

Enclosing All Solutions of Two-Point Boundary Value Problems for ODEs

Youdong Lin, Joshua A. Enszer, and Mark A. Stadtherr¹

Department of Chemical and Biomolecular Engineering
University of Notre Dame, Notre Dame, IN 46556, USA

July 10, 2007
(revised, August 19, 2007)

¹Author to whom all correspondence should be addressed. Phone: (574) 631-9318; Fax: (574) 631-8366;
E-mail: markst@nd.edu

Abstract

The two-point boundary value problem (TPBVP) occurs in a wide variety of problems in engineering and science, including the modeling of chemical reactions, heat transfer, and diffusion, and the solution of optimal control problems. A TPBVP may have no solution, a single solution, or multiple solutions. A new strategy is presented for reliably locating all solutions of a TPBVP. The method determines narrow enclosures of all solutions that occur within a specified search interval. Key features of the method are the use of a new solver for parametric ODEs, which is used to produce guaranteed bounds on the solutions of nonlinear dynamic systems with interval-valued parameters and initial states, and the use of a constraint propagation strategy on the Taylor models used to represent the solutions of the dynamic system. Numerical experiments demonstrate the use and computational efficiency of the method.

Keywords: Two-point boundary value problem; Ordinary differential equations; Initial value problem; Interval analysis; Taylor model

1 Introduction

Systems of ordinary differential equations (ODEs) arise in mathematical models throughout science and engineering. When an explicit condition (or conditions) that a solution must satisfy is specified at one value of the independent variable, usually its lower bound, this is referred to as an initial value problem (IVP). When the conditions to be satisfied occur at more than one value of the independent variable, this is referred to as a boundary value problem (BVP). If there are two values of the independent variable at which conditions are specified, then this is a two-point boundary value problem (TPBVP). TPBVPs occur in a wide variety of problems, including the modeling of chemical reactions, heat transfer, and diffusion. They are also of interest in optimal control problems.

For a typical TPBVP, when expressed as system of first-order ODEs, there are one or more unknown initial states. Solving the TPBVP is equivalent to finding these initial states. In other cases, the initial states are known, but there are unknown model parameters, and solving the TPBVP requires finding values for these parameters. There may also be a combination of unknown initial states and parameters that must be determined to solve the TPBVP. A TPBVP may have no solution, a single solution, or multiple solutions. There are many techniques available for the numerical solution of TPBVPs for ODEs (Heath, 2002). The standard techniques can be divided into two classes. The first class is based on solving a related IVP in order to obtain the TPBVP solutions. Typical of this class are various shooting and multi-shooting approaches. The other class involves converting the TPBVP into a system of algebraic equations, and includes methods based on various versions of finite difference or collocation. Methods for solving TPBVPs usually require users to provide an initial guess for the unknown initial states and/or parameters. A common limitation to all of these approaches is that they can, at best, achieve convergence to a local solution of the TPBVP, which means other solutions of interest may be missed.

The problem of reliably identifying *all* solutions of a TPBVP was apparently first addressed only recently, by Chen & Manousiouthakis (2005) and Chen (2006). They propose a type of shooting method based on interval-valued initial states, which encloses solutions to the TPBVP within very narrow intervals. To enclose the state variables in the associated interval IVPs, they propose using a fourth-order Runge-Kutta method implemented using interval operations. They discuss the need to bound local truncation error in order to guarantee the enclosures, and formulate an approach for doing so. However, in the example given, it is not clear that this approach for bounding truncation error is actually implemented.

In this paper, we will present a new approach that will *rigorously guarantee* the enclosure of all solutions to the TPBVP. Like the method of Chen & Manousiouthakis (2005) and Chen (2006), this method is based on interval analysis and involves a type of shooting approach. However, the underlying interval methods that we use are much different than those employed by Chen & Manousiouthakis (2005) and Chen (2006). A key feature is the use of a new validated solver (Lin & Stadtherr, 2007) for parametric ODEs, which is used to produce guaranteed bounds on the solutions of IVPs with both interval-valued initial conditions and parameters. This is combined with a constraint propagation scheme to yield an approach that is both rigorous and efficient.

The remainder of this paper is organized as follows. In Section 2, we provide a mathematical formulation of the TPBVP. In Section 3, we give a brief introduction to interval analysis and Taylor models. In Section 4, we review the new validated method for parametric ODEs, which makes use of interval analysis and Taylor models. Then, in Section 5, we describe a constraint propagation procedure based on Taylor models, and outline algorithmically the proposed method for enclosing all solutions of TPBVPs. Finally, in Section 6 we present the results of several numerical experiments that demonstrate the proposed approach.

2 Problem definition

This section provides the mathematical formulation of the two-point boundary value problem (TPBVP) for ordinary differential equations (ODEs). The general ODE problem can be written as an autonomous first-order system

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) \quad t \in [t_0, t_f], \quad (1)$$

where $\mathbf{x}(t)$ is the vector (length n) of unknowns (state variables), $\boldsymbol{\theta}$ is a vector (length p) of unknown parameters, and the independent variable bounds t_0 and t_f are finite. Here, only the autonomous ODE system is considered, since a non-autonomous system can be easily transformed into an autonomous one by introducing an additional state variable with derivative equal to one. A system of first-order ODEs like Eq. (1) is normally supplemented by $m = n + p$ boundary conditions, which for a TPBVP involving the two points $t = t_0$ and $t = t_f$ can be written

$$\mathbf{g}(\mathbf{x}(t_0), \mathbf{x}(t_f), \boldsymbol{\theta}) = \mathbf{0}. \quad (2)$$

Since $\mathbf{x}(t_f) = \mathbf{x}_f$ depends on $\mathbf{x}(t_0) = \mathbf{x}_0$ and $\boldsymbol{\theta}$, this represents a system of m equations in m unknowns, the n components of \mathbf{x}_0 and p components of $\boldsymbol{\theta}$. We assume that \mathbf{f} is $(k - 1)$ -times continuously differentiable with respect to the independent variable t , and $(q + 1)$ -times continuously differentiable with respect to the parameters $\boldsymbol{\theta}$ and the initial values \mathbf{x}_0 . Here k is the order of the truncation error in the interval Taylor series (ITS) method to be used in the integration procedure (to be discussed in Section 4), and q is the order of the Taylor model to be used to represent dependence on parameters and initial values (to be discussed in Section 3.2). We also assume that \mathbf{g} is $(q + 1)$ -times continuously differentiable with respect to $\boldsymbol{\theta}$ and \mathbf{x}_0 .

When a typical shooting method is used, values of the unknown parameters $\boldsymbol{\theta}$ and initial values \mathbf{x}_0 are assumed, and an IVP solver is applied to the ODE system to obtain \mathbf{x}_f . The assumed values of $\boldsymbol{\theta}$ and \mathbf{x}_0 are then iterated on until Eq. (2) is satisfied. As noted above, standard

numerical methods for solving Eq. (2) can, at best, achieve convergence only to a local solution of the TPBVP, which means that other solutions of interest may be missed. In this paper, we will describe an approach for rigorously enclosing all solutions of Eq. (2) within very narrow intervals. It is assumed that the solutions of interest lie in specified search intervals \mathbf{X}_0 and Θ ; that is, $\mathbf{x}_0 \in \mathbf{X}_0$ and $\theta \in \Theta$. These search intervals may be arbitrarily large, and in many practical cases are determined on the basis of physical bounds.

3 Background

The approach described here for solving the TPBVP is based on interval analysis and employs Taylor models. Thus, as background, we provide a brief summary of interval analysis and of Taylor models. Much more detail on both topics is available elsewhere.

3.1 Interval analysis

A real interval X is defined by $X = [\underline{X}, \overline{X}] = \{x \in \mathbb{R} \mid \underline{X} \leq x \leq \overline{X}\}$. Here an underline is used to indicate the lower bound of an interval and an overline is used to indicate the upper bound. A real interval vector $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ has n real interval components and can be interpreted geometrically as an n -dimensional rectangle or box. Note that in this context uppercase quantities are intervals, and lowercase quantities or uppercase quantities with underline or overline are real numbers. Basic arithmetic operations with intervals are defined by $X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}$, where $\text{op} \in \{+, -, \times, \div\}$. Interval versions of the elementary functions can be similarly defined. It should be emphasized that, when machine computations with interval arithmetic operations are done, the endpoints of an interval are computed with a directed (outward) rounding scheme. That is, the lower endpoint is rounded down to the next machine-representable number and the upper endpoint is rounded up to the next machine-representable number. In this

way, through the use of interval, as opposed to floating-point, arithmetic, any potential rounding error problems are avoided. Several good introductions to interval analysis, as well as interval arithmetic and other aspects of computing with intervals, are available (Hansen & Walster, 2004; Jaulin et al., 2001; Kearfott, 1996; Neumaier, 1990). Implementations of interval arithmetic and elementary functions are also readily available, and recent compilers from Sun Microsystems directly support interval arithmetic and an interval data type.

In working with intervals, it is important to understand the computation of interval function extensions, and the associated “dependency” problem. For an arbitrary function $f(\mathbf{x})$, the interval extension $F(\mathbf{X})$ encloses all possible values of $f(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}$. That is, $F(\mathbf{X}) \supseteq \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$ encloses the range of $f(\mathbf{x})$ over \mathbf{X} . It is often computed by substituting the given interval \mathbf{X} into the function $f(\mathbf{x})$ and then evaluating the function using interval arithmetic. This so-called “natural” interval extension is sometimes wider than the actual range of function values, though it always includes the actual range. For example, the natural interval extension of $f(x) = x/(x - 2)$ over the interval $X = [3, 4]$ is $F([3, 4]) = [3, 4]/([3, 4] - 2) = [3, 4]/[1, 2] = [1.5, 4]$, while the true function range over this interval is $[2, 3]$. This overestimation of the function range is due to the “dependency” problem, which may arise when a variable occurs more than once in a function expression. While a variable may take on any value within its interval, it must take on the *same* value each time it occurs in an expression. However, this type of dependency is not recognized when the natural interval extension is computed. In effect, when the natural interval extension is used, the range computed for the function is the range that would occur if each instance of a particular variable was allowed to take on a different value in its interval range. For the case in which $f(\mathbf{x})$ is a single-use expression, that is, an expression in which each variable occurs only once, the natural interval arithmetic will always yield the true function range. For example, rearrangement of the function expression used above gives $f(x) = x/(x - 2) = 1 + 2/(x - 2)$, and now $F([3, 4]) =$

$1 + 2/([3, 4] - 2) = 1 + 2/[1, 2] = 1 + [1, 2] = [2, 3]$, the true range.

In some situations, including one encountered in Section 5.1, dependency issues can be avoided through the use of the dependent subtraction operation (also known as the cancellation operation). Assume that there is an interval S that depends additively on the interval A . The dependent subtraction operation is defined by $S \ominus A = [\underline{S} - \underline{A}, \overline{S} - \overline{A}]$. For example, let $A = [0, 1]$, $B = [1, 2]$, $C = [2, 3]$ and $S = A + B + C = [3, 6]$. Say that only S is stored and that later it is desired to compute $A + B$ by subtracting C from S . Using the standard subtraction operation yields $S - C = [3, 6] - [2, 3] = [0, 4]$, which overestimates the true $A + B$. Using the dependent subtraction operation, which is allowable since S depends additively on C , yields $S \ominus C = [3, 6] \ominus [2, 3] = [1, 3]$, which is the true $A + B$. For more general situations, there are a variety of other approaches that can be used to try to tighten interval extensions (Hansen & Walster, 2004; Jaulin et al., 2001; Kearfott, 1996; Neumaier, 1990), including the use of Taylor models, as described in the next subsection.

3.2 Taylor models

Makino & Berz (1996) have described a remainder differential algebra (RDA) approach for bounding function ranges and control of the dependency problem of interval arithmetic (Makino & Berz, 1999). In this method, a function is represented using a model consisting of a Taylor polynomial and an interval remainder bound. Such a model is called a Taylor model.

One way of forming a Taylor model of a function is by using the Taylor theorem. Consider a function $f : \mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n \rightarrow \mathbb{R}$ that is $(q + 1)$ times partially differentiable on \mathbf{X} and let $\mathbf{x}_0 \in \mathbf{X}$. The Taylor theorem states that for each $\mathbf{x} \in \mathbf{X}$, there exists a $\zeta \in \mathbb{R}$ with $0 < \zeta < 1$ such that

$$f(\mathbf{x}) = p_f(\mathbf{x} - \mathbf{x}_0) + r_f(\mathbf{x} - \mathbf{x}_0, \zeta), \quad (3)$$

where p_f is a q -th order polynomial (truncated Taylor series) in $(\mathbf{x} - \mathbf{x}_0)$ and r_f is a remainder, which can be quantitatively bounded over $0 < \zeta < 1$ and $\mathbf{x} \in \mathbf{X}$ using interval arithmetic or other

methods to obtain an interval remainder bound R_f . A q -th order Taylor model $T_f = p_f + R_f$ for $f(\mathbf{x})$ over \mathbf{X} then consists of the polynomial p_f and the interval remainder bound R_f and is denoted by $T_f = (p_f, R_f)$. Note that $f \in T_f$ for $\mathbf{x} \in \mathbf{X}$ and thus T_f encloses the range of f over \mathbf{X} .

In practice, it is more useful to compute Taylor models of functions by performing Taylor model operations. Arithmetic operations with Taylor models can be done using the RDA operations described by Makino & Berz (1996, 1999, 2003), which include addition, multiplication, reciprocal, and intrinsic functions. Using these, it is possible to start with simple functions such as the constant function $f(x) = k$, for which $T_f = (k, [0, 0])$, and the identity function $f(x_i) = x_i$, for which $T_f = (x_{i0} + (x_i - x_{i0}), [0, 0])$, and then to compute Taylor models for very complicated functions. Therefore, it is possible to compute a Taylor model for any function representable in a computer environment by simple operator overloading through RDA operations. It has been shown that, compared to other rigorous bounding methods, the Taylor model often yields sharper bounds for modest to complicated functional dependencies (Makino & Berz, 1996, 1999; Neumaier, 2003). A discussion of the uses and limitations of Taylor models has been given by Neumaier (2003).

An interval bound on a Taylor model $T = (p, R)$ over \mathbf{X} is denoted by $B(T)$, and is found by determining an interval bound $B(p)$ on the polynomial part p and then adding the remainder bound; that is, $B(T) = B(p) + R$. The range bounding of the polynomial $B(p) = P(\mathbf{X} - \mathbf{x}_0)$ is an important issue, which directly affects the performance of Taylor model methods. Unfortunately, exact range bounding of an interval polynomial is NP hard, and direct evaluation using interval arithmetic is very inefficient, often yielding only loose bounds. Thus, various bounding schemes (Makino & Berz, 2004, 2005; Neumaier, 2003) have been used, mostly focused on exact bounding of the dominant parts of P , i.e., the first- and second-order terms. However, exact bounding of a general interval quadratic is also computationally expensive (in the worst case, exponential in the

number of variables). Thus, we have adopted a very simple compromise approach, in which only the first-order and the diagonal second-order terms are considered for exact bounding, and other terms are evaluated directly. That is,

$$B(p) = \sum_{i=1}^n \left[a_i (X_i - x_{i0})^2 + b_i (X_i - x_{i0}) \right] + Q, \quad (4)$$

where Q is the interval bound of all other terms, and is obtained by direct evaluation with interval arithmetic. In Eq. (4), since X_i occurs twice, there exists a dependency problem. However, we can rearrange Eq. (4) such that each X_i occurs only once; that is,

$$B(p) = \sum_{i=1}^n \left[a_i \left(X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} \right] + Q. \quad (5)$$

In this way, the dependence problem in bounding the interval polynomial is alleviated so that a sharper bound can be obtained. Since we prefer not to divide by a very small number, Eq. (5) will be used only if $|a_i| \geq \omega$, where ω is a very small positive number. If $|a_i| < \omega$, direct evaluation with Eq. (4) will be used instead.

4 Validating solver for parametric ODEs

When a typical shooting method is applied to solve a TPBVP, there is a related IVP that must be solved. For the method presented here for enclosing all solutions of the TPBVP, we also use a type of shooting method, but the related IVP has interval-valued initial states and parameters. Thus, we need an IVP solver for ODEs that can compute rigorous bounds on the state variables \boldsymbol{x} for the case in which the initial values and parameters are given by intervals. Interval methods (also called validated methods or verified methods) for ODEs (Moore, 1966) provide a natural approach for computing the desired enclosure of the state variables.

Interval methods for ODEs not only can determine guaranteed bounds on the state variables, but can also verify that a unique solution to the problem exists. Traditional interval methods

usually consist of two processes applied at each integration step (Moore, 1966). In the first process, existence and uniqueness of the solution are proved using the Picard-Lindelöf operator and the Banach fixed point theorem (Eijgenraam, 1981), and a rough enclosure of the solution is computed. In the second process, a tighter enclosure of the solution is computed. In general, both processes are realized by applying interval Taylor series expansions with respect to time, and using automatic differentiation to obtain the Taylor coefficients. An excellent review of interval methods for IVPs has been given by Nedialkov et al. (1999), and more recent work has been reviewed by Neher et al. (2007). Much work has been done for the case in which the initial values are given by intervals, and there are several available software packages that deal with this case, including AWA (Lohner, 1992), VNODE, (Nedialkov et al., 2001) and COSY VI (Berz & Makino, 1998). However, relatively little work has been done on the case in which parameters are given by intervals. In our method for enclosing all solutions of TPBVPs, we will use a new validated IVP solver for parametric ODEs (Lin & Stadtherr, 2007) called VSPODE (Validating Solver for Parametric ODEs), which is used to produce guaranteed bounds on the solutions of nonlinear dynamic systems with interval-valued initial states and parameters. This IVP solver makes use, in a novel way, of the Taylor model approach (Makino & Berz, 1996, 1999, 2003) to deal with the dependency problem on the uncertain variables (parameters and initial values). We will summarize here the basic ideas of the method used. Additional background and details are given by Lin & Stadtherr (2007).

Consider the following parametric ODE system, with state variables and parameters denoted by \mathbf{x} and $\boldsymbol{\theta}$, respectively:

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbf{X}_0, \quad \boldsymbol{\theta} \in \boldsymbol{\Theta}, \quad (6)$$

where $t \in [t_0, t_N]$ for some $t_N > t_0$. The interval vectors \mathbf{X}_0 and $\boldsymbol{\Theta}$ represent enclosures of initial values and parameters, respectively. It is desired to determine a validated enclosure of all possible solutions to this initial value problem. We denote by $\mathbf{x}(t; t_j, \mathbf{X}_j, \boldsymbol{\Theta})$ the set of solutions

$\mathbf{x}(t; t_j, \mathbf{X}_j, \Theta) = \{\mathbf{x}(t; t_j, \mathbf{x}_j, \theta) \mid \mathbf{x}_j \in \mathbf{X}_j, \theta \in \Theta\}$, where $\mathbf{x}(t; t_j, \mathbf{x}_j, \theta)$ denotes a solution of $\mathbf{x}' = \mathbf{f}(\mathbf{x}, \theta)$ for the initial condition $\mathbf{x} = \mathbf{x}_j$ at t_j . We will describe a method for determining enclosures \mathbf{X}_j of the state variables at each time step $j = 1, \dots, N$, such that $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \Theta) \subseteq \mathbf{X}_j$.

Assume that at t_j we have an enclosure \mathbf{X}_j of $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \Theta)$, and that we want to carry out an integration step to compute the next enclosure \mathbf{X}_{j+1} . Then, in the first phase of the method, the goal is to find a step size $h_j = t_{j+1} - t_j > 0$ and an a priori enclosure (coarse enclosure) $\widetilde{\mathbf{X}}_j$ of the solution such that a unique solution $\mathbf{x}(t; t_j, \mathbf{x}_j, \theta) \in \widetilde{\mathbf{X}}_j$ is guaranteed to exist for all $t \in [t_j, t_{j+1}]$, all $\mathbf{x}_j \in \mathbf{X}_j$, and all $\theta \in \Theta$. We apply a traditional interval method, with high order enclosure, to the parametric ODEs by using an interval Taylor series (ITS) with respect to time. That is, we determine h_j and $\widetilde{\mathbf{X}}_j$ such that for $\mathbf{X}_j \subseteq \widetilde{\mathbf{X}}_j^0$,

$$\widetilde{\mathbf{X}}_j = \sum_{i=0}^{k-1} [0, h_j]^i \mathbf{F}^{[i]}(\mathbf{X}_j, \Theta) + [0, h_j]^k \mathbf{F}^{[k]}(\widetilde{\mathbf{X}}_j^0, \Theta) \subseteq \widetilde{\mathbf{X}}_j^0. \quad (7)$$

Here $\widetilde{\mathbf{X}}_j^0$ is an initial estimate of $\widetilde{\mathbf{X}}_j$, k denotes the order of the Taylor expansion, and the coefficients $\mathbf{F}^{[i]}$ are interval extensions of the Taylor coefficients $\mathbf{f}^{[i]}$ of $\mathbf{x}(t)$ with respect to time. Satisfaction of Eq. (7) demonstrates that there exists a unique solution $\mathbf{x}(t; t_j, \mathbf{x}_j, \theta) \in \widetilde{\mathbf{X}}_j$ for all $t \in [t_j, t_{j+1}]$, all $\mathbf{x}_j \in \mathbf{X}_j$, and all $\theta \in \Theta$.

In the second phase of the method, we compute a tighter enclosure $\mathbf{X}_{j+1} \subseteq \widetilde{\mathbf{X}}_j$, such that $\mathbf{x}(t_{j+1}; t_0, \mathbf{X}_0, \Theta) \subseteq \mathbf{X}_{j+1}$. This will be done by using an ITS approach to compute a Taylor model $\mathbf{T}_{\mathbf{x}_{j+1}}$ of \mathbf{x}_{j+1} in terms of the initial values and parameters, and then obtaining the enclosure $\mathbf{X}_{j+1} = B(\mathbf{T}_{\mathbf{x}_{j+1}})$. For the Taylor model computations, we begin by representing the interval initial states and parameters by the Taylor models $\mathbf{T}_{\mathbf{x}_0}$ and \mathbf{T}_{θ} , respectively, with components

$$T_{x_{i0}} = (m(X_{i0}) + (x_{i0} - m(X_{i0})), [0, 0]), \quad i = 1, \dots, n, \quad (8)$$

and

$$T_{\theta_i} = (m(\Theta_i) + (\theta_i - m(\Theta_i)), [0, 0]), \quad i = 1, \dots, p. \quad (9)$$

Then, we can determine Taylor models $\mathbf{T}_{\mathbf{f}^{[i]}}$ of the interval Taylor series coefficients $\mathbf{f}^{[i]}(\mathbf{x}_i, \boldsymbol{\theta})$ by using RDA operations to compute $\mathbf{T}_{\mathbf{f}^{[i]}} = \mathbf{f}^{[i]}(\mathbf{T}_{\mathbf{x}_j}, \mathbf{T}_{\boldsymbol{\theta}})$. Using an interval Taylor series for \mathbf{x}_{j+1} with coefficients given by $\mathbf{T}_{\mathbf{f}^{[i]}}$, and incorporating a novel approach for using the mean value theorem on Taylor models, one can obtain a result for $\mathbf{T}_{\mathbf{x}_{j+1}}$ in terms of the parameters and initial states. In order to address the wrapping effect (Moore, 1966), results are propagated from one time step to the next using a new type of Taylor model, in which the remainder bound is not an interval, but a parallelepiped. That is, the remainder bound is a set of the form $\mathcal{P} = \{Av \mid v \in V\}$, where $A \in \mathbb{R}^{n \times n}$ is a real and regular matrix. If A is orthogonal, as from a QR-factorization, then \mathcal{P} can be interpreted as a rotated n -dimensional rectangle. Complete details of the computation of $\mathbf{T}_{\mathbf{x}_{j+1}}$ are given by Lin & Stadtherr (2007).

The approach outlined above, as implemented in VSPODE, has been tested by Lin & Stadtherr (2007), who compared its performance with results obtained using the popular VNODE package (Nedialkov et al., 2001) (in using VNODE, interval-valued parameters are treated as additional state variables with interval-valued initial states). For the test problems used, VSPODE provided tighter enclosures on the state variables than VNODE, and required significantly less computation time. Information about the availability of VSPODE can be obtained by contacting the authors.

5 Solution method for TPBVPs

The proposed approach for enclosing all solutions of TPBVPs is based on a branch and reduce framework. During the algorithm, the initial region $\mathbf{Z}^{(0)} = (\mathbf{X}_0^{(0)}, \boldsymbol{\Theta}^{(0)})^T = (\mathbf{X}_0, \boldsymbol{\Theta})^T$ is divided into a sequence of subregions $\mathbf{Z}^{(k)} = (\mathbf{X}_0^{(k)}, \boldsymbol{\Theta}^{(k)})^T$. Based on the solution criteria, as given by Eq. (2), certain subregions are dynamically refined (reduced) while others are excluded from consideration. For a given subregion $\mathbf{Z}^{(k)}$, with corresponding initial state interval $\mathbf{X}_0^{(k)}$ and parameter interval $\boldsymbol{\Theta}^{(k)}$, an interval IVP solver, as discussed in Section 4, can be used to verify that a unique

solution to the IVP exists for every $\mathbf{x}_0 \in \mathbf{X}_0^{(k)}$ and every $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$, and also to obtain the final state enclosure \mathbf{X}_f , which is guaranteed to enclose all solutions \mathbf{x}_f of the IVP for every $\mathbf{x}_0 \in \mathbf{X}_0^{(k)}$ and every $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$. When VSPODE (Lin & Stadtherr, 2007) is used, this final state enclosure is represented in terms of the parameters and the initial states by the Taylor model $\mathbf{T}_{\mathbf{x}_f}^{(k)}$ over $\mathbf{Z}^{(k)}$. A Taylor model $\mathbf{T}_g^{(k)}$ of the boundary condition functions $\mathbf{g}(\mathbf{x}_0, \mathbf{x}_f, \boldsymbol{\theta})$ can now be computed using the Taylor model operations discussed in Section 3.2. That is, $\mathbf{T}_g^{(k)} = \mathbf{g}(\mathbf{T}_{\mathbf{x}_0}, \mathbf{T}_{\mathbf{x}_f}^{(k)}, \mathbf{T}_{\boldsymbol{\theta}})$, where $\mathbf{T}_{\mathbf{x}_0}$ and $\mathbf{T}_{\boldsymbol{\theta}}$ are given by Eqs. (8-9) and $\mathbf{T}_{\mathbf{x}_f}^{(k)}$ is computed using VSPODE. This Taylor model can then be used, in a constraint propagation procedure, to reduce or eliminate the subregion $\mathbf{Z}^{(k)}$. For this purpose, a constraint propagation method based on Taylor models is described in the next subsection. This is followed by a summary of the overall algorithm.

5.1 Constraint propagation on Taylor models

Information given by a constraint can be used to eliminate inconsistent values from the domain of its variables. This domain reduction can then be propagated to all constraints on that variable, where it may be used to further reduce the domains of other variables. This process is known as constraint propagation (Hansen & Walster, 2004; Jaulin et al., 2001), and is widely used in various forms (e.g., hull consistency) in connection with interval methods. In this subsection, we show how to apply such a constraint propagation procedure using Taylor models.

Let T_c be the Taylor model of the function $c(\mathbf{x})$ over the interval $\mathbf{x} \in \mathbf{X}$, and say the equality constraint $c(\mathbf{x}) = 0$ needs to be satisfied. This means that, if it is possible to satisfy the constraint anywhere in the interval \mathbf{X} , the condition $0 \in B(T_c)$ must be satisfied. Thus, in the constraint propagation procedure (CPP) used, we first check if either $\underline{B}(T_c) > 0$ or $\overline{B}(T_c) < 0$. If so, then no $\mathbf{x} \in \mathbf{X}$ will ever satisfy the constraint; thus, the CPP can be stopped and \mathbf{X} discarded (eliminated from the domain of \mathbf{x}). If $\underline{B}(T_c) \leq 0 \leq \overline{B}(T_c)$, then it may still be possible to eliminate part of the

interval \mathbf{X} ; thus the CPP continues, using an approach based on the range bounding strategy for Taylor models described above.

For some component i of \mathbf{x} , let a_i and b_i be the polynomial coefficients of the terms $(x_i - x_{i0})^2$ and $(x_i - x_{i0})$ of T_c , respectively. Note that $x_{i0} \in X_i$ and is usually the midpoint $x_{i0} = m(X_i)$; the value of x_{i0} will not change during the CPP. For $|a_i| \geq \omega$, the bounds on T_c can be expressed using Eq. (5) as

$$B(T_c) = a_i \left(X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} + S_i, \quad (10)$$

where

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^n \left[a_j \left(X_j - x_{j0} + \frac{b_j}{2a_j} \right)^2 - \frac{b_j^2}{4a_j} \right] + Q. \quad (11)$$

We can reduce the computational effort to obtain S_i by recognizing that this quantity is just $B(T_c)$ less the i -th term in the summation, and $B(T_c)$ was already computed earlier in the CPP. Thus, for each i , S_i can be determined by dependent subtraction (see Section 3.1) using

$$S_i = B(T_c) \ominus \left[a_i \left(X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} \right]. \quad (12)$$

Now define the intervals $U_i = X_i - x_{i0} + \frac{b_i}{2a_i}$ and $V_i = \frac{b_i^2}{4a_i} - S_i$, so that $B(T_c) = a_i U_i^2 - V_i$. With $W_i = V_i/a_i$, then $B(T_c) = a_i(U_i^2 - W_i)$. The goal is to identify and retain only the part of X_i that contains values of x_i for which it is possible to satisfy $c(\mathbf{x}) = 0$. Since $B(T_c) = a_i(U_i^2 - W_i)$ bounds the range of $c(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}$, the constraint $c(\mathbf{x}) = 0$ can be satisfied if $0 \in B(T_c)$, that is, since $|a_i| \geq \omega$ is nonzero, if $0 \in U_i^2 - W_i$. Then, the set U_i that satisfies the constraint can be determined to be

$$U_i = \begin{cases} \emptyset & \text{if } \overline{W_i} < 0 \\ [-\sqrt{\overline{W_i}}, \sqrt{\overline{W_i}}] & \text{if } \underline{W_i} \leq 0 \leq \overline{W_i} \\ -\sqrt{\overline{W_i}} \cup \sqrt{\overline{W_i}} & \text{if } \underline{W_i} > 0 \end{cases} \quad (13)$$

The part of X_i retained is then $X_i = X_i \cap \left(U_i + x_{i0} - \frac{b_i}{2a_i} \right)$.

If $|a_i| < \omega$, then Eq. (5) should not be used (to avoid division by a very small number). However, if $|b_i| \geq \omega$ then Eq. (4) can be used instead. Following a procedure similar to that used above, we now define $U_i = X_i - x_{i0}$ and $V_i = -(B(T_c) \ominus b_i(X_i - x_{i0}))$, so that $B(T_c) = b_i U_i - V_i$. Note that all quadratic terms are now included in V_i . To identify bounds on the part of X_i that satisfies the constraint, we can now use the condition $0 \in b_i U_i - V_i$. Then, the set U_i that satisfies the constraint can be determined to be $U_i = V_i / b_i$. The part of X_i retained is then $X_i = X_i \cap (U_i + x_{i0})$. If both $|a_i|$ and $|b_i|$ are less than ω , then no CPP will be applied to reduce X_i .

The overall CPP is implemented by beginning with $i = 1$ and proceeding component by component. If, for any i , the result $X_i = \emptyset$ is obtained, then no $\mathbf{x} \in \mathbf{X}$ can satisfy the constraint; thus, \mathbf{X} can be discarded and the CPP stopped. Otherwise, the CPP proceeds until all components of \mathbf{X} have been updated. Note that, in principle, each time an improved (smaller) X_i is found, it could be used in computing S_i for subsequent components of \mathbf{X} . However, this requires recomputing the bound $B(T_c)$, which, for the function $c(\mathbf{x})$ that is of interest here, is expensive. Thus, the CPP for each component is done using the bounds $B(T_c)$ computed from the original \mathbf{X} . If, after each component is processed, \mathbf{X} has been sufficiently reduced (by more than $\omega_1 = 10\%$ by volume), then a new bound $B(T_c)$ is obtained, now over the smaller \mathbf{X} , and a new CPP is started. Otherwise, the CPP terminates.

5.2 Algorithm

A step-by-step summary of the algorithmic procedure used is given below. For simplicity, the counter k for the sequence of intervals tested is dropped, and \mathbf{Z} will always indicate the current subinterval being tested.

1. Initialization

- (a) Set the domain tolerance ϵ_x and the function tolerance ϵ_g .

(b) Set the current interval $\mathbf{Z} = (\mathbf{X}_0, \Theta)^\top$.

(c) Set the interval sequence (work list) $\mathcal{L} = \emptyset$.

(d) Set the result list $\mathcal{R} = \emptyset$.

2. Iteration

(a) Use VSPODE to compute a Taylor model \mathbf{T}_{x_f} of the final state over \mathbf{Z} , and then obtain

\mathbf{T}_g .

(b) Use \mathbf{T}_g to perform the CPP for $\mathbf{g} = \mathbf{0}$ to try to reduce \mathbf{Z} .

(c) If $\mathbf{Z} = \emptyset$, go to Step 4.

(d) If $\text{Size}(\mathbf{Z}) \leq \epsilon_x$ or $|\mathbf{B}(\mathbf{T}_g)| \leq \epsilon_g$, store the current interval \mathbf{Z} in \mathcal{R} and go to Step 4.

(e) If \mathbf{Z} is sufficiently reduced, go to Step 2a.

3. Branch

(a) Select a component to be bisected.

(b) Bisect the current interval into two subintervals.

(c) Store one of the two resulting subintervals in \mathcal{L} .

(d) Set the other subinterval to be the current interval \mathbf{Z} , and go to Step 2.

4. Subinterval selection

(a) If $\mathcal{L} = \emptyset$, go to Step 5.

(b) Remove one subinterval from \mathcal{L} as \mathbf{Z} , and go to Step 2.

5. Terminate: all solutions have been enclosed and stored in \mathcal{R} .

In Step 2(d), an ϵ -convergence test is performed. For $\text{Size}(\mathbf{Z})$ we use the width (diameter) of \mathbf{Z} , $w(\mathbf{Z}) = \max_i(\overline{Z}_i - \underline{Z}_i)$. That is, $\text{Size}(\mathbf{Z})$ is given by the maximum element width in \mathbf{Z} .

The interval magnitude $|\mathbf{B}|$ is given by $|\mathbf{B}| = \max\{|\underline{B}_i|, |\overline{B}_i|, i = 1, \dots, m\}$, that is, the largest magnitude element endpoint in \mathbf{B} . In Step 2(e), the interval \mathbf{Z} is considered sufficiently reduced if its volume at Step 2(e) is less than some percentage of its original volume at Step 2(a). In the current implementation of the algorithm, this percentage is set to be 50%. The volume of an interval is the product of its component widths. In Step 3(a), various strategies can be used to select the component to be bisected. For the problems solved here, the component with the largest relative width was selected for bisection. The relative width of component Z_i is defined as $(\overline{Z}_i - \underline{Z}_i) / \max\{|\overline{Z}_i|, |\underline{Z}_i|, 1\}$. In Step 3(c) and Step 4(b), we employ a depth-first strategy; that is, subintervals are added to and removed from the front of \mathcal{L} .

One drawback of shooting methods (Ascher et al., 1995) is that, for some values of the parameters $\boldsymbol{\theta}$ and initial values of \mathbf{x}_0 , the solution of the IVP may not exist for all $t \in [t_0, t_f]$ (i.e., it may become unbounded before the integration procedure reaches t_f). In such a case, the validating solver for parametric ODEs would fail to obtain any meaningful solution enclosures for the final state at t_f . Therefore, we need to identify the regions of \mathbf{Z} in which the solution of the IVP does not exist. In general, the nonexistence of a solution is associated with abnormal values of state. For engineering problems, natural bounds on the solution of the ODE problem are often known independently, and can be used to detect abnormal states. For example, mole fraction values that are negative or greater than one would represent abnormal states. Similarly, in a constant-volume chemical reactor, the concentration of reactant cannot at any time be negative or greater than its initial concentration. This natural bounding information can be used to discard regions of \mathbf{Z} for which solutions of the IVP do not exist because of abnormal state values. To do this, we enforce any specified natural bounds at each integration step. If, for a particular subinterval \mathbf{Z} , there is a time step after which there is no intersection between the natural bounds and the enclosures of state calculated using VSPODE, then this subinterval can be discarded since any point in it corresponds

to an IVP whose solution violates the natural bounds. If solutions to the IVP exist with abnormal state values, and it is of interest to find solutions to the TPBVP that traverse abnormal states, then the natural bounds need not be imposed.

6 Computational studies

In this section, six example problems are used to illustrate the theoretical and computational aspects of the proposed approach for enclosing all solutions of TPBVPs. All problems were solved on an Intel Pentium 4 3.2GHz machine running Red Hat Linux. The VSPODE package, with a $k = 17$ order interval Taylor series, $q = 5$ order Taylor model, and QR approach for wrapping, was used to integrate the ODE system in each problem. A variable step size procedure was used. Unless otherwise specified, a domain tolerance of $\epsilon_x = 10^{-6}$ was used and no function tolerance was used ($\epsilon_g = 0$). The algorithm was implemented in C++.

6.1 Bratu's equation

Bratu's equation is mathematically interesting as an example of bifurcation (Davis, 1962), and was first studied as a simple case of a second-order ODE by Bratu (1910). The equation arises when deriving the temperature distribution for a reaction in an infinite vessel with plane-parallel walls, and also in a simplification of a combustion reaction within a cylindrical vessel (Frank-Kamenetskii, 1969). The differential equation is

$$y'' + \lambda \exp(y) = 0, \quad t \in [0, 1] \tag{14}$$

with boundary conditions $y(0) = y(1) = 0$.

The equation can be written in general first-order ODE form, with $y = x_1$ as

$$\begin{aligned}
 x_1' &= x_2 \\
 x_2' &= -\lambda \exp(x_1) \\
 t &\in [0, 1] \\
 x_1(0) &= 0 \\
 x_1(1) &= 0.
 \end{aligned}
 \tag{15}$$

We will consider the case with $\lambda = 1$.

For this problem, all parameters are specified, as is one initial state, $x_1(0)$. To solve this TPBVP, we need to determine values of the unknown initial state, $x_2(0)$, for which the boundary conditions are satisfied. We will look for solutions in the search interval $x_2(0) \in [0, 20]$. Applying the algorithm described above, two solution enclosures for $x_2(0)$ were found, requiring 15 iterations, and 1.96 seconds of CPU time. The trajectories of $y = x_1$ and the corresponding enclosures of $x_2(0)$ are shown in Fig. 1. These results correspond to the known solution (Kierzenka, 2003) for this problem. This example demonstrates the ease with which the algorithm described here can automatically locate multiple solutions to the TPBVP.

6.2 Troesch's problem

Troesch's problem comes from the investigation of the confinement of a plasma column under radiation pressure. The problem was first described and solved by Weibel (1958). It has become a widely used test problem, and has been solved many times, including in analytical closed form (Roberts & Shipman, 1976), by using a shooting method (Troesch, 1976), by using a Laplace transform decomposition technique (Khuri, 2003), and most recently by using a modified homotopy

perturbation technique (Feng et al., 2007). The differential equation is

$$y'' = \lambda \sinh(\lambda y), \quad t \in [0, 1] \quad (16)$$

with boundary conditions $y(0) = 0$ and $y(1) = 1$.

This can be reformulated, with $y = x_1$, as the first-order ODE system

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= \lambda \sinh(\lambda x_1) \\ t &\in [0, 1] \\ x_1(0) &= 0 \\ x_1(1) &= 1. \end{aligned} \quad (17)$$

The known analytical, closed-form solution (Roberts & Shipman, 1976) of Troesch's problem is

$$y(t) = \frac{2}{\lambda} \sinh^{-1} \left[\frac{y'(0)}{2} \operatorname{sc} \left(\lambda y, 1 - \frac{1}{4}(y'(0))^2 \right) \right], \quad (18)$$

where $y'(0) = 2(1 - m)^{1/2}$ is the derivative at $t = 0$ and the constant m is the solution to the equation

$$\frac{\sinh(\lambda/2)}{(1 - m)^{1/2}} = \operatorname{sc}(\lambda, m), \quad (19)$$

with $\operatorname{sc}(\lambda, m)$ indicating the Jacobi elliptic function. We will consider the case with $\lambda = 0.5$.

Again in this problem, all parameters are specified, as is one initial state, $x_1(0)$. To solve the TPBVP, we need to determine the unknown initial state, $x_2(0)$. The search interval used is $x_2(0) \in [0, 10]$. Using the new TPBVP algorithm, a single solution enclosure $x_2(0) \in [0.95904379541306, 0.95904379541339]$ was found, requiring 5 iterations and 0.73 seconds of CPU time. Table 1 shows the corresponding enclosures of $y = x_1$ found using VSPODE. These results are interesting because they demonstrate that a published "exact" solution (Khuri, 2003; Feng et al., 2007) for this problem, obtained by numerical solution of the analytical closed form, is

in fact not very accurate, as can be seen in Table 1. Both Khuri (2003) and Feng et al. (2007) used this “exact” solution as a basis for comparison in assessing the performance of proposed numerical approximation schemes. If they had used the rigorous solution enclosure reported here as a basis for comparison, they would have found that their approximation methods were actually much more accurate than they realized.

6.3 Characteristic values of Mathieu’s equation

Mathieu’s equation (Pryce, 1993; Blanch, 1964) was originally derived in the determination of the vibrational modes of a stretched membrane with an elliptical boundary. It arises from the separation of the two-dimensional wave equation, transformed to elliptical coordinates, into two ODEs (McLachlan, 1947). The canonical form can be extracted from a number of applications, including vibrating columns of fluids (Benjamin & Ursell, 1954), the stability of an inverted pendulum (Acheson, 1993), and the vibration of a string (McLachlan, 1947). The problem in this example is to compute characteristic values (eigenvalues) of Mathieu’s equation corresponding to even, periodic solutions. These are the values of a for which solutions to the TPBVP

$$y'' + (a - 2\rho \cos 2t)y = 0, \quad t \in [0, \pi], \quad y'(0) = y'(\pi) = 0 \quad (20)$$

exist. The solution to the TPBVP is normalized so that $y(0) = 1$. We will consider the case of $\rho = 5$.

With $y = x_1$, the problem can be written in the general first-order ODE form

$$\begin{aligned}
 x_1' &= x_2 \\
 x_2' &= -(a - 2\rho \cos(2x_3))x_1 \\
 x_3' &= 1 \\
 t &\in [0, \pi] \\
 \mathbf{x}(0) &= (1, 0, 0)^T \\
 x_2(\pi) &= 0.
 \end{aligned} \tag{21}$$

Note that the additional state variable $x_3 = t$ is introduced to transform the problem into autonomous form.

In this example, the initial values are all specified, but there is an unknown parameter, the characteristic value a . To solve this TPBVP, we need to determine the values of a for which the boundary conditions are satisfied. The search interval of $[0, 100]$ was used for a . Applying the algorithm described above, nine solution enclosures for a were found, as shown in Table 2, requiring 53 iterations and 6.56 seconds of CPU time. These results are enclosures of the first- through ninth-order characteristic values, a_r , $r = 1, \dots, 9$, of Mathieu's equation. The value of r corresponds to the number of times $x_1(t)$ crosses zero between $t = 0$ and $t = \pi$ in the solution to the TPBVP. The enclosures found are consistent with the graphical results and numerical approximation formulae provided by Blanch (1964). There are a countably infinite set of characteristic values, of which we have enclosed all those in the search interval of $[0, 100]$. The corresponding state trajectories for x_1 and x_2 are shown in Fig. 2 and Fig. 3. This example demonstrates that the algorithm is able to easily deal with problems that have a large number of solutions.

6.4 Catalytic reaction in a flat particle

This example arises in a study of heat and mass transfer for a catalytic reaction within a porous catalyst flat particle (Hlaváček et al., 1968). The differential equation is the direct result of a material and energy balance. Assuming a flat geometry for the particle and that conductive heat transfer is negligible compared to convective heat transfer yields the differential equation

$$y'' = \lambda y \exp \left[\frac{\gamma \beta (1 - y)}{1 + \beta (1 - y)} \right] \quad t \in [0, 1], \quad (22)$$

where $\gamma = 20$ is a dimensionless energy of activation, and $\beta = 0.4$ is a dimensionless parameter describing heat evolution. The boundary conditions are $y'(0) = 0$ and $y(1) = 1$.

The problem can be written, with $y = x_1$, in the general first-order ODE form

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= \lambda x_1 \exp \left[\frac{\gamma \beta (1 - x_1)}{1 + \beta (1 - x_1)} \right] \\ t &\in [0, 1] \\ x_2(0) &= 0 \\ x_1(1) &= 1. \end{aligned} \quad (23)$$

Depending on the value of λ , there exists a different number of solutions. We consider the cases with $\lambda = 0.05$, 0.1 , and 0.15 .

For this problem, all parameters and one initial state, $x_2(0)$, are specified. To solve this TPBVP, we need to determine values of the unknown initial state, $x_1(0)$, for which the boundary conditions are satisfied. The search interval used is $x_1(0) \in [0, 1]$. The results of applying the new algorithm for TPBVPs, including computational performance, are shown in Table 3. Three solution enclosures are found for the case of $\lambda = 0.1$, and only one solution enclosure for both $\lambda = 0.05$ and $\lambda = 0.15$. These results are consistent with the bifurcation diagram given by Seydel (1999).

6.5 Tubular reactor model

This is the single example used by Chen & Manousiouthakis (2005) and Chen (2006) in connection with their approach for identifying all solutions to the TPBVP. The example involves a tubular reactor, whose model could be simplified to a system of two first-order ODEs:

$$\begin{aligned}x_1' &= x_2 \\x_2' &= 6(x_2 - 0.05(1 - x_1)e^{10x_1/(1+0.5x_1)}) \\t &\in [0, 1] \\x_2(0) &= 6x_1(0) \\x_2(1) &= 0.\end{aligned}\tag{24}$$

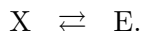
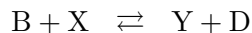
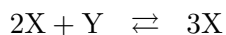
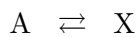
Here, t represents the dimensionless distance down the reactor, and $x_1(t)$ is the dimensionless concentration in the reactor at the position t . There are natural bounds on x_1 of $x_1 \in [0, 1]$. These natural bounds are checked at each VSPODE integration step, as described in Section 5.2.

In this example, there are no unknown parameters, but the initial value $x_1(0)$ is unknown. The natural bounds are used for the search interval, thus $x_1(0) \in [0, 1]$. Following Chen (2006), a domain tolerance of $\epsilon_x = 10^{-10}$ was used on this problem. Chen (2006) tried step sizes of both $h = 0.001$ and $h = 0.0001$ in the Runge-Kutta procedure used. As noted above, we used a variable step size procedure in VSPODE. Our algorithm for solving TPBVPs was applied and terminated after 1716 iterations and 657.5 seconds of computation time. Examining the results list \mathcal{R} showed that there were five solution enclosures. However, three of these were extremely close together, suggesting that only one of them actually enclosed a solution. In this case, one can either 1) examine each of the closely-spaced enclosures further until all but one can be eliminated, or 2) simply collapse the closely-spaced enclosures into one solution enclosure by taking their interval hull. The latter is usually sufficient, and is what is done here. Thus, the total of five solution enclosures is collapsed

to three. This is in contrast to the results of Chen & Manousiouthakis (2005) and Chen (2006), in which a total of 1627 solution enclosures were found for $h = 0.001$, and 1628 for $h = 0.0001$, with collapse to three enclosures in both cases. The three solution enclosures found using our algorithm and using the method of Chen (2006) are given in Table 4. The trajectories for x_1 and x_2 for each solution are shown in Figures 4 and 5. The computation time required by Chen’s (2006) method is not reported. Thus, to compare the performance of the two approaches, we use the widths of the enclosures found. Using the results shown in Table 4, it is seen that the enclosure widths for the method described here are 5.8% (enclosure 1), 0.24% (enclosure 2), and 0.25% (enclosure 3) of the enclosures found by Chen & Manousiouthakis (2005) and Chen (2006). Thus, for the second and third enclosures, the algorithm used here provides between two and three orders of magnitude better resolution.

6.6 Steady-state “Brusselator” with diffusion

This is a reaction-diffusion system in a fixed, bounded, one-dimensional medium (Kubíček et al., 1978). The reaction system is



The system is open, allowing the concentrations of the reactants A and B to be maintained at a constant level throughout the reaction/diffusion medium. For the case of all forward rate constants set to one and all reverse rate constants set to zero, the steady-state concentration profiles for X

and Y are described by

$$\begin{aligned} D_X \frac{\partial^2 X}{\partial z^2} &= BX + X - A - X^2 Y \\ D_Y \frac{\partial^2 Y}{\partial z^2} &= X^2 Y - BX. \end{aligned} \tag{25}$$

Here, A , B , X and Y are real and indicate the concentrations of the corresponding chemical species, D_X and D_Y are the diffusion coefficients for X and Y , respectively, and the spatial coordinate in the reaction/diffusion medium is the independent variable $z \in [0, L]$. We will consider first the problem with the fixed concentration boundary conditions $X(0) = X(L) = A$ and $Y(0) = Y(L) = B/A$.

This problem can be rewritten, with $x_1 = X$, $x_3 = Y$, and independent variable $t = z/L$ as the first-order ODE system

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= \frac{L^2}{D_X} [(B+1)x_1 - A - x_1^2 x_3] \\ x_3' &= x_4 \\ x_4' &= \frac{L^2}{D_Y} (x_1^2 x_3 - Bx_1) \\ t &\in [0, 1] \\ x_1(0) &= x_1(1) = A \\ x_3(0) &= x_3(1) = B/A. \end{aligned} \tag{26}$$

We use the constant values $D_X = 0.0016$, $D_Y = 0.008$, $A = 2$, and $B = 4.6$ (Kubíček et al., 1978; Seydel, 1988).

In this example, all parameters are specified, as are two of the initial states, $x_1(0)$ and $x_3(0)$. To solve the TPBVP, we need to determine the unknown initial states, $x_2(0)$ and $x_4(0)$. This TPBVP is known to be challenging; depending on the value of L , it may have a large number of solutions. We consider several cases using a series of values of L , namely, $L = 0.1, 0.15, 0.20, 0.25$,

and 0.30. The search intervals on $x_2(0)$ and $x_4(0)$ are set to $[-50, 50]$. Physically, x_1 and x_3 are concentrations, for which we assume natural bounds of $[0, 10]$, and x_2 and x_4 are concentration gradients (proportional to the diffusive fluxes of X and Y), for which we assume natural bounds of $[-50, 50]$. The results of applying the new algorithm, including its computational performance, are shown in Table 5, which gives point approximations of the narrow solution enclosures found. These results are consistent with the bifurcation diagrams given by Kubíček et al. (1978). Note that the trivial (spatially uniform) solution $x_2(0) = x_4(0) = 0$, which exists for any value of L , is not listed in the results, though the algorithm finds it in all cases. Especially for the larger values of L considered, this problem is indeed challenging, requiring a substantial computational effort. Nevertheless, the algorithm described here is able to rigorously enclose all solutions in the search space of interest.

We can also consider the case of zero-flux boundary conditions,

$$\begin{aligned} x_2(0) &= x_2(1) = 0 \\ x_4(0) &= x_4(1) = 0, \end{aligned} \tag{27}$$

which is also a challenging problem. In this case, we need to determine the unknown initial states, $x_1(0)$ and $x_3(0)$. As in the previous case, we assume natural bounds of $[-50, 50]$ for the concentration gradients and $[0, 10]$ for the concentrations. The search interval used for both unknown initial states (concentrations) is $[0, 10]$. The results for this case are summarized in Table 6. The trivial (spatially uniform) solution, $x_1(0) = 2$, $x_3(0) = 2.3$, which exists for any value of L , is not listed in the results, though the algorithm finds it in all cases. The results for $L = 0.1, 0.15, 0.20$, and 0.25 correspond to the bifurcation diagrams of Kubíček et al. (1978). However, for $L = 0.30$, four of the solution enclosures found do not appear in these bifurcation diagrams, and may represent solutions not found by Kubíček et al. (1978).

7 Concluding Remarks

We have presented here a new technique for solving the two-point boundary value problem (TPBVP). The method computes narrow and rigorously guaranteed enclosures of *all* solutions of a TPBVP occurring within a specified search interval. The technique is based on the use of interval analysis and features 1. Use of a new validated solver for parametric ODEs, which is used to produce guaranteed bounds on the solutions of nonlinear ODE systems with interval-valued parameters and initial states; 2. Use of a constraint propagation strategy on the Taylor models used to represent the solutions of the dynamic system. Several numerical experiments have demonstrated the usefulness of the method.

Acknowledgements

This work was supported in part by the State of Indiana 21st Century Research and Technology Fund under Grant #909010455, and by the Department of Energy under Grant DE-FG02-05CH11294.

References

- Acheson, D. J. (1993). A pendulum theorem. *Proceedings: Mathematical and Physical Sciences*, 443, 239–245.
- Ascher, U. M., Mattheij, R. M. M., & Russell, R. B. (1995). *Numerical solution of boundary value problems for ordinary differential equations*. Philadelphia, PA: SIAM.
- Benjamin, T. & Ursell, F. (1954). The stability of the plane free surface of a liquid in vertical periodic motion. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 225, 505–515.

- Berz, M. & Makino, K. (1998). Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4, 361–369.
- Blanch, G. (1964). Mathieu functions. In M. Abramowitz & I. A. Stegun, Eds., *Handbook of mathematical functions*, pp. 721–746, Washington, DC: National Bureau of Standards.
- Bratu, G. (1910). Sur certaines équations intégrales non linéaires. *Comptes Rendus*, 150, 896–899.
- Chen, Y. (2006). *Dynamic systems optimization*. Ph.D. thesis, University of California, Los Angeles, CA.
- Chen, Y. & Manousiouthakis, V. (2005). Identification of all solutions of TPBV problems. Presented at AIChE Annual Meeting, Cincinnati, OH, Paper #11e.
- Davis, H. (1962). *Introduction to nonlinear differential and integral equations*. New York, NY: Dover.
- Eijgenraam, P. (1981). The solution of initial value problems using interval arithmetic. Mathematical Centre Tracts No. 144, Stichting Mathematisch Centrum, Amsterdam, The Netherlands.
- Feng, X., Mei, L., & He, G. (2007). An efficient algorithm for solving Troesch’s problem. *Applied Mathematics and Computation*, 189, 500–507.
- Frank-Kamenetskii, D. A. (1969). *Diffusion and heat transfer in chemical kinetics*. New York, NY: Plenum Press.
- Hansen, E. & Walster, G. W. (2004). *Global optimization using interval analysis*. New York, NY: Marcel Dekker.
- Heath, M. T. (2002). *Scientific computing: An introductory survey*. 2nd ed., New York, NY: McGraw-Hill.

- Hlaváček, V., Marek, M., & Kubíček, M. (1968). Modelling of chemical reactors, X: Multiple solutions of enthalpy and mass balances for a catalytic reaction within a porous catalyst particle. *Chemical Engineering Science*, *23*, 1083–1097.
- Jaulin, L., Kieffer, M., Didrit, O., & É Walter (2001). *Applied interval analysis*. London, UK: Springer-Verlag.
- Kearfott, R. B. (1996). *Rigorous global search: Continuous problems*. Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Khuri, S. A. (2003). A numerical algorithm for solving Troesch's problem. *International Journal of Computer Mathematics*, *80*, 493–498.
- Kierzenka, J. (2003). *Tutorial on solving BVPs with BVP4C*. See http://www.mathworks.com/bvp_tutorial.
- Kubíček, M., Rýzler, V., & Marek, M. (1978). Spatial structures in a reaction-diffusion system—Detailed analysis of the “Brusselator”. *Biophysical Chemistry*, *8*, 235–246.
- Lin, Y. & Stadtherr, M. A. (2007). Validated solutions of initial value problems for parametric ODEs. *Applied Numerical Mathematics*, *58*, 1145–1162.
- Lohner, R. J. (1992). Computations of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In J. Cash & I. Gladwell, Eds., *Computational ordinary differential equations*, pp. 425–435, Oxford, UK: Clarendon Press.
- Makino, K. & Berz, M. (1996). Remainder differential algebras and their applications. In M. Berz, C. Bischof, G. Corliss, & A. Griewank, Eds., *Computational differentiation: Techniques, applications, and tools*, pp. 63–74, Philadelphia, PA: SIAM.

- Makino, K. & Berz, M. (1999). Efficient control of the dependency problem based on Taylor model methods. *Reliable Computing*, 5, 3–12.
- Makino, K. & Berz, M. (2003). Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 4, 379–456.
- Makino, K. & Berz, M. (2004). Taylor model range bounding schemes. Presented at Third International Workshop on Taylor Methods, Miami Beach, FL.
- Makino, K. & Berz, M. (2005). Verified global optimization with Taylor model-based range bounds. *Transactions on Computers*, 11, 1611–1618.
- McLachlan, N. W. (1947). *Theory and application of Mathieu functions*. Oxford, UK: Oxford University Press.
- Moore, R. E. (1966). *Interval analysis*. Englewood Cliffs, NJ: Prentice-Hall.
- Nedialkov, N. S., Jackson, K. R., & Corliss, G. F. (1999). Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105, 21–68.
- Nedialkov, N. S., Jackson, K. R., & Pryce, J. D. (2001). An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. *Reliable Computing*, 7, 449–465.
- Neher, M., Jackson, K. R., & Nedialkov, N. S. (2007). On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45, 236–262.
- Neumaier, A. (1990). *Interval methods for systems of equations*. Cambridge, UK: Cambridge University Press.
- Neumaier, A. (2003). Taylor forms—Use and limits. *Reliable Computing*, 9, 43–79.

- Pryce, J. (1993). *Numerical solution of Sturm-Liouville problems*. Charendon Press, Oxford, UK.
- Roberts, S. M. & Shipman, J. (1976). On the closed form solution of Troesch's problem. *Journal of Computational Physics*, *21*, 291–304.
- Seydel, R. (1988). *From equilibrium to chaos: practical bifurcation and stability analysis*. New York, NY: Elsevier.
- Seydel, R. (1999). *World of bifurcation: Online collection and tutorials of nonlinear phenomena*.
See <http://www.bifurcation.de>.
- Troesch, B. A. (1976). A simple approach to a sensitive two-point boundary value problem. *Journal of Computational Physics*, *21*, 279–290.
- Weibel, E. S. (1958). Confinement of a plasma column by radiation pressure. In R. K. M. Landshoff, Ed., *The plasma in a magnetic field*, pp. 60–76, Stanford, CA: Stanford University Press.

Table 1: Results for Troesch’s problem for $\lambda = 0.5$, showing VSPODE enclosures of the true solution and a published “exact” solution that is not accurate.

t	VSPODE enclosure of x_1	“exact” literature ^a solution for x_1
0.1	[0.095944349089, 0.095944349293]	0.0951769020
0.2	[0.192128747640, 0.192128747661]	0.1906338691
0.3	[0.288794400884, 0.288794400894]	0.2866534030
0.4	[0.386184846359, 0.386184846363]	0.3835229288
0.5	[0.484547164743, 0.484547164746]	0.4815373854
0.6	[0.584133248444, 0.584133248446]	0.5810019749
0.7	[0.685201148301, 0.685201148303]	0.6822351326
0.8	[0.788016522649, 0.788016522650]	0.7855717867
0.9	[0.892854216135, 0.892854216137]	0.8913669875

^a Khuri (2003), Feng et al. (2007)

Table 2: Enclosures of characteristic values of Mathieu's equation.

r	a_r
1	[1.8581875415, 1.8581875416]
2	[7.4491097395, 7.4491097396]
3	[11.5488320363, 11.5488320364]
4	[17.0965816843, 17.0965816844]
5	[25.5499717499, 25.5499717500]
6	[36.3608999793, 36.3608999794]
7	[49.2614549085, 49.2614549086]
8	[64.1988423870, 64.1988423871]
9	[81.1564549921, 81.1564549922]

Table 3: Results for catalytic reaction in a flat particle.

λ	Enclosure	$x_1(0)$	Iterations	CPU (s)
0.05	1	[0.970345, 0.970346]	17	2.12
	1	[0.922680, 0.922681]		
0.1	2	[0.505872, 0.505873]	28	4.30
	3	[0.064468, 0.064469]		
0.15	1	[0.016558, 0.016559]	26	5.01

Table 4: Results for tubular reactor problem. Underlines indicate the digits that are the same in both the lower and upper bounds.

Enclosure	$x_1(0)$ with new algorithm	$x_1(0)$ found by Chen (2006) ^a
1	[<u>0.010042462</u> 49794, <u>0.010042462</u> 55616]	[<u>0.010042462</u> , <u>0.010042463</u>]
2	[<u>0.018445428</u> 57226, <u>0.018445428</u> 62799]	[<u>0.018445417</u> , <u>0.018445440</u>]
3	[<u>0.037387123</u> 79708, <u>0.037387123</u> 97696]	[<u>0.037387088</u> , <u>0.037387160</u>]

^a step size $h = 0.0001$

Table 5: Computational results for “Brusselator” reaction-diffusion problem with fixed concentration boundary conditions. Spatially uniform solutions are not shown.

L	Solution	$(x_2(0), x_4(0))$	Iterations	CPU (s)
0.1	1	(15.6681, -4.8567)	20523	2847
	2	(-2.3150, 0.7457)		
0.15	1	(9.1534, -3.8407)	59677	18291
	2	(-4.6754, 2.1050)		
0.2	1	(1.8794, 0.4881)	86577	15037
	2	(18.6155, -6.5040)		
	3	(2.2066, -1.3165)		
	4	(-5.5272, 3.4277)		
	5	(-5.6169, 2.8125)		
	6	(-5.5457, 0.9602)		
0.25	1	(23.4979, -8.3461)	200097	40727
	2	(16.7635, -5.7769)		
	3	(-1.8836, 2.7344)		
	4	(-2.7949, 0.5932)		
	5	(-4.1053, 3.9949)		
	6	(-8.3434, 2.8572)		
0.3	1	(28.8501, -9.7524)	902361	212127
	2	(28.4637, -10.0349)		
	3	(23.5717, -8.9347)		
	4	(8.4488, -1.9617)		
	5	(-5.6412, 0.8180)		
	6	(-8.3076, 2.1976)		
	7	(-9.2360, 3.0261)		
	8	(-9.5101, 4.4103)		

Table 6: Computational results for “Brusselator” reaction-diffusion problem with zero-flux boundary conditions. Spatially uniform solutions are not shown.

L	Solution	$(x_1(0), x_3(0))$	Iterations	CPU (s)
0.1	1	(3.2401, 1.7756)	11380	1230
	2	(0.9895, 2.5105)		
0.15	1	(3.5793, 1.4852)	20249	2414
	2	(0.6837, 2.9160)		
0.2	1	(3.3498, 1.5895)	52385	7414
	2	(3.2401, 1.7756)		
	3	(1.9773, 2.6156)		
	4	(0.9895, 2.5105)		
0.25	1	(3.6295, 1.5463)	226765	42693
	2	(2.5340, 2.1115)		
	3	(0.7469, 2.6970)		
	4	(1.5099, 2.4063)		
0.3	1	(3.5595, 1.5794)	570842	124607
	2	(3.5793, 1.4852)		
	3	(3.2401, 1.7756)		
	4	(2.7700, 2.0559)		
	5	(0.9895, 2.5105)		
	6	(0.6837, 2.9160)		
	7	(0.8569, 2.5998)		
	8	(1.5169, 2.2721)		

List of Figure Captions

Figure 1. State trajectories of Bratu's equation.

Figure 2. Trajectories of x_1 for Mathieu's equation. The uppermost curve leaving $x_1(0) = 1$ on the ordinate corresponds to the characteristic value a_1 , the curve below that corresponds to a_2 , and so on, with the lowermost curve leaving $x_1(0) = 1$ corresponding to a_9 .

Figure 3. Trajectories of x_2 for Mathieu's equation. The uppermost curve leaving $x_2(0) = 0$ on the ordinate corresponds to the characteristic value a_1 , the curve below that corresponds to a_2 , and so on, with the lowermost curve leaving $x_2(0) = 0$ corresponding to a_9 .

Figure 4. Trajectories of x_1 in tubular reactor problem.

Figure 5. Trajectories of x_2 in tubular reactor problem.

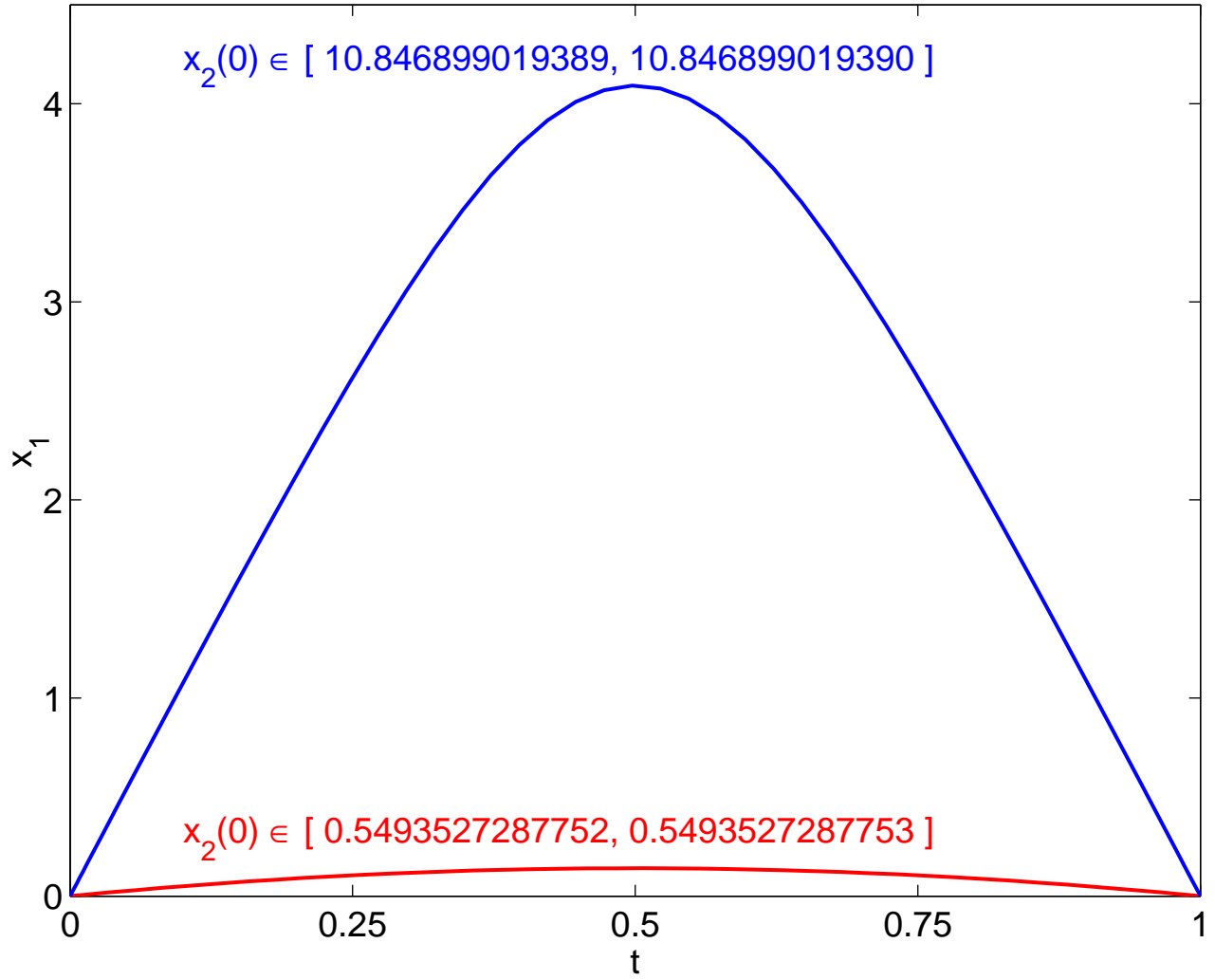


Figure 1: State trajectories of Bratu's equation.

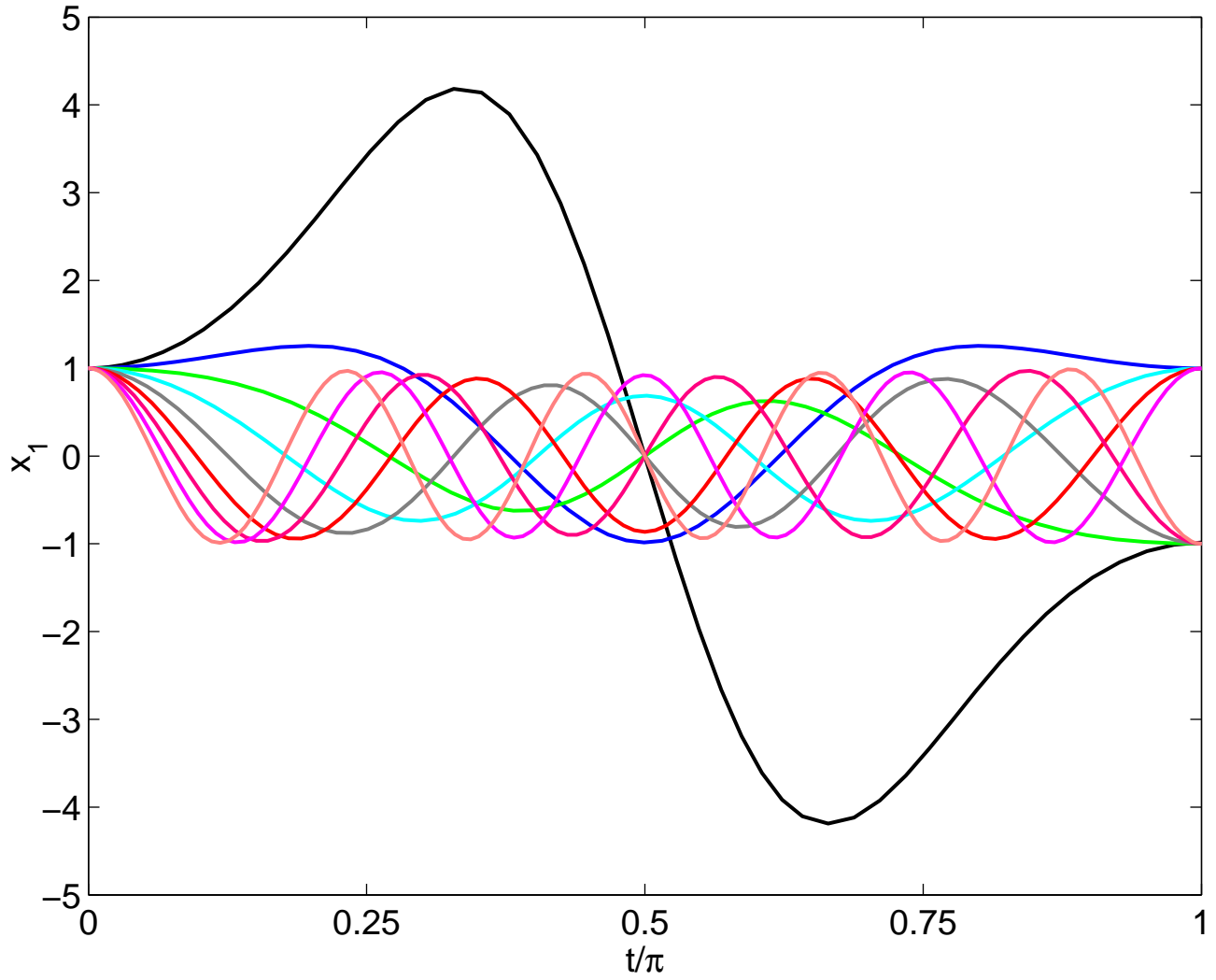


Figure 2: Trajectories of x_1 for Mathieu's equation. The uppermost curve leaving $x_1(0) = 1$ on the ordinate corresponds to the characteristic value a_1 , the curve below that corresponds to a_2 , and so on, with the lowermost curve leaving $x_1(0) = 1$ corresponding to a_9 .

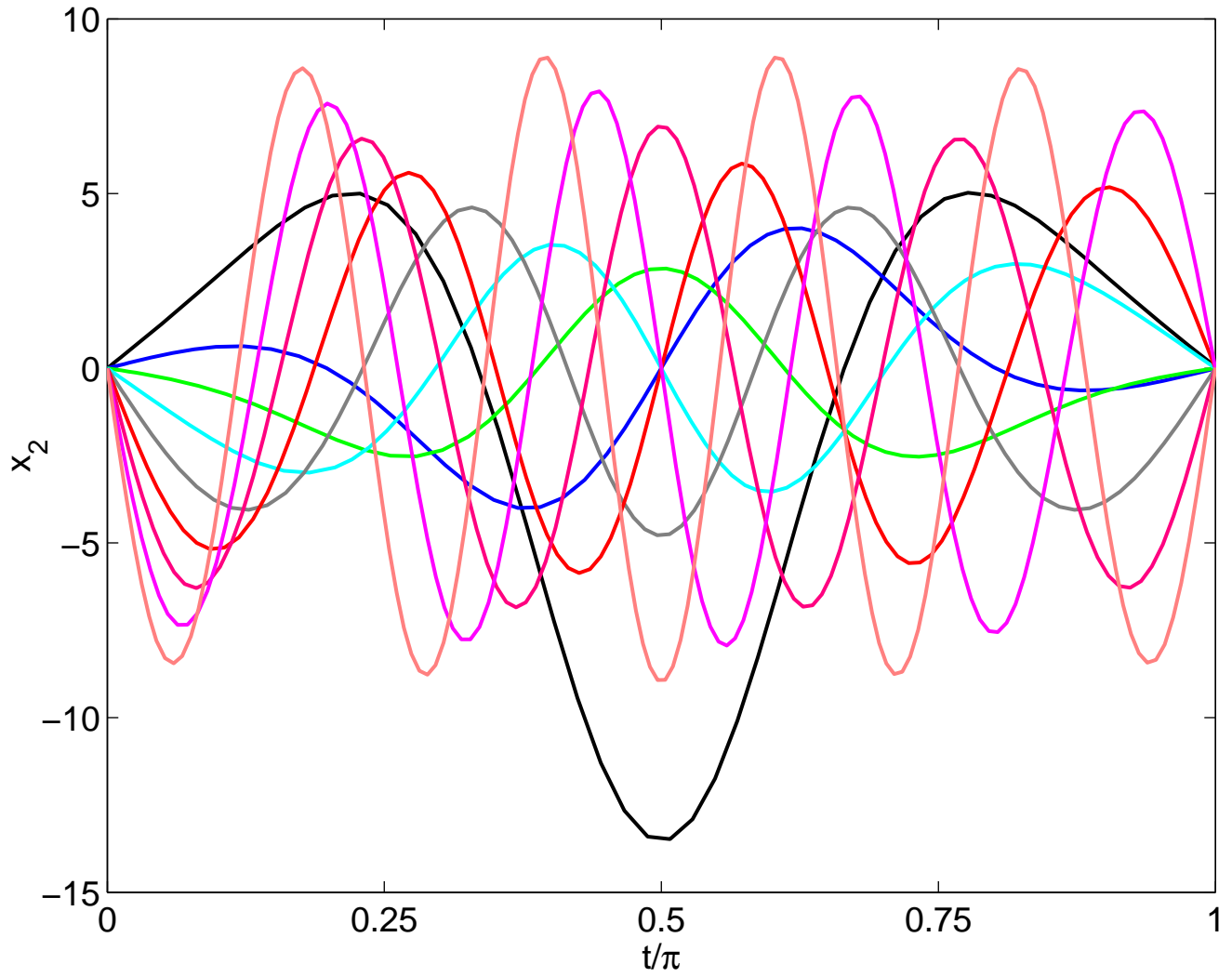


Figure 3: Trajectories of x_2 for Mathieu's equation. The uppermost curve leaving $x_2(0) = 0$ on the ordinate corresponds to the characteristic value a_1 , the curve below that corresponds to a_2 , and so on, with the lowermost curve leaving $x_2(0) = 0$ corresponding to a_9 .

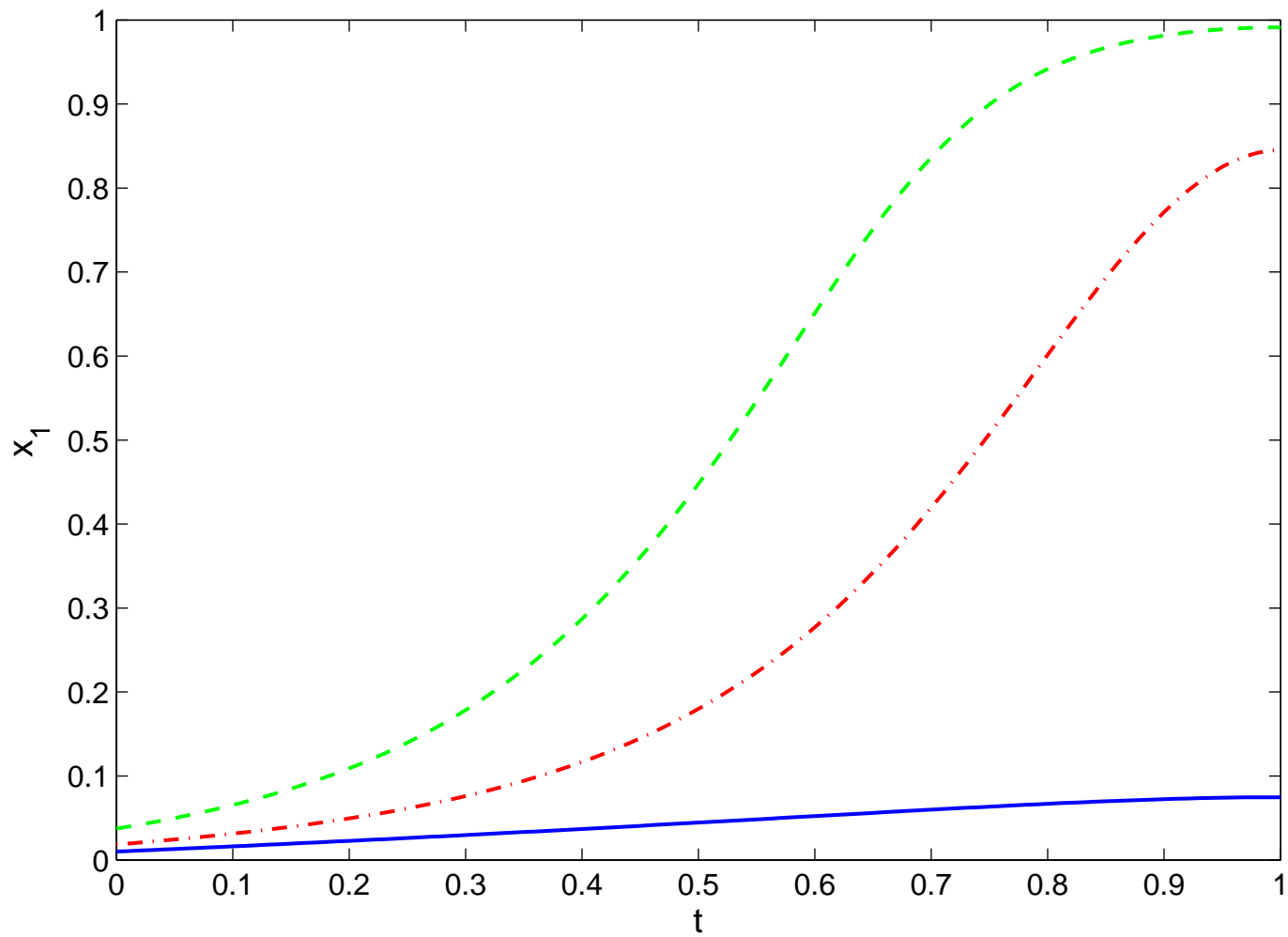


Figure 4: Trajectories of x_1 in tubular reactor problem.

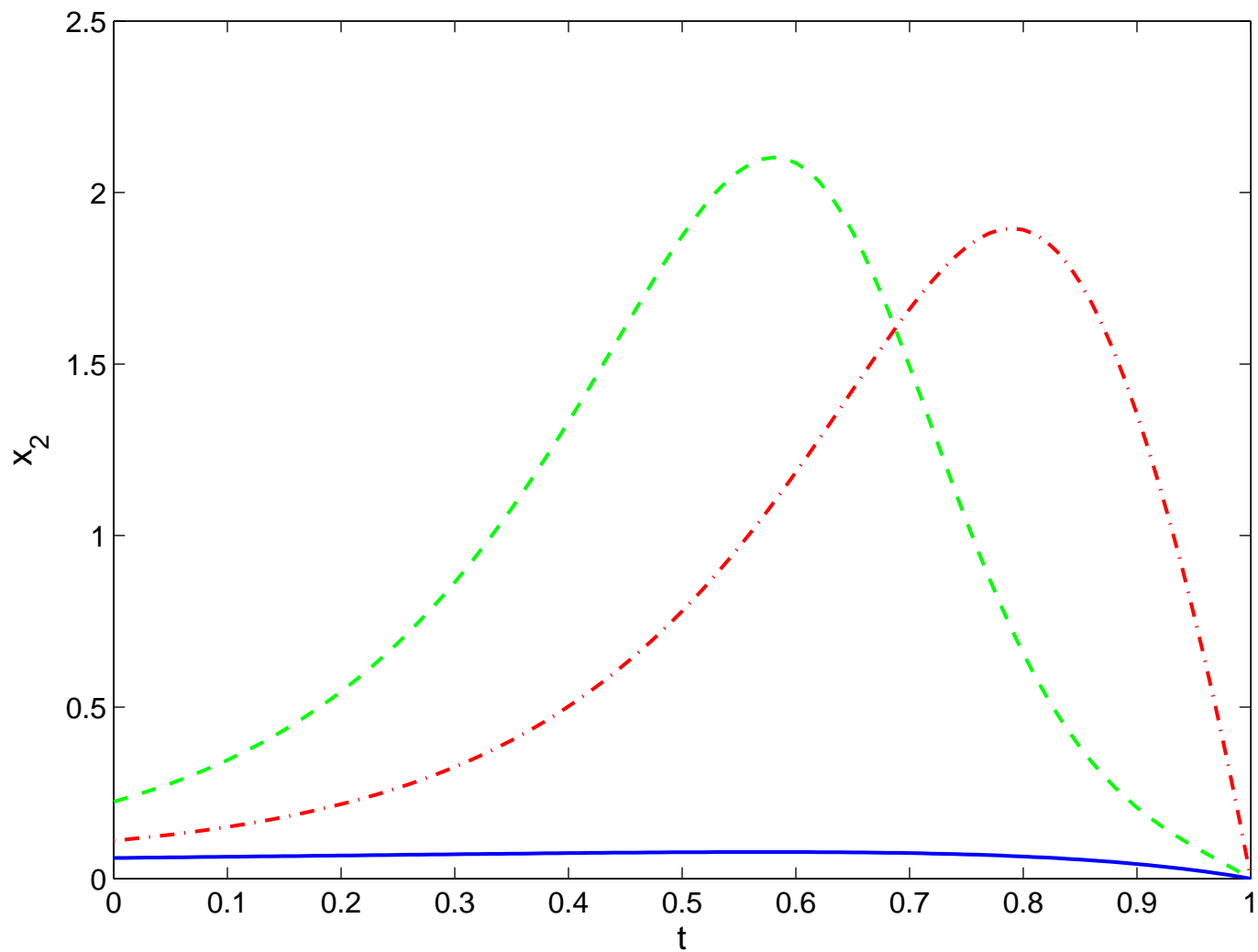


Figure 5: Trajectories of x_2 in tubular reactor problem.