



NOVA

University of Newcastle Research Online

nova.newcastle.edu.au

Stoddard, Jeremy G.; Welsh, James S.; Laver, Derek R. 'A parallel modelling algorithm for simulating calcium release in cells'. Published in Proceedings of the 6th IFAC Conference on Foundations of Systems Biology in Engineering [presented in IFAC Papers Online, Vol. 49 No. 26] (Magdeburg, Germany 9-12 October, 2016) p. 83-88 (2016)

**Available from:** <http://dx.doi.org/10.1016/j.ifacol.2016.12.107>

©2016 This manuscript version is made available under the CC-BY-NC-ND 4.0 license  
<http://creativecommons.org/licenses/by-nc-nd/4.0>

**Accessed from:** <http://hdl.handle.net/1959.13/1335994>

# A Parallel Modelling Algorithm for Simulating Calcium Release in Cells

Jeremy G. Stoddard\* James S. Welsh\* Derek R. Laver\*\*

\* *School of Electrical Engineering and Computer Science, The University of Newcastle, Australia,  
(email: jeremy.stoddard@uon.edu.au)*

\*\* *School of Biomedical Sciences and Pharmacy, The University of Newcastle, Australia*

---

## Abstract:

Using a spatially discretised model structure to represent the behaviour of calcium release sites in a cell, this paper presents a parallel solution algorithm which treats each release site as an independent sub-system, and manages inter-site data communication on a global timestep. When compared to the equivalent single-thread solution algorithm, the parallel method features a negligible reduction in accuracy, and improves computation time scaling from a quadratic,  $O(n^2)$  to a linear,  $O(n)$ , with respect to the number of release sites,  $n$ , in the model.

*Keywords:* Parallel, Modelling, Biology, Simulation

---

## 1. INTRODUCTION

Normal heart function relies on coordinated, rhythmic contractions of billions of muscle cells within the heart. In turn, rhythmic contraction of each cell depends on release of calcium ions ( $Ca^{2+}$ ) from approximately 20,000 release sites distributed throughout each cell. Parallel computing provides a tool for understanding how the physiological process within each release site and within each cell integrate to produce the observable heart rhythm.

A non-linear model of calcium release in heart cells, developed by Cannell et al. (2013a), considers mechanisms of buffering, diffusion and stochastic triggering for a small population of release sites inside a single cell. For large-scale simulations, the model has been discretised to decrease the structural complexity at each release site.

The intended purpose of the model is to understand how groups of  $Ca^{2+}$  release sites produce the experimentally observable phenomena of  $Ca^{2+}$  sparks and waves. Sparks, named for the bursts of light they produce in the presence of fluorescent indicators, are localised  $Ca^{2+}$  release events that occur at a single site (Kong et al., 2013).  $Ca^{2+}$  waves are release events which require the coordinated activation of adjacent release sites, and produce an increase in cytoplasmic calcium concentration which propagates like a wave throughout the cell. The physiological details of the wave phenomenon remain unclear, and a better understanding would enable further studies of the effect of various pharmaceutical drugs on wave properties and ultimately on heart rhythm.

With all major dynamics considered, the resulting model is stiff in nature, and requires a sophisticated solver algorithm to minimise computation time and ensure accuracy. This model had been solved as a single-thread, single-Jacobian process, causing simulations to become prohibitively long for models on the order of 100 sites,

while a full-scale 20,000-site cell model results in a system order greater than  $10^6$ . To enable such large scale simulations, finding a more time-efficient modelling approach is necessary such that the full-scale model can be explored, modified and interpreted in more convenient time frames.

Parallel model structures have been successfully developed to simulate various diffusion-based processes which are similar in nature to the calcium release problem. The results from Shaikh et al. (2011), Stern et al. (2014) and Li et al. (2010) show the potential of parallel methods to provide accurate results with reasonable computation times if the model is partitioned in a manner which exploits the different time constants of the system.

In this paper, the biological model is split into independent subsystems based on the varying diffusion dynamics in different regions of the system. A parallel solution algorithm is formulated which solves the individual subsystems in a multi-thread environment, with a communication update interval maintained on a global timestep. Several extrapolation methods are tested within the parallel framework to identify the most accurate and robust technique. The computation time of the new algorithm scales linearly  $O(n)$  with the number of release sites,  $n$ , in the model. Further improvements are also achieved through the treatment of the Jacobian and global step-size control.

The paper is organised as follows: Section II provides an overview of the discretised release model relevant to large-scale simulations, and highlights the dynamics present in a single release site. Section III presents the several competing methods of parallelisation and assesses their accuracy in the calcium release context. Section IV theoretically and experimentally validates the linearization of solution time scaling. Section V outlines some additional numerical techniques which can further optimise computation time, and Section VI concludes the paper.

## 2. THE CALCIUM RELEASE MODEL

### 2.1 Geometry

In heart muscle cells, the release of calcium into the general cytoplasm is dictated by the behaviour of dyads located throughout the cell's structure. The dyad consists of several components (Cannell et al., 2013a):

- (1) T-tubule which allows extracellular calcium to enter the cell at the dyad junction.
- (2) Distributed calcium store (LSR) inside the cell, with a terminal (TSR) located in close proximity to the ion channel of the T-tubule, creating a 'dyadic cleft'.
- (3) Cluster of calcium release channels (ryanodine receptors or 'RyRs') on the surface of the TSR, which open and close to control the diffusion of calcium from the store into the cleft and general cytoplasm of the cell.

In the simplified model, each dyad is contained within a cube shaped 'voxel' which contains discrete elements relevant to the release process. Each element will possess a lumped concentration for  $Ca^{2+}$ , as well as concentrations for a selection of buffers and indicators. The number of state variables in a single voxel is of the order of  $10^2$ . Figure 1 provides a conceptual diagram of the geometry for a discretised voxel, and indicates the typical flow of calcium within and between release sites.

### 2.2 Transport of Materials

In modelling the concentrations of substances in the cell, mechanisms of diffusion, buffering and active pumping are considered. With the model split into discrete blocks of cytoplasm, the equations governing these transport mechanisms must also be discretised around the defined volume elements. For the case of calcium, the concentration in a particular volume of cytoplasm is given by (Cannell et al., 2013a)

$$\frac{d[Ca^{2+}]}{dt} = \sum_{i=1}^p D_i([Ca_i^{2+}] - [Ca^{2+}]) + \sum_{i=1}^r F_{RyR,i} - F_{SERCA} - \sum_{i=1}^b F_{buffer,i}, \quad (1)$$

where  $[Ca^{2+}]$  is the calcium concentration in the volume. On the RHS of (1), the first term describes diffusion from  $p$  neighbouring volume elements, with their own calcium

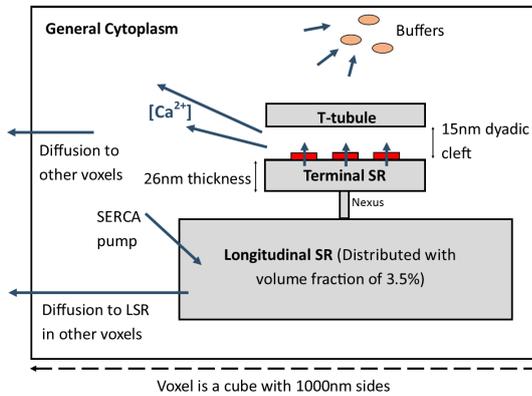


Fig. 1. Voxel geometry for a single dyad,  $Ca^{2+}$  release site

concentrations  $[Ca_i^{2+}]$ , and diffusion coefficients  $D_i$  to the volume of interest. The second term is an expression for the calcium fluxes,  $F_{RyR,i}$ , from  $r$  open receptors, which is applicable for volumes inside the dyad cleft. In some volume elements, a 'SERCA pump' is present which actively transfers calcium from the general cytoplasm to the longitudinal SR, giving rise to the  $F_{SERCA}$  flux. Lastly, there are several other substances which bind to calcium, removing the free ions from solution. These buffering reactions are summarised by  $F_{buffer,i}$  fluxes to each of the  $b$  buffers present in the volume.

The buffering fluxes have a linear form, and for each buffer  $B_i$  in the volume, the flux is defined by

$$F_{buffer,i} = k_{on,i}[B_i][Ca^{2+}] - k_{off,i}[CaB_i],$$

where  $k_{on,i}$  and  $k_{off,i}$  are buffer-specific constants,  $[B_i]$  is the concentration of the unbound buffer in the volume, and  $[CaB_i]$  is the concentration of the calcium-bound buffer. The SERCA flux out of the general cytoplasm is governed by the non-linear equation,

$$F_{SERCA} = k_1 \left( \frac{k_2[Ca^{2+}]^2}{1 + k_2[Ca^{2+}]^2} - k_3 \right),$$

for some fixed constants  $k_1$ ,  $k_2$  and  $k_3$ . The RyR fluxes further increase the model complexity, as they depend on the open probabilities of receptors in the area,

$$F_{RyR,i} = k_4 P_{RyR,i} ([Ca_{TSR}^{2+}] - [Ca^{2+}]),$$

where the calcium concentration of the terminal store (TSR) is given by  $[Ca_{TSR}^{2+}]$ ,  $k_4$  is a constant, and  $P_{RyR,i}$  is the open probability of the  $i$ 'th receptor in the volume. The open probabilities are themselves dependent on a stochastic triggering process which is executed at 50ms intervals. A more comprehensive mathematical description of the model can be obtained from Cannell et al. (2013a), and its supplement (Cannell et al., 2013b).

### 2.3 Release Site Dynamics

Ideally, in order to parallelise the release model, the boundaries for the proposed sub-systems should have a slow rate of information exchange with respect to the dynamics internal to the sub-systems. An investigation of the 'spark' dynamics at a single release site was performed to provide evidence that the voxels themselves are viable candidates for conversion to parallel subsystems.

A spark begins when the dyad is triggered and receptors latch open, allowing  $Ca^{2+}$  to flood out from the TSR into the junction before diffusing to the general cytoplasm and neighbouring release sites. After a brief period, the  $Ca^{2+}$  concentration in the TSR reduces to a level that cannot continue supporting flux into the dyadic cleft, hence the receptors begin re-closing (Cannell et al., 2013a), (Kong et al., 2013). Figure 2 shows the calcium concentrations for several important volumes during a release site spark. It can be seen in plot A that a sharp concentration spike occurs at the centre of the cleft, where calcium floods out of the store through the receptors. The TSR concentration (plot B) has a corresponding sharp decrease as its calcium store depletes. The general cytoplasm (plot C) has a significantly slower calcium influx than the dyadic cleft, and the relative magnitude of the increase is far smaller. The dominant behaviour in the LSR (plot D) is a slow rise in concentration driven by the SERCA pump.

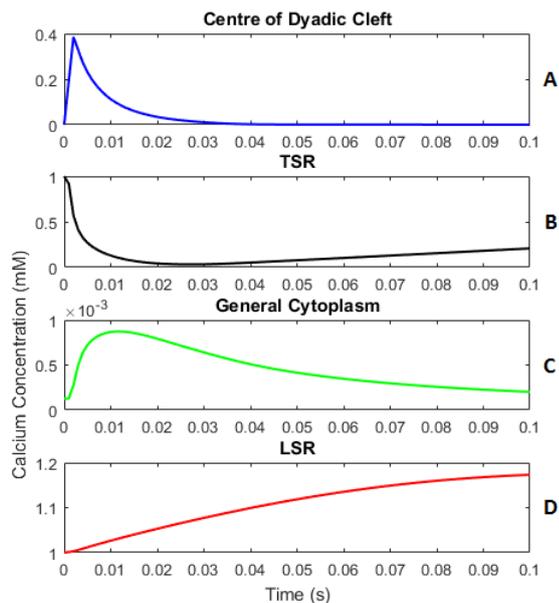


Fig. 2. Spark dynamics in modelled volume elements

The  $Ca^{2+}$  spark simulation reveals that the calcium dynamics in volumes connected to the boundary of the voxel (general cytoplasm and LSR) are significantly slower than some of the dynamics at the centre of the voxel in the dyadic cleft. This result highlights the potential for model-splitting at the voxel edges, such that each release site is an independent subsystem.

### 3. METHODS OF PARALLELISATION

The previous section provided evidence that the calcium release model could be partitioned into independent voxels and solved in parallel, however, several problems remain with respect to the implementation of a parallel solution scheme:

- Choosing the interval on which communication updates occur to exchange information between sites.
- Treatment of the diffusion fluxes from neighbouring release sites in the presence of outdated information.

A reasonable solution to the first problem comes from analysing the diffusion time constants for substances which cross the border between voxels. The smallest time constant occurs for calcium in the general cytoplasm; approximately 3ms for diffusion between release sites. This time constant can be used to inform a choice of ‘global step-size’ or timestep for communication updates, using the rule of thumb that the step size be at least 4 times smaller than the smallest time constant (Kulakowski et al., 2007).

The treatment of inter-site fluxes between communication updates warrants some further investigation, and several numerical methods were developed for application in the parallel-split model of calcium release.

#### 3.1 Constant Flux Method

The most computationally conservative approach would be to hold all fluxes into and out of the subsystem constant

until new fluxes can be computed at the next update interval, i.e. for some substance  $X$  in a boundary volume, the flux from the volumes of neighbouring sites is given by

$$F_{[X],ext}(t) = \sum_{i=1}^p D_i([X_i]_{t_0} - [X]_{t_0}) \forall t \in (t_0, t_1), \quad (2)$$

where  $t_0$  and  $t_1$  are the times of two consecutive communication updates. For  $p$  neighbouring sites,  $D_i$  is the diffusion constant from the  $i$ 'th neighbouring volume, and  $[X_i]_{t_0}$  and  $[X]_{t_0}$  are the concentrations at the last update interval for the  $i$ 'th external neighbour and volume of interest respectively.

#### 3.2 Dynamic Flux Methods

It is evident in (2) that the constant flux method does not take full advantage of the information available to a subsystem following a communication update. Firstly, the substance concentration in the volume of interest has a live value which can be accessed during the solver interval. Furthermore, the subsystem can be sent information on past values of neighbour concentrations in order to formulate reasonable predictions of their concentrations in the current interval. Using these additional assumptions, a new expression for the inter-site flux can be written as

$$F_{[X],ext}(t) = \sum_{i=1}^p D_i(P_{[X_i]}(\mathbf{a}, t) - [X](t)) \forall t \in (t_0, t_1), \quad (3)$$

where  $P_{[X_i]}$  is a function which predicts the  $i$ 'th neighbour's concentration at any time  $t$  based on parameter vector  $\mathbf{a}$  computed at  $t_0$  during the last communication update. More specifically, the prediction can be based on  $n$ 'th order polynomial extrapolations using the last  $n + 1$  concentration values up to and including  $[X_i]_{t_0}$ . It is clear that dynamic flux methods have higher computational requirements, require more memory, and increase communication time between release sites at each update interval, however, this should be offset by increased accuracy and stability in parallel solutions.

#### 3.3 Implementation

The original model code, developed prior to this research, was established in the MATLAB environment. Consequently, the parallelisation methods were also tested in MATLAB (2015b). Both the split and unsplit models made use of the ‘ode15s’ stiff solver routine, with 2<sup>nd</sup> order numerical differentiation formulas (NDFs). To enable parallel computation, the parallel jobs were dynamically allocated to a number of physical CPU cores on a 6-core 3.50 GHz Intel Xeon processor, using MATLAB's ‘parpool’ and ‘parfor’ commands.

#### 3.4 Accuracy Analysis

To assess the relative accuracy of the constant and dynamic flux methods, a small 8-site release model was simulated for 50ms with a single release site triggered. This allows calcium to flow into the cytoplasm of the triggered site and diffuse to the 7 neighbours. For the dynamic flux methods, three polynomial predictors were evaluated; constant, linear and 2nd order concentration predictions. The simulation was repeated for each parallel method using

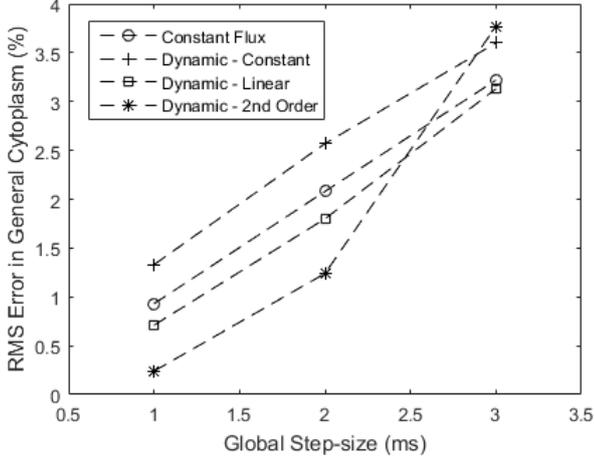


Fig. 3. RMS errors for parallelisation methods

several global timesteps, where the RMS errors in calcium concentration were examined in the general cytoplasm of each release site. The error was calculated with respect to a single-thread, single-jacobian model output for the same 8-site system.

Figure 3 shows the RMS error in sites directly neighbouring the triggered site. For small global timesteps, the 2nd order dynamic flux method is most accurate, however the accuracy decreases rapidly as step-size increases. The other dynamic and constant flux methods are more robust with respect to step-size, with linear prediction providing the highest accuracy. The other sites in the model exhibited similar error trends, and a dynamic flux method with linear prediction was chosen for use in all further simulations.

#### 4. COMPUTATION TIME

The calcium release model considered in this paper is constructed from a 3-dimensional network of interconnected voxels, and can have arbitrary size depending on the number of voxels chosen for the length of each dimension. Section I stated the requirement for 20000 release sites in a realistic cell model, but simulating such a large network of voxels as a single (stiff) ODE system would put immense stress on the solver algorithm, which must typically generate and manipulate a Jacobian of the system at each iteration of the solver (Jackson, 1996). Since the size of the Jacobian scales quadratically with system order, it is sensible to assume that the solution time will also scale quadratically with the number of release sites in the model.

If we consider an arbitrary stiff solver method with some set-up overhead, the 50ms solution time for an  $n$ -site unsplit model can be written as

$$t_n^{unsplit} = an + bn^2 \quad a, b \in \mathbb{R}^+, \quad (4)$$

where coefficients  $a$  and  $b$  describe the linear overhead and quadratic scaling of the main algorithm respectively. Note that the computation time required for 50ms of solution is important because, as explained in Section III, the entire solution process is paused at every 50ms interval for a stochastic triggering process.

If we now examine a site-split model, there are  $n$  sites which can be solved independently with separate solver

calls. We must call the solver more regularly, say  $x$  times in every 50ms window, with a communication update between each solver interval. We can assume for simplicity that the complexity of solutions remains constant between sites and throughout the 50ms window. It follows that

$$t_1^{split} = a + b/x,$$

where  $t_1^{split}$  is the computation time required to solve one site in one of the  $x$  sub-intervals. If the  $n$  release sites are split evenly among a number of computing cores,  $n_{cores}$ , the computation time for a full 50ms of solution can be expressed as

$$t_n^{split} = \frac{n}{n_{cores}} \cdot x \cdot t_1^{split} + x \cdot t_n^{update} + x \cdot t_{comms}, \quad (5)$$

where  $t_n^{update}$  is the time required to compute polynomial parameters for concentration predictions in the dynamic flux method, and  $t_{comms}$  is the time required to communicate these updates to the cores at the end of each sub-interval. The polynomial parameters are obtained for a set number of variables per site, hence,

$$t_n^{update} = cn \quad c \in \mathbb{R}^+.$$

Substituting  $t_1^{split}$  and  $t_n^{update}$  into (5) and rearranging gives the final expression,

$$t_n^{split} = \left( \frac{ax + b}{n_{cores}} + cx \right) n + x \cdot t_{comms}. \quad (6)$$

Assuming  $t_{comms}$  is negligible, the computation time scales linearly with the number of release sites in the model, since the Jacobians are now limited to the system order of one site. The gradient of the trend can be reduced by increasing the number of cores (up to a limit of  $n$ ), or by increasing the global timestep (i.e. reducing  $x$ ). There are strict limitations on the second strategy, however, since the global timestep will significantly affect accuracy and stability of parallel solutions.

The expressions derived in (4) and (6) were validated experimentally by measuring the computation time required for unsplit and parallel-split models of varying scales over a 50ms solution window. Figure 4 plots the results.

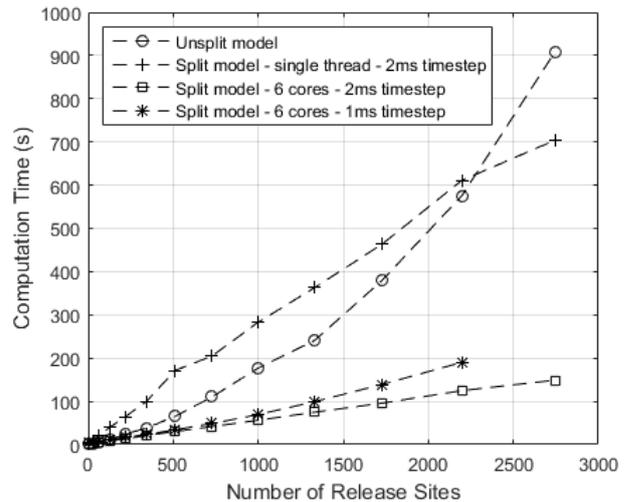


Fig. 4. Computation times for unsplit and split models

## 5. OPTIMISING COMPUTATION TIME

### 5.1 Jacobian Evaluation

Stiff ODE solver methods are typically defined recursively, and use Newton iteration to converge to the solution at each timestep of the solver (Jackson, 1996). Performing these iterations requires repeated evaluation of the ODE system's Jacobian, i.e. for an  $N$ 'th order system defined vectorially as

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}) = \mathbf{f}(y_1, y_2, \dots, y_N) = \begin{bmatrix} f_1(y_1, y_2, \dots, y_N) \\ f_2(y_1, y_2, \dots, y_N) \\ \vdots \\ f_N(y_1, y_2, \dots, y_N) \end{bmatrix},$$

evaluation of the Jacobian  $d\mathbf{f}/d\mathbf{y}$  is required for some values of  $\mathbf{y}$ . One basic strategy for computing this matrix is to apply the finite difference formula for each element of the Jacobian (Curtis et al., 1974). The  $(i, k)$ 'th element,  $\partial f_i / \partial y_k$ , is computed as

$$\frac{f_i(y_1, \dots, y_k + \delta, \dots, y_N) - f_i(y_1, \dots, y_k, \dots, y_N)}{\delta},$$

where  $\delta$  is an arbitrarily small constant (Olver, 2014). For large ODE systems, the method can become quite computationally intensive, as the ODE function must be evaluated multiple times for each element of the  $N \times N$  Jacobian. If the sparsity of the Jacobian is known, the method can be made somewhat faster by evaluating only those entries which are likely to be non-zero (Kelley, 2003). However, an analytical approach to Jacobian evaluation should prove the most time efficient, as the sparsity of the matrix can be exploited while also avoiding large numbers of ODE function evaluations.

In the parallel-split calcium release model, each release site is solved individually. All release sites have a generic structure, and are effectively identical, such that any method which can evaluate one site's system Jacobian will be sufficient for application to every subsystem. Furthermore, the release site Jacobians are significantly sparse (93% zeros), due to the diffusion-based nature of the model.

To investigate decreasing computation time, the analytical derivatives for a release site voxel were introduced to the solution code. For a voxel model with  $q$  state variables, the use of analytical expressions was optimised by splitting the Jacobian contributions into three distinct  $q \times q$  matrices:

- (1)  $\mathbf{J}_{constant}$  - Jacobian contributions which remain constant for all time. This matrix is calculated only once, at the start of simulation.
- (2)  $\mathbf{J}_{site}$  - Jacobian contributions which require knowledge of the location of the release site. The matrix is calculated at the beginning of each solver call for a given release site.
- (3)  $\mathbf{J}_{live}$  - Contributions which change dynamically with the system variables in the voxel. This matrix must be re-evaluated at every iteration of the solver.

The final system Jacobian provided to the solver method is simply the addition of these three matrices,

$$\frac{d\mathbf{f}}{d\mathbf{y}} = \mathbf{J}_{constant} + \mathbf{J}_{site} + \mathbf{J}_{live}.$$

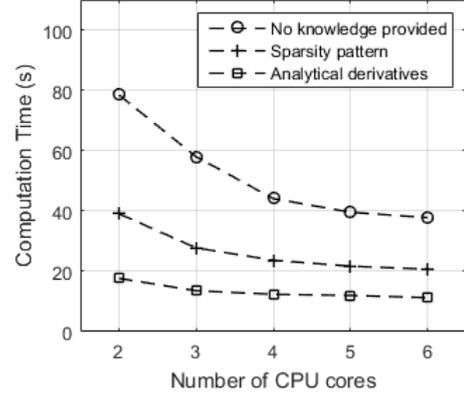


Fig. 5. Varying the solver's knowledge of the Jacobian

To compare the performance of the finite difference and analytical methods, a 125-site calcium release model was simulated with varying levels of Jacobian knowledge provided to the stiff solver. In Figure 5, the results show significant speedup when the analytical derivatives are supplied to the solver method.

### 5.2 Global Step-size Control

To choose the global step-size, it is desirable to have a timestep which is the maximum safe value for avoiding instability and maintaining solution accuracy. If step size could be dynamically chosen at any given point in the simulation, then such a method could take advantage of the time-variable dynamics in the release model. For example, when the model is 'inactive', there is minimal transport of materials between sites, and global step size can be increased without risk of instability. When a release site triggers, however, a portion of the model will experience relatively fast diffusion of materials between sites, requiring a reduction in the global time step.

A simple step size control algorithm can be constructed using only two possible global timesteps;  $t_{high}$  and  $t_{low}$ . At the end of each solver interval, the step size for the next interval is chosen based on current activity levels in the model. The activity level is quantified by inspecting the open probabilities of RyRs in each site's dyad, and finding the maximum probability for the entire model. Recall from Section II that the flow of calcium from the store into the dyadic cleft through an RyR is proportional to the receptor's open probability. If the model-wide maximum for open probability is labelled  $P_{max}$ , then the 'two-state' algorithm is given by

$$t_g = \begin{cases} t_{high} & \text{if } P_{max} < P_{thresh} \\ t_{low} & \text{otherwise} \end{cases}, \quad (7)$$

where  $t_g$  is the global step-size and  $P_{thresh}$  is some threshold between 0 and 1. A sensible value for  $P_{thresh}$  can be chosen by observing the path of  $P_{max}$  during a standard model simulation, shown in Figure 6. The periodic nature of the variable is a result of the stochastic triggering events which are initiated every 50ms. Between triggering events, open probabilities are observed to stay universally low for a significant period of time, and a threshold value between 0.1 and 0.4 would capture these low activity periods. Note that the exceptions to this periodicity, located at around

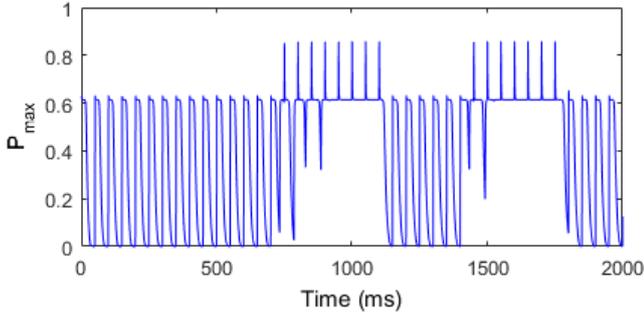


Fig. 6.  $P_{max}$  during a 450-site simulation

Table 1. Performance gains for global step-size control at two model scales

Model Scale (sites)	$P_{thresh}$	Average Computation Time (s)	Average Speedup (%)
80	No step-size control	312	-
	0.1	260	16.7
	0.2	250	19.9
	0.3	250	19.9
	0.4	248	20.5
1200	No step-size control	3788	-
	0.1	3432	9.4
	0.2	3373	10.9
	0.3	3311	12.6
	0.4	3247	14.3

1000 and 1600ms, are due to waves of calcium which are propagating through the cell.

To test the potential for computation time improvements using step-size control, the calcium release model was simulated for 2 seconds at various scales, with a number of  $P_{thresh}$  values at each scale. Due to the stochastic nature of the model, a Monte Carlo study was conducted where each simulation was repeated five times at each setting to obtain an average result. The two step-sizes,  $t_{low}$  and  $t_{high}$ , were chosen as 1ms and 2ms respectively. Results from the simulations are provided in Table 1.

At both model scales, the average speedup increased as the probability threshold for switching increased. While all simulations remained stable and maintained accuracy, there are significant risks in setting the switching threshold too high. Thresholds which are too large will make the model susceptible to numerical instability, since high RyR activity implies significant diffusion fluxes between sites.

The scale of the model was also seen to affect the size of computation speedup afforded by a two-state control algorithm. At the smallest scale, the highest average speedup was 20.5%, as compared to 14.3% at the largest scale. Extending the trend to a full-scale 20,000-site model would imply only a moderate speedup from global step size control. The relationship between scale and speedup is a product of using a model-wide maximum as the decision variable. As the scale of the model increases, the likelihood of any one RyR remaining active becomes larger during a low activity period.

## 6. CONCLUSION

In this paper, a scalable calcium release model was restructured into a network of independent subsystems, and

a parallel solution algorithm was proposed for application to the new model structure. An analysis of accuracy losses found that treating inter-site fluxes as dynamic, with a linear prediction of neighbouring concentrations, produced a method which is both accurate and robust with respect to global step-size changes.

The parallel solution method was found to have a computation time which scales linearly  $O(n)$  with the scale of the model  $n$ , as compared to quadratic scaling  $O(n^2)$  in the original unsplit model. The split-model computation times were found to improve even further if the stiff solver was supplied with an analytical means of computing system Jacobians, while a two-state global step-size control algorithm also provided some speed-up.

A parallel model solution algorithm has been combined with numerical techniques which exploit knowledge of the model behaviour, and this will enable significantly faster simulations of the full-scale heart cell model.

## REFERENCES

- Cannell, M.B., Kong, C.H.T., Imtiaz, M.S., and Laver, D.R. (2013a). Control of sarcoplasmic reticulum  $ca^{2+}$  release by stochastic ryr gating within a 3d model of the cardiac dyad and importance of induction decay for cicr termination. *Biophysical Journal*, 104, 2149–2159.
- Cannell, M.B., Kong, C.H.T., Imtiaz, M.S., and Laver, D.R. (2013b). Supporting material for: Control of sarcoplasmic reticulum  $ca^{2+}$  release by stochastic ryr gating within a 3d model of the cardiac dyad and importance of induction decay for cicr termination.
- Curtis, A.R., Powell, M.J.D., and Reid, J.K. (1974). On the estimation of sparse jacobian matrices. *J. Inst. Maths Applics*, 13, 117–119.
- Jackson, K.R. (1996). The numerical solution of large systems of stiff ivps for odes. *Applied Numerical Mathematics*, 20, 5–20.
- Kelley, C.T. (2003). *Solving Nonlinear Equations with Newton's Method*. SIAM, Philadelphia.
- Kong, C.H., Laver, D.R., and Cannell, M.B. (2013). Extraction of sub-microscopic  $ca$  fluxes from blurred and noisy fluorescent indicator images with a detailed model fitting approach. *PLoS Comput Biol*, 9, e1002931.
- Kulakowski, B., Gardner, J., and Shearer, J. (2007). Integration step size. In *Dynamic Modeling and Control of Engineering Systems*, 129. Cambridge University Press, Cambridge.
- Li, P., We, W., Cai, X., Soeller, C., Cannell, M.B., and Holden, A.V. (2010). Computational modelling of the initiation and development of spontaneous intracellular  $ca^{2+}$  waves in ventricular myocytes. *Phil. Trans. R. Soc. A*, 368, 3953–3965.
- Olver, P.J. (2014). Finite difference approximations. In *Introduction to Partial Differential Equations*, 182. Springer.
- Shaikh, M., Wall, D., and David, T. (2011). Macro-scale phenomena of arterial coupled cells: a massively parallel simulation. *J. R. Soc. Interface*, 9, 972–987.
- Stern, M., Maltseva, L., Juhaszova, M., Sollott, S., Lakatta, E., and Maltsev, V. (2014). Hierarchical clustering of ryanodine receptors enables emergence of a calcium clock in sinoatrial node cells. *Journal of General Physiology*, 143, 577–604.