

Model Reduction for Initial Value ODEs

Antonietta Ambuehl^a, Jonathan P Whiteley^{a,*}

^a*Department of Computer Science, Wolfson Building, Parks Road, Oxford OX1 3QD,
United Kingdom*

Abstract

Many physical phenomena in biology and physiology are described by mathematical models that comprise a system of initial value ordinary differential equations. Each differential equation may often be written as the sum of several terms, where each term represents a different physical entity. A wide range of techniques, ranging from heuristic observation to mathematically rigorous asymptotic analysis, may be used to simplify these equations allowing the identification of the key phenomena responsible for a given observed behaviour. In this study we extend an algorithm for automatically simplifying systems of initial value ordinary differential equations (Whiteley, *Mathematical Biosciences*, vol. 225, pp. 44-52, 2010) that is based on *a posteriori* analysis of the full system of equations. Our extensions to the algorithm make the following contributions: (i) each equation in a system of differential equations may be written as a finite sum of contributions (including the derivative term), and any one of these terms may be neglected (if it is appropriate to do so) in the simplified model; and (ii) a simplified model is generated that allows accurate prediction of one or more components of the solution at all times. These extensions are illustrated using examples

*Corresponding author

Email address: `jonathan.whiteley@cs.ox.ac.uk` (Jonathan P Whiteley)

drawn from enzyme kinetics and cardiac electrophysiology.

Keywords: *a posteriori analysis*, model reduction, initial value problem

1. Introduction

Many phenomena in biology and physiology are modelled by systems of initial value ordinary equations of the form

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, t), \quad \tau_0 < t < \tau_1, \quad \mathbf{u}(\tau_0) = \mathbf{u}_0, \quad (1)$$

where \mathbf{u} is a vector representing the solution of the system of differential equations, t is the independent variable, $\mathbf{f}(\mathbf{u}, t)$ is a specified vector-valued function often referred to as the “right hand side of the differential equation”, \mathbf{u}_0 is a specified vector containing the initial values for the system, and τ_0, τ_1 are specified constants. In detailed models, the vector \mathbf{u} may have many components, and each component of \mathbf{f} may comprise several separate terms that model processes that act on a wide range of scales. When using such a detailed model it can be useful to derive a simpler model that may be used identify the key components that cause a particular experimental observation.

We emphasise that, in this study, we are interested in simplified models generated from models of the form given by Eq. (1) that are sufficient to describe one specific experimental observation, and that a different simplified model may be derived to explain the phenomena responsible for a different experimental observation. For example, single cell models of cardiac electrophysiology are usually described by a system of equations such as Eq. (1) where the vector \mathbf{u} contains tens of components [12]. The quantities modelled by the governing equations represent the transmembrane potential, ion concentrations, and gating variables that control the flux of ions (and

therefore the current) across the cell membrane. Many cardiac cell models exhibit excitability, i.e. they require a stimulus that is sufficiently large to generate the variation in transmembrane potential, known as an action potential, that instigates the series of biochemical reactions that is required for the cell to perform many of its expected tasks. When using a cardiac cell model of the form given by Eq. (1), we may hope to derive a simpler model that accurately predicts whether a given stimulus is sufficiently large to generate an action potential for the original model. However, if we were to instead derive a simpler model from the same cardiac cell model that accurately predicts the original model's response to a drug over a longer time then we would expect that the two simpler models would contain different components of the original model.

The terms in an initial value problem of the form given by Eq. (1) that are principally responsible for an observed feature of the solution can be derived by applied mathematics techniques. Typically the equations are first nondimensionalised. This often results in some terms in the governing equations being multiplied by a small dimensionless parameter. Asymptotic techniques can then be used to justify neglecting some of the terms that appear in the system of equations, or the identification of different and widely separated scales on which some quantities vary. These predictions may then be used to generate simpler models, identifying the cause of some feature of the solution of the whole model. Many examples exist of the use of this approach; see, for example, [4] for a selection of applications.

The analytic approach described above requires significant mathematical expertise — concepts such as nondimensionalisation and asymptotic expansions are usually taught only to advanced undergraduate students, or to graduate students, on degree courses with a substantial mathematical

component. As a consequence, researchers who have expertise in different fields, and who derive systems of differential equations, may be unable to fully analyse the models that they develop. In an attempt to automate the process of model reduction recent publications [9, 14] have derived an algorithm to automate the approach of model simplification using the concept of *a posteriori* analysis of the solution of the full system of equations. The concept of *a posteriori* analysis is established in the field of finite element methods for the numerical solution of differential equations; it has been used both as an error estimator, and to generate a computational mesh that guarantees that the finite element solution of a system of differential equations is accurate to within a tolerance specified by the user [1, 5, 6]. In the *a posteriori* model reduction algorithm described in [14], the algorithm is seeded with a simplified model where the right hand side of all components of Eq. (1) are set to zero. A *a posteriori* analysis of the solution of both the full model and the simplified model then guides the algorithm to generate a simplified model where components of the vector valued function on the right hand side of Eq. (1) are included for appropriate intervals of t , and the algorithm terminates when the simplified model predicts the user-prescribed linear functional to within the tolerance specified by the user. We emphasise that the *a posteriori* approach is designed to identify contributions from the right hand side of Eq. (1) that are responsible for a given observation; as a consequence some components of the right hand side with a large magnitude may be neglected if this component is not a significant contribution to the observation of interest.

Other computational tools exist that automatically infer information about the underlying features of a system of equations such as Eq. (1). These techniques have a different focus to that considered in [14], and an-

answer a slightly different question to that addressed in this paper. For example, Clewley *et al.*, [3], have developed a technique that automatically identifies the dominant time scales at each point in time, allowing a system of differential-algebraic equations to be derived as an approximation to the full system of equations. A different approach was followed by Wirth *et al.*, [15]. These authors developed a method for generating a suitable basis of functions for the solution of Eq. (1). Although not an adaptive finite element scheme such as those described by [1, 5, 6], this method is in a similar spirit to these approaches, focusing on a suitable basis for the solution rather than on identifying terms in the governing equations that may be neglected.

The *a posteriori* model simplification approach described above has only been applied for linear functionals of the solution, such as one component of the solution at some point in time, or the average of one component of the solution over some interval of values of t . A more useful approach, which would follow more closely the aims of the analytic approach to model simplification described above, would be to generate a simplified model that accurately captures one or more components of the solution for all values of t . It would also be useful to have the capability to decompose each component of the right hand side of Eq. (1) into the sum of two or more terms and, when deriving the simplified model, to neglect any one of the components in this sum, or the time derivative. In this paper we extend the model reduction algorithm presented in [14] to include all of these features, and illustrate the application of this algorithm.

The remainder of this paper is structured as follows. In Section 2 we extend the algorithm presented by [14] to allow each equation in the system of ordinary differential equations to be decomposed into an arbitrary number of components, and for any one of these terms to be included or omitted

from the simplified model as appropriate. We also explain how the algorithm may be extended so that one or more components of the solution is accurately calculated for all values of t . The application of the algorithm is then illustrated in Section 3 by application to two model examples — enzyme kinetics and single cell cardiac electrophysiology. The enzyme kinetics examples are chosen because established mathematical analysis of this model already exists [11], allowing us to compare our automated technique with the technique it aims to complement, namely mathematical analysis. The cardiac electrophysiology examples allow us to demonstrate that our technique automatically generate information that is already known about this model, thus illustrating its effectiveness in this case. Finally, in Section 4 we summarise the conclusions from this paper.

2. The algorithm

In this section we explain how the algorithm presented in [14] may be extended to include the additional features described in Section 1. First we decompose component i on the right hand side of Eq. (1) into M_i terms f_{ij} , $j = 1, 2, \dots, M_i$ so that the initial value problem is given by

$$\frac{du_i}{dt} = \sum_{j=1}^{M_i} f_{ij}(\mathbf{u}, t), \quad \tau_0 < t < \tau_1, \quad i = 1, 2, \dots, N, \quad (2)$$

$$\mathbf{u}(\tau_0) = \mathbf{u}_0. \quad (3)$$

We will use a mesh of points $t_0, t_1, t_2, \dots, t_{N_{\text{mesh}}}$, where $t_0 = \tau_0$ and $t_{N_{\text{mesh}}} = \tau_1$, when calculating the numerical solution of both the full model and the simplified model, where this mesh has been chosen to be fine enough to calculate the solution of the full model sufficiently accurately. We will develop an algorithm that generates a simplified model where values of any specified

component of the solution are within a given tolerance of the solution of the full model at each of the points in the mesh $t_1, t_2, t_3, \dots, t_{N_{\text{mesh}}}$.

In Section 1 we explained that we will develop the model reduction algorithm to allow the derivative term in Eq. (2) to be neglected if it makes a sufficiently small contribution to calculating the quantity we are interested in. Neglecting a derivative may result in the simplified model having a discontinuous solution, and so the analysis presented in [14] must be extended to take account of these discontinuities. We therefore begin, in Section 2.1, by deriving a bound on the error between the full and simplified models for the case where the simplified model is permitted to contain discontinuities. In Section 2.2 we explain how this error bound may underpin an algorithm that allows the simplified model to accurately capture one or more components of the solution at all values of t in the interval $\tau_0 < t < \tau_1$.

2.1. An error bound

Suppose that the quantity of interest from the IVP given by Eqs. (2) and (3) is the value of u_p at one specified time $t = t_Q$ where t_Q is a node in the mesh used to calculate the numerical solution, i.e. we use the solution to allow us to compute

$$J(\mathbf{u}) = u_p(t_Q), \tag{4}$$

where p takes one of the values $1, 2, \dots, N$, and Q takes one of the values $1, 2, \dots, N_{\text{mesh}}$.

Let $\mathbf{U}(t)$ be piecewise continuous for $t_0 < t < t_Q$, with possible discontinuities at the mesh points $t = t_0, t_1, \dots, t_Q$; in later sections $\mathbf{U}(t)$ will represent the solution of the simplified model, and it may be useful to bear

this in mind in the analysis that follows. Define $A_{ijk}(t)$ by

$$A_{ijk}(t) = \int_0^1 \frac{\partial f_{jk}}{\partial u_i}(\boldsymbol{\alpha}(s, t), t) \, ds, \quad i, j = 1, 2, \dots, N, \quad k = 1, 2, \dots, M_j,$$

where the vector $\boldsymbol{\alpha}(s, t)$ is defined by

$$\boldsymbol{\alpha}(s, t) = s\mathbf{u}(t) + (1 - s)\mathbf{U}(t).$$

We then have, for $j = 1, 2, \dots, N$, $k = 1, 2, \dots, M_j$,

$$\begin{aligned} \sum_{i=1}^N A_{ijk}(t) (u_i(t) - U_i(t)) &= \sum_{i=1}^N \int_0^1 \frac{\partial f_{jk}}{\partial u_i}(\boldsymbol{\alpha}(s, t), t) (u_i(t) - U_i(t)) \, ds \\ &= \sum_{i=1}^N \int_0^1 \frac{\partial f_{jk}}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial s} \, ds \\ &= \int_0^1 \frac{df_{jk}}{ds} \, ds \\ &= f_{jk}(\mathbf{u}, t) - f_{jk}(\mathbf{U}, t). \end{aligned}$$

Suppose ϕ satisfies the dual problem, for $i = 1, 2, \dots, N$:

$$\frac{d\phi_i}{dt} = - \sum_{j=1}^N \sum_{k=1}^{M_j} A_{ijk}(t) \phi_j(t), \quad \tau_0 < t < t_Q, \quad i = 1, 2, \dots, N, \quad (5)$$

$$\phi_i(t_Q) = \begin{cases} 1, & i = p, \\ 0, & i \neq p. \end{cases} \quad (6)$$

We then have

$$\begin{aligned}
0 &= \sum_{i=1}^N \int_{t_0}^{t_Q} (u_i - U_i) \left(\frac{d\phi_i}{dt} + \sum_{j=1}^N \sum_{k=1}^{M_j} A_{ijk} \phi_j \right) ds \\
&= \sum_{q=1}^Q \sum_{i=1}^N \int_{t_{q-1}}^{t_q} (u_i - U_i) \left(\frac{d\phi_i}{dt} + \sum_{j=1}^N \sum_{k=1}^{M_j} A_{ijk} \phi_j \right) ds \\
&= \sum_{q=1}^Q \sum_{i=1}^N \left([(u_i - U_i) \phi_i]_{t_{q-1}}^{t_q} + \right. \\
&\quad \left. \int_{t_{q-1}}^{t_q} - \left(\frac{du_i}{dt} - \frac{dU_i}{dt} \right) \phi_i + \sum_{k=1}^{M_i} (f_{ik}(\mathbf{u}, t) - f_{ik}(\mathbf{U}, t)) \phi_i ds \right) \\
&= (u_p(t_Q) - U_p(t_Q)) + \\
&\quad \sum_{q=1}^Q \sum_{i=1}^N \left([U_i]_{t_{q-1}}^{t_q^+} \phi_i(t_{q-1}) + \int_{t_{q-1}}^{t_q} \left(\frac{dU_i}{dt} - \sum_{k=1}^{M_i} f_{ik}(\mathbf{U}, t) \right) \phi_i ds \right) \quad (7)
\end{aligned}$$

where

$$U_i(t_q^-) = \lim_{s \rightarrow 0} U_i(t_q - s), \quad U_i(t_q^+) = \lim_{s \rightarrow 0} U_i(t_q + s),$$

and s is positive. We set $U_i(t_0^-) = \mathbf{u}_{0i}$, where \mathbf{u}_0 is the vector of initial conditions. Using Eq. (7) we then obtain an expression for the difference between U_p and u_p at $t = t_Q$:

$$\begin{aligned}
U_p(t_Q) - u_p(t_Q) &= \\
&\sum_{i=1}^N \sum_{q=0}^{Q-1} \left([U_i]_{t_q^-}^{t_{q+1}^+} \phi_i(t_q) + \int_{t_q}^{t_{q+1}} \left(\frac{dU_i}{dt} - \sum_{k=1}^{M_i} f_{ik}(\mathbf{U}, t) \right) \phi_i ds \right). \quad (8)
\end{aligned}$$

This expression for the difference between the solution of component p the full model and the solution of component p of the simplified model at time t_Q highlights one of the key features of this approach, namely that components other than component p that need to be accurately calculated to calculate $u_p(t_Q)$ are identified, along with the time intervals where they need to be calculated accurately.

2.2. Implementation of the algorithm

We now describe the algorithm that uses the error bound given by Eq. (8) to generate a simplified model corresponding to the governing equations given by Eqs. (2) and (3). We define the set S_{ij} , $i = 1, \dots, N$, $j = 1, \dots, M_i$, to be the set of time intervals for which the term f_{ij} is included in the simplified model, and the set T_i , $i = 1, \dots, N$, to be the set of time intervals for which the time derivative du_i/dt is included in the simplified model. We then define the indicator functions $I_{S_{ij}}(t)$ and $I_{T_i}(t)$ by

$$I_{S_{ij}}(t) = \begin{cases} 1, & t \in S_{ij}, \quad i = 1, \dots, N, \quad j = 1, \dots, M_i, \\ 0, & t \notin S_{ij}, \quad i = 1, \dots, N, \quad j = 1, \dots, M_i, \end{cases}$$

$$I_{T_i}(t) = \begin{cases} 1, & t \in T_i, \quad i = 1, \dots, N, \\ 0, & t \notin T_i, \quad i = 1, \dots, N. \end{cases}$$

The solution of the simplified model is denoted by $\mathbf{U}(t)$, and satisfies

$$I_{T_i}(t) \frac{dU_i}{dt} = \sum_{j=1}^{M_i} I_{S_{ij}}(t) f_{ij}(\mathbf{U}, t), \quad \tau_0 < t < \tau_1, \quad i = 1, 2, \dots, N, \quad (9)$$

$$\mathbf{U}(\tau_0) = \mathbf{u}_0. \quad (10)$$

We insist that at least one component of every equation must be included in the simplified model at every time t . As a consequence, for each $i = 1, 2, \dots, N$ and all times t , at least one of $I_{S_{ij}}(t)$, $j = 1, 2, \dots, M_i$, and $I_{T_i}(t)$ must take the value 1. We initially seed the simplified model so that this condition is met; this is more flexible than that described by [14] where the model was seeded where S_{ij} was the empty set for $i = 1, 2, \dots, N$, $j = 1, 2, \dots, M_i$, and T_i was the whole time interval $\tau_0 < t < \tau_1$ for $i = 1, 2, \dots, N$.

Implementation of the algorithm is similar in spirit to that described in [14]. There are, however, significant differences: (i) the right hand side of

Eq. (9) is now decomposed into a sum of terms rather than a single term; (ii) the simplified model may be seeded so that it initially contains only one or more components of the right hand side of Eq. (9) and not the time derivative; and (iii) the simplified model generated is sufficiently accurate at all points t_Q , $Q = 1, 2, \dots, N_{\text{mesh}}$ rather than at a single point.

The algorithm proceeds as follows in the case where accuracy of only one component of the solution is required. Given a tolerance and an equation of interest, we first solve Eqs. (2) and (3) to calculate $\mathbf{u}(t)$, the solution of the full model. The model simplification algorithm may be described inductively; for each t_Q , where $Q = 1, 2, \dots, N_{\text{mesh}}$, we assume that a simplified model already exists such that the specified component of the solution is calculated sufficiently accurately for all times t_q , where $q = 1, 2, \dots, Q - 1$. To ensure that the specified component in the simplified model is also sufficiently accurate at time t_Q we then execute the following steps.

1. Solve Eqs. (9) and (10) to calculate $\mathbf{U}(t)$.
2. Use Eq. (4) to see if the required component of the simplified model is sufficiently accurate at time t_Q ; stop if this condition is met.
3. Solve Eqs. (5) and (6) to calculate $\phi(t)$.
4. Scan over all values of i, k, q that appear in Eq. (8) to identify the term with the largest absolute value that is not already included in the simplified model. If this term is the jump term in U_i or the time derivative then include the time derivative in this time interval in the simplified model. If this term is one of the f_{ik} terms then include this term in the simplified model.
5. Repeat steps 1-5.

By applying this algorithm sequentially at the times $t_1, t_2, t_3, \dots, t_{N_{\text{mesh}}}$, this

algorithm successively adds terms into the simplified model in a manner that component p of the simplified model is sufficiently accurate at all points in the mesh $t_1, t_2, t_3, \dots, t_{N_{\text{mesh}}}$.

If accuracy of more than one component of the solution is required at all points in time we modify the steps listed above so that steps 1-5 are repeated for each component of the solution for which accuracy is required.

3. Example applications of the algorithm

We now present the practical application of the theory above using three examples. The first example is a simple model problem that allows us to demonstrate the insight that may be gained by initially seeding the simplified model with the right hand side of the differential equation rather than the time derivatives. This motivates the second example, which is a model of enzyme kinetics. The third and final example is a model of cardiac cell electrophysiology, that illustrates the use of decomposing the right hand side of the differential equation into a sum of more than one term.

3.1. A model example

We illustrate the benefits of initially seeding the simplified model using the right hand side of the differential equation with the aid of the initial value problem defined by

$$\frac{du}{dt} = \frac{\cos t - u}{\epsilon}, \quad 0 < t < 10. \quad (11)$$

$$u(0) = 0, \quad (12)$$

This initial value problem has solution

$$u = \frac{1}{1 + \epsilon^2} \left(\cos t + \epsilon \sin t - e^{-t/\epsilon} \right). \quad (13)$$

Suppose $\epsilon \ll 1$. For times $t \gg \epsilon$ we may approximate this solution by $u = \cos t$, and we would like the simplified model to reflect this. We apply the model reduction algorithm to this differential equation, setting $\epsilon = 0.01$ and seeding the simplified model so that initially only the time derivatives are included. A uniform mesh with a spacing of 0.001 is used, and the numerical solution of the differential equation is calculated using the backward Euler method (see, for example, [13]). We demand that the solution of the simplified model at each point of the mesh is within 0.1 of the solution of the full model. The solution of both the simplified model (solid line) and the broken model (broken line) is plotted for the whole time interval in Figure 1(a), and for small times in Figure 1(b). It is evident that the simplified model is indeed a good approximation to the true solution.

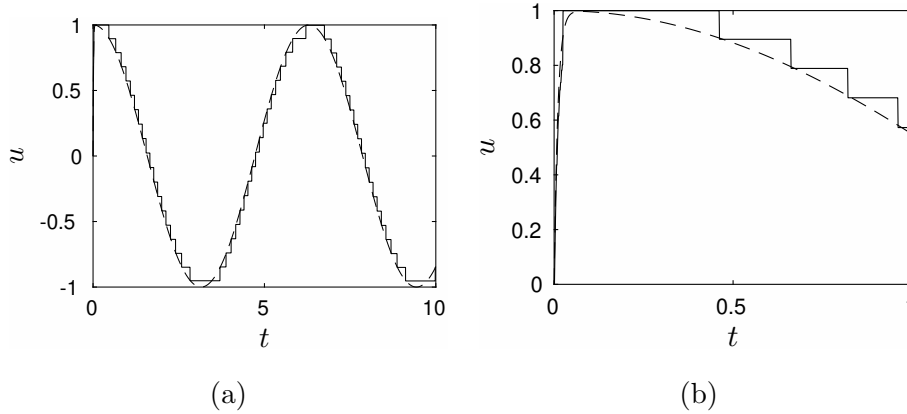


Figure 1: The simplified model corresponding to Eqs. (11)-(12) where the simplified model is seeded by initially including only the time derivatives. The solution of the simplified model is represented by the solid line, and the full model is represented by the broken line. (a) shows the whole time interval $0 < t < 10$; (b) shows only earlier times $0 < t < 1$.

In Figure 2 we plot the intervals in which the derivative is included in the simplified model, and the intervals in which the right hand side of Eq. (11)

is included in the simplified model. We observe that little can be said from this figure on whether the derivative or the right hand side of Eq. (11) is the dominant cause of the solution. As such, the simplified model doesn't yield the information that we would hope for.

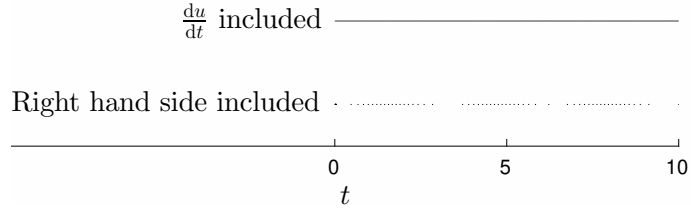


Figure 2: The terms included in the simplified model for the model given by Eqs. (11) and (12) where the simplified model is seeded by initially including only the time derivatives. A solid line represents the time interval where the indicated term is included in the simplified model.

We now seed the simplified model by including only the right hand side of Eq. (11) in the initial simplified model. The solution is shown in Figure 3, where we see that the accuracy of the simplified model is maintained. We then illustrate what terms are present in the simplified model in Figure 4. We see that this application of the algorithm gives a much clearer indication of what terms are driving the solution of the equation — both the derivative and the right hand side are required for the initial transient period, but only the right hand side is required for later times, in agreement with our expectations. We therefore see that our extension of the model reduction algorithm to allow different seedings of the initial simplified model can give more informative simplified models.

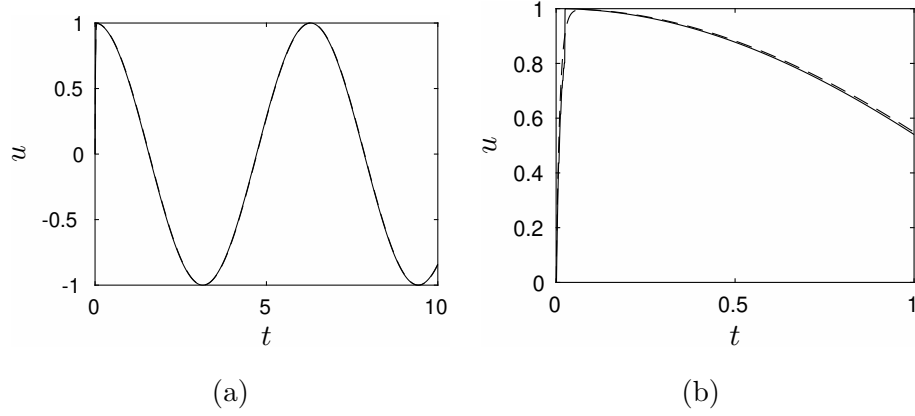


Figure 3: The simplified model corresponding to Eqs. (11)-(12) where the simplified model is seeded by initially using the right hand side of Eq. (11). The solution of the simplified model is represented by the solid line, and the full model is represented by the broken line. (a) shows the whole time interval $0 < t < 10$; (b) shows only earlier times $0 < t < 0.5$.

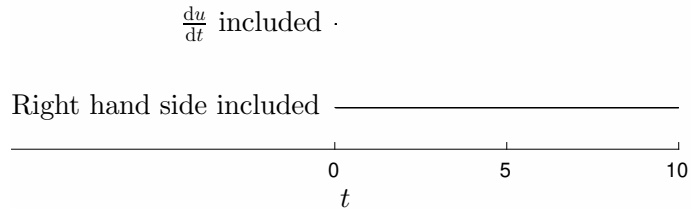


Figure 4: The terms included in the simplified model for the model given by Eqs. (11) and (12) where the simplified model is seeded by initially using the right hand side of Eq. (11). A solid line represents the time interval where the indicated term is included in the simplified model.

3.2. Enzyme kinetics

We now consider a dimensionless model of enzyme kinetics, based on Michaelis-Menten kinetics [10]. This model describes the combination of a substrate and enzyme to first form a complex, and to then form a product and the enzyme. The governing equations may be written, see for example

[11], as:

$$\frac{du}{dt} = -u + (u + K - \lambda)v, \quad (14)$$

$$\frac{dv}{dt} = \frac{u - (u + K)v}{\mu}, \quad (15)$$

where u is the dimensionless concentration of the substrate, v is the dimensionless concentration of the complex, t is dimensionless time, and K , λ , μ are constants satisfying $K > \lambda > 0$ and $\mu > 0$. Initial conditions are given by

$$u(0) = 1, \quad v(0) = 0. \quad (16)$$

We now investigate two distinguished limits of this system of equations. In both cases we first present an analytic simplification of the model, before illustrating that the model simplification algorithm gives identical predictions.

3.2.1. The distinguished limit $\mu \ll 1$

We first consider the case $\mu \ll 1$, and write $\epsilon = \mu$ to emphasise that μ is a small parameter. This case corresponds to the limit where the initial enzyme concentration is much smaller than the initial substrate concentration.

The time evolution of Eqs. (14)-(15) may be understood by application of multiple scales analysis. First we note that the system of equations is a singular perturbation problem due to the presence of the small parameter multiplying the derivative in Eq. (15); see, for example, [7] for more details on the approximation of singular perturbation problems. We therefore

rescale time so that $t = \epsilon\tau$ for $t = \mathcal{O}(\epsilon)$ and obtain

$$\frac{du_i}{d\tau} = \epsilon(-u_i + (u_i + K - \lambda)v_i), \quad (17)$$

$$\frac{dv_i}{d\tau} = u_i - (u_i + K)v_i, \quad (18)$$

$$u_i(0) = 1, \quad v_i(0) = 0, \quad (19)$$

where u_i, v_i are the solutions valid for $t = \mathcal{O}(\epsilon)$, with the subscript i denoting the inner solution. Eqs. (17)-(19) have solution

$$u_i(\tau) = 1, \quad v_i = \frac{1 - e^{-(1+K)\tau}}{1 + K}. \quad (20)$$

For longer times we let $\epsilon \rightarrow 0$ in Eqs. (14) and (15) and obtain, after a little manipulation,

$$\frac{du_o}{dt} = -u_o + (u_o + K - \lambda)v_o, \quad (21)$$

$$v_o = \frac{u_o}{u_o + K}, \quad (22)$$

where u_o, v_o are the outer solutions, valid for longer times. Matching the inner and outer solution then gives the initial condition for Eq. (21):

$$u_0(0) = 1. \quad (23)$$

In summary, a multiple scales analysis predicts that the derivative term in Eq. (15) may be neglected for sufficiently large times.

We now apply the model reduction algorithm described in Section 2.2 to Eqs. (14)-(16), using parameters $K = 1$, $\lambda = 0.5$, $\epsilon = 0.01$, the time interval $0 < t < 15$, and calculating numerical solutions using the backward Euler method with a step size of 0.001. We seed the simplified model with the right hand side of Eqs. (14)-(15). At every point in this mesh, we insist that the solution of the first equation of the simplified model is within an

absolute tolerance of 0.01 of the solution of the first equation of the full model.

In Figure 5 we show the solution of the simplified model (solid lines) and the full model (broken lines). Figure 5(a) shows the time interval $0 < t < 20$; Figure 5(b) zooms in to the region $0 < t < 1$. We see that the simplified model solution is accurate to the precision requested. In Figure 6 we show graphically the terms that are included in the simplified model. We see that the predictions are entirely consistent with the analytical predictions — both the time derivative and the right hand side of Eq. (15) are included for early times, after which only the right hand side of this equation is required.

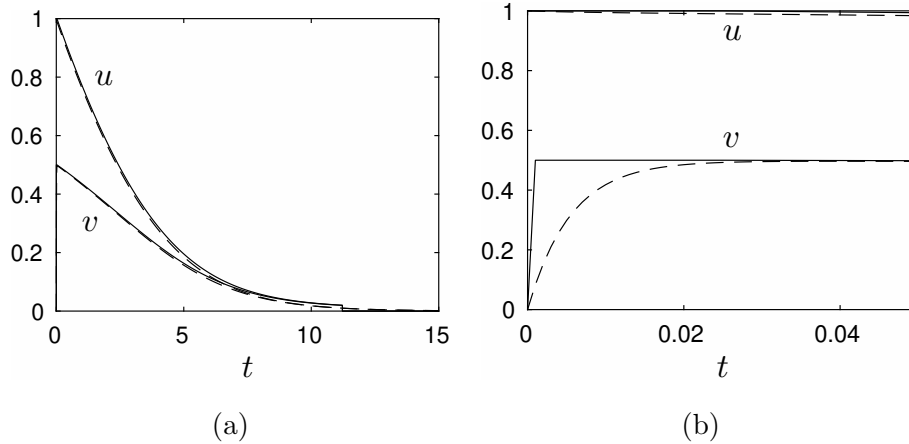


Figure 5: The simplified model for the enzyme kinetics described by Eqs. (14)-(16) in the limit $\mu \ll 1$ when the algorithm given in Section 2.2 is used. The solution of the simplified model is represented by solid lines, and the full model is represented by broken lines. (a) shows the whole time interval $0 < t < 20$; (b) shows earlier times $0 < t < 1$.

3.2.2. The distinguished limit $\lambda \ll 1$

We now consider the distinguished limit $\lambda \ll 1$, which corresponds to the dimensionless parameter $k_2/(k_1 s_0)$ being small, where k_2 is the catalytic

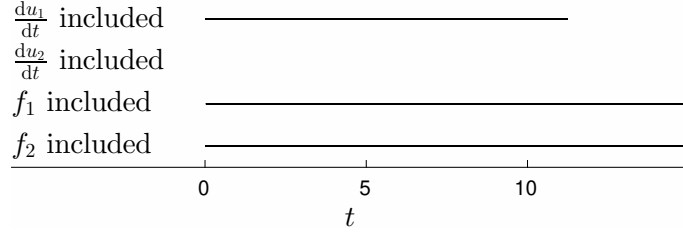


Figure 6: The terms included in the simplified model for the enzyme kinetics given by Eqs. (14)-(16) in the limit $\mu \ll 1$ when the algorithm given in Section 2.2 is used. A solid line represents the time interval where the indicated term is included in the simplified model. f_1 and f_2 represent the right hand side of Eqs. (14) and (15) respectively.

rate constant, k_1 is the forward rate constant and s_0 is the initial substrate concentration. Defining $\epsilon = \lambda$ to emphasise that λ is a small parameter, and defining a new variable w by $w = u + v$, we may write the governing system of equations, Eqs. (14)-(16), as

$$\frac{dw}{dt} = -\epsilon v, \quad (24)$$

$$\frac{dv}{dt} = \frac{w - v - (w - v - K)v}{\mu}, \quad (25)$$

$$w(0) = 1, \quad v(0) = 0. \quad (26)$$

It is evident from this system of equations that, to leading order, w is constant for $t = \mathcal{O}(1)$. The initial conditions and definition of w then allow us to deduce that, to leading order, $u + v = 1$ for $t = \mathcal{O}(1)$. A re-scaling time defined by $t = \epsilon^{-1}\tau$, together with the definition of w , allows us to deduce that for times $t = \mathcal{O}(\epsilon^{-1})$ the leading order contribution to Eqs. (25) is the quasi-static approximation

$$v = \frac{u}{u + K},$$

i.e., we may neglect the derivative term in Eq. (25) for times $t = \mathcal{O}(\epsilon^{-1})$.

We now apply our model simplification algorithm to investigate whether it is capable of automatically deducing the analytical predictions above. The numerical solution to Eqs. (14)-(16) is calculated using the backward Euler method with a stepsize of 0.1, and parameters $K = 1$, $\mu = 1$, $\lambda = 0.01$. The numerical solution to the full system of equations is plotted as broken lines in Figure 7, where plot (a) shows the solution over the time interval $0 < t < 1000$, i.e. including the times $t = \mathcal{O}(\epsilon^{-1})$, and plot (b) shows the solution for $t = \mathcal{O}(1)$. We apply the model simplification algorithm, demanding that the solution of the simplified model (denoted by solid lines) is such that the variable u calculated from the simplified model is always within a tolerance 0.01 of the true solution for u .

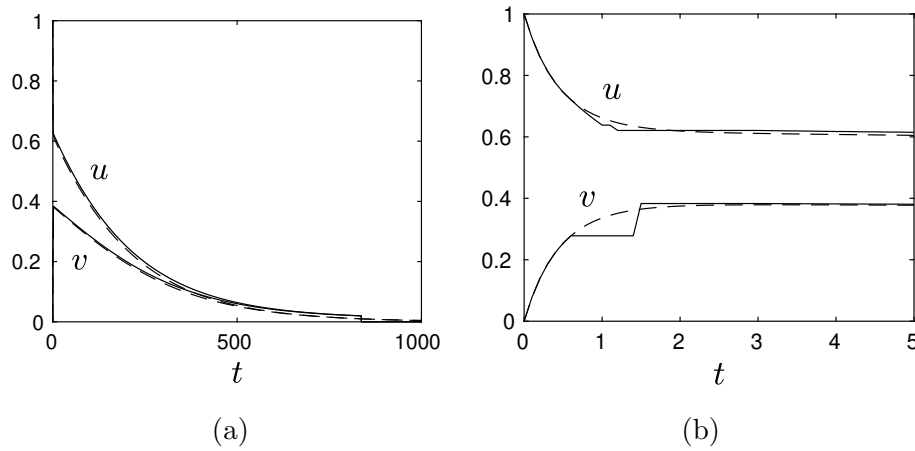


Figure 7: The simplified model for the enzyme kinetics described by Eqs. (14)-(16) in the limit $\lambda \ll 1$ when the algorithm given in Section 2.2 is used. The solution of the simplified model is represented by solid lines, and the full model is represented by broken lines. (a) shows the whole time interval $0 < t < 1000$; (b) shows earlier times $0 < t < 5$.

The terms that are included in the simplified model are plotted as a function of time in Figure 8. We see that the model simplification algorithm

successfully replicates our analytical observations; the time derivative dv/dt is included for $t = \mathcal{O}(1)$, but may be neglected completely for later times. Figure 8 allows us to illustrate a further feature of our model simplification technique. It is evident from this figure that, for $t \gg \mathcal{O}(1)$, we may neglect the term dv/dt , and also the term du/dt for large times when u is sufficiently close to the steady state. However, it is less clear when the term f_1 should be included, as it appears to oscillate rapidly between being included and not included for a large portion of the time interval shown. As there is no penalty to the accuracy of the simplified model in including this term, we can easily post-process the results of the model simplification algorithm to include terms where there is a rapid oscillation between this term being included and not included. It is then straightforward to interpret the model simplification presented in Figure 8 as predicting that both f_1 and f_2 should both be included for all times, that dv/dt need only be included for times $t = \mathcal{O}(1)$, and that the term du/dt need only be included for times until the solution is sufficiently close to the steady state. This model simplification is then entirely consistent with the analytic predictions made earlier.

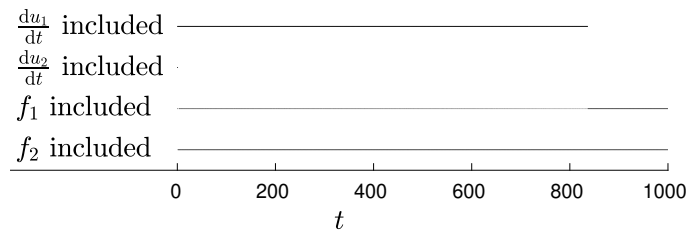


Figure 8: The terms included in the simplified model for the enzyme kinetics given by Eqs. (14)-(16) in the limit $\lambda \ll 1$ when the algorithm given in Section 2.2 is used. A solid line represents the time interval where the indicated term is included in the simplified model. f_1 and f_2 represent the right hand side of Eqs. (14) and (15) respectively.

3.3. Cardiac electrophysiology

Our final example is a single cell cardiac electrophysiology model. These models take a very similar form to the classical work of Hodgkin and Huxley for the squid giant axon [8], and are typically described by a system of initial value ordinary differential equations where the independent variable is time, and the dependent variables are the transmembrane potential, ionic concentrations and gating variables that control the flux of ions, and therefore the current, across the cell membrane. There are many models of different cardiac cell types; see [12] for a review. We use the model described by Beeler and Reuter [2]:

$$\frac{dV_m}{dt} = -\frac{i_{K_1} + i_{x_1} + i_{Na_1} + i_{Na_2} + i_{Ca} - i_e}{C_m}, \quad (27)$$

$$\frac{d[Ca^{2+}]_i}{dt} = -10^{-7}i_{Ca} + 0.07(10^{-7} - [Ca^{2+}]_i), \quad (28)$$

$$\frac{dG_a}{dt} = \alpha_a(V_m)(1 - G_a) - \beta_a(V_m)G_a, \quad G_a = x_1, m, h, j, d, f, \quad (29)$$

where: $C_m = 1$ is the membrane capacitance in units $\mu\text{F}/\text{cm}^2$; V_m is the transmembrane potential in mV; t is time measured in ms; i_{x_1} , i_{K_1} , i_{Na_1} , i_{Na_2} , i_{Ca} , i_e represent, respectively, the voltage and time dependent potassium current density, the time independent potassium current density, the gating variable dependent inward sodium current density, the gating variable independent inward sodium current density, the slow inward current density (comprising mainly calcium ions), and the external current density (with all current densities given in the units $\mu\text{A}/\text{cm}^2$); $[Ca^{2+}]_i$ is the intracellular concentration of calcium ions in units mole/litre; and G_a represents dimensionless gating variable a , where $a = x_1, m, h, j, d, f$. The current densities

are given by

$$\begin{aligned}
i_{k_1} &= \frac{1.4 (\exp (0.04 (V_m + 85)) - 1)}{\exp (0.08 (V_m + 53)) + \exp (0.04 (V_m + 53))} + \\
&\quad \frac{0.07 (V_m + 23)}{1 - \exp (-0.04 (V_m + 23))}, \\
i_{x_1} &= \frac{0.8x_1 (\exp (0.04 (V_m + 77)) - 1)}{\exp (0.04 (V_m + 35))}, \\
i_{Na_1} &= 4m^3hj (V_m - 50), \\
i_{Na_2} &= 0.003 (V_m - 50), \\
i_{Ca} &= 0.09df (V_m + 82.3 + 13.0287 \log [\text{Ca}^{2+}]_i),
\end{aligned} \tag{30}$$

and the functions that appear in the differential equations that determine the gating variables are given by

$$\begin{aligned}
\alpha_{x_1}(V_m) &= \frac{0.0005 \exp(0.083(V_m + 50))}{\exp(0.057(V_m + 50)) + 1}, \\
\beta_{x_1}(V_m) &= \frac{0.0013 \exp(-0.06(V_m + 20))}{\exp(-0.04(V_m + 20)) + 1}, \\
\alpha_m(V_m) &= -\frac{v + 47}{\exp(-0.1(V_m + 47)) - 1}, \\
\beta_m(V_m) &= 40 \exp(-0.056(V_m + 72)), \\
\alpha_h(V_m) &= 0.126 \exp(-0.25(V_m + 77)), \\
\beta_h(V_m) &= \frac{1.7}{\exp(-0.082(V_m + 22.5)) + 1}, \\
\alpha_j(V_m) &= \frac{0.055 \exp(-0.25(V_m + 78))}{\exp(-0.2(V_m + 78)) + 1}, \\
\beta_j(V_m) &= \frac{0.3}{\exp(-0.1(V_m + 32)) + 1}, \\
\alpha_d(V_m) &= \frac{0.095 \exp(-0.01(V_m - 5))}{\exp(-0.072(V_m - 5)) + 1}, \\
\beta_d(V_m) &= \frac{0.07 \exp(-0.017(V_m + 44))}{\exp(0.05(V_m + 44)) + 1}, \\
\alpha_f(V_m) &= \frac{0.012 \exp(-0.008(V_m + 28))}{\exp(0.15(V_m + 28)) + 1}, \\
\beta_f(V_m) &= \frac{0.0065 \exp(-0.02(V_m + 30))}{\exp(-0.2(V_m + 30)) + 1}.
\end{aligned}$$

Initial conditions are $V_m = 84$ mV, $[\text{Ca}^{2+}]_i = 10^{-7}$ mole/litre, and the gating variables take their steady state value. The stimulus current density i_e takes the value $11 \mu\text{A}/\text{cm}^2$ for $0 < t < 2.5$ ms, and zero at all other times.

To apply the model reduction algorithm we first write Eqs. (27)-(29) in the form of Eq. (2). There are 8 equations, and we name the dependent

variables as follows:

$$\begin{aligned} u_1 &= V_m, & u_2 &= [\text{Ca}^{2+}]_i, & u_3 &= x_1, & u_4 &= m, \\ u_5 &= h, & u_6 &= j, & u_7 &= d, & u_8 &= f. \end{aligned}$$

We also write

$$\begin{aligned} f_{11} &= -\frac{i_{K1}}{C_m}, & f_{12} &= -\frac{i_{x1}}{C_m}, & f_{13} &= -\frac{i_{Na1}}{C_m}, & f_{14} &= -\frac{i_{Na2}}{C_m}, \\ f_{15} &= -\frac{i_{Ca}}{C_m}, & f_{16} &= \frac{i_e}{C_m}, \\ f_{21} &= -10^{-7}i_{Ca}, & f_{22} &= 0.07(10^{-7} - u_2), \\ f_{n1} &= \alpha_n(u_1)(1 - u_n), & f_{n2} &= \beta_n(u_1)u_n, & n &= 3, 4, 5, 6, 7, 8. \end{aligned}$$

All numerical solutions of the system of differential equations are calculated using the backward Euler method with a timestep of 0.05 ms. The resulting nonlinear system of algebraic equations that arises on each timestep is solved using Newton's method.

3.3.1. Application of the model reduction algorithm

We first apply the model reduction algorithm given in Section 2.2, by insisting that the simplified model predicts a transmembrane potential that lies within 5 mV of the transmembrane predicted by the full system of equations given by Eqs. (27)-(29) for all times t . We plot the solutions given by the simplified model and full model in Figure 9, where we see in Figure 9(a) that the transmembrane potential is indeed predicted sufficiently accurately. We note from Figures 9(b), (c) and (d) that other dependent variables need not be predicted so accurately, especially for later times; the accuracy of these components is driven only by the requirement that the transmembrane potential is sufficiently accurate.

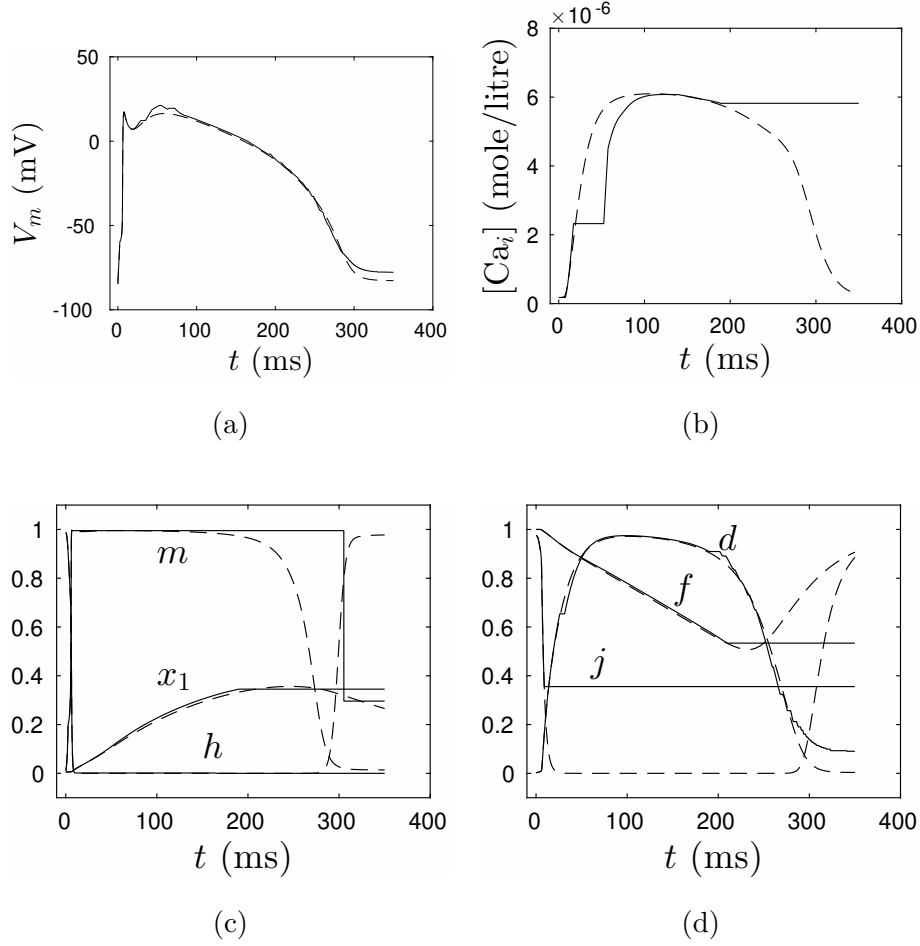


Figure 9: The simplified model for the Beeler-Reuter model of cardiac electrophysiology given by Eqs. (27)-(29) when the algorithm given in Section 2.2 is used with a tolerance of 5 mV for V_m . The solution of the simplified model is represented by solid lines, and the full model is represented by broken lines. (a) shows the transmembrane potential, (b) shows the intracellular calcium ion concentration, and (c) and (d) show the various gating variables contained in the model.

We illustrate the terms that are included in the simplified model in Figure 10. Most noticeable is that the right hand side of the differential equation corresponding to the equations representing the gating variables

m, h, j are only included for a very small period of time, mainly very early times. This is in agreement with known physiology; these variables control the sodium current given by Eq. (30), which causes the rapid increase in transmembrane potential for early times illustrated in Figure 9(a). Apart from that, it should be noted from Figure 10 that other right hand sides in the differential equation are more likely to be included for earlier times. This is to be expected, as we cannot hope for the solution to be correct at later times if it is not correct at earlier times. In particular, we note that there is a time around $t = 200$ ms, after which very few terms other than those on the right hand sides of the equations for the transmembrane potential V_m and the gating variable d are required. We see in Figure 9(a) that this is when the transmembrane potential is returning to its rest value and, for the model used, this requires only the gating variable d to vary with time. Observations such as this can be very useful in developing models; it is seen in [12] that many different models of action potentials have been proposed. The intracellular calcium ion concentrations predicted by these models can be very different. We see in Figure 10 that terms on the right hand side of the equation representing intracellular calcium ion concentration are included for earlier times, but not for later times. As a consequence, if a modeller is not satisfied with the accuracy of their model for the transmembrane potential they would be aware of the influence of the times where the evolution of intracellular calcium concentration affects their observations. Similar remarks apply to other components of the solution. We also remind the reader of the remarks made in Section 3.2.2, where if a term in an equation rapidly oscillates between being included and not included in the reduced model, then it may aid the interpretation of Figure 10 to assume that this term is included in the reduced model for the whole time period where it oscillates

between being included and not included.

We now apply the model reduction algorithm, demanding in the reduced model that, in addition to the transmembrane potential V_m being within 5 mV of its true value for all points in time as in the previous simulation, that the intracellular calcium ion concentration is within 3×10^{-7} mole/litre of its true value for all points in time. The reduced solutions (solid lines) and true solutions (broken lines) for each component of the solution are plotted in Figure 11. We see that the algorithm has successfully generated a reduced model that calculates V_m and $[C_a^{2+}]_i$ to within the specified tolerance. The components of the model that are included in the model are plotted in Figure 12. By comparing Figures 10 and 12 we see that the extra accuracy demanded of the intracellular calcium concentration has been achieved largely only at the expense of including extra terms in the equation for intracellular calcium ion concentration.

The applications of the model algorithm presented so far have all used tolerances that are homogeneous in time. We now apply the algorithm with a tolerance for V_m of 5 mV for $0 < t < 290$ ms, and a tolerance of 0.5 mV for $t \geq 290$ ms. This has the effect that the reduced model generated has a transmembrane potential that will return to its equilibrium value for larger times. The reduced solutions generated with this tolerance (solid lines) and true solutions (broken lines) for each component of the solution are plotted in Figure 13. We see that the algorithm has successfully generated a reduced model that calculates V_m to within the specified tolerance. However, to calculate V_m more accurately at later times we need to calculate this quantity more accurately at earlier times as well. As a consequence, the solution for V_m plotted in Figure 13(a) is more accurate than the solution plotted in Figure 9(a) for all times, even though the tolerance is the same

for $0 < t < 290$ ms. The components of the model that are included in the model are plotted in Figure 14. As would be expected, comparing Figures 10 and 14 we see that the extra accuracy required for V_m at all times results in many more components of the full model being required in the reduced model.

4. Discussion

In this study we developed the model reduction algorithm for systems of initial value ordinary differential equations described in [14]. There were two goals for extending this algorithm, the theory of which is described in Section 2. First, we demonstrated how the algorithm could be extended to systems of equations where the right hand side of each differential equation is decomposed into a finite sum, and we explained how the theory developed allows any one of the terms in this sum, or any time derivative, to be used in seeding the initial simplified model. This allowed the generation of simplified models that agreed with those that were derived using analytic techniques. Second, we extended the algorithm so that the simplified model could predict one or more components of the solution at all times. The theory was then successfully applied to two established models in Section 3, illustrating the practical applicability of the theory.

Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council through the Life Sciences Interface Doctoral Training Centre at the University of Oxford, grant number EP/F500394/1.

In compliance with the UK Engineering and Physical Sciences Research Council’s open access initiative, the data in this paper is available from <https://doi.org/10.5287/bodleian:aVBA14JKG> .

References

- [1] M. Ainsworth and J.T. Oden. *A posteriori* error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142:1–88, 1997.
- [2] G.W. Beeler and H. Reuter. Reconstruction of the action potential of ventricular myocardial fibres. *Journal of Physiology*, 268:177–210, 1977.
- [3] R. Clewley, H.G. Rotstein, and N. Kopell. A computational tool for the reduction of nonlinear ODE systems possessing multiple scales. *Multi-scale Modeling and Simulation*, 4:732–759, 2005.
- [4] A.C. Fowler. *Mathematical Models in the Applied Sciences*. Cambridge University Press, Cambridge, UK, 1997.
- [5] M.B. Giles and E. Süli. Adjoint methods for PDEs: *a posteriori* error analysis and postprocessing by duality. *Acta Numerica*, 11:145–236, 2002.
- [6] K. Harriman, D.J. Gavaghan, P. Houston, and E. Süli. Adaptive finite element simulation of currents at microelectrodes to a guaranteed accuracy. First-order EC’ mechanism at inlaid and recessed discs. *Electrochemical Communications*, 2:163–170, 2000.
- [7] E.J. Hinch. *Perturbation Methods*. Cambridge University Press, Cambridge, UK, 1991.

- [8] A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544, 1952.
- [9] P.J. Maybank and J.P. Whiteley. Automatic simplification of systems of reaction-diffusion equations by *a posteriori* analysis. *Mathematical Biosciences*, 248:146–157, 2014.
- [10] L. Michaelis and M.I. Menten. Die kinetik der invertinwirkung. *Biochemische Zeitschrift*, 49:333–369, 1913.
- [11] J.D. Murray. *Mathematical Biology*. Springer, Berlin, Heidelberg, New York, 1989.
- [12] D. Noble, A. Garny, and P.J. Noble. How the Hodgkin-Huxley equations inspired the Cardiac Physiome Project. *Journal of Physiology*, 590:2613–2628, 2012.
- [13] E. Süli and D. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, Cambridge, UK, 2006.
- [14] J.P. Whiteley. Model reduction using *a posteriori* analysis. *Mathematical Biosciences*, 225:44–52, 2010.
- [15] D. Wirtz, D.C. Sorensen, and B. Haasdonk. A posteriori error estimation for DEIM reduced nonlinear dynamical systems. *SIAM Journal on Scientific Computing*, 36:A311–A338, 2014.

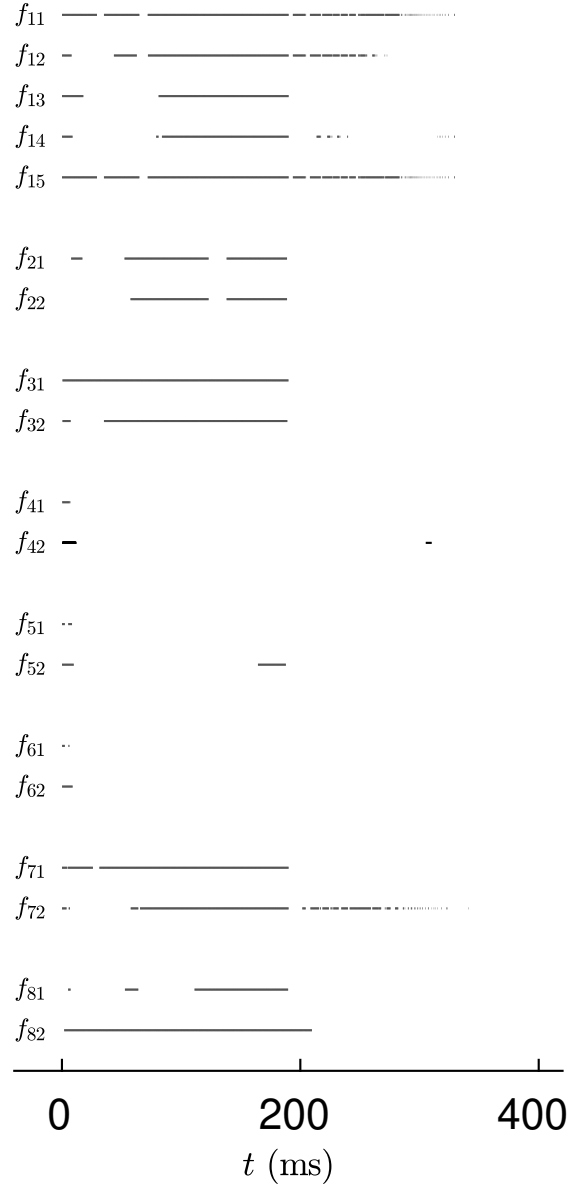


Figure 10: The terms included in the simplified model for the Beeler-Reuter model of cardiac electrophysiology given by Eqs. (27)-(29) when the algorithm given in Section 2.2 is used with a tolerance of 5 mV for V_m . A solid line represents the time interval where the indicated term is included in the simplified model.

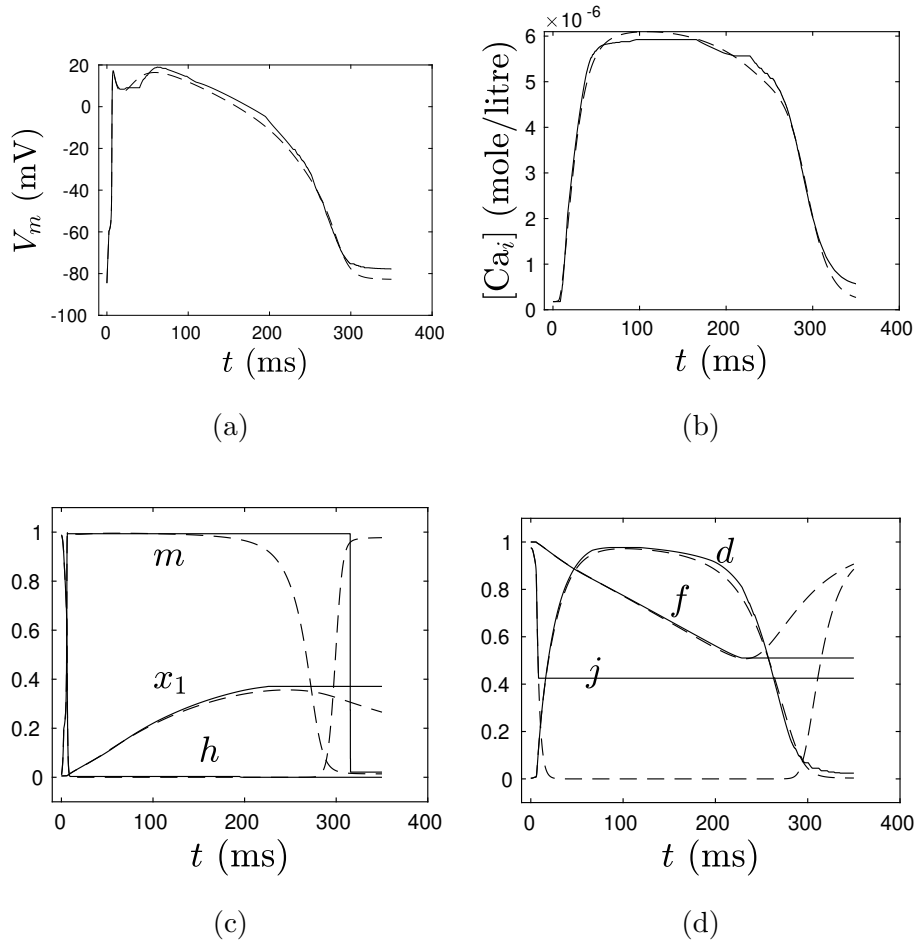


Figure 11: The simplified model for the Beeler-Reuter model of cardiac electrophysiology given by Eqs. (27)-(29) when the algorithm given in Section 2.2 is used with a tolerance of 5 mV for V_m , and a tolerance of 3×10^{-7} mole/litre for $[Ca^{2+}]_i$. The solution of the simplified model is represented by solid lines, and the full model is represented by broken lines. (a) shows the transmembrane potential, (b) shows the intracellular calcium concentration, and (c) and (d) show the various gating variables contained in the model.

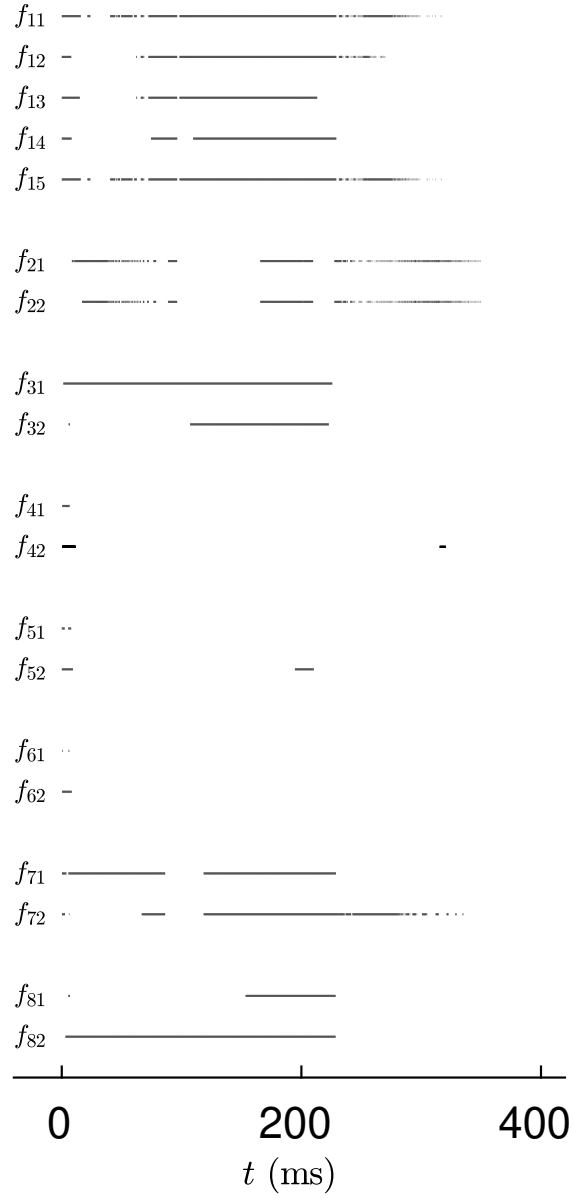


Figure 12: The terms included in the simplified model for the Beeler-Reuter model of cardiac electrophysiology given by Eqs. (27)-(29) when the algorithm given in Section 2.2 is used with a tolerance of 5 mV for V_m , and a tolerance of 3×10^{-7} mole/litre for $[C_a^{2+}]_i$. A solid line represents the time interval where the indicated term is included in the simplified model.

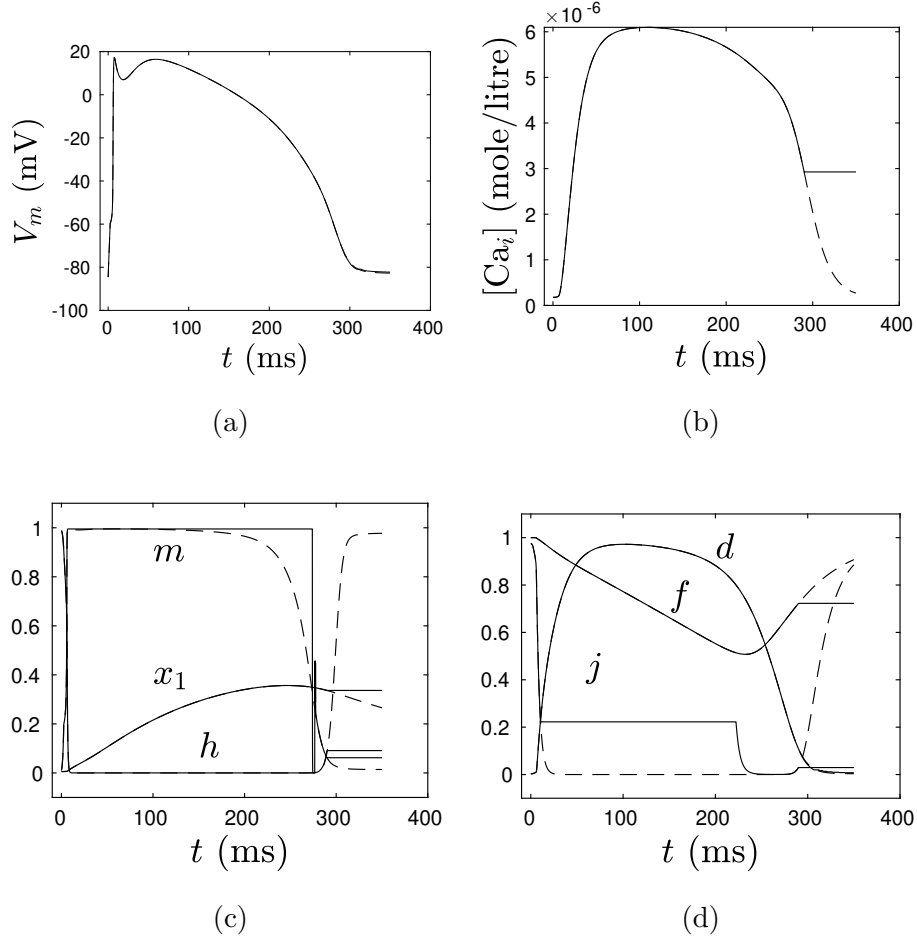


Figure 13: The simplified model for the Beeler-Reuter model of cardiac electrophysiology given by Eqs. (27)-(29) when the algorithm given in Section 2.2 is used with a tolerance for V_m which is heterogeneous in time. The solution of the simplified model is represented by solid lines, and the full model is represented by broken lines. (a) shows the transmembrane potential, (b) shows the intracellular calcium concentration, and (c) and (d) show the various gating variables contained in the model.

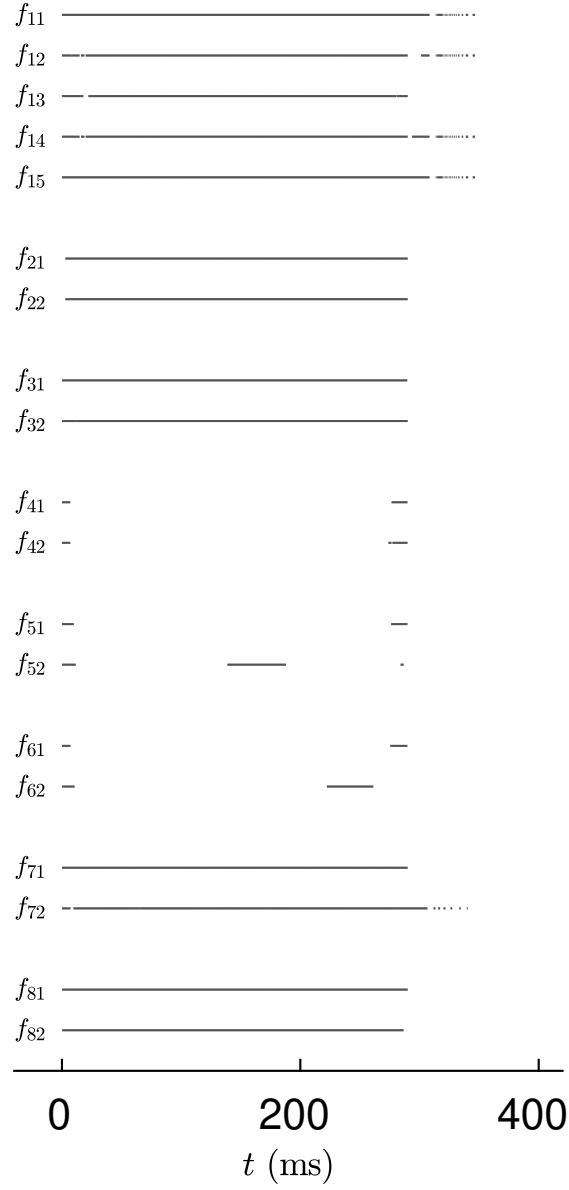


Figure 14: The terms included in the simplified model for the Beeler-Reuter model of cardiac electrophysiology given by Eqs. (27)-(29) when the algorithm given in Section 2.2 is used with a tolerance for V_m which is heterogeneous in time. A solid line represents the time interval where the indicated term is included in the simplified model.