Lima Research Group
Control,
Optimization, and
Design for
Energy and
Sustainability

Recent Developments and Implementations in Process Operability Analysis

Victor Alves

Department of Chemical and Biomedical Engineering
West Virginia University

Purdue University - PSE Seminar
February 9 , 2024

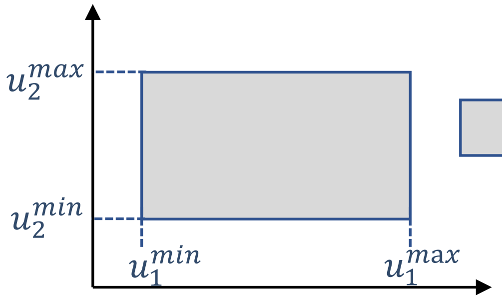
Outline

- Process Operability Concepts
  - NLP-Based Approach
  - Multimodel Approach
- Implicit Mapping
- Opyrability Development
  - Motivation
  - Software infrastructure
- Case Study: Direct Methane Aromatization Membrane Reactor (DMA-MR)
  - Multimodel representation
  - Inverse mapping
  - Implicit mapping
- Conclusions and Future Work

Process Operability Concepts

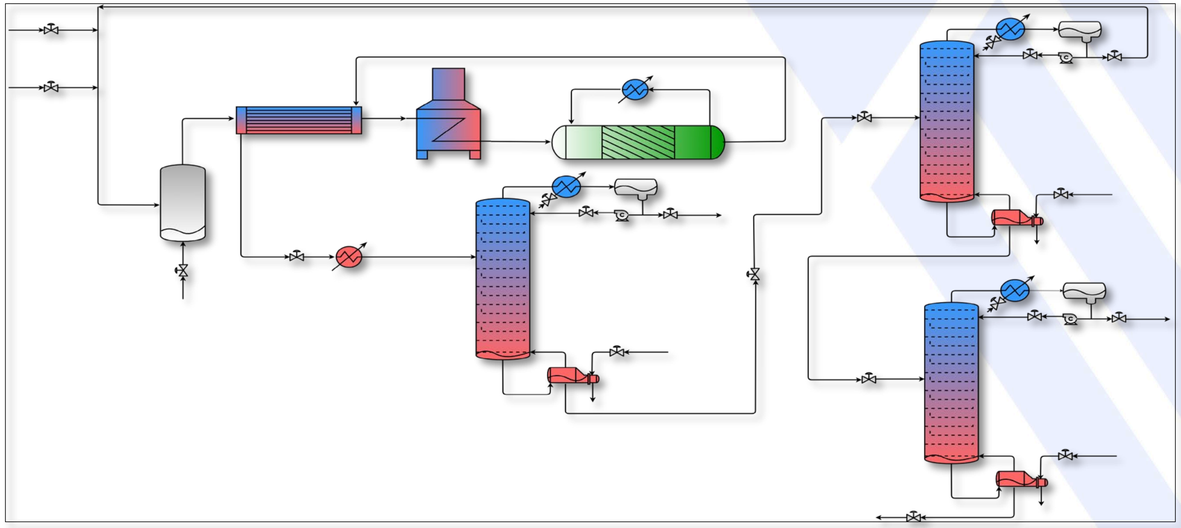
Process operability: A nonlinear measure of controllability and achievability

- Process Operability was introduced as a viable alternative to the sequential tasks of assessing process design and control, integrating both in the early design phase of industrial processes[1]
  - Fundamental initial idea: “A measure of output controllability”[2]
- Achievability of process design and control objectives are quantified using defined operating regions that are calculated considering
  - Available inputs, achievable outputs
  - Expected disturbances
  - Desired outputs/inputs of industrial processes

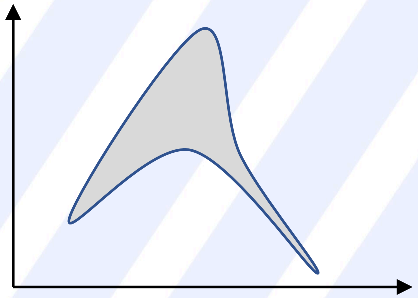


Available Input Set (AIS)

$\left\{u \mid u_i^{min} \leq u_i \leq u_i^{max}\right\}$



Process model (M)
First principles, process simulator, surrogate



Achievable Output Set (AOS)

$y = M(\mathrm{u}, \mathrm{d})$

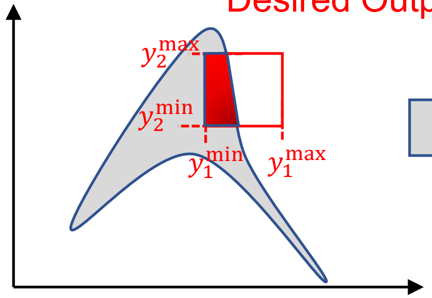
Under nominal disturbances

[1] Gazzaneo V. and Lima F. V. (2019). *Ind. Eng. Chem. Res.*
[2] Vinson, D. R. and Georgakis, C. (2000). *J. Process Control.*

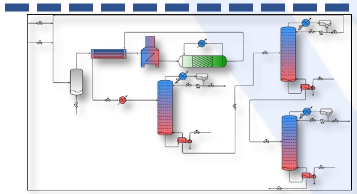
Process Operability Concepts

$\{y \mid y_i^{\min} \le y_i \le y_i^{\max}\}$

Desired Output Set (DOS)



Achievable Output Set (AOS)



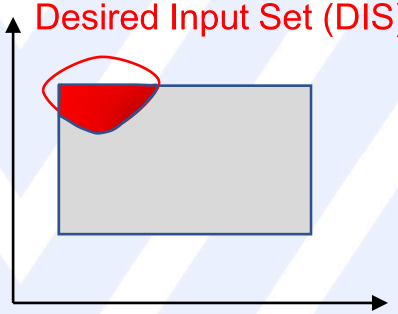
Process inverse mapping

$$M^{-1}$$

Under nominal disturbances

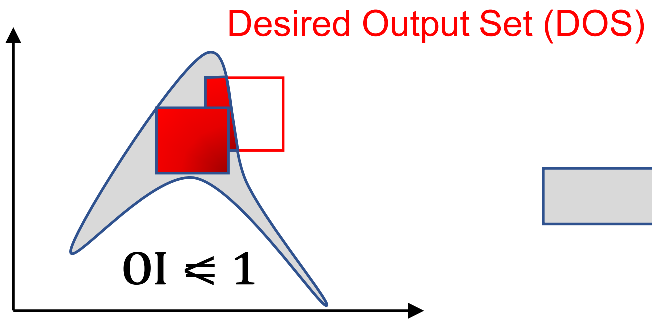
$\mathbf{u} = M^{-1}(y, \mathrm{d})$, **d fixed**

Desired Input Set (DIS)



Available Input Set (AIS)

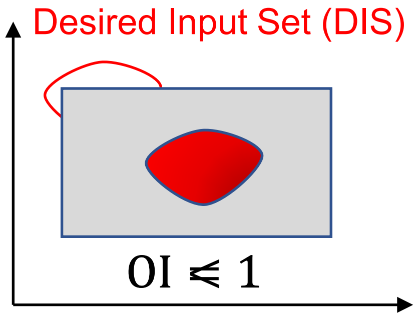
Process Operability Concepts



Achievable Output Set (AOS)

$$\mathrm{OI} = \frac{\mu(\mathrm{AOS} \cap \mathrm{DOS})}{\mu(\mathrm{DOS})}$$

Operability
Index (OI)



Available Input Set (AIS)

$$\mathrm{OI} = \frac{\mu(\mathrm{AIS} \cap \mathrm{DIS})}{\mu(\mathrm{DIS})}$$

μ = quantification of regions (length, area, volume, hypervolume)

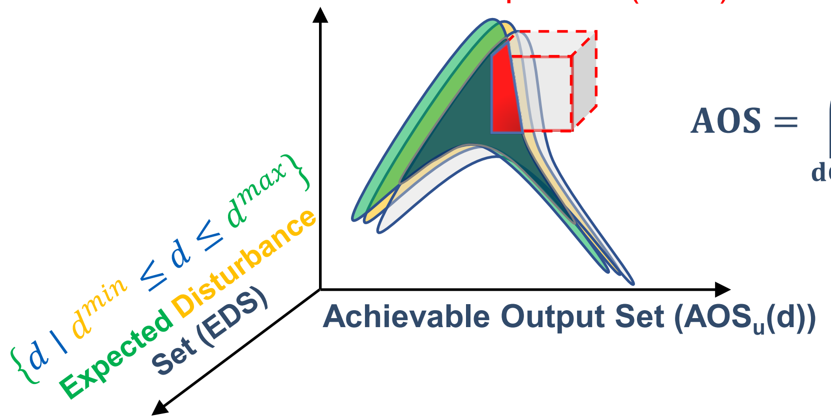
OI main features:

1. Inherently nonlinear measure
2. Independent of the type of the controller used and inventory control layer[1]
3. Can be used systematically to rank competing designs[2] and/or control structures
4. Allows for disturbances' evaluation under "worst-case" scenario situations

[1] Vinson, D. R. and Georgakis C. (2002). *Ind. Eng. Chem. Res.*
[2] Lima, F. V., Jia, Z., Ierapetritou, M. and Georgakis, C. (2010). *AIChE Journal.*

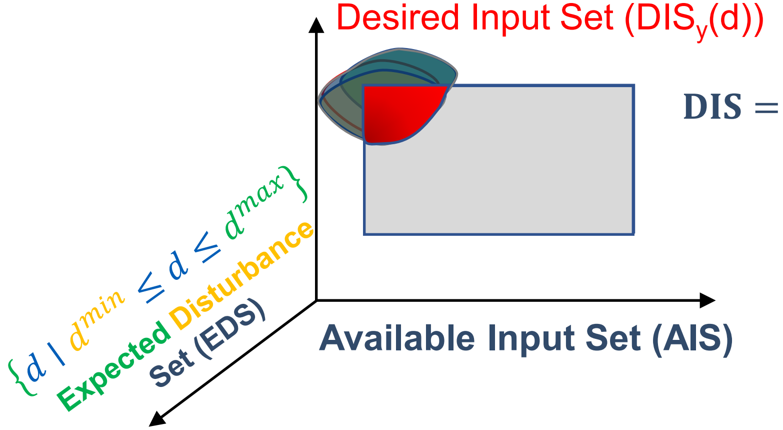
Process Operability Concepts

Introducing the effect of disturbances – Expected Disturbance Set (EDS)



$$\mathbf{AOS} = \bigcap_{\mathbf{d} \in \mathbf{EDS}} \mathbf{AOS_u(d)}$$

$$\mathrm{OI} = \frac{\mu(\mathrm{AOS} \cap \mathrm{DOS})}{\mu(\mathrm{DOS})}$$



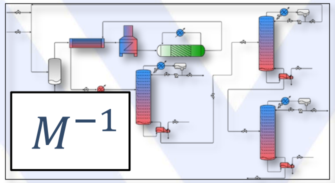
$$\mathbf{DIS} = \bigcup_{\mathbf{d} \in \mathbf{EDS}} \mathbf{DIS_y(d)}$$

$$\mathrm{OI} = \frac{\mu(\mathrm{AIS} \cap \mathrm{DIS})}{\mu(\mathrm{DIS})}$$

Operability Index (OI)

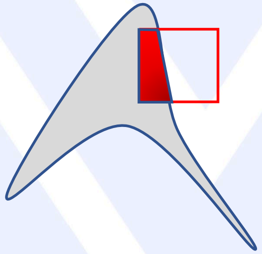
Main computational operations required

$\mathrm{DOS} \rightarrow \mathrm{DIS}$



Inverse mapping

Nonlinear programming (NLP)-based approach[1]



OI quantification

Multimodel approach[2]

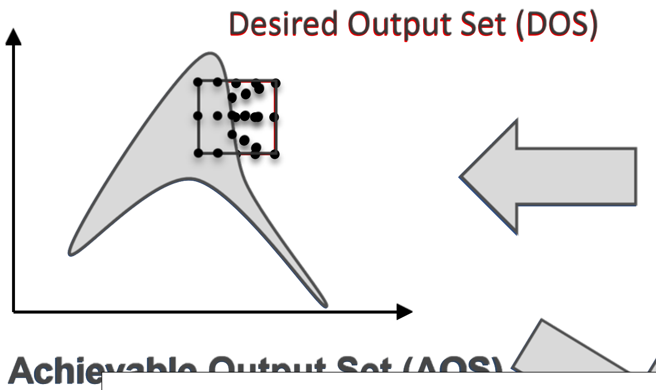
[1] Carrasco J. C. and Lima F.V. (2017). *Comput. Chem. Eng.*
[2] Gazzaneo V. and Lima. F.V. (2019). *Ind. Eng. Chem. Res.*

Process Operability Concepts

Nonlinear programming-based (NLP) approach[1]

Inverse mapping

Minimize the distance between desired (y^*) and actual (y) operation for each AOS/DOS point



$$\emptyset_k = \min_{u_k^*} \sum_{j=1}^{n} \left((y_{j,k} - y_{j,k}^*)/y_{j,k} \right)^2$$

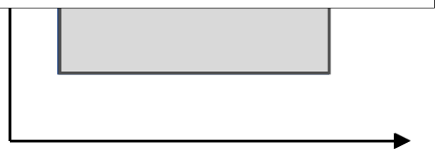
s.t: Process model

$$u_k^{min} \leq u_k^* \leq u_k^{max}$$

$$c_1(u_k^*) \leq 0$$

Takeaways:

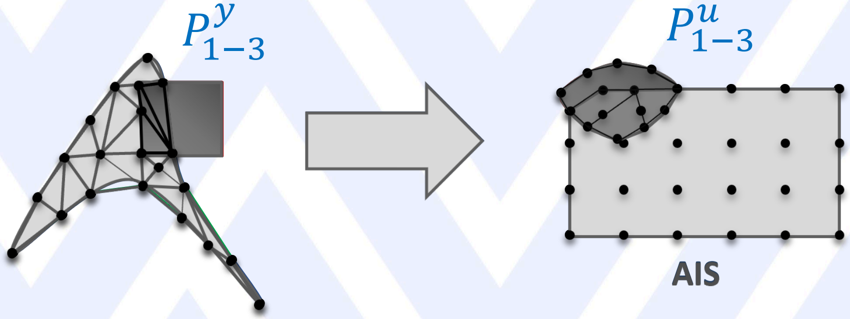
- **Shift the output points as close as possible to feasible operation**
- **Useful to search for new AIS unexplored regions to give insights about process feasibility**
- **Inverse mapping subject to process constraints**



Multimodel approach[2]

OI quantification

Process model can be substituted by paired polytopes $P_k = \{P_k^u, P_k^y\}$



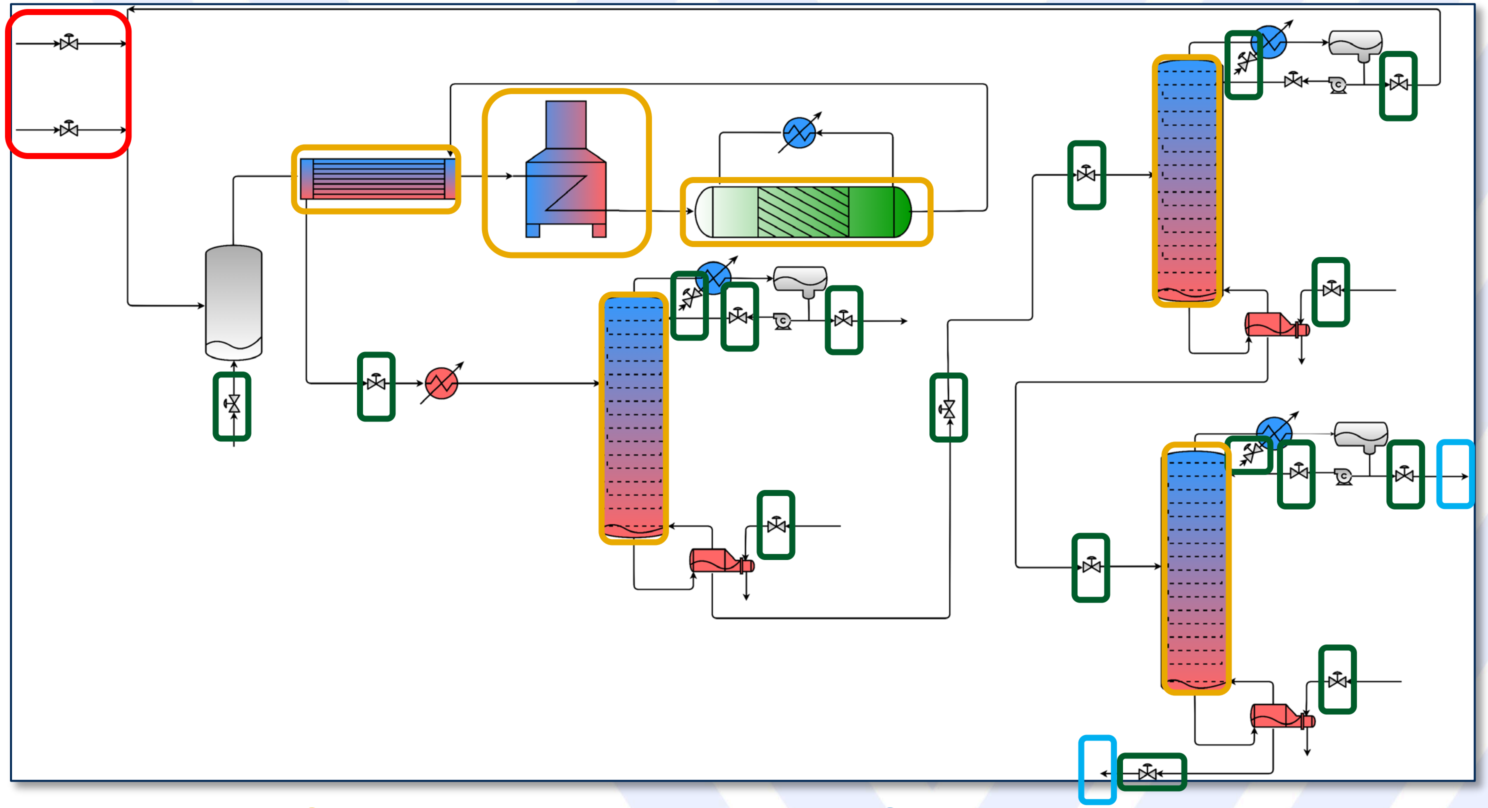
Takeaways:

- **Replacing nonconvex regions with paired polytopes allows efficient OI computation and representation of the operability sets**
- **OI can be used to rank competing designs and control structures**

[1] Carrasco J. C. and Lima F.V. (2017). *Comput. Chem. Eng.*
[2] Gazzaneo V. and Lima. F.V. (2019). *Ind. Eng. Chem. Res.*

Implicit Mapping - Motivation

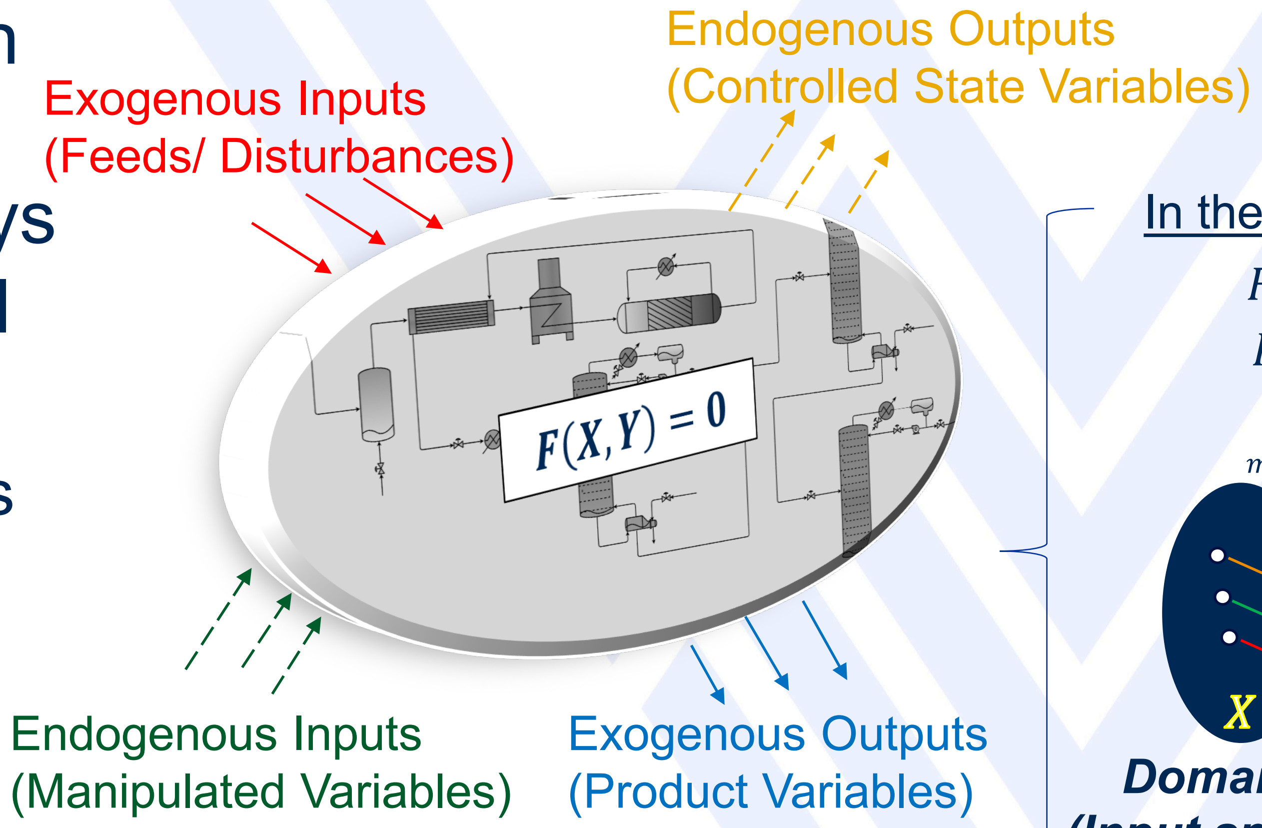
- Mapping of variables in process systems engineering (PSE) plays a vital role in additional important applications:
  - Optimization of process design/operating conditions[1]
  - Operability analysis[2]
  - Parameter estimation[3]
  - Control structure[4] selection

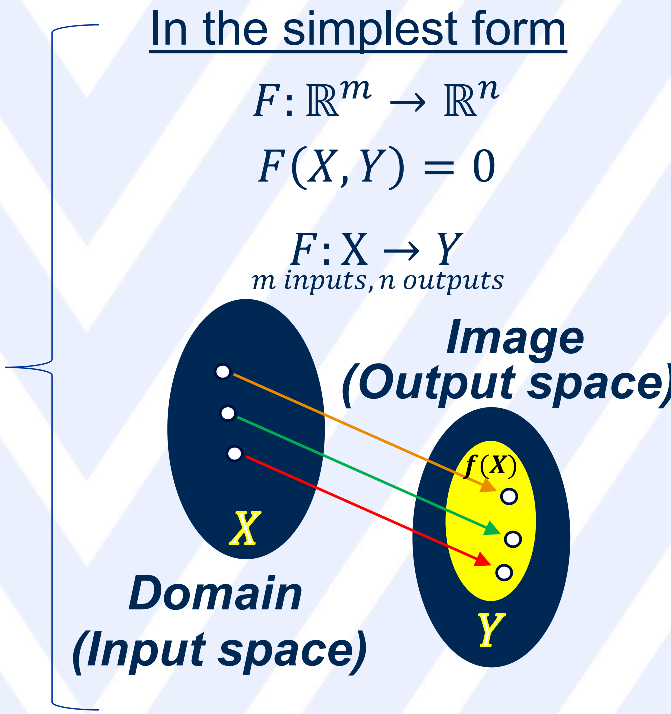


[1] Biegler L. T. (2010). *SIAM.*
[2] Vinson D. R. and Georgakis, C. (2000). *Journal of Proc. Control.*
[3] Aster R. C., Borchers B., Thurber C. H. (2018). *Elsevier.*
[4] Skogestad S. (2000). *Journal of Proc. Control.*

Implicit Mapping - Motivation

- Mapping of variables in process systems engineering (PSE) plays a vital role in additional important applications:
  - Optimization of process design/operating conditions[1]
  - Operability analysis[2]
  - Parameter estimation[3]
  - Control structure[4] selection





[1] Biegler L. T. (2010). *SIAM.*
[2] Vinson D. R. and Georgakis, C. (2000). *Journal of Proc. Control.*
[3] Aster R. C., Borchers B., Thurber C. H. (2018). *Elsevier.*
[4] Skogestad S. (2000). *Journal of Proc. Control.*

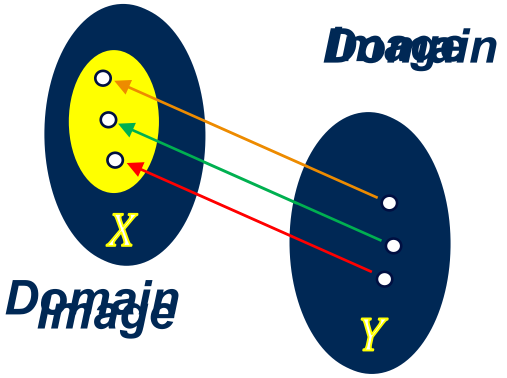
Implicit Mapping - Motivation

In the simplest form

$$F: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

$$F(X, Y) = 0$$

$$F: \mathrm{X} \rightarrow Y$$



- Forward mapping applications are typically straightforward
  - Sensitivity analysis
  - Sequential (classic) process design and control
- Inverse mapping applications are more complex
  - Inverse problems naturally arise[1]
  - General parameter estimation problems[2]
  - Optimal operability design regions from desired output specifications[3,4]
- The forward mapping is adequately understood and studied due to historical reasons[1], but the inverse is not necessarily

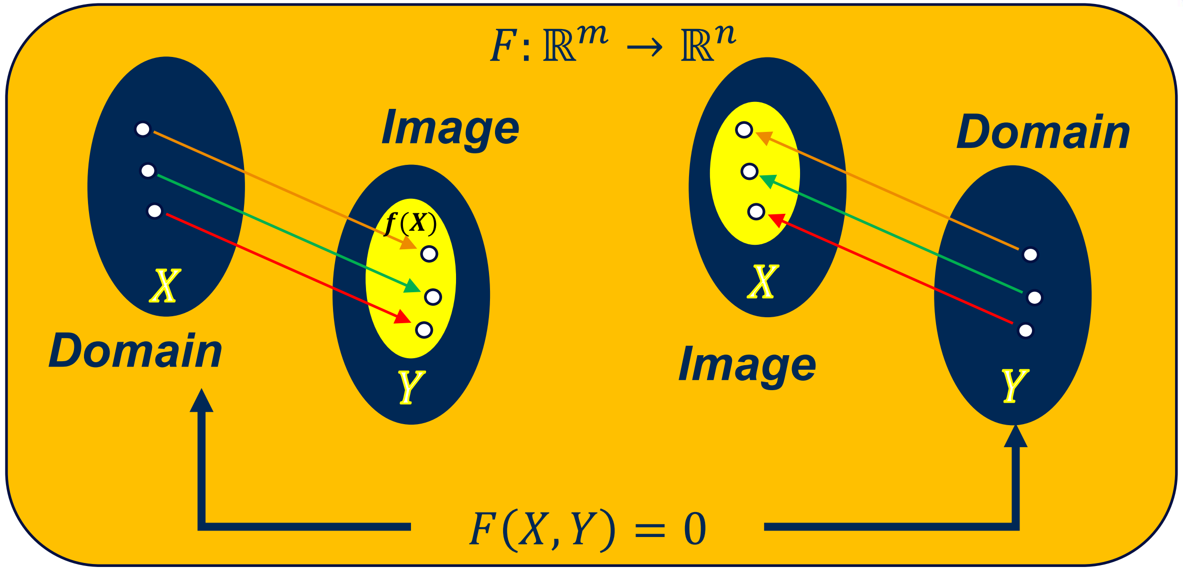
Typical approaches

- "Brute-force" (enumeration) in the forward mapping direction (e.g., *"lookup table")*: Feasible when the number of simulations/dimensionality are low
- Nonlinear programming-based (NLP)[3,4,5] formulations: Can be computationally expensive

[1] Keller J.B. Inverse problems. (1976) *Am Math Mon.*
[2] Aster R. C., Borchers B., Thurber C. H. (2018). *Elsevier.*
[3] Carrasco J. C., Lima F. V. (2017) *AIChE Journal.*
[4] Carrasco J. C., Lima F. V. (2018) AIChE Journal.
[5] Ceccon F. et al. (2022). *Journ. of Machine Learning. Res.*

Implicit Mapping - Motivation

- Process models are often implicit and forward/inverse mapping tasks are needed
  - Flexibility between forward and inverse maps may be required



- Under sufficient conditions[1], a vector-valued, implicit map can be differentiated using the implicit function theorem (IFT), but accurate derivatives are required

$$\nabla_X Y = -(\nabla_y F)^{-1} J_x$$

<u>Automatic differentiation (AD) can be employed</u>

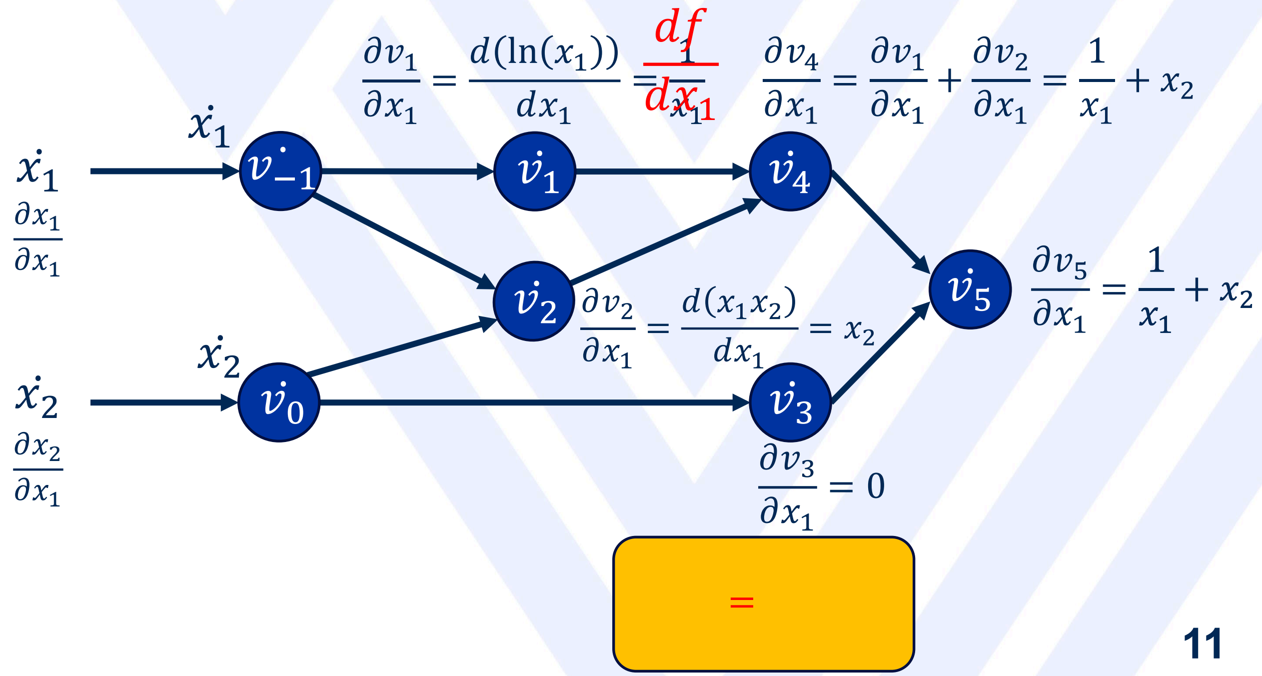
Effective way of accurately and timely evaluate high-order data, when compared against finite-differences and symbolic differentiation

<u>AD in a nutshell: Intelligent use of the chain rule in computer code:</u>

$$\underbrace{f(x_1, x_2)}_{v_{-1}, v_0} = \underbrace{\ln(x_1)}_{v_1} + \underbrace{x_1 x_2}_{v_2} - \underbrace{\sin(x_2)}_{v_3}$$ [2]

$v_4 = v_1 + v_2$

$v_5 = v_4 - v_3$



[1] Folland G. (2002). *Advanced Calculus. Prentice Hall.*
[2] Baydin A. G. *et al.* (2018). arXiv.

Implicit Mapping - Motivation

- Formulation of a framework for the implicit mapping evaluation employing the implicit function theorem and automatic differentiation as an alternative to the NLP-based approach
  - Recent advances in differentiable programming and automatic differentiation (AD) made the evaluation of high-dimensional, implicit derivatives a more straightforward task
  - If derivatives are readily available, the implicit mapping task can be performed directly via path integration from the output space to the input space (and vice-versa) with the aid of the implicit function theorem
- AD does not suffer from problems of finite-differences or symbolic differentiation, such as
  - Round-off errors (inaccuracies)[1]
  - Expensive calculations (for example, expression swell)[1]
- AD is increasingly popular in open-source programming languages (e.g., Python)
  - Google's JAX[2]
  - JAX is becoming an ecosystem for differentiable programming[3]

[1] Baydin A. G. *et. al. (2018). arXiv.*
[2] Bradbury et al. (2018). http://github.com/google/jax.
[3] Vadivelu N. (2023). https://github.com/n2cholas/awesome-jax.

Proposed Approach - Inverse mapping[1]

Set the model in implicit form

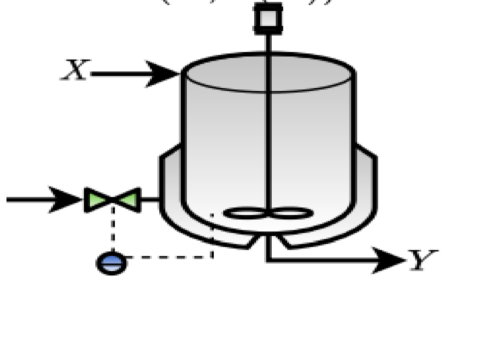
$$F(X,Y)=0$$

And use JAX's syntax

If a differentiable model cannot be developed, employ a differentiable surrogate[2]

Differentiable Process Model

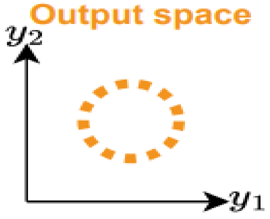
$$F(X,Y(X))=0$$



Can be expanded for any … functions of m inputs and n ou…

$$\nabla_y X = -J_x^{-1}(\nabla_y F)$$

1. **Perform differentiable modeling** — Process model to be compatible with automatic differentiation.
2. **Define output space region** — Define a boundary that represents desired operating region.
3.
4. **Obtain inverse derivatives** — Apply the implicit function theorem to obtain $\frac{dx_m}{dy_n}$: $\nabla_y X = -J_x^{-1}(\nabla_y F)$
5. **Perform Inverse mapping** — Use the high-order data to integrate along the output boundary to input space using an integration scheme.

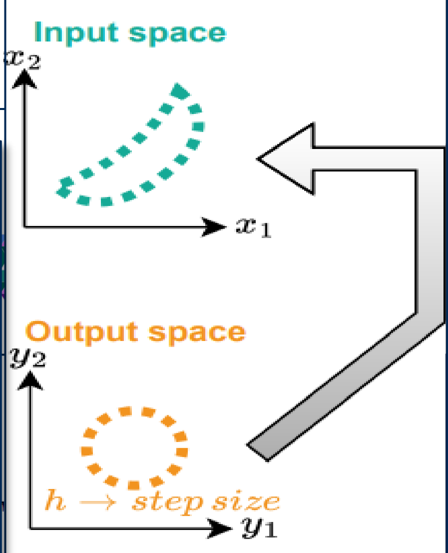


Parametric equations can be used:

- Circles
- Recte…

… supports AD of …

$$\nabla_y X \overset{\text{def}}{=} \begin{bmatrix} \frac{\partial x_1}{\partial y_1} & \cdots & \frac{\partial x_1}{\partial y_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial y_1} & \cdots & \frac{\partial x_m}{\partial y_n} \end{bmatrix}; J_x \overset{\text{def}}{=} \begin{bmatrix} \cdots \\ \frac{\partial F_n}{\partial y_1} & \cdots & \frac{\partial F_n}{\partial y_n} \end{bmatrix}$$

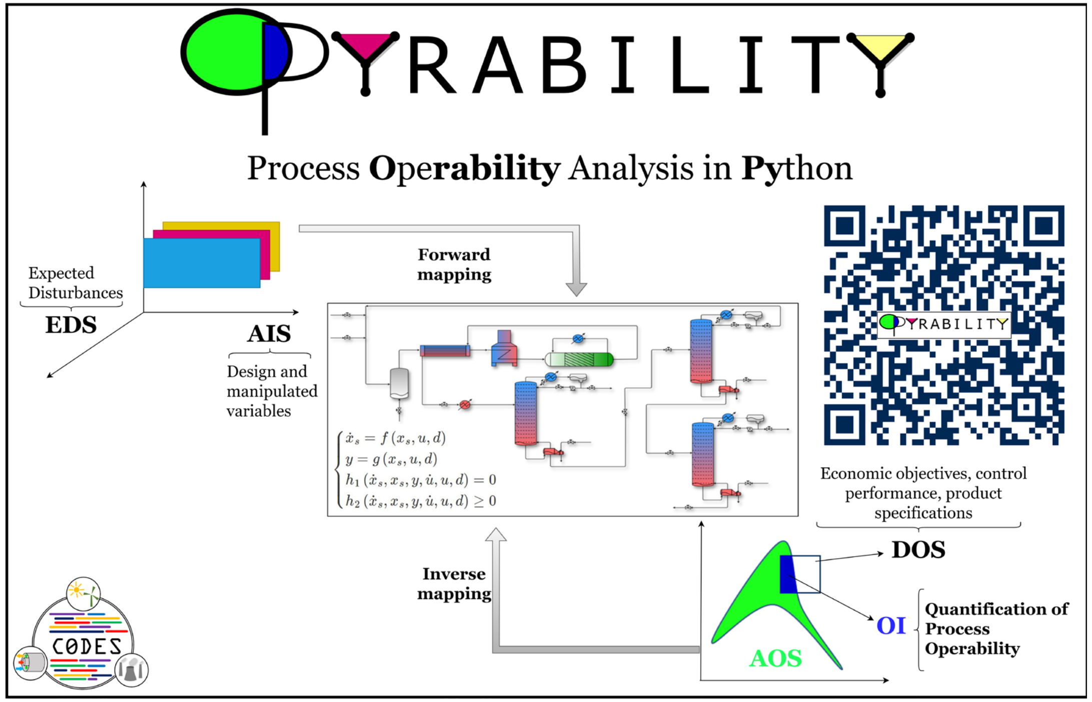


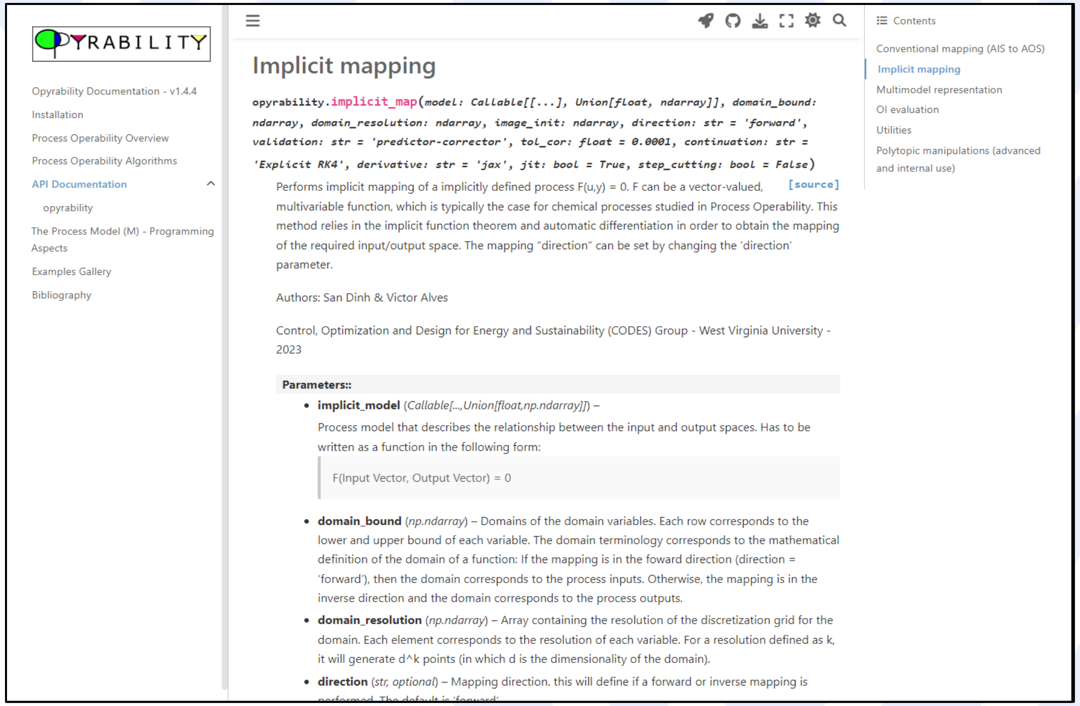
$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$
$$y_{n+1} = y_n + h$$

[1] Alves V. , Kitchin J. R. and Lima F. V. (2023). *AIChE Journal.*
[2] Pinder T. and Dodd D. (2022). *Journal of Open-Source Soft.*

Opyrability[1] Development: Motivation

- Programming aspects of Process Operability calculations
  - NLP-based approach: needs ***reliable NLP solvers***
  - Multimodel approach: ***computational geometry*** packages for boolean operations of sets (intersection, union, difference) ***are needed – non-trivial task***
  - ***Recent advances in inverse mapping using AD***





[1] Alves V., Dinh S., Kitchin J. R., Gazzaneo V., Carrasco J. C. and Lima F. V. (2024). Journ. of Open Source Soft.

Opyrability[1] Development: Motivation

- Programming aspects of Process Operability calculations
  - NLP-based approach: needs ***reliable NLP solvers***
  - Multimodel approach: ***computational geometry*** packages for boolean operations of sets (intersection, union, difference) ***are needed – non-trivial task***
  - ***Recent advances in inverse mapping using AD***

- Current CODES MATLAB App Project[2] uses
  - A modified version of Nelder-Mead downhill simplex with penalty functions to deal with nonlinear inequality/equality constraints[3]
  - Multi-parametric toolbox (MPT)[4] for computational geometry calculations
    - MPT has challenges when scale of the OI required calculations involve dimensions higher than 6-7
  - Nelder-Mead simplex with penalty functions might not be the most robust choice (derivative-free)
  - The MATLAB App Project ***is open-source***, but user ***needs*** a MATLAB license

[1] Alves V., Dinh S., Kitchin J. R., Gazzaneo V., Carrasco J. C. and Lima F. V. (2024). Journ. of Open Source Soft.
[2] Gazzaneo V., Carrasco J.C., Vinson D. R. and Lima F. V. (2020). *Ind. Eng. Chem. Res.*
[3] D'Errico J. (2012). *MATLAB File exchange*
[4] Multi-Parametric Toolbox 3. (2019). https://www.mpt3.org/

Opyrability[1] Development: Motivation

- Current effort: recreate the app project as a Python package (namely Opyrability – Python-based Process Operability) with enhanced features and recent developments
- Features of the proposed toolbox
  - Free and open-source programming language (Python)
  - Easy access to up-to-date, state-of the art optimization solvers (*IPOPT*[2], etc.)
  - Easier code maintainability
  - Collaborative/community-driven approach (hosting on *GitHub*)
    - Facilitate academia access, bugfixes, suggestions, pull requests, issues, etc.
- Process Operability is a versatile field with broad applications: Python implementation can ease its dissemination in academia/industry
  - Design and control structure synthesis[3]
  - System modularization[4,5]
  - Process intensification[4,5]

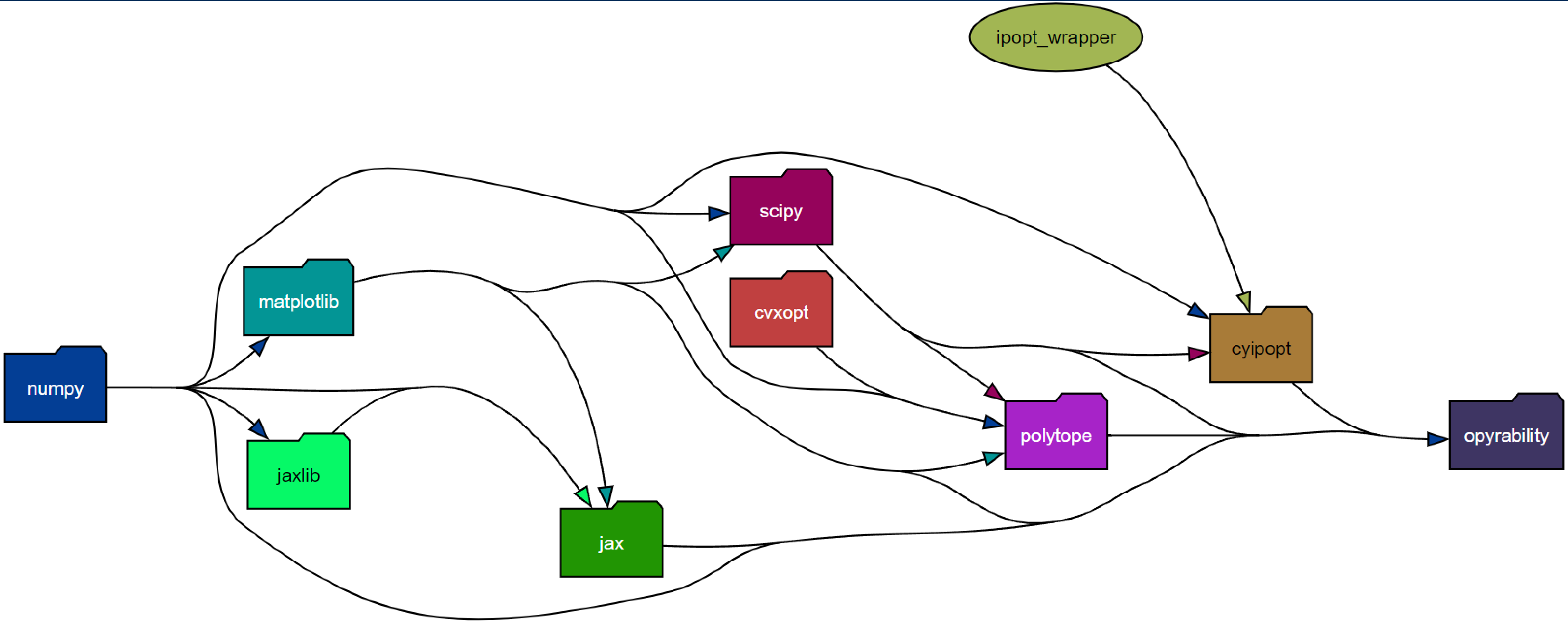
[1] Alves V., Dinh S., Kitchin J. R., Gazzaneo V., Carrasco J. C. and Lima F. V. (2024). Journ. of Open Source Soft.
[1] Wächter A. and Biegler L. T. (2006). *Math. Prog*.
[2] Lima F.V. And Georgakis C. (2010). Journal of Proc. Cont.
[3] Carrasco J. C. and Lima F.V. (2017). *Comput. Chem. Eng.*
[4] Gazzaneo V. and Lima. F.V. (2019). *Ind. Eng. Chem. Res.*

Opyrability[1] Development

- NLP-based approach
  - Different solver options available:
    - **CYIPOPT**[2] (Python wrapper around IPOPT): compatible with *automatic differentiation* (Google's JAX[3])
    - Differential evolution[4]
    - Sequential quadratic programming
    - Nelder-Mead simplex
- Multimodel approach
  - Currently relies on Caltech's *Polytope*[5] package for computational geometry operations of polytopes (intersection/union/difference)
  - All fundamental operations implemented (OI calculation and multimodel representation)
- Implicit mapping
  - Uses JAX as AD library
- ***Current state:*** built from scratch using mainly *numpy, scipy*, supporting the main Process Operability calculations
  - C/C++ codes are "indirect" dependencies due to *polytope* and IPOPT

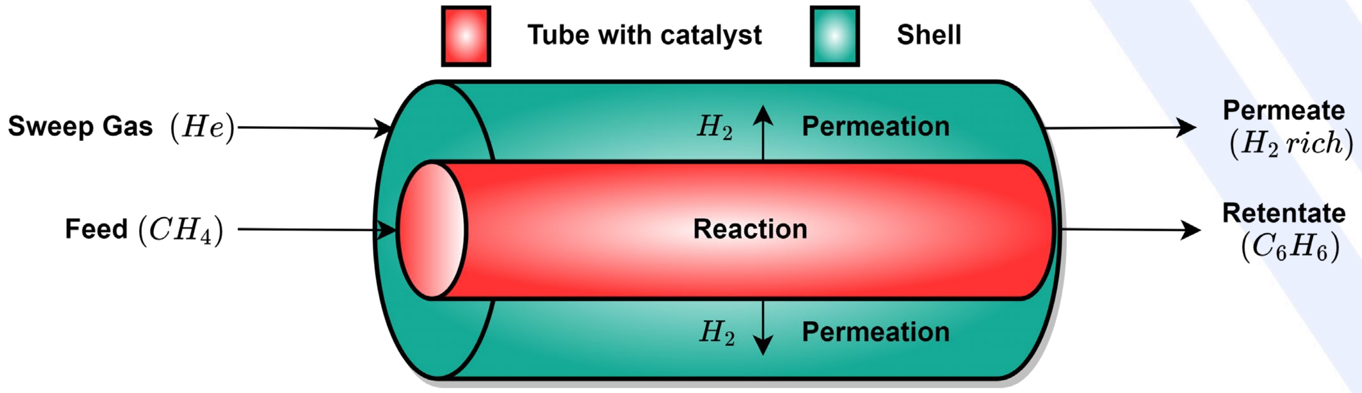
[1] Alves V., Dinh S., Kitchin J. R., Gazzaneo V., Carrasco J. C. and Lima F. V. (2024). Journ. of Open Source Soft.
[2] CYIPOPT GitHub Repository. (2022). https://github.com/mechmotum/cyipopt
[3] Bradbury et al. (2018). http://github.com/google/jax
[4] Storn R. and Price K. (1997). *Journ. of Glob. Opt.*
[5] Polytope GitHub Repository. (2022). https://github.com/tulip-control/polytope

Opyrability[1] Development – Software Infrastructure



[1] Alves V., Dinh S., Kitchin J. R., Gazzaneo V., Carrasco J. C. and Lima F. V. (2024). Journ. of Open Source Soft.

Case Study: Direct Methane Aromatization Membrane Reactor (DMA-MR)



Mechanism and Kinetics:

$$2CH_4 \rightleftharpoons C_2H_4 + 2H_2 \qquad r_1 = k_1 C_{CH_4} - k_1' \frac{C_{C_2H_4} C_{H_2}^2}{C_{CH_4}}$$

$$3C_2H_4 \rightleftharpoons C_6H_6 + 3H_2 \qquad r_2 = k_2 C_{C_2H_4} - k_2' \frac{C_{C_6H_6} C_{H_2}^3}{C_{C_2H_4}^2}$$

$$C = [mol/cm^3], k_i = [s^{-1}], k_i' = [cm^3/s.\,mol]$$

- Production of high-value added chemicals (C_6H_6 and H_2) from natural gas
- Combination of separation and reaction: higher conversion due to the selective H_2 removal[1] (Le Chatelier's principle)

Tasks and set up:

- Employ NLP-based approach to give insights about feasible designs
- Employ Multimodel approach to calculate the OI for a fixed design
- Employ Implicit mapping to inversely map design regions and disturbances
- Inputs – NLP-based and Implicit mapping:
  - Tube length [cm]
  - Tube diameter [cm]
- Inputs – Multimodel and Implicit mapping:
  - Shell and tube flow rates [mg/h]
- Outputs and desirable operating spaces:
  - Benzene production [mg/h]
  - Methane conversion [%]
- Process constraints:
  - Plug-flow operation: length/diameter ≥ 30
  - Maximum tube length: length < 300 [cm]

[1] Carrasco J. C. and Lima F.V. (2017). *Comput. Chem. Eng.*

Case Study: Direct Methane Aromatization Membrane Reactor (DMA-MR) - Multimodel

Operability Index (OI) using opyrability – "Control" problem, AIS has manipulated variables for a fixed MR design (tube length = 30 cm and tube diameter = 1 cm)

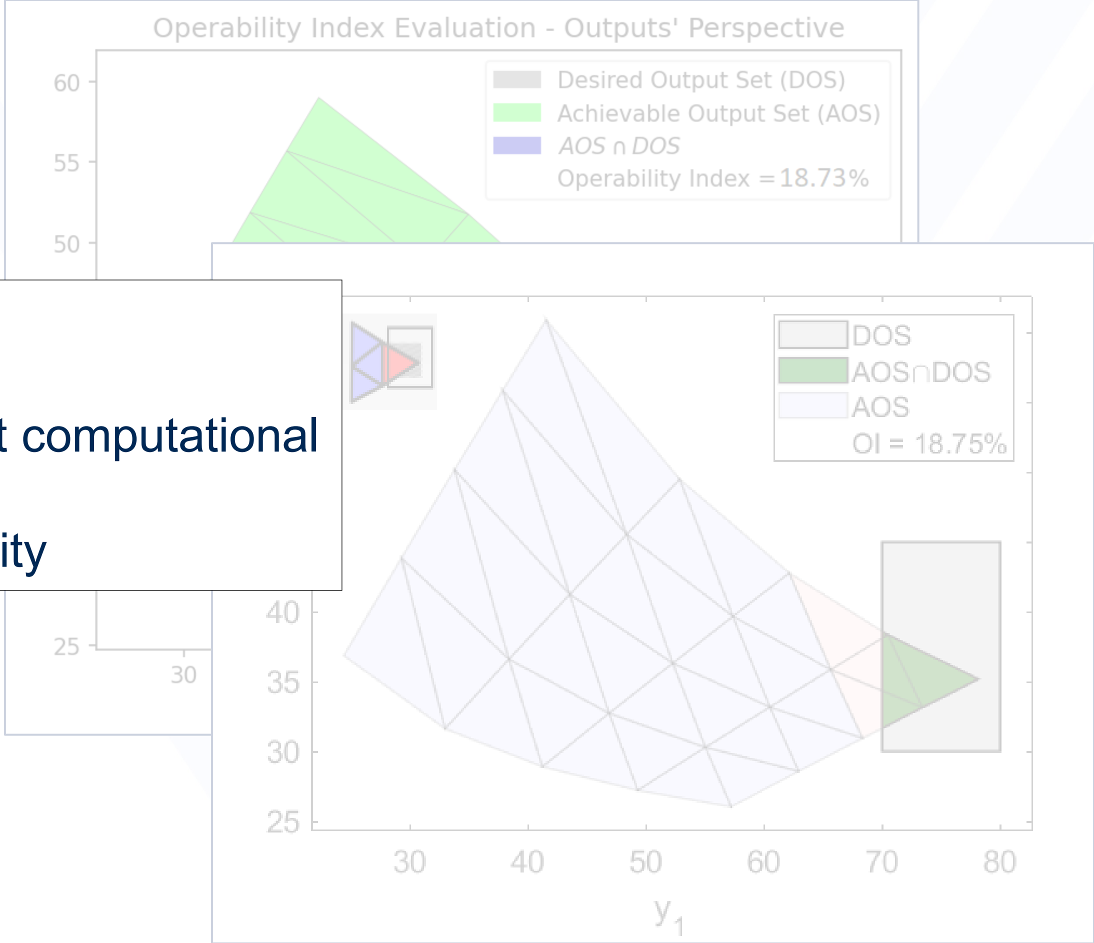
```python
from dma_mr import *
from opyrability import multimodel_rep, OI_eval
import numpy as np
```

Importing proposed tool and DMA-MR model

```python
# Defining AIS & DOS bounds.
DOS_bounds =  np.array([[70, 80],
                        [30, 45]])

AIS_bounds = np.array([[600, 1800],
                       [600, 1800]])
# Discretization Resolution.
AIS_resolution = [6, 6]

# Model assignment.
model = dma_mr_mv

# Obtain AOS.
AOS_region = multimodel_rep(model,
                            AIS_bounds,
                            AIS_resolution)

# Obtain Operability Index (OI).
OI = OI_eval(AOS_region, DOS_bounds)
```

Running multimodel representation and evaluating OI

Features:
- Easy setup for users
- Minimum knowledge about computational geometry required
- Seamless toolbox readability



Case Study: Direct Methane Aromatization Membrane Reactor (DMA-MR) – Inverse mapping

Operability analysis using opyrability – Inverse mapping, AIS has design variables for a nominal operating point

```python
from dma_mr import *
from opyrability import nlp_based_approach
import jax.numpy as np
```

Importing proposed tool and DMA-MR model

```python
# Lower and upper bounds for DOS definition.
DOS_bounds =  np.array([[15, 25],
                        [35, 45]])

# Discretization Resolution - 10x10 grid for DOS.
DOS_resolution = [10, 10]

# Lower and upper bounds of AIS (design)
lb = np.array([10, 0.1])
ub = np.array([300, 2])

# Initial estimate for NLP.
u0 = np.array([100, 1])
```

Defining AIS/DOS

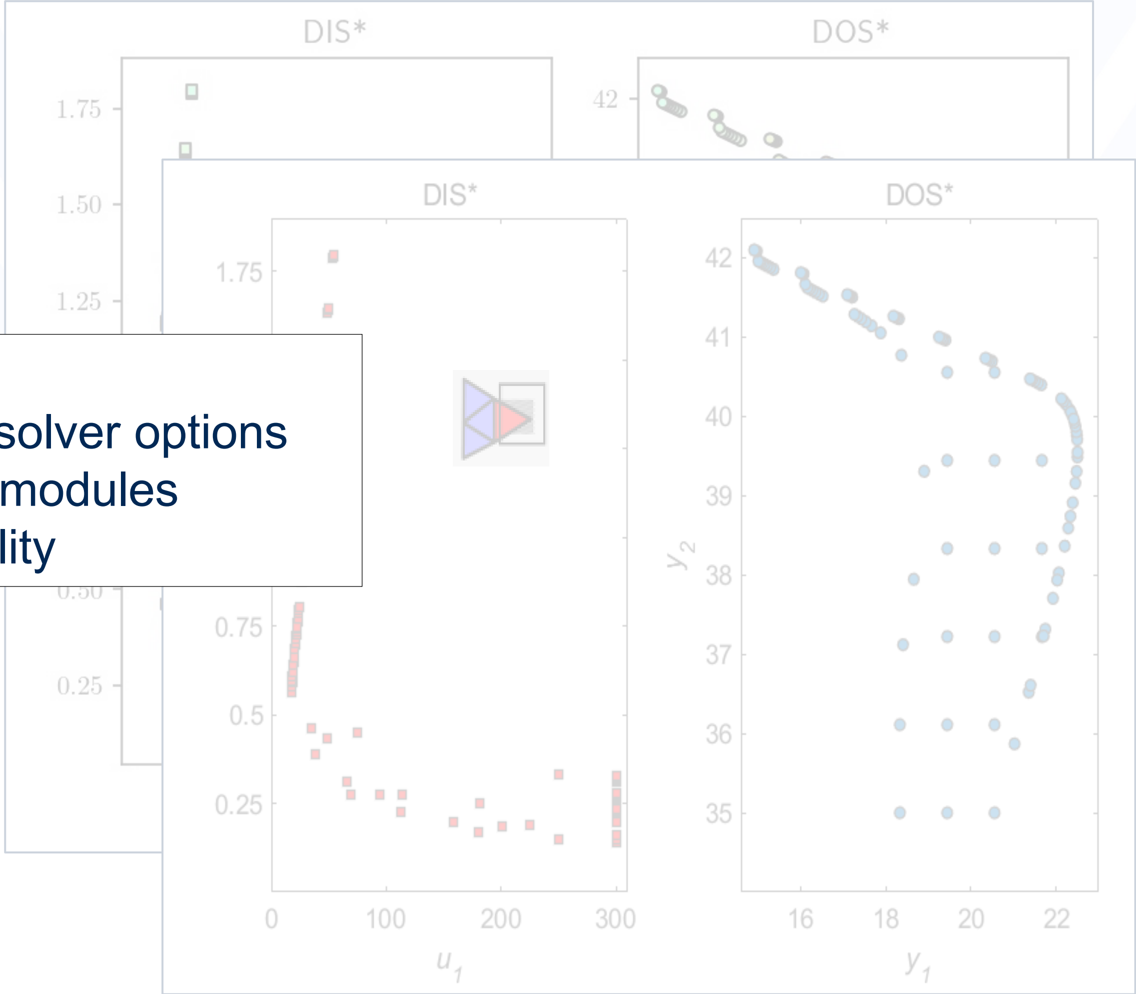
```python
# Plug-flow constraint definition: L/D >=30.
def plug_flow(u):
    return u[0] - 30.0*u[1]

con ={'type': 'ineq', 'fun': plug_flow}
# Model assignment - Design Problem - Inverse mapping.
model = dma_mr_design
```

Defining constraints

```python
# Obtain AOS.
fDIS, fDOS, _ = nlp_based_approach(model, DOS_bounds, DOS_resolution,
                                   u0, lb, ub,
                                   constr    = (con),
                                   method    = 'ipopt',
                                   plot      = True,
                                   ad        = True,
                                   warmstart = True)
```

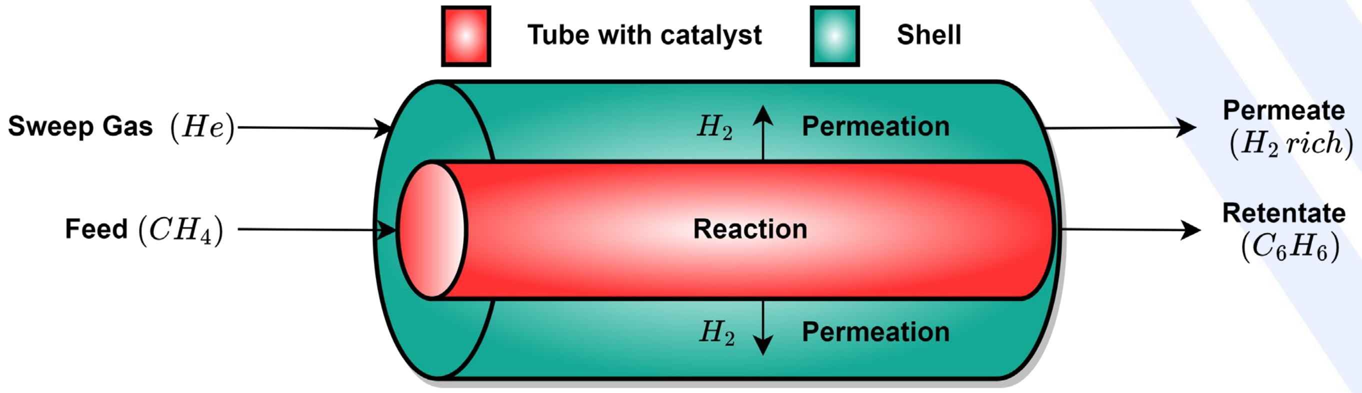
Running NLP-based approach with ***IPOPT/JAX***

Features:
- Highly customizable NLP solver options
- Consistent syntax among modules
- Seamless toolbox readability



Case Study – Implicit Mapping

- Direct Methane Aromatization Membrane Reactor (DMA-MR)



Mechanism and Kinetics:

$$2CH_4 \rightleftharpoons C_2H_4 + 2H_2 \qquad r_1 = k_1 C_{CH_4} - k_1' \frac{C_{C_2H_4} C_{H_2}^2}{C_{CH_4}}$$

$$3C_2H_4 \rightleftharpoons C_6H_6 + 3H_2 \qquad r_2 = k_2 C_{C_2H_4} - k_2' \frac{C_{C_6H_6} C_{H_2}^3}{C_{C_2H_4}^2}$$

$$C = [mol/cm^3], k_i = [s^{-1}], k_i' = [cm^3/s.mol]$$

- Production of high-value added chemicals (C_6H_6 and H_2) from natural gas
- Combination of separation and reaction: higher conversion due to the selective H_2 removal[1] (Le Chatelier's principle)

Main Objective: Identify design regions ensuring desired benzene production and natural gas conversion, addressing the inverse problem in process operability analysis[2]

<u>Problem set up:</u>

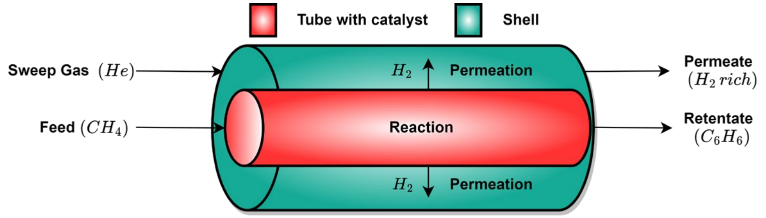
- Input variables (Image – Desired Input Set[2]):
  - Tube length [cm]
  - Tube diameter [cm]
- Outputs variables (Domain – Desired Output Set[2]):
  - Benzene production [mg/h]
  - Methane conversion [%]

- First-principles model defined by a set of 8 ordinary differential equations (for a distributed system)

[1] Carrasco J. C. and Lima F.V. (2017). *Comput. Chem. Eng.*
[2] Vinson D. R. and Georgakis C. (2000). *Journal of Proc. Control.*

Case Study – Implicit Mapping

- Direct Methane Aromatization Membrane Reactor (DMA-MR)



Mechanism and Kinetics:

$$2CH_4 \rightleftharpoons C_2H_4 + 2H_2 \qquad r_1 = k_1 C_{CH_4} - k_1' \frac{C_{C_2H_4} C_{H_2}^2}{C_{CH_4}}$$

$$3C_2H_4 \rightleftharpoons C_6H_6 + 3H_2 \qquad r_2 = k_2 C_{C_2H_4} - k_2' \frac{C_{C_6H_6} C_{H_2}^3}{C_{C_2H_4}^2}$$

$$C = [mol/cm^3],\ k_i = [s^{-1}],\ k_i' = [cm^3/s.\,mol]$$

- First-principles model defined by a set of 8 ordinary differential equations (for a distributed system)

$$\frac{dF_{t,CH_4}}{dz} = \eta r_1 A_t - \frac{Q}{\alpha_{H_2/CH_4}}\left(P_{t,CH_4}^{1/4} - P_{s,CH_4}^{1/4}\right)\pi D_t, \qquad \frac{dF_{s,CH_4}}{dz} = \frac{Q}{\alpha_{H_2/CH_4}}\left(P_{t,CH_4}^{1/4} - P_{s,CH_4}^{1/4}\right)\pi D_t,$$

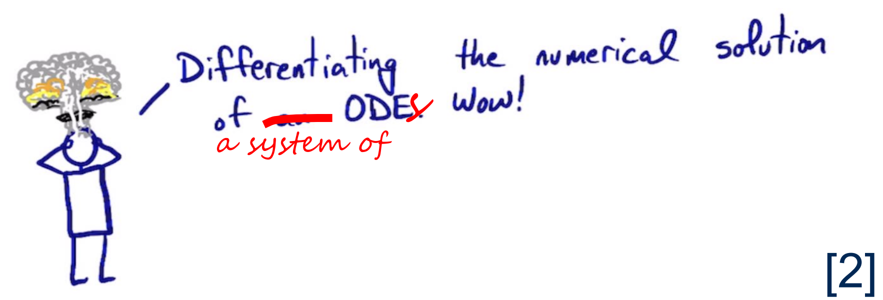
$$\frac{dF_{t,C_2H_4}}{dz} = -\eta\frac{r_1}{2}A_t + \eta r_2 A_t - \frac{Q}{\alpha_{H_2/C_2H_4}}\left(P_{t,C_2H_4}^{1/4} - P_{s,C_2H_4}^{1/4}\right)\pi D_t, \qquad \frac{dF_{s,C_2H_4}}{dz} = \frac{Q}{\alpha_{H_2/C_2H_4}}\left(P_{t,C_2H_4}^{1/4} - P_{s,C_2H_4}^{1/4}\right)\pi D_t,$$

$$\frac{dF_{t,H_2}}{dz} = -\eta r_1 A_t - \eta r_2 A_t - Q\left(P_{t,H_2}^{1/4} - P_{s,H_2}^{1/4}\right)\pi D_t, \qquad \frac{dF_{s,H_2}}{dz} = Q\left(P_{t,H_2}^{1/4} - P_{s,H_2}^{1/4}\right)\pi D_t,$$

$$\frac{dF_{t,C_6H_6}}{dz} = -\eta\frac{r_1}{3}A_t - \frac{Q}{\alpha_{H_2/C_6H_6}}\left(P_{t,C_6H_6}^{1/4} - P_{s,C_6H_6}^{1/4}\right)\pi D_t, \qquad \frac{dF_{s,C_6H_6}}{dz} = \frac{Q}{\alpha_{H_2/C_6H_6}}\left(P_{t,C_6H_6}^{1/4} - P_{s,C_6H_6}^{1/4}\right)\pi D_t$$

Some important properties/facts

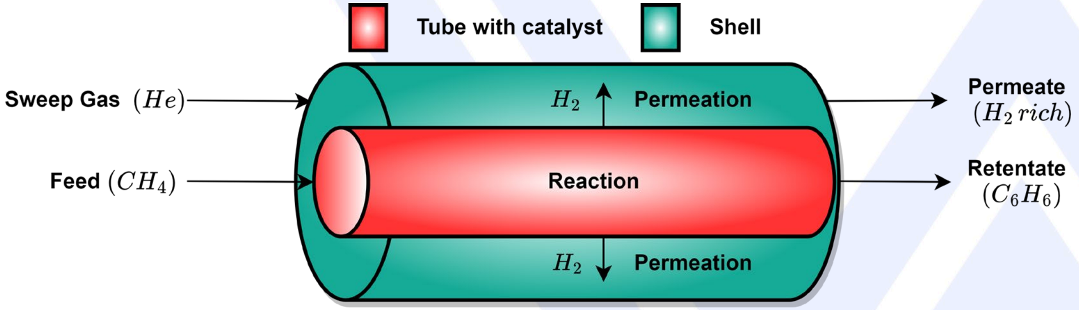
- The system needs to be numerically integrated (Using JAX's implementation of Dormand Prince[1])
- Non-convexities are present mainly due to the permeation terms ($^{1/4}$ power terms)
- Application of the proposed approach differentiates the integrated solution of the system of ordinary differential equations!



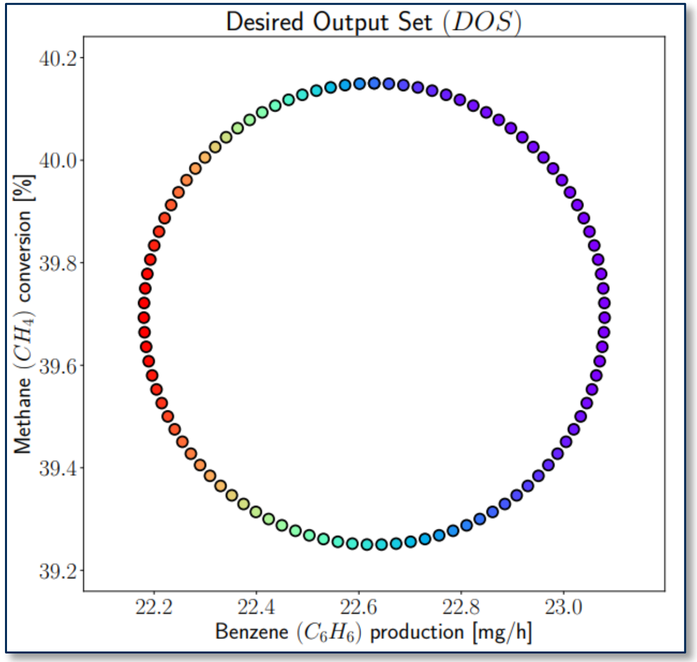
[2]

[1] Bradbury et al. (2018). http://github.com/google/jax.
[2] Kitchin J. R. (2023). https://pointbreezepubs.gumroad.com/l/pycse-ad.

Case Study – Implicit Mapping Results[1]

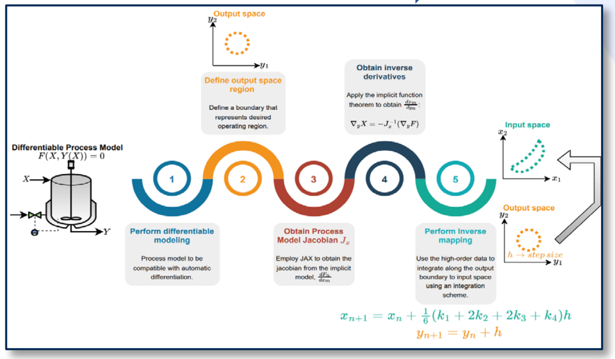


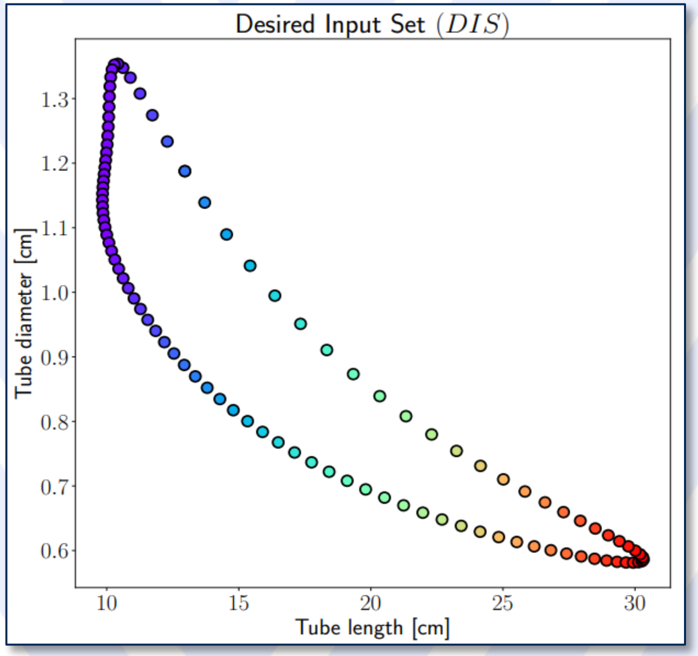
Test the proposed approach to find the inverse map for a given optimal operating region and compare it against the NLP-based approach (with enhanced features such as warm-start and AD)



Directly obtained inverse map

- Without NLP
- Analytical solution

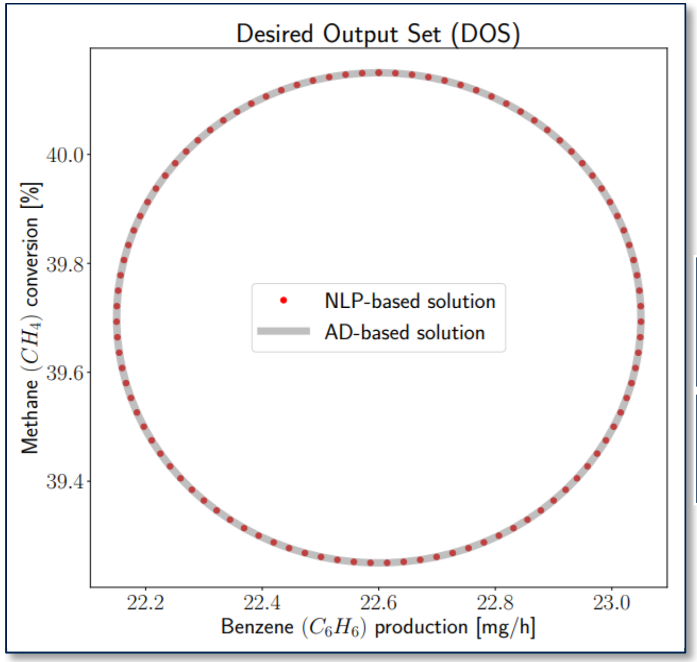




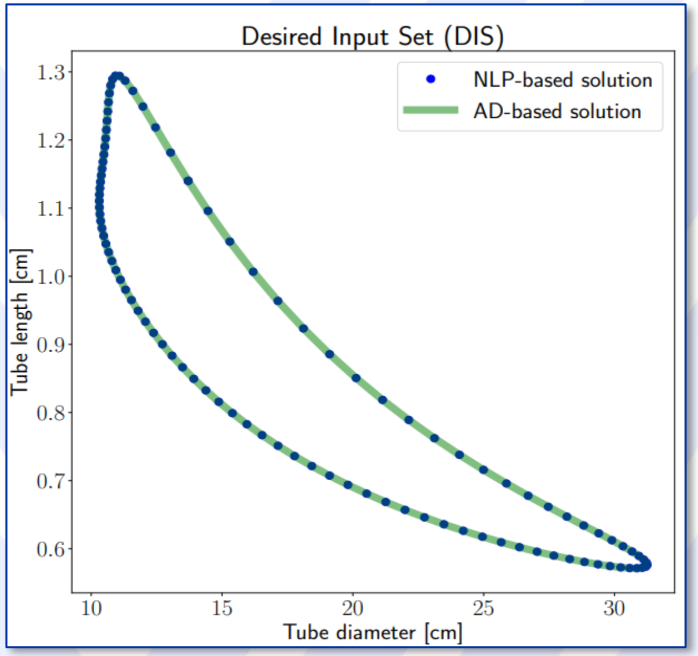
[1] Alves V. , Kitchin J. R. and Lima F. V. (2023). *AIChE Journal.*

Case Study – Implicit Mapping Results[1]

Comparison against the NLP-based approach: Accuracy and computational time



Input variable/ Relative error	Tube length [cm]	Tube diameter [cm]
$\Sigma \left\| \frac{NLP_{\text{value }j} - AD_{\text{value }j}}{NLP_{\text{value }j}} \right\| \cdot 100[\%]$	0.0186	0.0096



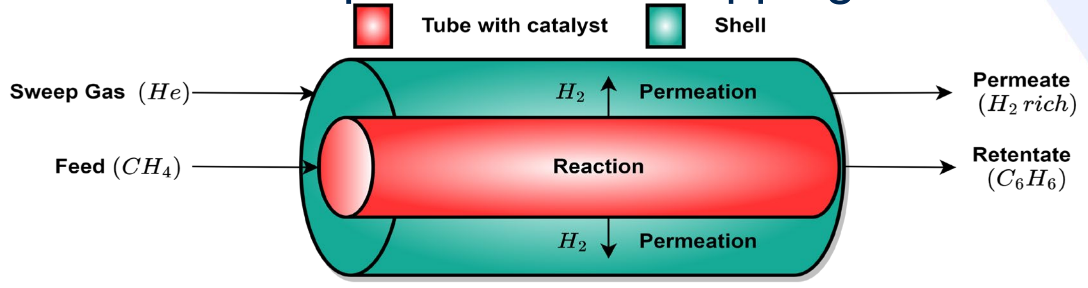
Solution approach	Time [min]	Times longer than proposed approach
Proposed approach	0.93	-
NLP ("cold-start" + finite differences)	17.84	19.18
NLP ("warm-start" + finite differences)	11.56	12.43
NLP ("warm-start" + AD)	1.97	2.11

The proposed approach outperforms NLP-based inverse mapping solutions even when using state-of-the-art implementations (e.g., AD for Jacobians/Hessians in the NLP and "warm-start")

[1] Alves V. , Kitchin J. R. and Lima F. V. (2023). *AIChE Journal*

Case Study – Implicit Mapping Results[1]

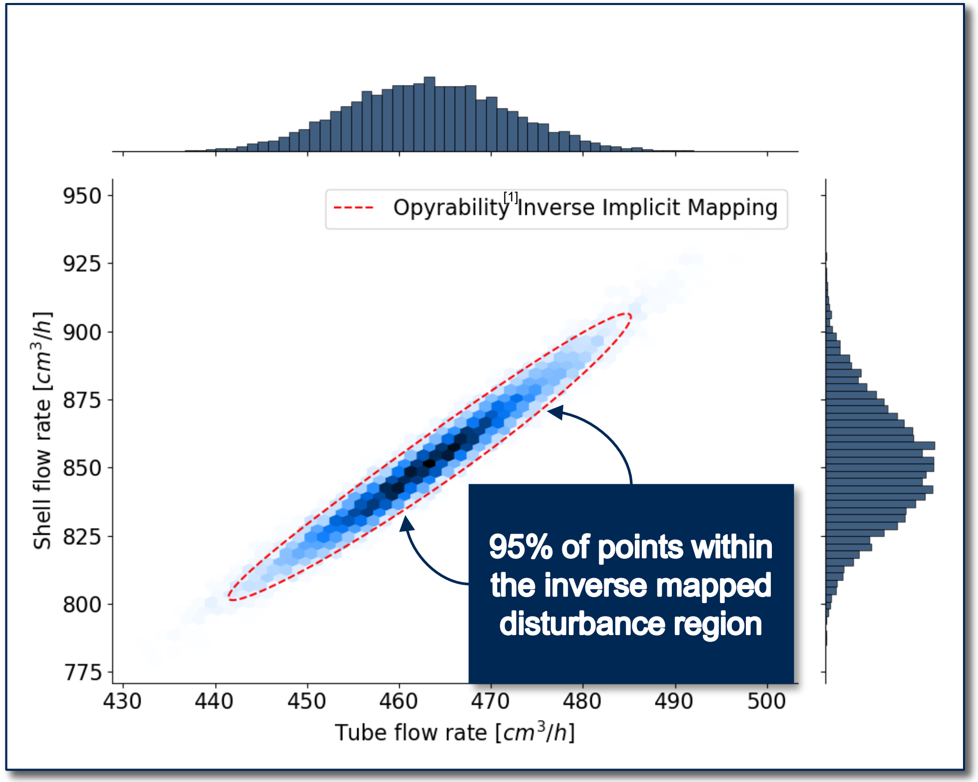
Implicit inverse mapping of disturbances from the output space to the input space



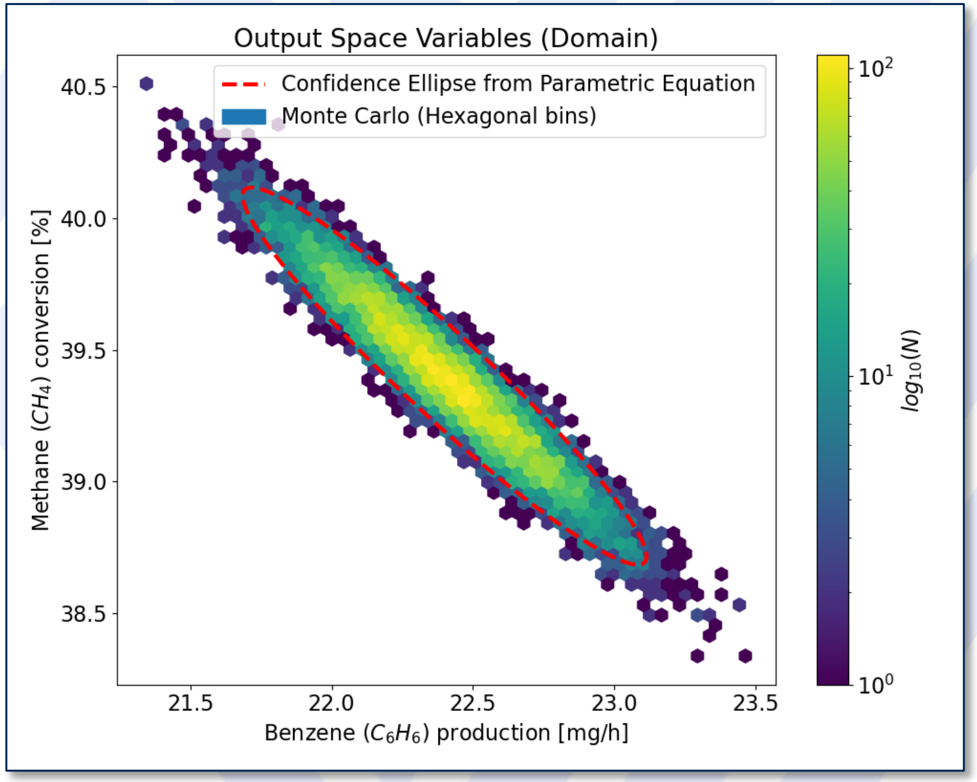
Problem set up:

1. Define a possibly correlated uncertainty confidence region
2. Map disturbances to input variables using the proposed approach
3. Compare against 10,000 Monte Carlo simulations (NLP)

- Input variables (Image)
  1. Tube flow rate [cm^3/h]
  2. Shell flow rate [cm^3/h]
- Outputs variables (Domain)
  1. Benzene production [mg/h]
  2. Methane conversion [%]



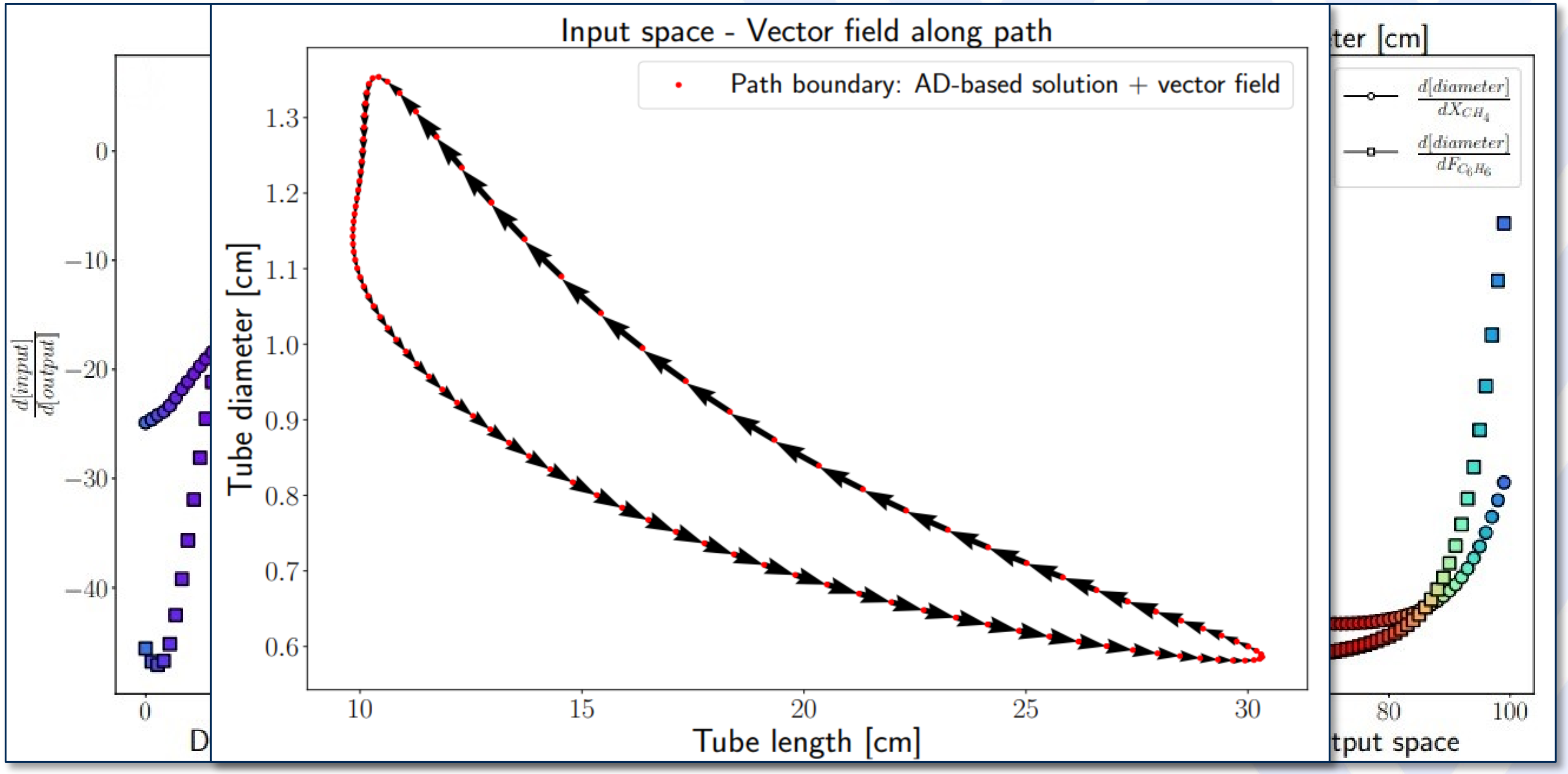
Implicit map of disturbance region



[1] Alves V., Dinh S., Kitchin J. R., Gazzaneo V., Carrasco J. C. and Lima F. V. (2023). Submitted to Journ. of Open Source Soft.

Case Study – Additional ("Free") Results[1]

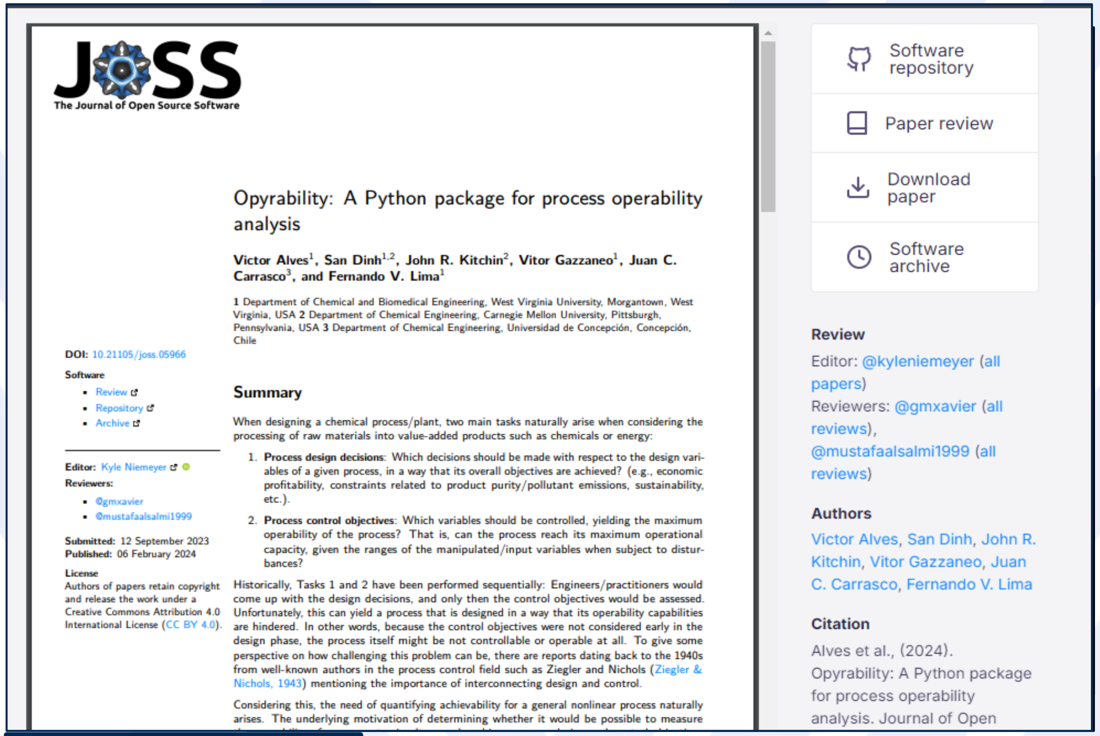
- Due to readily available high-order derivatives, phase-portraits and point-to-point derivative analysis (sensitivity) can be also performed in the inverse direction, not resorting to NLP-based solutions



[1] Alves V. , Kitchin J. R. and Lima F. V. (2023). *AIChE Journal.*

Conclusions and Future Work

- Initial release of the proposed tool
  - Computational aspects
    - NLP-based approach using different solvers
    - Support for automatic differentiation
    - Polytopic calculations for seamless OI quantification
  - Documentation available
    - Modules implemented and presented here
    - Case studies – examples from this presentation
- Future work
  - Implement multilayer framework for simultaneous solution of design and control problems[1]
  - Add support to Gaussian Process modeling within the Process Operability framework[2]
  - Implementation of dynamic Process Operability formulations
  - Interface with Pyomo[3]



JOSS
The Journal of Open Source Software

Opyrability: A Python package for process operability analysis

Victor Alves[1], San Dinh[1,2], John R. Kitchin[2], Vitor Gazzaneo[1], Juan C. Carrasco[3], and Fernando V. Lima[1]

1 Department of Chemical and Biomedical Engineering, West Virginia University, Morgantown, West Virginia, USA 2 Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA 3 Department of Chemical Engineering, Universidad de Concepción, Concepción, Chile

DOI: 10.21105/joss.05966

Software

- Review
- Repository
- Archive

Editor: Kyle Niemeyer

Reviewers:

- @gmxavier
- @mustafaalsalmi1999

Submitted: 12 September 2023
Published: 06 February 2024

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

When designing a chemical process/plant, two main tasks naturally arise when considering the processing of raw materials into value-added products such as chemicals or energy:

1. **Process design decisions**: Which decisions should be made with respect to the design variables of a given process, in a way that its overall objectives are achieved? (e.g., economic profitability, constraints related to product purity/pollutant emissions, sustainability, etc.).
2. **Process control objectives**: Which variables should be controlled, yielding the maximum operability of the process? That is, can the process reach its maximum operational capacity, given the ranges of the manipulated/input variables when subject to disturbances?

Historically, Tasks 1 and 2 have been performed sequentially: Engineers/practitioners would come up with the design decisions, and only then the control objectives would be assessed. Unfortunately, this can yield a process that is designed in a way that its operability capabilities are hindered. In other words, because the control objectives were not considered early in the design phase, the process itself might be not controllable or operable at all. To give some perspective on how challenging this problem can be, there are reports dating back to the 1940s from well-known authors in the process control field such as Ziegler and Nichols (Ziegler & Nichols, 1943) mentioning the importance of interconnecting design and control.

Considering this, the need of quantifying achievability for a general nonlinear process naturally arises. The underlying motivation of determining whether it would be possible to measure

Software repository
Paper review
Download paper
Software archive

Review
Editor: @kyleniemeyer (all papers)
Reviewers: @gmxavier (all reviews), @mustafaalsalmi1999 (all reviews)

Authors
Victor Alves, San Dinh, John R. Kitchin, Vitor Gazzaneo, Juan C. Carrasco, Fernando V. Lima

Citation
Alves et al., (2024). Opyrability: A Python package for process operability analysis. Journal of Open

OPYRABILITY

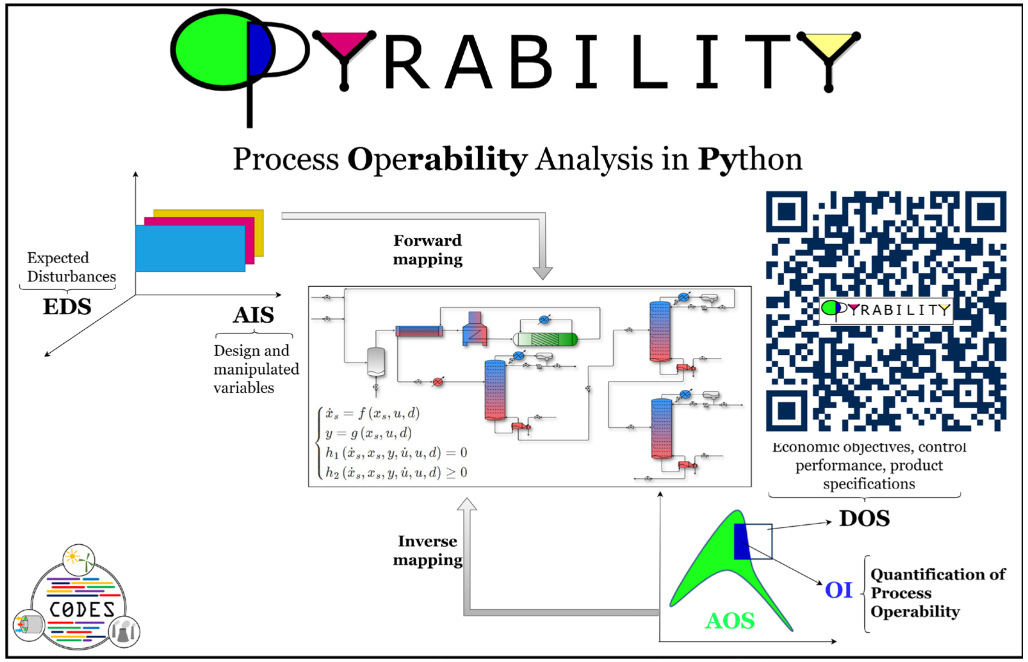
https://codes-group.github.io/opyrability

[1] Gazzaneo V. and Lima F. V. (2019). *Ind. Eng. Chem. Res.*
[2] Alves V. , Gazzaneo V. and Lima F. V. (2022). *Comput. Chem. Eng.*
[3] Bynum et al. (2021). *Pyomo – Optimization Modeling in Python*

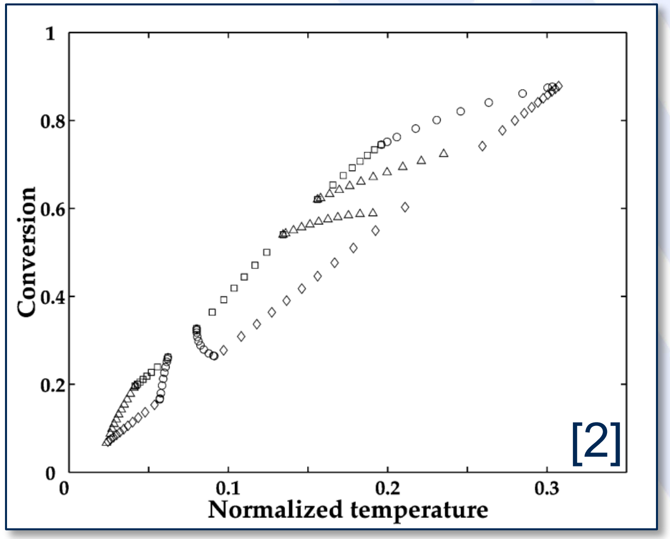
Conclusions and Future Work

- Implicit mapping shows potential to solve problems in PSE (but not limited to) with design and manufacturing nature
- Creative way of solving mapping problems when compared against typical approaches (e.g., enumeration and NLP)
- The intent is not to compete with previous approaches but rather to provide an alternative with complementary features

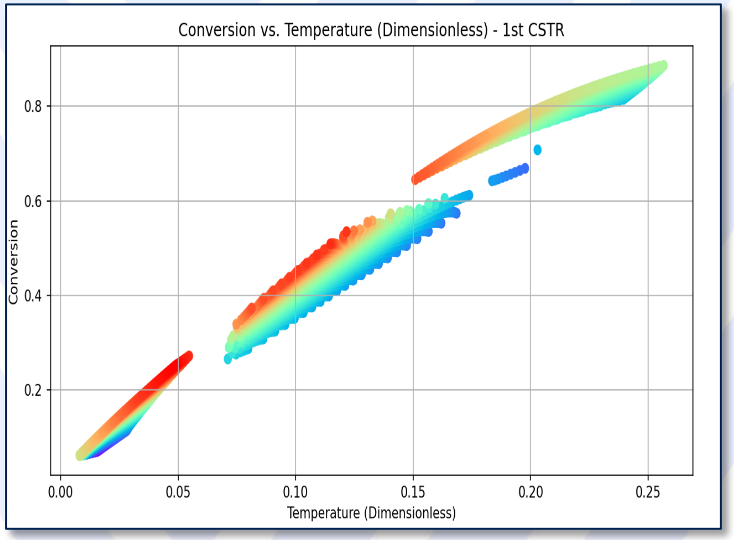
- Generalize it in terms of mapping direction and dimensionality (non-square systems)
  - Address high-dimensional implicit maps – implemented in Opyrability[1]
- Handle bifurcations in input/output spaces: Particular case in which the Jacobian is non-invertible
  - Implement arc-length continuation within the proposed approach – being implemented in Opyrability[1]
  - Detect multiple steady-states and input/output multiplicity using the proposed approach – Singularity/elementary catastrophe theory



Bifurcation in the output space of a CSTR



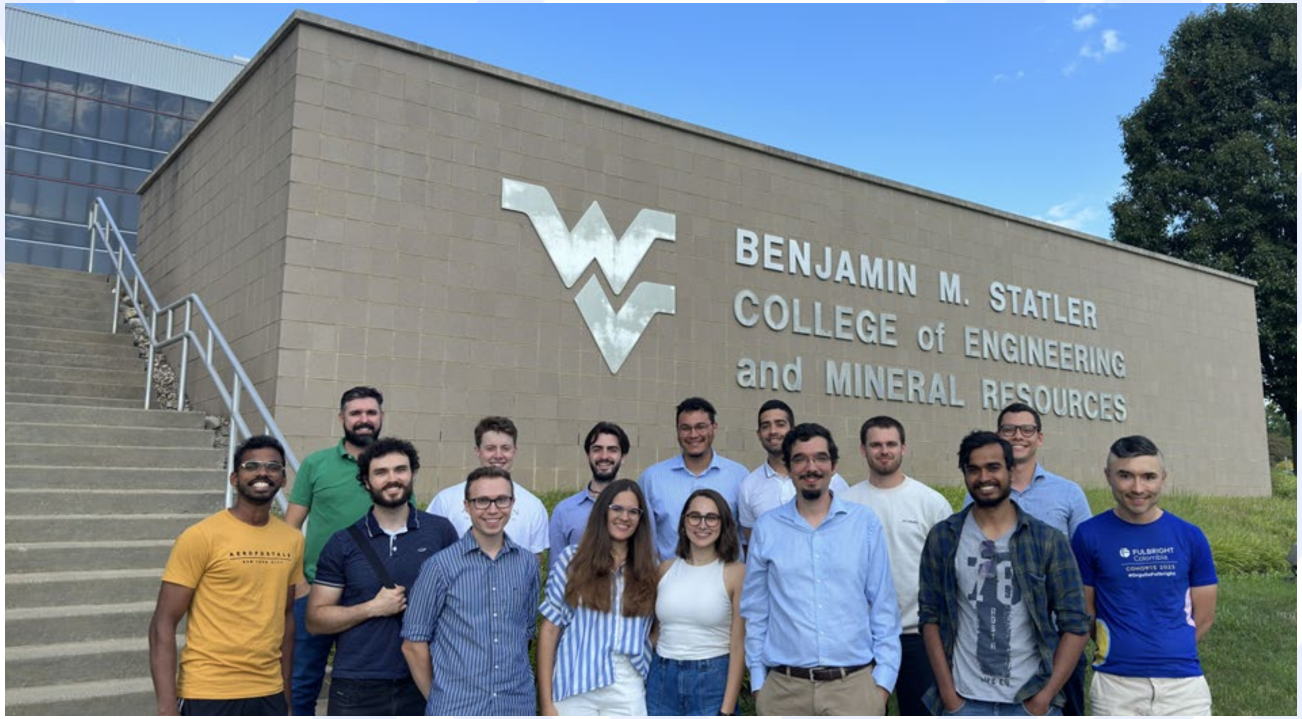
$$g = \frac{\partial g}{\partial x_2} = \frac{\partial^2 g}{\partial x_2^2} = \cdots = \frac{\partial^k g}{\partial x_2^k} = 0, \frac{\partial^{k+1} g}{\partial x_2^{k+1}} \neq 0$$



[1] Alves V., Dinh S., Kitchin J. R., Gazzaneo V., Carrasco J. C. and Lima F. V. (2024). Journ. of Open Source Soft.
[2] Subramanian S. and Georgakis C. (2001). *Chem. Eng. Sci.*

Acknowledgements

- WVU-UFPR (Brazil) project number 18.388.279-5 for the support
- CODES Research Group members
- Fernando V. Lima, John R. Kitchin, San Dinh, Vitor Gazzaneo, Juan Carrasco



Presenting author contact email:
Victor Alves
victor-alves.com
vc00004@mix.wvu.edu

victor-alves.com

CODES Research Group website:
https://fernandolima.faculty.wvu.edu/home
or by QR