Surrogate modeling of dynamical systems: deep learning or model-order reduction?

Luisa Peterson[2][0009−0006−0776−1517], Edgar Ivan Sanchez Medina[1][0000−0001−5413−2795], Md Meraj Khalid[2][0009−0003−0822−1836], and Kai Sundmacher[1,2][0000−0003−3251−0593]

[1] Chair for Process Systems Engineering, Otto-von-Guericke University,
Universitätsplatz 2, Magdeburg, 39106, Germany

[2] Process Systems Engineering, Max Planck Institute for Dynamics of Complex
Technical Systems, Sandtorstraße 1, Magdeburg, 39106, Germany

sundmacher@mpi-magdeburg.mpg.de

1 Motivation

Dynamical systems are found in many areas of science and engineering, ranging from climate systems to economic, chemical, and biomedical systems. The evolution of the properties of such systems often occurs not only over time, but also across spatial dimensions. As a result, dynamical systems are often abstracted as a set of partial differential equations (PDEs). Due to the high nonlinearity and coupled nature of chemical systems, the resulting PDEs often cannot be solved analytically. Therefore, the use of numerical strategies that discretize the system in space, such as finite difference or finite volume methods, is often the solution of choice. However, numerical methods also encounter difficulties when instabilities are present, or when the computational budget is limited for the desired decision process that one envisions with the dynamic model.

Focusing on these two challenges, i.e., time efficiency and numerical stability, several techniques have been proposed in the literature for modeling dynamic systems in a data-driven manner. Two types of methods stand out: deep learning-based and model-order reduction techniques. The use of these alternative modeling paradigms to tackle the complexity of dynamical systems appears to be promising in terms of time efficiency and numerical stability. Time-efficiency of data-driven approaches tends to be higher than traditional approaches not only in the phase of model development, but also in the phase of model use. And depending on the specific data-driven method, numerical stability can even be guaranteed.

In this work [4], we compare an example of each of these two methods: graph neural networks (representing the deep learning paradigm) and operator inference (representing the model order reduction strategy). The case study presented here consist in the dynamic simulation of a CO_2 methanation reactor, which is relevant in the context of renewable energy storage and power-to-x processes. We first introduce the overall problem, then we briefly introduce the graph neural network and operator inference approaches, and finally present the comparison results and the conclusions.

2 Case-study: CO_2 methanation reactor

The CO_2 methanation process converts carbon dioxide and hydrogen into methane and water. The core of this process typically takes place in fixed-bed reactors, whose operation must dynamically adapt to changing hydrogen loads resulting from the dynamic profiles of renewable energy used in the upstream electrolysis phase [1]. Since this reaction is highly exothermic, managing the heat profile of the reactor is key to avoiding potential hotspots that could damage the expensive catalyst inside and reduce the overall reactor efficiency.

In this work, we modeled the reactor behavior as one-dimensional and polytropic, assuming that a single reaction is happening and that axial dispersion effects can be neglected [7]. The resulting set of nonlinear and coupled PDEs can be written as:

$$\varepsilon_{\mathrm{R}} \frac{\partial X}{\partial t} = -u \frac{\partial X}{\partial z} + \frac{M_{\mathrm{CO_2}}}{\rho y_{\mathrm{CO_2,in}}} (1-\varepsilon_{\mathrm{R}}) \sigma_{\mathrm{eff}}, \tag{1}$$

$$\begin{aligned} (\rho c_{\mathrm{p}})_{\mathrm{eff}} \frac{\partial T}{\partial t} =& - u_{\mathrm{in}} \rho_{\mathrm{in}} c_{\mathrm{p}} \frac{\partial T}{\partial z} + \frac{\partial}{\partial z}\left[\Lambda_{\mathrm{ax}} \frac{\partial T}{\partial z}\right] - \frac{4U}{D}(T - T_{\mathrm{cool}}) \\ & - \Delta H_{\mathrm{R}} (1-\varepsilon_{\mathrm{R}}) \sigma_{\mathrm{eff}}, \end{aligned} \tag{2}$$

where, X refers to the CO_2 conversion and T refers to the temperature. Constants ε_{R}, $M_{\mathrm{CO_2}}$, $y_{\mathrm{CO_2,in}}$, u_{in}, ρ_{in}, D and L define the specific settings of the reactor. And the variables u, ρ, σ_{eff}, c_{p}, Λ_{ax}, U, T_{cool} and ΔH_{R} are determined via algebraic expressions. Precise meaning of these constants is provided in the Appendix. The model described by the equations 1 and 2 serves here as a simulated black-box model from which the dynamic data are retrieved for the posterior development of the data-driven models.

3 Graph neural networks

For the graph neural network approach, the dynamic problem is represented using two types of graphs. First, each time snapshot is represented as a chain-like graph, where nodes represent the specific spatial point along the axial axis of the reactor and (bidirected) edges represent the connection between contiguous points in space within the reactor. The nodes are attributed with the corresponding CO_2 conversion and temperature values. This graph is here denoted as the *snapshot graph.* Second, consecutive time snapshots are represented as a matrix-like graph, where multiple snapshot graphs are connected using directed edges connecting the same spatial point (node) from the past to the future. This graph is denoted as the *window graph.*

The task of the graph neural network is to predict the snapshot graph of the next time-step using the previous window graph. In this way, the prediction of the dynamic system is carried out in a roll-out fashion updating the window graph with the newly predicted snapshot graph while maintaining the same window size (i.e., number of observed time-steps in the past to perform the future prediction).

4 Operator inference

Starting from matrix $\mathbf{U} \in \mathbb{R}^{n\times(k+1)}$ containing the dynamic evolution of a reactor state (i.e., CO_2 conversion or temperature) across n spatial points and $(k+1)$ time-steps, the operator inference approach [3] starts by constructing a projection matrix $\mathbf{V} \in \mathbb{R}^{n\times r}$. The projection matrix is obtained by selecting the first dominant r basis vectors from a proper orthogonal decomposition. In this way, the data is effectively reduced, and the following optimization problem, which only involves an ordinary differential equation (ODE) system, is solved:

$$\min_{\hat{\mathbf{A}},\hat{\mathbf{H}},\hat{\mathbf{B}}} \left\| \dot{\hat{\mathbf{U}}} - \hat{\mathbf{A}}\hat{\mathbf{U}} - \hat{\mathbf{H}}\left(\hat{\mathbf{U}} \otimes \hat{\mathbf{U}} - \hat{\mathbf{B}}\right) \right\| + \alpha\mathcal{R}(\hat{\mathbf{A}},\hat{\mathbf{H}}), \tag{3}$$

where $\hat{\mathbf{A}}, \hat{\mathbf{H}}$ and $\hat{\mathbf{B}}$ are the operators to be inferred in the reduced space, and $\hat{\mathbf{U}}$ refers to the compressed dynamic trajectory obtained from $\mathbf{V}^\top\mathbf{U}$. The symbol $\otimes$ stands for the Kronecker product and the upper symbol $(\cdot)$ refers to the time-derivative. The $\mathcal{R}$ term refers to a regularization term to avoid ill-conditioning [6]. The enforced parametrization on the linear and quadratic operators results in guaranteed numerical stability according to [2].

Once the operators in the reduced-space are inferred, they can be used to reconstruct an approximation of the original black-box dynamical system in the original coordinate system via

$$\dot{\hat{\mathbf{u}}} = \hat{\mathbf{A}}\hat{\mathbf{u}} + \hat{\mathbf{H}}\left(\hat{\mathbf{u}} \otimes \hat{\mathbf{u}}\right) + \hat{\mathbf{B}} \tag{4}$$

5 Results

For testing the performance of both methods, the following simulation was carried out. The reactor model (i.e., equations 1 and 2) was solved considering a time-span of 35 seconds using a discretization of 1750 steps both in the spatial and the temporal dimensions. The simulated scenario consist in a load change from a relatively high flow rate of hydrogen to half the original flow rate. The first 20 seconds of the trajectory are used for model development, while the last 15 seconds are reserved for testing the approaches. This scenario helps on testing how the different approaches can learn the future behavior of the dynamical system from historical data, a situation often encountered in practice.

Table 1 shows a comparison of the predictive performance of both methods according to the relative Frobenius norm of the error matrix obtained by each of the methods. The graph neural network model achieves a higher error compared to the order-reduction technique. This also comes along the fact that the graph neural network model as a learned simulator of the dynamic process might be difficult to interpret compared to the reduced dynamic system obtained by operator inference.

The errors of the graph neural network tend to be more pronounced at points where the gradients of the state trajectories are large. In this specific case, this occurs close to the inlet of the reactor at the initial time-steps. This precise zone

Table 1. Relative Frobenius norm for CO_2 conversion X and temperature T, achieved by the graph neural network (GNN) and the operator inference (OpInf) starategies.

Method	Frobenius norm for X	Frobenius norm for T
GNN (train)	5.34 %	16.97 %
OpInf (train)	0.052%	0.082%
GNN (test)	4.24 %	9.28 %
OpInf (test)	0.050%	0.58%

in the reactor corresponds to the area where the hotspot starts to appear. Since the error occurs early on in the time dimension, the error is propagated in time altering the relative accuracy of the predictions. This exemplifies the difficulty of applying deep-learning methods to capture physical phenomena enclosed in just a minor proportion of the data available. The inclusion of a physics-based prior during training can also be envisioned here. The operation of the graph neural network model could also be carried out in the reduced space obtained by the operator inference method, which might improve the overall accuracy of the graph neural network. Moreover, the evaluation time of the graph neural network model is significantly smaller compared to the evaluation of the original PDE system.

In comparison, operator inference achieves a significantly lower error on the state trajectories prediction. The resulting set of ordinary-differential equations is also comparatively cheaper to evaluate compared to the solution of the black-box model. This attribute can be exploited not only in the context of data-driven modeling, but also in the context of surrogate generation. An additional advantage of the operator inference approach is the stability of the solutions that can be guaranteed [2].

6 Conclusion

While deep-learning methods are powerful approaches to approximate dynamical systems, they also come with certain limitations. Specially, in the context of modeling chemical dynamic systems (where normal operation data abound compared to out-of-normal operation data), deep-learning techniques should still be improved to weight the amount of data with respect to the amount of information that it provides. This is exactly an area where reduced-order techniques excel. As exemplified by the presented CO_2 reactor methanation problem, the issue of information extraction is highly relevant to achieve accurate predictions that could be potentially safely used in real-world scenarios. Despite this, in terms of the improvement in computational time compared to solving the original system of PDEs, both methods present a promising approach to tackle the modeling of complex dynamical systems.

7 Appendix

Equations 1 and 2 include the following constants: ε_{R} refers to the packed-bed reactor void fraction, $M_{\mathrm{CO_2}}$ to the molar mass of carbon dioxide, $y_{\mathrm{CO_2,in}}$ to the inlet mass fraction of carbon dioxide, u_{in} and ρ_{in} to the superficial gas velocity and gas mixture density at the inlet, respectively. D refers to the tube diameter, L to the reactor length, z to the axial and t to the time coordinate. In addition, there are non-constant parameters that are described by algebraic equations. These include the surperficial gas velocity u, the gas mixture density ρ, the effective reaction rate σ_{eff}, the heat capacity c_{p}, the axial heat conductivity Λ_{ax}, the heat transfer coefficient U, the cooling temperature T_{cool} and the enthalpy of reaction ΔH_{R}.

The specific GNN architecture used in this work corresponds to the GAT model, which incorporates an attention mechanism into the traditional message-passing scheme [5]. Thus, each GAT message-passing layer can be written as

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)\cup i} \alpha_{i,j}^{(l)}\theta^{(l)}\mathbf{h}_j^{(l)}\right), \tag{5}$$

where σ is a nonlinear function, $\theta^{(l)}$ is a learnable matrix at layer l, and the attention weight $\alpha_{i,j}^{(l)}$ measures the connection strength between the node i and its neighbor j.

References

1. Bremer, J., Rätze, K.H., Sundmacher, K.: $\mathrm{CO_2}$ methanation: Optimal start-up control of a fixed-bed reactor for power-to-gas applications. AIChE **63**(1), 23–31 (2017)
2. Goyal, P., Duff, I.P., Benner, P.: Guaranteed Stable Quadratic Models and their applications in SINDy and Operator Inference. 10.48550/arXiv.2308.13819 (2023)
3. Peherstorfer, B., Willcox, K.: Data-driven operator inference for non-intrusive projection-based model reduction **306**, 196–215 (2016). https://doi.org/https://doi.org/10.1016/j.cma.2016.03.025
4. Peterson, L., Forootani, A., Sanchez Medina, E.I., Gosea, I.V., Sundmacher, K., Benner, P.: Comparison of data-driven approaches for simulating the dynamic operation of a $\mathrm{CO_2}$ methanation reactor. Submitted to IEEE Transaction on Automation Science and Engineering (2024)
5. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. 10.48550/arXiv.1710.10903 (2017)
6. Yıldız, S., Goyal, P., Benner, P., Karasözen, B.: Learning reduced-order dynamics for parametrized shallow water equations from data. Int. J. Num. Methods Fluids **93**(8), 2803–2821 (2021)
7. Zimmermann, R.T., Bremer, J., Sundmacher, K.: Multi-Period Design Optimization of Flexible Fixed-Bed Reactors by Stoichiometry-Based Model Reduction. Computer Aided Chemical Engineering **50**, 947–952 (2021)