Modeling 3-D Calcium Waves from Stochastic Calcium sparks in a Sarcomere Using COMSOL

Zana Coulibaly[1], Leighton T. Izu*[2] and Bradford E. Peercy[1]

[1] University of Maryland, Baltimore County
[2] University of California Davis
* Corresponding author: itizu@ucdavis.edu

Abstract: This paper utilizes Comsol General Form PDE interface and Matlab to model stochastic calcium waves in a sarcomere. The model we present here shows the evolution of waves generated from calcium being released stochastically from sites modeled as point sources. The release sites are distributed on z-disc (planes) in an hexagonal pattern, and their opening allows calcium to diffuse and interact with different species of buffers. The release sites are sensitive to calcium levels and after opening and releasing calcium, undergo a refractory period during which they stay closed. The simulations obtained over a sarcomere domain shows individual stochastic releases of calcium self-organizing into propagating waves.

Key words: Calcium waves, reaction diffusion-equation, heart cell, stochastic simulation, point source, refractory period.

1 Introduction and Background

In many application problems dealing with randomly activated source terms (such as the one related to the release of calcium in cardiac cells), the process of release is defined by an inhomogeneous Poisson process. When dealing with more complicated process, where time of release also determines a refractory period after which the random process is re-enabled, implementation through Comsol alone becomes an issue as specific time points need to be kept track of. We were able to circumvent this hurdle by making Comsol interface with Matlab. The stochastic process we have implemented here has direct application in modeling organized calcium wave propagation in cardiac cells.

Organized calcium releases are the means through which the heart regulates the uniform contraction of individual cardiac cells during each heartbeat. The process of contraction starts with an electrical signal that propagates through the membrane of the cardiac cell and down T-tubules; the depolarization of the membrane causes L-type calcium sensitive channels to open and allow the flow of calcium from extracellular space to intracellular space (cytosol). The influx of calcium causes calcium sensitive channels (calcium release units or CRUs) to open and release more calcium in the cytosol. When a release from a CRU occurs, calcium diffuses through the cell, binds to contractile proteins and enables contraction to happen. Afterwards, calcium is pumped back into the sarcoplasmic reticulum. The process of release and uptake is known as calcium-induced calcium release (CICR). The intracellular release of calcium from a CRU is known as a calcium spark [1]. CICR can be modeled mathematically with a set of coupled partial differential equations.

CRUs are clusters of ryanodine receptors (RyR) and are point-like entities distributed along the plane defined by z-lines and referred to as z-discs; CRUs serve as release sites of calcium from intracellular calcium stores (sarcoplasmic reticulum) and are sensitive to the level of calcium in their proximities. After releasing calcium, CRUs enter a period of inhibition during which they stay closed and do not release calcium; only after the refractory period, do CRUs re-open stochastically and release calcium. It has been shown that under certain conditions, the release of calcium can occur without the external electrical stimulation and cause calcium sparks to self-organize into abnormal waves.

The process involved in modeling spontaneous calcium waves requires the implementation of the release mechanism of calcium and the interaction of calcium with various buffer species. This paper presents the use of COMSOL Multiphysics and Matlab to model the stochastic closing and opening mechanism of multiple CRUs. The resulting model, implemented on the domain of a sarcomere, can be used to simulate and capture calcium wave dynamics.

2 Model Description

We model the release of calcium in cardiac cells with the following system of reaction-diffusion equations taken from [2]

$$\begin{aligned}\frac{\partial c}{\partial t} &= \nabla \cdot (D_c \nabla c) - J_{\text{pump}} + J_{\text{leak}} \\ &\quad + J_{\text{release}} + \sum_i R_i(c, b_i, B_i), \end{aligned} \tag{1}$$

$$\frac{\partial b_i}{\partial t} = \nabla \cdot (D_{b_i} \nabla b_i) + R_i(c, b_i, B_i), \tag{2}$$

$$\frac{\partial B_i}{\partial t} = \nabla \cdot (D_{B_i} \nabla B_i) - R_i(c, b_i, B_i), \tag{3}$$

on a three-dimensional domain with a no-flux boundary conditions. The term c stands for calcium concentration, b_i corresponds to the concentration of free buffer species i, and B_i corresponds to the concentration of bound buffer species i. In what follows, we give more details about each term involved in Equation (1).

2.1 The pump and leak terms

The term J_{pump} represents the uptake of calcium through the sarcoplasmic reticulum (SR) and has the form

$$J_{\text{pump}}(c) = V_{\max} \frac{c^n}{K_{\text{pump}} + c^n};$$

the leak term $J_{\text{leak}} = J_{\text{pump}}(c_0)$, with $c_0 = 0.1$ μM, represents the leaking of calcium (at equilibrium) from the SR to the cytoplasm of the cell.

2.2 The buffering reaction terms

Our model uses various chemical buffering species. Each of them interact with calcium to produce a bound calcium form. Our model uses as buffers the chemical compounds Troponin, Fluo-4, ATP and their respective bound forms; hence, $i = 1, 2,$ and 3. We model these reactions of calcium with other buffers through the reaction

$$c + b_i \underset{k_i^-}{\overset{k_i^+}{\rightleftharpoons}} B_i. \tag{4}$$

From Equation (4), the rate of change of chemical species b_i only due to chemical interaction is:

$$R_i(c, b_i) = -k_i^+ c b_i + k_i^- B_i. \tag{5}$$

The rate of change of the bound form, B_i, is simply $-R_i(c, b_i)$. If we assume that at each point of space and time, we have conservation of moles so that $B_{i_T} = b_i + B_i$, and that $D_{B_i} = D_{b_i}$, the expression $R_i(c, b_i, B_i)$ simplifies to

$$R_i(c, b_i, B_i) = -k_i^+ c b_i + k_i^- (B_{i_T} - b_i),$$

where B_{i_T} is the total buffer concentration of buffer species i. Using the above assumptions, we simplify the model to equations (1) and (2).

2.3 The release term and mechanism

Each CRU is modeled as a point source that produces (releases) a molar flux σ; the term J_{release} represents this release and has the form

$$J_{\text{release}}(c, \mathbf{x}) = \sum_j \sigma(t, T_j^m) S(t; T_{open}) \delta(\mathbf{x} - \hat{\mathbf{x}}_j),$$

with j being the index on the number of CRUs, and where $\hat{\mathbf{x}}_j$ represents location of CRU j, T_j^m represents the m^{th} time CRU j opens, and $S(t; T_{open})$ is a stochastic indicator function. Essentially, $S(t; T_{open})$ determines whether or not an individual CRU, given a certain time t, is open or is in its refractory period. The stochastic indicator function has the form

$$S(t, T_{open}) = \begin{cases} 1 & \text{if } \alpha \le J_{\text{prob}}(c), \\ 0 & \text{if } \alpha > J_{\text{prob}}(c), \end{cases} \tag{6}$$

with probability

$$J_{\text{prob}}(c) = P_{\max} \frac{c^m}{K_{\text{prob}}^m + c^m},$$

where $P_{\max}$=0.3 spark/CRU/ms, $m = 1.6$, $K_{\text{prob}} = 5$ μM, and $0 \le \alpha \le 1$ is a number generated from a uniform distribution. Our model further assumes that the evaluation of $S(t, T_{\text{open}})$ is

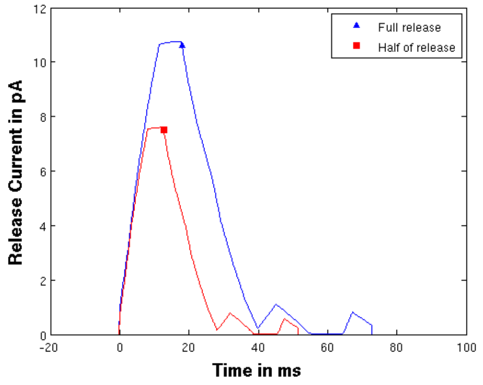


Figure 1: Release currents obtained from converting release flux σ.

done once every millisecond. The opening and closing of CRU can thus be thought of as an inhomogeneous Poisson process with rate $\lambda(c) = J_{\text{prob}}(c)$, which is calcium dependent; however this process goes dormant and undergoes a refractory period of t_{closed} after an opening has occurred. The term $\sigma(t, T^m)$ represents the release flux and is taken from [3]; Figure 1 shows various release currents obtained using the conversion $I = \sigma(t, T_j^m)/(2F)$ for a release starting at $t = 0$ ms; F is the Faraday constant.

3 Model Implementation

We are modeling the release and diffusion of calcium over a basic unit of a cardiac cell called a sarcomere (see Figure 2). The sarcomere domain consists of two z-discs, 2 μm apart, with CRUs arranged in a hexagonal pattern (thus the number of 37 per z-discs). The hexagonal pattern distribution allows CRUs on a same z-disc to be exactly $L = 0.6$ μm apart from one another. Figure 3 shows the distribution of CRUs and their indexing on a particular z-disc.

The implementation of the model is done using Comsol General PDE interface. The advantage of using Comsol for this model is that it allows flexible domain definition and advanced mesh refinements. The difficulty in our problem comes from properly implementing the release mechanism of a CRU. One needs to keep track of the release time of each individual CRU in order to determine if a particular CRU is in its refractory period; this

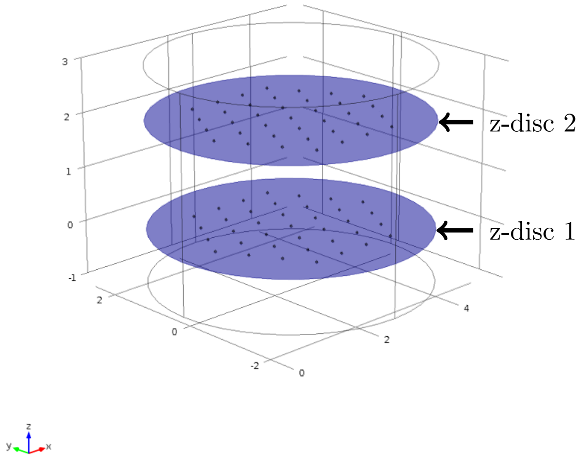


Figure 2: Sarcomere domain used in simulation. The domain consists of two z-discs and multiple CRUs.

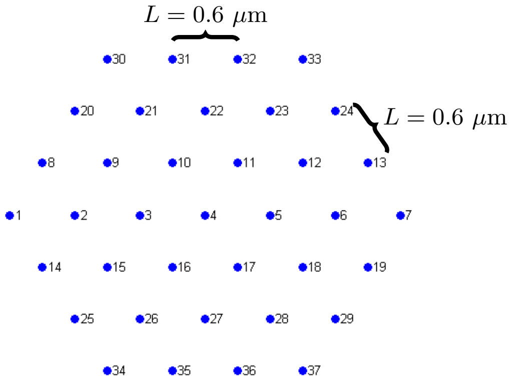


Figure 3: CRUs distribution and indexing on a z-disc. The distance between adjacent CRU is $L = 0.6$ μm.

Parameter/Variable	Description	Values/Units
c	Calcium concentration	μM
c_0	Initial calcium concentration	0.1 μM
D_c	Calcium diffusion coefficient	diag(150,150,300) μm^2/s
V_{pump}	Maximum pump strength	0.3 μMs^{-1}
J_{leak}	Leak term	$\approx$ 0.16 μMs^{-1}
n	Pump hill coefficient	4
m	Probability hill coefficient	1.6
K_{pump}	pump sensitivity	0.184 μM
K_{prob}	pump sensitivity	5 μM
σ	Source current amplitude	μmol s^{-1}
t_{closed}	CRU refractory period	100 ms

Table 1: Simulation parameters

	Name	k_n^-	k_n^+	B_{i_T}	D_{b_i}
b_1	Fluo-4	0.032	0.032	50	0.05
b_2	Troponin	0.011	5.7e-3	70	0
b_3	ATP	30	0.15	5,000	0.14

Table 2: Buffer parameters. k_n^- is in μM^{-1}ms^{-1}, k_n^+ is in (ms^{-1}), B_{i_T} is in (μM) and D_{b_i} is in (μm^2/s). The diffusion is isotropic. Values of parameters taken from [3].

task cannot be implemented in Comsol alone. Fortunately, we can tackle this problem by making Comsol interface with Matlab.

With Matlab, we are able to design a script that implements the release mechanism using the $x-, y-, z-$positions, time and concentration levels computed by Comsol. The obtained data is used to identify individual CRU and to evaluate the implementation of the firing model described by Equation (6). The script also keeps track of the opening time of each individual CRU (this task cannot be accomplished through Comsol alone); to achieve this purpose, our Matlab script uses global variables (refer to the Matlab documentation for more information on global variables). The use of global vector variables in Matlab allows us to meticulously keep track of each individual CRU opening time. In our case, global variables becomes useful as their scope allows them to be accessible both from the Matlab workspace and from within any function. Thus, during each separate function call, Matlab can access the vector of variables and determine if a CRU has fired, is firing, or is in its refractory period (is not able to fire for a certain period of time) and when a CRU has opened, Matlab updates the vector of variables with simulation information. One can also avoid the use of global variables by storing the opening times of each CRUs to a file; this method is however less efficient because it requires reading and writing to a file every time our Matlab function is called. Since a CRU is to be tested for the likelihood of opening every ms, it necessary to set the time stepping of the Comsol Time-Dependent Solver to *Intermediate* instead of the default option being *free*; this assures that Comsol solver will take the step size necessary to compute the solution every millisecond instead of taking large time-steps and potentially skip the millisecond mark at which we test for a CRU potential opening. Using these options for the Time-Dependent solver, we ensure that our Matlab script will be called at appropriate times (every millisecond) and evaluate the firing eligibility of CRU at those times. For more details on Comsol Time-Dependent Solver options, refer to the Comsol documentation.

In a non-linear and discontinuous problem such as ours, one needs to pay close attention to both the discretization of the domain and the ability of the solver to overcome time discrete events such as CRU firing. For our specific problem, we choose to manually refine the mesh around the location of release sites and set the solver to update the Jacobian after every iteration and set the maximum time-step to 0.5 ms. By choosing options as such, we force our time step to be very small ($\leq 10^{-8}$

ms) when there is a release of calcium. We ensure convergence by showing further refinements of the mesh of the mesh, for the same random seed, does not yield different simulation results. which simulation runs, for the same random seed, do not yield different outcomes.

In some rare cases (such as when the magnitude of the release is large), the solution returned by Comsol can become negative; this obviously should not happen as the solution of our problem, given the model statement, is to be positive. Negative solutions happen due to accumulation of very small errors; this case can happen when the concentration of the buffer species becomes in the proximity of the release site becomes very low (close to zero). To remedy this problem, we make the change of variables $c = e^w$ and $b_i = e^{w_i}$, and $B_i = e^{W_i}$ or $w = \log(c)$ and $w_i = \log(b_i)$ and obtain the equations

$$\begin{aligned} e^w \frac{\partial w}{\partial t} &= \nabla \cdot (e^w D_c \nabla w) + F(e^w, e^{w_i}, e^{W_i}), \\ e^{w_i} \frac{\partial w_i}{\partial t} &= \nabla \cdot (e^{w_i} D_{b_i} \nabla w_i) + R_i(e^{w_i}, e^{W_i}), \\ e^{W_i} \frac{\partial W_i}{\partial t} &= \nabla \cdot \left(e^{W_i} D_{B_i} \nabla W_i\right) - R_i(e^{w_i}, e^{W_i}), \end{aligned} \tag{7}$$

with initial conditions $w_0 = \log(c_0)$ and $w_{i_0} = \log(b_{i_0})$, $W_{i_0} = \log(B_{i_o})$ and where $F(e^w, e^{w_i}, e^{W_i})$ represents the leak, pump, release and reaction terms of Equations (1), (2) and (3) with the proper substitution. Alternatively, It is possible when concentrations reach a very small value to set them to zero; however, one can run into the risk of introducing artifacts in the solutions.

4 Simulation Results

We run simulations with parameters shown in Tables 1 and 2 over a period of $1,000$ ms. The simulations were ran on a 64-bit quadruple core AMD processor machine (3.20 GHz) with 8.00 GB of RAM. Figure 4 summarizes the result of a run obtained from using half a release strength; calcium concentration is plotted on each z-discs and critical isolevel of 5 μM are shown. The simulation resulted in a first spark appearing at around $t = 540$ ms; at time $t = 570$ ms, the CRU that opened at $t = 540$ ms has entered its refractory period and we see three other sparks at different CRU locations occurring. As time evolves, multiple other sparks start occurring and finally organize into a calcium wave. It took Comsol roughly 12 hours to complete the simulation.

We have also run a simulation where we initiate the release of calcium at $t = 1$ ms from seven CRUs located in the center hexagonal of z-disc 1. The result of the run is summarized in Figure 5. We see that the combined massive release of calcium causes neighboring CRUs on z-disc 1 to open and release calcium. In this particular case, the sparks organize into a wave within 70 ms. This simulation took Comsol roughly 18 hours.

The long simulation times are due to the non-linear aspect of the model (multiple point sources), the level at which the mesh is refined, and the fact that CRUs are active and their likelihood of opening needs to be checked every milliseconds.

5 Conclusion

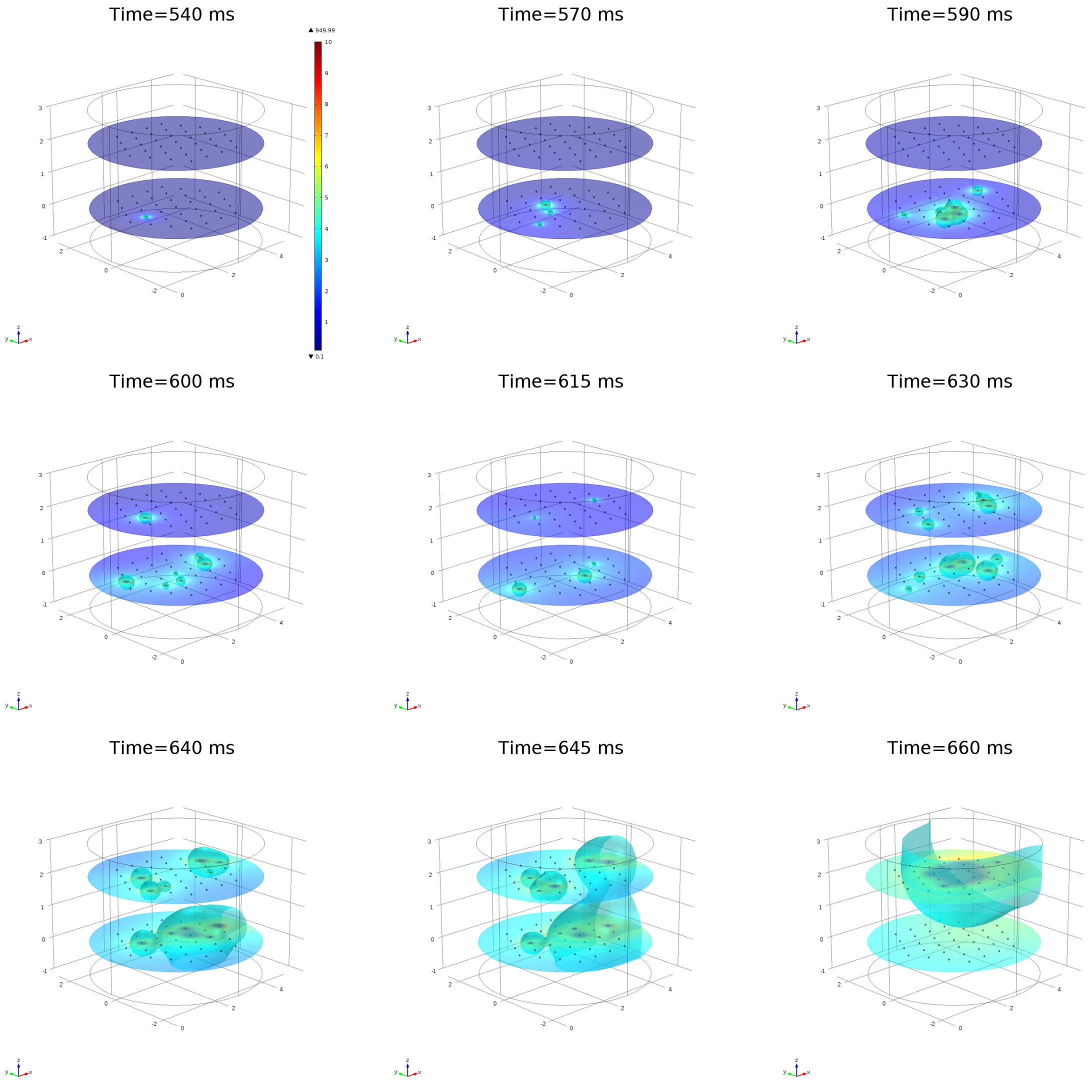
In this paper we have shown how using Comsol and Matlab, we can implement sophisticated stochastic simulations. This implementation allowed us to model calcium waves in a sarcomere. There is a possibility to expand the implementation to a whole cell, but the memory and computation time required to achieve such a purpose increases. The implementation presented here can be expanded to any kind of stochastic-based mathematical model where refractory and/or inhibition periods are assumed. Furthermore, we have shown that Comsol can handle non-linear problems and time discontinuous problems.

References

[1] Heping Cheng and W. J. Lederer. Calcium sparks. *Physiol. Rev.*, vol. 88, no. 4, pp. 1491–1545, 2008.

[2] Leighton T. Izu, W. Gil Wier, and C. William Balke. Evolution of cardiac calcium waves from stochastic calcium sparks. *Biophys. J.*, vol. 80, pp. 103–120, 2001.

[3] Vyacheslav M. Shkryl, Lothar A. Blatter, and Eduardo Rios. Properties of Ca^{2+} sparks revealed by four-dimensional confocal imaging of cardiac muscle. *JOURNAL OF GENERAL PHYSIOLOGY*, vol. 139, pp. 189–207, MAR 2012.



Time=540 ms

Time=570 ms

Time=590 ms

Time=600 ms

Time=615 ms

Time=630 ms

Time=640 ms

Time=645 ms

Time=660 ms

Figure 4: Simulation obtained from stochastic release of calcium at half release strength. Calcium concentration is plotted on each z-disc and the critical calcium isolevel of 5 μM is shown throughout the sarcomere domain. Stochastic release of calcium generates a self-organized wave that propagates through the sarcomere within 660 ms.

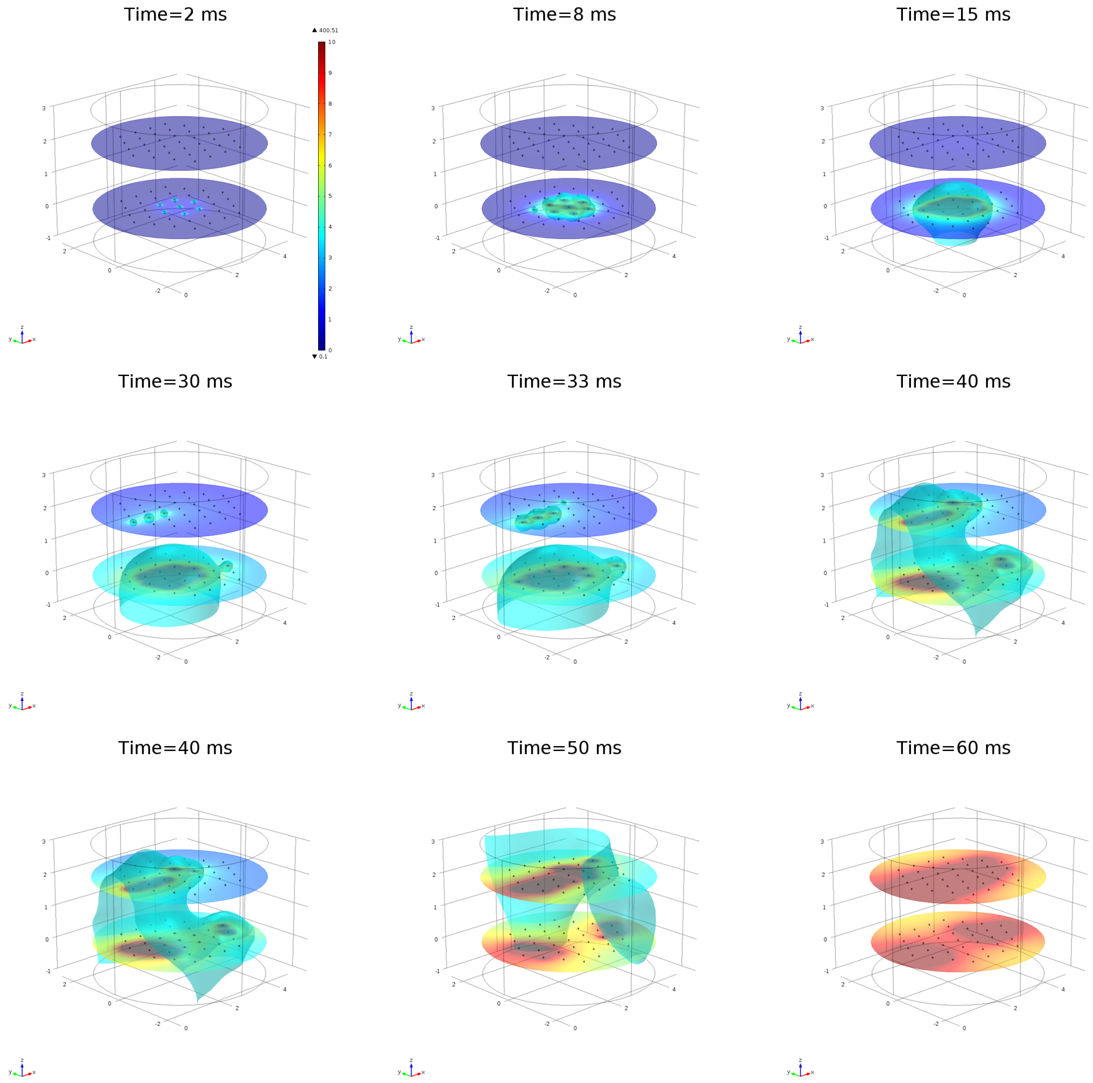


Figure 5: Simulation obtained from forcing the opening of 7 CRUs located in the center hexagonal of z-disc 1. Calcium concentration is plotted on each z-disc and critical calcium isolevel of 5 μM is shown throughout the whole domain. Stochastic release of calcium at half release strength generates a self-organized wave that propagates through the sarcomere within 60 ms.