. In this manuscript, we investigate and quantify the scalability of LBM-EP. We propose an implementation that is suitable for both multi-core and GPU architectures (Sec. 2). The method is evaluated on a synthetic setup as well as on one patient data (Sec. 3). . In particular, we demonstrate that GPU implementation of LBM-EP can reach near real-time computation, with a computation time of about 3s/beat. Sec. 4 concludes the manuscript.

2 Methods

2.1 Computational Domain Preparation from Medical Images

The first step of the proposed method consists in building the computational domain from patient's images. LBM-EP being solved on Cartesian grids (Sec. 2.3), its application to clinical images is relatively immediate. Starting from a cardiac image (e.g. from magnetic resonance imaging (MRI)), the left endocardium, right endocardium and epicardium are automatically segmented using a machine

learning approach [11]. The resulting surface meshes are fused in one surface representing the myocardium while preserving their anatomical label. A level-set representation of that surface is then computed on an isotropic Cartesian grid whose resolution is controlled by the user. Based on the labels, grid nodes lying at the heart endocardia are marked for fast conductivity, to mimic the Purkinje network. The nodes lying at the septal endocardia are identified as pacing sites, to capture the effects of the His bundle. Tissue anisotropy is taken into account through a model of fiber architecture, computed from the patient-specific anatomy by linearly interpolating fiber elevation angle, defined as the angle with respect to the short axis plane, from the epicardium to the endocardium ($-60°$ to $+60°$ for the left ventricle, $-80°$ to $+80°$ for the right ventricle [1]). If scars are present, they can be reported in the domain through level set. Finally, spatial heterogeneity of the action potential duration is modeled based on literature reports, including endo-, mid- and epi-cells, as illustrated in Fig. (1).

2.2 Lattice-Boltzmann Model of Cardiac Electrophysiology

We concern ourselves with general mono-domain equations of the form:

$$\chi \left(C_m \frac{\partial v}{\partial t} + I_{ion} \right) = \nabla \cdot (\mathbf{D}(\mathbf{f}) \nabla v) + I_{stim} \tag{1}$$

where, χ is the surface-to-volume ratio, C_m is the capacitance of the tissue, I_{ion} is the sum of ionic currents from the cell model, $\mathbf{D}$ is the conductivity of the tissue which depends on the local fiber direction vector $\mathbf{f}$, v is the transmembrane potential and I_{stim} is an applied stimulus current. The ionic currents I_{ion} are usually modeled using systems of ordinary differential equations for a set of gating variables. In this work, the Mitchell-Schaeffer model (MS) [3] is employed, which is a simplified cell-model using only a single gating variable, h. Another key advantage of the MS model is the fact that it is characterized by only 4 parameters, each of which is directly related to the shape of the action potential, enabling ease of personalization for patient-specific computations. It should be noted though that the proposed method is not limited to the MS model, any cellular model could be used.

Using the MS model, the general equation Eq. (1) becomes:

$$\frac{\partial v}{\partial t} = \nabla \cdot \mathbf{D}(\mathbf{f}) \nabla v + I_{in} + I_{out} + I_{stim} \tag{2}$$

$$I_{in} = h v^2 (1 - v) / \tau_{in} \tag{3}$$

$$I_{out} = -v / \tau_{out} \tag{4}$$

$$\frac{dh}{dt} = \begin{cases} (1 - h)/\tau_{open}, & \text{if } v < v_{gate} \\ -h/\tau_{close}, & \text{otherwise} \end{cases} \tag{5}$$

Eq. (2-5) are solved on the Cartesian grid computed from the images (Sec. 2.1) according to a 7-connectivity topology (6 connections + central position). Eq. (5)

is solved at every node of the grid using a forward Euler scheme while Eq. (2) is solved using the Lattice-Boltzmann method (LBM) under Neumann boundary conditions (isolated heart). The fundamental variable of LBM is the vector of distribution functions $\mathbf{f}(\mathbf{x}) = \{f_i(\mathbf{x})\}_{i=1...7}$, where $f_i(\mathbf{x})$ represents the probability of finding a particle travelling along the edge $\mathbf{e}_i$ of node $\mathbf{x}$. The governing equation at $\mathbf{x}$ for the edge $\mathbf{e}_i$ is composed of two successive steps:

$$f_i = f_i - A_{ij}\,(f_j - \omega_j v) + \delta t \omega_i (J_{in} + J_{out} + J_{stim}), \tag{6}$$

$$f_i(\mathbf{x} + \mathbf{e}_i, t + \delta t) = f_i(\mathbf{x}, t) \tag{7}$$

where, the collision matrix $\mathrm{A} = (A_{ij})_{i,j\in[\![1,7]\!]}$ relaxes the distribution function f_i towards the local value of the potential, v, and ω_i is a weighting factor that depends on lattice connectivity, here $\omega_i = 1/8$ for the edges to the six n eighbors and $\omega_i = 1/4$ for the central position. The transmembrane potential is related to the f_i's through $v(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t)$. In each computational time step δt, a simple and strictly local collision rule is applied to the distribution functions at each node (Eq. (6)). Post-collision, the distribution functions stream along their corresponding edges to the neighboring nodes (Eq. (7)). In the simplest form, the collision matrix A relaxes each component towards the local potential by a characteristic relaxation time τ, $\mathrm{A} = (1/\tau)\mathrm{I}$, where I is the 7×7 identity matrix. With $\mathbf{D} = (2\tau - 1)/8\mathbf{Id}$, it has been shown that this algorithm solves the isotropic reaction diffusion Eq. (2) [2]. The model is extended to the anisotropic diffusion by using the collision matrix $\mathrm{A} = \mathrm{M}^{-1}\mathrm{SM}$ [9], with

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & -6 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & -2 & -2 & 0 \end{pmatrix} \quad S^{-1} = \begin{pmatrix} \tau_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau_{11} & \tau_{12} & \tau_{13} & 0 & 0 & 0 \\ 0 & \tau_{21} & \tau_{22} & \tau_{23} & 0 & 0 & 0 \\ 0 & \tau_{31} & \tau_{32} & \tau_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \tau_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \tau_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_7 \end{pmatrix}$$

M can be interpreted as follows. The first row corresponds to $v = \sum_i f_i$, rows 2 to 4 compute the gradient of the potential and rows 5 to 7 are higher-order quantities which do not affect the diffusion problem but rather control the stability of the algorithm. The relaxation times $(\tau_{ij})_{i,j\in[\![1,3]\!]}$ are related to the components of the diffusion tensor $\mathbf{D}$: $\tau_{ij} = \delta_{ij}/2 + 4D_{ij}\delta t/\delta x^2$, while $\tau_1 = 1$ is related to the potential and τ_5, τ_6 and τ_7 do not directly effect the diffusion solution, but effect the stability of the method (here $\tau_5 = \tau_6 = \tau_7 = 1.33$).

Neumann boundary conditions are implemented by adding at the boundaries an additional incoming distribution function that is equivalent to the outgoing one. More precisely, it can be shown that the Neumann boundary condition for potential on a surface writes $\sum_i f_i \mathbf{e}_i \cdot \mathbf{n} = 0$. This simple relation naturally yields the incoming potential distribution to add in the case where the boundary surface is parallel to the grid. In more complex geometries, a level-set representation of the boundary is used to compute the incoming potential using linear interpolation [10], thus enabling simulations in complex domain without requir-

ing advanced meshing algorithms. The main steps of the solver are reported in Algo. 1.

Algorithm 1 LBM-EP: Lattice-Boltzmann Model of Cardiac Electrophysiology

Require: Cartesian grid, level-set domain boundaries, δt, t_{stop}, model parameters

1: **while** $t < t_{stop}$ **do**
2: $t \leftarrow t + \delta t$
3: **for** all node **x do**
4: $\forall i$, collide: $f_i(\mathbf{x})$ (Eq. 6)
5: Update $h(\mathbf{x})$ (Eq. 5)
6: **for** every node **x do**
7: $\forall i$, stream and apply boundary conditions: $f_i(\mathbf{x})$ (Eq. 7)
8: **return** $v = \sum_i f_i$, h.

2.3 GPU Implementation of LBM-EP

Since the volume occupied by the myocardium is only a small fraction of the total domain volume, the computational method and data structures have been specifically designed to exploit this sparsity. More precisely, the nodes belonging to the domain are encoded by a flat array. The mapping from the 3D space to the data-structure is ensured by an array of pairs of indices. Another array is used to get, for each node, the index of its six neighbors. The algorithm is then adapted as follows. The collide and stream computations are implemented in one unique kernel to maximize the amount of computation per byte of data transferred from global memory. To avoid racing condition, a temporary, swap array is locally used by each kernel to copy the results of the collide step, which are then processed by the stream step. Finally, data communication between CPU and GPU are minimized to their maximum: input data are transfered prior to the computation. The simulation is performed entirely on the GPU card. Results are then transfered to the CPU memory at the very end. Multi-core CPU implementation is straightforwardly obtained by using OpenMP routines.

3 Experiments and Results

To test the ability to model anisotropic tissue conductivity, we stimulated a square sample of tissue 30mm×30mm in the center. The longitudinal conductivity was set to 100mm^2/s and the transverse conductivity was set to 20 mm^2/s. The computation was run up to $t = 0.03$ s. The domain was discretized at resolutions of 0.6mm, 0.3mm, 0.15mm and 0.075mm. This test, even though simple, is a good indicator of anisotropic modeling capabilities since other Cartesian-grid based Finite-Difference Methods have shown some difficulty in modeling the eccentricity exactly([1]). The finite-difference results for a similar problem show the largest errors when the fiber direction is diagonal, which is the chosen

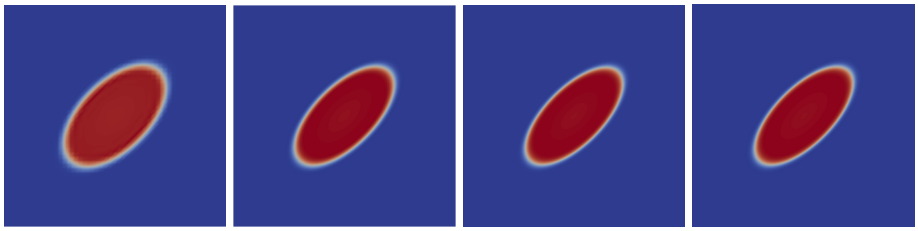

Fig. 2: Anisotropic conduction in a 30mm×30mm tissue with fiber direction along the diagonal direction. The resolutions are, from left to right, 0.6mm, 0.3mm, 0.15mm and 0.075mm.

orientation for our problem. The results of the computation are shown in Fig. (2).

All experiments were executed on a standard Windows desktop machine (Intel Xeon, 2.66GHz octo-core, 4GB RAM). LBM-EP was tested for both CPU (with OpenMP parallelization) and GPU implementation using a NVidia GTX 680 graphics card. A semi-implicit, anisotropic finite element implementation of Mitchell-Schaeffer model, called FEM-EP, was used for comparisons. FEM-EP was based on linear tetrahedra and parallel optimization (OpenMP). The run-times, in seconds, for the two implementations at a resolution of 0.075mm are shown in Table 1. It can be seen that the OpenMP implementation of LBM-EP achieves a factor of 6x speedup with 8 computational cores and an additional factor of 12x speedup for the GPU implementation using single precision arithmetic.

FEM-EP	**1 core**	**4 cores**	**8 cores**	**GPU (double)**	**GPU (float)**
142	117	30.8	21.4	3.11	1.67

Table 1: The run-times (in seconds) for a reference FEM-EP implementation (first column) and LBM-EP with different configurations.

We now consider the application of LBM-EP to real patient geometries. The algorithm was applied to compute the action-potential propagation in a patient during sinus rhythm. Starting from medical images, a model of the anatomy was extracted and the different anatomical zones identified automatically as shown in Fig.(1). A level-set representation was then produced on a Cartesian grid, using which the lattice-Boltzmann computations were performed. To observe how the computation time scales with the number of nodes, the geometry was discretized at resolutions of 2.0mm, 1.5mm, 1.2mm, 1.0mm, 0.9mm, 0.8mm and 0.7mm.

The resulting contours of the depolarization time and the average computational time for 1 heart cycle are shown in Fig. (3). It is seen that with modern graphical processing units and an algorithm which can exploit them efficiently, an entire heart cycle can be modeled (at 0.7mm resolution) in 15 secs/beat. This resolution is already finer than the resolution at which the medical images are typically acquired (1.5mm). At this resolution, the computation time

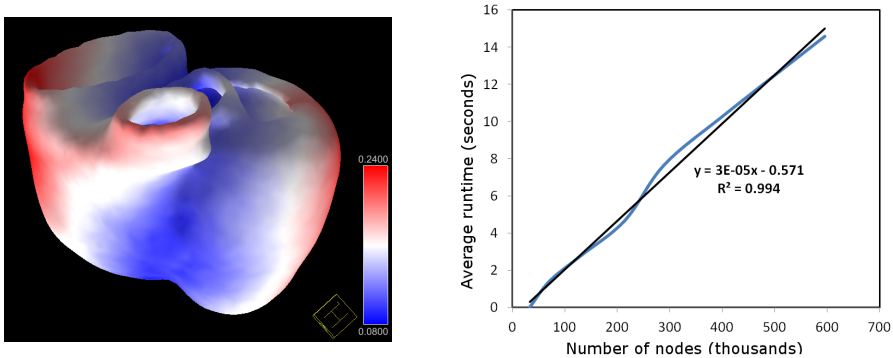

Fig. 3: Left: Contours of depolarization time computed using LBM-EP on a real patient geometry, Right: the dependence of the average computational time for 1 heart cycle on the number of computational nodes (in thousands)

reduces to only 3 secs/beat, making the model nearly real-time and applicable to patient-specific therapy planning and guidance.

4 Conclusions

In this paper, we have presented a novel algorithm for fast and accurate patient-specific simulations of cardiac electrophysiology utilizing the power of modern graphical processing units. The approach relies on Lattice-Boltzmann method for integrating any general monodomain electrophysiological cell models. Unlike other Cartesian-grid based methods, LBM-EP is able to model anisotropic conductivity in the tissue without losing accuracy. Using a level-set based formulation for representing the complex geometry, we have shown the ability to model real patient geometries without numerical artifacts. For the simplified Mitchell-Schaeffer cell model, we have achieved a remarkable computational time of 16 secs/beat at 0.7mm resolution and 3 secs/beat at 1.5mm, making the model nearly real-time. This opens many possibilities for patient-specific therapy planning and real-time guidance, which will be explored in the future.

References

1. Clayton, R., Bernus, O., Cherry, E., Dierckx, H., Fenton, F., Mirabella, L., Panfilov, A., Sachse, F., Seemann, G., Zhang, H.: Models of cardiac tissue electrophysiology: Progress, challenges and open questions. PBMB 104(1-3), 22 (2011)
2. Dawson, S.P., Chen, S., Doolen, G.D.: Lattice boltzmann computations for reaction-diffusion equations. J. Chemical Physics 98(2), 1514–1523 (1993)
3. Mitchell, C., Schaeffer, D.: A two-current model for the dynamics of cardiac membrane. Bulletin of mathematical biology 65(5), 767–793 (2003)

4. Ng, J., Jacobson, J.T., Ng, J.K., Gordon, D., Lee, D.C., Carr, J.C., Goldberger, J.J.: Virtual electrophysiological study in a 3-dimensional cardiac magnetic resonance imaging model of porcine myocardial infarction. Journal of the American College of Cardiology 60(5), 423–430 (2012)
5. Rapaka, S., Mansi, T., Georgescu, B., Pop, M., Wright, G., Kamen, A., Comaniciu, D.: Lbm-ep: lattice-boltzmann method for fast cardiac electrophysiology simulation from 3d images. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012, pp. 33–40. Springer (2012)
6. Sermesant, M., Chabiniok, R., Chinchapatnam, P., Mansi, T., Billet, F., Moireau, P., Peyrat, J.M., Wong, K., Relan, J., Rhode, K., et al.: Patient-specific electromechanical models of the heart for the prediction of pacing acute effects in crt: a preliminary clinical validation. Medical image analysis 16(1), 201–215 (2012)
7. Talbot, H., Marchesseau, S., Duriez, C., Sermesant, M., Cotin, S., Delingette, H.: Towards an interactive electromechanical model of the heart. Interface Focus 3(2) (2013)
8. Vigmond, E.J., Boyle, P.M., Leon, L.J., Plank, G.: Near-real-time simulations of biolelectric activity in small mammalian hearts using graphical processing units. In: Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE. pp. 3290–3293. IEEE (2009)
9. Yoshida, H., Nagaoka, M.: Multiple-relaxation-time lattice boltzmann model for the convection and anisotropic diffusion equation. J. Comput. Phys. 229, 7774–7795 (October 2010)
10. Yu, D., Mei, R., Luo, L., Shyy, W.: Viscous flow computations with the method of lattice boltzmann equation. Progress in Aerospace Sciences 39(5), 329 – 367 (2003)
11. Zheng, Y., Barbu, A., Georgescu, B., Scheuering, M., Comaniciu, D.: Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. IEEE TMI 27(11), 1668–1681 (2008)