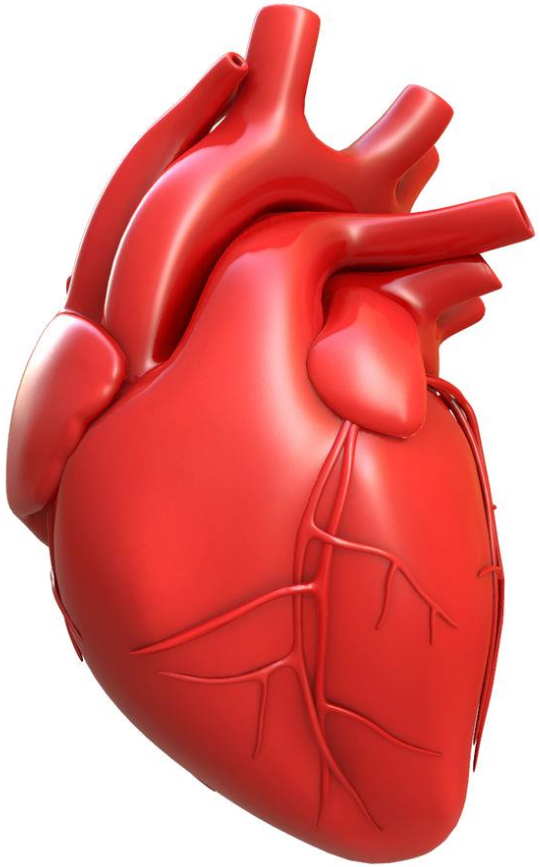


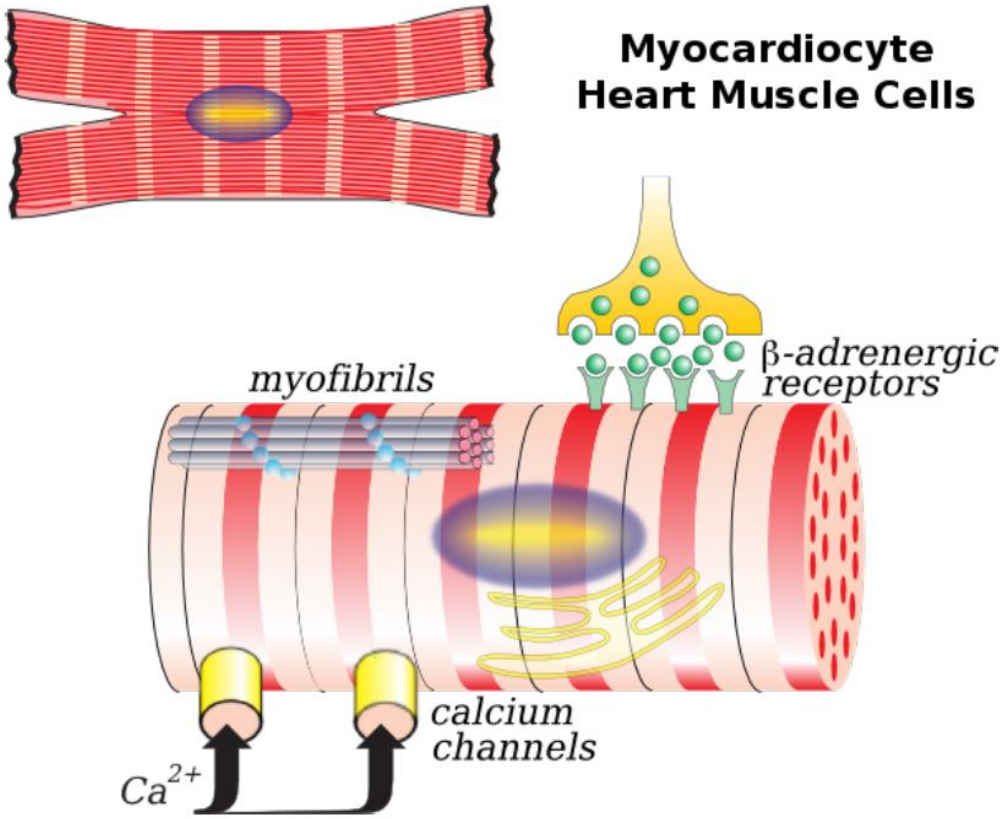
Обзор программного комплекса

Численное моделирование в кардиологии

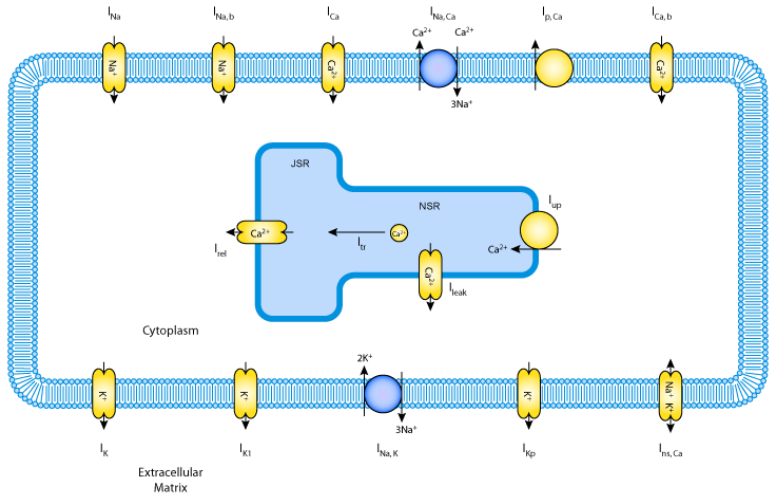
Введение в тематику

Миокард: физиология клеток

- Мышечная ткань сердца состоит из *кардиоцитов*
- Кардиоцит ≈ 0.5 (мышечная клетка + нервная клетка)
- Состояние клетки («возбуждена»/«в покое») – определяется величиной напряжения на мембране V
- Историческое название V – *мембранный потенциал*



Кардиомиоцит: модель Luo-Rudy II (1994)

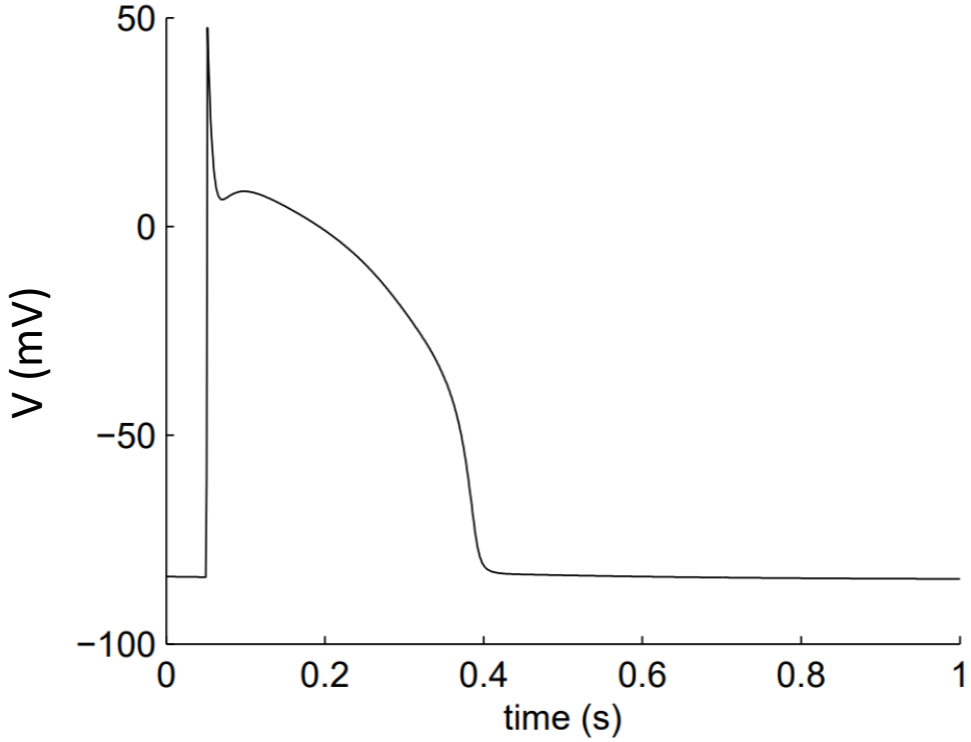


(возможность моделировать воздействие *лекарственных препаратов)*

$$\begin{cases} C_m \dfrac{dV}{dt} = -I_{ion}(\boldsymbol{u}, V) \\ \dfrac{d\boldsymbol{u}}{dt} = f(\boldsymbol{u}, V) \\ I_{ion} \equiv \sum_j I_j\,(\boldsymbol{u}, V), \qquad j = Na, K, Ca, \ldots \end{cases}$$

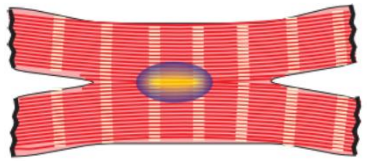
V – мембранный потенциал

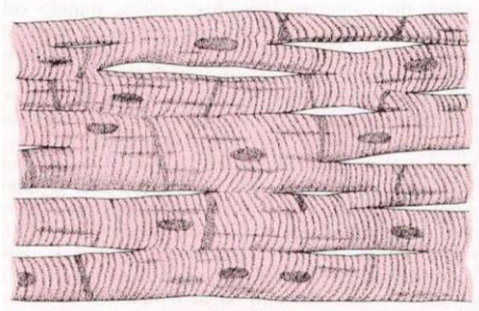
$\boldsymbol{u}$ – вектор воротных переменных



Модели связности миокарда

Кардиоциты соединены контактами (омическими сопротивлениями):





Модель миокарда как *сплошной* среды: $V = V(\boldsymbol{x}, t), \qquad \boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t), \qquad \varphi_e = \varphi_e(\boldsymbol{x}, t)$

Кабельное уравнение или «Monodomain»:

$$\begin{cases} C_m \dfrac{\partial V}{\partial t} = -I_{ion}(\boldsymbol{u}, V) + \nabla \cdot (\sigma_i \nabla V) \\ \dfrac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{f}(\boldsymbol{u}, V) \\ I_{ion} \equiv \sum_j I_j(\boldsymbol{u}, V), \qquad j = Na, K, Ca, \dots \end{cases}$$

«Bidomain»:

$$\begin{cases} C_m \dfrac{\partial V}{\partial t} = -I_{ion}(\boldsymbol{u}, V) + \nabla \cdot \left(\sigma_i \nabla (V + \varphi_e)\right) \\ \nabla \cdot (\sigma_i \nabla V + (\sigma_i + \sigma_e) \nabla \varphi_e) = 0 \\ \dfrac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{f}(\boldsymbol{u}, V) \\ I_{ion} \equiv \sum_j I_j(\boldsymbol{u}, V), \qquad j = Na, K, Ca, \dots \end{cases}$$

Chaste (2005 – H.B.)

Программный комплекс для численного моделирования электрической активности миокарда.

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE

Chaste

https://www.cs.ox.ac.uk/chaste/cardiac_index.html

https://github.com/Chaste/Chaste

You are here: Home > Research > Projects > Chaste

Cardiac Chaste: developing software for realistic heart simulations

Introduction and aims

Computational modelling of the heart is now recognised as a powerful technique in the detailed investigation of cardiac behaviour. One of the major contributions of computational approaches to cardiovascular research has been the ability to dissect various effects and to tease out important relations between parameters, which are not possible using current experimental techniques. Recently, advanced computer models of cardiac electro-mechanical activity have been developed. The electrical properties of the myocardium are generally described by the bidomain equations, a set of coupled parabolic and elliptic partial differential equations (PDEs) that represents the tissue as two separate, distinct continua - one intracellular and the other extracellular. The intracellular and the extracellular media are connected via the cell membrane, and thus the two PDEs are coupled at each point in space through a set of complex, non-linear ordinary differential equations (ODEs), which describe the ionic transport across the cell membrane. Certain modelling environments use the monodomain representation of cardiac activity, which involves solving a single parabolic PDE, by assuming either that the extracellular potentials are negligible, or that the anisotropy ratios are equal in the intracellular and the extracellular domains.

«Monodomain»: алгоритм численного решения

Переход с явного на неявный слой:

1. $\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{f}(\boldsymbol{u}, V)$ $\xrightarrow[\text{по времени}]{\text{Дискретизация}}$ $\frac{u^{n+1} - u^n}{\Delta t} = \boldsymbol{f}(\boldsymbol{u}^n, V^n)$

(алгебраические уравнения)

2. $C_m \frac{\partial V}{\partial t} = -I_{ion}(\boldsymbol{u}, V) + D \frac{\partial^2 V}{\partial x^2}$ $\xrightarrow[\textbf{только}\text{ по времени}]{\text{Дискретизация}}$ $C_m \frac{V^{n+1} - V^n}{\Delta t} = -I_{ion}(\boldsymbol{u}^n, V^n) + D \cdot \nabla^2 V^{n+1}$

(стационарное линейное УрЧП)

$$V^{n+1} - \frac{\Delta t}{C_m} D \cdot \nabla^2 V^{n+1} = V^n - \frac{\Delta t}{C_m} I_{ion}(\boldsymbol{u}^n, V^n)$$

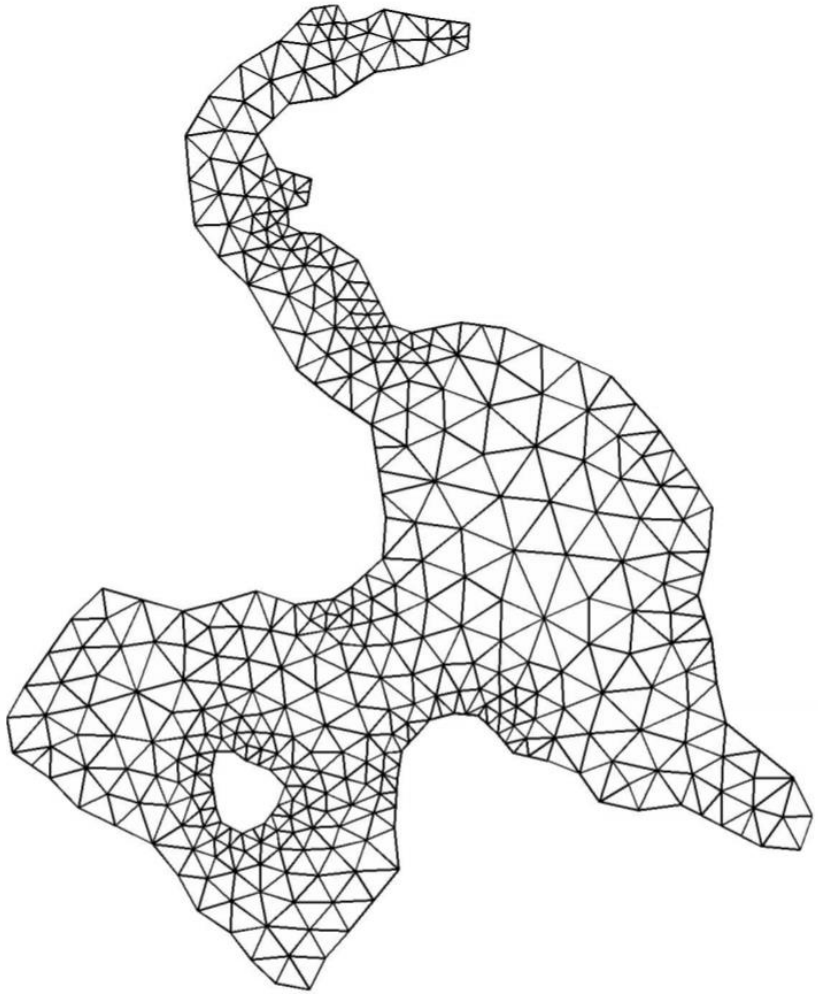
$$V^{n+1} \equiv y, \qquad B \equiv V^n - \frac{\Delta t}{C_m} I_{ion}(\boldsymbol{u}^n, V^n)$$

$$y - A\nabla^2 y = B, \qquad A \in \mathbb{R}, \qquad B - \text{функция}$$

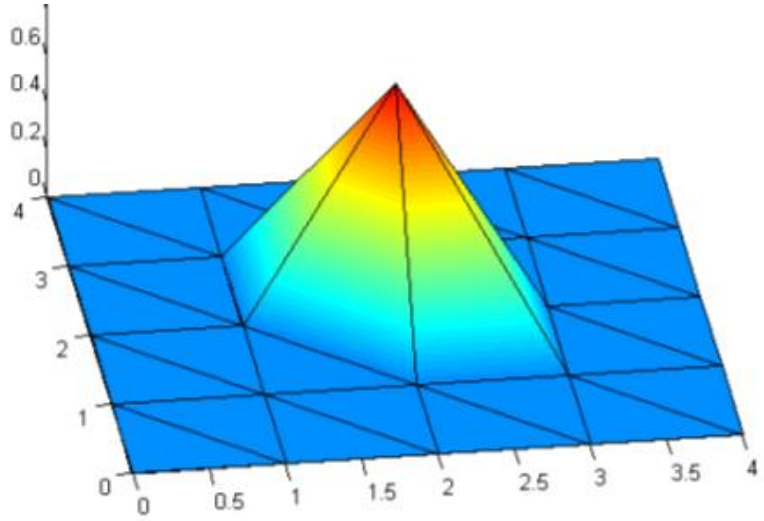
$$\boxed{\nabla^2 y = \frac{1}{A} y - \frac{1}{A} B} - \text{стационарное линейное УрЧП}$$

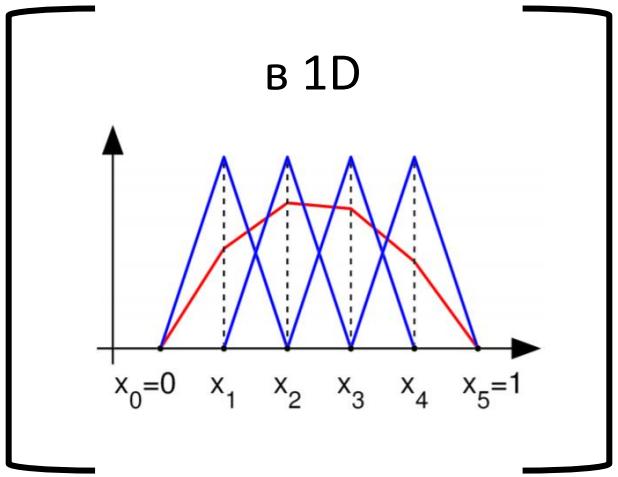
«Monodomain»: МКЭ-дискретизация по пространству

- Расчетная сетка



- Базисные функции в 2D





Chaste

«Monodomain»: МКЭ-дискретизация по пространству

$$\nabla^2 y = \frac{1}{A} y - \frac{1}{A} B$$

$$y_n = \sum_j \alpha_j \varphi_j\,,\ \alpha_i \in \mathbb{R},\ \varphi_j - \text{базисная функция}$$

Нахождение коэффициентов методом Галеркина:

$$\left(\frac{1}{A} y_n - \frac{1}{A} B - \nabla^2 y_n, \varphi_i\right) = 0, \qquad i = 0,1, \ldots$$

а)

$$(y_n, \varphi_i) = \left(\sum_j \alpha_j \varphi_j\,, \varphi_i\right) = \sum_j \alpha_j \left(\varphi_j, \varphi_i\right)$$

б) «Интегрирование по частям» в многомерном случае:

$$-(\nabla^2 y_n, \varphi_i) \equiv -\int_\Omega \nabla^2 y_n(\boldsymbol{x}) \varphi_i(\boldsymbol{x}) dV = \int_\Omega \nabla y_n(\boldsymbol{x}) \cdot \nabla \varphi_i(\boldsymbol{x}) dV - \underbrace{\int_{\partial\Omega} (\nabla y_n \cdot \boldsymbol{n}) \varphi_i(\boldsymbol{x}) dS}_{=0}$$

$$= \int_\Omega \nabla y_n(\boldsymbol{x}) \cdot \nabla \varphi_i(\boldsymbol{x}) dV \equiv \underbrace{a[y_n, \varphi_i]}_{\text{билинейная форма}} = a\left[\sum_j \alpha_j \varphi_j\,, \varphi_i\right] = \sum_j \alpha_j\, a[\varphi_j, \varphi_i]$$

«Monodomain»: МКЭ-дискретизация по пространству

Вводим обозначения:

$$M_{ij} = (\varphi_j, \varphi_i) - \text{матрица масс}$$

$$K_{ij} = a[\varphi_j, \varphi_i] - \text{матрица жесткости}$$

$$b_i = \frac{1}{A}(B, \varphi_i) - \text{вектор нагрузки.}$$

С учетом данных обозначений:

$$\frac{1}{A}\sum_j M_{ij}\alpha_j + \sum_j K_{ij}\alpha_j = b_i, \qquad i = 0,1, \ldots$$

$$\left(\frac{1}{A}M + K\right)\boldsymbol{\alpha} = \boldsymbol{b}.$$

Окончательный вид записи (СЛАУ):

$$T\boldsymbol{\alpha} = \boldsymbol{b}. (*)$$

Chaste

Алгоритм численного решения

1. Задать начальные условия для $\{V, u\}$
2. while $\mathrm{t}_{\mathrm{current}} < \mathrm{t}_{\mathrm{end}}$:
   a) Уравнения для воротных переменных:
      a) сделать шаг по времени для уравнений воротных переменных
   b) Уравнение для мембранного потенциала
      a) Вычислить матрицу T СЛАУ (*)
      b) Вычислить вектор нагрузки $\boldsymbol{b}$ СЛАУ (*)
      c) Решить СЛАУ

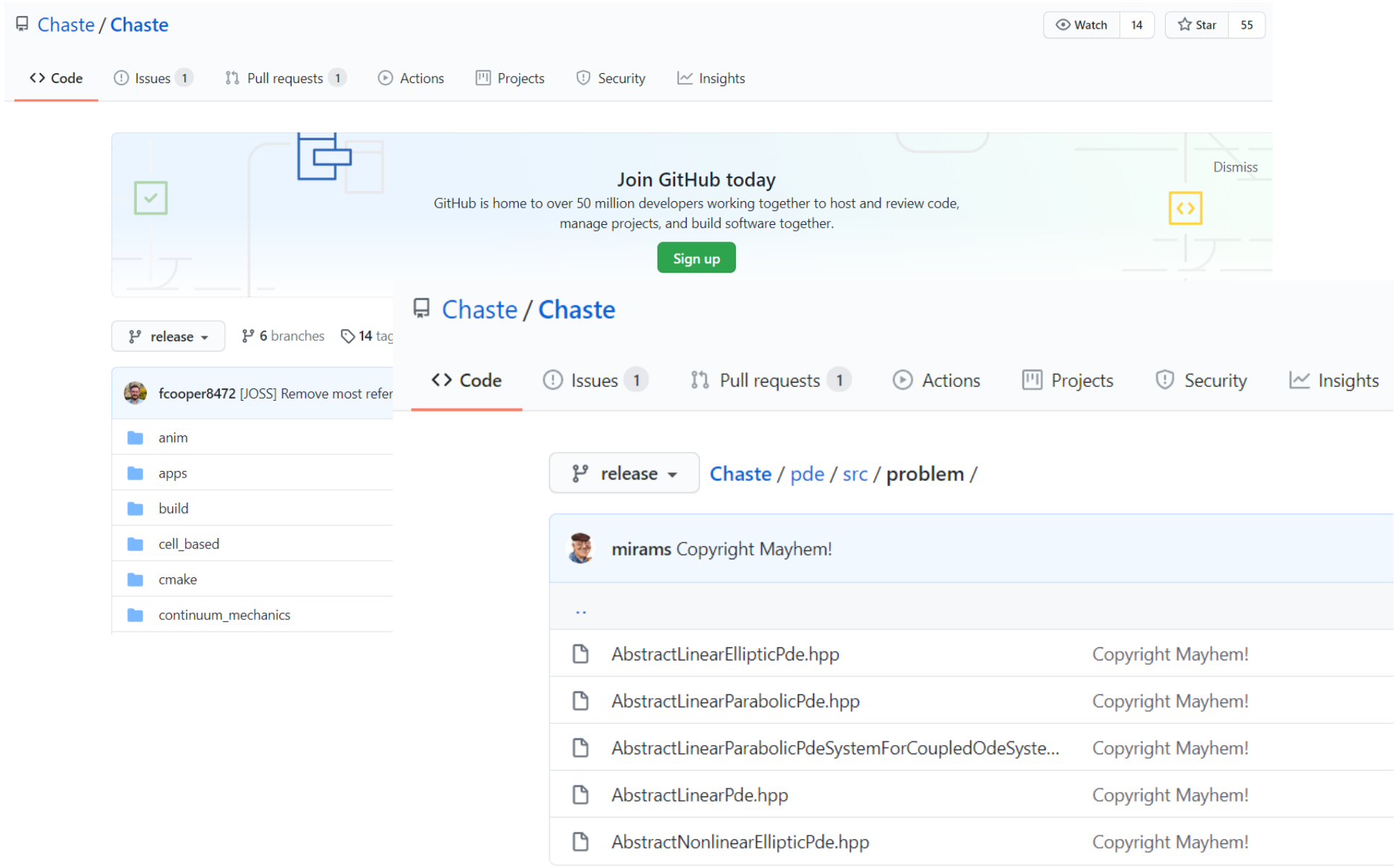
   3. Обработать результаты

ЗАМЕТКА по оптимизации: матрицу T не требуется вычислять на каждом шаге по времени, т.к. ее элементы не изменяются при переходе со слоя на слой.

Программная реализация

https://github.com/Chaste/Chaste

Репозиторий



Часть заголовочного файла (современный C++)

```
59  template<unsigned ELEMENT_DIM, unsigned SPACE_DIM=ELEMENT_DIM, unsigned PROBLEM_DIM=1>
60  class AbstractLinearParabolicPdeSystemForCoupledOdeSystem
61  {
62  public:
63      /**
64       * @return computed function c_i(x).
65       *
66       * @param rX the point x at which the function c_i is computed
67       * @param pdeIndex the index of the PDE, denoted by i above
68       */
69      virtual double ComputeDuDtCoefficientFunction(const ChastePoint<SPACE_DIM>& rX, unsigned pdeIndex)=0;
70
71      /**
72       * @return computed source term f_i(x, u_1, u_2, ..., u_p) at a point in space.
73       *
74       * @param rX the point x at which the nonlinear source term is computed
75       * @param rU the vector of dependent variables (u_1, u_2, ..., u_p) at the point x
76       * @param rOdeSolution the ODE system state vector (v_1, ..., v_q) at the point x (if an ODE system is present)
77       * @param pdeIndex the index of the PDE, denoted by i above
78       */
79      virtual double ComputeSourceTerm(const ChastePoint<SPACE_DIM>& rX,
80                                       c_vector<double,PROBLEM_DIM>& rU,
81                                       std::vector<double>& rOdeSolution,
82                                       unsigned pdeIndex)=0;
```

Обозначения

- Абстрактный класс/Чисто виртуальный метод
- Конкретный класс
- Реализованный метод

Модели кардиоцитов в Chaste

AbstractODESystem:

- StateVariables
- EvaluateYDetivatives(t, y)

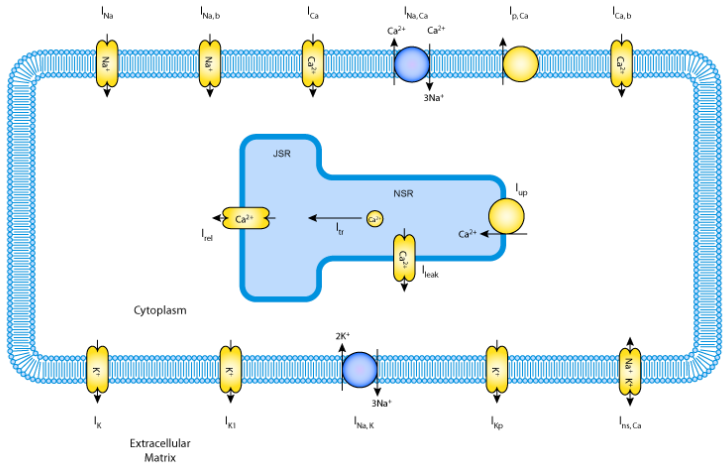
AbstractCardiacCell *наследуется от* AbstractODESystem:

- ODESolver
  - имеет тип AbstractOdeSolver()
- Compute(t0, t1)
  - использует ODESolver для проведения численного решения

LuoRudyCellModel *наследуется от* AbstractCardiacCell:

- Реализован: EvaluateYDerivatives(t, y)

$$\begin{cases} C_m \dfrac{dV}{dt} = -I_{ion}(\boldsymbol{u}, V) \\ \dfrac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{u}, V) \\ I_{ion} \equiv \sum_j I_j(\boldsymbol{u}, V), \qquad j = Na, K, Ca, \ldots \end{cases}$$

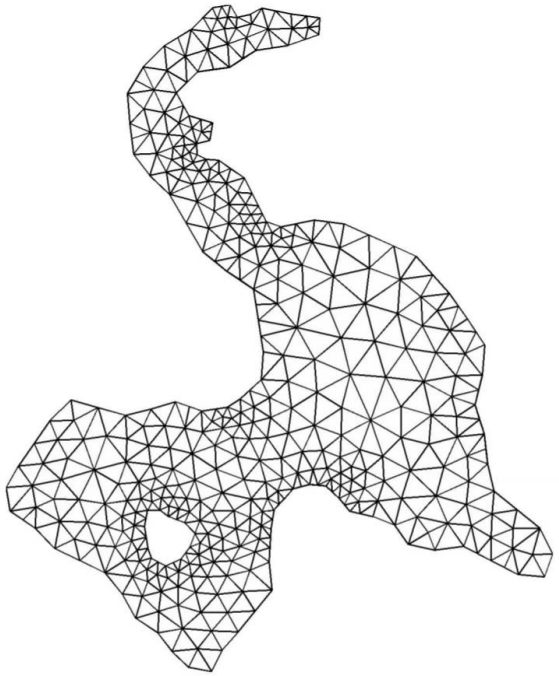


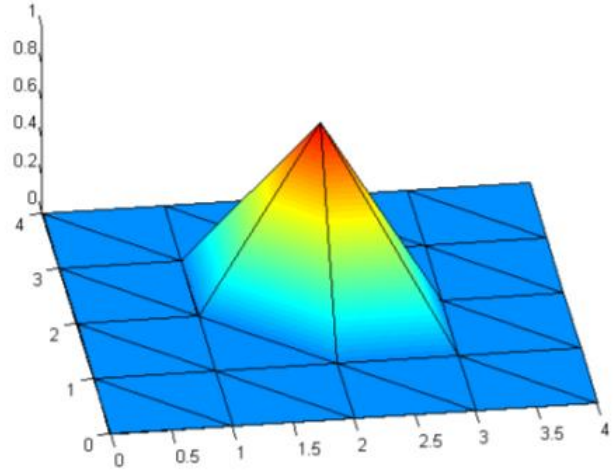
Chaste

Сетка и базисные функции 2D в Chaste: классы

Mesh:

- Nodes
- Elements
- BoundaryElements
- BoundaryNodeIndices





AbstractBasisFunction:

- GetValues(xi)
- GetTransformedDerivatives(xi, J)

LinearBasisFunction *наследуется от* AbstractBasisFunction:

QuadraticBasisFunction *наследуется от* AbstractBasisFunction:

Chaste

Солверы УрЧП в Chaste: иерархия

AbstractLinearPdeSolver:

SetupLinearSystem()

AbstractStaticPdeSolver *наследуется от* AbstractLinearPdeSolver:

Solve():

вызывает SetupLinearSystem() и затем решает СЛАУ

AbstractDynamicPdeSolver *наследуется от* AbstractLinearPdeSolver:

SetTimes(t0, t1)

SetInitialCondition(initialCondition)

Solve():

На каждой итерации цикла по времени вызывает SetupLinearSystem() и решает СЛАУ

MonodomainSolver *наследуется от AbstractDynamicPdeSolver*:

… (рассмотрим на следующем слайде)

Солверы УрЧП в Chaste

MonodomainSolver *наследуется от* *AbstractDynamicPdeSolver*:

MonodomainTissue:

набор AbstractCardiacCell

информация о проводимости σ_i

Реализован: SetupLinearSystem()

BidomainSolver *наследуется от* *AbstractDynamicPdeSolver*:

BidomainTissue:

набор AbstractCardiacCell

информация о проводимостях σ_i, σ_e

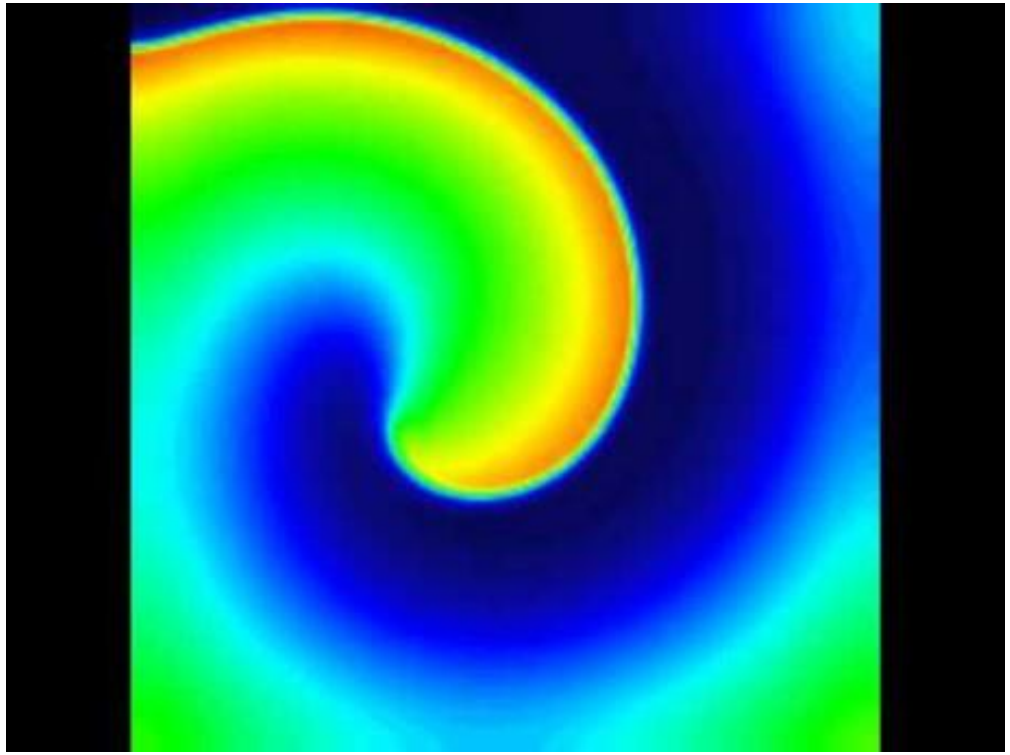
Реализован: SetupLinearSystem() (отличается от реализации в MonodomainSolver)

Результаты моделирования

Ревербератор (реентри) – спиральная волна

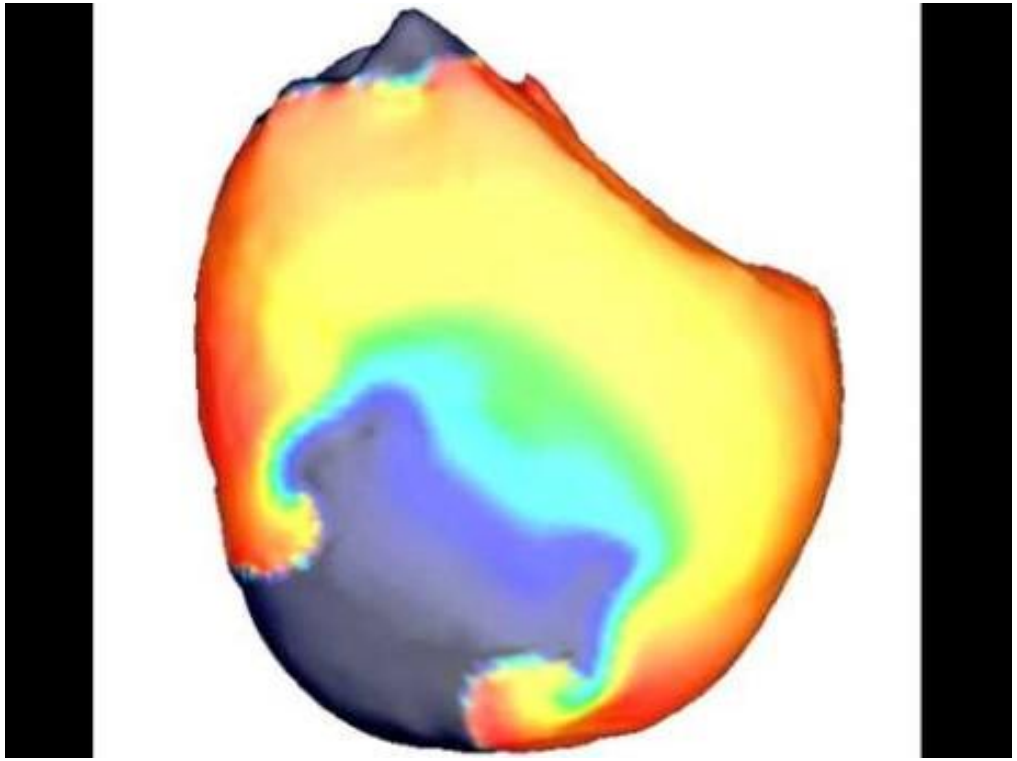
Может образовываться при определенных начальных условиях или при отмирании участка миокарда (при *инфаркте*).

Мембранный потенциал V, 2D



https://www.youtube.com/watch?v=0cgaUm5LPcU

Мембранный потенциал V, 3D



https://www.youtube.com/watch?v=5VY7wpHsezg

Chaste

Реентри → фибрилляция

Мембранный потенциал V , 3D

https://www.youtube.com/watch?v=eV9ajG-PGXg

Chaste

Спасибо за внимание.
Готов ответить на Ваши вопросы.

karpaev@phystech.edu

Спасибо за внимание.
Готов ответить на Ваши вопросы.

karpaev@phystech.edu