A new formulation for symbolic regression to identify physico-chemical laws from experimental data

Pascal Neumann,[a,b] Liwei Cao,[b,c] Danilo Russo,[b] Vassilios S. Vassiliadis,[b] Alexei A. Lapkin[b,c*]

[a]Aachener Verfahrenstchnik – Process Systems Engineering, RWTH Aachen University, Aachen, Germany

[b]Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge CB3 0AS, UK

[c]Cambridge Centre for Advanced Research and Education in Singapore, CARES Ltd. 1 CREATE Way, CREATE Tower #05-05, 138602 Singapore

Abstract

A modification to the mixed-integer nonlinear programming (MINLP) formulation for symbolic regression was proposed with the aim of identification of physical models from noisy experimental data. In the proposed formulation, a binary tree in which equations are represented as directed, acyclic graphs, is fully constructed for a pre-defined number of layers. The introduced modification results in the reduction in the number of required binary variables and removal of redundancy due to possible symmetry of the tree formulation. The formulation was tested using numerical models and was found to be more efficient than the previous literature example with respect to the numbers of predictor variables and training data points. The globally optimal search was extended to identify physical models and to cope with noise in the experimental data predictor variable. The methodology was proven to be successful in identifying the correct physical models describing the relationship between shear stress and shear rate for both Newtonian and non-Newtonian fluids, and simple kinetic laws of chemical reactions. Future work will focus on addressing the limitations of the present formulation and solver to enable extension of target problems to larger, more complex physical models.

* Corresponding author. A. Lapkin email: aal35@cam.ac.uk

Keywords: model identification; chemical process development; symbolic regression; automated model construction; mixed-integer nonlinear programming (MINLP); global optimization

1. Introduction

Today we experience rapid development of a new field of chemical science – digital molecular technology (DMT). This is evident by the increasing number of publications in which synthetic and computational chemistry, or materials development, are mixed with machine learning (ML), robotics and artificial intelligence (AI), for example in Refs. [1–6]. DMT promises to significantly expand the accessible chemical space, and to reduce the price of access to new functional molecules and materials. This will be achieved by enhanced capabilities in (*i*) discovery of molecules and reactions, and (*ii*) process development and optimization. The key component of the new DMT methods is the increased volume and quality of chemical data obtained both through data mining, computational chemistry tools and robotic experiments, as lack of data renders ML and AI methods inaccurate and not very useful [7]. Here we ask a question, *is it possible to make use of the increased availability of experimental data to enhance our capability in inferring physical knowledge from data by means of algorithmic research*?

Our curiosity is driven by the desire to develop predictive models of complex chemical processes, which could later be used in optimal control. The approach that we seek to develop should, ideally, not be based on selecting functional forms from a pre-defined set. This has already been demonstrated within DMT, for example, in selecting suitable kinetic expressions within an automated self-optimization system [8]. Our own interest is in the methods that are inherently not restricted to only the known functional forms and are, therefore, potentially capable of discovering new physical models.

Automation of chemical research has recently emerged as a highly promising broad methodology [9]. Significant benefits of automated systems include the precise control of the operating conditions, and the ability to generate large, reliable and reproducible datasets [10]. Thanks to the advances in computational power, automated systems can be combined with machine learning (ML) techniques and applied in the discovery of novel materials [11] and reactions [12], in process development [13] and process optimization [8]. Automated systems consist of three main components: a decision-making algorithm, a physical platform, capable of undertaking the desired class of experiments autonomously, and an automated inline/online

analytical setup [14]. Being the 'brain' of the automated experimental platforms, the decision algorithms play a vital role. On this basis, ML algorithms based on statistical surrogate modelling approaches have become increasingly popular, having shown that they provide accurate models for predictions [15].

On the contrary, the data-driven development of interpretable physical models resisted automation for a long time [16]. The possibility of building data-driven interpretable and generalizable models for complex and poorly understood physical systems is important as these models share a similar structure to those, based on first principles, and can be transferred to analogous systems, whereas surrogate models cannot be easily generalized [17]. This, in the longer term, can improve the time and resource efficiency for the product discovery and process optimization, especially for the manufacturability and the scale-up, for which a mechanistic understanding is often crucial [8,18,19].

The field of algorithmic search for physical models is relatively new, but has seen a number of important advances. There are two main types of methods in this field: parametric and non-parametric regression. In parametric regression, the potentially nonlinear functional form is known *a priori* or approached by a weighting of multiple known basis functions. As a parametric approach, ALAMO (automated learning of algebraic models for optimization) platform can identify surrogate models from small datasets that are as accurate and as simple as possible [20]. In [21] ALAMO was extended to incorporate *a priori* physical knowledge by enforcing physical constraints on the model resulting from parametric regression; this was further illustrated through application of ALAMO to learning problems in kinetics [22]. Similar approaches that linearly combine the candidate functions from a pre-defined library are numerous in the literature domain of sparse regression. The identification of a data-driven physical model (DDPM), specifically of sparse and interpretable (partial) differential equations of nonlinear dynamical systems, has been successful with parametric regression techniques [23–28]. For instance, the technique was successfully applied to the data-driven discovery of Navier-Stokes equations [24]. Furthermore, sparse regression was extended to include selection of candidate basis functions by dimensional analysis, and by determining the parameters including error bars by a hierarchical Bayesian framework [29].

In non-parametric regression, the *a priori* selection of basis functions is not needed. Therefore, the non-parametric methods allow us to find free form equations. Recently, Brunton *et al*.

introduced a sparse regression approach to discover equations governing the physics in a chaotic Lorenz system, and in a fluid vortex dynamic system [30]. However, this technique is restricted to a pre-defined algebraic model structure, as selected basis functions are linearly combined. Allowing free-form analytical equations, Bongard and Lipson [31] developed a criterion to find meaningful and complex mathematically invariant models by means of the ML method of symbolic regression (SR).

Recent applications of SR for physical models can be found in civil engineering [32] and material science [33]. Although successfully proven, the earlier proposed SR was based on a heuristic search that could terminate the optimization in local minima solutions, potentially producing less suitable models than possible. Additionally, as the structure of a model reflects the actual mechanistic interactions within the system studied, these approximate solutions cannot be used reliably to infer any mechanistic information about the system, *i.e.* to use it to identify chemical reaction mechanisms with certainty. Acknowledging this disadvantage, SR is formulated as a mixed-integer nonlinear programming (MINLP) in Refs. [34, 35] and solved to global optimality. However, until now the method remains in the mathematical domain and is yet to be applied to physical models and noisy experimental data.

This paper aims to advance the method of globally optimal symbolic regression towards automated, data-driven identification of physical models, and its applications to chemical engineering case studies. Compared to additive models in conventional regression and heuristic searches in SR, the globally optimal data-driven modelling technique, without any previously imposed model structure, is expected to discover true underlying relationships more reliably. To accomplish this, in this paper a modified optimization formulation of SR is developed and implemented in combination with a framework for physical model selection. As proof of concept, several case studies were investigated in the areas of rheology and reaction engineering. The purpose of this work is to illustrate an automated research pipeline deriving interpretable and generalizable models, and thereby providing access to physical knowledge from data. Within this big picture, closing the loop of utilizing the obtained physical models in further experimentation and generation of physical knowledge by (automated) interpretation, see Figure 1, remains the target for future research. The caveat to this is that we cannot expect such a methodology to be able to discover new phenomena for which no physical response is measured. The assumptions for the model are that all predictor variables are included in

measurements, all possible operators are included and the tree structure is large-enough to be able to discover the physical phenomena of a sufficiently complex system.

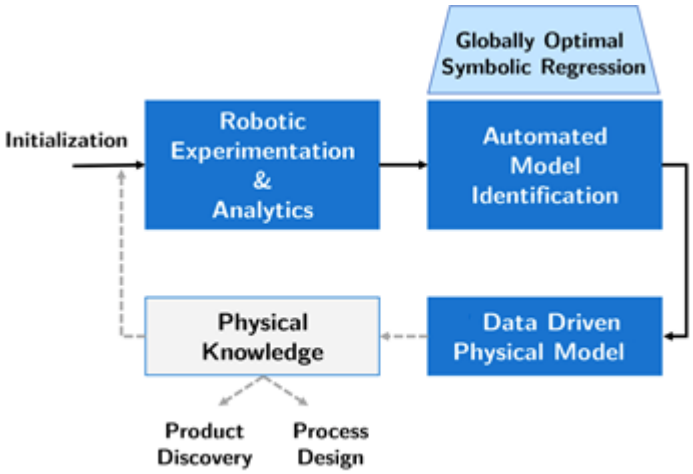


Figure 1. Schematic diagram of a concept of a closed loop automated physical model identification.

2. Materials and Methods

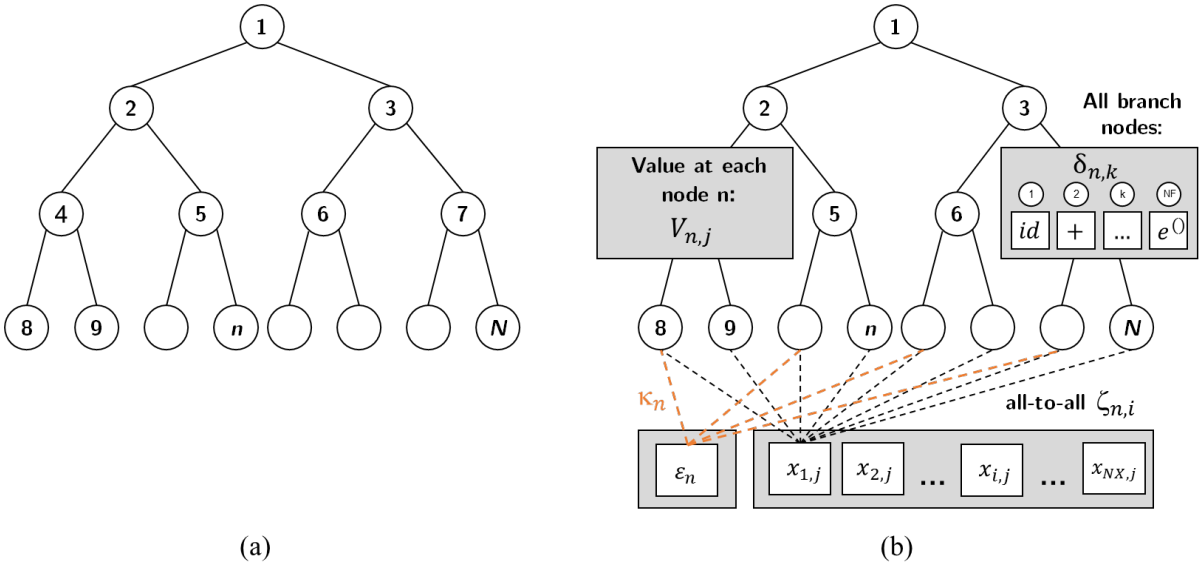
2.1. MINLP Formulation

Use will be made of the Directed Acyclic Graph (DAG) description of algebraic functions [36] throughout this work. The MINLP formulation is based on a balanced binary tree superstructure for representation of the equations describing a physical model. The overall goal is to enable the assembly of free-form algebraic functions by connecting predictor variables and operators, such that the resulting function predicts the dependent variable values accurately. As an example, the structure with nodes in a four-layer balanced binary tree is shown in Figure 2a.

The formulation presented here is based on Ref [35], but follows a different concept in the set-up of the binary tree, which allows us to reduce the number of binary variables in the global optimization. These modifications are detailed in Section 2.3.

An expression tree consists of $N = 2^{NL} - 1$ nodes, where NL defines the number of layers. All nodes that have a connection with nodes on a lower level, their child nodes, are called branch-nodes ($\mathcal{N}_b$) or non-leaf nodes, and house a mathematical operator. The nodes on the

lowest layer in the tree, referred to as leaf-nodes ($\mathcal{N}_l$), are assigned to a predictor variable ($x_{i,j}$) or a constant (ϵ_n). In the following sections we will refer to the total number of activated nodes in the tree as "complexity of the model" (except the ones with an identity operator). Each given data point deployed in the SR is described by two parameters: the value of the predictor variable ($x_{i,j}$) and the dependent variable value (y_j), which is predicted by the model for each training data point (j). As shown in Figure 2b, the input variables are assigned for selection at the leaf-nodes only, while the dependent variable values are used at the root node ($n = 1$) for comparison with the model prediction.



(a) (b)

Figure 2. An example of a directed acyclic graph. (a) Binary rooted tree, (b) MINLP set-up in connection with the expression tree.

Each node has a value for each data point ($V_{n,j}$), which is computed to be used as operator arguments on the layer above. The nodal values at the bottom of the tree are determined by the selection of an input variable or a constant. On branch node layers, nodal values are specified by the selected operator in combination with the node values of their children. The allocation of the input predictor variables ($x_{i,j}$) is implemented by the binary variables ($\zeta_{n,i}$). Continuous decision variables (ϵ_n) with bounds (ϵ_n^{lo}) and (ϵ_n^{up}) are designated for constants at every even leaf-node. To decide between a variable input and the selection of a constant at the even leaf-nodes, further binary variables (κ_n) are assigned. Both (κ_n) and (ϵ_n) are only assigned to the even leaf-nodes, as the left leaf-nodes in each last bifurcation only contain the constant values. With regard to the branch-nodes, there are binary variables ($\delta_{n,k}$) assigned for operator selection, where an operator is active at node n if $\delta_{n,k} = 1$ and inactive if $\delta_{n,k} = 0$. If active,

each binary operator is applied using both child nodes while a unary operator uses only the value of the left node ($V_{2n,j}$). Five binary operators (addition, subtraction, multiplication, division and power law) and three unary operators (identity, exponential and square root) were implemented. It should be noted that the list of operators can be extended further, such as cubic, square or logarithm operations as proven in Ref. [35].

Consequently, by using the tree structure and the index assignment (Tables 1-3), the optimization problem was formulated with the objective to minimize the sum of squared errors (SSE) between the values computed by the model and the experimental data, Eq. 1, according to Ref. [35].

Table 1. MINLP Notation: Set

Description	Index	Set	Value
Nodes	n	$\mathcal{N}$	$\{1, \dots, N\}$
Branch-nodes		$\mathcal{N}_b$	$\{1, \dots, 2^{NL-1} - 1\}$
Leaf-nodes		$\mathcal{N}_l$	$\{2^{NL-1}, \dots, N\}$
Even leaf-nodes		$\mathcal{N}_l^*$	$\{2^{NL-1},\ 2^{NL-1} + 2 \dots, N\}$
Algebraic Operators	n	$\mathcal{F}$	$\{+, -, \dots\}$
Predictor Variables	i	$\mathcal{X}$	$\{1, \dots, NX\}$
Input Data Points	j	$\mathcal{J}$	$\{1, \dots, NE\}$

Table 2. MINLP notation: Parameters.

Description	Parameter	
Input variable values	$x_{i,j}$	$i \in \mathcal{X}, j \in \mathcal{J}$
Dependent variable values	y_j	$j \in \mathcal{J}$

Table 3. MINLP notation: Decision variables.

Applicability	Description	Variable		Bounds
General	Nodal values	$V_{n,j}$	$n \in \mathcal{N}_b, j \in \mathcal{J}$	$\left[V_{n,j}^{lo}, V_{n,j}^{up}\right] \in \mathbb{R}$
Leaf-nodes	Variable selection	$\zeta_{n,i}$	$n \in \mathcal{N}_l, i \in \mathcal{X}$	$\{0,1\}$
	Constant selection	κ_n	$n \in \mathcal{N}_l^*$	$\{0,1\}$
	Value of constants	ϵ_n	$n \in \mathcal{N}_l^*$	$\left[\epsilon_{n,j}^{lo}, \epsilon_{n,j}^{up}\right] \in \mathbb{R}$

Operator selection	$\delta_{n,k}$	$n \in \mathcal{N}_b, k \in \mathcal{F}$	$\{0,1\}$

$$\min \sum_{j=1}^{NE} \left(y_j - V_{1,j}\right)^2 \tag{1}$$

$$f_k\left(V_{n,j}, V_{2n,j}, V_{2n+1,j}\right) \leq M_{n,j,k}^{up}\left(1-\delta_{n,k}\right),\ n \in \mathcal{N}_b, k \in \mathcal{F}, j \in \mathcal{T} \tag{2}$$

$$f_k\left(V_{n,j}, V_{2n,j}, V_{2n+1,j}\right) \geq M_{n,j,k}^{lo}\left(1-\delta_{n,k}\right),\ n \in \mathcal{N}_b, k \in \mathcal{F}, j \in \mathcal{T} \tag{3}$$

$$g_k\left(V_{n,j}, V_{2n,j}, V_{2n+1,j}\right) \leq G_{n,j,k}^{up}\left(1-\delta_{n,k}\right),\ n \in \mathcal{N}_b, k \in \mathcal{F}, j \in \mathcal{T} \tag{4}$$

$$V_{n,j} \leq V_{n,j}^{up} \sum_{k\in\mathcal{F}} \delta_{n,k},\ n \in \mathcal{N}_b, j \in \mathcal{T} \tag{5}$$

$$V_{n,j} \geq V_{n,j}^{lo} \sum_{k\in\mathcal{F}} \delta_{n,k},\ n \in \mathcal{N}_b, j \in \mathcal{T} \tag{6}$$

$$\sum_{k\in\mathcal{F}} \delta_{n,k} \leq 1,\ n \in \mathcal{N}_b \tag{7}$$

$$V_{n,j} = \sum_{i\in\mathcal{X}} \zeta_{n,i}\, x_{i,j} + \kappa_n \epsilon_n,\ n \in \mathcal{N}_l^*, j \in \mathcal{T} \tag{8}$$

$$V_{n,j} = \sum_{i\in\mathcal{X}} \zeta_{n,i}\, x_{i,j},\ n \in \mathcal{N}_l \backslash \mathcal{N}_l^*, j \in \mathcal{T} \tag{9}$$

$$\sum_{i\in\mathcal{X}} \zeta_{n,i} + \kappa_n \leq 1,\ n \in \mathcal{N}_l^* \tag{10}$$

$$\sum_{i\in\mathcal{X}} \zeta_{n,i} \leq 1,\ n \in \mathcal{N}_l \backslash \mathcal{N}_l^* \tag{11}$$

$$\sum_{n\in\mathcal{N}_l} \sum_{i\in\mathcal{X}} \zeta_{n,i} \geq 1 \tag{12}$$

$$\delta_{n,k} \in \{0,1\},\ n \in \mathcal{N}_b, k \in \mathcal{F} \tag{13}$$

$$\zeta_{n,i} \in \{0,1\},\ n \in \mathcal{N}_l, i \in \mathcal{X} \tag{14}$$

$$\kappa_n \in \{0,1\},\ n \in \mathcal{N}_l^* \tag{15}$$

$$V_{n,j} \in [V_{n,j}^{lo}, V_{n,j}^{up}],\ n \in \mathcal{N} \tag{16}$$

$$\epsilon_{n,j} \in [\epsilon_{n,j}^{lo}, \epsilon_{n,j}^{up}],\ n \in \mathcal{N}_l^* \tag{17}$$

Eqs. 2-4 enable the operator selection at branch-nodes with a big-M approach. With the aim of connecting the nodal values in the hierarchy of the tree, the functions (f_k) are introduced for each operator. If additional requirements apply for the selection of an operator, such as a non-zero denominator for division or a positive base for a power law, these are provided in the functions (g_k) enforcing a minimum distance to undefined regions. A detailed explanation of the implementation of big-M approach can be found in Refs. [35] and [37]. The idea of the big-M facilitates the transformation of the disjunctive choice between the operators into linear constraints [37]. If $\delta_{n,k} = 1$, the two inequality constraints reduce to $f_k = 0$, which enforces the respective mathematical operation. In contrast, if $\delta_{n,k} = 0$, large M-values, as lower and upper bounds, ensure the free selection of nodal values within their specific bounds. The

coefficients ($M^{up}_{n,j,k}$), ($M^{lo}_{n,j,k}$) and ($G^{up}_{n,j,k}$) were introduced to preserve constraint linearity in the binary decision variables, which is important to reduce non-convexities and for solving the MINLP efficiently.

If no operator is selected, its nodal value is set to zero by constraints 5-6. Additionally, either none or one operator can be selected at the branch-nodes. Hence, the sum of operator binaries must be less or equal to one, which is constrained by Eq. 7.

In contrast to the branch-node values, the values at the leaf-nodes are determined by equality constraints including the binary selection of predictor variables or constants, Eqs. 8-9. Also, Eqs. 10-11 make sure that either no operand, one variable or one constant can be assigned. Overall, the model should include at least one predictor variable, which is ensured by Eq. 12. For the purpose of completeness, Eqs. 13-17 depict the bounds on decision variables of the MINLP [35].

Due to the binary architecture and the commutative nature of addition and multiplication, the expression tree contains many mathematically invariant models (symmetries). The design of the formulation should, therefore, impede redundancies. Eqs. 18-19 resemble cuts in the tree such that, if a unary operator is selected, the unused part towards the right child node is set to zero [35]. Eqs. 20-23 assure that if an operator is selected on a lower layer of the expression tree, there is an operator attached to the parental node [35]. Likewise, it ensures that the children of a node with value zero also have no operator or variables attached.

Additionally, symmetry breaking cuts (SC) to remove redundant solutions, which are caused by the commutative nature of addition and multiplication, were implemented. Eq. 24 is sufficient for one data point $j = j'$ to impose an order on the values of the child nodes [35]. The symmetry breaking cuts also pose as big-M constraints where M^{*}_{n} is set using interval arithmetic on the bounds of the two child node values [37].

$$\sum_{k\in\mathcal{F}_{unary}} \delta_{n,k} \leq 1 - \sum_{k\in\mathcal{F}} \delta_{2n+1,k},\ n \in \{1,\dots,2^{NL-2}-1\} \quad (18)$$

$$\sum_{k\in\mathcal{F}_{unary}} \delta_{n,k} \leq 1 - \sum_{i\in\mathcal{X}} \zeta_{2n+1,i},\ n \in \{2^{NL-2},\dots,2^{NL-1}-1\} \quad (19)$$

$$\sum_{k\in\mathcal{F}} \delta_{n,k} \geq \sum_{k\in\mathcal{F}} \delta_{2n,k},\ n \in \{1,\dots,2^{NL-2}-1\} \quad (20)$$

$$\sum_{k\in\mathcal{F}} \delta_{n,k} \geq \sum_{k\in\mathcal{F}} \delta_{2n+1,k},\ n \in \{1,\dots,2^{NL-2}-1\} \quad (21)$$

$$\sum_{k\in\mathcal{F}} \delta_{n,k} \geq \sum_{i\in\mathcal{X}} \zeta_{2n,i} + \kappa_n, n \in \{2^{NL-2}, \dots, 2^{NL-1}-1\} \quad (22)$$

$$\sum_{k\in\mathcal{F}} \delta_{n,k} \geq \sum_{i\in\mathcal{X}} \zeta_{2n+1,i}, n \in \{2^{NL-2}, \dots, 2^{NL-1}-1\} \quad (23)$$

$$V_{2n,j'} - V_{2n+1,j'} \geq M_{n,j'}^{SC}(1 - \sum_{k=\{+,*\}} \delta_{n,k}), n \in \mathcal{N}_b, \exists j' \in \mathcal{J} \quad (24)$$

Without a doubt and from experience with big-M formulations in the Mathematical Programming community, this will lead to a rather loose lower bounding in the associated Branch-and-Bound (B&B) traditionally used to solve Mixed-Integer Linear (MILP) and Mixed-Integer Nonlinear Programming (MINLP) problems. Indeed, such were the observations reported later in the computational results of this work, and hence the serious limitations that is a challenge for future development of this rigorous methodology.

2.2 **Determination of Big-M Coefficients**

For the optimization, the MINLP variable bounds as well as appropriate big-M values must be provided. Keeping them as small as possible is expected to reduce convergence times in solving the MINLP. In the following, an automatable approach for any supplied dataset is proposed based on the bottom-up interval arithmetics on the training data itself. Initiated at the leaf-nodes, the node value bounds $V_{n,j}^{lo/up}$ can be determined based on the maximum and minimum values within the dataset ± a safety margin ϱ. At even leaf-nodes the bounds of constants are considered (Eqs. 25-26). The bounds on the constants $\epsilon_n^{lo/up}$ are defined *a priori*.

$$V_{n,j}^{lo} = \min\left[\epsilon_n^{lo}, x_{i,j} \forall i\right] - \varrho, j \in \mathcal{J}, n \in \mathcal{N}_l^* \quad (25)$$

$$V_{n,j}^{up} = \max\left[\epsilon_n^{up}, x_{i,j} \forall i\right] - \varrho, j \in \mathcal{J}, n \in \mathcal{N}_l^* \quad (26)$$

For the next layer above, and subsequently for all other branch-nodes, the overall nodal value bounds $V_{n,j}^{lo/up}$ are determined as maximum or minimum value of interval arithmetics based nodal bounds $V_{n,j,k}^{lo/up}$ calculated for each operator and training data point (Eqs. 27-28). Calculation of the operator specific $V_{n,j,k}^{lo/up}$ is defined in Table. 4. For improved readability in the table, the *j*-indices are omitted but must be considered in the calculation. As the intervals grow quickly with the number of layers, a pre-defined nodal value limit V_{limit} can be set. Moreover, the root node bounds are determined top-down from the values of the independent variable to be predicted. A model that remains within two standard deviations 2σ of the expected value is proposed in Eq. 29.

$$V_{n,j}^{lo} = \min\left[-V_{limit}, V_{n,j,k}^{lo}\ k \in \mathcal{F}\right], n \in \mathcal{N}_b, j \in \mathcal{J} \quad (27)$$

$$V_{n,j}^{up} = \max\left[V_{limit}, V_{n,j,k}^{up}\ k \in \mathcal{F}\right], n \in \mathcal{N}_b, j \in \mathcal{J} \quad (28)$$

$$\left[V_{1,j}^{lo}, V_{1,j}^{up}\right] = [y_j - 2, y_j + 2\sigma], j \in \mathcal{J} \quad (29)$$

The big-M value bounds for each data point and operator can then be calculated using the $V_{n,j,k}^{lo/up}$ with Eqs. 30-31. They can also be subject to a user-defined limit M_{limit} as they grow quickly with the number of layers. For division and square root, different rules for the M-values apply as their functions have different formulations [35]. These rules are given in Table 5 omitting the usually required *j*-indices.

$$M_{n,j,k}^{lo} = V_{n,j,k}^{lo} - V_{n,j}^{up}, n \in \mathcal{N}_b, k \in \mathcal{F}\backslash\{/, \sqrt{}\}, j \in \mathcal{J} \quad (30)$$

$$M_{n,j,k}^{up} = V_{n,j,k}^{up} + \left|V_{n,j}^{lo}\right|, n \in \mathcal{N}_b, k \in \mathcal{F}\backslash\{/, \sqrt{}\}, j \in \mathcal{J} \quad (31)$$

$$M_{n,j'}^{SC} = V_{2n,j'}^{lo} - V_{2n+1,j'}^{up}, n \in \mathcal{N}_b, \exists j' \in \mathcal{J} \quad (32)$$

The upper bounds $G_{n,j,k}^{up}$ in Eq. (4) are calculated applying the same logic. For the division it is simply the safety margin ϱ itself and in the cases of power law or square root the absolute value of the lower bound of the restricted nodal value in $g_{n,j}$ is added. For the successful application of symmetry breaking cuts, the big-M values $M_{n,j'}^{SC}$ are determined for a single data point j' following Eq. 32.

Table 4. Calculation of the nodal value bounds.

Description	Index	Lower Bound $V_{n,j,k}^{lo}$	Upper Bound $V_{n,j,k}^{up}$
Addition	$+$	$V_{2n}^{lo} + V_{2n+1}^{lo} - \varrho$	$V_{2n}^{up} + V_{2n+1}^{up} + \varrho$
Subtraction	$-$	$V_{2n}^{lo} - V_{2n+1}^{up} - \varrho$	$V_{2n}^{up} - V_{2n+1}^{lo} + \varrho$
Multiplication	$\cdot$	$\min(arg\,\cdot) - \varrho$	$\min(arg\,\cdot) + \varrho$
		$arg\,\cdot = [V_{2n}^{up}V_{2n+1}^{up}, V_{2n}^{up}V_{2n+1}^{lo}, V_{2n}^{lo}V_{2n+1}^{up}, V_{2n}^{lo}V_{2n+1}^{lo}]$	
Division	$/$	$\min(arg/) - \varrho$	$\min(arg/) + \varrho$
		$arg\,\cdot = [\frac{V_{2n}^{up}}{V_{2n+1}^{up}}, \frac{V_{2n}^{up}}{V_{2n+1}^{lo}}, \frac{V_{2n}^{up}}{\sqrt{\epsilon}}, \frac{V_{2n}^{lo}}{V_{2n+1}^{up}}, \frac{V_{2n}^{lo}}{V_{2n+1}^{lo}}, \frac{V_{2n}^{lo}}{\sqrt{\epsilon}}]$	
Power Law	^	$-\varrho$	$V_{2n+1}^{V_{2n}} + \varrho$
Identity	id	$V_{2n}^{lo} - \varrho$	$V_{2n}^{up} + \varrho$
Exponential	e	$-\varrho$	$e^{V_{2n}^{up}} + \varrho$
Square Root	$\sqrt{}$	$-\varrho$	$\sqrt{V_{2n}^{up}} + \varrho$

Table 5. Calculation of big-M values for division and square root.

Description	Index	Lower Bound $M_{n,j,k}^{lo}$	Upper Bound $M_{n,j,k}^{up}$
Division	/	$V_{2n}^{lo} - \left\|V_n^{up}\right\| \max(arg) - \varrho$	$V_{2n}^{up} - \left\|V_n^{lo}\right\| \max(arg) + \varrho$
		$arg = [V_{2n+1}^{up}, \left\|V_{2n+1}^{lo}\right\|]$	
Square Root	√	$V_{2n}^{lo} - \max\left(V_{2n}^{up}, \left\|V_{2n}^{lo}\right\|\right)^2 - \varrho$	$V_{2n}^{up} + \varrho$

2.3. Details of Modifications of the Formulation

The previously reported MINLP formulation [35] was modified in order to reduce the number of required binary variables. This is expected to advance the overall efficiency in solving the MINLP as fewer decision variables have to be determined in the global optimization. The main difference is that in the formulation used in this work the tree is always fully constructed for a given number of pre-defined tree layers *NL*. The predictor variables and constants can only be assigned to the lowest layer. The inclusion of an identity function allows to pass up the values without any change to a higher layer in the expression trees.

In comparison to that, in Ref. [35] predictor variables and constants are available for selection at every node in the expression tree superseding an identity function. If those leaf-node operands are selected on a higher layer, their child nodes, as well as the other subsequent lower levels, are discarded. Hence, the tree is not set up necessarily to the maximum of allowed layers. By introducing an identity function in the modified version of the MINLP, as was proposed in the theoretical formulation of the problem in the first recorded publication in the open literature on the topic in chemical engineering [34], the binaries on branch nodes for $x_{i,j}$ and ε_n can be spared. With $N = 2^{NL} - 1 = N_b + N_l$ and $N_l = 2^{(NL-1)}$, Eqs. 33-35 specify the number of required binary variables (B) for the two MINLP formulations in order to quantify the effect in reduction and scaling behaviour:

New formulation
$$B_{New} = N_b \cdot NF + N_l \cdot NX + \frac{N_l}{2} \quad (33)$$

Formulation in [35]
$$B_{Cozad} = N_b \cdot (NF + NX + 1) + N_l \cdot (NX + 1) \quad (34)$$

Delta
$$\Delta B = B_{New} - B_{Cozad} = -N_b \cdot (NX + 1) - \frac{N_l}{2} \quad (35)$$

The delta proves that the significance of the reduction increases with the number of layers (NL) and the number of overall considered input variables (NX) (see SI for a quantitative comparison of scaling behaviour). Furthermore, the full construction of the tree allowed

replacing big-M constraints at the leaf-nodes by equality constraints due to linearity in the binary variables. Another main difference is the asymmetric supply of continuous variables as constants. This design reduces the symmetries in the superstructure set-up and realizes a reduction in the number of the required binaries as discussed above.

2.4. Solver

For the aims of this work, the choice of solvers is limited to those that can deterministically solve MINLPs to the global optimum. According to Ref. [38], the general list of feasible non-convex global MINLP solvers contains ANTIGONE [39], BARON [40], Couenne [41], LindoGlobal [42], and SCIP [43]. According to the results of the comparative solver study [35], BARON solves more SR problems and converges faster than all other solvers. Hence, BARON (v. 18.5.9), as a commercial general-purpose solver in deterministic global optimization, was selected in combination with IBM CPLEX [44] as a sub-solver. The optimization problem was set-up and passed on to the solver using Pyomo (v. 5.5) [45]. Upon solver completion, the optimization results were analyzed using Python 3.6.5, allowing to translate the optimal decision variable values into the corresponding algebraic equation. This model can then be evaluated at different inputs for prediction as well. For further mathematical simplification of the equation, Python's library for symbolic mathematics SymPy [46] was used.

2.5. Physical Model Selection

The SR is to be performed with noisy experimental data. Following a globally optimal approach targeting model accuracy exclusively (Eq. 1), errors are represented in the final model, what is also known as overfitting. Hence, methodological measures have to be included to restrict the influence of experiment errors on the final model and assure generalisation capabilities with low errors to unseen data. In case of SR, the errors in the training data are propagated through the expression tree and the selected operators apply to the data including errors. The limited robustness to noise is especially prevalent among SR due to its maximal flexibility in constructing free-form models [47].

To only extract the relevant terms describing the main signal and to preferably exclude the errors, the complexity of the final model is penalized. Model complexity is restricted to a threshold (C). The identity function does not add to the complexity of a model. Consequently,

the true complexity must be discounted by nodes with an identity function assigned. These complexity criteria are included as additional constraints in the MINLP (Eq. 36).

$$\sum_{n\in\mathcal{N}_b}\sum_{k\in\mathcal{F}\setminus\{\mathrm{id}\}}\delta_{n,k}+\sum_{n\in\mathcal{N}_l}\sum_{i\in\mathcal{X}}\zeta_{n,i}+\sum_{n\in\mathcal{N}_l^*}\kappa_n\ \leq C \tag{36}$$

By limiting the flexibility allowed, overfitting can be reduced and sparse models found. This also increases interpretability. Furthermore, this constraint filters mathematical invariants including more terms from the search space.

Next, it is proposed to identify a portfolio of the most accurate models with varying complexity (C) by solving multiple MINLPs in parallel to global optimality. In statistics, there are multiple information theoretic criterion for model selection, such as Akaike information criterion (AIC) [48], the Bayesian information criterion (BIC) [49], and others. However, the extrapolation ability is considered as one of the key advantages of the physical model. Consequently, in the method proposed one model is selected that is as sparse as possible to allow interpretation and knowledge extraction but is also as complex as necessary to describe the underlying physical system without overfitting. Hence, the portfolios of models are to be compared with regard to validation error, defined as the sum of squared differences between the predicted and the experimental values of the validation data set. Due to the growing flexibility, the training error is assumed to be the lowest for the model with the highest allowed complexity. Without requiring assumptions about the underlying true model, these can be compared quantitatively by a data set for validation to check for overfitting. With the purpose of also comparing extrapolation capabilities of the models, the validation data set is created by extracting the data points at extrema of the predictor variables. Finally, the model selection can be based on the lowest validation error which also determines the required model complexity. The framework is illustrated in Figure 3.

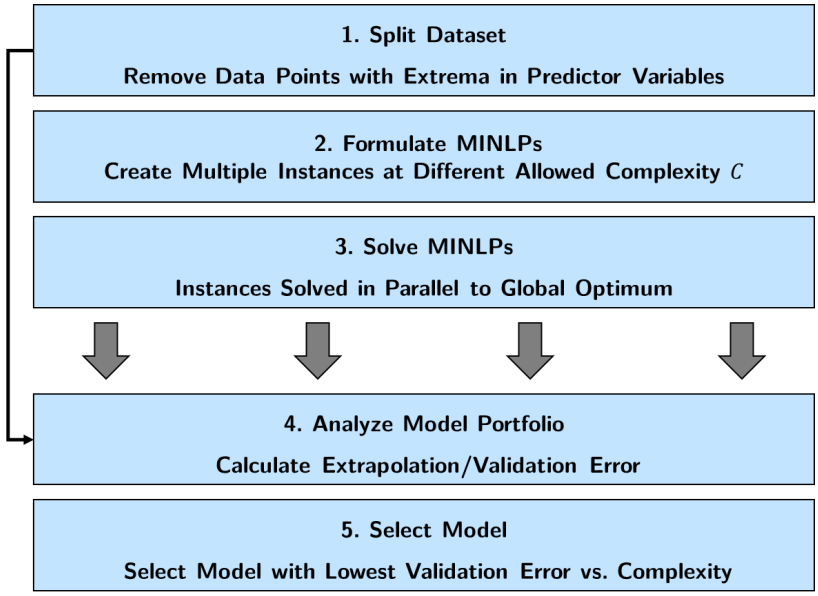


Figure 3. The proposed framework for the automated identification and selection of physical models *via* symbolic regression.

2.6. Experimental Procedures

Automated viscosity measurement

Samples of aqueous solutions were prepared as a batch of 14 samples by means of two R-Series pumping modules (Vapourtec Ltd.) coupled with a Gilson fraction collector, and controlled by a custom-written LabVIEW code. Carboxymethyl cellulose was pre-diluted in deionised water.

The viscosities of aqueous glycerol and carboxymethyl cellulose solutions were measured by means of the custom-built automated capillary viscometer shown schematically in Figure 4.

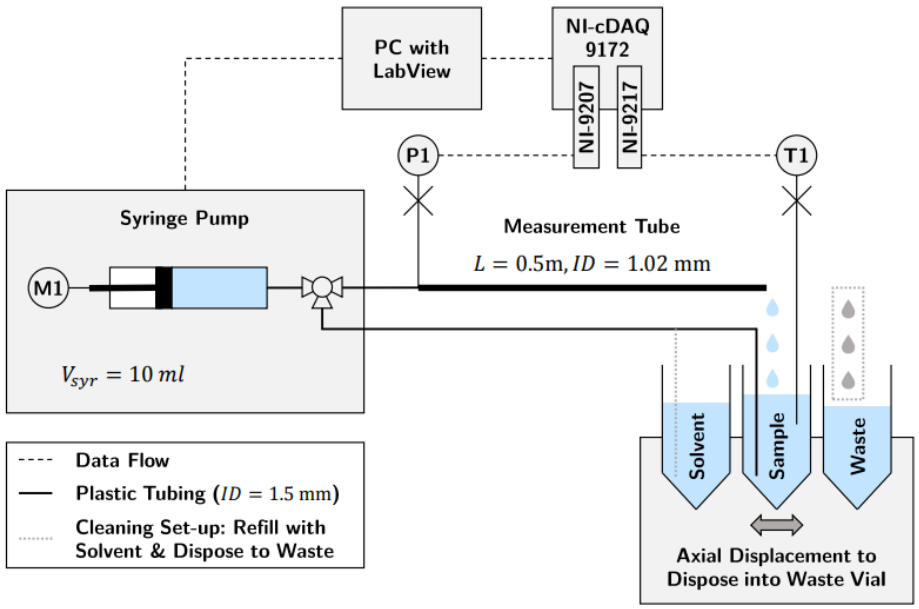


Figure. 4. Schematic diagram of the custom-built capillary viscometer. Solvent withdrawal and waste disposal for cleaning are indicated by dashed line. Vials are displaced axially for waste disposal.

To automatically control the system, a user interface was implemented in LabView 2014 (SP1), that allowed to visualize, analyse, and save the acquired signals. For the non-Newtonian solutions, a special mode was established to test multiple pre-defined flow rates in an automated sequence. A syringe pump (CETONI neMESYS 290N) was selected to ensure accurate and constant flows free from pulsation, and equipped with a 10 mL glass syringe (CETONI, ID 14.57 mm). The pump was connected to an automated three-port valve (CETONI valve, max. pressure 3 bar) to empty and re-fill the syringe using different routes of fluid flow. When the syringe was emptied, the valve directed the fluid through the stainless-steel tube (L = 0.5 m, ID = 1.02 mm, OD = 1.59 mm), placed horizontally at the same height as the syringe. The pressure drop relative to ambient pressure was measured using a microfluidic sensor (P1, Elveflow MPS3) with a range of 2 bar and accuracy of ±4 mbar. The pressure sensor signal was acquired via a voltage and current input module (National Instruments NI-9207). Temperature of the samples was constantly recorded by a resistance temperature detector (T1, Bearing Sensor Platinum, 100 Ω) and was found to be always in the range 21 ± 0.8 °C. The inlet line was automatically switched from the samples to the solvent, 2-propanol, and waste reservoirs for the cleaning and drying routines to prevent cross-contamination with the next sample. Each measurement was carried out in triplicate. The acquired time series data were processed to derive time-independent data series. This was done using an algorithmic scheme in Python to average only the steady-state pressure-drop signal values. The obtained flow rate (Q) – pressure drop (Δp) data can be used to calculate viscosity according to Hagen-Poiseuille law for Newtonian fluids (Eq. 37), and the Weissenberg-Rabinowitsch equation for non-Newtonian solutions (Eq. 38).

$$\eta = \frac{\pi R^4 \Delta p}{8\, Q\, L} \tag{37}$$

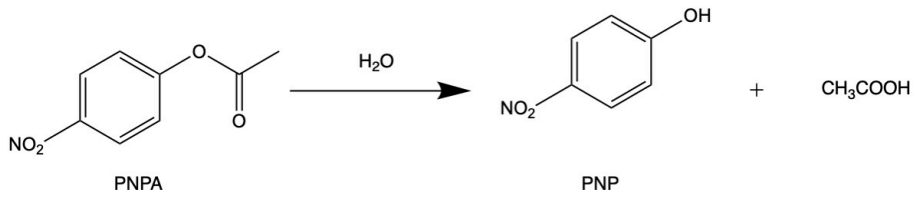
$$\eta = \frac{\pi R^4 \Delta p}{2\, Q\, L}\left(3 + \frac{d \ln Q}{d \ln \Delta p}\right)^{-1} \tag{38}$$

where R and L are the radius and the length of the adopted tubing, respectively.

Before taking advantage of the automated set-up, a calibration procedure was conducted to reduce the systematic measurements errors. This consisted of: (i) a pressure sensor calibration carried out using a digital pressure indicator (Druck, DPI 600/IS) applying a static pressure with air, and (ii) a full set-up calibration with aqueous glycerol solutions at known concentrations. In the latter case, viscosity of the same samples was also measured by a commercial rotational viscometer (Rheometric Scientific, ARES G2) in order to rule out the systematic error introduced by the automated set-up. Additional information is provided in the Electronic Supplementary Information (ESI).

Reaction kinetic experiments

Reaction kinetic data were collected for hydrolysis of para nitrophenyl acetate (PNPA) under basic conditions as a case study, Scheme 1. For each experimental run, three stock solutions were prepared consisting of PNPA (at the desired concentration) in 3.0 % (v/v) aqueous methanol, 3 mol·L^{-1} KCl, and an aqueous NaOH solution at a fixed pH. The adopted conditions for each kinetic experiment are provided in the ESI. 1 mL of each solution was directly mixed in a spectrophotometric agitated disposable cuvette. Absorption spectra (300-500 nm) were collected at fixed time intervals (Agilent, Cary 60). Absorption data at 400 nm were converted to PNP concentration. Calibration was carried out using different aqueous solutions at a known concentration of PNP at the same methanol and KCl concentrations and pH of the tested solutions. PNPA concentrations were calculated as its initial concentration minus the concentration of the formed product according to the literature [50,51], since no by-products formation was reported under the adopted conditions.



Scheme 1. Hydrolysis of para-nitrophenyl acetate (PNPA) case study reaction scheme.

2.7. Materials

All chemicals (glycerol ≥ 99.5%, isopropyl alcohol ≥ 99.5%, carboxymethyl cellulose, 4-nitrophenyl acetate (esterase substrate), 4-nitrophenol ≥ 99 %, potassium chloride ≥ 99.0%, sodium hydroxide ≥ 98.0 %, methanol ≥ 99.9 %) were purchased from Sigma Aldrich and used

as received. Water was obtained using a Maxima (USF) Milli-Q system. Viscosity experiments on a Newtonian fluid were carried out using a commercially available emulsion.

3. Results and Discussion

3.1. Method comparison: model identification from ideal data

In the first instance, the methodology described in Section 2.1. was applied to data without errors to assess global optimization in SR, gaining a deeper understanding of its performance, and comparing it with parametric regression. For reasons of simplicity a function with the same structure of Arrhenius law was considered, but without units and physically relevant parameters (Eq. 39). Arrhenius law is applicable in rheology as well as in reaction kinetics.

$$y = k \exp\left(\frac{\eta_0}{x}\right) \tag{39}$$

In our test case, $k = 3$, $\eta_0 = 8$, and $x = T$. 10 data points were randomly sampled in the x interval (10, 40). Unless specified otherwise, the following calculations were performed on an Intel® Core™ i5-3337U CPU @ 1.80 GHz processor.

The parametric method ALAMO (v. 2018.4.3), introduced earlier, was selected as the benchmark method due to its sparsity promoting techniques and openly accessible user-interface. For ALAMO, linear, exponential and constant basis functions were selected as well as discrete polynomial exponents up to third order. The modelling criterion was the corrected Akaike Information Criterion (AICc) [52]. For the dataset consisting of 10 training data points, ALAMO results in a polynomial equation of third order (Table 6, Figure 5). Even for noise-free data, only a surrogate model is found without any hints about the known underlying true relationship. This serves as an example for the limitations of parametric regression approaches in the discovery of true model structure. Nevertheless, it should be noted that the model was obtained during the first iteration in less than one second. This can be explained by the restricted search space and the high efficiency that can be achieved in solving parametric approaches.

Table 6. Comparison of ALAMO and symbolic regression proposed

Approach	Identified model	Sum of Squared Errors	Computational Time (s)

Parametric Regression ALAMO	$y = 12 - 0.81x + 0.028x^2 - 0.00032x^3$	0.04	<1S
Non-parametric Regression Symb. Reg. proposed	$y = \frac{e^{1.099}}{e^{\frac{-8.0}{x}}}$	1E-17	29

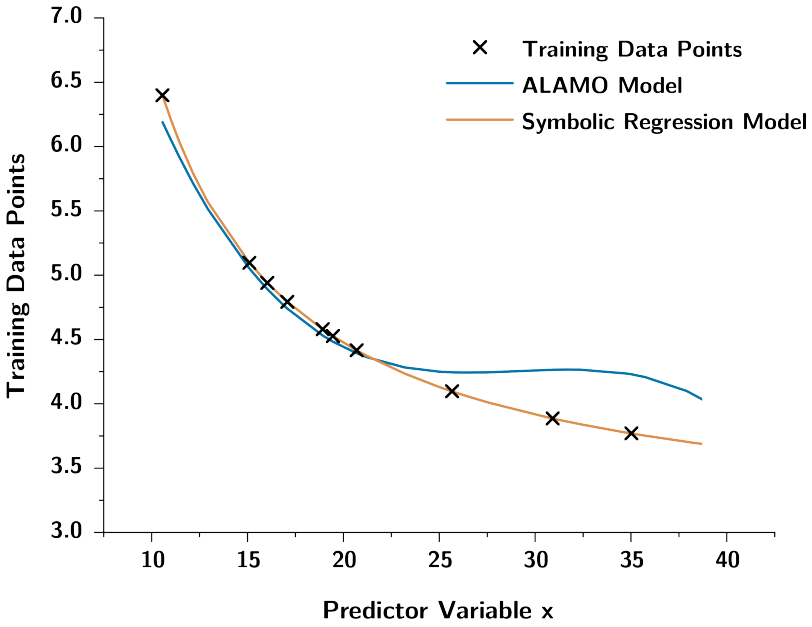


Figure 5. Graphical comparison of performance of ALAMO and the proposed symbolic regression in finding a model described by Eq. 27.

A schematic representation of a four-layers tree for Arrhenius law identification by SR is shown in Figure 6.

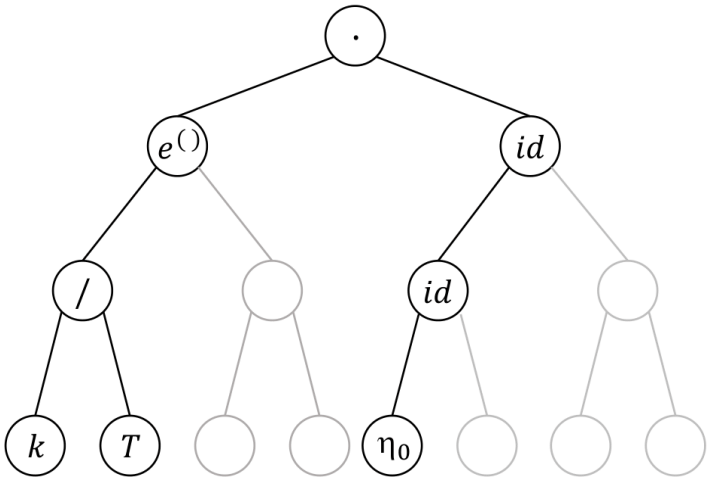


Figure 6. Binary expression tree for Arrhenius equation.

The resulting MINLP consisted of 54 binary and 154 continuous variables as well as 1174 constraints. The included operators were $\mathcal{F} = \{id, +, -, \cdot, /, exp\}$. The globally optimal model (Eq. 40) was found within 29 s. It is a mathematically invariant model of the true one as there is an unconstrained functional search space for symbolic regression.

$$y = \frac{\exp(1.099)}{\exp(-\frac{8.0}{x})} = 3.0 \exp\left(\frac{8.0}{x}\right) \tag{40}$$

In comparison to ALAMO, the SR solved to global optimality identified the true underlying model which allows significantly better generalization than the previously found surrogate model. Another advantage that can be stressed is the sparse dataset used in training. However, the structure identification without *a priori* knowledge comes at computational cost requiring more time to converge. It is worth mentioning that the choice of the basis set of functions is crucial in both cases. However, the presented DR method is able to construct more complex functions with a smaller number of basis functions.

In the second instance, the function shown in Eq. 41 was adapted from [35] to evaluate the impact of the modifications in the MINLP formulation (Section 2.1) in terms of CPU time until convergence.

$$y = \frac{2x_1}{5 - x_2} \tag{41}$$

In comparison to the previously reported formulation [35], the influence of the number of included operators (NF), data points (NE), predictor variables (NX), and tree layers (NL) is studied. Figure 7 summarises the obtained results.

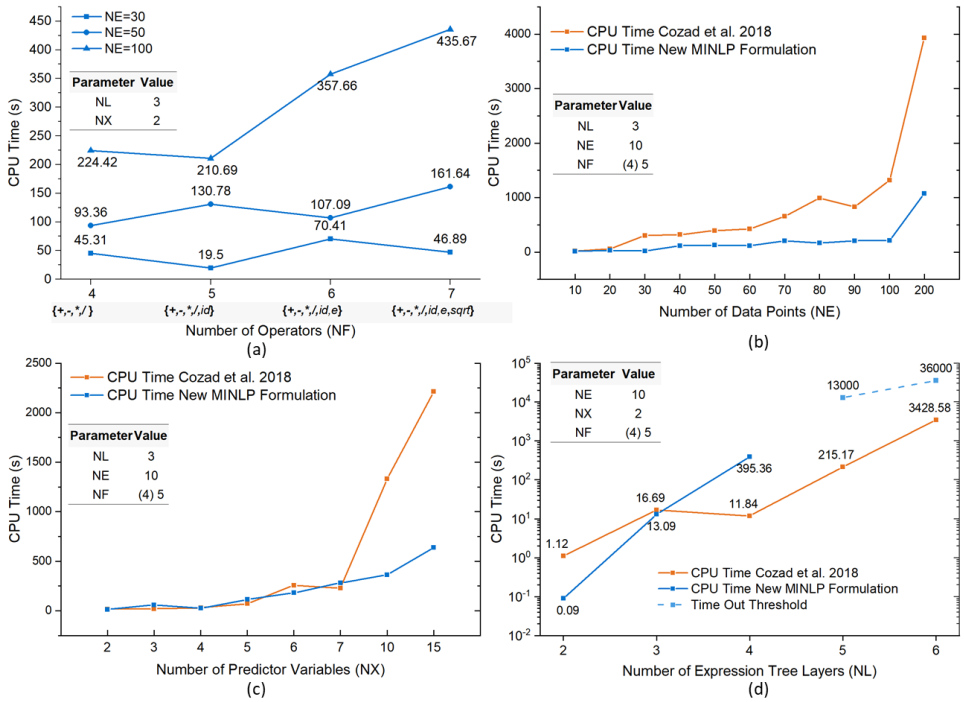


Figure 7. Scalability of the proposed methodology: (a) scalability in the number of included operators; (b) scalability in the number of data points; (c) scalability in the number of prediction variables; (d) scalability in the number of expression tree layers.

The scale-up with regard to an increasing number of potential operators was studied first (Figure 7a). A comparison with the previously published work of [35] is not provided as, for algebraic operators, there is neither a conceptual nor a mathematical difference in the two formulations. Since the number of binary variables and big-M constraints increases linearly with the number of operators, a growth in the computational time is expected. The results obtained for $NE = 30$ and $NE = 50$ did not show a consistent linear growth in CPU time. An overall linear trend became apparent when using an increasing number of data points.

The number of data points in the training set affects the number of nodal values and constraints in a linear fashion. The results obtained at different NE are summarised in Figure 7b and compared with the previously reported formulation [35]. For both formulations an increase in CPU time was observed, and in all cases, our adapted formulation showed improved scalability in the number of data points. We should point out that for the results in Figure 7, a single run was used, whereas the required CPU time may slightly change if this measure is averaged over

a number of runs. We do not anticipate a change that would affect conclusions in a significant way.

As described in Section 2.3, the conceptual difference in the modified formulation allows to reduce the number of binary variables which are required to allocate the predictor variables in the tree. As a result, a difference in performance should be observed when increasing the number of predictor variables. The expected improvements became evident at higher quantities of potential predictors and are shown in Figure 7c.

As the last parameter, the influence of the number of allowed layers in the expression tree was considered. By growing the tree in terms of the number of layers, an exponentially increasing number of nodes is added. Accordingly, the number of variables and constraints increases exponentially. As a result, the increase in CPU time is also exponential, as shown in Figure 7d. In the case of layer-scalability, the previous MINLP formulation is superior. The new proposed formulation timed out after a few hours for a number of layers greater than four. For the function under study a three-layered tree was sufficient and the ability to discard sub-layers is favoured over constructing the whole tree with identity functions. This advantage might diminish if more complex functions are sought within higher layered trees. Applicable to both formulations, this result confirms the high computational expense of SR due to the combinatorial search space. The exponential scale-up behaviour in tree layers could strongly limit the method's ability to identify more complex models. A limited investigation of the effect of the time-out time limitation for the larger trees suggests that a hyper-exponential behaviour may also be possible.

While at present the obtained results show that the methodology can be applied to relatively simple models (see further results below) due to the issue of scalability, this issue is largely due to the specifics of the solver, and are not insurmountable. Certainly, further results reported below show that the approach is indeed suitable to identification of models from noisy experimental data, which supports our primary hypothesis. A more detailed study of the scalability to more complex models is the subject of further study, which will be enabled by a significantly faster optimisation routine. An overview of the number of binaries, continuous decision variables as well as MINLP constraints for all the simulations is reported in the Supplementary Information.

3.2. Newton's Law of Viscosity

The data collected from a commercially available emulsion sample by means of the automated capillary viscometer were used to identify the simple linear relationship between shear stress (τ_w) and shear rate ($\dot{\gamma}_w$) at the wall of the tubing (Eq. 42).

$$\tau_w = \eta\, \dot{\gamma}_w \tag{42}$$

where

$$\tau_w = \frac{\Delta p\, R}{2\, L} \tag{43}$$

$$\dot{\gamma}_w = \frac{4\, Q}{\pi\, R^3} \tag{44}$$

For the parameter identification an expression tree with three layers, including the shear rate ($\dot{\gamma}_w$) as the only predictor variable, and ten experimental data points were used. The two extrema data points of the shear rate were not included in the training set and were used for the calculation of the extrapolation error. The set of operators included the basic operators and a power law $\mathcal{F} = \{id, +, -, \cdot, /, \hat{}\}$. The resulting MINLP consisted of 28 binary and 58 continuous variables and 434 constraints. Five different instances of different complexities $C = \{3,4,5,6,7\}$ were solved in parallel.

The obtained portfolio of models, Table 7, initially consisted of five models. As the complexity is constrained by an upper bound (inequality), similar models with the same complexity are identified. These, together with invariant models at higher complexities, were neglected. The shear rate and the shear stress were expressed as s^{-1} and Pa, respectively.

Table 7. Physical model identification: Newton's law of viscosity.

Model Complexity	Identified model	Training error	Extrapolation error	Computational time (s)
3	$\tau = (0.106 \pm 0.00042)\dot{\gamma}$	0.644	0.109	112
5	$\tau = ((0.099 \pm 0.000396)\dot{\gamma})^{(1.023\pm0.00102)}$	0.483	0.197	2926
7	$\tau = \dot{\gamma}^{(0.213\pm0.00053)\dot{\gamma}^{(0.188\pm0.00047)}}$	0.343	2.44	1139

To choose the best model among the identified ones, the prediction of the models was plotted together with the experimental data, see Figure 8, and the training and extrapolation errors were compared. In all cases, the error reported is the squared error, defined as the sum of the squared differences between the predicted and experimental values for each data set. As expected, the training error decreases with the complexity of the model as there is more flexibility allowed to SR. However, the comparison of the extrapolation errors shows the superiority of Newton's law model ($C = 3$), whereas the other identified models suffer from overfitting. Overall, the Newton's law model can be selected as the sparsest model with the highest generalisation capability, and can be easily interpreted to generate knowledge about the physics of the system under investigation.

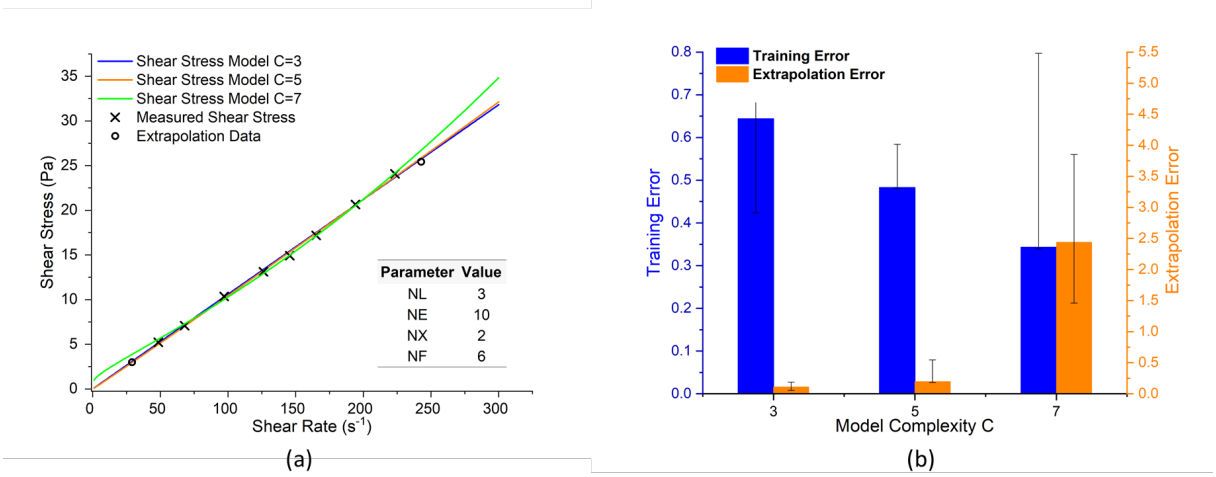


Figure 8. Physical model selection for Newtonian power law: (a) measured shear stress and shear rate: data used for model training; (b) errors in training and extrapolation data set for model identification.

The model identification was conducted with only 10 data points, highlighting the sparsity of required data in the presented method compared to other data-driven methods. This is especially beneficial in chemistry, where data points can be expensive to generate.

3.3. Non-Newtonian Power Law

Identification of a non-Newtonian power law was used to prove that the model selection framework favours higher complexity models where required. Eleven experimental data points were collected using 1% w/w aqueous carboxymethyl cellulose at different flow rates. As for the Newton's law identification, an expression tree with three layers was used, including the

shear rate ($\dot{\gamma}_w$) as the only predictor variable. Seven different instances of different complexities $C = \{1,2,3,4,5,6,7\}$ were solved in parallel. The resulting MINLPs have 24 binary and 65 continuous variables, and 486 constraints. The data were pre-processed, scaling down the apparent shear rate (Eq. 44) by a factor of 10 before training. This allowed to keep the variable bounds and big-M values calculated by interval arithmetic low, reducing the overall search space of the solver, especially in the cases of including the power law operations.

With these settings, the portfolio of three models, Table 8, was obtained within 13 min. The mathematically invariant and similar models were discarded. The models with complexities $C = 2$, $C = 4$, and $C = 6,7$ were invariant to the models with complexities $C = 1$, $C = 3$, and $C = 5$, respectively. The resulting portfolio consists of three different models of different complexity.

Table 8. Physical model identification: non-Newtonian power law.

Model Complexity	Identified model	Training error	Extrapolation error	Computational time (s)
1	$\tau = \dot{\gamma}$	242.3	16.02	60
3	$\tau = (4.448 \pm 0.3425) + \dot{\gamma}$	12.19	9.30	314.4
5	$\tau = (0.712 \pm 0.0044)\dot{\gamma}^{(0.671 \pm 0.00117)}$	1.63	0.81	758.5

It is noteworthy that all the computed models can be physically interpreted as the ones describing Newtonian fluids, Bingham fluids and non-Newtonian (power law) fluids. The model selection was carried out comparing their training and validation errors. Once again, two experimental data points at the edges of the investigated range of shear rates were taken aside and used to evaluate the extrapolation performance of the obtained models. The three candidate models together with their performance on the training and validation data are shown in Figure 9. In this case, both the training and the extrapolation errors decrease with complexity, indicating that the most complex power law is the most appropriate for the description of the experimental data.

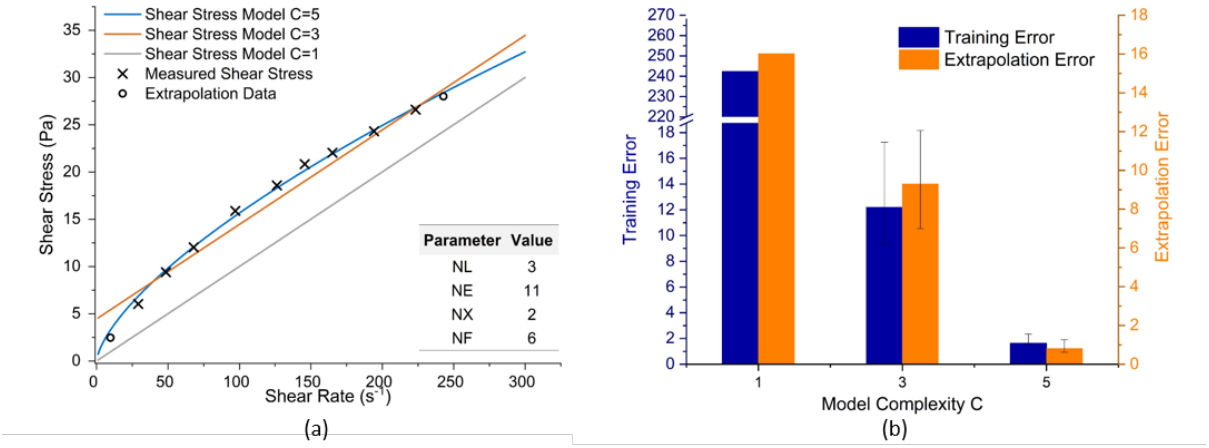


Figure 9. Physical model selection for non-Newtonian power law: (a) measured shear stress and shear rate: data used for model training; (b) errors in training and extrapolation data set for model identification.

3.4. First-order Kinetic Law

Previous examples show the potential of the adopted methodology to discover sparse and interpretable models to describe the viscous behaviour of different fluids with a limited amount of experimental data. Moreover, a simple procedure was proven to be effective to select the most appropriate model within the obtained portfolio. In the following, the same procedure was applied to learn a kinetic model of a simple test reaction for which a large amount of experimental data was available.

According to literature [53], hydrolysis of carboxylic acid esters can be described by first order kinetic law, Eq. 45.

$$r = k_h[PNPA] \tag{45}$$

where $[PNPA]$ is the molar concentration of the investigated compound (para-nitrophenyl acetate) and the kinetic constant k_h can be expressed as shown in Eq. 46.

$$k_h = k_N + k_A[H^+] + k_B[OH^-] \tag{46}$$

Under the adopted experimental conditions (pH > 10.52) the terms k_N and $k_A[H^+]$ are negligible and the overall kinetic law is given by Eq. 47.

$$r = k_B[OH^-]\,[PNPA] \tag{47}$$

In the first attempt, experimental data were collected at a fixed pH of 10.52, at different PNPA concentrations. Concentrations *vs* time data series were pre-processed to obtain an approximation of the reaction rate over time using the centered difference approximation.

For this example, a three-layer tree structure was allowed, including $[PNPA]$ as the only predictor variable. Due to the relatively low values of the measured reaction rates (10^{-7} – 10^{-9} mol·L^{-1}·s^{-1}), they were expressed as mmol·L^{-1}·h^{-1}.

Five MINLP instances were solved $C = \{3,4,5,6,7\}$ in parallel. The resulting MINLPs have 21 binary and 219 continuous variables and 1354 constraints. 39 experimental data points were split into 31 training examples and 8 validation data points in the range $[PNPA] \in (1.11 \cdot 10^{-3}$ – $4.63 \cdot 10^{-2}$ mmol·L^{-1}). The validation points were chosen at the lower/upper end of the dataset in order to test the extrapolation ability of the model.

The model portfolio without doubling is summarized in Table 9, and the errors are shown in Figure 10. The identified model with the lower extrapolation error is the true underlying first-order kinetic law governing the physics of the chemical system.

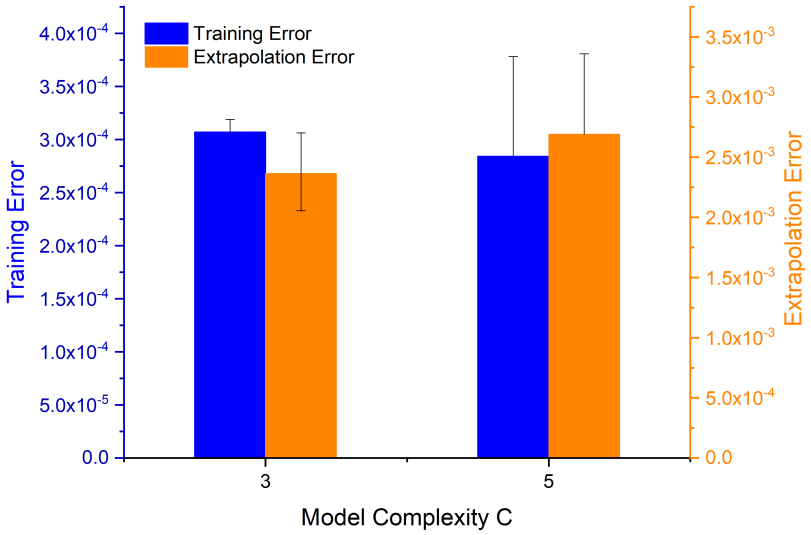


Figure 10. Physical model selection for first order kinetic law identification: Errors in training and extrapolation data set for model identification.

Table 9. Physical model identification: First order kinetic law.

Model Complexity	Identified model	Training error	Extrapolation error	Computational Time (s)
3	$r = (8.49 \pm 0.042)[PNPA]$	$3.070 \cdot 10^{-4}$	$2.364 \cdot 10^{-3}$	62.6
5	$r = (8.42 \pm 0.051)[PNPA] + (0.00133 \pm 0.00109)$	$2.840 \cdot 10^{-4}$	$2.692 \cdot 10^{-3}$	84.3

3.5. First-order Kinetic Law: dependence on pH

In the second attempt experimental data collected at different pH were included in the training algorithm to identify the dependence of the kinetic constant on the $[OH^-]$ concentration. The training data set consisted of 80 experimental data obtained varying $[PNPA]$ and $[OH^-]$ in the ranges $5.04 \cdot 10^{-4}$ – $4.55 \cdot 10^{-2}$ mmol·L^{-1} and $3.33 \cdot 10^{-1}$ – 4.33 mmol·L^{-1}, respectively.

An expression tree with three layers was used again. The set of operators included the basic operators and a power law $\mathcal{F} = \{id, +, -, \cdot, /, \hat{}\}$. The resulting MINLP consisted of 25 binary and 450 continuous variables, and 2973 constraints. Five different instances were solved in parallel of different complexities $C = \{3,4,5,6,7\}$. 80 experimental data points were split into 80 % training examples and 20 % validation data in the ranges $[PNPA] \in (5.00 \cdot 10^{-4}$ – $4.63 \cdot 10^{-2}$ mmol·L^{-1}) and $[OH] \in (3.33 \cdot 10^{-1}$ – $4.33 \cdot 10^{-2}$ mmol·L^{-1}). The validation points were chosen at the lower/upper end of the dataset in order to test the extrapolation ability of the model. Reaction rates were expressed as mmol·L^{-1}·h^{-1}.

According to the examples reported in Sections 3.4, invariant models were obtained for C = 4 and 6, and discarded. The obtained portfolio of models is summarized in Table 10 and errors - in Figure 11.

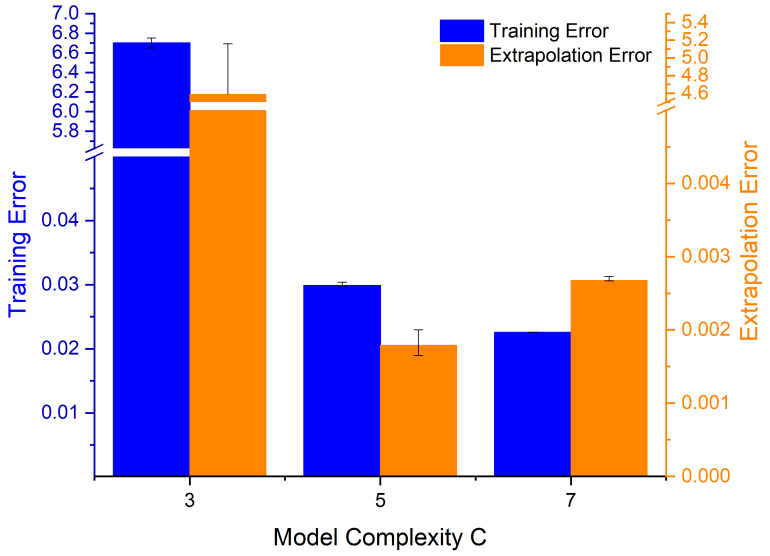


Figure 11. Physical model selection for pH-dependent kinetic law identification: Errors in training and extrapolation data set for model identification.

Table 10. Physical model identification: kinetics dependence on pH.

Model Complexity	Identified model	Training error	Extrapolation error	Computational Time (s)
3	$r = (33.2 \pm 1.7)[PNPA]$	6.70	4.59	241
5	$r = (26.6 \pm 0.08)[PNPA][OH^-]$	$2.99 \cdot 10^{-2}$	$1.79 \cdot 10^{-3}$	226
7	$r = (24.7 \pm 0.074)[OH^-])[PNPA][OH^-]$	$2.23 \cdot 10^{-2}$	$2.68 \cdot 10^{-3}$	4287

As shown in Figure 11, both errors are of 2 or 3 orders of magnitude higher when $C = 3$, whereas similar training errors were obtained when $C = 5$ and $C = 7$. However, the lowest extrapolation error suggests that the model with complexity $C = 5$ is the most suitable one for the description of the kinetic behaviour of the system.

4. Conclusions

Based on a MINLP formulation for global SR reported in the mathematical domain, a different approach in setting up the superstructure was introduced to reduce the number of binary

variables involved in globally optimal SR. In addition, this formulation is complemented with a framework to enable the automated identification of physical models from crude data.

The new approach was found to outperform the previously proposed ones in terms of computational time when increasing the number of included operators, predictor variable and experimental data. As examples, the developed method allowed to correctly identify the models underlying the rheological behaviour of Newtonian and non-Newtonian fluids, as well as simple kinetic laws, also in the case of sparse data sets, which is a common scenario in chemical process development.

A significant limitation of the methodology was found in the exponential, or even potentially hyper-exponential, scale-up of the computational time for an increasing number of adopted layers in the tree necessary to represent complex algebraic structures of analytical function type models. The issue of computational efficiency cannot be resolved by parallelization alone. This presently limits the approach to the identification of relatively simple models, which may still find significant applications in engineering, when approximate 'apparent' models are desired.

Work is currently underway to overcome these limitations using rigorous mathematical programming approaches, such as the one presented in this work, as well as complementary methodologies to derive globally optimal fitted model structures.

Acknowledgements

PN is grateful for his Erasmus funding received for the exchange between RWTH Aachen University and the University of Cambridge and that the exchange programme is co-funded by the Department of Chemical Engineering and Biotechnology, University of Cambridge, and Sustainable Reaction Engineering group of Prof. A. Lapkin. LC is grateful to BASF for co-funding her PhD. This project is also co-funded by the UKRI project "Combining Chemical Robotics and Statistical Methods to Discover Complex Functional Products" (EP/R009902/1), and co-funded by the National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program as a part of the Cambridge Centre for Advanced Research and Education in Singapore Ltd (CARES).

References

[1] C.W. Coley, R. Barzilay, T.S. Jaakkola, W.H. Green, K.F. Jensen, Prediction of Organic Reaction Outcomes Using Machine Learning, ACS Cent. Sci. 3 (2017) 434–443. doi:10.1021/acscentsci.7b00064.

[2] A. Echtermeyer, Y. Amar, J. Zakrzewski, A. Lapkin, Self-optimisation and model-based design of experiments for developing a C–H activation flow process, Beilstein J. Org. Chem. 13 (2017) 150–163. doi:10.3762/bjoc.13.18.

[3] H. Gao, T.J. Struble, C.W. Coley, Y. Wang, W.H. Green, K.F. Jensen, Using Machine Learning To Predict Suitable Conditions for Organic Reactions, ACS Cent. Sci. 4 (2018) 1465–1476. doi:10.1021/acscentsci.8b00357.

[4] M.I. Jeraal, N. Holmes, G.R. Akien, R.A. Bourne, Enhanced process development using automated continuous reactors by self-optimisation algorithms and statistical empirical modelling, Tetrahedron. 74 (2018) 3158–3164. doi:10.1016/J.TET.2018.02.061.

[5] C.W. Coley, W. Jin, L. Rogers, T.F. Jamison, T.S. Jaakkola, W.H. Green, R. Barzilay, K.F. Jensen, A graph-convolutional neural network model for the prediction of chemical reactivity, Chem. Sci. 10 (2019) 370–377. doi:10.1039/C8SC04228D.

[6] V. Duros, J. Grizou, W. Xuan, Z. Hosni, D.L. Long, H.N. Miras, L. Cronin, Human versus Robots in the Discovery and Crystallization of Gigantic Polyoxometalates, Angew. Chemie - Int. Ed. 56 (2017) 10815–10820. doi:10.1002/anie.201705721.

[7] G. Skoraczyński, P. Dittwald, B. Miasojedow, S. Szymkuć, E.P. Gajewska, B.A. Grzybowski, A. Gambin, Predicting the outcomes of organic reactions via machine learning: are current descriptors sufficient? Sci. Rep. 7 (2017) 3582. doi:10.1038/s41598-017-02303-0.

[8] B.J. Reizman, K.F. Jensen, Feedback in Flow for Accelerated Reaction Development, Acc. Chem. Res. 49 (2016) 1786–1796. doi:10.1021/acs.accounts.6b00261.

[9] C. Houben, A.A. Lapkin, Automatic discovery and optimization of chemical processes, Curr. Opin. Chem. Eng. 9 (2015) 1–7. doi:10.1016/J.COCHE.2015.07.001.

[10] A.A. Lapkin, P.K. Plucinski, Engineering Factors for Efficient Flow Processes in Chemical Industries, in: Chemical Reactions and Processes under Flow Conditions. Eds. S.V. Luis, E. Garcia-Verdugo, Royal Society of Chemistry, Cambridge, 2009: pp. 1–43. doi:10.1039/9781847559739-00001.

[11] C.J. Richmond, H.N. Miras, A.R. de la Oliva, H. Zang, V. Sans, L. Paramonov, C. Makatsoris, R. Inglis, E.K. Brechin, D.-L. Long, L. Cronin, A flow-system array for the discovery and scale up of inorganic clusters, Nat. Chem. 4 (2012) 1037–1043.

doi:10.1038/nchem.1489.
[12] D.W. Robbins, J.F. Hartwig, D.W.C. MacMillan, A simple, multidimensional approach to high-throughput discovery of catalytic reactions., Science. 333 (2011) 1423–7. doi:10.1126/science.1207922.
[13] A.A. Lapkin, P.K. Heer, P.-M. Jacob, M. Hutchby, W. Cunningham, S.D. Bull, M.G. Davidson, Automation of route identification and optimisation based on data-mining and chemical intuition, Faraday Discuss. 202 (2017) 483–496. doi:10.1039/C7FD00073A.
[14] J.P. McMullen, M.T. Stone, S.L. Buchwald, K.F. Jensen, An Integrated Microreactor System for Self-Optimization of a Heck Reaction: From Micro- to Mesoscale Flow Systems, Angew. Chemie Int. Ed. 49 (2010) 7076–7080. doi:10.1002/anie.201002590.
[15] D.P. Solomatine, A. Ostfeld, Data-driven modelling: some past experiences and new approaches, J. Hydroinformatics. 10 (2008) 3–22. doi:10.2166/hydro.2008.015.
[16] M. Schmidt, H. Lipson, Distilling Free-Form Natural Laws from Experimental Data, Science 324 (2009) 81-85.
[17] O. Wolkenhauer, Why model?, Front. Physiol. 5 (2014) 21. doi:10.3389/fphys.2014.00021.
[18] C.S. Horbaczewskyj, C.E. Willans, A.A. Lapkin, R.A. Bourne, An introduction to closed-loop process optimization and online analysis, in "Green Chemical Engineering", Ed. A. Lapkin, Wiley-VCH, Weinheim, 2018.
[19] A. A. Lapkin, A. Voutchkova, P. Anastas, A conceptual framework for description of complexity in intensive chemical processes, Chem. Eng. Process. Process Intensif. 50 (2011) 1027–1034. doi:10.1016/j.cep.2011.06.005.
[20] A. Cozad, N. V. Sahinidis, D.C. Miller, Learning surrogate models for simulation-based optimization, AIChE J. 60 (2014) 2211–2227. doi:10.1002/aic.14418.
[21] A. Cozad, N. V. Sahinidis, D.C. Miller, A combined first-principles and data-driven approach to model building, Comput. Chem. Eng. 73 (2015) 116–127. doi:10.1016/J.COMPCHEMENG.2014.11.010.
[22] Z.T. Wilson, N. V. Sahinidis, The ALAMO approach to machine learning, Comput. Chem. Eng. 106 (2017) 785–795. doi:10.1016/J.COMPCHEMENG.2017.02.010.
[23] N.M. Mangan, J.N. Kutz, S.L. Brunton, J.L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, Proc. R. Soc. A 473 (2017) 0009. doi:10.1098/rspa.2017.0009.
[24] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial

differential equations, Sci. Adv. 3 (2017) e1602614. doi:10.1126/sciadv.1602614.
[25] A. Narasingam, J. Sang-Il Kwon, Data-driven identification of interpretable reduced-order models using sparse regression, Comput. Chem. Eng. 119 (2018) 101–111. doi:10.1016/j.compchemeng.2018.08.010.
[26] H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization, Proc. R. Soc. A. 473 (2017) 0446. doi:10.1098/rspa.2016.0446.
[27] H. Schaeffer, G. Tran, R. Ward, L. Zhang, Extracting structured dynamical systems using sparse optimization with very few samples, https://github.com/linanzhang/SparseCyclicRecovery. (accessed August 5, 2019).
[28] S. Li, E. Kaiser, S. Laima, H. Li, S.L. Brunton, J. Nathan Kutz, Discovering time-varying aeroelastic models of a long-span suspension bridge from field measurements by sparse identification of nonlinear dynamical systems, https://arxiv.org/pdf/1809.05707.pdf (accessed August 5, 2019).
[29] G. Lin, S. Zhang, Robust data-driven discovery of governing physical laws with error bars, Proc. R. Soc. A. 474 (2018) 0305. doi:10.1098/rspa.2018.0305.
[30] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems., Proc. Natl. Acad. Sci. U. S. A. 113 (2016) 3932–7. doi:10.1073/pnas.1517384113.
[31] J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems, Proc. Nat. Acad. Sci., 104 (2007) 9943-9948.
[32] B. Tarawneh, W. AL Bodour, K. Al Ajmi, Intelligent Computing Based Formulas to Predict the Settlement of Shallow Foundations on Cohesionless Soils, Open Civ. Eng. J. 13 (2019) 1–9. doi:10.2174/1874149501913010001.
[33] Y. Wang, N. Wagner, J.M. Rondinelli, Symbolic regression in materials science, https://arxiv.org/pdf/1901.04136.pdf (accessed January 22, 2019).
[34] V.S. Vassiliadis, Y. Wang, H. Arellano-Garcia, Y. Yuan, A Novel Rigorous Mathematical Programming Approach to Construct Phenomenological Models, Comput. Aided Chem. Eng. 37 (2015) 707–712. doi:10.1016/B978-0-444-63578-5.50113-4.
[35] A. Cozad, N. V. Sahinidis, A global MINLP approach to symbolic regression, Math. Program. (2018) 1–23. doi:10.1007/s10107-018-1289-x.
[36] E.R. Gansner, S.C. North, K.P. Vo, DAG—a program that draws directed graphs, Softw. Pract. Exp. 18 (1988) 1047–1062. doi:10.1002/spe.4380181104.
[37] I.E. Grossmann, Review of Nonlinear Mixed-Integer and Disjunctive Programming

Techniques, Optim. Eng. 3 (2002) 227–252. doi:10.1023/A:1021039126272.
[38] J. Kronqvist, D. E. Bernal, A. Lundell, I. E. Grossmann, A review and comparison of solvers for convex MINLP, 20 (2019) 397–455. doi:10.1007/s11081-018-9411-8.
[39] R. Misener, C.A. Floudas, ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations, J. Glob. Optim. 59 (2014) 503–526. doi:10.1007/s10898-014-0166-2.
[40] M.R. Kılınç, N. V. Sahinidis, Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON, Optim. Methods Softw. 33 (2018) 540–562. doi:10.1080/10556788.2017.1350178.
[41] P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wächter, Branching and bounds tightening techniques for non-convex MINLP, Optim. Methods Softw. 24 (2009) 597–634. doi:10.1080/10556780903087124.
[42] Y. Lin, L. Schrage, The global solver in the LINDO API, Optim. Methods Softw. 24 (2009) 657–668. doi:10.1080/10556780902753221.
[43] S. Vigerske, A. Gleixner, SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework, Optim. Methods Softw. 33 (2018) 563–593. doi:10.1080/10556788.2017.1335312.
[44] ILOG CPLEX Optimization Studio - Overview | IBM, www.cplex.com (accessed August 6, 2019).
[45] W.E. Hart, J.-P. Watson, D.L. Woodruff, Pyomo: modeling and solving mathematical programs in Python, Math. Program. Comput. 3 (2011) 219–260. doi:10.1007/s12532-011-0026-8.
[46] A. Meurer, C.P. Smith, M. Paprocki, O. Čertík, S.B. Kirpichev, M. Rocklin, Am. Kumar, S. Ivanov, J.K. Moore, S. Singh, T. Rathnayake, S. Vig, B.E. Granger, R.P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M.J. Curry, A.R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz, SymPy: symbolic computing in Python, PeerJ Comput. Sci. 3 (2017) e103. doi:10.7717/peerj-cs.103.
[47] M. Quade, M. Abel, K. Shafi, R.K. Niven, B.R. Noack, Prediction of dynamical systems by symbolic regression, Phys. Rev. E. 94 (2016) 12214. doi:10.1103/PhysRevE.94.012214.
[48] H. Akaike, A New Look at the Statistical Model Identification, in: Selected Papers of Hirotugu Akaike, Springer, New York, NY, 1974: pp. 215–222. doi:10.1007/978-1-4612-1694-0_16.

[49] S.I. Vrieze, Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC)., Psychol. Methods. 17 (2012) 228–243. doi:10.1037/a0027127.
[50] J. Klausen, M.A. Meier, R.P. Schwarzenbach, Assessing the Fate of Organic Contaminants in Aquatic Environments: Mechanism and Kinetics of Hydrolysis of a Carboxylic Ester, J. Chem. Educ., 74 (1997) 1440-1444.
[51] P. S. Marrs, Class Projects in Physical Organic Chemistry: The Hydrolysis of Aspirin, J. Chem. Educ. 81 (2004) 870–873. www.JCE.DivCHED.org (accessed April 9, 2019).
[52] A. Cozad, N. V. Sahinidis, D.C. Miller, Learning surrogate models for simulation-based optimization, AIChE J. 60 (2014) 2211–2227. doi:10.1002/aic.14418.
[53] H.J. Goren, M. Fridkin, The Hydrolysis of p-Nitrophenylacetate in Water. Mechanism and Method of Measurement, Eur. J. Biochem. 41 (1974) 263–272. doi:10.1111/j.1432-1033.1974.tb03267.x.

Supplementary material

A new formulation for symbolic regression to identify physico-chemical laws from experimental data

Pascal Neumann,[a,b] Liwei Cao,[b,c] Danilo Russo,[b] Vassilios S. Vassiliadis,[b] Alexei A. Lapkin[b,c1]

[a]Aachener Verfahrenstchnik – Process Systems Engineering, RWTH Aachen University, Aachen, Germany
[b]Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge, CB3 0AS
[c]Cambridge Centre for Advanced Research and Education in Singapore, CARES Ltd.
1 CREATE Way, CREATE Tower #05-05, 138602 Singapore

Viscometer Automation and Calibration

Figure S1 shows the developed graphical user interphase in LabView. It implements all the main functions including the manual and automated control of the syringe pump, data acquisition and saving into a comma delimited file. The collected data of temperature, pressure and volumetric flow rate are acquired at a frequency of 1500 Hz with the described National Instruments modules and is then averaged to reduce random errors. For calibration purposes, the Hagen-Poiseuille law for Newtonian liquids and the empirical viscosity models for the viscosity of glycerol-water mixtures at varying weight contents were implemented. Moreover, the functionalities to measure the viscosity at different shear rates automatically and the drying procedure after the cleaning with isopropanol were designated in the interface [1,2].

[1] Corresponding author. A. Lapkin email: aal35@cam.ac.uk

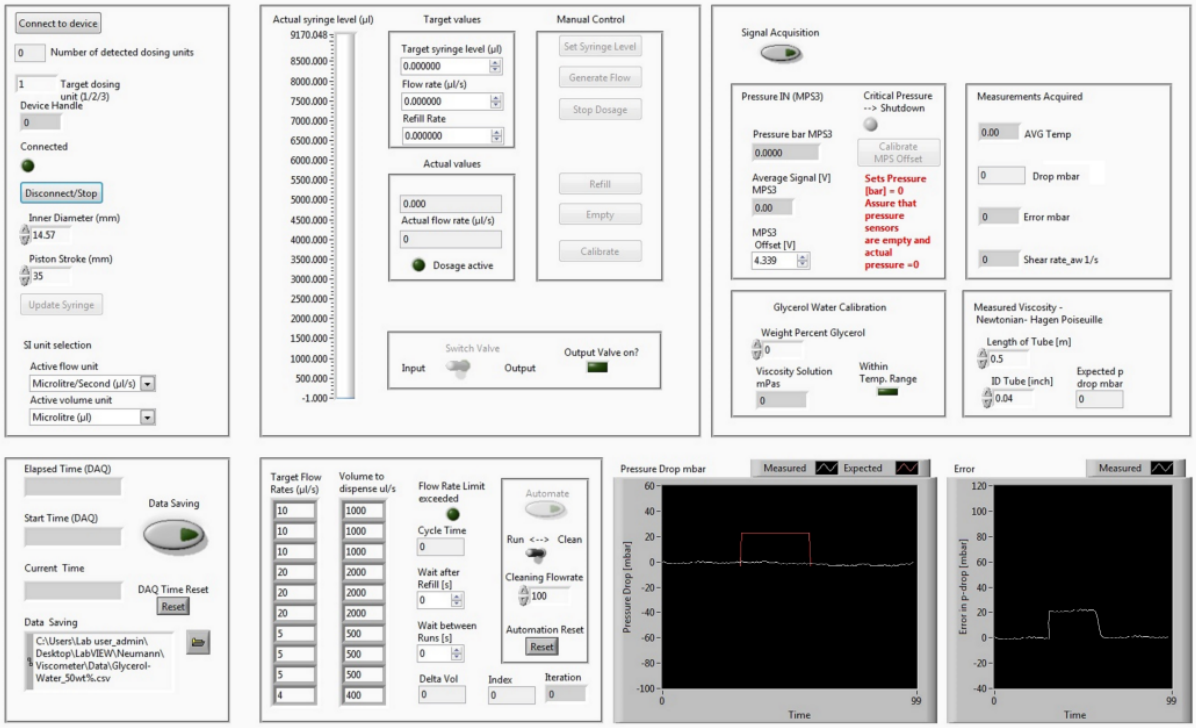


Figure S1. LabView front panel – GUI for the automated operation of the capillary viscometer.

Before the whole set-up was calibrated with the glycerol-water mixtures, the pressure sensor (Elveflow MPS3) was calibrated in combination with the NI-9702 module. This was necessary as the supplier calibration is based on their own Elveflow amplification and data acquisition system which was not available for the set-up. The sensor was calibrated with a static air pressure by closing the sensor outlet and applying a known pressure with the DRUCK digital pressure indicator. The voltage signals were then acquired in LabView for one to two minutes for each constant pressure within the interval (0,2] bar with steps of 0.1 bar. Based on the arithmetic mean for each, the calibration curve was plotted in Figure S2. The resulting linear fit was implemented in the LabView program, to convert the voltage signals into measured pressure.

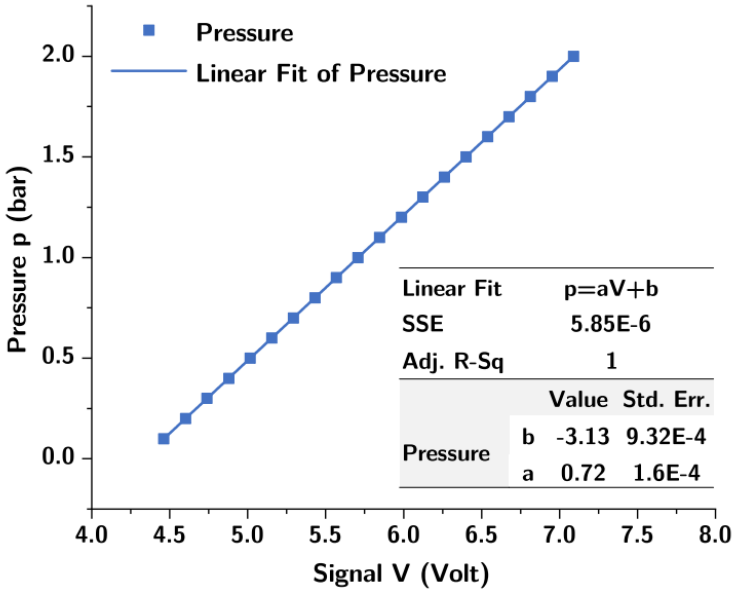


Figure S2. Calibration of the pressure sensor Elveflow MPS3 (0-2bar) with DRUCK DPI 600/IS.

Conditions adopted for the kinetic experiments

The initial conditions for all the carried-out kinetic experiments are summarized in Table S1. All the reaction mixtures contain 1 M KCl and 1% (v/v) MeOH.

Table S1. Initial conditions for the kinetic study of 4-nitrophenyl acetate (PNPA) hydrolysis reaction.

Run	$[PNPA]_0$ (mM)	pH
1	$3.43 \cdot 10^{-2}$	10.52
2	$5.76 \cdot 10^{-2}$	10.52
3	$1.95 \cdot 10^{-2}$	10.52
4	$1.28 \cdot 10^{-2}$	10.52
5	$1.30 \cdot 10^{-2}$	11.07
6	$2.86 \cdot 10^{-2}$	11.30
7	$3.36 \cdot 10^{-2}$	11.37
8	$3.70 \cdot 10^{-2}$	11.07

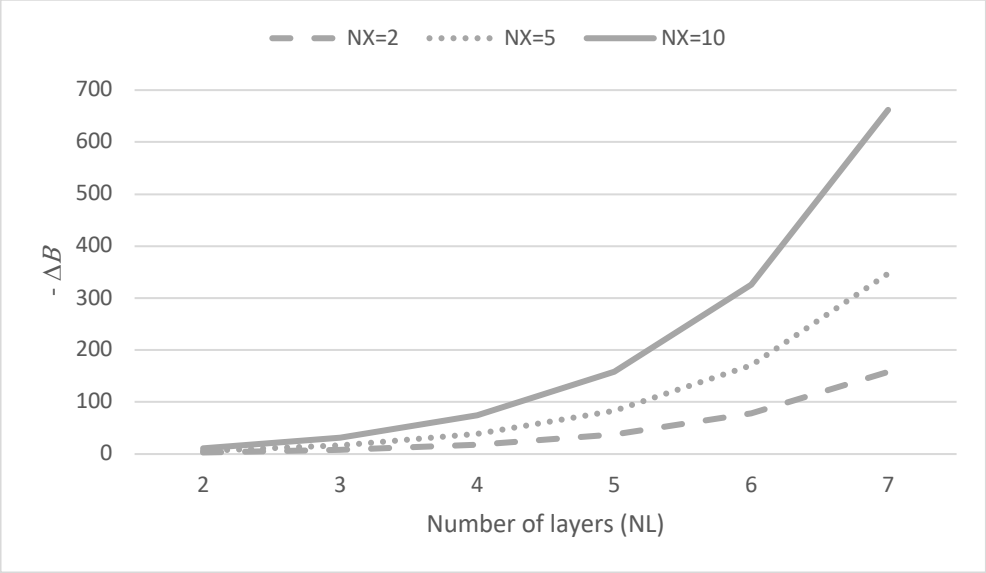


Figure S3. Scalability comparison – delta between MINLP binaries between the two compared formulations $\Delta B = B_{New} - B_{Cozad}$, NF=5 (4).

Table S2. Summary of the number of binaries, continuous decision variables and MINLP constraints for the scalability simulations.

NL=3, NX=2, NF=5 (4)	No. of Binary Variables		No. of Contin. Variables		No. of Constrains	
NE	New	Cozad et al	New	Cozad et al	New	Cozad et al
10	25	33	72	70	450	862
20			142	140	880	1692
30			212	210	1310	2522
40			282	280	1740	3352
50			352	350	2170	4182
60			422	420	2600	5012
70			492	490	3030	5842
80			562	560	3460	6672
90			632	630	3890	7502
100			702	700	4320	8332
200			1402	1400	8620	16632

NL=3, NX=2,
NE=10

NF	NE	Binaries	Con. Variables	Constraints
4	30	22	212	1127
	50		352	1867
	100		702	3717
5	30	25	212	1310
	50		352	2170
	100		702	4320
6	30	28	212	1490
	50		352	2470
	100		702	4920
7	30	31	212	1760
	50		352	2920
	100		702	5820

NL=3, NE=10, NF=5(4)	New	Cozad	New	Cozad	New	Cozad
NX	Binaries		Contin. Variables		Constraints	
2	25	33	72	70	450	862
3	29	40				1002
4	33	47				1142
5	37	54				1282
6	41	61				1422
7	45	68				1562
10	57	89				1982
15	77	124				2682

NE=10, NF=5(4), NX=2	New	Cozad	New	Cozad	New	Cozad
NL	Binaries		Contin. Variables		Constraints	
2	10	13	31	30	158	342
3	25	33	72	70	450	862
4	55	73	154	150	1034	1902
5	115	153	318	310	2202	3982
6	235	313	648	630	4538	8142

Table S3. Data set. Newton's law.

Training data set	
Shear stress (Pa)	Shear rate (s^{-1})
5.22	48.56
7.08	67.99
10.36	97.12
13.11	126.26
14.91	145.68
17.19	165.11
20.65	194.25
24.08	223.38
Validation data set	
3.00	29.14
25.42	242.81

Table S4. Data set. Non-Newtonian power law.

Training data set	
Wall Shear stress (Pa)	Apparent Shear rate $(ds)^{-1}$
6.03	2.914
9.39	4.856
12.02	6.799
15.87	9.712
18.57	12.626
20.82	14.568
22.03	16.511
24.30	19.425
26.59	22.338
Validation data set	
2.46	0.971
28.00	24.281

Table S5. Data set. First order kinetic law.

Training data set	
PNPA concentration (mM)	Reaction rate $(mM \cdot h^{-1})$
0.03432	0.2993
0.003556	0.03083
0.004627	0.04087
0.003858	0.03320
0.009999	0.08595
0.002937	0.02613
0.005013	0.04463
0.004332	0.03868
0.002954	0.02639

0.015852	0.13516
0.014771	0.1251
0.036614	0.3094
0.004719	0.03984
0.017024	0.1437
0.011981	0.1011
0.008703	0.07884
0.018246	0.1531
0.013767	0.1154
0.003982	0.03332
0.006349	0.05801
0.012848	0.1071
0.010787	0.09887
0.034153	0.2843
0.009354	0.07776
0.006836	0.06275
0.004407	0.03657
0.002578	0.02371
0.005071	0.04174
0.003564	0.03317
0.007038	0.05756
0.01958	0.1595
Validation data set	
0.001112	0.009741
0.002001	0.01648
0.002118	0.01951
0.002502	0.02311
0.04909	0.4562
0.04554	0.4102
0.04226	0.3741
0.03931	0.3387
0.001112	0.009741

Table S6. Data set. Kinetics dependence on pH.

Training data set		
PNPA concentration (mM)	OH- concentration (mM)	Reaction rate ($mM \cdot h^{-1}$)
0.01597	2.3333	0.9782
0.03337	1.1667	1.0221
0.03012	1.1667	0.9225
0.03511	1.1667	1.0770
0.03359	2.3333	2.05088
0.03432	0.3333	0.2992
0.007114	2.0000	0.3722

0.02862	1.1667	0.8794
0.02719	1.1667	0.8295
0.007759	2.3333	0.4732
0.006020	1.1667	0.1836
0.008517	2.0000	0.4447
0.03170	1.1667	0.9763
0.003556	0.3333	0.03083
0.005997	2.0000	0.3119
0.02458	1.1667	0.7454
0.02338	1.1667	0.7088
0.004627	0.3333	0.04087
0.02586	1.1667	0.7816
0.01053	2.3333	0.6537
0.02222	1.1667	0.6697
0.003858	0.3333	0.03320
0.009999	0.3333	0.08595
0.002937	0.3333	0.02613
0.005013	0.3333	0.04462
0.009079	1.1667	0.2730
0.004332	0.3333	0.02613
0.002954	0.3333	0.04462
0.005755	2.3333	0.3441
0.01585	0.3333	0.1352
0.007776	2.0000	0.3974
0.009962	1.1667	0.2787
0.01438	2.3333	0.9057
0.01477	0.3333	0.1251
0.01580	1.1667	0.4681
0.00950	1.1667	0.2824
0.02185	2.3333	1.3842
0.004719	0.3333	0.03984
0.02114	1.1667	0.6247
0.01702	0.3333	0.1437
0.008703	0.3333	0.07884
0.01764	2.3333	1.1208
0.008990	1.1667	0.3044
0.01123	2.0000	0.6123
0.01775	2.0000	0.9678
0.01825	0.3333	0.1531
0.01228	2.0000	0.6714
0.01377	0.3333	0.1154
0.02014	1.1667	0.5907
0.003982	0.3333	0.03332
0.01085	1.1667	0.2478
0.006349	0.3333	0.05801
0.004623	2.0000	0.2317
0.01971	2.3333	1.2629
0.01285	0.3333	0.1071

0.01079	0.3333	0.09887
0.03415	0.3333	0.2843
0.009354	0.3333	0.07776
0.006835	0.3333	0.06275
0.006970	2.3333	0.40550
0.03017	2.3333	1.9420
0.01917	1.1667	0.5570
0.01619	2.0000	0.8932
0.004407	0.3333	0.03657
Validation data set		
0.001112	0.3333	0.009741
0.001840	2.0000	0.09719
0.001845	2.3333	0.1117
0.04226	0.3333	0.3741
0.03931	0.3333	0.3387
0.03696	1.1667	1.1138
0.009484	2.3333	0.5899
0.01166	2.3333	0.7264
0.01296	2.3333	0.8153
0.0005041	2.3333	0.02100
0.002457	2.0000	0.1252
0.04554	0.3333	0.4102
0.03661	0.3333	0.3094
0.01198	0.3333	0.1011
0.003936	4.3333	0.4644
0.002578	0.3333	0.02371

References

[1] A. Volk, C.J. Kähler, Density model for aqueous glycerol solutions, Exp. Fluids. 59 (2018) 75. doi:10.1007/s00348-018-2527-y.

[2] N.S. Cheng, Formula for the viscosity of a glycerol-water mixture, Ind. Eng. Chem. Res. 47 (2008) 3285–3288. doi:10.1021/ie071349z.