

High-throughput cardiac science on the Grid

David Abramson, Miguel O. Bernabeu, Blair Bethwaite, Kevin Burrage, Alberto Corrias, Colin Enticott, Slavisa Garic, David Gavaghan, Tom Peachey, J. Pitt-Francis, E. Pueyo, Blanca Rodriguez, Anna Sher and Jefferson Tan

Phil. Trans. R. Soc. A 2010 **368**, 3907-3923
doi: [10.1098/rsta.2010.0170](https://doi.org/10.1098/rsta.2010.0170)

References

This article cites 19 articles, 3 of which can be accessed free

<http://rsta.royalsocietypublishing.org/content/368/1925/3907.full.html#ref-list-1>

Rapid response

Respond to this article

<http://rsta.royalsocietypublishing.org/letters/submit/roypta;368/1925/3907>

Subject collections

Articles on similar topics can be found in the following collections

[software](#) (21 articles)

[e-science](#) (16 articles)

[computational biology](#) (140 articles)

Email alerting service

Receive free email alerts when new articles cite this article - sign up in the box at the top right-hand corner of the article or click [here](#)

To subscribe to *Phil. Trans. R. Soc. A* go to:
<http://rsta.royalsocietypublishing.org/subscriptions>

High-throughput cardiac science on the Grid

BY DAVID ABRAMSON^{1,*}, MIGUEL O. BERNABEU², BLAIR BETHWAITE¹,
KEVIN BURRAGE², ALBERTO CORRIAS², COLIN ENTICOTT¹, SLAVISA GARIC¹,
DAVID GAVAGHAN², TOM PEACHEY¹, J. PITT-FRANCIS², E. PUEYO²,
BLANCA RODRIGUEZ², ANNA SHER³ AND JEFFERSON TAN¹

¹*Faculty of Information Technology, Monash University, Caulfield East,
Melbourne, Victoria 3145, Australia*

²*Oxford University Computing Laboratory, Wolfson Building, Parks Road,
Oxford OX1 3QD, UK*

³*Department of Physiology, Anatomy and Genetics, University of Oxford,
Le Gros Clark Building, South Parks Road, Oxford OX1 3QX, UK*

Cardiac electrophysiology is a mature discipline, with the first model of a cardiac cell action potential having been developed in 1962. Current models range from single ion channels, through very complex models of individual cardiac cells, to geometrically and anatomically detailed models of the electrical activity in whole ventricles. A critical issue for model developers is how to choose parameters that allow the model to faithfully reproduce observed physiological effects without over-fitting. In this paper, we discuss the use of a parametric modelling toolkit, called NIMROD, that makes it possible both to explore model behaviour as parameters are changed and also to tune parameters by optimizing model output. Importantly, NIMROD leverages computers on the Grid, accelerating experiments by using available high-performance platforms. We illustrate the use of NIMROD with two case studies, one at the cardiac tissue level and one at the cellular level.

Keywords: cardiac science; Grid computing; e-Science; bio-engineering

1. Introduction

Cardiac electrophysiology is a mature discipline, with the first model of a cardiac cell action potential (AP) having been developed in 1962 (Noble 1962). Current models of cardiac electrophysiology span the range from models of single ion channels (e.g. those of Capener *et al.* 2002), through very complex models of individual cardiac cells (see Rudy & Silva 2006 for a review), to geometrically and anatomically detailed models of the electrical activity in whole ventricles (see review by Kerckhoffs *et al.* 2006).

Software that simulates cardiac electrical activity varies in scale and complexity. Initially, codes were developed as stand-alone applications, coded in traditional scientific languages such as Fortran and C. More recently, languages such as Matlab have been used extensively because of the higher level and

*Author for correspondence (david.abramson@infotech.monash.edu.au).

One contribution of 16 to a Theme Issue ‘e-Science: past, present and future I’.

more mathematically oriented expressive power. Moreover, the Matlab runtime contains a number of powerful ‘tool boxes’ that accelerate coding by providing mature and efficient solvers for, for example, differential equations. While earlier research has made significant progress with these stand-alone codes, recent activity has focused on building specific software packages for modelling the heart. These packages include CARP (<http://carp.meduni-graz.at/>), CMISS (<http://www.cmiss.org/>), CONTINUITY (<http://www.continuity.ucsd.edu/>) and Chaste (<http://web.comlab.ox.ac.uk/chaste/>), among others.

As with many other simulation endeavours, there is no single model that can reproduce the behaviour of a complete organ such as a heart. The enormous differences in scale of the physical processes (10^9 spatially and 10^{15} temporally) mean that individual models can only represent a subset of the processes, and higher-level simulations need to incorporate parametrizations of lower-level processes. When done this way, it is common to develop a high-resolution model of some low-level process and use this to calibrate a coarser-grain model with parameters controlling key processes. This multi-scale simulation can then be used to investigate particular research questions related to behaviour of the organ under specific conditions—for example, to understand the effect of an infarct on cardiac propagation or external defibrillation on the heart. A critical issue then is how to choose parameters that allow the model to faithfully reproduce observed physiological effects without over-fitting.

Over the past 15 years, we have developed a family of software tools, called NIMROD, that perform parameter exploration on computationally expensive applications (Abramson *et al.* 1995, 2000*b*). NIMROD allows a user to describe an ‘experiment’ in which a model is run repeatedly across different parameter combinations. Because we expect models to be time consuming, they run on Grid-enabled resources, exploiting internal parallelism in the application (if it is available), and also external parallelism of many independent simulations. This means that we can achieve very high throughput given sufficient Grid resources, allowing us to explore many different parameter combinations.

In this paper, we discuss the varying approaches to modelling cardiac electrophysiological function, from the ionic to complete organ level. We then discuss the NIMROD family of tools in more detail, and show how these can be used in combination with cardiac modelling software to solve problems in cardiac simulation. We illustrate, using two different case studies, that when NIMROD is combined with existing modelling packages, it is possible to solve complex parameter fitting and inverse problems in cardiac science.

2. Cardiac electrophysiological modelling

This section briefly introduces cardiac modelling with special stress on computer simulation and the available software packages.

(a) *From ion channels to electrocardiogram*

Computational cardiac electrophysiology has rapidly evolved over the past 50 years, and it is now possible to simulate the electrophysiological activity of the heart from ion channels to the electrocardiogram (ECG) using anatomically based models.

(i) Single-cell cardiac modelling

Following the seminal work of Hodgkin and Huxley, Denis Noble published his first model of cardiac AP in 1960 (Noble 1960). Since then, a large number of mathematical models of the cellular AP have been developed, not only for ventricular myocytes but also for a range of cell types (e.g. Purkinje, atrial, sino-atrial) and species. These mathematical abstractions model the evolution of different cellular ionic concentrations and gating parameters by means of ordinary differential equations (ODEs). Their complexity varies, but the most refined models include over 50 ODEs. The CellML project (<http://www.cellml.org>) has carried out an important task of standardization in the field. Their XML-based (eXtensible Markup Language) description language is widely accepted by the community. A comprehensive collection of published models can be found on their website.

A cardiac AP is the variation in transmembrane potential that happens in a myocyte throughout the cardiac cycle. The semipermeable nature of the cell membrane and the ions flowing through it in an ordered manner give rise to that variation. This is the main force driving the mechanical activity of the heart, since the myocyte contraction is driven by the electrical activation. Common myocytes require an external electrical stimulus to trigger an AP. Under normal physiological conditions, this external stimulus is an AP potential happening in a neighbouring cell. Cells in pacemaker regions like the sino-atrial node, the atrio-ventricular node and the Purkinje fibres present self-stimulation capabilities.

(ii) Cardiac tissue modelling

As mentioned before, common myocytes require an external electrical stimulus to trigger an AP. Therefore, if we assume that some parts of the tissue receive an initial stimulus by a self-stimulated cell, cardiac myocytes propagate the AP locally in a reaction–diffusion manner. Activation happens across the tissue in the form of smooth wavefronts.

One of the most widely accepted mathematical abstractions of this reaction–diffusion process is the bidomain equations (Keener & Sneyd 2001). They consist of a coupled system of equations describing the evolution of the intracellular and extracellular potential fields, ϕ_i and ϕ_e , through the cardiac tissue. More precisely, two partial differential equations are coupled at each point in space, with a system of ODEs modelling the ionic current flowing across the membrane from one space into the other (as described in §2*a*(i)).

The transmembrane potential $V = \phi_i - \phi_e$ is often a magnitude of interest as well. The simulation of electrical activity at the whole-organ level is a computationally challenging problem and thus state-of-the-art numerical, computational and software engineering techniques need to be employed (Bernabeu *et al.* 2009*b*; Pitt-Francis *et al.* 2009).

*(b) Modelling technology**(i) Legacy languages*

Legacy languages such as Fortran77 or C have played a crucial role in the advance of numerical computing and science in general. On the one hand, it is frequently claimed that procedural languages such as Fortran or C are more

efficient than their object-oriented (OO) counterparts (e.g. C++, Java). It has been argued that procedural source code is closer to assembler code and therefore compilers can do a better job optimizing it. Although certainly true in some cases (note, for example, the performance penalties associated with the use of virtual methods in C++), there is no definitive answer to this question. Results are frequently dependent on the benchmark used and very much on the programmer skills (Cary *et al.* 1997).

On the other hand, OO languages are known to be well suited for modelling real-world systems (biological systems in our case): concepts such as class inheritance and polymorphism prove powerful when dealing with biological entities that share certain characteristics. Encapsulation and data abstraction help with code maintainability and documentation. The maturity and popularity of OO languages like C++ ensure availability of native libraries for the solution of common computational and numerical problems (e.g. message-passing parallel environments, numerical algorithms, testing suites, meshing).

(ii) *Fourth-generation modelling languages*

Fourth-generation modelling languages, such as Matlab, are often the language of choice when engineers and scientists, without any background in computer science, approach computer programming for the first time. Features like a user-friendly interface and syntax, an integrated debugging environment and not having to worry about compiling seem appealing to the neophyte. A large number of built-in modules and toolkits make developing models in many fields of science and engineering quick and simple. However, users often face performance issues as soon as they try to run their codes on inputs with size of scientific relevance. This is largely due to the interpreted nature of the language. Important efforts have been made in an attempt to address this issue (e.g. parallel Matlab, precompiled built-in code for critical functions). However, the Matlab code is still regarded as poor in terms of performance when compared against a C++ equivalent. Similar performance issues occur with other high-level interpreted languages and are not restricted to the Matlab product *per se*.

Nevertheless, fourth-generation modelling languages remain a powerful tool well suited for developing small-scale models or prototypes of larger ones. These prototypes are often developed in order to assess the viability of certain solutions.

(iii) *Domain-specific modelling frameworks*

Several packages exist for the simulation of electrical propagation in cardiac tissue, among them: CARP (<http://carp.meduni-graz.at/>; Plank *et al.* 2008), CMISS (<http://www.cmiss.org/>), CONTINUITY (<http://www.continuity.ucsd.edu/>) and MEMFEM. Many publications (Plank *et al.* 2008; Pitt-Francis *et al.* 2009) can be found containing results coming from these packages. However, each shares one or more of the following limitations:

- They have been developed with a single application in mind (i.e. study of cardiac arrhythmias) and therefore application to other fields of research is potentially restricted.

- No stress has been put on using state-of-the-art software engineering techniques during their development. Their reliability and correctness are therefore potentially compromised.
- They do not achieve maximum efficiency on high-performance computing platforms or do not have any parallel capabilities at all. This constrains the complexity of the studies and the areas of application.
- They are not freely available to the scientific community. This prevents external evaluation of their correctness from happening and reduces the chances of reproducing results.

In this context, a new software package called Chaste (<http://web.comlab.ox.ac.uk/chaste/>) has been developed by the members of the Computational Biology group at the University of Oxford. Chaste stands for ‘CANCER, HEART AND SOFT TISSUE ENVIRONMENT’; it tries to address these problems and therefore becomes the next generation of simulation software, not only in cardiac electrophysiology, but also in cardiac electromechanics, cancer and soft-tissue modelling. Chaste is currently able to simulate monodomain and bidomain electrical activity in any cardiac model provided by the user or automatically generated based on a geometric description. Several ionic models are available for the description of the membrane kinetics. Additionally, any model implemented in CellML can be easily imported into Chaste by means of PyCML (Cooper *et al.* 2006). Chaste is able to generate fibre orientation data for any cardiac model based on the mathematical formulation of Streeter (Bernabeu *et al.* 2008). Different stimulation and pacing protocols are available, as well as the possibility of specifying different tissue parameters such as electrical conductivities and numerical techniques and parameters. All the simulation parameters are supplied to the Chaste executable through its XML configuration file.

3. The NIMROD tool family

This section introduces the NIMROD tool family and explains how cardiac modelling software can take advantage of it in order to gain access to the Grid and perform automated tasks.

(a) *Grid computing*

The Grid provides a general platform for integrating computation, data and instruments (Foster & Kesselman 2003). It serves as the infrastructure for implementing novel applications, particularly in science and engineering. In particular, ‘computational’ grids have emerged as a viable platform for delivering on-demand access to a range of very high-performance machines. While it may not be possible to gain access to sufficient resources at any single site, computational grids can aggregate a number of otherwise separate resources into a single large super-computer. Such a virtual machine, or testbed, is an ideal base for simulating complex systems using computational models because the resources can be assembled at a period of peak demand and then released for use when not required. Such platforms have the potential to offer very cost-effective solutions, leveraging everything from spare cycles on high-end machines through to large pools of inexpensive desktops that are idle.

In spite of the enormous progress in building operation grids, and the significant effort in developing middleware, assembling such a testbed on demand is difficult. Most grids are built from different components, and this resource heterogeneity is a fact of life. Likewise, grids are built across multiple administrative and security domains, posing problems for aggregating them into a single virtual machine. Lack of a single owning organization also means that resource scheduling becomes complex—no single job scheduler can guarantee access to sufficient computational power, making it difficult to deliver the guaranteed levels of service. Importantly, grid application users do not want to know about the complexity of the underlying fabric, and wish to concentrate on their domain science.

Difficulty in using the Grid is not a hypothetical concern. Currently, very few scientists use the Grid routinely, and instead rely on local resources, which are under their control. This means that the scale and nature of the work is limited. Until we can make it easier to use, the Grid will never be adopted by more than the most hardy or desperate users.

Over the years, we have developed a strategy for delivering the high levels of performance, and have built software tools that make it easy for scientists to leverage the computational power of the Grid. Specifically, the NIMROD family of tools allows a non-expert to specify large computational experiments using legacy software, and execute these over a range of Grid resources. This is highly relevant for the work discussed in this paper because, as we have argued in §2*b*, complex computational models may be built using a variety of different programming techniques. Accordingly, no single programming environment will be used uniformly, and it is unlikely that all environments would ever be adapted to work on Grid resources.

NIMROD is not a single tool: it incorporates a component that distributes computations to the resources (NIMROD/G; Abramson *et al.* 1995, 2000*b*), a component that searches for ‘good’ solutions using nonlinear optimization algorithms (NIMROD/O; Abramson *et al.* 2000*a*, 2001), and a component that helps evaluate which parameter settings are important using the experimental design (NIMROD/E; Peachey *et al.* 2008). Most aspects of NIMROD have been written about extensively over the years, so we will only provide a cursory overview in §3*b*. More information, including download instructions, is available at <http://messagelab.monash.edu.au/nimrod>. Moreover, NIMROD has been widely applied to a variety of application domains, and some of these are listed at <http://messagelab.monash.edu.au/EScienceApplications>.

(b) The NIMROD tool family

Figure 1 shows the architecture of the NIMROD tool family and the interaction between the major components. Typically, users interact through the NIMROD portal; a single point of presence then directs traffic to one of three different components—NIMROD/G, which supports parameter studies and distributes the computations to the Grid, NIMROD/O, which performs optimization, and NIMROD/E, which uses experimental design techniques to scope parameter studies. Importantly, each of these components acts either as a user-level tool, or as middleware, depending on the client use. For example, NIMROD/G can interact directly with users using a Web-enabled interface, or can provide services to

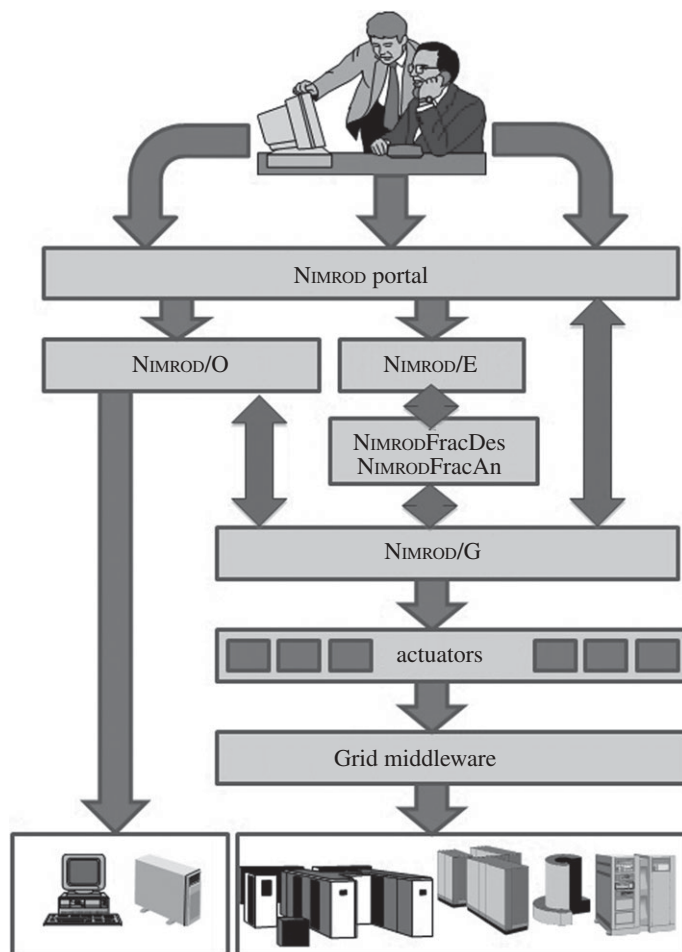


Figure 1. The NIMROD tool chain.

other software (such as NIMROD/E, NIMROD/O) via an application programming interface (API). Each of the applications discussed here leverages different aspects of the tools. In many cases, they used NIMROD/G to perform a crude sweep of the overall parameter space, and then launched NIMROD/O to refine the solutions.

An important aspect of the tool family is that they share a common specification language—which is written in a text document called a ‘plan’ file. This file contains details of the parameters and how to invoke the application, and is typically quite small. NIMROD/O plan files contain some additional information about which heuristics to use. This specifies the optimization algorithm, or algorithms, and associated settings. For example, the file may specify simulated annealing and the associated cooling regime. Starting points for iterative algorithms, also specified as NIMROD/O, can perform multiple concurrent searches. The NIMROD/E plan file contains information about which parameter combinations are to be estimated, and which are assumed negligible.

(c) NIMROD methodology

Each of the case studies discussed in the next section adopted the same overall methodology, regardless of which NIMROD tool was used. The following steps summarize this process:

- (1) *Testbed construction.* The user must decide which resources will be included in the Grid testbed, and configure NIMROD to use these. The NIMROD portal provides a number of high-level interfaces for making this fairly easy. NIMROD assumes that users already have accounts (and the necessary authentication) on each of the testbed resources.
- (2) *Software preparation.* Here the applications are compiled and tested on each of the Grid resources. This can be performed either manually by logging into each of the different remote resources or by using a tool like DISTANT (Goscinski & Abramson 2005), which manages the process through a single user-oriented client. Even when configured manually, it is possible to prepare the application binary on one machine and use NIMROD to distribute it to similar resources before execution.
- (3) *Determine which NIMROD tool to use.* As discussed, NIMROD has a number of different components. The user must select the most appropriate component, depending on whether a complete, partial or guided search is required.
- (4) *Describe how to execute the application, and which files are required for input and output.* These steps are described in the NIMROD plan file using a simple declarative language. NIMROD can be instructed to copy input files to each resource and return output files. Large output files can be left on remote resources for later analysis. NIMROD also managed parameter substitution via command line options or special control files.
- (5) *Determine the parameters and their ranges.* This will vary depending on the application requirements. These are then described in the NIMROD plan file using the ‘parameter’ keyword.
- (6) *Execute the experiment.* This is usually performed through the NIMROD portal, but it is also possible to use the NIMROD command line tools. Long-running experiments can be left unattended, and monitored using the NIMROD monitoring tools.
- (7) Analyse the results, possibly returning to step 5 to refine the parameter ranges.

In this paper, we illustrate combinations of various cardiac models with two members of the NIMROD tool family, namely NIMROD/G and NIMROD/O. NIMROD/G has proved to be of great value in exploring the dynamics of many computational models over the years, and is very relevant for exploring how cardiac models behave as their inputs change. NIMROD/O is very powerful for solving complex inverse problems, and can be combined with cardiac models to determine which parameters improve the skill of the models. These will be discussed in the next section.

4. Case studies

In this section, we show how we have combined the NIMROD tool set with two different modelling approaches to solve two problems in electrocardiology. The first concerns combining NIMROD with the Chaste platform discussed in §2*b*(iii). In this case, we want to observe the behaviour of the model as some critical input parameters are changed. The second example concerns an evaluation of the calcium dynamics of a Matlab cell model, and here we wish to compute parameter values that tune to the model to produce the correct output. Importantly, NIMROD is able to solve both of these problems even though they use different modelling approaches and have different sweep and search needs.

(a) Simulations of ion channel block effects on the ECG—a parameter sweep

Chaste was used to simulate propagation of the AP throughout the ventricles using the bidomain model. In this case, the Faber and Rudy model, a set of 25 ODEs (Faber & Rudy 2000), was used to describe ion channel kinetics and intracellular ionic concentration changes, characterizing the cellular AP at each node. Cardiac electrical activity was simulated over a computational finite-element mesh (discretized by 3 172 910 tetrahedral elements, and average distance between nodes was 250.74 μm) describing ventricular anatomy (figure 2). The ventricular wall was divided into three layers: epicardial, mid-myocardial and endocardial in relative proportions of 2:3:3. Owing to experimental evidence (McIntosh *et al.* 2000), each layer was characterized by different ionic current densities of the slow delayed rectifier (I_{Ks}) and transient outward (I_{to}) potassium currents.

Outside the heart region, the space was modelled as a passive resistive network where electrical propagation obeys the Laplace equation. The electrocardiogram (ECG) was extrapolated by plotting the potential over time at the surface of the control volume at a node located in the proximity of the heart. The control volume representing the body was a regular cuboid. The main advantage of this multi-scale approach is the ability to bridge the gap between micro-scale (ion channel kinetics) and macro-scale (ECG).

Here we present simulation results showing changes in the ECG caused by alterations in the conductance of the delayed rectifier K^+ current (I_{Kr}) (Bernabeu *et al.* 2009*a*), because of its importance in drug-induced arrhythmogenesis (see Fitton & Sorkin 1993; Pueyo *et al.* 2009). Six different degrees of I_{Kr} block were investigated and therefore six distinct parallel simulations (implementing the modelling framework presented in §2*b*(iii)) needed to be run. Within Chaste, the code of the cell model was altered by multiplying I_{Kr} by a variable scale factor whose value was read from the file. A text file (with extension .sub) containing the label of the variable to be swept over was uploaded to the NIMROD portal. Supplied with information regarding minimum, maximum and step values, NIMROD/G then performed the substitution of the label in the substitution file (.sub) and generated six files containing the actual desired values (one per file). These six files were placed in the appropriate locations for the Chaste code to read the value and assign it to the scale factor of the I_{Kr} variable. Six Chaste runs were therefore launched at once via NIMROD/G.

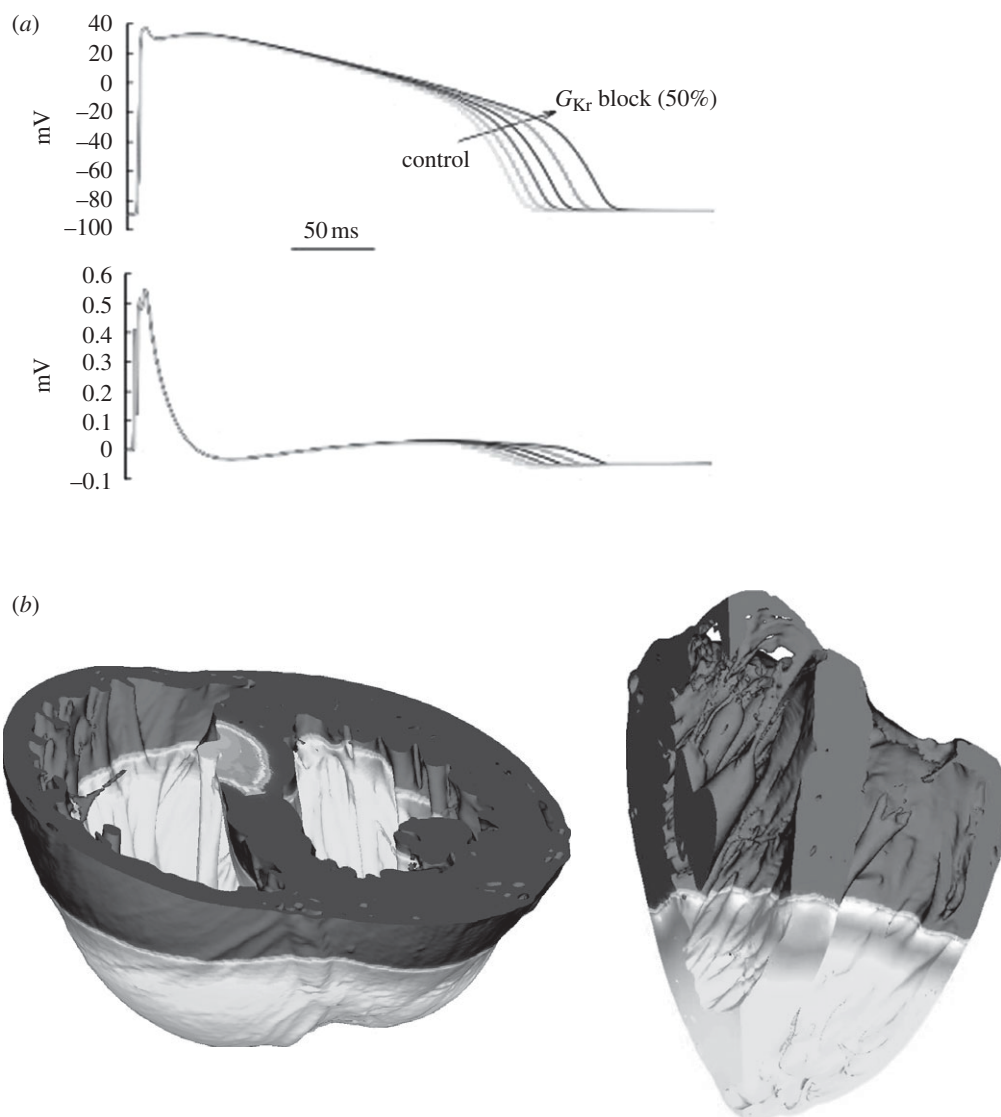


Figure 2. Results of whole ventricular simulations using the cardiac solver Chaste and NIMROD for parameter sweeping. (a) Effect of I_{K_r} block on the action potential and the ECG. (b) Propagation of electrical excitation in an anatomically based rabbit whole ventricular model.

Figure 2 presents the results of the experiment described above. The top panel shows different profiles for APs measured in an arbitrary cardiac cell located within the cardiac wall for different values (six in this case) of the I_{K_r} conductance (G_{K_r}). In the bottom panel, we can see how the ECG signal recorded at a node in the medium surrounding the heart varies as the K^+ current is gradually blocked. Both results are consistent and show that the prolongation of AP duration caused by the block of I_{K_r} directly translates into QT interval prolongation that can be appreciated from the ECG trace.

(b) *Enhanced calcium dynamics—an inverse problem*

At the single-cell level (a myocyte), the mechanisms of excitation–contraction coupling are closely regulated by calcium ion (Ca^{2+}) dynamics. Ca^{2+} entering the cell triggers the release of Ca^{2+} from the sarcoplasmic reticulum (SR), which is the organelle that stores calcium. The resulting rise of intracellular Ca^{2+} (Ca_i) activates the contraction of the cell. This phenomenon is known as Ca^{2+} -induced Ca^{2+} release (CICR). Local Ca^{2+} dynamics are characterized by the interactions within localized micro-domains (known as dyadic spaces) between L-type Ca^{2+} channels (LCCs) located on the transverse tubules (T-tubules), which are deep invaginations of the membrane into the cell, and closely opposed Ca^{2+} -release channels (known as ryanodine receptors, RyRs) located on the SR (figure 3). The SR is an extensive and well-organized network that repeatedly comes in contact with each T-tubule, so that the number of dyadic spaces throughout the cell has been estimated to be of the order of 50 000–300 000.

The local Ca^{2+} -release mechanisms are essential to reproduce the characteristic properties of the excitation–contraction coupling such as high-gain and graded Ca^{2+} release. However, most existing single-cell models lack the description of the biophysical nature of local Ca^{2+} dynamics.

In this case study, we present a methodology of how local Ca^{2+} dynamics can be efficiently incorporated into a single-cell model of ventricular myocyte in order to produce a biophysically accurate cell model (Sher *et al.* 2008). The two stages involved are (i) development of the Ca^{2+} subsystem and (ii) its incorporation into a single-cell model. The first stage is the generation of the local control CICR models (also known as the coupled LCC–RyR models) such as, for example, the ones that have been developed by Hinch *et al.* (2004) and Greenstein *et al.* (2006). The second stage, which is the focus of this case study, involves the incorporation of the coupled LCC–RyR models into a single-cell model. Specifically, the steps are as follows:

- The equations that describe Ca^{2+} dynamics in the original single-cell model (e.g. Noble 1998 model: Noble *et al.* 1998; figure 4) are substituted by equations of the biophysically detailed Ca^{2+} subsystem (e.g. baseline 40-state coupled LCC–RyR Greenstein 2006 model: Greenstein *et al.* 2006; figure 3), provided that units are modified accordingly;
- The parameters of the newly obtained single-cell model are refitted. This is done to ensure that the newly obtained single-cell model, which contains the replaced Ca^{2+} subsystem, is capable of reproducing the data of the original model. In particular, the specific aim is to fit the dynamics of the Ca^{2+} of the newly developed whole-cell model either to the dynamics of the Ca^{2+} of the original model (e.g. Noble 1998) and/or to the available experimental data (e.g. Ca_i transient, I – V curves, tail currents recorded from the voltage-clamp experiments, etc.). To achieve this, we need to *optimize* the parameters of the Ca^{2+} subsystem, or, in other words, to solve an inverse problem.

The models of the local Ca^{2+} dynamics is a system of ODEs of approximately 30–70 variables with up to 100 parameters. These ODEs do not exhibit stiffness, thus, time integrators such as the forward Euler integrator or a

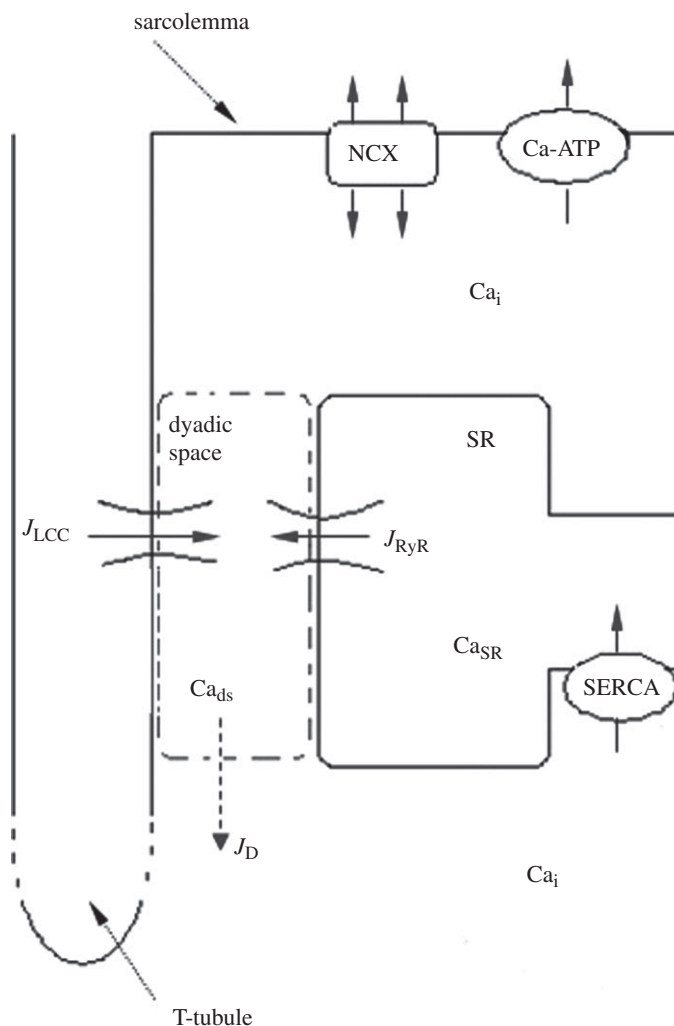


Figure 3. Local Ca^{2+} dynamics of the Greenstein 2006 model (Greenstein *et al.* 2006). Arrows represent the direction in which Ca^{2+} flows. Ca_i , Ca_{ds} and Ca_{SR} denote intracellular, dyadic and SR Ca^{2+} , respectively. The diagram illustrates the local control theory: LCCs and RyRs contribute to local J_{LCC} and J_{RyR} fluxes, respectively, within the dyadic space, J_D represents the diffusion of Ca^{2+} out of the dyad into the bulk myoplasm, sarco/endoplasmic reticulum Ca^{2+} -ATPase (SERCA) re-uptakes Ca^{2+} back into the SR, Ca^{2+} -ATPases and sodium–calcium exchangers (NCXs) pump Ca^{2+} out of the cell.

Runge–Kutta fourth-order method are appropriate to simulate these Markov models. The results presented below are simulated in Matlab 6.5 using an in-built ‘ode45’ solver—a one-step solver based on an explicit Runge–Kutta of fourth and fifth order, which is appropriate for non-stiff problems and has medium accuracy.

NIMROD/O provides a computationally effective manner of tuning the parameters and examining their effects within the newly developed models. NIMROD/O offers a variety of optimization methods, such as ‘subdivision search’

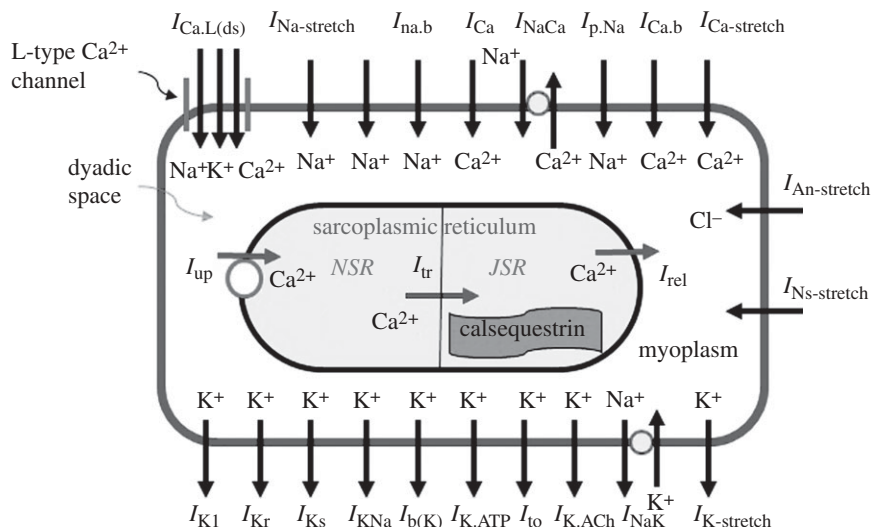


Figure 4. Schematic of the single-cell Noble 1998 model (Noble *et al.* 1998) (Adapted from a diagram at www.cellml.org model repository.) Arrows represent the direction in which the ions flow and the label of the corresponding ionic current is located above or below the arrow. The model includes four Ca^{2+} compartments, which are the intracellular, dyadic, network SR (NSR) and junctional SR (JSR). This model is an example of the deterministic non-common pool ventricular model that uses phenomenological description of Ca^{2+} in the dyadic space and Ca^{2+} -related currents. While such a model succeeds in producing graded SR Ca^{2+} release, it lacks the mechanistic description of local SR Ca^{2+} release, i.e. the stochastic interaction between LCCs and RyRs within the dyadic spaces.

and downhill-type search methods, etc. The simulation results presented below are obtained using the downhill simplex method of Nelder and Mead. The optimal set of parameters, calculated using the simplex method, is obtained by fitting the AP, Ca_i transient and I_{CaL} current, with the objective function calculated using the least-squares approximations.

The direct incorporation of the canine 40-state Greenstein 2006 coupled LCC-RyR model (Greenstein *et al.* 2006) into the Noble 1998 guinea pig model (Noble *et al.* 1998) (figure 4) results in a distorted electrical behaviour of the cell, such as a significant second peak in Ca_i transient and a pronounced plateau phase in an action potential (compare dashed and dotted curves in figure 5). An optimized set of parameters obtained using NIMROD/O significantly improves the dynamical behaviour of the modified Noble 1998 model (solid curve). Importantly, the new set of parameters, which falls within the physiologically acceptable ranges, results in an elimination of the second peak in Ca_i transient (middle panel in figure 5). Further, the results demonstrate that NIMROD/O provides a convenient, user-friendly framework for tuning the parameters in the cardiac cell models in an efficient computational manner by taking advantage of parallel batches of evaluations. This study provides a valuable platform for future incorporation of the biophysically detailed Ca^{2+} subsystems into whole-cell models of various species.

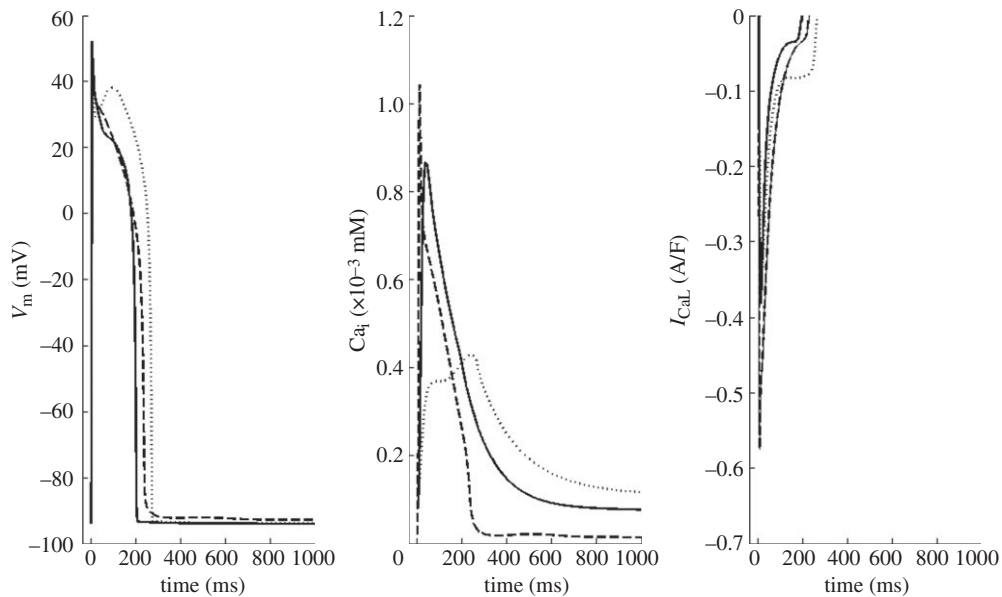


Figure 5. Cardiac model output: the Noble 1998 guinea pig ventricular model (Noble *et al.* 1998) modified to include local Ca^{2+} dynamics. Dashed curve represents the Noble 1998 model. Dotted curve shows the modified Noble 1998 model that incorporates the 40-state coupled LCC–RyR Greenstein 2006 model (Greenstein *et al.* 2006). Solid curve denotes the Noble 1998 model modified to include the Greenstein 2006 model Ca^{2+} dynamics with an optimized set of parameters.¹

5. Conclusions

In this paper, we have discussed various techniques used to model the heart—from a single cell to a complete organ. We have also briefly described approaches used for constructing models, ranging from applications written in legacy languages, through fourth-generation languages, to complete domain-specific platforms. We showed that in order to understand the behaviour of these models, we may need to perform parameter sweeps, changing critical parameters in order to observe the effects on the system. In one case study, we used the NIMROD/G tool to perform an example of this, and in a second, we showed that the more selective and efficient search functions in NIMROD/O allow us to compute parameter values that optimize some objective function. Although we did not demonstrate the third member of the NIMROD family, NIMROD/E, it has also been applied to solving complex problems in cardiac models in the past (Abramson *et al.* 2009).

¹The optimal set of parameters (corresponding to the solid curve) is as follows: an increase in the maximum rate of the SERCA pump (1.4-fold), an increase in the conductance of RyR (3.5-fold) and LCC (1.5-fold) channels, modified constants of the 10-state LCC Markov model (the transition rate to the Ca^{2+} -dependent inactivation (CDI) state by 1.33-fold; the transition rate out of the CDI-state constant by 0.34-fold; the transition rate out of the closed state by 2.2-fold; the transition state out of the open state by 0.59-fold) and of the four-state RyR Markov model (the transition rate into the open state from CDI-state 2 by 4.58-fold; the transition rate out of the open state into CDI-state 2 by 0.79-fold; the transition rate into the open state from CDI-state 4 by 2.4-fold; the transition rate out of the closed state by 2.1-fold).

Currently, we are building a new version of NIMROD, NIMROD/K, that better supports running sweeps over workflows in which multiple tools are connected to solve a single modelling problem (Abramson *et al.* 2008, 2009). NIMROD/K leverages KEPLER (Altintas *et al.* 2005), a multi-institution collaboration, to build a scientific workflow system for the grid. NIMROD/K adds novel parallel execution semantics to KEPLER, and also adds the existing search and sweep operations from the other NIMROD tools. We have started to apply NIMROD/K to cardiac models with considerable success (Abramson *et al.* 2009), and will develop more complex scientific workflows for cardiac science in the future.

This research has been supported by the Australian Research Council under the Discovery grant scheme, the EPSRC e-Science Pilot Project in Integrative Biology GR/S72023/01, UK, the UK EPSRC (EP/F011628/1 to M.O.B.), the European Commission preDiCT grant (DG-INFSo—224381 to A.C. and M.O.B), and an MRC Career Development Award (to B.R.). We also acknowledge our colleagues in the MeSSAGE Laboratory at Monash University for their support.

References

- Abramson, D., Sosic, R., Giddy, J. & Hall, B. 1995 Nimrod: a tool for performing parametrised simulations using distributed workstations. In *Proc. 4th IEEE Int. Symp. on High Performance Distributed Computing (HPDC 1995)*, Pentagon City, VA, 1–4 August, pp. 112–121. New York, NY: IEEE Computer Society. (doi:10.1109/HPDC.1995.518701)
- Abramson, D., Lewis, A. & Peachy, T. 2000a Nimrod/O: a tool for automatic design optimization. In *Algorithms and Architectures for Parallel Processing (ICA3PP)*, *Proc. 4th Int. Conf., Hong Kong, 11–13 December* (eds A. Goscinski, H. H. Ip & W. Jia). Singapore: World Scientific.
- Abramson, D., Giddy, J. & Kotler, L. 2000b High performance parametric modeling with Nimrod/G: killer application for the global grid? In *Proc. 14th Int. Parallel and Distributed Processing Symp. (IPDPS'00)*, Cancun, Mexico, 1–5 May, pp. 520–528. Los Alamitos, CA: IEEE Computer Society. (doi:10.1109/IPDPS.2000.846030)
- Abramson, D., Lewis, A., Peachey, T. & Fletcher, C. 2001 An automatic design optimization tool and its application to computational fluid dynamics. In *Proc. 2001 ACM/IEEE Conf. on Supercomputing, Denver, CO, 10–16 November*, paper 47. Los Alamitos, CA: IEEE Computer Society. (doi:10.1109/SC.2001.10054)
- Abramson, D., Enticott, C. & Altintas, I. 2008 Nimrod/K: towards massively parallel dynamic grid workflows. In *Proc. 2008 ACM/IEEE Conf. on Supercomputing, Austin, TX, 15–21 November*. New York, NY: ACM. (doi:10.1145/1413370.1413395)
- Abramson, D., Bethwaite, B., Enticott, C., Garic, S., Peachey, T., Michailova, A., Amirrazi, S. & Chitters, R. 2009 Robust workflows for science and engineering. In *Proc. 2nd Workshop on Many-Task Computing on Grids and Supercomputers*, at *Conf. on High Performance Networking and Computing, Portland, OR, 16 November*. New York, NY: ACM. (doi:10.1145/1646468.1646469)
- Altintas, I., Birnbaum, A., Baldrige, K., Sudholt, W., Miller, M., Amoreira, C., Potier, Y. & Ludaescher, B. 2005 A framework for the design and reuse of grid workflows. In *Scientific Applications of Grid Computing, First Int. Workshop (SAG 2004)*, Beijing, China, 20–24 September 2004. Lecture Notes in Computer Science, no. 3458, pp. 120–133. Berlin, Germany: Springer. (doi:10.1007/11423287_11)
- Bernabeu, M., Bishop, M., Pitt-Francis, J., Gavaghan, D., Grau, V. & Rodriguez, B. 2008 High performance computer simulations for the study of biological function in 3D heart models incorporating fibre orientation and realistic geometry at para-cellular resolution. In *Computers in Cardiology 2008, Bologna, Italy, 14–17 September*, vol. 35, pp. 721–724. See <http://www.cinc.org/archives/2008/pdf/0721.pdf>.

- Bernabeu, M. O. *et al.* 2009a Grid computing simulations of ion channel block effects on the ECG using 3D anatomically-based models. In *Computers in Cardiology 2009, Park City, UT, 13–16 September*, vol. 36, pp. 213–216. See <http://www.cinc.org/archives/2009/pdf/0213.pdf>.
- Bernabeu, M. O. *et al.* 2009b CHASTE: incorporating a novel multi-scale spatial and temporal algorithm into a large-scale open source library. *Phil. Trans. R. Soc. A* **367**, 1907–1930. (doi:10.1098/rsta.2008.0309)
- Capener, C. E., Kim, H. J., Arinaminpathy, Y. & Sansom, M. S. P. 2002 Ion channels: structural bioinformatics and modelling. *Human Mol. Genet.* **11**, 2425–2433. (doi:10.1093/hmg/11.20.2425)
- Cary, J. R., Shasharina, S. G., Cummings, J. C., Reynders, J. V. & Hinker, P. J. 1997 Comparison of C++ and Fortran 90 for object-oriented scientific programming. *Comput. Phys. Commun.* **105**, 20–36. (doi:10.1016/S0010-4655(97)00043-X)
- Cooper, J., McKeever, S. & Garny, A. 2006 *On the application of partial evaluation to the optimisation of cardiac electrophysiological simulations*, pp. 12–20. New York, NY: ACM.
- Faber, G. M. & Rudy, Y. 2000 Action potential and contractility changes in $[\text{Na}^+]_i$ overloaded cardiac myocytes: a simulation study. *Biophys. J.* **78**, 2392–2404. (doi:10.1016/S0006-3495(00)76783-X)
- Fitton, A. & Sorkin, E. M. 1993 Sotalol: an updated review of its pharmacological properties and therapeutic use in cardiac arrhythmias. *Drugs* **46**, 678–719. (doi:10.2165/00003495-199346040-00007)
- Foster, I. & Kesselman, C. (eds) 2003 *The grid: blueprint for a new computing infrastructure*, 2nd edn. Los Altos, CA: Morgan Kaufmann.
- Goscinski, W. & Abramson, D. 2005 Application deployment over heterogeneous grids using distributed ant. In *Proc. 1st Int. Conf. on e-Science and Grid Computing, Melbourne, Australia, 5–8 December*, pp. 361–368. Los Alamitos, CA: IEEE Computer Society. (doi:10.1109/E-SCIENCE.2005.19)
- Greenstein, J. L., Hinch, R. & Winslow, R. L. 2006 Mechanisms of excitation–contraction coupling in an integrative model of the cardiac ventricular myocyte. *Biophys. J.* **90**, 77–91. (doi:10.1529/biophysj.105.065169)
- Hinch, R., Greenstein, J. L., Tanskanen, A. J., Xu, L. & Winslow, R. L. 2004 A simplified local control model of calcium-induced calcium release in cardiac ventricular myocytes. *Biophys. J.* **87**, 3723–3736. (doi:10.1529/biophysj.104.049973)
- Keener, J. & Sneyd, J. 2001 *Mathematical physiology*. Berlin, Germany: Springer.
- Kerckhoffs, R. C. P., Healy, S. N., Usyk, T. P. & McCulloch, A. D. 2006 Computational methods for cardiac electromechanics. *Proc. IEEE* **94**, 769–783. (doi:10.1109/JPROC.2006.871772)
- McIntosh, M. A., Cobbe, S. M. & Smith, G. L. 2000 Heterogeneous changes in action potential and intracellular Ca^{2+} in left ventricular myocyte sub-types from rabbits with heart failure. *Cardiovasc. Res.* **45**, 397–409. (doi:10.1016/S0008-6363(99)00360-0)
- Noble, D. 1960 Cardiac action and pacemaker potentials based on the Hodgkin–Huxley equations. *Nature* **188**, 495–497. (doi:10.1038/188495b0)
- Noble, D. 1962 A modification of the Hodgkin–Huxley equations applicable to Purkinje fibre action and pacemaker potentials. *J. Physiol.* **160**, 317–352.
- Noble, D., Varghese, A., Kohl, P. & Noble, P. 1998 Improved guinea-pig ventricular cell model incorporating a dyadic space, I_{Kr} and I_{Ks} , and length- and tension-dependent processes. *Can. J. Cardiol.* **14**, 123–134.
- Peachey, T. C., Diamond, N. T., Abramson, D. A., Sudholt, W., Michailova, A. & Amirriazi, S. 2008 Fractional factorial design for parameter sweep experiments using Nimrod/E. *J. Sci. Prog.* **16**, 217–230.
- Pitt-Francis, J. *et al.* 2009 Chaste: a test-driven approach to software development for biological modelling. *Comput. Phys. Commun.* **180**, 2452–2471. (doi:10.1016/j.cpc.2009.07.019)
- Plank, G., Prassl, A., Hofer, E. & Trayanova, N. A. 2008 Evaluating intramural virtual electrodes in the myocardial wedge preparation: simulations of experimental conditions. *Biophys. J.* **94**, 1904–1915. (doi:10.1529/biophysj.107.121343)

- Pueyo, E., Corrias, A., Gavaghan, D., Burrage, K. & Rodriguez, B. 2009 Role of stochastic I_{Ks} gating kinetics in arrhythmogenesis. *Heart Rhythm* **6** (Suppl., May), S296.
- Rudy, Y. & Silva, J. R. 2006 Computational biology in the study of cardiac ion channels and cell electrophysiology. *Q. Rev. Biophys.* **39**, 57–116. (doi:10.1017/S0033583506004227)
- Sher, A., Abramson, D., Enticott, C., Garic, S., Gavaghan, D., Noble, D., Noble, P. & Peachey, T. 2008 Incorporating local Ca^{2+} dynamics into single cell ventricular models In *Computational Science—ICCS 2008, Part I, Proc. 8th Int. Conf., Kraków, Poland, 23–25 June*. Lecture Notes in Computer Science, no. 5101, pp. 66–75. Berlin, Germany: Springer. (doi:10.1007/978-3-540-69384-0_12)