Towards Detailed Tissue-Scale 3D Simulations of Electrical Activity and Calcium Handling in the Human Cardiac Ventricle

Qiang Lan[1,2,3], Namit Gaur[2], Johannes Langguth[2(✉)], and Xing Cai[2,3]

[1] National University of Defense Technology, Changsha 410073, China
[2] Simula Research Laboratory, 1324 Fornebu, Norway
{lanqiang,ngaur,langguth,xingca}@simula.no
[3] Department of Informatics, University of Oslo, 0316 Oslo, Norway

Abstract. We adopt a detailed human cardiac cell model, which has 10000 calcium release units, in connection with simulating the electrical activity and calcium handling at the tissue scale. This is a computationally intensive problem requiring a combination of efficient numerical algorithms and parallel programming. To this end, we use a method that is based on binomial distributions to collectively study the stochastic state transitions of the 100 ryanodine receptors inside every calcium release unit, instead of individually following each ryanodine receptor. Moreover, the implementation of the parallel simulator has incorporated optimizations in form of code vectorization and removing redundant calculations. Numerical experiments show very good parallel performance of the 3D simulator and demonstrate that various physiological behaviors are correctly reproduced. This work thus paves way for high-fidelity 3D simulations of human ventricular tissues, with the ultimate goal of understanding the mechanisms of arrhythmia.

Keywords: Calcium handling · Multiscale cardiac tissue simulation · Supercomputing

1 Introduction

Calcium handling dysfunction is considered the likely cause of several cardiac pathological conditions such as heart failure [6,10], cardiac hypertrophy [2], cardiomyopathies [16], and inherited disorders of calcium release processes in the sarcoplasmic reticulum (SR) [7,17]. Many of these pathological conditions originate from dysfunctions of subcellular calcium release processes that occur at the microscopic and nanoscopic levels, ranging from dyadic disorganization by t-tubule malformation in heart failure [8,9,15] to single ryanodine receptor (RyR) dysfunction occurring in inherited cardiopathologies [5,7].

Over the last several years, advances in numerical methods and computing techniques have enabled the development of cardiac cell models of electrophysiology and calcium handling that take into account the discrete nature of subcellular stochastic calcium release processes [4,11,19,22]. This new generation of

G. Wang et al. (Eds.): ICA3PP 2015, Part III, LNCS 9530, pp. 79–92, 2015.
DOI: 10.1007/978-3-319-27137-8_7

models of calcium handling and action potential has been immensely useful in the study of causative and preventive mechanisms of arrhythmogenesis, which originates from the local nanoscopic level of channel and dyadic dysfunction to the subcellular and cellular levels of membrane potential abnormalities in the form of delayed afterdepolarizations, early afterdepolarizations [20], and cardiac alternans [12,13,19]. Despite this advancement in our understanding of how cardiac arrhythmogenesis can progress at different scales, ranging from single channel to whole-cell action potentials, several challenges remain.

For one, cardiac arrhythmias occur at the tissue and organ scale. Extrapolating the insights obtained from cell-level studies to understand how arrhythmias occur at the tissue and/or organ scale is often unclear and at times counter-intuitive. Secondly, tissue-level studies have proven to be particularly difficult due to their huge computational demands. A typical human heart has around 2×10^9 cells [1], each of which has about 10^6 RyRs and roughly 10^5 L-type channels operating stochastically in response to membrane potentials and local calcium concentrations [3]. Realistic simulations at this level of detail require immense computational power of massively parallel computers, together with sophisticated algorithms, so that these cardiac excitation processes can be studied in reasonable time. Thus to date, to the best of our knowledge, arrhythmia mechanisms at the tissue level have not been investigated computationally with detailed cell models of calcium handling.

This paper reports our newly developed parallel 3D simulator of electrical activity and calcium handling in the cardiac ventricular tissue. We will start with describing the novel numerical and computational approaches that have enabled such simulations with a detailed model of calcium handling. Then we will examine the parallel performance of the 3D simulator, followed by presenting multiscale simulation results of cardiac activity, including single channel RyR stochastic behavior, local dyadic calcium dynamics, whole-cell action potential and calcium dynamics, as well as cardiac excitation activity in the 3D tissue.

2 Mathematical Models and Numerical Methods

2.1 At the Tissue Level

The cardiac tissue is modeled by the following partial differential equation, which is commonly called the monodomain model:

$$\frac{\partial V_m}{\partial t} = \frac{-I_{\text{ion}}}{C_m} + D_x \frac{\partial^2 V_m}{\partial x^2} + D_y \frac{\partial^2 V_m}{\partial y^2} + D_z \frac{\partial^2 V_m}{\partial z^2}, \tag{1}$$

where V_m is the membrane potential, I_{ion} the ionic current provided by the underlying multiscale cell model of calcium handling (see Sect. 2.2), $C_m = 1\mu\text{F cm}^{-2}$ the membrane capacitance of the cell, $D_x = D_y = D_z = 0.2$ mm^2/ms the voltage diffusion coefficients in three spatial directions. In this paper, we consider the 3D solution domain of (1) as a slab of tissue made up of cardiac cells. The finite difference method, combined with an operator-splitting approach [18], is used to discretize (1). This means that the diffusion terms are treated separately from the I_{ion} term, where the latter requires solving the following detailed cell model per cell.

2.2 At the Cell Level

Electrophysiology Component. The multiscale model of stochastic calcium handling in a ventricular myocyte [4] forms the basis of the cell model used in this work. The electrophysiological currents used in [4] were those of a guinea-pig. Therefore, to perform simulations of the human cardiac ventricular tissue, we replace the electrophysiological current formulation with the O'Hara-Rudy (ORd) model [14] of a healthy human cardiac ventricular action potential. The total ionic current I_{ion} needed in (1) is computed by summing up the total ionic currents obtained from solving the ordinary differential equations in the ORd model.

Calcium Handling Component. The cell model consists of 10000 calcium release units (CaRUs) or dyads arranged as an internal $100 \times 10 \times 10$ grid in each cell. Each CaRU consists of five calcium compartments: (1) myoplasm, (2) submembrane space, (3) network sarcoplasmic reticulum, (4) junctional sarcoplasmic reticulum, and (5) dyadic space. The reader is referred to [4] for the detailed equations and parameters. Of particular importance, however, is the dyadic space that contains 15 L-type calcium channels and 100 RyRs that operate stochastically. More specifically, each RyR can be in one of four states, denoted as C1, C2, C3, and O1, at any given time. Figure 1 shows the possible transitions between them, which occur stochastically with probabilities that are related to the local calcium concentrations. The number of open RyRs, i.e. those having state O1, is of principal interest due to the effect of calcium influx on a cell's interior voltage.

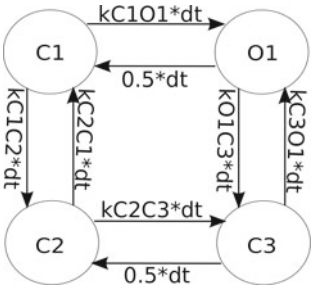

Fig. 1. The eight possible transitions between four states of a RyR, where the labels on the arrows indicate the probabilities of the transitions.

3 Implementation

3.1 Multi-level Parallelization

We adopt a multi-level parallelization strategy to facilitate our large scale tissue simulation. First, we divide the 3D tissue grid into a number of subgrids, each

being handled by one MPI process that occupies a compute node of a cluster. Then, OpenMP threads are used within each MPI process to parallelize all the computations at the cell level. These together provide the value of I_{ion} per cell. MPI communication is only needed at the tissue level, due to calculating the diffusion terms in (1). The majority of the computing time is spent on the cell-level computations.

3.2 Cell-Level Computations

Among all the cell-level computations, the most time-consuming part concerns calcium handling. We therefore direct our attention to the following function, named `computeCalciumInDyad`, which implements the calcium handling component (see Sect. 2.2):

```
void computeCalciumInDyad()
{
    generateRandData();
    for(int i=0;i<Ndyads;i++) {
        computateLocalLtypeCurrent();
        computeLocalSRCaRelease();
        computeLocalCaConcentration();
    }
    computeCaConcentrationDiffusion();
}
```

Fig. 2. The function that implements calcium handling per cell.

It can be seen that function `computeCalciumInDyad` invokes five functions, where function `generateRandData` generates all the needed random numbers per cell per time step, and function `computeCaConcentrationDiffusion` computes the diffusion of intracellular calcium concentrations between the dyads. The other three functions are invoked in every iteration of the `for` loop. In particular, the main part of function `computeLocalSRCaRelease` addresses the random state transitions for the 100 RyRs per dyad. In the following, we will present three programming and numerical techniques that have a positive impact on the performance of a baseline implementation of the parallel simulator. Note that the computational cost of the entire `computeCalciumInDyad` function is directly proportional to the number of dyads Fig. 6.

Avoiding Redundant Calculations. s Many variables are involved in the computation of each dyad. Some of these variables vary from dyad to dyad, whereas others remain constant. Considering that the number of dyads in our detailed cell model is 10000, pre-calculating the constant variables outside the `for`-loop can yield substantial performance gains. The effort required is to identify these loop-constant variables among the complicated equations involved. Figure 5 clearly shows that the effect of this optimization is considerable.

Vectorizing Intracellular Diffusion Calculations. Intracellular diffusion in the 10000 dyads, which form an internal $100 \times 10 \times 10$ grid, is needed for three calcium concentrations. The code segment in Fig. 3 shows the actual implementation of one such 3D diffusion computation. In order to utilize the vectorization capability of modern CPUs, it is important to use a data structure where elements in x are continguous in memory since the x loop is the innermost of the nested triple `for`-loops. In order to inform that the compiler that it is safe to vectorize this loop, we add the `ivdep`. Note that this pragma only asserts the data independency of the grid cells in the diffusion step. The actual vectorization is left to the compiler. However, even though the 256 bit vector length of our target machine suggests the possibility of a fourfold speeup, the fact that such stencil computations are generally memory bound implies that the actual performance gain will be far lower. Furthermore, the other parts of the computation are not vectorized manually and cannot be vectorized automatically by the compiler. The techniques required for efficient vectorization of such complex code are beyond the scope of this work.

Using Binomial Distribution. Recall from Sect. 2.2 that the dyadic space contains 100 RyRs, each of which can be in one of four states at any given time. Transitions between the states occur stochastically. To count the number of RyRs that have the open (O1) state, which determines the calcium influx on a cell's interior voltage, a straightforward approach is to individually simulate the random transition per RyR. In total 100 random numbers (lying uniformly between 0 and 1) are needed, one per RyR. Depending on the current state and on the corresponding transition probabilities, a RyR may change into another state. Such an approach is very computationally heavy, due to the need of generating many random numbers and the use of many `if`-tests in the implementation.

Therefore, it is computationally beneficial to collectively compute the number of RyRs in the four states, instead of individually tracing the transition of each RyR. We thus replace the 100 individual random experiments with eight random samples from a binomial distribution, one for each possible state transition, as shown in Fig. 1. The cumulative probability of having up to k successes in n trials with individual probabilities of success p is given as follows:

$$F(k, n, p) = Pr(X \leq k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}, \tag{2}$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

is the binomial coefficient.

As the number of RyRs in each state ranges from 0 to 100, we can precompute all the required binomial coefficients. This has a significant performance benefit. Using a random number r that is drawn from a uniform distribution in $[0, 1]$, we sample from the binomial distribution by finding the smallest k for

```
for(z=1;z<Nz_diff-1;z++)
    for(y=1;y<Ny_diff-1;y++) {
       int x,c,n,s,b,t;
       x=0;
       c=x+y*Nx_diff+z*Nx_diff*Ny_diff;
       n=c-Nx_diff;             s=c+Nx_diff;
       b=c-Nx_diff*Ny_diff;     t=c+Nx_diff*Ny_diff;

       U[c]=u[c]+fx*(2*u[c+1]-2*u[c])+fy*(u[n]+u[s]-2*u[c])
                +fz*(u[b]+u[t]-2*u[c]);
       #pragma ivdep
       for(x=1;x<Nx_diff-1;x++) {
          ++c; ++n; ++s; ++b; ++t;
          U[c]=u[c]+fx*(u[c-1]+u[c+1]-2*u[c])+fy*(u[n]+u[s]-2*u[c])
                   +fz*(u[b]+u[t]-2*u[c]);
       }
       U[c]=u[c]+fx*(2*u[c-1]-2*u[c])+fy*(u[n]+u[s]-2*u[c])
                                     +fz*(u[b]+u[t]-2*u[c]);
    }
```

Fig. 3. Pragma guided vectorization (in the x direciton) of one of three diffusion computations between the dyads in function `computeCaConcentrationDiffusion`.

which $r \leq F(k, n, p)$. While standard implementations of the binomial distribution incur a high computational cost, we use an efficient custom implementation which is shown in Fig. 4. Since the binomial coefficients are precomputed, we only need to multiply the base probability which starts at $(1-p)^n$ by $p/(1-p)$ in every round, store the result and multiply it by the binomial coefficient.

Let us denote the number of RyRs in the four states as x_1, x_2, x_3, x_4, and the number of RyRs which transition from state i to state j as x_{ij}. The values x_{ij} are obtained by sampling the binomial distribution as described above. Now, the number of RyRs in each state in the next time step is:

$$x_i = x_i - \sum_j x_{ij} + \sum_j x_{ji}. \tag{3}$$

Based on this, we add two optimizations according to the characteristics of the cell model. In our model, the transition probabilities from O1 to C1 and from C3 to C2 are constant at $p_c = 0.5 * dt$. Thus, we can pre-compute the entire cumulative probability function $F(k, n, p_c)$ in the same way that we pre-compute the binomial coefficient. In both cases, we need to store $101*100/2 = 5050$ values, which is no significant cost. In fact, storing a square table of 101^2 entries is more efficient and convenient.

In addition, we exploit a second property of the cell model. Most of time the RyRs are in state C2 and all the transition probabilities except the two constant ones mentioned above are close to 0. This means that it is quite likely that the result obtained by sampling from the binomial distribution is also 0. We make use of this by setting a small probability p_t as a threshold, and precompute

$F(0, n, p_t)$ for all $0 \leq n \leq 100$. Now, if $p \leq p_t$, for a given random number r we can simply check whether $r \leq F(0, n, p_t)$, and if that is the case, we obtain $k = 0$, i.e. no state transition happens, without computing the binomial distribution. Of course, if the number of RyRs in one state is 0, then the number of RyRs transitioning from that state is also 0. Figure 4 below shows our implementation of the optimized sampling from the binomial distribution.

Binomial distributions for modeling RyR transitions have been used in e.g. [19]. However, there the authors do not compute the actual binomial distribution. Instead, they approximate it using the normal and Poisson distributions, at the expense of accuracy.

4 Performance Results and Analysis

4.1 Hardware and Numerical Setup

Our test system is Abel [21], a supercomputer operated by the University of Oslo. The compute nodes on Abel are equipped with dual Intel Xeon E5-2670 (Sandy Bridge) processors. Each node has 16 physical compute cores running at 2.6 GHz. The interconnect is FDR (56 Gbps) Infiniband. We use Intel's *icc* compiler 15.1.0 for compilation and the Intel MPI 5.0.2 library for communication. Up to 128 compute nodes, i.e. 2048 CPU cores, have been used for the following numerical experiments.

For all the experiments, a fixed time step size of 0.05 ms is used at both the tissue level and the cell level. When doing tissue-scale simulations, we have chosen a fixed spatial mesh resolution of 0.5 mm to discretize the diffusion terms in (1).

4.2 Performance Optimization Experiment

The goal of our first numerical experiment is to test the improvement in performance obtained by our optimizations to the `compute_cell` function described in Sect. 3.2. To do so, we run 10000 time steps for a single cell with 10000 dyads. This is equivalent to simulating one cardiac beat of 500 ms. The cell is stimulated at $t = 50$ ms. Figure 5 shows the improvement in performance due to different optimizations. Removing redundant calculations yields a substantial improvement in three functions: `computateLocalLtypeCurrent` (by 37.5 %), `computeLocalSRCaRelease` (by 24.4 %) and `computeCaConcentrationDiffusion` (by 12.4 %). The vectorization accelerates the diffusion further by 25 %. Finally, using binomial distributions has the highest impact, accelerating SRCaRelease by another 70 % and RandData by 79.9 %. The latter improvement is solely due to the reduced number of random values required by the binomial method. Overall, the combined effect of all three optimizations reduces the computation time by 50.7 %.

4.3 Scaling Experiment

We perform both weak scaling and strong scaling tests for a simulation of 1000 ms, and for each test we run two types of simulation, one uses 100 dyads per

```
int constant_p_binomial(n,randValue) {
    k=0;
    while(randValue>Table[n,k])
        k++;
    return k;
}

int binomial(n,p,randValue) {
        if n = 0
        return 0;
    if p < Threshold AND randValue < Precomp[n]
        return 0;
    k = 0;
    p_current =pow((1-p),n);
    p_step = p/(1-p);
    while(randValue > 0) {
        k++;
        randValue -= Binom[n,k]*p_current;
        p_current *= p_step;
    }
    return k;
}
```

Fig. 4. Implementation of Binomial Distribution Method. Function `constant_p_binomial` simply finds k by using the precomputed lookup table. In function `binomial`, we first test if the computation can be skipped, using the threshold p_t and the precomputed value $F(0, n, p_t)$. If this is not the case, the distribution function $F(k, n, p)$ is computed iteratively by subtracting from `randValue` using the precomputed binomial coefficients.

cell and the other uses 10000 dyads per cell. For weak scaling tests, the number of compute nodes we use ranges from 1 to 128. When using 100 dyads, the tissue size is $64 \times 64 \times 64$ per node for a total of $512 \times 256 \times 256$ cells at 128 nodes. For 10000 dyads, the tissue size is $16 \times 16 \times 16$ per node, which amounts to $128 \times 64 \times 64$ cells when using 128 compute nodes. We measure scalability via the number of cell computations performed for each wall-clock second of simulation time used. Here, a cell computation is defined as computing one cell for a single time step. We then plot this metric against the number of nodes used. Figure 6 shows that we obtain very good weak scaling.

For the strong scaling test, the tissue size is fixed at $256 \times 256 \times 256$ cells for the 100 dyad case and at $32 \times 32 \times 32$ cells when using 10000 dyads per cell. Due to memory requirements, at least 8 compute nodes are needed. The same cell computation metric as in the weak scaling case is used. Figure 6 shows the results of our experiments. Compared with the result of weak scaling, we achieve almost the same performance. The difference can be explained by the communication overhead, which is not hidden by computation in our current implementation.

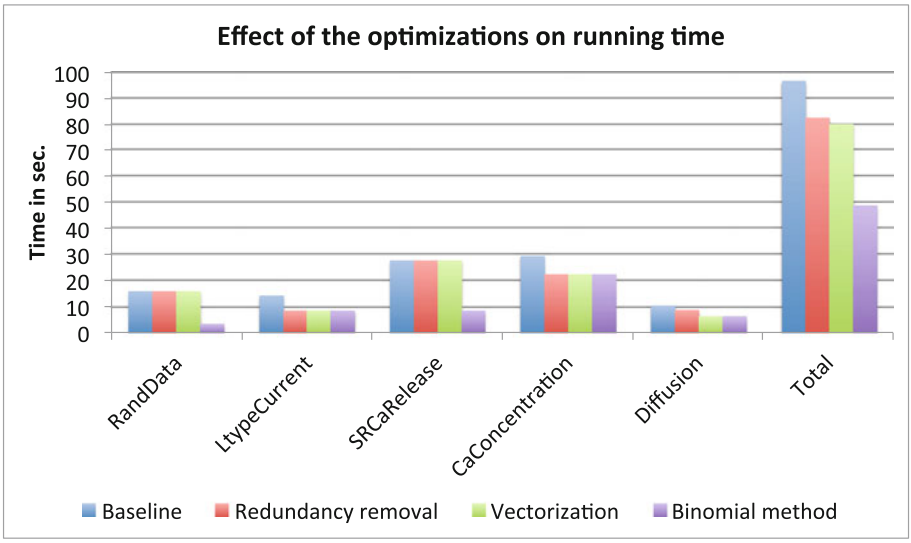

Fig. 5. Performance improvement of the individual functions in `computeCalciumInDyad` due to the different optimization techniques. The three optimizations are applied cumulatively. Thus, the values for the Binominal method reflect the sum of all improvements.

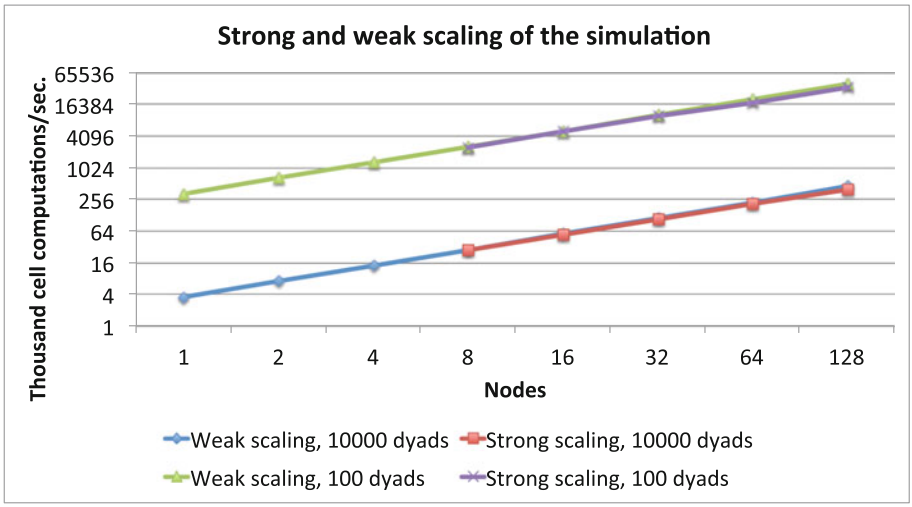

Fig. 6. Performance of weak and strong scaling tests of tissue simulations. The Y axis shows performance measured via the number of cell computations (i.e. time steps for a single cell) performed for each wall-clock second of simulation time used.

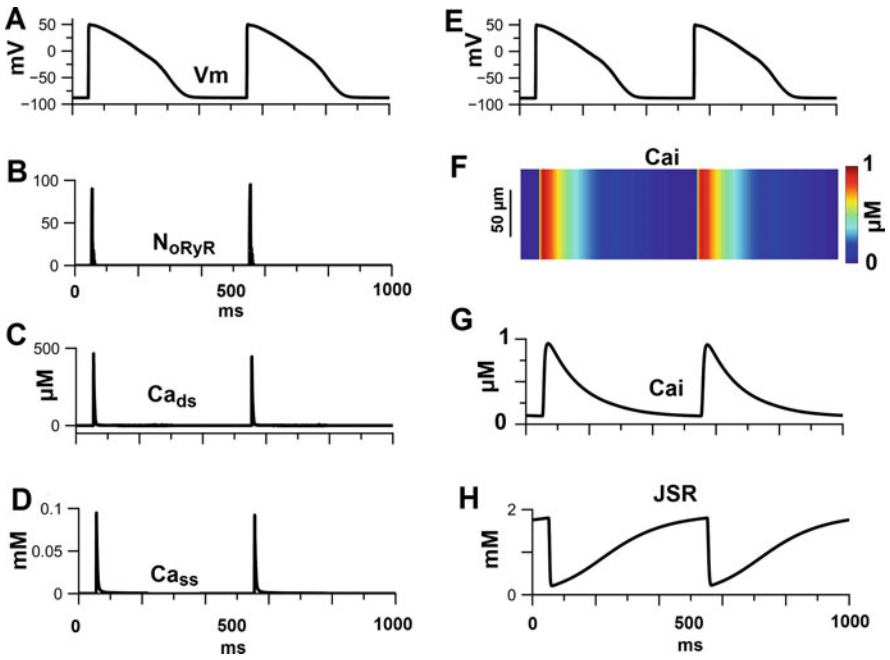

Fig. 7. Calcium handling in a cell in a human tissue during normal excitation. (A) Action Potential. (B) Number of open RyRs (N_oRyR) in the center dyad of the cell. (C) Dyadic space calcium (Ca_{ds}) (D) Submembrane space calcium (Ca_{ss}) (F) Simulated linescan image of intracellular Ca (Ca_i (G) Whole-cell Ca_i and (H) Whole-cell junctional sarcoplasmic reticulum (JSR). The 3D tissue was plane-stimulated at an edge at a cycle length (CL)=500 ms. Last two steady-state beats are shown.

In conclusion, due to the heavy cell computation, the simulation scales very well. Thus, using the metric of cell computations per second, one can easily predict the runtime of simulations using an arbitrary grid size, number of time steps, and compute nodes.

4.4 Calcium Handling in a Cell

In Fig. 7 we show calcium handling in a human cell during normal cardiac excitation. The calcium handling results are shown at different scales for the cell at the center of the tissue. Panel A shows action potential (AP) in two consecutive beats. Panel B indicates the number of open RyRs (N_{oRyR}) in the dyad at the geometric center of the cell. These numbers were computed using our binomial distribution method as described in Sect. 3.2. Simulations predict that early during the AP most of the RyRs are open. In response to these channel openings, Ca_{ds} (Panel C) in the dyad rises to a values as high as 500 μM and Ca_{ss} (panel D) to 0.1 mM. Within a dyad, the temporal profile of both Ca_{ds} and Ca_{ss} follow closely that of N_{oRyR}. At the subcellular scale, the simulated linescan image of

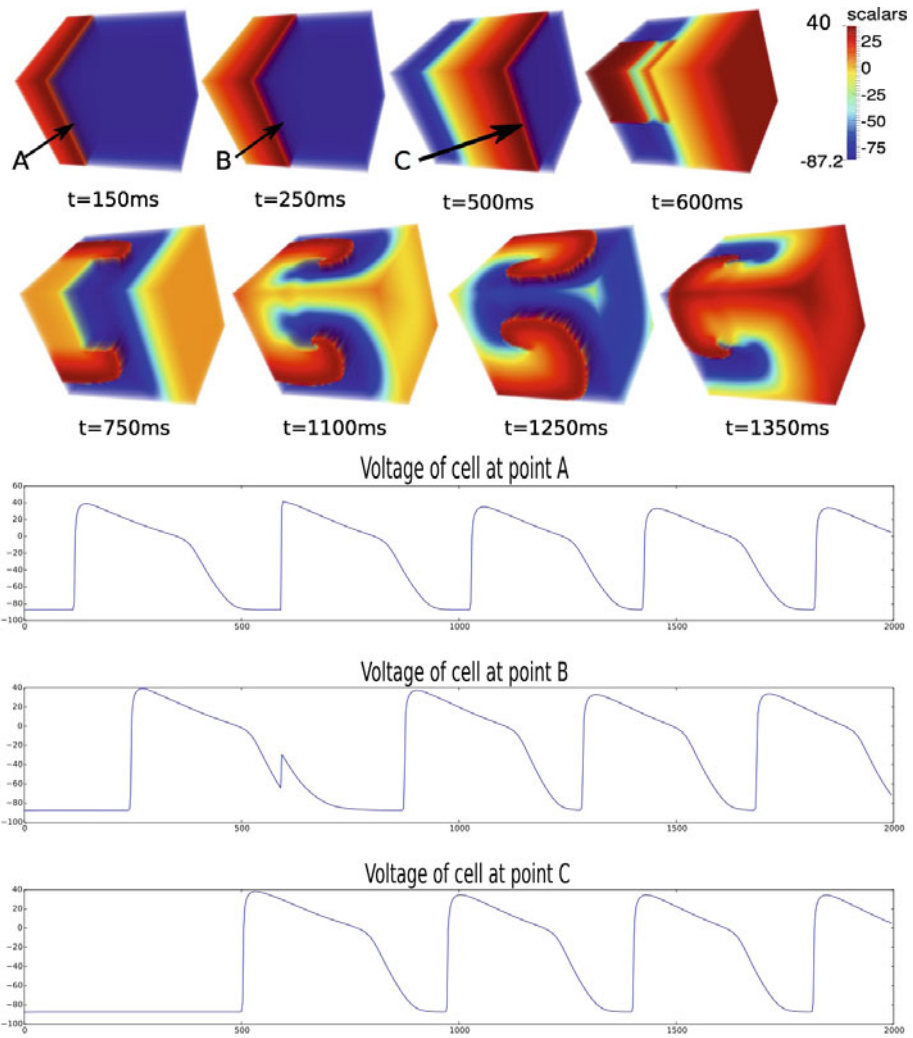

Fig. 8. Simulation of scroll waves in a 3D tissue. The top part shows voltage snapshots at different time points. The bottom section shows membrane potentials at three different locations in the tissue.

intracellular Ca (Ca_i) indicates that calcium release occurs synchronously across all the dyads (Panel F). The corresponding whole-cell Ca_i is indicated in Panel G. Panel H shows the whole-cell JSR. In summary, these results show calcium handling at different scales: dyad, subcell, and cell during tissue-simulations in a normal cardiac excitation process. The whole-cell values are consistent with [14].

4.5 Simulated Arrhythmia in Tissue

Figure 8 shows time snapshots of membrane potential in the 3D tissue. The tissue was stimulated at one edge at t = 0 ms. At t = 600 ms a portion of the tissue was cross field stimulated during the vulnerable window. This protocol initiates a scroll wave. The membrane potentials in the cell at three different tissue locations indicated by A, B, and C are shown below the snapshots. During the duration of this simulation the scroll waves were stable and did not degenerate into wave breaks. The simulations shows the ability of the tissue simulator to generate scroll waves which are commonly used to computationally study mechanisms of arrhythmia in cardiac tissue.

5 Summary and Conclusions

In summary, in this paper we report detailed 3D tissue simulations of electrical activity and calcium handling in a human cardiac ventricle using a multiscale model of calcium handling in a cell. Many previous 3D tissue simulation studies have employed whole-cell calcium handling models. To date multiscale calcium handling cell models have not been used for tissue simulations. This is both due to the immense computational power required to handle such simulations and limitations of current numerical and algorithmic approaches for these computations. Here, we have used optimizations such as avoiding redundant computations, vectorization and binomial distribution to reduce the computation time of 3D tissue simulations. Overall, the computation time was reduced by half. The most significant performance gain was due to using binomial distribution and the subsequent reduction in the random number generations required for the stochastic simulations that compute the SR Ca release flux and the L-type Ca flux. The weak scaling test indicates a near-constant compute time under varying tissue size. The strong scaling test indicates a near-linear decrease in compute time as a function of the number of compute nodes used. These results indicate a promising possibility of porting the tissue simulation to a massive compute cluster. Finally, we show the fidelity of our tissue simulations by demonstrating that calcium handling in a cell is consistent with the published human cell model [14]. We also show the ability to simulate reentrant arrhythmias in the 3D tissue.

To conclude, we have developed a detailed 3D tissue simulator of electrical activity and calcium handling in a human cardiac ventricle using algorithmic optimizations. The promising scalability suggests that whole-heart simulations are potentially within reach of the largest supercomputers available today. In future work, we intend to further improve the performance of our simulator and port it to hardware accelerator architectures. This will eventually open up the possibility of understanding the multiscale mechanisms of reentrant cardiac arrhythmias originating from calcium handling dysfunction.

Acknowledgements. The first author is supported by a mobility grant within UTFORSK project No. 2013-10091. The third and fourth authors are supported by FRINATEK project No. 214113. We gratefully acknowledge the computing time provided by NOTUR.

References

1. Adler, C., Costabel, U.: Cell number in human heart in atrophy, hypertrophy, and under the influence of cytostatics. Recent Adv. Stud. Card. Struc. Metab. **6**, 343–355 (1974)
2. Berridge, M.: Remodelling Ca2+ signalling systems and cardiac hypertrophy. Biochem. Soc. Trans. **34**(2), 228–231 (2006)
3. Cheng, H., Lederer, W., Cannell, M.B.: Calcium sparks: elementary events underlying excitation-contraction coupling in heart muscle. Science **262**(5134), 740–744 (1993)
4. Gaur, N., Rudy, Y.: Multiscale modeling of calcium cycling in cardiac ventricular myocyte: macroscopic consequences of microscopic dyadic function. Biophys. J. **100**(12), 2904–2912 (2011)
5. Jiang, D., Wang, R., Xiao, B., Kong, H., Hunt, D.J., Choi, P., Zhang, L., Chen, S.W.: Enhanced store overload-induced Ca2+ release and channel sensitivity to luminal Ca2+ activation are common defects of RyR2 mutations linked to ventricular tachycardia and sudden death. Circ. Res. **97**(11), 1173–1181 (2005)
6. Kubalova, Z., Terentyev, D., Viatchenko-Karpinski, S., Nishijima, Y., Györke, I., Terentyeva, R., da Cuñha, D.N., Sridhar, A., Feldman, D.S., Hamlin, R.L., et al.: Abnormal intrastore calcium signaling in chronic heart failure. Proc. Nat. Acad. Sci. USA **102**(39), 14104–14109 (2005)
7. Liu, N., Colombi, B., Memmi, M., Zissimopoulos, S., Rizzi, N., Negri, S., Imbriani, M., Napolitano, C., Lai, F.A., Priori, S.G.: Arrhythmogenesis in catecholaminergic polymorphic ventricular tachycardia insights from a RyR2 R4496C knock-in mouse model. Circ. Res. **99**(3), 292–298 (2006)
8. Louch, W.E., Bito, V., Heinzel, F.R., Macianskiene, R., Vanhaecke, J., Flameng, W., Mubagwa, K., Sipido, K.R.: Reduced synchrony of Ca2+ release with loss of T-tubulesa comparison to Ca2+ release in human failing cardiomyocytes. Cardiovasc. Res. **62**(1), 63–73 (2004)
9. Louch, W.E., Mørk, H.K., Sexton, J., Strømme, T.A., Laake, P., Sjaastad, I., Sejersted, O.M.: T-tubule disorganization and reduced synchrony of Ca2+ release in murine cardiomyocytes following myocardial infarction. J. Physiol. **574**(2), 519–533 (2006)
10. Marks, A.R., et al.: Calcium cycling proteins and heart failure: mechanisms and therapeutics. J. Clin. Investig. **123**(1), 46–52 (2013)
11. Nivala, M., de Lange, E., Rovetti, R., Qu, Z.: Computational modeling and numerical methods for spatiotemporal calcium cycling in ventricular myocytes. Front. Physiol. **3**, 114 (2012)
12. Nivala, M., Qu, Z.: Calcium alternans in a couplon network model of ventricular myocytes: role of sarcoplasmic reticulum load. Am. J. Physiol.-Heart Circulatory Physiol. **303**(3), H341–H352 (2012)
13. Nivala, M., Song, Z., Weiss, J.N., Qu, Z.: T-tubule disruption promotes calcium alternans in failing ventricular myocytes: mechanistic insights from computational modeling. J. Mol. Cell. Cardiol. **79**, 32–41 (2015)
14. O'Hara, T., Virág, L., Varró, A., Rudy, Y.: Simulation of the undiseased human cardiac ventricular action potential: model formulation and experimental validation. PLoS Comput. Biol. **7**(5), e1002061 (2011)
15. van Oort, R.J., Garbino, A., Wang, W., Dixit, S.S., Landstrom, A.P., Gaur, N., De Almeida, A.C., Skapura, D.G., Rudy, Y., Burns, A.R., et al.: Disrupted junctional membrane complexes and hyperactive ryanodine receptors after acute junctophilin knockdown in mice. Circulation **123**(9), 979–988 (2011)

16. Pieske, B., Kretschmann, B., Meyer, M., Holubarsch, C., Weirich, J., Posival, H., Minami, K., Just, H., Hasenfuss, G.: Alterations in intracellular calcium handling associated with the inverse force-frequency relation in human dilated cardiomyopathy. Circulation **92**(5), 1169–1178 (1995)
17. Priori, S.G., Chen, S.W.: Inherited dysfunction of sarcoplasmic reticulum Ca2+ handling and arrhythmogenesis. Circ. Res. **108**(7), 871–883 (2011)
18. Qu, Z., Garfinkel, A.: An advanced algorithm for solving partial differential equation in cardiac conduction. IEEE Trans. Biomed. Eng. **46**(9), 1166–1168 (1999)
19. Restrepo, J.G., Weiss, J.N., Karma, A.: Calsequestrin-mediated mechanism for cellular calcium transient alternans. Biophys. J. **95**(8), 3767–3789 (2008)
20. Song, Z., Ko, C.Y., Nivala, M., Weiss, J.N., Qu, Z.: Calcium-voltage coupling in the genesis of early and delayed afterdepolarizations in cardiac myocytes. Biophys. J. **108**(8), 1908–1921 (2015)
21. University of Oslo: Abel. http://www.uio.no/english/services/it/research/hpc/abel/
22. Williams, G.S., Chikando, A.C., Tuan, H.T.M., Sobie, E.A., Lederer, W., Jafri, M.S.: Dynamics of calcium sparks and calcium leak in the heart. Biophys. J. **101**(6), 1287–1296 (2011)