THE JOURNAL OF PHYSICAL CHEMISTRY A
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

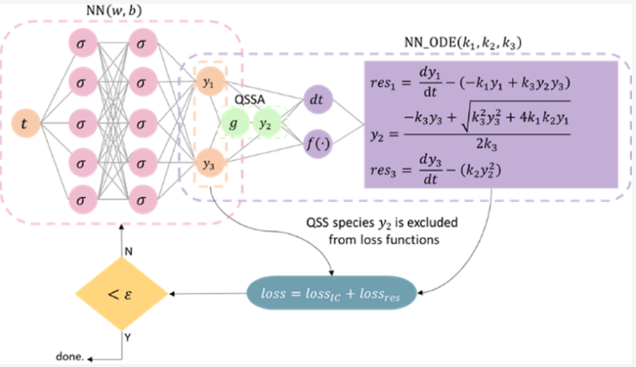
pubs.acs.org/JPCA Article

Stiff-PINN: Physics-Informed Neural Network for Stiff Chemical Kinetics

Weiqi Ji,*,[‖] Weilun Qiu,[‖] Zhiyu Shi,[‖] Shaowu Pan, and Sili Deng*

Cite This: J. Phys. Chem. A 2021, 125, 8098−8106 Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information


NN(w, b)
σ
t
y₁
y₃
QSSA
g
y₂
dt
f(·)
NN_ODE(k₁, k₂, k₃)
$res_1 = \frac{dy_1}{dt} - (-k_1y_1 + k_3y_2y_3)$
$y_2 = \frac{-k_3y_3 + \sqrt{k_3^2y_3^2 + 4k_1k_2y_1}}{2k_3}$
$res_3 = \frac{dy_3}{dt} - (k_2y_2^2)$
QSS species y₂ is excluded from loss functions
N
< ε
$loss = loss_{IC} + loss_{res}$
Y
done.

ABSTRACT: The recently developed physics-informed neural network (PINN) has achieved success in many science and engineering disciplines by encoding physics laws into the loss functions of the neural network such that the network not only conforms to the measurements and initial and boundary conditions but also satisfies the governing equations. This work first investigates the performance of the PINN in solving stiff chemical kinetic problems with governing equations of stiff ordinary differential equations (ODEs). The results elucidate the challenges of utilizing the PINN in stiff ODE systems. Consequently, we employ quasi-steady-state assumption (QSSA) to reduce the stiffness of the ODE systems, and the PINN then can be successfully applied to the converted non-/mild-stiff systems. Therefore, the results suggest that stiffness could be the major reason for the failure of the regular PINN in the studied stiff chemical kinetic systems. The developed stiff-PINN approach that utilizes QSSA to enable the PINN to solve stiff chemical kinetics shall open the possibility of applying the PINN to various reaction-diffusion systems involving stiff dynamics.

1. INTRODUCTION

Deep learning has enabled advances in many scientific and engineering disciplines, such as computer visions, natural language processing, and autonomous driving. Depending on the applications, many different neural network architectures have been developed, including deep neural network (DNN), convolutional neural network (CNN), recurrent neural network (RNN), and graph neural network. Some of them have also been employed for data-driven physics modeling,[1−8] including turbulent flow modeling[9] and chemical kinetic modeling.[10−14] These different neural network architectures introduce specific regularization to the neural network based on the nature of the task such as the scale and rotation invariant of the convolutional kernel in the CNN. Among them, the recently developed physics-informed neural network approach (PINN)[15−21] enables the construction of the solution space of differential equations using DNNs with space and time coordinates as the inputs. The governing equations (mainly differential equations) are enforced by minimizing the residual loss function using automatic differentiation, and thus, it becomes a physics regularization of the DNN. This framework permits solving differential equations (i.e., forward problems) and conducting parameter inference from observations (i.e., inverse problems). The PINN has been employed for predicting the solutions for Burgers' equation, the Navier−Stokes equations, and the Schrodinger equation.[16] To enhance the robustness and generality of the PINN, multiple variations of the PINN have also been developed, such as variational PINNs,[22] parareal PINNs,[23] and nonlocal PINNs.[24]

Despite the successful demonstration of the PINN in many of the above works, Wang et al.[25] investigated a fundamental mode of failure of the PINN that is related to numerical stiffness, leading to unbalanced back-propagated gradients between the loss function of initial/boundary conditions and the loss function of residuals of the differential equations during model training. In addition to the numerical stiffness, physical stiffness might also impose new challenges in the training of the PINN. While the PINN has been applied for solving chemical reaction systems involving a single-step reaction,[19] stiffness usually results from the nonlinearity and complexity of the reaction network, where the characteristic timescales for species span a wide range of magnitude. Consequently, the challenges for the PINN to accommodate stiff kinetics can potentially arise from several reasons, including the high dimensionality of the state variables (i.e.,

Received: June 9, 2021
Revised: August 19, 2021
Published: August 31, 2021

ACS Publications

https://doi.org/10.1021/acs.jpca.1c05102

the number of species), the high nonlinearity resulting from the interactions among species, the imbalance in the loss functions for different state variables since the species concentrations could span several orders of magnitudes. Nonetheless, stiff chemical kinetics is essential for the modeling of almost every real-world chemical system such as atmospheric chemistry and the environment, energy conversion and storage, materials and chemical engineering, and biomedical and pharmaceutical engineering. Enabling the PINN for handling stiff kinetics will open the possibilities of using the PINN to facilitate the design and optimization of these wide ranges of chemical systems.

In chemical kinetics, the evolution of the species concentrations can be described as ordinary differential equation (ODE) systems with the net production rates of the species as the source terms. If the characteristic timescales for species span a wide range of magnitude, integrating the entire ODE systems becomes computationally intensive. Quasi-steady-state assumption (QSSA) has been widely adopted to simplify and solve stiff kinetic problems, especially in the 1960s when efficient ODE integrators were unavailable.[26] A canonical example of the utilization of the QSSA is the Michaelis–Menten kinetic formula, which is still widely adopted to formulate enzyme reactions in biochemistry. Nowadays, the QSSA is still widely employed in numerical simulations of reaction-transport systems to remove chemical stiffness and enable the explicit time integration with relatively large time steps.[27−29] Moreover, imposing the QSSA also reduces the number of state variables and transport equations by eliminating the fast species such that the computational cost can be greatly reduced. From a physical perspective,[26,30] the QSSA identifies the species (termed as QSS species) that are usually radicals with relatively low concentrations. Their net production rates are much lower than their consumption and production rates and thus can be assumed zero. From a mathematical perspective,[26] the stiffness of the ODEs can be characterized by the largest absolute eigenvalue of the Jacobian matrix, that is, the Jacobian matrix of the reaction source term to the species concentrations. The QSSA identifies the species that correspond to the relatively large eigenvalues of the chemical Jacobian matrix and then approximate the ODEs with differential-algebraic equations (DAEs) to reduce the magnitude of the largest eigenvalue of the Jacobian matrix and thus the stiffness.

In the current work, we will evaluate the performance of the PINN in solving two classical stiff dynamics problems and compare it with the performance of the stiff-PINN, which incorporates the QSSA into the PINN to reduce stiffness. In Section 2, the two classical stiff kinetic systems, the corresponding PINN models, and the implementation of the QSSA to formulate the stiff-PINN models will be presented. In Section 3, the performances of the regular PINN and stiff-PINN in solving the two stiff problems will be investigated. Finally, conclusions and the outlook of future work will be presented.

2. METHODOLOGY

2.1. Stiff Chemical Kinetic Systems.

A homogeneous chemical reaction system can be modeled using the following ODEs

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = \boldsymbol{f}(t, \boldsymbol{y}), \qquad t_0 \le t \le t_{\mathrm{final}} \tag{2.1}$$

$$\boldsymbol{y}(t_0) = \boldsymbol{y}_0 \tag{2.2}$$

where $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T$ is the column vector of species concentrations, and N is the number of chemical species. t is the time, and the initial and final time are denoted as t_0 and t_{final}, respectively. $\boldsymbol{y}_0$ is the column vector of the initial species concentrations. The ODE system described by eq 2.1 with the initial conditions specified by eq 2.2 can be numerically solved using an ODE integrator such as the explicit Euler method or Runge–Kutta method. However, many ODEs for chemical kinetic models are stiff,[31] and solving stiff ODEs with an explicit method requires very small time steps such that the integration could be computationally intensive. Otherwise, implicit ODE integrators such as the backward differentiation formula (BDF) can be used. However, in general, solving stiff ODEs is time-consuming since the implicit method usually involves solving the nonlinear systems with Newton iteration. Therefore, it is still an active research area to efficiently solve stiff ODE systems,[32,33] which is an integral part of many reaction-diffusion systems, such as in chemical engineering, energy conversion, and biomedical applications.

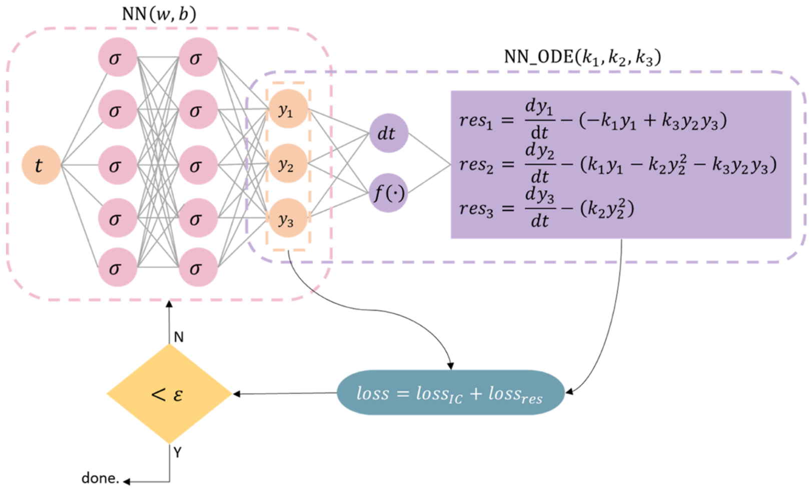
While it is difficult to give a precise definition of the stiffness of a chemical kinetic model, one criterion can be whether there are largely separated timescales for different species. For instance, some of the fast-evolving species have very short timescales, while some of the species evolve very slowly and have orders of magnitude longer timescales. To resolve these species concentrations with short timescales τ_{fast}, one has to use very small time steps in the explicit ODE integrators. However, to resolve the slowly evolving species, the number of integration steps scales with $S = \tau_{\mathrm{final}}/\tau_{\mathrm{fast}}$. If S is on the order of 1000 or larger, the system will be considered as stiff.[31] However, the shortest timescale τ_{fast} is defined locally and is evolving such that it is difficult to define the stiffness of a problem precisely. A practical approach to measure the stiffness is to compare the computational cost of explicit ODE integrators developed for nonstiff problems and implicit ones for stiff problems on a specific problem. If the computational cost using an implicit ODE integrator is much lower than the explicit ODE integrators, the problem can be regarded as a very stiff problem.

This work will investigate the performance of the PINN in two classical stiff chemical kinetic problems, ROBER[34] and POLLU,[35] which are extensively used for testing stiff ODE integrators. Specifically, the ROBER problem[34] consists of three species and five reactions, and the POLLU problem[35] consists of 20 species and 25 reactions describing the air pollutant formation in atmospheric chemistry. The formula of the ROBER problem (three ODEs) is presented here to illustrate the implementation of the PINN in the following sections, while the formula of the POLLU problem (20 ODEs) is presented in Supporting Information.

The ROBER problem refers to the following reaction network,

$$A \xrightarrow{k_1} B, \qquad B + B \xrightarrow{k_2} C + B,$$
$$B + C \xrightarrow{k_3} A + C \tag{2.3}$$

The reaction rate constants are $k_1 = 0.04$, $k_2 = 3 \times 10^7$, and $k_3 = 10^4$, and the initial conditions are $y_1(0) = 1$, $y_2(0) = 0$, and $y_3(0) = 0$, where y_1, y_2, and y_3 denote the concentrations of A, B, and C, respectively. The evolution of the species concentrations can be described by the following ODEs



NN(w, b)
σ
t
y_1
y_2
y_3
NN_ODE(k_1, k_2, k_3)
dt
$f(\cdot)$
$res_1 = \frac{dy_1}{dt} - (-k_1y_1 + k_3y_2y_3)$
$res_2 = \frac{dy_2}{dt} - (k_1y_1 - k_2y_2^2 - k_3y_2y_3)$
$res_3 = \frac{dy_3}{dt} - (k_2y_2^2)$
N
$< \varepsilon$
$loss = loss_{IC} + loss_{res}$
Y
done.

Figure 1. Schematic of the regular PINN for the ROBER problem. The only input is the time t, and the output is the solution vector $[y_1(t),y_2(t),y_3(t)]^T$, which has to satisfy the governing equations and the initial conditions. The two neutral networks, NN(w,b) and NN_ODE(k_1, k_2, k_3), share parameters, and both contribute to the loss function.

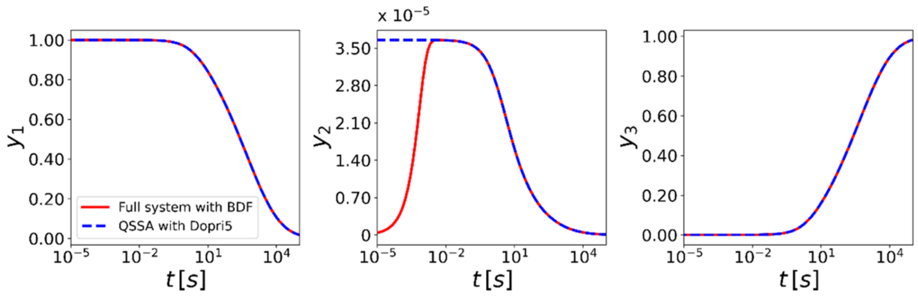


Figure 2. Comparisons of the solutions of the ROBER problem using the BDF solver and the solutions of the reduced system via the QSSA using Dopri5. The QSSA does not affect the solution of y_1 and y_3 and accurately predicts y_2 after the initial induction period. Note that the time is presented in the logarithmic scale to better illustrate the evolution of y_2 during the induction period.

$$\frac{\mathrm{d}y_1}{\mathrm{d}t} = -k_1y_1 + k_3y_2y_3,$$
$$\frac{\mathrm{d}y_2}{\mathrm{d}t} = k_1y_1 - k_2{y_2}^2 - k_3y_2y_3, \qquad \frac{\mathrm{d}y_3}{\mathrm{d}t} = k_2{y_2}^2 \tag{2.4}$$

The reaction rate constants vary in a range of 9 orders of magnitude, that is, $k_2/k_1 \sim 10^9$, resulting in a system with strong stiffness.

2.2. Physics-Informed Neural Network.

Without loss of generality, we shall use the ROBER problem to illustrate the framework of the PINN. Figure 1 illustrates the structure of the PINN informed by the ODEs of the ROBER problem.

The PINN framework consists of two neural networks. The first part of the framework is a neural network NN(w,b) that takes the time t as the input and outputs the concentrations of all of the species $\boldsymbol{y} = [y_1,y_2,y_3]^T$at that time. Then, the output of the NN is fed into a second network NN_ODE(k_1, k_2, k_3), which is essentially the governing differential equations of the ROBER problem, to evaluate the residuals of the ODEs. Finally, a loss function is constructed by combining the loss functions of the initial conditions and residuals. Specifically

$$\text{loss} = \text{loss}_{\text{IC}} + \text{loss}_{\text{res}},$$
$$\text{loss}_{\text{IC}} = \sum_{i=1}^{3} w_{\text{IC}}{}^{i}[y_i^{\text{NN}}(t = t_0) - y_i(t = t_0)]^2,$$
$$\text{loss}_{\text{res}} = \frac{1}{R}\sum_{j=1}^{R}\sum_{i=1}^{3} w_{\text{res}}{}^{i}[\text{res}_i(t = t_j)]^2 \tag{2.5}$$

where the weights of the initial conditions and the residuals for different species in the loss functions are rescaled by $w_{\text{IC}}{}^{i}$ and $w_{\text{res}}{}^{i}$ since they could be unbalanced with each other by orders of magnitude and hinder the training. The predicted initial conditions to evaluate loss_{IC} are from the first neural network NN, and the loss of the residuals is from the second network NN_ODE. The residual loss loss_{res} is evaluated at randomly sampled points in the computational domains, that is, $\{t_1,t_2,...,t_R\} \in [t_0,t_{\text{final}}]$. Backpropagation through the two networks is conducted on the auto-differentiation framework of PyTorch to compute the gradient of the loss functions to the weights of the neural networks. The time derivatives of $\mathrm{d}y_i/\mathrm{d}t$ used in the residual loss are obtained using auto-differentiation as well. The neural network is optimized via the first-order optimizer Adam.[36]

2.3. Quasi-Steady-State Assumption.

As previously discussed, the QSSA is often imposed on certain species to reduce the computational cost of simulating the evolution of the system. These QSS species, which are often radicals and

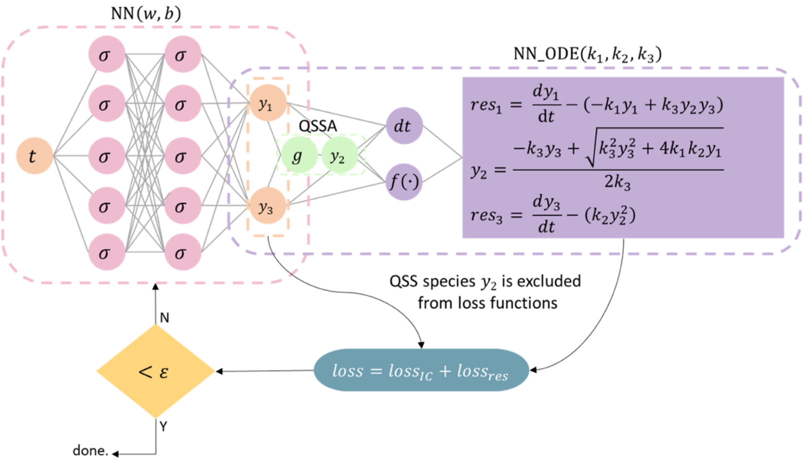

$NN(w, b)$
σ
t
y_1
y_3
QSSA
g
y_2
dt
$f(\cdot)$
$NN\_ODE(k_1, k_2, k_3)$
$res_1 = \frac{dy_1}{dt} - (-k_1y_1 + k_3y_2y_3)$
$y_2 = \frac{-k_3y_3 + \sqrt{k_3^2y_3^2 + 4k_1k_2y_1}}{2k_3}$
$res_3 = \frac{dy_3}{dt} - (k_2y_2^2)$
QSS species y_2 is excluded from loss functions
$loss = loss_{IC} + loss_{res}$
$< \varepsilon$
N
Y
done.

Figure 3. Schematic of the stiff-PINN for the ROBER problem. Comparing the model pipeline for the stiff-PINN with that for the regular PINN, species y_2 is assumed to be in the QSS and the residuals of these QSS species are excluded from the loss functions during the training of the stiff-PINN. By removing these relatively fast-evolving species from the loss function, the stiffness of the differential equations is reduced.

unstable intermediates, generally have shorter timescales compared to other species. By assuming that the net production rates of the QSS species are zero, the concentrations of these species can be expressed by algebraic equations instead of ODEs such that the number of ODEs to model the kinetic system is reduced. Note that although the net production rates of the QSS species are small, their production and consumption rates are not necessarily small. In a general form, the net production rate of species k can be written as

$$\frac{\mathrm{d}Y_\mathrm{k}}{\mathrm{d}t} = {\omega_\mathrm{k}}^{+} - {\omega_\mathrm{k}}^{-} \tag{2.6}$$

Y_k is the species concentration, and ω_k^{+}, ω_k^{-} are the production and consumption rates, respectively. The QSSA requires that

$$\left|\frac{\mathrm{d}Y_\mathrm{k}}{\mathrm{d}t}\right| \ll ({\omega_\mathrm{k}}^{+}, {\omega_\mathrm{k}}^{-}) \tag{2.7}$$

so that

$${\omega_\mathrm{k}}^{+} - {\omega_\mathrm{k}}^{-} \approx 0 \tag{2.8}$$

Take the ROBER problem as an example, as shown in Figure 2; the concentration of species y_2 increases sharply during the initial induction period, and then, it goes to a phase of slow change with considerably low concentrations compared to y_1 and y_3. The production rate of y_2 is comparable with the consumption rate of y_1, as can be seen from eq 2.3. However, since the net production rate over the entire integration range can be estimated to be proportional to the maximum species concentrations and the maximum concentration of y_2 is 5 orders of magnitude lower than that of y_1, the net production rate of y_2 is much slower than y_1. Similarly, the consumption rate of y_2 is comparable with the production rate of y_3, while the net production rate of y_2 is much slower than y_3. Therefore, the species y_2 is likely to be a QSS species compared to y_1 and y_3.

The assumption of y_2 as a QSS species implies that

$$0 = k_1y_1 - k_2{y_2}^2 - k_3y_2y_3 \tag{2.9}$$

and

$$y_2 = \frac{-k_3y_3 + \sqrt{{k_3}^2{y_3}^2 + 4k_1k_2y_1}}{2k_2} \tag{2.10}$$

Consequently, the original ROBER problem of three ODEs can be approximated by the following DAE and two ODEs.

$$\frac{\mathrm{d}y_1}{\mathrm{d}t} = -k_1y_1 + k_3y_2y_3,$$
$$y_2 = \frac{-k_3y_3 + \sqrt{{k_3}^2{y_3}^2 + 4k_1k_2y_1}}{2k_2}, \qquad \frac{\mathrm{d}y_3}{\mathrm{d}t} = k_2{y_2}^2 \tag{2.11}$$

The issue of stiffness in the system described by eq 2.11 should be reduced compared to the system described by eq 2.4 since the fast-evolving species y_2 is not explicitly solved. Therefore, we can use an explicit integrator to solve eq 2.10, and the results are shown in Figure 2. It is shown that the explicit Runge−Kutta method of the Dopri5 method well predicts y_1 and y_3 under the QSSA. In addition, the species profile of y_2 can be computed based on the algebraic equation in eq 2.10, and the approximated profile agrees well with the accurate solution of y_2 except for the initial induction period. Note that, in practice, what interests us are the stable species rather than unstable intermediates.

We can then integrate the QSSA into the framework of the PINN, and the new framework denoted as the stiff-PINN is illustrated in Figure 3. The key differences between the stiff-PINN and regular PINN are three folds: first, NN_ODE is informed by the reduced systems with the QSSA instead of the original ODEs. Second, NN will only output the non-QSS species, and the QSS species are approximated in NN_ODE. Third, the QSS species are excluded from the loss functions of the initial conditions and the loss functions of the residuals, and therefore, loss functions will only consist of the information of non-QSS species.

It is worthy to note that manually deriving explicit algebraic expressions for the QSS species could be challenging for complex systems. Automatic model reduction tools[37,38] could help tackle the challenges by linearizing the QSSA approximation. Alternatively, the QSS species have to be solved via an additional nonlinear optimization step.

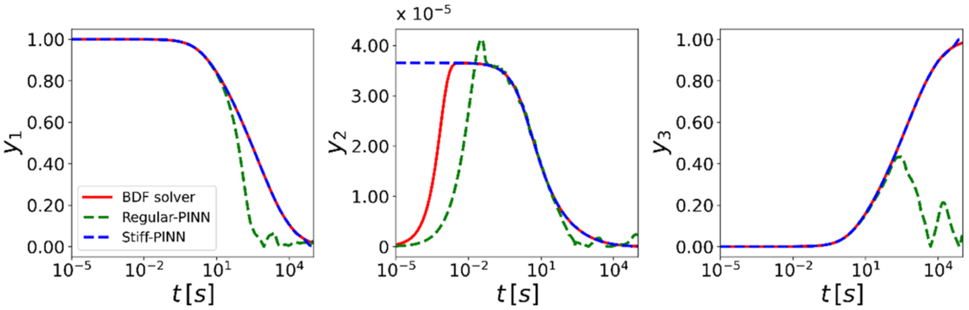


Figure 4. Solutions of the benchmark ROBER problem using the BDF solver (the exact solution), regular PINN, and stiff-PINN with the QSSA. While the regular PINN fails to predict the kinetic evolution of the stiff system, the stiff-PINN with the QSSA works very well.

3. RESULTS AND DISCUSSION

In this section, the performance of the regular PINN in two classical stiff chemical kinetic problems ROBER and POLLU will be investigated. As will be discussed, the stiffness is the main reason for the failure of the regular PINN, and the performance of the stiff-PINN in removing stiffness and predicting the species profiles will then be demonstrated. The code used to produce the results is available at https://github.com/DENG-MIT/Stiff-PINN.

3.1. ROBER Problem. We first employed a regular PINN to solve the ROBER problem. The NN has a single input of time t and outputs the concentrations of three species $[y_1, y_2, y_3]^T$. To avoid the imbalance between the loss of initial conditions and the loss of residuals, we hard-coded the initial conditions to the NN architecture, that is

$$y = y_0 + t^*NN(\log(t)) \tag{3.1}$$

Therefore, to train the regular PINN, only the loss functions of the residuals for the three species needed to be minimized. A total of 2500 data points is sampled uniformly in a logarithmic scale of the computational domain (time domain of $t \in [10^{-5}, 10^5]$s) to evaluate the residuals. The neural network had three hidden layers and 128 nodes per hidden layer, with the activation function of Gaussian error linear units.[39] The weights of the NN were initialized using Xavier[40] and optimized via Adam,[36] with the default learning rate of 0.001. The training was conducted via minibatch training with a minibatch size of 128. The training results are shown in Figure 4. Overall, the learned regular PINN can capture species y_1 and y_3 during the initial stage of $t \in [0,10\text{ s}]$ and then substantially deviate from the exact solutions. Although the adaptive weights were implemented for the loss components of the three species following,[20,25] the training of the regular PINN for the ROBER problem still failed. The deficiency of adaptive weights in the ROBER problem might be due to the fact that the stiffness here is due to the multiscale nature in the chemical dynamic system, while the stiffness in[25] is attributed to the imbalance among the loss functions of boundary conditions and residuals.

We hypothesize that the failure is due to the physical stiffness of the ROBER problem. A natural way to test this hypothesis is to convert this problem to a nonstiff problem and see whether the regular PINN would work. We then applied the stiff-PINN to the ROBER problem with the QSSA, described by eq 2.11, and the results are also shown in Figure 4. Although the species y_2 is not included in the output of the NN, it can be computed via eq 2.10. The hyperparameters for the stiff-PINN are almost identical to those of the regular PINN except that the stiff-PINN only outputs two species, while the regular PINN outputs three species. Here, we take the absolute value of y_1 and y_3 to ensure the validity of eq 2.10. As can be seen in Figure 4, the stiff-PINN accurately captures all three species profiles, including y_2. The history of the loss functions for both the regular PINN and stiff-PINN are presented in Figure 5. As expected, the loss function for the regular PINN stays at a very high value, while that of the stiff-PINN is decreased by a factor of 6 orders of magnitude.

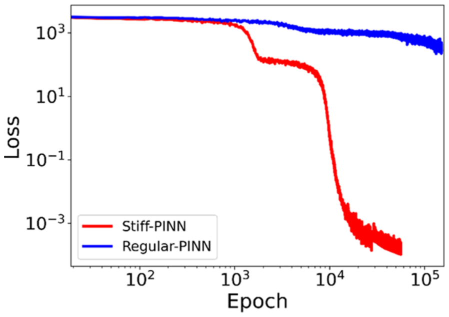


Figure 5. History of the loss functions of the regular PINN and stiff-PINN for the ROBER problem. The epoch here corresponds to each parameter update.

While analyzing the gradient flow dynamics is beyond the scope of this work, we draw some intuition on the difference in the performances between the stiff-PINN and regular PINN. In the stiff-PINN, y_2 is eliminated by the QSSA, and y_1 and y_3 are of the same order of magnitude, which makes their evolution much easier to be approximated with a single neural network. In addition, excluding y_2 from the loss function also mitigates the imbalance among the residual losses for fast and slow species. However, in the regular PINN, the neural network has to approximate all three species simultaneously, where $y_2 \sim O(10^{-5})$ while y_1,$y_3 \sim O(1)$. The large-scale separation is a great challenge to the approximation capacity of the neural network. Therefore, even if the full ODE system is modeled with a sufficiently large and deep neural network, the training would be very challenging.

We shall also briefly discuss the sensitivities of the stiff-PINN to the number of nodes and layers of the NN model. The performance is found to be sensitive to the size (neurons × layers) of the NN as shown in Table 1. The general trend is that the performance of deeper neural networks performs better by comparing the case of 64 × 5 with 64 × 4 and 128 × 3 with 128 × 2, respectively. Empirically, the architecture of three layers and 128 nodes per layer performs best in this work.

Table 1. Root-Mean-Square-Error (RMSE) of the Stiff-PINN with Different NN Sizes

neurons×layers	RMSE(y_1)	RMSE(y_3)
64 × 4	4.7954×10^{-3}	7.1774×10^{-3}
64 × 5	1.4469×10^{-3}	5.0369×10^{-3}
128 × 2	1.4805×10^{-2}	2.4651×10^{-2}
128 × 3	1.3628×10^{-3}	2.0699×10^{-3}
256 × 1	1.5961×10^{-2}	4.5460×10^{-2}

3.2. POLLU Problem. The POLLU problem consists of 20 species and 25 reactions. It is an air pollution model developed at the Dutch National Institute of Public Health and Environmental Protection.[35] The POLLU problem can be described mathematically by the following 20 nonlinear ODEs shown in eq 3.2. Full details about the model can be found in Supporting Information

$$\frac{d\boldsymbol{y}}{dt} = f(\boldsymbol{y}),\ \boldsymbol{y}(0) = \boldsymbol{y}_0, \qquad \boldsymbol{y} \in R^{20},\ 0 \le t \le 60\ \text{s} \tag{3.2}$$

As expected, the regular PINN also failed in the POLLU problem, and the performance is shown in Figure S1 of Supporting Information. We then empirically chose 10 species as QSS species based on their maximum concentrations, that is, lower than 1×10^{-4}, and derived the algebraic equations for these 10 species, as shown in Supporting Information. Then, the POLLU problem was reduced to 10 ODEs and 10 algebraic equations from 20 ODEs. The evolution of species concentrations described by the original ODEs and the reduced DAEs by imposing the QSSA is shown in Figures 6 and 7. The QSSA has little effects on the profiles of non-QSS species (Figure 6), while there are some discrepancies in those of the QSS species (Figure 7), especially during the initial induction period. However, this assumption does not affect the predictions of the stable species, which are often of greater interest than the unstable intermediates in practice.

Also included in Figures 6 and 7 are the solutions obtained with the stiff-PINN approach. To facilitate the training, different fixed weights for each species in the residual loss function were applied to balance them since the maximum concentrations of the non-QSS species still span several orders

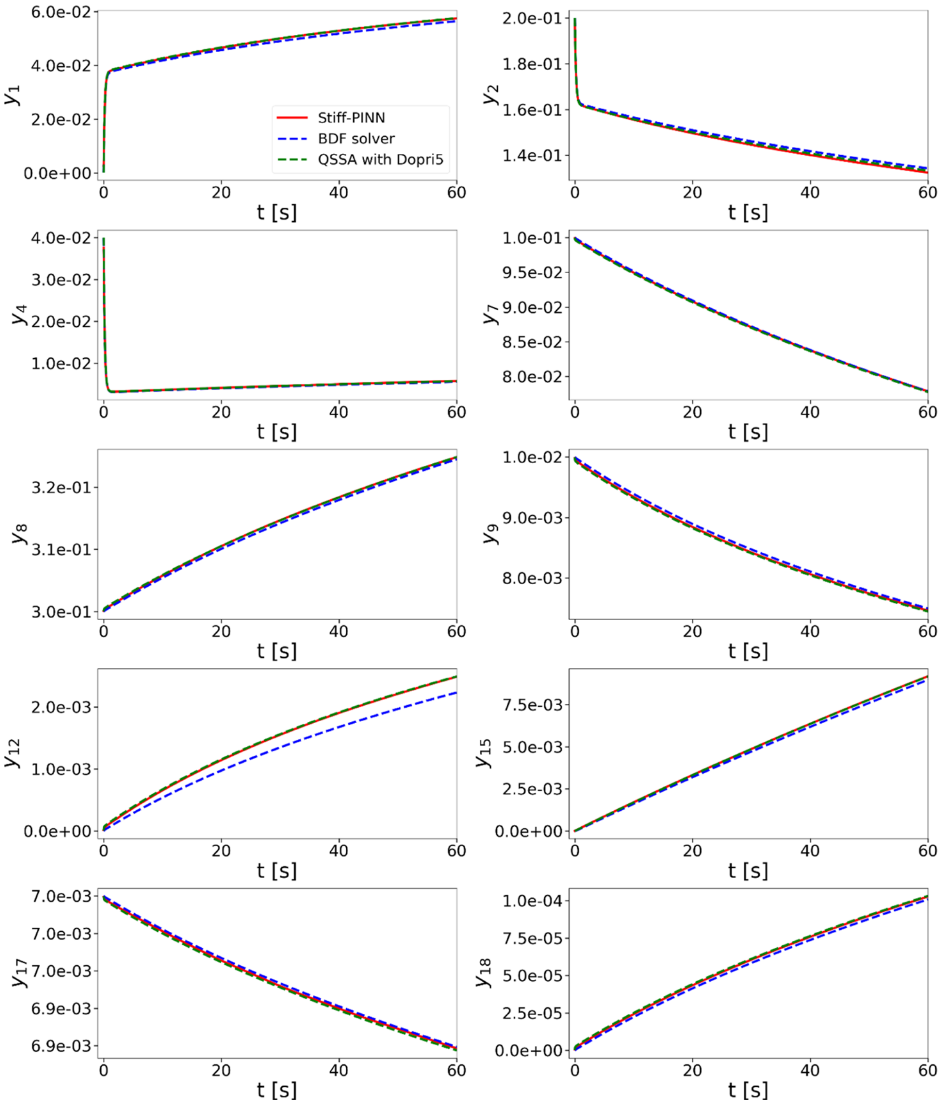


Figure 6. Evolution of the concentration of the 10 non-QSS species in the POLLU problem obtained by solving the original 20 ODEs using the BDF, the reduced DAEs using Dopri5, and stiff-PINN.

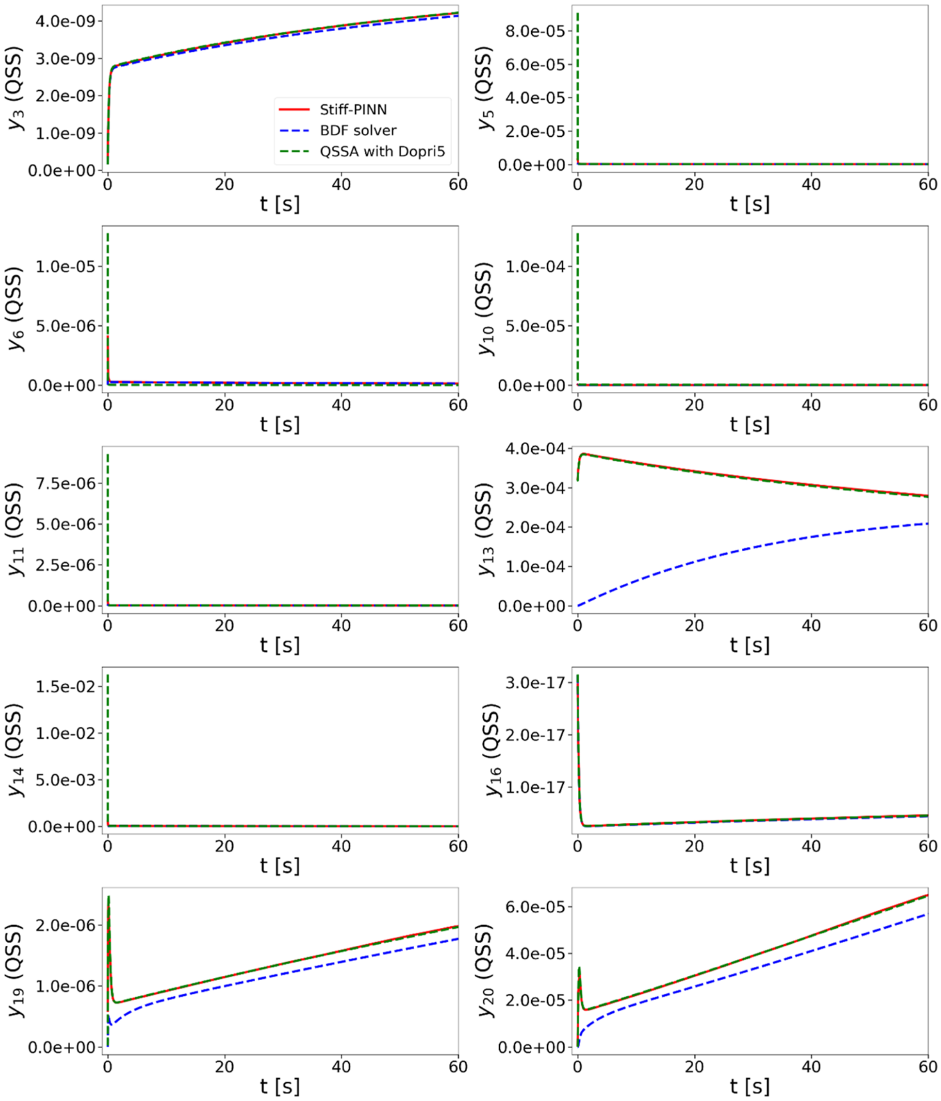


Figure 7. Evolution of the concentration of the 10 QSS species in the POLLU problem obtained by solving the original 20 ODEs using the BDF, the reduced DAEs using Dopri5, and stiff-PINN.

of magnitude. The stiff-PINN can accurately predict the evolution of non-QSS species and most of the QSS species. For QSS species [y_{13},y_{19},y_{20}], the concentrations of which are close to the non-QSS species, the QSSA tends to induce a larger error compared to the rest of the QSS species, which is expected and consistent with the results using ODE solvers. In general, the stiff-PINN can accurately solve the POLLU problem with stiffness removal via the QSSA. We also analyzed the history of the loss functions for the regular and stiff-PINN, as shown in Figure 8. Although the loss functions of both the regular PINN and stiff-PINN decrease as the training goes on, the loss in the stiff-PINN is 4 orders of magnitude smaller than that of the regular PINN.

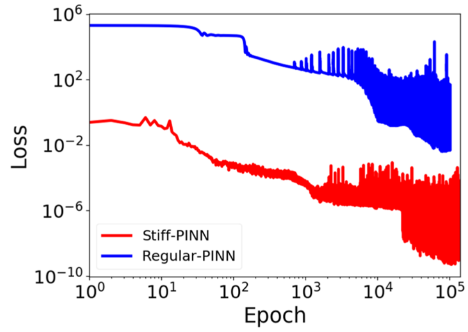


Figure 8. History of the loss functions of the regular PINN and stiff-PINN for the POLLU problem.

4. CONCLUSIONS AND OUTLOOK

The performance of the PINN in stiff chemical kinetic systems was investigated. In the two classical stiff chemical kinetic systems, ROBER and POLLU, the regular PINN failed to predict the evolution of the systems. By imposing QSSA on certain species in the kinetic systems and reducing the stiffness, the stiff-PINN well captured the dynamic responses of the systems. Therefore, it indicates the failure of the PINN was due to the physical stiffness, and the proposed stiff-PINN is an effective and promising approach for stiff chemical kinetic systems.

However, there are still a lot of open questions to be addressed in order to develop a robust and general PINN framework for stiff chemical kinetic problems. We shall highlight some of the challenges learned from performing this work that could guide future advances:

(i) How to handle the fast timescales associated with linear combinations of several species[29,41] rather than single species? More general stiffness-removal approaches may help address these challenges, such as computational singular perturbation (CSP),[42,43] intrinsic low-dimensional manifolds (ILDM),[44,45] and global quasi-linearization (GQL).[29,41] These advanced low-order modeling approaches will also help reduce the approximation error induced by the QSSA, for example, the species of y_{12} in the POLLU problem.

(ii) Manually deriving the QSSA formula for complex chemical systems can be time-consuming and requires a deep understanding of the process. How to identify QSS species and derive the corresponding DAEs automatically to be incorporated into the stiff-PINN framework? Automatic reduction tools[37,46] and above general low-order approximation approaches (e.g., CSP, ILDM, and GQL) may help tackle the challenge.

(iii) The stiffness in complex chemical systems may not be eliminated, and the reduced system may still show mild stiffness. Advancement in neural network optimizations is required to train the PINN for mild stiff systems. A possible solution is exploiting stiff ODE solvers as neural network optimizers.[47] One of the drawbacks of such an approach is that the Hessian matrix of the loss function with respect to the neural network parameters is required. Recent advancements in solving stiff ODEs using an explicit method[32] and semi-implicit method[33] may help mitigate the requirements of the Hessian matrix. Another possible direction is normalizing the loss functions on-the-fly based on the timescales (e.g., via the eigenvalue of the Jacobian matrix) during the training process.

Finally, while this work focuses on enabling the PINN for stiff systems, the idea of estimating the species that have slow timescales can also help tackle the challenges in other data-driven modeling approaches. For example, it has been shown that training neural ODEs for representing kinetic models using neural networks could be challenging for stiff chemical kinetic systems,[48] and excluding QSS species from the network could help the training.[49]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpca.1c05102.

Details of the POLLU model including the original full model and the QSSA reduction and training results for the POLLU model using the regular PINN (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Weiqi Ji − *Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0002-7097-0219; Email: weiqiji@mit.edu

Sili Deng − *Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0002-3421-7414; Email: silideng@mit.edu

Authors

Weilun Qiu − *College of Engineering, Peking University, Beijing 100871, China*

Zhiyu Shi − *College of Engineering, Peking University, Beijing 100871, China*

Shaowu Pan − *Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan 48109, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpca.1c05102

Author Contributions

‖W.J., W.Q. and Z.S. contributed equally to this work.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

S.D. would like to acknowledge the support from the d'Arbeloff Career Development allowance at the Massachusetts Institute of Technology.

■ REFERENCES

(1) Qin, T.; Wu, K.; Xiu, D. Data Driven Governing Equations Approximation Using Deep Neural Networks. *J. Comput. Phys.* **2019**, *395*, 620−635.

(2) Long, Z.; Lu, Y.; Ma, X.; Dong, B. PDE-Net: Learning PDEs from Data. **2017**, *arXiv:1710.09668*. arXiv Preprint.

(3) Raissi, M.; Wang, Z.; Triantafyllou, M. S.; Karniadakis, G. E. Deep Learning of Vortex-Induced Vibrations. *J. Fluid Mech.* **2019**, *861*, 119−137.

(4) Schweidtmann, A. M.; Rittig, J. G.; König, A.; Grohe, M.; Mitsos, A.; Dahmen, M. Graph Neural Networks for Prediction of Fuel Ignition Quality. *Energy Fuels* **2020**, *34*, 11395−11407.

(5) Bar-Sinai, Y.; Hoyer, S.; Hickey, J.; Brenner, M. P. Learning Data-Driven Discretizations for Partial Differential Equations. *Proc. Natl. Acad. Sci. U.S.A.* **2019**, *116*, 15344−15349.

(6) Champion, K.; Lusch, B.; Kutz, J. N.; Brunton, S. L. Data-Driven Discovery of Coordinates and Governing Equations. *Proc. Natl. Acad. Sci. U.S.A.* **2019**, *116*, 22445−22451.

(7) Xie, T.; Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **2018**, *120*, 145301.

(8) Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems*; Montreal, Canada: 2018; pp 6571−6583.

(9) Duraisamy, K.; Iaccarino, G.; Xiao, H. Turbulence Modeling in the Age of Data. *Annu. Rev. Fluid Mech.* **2019**, *51*, 357−377.

(10) Ranade, R.; Alqahtani, S.; Farooq, A.; Echekki, T. An Extended Hybrid Chemistry Framework for Complex Hydrocarbon Fuels. *Fuel* **2019**, *251*, 276−284.

(11) Ji, W.; Deng, S. Autonomous Discovery of Unknown Reaction Pathways from Data by Chemical Reaction Neural Network. *J. Phys. Chem. A* **2021**, *125*, 1082−1092.

(12) Blasco, J. A.; Fueyo, N.; Larroya, J. C.; Dopazo, C.; Chen, Y.-J. A Single-Step Time-Integrator of a Methane-Air Chemical System Using Artificial Neural Networks. *Comput. Chem. Eng.* **1999**, *23*, 1127−1133.

(13) Franke, L. L. C.; Chatzopoulos, A. K.; Rigopoulos, S. Tabulation of Combustion Chemistry via Artificial Neural Networks (ANNs): Methodology and Application to LES-PDF Simulation of Sydney Flame L. *Combust. Flame* **2017**, *185*, 245−260.

(14) Zhang, T.; Zhang, Y.; E, W.; Ju, Y. DLODE: A Deep Learning-Based ODE Solver for Chemistry Kinetics. In *AIAA Scitech 2021 Forum*. January 2021; p 1139.

(15) Raissi, M.; Yazdani, A.; Karniadakis, G. E. Hidden Fluid Mechanics: Learning Velocity and Pressure Fields from Flow Visualizations. *Science* **2020**, *367*, 1026−1030.

(16) Raissi, M.; Perdikaris, P.; Karniadakis, G. E. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686−707.

(17) Zhang, D.; Lu, L.; Guo, L.; Karniadakis, G. E. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Quantifying Total Uncertainty in Physics-Informed Neural Networks for Solving Forward and Inverse Stochastic Problems*, 2019; Vol. *397*.

(18) Lagaris, I. E.; Likas, A.; Fotiadis, D. I. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations. *IEEE Trans. Neural Network.* **1998**, *9*, 987−1000.

(19) Lu, L.; Meng, X.; Mao, Z.; Karniadakis, G. E. DeepXDE: A Deep Learning Library for Solving Differential Equations. **2019**. *arXiv:1907.04502*. arXiv Preprint.

(20) Jin, X.; Cai, S.; Li, H.; Karniadakis, G. E. *NSFnets (Navier-Stokes Flow Nets): Physics-Informed Neural Networks for the Incompressible Navier-Stokes Equations*, 2020.

(21) Sun, L.; Gao, H.; Pan, S.; Wang, J.-X. Surrogate Modeling for Fluid Flows Based on Physics-Constrained Deep Learning without Simulation Data. *Comput. Methods Appl. Mech. Eng.* **2020**, *361*, 112732.

(22) Kharazmi, E.; Zhang, Z.; Karniadakis, G. E. Variational Physics-Informed Neural Networks For Solving Partial Differential Equations, **2019**; Vol. *1*−*24*. *arXiv:1912.00873*. arXiv Preprint.

(23) Meng, X.; Li, Z.; Zhang, D.; Karniadakis, G. E. PPINN: Parareal Physics-Informed Neural Network for Time-Dependent PDEs. *Comput. Methods Appl. Mech. Eng.* **2020**, *370*, 113250.

(24) Haghighat, E.; Bekar, A. C.; Madenci, E.; Juanes, R. A Nonlocal Physics-Informed Deep Learning Framework Using the Peridynamic Differential Operator. **2020**. *arXiv:2006.00446*. arXiv Preprint.

(25) Wang, S.; Teng, Y.; Perdikaris, P. Understanding and Mitigating Gradient Pathologies in Physics-Informed Neural Networks. **2020**, *arXiv:2001.04536*. arXiv Preprint.

(26) Turányi, T.; Tomlin, A. S.; Pilling, M. J. On the Error of the Quasi-Steady-State Approximation. *J. Phys. Chem.* **1993**, *97*, 163−172.

(27) Lu, T.; Law, C. K.; Yoo, C. S.; Chen, J. H. Dynamic Stiffness Removal for Direct Numerical Simulations. *Combust. Flame* **2009**, *156*, 1542−1551.

(28) Felden, A.; Pepiot, P.; Esclapez, L.; Riber, E.; Cuenot, B. Including Analytically Reduced Chemistry (ARC) in CFD Applications. *Acta Astronaut.* **2019**, *158*, 444−459.

(29) Yu, C.; Bykov, V.; Maas, U. Global quasi-linearization (GQL) versus QSSA for a hydrogen-air auto-ignition problem. *Phys. Chem. Chem. Phys.* **2018**, *20*, 10770−10779.

(30) Law, C. K. *Combustion Physics*; Cambridge University Press: Cambridge, 2006.

(31) Seinfeld, J. H.; Lapidus, L.; Hwang, M. Review of Numerical Integration Techniques for Stiff Ordinary Differential Equations. *Ind. Eng. Chem. Fundam.* **1970**, *9*, 266−275.

(32) Bassenne, M.; Fu, L.; Mani, A. Time-Accurate and Highly-Stable Explicit Operators for Stiff Differential Equations. *J. Comput. Phys.* **2021**, *424*, 109847.

(33) Wu, H.; Ma, P. C.; Ihme, M. Efficient Time-Stepping Techniques for Simulating Turbulent Reactive Flows with Stiff Chemistry. *Comput. Phys. Commun.* **2019**, *243*, 81−96.

(34) Robertson, H. The Solution of a Set of Reaction Rate Equations. In *Numerical Analysis: An Introduction*; Walsh, J., Ed.; Academic Press: London, 1966; Vol. *178182*, pp 178−182.

(35) Verwer, J. G. Gauss-Seidel Iteration for Stiff ODES from Chemical Kinetics. *SIAM J. Sci. Comput.* **1994**, *15*, 1243−1250.

(36) Kingma, D. P.; Ba, J. *Adam: A Method for Stochastic Optimization*. 2014, *arXiv:1412.6980*. arXiv Preprint.

(37) Lu, T.; Law, C. K. A Criterion Based on Computational Singular Perturbation for the Identification of Quasi Steady State Species: A Reduced Mechanism for Methane Oxidation with NO Chemistry. *Combust. Flame* **2008**, *154*, 761−774.

(38) Lu, T.; Law, C. K. Systematic Approach To Obtain Analytic Solutions of Quasi Steady State Species in Reduced Mechanisms. *J. Phys. Chem. A* **2006**, *110*, 13202−13208.

(39) Hendrycks, D.; Gimpel, K. Gaussian Error Linear Units (GELUs). **2016**, *arXiv:1606.08415*. arXiv Preprint.

(40) Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. *J. Mach. Learn. Res.* **2010**, *9*, 249−256.

(41) Bykov, V.; Gol'dshtein, V.; Maas, U. Simple Global Reduction Technique Based on Decomposition Approach. *Combust. Theor. Model.* **2008**, *12*, 389−405.

(42) Lam, S. H.; Goussis, D. A. Understanding Complex Chemical Kinetics with Computational Singular Perturbation. *Symposium (International) on Combustion*; Elsevier, 1989; Vol. *22*, pp 931−941. DOI: 10.1016/s0082-0784(89)80102-x

(43) Lam, S. H.; Goussis, D. A. The CSP Method for Simplifying Kinetics. *Int. J. Chem. Kinet.* **1994**, *26*, 461−486.

(44) Maas, U.; Pope, S. B. Simplifying Chemical Kinetics: Intrinsic Low-Dimensional Manifolds in Composition Space. *Combust. Flame* **1992**, *88*, 239−264.

(45) Maas, U.; Pope, S. B. Implementation of Simplified Chemical Kinetics Based on Intrinsic Low-Dimensional Manifolds. *Symposium (International) on Combustion*; Elsevier, 1992; Vol. *24*, pp 103−112. DOI: 10.1016/s0082-0784(06)80017-2

(46) Pepiot, P. *Automatic Strategies to Model Transportation Fuel Surrogates*; Stanford University: Stanford, CA, 2008.

(47) Owens, A. J.; Filkin, D. L. Efficient Training of the Back Propagation Network by Solving a System of Stiff Ordinary Differential Equations. *Proceedings IEEE/INNS International Joint Conference of Neural Networks*, 1989; pp 381−386.

(48) Kim, S.; Ji, W.; Deng, S.; Rackauckas, C. Stiff Neural Ordinary Differential Equations. **2021**, *arXiv:2103.15341*. arXiv Preprint.

(49) Owoyele, O.; Pal, P. ChemNODE: A Neural Ordinary Differential Equations Approach for Chemical Kinetics Solvers. **2020**, *arXiv:2101.04749*. arXiv Preprint.