UCONN
UNIVERSITY OF CONNECTICUT

Global Optimization of Stiff Dynamical Systems

ME Wilhelm, AV Lee, **MD Stuber**

AIChE Annual Meeting 2019 – Orlando, FL
November 12, 2019

19 AIChE Annual Meeting, Orlando, FL

Process Systems and Operations Research Laboratory

- DOI 10.1002/aic.16836

Matthew Wilhelm
PhD Candidate
PSOR Lab
UConn

Anne Le
PhD Student
UMass Amherst

AIChE JOURNAL
An Official Publication of the American Institute of Chemical Engineers
AIChE

FUTURES ISSUE: PROCESS SYSTEMS ENGINEERING

Global Optimization of Stiff Dynamical Systems

Matthew E. Wilhelm, Anne V. Le, Matthew D. Stuber

First published: 16 October 2019 | https://doi.org/10.1002/aic.16836

This article has been accepted for publication and undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi: 10.1002/aic.16836.

PDF TOOLS SHARE

Abstract

We present a deterministic global optimization method for nonlinear programming formulations constrained by stiff systems of ordinary differential equation (ODE) initial value problems (IVPs). The examples arise from dynamic optimization problems exhibiting both fast and slow transient phenomena commonly encountered in model-based systems engineering applications. The proposed approach utilizes unconditionally-stable implicit integration methods to reformulate the ODE-constrained problem into a nonconvex nonlinear program (NLP) with implicit functions embedded. This problem is then solved to global optimality in finite time using a spatial B&B framework utilizing convex/concave relaxations of implicit functions constructed by a method which fully exploits problem sparsity. The algorithms were implemented in the Julia programming language within the EAGO.jl package and demonstrated on five illustrative examples with varying complexity relevant in process systems engineering. The developed methods enable the guaranteed global solution of dynamic optimization problems with stiff ODE-IVPs embedded.

Introduction

Want to solve dynamic optimization problems to guaranteed global optimality:

$$
\begin{aligned}
\phi^* = & \min_{\mathbf{p} \in P \subset \mathbb{R}^{n_p}} \phi(\mathbf{x}(\mathbf{p}, t_f), \mathbf{p}) \\
\text{s.t.} \ & \dot{\mathbf{x}}(\mathbf{p}, t) = \mathbf{f}(\mathbf{x}(\mathbf{p}, t), \mathbf{p}, t), \ \forall t \in I = [t_0, t_f] \\
& \mathbf{x}(\mathbf{p}, t_0) = \mathbf{x}_0(\mathbf{p}) \\
& \mathbf{g}(\mathbf{x}(\mathbf{p}, t_f), \mathbf{p}) \leq \mathbf{0}
\end{aligned}
$$

Introduction

Want to solve dynamic optimization problems to guaranteed global optimality:

$$\phi^* = \min_{\mathbf{p} \in P \subset \mathbb{R}^{n_p}} \phi(\mathbf{x}(\mathbf{p}, t_f), \mathbf{p})$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(\mathbf{p}, t) = \mathbf{f}(\mathbf{x}(\mathbf{p}, t), \mathbf{p}, t), \ \forall t \in I = [t_0, t_f]$$

$$\mathbf{x}(\mathbf{p}, t_0) = \mathbf{x}_0(\mathbf{p})$$

$$\mathbf{g}(\mathbf{x}(\mathbf{p}, t_f), \mathbf{p}) \leq \mathbf{0}$$

Parametric ordinary differential equation initial value problem (ODE-IVP) constraints.

Arise from optimal control, parameter estimation, etc.

Introduction

Why do we need guaranteed global optimality?

Introduction

Why do we need guaranteed global optimality?

- Most often when we’re dealing with uncertainty (and nonconvexity):
  - Robust control: given uncertainty are mitigating control actions stable?
  - Robust design: will our system operate safely in the presence of uncertainty?
  - Model validation: does a proposed model capture the observed behavior?

Introduction

Why do we need guaranteed global optimality?

- Most o
  - Ro ... stable?
  - Ro ... f
    und
  - Mo
    beh

Introduction

Why do we need guaranteed global optimality?

- Most often when we’re dealing with uncertainty (and nonconvexity):
  - Robust control: given uncertainty are mitigating control actions stable?
  - Robust design: will our system operate safely in the presence of uncertainty?
  - Model validation: does a proposed model capture the observed behavior?

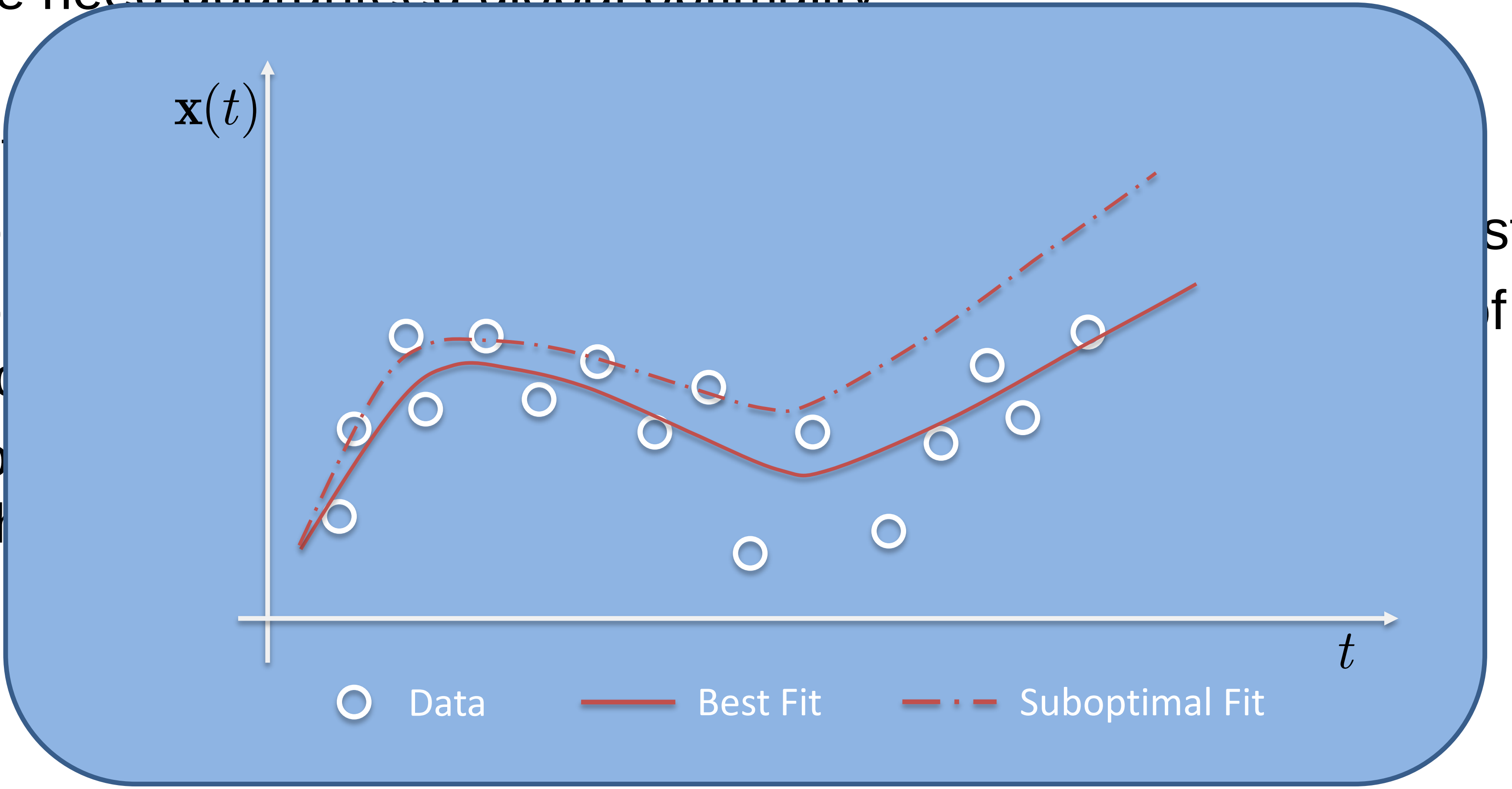
Introduction

Why do we need guaranteed global optimality?

- Most o
  - Ro
  - Ro
    und
  - Mo
    beh

stable?

f



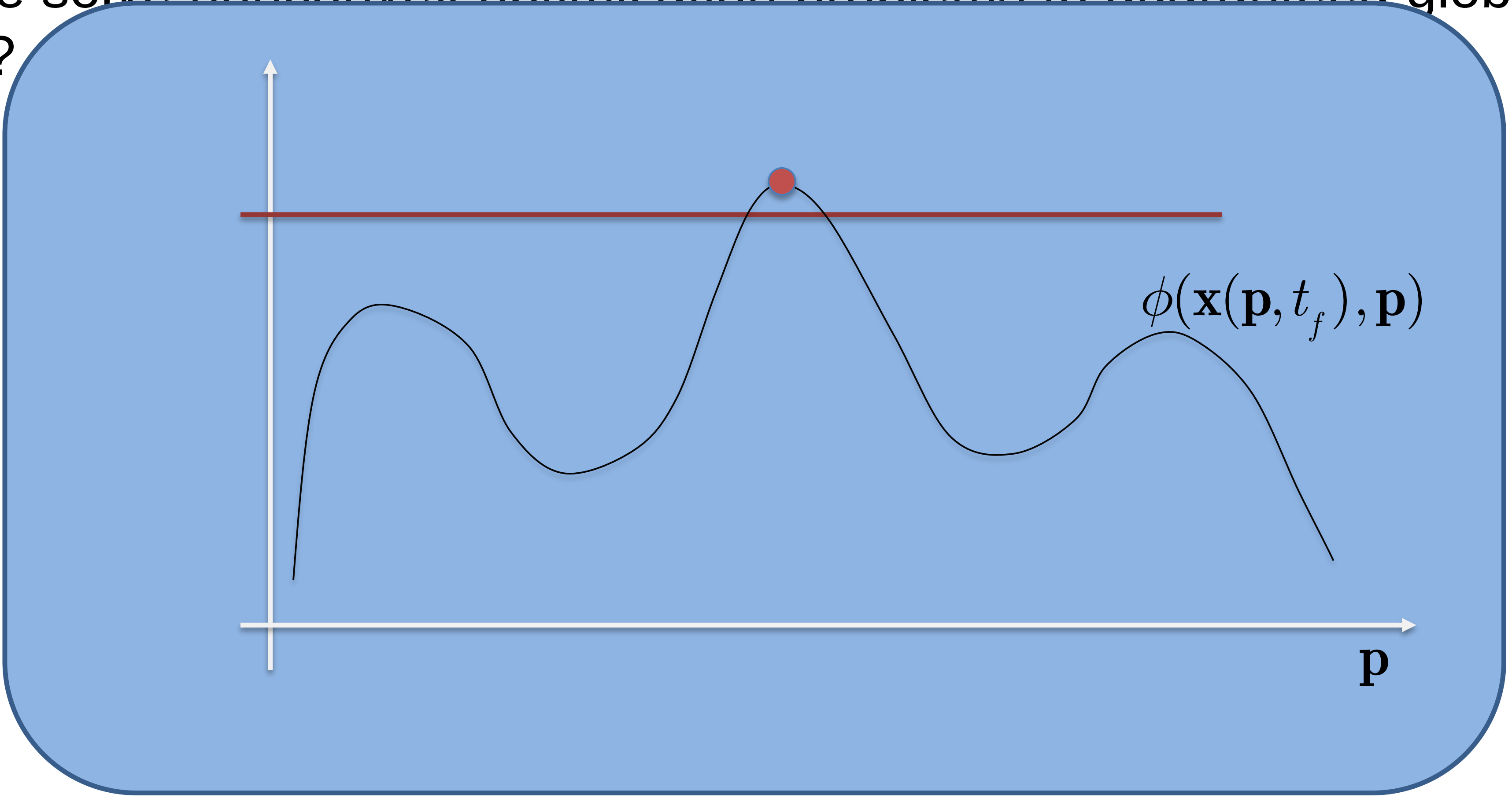
Introduction

How do we solve nonconvex optimization problems to guaranteed global optimality?

Branch-and-Bound Algorithm (and variants)

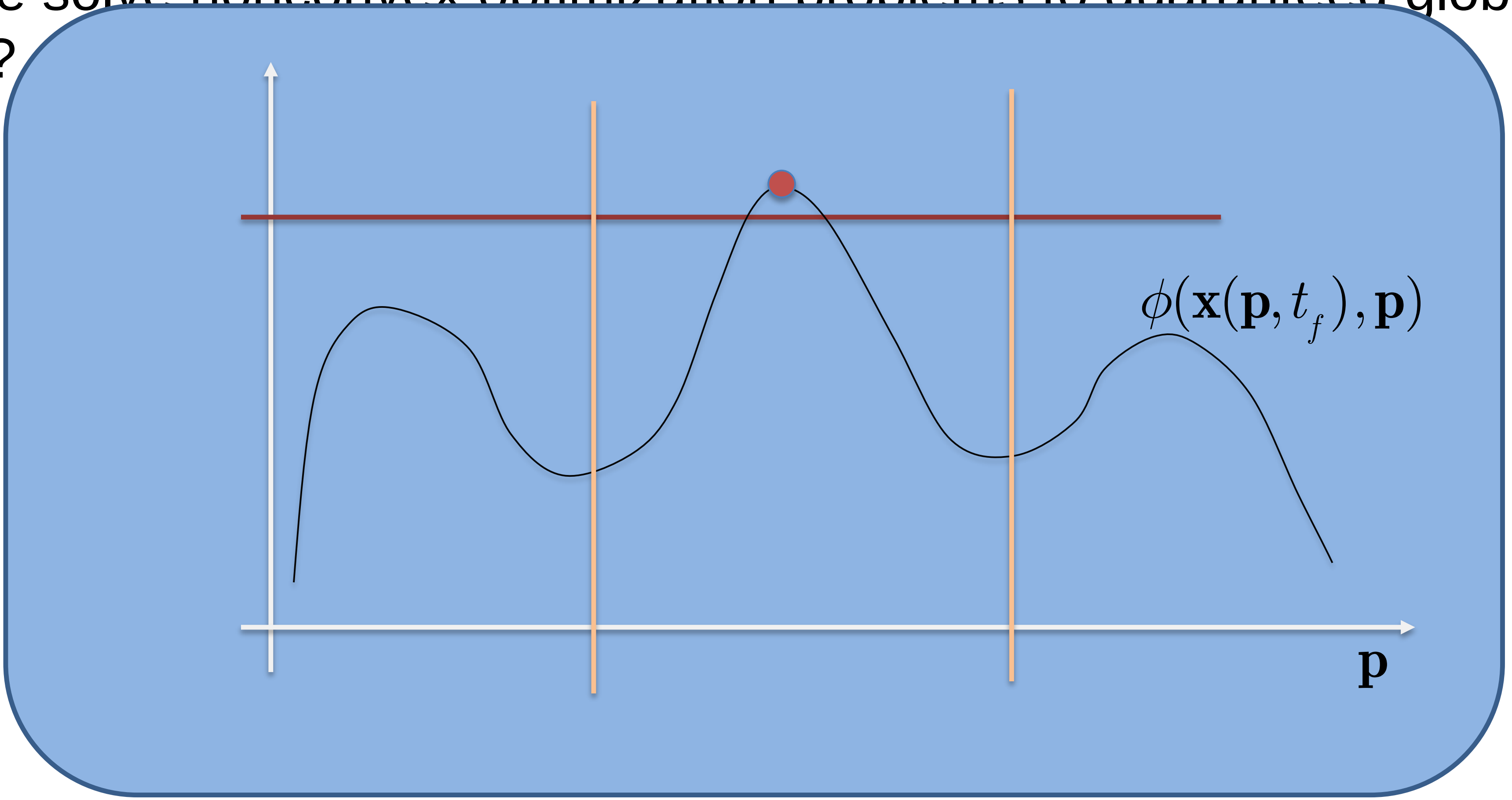
Introduction

How do we solve nonconvex optimization problems to guaranteed global optimality?



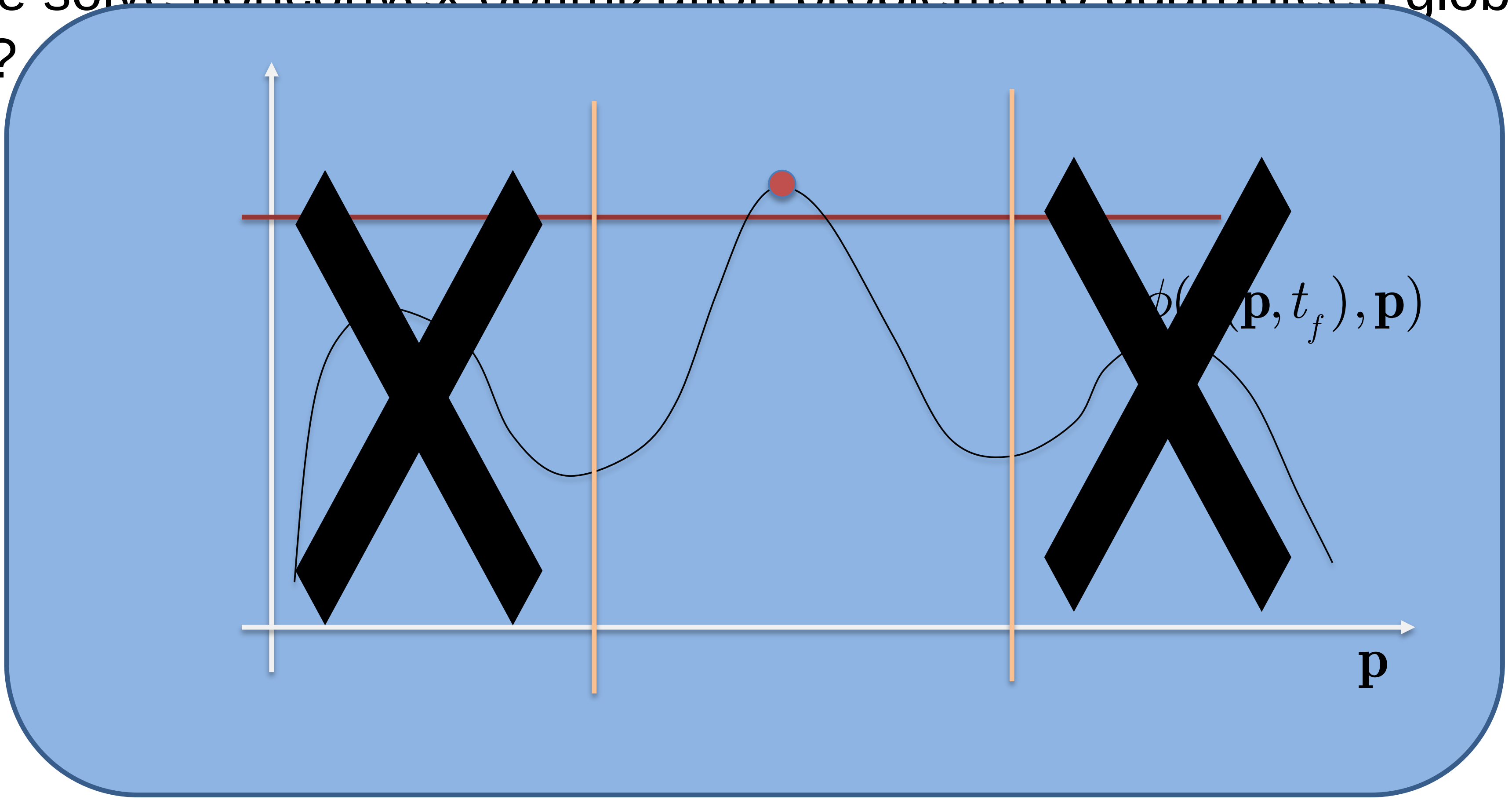
Introduction

How do we solve nonconvex optimization problems to guaranteed global optimality?



Introduction

How do we solve nonconvex optimization problems to guaranteed global optimality?



Introduction

How do we solve nonconvex optimization problems to guaranteed global optimality?

Branch-and-Bound Algorithm (and variants) is a deterministic search procedure that systematically partitions the decision space and rules out regions based on infeasibility and sub-optimality.

We want to employ spatial branch-and-bound to solve our dynamic optimization problem.

Introduction

How do we solve nonconvex optimization problems to guaranteed global optimality?

Branch-and-Bound Algorithm (and variants) is a deterministic search procedure that systematically partitions the decision space and rules out regions based on infeasibility and sub-optimality.

We want to employ spatial branch-and-bound to solve our dynamic optimization problem.

→ **must be able to calculate rigorous global bounds on all functions**

Introduction

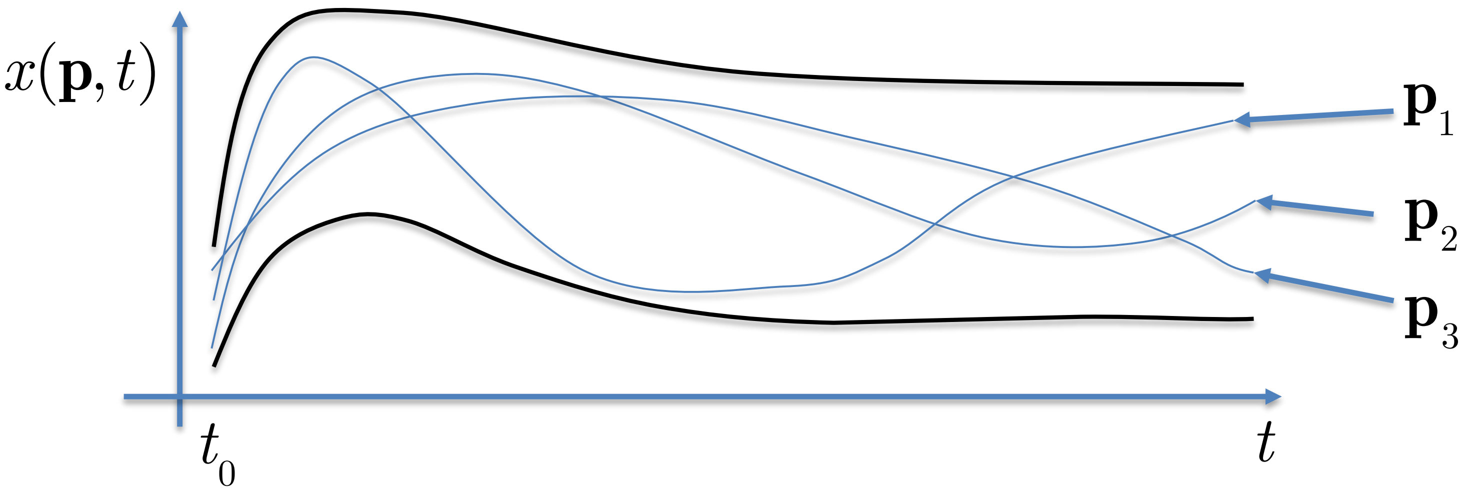
The most challenging task is then calculating rigorous bounds on the state trajectories over the entire parameter set:

$$\dot{\mathbf{x}}(\mathbf{p},t) = \mathbf{f}(\mathbf{x}(\mathbf{p},t),\mathbf{p},t), \forall t \in I = [t_0, t_f], \forall \mathbf{p} \in P$$
$$\mathbf{x}(\mathbf{p},t_0) = \mathbf{x}_0(\mathbf{p}), \forall \mathbf{p} \in P$$

Introduction

The most challenging task is then calculating rigorous bounds on the state trajectories over the entire parameter set:

$$\dot{\mathbf{x}}(\mathbf{p},t) = \mathbf{f}(\mathbf{x}(\mathbf{p},t),\mathbf{p},t),\ \forall t \in I = [t_0, t_f],\ \forall \mathbf{p} \in P$$
$$\mathbf{x}(\mathbf{p},t_0) = \mathbf{x}_0(\mathbf{p}),\ \forall \mathbf{p} \in P$$



Past Approaches

Explicit Discrete-Time

1. Express the ODE-IVP system as a discrete-time approximation using numerical integration schemes
2. Reformulate the dynamic optimization problem into a standard NLP with algebraic constraints (states become decision variables)

Biegler, L.T.: **Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes.** SIAM, Philadelphia (2010)

Past Approaches

Explicit Discrete-Time

1. Express the ODE-IVP system as a

2.

$$\dot{\mathbf{x}}(\mathbf{p},t) = \mathbf{f}(\mathbf{x}(\mathbf{p},t),\mathbf{p},t),\ \forall t \in I = [t_0, t_f]$$
$$\mathbf{x}(\mathbf{p},t_0) = \mathbf{x}_0(\mathbf{p})$$

$$\mathbf{z}_0 = \mathbf{x}_0(\mathbf{p})$$
$$\hat{\mathbf{z}}_1 - \mathbf{z}_0 - h\mathbf{f}(\hat{\mathbf{z}}_1,\mathbf{p},t_1) = \mathbf{0}$$
$$\vdots \qquad \vdots$$
$$\hat{\mathbf{z}}_K - \hat{\mathbf{z}}_{K-1} - h\mathbf{f}(\hat{\mathbf{z}}_K,\mathbf{p},t_K) = \mathbf{0}$$

Biegler, L.T.: **Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes.** SIAM, Philadelphia (2010)

Past Approaches

Explicit Discrete-Time

1. Express the ODE-IVP system as a discrete-time approximation using numerical integration schemes
2. Reformulate the dynamic optimization problem into a standard NLP with algebraic constraints (states become additional decision variables)
3. Calculate bounds of algebraic functions for the branch-and-bound algorithm

Past Approaches

Explicit Discrete-Time

1. Express the ODE-IVP system as a discrete-time approximation using numerical integration schemes
2. Reformulate the dynamic optimization problem into a standard NLP with algebraic constraints (states become additional decision variables)
3. Calculate bounds of algebraic functions for the branch-and-bound algorithm

Simulation-Based

1. Derive auxiliary system of ODE-IVPs guaranteed to bound all parametric trajectories.
2. Integrate these ODE-IVPs to provide valid bounds to the branch-and-bound algorithm (only **p** are decision variables)

Rihm, Robert. **Interval methods for initial value problems in ODEs.** *Topics in Validated Computations* (1994): 173-207.

Joseph K Scott, Paul I Barton. **Improved relaxations for the parametric solutions of ODEs using differential inequalities.** *Journal of Global Optimization*. 2013 (57): 143–176.

A.M. Sahlodin, Benoît Chachaut. **Discretize-then-relax approach for convex/concave relaxations of the solutions of parametric ODEs.** *Applied Numerical Mathematics*, 61 (179): 803 – 820, 2011

Past Approaches Downsides

Explicit Discrete-Time

- For accuracy, many discrete states are required (i.e., high-dimensionality NLP)
- No integration error control (higher-order methods are used)

Simulation-Based

- Can be slow (depending on numerical integrator and problem complexity)
- A discrete-time form was developed but only for explicit integration methods (not for stiff systems)

Our New Approach

We propose a discrete-time simulation-based approach:

1. Utilize implicit function theory to "solve" the discrete-time system of equations as implicit functions of the decision variables **p.**

$$
\begin{aligned}
\mathbf{z}_0 &= \mathbf{x}_0(\mathbf{p}) \\
\hat{\mathbf{z}}_1 - \mathbf{z}_0 - h\mathbf{f}(\hat{\mathbf{z}}_1, \mathbf{p}, t_1) &= \mathbf{0} \\
&\vdots \\
\hat{\mathbf{z}}_K - \hat{\mathbf{z}}_{K-1} - h\mathbf{f}(\hat{\mathbf{z}}_K, \mathbf{p}, t_K) &= \mathbf{0}
\end{aligned}
\quad \xrightarrow{\text{Solve}} \quad
\mathbf{z}(\mathbf{p}) = (\mathbf{z}_0(\mathbf{p}), \mathbf{z}_1(\mathbf{p}), \ldots, \mathbf{z}_K(\mathbf{p}))
$$

Our New Approach

We propose a discrete-time simulation-based approach:

1. Utilize implicit function theory to “solve” the discrete-time system of equations as implicit functions of the decision variables **p.**

$$
\begin{aligned}
\phi^* = \min_{\mathbf{p}\in P,\hat{\mathbf{z}}\in Z} & \ \phi(\hat{\mathbf{z}},\mathbf{p},t_f) \\
\text{s.t.} \ & \mathbf{z}_0 = \mathbf{x}_0(\mathbf{u},\mathbf{p}) \\
& \hat{\mathbf{z}}_1 - \mathbf{z}_0 - h\mathbf{f}(\hat{\mathbf{z}}_1,\mathbf{p},t_1) = \mathbf{0} \\
& \vdots \qquad \vdots \\
& \hat{\mathbf{z}}_K - \hat{\mathbf{z}}_{K-1} - h\mathbf{f}(\hat{\mathbf{z}}_K,\mathbf{p},t_K) = \mathbf{0} \\
& \mathbf{g}(\hat{\mathbf{z}}_K,\mathbf{p}) \le \mathbf{0}
\end{aligned}
$$

$$
\mathbf{z}(\mathbf{p}) = (\mathbf{z}_0(\mathbf{p}),\mathbf{z}_1(\mathbf{p}),\ldots,\mathbf{z}_K(\mathbf{p}))
$$

$$
\begin{aligned}
\phi^* = \min_{\mathbf{p}\in P} & \ \phi(\mathbf{z}(\mathbf{p}),\mathbf{p},t_f) \\
\text{s.t.} \ & \mathbf{g}(\mathbf{z}_K(\mathbf{p}),\mathbf{p}) \le \mathbf{0}
\end{aligned}
$$

Our New Approach

We propose a discrete-time simulation-based approach:

1. Utilize implicit function theory to "solve" the discrete-time system of equations as implicit functions of the decision variables **p.**

$$\begin{aligned}
\phi^* = \min_{\mathbf{p}\in P,\hat{\mathbf{z}}\in Z} & \ \phi(\hat{\mathbf{z}},\mathbf{p},t_f) \\
\text{s.t.}\ & \mathbf{z}_0 = \mathbf{x}_0(\mathbf{u},\mathbf{p}) \\
& \hat{\mathbf{z}}_1 - \mathbf{z}_0 - h\mathbf{f}(\hat{\mathbf{z}}_1,\mathbf{p},t_1) = \mathbf{0} \\
& \vdots \qquad \vdots \\
& \hat{\mathbf{z}}_K - \hat{\mathbf{z}}_{K-1} - h\mathbf{f}(\hat{\mathbf{z}}_K,\mathbf{p},t_K) = \mathbf{0} \\
& \mathbf{g}(\hat{\mathbf{z}}_K,\mathbf{p}) \le \mathbf{0}
\end{aligned}$$

$$\mathbf{z}(\mathbf{p}) = (\mathbf{z}_0(\mathbf{p}),\mathbf{z}_1(\mathbf{p}),\ldots,\mathbf{z}_K(\mathbf{p}))$$

This is not known analytically!

$$\begin{aligned}
\phi^* = \min_{\mathbf{p}\in P} & \ \phi(\mathbf{z}(\mathbf{p}),\mathbf{p},t_f) \\
\text{s.t.}\ & \mathbf{g}(\mathbf{z}_K(\mathbf{p}),\mathbf{p}) \le \mathbf{0}
\end{aligned}$$

Our New Approach

We propose a discrete-time simulation-based approach:

1. Utilize implicit function theory to “solve” the discrete-time system of equations as implicit functions of the decision variables **p.**
2. Utilize interval arithmetic and McCormick-based relaxations of implicit functions to rigorously bound $\mathbf{z}(\mathbf{p}) = (\mathbf{z}_0(\mathbf{p}), \mathbf{z}_1(\mathbf{p}), \ldots, \mathbf{z}_K(\mathbf{p}))$
   – Extend theory developed for steady-state systems

   Stuber, M.D., Scott, J.K., and P.I. Barton. Convex and Concave Relaxations of Implicit Functions. *Optimization Methods & Software* (2015)

Our New Approach

- Discrete-time representation is a nonlinear algebraic system that make up equality constraints (like a steady-state model)

$$\mathbf{h}(\hat{\mathbf{z}}, \mathbf{p}) = \begin{pmatrix} \mathbf{z}_0 - \mathbf{x}_0(\mathbf{p}) \\ \hat{\mathbf{z}}_1 - \mathbf{z}_0 - h\mathbf{f}(\hat{\mathbf{z}}_1, \mathbf{p}, t_1) \\ \vdots \\ \hat{\mathbf{z}}_K - \hat{\mathbf{z}}_{K-1} - h\mathbf{f}(\hat{\mathbf{z}}_K, \mathbf{p}, t_K) \end{pmatrix} = \mathbf{0}$$

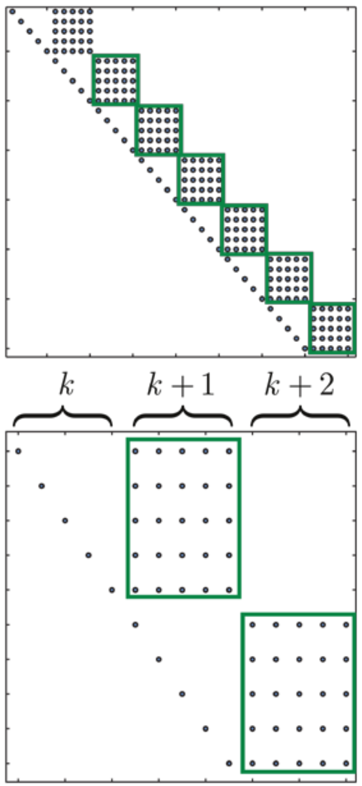
$$\mathbf{h} : \mathbb{R}^{n_x(K+1)} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x(K+1)}$$

Our New Approach

- Discrete-time representation is a nonlinear algebraic system that make up equality constraints (like a steady-state model)

$$\mathbf{h}(\hat{\mathbf{z}}, \mathbf{p}) = \begin{pmatrix} \mathbf{z}_0 - \mathbf{x}_0(\mathbf{p}) \\ \hat{\mathbf{z}}_1 - \mathbf{z}_0 - h\mathbf{f}(\hat{\mathbf{z}}_1, \mathbf{p}, t_1) \\ \vdots \\ \hat{\mathbf{z}}_K - \hat{\mathbf{z}}_{K-1} - h\mathbf{f}(\hat{\mathbf{z}}_K, \mathbf{p}, t_K) \end{pmatrix} = \mathbf{0}$$

$$\mathbf{h} : \mathbb{R}^{n_x(K+1)} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x(K+1)}$$



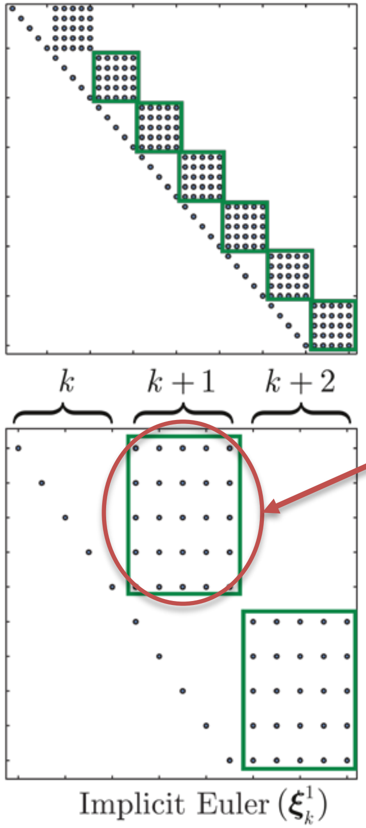
Implicit Euler ($\boldsymbol{\xi}_k^1$)

Our New Approach

- Discrete-time representation is a nonlinear algebraic system that make up equality constraints (like a steady-state model)

$$\mathbf{h}(\hat{\mathbf{z}},\mathbf{p}) = \begin{pmatrix} \mathbf{z}_0 - \mathbf{x}_0(\mathbf{p}) \\ \hat{\mathbf{z}}_1 - \mathbf{z}_0 - h\mathbf{f}(\hat{\mathbf{z}}_1,\mathbf{p},t_1) \\ \vdots \\ \hat{\mathbf{z}}_K - \hat{\mathbf{z}}_{K-1} - h\mathbf{f}(\hat{\mathbf{z}}_K,\mathbf{p},t_K) \end{pmatrix} = \mathbf{0}$$

$$\mathbf{h} : \mathbb{R}^{n_x(K+1)} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_x(K+1)}$$



$n_x \times n_x$ system of algebraic equations

Bounds on Implicit Functions

- How does this work mathematically?
  - Implicit function theorem

$$\mathbf{h}(\hat{\mathbf{z}},\mathbf{p})=\mathbf{0}\Rightarrow\hat{\mathbf{z}}=\mathbf{z}(\mathbf{p}):\mathbf{h}(\mathbf{z}(\mathbf{p}),\mathbf{p})=\mathbf{0}$$

  - (parametric) mean value theorem

$$\mathbf{M}(\mathbf{p})(\mathbf{z}(\mathbf{p})-\boldsymbol{\gamma}(\mathbf{p}))=-\mathbf{h}(\boldsymbol{\gamma}(\mathbf{p}),\mathbf{p})$$

  - Fixed-point iterations

$$\hat{\mathbf{z}}^{k+1}:=\boldsymbol{\Phi}(\hat{\mathbf{z}}^{k},\mathbf{p}),\{\hat{\mathbf{z}}^{k}\}\rightarrow\mathbf{z}(\mathbf{p})$$

  - Rigorous (global) set-valued arithmetic
    - interval arithmetic
    - generalized McCormick convex relaxations

Stuber, M.D. et al. (2015)

Our New Approach

- Parametric implicit linear multistep methods:

$$\boldsymbol{\zeta}(\hat{\mathbf{z}}_{k+s},\ldots,\hat{\mathbf{z}}_{k},\mathbf{p}) = \hat{\mathbf{z}}_{k+s} - \hat{\mathbf{z}}_{k+s-1} - h\sum_{j=0}^{s} b_j \mathbf{f}(\hat{\mathbf{z}}_{k+j},\mathbf{p},t_{k+j}) = \mathbf{0}$$

$$\boldsymbol{\xi}(\hat{\mathbf{z}}_{k+s},\ldots,\hat{\mathbf{z}}_{k},\mathbf{p}) = \hat{\mathbf{z}}_{k+s} + h\sum_{i=0}^{s-1} a_i \hat{\mathbf{z}}_{k+i} - hb_s \mathbf{f}(\hat{\mathbf{z}}_{k+s},\mathbf{p},t_{k+s}) = \mathbf{0}$$

Our New Approach

- Parametric implicit linear multistep methods:

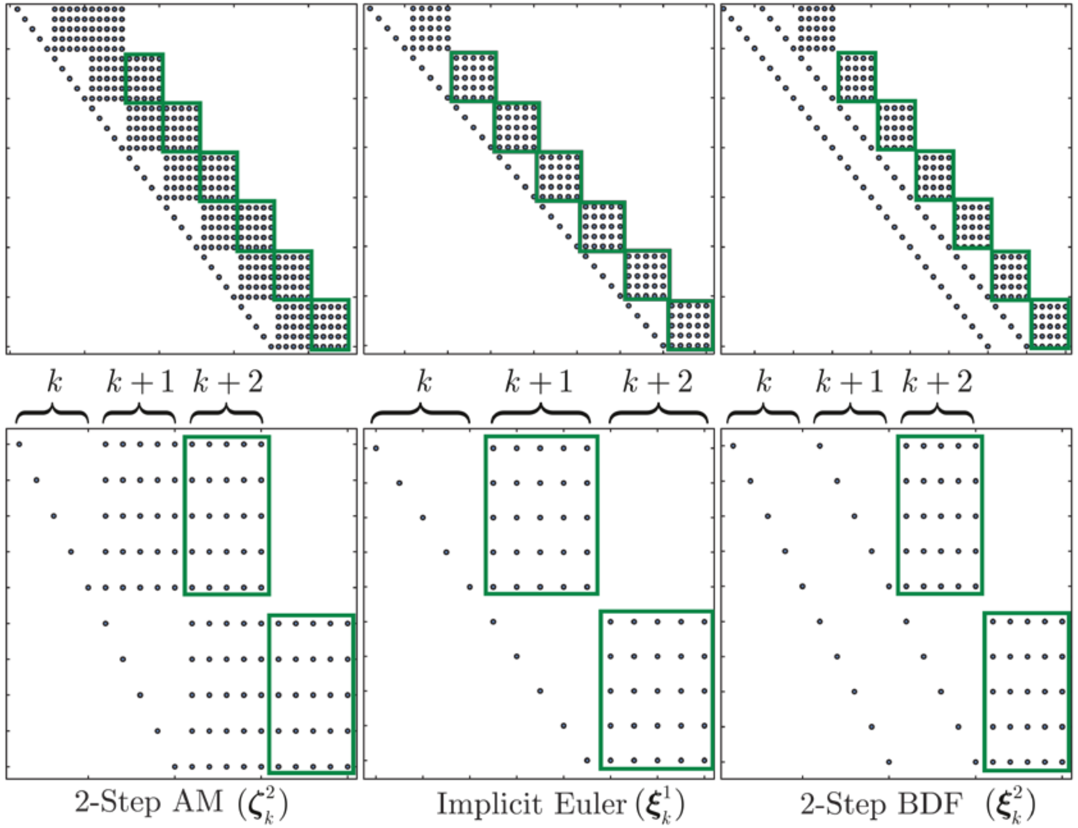
$$\boldsymbol{\zeta}(\hat{\mathbf{z}}_{k+s},\ldots,\hat{\mathbf{z}}_{k},\mathbf{p}) = \hat{\mathbf{z}}_{k+s} - \hat{\mathbf{z}}_{k+s-1} - h\sum_{j=0}^{s} b_j \mathbf{f}(\hat{\mathbf{z}}_{k+j},\mathbf{p},t_{k+j}) = \mathbf{0}$$

$$\boldsymbol{\xi}(\hat{\mathbf{z}}_{k+s},\ldots,\hat{\mathbf{z}}_{k},\mathbf{p}) = \hat{\mathbf{z}}_{k+s} + h\sum_{i=0}^{s-1} a_i \hat{\mathbf{z}}_{k+i} - h b_s \mathbf{f}(\hat{\mathbf{z}}_{k+s},\mathbf{p},t_{k+s}) = \mathbf{0}$$

- Focus on unconditionally-stable methods (*s*=1 and *s*=2)
  - Second-order methods provide an order-of-magnitude greater accuracy over first-order methods

Our New Approach

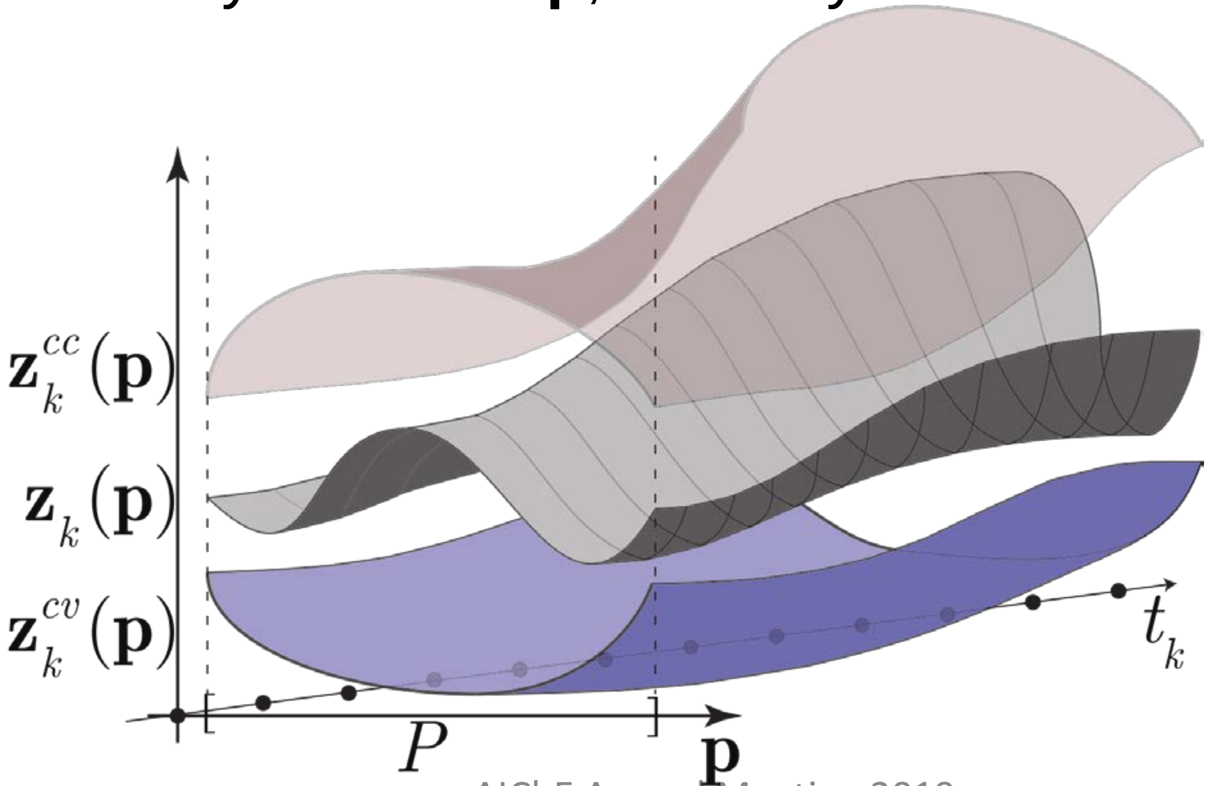
- Discrete-time representation is a nonlinear algebraic system that make up equality constraints (like a steady-state model)



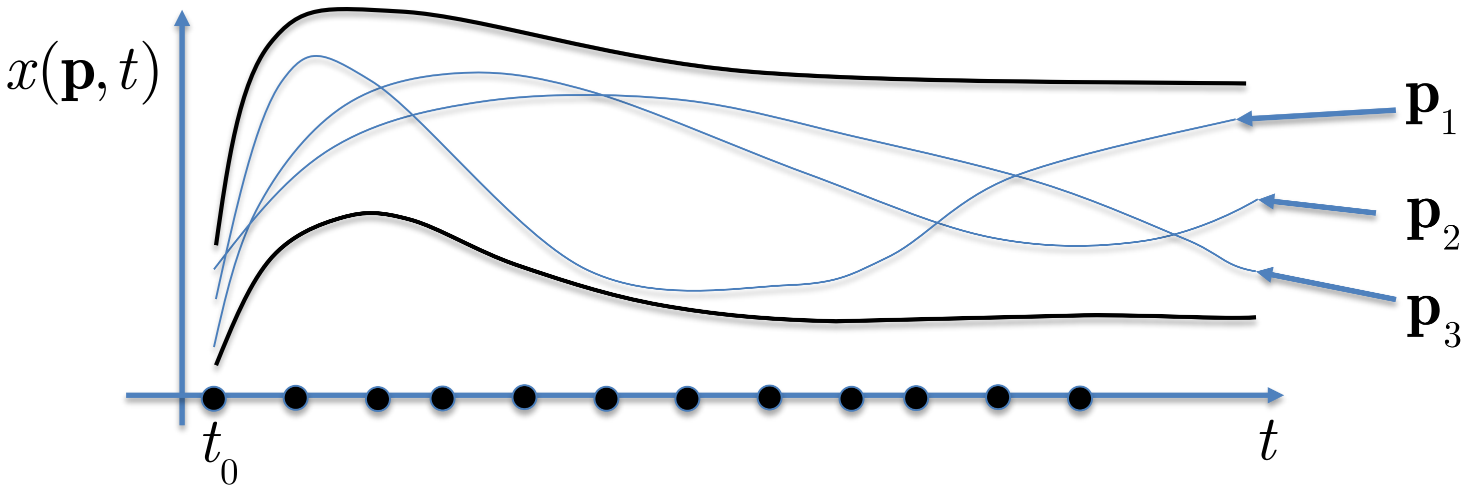
$$\mathbf{h} : \mathbb{R}^{n_x(K+1)} \times \mathbb{R}^{n_p} \longrightarrow \mathbb{R}^{n_x(K+1)}$$

Relaxations of Parametric Implicit Linear Multistep Methods

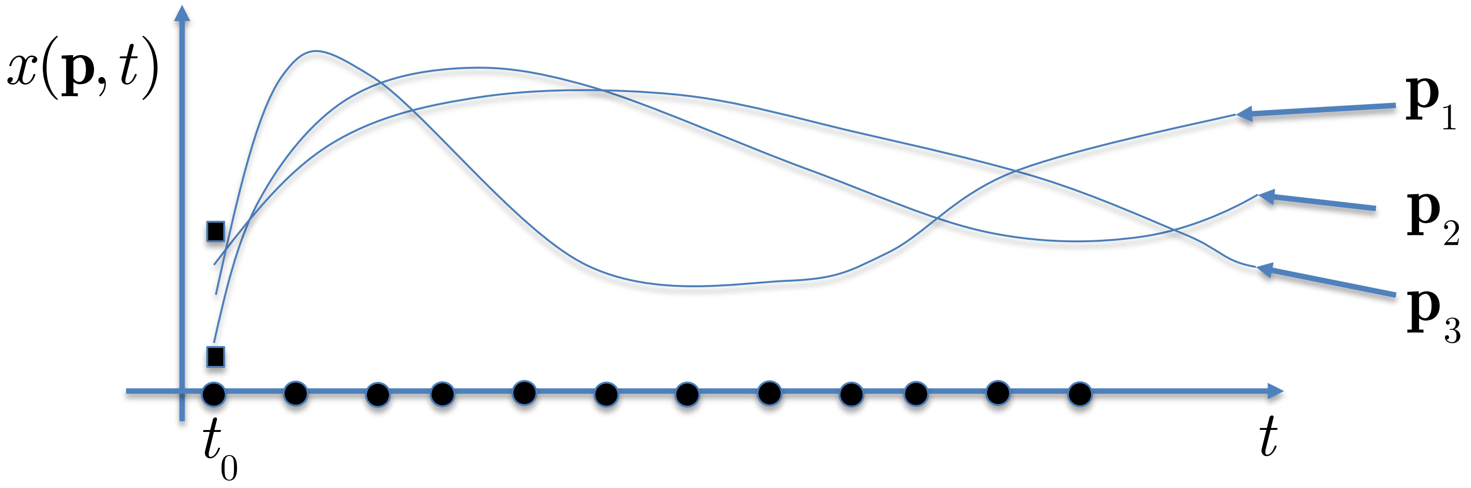
Apply the theory introduced previously for robust steady-state simulation to our system $\mathbf{h}(\hat{\mathbf{z}}, \mathbf{p}) = \mathbf{0}$ to calculate rigorous bounds on the state variables over the range of uncertainty variables **p**, block-by-block.



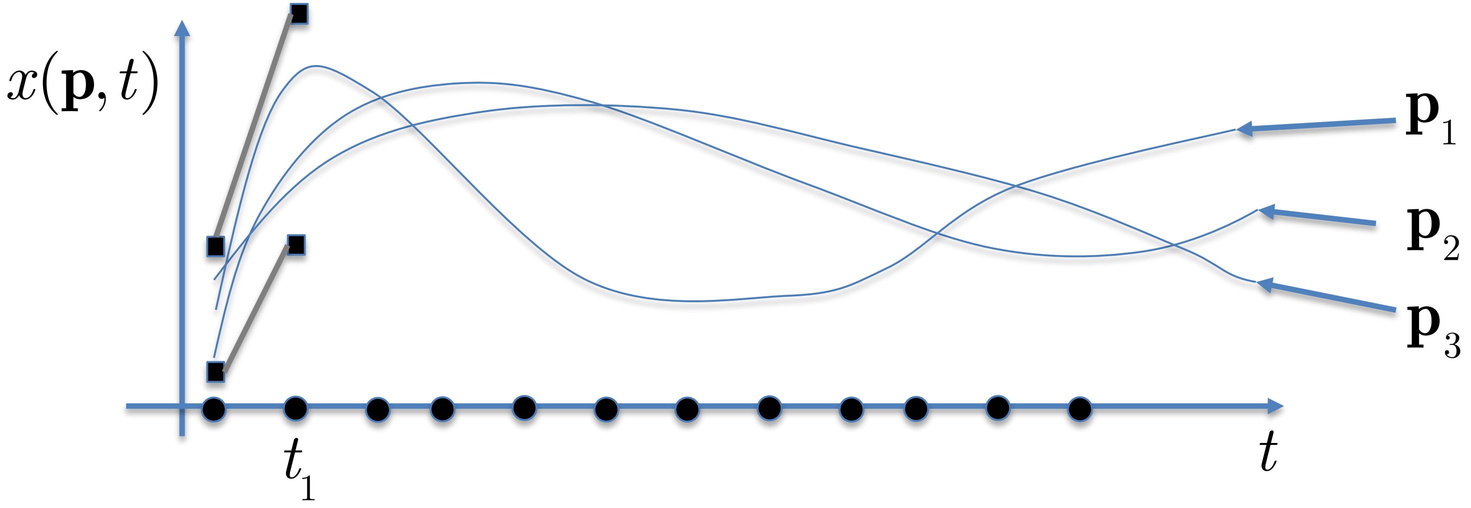
Relaxations of Parametric Implicit Linear Multistep Methods



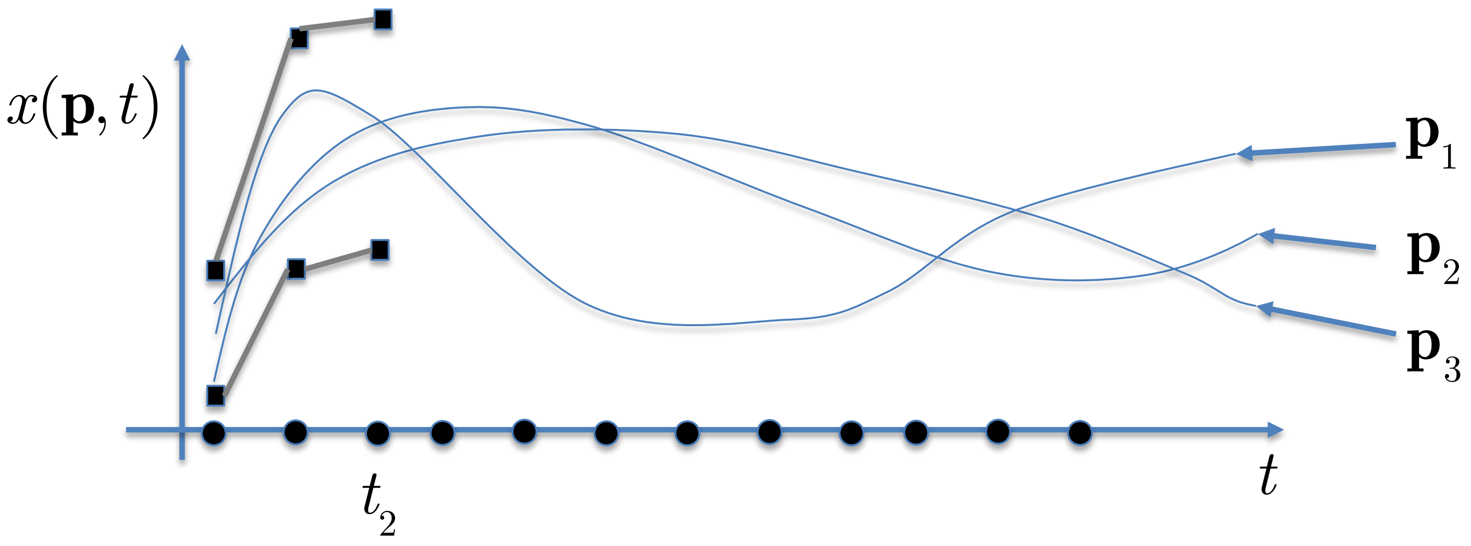
Relaxations of Parametric Implicit Linear Multistep Methods



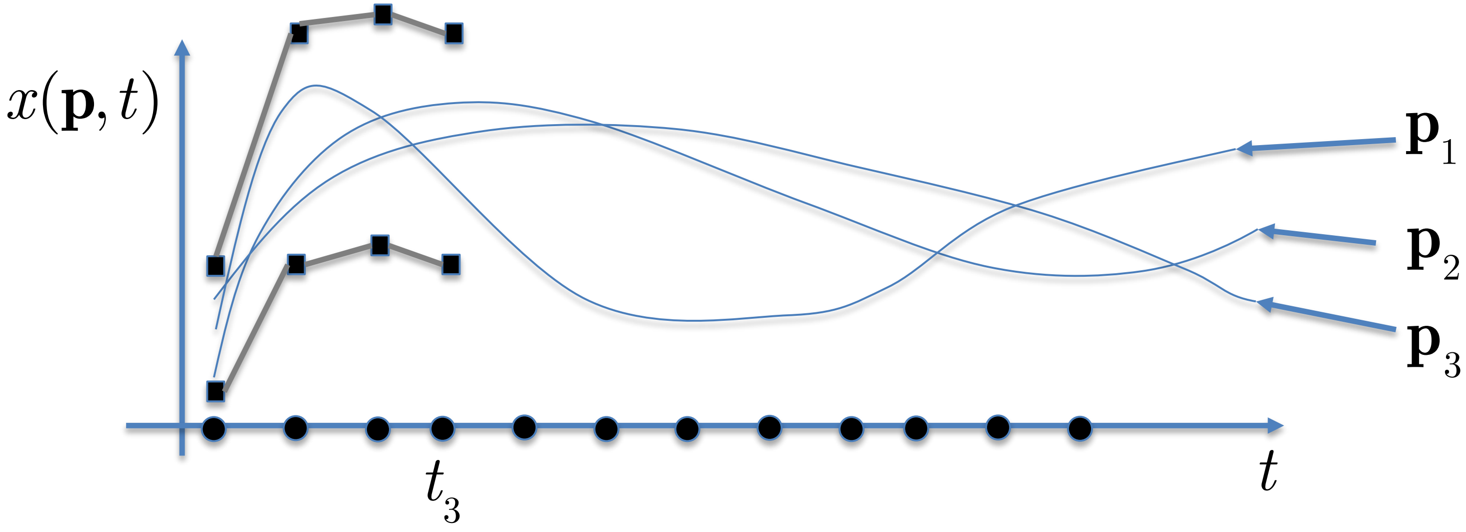
Relaxations of Parametric Implicit Linear Multistep Methods



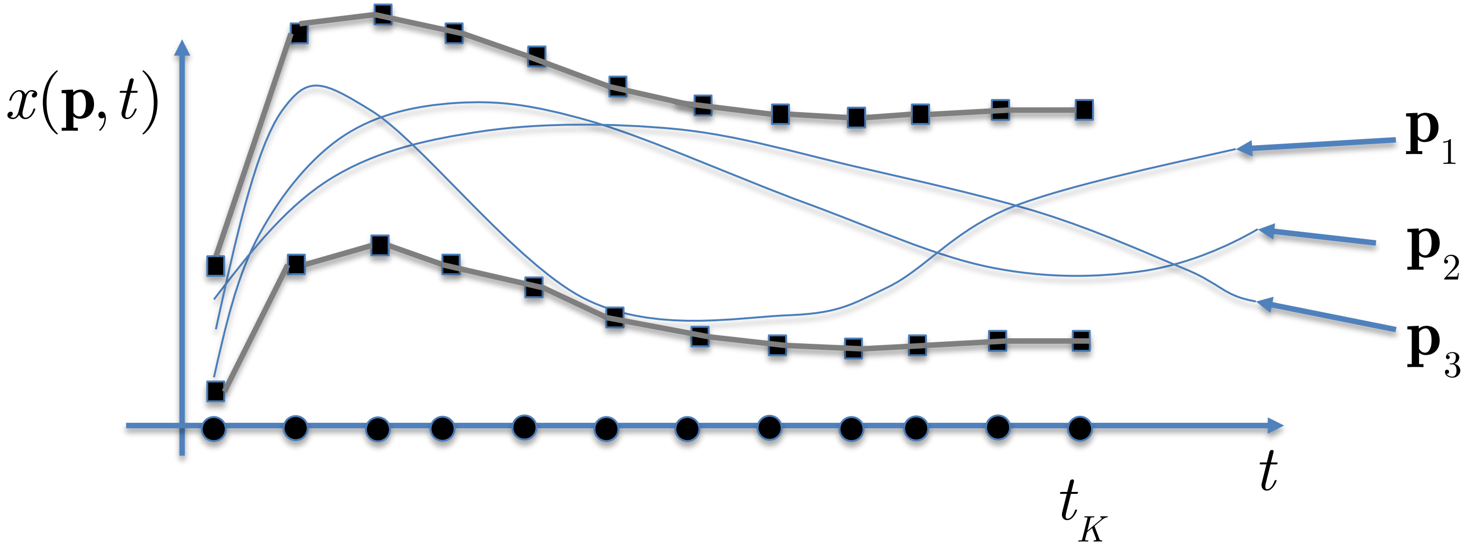
Relaxations of Parametric Implicit Linear Multistep Methods



Relaxations of Parametric Implicit Linear Multistep Methods



Relaxations of Parametric Implicit Linear Multistep Methods



Kinetic Model Example

Problem Statement

- Fit the rate constants (k_{2f}, k_{3f}, k_4) of oxygen addition to cyclohexadienyl radicals to data. [16]
- First addressed by global by Singer et al.[17]
- Explicit Euler form solved by Mitsos[18]
- Implicit Euler form addressed in Stuber[19]

ODE-IVP:

$$\dot{x}_A = k_1 x_Z x_Y - c_{O2}\left(k_{2f} + k_{3f}\right)x_A + \left(k_{2f}/K_2\right)x_D + \left(k_{3f}/K_3\right)x_B - k_5 x_A^2,$$

$$\dot{x}_B = k_{3f} c_{O2} x_A - \left(k_{3f}/K_3 + k_4\right)x_B, \qquad \dot{x}_D = k_{2f} c_{O2} x_A - \left(k_{2f}/K_2\right)x_D,$$

$$\dot{x}_Y = -k_{1s} x_Y x_Z, \qquad \dot{x}_Z = -k_1 x_Y x_Z,$$

$$\dot{\boldsymbol{x}}(t = 0) = (0, 0, 0, 0.4, 140)$$

Objective Variables

$$f^* = \min_{\boldsymbol{p}\in P} \sum_{i=1}^{n} \left(\mathrm{I}^i - \mathrm{I}^i_{\mathrm{data}}\right)^2$$

$$s.t. \quad \mathrm{I}^i = x_A^i + \frac{2}{21} x_B^i + \frac{2}{21} x_D^i$$

Decision Variables

$$\boldsymbol{p} = (k_{2f}, k_{3f}, k_4)$$

State Variables

$$\boldsymbol{x} = (x_A, x_B, x_D, x_Y, x_Z)$$

Constants

$$k_1, k_{1s}, k_5, K_2, K_3, c_{O2}, \Delta t, n$$

16. J. W. Taylor, et al. **Direct measurement of the fast, reversible addition of oxygen to cyclohexadienyl radicals in nonpolar solvents**, *Phys. Chem. A*, 108 (2004), pp. 7193–7203.
17. A. B. Singer et al., **Global dynamic optimization for parameter estimation in chemical kinetics** A. B. Singer et al., *J. Phys. Chem. A*, 110 (2006), pp. 971–976
18. Mitsos et al. **McCormick-based relaxations of algorithms.** *SIAM Journal on Optimization*, SIAM (2009) 20, 73-601
19. Stuber, M.D. et al. **Convex and concave relaxations of implicit functions.** *Optimization Methods and Software* (2015), 30, 424-460

Kinetic Model Example

Problem Statement

- Fit the rate const
  addition to cyclo
- First addressed b
- Explicit Euler for
- Implicit Euler for

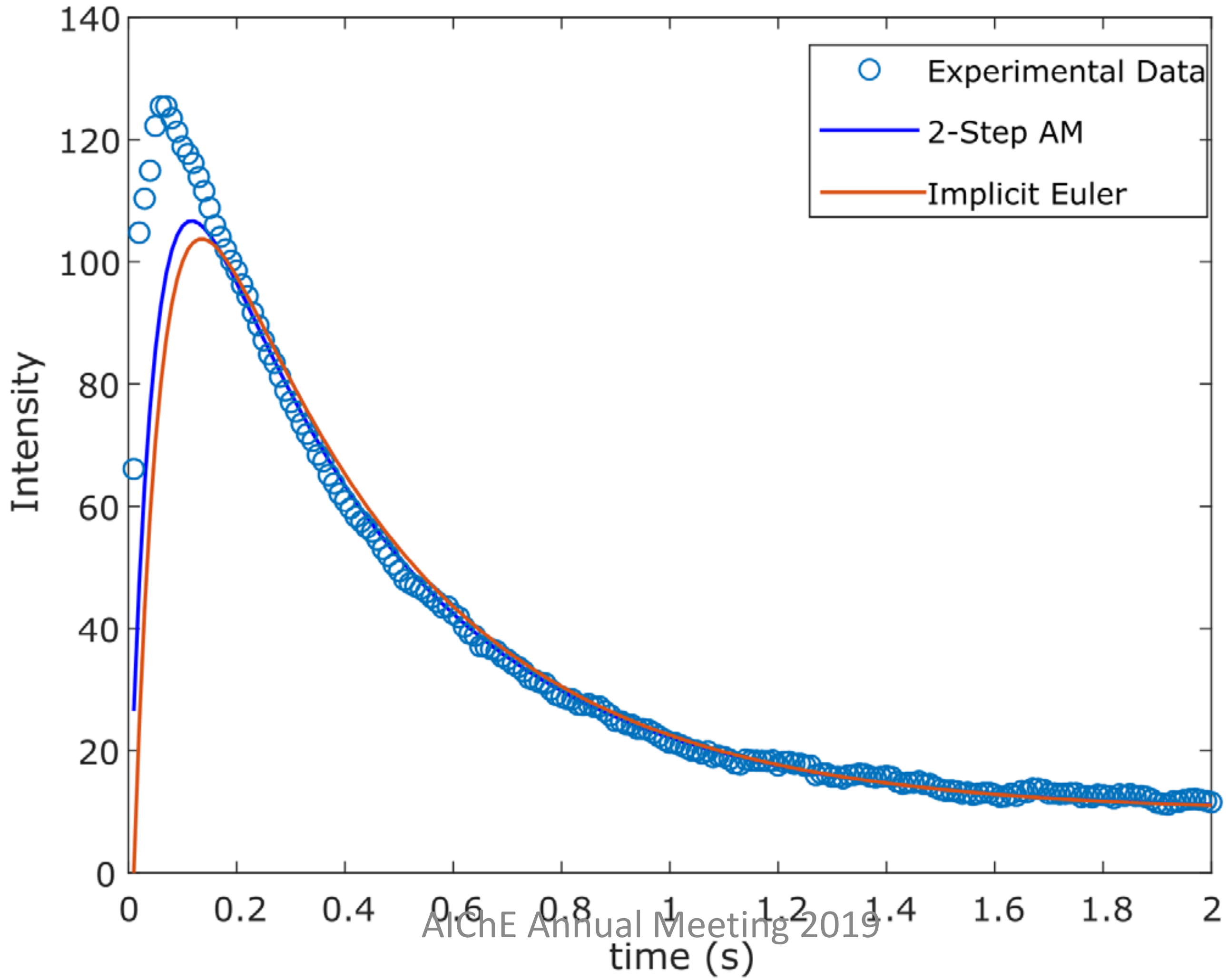
ODE-IVP:

$\dot{x}_A = k_1 x_Z x_Y - c_{O2}($

$\dot{x}_B = k_{3f} c_{O2} x_A - (k$

$\dot{x}_Y = -k_{1s} x_Y x_Z,$

$\dot{\boldsymbol{x}}(t=0) = (0, 0, 0,$



s

$- \mathrm{I}^i_{\text{data}})^2$

$\frac{2}{21} x_B^i + \frac{2}{21} x_D^i$

$_4)$

$x_Y, x_Z)$

$_3, c_{O2}, \Delta t, n$

16. J. W. Taylor, et al. **Direct meas** . A, 108 (2004), pp. 7193–7203.
17. A. B. Singer et al., **Global dyn** pp. 971–976
18. Mitsos et al. **McCormick-base**
19. Stuber, M.D. et al. **Convex an**

Conclusion

- We have developed a method for rigorously bounding the state trajectories of (stiff) parametric ODE-IVPs
- We have developed the theory for higher-order implicit integration methods (parametric implicit linear multistep methods).
- Focused on two-step (2nd-order) methods
  - Much greater accuracy than implicit Euler
  - Unconditionally stable
- Developed an open-source package for use with our EAGO.jl open-source solver

https://github.com/PSORLab/EAGO.jl
https://github.com/PSORLab/EAGODifferential.jl

Thank you!

Any Questions?

UNIVERSITY OF CONNECTICUT

The Global Home of Chemical Engineers

This material is based upon work supported by the National Science Foundation under Grant No.: 1560072, 1706343, 1932723. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Implementation

Problem Statement

- Fit the rate ... es
  addition ...
- First add ... $x^i_{\mathrm{data}})^2$
- Explicit ... $x^i_B + \frac{2}{21} x^i_D$
- Implicit

pODE IVP:

$\dot{x}_A = k_1 x_Z$ ...

$\dot{x}_B = k_{3f} C$ ... $, x_Z)$

$\dot{x}_Y = -k_1$ ...

$\dot{\boldsymbol{x}}(t = 0)$ ... $_{O2}, \Delta t, n$

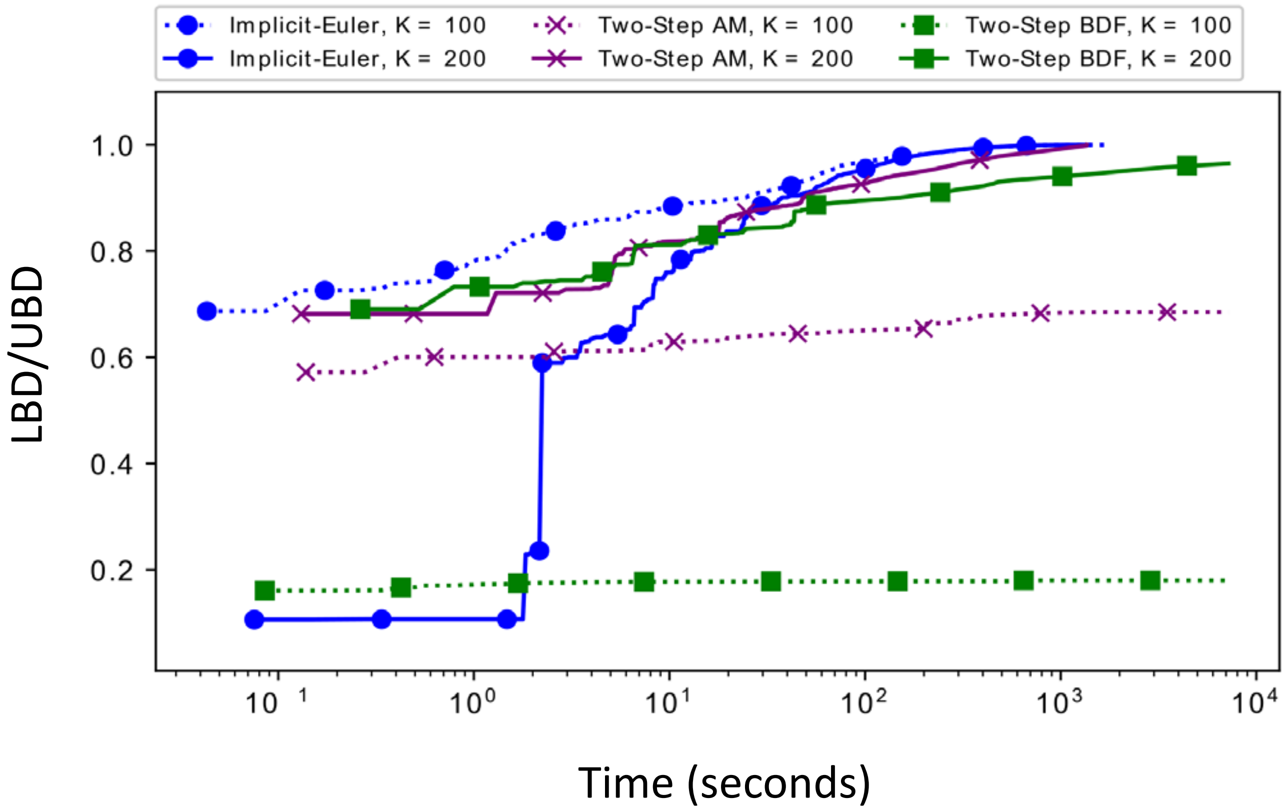
- Affine relaxations used to compute lower bound (CPLEX 12.8).
- Upper-bound computed by integrating ODE at midpoint of active node then evaluating objective & constraints.
- Duality-based bound tightening was performed. Two iterations of each PILMS with was used after five iterations of a block sequential parametric interval method.
- Absolute and relative convergence tolerances for the B&B algorithm of 10^{-2} and 10^{-5}, respectively.

16. J. W. Taylor, et al. D... (2004), pp. 7193–7203.
17. A. B. Singer et al., Glob... 71–976
18. Mitsos et al. **McCormick-based relaxations of algorithms.** *SIAM Journal on Optimization*, SIAM (2009) 20, 73-601
19. Stuber, M.D. et al. **Convex and concave relaxations of implicit functions.** *Optimization Methods and Software* (2015), 30, 424-460

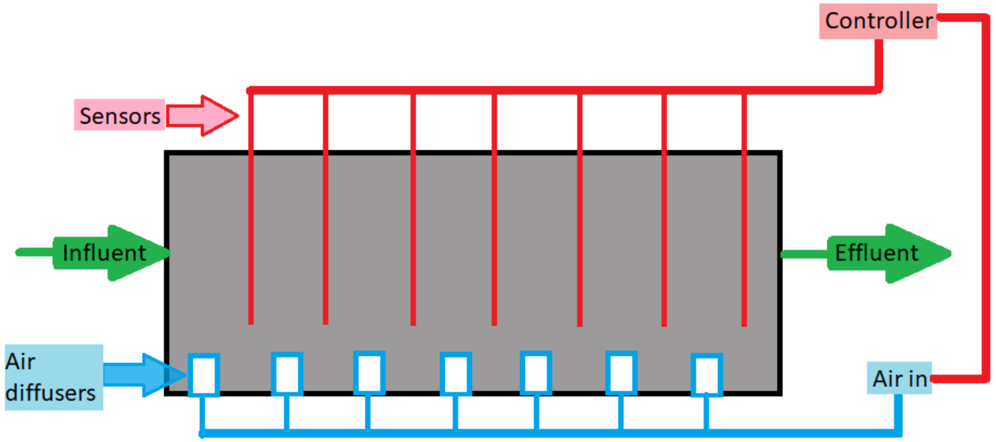
Kinetic Problem

Kinetic problem was solved subject to three discretization schemes for $K = 100$ and $K = 200$.

Solution Method	K	Iterations	Average time per iteration	Solution time	SSE at Solution
Implicit Euler	100	33987	45×10^{-3}s	29.7min	26947.246
	200	23,525	59×10^{-3}s	23.4min	16796.038
2-Step AM	100	62024	12×10^{-2}s	>2 h	N/A*
	200	6068	22×10^{-2}s	22.6min	13077.998
2-Step BDF	100	88408	81×10^{-3}s	>2 h	N/A*
	200	27600	26×10^{-2}s	>2 h	N/A*
Explicit Euler	100	>300,000	23×10^{-4}s	>2 h	N/A
	200	>300,000	24×10^{-4}s	>2 h	N/A



Wastewater Example



Control of a 9-species biological reaction for wastewater treatment.