Deterministic Global Optimization for Dynamic Systems Using Interval Analysis

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

• Background

• Tools
  – Interval Analysis
  – Taylor Models
  – Validated Solutions of Parametric ODE Systems

• Algorithm Summary

• Computational Studies

• Concluding Remarks
Background

- Many practically important physical systems are modeled by ODE systems.
- Optimization problems involving these models may be stated as

\[
\min_{\theta, x_\mu} \phi [x_\mu(\theta), \theta; \mu = 0, 1, \ldots, r] \\
\text{s.t.} \quad \dot{x} = f(x, \theta) \\
\quad x_0 = x_0(\theta) \\
\quad x_\mu(\theta) = x(t_\mu, \theta) \\
\quad t \in [t_0, t_r] \\
\quad \theta \in \Theta
\]

- Sequential approach: Eliminate \(x_\mu\) using parametric ODE solver, obtaining an unconstrained problem in \(\theta\)

- May be multiple local solutions – need for \textit{global} optimization
Deterministic Global Optimization of Dynamic Systems

Much recent interest, mostly combining branch-and-bound and relaxation techniques, e.g.,

- **Esposito and Floudas (2000):** \( \alpha \)-BB approach
  - Rigorous values of \( \alpha \) not used: no theoretical guarantees

- **Chachuat and Latifi (2003):** Theoretical guarantee of \( \epsilon \)-global optimality

- **Papamichail and Adjiman (2002, 2004):** Theoretical guarantee of \( \epsilon \)-global optimality

- **Singer and Barton (2006):** Theoretical guarantee of \( \epsilon \)-global optimality
  - Use convex underestimators and concave overestimators to construct two bounding IVPs, which are then solved to obtain lower and upper bounds on the state trajectories.
  - Bounding IVPs are not solved rigorously, so state bounds are not computationally guaranteed.
Deterministic Global Optimization of Dynamic Systems

Our approach: **Branch-and-reduce algorithm based on interval analysis and using Taylor models**

- **Basic ideas**
  - Use local optimizations to obtain an upper bound $\hat{\phi}$ on the global minimum
  - Compute Taylor models of the state variables using a **new validated solver for parametric ODEs** (VSPODE) (Lin and Stadtherr, 2006)
  - Compute the Taylor model $T_\phi$ of the objective function
  - Perform constraint propagation procedure using $T_\phi \leq \hat{\phi}$, to reduce the parameter (decision variable) space
  - Use branch-and-bound

- Can implement to obtain either an $\epsilon$-global minimum, or (using interval Newton approach) the **exact ($\epsilon = 0$) global minimum**
Interval Analysis

- A real interval $X = [a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$ is a segment on the real number line.

- An interval vector $X = (X_1, X_2, \ldots, X_n)^T$ is an $n$-dimensional rectangle.

- Basic interval arithmetic for $X = [a, b]$ and $Y = [c, d]$ is

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

- Interval elementary functions (e.g., $\exp(X)$, $\sin(X)$) are also available.

- The *interval extension* $F(X)$ encloses all values of $f(x)$ for $x \in X$; that is,

$$\{f(x) \mid x \in X\} \subseteq F(X)$$

- Interval extensions computed using interval arithmetic may lead to overestimation of function range (the interval “dependency” problem).
Taylor Models

- Taylor Model $T_f = (p_f, R_f)$: Bounds a function $f(x)$ over $X$ using a $q$-th order Taylor polynomial $p_f$ and an interval remainder bound $R_f$.

- Could obtain $T_f$ using a truncated Taylor series:

$$p_f = \sum_{i=0}^{q} \frac{1}{i!} [(x - x_0) \cdot \nabla]^i f(x_0)$$

$$R_f = \frac{1}{(q+1)!} [(x - x_0) \cdot \nabla]^{q+1} F [x_0 + (x - x_0)\zeta]$$

where,

$$x_0 \in X; \quad \zeta \in [0, 1]$$

$$[g \cdot \nabla]^k = \sum_{j_1 + \cdots + j_m = k} \frac{k!}{j_1! \cdots j_m!} g_1^{j_1} \cdots g_m^{j_m} \frac{\partial^k}{\partial x_1^{j_1} \cdots \partial x_m^{j_m}}$$

- Can also compute Taylor models by using Taylor model operations (Makino and Berz, 1996)
Taylor Model Operations

- Let $T_f$ and $T_g$ be the Taylor models of the functions $f(x)$ and $g(x)$, respectively, over the interval $x \in X$.

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

- Multiplication: $T_{f \times g} = (p_{f \times g}, R_{f \times g})$ with $p_{f \times g} = p_f \times p_g - p_e$ and $R_{f \times g} = B(p_e) + B(p_f) \times R_g + B(p_g) \times R_f + R_f \times R_g$

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

- Reciprocal operation and intrinsic functions can also be defined.

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

- Beginning with Taylor models of simple functions, Taylor models of very complicated functions can be computed.

- Compared to other rigorous bounding methods (e.g., interval arithmetic), Taylor models often yield sharper bounds for modest to complicated functional dependencies (Makino and Berz, 1999).
Taylor Models – Range Bounding

- Exact range bounding of the interval polynomials – NP hard
- Direct evaluation of the interval polynomials – overestimation
- Focus on bounding the dominant part (1st and 2nd order terms)
- Exact range bounding of a general interval quadratic – also worst-case exponential complexity
- A compromise approach – Exact bounding of 1st order and diagonal 2nd order terms

\[
B(p) = \sum_{i=1}^{m} \left[ a_i (X_i - x_{i0})^2 + b_i (X_i - x_{i0}) \right] + S
\]

\[
= \sum_{i=1}^{m} \left[ a_i \left( X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} \right] + S,
\]

where \( S \) is the interval bound of other terms by direct evaluation
Taylor Models – Constraint Propagation

- Consider constraint $c(x) \leq 0, \ x \in \mathbf{X}$. Goal – Eliminate parts of $\mathbf{X}$ in which constraint cannot be satisfied

- For each $i = 1, 2 \ldots, m$, shrink $X_i$ using:

\[
B(T_c) = B(p_c) + R_c = a_i \left( X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} + S_i \leq 0
\]

\[
\implies a_iU_i^2 \leq V_i, \quad \text{with } U_i = X_i - x_{i0} + \frac{b_i}{2a_i} \quad \text{and } V_i = \frac{b_i^2}{4a_i} - S_i
\]

\[
\implies U_i = \begin{cases} 
\emptyset & \text{if } a_i > 0 \text{ and } V_i < 0 \\
[-\sqrt{\frac{V_i}{a_i}}, \sqrt{\frac{V_i}{a_i}}] & \text{if } a_i > 0 \text{ and } V_i \geq 0 \\
(-\infty, \infty] & \text{if } a_i < 0 \text{ and } V_i \geq 0 \\
[-\infty, -\sqrt{\frac{V_i}{a_i}}] \cup \left[ \sqrt{\frac{V_i}{a_i}}, \infty \right] & \text{if } a_i < 0 \text{ and } V_i < 0
\end{cases}
\]

\[
\implies X_i = X_i \cap \left( U_i + x_{i0} - \frac{b_i}{2a_i} \right)
\]
Validated Solution of Parametric ODE Systems

- Consider the parametric ODE system

\[ \dot{x} = f(x, \theta) \]

\[ x(t_0) = x_0 \in X_0 \]

\[ \theta \in \Theta \]

- Validated methods:
  - Guarantee there exists a unique solution \( x(t) \) in \([t_0, t_f]\), for each \( \theta \in \Theta \) and \( x_0 \in X_0 \)
  - Compute an interval \( X_j \) that encloses all solutions of the ODE system at \( t_j \) for \( \theta \in \Theta \) and \( x_0 \in X_0 \)

- Tools are available – VNODE, COSY VI, AWA, etc.
  - May need to treat parameters as additional state variables with zero derivative

- New tool – VSPODE (Lin and Stadtherr, 2006): Deals directly with interval-valued parameters (and also interval-valued initial states)
New Method for Parametric ODEs

- Use interval Taylor series to represent dependence on time. Use Taylor models to represent dependence on uncertain quantities (parameters and initial states).

- Assuming $X_j$ is known, then
  - Phase 1: Same as “standard” approach (e.g., VNODE). Compute a coarse enclosure $\tilde{X}_j$ and prove existence and uniqueness. Use fixed point iteration with Picard operator using high-order interval Taylor series.
  - Phase 2: Refine the coarse enclosure to obtain $X_{j+1}$. Use Taylor models in terms of the uncertain parameters and initial states.

- Implemented in VSPODE (Validating Solver for Parametric ODEs) (Lin and Stadtherr, 2006).
Method for Phase 2

- Represent uncertain initial states and parameters using Taylor models $T_{x_0}$ and $T_{\theta}$, with components

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

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

- Bound the interval Taylor series coefficients $f_i$ by computing Taylor models $T_{f_i}$.
  - Use mean value theorem.
  - Evaluate using Taylor model operations.

- Reduce “wrapping effect” by using a new type of Taylor model involving a parallelepiped remainder bound.

- This results in a Taylor model $T_{x_{j+1}}$ in terms of the initial states $x_0$ and parameters $\theta$.

- Compute the enclosure $X_{j+1} = B(T_{x_{j+1}})$ by bounding over $X_0$ and $\Theta$. 
VSPODE Example

- Lotka-Volterra Problem
  \[
  \begin{align*}
  \dot{x}_1 &= \theta_1 x_1 (1 - x_2) \\
  \dot{x}_2 &= \theta_2 x_2 (x_1 - 1) \\
  x_0 &= (1.2, 1.1)^T \\
  \theta_1 &\in [2.99, 3.01] \\
  \theta_2 &\in [0.99, 1.01]
  \end{align*}
  \]

- Integrate from \( t_0 = 0 \) to \( t_N = 10 \).

- VSPODE run using \( q = 5 \) (order of Taylor model), \( k = 17 \) (order of interval Taylor series) and QR for wrapping.

- For comparison, VNODE was used, with interval parameters treated as additional state variables, and run using \( k = 17 \) order interval Hermite-Obreschkoff and QR for wrapping.

- Constant step size of \( h = 0.1 \) used in both VSPODE and VNODE (step size may be reduced if necessary in Phase 1).
Lotka-Volterra Problem

(Eventual breakdown of VSPODE at $t = 31.8$)
Summary of Global Optimization Algorithm

Beginning with initial parameter interval $\Theta^{(0)}$

- Establish $\hat{\phi}$, the upper bound on global minimum using $p^2$ local minimizations, where $p$ is the number of parameters (decision variables)

- Iterate: for subinterval $\Theta^{(k)}$
  1. Compute Taylor models of the states using VSPODE, and then obtain $T_\phi$
  2. Perform constraint propagation using $T_\phi \leq \hat{\phi}$ to reduce $\Theta^{(k)}$
  3. If $\Theta^{(k)} = \emptyset$, go to next subinterval
  4. If $(\hat{\phi} - B(T_\phi))/|\hat{\phi}| \leq \epsilon^{\text{rel}}$, or $\hat{\phi} - B(T_\phi) \leq \epsilon^{\text{abs}}$, discard $\Theta^{(k)}$ and go to next subinterval
  5. If $B(T_\phi) < \hat{\phi}$, update $\hat{\phi}$ with local minimization, go to step 2
  6. If $\Theta^{(k)}$ is sufficiently reduced, go to step 1
  7. Otherwise, bisect $\Theta^{(k)}$ and go to next subinterval

- This finds an $\epsilon$-global optimum with either a relative tolerance ($\epsilon^{\text{rel}}$) or absolute tolerance ($\epsilon^{\text{abs}}$)
Global Optimization Algorithm (cont’d)

- By incorporating an interval-Newton approach, this can also be implemented as an exact algorithm ($\epsilon = 0$).

- This requires the validated integration of the first- and second-order sensitivity equations. VSPODE was used.

- Interval-Newton steps are applied only after reaching an appropriate depth in the branch-and-bound tree.

- We have implemented the exact algorithm only for the case of parameter estimation problems with least squares objective.

- The exact algorithm may require more or less computation time than the $\epsilon$-global algorithm.
Computational Studies – Example 1

- Parameter estimation – Catalytic cracking of gas oil (Tjoa and Biegler 1991)

\[ A \xrightarrow{\theta_1} Q \]
\[ \xleftarrow{\theta_3} S \xrightarrow{\theta_2} \]

- The problem is:

\[
\begin{align*}
\min_{\theta} \phi &= \sum_{\mu=1}^{20} \sum_{i=1}^{2} (\hat{x}_{\mu,i} - x_{\mu,i})^2 \\
\text{s.t.} & \quad \dot{x}_1 = - (\theta_1 + \theta_3)x_1^2 \\
& \quad \dot{x}_2 = \theta_1 x_1^2 - \theta_2 x_2 \\
& \quad x_0 = (1, 0)^T; \quad x_{\mu} = x(t_{\mu}) \\
& \quad \theta \in [0, 20] \times [0, 20] \times [0, 20]
\end{align*}
\]

where \( \hat{x}_{\mu} \) is given (experimental data).
Example 1 (Cont’d)

- Solution: $\theta^* = (12.2139, 7.9798, 2.2217)^T$ and $\phi^* = 2.6557 \times 10^{-3}$

- Comparisons:

<table>
<thead>
<tr>
<th>Method</th>
<th>CPU time (s)</th>
<th>Reported</th>
<th>Adjusted*</th>
</tr>
</thead>
<tbody>
<tr>
<td>This work (exact global optimum: $\epsilon = 0$) (Intel P4 3.2GHz)</td>
<td>11.5</td>
<td>11.5</td>
<td></td>
</tr>
<tr>
<td>This work ($\epsilon_{rel} = 10^{-3}$) (Intel P4 3.2GHz)</td>
<td>14.3</td>
<td>14.3</td>
<td></td>
</tr>
<tr>
<td>Papamichail and Adjiman (2002) ($\epsilon_{rel} = 10^{-3}$) (SUN UltraSPARC-II 360MHz)</td>
<td>35478</td>
<td>4541</td>
<td></td>
</tr>
<tr>
<td>Chachuat and Latifi (2003) ($\epsilon_{rel} = 10^{-3}$) (Machine not reported)</td>
<td>10400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Singer and Barton (2006) ($\epsilon_{abs} = 10^{-3}$) (AMD Athlon 2000XP+ 1.667GHz)</td>
<td>5.78</td>
<td>2.89</td>
<td></td>
</tr>
</tbody>
</table>

*Adjusted = Approximate CPU time after adjustment for machine used (based on SPEC benchmark)
Computational Studies – Example 2

- Singular control problem (Luus, 1990)

- The original problem is stated as:

\[
\min_{\theta(t)} \phi = \int_{t_0}^{t_f} \left[ x_1^2 + x_2^2 + 0.0005(x_2 + 16t - 8 - 0.1x_3\theta^2)^2 \right] dt
\]

s.t. \[\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -x_3\theta + 16t - 8 \\
\dot{x}_3 &= \theta \\
x_0 &= (0, -1, -\sqrt{5})^T \\
t &\in [t_0, t_f] = [0, 1] \\
\theta &\in [-4, 10]
\end{align*}\]

- \(\theta(t)\) is parameterized as a piecewise constant control profile with \(n\) pieces (equal time intervals)
Example 2 (Cont’d)

- Reformulate problem to be autonomous, and introduce quadrature variable:

\[
\min_{\theta(t)} \phi = x_5(t_f)
\]

\[s.t.\]
\[
\dot{x}_1 = x_2 \\
\dot{x}_2 = -x_3\theta + 16x_4 - 8 \\
\dot{x}_3 = \theta \\
\dot{x}_4 = 1 \\
x_5 = x_1^2 + x_2^2 + 0.0005(x_2 + 16x_4 - 8 - 0.1x_3\theta^2)^2
\]

\[
x_0 = (0, -1, -\sqrt{5}, 0, 0)^T
\]

\[
t \in [t_0, t_f] = [0, 1]
\]

\[
\theta \in [-4, 10]
\]

- Do \(\epsilon\)-global optimization with \(\epsilon^{abs} = 10^{-3}\)
Example 2 (Cont’d)

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\phi^*$</th>
<th>$\theta^*$</th>
<th>CPU time (s)</th>
<th>This work</th>
<th>Singer and Barton (2006)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4965</td>
<td>(4.071)</td>
<td></td>
<td>0.02</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>0.2771</td>
<td>(5.575, -4.000)</td>
<td></td>
<td>0.32</td>
<td>11.3</td>
</tr>
<tr>
<td>3</td>
<td>0.1475</td>
<td>(8.001, -1.944, 6.042)</td>
<td>10.88</td>
<td></td>
<td>270.3</td>
</tr>
<tr>
<td>4</td>
<td>0.1237</td>
<td>(9.789, -1.200, 1.257, 6.256)</td>
<td>369.0</td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>0.1236</td>
<td>(10.00, 1.494, -0.814, 3.354, 6.151)</td>
<td>8580.6</td>
<td></td>
<td>—</td>
</tr>
</tbody>
</table>

*Approximate CPU time after adjustment for machine used
Computational Studies – Example 3

• Oil shale pyrolysis problem (Luus, 1990)

• The original problem is stated as:

\[
\begin{align*}
\min_{\theta(t)} & \quad \phi = -x_2(t_f) \\
\text{s.t.} & \quad \dot{x}_1 = -k_1 x_1 - (k_3 + k_4 + k_5) x_1 x_2 \\
& \quad \dot{x}_2 = k_1 x_1 - k_2 x_2 + k_3 x_1 x_2 \\
& \quad k_i = a_i \exp \left( \frac{-b_i}{R \theta} \right), i = 1, \ldots, 5 \\
& \quad x_0 = (1, 0)^T \\
& \quad t \in [t_0, t_f] = [0, 10] \\
& \quad \theta \in [698.15, 748.15].
\end{align*}
\]

• \( \theta(t) \) is paramaterized as a piecewise constant control profile with \( n \) pieces (equal time intervals)
Example 3 (Cont’d)

- Transformation of control variable

\[ \bar{\theta} = \frac{698.15}{\theta} \]

- The reformulated problem is stated as:

\[
\begin{align*}
\min_{\theta(t)} & \quad \phi = -x_2(t_f) \\
\text{s.t.} & \quad \dot{x}_1 = -k_1 x_1 - (k_3 + k_4 + k_5) x_1 x_2 \\
& \quad \dot{x}_2 = k_1 x_1 - k_2 x_2 + k_3 x_1 x_2 \\
& \quad k_i = a_i \exp(-\bar{\theta} b_i / R), \ i = 1, \ldots, 5 \\
& \quad x_0 = (1, 0)^T \\
& \quad t \in [t_0, t_f] = [0, 10] \\
& \quad \bar{\theta} \in [698.15/748.15, 1].
\end{align*}
\]

- Do \( \epsilon \)-global optimization with \( \epsilon^{\text{abs}} = 10^{-3} \)
## Example 3 (Cont’d)

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\phi^*$</th>
<th>$\bar{\theta}^*$</th>
<th>CPU time (s)</th>
<th>Singer and Barton (2006)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.3479</td>
<td>(0.984)</td>
<td>3.2</td>
<td>13.1</td>
</tr>
<tr>
<td>2</td>
<td>-0.3510</td>
<td>(0.970, 1.000)</td>
<td>26.8</td>
<td>798.7</td>
</tr>
<tr>
<td>3</td>
<td>-0.3517</td>
<td>(1.000, 0.963, 1.000)</td>
<td>251.6</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-0.3523</td>
<td>(1.000, 0.955, 1.000, 1.000)</td>
<td>2443.5</td>
<td>-</td>
</tr>
</tbody>
</table>

*Approximate CPU time after adjustment for machine used*
Concluding Remarks

- An approach has been described for deterministic global optimization of dynamic systems using interval analysis and Taylor models
  - A validated solver for parametric ODEs is used to construct bounds on the states of dynamic systems
  - An efficient constraint propagation procedure is used to reduce the incompatible parameter domain

- Can be combined with the interval-Newton method (Lin and Stadtherr, 2006)
  - True global optimum instead of $\epsilon$-convergence
  - May or may not reduce CPU time required
Acknowledgments

- Indiana 21st Century Research & Technology Fund
- U. S. Department of Energy