Modular Surrogate Models for Simulating the Amine Scrubbing Process

Ricardo Rendall*. Paris Papavasileiou, Preshit Dandekar***, Ivan Castillo****, You Peng*****

*The Dow Chemical Company, H.H. Dowweg 5, 4542 NM Hoek Netherlands (Tel: 0031-115-627-634; e-mail: rrendall1@ dow.com).
** Faculty of Science, Technology and Medicine, University of Luxembourg, Maison du Nombre, 6 Avenue de la Fonte, Esch-sur-Alzette, L-4364, Luxembourg (e-mail: paris.papavasileiou@uni.lu)
*** The Dow Chemical Company, Lake Jackson, USA, (e-mail: castillo2@dow.com)}
**** The Dow Chemical Company, Lake Jackson, USA, (e-mail: pdandekar@dow.com)}
***** The Dow Chemical Company,, Lake Jackson, USA, (e-mail: ypeng4@dow.com)}

Abstract: Surrogate models are becoming an important technique in process design, control, and optimization. These models are often developed at the process level, tailored to specific process configurations. Even when unit-level models are constructed, they rarely account for all units within a flowsheet. In this work, we use a unit-based surrogate model framework for an amine scrubbing process designed to remove CO_2, H_2S, and other sulfur species from gas streams. Unit-level Artificial Neural Network (ANN) surrogate models are created for each process unit. The developed unit-based models remain independent of process configuration and can be connected to recreate many possible arrangements of the amine scrubbing process. To demonstrate the versatility of our approach, we present two case studies. The first involves a typical amine scrubbing process configuration, while the second considers two absorber columns operating in parallel. This unit-based surrogate approach proves scalable and modular, enabling accurate predictions across diverse process configurations. By adopting this unit-based surrogate modeling framework, we can explore various process scenarios and simulate the amine scrubbing operations across multiple plant configurations.

Keywords: Surrogate Modeling, Machine Learning, Amine Scrubbing, Unit-based models.

1. INTRODUCTION

Process models are an important methodology in process development, optimization (Biegler, 2014), and control (Kothare, 2006). Through the years, first principles models have evolved through improved domain knowledge and can model the real processes with high accuracy. However, this higher accuracy often comes with increased complexity and computational costs that prevents their implementation in real-time applications. In such cases, data-driven surrogate models based on historical process data or simulations can help reduce the computational burden, while retaining satisfactory accuracy when predicting key process variables (KPIs) (McBride & Sundmacher, 2019). Surrogate models can assist in process design, helping to identify promising design regions and the main design trade-offs, but can also be used for real - time control and optimization. In this work, the main motivation for developing surrogate models is to provide real-time information about plant performance to Engineers and other personnel. This information will provide insights into the plant energy efficiency, production rates, profitability, as well as the ability to understand the impact of process changes/improvement activities.

Developing process surrogates using simulation data is challenging, as it requires many simulations for model training. Furthermore, the design space of these simulations must be carefully selected to account for the several possible process conditions and regimes. Surrogate models for an amine scrubbing process are usually developed at the process level (Chung & Lee, 2020; Li et al., 2015; Sipöcz et al., 2011), although there are examples in the literature of surrogate models at the unit level (Henao & Maravelias, 2011), and even on the thermodynamic/ property level (Goldstein et al., 2022; Misener & Biegler, 2023).

Process-level surrogates require simulations of process flowsheets containing all relevant units. Although the process fundamentals remain the same for all production plants, the process configuration might not be the same. In such cases, process-level surrogates must be trained specifically for the configuration of interest. On the other hand, developing unit-based surrogates requires simulations of flowsheets containing only the unit of interest. These flowsheets are less resource-intensive than full-process flowsheets. The usage of unit-based models as a way of replacing units in the flowsheet of a broader optimization framework has been documented in (Caballero & Grossmann, 2008; Quirante et al., 2015). This idea was further explored by (Lin et al., 2017), replacing almost all process

units with surrogate models with the goal of optimizing reactive extraction and reactive adsorption. Although unit-based surrogate models have many advantages, they also have drawbacks. For example, there is a risk for error propagation when the prediction of a model is the input to another model. In this case, a poor model can have a significant impact on the performance of other surrogate models as its predictions are used by subsequent models. Another drawback is that the overall complexity of the surrogate system is larger with unit-based models. When one develops a surrogate for a process, one can simply model the key outputs of interest. However, when the model is developed per process unit, many more outputs need to be modelled as those can be important inputs to the next process unit.

There are several examples of developing surrogate models for amine scrubbing processes in the literature, with most of them being at process-level (Chung & Lee, 2020; Hsiao & Chang, 2023a; Sipöcz et al., 2011). A few unit-based surrogate models have been proposed, for predicting column distribution coefficients as part of a wider optimization framework (Goldstein et al., 2022) or as a way of reducing the computational intensity of data acquisition when developing process-level surrogates (Hsiao & Chang, 2023b). In this work, we explore and demonstrate the modularity and scalability of unit-based models. We utilize unit-level Artificial Neural Network (ANN) models as building blocks, which are combined to recreate any process configuration on the amine scrubbing process, while also allowing for prediction of several intermediate material stream characteristics and process KPIs.

The remainder of this paper is structured as follows: Section 2 presents an overview of the process and the simulated dataset. The tools and strategies utilized for model development and connecting the unit-based models are presented in Section 3. Section 4 presents the main results for two different process configurations. Lastly, Section 5 contains the main conclusions of this work.

2. PROCESS DESCRITPION AND DATASETS

The sour gas amine scrubbing process is used for the removal of sulfur species and CO_2 from a sour gas stream using amine solvents, allowing it to meet product specifications. It consists of 3 main units: a) An absorber column, where CO_2 and sulfur species are removed from the sour gas stream, producing a treated gas stream (with very low concentrations of sulfur and CO_2) and a rich amine stream with high concentrations of sulfur and CO_2. b) a flash drum, where light hydrocarbons are removed from the rich amine stream and c) the regenerator column, where heat is used to remove large quantities of sulfur and CO_2, regenerating the amine stream. The regenerator column produces two streams, an acid gas stream, and a "lean" amine stream. The latter is recycled back to the absorber. A heat exchanger (HX) is also used between the columns for energy integration. A simplified process flowsheet of a typical configuration is presented in Figure 1. Nevertheless, there are cases where multiple similar units are used (i.e., 2 absorbers in parallel). One example of such a configuration is presented in Figure 2. These two process configurations will be used to demonstrate the effectiveness and scalability of the unit-based surrogate models proposed in this work.

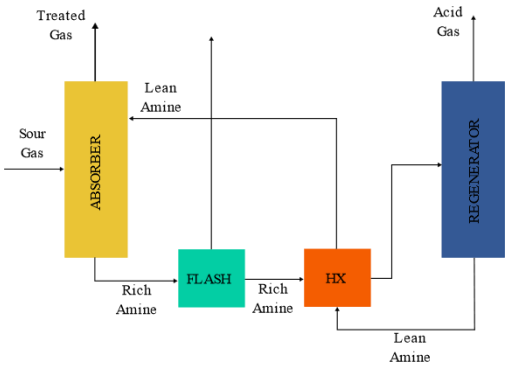

Fig. 1. Simplified flowsheet of the typical configuration of the sour gas amine scrubbing process.

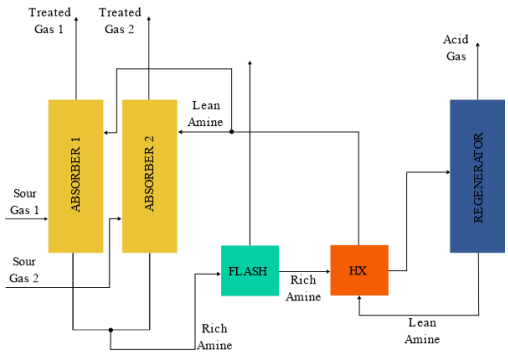

Fig. 2. Simplified flowsheet of the extended (2 absorber) configuration of the sour gas amine scrubbing process.

An in-house process simulation software is used for generating data for surrogate model training. Due to the modular scope of this work, each individual unit i.e., the absorber column, the flash drum and the regenerator column are simulated individually for a range of process inputs. A space filling Latin Hypercube Sampling was utilized to generate the input levels that were then simulated in the in-house simulator. An overview of the three distinct datasets collected for model development are presented in Table 1. This dataset is split into a training set containing 90% of the data for model development, and a test set with the remaining 10% to assess the model performance.

Table 1. Overview of the three datasets for model development.

Process unit	Number of response variables	Number of inputs	Number of simulations
Absorber	13	29	7600
Flash Drum	7	15	5886
Regenerator	4	20	2943

3. SURROGATE MODELING

3.1 Artificial Neural Networks

In this work, Artificial Neural Networks (ANNs) were utilized as the surrogate models to replace our detailed in-house simulator. In ANNs, neurons are organized into layers and the signal flows from the input layer towards the outputs layer. Each layer output is computed by passing the input signals though a non-linear function called the activation function. Additionally, every neuron has certain parameters (i.e., weights and biases) associated with it that are optimized or fitted based on available data during the model training phase.

ANNs have many hyper-parameters that need to be tuned for optimal performance. In this work, we use a Bayesian optimization approach (Snoek et al., 2012) implemented in the keras-tuner python library (O'Malley et al., 2019) to optimize the cross-validation error. Bayesian optimization is an efficient approach for hyper-parameter tunning since it uses a probabilistic model to sequentially identify the best next set of hyper-parameters. By comparison, randomly searching for the best hyper-parameter or testing the full set of parameter combinations can be computationally expensive. The main hyper-parameters tuned for the ANNs were the number of hidden layers, the number of neurons per layer, and the type of activation functions.

Using intuition and experts' knowledge regarding the amine scrubber process, the original input features were also combined to create new features that are expected to be predictive of the response variables (i.e., feature engineering). For example, the ratio of different types of amines in the lean amine stream (see Figure 1) will play a key role in the absorption process. Thus, we created new features containing all those ratios. The sour gas to amine flow ratio was also included in the model. Feature selection approaches were also tested to identify irrelevant predictors and remove them from the model. In this case, we used a tree-based method, Xgboost (Chen & Guestrin, 2016) , to rank features and remove those with an importance lower than 1%. The goal of this step was not to select important features but to remove those that are irrelevant. Thus, we used a conservative threshold and allow the ANN model the flexibility to use or not the remaining features. Additionally, variable transformations (e.g., logarithmic) were tested to assess their impact on model performance. The combinations of model architecture and transformations with the lowest root mean squared error (RMSE) were selected.

It is important to note that each output variable from each unit (see Table 1) is modelled independently. As an example, 13 ANNs models were built for the absorber, one per output variable. This simplifies the model development stage since developing one ANN to predict all 13 outputs would be challenging as they have different physical units and would need to be weighted accordingly.

3.2 Connecting process units into a flowsheet

After the developing the ANN models for each unit, they need to be connected to represent a process. This would be an easy task in the absence of the recycle streams (see Fig 1 and 2), and one could simply run the models of each unit sequentially. However, the recycle stream from the regenerator column to the absorber column turns this problem into a constrained optimization problem. To formulate and solve this problem, the Pyomo (Bynum et al., 2021) and OMLT (Ceccon et al., 2022) python packages are used.

Pyomo is an open-source software package developed in Python that provides a wide range of optimization capabilities. It allows users to formulate, solve, and analyse optimization models. Moreover, Pyomo also provides efficient auto-differentiation of nonlinear functions, which is an important advantage given the non-linear activation functions used by the developed ANN models. One other important feature of Pyomo is its block abstraction. This block component organizes constraints and variable groups. Many systems, like process flowsheets, have a hierarchical structure composed of repeated, conceptually related components. Pyomo lets modelers define and connect these fundamental building blocks in an object-oriented way.

OMLT (Ceccon et al., 2022) takes advantage of these block abstractions to encapsulate each surrogate model into the Pyomo framework so gradient-based optimization can be done conveniently with ANN models. OMLT encodes the structure and weights of the ANN models into a set of equality and inequality constraints, which can be readily used in the Pyomo optimization environment using different ANN formulations. The choice of formulation depends on the nature of the network activation functions. Nevertheless, several neural network formulations are available, both for smooth (Schweidtmann & Mitsos, 2019), and non-smooth (Yang et al., 2021) activation functions. In this work, the Reduced Space Formulation proposed by (Schweidtmann & Mitsos, 2019) is used for ANN models using smooth activation functions (i.e., sigmoid). This choice is expected to provide better performance than the Full Space Formulation. The formulation involving complementarity constraints presented in (Yang et al., 2021) is used for networks using the Relu activation function. This selection guarantees that the resulting optimization problem is non-linear, rather than a mixed-integer and non-linear optimization, which would be more difficult to solve. In summary, these formulations re-write and approximate the equations of an ANN, making them more tractable and easier to solve. The interested reader is referred to the literature for some formulation examples (Schweidtmann & Mitsos, 2019).

Connection of the inputs/outputs of units is achieved through constraints. Additionally, constraints are particularly useful when mixing streams, and one can impose a set of constraints reflecting the mass balance equations. The ability to mix streams like so is key to ensuring the scalability of this approach from a base case (1 absorber, 1 flash, 1 regenerator) to a larger number of units. Finally, the Ipopt (Wächter & Biegler, 2006) solver is used for the solution of the optimization problem. The problem simplifies to an optimization problem with zero degrees of freedom. The

solver, therefore, searches for a viable solution that satisfies all constraints.

One should note that integrating all the surrogate models will increase the chance for the individual models to extrapolate to conditions not observed in the training set, which can even lead to model infeasibilities. Thus, additional constraints are included to guarantee that the predicted output of each model is within the range observed in the training dataset. These constraints are included in Pyomo. Although these are not guarantees of no model extrapolation, they are a preliminary check on the validity of the results. Another test is to compare the predictions of the connected surrogate models to process simulations (containing all the units) for different scenarios, as will be presented in the results section. Still, further research is needed to automatically identify models that are not performing as expected and understand how their errors propagate through the surrogate system.

4. RESULTS

4.1 ANN model performance

Several ANN models were trained for each unit. Hyper-parameter tunning and different scaling approaches for the inputs and outputs were assessed during model training, and the best combinations were selected. The full list of model architectures and performance is not presented for simplicity, but model performances were generally high. All the coefficients of determination (R^2) are higher than 0.8, and most models had an R^2 higher than 0.95. The top three highest R^2 were above 0.99 and were related to flows, particularly the rich amine flow leaving the absorber. The smallest three R^2 were 0.8, 0.89, and 0.91, and are related to concentrations in different streams. Overall, the flash drum models showed the best performance, followed by the absorber column models. The regenerator models demonstrate the worse performance out of the three groups of models; however, their performance was still satisfactory. Figure 3 presents the test set parity plot for the H_2S in the rich amine stream after the absorber in scaled units. It should be noted that the higher the complexity of an ANN model, the higher the computational resources required both for training and prediction. Given the fact that these models will be evaluated simultaneously in the Pyomo-OMLT framework, where the flowsheet is solved iteratively, one should always consider the effect of ANN complexity on the full flowsheet case, even if, individually, these models have fast prediction times.

4.2 Performance of interconnected unit-based models

The converged optimization results for the typical process configuration are presented in Figure 4, where one can see that most models have a good performance with relative errors below 10%. Still, there is some spread of errors, which indicates that some models could be improved, potentially improving the overall surrogate system. This could be achieved by fine-tuning the current models with further hyper-parameter training, expanding the training dataset with additional simulations, etc. Still, the focus of this work falls not on only the individual model performance but on the predictive performance of the flowsheet containing modular units. It is the modularity of the units that allows recreating flowsheets for any configuration of an amine scrubbing process, as discussed below next.

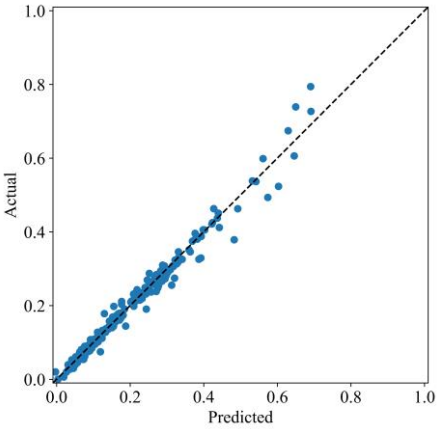

Fig. 3. Test set parity plot for H_2S in the rich amine stream leaving the absorber (scaled units).

The results for the process configuration with 2 parallel absorbers are presented in Figure 5. Apart from the input/output constraints necessary for the previous case, any configuration involving units in parallel also requires constraints that mix process streams. These constraints are based on mass balance equations, and they include all the outputs of parallel units that will become inputs to a downstream unit. Another difference between the previous case and the extended case is the increased computational resources required for the solution. This is a direct effect of the increased number of models and, thus, increased number of constraints and variables included in the optimization problem. Specifically, the optimization problem for configuration 1 (Figure 1) have 2729 variables and constraints whereas the problem for configuration 2 (Figure 2) contains of 5183 variables and constraints. In terms of computational effort, this configuration takes ~4min to run in the in-house simulator, whereas the surrogate models converge in ~40s. Besides the flexibility of the surrogates to adapt to multiple process configuration, they are also significantly faster at run time.

For this extended flowsheet, our implementation provides good predictions. Again, most relative errors are below 10%, and only one output is above 10% for absorber 2. It is interesting to note that absorbers 1 and 2 show a different profile for the relative errors, which is expected since they are simulated at different conditions. This example is a clear demonstration of the power of this modular approach. Its scalability can accommodate any process configuration, with the only downside being the increased computational cost as well as the size of the optimization problem that comes with the increased complexity. Although it is possible that an extreme number of units (e.g., 5 absorbers, 5 flash, and 5 regenerator units) could make the problem computationally intractable, the existence of such real-world process configurations is highly improbable. This method is generally

scalable and efficient, offering reliable estimates for different amine scrubbing process KPIs.

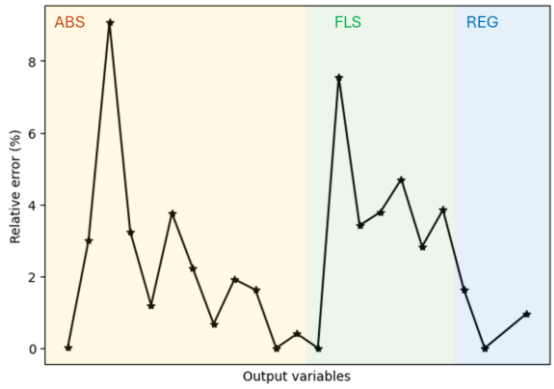

Fig. 4. Relative error for the test case with the following configuration: 1 absorber, 1 flash, and 1 regenerator.

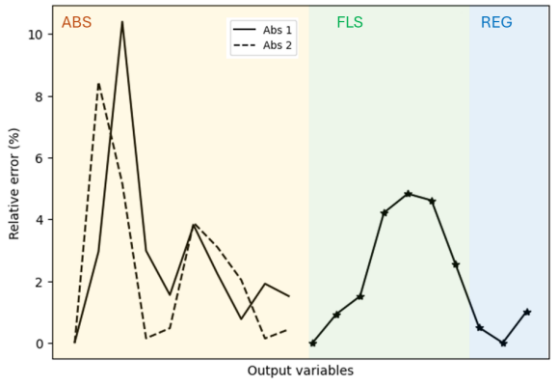

Fig. 5. Relative error for the test case with the following configuration: 2 absorbers, 1 flash, and 1 regenerator.

5. CONCLUSIONS

This work presented the development of a modular surrogate modelling approach for an amine scrubbing process. This modular approach involves unit-based ANN models trained with independent simulations for each unit. In other words, the absorber column models were trained using simulations containing only an absorber column unit. The same stands for the flash drum and regenerator column models. These ANN models can then be connected in the Pyomo/OMLT framework, allowing the reconstruction of any amine scrubbing process configuration regardless of the number of units used.

The modular approach is demonstrated on two different process configurations. One consists of one absorber column, one flash drum, and one regenerator column and another configuration containing two absorber units in parallel, one flash, and one regenerator unit. In contrast to the first case, the second case involves additional constraints to reflect the mass balance equations used for mixing the outlet streams from the two absorbers. The second case also comes with a higher computational cost due to the increased number of ANN models present in the flowsheet. In both cases, the modular surrogate modelling approach demonstrates good predictive performance, which can still be improved with the amelioration of the corresponding individual unit-based models.

The approach is strongly scalable and is expected to perform well in flowsheets containing more units. In the case of many units, attention is required when creating the mass balance constraints for the mixing or splitting of material streams. Furthermore, flowsheets involving high numbers of units come with a higher complexity due to the higher number of individual models involved. A higher number of individual models translates to more variables and constraints, resulting in a resource-intensive problem.

To conclude, the proposed modular surrogate approach could be very advantageous for processes involving different types of units and especially when the process configuration differs from site to site. The approach's scalability and modularity are its main strengths and can pave the way for other similar applications.

As future work, we intend to test the robustness of the unit-based surrogate system for other process configurations to have a more thorough understanding of its performance. Furthermore, individual surrogate models can be further improved. To achieve this, many approaches can be tested such as adaptive sampling, further hyper-parameter optimization, testing other methods besides ANNs, among others. These have the potential of improving the individual model performance, with an expected gain to the surrogate model system.

REFERENCES

Biegler, L. T. (2014). Recent Advances in Chemical Process Optimization. *Chemie Ingenieur Technik*, *86*(7), 943–952. https://doi.org/10.1002/cite.201400033

Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., & Woodruff, D. L. (2021). *Pyomo--optimization modeling in python* (Third, Vol. 67). Springer Science & Business Media.

Caballero, J. A., & Grossmann, I. E. (2008). An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE Journal*, *54*(10), 2633–2650. https://doi.org/10.1002/aic.11579

Ceccon, F., Jalving, J., Haddad, J., Thebelt, A., Tsay, C., Laird, C. D., & Misener, R. (2022). OMLT: Optimization & Machine Learning Toolkit. *ArXiv*.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system*. 785–794.

Chung, W., & Lee, J. H. (2020). Input–Output Surrogate Models for Efficient Economic Evaluation of Amine Scrubbing CO2 Capture Processes. *Industrial & Engineering Chemistry Research*, *59*(42), 18951–18964. https://doi.org/10.1021/acs.iecr.0c02971

Goldstein, D., Heyer, M., Jakobs, D., Schultz, E. S., & Biegler, L. T. (2022). Multilevel surrogate modeling of an amine scrubbing process for COcapture. *AIChE Journal*, *68*(6), e17705. https://doi.org/10.1002/aic.17705

Henao, C. A., & Maravelias, C. T. (2011). Surrogate-based superstructure optimization framework. *AIChE Journal*, *57*(5), 1216–1232. https://doi.org/10.1002/aic.12341

Hsiao, Y.-D., & Chang, C.-T. (2023a). *Expandable neural networks for efficient modeling of various amine scrubbing configurations for CO2 capture*. *281*, 119191. https://doi.org/10.1016/j.ces.2023.119191

Hsiao, Y.-D., & Chang, C.-T. (2023b). *Progressive learning for surrogate modeling of amine scrubbing CO2 capture processes*. *194*, 653–665. https://doi.org/10.1016/j.cherd.2023.05.016

Kothare, M. V. (2006). Process Control: Modeling, Design and Simulation [Book Review]. *IEEE Control Systems Magazine*, *26*(6), 106–108. https://doi.org/10.1109/mcs.2006.252814

Li, F., Zhang, J., Oko, E., & Wang, M. (2015). Modelling of a post-combustion CO2 capture process using neural networks. *The 10th European Conference on Coal Research and Its Applications*, *151*, 156–163. https://doi.org/10.1016/j.fuel.2015.02.038

Lin, Z., Wang, J., Nikolakis, V., & Ierapetritou, M. (2017). Process flowsheet optimization of chemicals production from biomass derived glucose solutions. *Sustainability & Energy Systems*, *102*, 258–267. https://doi.org/10.1016/j.compchemeng.2016.09.012

McBride, K., & Sundmacher, K. (2019). Overview of Surrogate Modeling in Chemical Process Engineering. *Chemie Ingenieur Technik*, *91*(3), 228–239. https://doi.org/10.1002/cite.201800091

Misener, R., & Biegler, L. (2023). *Formulating data-driven surrogate models for process optimization*. *179*, 108411. https://doi.org/10.1016/j.compchemeng.2023.108411

O'Malley, T., Bursztein, E., Long, J., Chollet, F., H.Jin, & Invernizzi, L. (2019). *KerasTuner*. https://github.com/keras-team/keras-tuner

Quirante, N., Javaloyes, J., & Caballero, J. A. (2015). Rigorous design of distillation columns using surrogate models based on Kriging interpolation. *AIChE Journal*, *61*(7), 2169–2187. https://doi.org/10.1002/aic.14798

Schweidtmann, A. M., & Mitsos, A. (2019). *Deterministic Global Optimization with Artificial Neural Networks Embedded*. *180*(3), 925–948. https://doi.org/10.1007/s10957-018-1396-0

Sipöcz, N., Tobiesen, F. A., & Assadi, M. (2011). *The use of Artificial Neural Network models for CO2 capture plants*. *88*(7), 2368–2376. https://doi.org/10.1016/j.apenergy.2011.01.013

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *ArXiv*. https://doi.org/10.48550/arxiv.1206.2944

Wächter, A., & Biegler, L. T. (2006). *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*. *106*(1), 25–57. https://doi.org/10.1007/s10107-004-0559-y

Yang, D., Balaprakash, P., & Leyffer, S. (2021). Modeling Design and Control Problems Involving Neural Network Surrogates. *ArXiv*.