Engineering with Computers
https://doi.org/10.1007/s00366-024-02094-9

ORIGINAL ARTICLE

A cyclical fast iterative method for simulating reentries in cardiac electrophysiology using an eikonal-based model

Cristian Barrios Espinosa[1] · Jorge Sánchez[2] · Stephanie Appel[1] · Silvia Becker[1,3] · Jonathan Krauß[1] · Patricia Martínez Díaz[1] · Laura Unger[1,4] · Marie Houillon[1] · Axel Loewe[1]

Received: 25 June 2024 / Accepted: 27 November 2024

Abstract
Computer models for simulating cardiac electrophysiology are valuable tools for research and clinical applications. Traditional reaction–diffusion (RD) models used for these purposes are computationally expensive. While eikonal models offer a faster alternative, they are not well-suited to study cardiac arrhythmias driven by reentrant activity. The present work extends the diffusion–reaction eikonal alternant model (DREAM), incorporating conduction velocity (CV) restitution for simulating complex cardiac arrhythmias. The DREAM modifies the fast iterative method to model cyclical behavior, dynamic boundary conditions, and frequency-dependent anisotropic CV. Additionally, the model alternates with an approximated RD model, using a detailed ionic model for the reaction term and a triple-Gaussian to approximate the diffusion term. The DREAM and monodomain models were compared, simulating reentries in 2D manifolds with different resolutions. The DREAM produced similar results across all resolutions, while experiments with the monodomain model failed at lower resolutions. CV restitution curves obtained using the DREAM closely approximated those produced by the monodomain simulations. Reentry in 2D sheets yielded similar results in vulnerable window and mean reentry duration for low CV in both models. In the left atrium, most inducing points identified by the DREAM were also present in the high-resolution monodomain model. DREAM's reentry simulations on meshes with an average edge length of 1600 µm were 87x faster than monodomain simulations at 200 µm. This work establishes the mathematical foundation for using the accelerated DREAM simulation method for cardiac electrophysiology. Cardiac research applications are enabled by a publicly available implementation in the openCARP simulator.

Keywords Fast iterative method · Reaction diffusion · Cardiac electrophysiology · Reentries

1 Introduction

Computer models have provided meaningful contributions to better understand the mechanisms of cardiac arrhythmia [1, 2]. An emerging application of cardiac modeling are tissue-level simulations to guide treatments such as ablation procedures [3]. Tissue-level simulations can be performed using reaction diffusion (RD) or eikonal models [4, 5]. RD models accurately capture the complex interplay between ion channels, cellular, and tissue-level behavior [4, 6]. However, they often require significant computing time, even on high performance computing systems [2]. Therefore, these models are hardly compatible with clinical time frames for intraprocedural decision support. Eikonal models are an alternative to investigate cardiac arrhythmias [5, 7–9]. They can be 3 orders of magnitude faster than RD models and thus better suited for use in clinical settings (e.g. planning ablation procedures) or uncertainty quantification [10, 11]. However, various eikonal-based models encounter distinct challenges that impede their ability to accurately simulate cardiac arrhythmias. These challenges include the absence of repolarization and reactivation phenomena, inadequate

✉ Axel Loewe
publications@ibt.kit.edu

[1] Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

[2] ITACA Institute, Universitat Politècnica de València, Valencia, Spain

[3] Klinik für Kardiologie und Angiologie, Universitäts-Herzzentrum Freiburg-Bad Krozingen, Bad Krozingen, Germany

[4] Medizinische Klinik IV, Städtisches Klinikum Karlsruhe, Karlsruhe, Germany

Published online: 07 January 2025

representation of conduction velocity (CV) and action potential duration (APD) restitution (i.e., their dependence on activation frequency), and the utilization of numerical methods unsuitable for anisotropic propagation [5, 10, 12]. While not all eikonal-based models face all these limitations, a model capable of addressing these challenges simultaneously could provide a more suitable tool for studying arrhythmia compatible with clinical time frames.

Numerical solutions for the eikonal equation are computationally inexpensive due to its simple formulation and the low mesh resolution requirements. The simplest formulations of the eikonal model are only capable of simulating one activation per node. Consequently, these versions do not account for recovery or reactivation of the cardiomyocytes [10, 13]. This shortcoming hinders the simulation of reentrant activity, which is a major limitation in the context of simulating arrhythmia. To overcome this problem, Pernod et al. modified the fast marching method (FMM) to allow reactivation of the nodes while solving the eikonal equation [5]. Later, Gassa et al. extended the method to enable the simulation of rotors [14]. Nonetheless, when simulating anisotropic propagation in cardiac tissue using single pass methods like the regular FMM, numercial errors can arise because the gradient directions of the eikonal equation solution do not align with the characteristic directions (i.e., the optimal trajectories). This discrepancy affects the accurate depiction of anisotropic wave propagation dynamics, particularly in regions where these directions do not lie within the same simplex in the mesh discretization. Further limitations of single pass methods like FMM in anisotropic media are addressed in more detail by Sethian and Vladimirsky[15]. Alternative methods have been proposed to solve the anisotropic eikonal equation such as the fast iterative method (FIM), the buffered FMM, and the anisotropic FMM [5, 16, 17].

Neic et al. used the FIM to develop the reaction eikonal (RE) model, which incorporates repolarization by linking the eikonal equation with a detailed ionic model [10]. However, it lacks the capability to simulate reactivation and reentry. Later, Campos et al. employed the RE model in virtual induction and treatment of arrhythmias (VITA), a method to investigate ventricular tachycardia [18]. While VITA can identify areas in the heart susceptible to isthmus-dependent reentry, it can only simulate the first reentry cycle and disregards CV restitution and functional reentry.

Iterative methods to solve the anisotropic eikonal equation are challenging when simulating reactivation and reentry phenomena. There are 2 main factors hindering the accurate simulation of reentries using iterative solution methods. First, activation times (ATs) can undergo multiple changes while solving the eikonal equation iteratively unlike in single pass methods. Second, ATs and repolarization times (RTs), i.e. the time after which a node can be reactivated, have a mutual dependency. The RT of every node in a given activation cycle depends on the effective refractory period (ERP) and the AT in the same activation cycle. Similarly, the AT depends on whether a node has fully recovered from the previous cycle's activation before the next activation attempt. Managing these conditions becomes intricate when computing the AT for the next activation cycle while constantly updating the AT from the previous activation cycle. Another important factor to reproduce physiological reentries is to incorporate CV restitution. This phenomenon adds additional complexity as CV becomes dependent on the previous AT, rather than remaining constant. Therefore, a way to calculate the frequency-dependent CV in the eikonal model must be included in the method. This can be achieved by taking the diastolic interval (DI) of previous activation cycles in the node that is being activated, or by considering the DI of the neighboring nodes [12].

This study builds on an initial version of the diffusion reaction eikonal alternant model (DREAM) enabling reactivation in anisotropic media through the solution of the eikonal equation using the FIM [19]. The DREAM introduced a new strategy that alternates between the eikonal and RD models. In the proposed update of the DREAM, a novel approach to CV restitution ensures a coherent set of CV, DI, and AT values for each revised node during the present activation cycle. This work extends the DREAM by CV restitution properties while preserving the model's other advantages.

2 Propagation models

2.1 Monodomain model

RD models faithfully represent the propagation of the electrical wavefront through the cardiac tissue [20–22]. These models are the most detailed because they incorporate more physiological mechanisms than other available models. However, numerical methods used to solve the RD equations rely on high resolution meshes, which is the main cause for their high computational cost [4, 23]. The most common examples of RD models are the bidomain and the monodomain models [24]. In this work, the latter is used as control to benchmark the DREAM.

Derived from the bidomain equations, the monodomain model assumes equal intracellular and extracellular anisotropy ratios and is, therefore, computationally more efficient. This assumption does not hold true particularly in scenarios such as the simulation of defibrillation, where the dynamics in the extracellular space play a significant role [25]. However, for the majority of cardiac electrophysiology simulations, the monodomain model proved to be adequate due to its ability to capture various electrophysiological mechanisms accurately, such as source-sink mismatch effects [4, 24]. The equations of the bidomain model are condensed to the monodomain equation:

$$\beta C_{\mathrm{m}} \frac{\partial V_{\mathrm{m}}}{\partial t} = \nabla \cdot (\boldsymbol{\sigma}_{\mathrm{m}} \nabla V_{\mathrm{m}}) - \beta (I_{\mathrm{ion}}(V_{\mathrm{m}}, \vec{\eta}) - I_{\mathrm{s}}) \quad \text{on } \Omega \subset \mathbb{R}^3, \tag{1}$$

$$(\boldsymbol{\sigma}_{\mathrm{m}} \nabla V_{\mathrm{m}}) \cdot \vec{n}_{\mathrm{surf}} = 0 \quad \text{on } \delta\Omega, \tag{2}$$

where β is the surface-to-volume ratio, while C_{m} denotes the membrane capacitance. The transmembrane voltage is indicated by V_{m} and $\boldsymbol{\sigma}_{\mathrm{m}}$ represents the tissue conductivity tensor. The ionic transmembrane current density, denoted as I_{ion}, depends on V_{m} and the state variables $\vec{\eta}$, which determine the behavior of the ion channels in the cell membrane and the sarcoplasmic reticulum. Additionally, I_{s} is the transmembrane stimulus current density. The myocardium is represented by Ω. There is a non-flux boundary condition at $\delta\Omega$, the boundary of the domain Ω. The outward surface normal vector is represented by $\vec{n}_{\mathrm{surf}}$.

2.2 Eikonal model

The anisotropic eikonal model, based on the macroscopic kinetics of wavefront propagation, seeks to determine the activation time (AT) of points within the myocardium through the following equation: [21, 22]:

$$\sqrt{\nabla T(X)^{\top} \mathbf{M}(X) \nabla T(X)} = 1 \quad \forall X \in \Omega, \tag{3}$$

$$T(X) = T_i \quad \forall X \in \Gamma_i \subset \Omega, \tag{4}$$

where $T : \Omega \to \mathbb{R}_{\geq 0} \cup \{\infty\}$, maps every point in the myocardium to its corresponding AT. Moreover, Γ_i denotes the subset of points in the myocardium where the i-th stimulus is applied at time $T_i \in \mathbb{R}_{\geq 0}$, for $i = 1, \ldots, n_s$, with $n_s \in \mathbb{N}$ representing the total number of stimuli. Additionally, if $T_i = T_j$ then $i = j$. $\mathbf{M} : \Omega \to \mathbb{R}^{3\times 3}$ maps points in the myocardium to their tensor of squared CV defined as:

$$\mathbf{M}(X) = (v_{\mathrm{l}}^2(X))\vec{l} \otimes \vec{l} + (v_{\mathrm{t}}^2(X))\vec{t} \otimes \vec{t} + (v_{\mathrm{n}}^2(X))\vec{n} \otimes \vec{n}. \tag{5}$$

Here, $\vec{l}$, $\vec{t}$, and $\vec{n}$ form an orthonormal system of vectors in the longitudinal, transversal, and sheet-normal directions, respectively. Importantly, $\vec{l}$ is aligned with the local preferential myocyte orientation. Furthermore, $v_{\mathrm{l}}, v_{\mathrm{t}}, v_{\mathrm{n}} : \Omega \to \mathbb{R}_{\geq 0}$ assign the CV values in their respective directions at X.

The FIM is an effective approach for solving the anisotropic eikonal equation as this algorithm is particularly suited for unstructured meshes and anisotropic local CV functions [16]. The single-thread version of the FIM is presented in Algorithm 1. Multi-thread versions are also available [16, 26, 27]. For this work, Ω is considered to be a 2D manifold embedded in $\mathbb{R}^3$. For this reason, it is assumed that $v_{\mathrm{n}} = v_{\mathrm{t}}$, however the effect of the normal component is small since most of the characteristic directions (i.e., optimal trajectories) are almost perpendicular to $\vec{n}$. To numerically approximate the viscosity solution of the eikonal equation, a triangulation $\mathcal{T} \subset \mathcal{P}(\Omega_{\mathcal{T}})$ is defined over a finite set of points $\Omega_{\mathcal{T}} \subset \Omega$, such that the convex hull of $\Omega_{\mathcal{T}}$ (i.e., the union of all triangles in $\mathcal{T}$) approximates Ω. The FIM approximates the viscosity solution of the eikonal equation only at the vertices of the triangles in $\mathcal{T}$ (i.e., points in $\Omega_{\mathcal{T}}$). When mentioning a node $X \in \Omega_{\mathcal{T}}$, we refer to both a vertex in the triangulation and its position in $\mathbb{R}^3$. This notation should not cause confusion, as each simulation in this work uses only a single mesh and coordinate system.

In the first step of the FIM, the boundary conditions of the system are defined by the AT of the source nodes (Eq. 4). Then, the neighbors of the nodes that belong to any Γ_i (for $i = 1, \ldots, n_s$) are included in a set of active nodes $L \subset \Omega_{\mathcal{T}}$, that is initialize as $\emptyset$, which contains the list of nodes that are being updated by the local solver. This local solver is referred to as the UPDATE() function. As soon as L is not empty, the list iteration begins. For each iteration, every node X in L is updated and the previous solution for its AT is replaced. If the difference between the old and the new solution is smaller than a certain threshold ε, the node is then removed from L and each of its neighbors, that are not presently in L, is analyzed. For each neighbor, a tentative new solution is calculated and only replaces the old solution if the new solution is smaller (i.e., earlier) than the old solution. If this condition is fulfilled, this neighbor is added to L. This process is repeated until L is empty.

Algorithm 1 Fast Iterative Method

```
$L = \emptyset$
for $X \in \Omega_{\mathcal{T}}$
    $T(X) \leftarrow \infty$
    for $i \in 1, \ldots, n_s$
        if $X \in \Gamma_i$
            $T(X) \leftarrow T_i$
            for adjacent neighbor $X_{\mathrm{NB}}$ of $X$
                add $X_{\mathrm{NB}}$ to $L$
            end for
        end if
    end for
end for
while $L \neq \emptyset$
    for $X \in L$
        $p \leftarrow T(X)$
        $q \leftarrow \mathrm{UPDATE}(X)$
        $T(X) \leftarrow q$
        if $|p - q| < \varepsilon$
            for adjacent neighbor $X_{\mathrm{NB}}$ of $X$
                if $X_{\mathrm{NB}}$ is not in $L$
                    $p \leftarrow T(X_{\mathrm{NB}})$
                    $q \leftarrow \mathrm{UPDATE}(X_{\mathrm{NB}})$
                    if $p > q$
                        $T(X_{\mathrm{NB}}) \leftarrow q$
                        add $X_{\mathrm{NB}}$ to $L$
                    end if
                end if
            end for
            remove $X$ from $L$
        end if
    end for
end while
```

When solving the eikonal equation on a triangular mesh, the local solver aims to determine the smallest AT that fits the eikonal equation at a specific node X. For this purpose, a tentative AT is calculated for every triangle containing X. Let (X, Y, Z) be a triangle in $\mathcal{T}$, with vertices $X, Y, Z \in \Omega_{\mathcal{T}}$:

$$\begin{aligned} T_{Y,Z}(X) &= \min_{\lambda \in [0,1]} \left(\lambda T(Y) + (1-\lambda) T(Z) + \frac{\sqrt{\overrightarrow{X_\lambda X}^{\top} \mathbf{D}(X)^{-1} \overrightarrow{X_\lambda X}}}{v_1(X)} \right), \\ \overrightarrow{X_\lambda X} &= \lambda \overrightarrow{YX} + (1-\lambda) \overrightarrow{ZX}, \\ \mathbf{D}(X) &= \mathbf{M}(X) \cdot v_1(X)^{-2}, \end{aligned} \tag{6}$$

where $T_{Y,Z}(X)$ is the tentative AT that is obtained if the characteristic direction lies within the triangle (X, Y, Z). Additionally, $\mathbf{D}(X)$ is a tensor that holds information about the anisotropy of conduction, and $v_1(X)$ is the CV along the longitudinal direction at node X. Finally, UPDATE(X) is set as the minimum AT among all the tentative ATs calculated from each triangle containing X:

$$\mathrm{UPDATE}(X) = \min_{(X,Y,Z) \in \mathcal{T}} T_{Y,Z}(X). \tag{7}$$

2.3 Diffusion reaction eikonal alternant model (DREAM)

The DREAM is a mixed model combining an approximation of the RD model and the eikonal model. The goal of this model is to simulate reactivation patterns on meshes with lower resolutions than required for comparable simulations with RD models, thereby increasing computational efficiency. The DREAM is inspired by the RE model [10] and the multi-frontal FMM [5]. A modified version of the FIM, named cyclical FIM (cycFIM), was implemented to solve the anisotropic eikonal equation allowing reactivations by alternating with the approximated RD model (Sect. 2.3.1).

Figure 1 shows a schematic representation of the steps involved in the DREAM. A single DREAM cycle encompasses the execution of steps A, B, C, and D where subscript indices represent the DREAM cycle's number. It is crucial to differentiate between the DREAM cycle, comprising steps A, B, C, and D as shown in Fig. 1b), and the concept of the activation cycle in cardiac tissue. Within a DREAM cycle, each call of the cycFIM is regarded as a cycFIM cycle, which computes one AT value per node for a subset of nodes. However, it is important to note that not every review of a node will result in an activation cycle for that node, as propagation failures can occur. Additionally, not more than one AT is calculated per DREAM cycle per node.

At first, the cycFIM applies the first stimulus (i.e., computes the first boundary condition) and iteratively reviews every node in L and updates the AT of each node by solving the eikonal equation. A list iteration refers to the process where each node in L is visited once. Therefore, every cycle encompasses one or more list iterations. At the end of each list iteration, t_{min} is the smallest absolute AT among all nodes in L. A parameter τ_{inc} is defined to limit the increment of t_{min}

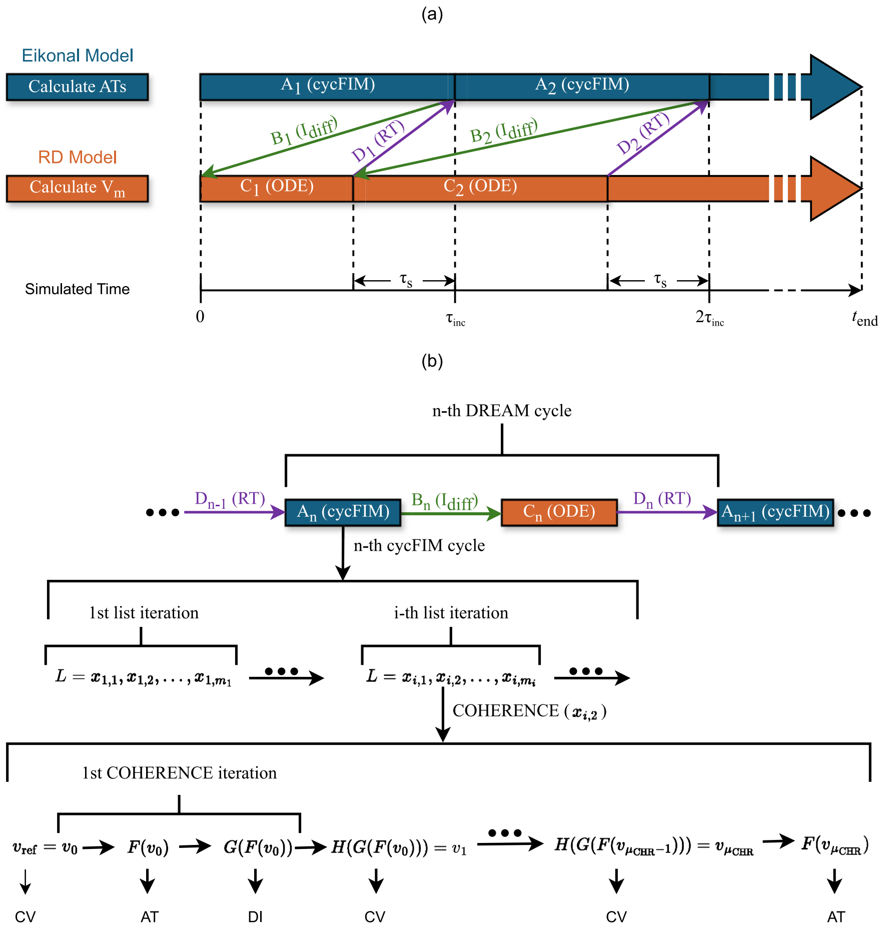


Fig. 1 **a** Steps of the diffusion reaction eikonal alternant model (DREAM) algorithm in simulated time: eikonal model and approximated reaction diffusion (RD) model alternate to calculate activation times (ATs) and transmembrane voltages (V_m), respectively. The cyclical fast iterative method (cycFIM) is used to calculate ATs (steps A_n). These ATs are utilized to compute I_{diff} needed in the RD model (steps B_n). The approximated RD model determines V_m by solving the ODE system of the ionic model and incorporating I_{diff} (steps C_n). V_m is used to get the repolarization times (RT) needed in the cycFIM to allow for reactivation when solving the eikonal equation (steps D_n). The subscript index n represents the cycle number in the sequence. The parameter τ_{inc} represents the increment of t_{min} (minimum AT of the nodes in L) with each cycFIM cycle. If L is never empty between stimuli, t_{min} tends to align approximately with multiples of τ_{inc}. The parameter τ_s is the temporal safety margin to avoid conflicts between DREAM cycles. **b** Steps of the DREAM algorithm, showing the sequence of a DREAM cycle. In step A_n, each call of the cycFIM is a cycFIM cycle, containing several list iterations. $x_{i,j}$ is the j-th node among m_i nodes in L at the start of the i-th iteration. Visiting a node calls COHERENCE(), leading to multiple COHERENCE() iterations. In each iteration, functions F, G, and H calculate AT, DI, and CV, respectively. After μ_{CHR} iterations, or upon convergence, the final output $F(v_{\mu_{CHR}})$ is used by the cycFIM

throughout the list iterations of a cycFIM. Once the total increment of t_{min} during a cycFIM cycle exceeds τ_{inc}, the cycFIM cycle ends (Fig. 1a), step A_1, Sect. 2.3.1). Hereby, τ_{inc} represents the increment of t_{min} each time cycFIM is called. Then, the determined ATs are used to calculate I_{diff}, which approximates the diffusion current expressed in the parabolic equation of the RD model. Then, I_{diff} triggers an action potential (AP) (Fig. 1a), step B_1 and Sect. 2.3.2), which allows to compute V_m in a low resolution mesh. The approximated RD model uses I_{diff} to compute V_m in all nodes of the mesh until $t = t_{min} - \tau_s$ (Fig. 1a), step C_1 and Sect. 2.3.3). τ_s is a safety margin in time, ensuring that only cells with converged ATs are stimulated by I_{diff}. Thereby, a reliable identification of repolarization times (RTs) is enabled. RTs are defined as the time points when V_m of an activated node X crosses the threshold of -40 mV with a negative slope. If V_m in an activated node does not reach this threshold before $t = t_{min} - \tau_s$, the ionic model is run independently for that node to determine its RT (Fig. 1a), step D_1 and Sect. 2.3.4). To start the next cycle in step A_2, the cycFIM iterates again until $t_{min} > t_{min,init} + \tau_{inc}$, where $t_{min,init}$ is the last calculated t_{min} of step A_1. Hereby, the previously

Table 1 DREAM parameters and variables: absolute times denote specific time points, with values spanning over the entire duration of the simulation

Absolute times		
Variable	Meaning	Value
$t_{\min}$	Minimum AT of nodes in L at the end of each list iteration	(–)[a]
$t_{\min,\mathrm{init}}$	Last $t_{\min}$ from the previous DREAM cycle	(–)[a]
t	Current time step in RD model	(–)[a]
Parameter	Meaning	Value
t_{end}	End of simulation	(–)[b]
Relative times		
Parameter	Meaning	Value
τ_{inc}	Maximum allowed increment of $t_{\min}$ in every DREAM cycle	100 ms
$\tau_{\max}$	Maximum allowed difference of AT values among nodes in L	10 ms
τ_{s}	Minimum allowed difference between $t_{\min}$ and t	10 ms
ε	Threshold of convergence	0.01 ms
Maximum iterations		
Parameter	Meaning	Value
μ_{L_1}	Maximum number of list iterations per node per entry into L	50
μ_{L_2}	Maximum number of node returns to L per activation	50
μ_{CHR}	Maximum number of COHERENCE() iterations per node per call	50

Relative times, fixed and typically smaller, constrain the occurrence of absolute times within specified intervals. Maximum permitted iteration parameters set limits for iterative processes within the cyclical FIM

[a] Variable values change throughout the simulation

[b] t_{end} changes across experiments

calculated RTs are considered by cycFIM to check whether a node can be reactivated. Then, steps B_2, C_2, and D_2 are performed in the same manner as steps B_1, C_1, and D_1. This process is repeated until t reaches t_{end}, corresponding to the end of the simulation time. Table 1 shows the variables and parameters of the DREAM including their meaning and values used in this work. The code to run simulations using the DREAM is available in the openCARP simulator [28]. In the next sections, each of the steps is explained in more detail.

2.3.1 Cyclical fast iterative method

The cycFIM is called in step A_n of the DREAM algorithm with $n \in \mathbb{N}$ representing the number of the cycle. Algorithm 2 describes this cycFIM. Some modifications were made to the single-thread FIM to allow alternation between the eikonal and RD models. Firstly, the boundary conditions for the eikonal equation need to be dynamically applied as their effect on the system varies depending on the refractory state of the nodes receiving the stimuli. Secondly, list iterations should handle RT and manage multiple ATs per node. Thirdly, the local solver must ensure coherence among the CV, DI, and AT for each node. If L is emptied before all the boundary conditions are computed, further aspects must be considered since $t_{\min}$ does no longer align with multiples of τ_{inc}(see Sect. 2.3.5).

Dynamic boundary conditions

Dynamic boundary conditions handle stimuli applied to specific areas of the tissue. These conditions vary dynamically as they depend on the refractory state of stimulated nodes when stimuli are applied. Unlike conventional eikonal models where boundary conditions are computed a priori irrespective of their timing, the cycFIM computes boundary conditions (i.e, stimuli) progressively during the simulation. For this reason, to compute boundary conditions in the cycFIM the order in which stimuli are applied must be considered. Let $T_1 < T_2 < \cdots < T_{n_s}$ be the times when stimuli -sets of nodes where this stimuli are applied. At the beginning of the n-th cycFIM cycle, let T_s, for $s \in \mathbb{N}$ and $1 \leq s \leq n_s$, be the time of the $s - th$ stimulus such that $T_1, \ldots, T_{s-1}$ have already been computed during the previous cycFIM cycles. Moreover, assume that $T_s, \ldots, T_{n_s}$ have not yet been computed. Therefore, T_s denotes the time of the earliest stimulus that has yet to be processed. Boundary conditions might be computed in 2 scenarios: at the beginning of each cycle if L is empty, or at the beginning of each list iteration if L is not empty.

In the first case, L can be empty for 2 reasons: the simulation has just started and $n = s = 1$ (i.e., the first stimulus must be computed during the first cycFIM cycle), or all activation times (ATs) associated with activation waves triggered by stimuli applied before T_s have converged, and L was emptied during the $(n-1)$-th cycFIM cycle. To incorporate new nodes into L and start a new list iteration, the s-th stimulus must be computed. In the second case, L is not empty and new stimuli might be applied. For instance, if T_s falls within the range of allowed activation times (ATs) for the nodes in L during this cycFIM cycle (i.e., $T_s \in [t_{\text{min,init}}, t_{\text{min,init}} + \tau_{\text{inc}} + \tau_{\max}]$), then the s-th stimulus can be computed. The parameter $\tau_{\max}$ which imposes an upper limit on the ATs of nodes in L will be explained in greater detail in the following subsection. Additionally, all the subsequent stimuli which times $T_{s+1}, T_{s+2}, \ldots$ are within this range, can be computed together.

To compute the s-th stimulus in both aforementioned cases, each node where the stimulus is applied is considered independently. Let $X \in \Gamma_s$ be a node where the s-th stimulus is applied at time T_s, and let $T(X)$ and $R(X)$ be its last computed AT and RT, respectively. Then, T_s is assigned to $T(X)$ if one of the following conditions is fulfilled:

$$\big(T(X) < R(X) < T_s\big) \vee \big(R(X) < T_s < T(X)\big). \tag{8}$$

In the first condition, the calculation for the present activation of X has not yet begun and that is the reason why $T(X) < R(X)$. In that case, the solution $T(X)$ is replaced if $T_s > R(X)$, indicating that at T_s, the node has already recovered from the previous activation and is ready to be activated again. In the second condition, $R(X) < T(X)$, i.e., the calculation for the present activation for X has already begun and it is still converging. The solution is updated if $R(X) < T_s$, signifying that the node X is ready to be activated at T_s and $T_s < T(X)$, which means T_s is a better (i.e., earlier) solution. The nodes with replaced ATs are removed from L, while their neighbors are added to L. To faithfully represent occurrence of unidirectional propagation, nodes that are successfully activated by a given stimulus cannot continue to activate nodes where an activation by this stimulus was unsuccessful before. Additionally, a delay of 5 ms in RT was added to nodes in areas that were unsuccessfully stimulated.

List iteration

The variable $t_{\text{min,init}}$ represents the minimum AT among the nodes in L reached at the end of the previous DREAM cycle. When L is not empty after computing the first stimulus, $t_{\text{min,init}}$ is initialized to 0 during the first call of cycFIM in the first cycle, or adjusted to the value of $t_{\min}$ calculated at the end of step A_{n-1} if $n > 1$. Additionally, $t_{\text{min,init}}$ serves as a reference to be compared to $t_{\min}$, the smallest AT in L at the end of the previous cycFIM cycle. L undergoes multiple iterations until $t_{\min} > t_{\text{min,init}} + \tau_{\text{inc}}$. Then, the function COHERENCE() (see next subsection) is applied to each node in L. Let X be the node that is being revised and $T(X)$ and $R(X)$ its last computed AT and RT, respectively. Then, a new solution q is computed by COHERENCE(X) to potentially replace $T(X)$. If the difference between the old and new AT is smaller than the threshold ε, or either the old or the new solution is infinite, then X is removed from L. To avoid that a node is activated twice in the same iteration of the list, an upper bound $\tau_{\max}$ is applied to the difference between the maximum AT of the nodes in L and $t_{\min}$. If a certain node has a calculated AT above the maximum allowed AT for a given iteration of cycFIM, i.e., $q > t_{\text{min,init}} + \tau_{\text{inc}} + \tau_{\max}$, then this node will not be removed from L and its neighbors will not be visited yet.

When a node is removed from L, its neighbors that are not in L are revised as well. Let X_{NB} be a neighbor of X with last computed AT and RT being $T(X_{\text{NB}})$ and $R(X_{\text{NB}})$, respectively. If X_{NB} is not in L and a new solution q_{NB} is computed, one of the following conditions (similar to those for stimuli) must be fulfilled for q_{NB} to replace $T(X_{\text{NB}})$:

$$\big(T(X_{\text{NB}}) < R(X_{\text{NB}}) < q_{\text{NB}}\big) \vee \big(R(X_{\text{NB}}) < q_{\text{NB}} < T(X_{\text{NB}})\big). \tag{9}$$

In the first condition, the calculation for the present activation of X_{NB} has not yet begun. In that case, the solution is replaced (i.e., q_{NB} is assigned as a new tentative AT for X_{NB}) if $q_{\text{NB}} > R(X_{\text{NB}})$, meaning that the node has recovered at q_{NB}. In the second condition, $R(X_{\text{NB}}) < T(X_{\text{NB}})$, i.e., calculation of the present activation has already started for X_{NB} and is still converging. The solution is replaced if $\text{RT}_{X_{\text{NB}}} < q_{\text{NB}} < T(X_{\text{NB}})$, which means that the node is ready to be activated at q_{NB} and q_{NB} is a better (i.e., earlier) solution. If q_{NB} satisfies either of these two conditions and replaces $T(X_{\text{NB}})$, then X_{NB} is added to the list of active nodes, L. Regardless of the previous conditions, if q_{NB} is infinite or smaller than t, X_{NB} will not be added to L. When all the nodes in L and the neighbors of converging nodes have been revised, $t_{\min}$ is recalculated to decide whether further iterations are required in step A_n. To reduce the computational cost and to prevent nodes from indefinitely exiting and returning to L or iterating inside of L, 2 limits were defined: μ_{L_1} and μ_{L_2}. The parameter μ_{L_1} represents the maximum number of list iterations per node per entry into L, and μ_{L_2}, denotes the maximum number of returns to L per node per activation.

Algorithm 2 Cyclical Fast Iterative Method within the n-th DREAM cycle

```
if $L = \emptyset$
    Compute boundary conditions
end if
if $n = 0$
    $t_{\text{min,init}} = 0$
else
    $t_{\text{min,init}} = t_{\text{min}}$
end if
repeat
    Compute boundary conditions
    for $X \in L$
        $p \leftarrow T(X)$
        $q \leftarrow \text{COHERENCE}(X)$
        $T(X) \leftarrow q$
        if $|p - q| < \varepsilon$
            for adjacent neighbor $X_{\text{NB}}$ of $X$
                if $X_{\text{NB}}$ is not in $L$
                    $p \leftarrow T(X_{\text{NB}})$
                    $q_{\text{NB}} \leftarrow \text{COHERENCE}(X_{\text{NB}})$
                    if $(p < R(X_{\text{NB}}) < q_{\text{NB}} \vee R(X_{\text{NB}}) < q_{\text{NB}} < p) \wedge (t < q_{\text{NB}} < \infty)$
                        $T(X_{\text{NB}}) \leftarrow q_{\text{NB}}$
                        add $X_{\text{NB}}$ to $L$
                    end if
                end if
            end for
            remove $X$ from $L$
        end if
    end for
    $t_{\text{min}} = \min\{T(X)|X \in L\}$
until $L = \emptyset$ OR $(t_{\text{min}} - t_{\text{min,init}}) \geq \tau_{\text{inc}}$
```

Coherence between conduction velocity, diastolic interval and activation time

The local solver for the DREAM seeks to calculate the AT for each node. Unlike the standard eikonal model, the DREAM accounts for the CV restitution at each node in each activation cycle instead of a fixed CV. To implement CV restitution, the function UPDATE() in Algorithm 1 is replaced by a new function COHERENCE() in Algorithm 2. The CV restitution is incorporated in the model by providing a template restitution curve, for example calculated from monodomain model simulations or inferred from clinical data [7]. To integrate this CV restitution phenomenon into the DREAM, the feedback loop among AT, CV, and DI must be acknowledged. This feedback loop means that calculating the AT depends on the CV, which relies on the DI, which in turn is influenced by the AT.

When applying the COHERENCE() function to a given node X, the reference CV of X, the RT of X from the previous activation, and ATs of the neighboring nodes in the present activation are known. The reference CV corresponds to the maximum possible CV, i.e. $v_1(X)$ when the DI is sufficiently long and CV becomes unaffected by restitution effects. Restitution curves with biphasic behavior, where the maximum CV is reached at intermediate DIs instead of long DIs, are excluded from this method. If an initial AT is estimated using the highest value of the reference CV, this AT will have the smallest (i.e., earliest) possible value, resulting in the shortest possible DI. If this DI is short, when applying the restitution curve, this leads to a slower CV than the reference CV and therefore in a higher (i.e., later) AT than the initially estimated AT. To calculate a tentative AT for X, the COHERENCE() function is utilized to iteratively recalculate CV, DI, and AT until a stable state of coherence between these 3 variables is reached. Note that these temporary values of the 3 variables are referring to the present activation of X. These 3 values are recalculated in each COHERENCE() iteration (not to be confused with list iteration see Fig. 1b). Each application of the local solver involves one or more COHERENCE() iterations, starting with the reference CV and ending with a tentative value for $T(X)$ to be used in cycFIM. Additionally, the

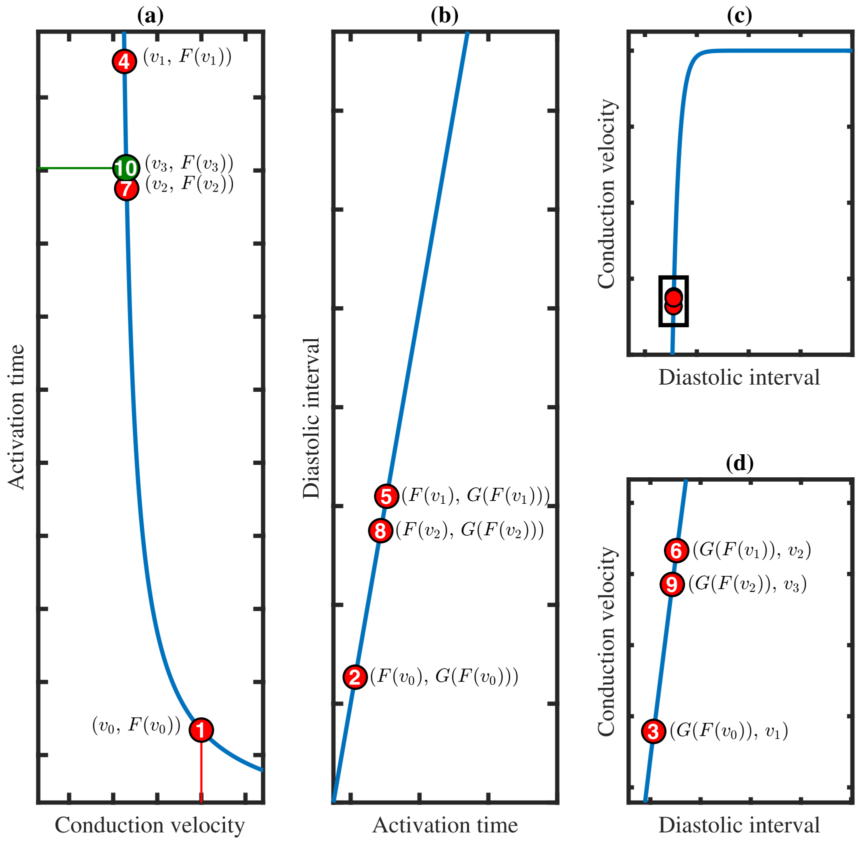


Fig. 2 COHERENCE() call with 3 iterations. Numbers indicate the sequence in which activation times (ATs), diastolic intervals (DIs), and conduction velocities (CVs) are calculated. Red circles correspond to temporary values and the green circle indicates the final output. **a** Function F mapping CV to AT. The vertical red line marks the initial input of COHERENCE() (i.e., $v_{\text{ref}} = v_0$) on the horizontal axis. The green red line marks the final output of COHERENCE() (i.e, $F(v_3)$) on the vertical axis, **b** Function G mapping AT to DI, **c** Function H mapping DI to CV, **d** Zoom-in of the boxed area in **c**, illustrating temporary DI and CV values (color figure online)

term CV can refer to the speed of the wavefront itself, the longitudinal CV when the wavefront moves in the preferential cardiomyocyte orientation, or the transversal CV when the wavefront moves perpendicularly to this orientation. Although the COHERENCE() function utilizes the longitudinal CV (i.e., $v_1(X)$) for its computation and equilibrium is reached among AT, DI, and longitudinal CV, this equilibrium also extends to the wavefront's CV.

COHERENCE() calculates the tentative AT of X using the functions F, G and H. Each of these functions also depends on X. To simplify notation, X as an argument of these functions will be omitted. Let $F : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ be the function that computes the AT from the CV:

$$F(v^*) = t^*, \tag{10}$$

where v^* is a temporary value for $v_1(X)$ and t^* is a temporary solution to the eikonal equation (Eq. 3) at node $X \in \Omega$ providing a tentative AT for this node. To solve the eikonal equation, Eq. 6 is used to collect all tentative candidates for t^*. The function F is similar to the UPDATE() function of FIM but considers additional conditions. Propagation block can occur in cycFIM, implying that $v = 0$, thus $F(0) = \infty$. On the other hand, when taking into account the neighboring nodes Y and Z for Eq. 6, the triangle XYZ is not considered if $R(Y) > T(Y)$ or $R(Z) > T(Z)$. This implies that the ATs for these nodes are not yet computed for the present activation cycle and, therefore, must be excluded from the calculation of t^*. Figure 2a shows a schematic representation of function F()'s morphology.

Now, let $G : \mathbb{R}_{\geq 0} \cup \{\infty\} \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$ be the function that computes the DI from t^*:

$$G(t^*) = d^* = t^* - R(X), \tag{11}$$

where $R(X)$ is the RT of node X from the previous activation cycle and d^* is a temporary value for DI in the present activation cycle. Figure 2b shows a schematic representation of function G()'s morphology.

Finally, let $H : \mathbb{R}_{\geq 0} \cup \{\infty\} \rightarrow \mathbb{R}_{\geq 0}$ be the function that computes the new temporary CV v^* from the temporary DI d^*:

$$H(d^*) = v^* = \begin{cases} v_{\text{ref}} \cdot \left(1 - \rho \cdot e^{-\frac{\log(\rho)}{\psi}(d^* + \kappa)}\right) & \text{if } d^* > \theta, \\ 0 & \text{if } d^* \leq \theta. \end{cases} \tag{12}$$

Here, v_{ref} represents the reference CV, while ρ, κ, θ, and ψ are predefined parameters chosen to fit restitution curves for example obtained from monodomain simulations. The formulation in Eq. 12 allows expressing the steepness of the restitution curve in terms of ρ, the shift in the x-axis (i.e., DI) in κ and the shortest propagating DI in θ. Figure 2c shows a schematic representation of function H()'s morphology.

A recursive sequence (v_i) can be defined using the composition of the functions H, G and F:

$$\begin{aligned} v_0 &= v_{\text{ref}}, \\ v_{n+1} &= H(G(F(v_n))). \end{aligned} \tag{13}$$

The COHERENCE() function approximates the AT that corresponds to the CV value which is the limit of the sequence defined in Eq. 13:

$$\text{COHERENCE}(X) \approx F(\lim_{n\to\infty} v_n). \tag{14}$$

Since we cannot calculate this limit analytically, the function is defined as:

$$\text{COHERENCE}(X) = F(v_m) \text{ such that } |F(v_m) - F(v_{m-1})| < \varepsilon \text{ OR } m = \mu_{\text{CHR}}, \tag{15}$$

where μ_{CHR} is the maximum number of COHERENCE() iterations allowed per call. Figure 2 illustrates an example of one COHERENCE() call with 3 iterations. The input and first element of the sequence is $v_0 = v_{\text{ref}}$, and the final output is $F(v_3)$, representing the AT calculated with v_3, corresponding to the CV obtained in the third iteration. This ensures that $|F(v_3) - F(v_2)| < \varepsilon$. The function typically converges quickly, often within 2 iterations, but an extreme case close to propagation failure was chosen in the example to demonstrate the convergence process more clearly. Moreover, the example was chosen to ensure that the function F has its simplest form, $t^* = a + \frac{b}{v^*}$, where a is the minimum AT among the neighbors of X, and b is the minimum possible value for $(\overrightarrow{X_\lambda X}^\top \mathbf{D}^{-1} \overrightarrow{X_\lambda X})^{\frac{1}{2}}$ in Eq. 6 among all triangles in $\mathcal{T}$ containing X and all $\lambda \in [0, 1]$ (Fig. 2a). If $\mathbf{D}$ is the identity matrix (i.e, in the isotropic case), then b is the minimum Euclidean distance between X and the perimeter of the polygon formed from the union of all triangles containing X. In this example, the function F is monotonically decreasing, continuous, and differentiable within $(0, +\infty)$. For this to occur, the neighboring node with the minimum AT must belong to the triangle where the minimum $(\overrightarrow{X_\lambda X}^\top \mathbf{D}^{-1} \overrightarrow{X_\lambda X})^{\frac{1}{2}}$ is found. In the general case where this condition is not necessarily met, the function F remains monotonous and continuous but may not be differentiable everywhere. Additionally, $F(v^*)$ tends to $+\infty$ as v^* approaches 0, and $F(v^*)$ tends to a as v^* approaches $+\infty$ for any situation.

The proof of convergence of the sequence described in Eq. 13 is based on the idea that the sequence can be partitioned into 2 converging subsequences: one increasing (v_{2n}) and the other decreasing (v_{2n+1}). Furthermore, these subsequences are bounded between v_0 and v_1, and the difference between v_n and v_{n+1} diminishes as n approaches infinity. In some cases, the sequence might need to take values of the DI below the parameter θ before converging. Therefore, when computing COHERENCE(), DI values below θ are allowed as long as the final DI is larger than θ, otherwise COHERENCE(X) = ∞.

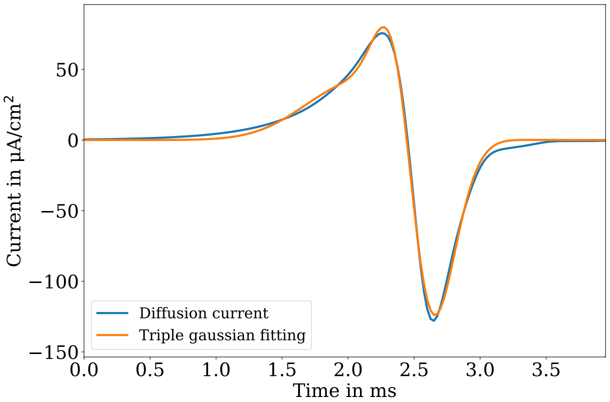


Fig. 3 I_{diff} fitted to the diffusion current obtained from a pure monodomain simulation of a planar wave on a high-resolution mesh. A triple Gaussian was used to approximate the diffusion current. Adapted from [19]

Table 2 Diffusion parameters I_{diff} for the models by Courtemanche et al., Bueno-Orovio et al., and Mitchell and Schaeffer used in the DREAM simulations, corresponding to each of the three Gaussian indices i

Parameter	i	Courtemanche et al./ Bueno-Orovio et al	Mitchell and Schaeffer
α_i (μA/cm²)	1	149.5	17.59
	2	41.2	− 228.91
	3	− 192.2	218.50
β_i (ms)	1	2.383	4.92
	2	2.07	5.19
	3	2.56	5.08
γ_i (ms)	1	0.22	0.49
	2	0.56	0.92
	3	0.29	0.92

2.3.2 Approximation of the diffusion current

To allow for V_{m} calculation in coarse meshes, a current I_{diff} was introduced following the idea of the RE model [10]. In step B$_n$, I_{diff} is computed by approximating $\nabla \cdot (\sigma_{\text{m}} \nabla V_{\text{m}})$ in Eq. 1. A similar approach is used in the eikonal model by Gassa et al. [14]. In the DREAM, I_{diff} is defined as a triple Gaussian function:

$$\nabla \cdot (\boldsymbol{\sigma}_\mathrm{m} \nabla V_\mathrm{m})\Big|_X \approx I_\mathrm{diff}(X) = \sum_{i=1}^{3} \alpha_i \exp\left(-\frac{(t - T(X) - \beta_i)^2}{\gamma_i^2}\right), \tag{16}$$

where the parameters α_i, β_i and, γ_i are optimized to reduce the difference between I_diff and the diffusion current specific to the used ionic model. A high-resolution mesh (i.e., average edge length 200 μm) was used to run a monodomain simulation in the optimization process. A planar wavefront was simulated by stimulating at the mesh border. The conductivities were adjusted to obtain a CV of 1000 mm/s. The diffusion current was calculated at a node located far from the boundaries and the stimulus location. I_diff serves as a trigger for an AP, regardless of whether the activation is initiated directly by a stimulus or through diffusion from neighboring nodes. Figure 3 presents the fit of the triple Gaussian model, using the parameters listed in Table 2, to the diffusion current obtained from the monodomain simulation as described above. Since the ionic models by Courtemanche et al. and Bueno-Orovio et al. yield AP and diffusion currents with similar morphology, the same set of parameters was used for I_diff with both models.

2.3.3 Approximated reaction diffusion model

After the n-th cycFIM cycle is finished and I_diff is computed, the approximated RD model will run (i.e., step C_n). The time interval for this step is: $[t_\mathrm{min,init} - \tau_\mathrm{s}, \min(t_\mathrm{min} - \tau_\mathrm{s}, t_\mathrm{end})]$ ensuring that the nodes that trigger an AP during this step have converged ATs. The diffusion term in the parabolic equation of the RD system is replaced by I_diff:

$$\beta C_\mathrm{m} \frac{\partial V_\mathrm{m}}{\partial t} = I_\mathrm{diff} - \beta I_\mathrm{ion}(V_\mathrm{m}, \vec{\eta}). \tag{17}$$

In this case, the diffusion current is approximated by I_diff and only the reaction (ODE) part of the RD system remains to be solved. Solving this ODE system independently for each node in $\Omega_\mathcal{T}$ will yield the values for V_m within the interval $[t_\mathrm{min,init} - \tau_\mathrm{s}, \min(t_\mathrm{min} - \tau_\mathrm{s}, t_\mathrm{end})]$.

2.3.4 Identification of repolarization times

Before the $(n+1)$-th cycFIM cycle starts, it is important to identify which nodes and when are available for reactivation. Ideally, V_m of every activated node would reach its RT during step C_n. However, recently activated nodes will not have recovered at the end of this RD iteration (i.e., at $t_\mathrm{min} - \tau_\mathrm{s}$). The aim in step D_n is to compute RTs of these nodes. To obtain these times, the ODEs of the single cell models are integrated in time until the threshold −40 mV is crossed:

$$C_\mathrm{m} \frac{\mathrm{d}V_\mathrm{m}}{\mathrm{d}t} = -I_\mathrm{ion}(V_\mathrm{m}, \vec{\eta}). \tag{18}$$

This process is implemented for each node that was depolarized but not yet repolarized in the previous step. The initial conditions are defined by the state variables at $t = t_\mathrm{min} - \tau_\mathrm{s}$ for each node. After the RT is calculated, V_m and $\vec{\eta}$ are reset to the values they had at $t = t_\mathrm{min} - \tau_\mathrm{s}$.

2.3.5 Considerations for empty list scenarios

Additional considerations are required if L is empty but not all the boundary conditions have been computed. Let T_s be the time when the next stimulus will be applied, then V_m is computed until $t = T_s - \tau_\mathrm{s}$. After identifying the RTs, the stimulus at T_s must be computed before iterating cycFIM again as detailed in Sect. 2.3.1. If L is not empty after computing the stimulus, cycFIM iterates until $t_\mathrm{min} > T_s + \tau_\mathrm{inc}$. Given that the list was empty at the end of the previous cycle, it is no longer feasible to establish $t_\mathrm{min,init}$. Hence, T_s is employed as the reference $t_\mathrm{min,init}$ to monitor the increment of t_min. If all nodes in Γ_s (i.e., the region in which the s-th stimulus is applied) are in the refractory period at time T_s, the list will remain empty. In that case, V_m is computed until $t = T_{s+1} - \tau_\mathrm{s}$, where T_{s+1} is the time of the $(s+1)$-th stimulus. If necessary, subsequent boundary conditions (i.e., stimuli applied at times $T_{s+2}, \ldots, T_{n_s}$) are computed until the list L gain nodes or $t = t_\mathrm{end}$.

3 Benchmarking

To assess the DREAM, it was compared with the monodomain model in 3 numerical experiments. First, multiple simulations were run on a 2D rectangular tissue mesh, each with 2 consecutive planar wavefronts per pacing cycle length (PCL) (i.e., interval between wavefronts). The PCL was varied across simulations to analyze CV restitution curves. In the second numercial experiment, functional reentry was examined in absence of structural abnormalities, i.e. their occurrence attributed solely to slow CVs and short ERPs. These reentries were induced using an S1-S2 protocol in a 2D square mesh. In the third numerical experiment, the pacing at the end of the effective refractory period (PEERP) protocol was used to investigate reentry in a realistic geometry of a left atrium [29]. These 3 numerical experiments were conducted using meshes composed of triangular elements with 4 different resolutions. The average edge lengths were 200, 400, 800, and 1600 μm. To facilitate the comparison between different meshes, each node in the lower resolution meshes was mapped to a corresponding node at the same position in the higher resolution meshes. To

Table 3 Conduction velocity restitution parameters for function H() in healthy and atrial fibrillation (AF) cases for ionic models by: Courtemanche et al., Bueno-Orovio et al., and Mitchell and Schaeffer, used in the DREAM simulations

	Courtemanche et al.		Bueno-Orovio et al.		Mitchell and Schaeffer	
Parameter	Healthy	AF	Healthy	AF	Healthy	AF
ρ (−)	811	100,000	55.13	100,000	4.41	2.90
κ (ms)	53	116.41	138.14	118.16	161.47	151.75
θ (ms)	137	66.28	139.73	70.64	129.30	103.12
ψ (ms)	159	159	159	159	159	159

execute monodomain simulations, tissue conductivities were adjusted to match the CV values desired for each scenario [30, 31]. Anisotropy ratios were set to 4 and 2 for tissue conductivity and CV respectively. The Courtemanche et al. model with standard parameters was used to represent the healthy ionic behavior of atrial cardiomyocytes in the three numerical experiments [32]. On the other hand, atrial fibrillation (AF) was modeled by modifying ion channel conductances of the Courtemanche et al. model [33]. To additionally assess the compatibility of the DREAM with different ionic models, the first numerical experiment (i.e., 2D rectangular tissue mesh) was also computed utilizing the simplified ionic models of Bueno-Orovio et al. [34] and Mitchell and Schaeffer [35]. These ionic models were originally developed for ventricles. Therefore, their parameters were adapted for atrial tissue to reflect cellular behavior of healthy and AF remodeled cardiomyocytes. The Bueno-Orovio et al. model was modified as proposed by Lenk et al. [36] providing parameter sets for both conditions. The Mitchell and Schaeffer ionic model was adjusted according to Gassa et al. [14] for healthy atrial cardiomyocytes and case 3 in Corrado et al. [37] for AF remodeled cardiomyocytes.

CV restitution curve parameters ρ, κ, and θ (Eq. 12) were tuned to align the CV restitution curve of the DREAM with either healthy or AF in monodomain simulations. This adjustment was based on the CV restitution curves obtained from monodomain simulations in a mesh with an average edge length of 200 μm (i.e., high resolution mesh) with 2 planar wavefronts resulting in the parameters in Table 3.

The Crank–Nicolson method was used to solve the parabolic equation of the monodomain model. Moreover, the methods described in Sect. 2.3 were used to solve the eikonal equation and approximate the diffusion current in the DREAM. The ODEs of the ionic model were solved using the forward Euler method for V_{m} and the Rush-Larsen method for the gating variables in both the monodomain model and the DREAM.

Sheet geometries were created using Gmsh [38] and processed with ParaView [39], Meshmixer [40] and MeshLab [41]. openCARP [30] was used to run simulations with both the DREAM and monodomain model. Plots were generated using the Python library matplotlib [42].

Specifically, the optimization of I_{diff} parameters was performed using MATLAB, utilizing the curve fitting toolbox. The optimization of the CV restitution parameters was conducted in Python, employing the *minimize* function from SciPy's optimize package [43]. In both optimization processes, the root mean square error was used as a loss function.

3.1 Multiple stimulations and CV restitution

To assess the DREAM's ability to faithfully replicate CV restitution, we initially examined its capacity to simulate paced beats across a range of frequencies specific for each condition. These experiments were conducted in a rectangular mesh of 50 mm × 10 mm × 0 mm, with the centroid of the mesh located at (0 μm, 0 μm, 0 μm) for each of the 4 resolutions. Preferential cardiomyocyte orientation was set as constant everywhere in the geometry and aligned with the x-axis (i.e., $\vec{l} = (1, 0, 0)$). Each experiment involved prepacing at a single-cell level for 50 activation cycles at a cycle length of 250 ms. For monodomain simulations, conductivities were tuned to match the target CV of 1000 mm/s.

In this numerical experiment, 2 planar wavefronts were generated for each PCL by stimulating the left side of the mesh with different frequencies. For monodomain simulations, the stimulus was applied as a transmembrane current density with an amplitude of 300 $\mu\mathrm{A/cm}^2$ for a duration of 2 ms. For the healthy atrial electrophysiology condition with the ionic model of Courtemanche et al., the PCL was decreased in 50 ms intervals from 950 to 400 ms, then in 10 ms intervals from 390 to 320 ms, and finally in 1 ms intervals from 320 to 318 ms. The different intervals were chosen to cover critical changes in electrophysiological behavior, as CV and ERP drastically decrease towards the last propagating PCL. For the ionic models of Bueno-Orovio et al. and Mitchell and Schaeffer, also shorter PCLs propagated on the mesh. Therefore, the PCL was further decreased in 1 ms intervals to 250 ms. In the AF condition, the PCL was decreased for all ionic models from 950 ms to 200 ms in 50 ms intervals, and from 199 to 118 ms in 1 ms intervals. The considered PCL values varied across conditions due to differences in the restitution curves, requiring a range of PCL values tailored to each condition.

After the experiments, the CV restitution curves were determined. To calculate the CV, the ATs of the mesh points $(-15,000\,\mu\text{m}, 0\,\mu\text{m}, 0\,\mu\text{m})$ and $(15,000\,\mu\text{m}, 0\,\mu\text{m}, 0\,\mu\text{m})$ were used. CV was normalized over v_{ref}. The ERP was defined as the time difference between ATs and RTs from the second activation cycle plus θ from Eq. 12 (i.e., shortest propagating DI).

3.2 Reentry in 2D sheets

To evaluate the impact of mesh resolution on reentry properties in both models, a 2D sheet geometry was subjected to S1-S2 protocols. The geometry was represented by squared unstructured meshes of size 51.2 mm × 51.2 mm × 0 mm, with the centroid of the mesh located at $(25,600\,\mu\text{m}, 25,600\,\mu\text{m}, 0\,\mu\text{m})$ for each of the 4 resolutions. Preferential cardiomyocyte orientation was set as constant everywhere in the geometry and aligned with the x-axis (i.e., $\vec{l} = (1, 0, 0)$). Experiments were conducted for homogeneous reference CV values of 200, 600 and 1000 mm/s. The Courtemanche et al. ionic model was pre-paced at 250 ms PCL for 50 activation cycles. The tissue was pre-paced 5 times with planar stimuli (S1) from the left border of the sheet. Subsequently, a single cross-field stimulus (S2) was applied in the bottom left quadrant at various times to induce reentry. After each simulation, it was assessed whether S2 was applied too early and got completely blocked, applied too late and propagated to all directions without any block, or if unidirectional block occurred as a prerequisite to induce reentry. Reentry duration was defined as the period from S2 application to the last AT in the simulation. ΔS was defined as the difference between the times when S2 and the last S1 were applied. The vulnerable window duration was determined as the difference between the earliest and latest ΔS values that induced reentry. The sample frequency for ΔS values was 1 ms. The mean and standard deviation of the local reentry cycle length were calculated at 4 nodes, each located at the same coordinates in all 4 mesh resolutions: $P_{\text{left}} = (12,800\,\mu\text{m}, 25,600\,\mu\text{m}, 0\,\mu\text{m})$, $P_{\text{down}} = (25,600\,\mu\text{m}, 12,800\,\mu\text{m}, 0\,\mu\text{m})$, $P_{\text{right}} = (38400\,\mu\text{m}, 25,600\,\mu\text{m}, 0\,\mu\text{m})$, and $P_{\text{up}} = (25,600\,\mu\text{m}, 38400\,\mu\text{m}, 0\,\mu\text{m})$. To calculate the local reentry cycle length, reentries were induced using DREAM and monodomain simulations with a reference conduction velocity (CV) of 200 mm/s and a stimulus interval (ΔS) of 225 ms in the 4 resolutions. The local reentry cycle length was defined as the difference between the ATs corresponding to 2 consecutive activation cycles of the same node after reentry was induced. To evaluate the long-term stability beyond 1 s of reentry, the simulation was run until the total simulated time reached 60 s. For each of the 438 cycles within this period, the mean and standard deviation of the local cycle length across all nodes were calculated.

3.3 Reentry in the left atrium

The PEERP method [29] was used to induce reentries in a realistic human left atrial geometry from a publicly available dataset [44]. This realistic geometry is derived from an instance of a statistical shape model [45]. Preferential cardiomyocyte orientation was assigned using a rule-based method [46]. The endocardial surface was extracted and remeshed at 4 different resolutions to achieve the desired average edge lengths. To benchmark the DREAM on coarse meshes, the adaptations detailed below were introduced; the remaining parameters were sourced from Azzolin et al. [29]. Modifications were applied equally to both DREAM and monodomain experiments. From the coarsest mesh (1600 µm average edge length), 21 points that were evenly spaced approximately 2 cm apart were selected. Since the lower resolution meshes were embedded in the higher resolution ones, each of the 21 points were located at positions where each of the 4 meshes had a corresponding node. The reference CV was reduced to 200 mm/s, and the conductances of the ionic model were adapted to represent AF to increase the likelihood of inducing reentry without the need for an additional arrhythmogenic substrate. Pre-pacing on single cell and tissue level was performed for 50 and 5 activation cycles, respectively, at a PCL of 250 ms. From the APs triggered by the last tissue pre-pacing, the RT was calculated and the ERP was defined as RT+θ. Afterwards, the tissue was stimulated at the end of the ERP to try to induce reentry at each of the 21 chosen locations. Stimulation comprised all nodes within a 5 mm-radius sphere centered at this location. Stimulations were stopped if a reentry was induced or the maximum number of 4 stimuli per location was reached.

3.4 Computing times

The simulations were executed on a 2023 MacBook Pro, equipped with a 11-Core Apple M3 Pro processor and 16 GB 6.4 GHz LPDDR5 memory. Computing times were benchmarked in 2 experiments across 4 different resolutions using the AF remodeled ionic model of Courtemanche et al. and a CV of 200 mm/s. In the first experiment, one single planar wavefront was stimulated in each of the 2D sheets described in Sect. 3.2. The computing time was measured from the onset of the stimulus until 500 ms of simulated time thereafter. The selected duration encompassed the repolarization phases of all nodes. In the second experiment, a reentry was induced using the S1-S2 protocol presented in Sect. 3.2. The computing time was measured starting 200 ms of simulated time after delivering the S2 stimulus and ending 500 ms of simulated time after delivering the S2 stimulus, i.e., in the interval [S2+200 ms, S2+500 ms].

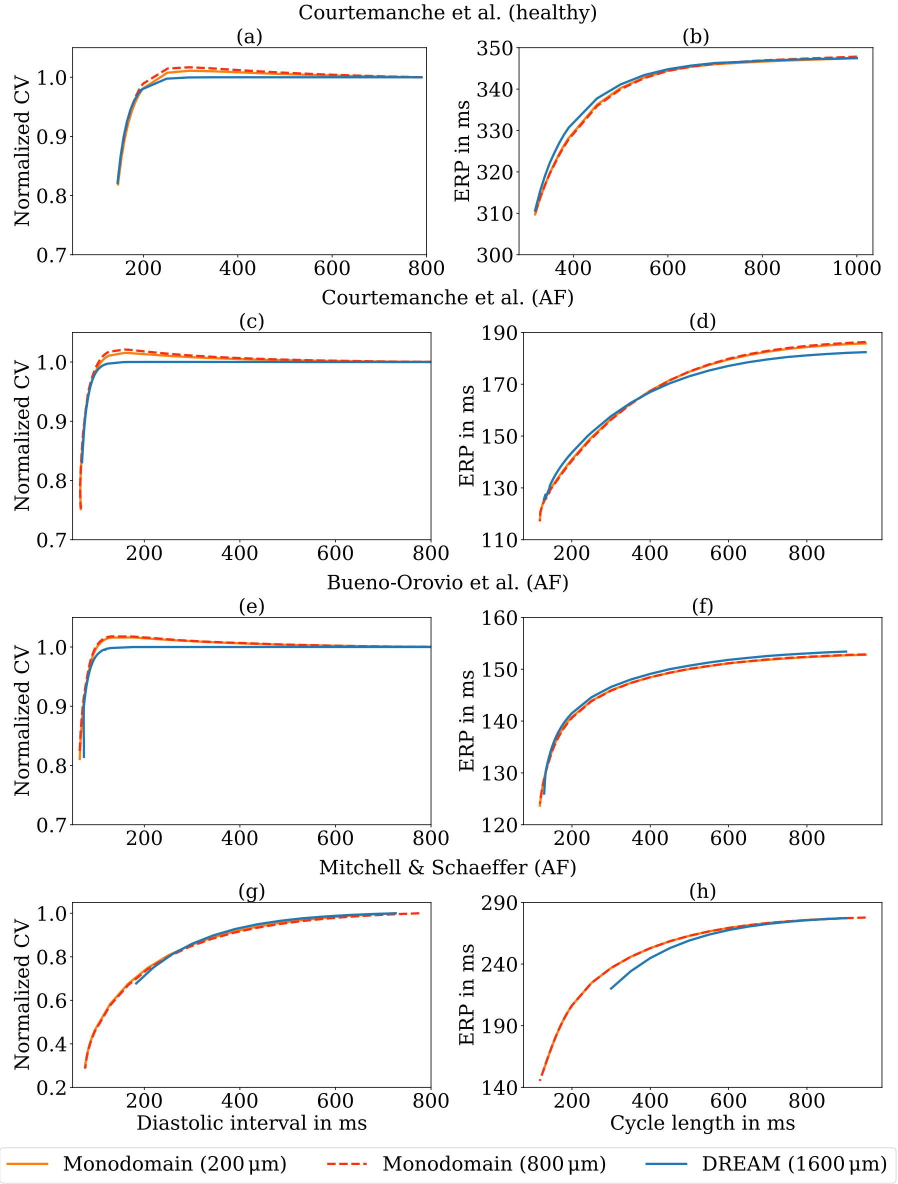


Fig. 4 Restitution curves for conduction velocity (CV) and effective refractory period (ERP) obtained from experiments using meshes with average element edge lengths of 200 µm and 800 µm for the monodomain model, and 1600 µm for the DREAM, with the healthy and AF remodeled Courtemanche et al. ionic model, as well as the AF remodeled ionic models of Bueno-Orovio et al. and Mitchell and Schaeffer. Results for the monodomain model with an edge length of 400 µm are not shown due to negligible differences compared to the 200 µm results, and the monodomain model with 1600 µm edge length is excluded due to propagation failure. Finer resolutions for the DREAM (i.e., 200 µm, 400 µm, and 800 µm) are not displayed as they are mostly indistinguishable from the 1600 µm results. **a, c, e, g** CV restitution curves normalized to the reference CV, $v_{\text{ref}} = 1000\,\text{mm/s}$ and **b, d, f, h** ERP restitution curves for the Courtemanche et al. ionic model representing healthy and AF remodeled behavior, as well as the AF remodeled ionic models of Bueno-Orovio et al. and Mitchell and Schaeffer, respectively

4 Results

4.1 Multiple stimulations and CV restitution

Figure 4 shows the comparison of CV and ERP restitution curves between the DREAM and monodomain models at resolutions of 200, 400, 800, and 1600 µm. The analysis reveals distinct characteristics for both the healthy and AF cases. Specifically, the ionic model of Courtemanche et al. was used for the healthy and AF remodeled case, whereas the AF remodeled case employed the ionic models of Bueno-Orovio et al. and Mitchell and Schaeffer. Table 4 shows the root mean square error (RMSE) of CV and ERP restitution curves for all considered ionic models with both healthy and AF conditions. Monodomain simulations on meshes with an average edge length 200 µm were used as reference solution to calculate RMSE. Addressing simulations with the Courtemanche et al. ionic model, DREAM and monodomain model exhibited similar steepness across all resolutions for the healthy case in both restitution curves (Fig. 4a, b), with RMSEs below 8.85 mm/s and 1.52 ms respectively. In the AF case (Fig. 4c, d), both propagation models maintained similar steepness and shortest propagating DI across resolutions. However, the DREAM notably did not represent the temporary raise of CV during intermediate DIs, while the monodomain model showed varying levels of biphasic restitution across resolutions, with lower resolutions having a higher maximum CV at intermediate DIs. The monodomain model demonstrated smaller last propagating cycle lengths, indicating the shortest possible

Table 4 Root mean square error of conduction velocity (CV) restitution and effective refractory period (ERP) restitution for healthy and atrial fibrillation (AF) conditions, in DREAM and monodomain model with Courtemanche et al., Bueno-Orovio et al., and Mitchell and Schaeffer embedded ionic models at various resolutions indicated by average edge length, compared to the restitution obtained with the monodomain model at 200 µm

			CV (mm/s)		ERP (ms)	
Ionic model	Propagation model	Resolution (µm)	Healthy	AF	Healthy	AF
Courtemache et al.	DREAM	200	8.85	7.19	1.52	2.39
		400	8.74	7.14	1.52	2.39
		800	8.47	7.14	1.50	2.39
		1600	8.01	7.26	1.49	2.38
	Monodomain[a]	200	0.00	0.00	0.00	0.00
		400	0.63	2.60	0.36	0.09
		800	5.88	5.17	0.25	0.46
		1600	(–)	(–)	(–)	(–)
Bueno-Orovio et al.	DREAM	200	7.21	15.95	1.03	1.11
		400	6.62	13.60	1.09	1.12
		800	6.48	11.49	1.02	1.12
		1600	7.07	12.08	1.05	1.17
	Monodomain[a]	200	0.00	0.00	0.00	0.00
		400	0.98	0.30	0.49	0.08
		800	14.36	4.13	1.53	0.18
		1600	(–)	(–)	(–)	(–)
Mitchell and Schaeffer	DREAM	200	17.36	10.82	7.09	6.38
		400	18.21	8.01	7.21	6.44
		800	17.65	7.03	7.22	6.46
		1600	19.28	7.76	7.24	6.46
	Monodomain[a]	200	0.00	0.00	0.00	0.00
		400	2.80	0.30	0.54	0.03
		800	2.20	5.33	0.52	0.04
		1600	(–)	(–)	(–)	(–)

[a] Propagation failure at 1600 µm average edge length

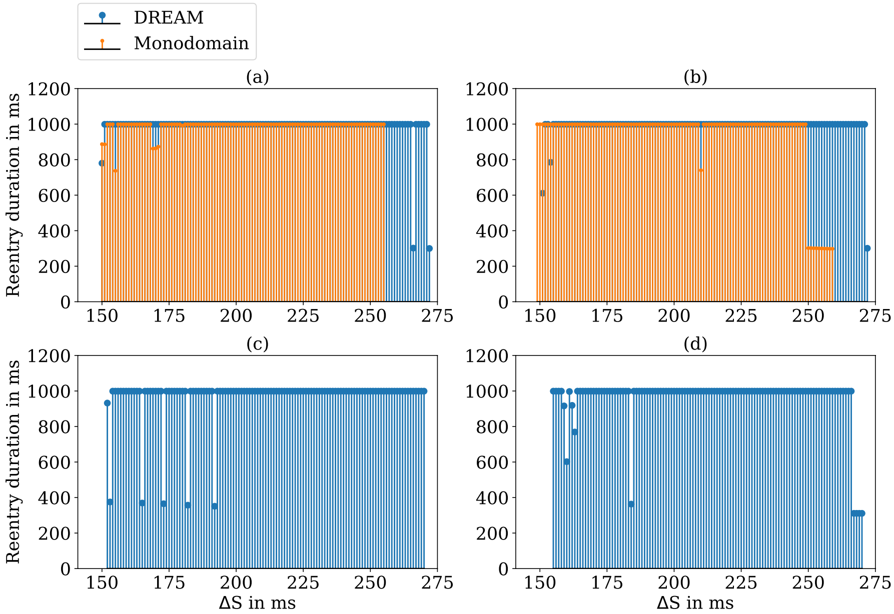


Fig. 5 Reentry duration at a conduction velocity of 200 mm/s for different mesh resolutions. Average edge length: **a** 200 µm, **b** 400 µm, **c** 800 µm, **d** 1600 µm. Monodomain simulations for coarser resolutions are not shown due to propagation failure in **c** and **d**. The horizontal axis shows ΔS (time difference between S2 and S1). The vertical axis shows reentry duration

DIs before propagating failure occurred. A more discernible difference arose in the CV resulting from each propagation model at this minimal DI. The DREAM demonstrated a higher minimum CV compared to the monodomain model (Fig. 4c). The ranges of both restitution curves obtained with DREAM and monodomain model for the Bueno-Orovio et al. ionic model resemble those for the Courtemanche et al. ionic model. The CV and ERP restitution curves (shown for the AF case in Fig. 4e, f) showed similar steepness. Similar to simulations with the DREAM and Courtemanche et al., the biphasic CV restitution obtained with the monodomain model and Bueno-Orovio et al. was not reproduced with the DREAM and Bueno-Orovio et al. The restitution curves for the ionic model of Mitchell and Schaeffer (shown for the AF remodeled case in Fig. 4g) and Fig. 4h) showed higher last propagating cycle lengths for the DREAM compared to the monodomain model. Pacing cycle length and DI for the last possible propagation were about 173 ms and 107 ms, respectively, longer for the DREAM compared to the monodomain models. Despite the absence of biphasic behavior in the monodomain CV restitution curve, the RMSE values were consistently higher across resolutions for the DREAM compared to Courtemanche et al. ionic model. Monodomain simulations resulted in smaller errors for both restitution curves under both conditions for resolutions without propagation failure. Moreover, DREAM simulations showed a tendency towards higher errors for ionic models with reduced levels of detail. Simulations across different resolutions for the DREAM showed similar RMSEs for each condition and ionic model. For instance, differences in RMSEs among resolutions in DREAM simulations were smaller than 5.46 mm/s and 0.5 ms for the CV and ERP restitution curves respectively per condition and ionic model.

4.2 Reentry in 2D sheets

Figure 5 shows the reentry duration for ΔS values within the vulnerable window for each of the resolutions at a CV of 200 mm/s. The DREAM produced similar results across all resolutions. In contrast, the monodomain results failed to propagate during pre-pacing for coarser resolutions (800 and 1600 µm average edge length). For higher resolutions (200 and 400 µm average edge length), both models showed reentries that lasted 1000 ms (i.e., until the simulation reached t_{end}) for most of the tested ΔS values. However, monodomain simulations produced a few reentries that stopped before 1000 ms for less than 10 ΔS values sparsely distributed across the vulnerable window. Moreover, vulnerable windows were shorter for the monodomain model mainly because it did not induce reentries for the longest ΔS values. For higher CVs, the monodomain model produced reentries that terminated before the end of the simulation, whereas simulations with the DREAM yielded a unidirectional block

Table 5 Mean ± standard deviation of the local reentry cycle length in ms at 4 points P_x across various mesh resolutions, indicated in average edge lengths in µm, for both the DREAM and monodomain models

Propagation model	Resolution	P_{left}	P_{down}	P_{right}	P_{up}
DREAM	200	154 ± 45	137 ± 11	151 ± 28	161 ± 45
	400	156 ± 46	138 ± 10	153 ± 28	163 ± 44
	800	158 ± 40	143 ± 9	154 ± 22	164 ± 38
	1600	157 ± 33	149 ± 10	154 ± 20	162 ± 32
Monodomain[a]	200	136 ± 28	157 ± 26	152 ± 28	146 ± 17
	400	162 ± 15	174 ± 24	174 ± 19	168 ± 24
	800	(–)	(–)	(–)	(–)
	1600	(–)	(–)	(–)	(–)

[a] Propagation failure at 800 and 1600 µm

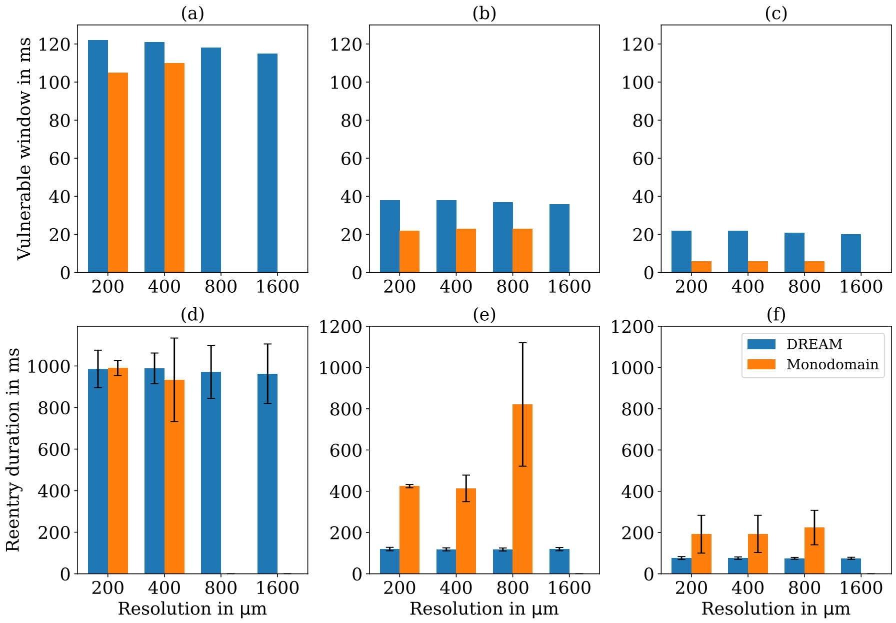


Fig. 6 **a** Vulnerable window at different mesh resolutions for a conduction velocity (CV) of 200 mm/s, **b** 600 mm/s and **c** 1000 mm/s. **d** Reentry duration (mean±std) for a CV of 200 mm/s, **e** 600 mm/s, **f** and 1000 mm/s

but did not reactivate the nodes where the S2 stimulus was applied.

Figure 6a–c show the vulnerable window duration for the 2 models at each of the 4 resolutions at different reference CVs. The vulnerable window duration was defined as the time between the earliest and the latest S2 time that induced a unidirectional block. Monodomain simulations on the mesh with an average edge length of 200 µm were taken as ground truth. The DREAM kept a stable error as mesh resolution decrease, performing equally even at the coarsest resolution of 1600 µm. Vulnerable windows were longer for all CVs and all the resolutions in the DREAM simulations. These differences in vulnerable window duration between DREAM simulations in all resolutions and monodomain simulation at finer resolutions did not increase in simulations with higher reference CV. Figure 6d–f show the reentry duration (mean ± std) for the same reference CVs. For the slowest CV both models produced longer reentries at most of ΔS values. In faster CV, both models resulted in shorter durations for all resolutions. The DREAM simulations resulted in reentry duration errors across all resolutions that increased with higher CVs. The discrepancy is grounded in the fact that the monodomain simulations produced self-terminated shorter reentries with a few turns within the 2D sheet at higher reference CV in this particular experiment. The DREAM simulations, on the other hand, resulted in propagation patterns with unidirectional blocks that were incapable of completing a full turn due to the lack of source-sink mismatch representation.

Table 5 presents the mean and standard deviation of local reentry cycle lengths observed in DREAM and monodomain simulations across various resolutions at nodes P_{left}, P_{down}, P_{right}, and P_{up}. In monodomain simulations with average edge lengths of 200 µm, mean local reentry cycle lengths ranged from 136 to 157 ms, with standard deviations varying from 17 to 28 ms. DREAM simulations consistently showed comparable mean and standard deviation values across different resolutions, ranging from 137 to 164 ms and from 9 to 46 ms, respectively. Notably, most of mean local reentry cycle lengths from DREAM simulations were closer to those observed in monodomain simulations at 200 µm than in simulations at 400 µm.

In the long-term stability analysis, the qualitative behavior of the reentry remained stable beyond the initial 1000 ms,

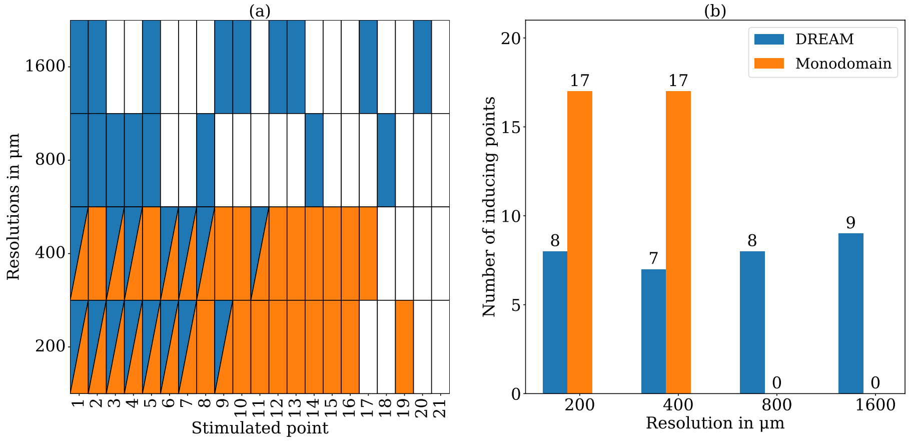


Fig. 7 Inducing points identified by the PEERP protocol. **a** Distribution of points that induced reentry in experiments with the DREAM, the monodomain model or both at different mesh resolutions (average edge length of 200, 400, 800, or 1600 µm). Each row represents an experiment, and each column represents one of the 21 stimulated points. The square's color indicates points that produced reentry with the DREAM, monodomain model, or both. **b** Number of inducing points per model and resolution (color figure online)

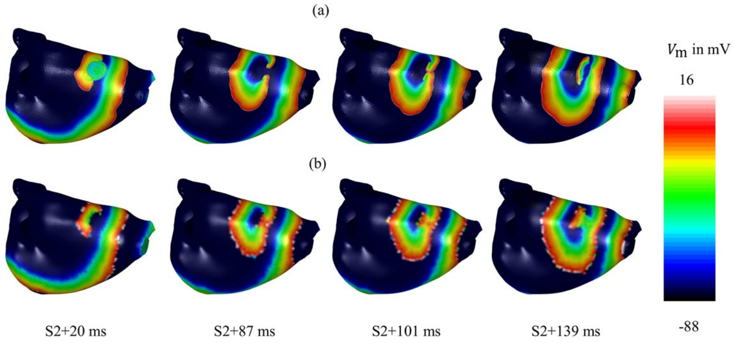


Fig. 8 Left atrium posterior view of transmembrane voltage maps of induced reentries at different time points after S2, stimulated at a point close to the roof. **a** Monodomain model with 200 µm average edge length, **b** DREAM with 1600 µm average edge length

Fig. 9 Left atrium posterolateral view of transmembrane voltage maps of induced reentries at different time points after S2, stimulated at a point near the mitral valve. **a** Monodomain model with 200 µm average edge length, **b** DREAM with 1600 µm average edge length

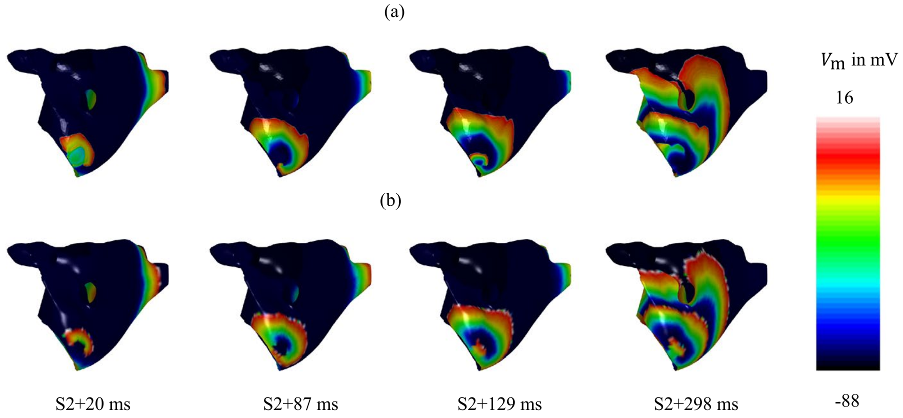


Table 6 Sensitivity and specificity for the DREAM and monodomain model at different mesh resolutions, expressed in average edge length in µm, considering monodomain experiments at 200 µm as the ground truth

Propagation model	Resolution	Sensitivity (%)	Specificity (%)
DREAM	200	47	100
	400	41	100
	800	41	75
	1600	41	50
Monodomain[a]	200	100	100
	400	94	75
	800	(–)	(–)
	1600	(–)	(–)

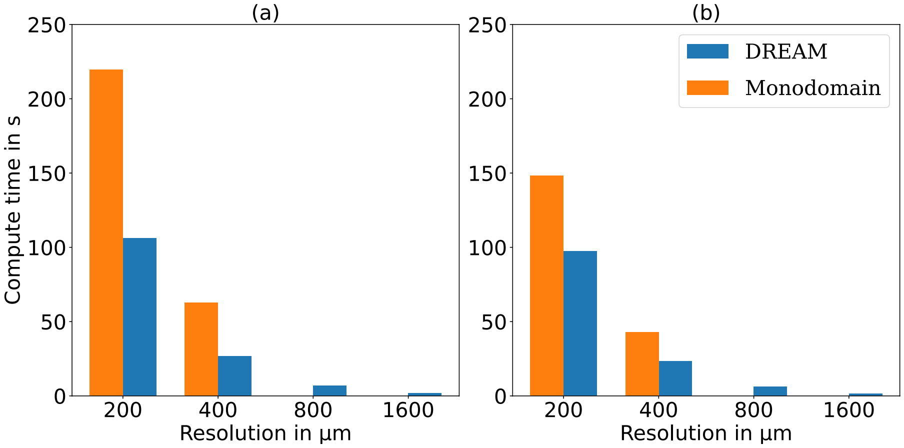
[a] Propagation failure at 800 and 1600 µm

continuing consistently until the simulation reached 60 s. However, after 13 s, small areas of the sheet began to show artifacts and activation discordance with the rest of the sheet, though this did not affect the overall behavior of the reentry. The mean local cycle length across nodes decreased from an initial value of 182 ms in the first cycle to 138 ms by the seventh cycle, remaining steady around this value until the 426th cycle, where it measured 137 ms. Similarly, the standard deviation of the local cycle length dropped from 27.1 to 0.8 ms between the first and seventh cycles, but then increased to 13 ms by the 426th cycle.

4.3 Reentry in the left atrium

Figure 7 shows the number of inducing points of the 21 stimulation points in the left atrium per experiment. Monodomain experiments at higher resolutions (average edge lengths 200 and 400 µm) showed the same number of inducing points. On the other hand, monodomain simulations failed to induce any reentry at lower resolutions (average edge lengths 800 and 1600 µm) due to propagation failure. DREAM experiments showed a lower number of inducing points compared to monodomain experiments. However, DREAM experiments showed a similar amount of inducing points across all resolutions. For some experiments, both models showed similar mechanisms of reentry at the same inducing point. For example, Fig. 8 shows simulations in the same inducing point in which both the DREAM and monodomain model produced a figure-of-eight reentry. Despite this example showing the same

Fig. 10 Computing time: **a** planar wavefront, **b** reentry



mechanism of reentry, the one induced by the DREAM lasted longer time. Figure 9 shows an example where the DREAM shows a figure-of-eight reentry while the monodomain model shows a spiral reentry. This difference arose from the lack of source-sink mismatch representation in the DREAM. In both cases, the reentries were sustained for the full 1000 ms simulated. Table 6 provides sensitivity and specificity of both models in the resolutions that did not have propagation failure running the PEERP protocol. Monodomain results at the highest resolution (i.e., average edge length of 200 μm) were used as ground truth. DREAM simulations exhibited better specificity than sensitivity, as most inducing points identified by the DREAM were also found by the monodomain model, whereas not all points found by the monodomain model were captured by the DREAM. On the other hand, DREAM performance was consistent across all resolutions, except for a significant drop in specificity for the coarsest resolution (i.e., average edge length of 1600 μm) to 50%.

4.4 Computing times

Figure 10 illustrates the computing times that the DREAM and monodomain model took for simulating a planar wavefront and reentry across different mesh resolutions. The proportion of computing time for the different steps of both models remained consistent across resolutions. Therefore, the detailed computing times for each step are mentioned only for simulations on a mesh with an average edge length of 200 μm. In both the planar wavefront and reentry experiment, for all resolutions, simulations with the DREAM were faster than those with the monodomain model. DREAM simulations on coarser resolution meshes (1600 μm) were 122 times faster in the planar wavefront experiment and approximately 87 times faster in the reentry experiment than monodomain simulations on finer mesh resolutions (200 μm).

In the DREAM simulations for the planar wavefront, the majority of the computing time was spent solving the ionic model equations (Fig. 1 step C), which took 83 s. The second most time-consuming step was calculating recovery times (Fig. 1 step D), which took 6 ms, followed by iterating the cycFIM (Fig. 1 step A), which took 5 s. Computing I_{diff} (Fig. 1 step B) required 2.5 s. In the same scenario, the monodomain model spent 80 s computing the ionic model and 138 s solving the parabolic equation.

When simulating the reentry with the DREAM, the majority of computing time was spent in step C, followed by steps A, D, and B, taking 51, 22, 16, and 2 s respectively (See Fig. 1). Given the significant role of diffusion currents in reentries, the monodomain simulations devoted most of their time to solving the parabolic equation (97 s), with only 51 s spent on computing the ionic current.

5 Discussion

5.1 Advantages and novel aspects of the DREAM

The DREAM benefits from the consistency of the eikonal model across mesh resolutions. Across all experiments, it was possible to obtain similar results with all tested resolutions. Unlike the monodomain model that requires fine mesh resolutions, hence more computational effort, the DREAM yields similar results at lower resolutions. This characteristic allows the DREAM to perform faster simulations.

Most of the eikonal models that have attempted to incorporate reentry phenomena use the FMM as a numerical method [5, 14] and implementations are not publicly available. Moreover, the regular FMM struggles with anisotropic propagation because this algorithm assumes the characteristic direction to be always colinear with the wavefront gradient, which is not true for anisotropic cases [15]. To address this limitation, Pernod et al. developed the anisotropic FMM, which incorporates an additional “CHANGED” list to enable recursive corrections [5]. Similarly, Cristiani proposed a similar method, introducing the “BUFFER” list with the same recursive correction concept [17]. In his study, he conducted various tests comparing the buffered FMM against the regular FMM, focusing on anisotropy problems. His findings revealed that the regular FMM calculated incorrect solutions with varying degrees of error across different anisotropic scenarios and resolutions. For instance, in an experiment with an anisotropy ratio of approximately 5 on a 2D grid of varying resolutions, the regular FMM was evaluated for its accuracy in solving the anisotropic eikonal equation, which in our case corresponds to finding ATs. The errors in the solution of the eikonal equation were 40 to 50 times larger compared to those of the buffered FMM, with the fast sweeping method, which is a classical iterative method, used as the reference solution [47].

Another constraint of the regular FMM is the inability to simulate reentry and reactivation. Pernod et al. also developed the multifrontal FMM, integrating a REFRACTORY list to include reentry [5]. However, this adaptation omitted the “CHANGED” list essential for managing anisotropic propagation, thereby compromising its efficacy in such scenarios. Although no reason for this omission is provided, it is possible that they faced similar challenges to those motivating this work. Specifically, changes in the solution of the eikonal equation make it difficult to determine definitive ATs for calculating RTs. Gassa et al. later incorporated ideas from the multifrontal FMM, the Dijkstra’s algorithm and the Mitchell and Schaeffer membrane model, reproducing spiral reentries in the atria [9, 12, 14, 35]. Notably, this adaptation neglected the use of recursive corrections or any other mechanism for better handling of anisotropy, possibly due to similar challenges encountered in Pernod et al.’s work.

The numerical solution of the anisotropic eikonal equation with regular FMM-based methods without recursive corrections does not lose stability under grid refinement [15]. The error from neglecting recursive corrections can be small, especially in experiments with small geometries, low anisotropy, and homogeneous properties [15, 17] and errors may be less apparent due to the stability of the solution [15]. However, arrhythmia simulations of multiple reentry cycles in realistic geometries with heterogeneous properties and high anisotropy may deviate significantly from the viscosity solution. For these complex cases, some of the adaptations made in the DREAM might be particularly useful.

The FIM is better suited to solve systems with anisotropic conditions [16]. However, the iteration process hinders the implementation of reactivation patterns and reentry. The DREAM uses the cycFIM framework to manage anisotropic propagation through correction mechanisms. The DREAM also addresses reentry by incorporating a “safety margin in time” τ_s, allowing for corrections before calculating RT based on a reliable AT. This approach could potentially also be applied for merging the anisotropic FMM and the multifrontal FMM, leveraging the strengths of both methods [5]. However, the computational cost of buffered FMM and the anisotropic FMM can exceed that of iterative methods in worst-case scenarios with high anisotropy. This is due to the increase in the “BUFFER” and “CHANGED” list sizes, respectively [5, 17]. Additionally, an FIM-based algorithm is more favorable because it can be parallelized more easily while maintaining computational density [16, 26, 27, 48]. The DREAM is the first model to use an iterative method such as cycFIM, ionic models and CV restitution to simulate functional reentry to the best of our knowledge.

Another method that uses the FIM in an alternative approach is the VITA method. This method employs the RE method based on the FIM approach [10, 16, 18]. Initially intended for studying ventricular tachycardia, VITA explores reentries using the FIM. While VITA relies on scar tissue with an isthmus to simulate anatomical reentry, the DREAM is capable of simulating reentry in the absence of structural abnormalities, including functional and anatomical reentries. Additionally, the cycFIM within the DREAM framework can simulate multiple ATs per node per reentry, considering CV restitution and ERP values to analyze reentry maintenance and average cycle length. Conversely, VITA does not compute multiple ATs per node during reentry. Instead, VITA calculates 2 types of ATs using the eikonal-based model. The first AT map identifies isochrone splitting and merging points, revealing isthmus exits where then elements are decoupled to induce unidirectional block for the second activation. This unidirectional block does not consider the electrophysiological properties at that location during repolarization. Subsequently, a second AT map is calculated for each isthmus exit, stimulating where the elements were decoupled. This second AT is the only one calculated during reentry. Then, the round-trip time (RTT) is defined as the time the wavefront takes to go around and reach the isthmus exit again. The RTT serves as a surrogate marker for possible ablation targets if its value is longer than 50 ms. One argument for using RTT as a surrogate instead of personalized CV restitution and ERP values is that the latter are rarely available in clinical settings. While most in silico studies do not personalize these variables but rather rely on literature references, some clinical and in silico studies have explored the effect of personalized CV restitution and ERP values on reentry patterns [49–52]. The DREAM in contrast can incorporate personalized CV restitution and ERP values extracted from patient measurements when available by adjusting parameters in the embedded ionic model and the COHERENCE() function.

While implementing τ_s allows to have reentries with multiple activation cycles in iterative methods, it also entails a new challenge successfully addressed by the DREAM. The cycFIM in step A_n of the n-th DREAM cycle and the RD portion in step C_{n-1} of the $(n-1)$-th DREAM cycle operate within different, non-overlapping time windows. Consequently, in scenarios such as reentry, it is common for the AT of a node to be calculated in step A_n without the previous RT for that node having been calculated by the RD model during step C_{n-1}, as required by the cycFIM. The DREAM overcomes this limitation in step D_{n-1} by briefly running the ionic model on the activated nodes until the voltage threshold is reached. This enables the ERP restitution to be directly obtained from the ionic model instead of relying on provided phenomenological curves [5, 7, 53].

This study also demonstrated how the DREAM could adapt to different ionic models by adjusting the corresponding parameters. This adaptability allows for the selection of the most suitable ionic model for a specific research question, depending on the required levels of computational speed and physiological detail. For instance, the Courtemanche et al. model is ideal when detailed biophysical information is needed, although it comes at the cost of computational speed [32]. In contrast, the simplified Bueno-Orovio et al. model, which includes the main ionic currents that modify the AP morphology and basic calcium dynamics, offers intermediate complexity with improved computational efficiency [34]. At the other end of the spectrum, the simplified model of Mitchell and Schaeffer combines fast inward and outward currents to represent basic AP dynamics, providing higher computational speed [35]. The DREAM was able to reproduce restitution behaviors very similar to the monodomain model when reducing the level of detail in the embedded ionic model for the sake of computational efficiency, as shown in the comparison between the Courtemanche et al. and Bueno-Orovio et al. ionic models. This demonstrates that the DREAM is flexible enough to be used in various research scenarios where computational resources are a constraint without significantly compromising on accuracy.

Another novel aspect of the DREAM lies in its approach to incorporate CV restitution. Previous methods typically involved incorporating the DI at every node from the previous activation cycle or the DI of neighboring nodes in the present activation cycle [7, 12]. In this study, CV is calculated using the DI of the present activation cycle, allowing for better adaptation to sudden changes in activation frequency and DI. Furthermore, as CV is assigned per node rather than per element, the minimization of AT is facilitated using analytical formulas such as Eq. 6. This eliminates the need for the Dijkstra's algorithm and additional pathways within the triangles can be considered [9]. In contrast, Corrado et al. proposed incorporating the DI within the Dijkstra's algorithm [12] which considers trajectories solely along the edges.

5.2 Limitations and future work

Several attempts have been made to incorporate additional properties such as curvature and diffusion effects into the pure eikonal model [54–57]. However, existing eikonal models still struggle to accurately represent complex activation patterns, such as multi-wavelet reentry, which are influenced by high wavefront curvatures, bath loading, wave collisions, and other source-sink mismatch effects. These properties are typical of propagation in fibrotic and heterogeneous tissue, a key factor in sustaining reentry [55]. To simulate these complex activation patterns accurately, it is necessary to accurately model diffusion and curvature effects, as well as reactivation and repolarization.

In this work, the approximation of the diffusion current was incorporated as a function that depends solely on the ATs with constant amplitude and duration. This function is unaffected by electrophysiological properties of the surrounding tissue. This limitation implies that the DREAM at this moment does not consider source-sink mismatch effects in the diffusion current and in the CV. Most of the differences observed between the DREAM and monodomain model in the experiments regarding reentries can be attributed to this phenomenon. In the RE$^+$ model described by Neic et al., I_{foot} (analogous to I_{diff} in the DREAM) is added to the parabolic portion of the RD model. On the other hand, the RE$^-$ model replaces the diffusion term $\nabla \cdot (\boldsymbol{\sigma}_{\mathrm{i}} \nabla V_{\mathrm{m}})$ with I_{foot}. By doing this, the RE$^+$ model managed to obtain similar repolarization phases as those obtained in the RD models when the tissue had heterogeneous APD. In the DREAM, I_{diff} is implemented as in the RE$^-$ model, resulting in sharp repolarization gradients that preserve the APD differences between neighboring cells. The primary reason for the sharp repolarization gradients is that I_{diff} does not account for the diffusion currents during the repolarization phase. Neglecting these diffusion currents during the repolarization phase can lead to the appearance of artifacts in localized regions during extended simulations. When implementing the RE$^+$ approach in the DREAM, one needs to ensure that the RD wavefront is never ahead of the eikonal wavefront to prevent major artifacts in the membrane models and their refractory behavior. This is more difficult for complex activation patterns with spatially heterogeneous curvature than for planar or radial excitation spread. For instance, during the repolarization phase, new activations (i.e., not predicted by the eikonal model) occurred when there was a strong gradient between cells with short APD and neighboring cells with longer APD. Since I_{diff} was calibrated using monodomain simulations with fixed conductivity values to obtain a CV of 1000 mm/s without considering restitution at shorter cycle lengths, this might result in small differences in AP amplitude.

In the present formulation of the DREAM, changes in sodium channel conductance are not explicitly linked to CV and I_{diff}. However, in the approach used to model AF in this manuscript, upstroke velocity changes are not critical for the dynamics under study [33]. While this approach may be less suitable for conditions where sodium channel dysfunction or block leads to a significant reduction in CV, we believe that CV can still be accurately fitted to patient-specific data or obtained from small-scale monodomain simulations for use in the DREAM model. In future work, further personalization of I_{diff} and CV to account for conductivity variations, I_{diff} restitution, and changes in ionic model conductances may enhance the model's accuracy.

While this paper is limited to triangles, the DREAM algorithm can be executed in tetrahedra by employing a similar approach as described in [16] and by implementing DREAM's adaptations to the tetrahedra equation within the local solver, as performed in Eq. 6. Nevertheless, further analysis of the computational efficiency is required for DREAM simulations in volumetric meshes. While the simulations in this paper were conducted in serial code, parallelization can further enhance the efficiency of the DREAM.

Another promising improvement to decrease the computing time of the DREAM would be to increase the integration time step when calculating the RT in cases where they were not yet provided by the ionic model. Currently, RTs are calculated twice at some nodes. In the first calculation, the membrane potential V_{m} is determined asynchronously for each cell, but only the RT is saved and passed to the eikonal model, without storing V_{m}. In the second calculation, the membrane potential V_{m} is computed in a synchronized manner across all nodes (i.e., each time step is calculated for all nodes simultaneously), and the V_{m} values are stored. The need for synchronizing the nodes for storing purposes is the reason why the state variables and V_m states are reset between the first and second calculation. Since diffusion currents during the repolarization phase are neglected in the latest version of the DREAM, this redundancy does not introduce discrepancies, as the same calculations are being repeated. However, future improvements could focus on optimizing the process to avoid redundant RTs calculations. Additionally, new considerations will be needed when diffusion currents during the repolarization phase are incorporated.

In this work, atrial cell models and CV values were tested in the context of AF. Nonetheless, other atrial or ventricular models can also be used with the DREAM as long as the parameters of CV restitution and AP properties are tuned accordingly. Investigating the tissue effects of additional changes in ionic models, such as pharmacological effects or channel mutations, represents another valuable area for exploration. On the other hand, the DREAM with the embedded ionic model of Mitchell and Schaeffer demonstrated challenges in reproducing the restitution behavior at short pacing cycle lengths observed in the corresponding monodomain model. Therefore, a better adjustment of the DREAM parameters is required to further improve the faithful representation of CV restitution curve and simulation of reentrant scenarios.

More systematic analysis of the new parameters introduced in the DREAM is required to further understand the optimal tuning that allows for a good balance between accuracy and computational efficiency. We expect that τ_s should increase when the anisotropy ratio increases as more changes are required. On the other hand, τ_inc must be smaller than the sum of the longest possible APD and the minimum DI, i.e., parameter θ in Eq. 12. Moreover, further investigation of the COHERENCE() function could enhance the approximation of CV restitution, particularly near propagation failure.

The cycFIM embedded in the DREAM could potentially also be used as an alternative method to simulate other applications of RD models. In this case, changing the ionic model according to the reaction part of the probable application would be necessary. Potential use cases include the RD model simulating cyclical phenomena like the Belousov–Zhabotinsky reaction, which can also exhibit spiral propagation patterns [58].

5.3 Conclusion

The DREAM presents several advancements in simulating cardiac arrhythmias compared to existing eikonal-based models while retaining their main advantages and making it accessible as part of the openCARP simulator [30]. By inheriting consistency across mesh resolutions from the eikonal model, the DREAM achieves faster computing times compared to RD models for a given desired accuracy. Additionally, the DREAM faithfully represents functional reentry without the need for structural abnormalities like scar tissue. The use of cycFIM enables multiple ATs per node, allowing for better analysis of reentry patterns in anisotropic media. Moreover, the DREAM permits modification of CV restitution and ERPs, enhancing the personalization of cardiac computer models. Ongoing work aims to overcome limitations in representing source-sink balance. Overall, the DREAM offers promising prospects for advancing our understanding and improving treatments of cardiac arrhythmias compatible with clinical time frames. Finally, the cycFIM may find applications beyond cardiac modeling in simulating cyclical phenomena.

Acknowledgements This work was supported from the European Union's Horizon research and Innovation programme under the Marie Skłodowska-Curie grant agreement No. 860974, by the Leibniz ScienceCampus "Digital Transformation of Research" with funds from the programme "Strategic Networking in the Leibniz Association", the European High-Performance Computing Joint Undertaking EuroHPC under grant agreement No 955495 (MICROCARD) co-funded by the Horizon 2020 programme of the European Union (EU) and the German Federal Ministry of Education and Research (BMBF), by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Project-ID 258734477—SFB 1173 and LO 2093/6-1 (SPP 2311) and—Project-ID 394433254 (LU 2294/1-1, DO 637/23-1, WA 4259/1-1), and by the MCIN\AEI and the European Union NextGenerationEU\RTR under grant PLEC2021-007614.

Author Contributions Conceptualisation: CBE, AL. Data curation: CBE, PMD. Formal Analysis: CBE, SA, SB. Funding Acquisition: AL. Investigation: CBE, SA, LU, SB. Methodology: CBE, JS, SA, AL. Project administration: CBE, AL. Software: CBE, JS, MH, SB, LU, SA, PMD, AL. Supervision: AL. Validation: JS, SB, SA. Visualization: CBE, SA, JK. Writing—original draft: CBE, SA. Writing—review and editing: CBE, JS, MH, LU, PMD, JK, SB, AL, SA.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability No original data was used for this work

Code availability The repository with all the original code is publicly available in the openCARP repository

Declarations

Conflict of interest The authors declare no Conflict of interest

References

1. Grandi E, Dobrev D, Heijman J (2019) Computational modeling: what does it tell us about atrial fibrillation therapy? Int J Cardiol 287:155–161. https://doi.org/10.1016/j.ijcard.2019.01.077
2. Trayanova NA, Lyon A, Shade J, Heijman J (2023) Computational modeling of cardiac electrophysiology and arrhythmogenesis. Physiol Rev. https://doi.org/10.1152/physrev.00017.2023
3. Jacquemet V (2016) Lessons from computer simulations of ablation of atrial fibrillation. J Physiol 594(9):2417–2430. https://doi.org/10.1113/JP271660
4. Nagel C, Barrios Espinosa C, Gillette K, Gsell MAF, Sanchez J, Plank G, Dossel O, Loewe A (2023) Comparison of propagation models and forward calculation methods on cellular, tissue and organ scale atrial electrophysiology. IEEE Trans Biomed Eng 70(2):511–522. https://doi.org/10.1109/tbme.2022.3196144
5. Pernod E, Sermesant M, Konukoglu E, Relan J, Delingette H, Ayache N (2011) A multi-front eikonal model of cardiac electrophysiology for interactive simulation of radio-frequency ablation. Comput Gr 35(2):431–440. https://doi.org/10.1016/j.cag.2011.01.008
6. Bishop MJ, Plank G (2011) Bidomain ecg simulations using an augmented monodomain model for the cardiac source. IEEE Trans Biomed Eng 58(8):2297–2307. https://doi.org/10.1109/TBME.2011.2148718
7. Loewe A, Poremba E, Oesterlein T, Luik A, Schmitt C, Seemann G, Dössel O (2019) Patient-specific identification of atrial flutter vulnerability-a computational approach to reveal latent reentry pathways. Front Physiol. https://doi.org/10.3389/fphys.2018.01910
8. Jacquemet V (2012) An eikonal-diffusion solver and its application to the interpolation and the simulation of reentrant cardiac activations. Comput Methods Programs Biomed 108(2):548–558. https://doi.org/10.1016/j.cmpb.2011.05.003
9. Wallman M, Smith NP, Rodriguez B (2012) A comparative study of graph-based, eikonal, and monodomain simulations for the estimation of cardiac activation times. IEEE Trans Biomed Eng 59(6):1739–1748. https://doi.org/10.1109/TBME.2012.2193398
10. Neic A, Campos FO, Prassl AJ, Niederer SA, Bishop MJ, Vigmond EJ, Plank G (2017) Efficient computation of electrograms and ecgs in human whole heart simulations using a reaction-eikonal model. J Comput Phys 346:191–211. https://doi.org/10.1016/j.jcp.2017.06.020
11. Niederer SA, Lumens J, Trayanova NA (2019) Computational models in cardiology. Nat Rev Cardiol 16(2):100–111. https://doi.org/10.1038/s41569-018-0104-y
12. Corrado C, Zemzemi N (2018) A conduction velocity adapted eikonal model for electrophysiology problems with re-excitability evaluation. Med Image Anal 43:186–197. https://doi.org/10.1016/j.media.2017.11.002
13. Pullan AJ, Tomlinson KA, Hunter PJ (2002) A finite element method for an eikonal equation model of myocardial excitation wavefront propagation. SIAM J Appl Math 63(1):324–350. https://doi.org/10.1137/S0036139901389513
14. Gassa N, Zemzemi N, Corrado C, Coudière Y (2021) Spiral waves generation using an eikonal-reaction cardiac electrophysiology model. In: International conference on functional imaging and modeling of the heart. Springer, pp 523–530. https://doi.org/10.1007/978-3-030-78710-3_50
15. Sethian JA, Vladimirsky A (2003) Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms. SIAM J Numer Anal 41(1):325–363. https://doi.org/10.1137/S0036142901392742
16. Fu Z, Kirby RM, Whitaker RT (2013) A fast iterative method for solving the eikonal equation on tetrahedral domains. SIAM J Sci Comput 35(5):473–494. https://doi.org/10.1137/120881956
17. Cristiani E (2009) A fast marching method for Hamilton–Jacobi equations modeling monotone front propagations. J Sci Comput 39(2):189–205. https://doi.org/10.1007/s10915-008-9257-x
18. Campos FO, Neic A, Costa CM, Whitaker J, O'Neill M, Razavi R, Rinaldi CA, Niederer SA, Plank G, Bishop MJ (2022) An automated near-real time computational method for induction and treatment of scar-related ventricular tachycardias. Med Image Anal 80:102483. https://doi.org/10.1016/j.media.2022.102483
19. Espinosa CB, Sánchez J, Dössel O, Loewe A (2022) Diffusion reaction eikonal alternant model: towards fast simulations of complex cardiac arrhythmias. Comput Cardiol 498:1–4. https://doi.org/10.22489/CinC.2022.054. (**IEEE**)
20. Tung L (1978) A bi-domain model for describing ischemic myocardial dc potentials. PhD thesis, Massachusetts Institute of Technology
21. Franzone PC, Pavarino LF, Scacchi S (2014) Mathematical cardiac electrophysiology. Springer, Berlin
22. Keener J, Sneyd J (2009) Mathematical physiology 1: cellular physiology. Springer, Berlin
23. Vigmond E, Dos Santos RW, Prassl A, Deo M, Plank G (2008) Solvers for the cardiac bidomain equations. Prog Biophys Mol Biol 96(1–3):3–18. https://doi.org/10.1016/j.pbiomolbio.2007.07.012
24. Potse M, Dubé B, Vinet A, Cardinal R (2006) A comparison of monodomain and bidomain propagation models for the human heart. In: 2006 International conference of the IEEE engineering in medicine and biology society, pp. 3895–3898. https://doi.org/10.1109/IEMBS.2006.259484. IEEE
25. Kandel SM (2015) The electrical bidomain model: a review. Sch Acad J Biosci 3:633–639
26. Fu Z, Jeong W-K, Pan Y, Kirby RM, Whitaker RT (2011) A fast iterative method for solving the eikonal equation on triangulated surfaces. SIAM J Sci Comput 33(5):2468–2488. https://doi.org/10.1137/100788951
27. Jeong W-K, Whitaker RT (2008) A fast iterative method for eikonal equations. SIAM J Sci Comput 30(5):2512–2534. https://doi.org/10.1016/j.procs.2014.05.170
28. openCARP Eikonal. GitLab. https://git.opencarp.org/openCARP/openCARP/-/tree/openCARP_DREAM
29. Azzolin L, Schuler S, Dössel O, Loewe A (2021) A reproducible protocol to assess arrhythmia vulnerability in silico: pacing at the end of the effective refractory period. Front Physiol 12:656411. https://doi.org/10.3389/fphys.2021.656411
30. Plank G, Loewe A, Neic A, Augustin C, Huang Y-LC, Gsell M, Karabelas E, Nothstein M, Sánchez J, Prassl A, Seemann G, Vigmond E (2021) The openCARP simulation environment for cardiac electrophysiology. Comput Methods Programs Biomed 208:106223. https://doi.org/10.1016/j.cmpb.2021.106223
31. Costa CM, Hoetzl E, Rocha BM, Prassl AJ, Plank G (2013) Automatic parameterization strategy for cardiac electrophysiology simulations. IEEE, pp 373–376
32. Courtemanche M, Ramirez RJ, Nattel S (1998) Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. Am J Physiol-Heart Circul Physiol 275(1):301–321. https://doi.org/10.1152/ajpheart.1998.275.1.H301
33. Loewe A (2016) Modeling human atrial patho-electrophysiology from ion channels to ECG-substrates, pharmacology, vulnerability, and P-waves, vol 23. KIT Scientific Publishing, Karlsruhe
34. Bueno-Orovio A, Cherry EM, Fenton FH (2008) Minimal model for human ventricular action potentials in tissue. J Theor Biol 253(3):544–560. https://doi.org/10.1016/j.jtbi.2008.03.029

35. Mitchell CC, Schaeffer DG (2003) A two-current model for the dynamics of cardiac membrane. Bull Math Biol 65(5):767–793. https://doi.org/10.1016/S0092-8240(03)00041-7
36. Lenk C, Weber FM, Bauer M, Einax M, Maass P, Seeman G (2015) Initiation of atrial fibrillation by interaction of pacemakers with geometrical constraints. J Theor Biol 366:13–23. https://doi.org/10.1016/j.jtbi.2014.10.030
37. Corrado C, Whitaker J, Chubb H, Williams S, Wright M, Gill J, O'Neill MD, Niederer SA (2016) Personalized models of human atrial electrophysiology derived from endocardial electrograms. IEEE Trans Biomed Eng 64(4):735–742. https://doi.org/10.1109/TBME.2016.2574619
38. Geuzaine C, Remacle J-F (2009) Gmsh: a 3-d finite element mesh generator with built-in pre-and post-processing facilities. Int J Numer Meth Eng 79(11):1309–1331
39. Ahrens J, Geveci B, Law C ParaView: an end-user tool for large data visualization
40. Schmidt R, Singh K (2010) meshmixer: an interface for rapid mesh composition. In: ACM SIGGRAPH 2010 Talks. SIGGRAPH '10. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1837026.1837034
41. Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F, Ranzuglia G MeshLab: an Open-Source Mesh Processing Tool. https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136
42. Hunter JD (2007) Matplotlib: a 2d graphics environment. Comput Sci Eng 9(3):90–95. https://doi.org/10.1109/MCSE.2007.55
43. ...Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat İ, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P (2020) SciPy 1.0 contributors: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods 17:261–272
44. Azzolin L, Eichenlaub M, Nagel C, Nairn D, Sánchez J, Unger L, Doessel O, Jadidi A, Loewe A 29 atrial models created with a patient-specific augmented atrial model generation tool (AugmentA). https://doi.org/10.5281/zenodo.5589289
45. Nagel C, Schuler S, Dössel O, Loewe A (2021) A bi-atrial statistical shape model for large-scale in silico studies of human atria: model development and application to ecg simulations. Med Image Anal 74:102210. https://doi.org/10.1016/j.media.2021.102210
46. Azzolin L, Eichenlaub M, Nagel C, Nairn D, Sánchez J, Unger L, Arentz T, Westermann D, Dössel O, Jadidi A (2023) Augmenta: patient-specific augmented atrial model generation tool. Comput Med Imaging Graph 108:102265. https://doi.org/10.1016/j.compmedimag.2023.102265
47. Qian J, Zhang Y-T, Zhao H-K (2007) A fast sweeping method for static convex Hamilton–Jacobi equations. J Sci Comput 31(1–2):237–271. https://doi.org/10.1007/s10915-006-9124-6
48. Capozzoli A, Curcio C, Liseno A, Savarese S (2013) A comparison of fast marching, fast sweeping and fast iterative methods for the solution of the eikonal equation. In: 2013 21st Telecommunications Forum Telfor (TELFOR), pp 685–688. https://doi.org/10.1109/TELFOR.2013.6716321 . IEEE
49. Unger LA, Azzolin L, Nothstein M, Sanchez J, Luik A, Seemann G, Yeshwant S, Oesterlein T, Doessel O, Schmitt C (2021) Cycle length statistics during human atrial fibrillation reveal refractory properties of the underlying substrate: a combined in silico and clinical test of concept study. EP Europace 23(Supplement-1):133–142. https://doi.org/10.1093/europace/euaa404
50. Deng D, Murphy MJ, Hakim JB, Franceschi WH, Zahid S, Pashakhanloo F, Trayanova NA, Boyle PM (2017) Sensitivity of reentrant driver localization to electrophysiological parameter variability in image-based computational models of persistent atrial fibrillation sustained by a fibrotic substrate. Chaos Interdiscip J Nonlinear Sci. doi 10(1063/1):5003340
51. Corrado C, Williams S, Karim R, Plank G, O'Neill M, Niederer S (2018) A work flow to build and validate patient specific left atrium electrophysiology models from catheter measurements. Med Image Anal 47:153–163. https://doi.org/10.1016/j.media.2018.04.005
52. Martínez Díaz P, Dasí A, Goetz C, Unger L, Haas A, Luik A, Rodríguez B, Dössel O, Loewe A (2024) Impact of effective refractory period personalization on arrhythmia vulnerability in patient-specific atrial computer models. medRxiv. https://doi.org/10.1101/2024.06.06.24308556
53. Serra D, Romero P, Garcia-Fernandez I, Lozano M, Liberos A, Rodrigo M, Bueno-Orovio A, Berruezo A, Sebastian R (2022) An automata-based cardiac electrophysiology simulator to assess arrhythmia inducibility. Mathematics 10(8):1293. https://doi.org/10.3390/math10081293
54. Sermesant M, Coudière Y, Moreau-Villéger V, Rhode KS, Hill DL, Razavi R (2005) A fast-marching approach to cardiac electrophysiology simulation for xmr interventional imaging. In: Medical image computing and computer-assisted intervention–MICCAI 2005: 8th international conference, Palm Springs, CA, USA, October 26–29, 2005, Proceedings, Part II 8. Springer, pp 607–615. https://doi.org/10.1007/11566489_75
55. Gander L, Krause R, Weiser M, Sahli Costabal F, Pezzuto S (2023) On the accuracy of eikonal approximations in cardiac electrophysiology in the presence of fibrosis. In: International conference on functional imaging and modeling of the heart. Springer, pp 137–146
56. Colli Franzone P, Guerri L, Rovida S (1990) Wavefront propagation in an activation model of the anisotropic cardiac tissue: asymptotic analysis and numerical simulations. J Math Biol 28:121–176. https://doi.org/10.1007/BF00163143
57. Keener JP (1991) An eikonal-curvature equation for action potential propagation in myocardium. J Math Biol 29(7):629–651. https://doi.org/10.1007/BF00163916
58. Keener JP, Tyson JJ (1986) Spiral waves in the Belousov–Zhabotinskii reaction. Phys D 21(2–3):307–324. https://doi.org/10.1016/0167-2789(86)90007-2