Delft University of Technology

Physics-informed neural networks and time-series transformer for modeling of chemical reactors

Lastrucci, Giacomo; Theisen, Maximilian F.; Schweidtmann, Artur M.

DOI
10.1016/B978-0-443-28824-1.50096-X

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the 34th European Symposium on Computer Aided Process Engineering

Citation (APA)
Lastrucci, G., Theisen, M. F., & Schweidtmann, A. M. (2024). Physics-informed neural networks and time-series transformer for modeling of chemical reactors. In F. Manenti, & G. V. Reklaitis (Eds.), *Proceedings of the 34th European Symposium on Computer Aided Process Engineering: 15th International Symposium on Process Systems Engineering (ESCAPE34/PSE24)* (pp. 571-576). (Computer Aided Chemical Engineering; Vol. 53). Elsevier. https://doi.org/10.1016/B978-0-443-28824-1.50096-X

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher
is the copyright holder of this work and the author uses the
Dutch legislation to make this work public.

Flavio Manenti, Gintaras V. Reklaitis (Eds.), Proceedings of the 34th European Symposium on
Computer Aided Process Engineering / 15th International Symposium on Process Systems
Engineering (ESCAPE34/PSE24), June 2-6, 2024, Florence, Italy
© 2024 Elsevier B.V. All rights reserved. http://dx.doi.org/10.1016/B978-0-443-28824-1.50096-X

Physics-informed neural networks and time-series transformer for modeling of chemical reactors

Giacomo Lastrucci[a], Maximilian F. Theisen[a], and Artur M. Schweidtmann[a,*]

[a] *Process Intelligence Research Group, Department of Chemical Engineering, Delft University of Technology, Van der Maasweg 9, Delft 2629 HZ, The Netherlands*
[*] *Corresponding author. Email: a.schweidtmann@tudelft.nl*

Abstract

Multiscale modeling of catalytical chemical reactors typically results in solving a system of partial differential equations (PDEs) or ordinary differential equations (ODEs). Despite significant progress, the numerical solution of such PDE or ODE systems is still a computational bottleneck. In the past, deep learning techniques have gained attention for developing surrogate models in chemical engineering. Also, hybrid models and physics-informed neural networks (PINNs) have been developed to integrate physical knowledge and data-driven approaches. However, it is often unclear how such modeling approaches compare for specific case studies. In this study, we investigate and compare state-of-the-art surrogate and hybrid models for the spatial evolution of the state variables in a packet-bed reactor for methanol production. Firstly, we develop a tailored hybrid model based on PINNs, thereby seamlessly integrating physical knowledge and data. Secondly, we investigate a recently-developed time-series transformer model to learn the spatial evolution of the state variables. As a benchmark model, we train a traditional multilayer perceptron (MLP) and compare the models to a standard numerical integration technique. We achieve orders of magnitude in speedup using MLPs and PINNs when compared to classical ODE solvers, while maintaining high levels of accuracy in modeling the underlying system.

Keywords: Reactor modeling, hybrid modeling, physics-informed machine learning, time-series transformer

1. Introduction

Multiscale modeling of catalytical reactors is a well-known discipline and commonly results in partial differential equation (PDE) or ordinary differential equation (ODE) systems describing the spatial and temporal evolution of the state variables involved (e.g., temperature, pressure, and composition). Despite significant progress, the numerical solution of such PDE or ODE systems is still a computational bottleneck when integrated into large, plant-wide process simulation and optimization studies.

The selection of suitable surrogate modeling approaches for chemical reactor systems is complex and remains to be decided circumstantially. Deep learning techniques have recently gained attention in their function as surrogate models in chemical engineering. The most popular deep learning technique, multilayer perceptrons (MLPs), consists of layers of nonlinear transformations that act as universal approximators. MLPs have been successfully applied to model many reactor systems. As an alternative, foundation models have recently yielded promising results in fields such as computer vision or natural language processing (Kolides, et al., 2023). In the context of chemical engineering, time-series transformers (TSTs) specifically allow for modeling of complex temporal dynamics and relationships within sequential data, enhancing the accuracy and efficiency of predictions. They have been applied to crystallization systems before (Sitapure and Kwon, 2023). However, the black-box nature of such deep learning models poses issues when applied outside training boundaries or when insufficient data is available (Schweidtmann, et al., 2021). Moreover, despite sufficient overall accuracy, physical consistency may not be ensured. Considering those challenges, hybrid models (von Stosch, et al., 2014) and physics-informed neural networks (PINNs) deep learning architectures (Raissi, et al., 2019) have been proposed to integrate physical knowledge and data-driven approaches. While many promising data-driven model architectures exist, the comparison and selection of suitable hybrid model approaches for chemical reactor surrogation is complex and remains an open research question.

In this study, we investigate and compare three state-of-the-art surrogate and hybrid models for describing the spatial evolution of the state variables in a packet-bed reactor for methanol production (Vanden Bussche and Froment, 1996). (1) We develop and train a tailored hybrid model based on PINNs, seamlessly integrating physical knowledge and data. We employ PINNs, commonly used in scientific machine learning, for solving Partial Differential Equations (PDEs) under a specific initial and boundary condition. We expand the PINN framework to enable predictions for a broad range of initial conditions. (2) We deploy a TST to model the sequential state evolution along the reactor length. (3) As a baseline surrogate model, we train a traditional multilayer perceptron. Finally, we systematically compare the developed surrogate and hybrid models with standard ODE solvers in terms of accuracy and runtime for potential application in complex plant-wide simulation and optimization.

2. Methodology

2.1. Physics-informed neural networks

Physics-informed neural networks are deep learning models aimed to bridge the gap between data-driven machine learning models and rigorous scientific computing (Lawal, et al., 2022). PINNs are neural networks trained in a supervised manner, while also minimizing errors on a given equation, e.g., physical law. They allow to approximate the solution of ODE and PDE systems, without performing numerical integration, thus potentially speeding up the computation.

We extend the original PINNs framework (Raissi et al., 2019) by considering the solution of an ODE system across a variety of initial conditions, specifically targeting relevant operating inlet variables. The original PINN formulation was developed to address forward PDE problems provided with fixed initial and boundary conditions. Our approach involves leveraging existing data to gain insights, while ensuring physical accuracy and interpretability through the enforcement of fundamental physical laws, such as material, energy, and momentum balances. Given an autonomous initial value problem (IVP):

$$\frac{d\boldsymbol{s}}{d\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{s}), \;\; \boldsymbol{s}(\boldsymbol{z} = \boldsymbol{0}) = \boldsymbol{s_0} \tag{1}$$

Based on (Eq. 1), a *residual* function $\boldsymbol{r}(z, \boldsymbol{s}) \coloneqq \frac{ds}{dz} - \boldsymbol{f}(\boldsymbol{s})$, is defined. An MLP can be trained to predict the solution $\boldsymbol{s}$ while also considering physical laws by optimizing the following multi-task loss function $L = L_{SD} + \sum_k L_{R_k}$. The first term ($L_{SD}$), known as -*sensors data loss*, represents the data-driven contribution, computed as the Mean Squared Error (MSE) between the available ground-truth in the domain and the model prediction. The second term ($\sum_k L_{R_k}$), known as *residual loss*, measures the physical discrepancy represented by the difference (residual) between the right-hand side $\boldsymbol{f}$ of the ODE system, and the spatial derivative of the state variables, computed through automatic differentiation:

$$\boldsymbol{L_{R_k}} = \frac{\boldsymbol{1}}{\boldsymbol{N_C}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N_C}} \boldsymbol{r_{ik}}^{\boldsymbol{2}} = \frac{\boldsymbol{1}}{\boldsymbol{N_C}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N_C}}\left(\left.\frac{\boldsymbol{d\hat{s}_k}}{\boldsymbol{dz}}\right|_{\boldsymbol{z_i}} - \boldsymbol{\hat{f}_k}(\boldsymbol{z_i}, \boldsymbol{s_i^0}, \boldsymbol{\theta})\right)^{\boldsymbol{2}} \tag{2}$$

where N_C is the number of *collocation points*, which can be arbitrary distributed in the domain of interest, and k is the kth equation in the IVP (Eq. 1). Note that the residual loss is entirely dependent on model prediction, parameters, and initial conditions, ensuring that spatial derivative and predictions comply with given differential equations, i.e., the physical laws. Beyond physical explainability, PINNs potentially facilitate extrapolation and generalization outside training boundaries, which can be achieved by placing collocation points where training data are unavailable.

2.2. Tailored training procedure for physics-informed neural networks

The network parameters are optimized with respect to the multitask loss through a gradient descent algorithm, but the PINN training is often challenging. As observed by S. Wang et al. (2020) the multiscale nature of both variables involved and loss terms can lead to preferential optimization, favoring the minimization of certain terms at expense of others. Also, it has been demonstrated that imbalanced contributions can lead to a "*stiff*" loss surface characterized by large eigenvalues of the hessian, ultimately resulting in gradient descent failure and training divergence. This has also been observed in our study.

To improve the PINN training, we implement a series of mitigation actions following the recommendations by Wang et al. (2023). Specifically, *(i)* we perform system nondimensionalization and *(ii)* we implement an adaptive loss balancing algorithm. The scaling is a best-practice for solving stability issues in gradient-based optimizers. Balancing the loss terms is crucial to avoid task dominance and "stiff" loss surface and several solutions have been proposed in the literature (McClenny and Brada-Neto, 2022). We develop a simple adaptive loss balancing routine based on losses magnitude, by defining the following weights:

$$\boldsymbol{\lambda^*_{SD}} = \frac{\boldsymbol{L^t_{SD}} + \sum_{\boldsymbol{k}} \boldsymbol{L^t_{R_k}}}{\boldsymbol{L_{SD}}} \tag{3}$$

$$\boldsymbol{\lambda^*_{R_k}} = \frac{\boldsymbol{L^t_{SD}} + \sum_{\boldsymbol{k}} \boldsymbol{L^t_{R_k}}}{\boldsymbol{L^t_{R_k}}} \qquad \boldsymbol{k = 1, \dots, N_{eq}} \tag{4}$$

We update the loss weights every j iterations, through a moving average $\boldsymbol{\lambda}^{t+1} = \alpha\boldsymbol{\lambda}^t + (1-\alpha)\boldsymbol{\lambda}^*$, where, j and α are tunable hyperparameters. We initialize the weights to be equal to 1 in the first iteration. Finally, we define the weighted loss function as $L = \lambda_{SD}L_{SD} + \boldsymbol{\lambda}_{\boldsymbol{R}}^T\boldsymbol{L}_{\boldsymbol{R}}$.

To comply with the physical system (i.e., all the variables must be greater than zero), we add the term $||\max(\boldsymbol{0}, -\boldsymbol{s})||_1$ to the physical loss function to penalize negative predictions. Moreover, the residual loss is excluded in case output variables are below zero to avoid numerical overflow due to physical inconsistency.

For the PINNs model, we uniformly distributed the collocation points along the reactor length to enforce physical consistency. In order to use mini-batch optimization, the number of collocation points should coincide with the batch size, which is set to 200. We trained the three model on the same generated *sensor* dataset (Sect. 3) using Adam optimizer and a learning rate of 10^{-5}.

2.3. Time-series transformers

Time-series transformers, a subset of transformers, have recently been developed to handle multivariate problems along a temporal dimension (Wen, et al., 2022).

Transformers are neural networks designed for autoregressive sequence prediction based on inputs and past outputs. They operate by embedding and processing initial input through an encoder. A decoder then uses this encoded information to predict outcomes one time step at a time. In TSTs, each prediction step corresponds to a system state, represented by a single vector for each timestep. We have adapted these transformers for spatial sequence modeling, where each step corresponds to a discretized spatial unit instead.

The foundation of transformer architectures is attention (Vaswani, et al., 2017). Attention measures the relatedness of tokens to each other in a computationally efficient way by calculating a dot product based similarity of the mapped inputs. For a detailed explanation of attention and its application to TSTs, we refer to Wen, et al. (2022).

3. Case study

3.1. Case study description

We investigate and compare the methodologies by modeling a tubular packet-bed reactor for methanol synthesis. For simplicity, we assume pseudo-homogeneous conditions and plug flow hydrodynamics with negligible axial dispersion. The model consists of an ODE system representing the material (Eq. 5), energy (Eq. 6) and momentum balance (Eq. 7) along the reactor axial coordinate.

$$\frac{d\dot{n}_i}{dz} = A\rho_C(1-\epsilon)\sum_{j}^{NR}\nu_{ij}r_j \qquad i = CO, CO_2, CH_3OH, H_2, H_2O, CH_4, N_2 \tag{5}$$

$$\frac{dT}{dz} = \frac{A}{\dot{m}_{tot}\tilde{c}_{P,mix}}\left(US_V(T_C - T) - \sum_{j}^{NR}\Delta H^0_{R_j} r_j \rho_C(1-\epsilon)\right) \tag{6}$$

$$\frac{dP}{dz} = -\left(150\frac{(1-\epsilon)^2}{d_{eq}^2\epsilon^3}\mu u_s + 1.75\frac{1-\epsilon}{d_{eq}\epsilon^3}\rho u_s^2\right) \tag{7}$$

In Eq.5, $\dot{n}$ is the molar flowrate, A is the cross sectional area of the tube, ρ_C is the catalyst density, ϵ is the bed void fraction, $\nu_{ij}r_j$ indicate the product between the reaction rate of the reaction *j* and the correspondent stoichiometric coefficient of species *i*. In the energy balance (Eq.6), $\dot{m}_{tot}$ is the constant total mass flowrate, S_V is the exchange area *per* unit of volume and T_C is the temperature of the coolant medium.

The reactive system comprises 7 species and 2 reactions, kinetically modelled by Vanden Bussche and Froment (1996). The reactor is cooled through boiling water at 38 bar and Antoine equation is considered. Specific heat capacity ($\tilde{c}_{P,mix}$) , reaction enthalpy (ΔH_R^0), overall heat transfer coefficient (U), void fraction (ϵ), viscosity (μ) are assumed to be constant along the reactor. The pressure drop follows the Ergun relation (Eq.7), where d_{eq} is the equivalent diameter of the cylindrical catalyst pellet, ρ is the gas density and u_s is the superficial velocity.

3.2. Data generation

We generate the training dataset by uniformly and randomly perturbating a main set of operating conditions (Table 1) and solving the system accordingly. The training dataset is composed of 5,000 distinct initial condition combinations, each varying within a +/-20% range from a central set of industrial relevant conditions. Each sample describes the spatial profile of the 9 state variables, discretized in 800 points along the domain. We solve the system in Python, using `scipy.integrate.solve_ivp` function, setting the absolute and relative tolerance to 10^{-12} and 10^{-8}, respectively. The dataset, comprising 4 million data points, is used to train the models, enabling them to forecast the values of state variables at a specific longitudinal coordinate based on a given set of initial conditions.

Table 1: Main set of initial conditions: the training dataset is generated within a range that spans +/- 20% of the standard operating conditions. The molar flowrate is expressed in mol/s *per* tube.

$\boldsymbol{T_0}$ [℃]	$\boldsymbol{P_0}$ $[\boldsymbol{bar}]$	$\dot{\boldsymbol{n}}_{\boldsymbol{CO,0}}$	$\dot{\boldsymbol{n}}_{\boldsymbol{CO_2,0}}$	$\dot{\boldsymbol{n}}_{\boldsymbol{CH_4,0}}$	$\dot{\boldsymbol{n}}_{\boldsymbol{CH_3OH,0}}$	$\dot{\boldsymbol{n}}_{\boldsymbol{H_2,0}}$	$\dot{\boldsymbol{n}}_{\boldsymbol{H_2O,0}}$	$\dot{\boldsymbol{n}}_{\boldsymbol{N_2,0}}$
245	65	0.0409	0.0764	0.1786	0.0032	0.4316	0.0007	0.0261

For training and test, we nondimensionalize the system, defining the nondimensional state variables T^*, P^*, and $\dot{n}_i^*$ and longitudinal coordinate z^*, by scaling the original values using the characteristic dimensions $< T >$, $< P >$, $< \dot{n}_i >$, and $< z >$, specifically taken as the average of each feature in the dataset. For a fair comparison, we trained and tested all the models on the scaled, nondimensional, dataset.

4. Results and discussion

We evaluate and compare the accuracy and runtime of the proposed data-driven and hybrid modeling methods with a conventional Python ODE solver. We build an MLP and PINN with 3 hidden layer with 512 neurons each. The TST consists of 5 encoder and decoder layers respectively.

4.1. Prediction runtime and accuracy

We compare the performance of a conventional ODE solver in Python (`scipy.integrate.solve_ivp`) with the presented alternatives in terms of accuracy and runtime. The test set, which includes 500 varied scenarios distributed within the same range as the training set, utilizes an ODE solver with very small tolerances, considered to be 100% accurate, and serves as the baseline for comparison. We assessed the solver's performance by adjusting the tolerance to quicken computation, albeit at the expense of reduced accuracy. Ultimately, we evaluated the outcome of MLP, PINNs and TST when applied to the unseen test dataset. We measure the accuracy as the complement of the Mean Absolute Percentage Error (MAPE).

Table 2 summarizes the main results. We evaluate the mean performance across 500 simulations using varying solver tolerances, MLP, PINN, and TST methodologies. As expected, the runtime of the ODE solver decreases significantly when the tolerances are reduced. Nonetheless, the MLP and PINNs offer a significantly shorter computational time, being 35 times faster than the least accurate version of the ODE solver, and still maintain a higher accuracy of approximately 99.5%. This outcome indicates a potential for utilizing MLP and PINNs models within larger optimization or simulation studies. For instance, in superstructure optimization problems, the reactor model often needs to be evaluated thousands of times at every iteration, depending on the plant complexity and assessed configurations. Thus, our work can potentially shift the overall computational time of such problems from several hours to few minutes. Considering that this preliminary analysis excludes hyperparameter tuning, it is anticipated that incorporating extended optimization analysis would likely improve accuracy of the models further. The computational time of the TST is higher due to its large architectural complexity. The accuracy of the TST prediction is simultaneously the lowest, suggesting the need for further investigation, for instance into more data-efficient modeling techniques.

Table 2: Runtime and accuracy comparison of the proposed methods. The ODE solver is tested when different relative (r) and absolute (a) tolerances are required. In the table, the tolerance numbers indicate the decimal digits (e.g., r10 stands for “relative tolerance = 10^{-10}”). The runtime is referring to CPU computation (11th Gen Intel Core i7).

Method	**Runtime [s]**	**Accuracy [%]**
ODESolver - r10a12 (baseline)	0.063	100.00
ODESolver - r3a3	0.013	99.97
ODESolver - r1a1	0.0056	98.55
MLP	0.00016	99.54
PINNs	0.00016	99.51
TST	12.06	96.87

5. Conclusions

We demonstrate that MLPs and PINNs can provide an alternative to classical modeling techniques given their high accuracy and low runtime. These considerations are especially important in the context of highly exhaustive modeling, for instance plant-wide optimization and simulation, where the reactor model needs to be solved repetitively. Notably, the MLP and PINNs demonstrate a 35-fold decrease in computation time compared to the least accurate version of the ODE Solver, while maintaining higher accuracy. Further investigation is needed to evaluate the ability of PINNs model to overcome the limitations in the application scope commonly associated with black-box models by extending the solution beyond the training limits. Finally, we envision the opportunity to embed physical knowledge into a simplified transformer architecture for steady-state and dynamic modeling of chemical systems.

Acknowledgements

This research is supported by Shell plc, for which we express sincere gratitude.

References

K. M. V. Bussche and G. F. Froment, 1996, A Steady-State Kinetic Model for Methanol Synthesis and the Water Gas Shift Reaction on a Commercial Cu/ZnO/Al2O3 Catalyst, Journal of Catalysis, Volume 161, June, p. 1–10.

A. Kolides, A. Nawaz, A. Rathor, D. Beeman, M. Hashmi, S.Fatima, D. Berdik, M. Al-Ayyoub, Y. Jararweh, 2023, Artificial intelligence foundation and pre-trained models: Fundamentals, applications, opportunities, and social impacts, Simulation Modelling Practice and Theory, Volume 126, July, p. 102754

L. McClenny and U. Braga-Neto, 2022, Self-Adaptive Physics-Informed Neural Networks using a Soft Attention Mechanism, ArXiv preprint at arXiv:2009.04544

M. Raissi, P. Perdikaris and G.E. Karniadakis, 2019, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, Volume 378, February, p. 686–707.

A.M., Schweidtmann, E. Esche, A. Fischer, M. Kloft, J. Repke, S. Sager, A. Mitsos, 2021, Machine Learning in Chemical Engineering: A Perspective, Chemie Ingenieur Technik, Volume 93, October, p. 2029–2039.

N. Sitapure and J.S. Kwon, 2023, Exploring the Potential of Time-Series Transformers for Process Modeling and Control in Chemical Systems: An Inevitable Paradigm Shift?, Chemical Engineering Research and Design, Volume 194, June, p. 461-477

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, 2017, Attention Is All You Need, 31st Conference on Neural Information Processing Systems (NIPS 2017)

M. von Stosch, R. Oliveira, J. Peres, and S.F de Azevedo, 2014, Hybrid semi-parametric modeling in process systems engineering: Past, present and future, Computers & Chemical Engineering, Volume 60, January, Pages 86–101.

S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, 2023, An Expert's Guide to Training Physics-informed Neural Networks, Arxiv preprint at arXiv:2308.08468

S. Wang, Y. Teng, and P. Perdikaris, 2020, Understanding and mitigating gradient pathologies in physics-informed neural networks, SIAM Journal on Scientific Computing, Volume 43, 5, p. A3055-A3081

Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, 2022, Transformers in Time Series: A Survey, Arxiv preprint at arXiv:2202.07125