UCONN
UNIVERSITY OF CONNECTICUT

Automatic Differentiation and Source Code Generation for Dynamic Modeling and Simulation of Brine Systems

Pengfei Xu (UConn)

Robert X. Gottlieb (UConn), E. Soraya Rawlings (SNL)

PI : Prof. Matthew Stuber (UConn)

Session : Applied Math for Energy and Environmental Applications

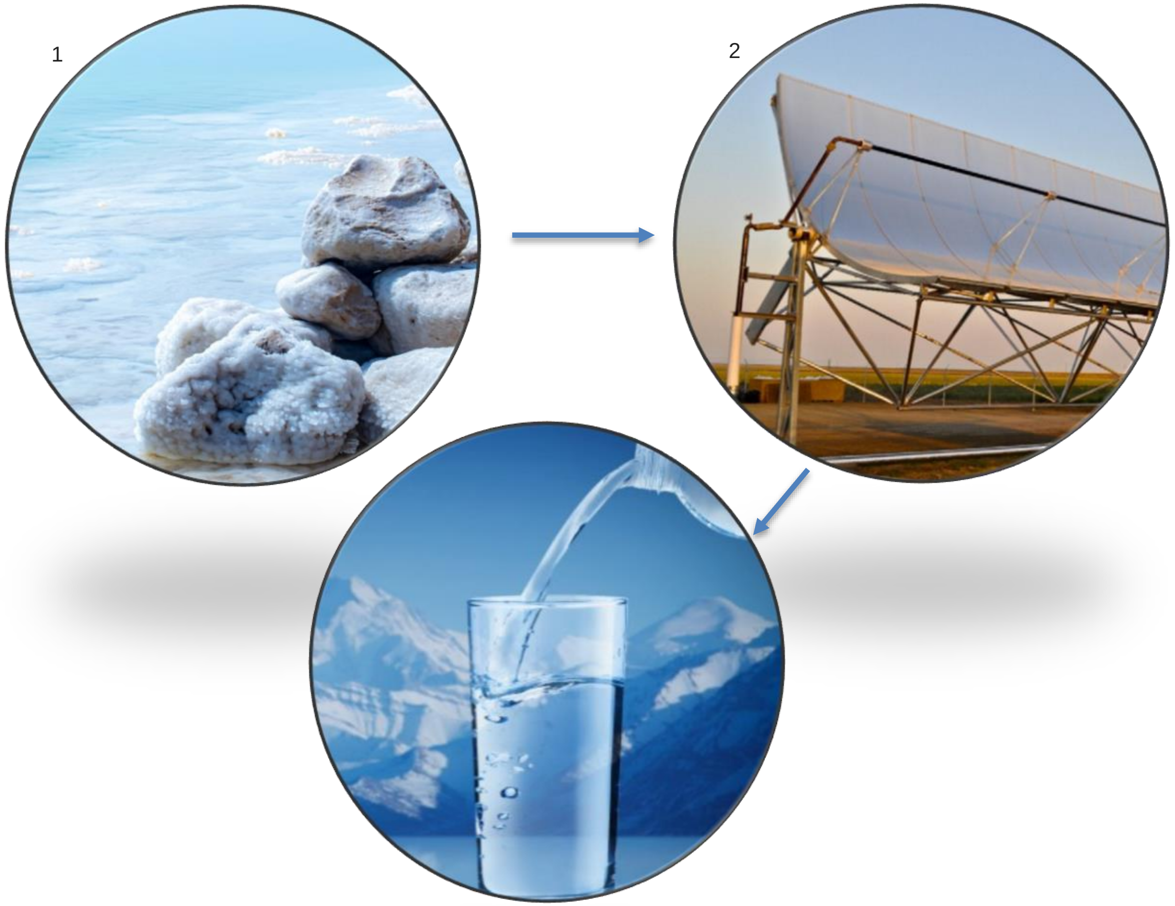
Nov 9th, 2023

2023 AIChE ANNUAL MEETING

Process Systems and Operations Research Laboratory

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

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)

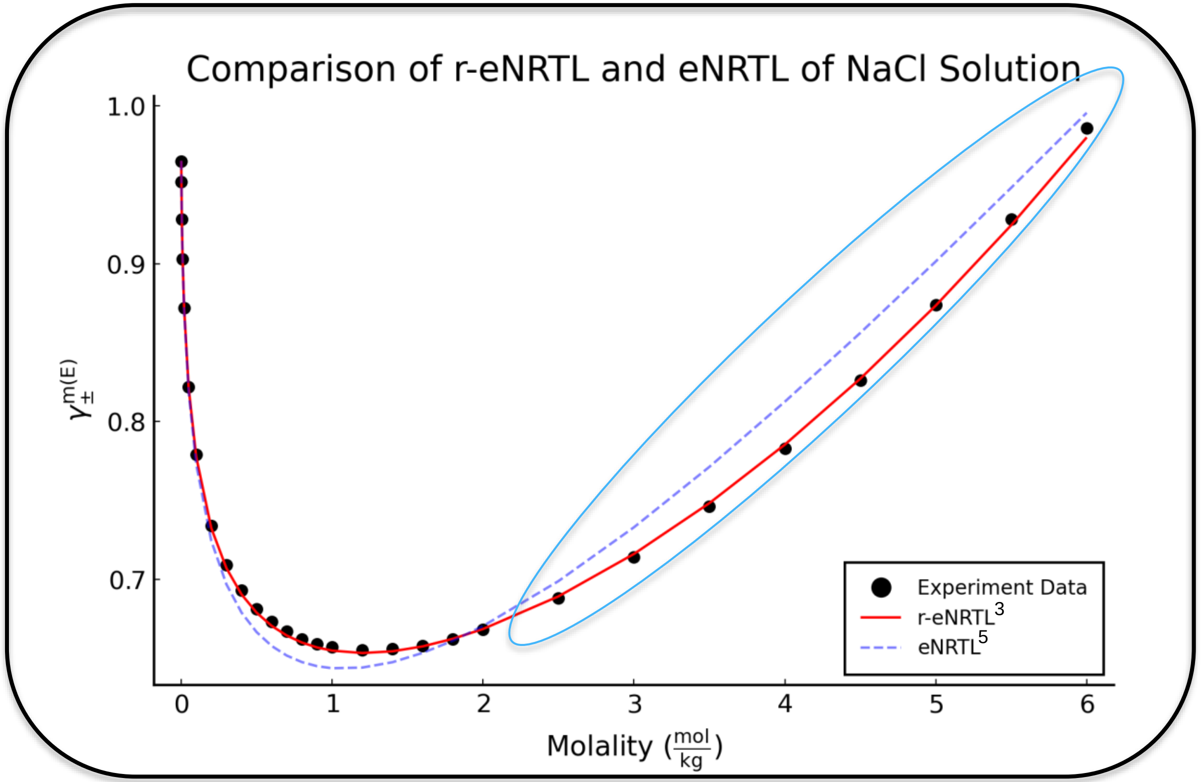
- Increase sustainability
- Reduce costs
- Improve system robustness

We need accurate mechanistic models that can enable

- Dynamic simulation
- Optimization-based design of water treatment technologies.

[1] Molinari, Raffaele., et al. **"Can brine from seawater desalination plants Be a source of critical metals?."** *CHEMVIEWS* (2022).
[2] Stuber, Matthew D., et al. **"Pilot demonstration of concentrated solar-powered desalination of subsurface agricultural drainage water and other brackish groundwater sources."** *Desalination* 355 (2015): 186-196.

Motivation



Modeling accuracy in high concentration regime

- Refined e-NRTL[3]

Modeling system dynamics

- ModelingToolkit[4]

[3] Bollas, G.M., et al. **Refined electrolyte-NRTL model: Activity coefficient expressions for application to multi-electrolyte systems**. *AIChE Journal* 54(6): 1608-1624 (2008).
[4] Ma, Yingbo, et al. **Modelingtoolkit: A composable graph transformation system for equation-based modeling**. *arXiv preprint arXiv:*2103.05244 (2021).
[5] Song, Yuhua, et al. **"Symmetric electrolyte nonrandom two-liquid activity coefficient model."** *Industrial & Engineering Chemistry Research* 48, no. 16 (2009): 7788-7797.

Refined eNRTL

Short Range Interaction

Long Range Interaction

Born term, for aqueous system = 0

$$G^{*,ex} = G^{*,SR} + G^{*,LR} + \Delta G^{*,Born}$$

$$= G^{*,SR} + (A^{*,LR} + PV) + 0$$

- Increase accuracy of activity coefficient
- Improve accuracy of vapor-liquid equilibrium in high concentration regime for simulating evaporator performance.

Challenges to Implement r-eNRTL

Published paper notation versus computer implementation

- Not in a tensor form
- Hard to get the physical meaning which is important to understanding

Complexity of the model and its use cases

- More complex than original eNRTL[6] which is already very complex

[6] Chen, Chau-Chyun, et al. **"A local composition model for the excess Gibbs energy of aqueous electrolyte systems."** *AIChE Journal* 32, no. 3 (1986): 444-454.

New Tensor Notation

Classical Notation[6]

$$\sum_a X_a \sum_c \left(\frac{X_c}{\sum_{c'} X_{c'}}\right)\left(\frac{\sum_j X_j G_{ja.ca}\tau_{ja.ca}}{\sum_j X_j G_{ja.ca}}\right)$$

Tensor Notation

$$\sum_{j=1}^{n_c} X_{c_j} \sum_{k=1}^{n_a}\left(\frac{X_{a_k}}{\sum_{k'=1}^{n_a} X_{a_{k'}}}\right)\left(\frac{\sum_{s\in\{m,a\}}\sum_{l=1}^{n_s} X_{s_l} F_{c_j,s_l,a_k}\tau_{c_j,s_l,a_k}}{\sum_{s\in\{m,a\}}\sum_{l=1}^{n_s} X_{s_l} F_{c_j,s_l,a_k}}\right)$$

Complexity of Refined eNRTL

[7]

$$\frac{G^{\mathrm{SR}}}{RT} = \sum_{j=1}^{n_m} X_{m_j} \left(\frac{\sum_{s\in\{m,a,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{m_j,s_l,m_j} \tau_{m_j,s_l,m_j}}{\sum_{s\in\{m,a,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{m_j,s_l,m_j}} \right)$$

$$+\sum_{j=1}^{n_a} X_{a_j} \sum_{k=1}^{n_c} \left(\frac{X_{c_k}}{\sum_{k'=1}^{n_c} X_{c_{k'}}} \right) \left(\frac{\sum_{s\in\{m,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{a_j,s_l,c_k} \tau_{a_j,s_l,c_k}}{\sum_{s\in\{m,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{a_j,s_l,c_k}} \right)$$

$$+\sum_{j=1}^{n_c} X_{c_j} \sum_{k=1}^{n_a} \left(\frac{X_{a_k}}{\sum_{k'=1}^{n_a} X_{a_{k'}}} \right) \left(\frac{\sum_{s\in\{m,a\}} \sum_{l=1}^{n_s} X_{s_l} F_{c_j,s_l,a_k} \tau_{c_j,s_l,a_k}}{\sum_{s\in\{m,a\}} \sum_{l=1}^{n_s} X_{s_l} F_{c_j,s_l,a_k}} \right)$$

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

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

Complexity of Refined eNRTL

$$\frac{G^{\mathrm{SR}}}{RT} = \sum_{j=1}^{n_m} X_{m_j} \left(\frac{\sum_{s\in\{m,a,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{m_j,s_l,m_j} \tau_{m_j,s_l,m_j}}{\sum_{s\in\{m,a,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{m_j,s_l,m_j}} \right)$$

$$+ \sum_{j=1}^{n_a} X_{a_j} \sum_{k=1}^{n_c} \left(\frac{X_{c_k}}{\sum_{k'=1}^{n_c} X_{c_{k'}}} \right) \left(\frac{\sum_{s\in\{m,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{a_j,s_l,c_k} \tau_{a_j,s_l,c_k}}{\sum_{s\in\{m,c\}} \sum_{l=1}^{n_s} X_{s_l} F_{a_j,s_l,c_k}} \right)$$

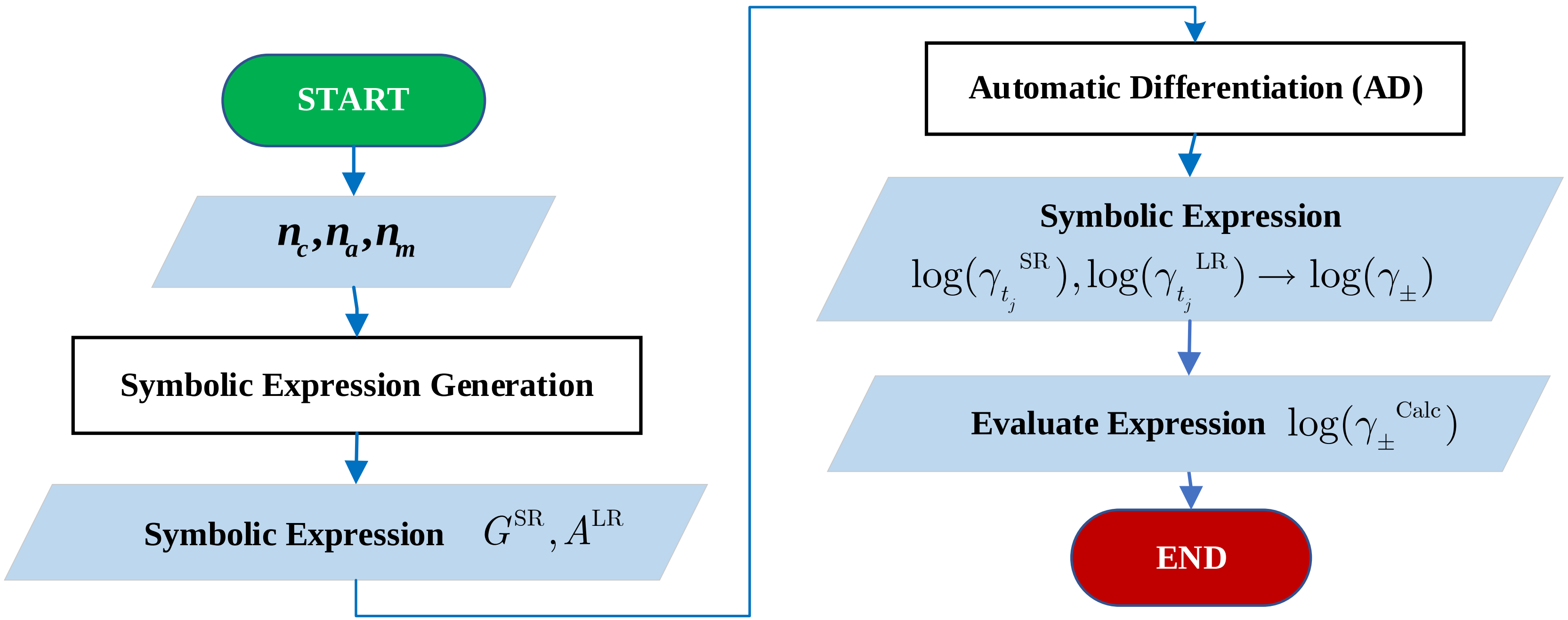
$$+ \sum_{j=1}^{n_c} X_{c_j} \sum_{k=1}^{n_a} \left(\frac{X_{a_k}}{\sum_{k'=1}^{n_a} X_{a_{k'}}} \right) \left(\frac{\sum_{s\in\{m,a\}} \sum_{l=1}^{n_s} X_{s_l} F_{c_j,s_l,a_k} \tau_{c_j,s_l,a_k}}{\sum_{s\in\{m,a\}} \sum_{l=1}^{n_s} X_{s_l} F_{c_j,s_l,a_k}} \right)$$

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

AD

Complicated Expression

Automatic Differentiation Workflow



n_c, n_a, n_m: Number of species in aqueous phase.

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

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

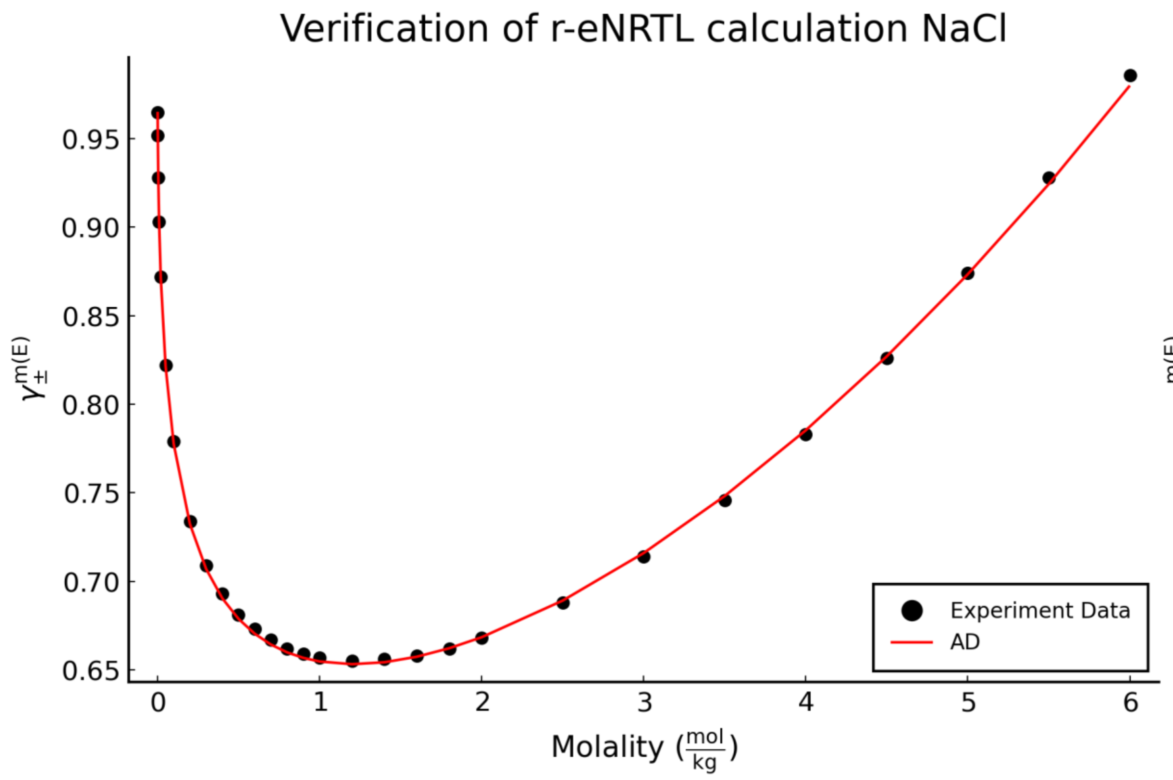
$\gamma_{t_j}^{SR}$: Short Range activity coefficient of species t_j.

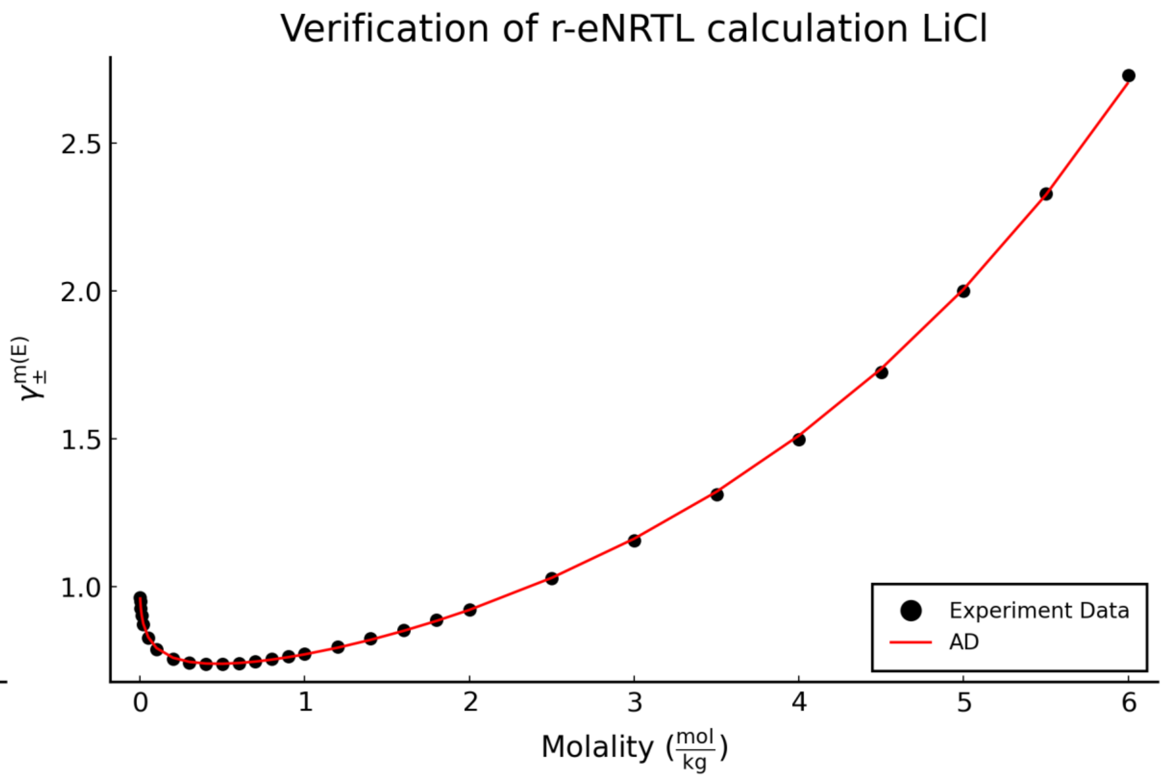
$\gamma_{t_j}^{LR}$: Long Range activity coefficient of species t_j.

t_j: Species with type t, $t \in \{a, c, m\}$ and index j.

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

Simulation Result with AD





Two Systems

ModelingToolkit[4] (Julia)

Dynamic system modeling using ModelingToolkit.jl library

SciML

Open Source Software for Scientific Machine Learning

https://sciml.ai

IDAES (Pyomo)

Steady-state IDAES model under nonideal equilibrium.

IDAES®

Institute for the Design of Advanced Energy Systems

[9] Stuber, M.D. (2023). NAWIConcentratedElectrolytes. GitHub. https://github.com/PSORLab/NAWIConcentratedElectrolytes/tree/main/flowsheets/old_Modelica

Two Systems

ModelingToolkit[4] (Julia)

Dynamic system modeling using ModelingToolkit.jl library

SciML

Open Source Software for Scientific Machine Learning

R-eNRTL

γ_{H_2O}

IDAES (Pyomo)

Steady-state IDAES model under nonideal equilibrium.

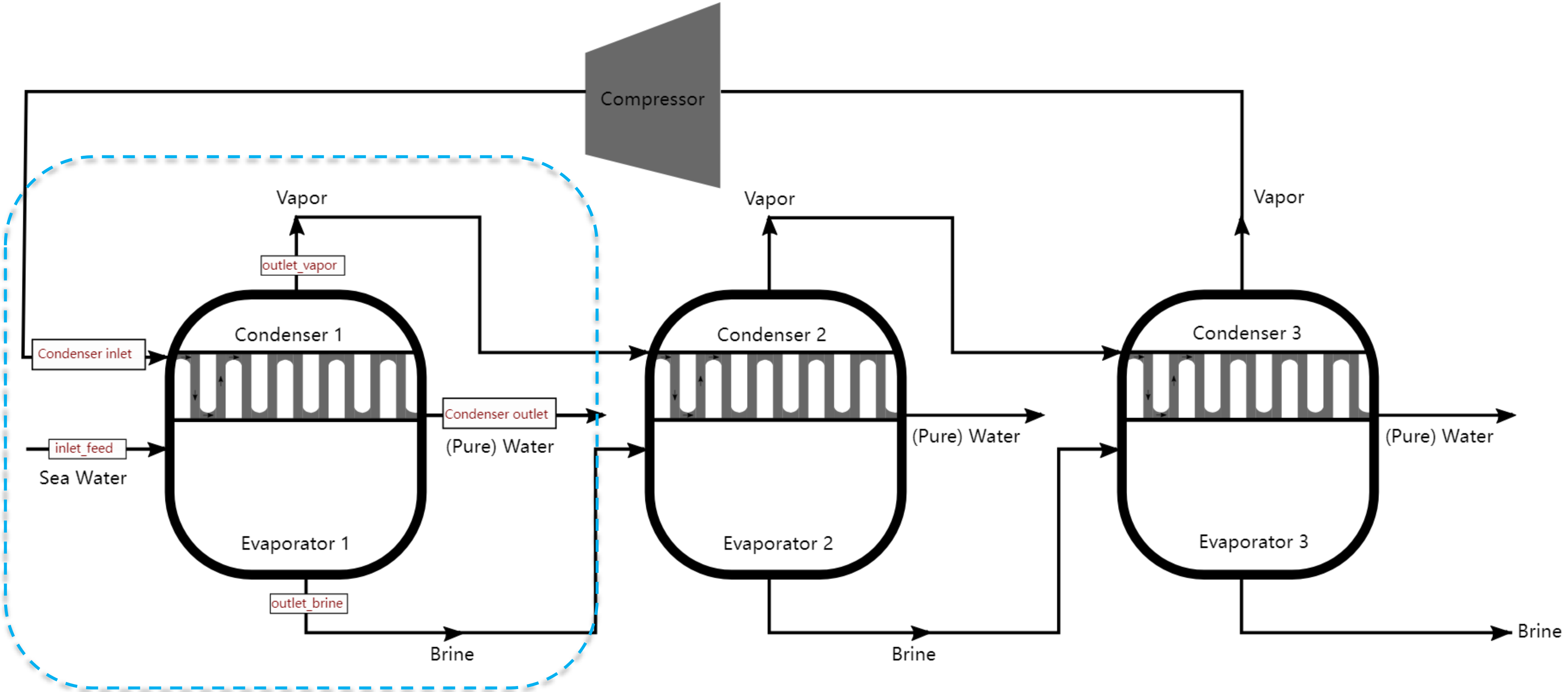
IDAES

Institute for the Design of Advanced Energy Systems

VLE Evaporator

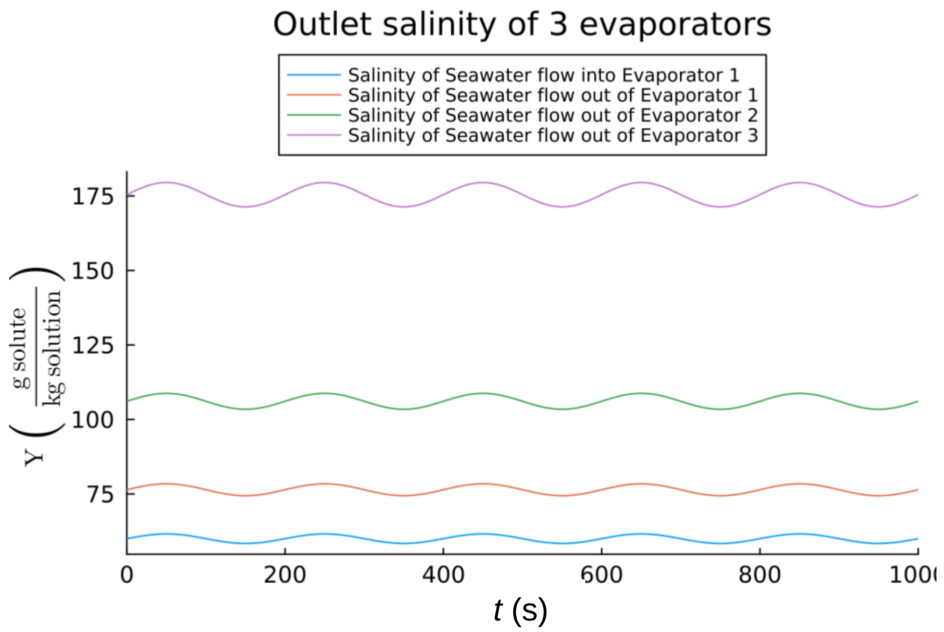
$$x_{H_2O}\gamma_{H_2O}P^{sat}_{H_2O} = y_{H_2O}P_{H_2O}$$

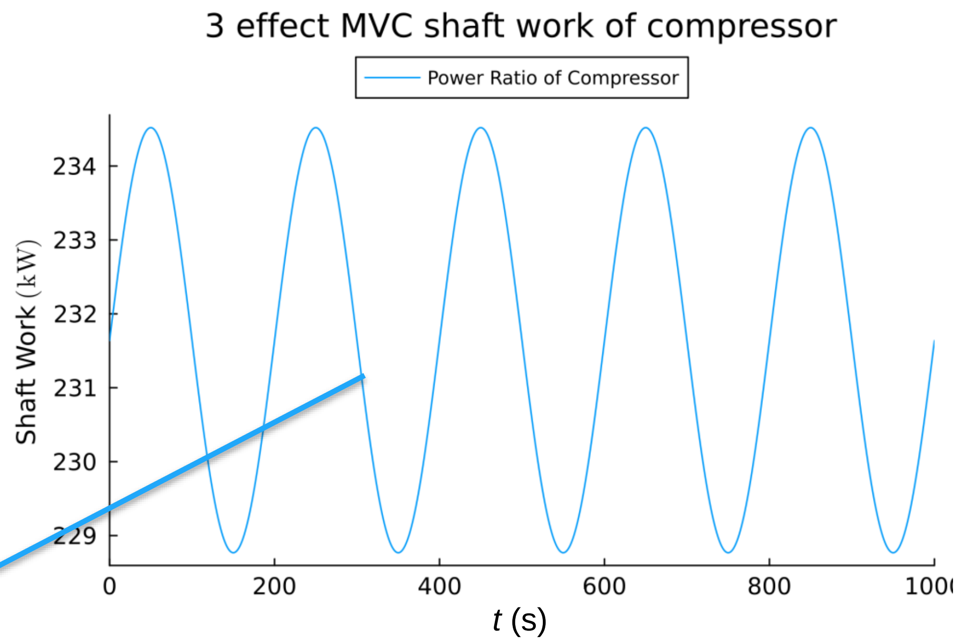
Julia Example: 3-Effect MVC-MED

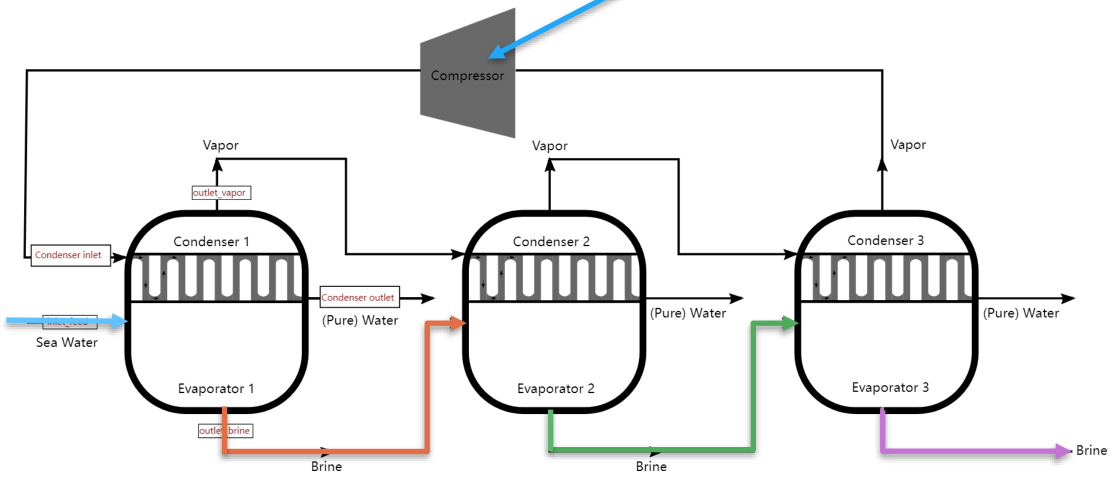


[10] Edna-Soraya, R. (2022). NAWIConcentratedElectrolytes. GitHub. https://github.com/PSORLab/NAWIConcentratedElectrolytes/tree/main/flowsheets/benchmark_system/mvc

Julia Example: Dynamic Simulation Results







IDAES

IDAES (Pyomo)

Steady-state IDAES model under nonideal equilibrium.

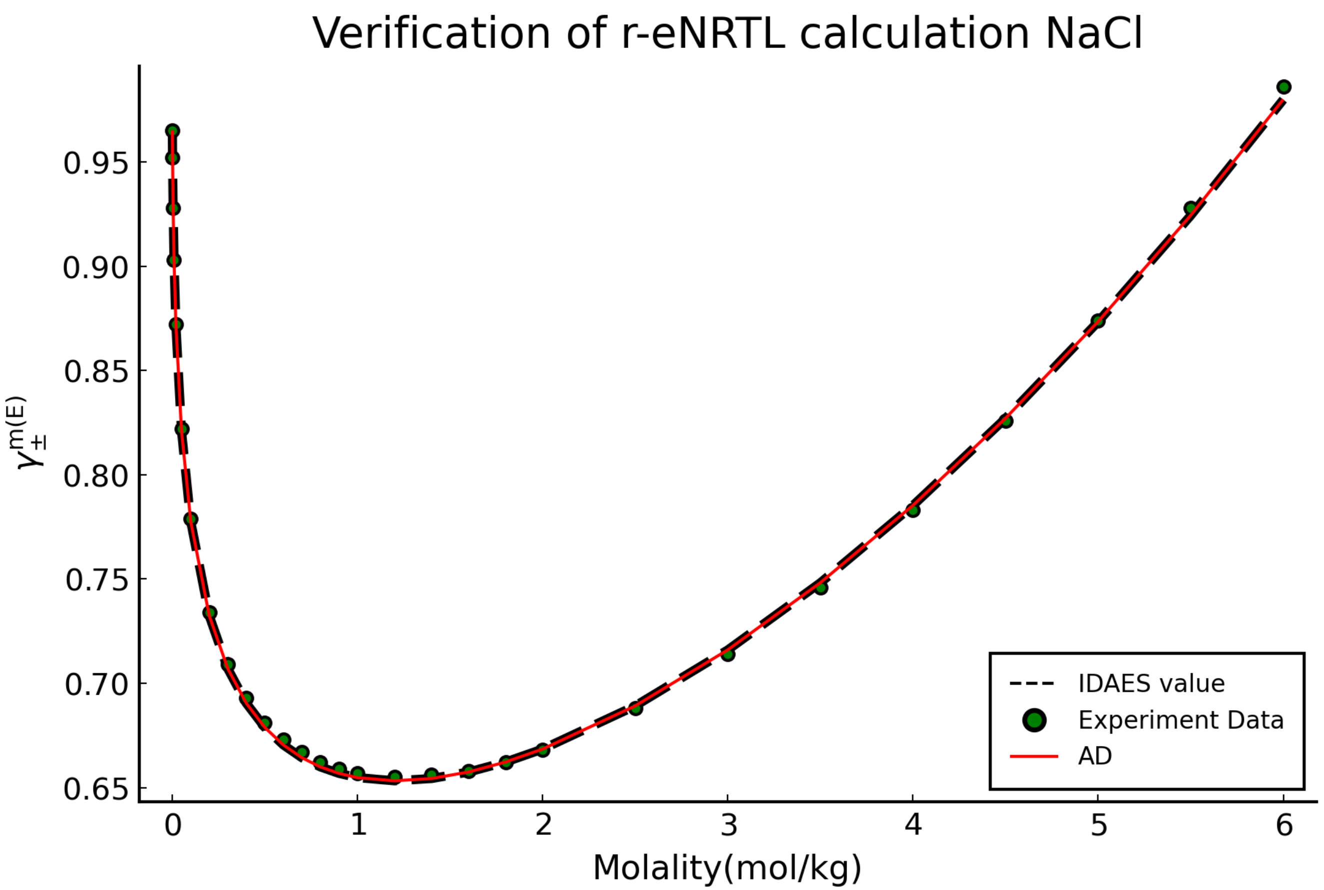
IDAES Institute for the Design of Advanced Energy Systems

Julia Translated r-eNRTL Model

IDAES 3-effect MVC model

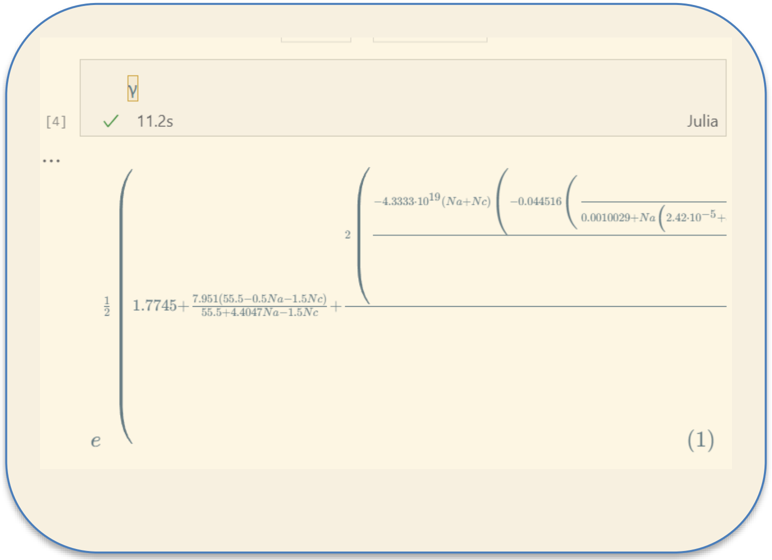
IDAES r-eNRTL model

IDAES vs. Julia r-eNRTL



Transpiler of Symbolic Expression from Julia to Pyomo

Julia



Julia generated the expression and save it into a txt file

Translator

```python
    line = line.replace('9'+"(","9*(")
    expression.append(line)

line_1 = "from pyomo.environ import exp,log"
line_2 = "def reNRTL_gamma_H2O(Na,Nc):"
line_3 = "    expr = "+expression[0]
line_4 = "    return expr"

more_lines = [line_1,line_2,line_3,line_4]
with open('sympy_script.py', 'w') as f:
    f.write('\n'.join(more_lines))

from sympy_script import *
```

Translator translate Julia expression into the python function syntax and then import it.

IDAES

```python
@m.fs.Constraint(m.fs.set_evaporators, doc="e
def eq_nonideal_activity_coefficient(b, e):
    x = b.evaporator[e].properties_brine[0].m
    x_m = x*58.443/(x*58.443+(1-x)*18.01528)
    return (
        b.act_coeff[e]
        == reNRTL_gamma_H2O(x_m,x_m)
    )

# Deactivate equilibrium equation from evapor
# new equilibrium equation as a Constraint th
# activity coefficient. Note that since water
# participating in the vapor-liquid equilibri
# coefficient and vapor pressure are of water
# solvent.
```

Call the function inside equality constraint where the activity coefficient are defined

Generated Function in IDAES

```python
# Save the calculated activity coefficient in the global
# activity coefficient variable.
@m.fs.Constraint(m.fs.set_evaporators, doc="eNRTL activity coefficient for water")
def eq_nonideal_activity_coefficient(b, e):
    x = b.evaporator[e].properties_brine[0].mole_frac_phase_comp["Liq", "TDS"]
    x_m = x*58.443/(x*58.443+(1-x)*18.01528)
    return (
        b.act_coeff[e]
        == reNRTL_gamma_H2O(x_m,x_m)
    )

# Deactivate equilibrium equation from evaporator and include a
# new equilibrium equation as a Constraint that includes the
# activity coefficient. Note that since water is the only solvent
# participating in the vapor-liquid equilibrium, the activity
# coefficient and vapor pressure are of water as the
# solvent.
```

In the MVC model, we integrate our generated function into the equality constraint, replacing the embedded refined eNRTL model in IDAES.

Comparison between Embedded Function and Translated Function

Julia Translated Function

```
--------------------------------------------------
Other variables:
--------------------------------------------------

 Energy input (kW): 304.4402
 Total water produced (gal/min): 107.0689
 Specific energy consumption (SC, kWh/m3): 12.5191
 Water recovery (%): 70.0000
 Molal conc solute evap 1 (mol/kg): 1.4557
 Molal conc solute evap 2 (mol/kg): 1.9279
 Molal conc solute evap 3 (mol/kg): 3.8498
```

IDAES Embedded Function

```
--------------------------------------------------
Other variables:
--------------------------------------------------

 Energy input (kW): 304.5390
 Total water produced (gal/min): 107.0689
 Specific energy consumption (SC, kWh/m3): 12.5232
 Water recovery (%): 70.0000
 Molal conc solute evap 1 (mol/kg): 1.4555
 Molal conc solute evap 2 (mol/kg): 1.9275
 Molal conc solute evap 3 (mol/kg): 3.8498
```

* Results may be slightly different due to numerical calculations

[11] Edna-Soraya, R. (2022). **watertap-renrtl**. GitHub. https://github.com/watertap-org/watertap-renrtl

Conclusion

- Implemented refined eNRTL model for single electrolyte with AD.
  - Developed a new notation for r-eNRTL
  - Generate activity coefficients as symbolic expressions
- Built a library to model dynamic evaporator systems that uses r-eNRTL in VLE expressions in Julia.
- Created a toolchain that generates callback functions for IDAES and Pyomo to use the r-eNRTL activity coefficients (and the same procedure can be extended to other properties)
- Currently implementing the multi-electrolyte form and expanding the AD work for all other thermodynamic properties

Thanks!

2023 AIChE ANNUAL MEETING

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

NAWI National Alliance for Water Innovation

Process Systems and Operations Research Laboratory

Funding:

National Alliance for Water Innovation (NAWI), DOE-EERE, Task 6.7

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 Alliance for Water Innovation or the U.S. government.

Questions?

Process Systems and
Operations Research
Laboratory

https://www.psor.uconn.edu