BMC Bioinformatics

BioMed Central

Methodology article

Open Access

Asynchronous adaptive time step in quantitative cellular automata modeling

Hao Zhu[1], Peter YH Pang[2], Yan Sun[1] and Pawan Dhar*[1]

Address: [1]Bioinformatics Institute, National University of Singapore, 30 Biopolis Street, Singapore 138671 and [2]Department of Mathematics, National University of Singapore, Lower Kent Ridge Road, Singapore 119260

Email: Hao Zhu - zhuhao@bii.a-star.edu.sg; Peter YH Pang - matpyh@nus.edu.sg; Yan Sun - sunyan@bii.a-star.edu.sg; Pawan Dhar* - pk@bii.a-star.edu.sg

* Corresponding author

Published: 29 June 2004

BMC Bioinformatics 2004, **5**:85 doi:10.1186/1471-2105-5-85

Received: 15 March 2004
Accepted: 29 June 2004

This article is available from: http://www.biomedcentral.com/1471-2105/5/85

Abstract

Background: The behaviors of cells in metazoans are context dependent, thus large-scale multi-cellular modeling is often necessary, for which cellular automata are natural candidates. Two related issues are involved in cellular automata based multi-cellular modeling: how to introduce differential equation based quantitative computing to precisely describe cellular activity, and upon it, how to solve the heavy time consumption issue in simulation.

Results: Based on a modified, language based cellular automata system we extended that allows ordinary differential equations in models, we introduce a method implementing asynchronous adaptive time step in simulation that can considerably improve efficiency yet without a significant sacrifice of accuracy. An average speedup rate of 4–5 is achieved in the given example.

Conclusions: Strategies for reducing time consumption in simulation are indispensable for large-scale, quantitative multi-cellular models, because even a small 100 × 100 × 100 tissue slab contains one million cells. Distributed and adaptive time step is a practical solution in cellular automata environment.

Background

Quantitative multi-cellular modeling, used, for example, in the electrophysiological modeling of the heart, in neural networks, and in the metabolic modeling of cells in tissues/organs [1,2], is an important area of computational biology [3]. Computational efficiency is an important issue for such *in silico* research, because, first, many ordinary differential equations (ODE) employed to describe intracellular activity are nonlinear; and second, such models often consist of a large number of homo- or heterogeneous cells (Fig 1). Electrophysiological modeling of the heart, in which the description of electrical activity in each of a large number of heterogeneous cells is based on Hodgkin-Huxley (HH) type membrane equations, is a typical case. Effective reduction of time required is essential to make simulation affordable on available computer resources.

Several methods, including the very popular adaptive time step method, have been proposed to enhance the computational efficiency of large-scale systems. Obviously, the effectiveness of a method is chiefly dependent on the details of implementation as well as the scale and nature of models. For example, the classic Rush & Larson method for runtime adjustment of time step, though very straightforward for single-cell modeling, faces such challenges as parallelism, asynchronism and heterogeneity when used in multi-cellular biological models [4]. Very

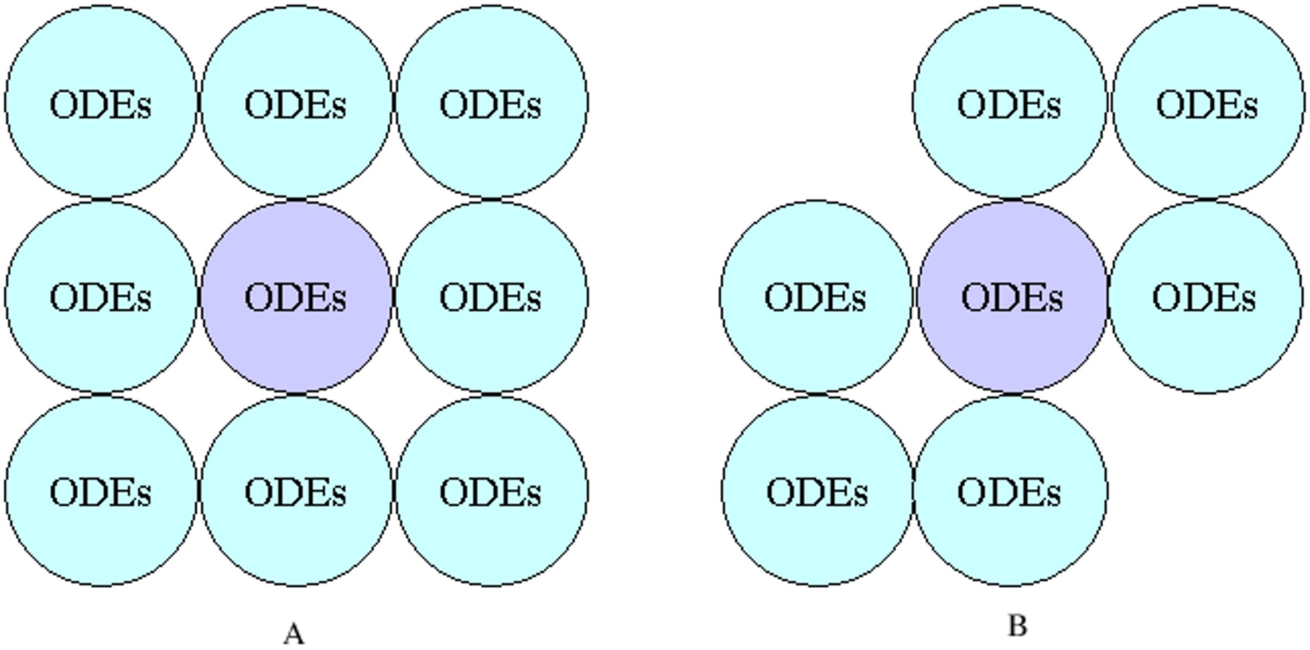


Figure 1
Quantitative cellular automata with different neighborhoods (radius = 1). (A) Moor neighborhood. (B) A user-defined neighborhood.

often, a method coded and workable in one programming environment cannot be directly adopted in another. Thus, new methods, accompanied by new modeling strategies and tools, are reported from time to time.

While partial differential equations (PDE) are often used to model propagation phenomena in multi-cellular situations [5], there are cases where they are not applicable, and propagation must be simulated at a cell-to-cell level. This occurs, for example, when a system is heterogeneous consisting of different kinds of components. When a system evolves dynamically and shows emergent behaviors, centralized, PDE based descriptions also become unfitting. Modeling arrhythmias with a heart model is such a case in which the electrical properties of assorted cardiac cells and gap junctions may change substantially in simulation. For such systems, decentralized modeling is needed, which calls for relevant optimal numerical methods to conquer the time consumption problem.

Modeling with cellular automata has gained much prevalence recently [6-9], especially for tissue/organ modeling, not only for its ability to simulate discrete intercellular communication, but also for its implicit large-scale parallelism. In such modeling, a natural mapping between each biological cell and each automaton cell is often assumed. Although most reported applications are qualitative in nature [9-12], cellular automata are not necessarily restricted to discrete modeling. Various extensions can be made to existing systems to facilitate various types of cellular automata computation [13]. Effective methods to improve the efficiency of quantitative cellular automata computation are therefore of much interest. The purpose of this short paper is to introduce a novel and widely applicable method to implement distributed and asynchronous adaptive time step in quantitative cellular automata modeling. A heterogeneous electrophysiological model of the heart built with five groups of membrane equations of cardiac cells is described as the test model, and performance evaluation based on it is made. As in Rush & Larson method, cyclic activity of cells, either synchronous or asynchronous, is the basis of the distributed asynchronous adaptive time step method.

Results

The test model is a two dimensional one consisting of six kinds, 4300 cardiac cells whose activity is described by relevant HH type membrane equations (see Fig 2). Electrical activity is cyclically created in the sinoatrial node cells by the membrane equations in these cells. Then, the potential difference between repolarized cells and resting cells triggers the membrane equations in resting cells, driving the propagation of cardioelectrical activity in the heart. The long repolarization and resting stages in each cell in

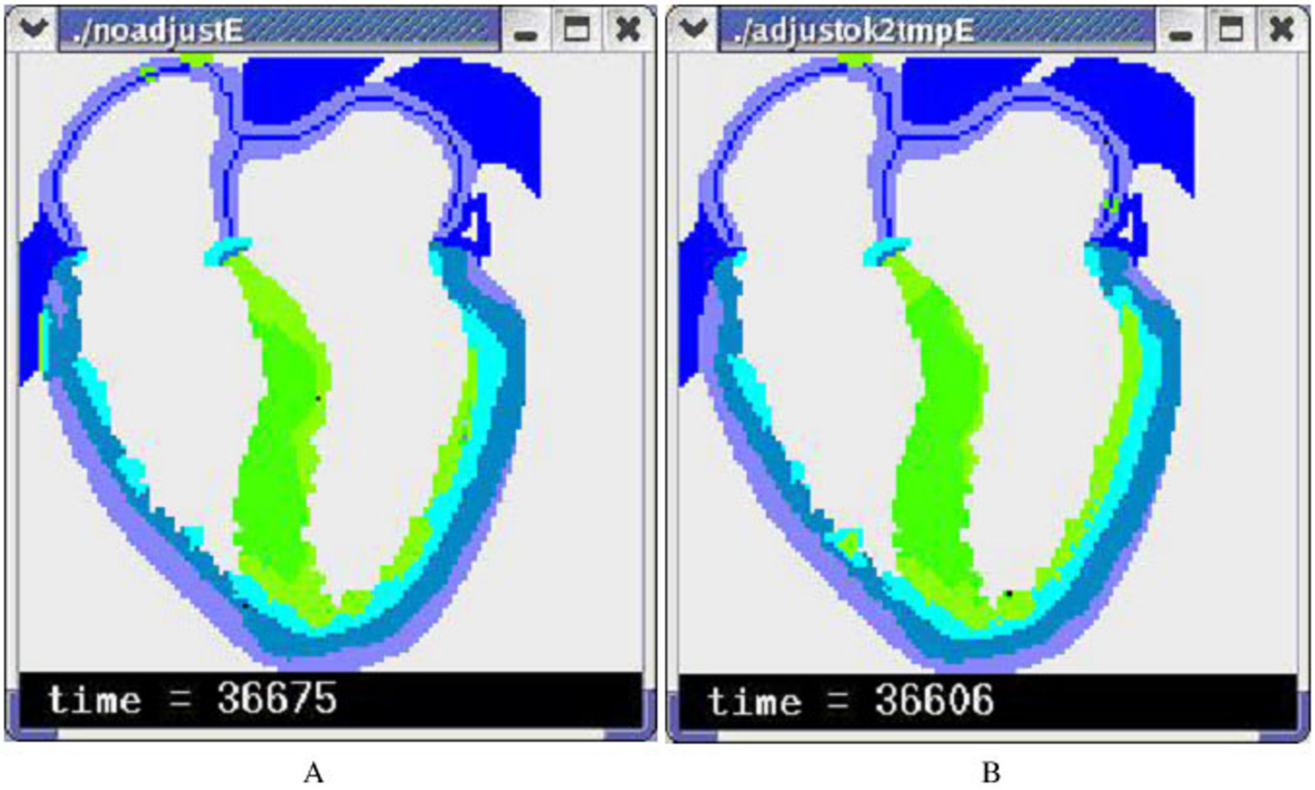
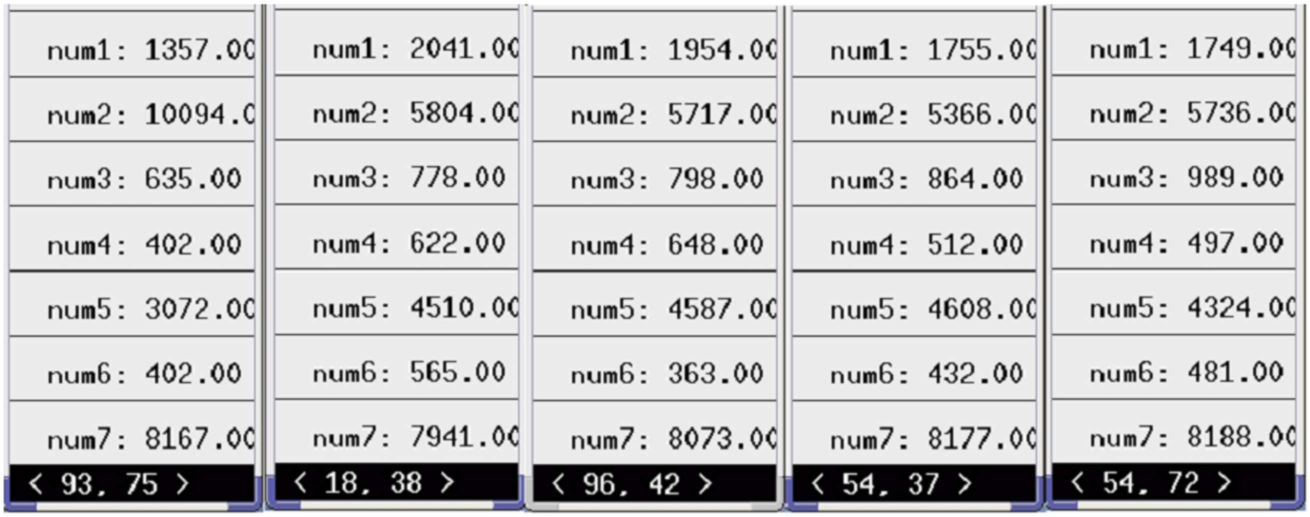


Figure 2
Simulation results of the heterogeneous 2D model locating in a 256 × 256 cell space. Different colors stand for different values of the dynamically computed membrane potential of cells. 36675 and 36606 are running steps; each step corresponds to 0.01 ms (mini second). The color strips between the inner surface and the outer surface show the normal endocardium-to-epicardium repolarization. At 366 ms most cells have undergone depolarization and are at the end of repolarization, the high accordance between picture A and picture B therefore indicates the validity and safety of the adaptive time step method. A. The distribution of membrane potential of cells in epicardium-to-endocardium repolarization without the adaptive time step. B. The distribution of membrane potential of cells with the adaptive time step. Pictures are captured with GIMP, a screen capture tool under Linux.

each cycle makes adaptive time step method useful. Simulations were carried out on a PC with a 1.7 GHz Pentium4 CPU. First, we used MatLab to perform simulation with two ventricular cells connected by a gap junction (the first cell is stimulated by an external current; the second one is triggered by the transjunctional current) to find out appropriate thresholds for *Ina* and *factor*. A workable (but possibly not optimal) set of *Ina* value for ventricular cells is 0.01, 0.001, 0.002 (mA), and the corresponding values of *factor* are 1, 2, 5, and 7. The second cell executes 10071 integrations in a 600 ms simulation. A speedup rate of 5.96 (= 60000/10071) is acquired.

To confirm the validity and stability of this method in a realistic heterogeneous model, we carried out simulation with the sparse 2D test model. It shows that the method worked reliably and produced an action potential distribution at repolarization stage that agrees quite well with that produced by simulation without adaptive time step (Fig 2). At the time of 100,000 steps (1000 ms), we checked five ventricular cells to see how many skips occurred in each. A group of extra fields, *num1* to *num7*, were set in the *Cellang* program to record the times of integration under different skips (see Fig 3). For these five cells, an average speedup rate of 4.456 was reached. As in the two-cell simulation, small fluctuations of transmembrane current and gating variable *m* occur in the phase 4 of action potential caused by larger time steps, but they did not affect action potential and transjunctional propagation.

Since the heterogeneous model does not occupy the whole cell space, the non-cardiac automaton cells should also spend a certain amount of time on housekeeping operations. To further check the efficiency of the method, we made 400 ms simulations with a 128 × 128 homoge-



num1: 1357.00	num1: 2041.00	num1: 1954.00	num1: 1755.00	num1: 1749.00
num2: 10094.0	num2: 5804.00	num2: 5717.00	num2: 5366.00	num2: 5736.00
num3: 635.00	num3: 778.00	num3: 798.00	num3: 864.00	num3: 989.00
num4: 402.00	num4: 622.00	num4: 648.00	num4: 512.00	num4: 497.00
num5: 3072.00	num5: 4510.00	num5: 4587.00	num5: 4608.00	num5: 4324.00
num6: 402.00	num6: 565.00	num6: 363.00	num6: 432.00	num6: 481.00
num7: 8167.00	num7: 7941.00	num7: 8073.00	num7: 8177.00	num7: 8188.00
< 93, 75 >	< 18, 38 >	< 96, 42 >	< 54, 37 >	< 54, 72 >

Figure 3

The number of integrations in a 1000 ms simulation. Five cells locating at <93,75>, <18,38>, <96,42>, <54,37>, <54,72> are chosen to display the number of integration happened in each cell in a simulation with the adaptive time step. numX gives the number of integrations; the real time step is Δ*t**X.

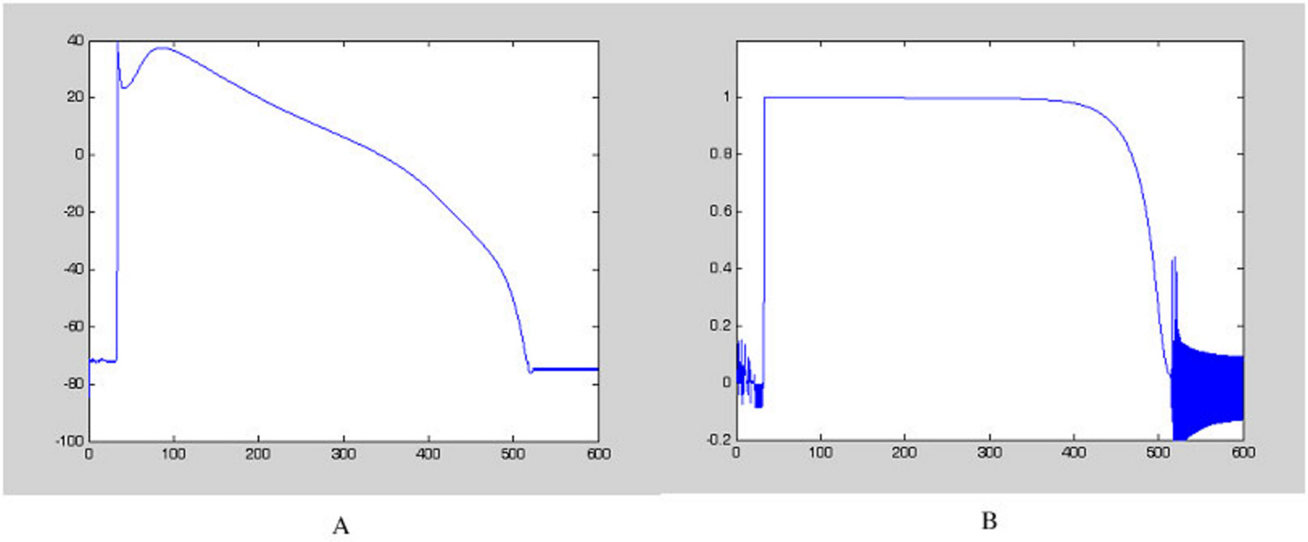


Figure 4

The two-cell simulation results (the second cell) using MatLab. A. The action potential. B. The state of gating variable *m* that controls the sodium channel. In phase 4, *m* fluctuates within a range, but the action potential keeps correct.

neous 2D model fully occupied by ventricular cells except 1 or 4 sinoatrial node cells locating at 1 or 4 corners to initiate excitation. With the adaptive time step method, the one-sinoatrial-node-cell version consumed 56 m 20 s to run 400 ms, the four-sinoatrial-node-cell version spent 56 m 58 s. However, without the adaptive time step method, the one-sinoatrial-node-cell version needs 142 m 56 s to make a 400 ms simulation. We remark that the fact that the agreement of the speedup rate for both propagation stimulated by one corner and propagation stimulated by four corners is significant for the simulation of abnormal (arrhythmic) propagation, as it indicates that highly asynchronous cellular activities, such as ectopic beats and reentrant movement, do not degrade the performance as they do in some methods [20].

Discussion

We described a novel method to implement distributed and asynchronous adaptive time step in a quantitative cellular automaton environment and showed that this method is numerically efficient. The variation of cellular activity that admits different scale of time step of integration is the precondition to employ the method. Simulation results acquired from two electrophysiological models prove the effectiveness of this method, in which a speedup rate of about 5 can be reached. Since each cell determines its time step independently based on the value of *Ina*, the dynamic adjustment of time step of all cells runs in a highly asynchronous way. Simulation results of the heterogeneous and sparse 2D model show that the method can work effectively in realistic models. Also, as we have indicated above, the method is effective in the presence of highly asynchronous cellular activities. We believe this is because the method itself works in a fully asynchronous way and a fine-grained and implicit domain decomposition exists naturally (each cell is a subdomain). When the global Δt is small enough and the threshold values of *factor* and *Ina*, the indicator of cellular activity, are set within suitable bounds determined by single-cell simulations, the method is safe and stable. Simulation shows that the computational gain always surpasses the overhead of using two extra fields to check the value of *Ina* and to keep tract of *skip*.

In comparison with Quan *et al*'s method [20], the advantage of this method is that there is neither an explicit domain decomposition nor a priority queue. Actually, a fine-grained domain decomposition is implicit in cellular automata computation as each automaton cell is both a computing unit and a subdomain. Thus, there is no extra cost of domain decomposition. Besides, in a complex and highly heterogeneous model, an algorithm for explicit domain decomposition may be very complicated and dependent on many details of the model such as dimension and geometrical properties. In contrast, the model based on quantitative cellular automata described here is completely general.

We remark that Quan *et al* obtained a reduction of computation as high as 17 times (from 3 to 17), with the highest speedup rate attained in the resting phase (phase 4) of the action potential [20]. This is, however, not supported by our results. Both the two-cell straightforward simulation and the sparse heterogeneous simulation show that the dynamics of the gating variable *m* is unstable in phase 4, especially when time step is large (Fig 4). By contrast, we obtained the highest stable speedup rate in the plateau phase. We found the overall 5-times speedup rate we obtained is the same as that reported by Cherry *et al* [21], who used an explicit mesh refinement method. How large a speedup rate can be obtained and when the largest speedup rate is attained depend on the dynamic properties of the ODEs involved. Our data show that the activity of ventricular cells described with the Luo-Rudy phase I model is more stable in phase 2 than in phase 4, in which a 7-times speedup rate cannot be sustained.

In a specific model built with quantitative cellular automata, how much time can be saved depends on several factors. The first is the structure of a model. The sparser a model is, like the heart with four chambers and an irregular surface, the less we benefit from the method, because of the housekeeping operations spent on non-biological cells. The second factor is the time spent on cell communication when *istim* is also used to jointly control the skip. In a 3D model with the Moore neighborhood, each cell has 26 neighboring cells. Thus, the time spent on computing *istim* will be tripled compared to a 2D model in which a cell has only 8 neighbors. In electrophysiological modeling, when a complex algorithm is used to compute transjunctional current [22], the time spent on cell communication can be considerable. Thirdly, the complexity of differential equations has a significant impact on the performance. If newly published action potential models, which use many more ODEs to describe more channels and the concentration fluctuation of ions, are used, the suspension of integration can significantly improve the performance of the simulation. If, on the other hand, the action potential model is very simple, for example, of the Fitzhugh-Nagumo type, the extra cost arising from the use of two additional fields might neutralize the benefit of adaptive time step. Finally, the proper definition of thresholds of *Ina* and *factor* is crucial.

Conclusion

Modeling with cellular automaton has become very popular recently, especially in the modeling of tissues/organs in computational biology. Our work demonstrates that quantitative modeling, which requires an extension of classical cellular automata, is feasible. Furthermore, the simple yet effective method described in this paper shows that distributed and asynchronous adaptive time step can be readily implemented and the performance of simulation can be significantly improved without a significant loss of accuracy of simulation. Besides electrophysiological modeling, this method can be applied to other biological modeling. For example, in cell cycle control and signal transduction models of tissue growth and morphogenesis [23,24], or in multi-cellular model of host-pathogen interaction [12], even a small-scale 100 × 100 × 100 3D model contains 1 million cells, whose complex, local activities described with ODEs will lead to much more complex, global behavior through cell to cell communication. The adaptive time step method in quantitative cellular automata environment described here may render the simulation practicable.

Methods

Quantitative Cellular Automata

Celluar is a cellular automata system with the programming language *Cellang* [14]. A *Cellang* program is shared by all automaton cells and describes the computation within and between cells. A data file, which can be edited manually or created by a program, specifies the location of cells in an n-dimensional cell array and the initial value of cell field(s) in each cell. The data file functions as input to the *Cellang* program. A predefined variable *time*, which increases by 1 after each step, provides synchronization for all cells. Though common arithmetical and logical operators are provided, *Cellang* lacks the necessary facilities for numerical computation.

Discrete in value should not be compulsory for modeling with cellular automata in many cases. Exploiting the fact that a *Cellang* program uses C files as intermediate codes, we built numerical computation facilities into it by adding floating-point data type, function calls (including the mathematical functions in the C library), and other quantitative facilities. With these extensions, numerical solutions of ODE can be coded and large-scale parallel solution of ODEs (over a large number of cells) can be realized in a simple and straightforward way. We have built several models with the extended system, including an electrophysiological model of the whole heart for which the proposed adaptive time step method is very important.

Models of the Heart

To validate the efficiency of the asynchronous adaptive time step method, two prototypical models were designed and tested. The first one is a 128 × 128 2D homogeneous ventricular sheet, with cells of sinoatrial node located at one or four corners to initiate excitation. The second one, also with a 128 × 128 resolution, is a 2D heterogeneous cardiac sheet consisting of ventricular and atrial cells, cells of atrioventricular and sinoatrial nodes, and those of conduction fibers. The electrical activities in these cells are described by the corresponding HH type action potential models [15-19]. We note that a 128 × 128 2D model involves 16384 action potential models, i.e., 16384 systems of ODEs. Numerical solution of ODEs is carried out using the explicit Euler method. In both models, a standard Moore neighborhood is adopted, i.e., each cell has 8 neighbors. A simple and static gap junction model, which uses the potential difference between two cells and the resistance of the gap junction to determine the transjunctional current, is adopted. The resistance of gap junctions does not change with membrane potentials.

Among the various quantitative cellular automata models in which intracellular activity is described by ODE, the heterogeneous electrophysiological model may be the best one to illustrate the parallel solution of nonlinear (HH type) equations and the implementation of asynchronous adaptive time steps. In this model, the activity of different kinds of cardiac cells, which we assume to be in one-to-one correspondence with automaton cells, is rendered as follows: A field *type* is defined in the *Cellang* program, with initial value set in the data file. According to the different value of *type*, the *Cellang* program is divided into several sections with the *if-then* statement. Different cells execute codes in different sections. A set of cell fields are declared to store the current value of the membrane potential, transjunctional currents and gating variables. Windows are created to display a selected field, such as the transmembrane potential, of all cells (Fig 2), as well as various cellular and channel electrical activities of any selected cells (Fig 3).

Asynchronous Adaptive Time Step

Although there are different methods to implement adaptive time step in biological models built with ODEs, they actually share the same biological basis, i.e., the fluctuation of cell activity such as cell cycle activity and electrical activity in which the change rate of key variables is different in different times. What we did is to implement the method in a quantitative and heterogeneous cellular automata environment. To use the quantitative cellular automata to simulate a system with ODEs in a large number of cells, besides the predefined *time* that is used to iterate integration in cells, a user-defined floating type time step Δt is defined, which is automatically shared by all cells, for the numerical solution of ODEs. The value of Δt depends on the nature of ODEs. In our two illustrative models it is selected to be 0.01 ms. Δt is changeable in runtime. However, to describe asynchronous cellular activities caused and connected by normal and abnormal cell communication (such as excitation propagation in this case), it is not feasible to dynamically change Δt. For a fixed Δt, to simulate 1000 ms of electrical activity, every cell in the model should run 100,000 times integration. Yet, there is another way to make usage of the cyclic property of cellular activity to reduce computation in simulation. In the case of the electrophysiological model described here, the integration of membrane equation can be skipped when the cellular activity is not very active.

As in other methods [20], for the modeling of ventricular cells we can use *Ina*, the ionic current of sodium, to reflect the activity of each ventricular cell. We use an extra cell field to store *Ina*. At any step of the computation, if a ventricular cell's *Ina* value is smaller than a threshold, the integration of all membrane equations at this step can be skipped. Based on results acquired from single-cell simulation, we can specify a set of *Ina* thresholds to determine how many steps of integration can be safely skipped in different situations. In our examples, four *Ina* thresholds are set to determine whether integration of the HH equations should be executed immediately, or skipped 1, 4, or 6 times respectively (specified by a temporary variable *factor* whose value is 1, 2, 5, 7). At the same time, we also record how many steps have been skipped since the last integration with a cell field *skip*. Thus, the real time step used in each integration for each cell is $\Delta t*skip$.

We note, however, that a static skip mechanism (a specific value of *Ina* determines a specific value of *factor*) cannot accurately deal with emergent depolarization that can be triggered by either normal or abnormal transjunctional currents. The latter situation frequently occurs in various arrhythmias. The *Ina* determined, unmodifiable *factor* makes the final $\Delta t*skip$ often too large, leading to errors or even an overflow. Hence, we make the *skip* dynamically adjustable as follows:

1. In each step, initially set the default value of *factor* to 1, meaning to do integration immediately;

2. Read the field *Ina*, according to its value and the preset thresholds of *Ina*, determine the value of *factor*;

3. If *factor* <= *skip*, do integration and reset *skip* to 1;

4. Otherwise, skip integration and increase the skip number by *skip* = *skip* + 1.

In this way, for every cell, the real time step used in each integration is dynamically adjusted at runtime according to the strength of its activity.

For cells where *Ina* is not the major inward depolarizing current, *dvdt*, the change rate of membrane potential, can be the indicator of cellular activity. Although the algorithm can be put before computing the transjunctional currents to skip even transjunctional currents computation, a safer choice is to put it after the computation and use *istim*, the sum of transjunctional currents that functions as the stimulating current to a cell, to jointly control the suspension of integration.

Authors' contributions

HZ invented the method and drafted the paper. PP provided critical comments on mathematical issues and polished the paper. YS was involved in implementing the method. PD was participated in and in charge of the project. All authors prepared, read and approved the final manuscript.

Acknowledgements

This work was supported by A-STAR, the Agency of Science, Technology And Research, of Singapore.

References

1. Thakor NV Jr, Ferrero JM, Saiz J, Gramatikov BI, Ferrero JM Sr: **Electrophysiologic models of heart cells and cell networks.** *IEEE Eng Med Biol Mag* 1998, **17(5):**73-83.
2. Segre D, Vitkup D, Church GM: **Analysis of optimality in natural and perturbed metabolic networks.** *Proc Natl Acad Sci* 2002, **99:**15112-15117.
3. Noble D: **The rise of computational biology.** *Nature Rev Mol Cell Biol* 2002, **3:**460-461.
4. Rush S, Larson H: **A practical algorithm for solving dynamic membrane equations.** *IEEE Trans Biomed Eng* 1978, **25:**389-392.
5. Qu Z, Garfinkel A: **An advanced algorithm for solving partial differential equation in cardiac conduction.** *IEEE Trans Biomed Eng* 1999, **46:**1166-1168.
6. Wolfram S: **Cellular automata as models of complexity.** *Nature* 1984, **311:**419-424.
7. Gutowitz H: **Cellular automata: theory and experiment.** *Physica D* 1990, **45:**1-3.
8. Wolfram S: *A New Kind of Science* Wolfram Media, Inc; 2002.
9. Ermentrout GB, Edelstein-Keshet L: **Cellular automata approach to biological modeling.** *J Theor Biol* 1993, **160:**97-133.
10. Siregar P, Sinteff JP, Julen N, Le Beux P: **An interactive 3D anisotropic cellular automata model of the heart.** *Comput Biomed Res* 1998, **31:**323-347.
11. Patel AA, Gawlinski ET, Lemieux KL, Gatenby RA: **A cellular automata model of early tumor growth and invasion: the effects of native tissue vascularity and increased anaerobic tumor metabolism.** *J Theor Biol* 2001, **213:**315-331.
12. Benyoussef A, El HafidAllah N, El Kenz A, Ez-Zahraouy H, Loulidi M: **Dynamics of HIV infection on 2D cellular automata.** *Physical A* 2003, **322:**506-520.
13. Griffeath D, Moore C: *New Constructions in Cellular Automata* Oxford University Press; 2003.
14. Eckart JD: **A cellular automata simulation system: version 2.0.** *SIGPLAN Notices* 1992, **27(8):**99-106.
15. Yanagihara K, Noma A, Irisawa H: **Reconstruction of sino-atrial node pacemaker potential based on the voltage clamp experiments.** *Japan J Physiol* 1980, **30:**841-857.
16. Liu Y, Zeng W, Delmar M, Jalife J: **Ionic mechanisms of electronic inhibition and concealed conduction in rabbit atrioventricular nodal myocytes.** *Circulation* 1993, **88:**1634-1646.
17. Nygren A, Fiset C, Firek L, Clark JW, Lindblad DS, Clark RB, Giles WR: **Mathematical model of an adult human atrial Cell: the role of K+ currents in repolarization.** *Circ Res* 1998, **82:**63-81.
18. Luo CH, Rudy Y: **A model of the ventricular cardiac action potential, depolarization, repolarization, and their interaction.** *Circ Res* 1991, **68:**1501-1526.
19. McAllister RE, Noble D, Tsien RW: **Reconstruction of the electrical activity of cardiac purkinje fibres.** *J Physiol* 1975, **251:**1-59.
20. Quan W, Evans SJ, Hastings HM: **Efficient integration of a realistic two-dimensional cardiac tissue model by domain decomposition.** *IEEE Trans Biomed Eng* 1998, **45:**372-384.
21. Cherry EM, Greenside HS, Henriquez CS: **A space-time adaptive method for simulating complex cardiac dynamics.** *Phys Rev Lett* 2000, **84:**1343-1346.
22. Vogel R, Weigart R: **Mathematical model of vertebrate gap junctions derived from electrical measurements on homotypic and heterotypic channels.** *J Physiol* 1998, **510:**177-189.
23. Chen KC, Csikasz-Nagy A, Gyorffy B, Val J, Novak B, Tyson JJ: **Kinetic analysis of a molecular model of the budding yeast cell cycle.** *Mol Biol Cell* 2000, **11:**369-391.
24. Schoeberl B, Eichler-Jonsson C, Gilles ED, Muller G: **Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors.** *Nat Biotechnol* 2002, **20:**370-375.