UCONN
UNIVERSITY OF CONNECTICUT

Solving Optimization Problems with Embedded Dynamical Systems

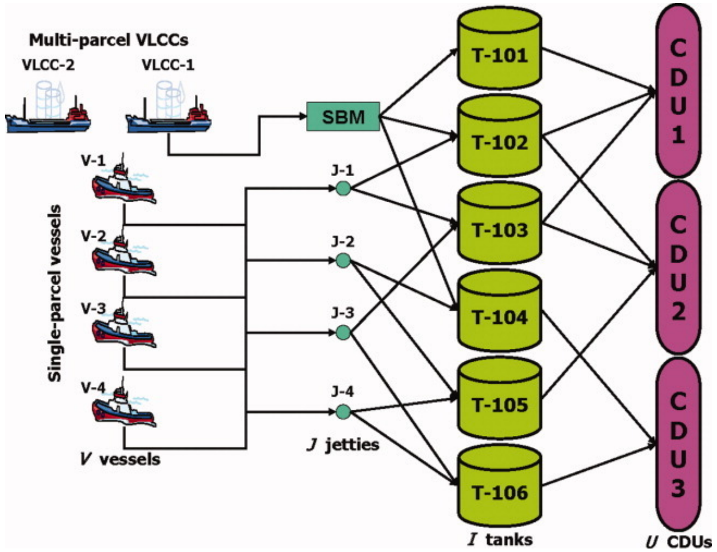
Matthew Wilhelm, PhD Candidate
Matthew Stuber, Assistant Professor

juliacon 2021

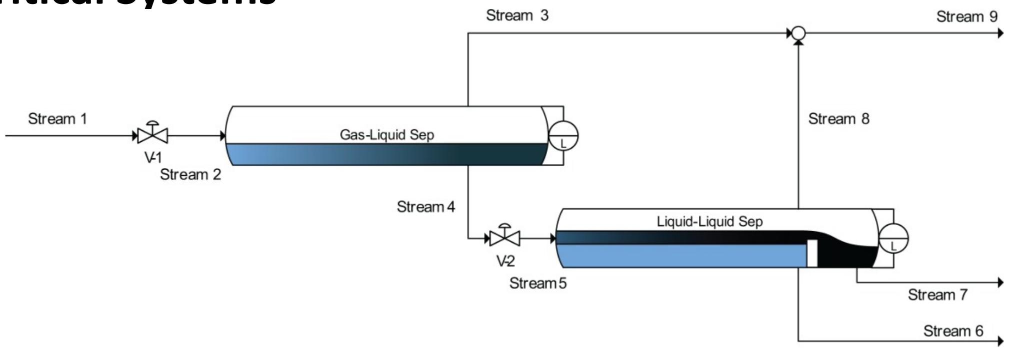
Process Systems and Operations Research Laboratory

Deterministic Global Optimization

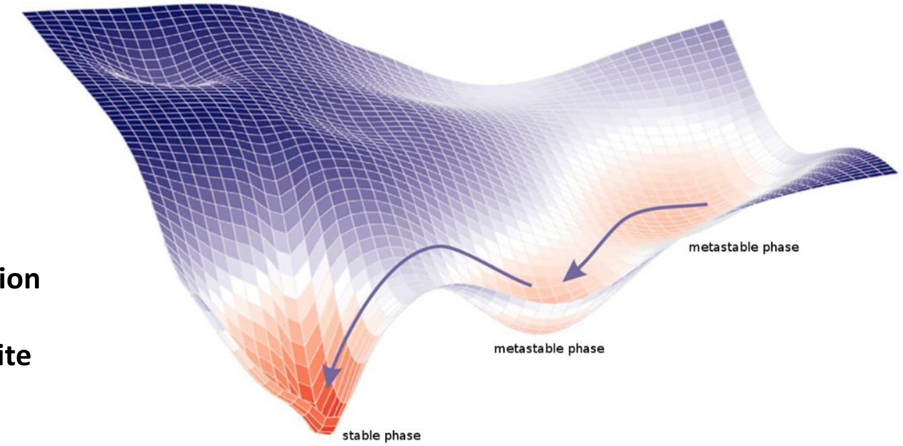
Mixing & Pooling Problems[1]



Safety Critical Systems[2]



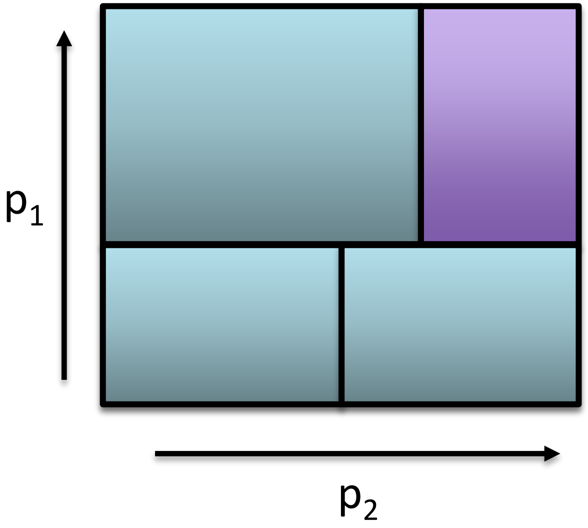
Physical Solutions[3]



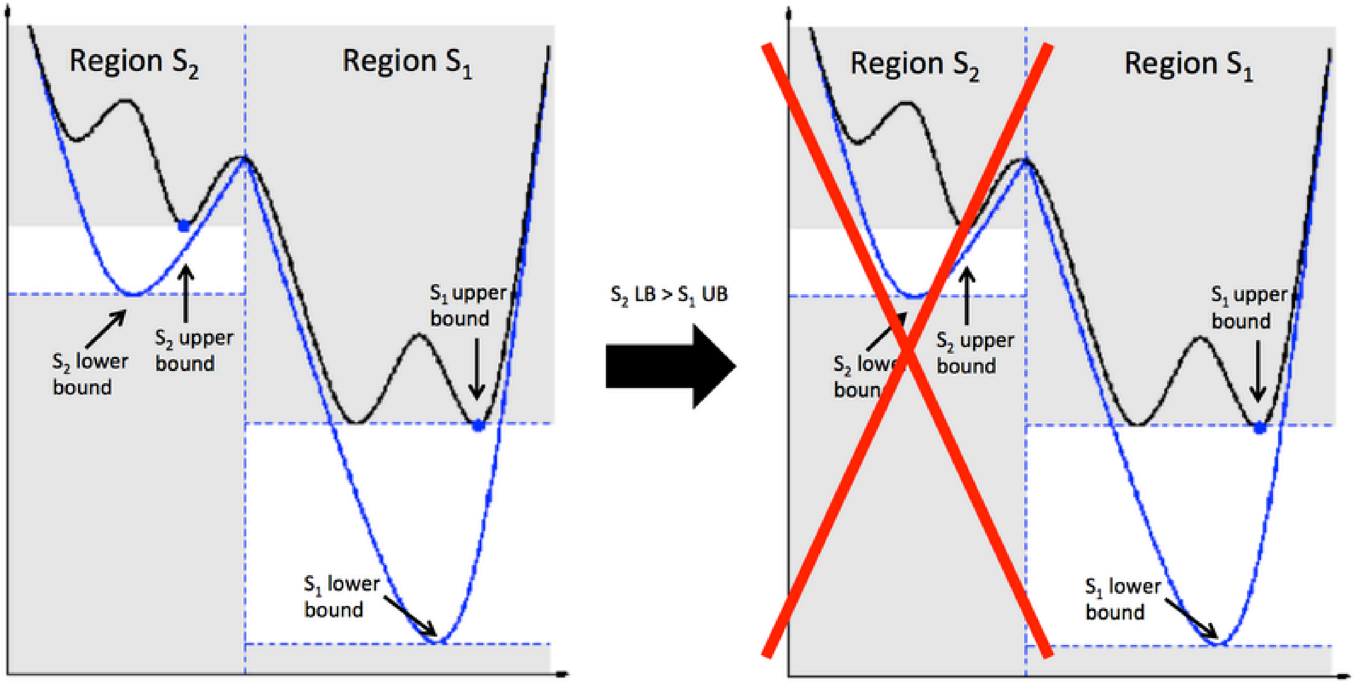
1. Li, J.; Misener, R.; and Floudas, CA. **Continuous-time modeling and global optimization approach for scheduling of crude oil operations**. *AIChE Journal* (2012): 205-226.
2. Stuber, MD et al. **Worst-case design of subsea production facilities using semi-infinite programming.** *AIChE Journal* (2014): 2513-2524.
3. Grajcarova, L. **Simulations of structural phase transitions in crystals using ab initio metadynamics**. INIS-IAEA (2013).

Spatial Branch & Bound

Select Node



Compute Bounds **Compute Fathom**



https://optimization.mccormick.northwestern.edu/index.php/Spatial_branch_and_bound_method

4. Horst, R., & Tuy, H. **Global optimization: deterministic approaches.** *Berlin: Springer* (1993).
5. Quesada, I., & Grossmann, I. E. **A global optimization algorithm for linear fractional and bilinear programs.** *Journal of Global Optimization, 6* (1995): 39-76.
6. Ryoo, H. S., & Sahinidis, N. V. **A branch-and-reduce approach for global optimization.** *Journal of Global Optimization 8* (1996): 107-138.

Reduced-Space Relaxations

Discretize-then-Relax

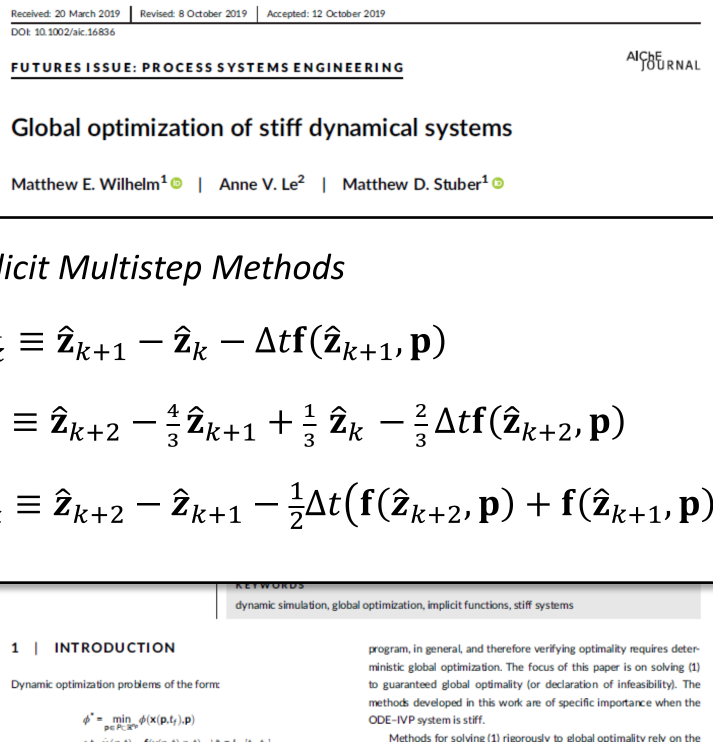
$$\mathbf{x}(\tau_{q+1},\mathbf{p}) \in \underbrace{\mathbf{x}(\tau_q,\mathbf{p}) + \sum_{j=1}^{p} \frac{h^j}{j!}\mathbf{f}^{(j)}(\mathbf{x}(\tau_q,\mathbf{p}),\mathbf{p})}_{\text{Taylor Series}} + \underbrace{\frac{h^{p+1}}{(p+1)!}\mathbf{f}^{(p+1)}(X(\tau_q),P)}_{\text{Remainder Bound}}$$

- ➢ Existence and uniqueness test followed by contraction [7,8].
- ➢ McCormick Relaxation method [7] & Taylor-McCormick relaxations [9]

Relax-then-Discretize

$$\dot{\mathbf{x}}^{cv}(t,\mathbf{p}) = \mathbf{f}^{cv}(t,\mathbf{p},\mathbf{x}^{cv}(t,\mathbf{p}),\mathbf{x}^{cc}(t,\mathbf{p})), \quad \mathbf{x}^{cv}(t_0,\mathbf{p}) = \mathbf{x}_0^{cv}(\mathbf{p})$$
$$\dot{\mathbf{x}}^{cc}(t,\mathbf{p}) = \mathbf{f}^{cc}(t,\mathbf{p},\mathbf{x}^{cv}(t,\mathbf{p}),\mathbf{x}^{cc}(t,\mathbf{p})), \quad \mathbf{x}^{cc}(t_0,\mathbf{p}) = \mathbf{x}_0^{cc}(\mathbf{p})$$

- ➢ Development of interval-based differential inequality [10,11,12]



Received: 20 March 2019 | Revised: 8 October 2019 | Accepted: 12 October 2019
DOI: 10.1002/aic.16836

FUTURES ISSUE: PROCESS SYSTEMS ENGINEERING

AIChE JOURNAL

Global optimization of stiff dynamical systems

Matthew E. Wilhelm[1] | Anne V. Le[2] | Matthew D. Stuber[1]

KEYWORDS

dynamic simulation, global optimization, implicit functions, stiff systems

1 | INTRODUCTION

Dynamic optimization problems of the form:

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

program, in general, and therefore verifying optimality requires deterministic global optimization. The focus of this paper is on solving (1) to guaranteed global optimality (or declaration of infeasibility). The methods developed in this work are of specific importance when the ODE–IVP system is stiff.

Methods for solving (1) rigorously to global optimality rely on the

Implicit Multistep Methods

$$\boldsymbol{\xi}_k^1 \equiv \hat{\mathbf{z}}_{k+1} - \hat{\mathbf{z}}_k - \Delta t\mathbf{f}(\hat{\mathbf{z}}_{k+1},\mathbf{p})$$

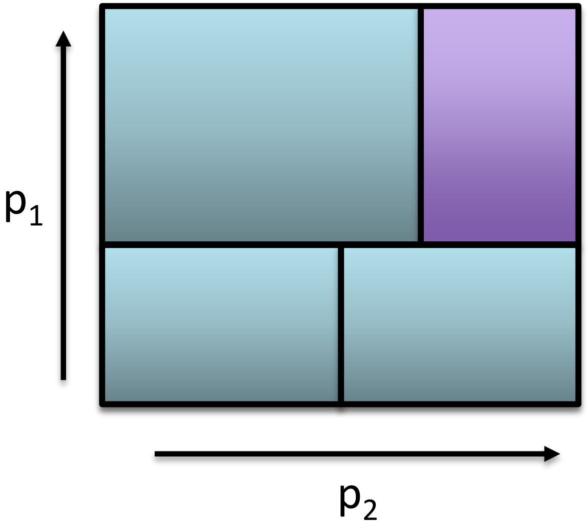
$$\boldsymbol{\xi}_k^2 \equiv \hat{\mathbf{z}}_{k+2} - \tfrac{4}{3}\hat{\mathbf{z}}_{k+1} + \tfrac{1}{3}\hat{\mathbf{z}}_k - \tfrac{2}{3}\Delta t\mathbf{f}(\hat{\mathbf{z}}_{k+2},\mathbf{p})$$

$$\boldsymbol{\zeta}_k^2 \equiv \hat{\mathbf{z}}_{k+2} - \hat{\mathbf{z}}_{k+1} - \tfrac{1}{2}\Delta t\left(\mathbf{f}(\hat{\mathbf{z}}_{k+2},\mathbf{p}) + \mathbf{f}(\hat{\mathbf{z}}_{k+1},\mathbf{p})\right)$$

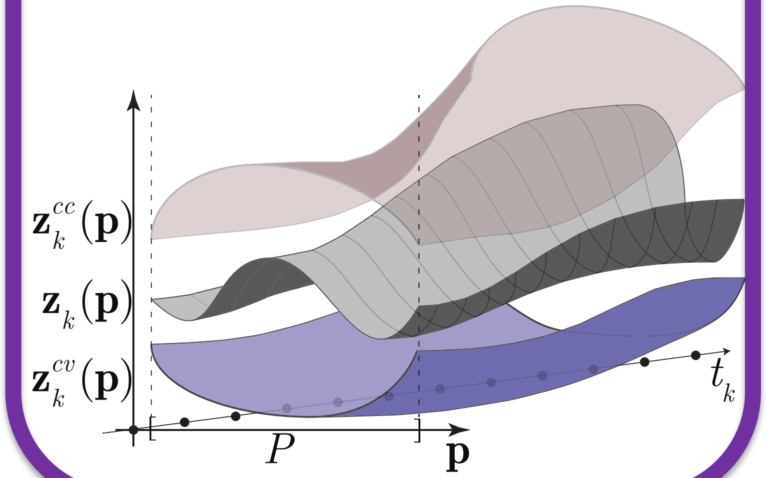
7. Sahlodin, Ali M., and Benoit Chachuat. **Discretize-then-relax approach for convex/concave relaxations of the solutions of parametric ODEs.** *Applied Numerical Mathematics* 61.7 (2011): 803-820.
8. Berz, Martin, and Georg Hoffstätter. **Computation and application of Taylor polynomials with interval remainder bounds.** *Reliable Computing* 4.1 (1998): 83-97.
9. Sahlodin, Ali Mohammad, and Benoit Chachuat. **Convex/concave relaxations of parametric ODEs using Taylor models.** *Computers & Chemical Engineering* 35.5 (2011): 844-857.
10. Scott, Joseph K., and Paul I. Barton. **Improved relaxations for the parametric solutions of ODEs using differential inequalities.** *Journal of Global Optimization* 57.1 (2013): 143-176.
11. Scott, Joseph K., and Paul I. Barton. **Bounds on the reachable sets of nonlinear control systems.** *Automatica* 49.1 (2013): 93-100.
12. Scott, Joseph K., Benoit Chachuat, and Paul I. Barton. **Nonlinear convex and concave relaxations for the solutions of parametric ODEs.** *Optimal Control Applications and Methods* 34.2 (2013): 145-163

Lower Bounding Problem

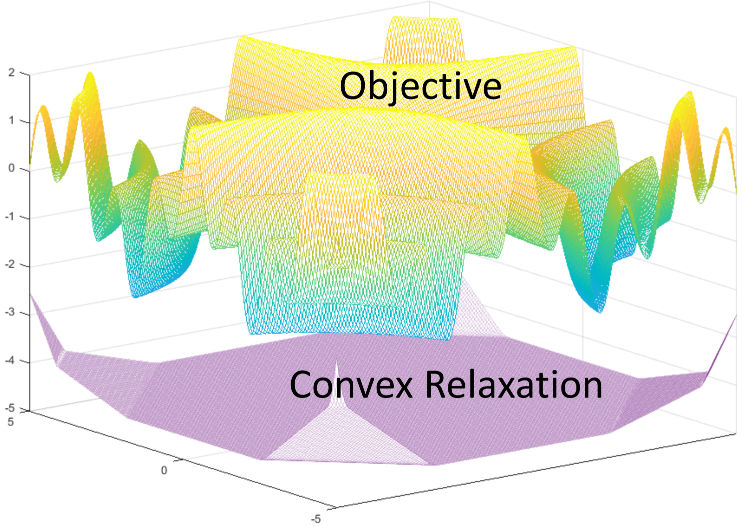
(Domain of Active Node)



(Relaxation of State Variables)[10]



(Relaxation of Objective and Constraints)

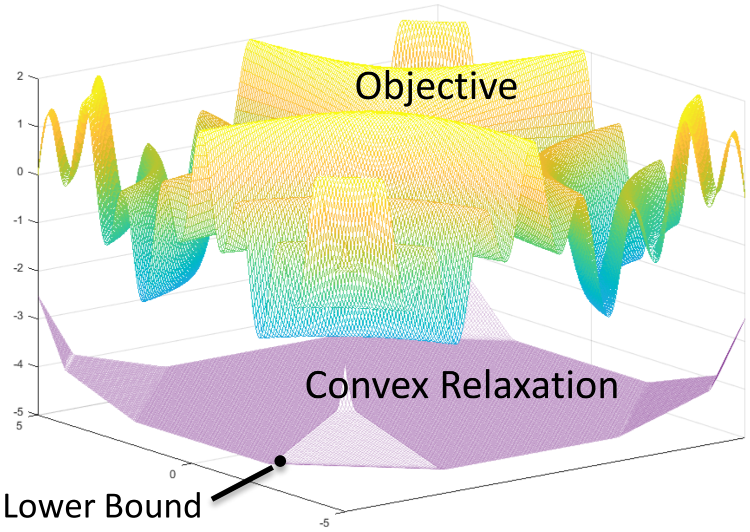


McCormick Operator Arithmetic[11,12,13]

13. Wilhelm, ME; Le, AV; and Stuber. MD. **Global Optimization of Stiff Dynamical Systems.** *AIChE Journal: Futures Issue,* 65 (12), *2019*
14. Mitsos, A, et al. **McCormick-based relaxations of algorithms.** *SIAM Journal on Optimization*, SIAM (2009) 20, 73-601.
15. Scott, JK, et al. **Generalized McCormick relaxations.** *Journal of Global Optimization* 51.4 (2011): 569-606.
16. Khan, K. et al. **Differentiable McCormick relaxations.** *Journal Global Optimization* (2017), 67(4), 687-729.

Lower Bounding Problem

***Solve convex nonlinear program*[4]**



***Solve linear program*[4]**

Construct polyhedral relaxations from convex relaxations by (sub)-gradients

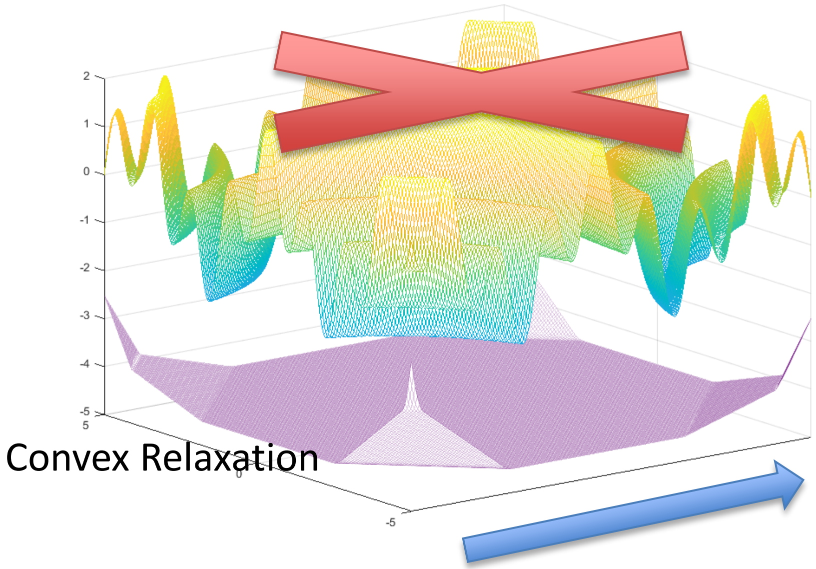
***Interval Bounds*[7]**

Refine bounds using natural interval using extensions of hyperplanes defining polyhedral relaxation

4. Horst, R., & Tuy, H. **Global optimization: deterministic approaches.** *Berlin: Springer* (1993).
7. Sahlodin, Ali M., and Benoit Chachuat. **Discretize-then-relax approach for convex/concave relaxations of the solutions of parametric ODEs.** *Applied Numerical Mathematics* 61.7 (2011): 803-820.

Domain Reduction

***Solve series of convex nonlinear program to shrink domain*[17]**



Objective → $\max_{p \in P} \pm p$

***Solve series of linear program*[17]**

Construct polyhedral relaxations from convex relaxations by (sub)-gradients

***Reverse Propagate Interval Bounds*[17]**

17. Puranik, Y., & Sahinidis, N. V. **Domain reduction techniques for global NLP and MINLP optimization.** *Constraints*, *22 (2017)*, 338-376.

Relaxation - Requirements

Solve convex nonlinear program

Solve linear program

Construct polyhedral relaxations from convex relaxations by (sub)-gradients

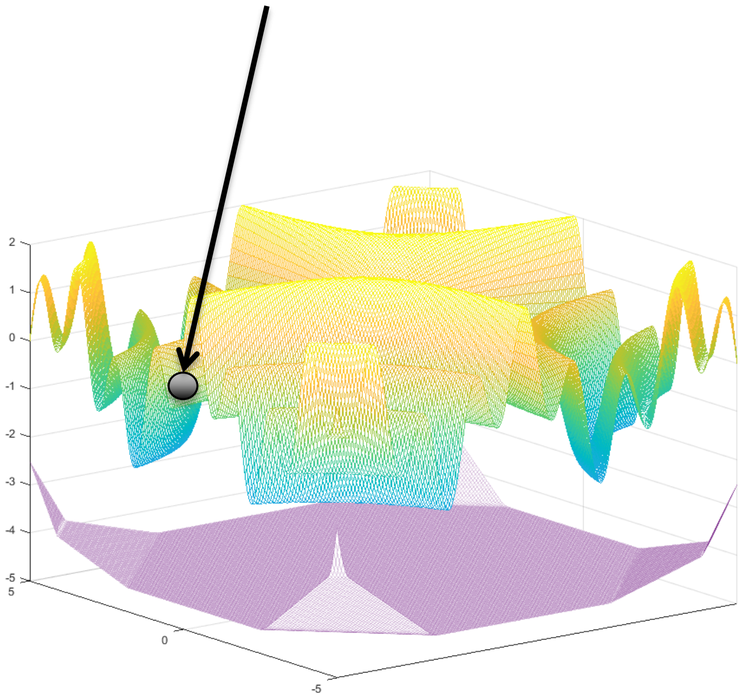
Interval Bounds

Refine bounds using natural interval using extensions of hyperplanes defining polyhedral relaxation

1. Convex/concave relaxation
2. (Sub)gradient of convex/concave relaxation
3. Interval bound
4. Status of the relaxation routine

Upper Bounding Problem

Compute a feasible point



Ideally, local minimum
–OR—
acceptably value.

- Evaluate ODE at value
- Evaluate objective/constraints
- Gradients
- Jacobian
- Hessian may be helpful

4. Horst, R., & Tuy, H. **Global optimization: deterministic approaches.** *Berlin: Springer* (1993).

Implementation

Core Algorithms

- **Relax-then-Discretize Algorithms**
- **Discretize-then-Relax Algorithms**

Extendable Global Optimizer

EAGO

18. Wilhelm, M. E., **DynamicBounds.jl**, (2020), GitHub repository, https://github.com/PSORLab/DynamicBounds.jl
19. Wilhelm, M. E., **EAGODynamicOptimizer.jl**, (2020), GitHub repository, https://github.com/PSORLab/EAGODynamicOptimizer.jl
20. Wilhelm, M. E., **McCormick.jl**, (2020), GitHub repository, https://github.com/PSORLab/McCormick.jl
21. Wilhelm, M. E., and M. D. Stuber. **EAGO. jl: easy advanced global optimization in Julia.** *Optimization Methods and Software* (2020): 1-26.
22. Bezanson, Jeff, et al. **Julia: A fresh approach to numerical computing.** *SIAM review* 59.1 (2017): 65-98.

Implementation

Core Algorithms

- **DynamicBoundspODEsDiscrete.jl**
  – Discrete time approaches
- **DynamicBoundspODEsIneq.jl**
  -- Continuous time approaches

Extendable Global Optimizer

EAGO

High performance needed (may resolve end up evaluating >1,000,000 trajectories for single global optimization problem).

Additional complexity in handling constraints results from infinitely parameterized variables.

18. Wilhelm, M. E., **DynamicBounds.jl**, (2020), GitHub repository, https://github.com/PSORLab/DynamicBounds.jl
19. Wilhelm, M. E., **EAGODynamicOptimizer.jl**, (2020), GitHub repository, https://github.com/PSORLab/EAGODynamicOptimizer.jl
20. Wilhelm, M. E., **McCormick.jl**, (2020), GitHub repository, https://github.com/PSORLab/McCormick.jl
21. Wilhelm, M. E., and M. D. Stuber. **EAGO. jl: easy advanced global optimization in Julia.** *Optimization Methods and Software* (2020): 1-26.
22. Bezanson, Jeff, et al. **Julia: A fresh approach to numerical computing.** *SIAM review* 59.1 (2017): 65-98.

PSOR

Implementation

Core Algorithms

- **DynamicBoundspODEsDiscrete.jl**
  – Discrete time approaches
- **DynamicBoundspODEsIneq.jl**
  -- Continuous time approaches

Abstract Layer

DynamicBounds.jl

- **DynamicBounds.jl**
- **DynamicBoundsBase.jl**

Extendable Global Optimizer

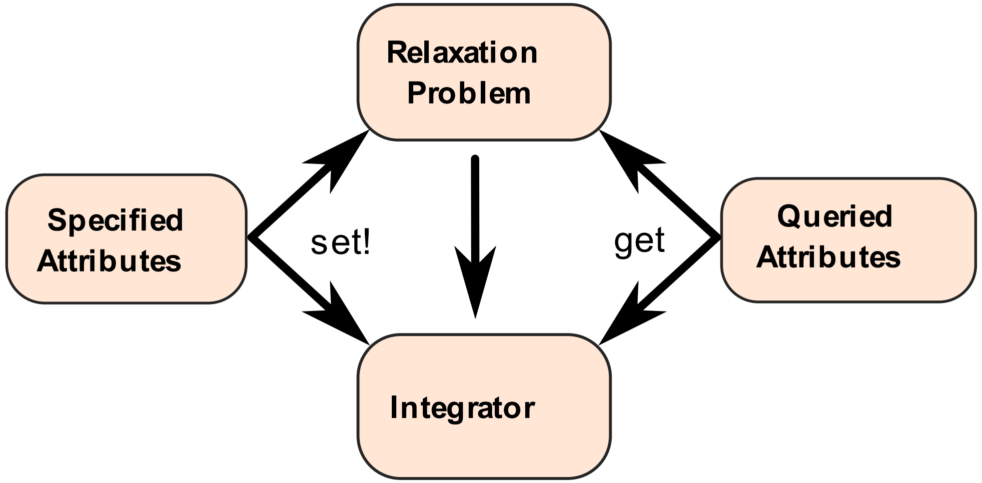
EAGO

EAGODynamicOptimizer.jl

18. Wilhelm, M. E., **DynamicBounds.jl**, (2020), GitHub repository, https://github.com/PSORLab/DynamicBounds.jl
19. Wilhelm, M. E., **EAGODynamicOptimizer.jl**, (2020), GitHub repository, https://github.com/PSORLab/EAGODynamicOptimizer.jl
20. Wilhelm, M. E., **McCormick.jl**, (2020), GitHub repository, https://github.com/PSORLab/McCormick.jl
21. Wilhelm, M. E., and M. D. Stuber. **EAGO. jl: easy advanced global optimization in Julia.** *Optimization Methods and Software* (2020): 1-26.
22. Bezanson, Jeff, et al. **Julia: A fresh approach to numerical computing.** *SIAM review* 59.1 (2017): 65-98.

Implementation

- ❑ Problem storage for parametric differential equations
- ❑ Integrator that evaluates trajectories (nominal, relaxed)
- ❑ Extendable get/set access to function attributes
- ❑ References for parameterized state variables



Parametric ODE Example

Parametric ODE System

$$\dot{x}_1(\mathbf{p},t) = -p_1x_1(\mathbf{p},t)x_2(\mathbf{p},t) + p_2x_3(\mathbf{p},t) + p_6x_6(\mathbf{p},t)$$
$$\dot{x}_2(\mathbf{p},t) = -p_1x_1(\mathbf{p},t)x_2(\mathbf{p},t) + p_2x_3(\mathbf{p},t) + p_3x_3(\mathbf{p},t)$$
$$\dot{x}_3(\mathbf{p},t) = p_1x_1(\mathbf{p},t)x_2(\mathbf{p},t) - p_2x_3(\mathbf{p},t) - p_3x_3(\mathbf{p},t)$$
$$\dot{x}_4(\mathbf{p},t) = p_3x_3(\mathbf{p},t) - p_4x_4(\mathbf{p},t)x_5(\mathbf{p},t) + p_5x_6(\mathbf{p},t)$$
$$\dot{x}_5(\mathbf{p},t) = -p_4x_4(\mathbf{p},t)x_5(\mathbf{p},t) + p_5x_6(\mathbf{p},t) + p_6x_6(\mathbf{p},t)$$
$$\dot{x}_6(\mathbf{p},t) = p_4x_4(\mathbf{p},t)x_5(\mathbf{p},t) - p_5x_6(\mathbf{p},t) - p_6x_6(\mathbf{p},t)$$

Initial Condition

$$\mathbf{x}_0(\mathbf{p}) = (34, 20, 0, 0, 16, 0)$$

Parameter Bounds

$$\mathbf{k}_0 = (0.1, 0.033, 16, 5, 0.5, 0.3)$$
$$\mathbf{p} = (p_1, \ldots, p_6) \in K = [\mathbf{k}_0, 10\mathbf{k}_0]$$

Specify Problem Type

```
using DynamicBounds

function f!(dx, x, p, t)
    dx[1] = -p[1]*x[1]*x[2] + p[2]*x[3] + p[6]*x[6]
    dx[2] = -p[1]*x[1]*x[2] + p[2]*x[3] + p[3]*x[3]
    dx[3] =  p[1]*x[1]*x[2] - p[2]*x[3] - p[3]*x[3]
    dx[4] =  p[3]*x[3]  - p[4]*x[4]*x[5] + p[5]*x[6]
    dx[5] = -p[4]*x[4]*x[5] + p[5]*x[6] + p[6]*x[6]
    dx[6] =  p[4]*x[4]*x[5] - p[5]*x[6] - p[6]*x[6]
    return
end
x0(p) = [34.0, 20.0, 0.0, 0.0, 16.0, 0.0]

tspan = (0.0, 18.0e-5*250)
pL = [0.1; 0.033; 16.0; 5.0; 0.5; 0.3]; pU = 10.0*pL

ode_prob = ODERelaxProb(f!, x0, tspan, pL, pU)
```

23. Scott JK and Barton, PI. **Tight, efficient bounds on the solutions of chemical kinetic models.** *Computers & Chemical Engineering* 34.5 (2010): 717-731.

Parametric ODE Example

Polyhedral Invariants[A]

$$G = \mathbf{z} \in X_{nat} : \mathbf{M}\mathbf{z} = \mathbf{b}$$

$$\mathbf{M} = \begin{bmatrix} 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & -1 & 0 & 1 & -1 & 0 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} -20 \\ -16 \\ -2 \end{bmatrix}$$

$$X_{nat} = [0,34] \times [0,20] \times [0,20] \times [0,34] \times [0,16] \times [0,16]$$

Specify Attributes

```
M = [0.0 -1.0 -1.0  0.0  0.0  0.0;
     0.0  0.0  0.0  0.0 -1.0 -1.0;
     1.0 -1.0  0.0  1.0 -1.0  0.0]
b = [-20.0; -16.0; -2.0]
linear_invariant = PolyhedralConstraint(M, :==, b)
set!(ode_prob, linear_invariant)
u_lo = zeros(6)
u_hi = [34.0; 20.0; 20.0; 34.0; 16.0; 16.0]
set!(prob, ConstantBounds(u_lo, u_hi))
```

23. Scott JK and Barton, PI. **Tight, efficient bounds on the solutions of chemical kinetic models.** *Computers & Chemical Engineering* 34.5 (2010): 717-731.

Parametric ODE Example

Specify a relaxation method (integrator)

```
integrator = DifferentialInequality(ode_prob)
```

Specify additional attributes

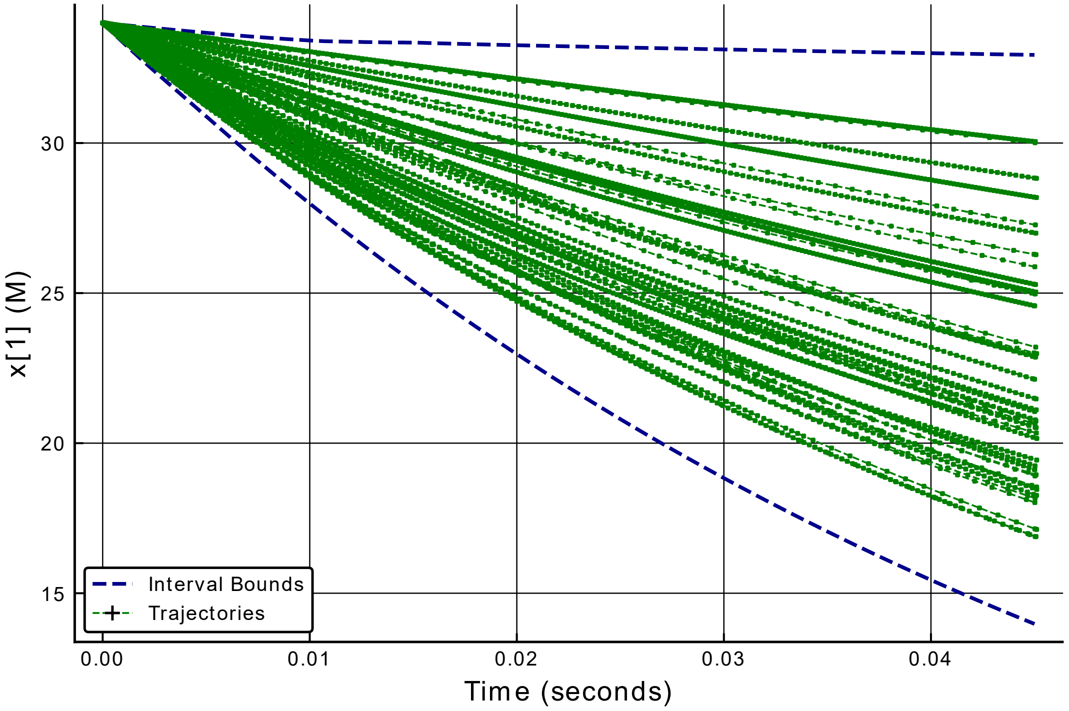
```
p = 0.5*(pL + pU)
set!(integrator, ParameterValue(), p)
```

Relax the problem

```
relax!(integrator)
```

Retrieve Attribute

```
lo = get(integrator, Bound{Lower}(3))
hi = get(integrator, Bound{Upper}(3))
cv = get(integrator, Relaxation{Lower}(3))
cc = get(integrator, Relaxation{Upper}(3))
```



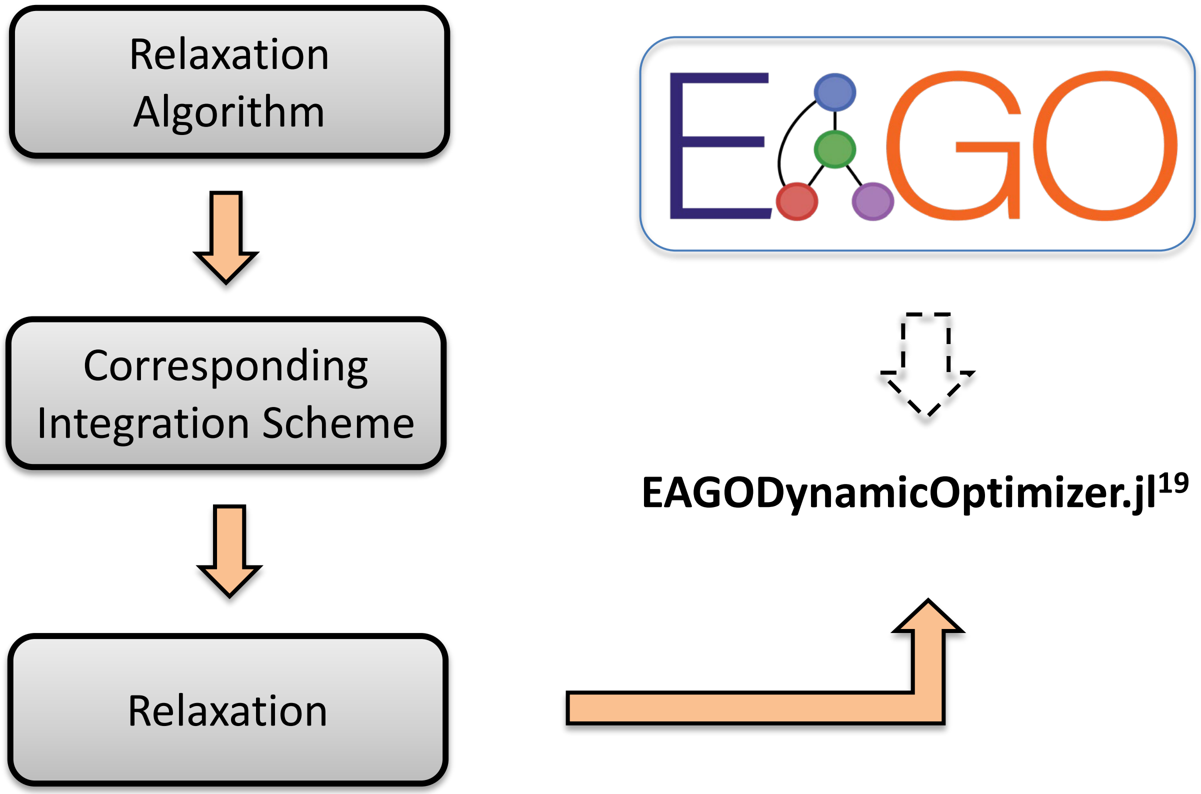
Parametric ODE Example

Integrate the problem

```
integrate!(integrator)
```

Retrieve Attributes

```
val = get(integrator, Value(3))
```

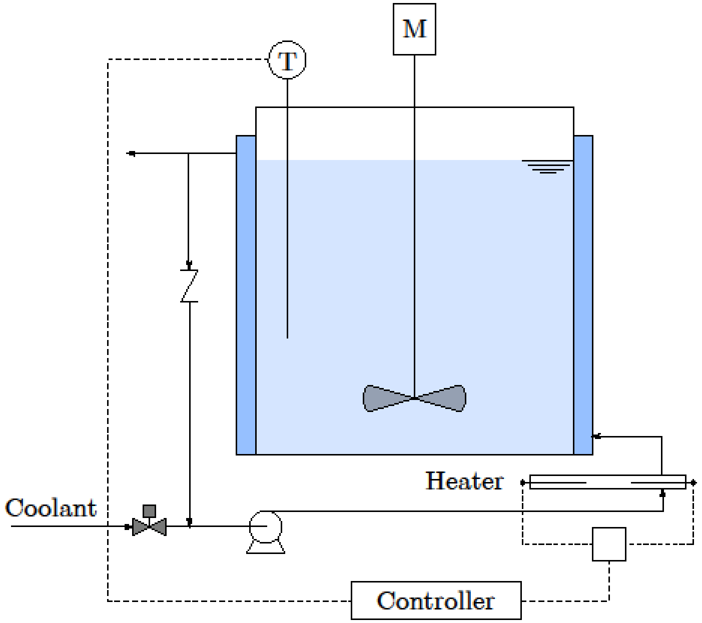


- ❑ Naïve support for in-place calculations
- ❑ Corresponding integration scheme

19. Wilhelm, M. E., **EAGODynamicOptimizer.jl**, (2020), GitHub repository, https://github.com/PSORLab/EAGODynamicOptimizer.jl

Robust Hybrid Dynamic Optimization

Exothermic Polymerization Reaction[24]



Adequate cooling at maximum temperature to withstand sensor fault?

Robust Operation SIP

$$\gamma^* = \max_{p \in P, \gamma \in \Gamma} \gamma$$
$$\text{s.t. } \gamma \le T(t_f, \mathbf{u}) - p, \ \forall \mathbf{u} \in U$$

- ❑ Embedded dynamic system
- ❑ Complex chemical kinetics (hybrid model desirable)
- ❑ Nonconvex semi-infinite program

24. Soroush M. and Kravaris, C.. **Nonlinear control of a batch polymerization reactor: An experimental study.** *AIChE Journal* 38, (1992): 1429-1448.

Robust Hybrid Dynamic Optimization

Robust Operation SIP

$$\gamma^* = \max_{p \in P, \gamma \in \Gamma} \gamma$$
$$\text{s.t. } \gamma \leq T(t_f, \mathbf{u}) - p, \ \forall \mathbf{u} \in U$$

Dynamical System (Mass & Energy Balance)

$$\frac{dC_m}{dt} = (1 + \epsilon C_m / C_{m_0}) R_m,$$
$$\frac{dC_i}{dt} = R_i + \epsilon C_i / C_{m_0} R_m,$$
$$\frac{dT}{dt} = \frac{\alpha_0 k_P \xi_0 C_m}{1 + \epsilon C_m / C_{m_0}} + \alpha_1 (T_j - T)$$

Rate constants (R_m, R_i) from pseudo-empirical models

Relaxation of Dynamic System

Rate Expression *(Greatly Simplified….)*

$$R_m = -C_m \xi_0 (k_P + k_{fm}),$$
$$R_i = -k_i C_i,$$

$$\xi_0^2 k_t(\xi_0, C_m, T) - 2\zeta k_i C_i = 0.$$

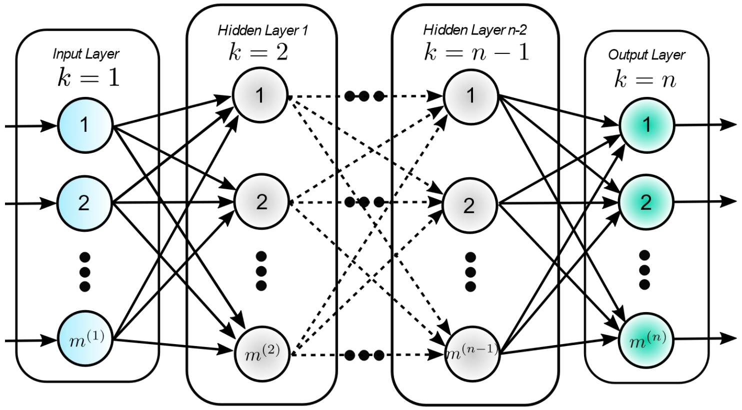
Use 3-layer GeLU ANN as in place of solving nonlinear equation from quasi-steady state assumption

Able to solve SIP in 57.8 using a modified SIPres algorithm.

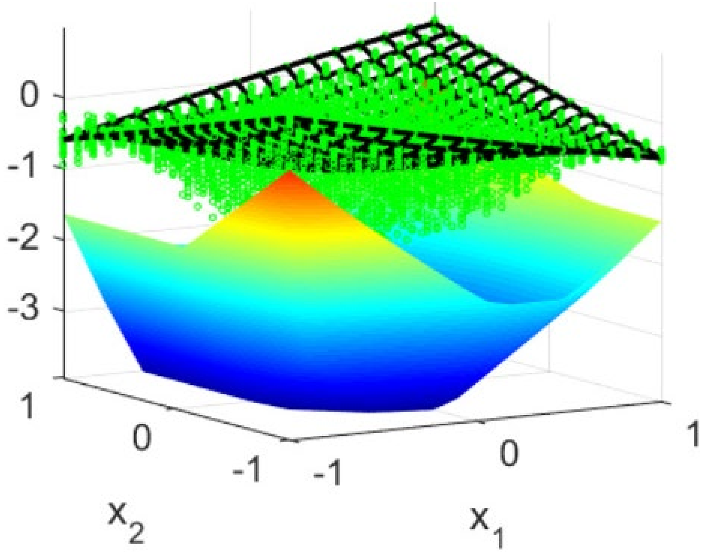
New Applications

Implicit Functions[25]

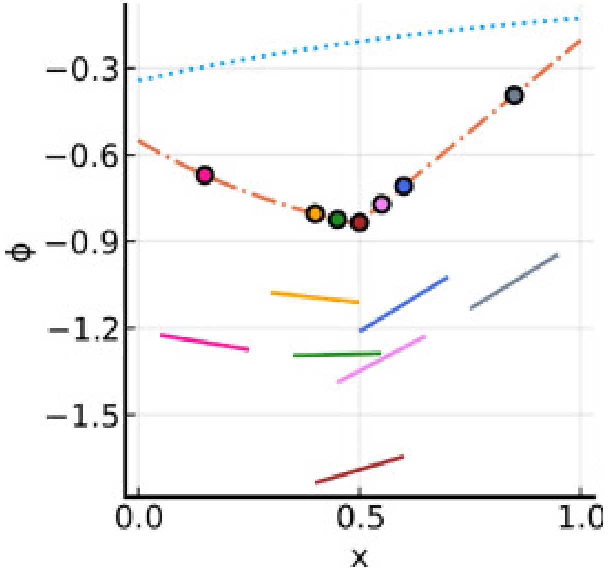
Neural Networks[26]



Continuous Random Variables[27]



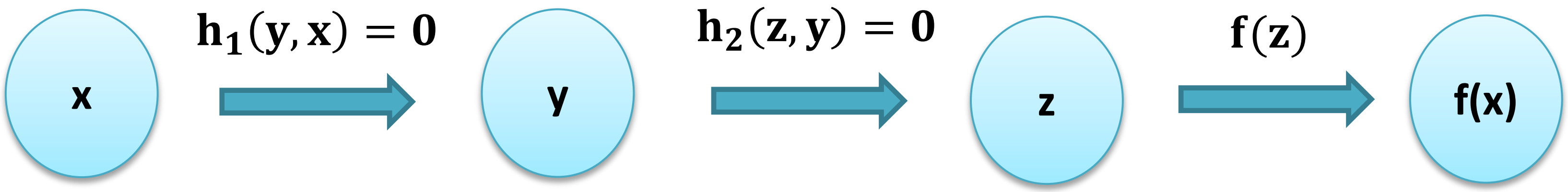
Blackbox Functions[28]



25. Stuber, MD et al. **Convex and concave relaxations of implicit functions.** *Optimization Methods and Software* (2015), 30, 424-460
26. Schweidtmann, AM, and Mitsos, A. **Deterministic global optimization with artificial neural networks embedded.** *Journal of Optimization Theory and Applications* (2019): 925-948.
27. Shao, Y and Scott JK. **Convex relaxations for global optimization under uncertainty described by continuous random variables**, *AIChE Journal*, (2018): 3023 – 3033.
28. Song, Y; Cao, H; Mehta, C; and Khan KA. **Bounding Convex Relaxations of Process Models from Below by Tractable Black-Box Sampling**, *Computers & Chemical Engineering*, In Press, (2021).

Future Work: Implementation

- Multiple input-output support in EAGO.jl[21].



- API through InfiniteOpt.jl[25].
- Extension to reverse propagation & novel algorithms.
- Wrappers for other Julia reachability methods TaylorModels.jl, etc.

21. Wilhelm, M. E., and M. D. Stuber. **EAGO.jl: easy advanced global optimization in Julia.** *Optimization Methods and Software* (2020): 1-26.
25. Stuber, M.D. et al. **Convex and concave relaxations of implicit functions.** *Optimization Methods and Software* (2015), 30, 424-460

Acknowledgements

Members of the process systems and operations research laboratory at the University of Connecticut (https://psor.uconn.edu/)

Funding:

National Science Foundation, Award No.: 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.