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
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.

Motivation

Modeling accuracy in high concentration regime

➢ Refined e-NRTL\(^3\)

Modeling system dynamics

➢ ModelingToolkit\(^4\)

---


Refined eNRTL

\[ G^{*,\text{ex}} = G^{*,\text{SR}} + G^{*,\text{LR}} + \Delta G^{*,\text{Born}} \]

- 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 which is already very complex

---

New Tensor Notation

Classical Notation\textsuperscript{6}

\[ \sum_{a} X_{a} \sum_{c} \left( \frac{X_{c}}{\sum_{c'} X_{c'}} \right) \left( \sum_{j} \frac{X_{j} G_{ja,ca}^{\tau}}{\sum_{j} X_{j} G_{ja,ca}} \right) \]

Tensor Notation

\[ \sum_{j=1}^{n_{c}} X_{c_{j}} \sum_{k=1}^{n_{a}} \left( \frac{X_{a_{k}}}{\sum_{k'=1}^{n_{a}} X_{a_{k'}}} \right) \left( \sum_{s \in \{m,a\}} \sum_{l=1}^{n_{s}} \sum_{c_{j},s_{l},a_{k}} X_{s_{l}}^{F} F_{c_{j},s_{l},a_{k}}^{T} \right) \]
Complexity of Refined eNRTL

\[
\frac{G_{SR}}{RT} = \sum_{j=1}^{n_m} X_{m_j} \left( \sum_{s \in \{m, a, c\}} \sum_{l=1}^{n_s} X_{s_l} F_{m_j, s_l, m_j} \right) \\
+ \sum_{j=1}^{n_a} \sum_{k=1}^{n_a} \left( \sum_{s \in \{m, c\}} \sum_{l=1}^{n_s} X_{s_l} F_{a_j, s_l, c_k} \right) \\
+ \sum_{j=1}^{n_c} \sum_{k=1}^{n_a} \left( \sum_{s \in \{m, a\}} \sum_{l=1}^{n_s} X_{s_l} F_{c_j, s_l, a_k} \right)
\]

\log \gamma_{i_j}^{SR} = \frac{\partial}{\partial N_{i_j}} \left( \sum_{i \in \{m, a, c\}} \sum_{j=1}^{n_i} N_{i_j} \frac{G_{SR}}{RT} \right)

Complicated Expression of Refined eNRTL

$$G_{SR}^{\text{SR}} = \sum_{j=1}^{m} X_{mj} \left( \sum_{s \in \{m,a,c\}} \sum_{l=1}^{n} X_{s_{l}} F_{m_{j},s_{l},m_{j}} \right)$$

$$+ \sum_{j=1}^{n_{c}} X_{a_{k}} \sum_{k=1}^{n_{c}} X_{c_{k}} \sum_{l=1}^{n} X_{s_{l}} F_{m_{j},s_{l},m_{j}}$$

$$\log \gamma_{t_{j}}^{\text{SR}} = \frac{\partial}{\partial N_{t_{j}}} \left( \sum_{i \in \{m,a,c\}} \sum_{j=1}^{n_{i}} N_{i_{j}} \frac{G_{SR}^{\text{SR}}}{RT} \right)$$

Complicated Expression
Automatic Differentiation Workflow

\[ \text{Symbolic Expression Generation} \]

- \( n_c, n_a, n_m \): Number of species in aqueous phase.
- \( G_{SR} \): Short range excess Gibbs free energy.
- \( A_{LR} \): Long range excess Helmholtz free energy.

\[ \text{Automatic Differentiation (AD)} \]

- \( \gamma_{SR}^{\text{Calc}}, \gamma_{LR}^{\text{Calc}} \rightarrow \log(\gamma_{\pm}^{\text{Calc}}) \)

Simulation Result with AD

Verification of r-eNRTL calculation NaCl

Verification of r-eNRTL calculation LiCl
Two Systems

ModelingToolkit⁴ (Julia)
Dynamic system modeling using ModelingToolkit.jl library

IDAES (Pyomo)
Steady-state IDAES model under nonideal equilibrium.

Two Systems

ModelingToolkit\textsuperscript{4} (Julia)

Dynamic system modeling using ModelingToolkit.jl library

IDAES (Pyomo)

Steady-state IDAES model under nonideal equilibrium.

VLE Evaporator

\[ x_{\text{H}_2\text{O}} \gamma_{\text{H}_2\text{O}} P_{\text{H}_2\text{O}}^{\text{sat}} = y_{\text{H}_2\text{O}} P_{\text{H}_2\text{O}} \]
Julia Example: 3-Effect MVC-MED

Julia Example: Dynamic Simulation Results

Outlet salinity of 3 evaporators

3 effect MVC shaft work of compressor
IDAES (Pyomo)

Steady-state IDAES model under nonideal equilibrium.

Julia Translated r-eNRTL Model

IDAES 3-effect MVC model

IDAES r-eNRTL model
Verification of r-eNRTL calculation NaCl

\[ Y^{\omicron(E)} \]

- IDAES value
- Experiment Data
- AD
Transpiler of Symbolic Expression from Julia to Pyomo

Julia generated the expression and save it into a txt file.

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

Call the function inside equality constraint where the activity coefficient are defined.
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): 1.4557
Molal conc solute evap 2 (mol/kg): 1.9279
Molal conc solute evap 3 (mol/kg): 3.8498

* Results may be slightly different due to numerical calculations

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!

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

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