Mathematical Biosciences 248 (2014) 146–157

Contents lists available at ScienceDirect

Mathematical Biosciences

journal homepage: www.elsevier.com/locate/mbs

Automatic simplification of systems of reaction–diffusion equations by *a posteriori* analysis

Philip J. Maybank, Jonathan P. Whiteley *

Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom

ARTICLE INFO

Article history:
Received 19 June 2013
Received in revised form 20 December 2013
Accepted 31 December 2013
Available online 11 January 2014

Keywords:
Model reduction
Reaction–diffusion
A posteriori

ABSTRACT

Many mathematical models in biology and physiology are represented by systems of nonlinear differential equations. In recent years these models have become increasingly complex in order to explain the enormous volume of data now available. A key role of modellers is to determine which components of the model have the greatest effect on a given observed behaviour. An approach for automatically fulfilling this role, based on *a posteriori* analysis, has recently been developed for nonlinear initial value ordinary differential equations [J.P. Whiteley, Model reduction using *a posteriori* analysis, Math. Biosci. 225 (2010) 44–52]. In this paper we extend this model reduction technique for application to both steady-state and time-dependent nonlinear reaction–diffusion systems. Exemplar problems drawn from biology are used to demonstrate the applicability of the technique.

1. Introduction

Simplification of mathematical models to identify the underlying mechanisms responsible for an observed behaviour is an integral component of mathematical modelling. This simplification often leads to reduced models that demonstrate key features with greater clarity. Using an example from the field of cell–level electrophysiology, the FitzHugh–Nagumo model was developed independently by both Fitzhugh [1] and Nagumo et al. [2] as a result of a perturbation analysis of the Hodgkin–Huxley equations [3] and allows a clear exposition of the property of excitability.

In recent years mathematical models in biology have become increasingly complex. Such models are often described by systems of differential equations that contain tens or hundreds of variables and parameters, several different types of differential operator [4–7], and describe processes that act across a wide range of spatial and temporal scales [8]. Perturbation methods, also known as asymptotic methods, are one family of techniques that may be used to simplify these models and identify the dominant terms in these equations that cause a given output. Applying these methods to a system of differential equations is, however, time–consuming and requires substantial mathematical expertise. Furthermore the difficulty of applying perturbation methods increases significantly with the number of equations and parameters in the system. The application of perturbation methods to mathematical models in biology has generally focused on systems of initial value ordinary differential equations where the system of differential equations takes a specific form or models a specific phenomena. One example of this approach is the work by Segel and Slemrod [9], who considered systems of equations that model chemical reactions. The different rates at which the reactions occur were utilised to generate a quasi–steady state assumption for some variables. Other examples of perturbation analysis being used to elucidate the important detail from systems biology models include approximations of a cardiac electrophysiology model [10], a comparison of the excitability properties exhibited by models of cardiac and nerve electrophysiology [11], a Calcium-Induced Calcium Release (CICR) model [12], and an NF-κB signalling model [13]. In a similar spirit, Maas and Pope [14] use a dynamical systems approach to automatically simplify systems of initial value ordinary differential equations that model chemical kinetics, Voit and Ferreira [15] propose simple models of chemical buffers that are demonstrated to stabilise the behaviour of other system variables, and Cha et al. [16] and Kepler et al. [17] have simplified models of excitability by replacing the differential equations modelling gating variables by algebraic equations.

One alternative, commonly used tool for the analysis of complex mathematical models is sensitivity analysis. Sensitivity analysis studies will often set out to answer the question, "Which parameter values in the model equations have the greatest effect on a model output of interest?". Answering this question is important for deciding which parameters in a model need to be accurately estimated, and also for predicting how system behaviour will change in response to different interventions. A closely related question that model analysis will often set out to answer is, "Which physical processes in the model equations have the

* Corresponding author. Tel.: +44 1865 273838.
E-mail addresses: philip.maybank@gmail.com (P.J. Maybank), jonathan.whiteley@cs.ox.ac.uk (J.P. Whiteley).

0025-5564/$ - see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.mbs.2013.12.011

greatest effect on a model output of interest?". Whereas sensitivity analysis methods [18–29] are more suited to answering the first question, an alternative method developed in this paper aims to answer the second one.

Model reduction using *a posteriori* analysis, the subject of this study, is an automated method for identifying key components of a mathematical model that affect a user–defined quantity of interest. This method borrows ideas from *a posteriori* error analysis of a finite element solution of a system of differential equations. Whereas *a posteriori* error analysis for finite element methods is used to construct a suitably accurate numerical discretisation of the differential operator by indicating regions of the computational domain where the finite element mesh needs to be made finer [30–33], model reduction using *a posteriori* analysis is used to construct a reduced system of differential equations that is sufficiently accurate to approximate the full system of equations. *A posteriori* model reduction has been applied to coupled nonlinear systems of initial value ordinary differential equations [34] to identify the key terms in the equations that are responsible for a specified output of the system. This is similar to the approach described by Clewley et al. [35] for initial value ordinary differential equations where all terms appearing in the equations are ranked in terms of their magnitude during each partition of time. Terms whose magnitude is smaller than some prescribed fraction of the term with largest magnitude are neglected in the reduced model. This approach has several advantages: for example it is intuitive and easy to implement. It does, however, neglect to take account of how a term that is small in some time interval may subsequently cause a significant effect on the solution at later times. This algorithm has been applied successfully to several problems exhibiting bursting and spiking behaviour [36–38] and has been shown to be compatible with the PyDSTool software environment [39].

The aim of this paper is to develop the *a posteriori* model reduction technique so that it can be applied to a wider range of differential equations than in previous work [34]. The idea underpinning this technique is fairly simple: this idea is presented for the case of a general differential equation in Section 2. In Section 3 we extend the methodology so that it may be applied to models described by boundary–value second order ordinary differential equations. In Section 4 we extend the methodology further so that it may be applied to systems of parabolic partial differential equations in one spatial dimension such as time-dependent reaction–diffusion equations.

2. Overview of the method

We first give an overview of the method in sketch form: a more rigorous and technical description will be given later. Given an output from the model, in the form of a linear functional of the solution, the aim of the method is to distinguish between regions of the computational domain where the full system of equations must be solved, and other regions where an approximate, simpler system is adequate. By "approximate, simpler system" we mean a simpler differential equation that approximates the original differential equation, and not a numerical approximation to the differential equation. The purpose of making such an approximation is not to create a more accurate or more computationally efficient scheme for obtaining a numerical solution. Rather, it is to identify the regions of the computational domain, and the components of the model, that have the greatest impact on some user–defined quantity of interest. We should emphasise that the reduced model generated is for a given user–defined output. Different outputs will be dependent on different physical mechanisms, and may therefore be described by a different reduced model.

Suppose the full model of the system is defined as,

$$\mathcal{D}u = p, \tag{1}$$

where $\mathcal{D}$ is a differential operator, including all necessary initial and boundary conditions, $u(x)$ is the solution to the differential equation, and $p(x)$ is independent of u. We define the approximate system by

$$\widehat{\mathcal{D}}U = p,$$

where $\widehat{\mathcal{D}}$ is an approximation to the differential operator $\mathcal{D}$, and U is the solution to this reduced differential equation. Suppose we have a user–defined quantity of interest that may be written as a linear functional of the solution u, i.e.,

$$J(u) = \langle u, g\rangle,$$

where $\langle u, g\rangle$ denotes an inner product between u and g. The key concept that underpins this model reduction technique is the adjoint differential equation to the original differential equation. An appropriate adjoint problem is: find ϕ such that

$$\mathcal{D}^*\phi = g. \tag{2}$$

This definition of the adjoint problem allows us to write

$$\begin{aligned} J(u) - J(U) &= \langle u - U, g\rangle \\ &= \langle u - U, \mathcal{D}^*\phi\rangle && (3)\\ &= \langle \mathcal{D}u - \mathcal{D}U, \phi\rangle \quad \text{for an appropriately chosen } \mathcal{D}^* && (4)\\ &= \langle p - \mathcal{D}U, \phi\rangle \\ &= \langle R(U), \phi\rangle, && (5)\end{aligned}$$

where $R(U)$ is defined by $p - \mathcal{D}U$, and is the residual obtained when the solution to the reduced model U is substituted into the full model given by Eq. (1). Details on how to construct the adjoint problem are given later in this paper.

Our *a posteriori* model reduction technique is based on the following observation. Clearly if U (the solution of the approximate system of differential equations) is a good approximation to u (the solution of the full system of differential equations) for the purposes of calculating the linear functional J, then $|J(u) - J(U)|$ will be small. Loosely speaking, this will be true if, for each component of the solution at every point of the computational domain, either: (i) the magnitude of the residual is small; or (ii) the magnitude of ϕ is small. As a consequence, if the magnitude of ϕ is small we do not need an accurate solution to the differential equation to calculate $J(u)$ accurately: an approximate model will suffice.

The contribution to $J(u) - J(U)$ given by Eq. (5) may be decomposed into local contributions from different regions of the computational domain, allowing us to identify regions with the highest contribution to the error in calculating the functional using the reduced model. The full model is then solved in these regions thus generating an updated reduced model. This process is repeated iteratively until sufficient accuracy is achieved.

Having illustrated the key concept behind our *a posteriori* model reduction technique, we shall now proceed to develop an algorithm for systematically using this technique for various classes of differential equations.

3. Boundary-value differential equations

In this section we consider systems of boundary value ordinary differential equations. We begin by deriving an *a posteriori* model reduction algorithm for general systems of nonlinear boundary value ordinary differential equations in Section 3.1. This algorithm is then applied first to an exemplar linear model problem in Section 3.2, and then to a nonlinear chemotaxis problem in Section 3.3.