A Physics-Constrained NeuralODE Approach for Robust Learning of Stiff Chemical Kinetics

Tadbhagya Kumar
Argonne National Laboratory
Lemont, IL
tkumar@anl.gov

Anuj Kumar
Argonne National Laboratory
Lemont, IL
akumar@anl.gov

Pinaki Pal
Argonne National Laboratory
Lemont, IL
pal@anl.gov

Abstract

The high computational cost associated with solving for detailed chemistry poses a significant challenge for predictive computational fluid dynamics (CFD) simulations of turbulent reacting flows. These models often require solving a system of coupled stiff ordinary differential equations (ODEs). While deep learning techniques have been experimented with to develop faster surrogate models, they often fail to integrate reliably with CFD solvers. This instability arises because deep learning methods optimize for training error without ensuring compatibility with ODE solvers, leading to accumulation of errors over time. Recently, NeuralODE-based techniques have offered a promising solution by effectively modeling chemical kinetics. In this study, we extend the NeuralODE framework for stiff chemical kinetics by incorporating mass conservation constraints directly into the loss function during training. This ensures that the total mass and the elemental mass are conserved, a critical requirement for reliable downstream integration with CFD solvers. Our results demonstrate that this enhancement not only improves the physical consistency with respect to mass conservation criteria but also ensures better robustness and makes the training process more computationally efficient.

1 Introduction

Computational fluid dynamics (CFD) modeling of turbulent combustion remains computationally demanding, which is attributed to the complex interaction of multiple physical phenomena, such as flow turbulence, heat transfer and chemical kinetics, spanning a range of spatio-temporal scales. Of these, modeling of detailed chemical kinetics presents the principal bottleneck. The associated kinetics is governed by an extremely stiff system of coupled ordinary differential equations (ODEs) for multiple reactive species whose dimensionality increases exponentially as reaction mechanisms get larger (in terms of number of species and chemical reactions). In order to make the computations tractable, mechanism reduction is typically performed, either through the elimination of certain reactions and species [1, 2, 3] or through analysis of the impact of timescales on the global reactions [4, 5, 6]. However, the reduced mechanisms often lead to less reliable description of chemical kinetics.

Recently, several Machine Learning (ML) approaches have been employed to emulate chemical kinetics and accelerate the associated chemistry integration. Some of these approaches reduce the dimensionality of the system through various dimensionality reduction techniques [7, 8] and

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

temporally evolve the kinetics in the reduced space [9]. Other approaches have trained neural networks aiming to predict chemical source terms as functions of the thermochemical state[10, 11, 12, 13, 14]. However, despite the potential benefits of these techniques, there are challenges. When coupled with a numerical solver, predicted solutions may diverge or become unstable. The nonlinearity of combustion process means even minor predictive errors can escalate into significant discrepancies in temporal evolution of thermochemical scalars over long-time horizon.

More recently, in an effort to alleviate these challenges, a first-of-its-kind deep learning framework based on neural ordinary differential equations (NeuralODEs) [15] known as ChemNODE [16], has been developed at Argonne National Laboratory for robust modeling of stiff chemical kinetics [16]. The approach obtains the solution vector through a stiff ODE solver operating on an ODE system parameterized by a neural network that outputs the chemical source terms. The neuralODE framework ensures that the obtained solution vectors, even after a long-time horizon, remain adherent to the ground-truth solution trajectory.

In the present work, ChemNODE [16] is enhanced by incorporating mass conservation constraints in the form of elemental mass conservation into the loss function during training, similar to PINNs [17]. It is shown that these modifications to the original approach not only improve the consistency of the resulting system to the physical laws but also make the training process computationally efficient and predictions from the neural network more robust.

2 NeuralODE Framework for Stiff Chemical Kinetics

For an unsteady chemically reacting system (with no diffusive and advective transport), the temporal evolution of reactive scalars (species) can be defined by:

$$\frac{dY_k}{dt} = \frac{\dot{\omega}_k}{\rho}, \;\; k = 1, 2, 3, ..., N_s \tag{1}$$

where Y_k is the mass fraction of species k, $\dot{\omega}_k$ is the corresponding chemical source term computed using law of mass action [18], ρ is the density, N_s is the number of chemical species. The temporal evolution of temperature is also governed by an ODE similar to Eq. 1. To calculate these source terms, one needs to account for several elementary reactions involving production and consumption of multiple species. As the chemical mechanism becomes larger, the number of intermediate reactions also increases when detailed kinetics is taken into account [19]. This leads to extreme computational costs, since all chemical time scales must be fully resolved. In the ChemNODE framework [16], the expensive physics-based computation of chemical source terms is replaced by a neural network, which can be described as:

$$\frac{d\mathbf{\Phi}}{dt} = f(\mathbf{\Phi}, t; \mathbf{\Theta}) \tag{2}$$

where $\mathbf{\Phi}$ is the vector of thermochemical state (temperature and species mass-fractions, $\mathbf{\Phi} = [T, Y_1, Y_2, ..., Y_N]$), and $f(\mathbf{\Phi}, t; \mathbf{\Theta})$ is a neural network parameterized by weights $\mathbf{\Theta}$. The problem is posed as an optimization problem of determining the optimal parameters that minimize the mean-squared error (MSE) loss function:

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{\Phi}_i - \hat{\mathbf{\Phi}}_i)^2 \tag{3}$$

where $\mathbf{\Phi}$ is the ground truth thermochemical state obtained by integrating Eq. 1, and $\hat{\mathbf{\Phi}}$ is the predicted state at different time instants during the neuralODE integration (Eq. 2).

2.1 Physics-constrained NeuralODE (PC-NODE)

Recent efforts have looked at combining domain-specific constraints as a means of embedding physics into neural network training through regularization of the loss function [17]. In this case, the system is required to adhere to the law of conservation of mass to be physically consistent. To this end, the loss function of neuralODE training is modified to include mass conservation in the form of elemental mass conservation constraints, as follows:

$$L_{PC-NODE} = L_{MSE} + \sum_{j=1}^{N_{ele}-1} \lambda_j L_{ele-j} \tag{4}$$

where L_{ele-j} incorporates the loss associated with mass conservation of element j (in the chemical system with a total of N_{ele} elements). For a mechanism with total of N_s reactive scalars (species), this term can be written as:

$$L_{ele-j} = \sum_{i=1}^{N} \frac{1}{N} \left(log \left(1 + \left| \sum_{k=1}^{N_s} \frac{N_j^k W_j}{W_k} \left(Y_{k,i} - \hat{Y}_{k,i} \right) \right| \right) \right)^2 \tag{5}$$

where W_j is the atomic mass of the element j, N_j^k is the number of atoms of element j in k^{th} species, W_k is the molecular weight of k^{th} species, and N is the number of training data points. In the formulation of the elemental mass constraints, the above logarithmic form is used because the input scalars are transformed to log basis.

3 Experiments

In reacting CFD simulations, it is a common numerical approach to decouple the chemistry from transport using operator splitting. The chemistry is solved independently from advective and diffusive transport within each computational grid cell considered as a homogeneous reactor. In this work, a canonical autoigniting homogeneous reactor of hydrogen-air combustion at constant pressure of 1 atm is considered for demonstration purposes. This configuration is compatible with the operator splitting approach in 3D CFD solvers. The detailed chemical mechanism [20] consists of 9 species ($H_2, O_2, O, OH, H_2O, H, HO_2, H_2O_2, N_2$) and 19 chemical reactions.

The training data is generated using Cantera [21], which solves the coupled ODE system (Eq. 1) using detailed chemistry. The initial conditions chosen for data generation consist of 5 equispaced initial temperatures in $T_i = [1000K, 1200K]$ and 6 equivalence ratios in $\phi_i = [0.5, 1.5]$ for a total of 30 initial condition pairs (T_i, ϕ_i). Each of these initial conditions is integrated to equillibrium, and the solution is saved at 50 points in time. The neuralODE based chemical kinetics model is initialized with the same initial conditions as the physics-based mechanism. A two-hidden layer feedforward neural network with 48 neurons in each hidden layer is used as the surrogate model ($f(\mathbf{\Phi}, t; \mathbf{\Theta})$), with hyperbolic tangent (tanh) activation function.

The input to the neural network is a vector $\mathbf{\Phi}$ containing temperature and species mass fractions (except of nitrogen (N_2) since it acts as an inert gas and does not vary with time) in the logarithmic space [16], and the output is a vector containing chemical source terms for the input vector. The forward pass through this system requires integrating the ODEs, for which an A-L 4th order ESDIRK method from Julia's DifferentialEquations.jl library [22] is used. To calculate the gradients for parameter update, forward mode automatic differentiation is employed. In this study, a second order Levenberg-Marquardt (LM) optimizer is used, wherein the weights are updated according to:

$$\mathbf{\Theta}^{n+1} = \mathbf{\Theta}^n - (\mathbf{J}^T\mathbf{J} + \nu\mathbf{I})^{-1}\mathbf{J}\mathbf{e} \tag{6}$$

where $\mathbf{J} = \partial\mathbf{e}/\partial\mathbf{\Theta}$, is the Jacobian matrix, $\mathbf{e} = \mathbf{\Phi} - \hat{\mathbf{\Phi}}$ is the error vector, and ν is a damping constant.

4 Results

In this section, results from the neuralODE trained with MSE loss (Eq. 3) and the physics-constrained loss (Eq. 4) are compared. For PC-NODE, the hydrogen(H) and oxygen(O) based elemental mass conservation terms are considered for the hydrogen-air chemistry in Eq. 5 with $\lambda_H = \lambda_O = 3$. Figure 1 compares the training loss decay (computed as MSE between ground truth and predictions) for neuralODEs trained with mean squared error loss (MSE) and with additional elemental mass constraints (PC-NODE). Incorporating the elemental mass conservation constraints in the loss function reduces the error and makes training more efficient.

Figure 2 shows profiles of the temporal evolution of temperature, mass fraction of one of the reactants (H_2), and product mass fraction (H_2O) inferred from PC-NODE. Overall, very good agreement can be seen between the ground truth (solid lines) and predictions (markers) across different initial temperatures and equivalence ratios. Further, it is seen that the PC-NODE also satisfies the overall species mass conservation better (Figure 3a), without an explicit constraint used for it during training. The ground truth value of sum of species mass fractions is 1, however there is marked deviation from this for the case where only MSE loss function is used. Figure 3b compares the elemental mass

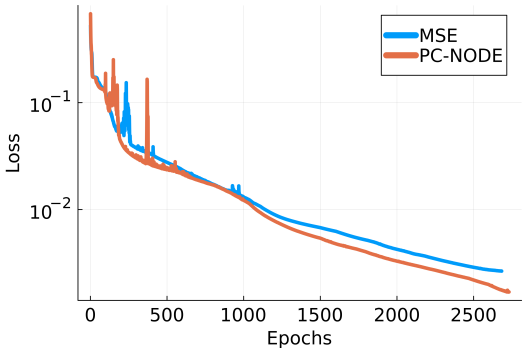
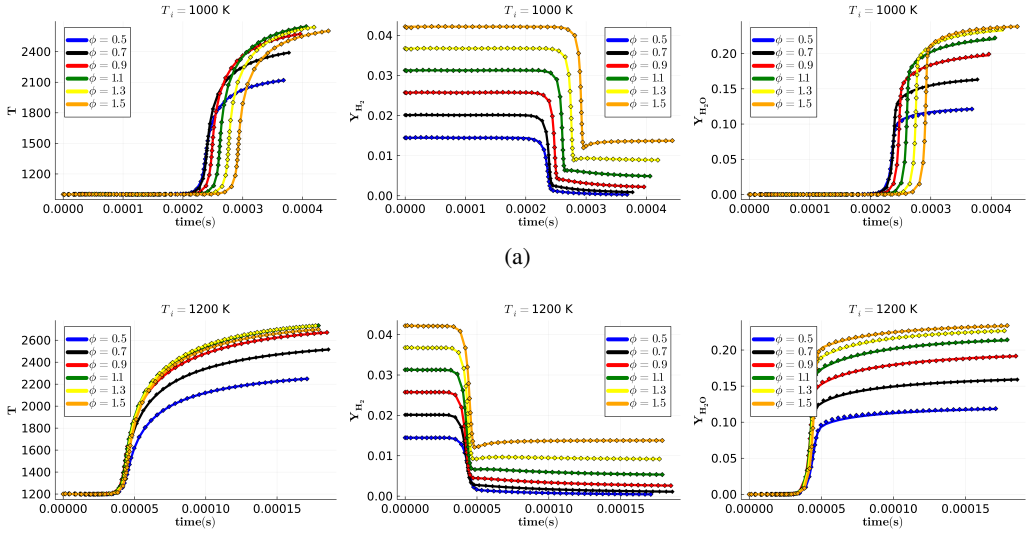


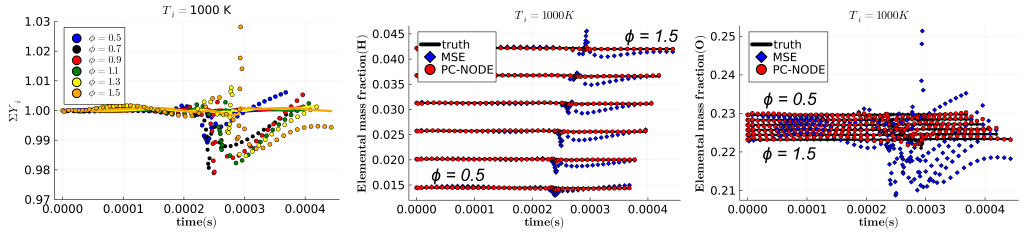
Figure 1: Mean squared error (between the ground truth and NODE predictions) evolution during training for the two cases trained with MSE loss function (Eq. MSELoss) and PC-NODE loss function (Eq. 4).



(a)

(b)

Figure 2: Temporal predictions from PC-NODE model for temperature and species mass fractions (Y_{H_2}, Y_{H_2O}) at initial temperatures: a) 1000 K, and b) 1200 K, and various equivalence ratios (ϕ). The solid lines represent the ground truth and markers represent the PC-NODE predictions.



(a)

(b)

Figure 3: Comparison of a) sum of species mass fractions between MSE trained neuralODE (marker) and the physics-constrained neuralODE (solid lines), and b) the elemental mass fraction comparison for hydrogen and oxygen at $T_i = 1000$ K, between the cases trained with MSE loss function (blue diamonds) and PC-NODE loss function (red circles). Results for different equivalence ratios ($\phi = 0.5, 0.7, 0.9, 1.1, 1.3, 1.5$) are shown in the same plot.

fractions of hydrogen and oxygen across different initial conditions, between the cases trained with and without the elemental mass constraints. With explicit constraints, elemental mass fraction for both H and O (red markers) in the predicted solution are much better conserved. The case without constraints (blue markers) displays significant deviations, which can result in non-physical solutions when combined with CFD solvers, as they require very low errors with respect to mass conservation laws.

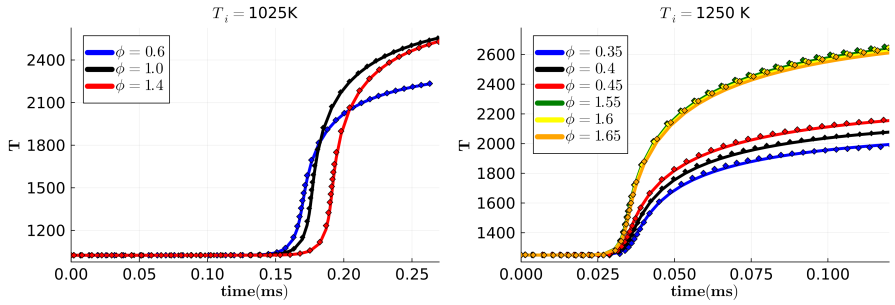


Figure 4: Temperature predictions from the PC-NODE model for out-of-sample initial temperatures of $T_i = 1025$K and $T_i = 1250$K, at various equivalence ratios (ϕ). The solid lines represent the ground truth and markers represent the PC-NODE predictions.

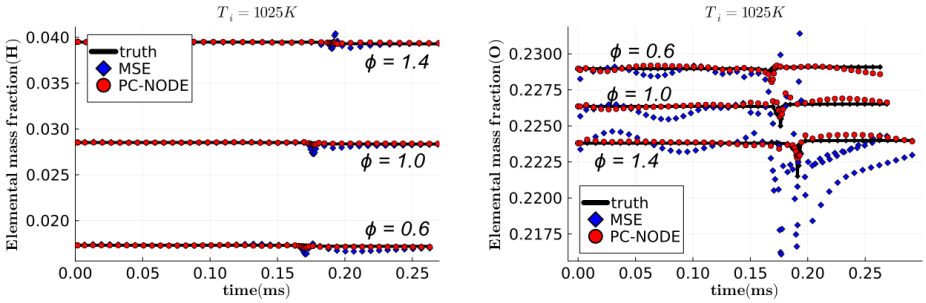


Figure 5: Comparison of elemental mass fractions of hydrogen and oxygen at $T_i = 1025$ K, between the cases trained with MSE loss function (blue diamonds) and PC-NODE loss function (red circles). Results for different equivalence ratios are shown in the same plot ($\phi = 0.6, 1.0, 1.4$).

To evaluate the robustness of the trained PC-NODE, tests are performed at thermochemical conditions which are not included in the training set. The first case consists of initial conditions that are within the bounds of those seen during training, but not included in the training set. Three different equivalence ratios ($\phi = 0.6, 1.0, 1.4$) are considered at an initial temperature of $T_i = 1025$K. To further evaluate PC-NODE, predictions are made at conditions that are outside the training bounds with an initial temperature of $T_i = 1250$K and six different equivalence ratios ($\phi = 0.35, 0.4, 0.45, 1.55, 1.6, 1.65$).

Figure 4 shows the profiles of the temporal evolution of temperature inferred from PC-NODE for both of these cases i.e unseen initial conditions within and outside the training range. It can be seen that PC-NODE predictions are in excellent agreement with the ground truth data. Further, figure 5 compares the elemental mass fractions of hydrogen and oxygen at $T_i = 1025$K, for the cases trained with and without the elemental mass constraints. Again, the elemental mass fractions of both H and O (red markers) in the predicted solution are much better conserved for PC-NODE (red). The case without constraints (blue markers), on the other hand, exhibits significant deviations.

5 Conclusion and Next Steps

In the present work, it is shown that incorporating conservation laws as physical constraints within the neuralODE approach of learning stiff chemical kinetics improves training efficiency and satisfaction of conservation laws. This work is in process of being scaled up to more complex kinetic mechanisms (for combustion of methane, ammonia etc.). In addition, these trained neural ODEs will be integrated with CFD solvers and demonstrated for multi-dimensional reacting flow simulations in future work.

Acknowledgments and Disclosure of Funding

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (Argonne). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DEAC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The research work was funded by the DOE Fossil Energy and Carbon Management (FECM) office through the Technology Commercialization Fund (TCF) program. Lastly, the authors would like to acknowledge the computing core hours available through the Bebop and Swing clusters provided by the Laboratory Computing Resource Center (LCRC) at Argonne National Laboratory.

References

[1] Tianfeng Lu and Chung K. Law. Toward accommodating realistic fuel chemistry in large-scale computations. *Progress in Energy and Combustion Science*, 35(2):192–215, 2009.

[2] W.P. Jones and Stelios Rigopoulos. Rate-controlled constrained equilibrium: Formulation and application to nonpremixed laminar flames. *Combustion and Flame*, 142(3):223–234, 2005.

[3] W. P. Jones and Stelios Rigopoulos. Reduced chemistry for hydrogen and methanol premixed flames via rcce. *Combustion Theory and Modelling*, 11(5):755–780, 2007.

[4] Mauro Valorani, Francesco Creta, Dimitris A. Goussis, Jeremiah C. Lee, and Habib N. Najm. An automatic procedure for the simplification of chemical kinetic mechanisms based on CSP. *Combustion and Flame*, 146(1):29–51, 2006.

[5] S. H. Lam. Using CSP to understand complex chemical kinetics. *Combustion Science and Technology*, 89(5-6):375–404, 1993.

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

[7] Dana H. Ballard. Modular learning in neural networks. In *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*, AAAI'87, page 279–284. AAAI Press, 1987.

[8] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[9] Anuj Kumar, Martin Rieth, Opeoluwa Owoyele, Jacqueline H. Chen, and Tarek Echekki. Acceleration of turbulent combustion dns via principal component transport. *Combustion and Flame*, 255:112903, 2023.

[10] J.A. 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):38–52, 1998.

[11] Rishikesh Ranade, Sultan Alqahtani, Aamir Farooq, and Tarek Echekki. An ann based hybrid chemistry framework for complex fuels. *Fuel*, 241:625–636, 04 2019.

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

[13] Weiqi Ji and Sili Deng. Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network. *The Journal of Physical Chemistry A*, 125(4):1082–1092, 2021.

[14] Majid Haghshenas, Peetak Mitra, Niccolò Dal Santo, and David P. Schmidt. Acceleration of chemical kinetics computation with the learned intelligent tabulation (lit) method. *Energies*, 14(23), 2021.

[15] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[16] Opeoluwa Owoyele and Pinaki Pal. ChemNODE: A neural ordinary differential equations framework for efficient chemical kinetic solvers. *Energy AI*, 7, 2022.

[17] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[18] Chung K. Law. *Combustion Physics*. Cambridge University Press, 2006.

[19] Tianfeng Lu and Chung K Law. Toward accommodating realistic fuel chemistry in large-scale computations. *Progress in Energy and Combustion Science*, 35(2):192–215, 2009.

[20] Marcus Ó Conaire, Henry J Curran, John M Simmie, William J Pitz, and Charles K Westbrook. A comprehensive modeling study of hydrogen oxidation. *International journal of chemical kinetics*, 36(11):603–622, 2004.

[21] DG Goodwin, HK Moffat, and RL Speth. Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes caltech. *Pasadena, CA*, 124, 2009.

[22] Christopher Rackauckas and Qing Nie. Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of open research software*, 5(1), 2017.