Order Investigation of Scalable Memory-Efficient Finite Volume Methods for Parabolic Advection-Diffusion-Reaction Equations with Point Sources

Xuan Huang,[1] Matthias K. Gobbert,[1] Bradford E. Peercy,[1] Stefan Kopecz,[2] Philipp Birken,[3] and Andreas Meister[2]

We investigate the order of convergence for a finite volume discretization in solving unsteady three-dimensional advection-diffusion-reaction equations. Our implementation demonstrates the effective solution of the calcium induced calcium release problem for long-time simulation in the presence of strong advection. We give numerical evidence of convergence for test problems with smooth as well as non-smooth source terms in two-dimensional and three-dimensional settings. A parallel implementation using MPI allows for efficient computations with a long simulation time. The ability to solve larger problems on finer mesh with stable efficiency is demonstrated in a weak scalability study.

KEY WORDS: Finite volume method; Dirac delta distribution; Matrix-free Newton-Krylov method; Calcium waves; Parallel computing

1 INTRODUCTION

Advection-diffusion-reaction equations occur in a wide variety of applications, for instance fluid flow, heat transfer, spread of pollutants, and transport-chemistry problems. We consider a very general framework of this problem as a testbed to investigate the choices of numerical methods in the face of Dirac delta distributions among the source terms. We also demonstrate the ability of memory-efficient parallel implementations of these methods to solve the problem on extremely fine meshes efficiently using a cluster with state-of-the-art CPU nodes and cutting-edge hybrid CPU/GPU nodes. Our consideration of this problem is inspired by the need to simulate Calcium Induced Calcium Release (CICR) in a heart cell [3, 4, 10]. CICR describes a physiological process within a cell where calcium is able to activate calcium release from the sarcoplasmic reticulum into the cytosol, which is crucial for excitation-contraction coupling in the cardiac muscle.

As a testbed, we consider the system of coupled, non-linear, time-dependent advection-diffusion-reaction equations

$$u_t^{(i)} - \nabla \cdot \left(D^{(i)} \nabla u^{(i)}\right) + \beta^{(i)} \cdot \left(\nabla u^{(i)}\right) = q^{(i)}, \quad i = 1, \ldots, n_s, \tag{1.1}$$

with functions $u^{(i)} = u^{(i)}(\mathbf{x}, t)$, $i = 1, \ldots, n_s$, of space $\mathbf{x} \in \Omega \subset \mathbb{R}^3$ and time $0 \le t \le t_{\text{fin}}$ representing the concentrations of the n_s species. The diffusivity matrices $D^{(i)} = \text{diag}\,(D_{11}^{(i)}, D_{22}^{(i)}, D_{33}^{(i)}) \in \mathbb{R}^{3\times 3}$ consists of positive entries and are assumed to dominate the advection velocity vectors $\beta^{(i)} \in \mathbb{R}^3$, so that numerical methods for parabolic problems are always justified. The right-hand side $q^{(i)}$ is written in a way that distinguishes the different dependencies and effects as

$$q^{(i)}(u^{(i)}, \ldots, u^{(n_s)}, \mathbf{x}, t) = s^{(i)}(u^{(i)}, \mathbf{x}, t) + r^{(i)}(u^{(1)}, \ldots, u^{(n_s)}) + f^{(i)}(\mathbf{x}, t). \tag{1.2}$$

[1]Department of Mathematics and Statistics, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, U.S.A. E-mail: gobbert@umbc.edu

[2]Institute of Mathematics, University of Kassel, Heinrich-Plett-Str. 40, 34109 Kassel, Germany

[3]Centre for Mathematical Sciences, Lund University, Box 118, 221 00 Lund, Sweden

The terms $s^{(i)}(u^{(i)}, \mathbf{x}, t)$ in (1.2) contain many thousands of point sources modeled by Dirac delta distributions on a large lattice throughout the cell. This crucial feature of the model is responsible for many of the challenges: (i) The numerical method used as spatial discretization will not be convergent to as high an order as conventional, since the source functions are not sufficiently smooth. (ii) The point sources are the crucial driver of the physiological effects. The domain of the cell needs to be discretized with a very fine mesh to accomodate the large number of point sources. (iii) For application problems involving chemical reactions, the system in (1.2) conserves mass, and thus, the numerical method should have this property, as well.

In (1.2), the reaction terms $r^{(i)} = r^{(i)}(u^{(1)}, \ldots, u^{(n_s)})$ are, in general, non-linear autonomous functions of all species and couple the reaction equations in the system (1.1). In (1.2), the term $q^{(i)}$ also includes the function $f^{(i)} = f^{(i)}(\mathbf{x}, t)$, so that the scalar linear test problem $u_t - \nabla \cdot (D \nabla u) + \beta \cdot \nabla u = f(x, t)$ is incorporated in the formulation. This combined formulation of the problems allows one to switch the code from one problem to the other by turning on and off terms and is useful in testing correctness of the code and convergence of the numerical methods in the testbed. These terms will be discussed in detail in Section 2.

Gobbert [3] (see also [4] for details) gives a matrix-free Newton-Krylov method for the simulation of calcium flow in a heart cell. The underlying model of calcium flow is given by a system of three coupled diffusion-reaction equations, in which the occurring source terms can be divided into linear and non-linear parts, as well as point sources. Due to the shape of a heart cell, a rectangular domain with a structured grid is a natural choice. The method is based on a finite element discretization and implemented in a matrix-free manner. The convergence of the finite element method in the presence of the measure valued source terms which occur in the calcium model was rigorously shown in [11].

In [10], a finite volume method for advection-diffusion-reaction systems with smooth and non-smooth sources was introduced. The finite volume method is designed for transport problems, since it satisfies mass conservation in the discretized equations. Moreover, since we intend to consider a more general class of equation systems of advection-diffusion-reaction (ADR) type, the finite volume method becomes a natural choice as opposed to finite element methods which would need additional stabilization terms [8]. However, it is not clear whether a higher order discretization of the advection term is necessary at the cost of more MPI communication among processes. This paper extends [10] by fully analyzing the finite volume method for advection-diffusion-reaction systems:

(i) We capture the effect of advection in three dimensional long-time simulation of the CICR model. This is achieved by having a non-negative value before the z-direction advection term. Different advection speeds of the calcium waves are observed, which in turn shows our numerical methods are producing expected results.

(ii) In the absence of a rigorous convergence theory for problems involving non-smooth sources, we demonstrate convergence of the method numerically and compare the results to those obtained by the finite element method from [3]. We show convergence for scalar test problems with choices between smooth versus non-smooth source terms and first-order versus second-order discretization for advection term, in both two and three dimensional mesh spacing. Results are compared to simulations using the finite element method in situations where there is no advection.

(iii) Our existing CICR model needs to be extended to incorporate one or two additional species and needs to be solved up to larger final time to be better comparable to laboratory experiments that might take a final time of one minute or more. Thus, our numerical methods and their parallel implementation need to demonstrate the potential to scale up. A conventional performance study in [10] demonstrates good strong scalibility by generally halving computing time when using twice as many resources. However, the fixed problem size in that study does not reflect the motivation to

solve larger problems. We introduce here weak scalability studies for a scalar test problem as well as CICR simulation with advection. Our code demonstrates the ability to handle larger problems with the same efficiency.

The outline of this paper is as follows. Section 2 gives a short description of the equations modeling the calcium induced calcium release in a heart cell as well as the scalar test cases that will be used to demonstrate the convergence of numerical method, as well as weak scalability. In Section 3 we explain the numerical method. In particular the finite volume descritization. Section 4 illustrates the applicability of the method to long time simulations of calcium induced calcium release in a heart cell, and shows the effect of advection. In Section 5 various convergence studies are presented. Finally, the weak scalability of the scheme in parallel computations will be presented in Section 6 along with comments on the parallel implementation of the method. Section **??** summarizes with conclusion.

2 GOVERNING EQUATIONS

In this section we describe the governing equations, first for the physiological application problem and then for the two scalar test problems with smooth and non-smooth source terms, respectively.

2.1 Calcium Induced Calcium Release with Advection

Calcium Induced Calcium Release (CICR) describes a physiological process where calcium is able to activate calcium release into the cytosol, which is crucial for excitation-contraction coupling in the cardiac muscle. This CICR model described in (1.1)–(1.2) was originally introduced in [5, 7], extended in [6], and its numerics discussed in [3, 4, 10, 11]. A complete list of the parameter values of the CICR model is given in Table 1. The problem can be modeled by the system of time-dependent advection-reaction-diffusion equations (1.1)–(1.2) coupled by non-linear reaction terms. We consider the rectangular spatial domain

$$\Omega = (-6.4\ \mu\text{m}, 6.4\ \mu\text{m}) \times (-6.4\ \mu\text{m}, 6.4\ \mu\text{m}) \times (-32.0\ \mu\text{m}, 32.0\ \mu\text{m}) \subset \mathbb{R}^3$$

that captures the essential size and elongated shape of a heart cell. This model consists of $n_s = 3$ equations corresponding to calcium ($i = 1$), an endogenous calcium buffer ($i = 2$), and a fluorescent indicator dye ($i = 3$).

We describe the three terms on the right-hand side (1.2) as follows:

(i) In the present model, the term $s^{(i)}(u^{(i)}, \mathbf{x}, t)$ applies only to the calcium species $i = 1$, which is implemented using the Kronecker delta function δ_{i1} in the definition

$$s^{(i)}(u^{(i)}, \mathbf{x}, t) = \left(J_{\text{SR}}(u^{(1)}, \mathbf{x}, t) - J_{\text{pump}}(u^{(1)}) + J_{\text{leak}}\right) \delta_{i1}, \quad i = 1, \ldots, n_s. \tag{2.1}$$

The key term J_{SR} houses the stochastic aspect of the model, since it models how calcium is released from the sarcoplasmic reticulum (SR) into the cytosol. The calcium release units (CRUs) which represent release sites on the sarcoplasmic reticulum are arranged discretely on a three-dimensional lattice. Each CRU has a probability of opening depending on the concentration of calcium present at that site. This process takes the form

$$J_{\text{SR}}(u^{(1)}, \mathbf{x}, t) = \sum_{\hat{\mathbf{x}} \in \Omega_s} g\, S_{\hat{\mathbf{x}}}(u^{(1)}, t)\, \delta(\mathbf{x} - \hat{\mathbf{x}}). \tag{2.2}$$

The equation models the superposition of calcium injection into the cell at CRUs, which are modeled as point sources at all $\hat{\mathbf{x}}$ in the set of CRU locations Ω_s. The Dirac delta distribution $\delta(\mathbf{x} - \hat{\mathbf{x}})$ together with the constant flux density g models a point source at a CRU located at $\hat{\mathbf{x}} \in \Omega_s$. $S_{\hat{\mathbf{x}}}$ is an indicator function, its value is either 1 or 0 indicating whether the CRU at $\hat{\mathbf{x}}$ is open or closed. The value of $S_{\hat{\mathbf{x}}}$ is determined by comparing a uniform random number to the value of the probability distribution

$$J_{\text{prob}}(u^{(1)}) = \frac{P_{\max}\,(u^{(1)})^{n_{\text{prob}}}}{(K_{\text{prob}})^{n_{\text{prob}}} + (u^{(1)})^{n_{\text{prob}}}}. \tag{2.3}$$

If the value of the probability distribution is higher than the random number, then the CRU switches on by setting $S_{\hat{\mathbf{x}}} = 1$, otherwise it remains closed by $S_{\hat{\mathbf{x}}} = 0$. When the CRU is open, it stays open for 5 ms, then it remains closed for 100 ms. In our physiological simulations, we study the self-initiation of calcium waves without stimulation, therefore, the comparison of J_{prob} to the uniform random number is the only mechanism available in the model to start a calcium wave.

The term $s^{(1)}(u^{(1)}, \mathbf{x}, t)$ also houses the non-linear pump term

$$J_{\text{pump}}(u^{(1)}) = \frac{V_{pump}(u^{(1)})^{n_{pump}}}{(K_{pump})^{n_{pump}} + (u^{(1)})^{n_{pump}}} \tag{2.4}$$

and the constant balance term J_{leak}. By design, these terms balance out as $J_{\text{leak}} = J_{\text{pump}}(0.1) \equiv$ constant (see Table 1) for the calcium concentration at basal level 0.1 μM.

(ii) The reaction terms in (1.2) are

$$r^{(i)}(u^{(1)}, \ldots, u^{(n_s)}) := \begin{cases} \sum_{j=2}^{n_s} R^{(j)}(u^{(1)}, u^{(j)}), & \text{for } i = 1, \\ R^{(i)}(u^{(1)}, u^{(i)}), & \text{for } i = 2, \ldots, n_s, \end{cases} \tag{2.5}$$

where the reaction rates are given by

$$R^{(i)} = -k_i^+ u^{(1)} u^{(i)} + k_i^- \left(\overline{u}_i - u^{(i)} \right), \quad \text{for } i = 2, \ldots, n_s, \tag{2.6}$$

are modeled as autonomous non-linear functions of the different species and couple the equations in the general system (1.1)–(1.2). Since the only sources/drains for the species $i = 2, 3$ are the terms that model the binding/unbinding reactions with calcium, the no-flow boundary conditions assure that the total concentration of these species bound and not bound with calcium remains constant, and this constant is denoted by $\bar{u}_i$, $i = 2, 3$; see Table 1.

(iii) The term $f^{(i)} = f^{(i)}(\mathbf{x}, t)$ is not physiological, but allows for the smooth scalar test problems described in Section 2.2. For the CICR problem, we set $f^{(i)} \equiv 0$ for all i.

The model uses the boundary conditions

$$\mathbf{n} \cdot \left(D^{(i)} \nabla u^{(i)} \right) = 0 \quad \text{for } x \in \partial\Omega,\ 0 < t \leq t_{\text{fin}} \tag{2.7}$$

and has the given initial conditions

$$u^{(i)}(\mathbf{x}, 0) = u_{\text{ini}}^{(i)}(\mathbf{x}) \quad \text{for } x \in \Omega,\ t = 0. \tag{2.8}$$

In the abscence of advection, these boundary conditions can be seen as no-flux boundary conditions. The concentration of calcium, $i = 1$, is initialized at basal level of $u_{\text{ini}}^{(i)} \equiv 0.1$ μM. The initial values of $u_{\text{ini}}^{(i)}$ for $i = 2, 3$ are calculated, such that the reaction terms $r^{(i)}$ on the right-hand side 1.2 cancel. The values of the initial concentrations are listed in Table 1. Thus, initially all terms on the right-hand side of (1.2) vanish, and our simulations address the question whether a calcium wave can self-initiate.

To demonstrate the effect of advection, we extend the model by artificially adding advection to the right in the z-direction, i.e., setting $\beta^{(i)} = (0, 0, \beta_3^{(i)})^T$, $i = 1, \ldots, n_s$, with $\beta_3^{(i)} > 0$. This changes the behavior at the boundaries compared to the purely diffusive case. In particular, the species flow in from the left and leave to the right end of the domain. Applying the numerical method to this non-linear CICR application problem in Sections 4 and 6 is the key extension to [10].

2.2 Scalar Test Problems

To demonstrate convergence and show weak scalability, we consider two scalar test problems, which are simplifications of the system (1.1). Here, we only consider the advection-diffusion-reaction equation, where there is only one species, hence we drop the superscripts. To maintain the connection to the application problem, the domain Ω is chosen to be the same as in the calcium problem as is the diffusion coefficient matrix $D = D^{(1)} \equiv \text{diag}(0.15, 0.15, 0.30)$. The advection velocity is designed as product of a weight ω and vector $(0.15, 0.15, 0.30)^T$ of the form $\beta \equiv \beta^{(1)} = \omega\,(0.15, 0.15, 0.30)^T$ such that we can control the magnitude of advection by varying the constant ω. For $\omega = 0$, there is no advection, and for $\omega = 1$, diffusion and advection are on the same order of magnitude. The

two scalar test problems are different in their right-hand side (1.2), as explained in the following subsections. Also to maintain the connection to the application problem, both scalar test problems use the same boundary conditions as the application problem specified above. The two-dimensional test problems are set up the same as the three-dimensional ones, except that the y-dimension is dropped from the domain and model parameters, that is, the domain retains its elongated shape in the last dimension. These scalar test problems are used in Sections 5 and 6.

2.2.1 Smooth Source Term

With respect to the right-hand side of the scalar test problem with smooth source term, we set $r \equiv 0$ and $s \equiv 0$ in (1.2). This results in the problem $u_t - \nabla \cdot (D\,\nabla u) + \beta \cdot \nabla u = f$. The right-hand side f is chosen such that the true solution is exactly known as

$$u(x,y,z,t) = \frac{1+\cos(\lambda_x x)e^{-D_x\lambda_x^2 t}}{2}\,\frac{1+\cos(\lambda_y y)e^{-D_y\lambda_y^2 t}}{2}\,\frac{1+\cos(\lambda_z z)e^{-D_z\lambda_z^2 t}}{2} \tag{2.9}$$

with $\lambda_x = \lambda_y = \pi/6.4$ and $\lambda_z = \pi/32$. The initial conditions are chosen consistent with the true solution.

2.2.2 Non-smooth Source Term

To show the method's convergence even in the presence of point sources, we set $f \equiv 0$ and $r \equiv 0$ in (1.2), and $J_{\text{pump}} \equiv J_{\text{leak}} \equiv 0$ in (2.1). The intention is to simplify the calcium problem by modelling a single CRU in the center of the domain, which opens at time $t = 1$ and remains open afterwards. Therefore, we set $\Omega_s = \{(0,0,0)\}$, $s \equiv s^{(1)}(u,\mathbf{x},t) = g\,S_{\hat{\mathbf{x}}}(u,t)\,\delta(\mathbf{x}-\hat{\mathbf{x}})$ with $\hat{\mathbf{x}} \in \Omega_s$ and g from Table 1, and control manually $S_{\hat{\mathbf{x}}}(u,t) = 0$ for $t < 1$ and $S_{\hat{\mathbf{x}}}(u,t) = 1$ for $t \geq 1$. This results in the problem $u_t - \nabla \cdot (D\,\nabla u) + \beta \cdot \nabla u = g\,S_{\hat{\mathbf{x}}}(u,t)\,\delta(\mathbf{x}-\hat{\mathbf{x}})$. The true solution is not available for this test problem. To maintain the connection to the application problem, the initial condition is chosen as $u_{\text{ini}} \equiv 0.1$.

Table 1: Table of parameters for the CICR model.

Parameter	Description	Values/Units
t	Time	ms
$\mathbf{x}$	Position	μm
u^i	Concentration	μM
Ω	Rectangular domain in μm	$(-6.4, 6.4) \times (-6.4, 6.4) \times (-32.0, 32.0)$
$D^{(1)}$	Calcium diffusion coefficient	$\text{diag}(0.15, 0.15, 0.30)$ μm^2/ ms
$D^{(2)}$	Mobile buffer diffusion coefficient	$\text{diag}(0.01, 0.01, 0.02)$ μm^2/ ms
$D^{(3)}$	Stationary buffer diffusion coefficient	$\text{diag}(0.00, 0.00, 0.00)$ μm^2/ ms
$\beta^{(i)}$	Advection velocity	μm / ms
$u_{\text{ini}}^{(1)}$	Initial calcium concentration	0.1 μM
$u_{\text{ini}}^{(2)}$	Initial mobile buffer concentration	45.9184 μM
$u_{\text{ini}}^{(3)}$	Initial stationary buffer concentration	111.8182 μM
Δx_s	CRU spacing in x-direction	0.8 μm
Δy_s	CRU spacing in y-direction	0.8 μm
Δz_s	CRU spacing in z-direction	0.2 μm
g	Flux density distribution	110.0 μM μm^3 / ms
P_{max}	Maximum probability rate	0.3 / ms
K_{prob}	Probability sensitivity	0.2 μM
n_{prob}	Probability Hill coefficient	4.0
Δt_s	CRU time step	1.0 ms
t_{open}	CRU opening time	5.0 ms
t_{closed}	CRU refractory period	100 ms
k_2^+	Forward reaction rate	0.08 / (μM ms)
k_2^-	Backward reaction rate	0.09 / ms
$\bar{u}_2$	Total of bound and unbound indicator	50.0 μM
k_3^+	Forward reaction rate	0.10 / (μM ms)
k_3^-	Backward reaction rate	0.10 / ms
$\bar{u}_3$	Total bound and unbound buffer	123.0 μM
V_{pump}	Maximum pump strength	4.0 μM / ms
K_{pump}	Pump sensitivity	0.184 μM
n_{pump}	Pump Hill coefficient	4
J_{leak}	Leak term	0.320968365152510 μM / ms

3 NUMERICAL METHOD

In this section we describe the numerical method for the solution of the system (1.1). In Section 3.1 we give an outline of the finite volume space discretization, a detailed explaination can be found in [10]. A brief description of the time integration, the non-linear solver, and the linear solver is given in Section 3.2. Section 3.3 discusses the parallel implementation using MPI.

3.1 Spatial Discretization

To derive the finite volume discretization, let $\mathcal{T}_h = \{K_1, \ldots, K_M\}$ be a mesh such that $\overline{\Omega} = \bigcup_{l=1}^{M} \overline{K}_l$. Each $K_l \in \mathcal{T}_h$ is an open subset of Ω and referred to as cell or control volume. Integating (1.1) over an arbitrary cell $K_l \in \mathcal{T}_h$ and applying the divergence theorem yields

$$\frac{d}{dt}\int_{K_l} u^{(i)}\,d\mathbf{x} - \int_{\partial K_l} (D^{(i)}\nabla u^{(i)} - u^{(i)}\beta^{(i)}) \cdot \mathbf{n}_l\,dS = \int_{K_l} q^{(i)}\,d\mathbf{x}, \tag{3.1}$$

where ∂K_l denotes the boundary of K_l and $\mathbf{n}_l$ its outward unit normal vector. This is the equation we are actually trying to solve, since it imposes less regularity on the solution than (1.1). In particular, solutions with discontinuities are now admissible. Denoting the volume of K_l by $|K_l|$, the spatial mean value of $u^{(i)}$ over K_l is given by

$$\bar{u}_l^{(i)}(t) := \frac{1}{|K_l|}\int_{K_l} u^{(i)}(\mathbf{x}, t)\,d\mathbf{x}. \tag{3.2}$$

With this notation, (3.1) can be rewritten as

$$\frac{d}{dt}\bar{u}_l^{(i)} - \frac{1}{|K_l|}\int_{\partial K_l} (D^{(i)}\nabla u^{(i)} - u^{(i)}\beta^{(i)}) \cdot \mathbf{n}_l\,dS = \frac{1}{|K_l|}\int_{K_l} q^{(i)}\,d\mathbf{x}. \tag{3.3}$$

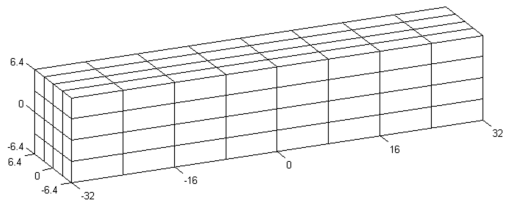
This is a system of ordinary differential equations for the temporal evolution of the mean values $\bar{u}_l^{(i)}$. With this expression, the crucial issues are to compute the boundary fluxes in terms of neighboring mean values as well as the volume integral on the right hand side.

Now, we introduce the mesh which is used for the discretization. In [3], a regular mesh $\Omega_h \subset \overline{\Omega}$ with constant mesh spacings Δx, Δy, and Δz was used for the finite element space discretization. Here we employ the corresponding dual mesh $\mathcal{T}_h$, which is constructed by connecting all centers of mesh cubes of Ω_h with lines parallel to the coordinate axes and extending these lines in a straight manner to the boundary. The dual mesh is a rectilinear mesh and each inner node of Ω_h is the center of a cell of $\mathcal{T}_h$ with volume $\Delta x\,\Delta y\,\Delta z$. Furthermore, the volume of a cell is reduced to $\Delta x\Delta y\Delta z/2$, $\Delta x\Delta y\Delta z/4$, or $\Delta x\Delta y\Delta z/8$, if this cell has a common face, edge, or corner, respectively, with the boundary $\partial\Omega$. The exterior views of a simple regular mesh and its corresponding dual mesh are depicted in Figures 1 (a) and (b), respectively.

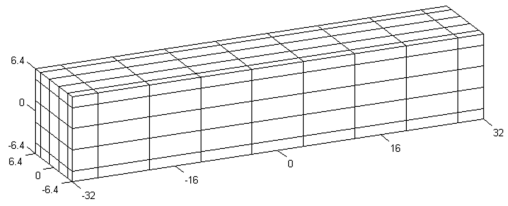
By construction, the number of nodes of Ω_h equals the number of cells in $\mathcal{T}_h$. In the following, we assume that $\mathcal{T}_h$ consists of $M = M_x M_y M_z$ control volumes, with M_x, M_y, and M_z denoting the number of cells in each direction and introduce the enumeration scheme

$$l = i + (j-1)M_x + (k-1)M_xM_y \tag{3.4}$$

for $1 \le i \le M_x$, $1 \le j \le M_y$, and $1 \le k \le M_z$. Thus, a control volume K_l has the neighbors K_{l-1} and K_{l+1} in the x-direction, K_{l-M_x} and K_{l+M_x} in the y-direction and $K_{l-M_xM_y}$ and $K_{l+M_xM_y}$ in the z-direction.



(a)



(b)

Figure 1: (a) Regular mesh of mesh resolution $4 \times 4 \times 8$ and (b) its dual mesh.

For a given control volume $K_l = (x_L, x_R) \times (y_L, y_R) \times (z_L, z_R) \in \mathcal{T}_h$ with spacings $\Delta x_l = x_R - x_L, \Delta y_l = y_R - y_L, \Delta z_l = z_R - z_L$, and $\beta = (\beta_1, \beta_2, \beta_3)$, the boundary fluxes from (3.1) can be written as

$$\begin{aligned}
&\int_{\partial K_l} (D\nabla u - u\beta) \cdot \mathbf{n}_l \, dS \qquad (3.5)\\
&\quad = \int_{z_L}^{z_R} \int_{y_L}^{y_R} (D_{11}\partial_x u - \beta_1 u)|_{x=x_R} - (D_{11}\partial_x u - \beta_1 u)|_{x=x_L} \, dy \, dz \\
&\quad + \int_{z_L}^{z_R} \int_{x_L}^{x_R} (D_{22}\partial_y u - \beta_2 u)|_{y=y_R} - (D_{22}\partial_y u - \beta_2 u)|_{y=y_L} \, dx \, dz \\
&\quad + \int_{y_L}^{y_R} \int_{x_L}^{x_R} (D_{33}\partial_z u - \beta_3 u)|_{z=z_R} - (D_{33}\partial_z u - \beta_3 u)|_{z=z_L} \, dx \, dy.
\end{aligned}$$

Here we exploited the fact that the faces of K_l are parallel to the planes defined by the coordinate axes, thus only one entry of the corresponding normal vectors is non-zero. Note that the superscripts (i) were dropped for readability and that the above equation is valid for one of the n_s species of the system.

In order to approximate the advective flux $\beta_j u(x_{l+1})$ in terms of the mean values $\bar{u}_l$, we introduce a numerical flux function $H = H(\hat{u}_l^+, \hat{u}_{l+1}^-)$. The values $\hat{u}_l^+$ and $\hat{u}_{l+1}^-$ are approximations to the unknown values of u on either side of x_{l+1}. Assuming that $\beta_j \geq 0$ as before, either a first-order or second-order accurate upwind flux function can be defined:

- Assuming a constant distribution of u in K_l leads to the first-order discretization

$$H_{a,j}(\hat{u}_l^+, \hat{u}_{l+1}^-) = \beta_j \hat{u}_l^+, \quad j = 1, 2, 3, \qquad \hat{u}_l^+ = \bar{u}_l. \qquad (3.6)$$

- Since the advection term is linear, no non-linear discretization is necessary, as would be standard for non-linear fluid mechanics. To obtain the second-order discretization via a

linear representation of the solution, we define a slope within the cell by

$$H_{a,j}(\hat{u}_l^+, \hat{u}_{l+1}^-) = \beta_j \hat{u}_l^+, \quad j = 1,2,3, \qquad \hat{u}_l^+ = \bar{u}_l + \frac{\bar{u}_l - \bar{u}_{l-1}}{\mathbf{m}_l - \mathbf{m}_{l-1}}(x_{l+1} - \mathbf{m}_l), \tag{3.7}$$

where $\mathbf{m}_l$ and $\mathbf{m}_{l-1}$ denote the barycenters of the cells belonging to $\bar{u}_l$ and $\bar{u}_{l-1}$, respectively.

To approximate the diffusive flux $D_{jj}\partial_l u(x_{l+1})$, we use a central difference of the neighboring mean values and define the diffusive flux function as

$$H_{d,j}(\bar{u}_l, \bar{u}_{l+1}) = D_{jj}\frac{\bar{u}_{l+1} - \bar{u}_l}{\mathbf{m}_{l+1} - \mathbf{m}_l}. \tag{3.8}$$

In this case, the mean values on either side of x_{l+1} are sufficient to obtain a scheme of second order. Now we can approximate the boundary fluxes as

$$\begin{aligned}
&\int_{\partial K_l} (D\nabla u - u\beta)\cdot \mathbf{n}_l \, dS \\
&\quad \approx \Delta y_l \Delta z_l \big(H_{d,1}(\bar{u}_l, \bar{u}_{l+1}) - H_{a,1}(\hat{u}_l^{1,+}, \hat{u}_{l+1}^{1,-})\big) \\
&\quad - \Delta y_l \Delta z_l \big(H_{d,1}(\bar{u}_{l-1}, \bar{u}_l) - H_{a,1}(\hat{u}_{l-1}^{1,+}, \hat{u}_l^{1,-})\big) \\
&\quad + \Delta x_l \Delta z_l \big(H_{d,2}(\bar{u}_l, \bar{u}_{l+M_x}) - H_{a,2}(\hat{u}_l^{2,+}, \hat{u}_{l+M_x}^{2,-})\big) \\
&\quad - \Delta x_l \Delta z_l \big(H_{d,2}(\bar{u}_{l-M_x}, \bar{u}_l) - H_{a,2}(\hat{u}_{l-M_x}^{2,+}, \hat{u}_l^{2,-})\big) \\
&\quad + \Delta x_l \Delta y_l \big(H_{d,3}(\bar{u}_l, \bar{u}_{l+M_x M_y}) - H_{a,3}(\hat{u}_l^{3,+}, \hat{u}_{l+M_M M_y}^{3,-})\big) \\
&\quad - \Delta x_l \Delta y_l \big(H_{d,3}(\bar{u}_{l-M_x M_y}, \bar{u}_l) - H_{a,3}(\hat{u}_{l-M_x M_y}^{3,+}, \hat{u}_l^{3,-})\big),
\end{aligned} \tag{3.9}$$

where the enumeration scheme (3.4) was used to describe the location of the input data of the flux functions. The notation $\hat{u}_l^{j,\pm}$ indicates that the approximation to the value of u lives in K_l and belongs to the face which is between K_l and its neighbor of positive or negative side.

The last step of the discretization is the treatment of the volume integral on the right hand side of (3.3). This can be approximated sufficiently using the midpoint rule

$$\int_{K_l} q(u^{(1)}, \ldots, u^{(n_s)}, \mathbf{x}, t)\, d\mathbf{x} \approx |K_l|\, q(\bar{u}_l^{(1)}, \ldots, \bar{u}_l^{(n_s)}, \mathbf{m}_l, t), \tag{3.10}$$

with $\mathbf{m}_l$ denoting the barycenter of K_l. In the special case of a Dirac delta distribution as source term, i.e., $q = \delta(\mathbf{x} - \hat{\mathbf{x}})$, the volume integral can be computed exactly. The Dirac delta distribution is defined by requiring $\delta(\mathbf{x} - \hat{\mathbf{x}}) = 0$ for all $\mathbf{x} \neq \hat{\mathbf{x}}$ and $\int_{\mathbb{R}^3} \psi(\mathbf{x})\delta(\mathbf{x} - \hat{\mathbf{x}})\, d\mathbf{x} = \psi(\hat{\mathbf{x}})$ for any function $\psi \in C_0^\infty(\mathbb{R}^3)$. Thus, we obtain

$$\int_{K_l} q(\mathbf{x})\, d\mathbf{x} = \int_{K_l} \delta(\mathbf{x} - \hat{\mathbf{x}}) \cdot 1\, d\mathbf{x} = \begin{cases} 1, & \hat{\mathbf{x}} \in K_l, \\ 0, & \hat{\mathbf{x}} \notin K_l. \end{cases} \tag{3.11}$$

This completes the description of the finite volume discretization and, after division by the local

volume $|K_l| = \Delta x_l \Delta y_l \Delta z_l$, (3.3) can now be written as

$$\begin{aligned}
&\frac{d}{dt}\bar{u}_l - \frac{1}{\Delta x_l}\big(H_{d,1}(\bar{u}_l, \bar{u}_{l+1}) - H_{a,1}(\hat{u}_l^{1,+}, \hat{u}_{l+1}^{1,-})\big) \\
&\quad + \frac{1}{\Delta x_l}\big(H_{d,1}(\bar{u}_{l-1}, \bar{u}_l) - H_{a,1}(\hat{u}_{l-1}^{1,+}, \hat{u}_l^{1,-})\big) \\
&\quad - \frac{1}{\Delta y_l}\big(H_{d,2}(\bar{u}_l, \bar{u}_{l+M_x}) - H_{a,2}(\hat{u}_l^{2,+}, \hat{u}_{l+M_x}^{2,-})\big) \\
&\quad + \frac{1}{\Delta y_l}\big(H_{d,2}(\bar{u}_{l-M_x}, \bar{u}_l) - H_{a,2}(\hat{u}_{l-M_x}^{2,+}, \hat{u}_l^{2,-})\big) \\
&\quad - \frac{1}{\Delta z_l}\big(H_{d,3}(\bar{u}_l, \bar{u}_{l+M_xM_y}) - H_{a,3}(\hat{u}_l^{3,+}, \hat{u}_{l+M_xM_y}^{3,-})\big) \\
&\quad + \frac{1}{\Delta z_l}\big(H_{d,3}(\bar{u}_{l-M_xM_y}, \bar{u}_l) - H_{a,3}(\hat{u}_{l-M_xM_y}^{3,+}, \hat{u}_l^{3,-})\big) \\
&\quad = q(\bar{u}_l^{(1)}, \ldots, \bar{u}_l^{(n_s)}, \mathbf{m}_l, t).
\end{aligned} \tag{3.12}$$

(3.13)

Let $\bar{\mathbf{u}}^{(i)} = (\bar{u}_1^{(i)}, \ldots, \bar{u}_M^{(i)})^T$, $\mathbf{q}^{(i)} = (q_1^{(i)}, \ldots, q_M^{(i)})^T$, $q_l^{(i)} = q^{(i)}(\bar{u}_l^{(1)}, \ldots, \bar{u}_l^{(n_s)}, \mathbf{m}_l, t)$, (3.12) now reads

$$\frac{d}{dt}\bar{\mathbf{u}}^{(i)} = \big(\mathbf{H}_{\text{diff}}^{(i)} - \mathbf{H}_{\text{adv}}^{(i)}\big)\bar{\mathbf{u}}^{(i)} + \mathbf{q}^{(i)}(\bar{\mathbf{u}}^{(1)}, \ldots, \bar{\mathbf{u}}^{(n)}), \tag{3.14}$$

where $\mathbf{H}_{\text{diff}}, \mathbf{H}_{\text{adv}} \in \mathbb{R}^{M\times M}$ are the flux matrices, built from terms containing $H_{d,j}$ and $H_{a,j}$ in the system of equations (3.12). Finally, collecting all n_s vectors $\bar{\mathbf{u}}^{(i)}$ in $\bar{\mathbf{U}} \in \mathbb{R}^{n_sM}$ the system (3.14) can be written as

$$\frac{d}{dt}\bar{\mathbf{U}}(t) = \mathbf{f}^{\text{ode}}(t, \bar{\mathbf{U}}(t)) \tag{3.15}$$

with $\mathbf{f}^{\text{ode}} = (\mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(n)})^T \in \mathbb{R}^{n_sM}$ and components

$$\mathbf{f}^{(i)} = \mathring{A}^{(i)}\bar{\mathbf{u}}^{(i)} + \mathbf{q}^{(i)}(\bar{\mathbf{u}}^{(1)}, \ldots, \bar{\mathbf{u}}^{(n_s)}), \tag{3.16}$$

with

$$\mathring{A}^{(i)} = \mathbf{H}_{diff}^{(i)} - \mathbf{H}_{adv}^{(i)}. \tag{3.17}$$

3.2 Time Integration and Matrix-Free Implementation

The spatial discretization of the application problem (3.15) with $n_s = 3$ species using the finite volume method with $M = M_x M_y M_z$ control volumes results in a system of non-linear ordinary differential equations (ODEs) with $n_{eq} = n_s M$ degrees of freedom (DOF). A method of lines discretization of advection-diffusion-reaction equations with second-order spatial derivatives results necessarily in a stiff ODE system, since the time step size restrictions due to the CFL condition are considered too severe to allow for explicit time-stepping methods. To reach the very large final times demanded to simulate laboratory time scales of the CICR application, we need to be able to take fairly large time steps whenever possible. This necessitates the use of a sophisticated ODE solver such as the family of numerical differentiation formulas (NDFk) with variable order $1 \leq k \leq 5$ and adaptively chosen time step size [12, 13]. This method is also used for a method of lines discretization using finite elements in [3]. We use relative and absolute tolerances of 10^{-6} and 10^{-8}, respectively, for the error estimator of the NDFk method. In studies for the CICR problem, the time step sizes vary widely, with fine step sizes on the order of 10^{-5} ms immediately after CRUs open or close. Since the parabolic system is smooth away from these times, the time steps increase steadily up to the order of 10^{-2} ms, all the while the error controller ensures that the total error incurred from the time-stepping remains bounded by the selected tolerances. This high variation in step size allows the solver to reach the desired final time of 1,000 ms in under 90,000 time steps for the finest mesh. The average method order observed is 3, showing that we are profiting significantly from the variable order method.

The implicit ODE method needed for a stiff problem demands the solution of a non-linear system $\mathbf{F}^{newt}(\bar{\mathbf{U}}) = 0$ of n_{eq} equations at every time step. The Newton method is used with Jacobian $\mathbf{J}^{newt}(\bar{\mathbf{U}}) = \nabla_{\bar{\mathbf{U}}} \mathbf{F}^{newt}(\bar{\mathbf{U}})$. This method profits from the low-order spatial discretization on a uniform mesh used, because we are able to compute analytically all matrices in (3.12), as originally demonstrated in [4] for the method of lines using finite elements. The purpose of this approach is to save memory and hence allow for computations on very fine meshes. In addition, the usage of the exact Jacobian should lead to quadratic convergence of the Newton method. The iteration is stopped if $\|\bar{\mathbf{U}}_{\text{new}}\| < \varepsilon^{\text{newt}} \|\bar{\mathbf{U}}_{\text{new}} - \bar{\mathbf{U}}_{\text{old}}\|$. We use the tolerance of $\varepsilon^{\text{newt}} = 10^{-4}$ and maximum number of Newton iterations of 4. Since the matrix-vector products in Krylov subspace methods [9] used as linear solvers below are implemented in matrix-free form, this Jacobian is automatically evaluated at the current Newton iteration without any additional cost.

At each Newton iteration, we need to solve a linear system with a non-symmetric system matrix for the n_{eq} unknowns. Numerical experiments demonstrate that the biconjugate gradient stabilized method (BiCGSTAB) is preferable over GMRES as well as to QMR, the latter one being used with the finite element method in [3]. We stop the iteration within the linear solver if the residual $\mathbf{r}$ satisfies the condition $\|\mathbf{r}\|_2 < \varepsilon^{lin} \|\mathbf{b}\|_2$ with $\mathbf{b}$ denoting the right-hand side of the linear system and a given tolerance ε^{lin}. We use a tolerance of $\varepsilon^{lin} = 10^{-6}$.

3.3 Parallel Implementation

The code uses MPI to perform the parallel communications and is an extension of the one described in [3]. To compute the matrix-vector products needed in the Krylov subspace methods, communications between neighboring processes are required. Non-blocking communications using `MPI_Isend` and `MPI_Irecv` are used for neighboring process communications. Furthermore, `MPI_Allreduce` is used for inner products and norm calculation. In the parallel implementation, all data are split in the z-direction such that the M_z mesh points are block-distributed to the p parallel processes. The division of the domain Ω into p subdomains in the long z-direction (instead of the shorter x- or y-directions) minimizes the amount of data that neighboring processes need to exchange across

the subdomain interfaces. For a fixed problem size with DOF approximately 25 million, parallel computing using MPI demostrates good strong scalability [10]. However, each process has less computation to perform after increasing the number of MPI processes, resulting in the degradation of speedup and efficiency. Section 6 discusses weak scability, to demonstate how efficcient the implementation is as we increase problem size, all the way to approximately 800 million DOF.

Regarding the advection matrix, using a constant distribution or a linear reconstruction yields different loads of communications between processes, which correcponds to first- or second-order schemes, see (3.6) and (3.7) for details. This raises the question whether it is preferable to use the first-order advection scheme, which sacrifices accuracy for smooth problems but restricts the communication with each neighboring process to one slice, or to use the second-order reconstruction and to accept the exchange of two slices between neighboring processes. This is a key question to be analyzed in Section 5.

But first, the parallel code enables the effective solution of the physiological application problem for long-time simulations in Section 4.

4 SIMULATION OF CALCIUM WAVES

We solve the model of CICR from Section 2.1 given by the system of coupled, time-dependent advection-reaction-diffusion equations (1.1)–(1.2), where the calcium injection is modeled by (2.2) with a constant uniform CRU flux density g. The parameters are given in Table 1. This section analyzes the influence of increasing advection in (1.1). The initial conditions are as specified in Section 2.1. All studies use the same seed to the random number sequence to allow for a physiological comparison of the simulations.

We first take the case where there are no advection effects, like in [3] and [2], to show three ways to visualize the simulation results. In this case, we have several waves self initiate and propagate throughout the cell, in both orientations of the z-direction. The first plotting method is called a CRU Plot, shown in Figure 2. The long-time simulations of CICR model go up to the large final time $t_{\text{fin}} = 1{,}000$ ms, as noted in Section 2.1. The plots in this figure show which CRUs are open at each timestep during the simulation. We see that at $t = 100$ a few CRUs are open, the wave mostly spreads along x and y dimensions at this point. Later on we see that the CRUs have begun to open on both sides of the cell and spread across it. During our simulation of 1,000 ms, several waves have been generated and run across the cell, with similar speed on both ways of the z-direction. The second plotting method is called an Isosurface Plot, shown in Figure 3. The plots in the figure take the same time steps as in the CRU plots, but they show calcium concentration instead of open CRUs. The Isosurface Plots give us a 3-dimensional representation how the calcium diffuses through the cell based on the concentration of calcium species u_1. The different shades of blue, red and yellow indicate the level of calcium concentration throughout the cell, while red indicates higher concentration, blue indicates lower. There is a critical value associated with our plots, $u_{\text{crit}} = 65$ μM. Inside the plotted area, the concentration is higher than u_{crit}, while outside the plotted area the concentration is lower than u_{crit}. On the boundary of the plotted area inside the cell the concentration is equal to u_{crit}, and the color is blue. However on the boundary of the cell domain, the concentration might be higher than u_{crit}, hence the color more is red. Again, we see that when $t = 100$ in Figure 3 there is a small amount of calcium in the cell. As time advances, we see that the amount of calcium in the cell increases and diffuses throughout the cell. The third plotting method is called a Confocal Image Plot, shown in Figure 4. The confocal images are meant to replicate what scientists see in the laboratory experiements using florescent dye to bind to the calcium in the heart cell. The lighter of green shades indicate higher calcium concentrations, while the darker green shades indicate lower concentrations of calcium. When $t = 100$, we see calcium start to diffuse across the cell as shown in Figure 4.

The key motivation to expand the current model to include advection effects comes from work such as [1] that details several mechanisms for intracellular calcium pathology and [14] that discusses mechanisms for intracellular calcium waves to impact multicellular electrical arrhythmias. This paper introduces a simple advection affect with constant advection velocity vectors $\beta^{(i)}$ in (1.1) (i) to demonstrate that the model behaves correctly with advection and (ii) to test the convergence of the numerical method for the advection term in Section 5. We study advection in the elongated z-direction of the cell by setting $\beta^{(i)} = (0, 0, \beta_3^{(i)})$, $i = 1, \ldots, n_s$, in the general system (1.1). We vary the value of the z-component $\beta_3^{(i)} = 0.01, 0.03, 0.05, 0.1, 0.2$. Figures 5, 7, 9, 11, and 13 show CRU plots for these values of β_3, analogous to Figure 2 without advection. These plots show that with larger β_3 the effect of advection is stronger, as demonstrated by the waves been pushed progressively more in the positive z-direction.

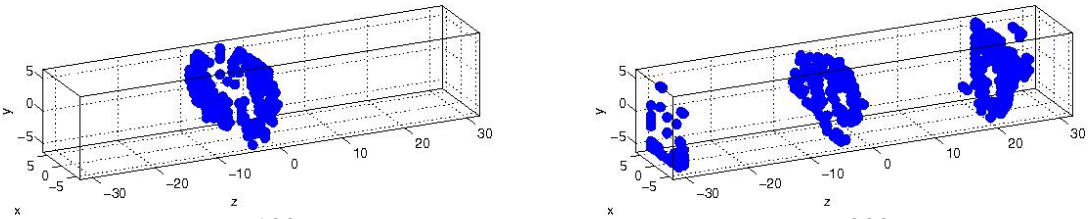
To demonstate the effect of advection, Figures 15 and 16 show line scan plots of the calcium concentration along the longitudinal axis of the cell. Each line scan plot shows the concentration at fixed values of x and y, for $-32 \leq z \leq 32$ on the vertical and $0 \leq t \leq 1{,}000$ on the horizontal

axis. The color represents the calcium concentration, with different shades of blue, yellow, and red indicating the level of calcium concentration, where blue indicates a lower concentration and red indicates a higher one.

Figure 15 shows line scan plots along the line segment for $x = y = 0$. Figure 15 (a) shows line scan plots when there is no advection. At $t = 0$ ms, we see constant blue, which represents the initial condition. At around $t = 50$ ms, we see lighter blue at some point P close to the center of the line segment, which means the calcium concentration is higher at this point. As t gets larger, we observe that the concentrations at neighboring points of P increase, and then the concentrations at the points further away increase, and so on. These points form two symmetric skew lines, capturing the propagation of the calcium waves along the line segment. We observe several pairs of symmetric skew lines, representing several waves, which we have observed in Figures 2, 3, and 4. The slopes of the lines are the speeds at which the calcium waves travel. In the case with no advection, the calcium wave travels at the same speed along the positive and negative z-direction, hence the skew lines are symmetric about the starting point P.

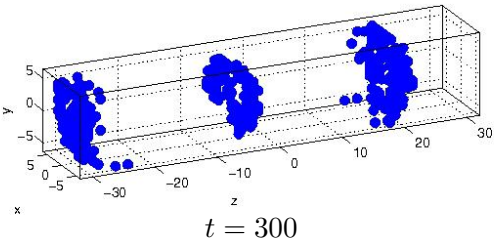
Figures 15 (b)–(f) show line scan plots with the z-component of advection velocity vector $\beta_3^{(i)} = 0.01, 0.03, 0.05, 0.1, 0.2$, respectively. We observe from Figure 15 (b) that the slopes of the skew lines on the positive side of P are slightly increased, meanwhile the slopes of the skew lines on the negative side of P are slightly decreased, compare to Figure 15 (a). This means that, as we add the effect of advection on the positive z-direction, the calcium wave traveling in the positive z-direction is faster, and the wave traveling in the negative z-direction is slower. We also observe from Figure 15 (c) that as we increase advection, the slopes of the skew lines on the positive side of P are further increased, and the concentrations on the skew lines are also higher. In the meantime, the slopes of the skew lines on the negative side of P decreased significantly, and the lines are much shorter. This indicates that the calcium waves cannot propagate to the far negative side of the domain. The CRUs there cannot open due to low calcium concentration. As the $\beta_3^{(i)}$ gets larger and larger, we observe from Figures 15 (d)–(f) that the slopes of the skew lines on the positive side of P gets larger and larger, indicating the speed at which calcium waves travel along the positive z-direction gets faster and faster. The calcium concentration on the skew lines are getting higher and higher as well. We also observe that fewer waves are generated. No lines are observed on the negative side of P. These plots match our observations from Figures 5, 7, 9, 11, and 13, and they clearly demonstrate the impact of advection on the speed of calcium waves.

Figure 16 shows line scan plots along the line segment for $x = y = 5.6$, which is near the edge of the domain Ω. In Figures 16 (a)–(f), We observe similar patterns just like Figures 15 (a)–(f), respectively, but the calcium concentrations are generally lower due to the position of the line segment in the domain. These plots confirm that we capture the effect of advection on calcium waves in the whole domain, not just along any single line segment.

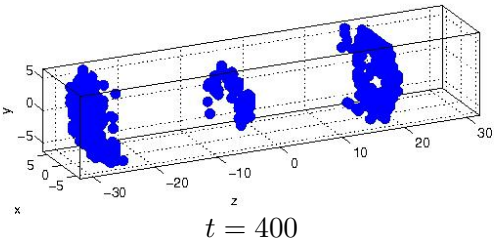


$t = 100$

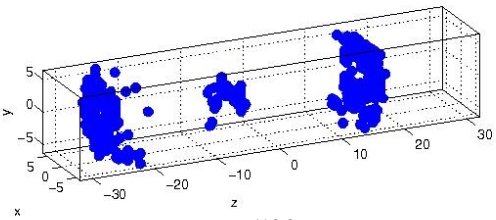
$t = 200$



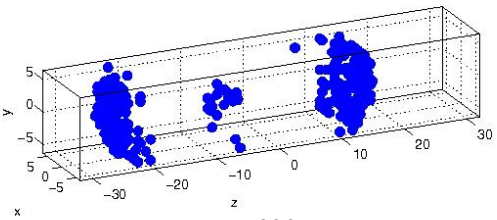
$t = 300$



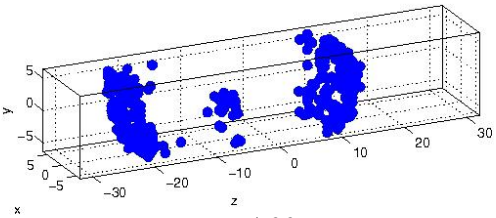
$t = 400$



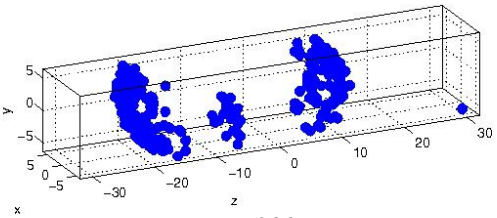
$t = 500$



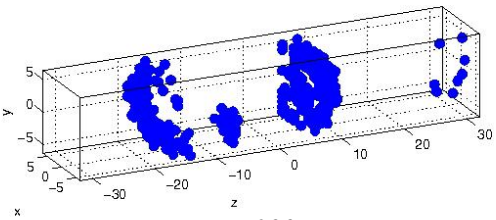
$t = 600$



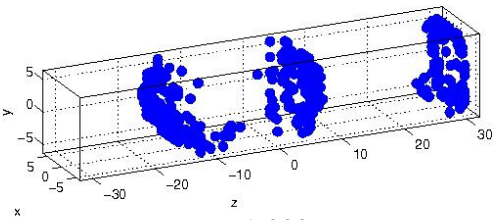
$t = 700$



$t = 800$



$t = 900$



$t = 1{,}000$

Figure 2: Open calcium release units throughout the cell using finite volume method without advection on mesh size $32 \times 32 \times 128$.

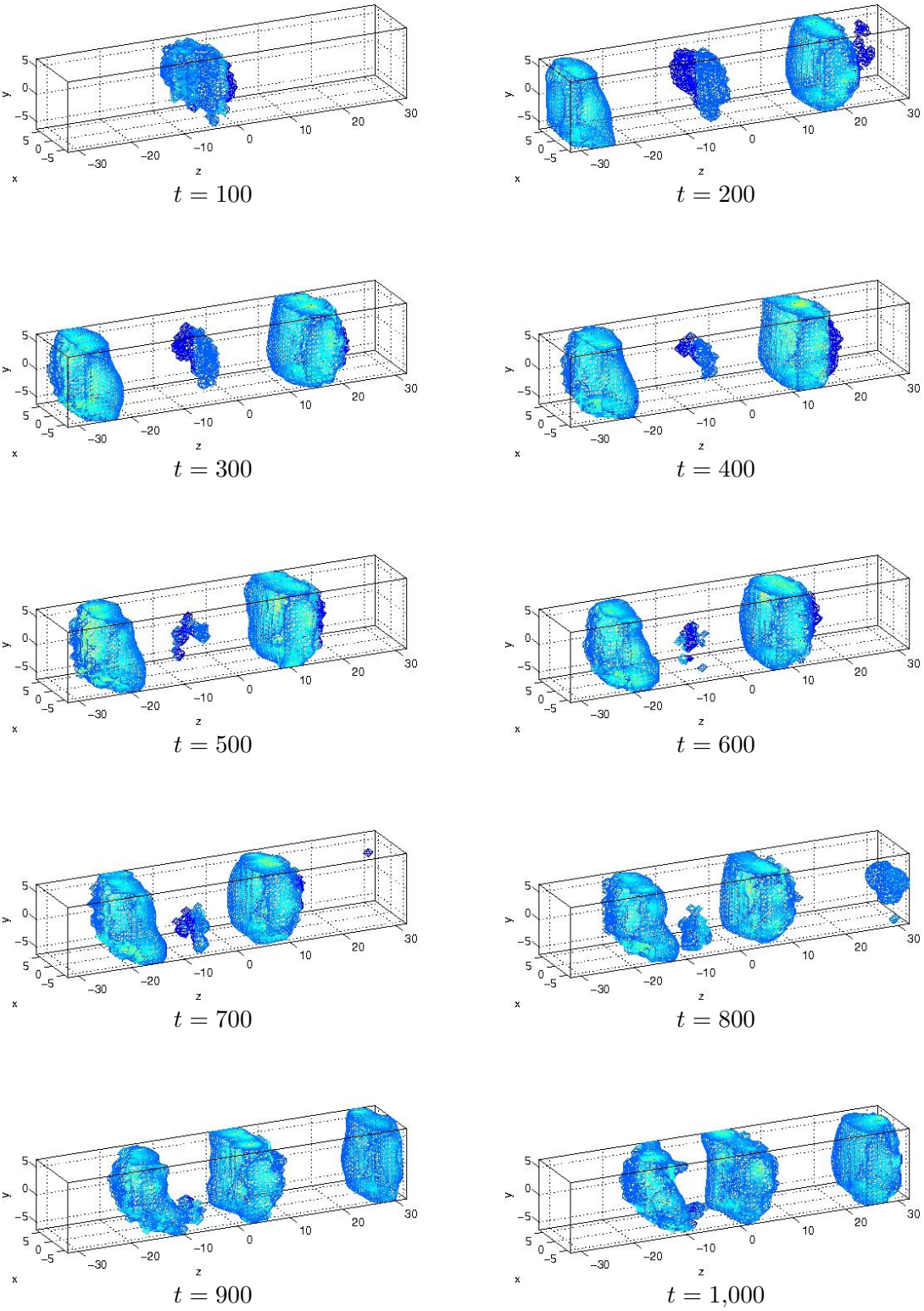


Figure 3: Isosurface plots of the calcium concentration using finite volume method without advection on mesh size $32 \times 32 \times 128$.

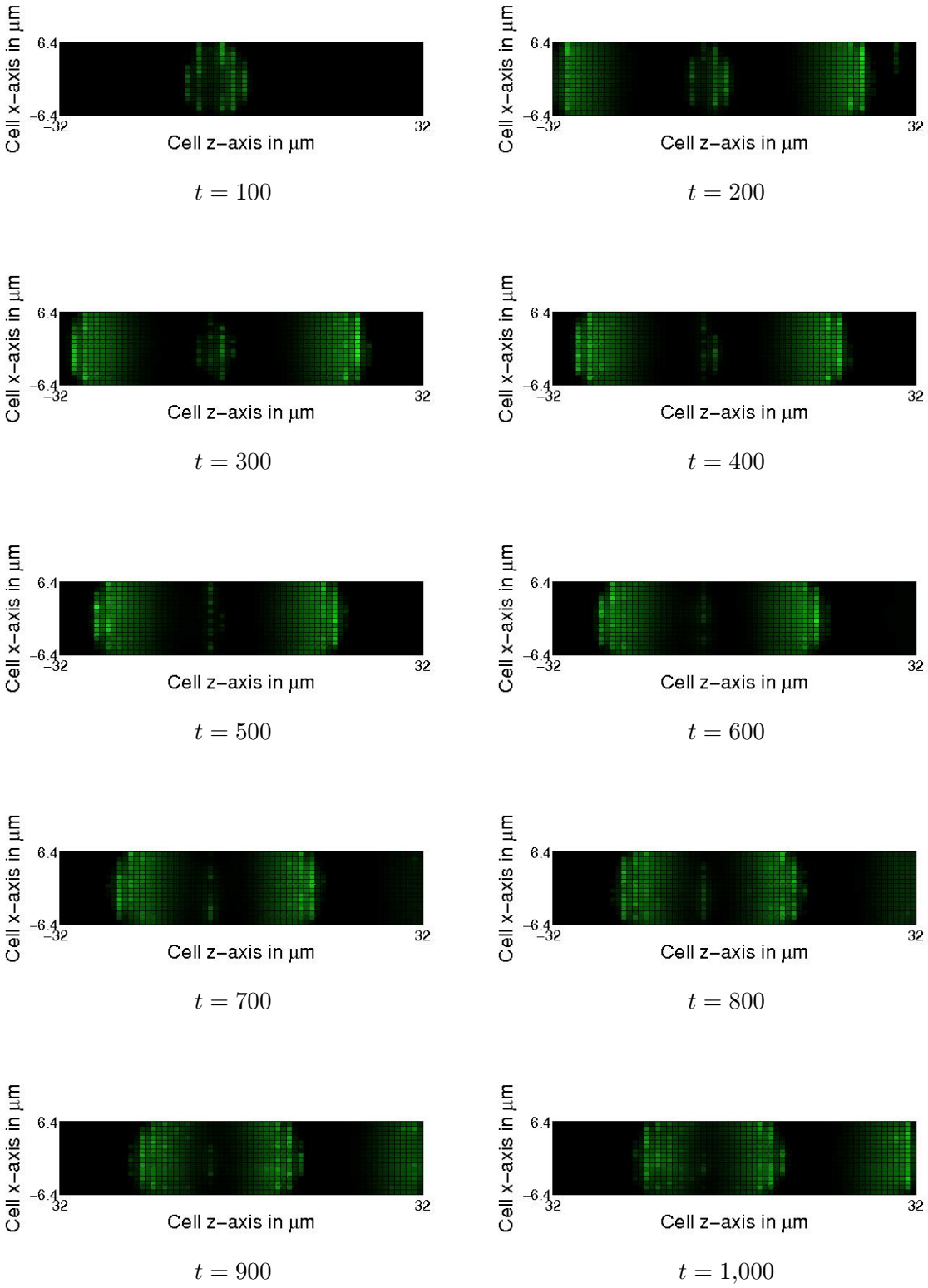
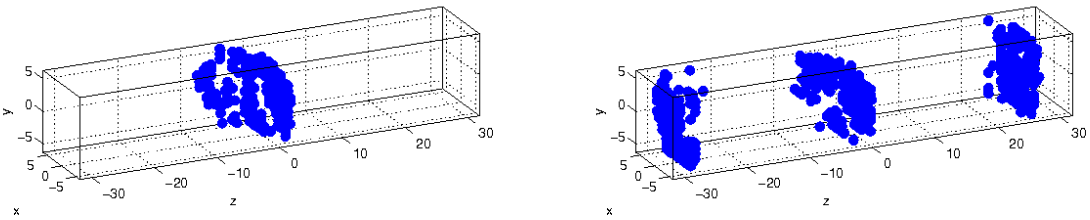
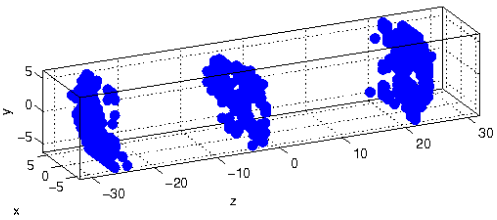


Figure 4: Confocal image plots of the calcium concentration using finite volume method without advection on mesh size $32 \times 32 \times 128$.

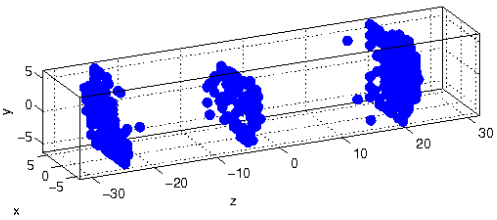


$t = 100$

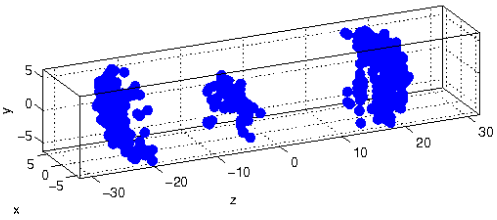
$t = 200$



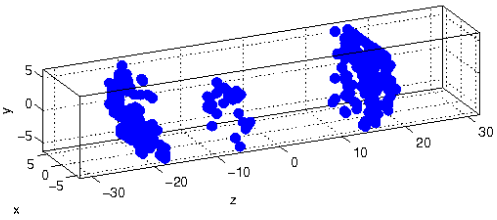
$t = 300$



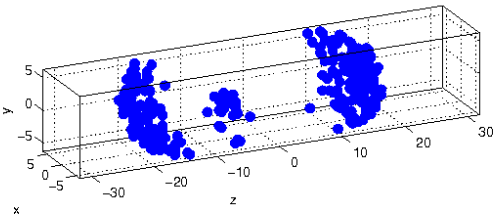
$t = 400$



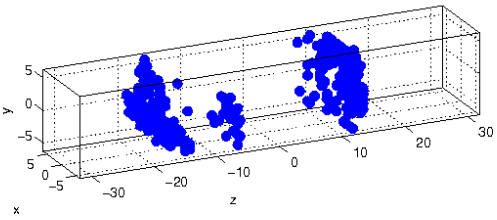
$t = 500$



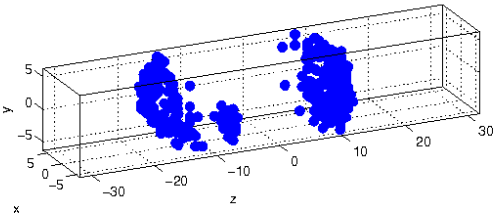
$t = 600$



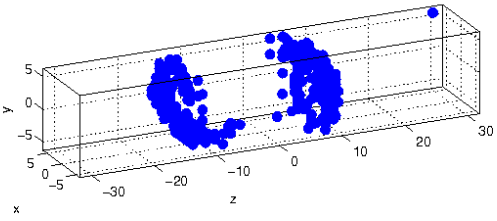
$t = 700$



$t = 800$

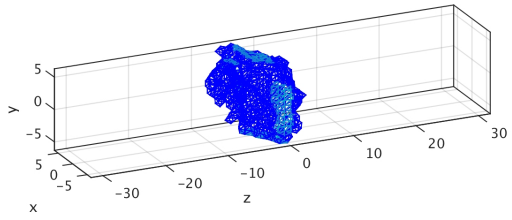


$t = 900$

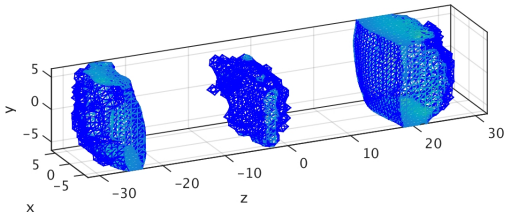


$t = 1{,}000$

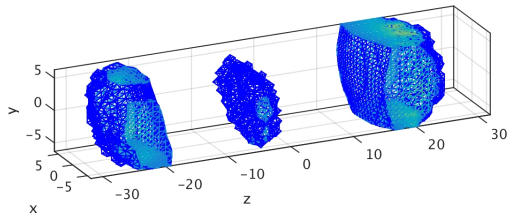
Figure 5: Open calcium release units throughout the cell using finite volume method with $\beta^{(i)} = (0, 0, 0.01), i = 1, \ldots, n_s$, on mesh size $32 \times 32 \times 128$.



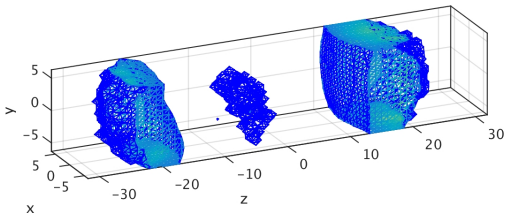
$t = 100$



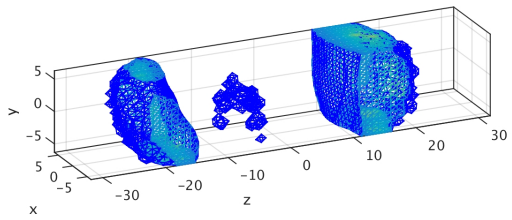
$t = 200$



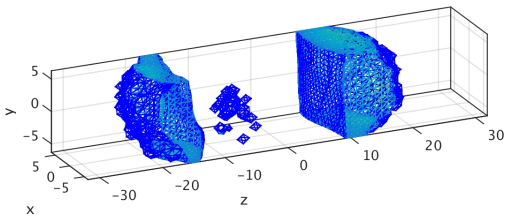
$t = 300$



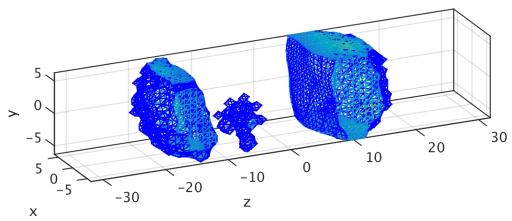
$t = 400$



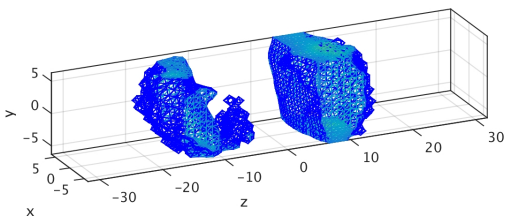
$t = 500$



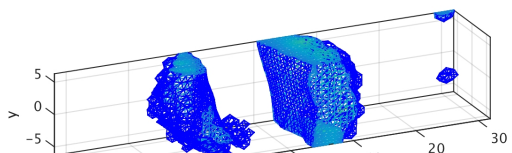
$t = 600$

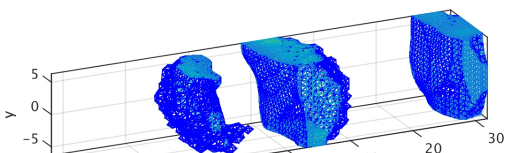


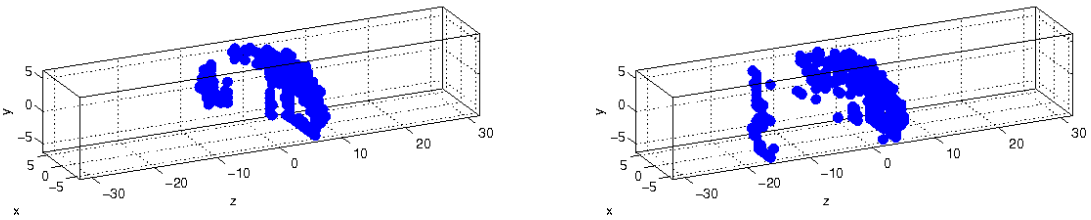
$t = 700$



$t = 800$

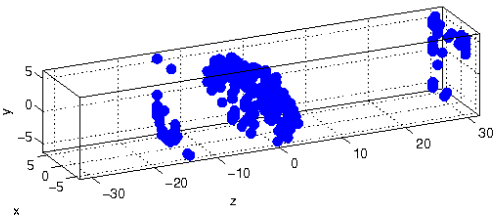




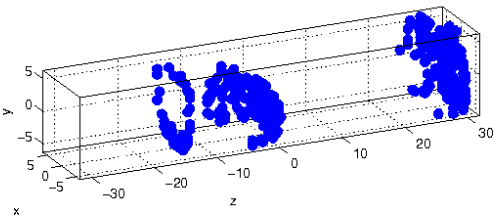


$t = 100$

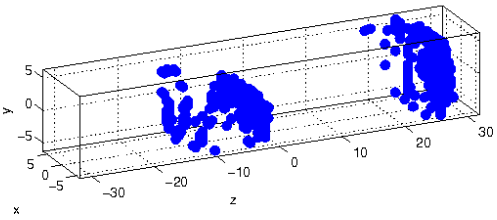
$t = 200$



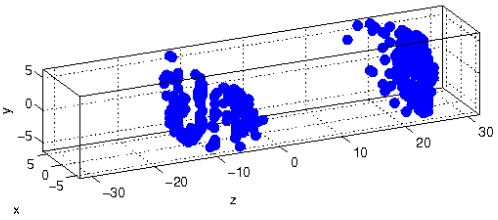
$t = 300$



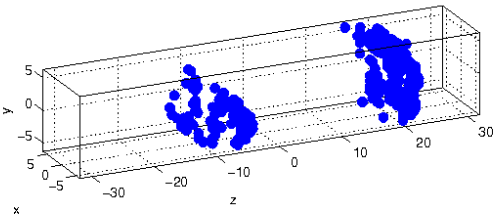
$t = 400$



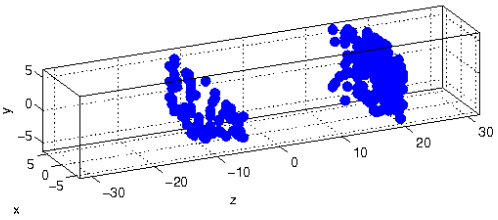
$t = 500$



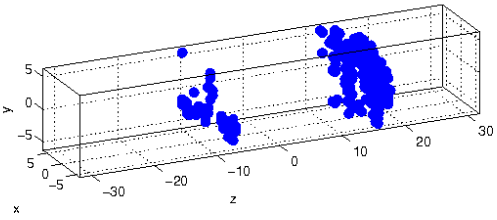
$t = 600$



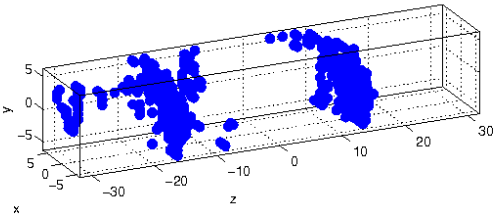
$t = 700$



$t = 800$

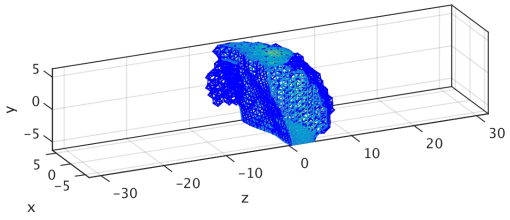


$t = 900$

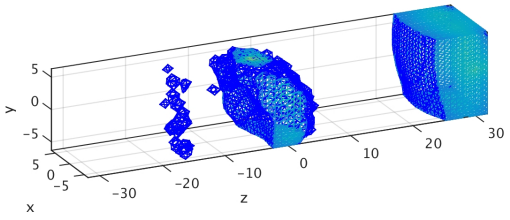


$t = 1{,}000$

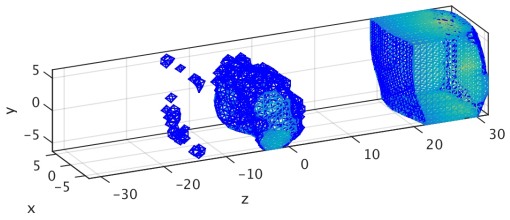
Figure 7: Open calcium release units throughout the cell using finite volume method with $\beta^{(i)} = (0, 0, 0.03), i = 1, \ldots, n_s$, on mesh size $32 \times 32 \times 128$.



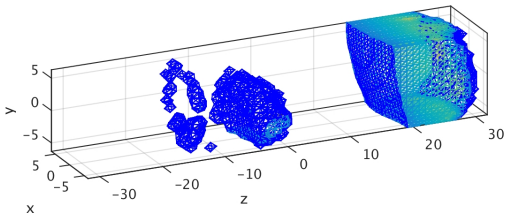
$t = 100$



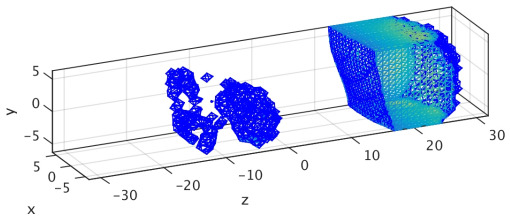
$t = 200$



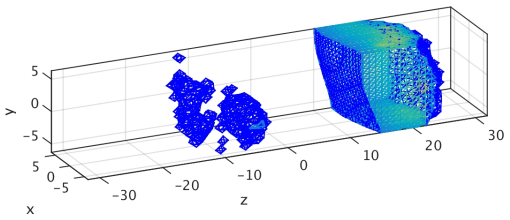
$t = 300$



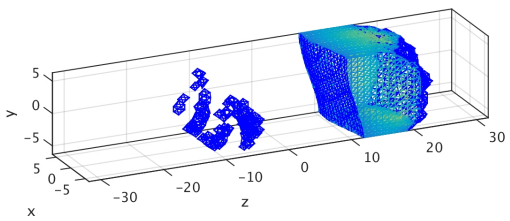
$t = 400$



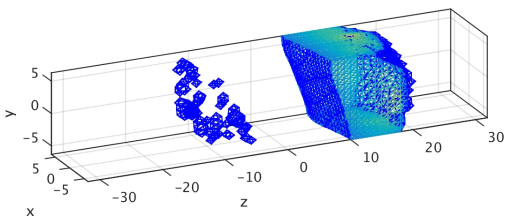
$t = 500$



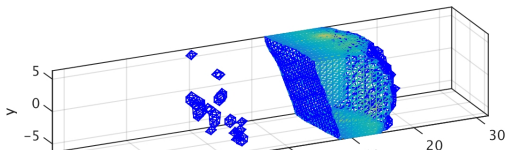
$t = 600$

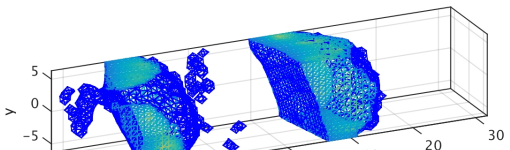


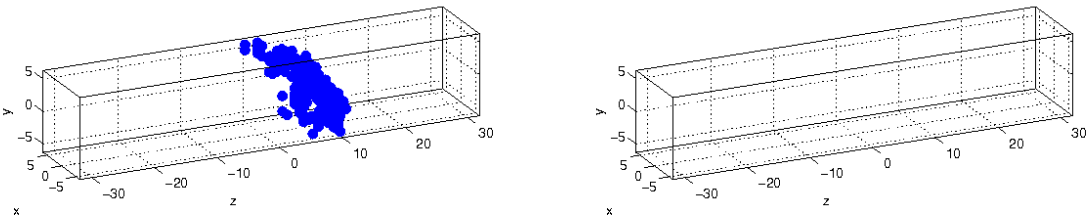
$t = 700$



$t = 800$

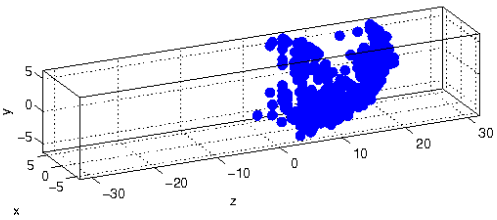




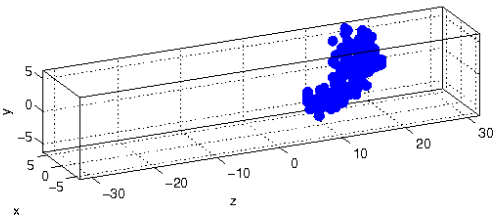


$t = 100$

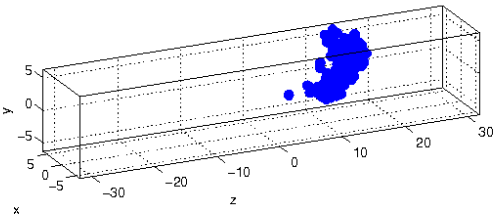
$t = 200$



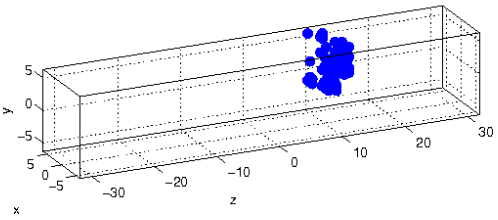
$t = 300$



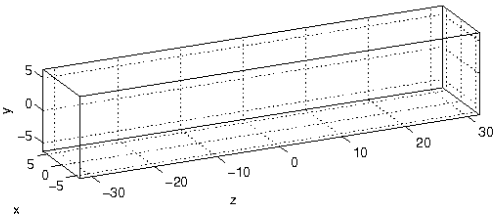
$t = 400$



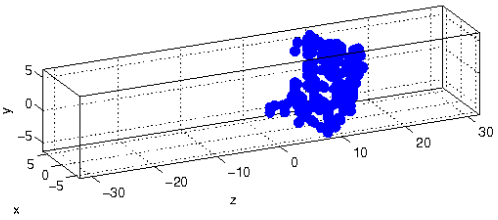
$t = 500$



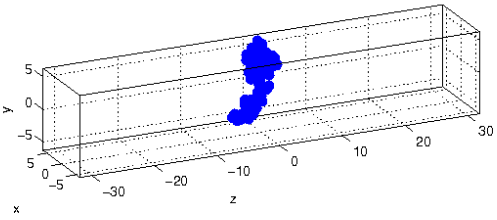
$t = 600$



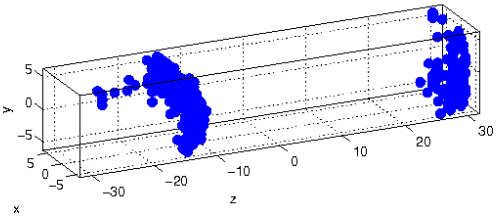
$t = 700$



$t = 800$

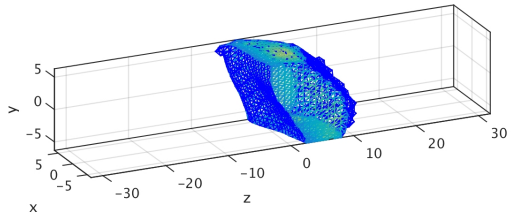


$t = 900$

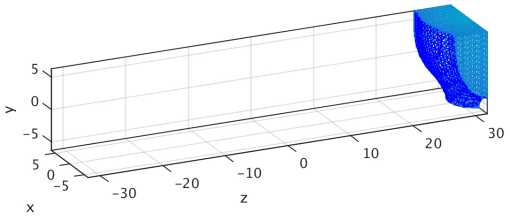


$t = 1{,}000$

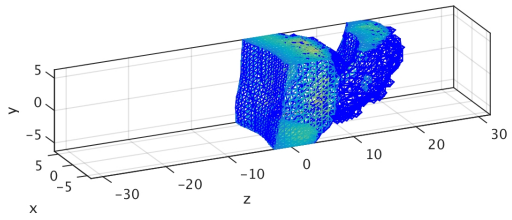
Figure 9: Open calcium release units throughout the cell using finite volume method with $\beta^{(i)} = (0, 0, 0.05), i = 1, \ldots, n_s$, on mesh size $32 \times 32 \times 128$.



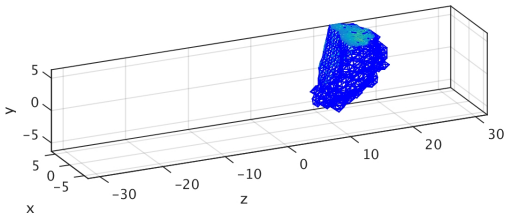
$t = 100$



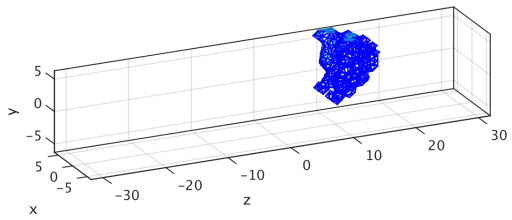
$t = 200$



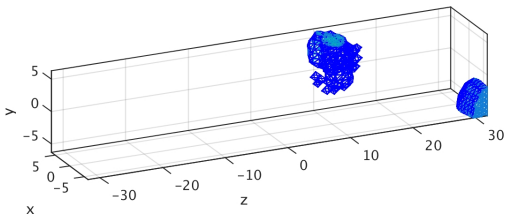
$t = 300$



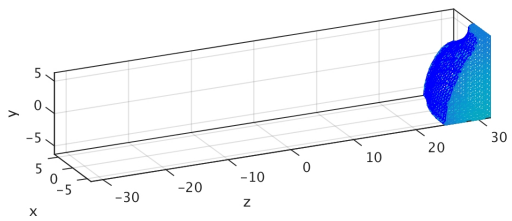
$t = 400$



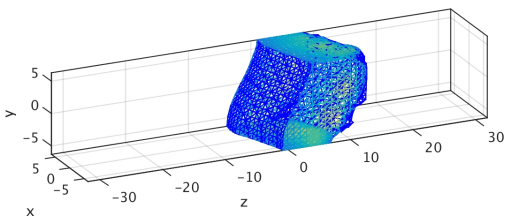
$t = 500$



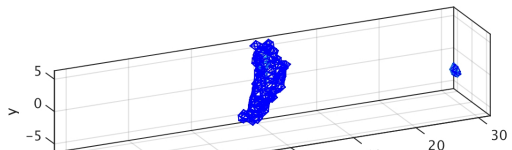
$t = 600$

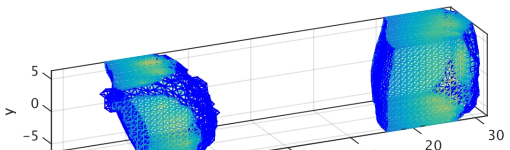


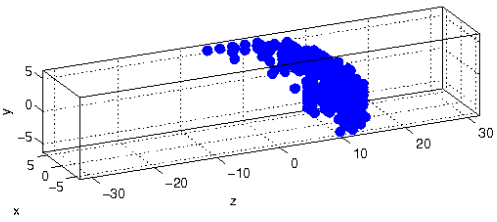
$t = 700$



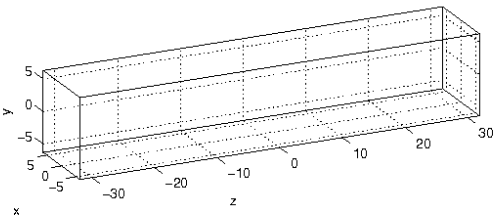
$t = 800$



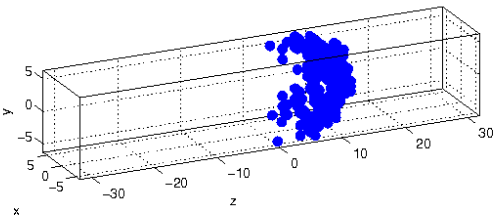




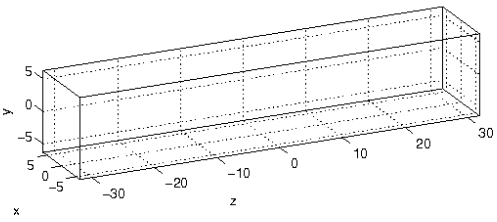
$t = 100$



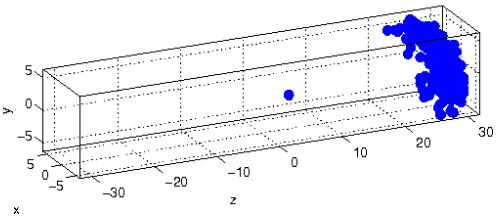
$t = 200$



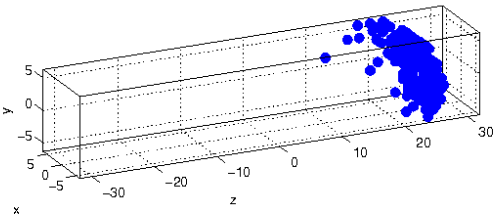
$t = 300$



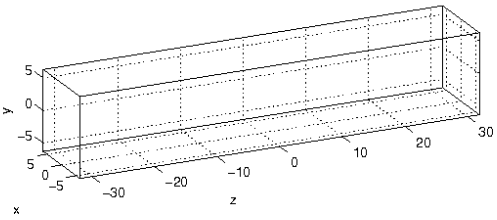
$t = 400$



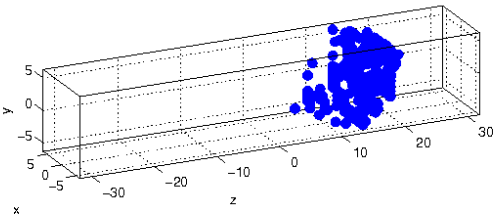
$t = 500$



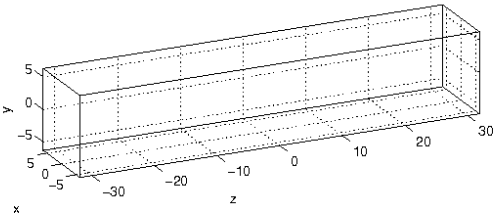
$t = 600$



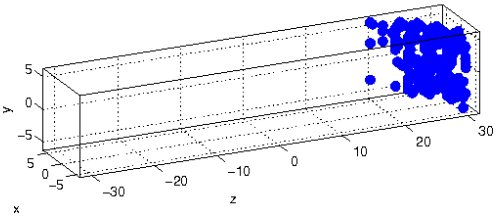
$t = 700$



$t = 800$

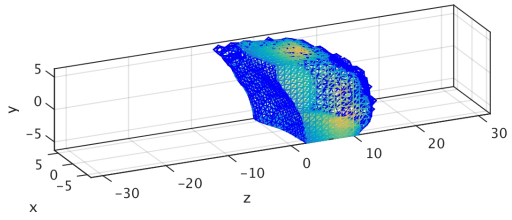


$t = 900$

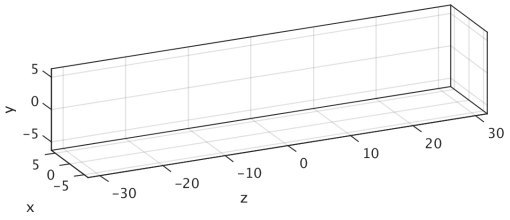


$t = 1{,}000$

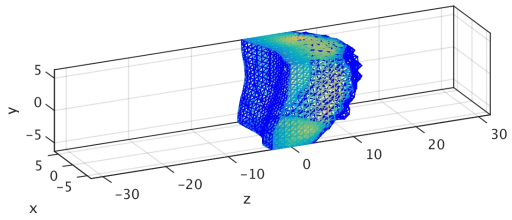
Figure 11: Open calcium release units throughout the cell using finite volume method with $\beta^{(i)} = (0, 0, 0.1), i = 1, \ldots, n_s$, on mesh size $32 \times 32 \times 128$.



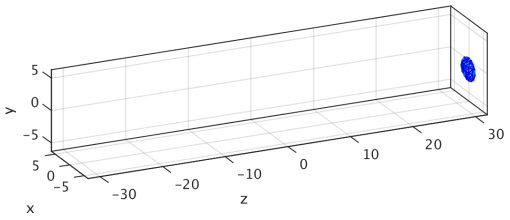
$t = 100$



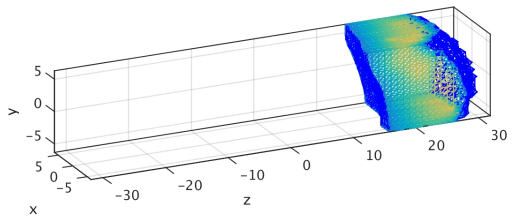
$t = 200$



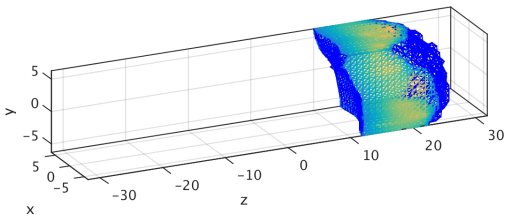
$t = 300$



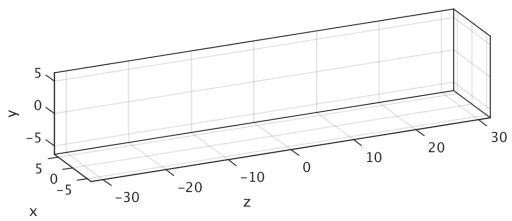
$t = 400$



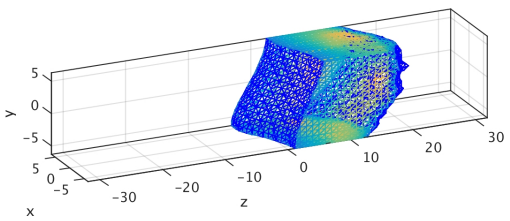
$t = 500$



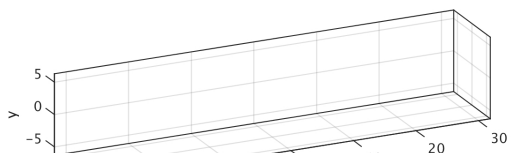
$t = 600$

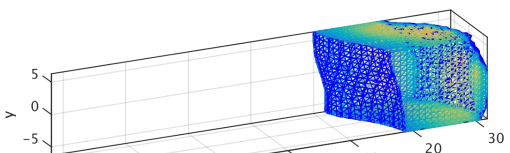


$t = 700$



$t = 800$





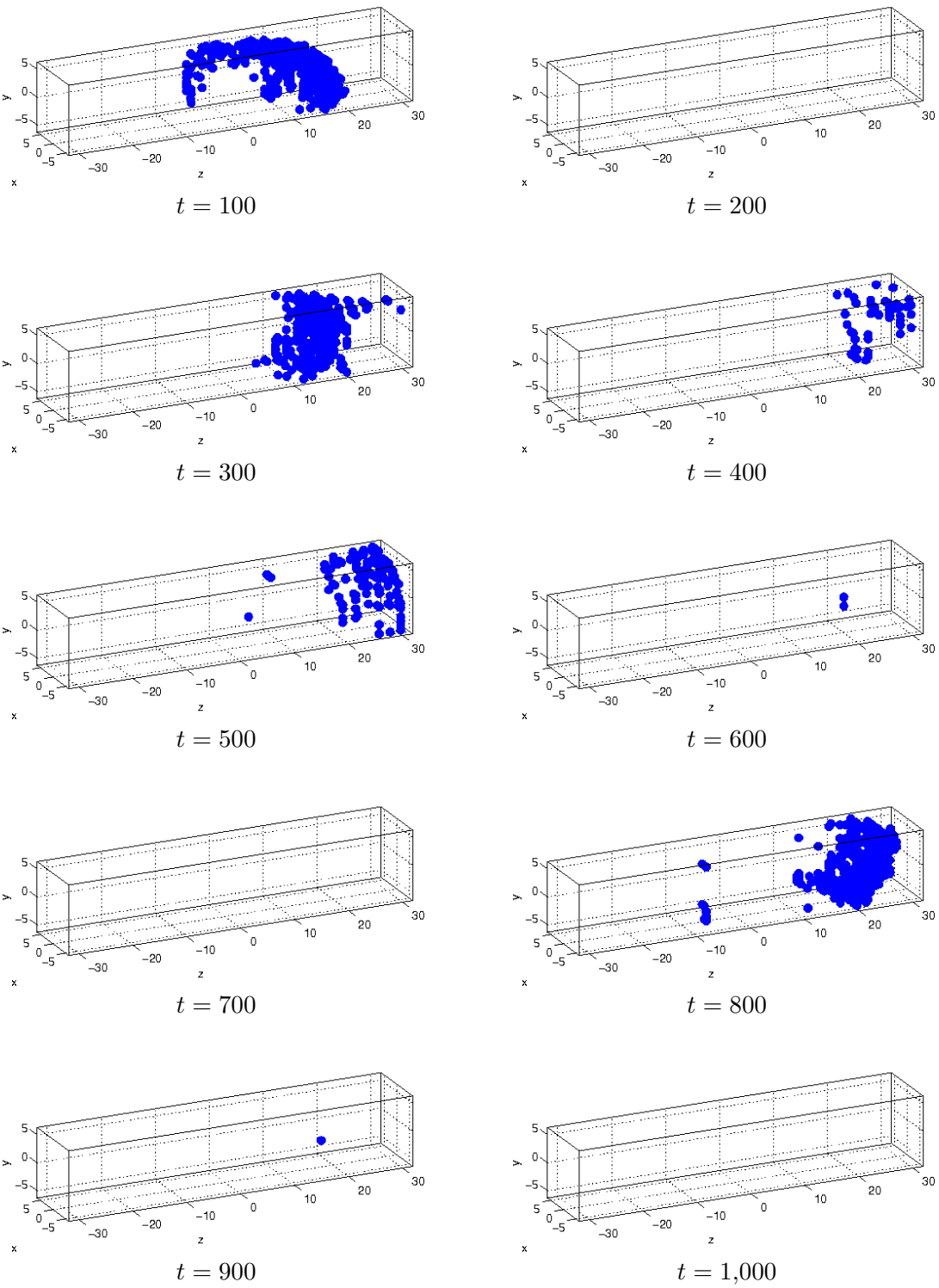
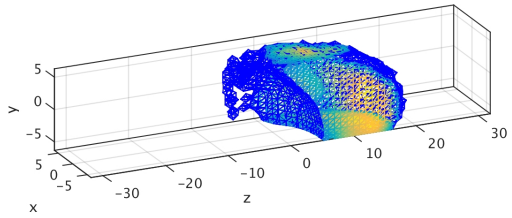
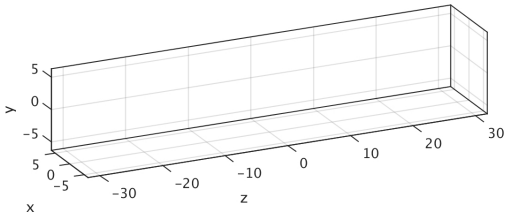


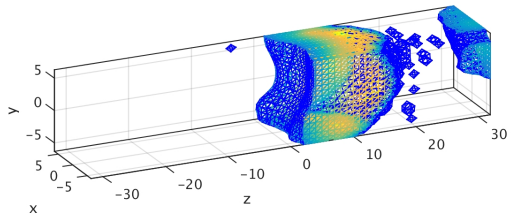
Figure 13: Open calcium release units throughout the cell using finite volume method with $\beta^{(i)} = (0, 0, 0.2), i = 1, \ldots, n_s$, on mesh size $32 \times 32 \times 128$.



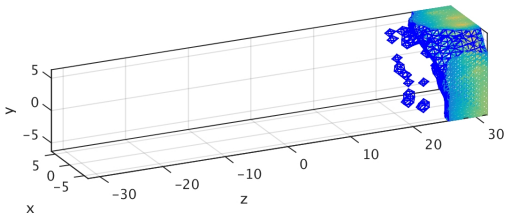
$t = 100$



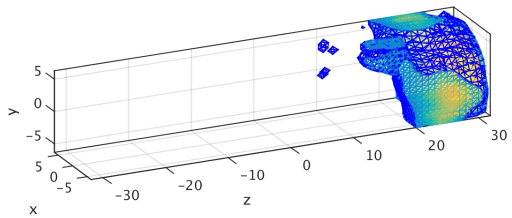
$t = 200$



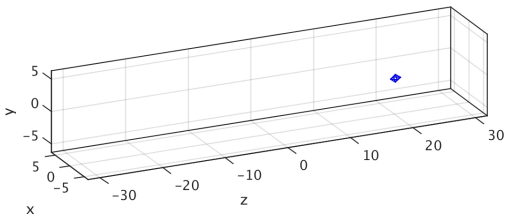
$t = 300$



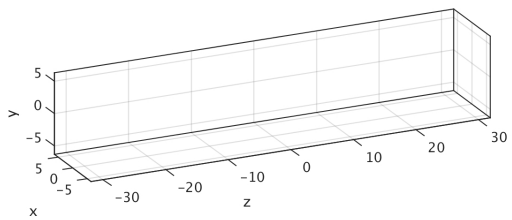
$t = 400$



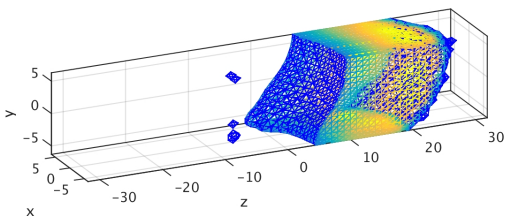
$t = 500$



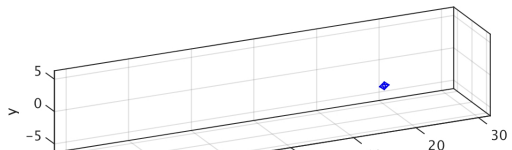
$t = 600$

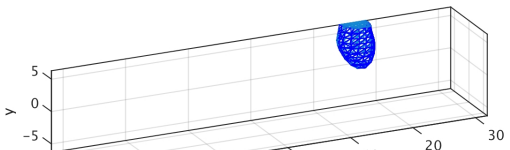


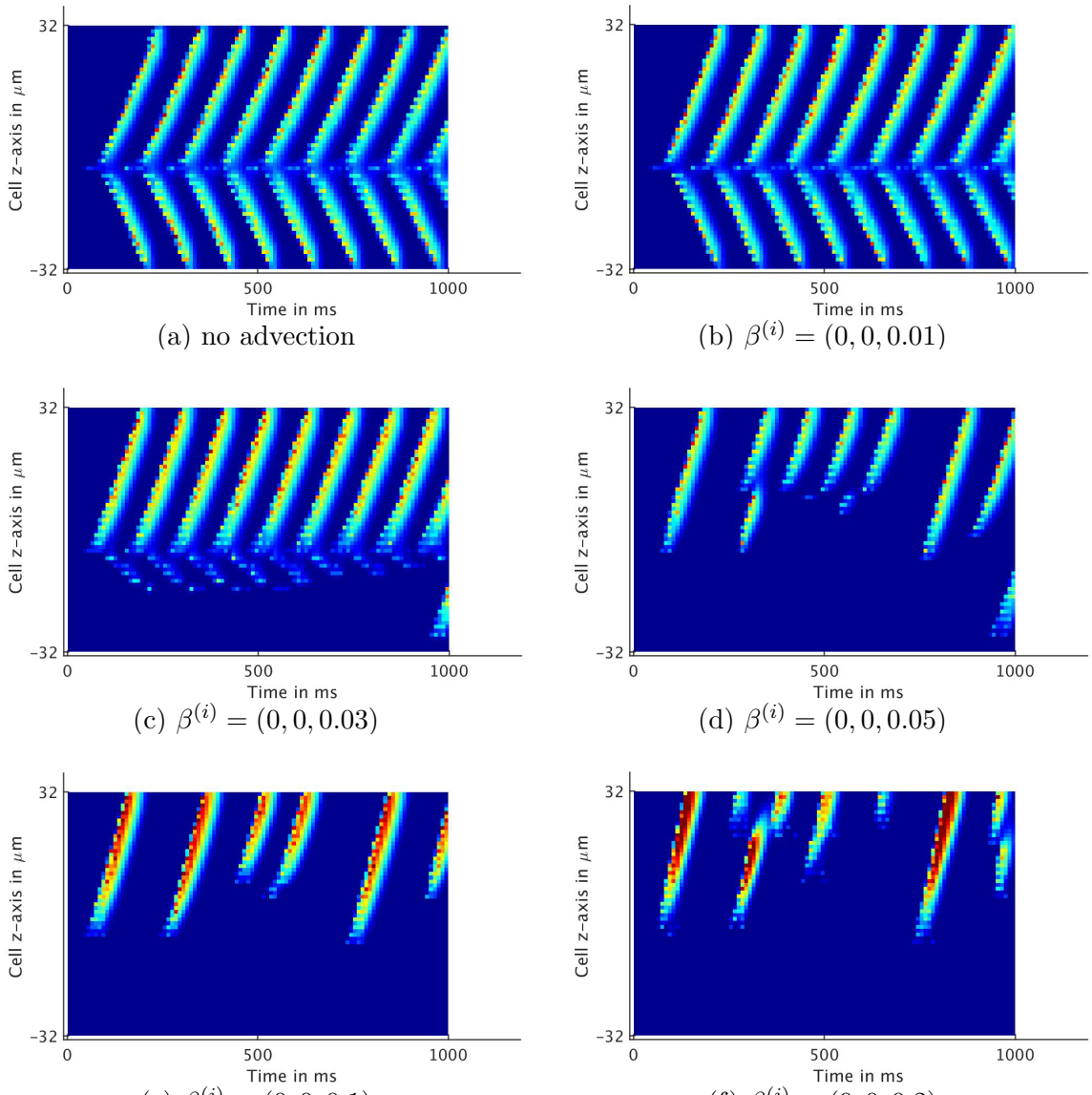
$t = 700$



$t = 800$





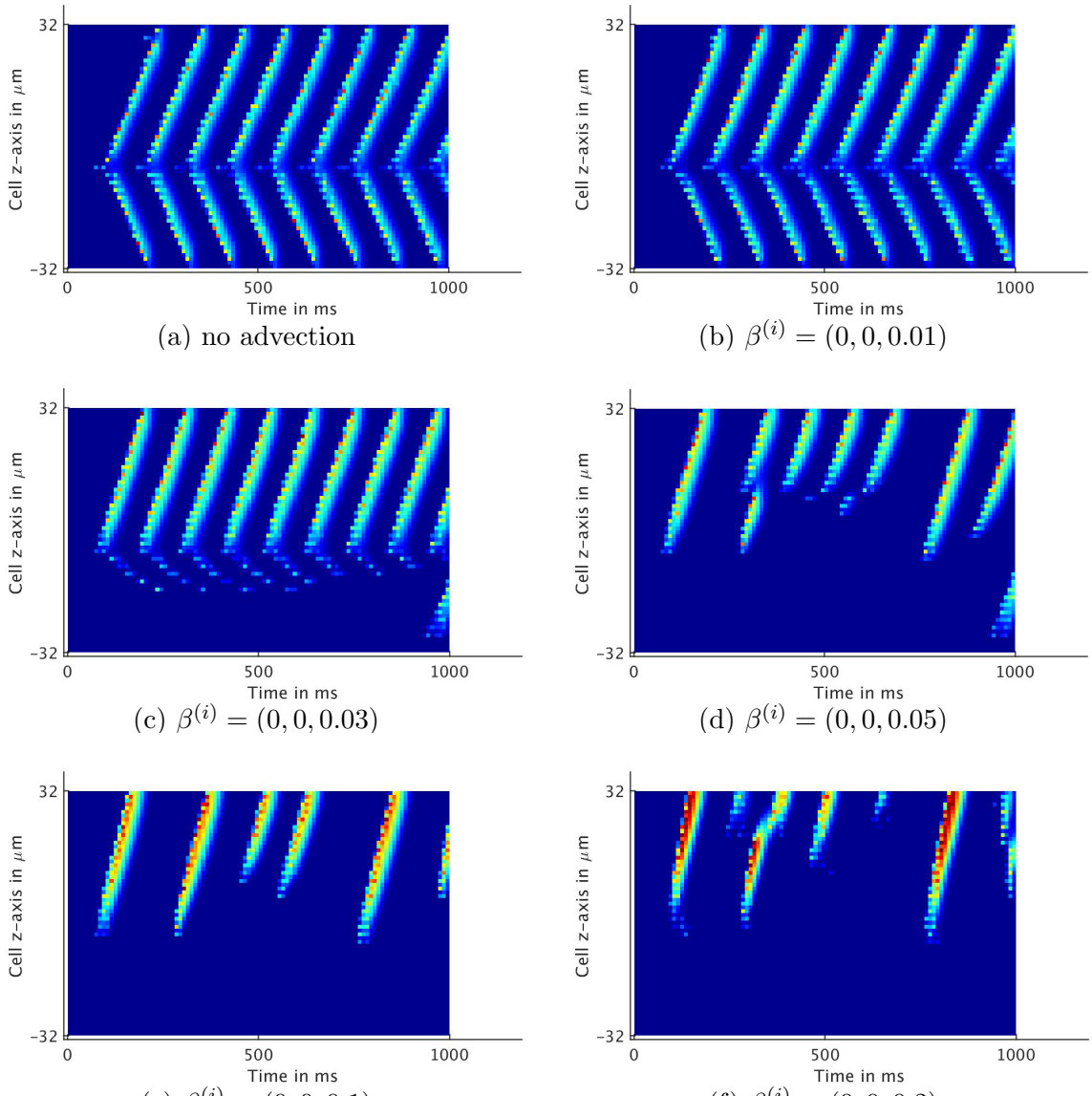


(a) no advection (b) $\beta^{(i)} = (0, 0, 0.01)$

(c) $\beta^{(i)} = (0, 0, 0.03)$ (d) $\beta^{(i)} = (0, 0, 0.05)$

(e) $\beta^{(i)} = (0, 0, 0.1)$ (f) $\beta^{(i)} = (0, 0, 0.2)$

Figure 15: Time evolution of the longitudinal line scan showing the calcium concentration along the line $x = y = 0$ μm, $0 \leq t \leq 1{,}000$ ms.



(a) no advection (b) $\beta^{(i)} = (0, 0, 0.01)$

(c) $\beta^{(i)} = (0, 0, 0.03)$ (d) $\beta^{(i)} = (0, 0, 0.05)$

(e) $\beta^{(i)} = (0, 0, 0.1)$ (f) $\beta^{(i)} = (0, 0, 0.2)$

Figure 16: Time evolution of the longitudinal line scan showing the calcium concentration along the line $x = y = 5.6$ μm, $0 \leq t \leq 1{,}000$ ms.

5 CONVERGENCE STUDIES

This section presents numerical studies of convergence order of the spatial discretization for the two scalar test problems of Section 2.2. If the true solution is denoted by u and its numerical approximation by u_h, then classical results for the spatial error in the L^2-norm have the form

$$\|u(\mathbf{x},t) - u_h(\mathbf{x},t)\|_{L^2(\Omega)} \leq C\, h^q, \quad \text{as } h \to 0, \tag{5.1}$$

for all $0 < t \leq t_{\text{fin}}$, where the constant C is independent of the mesh size h. The number q is the convergence order of the spatial discretization. Here, the $L^2(\Omega)$-norm is defined as

$$\|v\|_{L^2(\Omega)} = \left(\int_\Omega v^2 \, d\mathbf{x} \right)^{1/2}.$$

For the finite element method, the classical theory for linear elements specifies $q = 2$, which does not depend on space dimension, see, e.g., [15]. The classical theory requires the source terms to be in the function space $L^2(\Omega)$, which is not true for source terms involving point sources modeled by the Dirac delta distribution. For the finite element method, heuristic arguments and the computational results of [3] indicate that $q = 0.5$ in three spatial dimensions and $q = 1.0$ in two spatial dimensions. Motivated by these computational results, rigorous analysis in [11, Theorem 5.1] establishes that (5.1) holds with $q = 2 - d/2 - \varepsilon$ in dimensions $d = 2, 3$, which confirms the computational results.

For the finite volume method, we are not aware of any rigorous theory for problems involving non-smooth source terms. The purpose of this section is to analyze the convergence order of the finite volume method numerically, analogous to [3], and additionally (i) analyze the impact of increasing advection on the convergence order and (ii) compare the first- and second-order discretization of the advection term in (3.6) and (3.7), respectively.

The results in this section are collected in ten tables, with Tables 2 through 5 containing results for problems on the three-dimensional domain $\Omega \subset \mathbb{R}^3$, Tables 6 through 9 results for problems on the two-dimensional domain $\Omega \subset \mathbb{R}^2$, and Tables 10 and 11 providing summaries of the results. Specifically, we start with Tables 2 and 3 considering the same method as in [10], using the second-order discretization of the advection term (3.7). Then, Tables 4 and 5 provide the comparison to using the first-order discretization of the advection term (3.6). The two pairs of Tables 6 and 7 and Tables 8 and 9 repeat the comparison of second- and first-order advection discretizations for the problems in two dimensions.

Each of the Tables 2 through 9 contains five subtables. The entries in each subtable report the L^2-norm of the error $\|u(\cdot,t) - u_h(\cdot,t)\|_{L^2(\Omega)}$ and in parentheses a numerical estimate of the convergence order q from (**??**), for four progressively finer meshes. Given numerical solutions on two meshes with mesh widths h and $2h$, the order q can be estimated using the formula

$$q^{est} = \log_2 \left(\frac{\|u(\cdot,t) - u_{2h}(\cdot,t)\|_{L^2(\Omega)}}{\|u(\cdot,t) - u_h(\cdot,t)\|_{L^2(\Omega)}} \right). \tag{5.2}$$

For problems, whose true solution u is not available, it is customary to use the numerical solution on the finest available mesh as a reference solution in place of u in the norm of the errors. This procedure is necessary for the non-smooth test problem. For consistency, we apply this procedure also for the smooth test problem. Hence, all tables in this section show errors against a reference solution. However, for the smooth problem, we also obtained the analogous results of $\|u(\cdot,t) - u_h(\cdot,t)\|_{L^2(\Omega)}$ and q^{est} using the true solution and confirmed that the procedure with the reference solution worked correctly in this case.

First, let us discuss results in three spatial dimensions. For instance, Table 2 reports error and convergence order against reference solution using second-order discretization of the advection term. The first subtable 2 (a) reports the results for the finite element method. Since this method is not suitable for problems with advection, it is applied to the test problems without the advection term. The convergence order results for the finite element method agree with those in [3] and are included here to allow for a ready comparison. Recall from Section 2.2 and Table 1 that the scalar test equations use the calcium diffusion coefficient matrix $D = D^{(1)} = \text{diag}(0.15, 0.15, 0.30)$. The advection velocity is the product of the weight ω and vector $(0.15, 0.15, 0.30)^T$ of the form $\beta = \beta^{(1)} = \omega\,(0.15, 0.15, 0.30)^T$ such that we can control the magnitude of advection by varying the constant ω. For $\omega = 0$, there is no advection, and for $\omega = 1$, diffusion and advection are on the same order of magnitude. Table 2 (b)–(e) contain results for the finite volume method with second-order discretization of advection term described in (3.7), with $\omega = 0, 0.01, 0.1, 1$. We observe that both FEM and FVM with second-order advection discretization have a numerical convergence order of $q = 2$. Table 3 contains five subtables for the test problem with non-smooth source term. We observe a numerical convergence order of $q = 0.5$ for both FEM and FVM with second-order advection discretization. Table 4 and Table 5 repeat the tests above but use the first-order advection discretization as described in (3.6). Table 4 shows that as we increase the weight of advection ω, the numerical convergence order dropped from 2 to 1.5, indicating the convergence rate is lower when using first-order advection discretization for the test problem with smooth source term. From Table 5 we observe the numerical convergence order of $q = 0.5$ for both FEM and FVM with first-order advection discretization.

Notice that in Table 3 and Table 5 the actual observed errors have a magnitude of 10^{+1}. This is because the calcium release unit at the center of the domain is modeled by Dirac delta distribution, which is a highly non-smooth source term. Note that the apparently large values for the error must be viewed in the context of the large size of the domain Ω, which is $(12.8)(12.8)(64) = 10,\,485.76$. In the meantime, convergence studies in [3] have shown that the L^2-norms have the magnitude from 10^{-1} to 10^{-3} and converge quadratically on the domain Ω with a small area centered about the calcium release unit removed. This confirms that the non-smooth source term is the reason of observed convergence order $q = 0.5$.

Now, let us move on to results in two spatial dimensions. Table 6 and Table 7 show that with second-order discretization of the advecton term, the numerical convergence order $q = 2$ for finite volume method with smooth source term, and $q = 1$ with non-smooth source term. Table 8 and Table 9 show that with first-order discretization of advecton term, q drops below 2 while increasing the weight of advection with smooth source term, and $q = 1$ with non-smooth source term. This shows that the convergence order is dependent on the space dimensions while dealing with highly non-smooth source terms such as point sources here.

Table 2: L^2-error against reference solution (and estimated convergence order q^{est}) for scalar test problem with smooth source term in 3-D using a second-order advection discretization in the finite volume method.

(a) FEM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	2.4322e–01	2.1725e–01	1.9453e–01
$32 \times 32 \times 128$	6.0327e–02 (2.0114)	5.3869e–02 (2.0119)	4.8182e–02 (2.0134)
$64 \times 64 \times 256$	1.4488e–02 (2.0579)	1.2954e–02 (2.0561)	1.1598e–02 (2.0546)
$128 \times 128 \times 512$	3.0194e–03 (2.2626)	2.7163e–03 (2.2536)	2.4465e–03 (2.2450)
(b) 2nd order FVM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.1596e–01	4.0106e–01	3.8575e–01
$32 \times 32 \times 128$	1.0682e–01 (1.9612)	1.0245e–01 (1.9689)	9.8285e–02 (1.9726)
$64 \times 64 \times 256$	2.6685e–02 (2.0011)	2.5573e–02 (2.0022)	2.4520e–02 (2.0030)
$128 \times 128 \times 512$	6.9077e–03 (1.9498)	6.6055e–03 (1.9529)	6.3208e–03 (1.9558)
(c) 2nd order FVM with $\omega = 0.01$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.1588e–01	4.0096e–01	3.8564e–01
$32 \times 32 \times 128$	1.0683e–01 (1.9608)	1.0246e–01 (1.9684)	9.8294e–02 (1.9721)
$64 \times 64 \times 256$	2.6687e–02 (2.0011)	2.5574e–02 (2.0023)	2.4520e–02 (2.0031)
$128 \times 128 \times 512$	6.9076e–03 (1.9499)	6.6050e–03 (1.9530)	6.3202e–03 (1.9559)
(d) 2nd order FVM with $\omega = 0.1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.1551e–01	4.0076e–01	3.8570e–01
$32 \times 32 \times 128$	1.0699e–01 (1.9574)	1.0273e–01 (1.9639)	9.8654e–02 (1.9670)
$64 \times 64 \times 256$	2.6726e–02 (2.0011)	2.5638e–02 (2.0025)	2.4601e–02 (2.0037)
$128 \times 128 \times 512$	6.9112e–03 (1.9512)	6.6163e–03 (1.9542)	6.3338e–03 (1.9576)
(e) 2nd order FVM with $\omega = 1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.3888e–01	4.5043e–01	4.6647e–01
$32 \times 32 \times 128$	1.1749e–01 (1.9013)	1.2257e–01 (1.8777)	1.2900e–01 (1.8544)
$64 \times 64 \times 256$	2.9760e–02 (1.9811)	3.1218e–02 (1.9732)	3.3070e–02 (1.9638)
$128 \times 128 \times 512$	7.4323e–03 (2.0015)	7.5946e–03 (2.0393)	7.8554e–03 (2.0738)

Table 3: L^2-error against reference solution (and estimated convergence order q^{est}) for scalar test problem with non-smooth source term in 3-D using a second-order advection discretization in the finite volume method.

(a) FEM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.4249e+01	5.0676e+01	5.0706e+01
$32 \times 32 \times 128$	3.6689e+01 (0.5643)	3.7651e+01 (0.4286)	3.7806e+01 (0.4235)
$64 \times 64 \times 256$	2.6378e+01 (0.4760)	2.6396e+01 (0.5124)	2.6400e+01 (0.5181)
$128 \times 128 \times 512$	1.6083e+01 (0.7138)	1.6083e+01 (0.7147)	1.6084e+01 (0.7149)
(b) 2nd order FVM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	7.6451e+01	8.5493e+01	8.8933e+01
$32 \times 32 \times 128$	6.4231e+01 (0.2513)	6.5344e+01 (0.3877)	6.5578e+01 (0.4395)
$64 \times 64 \times 256$	4.6063e+01 (0.4797)	4.6135e+01 (0.5022)	4.6152e+01 (0.5068)
$128 \times 128 \times 512$	3.0898e+01 (0.5761)	3.0903e+01 (0.5781)	3.0905e+01 (0.5786)
(c) 2nd order FVM with $\omega = 0.01$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	7.6263e+01	8.5145e+01	8.8515e+01
$32 \times 32 \times 128$	6.4070e+01 (0.2513)	6.5172e+01 (0.3857)	6.5405e+01 (0.4365)
$64 \times 64 \times 256$	4.5999e+01 (0.4781)	4.6070e+01 (0.5004)	4.6087e+01 (0.5050)
$128 \times 128 \times 512$	3.0875e+01 (0.5751)	3.0880e+01 (0.5772)	3.0881e+01 (0.5776)
(d) 2nd order FVM with $\omega = 0.1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	7.4665e+01	8.2230e+01	8.5029e+01
$32 \times 32 \times 128$	6.2670e+01 (0.2527)	6.3683e+01 (0.3687)	6.3902e+01 (0.4121)
$64 \times 64 \times 256$	4.5425e+01 (0.4643)	4.5495e+01 (0.4852)	4.5512e+01 (0.4896)
$128 \times 128 \times 512$	3.0667e+01 (0.5668)	3.0672e+01 (0.5688)	3.0673e+01 (0.5693)
(e) 2nd order FVM with $\omega = 1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	6.5034e+01	6.6909e+01	6.7530e+01
$32 \times 32 \times 128$	5.2889e+01 (0.2982)	5.3328e+01 (0.3273)	5.3437e+01 (0.3377)
$64 \times 64 \times 256$	4.0584e+01 (0.3821)	4.0636e+01 (0.3921)	4.0648e+01 (0.3947)
$128 \times 128 \times 512$	2.8747e+01 (0.4975)	2.8751e+01 (0.4991)	2.8752e+01 (0.4995)

Table 4: L^2-error against reference solution (and estimated convergence order q^{est}) for scalar test problem with smooth source term in 3-D using a first-order advection discretization in the finite volume method.

(a) FEM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	2.4322e–01	2.1725e–01	1.9453e–01
$32 \times 32 \times 128$	6.0327e–02 (2.0114)	5.3869e–02 (2.0119)	4.8182e–02 (2.0134)
$64 \times 64 \times 256$	1.4488e–02 (2.0579)	1.2954e–02 (2.0561)	1.1598e–02 (2.0546)
$128 \times 128 \times 512$	3.0194e–03 (2.2626)	2.7163e–03 (2.2536)	2.4465e–03 (2.2450)
(b) 1st order FVM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.1596e–01	4.0106-e-01	3.8575e–01
$32 \times 32 \times 128$	1.0682e–01 (1.9612)	1.0245e–01 (1.9689)	9.8285e–02 (1.9726)
$64 \times 64 \times 256$	2.6685e–02 (2.0011)	2.5573e–02 (2.0022)	2.4520e–02 (2.0030)
$128 \times 128 \times 512$	6.9077e–03 (1.9498)	6.6055e–03 (1.9529)	6.3208e–03 (1.9558)
(c) 1st order FVM with $\omega = 0.01$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.1399e–01	3.9820e–01	3.8205e–01
$32 \times 32 \times 128$	1.0598e—01 (1.9658)	1.0122e–01 (1.9761)	9.6683e–02 (1.9824)
$64 \times 64 \times 256$	2.6350e–02 (2.0079)	2.5085e–02 (2.0126)	2.3890e–02 (2.0168)
$128 \times 128 \times 512$	6.8263e–03 (1.9486)	6.4894e–03 (1.9507)	6.1736e–03 (1.9522)
(d) 1st order FVM with $\omega = 0.1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	3.9932e–01	3.7901e–01	3.5979e–01
$32 \times 32 \times 128$	1.0109e–01 (1.9819)	9.5928e–02 (1.9822)	9.2084e–02 (1.9661)
$64 \times 64 \times 256$	2.5389e–02 (1.9934)	2.5096e–02 (1.9345)	2.5552e–02 (1.8495)
$128 \times 128 \times 512$	6.9876e–03 (1.8613)	7.2888e–03 (1.7837)	7.8148e–03 (1.7091)
(e) 1st order FVM with $\omega = 1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.4935e–01	7.1330e–01	8.7562e–01
$32 \times 32 \times 128$	2.4039e–01 (1.1923)	3.3733e–01 (1.0804)	4.2579e–01 (1.0401)
$64 \times 64 \times 256$	1.0429e–01 (1.2047)	1.4883e–01 (1.1805)	1.8866e–01 (1.1743)
$128 \times 128 \times 512$	3.5657e–02 (1.5484)	5.0768e–02 (1.5517)	6.4249e–02 (1.5541)

Table 5: L^2-error against reference solution (and estimated convergence order q^{est}) for scalar test problem with non-smooth source term in 3-D using a first-order advection discretization in the finite volume method.

(a) FEM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.4249e+01	5.0676e+01	5.0706e+01
$32 \times 32 \times 128$	3.6689e+01 (0.5643)	3.7651e+01 (0.4286)	3.7806e+01 (0.4235)
$64 \times 64 \times 256$	2.6378e+01 (0.4760)	2.6396e+01 (0.5124)	2.6400e+01 (0.5181)
$128 \times 128 \times 512$	1.6083e+01 (0.7138)	1.6083e+01 (0.7147)	1.6084e+01 (0.7149)
(b) 1st order FVM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	7.6451e+01	8.5493e+01	8.8933e+01
$32 \times 32 \times 128$	6.4231e+01 (0.2513)	6.5344e+01 (0.3877)	6.5578e+01 (0.4395)
$64 \times 64 \times 256$	4.6063e+01 (0.4797)	4.6135e+01 (0.5022)	4.6152e+01 (0.5068)
$128 \times 128 \times 512$	3.0898e+01 (0.5761)	3.0903e+01 (0.5781)	3.0905e+01 (0.5786)
(c) 1st order FVM with $\omega = 0.01$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	7.6299e+01	8.5218e+01	8.8596e+01
$32 \times 32 \times 128$	6.4101e+01 (0.2513)	6.5198e+01 (0.3863)	6.5426e+01 (0.4374)
$64 \times 64 \times 256$	4.6010e+01 (0.4784)	4.6080e+01 (0.5007)	4.6097e+01 (0.5052)
$128 \times 128 \times 512$	3.0881e+01 (0.5752)	3.0886e+01 (0.5772)	3.0887e+01 (0.5777)
(d) 1st order FVM with $\omega = 0.1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	7.4977e+01	8.2850e+01	8.5714e+01
$32 \times 32 \times 128$	6.2953e+01 (0.2522)	6.3918e+01 (0.3743)	6.4105e+01 (0.4191)
$64 \times 64 \times 256$	4.5538e+01 (0.4672)	4.5596e+01 (0.4873)	4.5609e+01 (0.4911)
$128 \times 128 \times 512$	3.0725e+01 (0.5676)	3.0729e+01 (0.5693)	3.0730e+01 (0.5697)
(e) 1st order FVM with $\omega = 1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	6.5199e+01	6.7564e+01	6.8367e+01
$32 \times 32 \times 128$	5.4115e+01 (0.2688)	5.4522e+01 (0.3094)	5.4708e+01 (0.3216)
$64 \times 64 \times 256$	4.1512e+01 (0.3825)	4.1587e+01 (0.3907)	4.1646e+01 (0.3936)
$128 \times 128 \times 512$	2.9290e+01 (0.5031)	2.9304e+01 (0.5050)	2.9316e+01 (0.5065)

Table 6: L^2-error against reference solution (and estimated convergence order q^{est}) for scalar test problem with smooth source term in 2-D using a second-order advection discretization in the finite volume method.

(a) FEM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.5341e–01	1.4291e–01	1.3300e–01
$32 \times 32 \times 128$	3.7920e–02 (2.0164)	3.5316e–02 (2.0167)	3.2857e–02 (2.0172)
$64 \times 64 \times 256$	8.9647e–03 (2.0806)	8.3605e–03 (2.0787)	7.7914e–03 (2.0762)
$128 \times 128 \times 512$	1.8359e–03 (2.2878)	1.7270e–03 (2.2753)	1.6268e–03 (2.2598)
(b) 2nd order FVM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.3268e–01	1.2941e–01	1.2582e–01
$32 \times 32 \times 128$	3.4128e–02 (1.9589)	3.3101e–02 (1.9671)	3.2096e–02 (1.9709)
$64 \times 64 \times 256$	8.5443e–03 (1.9979)	8.2794e–03 (1.9993)	8.0214e–03 (2.0005)
$128 \times 128 \times 512$	2.2398e–03 (1.9316)	2.1654e–03 (1.9349)	2.0924e–03 (1.9387)
(c) 2nd order FVM with $\omega = 0.01$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.3264e–01	1.2937e–01	1.2576e–01
$32 \times 32 \times 128$	3.4126e–02 (1.9586)	3.3097e–02 (1.9667)	3.2088e–02 (1.9706)
$64 \times 64 \times 256$	8.5434e–03 (1.9980)	8.2774e–03 (1.9994)	8.0183e–03 (2.0007)
$128 \times 128 \times 512$	2.2395e–03 (1.9317)	2.1649e–03 (1.9349)	2.0916e–03 (1.9387)
(d) 2nd order FVM with $\omega = 0.1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.3246e–01	1.2917e–01	1.2559e–01
$32 \times 32 \times 128$	3.4144e–02 (1.9558)	3.3128e–02 (1.9632)	3.2125e–02 (1.9670)
$64 \times 64 \times 256$	8.5449e–03 (1.9985)	8.2783e–03 (2.0006)	8.0182e–03 (2.0023)
$128 \times 128 \times 512$	2.2385e–03 (1.9326)	2.1635e–03 (1.9360)	2.0900e–03 (1.9398)
(e) 2nd order FVM with $\omega = 1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.4105e–01	1.4700e–01	1.5385e–01
$32 \times 32 \times 128$	3.7675e–02 (1.9046)	3.9556e–02 (1.8938)	4.1453e–02 (1.8920)
$64 \times 64 \times 256$	9.4614e–03 (1.9935)	9.8093e–03 (2.0117)	1.0087e–02 (2.0390)
$128 \times 128 \times 512$	2.3737e–03 (1.9949)	2.3841e–03 (2.0407)	2.4068e–03 (2.0673)

Table 7: L^2-error against reference solution (and estimated convergence order q^{est}) for scalar test problem with non-smooth source term in 2-D using a second-order advection discretization in the finite volume method.

(a) FEM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	6.4814e+01	5.6294e+01	5.4132e+01
$32 \times 32 \times 128$	3.2145e+01 (1.0117)	3.1586e+01 (0.8337)	3.1433e+01 (0.7842)
$64 \times 64 \times 256$	1.9717e+01 (0.7051)	1.9524e+01 (0.6941)	1.9461e+01 (0.6917)
$128 \times 128 \times 512$	9.4765e+00 (1.0570)	9.4550e+00 (1.0461)	9.4477e+00 (1.0425)
(b) 2nd order FVM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.2421e+01	6.5340e+01	7.1232e+01
$32 \times 32 \times 128$	3.7185e+01 (0.4954)	3.9076e+01 (0.7417)	3.9606e+01 (0.8468)
$64 \times 64 \times 256$	1.9847e+01 (0.9058)	2.0017e+01 (0.9650)	2.0073e+01 (0.9805)
$128 \times 128 \times 512$	9.6699e+00 (1.0373)	9.6869e+00 (1.0471)	9.6926e+00 (1.0503)
(c) 2nd order FVM with $\omega = 0.01$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.2229e+01	6.4951e+01	7.0739e+01
$32 \times 32 \times 128$	3.7048e+01 (0.4955)	3.8925e+01 (0.7387)	3.9452e+01 (0.8424)
$64 \times 64 \times 256$	1.9808e+01 (0.9033)	1.9978e+01 (0.9623)	2.0033e+01 (0.9777)
$128 \times 128 \times 512$	9.6608e+00 (1.0358)	9.6779e+00 (1.0456)	9.6835e+00 (1.0488)
(d) 2nd order FVM with $\omega = 0.1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.0582e+01	6.1664e+01	6.6600e+01
$32 \times 32 \times 128$	3.5859e+01 (0.4963)	3.7610e+01 (0.7133)	3.8115e+01 (0.8052)
$64 \times 64 \times 256$	1.9462e+01 (0.8817)	1.9630e+01 (0.9381)	1.9685e+01 (0.9533)
$128 \times 128 \times 512$	9.5802e+00 (1.0225)	9.5973e+00 (1.0324)	9.6029e+00 (1.0355)
(e) 2nd order FVM with $\omega = 1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.0331e+01	4.3925e+01	4.5555e+01
$32 \times 32 \times 128$	2.7520e+01 (0.5514)	2.8498e+01 (0.6242)	2.8846e+01 (0.6592)
$64 \times 64 \times 256$	1.6563e+01 (0.7326)	1.6710e+01 (0.7702)	1.6760e+01 (0.7834)
$128 \times 128 \times 512$	8.8417e+00 (0.9055)	8.8579e+00 (0.9156)	8.8632e+00 (0.9191)

Table 8: L^2-error against reference solution (and estimated convergence order q^{est}) for scalar test problem with smooth source term in 2-D using a first-order advection discretization in the finite volume method.

(a) FEM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.5341e–01	1.4291e–01	1.3300e–01
$32 \times 32 \times 128$	3.7920e–02 (2.0164)	3.5316e–02 (2.0167)	3.2857e–02 (2.0172)
$64 \times 64 \times 256$	8.9647e–03 (2.0806)	8.3605e–03 (2.0787)	7.7914e–03 (2.0762)
$128 \times 128 \times 512$	1.8359e–03 (2.2878)	1.7270e–03 (2.2753)	1.6268e–03 (2.2598)
(b) 1st order FVM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.3268e–01	1.2941e–01	1.2582e–01
$32 \times 32 \times 128$	3.4128e–02 (1.9589)	3.3101e–02 (1.9671)	3.2096e–02 (1.9709)
$64 \times 64 \times 256$	8.5443e–03 (1.9979)	8.2794e–03 (1.9993)	8.0214e–03 (2.0005)
$128 \times 128 \times 512$	2.2398e–03 (1.9316)	2.1654e–03 (1.9349)	2.0924e–03 (1.9387)
(c) 1st order FVM with $\omega = 0.01$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.3216e–01	1.2864e–01	1.2481e–01
$32 \times 32 \times 128$	3.3906e–02 (1.9626)	3.2769e–02 (1.9729)	3.1656e–02 (1.9792)
$64 \times 64 \times 256$	8.4566e–03 (2.0034)	8.1492e–03 (2.0076)	7.8502e–03 (2.0117)
$128 \times 128 \times 512$	2.2192e–03 (1.9300)	2.1353e–03 (1.9322)	2.0539e–03 (1.9344)
(d) 1st order FVM with $\omega = 0.1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.2828e–01	1.2350e–01	1.1877e–01
$32 \times 32 \times 128$	3.2633e–02 (1.9749)	3.1378e–02 (1.9767)	3.0445e–02 (1.9638)
$64 \times 64 \times 256$	8.2180e–03 (1.9895)	8.1770e–03 (1.9401)	8.3483e–03 (1.8667)
$128 \times 128 \times 512$	2.2688e–03 (1.8569)	2.3632e–03 (1.7908)	2.5289e–03 (1.7229)
(e) 1st order FVM with $\omega = 1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	1.6987e–01	2.1957e–01	2.7156e–01
$32 \times 32 \times 128$	7.1933e–02 (1.2397)	1.0169e–01 (1.1105)	1.3014e–01 (1.0612)
$64 \times 64 \times 256$	3.0802e–02 (1.2236)	4.4549e–02 (1.1908)	5.7368e–02 (1.1817)
$128 \times 128 \times 512$	1.0502e–02 (1.5523)	1.5167e–02 (1.5544)	1.9505e–02 (1.5564)

Table 9: L^2-error against reference solution (and estimated convergence order q^{est}) for scalar test problem with non-smooth source term in 2-D using a first-order advection discretization in the finite volume method.

(a) FEM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	6.4814e+01	5.6294e+01	5.4132e+01
$32 \times 32 \times 128$	3.2145e+01 (1.0117)	3.1586e+01 (0.8337)	3.1433e+01 (0.7842)
$64 \times 64 \times 256$	1.9717e+01 (0.7051)	1.9524e+01 (0.6941)	1.9461e+01 (0.6917)
$128 \times 128 \times 512$	9.4765e+00 (1.0570)	9.4550e+00 (1.0461)	9.4477e+00 (1.0425)
(b) 1st order FVM with $\omega = 0$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.2421e+01	6.5340e+01	7.1232e+01
$32 \times 32 \times 128$	3.7185e+01 (0.4954)	3.9076e+01 (0.7417)	3.9606e+01 (0.8468)
$64 \times 64 \times 256$	1.9847e+01 (0.9058)	2.0017e+01 (0.9650)	2.0073e+01 (0.9805)
$128 \times 128 \times 512$	9.6699e+00 (1.0373)	9.6869e+00 (1.0471)	9.6926e+00 (1.0503)
(c) 1st order FVM with $\omega = 0.01$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.2269e+01	6.5038e+01	7.0839e+01
$32 \times 32 \times 128$	3.7075e+01 (0.4955)	3.8942e+01 (0.7400)	3.9460e+01 (0.8441)
$64 \times 64 \times 256$	1.9811e+01 (0.9041)	1.9979e+01 (0.9628)	2.0032e+01 (0.9781)
$128 \times 128 \times 512$	9.6619e+00 (1.0360)	9.6785e+00 (1.0456)	9.6839e+00 (1.0487)
(d) 1st order FVM with $\omega = 0.1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	5.0941e+01	6.2423e+01	6.7459e+01
$32 \times 32 \times 128$	3.6108e+01 (0.4965)	3.7769e+01 (0.7249)	3.8200e+01 (0.8204)
$64 \times 64 \times 256$	1.9500e+01 (0.8888)	1.9641e+01 (0.9433)	1.9682e+01 (0.9567)
$128 \times 128 \times 512$	9.5911e+00 (1.0237)	9.6047e+00 (1.0321)	9.6088e+00 (1.0344)
(e) 1st order FVM with $\omega = 1$			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.0855e+01	4.5125e+01	4.7064e+01
$32 \times 32 \times 128$	2.8669e+01 (0.5110)	2.9562e+01 (0.6102)	3.0137e+01 (0.6431)
$64 \times 64 \times 256$	1.6990e+01 (0.7548)	1.7220e+01 (0.7797)	1.7454e+01 (0.7880)
$128 \times 128 \times 512$	8.9959e+00 (0.9173)	9.0551e+00 (0.9273)	9.1155e+00 (0.9371)

Table 10: Observed convergence orders for test problem with smooth source term.

Smooth source	1st order FVM		2nd order FVM	
	2-D	3-D	2-D	3-D
	Table 8	Table 6	Table 4	Table 2
FEM without advection ($\omega = 0$)	2.27	2.25	2.27	2.25
FVM without advection ($\omega = 0$)	1.94	1.95	1.94	1.95
FVM with advection weight $\omega = 0.01$	1.93	1.95	1.94	1.95
FVM with advection weight $\omega = 0.1$	1.79	1.78	1.94	1.95
FVM with advection weight $\omega = 1$	1.55	1.55	2.03	2.04

Table 11: Observed convergence orders for test problem with non-smooth source term.

Non-smooth source	1st order FVM		2nd order FVM	
	2-D	3-D	2-D	3-D
	Table 9	Table 7	Table 5	Table 3
FEM without advection ($\omega = 0$)	1.05	0.71	1.05	0.71
FVM without advection ($\omega = 0$)	1.04	0.58	1.04	0.58
FVM with advection weight $\omega = 0.01$	1.04	0.58	1.04	0.58
FVM with advection weight $\omega = 0.1$	1.03	0.57	1.03	0.57
FVM with advection weight $\omega = 1$	0.93	0.50	0.91	0.50

Tables 10 and 11 summarize observed convergence orders for smooth and non-smooth source terms, respectively, with both 1st order and 2nd order advection discretization. The convergence orders are averages from the previous tables, as noted in each column of Tables 10 and 11, of convergence rate at $t = 2, 3, 4$ on the finest mesh. First, we notice that using the finite volume method without advection, the convergence orders are consistent with those using the finite element method, in 2-D and 3-D. Second, we notice that for the problem with smooth source term in Table 10, the convergence order drops when increasing the weight of advection, if using FVM with first-order discritization. This is clearly expected and is overcome by using the second-order accurate advection discretization, which gives the optimal convergence order of 2 in all cases in Table 10. Third, in Table 11 for the problem with non-smooth source term, the spatial dimension determines the convergence orders of $q = 1$ for 2-D and $q = 0.5$ for 3-D, independent of discretization order and strength of advection. Therefore, we gain no benefit by using second-order advection discretization in the presense of non-smooth sources terms such as point sources.

Our recommendations based on observations above: (i) For smooth source term, use FVM with 2nd order advection discretization, since it remains most accurate despite increasingly dominant advection. (ii) For non-smooth source term, use FVM with 1st order advection discretization, since 2nd order advection discretization has no advantage and 1st order advection discretization requires less MPI communication.

6 PARALLEL PERFORMANCE STUDIES

In this section we demonstrate the weak scalability of the method. Parallel computing using MPI offers one key advantage: For efficient implementations of appropriate algorithms, problems can be solved significantly faster by pooling the processing power of several compute nodes. We have observed in [10] that the code demonstrates good strong scalability. That is, for a fixed problem size, as we double the number of processes, the wall clock time required for the simulation is nearly halved. However we also observe that the speedup and efficiency dropped significantly as we use large number of nodes. One of the reasons is that increasing the number of processes increases the cost of MPI communication among them. Another reason is, by splitting the fixed problem across more processes, each process has less computations to perform. The ratio of wall clock time on computation over wall clock time on MPI communication is smaller. Hence we lose the strong scalability as we reach certain large number of nodes. Moreover, the strong scability study failed to demonstrate how efficient the implementation is as we increase problem size. Another key advantage of parallel computing is, problems with larger scale can be solved within comparable time by pooling the processing power and memory of more compute nodes. A weak scalibility study is designed to show this, the basic idea is to fix the workload per node while doubling the number of nodes. Since in Section 5 we have demonstrated first-order finite volume method is preferable for CICR problem, we will use that for the remaining studies of this paper.

6.1 Weak Scalability Study Design

In order to maintain the workload for each node, while doubling the number of nodes, we first double the domain size along the z-direction. Then, for each resolution, we preserve the mesh resolution in x- and y-directions, and double the mesh resolution in z-direction.

We test the scalar test problem with smooth source term in Section 2.2 first, since it is much easier to control the workload per timestep for this basic problem. Then we move on to three-species application such as the CICR problem where the actual physiology is more complicated. In addition to doubling the domain and mesh on the z-direction, we also double the distance between the CRUs. This ensures that the total number of CRUs does not change. This increased space between CRUs is not physiologically realisitc, but is necessary to fix the sequence of opening and closing CRUs manually for all runs, allowing each run to have a stable and comparable workload per node.

Table 12 and Table 13 outline calculated number of degrees of freedom and estimated total memory usage for the scalar test problem and the CICR problem respectively. As shown in Tables 12 (a) and 13 (a), the number of degrees of freedom depends on the mesh resolution, ranging from tens of thousands all the way to 819 million for the CICR problem and 2 billion for the linear test problem. It also depends on the number of nodes used, since we attempt to fix the workload per node. Because each node has 8 parallel processes, we define $N_z = (4N_x)(p/8)$. When $p = 8$, only one node is used, and $N_z = 4N_x$ is the mesh setting we normally use to solve the CICR problem in Section 2.1. When the number of nodes doubles, the mesh resolution in z-direction doubles as well. For mesh resolution $N_x \times N_y \times N_z$, the number of control volumes $M = M_x M_y M_z = (N_x+1)(N_y+1)(N_z+1)$. Hence the number of degrees of freedom $DOF = n_s M = n_s(N_x+1)(N_y+1)(N_z+1)$. Our approach requires using 17 large arrays of length $n_s M$, therefore the memory estimation eqation is given by: Total memory $= 17 n_s M 8/(1024^3)$ GB. Here 8 represents 8 bytes of memory storage for one double precision number. Tables 12 (b) and 13 (b) show the estimated memory usage in total based on the degrees of freedom. It demonstrates that the workload per node is expected to grow with the number of nodes used, and we will compare them to the observed memory usage in Table 14 and Table 15. The reason the finest mesh in Table 13 (b) is not as fine as in Table 12 (b) is because it

would exceed the 24 GB memory per node.

Table 12: Calculated degrees of freedom and estimated total memory for weak scalability study of the scalar test problem with smooth source term. The mesh size in z-direction $N_z = (4N_x)(p/8)$ doubles as the number of processes doubles.

(a) Total number of degrees of freedom for the scalar test problem						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	18,785	37,570	75,140	150,280	300,560	601,120
32×32	140,481	280,962	561,924	1,123,848	2,247,696	4,495,392
64×64	1,085,825	2,171,650	4,343,300	8,686,600	17,373,200	34,746,400
128×128	8,536,833	17,073,666	34,147,332	68294664	136,589,328	273,178,656
256×256	67,700,225	135,400,450	270,800,900	541,601,800	1,083,203,600	2,166,407,200
(b) Estimated memory usage based on the degrees of freedom (GB)						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	0.00	0.00	0.01	0.02	0.04	0.08
32×32	0.02	0.04	0.07	0.14	0.28	0.57
64×64	0.14	0.28	0.55	1.10	2.20	4.40
128×128	1.08	2.16	4.33	8.65	17.30	34.60
256×256	8.57	17.15	34.30	68.60	137.20	274.40

Table 13: Calculated degrees of freedom and estimated total memory for weak scalability study of the CICR problem. The mesh size in z-direction $N_z = (4N_x)(p/8)$ doubles as the number of processes doubles.

(a) Total number of degrees of freedom for the calcium problem						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	56,355	112,710	225,420	45,0840	901,680	1,803,360
32×32	421,443	842,886	1,685,772	3,371,544	6,743,088	13,486,176
64×64	3,257,475	6,514,950	13,029,900	26,059,800	52,119,600	104,239,200
128×128	25,610,499	51,220,998	102,441,996	204,883,992	409,767,984	819,535,968
(b) Estimated memory usage based on the degrees of freedom (GB)						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	0.01	0.01	0.03	0.06	0.11	0.23
32×32	0.05	0.11	0.21	0.43	0.85	1.71
64×64	0.41	0.83	1.65	3.30	6.60	13.20
128×128	3.24	6.49	12.98	25.95	51.90	103.80

6.2 Weak Scalability Study Results

We present two weak scalability studies to estimate the parallel performance of the matrix-free method. Table 14 is based on the scalar test problem with smooth source term from Section 2.2 to final time $t_{\text{fin}} = 1{,}000$. Table 14 (a) reports the observed wall clock time in HH:MM:SS for each simulation. We observe that the wall clock times only increase slightly as the problem sizes increase in each row. This demonstrates weak scalability of our implementation. Table 14 (b) shows the number of time steps for the linear problem does not vary too much, even though we increase mesh resolution as well as the number of nodes. Table 14 (c) shows calculated wall clock time per time step based on data from the previous two subtables. In this case we see the same slight increase in each row as Table 14 (a). But this subtable is designed to give more insight when wall clock time and time steps differ. We also include observed memory usage per node as a relative indicator for the workload per node in Table 14 (d). In the mean time, Table 14 (e) reports the observed total memory usage. The numbers are very close to estimated numbers in Table 12 (b). These subtables demonstate we have indeed increased the workload both in terms of calculation and memory consumption. And if we increase the computing resources in the same pace, then we can expect the same wall clock time.

Table 15 shows a weak performance study for the CICR problem from Section 2.1 to final time $t_{\text{fin}} = 100$. It also has 5 subtables like Table 14. We want to bring about the same idea of doubling work load per time step while double the number of nodes used. One additional set up to ensure we get the same physiological outcome is to use an input file that specifies the open CRUs at each time step, forcing every simulation to reproduce the same CRU sequence. The input file is originally generated by a typical simulation as in Section 4. From Table 15 (a), we observe that the wall clock times in each row decrease slightly. Also, the number of time steps also decrease slightly as we increase the number of nodes used in Table 15 (b). These are different from what we observe from Table 14, where we see weak scalability but wall clock times increase slightly. This is due to the fact that we have the same number of CRUs as a typical CICR simulation, but since we increase the distances between CRUs on the z-direction, the effect of diffusion on that direction is less and the ODE problem is smoother. In Table 15 (c) We observe that the wall clock time per time step did not vary too much and memory usage per node stays the same in Table 15 (d). Hence we conclude the implementation demonstrates weak scalability, also for the CICR problem.

The hardware used for the performance studies presented in this section is part of the UMBC High Performance Computing Facility (`www.umbc.edu/hpcf`). The machine is a 86-node distributed-memory cluster with two quad-core Intel Nehalem X5550 processors (2.66 GHz, 8 MB cache) and 24 GB per node. All nodes and the 160 TB central storage are connected by an InfiniBand (QDR) interconnect network.

Table 14: Performance study for the scalar test problem with smooth source term, solved with first-order finite volume method to $t_{\text{fin}} = 1{,}000$ ms. The mesh size in z-direction $N_z = (4N_x)(p/8)$ doubles as the number of processes doubles.

(a) Wall clock time T_p in HH:MM:SS						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	00:00:21	00:00:20	00:00:21	00:00:22	00:00:25	00:00:29
32×32	00:00:39	00:00:40	00:00:39	00:00:41	00:00:45	00:00:55
64×64	00:04:02	00:04:07	00:04:16	00:04:31	00:04:58	00:05:45
128×128	00:45:34	00:46:11	00:46:56	00:49:11	00:52:52	00:59:21
256×256	10:13:48	10:17:59	10:28:33	10:41:11	11:09:21	12:09:12
(b) Time steps						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	2015	2015	2015	2015	2015	2015
32×32	2018	2018	2018	2018	2018	2018
64×64	2020	2021	2020	2020	2020	2020
128×128	2019	2019	2019	2019	2019	2019
256×256	2014	2014	2014	2014	2014	2014
(c) Wall clock time per time step in seconds						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	0.01	0.01	0.01	0.01	0.01	0.01
32×32	0.02	0.02	0.02	0.02	0.02	0.03
64×64	0.12	0.12	0.13	0.13	0.15	0.17
128×128	1.35	1.37	1.39	1.46	1.57	1.76
256×256	18.29	18.41	18.73	19.10	19.94	21.72
(d) Observed memory usage per node (GB)						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	0.10	0.12	0.12	0.13	0.10	0.10
32×32	0.12	0.14	0.14	0.15	0.12	0.11
64×64	0.26	0.27	0.27	0.28	0.25	0.25
128×128	1.34	1.34	1.34	1.35	1.31	1.31
256×256	9.73	9.73	9.73	9.73	9.70	9.70
(e) Observed total memory usage (GB)						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	0.10	0.24	0.49	1.04	1.61	3.05
32×32	0.12	0.27	0.55	1.18	1.90	3.61
64×64	0.26	0.55	1.10	2.26	4.05	7.91
128×128	1.34	2.68	5.36	10.76	21.04	41.86
256×256	9.73	19.47	38.91	77.85	155.22	310.25

Table 15: Performance study for the calcium problem, solved with first-order finite volume method to $t_{\text{fin}} = 100$ ms. The mesh size in z-direction $N_z = (4N_x)(p/8)$ doubles as the number of processes doubles.

(a) Wall clock time T_p in HH:MM:SS						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	00:00:15	00:00:14	00:00:13	00:00:13	00:00:14	00:00:14
32×32	00:02:26	00:02:21	00:02:14	00:02:06	00:02:01	00:01:56
64×64	00:27:18	00:26:35	00:25:08	00:23:50	00:22:54	00:21:35
128×128	05:03:11	04:50:14	04:38:24	04:34:29	04:20:49	04:17:25
(b) Time steps						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	2558	2401	2254	2126	1999	1893
32×32	3277	3158	2960	2738	2589	2424
64×64	4089	3913	3694	3458	3227	3030
128×128	4955	4724	4422	4153	3908	3679
(c) Wall clock time per time step in seconds						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	0.01	0.01	0.01	0.01	0.01	0.01
32×32	0.04	0.04	0.05	0.05	0.05	0.05
64×64	0.40	0.41	0.41	0.41	0.43	0.43
128×128	3.67	3.69	3.78	3.97	4.00	4.20
(d) Observed memory usage per node (GB)						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	0.11	0.13	0.13	0.13	0.11	0.10
32×32	0.16	0.18	0.18	0.19	0.16	0.15
64×64	0.57	0.58	0.58	0.59	0.56	0.55
128×128	3.74	3.75	3.75	3.76	3.73	3.72
(e) Total memory usage (GB)						
$N_x \times N_y$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$	$p = 256$
16×16	0.11	0.25	0.50	1.07	1.71	3.23
32×32	0.16	0.35	0.71	1.49	2.54	4.89
64×64	0.57	1.16	2.32	4.71	8.96	17.72
128×128	3.74	7.50	14.99	30.06	59.63	119.06

7 CONCLUSIONS

We considered the solution of advection-diffusion-reaction equations with non-linear as well as non-smooth reaction terms by a matrix-free implementation of a finite volume method. Our implementation demonstrates an effective solution of the calcium induced calcium release problem for long-time simulation in the presence of strong advection.

We give numerical evidence of convergence for test problems with smooth as well as non-smooth source terms in two-dimensional and three-dimensional settings. In the case of solving a three-dimensional problem with non-smooth source term, like the CICR model, we recommend using first-order discretization for the advection term, as opposed to second-order reconstruction. Because the latter requires MPI communication of more data among neighboring processes, but gain no benefit in terms of convergence order.

The scheme uses the implicit numerical differentiation formulas (NDF) for time integration. The matrix-vector product required in the Krylov subspace method is hardcoded without any approximations. A parallel implementation using MPI allows for efficient computations with a long simulation time. Weak scalability was introduced for the scalar test as well as CICR problem, demonstrating the ability to solve larger problem on finer mesh without significantly sacrificing efficiency.

ACKNOWLEDGMENTS

The hardware used in the computational studies is part of the UMBC High Performance Computing Facility (HPCF). The facility is supported by the U.S. National Science Foundation through the MRI program (grant nos. CNS–0821258 and CNS–1228778) and the SCREMS program (grant no. DMS–0821311), with additional substantial support from the University of Maryland, Baltimore County (UMBC). See www.umbc.edu/hpcf for more information on HPCF and the projects using its resources. Xuan Huang acknowledges support from HPCF and UMBC. Furthermore, the work of the authors Kopecz, Birken, Gobbert, and Meister was supported by the German Research Foundation as part of the SFB/TRR TR 30.

REFERENCES

[1] L. M. Blayney and F. A. Lai. Ryanodine receptor-mediated arrhythmias and sudden cardiac death. *Pharmacology & Therapeutics*, 123(2):151–177, 2009.

[2] Z. A. Coulibaly, B. E. Peercy, and M. K. Gobbert. Insight into spontaneous recurrent calcium waves in a 3-D cardiac cell based on analysis of a 1-D deterministic model. *Int. J. Comp. Math.*, 92(3):591–607, 2015.

[3] M. K. Gobbert. Long-time simulations on high resolution meshes to model calcium waves in a heart cell. *SIAM J. Sci. Comput.*, 30(6):2922–2947, 2008.

[4] A. L. Hanhart, M. K. Gobbert, and L. T. Izu. A memory-efficient finite element method for systems of reaction-diffusion equations with non-smooth forcing. *J. Comput. Appl. Math.*, 169(2):431–458, 2004.

[5] L. T. Izu, J. R. H. Mauban, C. W. Balke, and W. G. Wier. Large currents generate cardiac Ca^{2+} sparks. *Biophys. J.*, 80:88–102, 2001.

[6] L. T. Izu, S. A. Means, J. N. Shadid, Y. Chen-Izu, and C. W. Balke. Interplay of ryanodine receptor distribution and calcium dynamics. *Biophys. J.*, 91:95–112, 2006.

[7] L. T. Izu, W. G. Wier, and C. W. Balke. Evolution of cardiac calcium waves from stochastic calcium sparks. *Biophys. J.*, 80:103–120, 2001.

[8] H.-G. Roos, M. Stynes, and L. Tobiska. *Robust Numerical Methods for Singularly Perturbed Differential Equations*, volume 24 of *Springer Series in Computational Mathematics*. Springer-Verlag, 2nd edition, 2008.

[9] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, second edition, 2003.

[10] J. Schäfer, X. Huang, S. Kopecz, P. Birken, M. K. Gobbert, and A. Meister. A memory-efficient finite volume method for advection-diffusion-reaction systems with non-smooth sources. *Numer. Methods Partial Differential Equations*, 31(1):143–167, 2015.

[11] T. I. Seidman, M. K. Gobbert, D. W. Trott, and M. Kružík. Finite element approximation for time-dependent diffusion with measure-valued source. *Numer. Math.*, 122(4):709–723, 2012.

[12] L. F. Shampine. *Numerical Solution of Ordinary Differential Equations*. Chapman & Hall, 1994.

[13] L. F. Shampine and M. W. Reichelt. The MATLAB ODE suite. *SIAM J. Sci. Comput.*, 18(1):1–22, 1997.

[14] Y. Shiferaw, G. L. Aistrup, and J. A. Wasserstrom. Intracellular Ca2+ waves, afterdepolarizations, and triggered arrhythmias. *Cardiovascular Research*, 95(3):265–8, 2012.

[15] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*, volume 25 of *Springer Series in Computational Mathematics*. Springer-Verlag, second edition, 2006.