The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-–9 June 2022, Oslo, Norway

SIMULATION OF REACTING FLOWS USING ARTIFICIAL NEURAL NETWORKS: IGNITION TO PROPAGATION TRANSITION IN COMBUSTION SYSTEMS

CEDRIC MEHL[1] AND DAMIEN AUBAGNAC-KARKAR[1]

[1] IFPEN, 1-4 av. du Bois Preau, Rueil-Malmaison 92852, France, Institut Carnot IFPEN Transports Energie
e-mail: cedric.mehl@ifpen.fr

Key words: Premixed combustion, Artificial Neural Networks, Auto-ignition

Abstract. The simulation of reactive flows is a major challenge in several industrial sectors, such as aeronautics or energy production. The coupling between fluid dynamics and chemistry comes however at a cost, as chemical processes involve a wide range of spatial and temporal scales. The resulting equations are stiff and require specific, and expensive, numerical methods. The use of machine learning to estimate the reaction rates has been recently proposed. In particular, Artificial Neural Networks (ANN) have the ability to perform interpolation on high-dimensional data and are thus particularly adapted to chemistry problems. A major issue is then to select an appropriate database on which to train the ANN. It must: (i) be representative of the targeted application; (ii) be sufficiently quick to generate. A promising strategy is to use 0-D stochastics reactors, which mimic reactive and mixing processes in systems while being cheap to compute. This methodology has been successfully applied to non-premixed combustion in the literature. In the present work, the aim is to investigate the ability of the 0D stochastic reactors to be used as a database for a wider range of combustion systems. More specifically, the focus will be on the ability to predict auto-ignition followed by premixed flame propagation. To that purpose, a 2-D turbulent case involving the auto-ignition of a hotspot in a hydrogen/air mixture and the subsequent propagation of a premixed flame is proposed. An ANN model based on stochastic reactors is then built and tested on (i) a 0-D auto-ignition case; (ii) a 1-D laminar premixed flame propagation; (ii) the full 2-D turbulent configuration. Using adequate data transformation at the input and output of the neural network, accurate results are obtained, highlighting the ability of the proposed strategy to deal with a large range of combustion applications.

1 INTRODUCTION

Fluid flows involving reacting chemical species arise in plenty of industrial applications, such as chemical processes or combustion engines. Mathematically, the evolution of the mixture is described by the resolution of partial differential equations (PDE) for the mass of each species, and is often expressed for mass fractions Y_k:

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial}{\partial x_i}\left(\rho\left(u_i + V_{k,i}\right) Y_k\right) = \dot{\omega}_k, \;\; k \in [\![1, N_s]\!] \tag{1}$$

where N_s is the number of species, u_i the fluid velocity, ρ the density, $V_{k,i}$ the diffusion velocities and $\dot{\omega}_k$ the chemical reaction rates. In addition, an equation for energy, not shown here, has to be considered. In many numerical solvers, the chemistry is solved apart from the other physical phenomena using operator splitting techniques. Chemistry is then described by the following set of ordinary differential equations (ODE) for species mass fractions and temperature:

$$\frac{d\rho Y_k}{dt} = \dot{\omega}_k, \ k \in [\![1, N_s]\!] \tag{2}$$

$$\frac{dT}{dt} = \sum_{k=1}^{N_s} -\frac{\dot{\omega}_k h_k}{\rho C_p} \tag{3}$$

where h_k is species k enthalpy and C_p the heat capacity at constant pressure. A major issue with the solving of Eqs. (2) and (3) is that the scales of temporal evolution are very different between species. In order to deal with this issue, implicit solvers with adaptive internal time-stepping, such as CVODE [14], have been developed. Unfortunately, the computational costs associated to these solvers is high and rapidly increase for large chemical mechanisms.

A widely used approach for tackling the large computational costs is tabulation [17, 22]. The idea is to pre-compute chemical solutions in the desired region of the chemical state space, and during the actual simulation the reaction rates are simply read from tables. The main drawback is that the tables are limited to a small number of dimensions, and the method is thus unable to deal with very complex systems. An alternative technique is to use machine learning algorithms to replace the costly chemistry solvers, and more particularly neural networks. Neural networks are able to approximate high dimensional non-linear functions and do not suffer from the shortcomings of tabulation. Some pioneering work has been performed early on [9, 4, 5], while publications have recently flourished due to the strong gain of interest in deep learning techniques [7, 23, 20, 25, 8, 3, 21, 18]. The major challenge for the success of neural networks as a surrogate for chemistry solvers is the generation of relevant training databases. Indeed, neural networks have weak extrapolation capabilities and the database must be representative of the targeted application. Two strategies can be distinguished in the literature: (i) training databases may be sampled from the actual multi-dimensional simulations [2, 8, 3]; (ii) training databases may be generated using an abstract problem, often in 0 or 1 dimension, and mimicking the physics of the actual system [23, 24, 20]. For the former, if the database generation is performed prior to the desired simulation ("offline"), the gain in computational cost of the overall process is difficult. More recently, Chi *et al.* [8] have proposed to update the neural network during the simulation ("online" learning), obtaining accurate results and an overall gain in the computing time. In this work, we will however consider option (ii), which is an approach more readily coupled to traditional CFD codes. In particular, the method proposed by Wan *et al.* [23] is selected. It is based on the definition of a set of 0-D reactors, called stochastic reactors, which evolve in time according to chemical reactions as well as an operator for mixing. The principle is to mimic actual diffusion by this operator, so that the state covered by the stochastic reactors cover the state space of the target system. This approach has been shown to work well in cases involving non-premixed combustion [23, 24, 18].

In this paper, the objective is to evaluate the extension of the stochastic reactors database generation to a wider range of combustion applications. In particular, we will assess the ability of the method to reproduce adequately auto-ignition followed by laminar premixed flame propagation. The database generation strategy and the neural network based algorithm are detailed in Sec.2. The selected validation setup is presented in Sec. 3 and the obtained results are provided in Sec. 4.

2 DATABASE AND NEURAL NETWORK DEFINITIONS

The method retained to generate the training database is first presented in this section. Methods for pre-processing the data are then exposed, and the surrogate model for chemistry integration, based on artificial neural networks, is finally detailed.

2.1 Stochastic reactors

As emphasized in several studies, the training database for learning a chemistry integration problem should be built from generic canonical problems, which represent well the physics of the targeted simulation. In the present work, we consider the micro-mixing problem previously used in the literature for both chemistry reduction [15] and ANN-based integration of chemistry [23, 24, 18]. The principle is to define a set of time-evolving reactors, also named stochastic particles, which span the desired subspace of compositions. The state of a particle p is defined by the mass fractions of each species in the mixture, $Y_k^p, k \in [\![1, N_s]\!]$, and a sensible enthalpy h_s^p. The total number of particles is constant in time and is written N_{tot}. The initial states of the particles are chosen to be representative of the initial and boundary conditions of the system to be simulated. For instance, a combustor involving fuel and air inlets will be initialized with a set of particles at the air state and a set of particles at the fuel state. The number of particles is set so that the ratio of fuel to air particles correspond to the fuel to air mass flow ratio in the actual system. Particles at high temperatures may also be added to mimic spark ignition.

Once the initial states of the particles are set, their states are evolved in time as:

$$\frac{dY_k^p}{dt} = \dot{\omega}_k^p + \mathrm{MIX}_k^p(\tau_m) \tag{4}$$

$$\frac{dh_s^p}{dt} = \dot{\omega}_{h_s}^p + \mathrm{MIX}_{h_s}^p(\tau_m) \tag{5}$$

where $\dot{\omega}_k^p$ and $\dot{\omega}_{h_s}^p$ are source terms for the mass fractions and the enthalpy due to chemical reactions. MIX_k^p and $\mathrm{MIX}_{h_s}^p$ are micro-mixing terms, which aim at mimicking mixing processes taking place in turbulent flames. The selected mixing model is here the pair-wise modified Curl model [10]. In this model, particles pairs (p, q) are selected at each time step. The properties of the mixing particles p and q are updated by the following expressions:

$$\psi^{p,new} = \psi^p + \frac{1}{2} a \left(\psi^q - \psi^p\right) \tag{6}$$

$$\psi^{q,new} = \psi^q + \frac{1}{2} a \left(\psi^p - \psi^q\right) \tag{7}$$

where ψ is a generic particle property (here $\psi \in \{Y_k, h_s\}$), and $a \in [0, 1]$ is a random number. The number of particle pairs selected at each time step is controlled by the mixing time τ_m and is taken as $N_{pairs} = N_{tot}dt/\tau_m$.

Eqs. (4) and (5) are solved using an in-house code. The thermo-chemical calculations are performed using the open-source CANTERA package [12]. In particular, the chemistry integration is based on the CVODE library [14].

2.2 Database pre-processing

Using the data generated with the stochastic reactors methodology directly as inputs and outputs of an artificial neural network might lead to issues for the following two reasons: (i) species evolve over range of values which might differ in their order of magnitude. For instance, maximal values of radicals mass fractions are much smaller than those of the fuel, the oxidiser or the combustion products ; (ii) the distribution of data may be imbalanced. Indeed, a stochastic reactor will spend more time at the equilibrium state than in the strongly reacting region of the state space where the transition from fresh to burned state occurs. This can cause the ANN to have a globally poor performance. For these reasons, a pre-processing of the data is needed before it is fed to a neural network. In this work, data transformation and data clustering are considered and presented in Sec. 2.2.1 and 2.2.2, respectively.

2.2.1 Data transformation

Non-linear transform A common technique to improve the distribution of data is to apply a non-linear function. In the present work and similarly to other works [6, 8], the logarithm function is considered:

$$\widetilde{Y}_k = \log\left(Y_k\right) \tag{8}$$

where $\widetilde{Y}_k$ is the transformed mass fraction of species k. The choice of the logarithm is motivated by the fact that the evolution of some chemical species is clustered around zero. As will be shown later, this is particularly helpful for predicting the ignition of mixtures. Note that the temperature is left untouched, as the logarithm is not expected to have an influence on its distribution.

Data scaling A second pre-processing operation is data scaling. The purpose of scaling is to make the neural network training more efficient by centering input and output values around the same value and in the same range. The standard scaler is selected here. For a given variable ψ, the transformation reads:

$$\psi^n = \frac{\psi - \overline{\psi}}{\sigma_\psi} \tag{9}$$

where $\overline{\psi}$ and σ_ψ are respectively the mean and the root mean square (RMS) of variable ψ over the training dataset. If the logarithm transformation is used, the scaling is applied to the transformed data. The n superscript is used as an indicator of data normalization.

2.2.2 Data clustering

In many complex systems, the dominant physical phenomena vary in space and time. In the context of machine learning based models, it is important that the surrogate model is adapted to the local conditions. A common strategy to tackle this issue is to cluster the data, and generate different models for each of the clusters. Two options may be envisioned to cluster the data: (i) purely data-based methods, such as k-means or self-organizing maps [3, 18, 11, 5]; (ii) methods based on physical knowledge about the system [8]. While method (ii) enables a better understanding of the clusters, it may not be suited to very complex cases. Nevertheless, a physics based approach is selected in the present study as the selected combustion cases may be conveniently analyzed in terms of a progress variable c. As the fuel considered is H_2, a progress variable based on the single H_2O species is chosen:

$$c = \frac{Y_{H_2O}}{Y^{eq}_{H_2O}} \tag{10}$$

where $Y^{eq}_{H_2O}$ is the value of Y_{H_2O} at the chemical equilibrium. Clusters are then built based on range of values of the progress variable. One neural network is then trained per cluster.

2.3 Neural network model

An artificial neural network (ANN) is used to advance the solution of equations (2) and (3) in time. To do so, the neural network infers the state at time $t + dt$ from the state at time t, where t is any given time encountered in the solving process and dt is the time-step. dt is assumed constant throughout this work. The general workflow of the ANN-based time stepping is the following:

1. The input data is pre-processed. The pre-processing might involve a non-linear transformation and data scaling, as detailed in Sec. 2.2.1. The input vector is here composed of the species mass fraction and the temperature at time t, $(Y_k(t), T(t))$.

2. The ANN is applied to the processed data. Two options have been proposed in the literature regarding the output of the ANN: (i) mass fractions at time $t + dt$ [4, 11]; (ii) increment of mass fractions from time t to time $t + dt$ [13, 23]. Option (ii) is retained in this study. Two possible formulations for the increment are then proposed and will be confronted. The first possibility is to predict increments of mass fractions as $\Delta Y^n_k = Y^n_k(t + dt) - Y^n_k(t)$. A second possibility is to predict the increment of transformed mass fractions, written as $\Delta \widetilde{Y}^n_k = \widetilde{Y}^n_k(t + dt) - \widetilde{Y}^n_k(t)$. The predicted data is always normalized in order to ease the training process.

3. Finally, the ANN output is processed to get the mass fractions $Y_k(t + dt)$. If ΔY^n_k is predicted by the network, $Y^n_k(t+dt)$ is first de-normalized to get $\Delta Y_k = Y_k(t+dt) - Y_k(t)$ ($Y_k(t)$ is already known). If, on the other hand, the output of the network is $\Delta \widetilde{Y}^n_k$, $\widetilde{Y}^n_k(t + dt)$ is first de-normalized and then applied an inverse transform to get $Y_k(t + dt)$, and thus ΔY_k. In both cases, the desired mass fractions are computed as $Y_k(t + dt) = Y_k(t) + \Delta Y_k$.

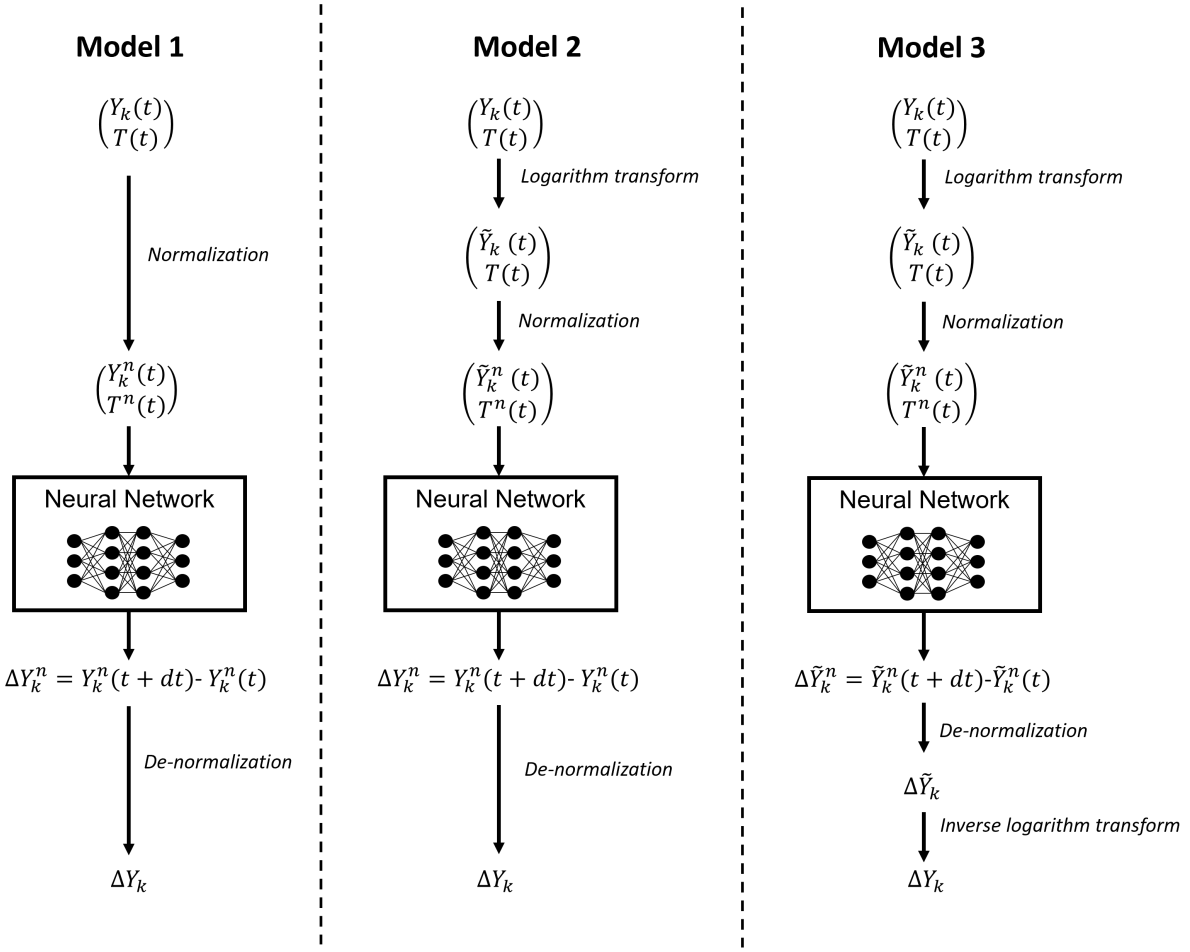

Figure 1: Summary of the ANN-based models retained in the present study.

As done elsewhere [11], the temperature at time $t + dt$ is estimated using the conservation of enthalpy. Additionally, N_2 is not considered in the network as it is a non reacting species. Note that the clustering operates here on a higher level, as the process described above is applied to each of the clusters independently. In order to assess the relevance of using logarithm transforms at the input and the output of the neural network, three ANN-based models will be compared in this study:

- **Model 1**: The input data is not transformed and the ANN outputs ΔY_k.
- **Model 2**: The input data is transformed and the ANN outputs ΔY_k.
- **Model 3**: The input data is transformed and the ANN outputs $\Delta \widetilde{Y}_k$.

A summary of the three considered models is provided in figure 1. In the next sections, the target setup will be presented and the three models will be challenged on their ability to accurately predict both ignition and premixed flame propagation at the same time.

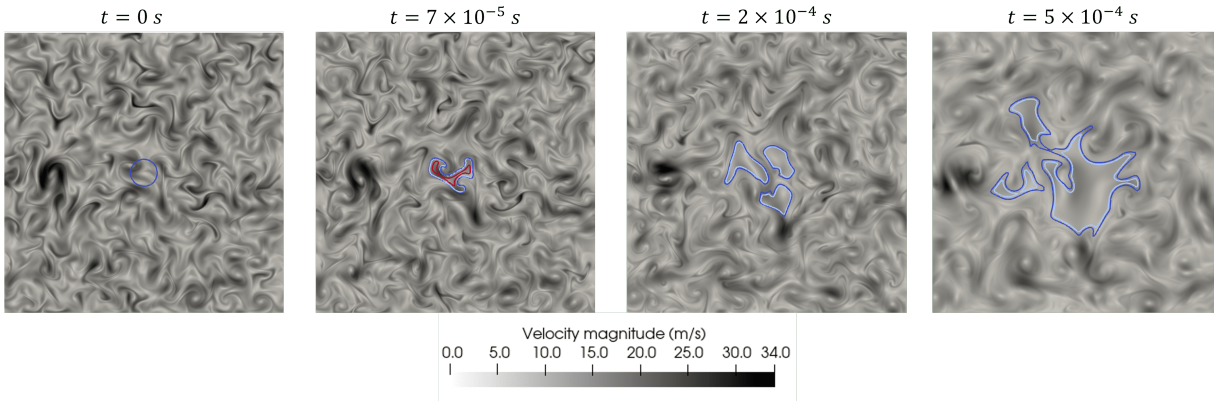

Figure 2: Evolution of a flame kernel in an initially turbulent H_2/air mixture. Lines represent iso-contours of temperature. Blue: $T = 1200$ K. Light blue: $T = 1500$ K. Red: $T = 1800$ K.

3 APPLICATION: IGNITION TO PROPAGATION TRANSITION

The objective of the present study is to apply the neural network based chemistry computation to a configuration involving a transition from ignition to premixed flame propagation. The targeted configuration is presented in Sec. 3.1 and the training database generated for this setup is analyzed in Sec. 3.2.

3.1 Targeted setup

3.1.1 Conditions

A setup involving the auto-ignition of a turbulent H_2/air mixture is selected in this work. A 2-D square domain is initially filled with H_2/air in atmospheric conditions ($p = 101325$ Pa, $T_0 = 300$ K), and at an equivalence ratio $\phi = 0.4$. The velocity field is initialized with Homogeneous Isotropic Turbulence (HIT). It is performed using an Inverse Fast Fourier Transformation method and an analytical Passot-Pouquet energy spectrum. The integral length scale is $L_t = 6.3 \times 10^{-4}$ m and the velocity fluctuation intensity is $u' = 7.78\ m.s^{-1}$. The simulation is run for one turbulent time $\tau_t = L_t/u'$ before ignition in order for the solution to converge towards a physical solution of Navier-Stokes equations.

At time $t = 0$, the temperature is set at $T_{hotspot} = 1200K$ in a circular kernel at the center of the domain. The radius of the circle is set to $3\delta_l^0$, where $\delta_l^0 = 3.12 \times 10^{-4}$ m corresponds to the laminar flame thickness of the hydrogen flame in atmospheric conditions. An illustration of the evolution of the flame kernel in the turbulent field is given in Fig. 2. Solutions are represented for times $t = 0$, 7×10^{-5}, 2×10^{-4} and 5×10^{-4} s. Iso-contours of temperature at $T = 1200$, 1500 and 1800 K are represented in blue, light blue and red, respectively. The initially hot kernel ignites after a delay theoretically estimated as $\tau_{ign} = 5.2 \times 10^{-5}$ s. The temperature then reaches values above 2200 K and progressively cools down due to diffusion processes. At this stage, the flame transitions from an auto-ignition combustion mode towards a freely propagating combustion mode. A thin flame front is created, and progressively consumes the surrounding

unburned gases.

3.1.2 Numerical solver

The CONVERGE CFD solver [19] is selected in the present work. A second-order spatial and temporal numerical scheme is used to solve the transport equations. The chemistry evolution equations are integrated using the CVODE algorithm. The domain is discretized using a uniform mesh with cell size $\Delta_x = 2.5 \times 10^{-5}$ m, which is sufficiently fine to solve the flame front, namely with 12.5 points, and the large scale turbulent structures, with 25 points.

3.2 Database generation

As highlighted in Fig. 2, the initial state of the system consists in a mixture of H_2/air in atmospheric conditions, ignited by a hot region in the center of the domain. The initial stochastic reactors states are selected to mimic this configuration: (i) 500 reactors are initialized with $T = 300$ K, $p = 101325$ Pa, $\phi = 0.4$ (the initial enthalpy $h_s^p(t = 0)$ and mass fractions $Y_k^p(t = 0)$ are derived from these variables); (ii) 100 reactors are initialized with $T = 1200$ K, $p = 101325$ Pa, $\phi = 0.4$.

Similarly to the real system, the hot particles will first auto-ignite and reach high temperatures, and will then mix with the unburned particles which will consequently burn. A low sensitivity of the results with respect to the number of reactors has been observed, although adding more reactors will certainly lead to a better covering of the state space. A condition for success is that the number of reactors at high temperature is sufficient to avoid quenching. The stochastic generation mixing time is set to $\tau_m = 4 \times 10^{-4}$ s, which is in the order of magnitude of both the flame time and the turbulent mixing time. This value leads to a satisfactory covering of the states in the present study. Note that the aim is not to have a precise estimation of mixing, rather to make sure that the targeted state space is encompassed in the training database. The stochastic reactors simulation is run for a time $10\tau_m = 4 \times 10^{-3}$ s.

The database is separated into two clusters using the progress variable, as explained in Sec. 2.2.1. As burning particles rapidly reach their equilibrium state, the database is highly skewed towards burned gases. We propose to use a specific cluster for burned gases and another cluster for both pre-igniting and reacting particles. The burned gas cluster is here defined by $c > 0.95$. Additionally, the particles with temperature $T < 600K$ are removed from the database, as no chemical reaction occur for low temperatures. Finally, the cluster for reacting particles contains 488 674 states and the cluster for burned particles 2 086 957 states.

4 VALIDATION OF THE MACHINE LEARNING STRATEGY

The validation of the ANN on the 2D turbulent case is performed in this section. The approach adopted is to first test the networks on pure ignition and pure propagation cases. The best model is then used on the turbulent case. Details about the neural networks are given in Sec. 4.1, and results are provided in Sec. 4.2, 4.3 and 4.4 for the 0-D, 1-D and 2-D cases respectively.

4.1 Neural network training

As the database is split between two clusters, two neural networks are used to predict the states. For the reacting particles cluster, the network contains two hidden layers with 80 units in each, while for the burned gas cluster, a network with two hidden layers with 40 units in each is considered. The *tanh* function is used as an activation function for all layers.

The Tensorflow [1] open-source package is used to perform the ANN training. The optimization is done using the ADAM algorithm [16]. 1000 epochs are used for training the reacting particles cluster and 300 epochs for the burned gases cluster.

4.2 Validation on auto-igniting reactors

The performance of the models on the prediction of pure ignition is tested first. A single auto-igniting 0-D reactor is considered to that purpose. The initial state of the reactor is set to the state of the hot gases in the setup described in Sec. 3.1 ($T = 1200$ K, $p = 101325$ Pa, $\phi = 0.4$). Fig. 3 shows the evolution of temperature T and mass fractions Y_{H_2}, Y_{H_2O}, Y_H in time for the three ANN-based models and the reference solution. We can see that the only model able to predict accurately the exact solution is model 3, where both inputs and outputs are transformed with the logarithm function. For the other two models, the ignition delay is under-estimated. The explanation lies in the fact that the evolution of the mixture before ignition is very slow, and better noticed at the logarithmic scale. The predicted shape of the profiles is however well predicted by the models. In particular, we can observe that the maximal value of radical H is satisfyingly reproduced by any of the three models. The failure of model 2 indicates that it is important to apply the same treatment to data at both inputs and outputs. This might be due to the fact that the network needs to learn an exponential function to get ΔY_k from $\widetilde{Y}_k$, which can cause a lack of robustness as the exponential function is sensitive to small changes.

4.3 Validation on laminar propagating flames

The models are then tested on their ability to predict the propagation of a purely premixed flame. A laminar freely propagating flame at $T = 300$ K, $p = 101325$ Pa, $\phi = 0.4$ is simulated. The laminar flame velocity, written S_l^0, is here computed from the consumption rate of H_2. Fig. 4 provides the evolution of S_l^0 for the three ANN models. The continuous black line corresponds to the solution obtained using CVODE. It converges towards a constant value, as physically expected. For each of the simulations, an averaged propagation velocity is estimated by averaging $S_l^0(t)$ for $t > 4.0\times10^{-3}$ s. The results are reported in Tab. 1. The evolution of S_l^0 for model 1 and 3 is close to constant as t increases, with only slight perturbations. However, model 2 shows large oscillations, meaning that the model is less stable. The averaged velocity is well recovered by any of the three models. These results suggest that the logarithm transformation is not needed for flames on propagating trajectories. It emphasizes also the fact that model 2 is not robust enough.

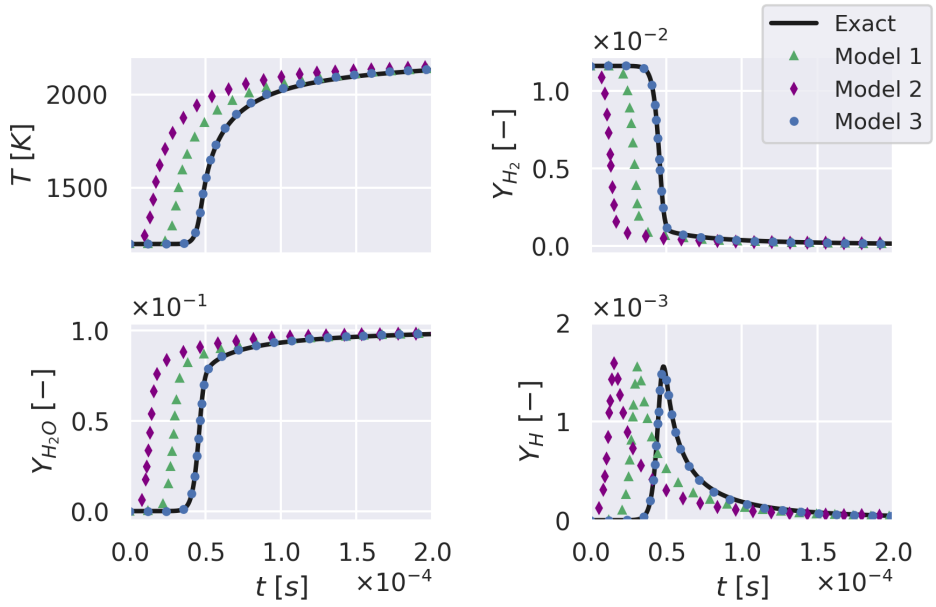

Figure 3: Comparison of ANN models on the prediction of a 0-D reactor at $T = 1200$ K, $p = 101325$ Pa, $\phi = 0.4$. Time evolutions of temperature (top left), Y_{H_2} (top right), Y_{H_2O} (bottom left) and Y_H (bottom right) are represented.

Table 1: Averaged flame speed obtained for each model on the 1-D laminar flame.

Chemistry	**Averaged flame speed** (m/s)	**Error** (%)
Exact	4.16×10^{-1}	–
Model 1	4.11×10^{-1}	1.18
Model 2	4.21×10^{-1}	1.25
Model 3	4.08×10^{-1}	1.91

4.4 Validation on turbulent auto-ignition to propagation

The only model leading to satisfactory results on both pure ignition and pure propagation is model 3, where both inputs and outputs are log-transformed. This model is here tested on the 2-D turbulent setup involving both ignition and premixed flame propagation. The evolution of the total heat release in the domain, written $\dot{q}$, is represented for the exact solver and the ANN model in Fig. 5. The evolution of the heat release is globally well predicted. A small deviation may be observed for $t > 5 \times 10^{-4}$ s. Fig. 6 and 7 provide snapshots of the evolution of T and Y_H at several instants, for the exact chemistry solver (top row) and the ANN model (bottom row). The evolution of the flame shape is well-reproduced by the ANN. As seen in Fig. 7, the peak Y_H value reaches a high value in the ignition phase and progressively decreases. This behavior is adequately reproduced by the ANN.

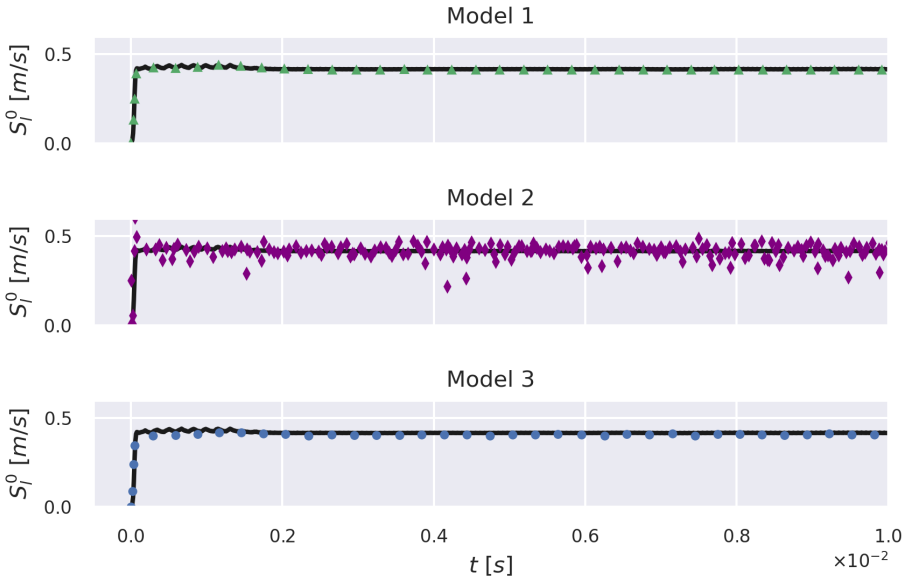

Figure 4: Comparison of ANN models on the prediction of a 1-D laminar premixed flame for $T = 300$ K, $p = 101325$ Pa, $\phi = 0.4$.

5 CONCLUSIONS

A growing interest for the use of machine learning to accelerate reacting flow simulations is observed in the recent literature. Neural networks have attracted most of the interest, due to their ability to fit complex non-linear functions, such as those implicitly defined by chemistry solvers. One of the main issue is the definition of the training database, which must be (i) representative of the targeted system as neural networks have poor extrapolation capabilities; (ii) cheap to generate as we are interested in saving time in the entire simulation workflow. A promising strategy recently proposed in the literature is based on the definition of stochastic 0-D reactors [23]. The principle is to evolve a set of reactors in time according to chemical reactions, as well as a mixing operator whose aim is to mimic mixing taking place in real systems. It has been successfully validated on non-premixed combustion [23, 24]. In the present study, we show that the applicability of this method may be extended to systems involving auto-ignition and premixed flame propagation. Several formulations of neural networks are proposed, each of them proposing a different pre-processing of the ANN input data or formulation of the ANN output. It is found that to correctly predict ignition, it is essential to use a logarithm transformation for the input data, and to predict increments of the log-transformed state at the output of the neural network. Regarding propagating flames, the use of networks with non-transformed data at input and output works equally well. Finally, the model involving log-transformed inputs and outputs has been successfully validated on a setup involving the auto-ignition of a premixed turbulent mixture of H_2 and air. Further efforts will evaluate the strategy on more complex configurations. Additionally, work will be performed to optimize the call to ANN inference in CFD codes in order to reach competitive code acceleration.

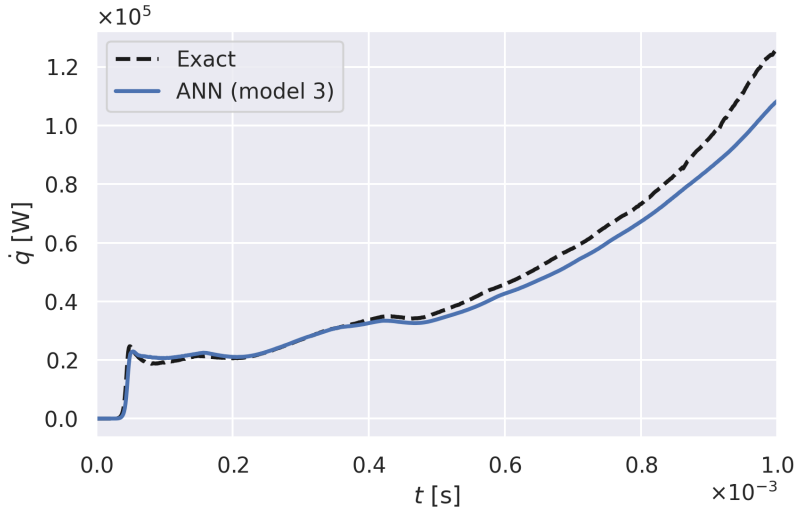

Figure 5: Evolution of the heat release rate $\dot{q}$ in time for the 2-D turbulent ignition test case.

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Technical Report arXiv:1603.04467, arXiv, Mar. 2016. arXiv:1603.04467 [cs] type: article.

[2] J. An, G. He, K. Luo, F. Qin, and B. Liu. Artificial neural network based chemical mechanisms for computationally efficient modeling of hydrogen/carbon monoxide/kerosene combustion. *International Journal of Hydrogen Energy*, 45(53):29594–29605, Oct. 2020.

[3] S. Barwey, S. Prakash, M. Hassanaly, and V. Raman. Data-driven Classification and Modeling of Combustion Regimes in Detonation Waves. *Flow, Turbulence and Combustion*, 106(4):1065–1089, Apr. 2021.

[4] J. Blasco, N. Fueyo, C. Dopazo, and J. Ballester. Modelling the Temporal Evolution of a Reduced Combustion Chemical System With an Artificial Neural Network. *Combustion and Flame*, 113(1-2):38–52, Apr. 1998.

[5] J. Blasco, N. Fueyo, C. Dopazo, and J.-Y. Chen. A self-organizing-map approach to chemistry representation in combustion applications. *Combustion Theory and Modelling*, 4(1):61–76, Mar. 2000.

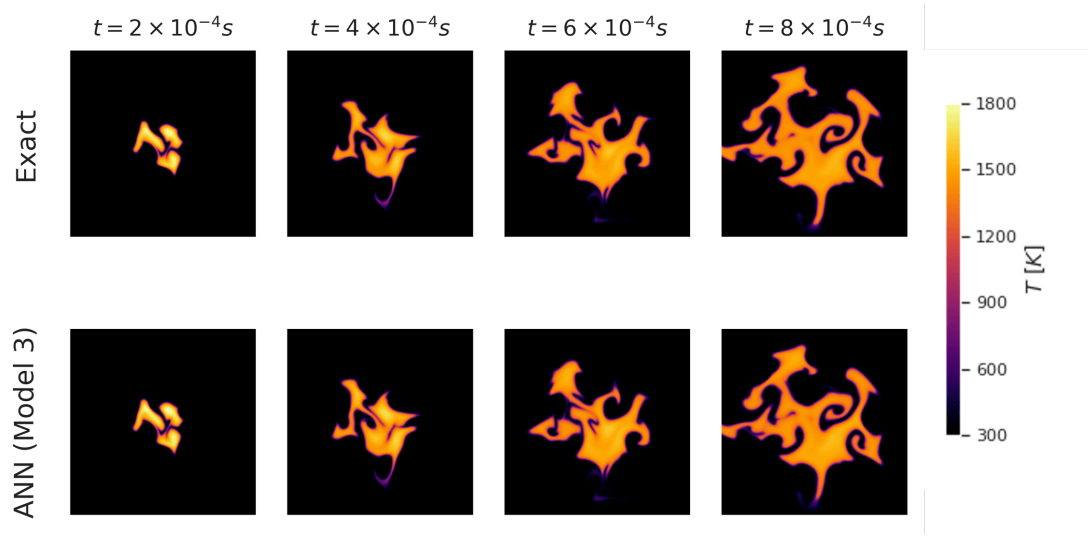

Figure 6: Temperature snapshots at several instants in time for the exact simulation (top row) and the ANN simulation with model 3 (bottom row).

[6] T. S. Brown, H. Antil, R. Löhner, F. Togashi, and D. Verma. Novel DNNs for Stiff ODEs with Applications to Chemically Reacting Flows. *arXiv:2104.01914 [cs, math]*, Apr. 2021. arXiv: 2104.01914.

[7] A. Chatzopoulos and 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. *Proceedings of the Combustion Institute*, 34(1):1465–1473, Jan. 2013.

[8] C. Chi, G. Janiga, and D. Thévenin. On-the-fly artificial neural network for chemical kinetics in direct numerical simulations of premixed combustion. *Combustion and Flame*, 226:467–477, Apr. 2021.

[9] F. Christo. Artificial neural network implementation of chemistry with pdf simulation of H2/CO2 flames. *Combustion and Flame*, 106(4):406–427, Sept. 1996.

[10] R. L. Curl. Dispersed phase mixing: I. Theory and effects in simple reactors. *AIChE Journal*, 9(2):175–181, Mar. 1963.

[11] L. L. Franke, A. K. Chatzopoulos, and S. Rigopoulos. Tabulation of combustion chemistry via Artificial Neural Networks (ANNs): Methodology and application to LES-PDF simulation of Sydney flame L. *Combustion and Flame*, 185:245–260, Nov. 2017.

[12] D. G. Goodwin, R. L. Speth, H. K. Moffat, and B. W. Weber. Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes, 2021.

[13] X. Han, M. Jia, Y. Chang, and Y. Li. An improved approach towards more robust deep learning models for chemical kinetics. *Combustion and Flame*, 238:111934, Apr. 2022.

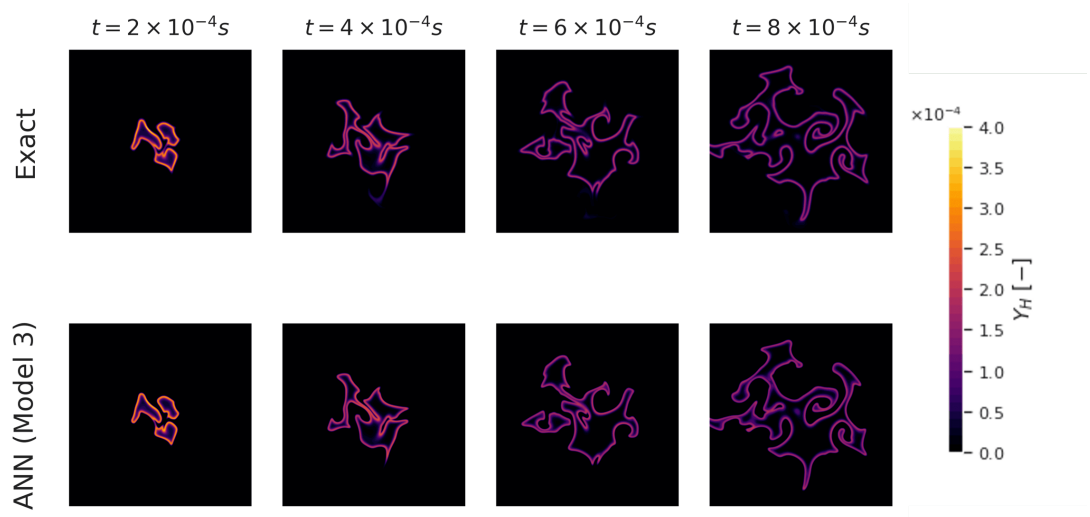

Figure 7: Y_H snapshots at several instants in time for the exact simulation (top row) and the ANN simulation with model 3 (bottom row).

[14] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, Sept. 2005.

[15] N. Jaouen, L. Vervisch, P. Domingo, and G. Ribert. Automatic reduction and optimisation of chemistry for turbulent combustion modelling: Impact of the canonical problem. *Combustion and Flame*, 175:60–79, Jan. 2017.

[16] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. Technical Report arXiv:1412.6980, arXiv, Jan. 2017. arXiv:1412.6980 [cs] type: article.

[17] U. Maas and S. Pope. Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space. *Combustion and Flame*, 88(3-4):239–264, Mar. 1992.

[18] H.-T. Nguyen, P. Domingo, L. Vervisch, and P.-D. Nguyen. Machine learning for integrating combustion chemistry in numerical simulations. *Energy and AI*, 5:100082, Sept. 2021.

[19] K. J. Richards, P. K. Senecal, and E. Pomraning. CONVERGE 3.0, 2022.

[20] B. A. Sen and S. Menon. Linear eddy mixing based tabulation and artificial neural networks for large eddy simulations of turbulent flames. *Combustion and Flame*, 157(1):62–74, Jan. 2010.

[21] A. J. Sharma, R. F. Johnson, D. A. Kessler, and A. Moses. Deep Learning for Scalable Chemical Kinetics. In *AIAA Scitech 2020 Forum*, Orlando, FL, Jan. 2020. American Institute of Aeronautics and Astronautics.

[22] J. van Oijen, F. Lammers, and L. de Goey. Modeling of complex premixed burner systems by using flamelet-generated manifolds. *Combustion and Flame*, 127(3):2124–2134, Nov. 2001.

[23] K. Wan, C. Barnaud, L. Vervisch, and 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. *Combustion and Flame*, 220:119–129, Oct. 2020.

[24] K. Wan, C. Barnaud, L. Vervisch, and P. Domingo. Machine learning for detailed chemistry reduction in DNS of a syngas turbulent oxy-flame with side-wall effects. *Proceedings of the Combustion Institute*, 38(2):2825–2833, 2021.

[25] L. Zhou, Y. Song, W. Ji, and H. Wei. Machine learning for combustion. *Energy and AI*, 7:100128, Jan. 2022.