

# Artificial Neural Network based Vapor-Liquid Equilibrium Modeling for Simulation of Transcritical Multiphase Flows

Navneeth Srinivasan\*, Hongyuan Zhang<sup>†</sup> and Suo Yang<sup>‡</sup>

*Department of Mechanical Engineering, University of Minnesota – Twin Cities, Minneapolis, MN 55455, USA*

The requirement of high power outputs and high efficiencies of combustion engines such as rocket engines, diesel engines, and gas turbines has resulted in the incremented of the system pressure close to the thermodynamically critical point. This increase in pressure often leads to the fluids becoming either transcritical or supercritical in state. This has led to increased interest in both the multi-component phase change phenomena as well as their chemical reactions. In this work, an artificial neural network (ANN) aided VLE model is coupled with a fully compressible computational fluid dynamics (CFD) solver to simulate the transcritical processes occurring in high-pressure liquid-fueled propulsion systems. The ANN is trained on Python using the TensorFlow library, optimized for inference (i.e., prediction) using ONNX Run-time (a cross-platform inference and training machine-learning accelerator), and coupled with a C++ based fully compressible CFD solver. This plug-and-play model/methodology can be used to convert any fully compressible and conservative CFD solver to simulate transcritical processes using only open-source packages, without the need of in-house VLE-based CFD development. The solver is then used to study high-pressure shock-droplet interaction in both two- and four-component systems where qualitative and quantitative agreement is shown with results based on both direct evaluation and the state-of-the-art *in situ* adaptive tabulation (ISAT) method. The ANN model is faster than the direct evaluation method and the ISAT model by 4 times for the four-component shock-droplet interaction. The ANN model also shows implicit load balancing as long as the MPI decomposition is performed uniformly amongst the number of cores chosen, as the inference time for ANN predict does not change with the change in thermodynamic state, unlike traditional VLE solvers. Regarding the parallel scalability of this model, good strong scaling characteristics with number of processors is also observed.

## I. Introduction

With ever increasing demand for high performance combustors, increasing the chamber pressure is one often sought after option. This leads to the working conditions to overlap with the supercritical regime of the reactants. Due to the high-pressure environment in engine combustors, the injected multi-component liquid propellants and fuel-air mixtures often go through thermodynamically transcritical processes during the spray breakup, evaporation, mixing, and combustion processes. Efficient spray breakup and evaporation of liquid fuels are the primary targets of engine combustor design and control to ensure sufficiently small “effective” evaporation time. The existing liquid fuel injectors and multi-component liquid fuels developed for low pressures are not optimal at high pressures, and hence require re-design and optimization.

To understand the transcritical/supercritical mixing and combustion process, high-fidelity simulation tools are needed. Since transcritical/supercritical regions are far from the ideal gas region, real-fluid effect needs to be considered to capture correct behavior. In addition, transcritical/supercritical fluid behavior can be peculiar because of the large variation of thermophysical properties such as density and specific heat near the critical point. As a result, the Computational Fluid Dynamics (CFD) modeling of transcritical/supercritical flows is very challenging. Since small changes in temperature and pressure can have large effects on the structure of a fluid near the critical point, local properties are very important. Furthermore, a supercritical fluid lacks surface tension, which means the modeling transcritical flow needs to capture the surface tension change when the fluid goes across phase boundary. This makes simulation of transcritical flows more challenging than supercritical flows.

\*Ph.D. Student, srini237@umn.edu, Student Member AIAA.

<sup>†</sup>Ph.D. Candidate, zhan6305@umn.edu, Student Member AIAA.

<sup>‡</sup>Richard & Barbara Nelson Assistant Professor, suo-yang@umn.edu (Corresponding Author), Senior Member AIAA.

The studies of transcritical/supercritical injection, mixing, and combustion have attracted much interest in the past 30 years. However, most of them were mainly concentrated on the single-component system, whose critical point is a constant value. As long as the fluid exceeds its critical point, it goes into the supercritical state, and the classical “dense-fluid” approach is used with the assumption of a single phase [1]. Since the real mixture critical pressure could be significantly higher than the critical pressure of each component [2], the accurate mixture critical point needs to be obtained.

The requirement for a thermophysical framework to capture the above motivated physical phenomena is achieved by using the vapor-liquid equilibrium (VLE) theory. The VLE theory enforces mechanical, thermal, and chemical equilibrium between the two phases and hence estimates the phase fraction, phase densities, and phase compositions. The real-fluid effects can be coupled with the VLE model by usage of a real-fluid equation of state (EOS). Zhang et al. [3] showed details regarding the VLE framework implementation. The computational costs associated with VLE calculations can occupy up to 90% of the overall computational costs as noted by Zhang and Yang [4]. Each VLE estimation has two loops, an inner loop to converge the phase separation and an outer loop to converge the chemical equilibrium check. Within the outer loop, a real-fluid properties estimation loop is also presented, which further adds to the computational costs. Real-fluid parameters are subjected to mixing rules and require nested loops to account for multi-component effects. Finally, both the loops are converged using a Newton iteration whose convergence characteristics are heavily dependent on the initial guess provided. VLE calculations as expected happen to be not only computationally expensive, but also often non-robust. These calculations often do not converge and result in interruptions to the CFD simulation (i.e., the CFD code crashes). The loss of robustness is associated with the multi-component VLE calculations as an optimal guess is very difficult to find. This can lead to further issues and now one would need to attempt multiple initial guesses until a converged solution can be obtained and hence further adding to the computational costs.

The computational costs associated with running VLE flash calculations for 3D simulations can quickly grow infeasible and requires techniques to mitigate this risk. Fathi et al. [5] developed a reduced-order VLE calculator based on reducing the multi-component interaction matrix. The results showed the reduction of overall computational costs associated with the VLE calculations for multi-component systems, but does not guarantee robustness. Zhang and Yang [4] introduced the usage of *in situ* adaptive tabulation (ISAT) for VLE calculations to reduce the computational costs. A 400 times speed up is achieved by the ISAT implementation showing its excellent performance, but ISAT also depends on the direct calculation of VLE solutions, which still suffers from robustness issues.

One solution to avoid online VLE calculation to avoid the robustness issues is to use pre-generated VLE look-up tables which can then be used in the CFD simulation. Though this method ideally is feasible, the generated table for multi-component system suffers from the curse of dimensionality: the table size can quickly exceed 1 TB (for only 4 components) and cannot be hosted on the random-access memory (RAM) of the state-of-the-art high-performance computing (HPC) systems. This motivates the usage of other data science and machine learning methods, such as artificial neural networks (ANNs), which can perform non-linear regression while only costing MBs in RAM requirement. ANNs have been predominantly used on fluid flow for multiple applications, such as turbulence LES modeling, ignition kernel clustering and capturing, image analysis for post processing, etc. ANNs have shown great potential for non-linear regression, especially when deep neural networks are employed. However, the biggest advantage with ANNs is the ability to fill in gaps in the data where the data generation algorithm failed (i.e., where VLE solution does not converge easily) such that the robustness issues of VLE can be resolved.

Usage of a well-trained ANN to perform the VLE calculations will result in increased robustness during CFD run time along with minimized computational costs. Yue et al. [6] used ANNs to fit experimental VLE data for diesel spray conditions and showed the methods’ applicability. In this study, we generate the data using a custom-written VLE code and then train the ANN. The trained ANN is then coupled with a fully compressible CFD solver and a shock-droplet interaction case is used to test the performance of this the ANN-VLE approach. The results generated using the ANN-based VLE CFD code is compared against both direct estimation as ISAT-based estimation to show the accuracy and computational efficiency/speed of the ANN-based approach.

The remaining of this paper is organized as follows: Section II discusses the numerical models used here, including the VLE model, the ANN model, and also the methodology followed to couple the ANN with the CFD code. The results and discussions are shown in Sec. III, and the paper is concluded in Sec. IV.

## II. Numerical Methods

### A. Models of thermodynamic and transport properties

Here, we use vapor-liquid equilibrium (VLE) solvers to capture the phase change and determine the critical point of multi-component mixture in high-pressure transcritical multiphase flow as described above. VLE describes the phase equilibrium between liquid and vapor phases and solving the set of VLE equations gives the phase fraction and compositions in the two phases. If the gas (vapor) phase mole fraction is equal to 0 or 1, then the system is in a purely liquid or gaseous phase, respectively. If the system falls into the two-phase region, gas phase mole fraction will be between 0 and 1, and an equilibrium between vapor and liquid will be observed. If at certain conditions, thermodynamic properties become identical between liquid and gas, it indicates the occurrence of transcritical transition from a subcritical state to a supercritical state (which could be either a liquid-like or gas-like state).

The fluid solver that is implemented is coupled with isobaric and isenthalpic (PHn) flash solver [7]. PHn flash and almost all other VLE solvers, are developed based on TPn flash. Specifically, PHn flash solves the VLE equation set at given enthalpy ( $H$ ) rather than temperature ( $T$ ). TPn flash is the most basic VLE solver, which solves the set of VLE equations at given temperature ( $T$ ), pressure ( $P$ ), and mole fraction of each component ( $n$ ) in the system.

**Isothermal and isobaric (TPn) flash:** VLE is governed by fugacity equality Eq. (1) and Rachford-Rice equation [8] Eq. (2), which is an additional constraint to the equilibrium solver as used in Saha and Carroll [9] and obtained from the conservation of each component.

$$f_{i,l} / f_{i,g} = 1, \quad (1)$$

$$\sum_{i=1}^N \left\{ z_i (1 - K_i) / \left[ 1 + (K_i - 1) \psi_g \right] \right\} = 0, \quad (2)$$

$$K_i = y_i / x_i, \quad (3)$$

$$\sum_{i=1}^N x_i = \sum_{i=1}^N y_i = 1, \quad (4)$$

where  $f_{i,p}$  is the fugacity of component  $i$  in phase  $p$  ( $p = l$ : liquid;  $p = g$ : gas),  $x_i$  is the mole fraction of component  $i$  in liquid phase,  $y_i$  is the mole fraction of component  $i$  in gas phase,  $z_i$  is the mole fraction of component  $i$  in the feed (i.e., the entire mixture, including both gas phase and liquid phase),  $\psi_g$  is the gas mole fraction,  $K_i$  is the equilibrium constant of component  $i$ .

The real-fluid properties are described using the Peng-Robinson equation of state (PR-EOS) [10] as:

$$P = \frac{RT}{V - b} - \frac{a}{V(V + b) + b(V - b)}, \quad (5)$$

where  $P$ ,  $R$ ,  $T$  and  $V$  are pressure, gas constant, temperature, and specific volume, respectively. For a single-component fluid, the PR-EOS parameters are given by

$$a = 0.45724 \frac{R^2 T_c^2}{p_c} \hat{a}, \quad b = 0.07780 \frac{RT_c}{p_c}, \quad (6)$$

$$\hat{a} = \left( 1 + \kappa \left( 1 - (T_r)^{1/2} \right) \right)^2, \quad \kappa = 0.37464 + 1.54226\omega - 0.26992\omega^2, \quad (7)$$

where subscript “ $c$ ” means the critical value, subscript “ $r$ ” means the reduced value (e.g.,  $T_r = T/T_c$ ),  $\omega$  is the acentric factor.

The liquid phase and the gas phase are described by two multi-component PR-EOS, respectively. The specific volume of each phase,  $V_p$ , is solved from PR-EOS. The compressibility factor of each phase ( $Z = PV/RT$ ) can also be obtained from the specific volumes.

The fugacity formula of PR-EOS is shown below [11]:

$$f_i = P \chi_i \exp \left[ \frac{B_i}{B_{mix}} (Z - 1) - \ln(Z - B_{mix}) - \frac{A_{mix}}{2\sqrt{2}B_{mix}} \left( \frac{2 \sum_j x_j A_j}{A_{mix}} - \frac{B_i}{B_{mix}} \right) \ln \left( \frac{Z + (1 + \sqrt{2})B_{mix}}{Z + (1 - \sqrt{2})B_{mix}} \right) \right], \quad (8)$$

where  $\chi_i$  is the mole fraction of component  $i$  (for liquid,  $\chi_i = x_i$ ; for gas phase,  $\chi_i = y_i$ ),

$$A_i = \frac{a_i p}{R^2 T^2}, \quad (9)$$

$$B_i = \frac{b_i p}{RT}, \quad (10)$$

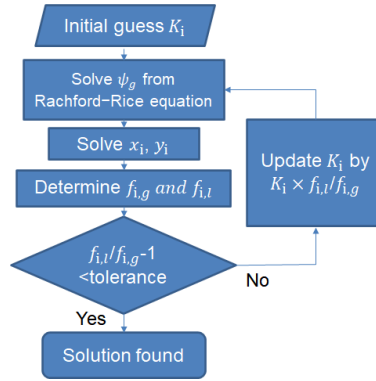
$$A_{mix} = \sum_i \sum_j x_i x_j (1 - b_{ij}) \sqrt{A_i A_j}, \quad (11)$$

$$B_{mix} = \sum_i x_i B_i \quad (12)$$

The equation set Eqs. (1-12) is solved using the Newton iteration method. The flow chart of the TPn flash is shown in Fig. 1. The initial guess is obtained using the Wilson equation [12]:

$$K_i = e^{5.373(1+\omega_i)(1-1/T_{r,i})} / P_{r,i}, \quad (13)$$

where  $\omega_i$  is the acentric factor of component  $i$ ;  $T_{r,i}$  and  $P_{r,i}$  are the reduced temperature and reduced pressure of component  $i$ , respectively.



**Fig. 1 Flow chart of the TPn flash solver.**

**PV flash and UV flash:** The PV flash and UV flash solvers are developed based on the TP flash. Both of them use iteration methods. Specifically, initial guesses (T for PV flash; T and P for UV flash) are obtained from the previous time step, and a TP flash problem is solved in each iteration. After several iterations, when the error is smaller than tolerance, the solver returns a solution.

In PV flash, since pressure (P) is already given as an input, only temperature (T) needs to be guessed and updated during the iteration. A secant method is used to avoid the expensive derivative computation in the Newton-Raphson method. In UV flash, two variables (both T and P) need to be guessed and updated simultaneously during the iteration, and hence the secant method cannot be applied. The Newton-Raphson method is used to solve the UV flash problems. The required Jacobian matrix is obtained using the analytical framework of Tudisco and Menon [13].

**Transport properties:** The dense fluid formula (i.e., Chung's method) [14] is used to evaluate the dynamic viscosity and thermal conductivity at high-pressure transcritical conditions. This method gives accurate estimations of viscosity and thermal conductivity of polar, non-polar and associating pure fluids and mixtures. Its dynamic viscosity and thermal conductivity have a similar formula:

$$\lambda = \lambda_0 \lambda^* + \lambda_p, \quad (14)$$

where  $\lambda$  represents dynamic viscosity or thermal conductivity.  $\lambda_0$  is the gas property at low pressures.  $\lambda^*$  and  $\lambda_p$  are high-pressure corrections. At high pressures,  $\lambda_p$  is the major contributing term comparing to  $\lambda_0 \lambda^*$ . On the other hand, at low pressures,  $\lambda^*$  is approaching unity, and the  $\lambda_p$  term is negligible such that Eq. 14 reduces to  $\lambda_0$ . Hence, the transition between subcritical and supercritical is smoothly described by the model.

For mass diffusivity, we used mixture-averaged mass diffusion model. The mass diffusion coefficient of specie  $i$ ,  $D_i$ , which was defined by Kee et al. [15]:

$$D_i = \frac{1 - Y_i}{\sum_{j \neq i}^N X_j / D_{j,i}}, \quad (15)$$

where  $Y_i$  and  $X_i$  are the mass and mole fractions of the  $i$ -th species, respectively;  $D_{i,j}$  is the binary diffusion coefficient, which is evaluated by Fuller's model [16] with Takahashi's correction for high pressures [17].

## B. Artificial Neural Network (ANN) Formulation

ANNs are computing systems, inspired by the biological neural networks of the brain that are very good at handling complex problems. ANN consists of an input layer, one or more hidden layers, and an output layer. The neurons in two adjacent layers connect with different weights and biases by transmitting signals. All signals received by a neuron are added together and an activation function is applied to determine the activation value. ANN's learn complex input-output relationships in a process called training, in which the weights and biases are adjusted to accurately represent the relation between a large number of example inputs and outputs. In general, an ANN can be trained to be more accurate with more training data available.

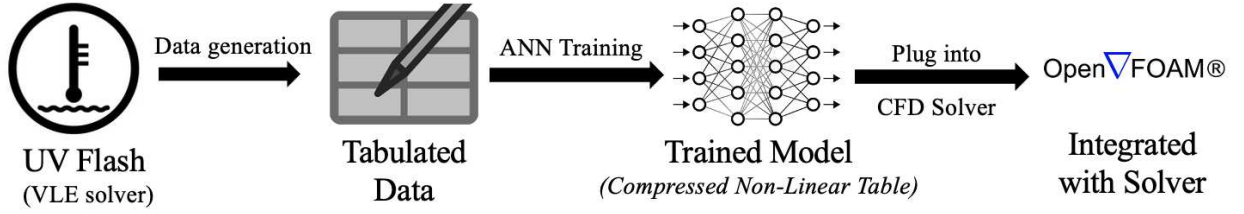


Fig. 2 Pipeline of the ANN-VLE Methodology.

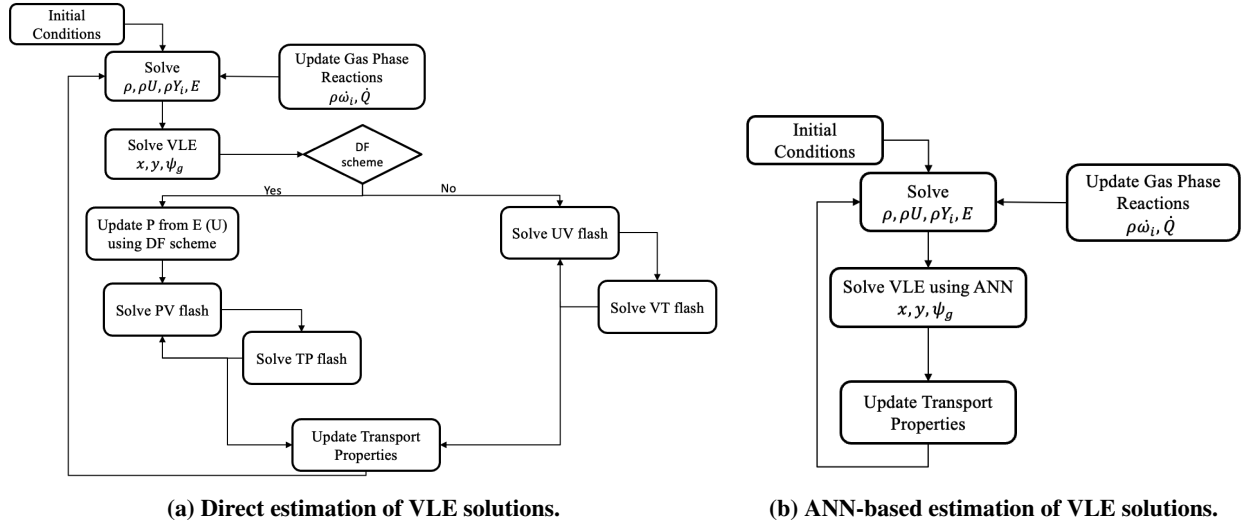
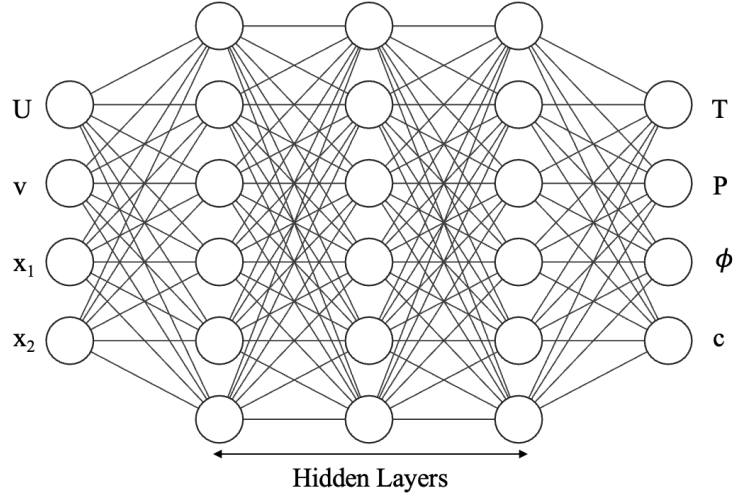


Fig. 3 VLE-based CFD algorithm flow chart.

The ANN training pipeline is shown in Fig. 2. The data generation is performed using an in-house VLE code developed on python. This code takes internal energy ( $U$ ), specific molar volume ( $V$ ) and global mole fractions of each component ( $n$ ) and returns temperature ( $T$ ), pressure ( $P$ ), phase fraction ( $\phi$ ) and speed of sound ( $c$ ). The solver can also output other details, such as phase composition and phase densities if needed. For a non-reacting fully compressible CFD solver, the required fields are  $T$ ,  $P$ ,  $\phi$  and  $c$ . The input space is swept through to generate sufficient training data points which are then used to train the neural network using the open-source TensorFlow [18] platform. Currently, input space selection is performed by running a coarse mesh based direct estimation VLE simulation.

The difference in the CFD algorithm for direct estimation of VLE solutions and ANN-based estimation of VLE flash can be seen in Fig. 3. The ANN-based estimation greatly simplifies the calculations and hence reducing the computational overhead of the VLE calculations by moving them offline. This method also moves any possible failure points in the VLE algorithm offline to guarantee the robustness of the VLE-based CFD simulation. Finally, the ANN-based VLE method is designed as a plug-and-play system, and hence can directly convert any fully compressible

CFD code into a real-fluid EOS and VLE-based fully compressible CFD code with minimal coding effort (just need to code up an interface).



**Fig. 4 ANN Configuration (hidden layer nodes are only for representation).**

The preferred ANN structure is shown in Fig. 4. The training and development of ANN aided VLE models are performed on Python using TensorFlow [18]. TensorFlow is an end-to-end open-source machine learning (ML) platform which provides an easy-to-use application programming interface (API) to train custom models. Trained models can be saved in a TensorFlow saved model format, which can then be used to run inference (i.e., make predictions) using the learned weights and biases. The model is defined using 4 hidden layers with 32 nodes in each layer and all the layers use the ‘sigmoid’ activation function. The selection of the number of layers, nodes per layer, and activation functions is determined through a systematic process involving sensitivity tests on the ANN model for these hyperparameters. The data set used for training is divided into training, cross-validation (1-5%), and test (1-5%) sets. The data is normalized using the mean and standard deviation of the entire data set before training, as shown in Eq. (16):

$$X_{transform} = \frac{X - \bar{X}}{\sigma_X} \quad (16)$$

Here,  $X$  represents the variable,  $\bar{X}$  represents the mean value and  $\sigma_X$  represents the standard deviation. All the inputs and outputs of the ANN, as shown in Fig. 4, are normalized, as shown in Eq. (16).

This normalization ensures individual inputs do not dominate the training process, as the orders of magnitudes of the inputs can be significantly different. For example, the internal energy can be in the orders of  $10^3 - 10^5$  but mole fractions are always bounded by 0 and 1. The outputs are also normalized in a similar way and the mean and standard deviation values are used to reconstruct the actual values for post-prediction

Since OpenFOAM is a C++ based code and the ANN models are trained using python for the ease of usage of machine learning tools, an application programming interface (API) is required to use the python trained models in the C++ code. First option available was to directly use the TensorFlow C api in OpenFOAM, but the usage of a TensorFlow SavedModel resulted in very slow inference speeds and was not suitable for CFD simulation. The second option was to use the python api in OpenFOAM by linking the python interpreter with C++. This has performance issues: the interpreter would have to be called every single time the model is run resulting in very poor run-time performance. The preferred direction was to use machine learning production methods by employing Microsoft’s ONNX [19]. After training, the TensorFlow model is converted to an ONNX model [19] to make use of the highly optimized ONNX run-time functionality [20]. The ONNX run-time provides almost a 200 times speed-up over running inference directly using TensorFlow. The ONNX saved model is coupled with OpenFOAM to perform the VLE estimation at the end of each time step.

### C. CFD Flow Solver Formulation

The equation of mass, momentum (neglecting body force), and energy conservation, together with component transport can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0; \quad (17)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau}; \quad (18)$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho E \mathbf{u}) = -\nabla \cdot (p \mathbf{u}) + \nabla \cdot (\mathbf{u} \cdot \boldsymbol{\tau}) + \nabla \cdot \mathbf{q}; \quad (19)$$

$$\frac{\partial \rho Y_s}{\partial t} + \nabla \cdot (\rho Y_s \mathbf{u}) = -\nabla \cdot \mathbf{j}_s, \quad s = 1, \dots, n_s; \quad (20)$$

where  $\rho$  is the mixture density,  $p$  is the pressure,  $\mathbf{u}$  is the velocity,  $\boldsymbol{\tau}$  is the viscous stress tensor,  $\mathbf{q}$  is the heat flux,  $E = e + \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$  is the total energy. In Eq. (20),  $Y_s$  and  $\mathbf{j}_s$  are the mass fraction and diffusion mass flux respectively, while  $n_s$  is the total number of species.

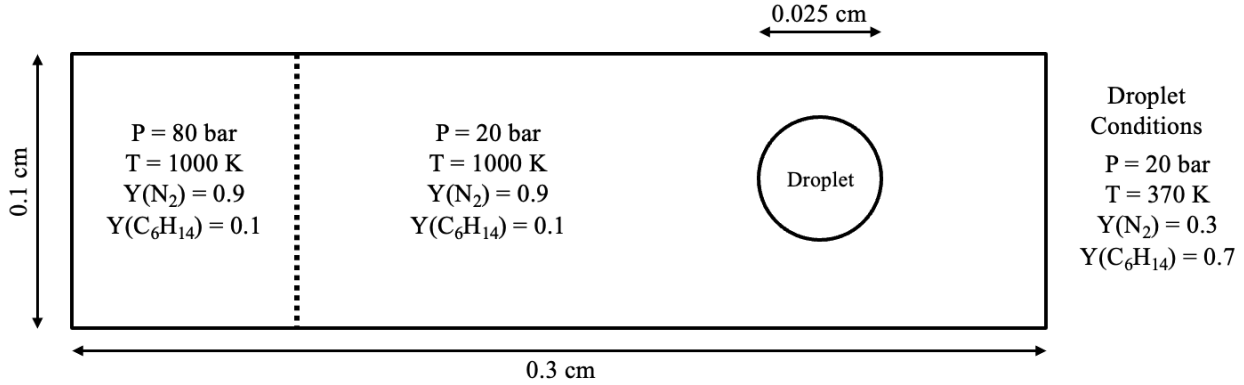


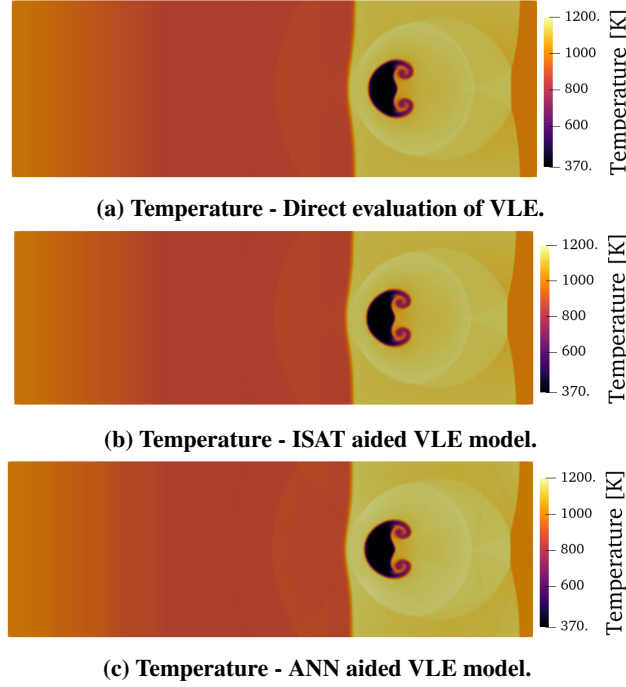
Fig. 5 Two-component shock-droplet interaction domain (1024x256 grid points).

## III. Results

### A. 2D Shock Droplet Interaction - Two Components

The simplest shock-droplet interaction case is a two-component system. In this case, a droplet primarily of  $\text{C}_6\text{H}_{14}$  at 20 bar and 370 K is placed in an environment primarily of  $\text{N}_2$  which is also at 20 bar but 1000 K. In order to trigger the propagation of a shock wave, a high-pressure region is initialized at 80 bar on the left side of the domain. The interfacial mass fraction of  $\text{N}_2$  is initialized using  $\tanh(x/\omega)$ , where  $\omega = 1 \times 10^{-6}$ . Similarly, the  $\tanh(x/\omega)$  is also used to define all the field values at the interface. The details regarding the domain are shown in Fig. 5. As indicated in the domain description, a small quantity of  $\text{C}_6\text{H}_{14}$  (mole fraction of  $\text{C}_6\text{H}_{14} = 0.1$ ) is introduced into the background fluid to enhance the multi-component effects.

Within the droplet, significant quantities of both  $\text{C}_6\text{H}_{14}$  and  $\text{N}_2$  are intentionally present, selected to create dynamic transcritical mixing states. This deliberate choice is made to challenge the convergence of the Vapor-Liquid Equilibrium (VLE) calculations, as it is generally more straightforward to achieve convergence when one component dominates (with a mass fraction exceeding 0.9). The 2D computational domain is discretized using a grid consisting of 1024 points in the x-direction and 256 points in the y-direction. On all four boundary faces, zero-gradient boundary conditions are applied for all fields. To ensure numerical stability, a maximum Courant-Friedrichs-Lewy (CFL) number of 0.1 is enforced during the time integration. The simulation is conducted over a total flow time of 2 microseconds, allowing sufficient time for the shock to completely traverse the droplet. A comprehensive evaluation of the results is undertaken, involving both qualitative and quantitative comparisons between direct evaluation, the *in situ* adaptive tabulation (ISAT)-aided



**Fig. 6 Temperature comparison after a flow time of  $2 \mu s$ .**

VLE model and the ANN aided VLE model. For additional insights into the ISAT methodology, readers are referred to the works of Zhang et al. [4, 21].

In Figure 6, a comparison of temperature contours for the shock-droplet interaction case at a flow time of 2 microseconds is presented. Notably, there is a remarkable agreement between the three methods in terms of temperature distribution. Additionally, the ANN-aided VLE model effectively captures all the reflected shocks, demonstrating its capability in accurately simulating the complex dynamics of the flow.

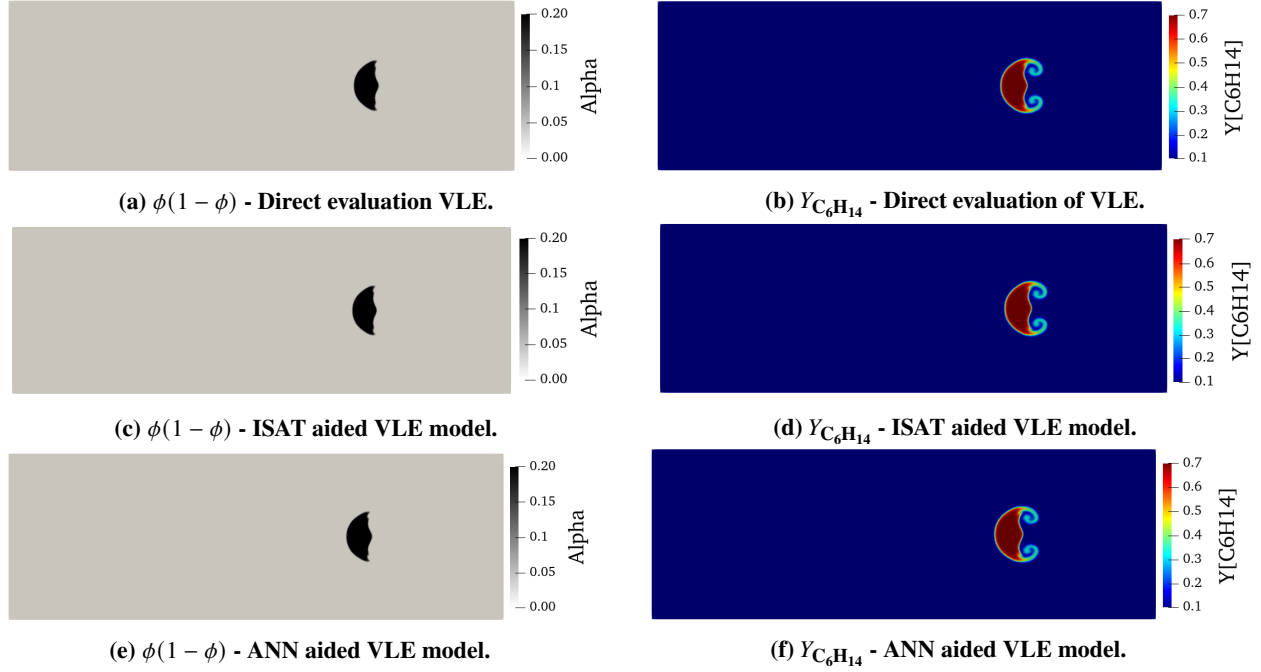
Figure 7 provides a comparison of the field  $\alpha = \phi(1 - \phi)$ , where  $\phi$  denotes the vapor fraction (based on the mole fraction of total gas phase moles to total moles). This field is employed to represent the two-phase region, as  $\alpha \neq 0$  only when  $\phi \neq 0$  and  $\phi \neq 1$ . Furthermore, it serves to delineate the boundaries of the droplet. Notably, within the droplet,  $\alpha$  is non-zero due to the presence of  $N_2$ , and the ANN-aided VLE model accurately captures this effect. This capability underscores the model's accuracy in simulating the intricate details of the flow field. Figure 7 also presents a comparison of the mass fraction of  $C_6H_{14}$  between the two methods. Remarkably, the shapes of the droplets are closely aligned, indicating a high degree of agreement in the evaluation of density and pressure between the ANN-aided VLE, direct evaluation and ISAT-aided VLE models. However, a slight disparity in the wake region of the droplet is noticeable, and this can be attributed to the presence of pressure oscillations resulting from the usage of the fully conservative (FC) scheme in all the three methods.

In Figure 8, a comparison of field values along the domain centerline is presented. This includes temperature, vapor fraction (based on moles of each phase), and the mass fraction of  $C_6H_{14}$  after a flow time of  $2 \mu s$ . Notably, an excellent agreement is observed between both methods for all these fields, reaffirming the consistency and accuracy of the ANN-aided VLE model in capturing essential flow characteristics.

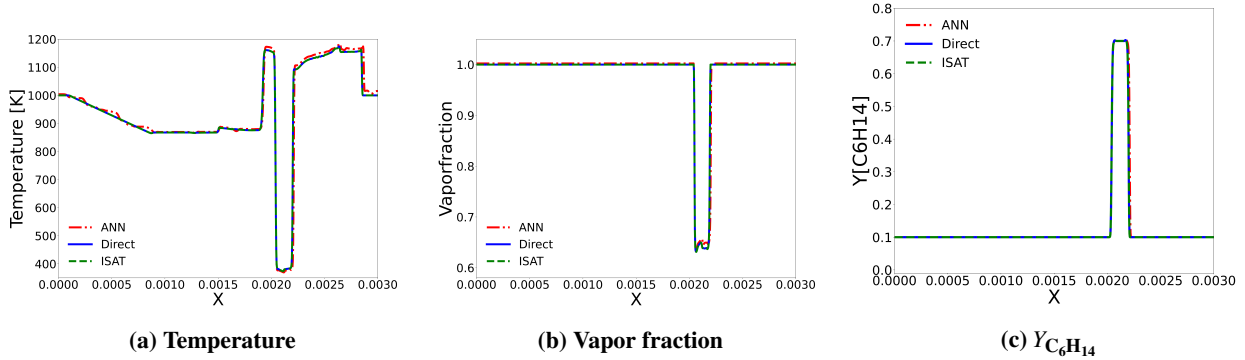
**Table 1 Comparison of total run time for two-component shock-droplet interaction (256 CPU cores).**

| Direct evaluation | ISAT aided model | ANN aided model |
|-------------------|------------------|-----------------|
| 410 s             | 351 s            | 301 s           |

The overall run-time comparison between the two models for the two-component shock-droplet interaction, when utilizing 256 CPU cores, is presented in Table 1. Notably, the ANN-aided VLE model outperforms both direct VLE evaluation and the ISAT-aided VLE model in terms of computational speed. Specifically, it achieves a 1.36x speed-up



**Fig. 7**  $\phi(1 - \phi)$  and  $Y_{C_6H_{14}}$  comparison after a flow time of  $2 \mu s$ .



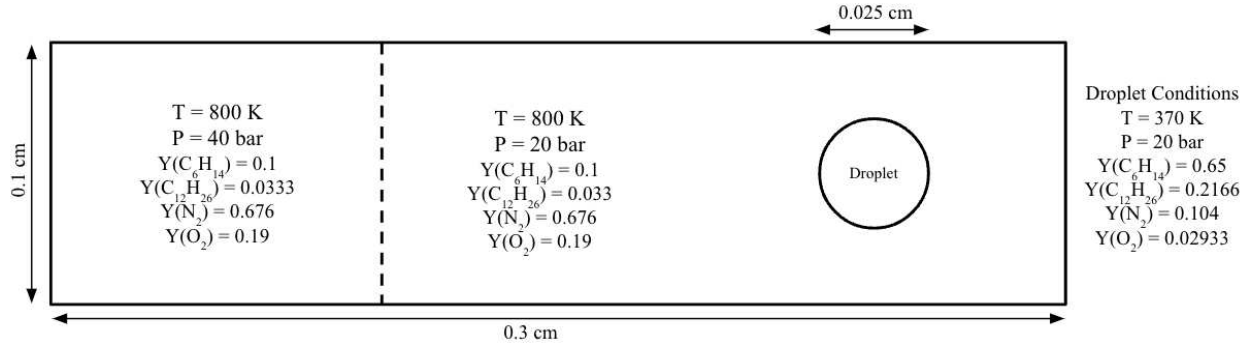
**Fig. 8** Centerline comparison after a flow time of  $2 \mu s$ .

compared to direct VLE estimation and a 1.17x speed-up compared to the ISAT-aided VLE model when considering the overall CPU time.

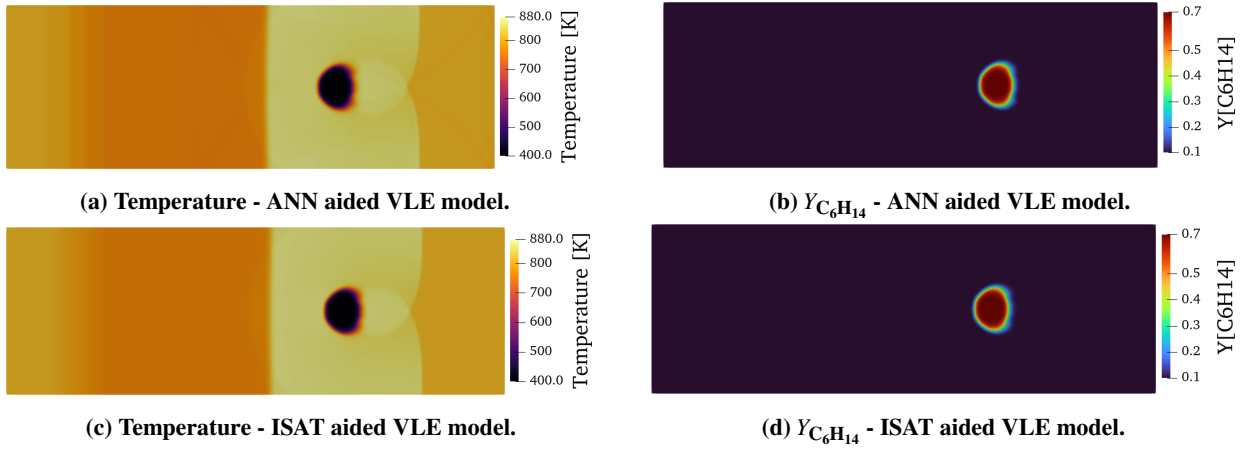
## B. 2D Shock Droplet Interaction - Four Components

In the previous section, the model's capabilities were demonstrated and validated against the ISAT-aided VLE model [21] in the context of a two-component shock-droplet interaction case. In this section, we delve into the intricacies of a four-component system, exemplifying the advantages of this approach in efficiently and robustly handling complex multi-component systems especially as it is well-acknowledged that the computational cost of VLE calculations tends to escalate exponentially as the number of components increases, as evidenced by Fathi et al. [5].

Specifically, in this section, we consider a multi-component fuel droplet comprising  $C_6H_{14}$  and  $C_{12}H_{26}$  in a 2:1 ratio (mole fraction) at a temperature of 370 K, suspended in an air environment at 800 K and 20 bar. The shock is initiated within the air domain, generated by a high-pressure region of 40 bar on the left. The configuration of the computational domain is depicted in Fig. 9.



**Fig. 9 Four-component shock-droplet interaction domain (512x128 grid points).**



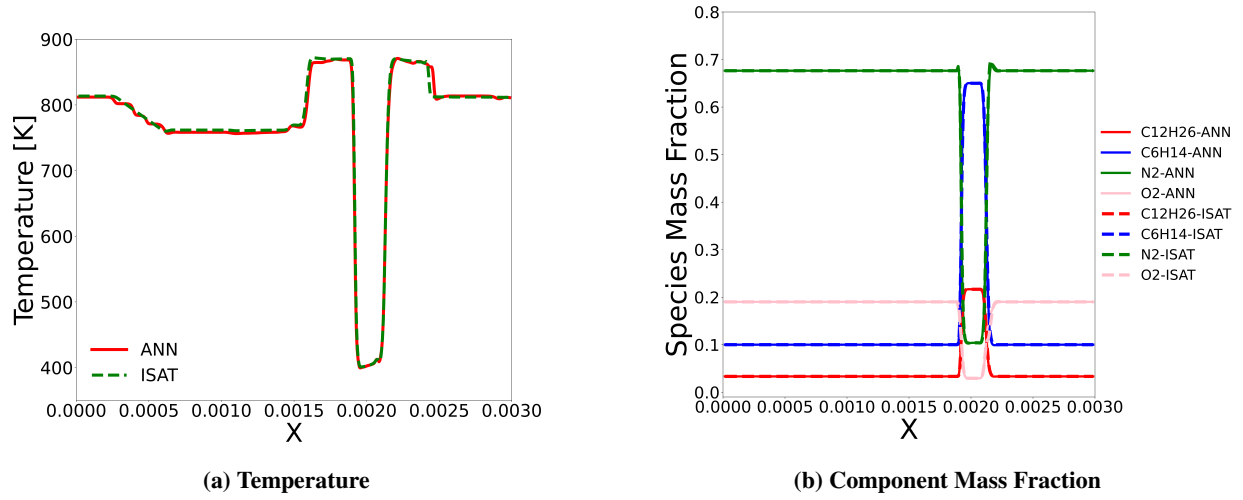
**Fig. 10 Temperature and  $Y_{C_6H_{14}}$  comparison after a flow time of  $2 \mu s$  for the four-component shock-droplet interaction (ANN vs. ISAT).**

The temperature contour comparison between the two methods is shown in Fig. 10. As seen in the previous case, the ANN aided VLE model captures the peak temperatures as well as reflected shock waves accurately (when compared to the ISAT model) even for the multi-component (more than 2 components) scenarios. The pressure oscillations occurring in a two-phase system with real-fluid equation of states (EOS) while solving the conservative formulation of the governing equations are shown clearly in Zhang et al. [21], and usually are mitigated using quasi-conservative methods, such as double flux schemes developed by Abgrall and Karni [22] and Ma et al. [23]. The current study retains the conservative formulation, and hence pressure oscillations are present in the CFD results of both the ISAT aided VLE model and the ANN aided VLE model. Therefore, considering the pressure oscillation uncertainty, the temperature contours comparison showcases a good agreement between the two models.

The mass fraction comparison of  $C_6H_{14}$  is shown in Fig. 10. The contours depict a very good agreement between the two models, with accurate capturing of both component mass fraction field as well as droplet shape. The shape of the droplet is different from that seen in Fig. 7 (two-component system) as the initialization pressure (40 bar) of the shock is lower than the two-component case (80 bar) and the density of the droplet is higher due to addition of  $C_{12}H_{26}$ . However, the trained ANN aided VLE model can still accurately predict the result.

Domain centerline comparison for the temperature as well as all the component mass fractions ( $C_6H_{14}$ ,  $C_{12}H_{26}$ ,  $N_2$ ,  $O_2$ ) are shown in Fig. 11. Temperature shows a very accurate match and all the component also show excellent accuracy when compared to the ISAT aided VLE model.

The major advantage produced by this ANN aided VLE model is the computational speed-up achieved when attempting to calculate VLE solutions for multi-component systems (more than two components). As shown by Fathi et al. [5], the computational costs exponentially increase with the increase in the number of components, and the chances of divergence also increase. The overall computational speed-up achieved by using the ANN aided VLE model in comparison to direct evaluation and ISAT aided VLE model is shown in Table 2. The speed-up factor achieved here



**Fig. 11 Centerline comparison after a flow time of  $2 \mu\text{s}$  for the four-component system (ANN vs. ISAT).**

(overall CPU time) is about 1.6x times speed up compared to direct evaluation and 1.4x compared to ISAT, even when using fewer points (512x128) compared to the two-component case (1024x256), showcasing the advantages of the ANN aided VLE model when more than two components are used in the system.

**Table 2 Total run time comparison for the four-component shock-droplet interaction case.**

| Direct evaluation | ISAT aided VLE model | ANN aided VLE model |
|-------------------|----------------------|---------------------|
| 156 s             | 136 s                | 96 s                |

## IV. Conclusions

In this study, an artificial neural network (ANN) aided Vapor-Liquid Equilibrium (VLE) model was introduced to address both the robustness and computational speed challenges associated with VLE calculations, especially in fully compressible conservative systems. VLE calculations are known to be computationally expensive, and they face robustness issues, particularly when energy-based flash solvers are required to converge with temperature-based flash solvers.

The ANN approach leverages the capacity of neural networks to perform nonlinear regression, enabling the prediction of outputs for given inputs through multiple neural nodes and layers. This approach provides an efficient way to store and estimate data, reducing the reliance on traditional multidimensional tables while using fewer points to learn the relationship between inputs and outputs. The ANN model was trained using TensorFlow in Python, employing normalized data to train a fully connected multi-layer perceptron model with four layers and 32 nodes per layer. The trained ANN model was then saved and converted to the ONNX format, which was subsequently coupled with a fully compressible central upwind-based solver developed in C++ using OpenFOAM.

Two cases were investigated to showcase the capabilities of the ANN-aided VLE model compared to both direct evaluation and the state-of-the-art In Situ Adaptive Tabulation (ISAT) aided VLE model. First, the two-component shock-droplet interaction case ( $\text{N}_2$  and  $\text{C}_6\text{H}_{14}$ ) demonstrated good qualitative and quantitative agreement between the three methods, with the ANN-aided model exhibiting a 1.36x speed-up against the direct evaluation method and a 1.17x speed-up against ISAT. A more challenging four-component case ( $\text{C}_6\text{H}_{14}$ ,  $\text{C}_{12}\text{H}_{26}$ ,  $\text{N}_2$ ,  $\text{O}_2$ ) validated the model's performance in scenarios with an increased number of components. Here, the ANN model showed good agreement with the ISAT model and a 1.4x speed-up in total runtime. This observation shows the potential speed up achievable while utilizing multiple components (more than 2) while employing the ANN model.

In conclusion, the ANN-aided VLE model shows promise in addressing the robustness and computational efficiency

challenges faced by VLE-based compressible flow solvers. It offers a computational speed-up compared to the state-of-the-art ISAT-aided VLE model and demonstrates good performance in scenarios with varying degrees of mixing between components while retaining parallel scaling.

### Acknowledgments

S. Yang gratefully acknowledges the support from the Office of Naval Research (ONR) grant under Award No. N00014-22-1-2287 under the supervision of project monitor Dr. Steven Martens, and the National Science Foundation (NSF) grant under Award No. CBET 2023932. The authors gratefully acknowledge the computing resources provided by the Minnesota Supercomputing Institute (MSI).

### References

- [1] Yang, V., "Modeling of supercritical vaporization, mixing, and combustion processes in liquid-fueled propulsion systems," *Proceedings of the Combustion Institute*, Vol. 28, No. 1, 2000, pp. 925–942.
- [2] Van Konynenburg, P., and Scott, R., "Critical lines and phase equilibria in binary van der Waals mixtures," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, Vol. 298, No. 1442, 1980, pp. 495–540.
- [3] Zhang, H., Yi, P., and Yang, S., "Multicomponent Effects on the Supercritical CO<sub>2</sub> Systems: Mixture Critical Point and Phase Separation," *Flow, Turbulence and Combustion*, Vol. 109, No. 2, 2022, pp. 515–543.
- [4] Zhang, H., and Yang, S., "Numerical investigation of high-pressure transcritical shock-droplet interaction and mixing layer using VLE-based CFD accelerated by ISAT," *AIAA SCITECH 2023 Forum*, 2023, p. 1857.
- [5] Fathi, M., and Hickel, S., "Rapid multi-component phase-split calculations using volume functions and reduction methods," *AIChE Journal*, Vol. 67, No. 6, 2021, p. e17174.
- [6] Yue, Z., Zhu, H., Wang, C., Li, Z., Wang, H., Yao, M., and Reitz, R. D., "Artificial neural network models for phase equilibrium predictions under engine trans/supercritical spray conditions," *Fuel*, Vol. 339, 2023, p. 127425.
- [7] Michelsen, M. L., "Multiphase isenthalpic and isentropic flash algorithms," *Fluid phase equilibria*, Vol. 33, No. 1-2, 1987, pp. 13–27.
- [8] Rachford Jr, H., Rice, J., et al., "Procedure for use of electronic digital computers in calculating flash vaporization hydrocarbon equilibrium," *Journal of Petroleum Technology*, Vol. 4, No. 10, 1952, pp. 19–3.
- [9] Saha, S., and Carroll, J. J., "The isoenergetic-isochoric flash," *Fluid phase equilibria*, Vol. 138, No. 1-2, 1997, pp. 23–41.
- [10] Peng, D.-Y., and Robinson, D. B., "A new two-constant equation of state," *Industrial & Engineering Chemistry Fundamentals*, Vol. 15, No. 1, 1976, pp. 59–64.
- [11] Yi, P., Yang, S., Habchi, C., and Lugo, R., "A multicomponent real-fluid fully compressible four-equation model for two-phase flow with phase change," *Physics of Fluids*, Vol. 31, No. 2, 2019, p. 026102.
- [12] Wilson, G. M., "Vapor-liquid equilibrium. XI. A new expression for the excess free energy of mixing," *Journal of the American Chemical Society*, Vol. 86, No. 2, 1964, pp. 127–130.
- [13] Tudosco, P., and Menon, S., "Analytical framework for real-gas mixtures with phase-equilibrium thermodynamics," *The Journal of Supercritical Fluids*, Vol. 164, 2020, p. 104929.
- [14] Chung, T. H., Ajlan, M., Lee, L. L., and Starling, K. E., "Generalized multiparameter correlation for nonpolar and polar fluid transport properties," *Industrial & engineering chemistry research*, Vol. 27, No. 4, 1988, pp. 671–679.
- [15] Kee, R. J., Rupley, F. M., Meeks, E., and Miller, J. A., "CHEMKIN-III: A FORTRAN chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics," Tech. rep., Sandia National Lab.(SNL-CA), Livermore, CA (United States), 1996.
- [16] Fuller, E. N., Schettler, P. D., and Giddings, J. C., "New method for prediction of binary gas-phase diffusion coefficients," *Industrial & Engineering Chemistry*, Vol. 58, No. 5, 1966, pp. 18–27.
- [17] Takahashi, S., "Preparation of a generalized chart for the diffusion coefficients of gases at high pressures," *Journal of Chemical Engineering of Japan*, Vol. 7, No. 6, 1975, pp. 417–420.

- [18] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” , 2015. URL <https://www.tensorflow.org/>, software available from tensorflow.org.
- [19] Bai, J., Lu, F., Zhang, K., et al., “ONNX: Open Neural Network Exchange,” <https://github.com/onnx/onnx>, 2019.
- [20] Developers, O. R., “ONNX Runtime,” <https://onnxruntime.ai/>, 2021.
- [21] Zhang, H., Srinivasan, N., and Yang, S., “In Situ Adaptive Tabulation of Vapor-Liquid Equilibrium Solutions for Multi-Component High-Pressure Transcritical Flows with Phase Change,” <https://dx.doi.org/10.2139/ssrn.4534330>, 2023.
- [22] Abgrall, R., and Karni, S., “Computations of compressible multifluids,” *Journal of computational physics*, Vol. 169, No. 2, 2001, pp. 594–623.
- [23] Ma, P. C., Lv, Y., and Ihme, M., “An entropy-stable hybrid scheme for simulations of transcritical real-fluid flows,” *Journal of Computational Physics*, Vol. 340, 2017, pp. 330–357.