Applications in Energy and Combustion Science 25 (2026) 100430

Contents lists available at ScienceDirect

Applications in Energy and Combustion Science

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

A latent variable framework for handling detailed chemistry in reacting flows☆

Eva Muñoz [a,b,*], Mohammad Rafi Malik [c], Alberto Cuoci [d], Hong G. Im [c], Alessandro Parente [a,b,e]

[a] ATM Laboratory, Université Libre de Bruxelles, Brussels, 1000, Belgium
[b] Brussels Institute for Thermal-fluid systems and clean Energy (BRITE), Belgium
[c] Clean Energy Research Platform (CERP), Physical Science and Engineering (PSE) Division, King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Saudi Arabia
[d] Politecnico di Milano, Piazza Leonardo da Vinci 32, Milan, 20131, Italy
[e] WEL Research Institute, Avenue Pasteur 6, Wavre, 1300, Belgium

ARTICLE INFO

Keywords:
Reduced-order models
Combustion
Detailed kinetics
ODE solver
Adaptive integration

ABSTRACT

Combustion involves multiple scales and intricate physical processes that make its simulation extremely demanding in terms of computational resources. To overcome this challenge, reduced-order models are essential for improving our understanding of the process and speeding up simulations. This work introduces a latent variable (LV) transport framework and demonstrates its compression capabilities through 0D batch reactor simulations. To improve solver efficiency and numerical stability, we introduce *projected tolerances*, which scale the ODE solver tolerances in the latent space, significantly improving numerical stability and computational efficiency compared to traditional *fixed tolerance* approaches.

The methodology is validated on 0D simulations using methane, propane, and n-heptane combustion with detailed and lumped kinetic mechanisms. Results show that the LV solver achieves substantial computational cost reductions (up to 50%) while preserving solution accuracy. We further analyse the influence of kinetic mechanisms, demonstrating that detailed mechanisms benefit the most from latent space compression due to their higher dimensionality.

The proposed LV framework offers a robust and efficient alternative to conventional solvers for detailed kinetics and establishes a foundation for its extension to multidimensional reactive flow simulations.

1. Introduction

The numerical simulation of reacting flows is highly demanding due to the complex, multiscale nature of combustion processes. Reduced-order models (ROMs) play a crucial role in addressing this challenge by enhancing computational efficiency while providing physical insights into the process. In this context, we focus on linear techniques to recast the original state variables into a new set of variables, called latent variables (LVs).

Principal Component Analysis (PCA) [1] is a common linear ROM used in combustion systems, as it maximizes the variance retained in the reduced set, whose basis is orthogonal. Compared to other reduced-order models, PCA offers a balance between low training cost, high accuracy, and interpretability [2]. Sutherland and Parente [3] proposed a framework, called PC-transport, based on deriving and solving a set of transport equations for the principal components (PCs), relying on the orthogonality and linearity of PCA. These PCs are then transported instead of the original variables and the entire state-space can be reconstructed from the principal component fields.

Previous studies have successfully demonstrated the feasibility of the PC-transport framework, in which both transport and chemistry steps can be formulated within the reduced space. However, the main challenge has consistently arisen from the highly nonlinear chemical source terms [4] when projected into a linear space. Thus, reconstructing the source terms from the reduced state variables often requires retaining a large number of PCs, limiting the achievable speed-up. These limitations are further exacerbated by the use of fixed solver tolerances, which lead to instabilities or excessive computational cost when dealing with detailed kinetic mechanisms. As a result, previous works with explicit source term computation were limited to small mechanisms and showed little reduction [5].

☆ This article is part of a Special issue entitled: 'AI for Combustion' published in Applications in Energy and Combustion Science.

* Corresponding author at: ATM Laboratory, Université Libre de Bruxelles, Brussels, 1000, Belgium.
E-mail addresses: eva.munoz.salamanca@ulb.be (E. Muñoz), Alessandro.Parente@ulb.be (A. Parente).

https://doi.org/10.1016/j.jaecs.2025.100430
Received 7 August 2025; Received in revised form 25 October 2025; Accepted 22 November 2025
Available online 27 November 2025

To address these challenges, prior studies have explored non linear regression of the source term, including multivariate adaptive spline regression (MARS) [6,7], Artificial Neural Networks (ANNs) [8,9] and Gaussian Progress Regression (GPR) [5,10].

More recently, deep-learning frameworks have also been employed to accelerate chemical kinetics computations by modelling species source terms. Owoyele [11] proposed a neural network in which the chemical source terms are directly integrated during training, with network weights updated through derivative-based optimisation, achieving a 2.3x speed-up in 0D reactor simulations. Similarly, Zhang [12] replaced the conventional ODE solver with a deep neural network trained on multi-scale sampled data, demonstrating 2–3x acceleration in 1D and 3D flame simulations.

Alternatively, model reduction techniques such as computational singular perturbation (CSP) [13] have been developed to simplify the system while preserving its essential dynamics. CSP-based solvers offer the advantage of reducing system stiffness by identifying fast modes and projecting them onto the slow subspace. However, their implementation typically requires an eigen-decomposition at every time step, resulting in high computational cost, or pre-tabulation strategies to mitigate this expense [14]. While these approaches can accelerate computations, they involve significant training cost and complexity.

In contrast, the present study proposes a self-contained strategy that enables the direct computation of source terms while improving numerical stability at minimal training cost. We extend the PC-transport approach into a generalized linear LV-model and introduce a novel solver formulation based on tolerance adaptation in the latent space. This advancement eliminates the need for source-term regression and allows stable, efficient simulations with large and detailed kinetic mechanisms.

The remainder of this paper is structured as follows. Section 2 details the methodology, including the LV framework and solver optimisation. Section 3 introduces the test cases and training datasets. Section 4 presents the results, including the *a priori* assessment, solver performance across score scales, fuels, and kinetic mechanisms, as well as robustness and stiffness analyses and a code profiling. Finally, the conclusions are summarized in Section 5.

2. Methodology

There are different ways to construct a linear projection for deriving the transport equation in a reduced-dimensional space. This section describes the procedure followed in this work, starting with the latent space parametrisation, followed by the latent variable model (LV-model), solver optimisation, and the metrics used to evaluate performance.

2.1. Latent space parametrisation

Different linear transformations produce latent spaces with distinct characteristics from a given dataset $\boldsymbol{\Phi} \in \mathrm{R}^{n\times p}$, where n and p are the number of samples and variables, respectively. In this application, the dataset consists of the species mass fractions sampled over time from a set of batch reactors, where the samples correspond to the collected observations (n), and the variables correspond to the species, with $p = n_s$.

Identifying a latent space parametrisation requires data collection, $\boldsymbol{\Phi}_*$, from which the latent features are extracted. The dataset is preprocessed via centring and scaling as, $\widetilde{\boldsymbol{\Phi}} = (\boldsymbol{\Phi}-\mathbf{C})\mathbf{D}^{-1}$, where $\mathbf{C} = \mathbf{1}_n\{c_j\}^T \in \mathrm{R}^{n\times p}$ and $\mathbf{D} = \mathrm{diag}(d_j)$ are the centring and scaling matrices, containing the corresponding parameters for each $j = 1,\dots,p$ variable. This work applies mean centring, $c_j = \overline{\phi}_{*_j}$, and Pareto scaling, $d_j = \sqrt{s_j}$, where s_j is the standard deviation of each value, as it was found to perform well in PC-transport [5].

Once the dataset is preprocessed, the reduced-order model (ROM) is built. Its selection impacts the latent space by modifying the basis.

Table 1
Basis matrix definitions for the three PC-scores.

Method	Direct basis matrix	Inverse basis matrix
U-scores	$\mathbf{A} = \mathbf{U}_q$	$\mathbf{A}_{inv} = \mathbf{U}_q^T$
V-scores	$\mathbf{A} = \mathbf{U}_q\mathbf{L}_q^{1/2}$	$\mathbf{A}_{inv} = \mathbf{L}_q^{-1/2}\mathbf{U}_q^T$
W-scores	$\mathbf{A} = \mathbf{U}_q\mathbf{L}_q^{-1/2}$	$\mathbf{A}_{inv} = \mathbf{L}_q^{1/2}\mathbf{U}_q^T$

The direct and inverse transformation matrices, $\mathbf{A}$ and $\mathbf{A}_{inv}$, project the dataset from the preprocessed high-dimensional space, $\widetilde{\boldsymbol{\Phi}}$, to the latent space, $\mathbf{Z}$, and vice versa, i.e. $\mathbf{Z} = \widetilde{\boldsymbol{\Phi}}\mathbf{A}$ and $\widetilde{\boldsymbol{\Phi}} = \mathbf{Z}\mathbf{A}_{inv}$.

PCA identifies the latent representation of the state-space that maximizes the retained variance in the latent space. It relies on the eigenvalue decomposition of the covariance matrix,

$$\mathbf{S} = \frac{1}{n-1}\widetilde{\boldsymbol{\Phi}}_*^T\widetilde{\boldsymbol{\Phi}}_* = \mathbf{U}\mathbf{L}\mathbf{U}^T, \tag{1}$$

where $\mathbf{U}$ and $\mathbf{L}$ are the eigenvectors and eigenvalues of $\mathbf{S}$, ordered in decreasing eigenvalue order. The reduced-order model is obtained by retaining the first $q < p$ eigenvalues, $\mathbf{L}_q$, and eigenvectors, $\mathbf{U}_q$.

Different approaches can be used to identify the latent variables, or scores, while preserving the explained variance. In this work, we explored the U, V, and W-scores. The basis matrix of U-scores is the eigenvector matrix, while the V- and W-score matrices weight the eigenvector matrix with the squared root of the eigenvalues and its inverse, respectively. The inverse transformation for PCA is computed as $\mathbf{A}_{inv} = \mathbf{A}^{-1}$ while accounting for the orthogonality of the eigenvectors, $\mathbf{U}$. The matrices definition is summarized in Table 1.

2.2. Latent variable model (LV-model)

LV-model generalizes the PC-transport approach proposed by Sutherland and Parente [3] to any latent space obtained through a linear transformation. This formulation reduces the number of scalar equations required to describe the evolution of n_s species, reducing them to $q < n_s$, and it is applicable for two- and three-dimensional reacting flows. In such applications, the operator-splitting method is commonly used to decouple the transport and chemical steps. The chemical step typically accounts for over 80% of the total computational cost, even for relatively small kinetic mechanisms [15]. Thus, improving the chemical step, modelled as a batch reactor, is key to optimising the approach.

The zero-dimensional batch reactor is solved in the species space as:

$$\frac{\partial \boldsymbol{\phi}_j}{\partial t} = S_{\boldsymbol{\phi}_j}, \tag{2}$$

where $S_{\boldsymbol{\phi}_j}$ denotes the source term for species j.

Given the basis matrix $\mathbf{A}$, which maps the state-space to the latent space, the chemistry step for the q latent variables, $\boldsymbol{z}_j(x,t)$, can be derived from Eq. (2) as:

$$\frac{\partial \boldsymbol{z}_j}{\partial t} = S_{\boldsymbol{z}_j}, \tag{3}$$

where:

$$\mathbf{S}_{\boldsymbol{z}} = \mathbf{S}_{\boldsymbol{\phi}}\mathbf{D}^{-1}\mathbf{A}, \tag{4}$$

represents the projection of $\mathbf{S}_{\boldsymbol{\phi}}$ onto the latent space as described in Ref. [16].

2.3. Optimisation of the LV-model solver

To efficiently integrate the latent-space equations, solver parameters must be adapted to the properties of the latent variables. In this work, the chemistry is computed in the latent space with direct computation of the source terms. At each step, the latent source term $\mathbf{S}_{\boldsymbol{z}}$ is derived

by projecting the state-space source term $\mathbf{S}_{\phi}$. To achieve this, the state-space $\boldsymbol{\Phi}$ is first reconstructed, after which the source terms $\mathbf{S}_{\phi}$ are computed and subsequently projected onto the latent space to obtain $\mathbf{S}_{\mathbf{z}}$ (Eq. (4)). Finally, the ODE solver computes the solution of Eq. (3).

The ODE solver is originally designed for species mass fractions (constrained to $[0, 1]$), whereas latent variables have no inherent physical bounds. A numerical safeguard is introduced by limiting the latent variable values to $\pm 10^{12}$.

With this approach, the solver runs successfully and yields accurate results, as demonstrated in the literature [17]. However, the solver is not tailored to the new variables, whose orders of magnitude differ not only from the species but also among the latent variables themselves.

The accuracy required by the ODE solver is controlled through various parameters. Among these, the variable tolerances directly depend on the specific variables being solved and must therefore be adjusted in the latent formulation.

The tolerance vector for the species, $\boldsymbol{\epsilon}(\boldsymbol{\Phi})$, is defined as:

$$\boldsymbol{\epsilon}(\boldsymbol{\Phi}) = \boldsymbol{\epsilon}_a^{\boldsymbol{\Phi}} + \boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}} \boldsymbol{\Phi}, \tag{5}$$

where $\boldsymbol{\epsilon}_a^{\boldsymbol{\Phi}} > 0$ and $\boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}} > 0$ are the absolute and relative tolerance vectors for the species, specified as input parameters.

The generalization of the standard approach to the latent solver is referred to as the *fixed tolerances* method. This approach applies the same tolerances defined for the species in the latent space. The corresponding equation follows the form of Eq. (5), replacing the species with the absolute value of the latent variables:

$$\boldsymbol{\epsilon}(\mathbf{Z}) = \boldsymbol{\epsilon}_a^{\boldsymbol{\Phi}} + \boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}} |\mathbf{Z}|, \tag{6}$$

where the absolute value ensures that the tolerance remains positive, as $\mathbf{Z}$ is not necessarily positive.

Since PCA transforms species mass fractions (ranging within $[0, 1]$) into latent variables with arbitrary magnitudes, *fixed tolerances* can induce numerical stiffness or instability. To prevent this, we introduce the *projected tolerances*, which adapt the solver accuracy to latent variable scales:

$$\boldsymbol{\epsilon}(\mathbf{Z}) = \boldsymbol{\epsilon}_a^{\mathbf{Z}} + \boldsymbol{\epsilon}_r^{\mathbf{Z}} \mathbf{Z}, \tag{7}$$

where $\boldsymbol{\epsilon}_a^{\mathbf{Z}}$ and $\boldsymbol{\epsilon}_r^{\mathbf{Z}}$ are vectors containing the absolute and relative tolerances for each latent variable.

The *projected tolerances* are defined as the projection of the species tolerance vector onto the latent space:

$$\boldsymbol{\epsilon}(\mathbf{Z}) = \boldsymbol{\epsilon}(\boldsymbol{\Phi}) \mathbf{D}^{-1} \mathbf{A}. \tag{8}$$

This formulation allows the absolute and relative tolerances in the latent space to be derived from the corresponding tolerances in the state-space. Substituting Eqs. (5) and (7) into Eq. (8), and expressing $\mathbf{Z}$ in terms of $\boldsymbol{\Phi}$ yields:

$$\boldsymbol{\epsilon}_a^{\mathbf{Z}} + \boldsymbol{\epsilon}_r^{\mathbf{Z}} (\boldsymbol{\Phi} - \mathbf{C}) \mathbf{D}^{-1} \mathbf{A} = (\boldsymbol{\epsilon}_a^{\boldsymbol{\Phi}} + \boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}} \boldsymbol{\Phi}) \mathbf{D}^{-1} \mathbf{A}, \tag{9}$$

from which the projected absolute and relative tolerances in the latent space can be extracted:

$$\boldsymbol{\epsilon}_a^{\mathbf{Z}} = (\boldsymbol{\epsilon}_a^{\boldsymbol{\Phi}} + \boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}} \mathbf{C}) \mathbf{D}^{-1} \mathbf{A} = \widetilde{\boldsymbol{\epsilon}} \mathbf{A}, \qquad \text{where } \widetilde{\boldsymbol{\epsilon}} = (\boldsymbol{\epsilon}_a^{\boldsymbol{\Phi}} + \boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}} \mathbf{C}) \mathbf{D}^{-1} \tag{10}$$

$$\boldsymbol{\epsilon}_r^{\mathbf{Z}} = \boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}}. \tag{11}$$

However, directly projecting the weighted tolerances, $\widetilde{\boldsymbol{\epsilon}}$, using matrix multiplication can lead to cancellations due to opposing signs in the basis $\mathbf{A}$. To address that, we use Gaussian error propagation of the tolerances, yielding:

$$\boldsymbol{\epsilon}(\mathbf{Z}) = \boldsymbol{\epsilon}_a^{\mathbf{Z}} + \boldsymbol{\epsilon}_r^{\mathbf{Z}} |\mathbf{Z}| \tag{12}$$

$$\epsilon_{a_j}^{\mathbf{Z}} = \sqrt{\sum_{i=1}^{n_s} (\widetilde{\epsilon}_i^2 a_{ij}^2)}, \qquad \boldsymbol{\epsilon}_r^{\mathbf{Z}} = \boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}}. \tag{13}$$

It is important to note that the proposed formulation does not introduce any additional hyper-parameters, as it is fully derived from the existing tolerance parameters ($\boldsymbol{\epsilon}_a^{\boldsymbol{\Phi}}$ and $\boldsymbol{\epsilon}_r^{\boldsymbol{\Phi}}$). Moreover, its implementation involves minimal computational overhead since the projected tolerances are computed only once at the initial time step and remain constant throughout the simulation.

2.4. Assessment metrics

The performance of the LV-model is evaluated based on the computational time and the number of steps required to solve the reactors, the reduction in number of variables, and the accuracy of the solution, quantified by the relative root mean square error. Furthermore, system stiffness is evaluated by analysing the chemical timescales of the system.

The relative root mean square error (RRMSE) quantifies the deviation between predicted (x_{pred}) and observed values (x_{obs}). It is defined as:

$$\text{RRMSE} = \frac{\|x_{obs} - x_{pred}\|_F}{\|x_{obs}\|_F} \tag{14}$$

where $\|\cdot\|_F$ indicates the Frobenius norm. In this study, the reconstructed species and source terms from the LV-model ($\boldsymbol{\Phi}^{LV}$ and $\mathbf{S}_{\boldsymbol{\phi}}^{LV}$) are compared with those from the original species model ($\boldsymbol{\Phi}^{SP}$ and $\mathbf{S}_{\boldsymbol{\phi}}^{SP}$). For the error computation, species are centred and scaled using the preprocessing parameters as follows:

$$x_{obs} = (\boldsymbol{\Phi}^{SP} - \mathbf{C}) \mathbf{D}^{-1}; \qquad x_{pred} = (\boldsymbol{\Phi}^{LV} - \mathbf{C}) \mathbf{D}^{-1}, \tag{15}$$

and the source terms are standardized.

To consistently evaluate and compare solver performance, several key compression levels are defined: q_{min}^{CPU} denotes the minimum number of latent variables for which the solver produces a stable solution; q_{min}^{acc} is the smallest number of variables that ensures solution accuracy; and q_{opt} corresponds to the configuration that minimizes computational time while preserving accuracy. Additionally, in extrapolation cases where accuracy cannot be guaranteed, q_{min}^{time} denotes the compression level that minimizes computational time regardless of accuracy. These compression levels are identified for each fuel and operating condition.

An ODE system is considered stiff when the system timescales are in a very broad range. To compute the timescales, the Jacobian of the system is first introduced, computed as $\boldsymbol{J}^{\phi} = \partial^{\phi} \mathbf{S}^{\phi}$ for the species space and $\boldsymbol{J}^{z} = \partial^{z} \mathbf{S}^{z} = \mathbf{A}_{inv} \mathbf{D}^{-1} \boldsymbol{J}^{\phi} \mathbf{D} \mathbf{A}$ for the latent space [18]. Then, an eigenvalue decomposition is made for the Jacobian matrices:

$$\boldsymbol{J}^{\boldsymbol{\phi}} = \boldsymbol{R}^{\boldsymbol{\phi}} \boldsymbol{\Lambda}^{\boldsymbol{\phi}} \mathbf{L}^{\boldsymbol{\phi}}; \qquad \mathbf{L}^{\boldsymbol{\phi}} = \boldsymbol{R}^{\boldsymbol{\phi}^{-1}}; \tag{16}$$

$$\boldsymbol{J}^{\mathbf{z}} = \boldsymbol{R}^{\mathbf{z}} \boldsymbol{\Lambda}^{\mathbf{z}} \mathbf{L}^{\mathbf{z}}; \qquad \mathbf{L}^{\mathbf{z}} = \boldsymbol{R}^{\mathbf{z}^{-1}}, \tag{17}$$

where the matrices $\mathbf{L}^{\boldsymbol{\phi}}$ and $\mathbf{L}^{\mathbf{z}}$ contain the left eigenvectors, $\boldsymbol{R}^{\boldsymbol{\phi}}$ and $\boldsymbol{R}^{z}$ the right eigenvectors and $\boldsymbol{\Lambda}^{\boldsymbol{\phi}}$ and $\boldsymbol{\Lambda}^{\mathbf{z}}$ are the diagonal matrices containing the eigenvalues. The timescales of the system are the inverse of the absolute value of the eigenvalues:

$$\boldsymbol{\tau}_i^{\phi} = 1/|\lambda_i^{\phi}|; \qquad \text{for } i = 1, \ldots, n_s; \tag{18}$$

$$\boldsymbol{\tau}_j^{z} = 1/|\lambda_j^{z}|; \qquad \text{for } j = 1, \ldots, q. \tag{19}$$

3. Test case and training dataset

The latent model was evaluated and compared against a 0D constant-pressure batch reactor model that directly solves the species equations. Three fuel/air mixtures (CH_4, C_3H_8, and n-C_7H_{16}) were considered over a range of equivalence ratios, φ, and initial mixture temperatures, T_0. Both lumped and more detailed kinetic mechanisms are used, excluding argon and helium. For methane and propane, the lumped mechanism is the C1C3 module from the CRECK group [19], comprising 111 species and 1998 reactions, while the detailed mechanism is the corresponding module from Aramco V1.3 [20], containing 251 species and 1537 reactions. For n-heptane, the lumped mechanism from the CRECK group [21] includes 252 species and 7567 reactions,

whereas the non-lumped LLNL 3.1 mechanism [22] consists of 654 species and 2827 reactions.

The training data used to construct the projection matrices was generated using OpenSMOKE++ [23] and its native ODE integrator for the same reactor configuration. A series of offline simulations were conducted, varying the initial reactor temperature, T_0, between 1400 K and 1500 K to produce different ignition trajectories, and by adjusting the equivalence ratio, φ, between 0.7 and 1.1, resulting in 99 unique combinations. For each fuel and mechanism, this yielded approximately 80,000 to 90,000 observations per state-space variable.

The projection bases were computed offline from the training dataset. After preprocessing, PCA was applied to extract the U, V, and W-scores. To validate and test the ROM, simulations were performed using the same ODE integrator (OpenSMOKE++ native ODE integrator) within the training range, including 56 test points, which correspond to 8 conditions on T_0 and 7 on φ. To assess interpolation capabilities, 45% of the test points were not seen by the ROM during training.

Furthermore, to assess ODE integrator independence, the same test cases were also computed using CVODE. To assess extrapolation capabilities, additional simulations were performed for $\varphi \in [0.3, 1.5]$ and $T_0 \in [1200, 1800]$ K.

4. Results

The latent variable model and its optimisation for a 0D batch reactor, based on *projected tolerances*, were validated using the test cases presented in Section 3 with detailed mechanisms. For all simulations, species tolerances were set uniformly for all the species as: $\epsilon_a^{\boldsymbol{\Phi}} = 10^{-10}$ and $\epsilon_r^{\boldsymbol{\Phi}} = 10^{-6}$. The *fixed tolerances* approach directly applies these values, as defined in Eq. (6), while the *projected tolerances* were computed according to Eqs. (12)–(13).

First, the solver was evaluated *a priori* across all fuels to quantify the projection error. Next, the adaptability of the optimised ODE solver (using *projected tolerances*) to different score scalings was evaluated by comparing its performance using U, V, and W-scores for methane combustion. Its performance was compared with that of the non-optimised solver (using *fixed tolerances*) across all fuels. The robustness of the approach was also assessed in terms of independence from the ODE integrator and for its capability to extrapolate to conditions outside of the training range. Finally, the influence of the kinetic mechanisms on solver performance was analysed.

4.1. A priori analysis of the projection errors

The information loss resulting from projecting species and source terms onto the latent space was evaluated *a priori*. The test cases included methane, propane and n-heptane fuels using detailed mechanisms at an equivalence ratio of $\varphi = 0.9$ and an initial temperature of $T_0 = 1440$ K. U-scores were employed as the latent variables. The choice of score does not affect the error, as will be shown in Section 4.2. The error was quantified as the relative root mean square error, RRMSE (Eq. (14)).

Fig. 1 compares the RRMSE of the reconstructed species and source terms as a function of the number of retained variables (q) for all the fuels. When many variables are retained, both errors are very small but increase as q decreases. However, the increase in species error is monotonic, while the source term error fluctuates due to its nonlinear dependence on species concentrations. As a result, improved species reconstruction does not necessarily lead to better source term accuracy. Overall, the source term reconstruction error is several orders of magnitude larger than that of species, due to the Arrhenius form of the source term. This can lead to nonlinear error propagation when solving the LV-model equation (Eq. (3)), potentially resulting in solver instability and limited compression. This observation motivated the use of nonlinear regression in the literature [5–10]. Additionally, the source term error is strongly influenced by the kinetic mechanism, as it defines the source term manifold. Fuels using the same mechanism, such as methane and propane, exhibit similar trends, whereas changing the mechanism, as in the case with n-heptane, significantly alters the error trends.

Table 2
Order of magnitude of the three PC-scores and their absolute tolerances on the optimised solver. The test case is methane using Aramco V1.3 mechanism at $\varphi = 0.9$ and $T_0 = 1440$ K.

	$\lvert\mathbf{Z}\rvert$ range	$\lvert\epsilon_a^{\mathbf{Z}}\rvert$ range
V-scores	[1e−25, 3e−01]	[1e−12, 4e-06]
U-scores	[1e−17, 6e−01]	[1e−07, 4e+02]
W-scores	[1e−09, 2e+00]	[1e−06, 4e+10]

4.2. Adaptability of the optimised solver to the score scale

The adaptability of the optimised solver to different score scalings was assessed by evaluating its performance when using U, V, and W-scores. The test case was methane with the detailed mechanism, Aramco V1.3, at an equivalence ratio of $\varphi = 0.9$ and an initial temperature of $T_0 = 1440$ K, using OpenSMOKE++ ODE integrator. Performance was measured using three metrics: computational time, number of steps required to solve the reactor, and the error (RRMSE) when comparing the latent and species models.

The minimum and maximum values of the three scores and of their corresponding absolute tolerance for the optimised solver are detailed in Table 2. As shown, the *projected tolerances* adapt to the magnitude of the scores, and both can be ordered in increasing magnitude: V-scores, U-scores, and W-scores.

Fig. 2 compares the performance of the latent approach for the optimised and non-optimised solver, using *projected* and *fixed tolerances* respectively. The performance metrics are shown as a function of the latent dimension, q, for the three PC-scores.

The error, RRMSE, exhibits a similar trend for both solvers. When no compression is applied, the error is small and it remains approximately constant when compressing up to approximately $q \sim 100$. Beyond this point, the error increases exponentially until it reaches the minimum number of latent variables for which the solver produces a stable solution, q_{min}^{CPU}. Notably, q_{min}^{CPU} and solver accuracy when compression provides computational benefits are independent of the score type and tolerance used. This occurs because all three scores capture the same amount of variance for a given dimensionality reduction, and both tolerance settings are sufficiently tight to accurately represent the underlying physics. The minimum error is slightly higher when using the *projected tolerances* compared to *fixed tolerances* ($\text{RRMSE}_{q=n_s} = 5 \cdot 10^{-7}$ vs $8 \cdot 10^{-8}$ for U-scores), but this difference is negligible for the current application.

On the other hand, the computational time and steps required depend significantly on the solver parameters used. When *projected tolerances* are applied, performance remains consistent across score types. In contrast, with *fixed tolerances*, performance is highly dependent on the score transported. When using *fixed tolerances*, W-scores require significantly more computational time and steps, since they have the largest magnitude. This leads to excessively small tolerances for the score, resulting in numerical instabilities that cause the solver failure at $q = 220$ due to reaching the minimum step size. On the contrary, V-scores, having the smallest magnitude, allow for less strict tolerances, resulting in a faster but less accurate computation.

In conclusion, all three PCA score types (U, V, W) capture the same variance but differ in scale; the performance comparison highlights the need for adaptive tolerances to maintain solver stability across these scales. As a result, the need to rescale or select alternative scores is eliminated, and for simplicity, U-scores are used for the remainder of this work.

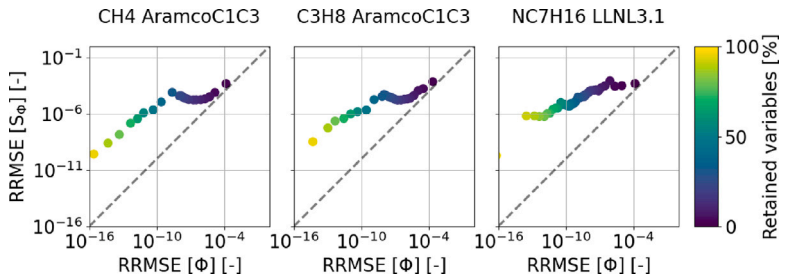


Fig. 1. Error associated to the latent projection, on species and source terms. The test cases are methane, propane and n-heptane using detailed mechanisms at $\varphi = 0.9$ and $T_0 = 1440$ K.

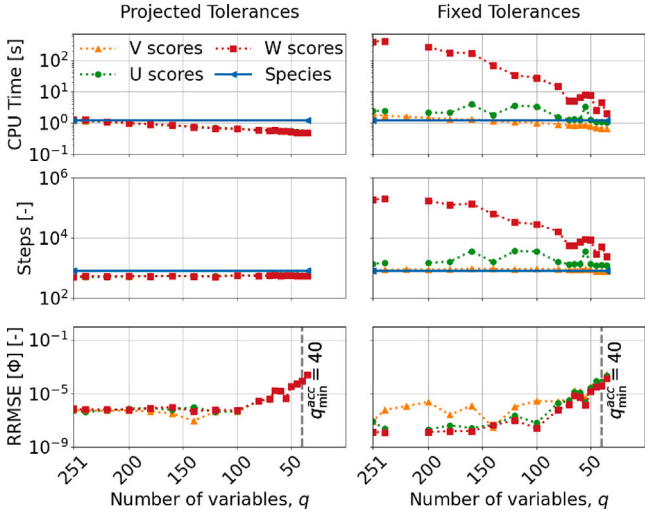


Fig. 2. Performance of the latent solver with projected and *fixed tolerances* using U, V, and W-scores. The test case is methane using Aramco V1.3 mechanism at $\varphi = 0.9$ and $T_0 = 1440$ K.

4.3. Performance of the solver for different fuels

The performance of the approach was then evaluated for all the fuels using detailed mechanisms, at $\varphi = 0.9$ and $T_0 = 1440$ K, using OpenSMOKE++ ODE integrator.

Fig. 3 shows the performance of both the optimised and non-optimised solvers and highlights the minimum number of retained variables required for accurate results, q_{min}^{acc}. To facilitate comparison across fuels, the computational time and steps are presented as percentage reductions relative to species-equation computation.

Across all fuels, the *projected tolerances* approach consistently outperforms the *fixed tolerances* approach, reducing computational time and solver steps. In contrast, applying *fixed tolerances* in the latent model often leads to inefficient computations, making it less advantageous than directly solving species equations. Furthermore, the error between both approaches is very similar, differing only in the low-compression region. In this region, the error is minimal (RRMSE $\sim [10^{-9}, 10^{-6}]$), making the difference negligible in absolute terms. However, for n-heptane, the *projected tolerances* result in higher errors, which do not impact the performance. The errors in the low-compression region are associated to the reconstruction of minor species with low concentration, which leads to high relative errors due to the weighting of each species by its scaling factor (Eqs. (14)–(15)). These results confirm that the *projected tolerances* approach outperforms the *fixed tolerances* approach across all the fuels, and *projected tolerances* will be used in the remainder of this work.

The performance of the latent model reflects the characteristics of both the solver and the reduced-order model used. When no reduction is applied ($q = n_s$), the number of equations and system complexity remain the same as in the species solver. However, the computational cost for the latent solver is higher due to two additional mapping steps required to compute the latent source term, $\mathbf{S}_z$. In particular, the species are first reconstructed from the latent variables, $\widetilde{\boldsymbol{\Phi}} = \mathbf{Z}\mathbf{A}_{inv}$, then the species source terms are computed, $\mathbf{S}_{\boldsymbol{\phi}}$, and finally, they are projected onto the latent space, $\mathbf{S}_z$ (Eq. (4)).

PCs are ordered by decreasing contribution: the first components capture most of the variance and represent the dominant physical behavior. In contrast, the last components account for only a small fraction of the variance, focusing on finer details that are not essential for approximating the species state. Fig. 3 illustrates this pattern clearly. When reducing the number of variables from $q = n_s$ the error remains very low and nearly constant until reaching a critical dimension q^*, which is approximately 100, 150 and 350 for the methane, propane and n-heptane cases respectively. In this region, the removed PCs have minimal impact on the reconstruction error while significantly reducing the computational cost by reducing the number of equations. For $q < q^*$, the reconstruction error increases exponentially with the reduction, until reaching the solver maximum compression limit, q_{min}^{CPU}, equal to 35, 60 and 310 for the methane, propane, and n-heptane cases, respectively. The existence of q_{min}^{CPU} is a consequence of the non-linear propagation of the error introduced by the mapping between species and source terms (Section 4.1). However, this effect is less abrupt for small fuel molecules, allowing for higher compression levels, though with some loss of accuracy, as shown for methane, with $q_{min}^{CPU} = 35$ and $q_{min}^{acc} = 40$. This demonstrates that for methane, the latent solver remains accurate even with up to 85% dimensionality reduction.

For all the fuels, the latent solver requires a lower number of steps compared to the species solver. However, reducing the dimensionality slightly increases the number of solver steps, as it must handle an approximate system of equations. Furthermore, these trends are not monotonic due to the nonlinear relationship between species and source terms. As explained in Section 4.1, adding a PC improves species reconstruction but it can worsen the mapping for the source term, leading to fluctuations in the performance trends.

The maximum CPU reductions while maintaining the accuracy are significant, reaching 56.6%, 45.4%, and 50.2%, for the methane, propane and n-heptane cases, respectively. These reductions were not been reported in previous PC-transport studies, as existing solvers were not optimised for detailed chemistry due to the limitations of *fixed tolerances*.

The accuracy of the approach is demonstrated in Fig. 4, where the profiles of the temperature and O_2, OH and fuel mass fractions from species and latent solvers are displayed for methane, propane and n-heptane fuels. The compression levels shown are the minimum q for which the solver provides a stable solution, q_{min}^{CPU}, and the minimum q for obtaining an accurate solution, q_{min}^{acc}. For propane and n-heptane, $q_{min}^{CPU} = q_{min}^{acc}$ and their profiles perfectly align with the species model profiles, demonstrating that the latent model is accurate for all q. However, in the methane case, discrepancies arise at the highest compression levels, particularly in capturing the ignition delay

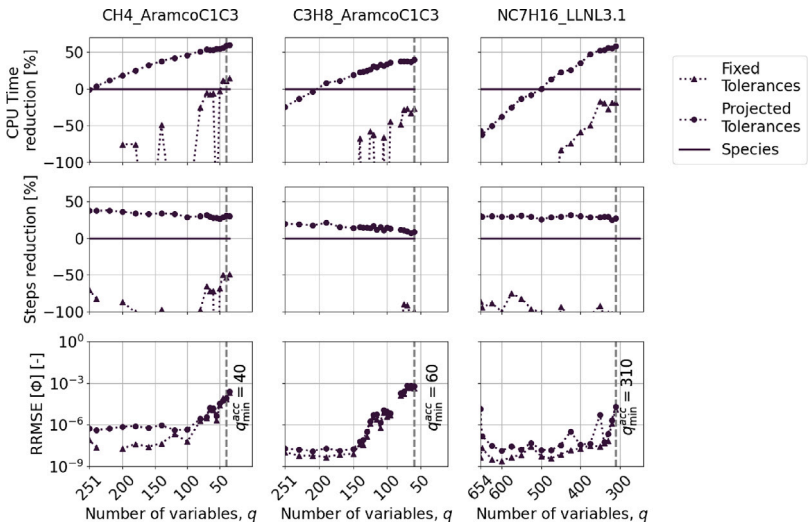


Fig. 3. Performance of the latent solver with *projected* and *fixed tolerances*. The test cases are methane, propane and n-heptane using detailed mechanisms at $\varphi = 0.9$ and $T_0 = 1440$ K.

time, in agreement with Ref. [4]. This suggests that while the latent solver maintains high accuracy for medium-to-large molecules such as propane and n-heptane, its reliability for smaller molecules like methane may be compromised under high dimensionality reduction.

4.4. Robustness assessment

The robustness of the proposed approach was evaluated with respect to two factors: its independence from the choice of ODE integrator and its extrapolation capability. The test case considered was methane with the detailed AramcoC1C3 mechanism.

4.4.1. Independence from ODE integrator

The robustness of the proposed approach with respect to the ODE integrator was evaluated by comparing computational gains using the native OpenSMOKE++ integrator and CVODE. The comparison was conducted across all test conditions (all φ and T_0, see Section 3 for details). For each test condition, the optimal compression level q_{opt} was selected, defined as the one that minimizes computational time while preserving accuracy.

Fig. 5 compares computational times for species and latent models across all test cases using both ODE integrators. Both integrators exhibit similar reductions, 56% for OpenSMOKE++ and 54% for CVODE, resulting in an average computational time of 0.54 s in both. This demonstrates that the latent approach performs consistently regardless of the ODE integrator employed.

OpenSMOKE++ uses a variable-step, variable-order Gear-type BDF algorithm implemented from scratch, tailored to combustion problems with very tight control over tolerances. CVODE, while also relying on BDF methods, incorporates different internal heuristics for step-size selection, Jacobian treatment, and error control. The consistency of results across these two different implementations reinforces the robustness of the proposed latent model solver.

4.4.2. Extrapolation performance

Extrapolation was assessed by testing the approach under conditions with equivalence ratios and initial temperatures beyond the training range, defined as $\varphi^{\text{train}} \in [0.7, 1.1]$ and $T_0^{\text{train}} \in [1400, 1500]$ K. The extrapolation conditions extended to $\varphi \in [0.3, 1.5]$ and $T_0 \in [1200, 1800]$ K.

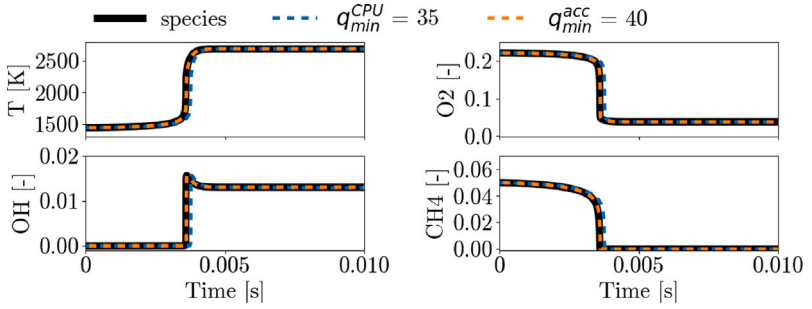
Fig. 6 presents the solver performance under these extrapolation conditions, measured by the computational time reduction and the species reconstruction error across compression levels from $q = 100$ to $q = 50$. A case within the training range ($T_0 = 1440$ K) is included for comparison. Across all test conditions and most compression levels, the latent solver consistently achieves computational savings similar to those observed in the training range, with no strong correlation to specific test conditions. In contrast, species reconstruction error exhibits identifiable trends. The error increases exponentially as q decreases from 100, with similar rates across both training and extrapolated conditions. Error sensitivity to the equivalence ratio is minimal, while sensitivity to initial temperature T_0 is significant. The lowest error occurs at $T_0 = 1440$ K (within the training range). For $T_0 > T_0^{\text{train}}$, errors remain low and comparable, maintaining accurate solutions. However, for $T_0 < T_0^{\text{train}}$, particularly at $T_0 = 1200$ K, errors increase significantly, compromising both accuracy and solver stability. For $T_0 < 1200$ K, the reduced accuracy leads to solver failure. This degradation is attributed to the influence of low-temperature chemistry, which is underrepresented in the training set.

Fig. 7 shows temperature profiles computed with the species and latent models under extrapolated conditions. The profiles correspond to either the optimal compression level, q_{opt}, if it exists, or the compression level that minimizes computational time, q^{time}_{min}, otherwise. When $T_0 > T_0^{\text{train}}$ (Fig. 7(a)), the latent solver accurately reproduces the temperature evolution. At lower temperatures (Fig. 7(b)), it performs well at $T_0 = 1300$ K but fails to capture ignition timing at $T_0 = 1200$ K. In contrast, variations in φ have minimal impact on the accuracy as shown also in Fig. 6. These results highlight the robustness of the latent solver when extrapolating accurately across a wide range of initial conditions beyond the training domain.

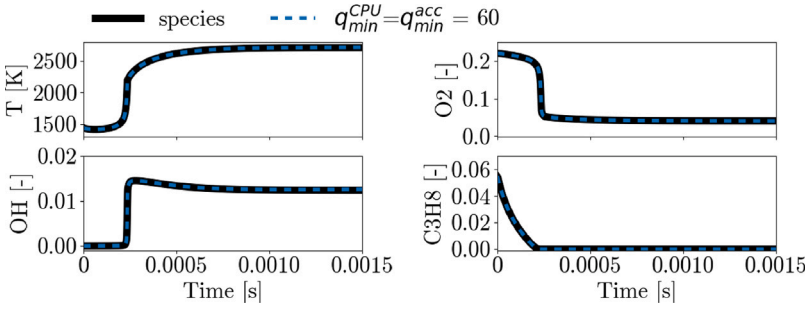
4.5. Stiffness analysis

The stiffness of the approach and its implications for the solver time step are analysed for both the full-order model (species space) and the reduced-order model (latent space). The test case considered is methane using the detailed AramcoC1C3 mechanism, integrated with the native OpenSMOKE++ solver.

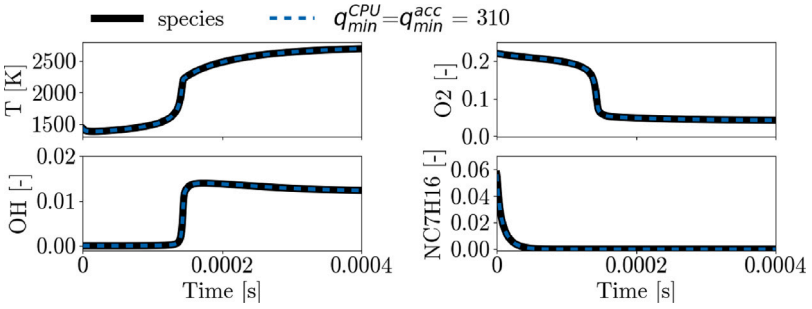
Fig. 8 shows the time steps used by the species and latent solvers at three compression levels within the range of accurate results. The time step adapts to the case, starting from a very small value and rapidly increasing until reaching approximately $\Delta t \approx 10^{-5}$. From this point, the cold state evolves while the time step gradually increases until the onset of the reacting phase. At this stage, the time step automatically decreases to better resolve the chemical reactions, followed by a gradual increase as the mixture approaches equilibrium. Although both the



(a) Methane



(b) Propane



(c) N-heptane

Fig. 4. Visualization of the temperature and O_2, OH and fuel mass fractions over the time of the species and latent models at q_{min}^{CPU} and q_{min}^{acc}. The test cases are methane, propane and n-heptane fuels with detailed mechanisms, at $\varphi = 0.9$ and $T_0 = 1440$ K.

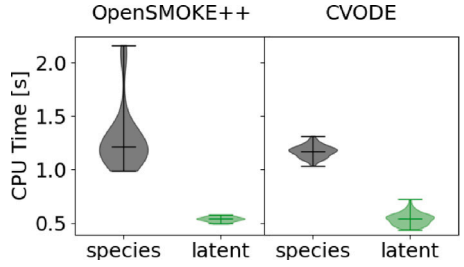


Fig. 5. Comparison of computational time for species and latent approaches using OpenSMOKE++ and CVODE integrators. The test case is methane using Aramco V1.3 mechanism across all test conditions.

species and latent models exhibit the same general behavior, the latent case allows for larger time steps, particularly before the reacting stage, thereby reducing the total number of steps required by the solver.

The chemical time scales τ of these simulations are presented in Fig. 9. When no compression is applied ($q = n_s$, Fig. 9(a)), the time scales of the species and latent models are equivalent, indicating that all time scales are preserved under the transformation. When an intermediate compression is applied (Fig. 9(b)), stiffness is reduced by removing the smallest and fastest time scales. This effect becomes more pronounced and reaches its maximum at the highest compression level without compromising accuracy ($q = q_{min}^{acc}$, Fig. 9(c)). These results demonstrate that a reduced model based on PCA decreases system stiffness, suggesting that the least contributing principal components are associated with the highest and smallest time scales.

However, since the main objective of the reduced-order model (ROM) is not stiffness reduction, this effect remains limited, and the increase in time step is only moderate. Further studies could explore combining the latent model with dedicated stiffness-reduction techniques.

4.6. Computational cost analysis and code profiling

This section analyses the computational origin of the savings achieved by the latent space solver using the built-in OpenSMOKE++ timers. The test case considered is methane with the detailed AramcoC1C3 mechanism under conditions $\varphi = 0.9$ and $T_0 = 1440$ K.

Table 3 summarizes the profiling results for the species and latent solvers at different compression levels. Reported quantities include the number of integration steps, total and per-step CPU time for ODE integration, and the relative contribution of space transformations and chemistry source term evaluation to the ODE integration time. As

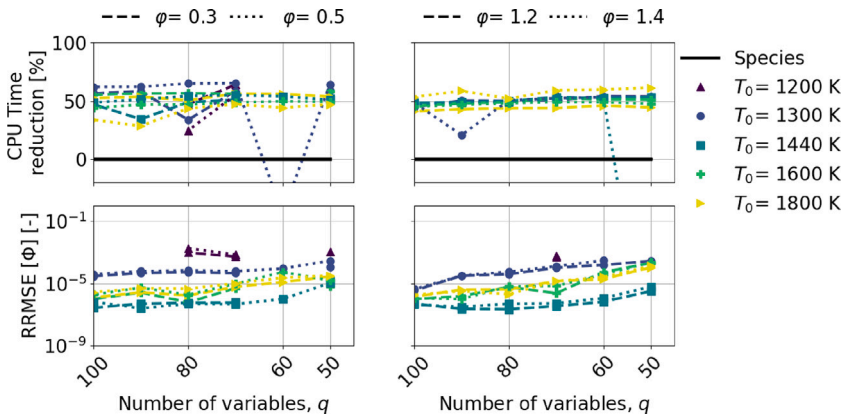
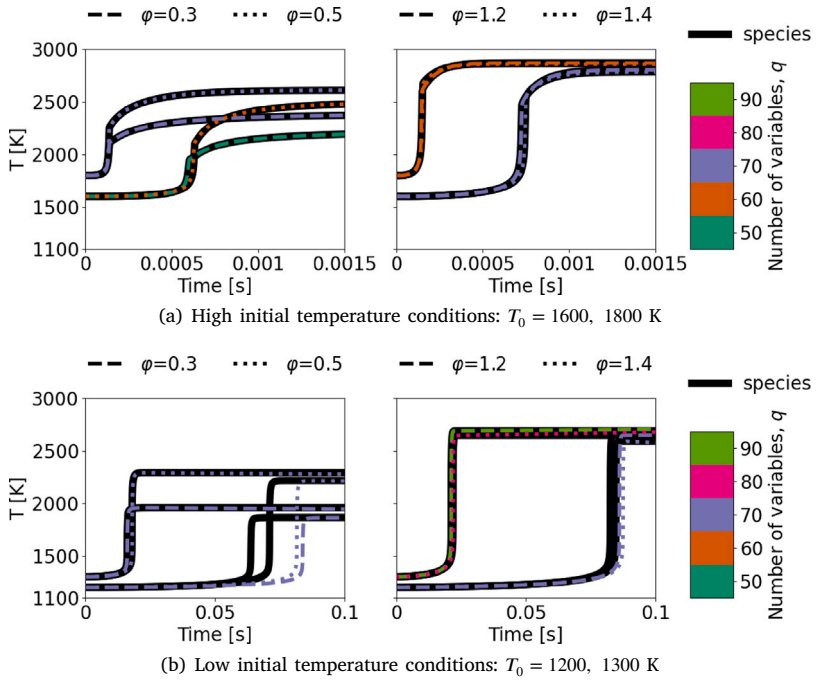


Fig. 6. Performance of the latent solver under extrapolation conditions φ and T_0, showing CPU time and error trends as a function of the retained number of variables (q). The test case is methane using Aramco V1.3 mechanism and initial conditions include $\varphi \in [0.3, 1.5]$ and $T_0 \in [1200, 1800]$ K.



(a) High initial temperature conditions: $T_0 = 1600,\ 1800$ K

(b) Low initial temperature conditions: $T_0 = 1200,\ 1300$ K

Fig. 7. Temperature profiles of the species and latent approaches under extrapolated conditions. Each case uses the optimal compression level, q_{opt}, when available, or q^{time}_{min}, otherwise. Test case is methane using Aramco V1.3 mechanism and initial conditions include $\varphi = 0.3,\ 0.5,\ 1.2,\ 1.5$ and $T_0 = 1200,\ 1300,\ 1600,\ 1800$ K.

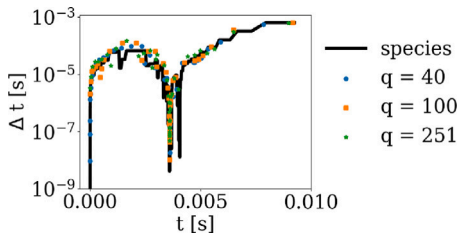


Fig. 8. Time steps Δt [s] used by the solver to compute the species and latent models at $q = 40, 100$ *and* 251. The test case is methane using Aramco V1.3 mechanism at $\varphi = 0.9$ and $T_0 = 1440$ K.

discussed in Section 4.3, integration in the latent space requires fewer steps due to improved numerical conditioning. In addition, the CPU time per step decreases with the number of retained variables q, reflecting the reduced dimensionality of the system. For the latent solver, the ODE integration time is composed mainly of two parts: (i) transformations between species and latent spaces, and (ii) evaluation of chemistry source terms. For high compression levels (e.g., $q = 251$), the cost of transformations becomes significant, up to 46% of the ODE integration time, but this contribution decreases for lower compression levels, indicating that the transformation overhead remains minor in practical configurations. Overall, the profiling confirms that the performance gain of the latent solver arises primarily from the reduced number of integration steps and the smaller per-step computational load, while projection operations contribute only modestly to the total cost.

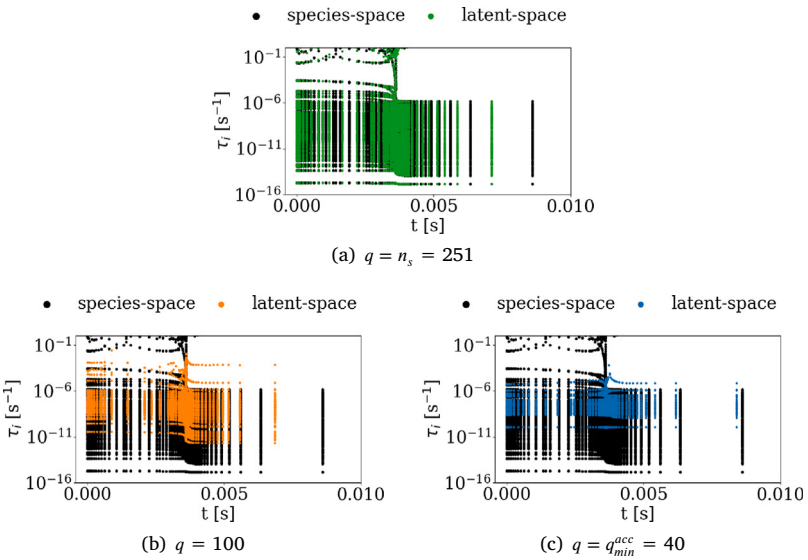


Fig. 9. Chemical time scales of the species system, $\boldsymbol{J}_{\phi}$, and the latent space, $\boldsymbol{J}_{\boldsymbol{z}}$ on time for three compression levels, $q = 40$, 100, 251. The test case is methane using Aramco V1.3 mechanism at $\varphi = 0.9$ and $T_0 = 1440$ K.

Table 3
Breakdown of ODE integration cost for the species and latent solvers at different compression levels q. The test case is methane oxidation using the AramcoC1C3 mechanism.

Number of steps	species 860	$q = 251$ 502	$q = 40$ 552
CPU Time for ODE integration [s]	0.354	0.588	0.152
CPU time per step [s]	4.1e−04	1.2e−03	2.8e−04
* Space transformations	0%	46%	14%
* Chemistry source term evaluation	100%	54%	86%

Table 4
Average computational speed-up of the latent variable solver as a function of mechanism size.

Fuel	Mechanism	Number of species	Speed-up factor
Methane	Lumped (CRECK)	111	1.81
Methane	Detailed (Aramco C1C3)	251	2.43
Propane	Lumped (CRECK)	111	1.43
Propane	Detailed (Aramco C1C3)	251	1.95
n-Heptane	Lumped (CRECK)	252	1.39
n-Heptane	Detailed (LLNL 3.1)	654	2.04

4.7. Impact of kinetic mechanisms on solver performance

This section examines how the choice of the kinetic mechanism affects solver performance. In particular, we compared the lumped mechanisms from the CRECK modelling group with the detailed mechanisms Aramco C1C3 and LLNL 3.1 for the propane and n-heptane cases, respectively. The comparison was conducted for both species and latent models across all test conditions (Section 3 for details) at the optimal reduction level, q_{opt}, using OpenSMOKE++ integrator.

Fig. 10 compares the computational cost of species and latent approaches for both lumped and detailed mechanisms in propane and n-heptane combustion. For propane (Fig. 10(a)), the latent solver shows the lowest variance in computational time across conditions, thanks to the use of a global basis trained over a range of equivalence ratios. The reduction in computational cost is more pronounced for the detailed mechanism (Aramco) than for the lumped one (CRECK), with average savings of 45% and 27%, respectively. Similar trends are observed for n-heptane (Fig. 10(b)), where the detailed mechanism achieves a reduction of about 50%, compared to 26% for the lumped mechanism. These results highlight significant computational savings for both fuels across all test conditions, with the most pronounced benefits observed for detailed mechanisms.

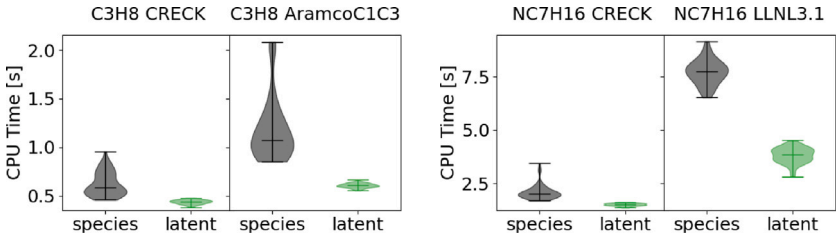
To further illustrate the scalability of the latent variable solver, Table 4 reports the average computational speed-up factor as a function of the total number of species for methane, propane, and n-heptane, considering both lumped and detailed mechanisms. The results confirm that the latent framework becomes increasingly advantageous as the kinetic mechanism size grows, with larger or more detailed mechanisms exhibiting greater computational gains.

The observed trends are closely related to the characteristics of both the kinetic mechanism and the latent model, which is highly sensitive to small reconstruction errors due to the non-linear error propagation (Section 4.1).

The CRECK mechanisms employ lumping strategies that reduce the number of species while preserving the overall reaction structure. However, since the number of reactions remains high, the dimensionality reduction potential in the latent space is inherently limited. In contrast, both detailed mechanisms include more species and fewer reactions, resulting in larger systems with more variables. This allows for a greater dimensionality reduction and, consequently, more substantial computational savings. However, the variable reduction when using C1C3 mechanism with propane as a fuel (i.e., activating its full reaction pathway set) is slightly more limited as the mechanism models a network of vastly different chemical reactivities, while the n-heptane mechanism is composed by subgroups with similar kinetics characteristics, allowing for a higher compression. Unlike lumped mechanisms, the reduction becomes more pronounced as the fuel molecule size increases, since the number of species grows and the number reaction-species ratio decreases [23].

5. Conclusions

This study presents a generalization of the PC-transport approach into a latent variable (LV) framework, demonstrated for 0D combustion simulations. The key contribution introduced in this work is the projection of species tolerances onto the latent space, allowing the solver to adapt tolerances individually for each variable. This resolves a limitation of PC-transport methods with direct chemistry computation,



(a) Propane (b) N-heptane

Fig. 10. Comparison on the computational time of species and latent model for propane and n-heptane fuels using lumped and detailed mechanisms, across all test conditions.

which were previously constrained to small kinetic mechanisms due to fixed tolerance settings. The proposed approach ensures accurate and stable integration, extending the applicability of latent space methods to detailed kinetic mechanisms. Importantly, the latent variable framework allow to reduce the dimensionality while keeping information about all the species initially present in the mechanism, which is a key difference compared to other chemistry reduction methods.

The solver was systematically validated across score scales, fuels, and kinetic mechanisms, demonstrating high accuracy and significant computational savings. The average speed-up factors were approximately 2x for propane and n-heptane and 2.4x for methane. Furthermore, the influence of kinetic mechanisms on the solver performance was assessed. For a given fuel, detailed mechanisms allowed for larger dimensionality reduction and computational cost savings compared to lumped mechanisms, for which speed-up factors were systematically $\approx$ 20–25% lower.

System stiffness was found to be reduced via the latent space projection, when using PCA for dimensionality reduction, as indicated by the larger time steps used by the integrator. Furthermore, robustness analyses confirmed that the LV solver maintains its efficiency regardless of the ODE integrator (OpenSMOKE++ or CVODE) and under moderate extrapolation of equivalence ratios and temperatures. Accuracy remains high, with only minor degradation observed under extreme low-temperature extrapolation due to unrepresented low-temperature chemistry in the training data.

Overall, the proposed LV framework is a robust and computationally efficient solution for simulating ideal reactors with detailed kinetic mechanisms. Combining dimensionality reduction with *projected tolerances* overcomes the primary barrier that previously limited PC-based latent solvers to small mechanisms. The approach not only allows for accurate and cost-effective zero-dimensional combustion simulations with detailed chemistry but also establishes the foundation for extension to multi-dimensional reactive flows, where computational cost can otherwise be prohibitive. In particular, the framework can be applied to multi-dimensional solvers based on the operator-splitting approach, to replace the 0D chemistry computation with the proposed LV framework.

CRediT authorship contribution statement

Eva Muñoz: Writing – original draft, Writing – review & editing, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Mohammad Rafi Malik:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization. **Alberto Cuoci:** Writing – review & editing. **Hong G. Im:** Funding acquisition. **Alessandro Parente:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

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.

Acknowledgments

E. Munoz acknowledges the financial support of the Fonds National de la Recherche Scientifique (FRS-FNRS), Belgium and Erasmus+ programme of the European Union.

Data availability

Data will be made available on request.

References

[1] Jolliffe IT. Principal component analysis. 2nd ed. New York: Springer; 2002.
[2] Muñoz E, Dave H, D'Alessio G, Bontempi G, Parente A, Clainche SL. Extraction and analysis of flow features in planar synthetic jets using different machine learning techniques. Phys Fluids 2023;35:94107.
[3] Sutherland JC, Parente A. Combustion modeling using principal component analysis. Proc Combust Inst 2009;32:1563–70.
[4] Zdybał K, Sutherland JC, Parente A. On the effect of manifold topology in reduced-order modeling of turbulent combustion. 2023.
[5] Malik MR, Isaac BJ, Coussement A, Smith PJ, Parente A. Principal component analysis coupled with nonlinear regression for chemistry reduction. Combust Flame 2018;187:30–41.
[6] Pope SB. Small scales, many species and the manifold challenges of turbulent combustion. Proc Combust Inst 2013;34:1–31.
[7] Biglari A, Sutherland JC. A filter-independent model identification technique for turbulent combustion modeling. Combust Flame 2012;159:1960–70.
[8] Mirgolbabaei H, Echekki T. The reconstruction of thermo-chemical scalars in combustion from a reduced set of their principal components. Combust Flame 2015;162:1650–2.
[9] Echekki T, Mirgolbabaei H. Principal component transport in turbulent combustion: A posteriori analysis. Combust Flame 2015;162:1919–33.
[10] Isaac BJ, Thornock JN, Sutherland J, Smith PJ, Parente A. Advanced regression methods for combustion modelling using principal components. Combust Flame 2015;162:2592–601.
[11] Owoyele O, Pal P. ChemNODE: A neural ordinary differential equations framework for efficient chemical kinetic solvers. Energy AI 2022;7:100118. http://dx.doi.org/10.1016/J.EGYAI.2021.100118.
[12] Zhang T, Yi Y, Xu Y, Chen ZX, Zhang Y, E W, Xu ZQJ. A multi-scale sampling method for accurate and robust deep neural network to predict combustion chemical kinetics. Combust Flame 2022;245:112319. http://dx.doi.org/10.1016/J.COMBUSTFLAME.2022.112319.
[13] Lam SH, Coussis DA. Understanding complex chemical kinetics with computational singular perturbation. In: Symposium (International) on Combustion. vol. 22, Elsevier; 1989, p. 931–41. http://dx.doi.org/10.1016/S0082-0784(89)80102-X.
[14] Galassi RM, Valorani M. A multi-resolution sparse hash table strategy for efficient numerical integration of stiff combustion systems using the G-scheme. Combust Theory Model 2025. http://dx.doi.org/10.1080/13647830.2025.2563536.
[15] Cuoci A, Frassoldati A, Faravelli T, Ranzi E. Numerical modeling of laminar flames with detailed kinetics based on the operator-splitting method. Energy Fuels 2013;27:7730–53.
[16] Malik MR, Vega PO, Coussement A, Parente A. Combustion modeling using principal component analysis: A posteriori validation on Sandia flames D, E and F. Proc Combust Inst 2021;38:2635–43.
[17] Coussement A, Isaac BJ, Gicquel O, Parente A. Assessment of different chemistry reduction methods based on principal component analysis: Comparison of the MG-PCA and score-PCA approaches. Combust Flame 2016;168:83–97.

[18] Malik MR, Galassi RM, Valorani M, Im HG. A combined PCA-CSP solver for dimensionality and stiffness reduction in reacting flow simulations. Proc Combust Inst 2024;40:105532.
[19] Ranzi E, Frassoldati A, Stagni A, Pelucchi M, Cuoci A, Faravelli T. Reduced kinetic schemes of complex reaction systems: Fossil and biomass-derived transportation fuels. Int J Chem Kinet 2014;46:512–42.
[20] Metcalfe WK, Burke SM, Ahmed SS, Curran HJ. A hierarchical and comparative kinetic modeling study of C1 C2 hydrocarbon and oxygenated fuels. Int J Chem Kinet 2013;45:638–75.
[21] Pelucchi M, Cavallotti C, Faravelli T, Klippenstein SJ. H-abstraction reactions by OH, HO2, O, O2 and benzyl radical addition to O2 and their implications for kinetic modelling of toluene oxidation. Phys Chem Chem Phys 2018;20:10607–27.
[22] Mehl M, Pitz W, Westbrook CK, Curran HJ. Kinetic modeling of gasoline surrogate components and mixtures under engine conditions. Proc Combust Inst - PROC COMBUST INST 2011;33:193–200.
[23] Cuoci A, Frassoldati A, Faravelli T, Ranzi E. OpenSMOKE++: An object-oriented framework for the numerical modeling of reactive systems with detailed kinetic mechanisms. Comput Phys Comm 2015;192:237–64.