

# A Memory-Efficient Finite Volume Method for Advection-Diffusion-Reaction Systems with Non-Smooth Sources

Jonas Schäfer,<sup>1</sup> Xuan Huang,<sup>2</sup> Stefan Kopecz,<sup>1,0</sup> Philipp Birken,<sup>1</sup> Matthias K. Gobbert,<sup>2</sup> and Andreas Meister<sup>1</sup>

<sup>1</sup>Institute of Mathematics, University of Kassel, 34132 Kassel, Germany

<sup>2</sup>Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

We present a parallel matrix-free implicit finite volume scheme for the solution of unsteady three-dimensional advection-diffusion-reaction equations with smooth and Dirac-Delta source terms. The scheme is formally second order in space and a Newton-Krylov method is employed for the appearing nonlinear systems in the implicit time integration. The matrix-vector product required is hardcoded without any approximations, obtaining a matrix-free method that needs little storage and is well suited for parallel implementation. We describe the matrix-free implementation of the method in detail and give numerical evidence of its second order convergence in the presence of smooth source terms. For non-smooth source terms the convergence order drops to one half. Furthermore, we demonstrate the method's applicability for the long time simulation of calcium flow in heart cells and show its parallel scaling.

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

## I INTRODUCTION

Advection-diffusion-reaction systems occur in a wide variety of applications, as for instance heat transfer or transport-chemistry problems. Long-time simulations of such real-life applications often require implicit methods on very fine computational grids, to resolve the desired accuracy, especially in three dimensions. When performing such simulations with methods storing system matrices, the availability of memory becomes an issue already on relatively coarse meshes. One way to address this problem is the use of parallel methods, which distribute the workload among several CPUs and use the memory associated with these CPUs to solve larger systems. If this approach does not provide enough memory to obtain the desired resolution, parallel matrix-free methods are an excellent choice, since in general most of the memory is used to store system matrices. In addition, matrix-free methods require less communication compared to classical schemes, thus these are more suited for use on parallel architectures and allow for better scalability.

To solve linear systems most matrix-free methods use Krylov subspace methods, which only require the results of matrix-vector products in every iteration. If these can be provided without storing the matrix, this leads to significant savings of memory and computations on high resolution meshes become feasible. A prominent example are Jacobian-free Newton Krylov (JFNK) methods, which approximate the Jacobian matrix by finite differences via function evaluations [1]. In the following we present another type of matrix-free Newton-Krylov method, which provides the matrix-vector products with the exact Jacobian by hardcoding the product. This method is thus specific to a given class of equations and grids and particularly suitable for computations on structured grids.

---

<sup>0</sup>Correspondence to: Stefan Kopecz, Institute of Mathematics, University of Kassel, Heinrich-Plett-Str. 40, 34132 Kassel, Germany (e-mail: kopecz@mathematik.uni-kassel.de)

In [2] (see also [3] for details) a matrix-free Newton-Krylov method for the simulation of calcium flow in heart cells was presented. 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 nonlinear parts, as well as point sources. Due to the elongated shape of a heart cell and the focus of the work on physiological parameter values, 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. Good parallel scalability of the parallel implementation was demonstrated in [2]. The convergence of the finite element method in presence of measure valued source terms, as they occur in the calcium model, was rigorously shown in [4], and numerical results agree well with the theoretical predictions.

In this paper we show the performance of a formally second order finite volume scheme in the framework of the matrix-free method from [2]. Since we intend to consider advection dominated systems in the future, this is done for the more general class of equation systems of advection-diffusion-reaction (ADR) type. For this type of problems finite volume methods are a natural choice as opposed to finite element methods which would need additional stabilization terms [5]. Additionally, we deal with point sources generated by Dirac delta distributions, as they occur in the calcium flow equations. There is very little convergence theory in this case. In [6], convergence of a finite volume method is shown for hyperbolic scalar conservation laws with point sources, but no order of convergence is proved. The negative-norm convergence order of DG schemes for problems containing Dirac sources was investigated in [7].

In the absence of a general convergence theory we demonstrate convergence of the method numerically and compare the results to those obtained by the finite element method from [2], for which a rigorous convergence theory is available. To demonstrate the method's power in real life applications, we show results of long-time simulations of calcium flow in heart cells, the motivating application problem from [2]. Furthermore, a parallel performance study demonstrates the good scalability of the parallel implementation and we show that the overall performance of the finite volume scheme is comparable to the finite element method with comparable errors and runtimes. As Krylov subspace method, BiCGSTAB is employed.

The outline of this paper is as follows. Section II introduces the system of ADR equations and gives a short description of the equations modeling the calcium flow in heart cells as well as the test cases that will be used to demonstrate the convergence of numerical method. In Section III we derive the finite volume method and describe its parallel matrix-free implementation in detail. In Section IV various convergence studies are presented to give numerical evidence of the convergence of the finite volume scheme. This includes test cases with smooth and non-smooth source terms. Whenever possible, the results are compared to those obtained by the finite element scheme of [2]. Section V illustrates the applicability of the method to long time simulations of calcium in heart cells and compares the finite volume results to those obtained by the finite element method. Finally, the scalability of the scheme in parallel computations will be presented in Section VI.

## II GOVERNING EQUATIONS

We consider the three-dimensional system of advection-diffusion-reaction equations

$$u_t^{(i)} - \nabla \cdot (\mathbf{D}^{(i)} \nabla u^{(i)} - u^{(i)} \boldsymbol{\beta}^{(i)}) + a^{(i)} u^{(i)} = q^{(i)}(u^{(1)}, \dots, u^{(n_s)}) \quad (2.1)$$

of  $i = 1, \dots, n_s$  species with  $u^{(i)} = u^{(i)}(\mathbf{x}, t)$  representing functions of space  $\mathbf{x} \in \Omega \subset \mathbb{R}^3$  and time  $0 < t \leq t_{fin}$ . Without loss of generality, the components of the constant velocity vectors  $\boldsymbol{\beta}^{(i)} \in \mathbb{R}^3$  are non-negative. The entries of the diffusive diagonal matrices  $\mathbf{D}^{(i)} = \text{diag}(D_{11}^{(i)}, D_{22}^{(i)}, D_{33}^{(i)}) \in \mathbb{R}^{3 \times 3}$  are positive constants, and the reactive parameter  $a^{(i)} \geq 0$  is a constant as well. The source terms  $q^{(i)}$  represent different types of sources. In the following we consider source terms which depend on space and time solely, nonlinear reaction terms

depending on  $u^{(1)}, \dots, u^{(n_s)}$ , and point sources containing Dirac delta distributions.

To describe the problem completely initial conditions  $u^{(i)}|_{t=0} = u_{ini}^{(i)}$  as well as boundary conditions on the domain's boundary  $\partial\Omega$  are necessary. We employ homogeneous Neumann boundary conditions  $\mathbf{D}^{(i)} \nabla u^{(i)} \cdot \mathbf{n} = 0$  for  $i = 1, \dots, n_s$ , in which  $\mathbf{n}$  is the outward pointing normal vector on  $\partial\Omega$ .

## II.A Calcium Flow Model

Within the general system (2.1) a model for the calcium flow in heart cells is included, see [2] and the references therein for details. This model consists of  $n_s = 3$  equations corresponding to calcium ( $i = 1$ ), an endogenous calcium buffer ( $i = 2$ ) and a fluorescent indicator dye ( $i = 3$ ). It is obtained by neglecting advective effects, i.e.  $\beta^{(i)} = 0$  for  $i = 1, 2, 3$ , and setting

$$q^{(i)}(u^{(1)}, u^{(2)}, u^{(3)}) = r^{(i)}(u^{(1)}, u^{(2)}, u^{(3)}) + (-J_{pump}(u^{(1)}) + J_{leak} + J_{SR}(u^{(1)}))\delta_{i1}.$$

The reaction terms  $r^{(i)}$  are nonlinear functions of the different species and couple the three equations. Terms belonging only to the first equation are multiplied with the Kronecker delta function  $\delta_{i1}$ . These are the nonlinear drain term  $J_{pump}$ , the constant balance term  $J_{leak}$ , and the key of the model in  $J_{SR}$ . This term, which is given by

$$J_{SR}(u^{(1)}, \mathbf{x}, t) = \sum_{\hat{\mathbf{x}} \in \Omega_s} \sigma S_{\hat{\mathbf{x}}}(u^{(1)}, t) \delta(\mathbf{x} - \hat{\mathbf{x}}),$$

models the release of calcium into the cell at special locations called calcium release units (CRUs). The indicator function  $S_{\hat{\mathbf{x}}}$  determines whether a CRU is open or not,  $\sigma$  controls the amount of calcium injected into the cell and  $\Omega_s$  represents the set of all CRUs. Furthermore,  $\delta(\mathbf{x} - \hat{\mathbf{x}})$  denotes a Dirac delta distribution for a CRU located in  $\hat{\mathbf{x}}$ . Thus,  $J_{SR}$  models the superposition of point sources at the locations of all CRUs. A complete list of the model's parameter values is given in Table 2.1.

## II.B Scalar Test Cases

In order to demonstrate the convergence of the method presented in Section III we will consider a number of test cases, which are simplifications of the system (2.1) or the calcium model from Section II.A, respectively.

### II.B.1 ADR Equation with Smooth and Non-Smooth Source Term

The first test case is the unsteady scalar ADR equation (with superscript dropped)

$$u_t - \nabla \cdot (\mathbf{D} \nabla u) + \beta \cdot \nabla u + au = f.$$

The domain and diffusive matrix  $\mathbf{D} = \mathbf{D}^{(1)}$  are chosen as for the calcium model, see Table 2.1. The velocity is set to  $\beta = (0.1, 0.2, 0.3)^T$  and the reactive parameter is  $a = 0.1$ . Thus, advection, diffusion and reaction are of equal order of magnitude.

With respect to the right hand side, we consider a smooth and a non-smooth  $f$ . In the first case, the initial condition  $u_{ini}$  and right hand side  $f$  are chosen such that

$$u(x, y, z, t) = (1/8)(1 + \cos(\lambda_x x) e^{-D_x \lambda_x^2 t})(1 + \cos(\lambda_y y) e^{-D_y \lambda_y^2 t})(1 + \cos(\lambda_z z) e^{-D_z \lambda_z^2 t}) \quad (2.2)$$

with  $\lambda_x = \lambda_y = \pi/6.4$  and  $\lambda_z = \pi/32$  is the true solution. To show the method's convergence even in the presence of point sources, we also consider

$$f = \sigma S_{\hat{\mathbf{x}}}(t) \delta(\mathbf{x} - \hat{\mathbf{x}}),$$

Table 2.1: Parameters of the calcium flow model with  $n_s = 3$  species. The concentration unit M is short for mol/L (moles per liter).

Domains in space and time		
$\Omega = (-6.4, 6.4) \times (-6.4, 6.4) \times (-32.0, 32.0)$ in units of $\mu\text{m}$		
$0 < t \leq t_{\text{fin}}$ with $t_{\text{fin}} = 1,000$ in units of ms		
Advection-diffusion-reaction equation		
$\mathbf{D}^{(1)} = \text{diag}(0.15, 0.15, 0.30) \mu\text{m}^2 / \text{ms}$		
$\mathbf{D}^{(2)} = \text{diag}(0.01, 0.01, 0.02) \mu\text{m}^2 / \text{ms}$		
$\mathbf{D}^{(3)} = \text{diag}(0.00, 0.00, 0.00) \mu\text{m}^2 / \text{ms}$		
$\boldsymbol{\beta}^{(i)} = (0, 0, 0), \quad a^{(i)} = 0, \quad f^{(i)} \equiv 0 \quad \text{for all } i$		
$u_{\text{ini}}^{(1)} = 0.1 \mu\text{M},$	$u_{\text{ini}}^{(2)} = 45.9184 \mu\text{M},$	$u_{\text{ini}}^{(3)} = 111.8182 \mu\text{M}$
CRU coefficients		
$\Delta x_s = 0.8 \mu\text{m},$	$\Delta y_s = 0.8 \mu\text{m},$	$\Delta z_s = 2.0 \mu\text{m}$
$\sigma = 110.0 \mu\text{M} \mu\text{m}^3,$	$F = 96,485.3 \text{ C} / \text{mol}$	(Faraday constant)
$P_{\text{max}} = 0.3 / \text{ms},$	$K_{\text{prob}} = 2.0 \mu\text{M},$	$n_{\text{prob}} = 4.0$
$\Delta t_s = 1.0 \text{ ms},$	$t_{\text{open}} = 5.0 \text{ ms},$	$t_{\text{closed}} = 100.0 \text{ ms}$
Reaction terms		
$k_1^+ = 0.08 / (\mu\text{M ms}),$	$k_1^- = 0.09 / \text{ms},$	$\bar{u}_1 = 50.0 \mu\text{M}$
$k_2^+ = 0.10 / (\mu\text{M ms}),$	$k_2^- = 0.10 / \text{ms},$	$\bar{u}_2 = 123.0 \mu\text{M}$
Pump and leak terms		
$V_{\text{pump}} = 4.0 \mu\text{M} / \text{ms},$	$K_{\text{pump}} = 0.184 \mu\text{M},$	$n_{\text{pump}} = 4$
$J_{\text{leak}} = 0.320968365152510 \mu\text{M} / \text{ms}$		

with  $\sigma$  and  $u_{\text{ini}} = u_{\text{ini}}^{(1)}$  chosen as in Table 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 ever after. Therefore, we set  $\hat{\boldsymbol{x}} = (0, 0, 0)$  and

$$S_{\hat{\boldsymbol{x}}}(t) = \begin{cases} 0, & t < 1, \\ 1, & t \geq 1. \end{cases}$$

Numerical results with the smooth as well as the non-smooth source are shown in Section IV.B.

### II.B.2 Diffusion with Homogeneous and Non-Smooth Source Term

To be able to compare finite volume results to those computed with the finite element scheme from [2], we consider the test case from Section II.B.1 without advection. This means we set  $\boldsymbol{\beta} = \mathbf{0}$  and  $a = 0$  and obtain

$$u_t - \nabla \cdot (\mathbf{D}\nabla u) = f.$$

Again, we allow for the two different source terms, but due to the absence of advection and reaction the smooth source term, which is chosen such that the true solution is given by (2.2), becomes  $f = 0$ . See Section IV.A for the numerical results.

### II.B.3 Pure Advection

Finally, we consider the linear advection equation

$$u_t + \boldsymbol{\beta} \cdot \nabla u = 0,$$

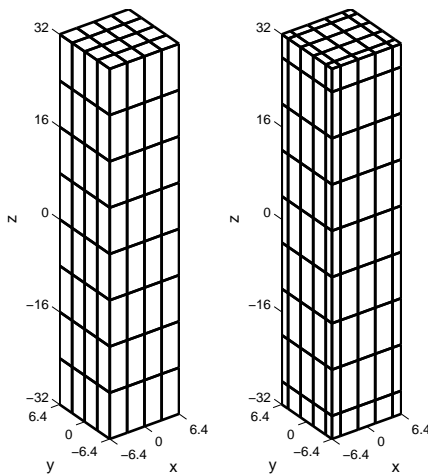


Figure 3.1: Regular mesh (left) and dual mesh (right).

with  $\boldsymbol{\beta} = (1, 1, 1)^T$  on the domain  $\Omega = (0, 10) \times (0, 10) \times (0, 10)$ . Since this is a hyperbolic problem, we only need to specify boundary conditions on the inflow portion  $\partial\Omega_{in} \subset \partial\Omega$  of the boundary. In this test case  $\partial\Omega_{in}$  is given by the intersection of  $\partial\Omega$  with the  $(x, y)$ ,  $(x, z)$  and  $(y, z)$  hyperplanes and we assume homogeneous Neumann boundary conditions on  $\partial\Omega_{in}$ . The problem's true solution is given by

$$u(x, y, z, t) = \exp\left(-5\left((x - \beta_1 t - 2)^2 + (y - \beta_2 t - 2)^2 + (z - \beta_3 t - 2)^2\right)\right),$$

when starting with initial data  $u_{ini}(x, y, z) = u(x, y, z, 0)$ . The numerical results of this test are shown in Section IV.C.

### III NUMERICAL METHOD

In this section we describe the numerical method for the solution of the system of ADR equations (2.1). In Section III.A the finite volume space discretization is explained. This also includes a discussion of the treatment of Dirac delta sources. A brief description of the time integration and the matrix-free method is given in Section III.B and the matrix-free implementation of the method is presented in detail in III.C. Finally, Section III.D discusses the parallel implementation of the method.

#### III.A Space Discretization

To derive the finite volume discretization, let  $\mathcal{T}_h = \{K_1, \dots, K_M\}$  be a mesh such that  $\bar{\Omega} = \bigcup_{l=1}^M \bar{K}_l$ . Each  $K_l \in \mathcal{T}_h$  is an open subset of  $\Omega$  and referred to as cell or control volume.

Integration of (2.1) over an arbitrary cell  $K_l \in \mathcal{T}_h$  and application of the divergence theorem yields

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

where  $\partial K_l$  denotes the boundary of  $K_l$  and  $\mathbf{n}_l$  its outward normal. This is the equation we are actually trying to solve, since it imposes less regularity on the solution. 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

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

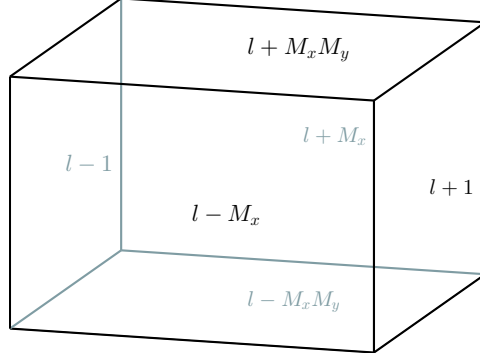


Figure 3.2: Control volume

With this notation, (3.1) can be rewritten as

$$\frac{d}{dt} \bar{u}_l^{(i)} - \frac{1}{|K_l|} \int_{\partial K_l} (\mathbf{D}^{(i)} \nabla u^{(i)} - u^{(i)} \boldsymbol{\beta}^{(i)}) \cdot \mathbf{n}_l ds + a^{(i)} \bar{u}_l^{(i)} = \frac{1}{|K_l|} \int_{K_l} q^{(i)} dx. \quad (3.2)$$

This is a system of ordinary differential equations for the temporal evolution of the mean values  $\bar{u}_l^{(i)}$ . In this regard, the crucial part is the computation of the boundary fluxes in terms of neighboring mean values.

In the following we only consider brick shaped domains. Thus, it is convenient to discuss the flux approximation in a one-dimensional context and without species index  $i$ , which will then be generalized to higher dimensions. Therefore, we consider  $\Omega = (X_{min}, X_{max})$  and the mesh

$$\mathcal{T}_h = \{K_l | K_l = (x_l, x_{l+1}), l = 1, \dots, M\},$$

with  $x_1 = X_{min}$  and  $x_{M+1} = X_{max}$ . The center of a cell  $K_l$  is given by  $m_l = (1/2)(x_l + x_{l+1})$ . Since the superscripts  $(i)$ , the right hand side and the reactive term, are of no importance for the discussion of the flux approximation, we drop them and (3.2) can be written as

$$\frac{d}{dt} \bar{u}_l - \frac{1}{\Delta x_l} (du'(x_{l+1}) - \beta u(x_{l+1}) - (du'(x_l) - \beta u(x_l))) = 0,$$

with  $\Delta x_l = x_{l+1} - x_l$ .

In order to approximate the advective flux  $\beta u(x_{l+1})$  by the mean values  $\bar{u}_l$  with  $l = 1, \dots, M$  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  $\beta \geq 0$  as before, the upwind flux function for the advective flux is given by

$$H_{upw}(\hat{u}_l^+, \hat{u}_{l+1}^-) = \beta \hat{u}_l^+. \quad (3.3)$$

It respects the direction of velocity and thus avoids instabilities in purely advective situations. Assuming a constant distribution of  $u$  in  $K_l$  and  $K_{l+1}$  leads to the definitions  $\hat{u}_l^+ = \bar{u}_l$  and  $\hat{u}_{l+1}^- = \bar{u}_{l+1}$ . Using these values as input of the upwind flux function, results in a method of first order convergence. The order can be increased using a ENO or WENO approach and providing input data based on a linear distribution of  $u$  within the cells. See [8] and the literature cited therein for details on WENO schemes. Since we consider linear advection there is no difference between ENO and WENO and the input for the flux function is given by

$$\hat{u}_l^+ = \bar{u}_l + \frac{\bar{u}_l - \bar{u}_{l-1}}{m_l - m_{l-1}} (x_{l+1} - m_l). \quad (3.4)$$

The choice of this particular linear distribution comes from the upwind idea that no information to the right of  $x_{l+1}$  should be used in the flux computation. Note that the value of  $\hat{u}_{l+1}^-$  has no influence when using the upwind flux function (3.3).

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

$$H_{diff}(\bar{u}_l, \bar{u}_{l+1}) = d \frac{\bar{u}_{l+1} - \bar{u}_l}{m_{l+1} - m_l}.$$

In this case, the mean values on either side of  $x_{l+1}$  are sufficient to obtain a scheme of second order.

So far, we have omitted the discussion of the homogeneous Neumann boundary conditions, which become  $du'(x_1) = du'(x_{M+1}) = 0$  in this one-dimensional context. In particular, the condition is equivalent to  $u'(x_1) = u'(x_{M+1}) = 0$ . The incorporation of this boundary condition is done by setting  $H_{diff}(\bar{u}_0, \bar{u}_1) = H_{diff}(\bar{u}_M, \bar{u}_{M+1}) = 0$  (or  $\bar{u}_0 = \bar{u}_1$  and  $\bar{u}_{M+1} = \bar{u}_M$ , respectively). The situation becomes more delicate for the advective flux, where two critical situations occur. These are the computations of  $H_{upw}(\hat{u}_0^+, \hat{u}_1^-)$  and  $H_{upw}(\hat{u}_1^+, \hat{u}_2^-)$ , for which it is unclear how to choose the linear distribution in  $K_1$ , since no Dirichlet data is available at the boundary. The simple remedy which we are pursuing here, is the assumption of a constant distribution in  $K_1$  and thus setting  $\hat{u}_1^+ = \bar{u}_1$  in  $H_{upw}(\hat{u}_1^+, \hat{u}_2^-)$ . Furthermore, we set  $\hat{u}_0^+ = \hat{u}_1^- = \bar{u}_1$  in  $H_{upw}(\hat{u}_0^+, \hat{u}_1^-)$  to incorporate the boundary condition  $u'(x_1) = 0$ . Note that there are no difficulties with  $H_{upw}(\hat{u}_M^+, \hat{u}_{M+1}^-)$ , since  $\hat{u}_M^+$  can be computed from (3.4) without any problems and the chosen value of  $\hat{u}_{M+1}^-$  has no further influence.

Now, we extend this approach to three dimensions and define the mesh which is used for the discretization. In [2] a regular mesh  $\Omega_h \subset \bar{\Omega}$  with constant mesh spacings  $\Delta x$ ,  $\Delta y$  and  $\Delta z$  was used for the finite element space discretization. Here we employ the corresponding dual mesh  $\mathcal{T}_h$ , which is constructed by connecting all inner nodes of  $\Omega_h$  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  $\Omega_h$  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  $\Delta x \Delta y \Delta z / 8$ , whether this cell has a common face, edge or corner with the boundary  $\partial\Omega$ . The exterior view on a typical regular mesh and the corresponding dual mesh is depicted in Figure 3.1. By construction the number of nodes of  $\Omega_h$  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

$$l = i + (j - 1)M_x + (k - 1)M_x M_y \quad (3.5)$$

for  $1 \leq i \leq M_x$ ,  $1 \leq j \leq M_y$  and  $1 \leq k \leq M_z$ , which is indicated in Figure 3.2. Thus, a control volume  $K_l$  has the neighbors  $K_{l-1}$  and  $K_{l+1}$  in  $x$ -direction,  $K_{l-M_x}$  and  $K_{l+M_x}$  in  $y$ -direction and  $K_{l-M_x M_y}$  and  $K_{l+M_x M_y}$  in  $z$ -direction.

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  $\Delta x_l = x_R - x_L$ ,  $\Delta y_l = y_R - y_L$ ,  $\Delta z_l = z_R - z_L$  the boundary fluxes from (3.2) can be written as

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

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

$$H_{a,j}(\hat{u}_L, \hat{u}_R) = \beta_j \hat{u}_L, \quad H_{d,j}(\bar{u}_L, \bar{u}_R) = d_{jj} \frac{\bar{u}_R - \bar{u}_L}{|\mathbf{m}_R - \mathbf{m}_L|}, \quad (3.6)$$

with  $\mathbf{m}_L$  and  $\mathbf{m}_R$  denoting the centers of the cells belonging to  $\bar{u}_L$  and  $\bar{u}_R$ , the boundary fluxes can be approximated as

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

where the enumeration scheme (3.5) 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 the element which lies in positive (+) or negative (-) direction of the  $j$ -th coordinate. The computation of  $\hat{u}_\alpha^{j,\pm}$  and the treatment of boundary conditions can be carried out as in the one-dimensional case.

The last step of the discretization is the treatment of the volume integral on the right hand side of (3.2). This can be approximated sufficiently using the midpoint rule

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

with  $\mathbf{m}_l$  denoting the center of  $K_l$ . In the special case of a Dirac delta function as source term, i.e.  $q(\mathbf{x}) = \delta(\mathbf{x} - \hat{\mathbf{x}})$ , the volume integral can be computed exactly. The Dirac delta function 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  $\psi \in C^0(\Omega)$ . Thus, we obtain

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

This completes the description of the finite volume discretization and after division by the local volume  $|K_l| = \Delta x_l \Delta y_l \Delta z_l$ , (3.2) can now be written as

$$\begin{aligned} & \frac{d}{dt} \bar{u}_l - \frac{1}{\Delta x_l} \left( H_{d,1}(\bar{u}_l, \bar{u}_{l+1}) - H_{a,1}(\hat{u}_l^{1,+}, \hat{u}_{l+1}^{1,-}) - (H_{d,1}(\bar{u}_{l-1}, \bar{u}_l) - H_{a,1}(\hat{u}_{l-1}^{1,+}, \hat{u}_l^{1,-})) \right) \\ & - \frac{1}{\Delta y_l} \left( H_{d,2}(\bar{u}_l, \bar{u}_{l+M_x}) - H_{a,2}(\hat{u}_l^{2,+}, \hat{u}_{l+M_x}^{2,-}) - (H_{d,2}(\bar{u}_{l-M_x}, \bar{u}_l) - H_{a,2}(\hat{u}_{l-M_x}^{2,+}, \hat{u}_l^{2,-})) \right) \\ & - \frac{1}{\Delta z_l} \left( H_{d,3}(\bar{u}_l, \bar{u}_{l+M_x M_y}) - H_{a,3}(\hat{u}_l^{3,+}, \hat{u}_{l+M_x M_y}^{3,-}) - (H_{d,3}(\bar{u}_{l-M_x M_y}, \bar{u}_l) - H_{a,3}(\hat{u}_{l-M_x M_y}^{3,+}, \hat{u}_l^{3,-})) \right) \\ & + a \bar{u}_l = q(\mathbf{m}_l, t, \bar{u}_l^{(1)}, \dots, \bar{u}_l^{(n_s)}). \quad (3.7) \end{aligned}$$

Setting

$$\bar{\mathbf{u}}^{(i)} = (\bar{u}_1^{(i)}, \dots, \bar{u}_M^{(i)})^T, \quad \mathbf{q}^{(i)} = (q_1^{(i)}, \dots, q_M^{(i)})^T, \quad q_l^{(i)} = q^{(i)}(\mathbf{m}_l, t, \bar{u}_l^{(1)}, \dots, \bar{u}_l^{(n_s)}), \quad (3.8)$$



this reads

$$\frac{d}{dt}\bar{\mathbf{u}}^{(i)} = (\mathbf{H}_{diff}^{(i)} - \mathbf{H}_{adv}^{(i)} - a^{(i)}\mathbf{I})\bar{\mathbf{u}}^{(i)} + \mathbf{q}^{(i)}(\bar{\mathbf{u}}^{(1)}, \dots, \bar{\mathbf{u}}^{(n_s)}), \quad (3.9)$$

where  $\mathbf{I} \in \mathbb{R}^{M \times M}$  is the identity matrix and the flux matrices  $\mathbf{H}_{diff}, \mathbf{H}_{adv} \in \mathbb{R}^{M \times M}$  will be derived in Section III.C. Their derivation is the key step towards a matrix-free implementation of the method.

Finally, collecting all  $n_s$  vectors  $\bar{\mathbf{u}}^{(i)}$  in  $\bar{\mathbf{U}} \in \mathbb{R}^{n_s M}$  the system (3.9) can be written as

$$\frac{d}{dt}\bar{\mathbf{U}}(t) = \mathbf{f}^{ode}(t, \bar{\mathbf{U}}(t)) \quad (3.10)$$

with  $\mathbf{f}^{ode} = (\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(n_s)})^T \in \mathbb{R}^{n_s M}$  and components

$$\mathbf{f}^{(i)} = \mathbf{A}^{(i)}\bar{\mathbf{u}}^{(i)} + \mathbf{q}^{(i)}(\bar{\mathbf{u}}^{(1)}, \dots, \bar{\mathbf{u}}^{(n_s)}), \quad (3.11)$$

with

$$\mathbf{A}^{(i)} = \mathbf{H}_{diff}^{(i)} - \mathbf{H}_{adv}^{(i)} - a^{(i)}\mathbf{I}.$$

### III.B Time Integration and Matrix-Free Krylov Subspace Method

To solve the ODE system (3.10) we use the numerical differentiation formulas (NDF $k$ ) with variable order  $1 \leq k \leq 5$  and adaptively chosen time step size, see [9, 10] for details. The time step and order selection is based on controlling the estimated truncation error [9] to tolerances of  $\varepsilon_{rel}^{ode}$  and  $\varepsilon_{abs}^{ode}$ .

This implicit method demands the solution of a nonlinear system  $\mathbf{F}^{newt}(\mathbf{U}) = \mathbf{0}$  in each time step. For its solution a matrix-free method is applied, which means that results of the Jacobian-vector products needed in the Krylov subspace method are provided directly without storing the Jacobian. This is possible since the Jacobian  $\mathbf{J}^{newt}$  has the form

$$\mathbf{J}^{newt}(t, \bar{\mathbf{U}}(t)) = \mathbf{I} - c\mathbf{J}^{ode}(t, \bar{\mathbf{U}}(t)), \quad (3.12)$$

with the identity  $\mathbf{I} \in \mathbb{R}^{n_s M \times n_s M}$ , the Jacobian  $\mathbf{J}^{ode}$  of  $\mathbf{f}^{ode}$ , a constant  $c \in \mathbb{R}$ , and we are able to compute all terms of (3.7) analytically. The purpose of this approach is to save memory and hence to 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  $\|\mathbf{F}^{newt}(\mathbf{U})\| < \varepsilon^{newt}$ .

From (3.11) we see that the Jacobian  $\mathbf{J}^{ode}$  has the structure

$$\mathbf{J}^{ode} = \begin{pmatrix} \mathbf{A}^{(1)} & & \\ & \ddots & \\ & & \mathbf{A}^{(n)} \end{pmatrix} + \begin{pmatrix} \frac{\partial \mathbf{q}^{(1)}}{\partial \bar{\mathbf{u}}^{(1)}} & \dots & \frac{\partial \mathbf{q}^{(1)}}{\partial \bar{\mathbf{u}}^{(n)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{q}^{(n)}}{\partial \bar{\mathbf{u}}^{(1)}} & \dots & \frac{\partial \mathbf{q}^{(n)}}{\partial \bar{\mathbf{u}}^{(n)}} \end{pmatrix}$$

and due to (3.8) each of the blocks  $\frac{\partial \mathbf{q}^{(i)}}{\partial \bar{\mathbf{u}}^{(j)}}$  is a diagonal matrix. Thus, the matrix-free implementation of the blocks  $\mathbf{A}^{(i)}$ , which contain the advective and diffusive flux matrices, is the essential part of a matrix-free implementation of the Jacobian  $\mathbf{J}^{newt}$ .

The Krylov subspace method used to obtain the results presented in the following sections is BiCGSTAB. Numerical experiments have shown that this method is at preferable over GMRES as well as to QMR, which has been used with the finite element method in [2]. We stop the iteration 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  $\varepsilon^{lin}$ . For an overview of Krylov subspace methods see [11].

### III.C Matrix-free Implementation of Diffusive and Advective Flux Matrices

We start with the derivation of  $\mathbf{H}_{diff}^{(i)}$  for some  $i = 1, \dots, n_s$ . Again the index  $(i)$  is dropped for clarity. From (3.7) and (3.6) we see, that multiplication of  $\mathbf{H}_{diff}(l, :)$ , the  $l$ -th row of  $\mathbf{H}_{diff}$ , with  $\bar{\mathbf{u}}$  gives

$$\begin{aligned} \mathbf{H}_{diff}(l, :)\bar{\mathbf{u}} &= \frac{1}{\Delta x_l}(H_{d,1}(\bar{u}_l, \bar{u}_{l+1}) - H_{d,1}(\bar{u}_{l-1}, \bar{u}_l)) + \frac{1}{\Delta y_l}(H_{d,2}(\bar{u}_l, \bar{u}_{l+M_x}) - H_{d,2}(\bar{u}_{l-M_x}, \bar{u}_l)) \\ &\quad + \frac{1}{\Delta z_l}(H_{d,3}(\bar{u}_l, \bar{u}_{l+M_x M_y}) - H_{d,3}(\bar{u}_{l-M_x M_y}, \bar{u}_l)) \\ &= \frac{d_{11}}{\Delta x_l} \left( \frac{\bar{u}_{l+1} - \bar{u}_l}{|\mathbf{m}_{l+1} - \mathbf{m}_l|} - \frac{\bar{u}_l - \bar{u}_{l-1}}{|\mathbf{m}_l - \mathbf{m}_{l-1}|} \right) + \frac{d_{22}}{\Delta y_l} \left( \frac{\bar{u}_{l+M_x} - \bar{u}_l}{|\mathbf{m}_{l+M_x} - \mathbf{m}_l|} - \frac{\bar{u}_l - \bar{u}_{l-M_x}}{|\mathbf{m}_l - \mathbf{m}_{l-M_x}|} \right) \\ &\quad + \frac{d_{33}}{\Delta z_l} \left( \frac{\bar{u}_{l+M_x M_y} - \bar{u}_l}{|\mathbf{m}_{l+M_x M_y} - \mathbf{m}_l|} - \frac{\bar{u}_l - \bar{u}_{l-M_x M_y}}{|\mathbf{m}_l - \mathbf{m}_{l-M_x M_y}|} \right). \end{aligned} \quad (3.13)$$

Since in most cases the distances between cell centers and the mesh widths  $\Delta x_l$ ,  $\Delta y_l$  and  $\Delta z_l$  are equal to the mesh widths  $\Delta x$ ,  $\Delta y$  and  $\Delta z$  of the regular mesh  $\Omega_h$ , it is convenient to express the matrix entries in terms of  $\Delta x$ ,  $\Delta y$  and  $\Delta z$ . For an inner control volume whose neighbors have no common boundary with the domain, i.e.

$K_l \in \{K_l | l = i + (j-1)M_x + (k-1)M_x M_y, 3 \leq i \leq M_x - 2, 3 \leq j \leq M_y - 2, 3 \leq k \leq M_z - 2\}$ , equation (3.13) leads to the classic 7-point finite difference stencil

$$\begin{aligned} \frac{d_{33}}{\Delta z^2} \bar{u}_{l-M_x M_y} + \frac{d_{22}}{\Delta y^2} \bar{u}_{l-M_x} + \frac{d_{11}}{\Delta x^2} \bar{u}_{l-1} - 2 \left( \frac{d_{11}}{\Delta x^2} + \frac{d_{22}}{\Delta y^2} + \frac{d_{33}}{\Delta z^2} \right) \bar{u}_l \\ + \frac{d_{11}}{\Delta x^2} \bar{u}_{l+1} + \frac{d_{22}}{\Delta y^2} \bar{u}_{l+M_x} + \frac{d_{33}}{\Delta z^2} \bar{u}_{l+M_x M_y}, \end{aligned}$$

since the local mesh widths and the distance between cell centers are of size  $\Delta x$ ,  $\Delta y$  or  $\Delta z$ . Thus, the only non-zero entries in the  $l$ -th row of  $\mathbf{H}_{diff}^{(i)}$  are

$$\begin{aligned} (h_{diff}^{(i)})_{l,l} &= -2 \left( \frac{d_{11}^{(i)}}{\Delta x^2} + \frac{d_{22}^{(i)}}{\Delta y^2} + \frac{d_{33}^{(i)}}{\Delta z^2} \right), & (h_{diff}^{(i)})_{l,l-1} &= (h_{diff}^{(i)})_{l,l+1} = \frac{d_{11}^{(i)}}{\Delta x^2}, \\ (h_{diff}^{(i)})_{l,l-M_x} &= (h_{diff}^{(i)})_{l,l+M_x} = \frac{d_{22}^{(i)}}{\Delta y^2}, & (h_{diff}^{(i)})_{l,l-M_x M_y} &= (h_{diff}^{(i)})_{l,l+M_x M_y} = \frac{d_{33}^{(i)}}{\Delta z^2}. \end{aligned}$$

If for instance  $K_{l-1}$  has a common boundary with the domain, the distance between the cell centers reduces to  $|\mathbf{m}_l - \mathbf{m}_{l-1}| = (3/4)\Delta x$ , which changes the entries belonging to  $\bar{u}_l^{(i)}$  and  $\bar{u}_{l-1}^{(i)}$  to

$$(h_{diff}^{(i)})_{l,l} = - \left( \frac{7}{3} \frac{d_{11}^{(i)}}{\Delta x^2} + \frac{2d_{22}^{(i)}}{\Delta y^2} + \frac{2d_{33}^{(i)}}{\Delta z^2} \right), \quad (h_{diff}^{(i)})_{l,l-1} = \frac{4}{3} \frac{d_{11}^{(i)}}{\Delta x^2}.$$

If even  $K_l$  has a common boundary with  $\partial\Omega$ , boundary conditions have to be taken into account. If for instance  $K_{l-1}$  does not exist, the term  $\frac{d_{11}^{(i)}}{\Delta x_l} \frac{u_l - u_{l-1}}{|\mathbf{m}_l - \mathbf{m}_{l-1}|}$  vanishes due to the treatment of boundary conditions described in Section III.A. This leads to

$$\begin{aligned} (h_{diff}^{(i)})_{l,l} &= - \left( \frac{8}{3} \frac{d_{11}^{(i)}}{\Delta x^2} + 2 \frac{d_{22}^{(i)}}{\Delta y^2} + 2 \frac{d_{33}^{(i)}}{\Delta z^2} \right), & (h_{diff}^{(i)})_{l,l-1} &= 0, \\ (h_{diff}^{(i)})_{l,l+1} &= \frac{8}{3} \frac{d_{11}^{(i)}}{\Delta x^2}. \end{aligned}$$

Similar modifications are required if one of the other neighbors of  $K_l$  has a common boundary with the domain or if  $K_l$  is a boundary cell at another boundary. Particular, near edges and corners of the domain several of these changes have to be considered.

To derive the advective flux matrix, we proceed as in the diffusive case ((i) abandoned for clearness) and obtain

$$\begin{aligned}
\mathbf{H}_{adv}(l, :)\bar{\mathbf{u}} &= \frac{1}{\Delta x_l} (H_{a,1}(\hat{u}_{l-1}^{1,+}, \hat{u}_l^{1,-}) - H_{a,1}(\hat{u}_l^{1,+}, \hat{u}_{l+1}^{1,-})) \\
&\quad + \frac{1}{\Delta y_l} (H_{a,2}(\hat{u}_{l-M_x}^{2,+}, \hat{u}_l^{2,-}) - H_{a,2}(\hat{u}_l^{2,+}, \hat{u}_{l+M_x}^{2,-})) \\
&\quad + \frac{1}{\Delta z_l} (H_{a,3}(\hat{u}_{l-M_x M_y}^{3,+}, \hat{u}_l^{3,-}) - H_{a,3}(\hat{u}_l^{3,+}, \hat{u}_{l+M_x M_y}^{3,-})) \\
&= \frac{\beta_1}{\Delta x_l} (\hat{u}_{l-1}^{1,+} - \hat{u}_l^{1,+}) + \frac{\beta_2}{\Delta y_l} (\hat{u}_{l-M_x}^{2,+} - \hat{u}_l^{2,+}) + \frac{\beta_3}{\Delta z_l} (\hat{u}_{l-M_x M_y}^{3,+} - \hat{u}_l^{3,+}), \quad (3.14)
\end{aligned}$$

in which the flux function data for the particular directions is given by

$$\begin{aligned}
\hat{u}_l^{1,+} &= \bar{u}_l - \frac{\Delta x_l}{2} \frac{\bar{u}_l - \bar{u}_{l-1}}{|\mathbf{m}_l - \mathbf{m}_{l-1}|}, \\
\hat{u}_l^{2,+} &= \bar{u}_l - \frac{\Delta y_l}{2} \frac{\bar{u}_l - \bar{u}_{l-M_x}}{|\mathbf{m}_l - \mathbf{m}_{l-M_x}|}, \\
\hat{u}_l^{3,+} &= \bar{u}_l - \frac{\Delta z_l}{2} \frac{\bar{u}_l - \bar{u}_{l-M_x M_y}}{|\mathbf{m}_l - \mathbf{m}_{l-M_x M_y}|}.
\end{aligned}$$

In the general case, if

$$K_l \in \{K_l | l = i + (j-1)M_x + (k-1)M_x M_y, 4 \leq i \leq M_x - 1, 4 \leq j \leq M_y - 1, 4 \leq k \leq M_z - 1\},$$

i.e.  $|K_l| = \Delta x \Delta y \Delta z$ , (3.14) can be written as

$$\begin{aligned}
\mathbf{H}_{adv}^{(i)}(l, :)\bar{\mathbf{u}}^{(i)} &= -\frac{1}{2} \frac{\beta_3^{(i)}}{\Delta z} \bar{u}_{l-2M_x M_y} - \frac{1}{2} \frac{\beta_2^{(i)}}{\Delta y} \bar{u}_{l-2M_x} - \frac{1}{2} \frac{\beta_1^{(i)}}{\Delta x} \bar{u}_{l-2} \\
&\quad + 2 \frac{\beta_3^{(i)}}{\Delta z} \bar{u}_{l-M_x M_y} + 2 \frac{\beta_2^{(i)}}{\Delta y} \bar{u}_{l-M_x} + 2 \frac{\beta_1^{(i)}}{\Delta x} \bar{u}_{l-1} - \frac{3}{2} \left( \frac{\beta_1^{(i)}}{\Delta x} + \frac{\beta_2^{(i)}}{\Delta y} + \frac{\beta_3^{(i)}}{\Delta z} \right) \bar{u}_l.
\end{aligned}$$

This gives the matrix coefficients

$$\begin{aligned}
(h_{adv}^{(i)})_{l,l} &= -\frac{3}{2} \left( \frac{\beta_1^{(i)}}{\Delta x} + \frac{\beta_2^{(i)}}{\Delta y} + \frac{\beta_3^{(i)}}{\Delta z} \right), \\
(h_{adv}^{(i)})_{l,l-2} &= -\frac{1}{2} \frac{\beta_1^{(i)}}{\Delta x}, & (h_{adv}^{(i)})_{l,l-1} &= 2 \frac{\beta_1^{(i)}}{\Delta x}, \\
(h_{adv}^{(i)})_{l,l-2M_x} &= -\frac{1}{2} \frac{\beta_2^{(i)}}{\Delta y}, & (h_{adv}^{(i)})_{l,l-M_x} &= 2 \frac{\beta_2^{(i)}}{\Delta y}, \\
(h_{adv}^{(i)})_{l,l-2M_x M_y} &= -\frac{1}{2} \frac{\beta_3^{(i)}}{\Delta z}, & (h_{adv}^{(i)})_{l,l-M_x M_y} &= 2 \frac{\beta_3^{(i)}}{\Delta z}.
\end{aligned}$$

Now we discuss some exemplary exceptions. If  $K_l$  doesn't have a right neighbor, i.e.  $K_{l+1}$  doesn't exist, but the other possible neighbors exist, then  $\Delta x_l = (1/2)\Delta x$  and  $|\mathbf{m}_l - \mathbf{m}_{l-1}| =$

$(3/4)\Delta x$ , which leads to

$$\begin{aligned} (h_{adv}^{(i)})_{l,l} &= -\frac{1}{6} \left( 16 \frac{\beta_1^{(i)}}{\Delta x} + 9 \frac{\beta_2^{(i)}}{\Delta y} + 9 \frac{\beta_3^{(i)}}{\Delta z} \right), \\ (h_{adv}^{(i)})_{l,l-1} &= \frac{11}{3} \frac{\beta_1^{(i)}}{\Delta x}, & (h_{adv}^{(i)})_{l,l-2} &= -\frac{\beta_1^{(i)}}{\Delta x}, \end{aligned}$$

while the remaining matrix entries are left unchanged. On the other side, if  $K_{l-2}$  has a common boundary with the domain, then  $|\mathbf{m}_{l-1} - \mathbf{m}_{l-2}| = (3/4)\Delta x$  and so we have

$$\begin{aligned} (h_{adv}^{(i)})_{l,l} &= -\frac{3}{2} \left( \frac{\beta_1^{(i)}}{\Delta x} + \frac{\beta_2^{(i)}}{\Delta y} + \frac{\beta_3^{(i)}}{\Delta z} \right), \\ (h_{adv}^{(i)})_{l,l-1} &= \frac{13}{6} \frac{\beta_1^{(i)}}{\Delta x}, & (h_{adv}^{(i)})_{l,l-2} &= -\frac{2}{3} \frac{\beta_1^{(i)}}{\Delta x}. \end{aligned}$$

In the case that  $K_{l-2}$  does not exist, since  $K_{l-1}$  has a common boundary with the domain, we have to take the boundary condition into account. As discussed above we set  $\hat{u}_{l-1}^{1,+} = \bar{u}_{l-1}$  and get

$$\begin{aligned} (h_{adv}^{(i)})_{l,l} &= -\frac{1}{6} \left( 10 \frac{\beta_1^{(i)}}{\Delta x} + 9 \frac{\beta_2^{(i)}}{\Delta y} + 9 \frac{\beta_3^{(i)}}{\Delta z} \right), \\ (h_{adv}^{(i)})_{l,l-1} &= \frac{5}{3} \frac{\beta_1^{(i)}}{\Delta x}, & (h_{adv}^{(i)})_{l,l-2} &= 0. \end{aligned}$$

Finally, if  $K_l$  has a common boundary with the domain on the left, we have  $|K_l| = (1/2)\Delta x\Delta y\Delta z$  and  $\hat{u}_l^{1,-} = \hat{u}_{l-1} = \bar{u}_l$ , which leads to

$$\begin{aligned} (h_{adv}^{(i)})_{l,l} &= -3 \left( \frac{\beta_2^{(i)}}{\Delta y} + \frac{\beta_3^{(i)}}{\Delta z} \right), \\ (h_{adv}^{(i)})_{l,l-2} &= 0, & (h_{adv}^{(i)})_{l,l-1} &= 0. \end{aligned}$$

Considering these exceptional cases additionally in the  $y$ - and  $z$ -direction, completes the derivation of the matrix structure.

In order to use QMR as linear solver the transpose  $(\mathbf{J}^{newt})^T$  must be available. Numerical experiments have shown that there is no benefit from using QMR, since for instance BiCGSTAB and GMRES are faster solvers which do not require matrix-vector products with the transpose. Hence, there is no benefit from an implementation of the matrix-vector product with  $(\mathbf{J}^{newt})^T$ .

### III.D Parallel Implementation

The code used to perform the parallel computations presented in this paper is an extension of the one described in [2], which uses MPI for parallel communications. Please note that the results in this paper were computed on a newer cluster than used in [2].

In order to distribute the vector of unknowns (3.8) among  $P$  processes, we split the mesh in  $z$ -direction and distribute the unknowns accordingly. In the following, we call a submesh of size  $M_x M_y$  with only one cell in  $z$ -direction a slice. Given a mesh with  $M_z$  cells in  $z$ -direction, each of the  $P$  parallel processes contains the data of a submesh of size  $M_x M_y (M_z/P)$ , i.e.  $M_z/P$  slices of the original mesh.

To compute the matrix-vector products needed in the Krylov subspace method, communication between neighboring processes is required. Denoting the portion of  $\bar{\mathbf{u}}$  which is available

on process  $0 \leq p < P$  by  $\bar{\mathbf{u}}_p = (\bar{u}_{(p-1)M_x M_y (M_z/P)+1}, \dots, \bar{u}_{pM_x M_y (M_z/P)})^T$ , we see from Section III.C that the computation of  $(\mathbf{H}_{diff} \bar{\mathbf{u}})_p$  can be accomplished locally on process  $p$  if and only if the first slice of  $\bar{\mathbf{u}}_{p+1}$  and the last slice of  $\bar{\mathbf{u}}_{p-1}$  are available besides  $\bar{\mathbf{u}}_p$ .

Regarding the advective matrix, Section III.C shows that the computation of  $(\mathbf{H}_{adv} \bar{\mathbf{u}})_p$  needs the last two slices of  $\bar{\mathbf{u}}_{p-1}$ , but none of  $\bar{\mathbf{u}}_{p+1}$ . Thus, one additional slice if needed from process  $p - 1$ .

On interconnect networks in clusters, latency is a much more significant issue than bandwidth. Therefore, the cost associated with communicating two slices from process  $p - 1$  and one slice from process  $p + 1$  in the finite volume method will not measurably change the parallel behavior of the code, compared to one slice only from each for the finite element method. Thus, we expect parallel performance of the finite volume method to be as good as that of the finite element method.

#### IV CONVERGENCE STUDIES

Here we present convergence studies for the test cases of Section II. Particularly, we show the convergence of the finite volume scheme, when point sources are present. In addition, we list the results obtained by the finite element scheme from [2] whenever possible, i.e. if the advection term is dropped.

For both methods, the  $L^2$ -norm  $\|u - u_h\|_{L^2(\Omega)}$  is used. The true solution is denoted by  $u$  and  $u_h$  is its numerical approximation. Classical results for the spatial error in this norm have the form

$$\|u(\cdot, t) - u_h(\cdot, t)\|_{L^2(\Omega)} \leq C h^q \quad \text{for all } 0 < t \leq t_{\text{fin}}, \quad (4.1)$$

where the constant  $C$  is independent of the mesh size  $h$ . The number  $q$  is the convergence order of the spatial discretization. For the finite element method, the classical theory for trilinear basis functions specifies  $q = 2$ , see, e.g., [12]. The classical theory requires the source terms to be in the function space  $L^2(\Omega)$ , which is not true for the Dirac delta distribution. The heuristic arguments and computational results of [2] indicate  $q = 0.5$  in three spatial dimensions, which has recently been rigorously confirmed [4]. For the finite volume method presented in the previous section, one would expect  $q = 2$  for smooth source terms, as well. For non-smooth sources however, there is no clear heuristic argument. It is the purpose of the following studies to determine this order.

Here we use this strategy for the test cases with nonsmooth source terms, otherwise we compare to the true solution.

All errors were measured in the  $L^2$ -norm and the error is computed as the difference of the numerical solution and a representation of the exact solution on a finer mesh. When using finite elements this representation is the trilinear nodal interpolant. In the finite volume case, we use the piecewise linear representation described in Section III.A.

In Sections IV.A and IV.B the finite element reference solution was computed on a mesh with  $256 \times 256 \times 1024$  equidistant cells and the finite volume reference on the corresponding dual mesh. Besides the errors, the following tables show the estimated convergence order, which was calculated using the formula

$$q^{est} = \log_2 \left( \frac{\|e_{2h}\|}{\|e_h\|} \right),$$

in which  $e_h$  denotes the error computed with respect to a numerical solution on a mesh with maximum mesh width  $h$ . For the finite volume results the mesh size given in the tables is the size of the primal mesh, instead of the size of the dual mesh.

To stop the BiCGSTAB iterations we choose  $\varepsilon^{lin} = 10^{-2}$ . The parameters to control the time step selection in the NDFk method are  $\varepsilon_{rel}^{ode} = 10^{-6}$  and  $\varepsilon_{abs}^{ode} = 10^{-8}$  and the tolerance for the Newton solver is  $\varepsilon^{newt} = 10^{-4}$ .

Table 4.1:  $L^2$ -error on  $\Omega$  against the true solution for the diffusive test problem with homogeneous right hand side for FE and FV method.

	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	2.4312e-01	2.1704e-01	1.9429e-01
$32 \times 32 \times 128$	6.0228e-02 (2.0132)	5.3671e-02 (2.0157)	4.7955e-02 (2.0184)
$64 \times 64 \times 256$	1.4390e-02 (2.0654)	1.2805e-02 (2.0674)	1.1416e-02 (2.0706)
$128 \times 128 \times 512$	2.9260e-03 (2.2981)	2.5918e-03 (2.3046)	2.2940e-03 (2.3152)
(a) Finite element method			
	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.1603e-01	4.0114e-01	3.8584e-01
$32 \times 32 \times 128$	1.0678e-01 (1.9620)	1.0243e-01 (1.9695)	9.8288e-02 (1.9729)
$64 \times 64 \times 256$	2.6435e-02 (2.0142)	2.5353e-02 (2.0144)	2.4333e-02 (2.0141)
$128 \times 128 \times 512$	6.2654e-03 (2.0769)	6.0116e-03 (2.0763)	5.7763e-03 (2.0747)
(b) Finite volume method			

#### IV.A Diffusion with Homogeneous and Non-Smooth Source Term

Here we consider the unsteady scalar diffusive test problems from Section II.B.2. The errors of both the finite element method and the finite volume method can be seen in Table 4.1 (homogeneous right hand side) and Table 4.2 (point source).

With respect to the homogeneous right hand side we observe second order convergence for both schemes, as expected by construction.

For the test problem with a point source, the convergence theory from [4] shows that such a source term reduces the convergence order of the finite element method to  $q = 0.5$  in three dimensions. For the finite volume method there is no such convergence theory available (at least not to our knowledge), but we can expect the order to be equally reduced.

This is confirmed by the numerical results shown in Table 4.2. We see that for both methods the convergence order is reduced from  $q = 2$  to  $q = 0.5$ . In particular, we observe the convergence of the finite volume scheme in the presence of point sources.

#### IV.B Advection-Diffusion-Reaction Equation

We now consider the advection-diffusion reaction equation with different source terms. In particular this equation contains an advective term. Since the finite element method from [2] is not applicable to problems including advection, we only present results obtained with the finite volume scheme. We consider the two different source terms, namely the smooth source term and the non-smooth source term, from Section II.B.1. The results can be found in Table 4.3.

As before in the purely diffusive test case, we observe second order convergence for the smooth source terms, whereas the convergence order drops to 0.5 for the non-smooth source term.

#### IV.C Pure Advection

This Section shows the errors and the convergence order for the pure advection test case from Section II.B.3. From Table 4.4 we observe second order convergence.

In order to solve this problem with a finite element method an extra stabilization mechanism would need to be introduced to avoid oscillations. There is a vast amount of literature concerning this problem, see [5] for example, but due to the upwind flux function this is not an issue regarding the finite volume method.

Table 4.2:  $L^2$ -error on  $\Omega$  against the reference solution for the diffusive test problem with non-smooth source term for FE and FV method.

	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	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)

(a) Finite element method

	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	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)

(b) Finite volume method

Table 4.3: Finite volume  $L^2$ -error on  $\Omega$  for the advection-diffusion-reaction problem with different source terms.

	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	4.3545e-01	4.3988e-01	4.4475e-01
$32 \times 32 \times 128$	1.1677e-01 (1.8988)	1.2002e-01 (1.8738)	1.2341e-01 (1.8495)
$64 \times 64 \times 256$	2.9780e-02 (1.9713)	3.1088e-02 (1.9489)	3.2453e-02 (1.9271)
$128 \times 128 \times 512$	7.2491e-03 (2.0385)	7.7004e-03 (2.0134)	8.1655e-03 (1.9907)

(a) Smooth source term, error compared to true solution

	$t = 2$	$t = 3$	$t = 4$
$16 \times 16 \times 64$	6.4421e+01	6.6058e+01	6.6523e+01
$32 \times 32 \times 128$	5.2605e+01 (0.2923)	5.2981e+01 (0.3183)	5.3058e+01 (0.3263)
$64 \times 64 \times 256$	4.0491e+01 (0.3776)	4.0534e+01 (0.3863)	4.0543e+01 (0.3881)
$128 \times 128 \times 512$	2.8725e+01 (0.4953)	2.8728e+01 (0.4967)	2.8729e+01 (0.4969)

(b) Non-smooth source term, error compared against reference solution

Table 4.4: Finite volume  $L^2$ -error on  $\Omega$  for the pure advection problem with homogeneous source term.

	$t = 2$	$t = 3$	$t = 4$
$128 \times 128 \times 128$	1.0605e-01	1.4547e-01	1.7831e-01
$256 \times 256 \times 256$	3.1166e-02 (1.7667)	4.5994e-02 (1.6612)	6.0303e-02 (1.5641)
$512 \times 512 \times 512$	7.9835e-03 (1.9649)	1.1946e-02 (1.9449)	1.5895e-02 (1.9237)
$1024 \times 1024 \times 1024$	1.9993e-03 (1.9975)	2.9984e-03 (1.9943)	3.9971e-03 (1.9915)

(a) Error compared to true solution

## V SIMULATION OF CALCIUM WAVES

The convergence tests of the Section IV had the purpose of ensuring that the implemented numerical method is reliable and trustworthy for the application problem with point sources. This section reports one set of results for the problem of self-initiated calcium waves, whose model was detailed in Section II.A as motivation. Table 2.1 lists the model parameters. Recall from there the spatial domain  $\Omega = (-6.4, 6.4) \times (-6.4, 6.4) \times (-32.0, 32.0)$  in units of  $\mu\text{m}$  and a large final time of  $t_{fin} = 1,000$  ms for long-time simulations. Simulations of this problem have two purposes: One is to determine if a given set of model parameters allows for self-initiation of a calcium wave. The other is to determine if self-initiation repeats, for an overall effect of several waves of increasing calcium concentration. To this end, it is vital that the final time be large enough to allow for several waves in the duration of the simulation.

The following results were obtained with the mesh size  $128 \times 128 \times 512$  using the finite volume method. Corresponding studies on other meshes with both the finite volume and finite element methods, except on very coarse meshes, gave equivalent results; due to the pseudo random number generator in the probabilistic term that determines the opening or closing of each calcium release unit (CRU), some variation in detail from run to run is expected.

Figures 5.1, 5.2, and 5.3 each show snapshots of time-dependent simulations at ten points in time. Figure 5.1 plots open calcium release units throughout the cell, each marked by a fat dot (the size of the dot does not have meaning). The simulation starts with no open CRUs at  $t = 0$  ms. By time  $t = 100$  ms, a few CRUs are open, as shown in the first sub-plot of Figure 5.1. They form a small cluster, as far as visible from this plot (but see Figure 5.3 below), which is consistent with the mechanism modeled: Once a CRU happens to open and it injects calcium into the cell, diffusion carries the calcium to its neighboring CRUs, whose probability for opening increasing with the increasing calcium concentration. This is how entire waves can self-initiate, which is born out by the second sub-plot for  $t = 200$  ms, in which the original wave has traveled in both directions through the cell. Moreover, we can see that a new wave is about to initiate at approximately the same location as the one for  $t = 100$  ms. The following sub-plots in Figure 5.1 bear out the repeated self-initiation and travel of waves through the cell. This confirms that the model is behaving correctly and the parameters in Table 2.1 are meaningful. Recall that there is a lattice of  $15 \times 15 \times 31$  CRUs throughout the interior of the cell. Of these 6,975 CRUs, on the order of hundreds are typically open at any given point in time.

Figure 5.2 presents isosurface plots of the calcium concentration at the same ten points in time. The critical value used in these plots is  $65 \mu\text{M}$ , which is above the rest value of  $0.1 \mu\text{M}$  and safely below the maximum value of several hundreds reached at an open CRU location. It is expected that the calcium concentration is high (above the critical value) around open CRUs. But what is also crucial to determine by this simulation is that the calcium concentration return to its rest value after a wave has passed and in between waves. This is clearly born out by the sub-plots at latter points in time. Notice that long-time simulations are necessary to confirm this fact; a short study up to, say, 200 ms can by contrast only confirm the self-initiation of a wave, not that the concentration returns to rest values repeatedly.

Figure 5.3 depicts confocal images of the simulation results at the ten points in time, which show the calcium concentration in a two-dimensional  $x$ - $z$ -cross-section of the domain at the plane  $y = 0$ . These images are designed to emulate the appearance of experimental results. They can provide additional detailed understanding of the shape of a wave. Recall that in Figure 5.1 at  $t = 100$  ms, one can only make out a cluster of open CRUs, and at  $t = 200$  ms, Figures 5.1 and 5.2 essentially show two blobs without discernable shape. By contrast, the sub-plots in Figure 5.3 present the shape of the CRU waves as spirals (top and bottom of domain cut off a portion). This confirms experimentally observed results and demonstrates that our simulator can reproduce this effect.



Table 6.1: Performance study of the calcium problem solved with the finite volume method. Mesh 1 represents  $16 \times 16 \times 64$ , Mesh 2 represents  $32 \times 32 \times 128$ , Mesh 3 represents  $64 \times 64 \times 256$ , Mesh 4 represents  $128 \times 128 \times 512$ . ET indicates “excessive time required” (more than 5 days), N/A indicates that the case is not feasible due to  $P \geq M_z$ .

Wall clock time $T_P$ in HH:MM:SS									
Mesh	$P = 1$	$P = 2$	$P = 4$	$P = 8$	$P = 16$	$P = 32$	$P = 64$	$P = 128$	$P = 256$
1	00:22:56	00:11:47	00:06:06	00:03:29	00:02:07	00:01:52	N/A	N/A	N/A
2	05:09:55	02:24:20	01:22:32	00:43:47	00:22:20	00:12:09	00:07:50	N/A	N/A
3	50:29:42	25:36:12	15:16:01	07:48:17	03:57:59	02:01:50	01:05:59	00:41:02	N/A
4	ET	ET	ET	82:30:31	41:51:02	21:34:45	11:25:35	06:19:12	03:52:28

The parameters of Table 2.1 were discovered by experiments of [13]. They are different from the parameters used in [2], whose simulations ended up saturating the cell (see Figure 4.5 in [2]), a physiologically incorrect effect.

## VI PARALLEL PERFORMANCE STUDIES

After the verification of the correctness of the calcium simulations in Section V, we now present parallel performance studies to demonstrate the method’s scalability. The hardware used for these performance studies is part of the UMBC High Performance Computing Facility ([www.umbc.edu/hpcf](http://www.umbc.edu/hpcf)). The machine has 82 compute nodes, each 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.

Ideally, a run using  $P$  processes should be  $p$  times as fast as the 1-process run. We use the wall clock time  $T_P$  to quantify the computing time which was required when using  $P$  processes. This time includes both the calculation time associated with arithmetic and similar operations that are local to a CPU and the communication time associated with the sending and receiving of messages between the parallel processes. For a fixed problem size, the speedup which is defined as  $S_P := T_1/T_P$  quantifies how much faster the  $P$ -process run is over one with only one process. For the  $128 \times 128 \times 512$  mesh in the finite volume method case, the definition of speedup is modified to  $S_P := 8T_8/T_P$ , since the 8-processor case is the first one to complete in reasonable time (less than 5 days). For the  $128 \times 128 \times 512$  mesh in the finite element method case, speedup is  $S_P := 16T_{16}/T_P$  for the same reason. The optimal value of  $S_P$  is  $P$ . Thus, by plotting  $S_P$  vs.  $P$ , one can get a visual impression how fast the actual performance deteriorates from the ideal one. Another way to quantify how close the speedup  $S_P$  is to its optimal value  $P$  is to plot efficiency  $E_P := S_P/P$  vs.  $P$ , whose optimal value is 1.

We present a performance study of the calcium problem, whose long run time also motivate the use of parallel computing in the first place. In addition, we compare the finite volume scheme and the finite element method from [2] with respect to scalability. The wall clock times for both methods are shown in Tables 6.1 and 6.2, respectively. The corresponding speedup and efficiency can be seen in Figures 6.1 and 6.2, respectively. Comparing speedup and efficiency, there is no significant difference between both methods. Thus, the matrix-free finite volume method is equivalently well suited for long time simulation of real-life applications as the finite element scheme.

## VII CONCLUSIONS

We considered the solution of advection diffusion reaction equations with nonlinear as well as nonsmooth reaction terms by a matrix-free implementation of a formally second order finite volume method. The order for smooth reaction terms was demonstrated to be 2, whereas it drops to 0.5 for nonsmooth source terms. The new algorithm gives similar results as a finite

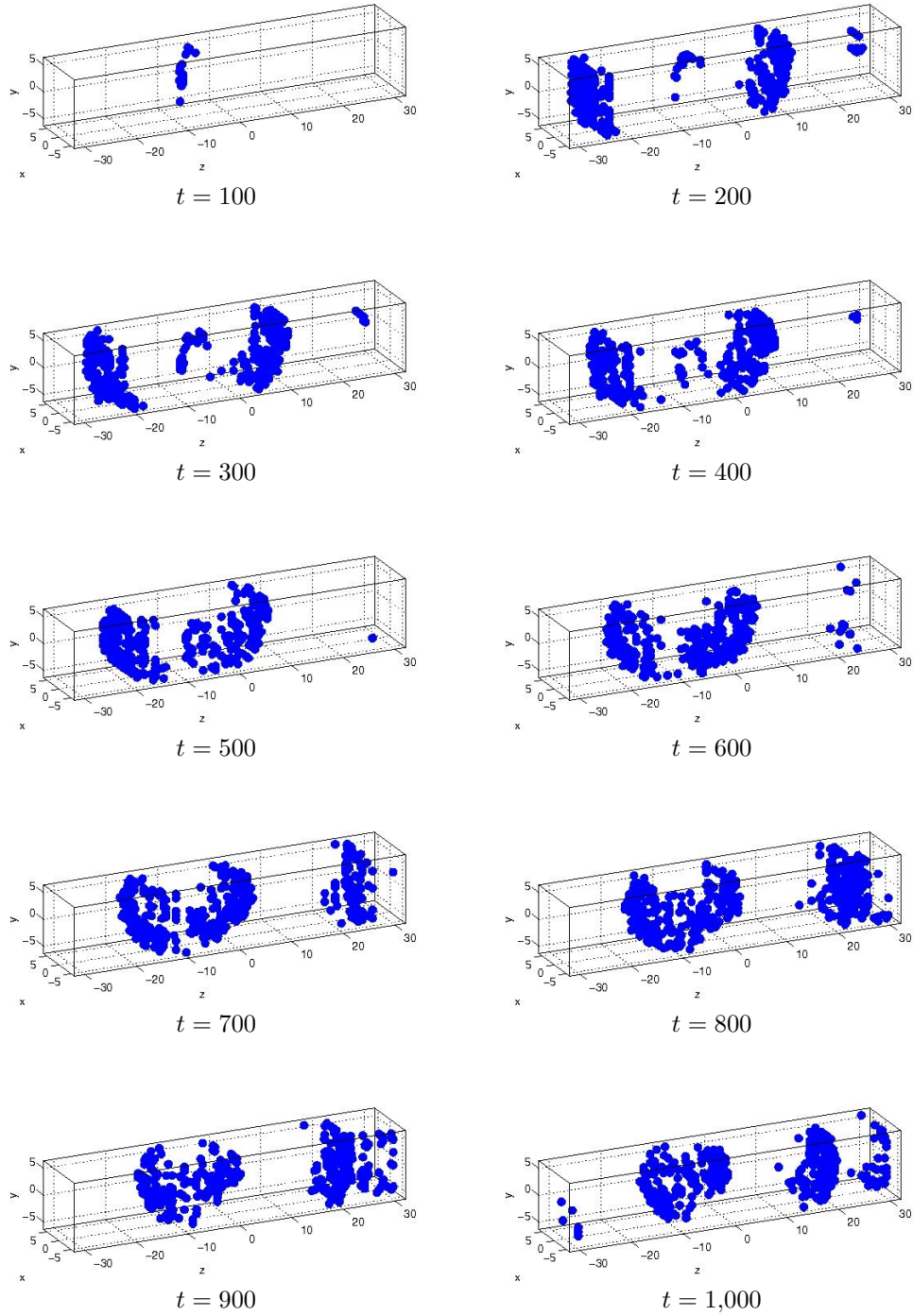


Figure 5.1: Open calcium release units throughout the cell using finite volume method with mesh size  $128 \times 128 \times 512$ .

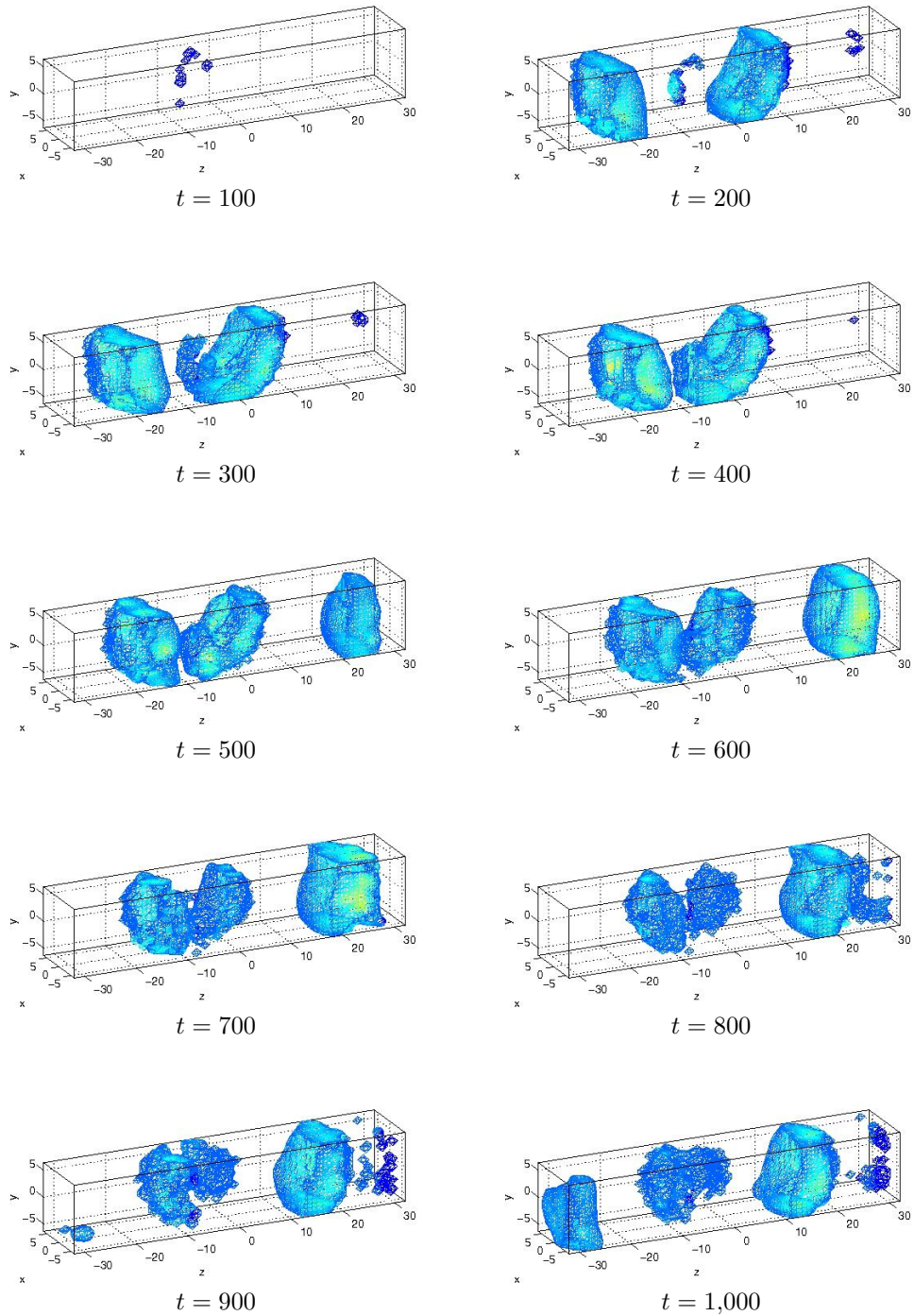


Figure 5.2: Isosurface plots of the calcium concentration using finite volume method with mesh size  $128 \times 128 \times 512$ .

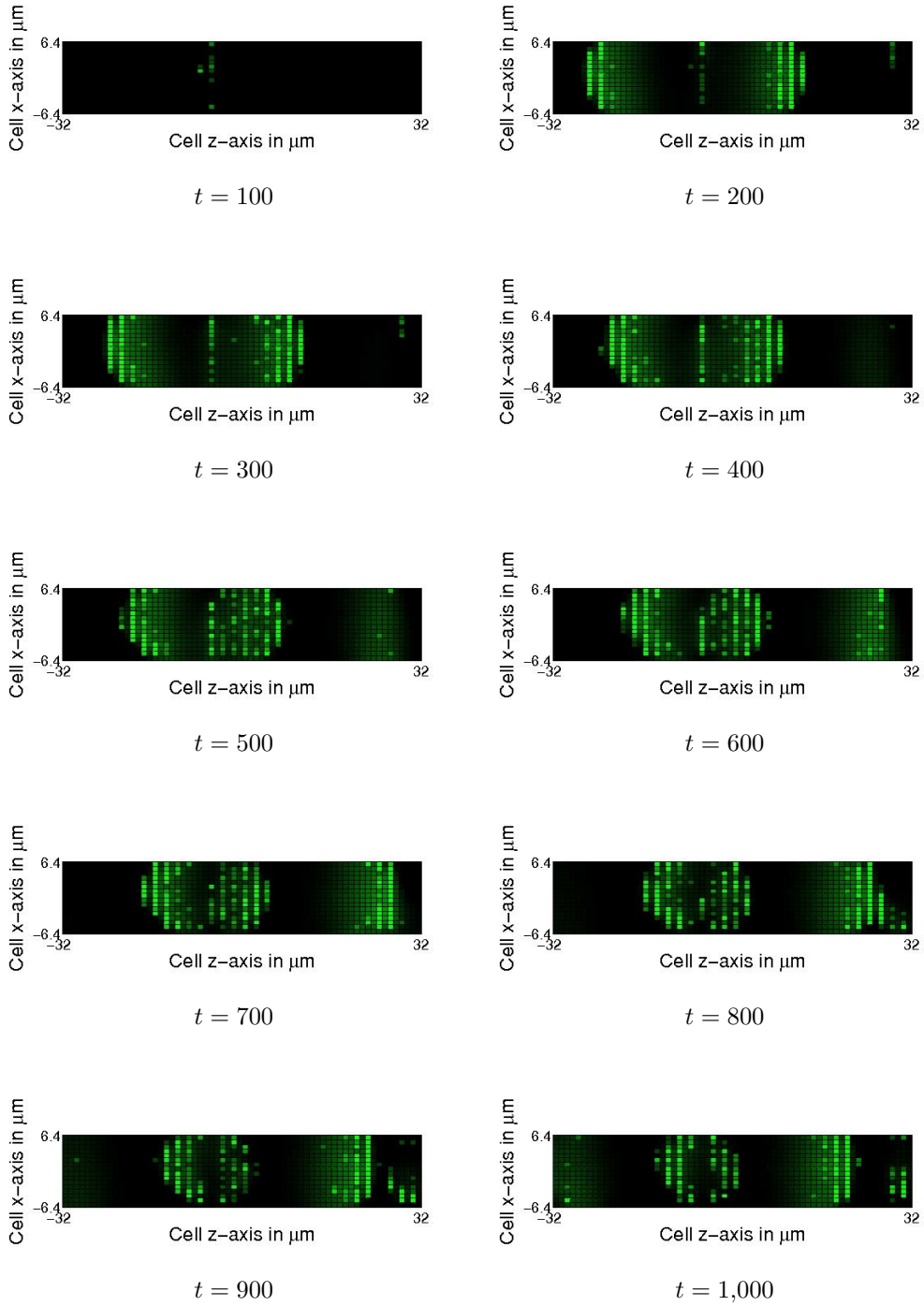


Figure 5.3: Confocal image plots of the calcium concentration using finite volume method with mesh size  $128 \times 128 \times 512$ .

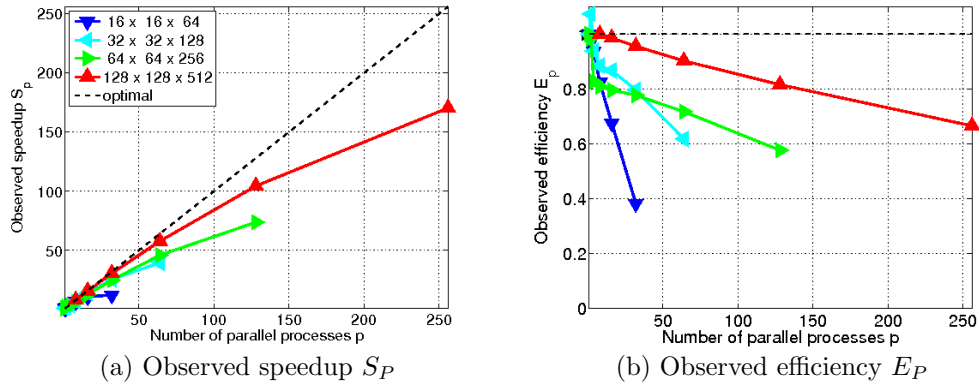


Figure 6.1: Performance study of the calcium problem solved with finite volume method.

Table 6.2: Performance study of the calcium problem solved with the finite element method. Mesh 1 represents  $16 \times 16 \times 64$ , Mesh 2 represents  $32 \times 32 \times 128$ , Mesh 3 represents  $64 \times 64 \times 256$ , Mesh 4 represents  $128 \times 128 \times 512$ . ET indicates “excessive time required” (more than 5 days), N/A indicates that the case is not feasible due to  $P > M_z$ .

Mesh	$P = 1$	$P = 2$	$P = 4$	$P = 8$	$P = 16$	$P = 32$	$P = 64$	$P = 128$	$P = 256$
1	00:08:41	00:04:37	00:02:13	00:01:20	00:00:46	00:00:31	00:00:24	N/A	N/A
2	09:23:19	04:43:52	02:33:54	01:19:35	00:40:40	00:21:53	00:12:57	00:08:45	N/A
3	67:04:28	32:09:20	18:10:07	09:17:32	04:42:50	02:25:47	01:19:28	00:45:09	00:29:06
4	ET	ET	ET	ET	74:11:02	38:00:33	19:57:57	10:49:08	06:16:11

element scheme considered in previous articles but can handle advection dominated problems. Finally, strong parallel scaling for up to 256 processes was demonstrated.

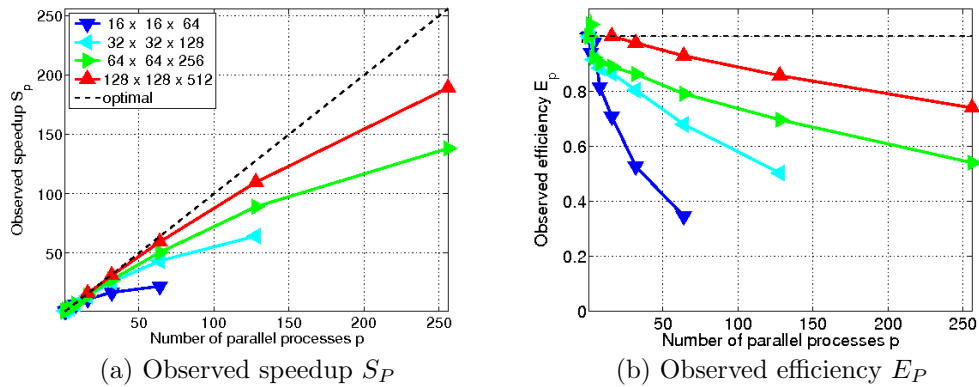


Figure 6.2: Performance study of the calcium problem solved with finite element method.

## ACKNOWLEDGMENTS

Xuan Huang acknowledges support from the UMBC High Performance Computing Facility (HPCF). The hardware used in the computational studies is part of 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](http://www.umbc.edu/hpcf) for more information on HPCF and the projects using its resources. Furthermore, the work of the authors Birken, Gobbert and Meister was supported by the German Research Foundation as part of the SFB/TRR TR 30, project C2.

## References

- [1] D. A. Knoll, D. E. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, *J. Comput. Phys.* 193 (2004) 357–397.
- [2] M. K. Gobbert, Long-time simulations on high resolution meshes to model calcium waves in a heart cell, *SIAM J. Sci. Comput.* 30 (6) (2008) 2922–2947.
- [3] A. L. Hanhart, M. K. Gobbert, 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) (2004) 431–458.
- [4] T. I. Seidman, M. K. Gobbert, D. W. Trott, M. Kružík, Finite element approximation for time-dependent diffusion with measure-valued source, *Numer. Math.* 122 (4) (2012) 709–723.
- [5] H.-G. Roos, M. Stynes, L. Tobiska, *Robust Numerical Methods for Singularly Perturbed Differential Equations*, 2nd Edition, Vol. 24 of Springer Series in Computational Mathematics, Springer-Verlag, 2008.
- [6] J. Santos, P. de Oliveira, A converging finite volume scheme for hyperbolic conservation laws with source terms, *J. Comput. Appl. Math.* 111 (1-2) (1999) 239–251.
- [7] Y. Yang, C.-W. Shu, Discontinuous galerkin method for hyperbolic equations involving  $\delta$ -singularities: negative-order norm error estimates and applications, *Numerische Mathematik* (2013) 1–29.  
URL <http://dx.doi.org/10.1007/s00211-013-0526-8>
- [8] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, *SIAM Rev.* 51 (1) (2009) 82–126.
- [9] L. F. Shampine, M. W. Reichelt, The MATLAB ODE suite, *SIAM J. Sci. Comput.* 18 (1) (1997) 1–22.
- [10] L. F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, 1994.
- [11] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd Edition, SIAM, 2003.
- [12] V. Thomée, *Galerkin Finite Element Methods for Parabolic Problems*, 2nd Edition, Vol. 25 of Springer Series in Computational Mathematics, Springer-Verlag, 2006.
- [13] Z. A. Coulibaly, B. E. Peercy, M. K. Gobbert, Spontaneous spiral wave initiation in a 3-D cardiac cell, in preparation.