



LJMU Research Online

Jalili, D, Jadidi, M, Keshmiri, A, Chakraborty, B, Georgoulas, A and Mahmoudi, Y

Transfer learning through physics-informed neural networks for bubble growth in superheated liquid domains

<http://researchonline.ljmu.ac.uk/id/eprint/26205/>

Article

Citation (please note it is advisable to refer to the publisher's version if you intend to cite from this work)

Jalili, D, Jadidi, M, Keshmiri, A, Chakraborty, B, Georgoulas, A and Mahmoudi, Y (2024) Transfer learning through physics-informed neural networks for bubble growth in superheated liquid domains. International Journal of Heat and Mass Transfer. 232. ISSN 0017-9310

LJMU has developed [LJMU Research Online](#) for users to access the research output of the University more effectively. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Users may download and/or print one copy of any article(s) in LJMU Research Online to facilitate their private study or for non-commercial research. You may not engage in further distribution of the material or use it for any profit-making activities or any commercial gain.

The version presented here may differ from the published version or from the version of the record. Please see the repository URL above for details on accessing the published version and note that access may require a subscription.

For more information please contact researchonline@ljmu.ac.uk

<http://researchonline.ljmu.ac.uk/>



Transfer learning through physics-informed neural networks for bubble growth in superheated liquid domains

Darioush Jalili^a, Mohammad Jadidi^a, Amir Keshmiri^a, Bhaskar Chakraborty^b, Anastasios Georgoulas^b, Yasser Mahmoudi^{a,*}

^a School of Engineering, The University of Manchester, Manchester, UK, M13 9PL

^b Advanced Engineering Centre, School of Architecture Technology and Engineering, University of Brighton, Brighton, UK, BN2 4GJ

ARTICLE INFO

Keywords:

Physics-informed neural networks
Two-phase flows
Bubble dynamics
Heat transfer
Phase-change
Inverse problems

ABSTRACT

In this paper, a physics-informed neural network (PINN) technique is developed to study the heat and mass transfer for the process of vapour bubble growth in a superheated liquid domain and tested using three working fluids including water, R-134a and FC-72. The work represents a novel step in the development of PINNs for phase change scenarios where surface tension effects dominate, and acts as a necessary validation stage before PINN techniques can be applied to complex boiling analysis. Initially, a forward analysis was performed using water and R-134a as working fluids. For each of these investigations, the PINN algorithm was trained on 50 % of the available CFD data. The proposed algorithm was able to accurately infer velocity fields, particularly in the near-interfacial region. The resultant circulatory flow was found to maintain the desired circular shape of the growing bubbles. As a result, when predicting the evolution of a water vapour bubble, the developed PINN algorithm produced a reduction in peak error by 0.87 % compared to CFD reference data, and 3.42 % reduction in peak error for prediction of the evolution of the R-134a vapour bubble. To test and optimise the transfer learning capabilities of the developed methodology, the evolution of an FC-72 vapour bubble in superheated FC-72 was predicted without supplying supporting observational data. For this scenario, the PINN algorithm produced a peak error within 1.3 % of the unobserved CFD reference data. The proposed approach confirms the robustness of PINN methodologies as a method of solving phase-change problems where surface tension plays a pivotal, promising to expedite parametric studies in practice. This study represents a pioneering effort in the development of PINNs for phase change by applying the current algorithm to investigate bubble growth within superheated liquid domains, serving as a basis for the application of PINNs for boiling problems and as a benchmark for inverse training strategy.

1. Introduction

Boiling has many applications across the fields of science and engineering, such as the thermal management of electronics [1], the distillation of fossil fuel [2] and the cooling of solar collectors [3]. Reliable modelling of bubble behaviour in real systems is a complex task because of the transient nature of bubble dynamics [4]. As products shrink in size and become increasingly powerful, it is becoming necessary to foray beyond accepted safety margins. To do this, tools must be developed which can model the local interfacial behaviours during the process of evaporation. Validation of numerical methods for fluid behaviours during evaporation must be performed against cases with a known analytical solution to understand and improve upon existing solver

limitations before one can begin the sequence of modelling more complex interactions with a solver [5,6]. Therefore, analytical solutions relating to the rate of growth of a bubble dependent on fluid properties provide a basis for many studies across an array of industries. Often, a compromise must be struck that balances the complexity of modelling algorithms and solution accuracy at the interfacial level [7]. One promising method of achieving this compromise for investigating phase-change phenomena is Physics-Informed Neural Networks (PINNs) [8]. The present work therefore marks a novel initiative in applying PINNs to study the growth of bubbles in superheated liquid domains to learn more about the complex phenomenon of boiling.

The growth of a bubble in a single-component fluid can be divided into three distinct stages; inertial, thermal, and diffusive growth [9]. At each stage, a different mechanism is responsible for restricting the rate

* Corresponding author.

E-mail address: yasser.mahmoudi@manchester.ac.uk (Y. Mahmoudi).

<https://doi.org/10.1016/j.ijheatmasstransfer.2024.125940>

Received 15 May 2024; Received in revised form 28 June 2024; Accepted 5 July 2024

Available online 27 July 2024

0017-9310/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Nomenclature	
C_p	Specific heat capacity at constant pressure (J/(kg·K))
D	Diameter (m)
Er	Percentage Error ($[(\hat{\Phi} - \Phi)/\Phi] \times 100$)
Fr	Froude number ($U_{ref} / \sqrt{g L_{characteristic}}$)
h	Enthalpy (J/kg)
j	Mass flux (evaporative) (kg/(m ² ·s))
Ja	Jakob number
L	Characteristic length (m)
N	Source term scaling coefficient
n	Cell surface normal vector
Pe	Peclet number ($uL\rho C_p / k$)
p	Pressure (Pa)
$p_{dynamic}$	Dynamic pressure (ρu^2) (Pa)
R	Radius (m)
$R_{gas, int}$	Thermal resistance (K/W)
Re	Reynolds number
T	Temperature (K)
t	Time (s)
u	Instantaneous streamwise velocity (m/s)
v	Instantaneous spanwise velocity (m/s)
U	Characteristic velocity (scalar) (m/s)
\mathbf{u}	Instantaneous velocity (vector) (m/s)
W	Weight parameter value
We	Weber number ($\rho_{ref} U_{ref}^2 L_{characteristic} / \sigma$)
x	Length in the streamwise direction (m)
y	Length in the spanwise direction (m)
Symbols	
α	Volume fraction
β	Growth rate constant
∇	Spatial differential (del)
η	Learning rate coefficient
Θ	Neural network trainable parameter
θ	Angle (Rads)
ι	Loss value (MSE loss in the current study)
κ	Interface curvature
ρ	Density (Kg/m ³)
σ	Surface tension (N/m)
ν	Kinematic viscosity (m ² /s)
γ	Evaporation coefficient
δ	Infinitesimal change
Φ	Data generated by CFD
$\hat{\Phi}$	Data predicted by PINN
Subscript	
cut	Threshold value
$data$	Training data
f	Cell face
Int	Interfacial
lv	Liquid-to-vapour
NN	Neural Network
ref	Pre-determined reference value
sat	Saturation
T	Temperature
0	Initial condition
1	Subsequent condition
Superscript	
$*$	Dimensionless parameter
T	Transpose
$\hat{\cdot}$	Predicted data by PINN
$\dot{\cdot}$	Rate of change
Abbreviation	
API	Application programming interface
CFD	Computational Fluid Dynamics
CNN	Convolutional neural network
DL	Deep learning
ML	Machine learning
MSE	Mean-squared error
NN	Neural network
PINN	Physics-informed neural networks
Tanh	Hyperbolic tangent (activation function)

of growth. It is the inertial growth which initially transforms the vapour from a nucleating seed to a small bubble. However, the overall time taken to complete inertial growth is on the order of milliseconds. Diffusive growth occurs due to gas molecules from the surrounding liquid diffusing into the bubble. This stage calls upon Fick's law of diffusion, which is analogous to Fourier's law of heat conduction [10]. Falling between inertial and diffusive growth is thermal growth. Thermal growth is driven by the heat energy consumed by vaporisation across the bubble interface. This process can be assumed to be isobaric and lasts significantly longer than the inertial growth stage. The rate of growth during the isobaric stage is much slower, though. It is this isobaric process which contributes most to the increase in the volume of a bubble [11]. The isobaric growth regime enables the bubble to grow equally in all directions, which provides predictability and stability for system operations. Management of the isobaric period enables precise control of bubble size, which is vital for applications ranging from boiling heat transfer [12] to drug delivery [13]. During this isobaric period, the rate of growth can be described by standard mass and momentum equations on the assumption that the vapour remains incompressible and non-viscous. This stage also neglects the effects of gravity. Therefore, isobaric growth of a bubble is a problem generalisable to many domains, and correct modelling of this regime has widespread

implications for the accuracy of many multiphase solvers which include phase-change. The theoretical solution to the growth of a single bubble in a superheated liquid has received much attention in literature. Assumptions made by Scriven [14] of constant pressure inside a symmetrical vapour bubble which obeys ideal gas laws provide a reliable analytical solution to the isobaric growth stage. The growth of a single bubble in a superheated liquid has been used in countless validations of numerical methods [15-17]. The classical Scriven case provides several useful simplifications compared to a real-world study. For example, the geometry for the Scriven problem is simplified to a 1D axisymmetric domain and the vapour interaction is only considered for the liquid phase. Interactions with walls are not considered, nor are any variations in material properties during the evaporation process. This procedure, along with similar analytical methods [18], is employed to ensure solvers under development produce sharp interfaces and accurately assign the transfer of mass under expected conditions of operation. Doing so requires precise calculation of thermal boundary layers in the liquid-vapour interface, thus ensuring the conservation of energy, momentum, and mass across the boundary. To date, there is a lack of literature which quantifies the level of accuracy attainable by PINN methods for predictions of evaporation using classical validation studies.

In recent times, an abundance of literature has been produced that centres on the application of physics-informed learning methods to increasingly complex fluid flow studies. Ranging from canonical problems [19,20] to questions which have eluded researchers for decades [21], PINNs are proving invaluable in the quest to minimise necessary assumptions when modelling physical processes [22] and improve understanding where fidelity in modelling results is insufficient through sparse or incomplete data [23-25]. By placing significant weight on the Boussinesq approximation, Cai et al. [26] successfully inferred the turbulent rise of vapour from a cup of espresso using tomographic background-oriented Schlieren (Tomo-BOS) measurement methods. The application of PINNs to heat transfer was taken further by Zhu et al. [27], who accurately predicted weld pool dynamics. Nikai et al. [28] extended this understanding to incorporate real-time simulations of resin cure kinematics. Masclans et al. [29] have applied PINNs to determine flow physics in scenarios which combine turbulence and heat transfer [29]. Kim et al. [30] applied a PINN algorithm to determine minimum film temperature, however, this relied on a large sample size of around 400 studies. Jahanbakhsh et al. [31] demonstrated that PINNs could be applied to complex geometries and attain high levels of accuracy for thin film evaporation investigations. Once again, this relied on a large training dataset. Despite these advances, PINN remains novel and as such gaps in knowledge of best practice remain. Most notably, there is a lack of true validation of PINN methodologies for the evaporation process. Some work on classification and guidelines has been performed by Sharma et al. [32], who delved into the fundamental architecture decisions faced by researchers when implementing PINN solutions, while Papadopoulos et al. [33] discussed the Bayesian optimisation for heat transfer problems. Royer et al. [34] discussed the process of interfacing ANNs with traditional numerical methods to achieve robust interpolative performances relating to radiative heat transfer processes. Recent research [35-37] has also encompassed the optimisation of inverse studies. Radhakrishnan et al. [38] have attempted to do this by applying a variation of transfer learning to predict the behaviour of a bubble in a vortex and also a bubble rising in a quiescent fluid. Understanding the most effective method of harnessing the inverse capabilities of PINN promises to enable rapid querying to understand the impact of changing simulation parameters. Consequently, there is significant momentum to fill the existing gap in the literature for the application of PINN algorithms to investigate phase-change processes especially when data availability is limited [39].

In the present work, a physics-informed neural network (PINN) technique is developed in order study the evaporative phase change process for the first time. The model is validated using the phase-change benchmark of the Scriven bubble problem, with significant focus on heat and mass transfer within this multiphase flow process. Initially, the proposed approach provides forward solutions for evaporation using two working fluids (water and refrigerant R-134a). Comparisons are made to analytical solutions and obtained CFD data for these fluids, which demonstrate that the PINN methodology can reliably and accurately obtain a sharp interfacial region. Next, the algorithm is robustly assessed on its ability to effectively leverage previous training through transfer learning by predicting the interface progression of a third working fluid (FC-72) for which no observational data was provided. Instead, the PINN algorithm relies on resolving PDEs and inferring behaviours observed in the forward cases. A detailed explanation of this transfer learning optimisation is included. Similar comparisons are made to assess the effectiveness of transfer learning methods when tasked with predicting a sharp and accurate interfacial region. An insightful analysis is performed on how one can optimise the process of obtaining inverse predictions for phase-change problems where surface tension is a deterministic factor. Ultimately, this process forms the foundations of a use-case for PINN as a rapid and flexible investigative method of determining behaviours of disparate fluids under the phase-change process. This contribution will guide future research on applying PINN methods to more complex phase-change phenomena.

2. Computational methodology

This section provides an overview of the PINN architecture and corresponding hyperparameter decisions. Additionally, the partial differential equations (PDEs) utilised by the solver are introduced, accompanied by a concise discussion on the computational resources necessary for training.

2.1. Governing equations

PINN techniques rely on a range of partial differential equations (PDEs) to comprehensively model the behaviour of two-phase systems [40]. In this work, liquid and vapour phases are treated as Newtonian and incompressible fluids, as the observed Mach numbers in the vapour region are always appropriately low. The current work encompasses 6 PDEs, to be solved simultaneously. Due to the assumptions made when modelling the Scriven problem, gravitational effects are neglected. The governing equations therefore include:

Mass conservation,

$$\frac{\partial \alpha}{\partial t^*} + \mathbf{u}^* \cdot \nabla^* \alpha = 0 \quad (1)$$

Momentum transport via the incompressible Navier-Stokes equations,

$$\rho^* \left(\frac{\partial \mathbf{u}^*}{\partial t^*} + (\mathbf{u}^* \cdot \nabla^*) \mathbf{u}^* \right) + \nabla^* p^* - \nabla^* \cdot \frac{1}{Re} (\nabla^* \cdot \mathbf{u}^* + \nabla^* \cdot \mathbf{u}^{*T}) - \frac{1}{We} \kappa \nabla^* \alpha = 0 \quad (2)$$

where,

$$\kappa = -\nabla^* \cdot \frac{\nabla^* \alpha}{|\nabla^* \alpha|} \quad (3)$$

$$We = \frac{\rho_{ref} U_{ref}^2 L_{characteristic}}{\sigma} \quad (4)$$

and an energy conservation equation based on the transient heat equation:

$$\mathbf{u}^* \cdot \nabla^* T^* + \frac{\dot{q}}{k} - \frac{1}{PrRe} \nabla^* \cdot \nabla^* T^* = 0 \quad (5)$$

Notably, Eq. (5) contains a Peclet number (Pe) term. The Peclet number represents the rate of advection to thermal diffusion, which ultimately determines heat and mass transfer across liquid-vapour interface. The Volume-of-Fluid (VOF) methodology [41] uses a colour function (α) to advect the phase interface. It offers notable advantages over other interface capturing methods such as level-set. One such advantage is preservation of volume for each phase, which promotes stability of the solution [42]. Fluids 1 and 2 correspond to cells where $\alpha = 0$ and $\alpha = 1$, respectively. Locations where α is neither 0 nor 1 identify cells as being cut by an interface. Piecing together these cells allows the true shape of the interface to be determined. In this work, it was exceptionally important that a sharp fluid interface was maintained to provide accurate details of fluid behaviour during the process of evaporation. To achieve suitably sharp interfaces in the training data used for this work, additional interfacial sharpening terms are utilised in the CFD solver by way of an artificial compressive velocity $U_{relative}$, however, this behaviour must be inferred by the PINN methodology where no such source term is explicitly specified. $U_{relative}$ is given by:

$$U_{relative} = n_{fmin} \left[C_{\gamma} \frac{\dot{m}}{|S_f|}, \max \left(\frac{\dot{m}}{|S_f|} \right) \right] \quad (6)$$

To ensure the compressive source term only acts in the interfacial region, the term is set to 0 unless $0.05 < \alpha < 0.999$. The term C_{γ} is set to 1 in accordance with literature [43].

The phase change model implemented in this body of work was taken from Hardt & Wondra [44]. The Hardt & Wondra model is self-described as a non-specific model which does not rely on any single numerical algorithm to ensure continuity of mass, momentum, enthalpy, and phase distribution. This flexible mass transfer model can be added to both level-set and VOF schemes because all required quantities are determined from scalar fields in the numerical grid. It can therefore be added to existing two-phase solvers, which means existing work using VOF in PINN [45] can be augmented to model temperature-dependent phase change. In addition to excellent comparisons with analytical solutions, this phase change model allows for a range of boundary conditions at the fluid interface and does not require that the fluidic interface is at a fixed temperature. This feature promises scope to interrogate complex boiling behaviour in future works. While the phase change models presented in [43,44] represent significant advances in CFD modelling, this work marks the inaugural application of these pertinent principles to PINNs.

The evaporative mass flux at the liquid-vapour interface is calculated by:

$$j_{evap} = \frac{T_{int} - T_{sat}}{R_{int} h_{lv}} \quad (7)$$

T_{sat} is the predetermined saturation temperature of the liquid phase, while T_{int} is the temperature at the interface. h_{lv} is the latent heat of evaporation, while R_{int} is the interfacial thermal resistance which is determined by:

$$R_{int} = \frac{2 - \gamma}{\gamma} \frac{\sqrt{2\pi R_{gas}}}{h_{lv}^2} \frac{T_{sat}^{3/2}}{\rho_{gas}} \quad (8)$$

Determination of R_{int} therefore relies on the parameter γ , whose value varies from $0 < \gamma < 1$. For the working fluids of water liquid and water vapour used in this investigation, a value of unity has been used following the literature [46-49]. All other parameters are predetermined as constants of the working fluids. To enhance stability, the mass flux calculations Eq. (7) must be incorporated into volumetric source terms. This is relatively straightforward:

$$\dot{\rho}_0 = j_{evap} |\nabla \alpha| \quad (9)$$

and subsequently,

$$\dot{m}_{int} = \iint \dot{\rho}_0 dA \quad (10)$$

Eq. (10) is used to ensure that rates of evaporation from the liquid phase are matched exactly by additional mass being supplied to the vapour phase. A Neumann boundary condition is imposed to artificially smoothen the source term in Eq. (11), where Δt is an artificial timestep. The value D must be adjusted to the mesh resolution to ensure that the source term field is sufficiently distributed across enough cells to stabilise the calculation.

$$\dot{\rho}_1 - \nabla \cdot [(D\Delta t)\nabla \dot{\rho}_1] = \dot{\rho}_0 \quad (11)$$

Scaling coefficients N_l and N_v from Georgoulas et al. [43] are determined by integration of the smoothed source term field in each phase such that:

$$N_l = \dot{m}_{int} \left[\iint (\alpha - 1 + \alpha_{cut}) \dot{\rho}_1 dA \right]^{-1} \quad (12)$$

$$N_v = \dot{m}_{int} \left[\iint (\alpha_{cut} - \alpha) \dot{\rho}_1 dA \right]^{-1} \quad (13)$$

where values of α fall beyond the region of $\alpha > 1 - \alpha_{cut}$ or $\alpha < \alpha_{cut}$ (α_{cut} is set to 1×10^{-3} in this work), the source terms are artificially reduced to zero. This cropping step ensures that source terms are shifted into the pure vapour and liquid cells are only in the vicinity of the interface. The

interface therefore is not subjected to any source terms and is only transported by the calculated velocity field. As a result, the transport algorithm for the volume fraction field as well as the associated interface compression can work efficiently without any interference with the source term field. The remaining source term field is scaled individually on the liquid and the vapour side through the application of appropriate scaling coefficients. This scaling step ensures that the mass is globally conserved and that the evaporating or condensing mass flow corresponds globally to the net mass flow through the interface. The final source term distribution is given by:

$$\dot{\rho} = N_v(\alpha_{cut} - \alpha)\dot{\rho}_1 - N_l(\alpha - 1 + \alpha_{cut})\dot{\rho}_1 \quad (14)$$

Finally, the enthalpy source term can be obtained with additional knowledge of the thermal properties of the working fluids:

$$\dot{h} = [N_v(1 - \alpha)C_{p,v} - N_l\alpha C_{p,l}]\dot{\rho}_1 T - \dot{\rho} h_{lv} \quad (15)$$

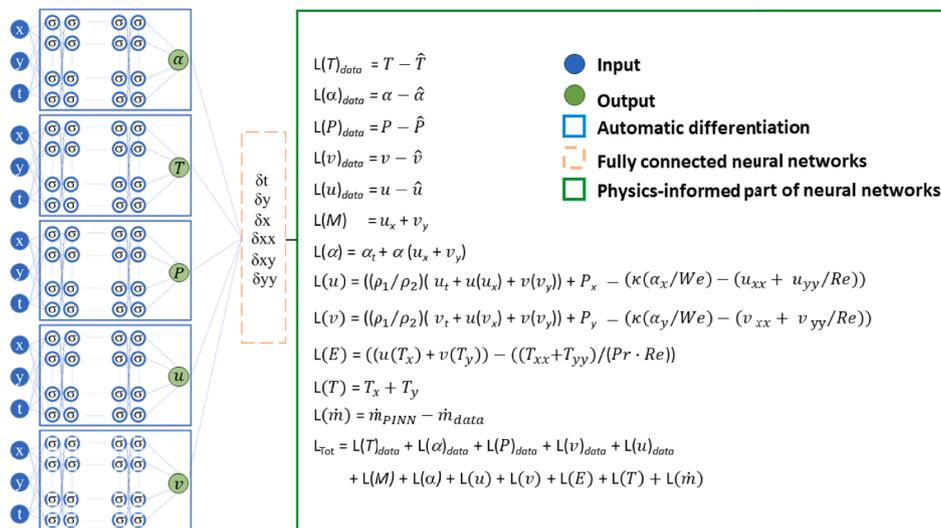
2.2. PINN methodology

Upon determining suitable PDEs, initial, and boundary conditions, a mapping exercise is performed to equate physical parameters to space and time:

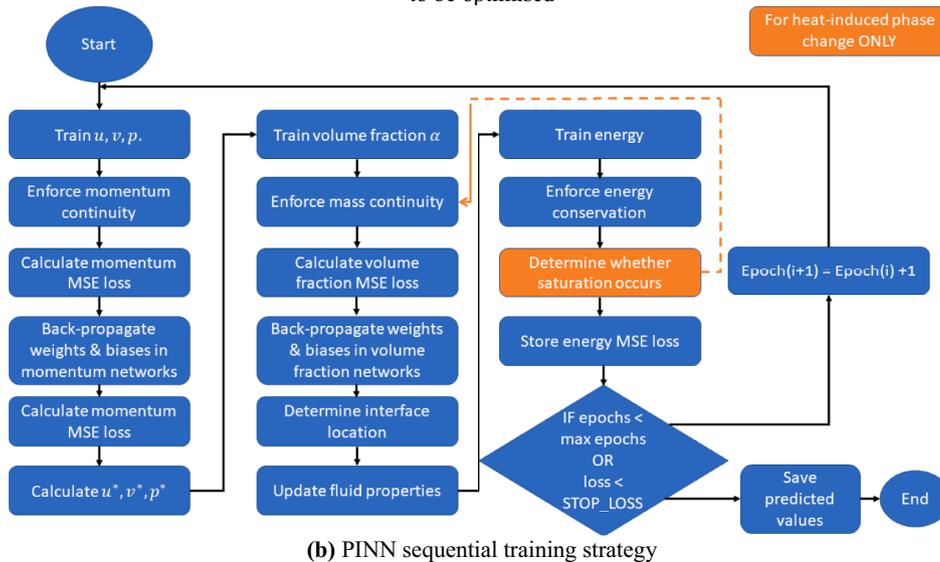
$$(u, v, p, \alpha, T) = F_{NN}(x, y, t, \Theta) \quad (16)$$

where, F_{NN} represents the total contributions of the sequential neural networks, determined by the trainable parameters Θ of weights W and biases B . In this work, each physical parameter is mapped to a unique network (Fig. 1(a)). Fig. 1(b) illustrates this ensemble learning approach. The ensemble approach effectively reduces the number of unknowns to be solved at any one time. The networks employed here also use an additional trainable nodewise coefficient a . The non-linear activation parameter a is applied at each component of the vector produced at nodal outputs. The inclusion of this parameter at each node increases the number of degrees of freedom in the network by providing each node with a unique activation slope. These nodewise coefficients effectively transfer the problem of optimisation to a nodal (rather than global) level [50], which provides improved performance when resolving non-linear relationships and ensures better generalisation to unseen data. This ultimately means improved resilience to overtraining. Furthermore, the enhanced parametric space enables complex patterns to be attained in fewer iterations compared to a standard network because each node need only be optimised over a much smaller range of scenarios [50]. While the high parametric space increases computational cost by around 1 % compared to a standard architecture, the additional draw of high dimensionality is significantly less than the increased number of iterations (epochs) required to meet the required low error threshold when using standard architecture (without the trainable coefficient a).

The total loss of PINN systems is determined by the summation of residual terms across each parameter network. These residuals are composed of the differences between predictions made by the PINN and known data or by direct solution of PDEs at a given collocation (assessment) point. Naturally, the lower the residuals, the more accurate the predictions become. Training continued until the residuals reached a predetermined threshold (Fig. 1(b)). To minimise these residuals, it was essential to determine the optimum architecture for each case. It is known that the batch size of training data plays a deterministic role in prediction accuracy [51]. So, a hyperparameter grid search to find the optimum batch size along with the number of hidden layers, activation function and number of nodes per layer, was performed using an alternative dataset which negated the effects of surface tension. This hyperparameter search extended to the learning rate, the hyperparameter responsible for governing the speed at which the weights and biases within the network are updated [52], based on the error calculated at each iteration. If the learning rate is too large, the algorithm is



(a) The Scriven problem studies used 5 fully connected neural networks. True architecture properties are shown in Table 1, but for representation purposes, each network displays 4 layers and 4 neurons per hidden layer. All derivatives are computed using automatic differentiation. Each loss term ‘L’ is converted to an MSE value. The sum of these MSE values (L_{Tot}) gives the total loss to be optimised



(b) PINN sequential training strategy

Fig. 1. Configuration and training strategy of the PINN algorithm; (a) network architecture; (b) training strategy. (a) The Scriven problem studies used 5 fully connected neural networks. True architecture properties are shown in Table 1, but for representation purposes, each network displays 4 layers and 4 neurons per hidden layer. All derivatives are computed using automatic differentiation. Each loss term ‘L’ is converted to an MSE value. The sum of these MSE values (L_{Tot}) gives the total loss to be optimised (b) PINN sequential training strategy.

likely to diverge [53], while a low learning rate can significantly extend training time [54]. Therefore, defining learning rate is a compromise between ensuring training time remains practical, and small enough to obtain accurate solutions. It was found that a fixed learning rate of 0.001 was sufficient to predict an interface of comparable sharpness to the CFD reference solution. The optimum parameters are shown in Table 1, which were found to optimise solution accuracy and computational cost.

Table 1
Network architecture for the Scriven cases studied in this work.

Scriven case	Nodes	Layers	Activation function	Batch size	Epochs
Water	150	10	Local ‘tan-h’	512	2000
R-134a	150	10	Local ‘tan-h’	512	3000
FC-72	150	10	Local ‘tan-h’	512	See Section 4.2

The computational cost was defined by core hours (number of computational cores \times number of hours). The quoted core hours come with the caveat that the total computational time is impacted by the processor clock. For both forward problems in this work, the optimum PINN architecture required 184 core hours. The optimum inverse solution required a total training time of 306 core hours. The PINN training time represents a significant reduction in computational expense compared to the investment required to generate the CFD training data. For the CFD models of water and R-134a, 2400 and 1400 core hours were required, respectively. For the FC-72 CFD solution, only 1624 core hours were needed. This is roughly 5 times the computational cost required to train the inverse PINN method, which could then also predict new solutions in a matter of seconds, unlike the CFD solution. Further details on computational cost are included in the Appendix. Moreover, the PINN method represents the possibility of more extreme savings in industrial and research applications; much of the data preparation relies

only on simple scripts which are not problem-specific, whereas much of the CFD preparation time in meshing and geometry design requires dedicated labour. The data used for architecture tuning modelled the phase transition of water-to-water vapour (properties shown in Table 2). It used 160,000 mesh cells and neglected the effects of surface tension $\sigma = 0 \text{ N/m}$ as in [55,56]. This artificial step was used specifically to provide valuable insights to the modelling of interface advection and the phase change algorithm. In ML applications, the hyperparameter tuning process is a vital step which ensures the relevant networks are of sufficient complexity to be capable of recognising desirable behaviours while being simple enough to remain computationally efficient. This could happen without the explicit consideration of surface tension, whose impact of affecting fluid mixing affects all networks equally. PINN predictions were performed using networks of $\text{node} \times \text{layer}$ arrangements ranging between (20×8) , (50×10) and (150×10) .

The difference in the magnitude of input parameters creates a systematic problem for PINNs. To overcome this, each parameter was made dimensionless. Quantitative comparisons against the analytical solution were made at regular intervals in time. As such, the time parameter was made dimensionless such that $t^* = t/(L/U)$, where U was taken as unity to enable the division. It was found that compared to the analytical solution, the (50×10) PINN architecture obtained an error of just 6.08 %. The (50×10) architecture produced a final mean-squared error (MSE) of 1.05×10^{-4} . This compared favourably to the (20×8) architecture, whose error peaked at 6.72 % for $t^* = 32$. MSE for this design was reduced to 5.7×10^{-5} . The (150×10) architecture obtained a peak error of 3.73 % at the same timestep. MSE for this architecture reached just 9.7×10^{-6} . A qualitative element was also considered when determining network architecture. All three architectures produced symmetrical bubbles but increasing network density resulted in a reduction of diffusion for the interfacial region. The ability to produce sharply reconstructed interfaces was a deterministic factor when assessing the capabilities of the PINN methodology. The (20×8) displayed an undesirable level of interfacial diffusion. The (50×10) architecture produced acceptable results, but the (150×10) architecture produced an interfacial region which was as sharp as the CFD reference data for the test case. Though the (150×10) architecture necessitated roughly 25 % additional training time, the increased qualitative accuracy was determined to provide good value for this investment. Since both the qualitative and quantitative assessments were satisfied by the (150×10) architecture, no further dense architectures were investigated. Therefore, (150×10) was deemed to be the most effective architecture in balancing computational cost and solution accuracy.

The bubble interface was defined as the location of $\alpha = 0.5$ for the entirety of this work. While Fig. 2(a-f) provides a qualitative comparison of PINN predictions and reference CFD data, a quantitative comparison

of the interface location was also made between the analytical solution, the CFD solution and PINN predictions (Fig. 3) to reveal a maximum error of 3.73 % against the analytical solution at $t^* = 37$. This improved upon the accuracy obtained by the CFD solution, which produced a maximum error of 6.1 % at $t^* = 84$. It is also important to note that the PINN prediction aligned increasingly accurately to the analytical solution the longer the growth was modelled, whereas the CFD solution continued to oscillate about the analytical solution. Accordingly, the architecture used to achieve these results was used to study the forward and inverse problems which involved surface tension effects with minor variations in training time.

3. Overview of test cases and numerical settings

The next validation case considers a spherical bubble which is heated uniformly and considers surface tension effects. This work is distinctive from the Stefan problem since surface tension plays a pivotal role [57], due to the small length scales associated with bubble growth. The other major factor in growth trajectory is the inertia of the fluids, but the bubble is large enough to neglect inertial effects once the bulk vapour is at saturation temperature [43]. Assumptions are made such that the bubble growth is assumed to be symmetrical and can be simplified to a circle accordingly. Thus, it is possible to simplify growth to a one-dimensional equation and provide an analytical solution to this problem. A one-dimensional heat conduction equation can be used to represent the temperature field in the domain. This begins by augmenting the equation of continuity (refer to Section 2.1) and augmenting it to account for spherical symmetry in an incompressible fluid, such that $u^2 = f(\theta)$. r is the radial coordinate here, whose datum is at the bubble centre. From this, the liquid velocity at any radius can be evaluated. As a result, liquid mass flow is determined by $\dot{m}_l = 4\pi R^2 \rho_l [\dot{R} - u(R)]$. To balance, the liquid mass flow must be equated with the additional mass in the vapour region, giving a mass balance Eq. (17):

$$\frac{d}{d\theta} \left(\frac{4}{3} \pi R^3 \rho_v \right) = 4\pi R^2 \rho_l [\dot{R} - u(R)] \quad (17)$$

Since changes to the bubble radius are orders of magnitude larger than the change to bubble density (ρ_v), it is possible to assume density is independent of time. This results in $u^2 = \epsilon \dot{R} R^2$. Here, $u(R) = \dot{R}(\rho_l - \rho_v)/\rho_l = \epsilon \dot{R}$. Further simplifications to the growth are made using the Clausius-Clapeyron equation and by neglecting the heat of mixing. A full explanation can be found in [14], where Scriven culminates by expressing the analytical solution to isobaric bubble growth as:

$$R = 2\beta\sqrt{\delta\theta} \quad (18)$$

Table 2

Material properties of working fluids water, R-134a and FC-72.

Property	Unit	Water		R-134a		FC-72	
		Liquid	Vapour	Liquid	Vapour	Liquid	Vapour
ρ	(kg/m ³)	958	0.597	1388	4.43	1621.20	13.49
C_p	(J/kg.K)	4220	2030	1270	720	1106.7	924.81
k	(W/m.K)	0.679	0.025	0.106	0.009	0.054	0.014
μ	(Pa.s)	2.77×10^{-4}	1.3×10^{-5}	4.01×10^{-4}	9.64×10^{-6}	4.13×10^{-4}	1.19×10^{-5}
h_{lv}	(J/kg)	2.26×10^6		2.195×10^5		8.36×10^4	
σ	(N/m)	0.059		0.016		0.0084	
P	(bar)	1.013		0.84		1.013	
Pe		297.7		831.49		1661.28	
T_{sat}	(K)	373.15		303.15		330.06	
β		14.59		8.75		7.69	
D	(m ² /s)	1.68×10^{-7}		6.01×10^{-8}		3.02×10^{-8}	
ΔT	(K)	5		5		5	
Ja	$(C_{p,L} \Delta T)/h_v$	9.35×10^{-4}		2.89×10^{-2}		6.62×10^{-2}	

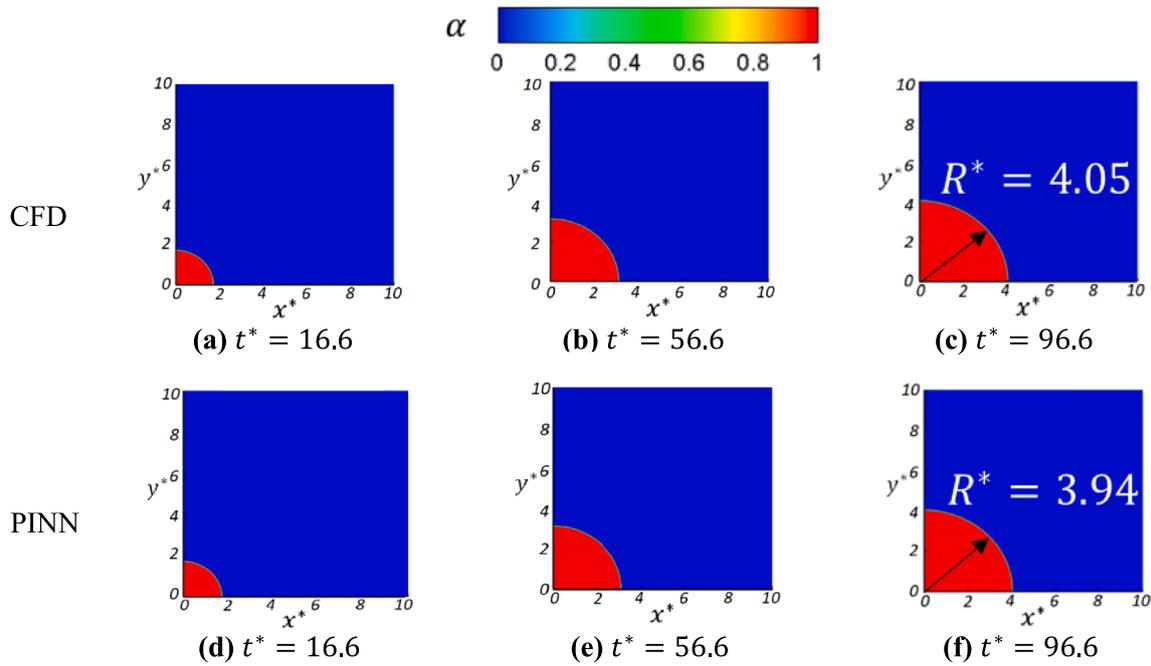


Fig. 2. Comparative analysis of CFD volume fraction (α) (without surface tension) and PINN Predictions for the 150×10 (optimal) architecture.

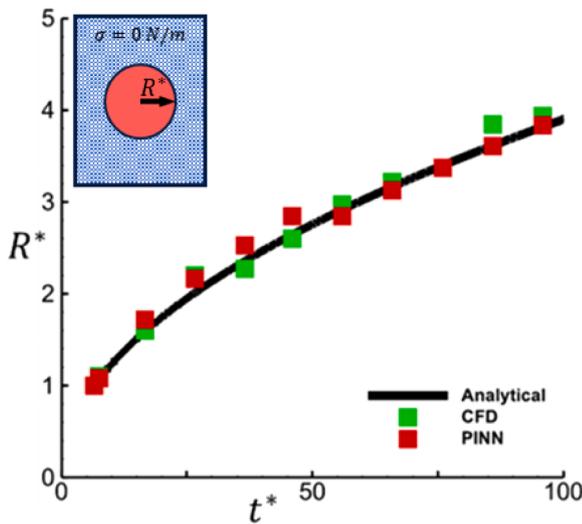


Fig. 3. Quantitative comparison of the analytical interface location evolution (R^*) against CFD and PINN predictions.

where δ is the thermal diffusivity of the liquid phase. The growth rate constant, β , varies depending on the level of superheat and relative densities of the liquid and vapour phases. For very small superheats, β is given by (19), otherwise, it can be assumed to be (20).

$$\beta = \sqrt{\frac{\Delta T}{2 \left(\frac{\rho_v}{\rho_l} \right) \left[\frac{L}{c_l} \pm \left(\frac{c_l - c_v}{c_l} \right) \Delta T \right]}}, \beta \rightarrow 0 \quad (19)$$

$$\beta \cong \sqrt{\left(\frac{3}{\pi} \right) \frac{\Delta T}{2 \left(\frac{\rho_v}{\rho_l} \right) \left[\frac{L}{c_l} \pm \left(\frac{c_l - c_v}{c_l} \right) \Delta T \right]}} \quad (20)$$

Table 2 displays the relevant properties of the working fluids which were liquid water and water vapour in this preliminary study. One such property is the Jakob number, which is evaluated using liquid properties

$c_{p,l}$ and ρ_l at T_{sat} . h_{lv} and ρ_v are also evaluated at T_{sat} .

A schematic of the Scriven problem is shown in Fig. 4.

3.1. Forward bubble growth study

Forward problems are defined by complete initial and boundary conditions [58]. They produce a unique solution which is insensitive to initial predictions of weight values in DL applications. In this work, the capabilities of the PINN algorithm are tested on two forward problems of the bubble growth of water vapour and R-134a vapour in superheated liquid water and liquid R-134a, respectively. In these cases, the initial condition is provided and various checkpoints in time are provided with all parameters relating to the motion of the interface (i.e., volume fraction, velocity, and pressure). At times when this data is not provided, the algorithm was required to rely on the physics it *learned* and infer the correct rate of growth. The inferred results have been presented in Section 4.1.

3.1.1. Water vapour bubble in superheated water

The first case to be assessed is the forward problem of water vapour growing in superheated water. This differs from the work performed in Section 2.2 by the inclusion of surface tension effects. Transformation of water to water vapour can occur in scenarios ranging from the mundane task of cooking food [59] to the application of liquid cooling of electronics [60]. As a result, there are many researchers interested in modelling this phase-change scenario. An analytical comparison of the bubble growth process is therefore necessary for many of these works. The computational domain for this study used a uniform hexahedral cell size of $1 \times 10^{-6}m$, resulting in a total of 116,579 mesh cells. PINN physics loss error was collocated at the same number of points. Boundary conditions can be found in Fig. 4. A comparison of the CFD simulation is made against the analytical solution in Section 4.1.1, produced using constants found in [43]. The PINN model was trained on a subset of the total temporal data. In this case, 50 % of the total data was used to train the model. The PINN predictions relied on trained models based on physical PDEs to determine interface locations. These assessment times fell between the temporal information provided by the CFD training data (i.e. the time at which the algorithm was tested was never observed by the PINN during training). Forward-problem PINN results

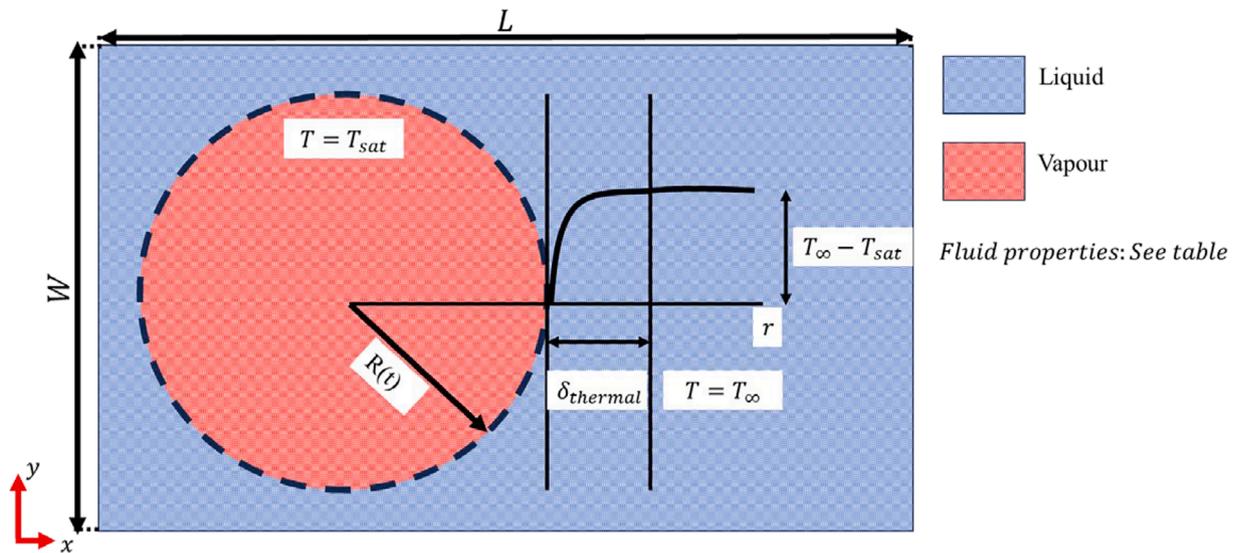


Fig. 4. Schematic of the analytical Scriven bubble growth problem. For water without surface tension, $R_0 = 1 \times 10^{-4}m$, $L = 0.5mm$ and $W = 0.22mm$. For water, R-134a, and FC-72 with surface tension $R_0 = 5 \times 10^{-5}m$, $L = 0.75mm$ and $W = 0.63mm$.

for water are discussed and compared against CFD and analytical solutions in Section 4.1.1.

3.1.2. R-134a vapour bubble in superheated water

R-134a is commonly used in refrigeration systems. It is currently being phased out due to its high global warming potential (GWP) value of 1430 [61] and has been banned from new vehicles in the EU since 2011 [62]. However, it is not possible to simply replace R-134a in systems designed for its specifications. This is because it is not flammable (as with all hydrofluoroolefins (HFOs)), nor can they withstand the critical pressure of many R-134a systems [63]. R-134a still forms a basis for many of the more environmentally friendly refrigerants because it is so effective [64]. Therefore, the modelling of R-134a is still of interest to researchers and industry alike. A similar process to that described in Section 3.1.1 was followed for forward modelling of R-134a vapour bubble growth.

The computational domain used a uniform mesh of 116,571 hexahedral cells with a side length of $1 \times 10^{-6}m$. CFD data was provided to train the PINN model, using 44 % of the available temporal data. The CFD solution was validated against the analytical solution, attaining a maximum discrepancy of 7.7 % at $t^* = 28$. The vast majority of the CFD simulated growth made minor over predictions of bubble growth rate until $t^* = 20$, where it began diverging more substantially. For rigour, CFD training data was provided in equally spaced intervals up to $t^* = 36$. Beyond this, the PINN algorithm relied upon trained weights to predict an extrapolative solution at $t^* = 38$. PINN physics error collocations were performed at the same 116,571 points as defined by the CFD grid. Forward-problem PINN results for R-134a are discussed and compared against CFD and analytical solutions in Section 4.1.2.

3.2. Inverse bubble growth study

In ML applications, the alternative to the ‘forward’ problem is the so-called *inverse* problem. Inverse problems are common across almost all scientific applications [65], as they arise whenever the quantity in question is not directly measured. To overcome the lack of direct measurement, a mathematical relationship must be defined to map the known quantities to those which must be discovered. FC-72 is a fully-fluorinated liquid, designed for electronics cooling [66]. FC-72 is non-flammable and non-toxic. It also evaporates without depositing residue on the heated surface. It is inert, and thus compatible with a wide variety of materials. FC-72 is therefore at the centre of many

studies performed in modelling liquid cooling systems. It is a better dielectric than water but requires more pumping effort than R-134a for an equivalent cooling load [67]. FC-72 also generally operates at higher pressures than R-134a. A complete list of relevant properties can be found in Table 2, however, there are several properties of major interest which makes the selection of FC-72 a challenge for the PINN methodology. The first is the density of both liquid and vapour forms of FC-72, which exceed densities observed in the training data by a factor of up to 1.5 in the liquid phase and up to 22.5 times in the vapour phase. The thermal conductivity of FC-72 liquid and vapour is much lower than either water or R-134a, as are also the surface tension σ and heat of vaporisation h_{lv} . Therefore, the PINN was required to utilise PDEs to successfully map this extrapolative behaviour.

To assess the inverse modelling capabilities of the PINN algorithm, additional investigations were performed using the models trained on water vapour and R-134a. The algorithm was trained sequentially using the R-134a ($Ja = 0.0289$, Section 3.1.2) dataset over 2000 epochs on a network initialised by the existing weights obtained from the forward study for water ($Ja = 0.000935$ – Section 3.1.1), before making predictions on the growth of a FC-72 bubble ($Ja = 0.0662$). This meant the PINN was trained for 4000 epochs in total, using 116,571 mesh cells and 50 % of the available temporal data, but vitally it did not observe any data for an equivalent growing bubble simulation where FC-72 was a working fluid. Through this method, PINN was confirmed as a robust method of mapping phase-change processes for canonical problems involving surface tension across a range of material properties which goes far beyond simple interpolation.

4. Discussion of Results

In this section, results obtained from PINN predictions are presented and bubble growth obtained by PINN is compared to both analytical and CFD-generated solutions.

4.1. Forward solution

4.1.1. Forward solution – water

PINN modelling of the Scriven problem using water produced accurate predictions of volume fraction and temperature distribution. Trends of predictions created using surface tension effects (for testing, $\sigma = 0.059 N/m$) are of a similar accuracy overall to those observed in the validation stage (Section 2.2, $\sigma = 0 N/m$). In both cases, the largest

discrepancy between predictions and the analytical result is present earlier in the growth process, similar to the CFD training data. For the growth of a water vapour bubble with surface tension, PINN predictions initially marginally underpredicted the growth rate, before it fell in line with the analytical solution. Fig. 5 shows the qualitative comparison between the CFD data used for training and PINN predictions at the same points in time. PINN predictions produced an interface with comparable sharpness to the CFD solution. While the PINN algorithm may have benefited from an explicit symmetry condition being specified on the domain boundaries, the bubble maintained a largely symmetrical growth without this being in place. Explicit enforcement of a symmetry condition would have served to reduce diffusion at the domain boundaries, however diffusion reduced as the predictions progressed in time. Part of this phenomenon can be accredited to the relatively large gap between the initial stationary condition and the first observed timestep, where the volume fraction predictions were excessively smoothed. More observation around the initial acceleration of the stagnant fluid would serve to improve early predictions by displaying the early development of advective behaviour. Increased accuracy of the velocity field beyond this point translated to an accurate representation of the Peclet effect in the convection-dominated regime, resulting in a sharpened interfacial region for subsequent predictions.

Fig. 6 confirms this assertion since the PINN prediction of the velocity field correctly ascertained aggressive circulation around the edges of the bubble (where $x^* \approx 6$ and $y^* \approx 6$). While the aggressive advection occurred at the boundary of the pure vapour volume fraction region, there was a lower velocity predicted in the area immediately surrounding the interface. The sharper velocity gradients shown in the PINN prediction caused dispersion across the fluid interface by slightly exaggerating the strength of the circulatory motion within the pure vapour region which was not entirely consistent with the CFD reference data. More observation at an earlier time would have helped to develop smoother gradients across the domain velocity field. Therefore, while the pure vapour region benefits from advective effects maintaining a circular bubble shape, the mixed-fluid zone experiences greater dispersion than the reference CFD data shows.

The time histories of the interface locations of the water vapour bubble obtained through analytical, CFD and PINN methods have been plotted in Fig. 7(a) (where $R^* = R_1/R_0$). A quantitative comparison between the analytical solution and PINN predictions revealed a maximum discrepancy of 3.6 % at $t^* = 6.5$. This compares favourably to

the CFD data, which has a maximum error of 4.47 % at $t^* = 20$. It is notable that beyond $t^* = 10$, the CFD solver began overpredicting the rate of vaporisation at the interface. It continued along this trajectory throughout the remainder of the simulation. However, the PINN algorithm initially deviated from the analytical solution earlier ($t^* = 6.5$) by underpredicting the rate of vaporisation at the bubble interface. Beyond this point, the error produced by the PINN model reduced slightly as time progressed. By $t^* = 21.5$, the PINN prediction aligned with the analytical solution. As part of the training process, the PINN algorithm observed 50 % of the available temporal CFD data. However, beyond $t^* = 25$, the PINN algorithm did not observe any CFD data. Therefore, it was necessary to predict an extrapolative solution at $t^* = 28$. This resulted in a minor overprediction of 1.82 % which remained closer to the analytical solution than the CFD solution managed, whose disparity to the analytical solution was 2.65 %. From this, it could be determined that the PINN methodology produced phase-change predictions of comparable accuracy to high-resolution CFD models. Fig. 7(b) confirms these assertions since the increases above fluid saturation temperature occurred only beyond the predicted pure vapour region. However, the diffusion visible in the volume fraction field was translated to the temperature field predictions. While the onset of increased temperature occurred at the correct radius, the diffused interfacial region caused a relatively gradual increase from saturation temperature to T_{max} . This diffusion culminated in the PINN predicted temperature field hovering around 0.99 (see localised view in Fig. 7(b)). Therefore, while the PINN temperature initially increased at the same rate as the CFD solution, the PINN predicted temperature required an additional 1.1 % distance to reach its maximum of $T = T_{max}$. This translated to a thermal interface thickness of 0.83 for the PINN prediction, compared to 0.64 for the CFD solution. However, this discrepancy was considered inconsequential to overall accuracy.

Therefore, this PINN methodology was deemed capable of solving the forward Scriven problem for water, and incorporation of the selected PDEs provided a demonstrable improvement over the CFD data used as a reference. This gave the confidence to proceed with the investigation into the forward Scriven problem for R-134a and the inverse investigation into the prediction of FC-72 bubble growth as the introduction of surface tension terms was not detrimental to PINN performance.

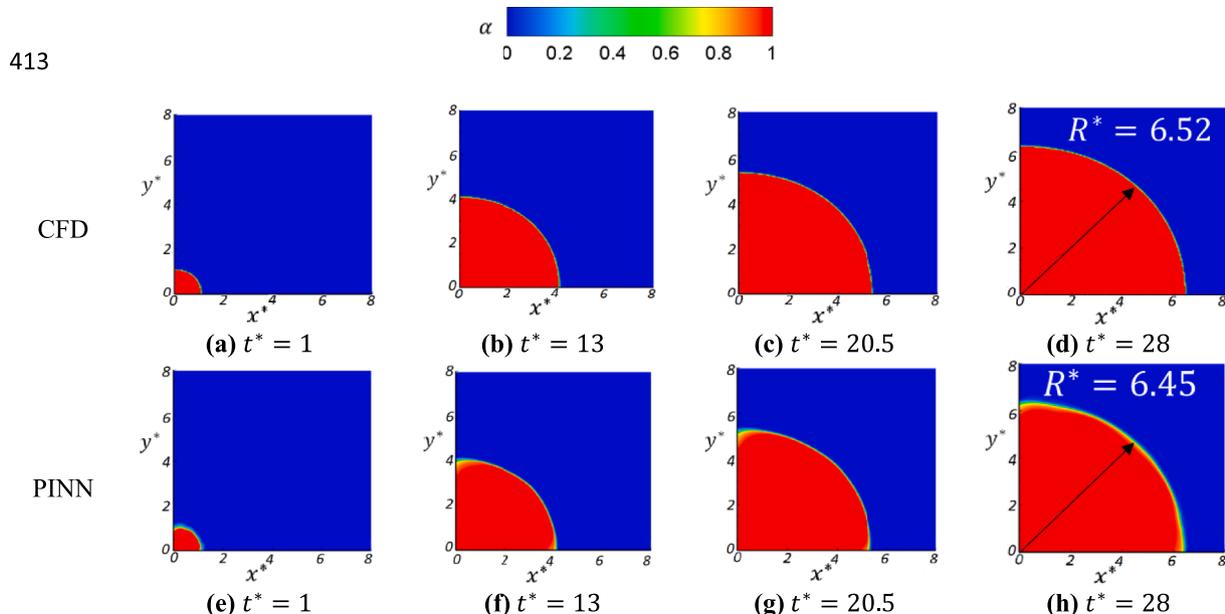


Fig. 5. Qualitative comparison of CFD (top row) and PINN (bottom row) results for volume fraction α for water vapour bubble growth.

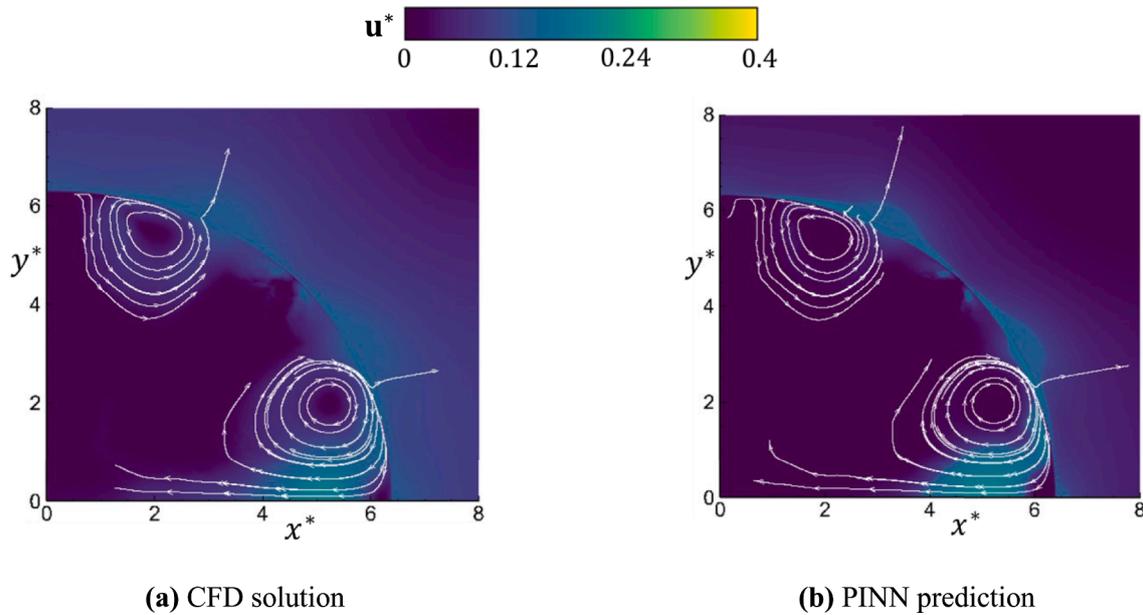


Fig. 6. Dimensionless velocity magnitude and streamlines during the evaporation of a water vapour bubble at $t^* = 28$ for (a) CFD reference data and (b) PINN inferred behaviour.

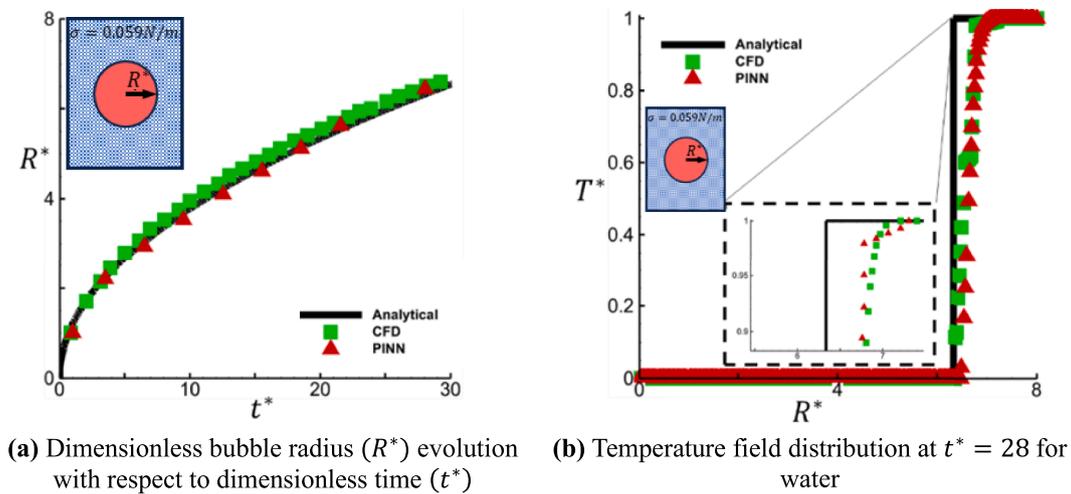


Fig. 7. Quantitative comparison of (a) interface location over time and (b) temperature field distribution for the water (with surface tension) PINN forward prediction compared against results achieved through analytical means and a reference CFD solution.

4.1.2. Forward solution – R-134a

The solution to the forward problem of R-134a evaporation was similarly successful when water was used as the working fluid. An additional 1000 epochs (a total of 3000 epochs) of training were required to enforce the maximum value of $\alpha = 1$. However, once this training was completed, the PINN results met the accuracy expectations set by the preceding work. For the duration of this case study, the PINN algorithm was able to predictions in line with the analytical solution. Fig. 8 displays the comparison of the CFD-generated volume fraction and PINN-generated volume fraction at a variety of times throughout the simulation.

Once again, the PINN was able to predict a sharp liquid-vapour interface. The interface generated for R-134a displayed less diffusion than the equivalent water study and was largely comparable to the sharpness presented by the CFD training data. The velocity magnitude and streamlines contours (Fig. 9) support this, since the PINN prediction

produced highly similar circulations approaching the interface within the pure vapour region. Compared to the CFD reference data, the circulations were in the correct position but displayed sharper gradients in magnitude. This is particularly noticeable around $y^* = 0$, where the flow velocity in the PINN prediction (Fig. 9(b)) exceeds that of the CFD solution (Fig. 9(a)). Immediately surrounding that circulation, the PINN-predicted velocity is lower than the CFD solution, however. This aggressive circulatory motion extended the thickness of the interfacial region compared to the CFD data because of the velocity discrepancy beyond the liquid-vapour interface, rather than desirable advection throughout the entire mixed fluid region. This was not detrimental to solution accuracy or overall interface position.

The level of accuracy is confirmed by Fig. 10(a), which revealed a maximum error in the PINN predictions of 6.47 % at $t^* = 26$. This compares favourably with the CFD-generated solution which produced a maximum error of 9.89 % at $t^* = 36$. Importantly, the PINN solution

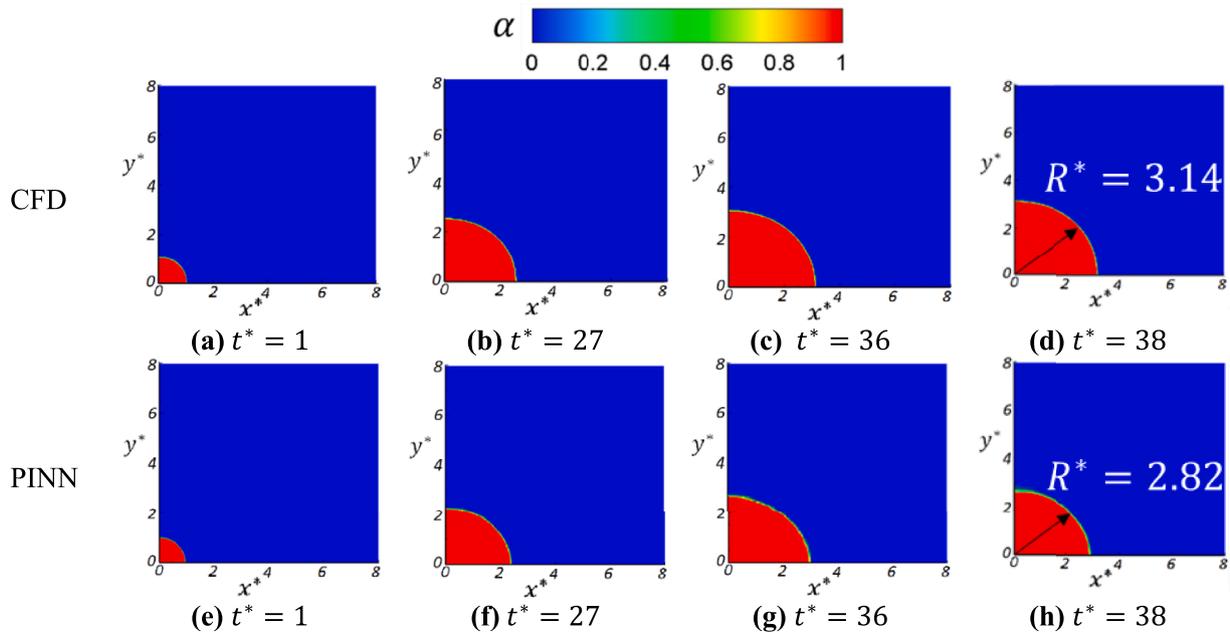


Fig. 8. Qualitative comparison of CFD (a-d) and PINN (e-h) results for volume fraction α for R-134a vapour bubble growth.

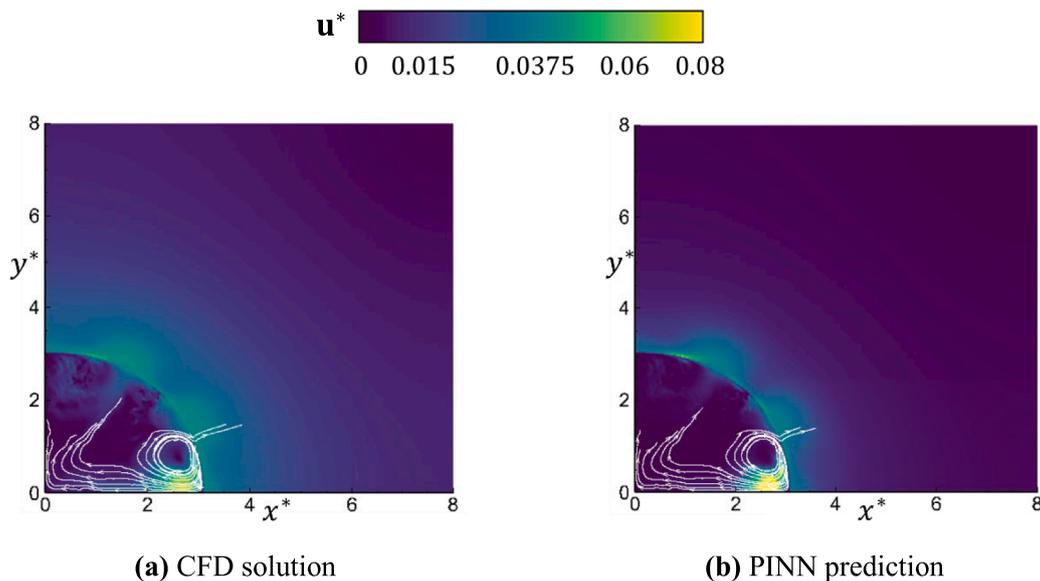


Fig. 9. Dimensionless velocity magnitude and streamlines during the evaporation of the R-134a vapour bubble at $t^* = 38$ for (a) CFD reference data and (b) PINN inferred behaviour.

produced a time history which more closely matched the analytical solution. The CFD solution began deviating from the analytical solution by overpredicting the rate of vaporisation at the interface beyond $t^* = 20$. In comparison, the PINN predictions provided a solution truer to the analytical solution with only minor oscillations about the true rate of growth. Fig. 10(b) validates the accuracy of the PINN predictions. Once again, the predicted temperature only increases beyond the saturation temperature ($T^* = 0$) within the interfacial region and beyond. This is consistent with the physics required by the analytical solution. Since the CFD solution modelled a larger vapour bubble at $t^* = 38$ compared to either the PINN prediction or analytical solution, the onset increasing temperature correspondingly occurred at a larger radius. The PINN prediction constructed the temperature field with a much steeper gradient across the interface in comparison to the CFD solution, whose

temperature field was significantly more diffused across the interfacial region. The PINN prediction provided a temperature field which was 8 % less diffused than the CFD solution overall by the time the PINN algorithm predicted the fluid reached T_{not} . The PINN algorithm predicted the thickness of the thermal interface to be 0.38, compared to the CFD-generated solution at 0.4. Both solutions were consistent with the analytical solution in predicting a constant maximum temperature within the liquid region.

4.2. Inverse solution

An extended study was performed to optimise the inverse capabilities of this PINN algorithm. This was necessary to determine the correct methodology for tuning the network weights and biases to predict fluid behaviour where material properties differ from those used for training. Three approaches were investigated. The first of these (hereby known as

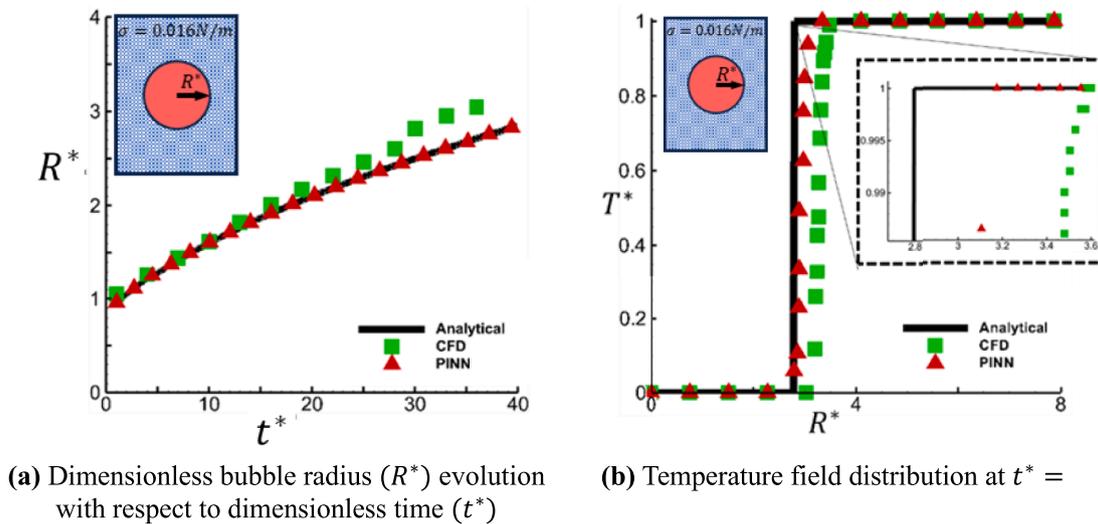


Fig. 10. Quantitative comparison of (a) interface location over time and (b) temperature field distribution for the R-134a PINN forward prediction compared against results achieved through analytical means and a reference CFD solution.

physics) relied solely on the governing PDEs and the initial condition of the bubble. The remaining methods used ‘prewarmed’ networks, which combined weights and biases generated from forward problems and some training on the physical properties of FC-72 using an initial condition defining the volume fraction and pressure fields at $t^* = 0$. The following combinations were used to assess predictions of prewarmed networks: weights and biases from the R-134a forward problem (hereby known as *R-134a*) and finally the combined weights and biases created by training the algorithm firstly using the water data, then by the R-134a data before training the algorithm on the material properties of FC-72 (hereby known as *combined*). These combined weights and biases were then augmented by training on the physical properties of FC-72 for 200 epochs. Prewarming of the algorithm using the weights from the forward problem for water was not assessed because of the disparity between the material properties of water and both R-134a and FC-72. The training specification for each approach is displayed in Table 3. The training time required by all PINN approaches was less than that required by the reference CFD solution, which needed 1640 core hours. To further improve the speed of preparation for prewarmed networks, future investigations will extend to the activation function and specifically a comparison between Rectified Linear Activation Unit function (ReLU or its derivatives) and the hyperbolic tangent function. ReLU is an increasingly popular function because it is computationally cheap and does not exhibit the vanishing gradient problems which exist to limit network architecture where ‘tan-h’ is the chosen activation function.

A complete quantitative comparison of the chosen approaches is shown in Fig. 11. As expected, most PINN approaches performed at their worst when tasked with predicting a timestep which had not been observed in training ($t^* = 38$), with the notable exception of the PINN-R-134a approach. Error for PINN-R-134a peaked at 21 % when $t^* = 12$. However, beyond this point, the R-134a approach tended to converge upon the analytical solution. The PINN-physics approach gave a

Table 3
Training specification for PINN-inverse approaches.

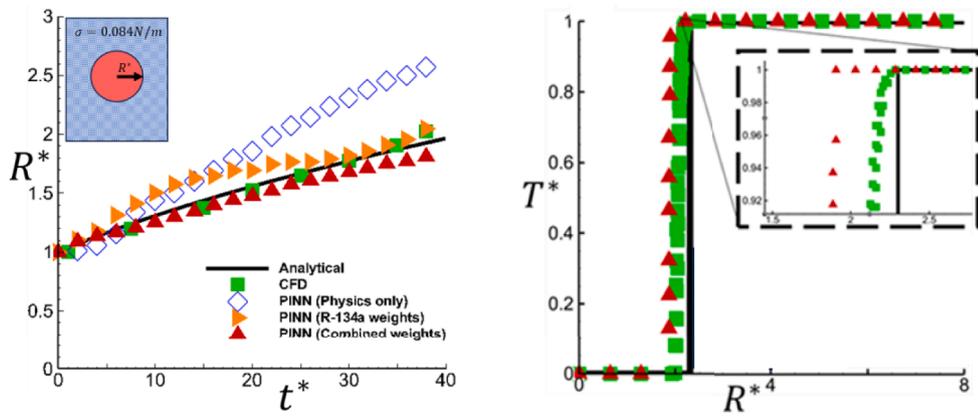
PINN approach	Training mechanism			Training time (core hours)
	Water weights/biases	R-134a weights/biases	Additional PDE training	
Physics	×	×	✓	64
R-134a	×	✓	✓	184
Combined	✓	✓	✓	306

reasonable forecast of bubble growth until $t^* = 14$, where it began to deviate significantly from the analytical solution.

Eventual error peaked at 33 % when $t^* = 38$. The PINN-combined approach provided a much better account of PINN inverse capabilities than the other two methods, providing results which were on par with the traditional CFD methodology. The bubble growth forecast provided by the combined approach showed no significant deviation from the analytical solution. While the CFD approach produced a maximum error of 4.7 % at $t^* = 38$, the PINN-combined approach compared well with a peak error of 6.1 % at the same timestep. The training process revealed that the MSE reached an optimum value of 1.2×10^{-5} using the combined approach. The underprediction of bubble radius by the inverse PINN is reflected in the corresponding combined temperature field (Fig. 11(b)).

As with other PINN temperature field predictions, the predicted temperature quickly reaches T_{max} beyond the pure vapour region. Notably, the transfer learning approach produced a temperature field prediction which resembled the desired analytical Heaviside step more closely than the CFD reference data. This is clear in the localised view (Fig. 11(b)), which highlights the relatively gradual transition of the CFD temperature field in comparison to the PINN prediction. The PINN thermal interface region thickness was predicted to be just 0.035, compared to 0.315 for the CFD prediction.

Fig. 12 shows a qualitative comparison between a CFD simulation of FC-72 bubble growth and PINN-combined predictions. The FC-72 CFD data was not observed by the PINN algorithm, it was only included here for comparative purposes. While the initial radius location on both the x and y-axes of the PINN-predicted FC-72 bubble was correct, the initial shape was deformed and not quite spherical. Training on relevant PDEs enforced the correct spherical nature of the vapour bubble at all subsequent timesteps. As a meshless solver, the PINN algorithm was able to produce an interfacial region with less diffusion than the comparative CFD simulation of an FC-72 vapour bubble. Fig. 13 displays that the velocity streamlines and contour of velocity magnitude of the predicted behaviour and the unobserved CFD data were very similar. The narrower interface produced by the PINN predictions can be credited to the highly concentrated circulation patterns, which tracked along the liquid-vapour interface. As the PINN algorithm could only resolve the bubble growth based on the supplied PDEs, the effects of timestep spacing from data supplied during the training process were minimised. Since no data fields were supplied to compare against the PINN algorithm resolved the interfacial region with less dispersion than the CFD solution, which closely followed the rapid flow circulations it predicted.



(a) Dimensionless bubble radius (R^*) evolution with respect to dimensionless time (t^*)

(b) Temperature field distribution at $t^* = 38$

Fig. 11. Quantitative comparison of interface location over time and temperature field distribution; (a) Dimensionless bubble radius (R^*) evolution with respect to dimensionless time (t^*) for FC-72, achieved through PINN predictions made using the physics, R-134a, and combined algorithms against analytical solution and an unobserved CFD prediction of FC-72 vapour bubble growth; (b) Temperature field distribution for the analytical solution, CFD reference prediction and combined PINN prediction at $t^* = 38$.

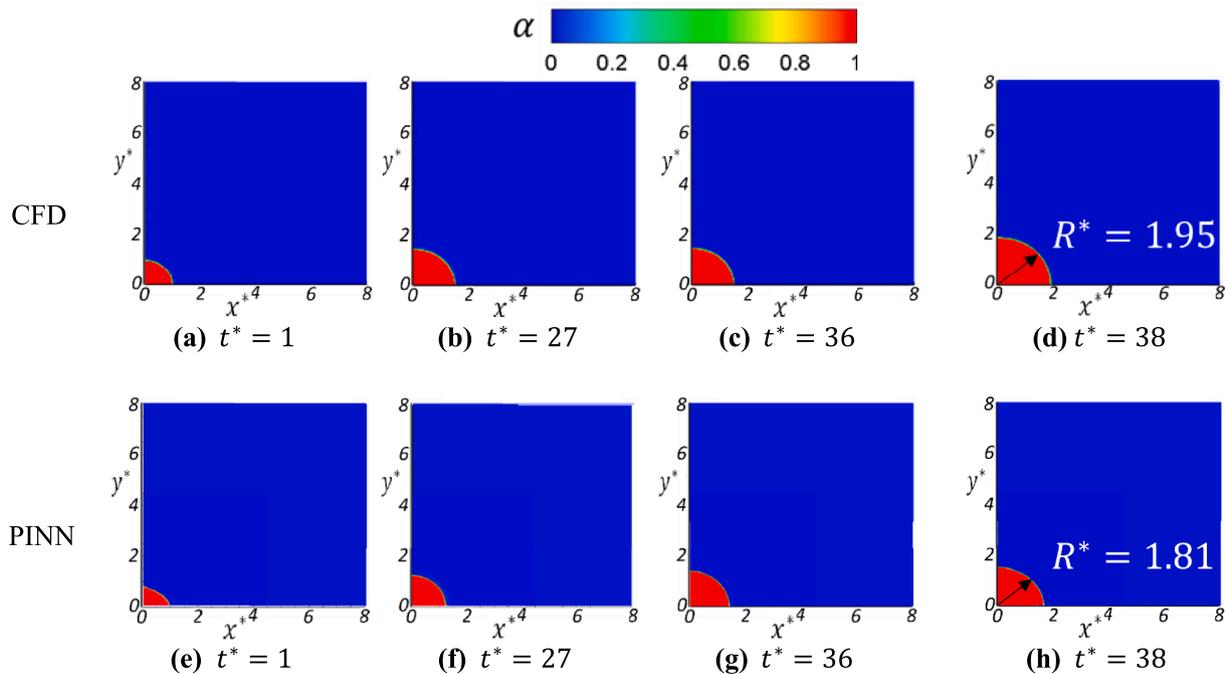


Fig. 12. Qualitative comparison of unobserved CFD (top) and PINN-combined (bottom) results for volume fraction α for inverse prediction of FC-72 vapour bubble growth.

This is confirmed by analysis of Fig. 13, which that the velocity magnitude gradients are smoother for the inferred case than the two forward cases. While the predicted velocity magnitude in the liquid region remains lower than in the CFD reference data, the difference is negligible. Due to the PINN algorithm being able to infer the steady development of the velocity profile across the interfacial region, the PINN results produced an interface of comparable sharpness to the CFD reference data and analytical solution. Despite this, the overall bubble size produced by PINN inference was slightly smaller than that produced by the analytical and CFD solution methods.

Though PINN methods have the capability of modelling problems by providing just the initial and boundary conditions, providing supplemental data significantly aids the inference process. The optimised

transfer learning approach has been shown to provide a feasible alternative to additional CFD studies in the parametric design process by being able to correctly predict the evaporation process of a vapour bubble during the isobaric growth stage. This study illustrates that the inverse capabilities of PINN methods can be unlocked to create a robust alternative to CFD methods when tasked with solving phase-change problems. Providing that the correct preparation is performed, the transfer learning PINN approach discussed in this work has the potential to be used to accelerate the design process of real systems reliant on boiling heat transfer and even become a part of a live response control system which calls upon learned trends and inferred physics to build a more complete picture of system operations than can be provided by discrete sensing locations. It is this compelling capability which will

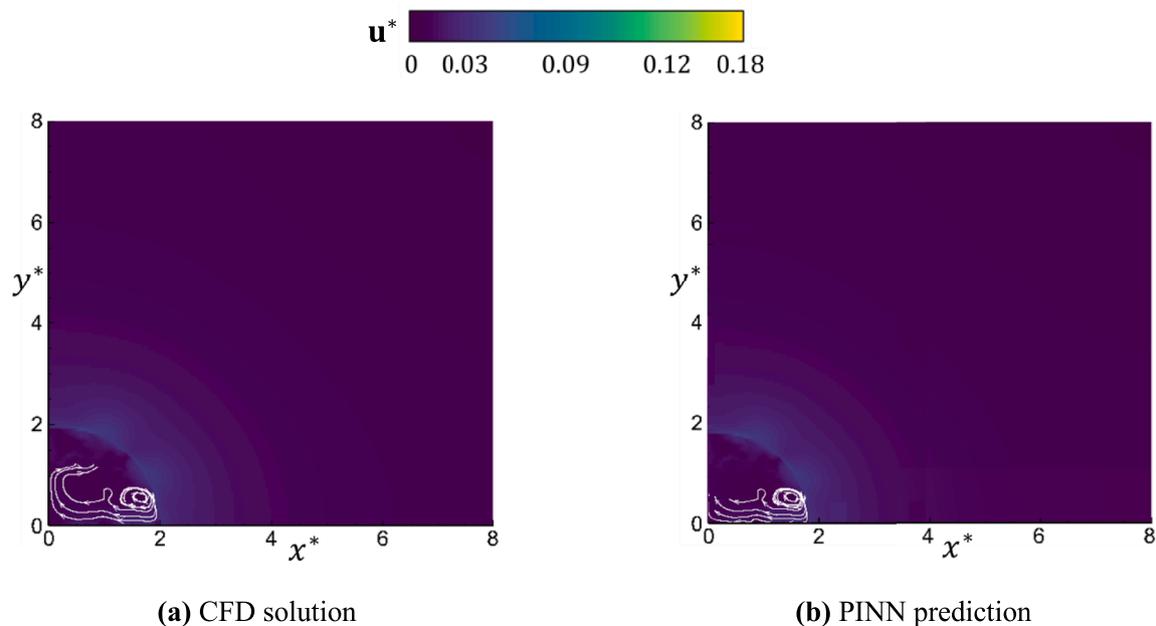


Fig. 13. Dimensionless velocity magnitude and streamlines during the evaporation of a FC-72 vapour bubble at $t^* = 38$ for (a) CFD reference data and (b) PINN inferred behaviour.

enable researchers and industry alike to reduce necessary safety margins and further optimise system performance.

5. Conclusions and future work

In the current study, a physics-informed neural network (PINN) has been developed to solve the process of evaporative phase change in a range of fluids and validated against both analytical solutions and CFD reference data for the evolution of vapour bubbles in superheated liquid for a range of working fluids. These findings demonstrate that the PINN approach is robust when modelling liquid-to-vapour phase change and accurately captures surface tension effects. As a result, the findings significantly enhance the capability of PINNs to be used as a method for solving boiling heat transfer problems.

The forward problem PINN volume fraction predictions for water-to-water vapour produced a peak error of 3.6 % which bettered the volume fraction error of 4.47 % produced by the traditional CFD reference solution. There was similar success for the forward problem predicting the evaporation of an R-134a bubble in superheated liquid R-134a, where PINN volume fraction predictions peaked at a percentage error of 6.47 %, which again exceeded the accuracy of the reference CFD solution whose error peaked at 9.89 %. PINN temperature field predictions convey the level of diffusion in the interfacial region displayed in PINN volume fraction predictions compared to the ideal analytical solution (where the interface is infinitely thin) and reference CFD solutions. The accurate inferences of the velocity field in both cases provided qualitative confirmation that the physics of the evaporation process was represented well, with circulations driven by interfacial advection clearly present, though diffusion across the interface was found to be caused by flow circulations which followed a sharp path around the pure vapour region. While excess interfacial interface diffusion caused an additional 1.1 % error in the temperature field compared to the CFD solution, all other studies demonstrated that the PINN prediction could achieve a thermal interface of reduced thickness and more closely aligned to the analytical solution than reference CFD data. This peaked at an 8 % reduction in thermal field diffusion for forward problems in comparison to reference CFD data. For these forward problems, the developed algorithm was also able to infer symmetry boundary conditions, meaning uniform bubble growth was predicted correctly. Additionally, the

interfacial diffusion produced by PINN predictions for forward problems using the 150×10 (layer \times node) architecture was low enough to be comparable to the diffusion present in reference CFD data.

The major highlight of this work is the extended inference study, which demonstrated the resilience of PINN in accurately determining the evaporation process and illustrated the level of agnosticism of this technique to relevant simulation properties such as saturation temperatures, conductivities, densities, and surface tension. The scope of the inference study included a comparison of the extrapolative and interpolative capabilities of the PINN algorithm, assisted by further training on relevant PDEs. It was found that by providing a spectrum of data to prewarm the PINN solver, accurate inferences could be made for different fluid properties. This resulted in an inferred solution for new fluid properties (FC-72) with a peak volume fraction interface location error of just 6.1 %, which was comparable to the equivalent CFD model. The CFD model bettered the inferred result by just 1.4 % for this case, despite the PINN algorithm observing no FC-72 data. Further, the thermal diffusion across the PINN-predicted bubble interface was more akin to the desirable Heaviside step of the analytical solution than the results achieved by the CFD solution. In this scenario, the inferred interfacial region was only around 10 % of the thickness compared to the interface produced by the reference CFD solution. This was found to be driven by aggressive advection behaviour on approach to the bubble interface. As with the PINN forward investigations, analysis of the inverse predictions revealed that the axisymmetric boundary conditions could be inferred. Thus, symmetric bubble growth was also predicted for the inverse investigation.

In future work, it may be beneficial to explore meshless VOF interface sharpening techniques which promise to further reduce the level of smearing at liquid-vapour interfaces. The authors also intend to utilise this phase-change algorithm to investigate more complex phase-change processes to enhance knowledge of boiling heat transfer and ease optimisation processes for real-world systems. Therefore, additional efforts in eliminating diffused interfacial regions may be required, since investigation of boiling heat transfer requires an extremely precise understanding of thermal properties at interfacial regions. The benefits of applying PINN techniques to boiling studies are two-fold; one advantage is the speed at which PINN methodologies will allow accurate parametric studies to be performed, while the overriding benefit will be the

opportunity to infer hidden variables where direct measurement is not possible. It is the latter outcome which provides great motivation to continue research into PINN methods. The ability to extend operators' knowledge of a system in real-time beyond information provided by fixed sensing locations has the potential to dramatically reduce the necessary safety margin required to maintain reliable system operation. In systems which rely on phase change, the implementation of PINN as a real-time control method would have profound implications and could significantly increase the heat transfer efficiency of such systems by minimising any necessary safety margins.

CRedit authorship contribution statement

Darioush Jalili: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis. **Mohammad Jadidi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Amir Keshmiri:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Bhaskar Chakraborty:** Resources, Data curation. **Anastasios Georgoulas:** Writing – review & editing, Validation, Methodology, Investigation, Funding acquisition, Conceptualization. **Yasser Mahmoudi:** Writing – review & editing, Supervision, Software, Resources, Project

Appendix

In this appendix, more information is provided pertaining to the computational resource required by both the CFD and PINN methodologies. For the investigations which form the basis of Section 4, CFD simulations were performed on an axisymmetric geometry containing 400,000 computational cells. Results were calculated using a remote-access Linux virtual machine high-performance computer (HPC). From this data, a 'slice' was extracted, containing 116,571 cells. Properties of velocity, pressure and volume fraction were tracked over time at equally spaced intervals of $t^* = 2$ until $t^* = 36$ (for R-134a) or $t^* = 26$ (for water). This represented at most 50 % of the available CFD timesteps. The PINN algorithm was trained on this data and used to predict the timesteps which were not observed during training. These were compared against the reference CFD data and analytical solutions.

The PINN algorithm was built using Keras (an API of TensorFlow for neural network development). Training was hosted on Manchester CSF3 (HPC), using 8 Nvidia v100 GPU nodes. Training was performed in parallel to expedite the process. The total computational resource (core hours) to produce solutions for both the CFD and PINN methods are displayed in Table A.1.

Table A.1
Tabulation of core hours for CFD-generated solutions and PINN predictions.

Methodology	Simulated fluid		
	Water	R-134a	FC-72 (Transfer learning)
CFD core hours	2400	1400	1624
PINN training epochs	2000	3000	See Section 4.2
PINN training core hours	184	184	306
PINN prediction core hours	0.03	0.03	0.03

Although training the R-134a solution required 1000 additional epochs compared to predicting water evaporation, both fluids utilized the same number of core hours. This is due to the slightly larger dataset used for training the water algorithm, which comprised 50 % of the available data, compared to 44 % for the R-134a solution.

References

- [1] S. Singh, S. Srivastava, M. Visaria, Numerical investigation of thermal performance of key components of electric vehicles using nucleate boiling, *J. Therm. Sci. Eng. Appl.* 13 (6) (2021) 061027.
- [2] S. Saleh, A. Mohammad, F.K. Al-Jubory, S. Barghi, CFD Assessment of uniform bubbly flow in a bubble column, *J. Petrol. Sci. Eng.* 161 (1) (2018) 96–107.
- [3] H.K. Gharitahre, R.K. Prasad, Application of ANN technique to predict the performance of solar collector systems - a review, *Renew. Sustain. Energy Rev.* 84 (1) (2018) 75–88.
- [4] K. Choi, H. Park, Interfacial phenomena of the interaction between a liquid–liquid interface and rising bubble, *Exp. Fluids* 62 (1) (2021) 126.
- [5] J. Shaikh, A. Sharma, R. Bhardwaj, On sharp-interface level-set method for heat and/or mass transfer induced stefan problem, *Int. J. Heat Mass Transf.* 96 (1) (2016) 458–473.
- [6] C. Reichi, S. Both, P. Mascherbauer, J. Emhofer, Comparison of Two CFD approaches using constant and temperature dependent heat capacities during the phase transition in PCMs with experimental and analytical results, *Processes* 10 (2) (2022) 302.
- [7] I. Perez-Raya, S.G. Kandlikar, Discretization and implementation of a sharp interface model for interfacial heat and mass transfer during bubble growth, *Int. J. Heat Mass Transf.* 116 (1) (2018) 30–49.
- [8] S. Cuomo, V.S.D. Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: where we are and what's next, *J. Sci. Comput.* 92 (1) (2022) 88.
- [9] S.A. Mohammadein, K.G. Mohamed, Growth of a vapour bubble in a viscous superheated liquid in two-phase flow, *Can. J. Phys.* 93 (7) (2014) 769–775.

- [10] R. Zenit, J. Rodriguez-Rodriguez, The fluid mechanics of bubbly drinks, *Phys. Today* 71 (11) (2018) 44–50.
- [11] M. Plesset, A. Zwick, The growth of vapour bubbles in superheated liquids, *J. Appl. Phys.* 25 (4) (1954) 493–500.
- [12] X. Dong, Z. Zhang, Mechanism study of bubble maximum diameter in the subcooled boiling flow for low-pressure condition, *Int. J. Heat Mass Transf.* 164 (1) (2021) 1–9.
- [13] J.D. Lopez-Lugo, J. Pimentel-Dominguez, J. Benitez-Martinez, J. Hernandez-Cordero, J.R. Velez-Cordero, F.M. Sanchez-Arevalo, Photomechanical polymer nanocomposites for drug delivery devices, *Molecules*. 26 (17) (2021) 1–23.
- [14] L.E. Scriven, On the dynamics of phase growth, *Chem. Eng. Sci.* 10 (1) (1959) 1–13.
- [15] H. Scheufler, J. Gerstmann, Heat and mass transfer in a cryogenic tank in case of active-pressurization, *Cryogenics* 121 (1) (2022) 103391.
- [16] L. Bures, Y. Sato, Direct numerical simulation of evaporation and condensation with the geometric VOF method and a sharp-interface phase-change model, *Int. J. Heat Mass Transf.* 173 (1) (2021) 121233.
- [17] G. Gennari, R. Jefferson-Loveday, S.J. Pickering, A phase-change model for diffusion-driven mass transfer problems in incompressible two-phase flows, *Chem Eng Sci* 259 (1) (2022) 117791.
- [18] G. Chen, T. Nie, X. Yan, An explicit expression of the empirical factor in a widely used phase change model, *Int. J. Heat Mass Transf.* 150 (1) (2020) 119279.
- [19] M. Raissi, P. Perdikaris, G.E. Karniadakis, Machine learning of linear differential equations using Gaussian processes, *J. Comput. Phys.* 348 (1) (2017) 683–693.
- [20] J.H.F. Pioch, A. Müller, F.-J. Peitzmann, D. Schramm, O. el Moctar, Turbulence modeling for physics-informed neural networks, *Fluids* 8 (43) (2023) 1–17.
- [21] A. Arzani, J.-X. Wang, R. D'Souza, Uncovering near-wall blood flow from sparse data with physics-informed neural networks, *Phys. Fluids* 33 (7) (2021) 1–10.
- [22] H. Eivazi, M. Tahani, P. Schlatter, R. Vinuesa, Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations, *Phys. Fluids* 34 (7) (2022) 1–11.
- [23] H. Gao, L. Sun, J. Wang, Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high resolution labels, *Phys. Fluids* 33 (7) (2021) 1–23.
- [24] V. Kag, K. Seshasayanan, V. Gopinath, Physics-informed data based neural networks for two-dimensional turbulence, *Phys. Fluids* 34 (5) (2022) 1–13.
- [25] F. Lorenzen, A. Zargaran, U. Janoske, Potential of physics-informed neural networks for solving fluid flow problems with parametric boundary conditions, *Phys. Fluids* 36 (3) (2024) 1–13.
- [26] S. Cai, Z. Wang, F. Fuest, Y. Jeon, C. Gray, G. Karniadakis, Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks, *J. Fluid Mech.* 915 (102) (2021) 1–17.
- [27] Q. Zhu, Z. Liu, J. Yan, Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks, *Comput. Mech.* 67 (1) (2021) 619–635.
- [28] S. Niaki, E. Haghghat, T. Campbell, A. Poursartip, R. Varzi, Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture, *Comput. Methods Appl. Mech. Eng.* 384 (1) (2021) 113959.
- [29] N. Masclans, F. Vasquez-Novoa, M. Bernades, R. Badia, L. Jofre, Thermodynamics-informed neural network for recovering supercritical fluid thermophysical information from turbulent velocity data, *Int. J. Thermofluids* 20 (1) (2023) 100448.
- [30] K. Kim, P. Hurley, J. Duarte, Physics-informed machine learning-aided framework for prediction of minimum film boiling temperature using a Physics-Informed Machine Learning-Aided Framework, *Int. J. Heat Mass Transf.* 191 (1) (2022) 1–15.
- [31] A. Jahanbakhsh, R. Firuznia, S. Nazifi, H. Ghasemi, Physics-informed neural network for thin-film evaporation in hierarchical structures, *Int. J. Heat Mass Transf.* 224 (1) (2024) 1–9.
- [32] P. Sharma, W.T. Chung, B. Akoush, M. Ihme, A review of physics-informed machine learning in fluid mechanics, *Energies* 16 (5) (2023) 2343.
- [33] S.B.L. Papadopoulos, S. Nikolopoulos, I. Kalogeris, V. Papadopoulos, A computational framework for the indirect estimation of interface thermal resistance of composite materials using XPINNs, *Int. J. Heat Mass Transf.* 200 (1) (2023) 1–14.
- [34] A. Royer, O. Farges, P. Boulet, D. Burot, A new method for modeling radiative heat transfer based on Bayesian artificial neural networks and Monte Carlo method in participating media, *Int. J. Heat Mass Transf.* (1) (2023) [Online]. Available, <https://www.sciencedirect.com/manchester.idm.oclc.org/science/article/pii/S0017931022010791>.
- [35] S. Xu, C. Yan, G. Zhang, Z. Sun, R. Huang, S. Ju, D. Guo, G. Yang, Spatiotemporal parallel physics-informed neural networks: a framework to solve inverse problems in fluid mechanics, *Phys. Fluids* 35 (1) (2023) 1–17.
- [36] Z. Liu, Y. Chen, G. Song, W. Song, J. Xu, Combination of physics-informed neural networks and single-relaxation-time lattice boltzmann method for solving inverse problems in fluid mechanics, *Mathematics* 11 (19) (2023) 1–29.
- [37] B. Zhang, G. Wu, Y. Gu, X. Wang, F. Wang, Multi-domain physics-informed neural network for solving forward and inverse problems of steady-state heat conduction in multilayer media, *Phys. Fluids* 34 (1) (2022) 1–17.
- [38] G. Radhakrishnan, A. Pattamatta, B. Srinivasan, Distributed Physics-Informed machine learning strategies for two-phase flows, *Int. J. Multiphase Flow* 177 (2024) 104861.
- [39] X. Zhao, R.K. Salko, K. Shirvan, Improved departure from nucleate boiling prediction in rod bundles using a physics-informed machine learning-aided framework, *Nucl. Eng. Design* 374 (1) (2021).
- [40] Y. Wang, L. Zhong, NAS-PINN: neural architecture search-guided physics-informed neural network for solving PDEs, *J. Comput. Phys.* 496 (1) (2024) 112603.
- [41] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1) (1981) 201–225.
- [42] S. Lan, An Evaporation Model for Interfacial Flows, MSc. Thesis, Mechanical Engineering, Eindhoven University of Technology, Eindhoven University Research Portal, 2011.
- [43] A. Georgoulas, M. Andreadaki, M. Marengo, An enhanced VOF method coupled with heat transfer and phase change to characterise bubble detachment in saturated pool boiling, *Energies* 10 (3) (2017) 272.
- [44] S. Hardt, F. Wondra, Evaporation model for interfacial flows based on a continuum-field representation of the source terms, *J. Comput. Phys.* 227 (1) (2008) 5871–5895.
- [45] D. Jalili, S. Jang, M. Jadidi, G. Guiustini, A. Keshmiri, Y. Mahmoudi, Physics-informed neural networks for heat transfer prediction in two-phase flows, *Int. J. Heat Mass Transf.* 221 (1) (2024) 125089.
- [46] C. Kunkelmann, K. Ibrahim, N. Schweizer, S. Herbert, P. Stephan, T. Gambaryan-Roisman, The effect of three-phase contact line speed on local evaporative heat transfer: experimental and numerical investigations, *Int. J. Heat Mass Transf.* 55 (1) (2012) 1896–1904.
- [47] C. Kunkelmann, P. Stephan, CFD simulation of boiling flows using the volume-of-fluid method within OpenFOAM, *Numerical Heat Transf., Part A: Appl.* 56 (8) (2009) 631–646. A.
- [48] C. Kunkelmann, Numerical Modelling and Investigation of Boiling Phenomena, PhD Thesis, Technische Universität Darmstadt Engineering Dept., Germany, 2011.
- [49] C. Kunkelmann, P. Stephan, Numerical simulation of the transient heat transfer during nucleate boiling of refrigerant HFE-7100, *Int. J. Refrigerat.* 33 (7) (2010) 1221–1228.
- [50] A.D. Jagtap, K. Kawaguchi, G.E. Karniadakis, Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks, in: *Proc. R. Soc. A* 476, 2020 20200334.
- [51] N.S. Keskar, R. Socher, Improving Generalization Performance by Switching from Adam to SGD, arXiv preprint, Ithaca, NY, 2017.
- [52] T. Gonsalves, Upadhyay, Integrated deep learning for self-driving robotic cars, in: *Artificial Intelligence for Future Generation Robotics*, Elsevier, 2021, pp. 93–118, ch 8.
- [53] Y. Rahul and S. Saha, A novel adaptive learning rate scheduler for deep neural networks, arXiv preprint, Ithaca, NY, 2019.
- [54] Y. Xiong, L. Lan, X. Chen, R. Wang, C. Hsieh, Learning to schedule learning rate with graph neural networks, in: *International Conference on Learning Representation*, Virtual, 2022.
- [55] F. Mucicchi, I. El Mellas, O.K. Matar, M. Magnini, Conjugate heat transfer effects on flow boiling in microchannels, *Int. J. Heat Mass Transf.* 195 (1) (2022) 123166.
- [56] S. Di Giorgio, S. Pirozzoli, A. Iafrafi, Evaluation of advection schemes and surface tension model for algebraic and geometric VOF multiphase flow solvers, *J. Comput. Phys.* 499 (2024) 112717.
- [57] K.J. Vachaparambil, K.E. Einarsrud, Numerical simulation of bubble growth in a supersaturated solution, *Appl. Math. Model.* 81 (1) (2020) 690–710.
- [58] M. Vauhkonen, T. Tavaainen, T. Lahivaara, Inverse problems, in: *Mathematical Modelling*, 1, Springer, 2017, pp. 207–227.
- [59] M. Tabacchi, C. Asensio, I. Pavon, M. Recuero, J. Mir, M.C. Artal, A statistical pattern recognition approach for the classification of cooking stages. the boiling water case, *Appl. Acoustic*. 74 (1) (2013) 1022–1032.
- [60] B. Benam, A. Sadaghiani, V. Yagci, M. Parlak, K. Sefiane, A. Kosar, Review on high heat flux flow boiling of refrigerants and water for electronics cooling, *Int. J. Heat Mass Transf.* 180 (1) (2021) 121787.
- [61] I. Hochst. R134a, its properties and alternatives, [https://www.infraserv.com/en/services/facility-management/expertise/f-gas/refrigerant/specific-refrigerant/r-134a.html#:~:text=R%2D134a%20\(chemical%20designation%3A,warning%20potential\)%20value%20of%201%2C430.](https://www.infraserv.com/en/services/facility-management/expertise/f-gas/refrigerant/specific-refrigerant/r-134a.html#:~:text=R%2D134a%20(chemical%20designation%3A,warning%20potential)%20value%20of%201%2C430.) (accessed 6/2/2024, 2024).
- [62] E. Comission, Refrigerants Used in Mobile Air Condition Systems (MAC) - State of Play, European Commission, 2024. https://ec.europa.eu/commission/presscorner/detail/en/MEMO_14_50 (accessed 6/2/2024).
- [63] R. Llopis, D. Sanchez, R. Cabello, J. Catalan-Gil, L. Nebot-Andres, Experimental analysis of R-450A and R-513A as replacements of R-134a and R-507A in a medium temperature commercial refrigeration system, *Int. J. Refrigerat.* 84 (1) (2017) 52–66.
- [64] I. Bell, P. Domanski, M. McLinden, G.T. Linteris, The hunt for nonflammable refrigerant blends to replace R-134a, *Int. J. Refrigerat.* 104 (1) (2019) 484–495.
- [65] A. Mohammad-Djafari, Regularization, bayesian inference, and machine learning methods for inverse problems, *Entropy* 23 (12) (2021) 1673.
- [66] 3M™ Fluorinert™ Electronic Liquid FC-72 [Online] Available: <https://multimedia.3m.com/mws/media/648920/3m-fluorinert-electronic-liquid-fc72-en.pdf>.
- [67] J.B. Campbell, L.M. Tolbert, C.W. Ayers, B. Ozpineci, K. Lowe, Two-phase cooling method using the R134a refrigerant to cool power electronic devices, *IEEE Trans. Ind. Appl.* 43 (3) (2007) 648–656.