

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/348249546>

On-the-fly artificial neural network for chemical kinetics in direct numerical simulations of premixed combustion

Article in Combustion and Flame · April 2021

DOI: 10.1016/j.combustflame.2020.12.038

CITATIONS

19

READS

391

3 authors, including:



Cheng Chi

Otto-von-Guericke-Universität Magdeburg

20 PUBLICATIONS 132 CITATIONS

[SEE PROFILE](#)



D. Thévenin

Otto-von-Guericke-Universität Magdeburg

436 PUBLICATIONS 7,519 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Lattice Boltzmann modeling of multi-species reacting flows [View project](#)



RETERO - Reduction of live fish testing through science and technology [View project](#)

On-the-fly artificial neural network for chemical kinetics in direct numerical simulations of premixed combustion

Cheng Chi^{a,b,*}, Gábor Janiga^a, Dominique Thévenin^a

^a*Laboratory of Fluid Dynamics and Technical Flows, University of Magdeburg “Otto von Guericke”, D-39106 Magdeburg, Germany*

^b*International Max Planck Research School (IMPRS) for Advanced Methods in Process and Systems Engineering, Magdeburg, Germany*

Abstract

In this study, an on-the-fly artificial neural network (ANN) framework has been developed for the tabulation of chemical reaction terms in direct numerical simulations (DNS) of premixed and igniting flames. The procedure does not require any preliminary knowledge to generate samples for ANN training; the whole training process is based on the detailed simulation results and takes place on-the-fly, so that the obtained ANN model is perfectly adapted to the specific problem considered. The framework combines direct integration (DI) and ANN model in an efficient way to overcome the extrapolation issue of the monolithic ANN model. Auto-ignition processes as well as the characteristics of established flames can be very well predicted using the ANN model. In the final simulations, involving a case with 3D turbulent hot-spot ignition, and a flame propagating in a turbulent flow, the developed procedure reduces the computational times by a factor of almost 5, while keeping the error for all species below 1% compared to the standard,

*Corresponding author.

Email address: cheng.chi@ovgu.de (Cheng Chi)

monolithic DI solution.

Keywords: Artificial Neural Network, On-the-fly, Direct Numerical Simulations, premixed combustion, ignition

1. Introduction

With the continuous advancement of computing power, direct numerical simulations (DNS) become more and more relevant for combustion modeling. It becomes increasingly possible to consider a larger domain in DNS, even for a realistic geometry (e.g., for internal combustion engines [1]) and for growing values of the Reynolds number – closer to values found in real processes. Though it is still impossible to simulate real-size configurations with full resolution of all scales in space and time using DNS, large-scale computations are now possible when accepting some additional assumptions. However, simplified chemical mechanisms are necessary for such large DNS configurations, due to the extremely large computational times required to take into account detailed mechanisms, involving hundreds of species and thousands of reactions for complex hydrocarbon fuels. Chemistry reduction approaches have then been developed [2].

The bottleneck for computing chemical kinetics is the integration of the stiff system of ordinary differential equations (ODEs). Instead of a direct integration, a tabulation approach can be a very efficient way to reduce the computational (CPU) cost related to chemical kinetics calculations. The first idea of tabulation is to compute the necessary chemical terms in advance and store all the useful quantities needed for the simulation in a discrete table covering the composition space. The later simulation relies directly on the

table through data retrieval and interpolation. The Look-Up Table (LUT) [3] is among the earliest examples of tabulation, but was only suitable for small mechanisms. Later, Maas and Pope [4] introduced a very efficient tabulation process based on intrinsic low-dimensional manifolds. This line of research has been pursued continuously since then [5–7]. As an alternative, In-Situ Adaptive Tabulation (ISAT) [8, 9] tabulates the really accessed composition space during the simulation by computing the chemical terms on-the-fly. This promising approach leads to complex challenges regarding storage space, efficient data retrieval and interpolation.

Another method to estimate the relevant reaction terms relies on Artificial Neural Networks (ANN). This method uses a non-linear regression between input and output datasets. The non-linear regression involves a combination of several non-linear basis functions and should be able to represent complex dynamics such as stiff chemical kinetics. Instead of storing large tables, as for the tabulation approaches described previously, only the coefficients (weights and biases) of the network are stored; in this manner, memory storage requirement is significantly reduced. Additionally, ANN deliver directly the output values, without any need for data retrieval and interpolation, reducing computing times.

To our knowledge, ANN was first used for chemistry tabulation in turbulent flames in [10, 11]. The ANN model was tested for a piloted H₂/CO₂ flame with simple chemistry. It was shown that ANN offers a significant reduction in memory storage and CPU time, compared to either LUT or direct integration. Later, ANN was used in [12] for CH₄ combustion with 3 networks to compute the evolution of the reactive species and one dedi-

cated network for density and temperature. Subsequently, Chen et al. [13] trained their ANN based on an already existing ISAT table, while Choi et al. [14] used ANN for ignition delay time prediction for combustion in homogeneously charged compression ignition (HCCI) engine. Besides, ANN have been used in [15, 16] to store flamelet libraries in order to save memory space, and in [17] to derive reduced kinetic mechanisms. In these previous studies, the problems were always pre-simulated to generate the training datasets, before using the ANN in the test-cases; in that way, the training of the ANN is specific to the corresponding datasets.

To obtain trained ANN models with a broader range of validity, Sen et al. [18, 19] generated the training datasets using simulations of a variety of flame/vortex interactions and stand-alone Linear Eddy Mixing calculations to obtain the thermo-chemical database representative of the relevant range of composition and turbulence. Zhou et al. [20] generated the training datasets using 1D pseudo-velocity disturbed flames at different turbulence scales. Alternatively, Chatzopoulos et al. [21] proposed an abstract problem for generating training datasets for turbulent flames, in which 100 non-premixed flamelets at varying strain rates are simulated to span the whole composition space expected in the test problems. The study documented in [22] extended the work of [21] and used it for turbulent non-premixed flames with local extinction and re-ignition. In [23], the authors used a turbulent non-adiabatic non-premixed canonical problem for the generation of the ANN database. By generating training datasets with an abstract problem, such investigations obtain trained ANN models suitable for a wider range of problems and conditions.

All the aforementioned studies rely on offline learning. The major issue associated to offline learning is the applicability of the developed ANN models to conditions outside the training regime. Even an abstract problem contains intrinsically a certain range of applicability. As an alternative, the present work develops an on-the-fly learning strategy. Instead of training the ANN using pre-simulation datasets, the learning process takes place directly during the test simulations on-the-fly, ensuring that the developed ANN model is perfectly suitable for the test problem, especially regarding data distribution. In this way, pre-simulations or associated abstract problems are not necessary anymore. The developed on-the-fly learning strategy can be in principle applied to any kind of turbulent reacting flows. In this study, the on-the-fly learning strategy is applied to direct numerical simulations (DNS). Such high-accuracy simulations involve a large number of grid points, leading to a large amount of generated data suitable as training dataset. Finally, the standard, Direct Integration (DI) model has been combined with the ANN model to predict the conditions outside the training regime during the simulation.

In Section 2, the ANN methodology and the choice of the ANN topology are introduced. Section 3 describes the detailed coupling procedure of the on-the-fly ANN process in the DNS. In Section 4, DNS cases have been setup and the on-the-fly ANN procedure has been validated, discussing key observations before concluding in Section 5.

2. ANN methodology

2.1. General idea

In high-fidelity combustion simulations, the computation of chemical reaction terms involves a system of ordinary differential equations:

$$\frac{dY_i}{dt} = f(Y_i, T, P) \quad (1)$$

where Y_i are the species mass fraction and $f(Y_i, T, P)$ can be calculated when knowing the reaction scheme and all corresponding kinetic parameters, along with temperature T and pressure P . Solving Eq. (1) is usually very expensive because of the extremely stiff chemistry and the exponential terms appearing in the reaction rates [24]. To solve this point, the basic idea in this project is to approximate the ODEs with a non-linear ANN regression, reducing considerably the computational cost. After proper training, the ANN shall be computationally very efficient and represent the dynamics of chemical kinetics.

The ANN regression is achieved through a multilayer perceptron (MLP), which involves a sum of non-linear basis functions (called activation functions) and coefficients (called biases and weights) to fit the input-output training dataset. More details concerning MLP can be found, e.g., in [25, 26]. The learning process consists in adjusting the biases and weights for each layer of the MLP to obtain a minimal error at the output layer. In the present study, the mean squared error (MSE) is evaluated at the output layer. An established back-propagation algorithm has been used to correct the biases and weights based on MSE. The conjugate gradient method has been used

to reduce the overall error, as has been done for instance in [12, 21, 22]. The back-propagation algorithm involves three steps during the learning process:

1. Forward propagation: compute the errors at the output layer from the input and initial weights and biases.
2. Backward propagation: compute the partial derivatives of the errors with respect to the weights and biases.
3. Update the weights and biases using the derivatives and learning rate.

Finally, the optimal weights and biases can be found that lead to a minimal error at the output layer for the corresponding input. These optimal weights and biases, along with the specific MLP configuration, constitute the ANN model. The ANN training framework employed in the present study is an in-house extension of the open-source package developed by Curcic and described in [27].

2.2. Data clustering

It is often not easy to obtain a single ANN model suitable for all the simulation data, especially considering that the complex dynamics controlling chemical kinetics lead to widely different behaviors. For example, the reaction zone shows very different characteristics and scales compared to the burned-gas region. In this project, the simulation data are divided into four clusters based on the local flame structure: 1) burned-gas zone, 2) reaction zone, 3) preheated zone, and 4) unburned zone (fresh gas region). The progress variable $c = Y_{\text{CO}_2} + Y_{\text{CO}}$ [28] has been used to classify these 4 zones. When dividing by the maximum value of this quantity, one obtains the normalized value \bar{c} . For the present project, $\bar{c} > 0.99$ delineates the burned-gas

zone (labeled as Cluster 1), $0.99 \geq \bar{c} > 0.2$ corresponds to the reaction zone (labeled as Cluster 2), $0.2 \geq \bar{c} > 0.001$ is the preheated zone (labeled as Cluster 3), and $\bar{c} \leq 0.001$ is the unburned zone (labeled as Cluster 4). This choice is based on trial-and-error tests during a systematic preliminary study and has been found appropriate to distinguish the different data clusters. These tests have also shown that slight changes of the threshold values did not change noticeably the final results. Since the above data clustering method has been only tested for premixed methane/air flames up to now, the clustering method and the corresponding thresholds will probably be different for other fuels, such as hydrogen or syngas, or for other burning regimes, such as MILD combustion. In principle, a suitable clustering procedure could be also identified thanks to unsupervised learning methods, such as k-means clustering [29]. The generality of the clustering method used in this project and possible improvements will have to be considered in future studies.

2.3. Configuration of the MLP

The configuration of the MLP is of course very important for an efficient and accurate ANN training process. For chemistry tabulation, 4 layers of neurons (1 input layer + 2 hidden layers + 1 output layer) have been found sufficient in past studies [17]. For this reason, 4 layers have been retained as well in the present study, as shown in Fig. 1. The first layer is the input layer, choosing as entries the species mass fractions Y_i and temperature T , so that $N_s + 1$ neurons are used in this layer, where N_s is the total number of species involved in the reaction scheme. The last layer (output layer) delivers the mass production rate of each species w_i , and involves therefore N_s neurons. These same input-output quantities have also been used e.g., in [17, 20] for

similar studies based on offline learning. The present study only considers atmospheric conditions, so that pressure is not an input parameter.

A different number of neurons have been used in the hidden layers for different clusters, depending on data complexity. In the reaction zone, topology $(N_s+1, 20, 20, N_s)$ has been used, with 20 neurons in both hidden layers. In the preheated zone and burned-gas zone, $(N_s+1, 8, 8, N_s)$ has been used, with fewer neurons in the hidden layers, because the diversity of the data is reduced. The number of neurons in the hidden layers have been chosen to be comparable to [17, 18, 22]. While it is expected that the present number of neurons should be valid in general for simple to intermediate chemical kinetics (species number $\ll 100$), it should certainly be increased for more complex chemical kinetics involving large fuel molecules, to avoid underfitting of the ANN models. For such conditions, it might even be necessary to add more hidden layers, depending on the complexity of the considered kinetics.

In the unburned zone, the output (mass production rate) is approximately 0, as there is almost no reaction there. Therefore, this data cluster is not included in the ANN procedure and is classified as inert cluster; there, the mass production rate is considered to be exactly 0 in the simulation. At the end, 3 training clusters are considered (for the reaction zone, preheated zone, and burned-gas zone).

Concerning the activation functions (AF), different solutions are possible. A sigmoid function can be used in all layers [12]; alternatively, the hyperbolic-tangent function $g(x) = (e^x - e^{-x})/(e^x + e^{-x})$ can be employed [19], or the modified hyperbolic tangent function $g(x) = \alpha \tanh(\beta x)$ with $\alpha = 1.7159$ and

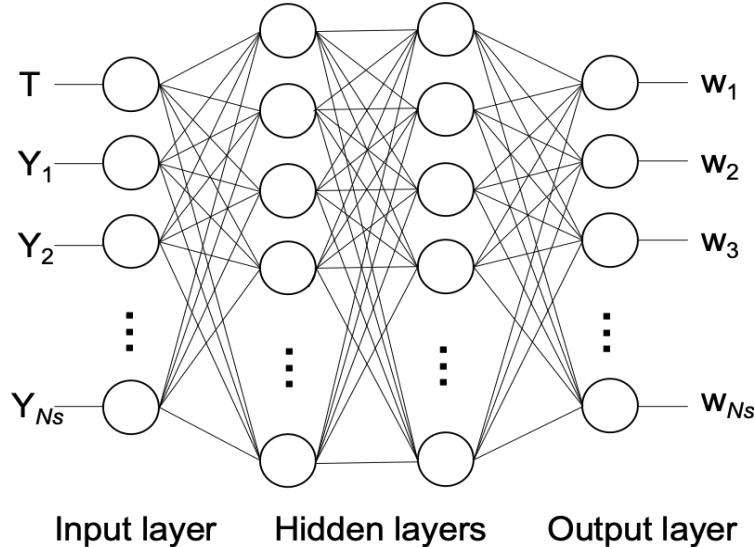


Figure 1: Topology of the MLP used for the present procedure.

$\beta = 2/3$ [22], or the rectified linear unit (ReLU) function [23]. Combinations are also possible, like a linear function in the last layer and a sigmoid function in the other layers [17]. In the present project, the performance of 3 basic activation functions (ReLU, sigmoid, and hyperbolic tangent function) has been checked in preliminary studies. Finally, the hyperbolic tangent function was found to deliver the smallest error with fewer training epochs, and has been retained for the rest of this project. Validation of the retained MLP topology and comparison of the activation functions are discussed in Section 4.1.

The learning rate of the training process decays with epochs according to [30]:

$$l_r = \frac{0.0003}{1.0 + 0.0001 \cdot n}. \quad (2)$$

The network has been trained during 400 epochs for the reaction zone (Cluster 2), 200 epochs for the preheated zone (Cluster 3), and 50 epochs for the burned zone (Cluster 1). Thanks to preliminary studies, these values have been found appropriate to reach convergence.

2.4. Input and output normalization

Proper normalization of the input and output data is essential for the success of ANN training. The data is usually normalized between 0 and 1. Since the training process is done separately in each cluster, normalization is also carried out separately in the present project. The output data are normalized as follows:

$$\bar{y} = \frac{y - y_{min}}{y_{max} - y_{min}}, \quad (3)$$

with y_{min} and y_{max} the minimal and maximum value of the corresponding quantity in the cluster. In this equation, y denotes the mass production rate of each species. Because the mass fraction distributions of minor species and radicals are highly skewed towards 0 (those minor species/radicals being generated and consumed very rapidly during the reaction process), the low-level contributions (compared to the corresponding peak value) are typically underestimated by standard normalization for minor radicals. However, these contributions are often very important, for instance to describe accurately ignition processes. Therefore, the input data for those minor radicals/species are normalized using a logarithmic scale to get a better balance of the stiff

distribution, as follows:

$$\bar{x} = \frac{\log(x + x_{max}) - \log(x_{min} + x_{max})}{\log(2 \cdot x_{max}) - \log(x_{min} + x_{max})}, \quad (4)$$

with x_{min} and x_{max} the minimal and maximum value in the cluster. In this equation, x represents the mass fraction of minor species/radicals. In the present study, all species with mole fraction lower than 1% during the whole simulation in time and space have been classified as “minor”. The input data for temperature and major species such as O₂, N₂, CO, CO₂, H₂O and CH₄ are normalized using the standard method in Eq. (3). To avoid divide-by-zero issue in the normalization process, the non-reacting trace species with $x_{max} - x_{min} < 1 \times 10^{-12}$ are skipped from normalization and the values are reset to zero. This treatment is physically meaningful for the ANN network as the contribution of such species to the reaction kinetics is negligible. Note that the minimum and maximum values are obtained separately in each cluster and are updated just before ANN training. After completion of training, these values are stored and used for normalization in the DNS iterations using the ANN models. As shown in Section 4.1, the normalization obtained with Eq. (4) leads to a more accurate ANN model for the region where radical species mass fractions are relatively low.

3. Coupling ANN on-the-fly in the DNS simulations

3.1. Direct numerical simulations

DNS is currently the most accurate numerical method for solving turbulent flows with chemical reactions, providing under ideal conditions an

exact solution for both fluid dynamics and flame structures. In the present study, the in-house low-Mach DNS solver *DINO* [31] has been used. The Navier-Stokes equations for multi-component species with detailed physicochemical models have been solved. The spatial derivatives are computed using a sixth-order centered explicit scheme. An implicit Williamson third-order Runge-Kutta time integrator with analytic Jacobian inversion (PyJAC [32]) is employed for temporal integration. *DINO* has been extensively used in numerous turbulent combustion studies, e.g., [33–35], in which the proper coupling between chemical reactions, turbulent transport, and heat exchange has been fully validated. Cantera [36] has been systematically used in *DINO* to compute all terms regarding chemical kinetics, thermodynamic, and transport coefficients. Premixed methane flames are considered, using the established detailed kinetic mechanism GRI-Mech 3.0 [37]. All simulations in the current study have been performed on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC, Germany).

3.2. Coupling procedure

ANN learning is embedded inside the time integration in DNS. The learning process involves 1) sample data generation; 2) ANN training, and finally 3) ANN network usage. Sample data are obtained directly from the instantaneous DNS results involving the full kinetic scheme as computed by direct integration using Cantera. As explained in Section 2.3, the results at all grid points in the computational domain are divided into 3 training clusters and 1 inert cluster. In each training cluster, a randomly-selected 10% subset of the results are taken as cross-validation dataset, while the remaining 90% subset are used as training dataset. The training process is done separately

for the 3 training clusters, resulting in 3 ANN models. The obtained ANN models are stored and used as initial ANN model for the training process at the next training iteration.

The stored ANN models are used to replace the expensive ODE integrations in the later simulation steps. Note that the ANN models do not deliver the species and temperature increments, but only the mass production rate of each species. Hence, species and temperature transport equations still need to be computed in order to take into account convection and diffusion through the Runge-Kutta procedure implemented in the DNS code. It is well known that ANN models cannot generally predict the conditions out of their training regime. To deal with this problem, direct integration is used again at the locations where conditions are found to lay outside of the training regime, as explained in the next Section 3.3.

After some given physical time, the ANN models have to be trained and updated again, to avoid that the previous ANN models stay too far away from the current dataset conditions. Updating the ANN models with time is of central importance for the on-the-fly ANN procedure, ensuring that these ANN models deliver permanently an accurate prediction for time-evolving simulations. As the training process is expensive, the frequency of the ANN updated-training determines to a large extent how expensive the whole simulation will be. Training and updating the ANN models too frequently obviously enhances the accuracy, but increases strongly the computational cost. Therefore, choosing a proper interval in physical time between two ANN trainings is critical for the balance between accuracy and CPU cost. The

total cost function for this balance is expressed as

$$G(m) = \bar{c}(m)^\lambda \bar{e}(m)^{1-\lambda}, \quad (0 \leq \lambda \leq 1), \quad (5)$$

where m is the interval in terms of iteration steps, $c(m)$ is the average CPU cost per iteration in the total simulation with ANN and $\bar{c}(m)$ is the same value normalized by the average CPU cost per iteration in the simulation without ANN. Thus, $\bar{c}(m) < 1$ indicates the ANN computation is cheaper than the direct integration by Cantera. The quantity $e(m)$ is the maximum mean squared root of the error during the simulation, which is computed as

$$e(m) = \max_t \sqrt{\frac{\sum_{i=1}^{N_p} (z_i - \hat{z}_i)^2}{N_p}}, \quad (6)$$

where z_i and \hat{z}_i are the OH mass fraction in direct integration and in ANN computation, respectively. The radical OH has been chosen due to its importance regarding ignition and combustion of most relevant fuels. In this equation, N_p is the total number of grid points in the simulation. The value $\bar{e}(m)$ is normalized by the maximum spatial OH mass fraction in the simulation without ANN and multiplied by 1000, so that $\bar{e}(m) > 1$ indicates a mean-squared root error for OH larger than 0.1%. The choice of another species as an indicator would be possible as well. The user-defined parameter λ in Eq. (5) is used to quantify the trade-off between CPU cost and accuracy. In the present study, $\lambda = 0.7$ has been employed to focus slightly more on CPU cost, while keeping a good accuracy. The cost function is evaluated through several preliminary small-scale test simulations to get a proper interval m before starting the full DNS simulation. More details can be found

in Section 4.

3.3. Regions identification for direct integration

To precisely identify the local regions where the current datasets are out of the training regime, the input data dimension ($N_s + 1$, corresponding to all species plus temperature for this atmospheric application) has to be reduced. The idea underlying intrinsic low-dimensional and flamelet-generated manifolds (ILDM, FGM, FPI [4–6, 38]) has been used to reduce the input data dimension from N_s+1 to only 2, based on 1) the progress variable c , and 2) the mixture fraction Z . For this purpose, the progress variable $c = Y_{\text{CO}} + Y_{\text{CO}_2}$ has been used. The mixture fraction Z has been considered as the normalized mass fraction of nitrogen N_2 , since this species is a purely inert gas in the case considered. Using this approach, a 2D probability density function (PDF) map for the input data distribution is obtained. A higher density indicates the local regions where more samples are involved in the training process, while low-density regions correspond to situations where a poor performance of the obtained ANN models is expected. These locations are labeled as “Cluster 5”. In such regions, instead of using the ANN estimation, the DI model is activated to get a more accurate solution.

To avoid insufficient training samples during the training process, the total number of training data in each training cluster is checked continuously. If the number of training datasets is not enough, the corresponding cluster is also classified as DI region, and the ANN estimate is not used. The corresponding threshold of the training samples for each cluster is determined to be 10 times the number of degrees of freedom for the network [39]. This means that the threshold is set to $N_2 = 10 \cdot ((N_s + 1) \cdot 20 + 20 \cdot 20 + 20 \cdot N_s) =$

$200 \cdot (2 \cdot N_s + 21)$ for Cluster 2, and $N_1 = 10 \cdot ((N_s + 1) \cdot 8 + 8 \cdot 8 + 8 \cdot N_s) = 80 \cdot (2 \cdot N_s + 9)$ for Cluster 1 and 3.

In summary, the pseudocode of the ANN learning procedure is as follows:

Algorithm 1: On-the-fly ANN learning procedure

```

Result:  $w_i = f(T, Y_1, Y_2, \dots, Y_{N_s})$  where  $i = 1, \dots, N_s$ 
if  $Iteration \geq Iteration\_start$  then
    if  $(Iteration - Iteration\_start)/Interval$  is integer then
         $w_i$  is computed by direct integration;
        Cluster sample dataset based on progress variable  $c$ ;
        Normalization of the dataset in each training cluster;
        Randomly select 90% of the dataset as the training set, and the other
        10% as the cross-validation set;
        Obtain the 2D PDF map  $p = f(c, Z)$ , and store the map;
        if  $Iteration == Iteration\_start$  then
            | Randomly initialize the ANN weights and biases between 0 and 1;
        else
            | Initialize the ANN weights and biases with the previously stored
            | ANN models;
        end
        Train and store the ANN models (weights and biases);
    else
        Cluster the input dataset;
        Normalization of dataset in each cluster based on the maximum and
        minimum values obtained in the previous training iteration step;
        Read the PDF map and identify the locations where direct integration is
        needed;
        Read the ANN models where accuracy is sufficient;
        Finally compute  $w_i = \delta \cdot$ 
         $\text{ANN}(T, Y_1, Y_2, \dots, Y_{N_s}) + (1 - \delta) \cdot \text{DI}(T, Y_1, Y_2, \dots, Y_{N_s})$  where  $\delta = 0$ 
        in the regions out of the training regime (DI required), otherwise  $\delta = 1$ ;
    end
else
    |  $w_i$  is computed by direct integration;
end

```

In the above procedure, $Iteration_start$ is the iteration step when the ANN learning starts, while $Interval$ denotes the user-chosen duration be-

tween two successive ANN trainings. All ANN models are updated on-the-fly during the simulation.

3.4. Parallelization

Parallelization of the DNS code *DINO* relies on the 2DECOMP&FFT library [40]. This library is called on top of the standard MPI and MPI-I/O libraries. The very good parallel efficiency of *DINO* on supercomputers has been discussed in [31]. In principle, the computational domain is decomposed into several subdomains and each subdomain is handled by one single CPU processor (domain-decomposition).

The implemented parallelization of the ANN learning process is fully compatible with this domain-decomposition technique. Looking at the back-propagation algorithm, both the forward and backward propagation process can be done on each subdomain using one single processor. The derivatives of the errors corresponding to each parameter (weights and biases) are then summed up for all the samples in the subdomain. In this manner, each processor has its own error derivative matrix. Then, the derivative matrixes are gathered from all processors onto the master processor. On the master, the parameters (ANN models) are updated and broadcasted back to all processors for the next training epoch. Since this update process involves only little communication and is very fast compared to forward and backward propagation, the parallel efficiency of the training process is high.

4. Validation and application

4.1. Turbulent premixed flame

The first validation case is a two-dimensional turbulent premixed flame, starting from a planar geometry. The computational domain is $3\text{ mm} \times 6\text{ mm}$, discretized by 256×512 grid points. The initial temperature is 300 K and the initial pressure is 1 bar. The flame front is initially placed at the center of the box, with burned gas and unburned gas corresponding to the same premixed methane-air mixture (equivalence ratio $\Phi = 1$) at the bottom and top side, respectively. The initial step in temperature and mass fractions in the middle of the box is smoothed out using a hyperbolic tangent function, $\psi(y) = 0.5(1 + \tanh(-\epsilon \cdot (y - 0.5 \cdot L_y)))$, where L_y is the total domain length in y direction and $\epsilon = 80,000\text{ 1/m}$ is the stiffness parameter. The bottom and top boundary conditions are a subsonic inlet and a subsonic outlet, respectively; the numerical box being open, the thermodynamic pressure stays equal to its initial value (atmospheric pressure). The left and right boundaries are periodic. A homogeneous isotropic turbulent field, which is generated based on an Inverse Fast Fourier Transform (IFFT) with an analytically prescribed turbulence spectrum following von Kármán with Pao correction (VKP spectrum), is superimposed in the domain at $t = 0$. The initial turbulent field has turbulence intensity $u'/s_L = 39.31$ and integral-scale Reynolds number $\text{Re}_t = u' l_t / \nu = 32.2$, where u' , s_L , l_t and ν are the root-mean-square (rms) of velocity fluctuations, laminar flame speed, integral length scale and mixture viscosity respectively. The Kolmogorov length scale is $\eta = 19.4\text{ }\mu\text{m}$, which is fully resolved by the fine grid employed (resolution of $11.7\text{ }\mu\text{m}$).

To obtain a suitable interval for the ANN training in this simulation, the cost function in Eq. (5) has been evaluated at different m (20, 50, 100, 200, 400 and 800). The test simulations have been run for 900 iterations and the ANN starts from iteration 100. Figure 2 shows the function curve of the cost function and the normalized CPU cost with the interval steps. It is seen that the cost function reaches minimum values for intervals m of 100 and 200. When the interval is smaller, the CPU cost increases rapidly. For this reason, the interval m has been set to 200 for all the rest of this study. The ANN procedure starts from iteration 600 (*Iteration_start* in the previous algorithm), at which time the turbulent flame structure is well established.

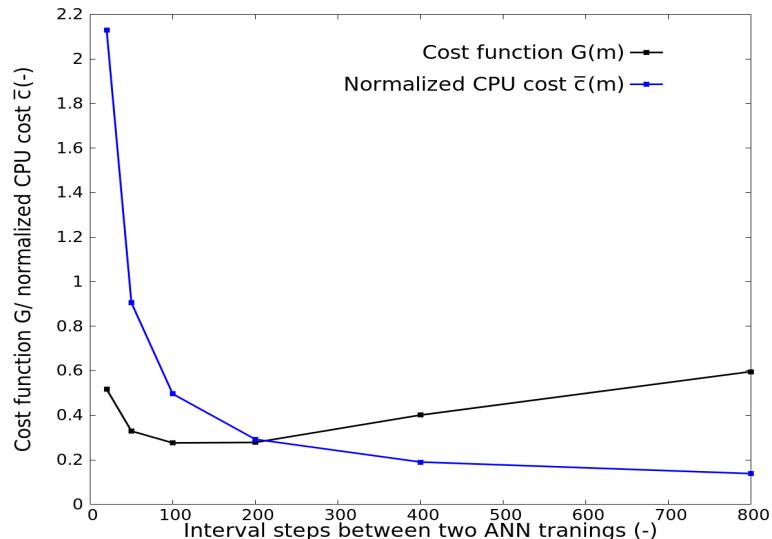


Figure 2: Cost function $G(m)$ and normalized CPU cost $\bar{c}(m)$ as function of the interval m in terms of iteration steps, as obtained for the test simulation involving a two-dimensional turbulent premixed flame.

4.1.1. Impact of MLP topology and activation function

The influence of the MLP topology has first been checked. Figure 3 shows the MSE of the training as function of the training epoch at iteration 800 for the reaction zone (Cluster 2) using different MLP configurations. The finally retained configuration (black line in Fig. 3, mostly hidden behind the blue line at the bottom of the figure) has topology $(N_s+1, 20, 20, N_s)$ in the reaction zone and relies on the tangent function as AF, as already mentioned in Section 2.3. For comparison, similar topologies with less neurons $(N_s+1, 8, 8, N_s)$ and more neurons $(N_s+1, 40, 40, N_s)$ in the hidden layers have been checked as well. As can be seen in Fig. 3, reducing the number of neurons leads to a noticeable increase of the error, while increasing further this number does not lead to a noticeable improvement. As a consequence, the topology $(N_s+1, 20, 20, N_s)$ has been kept for all further steps. Concerning the choice of the activation function, the finally retained hyperbolic-tangent function gives a much lower error compared to the sigmoid function (green line in Fig. 3) and ReLU function (pink line in Fig. 3).

As a further comparison, Figure 4 shows the instantaneous distribution of heat release for the same condition at the same time using Direct Integration (without ANN, top), with ANN and tangent function as AF (center), and with ANN and sigmoid function as AF (bottom). It is visually obvious that the simulation result with ANN and tangent function as AF (the finally retained configuration) is closer to the reference case (detailed chemistry DI computation without ANN).

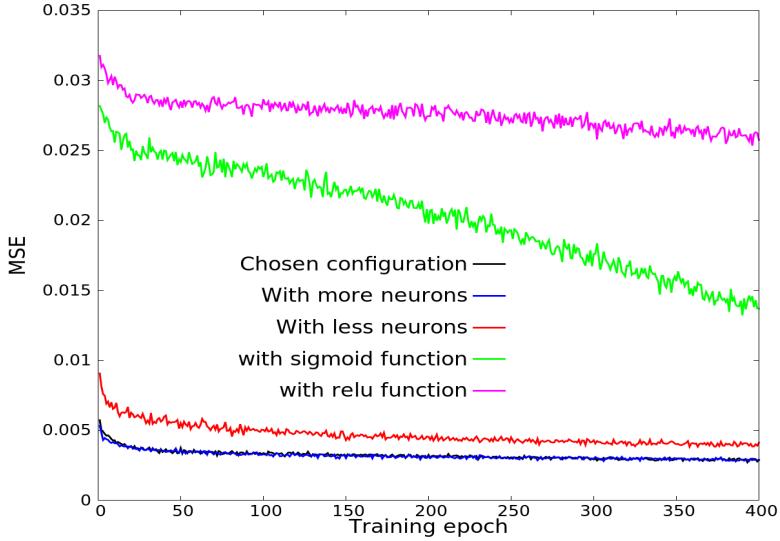


Figure 3: MSE as function of training epoch for different MLP configurations at iteration 800 for the reaction zone (Cluster 2) concerning a turbulent, initially planar two-dimensional flame.

4.1.2. Impact of normalization

Figure 5 shows the absolute errors regarding the mass fractions of species O, OH, H, CO and HO₂, as computed by the finally retained ANN compared to the direct integration using Cantera with the detailed GRI-3.0 mechanism. This error is computed as $e = |Y_{DI} - Y_{ANN}|$ with Y_{DI} and Y_{ANN} the normalized mass fractions computed by Cantera (DI) and ANN, respectively. All the results are shown at iteration 800. A log-scale normalization has been used for the input data (radical species) concerning the ANN in Fig. 5a, while a linear normalization has been used in Fig. 5b. By comparison, it can be seen that ANN with log-scale normalization is slightly more accurate, in particular for species with low mass fractions. However, the difference is small, and a linear normalization would be possible as well.

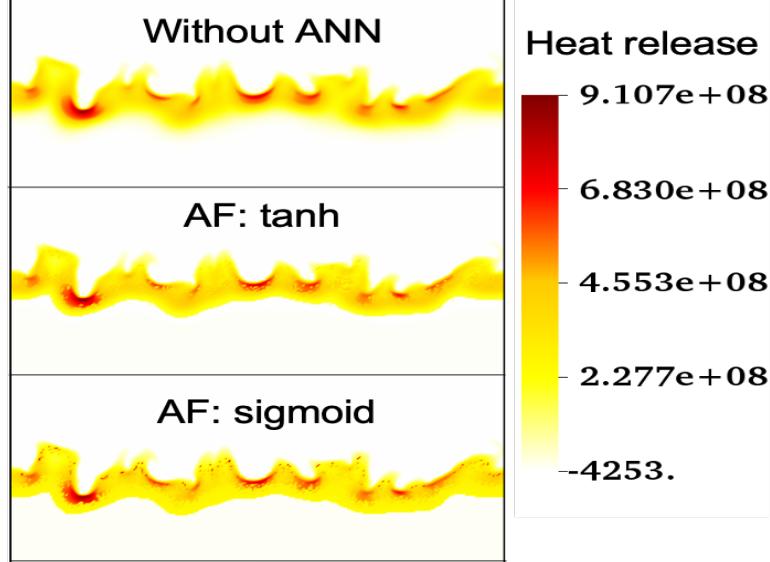


Figure 4: Comparison of the instantaneous distribution of heat release for the same condition at the same time for the turbulent, initially planar two-dimensional flame simulation with Direct Integration (without ANN, top), with ANN and hyperbolic tangent function as AF (center), and with ANN and sigmoid function as AF (bottom). The color scale is kept identical for all three subfigures.

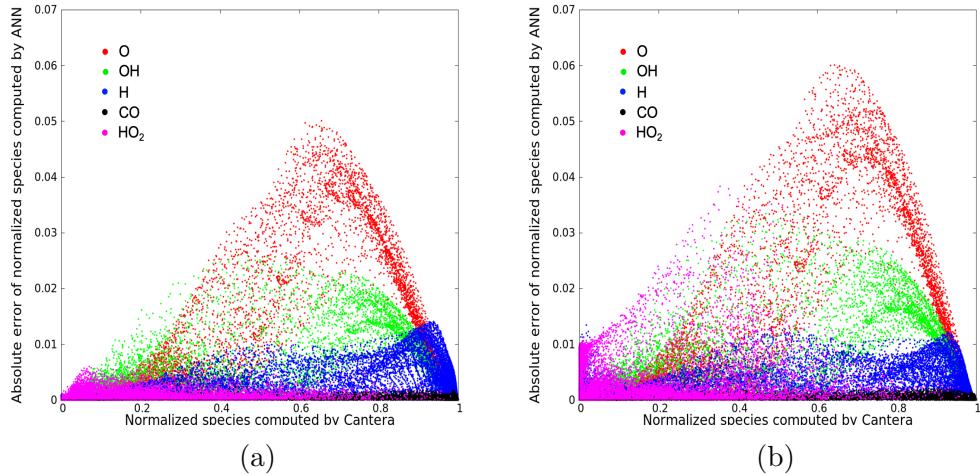


Figure 5: Plot of the absolute errors regarding the mass fractions of species O, OH, H, CO and HO_2 , as computed by the finally retained ANN compared to the direct integration using Cantera with the detailed GRI-3.0 mechanism at iteration 800 for (a) ANN radicals input with log-scale normalization, and (b) ANN input with linear normalization.

4.1.3. On-the-fly procedure

Configurations involving turbulent reacting flows are often associated with drastic changes in topology and flame regimes in space and/or time. Therefore, getting a good agreement at some specific time instant is not sufficient to ensure high accuracy at all times. This is the reason why the training procedure has been implemented on-the-fly and must be repeated when the conditions differ too much from the previous training set(s). To check this point, the accuracy of the implemented ANN prediction has also been checked at a later time, after re-training and updating twice the ANN models.

Figure 6a shows again the absolute errors regarding the mass fractions of five important species as computed by ANN compared to DI using Cantera and GRI-3.0 mechanism at a later time, iteration 1300. Log-scale normalization has been used for minor radical species. Compared to Fig. 5a, the points are even closer to the zero-error line, proving that the on-the-fly ANN model has been properly updated and the accuracy does not decrease at all with time. The accuracy can also be checked by comparing the instantaneous fields of heat release (Fig. 6b). The prediction obtained by the implemented combination of ANN and Direct Integration is visually identical to the reference solution obtained by pure DI at this later time.

4.1.4. Coupling procedure between ANN and DI

As already mentioned in Section 3.3, the identification of regions where DI is required for a high accuracy is based on the PDF map representing the distribution of the input dataset. Figure 7 shows this PDF map in the two relevant coordinates (reaction coordinate based on the progress variable,

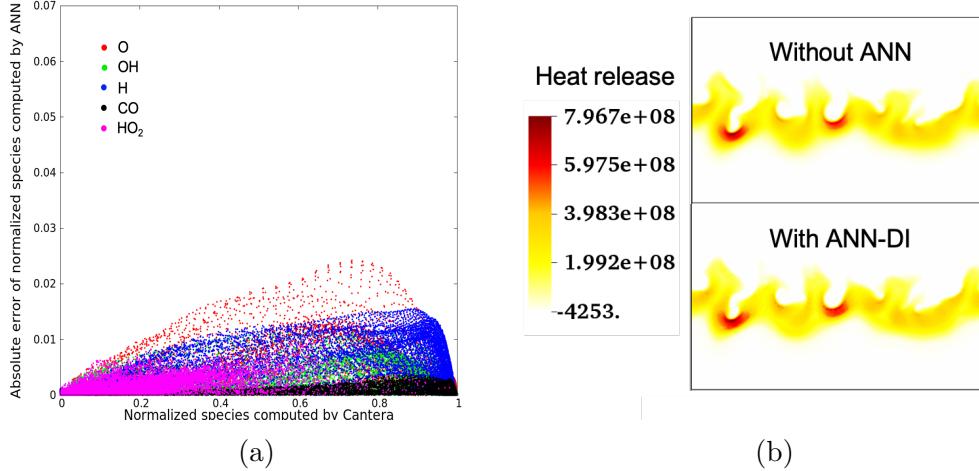


Figure 6: (a) Plot of the absolute errors regarding the mass fractions of five important species as computed by ANN compared to DI using Cantera and GRI-3.0 mechanism at a later time, iteration 1300; (b) Instantaneous heat release field at this same time obtained with Direct Integration (labeled “without ANN”, top) or with the implemented ANN procedure combined with DI in regions of insufficient accuracy.

vertical scale; mixture fraction based on normalized nitrogen mass fraction, horizontal scale) for the considered turbulent premixed flame, exemplarily at iteration 800. The graphical domain $[0; 1] \times [0; 1]$ plotted in Fig. 7 has been discretized into 100×100 equi-sized cells. All cells containing less than 3 samples are set as DI regions, while in all cells with 3 or more samples, the ANN model is activated. This is represented with red curves in Fig. 7, delineating the outer boundary of the region in which the ANN prediction is employed; outside, in the regions with lower point density, DI is used to ensure high accuracy. The threshold value of 3 has been retained after preliminary tests; a smaller threshold value induces a noticeable loss of accuracy, while a larger threshold leads to a strong reduction of the obtained speed-up.

As visible in Fig. 7, most points are predicted by ANN. Among all the points in the DNS grid, 93% are in the ANN region and only 7% are in the

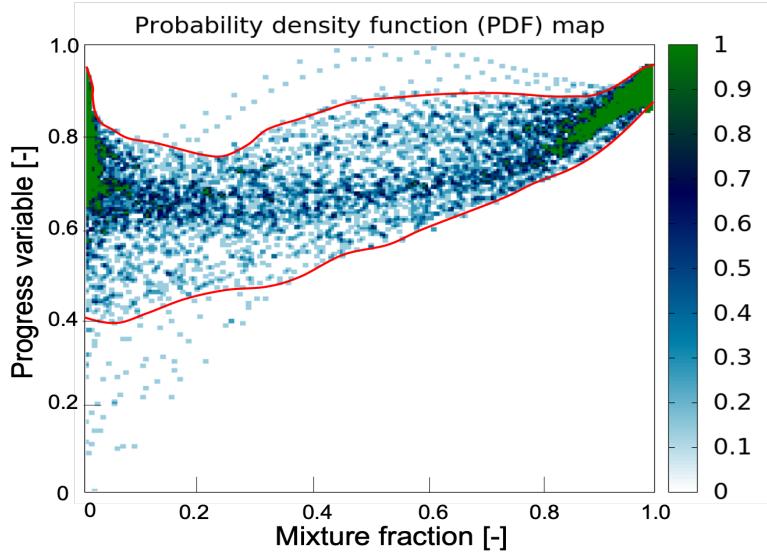


Figure 7: PDF map representing the distribution of the input dataset, exemplarily at iteration 800.

DI region at iteration 800. Still, combining the ANN prediction with DI in sparsely-populated regions is essential to ensure accuracy. This is demonstrated in Figure 8, which shows the instantaneous heat release field for the turbulent planar flame at iteration 810 obtained with monolithic ANN (i.e., without DI, top), and with the implemented ANN-DI combination (bottom). It can be clearly seen that using only the ANN prediction in the whole domain leads to a very large error, in particular in the (most important) high heat release region. To document further this point, the distribution of the different clusters is shown as well in the middle subfigure. After identification of the DI region (Cluster 5) following the procedure described in Fig. 7 and using here Direct Integration, the whole flame structure is extremely well reconstructed, as shown in the bottom subfigure of Fig. 8.

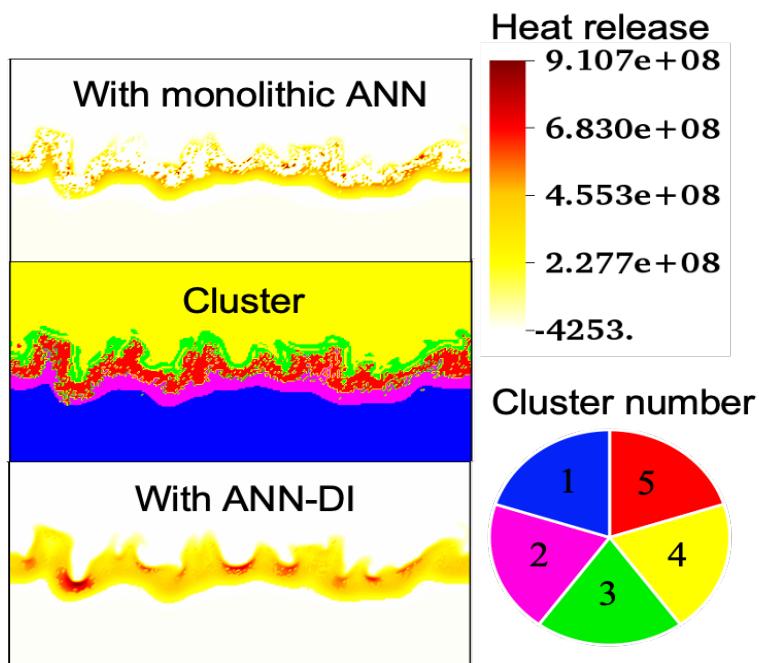


Figure 8: Instantaneous field of heat release distribution for the turbulent two-dimensional flame simulation with monolithic ANN (without DI, top), and with the implemented ANN-DI combination (bottom). In between, the distribution of the different clusters is shown (middle subfigure). As a reminder, Cluster #1, 2, 3, 4 represent burned zone, reaction zone, preheated zone, and unburned zone respectively; while Cluster 5 represents the regions associated to DI. All these results correspond to iteration 810.

4.1.5. Computational cost

It is now important to quantify the computational cost of the on-the-fly ANN procedure. Additionally, the whole simulation has been carried out using 16, 32, and 64 CPU cores to estimate roughly the parallel efficiency of the procedure. Table 1 shows the average wall-clock-time for a single iteration step. As it can be seen, the computational cost of the ANN training takes as expected orders of magnitude more time than computing the convection-diffusion-reaction terms. However, ANN training only takes place at certain iterations. And computing directly the reaction terms by DI (without ANN) is far more expensive than computing convection-diffusion terms (roughly by a factor 4 for this particular configuration). Due to this competition, the overall gain is not immediately clear.

Wall-clock-time (ms)	16 CPU cores	32 CPU cores	64 CPU cores
Convection-diffusion term	176	90	47
Reaction term without ANN (pure DI)	806	398	183
Reaction term with monolithic ANN	54	26	10
Reaction term with ANN-DI	75	37	16
One ANN training step	23,753	12,987	7,311

Table 1: Average wall-clock-time for a single iteration step, and cost of ANN training

For the above computations, the ANN model has been updated every 200 iterations, as stated at the beginning of Section 4.1. Thus, the total time for the ANN-DI reaction procedure over one training cycle (from one training iteration to the next one) using 64 CPU cores is 10678 ($= 183$ [generating data with DI] + 7311 [training] + $(200 - 1) \times 16$ [computation with ANN-DI]) ms, while for reaction without ANN the to-

tal time for the same duration in terms of physical time is 36600 ($= 200 \times 183$) ms. Therefore, it can be concluded that the developed on-the-fly ANN procedure is almost 3.5 times faster. Without any particular optimization, the parallel efficiency of the ANN training process reaches 91% for 32 cores and 81% for 64 cores, using the 16-core configuration as a reference. The disk storage space required for a single ANN model is only 64 kB; this is extremely small compared to classical look-up tables needed for ILDM, FGM or ISAT. In principle, the obtained ANN models can be also re-used for another simulation with a similar configuration, avoiding the expensive training process. In that case, the corresponding computational cost using 64 CPU during 200 iterations is 3200 ($= 200 \times 16$) ms, which is more than 10 times faster compared to a direct integration of the reaction terms. To illustrate this point, Figure 9 shows the simulation results of a another realization of the same planar premixed flame considered previously, meaning that another initial turbulent random field has been implemented, all other conditions being identical. The top figure shows the instantaneous distribution of heat release from DI (reference), while the bottom figure is obtained with the ANN models that have been saved from the previous DNS computations, both at iteration 900. It is seen that a direct re-use of the previously obtained ANN models in the new simulation also leads to a very good prediction for this independent realization, with a speed-up factor exceeding 10.

4.2. Turbulent hot-spot ignition

It is now essential to check if the developed procedure could be applied in general to other configurations of interest. Since our group is particularly interested in ignition (e.g., [33]), the ignition of a premixed system by a hot-

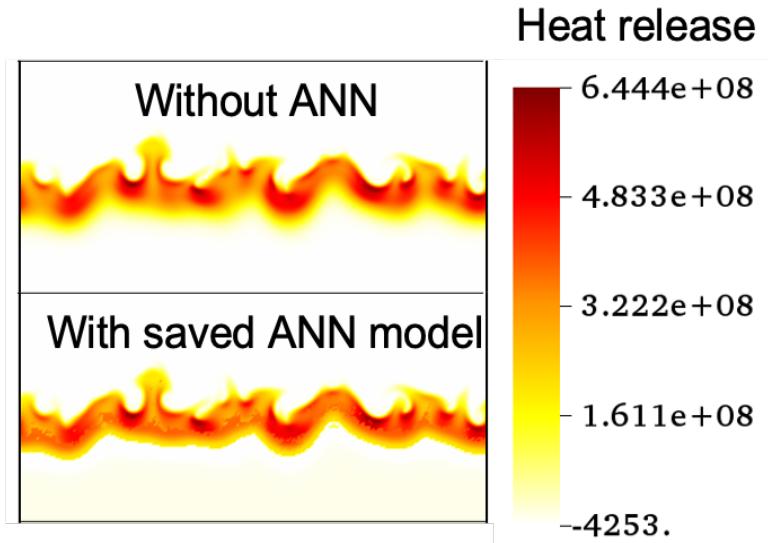


Figure 9: Comparison of the instantaneous distribution of heat release for the same condition at the same time (iteration 900) for another independent realization of the turbulent, initially planar two-dimensional flame simulation with Direct Integration (without ANN, top: reference) and using the previously saved ANN model (bottom).

spot in a turbulent environment has been considered. The configuration is similar to that considered extensively in [33]. The 2D computational domain has a size $8 \text{ mm} \times 8 \text{ mm}$ and is discretized by 1024×1024 grid points. The hot-spot is initially located at the center of the domain, with a radius of 1 mm and an initial temperature of 2594 K (high enough to successfully ignite the surrounding cold mixtures). The surrounding temperature is 300 K and the homogeneous pre-mixture is methane-air with equivalence ratio $\Phi = 1$. The initial step in temperature profile is again approximated by a hyperbolic tangent function [33]. The initial pressure is 1 bar. A homogeneous isotropic turbulent field is initially superimposed within the whole domain. The initial turbulent field has turbulence intensity $u'/s_L = 62.3$ and integral-scale Reynolds number $\text{Re}_t = 273.3$. The corresponding Kolmogorov length

scale is $\eta = 8.37 \mu\text{m}$, which is fully resolved by the fine grid ($7.81 \mu\text{m}$). All boundary conditions are periodic. The simulation has been computed using 256 CPU cores on the supercomputer JUWELS at Jülich Supercomputing Center. The on-the-fly ANN procedure starts from the very beginning and the interval between two ANN trainings is again set to 200, as previously.

Figure 10 shows the time evolution of the maximum temperature and peak OH mass fraction in the whole domain, comparing results for the implemented ANN-DI procedure, monolithic ANN, and reference solution with pure DI (i.e., without ANN). As can be seen, the ignition delay and the evolution of the relevant quantities in the domain are perfectly predicted by the ANN-DI procedure. On the other hand, the monolithic ANN cannot predict properly the fast increase of temperature and OH mass fraction after about $2 \mu\text{s}$. This confirms the need for a combination between ANN and DI in order to correctly predict such stiff and highly non-linear processes.

The instantaneous OH mass fraction and temperature fields at 3 different time instants ($t = 4.88, 5.35$ and $6.22 \mu\text{s}$ from left to right) are compared for the reference case (pure DI, without ANN: top) and using the developed ANN-DI procedure (bottom) in Fig. 11. The agreement is always excellent.

The CPU cost of the simulation without ANN for 1000 iteration steps is 2220 s using 256 CPU cores, while the simulation with ANN-DI for 1000 iteration steps costs 711 s, which means a speed-up factor of about 3.1.

4.3. Three-dimensional DNS involving ignition and flame propagation

Obviously, turbulence is intrinsically three-dimensional. In this last test-case, a 3D configuration is considered, involving both ignition (at the beginning) and propagation of a turbulent premixed flame (at a later stage).

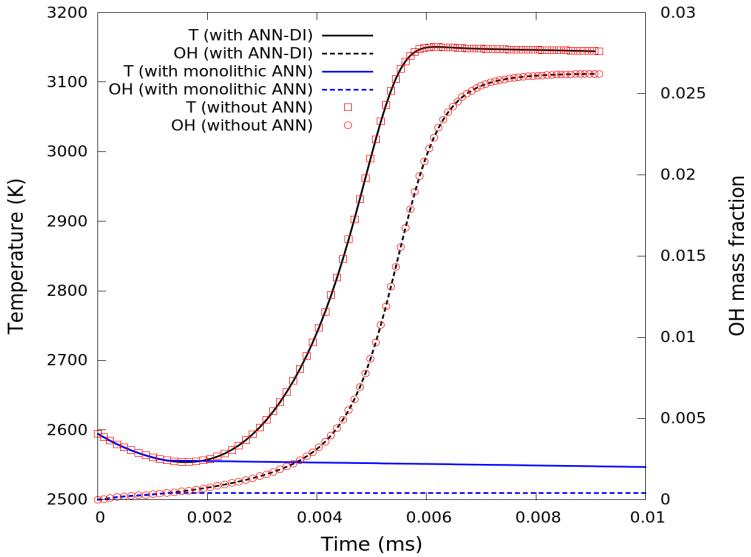


Figure 10: Time evolution of the maximum temperature and peak OH mass fraction in the numerical domain for hot-spot induced ignition in a turbulent flow, using our ANN-DI procedure (black curves), monolithic ANN (blue curves), or pure DI (without ANN, symbols).

In this configuration, all aspects can be checked simultaneously. Similar to the previous configuration, the 3D domain has a size $8 \text{ mm} \times 8 \text{ mm} \times 8 \text{ mm}$, discretized by $1024 \times 1024 \times 1024$ grid points. The initial turbulent field has turbulence intensity $u'/s_L = 72.6$ and an integral-scale Reynolds number $\text{Re}_t = 422.1$. All other conditions are the same as in the previous 2D case. The Kolmogorov scale in this case is $8.04 \mu\text{m}$, which is well resolved by the grid resolution ($7.81 \mu\text{m}$). The simulation has been carried out using 8192 CPU cores on JUWELS. The on-the-fly ANN procedure starts from the beginning, and the interval between two sequential ANN trainings has been slightly increased from 200 to 300, in order to reduce slightly the computational requirements for this quite expensive computation.

The ignition delay time $\tau = 5.1 \mu\text{s}$ has been calculated based on the maxi-

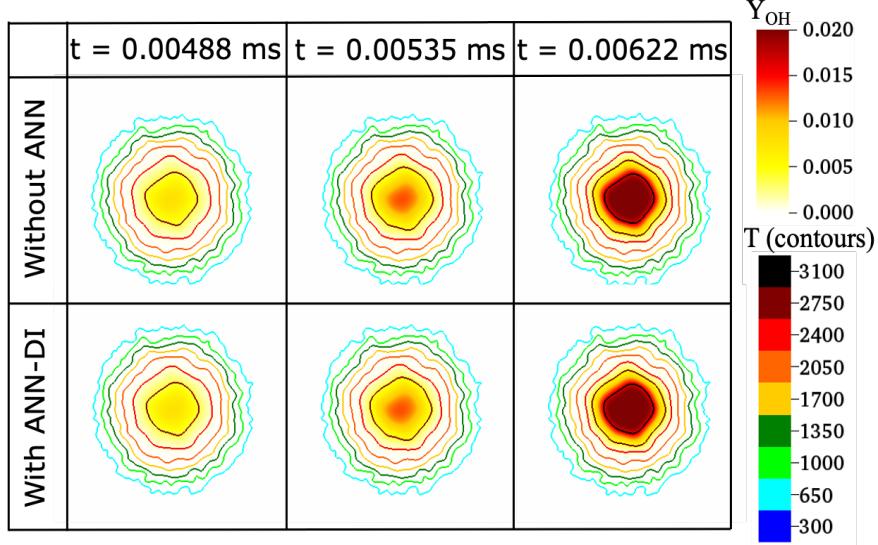


Figure 11: Instantaneous OH mass fraction and temperature fields at 3 different time instants ($t = 4.88, 5.35$ and $6.22 \mu\text{s}$ from left to right) during hot-spot induced ignition. Top: pure DI (reference, without ANN); Bottom: with the developed ANN-DI procedure. The same color scale is used in both cases.

mum gradient of peak OH concentration: exactly the same value is obtained with the reference simulation (DI, without ANN) and with the developed ANN-DI procedure. Figure 12 shows the spatial distribution of temperature and of selected mass fractions along a line extending from the center of the hot-spot toward the cold mixture outside in positive x -direction (arbitrary choice) at four successive times: $t = 4.38, 5.37, 8.85$ and $17.6 \mu\text{s}$. In the figure, the lines denote the reference solutions (pure DI, no ANN), while the circles correspond to the results obtained with the ANN-DI procedure presented in this article. The two figures on the top (Fig. 12a and 12b) describe mainly the initial ignition process, while Figs. 12c and 12d correspond to the establishment of the turbulent premixed flame. As is seen, the agreement is excellent. For all plotted mass fractions, the error is below 1% at all times.

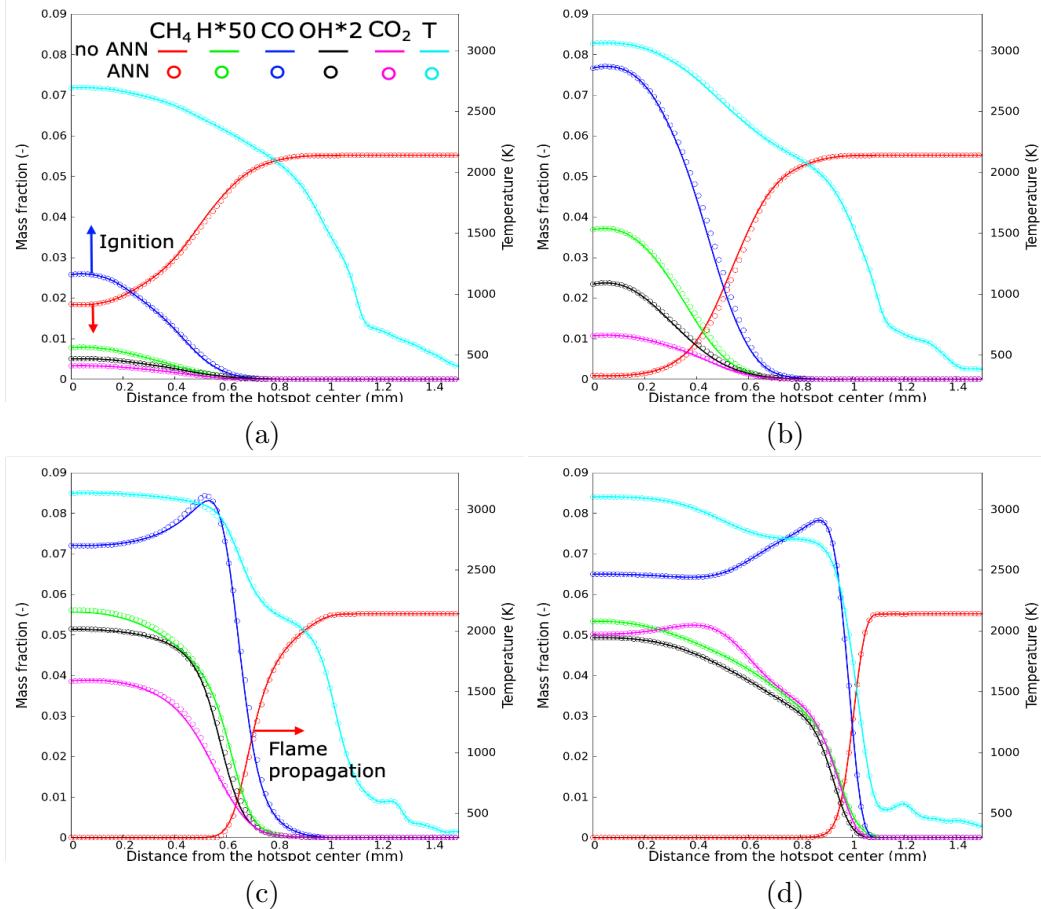


Figure 12: Distribution of temperature (light blue) and of selected species mass fractions along one line extending from the hot-spot center to the cold mixture outside in positive x -direction during turbulent hot-spot ignition at the successive times: $t = 4.38, 5.37, 8.85$ and $17.6 \mu\text{s}$. In the figure, the lines denote the reference solutions (pure DI, no ANN), while the circles correspond to the results obtained with the ANN-DI procedure presented in this article. The two figures on the top describe mainly the initial ignition process, while the two bottom figures correspond to the establishment of the turbulent premixed flame.

Figure 13 compares the instantaneous temperature contours for the reference simulation (left: pure DI, without ANN) and with the ANN-DI procedure (right) at $t = 9.3 \mu\text{s}$. The structure of the reaction front propagating toward the outside is clearly seen. The agreement between both results is excellent.

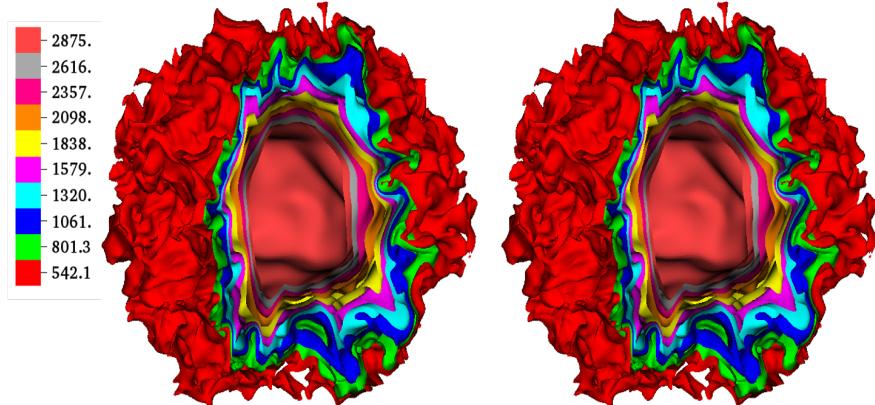


Figure 13: Instantaneous temperature contours for the propagating reaction front (left) without ANN (reference: pure DI), and (right) with the ANN-DI procedure at $t = 9.3 \mu\text{s}$.

In terms of computational costs, the reference simulation (no ANN, pure DI) takes 0.44 million CPU hours to compute 10 μs of physical time. For the same duration, the simulation relying on ANN-DI requires only 0.091 million CPU hours, including on-the-fly training. This means that the developed ANN-procedure leads to a speed-up by a factor of about 4.8 with negligible loss of accuracy.

5. Concluding remarks

An on-the-fly ANN procedure has been developed and successfully implemented within our DNS code. In contrast to previous ANN studies, the

employed ANN model is updated on-the-fly during the DNS simulation based on the training dataset obtained by Direct Integration during the simulation itself. Therefore, no abstract problem is needed and the obtained ANN model is perfectly fitted to the configuration considered. Issues connected to the extrapolation of a monolithic ANN model to conditions outside of the training regime have been solved by 1) identifying those regions, and 2) using Direct Integration there. The training dataset has been divided into 4 clusters, in order to increase the training and prediction accuracy in each cluster. The procedure parallelizes well on multiple CPUs, though no effort has been spent on this aspect yet. The developed approach has been tested for different configurations involving turbulent flames: 2D premixed flames, 2D hot-spot ignition, 3D ignition and propagating flame. These tests show that the ANN-DI framework reduces the computational cost by a factor between 3.1 and 5 (taking into account the training cost) with no noticeable loss in accuracy (errors below 1%). The storage requirement for the ANN model is negligible compared to that associated to classical look-up tables. The stored, time-evolving ANN models can in principle be directly re-used for other simulations considering the same configuration.

As a conclusion, the developed procedure appears to be extremely promising for large-scale simulations involving detailed kinetics, like DNS with complex reaction schemes. The computational and parallel efficiency of this framework could be probably improved further by spending more effort on that issue, as well as using GPUs instead of CPUs for the ANN procedure.

Acknowledgements

The financial support of the International Max Planck Research School Magdeburg for Advanced Methods in Process and Systems Engineering (IMPRS ProEng) is gratefully acknowledged. The DNS simulations have been made possible thanks to the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu), providing computing time on the Supercomputer JUWELS at Jülich Supercomputing Centre (JSC, Germany); the support of Dr. Koh at JSC is gratefully acknowledged.

Compliance with Ethical Standards

The authors declare that they have no conflict of interest.

References

- [1] M. Schmitt, C. E. Frouzakis, A. G. Tomboulides, Y. M. Wright, K. Boulouchos, Direct numerical simulation of the effect of compression on the flow, temperature and composition under engine-like conditions, Proc. Combust. Inst. 35 (2015) 3069–3077.
- [2] P. Pepiot-Desjardins, H. Pitsch, An efficient error-propagation-based reduction method for large chemical kinetic mechanisms, Combust. Flame 154 (2008) 67–81.
- [3] J.-Y. Chen, W. Kollmann, R. W. Dibble, PDF modeling of turbulent nonpremixed methane jet flames, Combust. Sci. Technol. 64 (1989) 315–346.

- [4] U. Maas, S. B. Pope, Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space, *Combust. Flame* 88 (1992) 239–264.
- [5] B. Fiorina, R. Baron, O. Gicquel, D. Thévenin, S. Carpentier, N. Darabiha, Modelling non-adiabatic partially premixed flames using flame-prolongation of ILDM, *Combust. Theory Model* 7 (2003) 449–470.
- [6] B. Fiorina, O. Gicquel, L. Vervisch, S. Carpentier, N. Darabiha, Approximating the chemical structure of partially premixed and diffusion counterflow flames using FPI flamelet tabulation, *Combust. Flame* 140 (2005) 147–160.
- [7] M. Benzinger, R. Schießl, U. Maas, A versatile coupled progress variable/REDIM model for auto-ignition and combustion, *Proc. Combust. Inst.* 36 (2017) 3613–3621.
- [8] S. B. Pope, Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation, *Combust. Theory Model* 1 (1997) 41–63.
- [9] J. Kim, S. B. Pope, Effects of combined dimension reduction and tabulation on the simulations of a turbulent premixed flame using a large-eddy simulation/probability density function method, *Combust. Theory Model* 18 (2014) 388–413.
- [10] F. C. Christo, A. R. Masri, E. M. Nebot, T. Turányi, Utilising artificial neural network and repro-modelling in turbulent combustion, *Proceed-*

ings of ICNN'95 - International Conference on Neural Networks 2 (1995) 911–916.

- [11] F. C. Christo, A. R. Masri, E. M. Nebot, S. B. Pope, An integrated PDF/neural network approach for simulating turbulent reacting systems, *Symposium (International) on Combustion* 26 (1996) 43–48.
- [12] J. A. Blasco, N. Fueyo, C. Dopazo, J. Ballester, Modelling the temporal evolution of a reduced combustion chemical system with an Artificial Neural Network, *Combust. Flame* 113 (1998) 38–52.
- [13] J.-Y. Chen, J. A. Blasco, N. Fueyo, C. Dopazo, An economical strategy for storage of chemical kinetics: Fitting in situ adaptive tabulation with artificial neural networks, *Proc. Combust. Inst.* 28 (2000) 115–121.
- [14] Y. Choi, J.-Y. Chen, Fast prediction of start-of-combustion in HCCI with combined artificial neural networks and ignition delay model, *Proc. Combust. Inst.* 30 (2005) 2711–2718.
- [15] A. Kempf, F. Flemming, J. Janicka, Investigation of lengthscales, scalar dissipation, and flame orientation in a piloted diffusion flame by LES, *Proc. Combust. Inst.* 30 (2005) 557–565.
- [16] M. Ihme, C. Schmitt, H. Pitsch, Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame, *Proc. Combust. Inst.* 32 (2009) 1527–1535.
- [17] F. N. Pereira, A. L. D. Bortoli, Solutions for a turbulent jet diffusion flame of ethanol with NO_x formation using a reduced kinetic mechanism obtained by applying ANNs, *Fuel* 231 (2018) 373–378.

- [18] B. A. Sen, S. Menon, Turbulent premixed flame modeling using artificial neural networks based chemical kinetics, *Proc. Combust. Inst.* 32 (2009) 1605–1611.
- [19] B. A. Sen, S. Menon, Linear eddy mixing based tabulation and artificial neural networks for large eddy simulations of turbulent flames, *Combust. Flame* 157 (2010) 62–74.
- [20] Z. Zhou, Y. Lu, Z. Wang, Y. Xu, J. Zhou, K. Cen, Systematic method of applying ANN for chemical kinetics reduction in turbulent premixed combustion modeling, *Chinese Science Bulletin* 58 (2013) 486–492.
- [21] A. K. Chatzopoulos, S. Rigopoulos, A chemistry tabulation approach via Rate-Controlled Constrained Equilibrium (RCCE) and Artificial Neural Networks (ANNs), with application to turbulent non-premixed CH₄/H₂/N₂ flames, *Proc. Combust. Inst.* 34 (2013) 1465–1473.
- [22] L. L. C. Franke, A. K. Chatzopoulos, S. Rigopoulos, Tabulation of combustion chemistry via Artificial Neural Networks (ANNs): Methodology and application to LES-PDF simulation of Sydney flame L, *Combust. Flame* 185 (2017) 245–260.
- [23] K. Wan, C. Barnaud, L. Vervisch, P. Domingo, Chemistry reduction using machine learning trained from non-premixed micro-mixing modeling: Application to DNS of a syngas turbulent oxy-flame with side-wall effects, *Combust. Flame* 220 (2020) 119–129.
- [24] T. Methling, M. Braun-Unkhoff, U. Riedel, A novel linear transforma-

tion model for the analysis and optimisation of chemical kinetics, *Combust. Theory Model* 21 (2017) 503–528.

- [25] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Inc., 1995.
- [26] S. S. Haykin, Neural Networks: A Comprehensive Foundation, International edition, Prentice Hall, 1999.
- [27] M. Curcic, A parallel Fortran framework for neural networks and deep learning, *ACM Fortran Forum* 38 (2019) 4–21.
- [28] B. Fiorina, O. Gicquel, L. Vervisch, S. Carpentier, N. Darabiha, Premixed turbulent combustion modeling using tabulated detailed chemistry and PDF, *Proc. Combust. Inst.* 30 (2005) 867–874.
- [29] D. Arthur, S. Vassilvitskii, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’07, Society for Industrial and Applied Mathematics, USA, 2007, pp. 1027–1035.
- [30] H. H. Aghdam, E. J. Heravi, Guide to convolutional neural networks, Springer, New York, 2017.
- [31] A. Abdelsamie, G. Fru, T. Oster, F. Dietzscht, G. Janiga, D. Thévenin, Towards direct numerical simulations of low-Mach number turbulent reacting and two-phase flows using immersed boundaries, *Comput. Fluids* 131 (2016) 123–141.
- [32] K. E. Niemeyer, N. J. Curtis, C.-J. Sung, pyJac: Analytical Jacobian

generator for chemical kinetics, *Comput. Phys. Commun.* 215 (2017) 188–203.

- [33] C. Chi, A. Abdelsamie, D. Thévenin, Direct Numerical Simulations of Hotspot-induced Ignition in Homogeneous Hydrogen-air Pre-mixtures and Ignition Spot Tracking, *Flow Turbul. Combust.* 101 (2018) 103–121.
- [34] C. Chi, G. Janiga, K. Zähringer, D. Thévenin, DNS study of the optimal heat release rate marker in premixed methane flames, *Proc. Combust. Inst.* 37 (2019) 2363–2371.
- [35] C. Chi, A. Abdelsamie, D. Thévenin, A directional ghost-cell immersed boundary method for incompressible flows, *J. Comput. Phys.* 404 (2020) 109122.
- [36] D. G. Goodwin, H. K. Moffat, R. L. Speth, available at [⟨http://www.cantera.org⟩](http://www.cantera.org), 2015.
- [37] G. P. Smith, D. M. Golden, M. Frenklach, N. W. Moriarty, B. Eiteneer, M. Goldenberg, C. T. Bowman, R. K. Hanson, S. Song, W. C. Gardiner, Jr., V. V. Lissianski, Z. Qin, GRI-Mech 3.0, [⟨http://www.me.berkeley.edu/gri_mech/⟩](http://www.me.berkeley.edu/gri_mech/), 2020.
- [38] O. Gicquel, N. Darabiha, D. Thévenin, Laminar premixed hydrogen/air counterflow flame simulations using flame prolongation of ILDM with differential diffusion, *Proc. Combust. Inst.* 28 (2000) 1901–1908.
- [39] V. Cherkassky, F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, Inc., 1st edition, 1998.

- [40] N. Li, S. Laizet, in: 2DECOMP&FFT – A Highly Scalable 2D Decomposition Library and FFT Interface, Cray user's group 2010 conference.