Engineering Analysis with Boundary Elements 149 (2023) 52–70

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

A combination of large eddy simulation and physics-informed machine learning to predict pore-scale flow behaviours in fibrous porous media: A case study of transient flow passing through a surgical mask

Mehrdad Mesgarpour [a], Rabeeah Habib [b], Mostafa Safdari Shadloo [c], Nader Karimi [b,d,*]

[a] Fluid Mechanics, Thermal Engineering and Multiphase Flow Research Lab. (FUTURE), Department of Mechanical Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi (KMUTT), Bangmod, Bangkok 10140, Thailand
[b] James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, United Kingdom
[c] CORIA-UMR 6614, Normandie University, CNRS-University & INSA of Rouen, 76000, Rouen, France
[d] School of Engineering and Materials Science, Queen Mary University of London E1 4NS, United Kingdom

ARTICLE INFO

Keywords:
Physics-informed machine learning
Large eddy simulation
Pore-scale
Fibrous porous media

ABSTRACT

A predictive method using physics-informed machine learning (PIML) and large eddy simulation (LES) is developed to capture the transient flow field through microscale porous media (PSPM). An image processing technique extracts the 3D geometry of the internal layers of the mask from 2D microscopy images, and then the fluid flow is first simulated numerically. The subsequently developed PIML method successfully predicts the transient flow patterns inside the porous medium. For the first time, 3D maps of time-dependent pressure, velocity, and vorticity are predicted across the fibrous porous medium. The results show that, compared to conventional computational fluid dynamics, the PIML method can reduce the computational cost by over 20 times. Further, the LES model can replicate the fine fluctuations caused by the flow passage through the porous medium. Therefore, the developed methodology allows for transient flow predictions in highly complex configurations at a substantially reduced cost. The results indicate that the PIML method can reduce the total computational time (including training and prediction) by 22.5 and 20.7 times over the standard numerical simulation, based on speeds of 0.1 and 0.5 m/s, respectively. Several factors including the inherent differences between CPUs and GPUs, algorithms and software, appear to influence this improvement.

1. Introduction

For centuries, mathematical methods have been among the most effective approaches to solving the most challenging problems [1]. In recent years, introducing high-performance processors and utilizing machine learning have significantly improved computational methods [2]. Inspired by these, the current study focuses on developing a method based on physics-informed machine learning to predict the transient, pore-scale hydrodynamics in porous media. This primarily reduces the heavy costs incurred by the numerical simulation of such flows. Yang et al. [3] investigated the distribution, evolution, and influencing factors of the remaining oil in the pore space after water flooding. The results showed that the complex pore structure would lead to oil remaining in the water during flooding. A phase circulation phenomenon was observed, which included a co-current lid-cavity-driven flow. In addition, phase recirculation resulted in greater viscous dissipation.

In addition to microscopic flow mechanisms and traditional experimental studies of tight oil reservoirs, it is important to understand how wettability influences spontaneous imbibition on a pore-scale level. As demonstrated in [4], the wettability of the imbibition fronts significantly impacts their morphological characteristics. During conditions of strong hydrophilia, wetting fluids preferentially invade the pore corners as angular flows. Some studies require multilayer calculations, including optimization and prediction, such as internal combustion engines [5]. An excellent example of big data post-processing is modern engines. To develop a low-fuel consumption internal combustion engine, the details of the problems of increasing efficiency and lifetime and reducing energy consumption should be considered [6]. This generates a tremendous amount of data [7]. Additional limitations include material and

Abbreviations: ANN, Artificial neural networks; MSE, Mean square error; LES, Large-eddy simulation; ML, Machine learning; MLP, Multilayer perceptron; LPM, Litter per minutes; PIML, Physics-informed machine learning; LES, Large eddy simulation.

* Corresponding author.
E-mail address: n.karimi@qmul.ac.uk (N. Karimi).

https://doi.org/10.1016/j.enganabound.2023.01.010
Received 26 September 2022; Received in revised form 9 January 2023; Accepted 11 January 2023
Available online 16 January 2023
0955-7997/© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Nomenclature

A	Surface area, [m^2]
A_o	Outside surface area, [m^2]
A_{min}	Minimum free flow area, [m^2]
$C_{3\varepsilon}$,$C_{1\varepsilon}$	Constant parameter
f	Friction factor
f_j	Values of j-th neuron of the hidden layer
f*	Approximate function
g	Activation function
k	Turbulence kinetic energy, [$m^2.s^{-2}$]
L	Length, [m]
N*	Index of the columns of the result matrix
n	Number of data points
O_i	Model output for sample i
p	Pressure, [Pa]
S_ε	Constant parameter
s*	Stride in convolutional neural networks
t	Time, [s]
u, v, w	Velocity in x,y and z directions, RANS, model, [$m.s^{-1}$]
u_x,v_y,w_z	Velocity in x,y and z directions, LES model, [$m.s^{-1}$]
x, y, z	x, y and z directions
V_{max}	Maximum velocity, [$m.s^{-1}$]
w	Weight of parameter
x_i	Values of the i-th neuron of the input layer

Greek symbols

τ_{xx},τ_{yx}, τ_{zx},τ_{ij}	Shear stress, [$N.m^{-2}$]
ρ	Density, [$kg.m^{-3}$]
μ	Viscosity, [$kg.m^{-1}.s^{-1}$]
σ	Fraction of the minimum free flow

working conditions that alter the pattern of the optimum target state [8]. In recent years, data science and numerical simulation have provided new perspectives on big data challenges [9]. Researchers have combined machine learning [10], deep learning [11], and artificial intelligence [12] to predict and analyze the behavior of complex systems. This hybridised method has impressive abilities to reduce the calculation cost and make long-term predictions. The computational cost is usually the most significant challenge when a large domain or complex geometry is involved [13].

For this reason, large domains and complex geometries are not desirables for numerical simulations. To determine how permeability estimations correlate with properties of porous media, including porosity and specific surface area, Basbug et al. [14] employed artificial neural networks (ANNs). According to the results, the permeability estimates generated by the ANN are consistent with those generated by the core analysis. Carbonate reservoirs exhibit complex relationships between permeability, porosity, specific surface area, and irreducible water saturation. Accordingly, the magnitude of irreducible water saturations is modified by changes in specific surface area, implying a connection between permeability and irreducible water saturation. Despite these achievements, severe challenges, including non-laminar flows [15], flow in microchannels [16] and processing of big data [17] still persist. Advanced neural networks like physics-informed machine learning (PIML) [18] and deep forwarded neural networks-boundary conditions (DFNN-BC) [19] are considered appropriate for tackling these challenges. Therefore, the current work aims to apply physics-informed machine learning to partial differential equations (PDEs) and enable the neural network to learn the solutions. Some of the first works in this area date back to the 1990s.

Modern neural network architectures have rekindled interest in this subject. PIML approaches are designed to provide data-driven solutions to nonlinear systems of equations and may offer an effective alternative to more conventional numerical methods for solving systems of equations, such as finite-difference and finite-volume. In PIML, the underlying governing equations are encoded by the developed neural network as prior knowledge is used for training. The neural network approach makes it possible to approximate any continuous function. A few classical fluid dynamics problems including Burgers' equation, Navier-Stokes were solved by Raissi et al. [20] using PIML. With a large number of collocation points and an extensive neural network architecture, the v improves the Reynolds stresses. It remains as a priori study, though, since it has yet to be determined if these enhanced Reynolds stresses can be propagated to produce a better velocity field.

To model turbulence predictively, they demonstrate a comprehensive framework of Physics-Informed Machine Learning (PIML), which included machine learning Reynolds stress discrepancy functions, predicting Reynolds stresses in different flows, and propagating them to mean flow fields. The development of PIML methods for various configurations and flow conditions has been the focus of numerous studies in recent years [21–24]. Developing a PINN method for pore-scale materials necessitates the inclusion of boundary conditions in order to solve the loss function equations. PIML models for pore-scale materials are difficult to develop because the matrix of geometry in pore-scale materials must be updated at each time step. Fully turbulent approaches to the PIML, on the other hand, require additional equations and boundaries.

This study develops a hybrid prediction method based on LES and PIML to predict a transient flow passing through a fibrous porous medium (a surgical mask). After 3D geometry extraction and verification, a transient flow passing through six distinct layers of a surgical mask is simulated numerically. The outcomes of this step are then utilised for the training layer. The transient matrix of parameters provides sufficient material for automatic differentiation based on the FNN method. To combine the LES and PIML, we developed a new technique for converting the intricate 3D geometry of fibers into a series of 2D layers. According to the boundary matrix, the PDE loss function regenerates initial and boundary conditions. The final combination of PIML and LES creates an entirely new basis for calculating the flow through complex geometries.

2. Problem statement and methodology

2.1. Mask imaging

In order to study the flow patterns through the micro-scale, fibrous porous media, the numerical simulations should be pushed to their limits. Flow is incorporated in microchannels in the layers of the mask, whereas the porosity and distribution of fibres in different locations are random. The goal is to examine the flow pattern through the pores with different boundary conditions. In order to create precise 3D geometry from 2D microscopic images of microporous media, a geometry detection approach is required [25]. Fig. 1a shows the results of converting 2D microscopic images of surgical masks to 3D geometry. According to Fig. 1b, cloud points of different layers were generated using image processing, curvature correction algorithms, and geometry regeneration algorithms (GRA) [26] to extract 3D geometry from microscopic images (Fig. 1b). The different diameters of fibres in this case study allowed six different layers of masks to be detected, while the features and specifications of each layer were distinct (Fig. 1c).

The properties of the surgical mask are based on those in Ref. [27]. Machine vision [28] extracts microscopic images of fibres and converts them to cloud points [29]. It helps in representing a boundary for fibres

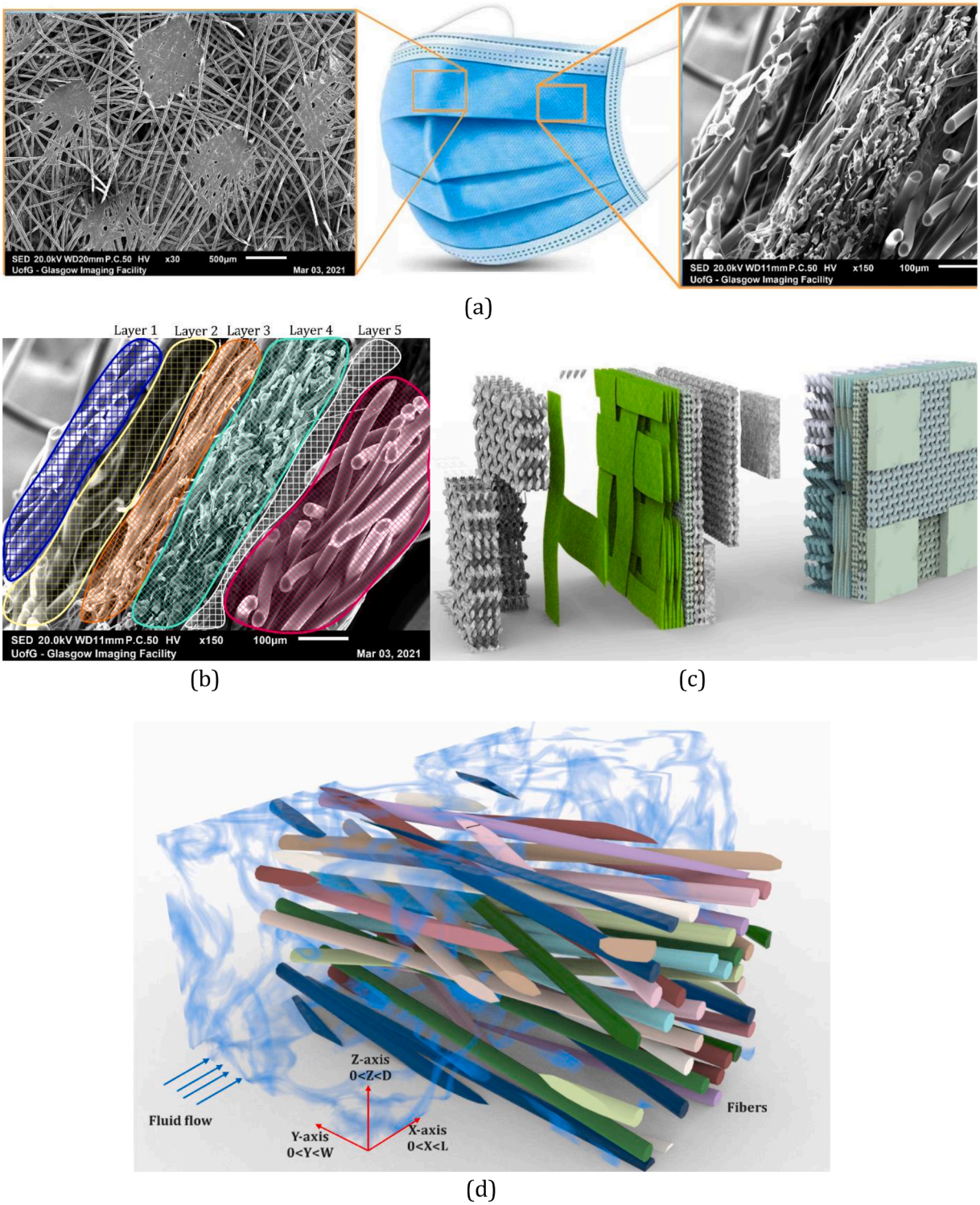


Fig. 1. Procedure for the extraction of 3D geometry form a) microscopic images of a surgical mask, b) image processing-based detected layers of mask based on fibres diameters and air gap, c) final 3D geometry, and d)graphical illustration of flow through the pore-scale fiber porous media.

[30]. The average fiber diameter and pore dimension have been determined to be within the parameters reported by Leonas et al. [31]. The quality of the generated three-dimensional geometry depends on the clarity of microscopic images [31], see Table 1 for further information. According to Fig. 1c, each layer has uniformly distributed porosity.

Furthermore, there is constant space between the layers based on the microscopic images. As shown in Fig. 1c, ultrasonically welded points are considered a square cross-sectional blanket on the first layer. This blanket has zero porosity, and Fig. 1d shows how fluid flow passes through the fibres through the channel with a 4mm^2 cross-section by removing the effects of fiber deformation. The mask layer was modelled in a square-cross-sectional tube ($2 \times 2 \times 0.7$ mm). A transient flow of air, with inlet flow velocities of 0.1 and 0.5 m/s, passes through the mask layers.

The surgical mask fibres include PSPM. Surgical mask properties and specifications are determined by Neupane et al. [32]. An image

Table 1
Geometrical properties of PSPM.

Layer	Pore-size (*μm*)		Layer thickness (*mm*)		
	Maximum	Minimum	Maximum	Minimum	Average
1	41.18	34.63	0.251	0.214	0.236
2	Space between Layers				
3	22.36	17.22	0.211	0.185	0.193
4	19.28	11.39	0.264	0.241	0.251
5	Space between Layers				
6	43.98	36.58	0.267	0.245	0.251
Total Surgical mask	43.98	24.95	0.267	0.185	0.232

Table 2
Specification of surgical mask.

Fiber diameter (mm)		Minimum pores dimension (*μm*)		Maximum pores dimension(*μm*)	
Case study	Leonas et al. [31]	Case study	Leonas et al. [31]	Case study	Leonas et al. [31]
0.323	0.1921	41.69	19.29	49.82	43.27

processing method, as explained by Raje et al. [33] was used to create the geometry from microscopic images. A detailed description of PSPM is given in Table 2. The JSM-IT100 in Touch ScopeTM Scanning Electron Microscope is shown in Fig. 2. Different mask slices were prepared to capture microscopic images at different angles.

2.2. Geometry extraction method

Fig. 3a shows the addition of a convolutional layer to an image of surgical mask layers. The photo's original resolution was 1600 × 1200 pixels. The top and cross-sectional views of each metal foams' top were analysed to determine the best results. Fig. 3b shows how three-dimensional geometry was converted from microscopic images. Point clouds could be generated from convolutional matrices [34]. By using commercial CAD software (i.e., SolidWorks), these points can be converted to solid geometry. The application of this method improves the prediction quality and enables a better understanding of the flow patterns generated by PSPM. According to Fig. 3b, the original picture was used as input data.

In digital image processing, image acquisition is the first step. It may be possible to obtain an image by receiving it in a digital format. Scaling and rotation are typically performed prior to image acquisition. For quality assessment, image enhancement methods were applied. Digital image processing is characterised by its straightforward and visually appealing qualities. An optimization method is fundamentally meant to reveal previously hidden detail or highlight certain aspects of an image. Image restoration is sometimes recommended when the image quality is low or there is a lot of noise.

Restoration techniques are objective as they are based on mathematical or probabilistic models of how an image degrades, while an enhancement technique is subjective. Following this phase are color image processing, wavelets and multi-resolution processing, compression, and morphological processing. A two-dimensional image can be converted through image processing to a two-dimensional point cloud. The values of the following feature maps are calculated based on the following formula: f^{**} represents the input image and q^{*} represents our

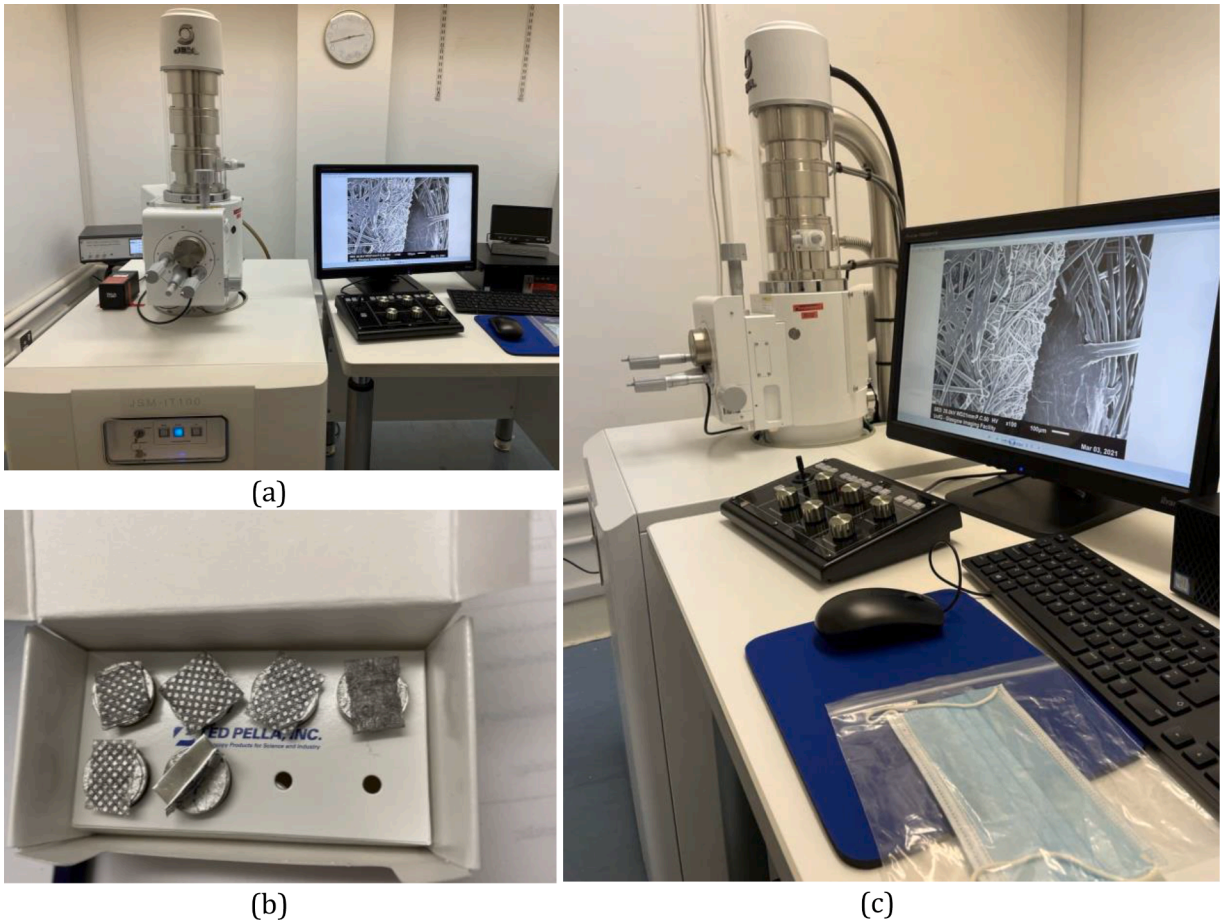


Fig. 2. Microscope and test sections.

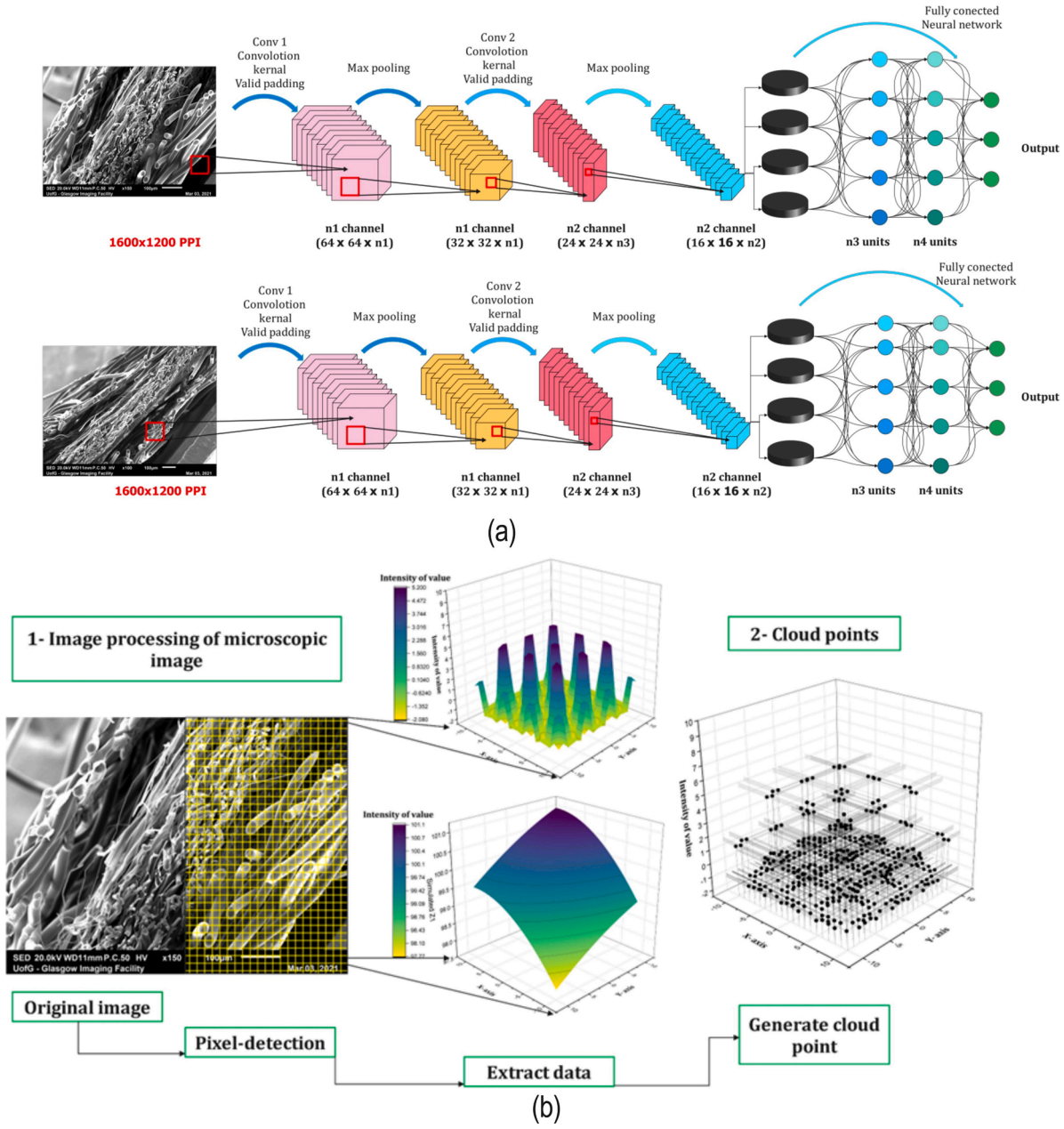


Fig. 3. Geometry extraction method, a) Details of convolutional layers, b) Steps of GEM and final geometries, and c) Rendered geometry of PSPM.

kernel. M^* and N^* are the indices of the rows and columns of the result matrix, respectively [33].

$$G[M^*, N^*] = (f^* \times q^*)[M^*, N^*] = \sum_j \sum_k q^*[j,k] f_i^*[M^* - j, N^* - k]. \quad (1)$$

When $p^{**}$is padding and f_i^* is filter dimension, this equation should satisfy the padding width [35]

$$p^{**} = \frac{f_i^* - 1}{2}. \quad (2)$$

By taking padding and stride into account, the following formula can be used to determine the dimensions of the output matrix.

$$n^{out} = \left[\frac{n^{in} + 2p^{**}}{s^*} + 1\right]. \quad (3)$$

A 3D matrix (or tensor) has the dimensions given by the following equation,

$$[n^*, n^*, nc^*] \times [f_i^*, f_i^*, nc^*] = \left[\left[\frac{n^* + 2p^{**}}{s^*} + 1\right], \left[\frac{n^* + 2p^{**} - f_i^*}{s^*} + 1\right], nf^*\right]. \quad (4)$$

In the core of GOM, layers of calculation and prediction are involved. These layers are defined as:

1 The Point Cloud Data
2 Geometric model selection
3 Parameter's Definition
4 Select a dataset wisely
5 Manual parameters configuration
6 Automatic parameters configuration
7 Model fitting with RANSAC
8 Point Cloud Binary Segmentation

Cheminformatics and material-informatics rely heavily on

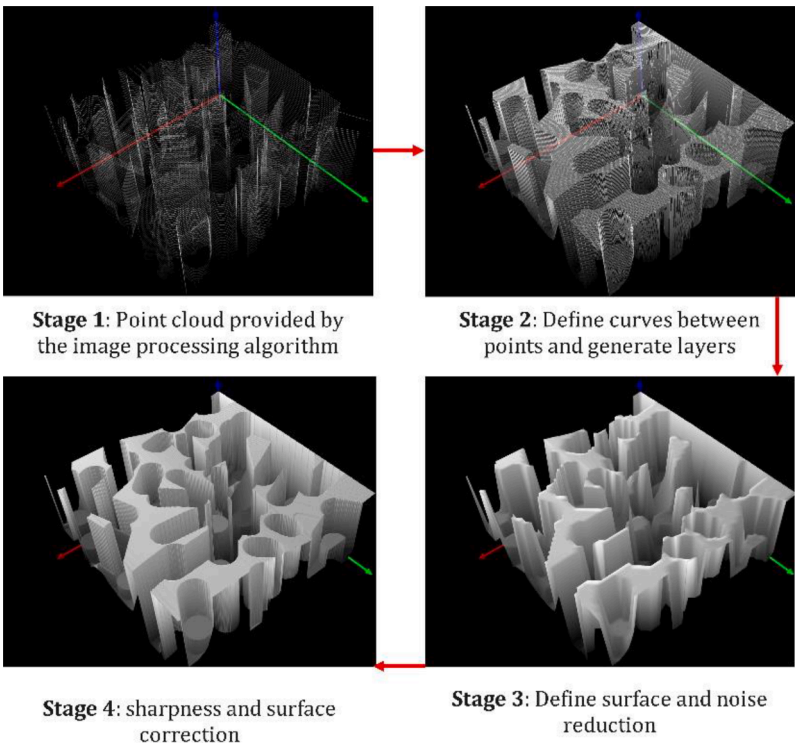


Fig. 4. Stages of converting point clouds to 3D geometry.

quantitative structure-activity relationships (QSAR) models constructed using machine learning algorithms. It is a widely used algorithm in image processing to remove noise from data sets using RANDOM SAmple Consensus (RANSAC) [36]. In addition to outlier removal, descriptor selection, model development, and predictions for test set samples, RANSAC could serve as an effective "one-stop shop" algorithm for developing and validating QSAR models. In the presence of numerous outliers in the data, RANSAC provides robust model fitting. According to Fig. 4, point clouds will be provided via image processing methods in 3D coordinates. Based on a series of microscopic images, RANSAC will convert point data to curves. As a result of this stage, layers of 3D geometry will be generated.

The third stage will transfer the curves and layers to the surface. In this stage, fibres and fluid domains can be generated into a 3D geometry. The final stage is developed to improve quality and reduce the noise of point clouds. Mao et al. [37] present details of noise reduction in ML method.

The input of a convolutional neural network (CNN) model is an image (x) and the output is cells of a specified size (y). The objectives of neural networks are regression and label prediction. Before being input into the CNN model, images were resized to 800 × 600 pixels. A normalized average value and standard deviation are also applied to every pixel in each RGB channel of the image. CNN was chosen because they provide a dependable image processing solution. PyTorch was utilised to build the CNN architecture. The neural network was constructed using 5 layers of learnable weights. A dense, fully connected layer was one with multiple inputs, whereas a convolutional layer had a single input. The final fully connected layer outputs a tuple of predicted horizontal and vertical cell sizes. Basic convolution layers consist of a convolution operation, a nonlinear activation function (rectified linear unit, or "ReLU"), and a max-pooling operation. Due to the max-pooling layer, the width and height of the image were reduced by a factor of four. The image size should therefore be divisible by two multiples. By utilizing batch normalization layers, learning could be accelerated, and generalization loss (or testing loss) could be reduced. Consequently, a batch normalization layer was implemented after the convolution operation and prior to the nonlinearity of the ReLU within the convolutional layer. In order to preserve their size, the layers that perform convolution were padded with zeros prior to passing their output to the max-pooling operation. Padding protects an image's corners from information loss and output size reduction. The second fully connected layer was given a dropout probability of 0.217 to prevent the network from becoming overfitted. In practice, the dropout layer further reduces generalization loss. 80% to 20% of the images in this 5-image dataset are used as a training set, while the remaining images are used as a test set. The network acquires knowledge through the modification of its filters. CNN weights are defined and modified in accordance with their hyperparameters. The network contains 1126 learnable parameters or weights in total. To perform a numerical analysis of the PSPM, its exact three-dimensional geometry was required. CNNs are a type of deep learning algorithm that analyses an image for various aspects/objects, assigns importance (learned weights and biases), and distinguishes between them. Compared to other classification algorithms, CNNs require considerably less pre-processing [38]. Typical 3D geometry methods cannot support the geometric specification of PSPM [39,40] due to the random distribution of pores. Through the combination of image processing (coevolutionary method) [41,42] and point cloud processing [30], this study converted 3D point clouds to geometry [34]. Image processing can convert a two-dimensional image to a two-dimensional point cloud.

The current analysis assumes stationary fluid at $t = 0$ s when a uniform flow front with a prescribed constant velocity starts passing through the mask layers. This sets a transient flow, which is simulated at $t = 2$ s. A combination of numerical simulation and PIML is assumed to identify the parameter pattern when transient flow passes through the PSPM. A turbulence model is applied to the extracted geometry when the inlet velocity is 0.1 m/s (Re_{inlet}=894). Due to the variation in porosity and concentration of fibres in different layers, the conventional approach to numerical simulation, such as scaling, did not produce the desired results. The present study assumed that full-scale geometry could determine the pattern of parameters such as pressure and velocity through and between layers. Although this method increased the cost of

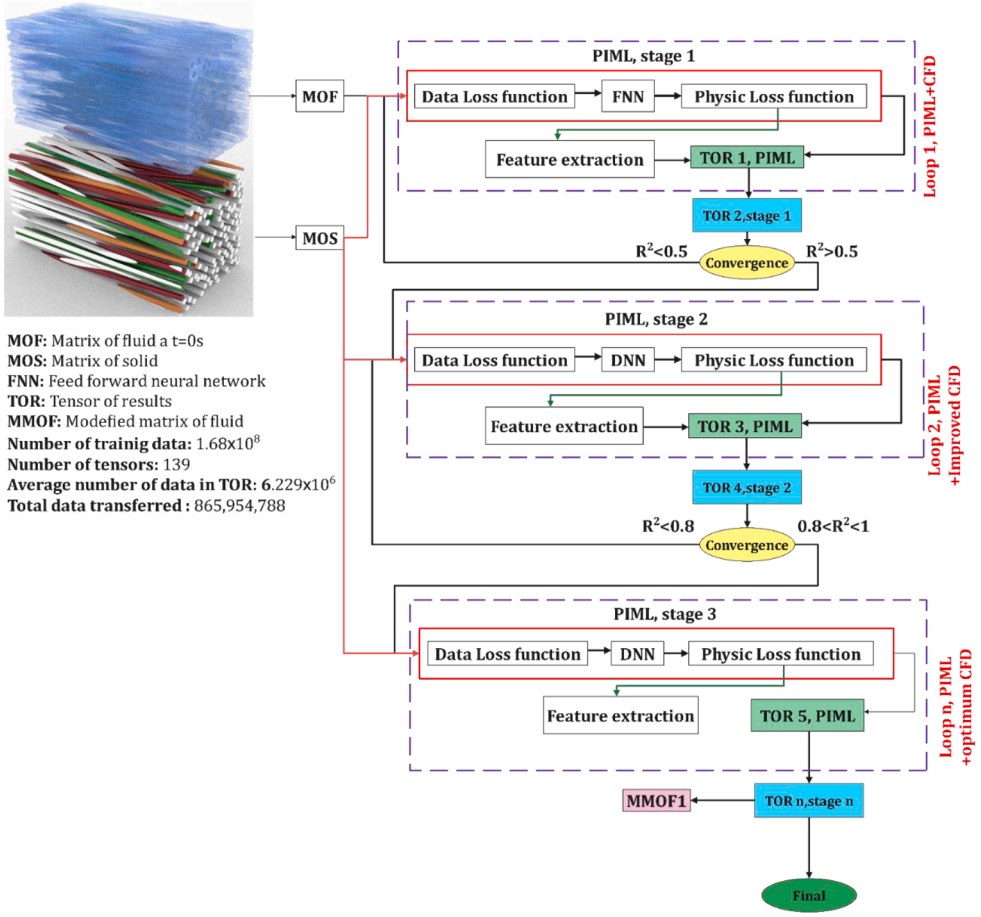


Fig. 5. Data flow for modified LES.

Table 3
Details of FNN for each *Re*.

Method	Calculation Layer	Number of input parameter	Number of trainings	Number of validations	Training ratio	Time step	Hidden Layer
FNN	1	16,988,617	15,336,078	1652,539	90:10	10^{-5}s	87
	2	28,474,366	22,551,479	5922,887	79:21		112
	3	33,352,028	26,369,988	6982,040	80:20		119

calculations, the results of a full-scale simulation were fed directly into the neural network. As depicted in Fig. 5, the outcomes of numerical simulation (e.g., velocity, vorticity, and pressure) directly fed the prediction layers as an input matrix fed each layer. Details of 3D extracted geometry and verification are presented in Appendix A1.

The fluid matrix provided the boundary and physical boundary and solid (MOF and MOS) to satisfy the output of automatic differentiation. As previously stated, due to the complexity of PSPM's geometry, the 3D geometry was converted to a series of 2D layers. Each layer featured a new geometrical boundary condition. PIML calculation layers fed by MOF and MOS regenerated the loss function. Further, to determine the physical loss resulting from the data loss function, we updated the loss functions for data, initial condition, and boundary condition at each time step. This study also took the Coefficient of Determination (R^2) as a criterion for convergence. Because of the uncertainties about the variation of parameters over time, training methods necessitate a variable number of layers and neurons to achieve the best R^2. According to Fig. 5, three calculation layers are assumed to incrementally improve the final value of R^2. Each calculation layer uses the FNN method to transfer data from data loss to physical losses. Although the weight of neurons will change in each layer of the second and third layers, the FNN method will be used to train with similar layers and neurons for all layers. The variable train-to-validation ratio method was used to maintain the quality of the results in the present study.

In this method, the output determines the proportion of training data to validation data, as detailed in Table 3.

It is noticeable that each training iteration updates the weight parameters based on a partial derivative of the loss function against the current weight. The calculated derivatives at the front of FNN cells may be vanishingly small when the gradient of the activation function is less than unity, effectively preventing the weight from changing. Therefore, training the entire neural network is extremely slow or even completely halted, particularly for the front FNN cells. When using a large number of FNN cells to simulate a longer time sequence, gradient vanishing occurs, which affects the performance of both deep neural networks and FNNs. This is similar to the problem of multiple hidden layers in deep neural networks.

A homemade Python- and Julia-based code is developed by our group to machine learning. In the numerical simulation, a SIMPLE algorithm (https://github.com/gaabnuneses/SIMPLE-in-Julia, https://github.com/DelNov/T-Flows) is used for flow passing through the layers. All numerical simulations are developed based on the second-order (upwind) of accuracy. TensorFlow (https://github.com/tensorflow/tensorflow) is used for machine learning. The geometry of

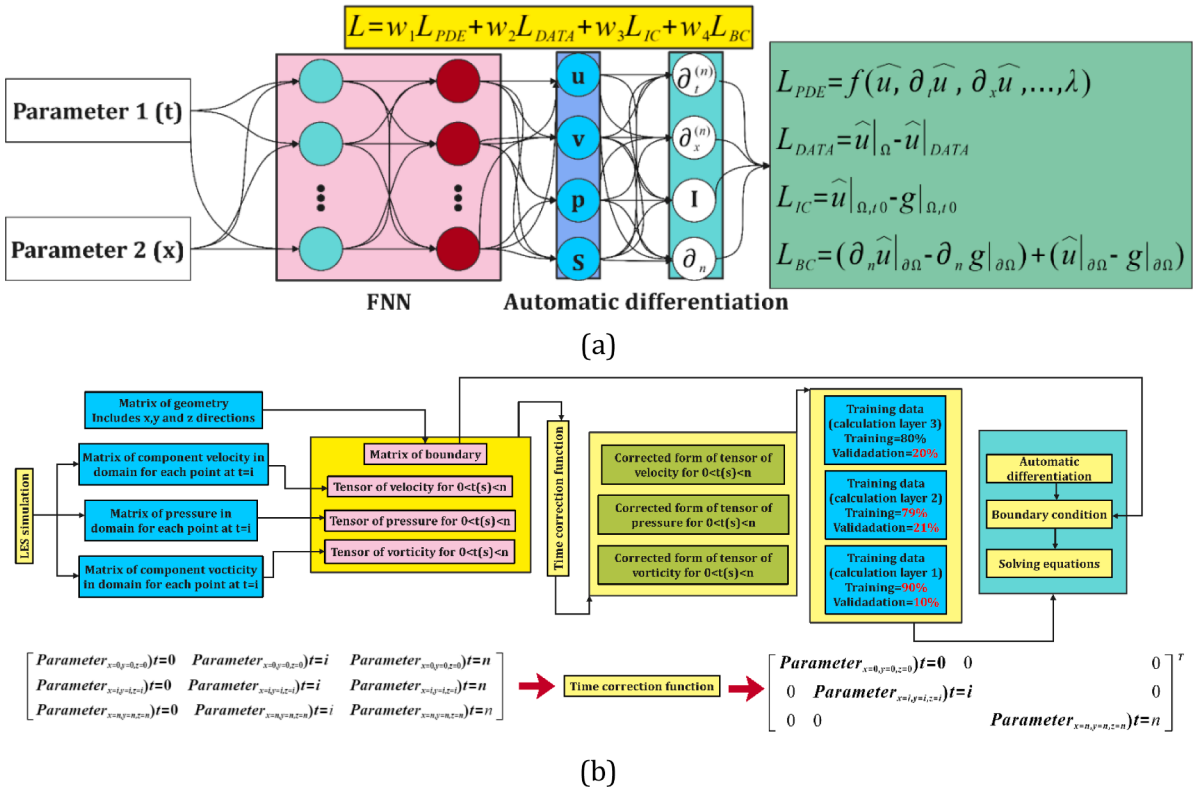


Fig. 6. Graphical illustration of a) Core of PIML algorithm including training, automatic differentiation, loss functions and boundary conditions and b) Training procedure.

Table 4
Variation of courant number based on pressure drop for different boundary conditions.

Inlet velocity (m/s)	Courant number	Pressure drop (average. Pa)
0.1	0.9	72
	0.84	68
	0.73	65
	0.62	65
0.5	1.3	480
	0.86	447
	0.77	443
	0.68	443

mask and its features are extracted using image processing technique, and OpenCV-Python (https://github.com/spmallick/learnopencv) is used for that purpose.

3. Governing equations of fluid flow

The continuity equation for PSPM is expressed as [43]

$$\frac{\partial\rho}{\partial t}+\frac{\partial(\rho u_x)}{\partial x}+\frac{\partial(\rho u_y)}{\partial y}+\frac{\partial(\rho u_z)}{\partial z}=0. \tag{5}$$

The PSPM time-dependent momentum equations are defined by the following [44]

$$\begin{aligned}&\rho\left(\frac{\partial(u_x)}{\partial t}+u_x\frac{\partial(u_x)}{\partial x}+u_y\frac{\partial(u_x)}{\partial y}+u_z\frac{\partial(u_x)}{\partial z}\right)=-\frac{\partial p}{\partial x}\\&-\frac{\partial}{\partial x}\left(-2\mu\frac{\partial(u_x)}{\partial x}+\frac{2}{3}\mu\left(\frac{\partial(u_x)}{\partial x}+\frac{\partial(u_y)}{\partial y}+\frac{\partial(u_z)}{\partial z}\right)\right)\\&-\frac{\partial}{\partial y}\left(-\mu\left(\frac{\partial(u_x)}{\partial y}+\frac{\partial(u_y)}{\partial x}\right)\right)=-\frac{\partial}{\partial z}\left(-\mu\left(\frac{\partial(u_x)}{\partial z}+\frac{\partial(u_z)}{\partial x}\right)\right),\end{aligned} \tag{6}$$

$$\begin{aligned}&\rho\left(\frac{\partial(u_y)}{\partial t}+u_x\frac{\partial(u_y)}{\partial x}+u_y\frac{\partial(u_y)}{\partial y}+u_z\frac{\partial(u_y)}{\partial z}\right)=-\frac{\partial p}{\partial y}\\&-\frac{\partial}{\partial x}\left(-\mu\left(\frac{\partial(u_x)}{\partial y}+\frac{\partial(u_y)}{\partial x}\right)\right)\\&-\frac{\partial}{\partial y}\left(-2\mu\frac{\partial(u_y)}{\partial y}+\frac{2}{3}\mu\left(\frac{\partial(u_x)}{\partial x}+\frac{\partial(u_y)}{\partial y}+\frac{\partial(u_z)}{\partial z}\right)\right)\\&-\frac{\partial}{\partial z}\left(-\mu\left(\frac{\partial(u_y)}{\partial z}+\frac{\partial(u_z)}{\partial y}\right)\right),\end{aligned} \tag{7}$$

$$\begin{aligned}&\rho\left(\frac{\partial(u_z)}{\partial t}+u_x\frac{\partial(u_z)}{\partial x}+u_y\frac{\partial(u_z)}{\partial y}+u_z\frac{\partial(u_z)}{\partial z}\right)=-\frac{\partial p}{\partial z}\\&-\frac{\partial}{\partial x}\left(-\mu\left(\frac{\partial(u_x)}{\partial z}+\frac{\partial(u_z)}{\partial y}\right)\right)\\&-\frac{\partial}{\partial y}\left(-2\mu\frac{\partial(u_z)}{\partial z}+\frac{2}{3}\mu\left(\frac{\partial(u_x)}{\partial x}+\frac{\partial(u_y)}{\partial y}+\frac{\partial(u_z)}{\partial z}\right)\right).\end{aligned} \tag{8}$$

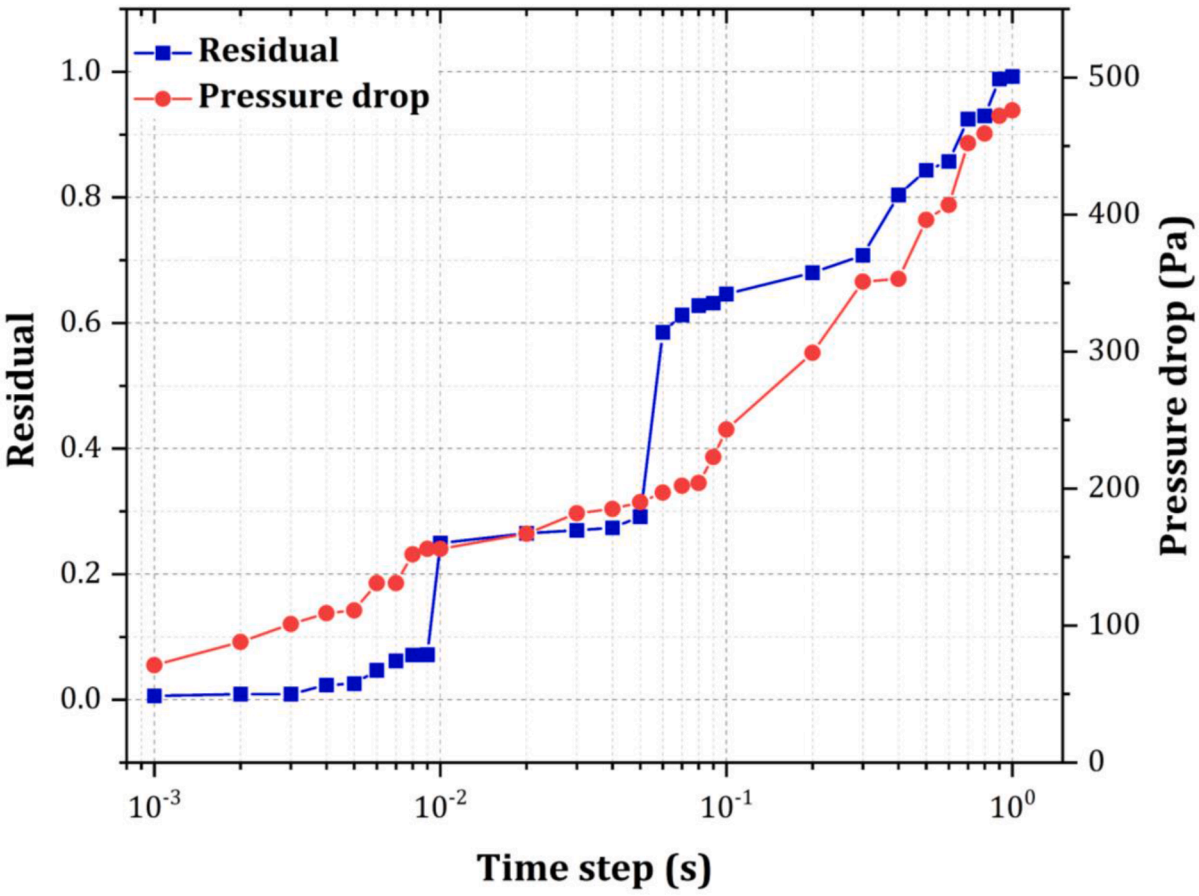
where ρ is density, p is pressure, and μ is viscosity.

3.1. LES model

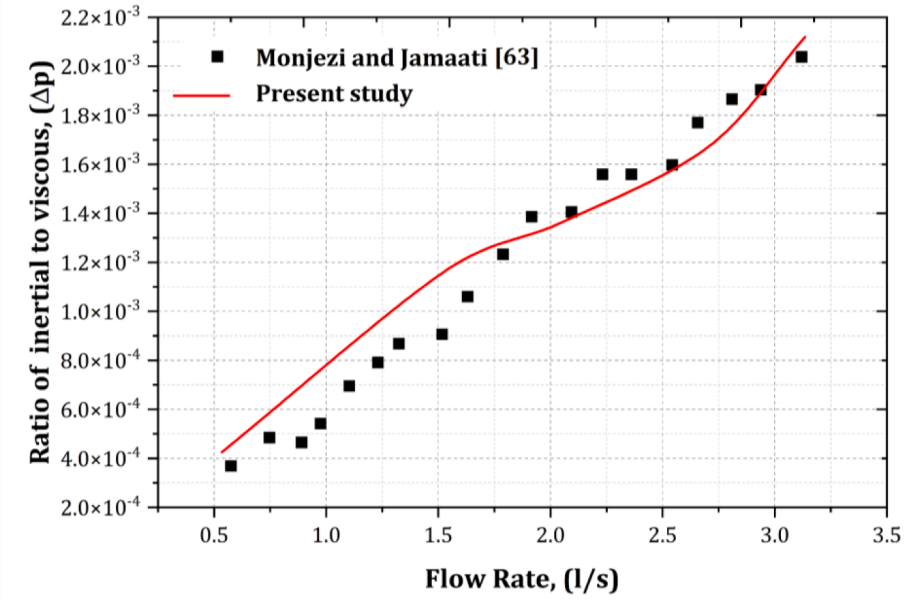
Transient eddies through the PSPM are studied using the WALE LES model. Multiple velocity components are used in large-eddy simulations. The resolved field is used to represent large eddies, while the sub-grid section is used to represent small eddies. Obtaining Navier-Stokes equations in space is achieved by convolution with a filter kernel G.

$$\overline{\varphi}(x)=\int_{Ds}\varphi(x')G(x;x')dx',\ \overline{\varphi}=\varphi+\varphi'. \tag{9}$$

In Eq. (9), the term of Ds refers to the fluid domain, while φ and φ' are resolved scale domain and sub grid scale part. The turbulent eddy viscosity (ν_t), is defined as [45]:

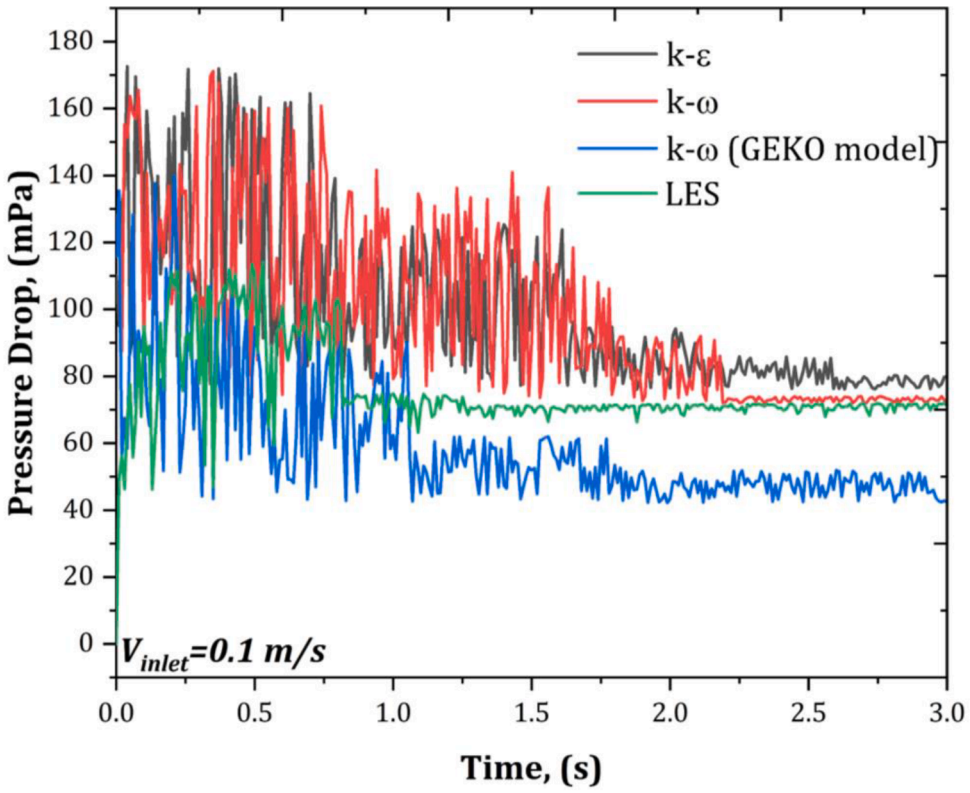


(a)

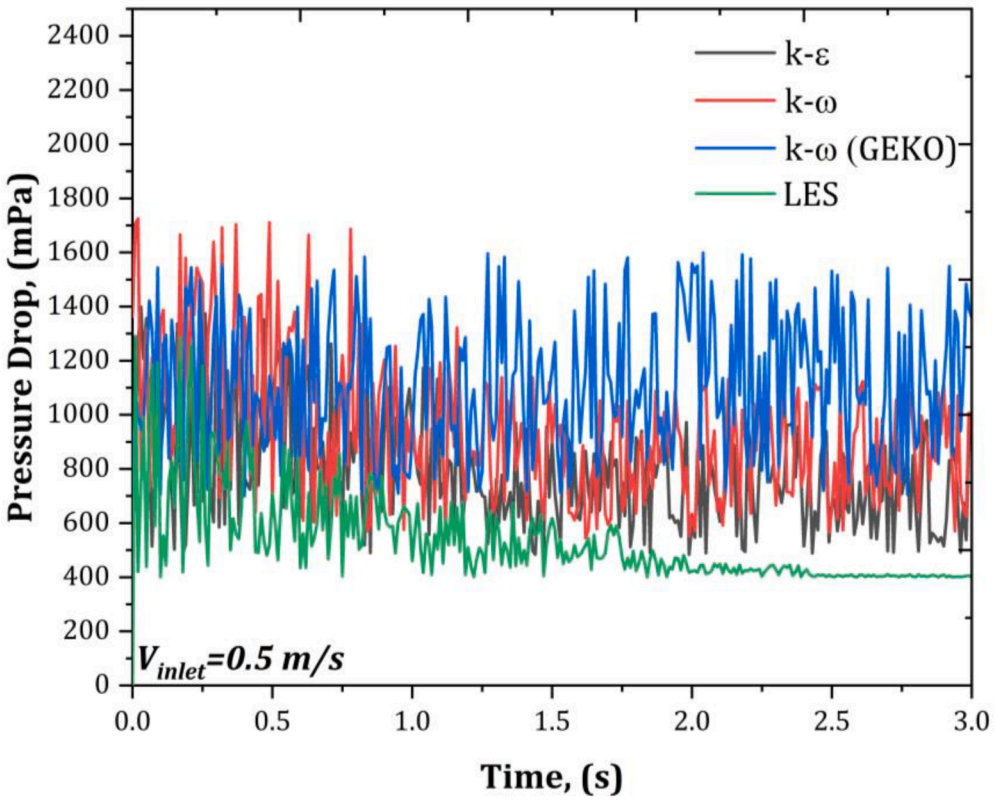


(b)

Fig. 7. a) Comparison between the residual of numerical simulation and pressure drop in the range of time steps, b) Comparison of current simulations and experimental data of Monjezi and Jamaati.



(a)



(b)

Fig. 8. Stability analysis for numerical simulation based on different models of turbulence method a)0.1 m/s and b) 0.5 m/s.

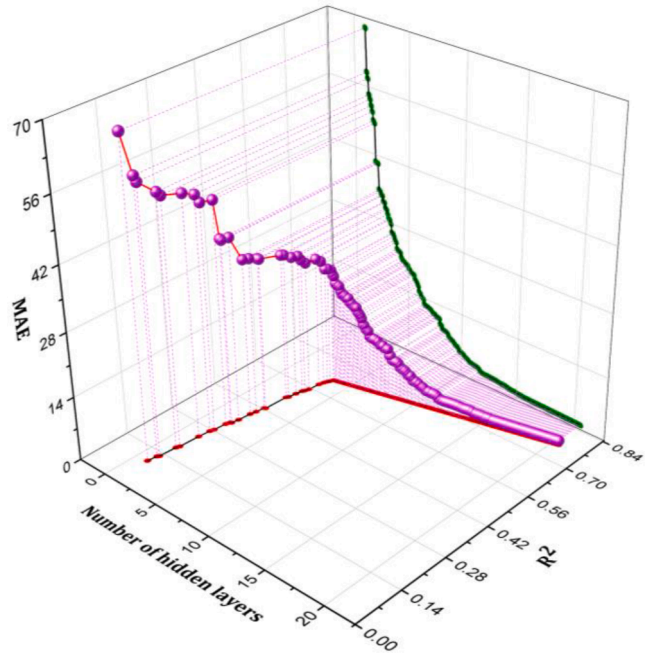


Fig. 9. The effect of hidden Layers on MAE and R^2.

$$\nu_t = C_X^2 \overline{\omega}. \tag{10}$$

In this case, Cx=0.5 represents the constant that varies with the LES model and $\overline{\omega}$ is the operator [46].

$$\overline{\omega} = \frac{\left(g_{i,j}^d g_{i,j}^d\right)^{\frac{3}{2}}}{\left(s_{i,j}s_{i,j}\right)^{\frac{5}{2}} + \left(g_{i,j}^d g_{i,j}^d\right)^{\frac{5}{4}}}. \tag{11}$$

$$g_{i,j}^d = s_{i,k}s_{k,j} + \Omega_{i,k}s_{k,j} - \frac{1}{3}\delta_{k,j}\left(s^2 - \Omega^2\right). \tag{12}$$

3.2. Reinforcement neural network models

The following examples are provided, (x_1,y_1), (x_2,y_2),..., (x_n,y_n) where $x_i \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$. Through the use of hidden layers underneath a hidden neuron, MLP learns a new function $f(x) = W_{2g}(W_1^T x + b_1) + b_2$, where $W_1 \in \mathbb{R}^m$ and $W_2, b_1, b_2 \in \mathbb{R}$. The layer($W_1$,$W_2$) represents the input layer's weights, and this layer represents its hidden layer's weights. The bias added to the hidden and output layers is represented by b1. In this case, $g(.)$: $R \rightarrow R$ is the hyperbolic tan activation function [47].

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \tag{13}$$

For binary classification, $f(x)$ a logistic process takes place $g(z) = \frac{1}{(1+e^{-z})}$ For example; a threshold of 0.5 would classify samples greater than 0.5 as positive, while the remaining samples would be classified as negative. The length of f(x) increases as the number of classes increases. As a result, a loop rather than a logistic process is used. To obtain output values between 0 and 1. It does not pass through the logistic function [48] but rather the softmax function [49], which is referred to by

$$softmax(z)_i = \frac{\exp(z_l)}{\sum_{l=1}^{k}\exp(z_l)}. \tag{14}$$

Softmax input z_l represents the l^{th} element of class l, and softmax input k represents the number of classes. A vector of probabilities that describes the membership of a sample to each class is generated as a result of this procedure. As a result of the fact that the output is determined by the probabilities associated with each class and that regression results in f(x), the output activation function is the same as the identity function. MLP makes use of a number of different loss functions, all of which are determined by the nature of the problem [50]. Cross-entropy [51] is the classification loss function, written as $(Loss(\hat{y}, y, W))$in the binary case.

Table 5
The effect of the calculation approach on runtime.

Layer	Calculation time (min) Star CCM+ simulations (CPU based) V_{inlet} =0.1 m/s	V_{inlet} =0.5 m/s	PIML (GPU based) V_{inlet} =0.1 m/s	V_{inlet} =0.5 m/s
1	–	–	15	17
2			24	36
3			47	85
4			41	74
5			58	67
6			94	77
Total	6300	7400	279	356

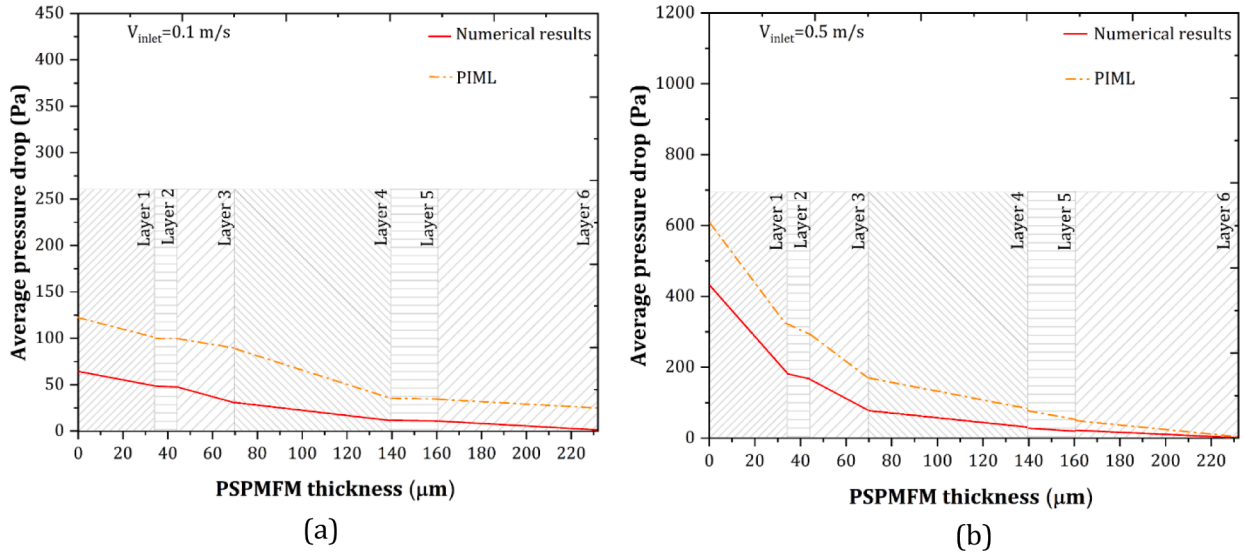


Fig. 10. Comparison of PIML and numerical simulation approaches the pressure distribution along the PSPM a) V_{inlet} = 0.1 m/s and b) V_{inlet} = 0.5 m/s.

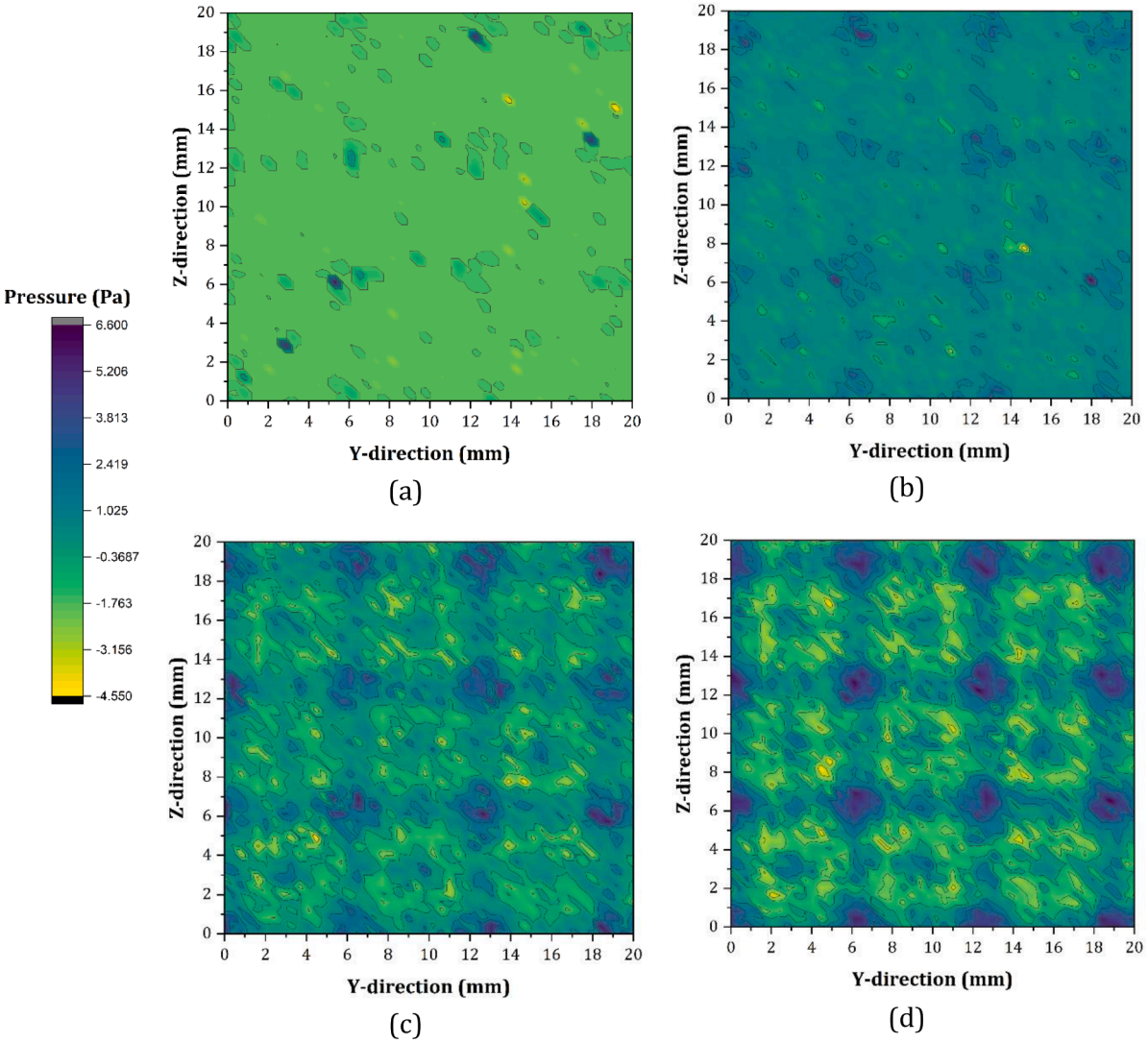


Fig. 11. Distribution of pressure through layer 1 for V_{inlet} =0.1 m/s when a) $t = 0.01$ s, b) $t = 0.05$ s, c) $t = 0.1$ s and d) $t = 0.5$ s.

$$Loss(\hat{y}, y, W) = \frac{1}{2}\hat{y} - y_2^2 + \frac{\alpha}{22}W_2^2. \tag{15}$$

MLPs minimize the loss function by updating the weights repeatedly, starting with random weights [52]. After the loss has been computed, a backward pass will propagate it from the output layer to the layers that came before it, updating each weight parameter in order to lower the loss. In gradient descent, the gradient $\nabla LossW$ of the loss with respect to the weights is computed and deducted from W.

3.2.1. *Physics-informed machine learning*

The present study uses a parametrised partial differential equation (PDE) form of equation as an example

$$f(x, t, \hat{u},\ \partial_x\hat{u},\ \partial_t\hat{u},\ \ldots,\ \lambda) = 0,\ x \in \Omega,\ t \in [0, T] \tag{16}$$

$$\hat{u}(x, t0) = g0(x),\ x \in \Omega \tag{17}$$

$$\hat{u}(x, t) = g_\Gamma(x),\ x \in \partial\Omega,\ t \in [0, T] \tag{18}$$

where $x \in \mathbb{R}$ is the spatial coordinate, and f is residual. A method of minimisation of the loss function L will be presented in this study by updating the PIMLs iteratively.

$$\theta = \frac{Weights}{Biases})$$

$$L = w_1 L_{PDE} + w_2 L_{DATA} + w_3 L_{IC} + w_4 L_{BC}, \tag{19}$$

where ω_{1-4} are the utilized weighting coefficients. This equation penalizes the residual of the governing equation using L_{PDE}. In addition, L_{DATA} is utilized to demonstrate data loss. Initial condition is represented by L_{IC}, while boundary conditions are represented by L_{BC}. Fig. 6a displays a diagrammatic representation of the PIML's fundamental structure. In this diagram, the unknown PDE parameters $\hat{u} = [u, v, p, \varnothing]$, $x = [x, y]$, and λ are depicted along with the weights $w_i = 1, 2, \ldots, 4$ within the PIML procedure, PIML training is regarded as one of the most important steps. Fig. 6b provides a detailed outline of the training procedure.

Here, the PIML framework involves the following steps [53].

1 Create a dataset for training with Ns random combinations of transmissions and storability and Nb batches.
2 Make sure the NNM has both an input layer and an output layer. The input layer should be used for storing storability and passing transmissions, while the output layer should be used to estimate fluid extraction rates for each extraction well.
3 Calculate the loss function in order to quantify flow parameters like pressure and velocity.
4 Make necessary adjustments to the NNM coefficients in accordance with the gradient of the loss function.

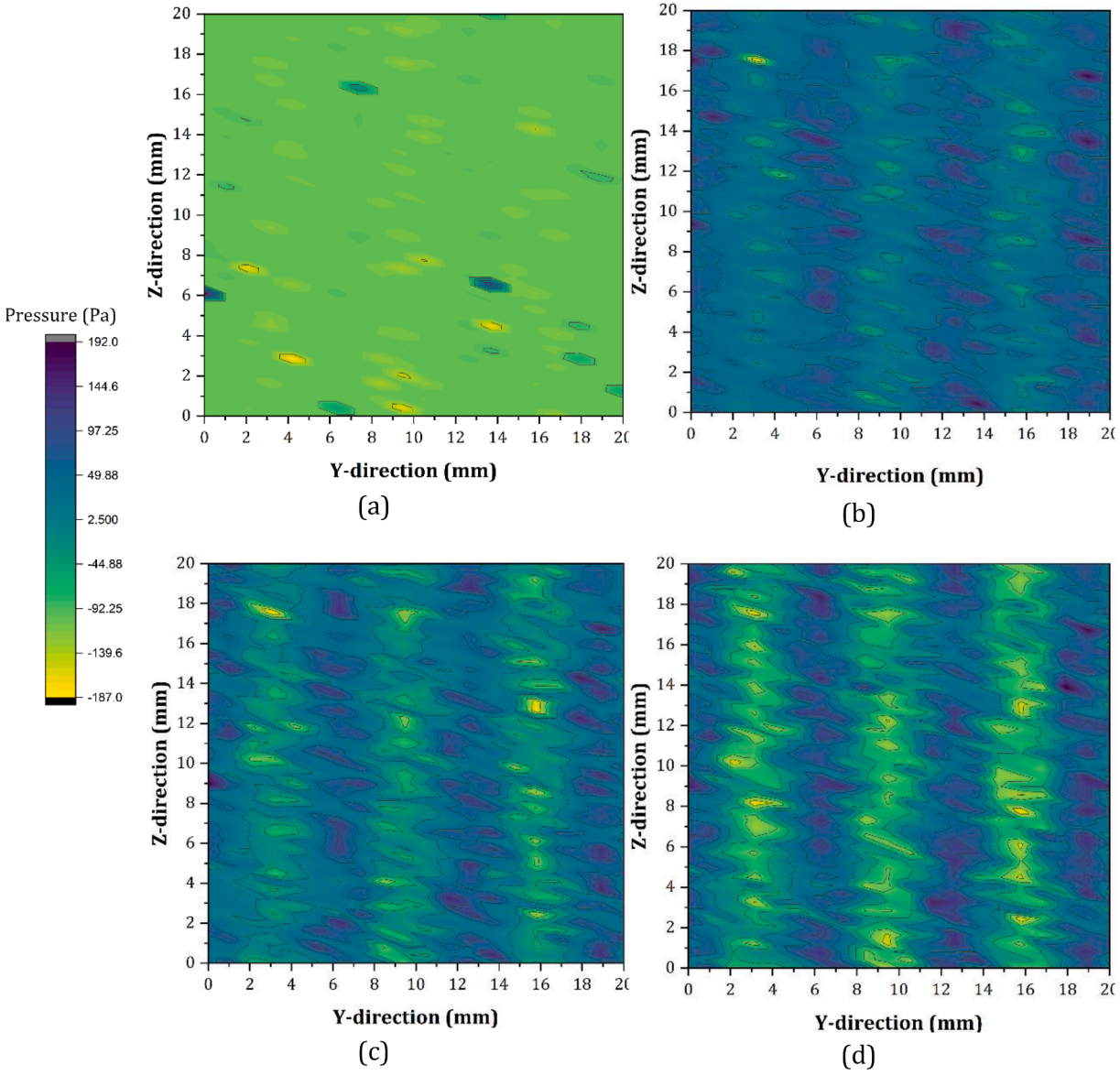


Fig. 12. Distribution of pressure through layer 2 for V_{inlet} =0.1 m/s when a) $t = 0.01$ s, b) $t = 0.05$ s, c) $t = 0.1$ s and d) $t = 0.5$ s.

Step 2 requires the use of a fully linked NNM with all hidden layers based on the layer of calculation, in which there should be a ratio of neurons to inputs and outputs. Sigmoid functions are used to activate all hidden layers [54].

$$\sigma(\psi) = 1/(1+\exp(-\psi)). \tag{20}$$

where ψ is the input to the neuron and σ(ψ) is the output. Consequently, the extraction rates are negative due to the negated soft plus function is defined as [55]

$$Q_{inj} = -ln(1+\exp(\psi)). \tag{21}$$

where ψ is an output of the NNM and $Q_{inj}$is the associated extraction rate. As a preliminary investigation, we explored the possibility of using more advanced deep neural networks. These factors did not influence the PIML training. These networks degraded training performance in some cases, most likely because larger networks produce more complex response surfaces, which increases the likelihood of the training being trapped in local minima.

The loss function $\mathscr{L}$ is defined as

$$\mathscr{L}(\Theta) = \sum_{i}^{N_c}\sum_{j}^{N_s}\left(\Delta\Gamma_i\left(\Theta^{NN}\left(\Theta, T_j, S_j\right), T_j, S_j\right) - \Delta\Gamma_i^t\right)^2. \tag{22}$$

where Nc is the number of layers, Γ_i is the simulated flow pattern (results of CFD), $\Delta\Gamma_i^t$ is the target flow pattern at the i^{th} location, Q^{NN} is the trained data, Θ contains the NNM coefficients. The root-mean-square error (RMSE) of the training dataset can be used as a proxy for the accuracy of the loss function in obtaining the desired flow pattern [56].

$$RMSE(\Theta) = \sqrt{\frac{1}{N_s * N_s}\sum_{i}^{N_c}\sum_{j}^{N_s}\left(\Delta h_i\left(\Theta^{NN}\left(\Theta, T_j, S_j\right), T_j, S_j\right) - \Delta h_i^t\right)^2}. \tag{23}$$

A weighted sum of the outputs of the input layer is applied to the output layer, which consists of neurons $m + 1$.

$$f_j = g\left(w1_{0,j} + \sum_{i=1}^{n} w1_{i,j}\, x_i\right). \tag{24}$$

A neuron's activation function determines whether it is active or not [57]. It determines how significant the neuron's input is during the prediction process through simpler mathematical calculations.

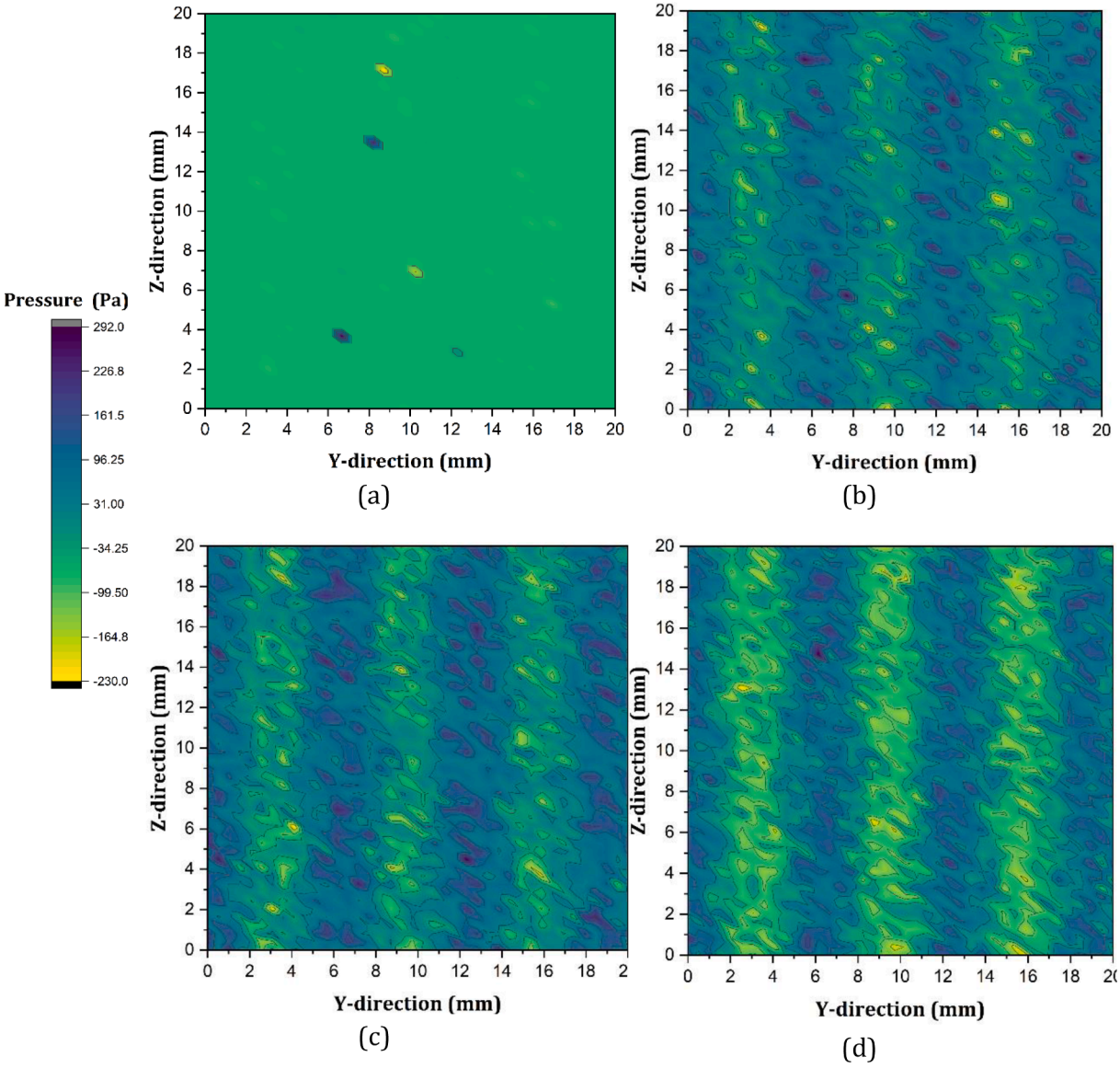


Fig. 13. Distribution of pressure through layer 3 for V_{inlet} =0.1 m/s when a) $t = 0.01$ s, b) $t = 0.05$ s, c) $t = 0.1$ s and d) $t = 0.5$ s.

Activation of the neurons in this layer generates the model's output, just as activation of neurons in the hidden layer does.

$$y_z = g\left(w2_{0,z} + \sum_{j=1}^{m} w2_{j,z}\, f_j\right). \tag{25}$$

$$y_z(M.L) = g(M.L)\left(w2_{0,z}(M.L) + \sum_{j(M.L)=1}^{m(M.L)} w2_{j,z}\ (M.L) f_j(M.L)\right). \tag{26}$$

where $y_z(ML)$ is the output of z^{th} neuron and $w2_{0,z}$ and $w2_{j,z}$are the weights of the bias. As a result of these calculations, the algorithm's current output is reduced to its expected output. This step determines an algorithm's model of the data. It is known as the mean square error (MSE) because it compares actual values with predicted values by the mean squared difference. MSE criterion is used as the loss function, and error backpropagation is used as the optimization algorithm [58].

$$\mathrm{MSE} = \frac{1}{\mathrm{n}} \sum_{i=1}^{\mathrm{n}} (\mathrm{O_i} - \mathrm{T_i})^2. \tag{27}$$

n is the number of samples, O_i and T_i are the model output.

3.2.2. Adam optimization method

In order to train neural networks in general and PIMLs in particular, an algorithm known as Adam optimiser [59] is utilized. Adam optimiser is a first-order stochastic gradient descent algorithm. This approach has proven to be effective in the completion of a variety of supervised learning projects thanks to the scalability it provides. PIMLs make it difficult to adjust the learning rate and weights associated with various loss terms, which makes multi-objective optimization a challenging problem to solve. In addition to the second-order Broyden–-Fletcher–Goldfarb–Shanno algorithm (BFGS) method [60], the Limited-Memory Broyden–Fletcher–Goldfarb–Shanno Algorithm (L-BFGS) [61] is also frequently used to train PIML models. This method becomes unusable as the number of parameters grows because it is both extremely slow and not scalable. In contrast to first-order methods, Hessian matrices do not have the capacity to automatically determine the learning rate. In Adam optimization, gradient descent is one of the techniques used, and one of the methods employed in moment estimation is adaptive moment estimation. When dealing with problems that have a large number of parameters, a method known as gradient descent combined with momentum and RMSP is an extremely effective solution. The Adam optimization technique places more of an emphasis on

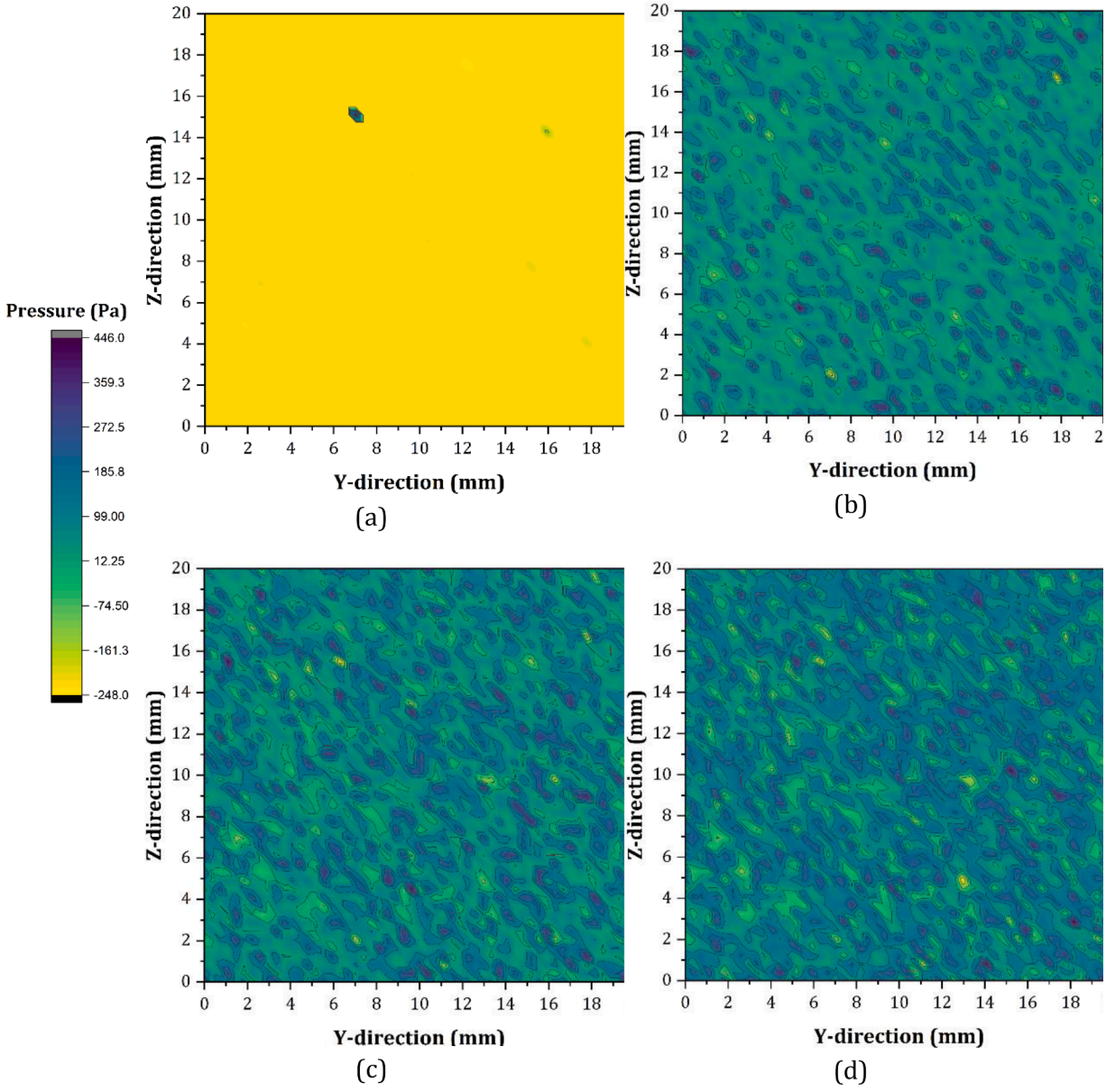


Fig. 14. Distribution of pressure through layer 4 for V_{inlet} =0.1 m/s when a) t = 0.01 s, b) t = 0.05 s, c) t = 0.1 and d) t = 0.5 s.

calculating the average second moment of gradients as opposed to modifying learning rates in accordance with the average first moment of gradients. With the help of this algorithm, gradients and square gradients can be determined. In addition, the decay rates of these moving averages are controlled by the parameters b_1 and b_2, respectively. Methods of momentum and RMSP gradient descent are utilized in the Adam optimization process.

3.2.3. *Adaptive weights*

After the training of the PIMLs, a gradient pathology that is made up of the total loss function is associated with each loss term. The gradient vectors for each loss term have greater gradients, so they dominate the other gradient vectors throughout the training process. This is true when comparing gradient vectors to network parameters. PIMLs should be avoided because of the high level of dependence that they have on the boundary conditions. [62].

4. Grid generation and validation

4.1. *Validation of numerical simulations*

Throughout the transient study, the Courant number is essential for determining the optimal time step. As shown in Table 4, the pressure drop varies depending on the inlet velocity and the current number. The results indicated that inlet velocities required different courant numbers on PSPM materials.

The difference in pressure drop and residuals caused by varying time steps is depicted in Fig. 7a. Fig. 7b presents a comparison between the current numerical simulations and the experimental data of Monjezi and Jamaati [63] on the pressure drop across a surgical mask. Based on the results, a time step of 10^{-3} s is appropriate for this study. The transient flow through the PSPM imposes severe limitations on numerical simulations. Due to the micro dimensions of pores, it might be necessary to modify the standard turbulence model. Fig. 8a and b illustrate how the turbulence model influences the pressure drop variation over 3 s. According to the results, LES presented pressure drop variations more

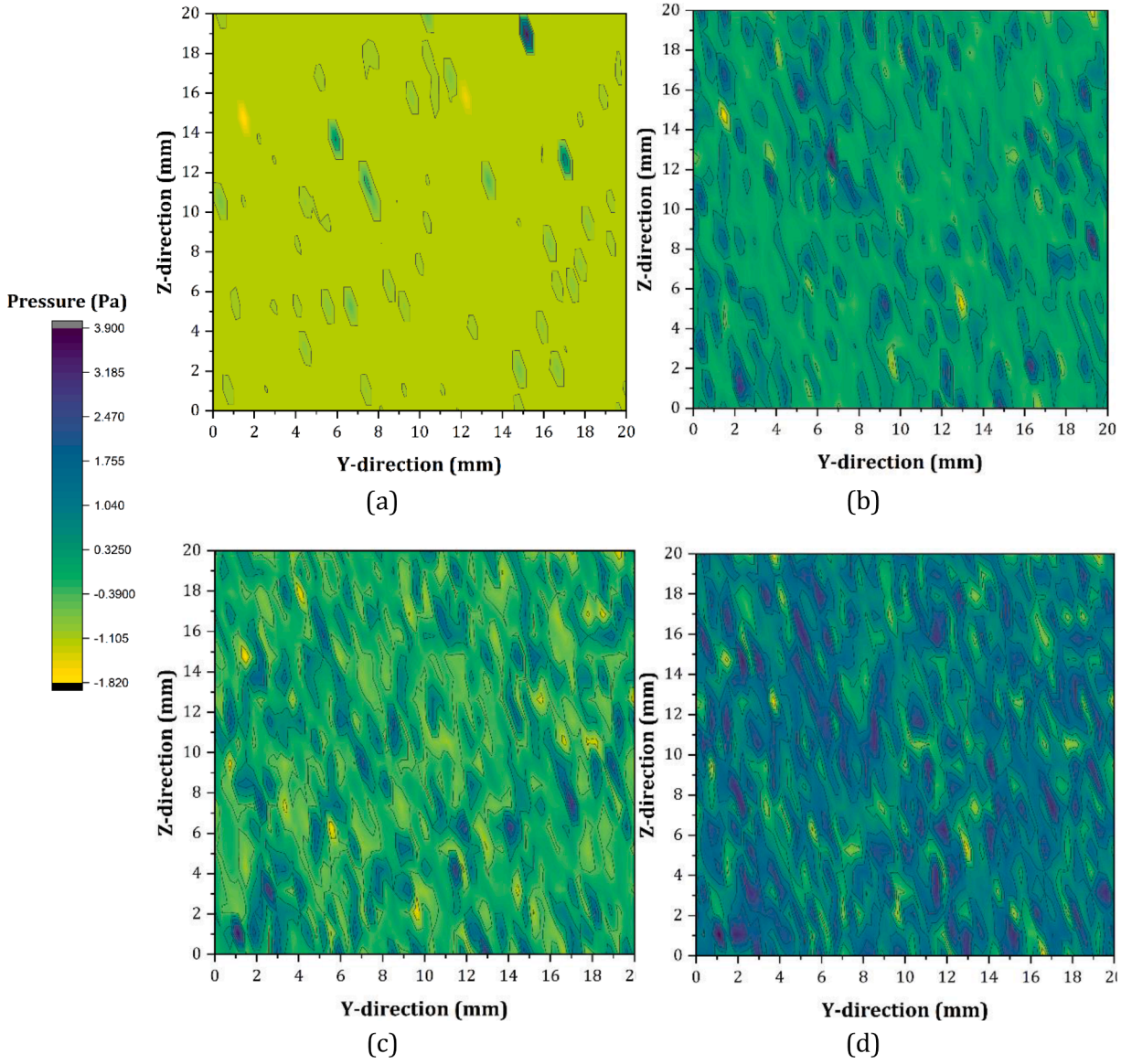


Fig. 15. Distribution of pressure through layer 5 for V_{inlet} =0.1 m/s when a) t = 0.01 s, b) t = 0.05 s, c) t = 0.1 and d) t = 0.5 s.

accurately than other methods. Turbulence modeling can directly impact runtime and precision. In order to find the proper turbulence model and understand the flow between layers, the present study simulated four different turbulence models in transient conditions to find the transition to steady-state time. Due to a serious lack of information in the verification data, pressure drop was assumed as a criterion for validation. It is important to note that unseen flow behaviours passing through layers and simulating the true flow between layers are affected by various factors such as mesh resolution.

Appendix A2 presented the CFL study in this project.

4.2. Validation of machine learning

Validation is important for machine learning and numerical simulation. An analysis of the neural network analysis and the hidden layer independence of machine learning is covered in this section. Here, we examine for the first time the number of hidden layers on MAE and R^2 using various machine learning methods. The optimal number of hidden layers can vary depending on the number of inputs and outputs. Fig. 9 depicts the optimal number of hidden Layers. According to the minimum R^2 and MAE values, no more than three hidden Layers are permitted.

5. Results and discussion

5.1. Comparison between numerical simulations and PIML results for the average pressure drop

The regular approach to porous media is based on the Darcy equation, simplifying the velocity-pressure relationship. A complex porous medium, such as a surgical mask, consists of multiple layers with distinct properties. Although certain parameters, such as fiber distribution, can vary between layers, the average properties, such as porosity and permeability, are comparable within a single layer. In this study, the properties of each layer are defined using microscopic images, so the geometrical characteristics of each layer are distinct. Fig. 10a illustrates the variation in pressure drop when flow passes through all six layers and V_{inlet} = 0.1 m/s. The results indicated that both models could predict the pressure drop pattern caused by PSPM. According to this figure, the pressure drop in the gap zones (Layers 2 and 5) follows a smooth descending pattern. Since the gap distance is in the micron range, the pressure drop pattern in these zones does not vary significantly.

Fig. 10b shows the pressure drop that occurs when V_{inlet} =0.5 m/s. According to the results, the PSPM experiences a significant decrease in

Table A1
Pixel study in image processing.

Layer	Pixel per inch square (PPI)	Number of points	Number of detected fibers	Minimum diameter [μm]	Maximum diameter [μm]
1	11	10,224	25	24	36
	29	11,304	32	22	38
	66	22,304	37	15	32
	78	**47,263**	**42**	**17**	**28**
	88	107,735	45	17	27
2	154	2367	15	12	19
	301	4784	11	7	22
	313	4934	8	2	3
	331	**4981**	–	–	–
	338	5152	–	–	–
3	753	52,186	54	19	28
	1926	35,760	86	14	22
	3072	434,555	91	12	19
	5207	**590,272**	**95**	**12**	**17**
	5578	511,488	112	12	17
4	3287	39,310	114	16	23
	3899	340,949	136	14	22
	7610	**355,999**	**174**	**13**	**19**
	8522	429,024	174	12	19
	8734	867,038	176	12	18
5	85	102	21	8	9
	76	164	19	9	9
	38	**1190**	–	–	–
	49	2230	–	–	–
	71	3203	–	–	–
6	4057	85,461	34	25	47
	2049	148,091	39	25	39
	4297	**289,587**	**57**	**24**	**38**
	5683	364,635	61	24	38
	5867	622,736	61	23	38

Table A2
Variation of pressure drop in different Layers.

Layer	Velocity (m/s)	Time step (s)	Pressure drop (Pa) Star CCM+ simulations Unsteady	In-house code unsteady
1	0.1	0.1	9.70	15.42
		0.001	8.86	14.23
		0.0001	4.54	7.56
	0.5	0.1	73.66	99.46
		0.001	65.02	61.96
		0.0001	39.51	45.69
2	0.1	0.1	2.42	8.185
		0.001	1.57	7.726
		0.0001	0.96	1.565
	0.5	0.1	9.61	15.64
		0.001	5.01	11.63
		0.0001	4.62	3.316
3	0.1	0.1	39.03	45.79
		0.001	21.02	27.17
		0.0001	19.77	19.25
	0.5	0.1	173.39	179.2
		0.001	141.91	147.5
		0.0001	132.06	124.09
4	0.1	0.1	68.60	74.43
		0.001	52.72	58.85
		0.0001	33.98	35.61
	0.5	0.1	520.83	526.81
		0.001	298.67	304.95
		0.0001	215.66	221.61
5	0.1	0.1	4.68	10.19
		0.001	3.68	6.60
		0.0001	1.35	2.23
	0.5	0.1	11.83	17.35
		0.001	9.05	15.23
		0.0001	7.31	9.44
6	0.1	0.1	6.76	12.35
		0.001	6.85	9.19
		0.0001	3.09	5.46
	0.5	0.1	61.42	67.39
		0.001	39.63	45.36
		0.0001	29.57	35.38
Total	0.1	0.1	93.39	99.14
		0.001	75.30	81.31
		0.0001	68.72	74.86
	0.5	0.1	931.06	937.97
		0.001	549.29	555.34
		0.0001	422.35	428.50

pressure when the flow passes through it (up to −34%). Additionally, the results show that PIML can accurately predict pressure drop behavior compared to the standard numerical method. The first step in PIML for predicting flow passes through PSPM and involves rewriting the continuity and turbulence model equation. A comparison of these two figures demonstrates that developing PIML based on MOF and MOS can satisfy the required PIML boundary condition.

Fig. 10a and b demonstrate unequivocally that a tighter coupling between numerical methods and PIML requires fluid pattern prediction with the PSPM. The behavior of predicted patterns generated by different methods can help explain the relative importance of training data and prediction methods. Run time is one of the most intriguing aspects of the ML method. Complex geometry can present a formidable computational challenge and to tackle that in the present study, a GPU-based PIML code was developed. Table 5 compares the calculation time required for numerical and PIML approaches.

According to Table 5, the PIML method can reduce calculation time by up to 22.5 and 20.7 times for the inlet flow velocities of 0.1 and 0.5 m/s, respectively. Numerous factors, such as the inherent difference between CPU and GPU, algorithms, and software, can influence this factor. Regarding computational cost, the architect of code, data transfer method, solver, and processor are the most important roles at run time.

5.2. *Predicting pressure fluctuations in distinct layers of the surgical mask (PIML results)*

Fig. 11 illustrates how PIML predicts the pressure pattern in layer 1 at $V = 0.1$ m/s for various times (0.01, 0.05, 0.1, and 0.5 s). According to the results, the pressure distribution in layer 1 follows a periodic pattern. In addition, Fig. 11a–d depict the impact of blankets on the flow pattern. One can observe a pressure drop of 1.78 Pa in layer 1. Even though layer 1 has a uniform porosity distribution, the results show that the pressure distribution in that layer is nearly uniform during time steps. The PIML method could detect pressure variations and account for fluctuations during forecasting. As already shown, each block was exposed to a complex and random mixture of fibres and fluid (boundaries) due to the presence of complex and random fibres in the PSPM. This lack of training data leads to numerous issues, such as difficulty matching blocks. Due to the numerous grids in PSPM, flow generated a substantial amount of data. Training the neural network with this volume of data requires substantial time and training data. Due to the random geometry of PSPM, the conventional approaches that train 80% of the total data to predict are unreliable and inconsistent.

As depicted in Fig. 11c, the maximum and minimum pressure values in layer 1 at $t = 0.1$ s are nearly three times greater at $t = 0.01$ s compared to $t = 0.1$ s. In complex geometries, it is essential to have as much information as possible regarding the fluid zone, such as pressure or vorticity. In complex geometry prediction, the quality of the initial simulation is essential. Most complex geometries, including PSPM, necessitate a high level of grid generation and validation for numerical simulation. Training a neural network with precise simulations can significantly affect its predictions.

Layer 2 is an intermediate layer between layers 1 and 3. Due to the thinness of this layer, there is a significant pressure drop across it. The pressure pattern in layer 1 determines the pattern in layer 2, as shown in Fig 12a. A fluid flow blanket can alter or even change the pressure pattern in layer 2. Certain fluid properties, including turbulence

intensity and vorticity, necessitate more complex loss functions. In addition, regenerating automatic differentiation for particular parameters, such as vorticity, via the PSPM enlarges the calculation.

Fig. 12a depicts how PIML predicts the pressure pattern when fed with inadequate training data. The flow generates numerous micro vortices through the PSPM. As a result of the random distribution of fibres, these vortices form in various orientations and intensities. For these parameters to be tracked through layer 2, layers 1 and 2 should be fully connected. Due to their consideration of the uniform block, PIML is able to track the vorticity of PSPM. Therefore, the initial state of layer 2 can be determined by the fluid flow pattern in layer 1. Maximum and minimum pressure values in PIML indicate the results of this characteristic (vortex detection). The method's severe limitations make it nearly impossible to develop a loss function for a turbulence model [69]. The issue here concerns tracking of vortices and the energy variation of fluid control volumes.

Fig. 12b to d shows the pressure distribution in layer 2 when $t = 0.1$ s and $t = 0.5$ s. Results indicate that pattern of transient vorticities can be captured between layer 1 and layer 2. Fig. 13a and b depict pressure variations throughout layer 3. Due to variations in the training data set, the results demonstrated that the concentration of fibres could affect the accuracy of prediction results. Tracking transient vortices through a complex geometry necessitates an increase in the average resolution of the mesh. As a result, the training data set will expand. According to the results, the training dataset for layer 3 has increased by 14.4% compared to other layers. Additionally, the prediction time was increased by 9.84% compared to other layers. The pressure patterns at $t = 0.1$ and $t = 0.5$ (Fig. 13b and c) demonstrate that the quality of training data directly impacts the output of PIML.

Fig. 14 presents a transient pressure through layer 4. According to the results, although unclassified data can detect patterns, their high standard deviation makes it impossible to predict the fluid flow behavior in this layer. The maximum variation of critical parameters such as vorticity, pressure drop, and dissipation rate occurs in the third layer. It will be easier to comprehend the effect of geometry on flow under steady-state conditions. Therefore, constant pressure, velocity, and vorticity will reach layer 4 regardless of transient vorticities. Due to the size and density of the fibres, layers 3 and 4 require a distinct method of solution. The geometry of layers 3 and 4 can be used to calculate the flow through each layer independently, superposition [69].

Nonetheless, a continuous method was used in this study due to the small size of the test section ($4mm^2$). When the initial condition came from layer 3, physics-based neural networks faced a new obstacle in predicting flow patterns. For each parameter, the loss function should be individually adjusted. When the boundary and initial conditions (passed flow through layer 3 and fiber density in layer 4) become more complex, it becomes harder to regenerate the loss function. In addition, solving the problem of automatic differentiation becomes more difficult.

Fig. 15 depicts the effect of high-density pore-scale materials on pressure drop while maintaining a uniform pressure distribution. Due to the high density of fibres and the average distance between layers, this layer's pattern of vorticity and turbulence intensity is extremely complex. Being a gap, layer 5 has simplified boundary conditions, whereas the initial condition for automatic differentiation is complex (passed flow through layers 3 and 4). It is simple to regenerate the loss function for the PIML method based on layer 5 due to the free space. Since numerical simulations accurately predict the flow through this layer, we anticipate that the outcomes of PIML to be comparable.

6. Conclusions

Flow simulation through complex geometries, such as pore-scale porous media (PSPM), is a true challenge in fluid mechanics. The grid generation for complex geometries is the primary barrier to developing numerical methods in these domains. The present study presented a novel solution for predicting flow through complex geometry. Physics-informed machine learning was developed based on transient and turbulence computational methods to predict transient flow through the microscale PSPM. To achieve this objective, a method of image processing was used to extract 3D geometry from a series of 2D microscopic images of a surgical mask. After validating the extracted geometry, commercial software (Star CCM+) was used to simulate a transient flow. For two inlet velocities (0.1 and 0.5 m/s), the simulated flow passed through six distinct layers of a surgical mask. Next, PSPM's 3D geometry was converted to a series of 2D layers in order to implement physics-informed machine learning (PIML). Automatic differentiation output utilised the boundary condition to solve the loss function. We used the Adam optimization method and a modified geometry matrix to meet PIML requirements (physical boundary condition). A feed-forward neural network with 13 layers of learnable weights and 18 neurons per layer has the highest coefficient of determination when used for training (R2). To determine the physical loss caused by the data loss function, we updated the data, initial condition, and boundary condition loss functions at each time step. The pattern of pressure, velocity, and vorticity were predicted for all six internal layers of the mask. The findings show that

1- Fluid flow experiences a large variety of pressure and vorticities intensity when passing through the PSPM.
2- The fiber concentration plays an important role in pressure drop in each layer. The results indicated that layer 3 has the strongest impact on the pressure drop due to its fiber concentration.
3- According to the results, the training dataset for layer 3 has increased by 14.4% compared to other layers. Additionally, the prediction time was increased by 9.84% compared to other layers.

The results indicated that the PIML method could reduce total calculation time (including training and prediction) by up to 22.5 and 20.7 times compared with standard numerical simulation, on average, for speeds of 0.1 and 0.5 m/s, respectively. Numerous factors, such as the inherent difference between CPU and GPU, algorithms, and software, can influence this factor.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

N. Karimi acknowledges the financial support by the Engineering and Physical Science Research Council, UK; through the grant number EP/V036777/1 Risk Evaluation Fast Intelligent Tool (RELIANT) for COVID 19. M. S. Shadloo acknowledge the access to French HPC resources provided by the French regional computing center of Normandy (CRIANN) (Grants No. 2017002).

Appendix A1. Verification of 3D extracted geometry

The PSPM geometry was determined using an approximation algorithm, so the effect of variable parameters on the final geometry needs to be validated and investigated. A strong correlation exists between the image quality and the image processing method. Table A1 details the image processing required to extract the geometry.

Appendix A2. CFL study

Table A2 shows the CFL analysis for different inlet velocities based on the LES model.

References

[1] Juandi, D. and M. Tamur. Review of problem-based learning trends in 2010-2020: a meta-analysis study of the effect of problem-based learning in enhancing mathematical problem-solving skills of Indonesian students. in Journal of Physics: Conference Series. 2021. IOP Publishing.

[2] Ortiz-Imedio R, Ortiz A, Ortiz I. Comprehensive analysis of the combustion of low carbon fuels (hydrogen, methane and coke oven gas) in a spark ignition engine through CFD modeling. Energy Convers Manage 2022;251:114918.

[3] Yang Y, Cai S, Yao J, Zhong J, Zhang K, Song W, Zhang L, Sun H, Lisitsa V. Pore-scale simulation of remaining oil distribution in 3D porous media affected by wettability and capillarity based on Volume of Fluid method. Int J Multiphase Flow 2021;143:103746.

[4] Lin W, Xiong S, Liu Y, He Y, Chu S, Liu S. Spontaneous imbibition in tight porous media with different wettability: pore-scale simulation. Phys Fluids 2021;33(3): 032013.

[5] Senecal, P., E. Pomraning, K. Richards, and S. Som. Grid-convergent spray models for internal combustion engine CFD simulations. in Internal Combustion Engine Division Fall Technical Conference. 2012. American Society of Mechanical Engineers.

[6] Kalvakala K, Pal P, Wu Y, Kukkadapu G, Kolodziej C, Gonzalez JP, Waqas MU, Lu T, Aggarwal SK, Som S. Numerical Analysis of Fuel Effects on Advanced Compression Ignition Using a Cooperative Fuel Research Engine Computational Fluid Dynamics Model. J Energy Resour Technol 2021;143(10):102304.

[7] Shah MI, Javed MF, Alqahtani A, Aldrees A. Environmental assessment based surface water quality prediction using hyper-parameter optimized machine learning models based on consistent big data. Process Saf Environ Prot 2021.

[8] Hasle G, Lie K-A, Quak E. Geometric modelling, numerical simulation, and optimization. Springer; 2007.

[9] Zhou K, Yang S. Understanding household energy consumption behavior: the contribution of energy big data analytics. Renewable Sustainable Energy Rev 2016; 56:810–9.

[10] Herrera GP, Constantino M, Tabak BM, Pistori H, Su J-J, Naranpanawa A. Long-term forecast of energy commodities price using machine learning. Energy 2019; 179:214–21.

[11] Torres JM, Aguilar RM. Using deep learning to predict complex systems: a case study in wind farm generation. complex 2018;2018.

[12] Saitta L, Zucker J-D. Abstraction in artificial intelligence and complex systems, 456. Springer; 2013.

[13] Anderson JD. Computational fluid dynamics, 206. McGraw-Hill; 1995.

[14] Basbug B, Karpyn ZT. A study of absolute permeability dependence on pore-scale characteristics of carbonate reservoirs using artificial intelligence. Int J Oil Gas Coal Technol 2008;1(4):382–98.

[15] Brenner M, Eldredge J, Freund J. Perspective on machine learning for advancing fluid mechanics. Phys Rev Fluids 2019;4(10):100501.

[16] Zhou L, Garg D, Qiu Y, Kim S-M, Mudawar I, Kharangate CR. Machine learning algorithms to predict flow condensation heat transfer coefficient in mini/micro-channel utilizing universal data. Int J Heat Mass Transf 2020;162:120351.

[17] Dulhare UN, Ahmad K, Ahmad KAB. Machine learning and big data: concepts, algorithms, tools and applications. John Wiley & Sons; 2020.

[18] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3(6):422–40.

[19] 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.

[20] Raissi M, Perdikaris P, Karniadakis GE. Machine learning of linear differential equations using Gaussian processes. J Comput Phys 2017;348:683–93.

[21] Zhang X, Wu J, Coutier-Delgosha O, Xiao H. Recent progress in augmenting turbulence models with physics-informed machine learning. J Hydrodyn Ser B (English Ed) 2019;31(6):1153–8.

[22] Wu J-L, Xiao H, Paterson E. Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. Phys Rev Fluids 2018;3(7):074602.

[23] Fuks O. Physics informed machine learning and uncertainty propagation for multiphase transport in porous media. Stanford University; 2020.

[24] Fuks O, Tchelepi HA. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. J Mach Learn Model Comput 2020; 1(1).

[25] Maragos, P., V. Charisopoulos, and E. Theodosis, Tropical geometry and machine learning. Proceedings of the IEEE, 2021. 109(5): p. 728–55.

[26] Murugappan M, Mutawa A. Facial geometric feature extraction based emotional expression classification using machine learning algorithms. PLoS One 2021;16(2): e0247131.

[27] Ma J, Chen F, Xu H, Jiang H, Liu J, Li P, Chen CC, Pan K. Face masks as a source of nanoplastics and microplastics in the environment: quantification, characterization, and potential for bioaccumulation. Environ Pollut 2021;288: 117748.

[28] Li B. Research on geometric dimension measurement system of shaft parts based on machine vision. EURASIP J Image Video Process 2018;2018(1):1–9.

[29] Luo L, Liu W, Lu Q, Wang J, Wen W, Yan D, Tang Y. Grape berry detection and size measurement based on edge image processing and geometric morphology. Machines 2021;9(10):233.

[30] Huang, T. and Y. Liu. 3d point cloud geometry compression on deep learning. in Proceedings of the 27th ACM International Conference on Multimedia. 2019.

[31] Leonas KK, Jones CR, Hall D. The relationship of fabric properties and bacterial filtration efficiency for selected surgical face masks. JTATM 2003;3:1–8.

[32] Neupane BB, Mainali S, Sharma A, Giri B. Optical microscopic study of surface morphology and filtering efficiency of face masks. PeerJ 2019;7:e7142.

[33] Kuo C-CJ. Understanding convolutional neural networks with a mathematical model. J Vis Commun Image Represent 2016;41:406–13.

[34] Pang, G. and U. Neumann. 3D point cloud object detection with multi-view convolutional neural network. in 2016 23rd International Conference on Pattern Recognition (ICPR). 2016. IEEE.

[35] Koushik, J., Understanding convolutional neural networks. arXiv preprint arXiv:1 605.09081, 2016.

[36] Fischler MA, Bolles RC. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 1981;24(6):381–95.

[37] Mao Z, Miki A, Mei S, Dong Y, Maruyama K, Kawasaki R, Usui S, Matsushita K, Nishida K, Chan K. Deep learning based noise reduction method for automatic 3D segmentation of the anterior of lamina cribrosa in optical coherence tomography volumetric scans. Biomed Opt Express 2019;10(11):5832–51.

[38] Albawi, S., T.A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. in 2017 international conference on engineering and technology (ICET). 2017. Ieee.

[39] Hatger C, Brenner C. Extraction of road geometry parameters from laser scanning and existing databases. Int Arch Photogram Remote Sens Spatial Inf Sci 2003;34(3/ W13):225–30.

[40] Gikas V, Stratakos J. A novel geodetic engineering method for accurate and automated road/railway centerline geometry extraction based on the bearing diagram and fractal behavior. IEEE Trans Intell Transp Syst 2011;13(1):115–26.

[41] Krawiec K, Bhanu B. Visual learning by evolutionary and coevolutionary feature synthesis. IEEE Trans Evol Comput 2007;11(5):635–50.

[42] Rosin P, Adamatzky A, Sun X. Cellular automata in image processing and geometry. Springer; 2014.

[43] Habib R, Karimi N, Yadollahi B, Doranehgard MH, Li LK. A pore-scale assessment of the dynamic response of forced convection in porous media to inlet flow modulations. Int J Heat Mass Transf 2020;153:119657.

[44] Mesgarpour M, Sakamatapan K, Dalkılıç AS, Alizadeh R, Ahn HS, Wongwises S. An investigation of the thermal behavior of constructal theory-based pore-scale porous media by using a combination of computational fluid dynamics and machine learning. Int J Heat Mass Transf 2022;194:123072.

[45] Pope SB. Turbulent flows. IOP Publishing; 2001.

[46] Weickert M, Teike G, Schmidt O, Sommerfeld M. Investigation of the LES WALE turbulence model within the lattice Boltzmann framework. Comput Math Appl 2010;59(7):2200–14.

[47] Carpenter GA, Grossberg S. A self-organizing neural network for supervised learning, recognition, and prediction, 30. IEEE Communications Magazine; 1992. p. 38–49.

[48] Berkson J. Application of the logistic function to bio-assay. J Am Stat Assoc 1944; 39(227):357–65.

[49] Wang M, Lu S, Zhu D, Lin J, Wang Z. A high-speed and low-complexity architecture for softmax function in deep learning. In: 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). IEEE; 2018.

[50] Janocha, K. and W.M. Czarnecki, On loss functions for deep neural networks in classification. arXiv preprint arXiv:1702.05659, 2017.

[51] De Boer P-T, Kroese DP, Mannor S, Rubinstein RY. A tutorial on the cross-entropy method. Ann Oper Res 2005;134(1):19–67.

[52] Riedmiller M, Lernen A. Multi layer perceptron. Machine learning lab special lecture. University of Freiburg; 2014. p. 7–24.

[53] Harp DR, O'Malley D, Yan B, Pawar R. On the feasibility of using physics-informed machine learning for underground reservoir pressure management. Expert Syst Appl 2021;178:115006.

[54] Kyurkchiev N, Markov S. Sigmoid functions: some approximation and modelling aspects. Saarbrucken: LAP LAMBERT Academic Publishing; 2015.

[55] Zheng, H., Z. Yang, W. Liu, J. Liang, and Y. Li. Improving deep neural networks using softplus units. in 2015 International Joint Conference on Neural Networks (IJCNN). 2015. IEEE.

[56] Chai T, Draxler RR. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geoscientif Model Dev 2014;7(3):1247–50.

[57] Elliott, D.L., A better activation function for artificial neural networks. 1993.

[58] Köksoy O. Multiresponse robust design: mean square error (MSE) criterion. Appl Math Comput 2006;175(2):1716–29.

[59] Zhang, Z. Improved adam optimizer for deep neural networks. in 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). 2018. Ieee.

[60] Yuan Y-x. A modified BFGS algorithm for unconstrained optimization. IMA J Numer Anal 1991;11(3):325–32.

[61] Moritz P, Nishihara R, Jordan M. A linearly-convergent stochastic L-BFGS algorithm. Artif Intell Stat 2016. PMLR.

[62] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J Sci Comput 2021;43(5): A3055–81.

[63] Monjezi M, Jamaati H. The effects of face mask specifications on work of breathing and particle filtration efficiency. Med Eng Phys 2021;98:36–43.