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, USA

2nd NSF Workshop on Reliable Engineering Computing, Georgia Tech University,
Savannah, GA, February 22–24, 2006

Outline

- Motivating Problem: Bioreactor Simulation
- Background
- Overview of Method
- Examples and Results
  - Lotka-Volterra Problem
  - Lorenz Problem
  - Double Pendulum Problem
  - Bioreactor Problem
- Concluding Remarks

Motivating Example – Bioreactor Simulation

- In a bioreactor, microbial growth may be described by

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

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

where X and S are concentrations of biomass and substrate, respectively.

- The growth rate μ may be given by

$$\mu = \frac{\mu_m S}{K_S + S} \quad \text{(Monod Law)}$$

or

$$\mu = \frac{\mu_m S}{K_S + S + K_I S^2} \quad \text{(Haldane Law)}$$

Motivating Example – Bioreactor Simulation

- Problem data

	Value	Units		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]	(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			

- Three parameters (μ_m, K_S and K_I) and one initial state (X_0) are uncertain and given by intervals.

- Problem: Determine a validated enclosure of all possible solutions to this ODE system.

- Issue: Standard tools for validated solution of ODEs are designed to deal with interval-valued initial states, not interval-valued parameters.

Problem Definition

- Consider

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \in \boldsymbol{X}_0$$

$$\boldsymbol{\theta} \in \boldsymbol{\Theta}$$

  $\boldsymbol{x}$ = state vector (m variables)

  $\boldsymbol{\theta}$ = parameter vector (p parameters)

  $\boldsymbol{X}_0$ = interval enclosure of $\boldsymbol{x}_0$

  $\boldsymbol{\Theta}$ = interval enclosure of $\boldsymbol{\theta}$

- Consider time steps $h_j = t_{j+1} - t_j$, $j = 0, \dots, N-1$
- Notation: $\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta})$ denotes a solution of $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ for the initial condition $\boldsymbol{x} = \boldsymbol{x}_j$ at $t = t_j$ and $\boldsymbol{x}(t; t_j, \boldsymbol{X}_j, \boldsymbol{\Theta})$ is the set of solutions $\boldsymbol{x}(t; t_j, \boldsymbol{X}_j, \boldsymbol{\Theta}) = \{\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta}) \mid \boldsymbol{x}_j \in \boldsymbol{X}_j, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$
- Problem: Determine enclosures $\boldsymbol{X}_j$ of the state variables at each time t_j, $j = 1, \dots, N$, such that $\boldsymbol{x}(t_j; t_0, \boldsymbol{X}_0, \boldsymbol{\Theta}) \subseteq \boldsymbol{X}_j$

Background – Interval Taylor Series

- In an Taylor series expansion of $\boldsymbol{x}(t)$ with respect to t, the coefficients can be obtained recursively in terms of $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ using

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

- Values of these coefficients can be easily generated using automatic differentiation techniques.
- For an interval Taylor series (ITS), the coefficients $\boldsymbol{F}^{[i]}$ are interval enclosures of $\boldsymbol{f}^{[i]}$.

Background – Taylor Models

- Taylor Model $T_f = (p_f, R_f)$: Bounds a function $f(\boldsymbol{x})$ over $\boldsymbol{X}$ using a q-th order Taylor polynomial p_f and an interval remainder bound R_f, usually from a truncated Taylor series.

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

$$R_f = \frac{1}{(q+1)!} \left[(\boldsymbol{x} - \boldsymbol{x}_0) \cdot \bigtriangledown \right]^{q+1} F\left[\boldsymbol{x}_0 + (\boldsymbol{x} - \boldsymbol{x}_0)\zeta \right]$$

where,

$$\boldsymbol{x}_0 \in \boldsymbol{X}; \quad \zeta \in [0, 1]$$

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

- Store and operate on coefficients of p_f only. Floating point errors are accumulated in R_f.

Background – Taylor Model Operations

- Taylor model of $f \pm g$

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

- Taylor model of $f \times g$

$$\begin{aligned} f \times g &\in (p_f, R_f) \times (p_g, R_g) \\ &\subseteq p_f \times p_g + p_f \times R_g + p_g \times R_f + R_f \times R_g \end{aligned}$$

Split $p_f \times p_g$ into q-th order part $p_{f \times g}$ and higher-order terms p_e. Then

$$T_{f \times g} = (p_{f \times g}, R_{f \times g})$$

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

$B(p)$ indicates an interval bound on the function p.

- Reciprocal operation and intrinsic functions can also be defined.

Background – Interval IVPs

- Consider standard ODE system (non-parametric)

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \in \boldsymbol{X}_0$$

- "Standard" approach (step $j+1$): Assuming $\boldsymbol{X}_j$ is known, then
  - Phase 1: Compute a coarse enclosure $\tilde{\boldsymbol{X}}_j$ and prove existance and uniqueness. Use fixed point iteration with Picard operator using high-order interval Taylor series.
  - Phase 2: Refine the coarse enclosure to obtain $\boldsymbol{X}_{j+1}$. Use high-order interval Taylor series with Taylor coefficients bounded using mean value theorem. Reduce wrapping effect using QR-factorization approach.
- Implementations include AWA and VNODE.

Method for Parametric ODEs

- Consider again parametric ODE system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \in \boldsymbol{X}_0$$

$$\boldsymbol{\theta} \in \boldsymbol{\Theta}$$

- To apply standard methods, can treat parameters as additional state variables with zero derivative (Lohner, 1988)
- Our method for parametric ODE system: Assuming $\boldsymbol{X}_j$ is known, then
  - Phase 1: Same as "standard" approach. Compute a coarse enclosure $\tilde{\boldsymbol{X}}_j$ and prove existance and uniqueness. Use fixed point iteration with Picard operator using high-order interval Taylor series.
  - Phase 2: Refine the coarse enclosure to obtain $\boldsymbol{X}_{j+1}$. Use Taylor models in terms of the uncertain parameters and initial states.
- Implemented in VSPODE (Validating Solver for Parametric ODEs) (Lin and Stadtherr, 2005).

Method for Phase 2

- Represent uncertain initial states and parameters using Taylor models $\boldsymbol{T}_{\boldsymbol{x}_0}$ and $\boldsymbol{T}_{\boldsymbol{\theta}}$, with components

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

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

- Compute Taylor models $\boldsymbol{T}_{\boldsymbol{f}^{[i]}}$ for the interval Taylor series coefficients using Taylor model operations and obtain the polynomial part of $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}$.
- Determine the remainder bound of $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}$ by the mean value theorem and reduce the wrapping effect using a QR factorization approach, where the remainder is represented by $R_{\boldsymbol{x}_{j+1}} = A_{j+1}\boldsymbol{V}_{j+1}$.
- Compute the enclosure $\boldsymbol{X}_{j+1} = B(\boldsymbol{T}_{\boldsymbol{x}_{j+1}})$ by bounding over $\boldsymbol{X}_0$ and $\boldsymbol{\Theta}$.

Examples and Results

- Computations done with Intel Pentium 4 3.2GHz CPU on a Linux workstation.
- For comparsions, VNODE was used, with interval parameters treated as additional state variables
- VSPODE run using

$\longrightarrow$ $q = 5$ (order of Taylor model)

$\longrightarrow$ $k = 17$ (order of interval Taylor series)

$\longrightarrow$ QR

- VNODE run using

$\longrightarrow$ $k = 17$ order interval Hermite-Obreschkoff

$\longrightarrow$ QR

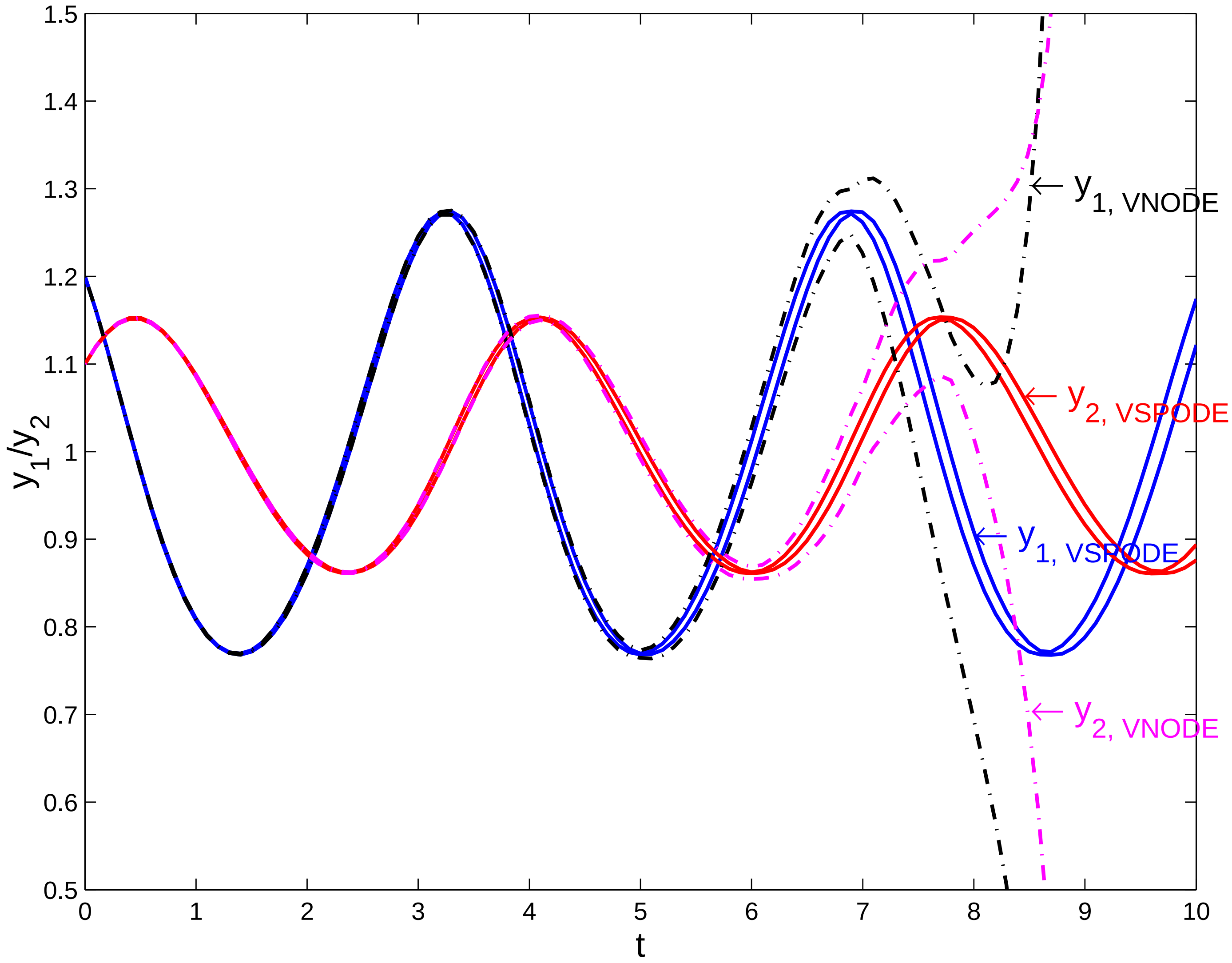
Example 1. Lotka-Volterra Problem

- ODE model is

$$\dot{x}_1 = \theta_1 x_1 (1 - x_2)$$
$$\dot{x}_2 = \theta_2 x_2 (x_1 - 1)$$
$$\boldsymbol{x}_0 = (1.2, 1.1)^{\mathrm{T}}$$
$$\theta_1 \in [2.99, 3.01]$$
$$\theta_2 \in [0.99, 1.01]$$

- Integrate from $t_0 = 0$ to $t_N = 10$.
- Constant step size of $h = 0.1$ used in both VSPODE and VNODE.

Example 1. Lotka-Volterra Problem



(Eventual breakdown of VSPODE at $t = 31.8$)

Example 1. Lotka-Volterra Problem

- To allow VNODE to integrate further:
  - Parameters intervals can be subdivided into equal-sized subintervals.
  - Apply VNODE to each parameter subinterval.
  - Final enclosure is the union of enclosures determined from each subinterval.
- VNODE-NN indicates use of NN parameter subintervals.

Example 1. Lotka-Volterra Problem

Method	Final Enclosure ($t = 10$)	Width	CPU time (s)
VSPODE	[1.120873, 1.173607]	0.052734	0.59
	[0.875994, 0.893471]	0.017477	
VNODE–16	[1.110859, 1.182814]	0.071955	1.42
	[0.872528, 0.898407]	0.025879	
VNODE–36	[1.116350, 1.177431]	0.061081	3.14
	[0.874924, 0.895612]	0.020688	
VNODE–64	[1.118151, 1.175692]	0.057541	5.59
	[0.875651, 0.894736]	0.019085	
VNODE–100	[1.118999, 1.174881]	0.055882	8.68
	[0.875975, 0.894337]	0.018362	

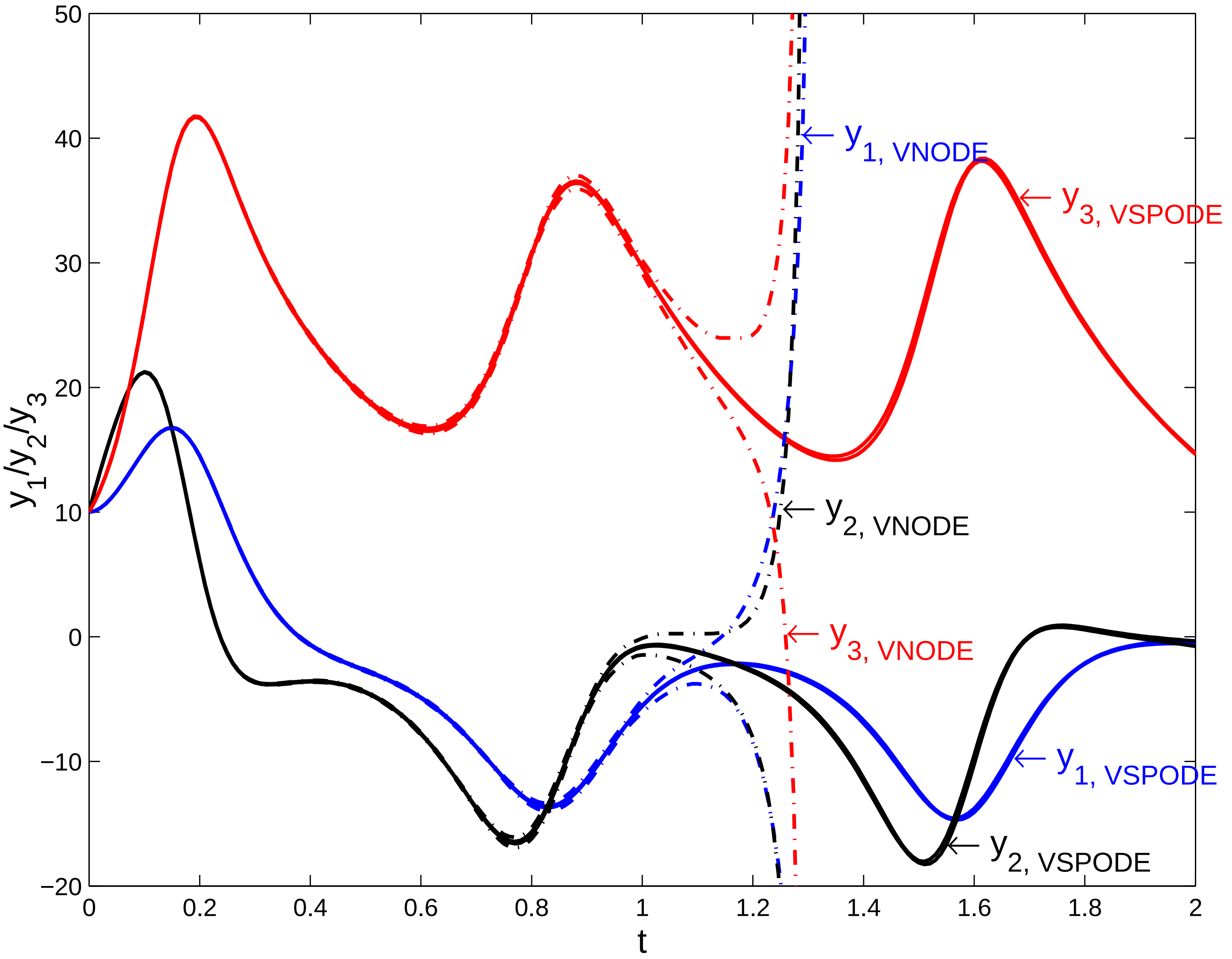
Example 2. Lorenz Problem

- ODE model is

$$
\begin{aligned}
\dot{x}_1 &= \theta_1(x_2 - x_1) \\
\dot{x}_2 &= x_1(\theta_2 - x_3) - x_2 \\
\dot{x}_3 &= x_1 x_2 - \theta_3 x_3 \\
\boldsymbol{x}_0 &= (10, 10, 10)^{\mathrm{T}} \\
\theta_1 &\in 10 + [-0.01, 0.01] \\
\theta_2 &\in 28 + [-0.01, 0.01] \\
\theta_3 &\in 8/3 + [-0.01, 0.01]
\end{aligned}
$$

- Integrate from $t_0 = 0$ to $t_N = 2$.
- Constant step size of $h = 0.01$ used in both VSPODE and VNODE.

Example 2. Lorenz Problem

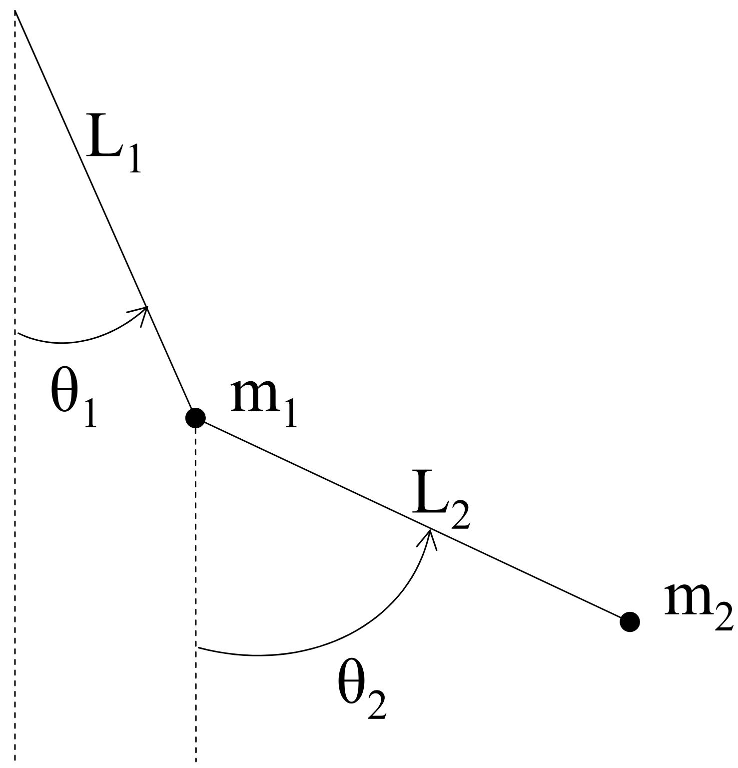


(Eventual breakdown of VSPODE at $t = 2.8$)

Example 2. Lorenz Problem

Method	Final Enclosure ($t = 2$)	Width	CPU time (s)
VSPODE	[-0.582033, -0.342358]	0.2397	2.66
	[-0.769513, -0.369357]	0.4002	
	[14.633803, 14.737535]	0.1037	
VNODE–125	[-8.663336, 7.988072]	16.6514	33.7
	[-10.060512, 8.797511]	18.8580	
	[9.031894, 21.106684]	12.0748	
VNODE–512	[-0.920184, 0.041287]	0.9615	141.5
	[-1.321734, 0.245595]	1.5673	
	[14.352124, 15.010891]	0.6588	
VNODE–1000	[-0.770156, -0.136139]	0.6340	263.1
	[-1.077794, -0.036474]	1.0413	
	[14.502030, 14.869122]	0.3671	

Example 3. Double Pendulum Problem



$$m_1 = m_2 = 1 \text{ kg}$$

$$L_1 = L_2 = 1 \text{ m}$$

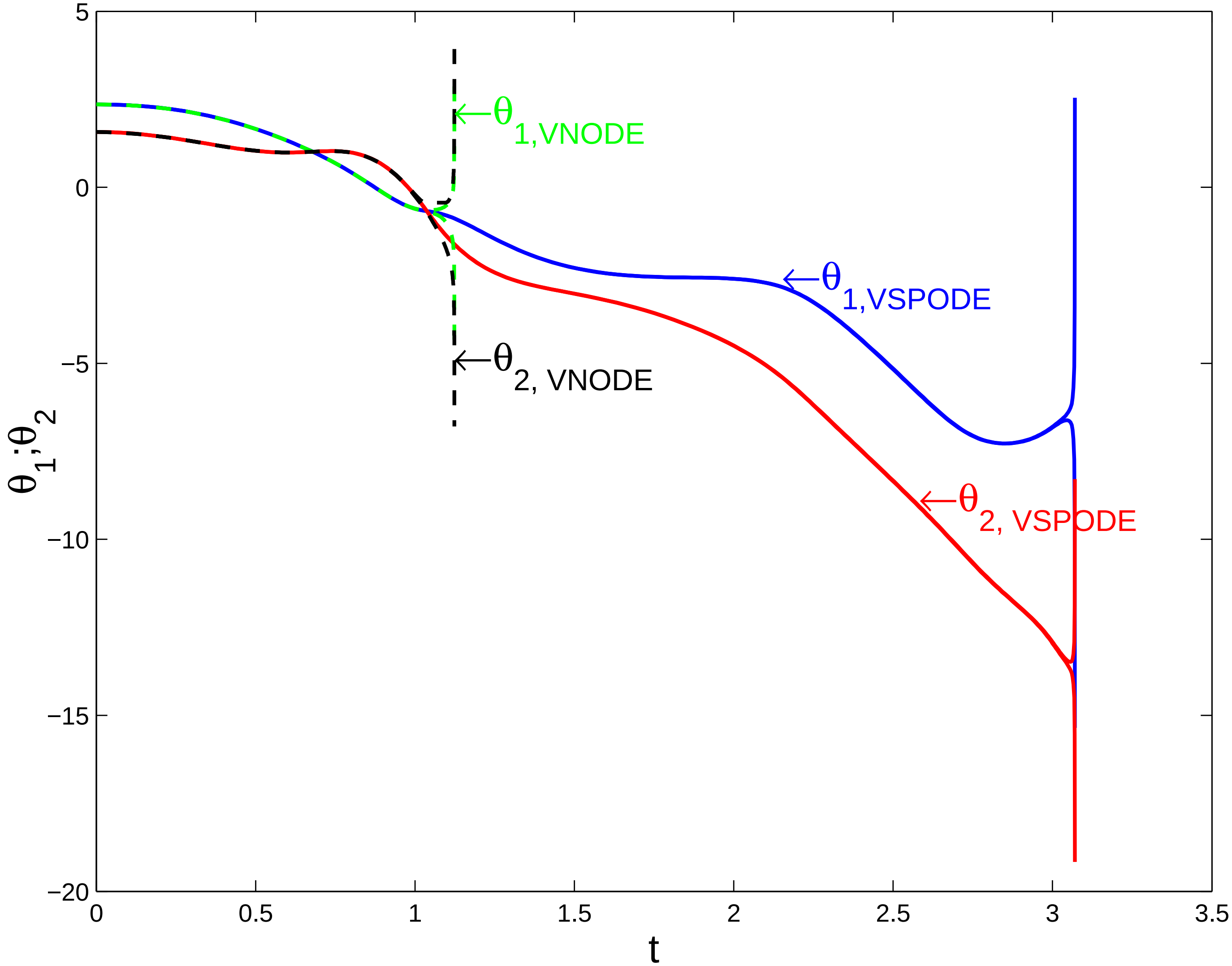
Example 3. Double Pendulum Problem

- ODE model is

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

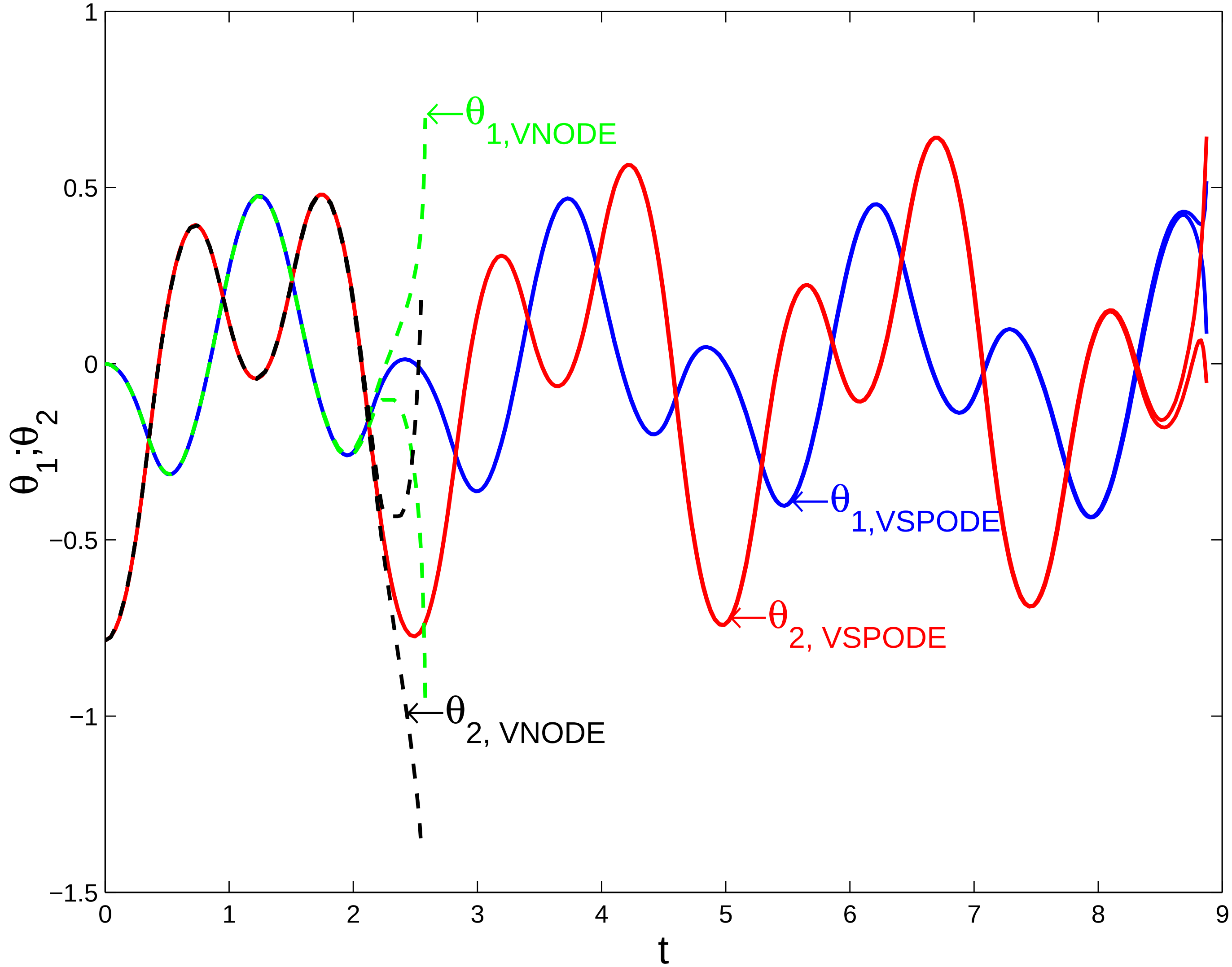
- Local acceleration of gravity $g \in [9.79, 9.81]$ m/s^2.
- This corresponds roughly to the variation in sea level g between 25° and 49° latitude (i.e. spanning the contiguous United States).
- Two cases for initial states:
  - Relatively high energy: $(\theta_1, \theta_2, \omega_1, \omega_2)_0 = (0.75\pi, 0.5\pi, 0, 0)$
  - Relatively low energy: $(\theta_1, \theta_2, \omega_1, \omega_2)_0 = (0, -0.25\pi, 0, 0)$
- Variable step size used in both VSPODE and VNODE.

Example 3. Double Pendulum Problem



Relatively high-energy case

Example 3. Double Pendulum Problem



Relatively low-energy case

Example 4. Bioreactor Problem

- In a bioreactor, microbial growth may be described by

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

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

where X and S are concentrations of biomass and substrate, respectively.

- The growth rate μ may be given by

$$\mu = \frac{\mu_m S}{K_S + S} \quad \text{(Monod Law)}$$

or

$$\mu = \frac{\mu_m S}{K_S + S + K_I S^2} \quad \text{(Haldane Law)}$$

Example 4. Bioreactor Problem

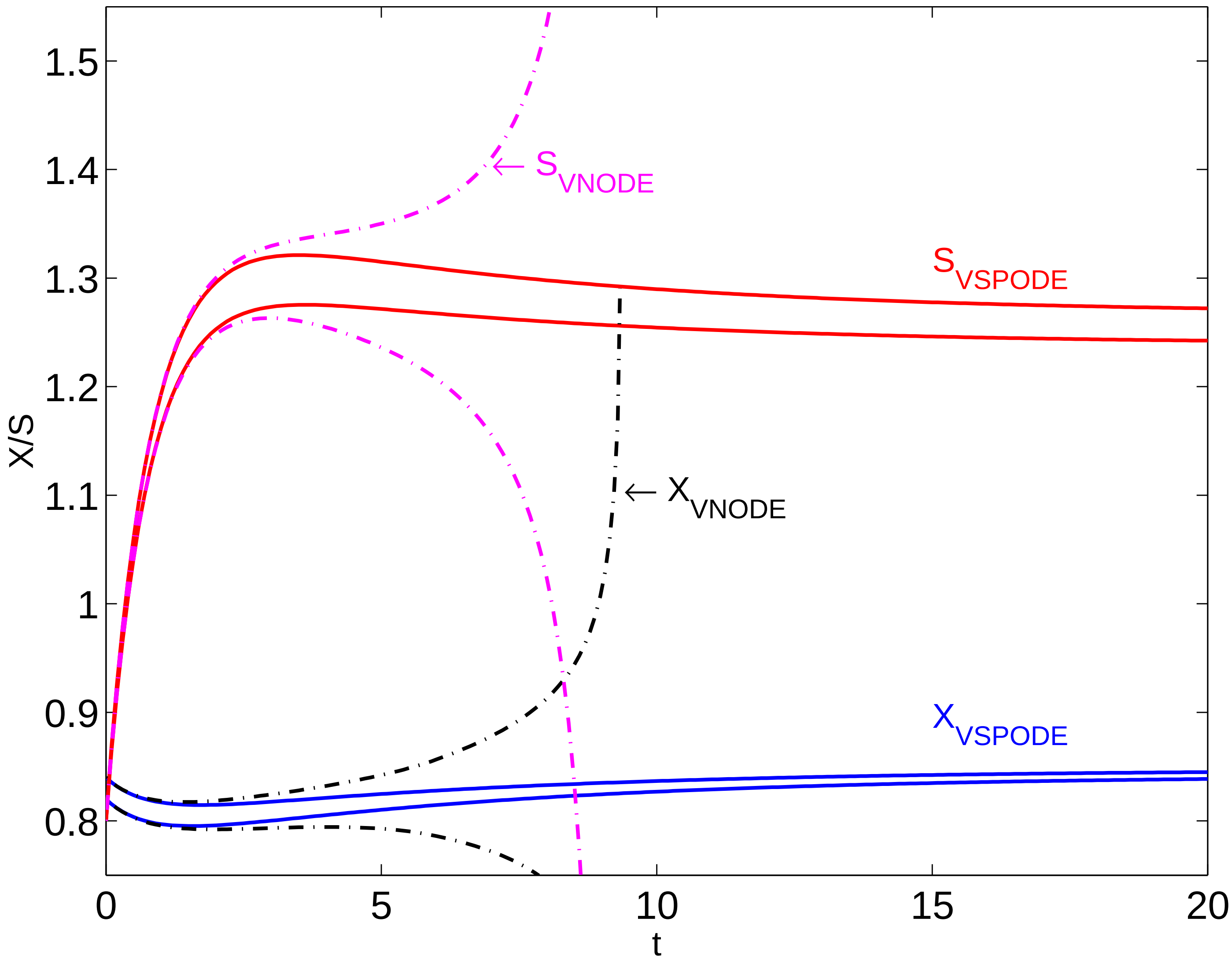
- Problem data

	Value	Units		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]	(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			

- Integrate from $t_0 = 0$ to $t_N = 20$.

- Constant step size of $h = 0.1$ used in both VSPODE and VNODE.

Example 4. Bioreactor Problem – Monod Law

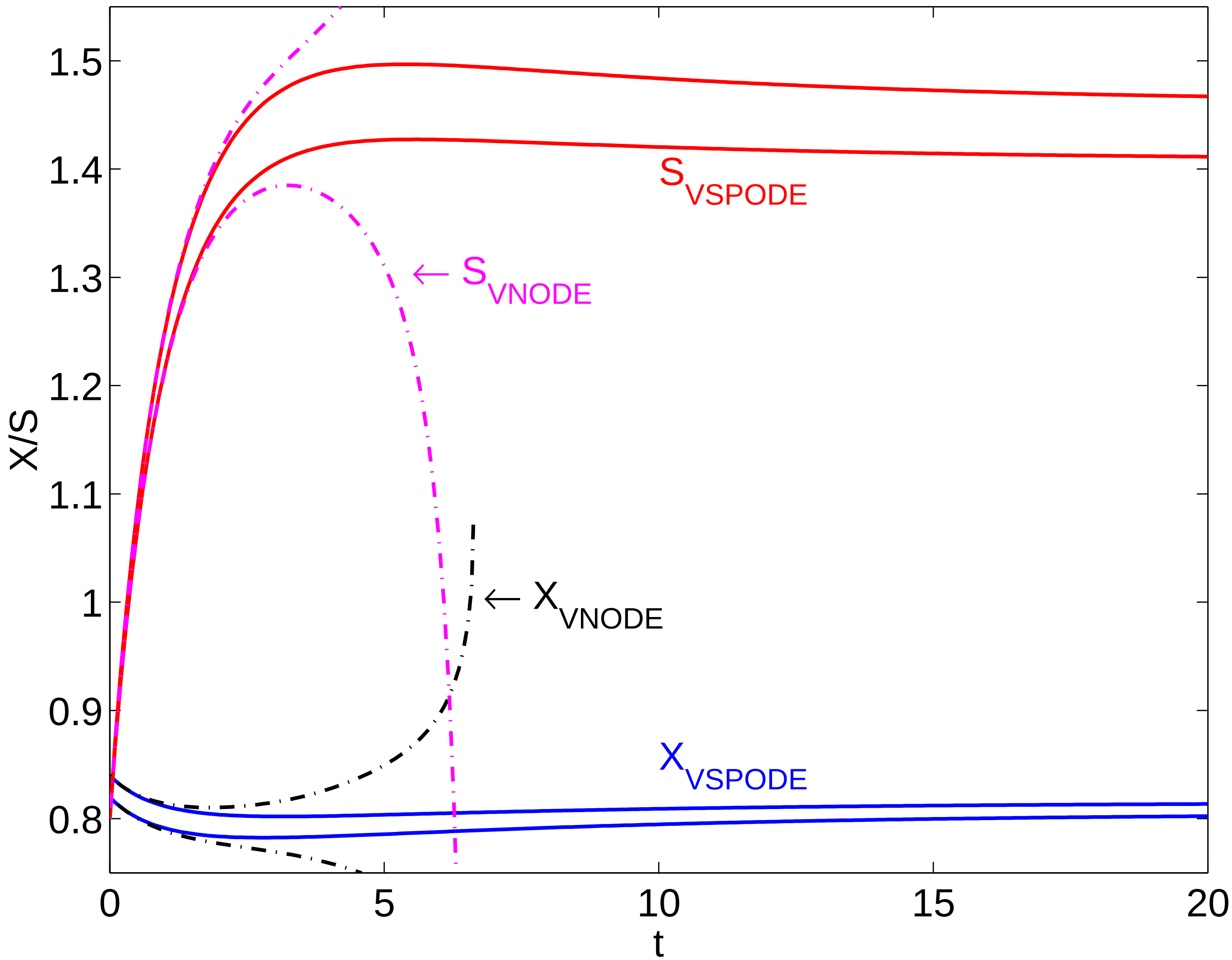


(VSPODE does not break down at longer t)

Example 4. Bioreactor Problem – Monod Law

Method	Final Enclosure ($t = 20$)	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	

Example 4. Bioreactor Problem – Haldane Law



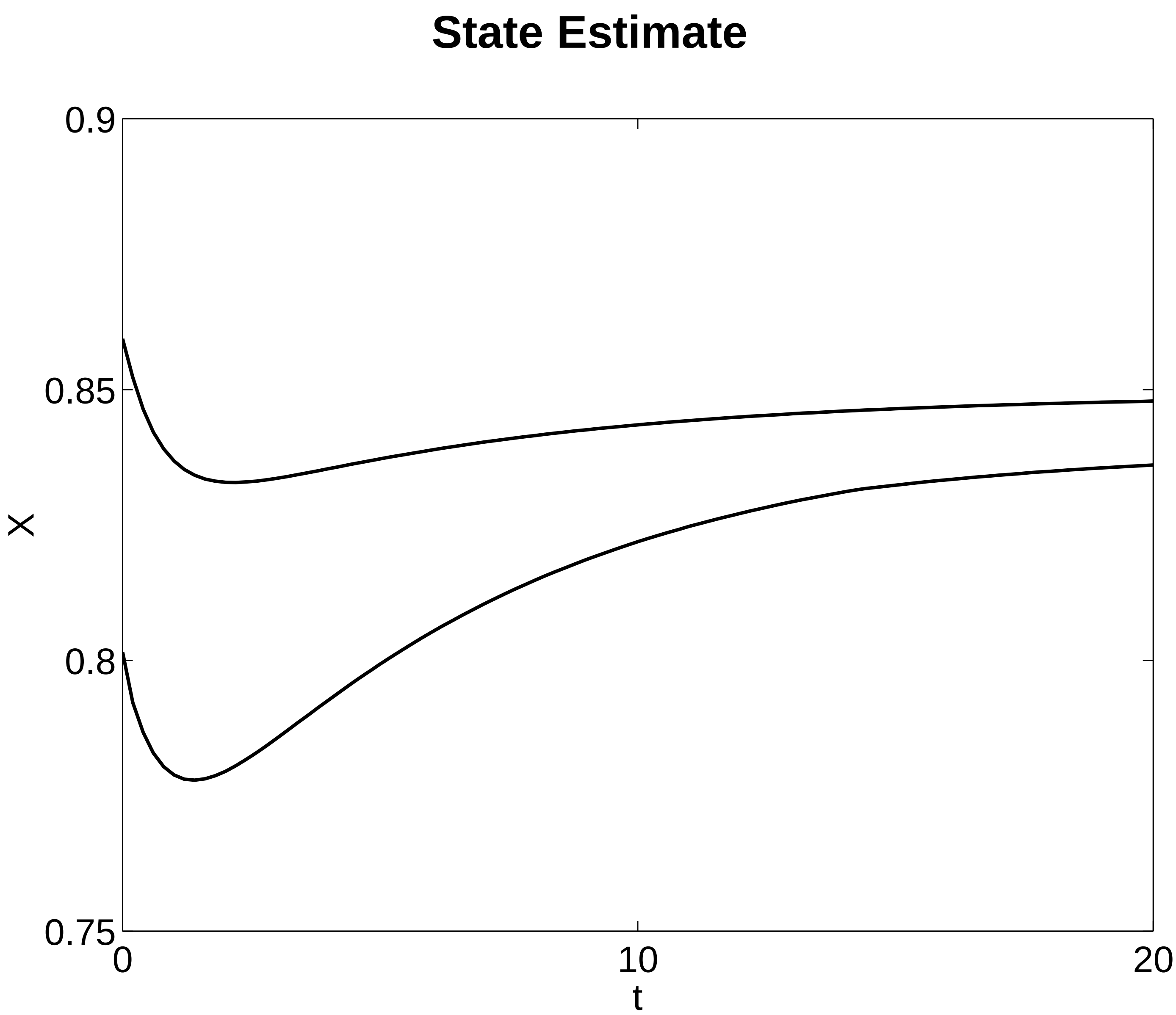
(VSPODE does not break down at longer t)

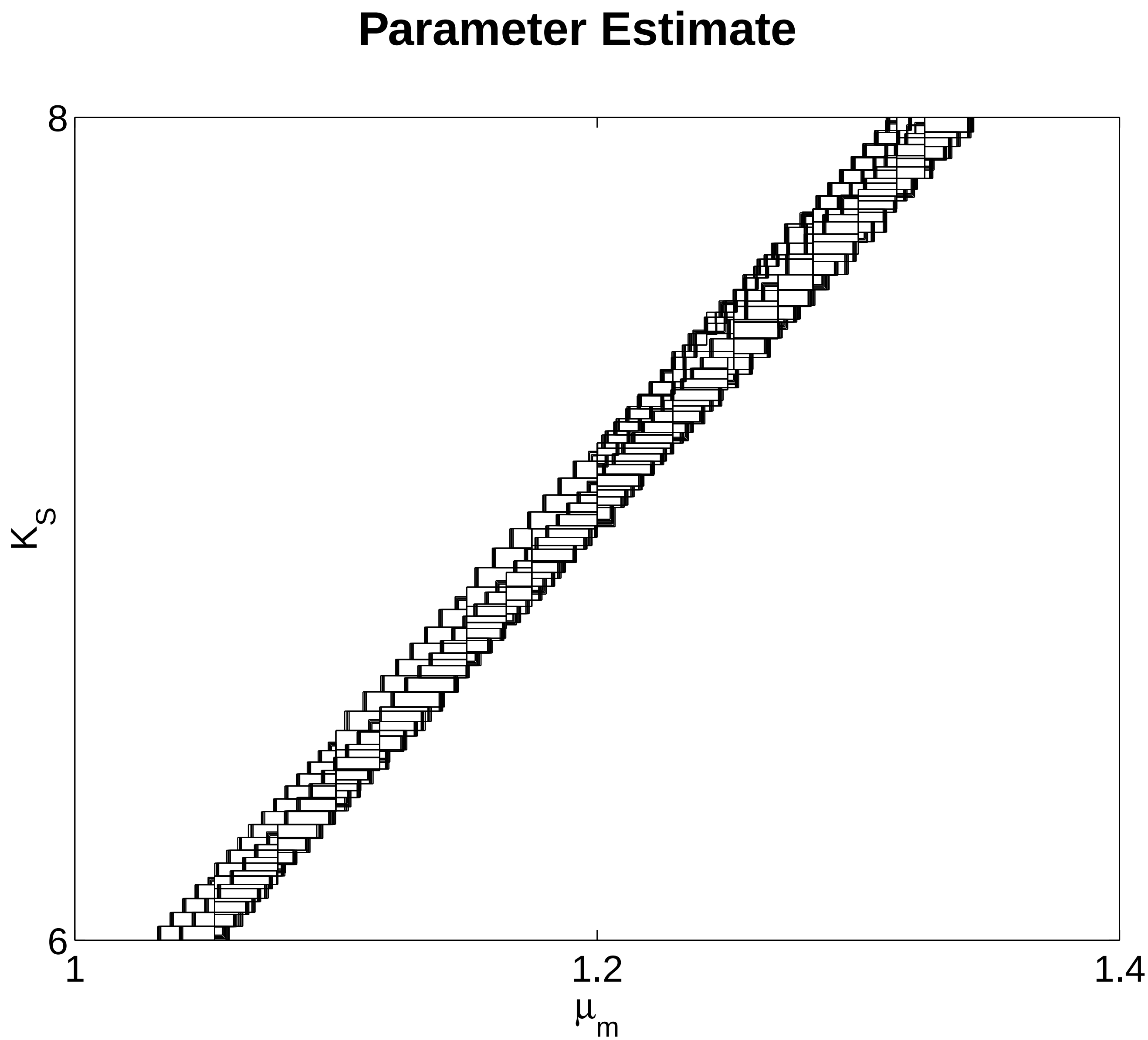
Related Problem – State and Parameter Estimation

- Consider again the bioreactor problem.
- Bounded-error (1%) measurements of S at $t_j, j = 1, \ldots, N$ are available.
- Estimate the other state variable X and the parameters μ_m, K_S and K_I.
- New problem data

	Value	Units		Value	Units
α	0.5	-	μ_m	[1.0, 1.4]	day^{-1}
k	10.53	g S/ g X	K_S	[6, 8]	g S/l
D	0.36	day^{-1}	K_I	[0.0025, 0.01]	(g S/l)$^{-1}$
S^i	5.7	g S/l	X_0	[0.4, 1.2]	g X/l
S_0	0.8 $\times$ [0.99, 1.01]	g S/l			

- Use VSPODE with constraint propagation procedure on Taylor models (Lin and Stadtherr, 2006).


State Estimate
0.9
0.85
0.8
0.75
X
0
10
20
t


Parameter Estimate
8
6
K_S
1
1.2
1.4
μ_m

Concluding Remarks

- The validated solution of parametric ODEs is a subproblem in many applications of interest.
- An approach was demonstrated for the direct handling of uncertainty in model parameters in the validated solution of ODEs.
- A standard two-phase approach was used
  - The dependence on t was handled using an interval Taylor series approach, as in standard methods (e.g. VNODE).
  - The dependence on parameters (and initial states) was handled using Taylor models in Phase 2 of the approach.
- Significant performance improvements were observed in comparison with VNODE.
- Funding
  - Indiana 21st Century Research & Technology Fund
  - U. S. Department of Energy