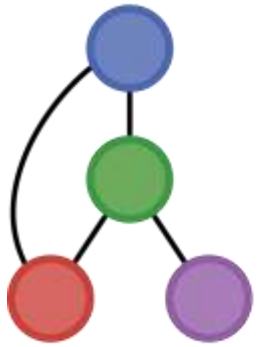
Automated Symbolic Thermodynamic Modeling and Global Parameter Estimation for Brine Electrolyte Systems

Presented by Pengfei Xu

Pengfei Xu, Robert Gottlieb, Wajeha Tauqir, George Bollas, and Matthew D. Stuber, Pratt & Whitney Associate Professor

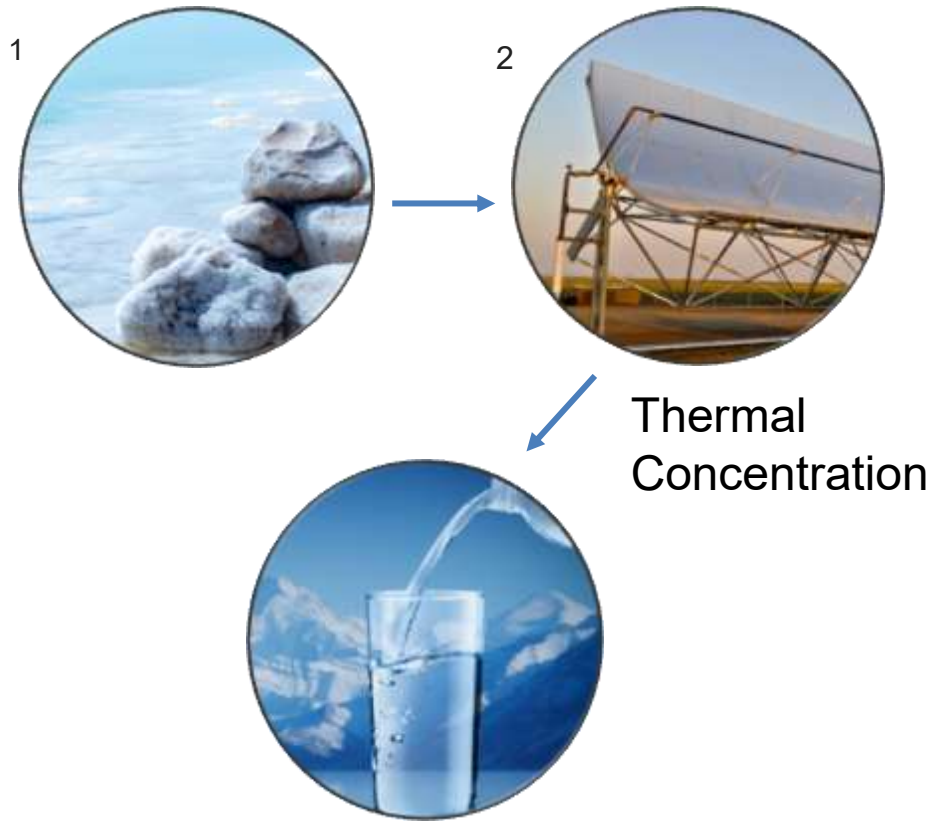
2025, Nov 5th

2025 AIChE ANNUAL MEETING



Process Systems and Operations Research Laboratory

Motivation

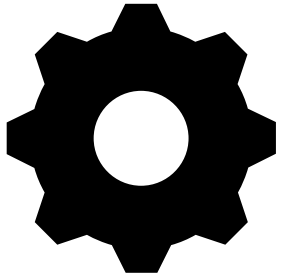


- Thermal brine separation is of critical importance to many industries with brine effluent streams and/or brine concentration needs (e.g., agriculture, power production, mining)
  - Increase sustainability
  - Reduce costs
  - Improve system robustness
- We need accurate mechanistic thermodynamics models that can enable:
  - Dynamic simulation
  - Optimization-based design of water treatment technologies

Accurate models enable efficient brine treatment.

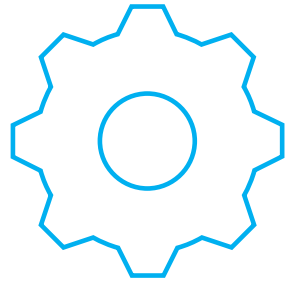
[1] Molinari, Raffaele., et al. **"Can brine from seawater des**
[2] Stuber, Matthew D., et al. **"Pilot demonstration of cond** **es."** *Desalination* 355 (2015): 186-196.

Mechanistic vs Empirical Model



Empirical Model

- Fast & simple in mathematics
- System specific
- Data-Driven



Mechanistic Model

r-eNRTL as an example

- Physically rigorous
- Thermodynamically consistent

- Empirical models are fast, reliable, and easy to apply within known systems.
- However, they are system-specific and lack thermodynamic consistency across mixtures.

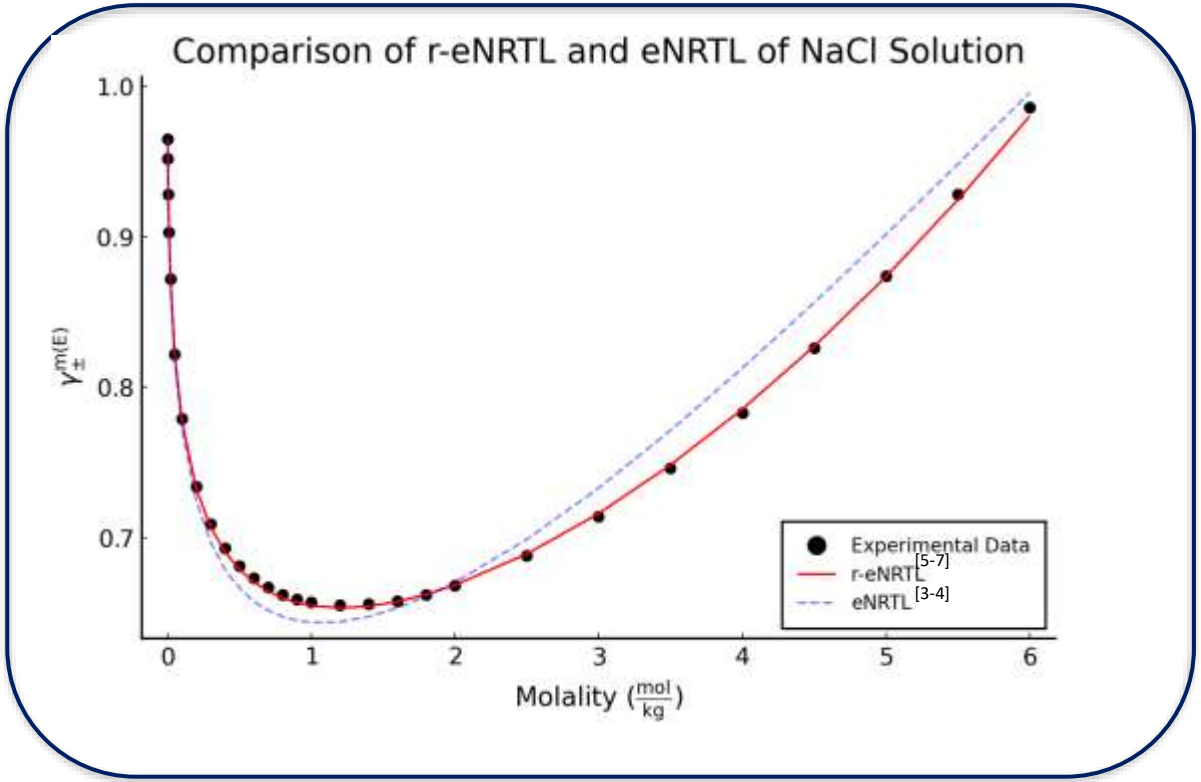
Goal

Goal: Modeling Thermal Brine Separation System

- **Capability:** Predicting **nonideal behaviors** across
  - **multi-ions**
  - **high concentration**
  - **variable temperature.**
  - **Transferrable between different platforms**

- **Method: Achieve** deterministic global optimal **parameter estimation** for the r-eNRTL model.

Thermodynamic Model



r-eNRTL accurately captures non-ideal electrolyte behavior across concentration and temperature.

Consistent for deriving enthalpy, entropy, and heat capacity (i.e., a first principal model)

However, this accuracy comes with a price:

Challenge: Algebraically complex → hard to differentiate/optimize

[3] Chen, C.C., Britt, H.I., J. F. Boston, and L. B. Evans. "Local composition model for excess Gibbs energy of electrolyte systems. Part I: Single solvent, single completely dissociated electrolyte systems." *AIChE Journal* 28, no. 4 (1982): 588-596.
[4] Chen, C. C., & Evans, L. B.. A local composition model for the excess Gibbs energy of aqueous electrolyte systems. *AIChE journal*, *32*(3) (1986), 444-454.
[5] Yang Xi, Barton, P. I. , & Bollas, G. M.. The significance of frameworks in electrolyte thermodynamic model development. (2020)
[6] Yang Xi, Barton, P. I. , & Bollas, G. M.. Refined electrolyte-NRTL model: Inclusion of hydration for the detailed description of electrolyte solutions. Part I. Single electrolytes up to moderate concentrations, single salts up to the solubility limit. (2020).
[7] Yang Xi, Barton, P. I. , Chen, C. C., & Bollas, G. M.. Refined electrolyte-NRTL model: Inclusion of hydration for the detailed description of electrolyte solutions. Part II. Detailed simulation of speciation and activities of aqueous solutions of acids and bases. (2020).

Complexity of Refined eNRTL

$$\frac{G^{\mathrm{SR}}}{RT} = \sum_{j=1}^{n_m} X_{m_j} \left(\frac{\sum_{s\in\{m,a,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{m_j,s_l,m_j} \tau_{m_j,s_l,m_j}}{\sum_{s\in\{m,a,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{m_j,s_l,m_j}} \right)$$
$$+ \sum_{j=1}^{n_a} X_{a_j} \sum_{k=1}^{n_c} \left(\frac{X_{c_k}}{\sum_{k'=1}^{n_c} X_{c_{k'}}} \right) \left(\frac{\sum_{s\in\{m,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{a_j,s_l,c_k} \tau_{a_j,s_l,c_k}}{\sum_{s\in\{m,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{a_j,s_l,c_k}} \right)$$
$$+ \sum_{j=1}^{n_c} X_{c_j} \sum_{k=1}^{n_a} \left(\frac{X_{a_k}}{\sum_{k'=1}^{n_a} X_{a_{k'}}} \right) \left(\frac{\sum_{s\in\{m,a\}} \sum_{l=1}^{n_s} X_{s_l} F_{c_j,s_l,a_k} \tau_{c_j,s_l,a_k}}{\sum_{s\in\{m,a\}} \sum_{l=1}^{n_s} X_{s_l} F_{c_j,s_l,a_k}} \right)$$

r-eNRTL is accurate but algebraically intense.

$$\log \gamma_{t_j}^{\mathrm{SR}} = \frac{\partial}{\partial N_{t_j}} \left(\sum_{\hat{t}\in\{m,a,c\}} \sum_{\hat{j}=1}^{n_{\hat{t}}} N_{\hat{t}_{\hat{j}}} \frac{G^{\mathrm{SR}}}{RT} \right)$$

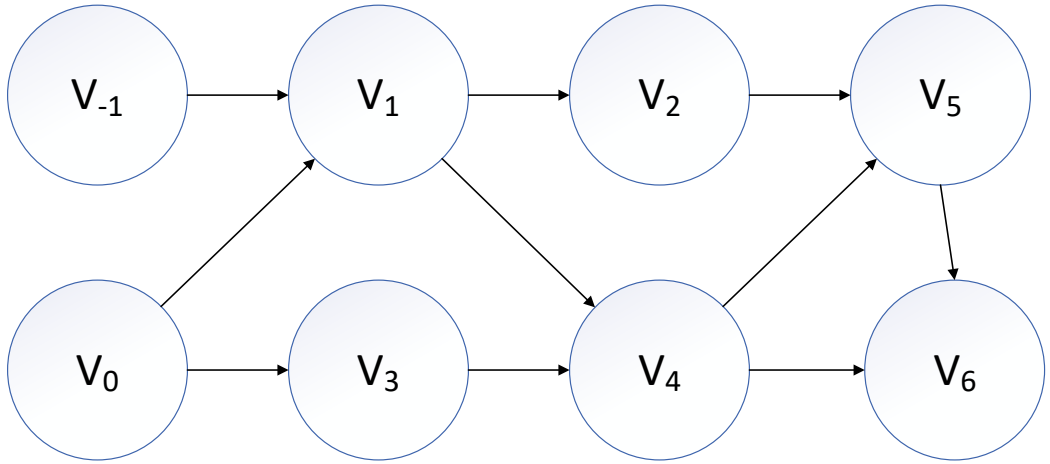
AD

Hundreds of nested terms after differentiation

[8] Gottlieb, Robert X., et al. "**Automatic Source Code Generation of Complicated Models For Deterministic Global Optimization With Parallel Architectures**."

How AD Works

- **Operator Overloading, AD**



$$y = \left[\sin(\frac{x_1}{x_2}) + \frac{x_1}{x_2} - \exp(x_2)\right] * \left[\frac{x_1}{x_2} - \exp(x_2)\right]$$ [9]

*Dot here is Derivative w.r.t. v_0

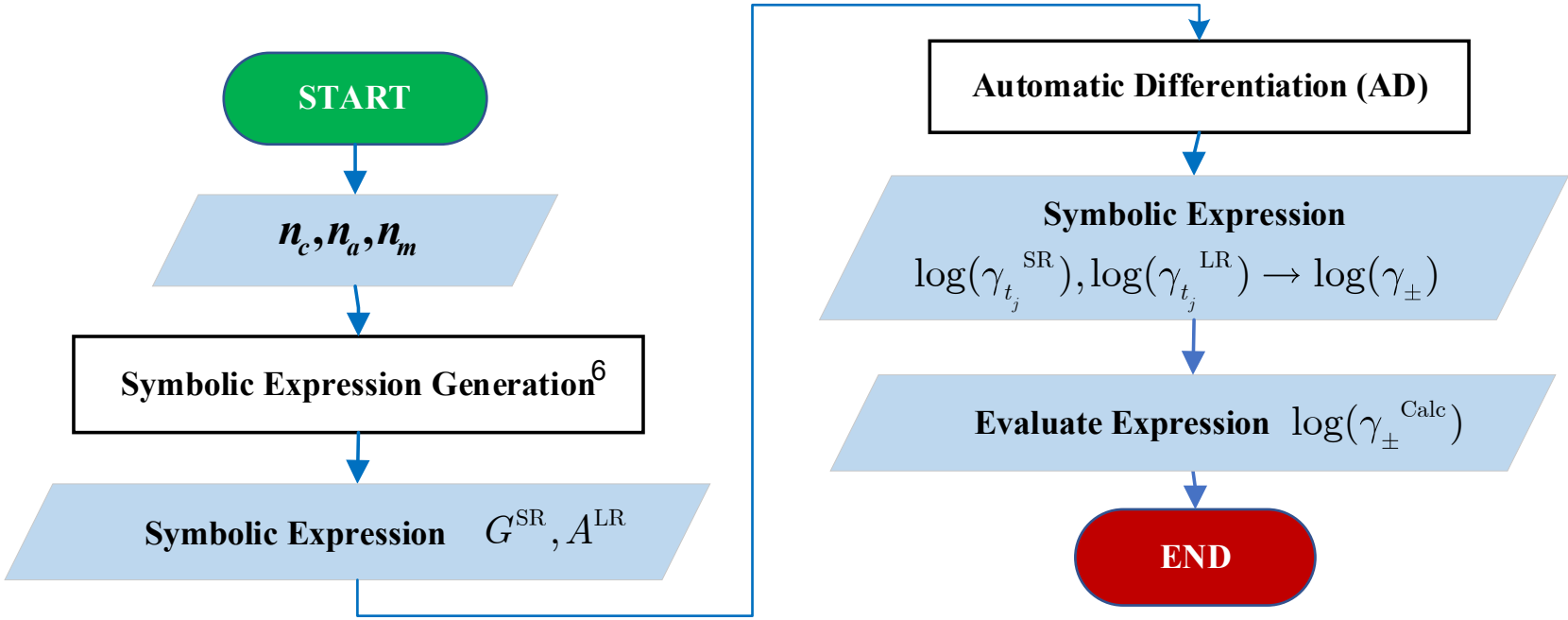
$v_{-1} = x_1$	$\dot{v}_{-1} = 0$
$v_0 = x_2$	$\dot{v}_0 = 1$
$v_1 = v_{-1} \,/\, v_0$	$\dot{v}_1 = -v_{-1} \,/\, v_0^2$
$v_2 = \sin(v_1)$	$\dot{v}_2 = \dot{v}_1 \cos(v_1)$
$v_3 = \exp(v_0)$	$\dot{v}_3 = \dot{v}_0 \exp(v_0)$
$v_4 = v_1 - v_3$	$\dot{v}_4 = \dot{v}_1 - \dot{v}_3$
$v_5 = v_2 + v_4$	$\dot{v}_5 = \dot{v}_2 + \dot{v}_4$
$v_6 = v_5 v_4$	$\dot{v}_6 = \dot{v}_5 v_4 + \dot{v}_4 v_5$
$y = v_6$	$\dot{y} = \dot{v}_6$

Each node stores both its value and derivative — gradients are computed automatically.

No manual differentiation required — AD handles everything.

[9] Griewank, A., & Walther, A.. *Evaluating derivatives: prin*
Society for industrial and applied mathematics. (2008)

Automatic Differentiation Workflow



γ_{t_j}: Activity coefficient of species t_j.

G^{SR}: Short range excess Gibbs free energy.

A^{LR}: Long range excess Helmholtz free energy.

[10] Gowda, S., et al. **"High-performance symbolic-numerics via multiple dispatch."** *ACM Communications in Computer Algebra* 55, no. 3 (2022): 92-96.

Challenge in Deterministic Global Parameter Estimation

- **Slow Evaluation**
  - r-eNRTL formulas are huge and deeply nested.
  - Hard to optimize directly.

Solution: Simplify expressions with auxiliary variables.

- **Slow Convergence**
  - Relaxations too loose for nonlinear exponentials.
  - Solver converges slowly.

Solution: Add tighter relaxations or constraints/cuts.

Both challenges limit deterministic global optimization efficiency

Research Focus: Simplifying r-eNRTL for Global Optimization

Research Focus

- **Complex Expression**
  - r-eNRTL formulas are very large and deeply nested.
  - Hard to optimize directly.

Solution: Simplify expressions with auxiliary variables.

Automatic Differentiation: Power and Pitfalls

- **Power:**
  - Enables automatic derivative generation for thermodynamic models
  - Provides accurate and portable expressions

- **Pitfall – Expression Swell [11]:**
  - Repeated differentiation makes expressions grow excessively large
  - Hard to simplify, slow to compute, and difficult to compile

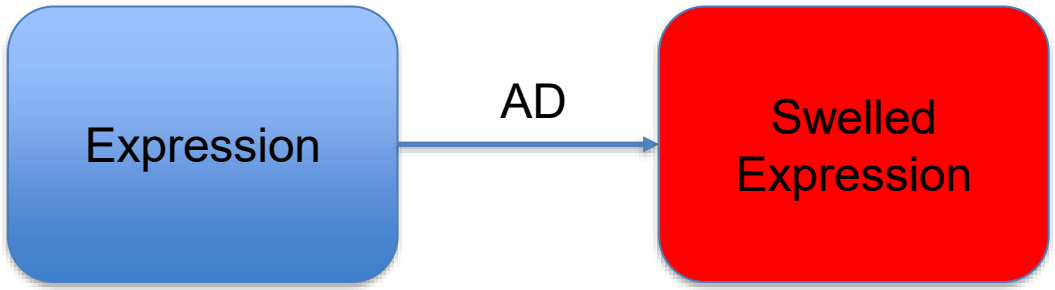
AD helps—but it introduces a new problem：

To address the expression swell problem and speed up global optimization, we developed an automated workflow.

[11] Laue, S. (2019). On the equivalence of automatic and symbolic differentiation. *arXiv preprint arXiv:1904.02990*.

Stage 1: Expression Swell During AD

AD expands all intermediate terms

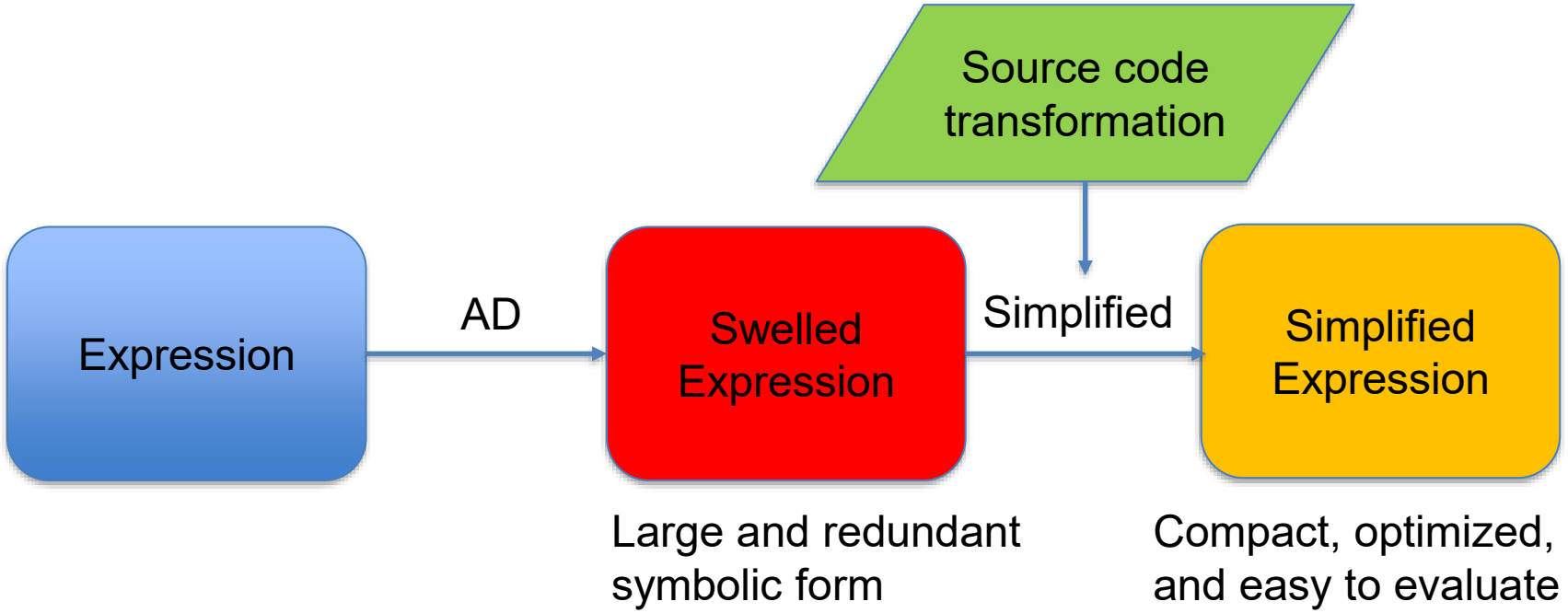


Large, redundant expression – slow and hard to optimize

Expression Swelling

Raw Symbolic Expression of Activity Coefficient of Single Electrolyte

Stage 2 – Simplified Expression Automatically



Simplified Expression Automatically

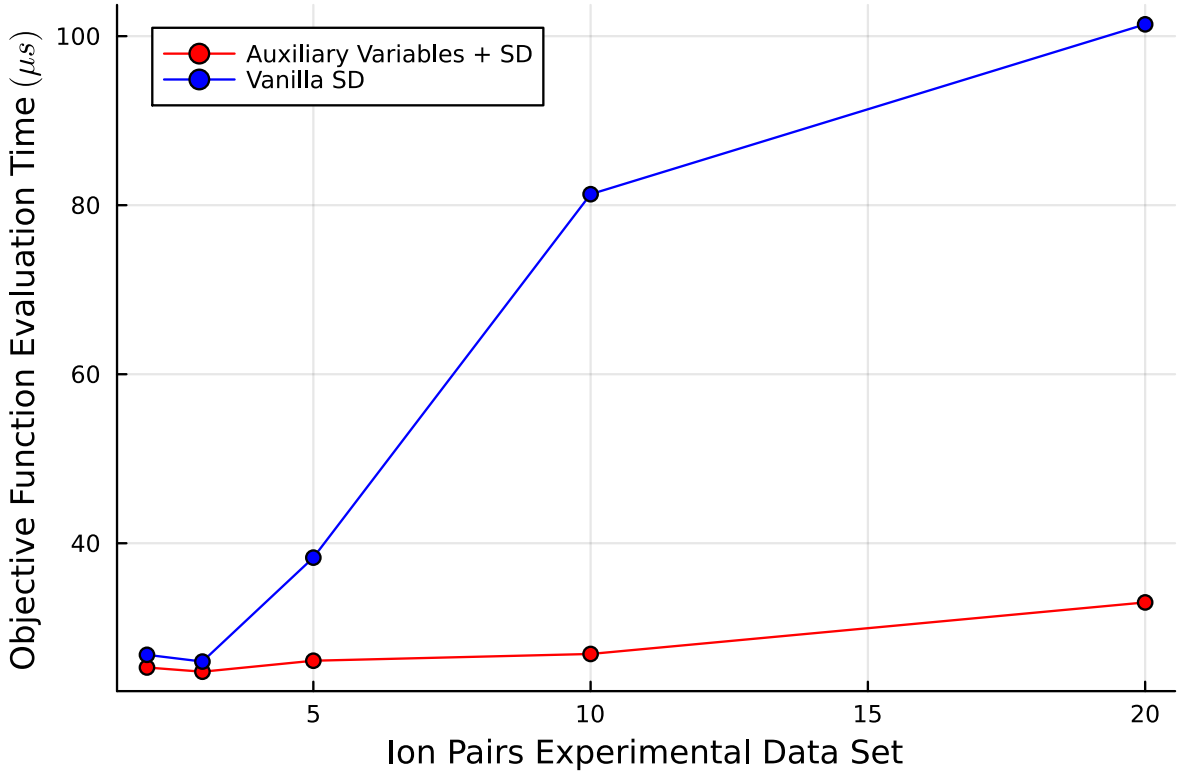
The Symbolic Expression of Activity Coefficient of NaCl after Auxiliary Variables Substitution

```
function gamma_fun1(Na1, Nc1, ha1, hc1, τM1C1A1, τC1A1M1)
    expC1A1M1 = exp(-0.2*τC1A1M1)
    expM1C1A1 = exp(-0.2*τM1C1A1)
    X5 = 55.5 + Na1 + Nc1  # PRE-CALC X5
    X6 =  - Z1 - Z2  # PRE-CALC X6
    X7 = ((Na1 + Nc1) / X5)  # PRE-CALC X7
    A, B, C, D = 1.061208, 3.2643568968539447, 5.929741, 3.2643568968539447  # Coefficient from the ma
    cbrt_sum = cbrt(A + B * (Z3)) + cbrt(C + D * (Z3))
    D1, D2, D3, D4, D5 = 0.0010028566098294884, 2.42e-5, 8.957027911831228e-6, -7.599999999999999e-6,
    sqrt_frac = sqrt((0.5 * (Na1 + 1.0^2*Nc1)) / (D1 + Na1 * (D2 + D3 * X7) + Nc1 * (D4 + D5 * X7)))
    X8 = 0.010392539429094808*sqrt_frac*(cbrt_sum)  # PRE-CALC X8
    X9 = 1.0 + X8
    X10 = (1 - X7)  # PRE-CALC X10
    X14 = X13^2  # PRE-CALC X14
    X15 = (X1)*τM1C1A1  # PRE-CALC X15
    X16 = -(Nc1+Na1)*τC1A1M1 / (X14)  # PRE-CALC X16
    X17 = -(Nc1+Na1)*τC1A1M1 / (X14 * expC1A1M1)  # PRE-CALC X17
    X19 = 1.0 - 2.0 * log(X9) + X8 + -1.0 / (X9)  # PRE-CALC X19
    X20 = (1 / cbrt_sum)^3.0  # PRE-CALC X20
    X21 = Nc1*τM1C1A1  # PRE-CALC X21
    X22 = 1.0  # PRE-CALC X22
    E1 = 1.736199415294409e-5  # Coefficient from the match
    r_term1 = E1*Nc1*X10 / X5
    E2 = 8.957027911831228e-6  # Coefficient from the match
    r_term2 = E2*Na1*X10 / X5
    X23 = ha1 + hc1  # PRE-CALC X23
```

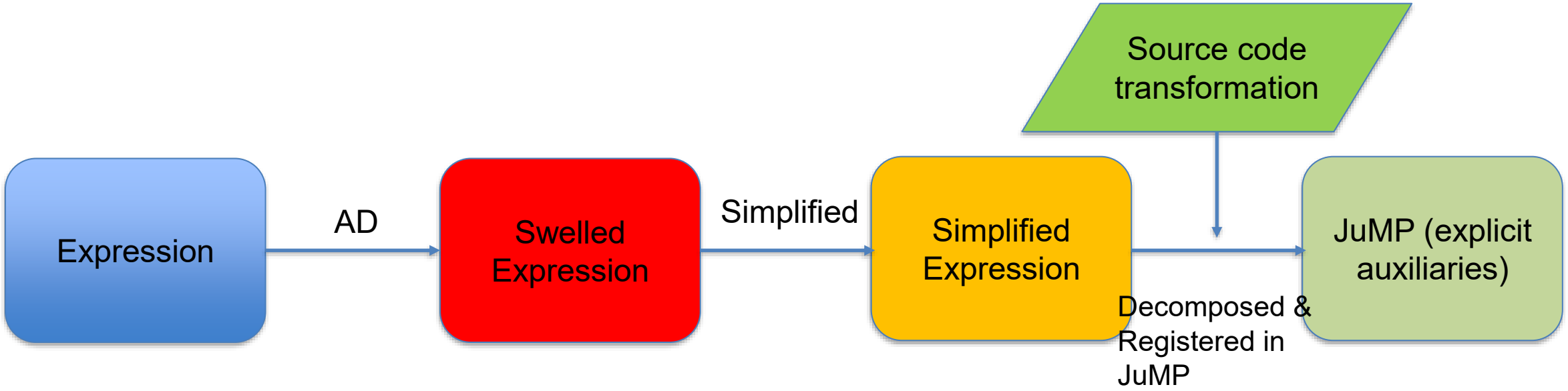
Automatically generated and simplified — ready to be linked with JuMP

Objective Function Evaluation Time Comparison

- Each data set = one electrolyte (NaCl, KCl, $MgCl_2$, $CaCl_2$, LiCl, etc.).
- Larger systems → longer evaluation time.
- **Our simplified expression keeps time almost constant.**



End-to-End Workflow: Symbolic → Simplified → JuMP



All subexpressions registered as JuMP vars → easy bounds & domain-safety

Automatically Generated JuMP Model

```
# 0) Experimental data (multiple NaCl samples)
Na_vec = Na1
Nc_vec = Nc1
@assert length(Na_vec) == length(Nc_vec) "Na_vec and Nc_vec must have the same length"
n = length(Na_vec)
obs_gamma = gammaExp1  # observed γ_pm

# ----------------------------
# 1) Modeling (NaCl has only one ion pair; ha corresponds to Na, hc corresponds to Cl)
factory = () -> EAGO.Optimizer(SubSolvers())
m = Model(optimizer_with_attributes(factory, "absolute_tolerance" => 1e-6,"time_limit" => 200.,"
```

Automatically generates JuMP code with all symbolic subexpressions and variables.

```
@variable(m, 0.0<=ha_Cl<=0.6)     # hc[Cl]
@variable(m, 7.0<=τM<=9.0)        # τM1C1A1[NaCl]
@variable(m, -5.0<=τC<=-3.5)        # τC1A1M1[NaCl]

JuMP.fix(ha_Cl,0.5; force=true)

@constraint(m, hc_Na - ha_Cl >= 0)  # Physical constraint: hc_Na ≥ ha_Cl

# Pack data into parameter expressions for reuse in expressions
@expression(m, Na[i=1:n], Na_vec[i])
@expression(m, Nc[i=1:n], Nc_vec[i])
```

Parameter Estimation Problem

Objective: Fit model parameters to experimental activity data.

$$\boldsymbol{\tau}^*, \mathbf{h}^* \in \arg\min_{\substack{\boldsymbol{\tau}\in\mathcal{T}\subset\mathbb{R}^{n_\tau} \\ \mathbf{h}\in\mathcal{H}\subset\mathbb{R}^{n_h}}} f^*(\mathbf{N}_{t_j}, \mathbf{P}, \boldsymbol{\tau}, \mathbf{h}) = \sum_{i=1}^{n}\left[\left(\frac{\gamma_i(\mathbf{N}_{t_j}, \mathbf{P}, \boldsymbol{\tau}, \mathbf{h})}{\gamma_i^{\exp}} - 1\right)^2 + \left(\frac{\phi_i(\mathbf{N}_{t_j}, \mathbf{P}, \boldsymbol{\tau}, \mathbf{h})}{\phi_i^{\exp}} - 1\right)^2\right]$$

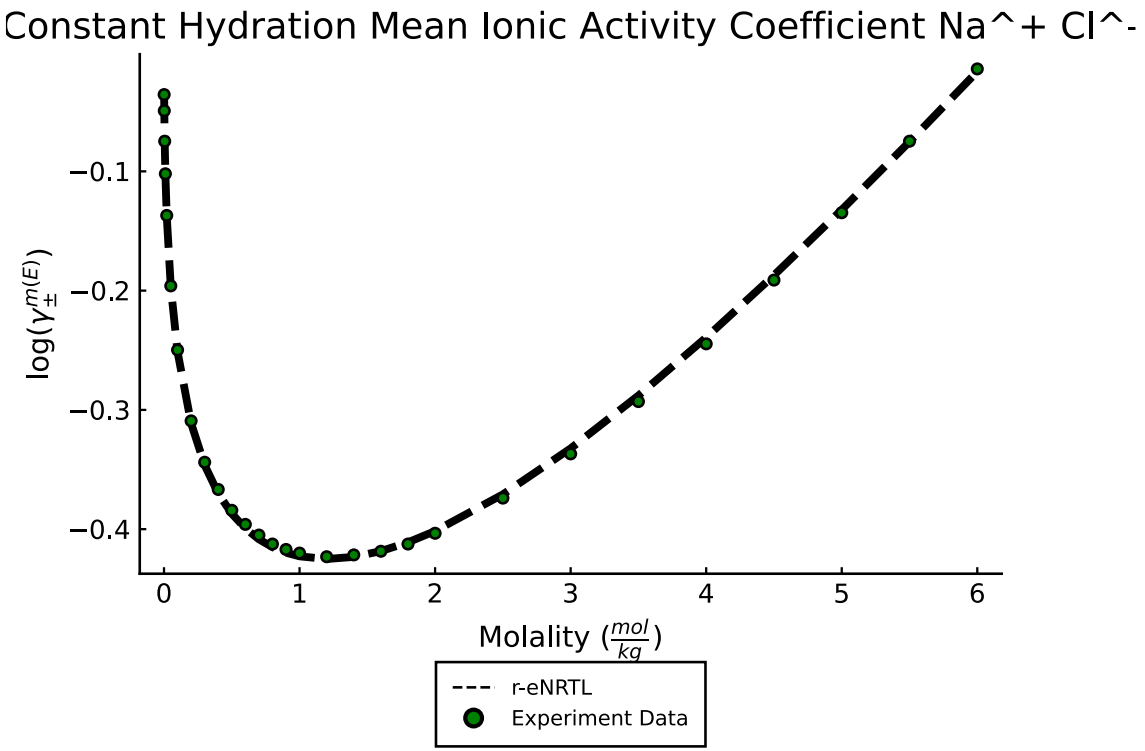
$$\text{s.t.} \quad \mathbf{h}(\boldsymbol{\tau}, \mathbf{h}, \mathbf{N}_{t_j}, \mathbf{P}) = 0,$$

$$\mathbf{g}(\boldsymbol{\tau}, \mathbf{h}, \mathbf{N}_{t_j}, \mathbf{P}) \le 0,$$

$$\boldsymbol{\tau}^{\mathrm{LB}} \le \boldsymbol{\tau} \le \boldsymbol{\tau}^{\mathrm{UB}},$$

$$\mathbf{h}^{\mathrm{LB}} \le \mathbf{h} \le \mathbf{h}^{\mathrm{UB}}.$$

Transfer Activity Coefficient Function in JuMP



After Simplification:
Around 100 seconds

```
Empty Stack: Exhaustive Search Finished
Optimal Solution Found at Node 183
Lower Bound: 3.977442239549105e-6
Upper Bound: 4.987392747813116e-6
Solution:
   hc_Na = 1.8021924307742863
   ha_Cl = 0.5
      τM = 7.834168160307493
      τC = -3.9071730716112896
     obj = 0.0

 99.707693 seconds (1.67 G allocations: 97.379 GiB
```

Before Simplification:
Around 600 seconds

```
Absolute Tolerance Achieved
First Solution Found at Node 1029
LBD = 0.0
UBD = 5.977314675518958e-6
Solution is:
    X[1] = 7.834167504201094
    X[2] = -3.907172657309239
    X[3] = 1.5789380575987575
    X[4] = 0.723255866122256
```

The first time r-eNRTL model solved in global optimality*

Next Step – Toward Multi-Electrolyte Parameter Estimation

Building on the NaCl case, the framework can generalize to multi-electrolyte systems.

Formulation of the Multi-Electrolyte Problem

$$\tau^{\mathrm{C},*}, \mathbf{h}^{*} \in \underset{\substack{\tau^{\mathrm{C}} \in \mathbf{T}^{\mathrm{C}} \subset \mathbb{R}^{n_{\tau^{\mathrm{C}}}} \\ \mathbf{h} \in \mathbf{H} \subset \mathbb{R}^{n_h}}}{\arg\min}\ f^{*}(\mathbf{N}_{\mathbf{t}_j}, \mathbf{P}, \tau^{\mathrm{C}}, \mathbf{h}) = \sum_{i=1}^{n} \left[\left(\frac{\gamma_i(\mathbf{N}_{\mathbf{t}_j}, \mathbf{P}, \tau^{\mathrm{C}}, \mathbf{h})}{\gamma_i^{\exp}} - 1 \right)^2 + \left(\frac{\phi_i(\mathbf{N}_{\mathbf{t}_j}, \mathbf{P}, \tau^{\mathrm{C}}, \mathbf{h})}{\phi_i^{\exp}} - 1 \right)^2 \right]$$

$$\text{s.t.} \quad \mathbf{h}(\tau, \mathbf{h}, \mathbf{N}_{\mathbf{t}_j}, \mathbf{P}) = \mathbf{0}$$

$$\mathbf{g}(\tau, \mathbf{h}, \mathbf{N}_{\mathbf{t}_j}, \mathbf{P}) \leq \mathbf{0}$$

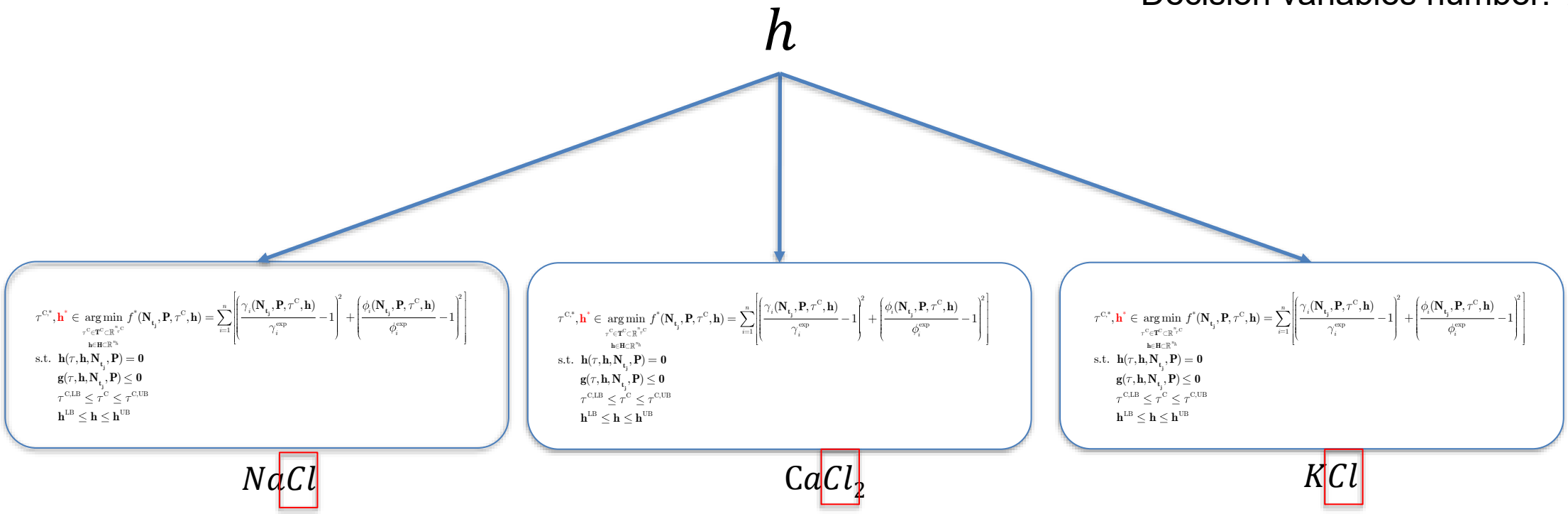
$$\tau^{\mathrm{C,LB}} \leq \tau^{\mathrm{C}} \leq \tau^{\mathrm{C,UB}}$$

$$\mathbf{h}^{\mathrm{LB}} \leq \mathbf{h} \leq \mathbf{h}^{\mathrm{UB}}$$

Parameters shared between ion sets, which causes the dimensionality problem!

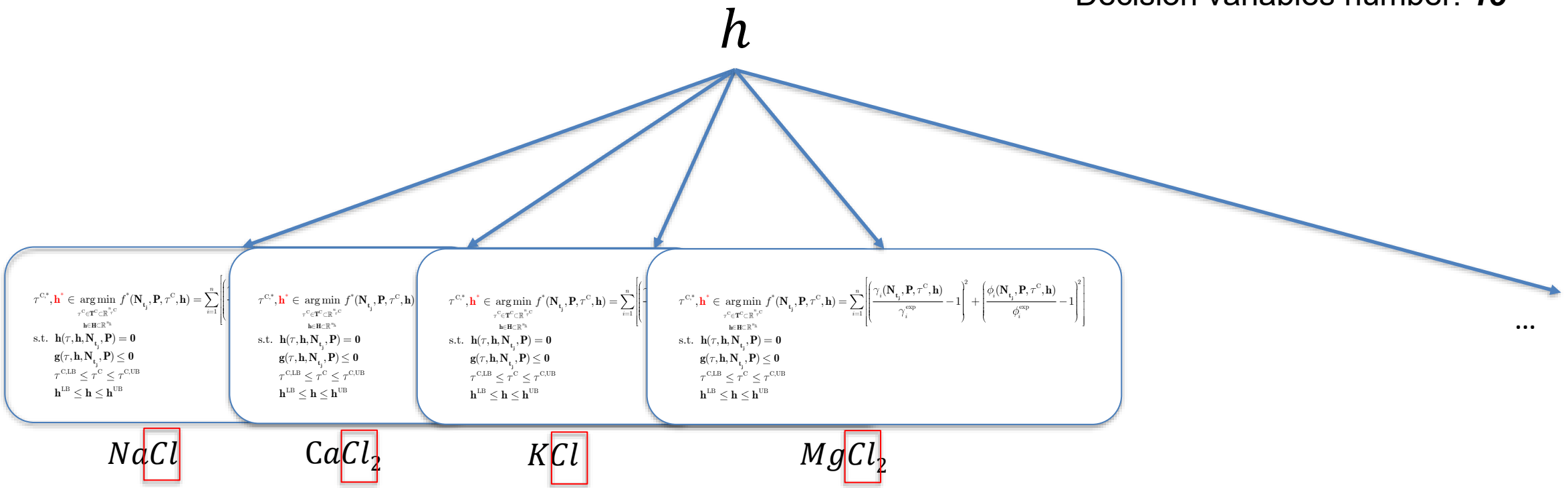
Why Dimensionality Matters – Shared Ion Parameters

Decision variables number: ***10***



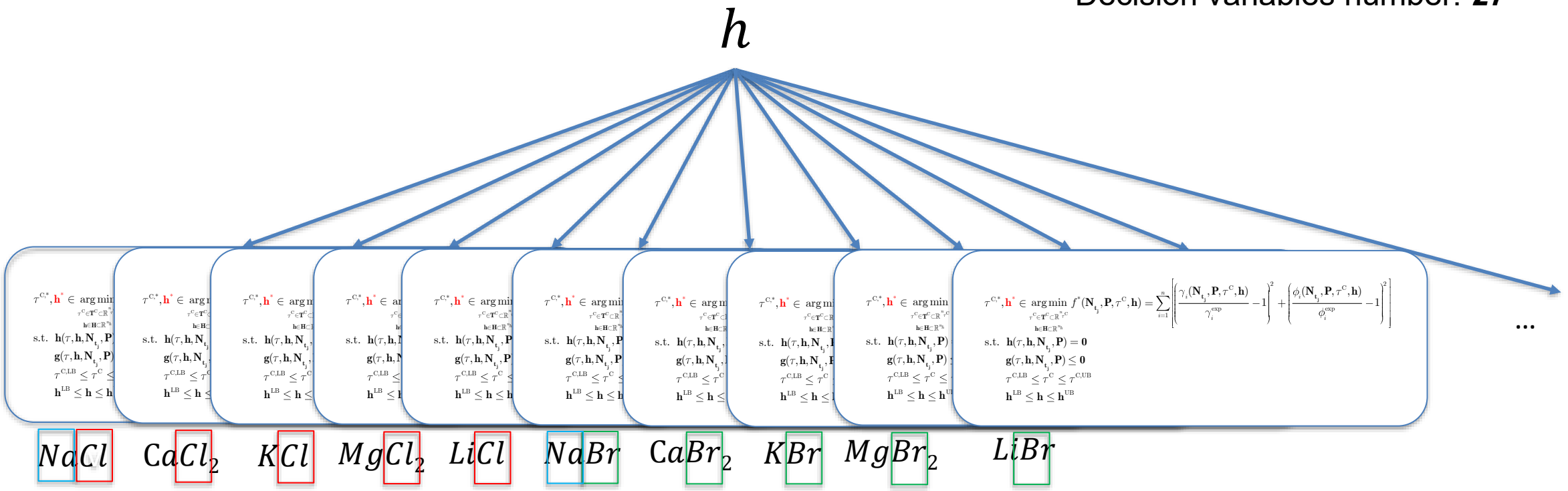
Scaling Up – Growth of Coupled Systems

Decision variables number: ***13***



Dimensionality Explosion in Multi-Electrolyte Optimization

Decision variables number: ***27***



Scaling Challenge in Multi-Electrolyte Parameter Estimation

- **5-electrolyte fitting (NaCl, KCl, $MgCl_2$, $CaCl_2$, LiCl):**
- 16 variables, 21 inequality constraints
- Converged in 25 min
- *Solver reached an early feasible solution within tolerance (abs tol = 0.1)*

```
Absolute Tolerance Achieved
First Solution Found at Node 1
LBD = 0.0
UBD = 0.053824729075278355
Solution is:
    X[1] = 8.452748324391644
    X[2] = -4.2504869814290815
    X[3] = 8.456633626574114
    X[4] = -4.250486981429058
    X[5] = 8.452627512151109
    X[6] = -4.250486981429081
    X[7] = 8.906709770849599
    X[8] = -4.583419233447454
    X[9] = 8.718299349191456
    X[10] = -4.390710964767369
    X[11] = 1.137714669045336
    X[12] = 4.939747886388841
    X[13] = 0.0002622026791630516
    X[14] = 1.7377654354619196
    X[15] = 5.4369243548955355
    X[16] = 0.17523358304442113
```

[12] Robinson, R. A., & Stokes, R. H. (2002). *Electrolyte solutions*. Courier Corporation.

Numerical Insights – Relaxation Effects

- Feasible and stable solutions across all mixtures
- Upper bounds remain small → consistent objective scaling
- Lower-bound stagnation observed with increasing coupling
- Indicates conservative relaxations, not solver failure
- Next: tighter relaxations and $\tau(T)$ reparameterization

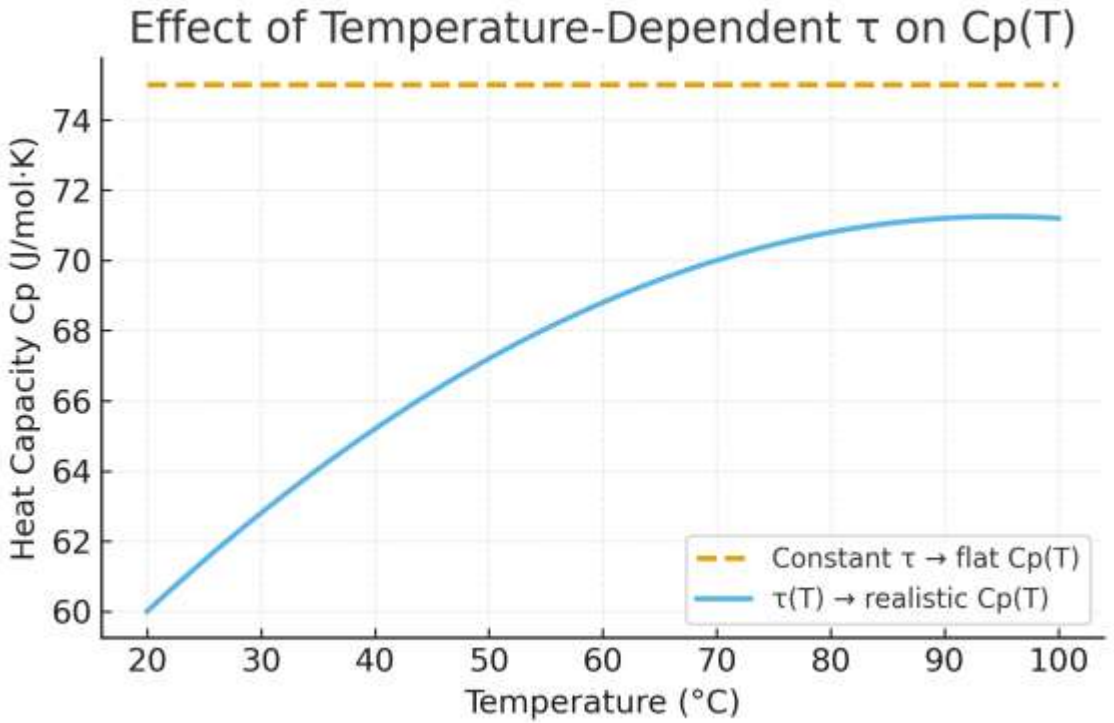
Iteration #	Nodes	Lower Bound	Upper Bound	Gap	Ratio	Timer	Time Left
100	73	0.000E+00	4.463E-02	4.463E-02	1.000E+00	119.42	180.58
200	103	0.000E+00	4.463E-02	4.463E-02	1.000E+00	197.34	102.66
300	51	0.000E+00	4.463E-02	4.463E-02	1.000E+00	247.96	52.04
376	61	0.000E+00	4.463E-02	4.463E-02	1.000E+00	300.72	-0.72

```
Time Limit Exceeded
Best Solution Found at Node 1
Lower Bound: 0.0
Upper Bound: 0.044633674354915144
Solution:
      hc_Na = 1.5978591772391082
      hc_Ca = 3.756234025771034
       hc_K = 0.25462592188560357
      hc_Li = 2.2224228870683294
      hc_Mg = 4.343045725163974
      ha_Cl = 0.6310796137595795
      ha_Br = 0.999999027967023
      τM[1] = 7.862077321458579
      τM[2] = 8.59627506967532
      τM[3] = 7.691723441677155
      τM[4] = 8.413952381334276
      τM[5] = 8.777201522423097
      τM[6] = 8.012780697836687
      τM[7] = 8.880017815959182
```

Solver reached small UB; LB stagnation due to relaxation looseness

Make Temperature-Dependent r-eNRTL Model

- **Why:**
  - Thermodynamic consistency requires τ to change with temperature.
  - Constant $\tau \Rightarrow$ zero higher-order temperature derivatives.
  - This yields unrealistic constant **Cp**, **enthalpy**.
- **How:**
  - Define $\tau(T) = f(T)$ and $h(T) = g(T)$.
  - Enables correct differentiation of Gibbs energy and accurate property prediction.



Calibrating τ and h Across Temperatures

- **Objective:**
  Estimate local τ and h at each experimental temperature T_j.

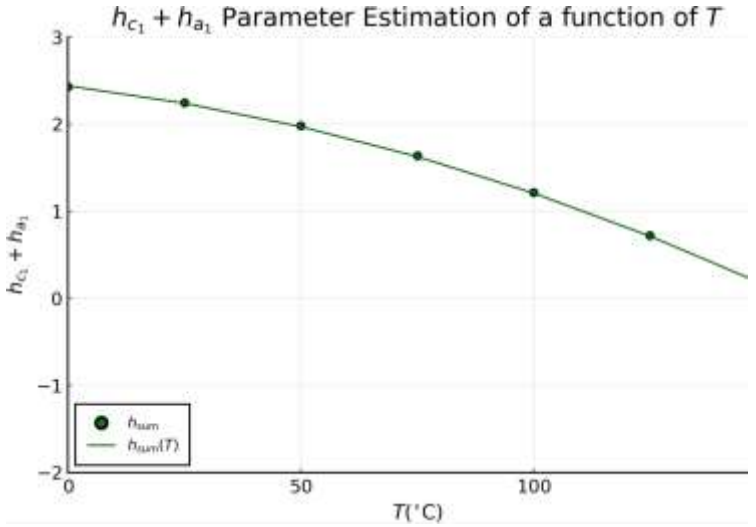
- **Method:**
  Independent parameter estimation problem solved for each temperature *(shown below)*.

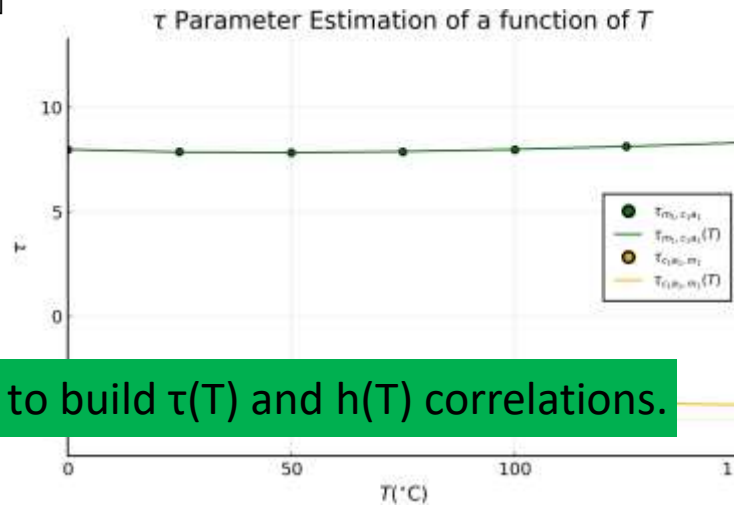
$$\tau^*, \mathbf{h}^* \in \arg\min_{\substack{\tau \in \mathcal{T} \subset \mathbb{R}^{n_\tau} \\ \mathbf{h} \in \mathcal{H} \subset \mathbb{R}^{n_h}}} f^*(\mathbf{N}_{t_j}, \mathbf{P}, \tau, \mathbf{h}, T) = \sum_{i=1}^{n} \left[\left(\frac{\gamma_i(\mathbf{N}_{t_j}, \mathbf{P}, \tau, \mathbf{h}, T)}{\gamma_i^{\exp}} - 1 \right)^2 + \left(\frac{\phi_i(\mathbf{N}_{t_j}, \mathbf{P}, \tau, \mathbf{h}, T)}{\phi_i^{\exp}} - 1 \right)^2 \right]$$

$$\text{s.t.} \quad \mathbf{h}(\tau, \mathbf{h}, \mathbf{N}_{t_j}, \mathbf{P}, T) = 0, \quad \mathbf{g}(\tau, \mathbf{h}, \mathbf{N}_{t_j}, \mathbf{P}, T) \le 0$$

$$\tau^{\mathrm{LB}} \le \tau \le \tau^{\mathrm{UB}}, \quad \mathbf{h}^{\mathrm{LB}} \le \mathbf{h} \le \mathbf{h}^{\mathrm{UB}}.$$

- **Output:**
  Temperature-specific parameter sets τ_j^*, h_j^*.





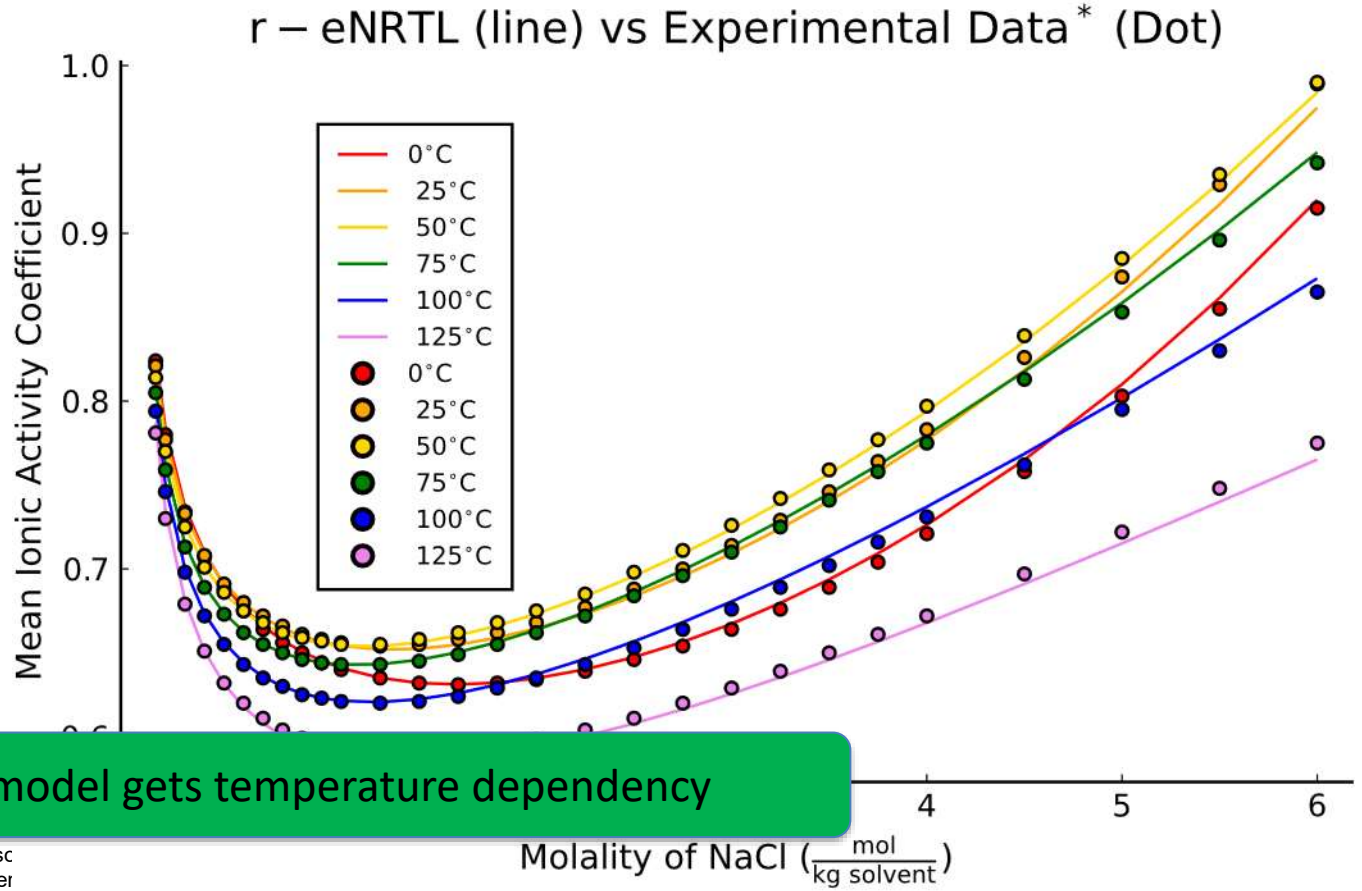
These temperature-specific parameters (dots) will be used next to build τ(T) and h(T) correlations.

Temperature-dependent r-eNRTL Reconstruction

- **Goal:**
  Show $\gamma(T)$ behavior after substituting fitted $\tau(T), h(T)$ into full r-eNRTL.
- **Implementation:**

$$h_{c_1} + h_{a_1} = B\exp(A(\frac{1}{T} - \frac{1}{T_{\text{ref}}})) + C$$

$$\tau_{ij} = C_{ij} + \frac{D_{ij}}{T} + E_{ij}\left[\frac{T_{ref} - T}{T} + ln\left(\frac{T}{T_{ref}}\right)\right]$$



The first time r-eNRTL model gets temperature dependency

[13] Lin, Y. J., Hsieh, C. J., & Chen, C. C. (2022). Association-based activity coefficient model for electrolyte sc
[14] Hossain, N., Bhattacharia, S. K., & Chen, C. C. (2016). Temperature dependence of interaction parameter
Journal, *62*(4), 1244-1253.
[15] * Silvester, L. F., & Pitzer, K. S. (1976). Thermodynamics of Geothermal Brines I. Thermodynamic Properties of Vapor-Saturated NaCl (aq) Solutions from 0-300℃.

Thermodynamic Property Expressions Generation

- Once $\tau(T)$ and $h(T)$ are defined, all thermodynamic properties can be **analytically derived** from the r-eNRTL expression.

- **Derived properties:**
  - Activity coefficient γ_{i}
  - Enthalpy H
  - Heat capacity C_{p}
  - Vapor pressure P_{v}

- **Key idea:**
  Everything comes from **differentiating** $\boldsymbol{\gamma(T)}$ — no empirical fitting needed.

[16]

$$H_{\mathrm{w}}^{0} = \Delta_{\mathrm{f}}H_{\mathrm{w}}^{\mathrm{ig}} + \int_{298.15}^{T} C_{\mathrm{p,w}}^{\mathrm{ig}}\,dT + \left[H_{\mathrm{w}}(T,P) - H_{\mathrm{w}}^{\mathrm{ig}}(T,P)\right] \quad (1)$$

$$H_{k}^{\infty,\mathrm{aq}} = \Delta_{\mathrm{f}}H_{k}^{\infty,aq} + \int_{298.15}^{T} C_{\mathrm{p,k}}^{\infty,\mathrm{aq}}\,dT \quad (2)$$

$$H^{*\mathrm{ex}} = -RT^{2}\sum_{i} x_{i}\frac{\partial \log \gamma_{i}}{\partial T} \quad (3)$$

$$H^{\mathrm{l}} = x_{\mathrm{w}}H_{\mathrm{w}}^{0} + \sum_{k} x_{k}H_{k}^{\infty,\mathrm{aq}} + H^{*\mathrm{ex}} \quad (4)$$

$$C_{\mathrm{p}} = \left(\frac{\partial H^{\mathrm{l}}}{\partial T}\right)_{P} \quad (5)$$

[16] Yan, Y., & Chen, C. C. (2011). Thermodynamic representation of the NaCl+ Na2SO4+ H2O system with electrolyte NRTL model. *Fluid phase equilibria*, *306*(2), 149-161.

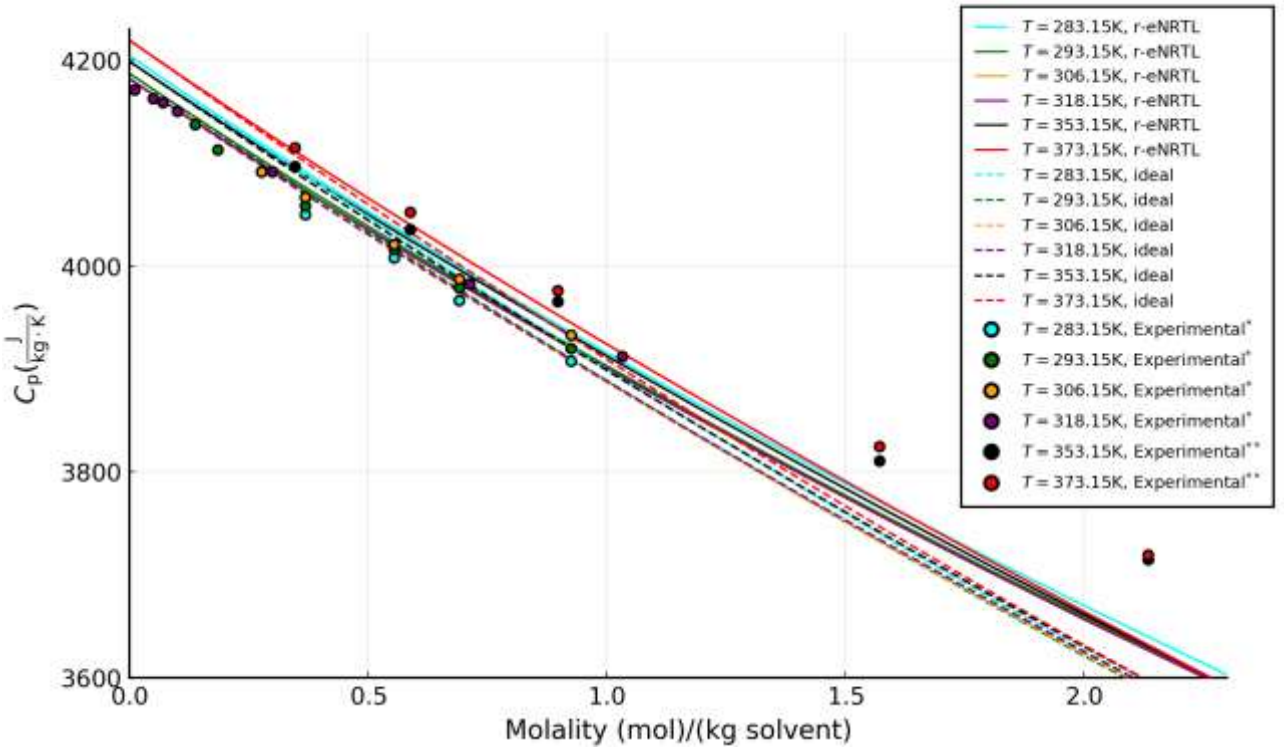
Heat Capacity Validation

- **Analytical result:**
  $C_{\mathrm{p}}(T)$ derived directly from the temperature-dependent r-eNRTL model.

- **Validation:**
  Confirms the correct $\gamma \rightarrow G^{\mathrm{n}} \rightarrow C_{\mathrm{p}}$ differentiation path.

- **Trend:**
  Reproduces the expected C_{p}–molality behavior,
  showing thermodynamic consistency.

- **Next:**
  Quantitative refinement is ongoing.



[17] (*,**) Likke, S., & Bromley, L. A. (1973). Heat capacities of aqueous sodium chloride, potassium chloride, magnesium chloride, magnesium sulfate, and sodium sulfate solutions between 80. deg. and 200. deg. *Journal of Chemical and Engineering Data*, *18*(2), 189-195.
[18] (***) Archer, D. G., & Carter, R. W. (2000). Thermodynamic properties of the NaCl+ H2O system. 4. Heat capacities of H2O and NaCl (aq) in cold-stable and supercooled states. *The Journal of Physical Chemistry B*, *104*(35), 8563-8584.

Conclusions

- **Conclusions**
  - Developed an **automated symbolic framework** for r-eNRTL.
  - **Simplified** complex symbolic expressions and **accelerated** optimization.
  - **Demonstrated global parameter estimation** across multiple electrolytes.
  - The temperature-dependent formulation, $\tau(T)$, **ensures consistent thermodynamic properties.**
  - **Validated a complete workflow** from symbolic modeling to optimization and property prediction.

Acknowledgements

Members of the Process Systems and Operations Research Laboratory at the University of Connecticut (https://www.psor.uconn.edu)

Funding:

Department of Energy, Award No.: DE-EE0010991

Pratt & Whitney Endowed Professorship

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 Department of Energy or the United States Government.