

Использование библиотеки FEniCS для моделирования левого желудочка сердца

Т. И. Епанчинцев^{1,2,3}
eti@imm.uran.ru

В. С. Зверев^{1,2}
vladimir.zverev@urfu.ru

А. В. Созыкин^{1,2,3}
avs@imm.uran.ru

1 – УрФУ (Екатеринбург)

2 – ИММ УрО РАН (Екатеринбург)

3 – Институт иммунологии и физиологии (Екатеринбург)

Аннотация

Исследование функциональных характеристик миокарда является сложной задачей и требует математических моделей, описывающих процессы на разных уровнях (клетка-ткань-орган). Целью работы является определение возможности применения библиотек автоматизированных научных вычислений для моделирования распространения потенциала действия с помощью параллельных вычислительных систем на примере модели Алиева–Панфилова. Представлен обзор существующих программных комплексов (OpenFOAM, OpenCMISS, FEniCS, Chaste), а также результаты тестирования производительности и масштабируемости реализации на основе FEniCS.

1 Введение

Научиться предсказывать сокращение сердечной мышцы – важная задача, которая открывает возможности по моделированию действия лекарств на конкретно взятого человека, а следовательно, новые перспективы в диагностике и лечении заболеваний. В данной статье мы рассматриваем левый желудочек сердца, так как он является самой большой камерой сердца, изгоняющей кровь ко всем органам и тканям.

Миокард является очень сложной системой и требует описания биофизических процессов, которые происходят на разных масштабах [1]: на клеточном уровне, на совокупности клеток (уровень ткани) и на уровне органа. Трудностей с моделированием такого объекта много как с точки зрения физиологии, так и с точки зрения описания найденных закономерностей, выбора методов решения моделей и многого другого.

Моделирование даже нескольких тактов работы сердца – вычислительно сложная задача. Невозможно провести такие расчеты на персональном компьютере за приемлемое время, поэтому реализация методов решения должна быть ориентирована на выполнение в параллельном режиме на крупных вычислительных системах. Как правило, программный код для таких крупных проектов пишется в большой группе и продолжительное количество времени. Долгое время уходит на изучение современных вычислительных архитектур и технологий, а также на адаптацию математической модели.

В то же время, архитектуры вычислительных систем постоянно развиваются. Появляются новые, такие как Nvidia GPGPU или Intel Xeon Phi. Какие-то технологии устаревают и их использование становится неактуальным. В условиях нынешнего прогресса аппаратных и программных технологий, новые разработки живут относительно недолго: от трех до пяти лет. Возможна ситуация, когда разработка проекта

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: A.A. Makhnev, S.F. Pravdin (eds.): Proceedings of the 47th International Youth School-conference “Modern Problems in Mathematics and its Applications”, Yekaterinburg, Russia, 02-Feb-2016, published at <http://ceur-ws.org>

начиналась с учетом особенностей одной технологии, которая через три года успевает устареть, и появляется необходимость в переписывании значительной части кодовой базы проекта, но уже с принятием во внимание специфики новой, более совершенной технологии.

В данных условиях постоянно приходится адаптировать и оптимизировать программный код под развивающиеся технологии. Дальнейшей целью нашей работы является многоуровневое моделирование, что влечет за собой дополнительные сложности в оптимизации. Изменения какого-либо участка кода на нижнем уровне требуют изменений на вышестоящих уровнях. Чем больше оптимизаций, тем сильнее программный код отдаляется от математической нотации. «Читабельность» кода теряется для математика (и тем более для физиологов), не знакомого с внутренним строением современных вычислительных архитектур и техниками оптимизации. Программа становится сложной и запутанной, а дальнейшее развитие и сопровождение проекта – нетривиальным.

В связи с вышесказанным можно выделить следующие основные сложности, которые возникают в крупных научных проектах, требующих массивных вычислений:

- 1) сохранение близости программного кода к математическим формулировкам;
- 2) достижение высокой скорости параллельных вычислений.

Как правило, чем код ближе к математической нотации, тем хуже он оптимизирован и наоборот. Существует ли решение данных проблем одновременно? Мы задались этим вопросом в начале нашего проекта и изучили ряд систем, которые предлагают это решение: OpenFOAM, OpenCMISS, Chaste, FEniCS.

В данной статье проведен обзор данных систем, выбрана наиболее подходящая для выполнения проекта, описана модель левого желудочка сердца, которая используется для тестовых расчетов, сформулирована задача моделирования и представлены результаты тестирования на вычислительном кластере.

2 Рассчитываемая модель

Модель Алиева–Панфилова принадлежит к группе так называемых феномологических моделей. От других видов эту группу отделяет наличие притягательного свойства простоты реализации, но в то же время они неспособны воспроизвести количественно важные характеристики исследуемой системы. Тем не менее, феномологические модели возбудимых сред предоставляют хорошее качественное соответствие свойств объекта моделирования и полезны, например, при исследовании сердечной аритмии [2].

Система уравнений Алиева–Панфилова [3] учитывает свойство реституции (restitution property) миокардиальной ткани и может быть применена для моделирования как целого органа, так отдельно левого желудочка. Модель включает следующие уравнения:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x_i} d_{ij} \frac{\partial u}{\partial x_j} - ku(u - a)(u - 1) - uv, \quad (1)$$

$$\frac{\partial v}{\partial t} = \left(\epsilon_0 + \frac{\mu_1 v}{\mu_2 + u} \right) (-v - ku(u - a - 1)), \quad (2)$$

где u – безразмерный трансмембранный потенциал клетки, v – безразмерная проводимость для калиевого тока, d_{ij} – коэффициенты диффузии, а ϵ_0 , a , k , μ_1 , μ_2 – параметры модели клетки. Везде индексы i, j принимают значения 1, 2, 3. Первое слагаемое в правой части (1) отвечает за описание пространственного изменения потенциала. Другие слагаемые моделируют быстрые процессы изменения трансмембранного потенциала, в то время как уравнение (2) описывает медленные процессы.

Для целей тестирования вычислительных инструментов мы использовали симметричную модель формы левого желудочка сердечной мышцы [5]. В начальный момент времени активировалась субэндокардиальная зона. Другими словами, в зоне, прилегающей к внутренней поверхности желудочка, переменной u было задано максимальное значение, а v была равна нулю. Вне этой зоны значения u и v были нулевыми. Внешняя граница желудочка предполагалась электрически изолированной, другими словами в качестве граничного условия бралось условие нулевого потока потенциала действия u через границу.

Модель Алиева–Панфилова была выбрана нами ввиду того, что, несмотря на свою относительную простоту, она является комбинацией нелинейного уравнения в частных производных параболического типа и жестких обыкновенных дифференциальных уравнений и отражает основные математические особенности моделей, которые используются для исследования свойств миокарда.

3 Существующие программные средства моделирования свойств тел сложной формы

3.1 OpenFOAM

OpenFOAM может быть охарактеризована как платформа для исследования моделей турбулентных течений, транспортных уравнений и других типов моделей, связанных с перераспределением различных веществ в пространстве. В её основе лежит метод конечных объемов, который наиболее пригоден для решения уравнений в частных производных, описывающих явления переноса. OpenFOAM состоит из двух частей. Одна из них представляет собой набор классов, которые используются в реализации численных процедур. Другая часть – библиотека программ-решателей ("solvers"). Такое разделение предоставляет программную среду для моделирования с помощью высокоуровневого кода, который напрямую отражает интегрируемые уравнения [6]. В качестве примера приведем листинг кода на языке программирования C++ для решения уравнения теплопроводности $T'_t = \nabla(D_T \nabla T)$, записанного с помощью классов OpenFOAM:

```
solve(fvm::ddt(T) - fvm::laplacian(D_T,T));
```

3.2 OpenCMISS

OpenCMISS – это часть международного проекта Physiome [7]. Система предоставляет среду с открытым исходным кодом для решения биоинженерных задач [8]. OpenCMISS использует унифицированный язык описания клеточной модели CellML, который позволяет перевести математическую запись обыкновенных дифференциальных уравнений в машиночитаемую форму. Однако, проект OpenCMISS находится в стадии разработки и недостаточно хорошо документирован.

3.3 Chaste

Другим пакетом моделирования, преследующим цель решения многомасштабных моделей и способным использовать описание CellML, является Chaste [9, 10]. Библиотека позволяет пользователям проводить расчеты как клеточных моделей, так и моделей, описывающих орган в целом. При этом могут использоваться разные типы уравнений в частных производных, которые применяются для изучения распространения электрических сигналов [10]. На текущий момент, насколько нам известно, Chaste не имеет средств для моделирования механической функции сердца.

3.4 FEniCS

FEniCS – это набор программ с открытым исходным кодом, создание которых началось в 2003 году с целью автоматического, эффективного решения дифференциальных уравнений в частных производных [11]. Реализация метода конечных элементов состоит из нескольких этапов: получение уравнений в слабой форме, дискретизация уравнений в слабой форме, сборка значений, рассчитанных на каждом элементе, решение системы алгебраических уравнений. За работу на каждом этапе в библиотеке FEniCS отвечает отдельный компонент.

Основа библиотеки FEniCS – это набор классов, написанных на компилируемом языке C++, которые имеют интерфейс для использования в интерпретируемом языке с неявной типизацией Python. В данном комплексе программ используется встраиваемый в синтаксис Python предметно-ориентированный язык (Embedded DSL). Например, для того чтобы решить уравнение Пуассона $\Delta u = -f$ в области Ω с граничными условиями типа Дирихле $u = u_0$, необходимо передать в функцию решения вариационную запись этого уравнения:

```
WeakForm = inner(grad(u), grad(v))*dx - f*v*dx
```

В вышеприведенном листинге переменной WeakForm присваивается следующее выражение:

$$\int_{\Omega} \nabla u \cdot \nabla v dx - \int_{\Omega} f v dx, \quad (3)$$

где ∇g обозначает градиент функции g , а $U \cdot V$ – скалярное произведение векторных функций U и V . Равенство нулю выражения (3) для всех гладких функций с ограниченным носителем v составляет

вариационную запись уравнения Пуассона. Как видно из примера, приведенный выше код во многом похож на запись формулы (3).

В библиотеке FEniCS эффективность кода на языке C++ при написании на Python удается сохранить благодаря модулю Instant. Этот модуль представляет собой динамический компилятор (just-in-time compiler), преобразующий код Python в модуль C++. Для уменьшения издержек перекомпиляции, уже готовый модуль получает свой индивидуальный номер (хеш, рассчитываемый по алгоритму SHA1). Для того чтобы иметь возможность использовать OpenMP и другие внешние библиотеки, Instant позволяет указывать заголовочные файлы, скомпилированные библиотеки, пути поиска библиотек и другие флаги компиляции.

Функции библиотеки FEniCS поддерживают параллельное выполнение как на нескольких потоках на одном вычислительном узле, так и на нескольких узлах с помощью технологии MPI. Однако, в обоих случаях требуется предварительная подготовка используемых конечноэлементных сеток.

Использование метода конечных элементов, возможность параллельного исполнения кода, хорошая документированность модулей, большое количество примеров, растущее количество библиотек, активно использующих возможности FEniCS, например, для решений задач механики сплошной среды, проблем формирования и распространения океанической волн, а также близость итогового кода к математическим обозначениям стали причиной нашего выбора этой библиотеки.

4 Вычислительные эксперименты

Тестирование производилось на кластере ИММ УрО РАН "Уран". Он состоит из 178 вычислительных узлов, установленных в модулях с высокой плотностью упаковки. Вычислительные узлы оснащены процессорами Intel Xeon, работающими на частотах 2.2-3 ГГц, 16-200 ГБ оперативной памяти и графическими ускорителями NVIDIA Tesla. В общей сложности пользователям доступно 1768 вычислительных ядер CPU, 362 платы GPU и 4 ТБ оперативной памяти. Для передачи данных между вычислительными узлами используется высокоскоростная сеть Infiniband с пропускной способностью 20 Гб/сек. Для параллельного запуска на кластере установлены различные версии MPI. Дополнительно на кластер была установлена библиотека FEniCS 1.6.0. В проведенных экспериментах мы использовали MPI MVAPICH2 Intel 14.0.

Ввиду того что FEniCS поддерживает запуск в параллельном режиме, дополнительных модернизаций кода не требовалось. Для запуска вычислений с MPI использовалась команда `mpirun -f [hosts]`, где `hosts` - файл с указанием вычислительных узлов и ядер, которые будут задействованы в ходе вычисления задачи.

В качестве начальных данных используется сетка, полученная путем разбиения симметричной модели левого желудочка на тетраэдры. На рис. 1 представлена схема генерации сетки. Вначале генерируются файлы полигонов в формате PLY согласно формулам, которые определяют поверхности эндо- и эпикарда. Также файлы PLY могут быть сгенерированы с помощью систем компьютерной алгебры, например, Maple. Функции из Enchanced.py удаляют повторяющиеся точки из файла PLY. Это необходимо для дальнейшей корректной работы Gmsh, который является генератором разделения расчетной области на подобласти-тетраэдры. Для того чтобы использовать сетку, полученную от Gmsh, в FEniCS, необходимо преобразовать ее в подходящий формат. Для этих целей предназначена утилита `dolfin-convert` (составная часть пакета FEniCS). Формат HDF5 можно читать в параллельном режиме, поэтому используется дополнительное преобразование из `xml` формата.

Для проведения тестов мы использовали сетку, имеющую 3771 вершину и 14544 тетраэдра. Программа вычисляет распространение электрической волны по левому желудочку в течение 100 безразмерных единиц времени с шагом 0.125. В начальный момент времени с помощью параметров задается зона активации.

Для оценки производительности нами была проведена серия экспериментов, в которых запуски производились на 1, 2, 4, 8, 16 и 32 ядрах. Конфигурация узлов представлена в табл. 1.

Целью эксперимента было оценить масштабируемость реализации моделирования сердца в зависимости от количества используемых ядер. Результаты представлены на рис. 2. Все тесты были запущены на одном узле, кроме теста с 32 ядрами.

В итоге, эксперимент показал хорошую масштабируемость реализации на FEniCS. Однако, ускорение замедляется с увеличением ядер из-за дополнительных затрат на передачу данных.

5 Заключение

В результате данной работы из ряда существующих программных средств моделирования свойств тел сложной формы была выбрана библиотека FEniCS. Благодаря этому наш код близок к математической

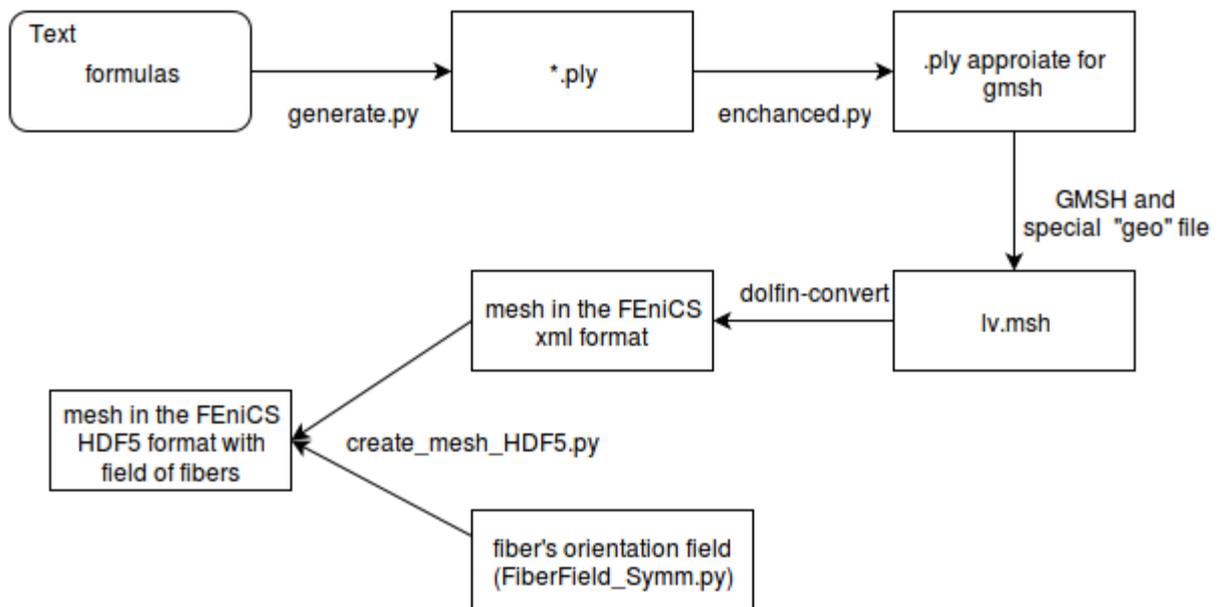


Рис. 1: Схема генерации сетки

Таблица 1: Конфигурация узлов кластера

Параметр	Значение
CPU	2 x Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz
RAM	94 GB
Interconnect	Infiniband DDR (20 Gbit)
Operating System	Linux RedHat 6.5

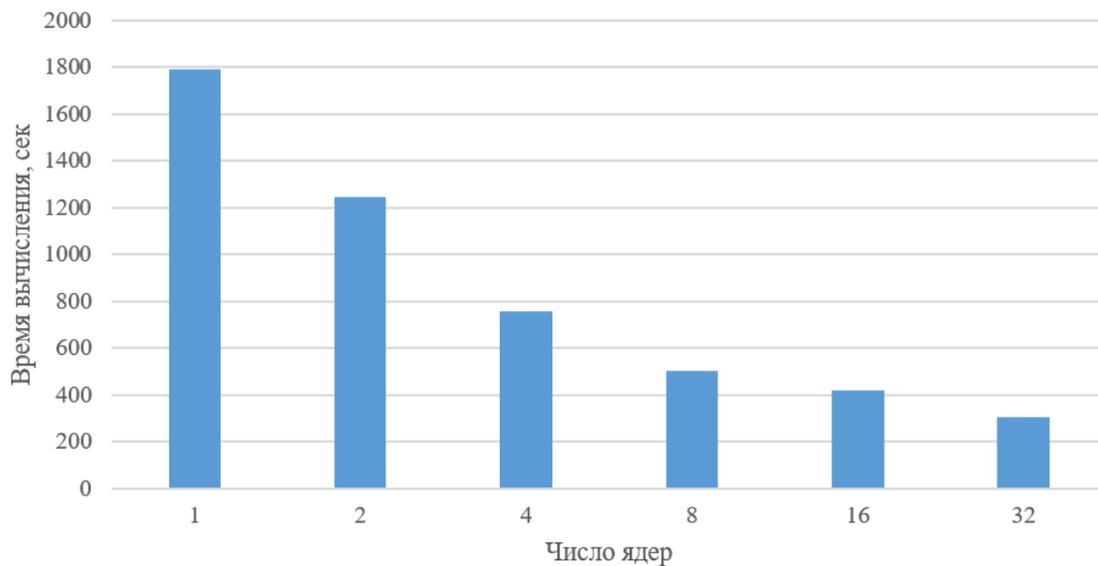


Рис. 2: Время моделирования в зависимости от числа использованных ядер

нотации. Он транслируется в высокопроизводительный код на C/C++ с поддержкой параллельного исполнения. Помимо этого, FEniCS предоставляет широкий набор готовых инструментов для решения систем дифференциальных уравнений.

На данный момент для изучения левого желудочка сердца мы используем модель Алиева–Панфилова, состоящую из двух нелинейных уравнений в частных производных.

Реализация модели Алиева–Панфилова с помощью библиотеки FEniCS продемонстрировала масштабируемость на кластере, близкую к линейной. Тестирование производилось на кластере «УРАН» ИММ УрО РАН.

В дальнейшем мы планируем добавить поддержку таких популярных ускорителей вычислений, как GPGPU и Intel Xeon Phi, использовать более сложные и приближенные к реальности модели, которые разрабатываются в ИИФ УрО РАН, и добавить к модели механический отклик на изменение потенциала действия [12].

Благодарности

Работа поддержана программой РАН I.33П «Фундаментальные проблемы математического моделирования. Фундаментальные проблемы факторизационных методов в различных областях. Алгоритмы и математическое обеспечение для вычислительных систем сверхвысокой производительности» (проект «Разработка вычислительных методов и программных средств для трехмерного моделирования сердечно-сосудистой системы», № 0401-2015-0025).

Список литературы

- [1] R. C. P. Kerckhoffs, S. N. Healy, T. P. Ustyk, A. D. McCulloch: Computational methods for cardiac electromechanics. *Proceedings of the IEEE*, 94:769–783, 2006.
- [2] A. V. Panfilov: Spiral breakup as a model of ventricular fibrillation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 8(1):57–64, 1998.
- [3] R. R. Aliev, A. V. Panfilov: A simple two-variable model of cardiac excitation. *Chaos, Solitons & Fractals*, 7(3):293–301, 1996.
- [4] S. Göktepe, E. Kuhl: Electromechanics of the heart: a unified approach to the strongly coupled excitation–contraction problem. *Computational Mechanics*, 45(2-3):227–243, Nov 2009.
- [5] S. F. Pravdin, V. I. Berdyshev, A. V. Panfilov, L. B. Katsnelson, O. Solovyova, V. S. Markhasin: Mathematical model of the anatomy and fibre orientation field of the left ventricle of the heart. *Biomedical engineering online*, 54(12):1–21, 2013.
- [6] H. Jasak, A. Jemcov, Z. Tukovic: Openfoam: A C++ library for complex physics simulations. In: *International workshop on coupled methods in numerical dynamics*, 1000:1–20, 2007.
- [7] E. J. Crampin, M. Halstead, P. Hunter, P. Nielsen, D. Noble, N. Smith, M. Tawhai: Computational physiology and the physiome project. *Experimental Physiology*, 89(1):1–26, 2004.
- [8] C. Bradley, A. Bowery, R. Britten, V. Budelmann, O. Camara, R. Christie, A. Cookson, A. F. Frangi, T. B. Gamage, T. Heidlauf, et al.: Openmiss: a multi-physics & multi-scale computational infrastructure for the vph/physiome project. *Progress in biophysics and molecular biology*, 107(1):32–47, 2011.
- [9] J. Pitt-Francis, P. Pathmanathan, M. O. Bernabeu, R. Bordas, J. Cooper, A. G. Fletcher, G. R. Mirams, P. Murray, J. M. Osborne, A. Walter, et al.: Chaste: a test-driven approach to software development for biological modelling. *Computer Physics Communications*, 180(12):2452–2471, 2009.
- [10] G. R. Mirams, C. J. Arthurs, M. O. Bernabeu, R. Bordas, J. Cooper, A. Corrias, Y. Davit, S. J. Dunn, A. G. Fletcher, D. G. Harvey, et al.: Chaste: an open source C++ library for computational physiology and biology. *PLoS Comput Biol*, 9(3):e1002970, 2013.
- [11] A. Logg, K. A. Mardal, G. Wells. Automated solution of differential equations by the finite element method: The FEniCS book. Volume 84. Springer Science & Business Media, 2012.

- [12] O. Solovyova, N. Vikulova, L. B. Katsnelson, V. S. Markhasin, P. Noble, A. Garny, P. Kohl, D. 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.

Application of FEniCS framework for simulation of the cardiac left ventricle

Timofei I. Epanchintsev^{1,2,3}, *Vladimir S. Zverev*^{1,2}, *Andrey V. Sozykin*^{1,2,3}

1 – Ural Federal University (Yekaterinburg, Russia)

2 – Krasovskii Institute of Mathematics and Mechanics (Yekaterinburg, Russia)

3 – Institute of Immunology and Physiology (Yekaterinburg, Russia)

Keywords: parallel computing, simulation, FEniCS, left ventricle.

Investigation of the functional characteristics of myocardium is a difficult task and requires mathematical models that describe processes on different levels. Toolkits for automated scientific computing frameworks have become popular. The aim of the paper is determine possibilities of application of such software for heart simulation on parallel computing systems. A small overview of existing frameworks (OpenFOAM, OpenCMISS, FEniCS, and Chaste) is presented. For our purposes FEniCS frameworks was chosen. We use the model of Aliev–Panfilov in order to investigate FEniCS facilities. Performance and scalability of FEniCS-based solution were studied.