Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

A Comparative Evaluation of Complexity in Mechanistic and Surrogate Modeling Approaches for Digital Twins

Shreyas Parbat[a]*, Isabell Viedt[a,c], Leon Urbas[a,b,c]

[a] Technische Universität Dresden, Process-to-Order Group, Process Systems Engineering Group, Dresden, Germany
[b] Technische Universität Dresden, Process-to-Order Group, Chair of Process Control Systems, Dresden, Germany
[c] Technische Universität Dresden, Process-to-Order Group, Process-to-Order Lab Learning Factory, Dresden, Germany
* Corresponding Author: shreyas.parbat@tu-dresden.de.

ABSTRACT

A Digital Twin (DT) is a purposeful digital representation of a physical entity that employs data, algorithms, and software to enhance operations, making it possible to e.g., forecast failures, or evaluate new designs through the simulation of real-world scenarios. DTs are enablers for real-time monitoring, simulation, and optimization. However, traditional simulation DTs often rely on complex, non-linear mechanistic models with high computational demands, complex structures, and a large number of specific parameters and thus pose quite a challenge to maintainability. Surrogate models, on the other hand, are simplified approximations of more complex, higher-order models. These approximations are typically built using data-driven approaches, such as Random Forest Regression, facilitating faster simulations, simpler adaptation, and quicker deployment. This study analyzes the complexity of mechanistic and surrogate modeling approaches in the context of DTs to aid model selection. A model with reduced complexity enhances computational efficiency, simplifies implementation, and supports real-time monitoring and predictive maintenance. Complexity analysis evaluates metrics such as analytical, structural, space, behavioral, training, and prediction complexity, resulting in an overall complexity score for model selection. However, the decision involves trade-offs, such as balancing high fidelity with low complexity or prioritizing high explainability over structural simplicity. Addressing these trade-offs is essential in selecting a model that balances the accuracy, usability, and efficiency of DTs. Using a stirred tank reactor as a use case, the mechanistic model is compared to a surrogate model to quantify complexity scores and select a less complex model for DT development.

Keywords: Surrogate Model, Digital Twin, Complexity metric, Complexity Score, Mechanistic Model

INTRODUCTION

Modeling and simulations provide the mathematical and conceptual foundation for Digital Twins (DTs) [1]. Multiple industries, including pharmaceuticals, manufacturing, chemicals, and food processing, heavily rely on modeling and simulations. One of the important aspects of the model-building process is model selection [2].

Mechanistic models are a critical foundation for DTs [3]. These mechanistic models are non-linear, time-consuming, expensive, and sometimes complex [4]. In such cases, surrogate models are used to approximate or regress the input-output relationships of a more complex, computationally demanding model [5] to reduce complexity by focusing on key elements and interactions, sacrificing some accuracy for efficiency and ease of use. Thermodynamic models for distillation and extraction processes, partial differential equations (PDEs), and computational fluid dynamics (CFD) simulations are examples of complex computational systems. Surrogate modeling techniques can effectively approximate these systems, reducing computational cost while preserving reasonable accuracy [6].

However, defining the complexity of such models is a crucial metric for supporting decision-making and selecting the most beneficial and optimal model for a particular process. The complexity level of a model is determined by its components, which depend on factors such as resolution, scope, size, and the interactions within the model. Assessing the complexity of a model is valuable

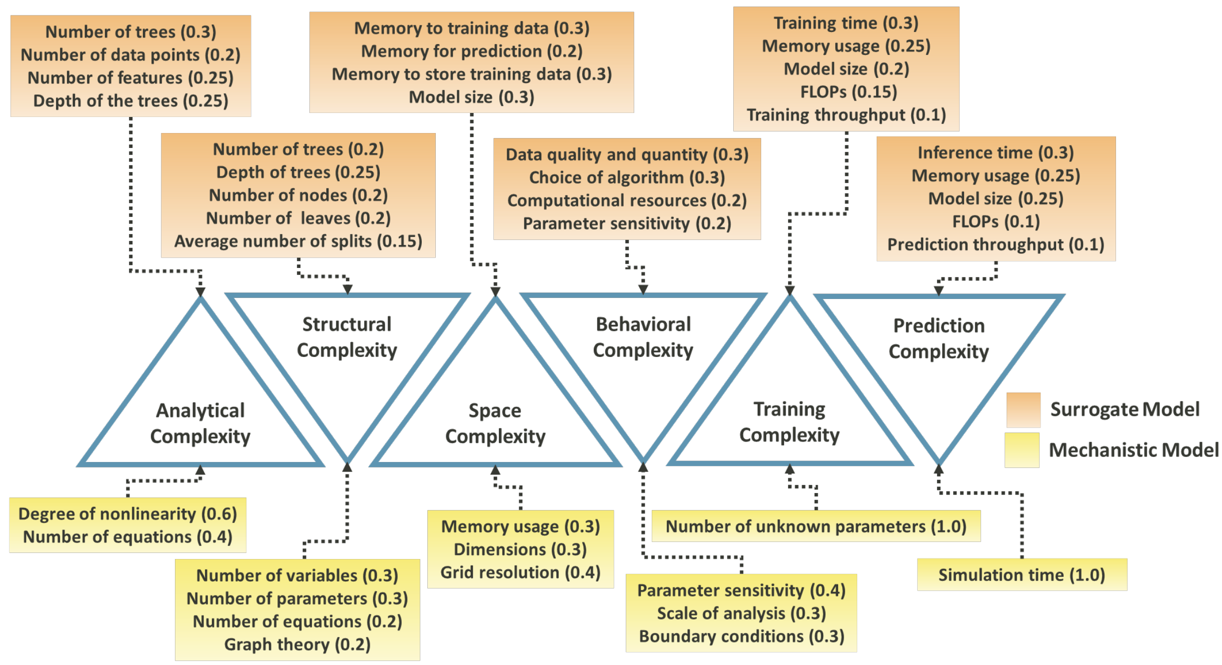

Figure 1: Complexity Metrics (analytical, structural, space, behavioral, training, and prediction complexity) and complexity contributors with individual weights

for several applications and is guided by several core principles that process engineers should consider [7]. For example, a Continuous Stirred Tank Reactor (CSTR) model may suffice for quick reactor analysis, but a CFD model might be needed for detailed heat or mass transfer studies. Process engineers must align model complexity with the application, data availability, and computational resources.

In this research, various complexity measures for mechanistic and surrogate models are investigated. The following section defines the different types of complexities associated with simulation models and analyzes their dependencies on several factors. The methodology for calculating complexity scores is presented, along with the rationale for assigning weights. A later section focuses on a simple use case of a stirred tank reactor, analyzing the complexities of both models in terms of reactor temperature. This case study illustrates the approach used to analyze model complexity, and the concept of the complexity score is then applied to these models. Finally, conclusions are drawn based on the proposed case study, and future work is discussed.

MODEL COMPLEXITY

Complexity can be defined as the state or quality of being intricate [8], with several components and their interactions within a system helping to determine its complexity. Thompson defines model complexity as “The aggregation of a model’s detail, external interactions and logical dependencies, and in many contexts its instantiation (input parameter settings) in a specific simulation scenario execution” [7]. In contrast, simple models are easier to understand, faster to implement, easier to verify and validate, more flexible, and simpler to modify [9]. However, some models are not easy to formulate, follow, validate, or verify, and such models can be classified as complex. In Modeling and Simulation, complexity can arise from the model itself or the software used to implement it [8].

Complexity Metrics

In this work, complexity metrics are defined based on analytical, structural, space, behavioral, training, and prediction complexity. Table 1 outlines several complexities associated with models. Fig. 1 shows the components and contributors of these complexities, highlighting their dependencies on several factors for the models, along with the individual weights of contributors.

Complexities discussed in Table 1 are essential as they cover the entire model lifecycle, from the structure and theoretical foundation to practical implementation and application. Each complexity type addresses a specific challenge in developing, deploying, and maintaining a model, making them collectively crucial for ensuring the overall effectiveness and applicability of the model.

Table 1: Complexity Metrics and Definitions.

Complexity metric	Definition
Analytical Complexity	Theoretical analysis of model complexity based on structure and operations
Structural Complexity	The complexity of the model architecture itself
Space Complexity	Memory required to store the model and data structures during the training and prediction phases
Behavioral Complexity	Complexity arising from how the model learns, adapts to data inputs, and responds to varying conditions
Training Complexity	Computational resources (time and space) required for model training
Prediction Complexity	Computational resources required for making predictions with a trained model

As Hu et al. [10] discuss, model complexity encompasses various dimensions critical to ensuring models are effective and scalable. These dimensions include analytical complexity, ensuring the theoretical foundation of a model is sound and aligned with its purpose, as unreliable designs can hinder scalability. Structural complexity addresses the model's architecture, balancing interpretability and usability with the risks of impracticality in overly complex designs. Space complexity focuses on memory usage, critical for deploying models in resource-constrained environments like edge devices or real-time systems. Behavioral complexity assesses how the model adapts to varying inputs, ensuring robustness and reliability in dynamic real-world conditions. Training complexity evaluates the computational cost and time needed for model development, especially important in large-scale or iterative applications. Prediction complexity highlights the need for efficient real-time performance in applications such as performance monitoring, optimization, and control systems. These complexities collectively ensure a model is effective, scalable, and practical.

Complexity Score

The simplest and most straightforward way to create an overall complexity score is the weighted aggregation of the individual complexity metrics. The overall complexity score and the individual complexity scores are calculated using Equations 1 and 2, respectively.

$$CS = \sum_{j=1}^{m}\left((CS_{ind})_j \cdot W_j\right) \tag{1}$$

$$CS_{ind} = \sum_{i=1}^{n} C_i \cdot w_i \tag{2}$$

$$C_i = \frac{Observed\ Value - Min\ Value}{Max\ Value - Min\ Value} \quad \forall\ i = 1 \dots n \tag{3}$$

where CS is the overall complexity score, CS_{ind} is the individual complexity score, C is the normalized complexity contributor value (e.g. Number of equations for the mechanistic model and Number of trees for the surrogate model) obtained using the min-max scaling normalization formula (Equation 3), w is the weight for the complexity contributors and W is the weight for the complexity.

Assignment of Weights to Complexity Metrics

The weights for the individual complexity metrics play an important role in determining the overall complexity score for a given model. These weights represent the relative contribution of each factor to the overall complexity of the model or system, determined primarily through logical reasoning and expert judgment. Table 2 provides the weights (W) assigned to the developed complexity metrics shown in Fig. 1.

Table 2: Weights (W) for Complexity Metrics.

Complexity metric	Weight
Analytical Complexity	0.15
Structural Complexity	0.2
Space Complexity	0.2
Behavioral Complexity	0.15
Training Complexity	0.15
Prediction Complexity	0.15

These weights (W) are selected based on the relative importance of each metric in the context of evaluating a model's overall complexity and considering the importance of each metric and contributor. Structural and space complexity metrics are weighted higher (0.2 each) because they have a more direct impact on the practical feasibility of the model, particularly in terms of usability, deployment, and hardware requirements [11]. Analytical, behavioral, training, and prediction metrics are equally weighted at 0.15, reflecting their secondary yet significant contributions [12]. They ensure the model is robust, adaptable, and efficient, but are less dominant than structural and space considerations in most applications.

Fig. 1 provides the weights (w) for individual contributors of model complexity, which are determined based on their relative contributions, primarily guided by logical reasoning and expert judgment. For mechanistic models, dominant contributors such as nonlinearity (analytical complexity) and grid resolution (space complexity) are assigned higher weights due to their significant anticipated computational impact [13]. In surrogate models, factors like tree depth and training time are prioritized, reflecting their expected influence on scalability and efficiency [11]. The weights are applied to ensure fair consideration of relevant factors, without allowing any single factor to disproportionately influence the outcome. Overlap, such as tree depth affecting both analytical and structural complexity, is mitigated through careful weight calibration. This approach ensures the complexity metrics highlight the most impactful aspects of each model.

COMPLEXITY EVALUATION – CASE OF A STIRRED TANK REACTOR

Reactor Temperature Modeling

To monitor and optimize energy consumption and loss in an insulated Stirred Tank Reactor (STR), a DT for energy Key Performance Indicators (eKPIs) needs to be developed. Understanding energy dynamics is crucial and can be done by analyzing the temperatures of the reactor and jacket. By determining the temperature gradient, the heat loss and process energy consumption between the reactor and the jacket can be calculated. Modular plants are known for their flexibility [14], relying on historical or statistical data for model development is often not feasible. Generating synthetic data enables the creation of more accurate and adaptive models. This case study applies the proposed complexity evaluation concept to guide model selection by comparing a mechanistic and a surrogate model, illustrating how the methodology aligns the model with system requirements and operational constraints.

Modeling of the STR is conducted using the mechanistic and a surrogate model. As shown in Fig. 2, the feed enters the reactor at a specified temperature. The reactor, with a volume of 4L, is agitated at a certain speed and equipped with a water-heated jacket. The fluid properties and the heat transfer coefficient are assumed to be constant, and there is no fouling. Energy balances (Equations 4 and 5) are used to obtain the temperature trajectories for both the reactor and the jacket.

$$\frac{dT_r}{dt} = \frac{1}{V_r \rho_r C_{pr}} * (kA(T_j - T_r)) \quad (4)$$

$$\frac{dT_j}{dt} = \frac{1}{V_j \rho_j C_{pj}} * (m_j C_{pj}(T_{jin} - T_j)) - (kA(T_j - T_r)) \quad (5)$$

Where, T_r is the reactor temperature, T_j is the jacket temperature, V_r is the reactor volume, ρ_r is the reactant density, C_{pr} is the reactant heat capacity, k is the heat transfer coefficient, A is the heat transfer area, V_j is the jacket volume, ρ_j is the jacket fluid density, C_{pj} is the jacket fluid heat transfer capacity, m_j is the jacket fluid mass and T_{jin} is the jacket inlet temperature. The initial temperature of reactor (T_r) and jacket (T_j) of 293.15 K and jacket inlet temperature (T_{jin}) of 283.15 K. The gPROMS simulation results, obtained after nine minutes of reaction time, are shown in Fig. 3. The generated data is used to train a Random Forest Regressor as the surrogate model. The reactor temperature predictions from the surrogate model (Mean Squared Error = 0.015) are shown in Fig. 4.

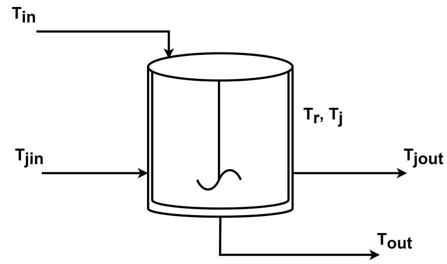

Figure 2. Stirred tank reactor: Energy flow.

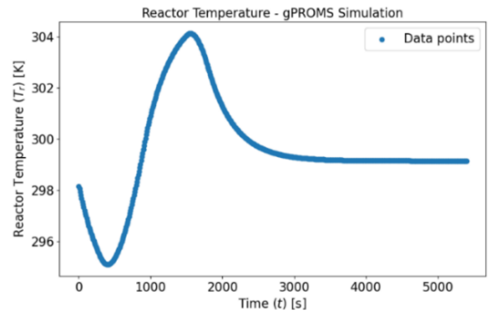

Figure 3. Reactor temperature: gPROMS simulation.

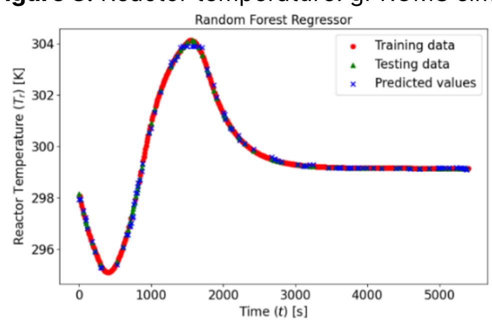

Figure 4. Reactor temperature: Random Forest model.

Complexity Analysis for the STR Case

The models are analyzed for complexity using the developed complexity metrics. In this study, a surrogate model was developed to enable a direct comparison of its complexity with that of the mechanistic model. The surrogate model would not always be developed unless complexity analysis identifies it as the more appropriate choice. A detailed analysis has been conducted to evaluate analytical, structural, space, behavioral, training, and prediction complexities.

Analytical Complexity

The analytical complexity of the mechanistic model is minimal due to its linear nature, with two equations and a constant heat transfer coefficient assumed in this study. This simplicity ensures computational efficiency. In contrast, the analytical complexity of the surrogate model arises from parameters like the number of trees (T=300), data points (N=440), features per split (F=1), and tree depth (D=7). While the surrogate model captures nonlinear behaviors effectively, this comes at a higher computational cost. This trade-off highlights the suitability of mechanistic models for systems with wellunderstood dynamics and the strength of surrogate models in handling complex, data-driven relationships.

Structural Complexity

The structural complexity of the mechanistic model is determined by ten parameters, two variables, two equations, and three boundary conditions, with a graph representation of twelve nodes and two edges. This ensures low complexity, simplicity, and interpretability. In contrast, the surrogate model exhibits considerably higher structural complexity, consisting of 300 trees with an average depth of seven. It contains 46,016 nodes, 23,158 leaves, and an average of 153.39 splits per tree. While this allows for greater flexibility and detailed system representation, it also increases storage and computational demands. This comparison highlights the suitability of mechanistic models for lightweight and interpretable applications, whereas surrogate models are ideal for capturing complex systems with detailed requirements.

Space Complexity

The space complexity of the mechanistic model is minimal, requiring storage for a 2×2 Jacobian matrix and a grid resolution of 1000 for discretization in two dimensions. For the surrogate model, space complexity includes memory for training data (0.006 MB), predictions (0 MB), and the model itself (3.25 MB), driven by the storage of tree structures. The low storage requirements of the mechanistic model make it ideal for resource-constrained, real-time applications. In contrast, the higher memory demands of the surrogate model may limit scalability but enable the handling of more complex systems with adequate computational infrastructure.

Behavioral Complexity

The behavioral complexity of the mechanistic model depends on system dynamics, parameter variability, and boundary conditions, making it reliable but less flexible for unknown dynamics. The surrogate model relies on data quality, algorithm choice, and computational resources to capture complex patterns. Parameter sensitivity and hyperparameter tuning also impact their performance. Mechanistic models are ideal for data-scarce systems due to their physical foundation, while surrogate models excel in handling high-variability behaviors, offering adaptability in data-rich scenarios. Synthetic data from mechanistic models can mitigate data scarcity and reduce the behavioral complexity of surrogate models.

Training Complexity

Training complexity for the mechanistic model is negligible since all parameters are known. For the surrogate model, training complexity is crucial, with metrics such as training time (117.32 s), peak memory usage (5.57 MB), model size (3.25 MB), and computational effort (924,000 FLOPs). With a throughput of 3.75 samples/second, the surrogate model demands considerable resources for data processing and hyperparameter tuning, making it the sole contributor to training complexity.

Prediction Complexity

In mechanistic models, simulation time depends on system dynamics, with this study reporting 2.2 s. Complex systems, though efficient, may have higher computational costs due to detailed calculations. The surrogate model, with a 0.02 s inference time, excels in rapid predictions. It uses 0 MB of memory, has a model size of 3.25 MB, a computational complexity of 2100 FLOPs, with a prediction throughput of 6850.39 predictions/seconds. Here, the mechanistic model is slower and more computationally expensive, while the surrogate model provides faster predictions with lower computational demands, making it ideal for high-throughput applications.

Complexity Score for the STR Case

Table 3 presents the individual complexity scores (CS_{ind}) for both modeling approaches, evaluated using Equation 2, along with dedicated weights for contributors and complexity metrics from Fig. 1.

Table 3: Individual complexity score of all complexities

Complexity Metric	Mechanistic Model	Surrogate Model
Analytical Complexity	0.4	0.533
Structural Complexity	0.9	1.0
Space Complexity	0.85	0.301
Behavioral Complexity	0.9	0.834
Training Complexity	0.0	0.95
Prediction Complexity	1.0	0.75

The complexity evaluation of both the mechanistic and surrogate models highlights the trade-offs between simplicity and flexibility. The mechanistic model scores higher in terms of space, behavioral, and prediction complexities, reflecting the efficiency and ease of implementation of the surrogate model. In contrast, the surrogate model exhibits higher analytical, structural, and training complexity, indicating that the mechanistic model is better suited to handle more complex behaviors.

Overall complexity scores (CS) for both models are determined using Equation 1, based on the individual complexity scores from Table 3 and their weights from Table 2. The overall complexity score shows that the surrogate model, with a CS of 0.72, exhibits higher complexity as compared to the mechanistic model, which has a

CS of 0.69 in this case study. This suggests that the surrogate model, while offering greater flexibility for complex scenarios, requires more resources, making the mechanistic model more suitable for simpler applications. These complexity scores enable developers to make informed choices based on the specific needs of the DT.

CONCLUSION AND OUTLOOK

The complexity of modeling for the simulation DT is analyzed using the developed complexity metric and score, applied to both mechanistic and surrogate models. The evaluation framework helps identify model components, gather requirements, and inform decisions on model selection, which is crucial for DT development. In the STR case, the surrogate model exhibits a higher complexity score than the mechanistic model. The mechanistic model benefits from simplicity due to its analytical foundation and minimal training requirements, while the increased complexity of the surrogate model arises from structural and training demands. While surrogate models may not always be the final choice, their development offers strategic flexibility and valuable insights, particularly in complex or data-driven scenarios. Dual-model approaches, though rare in real-world plants, improve model selection by addressing diverse challenges, making them valuable in complex, specialized applications.

Future work will focus on enhancing the model selection framework with accuracy and stability analyses. Efforts will aim at reducing surrogate model complexity, improving computational efficiency, and enabling real-time monitoring and predictive maintenance. Temperature-dependent parameters will introduce nonlinearity to the mechanistic model, and surrogate modeling will serve as an alternative if complexity increases. These ongoing developments will provide refined tools for model selection and further optimize DT applications.

ACKNOWLEDGEMENT

As part of the “Energy Efficiency in Industry”, research funding is provided by the Federal Ministry for Economic Affairs and Climate Action of Germany (BMWK) and Project Management Jülich (PtJ) under the title of project REUNION (FKZ 03EN2100C).

REFERENCES

1. Parbat S, Iqbal I, Viedt I, Urbas L. Architecture of digital twin for automating waste and recycling material sorting process. *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)* :1–4 (2024) https://doi.org/10.1109/ETFA61755.2024.10710670
2. Asprion N. Modeling, simulation, and optimization 4.0 for a distillation column. *Chemie Ingenieur Technik* 92:879–89 (2020) https://doi.org/10.1002/cite.202000001
3. Bárkányi Á, Chován T, Németh S, Abonyi J. Modelling for digital twins—potential role of surrogate models. *Processes* 9 (2021) https://doi.org/10.3390/pr9030476
4. Beisheim B, Rahimi-Adli K, Krämer S, Engell S. Energy performance analysis of continuous processes using surrogate models. *Energy* 183:776–87 (2019) https://doi.org/10.1016/j.energy.2019.05.176
5. McBride K, Sundmacher K. Overview of surrogate modeling in chemical process engineering. *Chemie Ingenieur Technik* 91:228–39 (2019) https://doi.org/10.1002/cite.201800091
6. Winz J, Nentwich C, Engell S. Surrogate modeling of thermodynamic equilibria: applications, sampling and optimization. *Chemie Ingenieur Technik* 93:1898–906 (2021) https://doi.org/10.1002/cite.202100092
7. Thompson JS, Hodson DD, Grimaila MR, Hanlon N, Dill R. Toward a simulation model complexity measure. *Information* 14 (2023) https://doi.org/10.3390/info14040202
8. Zhao X, Gray J. Towards a metrics suite for the complexity analysis of labview systems models. *Science of Computer Programming* 227:102931 (2023) https://doi.org/10.1016/j.scico.2023.102931
9. Robinson S. Exploring the relationship between simulation model accuracy and complexity. *Journal of the Operational Research Society* 74:1992–2011 (2023) https://doi.org/10.1080/01605682.2022.2122740
10. Hu X, Chu L, Pei J, Liu W, Bian J. Model complexity of deep learning: a survey. *Knowledge and Information Systems* 63:2585–619 (2021) https://doi.org/10.1007/s10115-021-01605-0
11. Paleyes A, Urma R-G, Lawrence ND. Challenges in deploying machine learning: a survey of case studies. *ACM Comput. Surv.* 55 (2022) https://doi.org/10.1145/3533378
12. Forrester AIJ, Bressloff NW, Keane AJ. Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 462:2177–204 (2006) https://doi.org/10.1098/rspa.2006.1679
13. Blondel VD, Tsitsiklis JN. A survey of computational complexity results in systems and control. *Automatica* 36:1249–74 (2000) https://doi.org/10.1016/S0005-1098(00)00050-9
14. Mädler J, Rahm J, Viedt I, Urbas L. A digital twin-concept for smart process equipment assemblies

supporting process validation in modular plants. *Computer Aided Chemical Engineering* 51:1435–40 (2022) https://doi.org/10.1016/B978-0-323-95879-0.50240-X

© 2025 by the authors. Licensed to PSEcommunity.org and PSE Press. This is an open access article under the creative commons CC-BY-SA licensing terms. Credit must be given to creator and adaptations must be shared under the same terms. See https://creativecommons.org/licenses/by-sa/4.0/