

Journal Pre-proof

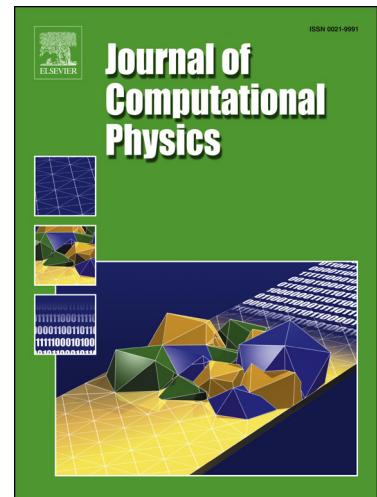
An adaptive time-integration scheme for stiff chemistry based on Computational Singular Perturbation and Artificial Neural Networks

Riccardo Malpica Galassi, Pietro Paolo Ciottoli, Mauro Valorani and Hong G. Im

PII: S0021-9991(21)00770-1

DOI: <https://doi.org/10.1016/j.jcp.2021.110875>

Reference: YJCPH 110875



To appear in: *Journal of Computational Physics*

Received date: 27 July 2021

Revised date: 27 October 2021

Accepted date: 24 November 2021

Please cite this article as: R. Malpica Galassi, P.P. Ciottoli, M. Valorani et al., An adaptive time-integration scheme for stiff chemistry based on Computational Singular Perturbation and Artificial Neural Networks, *Journal of Computational Physics*, 110875, doi: <https://doi.org/10.1016/j.jcp.2021.110875>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2021 Published by Elsevier.

Highlights

- Stiffness in kinetics ODE can be removed by projecting out the fast timescales.
- The CSP solver explicitly integrates the local non-stiff (slow) reduced order model.
- The slow system is evolved with a pace larger than that provided by implicit solvers.
- ANNs are employed to provide a surrogate model for the projection basis.
- The integration scheme is robust to the basis reconstruction errors.

1 An adaptive time-integration scheme for stiff chemistry based on
 2 Computational Singular Perturbation and Artificial Neural
 3 Networks

4 Riccardo Malpica Galassi^{a,*}, Pietro Paolo Ciottoli^b, Mauro Valorani^b, Hong G. Im^c

5 ^a*Aero-Thermo-Mechanics Laboratory, École Polytechnique de Bruxelles, Université Libre de Bruxelles, Avenue F.D.
 6 Roosevelt 50, Brussels 1050, Belgium;*

7 ^b*Mechanical and Aerospace Engineering Department, Sapienza University of Rome, Via Eudossiana, 18, Rome
 8 00184, Italy*

9 ^c*Combustion Research Center, King Abdullah University of Science and Technology, Thuwal, 23955-6900, Saudi
 10 Arabia.*

11 **Abstract**

12 We leverage the computational singular perturbation (CSP) theory to develop an adaptive time-
 13 integration scheme for stiff chemistry based on a local, projection-based, reduced order model
 14 (ROM) freed of the fast time-scales. Its construction is such that artificial neural networks (ANN)
 15 can be plugged-in as cheap surrogates of the local projection basis, which is a state function, to
 16 alleviate the computational cost, without sacrificing the geometrical and physical foundation of the
 17 method. In fact, the solver relies on the synthetic basis in place of the more expensive on-the-fly
 18 calculated basis, i.e. the eigenvectors of the Jacobian matrix of the chemical source term, to define
 19 the local *slow invariant manifold* (SIM) and the projection matrix, then integrates explicitly the
 20 projected, i.e., non-stiff, chemical source term.

21 We explore the feasibility of the ANN-accelerated CSP solver by training a set of ANNs to
 22 predict the projection basis vectors given the local chemical composition in a hydrogen/air homo-
 23 geneous reactor problem. To enhance the smoothness of the basis vectors and reduce the recon-
 24 struction error, we introduce a *constrained* Jacobian formulation which removes the state hetero-
 25 geneity due to the presence of temperature along with chemical species, and takes the derivatives
 26 enforcing the absolute enthalpy conservation. The test problem highlights the robustness of this
 27 ANN approach, arising from the relatively low requirements on the basis accuracy with respect to
 28 the requested integration accuracy.

29 **Keywords:** combustion; multi-scale dynamical systems; model reduction; slow invariant

1 manifold; autoignition

2 **1. Introduction**

3 We are experiencing the age of big data, witnessing an exponential growth of data availability,
 4 stemming from both numerical simulations and measurement devices, either in the form of ex-
 5 periments or operational sensors. This data prosperity, along with the tremendous advancements
 6 in the state-of-the-art computational resources, is driving towards an extensive usage of artificial
 7 neural networks (ANN) to learn from data and subsequently predict, especially when a model
 8 is not available or it is overly expensive. ANNs are learning algorithms that inductively model
 9 an input-output relationship through interconnected groups of nodes, or neurons, aggregated into
 10 layers. The learning procedure consists of the adaptation of the network, i.e. the adjustment of
 11 the weights assigned to each connection, to optimally handle a task represented by the implicit
 12 data input-output mapping. A trained network minimizes the error between its outputs and the
 13 data. Since ANNs are made of a multitude of identical neurons, they offer the possibility of be-
 14 ing massively parallelized by nature. This is one of the main reasons behind the extraordinary
 15 spread of today's ANN applications, especially on GPUs where the highly parallel nature of the
 16 networks architecture can be exploited to maximize the computational speed-up with respect to
 17 classic algorithms.

18 The extensive development and dissemination of numerical tools and libraries capable of train-
 19 ing ANNs with easily customizable features made the usage of ANNs highly attractive in several
 20 engineering areas, including the field of numerical combustion. The numerical simulation of react-
 21 ing flows is burdened by the large dimensionality and stiffness of chemical kinetics ODEs. Several
 22 widespread reduction techniques rely on an *a priori* tabulation of lower-dimensional manifolds
 23 in composition space, capable of describing the most important features of a multi-dimensional
 24 flame. The tabulated variables are then retrieved at run-time instead of solving the ODEs, carrying
 25 extensive computational savings by keeping sufficient accuracy. Numerous attempts of introduc-
 26 ing ANNs in the computational architecture of reacting CFD have been proposed, mainly with the

*riccardo.malpica.galassi@ulb.be

1 goal of replacing look-up tables with the ANN acting as surrogate model of the tabulated func-
 2 tions [1–6]. The absence of any data retrieval and interpolation necessities, together with a low
 3 memory storage compared to large look-up tables, make ANNs attractive for gaining significant
 4 speed-ups at the price of an acceptable accuracy loss.

5 On the other hand, the direct integration of chemical ODEs is desired when the tabulated
 6 manifolds are not able to accurately represent the key dynamical features of interest, e.g., in com-
 7 bustion regimes in which chemical and flow time/length-scales are comparable, encountered in
 8 high Reynolds, Karlovitz and Damköhler number conditions. In such cases, the hypothesis at
 9 the foundation of manifold methods, i.e. the assumption that the internal structure of the flame
 10 front remains unaffected by the flow field fluctuations, breaks down. Direct numerical simulations
 11 (DNS) with detailed chemistry are suited to investigate detailed flow-chemistry interactions as
 12 highly localized and transient phenomena, such as flame quenching, re-ignition, edge flame for-
 13 mation, soot formation and quenching. However, solving a large system of ODEs with chemical
 14 reaction terms heavily affects the computational budget because of (i) the large dimensionality due
 15 to the number of chemical species requiring an evolution equation, and (ii) the stiffness associated
 16 to the wide spectrum of intrinsic chemical timescales, which either limits the integration timestep
 17 of explicit methods to extremely slow paces, or substantially increases the computational cost by
 18 requiring an implicit, usually multi-step, integration method, such as the backward-differentiation
 19 formulas.

20 Hence, another realm of application of ANNs in reacting flow CFD is in the alleviation of
 21 the stiff kinetics ODEs integration cost. Wan et al. [7] employed convolutional neural networks
 22 (CNNs) trained with two-dimensional images of species mass fractions and temperature obtained
 23 with a 35-species detailed chemistry, to predict the chemical source terms of 11 target species in
 24 three-dimensional turbulent flame simulation. However, the CNN-based reduced chemical model
 25 requires 2D detailed simulations of a representative reacting flow to be trained. Alternatively, Peng
 26 and Pinkowski [8] used an ANN architecture to directly predict the chemical composition at the
 27 following time step of the ODE integration, effectively substituting the integrator. An individual
 28 ANN was trained for each of the 8 chemical species involved in their H₂/O₂ chemical system. They
 29 showed satisfactory accuracy in extrapolated predictions of homogeneous reactors, except at low

1 temperatures. A similar approach was pursued by Zhang et al. [9], who proposed a binary ANN
 2 structure in which two separate networks, a low-temperature and a high-temperature ones, predict
 3 the thermochemical state at a following time, outperforming CVODE in homogeneous reactors
 4 integration. Lapointe et al. [10] trained a neural network to select the most-efficient ODE solver
 5 in an operator-splitting context on a cell-by-cell basis by predicting CPU time and error of a set of
 6 ODE solvers for a given thermochemical state. The idea behind this study is that the selection of a
 7 specific ODE solver (and a specific set of tolerances) for a combustion problem is sub-optimal in
 8 numerous cells, depending on the stiffness of the chemical step. Explicit or implicit schemes may
 9 yield the best accuracy and performance depending on the thermochemical state to be integrated,
 10 e.g. a fully-burnt mixture or a low-temperature auto-igniting mixture.

11 Following the quest for optimal, reliable and affordable ODE solvers for combustion applica-
 12 tions, we propose an adaptive ODE integration explicit scheme for stiff chemistry and explore the
 13 opportunity for an ANN acceleration, in view of GPU-based massive DNS applications.

14 The integration scheme is based on the Computational Singular Perturbation (CSP) method
 15 [11–15]. There exists two variants of explicit stiff solvers that are constructed within the CSP
 16 framework. The first stiff solver, here referred to as the *CSP solver*, was presented in [16] and
 17 exploits the local CSP fast/slow decomposition to adaptively remove the stiffness by filtering out
 18 the fast scales. The second stiff solver, referred to as the *G-Scheme method*, presented in [17, 18],
 19 not only removes the stiffness adaptively by filtering out both the fast *and* the slow scales, leaving
 20 an intermediate set of active scales, but it also adapts the number of ODEs with the dimension of
 21 the active subspace dynamically during the time integration. This way the numerical integration
 22 is accomplished by solving a number of ODEs typically much smaller than the dimension of the
 23 original problem, with corresponding savings in computational cost.

24 Both solvers effectively replace the original stiff system with a non-stiff reduced order model
 25 (ROM) on the fly, where the contribution of the fast (and slow) time scales is not resolved but
 26 simply accounted for with a *closure* model involving only algebraic corrections to the numerically
 27 integrated active time scales. The non-stiff ROM can be integrated at the pace of the fastest active
 28 time scale which is typically much slower (by several orders of magnitude) than the fastest scale
 29 of the original system, with a corresponding significant saving in the computational workload of

1 the numerical integration of the detailed kinetics.

2 As a first attempt, the present study considers the first type of solver, the *CSP solver*. This
 3 scheme relies on the availability of the local CSP projection basis which in turn requires the com-
 4 putation of an eigen-system at each time-step. By exploiting the circumstance that the CSP pro-
 5 jection basis is a state function, we aim at replacing it with a less computationally demanding
 6 surrogate by employing a properly trained ANN.

7 In contrast to other techniques in which the ANN predicts the chemical state at a future time, in
 8 the present approach the unavoidable accuracy loss due to the ANN reconstruction does not affect
 9 the accuracy of determining the chemical state. In fact, small errors in the projection basis at most
 10 translate into contamination between fast and slow scales, moving the time-scales gap towards
 11 faster scales. Hence, instead of affecting the accuracy, such errors only reduce the computational
 12 efficiency by forcing the integrator to proceed at a reduced time step. This *dynamically informed*
 13 construction is indeed robust to small errors in the CSP projection basis.

14 The combination of machine learning with projection-based model reduction was similarly
 15 exploited in [19] to learn physics-inspired system parameters, such as the coefficients of a Proper
 16 Orthogonal Decomposition (POD).

17 The paper is structured as follows: we will briefly review the CSP theory and describe how
 18 it translates into a stiff integration scheme of the first type, the *CSP solver*. We will then present
 19 how ANNs can be employed to accelerate the integration. Lastly, we will validate the solver with
 20 reference to homogeneous batch reactor simulations.

21 2. The CSP solver

22 The CSP solver exploits the availability of a local low-dimensional manifold of the chemical
 23 dynamics to extract the fast timescales from the stiff chemical source term, thereby adaptively
 24 removing stiffness. Then, the resulting non-stiff equations are integrated with explicit schemes,
 25 such as Runge-Kutta, with larger time-steps. The computational bottleneck of the CSP solver is
 26 the calculation of the CSP projector, which is a state function resulting from the eigen-system of
 27 the Jacobian matrix of the local chemical source term. Especially with large kinetic mechanisms,

1 the cost of this evaluation becomes prohibitive, since it scales with $N^{2.5}$ to N^3 , where N is the
 2 number of chemical species.

3 In the following, we will briefly review the main conceptual framework to build the CSP solver.

4 Consider a set of autonomous stiff ODEs defined by

$$5 \quad \frac{dz}{dt} = g(z), \quad z \in \mathbb{R}^{N+1} \quad (1)$$

6 Here, g is a non-linear chemical source term, and the number of equations is $N + 1$, where N is the
 7 number of species present in the reactive mixture and the additional unknown is the temperature
 8 whose evolution equation is derived by enforcing the conservation of energy. Stiffness is often a
 9 feature of multi-scale dynamics for Eq. (1), so that this class of system can develop a dynamics
 10 characterized by combined slow and fast components. The role of the processes proceeding at
 11 the fast scales is to constrain the processes evolving at the slow scales within a lower dimensional
 12 manifold embedded in the $(N+1)$ -dimensional space of the full dynamics. This lower dimensional
 13 manifold is usually referred to as the Slow Invariant Manifold (SIM). A SIM attracts nearby orbits,
 14 possesses a lower dimensionality than the phase space and represents the backbone around which
 15 the trajectories of the system dynamics are organized. Intuitively, SIMs are invariant, exponentially
 16 attracting manifolds on which neighboring orbits collapse. This implies that, apart from a possibly
 17 short transient, the “slow dynamics” evolves onto a lower-dimensional manifold, along which the
 18 original stiffness of the system can be softened, because “fast” modes, which are exhausted, can
 19 be removed by projecting them out.

20 The mathematical procedure that allows us to solve the slow dynamics purified from the fast
 21 scales involves a projection operation of the original vector field $g(z)$ over the SIM. The projection
 22 operator can be obtained in many different ways [20, 21]. Here, we follow the CSP method so that
 23 the projection operator is built upon the so called CSP vectors.

24 Consider a set of linearly independent column vectors $\mathbf{A} = \{\mathbf{a}_i\}_{i=1,\dots,N+1}$ with $\mathbf{a}_i \in \mathbb{R}^{N+1}$ and a
 25 set of row vectors $\mathbf{B} = \{\mathbf{b}^i\}_{i=1,\dots,N+1}$ with $\mathbf{b}^i \in \mathbb{R}^{N+1}$ orthonormal to \mathbf{a}_i . For each thermochemical
 26 state $z = \{T, Y_i\}$, the column vectors of \mathbf{A} identify a frame of reference of the tangent space \mathbb{T}_z of
 27 the manifold \mathcal{M} within which the system dynamics evolves.

¹ The orthogonality property provides the following identities:

$$\begin{array}{ll} \text{2} & AB = I \quad BA = I \end{array} \quad (2)$$

³ Hence, the vector field in Eq. (1) can be expressed as the sum of $N + 1$ modal contributions as

$$\begin{array}{ll} \text{4} & \frac{dz}{dt} = I \cdot g = A(B \cdot g) = \sum_{i=1}^{N+1} \mathbf{a}_i f^i \end{array} \quad (3)$$

⁵ where f^i is the mode amplitude of the i -th mode given by

$$\begin{array}{ll} \text{6} & f^i = \mathbf{b}^i \cdot g, \quad i = 1, \dots, N + 1. \end{array} \quad (4)$$

⁷ It can be readily shown that the time evolution of the mode amplitudes f^i satisfies the following
⁸ set of ODEs

$$\begin{array}{ll} \text{9} & \frac{df^i}{dt} = \sum_{k=1}^{N+1} f^k \left(\frac{d\mathbf{b}^i}{dt} + \mathbf{b}^i \cdot J_g \right) \cdot \mathbf{a}_k, \quad i = 1, \dots, N + 1, \end{array} \quad (5)$$

¹⁰ where J_g is the Jacobian matrix of g with respect to z .

¹¹ The identification of the basis vectors \mathbf{a}_k with the right eigenvectors α_k of J_g , and consequently
¹² the co-vectors \mathbf{b}_i with the left eigenvectors β^k yields a first order approximation of the CSP vectors
¹³ (since it amounts to set the contribution $d\mathbf{b}^i/dt$ to zero in Eq. (5)), so that the time evolution of
¹⁴ each mode amplitude f^i is decoupled from the others, and this time evolution is associated with
¹⁵ λ_i , i.e., the i -th eigenvalue of J_g :

$$\begin{array}{ll} \text{16} & \frac{df^i}{dt} = \sum_{k=1}^{N+1} f^k (\mathbf{b}^i \cdot J_g) \cdot \mathbf{a}_k = \sum_{k=1}^{N+1} f^k \Lambda_k^i = f^i \lambda_i, \quad i = 1, \dots, N + 1, \end{array} \quad (6)$$

¹⁷ where Λ_k^i is the diagonal matrix (Jordan form) of the eigenvalues of J_g . Hence, a time-scale
¹⁸ $\tau^i := 1/|\lambda_i|$ is associated to each of the i -th mode contributions in Eq. (3), and all time-scales are
¹⁹ sorted from the smallest to the largest in magnitude¹, so that:

$$\begin{array}{ll} \text{20} & \tau^1 < \dots < \tau^M \ll \tau^{M+1} < \dots < \tau^{N+1} \end{array} \quad (7)$$

¹The eigenvalues can be complex conjugates as well as with positive or negative real part; in both events we sort the eigenvalues on the basis of their absolute value

1 where a large time scale gap between τ^M and τ^{M+1} is expected to exist and their ratio can take
 2 the role of the small parameter $\epsilon = |\tau^M/\tau^{M+1}| \ll 1$ in a singularly perturbed reduced model. We
 3 also assume that the fastest M time-scales are dissipative, that is, the components of the vector
 4 field that generate them drive the system towards the exponentially attracting SIM. The criterion
 5 to identify the gap and the number of exhausted modes M is detailed in [18].

6 It follows that the vector field \mathbf{g} in Eq. (3) can be partitioned in two contributions, one related
 7 to the M fast time scales, and the other to the $(N + 1) - M$ slow time scales, so that Eq. (1) reads

$$8 \quad \frac{dz}{dt} = \sum_{r=1}^M \mathbf{a}_r f^r + \sum_{s=M+1}^{N+1} \mathbf{a}_s f^s. \quad (8)$$

9 The occurrence of the time scale gap ϵ allows us to partition the tangent space \mathbb{T}_z at the location
 10 z in two sub-spaces: the fast sub-space \mathbb{F} spanned by the column vectors $\{\mathbf{a}_1 \dots \mathbf{a}_M\}$, and the
 11 slow sub-space \mathbb{S} spanned by the column vectors $\{\mathbf{a}_{M+1} \dots \mathbf{a}_{N+1}\}$, so that the tangent space can be
 12 obtained by the direct sum of the two sub-spaces, i.e., $\mathbb{T}_z = \mathbb{S} \oplus \mathbb{F}$. After a time period of the order
 13 of the fastest of the slow scales, τ^{M+1} , the mode amplitudes f^r in the fast sub-space \mathbb{F} have attained
 14 a negligibly small magnitude because of Eq. (5), so that the following algebraic equations hold:

$$15 \quad f^r \approx 0 \quad \text{for } r = 1, \dots, M, \quad (9)$$

16 All the state points that satisfy Eq. (9) form the locus of an $(N + 1 - M)$ -dimensional SIM in an
 17 $(N + 1)$ -dimensional space.

Such construction has already been exploited for the analysis of reacting flows [22–29] and the reduction of chemical models [30–33]. In this work, we specifically resort to the explicit time-integration framework proposed in [16] whose construction assumes that the SIM act as a constraint for the slow dynamics of the ODEs in Eq. (1), according to the reduced order model involving the following set of Differential-Algebraic Equations (DAE):

$$\frac{dz}{dt} \approx \sum_{s=M+1}^{N+1} \mathbf{a}_s f^s = \left(\sum_{s=M+1}^{N+1} \mathbf{a}_s \mathbf{b}^s \right) \mathbf{g} = P \mathbf{g} \quad (10a)$$

$$f^r \approx 0 \quad r = 1, \dots, M \quad (10b)$$

1 where P is a projection matrix defined as $P = A_s B^s = \left(\sum_{s=M+1}^{N+1} \mathbf{a}_s \mathbf{b}^s \right)$ and Eq. (10b) represents the
 2 SIM, i.e., the sub-space originating from the exhaustion of the fast scales.

3 The idea of a CSP-based stiff solver proposed in [16] was further explored in [34, 35] in search
 4 of ways to improve its computational efficiency by resorting to look-up tables for the evaluation
 5 of the projection matrix.

6 The numerical integration in [16] involved at each time-step a two-stage process: first, the ex-
 7 plicit integration of the slow dynamics, Eq.(10a), for a time-step $\delta t \sim \mathcal{O}(\tau^{M+1})$, and next, a *radical*
 8 *correction* operation to account for the action of the omitted fast processes. This construction em-
 9 bodies the idea of an adaptive, local reduced order model (ROM) in which the fast processes are
 10 not explicitly resolved over the fast scales, since their action is accounted for with the projection
 11 step that brings instantaneously any state off the SIM onto the SIM, so as to satisfy Eq.(10b).

12 Note that the role of the number of exhausted modes M is pivotal: it adaptively determines
 13 the degree of reduction of the model, i.e., the cutoff time-scale τ^{M+1} that separates the slow re-
 14 solved from the fast unresolved time scales. The criterion to determine M , as detailed in [18],
 15 compares the overall contribution in time (over their associated time-scale) of the modes $\delta z =$
 16 $\sum_{i=1}^M \int_t^{t+\tau^i} \mathbf{a}_i f^i dt$ with a measure of error representing the “zero” of Eq.(10b). The error vector is
 17 defined as $\delta z_{err} = rtol \cdot |z| + atol$. The choice of *rtol* and *atol* allows the user to trade accuracy for
 18 computational cost savings. In fact, higher values for *rtol* and *atol* induce a larger number M of
 19 exhausted modes, i.e., a higher degree of model reduction and a larger integration time-step, with
 20 loss of accuracy in the *fast* contributions only. This feature expresses one of the main advantages
 21 of a CSP solver with respect to canonical implicit solvers, in which accuracy has to be pursued
 22 with respect to all variables at all time scales. Note that the CSP tolerances above, which represent
 23 error thresholds on the SIM definition, should not be confused with the error tolerances on the
 24 solution accuracy employed as convergence criteria in stiff ODE solvers that solve the full model
 25 of ODEs.

26 In this work, we propose an improved version of the scheme introduced in [16], such that
 27 the radical correction step is performed *before* and *after* the slow dynamics integration, so that
 28 the slow-fast splitting scheme in [16] now becomes a fast-slow-fast splitting. The advantage of
 29 the fast-slow-fast splitting is that any possible rotation of the CSP basis due to the action of non-

1 linearities during the slow dynamic update is properly accounted for in the identification of the
 2 exhausted modes. This feature prevents the occurrence of numerical noise in the identification of
 3 the exhausted modes and effectively allows the integrator to march during the slow dynamics with
 4 the largest possible time step. Moreover, when an operator splitting approach is adopted to account
 5 for the coupling between transport and kinetics in spatially non homogeneous system described by
 6 set of PDEs, the application of the first radical correction step brings the state brought off-manifold
 7 by the action of the transport step back on the nearest kinetic SIM. This way the kinetic step can
 8 proceed at the τ^{M+1} scale without the need for any burn-in period, which instead deteriorates the
 9 performance of backward differentiation formula (BDF) solvers in operator splitting strategies,
 10 where re-initialization is mandatory at the beginning of each CFD time-step [36].

11 *2.1. Integration algorithm*

12 The integration algorithm consists of the following steps:

- 13 1. Given the state $\hat{z} = z(t_n)$ at the discrete time step t_n and the associated chemical source
 14 term $\mathbf{g}(\hat{z})$, compute the Jacobian matrix J_g and the CSP basis, e.g., the eigen-system of J_g ,
 15 comprising the timescales $\tau^i = 1/|\lambda_i|$ and the right and left eigenvectors $A = \{\mathbf{a}_i\}_{i=1,\dots,N+1}$ and
 16 $B = \{\mathbf{b}^i\}_{i=1,\dots,N+1}$
- 17
- 18 2. Compute the mode amplitudes $f^i = \mathbf{b}^i(\hat{z}) \cdot \mathbf{g}(\hat{z})$ for $i = 1, \dots, N+1$ and apply the first radical
 19 correction to approach the local manifold at time t_n :

$$20 \quad z^{\mathcal{M}} = \hat{z} - \sum_{r=1}^{M(z(t_{n-1}))} \mathbf{a}_r(\hat{z}) f^r(\hat{z}) \tau^r(\hat{z}) \quad (11)$$

21 Hence, the manifold is approached by relaxing each fast mode according to its time-scale
 22 with a leading order approximation. It is equivalent to one Newton iteration of Eq.(10b) and
 23 it acts along the fast directions only. Note that the number of exhausted modes $M(z(t_{n-1}))$ is
 24 the one computed in the previous time-step. In the case of the first time-step, $M(z(t_0))=0$, so
 25 that the first radical correction is not performed.

26

- 1 3. Evaluate at z^M the number of exhausted modes $M(z^M)$ by using the criterion detailed in [18]

2 employing the basis evaluated at state \hat{z} in step (1).

3

4 4. Build the slow projection matrix with the basis vectors computed in step (1) at state \hat{z} and

5 the number of exhausted modes $M(z^M)$ computed in step (3):

$$P = I - \sum_{r=1}^{M(z^{\mathcal{M}})} \mathbf{a}_r(\hat{z}) \mathbf{b}^r(\hat{z}) \quad (12)$$

- 8 5. Integrate the slow dynamics to advance z^M to z^* using the projected source term $\mathbf{g}_{slow} = Pg$.
9 By construction the state z^M lies in the SIM and the projected source term \mathbf{g}_{slow} is purified
10 from the fast scales². Thus, this integration is not affected by stiffness, and it can be advanced
11 with any explicit scheme of integration with a time-step $\delta t \approx \gamma \tau^{M+1}$, where $\gamma \sim \mathcal{O}(1)$ is a
12 safety factor.
13
14 6. Compute the mode amplitudes $f^i(\hat{\mathbf{z}}, z^*) = \mathbf{b}^i(\hat{\mathbf{z}}) \cdot \mathbf{g}(z^*)$ for $i = 1, \dots, N + 1$ and apply the
15 second radical correction to approach the local manifold at time $t_{n+1} = t_n + \delta t$ to eventually
16 obtain the state $\mathbf{z}(t_{n+1})$ at the next time step:

$$z(t_{n+1}) = z^* - \sum_{r=1}^{M(z^{\mathcal{M}})} a_r(\hat{z}) f^r(\hat{z}, z^*) \tau^r(\hat{z}) \quad (13)$$

The computational cost of this integration scheme is mostly associated with the calculations of the Jacobian matrix and its eigen-system which are both required to create the basis vectors in step (1). We showed in [18] that this cost scales with the power 2.5 to 3 of the number of system variables, i.e., chemical species, making this cost prohibitive for large mechanisms. In [35] the effort to enhance the efficiency of the CSP basis was addressed by tabulating the CSP objects

²Because all fast modes have zero amplitude on the SIM, it follows that for any state on the SIM

$$g_{slow}(z^{\mathcal{M}}) = g(z^{\mathcal{M}}) \quad \text{if} \quad z^{\mathcal{M}} \in SIM$$

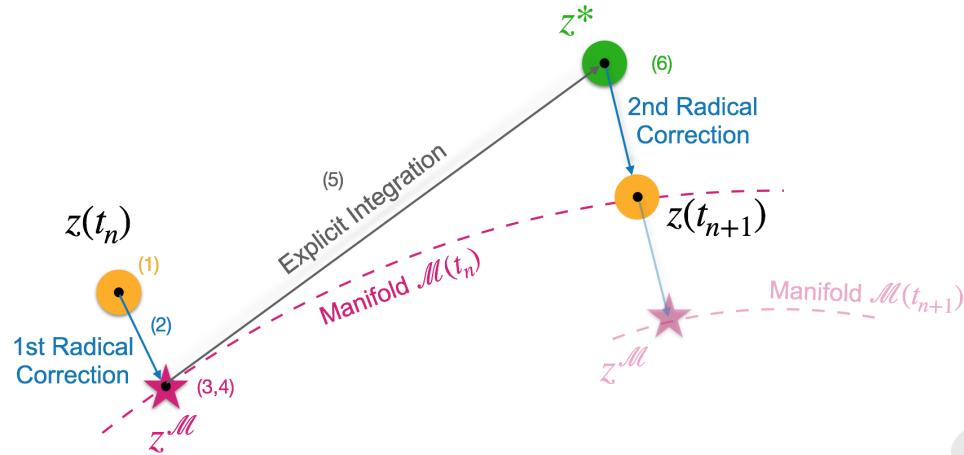


Figure 1: A pictorial visualization of the CSP stiff integration scheme.

needed to build the slow projection matrix, namely the number of exhausted modes M , the fast CSP vectors and co-vectors $\{\mathbf{a}^r, \mathbf{b}_r\}_{r=1,\dots,M}$, and the fast timescales τ^1, \dots, τ^r . Moreover, in [18] we investigated the idea of reusing the basis when its time rate of change is smaller than a certain measure, exploiting the fact that the *smooth* basis rotation is concentrated only where the effects of the system non-linearities are strong. A suitable and cost-effective measure of basis rotation is the norm of the directional derivative of the vector field \mathbf{g} evaluated at the state \mathbf{z} along a direction \mathbf{p} that perturbs all the entries of the state vector itself:

$$\mathbf{p}(\mathbf{z}) := rtol \frac{\mathbf{z}}{|\mathbf{z}|} + atol \quad (14)$$

$$\mathbf{g}_{,\mathbf{p}}(\mathbf{z}) := J_g(\mathbf{z}) \cdot \mathbf{p}(\mathbf{z}) \approx \frac{\delta \mathbf{g}(\mathbf{z} + \mathbf{p})}{\delta \mathbf{p}}. \quad (15)$$

¹ The computation of the norm of $\mathbf{g}_{,\mathbf{p}}(\mathbf{z})$ requires $N + 1$ derivatives only. Whenever this norm is smaller than a given threshold, the basis is not updated. We were able to reuse the basis over an average of 5 consecutive integration steps so as to achieve a computational speed-up factor of 2-5, depending on the mechanism size.

⁵ An alternative reduction of the computational cost of the CSP solver involves a neural-network model capable of replacing the Jacobian and eigen-system calculation by predicting the full CSP basis as a function of the thermo-chemical state. Subsequently, the calculations of the number of

1 exhausted modes M_{ANN} and the fast timescales are performed using the ANN-predicted basis, in
 2 order to improve the consistency between these objects and therefore the stability of the algorithm.
 3 In fact, M_{ANN} represents how many *reconstructed* modes are exhausted and the timescales are the
 4 inverse of the *reconstructed* eigenvalues.

5 Clearly, the idea of basis reuse is applicable in the ANN context. The main goal of using the
 6 ANN-basis is to enable the GPU operation and scalability of the solver for massive and parallel
 7 DNS calculations.

8 3. A discussion on the CSP basis

9 Figure 2 shows the temporal evolution of the $(N+1)$ eigenvalues in a homogeneous reactor in
 10 which the initial mixture is stoichiometric H₂/air at 1200 K and atmospheric pressure. The eigen-
 11 values, depicted in grey, are computed *a posteriori* on the states $\mathbf{y}(t)$ resulting from the solution
 12 of the initial value problem (1) with the 9-species kinetic mechanism by Yetter et al. [37], and
 13 are presented in the form $\Lambda_i = \text{sign}(\lambda_i) \cdot \log_{10}(1 + |\lambda_i|)$ to show their order of magnitude and sign
 14 simultaneously. Note that eigenvalues whose magnitude is smaller than $10^0=1$, corresponding to
 15 time-scales longer than 1 s, collapse on the x-axis.

16 Seven non-zero eigenvalues out of a 10-equations system are discernible, since the rank of
 17 J_g is deficient: 3 atomic conservation laws (H, O, N), expressed by linear combinations of the
 18 system variables rates-of-change, reveal themselves as 3 zero-eigenvalues. Each eigenvalue is
 19 associated to an eigenvector, which is a direction in the $(N+1)$ -dimensional space. Note that
 20 the eigenvectors associated to the zero-eigenvalues are the invariant directions associated to the
 21 atomic conservation: their co-vectors are such that $f^{cons} = \mathbf{b}^{cons} \cdot \mathbf{g} = 0$, that is the aforementioned
 22 linear combination. The 7 remaining directions span the tangent space which represents all local
 23 changes of mixture composition allowed by atomic conservation laws, that is the thermo-chemical
 24 composition space effectively visited by \mathbf{z} for a given initial condition.

25 Despite the eigenvalues smoothness noticeable in Fig. 2, the components of the associated
 26 eigenvectors do not exhibit a smooth behaviour. Such occurrence is due to the heterogeneity of
 27 the units of measure of the state \mathbf{z} , which contains both temperature (Kelvin) and chemical species

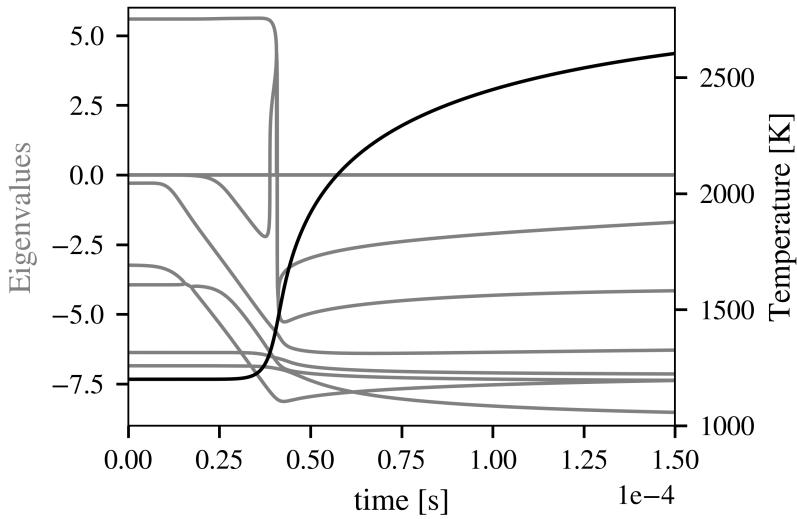


Figure 2: Time evolution of the chemical eigenvalues (grey) and system temperature (black) in a constant-pressure homogeneous H₂/air ignition at atmospheric pressure and stoichiometric equivalence ratio.

¹ (mass fractions), and manifests itself as highly stretched eigenvectors along the temperature direction. This heavily affects the smoothness of the other components, which are subject to magnitude ² spikes anytime the temperature component changes sign, crossing zero. Note that the temperature ³ source term is a combination of the species source terms in which the coefficients are the species ⁴ enthalpies:

$$\dot{\omega}_T = \sum_{k=1}^{N_s} h_k(T) \dot{\omega}_k(\mathbf{Y}, T). \quad (16)$$

⁶ The mixture enthalpy is a conserved property in an adiabatic homogeneous reactor; however, such ⁷ invariance serving as a nonlinear constraint does not affect the Jacobian rank. In fact, the non-linear ⁸ enthalpy constraint is violated by the linearized system, since the temperature derivatives in J_g are ⁹ taken without enforcing the algebraic constraint between temperature and composition given by ¹⁰ $h(T, \mathbf{Y}) = \text{const}$. This constraint is a manifold for which the solution set of $h(T, \mathbf{Y}) = \text{const}$ is ¹¹ represented by the graph of a function $T = f_T(\mathbf{Y})$, such that $h(f_T(\mathbf{Y}), \mathbf{Y}) = \text{const}$. In other words, ¹² there exists an additional conservation direction in phase space belonging to the local orthogonal ¹³ space, which removes one degree of freedom to the system dynamics. To make this explicit,

1 a straightforward application of the chain rule to compute derivatives of composite functions is
 2 sufficient. This results in an N -dimensional species-only Jacobian matrix with enforced enthalpy
 3 invariance, in which the (i, k) -th entry of the Jacobian matrix of the species-only chemical source
 4 term $\mathbf{g}_Y = \{g_{Y,i}\}$, i.e., the partial derivative of $g_{Y,i}$ with respect to the k -th mass fraction Y_k , is
 5 computed following the chain rule for the derivative of $g_{Y,i}$ with respect to $f_T(\mathbf{Y})$:

$$6 \quad \frac{dg_{Y,i}(\mathbf{Y}, f_T(\mathbf{Y}))}{dY_k} = \frac{\partial g_{Y,i}}{\partial Y_k}(\mathbf{Y}, T) + \frac{\partial g_{Y,i}}{\partial T}(\mathbf{Y}, T) \frac{\partial f_T(\mathbf{Y})}{\partial Y_k}. \quad (17)$$

7 where $T = f_T(\mathbf{Y})$ expresses the algebraic thermal constraint. Since this algebraic constraint pro-
 8 duces a 1-dimensional manifold in the $(N + 1)$ -dimensional space within which $D(h(\mathbf{Y}, T)) = 0$,
 9 we can compute the derivative $\partial f_T(\mathbf{Y})/\partial Y_l$ as follows:

$$10 \quad \frac{dh(\mathbf{Y}, f_T(\mathbf{Y}))}{dY_k} = \frac{\partial h}{\partial Y_k}(\mathbf{Y}, T) \Big|_{T, Y_{j \neq k}} + \frac{\partial h}{\partial T}(\mathbf{Y}, T) \Big|_{\mathbf{Y}} \frac{\partial f_T(\mathbf{Y})}{\partial Y_k} \Big|_{Y_{j \neq k}} = 0. \quad (18)$$

11 so that

$$12 \quad \frac{\partial f_T(\mathbf{Y})}{\partial Y_k} \Big|_{Y_{j \neq k}} = - \left(\frac{\partial h}{\partial T}(\mathbf{Y}, T) \Big|_{\mathbf{Y}} \right)^{-1} \frac{\partial h}{\partial Y_k}(\mathbf{Y}, T) \Big|_{T, Y_{j \neq k}} \quad (19)$$

13 Since the mixture enthalpy is defined as

$$14 \quad h(T, \mathbf{Y}) = \sum_{i=1, N_s} h_i(T) Y_i = \sum_{i=1}^{N_s} \left(\Delta h_{f,i} + \int C_{p,i}(T) dT \right) Y_i, \quad (20)$$

15 the numerator of Eq. (19) reads:

$$16 \quad \frac{\partial h}{\partial Y_k}(\mathbf{Y}, T) \Big|_{T, Y_{j \neq k}} = h_k(T) \quad (21)$$

17 and the denominator of Eq. (19) reads:

$$18 \quad \frac{\partial h}{\partial T}(\mathbf{Y}, T) \Big|_{\mathbf{Y}} = \sum_{i=1}^{N_s} C_{p,i}(T) Y_i = \tilde{C}_p(T, \mathbf{Y}) \quad (22)$$

19 Replacing Eq. (21) and Eq. (22) in Eq. (19) yields:

$$20 \quad \frac{\partial f_T(\mathbf{Y})}{\partial Y_k} \Big|_{Y_{j \neq k}} = - \frac{h_k(T)}{\tilde{C}_p(T, \mathbf{Y})} \quad (23)$$

21 The expression for the Jacobian J_C of \mathbf{g} with respect to \mathbf{Y} constrained within the manifold $T =$
 22 $f_T(\mathbf{Y})$ is finally:

$$23 \quad J_C = \frac{dg_{Y,i}(\mathbf{Y}, f_T(\mathbf{Y}))}{dY_k} = \frac{\partial g_{Y,i}}{\partial Y_k}(\mathbf{Y}, T) \Big|_{T, Y_{j \neq k}} - \frac{\partial g_{Y,i}}{\partial T}(\mathbf{Y}, T) \Big|_{\mathbf{Y}} \frac{h_k(T)}{\tilde{C}_p(T, \mathbf{Y})} \quad i, k = 1, N. \quad (24)$$

1 Note that, by definition, the dimensions of J_C are $(N \times N)$. This construction enables the treat-
 2 ment of a homogeneous state vector z composed of the species mass fractions only, which in turn
 3 exhibits smooth eigenvectors. Figure 3a shows the 6 non-zero eigenvalues of the *constrained* Jaco-
 4 bian for the same test-problem of Fig.2, overlapped with the eigenvalues of the full system. Note
 5 that one eigenvalue disappears, which is the one related to the violation of the conservation of the
 6 mixture enthalpy produced while perturbing the vector field with respect to temperature, since the
 7 perturbation of temperature is not properly associated with a perturbation in composition.

8 The CSP solver discussed in Sec.2 has been implemented in the species-only N-dimensional
 9 space without modifications, while temperature is updated after the radical corrections in steps (2)
 10 and (6) by inversion of the mixture enthalpy function in Eq.(20).

11 3.1. Eigenvalues ordering

12 The eigenvalues are state functions determined at a given time. The colouring of Fig.3a, i.e.,
 13 the identification of continuous functions in time, is not uniquely defined. In fact, when a set of
 14 eigenvalues is available at a given state z , these can be ordered, say, by the magnitude of their
 15 modulus. The effect of nonlinearities is to alter the magnitude (and thus the ordering) of the
 16 eigenvalues, so that it can happen that pairs of eigenvalues undergo what looks like a switch of
 17 color during the time evolution. It follows that their identification as time-evolving smooth objects
 18 can be carried out only when the full time-series is available. The colouring of Fig.3a represents
 19 the proper identification, based on state-wise correlation and second order-time derivative mini-
 20 mization. Figure 3b shows an enlargement in which *merging*, *splitting*, *crossing* and *mis-crossing*
 21 events are present.

22 A *merging* event happens when two real eigenvalues merge, becoming a pair of complex con-
 23 jugates (see in Fig. 3b the blue/orange pair at $t = 3.4 \times 10^{-5}$ s, and the red/purple pair at 4×10^{-5} s).
 24 A *splitting* event happens when two real eigenvalues emerge from a pair of complex conjugates. A
 25 *crossing* event happens when two real eigenvalues invert their magnitude without becoming com-
 26 plex conjugates. Note that the orange and purple eigenvalues in Fig. 3b (at $t = 3.1 \times 10^{-5}$ s) never
 27 merge or cross; they just approach one to another up to a point and immediately diverge later; this
 28 event is called a *mis-crossing*.

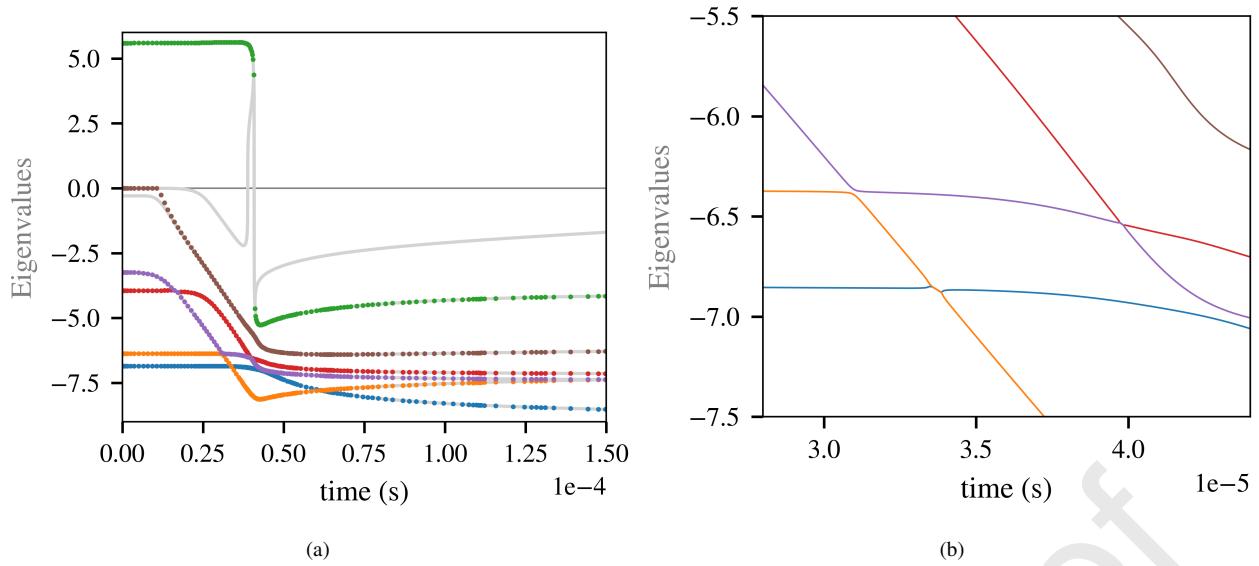


Figure 3: Time evolution of the chemical eigenvalues of the full-Jacobian (grey) and constrained-Jacobian (colored symbols) in a constant-pressure homogeneous H₂/air ignition at atmospheric pressure and stoichiometric equivalence ratio. On the right an enlarged view of the plot on the left to emphasize the occurrence of *merging*, *splitting*, *crossing* and *mis-crossing* events.

After a proper reordering of the eigenvalues, the time evolution of the right eigenvectors associated with the corresponding eigenvalues in Fig.3a, is shown in Fig.4. The eigenvectors are defined up to a multiplicative factor, hence they are normalized to have a unitary norm. The eigenvectors associated with the conserved modes are constant and not shown. The jumps visible in all the eigenvectors except #3 are due to mergings and splittings. When the associated eigenvalue is a complex conjugate, the two conjugate eigenvectors are linearly combined to yield two distinct real eigenvectors (with the first containing the real part and the second the imaginary part) whose magnitude is however uncorrelated with the pre- and post-merge magnitude.

This discussion is irrelevant when the CSP objects are obtained and interpreted as local information (at a given time), such as the local Jacobian and eigen-system. However, the smooth evolution of the objects over time is critical to neural network reconstruction in obtaining the smallest possible converged network.

¹³ For the training dataset purpose, we ignore the complex conjugate transitions, as they are a limited and narrow subset of the total states.

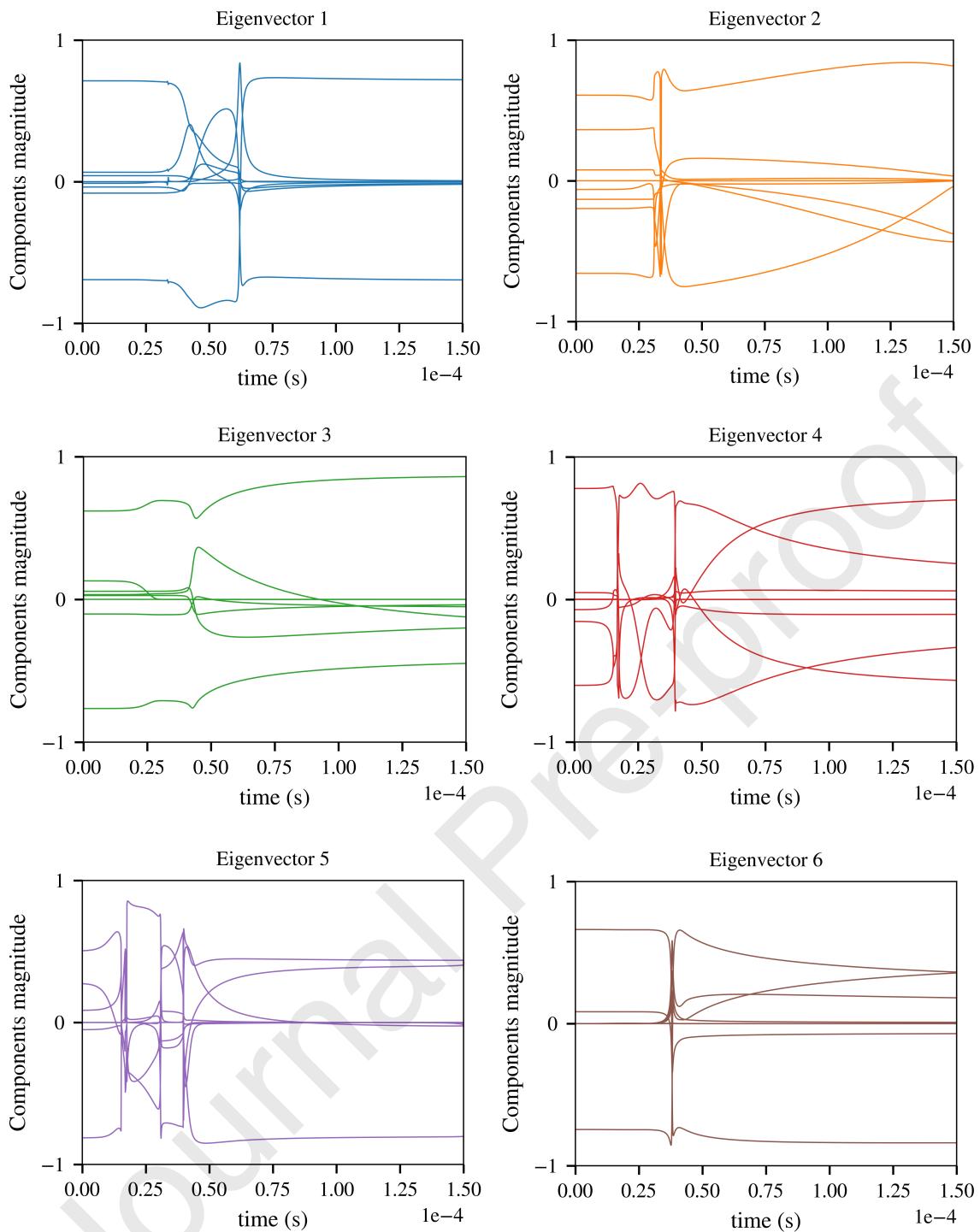


Figure 4: Time evolution of the chemical eigenvectors of the constrained-Jacobian, colored as the associated eigenvalue, in a constant-pressure homogeneous H₂/air ignition at atmospheric pressure and stoichiometric equivalence ratio. Each eigenvector has N components.

1 4. Training an ANN for the prediction of the CSP basis

2 The need for a GPU-ready, scalable and efficient chemistry solver motivates the adoption of
 3 a deep-learning based framework for the estimation of the CSP basis, e.g., the set of right eigen-
 4 vectors of $J_g(z)$ and the associated eigenvalues, which are state functions of z . More specifically,
 5 we train one network for the eigenvalues prediction/reconstruction and one network per each right
 6 eigenvector \mathbf{a}_i prediction/reconstruction. The trained networks will be employed at step # 1 of the
 7 CSP solver algorithm to return the (approximate) eigenvalues and the right eigenvectors in place of
 8 the eigen-system calculation. The left eigenvectors matrix B is computed at run-time by inversion
 9 of the A matrix.

10 A sketch of the CSP-ANN algorithm is depicted in Fig.5. The main aspects of the ANN
 11 modelling are described below.

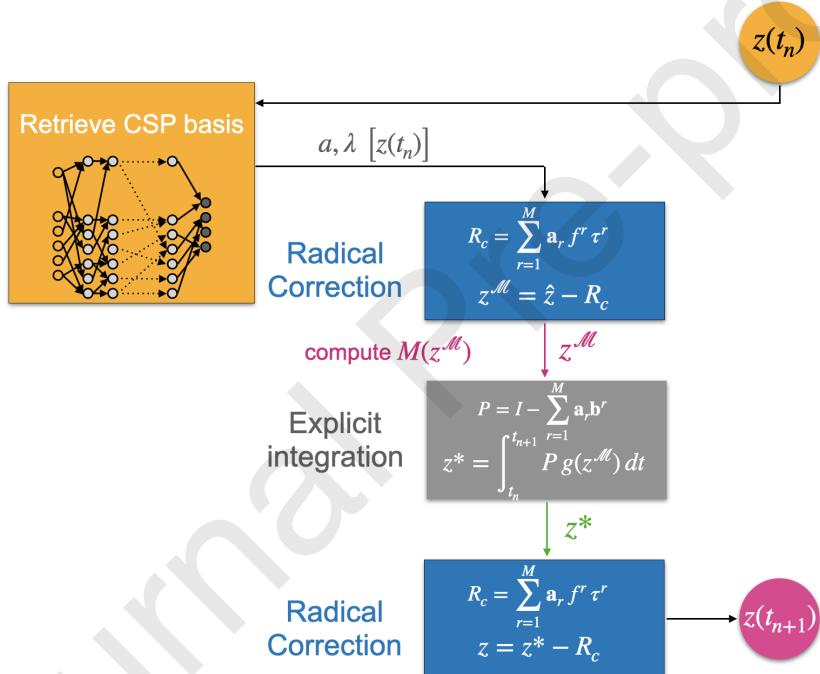


Figure 5: A sketch of the CSP-ANN algorithm.

12 **Data:** The training dataset consists of the thermo-chemical states $\mathbf{y} = \{Y_i\}_{i=1,\dots,N}$ and the corre-
 13 sponding eigen-objects, being either the set of eigenvalues $\Lambda = \{\lambda_i\}_{i=1,\dots,N}$ or the components of the

1 i-th eigenvector $\mathbf{a}_i = \{a_i^j\}_{j=1,\dots,N}$. For the H₂/Air ignition problem, given the 9-species mechanism
 2 by Yetter et al. [37], and considering the number of conserved modes equal to 3, the number of
 3 nontrivial eigenvectors is 6, hence we train 7 neural networks. Each network has N inputs (the
 4 chemical mass fractions) and N outputs (the components of the eigen-object). For the purpose of
 5 validating the solver, the training set is built with a series of 9 ignition trajectories, whose initial
 6 conditions are evenly distributed between 1190 and 1210 K, atmospheric pressure, and stoichio-
 7 metric mixture, and whose CSP eigen-objects are computed with PyCSP³. The total number of
 8 training states is 12300.

9 The dataset is split into three subsets: training, validation, and test sets. Each subset contains
 10 respectively, 80%, 10% and, 10% of the dataset entries. The training set is used to fit the network
 11 parameters. The performance of the architecture is assessed on the test dataset, which contains
 12 unseen samples, to measure the generalization capabilities of the model. A standard normalization
 13 operation, consisting in subtracting the mean and dividing by the standard deviation, is applied to
 14 the data to speed up the learning procedure.

15 **Architecture:** Seven feed forward neural network (FFNN) composed of 2 hidden layers, each
 16 containing 128 neurons, are adopted. After each hidden layer, the rectified linear unit [40] (RELU)
 17 activation function is employed, which is crucial for modelling the non-linear behaviour of the
 18 eigen-objects. The neural networks have been trained with the Pytorch library [41].

19 **Training procedure:** The training phase is performed with Adam optimizer [42] and a step
 20 learning rate scheduler which updates the network parameters based on the gradient of each pa-
 21 rameter, to minimize a cost function $L = \frac{1}{n} \sum_i cost(x_i, \hat{x}_i)$. The cost function is a modified mean
 22 square error (MSE) function:

$$23 L(x_i, \hat{x}_i) = \left(|x_i - \hat{x}_i| + \frac{1}{2} \left| \log_{10} \left(\frac{x_i}{\hat{x}_i} \right) \right|^2 \right), \quad (25)$$

³PyCSP [38] is a Python code which relies on Cantera [39] for the chemical source term estimation, then computes numerically the local Jacobian matrix using first order forward differences formulae, and computes the eigen-system using `numpy.linalg.eig`, which is a wrapper to LAPACK's `geev` routine.

1 where i denotes the index of each element of the ground truth x and predicted \hat{x} output. The
 2 logarithmic part is used to enhance the training accuracy over the smallest data. Each ANN is
 3 trained for 500 epochs with a batch size of 8. The training cost is approximately 1.5 CPU-hours
 4 on a desktop machine.

5 **Evaluation metrics:** The quality of the predictions of the ANN is given by the sum of the
 6 average and standard deviation of the squared error on all testing samples. During the training of
 7 the ANN, the criterion to select the *best model* is given by the minimum of the aforementioned
 8 metric in the validation phase. The *best models* metrics, evaluated on the testing samples, are
 9 reported in Table 1 for the 7 neural networks. Such metrics were also employed to tune the number
 10 of neurons in order to get an optimal ANN complexity, i.e., a complexity which neither under-fits
 11 nor over-fits the training data, represented by a minimum error in the testing dataset.

ANN	Mean Square Error	Mean Error	Max Error
Λ	$1.6 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$	$2.0 \cdot 10^{-1}$
a_1	$7.0 \cdot 10^{-7}$	$2.9 \cdot 10^{-4}$	$2.6 \cdot 10^{-2}$
a_2	$9.2 \cdot 10^{-5}$	$1.1 \cdot 10^{-3}$	$3.3 \cdot 10^{-1}$
a_3	$8.1 \cdot 10^{-9}$	$5.0 \cdot 10^{-5}$	$6.3 \cdot 10^{-4}$
a_4	$2.6 \cdot 10^{-5}$	$8.1 \cdot 10^{-4}$	$1.5 \cdot 10^{-1}$
a_5	$8.1 \cdot 10^{-4}$	$5.6 \cdot 10^{-3}$	$5.1 \cdot 10^{-2}$
a_6	$2.2 \cdot 10^{-7}$	$1.9 \cdot 10^{-5}$	$7.6 \cdot 10^{-3}$

Table 1: Testing metrics of the *best models* for the 7 ANNs.

12 5. Application

13 The goal of this section is twofold: first, the accuracy of the CSP solver (with on-the-fly eigen-
 14 system computation) is compared against a reference implicit BDF stiff solver, namely CVODE.
 15 Subsequently, the performance and robustness of the CSP solver with a reconstructed CSP basis
 16 are demonstrated. All the assessments and comparisons are performed in a test-bed application,
 17 namely an ignition of a 0-dimensional homogeneous reactor.

1 5.1. Accuracy of the CSP solver

2 Figures 6 and 7 show the temperature and H mass fraction profiles obtained via integration of
 3 Eqs. (1) with initial conditions $T = 1200$ K, pressure of 1 atmosphere and stoichiometric H₂/Air
 4 mixture, using CVODE (blue line), the CSP solver with on-the-fly eigen-system calculation (green
 5 line) and the CSP solver with ANN-reconstructed basis (magenta line). The accuracy in CVODE
 6 is prescribed with $rtol = 10^{-9}$ and $atol = 10^{-15}$, while the CSP solver, in both cases, adopts the
 7 CSP tolerances on exhausted modes $rtol = 10^{-2}$ and $atol = 10^{-8}$, and the safety factor γ of the
 8 algorithm step #5 equal to 0.5. In the case of the CSP-ANN integration, the ANNs predict the
 9 eigen-objects of the chemical states produced during the CSP time integration. By construction,
 10 such states are different from the ones employed to train the networks⁴.

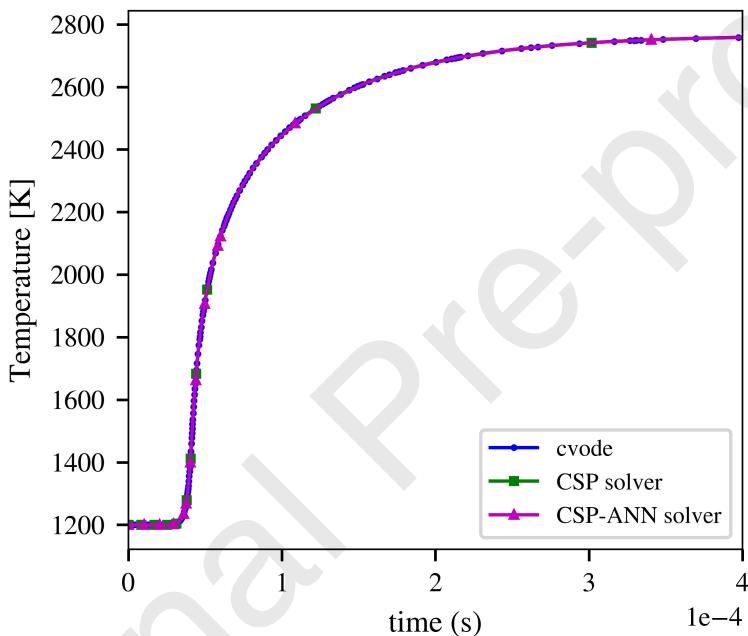


Figure 6: Time evolution of temperature in a constant-pressure homogeneous H₂/air ignition at atmospheric pressure and stoichiometric equivalence ratio, integrated in time with cvode, reference CSP solver and CSP-ANN solver.

11 The accuracy of the CSP solver, in both versions, is very high, as noticeable in Fig. 7b, which
 12 provides an enlarged view of the peak region of the time evolution of the H mass fraction. This

⁴Note that the training database was produced using CVODE, and that the reference trajectory (blue line) of Fig.6 was not included in the training database.

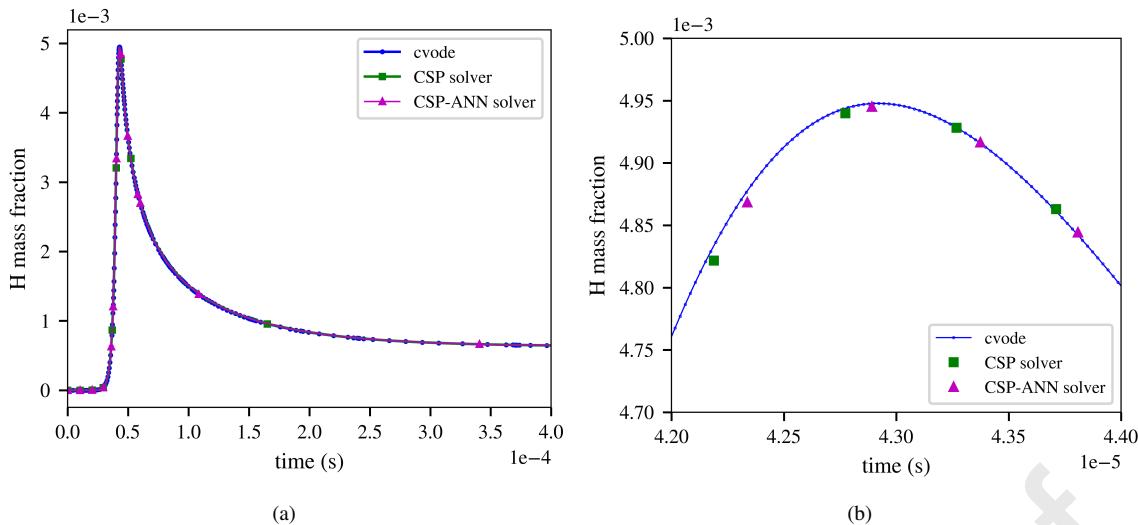


Figure 7: Time evolution of H mass fraction (left) and a zoom in its peak region (right) in a constant-pressure homogeneous H₂/air ignition at atmospheric pressure and stoichiometric equivalence ratio, integrated in time with cvode, reference CSP solver and CSP-ANN solver.

1 demonstrates the most important feature of the CSP solver, i.e. its ability to generate state points
2 that remain close to the reference trajectory, despite that the integration took a much longer time
3 scale compared to CVODE. Note that the CSP solver, being a reduced model, resolves only the
4 slow dynamics, while the fast scales are not resolved, but are filtered out. Nevertheless, the same
5 level of accuracy is preserved by the novel correction algorithm.

6 The integration time-step size along the integration is reported in Fig. 8a for both CVODE and
 7 the CSP solver: the CSP solver adopts time-steps having a magnitude one to two orders larger
 8 than those selected by CVODE. This is consistent with the goal of CVODE to reproduce all time
 9 scales, while the CSP solver aims at reproducing the slow scales only.

5.1.1. Adaptive reduction

11 The degree of reduction of the reduced order model at the core of the CSP solver is ascertained
 12 by the application of the criterion introduced in [18] to *dynamically* detect M , the number of
 13 CSP exhausted modes. As noted in section 2, the time-step size is related to the reciprocal of
 14 the eigenvalue of the $(M+1)$ -th mode; hence, the larger M , the smaller the number of truly active
 15 (slow) degrees of freedom, the larger the step-size can be.

1 The variation of M during the time evolution, as shown in Fig. 8b, demonstrates the ability of
 2 the CSP solver to pursue an automatic reduction of the original dynamical system, which auto-
 3 matically tailors its dimension on the current thermo-chemical state and its time-scale spectrum.
 4 For example, when the mixture is burnt and approaches equilibrium, only one mode is active over
 5 a total of 6, whose characteristic time-scale is of the order 10^{-4} s. It follows that the total number
 6 of steps taken by the CSP solver to approach equilibrium, either when the CSP basis is accurately
 7 evaluated or when it is approximated by an ANN, is 136, against the 1702 steps taken by CVODE.

8 Additional important advantages of the CSP solver compared to the state-of-the-art BDF algo-
 9 rithm are summarized as follows.

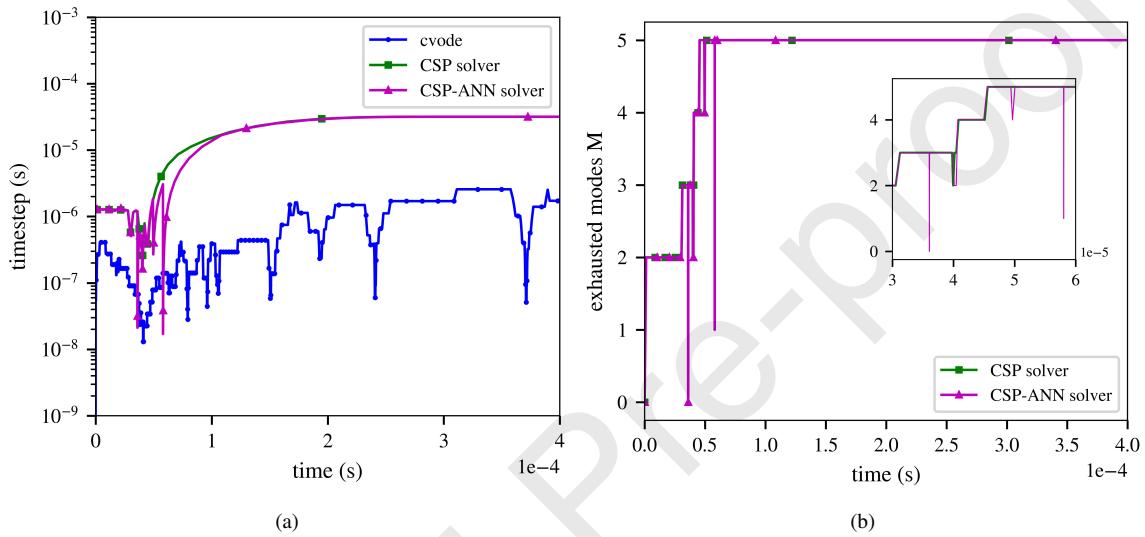


Figure 8: Time evolution of integration time-step (left) and number of exhausted modes (right) in a constant-pressure homogeneous H_2/air ignition at atmospheric pressure and stoichiometric equivalence ratio.

10 5.1.2. The reinitialization problem of BDF

11 It has been repeatedly noted (read for example [36]) that the BDF methods require a significant
 12 level of computational effort concentrated at the beginning of each CFD time step (in an operator
 13 splitting framework to solve PDEs problems) because any multi-step integration algorithm at the
 14 start of the integration step (i) needs to generate the intermediate states of the multi-step formula,
 15 and (ii) the optimal time step is found by a trial and error identification.

1 In contrast, the CSP solver has a negligible initial overhead because (i) it is a one-step scheme
 2 and (ii) the optimal time-step is found on the basis of local information known from the prescribed
 3 initial state.

4 Figure 9, which is an enlargement of Fig. 8a at the beginning of the dynamics, shows a compar-
 5 ison of the initialization phase with CVODE and the CSP solver in terms of integration time-step.
 6 The CSP solver is able to march with a $\delta t \approx 10^{-6}$ s right at the beginning of the integration, while
 7 CVODE takes 14 initialization steps to reach a comparable time-step size (10^{-6} s).

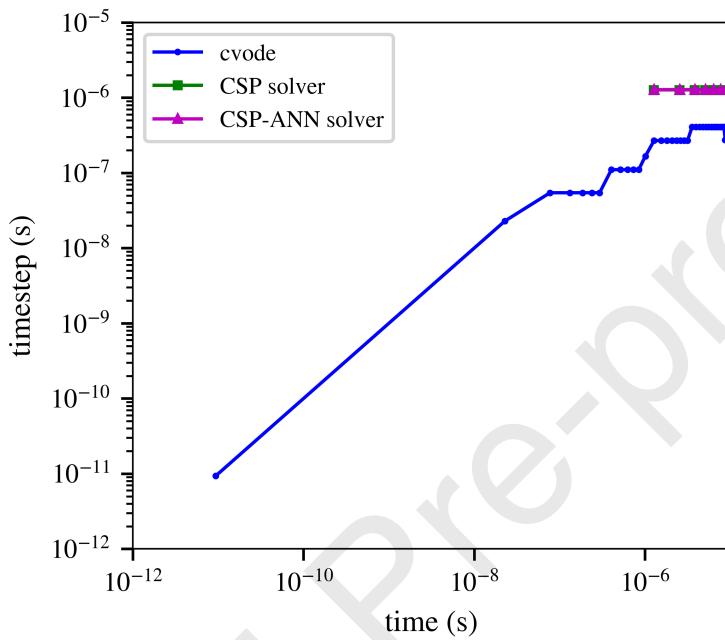


Figure 9: Reinitialization problem: burn-in period of integration time-step in a constant-pressure homogeneous H₂/air ignition at atmospheric pressure and stoichiometric equivalence ratio, integrated in time with cvode, reference CSP solver and CSP-ANN solver.

8 5.1.3. Accuracy of the ANN-basis surrogates

9 The ANN-basis surrogates performed remarkably well in the hydrogen ignition application as
 10 can be appreciated by inspection of the trajectories marked by purple triangles in Fig. 7b. Although
 11 the ANN-basis is only an approximation of the CSP basis, the solver accuracy is not affected, and
 12 in so doing it demonstrates the robustness of the solver design.

13 Inspection of Fig. 8b reveals that for just 2 integration steps out of the total of 136 the basis

1 inaccuracy drives the state far from the SIM. This is visible in Fig. 8b, where the magenta line falls
 2 to 0 or 1 in two occasions, and the time-step size decreases accordingly (Fig.8a). The solver senses
 3 this occurrence and lowers the number of exhausted modes M to drive the state back onto the SIM
 4 by explicitly integrating a larger number of degrees of freedom. In other words, it re-activates the
 5 fast scales to gain accuracy.

6 *5.2. Performance*

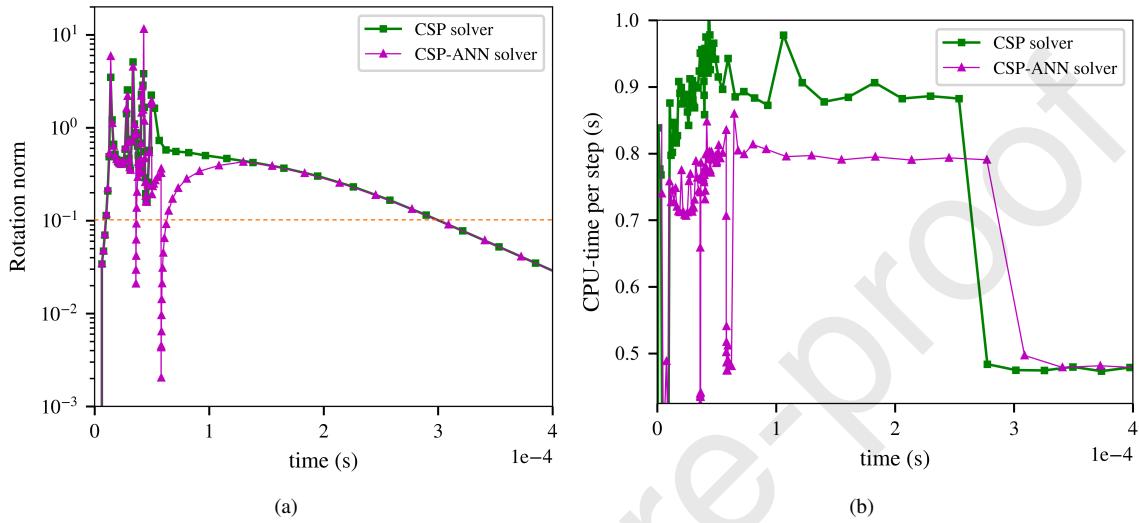


Figure 10: Time evolution of basis rotation norm with reuse threshold in orange (left) and normalized integration CPU-time per-step (right) in a constant-pressure homogeneous H₂/air ignition at atmospheric pressure and stoichiometric equivalence ratio.

7 Figure 10a shows the temporal evolution of the basis rotation norm (Eq.15) for the CSP solver
 8 and CSP-ANN solver. Of the 136 steps generated by the CSP solvers, 72 did not require the com-
 9 putation/retrieval of the basis, as shown by the rotation norm being lower than the reuse threshold
 10 in such steps. Note that this rotation measure is consistent with the actual basis rotation shown
 11 in Fig. 4, despite being only a rough estimate; the spikes observed in the time interval $0 < t <$
 12 0.7×10^{-4} s correspond to the larger variations and jumps in the eigenvectors components. When
 13 the norm is larger than the threshold, set at 0.1 in this calculation, the basis is computed/retrieved;
 14 otherwise it is reused from the previous integration step. Increasing the threshold implies reusing

1 the basis more often, and with a higher chance to fall off the SIM and bring the integration time-
 2 step to a smaller magnitude.

3 Figure 10b compares the performance of the CSP-ANN solver and the reference CSP solver in
 4 terms of CPU time on a desktop machine. The figure shows distinctly the computational load of
 5 the basis computation/retrieval, since the basis reuse, which takes place steadily after $t = 3 \times 10^{-4}$ s,
 6 almost halves the CPU time per-step. When the basis update is not necessary, the integration cost
 7 is mostly attributed to the explicit time integration (step #5 of the algorithm presented in Sec.2.1),
 8 which is remarkably low, especially if compared to implicit schemes. When the basis update is
 9 necessary ($t < 3 \times 10^{-4}$ s), the ANN basis brings a computational saving of $\approx 10\%$ with respect
 10 to the 9×9 eigen-system computation, the latter being already very cost-efficient due to its small
 11 size. Note that, as the number of equations grows, the CPU time breakdown becomes dominated
 12 by the eigen-system calculation (see [18]), which raises from the $\approx 50\%$ of the present H₂-system,
 13 to 90-95% of O(100)-species mechanisms.

14 5.3. Robustness of the CSP-ANN construction

15 The accuracy assessment in Sec. 5.1 revealed the algorithm's intrinsic mechanism that kicks
 16 in to preserve accuracy when errors in the basis reconstruction are present. In this subsection, we
 17 shed more light on the solver robustness by employing it outside the basis training range of [1190-
 18 1210] K, thus relying on extrapolation for the basis reconstruction. Figure 11(a) shows the number
 19 of integration steps taken by the CSP solver with on-the-fly eigen-system calculation (green line)
 20 and the CSP solver with ANN-reconstructed basis (magenta line) to reach chemical equilibrium,
 21 for varying initial temperature. The solvers' settings are those reported in Sec.5.1. As expected, the
 22 CSP solver takes approximately the same number of steps to reach equilibrium across the whole
 23 initial temperature range, since the ignition dynamics is similar in nature. Instead, the CSP-ANN
 24 solver takes an increasing number of steps when the initial temperature is decreased/increased out-
 25 side the training range. In fact, as the chemical states become further away from the training set,
 26 the basis reconstruction inaccuracies further increase and drive the state off the SIM, triggering the
 27 fast scales re-activation and increasing the number of steps taken. However, Fig.11(b) shows that

1 the solution accuracy, measured with the ignition delay time⁵, is almost unaffected by such occur-
 2 rence, and remains well within 5% of the CSP solver solution across the whole initial tempera-
 3 ture range.

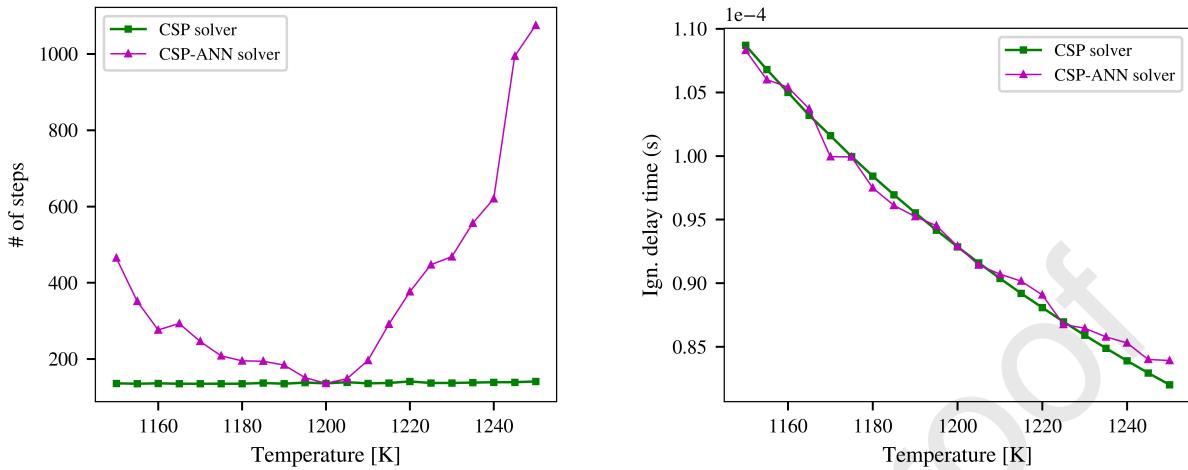


Figure 11: Number of integration steps taken to reach equilibrium (a) and ignition delay time (b), for varying initial temperature. CSP solver (green line) and CSP-ANN solver (magenta line).

6. Discussion

The framework presented in this work takes essential steps towards the application of machine learning techniques to chemical ODEs solvers for combustion. The example of Sec.5 served as a feasibility study to ascertain the accuracy, performance and robustness of a CSP-ANN construction. All these features tightly depend on the quality of the basis reconstruction delivered by the ANN, even though a certain degree of robustness to small reconstruction errors is assured by the integration algorithm construction. Note that the tested ANNs exhibit reconstruction errors (see Table 1) which are comparable to the chemical species concentrations: the direct application of such ANNs to either the source term or the chemical state prediction would most certainly fail. Hence, the physics-based algorithm construction helps lowering the ANNs accuracy requirement, thus size and associated computational cost.

⁵The ignition delay time is measured as the peak OH mass fraction.

1 The logical next step is to augment the ANNs' training dataset with a broader family of chem-
 2 ical states, e.g., ignition trajectories stemming from a larger spectrum of initial conditions. This
 3 would extend the simultaneous applicability of the solver, i.e., the comprehensiveness, which in
 4 the example of Sec.5 was limited to ignition trajectories originating from initial conditions similar
 5 to the ones employed in the relatively narrow training dataset generation. In doing so, additional
 6 challenges are expected to emerge: the structure of the eigen-objects observed in Figs.3a and 4 as
 7 a function of time has a tendency to morph into different patterns as the initial condition changes.
 8 Merging, splitting, crossing and mis-crossing events have substantially different time-behaviors,
 9 making the eigenvalue identification ambiguous when spanning over largely different initial condi-
 10 tions. As an example, Fig. 12 shows the eigenvalues patterns for the same homogeneous mixture,
 11 igniting from T=1200 K (left) and T=900 K (right). The eigenvalues depicted in red and purple
 12 are characterized by substantially different behaviors and mutual interactions, e.g., at T=900 K the
 13 merging is not present and the mis-crossing has become a crossing. The associated eigenvectors
 14 (not shown) experience abrupt changes of magnitude and direction accordingly. This example re-
 15 veals discontinuities in eigenvalues and eigenvectors across the initial conditions, which should be
 16 regarded as surfaces separating contiguous regions of coherent patterns in the chemical composi-
 17 tion space. Such discontinuities cause the ANNs' training to be less efficient or even impractical.

18 As such, the eigenvalues' patterns will need further investigation and additional developments
 19 will be essential to overcome this obstacle: multiple ANNs, associated to different patterns, might
 20 be necessary to efficiently learn the underlying eigen-structures. Such patterns might not neces-
 21 sarily occur in the time domain, but rather in the chemical composition space. This recognition
 22 activity lends itself to the application of clustering techniques: in a large database of ignition tra-
 23 jectories, groups of chemical states, sharing a similar eigen-object pattern, should be identified,
 24 e.g., with unsupervised machine learning techniques, and affiliated to a specific ANN. This leads
 25 to a *polyadic* ANN strategy, in which the appropriate ANN is chosen at run-time based on the
 26 proximity of the chemical state to the ANN's training cluster. This approach is similar to the one
 27 in [9], where a low-temperature and a high-temperature ANNs are present. The recognition and
 28 assignment of a chemical state, i.e., CFD grid point, to a pre-computed cluster was already pursued
 29 in [43]. Such a strategy also keeps the ANNs' size and cost low, and limits the need for extrapola-

1 tion. We plan to develop these techniques as we proceed to the application of the CSP-ANN solver
 2 to reacting CFD. Besides, note that the ANNs' training should be tailored to the final application:
 3 the applicability generalization goes against computational efficiency.

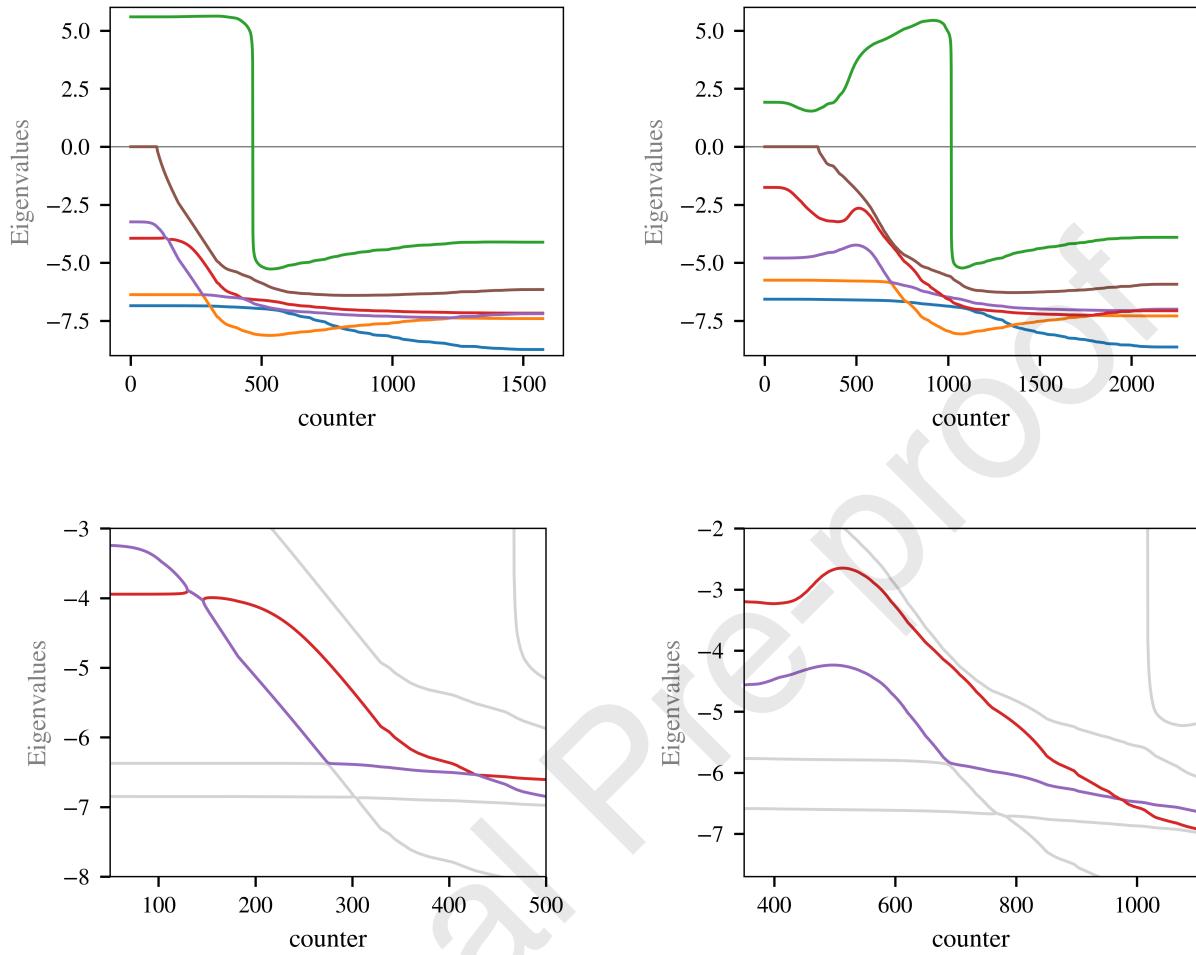


Figure 12: Patterns of the chemical eigenvalues in a constant-pressure homogeneous H_2/air ignition at atmospheric pressure, stoichiometric equivalence ratio, $T=1200$ K (left) and $T=900$ K (right). Enlarged views (bottom row) to emphasize the occurrence of different *merging*, *splitting*, *crossing* and *mis-crossing* events involving the red and purple eigenvalues.

4 The performance analysis of the example in Sec.5 reveals that the improvement may not ap-
 5 pear to be ground-breaking. However, there are two points that merit further investigations on
 6 the CSP-ANN solver, namely (i) the possibility of GPU application, which is an enabling step

1 towards reacting CFD on GPUs and carries along well-known scalability advantages, and (ii) the
 2 ANN retrieval cost, which is expected to scale more effectively with the number of species, with
 3 respect to the eigen-system calculation, which scales as $N^{2.5}$ to N^3 . The latter point will depend
 4 on the training efficacy of the larger ANNs in case of more complex kinetic mechanisms. Unlike
 5 classic eigen-system computation algorithms, whose improvement is a niche topic in the scientific
 6 community, the interest in machine learning, especially in ANN performance scaling with data
 7 and dimensions, is wide and growing.

8 Nonetheless, the extension to larger kinetic mechanisms remains an open issue, since with in-
 9 creasing complexity the eigen-objects grow in dimensionality and weaken in smoothness. More-
 10 over, the spectral density increases, causing a multiplication of crossing events that amplify the
 11 obstacle described in the paragraph above. This effort, as well as suitable testings in reactive CFD
 12 on GPU architectures, will be subjects of future work.

13 7. Conclusions

14 The evermore increasing interest in high-fidelity numerical simulations of combustion with
 15 detailed chemistry, burdened by the well-known challenges in terms of dimensionality and stiffness
 16 of the chemical dynamics, may significantly benefit from the simultaneous growth and spread
 17 of the usage of artificial neural networks on GPUs, which in turn are exploiting today's large
 18 availability of data. In this context, we proposed a stiff integration scheme for chemistry based on
 19 the fast time-scales filtering provided by computational singular perturbation, whose construction
 20 naturally lends itself to ANN-acceleration. The projection of the chemical source terms onto
 21 a basis obtained by solving the eigen-system of its Jacobian matrix allows to remove the fast
 22 degrees of freedom and evolve the slow dynamics only, with a pace larger than that provided by
 23 implicit solvers, making it attractive for reactive CFD within operator splitting strategies, which
 24 will additionally benefit from the absence of an initialization overhead [36].

25 The neural-network model replaces the most computationally expensive part of the algorithm,
 26 i.e. the local basis calculation, while maintaining a high level of robustness: the integration scheme
 27 still relies on physics and geometric foundations, and errors in the ANN reconstruction, i.e. mis-
 28 alignments of the reconstructed *slow invariant manifold* with respect to the true manifold, are felt

1 as detours to higher-dimensional spaces, which are naturally recovered with a slower integration
 2 pace. In order to increase the reconstruction accuracy, the smoothness of the basis vectors was im-
 3 proved with the introduction of a *constrained* Jacobian formulation, whose definition stems from
 4 the enthalpy conservation principle. Smoother eigen-objects allowed accurate predictions with
 5 relatively small networks characterized by 2 hidden layers containing 128 neurons each.

6 We demonstrated the feasibility and robustness of the CSP-ANN construction within a 0-D
 7 homogeneous reactor simulation, featuring a hydrogen-air mixture at constant pressure. The pres-
 8 ence of the ANN model as basis surrogate did not affect accuracy and slightly boosted the compu-
 9 tational performance. We expect that more significant savings will be achieved with larger kinetic
 10 mechanisms, in which the eigen-system operation cost dominates the overall CPU time. Such
 11 performance assessment is left to a future work. Moreover, the algorithm performance benefits
 12 from every improvement in the ANN domain, applied to both the basis reconstruction, as we
 13 showed in this work, and in other computationally expensive steps, such as the chemical source
 14 term calculation, as addressed by Barwey and Raman [44].

15 Finally, the ANNs' architecture employed in this work can be improved using most sophisti-
 16 cated hyper-parameters' tuning procedures, e.g., optimization, and regularization. Moreover, other
 17 machine learning techniques deserve to be explored, such as decision trees and k-nearest neigh-
 18 bors, following the quest for low cost and high effectiveness of the basis surrogate. These are again
 19 left for future investigation.

20 Acknowledgements

21 We acknowledge the fruitful discussions and the technical support kindly offered by Dr. Shivam
 22 Barwey and Professor Venkat Raman at the University of Michigan, Ann Arbor, MI, USA, and Dr.
 23 Mattia Soldan at KAUST, Saudi Arabia. R. Malpica Galassi acknowledges the financial support
 24 of the Fédération Wallonie-Bruxelles (FWB), Cellule Europe.

25 Disclosure statement

26 The authors declare that they have no conflict of interest.

1 **Funding**

2 This work was supported by KAUST OSR-2020-CCF-1975-35 Subaward Agreement.

3 **References**

- 4 [1] M. Hansinger, Y. Ge, and M. Pfitzner. Deep residual networks for flamelet/progress variable tabulation with
5 application to a piloted flame with inhomogeneous inlet. *Combustion Science and Technology*, 2020.
- 6 [2] S. Demir, P. Kundu, and O. Owoyole. Implementation of high dimensional flamelet manifolds for supersonic
7 combustion using deep neural networks. 2020.
- 8 [3] Matthias Ihme, Christoph Schmitt, and Heinz Pitsch. Optimal artificial neural networks and tabulation meth-
9 ods for chemistry representation in les of a bluff-body swirl-stabilized flame. *Proceedings of the Combustion*
10 *Institute*, 32(1):1527–1535, 2009.
- 11 [4] Lucas L.C. Franke, Athanasios K. Chatzopoulos, and Stelios Rigopoulos. Tabulation of combustion chemistry
12 via artificial neural networks (anns): Methodology and application to les-pdf simulation of sydney flame 1.
13 *Combustion and Flame*, 185:245–260, 2017.
- 14 [5] O. Owoyole, P. Kundu, M.M. Ameen, T. Echekki, and S. Som. Application of deep artificial neural networks to
15 multi-dimensional flamelet libraries and spray flames. *International Journal of Engine Research*, 2020.
- 16 [6] Lorenzo Angelilli, Pietro Paolo Ciottoli, Riccardo Malpica Galassi, Francisco E. Hernandez Perez, Mattia Sol-
17 dan, Zhen Lu, Mauro Valorani, and Hong G. Im. *Large eddy simulation with flamelet progress variable approach*
18 *combined with artificial neural network acceleration*. AIAA Scitech 2021 Forum.
- 19 [7] Kaidi Wan, Camille Barnaud, Luc Vervisch, and Pascale Domingo. Machine learning for detailed chemistry re-
20 duction in DNS of a syngas turbulent oxy-flame with side-wall effects. *Proceedings of the Combustion Institute*,
21 000:1–9, 2020.
- 22 [8] W. Y. Peng and Nicolas H. Pinkowski. Efficient and accurate time-integration of combustion chemical kinetics
23 using artificial neural networks. 2017.
- 24 [9] Tianhan Zhang, Yaoyu Zhang, Weinan E, and Yiguang Ju. A deep learning-based ODE solver for chemical
25 kinetics. *arXiv e-prints*, page arXiv:2012.12654, November 2020.
- 26 [10] Simon Lapointe, Sudeeptha Mondal, and Russell A. Whitesides. Data-driven selection of stiff chemistry ode
27 solver in operator-splitting schemes. *Combustion and Flame*, 220:133–143, 2020.
- 28 [11] Sau-Hai Lam and Dimitris A. Goussis. Understanding complex chemical kinetics with computational singular
29 perturbation. *Proc. Comb. Inst.*, 22:931–941, 1988.
- 30 [12] Sau-Hai Lam and Dimitris A. Goussis. Conventional Asymptotics and Computational Singular Perturbation
31 for Simplified Kinetics Modelling. In M.O. Smooke, editor, *Reduced Kinetic Mechanisms and Asymptotic*

- 1 *Approximations for Methane-Air Flames*, number 384 in Springer Lecture Notes, chapter 10, pages 227–242.
 2 Springer Verlag, 1991.
- 3 [13] Sau-Hai Lam and Dimitris A. Goussis. A study of homogeneous methanol oxidation kinetic using csp. *Proc.*
 4 *Comb. Inst.*, 24:113–120, 1992.
- 5 [14] S.H. Lam. Using CSP to Understand Complex Chemical Kinetics. *Combustion Science and Technology*, 89:375–
 6 404, 1993.
- 7 [15] S.H. Lam and D.A. Goussis. The CSP Method for Simplifying Kinetics. *International Journal of Chemical*
 8 *Kinetics*, 26:461–486, 1994.
- 9 [16] Mauro Valorani and Dimitrios A. Goussis. Explicit time-scale splitting algorithm for stiff problems: Auto-
 10 ignition of gaseous mixtures behind a steady shock. *Journal of Computational Physics*, 169(1):44–79, 2001.
- 11 [17] Mauro Valorani and Samuel Paolucci. The G-Scheme: A framework for multi-scale adaptive model reduction.
 12 *Journal of Computational Physics*, 228(13):4665–4701, 2009.
- 13 [18] Mauro Valorani, Pietro Paolo Ciottoli, Riccardo Malpica Galassi, Samuel Paolucci, Temistocle Grenga,
 14 and Emanuele Martelli. Enhancements of the G-Scheme Framework. *Flow, Turbulence and Combustion*,
 15 101(4):1023–1033, 2018.
- 16 [19] Renee Swischuk, Laura Mainini, Benjamin Peherstorfer, and Karen Willcox. Projection-based model reduction:
 17 Formulations for physics-based machine learning. *Computers and Fluids*, 179:704–717, 2019.
- 18 [20] Alessandra Adrover, Francesco Creta, Massimiliano Giona, Mauro Valorani, and Valerio Vitacolonna. Natu-
 19 ral tangent dynamics with recurrent biorthonormalizations: A geometric computational approach to dynamical
 20 systems exhibiting slow manifolds and periodic/chaotic limit sets. *Physica D*, 213:121–146, 2006.
- 21 [21] Alessandra Adrover, Francesco Creta, Massimiliano Giona, and Mauro Valorani. Stretching-based diagnostics
 22 and reduction of chemical kinetic models with diffusion. *Journal of Computational Physics*, 226(2):1442–1471,
 23 2007.
- 24 [22] Dimitris M. Manias, Efstathios-Al. Tingas, Francisco E. Hernández Pérez, Riccardo Malpica Galassi, Pietro
 25 Paolo Ciottoli, Mauro Valorani, and Hong G. Im. Investigation of the turbulent flame structure and topology at
 26 different Karlovitz numbers using the tangential stretching rate index. *Combustion and Flame*, 200:155–167,
 27 2019.
- 28 [23] Abdullah S. AlRamadan, Riccardo Malpica Galassi, Pietro P. Ciottoli, Mauro Valorani, and S. Mani Sarathy.
 29 Multi-stage heat release in lean combustion: Insights from coupled tangential stretching rate (TSR) and compu-
 30 tational singular perturbation (CSP) analysis. *Combustion and Flame*, 219:242–257, 2020.
- 31 [24] Pietro Paolo Ciottoli, Riccardo Malpica Galassi, Lorenzo Angelilli, Alberto Cuoci, Hong G. Im, and Mauro
 32 Valorani. Analysis of wall–flame interaction in laminar non-premixed combustion. *Combustion Science and*
 33 *Technology*, 0(0):1–14, 2019.
- 34 [25] Zhiyi Li, Riccardo Malpica Galassi, Pietro Paolo Ciottoli, Alessandro Parente, and Mauro Valorani. Charac-

- 1 terization of jet-in-hot-coflow flames using tangential stretching rate. *Combustion and Flame*, 208:281–298,
 2 2019.
- 3 [26] Zhiyi Li, Alberto Cuoci, and Alessandro Parente. Large eddy simulation of mild combustion using finite rate
 4 chemistry: Effect of combustion sub-grid closure. *Proceedings of the Combustion Institute*, 2018.
- 5 [27] Efstathios-Al. Tingas. The chemical dynamics of hydrogen/hydrogen peroxide blends diluted with steam at
 6 compression ignition relevant conditions. *Fuel*, 296:120594, 2021.
- 7 [28] Nurina Sharmin and Efstathios-Al Tingas. Dynamics analysis of a jet-fuel surrogate and development of a skele-
 8 tal mechanism for computational fluid dynamic applications. *Journal of Energy Engineering*, 146(6):04020064,
 9 2020.
- 10 [29] Dimitris M. Manias, Efstathios-Al. Tingas, Yuki Minamoto, and Hong G. Im. Topological and chemical char-
 11 acteristics of turbulent flames at mild conditions. *Combustion and Flame*, 208:86–98, 2019.
- 12 [30] Mauro Valorani, Francesco Creta, Filippo Donato, Habib N Najm, and Dimitris A Goussis. Skeletal mechanism
 13 generation and analysis for n -heptane with CSP. *Proceedings of the Combustion Institute*, 31:483–490, 2007.
- 14 [31] Pietro P Ciottoli, Riccardo Malpica Galassi, Pasquale E Lapenna, G Lecce, D Bianchi, F Nasuti, F Creta, and
 15 M Valorani. CSP-based chemical kinetics mechanisms simplification strategy for non-premixed combustion :
 16 An application to hybrid rocket propulsion. *Combustion and Flame*, 186:83–93, 2017.
- 17 [32] Riccardo Malpica Galassi, Pietro P. Ciottoli, Subram Mani Sarathy, Hong G. Im, Samuel Paolucci, and Mauro
 18 Valorani. Automated chemical kinetic mechanism simplification with minimal user expertise. *Combustion and*
 19 *Flame*, 197:439–448, 2018.
- 20 [33] Riccardo Malpica, Mauro Valorani, Habib N Najm, Cosmin Safta, Mohammad Khalil, and Pietro P Ciottoli.
 21 Chemical model reduction under uncertainty. *Combustion and Flame*, 179:242–252, 2017.
- 22 [34] J. C. Lee, H. N. Najm, S. Lefantzi, J. Ray, M. Frenklach, M. Valorani, and D. A. Goussis. A CSP and tabulation-
 23 based adaptive chemistry model. *Combustion Theory and Modelling*, 11(1):73–102, 2007.
- 24 [35] Bert J. Debusschere, Youssef M. Marzouk, Habib N. Najm, Blane Rhoads, Dimitris A. Goussis, and Mauro Valo-
 25 rani. Computational singular perturbation with non-parametric tabulation of slow manifolds for time integration
 26 of stiff chemical kinetics. *Combustion Theory and Modelling*, 16(1):173–198, 2012.
- 27 [36] A. Imren and D. C. Haworth. On the merits of extrapolation-based stiff ODE solvers for combustion CFD.
 28 *Combustion and Flame*, 174:1–15, 2016.
- 29 [37] R. A. Yetter, F. L. Dryer, and H. Rabitz. A comprehensive reaction mechanism for carbon monox-
 30 ide/hydrogen/oxygen kinetics. *Combustion Science and Technology*, 79(1-3):97–128, 1991.
- 31 [38] Riccardo Malpica Galassi. PyCSP: A collection of python tools based on computational singular perturbation for
 32 the analysis of chemically reacting systems. <https://github.com/rmalpica/PyCSP>, 2021. Version 1.0.0.
- 33 [39] David G. Goodwin, Raymond L. Speth, Harry K. Moffat, and Bryan W. Weber. Cantera: An object-oriented
 34 software toolkit for chemical kinetics, thermodynamics, and transport processes. <https://www.cantera.org>,

- 1 2018. Version 2.4.0.
- 2 [40] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung.
3 Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–
4 951, 2000.
- 5 [41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen,
6 Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary De-
7 Vito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith
8 Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle,
9 A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing
Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- 10 [42] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint
arXiv:1412.6980*, 2014.
- 11 [43] Giuseppe D'Alessio, Alessandro Parente, Alessandro Stagni, and Alberto Cuoci. Adaptive chemistry via pre-
12 partitioning of composition space and mechanism reduction. *Combustion and Flame*, 211:68–82, 2020.
- 13 [44] Shivam Barwey and Venkat Raman. A neural network-inspired matrix formulation of chemical kinetics for
14 acceleration on gpus. *Energies*, 14(9):2710, May 2021.

Declaration of interests

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.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:



CRediT author statement

Riccardo Malpica Galassi: Methodology, Software, Investigation, Validation, Writing, Visualization.

Pietro Paolo Ciottoli: Methodology, Software, Visualization.

Mauro Valorani: Conceptualization, Supervision.

Hong G Im: Funding acquisition, Project administration.