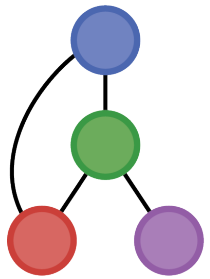
UCONN
UNIVERSITY OF CONNECTICUT

# Recent Developments in EAGO.jl (Easy Advanced Global Optimization in Julia)

**Robert Gottlieb, PhD Student**
Matthew Wilhelm, PhD
Pengfei Xu, PhD Student
Matthew Stuber, Assistant Professor

November 15th, 2022

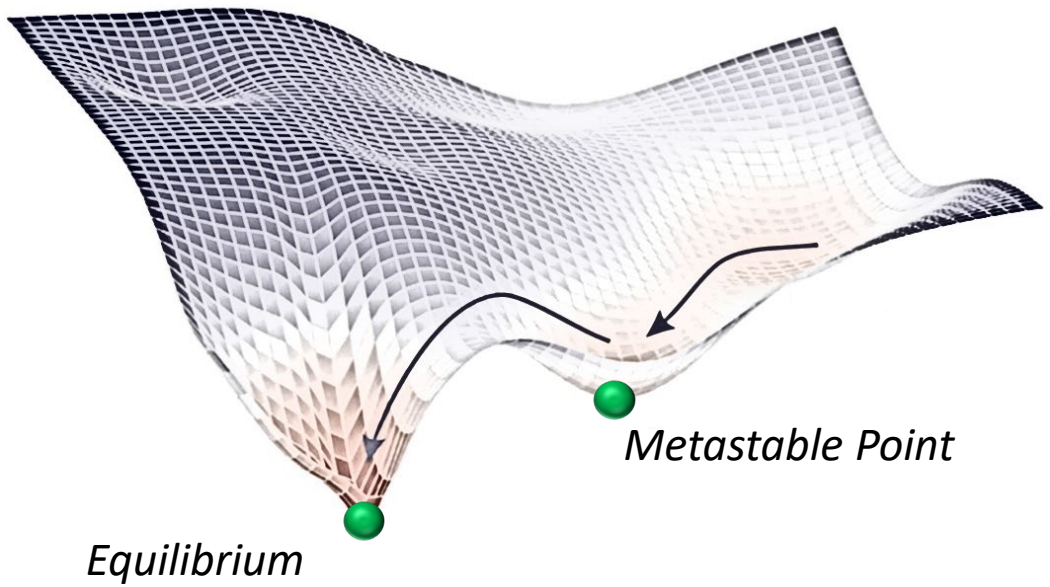
2022 AIChE ANNUAL MEETING



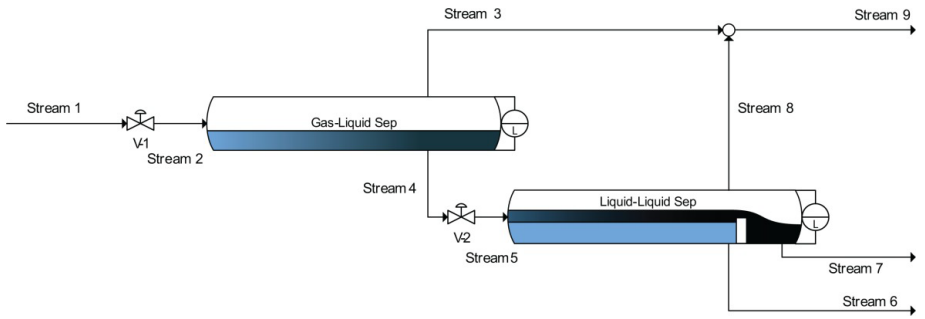
Process Systems and Operations Research Laboratory

# Importance of Global Optimization

***Physically Meaningful Computations[1]***




***Safety-Critical Systems[2]***

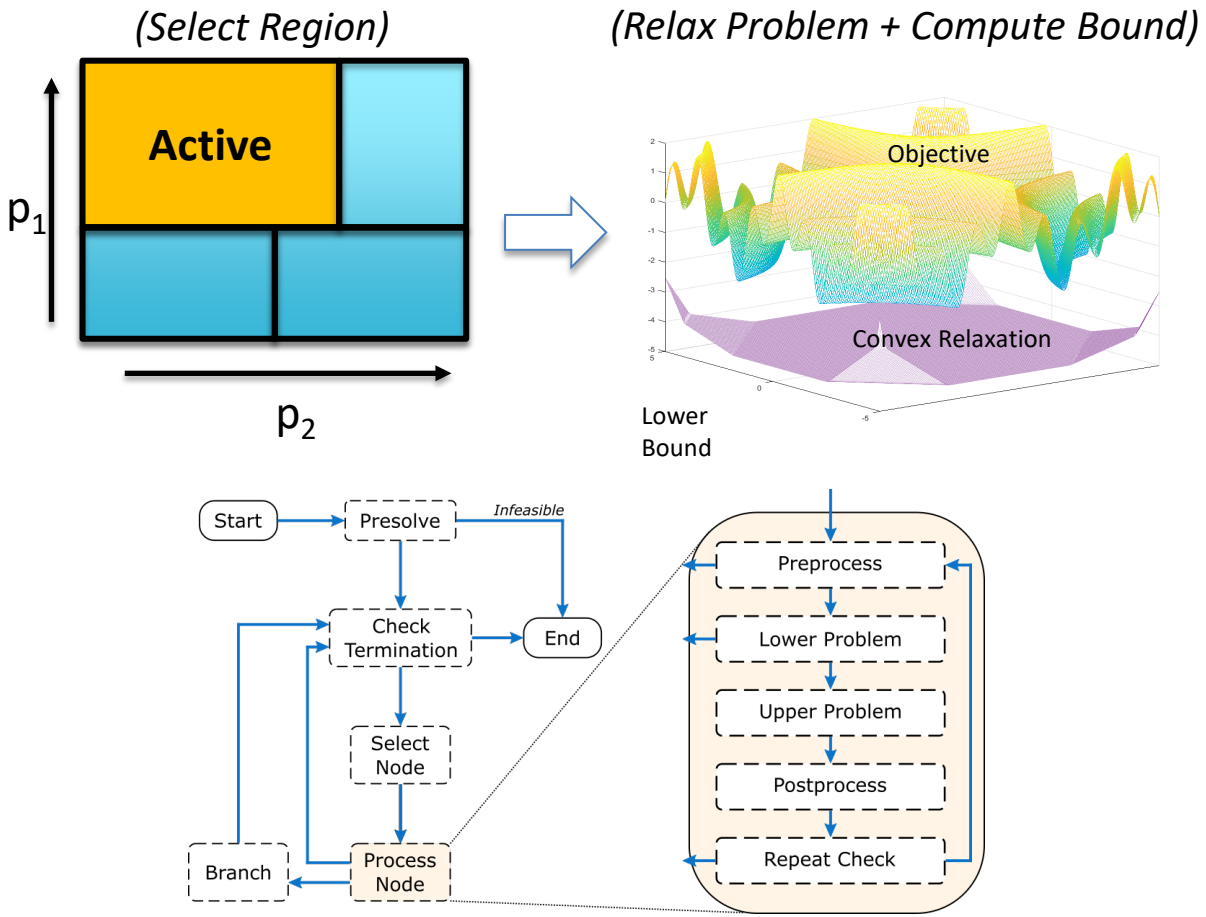



***Better quality solutions than local methods***


1. Grajcarova, L. **Simulations of structural phase transitions in crystals using ab initio metadynamics**. INIS-IAEA (2013)
2. Stuber, M.D. et al. **Worst-case design of subsea production facilities using semi-infinite programming.** *AIChE Journal* (2014): 2513-2524.

# Global Optimization

- Nonconvex MINLP formulations naturally arise in many applications
- MINLP solvers generally rely on some variation of spatial branch-and-bound[3,4]
- Relaxed subproblems are used to compute bounds and are often derived from relaxed functions[3,4]



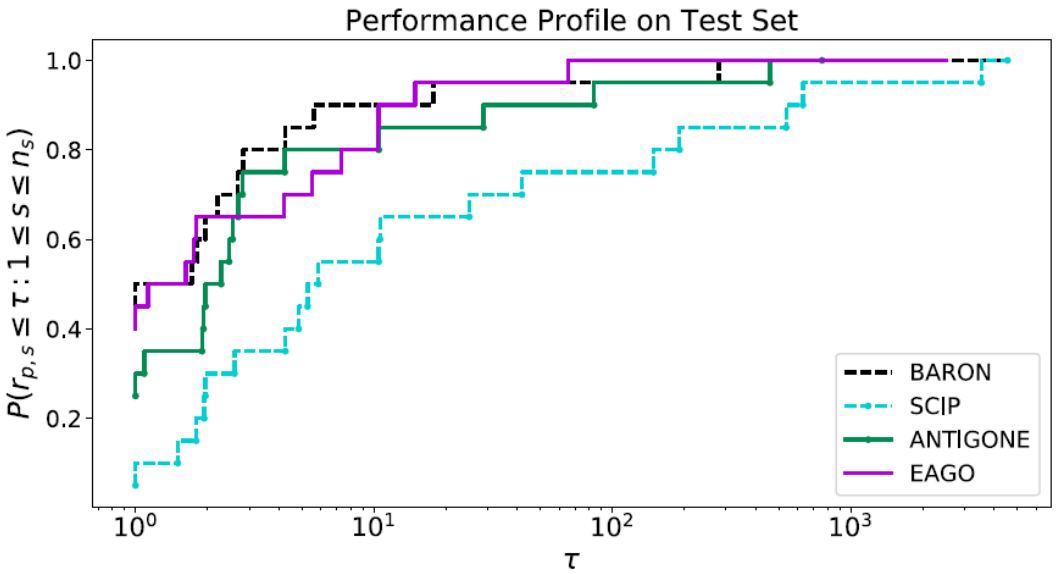


3. Wilhelm, M.E., and Stuber, M.D.. **EAGO.jl: easy advanced global optimization in Julia.** *Optimization Methods and Software*, (2020): 1-26.
4. Horst, R., and Tuy, H. *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013.

# EAGO.jl

Deterministic global optimizer[3]

- High performance solver
- Open-source and free for non-commercial use
- Designed for user-defined functions and routines



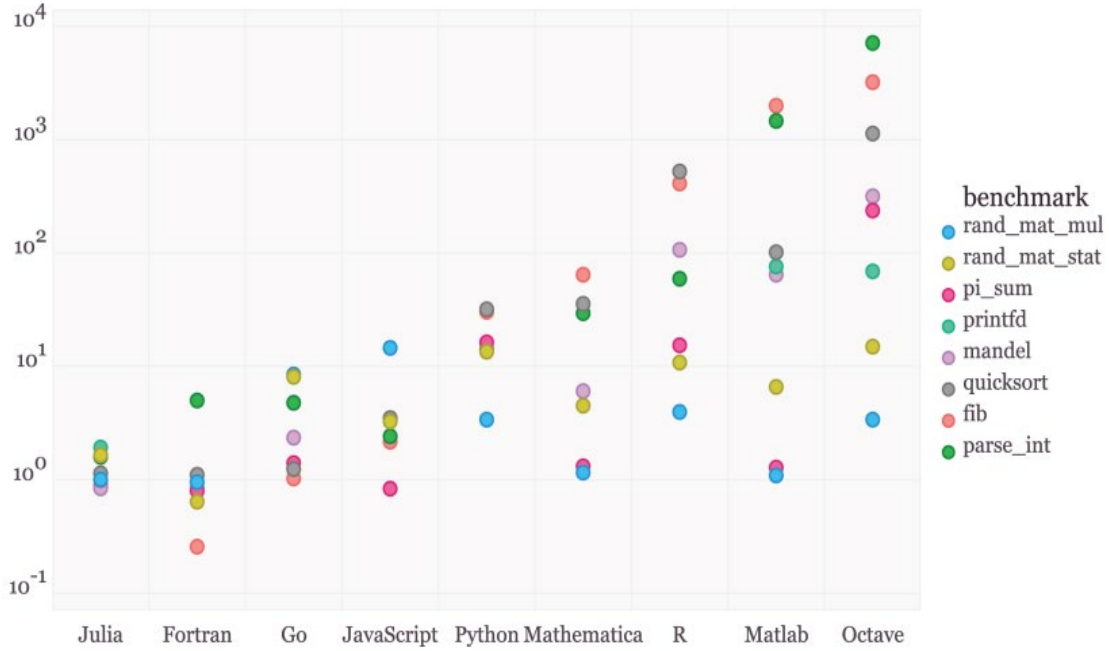

3. Wilhelm, M.E., and Stuber, M.D.. **EAGO.jl: easy advanced global optimization in Julia.** *Optimization Methods and Software*, (2020): 1-26.

# Language/Solver Capabilities

julia / JuMP

- Performance for low-level routines
- Multiple dispatch & contextual programming allow for ready extensibility (e.g., GPU parallelism)
- Ease of setup and distribution

Run-time benchmark comparison[5]




*Performance comparison of various languages performing simple microbenchmarks. Benchmark execution time relative to C. (Smaller is better; C performance = 1.0.)*

5. Bezanson, J., et al. **Julia: A Fresh Approach to Numerical Computing**. *SIAM Review*, 59: 65–98 (2017).

# McCormick Relaxations of Factorable Functions

$$\mathbf{y} = \mathbf{f}(\mathbf{g}(\mathbf{x}), \dots, \mathbf{h}(\mathbf{x}))$$

*Auxiliary Variable Method*

$$\mathbf{z_1} = \mathbf{g}(\mathbf{x})$$

$$\vdots$$

$$\mathbf{z_n} = \mathbf{h}(\mathbf{x})$$

$$\mathbf{y} = \mathbf{f}(\mathbf{z_1}, \dots, \mathbf{z_n})$$

→ **Relaxations of *f at (x, z) on (X, Z)***

*McCormick-Based Relaxations*[5,6]

**Relaxations of g at x on X**

⋮

**Relaxations of h at x on X**

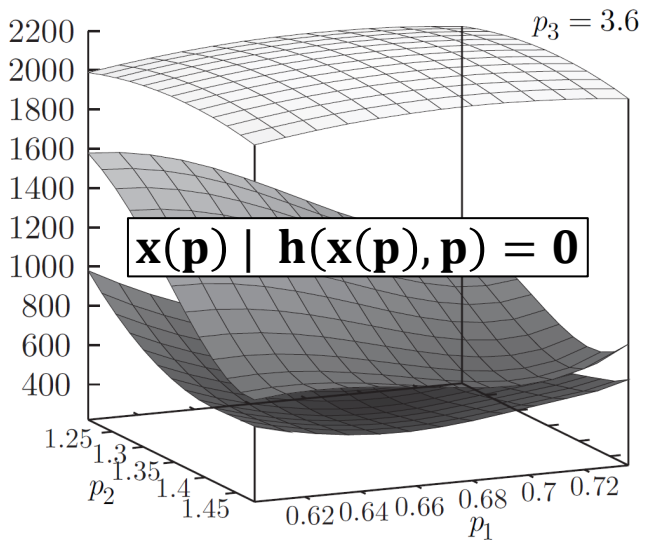
*Apply composite relaxation rules* →

**Relaxations of *f at x on X***

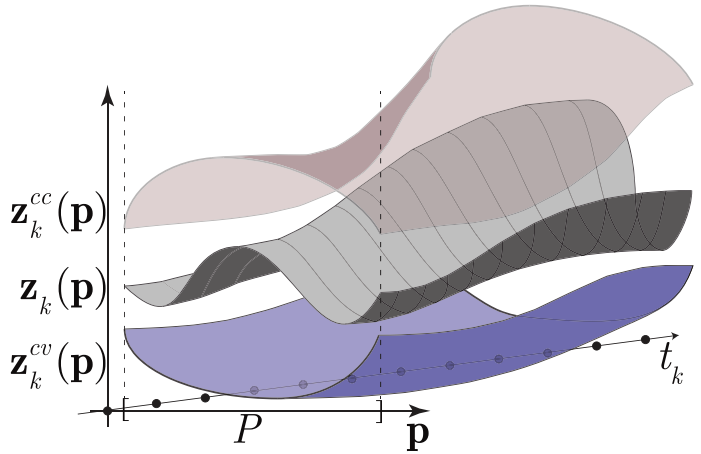

5. Mitsos, A., et al. **McCormick-based relaxations of algorithms.** *SIAM Journal on Optimization*, SIAM (2009) 20, 73-601.
6. Scott, J.K., et al. **Generalized McCormick relaxations.** *Journal of Global Optimization* 51.4 (2011): 569-606.

# Reduced Space Relaxations

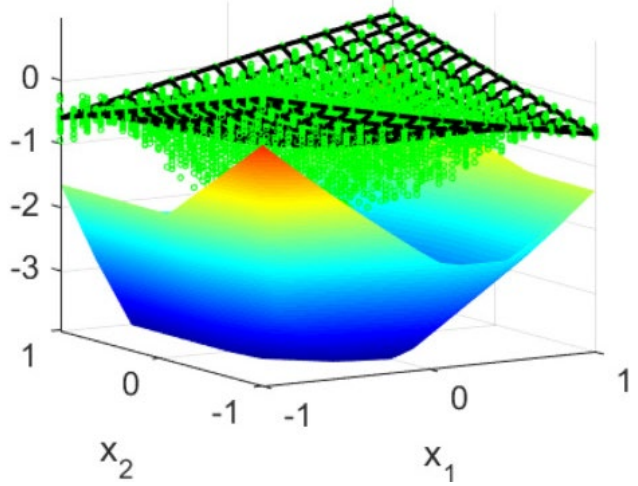
### Implicit Functions[7]



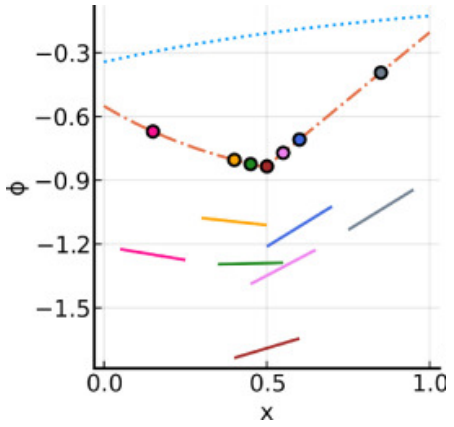

### ODEs and DAEs[8]




### Continuous Random Variables[9]




### Blackbox Functions[10]





7. Stuber, M.D. et al. **Convex and concave relaxations of implicit functions.** *Optimization Methods and Software* 30, (2015), 424-460
8. Wilhelm, M.E.; Le, A.V.; and Stuber, M.D. **Global Optimization of Stiff Dynamical Systems.** *AIChE Journal: Futures Issue,* 65 (12), (2019).
9. Shao, Y. and Scott J.K. **Convex relaxations for global optimization under uncertainty described by continuous random variables**, *AIChE Journal*, (2018): 3023 – 3033.
10. Song, Y.; Cao, H.; Mehta, C.; and Khan K.A.. **Bounding Convex Relaxations of Process Models from Below by Tractable Black-Box Sampling**, *Computers & Chemical Engineering*, In Press, (2021).

# Original EAGO Capabilities

## Extendable Support for Script Defined Functions

*Parameter Estimation for Nonideal Two-Liquid Mixture*

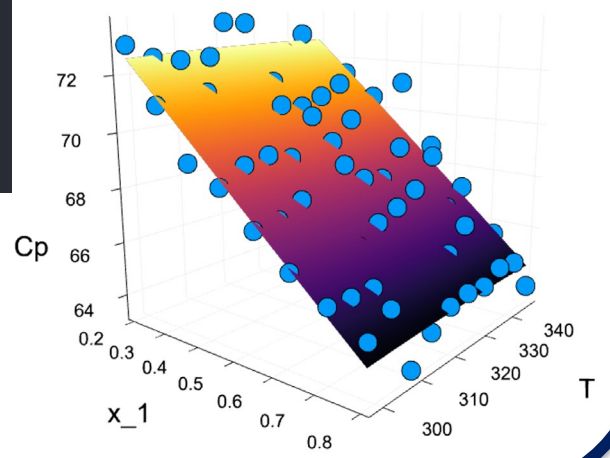
```
exGibbs(T,x1,p) = R*T*(x1*(1-x1)^2*(p[1]*T+p[2]*T^2+p[3]*log(T))+
                       (1-x1)*x1^2*(p[1]*T+p[2]*T^2+p[3]*log(T)))
GibbsA(T) = CpA*(T-T0)-T*CpA*log(T/T0)
GibbsW(T) = CpW*(T-T0)-T*CpW*log(T/T0)
Gibbs(T,x1,p) = x1*GibbsA(T)+(1-x1)*GibbsW(T)+
                R*T*(x1*log(x1)+(1-x1)*log(1-x1))+exGibbs(T,x1,p)
Cp(T,x1,p) = -T*ForwardDiff.derivative(T->ForwardDiff.derivative(T->Gibbs(T,x1,p),T),T)
```

```
function objective(T::Vector,x1::Vector,Cp_exp::Matrix,p...)
    SSE = 0.0
    for i = 1:length(T)
        for j = 1:length(x1)
            SSE += (Cp(T[i],x1[j],p)-Cp_exp[i,j])^2
        end
    end
    return SSE
end
```

```
fobj(p...) = objective(Tdata,x1data,Cp_exp,p...)
```

$$\min_{\mathbf{p}\in P} \sum_i \left(C_P^{mod}(T_i, x_i, \mathbf{p}) - C_P^{exp}(T_i, x_i)\right)$$

$$C_P^{mod}(T_i, x_i, \mathbf{p}) = -T_i \left.\frac{\partial^2 G}{\partial T^2}\right|_P (T_i, x_i, \mathbf{p})$$

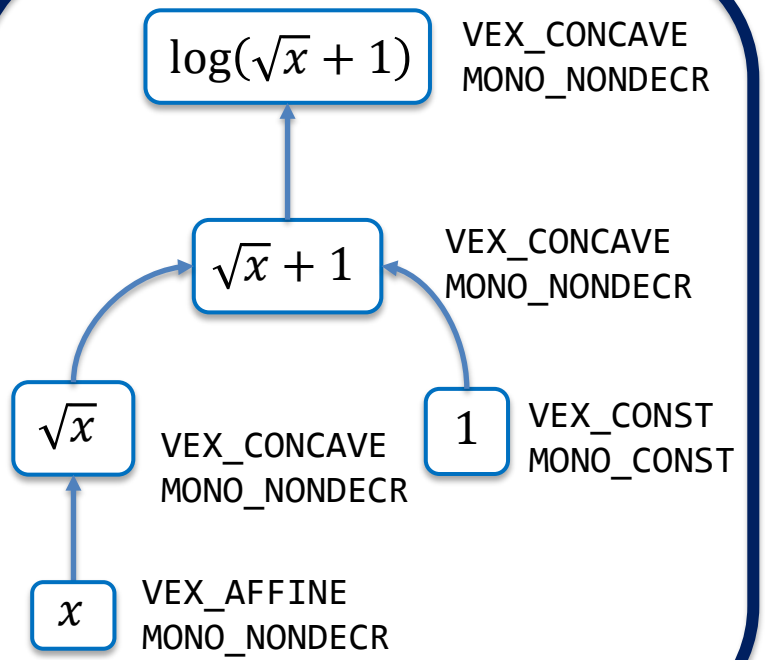



## Performant Subroutines

- Nonconvex MINLP & SIP solver
- Bounds Tightening Routines:
  - Optimization-based
  - Feasibility-based
- Preprocessing Routines:
  - Algebraic Rearrangements
  - Subexpression Elimination
  - Regular Problem Classification

# Additional Features

## Built-in Convexity Detection (Disciplined Convex Programming[11], etc.)

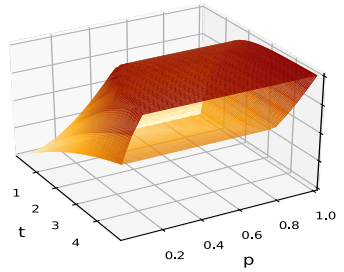



## Global Dynamic Optimization

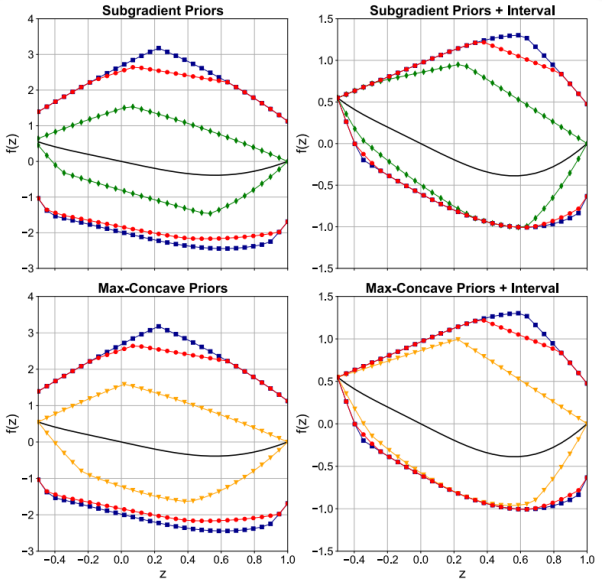
**DynamicBounds.jl[12]**

$$\frac{dx}{dt} = \exp(p)\sin(x)(2-x),$$

$$x(0) = 1, \quad p \in [0.01,1], \quad t \in [0,5]$$




## Improved Composite Bilinear Relaxations




McCormick Relaxation[5]
Multivariate McCormick Relaxation[13]
A priori Calculation (Subgradient)
A priori Calculation (Max-Concave)


5. Mitsos, A., et al. **McCormick-based relaxations of algorithms.** *SIAM Journal on Optimization*, SIAM (2009) 20, 73-601.
11. Grant, M., Boyd, S., & Ye, Y. (2006). **Disciplined convex programming.** In *Global optimization* (pp. 155-210). Springer, Boston, MA.
12. Wilhelm, M. E., **DynamicBounds.jl**, (2020), GitHub repository, https://github.com/PSORLab/DynamicBounds.jl
13. Tsoukalas, A., and Mitsos, A. Multivariate McCormick relaxations. Journal of Global Optimization 59.2-3 (2014): 633-662.

# Extensibility

**Seamless Integration with Defaults**

- Any function can be replaced with a user-defined routine
- EAGO automatically incorporates new routines with default operation

Applications:

- Test new bounding methods
- Solve unique problem types
- Modify existing methods
- […]

$$\min_{\mathbf{x}\in X} \quad \sin(x_1)x_2^2 - \cos(x_3)/x_4$$

In [1]:
```
using EAGO, IntervalArithmetic
```

In [2]:
```
struct IntervalExt <: EAGO.ExtensionType end
```

In [3]:
```
import EAGO: lower_problem!
function lower_problem!(t::IntervalExt, x::EAGO.GlobalOptimizer)
    # Retrieve bounds at current node
    n = x._current_node
    lower = n.lower_variable_bounds
    upper = n.upper_variable_bounds

    # Define X for the node and compute the interval extension
    x_value = Interval.(lower, upper)
    F = sin(x_value[1])*x_value[2]^2-cos(x_value[3])/x_value[4]
    x._lower_objective_value = F.lo
    x._lower_solution = IntervalArithmetic.mid.(x_value)
    x._lower_feasibility = true
    return
end
```

# What's New in EAGO

**Documentation**

**Improvements to McCormick.jl**

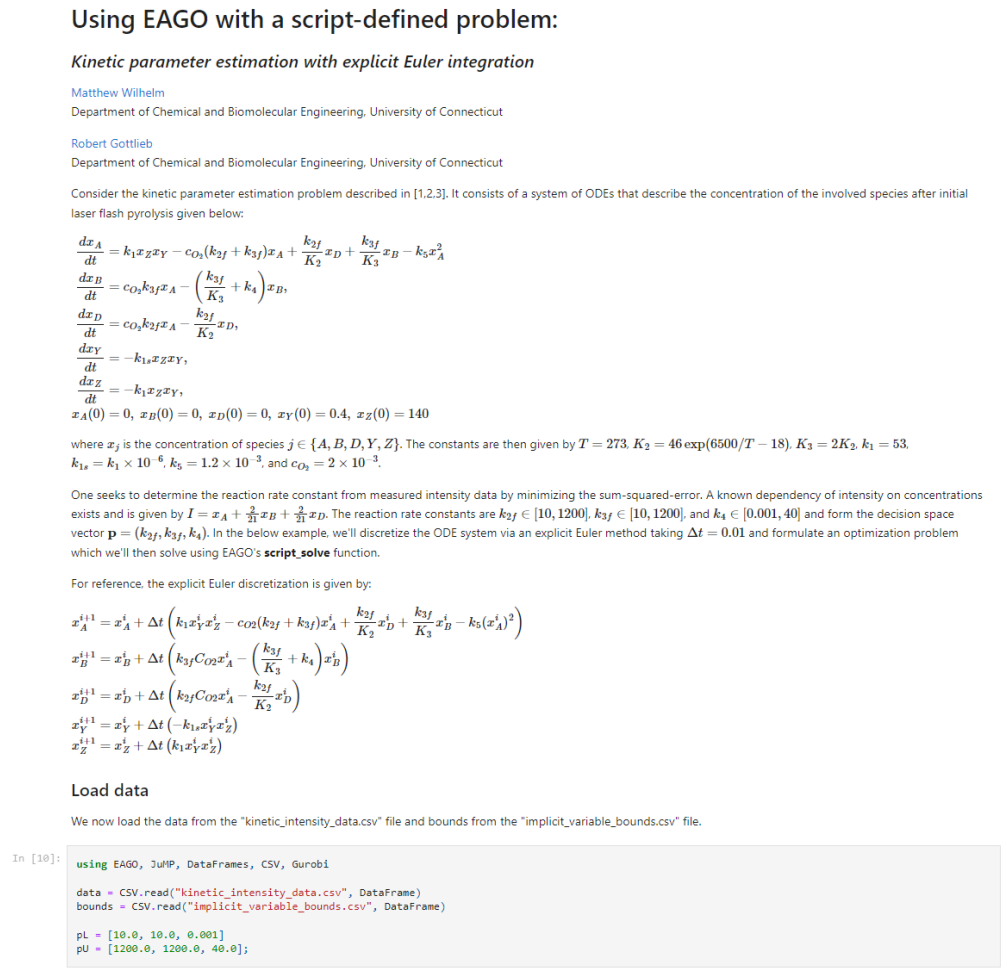
**Parallelization**

**Maintenance**

# Documentation

**Updates in progress:**

- Updated Jupyter notebook examples: https://github.com/PSORLab/EAGO-notebooks
- Overhaul of documentation website
  - Quick-start guide
  - Use examples with varying complexity
  - Guide to creating extensions
- New docstrings for all user-facing functions
- Fixes for all incomplete and outdated docstrings



Using EAGO with a script-defined problem:

*Kinetic parameter estimation with explicit Euler integration*

Matthew Wilhelm
Department of Chemical and Biomolecular Engineering, University of Connecticut

Robert Gottlieb
Department of Chemical and Biomolecular Engineering, University of Connecticut

Consider the kinetic parameter estimation problem described in [1,2,3]. It consists of a system of ODEs that describe the concentration of the involved species after initial laser flash pyrolysis given below:

$$\frac{dx_A}{dt} = k_1 x_Z x_Y - c_{O_2}(k_{2f} + k_{3f})x_A + \frac{k_{2f}}{K_2}x_D + \frac{k_{3f}}{K_3}x_B - k_5 x_A^2$$
$$\frac{dx_B}{dt} = c_{O_2}k_{3f}x_A - \left(\frac{k_{3f}}{K_3} + k_4\right)x_B,$$
$$\frac{dx_D}{dt} = c_{O_2}k_{2f}x_A - \frac{k_{2f}}{K_2}x_D,$$
$$\frac{dx_Y}{dt} = -k_{1s}x_Z x_Y,$$
$$\frac{dx_Z}{dt} = -k_1 x_Z x_Y,$$
$$x_A(0) = 0,\ x_B(0) = 0,\ x_D(0) = 0,\ x_Y(0) = 0.4,\ x_Z(0) = 140$$

where $x_j$ is the concentration of species $j \in \{A, B, D, Y, Z\}$. The constants are then given by $T = 273$, $K_2 = 46\exp(6500/T - 18)$, $K_3 = 2K_2$, $k_1 = 53$, $k_{1s} = k_1 \times 10^{-6}$, $k_5 = 1.2 \times 10^{-3}$, and $c_{O_2} = 2 \times 10^{-3}$.

One seeks to determine the reaction rate constant from measured intensity data by minimizing the sum-squared-error. A known dependency of intensity on concentrations exists and is given by $I = x_A + \frac{2}{21}x_B + \frac{2}{21}x_D$. The reaction rate constants are $k_{2f} \in [10, 1200]$, $k_{3f} \in [10, 1200]$, and $k_4 \in [0.001, 40]$ and form the decision space vector $\mathbf{p} = (k_{2f}, k_{3f}, k_4)$. In the below example, we'll discretize the ODE system via an explicit Euler method taking $\Delta t = 0.01$ and formulate an optimization problem which we'll then solve using EAGO's **script_solve** function.

For reference, the explicit Euler discretization is given by:

$$x_A^{i+1} = x_A^i + \Delta t\left(k_1 x_Y^i x_Z^i - c_{O2}(k_{2f} + k_{3f})x_A^i + \frac{k_{2f}}{K_2}x_D^i + \frac{k_{3f}}{K_3}x_B^i - k_5(x_A^i)^2\right)$$
$$x_B^{i+1} = x_B^i + \Delta t\left(k_{3f}C_{O2}x_A^i - \left(\frac{k_{3f}}{K_3} + k_4\right)x_B^i\right)$$
$$x_D^{i+1} = x_D^i + \Delta t\left(k_{2f}C_{O2}x_A^i - \frac{k_{2f}}{K_2}x_D^i\right)$$
$$x_Y^{i+1} = x_Y^i + \Delta t\left(-k_{1s}x_Y^i x_Z^i\right)$$
$$x_Z^{i+1} = x_Z^i + \Delta t\left(k_1 x_Y^i x_Z^i\right)$$

Load data

We now load the data from the "kinetic_intensity_data.csv" file and bounds from the "implicit_variable_bounds.csv" file.

In [10]:

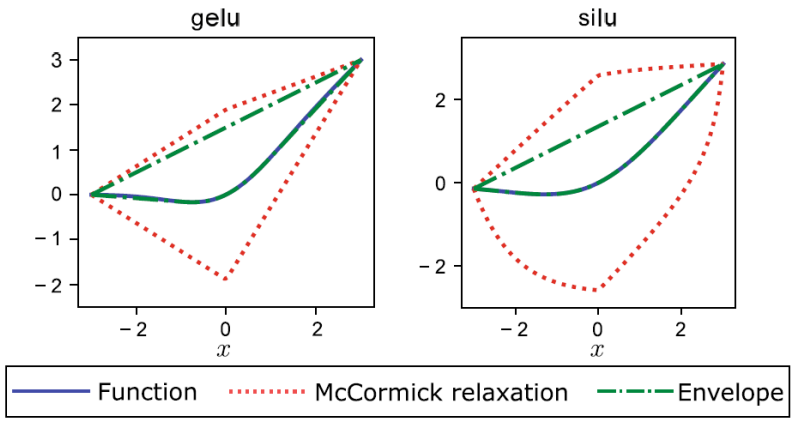
```
using EAGO, JuMP, DataFrames, CSV, Gurobi

data = CSV.read("kinetic_intensity_data.csv", DataFrame)
bounds = CSV.read("implicit_variable_bounds.csv", DataFrame)

pL = [10.0, 10.0, 0.001]
pU = [1200.0, 1200.0, 40.0];
```

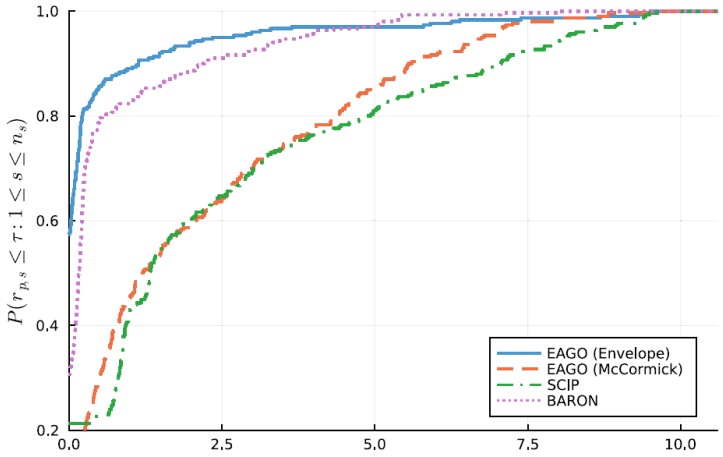
# Improvements to McCormick Relaxation Library

**New Publication:**

Wilhelm, M.E., Wang, C., Stuber, M.D. **Convex and concave envelopes of artificial neural network activation functions for deterministic global optimization.** *Journal of Global Optimization* (2022)

- Tighter relaxations of many common ANN activation functions
- Implemented within McCormick.jl
- Substantial speedup compared to naïve relaxations
- Enables global optimization with embedded ANN surrogate models

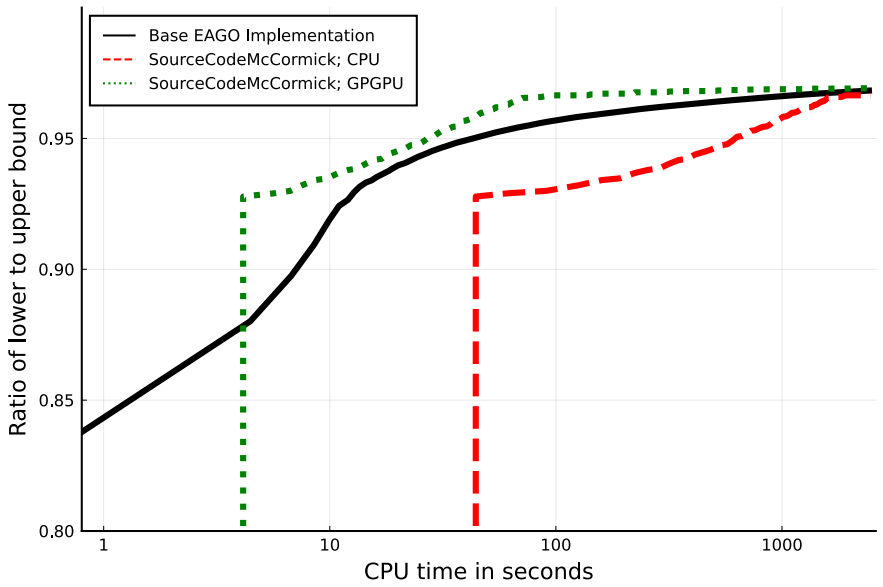


# Parallelization Efforts

**Behind-the-scenes development:**

- First deterministic global optimization algorithm to be implemented on a GPU
- Packages that extend EAGO to parallelize some aspects of B&B
- Come see us at FOCAPO / CPC 2023!



# Maintenance: JuMP and MOI

**Current Status**


v1.1.1

Nonlinear functionality handled by **JuMP**

**Update in Progress**


v1.4.0

Nonlinear functionality handled by **MathOptInterface** abstraction layer

# Conclusions

**EAGO — An extensible deterministic global optimizer**

- High performance solver
- Designed for user-defined functions and routines
- Open-source and free for non-commercial use

Process Systems and Operations Research Laboratory

EAGO

julia

**Future Outlook**

- Parallel computing capability
- Regular updates to promote ease of use
- Performance improvements in core EAGO algorithms
- Exploring compiled versions for use with GAMS

# 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.

# Questions?

Process Systems and Operations Research Laboratory

https://www.psor.uconn.edu

EAGO

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