Validated Solution of Initial Value Problems for ODEs with Interval Parameters

Youdong Lin and Mark A. Stadtherr*
Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

Abstract. In initial value problems for ODEs with interval-valued parameters, it is desirable in many applications to be able to determine a validated enclosure of all possible solutions to the ODE system. Much work has been done for the case in which initial values are given by intervals, and there are several available software packages that deal with this case. However, relatively little work has been done on the case in which parameters are given by intervals. We demonstrate here a new method for obtaining validated solutions of initial value problems for ODEs with interval-valued parameters. The method also accounts for interval-valued initial values. The effectiveness of the method is demonstrated using numerical examples involving kinetics in a bioreactor and motion of a double pendulum.

Keywords: ODE, IVP, Parametric uncertainty, Bioreactor kinetics, Double pendulum

1. Introduction

Initial value problems for ODEs arise naturally in many applications in engineering and science. It is often the case that the problem involves parameters and/or initial values that are not known with certainty but that can be expressed as intervals. For this situation it is desirable to be able to determine an enclosure of all possible solutions to the ODEs. Interval methods (validated methods) not only can determine such guaranteed error bounds on the true solution, but can also verify that a unique solution to the problem exists. An excellent review of interval methods for initial value problems has been given by Nedialkov et al. (1999). Much work has been done for the case in which the initial values are given by intervals, and there are several available software packages, including AWA (Lohner, 1992), VNODE (Nedialkov et al., 2001) and COSY VI (Berz and Makino, 1998), that deal with this case. However, relatively little work has been done on the case in which parameters are given by intervals. We concentrate here on the case of such parametric ODEs. However, the method demonstrated will also account for interval-valued initial values.

Since available general-purpose validated ODE solvers are focused on dealing with uncertainties in the initial values, the presence of interval parameters can cause inefficiencies because they lead to a wrapping effect. An alternative approach is to treat time-invariant interval parameters as additional state variables, with zero first-order derivatives, as suggested by Lohner (1988). Since the parameters are now treated as independent variables, tighter enclosures can be obtained.

* Author to whom all correspondence should be addressed. E-mail: markst@nd.edu

© 2006 *by authors. Printed in USA.*

However, the increase in the number of state variables, m, can result in a significant increase in the computational expense. For example, a matrix of order m must be factored at each time step in the usual methods (e.g., QR factorization) for controlling the wrapping effect. In this work, we will demonstrate a new method for efficiently determining validated solutions of ODEs with interval parameters; instead of increasing the number of state variables, this method will treat the parametric uncertainty directly. The method makes use, in a novel way, of the Taylor model approach that Makino and Berz (1996) used to deal with the dependence problem in interval arithmetic, and which they applied in COSY VI (Berz and Makino, 1998).

2. Background

2.1. Interval analysis

A real interval X is defined as the set of real numbers lying between (and including) given upper and lower bounds; that is,

$$X = \left[\underline{X}, \overline{X}\right] = \left\{x \in \Re \mid \underline{X} \leq x \leq \overline{X}\right\}. \tag{1}$$

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 $\boldsymbol{X} = (X_1, X_2, \cdots, X_n)^T$ 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

$$X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}, \tag{2}$$

where op $= \{+, -, \times, \div\}$. Interval versions of the elementary functions can be similarly defined. It should be emphasized that, when machine computations with interval arithmetic operations are done, as in the procedures outlined below, the endpoints of an interval are computed with a directed (outward) rounding. 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 (Jaulin et al., 2001; Hansen and Walster, 2004; 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.

For an arbitrary function $f(\boldsymbol{x})$, the interval extension $F(\boldsymbol{X})$ encloses all possible values of $f(\boldsymbol{x})$ for $\boldsymbol{x} \in \boldsymbol{X}$; that is, it encloses the range of $f(\boldsymbol{x})$ over $\boldsymbol{X}$. It is often computed by substituting the given interval $\boldsymbol{X}$ into the function $f(\boldsymbol{x})$ and then evaluating the function using interval arithmetic. This so-called "natural" interval extension is often wider than the actual range of function values, though it always includes the actual range. 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. There are a variety of approaches that can be used to try to tighten interval extensions (Jaulin et al., 2001; Hansen and Walster, 2004; Kearfott, 1996; Neumaier, 1990), including the use of Taylor models, as described in the next subsection.

2.2. Taylor models

Makino and Berz (1996; 1999) have described a remainder differential algebra (RDA) approach for bounding function ranges and controlling the dependency problem of interval arithmetic. This method employs high-order computational differentiation to express a function by a model consisting of a Taylor polynomial, usually a truncated Taylor series, and an interval remainder bound.

Consider a function $f : \boldsymbol{x} \in \boldsymbol{X} \subset \mathbb{R}^m \to \mathbb{R}$ that is $(q+1)$ times partially differentiable on $\boldsymbol{X}$ and let $\boldsymbol{x}_0 \in \boldsymbol{X}$. The Taylor theorem states that for each $\boldsymbol{x} \in \boldsymbol{X}$, there exists a $\zeta \in \mathbb{R}$ with $0 < \zeta < 1$ such that

$$f(\boldsymbol{x}) = \sum_{i=0}^{q} \frac{1}{i!} \left[(\boldsymbol{x} - \boldsymbol{x}_0) \cdot \bigtriangledown\right]^i f(\boldsymbol{x}_0) + \frac{1}{(q+1)!} \left[(\boldsymbol{x} - \boldsymbol{x}_0) \cdot \bigtriangledown\right]^{q+1} f\left[\boldsymbol{x}_0 + (\boldsymbol{x} - \boldsymbol{x}_0)\zeta\right], \tag{3}$$

where the partial differential operator $[\boldsymbol{g} \cdot \bigtriangledown]^k$ is

$$[\boldsymbol{g} \cdot \bigtriangledown]^k = \sum_{\substack{j_1+\cdots+j_m=k \\ 0 \le j_1, \cdots, j_m \le k}} \frac{k!}{j_1! \cdots j_m!} g_1^{j_1} \cdots g_m^{j_m} \frac{\partial^k}{\partial x_1^{j_1} \cdots \partial x_m^{j_m}}. \tag{4}$$

The last (remainder) term in (3) can be quantitatively bounded over $0 < \zeta < 1$ using interval arithmetic or other methods to obtain an interval remainder bound. The Taylor model for $f(\boldsymbol{x})$ then consists of a q-th order polynomial in $(\boldsymbol{x} - \boldsymbol{x}_0)$, $p_f(\boldsymbol{x} - \boldsymbol{x}_0)$ (the summation in (3)), and an interval remainder bound R_f. This Taylor model is denoted by $T_f = (p_f, R_f)$.

Arithmetic operations with Taylor models can be done using the RDA approach described by Makino and Berz (1996; 1999; 2003). Let T_f and T_g be the Taylor models of the functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ respectively over the interval $\boldsymbol{x} \in \boldsymbol{X}$. The Taylor model of $f \pm g$ can be represented as

$$T_{f\pm g} = (p_f, R_f) \pm (p_g, R_g) = (p_f \pm p_g, R_f \pm R_g) = (p_{f\pm g}, R_{f\pm g}). \tag{5}$$

For the the product $f \times g$,

$$f \times g \in (p_f, R_f) \times (p_g, R_g) \subseteq p_f \times p_g + p_f \times R_g + p_g \times R_f + R_f \times R_g. \tag{6}$$

Note that $p_f \times p_g$ is a polynomial of order $2q$. In order to be consistent with the q-th order polynomial in a Taylor model, this term is split into the sum of a polynomial $p_{f\times g}$ of up to q-th order, and an extra polynomial p_e containing the higher order terms. A Taylor model for the product $f \times g$ can then be given by $T_{f\times g} = (p_{f\times g}, R_{f\times g})$, with

$$R_{f\times g} = B(p_e) + B(p_f) \times R_g + B(p_g) \times R_f + R_f \times R_g. \tag{7}$$

Here $B(p) = P(\boldsymbol{X} - \boldsymbol{x}_0)$ denotes an interval bound of the polynomial $p(\boldsymbol{x} - \boldsymbol{x}_0)$ over $\boldsymbol{x} \in \boldsymbol{X}$. Similarly, an interval bound on an overall Taylor model $T = (p, R)$ will be denoted by $B(T) = B(p) + R$.

In storing and operating on a Taylor model, only the coefficients of the polynomial part $p(\boldsymbol{x} - \boldsymbol{x}_0)$ are used, and these are point valued. However, when these coefficients are computed in floating point arithmetic, numerical errors may occur and they must be bounded. To do this in our current implementation of Taylor model arithmetic, we have used the "tallying variable" approach, as described by Makino and Berz (2003). This approach has been analyzed in detail by Revol et al. (2005). This results in an error bound on the floating point calculation of the coefficients in $p(\boldsymbol{x}-\boldsymbol{x}_0)$ being added to the interval remainder bound R.

Taylor models for the reciprocal operation, as well as the intrinsic functions (exponential, logarithm, square root, sine, cosine, etc.) can also be obtained (Makino, 1998; Makino and Berz, 1996; Makino and Berz, 2003). Using these, together with the basic arithmetic operations defined above, it is possible to start with simple functions such as the constant function $k(\boldsymbol{x}) = k$, for which $T_k = (k, [0, 0])$, and the identity function $i(x_i) = x_i, i = 1, \cdots, m$, for which $T_i = (x_{i0} + (x_i - x_{i0}), [0, 0])$, and to then compute Taylor models for very complicated functions. Altogether, it is possible to compute a Taylor model for any function that can be represented 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 and Berz, 1996; Makino and Berz, 1999; Neumaier, 2002).

3. Validated Solution of Parametric ODEs

Traditional interval methods usually consist of two processes applied at each integration step (Moore, 1966; Nedialkov et al., 1999). In the first process, existence and uniqueness of the solution are proven using the Picard-Lindelöf operator and the Banach fixed point theorem (Eijgenraam, 1991), 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 (ITS) expansions with respect to time, and using automatic differentiation to obtain the Taylor coefficients. We will demonstrate here the use of a new method (Lin and Stadtherr, 2005) for the validated solution of parametric ODEs, which is used to produced guaranteed bounds on the solutions of dynamic systems with interval-valued initial states and parameters. The method uses the traditional two-phase approach, but in the second phase makes use of Taylor models to deal with the uncertain quantities (parameters and initial values). We will summarize here the basic ideas of this approach. Additional details are given by Lin and Stadtherr (2005).

Consider the following parametric ODE system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \in \boldsymbol{X}_0, \quad \boldsymbol{\theta} \in \boldsymbol{\Theta}, \tag{8}$$

where $\boldsymbol{x}$ is the m-dimensional state vector, $\boldsymbol{\theta}$ is a p-dimensional parameter vector, and $t \in [t_0, t_N]$ for some $t_N > t_0$. The interval vectors $\boldsymbol{X}_0$ 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. Also note that nonautonomous (time dependent) problems can be converted to the autonomous form given in (8). We denote by $\boldsymbol{x}(t; t_j, \boldsymbol{X}_j, \boldsymbol{\Theta})$ the set of solutions $\boldsymbol{x}(t; t_j, \boldsymbol{X}_j, \boldsymbol{\Theta}) = \{\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta}) \mid \boldsymbol{x}_j \in \boldsymbol{X}_j, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, where $\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta})$ denotes a solution of $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ for the

initial condition $\boldsymbol{x} = \boldsymbol{x}_j$ at t_j. We will describe a method for determining enclosures $\boldsymbol{X}_j$ of the state variables at each time step $j = 1, \ldots, N$, such that $\boldsymbol{x}(t_j; t_0, \boldsymbol{X}_0, \boldsymbol{\Theta}) \subseteq \boldsymbol{X}_j$.

Assume that at t_j we have an enclosure $\boldsymbol{X}_j$ of $\boldsymbol{x}(t_j; t_0, \boldsymbol{X}_0, \boldsymbol{\Theta})$, and that we want to carry out an integration step to compute the next enclosure $\boldsymbol{X}_{j+1}$. 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 a prior enclosure $\tilde{\boldsymbol{X}}_j$ of the solution such that a unique solution $\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta}) \in \tilde{\boldsymbol{X}}_j$ is guaranteed to exist for all $t \in [t_j, t_{j+1}]$, all $\boldsymbol{x}_j \in \boldsymbol{X}_j$, and all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. We apply the 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 $\tilde{\boldsymbol{X}}_j$ such that for $\boldsymbol{X}_j \subseteq \tilde{\boldsymbol{X}}_j^0$,

$$\tilde{\boldsymbol{X}}_j = \sum_{i=0}^{k-1} [0, h_j]^i \boldsymbol{F}^{[i]}(\boldsymbol{X}_j, \boldsymbol{\Theta}) + [0, h_j]^k \boldsymbol{F}^{[k]}(\tilde{\boldsymbol{X}}_j^0, \boldsymbol{\Theta}) \subseteq \tilde{\boldsymbol{X}}_j^0. \tag{9}$$

Here k denotes the order of the Taylor expansion, and the coefficients $\boldsymbol{F}^{[i]}$ are interval extensions of the Taylor coefficients $\boldsymbol{f}^{[i]}$ of $\boldsymbol{x}(t)$ with respect to time, which can be obtained recursively in terms of $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ by

$$\begin{aligned} \boldsymbol{f}^{[0]} &= \boldsymbol{x} \\ \boldsymbol{f}^{[1]} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) \\ \boldsymbol{f}^{[i]} &= \frac{1}{i} \left(\frac{\partial \boldsymbol{f}^{[i-1]}}{\partial \boldsymbol{x}} \boldsymbol{f} \right) (\boldsymbol{x}, \boldsymbol{\theta}), \qquad i \geq 2. \end{aligned} \tag{10}$$

Satisfaction of (9) demonstrates that there exists a unique solution $\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta}) \in \tilde{\boldsymbol{X}}_j$ for all $t \in [t_j, t_{j+1}]$, all $\boldsymbol{x}_j \in \boldsymbol{X}_j$, and all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

In phase 2, we compute a tighter enclosure $\boldsymbol{X}_{j+1} \subseteq \tilde{\boldsymbol{X}}_j$, such that $\boldsymbol{x}(t_{j+1}; t_0, \boldsymbol{X}_0, \boldsymbol{\Theta}) \subseteq \boldsymbol{X}_{j+1}$. This will be done by using an ITS approach to compute a Taylor model $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}$ of $\boldsymbol{x}_{j+1}$ in terms of the initial values and parameters, and then obtaining the enclosure $\boldsymbol{X}_{j+1} = B(\boldsymbol{T}_{\boldsymbol{x}_{j+1}})$. For the Taylor model computations, we begin by representing the interval initial states and parameters by the Taylor models $\boldsymbol{T}_{\boldsymbol{x}_0}$ and $\boldsymbol{T}_{\boldsymbol{\theta}}$, respectively, with components

$$T_{x_{i0}} = (m(X_{i0}) + (x_{i0} - m(X_{i0})), [0, 0]), \quad i = 1, \cdots, m, \tag{11}$$

and

$$T_{\theta_i} = (m(\Theta_i) + (\theta_i - m(\Theta_i)), [0, 0]), \quad i = 1, \cdots, p. \tag{12}$$

Then, we can determine Taylor models $\boldsymbol{T}_{\boldsymbol{f}^{[i]}}$ of the interval Taylor series coefficients $\boldsymbol{f}^{[i]}(\boldsymbol{x}_i, \boldsymbol{\theta})$ by using RDA operations to compute $\boldsymbol{T}_{\boldsymbol{f}^{[i]}} = \boldsymbol{f}^{[i]}(\boldsymbol{T}_{\boldsymbol{x}_j}, \boldsymbol{T}_{\boldsymbol{\theta}})$. Using an interval Taylor series for $\boldsymbol{x}_{j+1}$ with coefficients given by $\boldsymbol{T}_{\boldsymbol{f}^{[i]}}$, and incorporating a novel approach for using the mean value theorem on Taylor models, one can obtain a result for $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}$ in terms of the parameters and initial states. This result can be improved (tightened) by applying additional steps, based on a QR factorization approach, to further control the wrapping effect. Complete details of the computation of $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}$ are given by Lin and Stadtherr (2005).

4. Results

We now report results of numerical experiments using a C++ implementation of the method outlined above. This implementation is called VSPODE (Validating Solver for Parametric ODEs). The results for VSPODE were obtained using a $k = 17$ order interval Taylor series method, and with a $q = 5$ order Taylor model. All tests were performed on a workstation running Linux with an Intel Pentium 4 3.2GHz CPU.

4.1. Bioreactor kinetics

In a bioreactor, a simple microbial growth process (Bastin and Dochain, 1990), which involves a single biomass and single substrate, can be described using the following ODE model,

$$\dot{X} = (\mu - \alpha D)X \tag{13}$$

$$\dot{S} = D(S^i - S) - k\mu X, \tag{14}$$

where X and S are concentrations of biomass and substrate, respectively; α is the process heterogeneity parameter; D and S^i are the dilution rate and the influent concentration of substrate, respectively; k is the yield coefficient; and μ is the growth rate, which is dependent on S. We consider two models for μ, the Monod law,

$$\mu = \frac{\mu_m S}{K_S + S}, \tag{15}$$

and the Haldane law,

$$\mu = \frac{\mu_m S}{K_S + S + K_I S^2}, \tag{16}$$

where μ_m is the maximum growth rate, K_S is the saturation parameter, and K_I is the inhibition parameter. In this study, the initial value of biomass concentration X_0, and the process kinetic parameters (μ_m, K_S, and K_I) are assumed to be uncertain and given by intervals. Thus, for the Monod law, there are three uncertain quantities, and four for the Haldane law. The values of the initial conditions (X_0, S_0), the inputs (D and S^i), and parameters (α, k, μ_m, K_S, and K_I) are given in Table I.

For purposes of comparison, as a representative of traditional interval methods, we used the popular VNODE package (Nedialkov et al., 2001), with a $k = 17$ order interval Hermite-Obreschkoff QR method. Though, like other available solvers, VNODE is designed to deal with uncertain initial values, it can take interval parameter values as input. However, better performance can be obtained by treating the uncertain parameters as additional state variables with zero time derivatives; thus the parametric uncertainties become uncertainties in the initial values of the extra state variables.

Enclosures of the state variables S and X for $t \in [0, 20]$ were computing using VSPODE and VNODE with constant step size $h = 0.1$. The results were shown in Fig. 1 and Fig. 2 for the Monod law and the Haldane law, respectively. VSPODE clearly provides a better enclosure, with VNODE failing at $t = 9.3$ for the Monod law, and at $t = 6.6$ for the Haldane law. In order to allow VNODE to solve the problem all the way to $t_N = 20$, we divided the intervals into a

Table I. Bioreactor microbial growth parameters

Parameter	Value	Units	Parameter	Value	Units
α	0.5	-	μ_m	[1.19, 1.21]	day^{-1}
k	10.53	g S/ g X	K_S	[7.09, 7.11]	g S/l
D	0.36	day^{-1}	K_I	[0.49, 0.51]	$(\text{g S/l})^{-1}$
S^i	5.7	g S/l	X_0	[0.82, 0.84]	g X/l
S_0	0.80	g S/l			

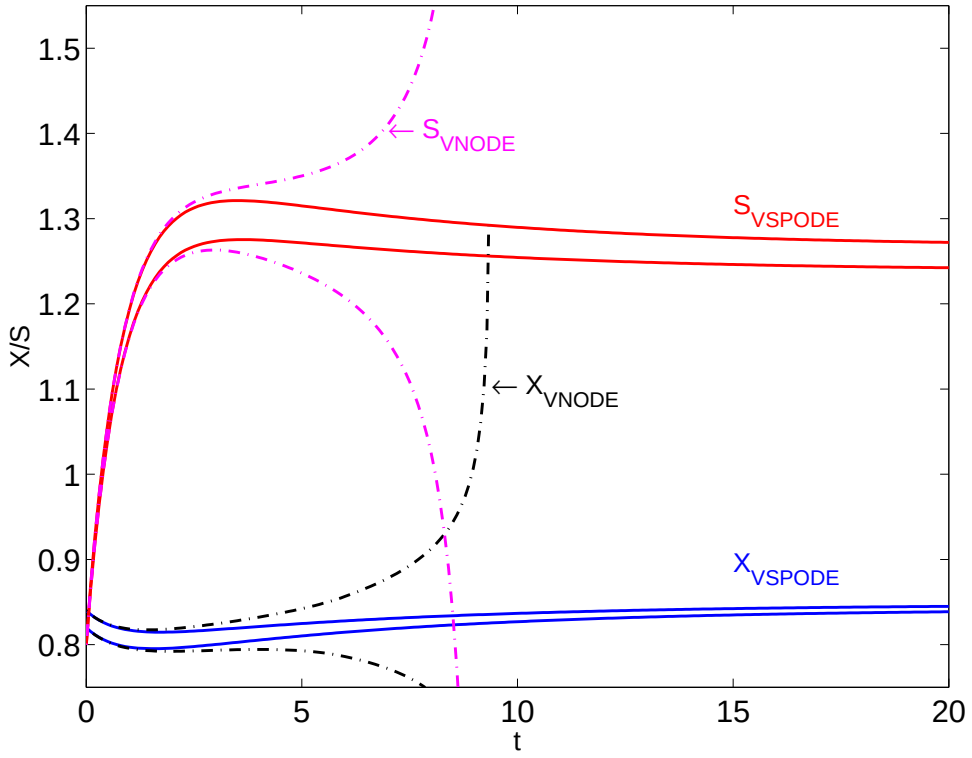

Figure 1. Enclosures for bioreactor state using the Monod law

certain number of equal-sized sub-boxes and then used VNODE to determine the solution for each sub-box. The final solution enclosure is then the union of all the enclosures resulting from each sub-box. Results showing the final solution enclosures ($t_N = 20$) and their widths, as determined using VSPODE (with no box subdivision) and VNODE with an increasing number of sub-boxes, are given in Table II for the Monod law. For example, VNODE-1000 in Table II indicates the use of 1000 sub-boxes in VNODE. Even with 1000 sub-boxes, the solution enclosure determined by VNODE is still significantly wider than that obtained from a single calculation with VSPODE, and requires about 200 times more computational time.

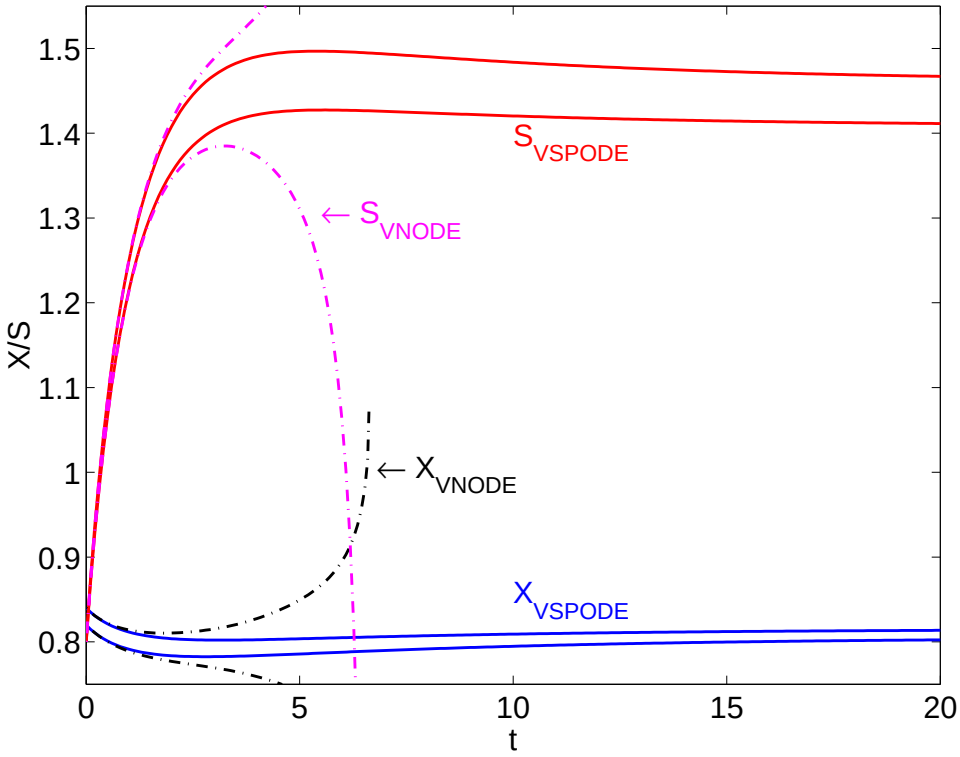

Figure 2. Enclosures for bioreactor state using the Haldane law

Table II. Results for the Monod law, showing final enclosures ($t_N = 20$).

Method	Enclosure	Width	CPU time (s)
VSPODE	[0.8386, 0.8450]	0.0064	1.34
	[1.2423, 1.2721]	0.0298	
VNODE–343	[0.8359, 0.8561]	0.0202	68.6
	[1.2309, 1.2814]	0.0505	
VNODE–512	[0.8375, 0.8528]	0.0153	102.8
	[1.2331, 1.2767]	0.0436	
VNODE–1000	[0.8380, 0.8502]	0.0122	263.1
	[1.2359, 1.2732]	0.0373	

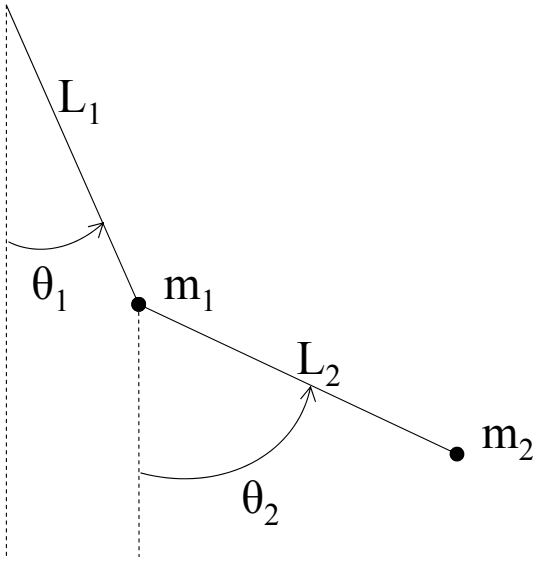

Figure 3. Schematic of double pendulum

4.2. Double pendulum

In this problem, we consider the motion of a double pendulum, as depicted in Fig. 3. This system is described by the nonlinear state equations:

$$\begin{aligned}
\dot{\theta}_1 &= \omega_1 \\
\dot{\theta}_2 &= \omega_2 \\
\dot{\omega}_1 &= \frac{-g(2m_1+m_2)\sin\theta_1 - m_2 g\sin(\theta_1-2\theta_2) - 2m_2\sin(\theta_1-\theta_2)\left[\omega_2^2 L_2 - \omega_1^2 L_1\cos(\theta_1-\theta_2)\right]}{L_1\left[2m_1+m_2-m_2\cos(2\theta_1-2\theta_2)\right]} \\
\dot{\omega}_2 &= \frac{2\sin(\theta_1-\theta_2)\left[\omega_1^2 L_1(m_1+m_2) + g(m_1+m_2)\cos\theta_1 + \omega_2^2 L_2 m_2\cos(\theta_1-\theta_2)\right]}{L_2\left[2m_1+m_2-m_2\cos(2\theta_1-2\theta_2)\right]},
\end{aligned} \tag{17}$$

where θ_1 and θ_2 are the angles of the pendulum rods (0 = vertical downwards, counter-clockwise is positive), and ω_1 and ω_2 are the angular velocities of the top and bottom rod, respectively. The mass parameters are set to $m_1 = m_2 = 1$ kg and the length parameters are set to $L_1 = L_2 = 1$ m. The parameter g is the local acceleration of gravity, which varies with latitude (greatest at the poles, lowest at the equator) and altitude. In this problem, we will treat g as an uncertain parameter in the interval $[9.79, 9.81]$ m/s^2. This corresponds roughly to the variation in the sea level value between 25° and 49° latitude (i.e., spanning the contiguous United States). The initial conditions determine the amount of potential and kinetic energy given to the system. We consider two set of initial values: 1) a relatively high-energy case with initial state of $(\theta_1, \theta_2, \omega_1, \omega_2)_0 = (0.75\pi, 0.5\pi, 0, 0)$ and 2) a relatively low-energy case with initial state of $(\theta_1, \theta_2, \omega_1, \omega_2)_0 = (0, -0.25\pi, 0, 0)$.

Enclosures of the state variables for both cases were computed using VSPODE with variable step size (automatically determined by program). The results for θ_1 and θ_2 are shown in Fig. 4 for

the high-energy case and Fig. 5 for the low-energy case. The computational times were 8.1 and 12.7 seconds, respectively. For the high-energy case, good enclosures were maintained through two full rotations of the lower pendulum and one of the upper. For the low-energy case, good enclosures were maintained through several cycles of motion. The enclosures of all state variables at some time instances, as well as the break-down time, are shown in Table III and Table IV.

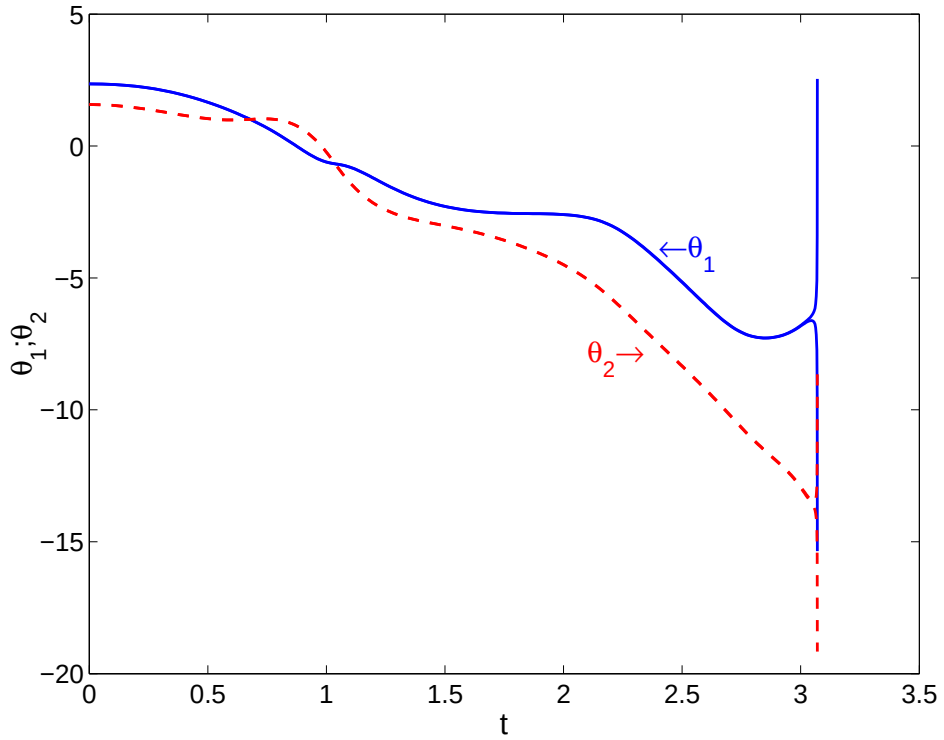

Figure 4. Enclosures of θ_1 and θ_2 for the double pendulum model, high-energy case.

Table III. Enclosures for the double pendulum model, high-energy case.

	Enclosure			
t	θ_1	θ_2	ω_1	ω_2
0.5	[1.6563, 1.6579]	[1.0368, 1.0372]	[-3.0472, -3.0408]	[-0.9327, -0.9298]
1.0	[-0.6093, -0.6067]	[-0.2392, -0.2277]	[-2.5851, -2.5397]	[-11.3007, -11.2407]
1.5	[-2.2913, -2.2883]	[-3.0230, -3.0203]	[-1.9772, -1.9647]	[-1.7512, -1.7474]
2.0	[-2.5990, -2.5978]	[-4.5055, -4.4958]	[-0.5943, -0.5778]	[-4.7957, -4.7668]
2.5	[-5.1731, -5.1512]	[-8.3532, -8.3308]	[-8.5972, -8.5801]	[-8.7733, -8.7566]
3.0	[-6.8254, -6.8000]	[-12.9548, -12.9121]	[6.3803, 6.4705]	[-12.7200, -12.5692]
3.07	FAIL			

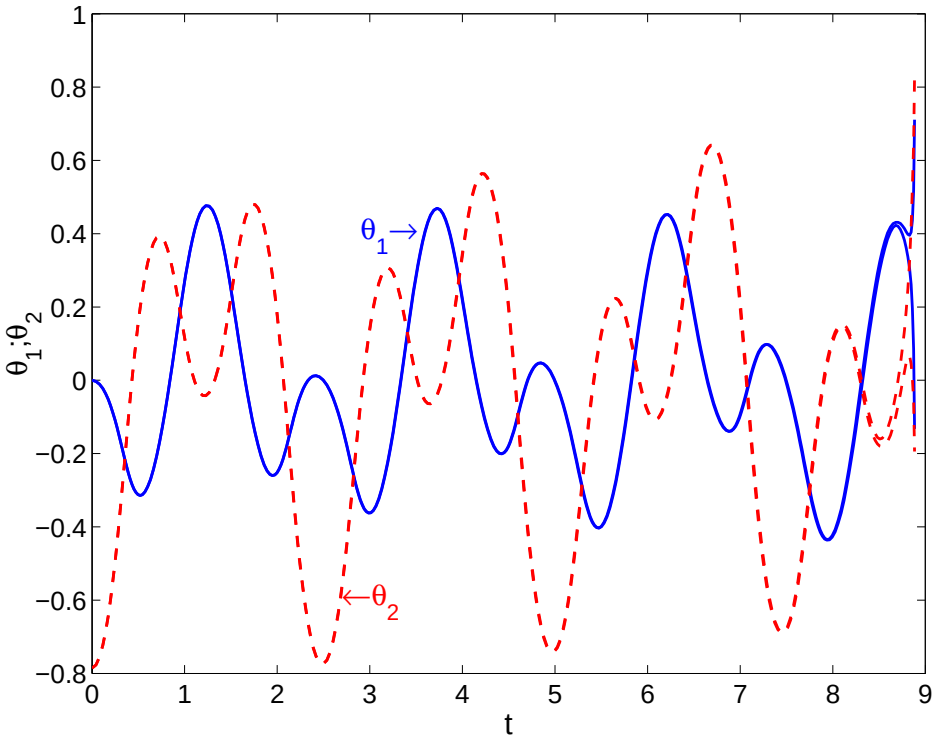

Figure 5. Enclosures of θ_1 and θ_2 for the double pendulum model, low-energy case.

Table IV. Enclosures for the double pendulum model, low-energy case.

	Enclosure			
t	θ_1	θ_2	ω_1	ω_2
1.0	[0.2688, 0.2704]	[0.1153, 0.1167]	[1.5710, 1.5734]	[-1.3996, -1.3971]
2.0	[-0.2517, -0.2509]	[0.1744, 0.1792]	[0.3534, 0.3688]	[-2.3539, -2.3340]
3.0	[-0.3623, -0.3622]	[0.1393, 0.1444]	[0.0310, 0.05463]	[1.6578, 1.6810]
4.0	[0.2161, 0.2224]	[0.3409, 0.3483]	[-1.5371, -1.5321]	[1.7902, 1.8019]
5.0	[-0.0034, -0.0004]	[-0.7395, -0.7382]	[-0.6020, -0.5848]	[0.2325, 0.2808]
6.0	[0.2927, 0.3014]	[-0.0856, -0.0818]	[1.3907, 1.4213]	[-0.6345, -0.5891]
7.0	[-0.0976, -0.0922]	[0.1977, 0.2179]	[0.7267, 0.7672]	[-2.8614, -2.8031]
8.0	[-0.4260, -0.4213]	[0.1059, 0.1150]	[0.3924, 0.4578]	[0.7548, 0.8205]
8.89	FAIL			

5. Concluding Remarks

We have demonstrated a new method for obtaining validated solutions of initial value problems for ODEs with interval-valued parameters and initial values. The dependence of the solution on t is handled using ITS methods, as in VNODE (Nedialkov et al., 2001). However, the dependence on the parameter vector $\boldsymbol{\theta}$ and the initial state $\boldsymbol{x}_0$ is handled through a novel use of Taylor models of the form described by Makino and Berz (Makino and Berz, 1996; Makino and Berz, 2003). Numerical results on a bioreactor kinetics problem and a double pendulum motion problem demonstrate that this approach provides a very efficient way to obtain a tight enclosure of all possible solutions to a parametric ODE system under uncertain conditions.

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

Bastin, G. and D. Dochain: 1990, *On-line Estimation and Adaptive Control of Bioreactors*. Amsterdam: Elsevier.

Berz, M. and K. Makino: 1998, 'Verified Integration of ODEs and Flows Using Differential Algebraic Methods on High-Order Taylor Models'. *Reliable Computing* **4**, 361–369.

Eijgenraam, P.: 1991, 'The Solution of Initial Value Problems Using Interval Arithmetic'. Technical Report Mathematical Centre Tracts No. 144, Stichting Mathematisch Centrum, Amsterdam.

Hansen, E. and G. W. Walster: 2004, *Global Optimization Using Interval Analysis*. New York: Marcel Dekker.

Jaulin, L., M. Kieffer, O. Didrit, and É. Walter: 2001, *Applied Interval Analysis*. London: Springer-Verlag.

Kearfott, R. B.: 1996, *Rigorous Global Search: Continuous Problems*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Lin, Y. and M. A. Stadtherr: 2005, 'Validated Solutions of Initial Value Problems for Parametric ODEs'. Submitted.

Lohner, R. J.: 1988, 'Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaugfgaben und Anwendungen'. Ph.D. thesis, Universität Karlsruhe.

Lohner, R. J.: 1992, 'Computations of Guaranteed Enclosures for the Solutions of Ordinary Initial and Boundary Value Problems'. In: I. G. J. Cash (ed.): *Computational Ordinary Differential Equations*. Oxford: Clarendon Press, pp. 425–435.

Makino, K.: 1998, 'Rigorous Analysis of Nonlinear Motion in Particle Accelerators'. Ph.D. thesis, Michigan State University, East Lansing, Michigan, USA.

Makino, K. and M. Berz: 1996, 'Remainder Differential Algebras and Their Applications'. In: M. Berz, C. Bishof, G. Corliss, and A. Griewank (eds.): *Computational Differentiation: Techniques, Application, and Tools*. Philadelphia: SIAM, pp. 63–74.

Makino, K. and M. Berz: 1999, 'Efficient Control of the Dependency Problem Based on Taylor Model Methods'. *Reliable Computing* **5**, 3–12.

Makino, K. and M. Berz: 2003, 'Taylor Models and Other Validated Functional Inclusion Methods'. *Int. J. Pure Appl. Math.* **4**, 379–456.

Moore, R. E.: 1966, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall.

Nedialkov, N. S., K. R. Jackson, and G. F. Corliss: 1999, 'Validated Solutions of Initial Value Problems for Ordinary Differential Equations'. *Appl. Math. Comput.* **105**, 21–68.

Nedialkov, N. S., K. R. Jackson, and J. D. Pryce: 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.

Neumaier, A.: 1990, *Interval Methods for Systems of Equations*. Cambridge, England: Cambridge University Press.

Neumaier, A.: 2002, 'Taylor Forms–Use and Limits'. *Reliable Computing* **9**, 43–79.

Revol, N., K. Makino, and M. Berz: 2005, 'Taylor Models and Floating Point Arithmetic: Proof That Arithmetic Operations Are Bounded in COSY'. *J. Logic Algebr. Progr.* **64**, 135–154.