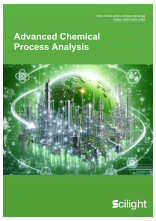
Scilight

Advanced Chemical Process Analysis

https://www.sciltp.com/journals/acpa



Article

Supplementary Materials: Capability of Digital Twins for Representing a Complex Process—Prediction of Multicomponent Adsorption Breakthrough Curves and Times

Jose Luis Valverde [1,*], Victor Roberto Ferro [2,†], María Mercedes Romero-Díaz [1,†] and Anne Giroir-Fendler [3,†]

[1] Department of Chemical Engineering, University of Castilla La Mancha, Avenida Camilo José Cela 10, 13071 Ciudad Real, Spain
[2] Department of Chemical Engineering, Universidad Autónoma de Madrid, C. Francisco Tomás y Valiente 7, Fuencarral-El Pardo, 28049 Madrid, Spain
[3] Department of Chemistry and Biochemistry, Université Claude Bernard Lyon 1, CNRS, IRCELYON, 2 Avenue Albert Einstein, F-69622 Villeurbanne, France
* Correspondence: joseluis.valverde@uclm.es; Tel.: +34-926-295-300
† These authors contributed equally to this work.

How To Cite: Valverde, J.L.; Ferro, V.R.; Romero-Díaz, M.M.; et al. Capability of Digital Twins for Representing a Complex Process—Prediction of Multicomponent Adsorption Breakthrough Curves and Times. *Advanced Chemical Process Analysis* **2025**, *1*(1), 2. https://doi.org/10.53941/acpa.2025.100002

1. Neural Network Training and Validation

For this study, a multilayer perceptron was constructed. It had the following three layers: (i) one input layer, (ii) one output layer, and (iii) one or more hidden layers. Neurons were connected from one layer to the next without feedback, this means a non-recurrent ANN. The information was transferred from the input to the output layers. The scalar input (p_i) from a neuron in each layer was multiplied by the corresponding scalar weight (w_{ij}) of the layer, forming the term $w_{ij}p_i$, which was sent to the sum [1]. The sum also had as input the multiplication of a constant (S1) by a bias (b_j). During the learning process, the weight that corresponded to the connection of units from i to j, as well as the bias, were continuously adjusted. The output (n_j), which is also referred to as the net input, was expressed as:

$$n_j = \sum_{i=1}^{N} w_{ij} p_i + b_j \quad \text{(S1)}$$

the n_j went into an activation function (f) which produced the scalar neuron output (a_j) that can be expressed as:

$$a_j = f(n_j) \quad \text{(S2)}$$

For example, for the Log-Sigmoid activation function, the scalar neuron output (a_j) would be:

$$a_j = \frac{1}{1 + e^{-n_j}} \quad \text{(S3)}$$

In the input layer, the number of input variables (independent variables) fixed the number of neurons, whereas in the output layer (just including linear activation functions), they were fixed by the number of output variables (dependent variables).

Levenberg-Marquardt algorithm [2] coupled to an ANOVA analysis (with $\alpha = 0.05$) was used for training the network [3–6]. The training process was performed by minimizing the function χ_j^2 by nonlinear regression:

$$\chi_j^2 = \sum_{i=1}^{m} \left[\sum_{k=1}^{n} \left(y_{ik} - a_{ikj}^{s} \right)^2 \right] \quad \text{(S4)}$$

where m is the number of experiments, n is the number of output variables, y_{ik} and a^s_{ikj} are the values of the output experimental variables and the values predicted by the neural network that corresponds to scenario j, respectively.

The regression procedure was maintained whenever the relative error (RE) was lower than 10^{-4}. The RE was defined as follows:

$$RE = \left| \frac{(\chi^2_k)_{j+1} - (\chi^2_k)_j}{(\chi^2_k)_j} \right| \tag{S5}$$

with j being the number of iterations in the non-linear regression procedure, comparing the results of the prediction made by the ANN in the scenario j and the scenario $j + 1$.

Once the fitting process for a given scenario is finished, the above-defined function χ^2_j, the Pearson's ratio coefficient for the model, r, and the root mean squared error (RSME) were collected. The RMSE was defined as:

$$RMSE_J = \sqrt{\frac{\chi^2_j}{M}} \tag{S6}$$

To minimize overfitting in the validation process, a discrimination procedure was established. For this purpose, a function presented by Equation (S4) was defined for the experiments used for the validation (m_V), obtaining Equation (S7):

$$(\chi^2_j)_V = \sum_{i=1}^{m_V} \left[\sum_{k=1}^{n} (y_{ik} - a^s_{ikj})^2 \right] \tag{S7}$$

This way, a new function was defined from Equations (4) and (7):

$$(\chi^2_j)_T = \chi^2_j + (\chi^2_j)_V \tag{S8}$$

The scenario with the lower value of the function represented by Equation (S8), was selected, since it was the scenario that led to an optimized result. Finally, non-linear regression was taken as the reference of the selected model until a new value of RE lower than 10^{-6} was reached. With the values of the parameters finally obtained, the validation procedure was completed. In both cases, the final fitting and validation, the above defined function χ^2_j plus those defined for each output variable k, can be expressed as χ^2_{jk}:

$$\chi^2_{jk} = \sum_{i=1}^{m} (y_{ik} - a^s_{ikj})^2 \tag{S9}$$

2. Figures

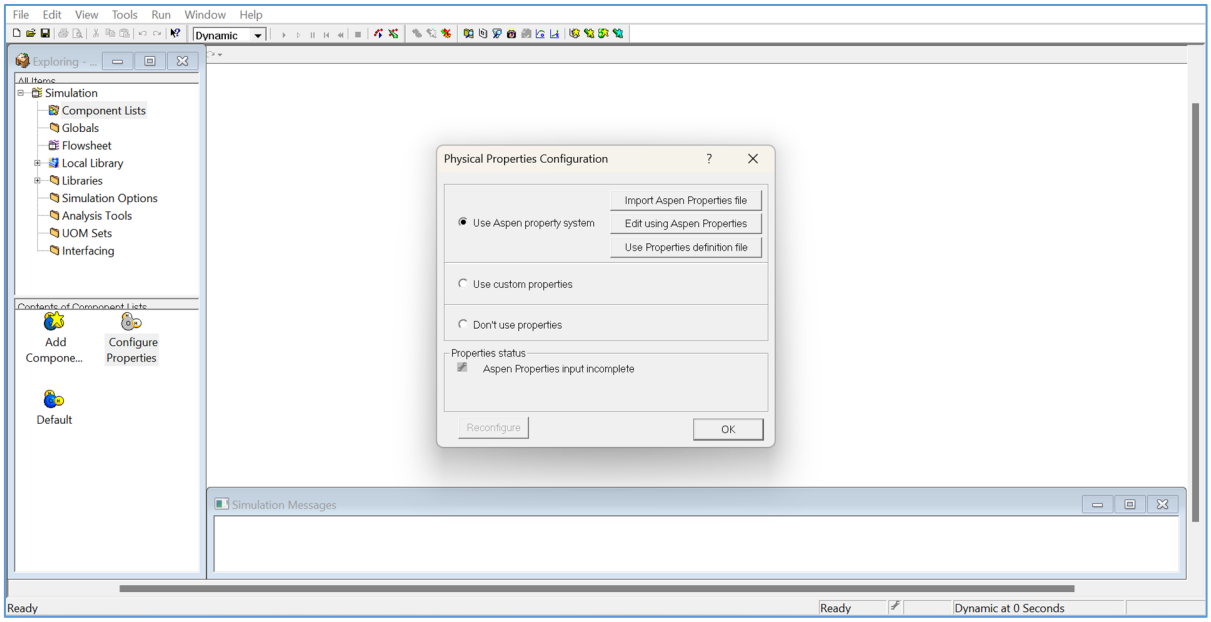


Figure S1. Components selection.

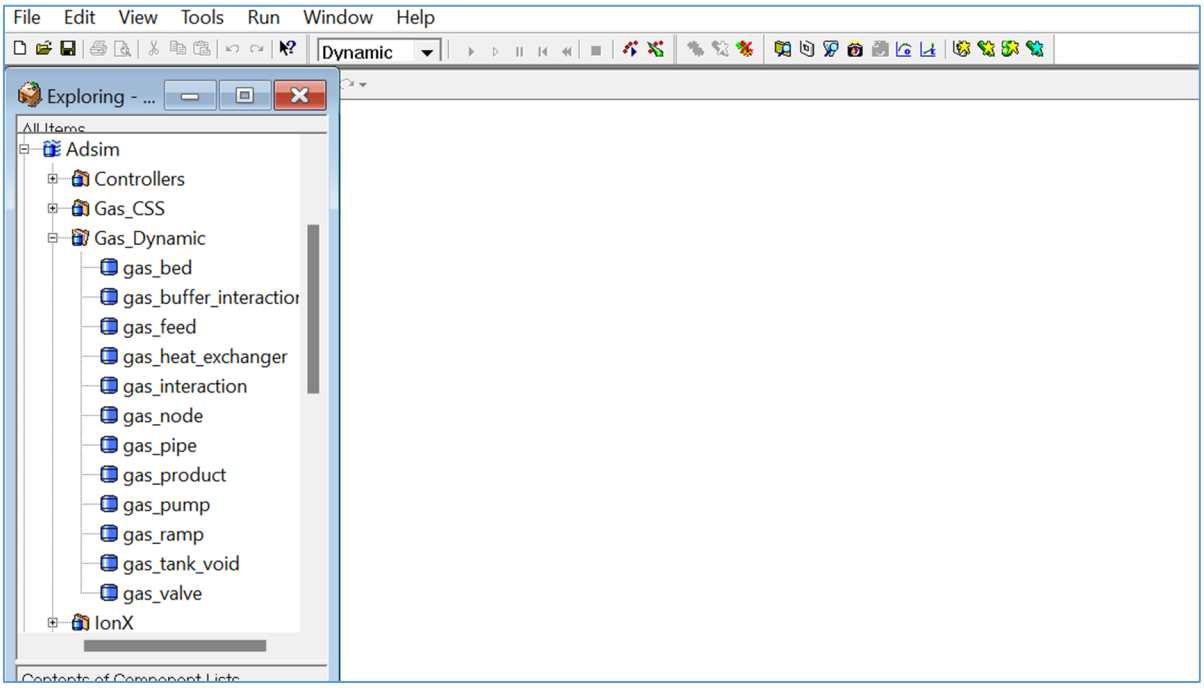


Figure S2. Windows with the different blocks to be selected.

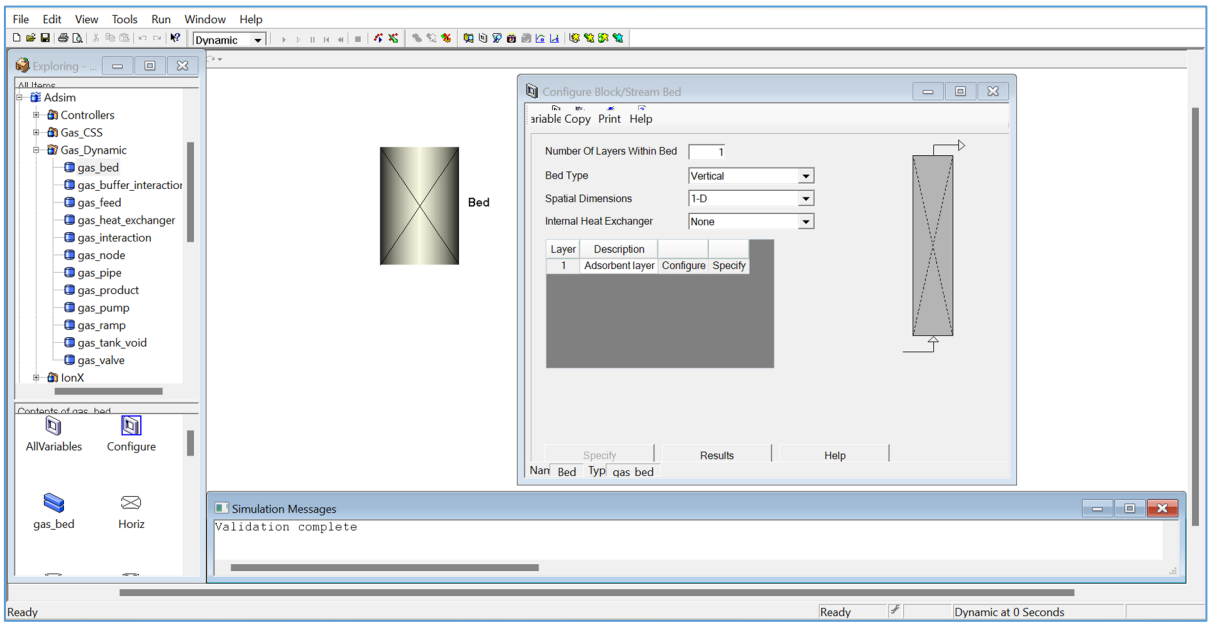


Figure S3. Configuration of the geometric properties of the adsorption fixed bed.

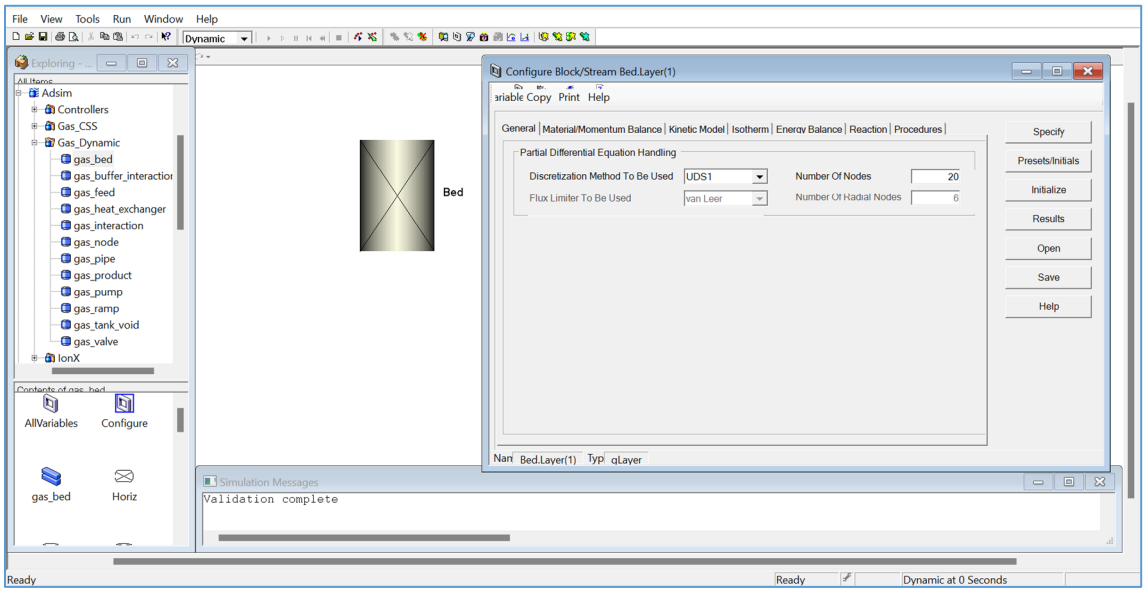


Figure S4. Configuration of the fixed bed layers.

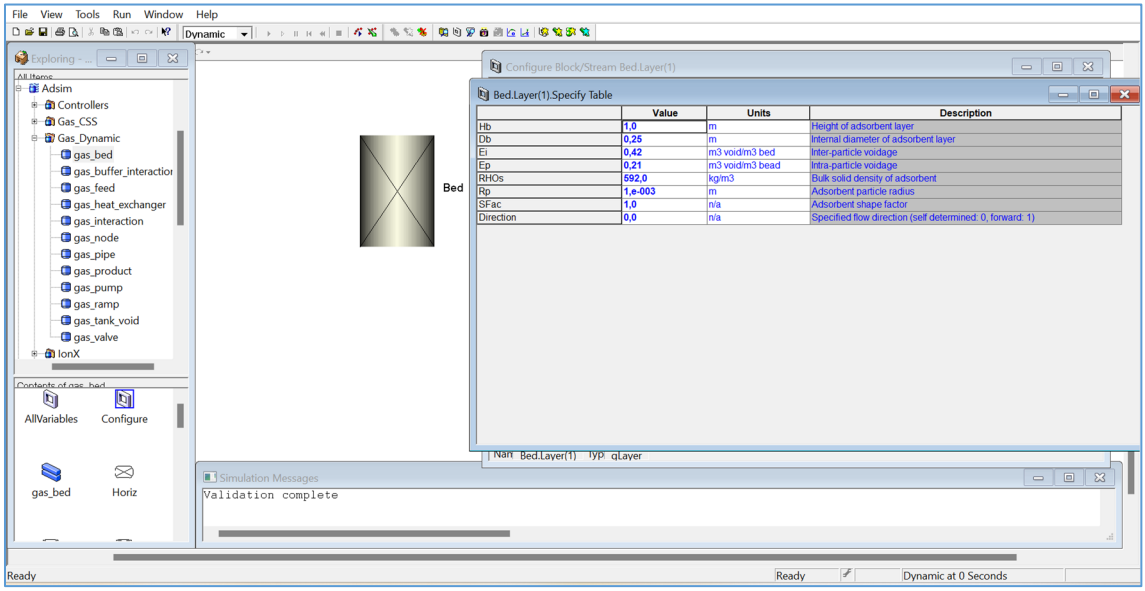


Figure S5. Bed layer (1).

3. Table

Table S1. The five best ANN models fitted the simulation data obtained from Aspen Adsorption™ by considering a time on stream of 100 s.

Order	Model	RMSE	$(r^2)_{training}$	$(\chi^2)_T$	$(r^2)_{validation}$
1	7-ELU-5-LINEAR-3	2.311×10^{-2}	0.990	1.363×10^{1}	0.987
2	7-POL-2-HTANSIG-3-LINEAR-3	5.366×10^{-2}	0.942	7.170×10^{1}	0.9861
3	7-POL-2-HTANSIG-4-LINEAR-3	5.652×10^{-2}	0.935	7.859×10^{1}	0.938
4	7-POL-4-RELU-2-LINEAR-3	6.065×10^{-2}	0.926	8.991×10^{1}	0.931
5	7-LOGSIG-3-SWISH-4-LINEAR-3	6.436×10^{-2}	0.916	9.983×10^{1}	0.931

References

1. Hagan, M.T.; Demuth, H.B.; Beale, M. *Neural Network Design*; PWS Publishing Co.: Stillwater, OK, USA, 1997.
2. Hagan, M.T.; Menhaj, M.B. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993. https://doi.org/10.1109/72.329697.
3. Barbero-Sánchez, J.; Megía-Ortega, A.; Ferro, V.R.; et al. Exploring alternatives to create digital twins from and for process simulation. *J. Comput. Sci. Res.* **2024**, *6*, 16–30. https://doi.org/10.30564/jcsr.v6i1.6168

4. Valverde, J.L.; Ferro, V.R.; Giroir-Fendler, A. Prediction of the solid-liquid equilibrium of ternary and quaternary salt-water systems. Influence of the e-NRTL interaction parameters. *Fluid Phase Equilib.* **2023**, *572*, 113832. https://doi.org/10.1016/j.fluid.2023.113832.
5. Valverde, J.L.; Ferro, V.R.; Giroir-Fendler, A. Application of the e-NRTL model to electrolytes in mixed solvents methanol-, ethanol-water, and PEG-water. *Fluid Phase Equilib.* **2022**, *560*, 113516. https://doi.org/10.1016/j.fluid.2022.113516.
6. Valverde, J.L.; Ferro, V.R.; Giroir-Fendler, A. Estimation of e-NRTL binary interaction parameters and its impact on the prediction of thermodynamic properties of multicomponent electrolyte systems. *Fluid Phase Equilib.* **2022**, *551*, 113264. https://doi.org/10.1016/j.fluid.2021.113264.