Contents lists available at ScienceDirect

Energy and AI

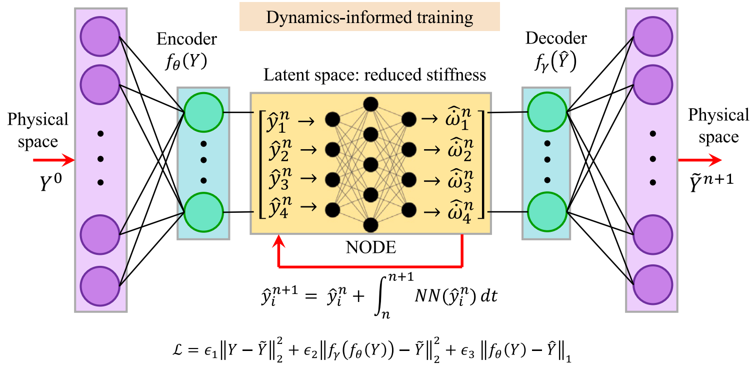
journal homepage: www.elsevier.com/locate/egyai

A data-driven reduced-order model for stiff chemical kinetics using dynamics-informed training

Vijayamanikandan Vijayarangan [a], Harshavardhana A. Uranakara [a], Shivam Barwey [b], Riccardo Malpica Galassi [c], Mohammad Rafi Malik [a], Mauro Valorani [c], Venkat Raman [d], Hong G. Im [a,*]

[a] CCRC, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
[b] Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, IL 60439, USA
[c] Mechanical and Aerospace Engineering Department, Sapienza University of Rome, Via Eudossiana, 18, Rome 00184, Italy
[d] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

GRAPHICAL ABSTRACT


Dynamics-informed training
Encoder $f_\theta(Y)$
Decoder $f_\gamma(\hat{Y})$
Latent space: reduced stiffness
Physical space
Y^0
Physical space
$\tilde{Y}^{n+1}$
$\hat{y}_1^n$ $\hat{y}_2^n$ $\hat{y}_3^n$ $\hat{y}_4^n$
$\hat{\omega}_1^n$ $\hat{\omega}_2^n$ $\hat{\omega}_3^n$ $\hat{\omega}_4^n$
NODE
$\hat{y}_i^{n+1} = \hat{y}_i^n + \int_n^{n+1} NN(\hat{y}_i^n)\,dt$
$\mathcal{L} = \epsilon_1 \|Y - \tilde{Y}\|_2^2 + \epsilon_2 \|f_\gamma(f_\theta(Y)) - \tilde{Y}\|_2^2 + \epsilon_3 \|f_\theta(Y) - \hat{Y}\|_1$

ARTICLE INFO

Keywords:
Stiff system
Chemical kinetics
Reacting flows
Autoencoders
Neural ODE

ABSTRACT

A data-based reduced-order model (ROM) is developed to accelerate the time integration of stiff chemically reacting systems by effectively removing the stiffness arising from a wide spectrum of chemical time scales. Specifically, the objective of this work is to develop a ROM that acts as a non-stiff surrogate model for the time evolution of the thermochemical state vector (temperature and species mass fractions) during an otherwise highly stiff and nonlinear ignition process. The model follows an encode-forecast-decode strategy that combines a nonlinear autoencoder (AE) for dimensionality reduction (encode and decode steps) with a neural ordinary differential equation (NODE) for modeling the dynamical system in the AE-provided latent space (forecasting step). By means of detailed timescale analysis by leveraging the dynamical system Jacobians, this work shows how data-based projection operators provided by autoencoders can inherently construct the latent spaces by removing unnecessary fast timescales, even more effectively than physics-based counterparts based on an eigenvalue analysis. A key finding is that the most significant degree of stiffness reduction is achieved through an end-to-end training strategy, where both AE and neural ODE parameters are optimized simultaneously, allowing the discovered latent space to be *dynamics-informed*. In addition to end-to-end training, this work highlights the vital contribution of AE nonlinearity in the stiffness reduction task. For the prediction of

[*] Corresponding author.
E-mail address: hong.im@kaust.edu.sa (H.G. Im).

https://doi.org/10.1016/j.egyai.2023.100325
Received 15 August 2023; Received in revised form 22 October 2023; Accepted 25 November 2023
Available online 30 November 2023

homogeneous ignition phenomena for H_2-air and C_2H_4-air mixtures, the proposed ROM achieves several orders-of-magnitude increase in the integration time step size when compared to (a) a baseline CVODE solver for the full-chemical system, (b) statistical technique – principal component analysis (PCA), and (c) computational singular perturbation (CSP), a vetted physics-based stiffness-reducing modeling framework.

1. Introduction

High-fidelity numerical simulations of turbulent reacting flows are one of the most demanding computational tasks due to (i) a large number of reactive scalar variables which translates into a large number of transport equations, and (ii) the wide range of spatio-temporal scales arising from chemical reactions and transport phenomena [1–4]. The Kolmogorov scales and flame thickness dictate the spatial resolution requirements, and the detailed chemical kinetic mechanism imparts complex nonlinear reaction source terms with a range of timescales from pico- to a few seconds arising from different elementary reaction rates. Such temporal stiffness is truly a major challenge that cannot be easily resolved by the common parallel computing framework.

According to the method of lines, the system of conservation equations are cast into a large number of coupled ordinary differential equations (ODEs). To address the disparities between the chemical and transport timescales, reacting flow simulations often resort to operator splitting strategies that decouple the effects of chemistry from transport, such that a larger simulation time step is used for integrating the transport processes (convective and diffusive timescales), while the stiff chemical source terms are integrated with a dedicated time integrator. This approach not only needs to deal with splitting errors carefully [5] but also does not eliminate the stiffness associated with a large spectrum of chemical timescales. For this, implicit schemes based on backward differentiation formula (e.g., CVODE [6]) are commonly used at a substantial computational overhead required for the operation of a large Jacobian matrix. Hardware-oriented acceleration pathways have been developed in recent years to alleviate this cost, such as GPU-optimal treatment of chemical source term evaluations [7,8] and load-balanced strategies for GPU-based time integration [9–11]. To achieve high-fidelity reacting flow simulations at large scales, however, advanced model-oriented solutions that accelerate chemical time integration by many orders of magnitude, without loss of fidelity, are needed.

Common approaches to accelerate chemistry computations are categorized as follows: (1) mechanism simplification by removing species/reactions, (2) tabulation of chemical source terms via composition-space partitioning or neural network-based strategies, and (3) reduced-order models (ROMs) to alleviate the stiffness in chemical timescales. It should be noted that conventional modeling approaches in all of these categories can be either physics-based or data-based [3]. Data-based methods benefit from being able to model highly complex physics present in either experimental or numerical high-fidelity datasets directly, albeit at the cost of a potentially expensive training stage. In recent years, the class of "physics-informed" data-based models has emerged, attempting to combine the two pathways to reduce model dependence on the training stage and physical configuration [12]. In case the model speedup benefit outweighs this training cost, data-driven methods based on machine learning have shown considerable promise over purely physics-based models in many reacting flow applications [13].

The first class of models consists of mechanism simplification strategies that seek to reduce the number of species and reactions in the detailed kinetic mechanisms, based on optimally preserving some macroscopic quantity of interest (e.g., ignition delay time). These "skeletal" kinetic mechanisms can be generated upfront [14–18] or on the fly [19] by eliminating certain species that are deemed less significant or negligible in the overall reaction process. Though the number of ODEs is reduced, the simplified system does not explicitly guarantee a reduction in chemical stiffness, since both fast and slow chemical pathways are typically retained. As such, chemical time integration often remains expensive even when skeletal mechanisms are utilized.

The second class of models accelerates evaluations of the expensive chemical source term using tabulation strategies. Physics-based approaches in this category, such as in-situ adaptive tabulation (ISAT) [20] and the PRISM approach [21], tabulate the thermochemical state space using chemical source term characterizations. For example, in ISAT, the region of influence of a tabulation point is governed by an ellipsoid defined by a singular value decomposition of the Jacobian of the chemical source term. Data-based tabulation approaches typically leverage artificial neural networks (ANNs) for memory-efficient accelerated source term tabulation [22–24], with recent efforts leveraging unsupervised data-based partitioning strategies (e.g., local principal component analysis [17,25], random forests [26], and K-means clustering [27]) to guide localization of tabulation strategies and other models.

The third class of models (ROMs or similar surrogates) leverage projection operations to transform the original dynamical system into an alternative representation that is easier to integrate by utilizing the reduction of dimensionality or stiffness, or both. A popular approach is the principal component analysis (PCA), in which linear projection operators derived from data are used to produce a low-dimensional latent representation of the thermochemical state [28,29]. These latent representations can then be integrated using the projected governing equations or data-based prognostic models. Autoencoders (AEs), a nonlinear generalization of PCA, have also been successfully used in recent work for similar dimensionality reduction and ROM tasks [30,31]. Still, conventional PCA and AE-based projections do not guarantee stiffness reduction, as the PCA method has no knowledge of the temporal dynamics and therefore cannot be expected to act on intrinsic system timescales.

On the other hand, surrogates based on temporal dynamics, such as the quasi-steady state assumption (QSSA) [32], directly reduce both the stiffness and size of the chemical system by replacing the ODEs for the *fast* variables with algebraic relations. Successful implementation of QSSA relies on user expertise in identifying the fast variables, and thus, the method cannot be generalized for larger systems. This issue was successfully addressed by the computational singular perturbation (CSP) technique [33,34], and similarly by the intrinsic low dimensional manifold (ILDM) approach [35]. By constructing CSP basis vectors of the Jacobian matrix of the local chemical source terms [36–38], the fast time scales originating from the chemical source terms are segregated by projecting the dynamics onto its local slow invariant manifold (SIM), therefore constructing a *slow* reduced-order model. The resulting non-stiff (algebraically constrained to the SIM) equations can be solved using an explicit ODE solver with a much bigger time step size, typically equal to the fastest of the slow intrinsic time scales. However, one of the major issues in this method is that the calculation of the projection basis becomes increasingly expensive for larger mechanisms. Machine learning strategies have recently been used to tabulate the evaluation of the CSP projection basis to reduce this cost [39], but ultimately, efficient computation of a local basis remains a challenge in achieving accelerated computing.

Earlier application of deep neural networks (DNNs) was mainly to predict the chemical source terms by taking the thermochemical state as an input [22]. In this approach, the training dataset consists of the thermochemical state and the corresponding source terms. The fully trained network is integrated with the flow solver and advanced in time using an explicit ODE solver. Due to the strong nonlinearities, however, the errors in the chemical source term predicted by the DNN accumulate over time and often result in failure of time integration. Recently, sequence models were used to improve stiff ODE integration and time evolution of the thermochemical state. In this regard, continuous time echo state networks are used to model the stiff chemical kinetics [40] and predict the thermochemical state at any time instant by adaptively changing the time step size. Alternatively, Ji et al. [41] proposed a PINN-based framework for the solution of chemical kinetics, where QSSA is employed for neural network training.

In the meantime, a different approach referred to as the neural ODEs (NODE) [42] have been adopted to solve the stiff chemistry ODEs [43]. The NODE replaces the memory-intensive backpropagation step with an adjoint sensitivity method, which results in efficient neural network training (with multiple parameters). This approach has been used [43] to solve the chemistry ODEs by assigning each chemical state with a neural network, as the native NODE implementation has difficulties in handling stiff systems, because solving an adjoint system as a final value problem results in instability during the training. To overcome this, Kim et al. [44] introduced a forward mode adjoint sensitivity method, but its computational cost hinders its feasibility for larger neural networks. These methods primarily focus on modeling stiff chemical systems without altering the degree of stiffness, thus still requiring small time step sizes or needing to rely on implicit ODE solvers. As an alternative to a stiff ODE solution, Dikemann et al. [30] combined autoencoder and neural ODE with a stiffness-based regularizer. This approach enhanced accuracy and achieved a maximum speedup of 3.1× for a perfectly stirred reactor model.

The above pioneering studies laid the foundation for employing data-driven methods for the accelerated solution of stiff chemical systems while preserving fidelity. In contrast to traditional techniques such as QSSA and CSP, which are theoretically founded and explicitly target reduced-order models by actively eliminating fast time scales, data-driven methods were able to learn global nonlinear coordinate transformations from the data that are capable of significantly reducing the stiffness of the system. However, questions such as how the data-driven methods achieve stiffness removal and nonlinear dimensionality reduction, and how the two objectives are inter-related, need to be better understood.

Motivated by the above, this study aims to develop a stiffness-eliminating data-based ROM that fits within the class-(3) above, by leveraging machine learning to simultaneously learn (a) a nonlinear mapping between the full-order thermochemical space and a low-dimensional latent space with larger characteristic timescales, and (b) a surrogate model for the dynamical system in this discovered non-stiff latent space. This is accomplished with a combination of autoencoders (AEs) and neural ODE (NODE) [42]. The goal is to ensure that the learned projection operator results in a latent space that eliminates unnecessarily fast timescales. In contrast to CSP, this work obtains a global stiffness-elimination basis from data through a *dynamics-informed* training strategy. In this context, *dynamics-informed* means that the dimensionality reduction process is informed by the latent dynamical evolution rule during the training procedure.

To this end, we train an autoencoder with neural ODE (AE+NODE) architecture to develop a non-stiff ROM for integrating the stiff chemistry ODEs. Employing the H_2-air [45] and C_2H_4-air [46] mechanisms and zero-dimensional homogeneous ignition reactor datasets, we examine: (1) the time integration accuracy of the obtained ROM compared to a classic implicit ODE solver (i.e., Cantera-based CVODE [47]), (2) the computational gain in terms of time step requirement compared to CVODE, PCA, and the CSP solver, (3) the intrinsic timescales developed by the model in the latent space, and (4) the mechanism of stiffness removal using the evolution of the state variables in the latent space. Furthermore, we explore the relationship between the dimensionality reduction method (linear vs. nonlinear AE), latent space dimension, number of hidden layers, and simultaneous vs. two-step training of AE and neural ODE on stiffness removal and computational gain.

The rest of the manuscript is organized as follows: Section 2 describes the computational methodology employed in two parts (theory and neural network architecture). The results are presented in Section 3 with a discussion of the important contributions of the present study. Finally, Section 4 summarizes the work and suggests possible future directions.

2. Methodology

This section describes the computational methodology in two subsections. The first part reviews the definition of stiffness in chemical kinetics, followed by the description of the CSP theory and PCA analysis. The second part discusses the neural network architecture employed, which is further organized into three subsections to describe (a) autoencoders and their working principles, (b) the concept of neural ODE as a direct integration by approximating the latent source terms, and (c) the neural network architecture combining autoencoders and neural ODE.

2.1. Stiffness, CSP, and PCA

2.1.1. Stiffness in chemical kinetics

A system of differential equations is considered stiff if it constitutes a wide range of time scales. Since the fast and slow chemical timescales act on the solution variables in a highly coupled manner, it is difficult to define and distinguish the fast and slow characteristics of different chemical species in a realistic system. Consider a linearized ODE system (Eq. (4)) in a given state point, $\psi' = \mathbf{J}\psi$, where $\mathbf{J}$ is the Jacobian matrix of the chemical source term given by

$$J_{i,j} = \frac{\partial f_i}{\partial \varphi_j} = \begin{bmatrix} \frac{\partial \dot{T}}{\partial T} & \frac{\partial \dot{T}}{\partial Y_1} & \cdots & \frac{\partial \dot{T}}{\partial Y_{N_{sp}}} \\ \frac{\partial \dot{Y}_1}{\partial T} & \frac{\partial \dot{Y}_1}{\partial Y_1} & \cdots & \frac{\partial \dot{Y}_1}{\partial Y_{N_{sp}}} \\ . & . & \cdots & . \\ \frac{\partial \dot{Y}_{N_{sp}}}{\partial T} & \frac{\partial \dot{Y}_{N_{sp}}}{\partial Y_1} & \cdots & \frac{\partial \dot{Y}_{N_{sp}}}{\partial Y_{N_{sp}}} \end{bmatrix} \tag{1}$$

having dimension $\left(N_{sp}+1\right) \times \left(N_{sp}+1\right)$. The local dynamics of the system are characterized by the eigenvalues and eigenvectors of $\mathbf{J}$, obtained by its spectral decomposition. The reciprocals of the eigenvalues λ_j of $\mathbf{J}$ represent the intrinsic timescales of the reacting system. The stiffness, S, of the system is defined as the ratio of the maximum (fastest scale) to the minimum (slowest scale) eigenvalues of $\mathbf{J}$:

$$S = \frac{\max_j \{|Re(\lambda_j)|\}}{\min_j \{|Re(\lambda_j)|\}_{\lambda_j \neq \{0\}}} = \frac{\tau_{slowest}}{\tau_{fastest}} \tag{2}$$

The stiff nature of the chemical kinetics yields a large value of S. In many complex combustion systems, the fastest time scale may be as small as 10^{-12} s, several orders of magnitude smaller than that of the transport processes. Nonetheless, typical chemical kinetic systems fall under a different class of ODE systems called the *singularly perturbed* system [48], where the fast and slow dynamics are largely separated into two groups. The fundamental approach to treating such multi-scale dynamics is to use a suitable coordinate transformation, which decouples the fast and slow dynamics. Then, computing the decay of fast motions aids in identifying an attractive slow invariant manifold (SIM), over which the slow dynamics evolve and thereby present an opportunity for a simplified treatment of the ODE problem. Such SIMs are well approximated by the eigenspaces of the linearized system. In the high-fidelity simulation of reacting flows, a different approach called operator splitting is employed to improve the overall time step size and is described as follows.

In an operator splitting framework [49], chemistry is separated from transport (convection + diffusion), and sub-cycling is employed to solve chemistry ODEs. This was commonly implemented in the direct numerical simulation (DNS), such as KAUST adaptive reacting

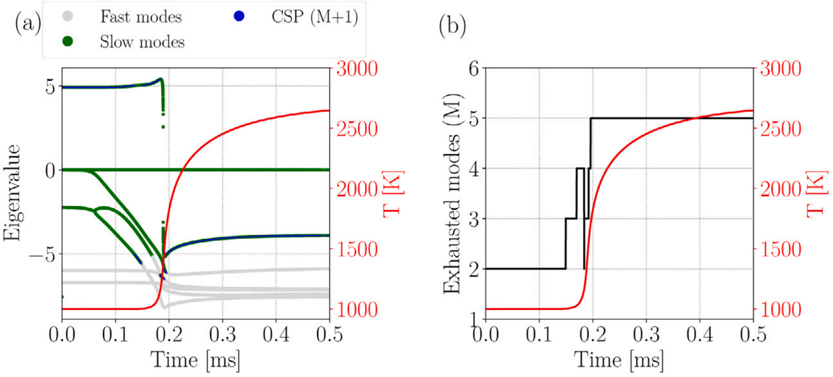


Fig. 1. Constant-pressure homogeneous ignition of H_2-air mixture at P = 1 atm, $\phi = 1.0$, and T_{init} = 1000 K. Time evolution of (a) eigenvalues of **J** and temperature (b) number of exhausted modes (M) and temperature. Eigenvalues associated with the fast/slow modes are shown in gray/green, (M+1)-th eigenvalue is shown in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

flow solver (KARFS) [50,51], and the speedup obtained by employing operator splitting with chemistry sub-cycling has been demonstrated. However, integrating these ODEs using an implicit solver consumes a significant portion of computational time (approximately 30% to 50%) and is limited by the time step size. The state variables governing the operator split chemistry are given by [52]:

$$\varphi = \left[T, Y_1, Y_2, \dots, Y_{N_{sp}}\right]^{\intercal} \tag{3}$$

and the nonlinear ODEs governing the time evolution of the above state variables are given by:

$$f = \frac{\partial\varphi}{\partial t} = \left[\frac{\partial T}{\partial t}, \frac{\partial Y_1}{\partial t}, \frac{\partial Y_2}{\partial t}, \dots, \frac{\partial Y_{N_{sp}}}{\partial t}\right]^{\intercal} \tag{4}$$

where

$$\begin{aligned} \frac{\partial T}{\partial t} &= -\frac{1}{\rho c_p}\sum_{k=1}^{N_{sp}} h_k W_k \dot{\omega}_k \\ \frac{\partial Y_k}{\partial t} &= \frac{1}{\rho} W_k \dot{\omega}_k \end{aligned} \tag{5}$$

and T is temperature, ρ is the mixture density, c_p is the mixture-averaged constant pressure specific heat, Y_k, h_k, W_k, $\dot{\omega}_k$ are the mass fraction, enthalpy, molecular weight, net production/consumption rate due to chemical reactions, respectively, of the kth species, and N_{sp} is the number of species and t is time.

2.1.2. CSP theory

The CSP method [53] is a widely known technique used to approximate local slow manifolds and, therefore, characterize the spectral behavior of kinetic systems. In the CSP method, the vector field is decomposed into

$$\frac{\partial\varphi}{\partial t} = f = a_1 h^1 + \cdots + a_M h^M + a_{M+1} h^{M+1} + \cdots + a_N h^N \tag{6}$$

where the CSP modes a_i can be approximated with leading order accuracy by the right eigenvectors of **J** and the mode amplitudes h^i are defined as $h^i := b^i \cdot f$, with b^i being the left eigenvectors of **J**. By definition, the right and left eigenvectors are orthonormal to each other such that $b^i \cdot a_j = \delta_{ij} = \mathbf{I}$, where δ_{ij} is Kronecker delta and **I** is the identity matrix, thus satisfying the equality in Eq. (6). Each mode amplitude in Eq. (6) evolves according to an exponential decay/growth exhibiting a time scale $\tau_i = 1/|\lambda_i|$. Based on the time scale magnitude, we arrange the terms in ascending order, given as $\tau_1 < \cdots < \tau_M \ll \tau_{M+1} < \cdots < \tau_N$. The CSP technique identifies [54] the spectral gap between τ_M and τ_{M+1}, therefore separating a fast subspace spanned by the eigenvectors $\{a_i\}_{i=1,\dots,M}$, and a slow subspace, spanned by $\{a_i\}_{i=M+1,\dots,N}$. The fast (exhausted) modes are defined as the modes having a vanishing amplitude within a $\mathcal{O}(\tau_{M+1})$ time frame, thus establishing a set of M algebraic constraints $h^i(\varphi) \approx 0$ that define a $(N-M)$-dimensional SIM in the thermo-chemical composition space. Consequently, a slow submodel $\left.\frac{\partial\varphi}{\partial t}\right|_{slow} \approx \sum_{M+1}^{N} a_i h^i$ evolves over the SIM with a driving (fastest of the slow) timescale of order τ_{M+1}.

The CSP method exploits a local (linearized) coordinate transformation that projects the physical variables φ, affected by the coupled fast/slow processes, into the intrinsic CSP variables $\eta_i := b^i \cdot \varphi$. Each linear combination η_i is decoupled in time from the others, being either a fast or a slow variable, with simple dynamics given by $\frac{d}{dt}\eta_i = \lambda_i \eta_i$. To draw a parallel with other projection-based methods, the CSP slow variables $\{\eta_i\}_{i=M+1,\dots,N}$ can be thought of as the latent variables in the slow embedding. However, because of the non-linearity of the vector field in the ODE system (Eq. (4)), the eigendecomposition of its linearized version varies slowly along a trajectory, implying that the local tangent space also rotates moving along the SIM so that the linear combinations defining the latent variables change as well. Fig. 1a shows the 10 eigenvalues[1] $\{\lambda_i\}_{i=1,\dots,N_{sp}+1}$ computed along an ignition trajectory of a hydrogen-air mixture in a constant-pressure homogeneous reactor. The eigenvalues are plotted in the form $log_{10}(abs[\lambda_{real}]) \times sign(\lambda_{real})$ in order to visualize both the magnitudes and signs simultaneously.

As expected, the eigenvalues are not constant but vary in time because of the nonlinearity of f, although they become nearly constant when the system approaches the equilibrium point. One positive eigenvalue shows up in the pre-ignition part of the trajectory, denoting an explosive character of the system. The CSP decomposition allows the separation of the eigenvalues associated with fast (gray) and slow (green) modes by recognizing the number M of fast modes. Therefore, the $(M+1)$-th eigenvalue (blue) denotes the fastest eigenvalue of the slow subspace. The temporal variation of the fast subspace dimension M implies that the slow manifold dimension (N-M) changes along the system trajectory. A smaller M implies that a smaller number of modes are exhausted and fast, and a larger number of modes are active and slow, and because of this, the driving time scale τ_{M+1} is faster, i.e., the dynamics on the SIM is faster and higher-dimensional. Fig. 1b shows that this happens around $t = 0.2$ ms when only 2 modes are exhausted and the slow dynamics advances with a time scale of $\mathcal{O}(10^{-6}s)$ over a 4-dimensional SIM. Note the presence of a positive eigenvalue in the initial part of the ignition trajectory. Likewise, a larger M implies that most of the degrees of freedom are exhausted and the slow dynamics are lower-dimensional. When the ignition phase is concluded ($t > 0.2$ ms), the system approaches its equilibrium and the modes progressively turn fast/exhausted, until the slow dynamics become one-dimensional (only 1 mode is active) towards the equilibrium point, with

[1] the actual number of non-zero eigenvalues is $N_{sp} + 1 - N_e - 1$ because the N_e atomic conservation laws and the enthalpy conservation of this specific model problem reveal themselves as zero eigenvalues. Therefore, we have 6 degrees of freedom.

a leading (unique) time scale of $\mathcal{O}(10^{-4}s)$. In summary, the dynamics of the hydrogen ignition problem, with a total of 6 degrees of freedom, evolves on a SIM whose dimension varies from 4 to 1.

As will be shown later, we resort to data-based projections using auto-encoders to identify an abstract latent space, which can reduce the system dimensionality *and* simultaneously eliminate the stiffness.

2.1.3. Principal component analysis (PCA)

Principal component analysis (PCA) is a statistical technique used to decrease the dimensionality of a system by identifying a reduced set of uncorrelated variables, called principal components (PCs), with a minimum loss of information [55]. The principal components are linear combinations of the original Q variables, and the first q PCs (with $q < Q$) allow for a low-dimensional representation of the system as they contain most of the information present in the original data. The scores of each PC are then transported in a numerical simulation, leading to a reduced set of equations [28].

Starting from a centered and scaled input data matrix $\mathbf{X}$, consisting of n observations of p variables, the PCs are obtained by means of an eigenvalue decomposition of the system's covariance matrix $\mathbf{S}$:

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X} \tag{7}$$

$$\mathbf{S} = \mathbf{A}\mathbf{L}\mathbf{A}^T. \tag{8}$$

The columns of the basis matrix $\mathbf{A}$ represent an orthonormal basis of eigenvectors (the PCs) and the diagonal elements of $\mathbf{L}$ are the associated eigenvalues. Thereafter, the original variables are recast into a new set of uncorrelated variables, the principal component scores $\mathbf{Z} = \mathbf{X}\mathbf{A}$. Those scores represent the latent variables in the PCA lower dimensional space. Given the orthonormality of $\mathbf{A}$ ($\mathbf{A}^T = \mathbf{A}^{-1}$), it is possible to uniquely recover the original variables from the latent variables, $\mathbf{X} = \mathbf{Z}\mathbf{A}^T$. The system dimensionality is reduced by truncating $\mathbf{A}$, i.e., by retaining only the first q PCs (with $q < Q$). Using the truncated basis matrix ($\mathbf{A}_q$), a reduced set of latent variables (scores) can be obtained $\mathbf{Z}_q = \mathbf{X}\mathbf{A}_q$, and the original variables can be reconstructed from the reduced dimensionality space $\mathbf{X} \approxeq \mathbf{X}_q = \mathbf{Z}_q\mathbf{A}_q^T$. The quality of the reduction is assessed by means of the low-rank approximation error $\|\mathbf{X}_q - \mathbf{X}\|_2$.

In essence, PCA discovers a new set of coordinates wherein the first coordinate exhibits the highest variance, and each subsequent coordinate has the maximum possible variance while being uncorrelated with the preceding ones.

We note that the manifolds discovered by the CSP and PCA projections are substantially different in principle. CSP is a dynamics-based local (pointwise) projection that exploits scale separation, while PCA is a global data-based projection that exploits correlations among the state variables. A stiffness reduction in the PCA projection is not expected, albeit possible due to fortuitous correlations between fast and slow processes, i.e. fast variables that are slaved to the slow dynamics, as in the case of SIMs. Nonetheless, the global and linear nature of the PCA projection contrasts with the search for a curvilinear SIM. In the present study, we explore autoencoders to inject non-linearity in the discovery of latent spaces from data analysis, with the final goal of driving the learning process with information from the system's dynamics.

2.2. Neural network-based models

2.2.1. Autoencoders (AE)

An autoencoder (AE) is a multilayer perceptron (MLP) that transforms the variables from an input space to a representational space, referred to as the *latent space*, with minimal distortion [56]. Fig. 2 shows the schematic of a single layer AE which consists of two components: (i) a feedforward encoder that maps/transforms the physical space variables ($\mathbf{Y} \in N_p$) to latent space variables ($\hat{\mathbf{Y}} \in N_L$) and (ii) a decoder which reconstructs the original input space variables from the latent space, both using a set of special operations. These special operations are through piece-wise non-linear activation functions, such that the decoder is not an inverse operation of the encoder.

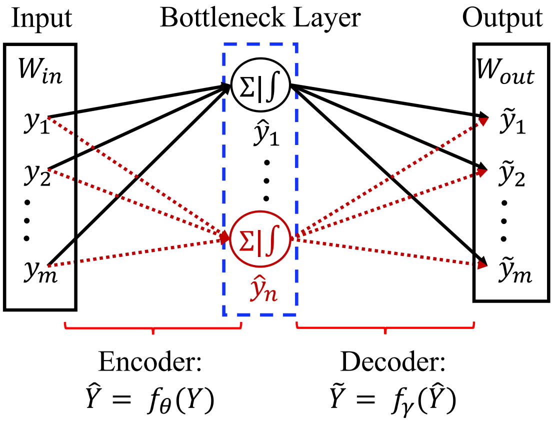


Fig. 2. Schematic representation of a single layer autoencoder.

Therefore, autoencoders decrease the dimension of the input state by projecting it onto a latent space with a dimension $N_L < N_p$. This property of AE has made them prevalent in the development of reduced-order models [57]. The reduced set of variables from the encoder is used to develop the surrogate models, which can reduce computational costs significantly.

This work focuses on a stacked autoencoder with a fully connected MLP architecture for both the encoder and decoder. Consider an input matrix $\mathbf{Y} \in \mathbb{R}^{N_p \times N_{samples}}$, whose columns are the input vectors. The output of the AE ($\tilde{\mathbf{Y}}$) is constrained to minimize the reconstruction error ($\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_2^2$) to match the input data $\mathbf{Y}$. Note that an autoencoder with linear activation functions along with a single hidden layer is closely related to PCA, although the latent coordinates are correlated and are not sorted in descending order of variance [58]. The main drawback of AE is that the latent space dimension has to be fixed during the training phase, unlike PCA which has the flexibility to choose the required number of latent dimensions *a posteriori*. As will be shown later, however, the non-linear AE has a strong advantage of discovering highly complex projections, which is not possible in any constrained/unconstrained linear projection techniques.

2.2.2. Neural ODE (NODE)

We propose to use neural ODE to approximate the latent source terms and integrate the ODEs in the latent space. Here, we briefly describe the working principle of the neural ODE architecture employed in this work. Neural ODE [42,59] is a deep learning-based framework used to approximate the continuous-time dynamics of a system governed by the ODE of the form:

$$\frac{dy(t)}{dt} = \mathbf{f}(y(t), t) \tag{9}$$

Employing any standard numerical method (e.g., Euler's method) to solve Eq. (9), time integration has to be performed from an initial time to the final time, for which $\mathbf{f}$ has to be evaluated as a function of both y and t given by the conservation equation. In the neural ODE framework, however, the hidden state $y(t)$ is the solution to the initial-value problem (IVP) and is governed by:

$$\frac{dy(t)}{dt} = \mathbf{f}_\beta(y(t), t) \tag{10}$$

where $\mathbf{f}_\beta$ denotes the dynamics of the hidden state and is modeled using a neural network with parameters β. Given an initial condition $y(t = 0)$, Eq. (10) can be integrated using any numerical ODE solver to obtain the solution $y(t)$ at the desired time t and to a required accuracy:

$$y_1, y_2, \dots, y_N = \text{ODESolve}(\mathbf{f}_\beta, y_0, (t_0, t_1, \dots, t_N)) \tag{11}$$

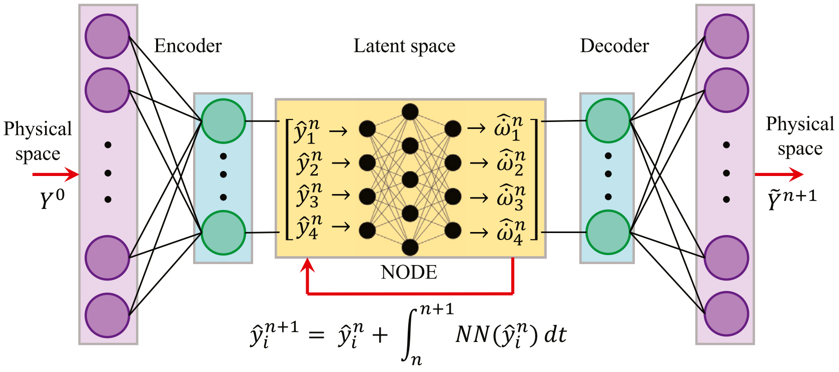


Fig. 3. Schematic representation of AE+neural ODE architecture.

To minimize the loss function with neural network parameters, neural ODE employs the backpropagation algorithm with an adjoint sensitivity method. The augmented system of the adjoint state along with the constraint and the modeled neural network function, $\mathbf{f}_\beta$, is integrated backward in time (final value problem) to compute the gradients. This results in memory-efficient training of the neural network.

2.2.3. Combined AE and neural ODE architecture

Fig. 3 shows the neural network architecture under study, where an AE is integrated with NODE. As explained in Section 2.2.1, the variables (T and Y_k's) are mapped from a physical to a latent space using an encoder. Then the neural ODE approximates the chemical source terms in the latent space and advances the latent state space from $\hat{Y}^n$ to $\hat{Y}^{n+1}$. We adopted the fourth-order Runge–Kutta (RK45) method for time integration in this study. Finally, the decoder recovers the variables ($\tilde{Y}_k^{n+1}$) in the physical space. The dimension of the input and output vectors of the encoder is given by N_p and N_L, respectively, and vice versa for the decoder. Therefore, N_L latent variables are integrated within the neural ODE algorithm.

In the present study, the reduced latent state with N_L variables and its evolution is given by,

$$\hat{\varphi} = \left[\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_{N_L}\right]^\intercal \tag{12}$$

$$\hat{f} = \frac{\partial\hat{\varphi}}{\partial t} = \left[\frac{\partial\hat{Y}_1}{\partial t}, \frac{\partial\hat{Y}_2}{\partial t}, \ldots, \frac{\partial\hat{Y}_{N_L}}{\partial t}\right]^\intercal \tag{13}$$

where the latent variables $\hat{Y}_k$'s are obtained by a nonlinear combination of physical space variables in Eq. (3), using an encoder. Therefore, the one-to-one correspondence of Eqs. (3) and (12) and Eqs. (4) and (13) is not guaranteed. The resulting Jacobian in the latent space is given by,

$$\hat{J}_{i,j} = \frac{\partial\hat{f}_i}{\partial\hat{Y}_j} = \begin{bmatrix} \frac{\partial\hat{\omega}_1}{\partial\hat{Y}_1} & \frac{\partial\hat{\omega}_1}{\partial\hat{Y}_2} & \ldots & \frac{\partial\hat{\omega}_1}{\partial\hat{Y}_{N_L}} \\ \frac{\partial\hat{\omega}_2}{\partial\hat{Y}_1} & \frac{\partial\hat{\omega}_2}{\partial\hat{Y}_2} & \ldots & \frac{\partial\hat{\omega}_2}{\partial\hat{Y}_{N_L}} \\ . & . & \ldots & . \\ \frac{\partial\hat{\omega}_{N_L}}{\partial\hat{Y}_1} & \frac{\partial\hat{\omega}_{N_L}}{\partial\hat{Y}_2} & \ldots & \frac{\partial\hat{\omega}_{N_L}}{\partial\hat{Y}_{N_L}} \end{bmatrix} \tag{14}$$

whose dimension is $N_L \times N_L$.

The PyTorch [60] implementation of neural ODEs [61] is used to train the neural network consisting of 5 hidden layers with one input and one output layer in each encoder, decoder, and neural ODE. Each hidden layer consists of 100 neurons with the exponential linear unit (ELU) activation to model the nonlinear reaction rates. In this entire study, the feed-forward neural network with fully connected layers is used without any additional dropout or regularization techniques (L1/L2 penalties, etc.). Linear activation units are used in the bottleneck/output layers of the encoder, decoder, and neural ODE. The parameters of the network are optimized with the Adam optimizer. The total loss function is given by

$$Loss = \varepsilon_1\mathcal{L}_1 + \varepsilon_2\mathcal{L}_2 + \varepsilon_3\mathcal{L}_3 \tag{15}$$

where $\mathcal{L}_1 = \|Y - \tilde{Y}\|_2^2$ is the combined encoder+NODE+decoder reconstruction error, $\mathcal{L}_2 = \|f_\gamma(f_\theta(Y)) - \tilde{Y}\|_2^2$ is the encoder+decoder error without NODE, and $\mathcal{L}_3 = \|f_\theta(Y) - \hat{\tilde{Y}}\|_1$ is the encoder+NODE error without decoder, and all the weights ε are set to 1 (i.e., $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 1$). $\mathcal{L}_1$ minimizes the error between ground truth (Y_k) and the AE+NODE predictions ($\tilde{Y}_k$). The additional loss terms $\mathcal{L}_2$ and $\mathcal{L}_3$ ensure the mapping of the encoder and decoder is bijective (one-to-one correspondence) or unique. This implies the trajectories of the physical variables can be reproduced uniquely from the latent state. A similar loss function has been incorporated in Ref. [31].

The loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$ are minimized by evaluating mean squared error (MSE) between the ground truth and the prediction, which is given by Eq. (16). However, $\mathcal{L}_3$ is minimized by evaluating the L^1-norm, which sparsify the latent variables and improves the rate of convergence. Furthermore, in the results section, the MSE combined with the relative root mean squared error (RRMSE) [62], given by Eq. (17), are used to validate the accuracy of the AE+NODE-based ROM predictions ($\tilde{\varphi}$) against the data obtained from Cantera simulation (φ) with n number of observations.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\varphi^i - \tilde{\varphi}^i\right)^2 \tag{16}$$

$$RRMSE = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\varphi^i - \tilde{\varphi}^i\right)^2}}{\frac{1}{n}\sum_{i=1}^{n}\varphi^i} \times 100 \tag{17}$$

3. Results and discussions

This section describes the predictions from the AE+NODE-based ROM, analyzed in terms of accuracy and performance compared to direct, physics-based, and data-driven integration methods. The main objective is to provide an understanding of the underlying mechanism of the computational gain obtained in time step size by employing the AE+NODE framework. The results are presented in five subsections: (i) the data acquisition and training of the neural network are explained; (ii) results obtained from the present neural network architecture are validated with Cantera; (iii) the computational gain obtained in the time step size is described using eigenspectrum analysis; (iv) a comparison between linear and nonlinear AE with NODE is made using the eigenvalue analysis; and (v) the impact of the latent space dimension (N_L) and the number of hidden layers (H) in the nonlinear AE on the stiffness reduction and time step gain is presented.

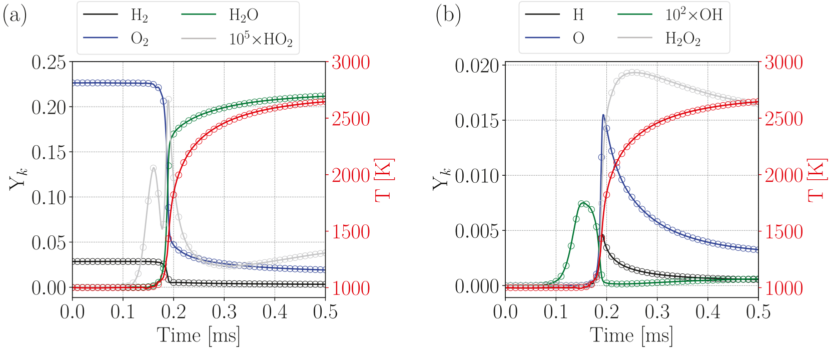


Fig. 4. Time evolution of physical state variables during the constant-pressure homogeneous ignition of H_2-air mixture (P = 1 atm, ϕ = 1.0, and T$_{init}$ = 1000 K): Cantera (circle) and E2E nonlinear AE+NODE (solid line).

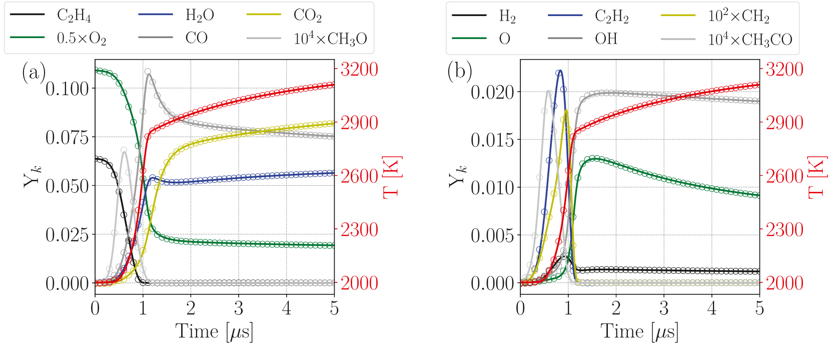


Fig. 5. Time evolution of physical state variables during the constant-pressure homogeneous ignition of C_2H_4-air mixture (P = 10 atm, ϕ = 1.0, and T$_{init}$ = 2000 K): Cantera (circle) and E2E nonlinear AE+NODE (solid line).

Table 1
Thermodynamic conditions used to compute the constant-pressure homogeneous ignition curve data for training, validation, and testing.

Mixture	P [atm]	T$_{init}$ [K]	ϕ
H_2-air	1 (Fixed)	1000 – 2000 ΔT = 100	0.5 – 1.5 $\Delta\phi$ = 0.02
C_2H_4-air	10 (Fixed)	1500 – 2000 ΔT = 100	1.0 – 1.4 $\Delta\phi$ = 0.02

3.1. Data generation and training

To generate the training data, two different chemical mechanisms are considered: (i) H_2-air (10 species and 27 reactions) [45] and (ii) C_2H_4-air (30 species and 231 reactions) [46,63]. As a reference, the equations for each mixture are integrated using a homogeneous constant-pressure batch reactor to obtain ignition trajectories, using Cantera [47].

Table 1 shows the initial conditions employed to generate the training dataset, where Δ is the sampling interval in the initial temperature/composition space. 561 and 126 datasets are generated for H_2-air and C_2H_4-air mixtures, respectively, covering the range specified in Table 1. The overall computational cost required to generate the training datasets is 35.793 s for H_2-air, and 74.507 s for C_2H_4-air. The datasets from these simulations consist of temperature and species mass fraction profiles, which are accessed randomly to generate the encoder input matrix. Furthermore, these datasets are split in the ratio 75:15:10 for training, testing, and validation, respectively, and normalized (by subtracting the mean and dividing by the standard deviation) to accelerate the training process. The inert species, N_2 and Ar for the H_2-air mixture and N_2 for the C_2H_4-air mixture, are not included in the training/testing/validation datasets.

Two different AE+NODE training frameworks are tested for the performance comparison: (i) end-to-end (E2E) training with linear and nonlinear AE, which involves simultaneous optimization of the AE and NODE parameters from a random initialization and (ii) step-by-step (SBS) training with a nonlinear AE, in which the parameters of an AE are first optimized to obtain the latent variables, and subsequently used as input for NODE parameters optimization (two-step training/optimization). For this initial comparison, the bottleneck size (latent space dimension) is fixed to N_L = 5 and H = 5. The training was performed using double precision floating point on a single NVIDIA Quadro RTX 4000 GPU and each of these training frameworks consumed about 5 to 8 h of computational time to reach the optima.

3.2. Validation of AE+NODE architecture

As a first step, we validate the E2E-trained nonlinear AE+NODE predictions against the Cantera results. Figs. 4 and 5 compare the T and Y_k profiles for H_2-air (P = 1 atm, ϕ = 1.0, and T$_{init}$ = 1000 K) and C_2H_4-air (P = 10 atm, ϕ = 1.0, and T$_{init}$ = 2000 K) mixtures, respectively, obtained from the integration of constant pressure batch reactor. Similarly, Tables 2 (H_2-air) and 3 (C_2H_4-air) list the MSE (Eq. (16)) and the RRMSE (Eq. (17)) compared to the baseline Cantera solution. These conditions are not part of the training dataset.

The solutions obtained by the Cantera and AE-NODE (N_L = 5 and H = 5) integration are nearly identical for both mixtures shown in Figs. 4 and 5 and reveal that the newly proposed ROM is able to accurately reproduce the ignition phenomena. Tables 2 and 3 further show a maximum RRMSE of ∼4.5% and ∼9%, respectively, which are within acceptable limits.

3.3. Computational efficiency

Now we discuss the computational efficiency gain in terms of time step size (Δt) by using AE+NODE (N_L = 5 and H = 5). The explicit fourth-order Runge–Kutta with error control (RK45) ODE integrator [64] is used in all three different training methods with absolute

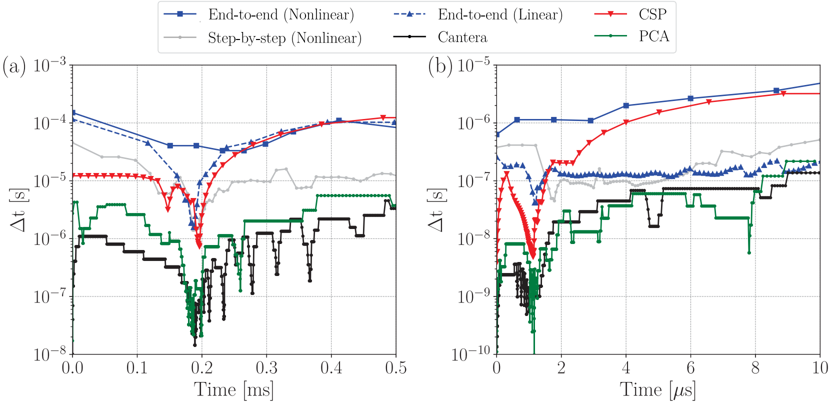


Fig. 6. Time evolution of the *Δt* during the constant-pressure homogeneous ignition phase for various methods/ROMs (a) P = 1 atm, ϕ = 1.0, and T_{init} = 1000 K using H_2-air mixture (b) P = 10 atm, ϕ = 1.0, and T_{init} = 2000 K using C_2H_4-air mixture.

Table 2
MSE and RRMSE of predicted thermochemical state from nonlinear AE+NODE (E2E) compared to Cantera — for the constant-pressure homogeneous ignition of H_2-air at P = 1 atm, ϕ = 1.0, and T_{init} = 1000 K.

T/Y_k	T [K]	H_2	H_2O	OH	HO_2
MSE	26.42	1.668e−08	6.638e−07	7.745e−09	1.830e−13
RRMSE [%]	0.201	1.759	0.494	0.420	4.596

Table 3
MSE and RRMSE of predicted thermochemical state from nonlinear AE+NODE (E2E) compared to Cantera — for the constant-pressure homogeneous ignition of C_2H_4-air at P = 10 atm, ϕ = 1.0, and T_{init} = 2000 K.

T/Y_k	T [K]	CH_4	CO_2	CO	CH_3CO
MSE	18.31	6.587e−11	1.497e−07	2.529e−07	1.969e−16
RRMSE [%]	0.160	4.162	0.697	0.646	9.124

Table 4
Comparison of time step size during the constant-pressure homogeneous ignition phase for different time integration methods.

Mixture	Cantera (CVODE) (or) PCA	CSP	AE+NODE (N_L = 5 and H = 5)		
			E2E nonlinear	E2E linear	SBS nonlinear
H_2-air	1e−8 s	1e−6 s	1e−4 s	1e−6 s	1e−6 s
C_2H_4-air	1e−10 s	1e−8 s	1e−6 s	1e−7 s	1e−7 s

error ε_{abs} = 10^{-5} and relative error ε_{rel} = 10^{-3}. It is worth mentioning that the AE+NODE integrates the transformed ODEs in the latent space and the integrated variables are not the same as the physical state. The AE+NODE time step size is then compared against that used in three different integration methods: (1) the intrinsic Δt_{CVODE} provided by the CVODE solver (in its Cantera implementation) using tolerances ε_{abs} = 10^{-16} and ε_{rel} = 10^{-9}, (2) $\Delta t_{CSP} = \tau^{M+1}$, the fastest timescale of the slow modes used in the CSP solver along with RK4 [39], and (3) Δt_{PCA} estimated by CVODE while integrating the first 8 PCA scores[2].

Figs. 6a and b compare the evolution of the *Δt* for the H_2-air and C_2H_4-air mixtures, respectively, obtained from Cantera, CSP, PCA, and the three different AE+NODE training frameworks. During the ignition phase (t ≈ 0.2 ms for H_2-air and t ≈ 1 μs for C_2H_4-air), *Δt* becomes significantly small due to the inherent stiffness of the system. The minimum *Δt* for different integration methods are tabulated in Table 4. As the CVODE or PCA-based methods do not inherently reduce the system's stiffness, the required *Δt* is very small and approaches the fastest time scale of the system. On the other hand, CSP uses a local linear projection to eliminate fast time scales, resulting in nearly two orders-of-magnitude larger *Δt* compared to the CVODE solver. Among the AE+NODE methods, the E2E-trained nonlinear AE+NODE ROM achieves the best performance, with *Δt* approximately two orders-of-magnitude larger compared to the CSP solver, the best analytical ROM. Both E2E linear and SBS nonlinear integrators perform comparably with the CSP solver up to the point of ignition, while requiring an order-of-magnitude smaller *Δt* in the post-ignition phase, although the E2E linear integrator performs better in the simpler H_2-air mixture. In summary, the E2E nonlinear AE+NODE integrator shows the overall best performance, significantly better than the theoretically optimized CSP solver.

It is remarkable to find that the E2E training with nonlinear transformation is most effective in eliminating the stiffness without resorting to rigorous analysis of characteristic time scales (as in CSP) and without loss of solution fidelity. This leads to two important postulates about the AE+NODE process. First, the E2E training with the tolerance constraints results in an intelligent selection of the latent space dynamics while abandoning unnecessary fast-time dynamics. This process is referred to as a *dynamics-informed* construction of latent space. In other words, the data-based projection of variables onto the latent space undergoes a process equivalent to constraining the latent variables onto a slow manifold that is necessary to capture the essential dynamics. Secondly, within the E2E training, the *nonlinear* transformation is crucial in eliminating the stiffness. An important finding of the present study is that the performance gain achieved using nonlinear AEs with E2E training is enabled by the nonlinear projection capable of identifying a non-stiff curved SIM (latent space), equivalent to the one obtained with the rotation of the CSP basis vectors, which cannot be retrieved using linear AEs. The exact physical and mathematical reasoning behind the effect of the nonlinear AEs needs further investigation.

The stiffness reduction in the latent space can also be visualized by examining the evolution of the eigenvalue spectrum of **J** and $\hat{\mathbf{J}}$. Figs. 7a and b compare the eigenvalues of the chemical Jacobian in both physical (**J**, Eq. (1)) and latent ($\hat{\mathbf{J}}$, Eq. (14), and N_L = 5) spaces. The largest eigenvalue in the CSP slow mode, CSP (M+1), is also overlaid. These eigenvalues are plotted in the form $log_{10}(abs[\lambda_{real}]) \times sign(\lambda_{real})$, which represents the inverse of the intrinsic timescales. A wider range of the eigenvalue magnitudes implies a greater degree of stiffness. The largest eigenvalue (in absolute value) corresponds to the *Δt* employed by an explicit time integrator. It is seen that the range of the eigenvalue spectrum shrinks/narrows in the latent space compared to the physical space. During the ignition phase, the range of eigenvalues is stretched in the physical space, but this behavior is attenuated in the latent space. The maximum latent eigenvalue is always found to be smaller than

[2] At least 8 components out of 10 were required for this system to obtain an accurate solution

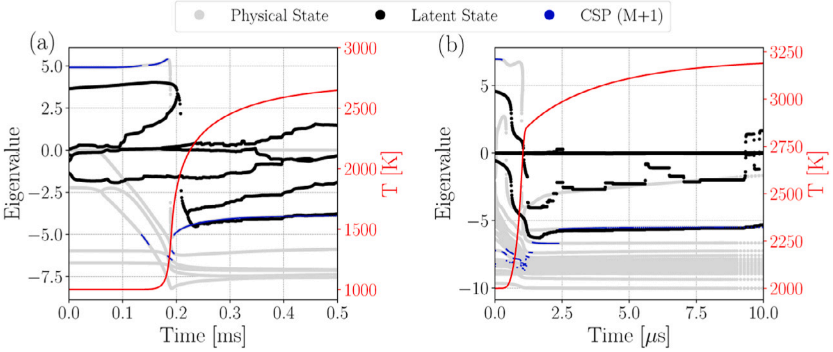


Fig. 7. Eigenspectrum of the chemical Jacobian in physical space (**J**) vs. latent space ($\hat{\mathbf{J}}$) (E2E nonlinear AE (N_L = 5)+NODE) during the constant-pressure homogeneous ignition of (a) H_2-air at P = 1 atm, ϕ = 1.0, and T_{init} = 1000 K (b) C_2H_4-air at P = 10 atm, ϕ = 1.0, and T_{init} = 2000 K.

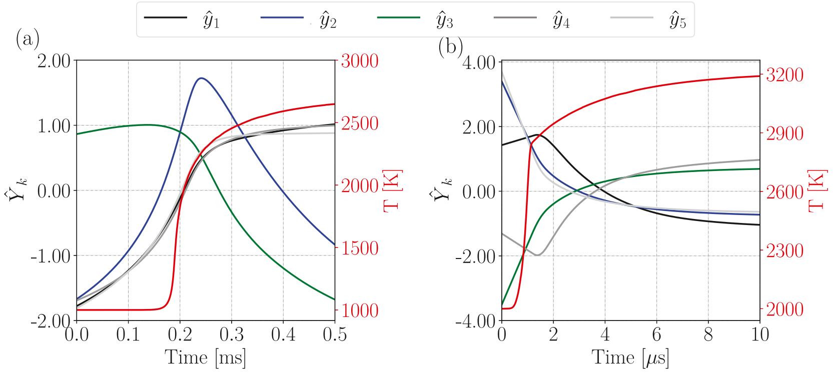


Fig. 8. Time evolution of variables in the latent space for the constant-pressure homogeneous ignition of (a) H_2-air (P = 1 atm, ϕ = 1.0 and T_{init} = 1000 K) and (b) C_2H_4-air (P = 10 atm, ϕ = 1.0, and T_{init} = 2000 K) mixtures during the ignition run – E2E-trained nonlinear AE (N_L = 5)+NODE.

that in the physical space, as well as the (M+1)-th CSP eigenvalue. This further confirms that the AE-NODE training process compresses the range of eigenvalues and smoothens their temporal evolution in the latent space.

Therefore, it is concluded that the E2E training with nonlinear dimensionality reduction is the preferred choice in developing a ROM for accelerated computation of the dynamics arising from nonlinear chemical reactions.

Figs. 8a and b plots the evolution of solution variables (N_L = 5) in the latent space for H_2-air and C_2H_4-air mixtures, respectively, with the E2E nonlinear AE+NODE. It is visually evident that the variables evolve much more smoothly in the latent space, without sharp peaks or troughs, compared to the physical space (see Figs. 4 and 5 for comparison). Despite the significantly smaller number of latent variables, the ignition dynamics were reproduced accurately with a significant time step gain. This implies the data-based construction of latent space is able to capture the essential system dynamics with a significantly smaller number of degrees of freedom and a lower stiffness.

3.4. Impact of linear vs. nonlinear AE with NODE

This section quantitatively compares the effect of the number of hidden layers (H) in the encoder and decoder on stiffness reduction, in order to provide guidelines for designing autoencoders using the E2E training in chemically reacting systems. Fig. 9 plots the eigenvalues for the different hidden layers generated by linear and nonlinear AE+NODE during the ignition of H_2-air mixture. For a linear AE+NODE (Figs. 9a and b), the range of eigenvalues becomes narrower in latent space compared to physical space, while the level of compression barely changes with H = 1 and 5. It is observed that the positive maximum eigenvalue in physical and latent spaces coincide, whereas the largest negative values are reduced in the latent space. This behavior suggests that although linear AEs are effective in reducing stiffness (as seen in Fig. 6), using a single hidden layer results in higher prediction errors.

In contrast, Figs. 9c–f compares the eigenvalues for 1–4 hidden layers, respectively, with the use of a nonlinear AE+NODE. With H = 1, nonlinear AE behaves similarly to linear AE. However, an increase in the number of hidden layers from 2 to 4 successively reduces the range of the eigenvalues, resulting in a less stiff latent space. The nonlinear AE with H = 4 shows much better performance in stiffness removal compared to the linear AE with H = 5, demonstrating the superior performance of the nonlinear reduction. Adding more hidden layers in a linear AE will produce the same effect of H = 1 as all the layers can be collapsed into one due to the linearity. Contrarily, in a nonlinear AE, adding an additional hidden layer produces a different outcome due to the nonlinearity as explained above. Therefore, the number of hidden layers should be chosen to achieve better accuracy and a maximum degree of stiffness reduction. In the following subsection, the impact of the number of hidden layers on the solution accuracy is systematically studied.

3.5. Effect of latent dimension and hidden layers on time step size

The effect of latent space dimensions and the number of hidden layers on stiffness removal is further assessed by using the E2E-trained nonlinear AE with NODE. This exercise was performed for the H_2-air mixture at P = 1 atm, ϕ = 1.0, and T_{init} = 1000 K. Fig. 10 compares the time steps size evolution during the ignition sequence of H_2-air mixture for (a) varying latent dimension (N_L = 2, 4, 6, and 8) for a fixed H(=5) and (b) varying the number of hidden layers (H = 1 to 4) for a constant N_L(=5). Fig. 10a shows that different latent space dimension has little effect on Δt, while a smaller latent dimension results in a larger prediction error as shown in Table 5. Therefore, the latent space dimension has to be chosen to minimize

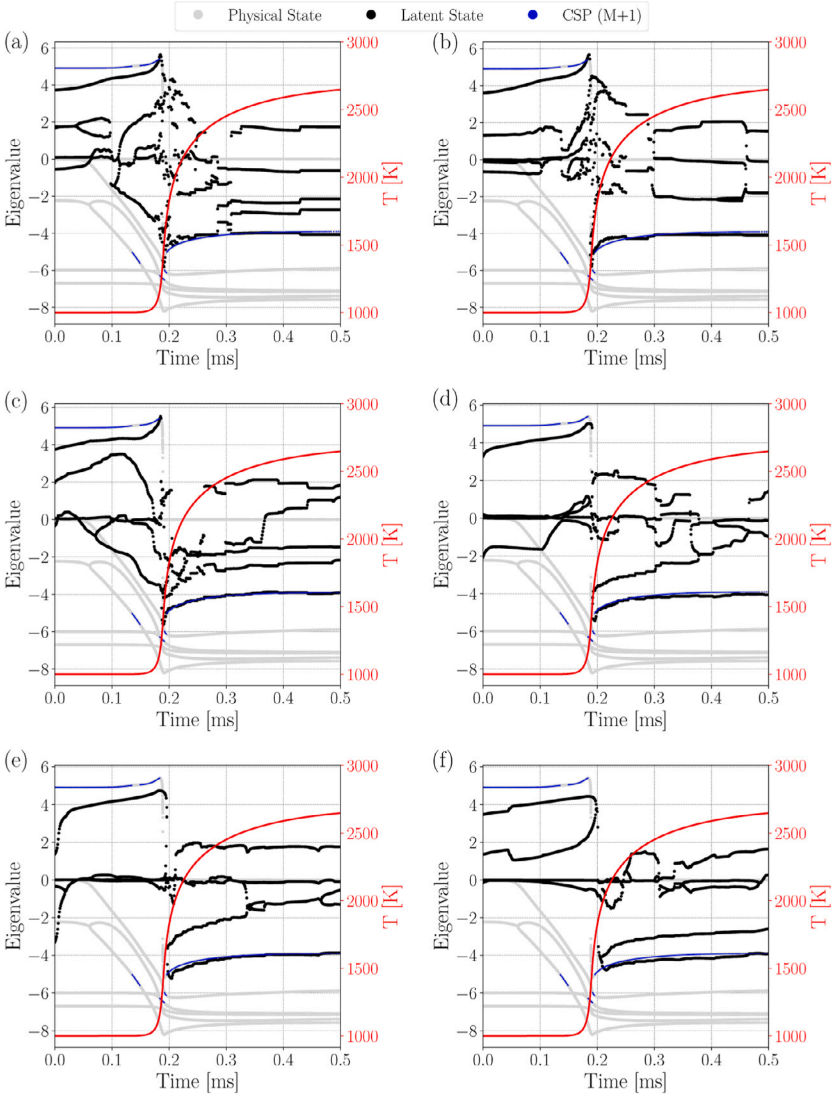


Fig. 9. Time evolution of eigenvalues in the latent space ($\hat{\mathbf{J}}$, black) vs physical space ($\mathbf{J}$, gray) for the constant-pressure homogeneous ignition of H_2-air mixture at $\phi = 1.0$, T_{init} = 1000 K. Top row — linear AE+NODE with (a) $H = 1$ and (b) $H = 5$; middle row — nonlinear AE+NODE with (c) $H = 1$ and (d) $H = 2$; bottom row — nonlinear AE+NODE with (e) $H = 3$ and (f) $H = 4$.

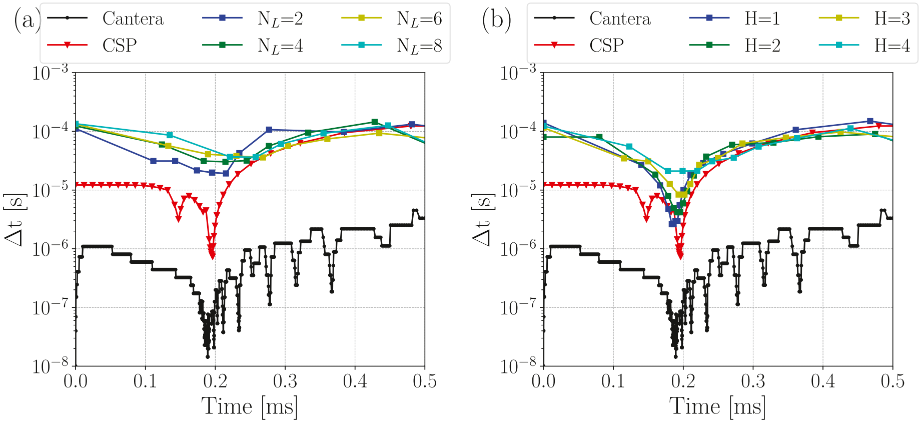


Fig. 10. Time evolution of Δt during the constant-pressure homogeneous ignition of H_2-air at P = 1 atm, $\phi = 1.0$, and T_{init} = 1000 K using nonlinear AE+NODE with (a) varying latent space dimension (N_L) with $H = 5$ and (b) varying hidden layers (H) with $N_L = 5$.

Table 5
RRMSE[%] of the predicted thermochemical state from nonlinear AE+NODE (E2E) (for varying N_L and fixed $H = 5$) compared to Cantera — for H_2-air mechanism at $P = 1$ atm, $\phi = 1.0$, and $T_{init} = 1000$ K.

N_L	RRMSE				
	T [K]	Y_{H_2}	Y_{H_2O}	Y_{OH}	Y_{HO_2}
2	0.900	6.425	1.603	1.783	12.489
4	0.115	0.840	0.191	0.300	2.559
5	0.201	1.759	0.494	0.420	4.596
6	0.091	0.651	0.177	0.254	1.832
8	0.174	1.883	0.515	0.433	0.515

Table 6
RRMSE[%] of the predicted thermochemical state from nonlinear AE+NODE (E2E) (for varying H and fixed $N_L = 5$) compared to Cantera — for H_2-air mechanism at $P = 1$ atm, $\phi = 1.0$, and $T_{init} = 1000$ K.

H	RRMSE				
	T [K]	Y_{H_2}	Y_{H_2O}	Y_{OH}	Y_{HO_2}
1	1.220	4.050	0.856	1.099	54.543
2	0.270	2.968	0.949	0.734	9.216
3	0.305	2.174	0.727	0.710	4.809
4	0.142	0.872	0.220	0.301	2.433
5	0.201	1.759	0.494	0.420	4.596

the prediction error. In contrast, the different number of hidden layers (H) has a larger impact on Δt as shown in Fig. 10b. AE with one hidden layer performs similarly as the CSP solver, whereas a large H AE shows a significant improvement in Δt compared to CVODE ($\mathcal{O}(4)$) and CSP ($\mathcal{O}(2)$) solvers. The prediction errors (RRMSE) associated with the varying hidden layers are tabulated in Table 6, which shows that although a shallow AE provides the computational gain over the CVODE/CSP solvers, the resulting prediction errors may be large. This suggests that deeper networks with optimum latent space dimension are required to disentangle the complex representation and result in a significant computational gain. However, the effect of other important aspects such as the learning rate, activation function, batch size, etc. will be investigated in the follow-up work.

4. Conclusions and future work

An efficient data-based reduced-order model is developed by combining an autoencoder (AE) with neural ODE (NODE) to integrate stiff chemistry ODE systems. The AE+NODE framework was used to integrate the homogeneous reactor consisting of two different fuel–air mixtures. The resulting AE+NODE predictions were found to be in good agreement with the Cantera-based results, exhibiting adequate mean squared error and relative root mean squared error.

The computational gain (in terms of Δt) obtained using AE+NODE has been studied using both linear and nonlinear AE combined with NODE for end-to-end and step-by-step training methods. Among various options, the end-to-end trained nonlinear AE with neural ODE was found to be most effective in accelerated integration with the maximum level of stiffness removal. With the same level of solution fidelity, the integration time step size obtained from the E2E-trained nonlinear AE with neural ODE was found to be four and two orders of magnitude larger compared to the CVODE and CSP solvers, respectively. In comparison, the E2E-trained linear AE+NODE and SBS-trained nonlinear AE+NODE resulted in moderate improvements, with Δt comparable to or somewhat larger than that with the CSP solver.

The strong performance by the E2E-trained nonlinear AE+NODE is attributed to the inherent characteristics of a *dynamics-informed* construction of the latent space, identifying an appropriate SIM without rigorous analytical computation. The analysis of the eigenvalue spectrum of the chemical Jacobian further confirmed that an E2E-trained nonlinear AE, effectively compresses and smoothens the eigenvalue spectrum in the latent space, leading to an effective stiffness removal. The more detailed parametric analysis yielded that the E2E-trained nonlinear AE+NODE with five hidden layers provides an optimal combination for eliminating stiffness. The latent space dimension in the E2E-trained nonlinear AE+NODE was found to have little effect on the stiffness removal, while it needs to be chosen to minimize the prediction error. In contrast, the number of hidden layers was found to affect Δt significantly, with a larger number of hidden layers progressively removing the stiffness further. For a larger number of species $\sim \mathcal{O}(10^3)$ and reactions $\sim \mathcal{O}(10^4)$, the time required to obtain the training datasets and training costs increases and demands a distributed computing framework for training. However, the computational gain achieved using the AE+NODE ROM is expected to far outweigh the training cost requirements, leading to significant improvement in overall computational efficiency.

As final remarks, the present study provided explanations about how the data-driven ROM for stiff ODE solvers eliminates fast timescales. For some test cases under study, it was found to outperform the physics-based CSP solver. However, one must not hasten to conclude that the AE approach is absolutely superior and makes the rigorous analytical approach obsolete. For one, the overall work required for the AE+NODE as a comprehensive predictive tool for a wide range of conditions can be highly demanding. More importantly, considering the performance difference between the linear and nonlinear AE in the present study, it is postulated that the improved performance is mainly attributed to the nonlinear mapping process, which is most effective during the induction period of ignition when the nonlinearity in the physical dynamics is large. This implies that the CSP solver may also be able to enhance its performance by employing nonlinear mapping into the algorithm. The generalization of the present AE+NODE approach to more complex systems of partial differential equations and chemical reactions, along with a more systematic assessment of its performance against the analytical approach, needs to be further investigated.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was sponsored by the King Abdullah University of Science and Technology (KAUST), Saudi Arabia and utilized the computational resources of the KAUST Supercomputing Laboratory (KSL). SB acknowledges support from the Argonne Leadership Computing Facility, which is a U.S. Department of Energy Office of Science User Facility operated under contract DE-AC02-06CH11357. VR acknowledges the support of ONR, United States Grant No. N00014-21-1-2475 with Dr. Eric Marineau as Program Manager. MV and RMG acknowledge the financial support from the Italian Ministry of University and Research in the framework of the project "National Center for High Performance Computing, Big Data and Quantum Computing" funded by European Union – NextGenerationEU.

References

[1] Oran ES, Boris JP. Numerical approaches to combustion modeling. New York, NY (United States): American Institute Aeronaut and Astronaut; 1991.
[2] Poinsot T, Veynante D. Theoretical and numerical combustion. RT Edwards, Inc.; 2005.
[3] Raman V, Hassanaly M. Emerging trends in numerical simulations of combustion systems. Proc Combust Inst 2019;37(2):2073–89.
[4] Raman V, Prakash S, Gamba M. Nonidealities in rotating detonation engines. Annu Rev Fluid Mech 2023;55:639–74.

[5] Lanser D, Verwer JG. Analysis of operator splitting for advection–diffusion–reaction problems from air pollution modelling. J Comput Appl Math 1999;111(1–2):201–16.

[6] Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, et al. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. ACM Trans Math Softw 2005;31(3):363–96.

[7] Barwey S, Raman V. A neural network-inspired matrix formulation of chemical kinetics for acceleration on GPUs. Energies 2021;14(9):2710.

[8] Uranakara HA, Barwey S, Pérez FEH, Vijayarangan V, Raman V, Im HG. Accelerating turbulent reacting flow simulations on many-core/GPUs using matrix-based kinetics. Proc Combust Inst 2022.

[9] Curtis NJ, Niemeyer KE, Sung C-J. An investigation of GPU-based stiff chemical kinetics integration methods. Combust Flame 2017;179:312–24.

[10] Kim K, Díaz-Ibarra OH, Najm HN, Zádor J, Safta C. TChem: A performance portable parallel software toolkit for complex kinetic mechanisms. Comput Phys Comm 2023;285:108628.

[11] Bielawski R, Barwey S, Prakash S, Raman V. Highly-scalable GPU-accelerated compressible reacting flow solver for modeling high-speed flows. Comput & Fluids 2023;105972.

[12] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3(6):422–40.

[13] Zhou L, Song Y, Ji W, Wei H. Machine learning for combustion. Energy AI 2022;7:100128.

[14] Turányi T, Tomlin AS. Analysis of kinetic reaction mechanisms, vol. 20. Springer; 2014.

[15] Galassi RM, Ciottoli PP, Sarathy SM, Im HG, Paolucci S, Valorani M. Automated chemical kinetic mechanism simplification with minimal user expertise. Combust Flame 2018;197:439–48.

[16] Lu T, Law CK. A directed relation graph method for mechanism reduction. Proc Combust Inst 2005;30(1):1333–41.

[17] D'Alessio G, Parente A, Stagni A, Cuoci A. Adaptive chemistry via pre-partitioning of composition space and mechanism reduction. Combust Flame 2020;211:68–82.

[18] Newale A, Pope S, Pepiot P. Computationally-efficient and accurate particle PDF simulations of turbulent combustion using coupled pre-partitioned adaptive chemistry and tabulation. Proc Combust Inst 2021;38(2):2721–9.

[19] Liang L, Stevens JG, Farrell JT. A dynamic adaptive chemistry scheme for reactive flow computations. Proc Combust Inst 2009;32(1):527–34.

[20] Pope SB. Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. Combust Theory Model 1997;1:1:41–63.

[21] Tonse SR, Moriarty NW, Brown NJ, Frenklach M. PRISM: Piecewise reusable implementation of solution mapping. An economical strategy for chemical kinetics. Isr J Chem 1999;39(1):97–106.

[22] Christo FC, Masri AR, Nebot EM, Pope SB. An integrated PDF/neural network approach for simulating turbulent reacting systems. In: Symposium (International) on combustion, vol. 26, no. 1. Elsevier; 1996, p. 43–8.

[23] Sen BA, Hawkes ER, Menon S. Large eddy simulation of extinction and reignition with artificial neural networks based chemical kinetics. Combust Flame 2010;157(3):566–78.

[24] Franke LL, Chatzopoulos AK, 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;185:245–60.

[25] Coussement A, Gicquel O, Parente A. MG-local-PCA method for reduced order combustion modeling. Proc Combust Inst 2013;34(1):1117–23.

[26] Lei C, Deng J, Cao K, Ma L, Xiao Y, Ren L. A random forest approach for predicting coal spontaneous combustion. Fuel 2018;223:63–73.

[27] Barwey S, Prakash S, Hassanaly M, Raman V. Data-driven classification and modeling of combustion regimes in detonation waves. Flow Turbul Combust 2021;106:1065–89.

[28] Sutherland JC, Parente A. Combustion modeling using principal component analysis. Proc Combust Inst 2009;32(1):1563–70.

[29] Malik MR, Khamedov R, Pérez FEH, Coussement A, Parente A, Im HG. Dimensionality reduction and unsupervised classification for high-fidelity reacting flow simulations. Proc Combust Inst 2022.

[30] Dikeman HE, Zhang H, Yang S. Stiffness-reduced neural ODE models for data-driven reduced-order modeling of combustion chemical kinetics. In: AIAA SCITECH 2022 forum. 2022, p. 0226.

[31] Grassi T, Nauman F, Ramsey J, Bovino S, Picogna G, Ercolano B. Reducing the complexity of chemical networks via interpretable autoencoders. Astron Astrophys 2022;668:A139.

[32] Goussis DA. Quasi steady state and partial equilibrium approximations: their relation and their validity. Combust Theory Model 2012;16(5):869–926.

[33] Goussis D, Lam S. A study of homogeneous methanol oxidation kinetics using CSP. In: Symposium (International) on combustion, vol. 24, no. 1. Elsevier; 1992, p. 113–20.

[34] Goussis DA, Im HG, Najm HN, Paolucci S, Valorani M. The origin of CEMA and its relation to CSP. Combust Flame 2021;227:396–401.

[35] Maas U, Pope SB. Implementation of simplified chemical kinetics based on intrinsic low-dimensional manifolds. In: Symposium (International) on combustion, vol. 24, no. 1. Elsevier; 1992, p. 103–12.

[36] Valorani M, Goussis DA. Explicit time-scale splitting algorithm for stiff problems: auto-ignition of gaseous mixtures behind a steady shock. J Comput Phys 2001;169(1):44–79.

[37] Valorani M, Goussis DA, Creta F, Najm HN. Higher order corrections in the approximation of low-dimensional manifolds and the construction of simplified problems with the CSP method. J Comput Phys 2005;209(2):754–86.

[38] Valorani M, Paolucci S. The G-scheme: A framework for multi-scale adaptive model reduction. J Comput Phys 2009;228(13):4665–701.

[39] Galassi RM, Ciottoli PP, Valorani M, Im HG. An adaptive time-integration scheme for stiff chemistry based on computational singular perturbation and artificial neural networks. J Comput Phys 2022;451:110875.

[40] Anantharaman R, Ma Y, Gowda S, Laughman C, Shah V, Edelman A, et al. Accelerating simulation of stiff nonlinear systems using continuous-time echo state networks. 2020, arXiv preprint arXiv:2010.04004.

[41] Ji W, Qiu W, Shi Z, Pan S, Deng S. Stiff-PINN: Physics-informed neural network for stiff chemical kinetics. J Phys Chem A 2021;125(36):8098–106.

[42] Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. Adv Neural Inf Process Syst 2018;31.

[43] Owoyele O, Pal P. ChemNODE: A neural ordinary differential equations framework for efficient chemical kinetic solvers. Energy AI 2022;7:100118.

[44] Kim S, Ji W, Deng S, Ma Y, Rackauckas C. Stiff neural ordinary differential equations. Chaos 2021;31(9):093122.

[45] Mueller M, Kim T, Yetter R, Dryer F. Flow reactor studies and kinetic modeling of the H2/O2 reaction. Int J Chem Kinetics 1999;31(2):113–25.

[46] Smith G, Tao Y, Wang H. Foundational fuel chemistry model version 1.0 (FFCM-1). 2016, https://web.stanford.edu/group/haiwanglab/FFCM1/pages/FFCM1.html.

[47] Goodwin DG, Moffat HK, Schoegl I, Speth RL, Weber BW. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes. 2022, http://dx.doi.org/10.5281/zenodo.6387882, https://www.cantera.org, Version 2.6.0.

[48] Kuehn C. Multiple time scale dynamics. Applied mathematical sciences, Springer International Publishing; 2015.

[49] Lukassen AA, Kiehl M. Operator splitting for chemical reaction systems with fast chemistry. J Comput Appl Math 2018;344:495–511.

[50] Pérez FEH, Mukhadiyev N, Xu X, Sow A, Lee BJ, Sankaran R, et al. Direct numerical simulations of reacting flows with detailed chemistry using many-core/GPU acceleration. Comput & Fluids 2018;173:73–9.

[51] Desai S, Kim YJ, Song W, Luong MB, Pérez FEH, Sankaran R, et al. Direct numerical simulations of turbulent reacting flows with shock waves and stiff chemistry using many-core/GPU acceleration. Comput & Fluids 2021;215:104787.

[52] Niemeyer KE, Curtis NJ, Sung C-J. pyJac: Analytical Jacobian generator for chemical kinetics. Comput Phys Comm 2017;215:188–203.

[53] Valorani M, Creta F, Ciottoli PP, Malpica Galassi R, Goussis DA, Najm HN, et al. In: Pitsch H, Attili A, editors. Data analysis for direct numerical simulations of turbulent combustion: From equation-based analysis to machine learning. Cham: Springer International Publishing; 2020, p. 43–64.

[54] Valorani M, Ciottoli PP, Malpica Galassi R, Paolucci S, Grenga T, Martelli E. Enhancements of the G-scheme framework. Flow Turbul Combust 2018;101:1023–33.

[55] Parente A, Sutherland JC, Tognotti L, Smith PJ. Identification of low-dimensional manifolds in turbulent flames. Proc Combust Inst 2009;32(1):1579–86.

[56] Yu S, Principe JC. Understanding autoencoders with information theoretic concepts. Neural Netw 2019;117:104–23.

[57] Lee K, Parish EJ. Parameterized neural ordinary differential equations: Applications to computational physics problems. Proc R Soc Lond Ser A Math Phys Eng Sci 2021;477(2253):20210162.

[58] Plaut E. From principal subspaces to principal components with linear autoencoders. 2018, arXiv preprint arXiv:1804.10253.

[59] Rubanova Y, Chen RT, Duvenaud DK. Latent ordinary differential equations for irregularly-sampled time series. Adv Neural Inf Process Syst 2019;32.

[60] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. Pytorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 2019;32.

[61] Chen RTQ. Torchdiffeq. 2018, URL https://github.com/rtqichen/torchdiffeq.

[62] Despotovic M, Nedic V, Despotovic D, Cvetanovic S. Evaluation of empirical models for predicting monthly mean horizontal diffuse solar radiation. Renew Sustain Energy Rev 2016;56:246–60.

[63] Xu R, Wang H. Personal communication. 2019.

[64] Dormand JR, Prince PJ. A family of embedded runge-kutta formulae. J Comput Appl Math 1980;6(1):19–26.