Journal of Computational Interdisciplinary Sciences (2012) 3(1-2): 3-18

Printed version ISSN 1983-8409 / Online version ISSN 2177-8833
http://epacis.net
doi: 10.6062/jcis.2012.03.01.0047

Whole-cell simulations of hybrid stochastic and deterministic calcium dynamics in 3D geometry

Chamakuri Nagaiah[1] and Sten Rüdiger[2]

Manuscript received on June 11, 2012 / accepted on October 8, 2012

ABSTRACT

We developed a 3D finite element simulator interface for the numerical simulation of stochastic and deterministic equations for single and multiple clusters of Ca^{2+} releasing channels. Mathematically, diffusion, reactions and membrane transport of calcium ions in cells are represented by a coupled system of reaction-diffusion equations. We adopted a hybrid algorithm to address the coupling of the deterministic reaction-diffusion equations for Ca^{2+} and Ca^{2+} buffers and Markovian dynamics of IP_3R channel gating. Using highly unstructured meshes, our method bridges many orders of magnitude to represent accurately the Ca^{2+} distribution from the single channel to entire cells with multiple clusters of channels. To save computational time, a conforming finite element method is employed for the spatial discretization and adaptive and higher order linearly implicit methods, Rosenbrock type methods, are used for the time integration. This allows an efficient representation of inhomogeneous intra-cluster Ca^{2+} distribution at the nanometer scale even for whole-cell simulations with multiple clusters. Numerical results are demonstrated for different fine spatial resolution meshes as well as different higher order time integrators to insure the numerical convergence of schemes which we apply to study the long time behavior. The parallelization is shown to be essential by the numerical study of long time behavior of calcium concentration. We further present the parallel scalability of the deterministic equations for different arrangements of clusters. The main emphasis is on large scale and long time behavior of the studied equations that capture the detailed local dynamics as well as the temporal hierarchy of dynamical processes. Our approach thus extends our earlier simulations of release from single channels and clusters of channels and systematically integrates stochasticity on all scales of a cell's calcium dynamics.

Keywords: Intracellular calcium signals, Gillespie method, finite elements, parallel computations.

1 INTRODUCTION

Release of Ca^{2+} from intracellular stores plays a key role in the regulation of a variety of cellular activities. The Ca^{2+} often arises from channels located in the membrane of the endoplasmic reticulum (ER) and its release is modulated by the cytosolic Ca^{2+} concentration itself [5, 6, 22]. In many cell types, the main Ca^{2+} release from the ER compartment is controlled by channels such as the inositol 1,4,5-trisphosphate (IP_3) receptor channel (IP_3R). The combination of Ca^{2+} release and Ca^{2+} diffusion evokes a variety of Ca^{2+} signals depending on the number of the channels that participate in them. By way of Ca^{2+} fluorescence imaging, a hierarchy of Ca^{2+} signals has been observed in *Xenopus laevis* oocytes and other cells, which ranges from very localized events, which are called puffs and blips, to waves that propagate through the cell [23, 18] or global oscillations [4, 24]. In many cells, the

Correspondence to: Sten Rüdiger – E-mail: sten.ruediger@physik.hu-berlin.de
[1]Radon Institute for Computational and Applied Mathematics, Linz, Austria – E-mail: nagaiah.chamakuri@oeaw.ac.at
[2]Department of Physics, Humboldt University at Berlin, Germany

Ca^{2+} release channels are grouped into clusters and with current optical techniques it is difficult to resolve the individual channels within a cluster [9]. Thus, modeling tools are necessary to understand the detailed structure of calcium release and any complex type of release behavior.

Ca^{2+} puffs and waves can be modeled by a system of nonlinear partial differential equations for the concentrations of the chemical species involved. The general form is that of reaction-diffusion equations and can be written in the following way

$$\frac{\partial \mathbf{u}}{\partial t} = D\Delta \mathbf{u} + F(\mathbf{u}), \tag{1}$$

with suitable boundary and initial conditions. Here $\mathbf{u} \in \mathbb{R}^n$ denotes the concentrations of n-chemical species, D is the diffusion constant matrix and $F(\mathbf{u})$ describes the chemical reactions. On one hand, the strong localization of the released Ca^{2+} following a channel opening requires fine scales of spatial and temporal discretizations in numerical computations. On the other hand, the Ca^{2+} concentration rapidly changes over distances of several micrometers. In fact, Ca^{2+} signals are generated by local feedback as well as coupling to distant channels by transport over a μm scale. Examples of such dynamics for simulations in 2D and with limited cluster resolution were given in [20], whereas here we emphasis an efficient and accurate numerical modeling of processes on all scales in three spatial dimensions. To capture the local events within the area of a channel, a sufficiently fine spatial grid resolution is necessary. For instance, in [14, 21] the finite element implementation of calcium release is studied for cardiac myocytes. In [14], very fine structured grids are used to resolve the smallest spatial scales and the authors implemented matrix free finite element method to reduce the memory consumption on these fine spatial grids. Use of a corresponding structured grid involves several millions of degrees of freedom which requires an enormous computer memory and tremendous computational power. In the present work, the finer scales around the channels are accomplished via highly unstructured meshes. This allows a considerably coarser grid far from the release channel to be used. To this end, a conforming finite element method is employed for the spatial discretization of deterministic equations.

The Ca^{2+} release through the ionic channels depends on the open/close state of a channel, which itself is an intrinsically stochastic state variable. Changes in the Ca^{2+} concentration after channel opening/closing occur on the order of microseconds and these small time scales cannot be ignored. The opening of channels occurs at the order of microseconds and the time step recovers to order of milliseconds if all open channels are closed. These issues require robust time stepping schemes and an adaptive time step to get an accurate and efficient numerical representation. Here, we employed higher order linearly implicit methods, Rosenbrock type methods, for the time integration, which can be implemented with adaptive time step strategies.

The dynamics of channel states are stochastic because of the random binding and unbinding of messengers to the receptors. Several methods were developed to efficiently trace the noisy chemical dynamics starting with the pioneering work of Gillespie [11]. In our computations, the realizations of the stochastic solver is based on the Gillespie method. The usual Gillespie method solves stochastic processes where the propensities are constant during the subsequent transitions. However, the concentration and propensities are changing in our system because of the channel opening and closing and the ensuing dynamics of Ca^{2+}concentration. To tackle the computational complexity of stochastic simulations we follow our earlier hybrid algorithm which couples the deterministic and stochastic equations, see [27, 1].

The computation of such complex problem poses a significant demand on the computer memory as well as computational times due to the inclusion of the large number of open channels and highly stochastic nature of the channel transitions. It is therefore most natural to employ parallelization to solve such complex problems. We parallelized our hybrid algorithm codes based on the freely available public domain package DUNE [3].

The numerical results demonstrate the convergence of different space resolutions as well as different adaptive and high order time stepping schemes. The numerical results based on the hybrid algorithm will be presented for a one cluster setup and a seven cluster setup, see Section 5. Besides that the parallel efficiency is discussed at the end. We also study the hybrid method by combining the adaptive simulation scheme of the deterministic dynamics and the stochastic simulation technique for a multi-cluster setup. The numerical tests illustrate the variability in local Ca^{2+} fluctuations as induced by bursts of channel openings.

The remaining of the paper is organized as follows. A brief description of the mathematical model equations is given in the next section. In Section 3 the stochastic approach is introduced and the stochastic solver is explained. The spatial and time discretization methods to solve the deterministic equations are given in Section 4. At the end of the section, the generation of unstructured grids which were used for our computations is explained. In Section 5, the numerical results to test the convergence of spatial resolutions and for different time integrators are demonstrated

for a single cluster setup. Along with that the parallel efficiency is discussed. The numerical results for the hybrid method is discussed for the single cluster setup. Besides that the hybrid method numerical results are exhibited for 7 cluster setup as well. Finally, a summary of the work is given.

2 GOVERNING EQUATIONS

We aim to simulate the release of Ca^{2+} in a cube volume divided by the lumenal membrane. The smaller part represents the ER and the larger part is the cytosol. Channels are pores in the ER membrane with radius R_s, see Figure 1. We further consider contributions of current through the ER membrane arising from a leak current (from ER to cytosol) and ionic pumps (from cytosol to ER). Mathematically, the initial condition is the stationary Ca^{2+} - distribution resulting from a balance of currents from pumps and the leak. No-flux boundary conditions were applied at the upper surfaces of the box. We chose Cartesian coordinates for our simulations with the positive z-direction pointing from top to bottom in Figure 1. More details regarding 3D modeling can be found in [27, 26].

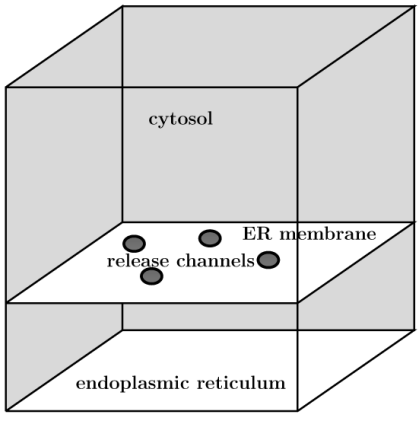


Figure 1 – The computational domain is the cuboid above the ER membrane, representing the cytosolic space. In the ER membrane several channels are placed. In the simulations there can be either one channel, a cluster of channels or several randomly placed clusters, each comprising several release channels.

The model for the calcium concentration flow is described by a system of coupled time dependent reaction-diffusion equations in three space dimensions. Here we denote the computational domain $\Omega \subset \mathbb{R}^3$ and the final time is t_f. Also, the space and time domain is denoted by $Q = \Omega \times [0, t_f]$. The equations read

$$\begin{aligned} \frac{\partial c}{\partial t} &= D_c \Delta c - \sum_j H_j(c, b_j), \\ &\forall (\mathbf{x}, t) \in \mathrm{Q} \end{aligned} \tag{2}$$

$$\begin{aligned} \frac{\partial b_j}{\partial t} &= D_j \Delta b_j + H_j(c, b_j), \quad j = s, m, d \\ &\forall (\mathbf{x}, t) \in \mathrm{Q} \end{aligned} \tag{3}$$

where $c(\mathbf{x}, t) : \mathrm{Q} \to \mathbb{R}$ is a cytosolic Ca^{2+} concentration and the buffer concentrations are denoted by $b_j(\mathbf{x}, t) : \mathrm{Q} \to \mathbb{R}$ in the cytosol. Here $j = s, d, m$ represents the buffer concentrations where $b_s(\mathbf{x}, t)$ denotes a stationary, $b_m(\mathbf{x}, t)$ a mobile and $b_d(\mathbf{x}, t)$ a dye buffers. The diffusive transport of each concentration is characterized by the diagonal and positive definite matrices $D_j \in \mathbb{R}^{3\times 3}$, $j = c, s, d, m$. In the model problem, no-flux boundary conditions are imposed for the buffer concentrations. The currents (channels, pumps, leak) are incorporated into the volume dynamics by the flux conditions at the membrane separating ER and cytosol

$$\begin{aligned} (D_c \nabla c) \cdot \eta &= (P_l + P_c(r))(E - c) \\ &- P_p \frac{c^2}{K_d^2 + c^2}, \quad \text{on } \partial\Omega_N. \end{aligned} \tag{4}$$

where $\partial\Omega_N$ represents the ER membrane surface and η is outward unit normal vector. Here, the values of c have to be taken at the membrane, E denotes the given concentration of Ca^{2+} within the cytosol and will be set to 700 μM in all of the simulations below. The channel radius is denoted by R, P_l is the coefficient of the leak flux density, P_p the maximal pump strength and the pump dissociation coefficient is denoted by K_d. The term with factor $P_c(r)$ in Eq. (4) represents the current through channels. In fact, this current depends on the cross-membrane concentration difference. This current flux is modeled as a source with constant density in a specified channel area. Here we denote by R_{ij} the radius of the jth open channel in the ith cluster. The channels are situated at fixed position $\mathbf{x}_j$ on the membrane and the channel flux term is defined by

$$P_c(\mathbf{x}, t) = \begin{cases} P_{ch} & \text{if } \|\mathbf{x} - \mathbf{x}_j\| < R_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

The P_c term establishes the channel transition at a given time instance and position of the channel. The corresponding value of P_{ch} can be found in Table 2. The channels are placed as a row by column fashion in a cluster. The number of open/closed channels is time dependent and based on the channel gating in the stochastic modeling. The reaction terms, buffer binding and unbinding of calcium, are modeled by the usual mass-action kinetic terms

$$\begin{aligned} H_j(c, b_j) &= k_j^+ (B_j - b_j)c - k_j^- b_j, \\ &\text{where } j = s, d, m, \end{aligned} \tag{5}$$

These reaction terms are nonlinear functions and couple the concentrations with each other. On and off rates of buffers, k_j^+ and k_j^-, total concentrations of buffers and remaining parameter values are given in Table 2.

3 STOCHASTIC CHANNEL DYNAMICS

One of the principal reasons that modelers and computational scientists have become more interested in Ca^{2+} dynamics is that the concentration of Ca^{2+} shows highly fluctuating spatio-temporal behavior. The process causing random behavior in intracellular Ca^{2+} dynamics is the transition between the different states of the channel (stochastic channel gating). As molecular entities, channels open and close randomly.

In this subsection, the stochastic model for the gating of IP_3R channels is explained. This model is based on the DeYoung-Keizer (DYK) model, which is based on the fact that an IP_3R channel consists of identical subunits [10]. Details of the modified DYK model which is used in our numerical calculations can be found in [27]. It is known that each of the four subunits has binding sites for IP_3 and Ca^{2+}. Based on the results of Bezprozvanny et al. [7], DeYoung & Keizer [10] proposed a model for a single subunit. The model by DeYoung & Keizer was set up as a deterministic model and used later on as a stochastic scheme. The subunit has three binding sites: an activating and an inhibitory Ca^{2+} as well as an activating IP_3 binding sites.

In this work the stochastic solver is a hybrid version of the Gillespie method [11]. The standard Gillespie algorithm uses random number pairs (r_1, r_2) and the equations

$$a_0 \cdot \tau = \ln(1/r_1)\,, \quad \sum_{j=1}^{i} a_j \le a_0 \cdot r_2 < \sum_{j=1}^{i+1} a_j\,. \qquad (6)$$

Here the reaction propensities are denoted by a_i and a_0 is sum of all propensities. Using the random numbers r_1 and r_2, which are homogeneously distributed between 0 and 1, we can find the next event R_i and that it will occur after time τ.

The Gillespie method is based on the assumption that during successive stochastic events the propensities a_i do not change. However, when linking the stochastic channel dynamics to the calcium dynamics, we expect the propensity a_i to change in time due to its dependence on the local calcium concentration c. This effect will be particularly strong for openings and closings of channels, since after such events the local calcium concentration c changes dramatically by orders of magnitude. Thus the propensities can change very rapidly over small time intervals.

To overcome those problems, a hybrid method was adopted by Rüdiger et al. [27] in which the reaction-diffusion equations are deterministic and the opening/closing of channels is considered as stochastic part. Here we will give a brief explanation of the hybrid method.

Within the hybrid method [1] the time τ to the next stochastic event is determined by solving

$$g(t+\tau|t) = \int_t^{t+\tau} a_0(c(s), s)\, ds = \xi\,, \qquad (7)$$

with $\xi = \ln(1/r_1)$, where the sum of propensities a_0 may explicitly depend both on time and the local calcium concentration. The function $g(t+\tau|t)$ is non-decreasing for $t+\tau > t$, since the propensities a are non-negative by definition. Note that the above equation simplifies to the equation determining τ in Eq. (6) in the case of constant a_0. To determine the time of next reaction τ, condition Eq. (7) is conveniently rewritten in differential form by introducing a variable $g(t)$ and solving

$$\dot{g}(s) = a_0(c, s) \qquad (8)$$

with initial condition $g(0) = 0$, along with the deterministic equations for c and buffers. To calculate the propensities we use the kinetics of the DYK model.

4 DISCRETIZATION

In this section we describe the spatial and temporal discretization approach to solve the Eqs. (2)-(3) along with boundary conditions (4).

4.1 Spatial discretization

Here we give a brief description of spatial discretization by using piecewise linear finite element method for the solution of three space dimensional reaction-diffusion model including the complex distribution of IP_3R channels which are arranged randomly at the ER membrane. Multiplying the state equations by an arbitrary test function $\phi \in C^\infty(\Omega)$, by integrating over the computational domain Ω and using the Robin type boundary conditions Eq. (4) we obtain the following variational formulation: find $c, b_s, b_m, b_d \in H^1(\Omega)$

$$\begin{aligned}\int_\Omega \frac{\partial c}{\partial t}\phi \,\mathrm{d}x \;=\; & \int_\gamma ((D_c \nabla c).\eta)\phi \,\mathrm{d}s \\ & - \int_\Omega (D_c \nabla c)\cdot\nabla\phi \,\mathrm{d}x \\ & - \int_\Omega \phi f(c, b_s, b_m, b_d)\,\mathrm{d}x\end{aligned}$$

$$\int_\Omega \frac{\partial b_s}{\partial t}\phi \,\mathrm{d}x = -\int_\Omega (D_s \nabla b_s)\cdot\nabla\phi \,\mathrm{d}x + \int_\Omega \phi H_s(c,b_s)\,\mathrm{d}x$$

$$\int_\Omega \frac{\partial b_m}{\partial t}\phi \,\mathrm{d}x = -\int_\Omega (D_m \nabla b_m)\cdot\nabla\phi \,\mathrm{d}x + \int_\Omega \phi H_m(c,b_m)\,\mathrm{d}x$$

$$\int_\Omega \frac{\partial b_d}{\partial t}\phi \,\mathrm{d}x = -\int_\Omega (D_d \nabla b_d)\cdot\nabla\phi \,\mathrm{d}x + \int_\Omega \phi H_d(c,b_d)\,\mathrm{d}x$$

where

$$f(c,b_s,b_m,b_d) = H_s(c,b_s)+H_m(c,b_m)+H_d(c,b_d).$$

Let $V_h \subset H^1(\Omega)$ be the finite dimensional subspace of piecewise linear basis functions with respect to the spatial grid. The approximate solutions c^h, b_s^h, b_m^h and b_d^h can be expressed in the form

$$c^h(t,x) = \sum_{i=0}^{N} c^i(t)\phi_i(x),$$

$$b_s^h(t,x) = \sum_{i=0}^{N} b_s^i(t)\phi_i(x),$$

$$b_m^h(t,x) = \sum_{i=0}^{N} b_m^i(t)\phi_i(x) \quad \text{and}$$

$$b_d^h(t,x) = \sum_{i=0}^{N} b_d^i(t)\phi_i(x)$$

respectively where $\{\phi_i(x)\}_{i=0}^N$ denote the basis functions.

Here we use the standard definitions of the mass matrix $M \in \mathbb{R}^{(N+1)\times(N+1)}$ whose elements are defined by $M_{kl} = \int_\Omega \phi_k\phi_l \,\mathrm{d}x$ and the stiffness matrix $A \in \mathbb{R}^{(N+1)\times(N+1)}$ where the elements defined by $A_{kl} = \int_\Omega \nabla\phi_k\nabla\phi_l \,\mathrm{d}x$. This semi-discretization in space results in the differential algebraic system,

$$M\frac{\partial c}{\partial t} = -Ac + f\,,$$

$$M\frac{\partial b_j}{\partial t} = -Ab_j + h_j\,, \quad j = s,m,d$$

where

$$f = f\left(\sum_{i=0}^{N} c^i\phi_i, \sum_{i=0}^{N} b_s^i\phi_i, \sum_{i=0}^{N} b_m^i\phi_i, \sum_{i=0}^{N} b_d^i\phi_i\right),$$

$$h_j = H_j\left(\sum_{i=0}^{N} c^i\phi_i, \sum_{i=0}^{N} b_j^i\phi_i\right), \quad \text{and} \quad j = s,m,d\,.$$

Finally, the above mentioned ordinary differential system can be written in the block matrix form as follows

$$M\frac{\partial u}{\partial t} = -Au + F(u) \tag{9}$$

where $u = (c, b_s, b_m, b_d)^T$ and $F(u)$ is a vector based on the reaction terms.

4.2 Time discretization

Now we will turn our discussion to the time discretization for the Eq. (9). Conventionally, the semi discretization of reaction-diffusion systems produces a stiff ordinary differential equations with multiple time scales. The main source of the stiffness comes from the discrete approximation of the diffusion operator and of the source term operator when channel transition occurring. We employ linearly implicit Runge-Kutta methods to solve such a stiff system in a stable and efficient way. In general, the system (9) can be expressed in the following general form,

$$M\frac{\mathrm{d}u}{\mathrm{d}t} = G(u), \qquad u(t^0) = u^0. \tag{10}$$

To solve (10), we introduce discrete steps:

$$0 = t^0, t^1, \ldots, t^n = T\,,$$

which are not necessarily equidistant. We further set $\tau^i = t^{i+1} - t^i$ and denote by u^i the numerical solution at time t^i. In computations, linearly implicit Runge-Kutta methods, specifically Rosenbrock methods, are used for the time discretization. These belong to a large class of methods which try to avoid the nonlinear system and replace it by a sequence of linear ones. Also, this allows the adaptive timescales during the simulations. An s-stage *Rosenbrock* method of order p with embedding of order $\hat{p} \neq p$ has the form

$$\left(\frac{1}{\tau^i\gamma}M - K\right)k_j = G\left(t^i + \tau^i\alpha_j, u^i + \sum_{l=1}^{j-1} a_{jl}k_l\right) - M\sum_{l=1}^{j-1}\frac{c_{lj}}{\tau^i}k_l\,, \quad j = 1,\ldots,s, \tag{11}$$

$$u^{i+1} = u^i + \sum_{l=1}^{s} m_l k_l \,, \tag{12}$$

$$\hat{u}^{i+1} = u^i + \sum_{l=1}^{s} \hat{m}_l k_l \,. \tag{13}$$

The coefficients γ, α_j, a_{jl}, c_{jl}, m_l, and $\hat{m}_l$ are chosen in such a way that certain consistency order conditions are fulfilled to obtain a sufficiently high convergence order. For the construction of the Jacobian matrix K we used exact derivatives of the vector $\mathbf{G}(\mathbf{u})$. We assume that $p > \hat{p}$ which is reasonable since one would prefer to continue the integration with the higher order solution u. After time discretization one ends up with a system of linear equations which needs to be solved by efficient iterative solvers. Here we use the BiCGSTAB [28] method with Jacobi preconditioning.

Due to the stochastic transition of IP_3R channels the discretized reaction-diffusion equations produce the stiff ordinary differential equations which requires small time scales. In this case the adaptive time stepping schemes are felicitous to speed up the simulations. These adaptive time steps were constructed by using the second solution $\hat{u}^{i+1}$ to compute a local temporal error between different orders of solutions. After the i-th integration step the value $\epsilon^{i+1} = \left\| u^{i+1} - \hat{u}^{i+1} \right\|$ is taken as an estimator of the local temporal error. In computations a new time step τ_{new}, see Gustafsson et al. [13], is calculated on the basis of

$$\bar{\tau} := \beta \tau^i \left(\frac{TOL_t}{\epsilon^{i+1}} \right)^{\frac{1}{2}} ,$$

$$\tau_{\text{new}} = \begin{cases} \beta_{\max} \tau^i, & \bar{\tau} > \beta_{\max} \tau^i, \\ \beta_{\min} \tau^i, & \bar{\tau} < \beta_{\min} \tau^i, \\ \bar{\tau}, & \text{otherwise.} \end{cases} \tag{14}$$

The parameter $\beta > 0$ is a safety factor. In order to restrict the time step jumps, the factors $\beta_{\min}$ and $\beta_{\max}$ are used. If $\epsilon^{i+1} < TOL_t$ we proceed to the next time step, otherwise the time step has to be shortened and repeated. We set $TOL_t = 0.001$ in our computations. The adaptive time stepping scheme typically allowed the time step to increase to a maximum of 0.1 s during the recovery after firing the channels and also during the resting state. We also manually set the time steps to 1 μs when the channel transition occurs in order to avoid too many time step rejections by the time stepping criteria which saves a considerable CPU time in the computations.

4.3 Generation of initial unstructured grids

Here, we briefly overview the spatial resolution of the computational domain which is adequate in these simulations. When a channel is open, the spatially localized source terms are present and require a fine spatial resolutions around the channel. For this type of complex problems, considering the fine spatial resolution of the total computational domain puts high demand on the memory as well as computational time. To resolve this issue, we used in our finite element simulations locally refined tetrahedral meshes around the channels. We put more nodal points within the pore of channels and take a sufficiently coarse mesh remote from the channels. We used a spatially exponential decay function to generate the fine meshes within the channel area. In our simulations we assumed that the radius of the channel pore is 6 nm so that we generate a grid length which is smaller than 1 nm within the area of the pore and we increase the grid length up to several tens of nm far from the channel area. We considered a domain of size $8360 \times 8360 \times 5000$ nm^3 for single cluster simulations. We refine the grid until the smallest edge is less than a nano meter at the channel location by increasing the number of refinement levels. For single cluster simulations, where a cluster consists of 9 channels, the smallest edge size is 0.976562 nm and the largest edge size is 466.701 nm and the mesh comprised 270,745 tetrahedral elements and 50,168 nodes. The complete mesh at the membrane ($z = 0$) for a single cluster is seen on the left panel of Figure 2 and a zoom of the mesh within the cluster is seen on the right hand side of Figure 2.

For the 7 cluster setup, the coarse grid consists of 605,468 tetrahedral elements and 113,498 nodes where the computational domain size is $14000 \times 14000 \times 5000$ nm^3. The corresponding mesh incorporates the smallest edge size of 0.976562 nm and the largest edge size 705.731 nm, see left hand side of Figure 3 which shows the grid at the membrane i.e. $z = 0$ plane. The right hand side of Figure 3 shows the grid structure at vertical cross section in yz-plane which is obtained by cutting the computational domain at the midpoint in the x-direction. It also matters to have a fine grid in height dependency near the channel transition to get the accurate numerical results.

4.4 Parallelization

Given the fine spatial resolution of the reaction-diffusion system and its coupling to stochastic equations with many transitions in the open/close states channels, the computational challenge is to achieve realistic simulations at acceptable computational times. The usage of parallel algorithms is therefore desirable. In this

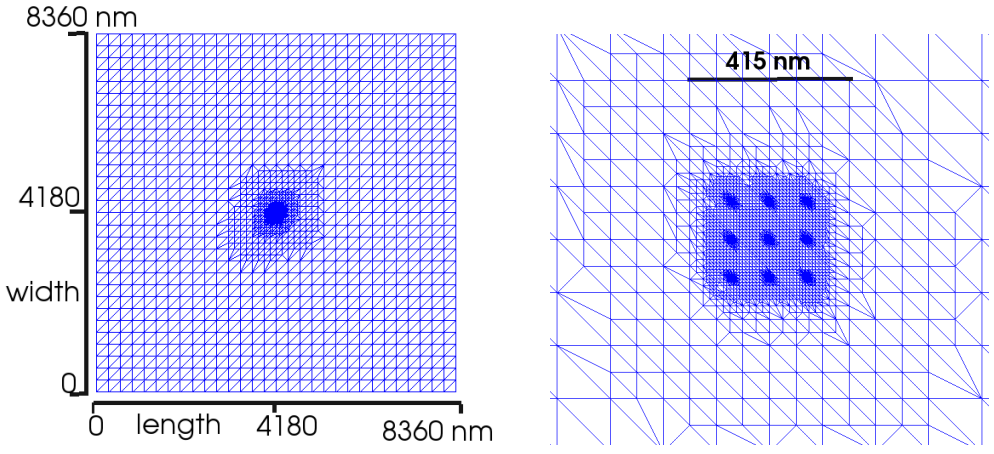


Figure 2 – Spatial resolution mesh at the membrane for a single cluster setup (left panel) and a close-up of the cluster area where the location of 9 channels is visible (right panel).

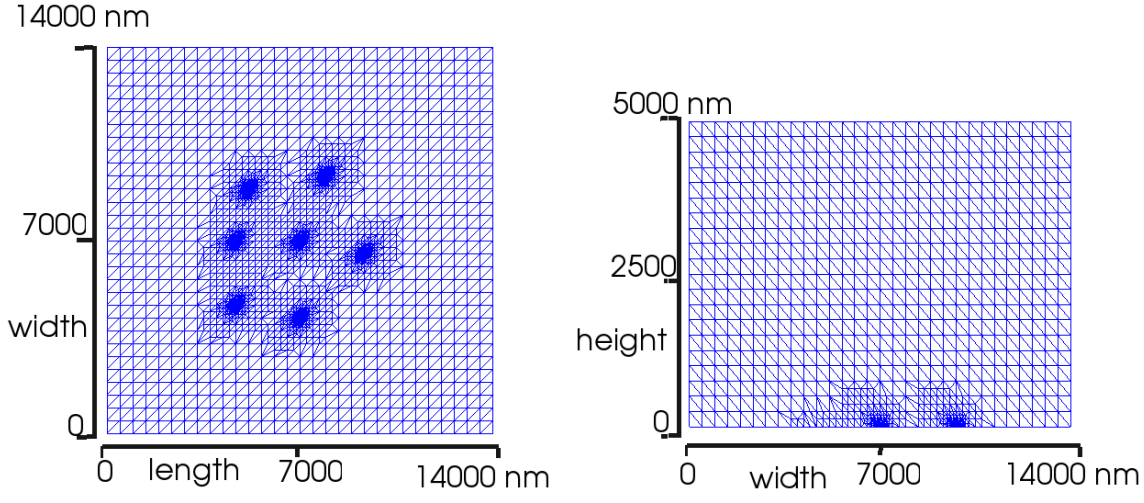
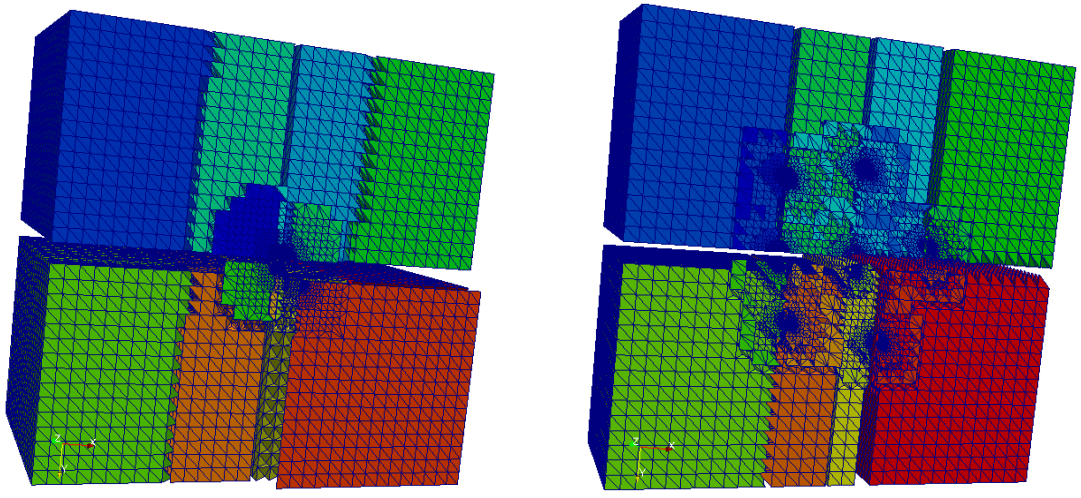


Figure 3 – The mesh for a setup with 7 clusters: mesh on the membrane ($z = 0$, left panel), vertical cross section of mesh in yz-plane at midpoint of the x-axis (right panel).

subsection we demonstrate the parallel implementation to solve the fully adaptive stochastic-deterministic system. Recent research on parallel tools and high performance computing of reaction-diffusion equations in calcium dynamics is devoted to those issues, see [12, 19]. In their work, structured and fine resolution of spatial domain is considered for the simulation of calcium dynamics in a heart cell. In our setup, the spatial grid is highly unstructured, which poses an additional questions related to load balancing.

The essence of parallel numerical simulation is to distribute the computational load evenly on all processors. This is the task of the domain decomposition algorithms. Our numerical implementation is based on the software package called DUNE [3]. The external parallel UG grid [2] interface is used for parallel grid constructions, which supports non-overlapping grids. In the UG grid manager, the recursive coordinate bisection algorithms, a divide and conquer scheme, is used for domain decomposition. In each step of this algorithm the dual graph vertices are sorted according to their coordinate values and then the vertices are bisected into two equal sets using the mean value. The sets are then further divided by the recursive application of the same splitting algorithm until the number of sets equals the number of processors. The domain decomposition of the spatial grid for one cluster setup and 7 cluster setup is shown in Figure 4(a) and Figure 4(b) respectively. Here we give a brief description of our parallel implementation to solve the hybrid algorithm.



(a) one cluster setup (b) 7 cluster setup

Figure 4 – Decomposition of the integration domain for a simulation on 16 processors. Different colors show parts of the mesh assigned to different processor for a one cluster setup (a) and a multi-cluster setup (b). The domain is rotated in space so that the membrane appears in front. Domains are spatially separated for better visualization.

Domain decomposition methods allow for the effective implementation of numerical techniques for the solution of PDEs on parallel architectures. Let us say that a original domain is partitioned into sub domains and each sub domain is assigned to a processor in a one-to-one mapping. In this case each sub domain yields a family of sub-problems of reduced size which are coupled with each other through the values of the unknown solution at the sub-domain's interfaces. The interface coupling can be relaxed by introducing additional communication in the solution of the resulting algebraic system. Our parallel implementation of the discretization routines are based on the DUNE package [3], especially the dune-pdelab discretization module. The parallel linear solvers depend on the dune-istl module. In this regard we parallelized our code by using a few global communications in our stochastic part. The BiCGSTAB [28] with Jacobi preconditioner is used as the linear solver and the relative tolerance 10^{-6} is used for the stopping criteria for the linear solver in each stage of ODE time integrator.

5 NUMERICAL RESULTS

We next describe the numerical simulations that are performed on a cubic geometry where the computational domain comprises of tetrahedral elements. The parameters that are used in the numerical simulations are shown in Table 2. The initial solution for concentrations and buffers are constants over the computational domain.

The numerical computations were performed on a Linux machine clocked at 3 GHz processor with 24 GB RAM and the program package DUNE [3] which is a public domain and written in C^{++}.

5.1 Numerical results for the single cluster setup

The numerical results for different time integrators, fine resolution time integrators and parallel efficiency for single cluster setup with deterministic (fixed) open/close dynamics of the channels are presented in this subsection. In its last part the subsection presents hybrid simulations with stochastic channel opening/closing.

5.1.1 Results on different time integrators

First, numerical experiments of non-stochastic channel gating are carried out for different time integrators. For all computations pre-generated unstructured grids, see left hand side of Figure 2, are used. We validate the numerical results based on different Rosenbrock time integrators such as Ros2 [8], Ros3p [16], Rowda [15] and Ros3pl [17] methods. The adaptive time stepping method is employed in these methods and the tolerance is adjusted accordingly to get the accurate results. The tolerances are: Ros2 method 2e-4, Ros3p method 5.0e-7, Rowda method 1e-3, Ros3pl method 5e-4. The total simulation time is $t = 0.04$ s where the channel is open from $t = 0$ s to 0.01 s and closed from time $t = 0.01$ s to $t = 0.02$ s. Likewise the channel is opened and closed once more with the same duration.

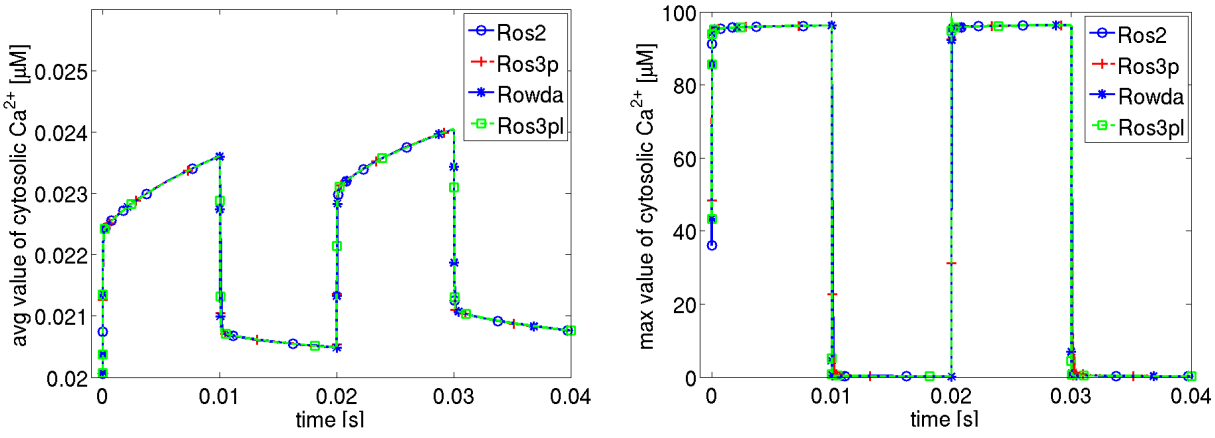


Figure 5 – The convergence of numerical solution for different time integrators, the average value and maximum value of cytosolic calcium concentration at left and right respectively.

The average and the maximum value of cytosolic Ca^{2+} concentration is shown in Figure 5 for different time integrators. We can observe that all methods exhibit similar results in the case of average value of cytosolic Ca^{2+} concentration. Ros3p method shows numerical oscillations in the case of maximum cytosolic Ca^{2+} concentration when the channel transition occurs.

To compute the simulation time $t = 0.04$ s the Rowda method takes 69 accepted time steps and rejected 45 times. Moreover, the computational time is 1893.7 seconds. The Ros2 method takes 166 accepted time steps and 6 rejected time steps. Due to the two internal stages and smaller tolerance it takes more iteration compared with the Rowda method. The Ros2 method takes 1.1838 times the Rowda method CPU time. The Ros3p method takes 140 accepted time steps and 17 rejected time steps. This method takes 1.1565 times the CPU time of the Rowda method. The Ros3pl method takes 60 accepted time steps and 54 rejected time steps to converge the solution. Due to 4 internal stages, it takes less iterations and has many time step rejections as well. To keep the computational time competing with other methods, we have chosen the same tolerance for all those methods which are mentioned. It exhibited a smaller number of rejected time steps while increasing the CPU time. In this case, the Ros3pl method takes 1.1146 times the Rowda method CPU time. From the above discussions, we concluded that the Rowda method performed best in terms of computational time. Thus, from here on we employ the Rowda method for further test cases.

5.1.2 Investigation on fine mesh resolutions

Due to the large number of grid points, if spatial resolution is at nanometer scale, it is essential to study the convergence of spatial resolution. In this regard, we present the numerical results based on the different resolutions of meshes. To generate the spatial grids we employed the adaptive strategy to place more points within and at the neighborhood of a channel, which is explained in subsection 4.3. We employ the grids based on different levels of refinement from level 6 to level 9. The level 6 spatial grid comprises of 228,389 tetrahedral elements and 42,289 nodes where the smallest edge size 1.95312 nm and the largest edge size 466.701 nm. The 7th level mesh consists of 270,745 tetrahedral elements and 50,168 nodal points. In this case the smallest edge size is 0.976563 nm. The level 8 mesh consists of 369,595 tetrahedral elements and 72,913 nodes where the smallest edge size is 0.488281 nm. The level 9 contains of 840,754 tetrahedral elements and 151,005 nodes where smallest edge size is 0.24414 nm.

The average and maximum value of cytosolic Ca^{2+} concentration is depicted in Figure 6 for different grid levels of refinement. We can observe that the level 6 solution deviates from other levels of solution and it is sufficient to consider the level 7 grid to achieve the accurate numerical solution for this problem.

5.1.3 Parallel efficiency

In this subsection we describe the parallel efficiency for different numbers of processors for the simulation of a single cluster arrangement. The sufficiently fine resolution mesh and a sufficient number of nodal points on each core for a 16 CPU cores simulation, level 8 grid, which consists of 369,595 tetrahedral elements and 72,913 nodes, is used for the comparison of parallel efficiency. To obtain parallel efficiency we considered the deterministic equations where all the channels open and close once

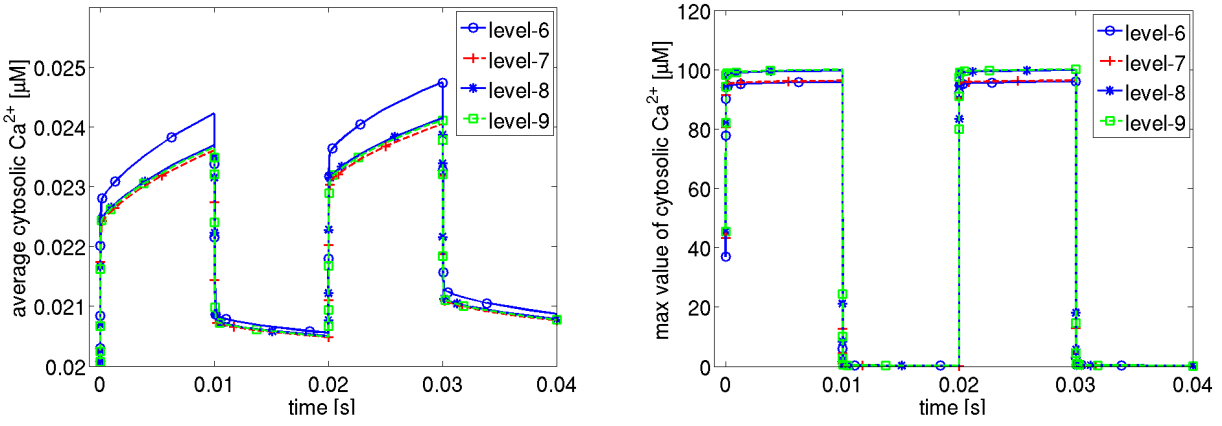


Figure 6 – The convergence of numerical solution for different levels of mesh refinement at the calcium channels.

during the simulation time for $t = 0.02$ s. To this end, the channels are opened at time $t = 0$ s and then the open channel closes at time $t = 0.01$ s. We use a Rowda method as the time integrator to solve the discretized deterministic equations. The parallel computations are done on a Linux cluster with two nodes, each of them consisting of 8 quad-core AMD Opteron processors 8356 clocked at 2.3 GHz, and equipped with 24 GB RAM.

Table 1 – Comparison of CPU times and efficiency using different numbers of processors for solution of deterministic equations.

procs (P)	CPU time	efficiency
1	3149	1.0
2	2047	0.76
4	1066	0.73
8	570	0.69
16	310	0.61

Here we have tested the parallel efficiency for up to 16 cores. Results are listed in Table 1. The parallel efficiency is measured using the following strong scaling where $T(1)$ is the CPU time on one CPU and $T(p)$ is the CPU time on p processors.

$$e = \frac{1}{p}\frac{T(1)}{T(p)} \tag{15}$$

For highly unstructured grids, the grid distribution to the processors is not trivial. We have observed that 15% of the computational mesh nodes are imbalanced and this leads to a reduction of the parallel efficiency. We can see that the parallel efficiency decreases down to 61% on 16 cores.

5.1.4 Hybrid stochastic and deterministic results

The results of full stochastic-deterministic simulations are presented in this subsection for a single cluster arrangement which consists of 9 channels. For these simulations we use the grid level 7 which consists of 270,745 tetrahedral elements and 50,168 nodes which is sufficient to get accurate numerical results based on the previous experiments. The Rowda method is employed as the time integrator. Here we stress that both the deterministic and stochastic solvers utilize adaptive time steps which are essential to get a numerical result efficiently. The simulations are done for 10 s of real simulation time.

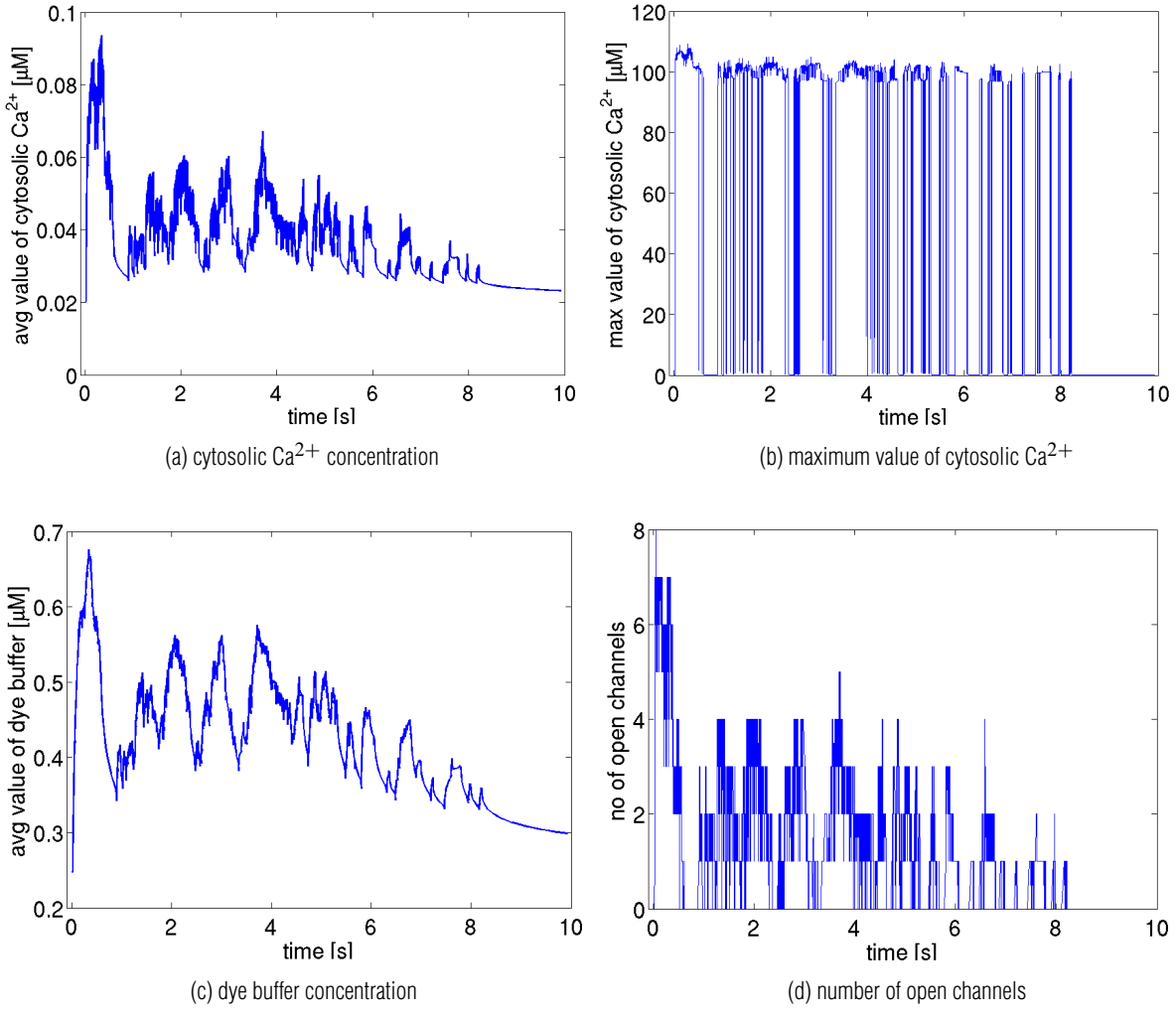
A typical simulation run is shown in Figure 7, which presents the average value of cytosolic Ca^{2+} concentration, the maximum Ca^{2+} concentration, the average dye buffer concentration and the number of open channels over the simulation time. We can observe that the opening of channels occurs in a burst-like and stochastic way. Importantly, release of Ca^{2+} couples the gating dynamics of channels and stimulates the opening of almost all channels in the cluster.

The observed release of Ca^{2+} lasts for several seconds and the dynamics is split into a first short spike and a long release tail. This finding shares its characteristics with long release events in cells, such as those observed during Ca^{2+} waves. For a discussion of the implications of this behavior in a strongly reduced model we refer to [25].

The computational time is about 13 days, 22 hours on single core machine and we observed that the parallel run on 16 cores shows parallel efficiency of up to 58% in terms of strong scaling.

5.2 Multi cluster simulations

In this subsection we present the numerical results for the 7 cluster setup which consists of maximum 9 open channels in each respective cluster. The fine grid which consists of 605,468 tetra-



(a) cytosolic Ca^{2+} concentration

(b) maximum value of cytosolic Ca^{2+}

(c) dye buffer concentration

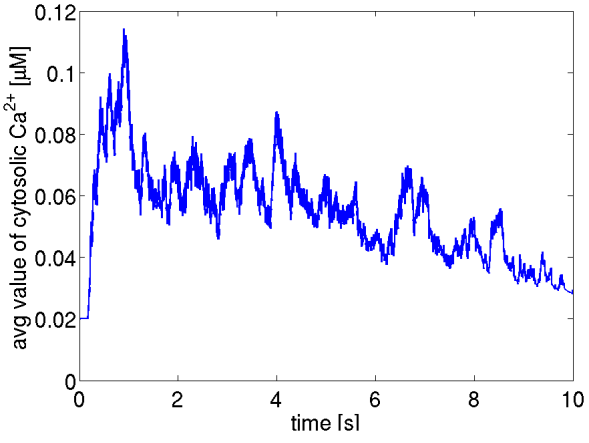
(d) number of open channels

Figure 7 – The average and maximum value of cytosolic Ca^{2+} concentration in (a) and (b), the average value of dye buffer concentration in (c) and the number of open channels in (d) over time is plotted.

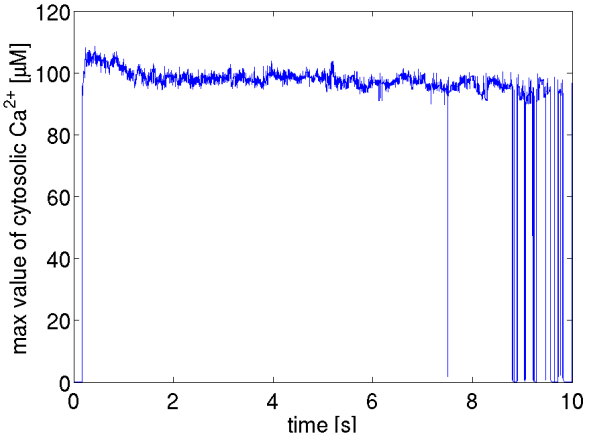
hedral elements and 113,498 nodes is considered for the numerical computations, see left hand side of Figure 3, and we used the ROWDA method as a time integrator. The computation comprises of 453,992 degrees of freedom and computed 10 s of real time.

The multi cluster setup was developed to study the bursting of calcium channels across the whole cell. First, we assume that all channels are closed before simulation starts. During the hybrid stochastic simulations, the initial opening of a channel occurs in the middle of the domain at time $t = 0.17$ s, see Figure 9(a). Here the maximum cytosolic calcium concentration rises rapidly to 105 μM, see Figure 8(b). On the other hand the average calcium concentration increases slowly based on how fast the calcium diffuses and number of open channels on a membrane. Then the released calcium diffuses to neighboring clusters slowly and it opens the few channels in a systematic way at all clusters subsequently, see rest of the figures in 9. The wave appears at the time $t = 0.83$ s, then channels are open in all 7 clusters for a while and the wave terminates after a few seconds in some of the clusters, see Figure 8(e). The corresponding number of open channels is depicted in Figure 8(f). The maximum average calcium concentration is attained after about 1 s, see Figure 8(a).

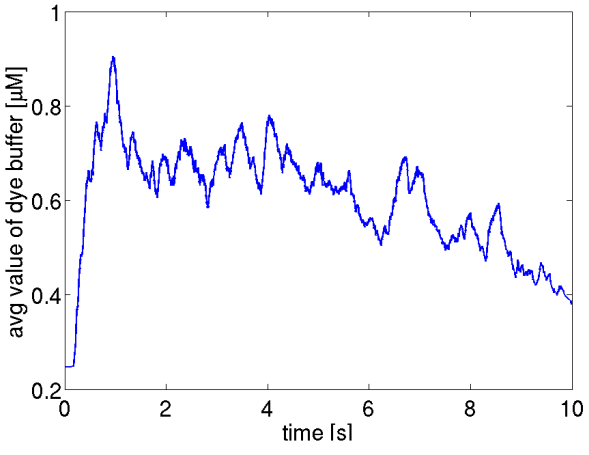
The observed behavior very much resembles the synchronized openings of channels in many clusters in oscillating cells. It should be noted that the Ca^{2+} concentration during release is large only in small domains around the open channels. Therefore overall levels of Ca^{2+} in the cell remain small (see Fig. 8(a)), but they are sufficient to elicit the synchronized release in terms of cluster-to-cluster coupling.



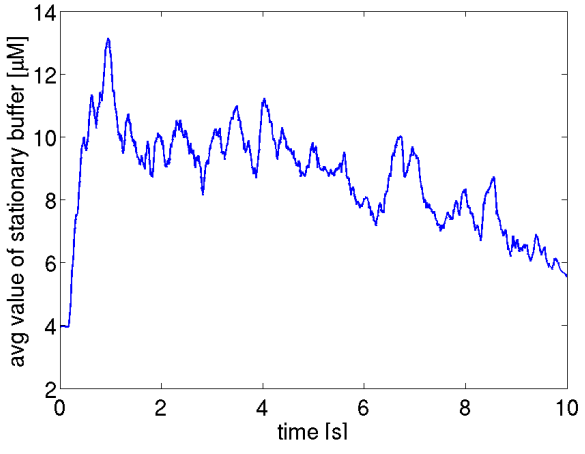
(a) cytosolic Ca^{2+} concentration



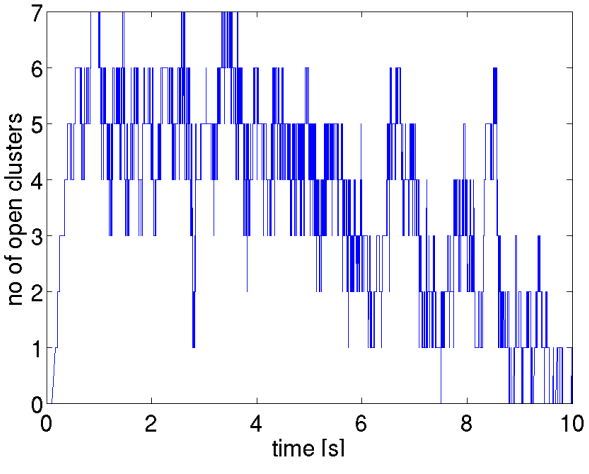
(b) maximum value of cytosolic Ca^{2+}



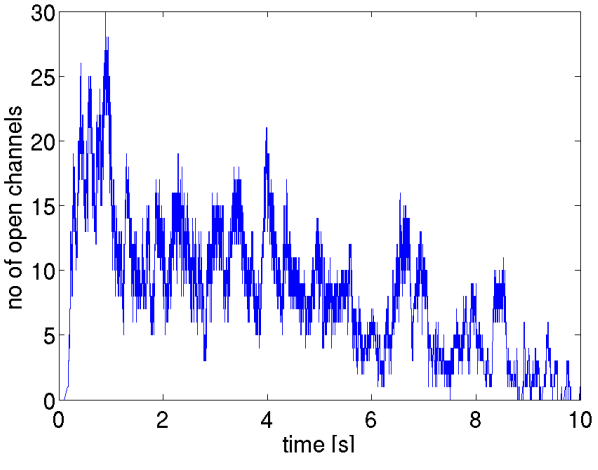
(c) dye buffer concentration



(d) stationary buffer concentration



(e) number of open clusters



(f) number of open channels

Figure 8 – The average and maximum value of concentrations and the number of open clusters and open channels for the arrangement of 7 cluster over the time is plotted.

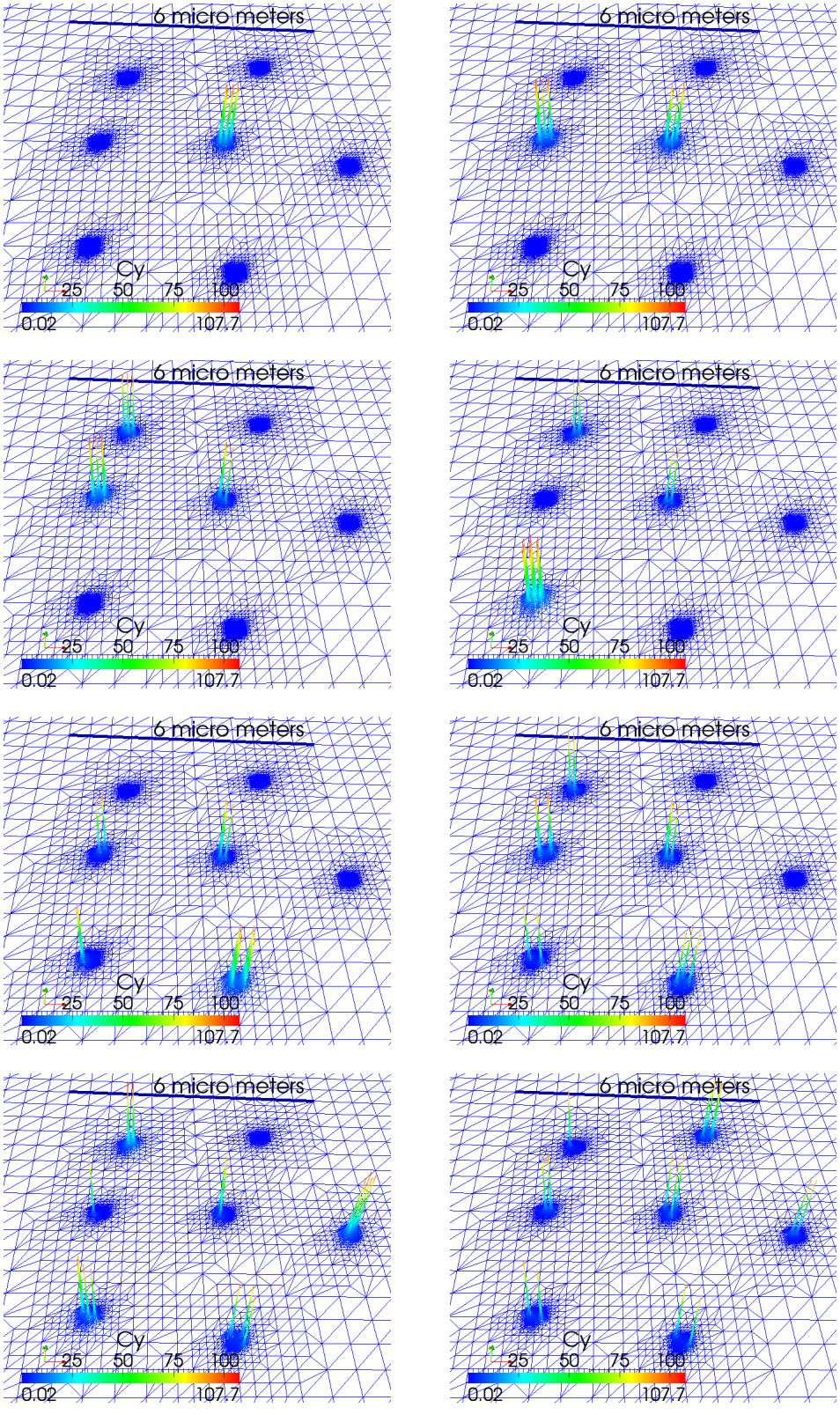


Figure 9 – Surface plot of the cytosolic calcium concentration at the bottom surface for channel transitions at times 0.1813 s, 0.2184 s, 0.2583 s, 0.2952 s, 0.3540 s, 0.4062 s, 0.5602 s and 0.830 s (from top left to bottom right).

Table 2 – Parameters used in the numerical simulations.

Parameter	Value	Unit
leak flux coefficient P_l	0.04	s^{-1}
channel flux coefficient P_{ch}	6.32×10^3	$\mu m\ s^{-1}$
single channel radius R_s	0.006	μM
pump flux coefficient P_p	400	$\mu m\ \mu M\ s^{-1}$
pump dissociation coefficient K_d	0.2	μM
diffusion coefficient D of free cytosolic Ca^{2+}	223	$\mu m^2\ s^{-1}$
diffusion coefficient D of free ER Ca^{2+}	95	$\mu m^2\ s^{-1}$
diffusion coefficient D_m of mobile buffer	20.0	$\mu m^2\ s^{-1}$
diffusion coefficient D_s of stationary buffer	0.0	$\mu m^2\ s^{-1}$
on-rates of fast buffers:		
k_s^+	50	$(\mu M\ s)^{-1}$
k_m^+	6	$(\mu M\ s)^{-1}$
k_d^+	150	$(\mu M\ s)^{-1}$
dissociation constants of buffers $k_i = \frac{k_i^-}{k_i^+}$:		
k_s	2	μM
k_m	0.1666	μM
k_d	2	μM
total concentrations of buffers:		
B_s	400	μM
B_m	0.0	μM
B_d	25.0	μM
subunit kinetics, note $b_i = a_i d_i$, i=1,. . .,5		
IP3 binding		
a_1, a_3	0.014	$(\mu M\ s)^{-1}$
d_1	0.001	μM
d_3	0.7	μM
inhibiting, with IP3		
a_2	0.02	$(\mu M\ s)^{-1}$
d_2	78	μM
inhibiting, without IP3		
a_4	0.1	$(\mu M\ s)^{-1}$
d_4	0.1114	μM
activating		
a_5	100.0	$(\mu M\ s)^{-1}$
d_5	0.25	μM

We used 16 cores for the simulation of the above 7 cluster setup and 10 seconds of simulation time. As was mentioned above, the external UG [2] grid interface is used for the grid generation and the domain decomposition. The load balance of the locally refined grid is shown in Figure 4(b). Here we observed that on 16 cores there is a 16% of load imbalance of the computational grid. Due to this load imbalance and may be smaller local computational workload, the parallel efficiency is degraded to 53% for this test case. In this regard, the parallel run took 4 days and 7 hours of computational time to finish the 10 s of simulation time.

6 DISCUSSION

In this study, we have developed and studied a finite element code for the hybrid stochastic and deterministic equations of calcium dynamics. One of the main goals of this work is to develop an efficient simulation tool to study particularly the long time behavior of the entire cell and exploit the numerical methods using highly parallel architectures for the current problem. In this regard, we simulated our 3D multi cluster setup on parallel machines where the release channels are incorporated in simulations by allowing highly unstructured and locally adapted grids. This enabled us, for the first time, to simulate the whole-cell calcium signals while at the same time resolving the spatial distribution within each cluster and around each channel at nanometer scale. Locally refined meshes are superior for representing channel release fluxes at the membrane, so that it saves enormous computational time. The convergence of numerical solution for different locally refined meshes were studied which demonstrates that these meshes are sufficient to get accurate results with small numbers of nodal points. Also we have investigated different higher order time integrators for this complex problem which is a crucial step to choose the right time step method during the channel transition in hybrid simulations. We have accomplished the parallel efficiency study for the single and 7 cluster setup in hybrid stochastic and deterministic models. We attained a good parallel scalability up to 8 cores and considerable parallel efficiency on 16 cores for these locally refined meshes. The poorer scalability of the presented results on 16 cores for the 7 cluster setup is due to several factors including solving a problem that is not large enough on 16 cores, such that communication and synchronization times are significant compared to computation times in the each iteration of linear solver and the improper load balance of the highly locally refined grid.

REFERENCES

[1] ALFONSI A, CANCES E, TURINICI G, DI VENTURA B & HUISINGA W. 2004. Exact simulation of hybrid stochastic and deterministic models for biochemical systems. INRIA Rapport de Recherche, Thèmes NUM et BIO, 5435.

[2] BASTIAN P, BIRKEN K, LANG S, JOHANNSEN K, NEUß N, RENTZ-REICHERT H & WIENERS C. 1997. UG: A flexible software toolbox for solving partial differential equations. Computing and Visualization in Science, 1: 27–40.

[3] BASTIAN P, BLATT M, DEDNER A, ENGWER C, KLÖFKORN R, KORNHUBER R, OHLBERGER M & SANDER O. 2008. A generic grid interface for parallel and adaptive scientific computing. Part II: implementation and tests in DUNE. Computing, 82(2): 121–138, July.

[4] BERRIDGE MJ. 1990. Calcium oscillations. J. Biol. Chem., 265(17): 9583–9586.

[5] BERRIDGE MJ. 1993. Inositol trisphosphate and calcium signalling. Nature, 361: 315.

[6] BERRIDGE MJ, LIPP P & BOOTMAN MD. 2000. The versatility and universality of calcium signalling. Nature Rev. Mol. Cell Biol., 1: 11–22.

[7] BEZPROZVANNY I, WATRAS J & EHRLICH BE. 1991. Bell-shaped calcium-response curves of Ins(1,4,5)P3- and calcium-gated channels from endoplasmatic reticulum of cerebellum. Nature, 351: 751–754.

[8] DEKKER K & VERWER JG. 1984. Stability of Runge Kutta methods for stiff nonlinear differential equations. North Holland Elsevier Science Publishers.

[9] DEMURO A & PARKER I. 2008. Multi-dimensional resolution of elementary Ca^{2+} signals by simultaneous multi-focal imaging. Cell Calcium, 43(4): 367–374.

[10] DeYOUNG G & KEIZER J. 1992. A single-pool inositol 1,4,5-trisphosphate-receptor-based model for agonist-stimulated oscillations in Ca^{2+} concentration. Proc. Natl. Acad. Sci. USA, 89: 9895–9899.

[11] GILLESPIE DT. 1977. Exact stochastic simulating of coupled chemical reactions. J. Phys. Chem., 81: 2340–2361.

[12] GOBBERT MK. 2008. Long-time simulations on high resolution meshes to model calcium waves in a heart cell. SIAM J. Sci. Comput., 30: 2922–2947, October.

[13] GUSTAFSSON K, LUNDH M & SÖDERLIND G. 1988. A PI step-size control for the numerical solution of ordinary differential equations. BIT, 28(2): 270–287.

[14] HANHART AL, GOBBERT MK & IZU LT. 2004. A memory-efficient finite element method for systems of reaction-diffusion equations with non-smooth forcing. J. Comput. Appl. Math., 169(2): 431–458, August.

[15] LANG J. 2001. Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems, volume 16 of Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin.

[16] LANG J & VERWER J. 2001. ROS3P – an accurate third-order Rosenbrock solver designed for parabolic problems. BIT, 41(4): 730–737.

[17] LANG J & TELEAGA D. 2008. Towards a fully space-time adaptive FEM for magnetoquasistatics. IEEE Transactions on Magnetics, 44(6): 1238–1241.

[18] MARCHANT JS & PARKER I. 2001. Role of elementary Ca^{2+} puffs in generating repetitive Ca^{2+} oscillations. The EMBO Journal, 20(1&2): 65–76.

[19] MEANS S, SMITH AJ, SHEPHERD J, SHADID J, FOWLER J, WOJCIKIEWICZ RJH, MAZEL T, SMITH GD & WILSON BS. 2006. Reaction diffusion modeling of calcium dynamics with realistic er geometry. Biophysical Journal, 91(2): 537–557.

[20] NAGAIAH Ch, RÜDIGER S, WARNECKE G & FALCKE M. 2012. Adaptive space and time numerical simulation of reactiondiffusion models for intracellular calcium dynamics. Applied Mathematics and Computation, 218(20): 10194–10210.

[21] OKADA J-i, SUGIURA S, NISHIMURA S & HISADA T. 2005. Three-dimensional simulation of calcium waves and contraction in cardiomyocytes using the finite element method. American Journal of Physiology – Cell Physiology, 288(3): C510–C522.

[22] PUTNEY JW & BIRD GSJ. 1993. The inositolphosphate-calcium signaling system in nonexcitable cells. Endocrine Reviews, 14(5): 610–631.

[23] RIDGWAY EB, GILKEY JC & JAFFE LF. 1977. Free calcium increases explosively in activating medaka eggs. Proc. Natl. Acad. Sci. USA, 74: 623–627.

[24] ROONEY TA, SASS EJ & THOMAS AP. 1989. Characterization of cytosolic calcium oscillations induced by phenylephrine and vasopressin in single fura-2-loaded hepatocytes. J. Biol. Chem., 264: 17131–17141.

[25] RÜDIGER S, JUNG P & SHUAI JW. 2012. Termination of Ca^{2+} release for clustered IP3R channels. PLoS Comput. Biol., 8(5): e1002485.

[26] RÜDIGER S, NAGAIAH Ch, WARNECKE G & SHUAI JW. 2010. Calcium domains around single and clustered IP3 receptors and their modulation by buffers. Biophysical Journal, 99(1): 3–12.

[27] RÜDIGER S, SHUAI JW, HUISINGA W, NAGAIAH Ch, WARNECKE G, PARKER I & FALCKE M. 2007. Hybrid stochastic and deterministic simulations of calcium blips. Bio-Phys. J., 93: 1847–1857.

[28] VAN DER VORST HA. 1994. Bi-CGSTAB: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput., 13: 631–644.