Journal of Computational Physics (2020)

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A fast and scalable mesh generation method of densely packed hollow fibers for membrane separations: application to direct contact membrane distillation

Albet S. Kim[a,*], Hyeon-Ju Kim[b], Deok-Soo Moon[c]

[a]*Civil and Environmental Engineering, University of Hawaii at Manoa, 2540 Dole Street, Holmes Hall 383, Honolulu, Hawaii 96822, USA*
[b]*Offshore Plant and Marine Energy Research Division, Korea Research Institute of Ships and Ocean Engineering, 32, Yuseong-Daero, 1312 Beon-Gil, Daejeon, 305-343, Republic of Korea*
[c]*Seawater Energy Plant Research Center, Korea Research Institute of Ships and Ocean Engineering, 124-32 Simchungsu-Gil, Jukwang-Myon, Kosung-Gun, Gangwon-do, Republic of Korea*

ARTICLE INFO

Article history:
Received
Received in final form
Accepted
Available online

Communicated by

Keywords: Computational fluid dynamics, tetrahedron mesh, unit-cell meshing, mesh assembling, stitching and merging, hollow fibers, membrane distillation

ABSTRACT

Modeling research on membrane distillation requires simulations of coupled momentum, mass, and heat transfer phenomena. Hollow fiber modules are preferred in industrial applications due to their high packing ratio, resulting in the number of fibers on the order of $O\left(10^2 - 10^3\right)$ packed in a vessel. In hollow fiber membrane distillation processes, computational fluid dynamics (CFD) simulations of multi-physics require high-quality meshes for accurate calculations. Mesh interfaces between two different phase regions should conform to satisfy continuity conditions of coupled momentum, heat, and mass transfer. Due to the distinct geometric characteristics of HF packing structures, a scalable meshing method is of great necessity but has not been actively researched. This work developed a numerical method to generate hexagonally packed structures of many fibers by forming a hexagonal unit-cell, consisting of the lumen, membrane, and shell regions. Individual cells are made by duplicating a seed cell and packed to create a self-similar, hexagonal packing of hollow fiber membranes for multi-physics CFD simulations. A new theoretical approach was developed to represent the effusion-like convective mass transfer as conductive heat transfer, and our CFD results were in good agreement with experimental observations. Theoretical methods and numerical algorithms developed in this study can contribute to the improved scalability of CFD simulations from lab-scale modules to pilot-scale systems.

⋆This is an example for title footnote coding.

*Corresponding author
e-mail: albertsk@hawaii.edu (Albet S. Kim)
URL: http://albertsk.org (Albet S. Kim)

1. Introduction

Membrane distillation (MD) is receiving close attention in current literature due to the ever-increasing global demand for fresh water and renewable energy production [1, 2]. During MD processes, the hot feed stream enters an MD module, within which water molecules change the thermodynamic phase between liquid and gas, leaving solutes in the liquid phase[3–5]. Operation types of MD, as classified by vapor condensation methods, include direct contact membrane distillation (DCMD) [6–11], vacuum membrane distillation (VMD) [11–15], sweep-gas membrane distillation (SGMD) [16–18], air-gap membrane distillation (AGMD) [19–22], and liquid-gap membrane distillation (LGMD) [22–24]. Among the above MD processes, DCMD and VMD are widely used due to the feasibility of building systems and performance of producing freshwater, respectively. As DCMD uses a cold freshwater stream to condense vapor gases on the distillate-membrane interface, external condenser units are not required. The direct contact of the migrating vapor and the cold freshwater makes, however, makes DCMD more vulnerable to membrane wetting through a few large pores [25, 26]. VMD uses a vacuum pump to extract evaporated water molecules into the distillate side and condenses the vapor stream in an external condenser. As compared to other MD processes, VMD requires higher energy consumption for the vacuum pumping, but has less wetting vulnerability.

Fig. 1 shows the schematics of membrane distillation, such as, having hot and cold counter-current streams in a DCMD process. The hot-feed flow often carries waste thermal energy with a typical temperature (between 50°C and 80°C). Chemical concentrations of the feed stream are usually much higher than those of drinking water. The cold flow is the distillate product, theoretically, as clean as pure water. The pore spaces are initially filled with air at the given atmospheric pressure. Two membrane interfaces in contact with the hot and cold streams are maintained at vapor pressures of the stream temperatures. Due to the gradient of partial pressure maintained along the pore, water molecules evaporate at the feed-membrane interface, migrate through the pore spaces, and condense at the membrane-distillate interface of the low temperature of cold distillate. The evaporation process at the pore inlet consumes the latent heat (i.e., evaporation enthalpy) for the evaporation of water on the feed side. Therefore, the feed temperature decreases along the same direction of the feed stream. The migrating vapor molecules carry the thermal energy along the pore channel to the cold-stream side, and condense at the membrane-distillate interface maintained at the cold temperature.

Due to the unavoidable latent-heat loss from the transmembrane evaporation, the feed temperature consistently decreases in the longitudinal flow direction, and consequently, the cold-stream temperature increases in the opposite direction to the variation of the feed-stream temperature. The length-averaged driving force (i.e., the transmembrane temperature difference) decreases as the membrane length increases. For example, instead of a long hollow fiber of 120 cm, using four 30 cm fibers aligned in parallel would yield better results in terms of the distillate volume produced per unit time per unit membrane surface area. Because shorter hollow fiber membranes can provide better performance and lower fouling tendency, the number of fibers packed in a vessel often increases tremendously to enhance the thermal efficiency of the evaporation-based membrane distillation processes. Fundamental transport mechanisms and their full mathematical descriptions can be found elsewhere, especially for the hollow fiber membrane distillation [27, 28].

The length scale of the water molecules can be represented using the hydrogen bond length of ~ 0.1 nm, and the membrane pore size is around 0.1μm, which is three orders-of-magnitude larger than the bond length. Moreover, the thickness Δr and length L_f of hollow-fiber membranes are of orders of 0.1 mm and 0.1 m, respectively. Ratios from molecular to pore, pore to thickness, and thickness to length are similarly of an order of $O\left(10^3\right)$. In most MD operations, the produced distillate flux is of an order of 10 LMH ($\mathrm{L/m^2 \cdot hr}$), equivalent to the permeate velocity of $O(10^{-6})$ m/s, and the cross-flow velocity is of an order of $O(0.1) - O(1)$ m/s [23, 29, 30] that is 10^5 times the distillate flux. Because the latent heat (or the evaporation enthalpy) of water is 2.4×10^3 kJ/kg (at 40°C), the rate of heat loss is estimated as $24,000\,\mathrm{kJ/m^2 \cdot h}$, as a product of the distillate flux and the latent heat. The above characteristic parameters of MD processes indicate the feed convection as the basis process in which heat and mass transfer phenomena exist as perturbative sub-processes. The trans-membrane heat transfer from the feed stream to the distillate side strongly depends on the thermal convection rate of the feed in the longitudinal direction, and the mass transfer across the membrane is entirely governed by the heat transfer driven by the transmembrane thermal gradient. The distillate production, preceded by the consumption of the latent heat, is therefore a perturbative phenomena of mass-transfer along the longitudinal direction of the membrane, while the latent heat loss across the membrane surface is comparative to the convective heat transfer from the feed stream to the distillate stream.

A standard module structure of a hollow-fiber module can be described as follows. A cylindrical vessel of a

diameter D_v contains a number of fibers having outer and inner diameters of d_{fo} and d_{fi}, respectively. The number of fibers, N_f, in the vessel can be estimated as

$$N_f = \phi \left(\frac{D_v}{d_{fo}} \right)^2 \tag{1}$$

where ϕ is the occupancy fraction of the fibers estimated using the outer diameter d_{fo}. For example, if $\phi = 0.5$, $d_{fo} = 1.0\,\text{mm}$, and $D_v = 10\,\text{cm}$, then the number of packed fibers is calculated as $N_f = 5,000$. More specific analysis of the fiber packing can be found elsewhere [31–34]. Since the flow distribution in the shell side of fibers is not entirely uniform, operating conditions set for each fiber are different along the radial direction from the central to edge zones of the vessel. Fibers located near the vessel wall may experience higher gradients of temperature and flow velocity in comparison to those near the central vessel zone. For an accurate prediction of the hollow fiber membrane distillation (HFMD), it is required to have either an accurate single-fiber model to represent the mean distillate production of many packed fibers or an efficient computational method that can manage at least a few hundred fibers [35, 36]. Cell model approaches have been successfully applied to many engineering processes including packed-beds, aggregation, and porous media flow, providing reasonably sufficient statistical analyses [7, 37–40]. The representative cell models are, however, incapable of investigating the local variations and fluctuations inside the packed media, but instead, are limited to provide characteristic mean values of variables of interest.

For multi-physics simulations, computational fluid dynamics plays a significant role in understanding engineering phenomena. The two primary categories of CFD are Eulerian and Lagrangian. Eulerian requires a high-quality meshing technique to create the spatially fixed grid within a computational domain [41–43]. Lagrangian, often represented as smoothed particle dynamics (SPH), requires an accurate time-evolution scheme to keep track of fluid elements and proper kernel functions, depending on the researcher's choice [44–48]. To the best of our knowledge given the current research, SPH provides accurate and reasonable results primarily for open channel flows as compared to those of pressurized flows. The solution accuracy and computational efficiency of (Eulerian) CFD simulations strongly depend upon not only the numerical algorithm but also predetermined mesh structures. In theory, a porous medium can be characterized using spatially varying functional forms of hydraulic permeabilities [49, 50]. Similarly, an anisotropic or uni-directional permeability equation can be applied to linearly aligned fibers within a containing vessel. This continuum permeability approach, however, may not provide tangible results because multiple phases are maintained on membrane interfaces, providing discontinuous hydraulic permeabilities.

Kaya et al. [51] studied the shear stress distribution and pressure loss in hollow fiber modules by solving the Reynolds averaged Navier-Stokes equation using the κ-ϵ turbulence model. They concluded that tangential inlet and outlet structures create a rotational flow that causes higher shear stress inside the module. The 127 fibers used in their CFD study were radially concentrated near the center of the cylindrical vessel. Zhuang et al. [52] developed an innovative CFD model to numerically analyze the effects of an inlet manifold on energy consumption and flow distribution and to investigate the non-uniform flow within the membrane module. In the mesh structure that they constructed, 2,682 fibers were evenly spaced in a regular triangular grid, but simulations used only 1/6 of the computational domain, extracted from the full domain using the azimuthal symmetry. The number of fibers, therefore, reduced to 1/6 of the total number. The meshing strategy used by Zhuang et al. [52] is based on the grid convergence index (GCI) check using a bundle of 37 fibers. As described above, the effective number of fibers used for CFD simulations were reported as an order of $O\left(10^2\right)$ [51, 52], which is at least one order of magnitude lower than those found in industrial applications. Both Kaya et al. [51] and Zhuang et al. [52] used ANSYS FLUENT, a popular and reliable commercial CFD software of which parallel meshing capability is limited to only prism-shaped meshes.

In this case, meshing the fluid volume of the shell regions becomes more challenging because the computer runtime for meshing drastically increases with respect to the number of fibers. Among all the potential areas in which to improve the capability of CFD simulations, the primary bottleneck of current simulation research lies within, in our opinion, the maximum number of fibers and efficient, scalable meshing strategies. In this work, we develop an intrinsically scalable and parallelizable meshing technique for a large number of evenly distributed fibers within a vessel. Generated meshes can be used for, but are not limited to, hollow fiber membrane separations such as MD using OpenFOAM [53], an open-source CFD software package.

2. Theory and simulations

2.1. Meshing using the unit-cell approach

2.1.1. Overview

A direct meshing of a few thousand fibers in a cylindrical vessel is a prerequisite for CFD simulations, but is a technically difficult task at the level of industrial applications. A conventional method for the mesh generation has a few standard steps as follows. A geometry or structure file is either made or given, having one of the standard formats such as stereolithographic (STL), the standard for the exchange of product (STEP), and other model data. The geometry file is inputted to a mesh generation software such as NETGEN [54–56], TETGEN [57, 58], and GMSH [59] that are open-sourced, and other commercially available ones. After surface and volume meshes are generated and boundary names are assigned, the generated mesh is then imported to specific CFD solvers. On the other hand, `blockMesh` and `snappyHexMesh` are meshing utilities included in OpenFOAM using input files of STL or STEP formats.

Aftosmis et al. [60] developed a generation method for rapid and robust Cartesian meshes using a component-based geometry. Their algorithm intersects the components so as to extract the wetted surface. To generate the volume mesh, surface triangulation was used to divide cells composing the initial uniform coarse grid. Outer component interfaces match either open or solid surfaces. For large-scale mesh construction, parallel meshing using multiple cores is an efficient method, which has been actively researched for only a few decades [60–65]. Within our current research, an innovative meshing approach was developed based on a unit-cell model, which can be viewed conceptually as an extended work of Aftosmis et al. [60]'s component-based method. Unlike Aftosmis's work, the new method combines unit components consisting of lumen and shell regions of fluids across an HF membrane. The solid interfaces include the inner and outer surfaces of the HF membrane and the inner wall of the containing vessel. In reality, identical fibers are packed semi-regularly in cylindrical vessels. A single fiber within an imaginary cylindrical cell can be used as a representative system to predict the production performance of membrane distillation processes. To the best of our knowledge, however, sub-domain volume per unit fiber can be ideally described using hexagonal rod-like cell structures to include multiple fibers within the vessel. Our unit-cell approach, therefore, uses hexagonal cells that are perfectly packable in the longitudinal direction without any interstitial gaps between adjacent hexagonal cells.

2.1.2. Hexagonal layer formation

Fig. 2 shows the mesh generated for the lumen region in the (a) orthogonal and (b) perspective views, which is divided into inner-square and outer-radial zones. On a surface normal to the axial direction, the inner-square zone has $n_x \times n_y$ grid, where $n_x = n_y = 12$, and the outer zone is radially divided by $n_r = 8$ layers. The fiber is divided longitudinally by n_z, calculated as $\text{int}\left(2L_f/d_{\text{fo}}\right) + 1$, where $\text{int}\left(2L_f/d_{\text{fo}}\right)$ means a rounded-down length-to-radius ratio. The inner-square zone is made to avoid unnecessarily concentrated mesh vertices near the center and hence to generate an evenly distributed, central mesh structure. The radial mesh in the outer lumen zone will match the lumen-side interface of the porous HF membrane surrounding the void lumen zone. Fig. 3 shows the hexagonal unit-cell structure, consisting of three regions, where the central zone is identical to the lumen mesh, shown in Fig. 2. The green radial-zone mesh indicates the solid-membrane section without including micro-porous structures. The first circular boundary from the center indicates an interface between the void lumen and solid-membrane surfaces, and the second circular boundary connects the solid membrane (green) and void shell regions (red). The outermost zone between the external fiber surface (at $r = d_{\text{fo}}/2$) and the hexagonal boundaries make shell spaces, which will contact six adjacent surfaces from six different hexagonal cells. The unit hexagonal mesh cell is fully described using four length parameters, the inner and and outer diameters of the hollow fiber, i.e., d_{fi} and d_{fo}, respectively, the length of the fiber, L_f, and the hexagon diameter d_{hx}, i.e., the farthest distance between two hexagon vertices. As the occupancy fraction ϕ of Eq. (1) can be interpreted as the ratio of the circular area of diameter d_{fo} to the hexagon area of diameter d_{hx}, the ratio of d_{hx} to d_{fo} is calculated as

$$\frac{d_{\text{hx}}}{d_{\text{fo}}} = \sqrt{\frac{2\pi}{3\sqrt{3}\,\phi}} = \frac{1.0996}{\sqrt{\phi}} \tag{2}$$

In the exemplary case, we used $d_{\text{fi}} = 0.35$ mm, $d_{\text{fo}} = 0.70$ mm, $L_f = 71$ mm, and $\phi = 0.64$ and calculated $d_{\text{hx}}/d_{\text{fo}} = 1.3745$ or $d_{\text{hx}} = 0.9622$ mm.

At the lumen/membrane and membrane/shell interfaces, the surface meshes of the two regions are perfectly matched not to require extra interpolations for real-time data-transfer between regions. The hexagonal shells of Fig. 3 are duplicated, rotated, and translated to pack in a hexagonal grid for a large-scale mesh generation.

2.1.3. The number of fibers

Fig. 4 shows a sequential method to form a hexagonal structure of packed fibers. A numbered point indicates the lumen center of a hexagonal cell, shown in Figs. 2 and 3. For example, point 0 is the original seed cell located at the center of the cylindrical vessel, which is named as the zeroth layer with the layer index $n_l = 0$ having a single fiber. Six points from 1 to 6 around 0 makes the first layer of $n_l = 1$. The second layer ($n_l = 2$) has 12 ($= 6n_l$) points from 7 to 18. From the zeroth to the second layer, the total number of fibers is therefore $1 + 6 + 12 = 19$. The third layer ($n_l = 3$), similar to the second layer, has 18 ($= 6n_l$) points after the 19 points from the zeroth to the second layers. Then, one can conclude that the $n_l{}^{\text{th}}$ layer has $6n_l$ points. The total number of fibers, n_f, from the 0^{th} to $n_l{}^{\text{th}}$ layer is then equal to

$$n_f = 1 + \sum_{i=1}^{n_l} 6i = 1 + 3n_l\,(n_l + 1) \tag{3}$$

Table 1 shows n_f versus n_l data for selected number of layers from 0 to 100, indicating that n_f becomes orders of $O\left(10^2\right)$, $O\left(10^3\right)$, and $O\left(10^4\right)$ with $n_l = 6$, 18, and 60, respectively. The lowest number of n_l for $n_f \geq 1,000$ is 18. In typical industrial cases, n_f is close to 5,000, which can be modeled using $n_l = 40$ or 41, giving $n_f = 4921$ or 5167, respectively. Fig. 4 implies that the HF hexagonal packing to form n_l^{th} layer follows the helical sequential addition of fibers as many as $6n_l$ from fiber index $1 + 3n_n\,(n_l - 1)$ to $3n_l\,(n_l + 1)$. For example, the third layer ($n_l = 3$) has a total of 18 fibers from 19 to 36.

Fig. 5 shows a closely packed hexagonal structure consisting of (a) $n_l = 4$ with $n_f = 61$ and (b) $n_l = 20$ with $n_f = 1261$ fibers, which are to visually explain a medium sized hexagonal packing to show collective features of packed fibers and to demonstrate the scaling capability of our new method to include a large number of fibers of an order of $O\left(10^3\right)$, respectively. Each fiber has 6 touching interfaces from the six nearest neighbors except ones on edges. The fiber configuration is identical to those of points (of lumen centers) shown in Fig. 4. Within each hexagonal cell (as shown in Fig. 3), there are three physical and four mesh zones. The extended radial mesh in the shell region was bounded by 6 equal-sized planes in contact with those of 6 nearest neighbor plates (except the fibers of the highest n_l). Within a hexagonal cell, vertices and edges are perfectly matched along both radial and longitudinal directions at any interfaces between two adjacent regions, which automatically ensures the perfect mesh conformation. Therefore, no specific interpolation algorithm is necessary to transfer data between adjacent cells. The link between two adjacent hexagonal cells requires a nearest neighbor list of i^{th} cell's $S_i{}^{\text{th}}$ surface, touching with j^{th} cell's $S_j{}^{\text{th}}$ surface, where the fiber indices i and j range from 0 to $n_f - 1$, and surface indices S_i and S_j vary from 0 to 5. Table 2 shows a complete list of surface S_i of the i^{th} hexagon paired with surface S_k of the k^{th} hexagon. This pair list was used to stitch conforming meshes using the OpenFOAM utility, `stitchMesh`, to test generated meshes (see section (2.2)). Regardless of the conformity of the interface meshes of two hexagonal cells, this list can be used to apply the arbitrary mesh interface (AMI) boundary conditions [66] at the six interfaces per unit hexagonal cell. As the meshes of the unit elements, i.e., the lumen, membrane, and shell, are algebraically designed for the perfect match, numerical errors due to the mesh merging are automatically maintained at the minimum.

2.1.4. Matching interface pairs

In the above packing method, layer 0 has one hollow fiber of an index HF-0 and layer 1 has hollow fibers of indices from HF-1 to HF-7. Similarly, an arbitrary n_l (> 0) has k fibers, satisfying a condition:

$$3\,(n_l - 1)\,n_l + 1 \leq k \leq 3n_l\,(n_l + 1) \tag{4}$$

For example, k ranges from 7 to 18 for $n_l = 2$ and from 19 to 36 for $n_l = 3$, of which fiber locations are shown in Fig. 4. Note that for an arbitrary HF-k, there are always 6 nearest neighbors. For example, HF-14 in the 2^{nd} layer of Fig. 4 has nearest neighbors of 5 and 4 in the first layer, 13 and 15 in the second layer, and 29 and 30 in the third layer. These 6 neighbors are located in the same and neighbor layers of HF-14, i.e., n_l and $n_l \pm 1$. Among the 37 fibers (from $k = 0 - 36$), finding k^{th} fiber's nearest neighbors is of necessity to match its six neighboring interfaces. The simplest method is to calculate the center-to-center distance between HF of indices k and k', both ranging from 0 to

36 except $k = k'$. Due to the geometric characteristic of the hexagonal packing, distances from the center of HF-k to the centers of the six nearest neighbors ($k' = 0 - 5$) are equally 2 in a dimensionless scale, which is shown in Fig. 4. For a hexagonal packing of a larger n_l (> 3), searching for the nearest neighbors of HF-k in layer n_l can be simplified by calculating the center-to-center distances of HF-j, located only in layer n_l and $n_l \pm 1$, using Eq. (4).

If HF-j is a nearest neighbor of HF-k, then, one can calculate the relative position of j to that of k, i.e., $\boldsymbol{r}_{jk} = \boldsymbol{r}_j - \boldsymbol{r}_k$, its unit vector

$$\boldsymbol{n}_{jk} = \frac{\boldsymbol{r}_{jk}}{\left|\boldsymbol{r}_{jk}\right|} = \cos\theta_{jk}\hat{\boldsymbol{x}} + \sin\theta_{jk}\hat{\boldsymbol{y}} \tag{5}$$

and the relative angle, θ_{jk}, ranging from $-\pi$ to π as

$$\theta_{jk} = \tan^{-1}\left(\frac{\boldsymbol{n}_{jk}\cdot\hat{\boldsymbol{y}}}{\boldsymbol{n}_{jk}\cdot\hat{\boldsymbol{x}}}\right) \tag{6}$$

where $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ are unit vectors in x- and y-directions. To have θ between 0 and 2π, one can use the periodicity of the polar angle θ:

$$\theta = \mathrm{mod}\left(\theta + 2\pi, 2\pi\right) \tag{7}$$

where the operator `mod` calculates the remainder of base 2π (after the comma). As the surface index of HF j and k, i.e., S_j and S_k, respectively, vary from 0 to 5 as shown in Fig. 6, the angular difference θ_{jk} is calculated and listed in Table 3. A relationship between S_j and S_k is obtained as

$$S_k = \mathrm{mod}\left(S_j + 3, 6\right) \tag{8}$$

and that of I_θ and S_k as

$$S_k = \mathrm{mod}\left(\left(I_\theta + 1\right)/2, 6\right) \tag{9}$$

where $I_\theta = \theta_{jk}/\frac{\pi}{6}$, which has a relationship with S_j as follows:

$$I_\theta = \mathrm{mod}\left(2\left(S_j + 3\right) - 1, 12\right) \tag{10}$$

By calculating θ_{jk} using Eq. (7) between HF-j and HF-k, one can determine the indexes of touching surfaces S_j and S_k using Eqs. (8)–(10).

Figs. 7(a) and (b)-(e) show mesh structures generated within a circular boundary of a cylindrical vessel at global and local aspects, respectively. We restricted our current simulation study to shell flows because the membrane zone will be later used for conductive and convective heat transfer and lumen flows are independent to each other due to the same inlet boundary condition. Fig. 7(a) includes several fiber numbers, which are identically located to those in Fig. 4. HF-0 is the seed fiber to be duplicated and translated to form the whole hexagonal fiber packing structure, HF-19 and HF-20 are the first and second fibers in the 3rd layer, whose rotational symmetric counterparts are HF-28 and HF-29, respectively. HF-31 is the top fiber in the packing structure, whose mesh grid is identical to those of HF-19, HF-28, and three other corner fibers. The number of radial divisions is equal for all corner fibers along the distance from any outer-most sides of the corner fibers to the vessel wall, and therefore HF-19 has a very dense grid toward the wall, while other edge fibers have more sparse grids. The division numbers, i.e., n_x, n_y, and n_z, should be carefully selected to conduct accurate as well as fast-running simulations. As most CFD simulation studies are to fundamentally understand, rigorously predict, and optimally design fluid mechanical systems, simulation results are often provided without 1:1 experimental verification, and therefore the simulation outcomes should be self-consistent physically and self-verifying numerically. (See section 2.2.3 for details.) Fig. 7(b) shows the representative hexagonal cell of HF-0, consisting of only shell spaces, having six interfaces with the same number of neighbors. The black solid lines indicate boundaries between touching neighbor cells, along which mesh grids (in the radial direction) are perfectly matched and hence conformed. Fig. 7(c) shows the top HF-31 as one of six corner fibers that are HF-19, 22, 25, 28, and 34, as discussed above. Note that these corner fibers have very dense grid sizes between their hexagonal surfaces and the inner vessel wall. The fine-grid meshes are identified by enclosing black lines. An edge mesh zone of HF-19 adjacent to HF-36 is symmetric to that adjacent to HF-20. Similarly, two edge mesh zones of HF-20 and HF-29 are symmetric along the dashed lines from their outermost vertices to the vessel wall. Figs. 7(d)-(e) shows how the edge-zone meshes are generated by filling gaps between hollow fibers and the inner vessel wall. HF-19 and HF-28 have three

extra edge-zones: one almost touching the inner wall, and two others having symmetrically identical configurations. All other edge-located fibers have only two extra edge-zones, similar to HF-20 and HF-29. Radial extending of two interface-lines can define the extra edge-zones. Filling the edge-gaps makes the computational domain more closely mimic the industrial applications although the total number of fibers used here, 37, is still much smaller than that of real-world applications, but a reasonable number for lab-scale experiments. As each fiber's hexagonal cell is identical to each other, the same mesh configuration exists at both the inlet and outlet surfaces. The shell meshes at the inlet surface are merged to a 2D plane mesh having 36 holes, of which diameter is equal to the outer diameter of the fibers. At the outlet surface, however, the individual shell-outlet meshes remained separated, but stitched to allow any inter-fiber transport. In this case, one can investigate performance of all individual fibers regardless of the total number. A run-time for each pairwise stitching is proportional to the domain mesh size because searching for vertex indices of two surfaces is required in the computer memory spaces. On the other hand, the AMI boundary conditions can be implemented by indicating two touching shell-surfaces for OpenFOAM simulations. The AMI is a competitive method with permanent stitching for a fiber packing of three or more layers.

2.1.5. Application generality and conversion utilities

In this work, we developed a new algorithmic paradigm to generate meshes for regularly packed, equal-sized cylindrical objects and applied the technique to model HFMD processes. We generate a unit-cell mesh (i.e., the representative mesh-grid unit), duplicate it as many times as the necessary number of hollow fibers in a membrane module, and merge them using the AMI technique [66] or `stitchMesh` utility. In OpenFOAM, application of the AMI method does not require computationally expensive post-processing but simply instructs the OpenFOAM solver to touch pairs of neighboring cells. The unit-cell mesh can be made by using `blockMesh` utility or any other available mesh software programs, as long as the stitching or merging is possible as a preprocessing. OpenFOAM has a variety of mesh conversion utilities to or from popular mesh formats. Therefore, the unit-cell mesh can be generated using any mesh software and converted into OpenFOAM, or a full mesh formed using the current method can be exported to other popular mesh formats [67].

2.1.6. Performance and scalability of mesh generation

A mesh of many hollow fibers is created by generating a seed unit-cell mesh and duplicating the seed mesh $n_f - 1$ times. In this case, the generation time of the entire mesh is equal to the time required to generate a unit-cell mesh plus $n_f - 1$ times duplication using a "copy and paste" utilities included in any operating systems. To pack a number of fibers in the hexagonal grid shown in Fig. 3, the unit-cell needs to be copied, rotated, and translated to a specific position. Afterward, touching surfaces of two adjacent shells, i.e., the nearest-neighbor cell's touching side-surfaces, are linked as arbitrary mesh interface (AMI) boundaries. Phase interfaces between the shell (liquid) and fiber (solid) and between fiber (solid) and lumen (liquid) are linked to exchange temperature information between two different thermodynamic phases.

To check the mesh generation's performance, we select a sample hollow fiber to be used for a full-scale simulation in section 2.3. The fiber's inner and outer diameters are 0.6 mm and 0.82 mm, respectively, and its length is 200 mm, with the fiber packing fraction of 0.222. The fiber's cross-sectional surface is similar to those shown in Fig. 3, having 30 radial and 32 azimuthal divisions, respectively. The number of longitudinal division is set equal to int $\left(l_f / (3d_{\text{fi}}) + 1\right)$, which is calculated as 112 in this case. We used two different computer hardware systems to measure generation time of this HF unit-cell mesh, having the same random access memory of 128 Gigabyte. Using a single-core, an Intel processor (Intel(R) Core(TM) i9-10920X CPU @3.50 GHz) took 6.97 seconds, and AMD Ryzen Threadripper 3960X processor took 71.45 % of the run time of the Intel Processor, as summarized in Table 4.

After we duplicated this unit-cell mesh using system commands, we assigned different patch names using text-editing scripts. Translation and rotation of the duplicated unit-cell mesh were conducted using `transformPoints`, a utility embedded in OpenFOAM v1812. Because this assembling process does not require any additional mesh generation, the run time needed for assembling depends on the I/O speed of the computer hardware, which is often negligible compared to that for the initial unit-mesh generation. We developed and used Python and BASH scripts to run the assembling process sequentially for n_f hollow fibers, to assign the AMI boundary pairs, and to make regional interfaces between the liquid phase (of the lumen and shell) and the solid phase (of the porous membrane). Therefore, the run-time to generate a entire mesh of n_f packed hollow fibers are only a few folds of the initial generation time

for a unit-cell mesh. If the run-time required to perform system calls and commands is negligible, then our meshing performance is almost independent of the number of hollow fibers to be assembled.

2.2. CFD Simulations for flow

To the best of our knowledge, OpenFOAM does not have official solvers for membrane distillation due to the intrinsic coupling between the vapor migration and the transmembrane heat-conduction. One possible method is to calculate the difference in heat transfer rates of two ideal cases [68]. The first case is to use the membrane as a porous material, as it is, allowing convective heat transfer carried by migrating vapor molecules. The second case is to treat the membrane as porous but impermeable to vapor migration, containing stationary vapor within the pore spaces. The difference in the heat flux can be calculated as a function of position on the membrane interfaces, i.e., inner to the lumen and outer to the shell zones and then converted to the vapor generation rate, equal to the condensation or distillate production rate. Nevertheless, as the heat transfer occurs on the basic framework of the momentum transfer, only the basic fluid flow is simulated in this section to test the meshing method only.

As lumen spaces are geometrically identical and mechanically separated from each other, a lumen region can be treated as an isolated, long cylindrical channel, where the Hagen–Poiseuille equation can be used as a good approximate solution. Complexity of the fluid flow occurs in the interstitial shell spaces between packed fibers, which was investigated using boundary conditions summarized in Table 5. The inlet indicates one end of the long hexagonal packing where the fluid flow enters, and the outlet is the opposite end to the inlet, where the fluid flow exits. The surface indicates the outer boundary of each fiber (at $r = d_{\text{fo}}/2$ from lumen centers), which is the interface between the solid membrane and fluidic shell region. At the inlet, the fluid velocity is set as a constant so that the gradient of the pressure is zero. At the outlet, the pressure is set to be zero, by which the inlet pressure value is calculated as correlated to the longitudinal fluid velocity. This type of the outlet flow condition used is "`PressureInletOutletVelocity`", one of the embedded boundary conditions in OpenFOAM, which means that zero gradient of U ($\nabla U = 0$) is applied for outflow, and otherwise the velocity is obtained from the patch-face normal component of the internal-cell value for inflow. In our current applications, the inflow does not occur so that the outlet boundary condition for U is equivalent to its zero gradient. From the shell side, the fluid cannot penetrate the fiber outer surface, on which the pressure gradient does not exist. As a typical pore size of MD membranes is of an order of $O\left(10^{-7}\right)$ m, the no-slip boundary condition ($U = 0$) is employed at the outer surface of HF membranes. In reality, vapor evaporates on the membrane surface, which is assumed to have no impact on the no-slip boundary condition.

We assumed the presence of a steady state to verify the mesh conformation, which should be checked using a transient CFD solver for more rigorous simulations with grid convergence index (GCI) analysis. To investigate the pressure and flow profiles at a transient state, `pisoFoam` is used, which is a transient solver included in OpenFOAM for incompressible, turbulent flow, using the PISO (Pressure-Implicit with Splitting of Operators) algorithm [69–71]. As the Reynolds number of most HF membrane filtration is small, the turbulent option was turned off in testing runs of `pisoFoam` solver.

2.2.1. Momentum Transfer

The Navier-Stokes (NS) equation describes the momentum transfer phenomena in the lumen and shell regions. The most general form of the NS equation is

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho}\nabla P + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{g} \tag{11}$$

where P is hydraulic pressure, $\boldsymbol{u}$ is the fluid velocity, ρ and ν are the mass density and the kinematic viscosity of the fluid, and $\boldsymbol{g}$ is the gravitational acceleration. In a steady-state having insignificant viscous heating, Eq. (11) is simplified to

$$\nu \nabla^2 \boldsymbol{u} = \frac{1}{\rho}\nabla P + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{g} \tag{12}$$

Because the mass flux across the membrane in the radial direction is much smaller than the convection in the longitudinal direction, we assume that mass flow rates at the inlet are equal to that at the outlet for both the lumen and shell

regions. To simplify the momentum transfer in the lumen and shell regions, we ignored mass of water, evaporating on the feed-membrane interfaces. Reynolds number of the lumen flow is approximately

$$\mathrm{Re} = \frac{Du}{\nu} \simeq \frac{O(10^{-3})\mathrm{m} \times O(10^{-1} - 10^{0})\mathrm{m/s}}{O(10^{-6})\,\mathrm{m^2/s}} = O(10^{2} - 10^{3})$$

which indicates that the lumen and shell flows are most likely within the laminar regime unless the fluid speed is close to or exceed 1.0 m/s. In this case, we restrict ourselves to a membrane distillation operation of the laminar flow regime for mesh testing. We also assume that water's kinematic viscosity does not change significantly from that at inlet temperature, which was improved in section 2.3 using the temperature-dependent kinematic viscosity in the Navier-Stokes equation. Across the porous membrane, the heat transfer consists of the conductive and convective transfer. The conductive heat transfer occurs through the solid part of the porous membrane, and the convective heat transfer is originated by evaporation enthalpy carried by diffusing vapor molecules. The vapor's mass transfer rate through the void membrane spaces is several orders-of-magnitude smaller than the feed and distillate flow velocities, which cannot be directly calculated using any regular solvers of OpenFOAM.

2.2.2. *Flow and Pressure*

Fig. 8 shows profiles of velocity magnitude $|\boldsymbol{U}|$ at (a) the inlet and (b) outlet surfaces along the longitudinal direction, obtained using the boundary condition described in Table 5. Near the outlet, flow speed is observed inhomogeneous around the fiber surfaces, but angle-dependent, i.e., seemingly $\pi/6$ periodic. The angular periodicity strongly implies that an azimuthal variation of heat and mass transfer around the outer fiber surface needs to be accurately investigated so as to calculate the overall distillate production rate. The no-slip boundary condition applied to the outermost surfaces confirms negligible velocity, as shown in blue.

Fig. 9 shows well aligned profiles of pressure p and flow speed U, where p calculated in OpenFOAM indicates the hydraulic pressure divided by the density of incompressible fluid having a unit of [J/kg]. At the inlet and outlet boundaries, the normal gradient and the absolute value of the pressure are set to be zero. Due to the presence of fibers, perfectly aligned along the longitudinal axis, the fluid flow must be dominant in the longitudinal direction, as verified by the monotonous and linear variation of the pressure in the x-axis, as shown in Fig. 9(a). The three horizontal stripes indicates exterior surface of HF-18, 0, and 12 (as shown in Fig. 8) from the bottom to top, where the pressure profile shows a linear variation in the longitudinal direction, but the pressure gradient normal to the fiber surface is set to be zero. In Fig. 9(b), the three blue stripes indicate zero velocity on the outer walls of HF-18, 0, and 12 due to the no-slip boundary condition assigned (see Table 5.). The six thin red stripes beside the blue stripes show fast flow velocities generated in the triangular interstitial spaces formed by HF sets of 36-18-35, 18-1-6, 1-0-6, 0-3-4, 3-12-4, and 12-26-27. The four yellow stripes mean slow longitudinal flow between inter-cylinder surfaces between HF sets of 35-36, 6-1, 4-3, and 27-26. In addition, the two thick red stripes at the lower and upper boundary show edge flows as fast as 0.16 m/s around HF sets of 26-27 and 35-36. This simulation implies that inflow patterns near the inlet may not cause noticeable differences in membrane performance if the fiber length is much longer than the fiber outer diameter.

Fig. 10 shows flow rate values calculated around individual fibers: (a) parallel case is the default case with uniform inlet velocity of $u = (u_{x0}, 0, 0)$ m/s, where $u_{x0} = 0.1$ m/s, (b) axial case has a lateral velocity component in addition to the velocity profile of the parallel case, i.e., $u = (u_{x0}, u_{x0}\sin\theta\cos\phi, u_{x0}\sin\theta\sin\phi)$ m/s, where $\theta = \pi/6$ and $\phi = \pi/4$, and (c) oblique case has an inlet flow profile of $u = (u_{x0}\cos\theta, u_{x0}\sin\theta\cos\phi, u_{x0}\sin\theta\sin\phi)$ m/s. The 37 fibers (of index from 0 to 36) are divided into central and edge groups. The central group consists of 19 fibers (0–18) located near the center. Within the edge group, fibers of index 19, 22, 25, 28, 31, and 34 are located at six corners of the global hexagonal packing structure. For each run, flow rate per unit fiber is calculated around the fiber. Fibers in the central zone are closely packed and therefore the flow rate is lowest. The edge-group fibers have two zones to calculate the flow rate: default hexagonal cell area and extra area between the fiber wall surfaces to edge gaps. Since these edge gaps provide extra void spaces between the containing wall and edge fibers, the fibers in the edge-group provide the highest flow rate, followed by the corner fibers.

2.2.3. *Grid Convergence Index*

The validity of simulation results are verified using the grid convergence index (GCI) [72], of which the goal is written as

"The Grid Convergence Method (GCI) method (and is based on Reynolds number) described herein is an acceptable and a recommended method that has been evaluated over several hundred CFD cases. If authors choose to use it, the method per se will not be challenged in the paper review process".

To calculate a GCI, we solve the shell flow problem using three grid sizes based on $n_x = n_y$ and n_r, as shown in Table 6. The basic procedures are as follows:

1. Set three grid sizes h_i for $i = 1 - 3$, where h_1 and h_2 indicates the finest and intermediate values, and h_3 is the most coarse one, i.e., $h_1 < h_2 < h_3$, satisfying $r_{21} = h_2/h_1 > 1.3$ and $r_{32} = h_3/h_2 > 1.3$, where the value 1.3 is based on experience.
2. Perform three simulations using the three values of h_i and calculate a representative value f_i for each h_i, and calculate $\epsilon_{32} = f_3 - f_2$, $\epsilon_{21} = f_2 - f_1$, and
$$s = \text{sign}\left(\epsilon_{32}/\epsilon_{21}\right) \tag{13}$$
3. Using values obtained in the above steps, calculate using an initial guessed value of $p > 1$, e.g., $p = 1.2$:
$$q = \ln\left(\frac{r_{21}^p - s}{r_{32}^p - s}\right) \tag{14}$$
$$p = \frac{1}{\ln\left(r_{21}\right)}\left|\ln\left(\left|\frac{\epsilon_{32}}{\epsilon_{21}}\right|\right) + q\right| \tag{15}$$
4. If the updated p is equal to its previous value within a tolerance error, then the updated p is the GCI value. Otherwise, update p again by calculating q.

Fig. 11 shows GCI values of flow rates calculated over cross-sections perpendicular to the flow direction around fibers. Fibers in the central group have highest but similar GCI values around 1.2%. The edge group consists of the rest of 18 fibers (19–36) located in the outermost layer, surrounded by 5 (or less) fibers and facing the inner surface of the containing wall. GCI values of edge-group fibers are below 1.0%. Among fibers in the edge-group, six fibers of index 19, 22, 25, 28, 31, and 34 have GCI values (0.796 ± 0.121 %) higher than other edge-group fibers (0.345 ± 0.0374 %) but lower than that of central-group fibers (1.2 ± 0.0081 %). These fibers are located at the six corners of the global hexagonal packing, close to the inner containing wall. The fibers in the edge group, but not located at one of six corners have lower GCI values (triangular symbols in Fig. 11). By this observation one can conclude that fibers tightly surrounded by neighbor fibers or located close to a containing wall have higher GCI values. All the fibers in this test have GCI values below 1.2%, which indicates that the mesh generated using $n_x = n_y = 12$ and $n_z = 8$ gives acceptable discretization error. A same accuracy-level of GCI tests were attempted for tetrahedron meshes, but the short distances in the radial direction precluded unconditionally successful mesh generations. Detailed discussion is included in Appendix (Appendix A.1)

2.3. Coupled Transfer Phenomena

Now we apply the mesh generated using the unit-cell approach described in the previous sections to a direct contact membrane distillation (DCMD) process. A hot-out-cold-in operation in a counter-current mode is considered, i.e., a hot feed and cold distillate streams flow in the shell and lumen spaces, respectively, in opposite directions to maintain a pseudo-constant transmembrane temperature gradient. We ran a series of CFD simulations using an OpenFOAM solver, `chtMultiRegionSimpleFoam`, and compared results with experimental observations in the literature [73]. The thermal energy carried by the feed stream is transferred across the membrane to the distillate side in two different pathways. The first pathway is the thermal conduction through the solid portion of the porous membrane, made of a hydrophobic polyvinylidene difluoride (PVDF). A conventional heat-conduction equation may describe the heat-transfer phenomena, often converging to Laplace's equation for temperature in the absence of a heat source or sink. The second pathway is the convective transfer by migrating vapor molecules through the void membrane spaces initially filled with air. This convective heat transfer requires the evaporation of water at the feed-side membrane boundary by consuming thermal energy, supplied by the hot-feed stream, as much as the latent heat. At the distillate-side membrane surface, maintained at a temperature as low as the incoming cold distillate flow, the migrated vapor condenses to a liquid phase (i.e., freshwater), releasing heat as much as the condensation enthalpy per vapor molecule.

2.3.1. Heat transfer

The most general governing equation for heat transfer in an incompressible fluid is

$$\rho c_p \frac{DT}{Dt} = \nabla \cdot (\boldsymbol{\kappa} \cdot \nabla T) + \frac{\eta}{2}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right)^2 + S_h \tag{16}$$

where T is temperature, c_p is the specific heat at a constant pressure, κ is the thermal conductivity tensor often assumed to be isotropic, and S_h is a heat source or sink. In a steady state where the viscous heating is not significant, Eq. (16) is reduced to

$$\rho c_p \boldsymbol{u} \cdot \nabla T = \nabla \cdot (\boldsymbol{\kappa} \cdot \nabla T) \tag{17}$$

or

$$\nabla \cdot (\boldsymbol{\chi} \cdot \nabla T) - \boldsymbol{u} \cdot \nabla T = 0 \tag{18}$$

where $\boldsymbol{u}$ is the fluid velocity, coupled in Eqs. (12) and (17), and the thermal diffusivity

$$\chi \equiv \frac{\kappa}{c_p \rho} = \frac{\nu}{\mathrm{Pr}}$$

is represented using Prandtl number

$$\mathrm{Pr} = \frac{c_p \mu}{\kappa} \tag{19}$$

as a ratio of diffusive and thermal diffusion phenomena. Within the membrane material including its interfaces to the feed and distillate streams, the fluid velocity is set to be zero, i.e., $\boldsymbol{u} = 0$, so that the heat equation (17) is reduced to Laplace's equation for T if the thermal conductivity is isotropic constant.

2.3.2. Mass transfer

An engineering goal of MD processes is to maximize the transmembrane mass transfer, which is determined by the effectiveness of the convective heat transfer. Because the distillate flux stems from the vapor diffusion through the membrane pore spaces at the molecular level, direct calculation of the distillate flux is a challenging task at the continuum level of CFD simulations. A possible method present in this work is to construct an equivalent conduction form to the convective vapor transport in the governing heat equation.

When the pore length, i.e., hollow fiber thickness, is much longer than the pore size, the vapor migration within pores can be approximated as the standard "effusion" phenomena[14]. The inlet and outlet of a long channel are maintained at different pressures and temperature, and the transfer of rarefied gases is primarily driven by Knudsen diffusion. In this case, the distillate flux along the long straight pore can be represented as

$$J_w = \frac{4}{3} d_p \frac{\Delta\gamma}{l_p} \tag{20}$$

where d_p and l_p are the diameter and the length of a membrane pore, respectively, γ is the incidence rate[14], defined as

$$\gamma = \frac{P}{\sqrt{2\pi M_w \mathcal{R} T}} \tag{21}$$

and $\Delta\gamma$ is a transmembrane difference of γ between the hot feed and cold distillate regions.

At each side of a membrane pore, pressure P is approximate to the vapor pressure $P_v(T)$ at equilibrium temperature T at the membrane-fluid interfaces. In this case, Eq. (20) can be generalized in a vector form of

$$\boldsymbol{J}_w = -\frac{4}{3}\frac{d_p}{\tau}\gamma_T \nabla T \tag{22}$$

where τ is diffusive tortuosity, representing the reduction in diffusive flux due to the tortuous diffusive paths as compared to mono-dispersed, straight pores having $\tau = 1$. We employed a diffusive tortuosity known to be valid for random arrays of freely overlapping cylinders[75]:

$$\tau = 1 - \ln \epsilon \tag{23}$$

Using the given porosity of $\epsilon = 0.738$ for the PVDF membrane [73], we used $\tau = 1.3038$ in the following simulations. In Eq. (22), γ_T indicates how the incidence rate changes across the membrane with respect to the transmembrane temperature difference:

$$\gamma_T = \left[\frac{\mathrm{d}\gamma}{\mathrm{d}T}\right] \simeq \frac{\gamma(T_h) - \gamma(T_c)}{T_h - T_c} \tag{24}$$

where T_h and T_c are temperatures of the hot-feed and cold-distillate, respectively. Note that γ_T of Eq. (24) can be approximated as a pseudo-constant along the membrane surfaces because the incidence rate increases with the equilibrium temperature at interfaces. Then, the effective transmembrane heat flux $\boldsymbol{q}_m$ can be represented as a superposition of the conductive and convective heat fluxes, denoted as $\boldsymbol{q}_{m,v}$ and $\boldsymbol{q}_{m,d}$, respectively:

$$\boldsymbol{q}_m = \boldsymbol{q}_{m,d} + \boldsymbol{q}_{m,v} \tag{25}$$

The conductive heat equation is of a type of Fourier's equation that the heat flux is proportional to the negative temperature gradient

$$\boldsymbol{q}_{m,d} = -\bar{\kappa}_m \nabla T \tag{26}$$

where $\bar{\kappa}_m$ is the effective thermal conductivity of the membrane consisting of solid regions of a specific material and void regions often filled with confined gas such as air. We used Maxwell's relationship to have the effective thermal conductivity[74] of the porous membrane

$$\bar{\kappa}_m = \kappa_g \frac{1 + 2\beta_k(1-\epsilon)}{1 - \beta_k(1-\epsilon)} \tag{27}$$

$$\beta_k = \frac{\kappa_s - \kappa_g}{\kappa_s + 2\kappa_g} \tag{28}$$

where ϵ is the membrane porosity, κ_s and κ_g are thermal conductivities of the solid membrane part and air within pores, respectively[73, 76]. (See Appendix (Appendix A.2) for details of κ_g as a function of temperature T.) Thus, the convective heat flux is written in a mathematical form of thermal conduction proportional to the temperature gradient:

$$\boldsymbol{q}_{m,v} = \Delta H_v \cdot \boldsymbol{J}_w = -\epsilon \kappa_e \nabla T \tag{29}$$

where ΔH_v is the evaporation enthalpy at the feed-membrane interface and $\boldsymbol{J}_w$ is the diffusive flux of vapor through membrane pores. In Eq. (29), the evaporation enthalpy carried by the convective vapor flux $\boldsymbol{J}_w$ is estimated as a function of the interfacial temperature:

$$\Delta H_v = l_h(T) + c_{pw}T = l_0 + \left(c_{pw} - l_1\right)T = 3170.83 + 1.7489\,T \ \ [\mathrm{kJ/kg}] \tag{30}$$

where $l_h(T) = l_0 - l_1 T$ ($l_0 = 57.075\,\mathrm{kJ/mol} = 3170.83\,\mathrm{kJ/kg}$ and $l_1 = 4.3856 \times 10^{-2}\,\mathrm{kJ/mol \cdot K} = 2.4364\,\mathrm{kJ/kg \cdot K}$) is an expression of the latent heat of water[6] and $c_{pw} = 7.5336 \times 10^{-2}\,\mathrm{kJ/mol \cdot } K = 4.1853\,\mathrm{kJ/kg \cdot K}$ is the (pseudo-constant) specific heat of water at constant pressure. Note that in Eq. (29), the convective heat transfer is rewritten as an equivalent form to the conductive transfer, and therefore the equivalent conductivity is

$$\kappa_e = \frac{4}{3}\frac{d_p \gamma_T}{\tau} \cdot \Delta H_v(T) \tag{31}$$

For a fast shell flow, the temperature variation along the tangential direction is small, and the evaporation enthalpy ΔH_v can be approximated at the hot-feed temperature T_h [K], i.e., $\Delta H_v \simeq \Delta H_v(T_h)$. Superposition of Eqs. (26) and (29) into Eq. (25) gives an equivalent transmembrane heat flux in a conductive form of Fourier's law:

$$\boldsymbol{q}_m = -K_m \nabla T \tag{32}$$

where$K_m = \bar{\kappa}_m + \epsilon\kappa_e$ is the effective thermal conductivity of the membrane of porosity ϵ.

2.3.3. Boundary conditions: theoretical

In the DCMD process, membranes play the role of porous barriers, in which water evaporates on the feed-side of the membrane surface and condenses on the distillate-side of the membrane. The coupled heat-mass transfer phenomena on the membrane surfaces can be appropriately described using proper boundary conditions. Because heat is neither generated nor annihilated within the membrane, heat fluxes at each side of an interface should be continuous such that

$$-\kappa_w \nabla T \cdot \boldsymbol{n} = -K_m \nabla T \cdot \boldsymbol{n} \quad \text{for both membrane surfaces} \tag{33}$$

where κ_w is the thermal conductivity of (liquid) water of either the feed or distillate side, and $\boldsymbol{n}$ is a unit normal vector perpendicular to the membrane surfaces. For hollow fiber membranes, $\boldsymbol{n}$ is equal to the radial direction from the center of each fiber. Although Eq. (33) is valid for both feed and distillate sides of membrane surfaces, an integral form of the boundary condition expresses the thermal energy conservation for hollow fiber membranes:

$$Q_m = \int_{\text{lumen}} (-)\, \kappa_w \nabla T \cdot \boldsymbol{n} \mathrm{d}A = \int_{\text{lumen}} (-)\, K_m \nabla T \cdot \boldsymbol{n} \mathrm{d}A = \int_{\text{shell}} (-)\, K_m \nabla T \cdot \boldsymbol{n} \mathrm{d}A = \int_{\text{shell}} (-)\, \kappa_w \nabla T \cdot \boldsymbol{n} \mathrm{d}A \tag{34}$$

where Q_m denotes the total thermal energy transferred across the porous membrane in a steady state and the integrals are over either lumen or shell surfaces, touching adjacent fiber surfaces. In simulations, Eq. (33) is used as a boundary condition on the membrane surface to calculate spatial profiles of temperature T and its gradient ∇T within the preset computational domain; and Eq. (34) is used to calculate the transmembrane heat fluxes on the lumen and shell sides of each fiber. We calculate the distillate flux as a surface integral of heat flux (of Eq. (29)):

$$\langle J_w \rangle = -\frac{1}{A} \int \frac{\epsilon \kappa_e}{\Delta H_v} \nabla T \cdot \boldsymbol{n} \mathrm{d}A \tag{35}$$

which is compared with experimental observation in the next section.

2.4. Model Verification

2.4.1. Simulation details

Fig. 12 shows steady-state simulation results, obtained by using `chtMultiRegionSimpleFoam`, a steady-state solver of OpenFOAM, developed for buoyant, turbulent fluid flow and solid heat conduction with conjugate heat transfer between solid and fluid regions. The nearest neighbor list of this $n_l = 3$ case is discussed in Appendix (Appendix A.3). To include the turbulence effects, we use the $k-\epsilon$ model with default parameter settings. We used boundary conditions in Table 7 and temperature-dependent parameters in Table 8 in a temperature range from 17.5°C to 80°C. Fig. 12(a) shows a 3D semi-transparent color plot of temperature in the entire computational domain along the x-axis of the same direction of the lumen flow. On the other hand, the hot feed flow in the shell region enters the module at $x = l_f$ and exits at $x = 0$. Therefore, the inlet and outlet sides of the lumen and shell regions are switched in the counter-current flow system. Fig. 12(b) shows the internal temperature distribution on the normal cut surface to $-\boldsymbol{y}$ direction. Similar to Fig. 9(c) of a shorter membrane, temperature profiles in the lumen and shell regions are clearly separated. T_0 to T_4 indicate longitudinal temperature profile along the x-axis passing through a specific point in $y-z$ plane. The passing points are the lumen center located at $(0,0)$ for T_0, the inner-fiber surface at $\left(0, R_{fi}\right)$ for T_1, the outer-fiber surface at $\left(0, R_{fo}\right)$ for T_2, the midpoint between HF-0 and HF–5 for T_3, and another midpoint of triangle consisting of centers of three hollow fibers: HF-0, HF-4 and HF-5 for T_4. It is visually observed that the temperature variation in each of the lumen and shell regions are negligible in the longitudinal direction, i.e., $T_0 \simeq T_1$ and $T_2 \simeq T_3 \simeq T_4$. However, T_0 and T_1 rapidly increase from the inlet temperature of 290.65 K (17.5°C) within the first 10 mm of the lumen inlet and stay around 310 K (36.85 °C). On the other hand, the inlet temperature of the shell region is maintained at the feed temperature of 323.15K (50.0 °C) within the shell-inlet zone of 50 mm (from $x = 50$ mm to 100 mm). Afterward, T_k for $k \geq 1$ gradually decreases to 320 K (46.85 °C). Therefore, the transmembrane temperature difference is at least 10 K (=320 K - 310 K), of which region is marked yellow, across the membrane in the longitudinal direction. The rapid variations of the lumen temperature is ascribed to its slower fluid velocity and smaller cross-sectional area in comparison to those of the shell region.

Figs. 12(d) and (e) show the temperature T and the longitudinal velocity u_x along the z-axis passing through $(0,0)$, as shown as a white upward arrow in Fig. 12(b). From the bottom to the top of the containing vessel along the z-axis, T and u_s are measured at each region of shell, fiber, and lumen are alternatively along the z-axis. Two yellow

bars in Figs. 12(d) and (e) indicate fiber regions. Fig. 12(d) confirms the distinct temperatures shown in Fig. 12(b), especially at $(0, 0)$. Across the thin porous membrane of thickness 0.11 mm, temperature rapidly increases from the cold lumen to the hot shell regions. Fig. 12(e) shows the u_x profile along the same z-direction passing through $(0, 0)$. The lumen velocity has positive values as it flows in the $+\boldsymbol{x}$ direction with its inlet velocity of 0.8 m/s. The negative value of u_x indicates the counter-current shell flow in the $-\boldsymbol{x}$ direction. As the shell region has a bigger cross-sectional area, the velocity profile gradually changes along the z-direction. In the lumen side, u_x shows a non-Hagen–Poiseuille type flow pattern, which may be due to the temperature dependent viscosity and a degree of turbulence. In both Figs. 12(d) and (e), the temperature and flow patterns show periodic trends in the z-direction.

2.4.2. Experimental verification

We verify our new approaches of the unit-cell meshing technique and the equivalent conduction analogy by comparing our CFD simulation results with experimental work by Wang et el.[73], who used hydrophobic PVDF hollow fiber membrane for fresh water production. Detailed experimental conditions and parameters are listed in Table 8. For water properties, temperature-dependent c_p and μ are used at the common distillate temperature of $T_d = 17.5\,^{\circ}\mathrm{C}$ and also various hot feed temperatures from 30°C to 80°C. The conductivity of the porous membrane (marginally) changes from 43.24 mW/m K to 46.09 mW/m K for the feed temperature range, because the thermal conductivity of confined air in pore spaces varies with temperature. Specific correlation of the thermal conductivity of humid air can be found in Appendix Appendix A.2. The strong dependence of $\epsilon\kappa_e$ on temperature primarily stems from the non-linear variation of water vapor pressure P_v with T through the incidence rate per temperature, γ_T of Eq. (24). The experimental data of the distillate flux were extracted from the original reference[73] and compared with OpenFOAM CFD results in Fig. 13. Throughout the entire temperature range, the CFD results are in an excellent agreement with the experimental observations except the highest feed temperature of 80°C, where $\bar{\kappa}_m$ starts decreasing with respect to temperature. Overall, the monotonous increase in the distillate flux with respect to the feed temperature is captured by the OpenFOAM simulation using the equivalent thermal conductivity of Eq. (31).

3. Concluding Remarks

Hollow fibers are often preferred in industrial applications due to higher packing density. Still, their predictive modeling is challenging due to the complex geometry formed within many fibers. A computational meshing task drastically increases with the number of fibers and is generally challenging to parallelize using conventional meshing tools. This paper developed a hybrid meshing technique by creating a unit-cell mesh consisting of the lumen, fiber, and shell regions and packing many duplicated unit-cells after geometrical transformation. This method's unique advantage is that one can predetermine mesh quality within the unit-cell of the smallest mesh unit and pack fibers layer by layer in the overall hexagonal grid. Changing grid size for consistent GCI analysis is straightforward and easily applied to other applications using cylindrical membranes. Sample CFD simulations using the ideal hexagonal packing imply that the close-packing of fibers will create azimuthal variations of flow velocities and shear stresses around the fibers' outer surfaces, which will also influence the overall performance of HF membrane modules. On the other hand, coupled CFD simulations for the momentum and heat transfer in a sparse fiber packing indicate that the significant temperature variation occurs in the x-direction, parallel to the lumen and shell flow directions. Theoretically, the meshing time for an arbitrary number of fibers is equal to that for a single fiber, and therefore the meshing time is independent of the number of fibers. Additional variations include randomly aligned fibers, touching neighbors, and slightly curved, which will be in our future research topics.

Acknowledgment

This research was supported by a grant from the Endowment Project of "Development of Seawater Desalination-Cooing Plant using Unutilized Energy (2/5)" funded by the Korea Research Institute of Ships and Ocean Engineering (PES3520).

n_l	n_f		n_l	n_f		n_l	n_f
0	1		10	331		20	1261
1	7		11	397		30	2791
2	19		12	469		40	4921
3	37		13	547		41	5167
4	61		14	631		50	7651
5	91		15	721		60	**10981**
6	**127**		16	817		70	14911
7	169		17	919		80	19441
8	217		18	**1027**		90	24571
9	271		19	1141		100	30301

Table 1. The number of total fiber n_f versus the number of layers n_l. For n_l the fiber index n_f ranges from 0 to $n_l - 1$.

n_s	i	S_i	j	S_j	n_s	i	S_i	j	S_j
1	0	0	1	3	...				
2	0	1	2	4	895	294	4	295	1
3	0	2	3	5	896	295	4	296	1
4	0	3	4	0	897	296	4	297	1
5	0	4	5	1	898	297	4	298	1
6	0	5	6	2	899	298	4	299	1
7	1	0	7	3	900	299	4	300	1
8	1	1	8	4	901	300	4	301	1
9	1	2	2	5	902	301	5	302	2
10	1	4	6	1	903	302	5	303	2
11	1	5	18	2	904	303	5	304	2
12	2	0	8	3	905	304	5	305	2
13	2	1	9	4	906	305	5	306	2
14	2	2	10	5	907	306	5	307	2
15	2	3	3	0	908	307	5	308	2
16	3	1	10	4	909	308	5	309	2
17	3	2	11	5	910	309	5	310	2
18	3	3	12	0	911	310	5	311	2
19	3	4	4	1	912	311	0	312	3
20	4	2	12	5	913	312	0	313	3
21	4	3	13	0	914	313	0	314	3
22	4	4	14	1	915	314	0	315	3
23	4	5	5	2	916	315	0	316	3
24	5	0	6	3	917	316	0	317	3
25	5	3	14	0	918	317	0	318	3
26	5	4	15	1	919	318	0	319	3
27	5	5	16	2	920	319	0	320	3
28	6	0	18	3	921	320	0	321	3
29	6	4	16	1	922	321	1	322	4
30	6	5	17	2	923	322	1	323	4
31	7	0	19	3	924	323	1	324	4
32	7	1	20	4	925	324	1	325	4
33	7	2	8	5	926	325	1	326	4
34	7	4	18	1	927	326	1	327	4
35	7	5	36	2	928	327	1	328	4
36	8	0	20	3	929	328	1	329	4
...					930	329	1	330	4

Table 2. A partial interfacial pair list of nearest neighbor cells with $n_l = 10$ and $n_f = 331$. Here, i and j are fiber indexes from 0 to 330, and S_i and S_j are the surface indexes from 0 to 5 of i and j, respectively. The left and right columns indicate the first and last 36 pairs. A higher fiber index j does not have the full surface index from 0 to 5 because they are located at the edges.

θ_{jk} [degree]	θ_{jk} [rad]	I_θ	S_j	S_k
30°	$1 \cdot \frac{\pi}{6}$	1	4	1
90°	$3 \cdot \frac{\pi}{6}$	3	5	2
150°	$5 \cdot \frac{\pi}{6}$	5	0	3
210°	$7 \cdot \frac{\pi}{6}$	7	1	4
270°	$9 \cdot \frac{\pi}{6}$	9	2	5
330°	$11 \cdot \frac{\pi}{6}$	11	3	0

Table 3. Possible matching surface indexes S_j and S_k as a function of the angle θ_{jk}, where $I_\theta = \theta_{jk}$ [rad] / $\left(\frac{\pi}{6}\right)$.

	Radial divisions	Azimuthal division	Run time [sec]
Intel i-core 9	30	32	6.97
AMD Ryzen	30	32	4.98

Table 4. Run time comparison between two processors for the unit-cell mesh generation. Intel(R) Core(TM) i9-10920X CPU @3.50 GHz and AMD Ryzen Threadripper 3960X processor are used to generate a unit mesh of 30 and 32 divisions in the radial and the longitudinal directions, respectively. For each case of the CPUs, a unit-cell mesh was generated within a few second, which can be considered as the universal meshing time because this unit-cell mesh will be duplicated and relocated to generate the entire hollow fiber meshed within a cylindrical vessel. This mesh-assembling time is proportional to the total number of fibers, but does not required substantial run time.

	pressure	velocity
inlet	$\nabla p = 0$	$U = U_0\,(= 0.1\,\mathrm{m/s})$, a fixed value
outlet	$p = 0$	‘‘`PressureInletOutletVelocity`’’
surface	$\nabla p = 0$	$U = 0$

Table 5. Boundary conditions at the inlet ($z = 0$), outlet ($z = L_\mathrm{f}$), and outer surface ($r = d_\mathrm{fo}/2$) of a fiber, used for the flow simulations.

	$n_x = n_y$	n_r	$h \equiv 1/n_r$
1	12	8	0.1250
2	9	6	0.1667
3	6	4	0.2500

Table 6. Parameters used to calculate the GCI, where the grid length scale h is defined as the inverse of n_r, and their ratios of $r_{21} = h_2/h_1 = 1.333$ and $r_{32} = h_3/h_2 = 1.50$ are larger than the empirical minimum of 1.3.

region		pressure	velocity	temperature
lumen	inlet ($x = 0$)	$\nabla p = 0$	$\boldsymbol{U} = (0.8, 0, 0)$ m/s, a fixed value	17.5 °C
	outlet ($x = l_f$)	$p = 0$	‘‘`PressureInletOutletVelocity`’’	zero gradient
shell	inlet ($x = l_f$)	$p = 0$	$\boldsymbol{U} = (-1.6, 0, 0)$ m/s, a fixed value	50.0 °C
	outlet ($x = 0$)	$\nabla p = 0$	‘‘`PressureInletOutletVelocity`’’	zero gradient
	all walls	$\nabla p = 0$	$U = 0$	zero gradient

Table 7. Boundary conditions at the inlet ($z = 0$), outlet ($z = L_\mathrm{f}$), and outer surface ($r = d_\mathrm{fo}/2$) of a fiber, used for the flow simulations.

	Water			Membrane		Distillate Flux [LMH]	
T [°C]	c_p [J/kg · K]	η [mPa · s]	Pr [–]	$\bar{\kappa}_m$ [W/mK]	$\epsilon\kappa_e$ [W/m · K]	Exp. [73]	CFD
17.5	4184.0	1.0653	7.0440	0.04225	0.08335		
30.0	4179.0	0.7970	6.7696	0.04324	0.10960	2.603	2.798
40.0	4179.0	0.6539	6.5970	0.04420	0.14490	6.027	8.495
50.0	4181.0	0.5474	6.4487	0.04516	0.19141	11.507	11.898
60.0	4185.0	0.4656	6.3214	0.04609	0.25177	19.863	22.143
70.0	4190.0	0.4024	6.2101	0.04701	0.32895	30.548	29.596
80.0	4196.0	0.3537	6.1130	0.04609	0.42617	42.466	36.448

Table 8. Simulation parameters to reproduce Wang et el.'s experimental work work [73]. Other parameters include: for membrane, porosity $\epsilon = 0.738$**, pore diameter** $d_p = 0.16\,\mu$m**,** $c_p = 1.170$ [kJ/kg · K]**. Pr indicates Prandtle number of water defined as** $\text{Pr} = c_p\mu/\kappa$**, and LMH indicates** liter/hr · m^2**. The velocity and temperature of the lumen region are 0.8 m/s and 17.5°C and those of the shell region are -1.6 m/s and from 30–80°C, respectively. The negative sign of the shell velocity indicates that the shell and lumen velocities are in the opposite directions.**

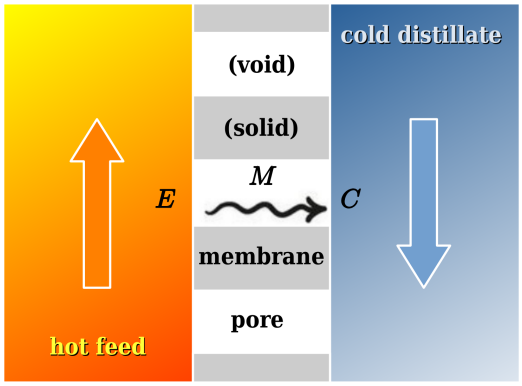


Figure 1. A schematic of counter-current direct contact membrane distillation where E, M, and C indicate evaporation, migration and condensation, respectively, of water molecules passing through the void pore spaces. Temperature range of the hot feed is 50°C to 80°C and the cold distillate stream is often maintained at 25°C or below. Heat transfer from the hot feed to the cold distillate streams consist of conductive transport through the solid part of the membrane and convective transport through the migrating water molecules.

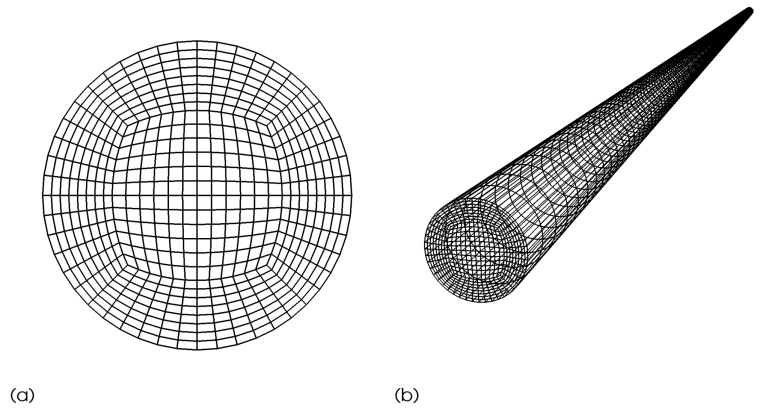


Figure 2. Lumen-zone meshing of (a) orthogonal and (b) perspective views, which shows $n_x = n_y = 12$ and $n_r = 8$.

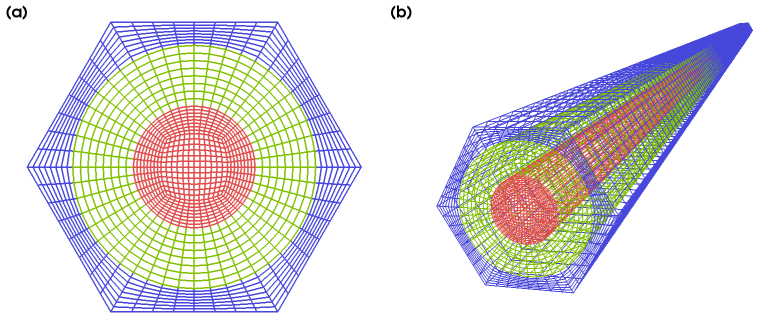


Figure 3. A unit mesh-cell made by encompassing the fiber using a hexagonal shell (marked in blue, having 6 side walls): (a) orthogonal and (b) perspective views. The lumen (red) zones are identical to those of Fig. 2. Membrane (green) zone is seamlessly connected to lumen (brown) and shell (blue) zones. The lumen zone has two sub-regions: the round cube at the center and the arc-shaped intermediate sub-region between the lumen core and inner fiber surfaces. The round cube zone of the lumen is divided by six in both x and y directions. All other sub-regions are radially divided by 8, including the lumen edge, fiber, and shell regions. Each of four arc-zones of the lumen is azimuthally divided by 8. Therefore, the total numbers of division in the radial and azimuthal directions are 30 and 32, respectively. The unit length in the longitudinal direction is set three times the inner radius of the hollow fiber.

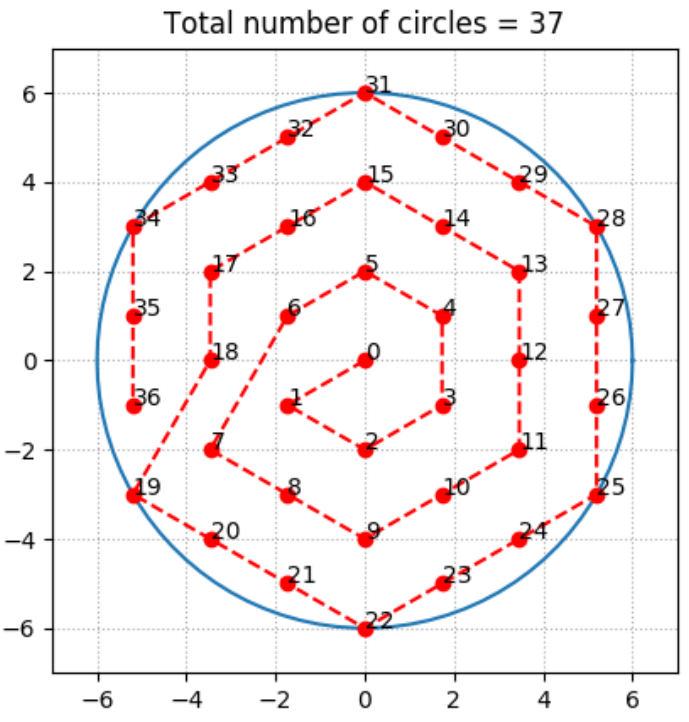
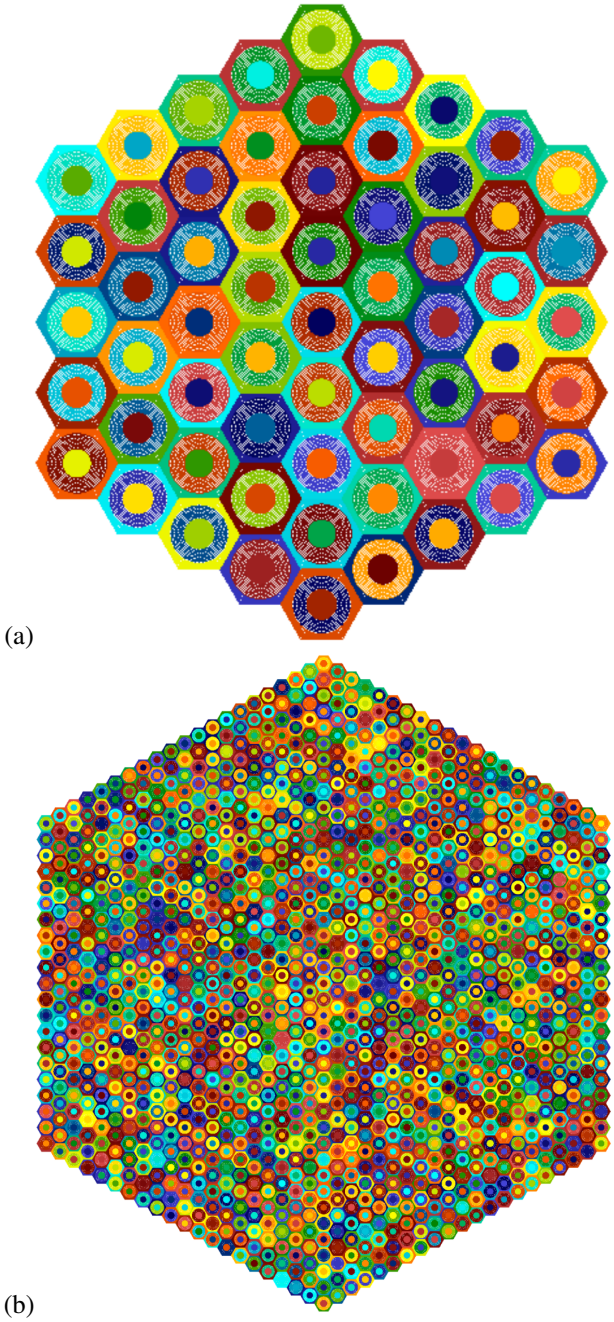


Figure 4. Hexagonally-layered sequential packing of $n_l = 3$ having 37 fibers.



(a)

(b)

Figure 5. Hexagonal packing of four layers (a) $n_l = 4$ **and** $n_f = 61$**, and (b)** $n_l = 20$ **and** $n_f = 1261$**.**

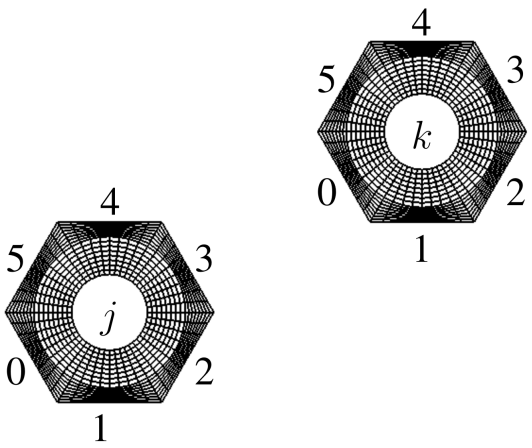


Figure 6. Boundary surfaces from 0 to 5 of two arbitrary hollow fibers of j and k. The centers of j and k can be visualized in the same way of Fig. 4.

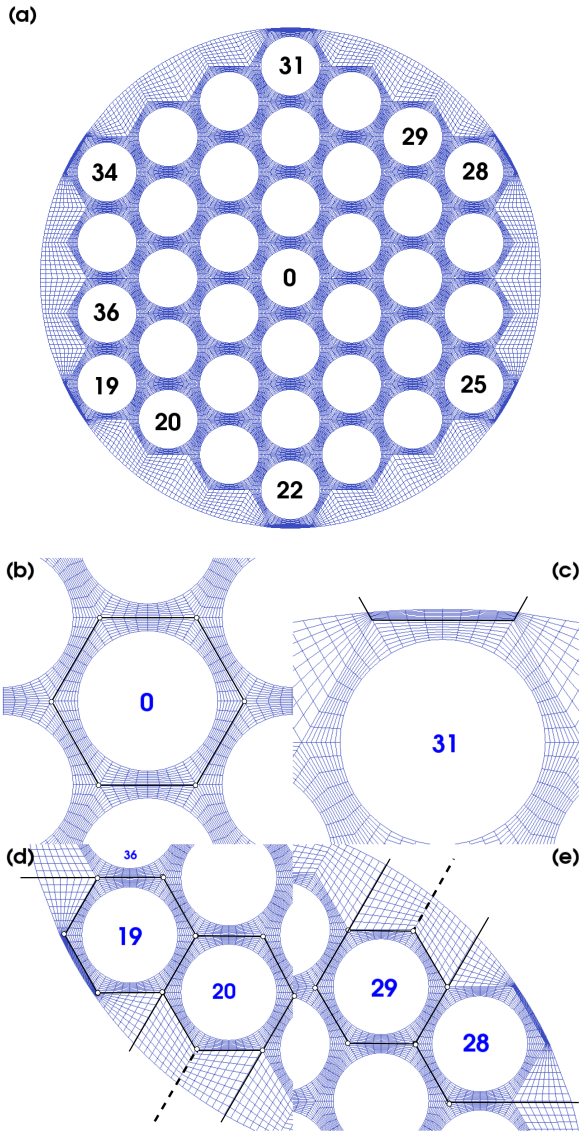


Figure 7. Mesh structure of hexagonal unit-cells and edge zones within a cylindrical vessel container: (a) overall, (b) central, (c) top, (d) bottom-left, and (e) top-right view-points.

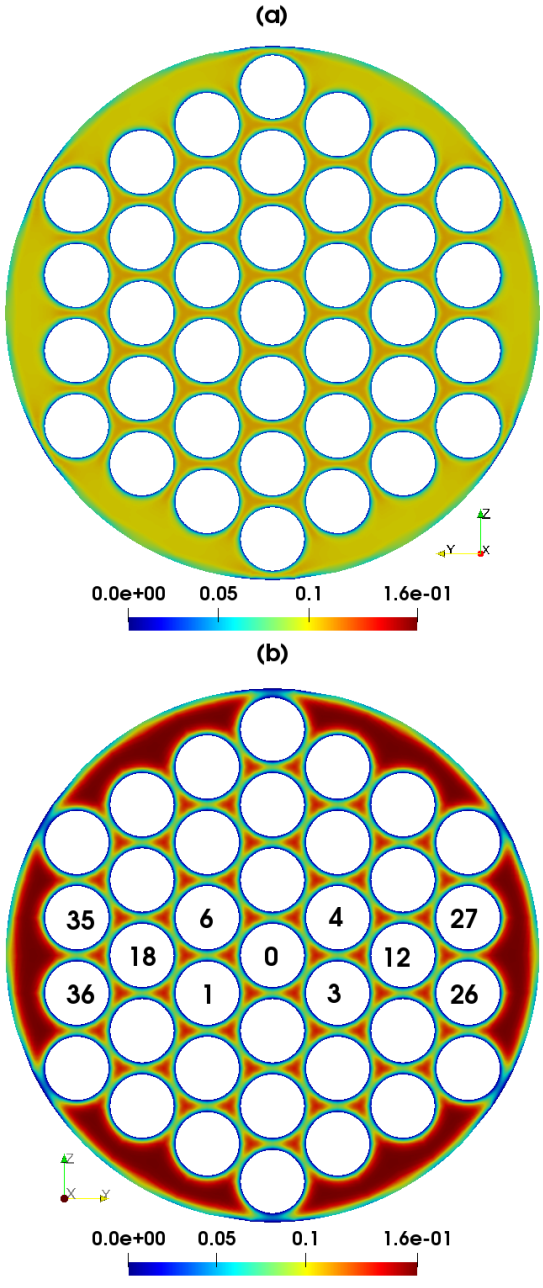


Figure 8. Color plots of flow speed U [m/s] at the (a) inlet (normal to $-\hat{x}$) and (b) outlet (normal to $+\hat{x}$) cross-section surfaces. The inlet and outlet speeds are scaled from zero to the maximum outlet speed.

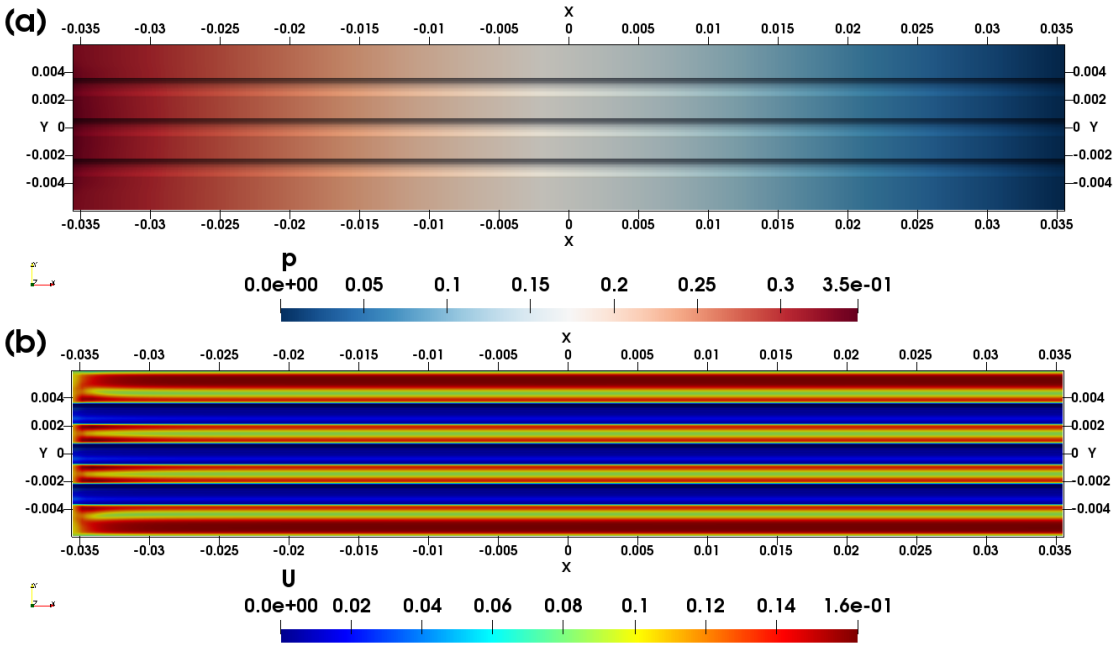


Figure 9. Profiles of (a) hydraulic pressure p divided by fluid mass density ρ, and (b) fluid speed along the longitudinal axis along the shell alignment.

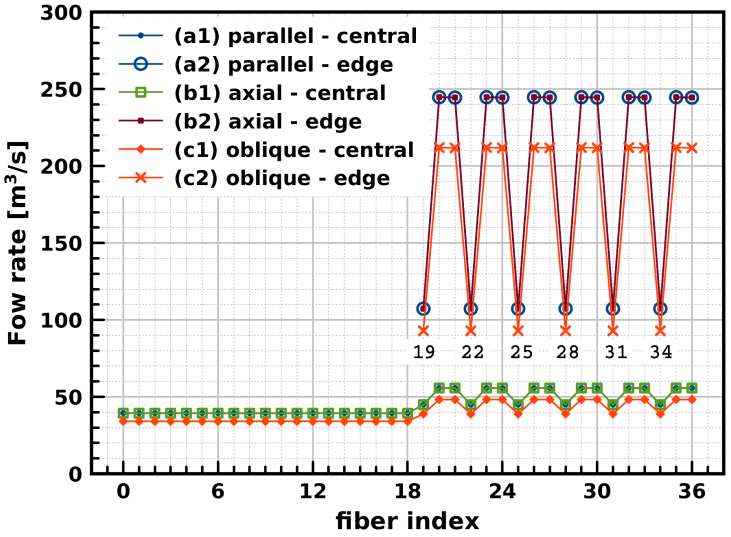


Figure 10. Flow rate passing around fibers located near (1) central and (2) edge zones of (a) parallel with v = (0.1, 0.0, 0.0) m/s**, (b) axial with** v = (0.086603, 0.035355, 0.035355) m/s**, and (c) oblique with** v = (0.1, 0.035355, 0.035355) m/s **inflow velocities. The central fiber zone indicates void spaces within each hexagonal cell, and the edge fiber zone means gap spaces between outermost fibers and the inner wall of the cylindrical vessel.**

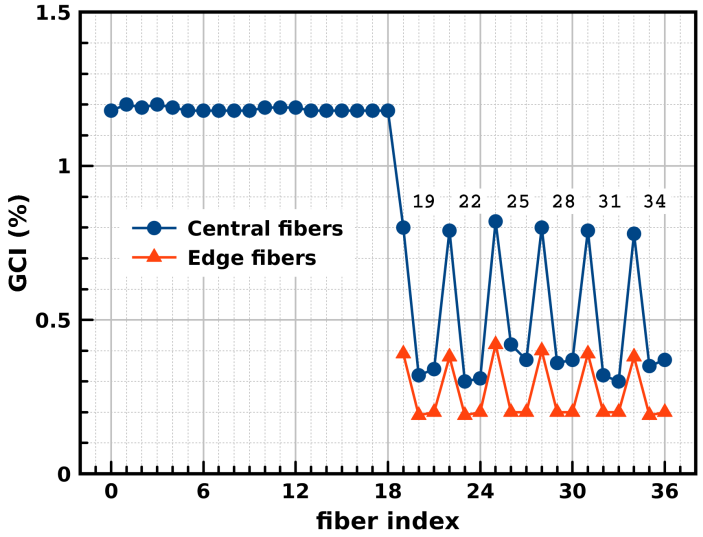


Figure 11. Grid convergence index of flow rates around fibers. The meaning of central and edge zone fibers are as same as those explained in Fig. 10.

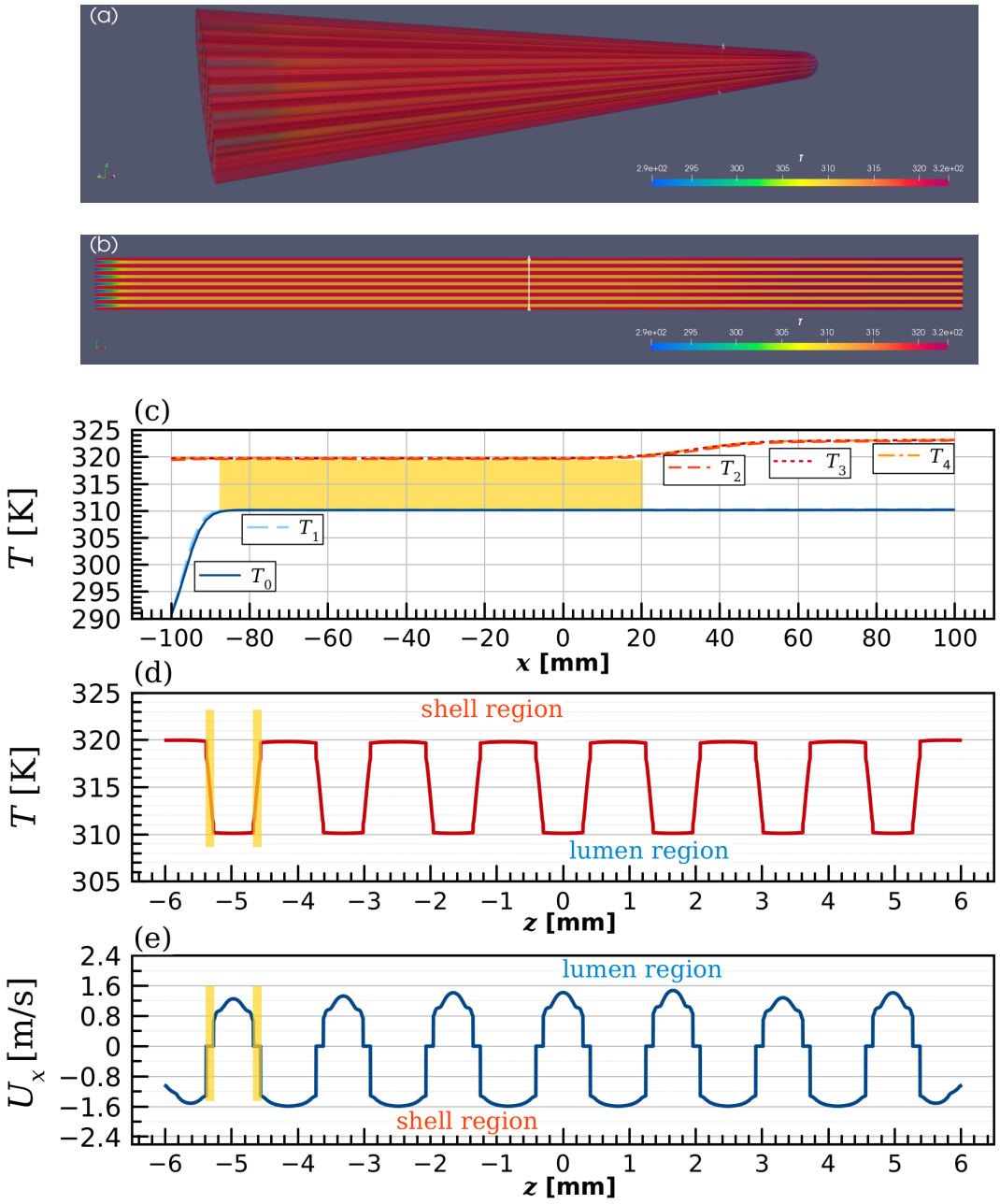


Figure 12. Simulation results of `chtMultiRegionSimpleFoam` (a) a 3D perspective, semi-transparent overview of temperature, (b) a clip view of temperature at a mid-plane normal to $-y$, (c) temperature of hollow fiber (of index 0, located at the center) along the $x-$axis (T_0 along the lumen center, T_1 on the inner lumen surface, T_2 on the fiber outer surface, T_3 along the midpoint between fiber 0 and fiber 5 (one above fiber 0), and T_4 along the center of a triangle made of fiber 0, 4, and 5, (d) temperature along the $z-$axis from the bottom to the top of the cylindrical vessel, and (e) $x-$directional fluid velocity of lumen and shell region, of which inlet velocities are +1.6 m/s and -0.8 m/s.

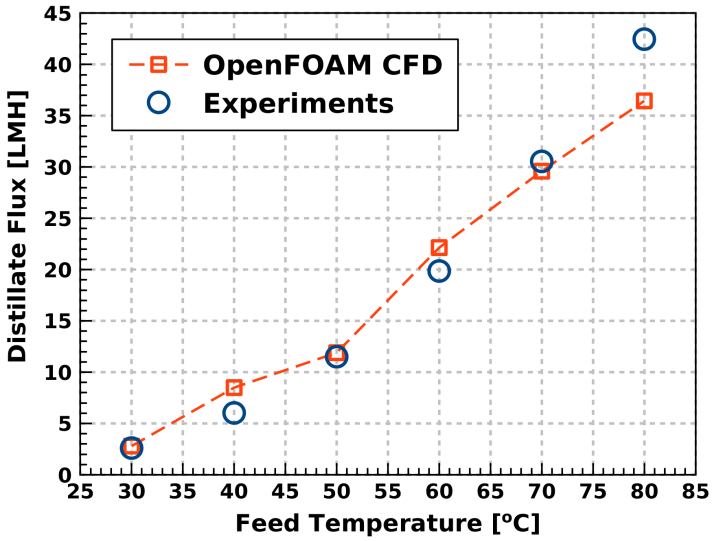


Figure 13. Comparison between the current CFD simulations for DCMD and experimental observations [73]. Standard deviations of the simulation results are 4.47% and 1.63% for the feed temperature of 30°C and 40°C, respectively, and maintained below 1.0% for higher temperatures. CFD results for the DCMD process are in good agreement with the experimental data, except the highest feed temperature of 80°C.

Appendix A. Supplementary calculations and verification

Appendix A.1. Tetrahedron meshes

Fig. A.1 shows three tetrahedron mesh structures, generated using the open-source mesh software NETGEN [54–56]. Meshes of Fig. A.1(a) and (b) were generated using the default options indicated as "moderate" and "very fine", respectively, of the software, and (c) using customized mesh parameters for the finest mesh. In NETGEN, the maximum mesh-size parameter (s) controls the size of the largest tetrahedron cell. By default, s is often set unnecessarily large for a fast, flexible meshing process. A smaller s value requires more iterations for Delaunay triangulation and further geometrical optimizations. In this case, the s value effectively controls the mesh-grid sizes, if it is of the same order of an average gap-distance between two fibers. The values of s used for Fig. A.1(a), (b), and (c) are 35.0, 0.35 and 0.035 mm, respectively. Interstitial meshes of Fig. A.1(a) connects outer surfaces of neighboring fibers, which is clearly shown in the zoomed-in version of Fig. A.1(d). The absence of interstitial vertices is due to the value of s for (a) and (d), being much larger than the fiber diameter of 0.7 mm. Fig. A.1(b) shows a denser mesh structure, generated using $s = 0.35$ mm, than that of Fig. A.1(a) and represents a few mesh layers between two adjacent fibers in its zoomed-in version of Fig. A.1(e). Because outer surfaces of fibers are often treated as no-slip boundary walls with zero fluid velocity, much finer mesh structures—than those of Figs. A.1(a) and (b)—are necessary to perform CFD simulations for stiff shear gradients on the fiber surfaces. It is, therefore, appropriate to set s smaller than a distance representing the inter-fiber gap spaces. Fig. A.1(c) uses $s = 0.035$ mm, which is 10 percent of the fiber diameter, to generate a finer mesh, as also shown in Fig. A.1(d). As NETGEN generates unstructured/unsupervised tetrahedron meshes, the fine mesh structure of Fig. A.1(f) is neither ideally symmetrical nor regularly grided in comparison to those in Fig. 7. Although tetrahedron meshes can be efficiently used in particle hydrodynamics simulation to interpolate the flow velocities, vorticities, and strain rates [77], much more tetrahedron cells are required to cover 3D domain spaces to avoid skewed elements of triangular surfaces and tetrahedron cells. The unit-cell mesh method developed in this study can be, in principle, applied to a tetrahedron-based mesh generation. There are, however, practical disadvantages of using tetrahedron-based unit-cells. Usually, tetrahedron cells grow from inner surfaces and merge with neighboring cells grown from the opposite directions. Hence, meshes in the central zone are less structured than those located near the surfaces. If two surfaces of different objects are merged, then the interfacial mesh does not require any boundary conditions, but simply transfer data from one region to the other. Due to the unstructured geometry of the surface meshes, interfacial vertices and edges are neither perfectly matched nor ideally conformed. Each mesh in Fig. A.1(a)-(f) is generated simultaneously without a post-process of merging neighboring unit-cell meshes.

If the number of fibers increases, or a finer mesh is generated, the computational load for the iterative Delaunay triangulation often increases non-linearly. Using a desktop workstation under Linux OS (Ubuntu LTS 18.08), consisting of Intel Core i9-10920X CPU (12 physical cores) @ 3.50 GHz with 128 GB of random access memory, meshing process for Fig. A.1(a) takes 92.68 seconds to generate 217,231 points and 767,877 elements. Similarly, meshing for Fig. A.1 (b) takes 98.24 seconds to form 229,957 points and 831,875 elements. These measurements indicate that values of s larger than the minimum representative length scale (i.e., the average inter-fiber gap distance) provide assimilated mesh structures, characterized by the numbers of vertices and edges. The difference between run times of Fig. A.1(a) and (b) is therefore only 6 percent. On the other hand, Fig. A.1(c) requires an enormously longer run time of 28,130.4 seconds (7h 48m 50s) to generate the finest meshes zoomed in Fig. A.1(e), consisting of 33,225,748 points and 188,767,067 elements. But, only a few mesh layers are formed between neighboring fibers. As compared to the tetrahedron mesh generation, the run time of the present meshing process is negligible, because once the unit-cell mesh is formed, AMI-based merging and data interpolation are conducted during OpenFOAM simulations. Based on the fundamental heterogeneity between the conventional and new meshing approaches, their direct comparison is not meaningful, as the `blockMesh` generation is almost immediate.

Appendix A.2. Thermodynamic properties of water

To calculate the thermal conductivity of a porous membrane consisting of the solid part and void region filled with air, we use the following representation

$$\kappa_g(T) = \sum_{i=0}^{4} K_i T_C^i \ [\mathrm{W/m \cdot K}] \tag{A.1}$$

where T_C is temperature [°C] and the coefficients are $K_0 = 2.40073953 \times 10^{-2}$, $K_1 = 7.278410162 \times 10^{-5}$, $K_2 = -1.788037411 \times 10^{-20}$, $K_3 = -1.351703529 \times 10^{-9}$, and $K_4 = -3.322412767 \times 10^{-11}$.

Appendix A.3. Additional information for mesh generation

Table A.1 shows the complete list of nearest neighbors between fiber i and j for $n_l = 3$ case. A shell region has six lateral sides, and each side is share by one of six nearest neighbors. On the average, the total number of shared interfaces is equal to $3n_f$, where n_f of n_l layers is determined using Eq. (3). In the outer most layer, i.e., $n_l{}^{\text{th}}$ layer, there are six shells located at corner positions and others at lateral sides. The corner and lateral shells has three and four nearest neighbors, respectively. Therefore, the total number of shell surfaces not shared with neighbors is $6\,(3 \times 1 + 4 \times (n_l - 1))$. Then the total number of touching shell-surfaces is

$$N_{nn} = 3n_f - \tfrac{1}{2}6\,(3 + 2\,(n_l - 1)) = 3n_l\,(3n_l + 1) \tag{A.2}$$

which indicates the 90 nearest-neighbor pairs for $n_l = 3$, as shown in Table (A.1).

n_s	i	S_i	j	S_j	n_s	i	S_i	j	S_j	n_s	i	S_i	j	S_j
1	0	0	1	3	31	7	0	19	3	61	15	3	30	0
2	0	1	2	4	32	7	1	20	4	62	15	4	31	1
3	0	2	3	5	33	7	2	8	5	63	15	5	32	2
4	0	3	4	0	34	7	4	18	1	64	16	0	17	3
5	0	4	5	1	35	7	5	36	2	65	16	4	32	1
6	0	5	6	2	36	8	0	20	3	66	16	5	33	2
7	1	0	7	3	37	8	1	21	4	67	17	0	35	3
8	1	1	8	4	38	8	2	9	5	68	17	1	18	4
9	1	2	2	5	39	9	0	21	3	69	17	4	33	1
10	1	4	6	1	40	9	1	22	4	70	17	5	34	2
11	1	5	18	2	41	9	2	23	5	71	18	0	36	3
12	2	0	8	3	42	9	3	10	0	72	18	5	35	2
13	2	1	9	4	43	10	1	23	4	73	19	2	20	5
14	2	2	10	5	44	10	2	24	5	74	19	4	36	1
15	2	3	3	0	45	10	3	11	0	75	20	2	21	5
16	3	1	10	4	46	11	1	24	4	76	21	2	22	5
17	3	2	11	5	47	11	2	25	5	77	22	3	23	0
18	3	3	12	0	48	11	3	26	0	78	23	3	24	0
19	3	4	4	1	49	11	4	12	1	79	24	3	25	0
20	4	2	12	5	50	12	2	26	5	80	25	4	26	1
21	4	3	13	0	51	12	3	27	0	81	26	4	27	1
22	4	4	14	1	52	12	4	13	1	82	27	4	28	1
23	4	5	5	2	53	13	2	27	5	83	28	5	29	2
24	5	0	6	3	54	13	3	28	0	84	29	5	30	2
25	5	3	14	0	55	13	4	29	1	85	30	5	31	2
26	5	4	15	1	56	13	5	14	2	86	31	0	32	3
27	5	5	16	2	57	14	3	29	0	87	32	0	33	3
28	6	0	18	3	58	14	4	30	1	88	33	0	34	3
29	6	4	16	1	59	14	5	15	2	89	34	1	35	4
30	6	5	17	2	60	15	0	16	3	90	35	1	36	4

Table A.1. A complete interfacial pair list of nearest-neighbor cells with $n_l = 3$ and $n_f = 37$. Here, i and j are fiber indexes from 0 to 37, and S_i and S_j are the surface indexes from 0 to 5 of i and j, respectively. A higher fiber index j does not have the full surface index from 0 to 5 because they are located at the edges.

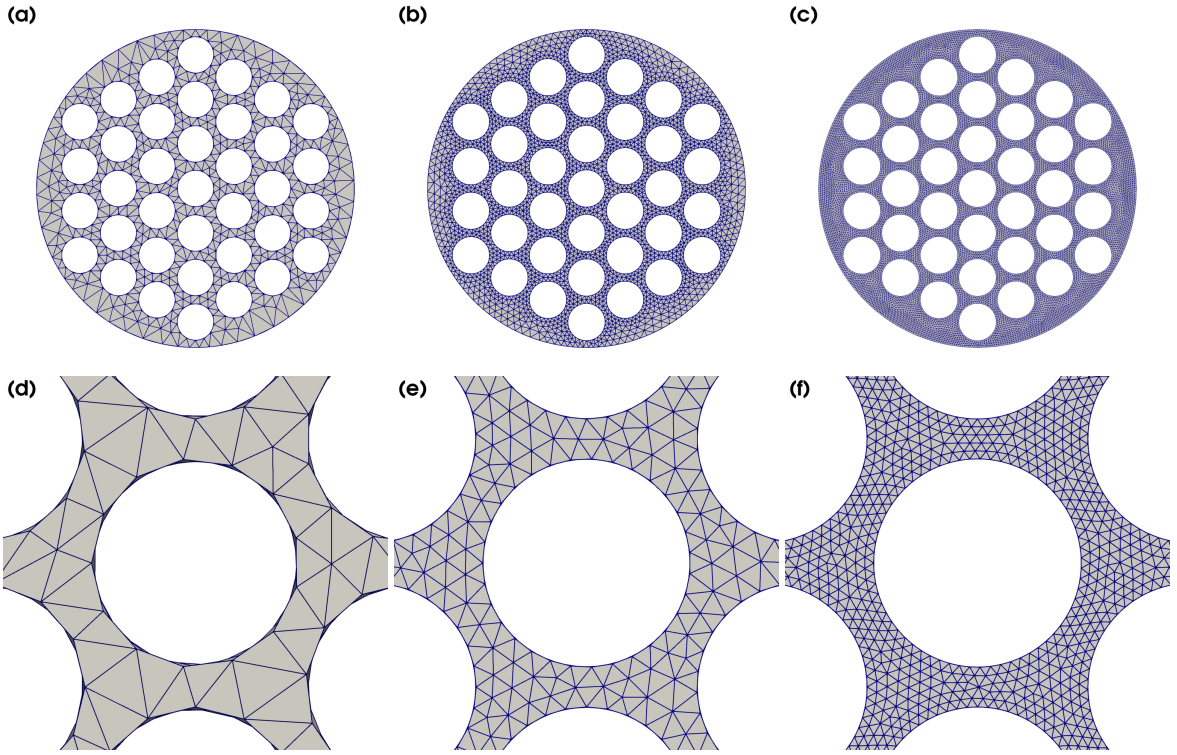


Figure A.1. Tetrahedron mesh structures generated using NETGEN: (a) moderate ($s = 35$, $g = 0.3$), (b) very fine ($s = 0.35$, $g = 0.1$), and (c) custom meshing($s = 0.035$, $g = 0.1$) options and (d)–(f) are zoomed images of the central fiber of (a)–(c), respectively, where s [mm] and g are NETGEN options of maximum mesh-size and mesh-size grading.

References

[1] A. Deshmukh, M. Elimelech, Un derstanding the impact of membrane properties and transport phenomena on the energetic performance of membrane distillation desalination, Journal of Membrane Science 539 (2017) 458–474. doi:10.1016/j.memsci.2017.05.017.

[2] A. Deshmukh, C. Boo, V. Karanikola, S. Lin, A. P. Straub, T. Tong, D. M. Warsinger, M. Elimelech, Membrane distillation at the water-energy nexus: Limits, opportunities, and challenges, Energy & Environmental Science 11 (5) (2018) 1177–1196, 00014. doi:10.1039/C8EE00291F.

[3] M. Khayet, T. Matsuura, Membrane Distillation: Principles and Applications, Elsevier, New York, 2011.

[4] A. Alkhudhiri, N. Darwish, N. Hilal, Membrane distillation: A comprehensive review, Desalination 287 (2012) 2–18. doi:10.1016/j.desal.2011.08.027.

[5] E. Drioli, A. Ali, F. Macedonio, Membrane distillation: Recent developments and perspectives, Desalination 356 (2015) 56–84.

[6] A. S. Kim, A two-interface transport model with pore-size distribution for predicting the performance of direct contact membrane distillation (DCMD), Journal of Membrane Science 428 (2013) 410–424. doi:10.1016/j.memsci.2012.10.054.

[7] A. S. Kim, Cylindrical cell model for direct contact membrane distillation (DCMD) of densely packed hollow fibers, Journal of Membrane Science 455 (2014) 168–186. doi:10.1016/j.memsci.2013.12.067.

[8] G. Guan, X. Yang, R. Wang, R. Field, A. G. Fane, Evaluation of hollow fiber-based direct contact and vacuum membrane distillation systems using aspen process simulation, Journal of Membrane Science 464 (2014) 127–139. doi:10.1016/j.memsci.2014.03.054.

[9] S. J. Ki, H.-J. Kim, A. S. Kim, Big data analysis of hollow fiber direct contact membrane distillation (HFDCMD) for simulation-based empirical analysis, Desalination 355 (2015) 56–67. doi:10.1016/j.desal.2014.10.008.

[10] E.-J. Lee, A. K. An, T. He, Y. C. Woo, H. K. Shon, Electrospun nanofiber membranes incorporating fluorosilane-coated TiO2 nanocomposite for direct contact membrane distillation, Journal of Membrane Science 520 (2016) 145–154. doi:10.1016/j.memsci.2016.07.019.

[11] A. S. Kim, H.-S. Lee, D.-S. Moon, H.-J. Kim, Self-adjusting, combined diffusion in direct contact and vacuum membrane distillation, Journal of Membrane Science 543 (2017) 255–268. doi:10.1016/j.memsci.2017.08.059.

[12] A. S. Alsaadi, L. Francis, G. L. Amy, N. Ghaffour, Experimental and theoretical analyses of temperature polarization effect in vacuum membrane distillation, Journal of Membrane Science 471 (2014) 138 – 148. doi:http://dx.doi.org/10.1016/j.memsci.2014.08.005.

[13] M. A. E. R. Abu-Zeid, Y. Zhang, H. Dong, L. Zhang, H. L. Chen, L. Hou, A comprehensive review of vacuum membrane distillation technique, Desalination 356 (2015) 1–14. doi:10.1016/j.desal.2014.10.033.

[14] A. S. Kim, H.-S. Lee, D.-S. Moon, H.-J. Kim, Statistical theory of vapor transport through hollow fiber membranes in vacuum membrane distillation: Effusion analogy, Desalination 410 (2017) 77–90, 00002. doi:10.1016/j.desal.2017.01.033.

[15] A. Criscuoli, E. Drioli, Vacuum membrane distillation for the treatment of coffee products, Separation and Purification Technology 209 (2019) 990–996. doi:10.1016/j.seppur.2018.09.058.

[16] S. Shukla, N. Benes, I. Vankelecom, J. Méricq, M. Belleville, N. Hengl, J. S. Marcano, Sweep gas membrane distillation in a membrane contactor with metallic hollow-fibers, Journal of Membrane Science 493 (2015) 167–178. doi:10.1016/j.memsci.2015.06.040.

[17] V. Karanikola, A. F. Corral, H. Jiang, A. E. Sáez, W. P. Ela, R. G. Arnold, Sweeping gas membrane distillation: Numerical simulation of mass and heat transfer in a hollow fiber membrane module, Journal of Membrane Science 483 (2015) 15–24. doi:10.1016/j.memsci.2015.02.010.

[18] S. E. Moore, S. D. Mirchandani, V. Karanikola, T. M. Nenoff, R. G. Arnold, A. E. Sáez, Process modeling for economic optimization of a solar driven sweeping gas membrane distillation desalination system, Desalination 437 (2018) 108–120. doi:10.1016/j.desal.2018.03.005.

[19] H. C. Duong, M. Duke, S. Gray, P. Cooper, L. D. Nghiem, Membrane scaling and prevention techniques during seawater desalination by air gap membrane distillation, Desalination 397 (2016) 92–100. doi:10.1016/j.desal.2016.06.025.

[20] Y. C. Woo, L. D. Tijing, M. J. Park, M. Yao, J.-S. Choi, S. Lee, S.-H. Kim, K.-J. An, H. K. Shon, Electrospun dual-layer nonwoven membrane for desalination by air gap membrane distillation, Desalination 403 (2017) 187–198. doi:10.1016/j.desal.2015.09.009.

[21] A. Alkhudhiri, N. Hilal, Air gap membrane distillation: A detailed study of high saline solution, Desalination 403 (2017) 179–186. doi:10.1016/j.desal.2016.07.046.

[22] L. Cheng, Y. Zhao, P. Li, W. Li, F. Wang, Comparative study of air gap and permeate gap membrane distillation using internal heat recovery hollow fiber membrane module, Desalination 426 (2018) 42–49. doi:10.1016/j.desal.2017.10.039.

[23] A. E. Khalifa, Water and air gap membrane distillation for water desalination – an experimental comparative study, Separation and Purification Technology 141 (2015) 276–284. doi:10.1016/j.seppur.2014.12.007.

[24] J. Swaminathan, H. W. Chung, D. M. Warsinger, F. A. AlMarzooqi, H. A. Arafat, J. H. L. V, Energy efficiency of permeate gap and novel conductive gap membrane distillation, Journal of Membrane Science 502 (2016) 171–178. doi:10.1016/j.memsci.2015.12.017.

[25] F. E. Ahmed, B. S. Lalia, R. Hashaikeh, Membrane-based detection of wetting phenomenon in direct contact membrane distillation, Journal of Membrane Science 535 (2017) 89–93. doi:10.1016/j.memsci.2017.04.035.

[26] Y. Chen, M. Tian, X. Li, Y. Wang, A. K. An, J. Fang, T. He, Anti-wetting behavior of negatively charged superhydrophobic PVDF membranes in direct contact membrane distillation of emulsified wastewaters, Journal of Membrane Science 535 (2017) 230–238. doi:10.1016/j.memsci.2017.04.040.

[27] A. S. Kim, H.-S. Lee, D. S. Moon, and H. J. Kim, Self-Adjusting, Combined Diffusion in Direct Contact and Vacuum Membrane Distillation, Journal of Membrane Science, 543 (2017), 255-268. doi:10.1016/j.memsci.2017.08.059.

[28] A. S. Kim, Cylindrical Cell Model for Direct Contact Membrane Distillation (DCMD) of Densely Packed Hollow Fibers, Journal of Membrane Science 455 (2014) 168–186. doi:10.1016/j.memsci.2013.12.067.

[29] G. Naidu, S. Jeong, S. Vigneswaran, Influence of feed/permeate velocity on scaling development in a direct contact membrane distillation, Separation and Purification Technology 125 (2014) 291–300. doi:10.1016/j.seppur.2014.01.049.

[30] B. Ashoor, S. Mansour, A. Giwa, V. Dufour, S. Hasan, Principles and applications of direct contact membrane distillation (DCMD): A comprehensive review, Desalination 398 (2016) 222–246. doi:10.1016/j.desal.2016.07.043.

[31] M. M. Teoh, S. Bonyadi, T.-S. Chung, Investigation of different hollow fiber module designs for flux enhancement in the membrane distillation process, Journal of Membrane Science 311 (1-2) (2008) 371–379. doi:10.1016/j.memsci.2007.12.054.

[32] X. Yang, R. Wang, A. G. Fane, Novel designs for improving the performance of hollow fiber membrane distillation modules, Journal of Membrane Science 384 (1-2) (2011) 52–62. doi:10.1016/j.memsci.2011.09.007.

[33] N. C. Mat, Y. Lou, G. G. Lipscomb, Hollow fiber membrane modules, Current Opinion in Chemical Engineering 4 (nil) (2014) 18–24. doi:10.1016/j.coche.2014.01.002.

[34] C. F. Wan, T. Yang, G. G. Lipscomb, D. J. Stookey, T.-S. Chung, Design and fabrication of hollow fiber membrane modules, Journal of Membrane Science 538 (2017) 96–107. doi:10.1016/j.memsci.2017.05.047.

[35] H. Lee, F. He, L. Song, J. Gilron, K. K. Sirkar, Desalination with a cascade of cross-flow hollow fiber membrane distillation devices integrated with a heat exchanger, AIChE Journal 57 (7) (2010) 1780–1795. doi:10.1002/aic.12409.

[36] A. S. Kim, M. S. Park, J. H. Mi, Computational fluid dynamics simulation of LiquiCel Membrane using OpenFOAM: Implication for membrane distillation, Desalination and Water Treatment 58 (2017) 360–367. doi:10.5004/dwt.2017.11422.

[37] J. Happel, Viscous flow in multiparticle systems: Slow motion of fluids relative to beds of spherical particles, AIChE J. 4 (2) (1958) 197–201. doi:DOI: 10.1002/aic.690040214.

[38] R. Tal(Thau), W. A. Sirignano, Cylindrical cell model for the hydrodynamics of particle assemblages at intermediate reynolds numbers, AIChE Journal 28 (2) (1982) 233–237. doi:10.1002/aic.690280210.

[39] J. Happel, H. Brenner, Low Reynolds Number Hydrodynamics, 1991.

[40] W. Bowen, F. Jenner, Dynamic ultrafiltration model for charged colloidal dispersions: A wigner-seitz cell approach, Chemical Engineering Science 50 (11) (1995) 1707–1736. doi:10.1016/0009-2509(95)00026-2.

[41] M. Zare, F. Z. Ashtiani, A. Fouladitajar, CFD modeling and simulation of concentration polarization in microfiltration of oil–water emulsions; application of an eulerian multiphase model, Desalination 324 (2013) 37–47. doi:10.1016/j.desal.2013.05.022.

[42] H. Lotfiyan, F. Z. Ashtiani, A. Fouladitajar, S. B. Armand, Computational fluid dynamics modeling and experimental studies of oil-in-water emulsion microfiltration in a flat sheet membrane using Eulerian approach, Journal of Membrane Science 472 (2014) 1–9. doi:10.1016/j.memsci.2014.08.036.

[43] J. Lou, J. Vanneste, S. C. DeCaluwe, T. Y. Cath, N. Tilton, Computational fluid dynamics simulations of polarization phenomena in direct contact membrane distillation, Journal of Membrane Science 591 (2019) 117150. doi:10.1016/j.memsci.2019.05.074.

[44] J. J. Monaghan, Smoothed particle hydrodynamics, Reports on Progress in Physics 68 (8) (2005) 1703–1759. doi:10.1088/0034-4885/68/8/R01.

[45] M. Kelager, Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics, 2006.

[46] J. W. Swegle, D. L. Hicks, S. W. Attaway, Smoothed Particle Hydrodynamics Stability Analysis, Journal of Computational Physics (2014) 1–12.

[47] R. A. Booth, D. Sijacki, C. J. Clarke, Smoothed particle hydrodynamics simulations of gas and dust mixtures, Monthly Notices of the Royal Astronomical Society 452 (4) (2015) 3932–3947. doi:10.1093/mnras/stv1486.

[48] P. Kumar, Q. Yang, V. Jones, L. McCue-Weil, Coupled SPH-FVM Simulation within the OpenFOAM Framework, Procedia IUTAM 18 (2015) 76–84. doi:10.1016/j.piutam.2015.11.008.

[49] S. Miyauchi, S. Takeuchi, T. Kajishima, A numerical method for mass transfer by a thin moving membrane with selective permeabilities, Journal of Computational Physics 284 (nil) (2015) 490–504. doi:10.1016/j.jcp.2014.12.048.
URL `https://doi.org/10.1016/j.jcp.2014.12.048`

[50] S. Miyauchi, S. Takeuchi, T. Kajishima, A numerical method for interaction problems between fluid and membranes with arbitrary permeability for fluid, Journal of Computational Physics 345 (nil) (2017) 33–57. doi:10.1016/j.jcp.2017.05.006.
URL `https://doi.org/10.1016/j.jcp.2017.05.006`

[51] R. Kaya, G. Deveci, T. Turken, R. Sengur, S. Guclu, D. Y. Koseoglu-Imer, I. Koyuncu, Analysis of wall shear stress on the outside-in type hollow fiber membrane modules by CFD simulation, Desalination 351 (2014) 109–119. doi:10.1016/j.desal.2014.07.033.

[52] L. Zhuang, H. Guo, G. Dai, Z.-l. Xu, Effect of the inlet manifold on the performance of a hollow fiber membrane module-A CFD study, Journal of Membrane Science 526 (2017) 73–93, 00004. doi:10.1016/j.memsci.2016.12.018.

[53] OpenFOAM Limited, OpenFOAM, The Open Source CFD Toolbox User Guide, Version v1812 (2018).

[54] J. Schoberl, NETGEN an advancing front 2d/3d-mesh generator based on abstract rules, Computing and Visualization in Science 1 (1) (1997) 41–52. doi:10.1007/s007910050004.

[55] J. Schoberl, NETGEN-4.x Manual (2009). {http://dx.doi.org/10.1007/s007910050004.

[56] Y. Yilmaz, C. Ozturan, Using sequential NETGEN as a component for a parallel mesh generator, Advances in Engineering Software 84 (2015) 3–12. doi:10.1016/j.advengsoft.2014.12.013.

[57] H. Si, TetGen: A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator, Version 1.5 (2013).

[58] H. Si, TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator, ACM Trans. Math. Softw. 41 (2) (2015) 11:1–11:36. doi:10.1145/2629697.

[59] C. Geuzaine, J. F. Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (11) (2009) 1309–1331.

[60] M. J. Aftosmis, M. J. Berger, J. E. Melton, Robust and efficient cartesian mesh generation for component-based geometry, AIAA Journal 36 (6) (1998) 952–960. doi:10.2514/2.464.

[61] A. Khan, B. Topping, Parallel adaptive mesh generation, Computing Systems in Engineering 2 (1) (1991) 75–101. doi:10.1016/0956-0521(91)90041-3.

[62] R. Said, N. Weatherill, K. Morgan, N. Verhoeven, Distributed parallel delaunay mesh generation, Computer Methods in Applied Mechanics and Engineering 177 (1-2) (1999) 109–125. doi:10.1016/s0045-7825(98)00374-0.

[63] N. Chrisochoides, D. Nave, Parallel delaunay mesh generation kernel, International Journal for Numerical Methods in Engineering 58 (2) (2003) 161–176. doi:10.1002/nme.765.

[64] Y. Ito, A. M. Shih, A. K. Erukala, B. K. Soni, A. Chernikov, N. P. Chrisochoides, K. Nakahashi, Parallel unstructured mesh generation by an advancing front method, Mathematics and Computers in Simulation 75 (5-6) (2007) 200–209. doi:10.1016/j.matcom.2006.12.008.

[65] W. Dawes, S. Harvey, S. Fellows, N. Eccles, D. Jaeggi, W. Kellar, A practical demonstration of scalable, parallel mesh generation, in: 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics, 2009. doi:10.2514/6.2009-981.

[66] P.E. Farrell and J.R. Maddison. Conservative interpolation between volume meshes by local Galerkin projection, Computer Methods in Applied Mechanics and Engineering, 200 (1–4) (2011) 89–100. doi:10.1016/j.cma.2010.07.015.

[67] https://cfd.direct/openfoam/user-guide/v7-standard-utilities/#x14-1120003.6.3

[68] A. S. Kim, S. J. Ki, H.-J. Kim, Research perspective of membrane distillation: Multi-scale and multi-physics phenomena, Desalination and

Water Treatment 58 (2017) 351–359. doi:10.5004/dwt.2017.11423.
[69] R. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational Physics 62 (1) (1986) 40–65. doi:10.1016/0021-9991(86)90099-9.
[70] R. Issa, A. Gosman, A. Watkins, The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme, Journal of Computational Physics 62 (1) (1986) 66–82. doi:10.1016/0021-9991(86)90100-2.
[71] R. I. I. Paulo J. Oliveira, An improved {PISO} algorithm for the computation of buoyancy-driven flows, Numerical Heat Transfer, Part B: Fundamentals 40 (6) (2001) 473–493. doi:10.1080/104077901753306601.
URL https://doi.org/10.1080/104077901753306601
[72] American Society of Mechanical Engineers, Procedure for estimation and reporting of uncertainty due to discretization in {CFD} applications, Journal of Fluids Engineering 130 (7) (2008) 078001. doi:10.1115/1.2960953.
URL https://doi.org/10.1115/1.2960953
[73] K. Y. Wang, T.-S. Chung, and M. Gryta, Hydrophobic PVDF Hollow Fiber Membranes With Narrow Pore Size Distribution and Ultra-Thin Skin for the Fresh Water Production Through Membrane Distillation, Chemical Engineering Science 63(9) (2008) 2587–2594. doi:10.1016/j.ces.2008.02.020.
[74] J. C. Maxwell, Treatise on Electricity and Magnetism, Dover, New York, 1881.
[75] M.M. Tomadakis, S.V. Sotirchos, Transport properties of random arrays of freely overlapping cylinders with various orientation distributions. Journal of Chemical Physics 98 (1983) 616–626. doi:10.1063/1.464604.
[76] P.T. Tsilingiris, Thermophysical and transport properties of humid air at temperature range between 0 and 100 °C, Energy Conversion and Management, 49(5) (2008) 1098-1110. doi.org/10.1016/j.enconman.2007.09.015.
[77] A. S. Kim and H.-J Kim, A Coupling Algorithm of Computational Fluid and Particle Dynamics (CFPD), in *Advanced Computational Fluid Dynamics for Emerging Engineering Processes - Eulerian vs. Lagrangian*, edited by A. S. Kim, IntechOpen, 2019. doi:10.5772/intechopen.86895.