Farina *et al. Adv. Model. and Simul. in Eng. Sci.(2021)8:5*
https://doi.org/10.1186/s40323-021-00191-8

Advanced Modeling and Simulation in Engineering Sciences

RESEARCH ARTICLE **Open Access**

# A cut finite element method for spatially resolved energy metabolism models in complex neuro-cell morphologies with minimal remeshing

Sofia Farina[1], Susanne Claus[3], Jack S. Hale[1], Alexander Skupin[1,2] and Stéphane P. A. Bordas[1*]

*Correspondence: stephane.bordas@alum.northwestern.edu, stephane.bordas@uni.lu
[1]Institute of Computational Engineering, University of Luxembourg,, Maison du Nombre, 6 Avenue de la Fonte, 4364 Esch-sur-Alzette, Luxembourg
Full list of author information is available at the end of the article

**Abstract**

A thorough understanding of brain metabolism is essential to tackle neurodegenerative diseases. Astrocytes are glial cells which play an important metabolic role by supplying neurons with energy. In addition, astrocytes provide scaffolding and homeostatic functions to neighboring neurons and contribute to the blood–brain barrier. Recent investigations indicate that the complex morphology of astrocytes impacts upon their function and in particular the efficiency with which these cells metabolize nutrients and provide neurons with energy, but a systematic understanding is still elusive. Modelling and simulation represent an effective framework to address this challenge and to deepen our understanding of brain energy metabolism. This requires solving a set of metabolic partial differential equations on complex domains and remains a challenge. In this paper, we propose, test and verify a simple numerical method to solve a simplified model of metabolic pathways in astrocytes. The method can deal with arbitrarily complex cell morphologies and enables the rapid and simple modification of the model equations by users also without a deep knowledge in the numerical methods involved. The results obtained with the new method (CutFEM) are as accurate as the finite element method (FEM) whilst CutFEM disentangles the cell morphology from its discretisation, enabling us to deal with arbitrarily complex morphologies in two and three dimensions.

**Keywords:** CutFEM, Unfitted methods, FEM, Level sets, Reaction diffusion system, Energy metabolism

## Introduction

We propose to test and verify a simple numerical framework to solve a simplified model of metabolic pathways, representative of cellular metabolism in the brain. Metabolic models can aid the understanding of cell behaviour. Most metabolic models involve the solution of a system of ODEs [1] leaving open the question of how the geometry and spatial organization affect cell behaviour. This work presents a method that is a first step towards extending existing models to answer this question. The method, based on recent develop-

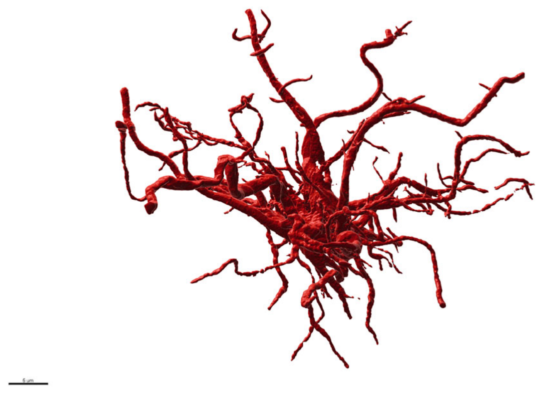


**Fig. 1** The complex morphology of an astrocyte from a human post mortem sample obtained by fluorescent (GFAP) super-resolution light microscopy [63] and reconstructed by the machine learning based tool MicMac [5]

ments in unfitted finite element methods is general, and can deal with an arbitrary number of coupled reaction diffusion equations and handle complex cell morphologies. Thanks to the versatility of the automatic code infrastructure provided by FEniCS, the code is well-suited to newcomers to finite element methods interested in modelling biological systems.

To test the new framework, we address complex cell geometries in two and three dimensions and compare the method to the standard finite element method in terms of accuracy. The results obtained with the new method CutFEM are as accurate as the finite element method (FEM) whilst CutFEM disentangles the cell morphology from its discretisation, enabling to deal with arbitrarily complex morphologies, including kinks and triple junctions, in two and three dimensions.

A thorough understanding of brain metabolism is essential to tackle neurodegenerative diseases [2,3]. Astrocytes are glial cells which play an important metabolic role by supplying neurons with energy [4]. These cells also provide scaffolding and help repair neighboring neurons, where they maintain balanced ionic concentrations (homeostasis) and contribute to the blood–brain barrier by preventing the diffusion of large molecules into the brain.

Recent investigations show that the morphology of astrocytes, which can be complex (see Fig. 1), impacts upon their function, in particular the efficiency with which these cells metabolise nutrients and provide neurons with energy [5]. Modelling and simulation could be effective in furthering our understanding brain metabolism by enabling to test biological hypotheses and investigate the relative importance of model parameters on quantities of interest to biologists.

The challenges involved with modelling and simulating metabolic activity in cells include:

1. Building a representative model of the metabolic pathways, usually a set of reaction–diffusion equations;
2. Identifying the parameters for these partial differential equations (PDEs) as well as the sensitivity of the system to these parameters [6–11];
3. Discretising the complex and evolving geometries of the cells;

4. Solving this set of PDEs on these complex domains;
5. Ensuring the accuracy of the solution by measuring discretisation error [12–14];
6. Ensure the usability of the numerical framework by non-experts.

This last point aims at simplifying multi-disciplinary interactions between computational, data science and domain experts, it is also becoming increasingly important to devise numerical frameworks which can be used and enhanced without being a computational science expert. To do so, open-source frameworks such as FEniCS [15–17], getFEM [18], FreeFEM [19], Deal.ii [20] are all possible candidates. These open-source frameworks enable the user to write models in a language which is natural to them and requires minimal interaction with technical details associated with well-established numerical methods.

Concerning metabolic activities, some models focused their attention to the metabolic pathways which they describe as a series of chemical reactions that enable the synthesis and breakdown of molecules as [21–24].

So far, very little attention has been paid to the role of cell morphology on the modelling [25,26]. However, cell morphology is known to be important to describe the state of the cell. For example, [27] show the importance of cell morphology on the development of Alzheimer's disease.

In this paper, we tackle points 1., 3., 4., and 6., above, and leave points 2. and 5. for further communications, as well as the extension to evolving domains.

*Model* We present a simple model of energy metabolism which takes into account the main pathways of the metabolic process in a single cell. Here, we focus on an astrocyte as a specific cell, but the developments are general. In particular, we include glycolysis, Lactate Dehydrogenase, TCA cycle and basal cellular activity explicitly in the cell model. Each pathway is described by a chemical reaction leading to a coupled reaction diffusion system.

*Cell geometry* Our ultimate goal is to enable the use of microscopy to produce input geometries for our computational framework and include the evolution of the cellular domain. To enable this, we define cell geometries using signed distance functions, also known as level-set functions [28,29], which were successfully deployed in modeling other biological phenomena with moving interfaces, such as [30,31].

*Discretisation*

The solution of the set of coupled reaction diffusion equations has two characteristics, which create challenges for the standard finite element method:

- Local gradients in metabolite concentrations;
- Discontinuities across cell boundaries.

Standard finite element method would require to create a mesh that fits the boundary of the object. Even though much progress has been made in meshing technology [32], the advantages of methods that separate the geometry from the object are very appealing for our ultimate objective of describing a cell evolving in time. Classic FEM would require to build a mesh conforming the object at each time step leading to very high computational drawbacks.

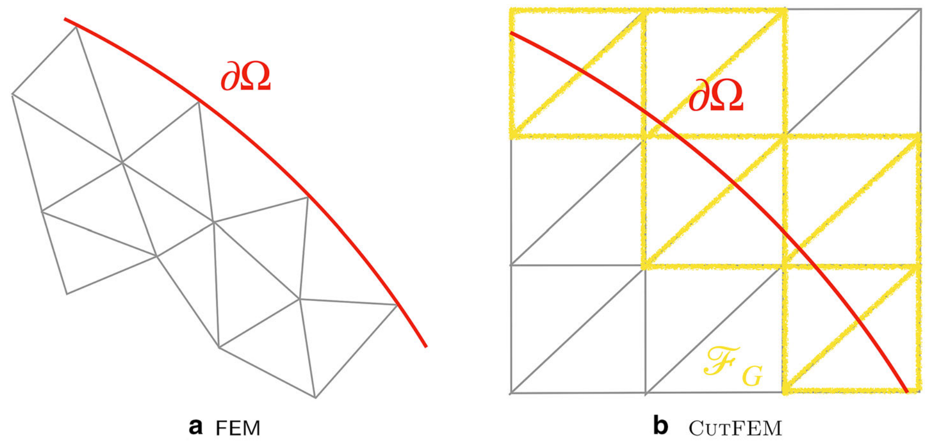


**Fig. 2** **a** Standard FEM requires the mesh to be conform the boundary of the domain $\partial\Omega$ **b** CutFEM embeds the implicitly defined interface $\partial\Omega$ into a background mesh, in yellow the set of ghost penalty facets $\mathcal{F}_G$

Enriched finite element methods such as the partition of unity FEM (PUFEM) [33, 34], the generalized finite element method (GFEM) [35] and the extended finite element method [36] are ideal to tackle these two challenges [30,37].

Indeed, these methods enable the local enrichment of the discrete solution space with known features about the solution, including discontinuities and sharp gradients or singularities. This makes it possible to handle arbitrarily complex geometries quasi-independently of the mesh (see Fig. 2).

Nonetheless, without preconditioning or special treatment [38–44] enriched finite elements cannot natively deal with arbitrarily complex geometries because of particular geometrical limit cases (interfaces passing close to a degree of freedom) and ill-conditioning stemming from linear dependencies due to complex enrichment functions acting upon large parts of the computational domain.

CutFEM [45–51] is an extension of XFEM that naturally addresses limit cases associated with complex geometries, and lends itself to image-based simulations. Moreover, we use the libCutFEM library, which is a cut finite element extension of the open-source framework of the FEniCS Project [15–17,47]. FEniCS offers a highly flexible and easy way of transforming models expressed as partial differential equations into numerical methods based on the finite element method through the Unified Form Language (UFL) [52], a domain-specific programming language for writing the weak form of partial differential equations. The UFL specification of the finite element problem is then automatically translated by the other components of the FEniCS Project (DOLFIN [16] and the FEniCS Form Compiler (FFC) [15]) into high-performance specific C++ code with little or no user intervention.

*Solution scheme* To solve the set of coupled, time-dependent and non-linear PDEs, we first discretise in time using a standard implicit time stepping scheme. The non-linear equation is solved using a Newton–Raphson scheme. The Jacobian, required for the Newton–Raphson scheme, is calculated automatically at the symbolic level by FEniCS. This greatly eases the implementation from the user's perspective, as deriving and implementing the consistent Jacobian manually can be a tedious and error prone task. The resulting linear systems at each step of the Newton–Raphson algorithm is solved using standard linear solvers from the PETSc library [53].

*Usability and extensibility of the framework*

Thanks to the flexibility of FEniCS, the framework is generic and usable by biologists without a background or specific training in Computational Sciences or Applied Mathematics. The code used to create the two of the examples is freely online accessible (see "Availability of data and materials") and can be adapted to other problems straightforwardly.

The paper is organized as follows. In "The problem formulation" section, we present the biological model for energy metabolism followed by the governing equation and its weak form. The FEM and CutFEM discretization can be found in "Discretization" section. In "Implementation" section a detailed explanation of how implement our model in CutFEM is presented. Our numerical results are introduced in "Numerical results" section and the conclusions follow in "Conclusion and discussion" section.

## The problem formulation

The aim of this section is to introduce the energy metabolism model, which can be expressed as a set of partial differential equations, in its strong and weak form.

### Basic model for energy metabolism

The scope of this model is to isolate conceptually the essential mathematical properties of the metabolic processes of a cell, the mechanism that generate energy for cellular activities from the synthesis and breakdown of nutrients [1].

The simplified model of metabolic pathway in a cell is sketched in Fig. 3 and described by the following reactions (see "Abbreviations" for details)

$$\mathrm{HXK} := \mathrm{GLC} + 2\,\mathrm{ATP} \rightarrow 2\,\mathrm{ADP} + 2\mathrm{GLY} \tag{1}$$

$$\mathrm{PYRK} := \mathrm{GLY} + 2\,\mathrm{ADP} \rightarrow 2\,\mathrm{ATP} + \mathrm{PYR} \tag{2}$$

$$\mathrm{LDH} := \mathrm{PYR} \rightarrow \mathrm{LAC} \tag{3}$$

$$\mathrm{Mito} := \mathrm{PYR} \rightarrow 28\,\mathrm{ATP} \tag{4}$$

$$\mathrm{act} := \mathrm{ATP} \rightarrow \mathrm{ADP}. \tag{5}$$

The pathway starts when the molecules derived from food (nutrients) enter the cytosol of the cell, and the process called glycolysis starts the breakdown of glucose (GLC) into two molecules of pyruvate (PYR). The glycolysis process can be split, to our modeling purpose, into two main chemical reactions: ATP-consuming (1) and ATP-producing (2). The pyruvate produced by glycolysis can, then, be converted into lactate (LAC) by the enzyme lactate dehydrogenase (LDH) simplified with the chemical reaction (3) or enter mitochondria and used to produce ATP through the TCA cycle shown in reaction (4).

Last, we take into account the activity of the cell which uses ATP for its own sustenance represented by the chemical reaction (5).

In order to facilitate our model we consider that the backward reactions are negligible. Moreover, we consider that the enzymes that catalyzed the chemical reactions are located in a specific region of the cellular domain.

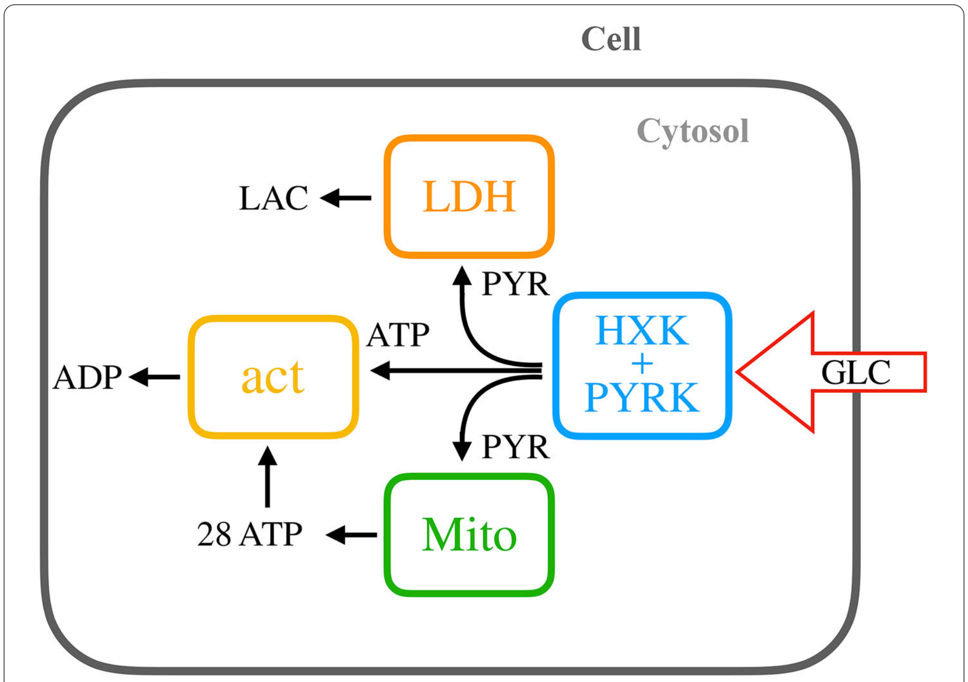


**Fig. 3** The main pathways we consider in our Simplified Model of Energy Metabolism of a cell. The glucose (GLC) enters inside the cell into the cytosol and takes part into the glycolysis composed by the two reactions HXK and PYRK. The products of the glycolysis, PYR and ATP, are then used into LDH reaction, that generates LAC, into act reaction, that describes the cellular activity producing ADP, and inside the mitochondria where the reaction denoted Mito happens generating the major source of ATP for the cell

**Strong formulation of governing equations**

Let $\Omega$ be an open and bounded subset of $\mathbb{R}^d$ (d = 2 or 3) with Lipschitz-continous boundary and we denote the concentration of the chemical species using the bracket notation, $[\,\cdot\,]$, as a function

$$[\,\cdot\,] : \Omega \times [0, T] \to \mathbb{R}.$$

Mathematically, we can express the sequence of reactions Eqs. (1)–(5) with a coupled system of semi-linear parabolic reaction diffusion equations [54]. We consider that the species involved are subject to diffusion in the domain and we denote $D_{[\,\cdot\,]} > 0$ the diffusive constants. The reactions obey to the law of mass action [55] where $K_{\mathrm{HXK}}, K_{\mathrm{PYRK}}, K_{\mathrm{LDH}}, K_{\mathrm{Mito}}$ and $K_{\mathrm{act}}$ are the rate constants and we introduce Gaussian functions, indicated with $\mathcal{G}_{\mathrm{HXK}}(x_0, \sigma)$, $\mathcal{G}_{\mathrm{PYRK}}(x_0, \sigma)$, $\mathcal{G}_{\mathrm{LDH}}(x_0, \sigma)$, $\mathcal{G}_{\mathrm{Mito}}(x_0, \sigma)$ and $\mathcal{G}_{\mathrm{act}}(x_0, \sigma)$, to locate the region where the reactions are happening

$$\mathcal{G}_{\cdot} = \mathcal{G}(x_{0,\cdot}, \sigma_{\cdot}) = \frac{1}{\sqrt{2\pi\sigma_{\cdot}^2}} \exp - \frac{(x - x_{0,\cdot})^2}{2\sigma_{\cdot}^2},$$

where $x_{0,\cdot} \in \Omega_{\cdot}$ and $\sigma \in \mathbb{R}$. Eventually, to represent the entrance of the glucose inside the cytosol we define a source term function $f : A \times [0, T] \to \mathbb{R}$, where $A$ is a subset of $\Omega$ as

$$f(x, t) = \begin{cases} \alpha \in \mathbb{R} & \text{if} \quad (x, t) \in A \times [0, 1], \\ 0 & \text{otherwise.} \end{cases}$$

The strong form of the reaction diffusion system is then, finding the concentrations $[\mathrm{GLC}](x,t)$, $[\mathrm{ADP}](x,t)$, $[\mathrm{ATP}](x,t)$, $[\mathrm{GLY}](x,t)$, $[\mathrm{PYR}](x,t)$ and $[\mathrm{LAC}](x,t)$ for all $x \in \Omega$ and $t \in [0,T]$ such that

$$\begin{cases} \frac{\partial[\mathrm{GLC}]}{\partial t} = & D_{[\mathrm{GLC}]}\nabla^2[\mathrm{GLC}] - K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}} + f \\ \frac{\partial[\mathrm{ATP}]}{\partial t} = & D_{[\mathrm{ATP}]}\nabla^2[\mathrm{ATP}] - 2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}} \\ & +2K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}} + 28K_{\mathrm{Mito}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{Mito}} \\ & -K_{\mathrm{act}}[\mathrm{ATP}]\mathcal{G}_{\mathrm{act}} \\ \frac{\partial[\mathrm{ADP}]}{\partial t} = & D_{[\mathrm{ADP}]}\nabla^2[\mathrm{ADP}] + 2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}} \\ & -2K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}} + K_{\mathrm{act}}[\mathrm{ATP}]\mathcal{G}_{\mathrm{act}} \\ \frac{\partial[\mathrm{GLY}]}{\partial t} = & D_{[\mathrm{GLY}]}\nabla^2[\mathrm{GLY}] + 2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}} \\ & -K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}} \\ \frac{\partial[\mathrm{PYR}]}{\partial t} = & D_{[\mathrm{PYR}]}\nabla^2[\mathrm{PYR}] + K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}} \\ & -K_{\mathrm{LDH}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{LDH}} - K_{\mathrm{Mito}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{Mito}} \\ \frac{\partial[\mathrm{LAC}]}{\partial t} = & D_{[\mathrm{LAC}]}\nabla^2[\mathrm{LAC}] + K_{\mathrm{LDH}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{LDH}}. \end{cases} \tag{6}$$

The system is completed with homogeneous Neumann boundary conditions on $\partial\Omega$ and initial conditions at time $t = 0$

$$\begin{cases} [\mathrm{GLC}](x, t=0) = a_0(x) & x \in \Omega \\ [\mathrm{ATP}](x, t=0) = b_0(x) & x \in \Omega \\ [\mathrm{ADP}](x, t=0) = c_0(x) & x \in \Omega \\ [\mathrm{GLY}](x, t=0) = d_0(x) & x \in \Omega \\ [\mathrm{PYR}](x, t=0) = e_0(x) & x \in \Omega \\ [\mathrm{LAC}](x, t=0) = f_0(x) & x \in \Omega. \end{cases} \tag{7}$$

where $a_0, b_0, c_0, d_0, e_0, f_0 : \Omega \to \mathbb{R}$.

Precise questions on global existence of solutions of reaction–diffusion systems are still an open problem, we refer the reader to [56]. In this work, we consider all diffusion coefficients $D_{[\cdot]}$ equals, ensuring that a global solution of the system (6) with initial condition (7) exists.

## Weak formulation of governing equations

In this section we convert the strong form of the PDEs in Eq. (6) into a corresponding weak form. This is necessary step in order to discretise both the FEM and CutFEM methods. For further details see e.g. [57].

We define the standard Hilbert space $V = v \in H^1(\Omega)$ and we denote with $\overline{V} = V \times V \times V \times V \times V \times V$ the product space.

The weak form of system (6) can be found multiplying each equation by a test function $v_1, v_2, v_3, v_4, v_5, v_6 \in \overline{V}$ and integrating over the space domain $\Omega$

$$\begin{cases}
\int_\Omega \frac{\partial[\mathrm{GLC}]}{\partial t}\, v_1\,\mathrm{d}x = & \int_\Omega D_{[\mathrm{GLC}]}\nabla^2[\mathrm{GLC}]\, v_1\,\mathrm{d}x - \int_\Omega K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}}\, v_1\,\mathrm{d}x \\
& + \int_\Omega f\, v_1\,\mathrm{d}x \\
\int_\Omega \frac{\partial[\mathrm{ATP}]}{\partial t}\, v_2\,\mathrm{d}x = & \int_\Omega D_{[\mathrm{ATP}]}\nabla^2[\mathrm{ATP}]\, v_2\,\mathrm{d}x - \int_\Omega 2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}}\, v_2\,\mathrm{d}x \\
& + \int_\Omega 2K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}}\, v_2\,\mathrm{d}x \\
& + \int_\Omega 28K_{\mathrm{Mito}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{Mito}} v_2\,\mathrm{d}x - \int_\Omega K_{\mathrm{act}}[\mathrm{ATP}]\mathcal{G}_{\mathrm{act}} v_2\,\mathrm{d}x \\
\int_\Omega \frac{\partial[\mathrm{ADP}]}{\partial t}\, v_3\,\mathrm{d}x = & \int_\Omega D_{[\mathrm{ADP}]}\nabla^2[\mathrm{ADP}]\, v_3\,\mathrm{d}x + \int_\Omega 2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}}\, v_3\,\mathrm{d}x \\
& - \int_\Omega 2K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}}\, v_3\,\mathrm{d}x + \int_\Omega K_{\mathrm{act}}[\mathrm{ATP}]\mathcal{G}_{\mathrm{act}}\, v_3\,\mathrm{d}x \\
\int_\Omega \frac{\partial[\mathrm{GLY}]}{\partial t}\, v_4\,\mathrm{d}x = & \int_\Omega D_{[\mathrm{GLY}]}\nabla^2[\mathrm{GLY}]\, v_4\,\mathrm{d}x + \int_\Omega 2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}}\, v_4\,\mathrm{d}x \\
& - \int_\Omega K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}}\, v_4\,\mathrm{d}x \\
\int_\Omega \frac{\partial[\mathrm{PYR}]}{\partial t}\, v_5\,\mathrm{d}x = & \int_\Omega D_{[\mathrm{PYR}]}\nabla^2[\mathrm{PYR}]\, v_5\,\mathrm{d}x + \int_\Omega K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}}\, v_5\,\mathrm{d}x \\
& - \int_\Omega K_{\mathrm{LDH}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{LDH}}\, v_5\,\mathrm{d}x - \int_\Omega K_{\mathrm{Mito}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{Mito}}\, v_5\,\mathrm{d}x \\
\int_\Omega \frac{\partial[\mathrm{LAC}]}{\partial t}\, v_6\,\mathrm{d}x = & \int_\Omega D_{[\mathrm{LAC}]}\nabla^2[\mathrm{LAC}]\, v_6\,\mathrm{d}x + \int_\Omega K_{\mathrm{LDH}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{LDH}}\, v_6\,\mathrm{d}x.
\end{cases} \tag{8}$$

Since $D_{[\cdot]}$ is constant, we apply integration by parts to the second-order spatial derivative

$$-\int_\Omega D_{[\,\cdot\,]}\nabla^2[\,\cdot\,]\, v_i\,\mathrm{d}x = \int_\Omega D_{[\,\cdot\,]}\nabla[\,\cdot\,]\,\nabla v_i\,\mathrm{d}x - \int_{\partial\Omega} D_{[\,\cdot\,]}\frac{\partial[\,\cdot\,]}{\partial n} v_i \mathrm{d}s. \tag{9}$$

Since we have specified pure Neumann boundary conditions for each concentration species, the boundary terms vanish. Leading to

$$-\int_\Omega D_{[\,\cdot\,]}\nabla^2[\,\cdot\,]\, v_i\,\mathrm{d}x = \int_\Omega D_{[\,\cdot\,]}\nabla[\,\cdot\,]\,\nabla v_i\,\mathrm{d}x. \tag{10}$$

We can, then, substitute Eq. (10) into Eq. (8) and use the following compact notation

$$(u, v) = \int_\Omega uv\,\mathrm{d}x. \tag{11}$$

allowing us to state the weak form of the problem problem as: we seek the solutions [GLC],[ATP], [ADP],[GLY], [PYR], [LAC] in the space $\overline{V}$ such that for all $t \in [0, T]$ and for all test functions $v_1, v_2, v_3, v_4, v_5, v_6 \in \overline{V}$

$$
\begin{cases}
\left(\frac{\partial[\mathrm{GLC}]}{\partial t}, v_1\right) = & \Big(-D_{[\mathrm{GLC}]}\nabla[\mathrm{GLC}], \nabla v_1\Big) + \Big(-K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}} \\
& +f, v_1\Big) \\
\left(\frac{\partial[\mathrm{ATP}]}{\partial t}, v_2\right) = & \Big(-D_{[\mathrm{ATP}]}\nabla[\mathrm{ATP}], \nabla v_2\Big) + \Big(-2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}} \\
& +2K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}} + 28K_{\mathrm{Mito}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{Mito}} \\
& -K_{\mathrm{act}}[\mathrm{ATP}]\mathcal{G}_{\mathrm{act}}, v_2\Big) \\
\left(\frac{\partial[\mathrm{ADP}]}{\partial t}, v_3\right) = & \Big(-D_{[\mathrm{ADP}]}\nabla[\mathrm{ADP}], \nabla v_3\Big) + \Big(+2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}} \\
& -2K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}} + K_{\mathrm{act}}[\mathrm{ATP}]\mathcal{G}_{\mathrm{act}}, v_3\Big) \\
\left(\frac{\partial[\mathrm{GLY}]}{\partial t}, v_4\right) = & \Big(-D_{[\mathrm{GLY}]}\nabla[\mathrm{GLY}], \nabla v_4\Big) + \Big(+2K_{\mathrm{HXK}}[\mathrm{GLC}][\mathrm{ATP}]^2\mathcal{G}_{\mathrm{HXK}} \\
& -K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}}, v_4\Big) \\
\left(\frac{\partial[\mathrm{PYR}]}{\partial t}, v_5\right) = & \Big(-D_{[\mathrm{PYR}]}\nabla[\mathrm{PYR}], \nabla v_5\Big) + \Big(+K_{\mathrm{PYRK}}[\mathrm{ADP}]^2[\mathrm{GLY}]\mathcal{G}_{\mathrm{PYRK}} \\
& -K_{\mathrm{LDH}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{LDH}} - K_{\mathrm{Mito}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{Mito}}, v_5\Big) \\
\left(\frac{\partial[\mathrm{LAC}]}{\partial t}, v_6\right) = & \Big(-D_{[\mathrm{LAC}]}\nabla[\mathrm{LAC}], \nabla v_6\Big) + \Big(+K_{\mathrm{LDH}}[\mathrm{PYR}]\mathcal{G}_{\mathrm{LDH}}, v_6\Big).
\end{cases}
\tag{12}
$$

In the subsequent section we discuss the discretization of Eq. (12) using the standard finite element method and the CutFEM.

## Discretization

In order to solve Eq. (12), we must discretise it in both space and time. Discretisation is a process by which continuous mathematical objects (e.g. $v_1$) are transformed into a discrete counterpart that can be manipulated on a computer. We choose to discretise in space using the classical Finite Element Method [58] (FEM) and then using the Cut Finite Element Method (CutFEM) [47]. The FEM results serve as a baseline for comparison of the CutFEM method. For both FEM and CutFEM we discretize in time using a standard finite difference method.

The important distinction between FEM and CutFEM from the point of view of the user is that the FEM requires an extra mesh generation step; a mesh must be generated that conforms to the boundary of the domain, before the simulation can take place. This can be a difficult task, as the mesh must be of sufficiently good quality to ensure an accurate solution, while still conforming to the boundary. In contrast, CutFEM removes the need for a conforming mesh generation step. The boundary of the domain is described implicitly as a level set function that can be extracted directly from e.g. processed image data, or using constructive solid geometry (CSG) [59]. The promise of CutFEM is that discretization of geometry can be performed *automatically* without a mesh generation step that often requires lengthy manual intervention.

We recognise that both the Finite Element Method, and even more so, the CutFEM are highly technical and take some time to understand. The point of this section then is not to give a full and detailed exposition of both of these methods. Instead we aim show a precise derivation of the discrete weak forms and then in the subsequent section we show their translation into the FEniCS Project domain specific language, called the Unified Form Language (UFL) [52]. In practice, if the user can convert their problem into a discrete weak formulation then the the FEniCS Project and the libCutFEM library can automatically perform all of the subsequent discretization steps. For full details of how this process takes places the reader is referred to [16,47].

## FEM

To discretise our problem in time we choose a finite difference approximation, specifically, the backward Euler method [60]. First, we discretize the time interval $[0, T]$, denoting $\Delta t$ the size of the timestep. We denote with subscript $n$ a concentration of (possibly multiple) chemical species at time $t_n$, where $0 \leq n < N$ with $N = T/\Delta t$ is an integer that counts the time steps. The backward Euler method then approximates the continuous time derivative as

$$\frac{\partial}{\partial t}[\,\cdot\,] \approx \frac{[\,\cdot\,]_{n+1} - [\,\cdot\,]_n}{\Delta t}. \tag{13}$$

We discretize the spatial domain $\Omega$ using a triangulation $\mathcal{T}_h$ that conforms (matches) the exact boundary $\partial\Omega$. On this mesh we define the space of piece wise Lagrange finite elements of degree one as $V$ and with $\overline{V}$ the product $\overline{V} = V \times V \times V \times V \times V \times V$. The space $\overline{V}$ can then be used to discretize the weak form equations (12).

We denote $U = ([\mathrm{GLC}], [\mathrm{ATP}], [\mathrm{ADP}], [\mathrm{GLY}], [\mathrm{PYR}], [\mathrm{LAC}])$ the vector of all the concentration solutions and $v = (v_1, v_2, v_3, v_4, v_5, v_6) \in \overline{V}$ the corresponding vector of test functions. The initial conditions are denoted $[U]_0$. We then solve a sequence of problems: find $[U]_{n+1} \in \overline{V}$ for $n = 0, \ldots, N-1$ such that

$$f_h([U]_{n+1}, [U]_n; v) = 0 \quad \forall v \in \overline{V}, \tag{14}$$

where

$$\begin{aligned}
f_h([U]_{n+1}, [U]_n; v) = \int_\Omega \Big( & \Delta t^{-1}([\mathrm{GLC}]_{n+1} - [\mathrm{GLC}]_n)v_1 + D_{[\mathrm{GLC}]}\nabla[\mathrm{GLC}]_{n+1}\nabla v_1 + \\
& - f v_1 + K_{\mathrm{HXK}}[\mathrm{GLC}]_{n+1}[\mathrm{ATP}]_{n+1}^2 v_1 \mathcal{G}_{\mathrm{HXK}} + \\
& + \Delta t^{-1}([\mathrm{ATP}]_{n+1} - [\mathrm{ATP}]_n)v_2 + D_{[\mathrm{ATP}]}\nabla[\mathrm{ATP}]_{n+1}\nabla v_2 + \\
& + 2K_{\mathrm{HXK}}[\mathrm{GLC}]_{n+1}[\mathrm{ATP}]_{n+1}^2 v_2 \mathcal{G}_{\mathrm{HXK}} + \\
& - 2K_{\mathrm{PYRK}}[\mathrm{GLY}]_{n+1}[\mathrm{ADP}]_{n+1}^2 v_2 \mathcal{G}_{\mathrm{PYRK}} + \\
& - 28K_{\mathrm{Mito}}[\mathrm{PYR}]_{n+1} v_2 \mathcal{G}_{\mathrm{Mito}} + K_{\mathrm{act}}[\mathrm{ATP}]_{n+1} v_2 \mathcal{G}_{\mathrm{act}} + \\
& + \Delta t^{-1}([\mathrm{ADP}]_{n+1} - [\mathrm{ADP}]_n)v_3 + D_{[\mathrm{ADP}]}\nabla[\mathrm{ADP}]_{n+1}\nabla v_3 \\
& - 2K_{\mathrm{HXK}}[\mathrm{GLC}]_{n+1}[\mathrm{ATP}]_{n+1}^2 v_3 \mathcal{G}_{\mathrm{HXK}} + \\
& + 2K_{\mathrm{PYRK}}[\mathrm{GLY}]_{n+1}[\mathrm{ADP}]_{n+1}^2 v_3 \mathcal{G}_{\mathrm{PYRK}} - K_{\mathrm{act}}[\mathrm{ATP}]_{n+1} v_3 \mathcal{G}_{\mathrm{act}} \\
& + \Delta t^{-1}([\mathrm{GLY}]_{n+1} - [\mathrm{GLY}]_n)v_4 + D_{[\mathrm{GLY}]}\nabla[\mathrm{GLY}]_{n+1}\nabla v_4 + \\
& - 2K_{\mathrm{HXK}}[\mathrm{GLC}]_{n+1}[\mathrm{ATP}]_{n+1}^2 v_4 \mathcal{G}_{\mathrm{HXK}} + \\
& + K_{\mathrm{PYRK}}[\mathrm{GLY}]_{n+1}[\mathrm{ADP}]_{n+1}^2 v_4 \mathcal{G}_{\mathrm{PYRK}} \\
& + \Delta t^{-1}([\mathrm{PYR}]_{n+1} - [\mathrm{PYR}]_n)v_5 + D_{[\mathrm{PYR}]}\nabla[\mathrm{PYR}]_{n+1}\nabla v_5 + \\
& - K_{\mathrm{PYRK}}[\mathrm{GLY}]_{n+1}[\mathrm{ADP}]_{n+1}^2 v_5 \mathcal{G}_{\mathrm{PYRK}} + \\
& + K_{\mathrm{LDH}}[\mathrm{PYR}]_{n+1} v_5 \mathcal{G}_{\mathrm{LDH}} + K_{\mathrm{Mito}}[\mathrm{PYR}]_{n+1} v_5 \mathcal{G}_{\mathrm{Mito}}
\end{aligned}$$

$$+ \Delta t^{-1}([\mathrm{LAC}]_{n+1} - [\mathrm{LAC}]_n)v_6 + D_{[\mathrm{LAC}]}\nabla[\mathrm{LAC}]_{n+1}\nabla v_6 + \\ - K_{\mathrm{LDH}}[\mathrm{PYR}]_{n+1}v_6\mathcal{G}_{\mathrm{LDH}}\Big)\,\mathrm{d}x. \tag{15}$$

Since Eq. (14) is a non-linear function of the unknown solution $[U]_{n+1}$ we choose to use a Newton–Raphson type algorithm to solve it. This Newton–Raphson algorithm requires the computation of the derivative of $f_h$ with respect to $U$ (commonly called the Jacobian of $f_h$). We do not perform this step manually, but instead use the automatic differentiation capabilities of UFL [52], as shown in "Implementation" section.

### CutFEM

Instead of discretizing the spatial domain $\Omega$ using a mesh that conforms to the boundary, in CutFEM, the problem domain $\Omega$ is described by a level-set function. The level set function is a scalar function on $\mathbb{R}^d$, such that $\phi(x) < 0$ for $x \in \Omega$, $\phi(x) > 0$ for $x \notin \Omega$ and $\phi(x) = 0$ for $x \in \partial\Omega$. We then cover the domain $\Omega$ by a regular background mesh $\Lambda$ ($\Omega \subseteq \Lambda$) of simple shape, e.g. a box containing $\Omega$ meshed with a uniform triangulation. Let $K$ denote a triangle/tetrahedron in this triangulation. Now, let $\tilde{\mathcal{T}}_h$ be the fictitious domain mesh composed by all elements $K \in \Lambda$ such that $K \cap \Omega \neq 0$ ($\Omega \subseteq \tilde{\mathcal{T}}_h \subseteq \Lambda$).

Furthermore, the union of all elements in $\tilde{\mathcal{T}}_h$ is called the fictitious domain $\tilde{\Omega}$. We denote with

$$\mathcal{G}_h^* := \{K \in \tilde{\mathcal{T}}_h : K \cap \partial\Omega \neq \emptyset\}$$

the set of elements intersected by the interface, and we define the set of so-called ghost penalty facets [46] (see Fig. 2)

$$\mathcal{F}_G = \{F \text{ facet in } \tilde{\mathcal{T}}_h : F = K \cap K' \quad \text{where} \quad K \in \mathcal{G}_h^* \quad \text{or} \quad K' \in \mathcal{G}_h^*\}.$$

The stabilisation term introduced in the next section will be applied to this subset of facets. We consider the space

$$W = \{v \in C^0(\tilde{\Omega}) : v|_K \in P^1(K), \forall K \in \tilde{\mathcal{T}}_h\}$$

and the jump gradient is defined for all facet $F$ and $v \in W$ by $[\![\partial_{n_F} v]\!] = n_F \cdot \nabla v|_K - n_F \cdot \nabla|_{K'}$ where $n_F$ denotes the unit normal to $F$ in fixed but arbitrary direction. We use the same notation as the previous section, denoting the solution $U = ([\mathrm{GLC}], [\mathrm{ATP}], [\mathrm{ADP}], [\mathrm{GLY}], [\mathrm{PYR}], [\mathrm{LAC}])$, $v = (v_1, v_2, v_3, v_4, v_5, v_6)$ and the product space $\overline{W} = W \times W \times W \times W \times W \times W$, We then solve a sequence of problems: find $[U]_{n+1} \in \overline{V}$ for $n = 0, \ldots, n-1$ such that

$$f_h([U]_{n+1}, [U]_n; v) + j([U]_{n+1}, v) = 0 \quad \forall v \in \overline{W} \tag{16}$$

where $f_h([U]_{n+1}, [U]_n; v)$ is identical to the standard FEM Equation (15). Here, $j([U]_{n+1}, v)$ denotes the following stabilization terms

$$j([U]_{n+1}, v) = \sum_{F \in \mathcal{F}_G} \big((\gamma h D_{[\mathrm{GLC}]}[\![\partial_{n_F}[\mathrm{GLC}]_{n+1}]\!], [\![\partial_{n_F} v_1]\!])_F \tag{17}$$

$$+(\gamma h D_{[\mathrm{ATP}]}[\![\partial_{n_F}[\mathrm{ATP}]_{n+1}]\!], [\![\partial_{n_F} v_2]\!])_F \tag{18}$$

$$+(\gamma h D_{[\mathrm{ADP}]}[\![\partial_{n_F}[\mathrm{ADP}]_{n+1}]\!], [\![\partial_{n_F} v_3]\!])_F \tag{19}$$

$$+(\gamma h D_{[\mathrm{GLY}]}[\![\partial_{n_F}[\mathrm{GLY}]_{n+1}]\!], [\![\partial_{n_F} v_4]\!])_F \tag{20}$$

$$+(\gamma h D_{[\mathrm{PYR}]} [\![\partial_{n_F} [\mathrm{PYR}]_{n+1}]\!], [\![\partial_{n_F} v_5]\!])_F \tag{21}$$

$$+(\gamma h D_{[\mathrm{LAC}]} [\![\partial_{n_F} [\mathrm{LAC}]_{n+1}]\!], [\![\partial_{n_F} v_6]\!])_F) \tag{22}$$

inspired by [47]. Here, $\gamma$ is a positive penalty parameter. These stabilization terms extend the solution from the physical domain $\Omega$ onto the fictitious domain $\tilde{\Omega}$, it is consistent with the continuous system. They prevent ill-conditioning of the system matrices in case only small parts of $\Omega$ are contained in an element near the boundary $\partial\Omega$. This stabilisation is critical for the robustness and reliability of CutFEM.

## Implementation

The finite element method discretization has been implemented using Python with the open source finite element solver DOLFIN from the FEniCS Project, see [15,16]. The CutFEM discretization has been implemented using Python using the libCutFEM library [47] which builds on top of DOLFIN and the rest of the FEniCS Project. In this section we show *parts* of the code for the libCutFEM example that highlight the close link between the mathematics and the concrete computer implementation. The standard FEniCS Project implementation is similar so we have chosen to show only the libCutFEM implementation for reasons of brevity. The reader should refer to the free online repository for two full working examples that are around 250 lines of code each. The precise problem setup and results from this example are shown in "Two-dimensional non-Lipschitz domain" section.

We import the `dolfin` and `cutfem` Python modules. These two modules contain all of the functionality we need to solve the problem using the CutFEM approach.

```
from dolfin import *
from cutfem import *
```

We create the background mesh $\Lambda$ and then define a level set function describing the heart-shaped domain.

```
# Create background mesh
bg_mesh = RectangleMesh(-1, -1, 1., 2., N, N)
...
# Define heart-shaped level set function
level_set = Expression('x[0] < 0. ? pow(x[1] - sqrt(-x[0]), 2)-1 + pow
    (x[0], 2): pow(x[1] - sqrt(x[0]), 2) - 1 + pow(x[0], 2)', degree
    =2)
```

In the next part of the code, we use two special libCutFEM methods to 1. create a special cut `mesh`, and 2. a `mesh_cutter` that intersects the mesh with the level set and computes the distinct sets of cells (i.e. those that are on the inside of the level set and those that are outside).

```
# Define fictitious domain
mesh = CutFEMTools_fictitious_domain_mesh(bg_mesh, level_set, 0, 0)
...
# Compute mesh to levelset intersection and corresponding marker
mesh_cutter = MeshCutter(mesh, level_set)
```

With these special objects in hand we can define special CutFEM-specific UFL `Measure` objects that will subsequently allow us to write the residual and the stabilisation weak forms. Simply put, a measure defines regions of the problem mesh (cells, edges, parts inside the level set, and parts outside etc.) on which FEniCS will integrate different parts of a weak form.

```
# Measure on all the cells \upd{having an intersection with the
    levelset
# Marker 0 if the cell have an intersection with \{ level set < 0 \}
    and 1 otherwise}
dx = Measure("dx")[mesh_cutter.domain_marker()]
\upd{# Measure on the fictitious boundary defined by the level set
# Marker 0 if the facet have an intersection with \{ level set < 0 \}
    and 1 otherwise}
dS = Measure("dS")[mesh_cutter.interior_facet_marker(0)]
# Measure on the interior part of cells cut by the level set
dxq = dc(0, metadata={"num_cells": 1})

# We combine the measure on the inside of the domain and the measure
# on the interior part of the cut cells
dxc = dx(0) + dxq
```

We create a finite element function space $\bar{V}$ that we can use to further define the UFL algebraic objects that we need.

```
V = FunctionSpace(mesh, "P", 1)
V_bar = MixedFunctionSpace([V, V, V, V, V, V])
v = TestFunction(V_bar)
v_1, v_2, v_3, v_4, v_5, v_6 = v.split()
```

Now we have everything that we need to write the weak residual $f_h([U]_{n+1}, [U]_n; v)$ in Eq. (14) using the UFL. We use the integration measure `dxc` which indicates the integration on cells inside the domain $\Omega$ and in parts of the cut cells inside $\Omega$. This form would look identical in the standard FEniCS Project code except that we would use the `dx` measure that denotes integration over all cells of the mesh.

```
...
u = Function(V_bar)
u_n = project(u_0, V_bar) # Initial condition, details not shown
# Solution at next timestep
a, b, c, d, e, f = u.split() # GLC, ATP, ADP, GLY, PYR, LAC
# Solution at current timestep
a_n, b_n, c_n, d_n, e_n, f_n = u_n.split()

F = ((a - a_n) / k)*v_1*dxc + D_glc*dot(grad(a), grad(v_1))*dxc +
    K_hxk*a*b**2*v_1*g_hxk*dxc - f_glc*v_1*dxc + ((b - b_n) / k)*v_2*
    dxc + D_atp*dot(grad(b), grad(v_2))*dxc + 2*K_hxk*a*b**2*v_2*g_hxk
    *dxc - 2*K_pyrk*d*c**2*v_2*g_pyrk*dxc - 28*K_Mito*e*v_2*g_mito*dxc
     + K_act*b*v_2*g_act*dxc + ((c - c_n) / k)*v_3*dxc + D_adp*dot(
    grad(c), grad(v_3))*dxc - 2*K_hxk*a*b**2*v_3*g_hxk*dxc + 2*K_pyrk*
    d*c**2*v_3*g_pyrk*dxc - K_act*b*v_3*g_act*dxc + ((d - d_n) / k)*
    v_4*dxc + D_gly*dot(grad(d), grad(v_4))*dxc - 2*K_hxk*a*b**2*v_4*
    g_hxk*dxc + K_pyrk*d*c**2*v_4*g_pyrk*dxc + ((e - e_n) / k)*v_5*dxc
     + D_pyr*dot(grad(e), grad(v_5))*dxc - K_pyrk*d*c**2*v_5*g_pyrk*
    dxc + K_ldh*e*v_5*g_ldh*dxc + K_Mito*e*v_5*g_mito*dxc + ((f - f_n)
     / k)*v_6*dxc + D_lac*dot(grad(f), grad(v_6))*dxc - K_ldh*e*v_6*
    g_ldh*dxc
```

The stabilization term $j([U]_{n+1}, v)$, Eq. (17) is written in UFL using the integration measure `dS(1)` which integrates on the ghost penalty facets $\mathcal{F}_G$ (Fig. 4).

```
j = avg(gamma)*avg(h)*D_glc*dot(jump(grad(a), n), jump(grad(v_1), n))*
    dS(1) + avg(gamma)*avg(h)*D_atp*dot(jump(grad(b), n), jump(grad(
    v_2), n))*dS(1) + avg(gamma)*avg(h)*D_adp*dot(jump(grad(c), n),
    jump(grad(v_3), n))*dS(1) + avg(gamma)*avg(h)*D_gly*dot(jump(grad(
    d), n), jump(grad(v_4), n))*dS(1) + avg(gamma)*avg(h)*D_pyr*dot(
    jump(grad(e), n), jump(grad(v_5), n))*dS(1) + avg(gamma)*avg(h)*
    D_lac*dot(jump(grad(f), n), jump(grad(v_6), n))*dS(1)
```

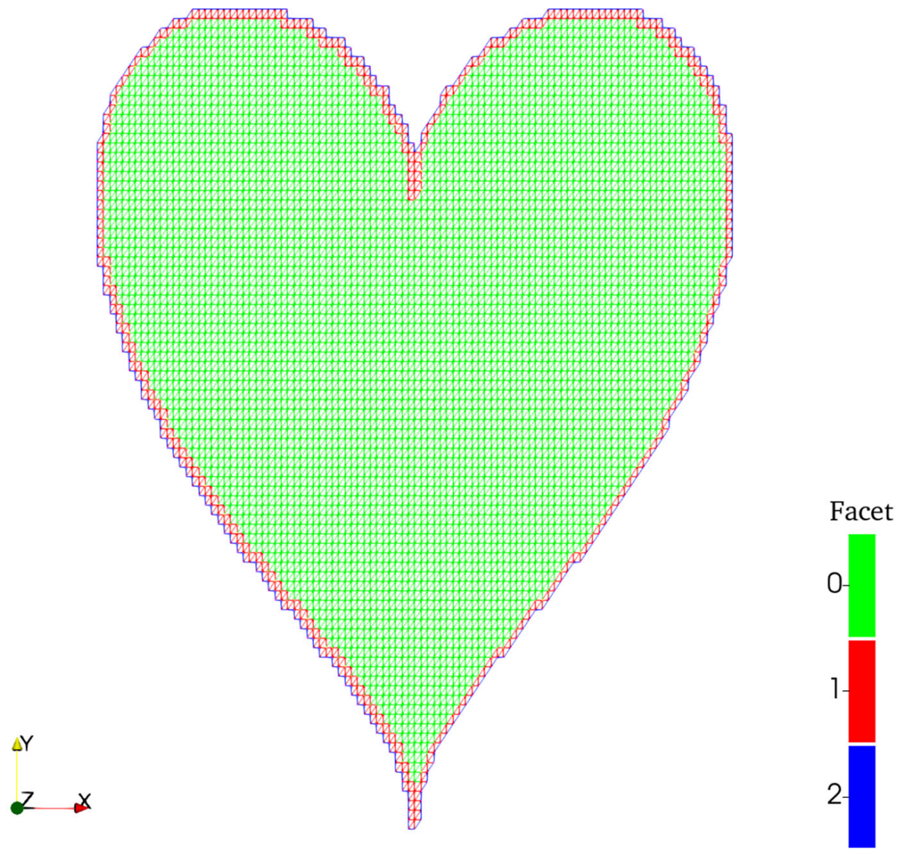


**Fig. 4** We can see the fictitious domain $\tilde{\Omega}$ for the heart shaped function, and the facets marked. In red the set of facets $\mathcal{F}_G$ where we apply the stabilization term $j([U]_{n+1}, v)$

For both forms F and j we remark how similar the UFL notation is to the mathematical notation in Eqs. (14) and (17). Calculating the UFL expression for the Jacobian $\mathbb{J}$ can be performed automatically using the `derivative` function.

```
F += j
J = derivative(F, u)
```

The last step before the solver is to use the Composite framework of CutFEM to define the problem on different parts of the mesh domain.

```
# Fictitious domain
composite_mesh = CompositeMesh()
composite_mesh.add(mesh)

W = CompositeFunctionSpace(composite_mesh)
W.add(V_bar);
W.build();

# Constrain dofs outside
FidoTools_compute_constrained_dofs(W, mesh_cutter)

a = FidoForm(W,W)
form_a = create_dolfin_form(J)
a.add(form_a, mesh_cutter)

L = FidoForm(W)
form_L = create_dolfin_form(F)
L.add(form_L, mesh_cutter);
```

Eventually, we can solve the non-linear and time-dependent problem with the following two nested loops, the outer one for the timestepping, and the inner one for the Newton–Raphson algorithm. The FEniCS Project infrastructure (UFL + FFC + DOLFIN) automatically generates and executes low-level code to assemble the sparse matrices A and b. This linear system is then solved using PETSc and MUMPS.

```
...
# Timestep loop
max_iters = 50
toll_a = 1.e-10
for i in range(num_steps):
    j = 0
    # Newton-Raphson algorithm
    while j < max_iters:
        A = composite_assemble(a)
        b = composite_assemble(L)

        uc = CompositeFunction(W)

        # Solve linear system for Newton step using MUMPS direct
    solver
        solve(A, uc.vector(), -b, 'mumps')

        # Newton update
        u.vector().axpy(1.0, uc.part(0).vector())

        # Terminate if tolerance reached
        if uc.part(0).vector().norm("l2") < toll_a:
            break
        else:
            j += 1

    ...
    # Update the solution
    u_n._assign(u)
    # Update the time step
    t[0] = t[0] + dt
    # Update the source term
    f_glc.t = t[0]
```

In the previous piece of code, we used the increment as stopping criteria for the Newton–Raphson algorithm. We refer the reader to the free online repository with an example where the stopping criteria is the tolerance to the residual. In the full example in the free online repository the solution at each timestep is outputted to a VTK file that can be opened with Paraview (https://paraview.org) for visualisation.

## Numerical results

In this section we evaluate the accuracy of the CutFEM discretisation scheme for different geometries by comparing the CutFEM solution to the standard FEM solution. In addition, we want to confirm that at the steady state solutions (for large values of time $t$) predicted by CutFEM tend towards the asymptotic solutions of the associated ordinary differential equation (ODE) system. Then, we investigate the accuracy of CutFEM in comparison with the standard FEM for a simple circular geometry. We increase the complexity of the geometry and we show the ability of CutFEM to straightforwardly solve a test case within a non-Lipschitz domain. Lastly, we consider a three dimensional domain.

The numerical scheme was implemented using the CutFEM library [47] based on FEniCS Project [15–17]. The linear systems arising in the numerical experiments are solved using a direct (MUMPS) solver for the two-dimensional examples and a iterative (CG) solver with algebraic multigrid preconditioning (hypre) for the three-dimensional example.

### Asymptotic solution ODEs

The aim of this section is to validate our CutFEM implementation highlight that the solution of the reaction–diffusion system tends to the solution of the ODE system associated to the chemical reactions Eqs. (1)–(5) for time going to infinity. The solution of the ODE system is computed in two ways 1. we use the *solve_ivp* of the package *scipy* in python 2. we manually compute the asymptotic solutions, which can be found in Appendix A together with the ODE system in Eq. (24).

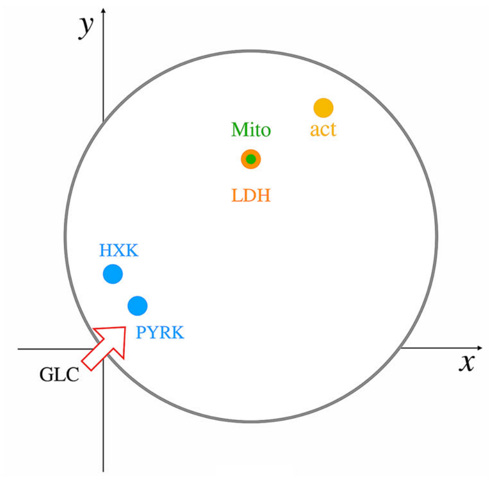
For the PDEs, we solve the reaction diffusion system in a circular domain defined in CutFEM using the level set function $\phi(x, y) = (x-4)^2 + (y-3)^2 - 25$. The Gaussian parameters locating the chemical reactions, that are shown in Fig. 5a, have been set as following: $\mathcal{G}_{\text{HXK}}(x_0 = 0.5, y_0 = 2.0, \sigma = 0.1)$, $\mathcal{G}_{\text{PYRK}}(x_0 = 1.1, y_0 = 1.2, \sigma = 0.1)$, $\mathcal{G}_{\text{LDH}}(x_0 = 4.0, y_0 = 5.0, \sigma = 0.1)$, $\mathcal{G}_{\text{Mito}}(x_0 = 4.0, y_0 = 5.0, \sigma = 0.1)$ and $\mathcal{G}_{\text{act}}(x_0 = 6.0, y_0 = 6.5, \sigma = 0.1)$. Note, we have co-localized the reactions (3) and (4), in order to obtain that PYR is used equally in the two reactions, such that we can set the parameter $\alpha$ of the ODE asymptotic solutions of Eq. (26) equal to 0.5.

The influx of GLC entering the cell domain has been set equal to 100 until time $t = 1$ and is entering the cell into a circular subdomain with radius 0.3 and center the origin.

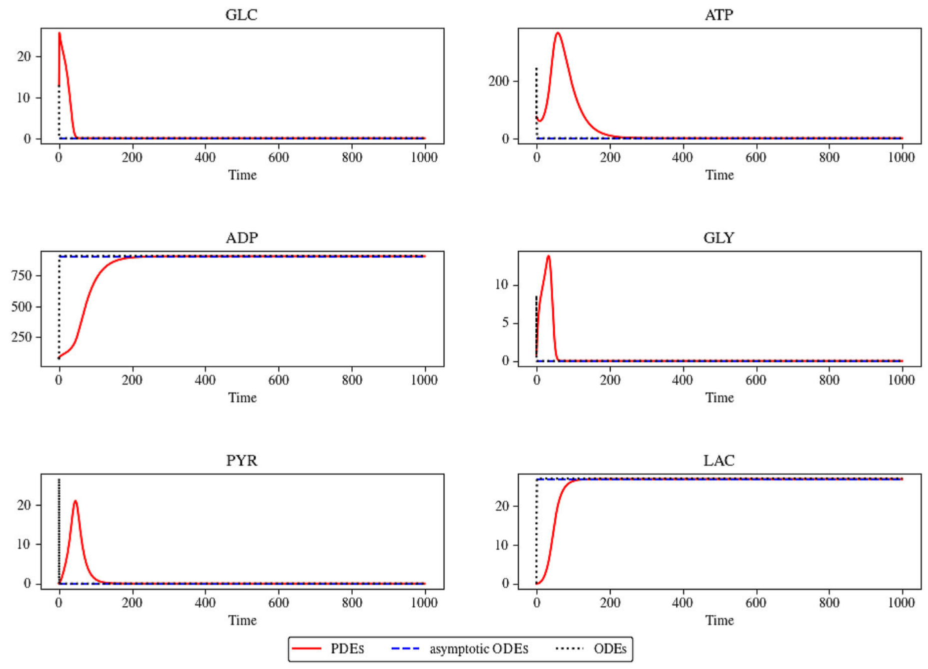
The rate constants $K_{\cdot}$ of the chemical reactions are set equal to 10.0, in order to avoid that one reaction dominates over, and the diffusive parameters $D_{[\cdot]}$ of each chemical species is 100.0 to accelerate the convergence to the ODEs solutions. The initial amount of concentrations inside the cell are set to zero except for [ATP] and [ADP] that are equal to 1. Final time is 1000, as well as the number of time step. The CutFEM penalty parameter $\gamma$ has been set equal to 0.1. The mesh size is set to a maximum diameter of 0.1744. In this experiment, a finer mesh is not required for proving the convergence to the ODEs solutions.

As initial condition, the asymptotic ODEs use the solution of PDEs at time equal to 1 when the source term of glucose stops, which keeps the total amounts of concentration unchanged, as shown in Appendix A.

To use *scipy* to solve the ODEs we set all the parameters as the PDEs, the influx of GLC and the initial conditions are obtained from the PDE ones integrating over the domain.



a



b

**Fig. 5** **a** The locations of the chemical reactions Eqs. (1)–(5) and the influx of GLC are shown in the circular domain in order to prove the convergence of the **PDEs** solutions to the ODEs. Note, the chemical reactions LDH Eq. (3) and Mito Eq. (4) have been placed in the same location. **b** The solutions of the **PDEs** obtained using **CutFEM** and integrated over the domain (red line) are plot in comparison with the asymptotic solutions of the ODEs for each concentration (blue line) and with the ODEs solved using *scipy*. The convergence of the **PDEs** to the ODEs solutions is reached for $t > 200$

**Table 1 In the table shows the maximum cell diameter, the number of DOFs, and the computational time for FEM and CutFEM in a circular domain and for CutFEM for the perturbed boundary domain**

| | Maximum cell diameter | DOF | Computational time |
|---|---|---|---|
| Circle FEM | 0.049999 | 477114 | 4 h 30 m |
| Circle **CutFEM** | 0.049741 | 387786 | 23 h |
| Perturbed circle **CutFEM** | 0.048893 | 402288 | 21 h |

In order to compare the numerical solution of the system (6) solved using CutFEM, we integrate the solutions of the concentrations over the domain $\Omega$ to obtain the average chemical concentration of each species at each time step. We use the following notation

$$\tilde{[\cdot]}(t) := \int_{\Omega} [\cdot](t)\, \mathrm{d}x \quad \forall t \in [0, T]. \tag{23}$$

In Fig. 5b, we plot the solutions obtained using the Formula (23) for the PDEs, the asymptotic ODE solutions obtained using Eq. (26) and the ODE solutions with *scipy*. As expected from the asymptotic solutions computed in Appendix A, the average concentrations of GLC, ATP, GLY and PYR go to zero whilst the products of the system are LAC and ADP. We can see that the ODE solutions with *scipy* converge to the asymptotic solution very quickly, and after time $t = 200$ also the PDE solutions tend to the same values.

**Example one: two-dimensional circular domain**

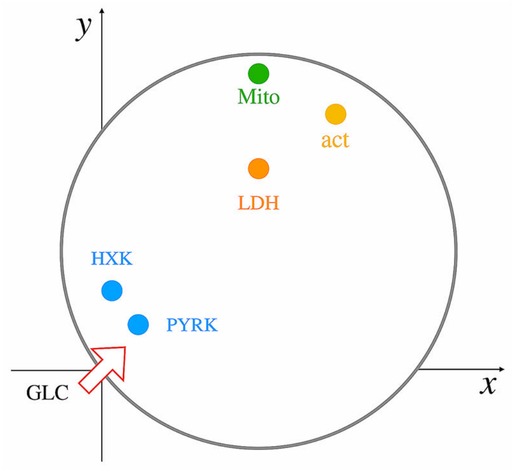
In this section, we assess the accuracy of the CutFEM solution. First, we consider the cell, $\Omega$, as a circle defined by the level set function $\phi(x, y) = (x-4)^2 + (y-3)^2 - 25$ and we show that the results obtained with CutFEM are of the same order as those obtained with FEM, where the circular domain is explicitly meshed using package mshr.

To investigate the behaviour of CutFEM for more complex boundaries we consider an irregular cell geometry defined by a perturbed circle represented by the level set function $\phi(x, y) = (x-4)^2 + (y-3)^2 - \cos(4x)\cos(5x) - \sin(4y)\cos(5y) - 25$. We highlight how it is easy for CutFEM to work with a more complex shape just by changing the level set function.

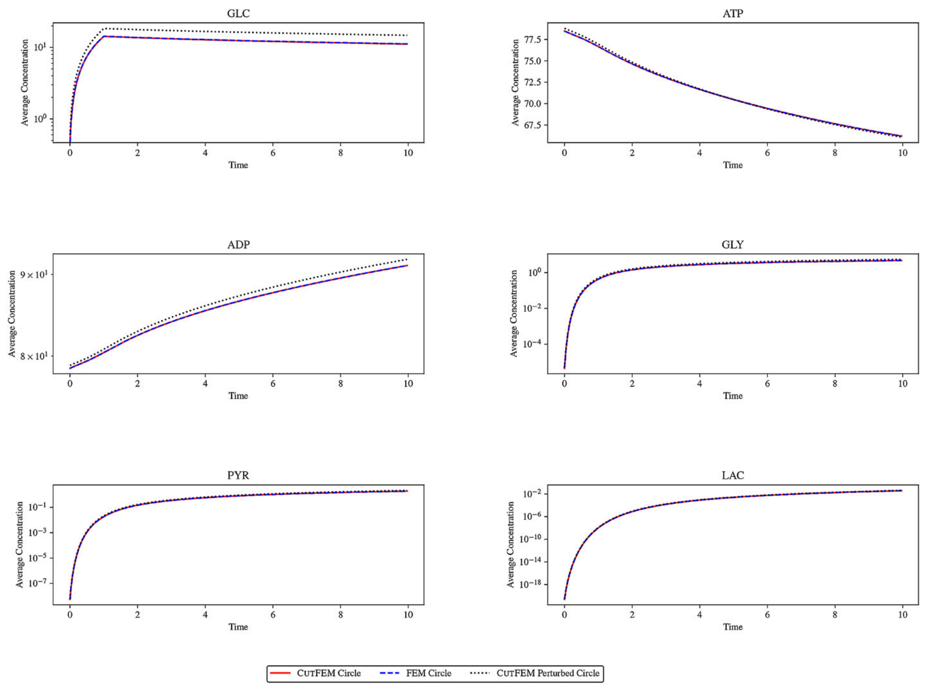
In this experiment the glucose source term and the reaction locations and parameters are set as in section , with the exception of the Mito reaction that is located at $\mathcal{G}_{\mathrm{Mito}}(x_0 = 4.0, y_0 = 7.5, \sigma = 0.1)$. The test case is represented schematically in Fig. 6a.

The chemical species are allowed to diffuse inside the cell with diffusive constants $D_{[\cdot]} = 1.0$ for each species. The rate constants of the chemical reaction and the initial chemical concentrations are set as in the previous test case, see Section . Final time is 10 and we set the number of time steps as 300. As before, the penalty parameter of CutFEM is set to 0.1. The choice of the mesh size is shown in Table 1 in accordance with the size of the Gaussian parameters $\sigma$.

In Fig. 6b, we investigate the average concentrations inside the three domains at each time step using Eq. (23). The average concentrations for each species computed in the circular domain with FEM and CutFEM show equivalent results. The solutions obtained using the perturbed domain show higher average concentration of GLC and ADP than the results in the circular domain. We remind the reader that the system solved inside the



**a**



**b**

**Fig. 6** **a** The chosen location of the chemical reactions Eqs. (1)–(5) and of the influx of GLC are shown to compare FEM and CutFEMresults in a circular domain. **b** In the y-axis is plot the integral over the domain at each time step using Eq. (23), giving us the average concentration, of CutFEM and FEM in a circular domain and CutFEM solved in a circular domain with perturbed boundary. The average concentration of each species of FEM (blue line) and CutFEM (red line) are equivalent while the CutFEM solutions of the perturbed domain (black line) share the same results for concentrations ATP, GLY, PYR, LAC and differs but show the same trend for GLC and ADP perturbed domain is not supposed to produce same results as the circular one, however the dynamics show the same trend as shown in Fig. 6b.

For time equal to 5, we plot the solutions of the concentrations for each chemical species in Fig. 7. We notice how the Gaussian functions drive the chemical reactions in the regions defined in Fig. 6a. For example, we can notice that ATP is consumed in two regions that locate respectively HXK and act, while ADP is produced. The GLC entered the cell is diffusing in the domain and taking part to the HXK reaction. In comparison, the results of FEM and CutFEM are almost identical.

To further analyze the plot shown in Fig. 7, we compare in Fig. 8 FEM and CutFEM solutions line $x = y$ at time 5 (Fig. 8a) and final time 10 (Fig. 8b). We can recognize where the reactions are from the bump in the curve. The plot indicates that the results are equivalent.

Our experiments suggest that CutFEM produces similar solutions to the FEM but the computational time needed to solve CutFEM is five times as high as standard FEM. The number of DOFs is smaller for CutFEM compared to FEM, as shown in Table 1.

Concerning the larger computational time requires for CutFEM, the CutFEM implementation is based on FEniCS 1.5.0 released in 2015, while the FEM ones on FEniCS 2019.1.0 released in 2019 with significant improvements and computational optimizations. Since CutFEM relies in large parts on the FEniCS functions, its computational time would be highly reduced when using a newer version of FEniCS. Breaking down the total computational time, we identified that the CutFEM implementation of assembly is significantly slower than the FEniCS implementation of assembly. However, critically, we have verified that both implementations scale at the expected optimal rate $O(N)$ in the number of cells $N$ (results not shown). The larger constant in the $O(N)$ scaling in the CutFEM implementation points to certain computational kernels not optimised for efficiency.

In conclusion, in this section we showed how CutFEM and FEM gives equivalent results for a circular domain. Moreover, we investigated the behaviour of CutFEM with a more complex domain, the perturbed circle.

### Two-dimensional non-Lipschitz domain

In this section, we consider an irregular boundary, i.e. where the normal field along the surface is non-smooth. We choose a heart-shaped domain, which has two singularities (this is known as a non-Lipschitz domain).

The heart-shaped is described by the level set function $\phi(x, y) = (y - \sqrt{|x|})^2 - 1 + x^2$. The description of the problem setting is shown in Fig. 9. The gaussian functions are defined as $\mathcal{G}_{\text{HXK}}(x_0 = 0.1, y_0 = -0.5, \sigma = 0.1)$, $\mathcal{G}_{\text{PYRK}}(x_0 = 0.3, y_0 = 10.0, \sigma = 0.1)$, $\mathcal{G}_{\text{LDH}}(x_0 = -0.5, y_0 = 0.5, \sigma = 0.1)$, $\mathcal{G}_{\text{Mito}}(x_0 = 0.5, y_0 = 0.7, \sigma = 0.1)$ and $\mathcal{G}_{\text{act}}(x_0 = 0.0, y_0 = 0.9, \sigma = 0.1)$. We consider a source term for GLC active within a disk of radius 0.3, centered at the cusp and take the GLC influx to be 100.

The diffusive constants, the reaction rate and the initial concentrations are as in "Example one: two-dimensional circular domain" Section, as well as the penalty parameter for the stabilization term.

We solve the reaction diffusion system (6) using CutFEM, where we have set a background mesh with a maximum cell diameter equal to 0.01803, the number of DOFs is 129,606 and the total time to run the simulation was approximately 4 h.

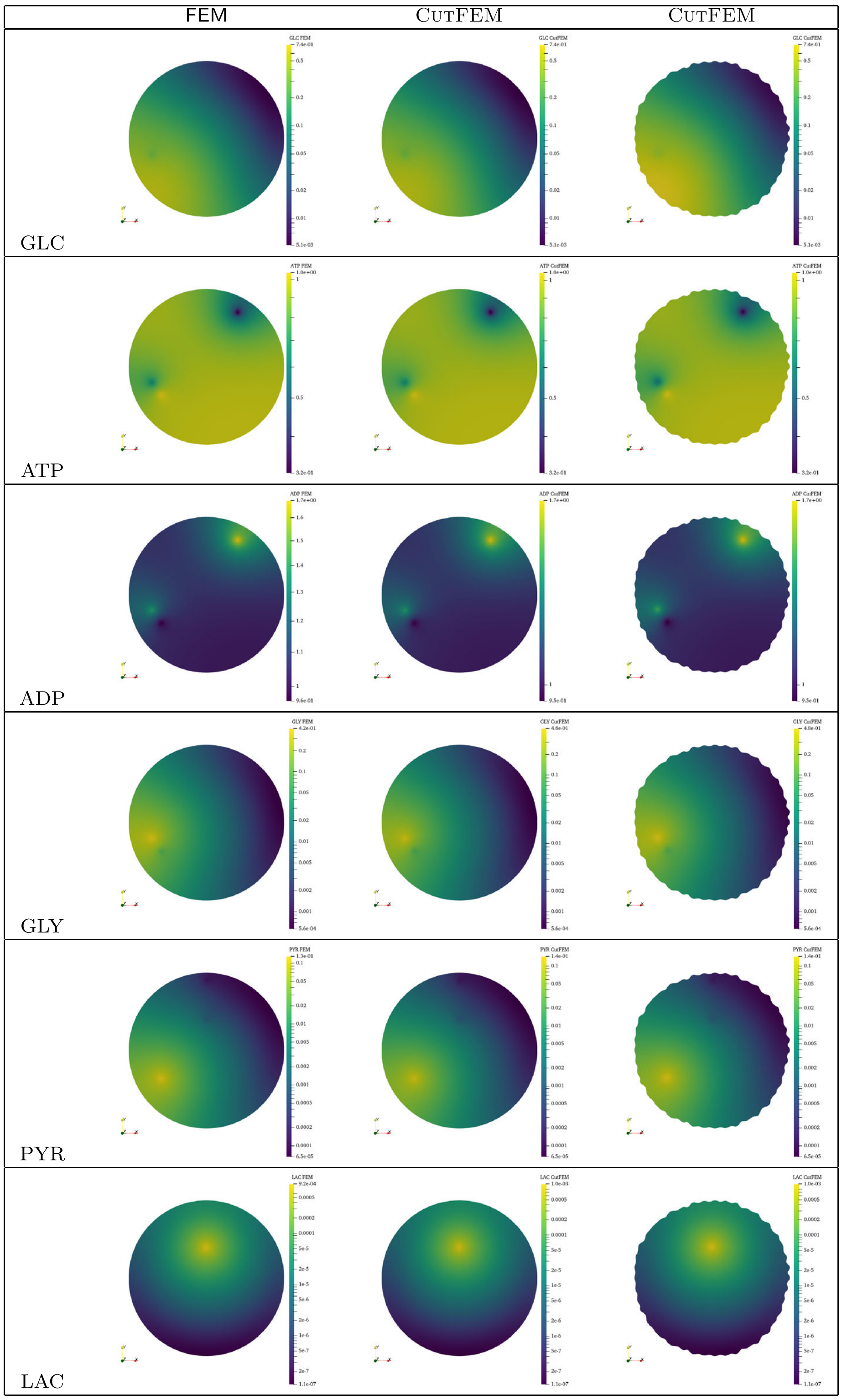


**Fig. 7** Solutions of the concentrations, from top to bottom: GLC, ATP, ADP, GLY, PYR and LAC, at time $t = 5$ in a circular domain solved with FEM (Left), CutFEM (Center) and CutFEM in a perturbed boundary (Right). The source term of GLC is no longer active, and the quantity of GLC is spreading through the domain and participating into the reaction HXK. ATP and ADP show that in the region where HXK, PYRK and act are located, when one is consumed the other is produced. We can notice production of GLY, PYR and LAC. Comparing the results in the three domain, they are extremely close to each other and visually not distinguishable

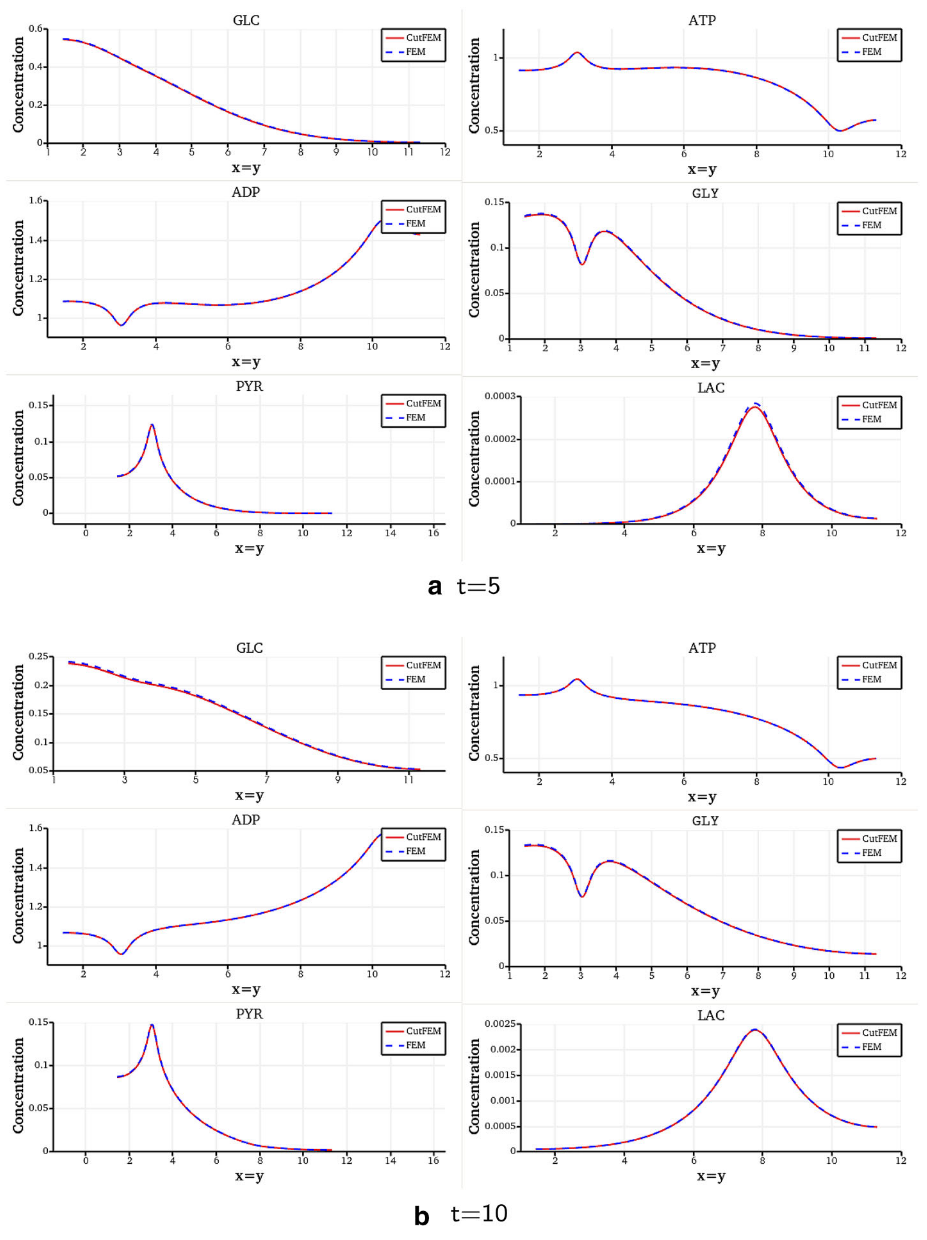


**Fig. 8** We plot the solutions of the concentrations in a circular domain obtained with FEM and CutFEM along the line $x = y$ at time $t = 5$ (**a**) and final time $T = 10$ (**b**). We can notice in the plot the location where the reactions are happening thanks to the bump in the plot. The only small differences detectable are in LAC at $t = 5$ and in GLC at $t = 10$ where CutFEM is slightly below the FEM curve

Figures 10 and 11 show the solutions obtained in the heart-shaped domain for each chemical species at the initial and final time. At the initial time, we can see how the GLC (Fig. 10a) enters the cellular domain, diffuses and takes part in HXK. On the other hand, the region where the reaction act (consuming ATP and producing ADP) is clearly visible (Fig. 10b, c). Observing the plot of GLY (Fig. 10d) we can see the region where GLY is produced by HXK and at the same time consumed by PYRK. Interestingly, we can notice that at the initial time, the PYR produced by PYRK contributes to generating LAC,

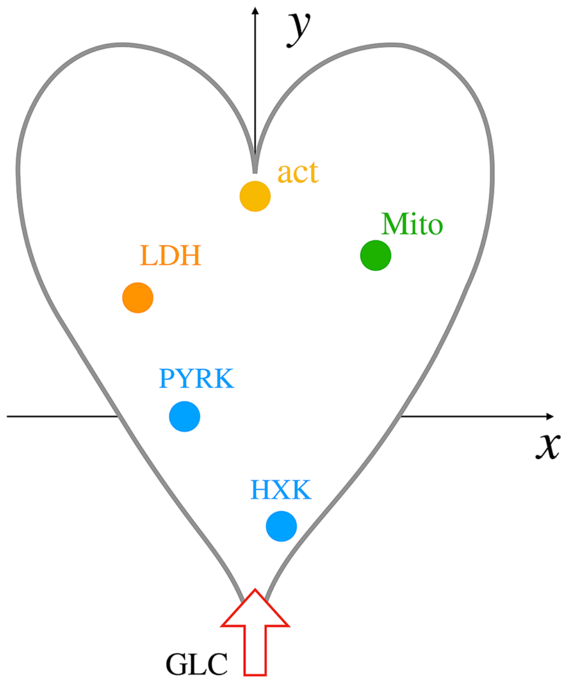


**Fig. 9** Location of the chemical reactions Eqs. (1)–(5) and the source term of GLC in the heart shaped domain

however PYR has not reached the Mito reaction site, so that there is no production of ATP from this reaction (Fig. 10e, f).

At the final time ($T = 10$), the GLC has diffused into the domain, and is still used to generate HXK (Fig. 11a). Concentrations ADP, PYR and LAC have reached a stable state (Fig. 11c, e, f), while ATP is consumed by the act reaction and is produced by the Mito reaction (Fig. 11b). GLY is still consumed by PYRK and produced by HXK (Fig. 11d).

Briefly, we observe that CutFEM solves the test case inside the heart shaped domain, the singularities are well described thanks to the choice of a fine mesh. The computational time of CutFEM in this test case is smaller than in the previous experiment since the number of DOFS is 4 times less then the previous test case.

### Experiment three: 3 dimensional complex domain

For the last experiment, we work with a 3 dimensional domain. We consider a union of six spheres that generate a pop-corn shape. To create the shape we use the union of the following level set functions

$$\begin{aligned}
\phi_1(x, y, z) &= (x-4)^2 + (y-3)^2 + z^2 - 25\\
\phi_2(x, y, z) &= (x-12)^2 + (y-3)^2 + z^2 - 25\\
\phi_3(x, y, z) &= (x-8)^2 + (y-7)^2 + z^2 - 25\\
\phi_4(x, y, z) &= (x-8)^2 + (y+1)^2 + z^2 - 25\\
\phi_5(x, y, z) &= (x-8)^2 + (y-3)^2 + (z+4)^2 - 25\\
\phi_6(x, y, z) &= (x-8)^2 + (y-3)^2 + (z-4)^2 - 25
\end{aligned}$$

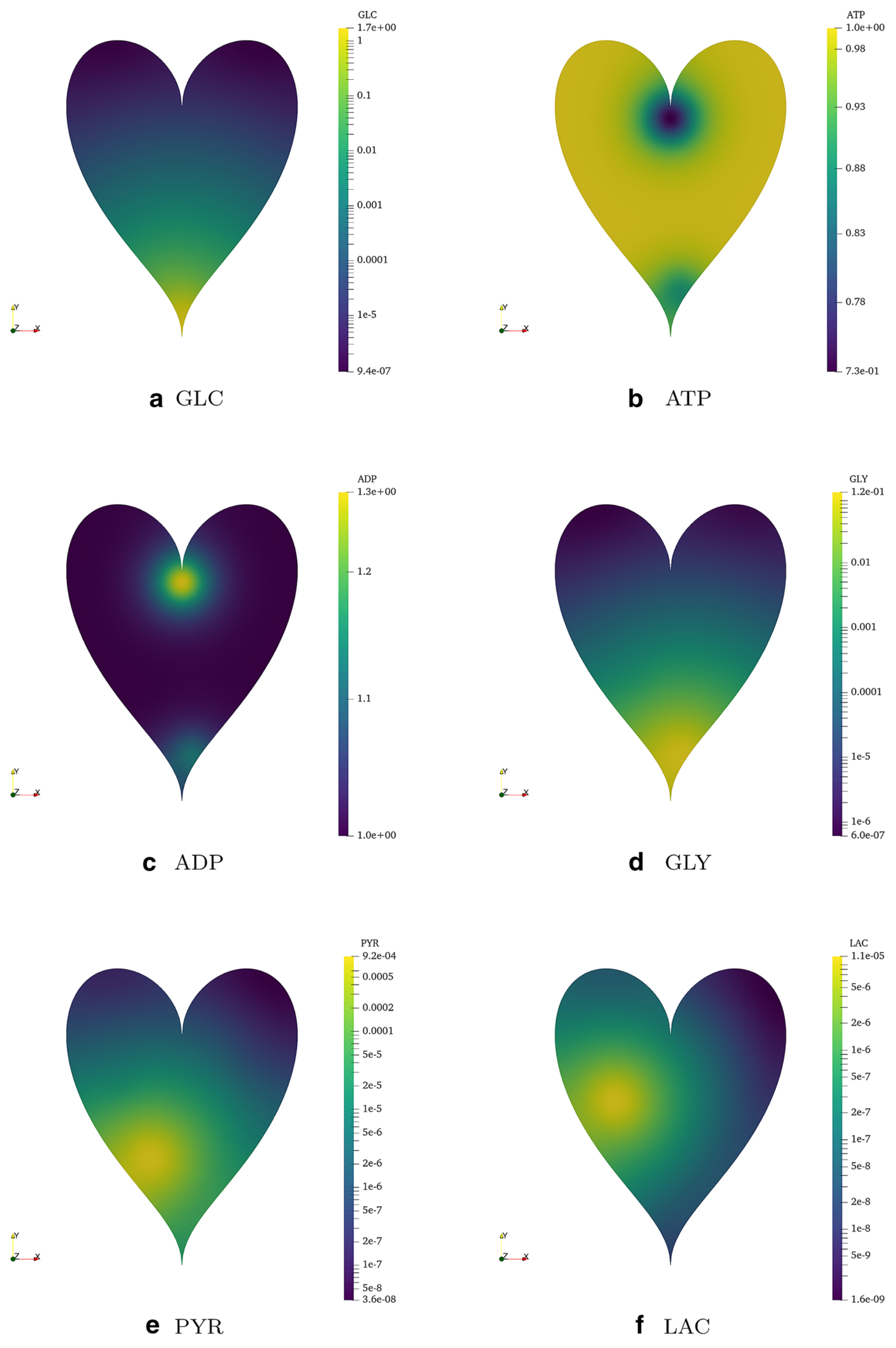


**Fig. 10** Solutions of the Model obtained in a heart-shaped domain at the Initial time ($t = 0$). GLC is entering the domain from the infimum of the domain. ATP is used in the region where act and HXK are located, vice versa ADP is produced. GLY is produced from HXK. We can see PYR and LAC produced by PYRK and LDH, respectively

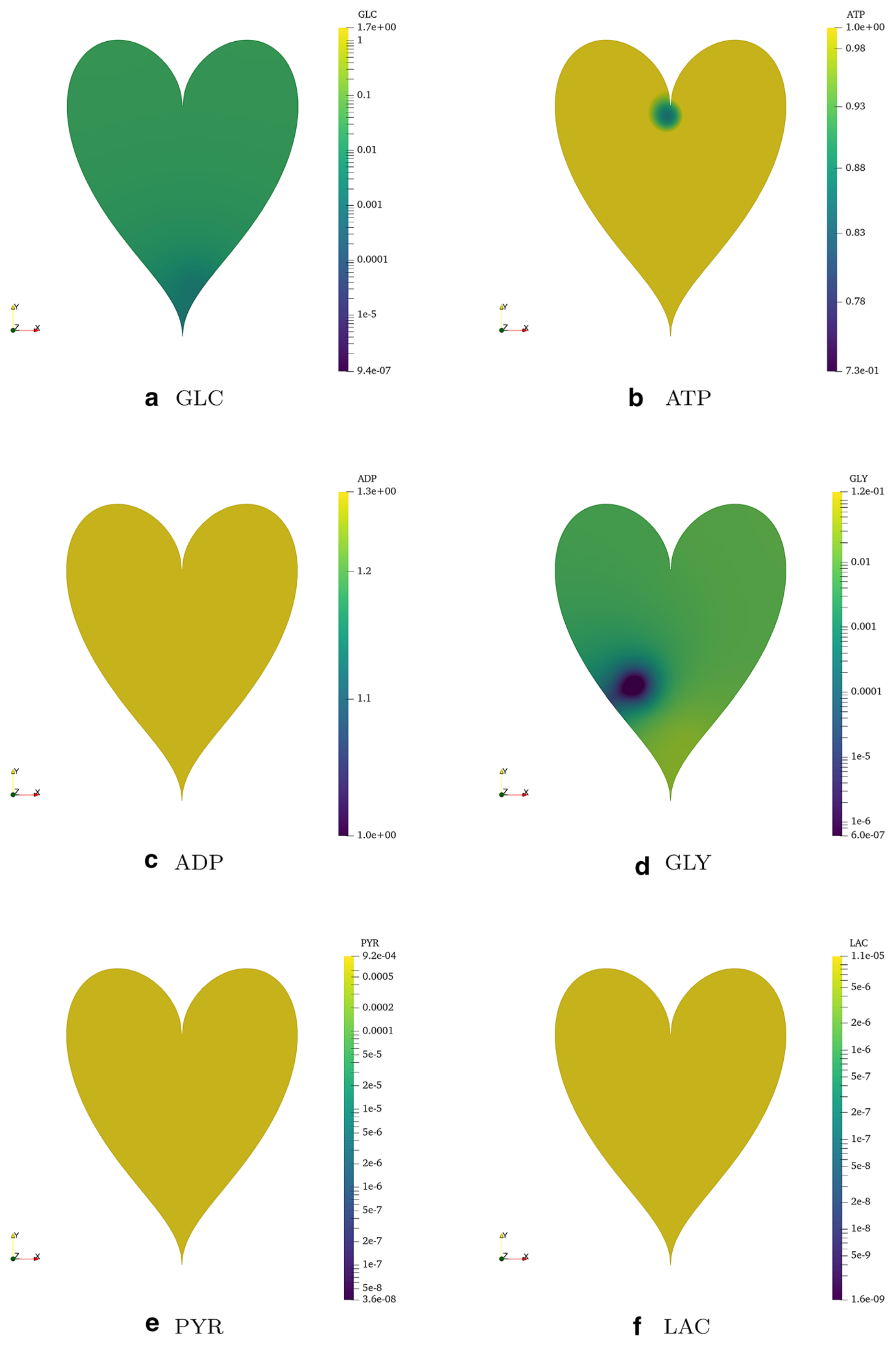
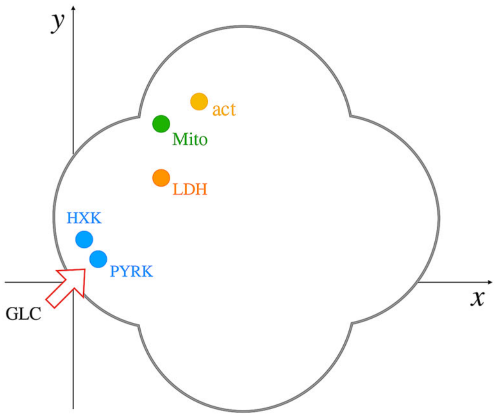
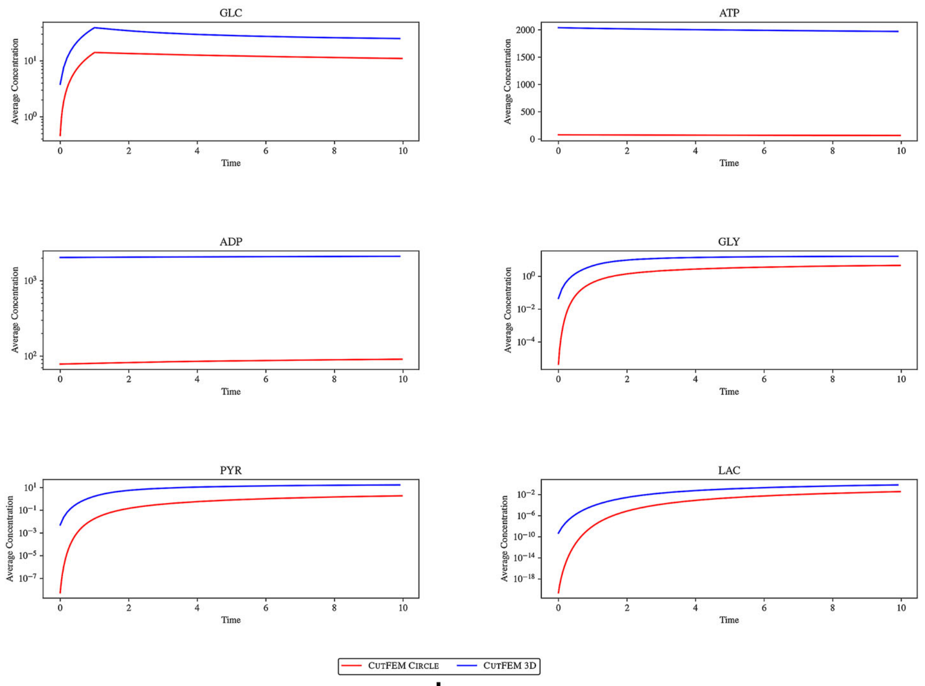


**Fig. 11** Solution of the Model obtained in a heart-shaped domain at the final time (T = 10). From top to bottom from left to right: [GLC], [ATP], [ADP], [GLY], [PYR], [LAC]. GLC is still consumed in the infimum where HXK is. ADP, PYR and LAC have reached a state where the amount of concentration is uniformly distributed into the domain. ATP is consumed to produce ADP by act and produced by the Mito reaction. GLY is consumed by PYRK and produced by HXK



**a**



**b**

**Fig. 12** **a** Setting of chemical reactions Eqs. (1)–(5) and the source term of GLC for the 3D domain, projected into the x–y axes. **b** The plot shows the average concentrations of each species computed in the three-dimensional example (blue line) compared with the one obtained in Fig. 6b with CutFEM in the circle

We sketch in Fig. 12a the problem setting projected onto the $xy$ plane. The parameters chosen for the reaction locations are: $\mathcal{G}_{\mathrm{HXK}}(x_0 = 0.5, y0 = 2.0, z_0 = 0.0, \sigma = 0.5)$, $\mathcal{G}_{\mathrm{PYRK}}(x_0 = 1.1, y_0 = 1.2, z_0 = 0.0, \sigma = 0.5)$, $\mathcal{G}_{\mathrm{LDH}}(x_0 = 4.0, y_0 = 5.0, z_0 = 0.0, \sigma = 0.5)$, $\mathcal{G}_{\mathrm{Mito}}(x_0 = 4.0, y_0 = 7.5, z_0 = 0.0, \sigma = 0.5)$ and $\mathcal{G}_{\mathrm{act}}(x_0 = 6.0, y_0 = 8.5, z_0 = 0.0, \sigma = 0.5)$. The source term is located in a ball of radius 0.3 centered on the origin, and we increase the source influx to 1000 until time 1. The diffusive constants, the reaction rates and the initial conditions are set as in "Asymptotic solution ODEs" section.

Based on the converged mesh sizes obtained for the two-dimensional examples, we set the mesh size to $h = 0.3772$. Note that the choice of the Gaussian parameters $\sigma$ is designed to make the reactions significant since the mesh is coarser than the meshes used in the previous examples. Also, we increase the number of time steps to 100, while the final time is 10. The CutFEM penalty parameter is 0.1.

In Fig. 12b, we plot the average concentration of each species using Eq. (23). We examine the trend of the average concentrations plotting them with the ones computed in "Example one: two-dimensional circular domain" section. Even though the system solved one in 2D and one in 3D gives different results, we expect their average concentrations to behave similarly. Clearly, we have a higher influx of GLC entering the three-dimensional domain, as well as higher initial concentrations of ATP and ADP. As expected, the average concentrations are higher in the three-dimensional domain but they all follow the same trend as the two-dimensional cases.

We show the results obtained in the three-dimensional domain at time 5 in Fig. 13. The results are in accordance with the ones showed in Section . We notice that the species are diffusing inside the domain and the reactions are happening in the location we indicated with the Gaussian functions.

In conclusion, in our three-dimensional experiment we can see how the chain of chemical reactions describe the energy metabolism of a cell in a complex structure. CutFEM is able to work with complex three-dimensional geometries, although the computational time is large (is 3 days in this case).

## Conclusion and discussion

Studying brain energy metabolism can be helpful to further our understanding of several neurodegenerative diseases such as Alzheimer's and Parkinson's but a mechanistic perspective is still lacking. To investigate energy metabolism in biological relevant morphologies, we presented a simplified model of metabolism in a single cell using reaction–diffusion equations and demonstrated how to discretize the model using the FEM and CutFEM numerical methods. CutFEM has the advantage of enabling the use of a single level set function per cell, independent of the complexity of the cell geometry. Implementing using FEniCS and libCutFEM ensures the usability by non-experts. In particular, modifying one or several of the reaction diffusion equations can be made by altering only a few lines of code and any linearisation and parallelisation is done automatically by the FEniCS framework. One of the difficulties of non-conforming methods for implicit domain definition [40,61,62] is due to the ill-conditioning of the system matrices when the interfaces or the boundary of the domain passes close to a node, leading to very large or very small diagonal terms. The consistent stabilisation term used by CutFEM prevents this issue and leads to a stable and convergent scheme. Other approaches are provided in [42–44].

Our results indicate that CutFEM is a valuable approach to deal with biological problems with arbitrarily complex cell morphologies. The appeal of level set descriptions coupled to enriched finite element approaches such as CutFEM lies in the fact that the motion of geometries over time only requires updating the level set function, without modifying the mesh. This will be instrumental to consider complex and time dependent shapes of cells such as astrocytes.

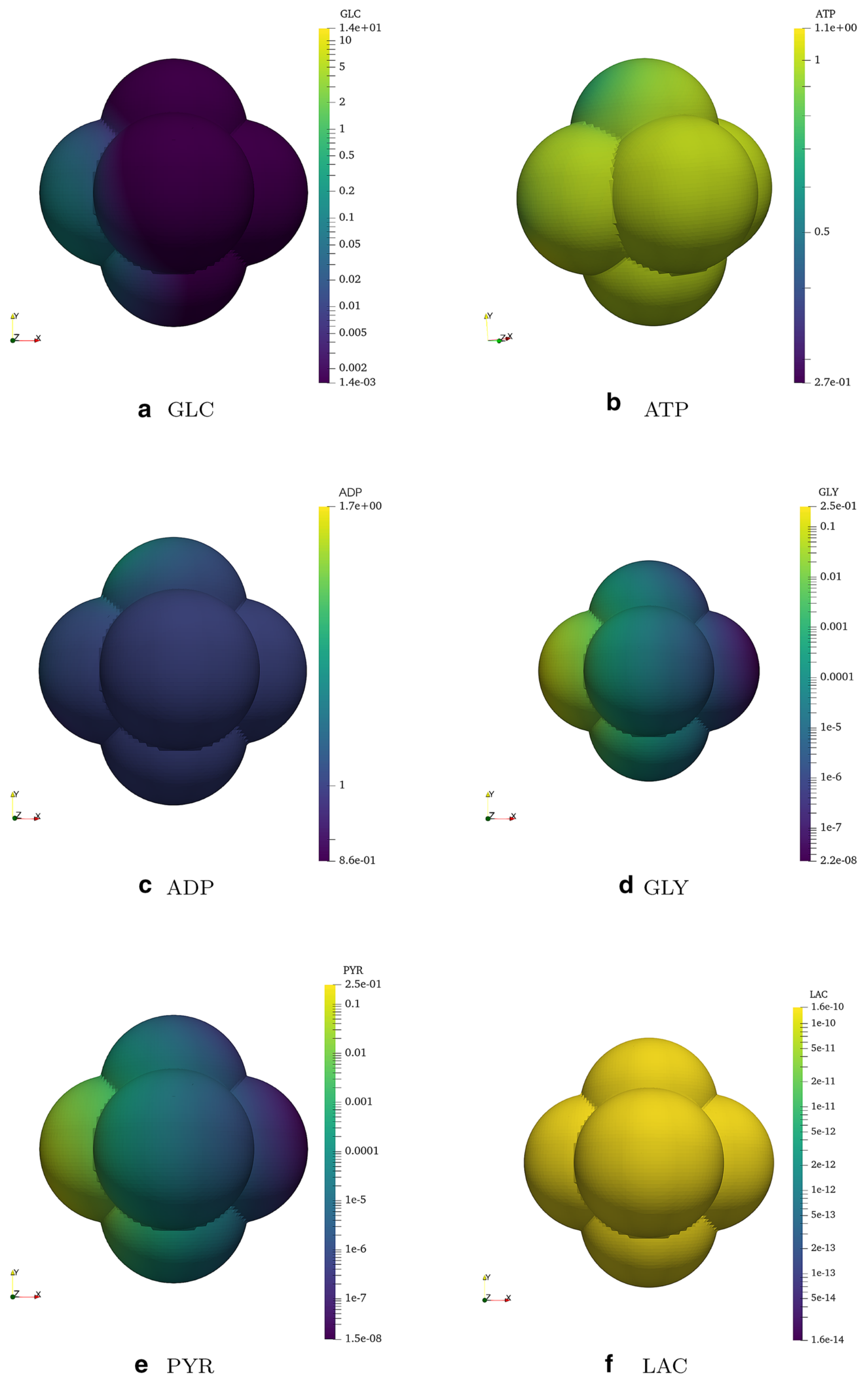


**Fig. 13** Solution of the reaction–diffusion system at time $t = 5$ solved using **CutFEM** inside a 3 dimensional pop corn shape. From top to bottom from left to right [GLC], [ATP], [ADP], [GLY], [PYR], [LAC]. The GLC is diffusing into the domain starting from the influx location. ATP and ADP are consumed and produced by the reaction act. GLY is produced by HXK and PYR produced by PYRK

Our next steps will be to initiate the geometry of the cells through microscopy images 1 and to investigate the effect of perturbed astrocytic metabolism on neuronal support, which could lead to a better understanding of mechanisms in neurodegeneration.

**Abbreviations**
GLC: Glucose; ATP: Adenosine triphosphate; ADP: Adenosine diphosphate; GLY: Glyceraldehyde 3-phosphate; PYR: Pyruvate; LAC: Lactate; $[\,\cdot\,]$: Concentration; HXK: Hexokinase; PYRK: Pyruvate kinase; Mito: TCA cycle activity; LDH: Lactate dehydrogenase; act: ATP consuming cellular activity; $\Omega$: Domain; $D_{[\cdot]}$: Diffusion term; $K_{\cdot}$: Reaction rate chemical reaction; $f$: Source term; $\mathcal{G}_{\cdot}$: Gaussian function; **CutFEM**: Cut Finite Element Method; **PDEs**: Partial Differential Equations; ODEs: Ordinary Differential Equations.

**Acknowledgements**
We would like to thank Niko Komin for his contribution in the simplified model of energy metabolism and David Bouvier and Corrado Ameli to provide the reconstruction of the astrocyte shown in Fig. 1.

**Authors' contributions**
SF developed the content and the code based on the model of AS. JH and SC contributed to the **FEniCS** Project and **CutFEM** implementation. SB supervises the project. All authors have contributed to the preparation of the final manuscript. All authors read and approved the final manuscript.

**Funding**
This work is part of the DRIVEN project funded by Luxembourg National Research Fund under the PRIDE programme (PRIDE17/12252781).

**Availability of data and materials**
Two fully commented examples can be found at https://bitbucket.org/sofiafarina/cutfem-energy-metabolism/ and are licensed under the LGPLv3. The standard FEM example was written against **FEniCS** Project version 2019.2.0. To run the CutFEM example access to the CutFEM library is required. This access can be granted on written request to the second author.

**Declarations**

**Competing interests**
The authors declare that they have no competing interests.

**Author details**
[1]Institute of Computational Engineering, University of Luxembourg,, Maison du Nombre, 6 Avenue de la Fonte, 4364 Esch-sur-Alzette, Luxembourg, [2]LCSB-Luxembourg Centre for Systems Biomedicine, 7 Avenue des Hauts-Fourneaux, 4362 Esch-sur-Alzette, Luxembourg, [3]Onera, 6 Chemin de la Vauve aux Granges, 91120 Palaiseau, France.

## Appendix A: Asymptotic solution

The system of ordinary differential equations is obtained applying the law of mass action to Eqs. (1)–(5). To avoid misinterpretation, we denote the concentration $A = [\text{GLC}]$, $B = [\text{ATP}]$, $C = [\text{ADP}]$, $D = [\text{GLY}]$, $E = [\text{PYR}]$, $F = [\text{LAC}]$ and we obtain the system

$$\begin{cases} \frac{dA}{dt} = -K_{HXK}AB^2 \\ \frac{dB}{dt} = -2K_{HXK}AB^2 + 2K_{PYRK}C^2D + 28K_{Mito}E - K_{act}B \\ \frac{dC}{dt} = 2K_{HXK}AB^2 - 2K_{PYRK}C^2D + K_{act}B \\ \frac{dD}{dt} = 2K_{HXK}AB^2 - K_{PYRK}C^2D \\ \frac{dE}{dt} = K_{PYRK}C^2D - K_{LDH}E - K_{Mito}E \\ \frac{dF}{dt} = K_{LDH}E \end{cases} \tag{24}$$

The initial condition for Eq. (24) are chosen using the solution of the PDEs system Eq. (6). We compute the integral over the domain $\Omega$ using Eq. (23) of the solutions of the PDEs when the source term ends, making stable the total amount of concentrations inside the domain

$$\begin{cases} A(t=0) = \int_\Omega [GLC](t=1)\mathrm{d}x := A_0 \\ B(t=0) = \int_\Omega [ATP](t=1)\mathrm{d}x := B_0 \\ C(t=0) = \int_\Omega [ADP](t=1)\mathrm{d}x := C_0 \\ D(t=0) = \int_\Omega [GLY](t=1)\mathrm{d}x := D_0 \\ E(t=0) = \int_\Omega [PYR](t=1)\mathrm{d}x := E_0 \\ F(t=0) = \int_\Omega [LAC](t=1)\mathrm{d}x := F_0 \end{cases} \tag{25}$$

Considering abundance of ATP and ADP inside the domain, this lead to only a possible system of steady state solution, that is when all the GLC is consumed by reaction HXK Eq. (1), GLY is consumed by reaction PYRK Eq. (2) and ATP is fully transformed into ADP from Eq. (5):

$$\begin{cases} A_\infty = 0 \\ B_\infty = 0 \\ C_\infty = C_0 + B_0 + 28(E_0 + D_0 + 2A_0)\alpha \\ D_\infty = 0 \\ E_\infty = 0 \\ F_\infty = F_0 + (E_0 + 2A_0 + D_0)(1-\alpha) \end{cases} \tag{26}$$

In the solutions we have introduced a parameter $\alpha \in [0, 1]$ which take into account the fact that the concentration of PYR is transformed not in equal part to LAC and ATP from the reaction LDH Eq. (3) and reaction Mito Eq. (4).

Received: 3 April 2020 Accepted: 16 February 2021
Published online: 22 March 2021

## References

1. Cloutier M, Bolger FB, Lowry JP, Wellstead P. An integrative dynamic model of brain energy metabolism using in vivo neurochemical measurements. J Comput Neurosci. 2009;27(3):391. https://doi.org/10.1007/s10827-009-0152-8.
2. Magistretti P, Pellerin L. Cellular mechanisms of brain energy metabolism. Relevance to functional brain imaging and to neurodegenerative disorders a. Ann N Y Acad Sci. 1996;777(1):380–7. https://doi.org/10.1111/j.1749-6632.1996.tb34449.x.
3. Camandola S, Mattson M.P. Brain metabolism in health, aging, and neurodegeneration. EMBO J. 2017;36(11):1474–92. https://doi.org/10.15252/embj.201695810.
4. Bélanger M, Allaman I, Magistretti PJ. Brain energy metabolism: focus on astrocyte-neuron metabolic cooperation. Cell Metab. 2011;14(6):724–38. https://doi.org/10.1016/j.cmet.2011.08.016.
5. Salamanca L, Mechawar N, Murai KK, Balling R, Bouvier DS, Skupin A. Mic-mac: an automated pipeline for high-throughput characterization and classification of three-dimensional microglia morphologies in mouse and human postmortem brain samples. Glia. 2019;67(8):1496–509. https://doi.org/10.1002/glia.23623.
6. Rappel H, Beex LA, Hale JS, Bordas S. Bayesian inference for the stochastic identification of elastoplastic material parameters: introduction, misconceptions and insights. arXiv preprint arXiv:1606.02422. 2016.
7. Hauseux P, Hale JS, Cotin S, Bordas SP. Quantifying the uncertainty in a hyperelastic soft tissue model with stochastic parameters. Appl Math Model. 2018;62:86–102. https://doi.org/10.1016/j.apm.2018.04.021.
8. Rappel H, Beex LA, Bordas SP. Bayesian inference to identify parameters in viscoelasticity. Mech Time Depend Mater. 2018;22(2):221–58. https://doi.org/10.1007/s11043-017-9361-0.
9. Rappel H, Beex LA, Hale JS, Noels L, Bordas S. A tutorial on Bayesian inference to identify material parameters in solid mechanics. Arch Comput Methods Eng. 2019. https://doi.org/10.1007/s11831-018-09311-x.
10. Rappel H, Beex LA, Noels L, Bordas S. Identifying elastoplastic parameters with Bayes' theorem considering output error, input error and model uncertainty. Probabilistic Eng Mech. 2019;55:28–41. https://doi.org/10.1016/j.probengmech.2018.08.004.
11. Rappel H, Beex L. Estimating fibres' material parameter distributions from limited data with the help of Bayesian inference. Eur J Mech A Solids. 2019;75:169–96. https://doi.org/10.1016/j.euromechsol.2019.01.001.
12. Bui HP, Tomar S, Courtecuisse H, Cotin S, Bordas SP. Real-time error control for surgical simulation. IEEE Trans Biomed Eng. 2017;65(3):596–607. https://doi.org/10.1109/TBME.2017.2695587.

13. Duprez M, Bordas S, Bucki M, Bui HP, Chouly F, Lleras V, Lobos C, Lozinski A, Rohan P-Y, Tomar S. Quantifying discretization errors for soft-tissue simulation in computer assisted surgery: a preliminary study. arXiv preprint arXiv:1806.06944 (2018). https://doi.org/10.1016/j.apm.2019.07.055.
14. Bui HP, Tomar S, Courtecuisse H, Audette M, Cotin S, Bordas SP. Controlling the error on target motion through real-time mesh adaptation: applications to deep brain stimulation. Int J Numer Methods Biomed Eng. 2018;34(5):2958. https://doi.org/10.1002/cnm.2958.
15. Logg A, Mardal K-A, Wells G. Automated solution of differential equations by the finite element method: the FEniCS book. Heidelberg: Springer; 2012.
16. Alnæs M.S, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, Richardson C, Ring J, Rognes M.E, Wells G.N. The FEniCS project version 1.5. Arch Numer Softw. 2015;3(100):9–23. https://doi.org/10.11588/ans.2015.100.20553.
17. Hale JS, Brunetti M, Bordas SP, Maurini C. Simple and extensible plate and shell finite element models through automatic code generation tools. Comput Struct. 2018;209:163–81. https://doi.org/10.1016/j.compstruc.2018.08.001.
18. Renard Y, Pommier J. Getfem finite element library. 2007. http://home.gna.org/getfem.
19. Hecht F. New development in FreeFem++. J Numer Math. 2012;20(3–4):251–65.
20. Arndt D, Bangerth W, Clevenger T.C, Davydov D, Fehling M, Garcia-Sanchez D, Harper G, Heister T, Heltai L, Kronbichler M, Kynch R.M, Maier M, Pelteret J.-P, Turcksin B, Wells D. The `deal.II` library, version 9.1. J Numer Math. 2019;27(4):203–13. https://doi.org/10.1515/jnma-2019-0064.
21. Pellerin L, Magistretti PJ. Glutamate uptake into astrocytes stimulates aerobic glycolysis: a mechanism coupling neuronal activity to glucose utilization. Proc Natl Acad Sci. 1994;91(22):10625–9.
22. Michaelis L, Menten ML. Die Kinetik der Invertinwirkung. Universitätsbibliothek Johann Christian Senckenberg. 2007.
23. Çakır T, Alsan S, Saybaşılı H, Akın A, Ülgen KÖ. Reconstruction and flux analysis of coupling between metabolic pathways of astrocytes and neurons: application to cerebral hypoxia. Theor Biol Med Model. 2007;4(1):48. https://doi.org/10.1186/1742-4682-4-48.
24. Sertbaş M, Ülgen K, Çakır T. Systematic analysis of transcription-level effects of neurodegenerative diseases on human brain metabolism by a newly reconstructed brain-specific metabolic network. FEBS Open Bio. 2014;4(1):542–53. https://doi.org/10.1016/j.fob.2014.05.006.
25. Khalid MU, Tervonen A, Korkka I, Hyttinen J, Lenk K. Geometry-based computational modeling of calcium signaling in an astrocyte. In: Eskola H, Väisänen O, Viik J, Hyttinen J, editors. EMBEC NBC 2017. Singapore: Springer; 2017. p. 157–60.
26. Ellingsrud A, Solbrå A, Einevoll G, Halnes G, Rognes M. Finite element simulation of ionicelectrodiffusion in cellular geometries. arXiv preprint arXiv:1911.03211. 2019.
27. Arranz AM, De Strooper B. The role of astroglia in Alzheimer's disease: pathophysiology and clinical implications. Lancet Neurol. 2019. https://doi.org/10.1016/S1474-4422(18)30490-3.
28. Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. J Comput Phys. 1988;79(1):12–49. https://doi.org/10.1016/0021-9991(88)90002-2.
29. Sethian JA. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge: Cambridge University Press; 1999.
30. Duddu R, Bordas S, Chopp D, Moran B. A combined extended finite element and level set method for biofilm growth. Int J Numer Methods Eng. 2008;74(5):848–70. https://doi.org/10.1002/nme.2200.
31. Duddu R, Chopp DL, Moran B. A two-dimensional continuum model of biofilm growth incorporating fluid flow and shear stress based detachment. Biotechnol Bioeng. 2009;103(1):92–104. https://doi.org/10.1002/bit.22233.
32. Marot C, Pellerin J, Remacle J-F. One machine, one minute, three billion tetrahedra. Int J Numer Methods Eng. 2019;117(9):967–90. https://doi.org/10.1002/nme.5987.
33. Melenk JM, Babuška I. The partition of unity finite element method: basic theory and applications. Comput Methods Appl Mech Eng. 1996;139(1–4):289–314. https://doi.org/10.1016/S0045-7825(96)01087-0.
34. Babuška I, Melenk JM. The partition of unity method. Int J Numer Methods Eng. 1997;40(4):727–58.
35. Strouboulis T, Copps K, Babuška I. The generalized finite element method. Comput Methods Appl Mech Eng. 2001;190(32–33):4081–193. https://doi.org/10.1016/S0045-7825(01)00188-8.
36. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. Int J Numer Methods Eng. 1999;45(5):601–20. https://doi.org/10.1002/(SICI)1097-0207(19990620)45:53.0.CO;2-S.
37. Bordas S, Moran B. Enriched finite elements and level sets for damage tolerance assessment of complex structures. Eng Fract Mech. 2006;73(9):1176–201. https://doi.org/10.1016/j.engfracmech.2006.01.006.
38. Béchet É, Minnebo H, Moës N, Burgardt B. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. Int J Numer Methods Eng. 2005;64(8):1033–56. https://doi.org/10.1002/nme.1386.
39. Laborde P, Pommier J, Renard Y, Salaün M. High-order extended finite element method for cracked domains. Int J Numer Methods Eng. 2005;64(3):354–81. https://doi.org/10.1002/nme.1370.
40. Menk A, Bordas SP. A robust preconditioning technique for the extended finite element method. Int J Numer Methods Eng. 2011;85(13):1609–32. https://doi.org/10.1002/nme.3032.
41. Gupta V, Duarte CA, Babuška I, Banerjee U. Stable GFEM (SGFEM): improved conditioning and accuracy of GFEM/XFEM for three-dimensional fracture mechanics. Comput Methods Appl Mech Eng. 2015;289:355–86. https://doi.org/10.1016/j.cma.2015.01.014.
42. Agathos K, Chatzi E, Bordas SP, Talaslidis D. A well-conditioned and optimally convergent XFEM for 3D linear elastic fracture. Int J Numer Methods Eng. 2016;105(9):643–77. https://doi.org/10.1002/nme.4982.
43. Agathos K, Chatzi E, Bordas SP. Multiple crack detection in 3D using a stable XFEM and global optimization. Comput Mech. 2018;62(4):835–52. https://doi.org/10.1007/s00466-017-1532-y.
44. Agathos K, Bordas SP, Chatzi E. Improving the conditioning of XFEM/GFEM for fracture mechanics problems through enrichment quasi-orthogonalization. Comput Methods Appl Mech Eng. 2019;346:1051–73. https://doi.org/10.1016/j.cma.2018.08.007.
45. Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. Comput Methods Appl Mech Eng. 2002;191(47–48):5537–52. https://doi.org/10.1016/S0045-7825(02)00524-8.
46. Burman E. Ghost penalty. Comptes Rendus Mathematique. 2010;348(21–22):1217–20. https://doi.org/10.1016/j.crma.2010.10.006.

47. Burman E, Claus S, Hansbo P, Larson MG, Massing A. CutFEM: discretizing geometry and partial differential equations. Int J Numer Methods Eng. 2015;104(7):472–501. https://doi.org/10.1002/nme.4823.
48. Boiveau T, Burman E, Claus S, Larson M. Fictitious domain method with boundary value correction using penalty-free Nitsche method. J Numer Math. 2018;26(2):77–95. https://doi.org/10.1515/jnma-2016-1103.
49. Claus S, Kerfriden P. A stable and optimally convergent Latin-cutFEM algorithm for multiple unilateral contact problems. Int J Numer Methods Eng. 2018;113(6):938–66. https://doi.org/10.1002/nme.5694.
50. Claus S, Bigot S, Kerfriden P. CutFEM method for Stefan–Signorini problems with application in pulsed laser ablation. SIAM J Sci Comput. 2018;40(5):1444–69. https://doi.org/10.1137/18M1185697.
51. Claus S, Kerfriden P. A cutFEM method for two-phase flow problems. Comput Methods Appl Mech Eng. 2019;348:185–206. https://doi.org/10.1016/j.cma.2019.01.009.
52. Alnæs MS, Logg A, Ølgaard KB, Rognes ME, Wells GN. Unified form language: a domain-specific language for weak formulations of partial differential equations. ACM Trans Math Softw (TOMS). 2014;40(2):1–37. https://doi.org/10.1145/2566630.
53. Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp WD, Karpeyev D, Kaushik D, Knepley MG, May DA, McInnes LC, Mills RT, Munson T, Rupp K, Sanan P, Smith BF, Zampini S, Zhang H, Zhang H. PETSc users manual. Technical Report ANL-95/11—Revision 3.12, Argonne National Laboratory; 2019. https://www.mcs.anl.gov/petsc.
54. Murray JD. Mathematical biology: I. An introduction, vol. 17. New York: Springer; 2007. p. 175–500.
55. Waage P, Gulberg CM. Studies concerning affinity. J Chem Educ. 1986;63(12):1044. https://doi.org/10.1021/ed063p1044.
56. Pierre M. Global existence in reaction-diffusion systems with control of mass: a survey. Milan J Math. 2010;78(2):417–55. https://doi.org/10.1007/s00032-010-0133-4.
57. Brenner S, Scott R. The mathematical theory of finite element methods. In: Texts in applied mathematics, 3rd edition. New York: Springer; 2008. https://doi.org/10.1007/978-0-387-75934-0. https://www.springer.com/gp/book/9780387759333. Accessed 08 Jan 2020.
58. Allik H, Hughes TJ. Finite element method for piezoelectric vibration. Int J Numer Methods Eng. 1970;2(2):151–7. https://doi.org/10.1002/nme.1620020202.
59. Requicha AA, Voelcker HB. Constructive solid geometry. 1977.
60. Quarteroni A, Valli A. Numerical approximation of partial differential equations. Heidelberg: Springer; 2008.
61. Moumnassi M, Belouettar S, Béchet É, Bordas SP, Quoirin D, Potier-Ferry M. Finite element analysis on implicitly defined domains: an accurate representation based on arbitrary parametric surfaces. Comput Methods Appl Mech Eng. 2011;200(5–8):774–96.
62. Moumnassi M, Bordas SPA, Figueredo R, Sansen P. Analysis using higher-order XFEM: implicit representation of geometrical features from a given parametric representation. Mech Ind. 2014;15(5):443–8.
63. Quesseveur G, d'Hérouël AF, Murai KK, Bouvier DS. A specialized method to resolve fine 3D features of astrocytes in nonhuman primate (marmoset, callithrix jacchus) and human fixed brain samples. In: Benedetto BD, editor. Astrocytes. Regensburg: Springer; 2019. p. 85–95. https://doi.org/10.1007/978-1-4939-9068-9_6.

## Publisher's Note