Computers and Fluids 285 (2024) 106449

Contents lists available at ScienceDirect

Computers and Fluids

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

A numerical method for simulating variable density flows in membrane desalination systems

Federico Municchi [a,b], Yiming Liu [c], Jingbo Wang [c], Tzahi Y. Cath [d], Craig S. Turchi [e], Michael B. Heeley [f], Eric M.V. Hoek [c,g], David Jassby [c], Nils Tilton [b,*]

[a] Computational Science Center, National Renewable Energy Laboratory, Golden, CO, USA
[b] Colorado School of Mines, Department of Mechanical Engineering, Golden, CO, USA
[c] University of California Los Angeles (UCLA), Department of Civil & Environmental Engineering, California NanoSystems Institute and Institute of the Environment & Sustainability, Los Angeles, CA, USA
[d] Colorado School of Mines, Department of Civil & Environmental Engineering, Golden, CO, USA
[e] Buildings & Thermal Science Center, National Renewable Energy Laboratory, Golden, CO, USA
[f] Center for Entrepreneurship, College of Engineering, University of Michigan, Ann Arbor, MI, USA
[g] Energy Storage & Distributed Resources Division, Lawrence Berkeley National Lab, Berkeley, CA, USA

ARTICLE INFO

Keywords:
Membrane desalination
Variable density flow
Computational fluid dynamics

ABSTRACT

We present a novel method for simulating unsteady, variable density, fluid flows in membrane desalination systems. By assuming the density varies only with concentration and temperature, the scheme decouples the solution of the governing equations into two sequential blocks. The first solves the governing equations for the temperature and concentration fields, which are used to compute all thermophysical properties. The second block solves the conservation of mass and momentum equations for the velocity and pressure. We show that this is computationally more efficient than schemes that iterate over the full coupled equations in one block. We verify that the method achieves second-order spatial–temporal accuracy, and we use the method to investigate buoyancy-driven convection in a desalination process called vacuum membrane distillation. Specifically, we show that with gravity properly oriented, variations in temperature and concentration can trigger a double-diffusive instability that enhances mixing and improves water recovery. We also show that the instability can be strengthened by providing external heating.

1. Introduction

Membrane desalination processes have important applications to desalinating seawater and brackish water, and treating municipal and industrial wastewaters [1–3]. The energy efficiency and water recovery of membrane desalination systems depend on heat and mass transport that are challenging to observe experimentally, because they occur between optically opaque membrane sheets. Computational fluid dynamics (CFD) consequently plays an important role in elucidating the physics of membrane desalination [4–7]. The current study is motivated by the fact that fluid flows in membrane desalination systems can experience rapid variations in temperature and solute concentration near the membrane surfaces. The impacts of these variations on fluid density and buoyancy remain largely unexplored. Simulating these variations numerically faces two hurdles. First, the density in some cases varies by roughly 10%, pushing the limits of the Boussinesq assumption. Second, most variable-density, low-mach-number, CFD schemes are tailored to flows in which the fluid density is coupled to the pressure [8]. Such schemes typically iterate over all transport equations in one block [9]. In membrane desalination, however, the density varies primarily with concentration and temperature. This opens the door to a more efficient iterative procedure, that to our knowledge, remains unexplored.

Thus motivated, we develop a numerical method tailored to simulating variable-density flows in membrane desalination systems. By assuming the density varies only with concentration and temperature, the scheme decouples the solution of the governing equations into two sequential blocks. The first solves the governing equations for the temperature and concentration, which we then use to compute all thermophysical properties. The second uses those properties to solve conservation of mass and momentum for the velocity and pressure. We show this maintains second-order spatio-temporal accuracy, but is computationally more efficient than iterating over the full coupled equations as in [9].

To demonstrate our method, we explore density variations in a desalination process called *vacuum membrane distillation* (VMD) [10,11].

* Corresponding author.
E-mail address: ntilton@mines.edu (N. Tilton).

https://doi.org/10.1016/j.compfluid.2024.106449
Received 17 September 2023; Received in revised form 5 August 2024; Accepted 2 October 2024
Available online 11 October 2024

Nomenclature

Superscrips

$*$	Denotes a variable extrapolated in time, as in Eq. (24)
i	Denotes a variable at iteration i, as in Eq. (3.1)
n	Denotes a variable evaluated at time step n, as in Eq. (20)

Subscripts

C	Denotes a variable evaluated at a cell centroid
e	Denotes the manufactured solution defined in Eqs. (51)–(53)
f	Denotes a variable evaluated at a face centroid of a cell

Other Symbols

Δt	Time step (s)
δ	Membrane pore length (μm)
ϵ	Membrane porosity (unitless)
Γ	Molecular diffusivity (m^2/s)
κ	Thermal conductivity (J/m s °C)
λ	Second viscosity coefficient (Pa s)
λ_v	Latent heat of evaporation (J/kg)
$\langle c_m \rangle$	Time averaged concentration on membrane surface (g/l)
$\langle j_v \rangle$	Time averaged permeate flux (kg/ m^2 s), see Eq. (59)
$\mathbf{D}$, $\mathbf{M}$	Matrices defined in Eqs. (37)–(39)
$\mathbf{g}$	Gravitational vector (m/s^2)
$\mathbf{H}$	Vector defined in Eq. (39)
$\mathbf{I}$	Identity tensor (unitless)
$\mathbf{u}$	Velocity vector (m/s)
μ	Dynamic viscosity (Pa s)
ρ	Density (kg/m^3)
ρ_m	Local density of water on the feed surface of the membrane (kg/m^3)
$\boldsymbol{\tau}$	Viscous stress tensor (Pa)
τ	Membrane tortuosity (unitless)
$\hat{p}$	Reduced pressure $\hat{p} = p - \rho g y$ (Pa)
ξ	Mass fraction (unitless)
a_w	Activity coefficient (unitless)
B	Membrane vapor permeability (kg/m^2 s Pa)
B_{DG}	Membrane vapor permeability in the dusty gas model (kg/m^2 s Pa)
c	Salt concentration (g/l)
c_p	Specific heat capacity (J/kg °C)
C_S	Control surface (m^2)
C_V	Control volume (m^3)
C_{in}	Inlet concentration (g/l)
c_{max}	Local maximum concentration downstream of x_{on} (g/l), see Fig. 8
Co	Courant number (unitless), see Eq. (58)
$d\mathbf{S}$	Infinitesimal surface vector normal to C_S (m^2)
E_N	Relative spatial error (unitless), see Eq. (55)
$E_{\Delta t}$	Relative temporal error (unitless), see Eq. (56)
g	Gravitational acceleration (9.81 m/s^2)
H	Channel height (m)
h	Specific enthalpy (J/kg)
j_v	Transmembrane vapor mass flux (kg/m^2 s)
J_{ave}	Average permeate flux (kg/ m^2 s), see Eq. (60)
j_{max}	Local maximum in $\langle j_v \rangle$ (kg/ m^2 s), see Fig. 8
K	Kinetic energy per unit volume (J/m^3), $K = \rho(\mathbf{u}\cdot\mathbf{u})/2$
L	Channel length (m)
M_w	Molar weight of water (Kg/mol)
p	Pressure (Pa)
p_m	Partial vapor pressure on the feed surface of the membrane (Pa)
p_o	Average pressure prescribed at outlet (Pa)
p_{sat}	Saturation pressure (Pa)
p_{vac}	Absolute pressure on distillate side of membrane (Pa)
R	Universal constant of gases (J/mol K)
r_p	Membrane pore radius (μm)
r_{eos}	Unitless residual defined in Eq. (34)
Re	Inlet Reynolds number (unitless), $Re = \rho_{in} U_{in} H/\mu_{in}$.
S_c	Source term defined in Eq. (30)
T	Temperature (°C)
t	Time (s)
T_{in}	Inlet temperature (°C)
u, v, w	Velocity components in x, y, z directions, respectively (m/s)
U_{in}	Mean inlet velocity (m/s)
$u_{in}(y)$	Parabolic inlet velocity profile (m/s), see Eq. (13)
x, y, z	Cartesian coordinates (m)
x_{max}	Downstream distance where j_{max} occurs (m), see Fig. 8
x_{on}	Onset distance where $\langle j_v \rangle$ begins increasing (m), see Fig. 8

VMD flows warm feed water over a hydrophobic porous membrane, as in Fig. 1(*a*). A difference in partial vapor pressure across the membrane causes water to evaporate from the feed and travel through the membrane as vapor. Nonvolatile solutes remain in the feed. The vapor on the distillate side of the membrane is maintained at low pressure by a vacuum pump that draws the vapor into a condenser. Modern desalination systems currently rely on a process called reverse osmosis (RO). In comparison to RO, the advantage of VMD is that it is less sensitive to osmotic pressure. This allows VMD to concentrate NaCl solutions to above 300 g/l, compared to only 70 g/l using RO [2,12].

VMD faces two long-standing challenges. The first, called *temperature polarization*, is the cooling of the feed in a thermal boundary layer growing along the membrane (Fig. 1*a*). This reduces the feed's partial vapor pressure, and reduces vapor production. The second challenge, called *concentration polarization*, is the accumulation of solutes in a concentration boundary layer at the membrane. Though this also decreases the feed's partial vapor pressure, the greater risk is that it causes *mineral scaling*, which is the precipitation of solutes onto the membrane, where they block pores. Surprisingly little work considers that temperature and concentration polarization increase the feed density near the membrane [13–16]. We show that when gravity is properly oriented, this can trigger a double-diffusive buoyancy-driven instability, in which plumes of dense feed sink away from the membrane, as in Fig. 1(*b*). This brings warm, low-concentration, feed to the membrane, which increases vapor production.

The remaining manuscript is structured as follows. Section 2 describes the governing transport equations for VMD. Section 3 presents our spatial–temporal discretization and decoupling strategy. Section 4 verifies our method's spatial–temporal accuracy and compares with

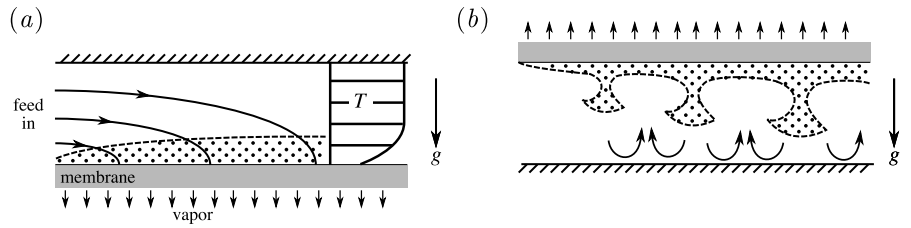

Fig. 1. (*a*) Sketch (not to scale) of a VMD system in which gravity points towards the membrane. The membrane is shaded gray. The dotted region shows the concentration boundary layer. The temperature profile illustrates temperature polarization. (*b*) Sketch of buoyancy driven instability when gravity points away from the membrane.

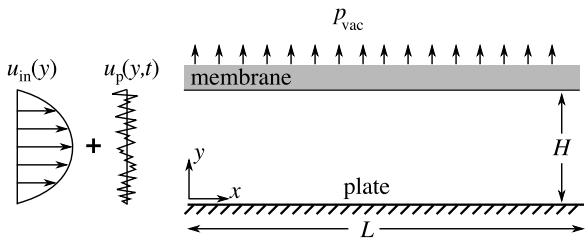

Fig. 2. Sketch (not to scale) of the VMD system considered in this work. The inlet velocity profile is the sum of a parabolic flow $u_{in}(y)$ and a perturbation $u_p(y,t)$.

experiments. Section 5, explores buoyancy-driven convection in VMD, and the stability of the proposed algorithm. Section 6 summarizes our results and conclusions.

2. Mathematical description

We consider a 2D channel of length L with a flat-sheet membrane at $y = H$ and an impermeable wall at $y = 0$, as in Fig. 2. A Newtonian feed solution of water and NaCl enters at $x = 0$ with a mean velocity U_{in}, uniform NaCl concentration C_{in}, and uniform temperature T_{in}. Concentrate exits the domain at $x = L$. As the feed flows downstream, water vapor exits the domain through the membrane. We assume a vacuum pump maintains a constant vapor pressure p_{vac} on the vapor side of the membrane. We consider cases where the gravitational vector $\boldsymbol{g}$ points in the negative or positive y-direction.

2.1. Governing equations

The feed mixture velocity $\mathbf{u} = [u\, v\, w]$ and pressure p are governed by the conservation of mass and momentum equations, where the feed density ρ is a function of concentration and temperature only,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0\,, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \rho \boldsymbol{g} + \nabla \cdot \boldsymbol{\tau} + \mathbf{F}_{\mathbf{u}}, \tag{2}$$

$$\boldsymbol{\tau} = \mu \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right] - \frac{2\mu}{3}(\nabla \cdot \mathbf{u})\mathbf{I}\,,$$

where μ is the dynamic viscosity, $\mathbf{I}$ is the identity matrix, $(\nabla \mathbf{u})^T$ is the gradient transpose, and $\mathbf{F}_{\mathbf{u}}$ is a forcing term used for verification in Section 4.1. Heat transport is governed by the thermal energy equation [17],

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\rho \mathbf{u} h) + \frac{DK}{Dt} - \frac{\partial p}{\partial t} = \nabla \cdot \left(\frac{\kappa}{c_p} \nabla h\right) + \rho(\boldsymbol{g} \cdot \mathbf{u}) + F_h\,, \tag{3}$$

where h, c_p, and κ are the mixture's specific enthalpy, specific heat capacity, and thermal conductivity, respectively, $DK/Dt = \partial K/\partial t + \nabla \cdot (\mathbf{u}K)$, and $K = \rho(\mathbf{u} \cdot \mathbf{u})/2$. The term F_h is a forcing term used for verification purposes in Section 4.1. Solute transport is governed by an advection-diffusion equation,

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u} c) = \nabla \cdot (\rho \Gamma \nabla \xi) + F_c\,, \tag{4}$$

where c is the concentration (mass of solute per liter of solution), $\xi = c/\rho$ is the mass fraction (mass of solute per mass of solution), Γ is the mass diffusivity of NaCl, and F_c is a forcing term used for verification in Section 4.1. The specific enthalpy is related to the temperature (T) through the relation

$$h(T) = \int_{T_{\text{ref}}}^{T} c_p(T')\, dT', \tag{5}$$

where T_{ref} is a reference temperature.

We express the variation of all thermophysical properties (μ, ρ, c_p, κ, Γ) with temperature and concentration using bivariate polynomials of the form

$$f(T, c) = \sum_{n=0}^{3} \sum_{m=0}^{4} a_{nm}\, T^n\, c^m. \tag{6}$$

We determine the coefficients a_{nm} for μ, ρ, c_p, and Γ by fitting (using MATLAB's "polyfit" command) to data from thermodynamic software [18]. The coefficients for κ are fit to data from Jamieson and Tudhope [19]. The fit requires c in Eq. (6) to be replaced with the practical salinity defined as $S_p = 1000\xi/1.00472$. All correlations are fit using data in the range $10 \le c \le 257$ (g/l) and $298.15 \le T \le 373.15$ (K), except for κ, which is valid in the range $0 \le S_p \le 160$ and $273.15 \le T \le 453.15$ (K). The maximum deviation between the correlations and data is 1.18%. Model coefficients are provided in the Supplementary Material.

2.2. Transmembrane heat and mass transport

Transport through the membrane is modeled on a macroscopic scale using the popular model of Schofield et al. [10], whose validity has been confirmed experimentally [20,21] for the conditions explored here. The model assumes the transmembrane vapor mass flux j_v is proportional to the transmembrane difference in partial vapor pressure,

$$j_v = -B\left[p_{\text{m}}(x,t) - p_{\text{vac}}\right]\,, \tag{7}$$

where p_{m} is the partial vapor pressure of the solution on the feed side of the membrane, and B is the vapor permeability [10,20,22]. There remains some discussion in the literature regarding B [23]. Lawson and Loyd [11] suggest B depends on the local temperature, while other studies [10,20,22] suggest B is well approximated as a constant membrane property. Though our method permits either approach, we assume B is constant for demonstration purposes.

Condition (7) produces the following condition for v on the membrane,

$$v\Big|_{y=H} = \frac{j_v}{\rho_m} = -\frac{B\left(p_{\text{m}} - p_{\text{vac}}\right)}{\rho_m}\,, \tag{8}$$

where ρ_m is the local water density at the membrane surface ($y = H$). Because no solutes traverse the membrane, we compute ρ_m as the density of pure water ($c = 0$ g/l). As in most prior literature, we apply the no-slip condition ($u = 0$) on the membrane. Throughout this study, *no-slip* implies the velocity tangential to a stationary surface is zero. The

term *no-penetration* implies the velocity normal to a stationary surface is zero [24].

We express the dependence of the partial vapor pressure p_m on temperature and concentration as in Schofield et al. [10],

$$p_\mathrm{m} = a_w p_\mathrm{sat}\,, \quad p_\mathrm{sat} = \exp\left(23.238 - \frac{3841}{T_m - 318}\right), \tag{9}$$

where the saturation pressure p_sat is computed using the Antoine equation [20], T_m is the local feed temperature on the membrane ($y = H$), and a_w is an activity coefficient given by

$$a_w = 1 - 0.03112\,c\big|_{y=H} - 0.001482\,c^2\big|_{y=H}\,. \tag{10}$$

Heat traverses the membrane due to advection by the vapor and conduction through the vapor and membrane material. In VMD, the contribution due to conduction is usually negligible due to the low thermal conductivity of the rarefied vapor opposite the membrane. Using that common assumption, conservation of energy [6] requires the following boundary condition,

$$-\kappa\frac{\partial T}{\partial y}\bigg|_{y=H} = j_v\lambda_v\,, \quad \lambda_v = 3177800 - 2464.4\,T_m\,, \tag{11}$$

where λ_v is the latent heat of evaporation of pure water. Consistent with all prior literature, we assume there is no transmembrane flux of NaCl. This requires

$$\left(j_v\xi - \rho\Gamma\frac{\partial\xi}{\partial y}\right)\bigg|_{y=H} = 0\,. \tag{12}$$

2.3. Inlet, outlet, and plate boundary conditions

Feed enters at $x = 0$ with uniform temperature and concentration,

$$T\big|_{x=0} = T_\mathrm{in}\,, \quad c\big|_{x=0} = C_\mathrm{in}\,. \tag{13}$$

To trigger buoyancy-driven instability independently of numerical noise, we impose an inlet parabolic velocity profile (u_{in}) with mean velocity U_in and an added perturbation (u_p),

$$u\big|_{x=0} = \underbrace{6U_\mathrm{in}\left(\frac{y}{H} - \frac{y^2}{H^2}\right)}_{\text{parabolic profile } u_{in}(y)} + \underbrace{A_p U_\mathrm{in} s(y,t)}_{\text{perturbation } u_p(y,t)}, \tag{14}$$

where $s(y,t) \in [-1,1]$ is a random number generated at each time step for each inlet face of our finite volume discretization. A_p is the perturbation magnitude relative to U_{in}, which we set to $A_p = 0.0075$. To accommodate the effects of the membrane at the inlet, we apply an inlet Neumann condition to v,

$$\frac{\partial v}{\partial x}\bigg|_{x=0} = 0\,. \tag{15}$$

At the outlet, we impose the following Neumann conditions,

$$\frac{\partial T}{\partial x}\bigg|_{x=L} = 0\,, \quad \frac{\partial c}{\partial x}\bigg|_{x=L} = 0\,, \quad \frac{\partial u}{\partial x}\bigg|_{x=L} = \frac{\partial v}{\partial x}\bigg|_{x=L} = 0\,. \tag{16}$$

Prior VMD simulations typically fix the outlet pressure [6,7,25]. However, we found that produces unphysical velocity fields at the outlet when ρ varies with T and c. This likely occurs because the transmembrane mass flow generates a vertical pressure profile that is inconsistent with a constant outlet pressure. We consequently fix an average outlet pressure p_o through the condition

$$\frac{1}{H}\int_0^H p\big|_{x=L}\,dy = p_o\,, \tag{17}$$

with $p_o = 10^5$ Pa. Our numerical tests showed that conditions (16)–(17), combined with a TVD (Total Variation Diminishing) van Leer scheme [26] (described in Section 3), allows flow structures to exit the domain cleanly.

At the lower plate, we apply no-slip and no-penetration conditions to the velocity, and zero-flux conditions to c and T,

$$u\big|_{y=0} = v\big|_{y=0} = 0\,, \quad \frac{\partial T}{\partial y}\bigg|_{y=0} = 0\,, \quad \frac{\partial c}{\partial y}\bigg|_{y=0} = 0\,. \tag{18}$$

Note that in Section 5.3, we explore the impact of heating the lower plate. For that, we use the following boundary condition,

$$\left(\kappa\frac{\partial T}{\partial y}\right)\bigg|_{y=0} = -\dot{q}_s\,, \tag{19}$$

where $\dot{q}_s$ is the provided heat flux.

3. Numerical formulation

We discretize Eqs. (1)–(4) in time using the BDF2 scheme [27]. For brevity, we neglect the forcing terms $\mathbf{F}_\mathbf{u}$, F_h, and F_c in Eqs. (2)–(4) and present our method assuming a constant time step Δt. More generally, we use adaptive time stepping, with details given in Supplementary Material. With a constant time step, we can write the governing transport equations in semidiscrete form as

$$\frac{3\rho^{n+1} + 4\rho^n - \rho^{n-1}}{2\Delta t} + \nabla\cdot(\rho\mathbf{u})^{n+1} = 0\,, \tag{20}$$

$$\frac{3(\rho\mathbf{u})^{n+1} + 4(\rho\mathbf{u})^n - (\rho\mathbf{u})^{n-1}}{2\Delta t} + \nabla\cdot(\rho\mathbf{u}\mathbf{u})^{n+1} = \nabla\cdot\boldsymbol{\tau}^{n+1} - \nabla\hat{p}^{n+1} + gy\nabla\rho^{n+1}\,, \tag{21}$$

$$\frac{3(\rho h)^{n+1} + 4(\rho h)^n - (\rho h)^{n-1}}{2\Delta t} + \nabla\cdot(\rho\mathbf{u}h)^{n+1} = -\frac{DK}{Dt}^{n+1} + \frac{\partial p}{\partial t}^{n+1} + \nabla\cdot\left(\frac{\kappa}{c_p}\nabla h\right)^{n+1} + \boldsymbol{g}\cdot(\rho\mathbf{u})^{n+1}\,, \tag{22}$$

$$\frac{3(\rho\xi)^{n+1} + 4(\rho\xi)^n - (\rho\xi)^{n-1}}{2\Delta t} + \nabla\cdot(\rho\mathbf{u}\xi)^{n+1} = \nabla\cdot(\rho\Gamma\nabla\xi)^{n+1}\,, \tag{23}$$

where n is the time step index, $\hat{p} = p - \rho(\boldsymbol{g}\cdot\boldsymbol{x})$ is the modified pressure, and $\boldsymbol{x} = [x, y]$ is the coordinate vector. Note that the term $(\boldsymbol{g}\cdot\boldsymbol{x})\nabla\rho$ is present in each component of the momentum equation.

Several methods of decoupling Eqs. (20)–(23) are proposed in prior literature [8,9,28–34]. These typically iterate over the full set of governing equations in a single block to obtain a density field that satisfies both mass conservation and an equation of state [9,31]. In contrast, we perform two segregated inner iterative loops, which provides two advantages in terms of performance. First, the overall convergence rate is improved because the flow variables are fixed while solving the first block, and the thermophysical properties are fixed while solving the second block. Second, each block only performs the iterations required to achieve the desired accuracy, while a full coupling would perform as many iterations as required by the slowest converging block. One drawback is that the first block must predict several flow variables using temporal extrapolation. This requires stricter conditions on the time step to maintain stability, compared to a fully coupled algorithm.

3.1. Solution algorithm

The solution algorithm consists of three steps summarized in Algorithm 1. The first step predicts $(\rho\mathbf{u})$, ρ, $\partial p/\partial t$, $\mathbf{u}$, and DK/Dt using second-order extrapolation in time. The second step iteratively solves the advection-diffusion and energy equations to update the equation of state and compute the thermophysical properties. Finally, the third step iteratively solves the pressure-velocity coupling using a Pressure Implicit Operator Splitting (PISO) method [35].

Step 1 - The advection-diffusion and energy Eqs. (3)–(4) depend on ρ^{n+1}, $\mathbf{u}^{n+1}$, $(DK/Dt)^{n+1}$, and $(\partial p/\partial t)^{n+1}$. Step 1 estimates these using second-order extrapolation in time. We first compute the extrapolated mass flux

$$(\rho\mathbf{u})^* = 2(\rho\mathbf{u})^n - (\rho\mathbf{u})^{n-1}\,, \tag{24}$$

where the asterisk denotes an approximation to $(\rho\mathbf{u})^{n+1}$. We then substitute $(\rho\mathbf{u})^*$ into Eq. (20) to compute an extrapolated density and velocity

$$\frac{3}{2\Delta t}\rho^* = \nabla\cdot(\rho\mathbf{u})^* - \frac{4\rho^n - \rho^{n-1}}{2\Delta t}\,, \qquad \mathbf{u}^* = \frac{(\rho\mathbf{u})^*}{\rho^*}\,. \tag{25}$$

Algorithm 1 Solution algorithm

```
for t < t_max do                                              ▷ Time Loop
    t = t + Δt                                                ▷ Compute time step Δt and update time
    Compute the extrapolated flow variables                   ▷ Equations (24)–(27)
    r_eos = 0                                                 ▷ Initialize residual for density
    while r_eos ≥ tolerance do                                ▷ Advection-diffusion/Energy loop
        Solve the advection-diffusion equation for ξ^{i+1}    ▷ Equation (29)
        Solve the energy equation for h^{i+1}                 ▷ Equation (33)
        Update thermophysical properties and r_eos
    end while
    Compute u^i by solving equation (36) with p̂^n            ▷ Predictor step
    corr = 0                                                  ▷ Initialize pressure corrector
    while corr ≤ maximum iterations do                        ▷ Correction loop
        Solve the Poisson equation for p̂^{i+1}                ▷ Equation (41)
        Update mass flux (ρu)^{i+1} and velocity u^{i+1}      ▷ Equation (39)
        corr = corr + 1
    end while
    p̂^{n+1} = p̂^{i+1}, (ρu)^{n+1} = (ρu)^{i+1}, u^{n+1} = u^{i+1}
end for
```

We then compute the extrapolated derivatives of K and p,

$$\left(\frac{DK}{Dt}\right)^{*} = 2\left(\frac{DK}{Dt}\right)^{n} - \left(\frac{DK}{Dt}\right)^{n-1}, \tag{26}$$

$$\left(\frac{\partial p}{\partial t}\right)^{*} = 2\left(\frac{\partial p}{\partial t}\right)^{n} - \left(\frac{\partial p}{\partial t}\right)^{n-1}, \tag{27}$$

where the time derivatives are computed using the BDF2 scheme. All extrapolated quantities are kept constant for the rest of the algorithm, and are only computed at the beginning of each temporal step.

Step 2 - The second step solves the advection-diffusion and energy equations iteratively by updating the thermophysical properties within an iterative loop. Introducing the iteration index i, we first initialize the following variables

$$\xi^{i} = \xi^{n}, \quad h^{i} = h^{n}, \quad \rho^{i} = \rho^{*}, \quad c_p^{i} = c_p\left(c^{n}, T^{n}\right),$$

$$\kappa^{i} = \kappa\left(c^{n}, T^{n}\right), \quad \Gamma^{i} = \Gamma\left(c^{n}, T^{n}\right). \tag{28}$$

We then solve the advection-diffusion Eq. (23) in conservative form,

$$\frac{3}{2\Delta t}\rho^{i}\xi^{i} + \nabla\cdot\left[(\rho\mathbf{u})^{*}\,\xi^{i+1}\right] = \nabla\cdot\left(\rho^{i}\Gamma^{i}\nabla\xi^{i+1}\right) + \xi^{i+1}S_c^{i} - \frac{4\rho^{n}\xi^{n} - \rho^{n-1}\xi^{n-1}}{2\Delta t}, \tag{29}$$

where the source term S_c^{i} is given by

$$S_c^{i} = \frac{3\rho^{i} + 4\rho^{n} - \rho^{n-1}}{2\Delta t} + \nabla\cdot(\rho\mathbf{u})^{*}. \tag{30}$$

Note that S_c^{i} is not zero at convergence, because it accounts for the difference between ρ^{*} and ρ^{i}. At the end of the loop, when the new density ρ^{n+1} is computed, it can be shown that

$$S_c^{n+1} = \frac{\partial\rho}{\partial t}^{n+1} - \frac{\partial\rho}{\partial t}^{*}. \tag{31}$$

Because both time derivatives in Eq. (31) are discretized using BDF2, the error introduced by S_c^{i} is at most $\mathcal{O}\left(\Delta t^{2}\right)$. As described in the next step, the mass flux is consistent with the equation of state, so that the extrapolated time derivative of the density is given by

$$\frac{\partial\rho}{\partial t}^{*} = 2\frac{\partial\rho}{\partial t}^{n} - \frac{\partial\rho}{\partial t}^{n-1} = -\nabla\cdot\left[2(\rho\mathbf{u})^{n} - (\rho\mathbf{u})^{n-1}\right] = -\nabla\cdot(\rho\mathbf{u})^{*}. \tag{32}$$

Hence, the divergence of the extrapolated mass flux can be seen as a second-order accurate extrapolation of the time derivative of the density, consistent with the equation of state. After solving the advection-diffusion Eq. (29), we solve the energy equation using the source term S_c^{i},

$$\frac{3}{2\Delta t}\rho^{i}h^{i+1} + \nabla\cdot\left[(\rho\mathbf{u})^{*}\,h^{i+1}\right] = -\left(\frac{DK}{Dt}\right)^{*} + \left(\frac{\partial p}{\partial t}\right)^{*} + \nabla\cdot\left(\frac{\kappa^{i}}{c_p^{i}}\nabla h^{i+1}\right) + \boldsymbol{g}\cdot\left(\rho^{i}\mathbf{u}^{*}\right) + h^{i+1}S_c^{i} - \frac{4\rho^{n}h^{n} - \rho^{n-1}h^{n-1}}{2\Delta t}. \tag{33}$$

After computing h^{i+1}, the temperature T^{i+1} is obtained from Eq. (5) using Newton iterations. The new solute concentration is computed as $c^{i+1} = \rho^{i}\xi^{i+1}$, and all thermophysical properties are updated to $i+1$ using c^{i+1} and T^{i+1}. To quantify the convergence of the iterative loop in step 2, we compute the following residual r_{eos} on the equation of state using the infinity norm,

$$r_{\text{eos}} = \frac{\|\rho^{i+1} - \rho^{i}\|_{\infty}}{\|\rho^{i+1}\|_{\infty}}. \tag{34}$$

When r_{eos} falls below a preset threshold, the loop is interrupted, and the fields at iteration $i+1$ are taken as the values for time step $n+1$.

Step 3 - This final step solves the conservation of mass and momentum Eqs. (1)–(2) using a modified PISO method with second-order temporal accuracy. Using i as the iteration index, we initialize the flow variables as

$$\mathbf{u}^{i} = \mathbf{u}^{*}, \quad p^{i} = p^{n}, \quad (\rho u)^{i} = (\rho u)^{*}. \tag{35}$$

The momentum equation is then written in the linearized form

$$\frac{3\rho^{n+1}}{2\Delta t}\mathbf{u}^{i+1} + \nabla\cdot\left[(\rho\mathbf{u})^{*}\,\mathbf{u}^{i+1}\right] = \nabla\cdot\boldsymbol{\tau}^{i+1} - \nabla\hat{p}^{i+1} + gy\nabla\rho^{n+1} - \frac{4\rho^{n}\mathbf{u}^{n} - \rho^{n-1}\mathbf{u}^{n-1}}{2\Delta t}, \tag{36}$$

where we discretize the stress tensor temporally as

$$\boldsymbol{\tau}^{i+1} = \nabla\cdot\left(\mu^{n+1}\nabla\mathbf{u}^{i+1}\right) + \nabla\cdot\left[\mu^{n+1}\left(\nabla\mathbf{u}^{*}\right)^{T} + \lambda^{n+1}\left(\nabla\cdot\mathbf{u}^{*}\right)\mathbf{I}\right]. \tag{37}$$

In the PISO formulation, it is convenient to represent the semi-discrete momentum equation in the compact form

$$\mathbf{M}\rho^{n+1}\mathbf{u}^{i+1} = \boldsymbol{b}_{\mathbf{u}}^{n} - \nabla\hat{p}^{i+1} + gy\nabla\rho^{n+1}, \tag{38}$$

where $\mathbf{M}$ is the matrix arising from the spatio-temporal discretization of the differential operators, and $\boldsymbol{b}_{\mathbf{u}}^{n}$ is a vector of explicit terms, such as those arising from temporal discretization. We then apply the decomposition $\mathbf{M} = \mathbf{D} + \mathbf{N}$, where $\mathbf{D}$ is the diagonal of $\mathbf{M}$, and we define the vector $\mathbf{H}^{i} = -\mathbf{N}\rho^{n+1}\mathbf{u}^{i} + \boldsymbol{b}_{\mathbf{u}}^{n}$. After multiplying both sides by of Eq. (39) by $\mathbf{D}^{-1}$ we obtain the expression

$$\rho^{n+1}\mathbf{u}^{i+1} = \mathbf{D}^{-1}\left[\mathbf{H}^{i} - \nabla\hat{p}^{i+1} + \left(gy\nabla\rho^{n+1}\right)\right]. \tag{39}$$

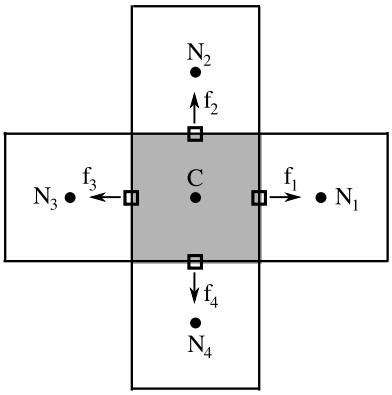

Fig. 3. Sketch of a cell C (shaded gray). Dots denote cell centers. Squares denote face centers. Arrows indicate face-normal vectors.

Eq. (39) can be interpreted as a Jacobi decomposition of Eq. (36). Taking the divergence of Eq. (39) produces the pressure Poisson equation

$$\nabla\cdot\left(\mathbf{D}^{-1}\nabla\hat{p}^{i+1}\right)=\nabla\cdot\left[\mathbf{D}^{-1}\left(\mathbf{H}^{i}+g\nabla\rho^{n+1}\right)\right]-\nabla\cdot\left(\rho^{n+1}\mathbf{u}^{i+1}\right)\,. \tag{40}$$

We leverage mass conservation to replace $(\partial p/\partial t)^{n+1}$ with $(\rho\mathbf{u})^{n+1}$, resulting in

$$\nabla\cdot\left(\mathbf{D}^{-1}\nabla\hat{p}^{i+1}\right)=\nabla\cdot\left[\mathbf{D}^{-1}\left(\mathbf{H}^{i}+g\nabla\rho^{n+1}\right)\right]+\frac{3\rho^{n+1}+4\rho^{n}-\rho^{n-1}}{2\Delta t}\,. \tag{41}$$

Boundary conditions for Eq. (41) are discussed in Section 3.2. The initialization of the PISO algorithm consists of a predictor step in which Eq. (36) is solved implicitly using $\hat{p}^{n}$ to initialize $\mathbf{u}^{i}$. The Poisson Eq. (41) is then solved using a corrector iterative loop to compute $\hat{p}^{i+1}$, and Eq. (39) is used to update $\mathbf{u}^{i+1}$, a strategy commonly employed in projection methods and SIMPLE-like schemes [36]. The corrector loop is repeated until the maximum number of iterations is reached. The number of iterations is chosen to reduce the residuals on the Poisson equation below a set threshold. Finally, $\mathbf{u}^{n+1}$, $(\rho\mathbf{u})^{n+1}$, and p^{n+1} are updated using values from the last iteration of the PISO scheme.

3.2. Finite volume discretization

We discretize Eqs. (20)–(23) spatially using a colocated finite volume method. The dependent variables $\mathbf{u}$, p, h, ρ, and ξ are stored at the cell centroids (dots in Fig. 3). The mass fluxes $\rho\mathbf{u}$ are stored at face centers (squares in Fig. 3). For demonstration, consider the conservation of mass Eq. (25). Finite volume methods discretize the governing transport equations in control volume form. Eq. (25) consequently takes the form

$$\int_{C_V}\left(\frac{3\rho^{*}+4\rho^{n}-\rho^{n-1}}{2\Delta t}\right)dV+\int_{C_S}(\rho\mathbf{u})^{*}\cdot d\boldsymbol{S}=0\,, \tag{42}$$

where C_V is the control volume, C_S is the control surface, and $d\boldsymbol{S}$ is the infinitesimal surface vector normal to C_S. The integrals in Eq. (42) are approximated using mid-point Gauss quadrature, producing the discrete form

$$V_C\left(\frac{3\rho_C^{*}+4\rho_C^{n}-\rho_C^{n-1}}{2\Delta t}\right)+\sum_{\mathrm{f}}(\rho\mathbf{u})_{\mathrm{f}}^{*}\cdot\boldsymbol{S}_{\mathrm{f}}=0\,, \tag{43}$$

where the subscripts C and f denote values at the cell and face centroids, respectively, V_C is the cell volume, and $\boldsymbol{S}_{\mathrm{f}}$ is the area vector normal to face f. Summations over f are carried over all cell faces. Using a similar approach, the momentum Eq. (36) takes the form

$$\begin{aligned}V_C\frac{3\rho_C^{n+1}}{2\Delta t}\mathbf{u}_C^{i+1}&+\sum_{\mathrm{f}}(\rho\mathbf{u})_{\mathrm{f}}^{*}\overline{\mathbf{u}_{\mathrm{f}}^{i+1}}\cdot\boldsymbol{S}_{\mathrm{f}}=\sum_{\mathrm{f}}\boldsymbol{\tau}_{\mathrm{f}}^{i+1}\cdot\boldsymbol{S}_{\mathrm{f}}\\&-V_C\left[\overline{\left(\nabla\hat{p}^{i+1}+gy\nabla\rho^{n+1}\right)_C}+\frac{4\rho_C^{n}\mathbf{u}_C^{n}-\rho_C^{n-1}\mathbf{u}_C^{n-1}}{2\Delta t}\right]\,,\end{aligned} \tag{44}$$

where the face-interpolated stress tensor is written as

$$\boldsymbol{\tau}_{\mathrm{f}}^{i+1}\cdot\boldsymbol{S}_{\mathrm{f}}=\mu_{\mathrm{f}}^{i}(\nabla\mathbf{u})_{\mathrm{f}}^{i+1}\cdot S_{\mathrm{f}}+\left[\mu_{\mathrm{f}}^{i}\overline{(\nabla\mathbf{u}^{*})_{\mathrm{f}}^{T}}+\lambda_{\mathrm{f}}^{i}\overline{(\nabla\cdot\mathbf{u}^{*})}_{\mathrm{f}}\mathbf{I}\right]\cdot\boldsymbol{S}_{\mathrm{f}}\,. \tag{45}$$

Overlines in Eqs. (44)–(45) indicate quantities computed using special interpolation or reconstruction methods. The face velocity in the advective term of Eq. (44) is computed using a van Leer TVD scheme [26], while the gradient and divergence in Eq. (45) are first computed on cell C using the Gauss theorem, and then linearly interpolated to the face f. To avoid the checkerboard instability [37], the pressure source term in Eq. (44) is reconstructed using the scheme of Weller et al. [38,39],

$$\overline{(\nabla\hat{p}-gy\nabla\rho)}_C=\left(\sum_{\mathrm{f}}\frac{\boldsymbol{S}_{\mathrm{f}}\boldsymbol{S}_{\mathrm{f}}}{|\boldsymbol{S}_{\mathrm{f}}|}\right)^{-1}\cdot\sum_{\mathrm{f}}\left[(\nabla\hat{p})_{\mathrm{f}}-gy\,(\nabla\rho)_{\mathrm{f}}\right]\cdot\frac{\boldsymbol{S}_{\mathrm{f}}\boldsymbol{S}_{\mathrm{f}}}{|\boldsymbol{S}_{\mathrm{f}}|}\,, \tag{46}$$

where $|\boldsymbol{S}_{\mathrm{f}}|$ denotes the face area, and g is the magnitude of the gravitational acceleration. The face gradients are discretized using a linear finite difference stencil. This requires the inversion of a 3×3 matrix in 3D. The same approach is used to reconstruct the velocity field from the mass fluxes in Eq. (25),

$$\mathbf{u}_C^{*}=\left(\sum_{\mathrm{f}}\frac{\boldsymbol{S}_{\mathrm{f}}\boldsymbol{S}_{\mathrm{f}}}{|\boldsymbol{S}_{\mathrm{f}}|}\right)^{-1}\cdot\sum_{\mathrm{f}}\frac{(\rho\mathbf{u})_{\mathrm{f}}^{*}}{\overline{\rho}_{\mathrm{f}}^{*}}\cdot\frac{\boldsymbol{S}_{\mathrm{f}}\boldsymbol{S}_{\mathrm{f}}}{|\boldsymbol{S}_{\mathrm{f}}|}\,, \tag{47}$$

where $\overline{\rho}_{\mathrm{f}}^{*}$ is the linearly interpolated density at face f. The advection-diffusion (29) and energy Eqs. (33) are similarly discretized in the Supplementary Material.

In the PISO corrector step, the velocity Eq. (39) is interpolated to the faces to obtain an expression for the face fluxes. To prevent checkerboarding, we compute the mass flux using Rhie-Chow interpolation [40]

$$(\rho\mathbf{u})_{\mathrm{f}}^{i+1}=\overline{\mathbf{D}}_{\mathrm{f}}^{-1}\left[\widetilde{\mathbf{H}^{i}}_{\mathrm{f}}-(\nabla\hat{p}^{i+1})_{\mathrm{f}}+\left(gy\nabla\rho^{n+1}\right)_{\mathrm{f}}\right]\,, \tag{48}$$

where $\overline{\mathbf{H}^{i}}_{\mathrm{f}}$ is linearly interpolated at face f. Eq. (48) is used to update the mass flux and to obtain boundary conditions for the Poisson Eq. (41) in the form of a Neumann condition,

$$\mathbf{n}_{\mathrm{f}}\cdot(\nabla\hat{p}^{i+1})_{\mathrm{f}}=\mathbf{n}_{\mathrm{f}}\cdot\left[\widetilde{\mathbf{H}^{i}}_{\mathrm{f}}+\left(gy\nabla\rho^{n+1}\right)_{\mathrm{f}}-\mathbf{D}(\rho\mathbf{u})_{\mathrm{f}}^{i+1}\right]\,. \tag{49}$$

This condition is applied at all boundary faces, except for the outlet, where we apply Eq. (17). Finally, the velocity correction Eq. (39) is discretized using the reconstruction scheme in Eq. (46), consistent with Eq. (36),

$$\rho_C^{n+1}\mathbf{u}_C^{i+1}=\mathbf{D}_C^{-1}\left[\mathbf{H}_C^{i}-\overline{\left(\nabla\hat{p}^{i+1}+gy\nabla\rho^{n+1}\right)}_C\right]\,. \tag{50}$$

4. Verification and validation

4.1. Verification

We implement our method using the C++ library OpenFOAM®, which provides a broad choice of methods for parallel linear algebra and finite volume discretization [41]. We verify the spatial–temporal accuracy of the proposed algorithm using the method of manufactured solutions [42]. The method chooses desired exact solutions for $\mathbf{u}$, p, T, and c, which are substituted into Eqs. (2)–(4) to compute the required forcing terms $\mathbf{F}_{\mathbf{u}}$, F_h, and F_c analytically. The manufactured solutions are chosen to satisfy conservation of mass, so that no source term is required in Eq. (1). For our purposes, we built the following manufactured solution in a square periodic domain $x\in[0,2\pi]$, $y\in[0,2\pi]$,

$$\begin{bmatrix}u_e\\v_e\\p_e\end{bmatrix}=\begin{bmatrix}\sin(x)\cos(y)\\-\cos(x)\sin(y)\\\sin(x)\sin(y)\end{bmatrix}\cos(\omega t) \tag{51}$$

$$c_e=100+50\left[\sin(x)\sin(y)-\cos(x)\cos(y)\right]\cos(\omega t)\,, \tag{52}$$

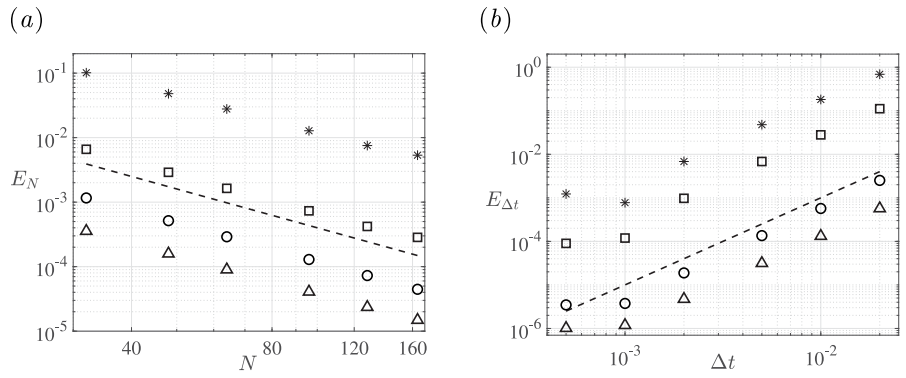

Fig. 4. (*a*) E_N vs. N. (*b*) $E_{\Delta t}$ vs. Δt. The symbols denote results for p (asterisks), v (squares), c (circles), and T (triangles). Results for u are not shown, because they are indistinguishable from those for v. The dashed black lines show second-order accuracy.

$$T_e = 300 + 30\,[\sin(x)\sin(y) + \cos(x)\cos(y)]\cos(\omega t)\,, \tag{53}$$

where ω is an angular frequency. The solution satisfies the constitutive equations

$$\Gamma = 1 + 0.001c - 0.002T\,, \quad \rho = 1 + 0.003c + 0.005T\,,$$
$$\mu = 10 + 0.01c - 0.01T\,, \quad c_p = 1\,, \quad \kappa = 1\,, \tag{54}$$

We omit units in Eqs. (51)–(54) because they play no role in the analysis.

To verify the spatial accuracy of our method, we set $\omega = 0$ so the manufactured solution is steady. We consider a uniform grid with N^2 cells (N in each direction), and integrate in time from the initial condition $\mathbf{u}^0 = p^0 = 0$, $c^0 = 100$, $T^0 = 300$ to steady-state. We then compute the relative spatial error

$$\mathrm{E}_N = \frac{\|\psi_N - \psi_e\|_\infty}{\|\psi_e\|_\infty}\,, \tag{55}$$

where ψ_N and ψ_e are the numerical and exact solutions of $\mathbf{u}$, T, ξ, and p respectively. Fig. 4(*a*) shows that the method achieves second-order spatial accuracy for all variables.

To verify the temporal accuracy of the method, we impose the manufactured solution at time $t = 0$ as the initial condition. We then integrate in time from $t = 0$ to $t = 2$ using $\omega = 2\pi$. At $t = 2$, we compute the relative temporal error

$$\mathrm{E}_{\Delta t} = \frac{\|\psi_{\Delta t} - \psi_e\|_\infty}{\|\psi_e\|_\infty}\,, \tag{56}$$

where $\psi_{\Delta t}$ is the field computed using a time step Δt. To minimize the spatial error, we set $N = 750$, which is the largest value that maintained stability over the range of time steps considered. Fig. 4(*b*) shows the method achieves second-order temporal accuracy when $\Delta t > 10^{-3}$. For smaller time steps, the error plateaus to that generated by the spatial discretization. Note that the method achieves second-order temporal accuracy for the pressure in the PISO scheme thanks to the flux extrapolation in Eq. (24), as reported by Lee [43].

4.2. Validation

Prior experimental studies of VMD are of limited value for validating CFD, because to our knowledge, no prior work explores buoyancy-driven convection, and the gravitational orientation is rarely reported. VMD is not a mature technology, and most experiments use short feed channels over which inlet and outlet effects are important. Feed typically enters and exits the feed channels through holes in the plate opposite the membrane, and the geometry of these holes is rarely reported. We nevertheless compare the predictions of our method with Anqi et al. [25], who performed experiments and CFD simulations of a VMD system in which the feed channel is a 3D duct of $H = 3$ mm, $L = 7$ cm, and a spanwise width of 1.5 cm. Gravity was oriented as in Fig. 1(*a*), such that there was no potential for buoyancy-driven convection. Anqi et al. use the standard incompressible flow solver of OpenFOAM®, which ignores gravity and variations in thermophysical properties.

Table 1
Comparison of the average permeate flux predicted in the current study (column 2) at steady-state with the experiments (column 3) and simulations (column 4) of Anqi et al. [25]. Results are shown for the lean inlet feed velocities shown in column 1.

U_{in} [m/s]	Average vapor flux [kg m^{-2} s^{-1}]		
	Current work	Experiments [25]	CFD [25]
0.04	13.39	11.57	14.15
0.11	17.40	16.93	17.86
0.19	19.66	19.7	20

Because Anqi et al. do not detail how feed enters and exits their feed channel, we assume a parabolic inlet velocity profile, as in their simulations. To be consistent with their CFD, we approximate the permeability B using the dusty gas model [11], in which B is expressed as a combination of a diffusive component that models collisions between vapor molecules, and a viscous component that models interactions between the vapor and the membrane material. In VMD, only the diffusive component is relevant [44], and we therefore express B as [11]

$$B_{\mathrm{DG}} = \frac{2}{3}\frac{\varepsilon r_p}{\tau^2 R T_m \delta}\sqrt{\frac{8RT_m}{\pi M_{\mathrm{w}}}}\,, \tag{57}$$

where $r_p = 0.2$ μm is a pore radius, $\varepsilon = 0.8$ and $\tau = 1.25$ are the membrane porosity and tortuosity, respectively, $R = 8.3144$ J/(mol K) is the universal gas constant, $M_{\mathrm{w}} = 0.01803$ kg/mol is the molecular weight of water, and $\delta = 100$ μm is the pore length. These values are all taken from Anqi et al. [25]. Though Anqi et al. performed 3D simulation; our simulations showed good agreement using a 2D grid of 185 × 64 cells, detailed in the Supplementary Material. As in Anqi et al., we set $T_{in} = 50$ °C and $T_{in} = 90$ g/l.

Table 1 compares our predictions for the average transmembrane vapor flux with the experiments and simulations of Anqi et al. [25]. The vapor flux is averaged over the membrane length at steady state. Due do the lack of information regarding the inlet and outlet geometry, the comparison is only qualitative. Future work should perform dedicated experiments to properly validate our mathematical model, which is beyond the scope of the present work.

5. Buoyancy-driven convection in membrane desalination

To investigate buoyancy-driven convection in VMD, we set the permeability to $B = 1.8676 \times 10^{-6}$ kg/m^2 s Pa, which models a membrane studied in Lou et al. [20]. We fix the channel height $H = 3$ mm, length $L = 15$ cm, vacuum pressure $p_{\mathrm{vac}} = 5000$ Pa, inlet temperature

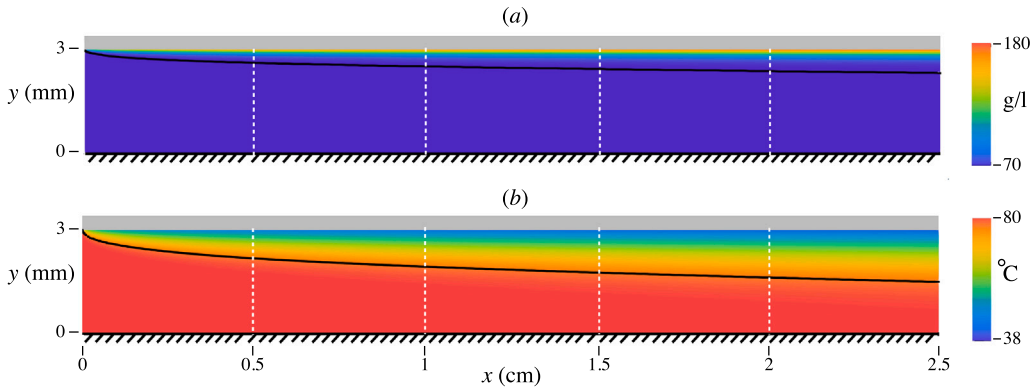

Fig. 5. The steady-state concentration (panel *a*) and temperature (panel *b*) fields when $Re = 75$ and $\mathbf{g} = [0, 9.81]$ [m/s^2] (stable configuration). Results are shown for $x \leq 2.5$ cm. The black lines show where $c = 1.01\, C_{\text{in}}$ and $T = 0.99\, T_{\text{in}}$. To aid visualization, the membrane is shaded gray at $y = 3$ mm, and the plate is sketched at $y = 0$.

$T_{\text{in}} = 80$ °C, and inlet concentration $c_{\text{in}} = 70$ g/l. These are typical of a bench-scale system treating a hypersaline brine [45]. We vary U_{in} so that the inlet Reynolds number varies between $50 \leq Re \leq 100$, where $Re = \rho_{in} U_{\text{in}} H / \mu_{in}$ is defined using the inlet density ρ_{in} and viscosity μ_{in}. We initiate all simulations with $u^0 = u_{in}(y)$, $v^0 = 0$, $c^0 = c_{\text{in}}$, $p^0 = 10^5$ Pa, and $T^0 = 33$ °C, where $u_{in}(y)$ is given in Eq. (14). We use a grid that is uniform in x, but that concentrates cells near the membrane, as detailed in the Supplementary Material.

Our simulations use adaptive time stepping to satisfy the condition $Co < 0.5$, where Co is a face-based Courant number,

$$Co = \frac{\Delta t\, \mathbf{u}_{\text{f}} \cdot \boldsymbol{S}_{\text{f}}}{|\boldsymbol{S}_{\text{f}}||\boldsymbol{d}_{\text{f}}|}, \tag{58}$$

where $|\boldsymbol{d}_{\text{f}}|$ is the distance between the centroids of the cells straddling face f. The limit $Co = 0.5$ is based on a compromise between stability, accuracy, and efficiency, and is in the stable range of Co for the PISO scheme at the Reynolds numbers considered [46]. We mitigate the initial transient by initiating simulations with $\Delta t = 10^{-10}$ s (producing $Co \ll 0.5$), and we limit the increase in Δt between two time steps to a factor of two. This allows Δt to grow smoothly.

From the available theory for Rayleigh-Bénard-Poiseuille flow [47–51], we expect the dominant modes of instability (i.e., the least stable modes) in VMD to be 3D. However, due to the computational cost of 3D simulations, and the limited objectives of this study, we focus on a 2D study in Sections 5.1–5.4. That study shows that density variations are indeed important in VMD, and deserve future study. We conclude with a 3D simulation in Section 5.5. The simulation confirms that 3D effects further destabilize the flow.

5.1. Impact of gravitational orientation

To demonstrate the impact of gravitational orientation, we begin by setting $Re = 75$ and $\mathbf{g} = [0, 9.81]$ [m/s^2] (pointing in the y-direction). The density gradient is stable, and the simulation is run to steady-state. Fig. 5 shows the resulting steady-state concentration (panel *a*) and temperature (panel *b*) fields for $x \leq 2.5$ cm. We identify the boundary layers by plotting the black iso-lines $c = 1.01\, c_{\text{in}}$ and $T = 0.99\, T_{\text{in}}$. The concentration boundary layer is thinner than the thermal layer, because the mass diffusivity is much smaller than the thermal diffusivity. Though not shown, for brevity, the boundary layers continue growing monotonically with downstream distance for $x > 2.5$ cm.

Keeping $Re = 75$, we now set $\mathbf{g} = [0, -9.81]$ [m/s^2], so that gravity points away from the membrane. In this case, our simulation predicts there is no steady-state. Figs. 6 and 7 show the instantaneous concentration and temperature fields, respectively, at $t = 300$ s for different downstream sections (see captions). In panel (*b*) of both figures, a sinusoidal perturbation becomes evident in the iso-line around $3 \leq x \leq 4$ cm, and plumes become evident for $x > 4.25$ cm. Because the thermal diffusivity is orders-of-magnitude larger than the mass diffusivity, energy mixing occurs more rapidly than solute mixing, and the iso-line for $T = 0.99 T_{in}$ disappears around $x \approx 5$ cm, while those for $c = 1.01 c_{in}$ after $x \approx 10$ cm. In both the stable and unstable configurations shown in Figs. 5–7, the thermophysical properties of the feed vary significantly. The density varies by approximately 8%, the viscosity by more than 100%, the mass diffusivity by roughly 50%, and the thermal conductivity and capacity by approximately 30%.

To compare stable and unstable cases, we compute the local time-averaged permeate flux $\langle j_v \rangle (x)$ and solute concentration on the membrane surface $\langle c_m \rangle (x)$,

$$\langle j_v \rangle (x) = \frac{1}{\Delta t_s} \int_{t_0}^{t_0 + \Delta t_s} j_v(x, t)\, dt\,, \quad \langle c_m \rangle (x) = \frac{1}{\Delta t_s} \int_{t_0}^{t_0 + \Delta t_s} c(x, H, t)\, dt\,, \tag{59}$$

where Δt_s is a sampling interval taken sufficiently large to achieve a statistically steady state, and t_0 is a start time greater than the initial transient. We also compute the average permeate flux over the full channel length,

$$J_{ave} = \frac{1}{L} \int_0^L \langle j_v \rangle\, dx. \tag{60}$$

In the following, because the variation of the temperature over the membrane surface closely follows that of the vapor flux, we do not show temperature profiles along the membrane.

Fig. 8(*a*) shows the downstream variation of $\langle j_v \rangle$ for the stable (dashed line) and unstable (solid line) cases shown in Figs. 5–7. For the stable case, $\langle j_v \rangle$ decreases monotonically downstream. The unstable case shows a similar decrease up to $x = 3.62$ cm (open circle), after which $\langle j_v \rangle$ begins increasing. We hereinafter refer to the x-location where $\langle j_v \rangle$ begins increasing as the *onset distance* x_{on}. Downstream of x_{on}, the flux $\langle j_v \rangle$ for the unstable case reaches a local maximum $j_{max} = 29.7$ kg/(m^{2}h) (solid dot), and then tends to decrease downstream. One potential explanation for this behavior is that for $x < x_{on}$, perturbation growth is dominated by linear instability mechanisms that have no net impact on local permeate production. Downstream of x_{on}, the onset of nonlinear mechanisms triggers the increase in $\langle j_v \rangle$. However, this also increases the rate of energy lost to evaporation, causing the instability to quench downstream. This interpretation is incomplete, as it does not explain the secondary maximum in $\langle j_v \rangle$ near $x = 12.3$ cm. Overall, the unstable case produces the average permeate flux $J_{ave} = 20.5$ kg/(m^{2}h). This is a 21% relative increase compared to $J_{ave} = 16.9$ kg/(m^{2}h) produced by the stable case.

Fig. 8(*b*) shows the corresponding results for $\langle c_m \rangle$. The open circle marks x_{on}. The increase in vapor flux at x_{on} generates a corresponding increase in $\langle c_m \rangle$, followed by a downstream maximum of $c_{max} = 177$ g/l at $x = 5.06$ cm. Note from Fig. 8 that in the linear regime $x < x_{on}$, the gravitational orientation has little impact on $\langle j_v \rangle$, but significantly

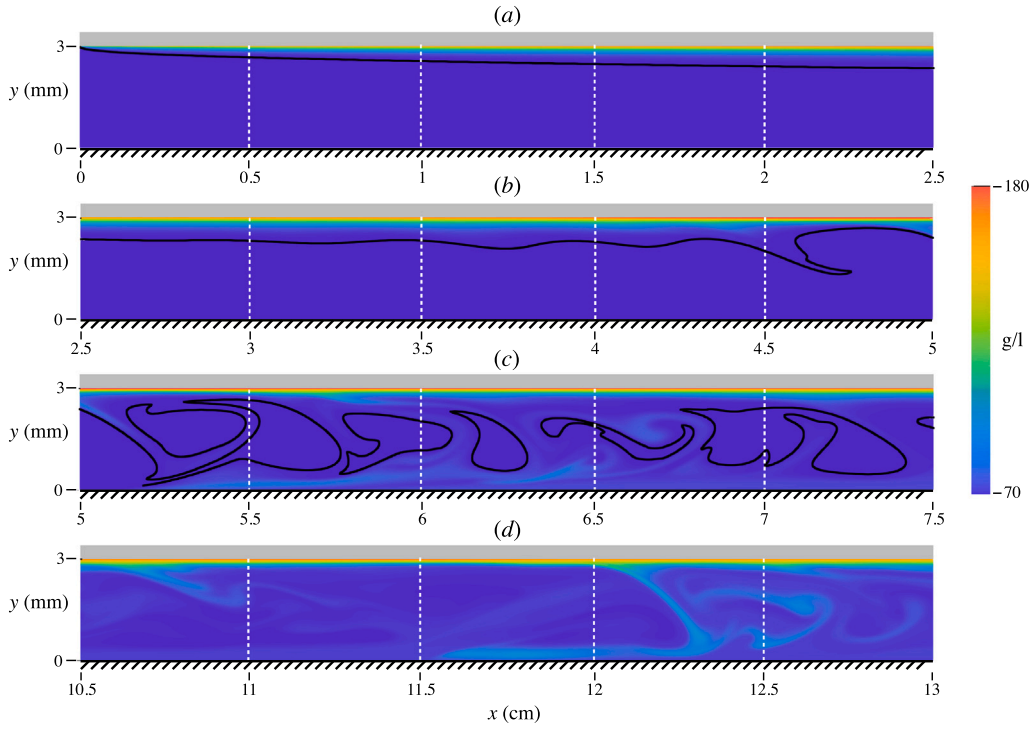

Fig. 6. Instantaneous snapshots of the concentration field when $Re = 75$ and $\mathbf{g} = [0, -9.81]$ [m/s^2] (unstable configuration). Results are shown in the ranges $x \in [0, 2.5]$ cm (panel *a*), $x \in [2.5, 5]$ cm (panel *b*), $x \in [5, 7.5]$ cm (panel *c*), and $x \in [10.5, 13]$ cm (panel *d*). The black lines show where $c = 1.01\, C_{\text{in}}$.

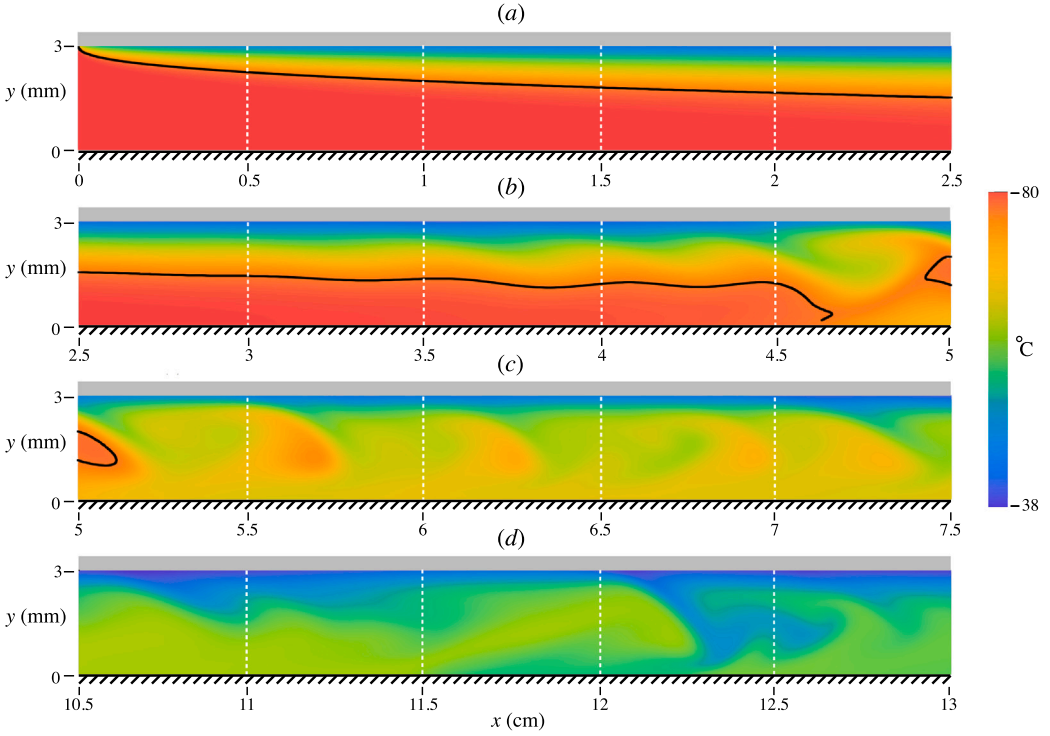

Fig. 7. Instantaneous snapshots of the temperature field when $Re = 75$ and $\mathbf{g} = [0, -9.81]$ [m/s^2] (unstable configuration). Results are shown in the ranges $x \in [0, 2.5]$ cm (panel *a*), $x \in [2.5, 5]$ cm (panel *b*), $x \in [5, 7.5]$ cm (panel *c*), and $x \in [10.5, 13]$ cm (panel *d*). The black lines show where $T = 0.99\, T_{\text{in}}$.

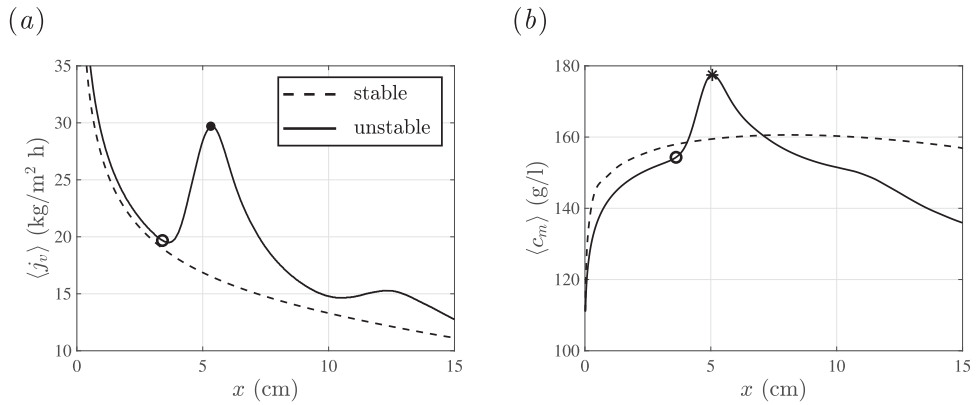

Fig. 8. Downstream variation of $\langle j_v \rangle$ (panel *a*) and $\langle c_m \rangle$ (panel *b*) when $Re = 75$ for the stable (dashed line) and unstable (solid line) cases. The open circles mark x_{on}. The solid dot marks the local maximum j_{max}. The asterisk marks c_{max}.

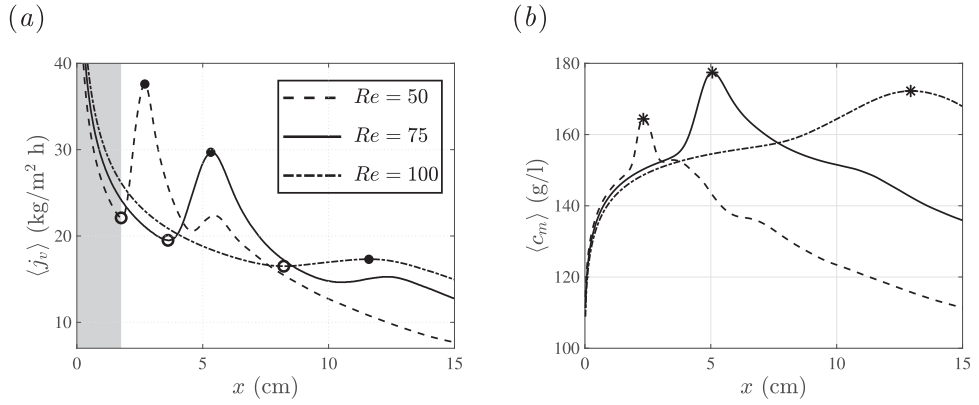

Fig. 9. Downstream variation of $\langle j_v \rangle$ (panel *a*) and $\langle c_m \rangle$ (panel *b*) when $Re = 50$ (dashed line), 75 (solid line), and 100 (dash-dotted line). In panel (*a*), the open circles mark x_{on}, and the solid dots mark the local maxima j_{max}. In panel (*b*), the asterisks mark the maximum concentration c_{max}.

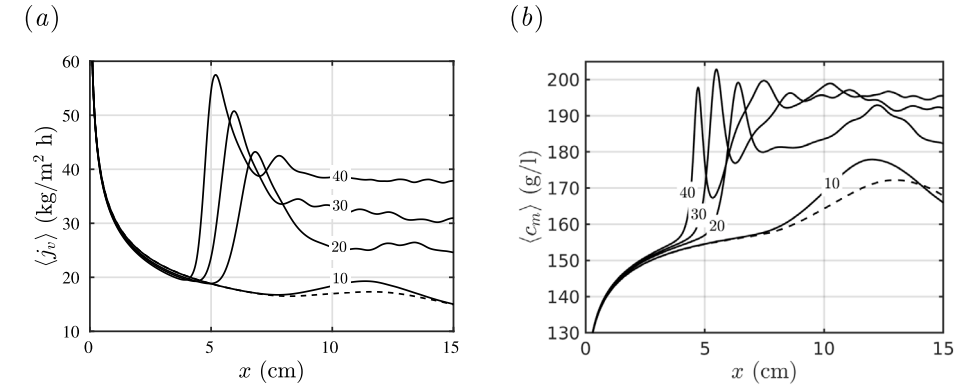

Fig. 10. Downstream variation of $\langle j_v \rangle$ (panel *a*) and $\langle c_m \rangle$ (panel *b*) when $Re = 100$ and $\dot{q}_s = 0$ (dashed line), 10, 20, 30, and 40 kW/m^2. The heat values are labeled in both panels.

impacts $\langle c_m \rangle$. This is because the feed saturation pressure p_{sat} varies exponentially with temperature, but only weakly with concentration c_m.

5.2. Impact of inlet Reynolds number

To investigate the impact of the Reynolds number, we perform simulations with $Re = 50$, 75, and 100 in the unstable orientation. Fig. 9(*a*) shows $\langle j_v \rangle$ for each case (see legend). Increasing Re delays the onset distance (open circles) from $x_{on} = 1.76$ for $Re = 50$ to $x_{on} = 8.22$ for $Re = 100$, and also decreases the local maxima j_{max} (solid dots). However, in the upstream region shaded gray, increasing Re increases vapor production. This occurs because in the absence of convection, increasing Re (by increasing U_{in}) decreases the thermal boundary layer thickness and mitigates temperature polarization [20]. Increasing Re also increases vapor production in the downstream region $x > 10$, because quenching is delayed downstream. The simulations in Fig. 9(*a*) produce $J_{ave} = 18.6$, 20.5, and 20.3, kg/(m^2 h) for $Re = 50$, 75, and 100, respectively.

Fig. 9(*b*) shows the corresponding results for $\langle c_m \rangle$. Though $Re = 50$ produces the largest j_{max}, it produces the smallest c_{max}. This might be explained by the fact that j_{max} occurs further upstream for $Re = 50$, where the solutal boundary layer is less developed. Overall, Fig. 9(*a*) suggests there is an optimal Re that maximizes vapor, while Fig. 9(*b*)

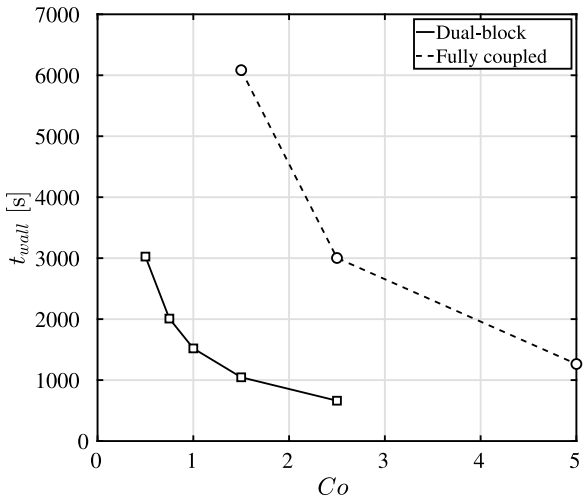

Fig. 11. Wall time vs. Co for the dual-block and the coupled algorithm.

suggests there is a separate Reynolds number that minimizes c_{max}, and consequently the risk of mineral scaling.

5.3. Sustaining convection through wall heating

Convection can be strengthened and sustained over long membranes by heating the channel wall opposite the membrane. This generates rising plumes of warm fluid that replenish heat lost to evaporation. For demonstration, we set $\mathbf{g} = [0, -9.81]$ [m/s^2] (unstable orientation) and $Re = 100$, and perform five simulations in which we deliver $\dot{q}_s = 0$, 10, 20, 30, and 40 kW/m^2 to the channel wall, using condition (19). Fig. 10 shows the resulting downstream variation of $\langle j_v \rangle$ (panel *a*) and $\langle c_m \rangle$ (panel *b*). Results for $\dot{q}_s = 0$ are shown as a dashed-line. Wall heating moves x_{on} upstream and increases j_{max}. Moreover, for $\dot{q}_s \geq 20$, the downstream vapor production stabilizes to non-zero values. However, panel (*b*) shows that wall heating also increases concentration polarization, and risk of mineral scaling. Future study is required to determine what operating conditions limit concentration polarization to acceptable values.

5.4. Numerical stability

To decouple the solution of the concentration and energy equations from the mass and momentum equations, our method uses temporal extrapolation in Eqs. (24), (26), and (27). Our method is consequentially less stable than those that iterates through the full governing equations in a single block. To explore the trade-off between stability and speed, we run a series of simulations at different Courant numbers, and compute the wall time t_{wall} (total time required by the computation using a standard clock) required to simulate 100 s. We consider the case described in Section 5.2 for $Re = 50$, and we repeat the simulations using our dual-block method and a fully coupled algorithm detailed in the Supplementary Material. The simulations are run on 16 physical cores of a workstation mounting two 22-Core Intel Xeon E5-2699 v4 and running on OpenSUSE Tumbleweed.

Fig. 11 reports the wall time for different Co. The dual-block algorithm (dashed line) is 5–6 times faster than the fully coupled algorithm. However, for the problem tested, the dual-block algorithm could not be pushed beyond $Co = 2.5$, even when we added under-relaxation. Meanwhile, using a relaxation factor, the coupled algorithm could be pushed beyond $Co = 5$. In this manner, the fully coupled algorithm can achieve similar performance to the dual-bloc algorithm by increasing Co beyond 2.5. However, this comes at the expense of reduced temporal resolution. When the loss of accuracy is not relevant, such as for steady-state cases, the fully coupled algorithm is a viable choice for simulating VMD. In the unstable orientation, however, the dual-block method is more attractive for resolving the dynamics of buoyancy-driven instabilities.

5.5. 3D example

As a final demonstration of our method, we perform a 3D simulation at $Re = 50$ for the conditions described in Section 5.2, but in a 3D domain $H \times L \times 4H$, where the last length corresponds to the spanwise width in the z-direction. We set periodic boundary conditions at $z = 0$ and $z = 4H$. To reduce the computational time and data storage, we use a grid with $N_b = 50$, and the spanwise direction is discretized using 100 cells.

Fig. 12 shows the instantaneous temperature (panel *a*) and concentration (panel *b*) fields at the membrane and four downstream cross-sections at $t = 300$ s. The cross-sections reveal that the onset of instability has a spanwise wavelength comparable to H. Downstream, the plumes coarsen with an increasing wavelength. This coarsening may be related to the downstream quenching of the instability. The coarsening is more evident in Fig. 13(*a*), where we show isosurfaces of $T - \langle T \rangle$ at $t = 300$ s, where $\langle T \rangle$ is the temporal average of T. Fig. 13 (*b*) similarly shows isosurfaces of $c - \langle c \rangle$, where $\langle c \rangle$ is the temporal average of c. Plumes correspond to regions with positive concentration difference and negative temperature difference. A more thorough investigation of 3D instabilities is left to future work. We note that compared to the 2D simulation in Section 5.2, the 3D simulation predicts a permeate flux $J_{ave} = 24.2$ (kg/m^{2}h). This is a 43% increase in J_{ave} compared to that observed in our 2D simulations, indicating that the spanwise modes enhance permeate production.

6. Conclusions

We present a novel method of simulating variable density flows in membrane desalination systems. The method leverages the fact that the feed density varies primarily with the temperature and concentration, but not pressure. The method is based on a dual-block algorithm that solves all governing equations in conservation form and is formally second-order accurate in time and space. We verify the accuracy using manufactured solutions, and demonstrated that the proposed algorithm is significantly faster than fully-coupled schemes.

To demonstrate our method, we simulate buoyancy-driven convection in vacuum membrane distillation (VMD). The convection arises because the temperature and concentration fields trigger a double-diffusive instability in which plumes of dense feed sink away from the membrane surface. This brings warm low-concentration feed to the membrane, and increases vapor production. Our simulations suggest that buoyancy-driven convection generates three distinct downstream regions in a VMD channel. First, there is an initial distance $0 \leq x < x_{on}$ in which the local vapor flux decreases with downstream distance. Second, at the onset distance $x = x_{on}$, the onset of nonlinear mechanisms cause the local vapor flux $\langle j_v \rangle$ to increase rapidly, reaching a local maximum j_{max}. Third, downstream of j_{max}, the local vapor flux begins decreasing as the instability quenches due to energy lost to evaporation. For the operating conditions considered, we found that increasing the Reynolds number inhibits buoyancy-driven convection, such that x_{on} is pushed downstream and j_{max} decreases. We show, however, that convection can be strengthened and sustained over long membranes by heating the channel wall opposite the membrane.

Finally, we performed a 3D simulation of buoyancy-driven convection in VMD. This showed that instability in a real-world system likely occurs to 3D modes of instability, that are less stable than 2D modes. The simulation found that transport is further enhanced in the spanwise direction, producing a larger increase in vapor production that observed in 2D simulations. Future work is now required to expand our simulations of VMD in 3D, and determine optimal operating conditions

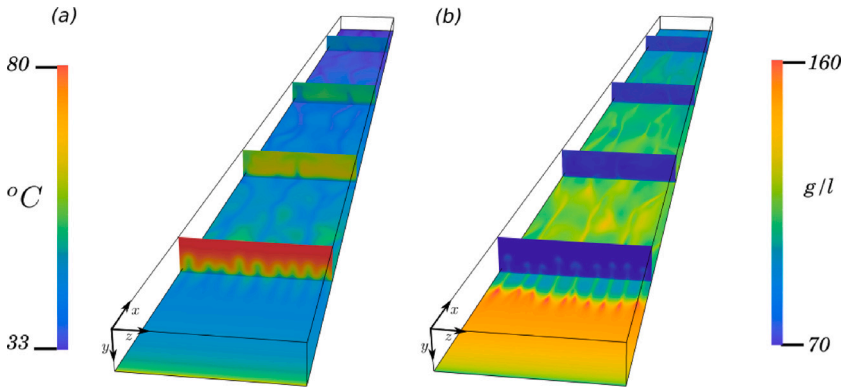

Fig. 12. Instantaneous temperature (panel *a*) and concentration (panel *b*) fields in a 3D simulation at $Re = 50$. The channel is flipped on the y axis for visualization purposes.

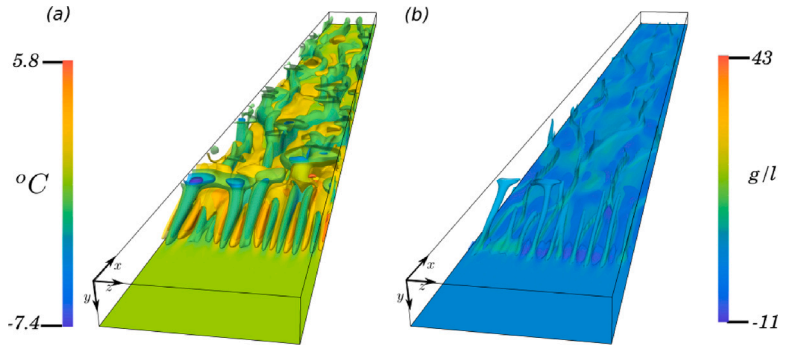

Fig. 13. Difference between the instantaneous and mean temperature (panel *a*) and concentration (panel *b*) in a 3D simulation at $Re = 50$. The channel is flipped on the y axis for visualization purposes.

and wall heating that maximize vapor production while mitigating the risk of mineral scaling. More broadly, our method opens the door to investigating variable density flows in temperature and pressure-driven membrane filtration processes with applications in the biofuel, medical, pharmaceutical, chemical, food, and dairy industries.

CRediT authorship contribution statement

Federico Municchi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Yiming Liu:** Investigation, Data curation. **Jingbo Wang:** Investigation, Data curation. **Tzahi Y. Cath:** Writing – review & editing, Resources, Investigation, Funding acquisition, Data curation, Conceptualization. **Craig S. Turchi:** Funding acquisition, Data curation, Conceptualization. **Michael B. Heeley:** Funding acquisition, Data curation, Conceptualization. **Eric M.V. Hoek:** Funding acquisition, Data curation, Conceptualization. **David Jassby:** Supervision, Project administration, Funding acquisition, Data curation, Conceptualization. **Nils Tilton:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

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

Acknowledgments

This work was generously supported by the U.S. Department of Energy (Award No. DE-EE0008391) and the National Science Foundation, USA (grant numbers 1752531 and 2306329). This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compfluid.2024.106449.

Data availability

Data will be made available on request,.

References

[1] Alkhudhiri A, Darwish N, Hilal N. Membrane distillation: A comprehensive review. Desalination 2012;287:2–18.
[2] Hickenbottom KL, Cath TY. Sustainable operation of membrane distillation for enhancement of mineral recovery from hypersaline solutions. J Membr Sci 2014;454:426–35, URL https://www.sciencedirect.com/science/article/pii/S0376738813009964.
[3] Bouchrit R, Boubakri A, Hafiane A, Bouguecha SA-T. Direct contact membrane distillation: Capability to treat hyper-saline solution. Desalination 2015;376:117–29, URL https://www.sciencedirect.com/science/article/pii/S0011916415300515.
[4] Fimbres-Weihs G, Wiley D. Review of 3D CFD modeling of flow and mass transfer in narrow spacer-filled channels in membrane modules. Chem Eng Process: Process Intensif 2010;49(7):759–81.

[5] Keir G, Jegatheesan V. A review of computational fluid dynamics applications in pressure-driven membrane filtration. Rev Environ Sci Biotechnol Vol 2014;13:183–201.

[6] Lou J, Johnston J, Tilton N. Application of projection and immersed boundary methods to simulating heat and mass transport in membrane distillation. Comput & Fluids 2020;212.

[7] Johnston J, Lou J, Tilton N. Application of projection methods to simulating mass transport in reverse osmosis systems. Comput & Fluids 2022;232(September 2021):105189.

[8] Knikker R. A comparative study of high-order variable-property segregated algorithms for unsteady low Mach number flows. Internat J Numer Methods Fluids 2011;66(4):403–27, URL https://onlinelibrary.wiley.com/doi/10.1002/fld.2261.

[9] Sewall EA, Tafti DK. A time-accurate variable property algorithm for calculating flows with large temperature variations. Comput & Fluids 2008;37(1):51–63, URL https://linkinghub.elsevier.com/retrieve/pii/S0045793007000904.

[10] Schofield RW, Fane AG, Fell CJ. Heat and mass transfer in membrane distillation. J Membr Sci 1987;33(3):299–313.

[11] Lawson KW, Lloyd DR. Membrane distillation. J Membr Sci 1997;124(1):1–25.

[12] Chen X, Yip NY. Unlocking high-salinity desalination with cascading osmotically mediated reverse osmosis: Energy and operating pressure analysis. Environ Sci Technol 2018;52(4):2242–50.

[13] Gryta M, Tomaszewska M, Morawski A. Membrane distillation with laminar flow. Sep Purif Technol 1997;11(2):93–101, URL https://linkinghub.elsevier.com/retrieve/pii/S1383586697000026.

[14] Bouguecha S, Chouikh R, Dhahbi M. Numerical study of the coupled heat and mass transfer in membrane distillation. Desalination 2003;152(1–3):245–52, URL https://linkinghub.elsevier.com/retrieve/pii/S0011916402010706.

[15] Tan YZ, Han L, Chow WH, Fane AG, Chew JW. Influence of module orientation and geometry in the membrane distillation of oily seawater. Desalination 2017;423:111–23, URL https://linkinghub.elsevier.com/retrieve/pii/S0011916417318519.

[16] Caspar J, Xue G, Asiri J, Oztekin A. The effect of buoyancy-driven convection in vacuum membrane distillation module. ASME international mechanical engineering congress and exposition, Volume 8: Fluids Engineering; Heat Transfer and Thermal Engineering, 2022, V008T10A031.

[17] Bird RB, Stewart WE, Lightfoot EN. Transport phenomena. revised 2nd ed.. 2006, p. 928.

[18] Wang P, Anderko A, Young R. A speciation-based model for mixed-solvent electrolyte systems. Fluid Phase Equilib 2002;203:141–76.

[19] Jamieson D, Tudhope J. Physical properties of sea water solutions: thermal conductivity. Desalination 1970;8(3):393–401, URL https://linkinghub.elsevier.com/retrieve/pii/S0011916400802404.

[20] Lou J, Vanneste J, DeCaluwe SC, Cath TY, Tilton N. Computational fluid dynamics simulations of polarization phenomena in direct contact membrane distillation. J Membr Sci 2019;591(June):117150.

[21] Vanneste J, Bush JA, Hickenbottom KL, Marks CA, Jassby D, Turchi CS, Cath TY. Novel thermal efficiency-based model for determination of thermal conductivity of membrane distillation membranes. J Membr Sci 2018;548:298–308.

[22] Yu H, Yang X, Wang R, Fane AG. Numerical simulation of heat and mass transfer in direct membrane distillation in a hollow fiber module with laminar flow. J Membr Sci 2011;384(1–2):107–16.

[23] Hitsov I, Maere T, De Sitter K, Dotremont C, Nopens I. Modelling approaches in membrane distillation: A critical review. Sep Purif Technol 2015;142:48–64.

[24] Day MA. The no-slip condition of fluid dynamics. Erkenntnis 1990;33(3):285–96.

[25] Anqi AE, Usta M, Krysko R, Lee J-G, Ghaffour N, Oztekin A. Numerical study of desalination by vacuum membrane distillation – transient three-dimensional analysis. J Membr Sci 2020;596(5):117609, URL https://linkinghub.elsevier.com/retrieve/pii/S0376738819305319.

[26] van Leer B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. J Comput Phys 1979;32:101–36.

[27] Gear CW. Numerical initial value problems in ordinary differential equations. Englewood Chiffs, N.J.: Prentice Hall; 1971.

[28] Desjardins O, Blanquart G, Balarac G, Pitsch H. High order conservative finite difference scheme for variable density low Mach number turbulent flows. J Comput Phys 2008;227(15):7125–59, URL https://linkinghub.elsevier.com/retrieve/pii/S0021999108001666.

[29] McMurtry PA, Jou W-H, Riley J, Metcalfe RW. Direct numerical simulations of a reacting mixing layer with chemical heat release. AIAA J 1986;24(6):962–70, URL https://arc.aiaa.org/doi/10.2514/3.9371.

[30] Lessani B, Papalexandris MV. Time-accurate calculation of variable density flows with strong temperature gradients and combustion. J Comput Phys 2006;212(1):218–46, URL https://linkinghub.elsevier.com/retrieve/pii/S0021999105003189.

[31] Wall C, Pierce CD, Moin P. A semi-implicit method for resolution of acoustic waves in low Mach number flows. J Comput Phys 2002;181(2):545–63, URL https://linkinghub.elsevier.com/retrieve/pii/S002199910297141X.

[32] Boscheri W, Pareschi L. High order pressure-based semi-implicit IMEX schemes for the 3D Navier–Stokes equations at all Mach numbers. J Comput Phys 2021;434:110206, URL https://linkinghub.elsevier.com/retrieve/pii/S0021999121001017.

[33] Dumbser M, Casulli V. A conservative, weakly nonlinear semi-implicit finite volume scheme for the compressible Navier–Stokes equations with general equation of state. Appl Math Comput 2016;272:479–97, URL https://linkinghub.elsevier.com/retrieve/pii/S0096300315010978.

[34] Busto S, Tavelli M, Boscheri W, Dumbser M. Efficient high order accurate staggered semi-implicit discontinuous Galerkin methods for natural convection problems. Comput & Fluids 2020;198:104399, URL https://linkinghub.elsevier.com/retrieve/pii/S0045793019303573.

[35] Issa R. Solution of the implicitly discretised fluid flow equations by operator-splitting. J Comput Phys 1986;62(1):40–65, URL https://linkinghub.elsevier.com/retrieve/pii/0021999186900999.

[36] Ni M, Abdou M. A bridge between projection methods and SIMPLE type methods for incompressible Navier–Stokes equations. Internat J Numer Methods Engrg 2007;72(12):1490–512.

[37] Shashkov M, Swartz B, Wendroff B. Local reconstruction of a vector field from its normal components on the faces of grid cells. J Comput Phys 1998;139:406–9.

[38] Weller H, Shahrokhi A. Curl-free pressure gradients over orography in a solution of the fully compressible euler equations with implicit treatment of acoustic and gravity waves. Mon Weather Rev 2014;142:4439–57.

[39] Weller H. Non-orthogonal version of the arbitrary polygonal C-grid and a new diamond grid. Geosci Model Dev 2014;7:779–97.

[40] Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA J 1983;21(11):1525–32, URL https://arc.aiaa.org/doi/10.2514/3.8284.

[41] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 1998;12(6):620, URL http://scitation.aip.org/content/aip/journal/cip/12/6/10.1063/1.168744.

[42] Roache PJ. Code verification by the method of manufactured solutions. J Fluids Eng Trans ASME 2002;124(1):4–10.

[43] Lee SB. A study on temporal accuracy of OpenFOAM. Int J Nav Archit Ocean Eng 2017;9(4):429–38, Publisher: Elsevier Ltd.

[44] Matsuura T. Synthetic membranes and membrane separation processes. CRC Press; 2020, URL https://www.taylorfrancis.com/books/9781000102901.

[45] Liu Y, Wang J, Hoek EM, Municchi F, Tilton N, Cath TY, Turchi CS, Heeley MB, Jassby D. Multistage surface-heated vacuum membrane distillation process enables high water recovery and excellent heat utilization: A modeling study. Environ Sci Technol 2023;57(1):643–54.

[46] Venier CM, Pairetti CI, Damian SM, Nigro NM. On the stability analysis of the PISO algorithm on collocated grids. Comput & Fluids 2017;147:25–40.

[47] Martinand D, Carrière P, Monkewitz PA. Three-dimensional global instability modes associated with a localized hot spot in Rayleigh–Bénard–Poiseuille convection. J Fluid Mech 2006;551(-1):275.

[48] Carrière P, Monkewitx PA, Martinand D. Envelope equations for the RayleighBnardPoiseuille system. Part 1. Spatially homogeneous case. J Fluid Mech 2004;502:153–74.

[49] Carrière P, Monkewitx PA. Convective versus absolute instability in mixed Rayleigh–Bénard–Poiseuille convection. J Fluid Mech 1999;384:243–62.

[50] Fujimura K, Kelly RE. Interaction between longitudinal convection rolls and transverse waves in unstably stratified plane Poiseuille flow. Phys Fluids 1995;7(1):68–79, URL https://pubs.aip.org/pof/article/7/1/68/439997/Interaction-between-longitudinal-convection-rolls.

[51] Kelly R. The onset and development of thermal convection in fully developed shear flows. 1994, p. 35–112. http://dx.doi.org/10.1016/S0065-2156(08)70255-2, URL https://linkinghub.elsevier.com/retrieve/pii/S0065215608702552.