Procedia Computer Science

Procedia Computer Science 101, 2016, Pages 68 – 75

YSC 2016. 5th International Young Scientist Conference on Computational Science

Parallel Simulation of Scroll Wave Dynamics in the Human Heart Using the FEniCS Framework

Timofei Epanchintsev[1,2,3], Sergei Pravdin[1,3], Andrey Sozykin[1,2,3], and Vladimir Zverev[3,1]

[1] Krasovskii Institute of Mathematics and Mechanics, Ekaterinburg, Russia
[2] Institute of Immunology and Physiology UrB RAS, Ekaterinburg, Russia
[3] Ural Federal University, Ekaterinburg, Russia

Abstract
In the heart, scroll waves are three-dimensional self-sustaining spiral waves of electrical excitation. They arise only in pathology and underlie dangerous arrhythmias. Computer simulation of human cardiac arrhythmias is time-consuming and requires parallel computing. In addition, heart simulation is a complicated multilevel task (cell, tissue, organ); parallel multilevel simulation codes are difficult to create and maintain. Here we present an approach to parallel simulation of scroll waves in the left ventricle of the human heart utilizing the FEniCS framework, which allows developing software using near-mathematical notation and provides automatic parallelization. We used the *Aliev–Panfilov* cell model of heart electrical activity and studied the scalability of FEniCS implementation on parallel computing systems.

Keywords: heart simulation, finite element method, scalability, FEniCS, parallel computing, scroll wave

1 Introduction

Waves in media have been an object of investigation for many years. Some of physical and chemical media, biological tissues and populations are excitable and provide examples of waves of special kind, the autowaves. One such form are spiral waves. They appear, for instance, in chemical (Belousov–Zhabotinsky) reactions and biological tissues (retina, myocardium). Spiral waves in the heart are of a particular interest because they arise only in pathology and underlie dangerous arrhythmias. Three-dimensional spiral waves are called *scroll waves*. Investigation of scroll waves in the heart *in vivo* and *in vitro* is associated with technical and ethical complications and prohibitions, so they are often studied *in silico*. Computer simulation of cardiac arrhythmias is most important and also time-consuming if one studies the human ventricles since they are bigger and require more mesh nodes than human atria or the heart of small-sized laboratory animals. Parallel computing is a way to accelerate such simulation by means of the modern hardware and software.

Peer-review under responsibility of organizing committee of the scientific committee of the 5th International Young Scientist Conference on Computational Science
© 2016 The Authors. Published by Elsevier B.V.

doi:10.1016/j.procs.2016.11.010

Moreover, heart simulation is a complicated multilevel task (cell, tissue, organ) that requires strong knowledge of each layer. Multilevel simulation codes are difficult to maintain manually because changes in one layer may lead to changes in others. To overcome these obstacles, automated scientific computing frameworks were created that compensate for the layer dependencies. In addition, some frameworks provide automatic parallel simulation.

We use the FEniCS framework [3] as an automated scientific tool. It provides domain–specific language with near–mathematical notation and automatic solution of the partial differential equations using the finite element method on MPI clusters. Computations are performed by highly optimized algebra backends such as PETSc or Hypre.

Previously we have implemented *ten Tusscher–Panfilov* [17] and *Ekaterinburg–Oxford* [16] cell models using FEniCS. The implementation showed good performance and scalability on hundreds of CPU cores.

In a continuation of our work on the parallel cardiac activity simulation using the FEniCS framework, we present herein an effort to use FEniCS for investigation of scroll wave dynamics in the left ventricle of a human heart. We used the *Aliev–Panfilov* cell model [1] of heart electrical activity. The scalability of FEniCS implementation was studied.

2 Physiological Background

2.1 Scroll Waves and Arrhythmia

The heart is a muscular organ which mechanical activity, pumping blood, is orchestrated by electrical processes. There are four cardiac chambers, two atria and two ventricles. The left and right atria receive blood from the veins. Under normal circumstances, the atrial myocardium is activated when the ventricular muscle is not, and then the blood is ejected into the ventricles. After that, the activation reaches the ventricular myocardium, and the ventricles pump blood to the arteries. Finalizing the cardiac cycle, all the myocardium takes a pause and rest to become ready for the next activation. In pathology, this rhythmical process is disrupted, that leads to some disturbance of the mechanical activity of the heart. There is a class of arrhythmias that are characterized by the appearance of special pattern waves, namely spiral and, in 3-dimensional space, scroll waves. Such waves can occupy all the ventricular myocardium and fully silence the normal waves originating from the sinus node. On the isotropic homogeneous flat surface, the spiral waves have the shape of the Archimedean spiral. In the anisotropic heterogeneous spatial bodies, they can make much more complicated patterns but all the patterns have the same peculiarity: at any moment, there is an activated and an unactivated part of the medium. This leads to a fall of cardiac performance culminating in a sudden cardiac event, which may cause death and requires immediate treatment.

2.2 Description of the Heart Model

The *Aliev–Panfilov* model [1] was used to simulate the electrical activity of the heart. This model contains a description of the working myocardium of the human ventricles. As a mathematical expression, the electrical and chemical processes are presented as a reaction-diffusion system by one partial and one first-order ordinary differential equations:

$$\frac{\partial V}{\partial t} = \nabla \cdot (D \nabla V) + f(V, S), \tag{1}$$

$$\frac{dS}{dt} = g(V, S), \tag{2}$$

where state variables are V, the action potential, and S, representing the K current conductance of the cell membrane. The scalar-valued functions f and g describe the time evolution of each variable. D is the diffusion matrix 3×3 defining anisotropy of the tissue, and ∇ is the traditional Del operator. System (1),(2) is defined at each point of the heart tissue, and, thus, we should solve it for each node of the computational mesh.

We used the standard "no-flux" boundary conditions providing an electrical isolation of the ventricle.

The *Aliev–Panfilov* cell model was chosen as the first step of research. The model is simple but it allows to investigate waves of excitation in the left ventricle and consequently to estimate scroll wave simulation scalability.

3 The FEniCS Framework

FEniCS is an implementation of the *finite element method* (FEM) that provides automated solution of partial differential equations (PDE). This automation relies on three key steps [3]: automation of discretization, discrete solution and error control.

The framework is able to solve PDE by two approaches: direct and iterative. Direct solvers cope with nonlinear equations, but they use essential memory space and are difficult to parallelize. Iterative solvers require equations in linear form only, but they use less memory and are easier to parallelize.

The FEniCS software subprojects are devoted to the formulation of variational forms (UFL [2]), the discretization of variational forms (FIAT [5], FFC [8]), and the assembly of the corresponding discrete operators (UFC, DOLFIN [9]). To solve the defined problem, FEniCS uses various algebra backends such as PETSc and Hypre.

UFL is a domain specific language designed for convenient and understandable formulation of variational forms. It can be considered as an interface for mathematicians.

Variational forms are discretized by the generation of arbitrary–order instances of the Lagrange elements on lines, triangles, and tetrahedra (FIAT) and a compilation of efficient low-level C++ code that can be used to assemble the corresponding discrete operator (FFC). FEniCS already has built-in optimizations for these steps, which are launched once at the beginning of a solution and take constant time for the generation regardless size of the input data.

The assembly of the discrete operators is related only to iterative methods. It is crucial for acceleration on parallel computing systems. The idea is to split the mesh among processing units, compute the local matrix and insert values back into the global matrix. The FEniCS team has designed a local-to-global mapping algorithm [6] to map values between local and global matrices.

4 Implementation

In order to implement the *Aliev–Panfilov* model in FEniCS, we transformed the nonlinear system (1),(2) into a linear one, which let us use iterative solvers. The transformation was performed according to the first order operator splitting scheme (the Marchuk–Yanenko method) [7].

The implementation of the *Aliev–Panfilov* model was performed using the UFL. The code fragment for the ODE problem formulation is presented in Listing 1. First, a finite element mesh is created and loaded from the file. After that the vector function space for cell variables is defined. Then cell governing equations are defined using specified coefficients. FEniCS uses the term *trial function* to specify the unknown function that should be approximated (the variable *ode_vars* contains a trial function and the *ode_vars0* variable contains the initial values). The next step is to define the linear variational problem for the equations. Lastly, the ODE solver is created.

Listing 1: Formulation of the ODE variational problem

```
mesh = Mesh()
# Code for loading mesh from the file
# Building vector function space for cell variables
Space_ode_var = VectorFunctionSpace(mesh,
        "Lagrange", 1, number_ode_vars)
ode_vars = TrialFunction(Space_ode_var)
ode_vars0 = Function(Space_ode_var)
# Initialization of equations
ode_eq = ode_rhs(beta_t, beta_phi, delta_phi, ode_vars0)
# Definition of the ODE problem
Dphase_Dt = (ode_vars - ode_vars0)/dt
OdePart = (inner(Dphase_Dt, q2) - inner(ode_eq,q2))*dx(mesh)
ODEproblem = LinearVariationalProblem(lhs(OdePart),
        rhs(OdePart), ode_vars, bcs=[])
# Creating ODE solver
ODEsolver = LinearVariationalSolver(ODEproblem)
```

Parallel execution is automatically provided by FEniCS and its components. Firstly, the mesh is distributed among the MPI-processes. Secondly, FEniCS creates functional spaces over these parts. Finally third, local matrices are computed using the high performance linear algebra backends that support parallel solution by default.

It is well-known that myocardium is highly anisotropic; it has one direction of high speed signal propagation (along the fibres). In all directions orthogonal to that, the speed is less in approximately 3-4 times. Taking the anisotropic structure into account requires twice the simulation time than the isotropic structure due to more complex diffusion equation.

We used protocol S1S2 [12] to initiate a scroll wave. This protocol involves stimulation of two zones of the myocardium at two different moments. First, we make an S1 stimulus of a small region to get an ordinary (not scroll) wave. Such a wave has a sharp anterior front and a blurry posterior front. The second stimulus, S2, is given on a region crossing the posterior front. A scroll wave appears near the intersecting point of the posterior front and the S2 boundary. Such a wave can exist for an indefinitely long period of time unless disappears due to approaching the basal margin of the left ventricle. We simulated the scroll wave for 30 seconds of the model time.

5 Performance Evaluation

In order to evaluate the performance and scalability of the proposed implementation, we carried out a series of experiments. We simulated the spiral wave in the human heart left ventricle using the asymmetric anatomical model that have been previously developed by our group [15] (an example of the spiral wave is presented in Fig. 1). We used the tetrahedral mesh with the length of the tetrahedrons from 2 to 4 mm; the mesh contained 7178 points and 26156 tetrahedra.

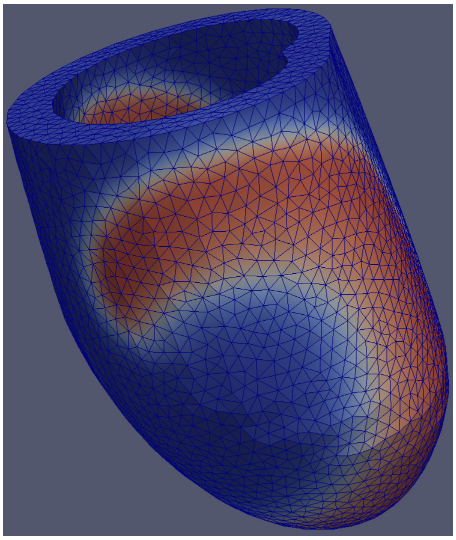

Figure 1: An example of spiral wave in the left ventricle

Table 1: Configuration of computational nodes

Configuration parameter	**Value**
CPU	2 x Intel(R) Xeon(R) CPU X5675 @ 3.07GHz
RAM	48 GB
Interconnect	Infiniband DDR (20 Gbit/s)
Operating System	CentOS 7

An integration time step of 0.05 time units was chosen (1 time unit of Aliev–Panfilov model is equal to 12.9 ms). The linear system of size 21588×21588 was solved in each time step.

The experiments were carried out on the *Uran* supercomputer of the Krasovskii Institute of Mathematics and Mechanics (the configuration is presented in Table 1). FEniCS version 1.6.0 was used.

The spiral wave simulation was executed in parallel mode using various numbers of CPU cores, from 1 to 240. We used the *conjugate gradient method* combined with a *successive over-relaxation* preconditioner for solving linear systems. The simulation time depending on the number of CPU cores is presented in Fig. 2, the achieved speedup is demonstrated in Fig. 3.

6 Discussion

During the experiments, the simulation time was decreased twelve-fold: from 1254 minutes on 1 CPU core to 105 minutes on 48 cores. However, the scalability of FEniCS-based scroll wave simulation software is limited. Increasing the number of CPU cores beyond 48 did not improve performance due to communication overhead. The problem has a fixed size, and the predefined mesh is automatically distributed over the MPI-processes before the start of computation. Increasing the number of processes leads to decrease in data size of each process. When using more than 48 CPU cores, each process spends more time on communication than on useful computation. Although the problem size can be increased by the mesh refinement, it has no

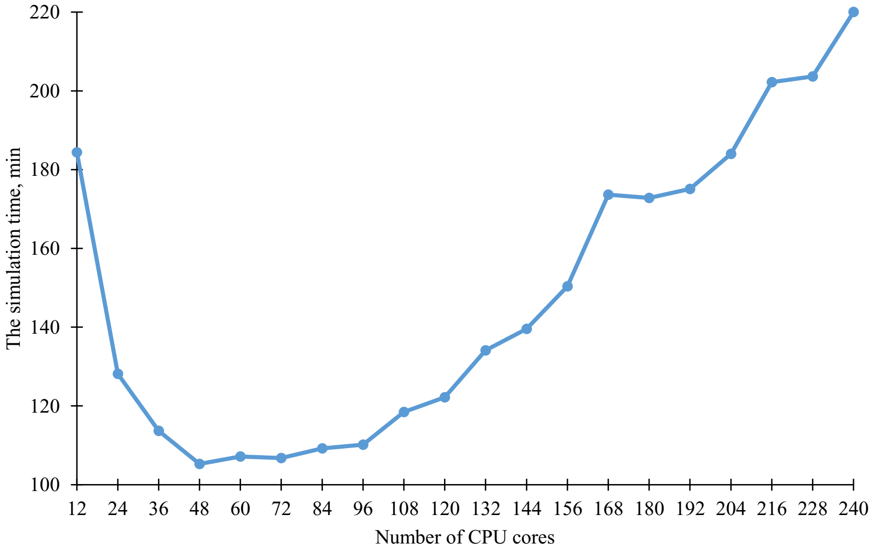

Figure 2: The simulation time depending on the number of CPU cores

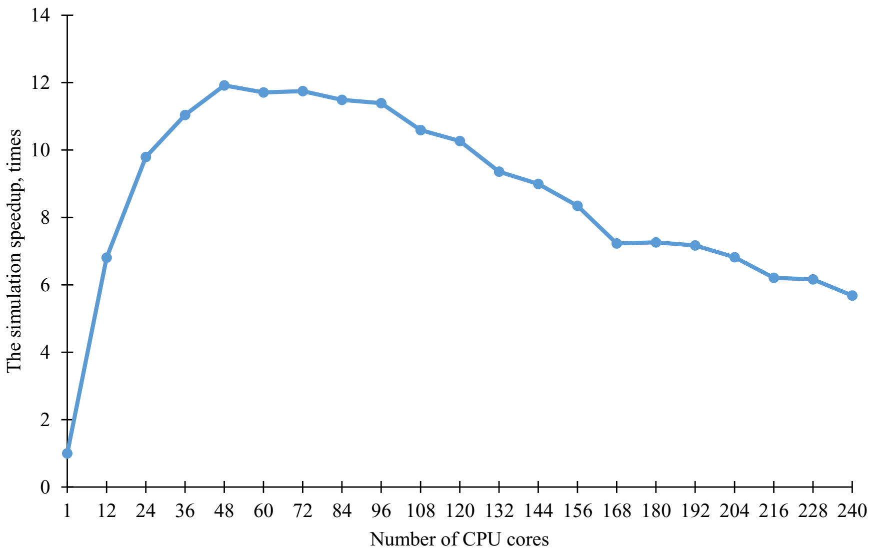

Figure 3: Simulation speedup depending on the number of CPU cores

biophysical significance in case of the *Aliev–Panfilov* model.

The limited scalability is caused by the simplicity of the *Aliev–Panfilov* model, which includes only two state variables. Other models of heart electrical activity are significantly more robust and complex. For example, the *ten Tusscher–Panfilov* model [17] includes 18 state variables and the *Ekaterinburg–Oxford* model [16] incorporates 30 state variables. In future, we plan to use these models for scroll waves investigation. Since the models are more computationally intensive, we expect an improvement of scalability.

An advantage of the FEniCS-based implementation is the utilization of near-mathematical notation. As a result, the software can be used and modified by biophysicists, computational mathematicians, and other researchers without strong computer science skills. They can easily change the model parameters, implement another scroll wave initiation protocol, or even replace the entire cardiac electrical activity model. In addition, they do not have to deal with parallel programming, because the FEniCS framework provides the automatic parallelization.

7 Related Work

Chaste, which is another automated scientific computing framework, was used for spiral wave simulation [11]. A stable spiral wave was generated on a two-dimensional mesh (3 cm $\times$ 3 cm), by an appropriate stimulation and use of a modified *Luo–Rudy* action-potential model [10]. The Chaste-based implementation showed a good scaling up to 2048 cores. The Chaste framework uses almost the same backend as FEniCS: METIS/parMETIS, uBLAS, PETSc, and allows for the storage of data in HDF5 format. Chaste includes a various number of ready-to-use cell-models implementations. The role of a programmers is to define the mesh, set variables and choose an appropriate solver. But in order to implement models or solvers, a developer needs to deal with an internal Chaste structure that is not presented in near–mathematical notation.

Most researchers, who conducted spiral wave simulation, have focused on biophysical results and have not paid attention to parallel computing. In the paper [4], the spiral wave was simulated by manual implementation of an incremental iterative the Newton–Raphson scheme. Another approach uses the Fenton–Karma model for cardiac excitation [13] with the help of hybrid approach that combines an explicit Euler scheme for the reaction-diffusion system, with nonlinear finite element techniques for large deformation mechanics.

8 Conclusion

We presented a first step of scroll waves simulation in the left ventricle of the human heart on parallel computing systems using the FEniCS framework and the *Aliev–Panfilov* cardiac electrical activity model. The twelve times speedup has been achieved on 48 CPU cores. However, the scalability of the system is limited, as using more than 48 CPU cores leads to performance degradation.

The main direction of the future research work is to use more accurate and computationally intensive cardiac activity models, such as *ten Tusscher–Panfilov* and *Ekaterinburg–Oxford*, for scroll waves simulation. Another important task is to find filaments [14] from obtained results and use it for investigations of spiral wave drift.

Acknowledgments

The work is supported by the RAS Presidium grant I.33P "Fundamental problems of mathematical modeling", project no. 0401-2015-0025 and by the Research Program of Ural Branch of RAS, project no. 15-7-1-26. Our study was performed using the *Uran* supercomputer of the Krasovskii Institute of Mathematics and Mechanics and the computational cluster of the Ural Federal University.

References

[1] Rubin R. Aliev and Alexander V Panfilov. A Simple Two-variable Model of Cardiac Excitation. *Chaos, Solitons and Fractals*, 7(3):293–301, 1996.

[2] Martin S. Alnæs. *UFL: a Finite Element Form Language*, chapter 17. Springer, 2012.

[3] Timothy J Barth and Dirk Roose. *Automated Solution of Differential Equations by the Finite Element Method - The FEniCS Book*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, 2012.

[4] Serdar Göktepe, Jonathan Wong, and Ellen Kuhl. Atrial and ventricular fibrillation: computational simulation of spiral waves in cardiac tissue. *Archive of Applied Mechanics*, 80(5):569–580, 2010.

[5] Robert C. Kirby. *FIAT: Numerical Construction of Finite Element Basis Functions*, chapter 13. Springer, 2012.

[6] Robert C. Kirby and Anders Logg. A compiler for variational forms. *ACM Transactions on Mathematical Software*, 32(3):417–444, Sept. 2006.

[7] YS Li and CP Chen. An efficient split-operator scheme for 2-D advection-diffusion simulations using finite elements and characteristics. *Applied Mathematical Modelling*, 13(4):248–253, 1989.

[8] Anders Logg, Kristian B. Ølgaard, Marie E. Rognes, and Garth N. Wells. *FFC: the FEniCS Form Compiler*, chapter 11. Springer, 2012.

[9] Anders Logg and Garth N. Wells. Dolfin: Automated finite element computing. *ACM Transactions on Mathematical Software*, 37(2), 2010.

[10] C. H. Luo and Y. Rudy. A model of the ventricular cardiac action potential. Depolarization, repolarization, and their interaction. *Circulation Research*, 68(6):1501–1526, Jun. 1991.

[11] Gary R. Mirams, Christopher J. Arthurs, Miguel O. Bernabeu, Rafel Bordas, Jonathan Cooper, Alberto Corrias, Yohan Davit, Sara-Jane Dunn, Alexander G. Fletcher, Daniel G. Harvey, Megan E. Marsh, James M. Osborne, Pras Pathmanathan, Joe Pitt-Francis, James Southern, Nejib Zemzemi, and David J. Gavaghan. Chaste: An open source C++ library for computational physiology and biology. *PLoS Comput Biol*, 9(3):1–8, 03 2013.

[12] Kazuo Nakazawa et al. *Computational Analysis and Visualization of Spiral Wave Reentry in a Virtual Heart Model*, pages 217–241. Springer Tokyo, 2000.

[13] A. V. Panfilov, R. H. Keldermann, and M. P. Nash. Drift and breakup of spiral waves in reaction-diffusion-mechanics systems. *Proceedings of the National Academy of Sciences*, 104(19):7922–7926, 2007.

[14] A.V. Panfilov and A.N. Rudenko. Two regimes of the scroll ring drift in the three-dimensional active media. *Physica D: Nonlinear Phenomena*, 28(1):215 – 218, 1987.

[15] Sergei Pravdin. Nonaxisymmetric mathematical model of the cardiac left ventricle anatomy. *Russian Journal of Biomechanics*, 17(4):75–94, 2013.

[16] Olga Solovyova, Nathalie Vikulova, Leonid B Katsnelson, Vladimir S Markhasin, PJ Noble, A Garny, Peter Kohl, and Denis Noble. Mechanical interaction of heterogeneous cardiac muscle segments in silico: effects on Ca^{2+} handling and action potential. *International Journal of Bifurcation and Chaos*, 13(12):3757–3782, 2003.

[17] Kirsten HWJ Ten Tusscher and Alexander V Panfilov. Alternans and spiral breakup in a human ventricular tissue model. *American Journal of Physiology-Heart and Circulatory Physiology*, 291(3):1088–1100, 2006.