Computational toolkits for model-based design and optimization

Damian T. Agi, Kyla D. Jones, Madelynn J. Watson, Hailey G. Lynch, Molly Dougher, Xinhe Chen, Montana N. Carlozo, Alexander W. Dowling*

Department of Chemical and Biomolecular Engineering
University of Notre Dame, Notre Dame, IN 46556

Abstract

We review recent advances in software platforms for model-based design (MBD) organized in five overarching themes — from (1) simulation to optimization, (2) commercial to open-source, (3) process-centric to multi-scale, (4) mechanistic to data-driven, and (5) deterministic to uncertain — illustrated with several recent examples in membrane system design. We posit MBD provides (chemical) engineers with principled frameworks to tackle global grand challenges such as sustainable energy, clean water, and equitable access to healthcare by integrating knowledge across disciplines. As such, we predict MBD software, which has historically focused on engineered systems, will evolve to interact with models for natural and social systems more holistically. Finally, we emphasize the importance of open-source software development, especially by users who become contributors.

Keywords: model-based optimization, dynamic optimization, uncertainty quantification, multi-scale design, data-driven models, open-source software

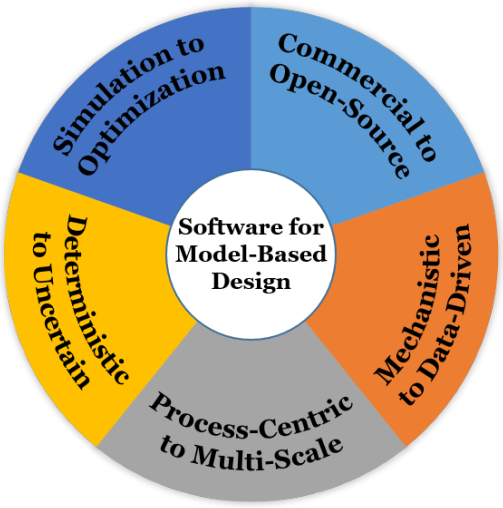
Highlights

- Systematically review 82 model-based design (MBD) toolkits
- Dynamic, multiscale, and interdisciplinary grand challenges drive software

*Corresponding author
Email address: adowling@nd.edu (Alexander W. Dowling)

trends

- Organize MBD software advances in five themes
- Contemporary membrane examples illustrate the confluence of MBD themes



Graphical Abstract

Model-Based Design for Chemical Engineering and Beyond

Chemical engineers are uniquely positioned to create innovative solutions for the United Nation's Sustainable Development Goals, including equitable access to clean water, health services, affordable sustainable energy, and circular economies. Yet, these and related grand challenges are "wicked problems" with complex interdependencies across natural, social, and engineered systems [1]. As such, (chemical) engineering design has evolved to contemplate coupled decisions across molecular to infrastructure scales, often operating dynamically away from steady-state under uncertainty [2]. We argue that model-based approaches, grounded in fundamental engineering science, provide the necessary

abstractions and predictive capabilities to tackle these wicked problems. This paper reviews software platforms for model-based design (MBD), organized in the five overarching themes shown in Figure 1. These five themes align with the contemporary and emerging research directions in process systems engineering [3]. Finally, several examples from model-based membrane design illustrate the confluence of these five themes.

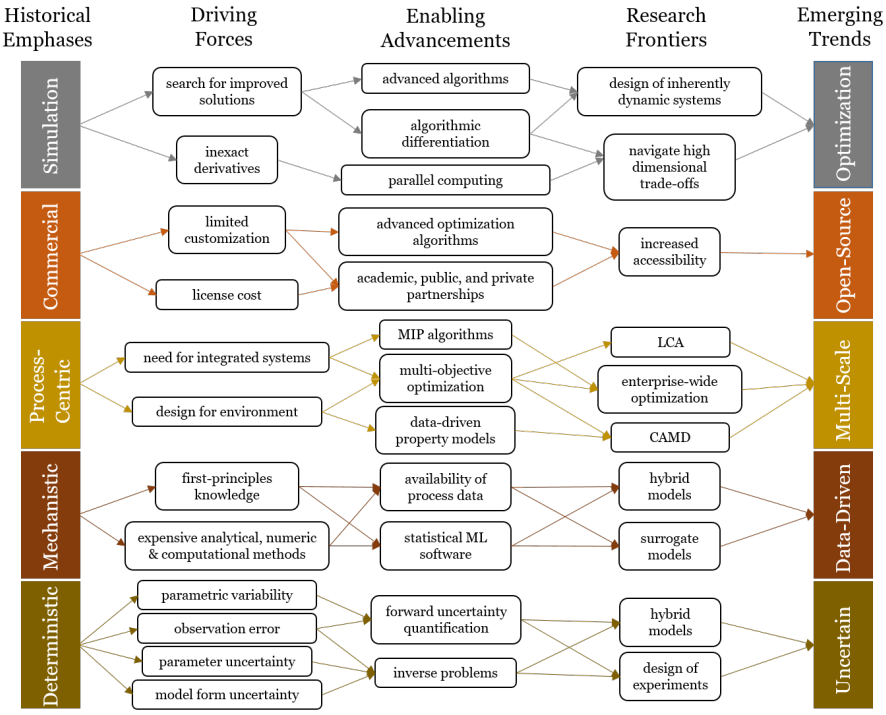


Figure 1: Five overarching themes summarizing the evolution of model-based design. Acronyms: ML—machine learning, MIP—mixed integer programming, LCA—life cycle analysis, CAMD—computer-aided molecular design.

Theme 1: Simulation to Optimization

Chemical process synthesis has evolved from empirical correlations and heuristics to MBD, including decomposition strategies and optimization-based methods [4, 5]. Decomposition-based designs hierarchically organize process design

tasks in clusters of common fundamental concepts, whereas optimization-based designs aggregate possible designs into a superstructure and solve computational optimization problems to find the best process configuration [5]. The implementation of optimization-based design relies on the simulation approach.

Process simulation for MBD has two prevailing paradigms: sequential modular (SM) and equation-oriented (EO). SM exploits the modular nature of chemical processes to successively converge individual unit models, in which the output of one unit becomes the input of the next. SM is used in Aspen Hysys® [6], Aspen Plus® [7], DWSIM [8], and other popular software (see Table 1), but may require many iterations to converge flowsheets (especially with complex recycles). Moreover, the sequential convergence of unit models further exacerbates numerical noise in derivative estimates via finite differences, which can make steady-state optimization with gradient-based methods unreliable. As such, optimization with SM simulators is often restricted to a handful of variables, which can make multi-stage design, uncertainty quantification (UQ), and dynamic optimization intractable.

EO environments such as Aspen Custom Modeler® (ACM) and gPROMS® efficiently converge large-scale systems of nonlinear equations using a numerical method (e.g., Newton-Raphson) with exact derivatives from automatic differentiation (AD). As such, optimization with EO models using state-of-the-art algorithms and software often requires similar computational effort as SM simulation [9]. Initialization, scaling, and model diagnostics remain the most significant challenges for EO methods [10, 11]. A common strategy is to initialize with SM simulations to provide reliable starting points for optimization with EO, especially for commercial tools such as AspenTech products. Commercial EO environments often include only a handful of solvers and do not provide complete interfaces to call state-of-the-art optimization algorithms to protect proprietary models and databases.

Algebraic modeling languages (AMLs) such as General Algebraic Modeling System (GAMS) [12], A Mathematical Programming Language (AMPL®) [13], AIMMS [14], Pyomo [15], and JuMP [16] offer the most flexibility for

users to define the EO model equations directly. These AMLs leverage AD and standardized interfaces to provide exact derivatives to the optimization solvers. Moreover, the flexibility of interacting with models in Pyomo and JuMP has catalyzed specialized solver development, thus connecting algorithmic advances with large-scale industrially relevant applications. Extensions such as Pyomo.DAE [17] transcribe (partial) differential-algebraic equations into algebraic models to facilitate numeric solutions. Complementary to AMLs, dynamic modeling toolkits including CasADi [18], OpenModelica [19], and APMonitor [20] specialize in simulation and optimization using numeric integration methods (e.g., multiple shooting).

Most AMLs and some dynamic modeling toolkits are generalized environments that lack modeling libraries to facilitate MBD, which requires extensive custom model development by the user. To address this gap, the Institute for the Design of Advanced Energy Systems Process Systems Engineering (IDAES®-PSE) platform provides an object-oriented modeling library built on Pyomo [21]. Recent application-specific extensions of IDAES-PSE include the Design Integration and Synthesis Platform to Advance Tightly Coupled Hybrid Energy Systems (DISPATCHES), Water treatment Techno-economic Assessment Platform (WaterTAP), the Produced Water Application for Beneficial Reuse, Environmental Impact and Treatment Optimization (PARETO), and the Process Optimization and Modeling for Minerals Sustainability (PrOMMiS) project. Compared to most legacy MBD tools focused on simulation, these projects emphasize computational optimization.

Name	Interface(s)/Language(s)	License	Equation Oriented	Sequential Modular	Dynamics	Process Design	Supply Chain	Scheduling	Planning	Parameter Estimation	UQ	LCA	Description
Advanced Simulation Library [22]	OpenCL, C++	AGPL	✓		✓	✓							Platform for multiphysics simulation, PDEs
AIMMS [14]	IDE, custom scripting	Proprietary	✓		✓	✓	✓	✓	✓				AML popular for logistics & data science
AMPL® [13]	IDE, custom scripting	Proprietary	✓		✓	✓	✓	✓	✓				AML for large-scale optimization
Aspen Custom Modeler® [23]	Fortran-like	Proprietary	✓		✓	✓							Equation-oriented modeling platform compatible with Aspen Plus® and Aspen Hysys®
Aspen Hysys® [6]	GUI	Proprietary		✓	✓*	✓		✓	✓				Chemical process simulator popular for petrochemical process design and optimization
Aspen Plus® [7]	GUI	Proprietary		✓	✓*	✓							General purpose chemical process simulator
Aspen Plus Dynamics [24]	GUI	Proprietary	✓		✓	✓							Dynamic process simulator integrated with Aspen Plus
BioSTEAM [25]	Python	UIUC		✓		✓						✓	Simulation, techno-economic analysis and LCA of biorefineries
CasADi [18]	C++, Python, MATLAB®, Octave	LGPL v3.0			✓	✓				✓			Dynamic modeling, algorithmic differentiation, optimization
CHEMCAD	GUI	Proprietary		✓	✓	✓				✓			General purpose chemical process simulator
COCO [26]	GUI	COCO v3.6		✓		✓							General purpose chemical process simulator
DWSIM [8]	GUI	GPL v3.0		✓	✓	✓							General purpose chemical process simulator
Dyssol [27]	C++	BSD-3		✓	✓	✓							Simulator designed for solids processes
GaBi [28]	GUI	Proprietary		✓								✓	LCA tool for emissions and energy use
GAMS [12]	IDE, custom scripting	Proprietary	✓		✓	✓	✓	✓	✓				AML for large-scale optimization
gPROMS® [29]	GUI	Proprietary	✓		✓	✓				✓			EO process design and optimization environment
GREET® [30]	GUI	CC NC v4		✓								✓	LCA tool for transportation emissions and energy use calculations
IDAES®-PSE [21]	Python	BSD-3	✓		✓	✓				✓	✓		Pyomo-based object-oriented modeling for multi-scale optimization
JuMP [16]	Julia	MPL v2.0	✓		✓	✓	✓	✓	✓				AML for large-scale optimization in Julia
openLCA [31]	GUI	MPL v2.0		✓								✓	Modular framework for sustainability assessment and life cycle modelling
OpenModelica [19]	Modelica, Python, Julia, MATLAB®	OSMC-PL	✓		✓	✓				✓			Dynamic modeling and optimization
PharmaPy [32]	Python	BSD-3		✓	✓	✓							MBD of pharmaceutical processes
PRO/II™ [33]	GUI	Proprietary		✓		✓							General purpose chemical process simulator
Pyomo [15]	Python	BSD-3	✓		✓	✓	✓	✓	✓	✓*	✓*		AML for large-scale optimization in Python
SimaPro [34]	GUI	Proprietary		✓								✓	Sustainability assessments & product design
Umberto® [35]	GUI	Proprietary		✓								✓	Software for LCA and cost calculations
ProCAFD® [36]	GUI	Proprietary	✓	✓		✓						✓*	Systematically generates, analyzes, and evaluates chemical process designs
ProCAPE® [36]	GUI	Proprietary		✓						✓	✓	✓*	Molecular property estimation
ProREFD® [37]	GUI	Proprietary		✓		✓				✓	✓	✓*	Refrigerant selection, design, and verification

Table 1: Toolkits for model-based design and optimization. *Enabled through extensions/packages. Acronyms: IDE—integrated development environment, GUI—graphical user interface, UQ—uncertainty quantification, LCA—life cycle analysis, AML—algebraic modeling language.

Theme 2: Commercial to Open-Source

Over the past two decades, many open-source MBD software have emerged. Commercial platforms such as the AspenTech suite, gPROMS, PRO/II™, and CHEMCAD (Chemstations™) offer a wide range of tools for process development such as simulation, optimization, and cost estimation. These commercial tools, as well as commercial AMLs such as GAMS, AMPL, and AIMMS, offer support and contract model development, which is essential for many organizations. Academic and government researchers, motivated by cost and ease of customization, have led the creation of open-source tools, including DWSIM, CasADi, Pyomo, and JuMP. Jusevičius et al. [38] compared the performance of three commercial AMLs (AMPL, AIMMS, GAMS) and two open-source AMLs (JuMP, Pyomo). Direct comparison of commercial process simulators and custom models is often challenging, as commercial tools include proprietary databases for physical property models, which are difficult to reproduce in custom environments. The open-source license impacts the adoption of tools. Licenses such as MIT or BSD are the most flexible for commercialization, whereas "copyleft" licenses such as GPL require all derivatives to be released under the same open-source license, thus inhibiting proprietary extensions.

Theme 3: Process-Centric to Multi-Scale

Over the past two decades, MBD has shifted from process-centric analysis to a holistic perspective spanning molecular to global infrastructure scales. At larger scales, enterprise-wide optimization (EWO) optimizes manufacturing and distribution facilities, supply chains, R&D portfolios, and beyond to maximize profits, responsiveness to customers, and asset utilization while minimizing costs, inventory levels, and ecological footprints [39]. EWO problems are formulated as mixed-integer linear programs, mixed-integer nonlinear programs (MINLP), mixed-integer dynamic optimization problems, or logic-based models such as general disjunctive programming problems and constraint programming problems [39, 40]. Oliveira et al. [41] review common software for simulation and

optimization for supply chain management. While some commercial software exists for specialized planning problems, EWO models are often implemented in AMLs due to their flexibility and leverage either off-the-shelf solvers (see review by Grossmann [39]) or logic-based algorithms such as Pyomo.GDP [15] or LOGMIP and EMP in GAMS. Pyomo and JuMP make it easier to develop decomposition algorithms, e.g., Coramin [42], Mindtpy [43] and Parapint [44], that exploit problem structure to accelerate the solution of large-scale (mixed integer) (nonlinear) optimization problems. Similarly, Octeract Neural [45], which uses generative AI, demonstrates the potential for a new generation of mathematical optimization algorithms for extremely large multiscale problems.

Life cycle analysis (LCA) has become a crucial step in EWO and design for environment frameworks that consider each step of the product life cycle: design/development, raw material acquisition, manufacturing, distribution, use/maintenance/reuse, and end-of-life activities [46]. Choosing the software tool for LCA is critical, as a wide range of commercial software is available, varying in functionality, database availability, user interface, data quality management, and modeling principles, which can result in different LCA results [47]. Popular LCA software tools include SimaPro, GaBi, Umberto®, openLCA, and GREET [47]. Mahmud et al. [46] extensively review LCA software tools and databases.

Complementing EWO and LCA, computer-aided molecular design (CAMD) simultaneously designs new molecules and optimizes their manufacturing process or their usefulness in other processes [48]. Often trained via machine learning (ML) techniques, quantitative structure-property relationships bridge models across molecular, material, device, and process scales. CAMD problems are usually formulated as MINLP and solved with off-the-shelf algorithms, specialized decomposition strategies, or heuristic search [49]. Popular CAMD packages include ProCAFD® [36] which fine-tunes chemical process designs; ProCAPD® [50] which optimizes product designs; and ProCAPE® [36] and ProREFD® [37] which are used to estimate molecular properties [51]. These often interface with LCSoft and ECON to create sustainable process and molecular designs that

satisfy additional economic, environmental, and safety constraints [52].

Theme 4: Mechanistic to Data-Driven

Mechanistic models (a.k.a., first-principles, phenomenological, white- or glass-box models) have been central to process monitoring, design, control, and optimization. Mechanistic models are constructed using knowledge of the fundamental science (e.g., reaction kinetics, transport phenomena, boundary conditions, and thermodynamics) to facilitate safer extrapolation and technological innovation. Once constructed, mechanistic models are solved with analytical and numerical methods. In process industries, software such as MATLAB and Simulink® [53], COMSOL Multiphysics® [54], gPROMS [29], AspenTech [55], and Simcenter STAR-CCM+ [56] are central to these tasks [57].

In a recent review, Sansana et al. [58] note four primary shortcomings of mechanistic modeling. First, mechanistic modeling is fragmented by a lack of generalizability, as this paradigm depends on bespoke implementations that are difficult to reuse. Second, practitioners must balance a trade-off between developing high-fidelity models to improve accuracy and their increased computational expense. Third, mechanistic models are often time-consuming to formulate and expensive to maintain. Finally, mechanistic modeling does not leverage the ever-increasing availability of process data in chemical engineering. Toward these ends, surrogate modeling has become a popular alternative to mechanistic modeling.

A surrogate model (a.k.a., data-driven, black-box, statistical, emulator, or meta-model) is a computationally inexpensive substitute for a mechanistic model that relies solely on paired input-output data from the system to make predictions. Types of surrogate models include (polynomial) response surfaces, support vector machines, kriging, radial basis functions, artificial neural networks (ANNs), multivariate adaptive regression splines, Fourier, and random forest models. Popular surrogate modeling tools include Automatic Learning of Algebraic Models for Optimization (ALAMO) [59], Python-based Surrogate

Modelling Objects (PySMO), Reaction Identification and Parameter Estimation (RIPE), and HELMholtz Energy Thermodynamics, which are all available in the IDAES-PSE ecosystem [21], as well as TensorFlow [60], scikit-learn [61], and SciML Surrogates.jl [62].

The data-driven nature of surrogate models leads to three challenges discussed by Shulkind et al. [63]. First, surrogate models are limited in physical interpretability due to the agnosticism of their underlying functional form. Second, surrogate models rely entirely on data availability for sufficient fidelity. Third, the lack of extreme scenario data may lead to poor out-of-sample prediction performance. Thus, the ideal modeling paradigm must encompass prior knowledge and process data.

Hybrid (a.k.a., grey-box) models overcome many limitations of both mechanistic and surrogate models. Hybrid models combine a priori knowledge of the system with data-driven insights [64]. Available software packages for hybrid modeling include Novasign Hybrid Modeling Toolbox [65], Aspen Hybrid Models™ [55], HybridML [66], and PharmaPy [32]. In a recent review, Bradley et al. [67] note several use cases for hybrid models, including emulation, physics-informed ML, estimation, correction, and calibration. In the emulation use case, surrogate models replace computationally expensive high fidelity computer models. Similarly, physics-informed ML leverages a surrogate model to replace a mechanistic model while enforcing physical constraints in its training. Unlike emulation and physics-informed ML use cases, correction, estimation, and calibration cases leverage data-driven models to augment a mechanistic model. Regarding estimation, a data-driven model aims to learn a specific phenomenological relationship embedded in a mechanistic model. In contrast, correction use cases deploy data-driven models to learn the bias between a mechanistic model and data. Finally, calibration is a generalization of correction cases in which the user performs joint inference of the mechanistic model parameters and the bias.

Surrogate-assisted optimization uses data-driven or hybrid models as emulators for computationally expensive functions to reduce the solution time [68].

For example, algebraic surrogate models trained with ALAMO are well-suited for global or mixed integer optimization. More recently, the open-source OMLT [69] package facilitates embedding pre-trained ML models (ANNs and gradient-boosted trees) with the Pyomo environment as constraints. A key challenge with surrogate-assisted optimization is managing model error, especially near the optimal solution. The trust-region algorithm helps maintain the surrogate model's accuracy by controlling the trust region's size [70]. Recently, Chen et al. [71] used the trust-region algorithm with an algebraic surrogate model to successfully emulate and optimize the fluidized catalytic cracker in a large-scale refinery.

Theme 5: Deterministic to Uncertain

Quantifying the uncertainty of model predictions is critical for safe engineering practices and risk management in technology scale-up and deployment. There are several sources of uncertainty, of which we review four relevant to the discussion in this section [72]:

1. Parameter uncertainty – This uncertainty arises from the fact that the parameter values of mathematical models are not known a priori and must be estimated from data.
2. Model inadequacy – Also known as systematic bias or model discrepancy, model inadequacy arises when the mean value of the measured process does not match the mathematical model's output given the inputs' actual values.
3. Parametric variability – The predicted process value acquires an extra uncertainty attributed to the variation of the model inputs.
4. Observation error – Also known as measurement uncertainty, observation error arises when the observed (measured) value of the process does not match the model output.

UQ methods consider these uncertainties in two types of problems: forward and inverse. Forward problems aim to quantify the influence of parametric

variability on the outputs. That is, forward methods provide an estimate to the question: "How does uncertainty in the model inputs affect the model outputs?". Conversely, inverse UQ problems aim to infer cause from effect. That is, given experimental data subject to observation error, inverse UQ problems aim to infer the model parameters' optimal values and estimate their uncertainty. When the model is inadequate, inverse UQ estimates the discrepancy between the model and the data. This process is also known as "bias correction" or "discrepancy modeling," closely related to correction and calibration hybrid models (see *Theme 4: Mechanistic to Data-Driven*). Table 2 shows popular general-purpose UQ tools.

Often, models with quantified uncertainty are used for decision-making via optimization. While many of these tools in Table 2 support optimization under uncertainty (OUU), we highlight mpi-sppy [73] and PyROS [74], which provide stochastic programming and robust optimization capabilities, respectively, in Pyomo.

Design of experiments (DoE) methods determine the best experimental campaigns to learn information, minimize uncertainty, or discriminate between candidate models [90, 91, 92]. DoE methodologies include classical (e.g., full-factorial, fractional-factorial, response-surface, mixture), randomized (e.g., space-filling such as Latin hypercube, Sobol sequence), and optimal designs [93]. These methods quantify the interactions between input and output variables (i.e., factors and responses) in experimental designs through predictive analysis. In addition, model-based design of experiments (MBDoE) computes the next best set of experiments for model selection or parameter estimation tasks by directly exploiting (mechanistic) model predictions. Often, MBDoE is deployed sequentially, where the model is re-calibrated as new information becomes available to determine the next batch of optimal experiments. Thus, there is a strong connection between MBDoE and inverse UQ.

Table 2 indicates UQ software with DoE capabilities, whereas Table 3 outlines DoE software used frequently in chemical engineering. gPROMS is the most popular commercial software for MBDoE, while Table 3 highlights many

Name	Language(s)	License	Reliability Analysis	Sensitivity Analysis	Inverse UQ	Forward UQ	Surrogate Modeling	DoE	OUU
Dakota [75]	C++	LGPL	✓	✓	✓	✓	✓		✓
EasyVVUQ [76, 77]	Python3	LGPL v3.0		✓			✓		
FERUM [78]	MATLAB®	GPL v3.0	✓	✓					✓
FOQUS* [79]	GUI	BSD3-like		✓		✓	✓	✓	✓
MUQ [80]	C++, Python	GPL v2.0			✓	✓	✓		✓
OpenCOSSAN [81]	MATLAB®	LGPL	✓	✓		✓	✓		✓
OpenTURNS [82]	C++, Python	LGPL	✓	✓	✓	✓	✓		✓
PSUADE [83]	Shell Scripting	LGPL		✓	✓	✓		✓	✓
SUMO Toolbox [84]	MATLAB®	AGPL v3.0					✓	✓	✓
SmartUQ® [85]	GUI	Proprietary		✓	✓	✓	✓	✓	✓
UQLab [86]	MATLAB®	BSD-3	✓	✓	✓	✓	✓		✓
UQPy [87]	Python	MIT			✓	✓		✓	✓
UQTk [88, 89]	C++, Python	BSD-3		✓	✓	✓	✓		✓

Table 2: General purpose UQ software and their capabilities for reliability and sensitivity analysis, inverse and forward UQ, surrogate modeling, design of experiments (DoE), and optimization under uncertainty (OUU). *FOQUS provides an interface to UQ capabilities in PSUADE.

new open-source options. Notably, Pyomo.DoE [94] is a Python package for MBDoE that exploits EO Pyomo models for gradient-based optimization of Fisher information-based objectives. Pydex also leverages Pyomo models but discretizes the MBDoE decision space to exploit advances in convex optimization [95].

Name	Language	License	Factorial	Response Surface	Mixture	Space filling	Optimal	MBDoE
Design-Expert® [96]	GUI	Proprietary	✓	✓	✓	✓	✓	
dexpy [97]	Python	Apache	✓	✓	✓		✓	
doepy [98]	Python	MIT	✓			✓		
GPdoemd [99]	Python	MIT						✓
gPROMS® [29]	GUI	Proprietary						✓
JMP® [100]	GUI	Proprietary	✓	✓	✓	✓	✓	
Minitab® [101]	GUI	Proprietary	✓	✓	✓		✓	
MODDE® [102]	GUI	Proprietary	✓	✓	✓		✓	
NCSS [103]	GUI	Proprietary	✓	✓	✓		✓	
pydex [95]	Python	MIT						✓
pyDOE [104]	Python	BSD-3	✓	✓		✓		
Pyomo.DoE [94]	Python	BSD-3						✓
SAS® [105]	GUI	Proprietary	✓	✓	✓	✓	✓	
Statgraphics® [106]	GUI	Proprietary	✓	✓	✓		✓	
Statistics and Machine Learning toolbox [107]	MATLAB®	Proprietary	✓	✓		✓	✓	
Synthace [108]	GUI	Proprietary	✓	✓	✓	✓	✓	

Table 3: Popular DoE toolkits in chemical engineering. Readers are also encouraged to explore DoE software packages in R [109], which are more commonly used in the applied statistics community.

Five Themes Applied to Model-Based Membrane Design

These five themes for MBD are often applied in conjunction, as illustrated by the following contemporary examples of process intensification with membranes.

Clean Water. Di Martino et al. [110] implemented ANN surrogate models to optimize reverse osmosis systems for water desalination (*themes 1, 3*). Recently, Razman et al. [111] overviewed LCA methods for membrane-based systems for wastewater treatment and common challenges, such as the lack of consistency (*theme 4*).

Sustainable Energy. Recently, Cherif et al. [112] used MBD for the multi-objective optimization of a palladium-based membrane steam methane reformer for hydrogen production (*themes 1, 4*). Sarma and Ganguly [113] implemented a model-based optimization of proton-exchange membrane fuel cell-battery-hybrid energy systems to minimize fuel consumption subject to dynamic power balances and other operating constraints (*themes 1,4*). Wamble et al. [114] used superstructure optimization implemented in Pyomo to design diafiltration cascades for lithium-ion battery recycling and quantify the systems-scale benefits of membrane material improvements (*themes 1, 2, 4*). Similarly, Zach et al. [115] proposed a general-purpose, model-based tool based on GAMS for optimizing membrane separators for post-combustion carbon capture (*themes 1, 3, 4*).

Medical Sciences. Khulu et al. [116] optimized the membrane-assisted extraction of therapeutic pharmaceuticals from surface water using a central composite DoE technique (*themes 1, 3, 5*). Finally, Bayazidi et al. [117] used computational fluid dynamics and DoE for MBD, optimization, and parameter identification of a micropump embedded with a vibrating membrane for applications in drug delivery (*themes 3, 5*).

Future Outlook

We predict chemical engineers will increasingly utilize MBD to guide interdisciplinary teams focused on convergent research for global grand challenges.

Modeling abstractions in process systems engineering provide coherent frameworks to integrate knowledge across disciplines. As such, we expect MBD software to expand, analogous to the growth of EWO paradigms over the last decade, to emphasize interactions with natural and social system models. Likewise, we anticipate a continued proliferation of open-source software but note the challenges with actively maintaining research software, especially for small academic groups. As such, we see the continued importance of investments in large software projects, such as Pyomo, JuMP, CasADi, and IDAES-PSE, that provide extensive benefits to the broad community. We encourage more contributions to these open-source packages and training of researchers in software carpentry skills.

Acknowledgements

We acknowledge support from the following grants, projects, and fellowships: NSF Cooperative Agreement No. EEC-1647722 (D.T.A., M.J.W., A.W.D.); NSF Award CBET-1941596 (K.D.J., A.W.D.); the Graduate Assistance in Areas of National Need fellowship from the Department of Education via grant number P200A210048 (M.J.W., M.C.) with extra support from the University of Notre Dame via the College of Engineering and the Graduate School (M.J.W., M.C., K.D.J.); the Lucy Institute for Data & Society Lucy Graduate Scholar's Fellowship (K.D.J.); the Arthur J. Schmitt Presidential Leadership Fellowship (M.J.W., M.D.); the Remick Graduate Fellowship in Engineering (H.G.L.); the Advance Tightly Coupled Hybrid Energy Systems (DISPATCHES) project through the Grid Modernization Lab Consortium with funding from the U.S. Department of Energy's (DOE) Office of Fossil Energy and Carbon Management, Office of Nuclear Energy, and Hydrogen and Fuel Cell Technology Office (X.C., A.W.D).

This paper was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or im-

plied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Conflict of interest statement

The authors declare no conflict of interest

References

[1] J. Lönngren, K. Van Poeck, Wicked problems: A mapping review of the literature, International Journal of Sustainable Development & World Ecology 28 (2021) 481–502.

[2] E. A. Eugene, W. A. Phillip, A. W. Dowling, Data science-enabled molecular-to-systems engineering for sustainable water treatment, Current Opinion in Chemical Engineering 26 (2019) 122–130. doi:https://doi.org/10.1016/j.coche.2019.10.002.

[3] E. N. Pistikopoulos, A. Barbosa-Povoa, J. H. Lee, R. Misener, A. Mitsos, G. V. Reklaitis, V. Venkatasubramanian, F. You, R. Gani, Process systems engineering – the generation next?, Computers & Chemical Engineering 147 (2021) 107252. URL: https://www.sciencedirect.com/science/article/pii/S0098135421000302. doi:https://doi.org/10.1016/j.compchemeng.2021.107252.

[4] A. W. Dowling, L. T. Biegler, A framework for efficient large scale equation-oriented flowsheet optimization, Computers & Chemical Engineering 72 (2015) 3–20. URL: https://www.sciencedirect.com/science/article/pii/S009813541400163X. doi:https://doi.org/10.1016/j.compchemeng.2014.05.013.

[5] Q. Chen, I. Grossmann, Recent developments and challenges in optimization-based process synthesis, Annual Review of Chemical and Biomolecular Engineering 8 (2017) 249–283. doi:https://doi.org/10.1146/annurev-chembioeng-080615-033546.

[6] Aspen HYSYS®, 2023. URL: www.aspentech.com/en/products/engineering/aspen-hysys, accessed on July 14, 2023.

[7] Aspen Plus®, 2023. URL: www.aspentech.com/en/products/engineering/aspen-plus, accessed on July 14, 2023.

[8] Daniel Medeiros, DWSIM, 2023. URL: http://dwsim.inforside.com.br, accessed on July 14, 2023.

[9] L. T. Biegler, New directions for nonlinear process optimization, Current Opinion in Chemical Engineering 21 (2018) 32–40. doi:https://doi.org/10.1016/j.coche.2018.02.008.

[10] A. W. Dowling, L. T. Biegler, Degeneracy hunter: An algorithm for determining irreducible sets of degenerate constraints in mathematical programs, in: Computer Aided Chemical Engineering, volume 37, Elsevier, 2015, pp. 809–814.

[11] R. B. Parker, B. L. Nicholson, J. D. Siirola, L. T. Biegler, Applications of the dulmage-mendelsohn decomposition for debugging nonlinear optimization problems, Computers & Chemical Engineering (2023) 108383.

[12] GAMS Development Corporation, General algebraic modeling system (GAMS), 2023. URL: www.gams.com/, accessed on July 14, 2023.

[13] R. Fourer, D. M. Gay, B. W. Kernighan, A modeling language for mathematical programming, Management Science 36 (1990) 519–554. doi:https://doi.org/10.1287/mnsc.36.5.519, Software URL: https://ampl.com/.

[14] AIMMS Language Reference, 2022. URL: documentation.aimms.com/language-reference/index.html, accessed on July 14, 2023.

[15] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, D. L. Woodruff, Pyomo–optimization modeling in python, volume 67, third ed., Springer Science & Business Media, 2021. Software URL: https://pyomo.readthedocs.io/en/stable/,
⋆ ⋆ This book provides a guide to the Python-based open-source algebraic modeling language, Pyomo. It discusses the modeling capabilities in Pyomo with extensive illustrative examples.

[16] M. Lubin, O. Dowson, J. D. Garcia, J. Huchette, B. Legat, J. P. Vielma, Jump 1.0: Recent improvements to a modeling language for mathematical optimization, Mathematical Programming Computation (2023). doi:10.1007/s12532-023-00239-3, Software URL: https://jump.dev/JuMP.jl/stable/,
⋆ ⋆ This paper describes a decade of developments for the JuMP algebraic modeling language that led to the release of JuMP 1.0, including the migration to MathOptInterface, which allows greater flexibility with solver parameter settings.

[17] B. Nicholson, J. D. Siirola, J.-P. Watson, V. M. Zavala, L. T. Biegler, pyomo.dae: a modeling and automatic discretization framework for optimization with differential and algebraic equations, Mathematical Programming Computation 10 (2018) 187–223. doi:https://doi.org/10.1007/s12532-017-0127-0, Software URL: https://pyomo.readthedocs.io/en/stable/modeling_extensions/dae.html.

[18] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, M. Diehl, CasADi

– A software framework for nonlinear optimization and optimal control, Mathematical Programming Computation 11 (2019) 1–36. doi:10.1007/s12532-018-0139-4, Software URL: https://web.casadi.org/.

[19] P. Fritzson, A. Pop, K. Abdelhak, A. Ashgar, B. Bachmann, W. Braun, D. Bouskela, R. Braun, L. Buffoni, F. Casella, R. Castro, R. Franke, D. Fritzson, M. Gebremedhin, A. Heuermann, B. Lie, A. Mengist, L. Mikelsons, K. Moudgalya, L. Ochel, A. Palanisamy, V. Ruge, W. Schamai, M. Sjölund, B. Thiele, J. Tinnerholm, P. Östlund, The OpenModelica integrated environment for modeling, simulation, and model-based development, Modeling, Identification and Control 41 (2020) 241–295. doi:10.4173/mic.2020.4.1, Software URL: https://openmodelica.org/.

[20] L. Beal, D. Hill, R. A. Martin, J. Hedengren, GEKKO optimization suite, 2018. doi:https://doi.org/10.3390/pr6080106, Software URL: https://apmonitor.com/wiki/index.php.

[21] A. Lee, J. H. Ghouse, J. C. Eslick, C. D. Laird, J. D. Siirola, M. A. Zamarripa, D. Gunter, J. H. Shinn, A. W. Dowling, D. Bhattacharyya, L. T. Biegler, A. P. Burgard, D. C. Miller, The IDAES process modeling framework and model library—flexibility for process simulation and optimization, Journal of Advanced Manufacturing and Processing 3 (2021) e10095. doi:https://doi.org/10.1002/amp2.10095, Software URL: https://idaes-pse.readthedocs.io/en/stable/.

⋆ ⋆ This paper details the core modeling framework for IDAES, an open-source modeling environment that integrates model libraries with the Pyomo algebraic modeling language to enable large-scale dynamic optimization of energy systems under uncertainty. A case study simulation and stochastic optimization are presented on the hydroalkylation of toluene to produce benzene.

[22] ASL User Guide, 2016. URL: https://github.com/AvtechScientific/ASL/wiki/User-Guide, accessed on July 14, 2023.

[23] Aspen Custom Modeler®, 2023. URL: https://www.aspentech.com/en/products/engineering/aspen-custom-modeler, accessed on July 14, 2023.

[24] Aspen Plus Dynamics, 2023. URL: /www.aspentech.com/en/products/engineering/aspen-plus-dynamics, accessed on July 14, 2023.

[25] Y. Cortes-Peña, D. Kumar, V. Singh, J. S. Guest, BioSTEAM: A fast and flexible platform for the design, simulation, and techno-economic analysis of biorefineries under uncertainty, ACS Sustainable Chemistry & Engineering 8 (2020) 3302–3310. doi:https://doi.org/10.1021/acssuschemeng.9b07040, Software URL: https://biosteam.readthedocs.io/en/latest/.

[26] Cape-Open to Cape-Open simulation environment, https://www.cocosimulator.org/index.html, 2022. Accessed on July 14, 2023.

[27] V. Skorych, M. Dosta, S. Heinrich, Dyssol—an open-source flowsheet simulation framework for particulate materials, SoftwareX 12 (2020) 100572. doi:https://doi.org/10.1016/j.softx.2020.100572, Software URL: https://www.dyssoltec.com/.

[28] Sphera®, Sphera | Sustainability, operational risk management & EHS software, 2023. URL: https://sphera.com, accessed on July 14, 2023.

[29] Process Systems Enterprise, gPROMS® v. 2022.1.0, 2022. URL: https://www.psenterprise.com/products/gproms, accessed on July 14, 2023.

[30] M. Wang, A. Elgowainy, U. Lee, A. Bafana, P. T. Benavides, A. Burnham, H. Cai, Q. Dai, U. R. Gracida-Alvarez, T. R. Hawkins, P. V. Jaquez, J. C. Kelly, H. Kwon, Z. Lu, X. Liu, L. Ou, P. Sun, O. Winjobi, H. Xu,

E. Yoo, G. G. Zaimes, G. Zang, Summary of expansions and updates in GREET® 2020 (2020). doi:doi.org/10.2172/1671788, Software URL: https://greet.es.anl.gov/greet/documentation.html.

[31] A. Ciroth, ICT for environment in life cycle applications openLCA — a new open source software for life cycle assessment, The International Journal of Life Cycle Assessment 12 (2007) 209–210. doi:https://doi.org/10.1065/lca2007.06.337, Software URL: https://www.openlca.org.

[32] D. Casas-Orozco, D. Laky, V. Wang, M. Abdi, X. Feng, E. Wood, C. Laird, G. V. Reklaitis, Z. K. Nagy, Pharmapy: An object-oriented tool for the development of hybrid pharmaceutical flowsheets, Computers & Chemical Engineering 153 (2021) 107408. doi:https://doi.org/10.1016/j.compchemeng.2021.107408, Software URL: https://github.com/CryPTSys/PharmaPy.

[33] AVEVA Group Limited, AVEVA™ PRO/II™ Simulation, 2023. URL: https://www.aveva.com/en/products/pro-ii-simulation/, accessed on July 14, 2023.

[34] PRe Sustainability B.V., SimaPro | LCA software for informed Change-Makers, https://simapro.com/, 2020. Accessed on July 14, 2023.

[35] Umberto® LCA+ v10 user Manual, ifu Hamburg GmbH, Max-Brauer-Allee 50, 22765 Hamburg / Germany, 2017. URL: https://www.ifu.com/umberto/lca-software/, accessed on July 14, 2023.

[36] A. K. Tula, M. R. Eden, R. Gani, Hybrid method and associated tools for synthesis of sustainable process flowsheets, Computers & Chemical Engineering 131 (2019) 106572. doi:https://doi.org/10.1016/j.compchemeng.2019.106572.

[37] K. Udomwong, A. Robin, N. Kuprasertwong, O. Padungwatanaroj, A. K. Tula, L. Zhu, L. Zhou, S. Wang, R. Gani, B. Wange, ProREFD: Tool

for automated computer-aided refrigerant design, analysis, and verification, in: M. Türkay, R. Gani (Eds.), 31st European Symposium on Computer Aided Process Engineering, volume 50 of *Computer Aided Chemical Engineering*, Elsevier, 2021, pp. 457–462. doi:https://doi.org/10.1016/B978-0-323-88506-5.50072-3, Software URL: https://www.pseforspeed.com/cpt_product/prorefd/.

[38] V. Jusevičius, R. Oberdieck, R. Paulavičius, Experimental analysis of algebraic modelling languages for mathematical optimization, Informatica 32 (2021) 283–304. doi:10.15388/21-INFOR447.

[39] I. E. Grossmann, Advances in mathematical programming models for enterprise-wide optimization, Computers & Chemical Engineering 47 (2012) 2–18. doi:https://doi.org/10.1016/j.compchemeng.2012.06.038.

[40] M. Rafiei, L. A. Ricardez-Sandoval, New frontiers, challenges, and opportunities in integration of design and control for enterprise-wide sustainability, Computers & Chemical Engineering 132 (2020) 106610. doi:https://doi.org/10.1016/j.compchemeng.2019.106610.

[41] J. Oliveira, M. Jin, R. Lima, J. Kobza, J. Montevechi, The role of simulation and optimization methods in supply chain risk management: Performance and review standpoints, Simulation Modelling Practice and Theory 92 (2019) 17–44. URL: https://www.sciencedirect.com/science/article/pii/S1569190X18301734. doi:https://doi.org/10.1016/j.simpat.2018.11.007.

[42] A. Bynum, C. Laird, F. Ceccon, J. Siirola, Coramin, python-based tools for developing tailored algorithms for mixed-integer nonlinear programming problems, 2022. URL: https://github.com/Coramin/Coramin.

[43] D. E. Bernal, Q. Chen, F. Gong, I. E. Grossmann, Mixed-integer nonlinear decomposition toolbox for pyomo (mindtpy), in: Computer Aided Chemical Engineering, volume 44, Elsevier, 2018, pp. 895–900.

[44] J. S. Rodriguez, R. B. Parker, C. D. Laird, B. L. Nicholson, J. D. Siirola, M. L. Bynum, Scalable parallel nonlinear optimization with pynumero and parapint, INFORMS Journal on Computing 35 (2023) 509–517.

[45] O. N. Team, Octeract neural, solve nonlinear optimization problems with generative ai, 2023. URL: https://octeract.gg/, accessed on August 31, 2023.

[46] R. Mahmud, S. M. Moni, K. High, M. Carbajales-Dale, Integration of techno-economic analysis and life cycle assessment for sustainable process design–a review, Journal of Cleaner Production 317 (2021) 128247. doi:https://doi.org/10.1016/j.jclepro.2021.128247,
⋆ The authors present a comprehensive review of the convergence of techno-economic analysis and life cycle analysis for sustainable process design and the development of emerging technologies. They highlight the need for standardized guidelines and software compatibility across these two methodologies.

[47] D. A. L. Silva, A. O. Nunes, C. M. Piekarski, V. A. da Silva Moris, L. S. M. de Souza, T. O. Rodrigues, Why using different life cycle assessment software tools can generate different results for the same product system? A cause–effect analysis of the problem, Sustainable Production and Consumption 20 (2019) 304–315. doi:https://doi.org/10.1016/j.spc.2019.07.005.

[48] L. Zhang, H. Mao, Q. Liu, R. Gani, Chemical product design–recent advances and perspectives, Current Opinion in Chemical Engineering 27 (2020) 22–34. doi:https://doi.org/10.1016/j.coche.2019.10.005.

[49] N. D. Austin, N. V. Sahinidis, D. W. Trahan, Computer-aided molecular design: An introduction and review of tools, applications, and solution techniques, Chemical Engineering Research and Design 116 (2016) 2–26. doi:https://doi.org/10.1016/j.cherd.2016.10.014.

[50] S. Kalakul, L. Zhang, Z. Fang, H. A. Choudhury, S. Intikhab, N. Elbashir, M. R. Eden, R. Gani, Computer aided chemical product design – ProCAPD and tailor-made blended products, Computers & Chemical Engineering 116 (2018) 37–55. doi:https://doi.org/10.1016/j.compchemeng.2018.03.029, Software URL: https://www.pseforspeed.com/cpt_product/procapd/.

[51] PSE Tools Overview, https://www.pseforspeed.com/tools/, 2021. Accessed on July 14, 2023.

[52] A. K. Tula, M. R. Eden, R. Gani, Computer-aided process intensification: Challenges, trends and opportunities, AIChE Journal 66 (2020) e16819. doi:https://doi.org/10.1002/aic.16819.

[53] MATLAB®, v. 9.14.0 (R2023a), 2023. URL: https://www.mathworks.com, accessed on July 14, 2023.

[54] COMSOL Multiphysics® v. 6.1, 2022. URL: https://www.comsol.com, accessed on July 14, 2023.

[55] AspenTech®, 2023. URL: www.aspentech.com/en/products/full-product-listing, accessed on July 14, 2023.

[56] Simcenter STAR-CCM+, v. 2306, 2023. URL: https://www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html, accessed on July 14, 2023.

[57] Y. Chen, O. Yang, C. Sampat, P. Bhalode, R. Ramachandran, M. Ierapetritou, Digital twins in pharmaceutical and biopharmaceutical manufacturing: A literature review, Processes 8 (2020). URL: https://www.mdpi.com/2227-9717/8/9/1088. doi:10.3390/pr8091088.

[58] J. Sansana, M. N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L. H. Chiang, M. S. Reis, Recent trends on hybrid modeling for industry 4.0, Computers & Chemical Engineering 151 (2021) 107365. doi:https:

//doi.org/10.1016/j.compchemeng.2021.107365.

⋆ This review concisely discusses advancements in hybrid modeling techniques over the past three decades—serial, parallel, surrogate, and alternative modeling structures—and other related modeling techniques such as data-driven modeling (e.g., neural networks, support vector machines) and system identification methodologies. The authors emphasize the need for modeling frameworks that balance prior knowledge with big data.

[59] Z. T. Wilson, N. V. Sahinidis, The ALAMO approach to machine learning, Computers & Chemical Engineering 106 (2017) 785–795. doi:https://doi.org/10.1016/j.compchemeng.2017.02.010, Software URL: https://www.minlp.com/alamo-modeling-tool.

[60] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: https://www.tensorflow.org/, Software URL: https://www.tensorflow.org/.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830. Software URL: https://scikit-learn.org/stable/.

[62] Surrogates.jl: Surrogate models and optimization for scientific machine learning, 2022. URL: https://docs.sciml.ai/Surrogates/stable/, accessed on July 14, 2023.

[63] G. Shulkind, L. Horesh, H. Avron, Experimental design for nonparametric correction of misspecified dynamical models, SIAM/ASA Journal on Uncertainty Quantification 6 (2018) 880–906. doi:https://doi.org/10.1137/17M1128435.

[64] M. von Stosch, S. Davy, K. Francois, V. Galvanauskas, J.-M. Hamelink, A. Luebbert, M. Mayer, R. Oliveira, R. O'Kennedy, P. Rice, J. Glassey, Hybrid modeling for quality by design and PAT-benefits and challenges of applications in biopharmaceutical industry, Biotechnology Journal 9 (2014) 719–726. doi:https://doi.org/10.1002/biot.201300385.

[65] Novasign Hybrid Modeling Toolbox, 2022. URL: https://www.novasign.at/products-services-toolbox, accessed on July 14, 2023.

[66] K. Merkelbach, A. M. Schweidtmann, Y. Müller, P. Schwoebel, A. Mhamdi, A. Mitsos, A. Schuppert, T. Mrziglod, S. Schneckener, Hybridml: Open source platform for hybrid modeling, Computers & Chemical Engineering 160 (2022) 107736. doi:https://doi.org/10.1016/j.compchemeng.2022.107736, Software URL: https://github.com/JRC-COMBINE/HybridML.

[67] W. Bradley, J. Kim, Z. Kilwein, L. Blakely, M. Eydenberg, J. Jalvin, C. Laird, F. Boukouvala, Perspectives on the integration between first-principles and data-driven modeling, Computers & Chemical Engineering 166 (2022) 107898. doi:https://doi.org/10.1016/j.compchemeng.2022.107898,

⋆ This review paper describes the integration of first-principles and data-driven modeling, providing informed perspectives on the strengths, limitations, and recent advancements in modeling paradigms that couple first-principles knowledge with data.

[68] J. Stork, M. Friese, M. Zaefferer, T. Bartz-Beielstein, A. Fischbach, B. Breiderhoff, B. Naujoks, T. Tušar, Open issues in surrogate-assisted op-

timization, High-performance simulation-based optimization (2020) 225–244. doi:https://doi.org/10.1007/978-3-030-18764-4_10.

[69] F. Ceccon, J. Jalving, J. Haddad, A. Thebelt, C. Tsay, C. D. Laird, R. Misener, OMLT: Optimization & machine learning toolkit, 2022. arXiv:2202.02414, Software URL: https://omlt.readthedocs.io/en/latest/index.html,

⋆ The authors develop OMLT, an open-source tool for integrating surrogate models—neural networks (NNs) and gradient-boosted trees (GBTs)—with larger optimization problems for decision-making. This tool automates the translation of NNs and GBTs into mathematical programming syntax, allowing users to embed trained surrogates into optimization formulations in algebraic modeling languages.

[70] J. P. Eason, L. T. Biegler, A trust region filter method for glass box/black box optimization, AIChE Journal 62 (2016) 3124–3136.

[71] X. Chen, K. Wu, A. Bai, C. M. Masuku, J. Niederberger, F. S. Liporace, L. T. Biegler, Real-time refinery optimization with reduced-order fluidized catalytic cracker model and surrogate-based trust region filter method, Computers & Chemical Engineering 153 (2021) 107455. doi:https://doi.org/10.1016/j.compchemeng.2021.107455.

[72] M. C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63 (2001) 425–464. doi:https://doi.org/10.1111/1467-9868.00294.

[73] B. Knueven, D. Mildebrath, C. Muir, J. D. Siirola, J.-P. Watson, D. L. Woodruff, A parallel hub-and-spoke system for large-scale scenario-based optimization under uncertainty, 2020. Software URL: https://github.com/Pyomo/mpi-sppy.

[74] N. M. Isenberg, P. Akula, J. C. Eslick, D. Bhattacharyya, D. C.

Miller, C. E. Gounaris, A generalized cutting-set approach for nonlinear robust optimization in process systems engineering, AIChE Journal 67 (2021) e17175. Software URL: https://pyomo.readthedocs.io/en/stable/contributed_packages/pyros.html.

[75] B. Adams, W. Bohnhoff, K. Dalbey, M. Ebeida, J. Eddy, M. Eldred, R. Hooper, P. Hough, K. Hu, J. Jakeman, M. Khalil, K. Maupin, J. Monschke, E. Ridgway, A. Rushdi, D. Seidl, J. Stephens, L. Swiler, J. Winokur, Dakota, A multilevel parallel object-Oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.15 User's Manual, Technical Report SAND2020-12495, Sandia National Laboratories, 2021. doi:https://doi.org/10.2172/1630693, Software URL: https://dakota.sandia.gov/.

[76] R. A. Richardson, D. W. Wright, W. Edeling, V. Jancauskas, J. Lakhlili, P. V. Coveney, EasyVVUQ: A library for verification, validation and uncertainty quantification in high performance computing, Journal of Open Research Software (2020). doi:10.5334/jors.303, Software URL: https://easyvvuq.readthedocs.io/en/dev/.

[77] D. W. Wright, R. A. Richardson, W. Edeling, J. Lakhlili, R. C. Sinclair, V. Jancauskas, D. Suleimenova, B. Bosak, M. Kulczewski, T. Piontek, P. Kopta, I. Chirca, H. Arabnejad, O. O. Luk, O. Hoenen, J. Węglarz, D. Crommelin, D. Groen, P. V. Coveney, Building confidence in simulation: Applications of EasyVVUQ, Advanced Theory and Simulations 3 (2020) 1900246. doi:https://doi.org/10.1002/adts.201900246, Software URL: https://easyvvuq.readthedocs.io/en/dev/.

[78] J.-M. Bourinet, C. Mattrand, V. Dubourg, A review of recent features and improvements added to FERUM software, in: H. Furuta, D. M. Frangopol, M. Shinozuka (Eds.), Proc. 10$^{\text{th}}$ International Conference on Structural Safety and Reliability (ICOSSAR 2009), Osaka, Japan, September 13–17,

2009, CRC Press, 2009. Software URL: https://www.sigma-clermont.fr/en/ferum.

[79] D. Miller, D. Agarwal, D. Bhattacharyya, J. Boverhof, Y. Chen, J. Eslick, J. Leek, J. Ma, P. Mahapatra, B. Ng, N. Sahinidis, C. Tong, S. Zitney, Innovative computational tools and models for the design, optimization and control of carbon capture processes, in: A. Papadopoulos, P. Seferlis (Eds.), Process Systems and Materials for CO_2 Capture: Modelling, Design, Control and Integration, John Wiley & Sons Ltd, Chichester, UK, 2017, pp. 311–342.

[80] M. Parno, A. Davis, L. Seelinger, MUQ: The MIT uncertainty quantification library, Journal of Open Source Software 6 (2021) 3076. doi:https://doi.org/10.21105/joss.03076, Software URL: https://mituq.bitbucket.io/source/_site/index.html.

[81] E. Patelli, COSSAN: A multidisciplinary software suite for uncertainty quantification and risk management, in: Handbook of Uncertainty Quantification, Springer International Publishing, 2015, pp. 1–69. doi:10.1007/978-3-319-11259-6_59-1, Software URL: https://cossan.co.uk/software/open-cossan-engine.php.

[82] M. Baudin, A. Dutfoy, B. Iooss, A.-L. Popelin, Open turns: An industrial software for uncertainty quantification in simulation, 2015. arXiv:1501.05242, Software URL: https://openturns.github.io/www/index.html.

[83] PSUADE: Software, 2018. URL: https://computing.llnl.gov/projects/psuade/software, accessed on July 14, 2023.

[84] D. Gorissen, K. Crombecq, T. D. I. Couckuyt, P. Demeester, A surrogate modeling and adaptive sampling toolbox for computer based design, Journal of Machine Learning Research 11 (2010) 2051–2055. Software URL: https://sumo.ilabt.imec.be/home/software/sumo.

[85] SMARTUQ® - AI for an Uncertain World, 2023. URL: https://www.smartuq.com/, accessed on July 14, 2023.

[86] S. Marelli, B. Sudret, UQLab: A framework for uncertainty quantification in MATLAB®, 2014, pp. 2554–2563. doi:https://doi.org/10.1061/9780784413609.257, Software URL: https://www.uqlab.com/.

[87] A. Olivier, D. G. Giovanis, B. Aakash, M. Chauhan, L. Vandanapu, M. D. Shields, UQPy: A general purpose Python package and development environment for uncertainty quantification, Journal of Computational Science 47 (2020) 101204. doi:https://doi.org/10.1016/j.jocs.2020.101204, Software URL: https://uqpyproject.readthedocs.io/en/latest/.

[88] B. J. Debusschere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, O. P. Le Maître, Numerical challenges in the use of polynomial chaos representations for stochastic processes, SIAM Journal on Scientific Computing 26 (2004) 698–719. doi:http://dx.doi.org/10.1137/S1064827503427741, Software URL: https://www.sandia.gov/uqtoolkit/.

[89] B. Debusschere, K. Sargsyan, C. Safta, K. Chowdhary, The uncertainty quantification toolkit (UQTk), in: R. Ghanem, D. Higdon, H. Owhadi (Eds.), Handbook of Uncertainty Quantification, Springer, 2017, pp. 1807–1827. doi:https://doi.org/10.1007/978-3-319-12385-1, Software URL: https://www.sandia.gov/uqtoolkit/.

[90] J. C. Morgan, A. S. Chinen, C. Anderson-Cook, C. Tong, J. Carroll, C. Saha, B. Omell, D. Bhattacharyya, M. Matuszewski, K. S. Bhat, et al., Development of a framework for sequential bayesian design of experiments: Application to a pilot-scale solvent-based CO_2 capture process, Applied Energy 262 (2020) 114533. doi:https://doi.org/10.1016/j.apenergy.2020.114533.

[91] F. B. Soepyan, C. M. Anderson-Cook, J. C. Morgan, C. H. Tong, D. Bhattacharyya, B. P. Omell, M. S. Matuszewski, K. S. Bhat, M. A. Zamar-

ripa, J. C. Eslick, et al., Sequential design of experiments to maximize learning from carbon capture pilot plant testing, in: Computer Aided Chemical Engineering, volume 44, Elsevier, 2018, pp. 283–288. doi:https://doi.org/10.1016/B978-0-444-64241-7.50042-2.

[92] M. Van der Spek, T. Fout, M. Garcia, V. N. Kuncheekanna, M. Matuszewski, S. McCoy, J. Morgan, S. M. Nazir, A. Ramirez, S. Roussanaly, et al., Uncertainty analysis in the techno-economic assessment of CO_2 capture and storage technologies. critical review and guidelines for use, International Journal of Greenhouse Gas Control 100 (2020) 103113. doi:https://doi.org/10.1016/j.ijggc.2020.103113.

[93] D. C. Montgomery, Design and analysis of experiments, John Wiley & Sons, Hoboken, NJ, 2017.

[94] J. Wang, A. W. Dowling, Pyomo.DOE: An open-source package for model-based design of experiments in Python, AIChE Journal 68 (2022) e17813. Software URL: https://pyomo.readthedocs.io/en/stable/contributed_packages/doe/doe.html.

⋆ This paper introduces Pyomo.DOE, an open-source Pyomo package for model-based design of experiment (MBDoE). It includes a mathematical primer on MBDoE and illustrative case studies on reaction kinetics and fixed-bed breakthrough experiments.

[95] K. P. Kusumo, K. Kuriyan, S. Vaidyaraman, S. García-Muñoz, N. Shah, B. Chachuat, Risk mitigation in model-based experiment design: a continuous-effort approach to optimal campaigns, Computers & Chemical Engineering 159 (2022) 107680. doi:https://doi.org/10.1016/j.compchemeng.2022.107680.

[96] Design-Expert®, 2023. URL: https://www.statease.com/docs/v11/, accessed on July 14, 2023.

[97] dexpy - Design of Experiments (DOE) in Python, 2017. URL: https://statease.github.io/dexpy/, accessed on July 14, 2023.

[98] Welcome to DOEPY. Design of Experiment Generator in Python, 2019. URL: https://doepy.readthedocs.io/en/latest/, accessed on July 14, 2023.

[99] S. Olofsson, L. Hebing, S. Niedenführ, M. P. Deisenroth, R. Misener, Gpdoemd: A python package for design of experiments for model discrimination, Computers & Chemical Engineering 125 (2019) 54–70. doi:https://doi.org/10.1016/j.compchemeng.2019.03.010, Software URL: https://github.com/cog-imperial/GPdoemd.

[100] JMP Documentation, 2023. URL: https://www.jmp.com/en_us/support/jmp-documentation.html, accessed on July 14, 2023.

[101] Minitab 17 Statistical Software, 2010. URL: www.minitab.com, accessed on July 14, 2023.

[102] Design of Experiments Software That Accelerates Progress, 2023. URL: https://www.sartorius.com/en/products/process-analytical-technology/data-analytics-software/doe-software/modde, accessed on July 14, 2023.

[103] NCSS 2023 Statistical Software, 2023. URL: https://www.ncss.com/software/ncss, accessed on July 14, 2023.

[104] pyDOE: The experimental design package for python, 2014. URL: https://pythonhosted.org/pyDOE/, accessed on July 14, 2023.

[105] SAS® Documentation, 2023. URL: https://support.sas.com/en/documentation.html, accessed on July 14, 2023.

[106] Statgraphics19® Resources Downloads, 2022. URL: https://www.statgraphics.com/resources-downloads, accessed on July 14, 2023.

[107] Design of Experiments (DOE), 2023. URL: https://www.mathworks.com/help/stats/design-of-experiments-1.html, accessed on July 14, 2023.

[108] Synthace: Digital Experiment Platform, 2023. URL: https://www.synthace.com/, accessed on July 14, 2023.

[109] U. Groemping, T. Morgan-Wall, CRAN Task View: Design of Experiments (DoE) & Analysis of Experimental Data. Version 2023-04-05., 2023. URL: https://cran.r-project.org/web/views/ExperimentalDesign.html, accessed on July 14, 2023.

[110] M. Di Martino, S. Avraamidou, E. N. Pistikopoulos, A neural network based superstructure optimization approach to reverse osmosis desalination plants, Membranes 12 (2022) 199. doi:https://doi.org/10.3390/membranes12020199,

⋆ ⋆ This work demonstrates model-based design for multi-objective optimization of a network of membrane modules for reverse osmosis water desalination. The framework uses neural network surrogates to encode information from computational fluid dynamics simulations in the optimization formulation.

[111] K. K. Razman, M. M. Hanafiah, A. W. Mohammad, An overview of LCA applied to various membrane technologies: Progress, challenges, and harmonization, Environmental Technology & Innovation (2022) 102803. doi:https://doi.org/10.1016/j.eti.2022.102803.

[112] A. Cherif, R. Nebbali, C.-J. Lee, Design and multiobjective optimization of membrane steam methane reformer: A computational fluid dynamic analysis, International Journal of Energy Research 46 (2022) 8700–8715. doi:https://doi.org/10.1002/er.7750.

[113] U. Sarma, S. Ganguly, An energy management optimization approach for proton exchange membrane fuel cell-battery hybrid energy system for railway applications, Electrical Engineering 104 (2022) 4179–4195. doi:https://doi.org/10.1007/s00202-022-01617-1.

[114] N. P. Wamble, E. A. Eugene, W. A. Phillip, A. W. Dowling, Optimal

diafiltration membrane cascades enable green recycling of spent lithium-ion batteries, ACS Sustainable Chemistry & Engineering 10 (2022) 12207–12225. doi:https://doi.org/10.1021/acssuschemeng.2c02862.

[115] B. Zach, J. Pluskal, R. Šomplák, J. Jadrný, M. Šyc, Tool for optimization of energy consumption of membrane-based carbon capture, Journal of Environmental Management 320 (2022) 115913. URL: https://www.sciencedirect.com/science/article/pii/S0301479722014864. doi:https://doi.org/10.1016/j.jenvman.2022.115913.

[116] S. Khulu, S. Ncube, Y. Nuapia, L. M. Madikizela, H. Tutu, H. Richards, K. Ndungu, E. Mavhunga, L. Chimuka, Multivariate optimization of a two-way technique for extraction of pharmaceuticals in surface water using a combination of membrane assisted solvent extraction and a molecularly imprinted polymer, Chemosphere 286 (2022) 131973. doi:https://doi.org/10.1016/j.chemosphere.2021.131973.

[117] S. Bayazidi, M. Mojaddam, A. Mohseni, Performance optimization of nozzle-diffuser piezoelectric micropump with multiple vibrating membranes by design of experiment (DOE) method, Journal of Applied Fluid Mechanics 16 (2023) 1356–1370. doi:10.47176/jafm.16.07.1539.