Data-driven regression of thermodynamic models in entropic form using physics-informed machine learning

Bunschoten, Evert; Cappiello, Alessandro; Pini, Matteo

Important note
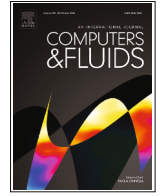To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Data-driven regression of thermodynamic models in entropic form using physics-informed machine learning

Evert Bunschoten [ID] *, Alessandro Cappiello [ID] , Matteo Pini [ID]

Propulsion & Power, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, The Netherlands

ARTICLE INFO

ABSTRACT

This article presents a data-driven method to evaluate thermodynamic properties of pure fluids and mixtures of fixed composition in the ideal- and nonideal thermodynamic states. Thermodynamic consistency is ensured by computing the fluid properties on the basis of the entropy potential and its first- and second- order derivatives, calculated with a physics-informed neural network. The computational performance of the method was investigated by implementing the resulting data-driven model in the open-source SU2 CFD software and by performing RANS simulations of the nonideal compressible flows through an organic Rankine cycle turbine cascade. Compared to using a multiparameter equation of state through a thermodynamic library coupled with SU2, the method was found to be 60 % more computationally efficient while maintaining high accuracy.

1. Introduction

Nonideal compressible fluid dynamics (NICFD) is a branch of fluid mechanics that studies fluid flows in the nonideal thermodynamic state. Such flows are relevant in many engineering applications [1]. Examples include the supersonic flow of dense gases in organic Rankine cycle turbines [2], supercritical flows in centrifugal compressors, fuel injectors of gas turbines, and regenerative cooling nozzles of rocket engines [3,4], as well as compressible high pressure liquid flows in turbo-pumps [5].

Properties of fluids whose thermodynamic state is in the nonideal thermodynamic region are usually calculated using multiparameter equation of state models (MEoS) in the Helmholtz free-energy form, namely fundamental equations from which primary (e.g., p, T) and secondary (e.g., c_p, c_v) fluid properties are derived as function of ρ and T. Such models are highly accurate, but they are computationally expensive when used in computer programs, for example in CFD solvers [1].

A more computationally efficient alternative to MEoS are EoS models constructed using data-driven methods (DDEoS). In DDEoS, the set of primary and secondary fluid properties, **Y**, calculated as function of two independent thermodynamic variables **x** (e.g., p, T) can be expressed as

$$\mathbf{Y} = \psi(\mathbf{x}), \tag{1}$$

where ψ denotes a generic inexpensive functional constructed from a set of linear or nonlinear basis functions, with coefficients determined based on thermodynamic datasets. Examples of functionals are polynomials

used to interpolate fluid properties based on tabular methods [6,7] or artificial neural networks used for regression [8–10].

Although high accuracy and computational efficiency can be achieved in the calculation of properties with the use of DDEoS, thermodynamic consistency can only be ensured if the functional ψ is derived from a thermodynamic potential Ψ. Thermodynamic consistency of an EoS model is such that all primary **P** (e.g., T) and secondary fluid properties **S** (e.g., c_p) are calculated by partial differentiation of the potential (e.g., Gibbs or Helmholtz free energy), namely, using its first- and second-order derivatives with respect to its *canonical* or *natural* variables, **x**. Denoting with J^{Ψ}, H^{Ψ} the Jacobian and Hessian of the potential

$$\mathbf{P} = \mathcal{P}\left(\Psi(\mathbf{x}), J^{\Psi}_{\mathbf{x}}\right) \tag{2}$$

$$\mathbf{S} = S\left(J^{\Psi}_{\mathbf{x}}, H^{\Psi}_{\mathbf{x}}\right) \tag{3}$$

where $\mathcal{P}$ and S denote the relations for calculating the primary and secondary fluid properties. For example, given the potential in terms of Helmholtz free energy, $a = a(T, v)$, with v the specific volume, the pressure, p, and the isochoric specific heat, c_v can be expressed as

$$p = -\left.\frac{\partial a}{\partial v}\right|_T \tag{4}$$

and

$$c_v = -T\frac{\partial^2 a}{\partial T^2}. \tag{5}$$

Consistency and accuracy of a DDEoS model can be ensured if the primary and secondary fluid properties are computed according to Eqs. (2),(3) and if the values of $\Psi(\mathbf{x})$ and its derivatives are accurate with respect to those provided by the reference MEoS. Attempts to develop consistent DDEoS for fluids in nonideal thermodynamic states were performed in several works. In [6], a method for generating thermodynamically consistent look-up tables (LuT) was presented. Thermodynamic consistency was prescribed by using interpolation functions of the fifth-order to reconstruct the Helmholtz free energy potential, where the coefficients of the interpolation function were calculated to satisfy the Maxwell relations. An alternative method was used in [10], where a physics-informed machine learning (PIML) approach was developed to compute the Helmholtz free energy as a function of temperature and density, while satisfying Maxwell relations by introducing learning bias. While the intended application of physics-informed neural networks (PINN) was to solve partial differential equations [11], the approach has also proven highly effective for developing thermodynamically DDEoS models. The use of PIML for the development of DDEoS was also demonstrated in [12] for modeling the properties of fluids based on the Mie potential and [13] presented a method to accurately model phase transition. Recent research on DDEoS has demonstrated the suitability of deep, dense multi-layer perceptrons (MLP) to evaluate the thermodynamic potential [10,12–14] over methods such as Gaussian Process Regression (GPR)[15,16]. Although GPR models can be trained using relatively few data points and provide additional uncertainty quantification, MLP models can be scaled more easily to larger data sets and are less computationally expensive to evaluate, which is a desirable feature for CFD applications.

The aforementioned works on consistent DDEoS [10,12–16] present EoS models in the Helmholtz free-energy form, with temperature and density as natural variables. However, consistent DDEoS based on density and internal energy can provide a significant reduction in computational cost in density-based CFD solvers, since the flow governing equations are solved in the so-called conservative form, from which ρ and e can be directly calculated. Two recent studies were concerned with the development of DDEoS using thermodynamic relations with density and internal energy as independent variables. A consistent equation of state model in entropy form, namely defining the model in terms of the less common *entropy potential*, was presented in [17]. The model was developed on the basis of a bilinear polynomial that was fitted to accurately provide the compressibility factor of siloxane MM. However, in this study, the secondary properties such as the speed of sound were calculated in a non-consistent manner using finite differences. The method was demonstrated to be suitable for direct numerical simulations (DNS). Similarly, a data-driven EoS was developed in the work documented in [9] on the basis of the entropy potential (EEoS). The Jacobian and Hessian of the potential, which were used to compute the thermodynamic state, were evaluated using a set of multitask neural networks (MTNNs). Although the method was successfully adopted for the numerical simulation of nonideal flows in nozzles with the SU2 open source CFD software [18], it was found to be not sufficiently robust when applied to the simulation of more complex flows, e.g., nonideal supersonic flows through turbine cascades.

This work extends the approach documented in [9] by proposing a method based on PINNs, similar to the methods presented in [10,12,13] to develop thermodynamically consistent EEoS models. The database of fluid properties needed for training the data-driven model was obtained by resorting to a thermodynamic software library [19]. The method was implemented into the open source CFD software SU2 [18] and its performance was verified through RANS simulations of nonideal supersonic flow of siloxane MM flow through a linear turbine cascade designed by the Propulsion & Power group of Delft University of Technology. The computational efficiency and accuracy of the method presented in this work were compared to those obtained by the method documented in [9], and by directly evaluating the properties by means of the reference MEoS through an external thermodynamic software library [19].

2. Methodology

2.1. Fundamental thermodynamic model in entropic form

Equation of state models in entropic form $s = s(\rho, e)$ are rarely considered, but they are highly suited for the calculation of thermodynamic properties of fluids in density-based NICFD solvers. As the natural variables associated with the entropy potential, namely density ρ and specific internal energy e directly derive from the conserved variables of the solver, most of the thermodynamic calls during the solution process do not involve expensive iterations to compute the fluid properties.

Assuming that the entropy potential is continuous and differentiable, the temperature, T, pressure, p, and speed of sound, c, are expressed in terms of Jacobian and Hessian with respect to density and internal energy through Eqs. (6)–(8)

$$T = \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \tag{6}$$

$$p = -\rho^2 \, T \left. \frac{\partial s}{\partial \rho} \right|_e \tag{7}$$

$$
\begin{aligned}
c^2 &= \left. \frac{\partial p}{\partial \rho} \right|_e - \left. \frac{\partial p}{\partial e} \right|_\rho \left. \frac{\partial s}{\partial \rho} \right|_e \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} = \\
&\quad -\rho \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \left\{ \left. \frac{\partial s}{\partial \rho} \right|_e \left[2 - \rho \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \frac{\partial^2 s}{\partial \rho \partial e} \right] + \rho \frac{\partial^2 s}{\partial \rho^2} + \right. \\
&\quad \left. -\rho \left[\frac{\partial^2 s}{\partial \rho \partial e} - \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \frac{\partial^2 s}{\partial e^2} \left. \frac{\partial s}{\partial \rho} \right|_e \right] \left. \frac{\partial s}{\partial \rho} \right|_e \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \right\}
\end{aligned}
\tag{8}
$$

In addition, the temperature and pressure derivatives with respect to ρ and e that are needed to calculate the remaining fluid properties read

$$\left. \frac{\partial T}{\partial e} \right|_\rho = - \left. \frac{\partial s}{\partial e} \right|_\rho^{-2} \frac{\partial^2 s}{\partial e^2} \tag{9}$$

$$\left. \frac{\partial T}{\partial \rho} \right|_e = - \left. \frac{\partial s}{\partial e} \right|_\rho^{-2} \frac{\partial^2 s}{\partial \rho \partial e} \tag{10}$$

$$\left. \frac{\partial p}{\partial e} \right|_\rho = -\rho^2 \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \cdot \left\{ -\left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \cdot \frac{\partial^2 s}{\partial e^2} \cdot \left. \frac{\partial s}{\partial \rho} \right|_e + \frac{\partial^2 s}{\partial \rho \partial e} \right\} \tag{11}$$

$$\left. \frac{\partial p}{\partial \rho} \right|_e = -\rho \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \cdot \left\{ \left. \frac{\partial s}{\partial \rho} \right|_e \cdot \left[2 - \rho \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \cdot \frac{\partial^2 s}{\partial \rho \partial e} \right] + \rho \frac{\partial^2 s}{\partial \rho^2} \right\} \tag{12}$$

The expressions for the fluid specific heat at constant volume, c_v, and at constant pressure, c_p, follow from Eqs. (9)–(12)

$$c_v = -\left(\frac{\partial^2 s}{\partial e^2} \right)^{-1} \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^2 \tag{13}$$

$$c_p = \left. \frac{\partial h}{\partial e} \right|_p \left(\left. \frac{\partial T}{\partial e} \right|_p \right)^{-1} \tag{14}$$

where

$$\left. \frac{\partial h}{\partial e} \right|_p = \left(1 + \rho^{-1} \left. \frac{\partial p}{\partial e} \right|_\rho \right) + \left(\left. \frac{\partial p}{\partial \rho} \right|_e \right)^{-1} \left. \frac{\partial p}{\partial e} \right|_\rho \left(\rho^{-2} p - \rho^{-1} \left. \frac{\partial p}{\partial \rho} \right|_e \right) \tag{15}$$

and

$$\left. \frac{\partial T}{\partial e} \right|_p = -\left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-2} \frac{\partial^2 s}{\partial e^2} + \left(\left. \frac{\partial p}{\partial \rho} \right|_e \right)^{-1} \left. \frac{\partial p}{\partial e} \right|_\rho \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-2} \frac{\partial^2 s}{\partial e \partial \rho} \tag{16}$$

Likewise, an analytical expression for the fundamental derivative of gas dynamics, Γ, is given by

$$\Gamma = 1 + \frac{\rho}{c} \left. \frac{\partial c}{\partial \rho} \right|_s = 1 + \frac{\rho}{c} \left[\left. \frac{\partial c}{\partial \rho} \right|_e - \left. \frac{\partial c}{\partial e} \right|_\rho \left. \frac{\partial s}{\partial \rho} \right|_e \left(\left. \frac{\partial s}{\partial e} \right|_\rho \right)^{-1} \right] \tag{17}$$

Analytical expressions for $\left. \frac{\partial c}{\partial \rho} \right|_e$ and $\left. \frac{\partial c}{\partial e} \right|_\rho$ can be derived from Eq. (8), and they omitted here for brevity.

Table 1

EEoS verification results: Root-Mean-Square error of the relative deviation between the thermodynamic terms as per the EEoS and HEoS evaluated by CoolProp.

	T	p	c	$\frac{\partial T}{\partial \rho}\big|_e$	$\frac{\partial T}{\partial e}\big|_\rho$	$\frac{\partial p}{\partial \rho}\big|_e$	$\frac{\partial p}{\partial e}\big|_\rho$	c_v	c_p	Γ
$\mathcal{O}(\log \epsilon)$[-]	−16	−15	−15	−16	−16	−15	−16	−16	−14	−16

Table 2

Fluid data grid parameters used to generate the reference data for training the PINN.

	min	max	N_p	distribution
Density (ρ)	$1.0 \times 10^{-1}\,\mathrm{kgm^{-3}}$	$3.0 \times 10^{2}\,\mathrm{kgm^{-3}}$	500	cosine
Energy (e)	$2.5 \times 10^{5}\,\mathrm{Jkg^{-1}}$	$5.5 \times 10^{5}\,\mathrm{Jkg^{-1}}$	500	linear

The presented theoretical model is intrinsically consistent and is valid for monocomponent fluids in single and two-phase, and for mixtures of fixed composition. In this study, however, only single-phase flows of siloxane MM are considered.

2.1.1. Model verification

To verify the correctness of the analytical expressions previously reported, the deviations between the LHS and the RHS of Eqs. (9)–(13) were evaluated by substituting all terms in the equations with those obtained by the multiparameter equation of state in Helmholtz form (HEoS) for a range of density and specific internal energy values corresponding to thermodynamic states of siloxane MM in the ideal and nonideal gas region.

The HEoS was evaluated using CoolProp [19] for approximately 1.6×10^5 fluid data points that ranged in density and internal energy between $1 \times 10^{-1}\,\mathrm{kgm^{-3}}$ and $3 \times 10^{2}\,\mathrm{kgm^{-3}}$ and $2.5 \times 10^{5}\,\mathrm{Jkg^{-1}}$ and $5.5 \times 10^{5}\,\mathrm{Jkg^{-1}}$. The relative root mean square error values ϵ between the LHS and the RHS of Eqs. (9)–(13) were computed and listed in Table Table 1. The values were found all close to machine accuracy, thus validating the theoretical framework of the EEoS.

2.2. Data-driven regression with physics-informed neural networks

In the work by the authors documented in [9], the Jacobian and Hessian terms of the entropy potential appearing in Eqs. (6)–(13) were provided by different MTNNs that were trained using data fitting. However, as observed in [10], the use of this approach can lead to inconsistencies in the calculation of the thermodynamic state. Although the thermodynamic state according to the MTNN model presented in [10] was evaluated with high accuracy, the Maxwell relations were not satisfied, indicating thermodynamic inconsistency.

Therefore, the thermodynamic consistency of the model was enforced by leveraging PINN, in which the analytical relations of the EEoS are embedded into the training process of a multilayer perceptron (MLP). This was achieved by indirectly applying physics-based constraints to the network output through adaptation of the loss function.

Fig. 1 illustrates the method by which the network hyperparameters (θ) were updated during the physics-informed machine learning (PIML) process. The network was defined as a deep, feed-forward MLP with two inputs corresponding to ρ and e and a single output for s. In step 1 of Fig. 1, s is calculated by the MLP on the basis of the density and specific internal energy of the batch of training data, ρ_b, e_b. The Jacobian, $J^s_{\rho,e}$, and Hessian, $H^s_{\rho,e}$, of the network output were retrieved using algorithmic differentiation (AD), where

$$J^s_{\rho,e} = \begin{bmatrix} \frac{\partial s}{\partial \rho}\big|_e \\ \frac{\partial s}{\partial e}\big|_\rho \end{bmatrix} \quad H^s_{\rho,e} = \begin{bmatrix} \frac{\partial^2 s}{\partial \rho^2} & \frac{\partial^2 s}{\partial \rho \partial e} \\ \frac{\partial^2 s}{\partial e \partial \rho} & \frac{\partial^2 s}{\partial e^2} \end{bmatrix} \tag{18}$$

In step 2 of Fig. 1, the vector of thermodynamic properties Y_p is calculated according to Eqs. (9)–(14) using the Jacobian and Hessian terms that were calculated in step 1.

The loss function of the training process was evaluated in step 3 of Fig. 1 and was defined as

$$\mathcal{L}_Y = \frac{1}{N_b} \sum_{}^{N_b} \left(\overline{Y}_p \left(J^s_{\rho,e}, H^s_{\rho,e} \right) - \overline{Y}_r \right)^2 \quad Y \in s, T, p, c^2, \tag{19}$$

where the loss value $\mathcal{L}_Y$ equates the mean square error of the scaled properties Y, computed according to the network prediction, Y_p, and the labeled data Y_r over a batch of N_b training data. The network was trained for 1000 epochs.

The loss function sensitivity with respect to the network hyperparameters $\nabla_\theta \mathcal{L}_Y$ was evaluated with AD and used together with the learning rate value of the current update step $r_{l,i}$ by the Adam stochastic gradient descent (SGD) algorithm [20] to update the network hyperparameters. Here, the value of the learning rate was calculated following an exponential decay function

$$r_{l,i} = r_{l,0} d_r^{i/N_d}, \tag{20}$$

where $r_{l,0}$ is the initial value of the learning rate, d_r the learning rate decay parameter, and N_d the number of decay steps

$$N_d = 10^{-3} N_{\mathrm{epoch}} N_Y \frac{N_{\mathrm{train}}}{N_b}, \tag{21}$$

where N_Y is the number of thermodynamic variables included in the loss function, which according to Eq. (19) was set to four. N_{train} is the number of data points in the training data set and N_{epoch} the number of epochs. The thermodynamic properties Y for which $\mathcal{L}_Y$ was evaluated were, alternatively, s, T, p and c^2 in each update step. Changing $\mathcal{L}_Y$ between $\mathcal{L}_s, \mathcal{L}_T, \mathcal{L}_p$ and $\mathcal{L}_{c^2}$ during the training process was found to result in higher prediction accuracy compared to using a more typical approach where $\mathcal{L}_Y$ is defined as the average value of $\mathcal{L}_s, \mathcal{L}_T, \mathcal{L}_p$ and $\mathcal{L}_{c^2}$.

2.3. Training data and network architectures

The accuracy and internal consistency of PINN were compared to those of MTNN by comparing the values of the first- and second-order derivatives of the entropy potential. Both types of networks were trained to provide the thermodynamic properties of siloxane MM in vapor phase.

Data used for training both types of networks were generated with HEoS using the Python wrapper of CoolProp [19] on an ensemble of thermodynamic states defined with density and specific internal energy according to the parameters listed in Table 2. In total, 1.6×10^5 unique thermodynamic states were generated, of which 80 % was used for training , 10 % for testing, and the remaining 10 % for validation.

The two MTNNs were trained using the method described in [9], in which the loss function was defined as the mean square error between the network output and the reference data for the Jacobian and Hessian of s with respect to ρ and e. The training method for the EEoS-MTNN method is elaborated further in [9]. One of the networks was trained on density values between $1.0 \times 10^{-1}\,\mathrm{kgm^{-3}}$ and $10.0\,\mathrm{kgm^{-3}}$, while the other was trained for density values between $10.0\,\mathrm{kgm^{-3}}$ and $300.0\,\mathrm{kgm^{-3}}$. The subdivision of the data into two batches was necessary to increase the network accuracy for predicting the Hessian in the ideal-gas thermodynamic region.

The number of nodes per hidden layer, N_H, and the learning rate parameters of the PINN and MTNNs are listed in Table 3.

The initial learning rate $r_{l,0}$, the learning rate decay parameter d_r, the mini-batch size N_b, the hidden layer activation function $\phi(x)$, and the number of neurons per hidden layer of the MTNNs were determined by optimization of hyperparameters, while those of the PINN were selected by trial and error. The hidden layer activation function for the PINN was chosen to be the exponential function to reduce the computational cost of evaluating the first- and second-order derivatives of the
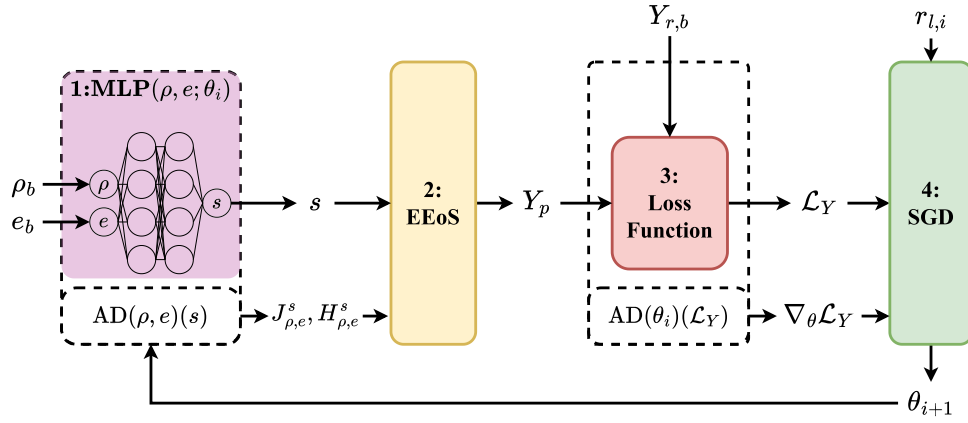
Fig. 1. Diagram illustrating the process of training the PINN to compute the thermodynamic state Q_p using the EEoS.

Table 3
Network architectures and learning rate parameters for the PINN and MTNNs used to evaluate the components of the entropy Jacobian and Hessian.

	PINN	MTNN $\rho \leq 10\,\mathrm{kgm^{-3}}$	MTNN $\rho > 10\,\mathrm{kgm^{-3}}$
N_H	[12,12]	[11,14]	[13,15,12]
$\log_{10}(r_{l,0})$	−3.000	−3.115	−3.072
d_r	9.8787e−01	9.8732e−01	9.250e−01
$\log_2(N_b)$	6	5	5
$\phi(x)$	exponential	swish	gelu
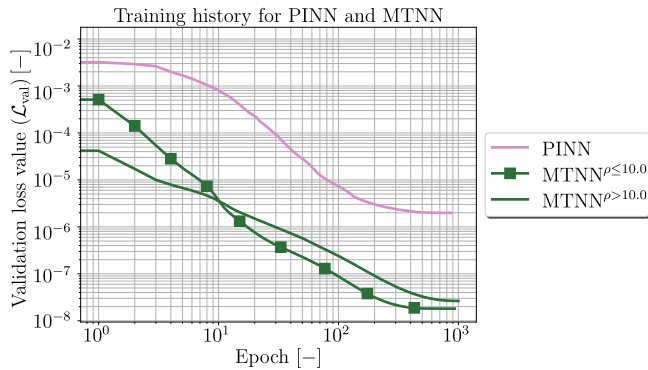
Fig. 2. Mean validation loss value trends of MTNNs and PINN during training.

Fig. 3. Process diagram for the computation of the thermodynamic state on the basis of the flow solutions of density and internal energy with the PINN and MTNN approach during the flow solution process.

network output during numerical simulations. This is elaborated upon in Section 2.4.

The convergence of the training process for both MTNNs and PINN is displayed in Fig. 2, which shows the mean loss function value $\overline{\mathcal{L}}$ computed with the set of validation data. The trend of the loss function of the PINN shown in Fig. 2 represents the mean loss value between the entropy, temperature, pressure, and speed of sound

$$\mathcal{L}_{\mathrm{PINN}} = \frac{1}{4}\left(\mathcal{L}_s + \mathcal{L}_T + \mathcal{L}_p + \mathcal{L}_{c^2}\right). \tag{22}$$

The difference between the loss values of the validation set and the test set after concluding the training process was 2.79 % for the PINN, and 0.94 % and 7.85 % for the MTNNs trained on the lower and higher density ranges, indicating that the level of over-fitting was negligible for both sets of networks.

2.4. Implementation of the neural networks in SU2

The data-driven EEoS described in this work was implemented in the open-source software suite SU2 [18] version 8.1.0 for the numerical simulation of nonideal compressible flows.
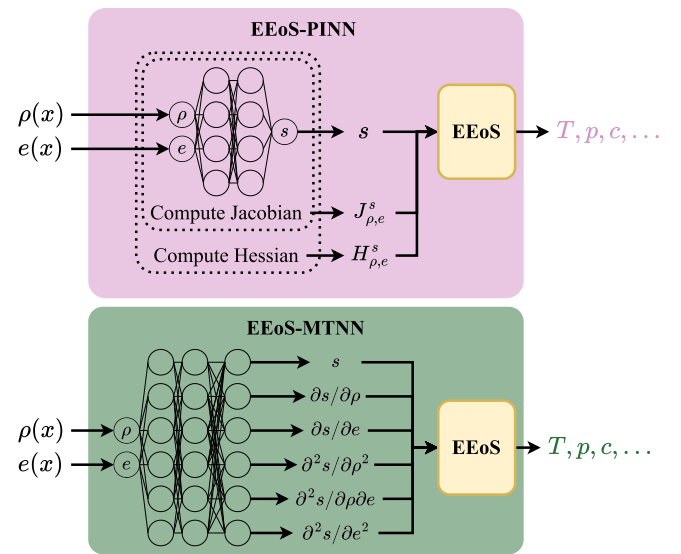
The process by which the thermodynamic state of the fluid is computed during the CFD solution process is visualized in Fig. 3. At each iteration of the flow solver, the local values of density $\rho(\vec{x})$ and specific internal energy $e(\vec{x})$ are passed to the MLP, coded in a separate solver class. The Jacobian and Hessian of the entropy potential are directly retrieved from the network output when using the MTNN approach, while they are analytically evaluated when using the PINN approach. These terms are then used to compute the thermodynamic properties according to the EEoS, which are then passed back to the flow solver.

For certain types of boundary conditions, as well as for the initialization of the flow solution, the state of the fluid must be calculated based on a pair of thermodynamic variables different from ρ and e. For such instances, the corresponding values for ρ and e are calculated through inverse regression using a Newton root-finding algorithm.

The analytical expressions of the Jacobian and Hessian terms of the PINN were derived as follows. In a deep, dense MLP with L layers, the output value y for the neuron j with hidden layer index i equals

$$y^{i,j} = \begin{cases} x_j^* & i = 0 \\ \phi\left(\sum_{k=0}^{N_{i-1}} w_{i,j,k}\, y^{i-1,k}\right) & i = 1, 2 \dots L \end{cases} \tag{23}$$

where x^* is the (scaled) network input, ϕ the hidden layer activation function, N_{i-1} the number of neurons in the previous layer, and w_i the matrix of weights connecting hidden layers i and $i-1$. The Jacobian of this neuron output with respect to input variable n is analytically expressed as

$$J_n^{i,j} = \phi'^{i,j} \sum_{k=0}^{N_{i-1}} w_{i,j,k} J_n^{i-1,k} \quad i = 1, 2 \dots L \tag{24}$$

where $\phi'^{i,j}$ is the analytical derivative of the hidden layer activation function. The Jacobian of the input layer, $J_n^{0,j}$ is a square, unity, diagonal matrix.

$$J_n^{0,j} = \begin{cases} 1, & \text{if } j = n, \\ 0, & \text{if } j \neq n \end{cases} \tag{25}$$

Adopting the same approach used for computing the analytical Jacobian, the Hessian of the neuron output with respect to the network inputs n and m is expressed as

$$H_{n,m}^{i,j} = \phi''^{i,j} \sum_{k=0}^{z_{i-1}} w_{i,j,k} J_n^{i-1,k} \sum_{k=0}^{z_{i-1}} w_{i,j,k} J_m^{i-1,k}$$
$$+ \phi'^{i,j} \sum_{k=0}^{z_{i-1}} w_{i,j,k} H_{n,m}^{i-1,k} \quad i = 1, 2 \dots L \tag{26}$$

where $\phi''^{i,j}$ is the analytical, second order derivative of the hidden layer activation function. The Hessian of the input layer is simply zero.

$$H_{n,m}^{0,j} = 0 \quad \forall j, n, m \tag{27}$$

The added computational cost of analytically evaluating the Jacobian and Hessian network output scales with the network size, as well as with the cost of evaluating the derivatives of the hidden layer activation function ϕ' and ϕ''. In order to reduce the computational cost of determining these quantities, the exponential function was used for the hidden layers, since

$$\phi(x) = e^x = \phi'(x) = \phi''(x) \tag{28}$$

As a result, the hidden layer activation function had to be evaluated only once for each neuron while computing the Jacobian and Hessian terms of the network. It was found to be very difficult to accurately train the network using the PIML method described in Section 2.2 when the weights were initialized randomly. This difficulty was caused from the unbounded nature of the exponential function and the fact that the network was trained using derivative information. This challenge was also reported in [12], where the stability of the training process was improved using a very small value for the learning rate and training for a large number of epochs. In this work, the stability of the PIML training process was improved by first training the network for 1000 epochs through data-fitting by including only s in the loss function shown in Eq. (19). After preconditioning the weights of the network by data fitting, the PIML training process illustrated in Fig. 1 was started while including T, p, and c^2 in the loss function.

3. Case study

The accuracy, consistency, and computational cost of the data-driven EEoS method presented in this work were evaluated by simulating the flow of siloxane MM through a linear turbine cascade and comparing the simulation results with those obtained by reference models. The blade geometry of the turbine cascade was designed at Delft University of Technology for the expansion of organic fluids in the ORCHID experimental facility [21] and will hereafter be referred to as the ORCHID linear cascade.

The geometry and boundary conditions of the ORCHID linear cascade, together with the computational domain, are illustrated in Fig. 4. The computational domain was discretized with the UMG2 mesh tool [22] by means of a hybrid grid composed of triangular elements for the main flow passage and quadrilateral cells for the refinement of

Table 4
ORCHID stator simulation boundary conditions.

Boundary	Type	Values
Inlet	p_t, T_t, non-reflective	$1.8 \times 10^6 \text{Pa}, 523\text{K}$
Outlet	p_s, non-reflective	$2 \times 10^5 \text{Pa}$
Blade	no-slip wall	-

the boundary layer in the near-wall region with a calculated maximum value $y+$ of 0.4 near the blade surface.

The boundary conditions applied in the numerical simulation are listed in Table 4. Non-reflecting boundary conditions have been used for the inlet and outlet boundaries to avoid spurious wave reflections that may result from the supersonic flow encountered at the outlet of the cascade. Additionally, an adiabatic non-slip condition has been imposed on the blade wall. Throughout the flow domain, the fluid expands from the highly dense vapor state ($Z \approx 0.5$) at the inlet to the nearly-ideal gas state ($Z \approx 0.95$) at the outlet, with Z being the compressibility factor defined according to Eq. (29):

$$Z = \frac{p}{R\rho T} \tag{29}$$

where R is the gas constant for siloxane MM with a value of $51.2 \text{Jkg}^{-1}\text{K}^{-1}$. The dynamic viscosity of the fluid and the Prandtl number for both laminar and turbulent flow were kept constant at $1.162 \times 10^{-5} \text{kgm}^{-1}\text{s}^{-2}$ and 1.230 respectively, corresponding to a Reynolds number of 5.5e7.

Fig. 5 shows the corresponding isentropic expansion process from the inlet to the outlet conditions, superposed to the contour of the compressibility factor in the reduced pressure and temperature diagram $(p/p_c, T/T_c)$, in which "cp" denotes the critical point of the fluid.

The numerical simulation was performed using the SU2 RANS solver with the SA turbulence model. Spatial gradients were computed using the weighted least-squares method and at each pseudo-time marching step the resulting linear system of equations was solved using the FGM-RES with a minimum threshold of 1e-5. Initially, the simulation was run for 1000 iterations using the first-order Roe convective scheme without spatial gradient reconstruction. To increase the stability of the solution, a gradual decrease in the outlet static pressure from the inlet stagnation pressure to the target outlet static pressure was imposed during the first 200 iterations. After completing 1000 iterations using the first-order scheme, the simulation was restarted with the JST convective scheme in order to converge to a second-order accurate solution.

The thermodynamic state of the fluid was calculated by the EEoS in SU2 using the PINN and MTNN models, referred to as EEoS-PINN and EEoS-MTNN in the following. The convergence rate, computational cost, and numerical solution obtained with both approaches were compared with those obtained by using the reference HEoS through an external thermodynamic library and the built-in Peng-Robinson cubic equation of state (CEoS).

4. Results

The accuracy and computational cost of the flow simulation within the ORCHID linear cascade obtained with the EEoS-PINN, the EEoS-MTNN and the HEoS were compared.

From left to right, the contours of the temperature and Mach number calculated with the HEoS, EEoS-PINN, and EEoS-MTNN models are shown in Fig. 6. As can be observed, the flow fields computed with HEoS and EEoS-PINN are qualitatively in agreement. In contrast, the flow fields of temperature and Mach number obtained with the EEoS-MTNN model show non-physical characteristics in the so-called base region near the trailing edge of the blade. The cause of such issue was initially attributed to a discontinuity in the thermodynamic state variables, as the continuity of output of the MTNNs was not enforced at $\rho = 10.0 \text{kgm}^{-3}$. However, it did not turn out to be the root cause of
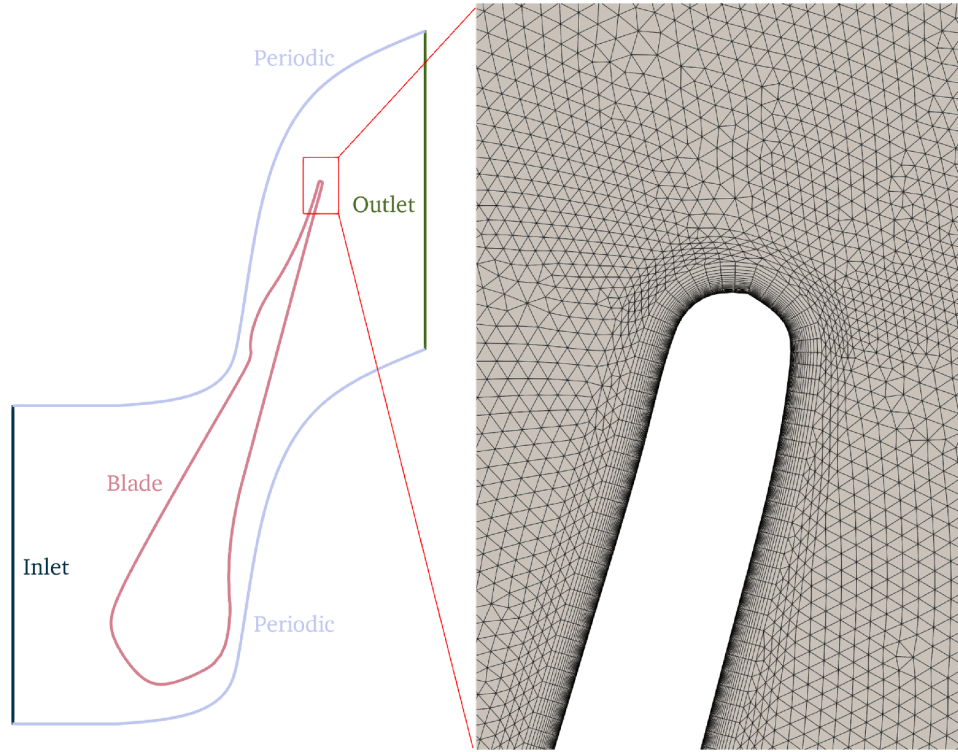
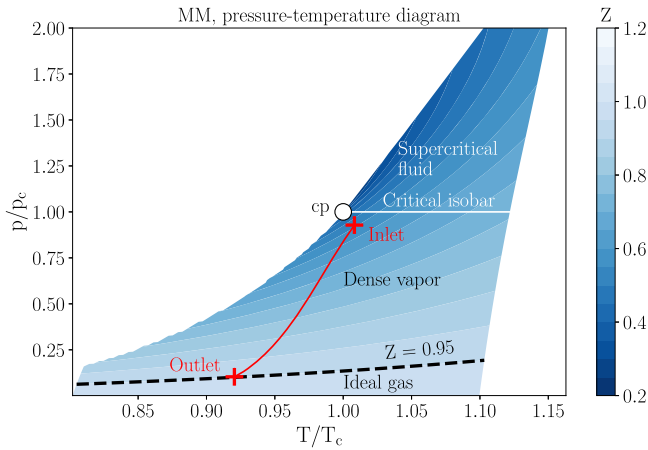**Fig. 4.** ORCHID linear cascade domain boundaries and trailing edge mesh topology.

Fig. 5. Fluid thermodynamic diagram of syloxane MM with boundary conditions.

Table 5
Measured, average iteration time and speed-up for each fluid model.

	HEoS	EEoS-MTNN	EEoS-PINN	CEoS
$t_{avg}[s]$	5.16e-02	3.64e-02	2.01e-02	7.24e-03
$\mathcal{O}(\sigma)[s]$	10^{-6}	10^{-5}	10^{-6}	10^{-5}
$t_{avg}/t_{avg,HEoS}$	1	7.05e-01	3.89e-01	1.40e-01

ulations performed with the HEoS and EEoS-PINN yielded an average CFL number of around 30, and the trends of the adaptive CFL almost completely overlap.

Although the flow solutions of the HEoS, CEoS, and EEoS-PINN fluid models converged at a similar rate, there was a significant difference among the fluid models in terms of computational cost. Table 5 lists the CPU time (t_{avg}), averaged over the last 500 iterations for each flow simulation and the speed-up with respect to the simulation performed with the HEoS fluid model. As expected, the flow simulations performed with the EEoS-MTNN and EEoS-PINN models were found to be less expensive than those carried out with the reference HEoS model. Specifically, the cost of the EEoS-PINN simulation was about 2.6 times lower, while that of the EEoS-MTNN was only 1.3 times lower. The computational cost was found to be about 7.1 and 2.8 times lower than that achieved with the HEoS and EEoS-PINN models. This discrepancy in the computational cost between the CEoS and EEoS-PINN model was due to the higher number of arithmetic operations required to evaluate the properties by the data-driven model. As the computational cost of the PINN scales quadratically with the number of nodes in the hidden layers of the network, the computational cost of the EEoS-PINN model may be reduced by optimizing the hidden layer architecture of the network.

The accuracy of the flow solution obtained using EEoS-PINN was subsequently assessed. The normalized static pressure, $p/p_{t,Inlet}$ and the normalized static temperature, $T/T_{t,Inlet}$, along the blade surface and outlet boundaries are illustrated in Fig. 8 and Fig. 9. The Mach number distribution at the outlet boundary for the three converged
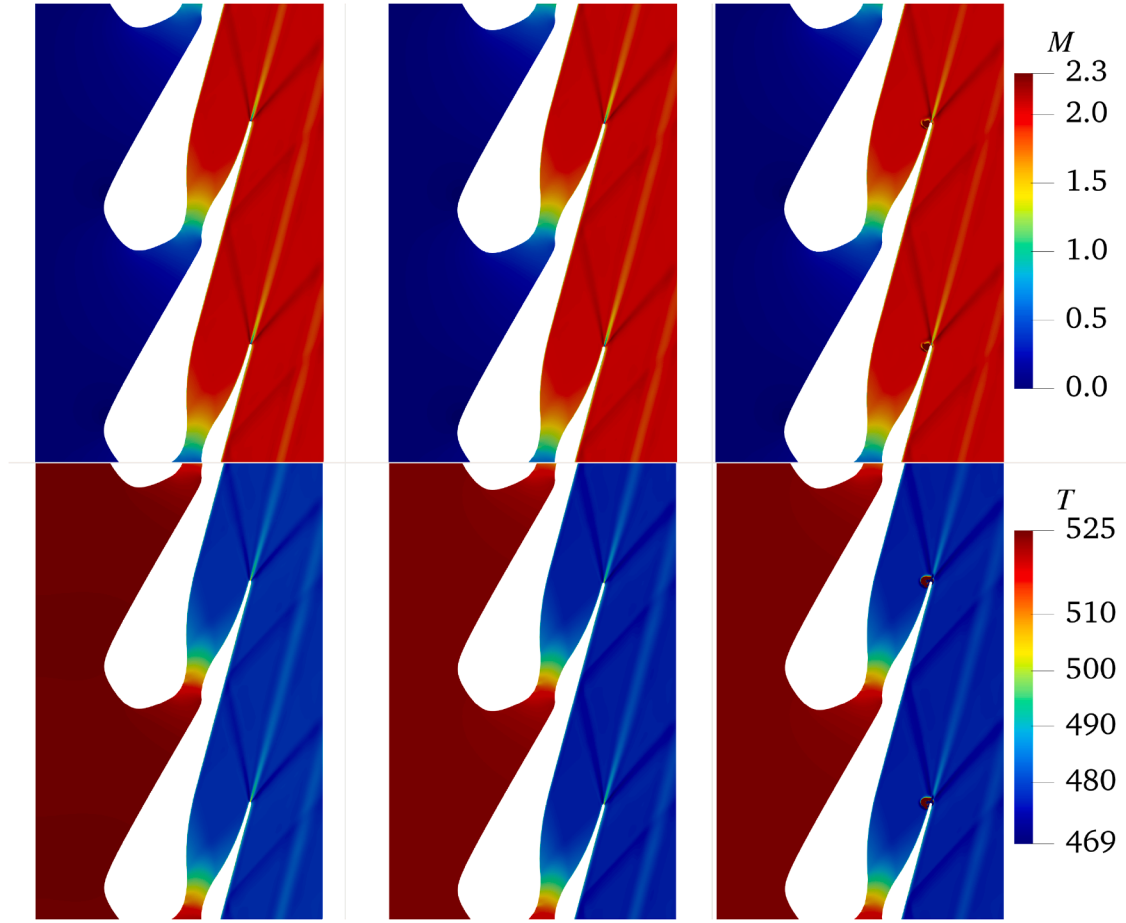
the divergence, as the fluid density in this region is significantly lower $\left(\rho \approx 2.5 \mathrm{kgm}^{-3}\right)$, and no discontinuities in the thermodynamic state were observed in the solution where $\rho = 10.0 \mathrm{kgm}^{-3}$. By closer inspection of the results, it was found that the fluid thermodynamic state in the base region is such that the flow behaves as an ideal gas ($Z > 0.95$), where the magnitudes of the Jacobian and Hessian of the entropy potential vary significantly, making it very difficult for MTNN to accurately evaluate these properties, as was also found in [9].

The convergence trends of the simulations are shown on the left side of Fig. 7 in terms of density residuals for the four thermodynamic models. The flow solutions obtained with the HEoS, CEoS, and EEoS-PINN converged at approximately the same rate, while the calculations diverged when the EEoS-MTNN was used.

The evolution of the adaptive CFL number shown on the right side of Fig. 7 indicates the robustness of the numerical simulation. The sim-

Fig. 6. Solutions for Mach number (top) and temperature (bottom) of the numerical simulation of the ORCHID linear cascade obtained with the HEoS fluid model (left), the EEoS-PINN fluid model (middle), and the EEoS-MTNN fluid model (right).
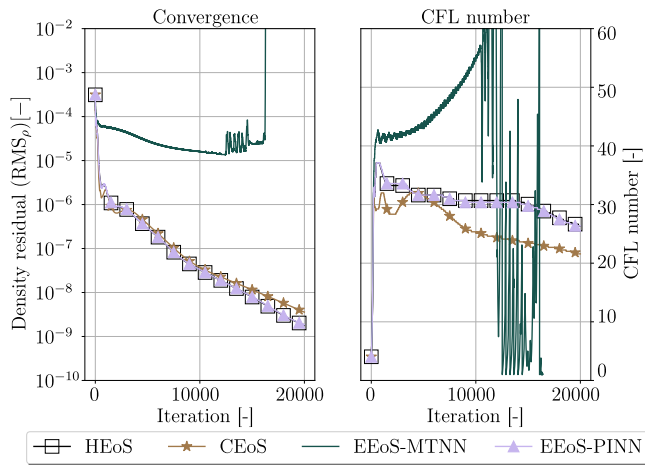

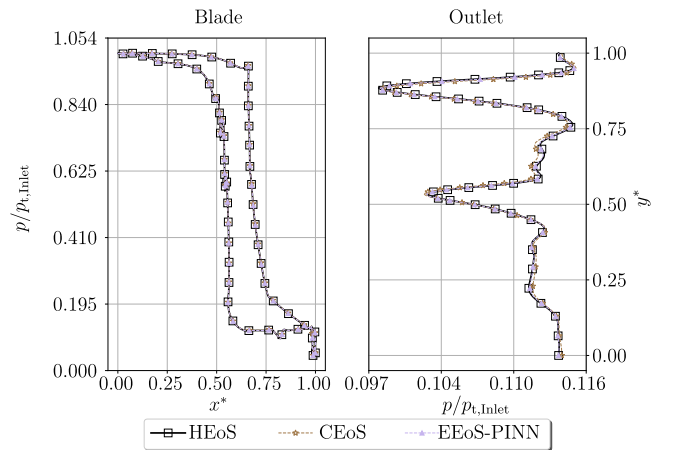**Fig. 7.** Density residual and average CFL number for HEoS and EEoS simulations of the ORCHID linear cascade.

Fig. 8. Normalized static pressure along the blade surface according to the simulation results obtained with the HEoS, CEoS, and EEoS-PINN models.

simulations is reported in Fig. 10, where the normalized axial and pitch-wise coordinates are defined as

$$x^* = \frac{x}{c_x}, \quad y^* = \frac{y}{\Delta y} \quad (30)$$

where c_x and Δy are defined as the blade axial chord length $(3.0 \times 10^{-2} m)$ and tangential blade pitch $(4.5 \times 10^{-2} m)$.

As shown in Fig. 10, the pitchwise location and strength of the shock waves forming at the blade trailing edge and the wake are accurately captured by the flow solution calculated with the EEoS-PINN model. The most notable difference between the HEoS and EEoS-PINN simulation results was a nearly constant, albeit small, offset in the static temperature distributions on the blade surface and at the outlet boundary. The mean, absolute relative percentage error values of the pressure and temperature along the blade surface and outlet for the EEoS-PINN and
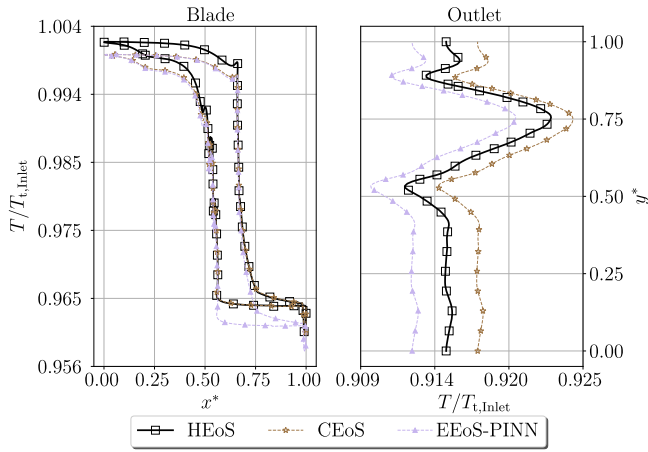
Fig. 9. Normalized static temperature distribution along the blade surface and the outlet boundary according to the simulation results obtained with the HEoS, CEoS, and EEoS-PINN models.
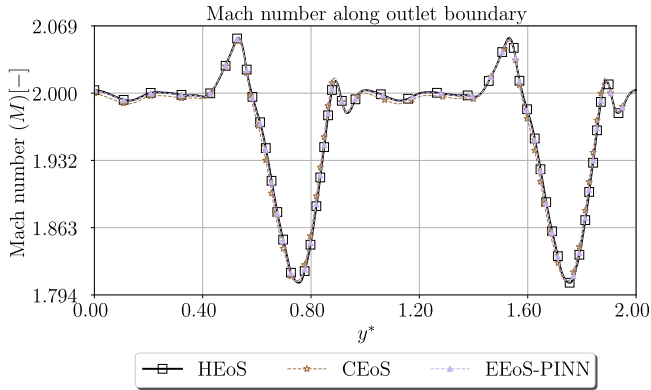

Fig. 10. Pitchwise distribution of the Mach number along the outlet boundary computed with three EoS models.

Table 6
Root-mean-square relative percentage error between CoolProp and EEoS-PINN and CEoS along boundaries.

State variable	Blade (EEoS-PINN/CEoS)	Outlet (EEoS-PINN/CEoS)
p	1.745e-1 / 5.142e-1	8.803e-2 / 3.867e-1
T	2.495e-1 / 1.119e-1	2.828e-1 / 2.474e-1
M	- / -	4.573e-2 / 2.914e-1

CEoS models are reported in Table 6. While from Fig. 9 it appears that the deviation in terms of the temperature is much higher than that of the pressure, the relative differences are very similar.

Given that the ultimate goal of NICFD simulations based on EEoS-PINN is to reduce the cost of automated design optimization methods for turbomachinery applications, it is crucial that the values of the loss coefficient ζ, the mass flow averaged outlet flow angle α_{Outlet} and the mass flux per unit span $\dot{m}$ are calculated accurately. ζ was defined as

$$\zeta = \frac{\overline{T}_{\text{Outlet}}\left(\overline{s}_{\text{Outlet}} - \overline{s}_{\text{Inlet}}\right)}{\frac{1}{2}|u|^2}, \tag{31}$$

where $\overline{T}_{\text{Outlet}}, \overline{s}_{\text{Outlet}}, \overline{s}_{\text{Inlet}}$ are the mass flow averaged outlet static temperature, outlet entropy, and inlet entropy. According to the values listed in Table 7, the performance metrics obtained with the EEoS-PINN model deviate by less than 1 % relative to those calculated with the HEoS model.

Table 7
Loss coefficient (ζ), outlet flow angle (α_{Outlet}), and mass flux ($\dot{m}$) according to the solutions obtained by HEoS, EEoS-PINN, and CEoS. The error values (ϵ) are taken with respect to the HEoS value.

	HEoS	EEoS-PINN($\epsilon[\%]$)	CEoS($\epsilon[\%]$)
$\zeta[-]$	3.2386e-02	3.2308e-02 (2.4e-1)	3.2160e-02 (7.0e-1)
$\alpha_{\text{Outlet}}[\circ]$	75.38	75.36 (2.10e-2)	75.284 (1.24e-1)
$\dot{m}[\text{kgm}^{-1}\text{s}^{-1}]$	2.916e+01	2.926e+01 (3.38e-1)	2.925e+01 (2.87e-1)

4.1. Further analysis on the accuracy of the EEoS-MTNN and EEoS-PINN models

In Section 4, it was reported that the flow simulation did not converge when adopting the EEoS-MTNN model.

It was hypothesized that the nonphysical behavior of the solution shown in the third column of Fig. 6 was due to the model not guaranteeing thermodynamic consistency. Therefore, a more in-depth investigation was conducted by comparing the accuracy and thermodynamic consistency of the EEoS-MTNN and EEoS-PINN models.

The accuracy in the calculation of the thermodynamic properties with the two EEoS models was assessed in terms of mean absolute percentage error ϵ_P against the property values obtained with the reference HEoS model. The density and internal energy were extracted from the flow solution calculated with the HEoS and used as independent variables to compute the fluid properties with the EEoS-MTNN and EEoS-PINN models. The resulting deviations for various primary and secondary properties are shown in Fig. 11.

It can be observed from Fig. 11 that nearly all thermodynamic state quantities are more accurately evaluated by the EEoS-MTNN model compared to the EEoS-PINN model. This observation is consistent with the lower validation losses of the MTNNs during training, as shown in Fig. 2.

Although the EEoS-MTNN model was more precise, the EEoS-PINN model was found to be significantly more consistent. The thermodynamic consistency of the fluid property models was quantified by calculating the mean absolute percentage error between the values of the derivatives of pressure and temperature according to Eqs. (9)–(12) and those obtained by finite differences (FD) along the streamline data. This deviation, referred to as the consistency error ϵ_C for a generic thermodynamic property Q, was defined as

$$\epsilon_C = \frac{1}{N}\sum_{i=1}^{N} 100 \left| \frac{J_{\text{EoS}}^Q - J_{\text{FD}}^Q}{J_{\text{FD}}^Q} \right| \tag{32}$$

where

$$J_{\text{EoS}}^Q = \begin{bmatrix} \frac{\partial Q}{\partial \rho}\Big|_e (\rho, e) \\ \frac{\partial Q}{\partial e}\Big|_\rho (\rho, e) \end{bmatrix} \tag{33}$$

and

$$J_{\text{FD}}^Q = \begin{bmatrix} \frac{Q(\rho+\Delta\rho,e)-Q(\rho-\Delta\rho,e)}{2\Delta\rho} \\ \frac{Q(\rho,e+\Delta e)-Q(\rho,e-\Delta e)}{2\Delta e} \end{bmatrix} \tag{34}$$

where the increments in density $\Delta\rho$ and internal energy Δe were set to 1×10^{-5} of their respective values. The values of ϵ_C according to the EEoS-PINN and EEoS-MTNN models are shown side-by-side on the bottom figure of Fig. 11 for several secondary thermodynamic terms.

Given that the consistency error values are significantly lower for the EEoS-PINN model, it can be concluded from Fig. 11 that physics-informed neural networks allow us to derive fundamental relations from the same initial data set that significantly improve thermodynamic consistency. The benefits are largely evident from the CFD study that was conducted. An analogous conclusion was drawn by Rosenberger [10],
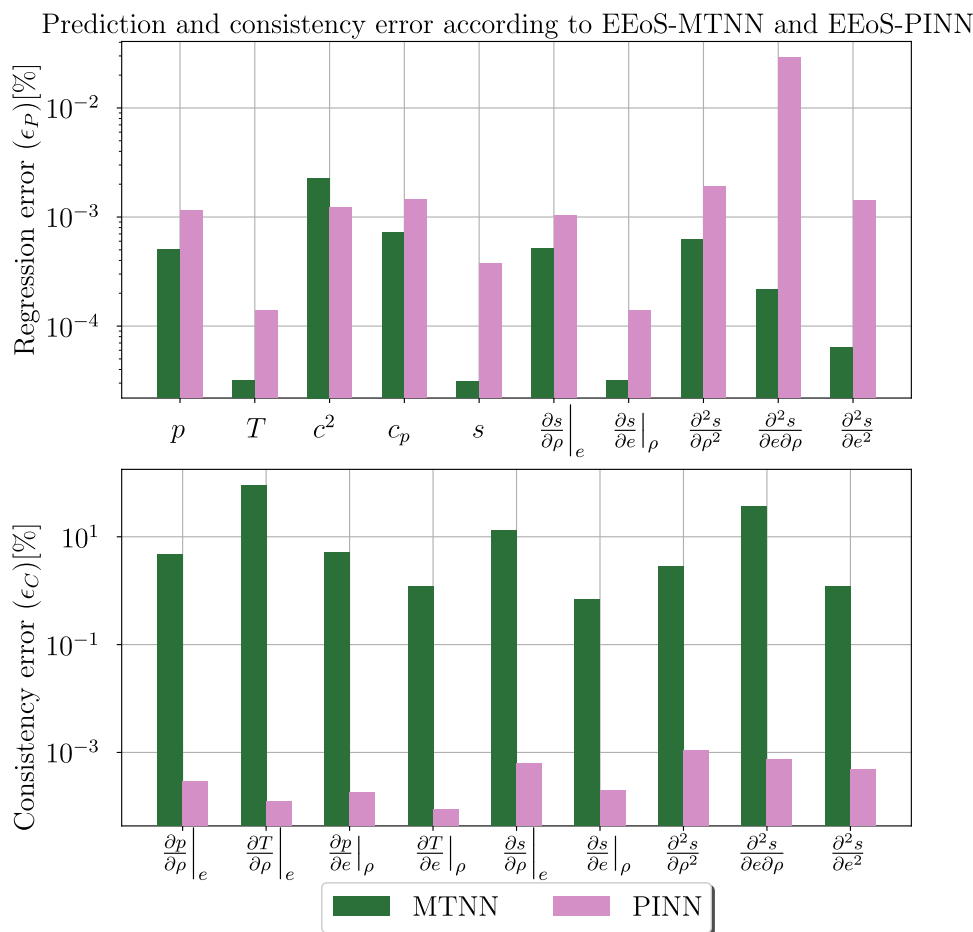
Prediction and consistency error according to EEoS-MTNN and EEoS-PINN

Fig. 11. Mean absolute percentage error of the thermodynamic state quantities according to EEoS-MTNN and EEoS-PINN (top) and consistency error (bottom) with respect to HEoS streamline data.

who compared the Jacobian terms obtained from a data-driven model in Helmholtz free energy form, using both MTNN and PINN.

5. Conclusions

A data-driven method for calculating the properties of pure fluids and mixtures of fixed composition in the nonideal thermodynamic region has been documented. The method enables the implementation of fundamental thermodynamic equations based on the entropy potential using physics-informed neural networks. The resulting model was implemented in the open-source CFD software SU2 for NICFD simulations. The method is inherently consistent, and higher accuracy is achieved through the solution of an optimization problem in which the difference between the property values calculated with a thermodynamic library and those calculated by the thermodynamic model is minimized. Method verification was carried out by simulating the turbulent flow of siloxane MM in the highly nonideal thermodynamic state through a two-dimensional turbine cascade and comparing the simulation results to those obtained with the reference Helmholtz equation of state and with the Peng-Robinson cubic equation of state models. It was found that the data-driven equation of state model provides highly accurate solutions, that the flow solver robustness was not affected, and that the use of the data-driven model requires about 60 % less computational time than the time needed to run the simulation with the Helmholtz equation of state from the external thermodynamic library CoolProp. However, the cost of the simulation with the data-driven model is nearly three times higher than that required when using the Peng-Robinson model.

An additional flow calculation was performed in which the thermodynamic properties were calculated using the data-fitting approach. De-

spite the high accuracy of this method, the simulation diverged because of the inconsistency of the thermodynamic model. This highlights the importance of thermodynamic consistency of data-driven fluid models used in CFD.

Future work will be devoted to the application of the data-driven model in the context of adjoint-based shape optimization of turbomachinery components operating with nonideal compressible flows and improving the computational efficiency and accuracy of the method through optimization of the network hyper-parameters.

CRediT authorship contribution statement

Evert Bunschoten: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation; **Alessandro Cappiello:** Writing – review & editing, Writing – original draft, Methodology, Formal analysis; **Matteo Pini:** Writing – review & editing, Writing – original draft, Supervision, Project administration.

Supplemental material

The EEoS-PINN fluid model was implemented in the open source CFD suite SU2 version 8.1.0. The fluid data were generated and the artificial neural networks were trained using SU2 DataMiner.[1] Step-by-step instructions on how to reproduce the results presented in this manuscript

---

are available on the GitHub repository of the Propulsion and Power group of the Faculty of Aerospace Engineering at TU Delft.[2]

References

[1] Guardone A, Colonna P, Pini M, Spinelli A. Nonideal compressible fluid dynamics of dense vapors and supercritical fluids. Annu Rev Fluid Mech 2024;56:241–69. https://doi.org/10.1146/annurev-fluid-120720-033342

[2] Colonna P, Casati EIM, Trapp C, Mathijssen T, Larjola J, Turunen-Saaresti TE, et al. Organic rankine cycle power systems: from the concept to current technology, applications and an outlook to the future. J Eng Gas Turbin Power 2015;137(10). https://doi.org/10.1115/1.4029884

[3] Banuti DT. A thermodynamic look at injection in aerospace propulsion systems. In: AIAA Scitech 2020 forum. No. 0 in AIAA SciTech Forum; American Institute of Aeronautics and Astronautics; https://doi.org/10.2514/6.2020-1154

[4] Banuti D. Why we need to care about supercritical and non-ideal injection. In: APS Division of fluid dynamics meeting abstracts. AA(University of New Mexico); 2021, p. T01.008.

[5] Romei A, Gaetani P, Persico G. Computational fluid-dynamic investigation of a centrifugal compressor with inlet guide vanes for supercritical carbon dioxide power systems. Energy 2022;255. https://doi.org/10.1016/j.energy.2022.124469

[6] SwestyF, Douglas. Thermodynamically consistent interpolation for equation of state tables. J Comput Phys 1996;127:118–27. https://doi.org/10.1006/jcph.1996.0162

[7] Pini M, Spinelli A, Persico G, Rebay S. Consistent look-up table interpolation method for real-gas flow simulations. Comput Fluids 2014;107. https://doi.org/10.1016/j.compfluid.2014.11.001

[8] Milan PJ, Hickey J-P, Wang X, Yang V. Deep-learning accelerated calculation of real-fluid properties in numerical simulation of complex flowfields. J Comput Phys 2021;444:110567. https://doi.org/10.1016/j.jcp.2021.110567

[9] Pini M, Guiffre A, Cappiello A, Majer M, Bunschoten E. Data-driven regression of thermodynamic models in entropic form. Proceeding of the 4th International Seminar on Non-Ideal Compressible Fluid Dynamics for Propulsion and Power https://doi.org/10.1007/978-3-031-30936-6_3

[10] Rosenberger D, Barros K, Germann TC, Lubbers N. Machine learning of consistent thermodynamic models using automatic differentiation. Phys Rev E 2022;105(4):045301– . https://doi.org/10.1103/PhysRevE.105.045301

[11] Raissi M, Perdikaris P, Karniadakis G. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045

[12] Chaparro G, Muller E. Development of thermodynamically consistent machine-learning equations of state: application to the mie fluid. J Chem Phys 2023;158. https://doi.org/10.1063/5.0146634

[13] Kevrekidis G, Serino D, Kaltenborn M, Gammel J, Burby J, Klasky M. Neural network representations of multiphase equations of state. Sci Rep 2024;14. https://doi.org/10.1038/s41598-024-81445-4

[14] Kuang D, Li S, Wang B, Xiong C, Zhang S, Zhang Y. Joint learning equation of state surfaces with uncertainty-aware physically regularized neural networks. Sci Rep 2025;15. https://doi.org/10.1038/s41598-025-11874-2

[15] Sharma H, Gaffney J, Tsapetis D, Shields M. Learning thermodynamically constrained equations of state with uncertainty. APL Machine Learn 2024;2. https://doi.org/10.1063/5.0165298

[16] Kezheng Z, Müller EA. Generating a machine-learned equation of state for fluid properties. J Phys Chem B 2020;124. https://doi.org/10.48550/arXiv.2007.14689

[17] Wheeler APS. High fidelity simulation of dense vapours with thermodynamic consistent modelling. Comput Fluids 2024;268. https://doi.org/10.1016/j.compfluid.2023.106088

[18] Economon TD. Simulation and adjoint-based design for variable density incompressible flows with heat transfer. AIAA J 2019;58(2):757–69 . https://doi.org/10.2514/1.J058222

[19] Bell IH, Wronski J, Quoilin S, Lemort V. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library coolprop. Indust Eng Chem Res 2014;53(6):2498–508. https://doi.org/10.1021/ie4033999

[20] Kingma DP, Ba JL. Adam: a method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings 2015;:1–15 https://doi.org/10.48550/arXiv.1412.6980

[21] Head AJ. Novel experiments for the investigation of non-ideal compressible fluid dynamics: The ORCHID and first results of optical measurements. Ph.D. thesis; Delft University of Technology; 2021. https://doi.org/10.4233/uuid:a3b03976-2df6-435c-b7df-1505718fcd3a

[22] Ghidoni A, Pelizzari E, Rebay S, Selimin V. 3D anisotropic unstructured grid generation. Int J Numer Methods Fluids 2005;51(9–10):1097–115. https://doi.org/10.1002/fld.1151