A Physics-Informed Autoencoder-NeuralODE Framework (Phy-ChemNODE) for Learning Complex Fuel Combustion Kinetics

Tadbhagya Kumar
Argonne National Laboratory
Lemont, IL 60439
tkumar@anl.gov

Anuj Kumar
North Carolina State University
Raleigh, NC 27695
akumar35@ncsu.edu

Pinaki Pal
Argonne National Laboratory
Lemont, IL 60439
pal@anl.gov

Abstract

Predictive numerical simulations of energy conversion systems involving reacting flows are accompanied by high computational cost of solving a system of stiff ordinary differential equations (ODEs) associated with detailed fuel chemistry. This bottleneck becomes more prohibitive for complex hydrocarbon fuels with an increase in the number of reactive species and chemical reactions governing chemical kinetics. In this work, a physics-informed Autoencoder (AE)-neural ODE framework (known as Phy-ChemNODE) is developed for data-driven modeling of stiff chemical kinetics, wherein a non-linear autoencoder (AE) is employed for dimensionality reduction of the thermochemical state and the NODE learns the temporal evolution of the dynamical system in the latent space obtained from the AE. Both the AE and NODE are trained together in an end-to-end manner. We further enhance the approach by incorporating elemental mass conservation constraints directly into the loss function during model training. Demonstration studies are performed for methane-oxygen combustion kinetics (32 species, 266 chemical reactions) over a wide thermodynamic and composition space at high pressure. Effects of model hyperparameters, such as relative weighting of different terms in the loss function and dimensionality of the AE latent space, are assessed on the accuracy of Phy-ChemNODE. *A posteriori* autoregressive inference tests show that Phy-ChemNODE achieves 1-3 orders of magnitude speedup relative to the methane-oxygen chemical mechanism depending on the type of the ODE solver (implicit or explicit) used, while ensuring prediction fidelity and mass conservation.

1 Introduction

Computational fluid dynamics (CFD) modeling of reacting flows, such as those encountered in gas turbine combustors and piston engines, are computationally demanding due to the complex interactions among multiple physico-chemical phenomena and the need to resolve a wide range of spatiotemporal scales governing the evolution of a large number of reactive scalars (chemical species). Of these, modeling of detailed chemical kinetics presents a major bottleneck, which is governed by a stiff system of coupled ordinary differential equations (ODEs) and characterized by high condition number of the corresponding chemical Jacobian matrix [1].

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

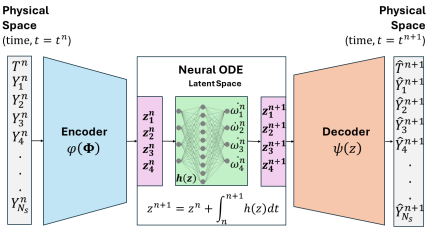


Figure 1: Schematic of the coupled AE-NODE framework.

Data-driven approaches ([2, 3, 4]) have become mainstream in an effort to emulate chemical kinetics and accelerate the associated detailed chemistry computations. In particular, deep learning based approaches have been commonly used for predicting chemical source terms from the thermochemical state ([5, 6, 7]) in an *a priori* sense, where the networks are trained in an offline manner to predict the source terms and later coupled with ODE solvers. An alternative and more robust technique for source term computations, first developed at Argonne National Laboratory, is based on neural ODEs and known as ChemNODE ([8, 9, 10, 11, 12]). It combines the source term predictions with ODE integration in an *a posteriori* learning paradigm, where the source terms predicted by the neural network are passed to the ODE solver, and the neural network weights are optimized to minimize the loss computed between the predicted and ground truth thermochemical states (comprised of species mass fractions and thermodynamic variables). This neural ODE (NODE) framework ensures that the obtained solution vectors, even after a long-time horizon, remain adherent to the ground truth solution trajectory. In more recent studies, the neural ODE based framework for chemical kinetics was further extended by incorporating mass conservation constraints[11] directly into the loss function during training, similar to PINNs [13]. This ensures that the total mass and the elemental mass are conserved. For relatively larger chemical mechanisms, the coupling of a non-linear autoencoder (AE) to perform dimensionality reduction and a neural ODE to evolve the dynamics in the lower-dimensional latent space has shown promise ([12]).

In the present work, the AE-NODE framework (Figure 1) is trained with mass conservation laws embedded within the loss function as additional constraint terms, and the effectiveness of the physics-constrained framework (Phy-ChemNODE) is highlighted. Both the AE and NODE are trained in an end-to-end manner. Demonstration studies are presented for methane-oxygen chemical kinetics, and it is shown that mass conservation constraints improve the physical consistency of the resulting data-driven model, and result in more efficient training process and robust predictions.

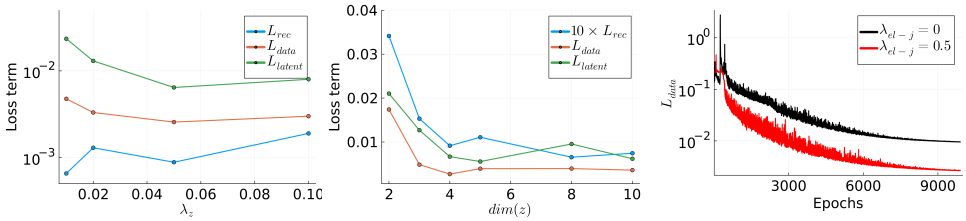
2 Physics-Constrained Autoencoder-NODE (Phy-ChemNODE) Framework for Stiff Chemical Kinetics

In combustion 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, which is equivalent to solving a system of stiff ODEs. The temporal evolution of N_s reactive scalars (chemical 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 specie k (N_s being the total number of species), $\dot{\omega_k}$ is the corresponding chemical source term computed using law of mass action, and ρ refers to density. 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 chemical species and reactions also increase [14]. This leads to prohibitive computational costs since all chemical time scales must be fully resolved. In the NODE-based data-driven framework, the expensive physics-based computation of chemical source terms is replaced by a neural network, which can be described as:

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



(a) (b) (c)

Figure 2: Comparison of loss terms (on validation set) across hyperparameter experiments: a) varying λ_z, b) varying latent space size ($dim(z)$), and c) data loss (L_{data}) evolution with ($\lambda_{el-j} = 0.5$) and without ($\lambda_{el-j} = 0$) elemental mass constraints.

where $\Phi = [T, Y_1, Y_2, \ldots, Y_{Ns}]$ is the vector of thermochemical state (temperature and species mass fractions), and $f(\Phi, t; \Theta)$ is a neural network parameterized by weights Θ. For larger chemical mechanisms, Φ increases in dimensionality and stiffness. To alleviate this, an AE is coupled with the NODE [12] for dimensionality reduction, so that the NODE learns the temporal evolution of the dynamical system in a reduced-order latent space obtained from the AE. The data-driven learning process is posed as an optimization problem of determining the optimal network parameters of the encoder (φ), NODE ($h(z)$), and the decoder (ψ), that minimize the loss function defined as:

$$L_{Phy-ChemNODE} = \lambda_{rec} L_{rec} + L_{data} + \lambda_z L_z + \sum_{j=1}^{N_{el}} \lambda_{el-j} L_{el-j} \tag{3}$$

where the reconstruction loss $L_{rec} = \mathrm{L}(\Phi, \tilde{\Phi})$ measures the loss between ground truth (Φ) and corresponding encoder-decoder mapping ($\tilde{\Phi} = \psi(\varphi(\Phi))$, the data loss $L_{data} = \mathrm{L}(\Phi, \hat{\Phi})$ measures the loss between ground truth and encoder+NODE+decoder prediction ($\hat{\Phi}$), and the latent loss $L_z = L(\bar{z}, z)$ measures the loss between encoder mapping of ground truth ($\bar{z} = \varphi(\Phi)$) and encoder+NODE prediction. Each of these loss terms is chosen to be in mean absolute error (MAE) form. Lastly, L also contains the elemental mass conservation constraints [15], defined as follows:

$$L_{el-j} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N_s} \frac{N_j^k W_j |Y_{k,i} - \hat{Y}_{k,i}|}{W_k} \tag{4}$$

where L_{el-j} refers to the loss associated with mass conservation of element j (in the chemical system with a total of N_{el} elements). $\hat{Y}_{k,i}$ and $Y_{k,i}$ correspond to the AE+NODE predicted and ground truth mass fractions of k^{th} specie, respectively. W_j is the atomic mass of element j, N_j^k is the number of atoms of element j in k^{th} specie, W_k is the molecular weight of k^{th} species, and N is the number of training data points. Lastly, the weights λ_{rec}, λ_z, and λ_{el-j} in Eq. (3) balance the contributions from the different loss terms detailed above.

3 Experiments

For proof-of-concept demonstration of Phy-ChemNODE, an auto igniting methane-oxygen (CH_4-O_2) homogeneous 0D reactor at constant pressure of 20 atm is considered. The detailed chemical mechanism [16] consists of 32 species and 266 chemical reactions. The ground truth data for model training is generated using Cantera [17], which solves the coupled ODE system (Eq. (1)). The thermodynamic and composition space chosen for data generation comprises 9 equispaced initial temperatures in the range T_i = [1600 K, 2000 K] and 11 equivalence ratios within $\phi = [1.0, 1.5]$ resulting in a total of 99 initial conditions. Each of these initial conditions is integrated to chemical equilibrium, and the thermochemical state solution is saved at 200 points in time. A $70\%, 20\%, 10\%$ random split (based on initial conditions) is used to obtain the training, validation, and test datasets, respectively. The data-driven model is initialized with the same initial conditions (during training) as the physics-based simulations. The input to the encoder is the vector Φ containing the temperature (T) and species mass fractions (Y), which is scaled using the maximum and minimum of the training data, respectively, and the output is a vector in the latent space (z). The decoder has the same dense

architecture as the encoder, with an input size equal to the latent dimension ($dim(z) = 4$) and the output size equal to the physical space vector ($dim(\Phi) = 33$). Both the encoder and decoder have 5 hidden layers with 64 neurons each and Exponential Linear Unit (ELU) activation function. The NODE has the same input and output dimensions as the latent space ($dim(z) = 4$), and a 4 hidden-layer dense network with 64 neurons in each hidden layer with ELU activation function, to model the chemical source terms in the latent space. The output layers for the encoder, NODE, and the decoder are considered to be linear. The forward pass through the NODE requires time integration, for which a 4th order explicit Runge-Kutta (RK) solver from Julia's DifferentialEquations.jl library [18] is used. Once the time integration is completed, the trajectories are mapped back to physical space and the loss is computed using Eq.(3). The gradients for updating the neural network parameters are calculated using backward adjoint automatic differentiation and ADAM optimizer with exponential learning rate decay (every 200 epochs) is used. The model is trained for 10000 epochs. To ensure that all loss terms are of similar magnitude, $\lambda_{rec} = 5.0$, $\lambda_z = 0.05$ are used. Elemental mass constraint weights (λ_{el-j}) for the three constituent elements carbon (C), hydrogen (H), and oxygen (O) are chosen as $\lambda_{el-H} = \lambda_{el-C} = \lambda_{el-O} = 0.5$. These values were determined based on a hyperparameter sweep and yielded the lowest validation loss. The training framework was implemented in Julia programming language and the model was trained using 2 AMD EPYC 7713 64-core processors for a walltime of 96 hours.

4 Results

In this section, results from the trained Phy-ChemNODE framework are presented. To determine the optimal weighting of terms in the loss function and the latent space dimensionality, hyperparameter studies were carried out. Figure 2a compares the validation set loss terms (post training) corresponding to different λ_z values for $\lambda_{rec} = 5$, $\lambda_{el-j} = 0.5$ and fixed size of the latent space (dim(z) = 4), and Figure 2b shows a similar comparison for varying size of the latent space (dim(z)) with $\lambda_{rec} = 5$, $\lambda_z = 0.05$, and $\lambda_{el-j} = 0.5$.

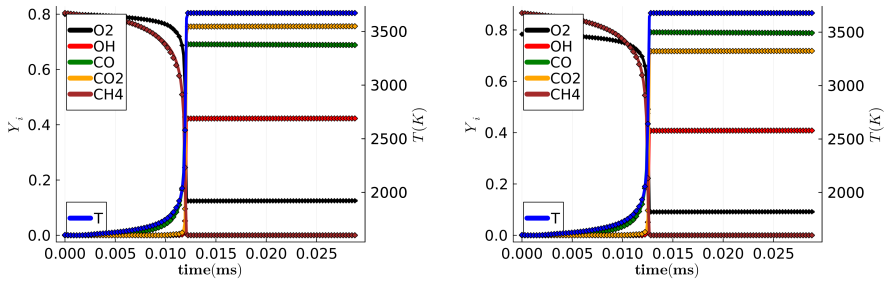
To compare the effect of adding elemental loss constraints to the training objective, Figure 2c compares the evolution of data loss (during training) between the cases trained with ($\lambda_{ele-j} = 0.5$) and without elemental mass constraints ($\lambda_{ele-j} = 0$) on the validation set, and shows that the inclusion of constraints enable more efficient model training. The trained framework is then used to predict the temporal evolution of thermochemical scalars, as shown in Figure 3 for temperature (T) and species mass fractions (CH_4, CO, CO_2, OH, and O_2) for an initial condition (corresponding to $T_0 = 1600$K) in the training set ($\phi = 1.0$) and test set ($\phi = 1.1$), with the ground truth depicted in solid lines and the predicted Phy-ChemNODE solutions (from *a posteriori* autoregressive runs) shown in markers. Excellent agreement can be observed between the two. Figure 4 shows the temporal evolution of a few intermediate species for another set of initial conditions in the training set ($T_0 = 1650$K, $\phi = 1.0$) the test set ($T_0 = 1700$K, $\phi = 1.05$), demonstrating high accuracy. As further quantification of the accuracy of Phy-ChemNODE framework, Figure 5 shows the test set MAEs for some of the thermochemical scalars, scaled by their corresponding data ranges. Based on inference on an Intel i7-1165G7 workstation with 16 cores, Phy-ChemNODE yields speedups of 6x and 860x over the full chemical mechanism in terms of overall simulation walltime, when deployed with implicit (BDF) and explicit (RK45) solvers, respectively. Lastly, Figure 6 shows the temporal evolution of C, H, and O mass fractions for certain initial conditions from the training and test sets. Evidently, the model trained with elemental mass conservation constraints in the loss function obeys mass conservation better than the case without constraints during deployment.

Conclusion and Next Steps

The NODE approach of learning stiff chemical kinetics is enhanced by coupling with an autoencoder to perform dimensionality reduction and incorporating elemental (atom) mass constraints in the loss function (Phy-ChemNODE). In future work, the training framework will be scaled to handle wider range of thermodynamic conditions (needed for practical deployment in combustion CFD simulations) and uncertainty quantification will be incorporated.

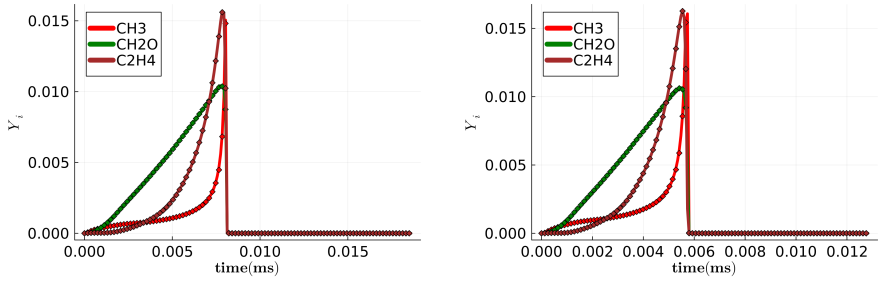
Acknowledgments and Disclosure of Funding

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,



(a) (b)

Figure 3: Temporal evolution of temperature (T) and mass fractions of CH_4, CO, CO_2, OH, and O_2 corresponding to initial conditions in a) training set ($T_i = 1600$K, $\phi = 1.0$) and b) test set ($T_i = 1600$K, $\phi = 1.1$). The mass fractions of CH_4, CO_2 and OH are scaled by 4, and that of CO by 3 for ease of plotting. Solid lines denote ground truth and markers denote Phy-ChemNODE predictions.



(a) (b)

Figure 4: Temporal evolution of intermediate species (CH_3, CH_2O and C_2H_4) corresponding to initial conditions in a) training set ($T_i = 1650$K, $\phi = 1.0$) and b) test set ($T_i = 1700$K, $\phi = 1.05$). Solid lines denote ground truth and markers denote Phy-ChemNODE predictions.

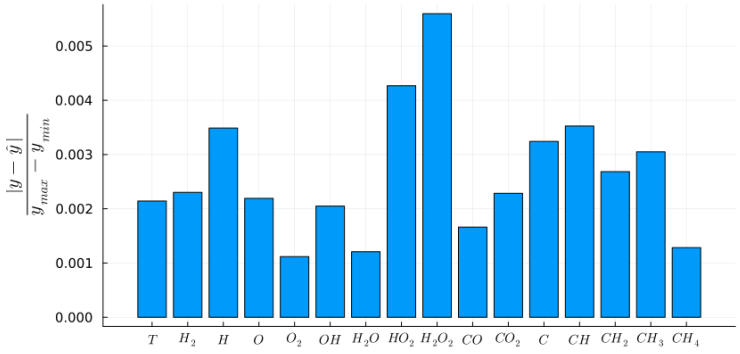
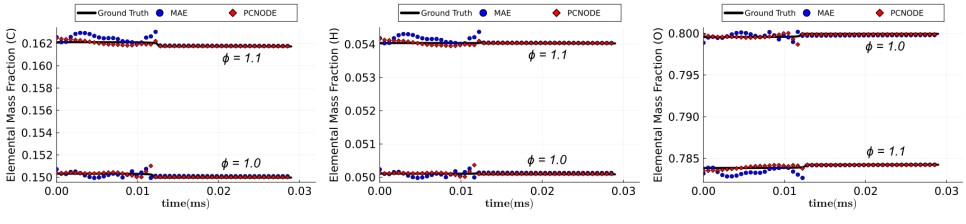


Figure 5: Scaled test set MAEs for temperature (T) and a few species mass fractions.



(a) (b) (c)

Figure 6: Temporal evolution of mass fractions of: a) C, b) H, and c) O elements corresponding to initial conditions in the training ($T_0 = 1600$K, $\phi = 1.0$) and test ($T_0 = 1600$K, $\phi = 1.1$) sets.

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] Lawrence F Shampine. Ill-conditioned matrices and the integration of stiff odes. *Journal of computational and applied mathematics*, 48(3):279–292, 1993.

[2] 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.

[3] 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.

[4] 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.

[5] F.C. Christo, A.R. Masri, E.M. Nebot, and S.B. Pope. An integrated PDF/neural network approach for simulating turbulent reacting systems. *Symposium (International) on Combustion*, 26(1):43–48, 1996.

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

[7] Kaidi Wan, Camille Barnaud, Luc Vervisch, and Pascale 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, 2020.

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

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

[10] Suyong Kim, Weiqi Ji, Sili Deng, Yingbo Ma, and Christopher Rackauckas. Stiff neural ordinary differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(9):093122, 09 2021.

[11] Tadbhagya Kumar, Anuj Kumar, and Pinaki Pal. A physics-constrained neural ODE approach for robust learning of stiff chemical kinetics. In *NeurIPS Machine Learning and the Physical Sciences Workshop*, 2023.

[12] Vijayamanikandan Vijayarangan, Harshavardhana A. Uranakara, Shivam Barwey, Riccardo Malpica Galassi, Mohammad Rafi Malik, Mauro Valorani, Venkat Raman, and Hong G. Im. A data-driven reduced-order model for stiff chemical kinetics using dynamics-informed training. *Energy and AI*, 15:100325, 2024.

[13] 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.

[14] 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.

[15] Tadbhagya Kumar, Anuj Kumar, and Pinaki Pal. A posteriori evaluation of a physics-constrained neural ordinary differential equations approach coupled with CFD solver for modeling stiff chemical kinetics. *arXiv preprint arXiv:2312.00038*, 2023.

[16] Gregory P Smith, Y Tao, and H Wang. Foundational fuel chemistry model version 1.0 (ffcm-1). *epub, accessed July*, 26:2018, 2016.

[17] 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.

[18] 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.