Ion Exchange for the Separation of Rare Earth Elements

Edna Soraya Rawlings, Bethany Nicholson, John Siirola
Sandia National Laboratories

PROMMIS
Process Optimization and Modeling for Minerals Sustainability

Motivation

- Ion exchange (IX) separates ions from a liquid solution using targeted ion-exchange resins and has many benefits for REE separation:
  - High selectivity
  - Environmentally benign
  - Reversible and reusable process
  - Effective for low concentration REEs

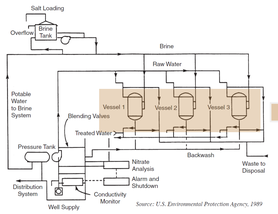


Fig. 1. IX treatment system[1].

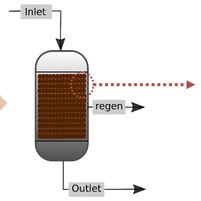


Fig. 2. Scheme of an IX.

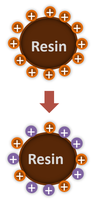


Fig. 3. Scheme describing ion exchange in resins.

Guiding Questions

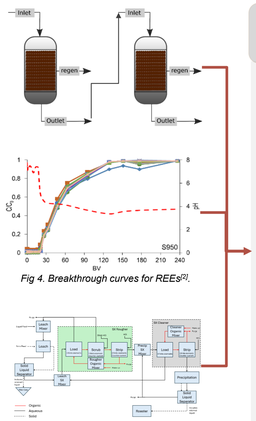


Fig 4. Breakthrough curves for REEs[2].

Fig. 5. University of Kentucky flowsheet[3].

How to **extract REEs** from liquid streams using IX?

- What can be reused from the **WaterTAP** IX model and what needs to be modified?
- What **data** is needed and is it available?
- What are the equilibrium assumptions in the **absorption processes**?
- Where to **integrate IX** in a large process?
- Can IX be combined with other separation techniques to **reduce costs**?
- How does separation selectivity impact overall **process flexibility**?

Methodology

REE removal using IX

IX model development

Equilibrium parameter estimation

Integration with process

Model Validation

IX model based on unit model from WaterTAP
- Include resin-specific equations and data
- Add support for single and multiple ion extraction

Experimental equilibrium data analysis
- Langmuir and Freundlich isotherms
- Parameter estimation using data from breakthrough curves

Integration with REE separation process
- Add IX to University of Kentucky (Uky) flowsheet
- Optimize process capital costs and IX design

IX Unit Model Development

- Reworked the IX unit model from WaterTAP to facilitate switching resins
- IX model supports single and multiple ion extraction leveraging a multi-component aqueous solution property package[6]

```python
def add_properties_resin(self):
    """
    Method to add resin-specific property calculations
    """
    # Pressure drop (psi/m of resin bed depth) is a function of
    # loading rate (vel_bed) in m/hr
    @self.Expression(doc="Pressure drop")
    def pressure_drop(b):
        vel_bed = pyunits.convert(b.vel_bed,
                                  to_units=pyunits.m / pyunits.hr)
        return (
            b.pressure_drop_param_A
            + b.pressure_drop_param_B * vel_bed
            + b.pressure_drop_param_C * vel_bed**2
        ) * b.bed_depth

    # Bed expansion is calculated as a fraction of the bed_depth
    # The bed_expansion_frac is calculated as a function of
    # backwash rate (bw_rate, m/hr)
    @self.Expression(doc="Bed expansion fraction from backwashing")
    def bed_expansion_frac(b):
        return (
            b.bed_expansion_frac_param_A
            + b.bed_expansion_frac_param_B * b.bw_rate
            + b.bed_expansion_frac_param_C * b.bw_rate**2
        )
```

```python
def _add_parameters_resin(self, resin=None):
    """
    Method to add resin-specific parameters
    """
    # Read resin parameters from json file
    with open("resin_data.json") as data_file:
        resin_params = json.load(data_file)

    # Extract parameters needed for property calculations
    self.pressure_drop_param_A = pyo.Param(
        initialize= resin_params[resin]["bed_expansion"]["param_a"],
        mutable=True,
        units=pyunits.psi / pyunits.m,
        doc="Parameter a for pressure drop calculation")
    self.pressure_drop_param_B = pyo.Param(
        initialize= resin_params[resin]["bed_expansion"]["param_b"],
        mutable=True,
        units=(pyunits.psi * pyunits.hr) / pyunits.m**2,
        doc="Parameter b for pressure drop calculation")
    self.pressure_drop_param_C = pyo.Param(
        initialize= resin_params[resin]["bed_expansion"]["param_c"],
        mutable=True,
        units=(pyunits.psi * pyunits.hr**2) / pyunits.m**3,
        doc="Parameter b for pressure drop calculation")
    ...
```

(same for bed expansion parameters)

IX model can be easily adapted to new REE species and resins

```
{"XAD7HP": {"type": "Dupont Amberlite XAD(TM)7HP Polymeric Adsorbent",
            {"A850": { "type": "Strong-base Type I Acrylic Anion Exchange",
                       {"S950": {"type": "Chelating Polystyrene-Divinylbenzene",
                                 "bed_expansion": {"units": "fraction",
                                                   "temperature_ref": 20,
                                                   "temperature_units": "C",
                                                   "param_a": 0.0004,
                                                   "param_b": 0.0587,
                                                   "param_c": 0},
                                 "pressure_drop": {"units": "psi/m",
                                                   "temperature_ref": 25,
                                                   "temperature_units": "C",
                                                   "param_a": 0.2332,
                                                   "param_b": 0.1714,
                                                   "param_c": 0},
                                 "diameter": {"units": "m",
                                              "diam": 750e-6},
                                 …
                                 "reference": [" M. Hermassi et al., Journal of
                                                Cleaner Production 379…"]
                                 }
                       }
            }
}
```

Equilibrium Data Analysis

- Parameter estimation to calculate Freundlich data using Parmest and WaterTAP[4]

$$c/c_0 = \frac{1}{\left[1 + (2^{n-1} - 1)\, exp\left(\frac{k_T x (n-1)}{BV_{50} v}(BV_{50} - BV)\right)\right]^{\frac{1}{n-1}}}$$

Fig 6. Clark model equation[4].

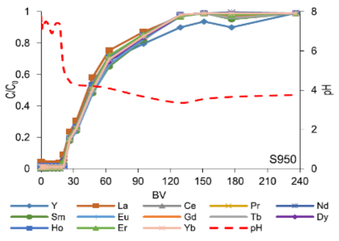


Fig 7. Breakthrough curves for the recovery of REEs[2].

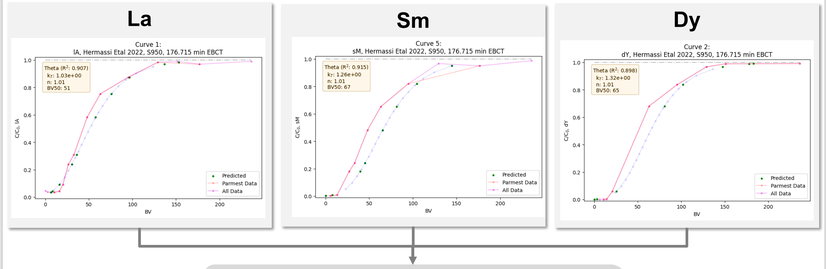
Parameter estimation determines:
- n Freundlich constant
- k_T Mass transfer coefficient
- BV_{50} Bed volume at 50% breakthrough

Data from **breakthrough curves**[4]:
Lanthanum (La)
Dysprosium (Dy)
Samarium (Sm)
Ytterbium (Yb)
Holmium (Ho)
Erbium (Er)
Need curve **for every ion-resin** pair

Absorption extent is determined by breakthrough curves and is highly dependent on **experimental conditions**

Flowsheet Development

Results from parameter estimation



Set of n, k_T, and BV_{50} values at given resin and column conditions

- Simulation and optimization example to determine **design parameters and operating conditions** to achieve high REE removal
- **Added IX to UKy flowsheet** as a pre-treatment step to remove **targeted ions**

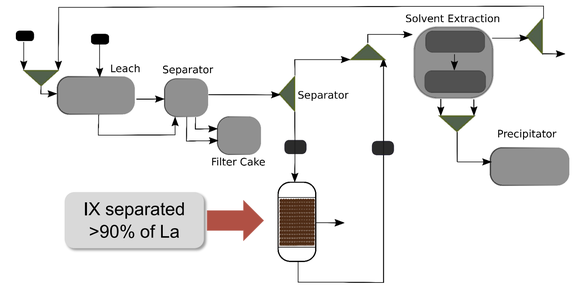


Fig 8. Fragment of UKy flowsheet integrated with IX for the separation of REEs from a coal waste stream.

Future Work

- IX model validation using experimental data.
- Develop additional case studies to answer guiding questions.
- Continue building collaborations with research community and leverage existing capabilities and data: WaterTAP, NETL, Wayne State University

Acknowledgements

This effort was funded by the U.S. Department of Energy's Process Optimization and Modeling for Minerals Sustainability (PrOMMiS) Initiative, supported by the Office of Fossil Energy and Carbon Management's Office of Resource Sustainability. For more information, please reach out to Thomas Tarka, PrOMMiS Technical Director (Thomas.Tarka@netl.doe.gov)

References

[1] Tech Brief Four, A National Drinking Water Clearinghouse Factsheet, "Ion Exchange and Demineralization". May 1997
[2] Hermassi, Mehrez, et al. "Impact of functional group types in ion exchange resins on rare earth element recovery from treated acid mine waters." Journal of Cleaner Production 379 (2022): 134742.
[3] University of Kentucky flowsheet from PrOMMiS GitHub repository at https://github.com/prommis/prommis
[4] Croll, Henry C., et al. "Fundamental kinetic constants for breakthrough of per-and polyfluoroalkyl substances at varying empty bed contact times: Theoretical analysis and pilot scale demonstration." Chemical Engineering Journal 464 (2023): 142587.
[5] Parmest example from WaterTAP GitHub repository: https://github.com/watertap-org/watertap/blob/pfas_treatment_analysis/watertap/examples/flowsheets/pfas_treatment_analysis/ix/IXParmest.py
[6] Property package that computes a multi-component aqueous solution from WaterTAP GitHub repository: https://github.com/watertap-org/watertap/property_models/multicomp_aq_sol_prop_pack.py

NETL NATIONAL ENERGY TECHNOLOGY LABORATORY · BERKELEY LAB · Sandia National Laboratories · Carnegie Mellon · WestVirginiaUniversity · UNIVERSITY OF NOTRE DAME · Georgia Tech · U.S. DEPARTMENT OF ENERGY