Contents lists available at ScienceDirect

Computer Physics Communications

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

Computer Programs in Physics

openFuelCell2: A new computational tool for fuel cells, electrolyzers, and other electrochemical devices and processes[☆,☆☆]

Shidong Zhang[a,1], Steffen Hess[a], Holger Marschall[b], Uwe Reimer[a], Steven Beale[a,c,*], Werner Lehnert[a,d]

[a] Institute of Energy and Climate Research, IEK-14, Forschungszentrum Jülich, 52425 Jülich, Germany
[b] Center of Smart Interfaces, Technische Universität Darmstadt, Alarich-Weiss Straße 10, 64287 Darmstadt, Germany
[c] Department of Mechanical and Materials Engineering, Queen's University, Kingston ON K7L 3N6, Canada
[d] Modeling in Electrochemical Process Engineering, RWTH Aachen University, Aachen, 52056, Germany

ARTICLE INFO

Keywords:
Computational fluid dynamics
Open-source
Fuel cell
Electrolyzer
Multi-region
Multi-physics
OpenFOAM

ABSTRACT

Fuel cells/electrolyzers are efficient and clean electrochemical devices that convert chemical energy directly into electricity and vice versa. They have attracted sustainable attention over the past decade from multiple experimental and numerical studies. However, detailed experimental investigations are typically expensive and challenging for providing a number of operating conditions and designs. Computational analysis offers an alternative approach for these studies. With the steadily increasing high-performance computing resources available, the limitations of numerical simulations have substantially decreased. This contribution details the design choice and code structure of modern electrochemical devices, which have been implemented as a versatile C++ library named `openFuelCell2` within the open-source platform `OpenFOAM`, allowing for large-scale parallel calculations to be performed. The solver considers the major transport phenomena in a typical electrochemical device, including fluid flow, heat and mass transfer, species and charge transfer, and electrochemical reaction. This enables numerical simulations on popular electrochemical devices, such as fuel cells and electrolyzers, to be conducted. The paper also describes the domain decomposition, and parallel performance issues, as well as future applications.

Program summary
Program Title: `openFuelCell2`
CPC Library link to program files: https://doi.org/10.17632/cvmb5xgy7h.1
Developer's repository link: https://github.com/openFuelCell2/openFuelCell2
Licensing provisions: GPLv3
Programming language: C++
Journal reference of previous version: S.B. Beale, H.-W. Choi, J.G. Pharoah, H.K. Roth, H. Jasak, D.H. Jeon, Open-source computational model of a solid oxide fuel cell, Computer Physics Communications 200 (2016) 15–26.
Does the new version supersede the previous version?: No
Reasons for new version: This new version aims to account for additional electrochemical applications.
Summary of revisions: The new version employs new code structure designs and is capable of considering two different electric potential fields and two-phase flows.
Nature of problem: This software library provides a set of models for simulating electrochemical devices, such as fuel cells and electrolyzers, as well as other similar devices. These systems consist of multiple components with distinct features but are coupled through properties such as temperature, species concentrations, and electric current/potentials. This software utilizes a computational fluid dynamics (CFD) approach to enable the simulation of multiregion and multiphysics problems, accounting for heat and mass transfer, single and two-

[☆] The review of this paper was arranged by Prof. Andrew Hazel.
[☆☆] This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).
[*] Corresponding author at: Institute of Energy and Climate Research, IEK-14, Forschungszentrum Jülich, 52425 Jülich, Germany.
E-mail address: s.beale@fz-juelich.de (S. Beale).
[1] Present address at: Institute of Energy and Climate Research, IEK-9, Forschungszentrum Jülich, 52425 Jülich, Germany.

https://doi.org/10.1016/j.cpc.2024.109092
Received 26 June 2023; Received in revised form 8 December 2023; Accepted 9 January 2024
Available online 19 January 2024

phase flow, multicomponent diffusion, electron and ion transfer, and electrochemical reactions. This toolbox provides researchers with a powerful tool for studying electrochemical devices in a comprehensive and efficient manner.

Solution method: This software utilizes the finite volume method (FVM) to discretize and solve all conserved variables governed by partial differential equations. It employs a variant of the SIMPLE and PISO algorithms, known as PIMPLE, to solve the pressure-velocity coupling. All of the transport equations are solved sequentially, except for the enthalpy equation which is solved in an implicitly coupled manner within the global domain.

Additional comments including restrictions and unusual features:

- The two-phase flow consists of continuous and dispersed phases, without resolving the phase interfaces.
- The two-phase flow is solved with an Eulerian-Eulerian approach.
- A local-time-stepping method was typically used in the two-phase solution.
- The electrochemical reactions take place in volumetric regions.
- Species diffusion is described by Fick's law.

1. Introduction

Green hydrogen represents a viable pathway for the defossilization of the energy, chemical, and transportation sectors. The growing concern over climate change and increasing demand for energy has led to the development of renewable and clean energy technologies, such as fuel cells and electrolyzers. These electrochemical devices convert chemical energy directly into electricity or vice versa, offering a promising alternative to conventional fossil fuel-based technologies. However, designing and optimizing such devices typically involves complex multi-physics phenomena, which are difficult and expensive to study experimentally. To address this challenge, numerical simulations have proved a powerful tool for predicting the behavior and performance of these devices [1–3]. In recent years, advances in computational techniques, hardware, and software have significantly improved the accuracy and efficiency of numerical simulations, making them an indispensable tool for developing and optimizing renewable and clean energy technologies.

As an alternative approach to experimental measurements, numerical analysis techniques, such as the finite difference method (FDM), finite volume method (FVM), and finite element method (FEM), are attracting attention for the investigation and understanding of the physical mechanism in many industrial and academic applications. Among these, multi-physics in multiple regions are usually involved. The comprehensive consideration of them turns out to be a complex and challenging endeavor. Nowadays, the implementation of numerical models for this purpose has primarily been limited to commercial software or in-house designed code packages. For example, various commercial software packages, such as PHOENICS, ANSYS Fluent, Star-CCM+, and COMSOL Multiphysics have been adapted to fuel cell and electrolysis investigations [4–10], in single repeating unit or stacks consisting of numerous repeating units. Meanwhile, there are limited open-source packages to enable the simulations of the typical electrochemical devices to be considered [11–13]. The book [14] edited by Beale and Lehnert introduces electrochemical calculations using the open-source library, `OpenFOAM` (Open Field Operation and Manipulation), which range from micro- to cell- and stack-scale models, with numerous illustrations and programming examples. The `openFuelCell` suite was first reported in the present journal by Beale et al. [15]. The further development of `openFuelCell` by the present authors has witnessed research activities in different electrochemical devices [3], i.e., solid oxide fuel cell (SOFC) [15–18], solid oxide electrolyzer [19–21], and proton exchange membrane fuel cell (PEMFC) [22–25]. They represent work that has been conducted and published during the development phase of the present particular framework. More recently, Zhao et al. [26] also constructed a framework considering SOFCs based on `OpenFOAM`. Unlike commercial software, open-source packages offer several advantages [1], which are essential for promoting collaboration, reproducibility, and innovation in computational research,

1. Transparency: The source code and data files are accessible by other researchers.
2. Implementation: The new features or mathematical models can be easily implemented.
3. Sharing: It is simple to work on common projects with others in the research community.
4. Validation: Experimental corroboration of results with open data sets.
5. Interoperability: Coupling with other programs is possible. For instance, preCICE [27] offers the possibility of pairing `OpenFOAM` with other open-source programs, such as Calculix [28], deal.II [29], or other solvers.

The user community of `OpenFOAM` has grown in size and application since the initial release in 2004. The development of `OpenFOAM` employed the object-oriented programming language, C++, for core concept design. The package provides easy-readable syntax that mimics conventional mathematical notations for partial differential equations, as well as a library for various physical modules. In addition, it offers an extensive framework for applications and implementations of new models. This can provide researchers with access to advanced simulation tools and allow for collaboration and community-driven innovation.

The present work is a substantially modified/improved code, based on `openFuelCell` [30]. The source code is deeply integrated with the multiphase system that was first introduced in `OpenFOAM` version 6, to account for two-phase liquid-gas flow options. A branch of modules used to describe electrochemical reaction systems are also implemented to simulate electrical currents and potentials. The new code is able to predict the performance of numerous electrochemical devices for both low-temperature ($< 100\,^\circ$C) and high-temperature ($> 100\,^\circ$C) applications. It is also able to conduct other different numerical simulations, including conjugate heat transfer problems, combustion/chemical reactions in multiple regions, etc. These modules include,

- Flexible combination of different types of regions.
  - fluid regions
  - solid regions
  - electric regions
- Volumetric electrochemical reactions
  - electrochemical reaction class
  - Nernst potential class
  - activation overpotential class
- Two-phase Eulerian-Eulerian model
- Two-field electric potential model

This article is organized as follows: Section 2 presents the governing equations to be solved and the assumptions made during model development; Section 3 illustrates the design of the present software and the structure of the source code; in section 4, the individual components of the `openFuelCell2` software are presented; user guides/instructions

List of Symbols

Abbreviations

CFD	Computational fluid dynamics
CL	Contact layer
EOD	Electroosmotic drag
FL	Function layer
FVM	Finite volume method
GDL	Gas diffusion layer
HT	High temperature
LT	Low temperature
LTS	Local time stepping
MEA	Membrane electrode assembly
PEMFC	Proton exchange membrane fuel cell
PEMFC	Proton exchange membrane electrolyzer cell
PEN	Positive electrolyte negative
RH	Relative humidity
SOFC	Solid oxide fuel cell
SOEC	Solid oxide electrolyzer cell

Greek symbols

α	Phase saturation	[−]
α	Transfer coefficient	[−]
η	Activation overpotential	[V]
Γ	Thermal diffusivity	[m^2/s]
λ	Stoichiometric factor	[−]
λ	Water content	[mol/m^3]
μ	Dynamic viscosity	[kg/(m·s)]
ω	Reaction coefficient	[−]
φ	Electric field potential	[V]
ρ	Density	[kg/m^3]
σ	Electric conductivity	[S/m]
σ	Surface tension	[N/m]
θ	Contact angle	[rad]
ε	Porosity	[−]

Roman symbols

$\mathbf{g}$	Acceleration due to gravity	[m/s^2]
$\mathbf{i}$	Current density	[A/m^2]
$\mathbf{M}$	Interphase momentum transfer	[kg/(m^2·s^2)]
$\mathbf{q}$	Heat flux	[W/m^2]
$\mathbf{U}$	Mixture velocity	[m/s]
C	Concentration	[mol/m^3]
C_d	Drag coefficient	[−]
C_p	Specific heat	[J/(kg·K)]
D	Diffusion coefficient	[m^2/s]
F	Faraday's constant	[C/mol]
H	Enthalpy	[J/mol]
h	Specific enthalpy	[J/kg]
i_0	Exchange current density	[A/m^3]
J	Electric transfer source/sink	[A/m^3]
J	Leverett J function	[−]
j	Electrochemical reaction rate	[A/m^3]
K	Permeability	[m^2]
k	Thermal conductivity	[W/(m·K)]
K_r	Relative permeability	[−]
$\dot{m}$	Mass flow rate	[kg/s]
n_d	Electroosmotic drag coefficient	[−]
p	Pressure	[Pa]
Q	Heat source/sink	[W/m^3]
R	Mass source/sink	[kg/(m^3·s)]
R_g	Universal gas constant	[J/(K·mol)]
S	Entropy	[J/(mol·K)]
S_Darcy	Darcy's term	[kg/(m^2·s^2)]
T	Temperature	[K]
t	Time	[s]
Y	Mass fraction	[−]
z	Number of electrons transfer	[−]
EW	Equivalent weight	[mol/m^3]

Subscript symbols

ae	Air electrode
c	Capillary
d	Drag
E	Electrode
fe	Fuel electrode
g	Gas
i	Species index
int	Interface
I	Ionic
l	Liquid
m	Membrane
nC	Number (n) of used computational cores
ϕ	Phase index
r	Relative
s	Solid
sat	Saturation

Superscript symbols

$''$	Per unit area
β	Exponent of phase saturation
eff	Effective
ξ	Exponent of species concentration

can be found in Section 5; Section 6 displays selected simulation results with validation; Section 7 provides the reader with practical advice on how to perform numerical simulations. Finally, a summary and outlook for the future are presented in Section 8.

2. Theoretical background

Recent decades have witnessed the improvement and development of technology, applications, and fundamental research in the electrochemistry research community. Numerical models attracted much attention and gained momentum, ranging from simple one-dimensional (1-D) to fully coupled three-dimensional (3-D), microscopic to macroscopic scale, and single-phase flow to multi-phase flow [31,32,1,2,8,33,34]. Among these studies, water transport/management and multiphase flow are very important issues. One widely-used method refers to the volume of fluid (VOF) method [35], which enables phase interface(s) to be tracked. However, it is computationally expensive [36]. As an alternative, a two-phase Eulerian-Eulerian approach [37–43] may be applied to describe the fluid motion [44–46]. In such a case, liquid water is frequently assumed to be mist and treated as spherical droplets/bubbles. The size of these can be obtained from VOF/Lattice Boltzmann Method (LBM) simulations [47–52], and experimental measurements [53,54]. This simplifies the computational requirements and provides the possibility to couple electrochemical processes, species transport, and ion transfer.

2.1. Transport phenomena

An electrochemical device consists of a number of components/parts, including solid interconnects, gas diffusion layers or support lay-

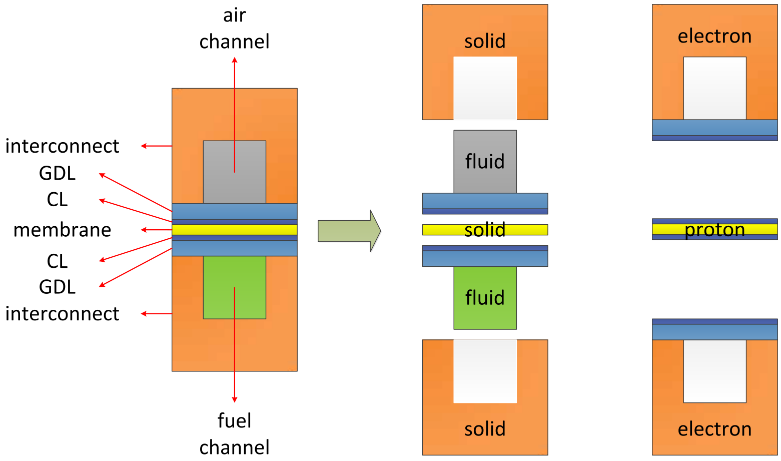


Fig. 1. Computational domains: global domain and local domains [55].

ers (GDLs/SLs), catalyst layers or functional layers (CLs/FLs), membranes/electrolytes, etc. A number of coupled physical transport processes take place simultaneously in these components. Fig. 1 displays a schematic view of an assembled standard repeating unit. Further designs can be found in the literature, e.g., designs without channels, but with expanded mesh as an alternative [56,20]. In order to account for the sub-processes within each component, a domain composed of global and sub-regions, sometimes referred to as 'parent' and 'child' meshes [15], is constructed. These are referred to below, as 'main' and 'branch' meshes. Based on the type of transport processes, the global domain is divided into three sub-types (branch types): namely, fluid, solid, and electric regions.

In this code, the overall temperature is solved in the global region. The solid regions contain the material properties, whereas the fluid regions address specific processes regarding fluid flow, species transport, and/or heat transfer. In the electric regions, ionic and electronic charge transfer are taken into account; additionally, the transfer of water dissolved in the membrane is also considered in the proton-conducting electric region(s) e.g., in PEMFCs. Transport phenomena in all of the local domains are coupled and solved simultaneously. The physical phenomena are presumed to be governed by a set of coupled equations based on the following assumptions.

2.1.1. *Assumptions*

- All gas and liquid phase flows are in the laminar regime.
- Phases (gases and/or liquids) are incompressible and gas mixtures obey the ideal gas law.
- Fick's law may be used for multi-component species transfer.
- The properties of the porous materials are treated as homogeneous.
- The membrane is only permeable to water. No crossover of gases is considered for the membrane/electrolyte.
- Produced water is in liquid form at low operating temperatures and in vapor form if the temperature is above 100 ^{o}C (ambient pressure).
- The gases dissolved in liquid water are not considered.
- Dispersed phases in the gas channels (if present) are treated as if composed of spherical droplets/bubbles.

2.1.2. *Fluid region*

Continuity and momentum equations in flow channels

In the Eulerian-Eulerian two-phase flow algorithm, the phases are typically considered separately, with one phase treated as the 'continuous' phase and the other as 'dispersed'. The model is also referred to as a 'two-fluid' or 'multi-fluid' model in the PEMFC community, and has been described in detail by previous researchers [32,2]. Given the assumptions described in the present study and reasonable averaging techniques, the conditionally averaged continuity and momentum equations are as follows,

$$\frac{\partial(\rho_\phi \alpha_\phi)}{\partial t} + \nabla \cdot (\alpha_\phi \rho_\phi \mathbf{U}_\phi) = R_\phi \tag{1}$$

$$\frac{\partial(\rho_\phi \alpha_\phi \mathbf{U}_\phi)}{\partial t} + \nabla \cdot (\alpha_\phi \rho_\phi \mathbf{U}_\phi \mathbf{U}_\phi) + \nabla \cdot (\alpha_\phi \tau^\phi) = -\alpha_\phi \nabla p + \alpha_\phi \rho_\phi \mathbf{g} + \mathbf{M}_\phi \tag{2}$$

where the subscript ϕ indicates liquid and gas phases; α denotes the phase volume fraction, ρ represents the mixture/pure phase density, $\mathbf{U}$ is the phase velocity, R means the mass source/sink, p is the shared pressure for both phases, and $\mathbf{g}$ refers to gravity. The term τ is expressed in this study as,

$$\tau^\phi = \mu_\phi \nabla \mathbf{U}_\phi \tag{3}$$

μ_ϕ is the dynamic viscosity of phase ϕ. The term $\mathbf{M}_\phi$ represents the interaction between each phase [39,41], for instance, interfacial drag force, lift force, virtual mass, etc.: $\mathbf{M}_\phi = \alpha_\phi(\mathbf{F}_\mathrm{d} + \mathbf{F}_\mathrm{l} + \mathbf{F}_\mathrm{vm})/V$, in which, $\mathbf{F}_\mathrm{d}$, $\mathbf{F}_\mathrm{l}$, and $\mathbf{F}_\mathrm{vm}$ refer to the drag force, lift force, and virtual mass, respectively. The drag force is non-trivial if a relative velocity (slip) exists between phases,

$$\mathbf{F}_\mathrm{d} = \frac{1}{2}\rho_\phi A C_\mathrm{d} |\mathbf{U}_\mathrm{r}| \mathbf{U}_\mathrm{r} \tag{4}$$

where A represents the projected area, normal to the relative velocity, for spherical droplets/bubbles, and $\mathbf{U}_\mathrm{r}$ is the relative velocity. The drag coefficient, C_d, depends on the properties of droplets/bubbles and can be obtained empirically, such as in the Schiller and Numann drag model [57] that gives $C_\mathrm{d} = 24(1 + 0.15\mathrm{Re}^{0.687})/\mathrm{Re}$, where Re is the Reynolds number based on the particle diameter, d, and relative velocity.

Momentum equations in porous electrodes

Darcy's law is usually applied to describe the gas/liquid flows in porous electrodes. Therefore, a Darcy-modified Navier-Stokes equation is employed [58],

$$\begin{aligned}&\frac{\partial}{\partial t}\left(\rho_\phi \alpha_\phi \mathbf{U}_\phi\right) + \nabla \cdot (\alpha_\phi \rho_\phi \mathbf{U}_\phi \mathbf{U}_\phi) \\ &\quad = -\alpha_\phi \nabla p_\phi + \nabla \cdot (\alpha_\phi \mu_\phi \nabla \mathbf{U}_\phi) + \alpha_\phi \rho_\phi \mathbf{g} + \mathbf{M}_\phi + \alpha_\phi S_{\mathrm{Darcy},\phi}\end{aligned} \tag{5}$$

where $S_{\mathrm{Darcy},\phi}$ represents the porous resistance, which can be expressed as,

$$S_{Darcy,\phi} = -\frac{\alpha_\phi \mathbf{U}_\phi \mu_\phi}{K \cdot K_{\mathrm{r},\phi}} \tag{6}$$

where K is the absolute permeability, and $K_{\mathrm{r},\phi}$ is the relative permeability, which may be presumed to be of the form, $K_{\mathrm{r},\phi} = K_{\mathrm{r},\phi}\left(\alpha_\phi\right)$ [59], e.g. $K_{\mathrm{r},\phi} = \alpha_\phi^3$ [60].

It should be noted that the phase pressure p_ϕ appears in Eq. (5), rather than the shared pressure. Water movement in the porous GDL is driven by the capillary pressure [32], which results in the following relation,

$$p_l = \begin{cases} p_g & \text{in the channel} \\ p_g - p_c & \text{in the porous region} \end{cases} \tag{7}$$

where the subscripts l and g represent the liquid and gas phases, respectively. The capillary pressure is frequently computed using a Leverett J function [61],

$$p_c = \frac{\sigma \cos\theta}{\sqrt{K/\varepsilon}} J(\alpha_l) \tag{8}$$

where σ is the surface tension of the liquid-gas pair, θ is the contact angle of the porous material, ϵ is the porosity, and the formula suggested by Udell [62] and Pasaogullari and Wang [63] for the Leverett J function is,

$$J = \begin{cases} 1.417(1-\alpha_l) - 2.120(1-\alpha_l)^2 + 1.263(1-\alpha_l)^3 & \theta > 90\,\text{deg} \\ 1.417\alpha_l - 2.120\alpha_l^2 + 1.263\alpha_l^3 & \theta < 90\,\text{deg} \end{cases} \tag{9}$$

the formulations of J are not limited [64]. It should be noted that this formulation may not be able to quantitatively predict the behavior of liquid water in the porous material, while it can be used qualitatively [65].

Mass transfer

It is assumed that liquid water dissolves negligible amounts of the gases. Mass transfer here, refers to the diffusion of different components in the gas phase. The governing equation is as follows,

$$\frac{\partial}{\partial t}\left(\rho_\phi \alpha_\phi Y_\mathrm{i}\right) + \nabla \cdot (\alpha_\phi \rho_\phi \mathbf{U}_\phi Y_\mathrm{i}) = \nabla \cdot (\alpha_\phi \rho_\phi D_\mathrm{i}^{\mathrm{eff}} \nabla Y_\mathrm{i}) + \alpha_\phi R_\mathrm{i} \tag{10}$$

where the subscript i indicates the species component, Y represents the mass fraction, R is the species mass source/sink, and D^{eff} is the effective species diffusion coefficient. The model used to calculate the diffusion coefficients is nearly identical to previous work [15], to which the reader may refer for a detailed description.

Heat transfer

Temperature is a global parameter that is primarily addressed within the main mesh. However, when multi-phase flow is present, it becomes necessary to account for the temperature of each phase involved. In this scenario, the temperature of the continuous phase is solved implicitly in a global manner, whereas the temperature of the other phases are addressed within their respective local fluid regions. The equation for calculating the enthalpy of each segregated phase to be solved is as follows,

$$\frac{\partial}{\partial t}\left(\alpha_\phi \rho_\phi h_\phi\right) + \nabla \cdot (\alpha_\phi \rho_\phi \mathbf{U}_\phi h_\phi) = \nabla \cdot (\alpha_\phi \Gamma_\phi^{\mathrm{eff}} \nabla h_\phi) + \alpha_\phi Q_\phi \tag{11}$$

where h represents the enthalpy, Γ^{eff} is the effective thermal diffusivity, and Q is the thermal heat source/sink due to phase change, interface heat transfer, and/or electrochemical reactions. In the case of two-phase flows, however, the enthalpy of the dispersed phase is computed in the fluid sub-regions (branch meshes).

Condensation and evaporation

Depending on the operating temperatures, water may exist in both vapor and liquid states (phases). Condensation and evaporation between the two [66,67] is important. Fig. 2 illustrates schematically the physical processes at a virtual interface between phases 1 (gas) and 2 (liquid).

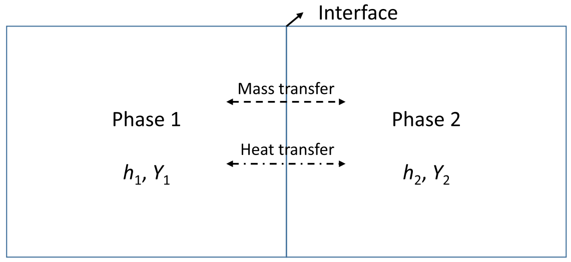


Fig. 2. 'Interface' between phases 1 and 2.

From a physical point-of-view, a heat balance and mass balance must simultaneously be satisfied due to conservation requirements. The equations of mass and heat transfer are as follows,

$$\dot{m}_{21}'' = -(\rho \mathbf{U} Y - \Gamma \nabla Y)_1 = (\rho \mathbf{U} Y - \Gamma \nabla Y)_2 \tag{12}$$

$$\dot{m}_{21}'' h_{21} = -(\rho \mathbf{U} h - \lambda \nabla h)_1 = (\rho \mathbf{U} h - \lambda \nabla h)_2 \tag{13}$$

where Eq. (12) represents the mass balance between phases 1 and 2, and Eq. (13) the heat balance. (NB: $\rho \mathbf{U} = \dot{m}''$)

In order to obtain the rates of heat and mass transfer through the interface, either the interfacial mass fraction or the interfacial temperature must be prescribed. Assuming the latter, if T_{int} is the prescribed value, then the interfacial mass fraction of the species i can be obtained as,

$$Y_{\mathrm{int,\,i}} = \frac{M_\mathrm{i}}{M_{\mathrm{mix}}} \frac{p_{\mathrm{sat}}(T_{\mathrm{int}})}{p_\mathrm{i}} \tag{14}$$

where $Y_{\mathrm{int,\,i}}$ indicates the interface mass fraction of the transferred species i, M_{mix} represents the mixture molar mass, and p_{sat} is the saturation pressure of the liquid in the gas that is a function of temperature, and p is the gas partial pressure.

The rate of mass transfer can be expressed as,

$$\dot{m}_{21}'' = \dot{m}_{21}'' Y_{\mathrm{int}} + \Gamma/\delta (Y_{\mathrm{int}} - Y_1) \tag{15}$$

$$\dot{m}_{21}'' = \frac{\Gamma}{\delta} \frac{Y_1 - Y_{\mathrm{int}}}{Y_{\mathrm{int}} - 1} \tag{16}$$

where δ is the distance (between the bulk and interface), and a central-difference averaging scheme is employed to weigh the influences of convection and diffusion. From the heat balance,

$$(\rho \mathbf{U} h - \lambda \nabla h)_2 + (\rho \mathbf{U} h - \lambda \nabla h)_1 = 0 \tag{17}$$

a new expression can be obtained,

$$(\lambda \nabla h)_2 + (\lambda \nabla h)_1 = \dot{m}_{21}'' h_{21} \tag{18}$$

Then,

$$k_2 \frac{T_{\mathrm{int}} - T_2}{\delta_2} + k_1 \frac{T_{\mathrm{int}} - T_1}{\delta_1} = \dot{m}_{21}'' h_{21} \tag{19}$$

where k represents the phase thermal conductivity.

In Eq. (19), all of the values, except for the prescribed value of T_{int}, are known. Therefore, T_{int} can be solved based on a root-finding algorithm (Newton's method in this work).

2.1.3. Solid region

Within the current framework, no governing equations are explicitly solved within the solid regions. From a physical standpoint, the solid regions dissipate heat between different components in the electrochemical device assembly and provide necessary mechanical support. Heat transfer is numerically considered in the global domain, see Section 2.2. However, mechanical stresses are not considered in the current work but may be considered in the future.

2.1.4. Electric region

In a closed system/circuit, ions/protons are transferred via the electrolyte/membrane, whereas electrons are conducted through the electrodes, interconnects, and the external load. Both may be considered to follow a Poisson (Ohm's law) type relationship,

$$\nabla \cdot (\sigma_{\mathrm{E}} \nabla \varphi_{\mathrm{E}}) = J_{\mathrm{E}} \tag{20}$$

$$\nabla \cdot (\sigma_{\mathrm{I}} \nabla \varphi_{\mathrm{I}}) = J_{\mathrm{I}} \tag{21}$$

where φ represents the potential, the subscripts E and I indicate electronic and ionic values, respectively, σ is the electric conductivity, and J refers to the source/sink of charge, due to the electrochemical reaction, e.g., at the air electrode:

$$\begin{cases} J_{\mathrm{E}} = -j_{\mathrm{ae}} \\ J_{\mathrm{I}} = j_{\mathrm{ae}} \end{cases} \tag{22}$$

and, at the fuel electrode:

$$\begin{cases} J_{\mathrm{E}} = j_{\mathrm{fe}} \\ J_{\mathrm{I}} = -j_{\mathrm{fe}} \end{cases} \tag{23}$$

The calculation of j can be found in Section 2.3. The materials employed in typical fuel cells/electrolyzers exhibit much higher electronic than ionic conductivity and therefore the former can often be neglected [15]. The ionic conductivity in the membrane/electrolyte is usually a function of temperature and water content in a low-temperature PEMFC, or of phosphoric acid doping level in a high-temperature PEMFC [68], etc. As an example, the expression for a Nafion-based membrane is [69],

$$\sigma_{\mathrm{I}} = \begin{cases} (0.514\lambda - 0.326)\exp\left(1268(\frac{1}{303} - \frac{1}{T})\right) & \lambda \geq 1 \\ 0.1879\lambda\exp\left(1268(\frac{1}{303} - \frac{1}{T})\right) & \lambda < 1 \end{cases} \tag{24}$$

Water transport through the membrane

The membrane is assumed to be impermeable to reactants/products. However, it selectively permits the cross-over of water. The mechanisms of water transfer include diffusion, electroosmotic drag (EOD), and hydraulic permeation. Diffusion and EOD are usually considered dominant in PEMFCs, and the hydraulic permeability coefficient is typically negligibly small. A non-equilibrium transport equation for water transfer in the membrane can be written as follows,

$$\frac{\rho_{\mathrm{m}}}{\mathrm{EW}} \frac{\partial \lambda}{\partial t} + \nabla \cdot \left(\frac{\mathbf{i}_{\mathrm{I}}}{F} n_d \right) = \frac{\rho_{\mathrm{m}}}{\mathrm{EW}} \nabla \cdot (D_{\mathrm{m}}^{\mathrm{eff}} \nabla \lambda) + R_{\lambda} \tag{25}$$

where ρ_{m} is the (dry) membrane density, EW is the equivalent weight of the membrane, λ represents the water content in molar concentration, $\mathbf{i}_{\mathrm{I}}$ is the ionic current density, n_d is the EOD coefficient, $D_{\mathrm{m}}^{\mathrm{eff}}$ represents the effective water diffusion coefficient in the membrane, and R_{λ} is the source/sink term of water desorption/adsorption.

2.2. Global domain

In the absence of effects due to pressure work and body forces, concentration gradients (Dufour), and viscous dissipation, the energy equation in the global region (main mesh) may be written as,

$$\frac{\partial}{\partial t}\left(\rho C_p T\right) + \rho C_p \mathbf{U} \cdot \nabla T = \nabla \cdot (k^{\mathrm{eff}} \nabla T) + Q \tag{26}$$

where C_p is the specific heat capacity, k^{eff} is the effective thermal conductivity, and Q denotes the heat sources/sinks mapped from all sub-regions (Note: $\mathbf{U}$ is identically zero in all solid regions). The effective thermal conductivity in porous zones is calculated as,

$$k^{\mathrm{eff}} = \varepsilon \alpha_{\mathrm{l}} k_{\mathrm{l}} + \varepsilon \alpha_{\mathrm{g}} k_{\mathrm{g}} + (1 - \varepsilon) k_{\mathrm{s}} \tag{27}$$

2.3. Electrochemistry

The membrane electrode assembly (MEA) or positive electrolyte negative (PEN) electrode assembly is electrochemically active and consists of an ion/proton conducting electrolyte/membrane and two adjacent active layers. The oxidation and reduction reactions take place at the air and fuel electrodes, respectively. The relation between the electronic and ionic/protonic potentials can be described by suitable electrochemical kinetic expressions, such as Butler-Volmer and Tafel equations; for example, for the air electrode, (with similar expressions possible at the fuel electrode),

$$\begin{aligned} j_{\mathrm{ae}} = i_{\mathrm{ae,\,0}}(1 - \alpha_l)^{\gamma} \prod_i \left(\frac{C_{\mathrm{i}}}{C_{\mathrm{ref}}} \right)^{\xi} \\ \left[\exp\left(\frac{-\alpha_{\mathrm{ae}} z F \eta_{\mathrm{ae}}}{R_g T} \right) - \exp\left(\frac{(1 - \alpha_{\mathrm{ae}}) z F \eta_{\mathrm{ae}}}{R_g T} \right) \right] \end{aligned} \tag{28}$$

or,

$$\begin{aligned} j_{\mathrm{ae}} = i_{\mathrm{ae,\,0}}(1 - \alpha_l)^{\gamma} \\ \left[\prod_{\mathrm{reac,i}} \left(\frac{C_{\mathrm{i}}}{C_{\mathrm{ref}}} \right)^{\xi} \exp\left(\frac{-\alpha_{\mathrm{ae}} z F \eta_{\mathrm{ae}}}{R_g T} \right) \right. \\ \left. - \prod_{\mathrm{prod,i}} \left(\frac{C_{\mathrm{i}}}{C_{\mathrm{ref}}} \right)^{\xi} \exp\left(\frac{(1 - \alpha_{\mathrm{ae}}) z F \eta_{\mathrm{ae}}}{R_g T} \right) \right] \end{aligned} \tag{29}$$

where i_0 denotes the exchange current density, z the number of electrons transferred, F Faraday's constant, C_{i} the reactant concentration, C_{ref} the corresponding reference concentration (inlet species concentration), α_{ae} the transfer coefficient, β the power exponent, γ the reaction order, and η_{ae} the activation overpotential,

$$\eta_{ae} = \varphi_{\mathrm{E}} - \varphi_{\mathrm{I}} - E_{\mathrm{Nernst,\,ae}} \tag{30}$$

where $E_{\mathrm{Nernst,\,ae}}$ denotes the Nernst potential which is calculated as:

$$E_{\mathrm{Nernst,\,ae}} = \sum_{\mathrm{i}} \frac{-(H_{\mathrm{i}} - T\, S_{\mathrm{i}})\omega - R_g T \omega \ln p_{\mathrm{i}}}{zF} \tag{31}$$

where G is Gibbs free energy, p_{i} the partial pressure of species i, H the enthalpy, S the entropy, and ω a reaction number that is negative for consumption and positive for production, corresponding to the transfer of z electrons.

3. Software design and code structure

The open-source library `OpenFOAM` serves as the development platform for the multi-region, multi-physics, and multi-scale calculations for electrochemical devices. The majority of the present library, `openFuelCell2`, is written in the object-oriented programming language, C++. Users/developers are provided with access to the entire code. It is relatively easy for users/developers to modify and extend parts of the code for their own usage. It is also not difficult to locate and apply features, such as mesh manipulation, field creation, numerical discretization, parallel computing, etc., based on built-in `OpenFOAM` functionality.

Fig. 3 illustrates the file structure of the source code repository. This is similar to the first version of the `openFuelCell` code, which dates back to the open-source model of SOFCs, developed by Beale et al. [15]. The applications of the original solver were limited to high-temperature fuel cells [70], e.g. SOFCs, high-temperature PEMFCs [23,24], and short stacks [22]. To broaden the scope of possible applications, e.g., low-temperature fuel cells and electrolyzers, for which a substantial market exists, and where two-phase (liquid/gas) flow is important, major modifications were implemented so as to enable flexible combinations of various types of regions and physical processes to be accounted for. Among the most important issues related to the design is the coupling and decoupling of multi-physics in multiple regions.

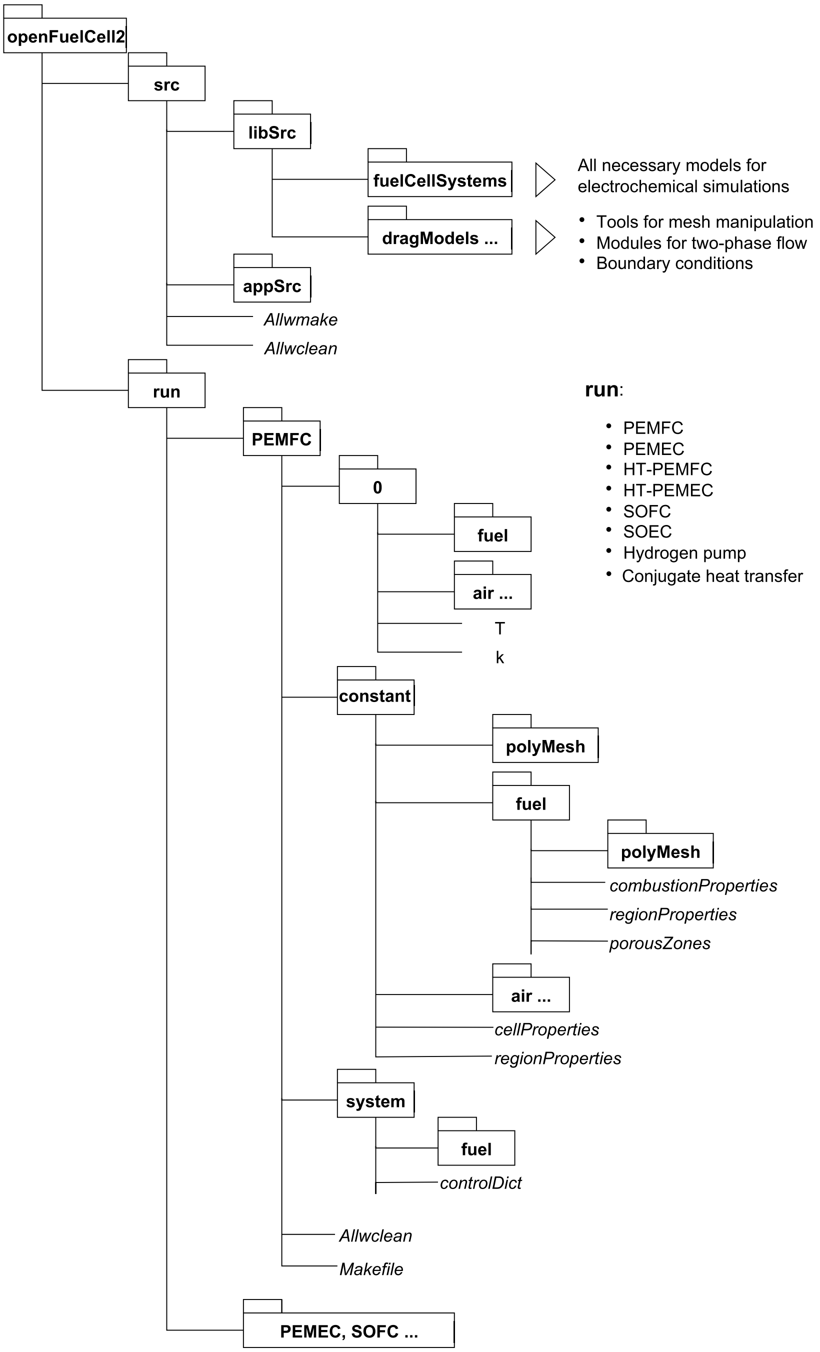


Fig. 3. Dictionary and file structure of the repository.

3.1. Domain decomposition

In the FVM, the governing partial differential equations are integrated over finite volumes, discretized e.g., with second-order accuracy, and solved for over the computational domain (for example using polyhedral meshes). In `OpenFOAM`, the meshes consist of a set of consecutively numbered cells or volumes that cover a region of space and are bounded externally by a set of surfaces or patches. In the present design, the global region of space is further decomposed into a number of local regions, known as cell zones, according to (but not limited to) different material properties and physical processes. OpenFOAM introduces two distinct types of boundary patches, namely, `regular` and `coupled`. The former refers to the more usual boundary condition types, such as fixed value (Dirichlet), fixed gradient (Neumann), combined/mixed (Robin), symmetry, and wedge. The latter provides the potential of implicitly coupling with other computational meshes and consists of several types of boundary conditions, e.g. periodic (cyclic/cyclicAMI), jump, and conjugate-coupled.

Three mechanisms can be employed to handle the complex physical processes in different electrochemical devices, for example, [15]:

- *Monolithic approach*: All the different component zones are tessellated with a single computational mesh, and the full set of finite-volume equations are solved throughout, with terms being disabled or parameters being artificially manipulated as necessary within appropriate sub-zones.
- *Partitioned approach*: Each part is individually tessellated with a computational mesh, and the corresponding set of finite-volume

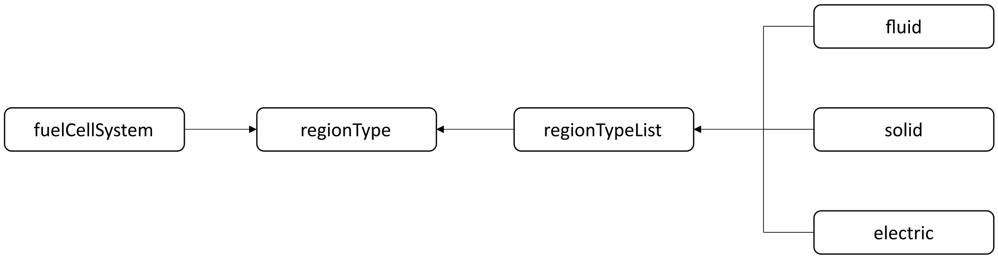


Fig. 4. The top-level design of the class `fuelCellSystem`.

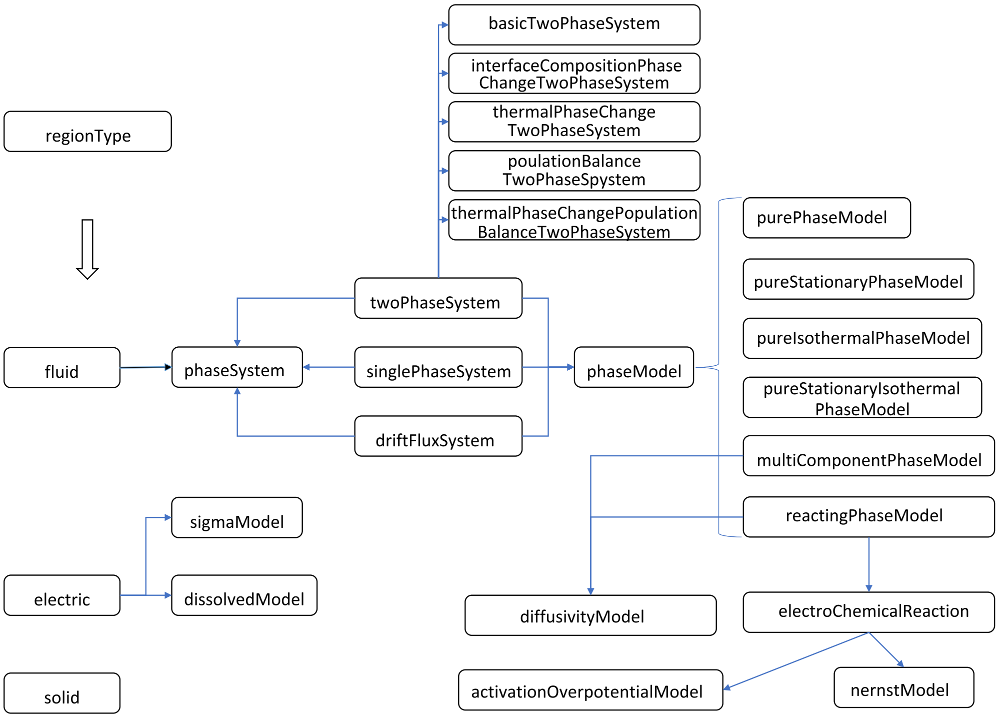


Fig. 5. The design of class `regionType`.

the different meshes are then coupled, as required.

equations are discretized and solved, locally. The patches between the different meshes are then coupled, as required.

- *Integrated cell concept*: A single global mesh is created and then decomposed into a number of individual component meshes. Global transport phenomena, e.g., heat transfer, can be calculated on the global mesh, whereas the individual computational meshes address the specific physical processes unique to the component parts, accordingly.

The pros and cons of these mechanisms were discussed by Beale et al. [15]. Therefore, the top-level design of the present solver employs the same integrated cell concept for the multiple region cases, which is identical to the previous work. This does not necessarily mean the integrated cell approach performs best under all circumstances as the assembled parts may contain adjacent and/or overlapping spatial regions, which makes mesh decomposition complicated. The partitioned approach may offer advantages under certain circumstances. Until comprehensive comparisons of the three methods are conducted, however, the authors follow the selection of the domain decomposition rationally selected in `openFuelCell` [15], on the basis of the rationale “If it works, don’t fix it”.

4. Code design and class structure

In order to achieve a robust solution framework, the `runTimeSelection` mechanism, as is widely used in the `OpenFOAM` libraries, is employed in the present code. This normally refers to a templated implementation via a ‘virtual constructor’ in C++, most of which can only be seen from macros. In object-oriented programming languages, an object usually interacts with a generic base class to manipulate derived classes. This greatly simplifies the top-level code design. A group of objects performing the same, or similar functions under a common interface can easily be created. This type of design makes the choice of different models simple, and avoids the need to utilize `if-then-else` blocks during programming. Examples can easily be found in many code extractions of `OpenFOAM`, such as boundary conditions, temporal/gradient/laplacian discretization schemes, linear equation solvers, turbulence models, property libraries, and so on. Many selections and setups of physical models and corresponding parameters are decided at run time. Regarding specific electrochemical devices, the present code takes care of selections with different types of regions/solvers, as well as necessary physical models to be included. For instance, the types of regions can be seen in Fig. 1.

4.1. Individual software components

Starting with the presentation of the class structure, Figs. 4 and 5 represent the simplified outline of the overall software design used in this procedure. They both illustrate the individual libraries (classes), partially shown by created typedefs (similar to an abbreviation), as well as the built-in modules in `OpenFOAM`. As was previously mentioned, the design widely employs inheritance and polymorphism mechanisms,

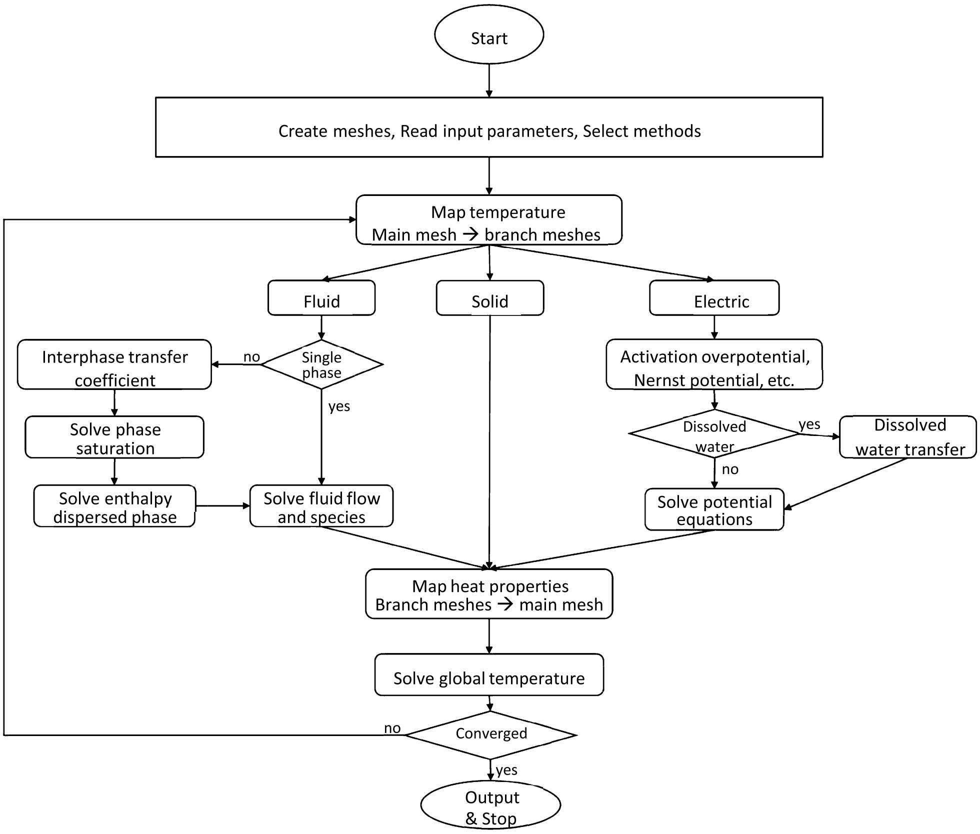

Start
Create meshes, Read input parameters, Select methods
Map temperature
Main mesh → branch meshes
Fluid
Solid
Electric
Interphase transfer coefficient
no
Single phase
yes
Solve phase saturation
Solve enthalpy dispersed phase
Solve fluid flow and species
Activation overpotential, Nernst potential, etc.
Dissolved water
yes
Dissolved water transfer
no
Solve potential equations
Map heat properties
Branch meshes → main mesh
Solve global temperature
no
Converged
yes
Output & Stop

Fig. 6. Solution sequence of the solver.

templates, and especially the runTimeSelection feature to ensure the minimization of code duplication and to make the solver applicable to a variety of cases. The top-level design fuelCellSystem, shown in Fig. 4, contains the global mesh and thermal properties, a list of sub-meshes, and necessary interfaces, like solve, correct, and map functions.

Inheritance is indicated in Fig. 5; for example, the fluid class, inherits the attributes of the regionType class, which is a fvMesh, itself. The regionType class provides, among other things, the mapping functionalities and characteristics of a mesh, and, in addition, the fluid defines its individual attributes and functions. Adhering to such a framework makes it straightforward to add new derived classes to extend and enhance the existing code.

The fuelCellSystem comprises the different regionTypes and their coupling strategies, where the different physical systems are built up and calculated. These regions on their own do not necessarily have to know how the overall system is created. The regionTypes, fluid, electric and solid create the main framework for the simulation of the multiphysical system, as shown in Fig. 4.

Accordingly, each region accounts for one or more physical-electrochemical aspects. The fluid class, for instance, holds the phaseSystem, which consists of the singlePhaseSystem, twoPhaseSystem, and driftFluxSystem. The selected phaseSystem characterizes the interactions, such as the exchange of momentum and enthalpy between the different phases, and collects the contributions of the individual phases to construct the governing equations for the system of phases. A twoPhaseSystem object, for instance, holds two different phaseModels and captures their phase fractions by solving an additional phase fraction equation. With the aid of a run-time-selected phaseModel, it is possible to specify in which way the governing equations themselves are solved and coupled with each other. Here again, it must be mentioned that the phaseModels are actually type definitions of large multi-layered templated classes, which themselves contain a number of templated classes, to be able to describe all aspects of the multiphysics that take place in the fluid regions. That is, a suitable phaseModel could specify the fluid motion, species transport, and energy/enthalpy conservation within the fluid, and electrochemistry. The treatment of the electrochemistry, which is only integrated into the reactingPhaseModel, is highlighted here by the addition of the electroChemicalReaction, activationOverpotentialModel and nernstModel classes in Fig. 5. These templated classes provide the necessary framework to calculate many aspects of electrochemistry, such as the overpotentials, sinks/sources of the reacting species, and the Nernst potentials, for instance. Besides the fluid class, the equations for the electrochemical potentials, both for ions and electrons, are solved within the electric class. The required electronic and ionic conductivities are calculated by the sigmaModel. In order to connect the electrical potentials and the electrochemical reactions within the fluid region(s), mutual data access is required. This is generally realized via access of the data over the object registry or mesh-to-mesh mapping algorithms, as it is also performed in relation to communicating with the global mesh. The solid class, for now, does not have any functionalities apart from the transfer of thermal properties to the global region (main mesh). In order to get a more detailed overview of this multilayered code, the use of visualization tools such as sourcetrail [71] is recommended.

4.2. Solution process

The solution processes might differ for different applications, providing varying combinations of regions and physics, for example, a conjugate heat transfer problem contains fluid and solid regions, without electric ones. The principal concepts are similar to the previous solution algorithm [15]. The global region (main mesh) accounts for the overall conservation of energy; other physical processes are taken into consideration in the sub-meshes (branch meshes); communications between the main and branch meshes include passing values of temperature and thermal properties, thermal conductivity, and specific heat, etc. Fig. 6 exhibits a typical simulation sequence for fuel cells/electrolyzers,

5. User instructions

The current section provides additional information about:

- How to obtain the code
- How to compile the code
- How to use the code

In particular, it should be noted that it is not possible to use the code with all OpenFOAM versions, as there are differences in the codes, especially between the different forks [72,73], as well as in new releases of the similar fork series.

5.1. Prerequisites and how to obtain the code

In the present development, the code openFuelCell2 is compatible with several versions of OpenFOAM, namely ORG (V10, V8) and COM (2306, 2212...). The code is maintained by the present authors. Updates are available on GitHub with the link: https://github.com/openFuelCell2/openFuelCell2. Users are able to obtain the source code either via an archived file or git. It is recommended to download the code in the latter way, as git provides a control system for distributed revisions.

```
## <branch> represents the specific branch name
## including of8, of10, of2212...
## <remote_repo> the remote repository
## https://github.com/openFuelCell2/
openFuelCell2.git
## git clone -b <branch> <remote_repo>
git clone https://github.com/openFuelCell2/
openFuelCell2.git
```

5.2. How to use openFuelCell2

The compilation of the source code requires some fundamental knowledge of OpenFOAM. The compilation process is similar to the built-in modules in standard OpenFOAM repositories. A particular version of OpenFOAM should be installed/complied with at the first step. Suitable versions include foundation versions 9 and others, and also the versions developed by OpenCFD/ESI, versions 2306 and others. The auto-compiling (executable) script Allwmake can be found in openFuelCell/src. With ./Allwmake or ./Allwmake -j n, where *n* describes the number of cores used, the source code can be compiled in serial or parallel (n is the number of cores), respectively. Once the compilation is finished, the executable application openFuelCell2 becomes available. It should be noted that there may exist some warnings during compiling, however, they should have no effects on the execution. With ./Allclean, the libraries and executable files can be cleaned up.

Test cases can be found in openFuelCell/run. The applicable cases are not limited to these alone, however. Inside each test case, the Makefile contains useful commands to perform the simulation. The user needs to type make mesh and make srun in the terminal to create the computational meshes and conduct the numerical simulation in serial, respectively. When the calculation comes to an end, the scripts make view and make viewAll can be executed to convert the latest time results and full-time results into the Visualization Toolkit (VTK) format which is a supported file format by paraView [74]. A parallel simulation on the other side requires make mesh, make decompose, make parallel, and make run. The situation is more complicated for the parallel simulations due to multi-region coupling. It is required that the coupled regions/patches are located on the same cores/processors, at the time of execution. Existing standard OpenFOAM decomposition methods do not meet the requirements, at least for complex geometries. Therefore, the meshes are decomposed with the manual option. In order to achieve this, the values of nx and ny have to be prescribed in constant/cellProperties, which represents the number of divisions in the x and y directions, respectively. The number of decompositions in system/decomposeParDict and the number of processors in parallel running must be identical to the product of nx × ny.

5.3. Tips and tricks

Typical electrochemical devices include multi-region and multi-physics coupling. Obtaining stable/converged numerical results requires effort. It is not always easy to achieve a stable and converged solution in numerical simulations, especially when it comes to two-phase systems. Here are some suggestions to make the solution process more stable and converge faster:

- Improve the mesh quality.
- Decrease the Courant number in transient and local-time-step (LTS) simulations.
- Adjust relaxation factors for equations and fields.
- Adjust the correction coefficients, for instance, to the electric potential.
- Apply the faceMomentum method in two-phase flow cases. This formulation provides C-grid-like pressure-flux staggering on an unstructured mesh which is hugely beneficial for Euler-Euler multiphase equations as it allows for all forces to be treated in a consistent manner on the cell-faces which provides better balance, stability, and accuracy [75].
- Turn on consistent in PIMPLE dictionary for single-phase solutions. This activates the option of the SIMPLEC algorithm, which makes the solution more stable [76].

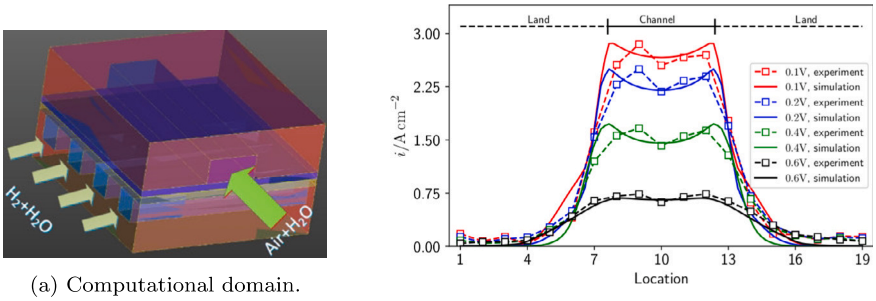
6. Validation and results

The following section investigates various aspects of multiple applications to showcase the capabilities and possibilities of the openFuelCell2 framework. The results obtained from the simulations are validated against experimental data, including data generated from in-house measurements. Furthermore, a brief comparison is made between the openFuelCell2 and the previous openFuelCell code [15] to demonstrate the improvements and extensions made in the latest version.

6.1. Low temperature PEMFC

In this section, a PEMFC prototype is selected from the work of Schneider et al. [77]. The dimension of the prototype was 0.7 × 0.8 cm^2. Detailed geometrical information can be found in the original paper. The fuel cell was operated with fully humidified hydrogen and air at a temperature of 70 °C under ambient pressure. All operations were conducted in potentiostatic mode and the inlet fluxes were fixed. The stoichiometric factors, λ, (the ratio of supply to consumed reactant), were 20 and 24 for the cathode and anode sides, respectively, for a mean reference current density of 1.0 A cm^{-2}.

The current density variations in the air flow direction remain almost constant provided that the stoichiometric factors are large at both sides. The local current density variations in the center of the membrane were considered by choosing a straight line normal to the airflow direction. The computational domain and flow configurations are shown in Fig. 7a. Fig. 7b which display a comparison of local current densities at various cell voltages. It can be seen that the local current densities are predicted by the present code in comparison to the experimentally measured data for different cell voltages, with minor deviations. The current



(a) Computational domain.

(b) Local current density variations.

Fig. 7. Comparison of local current density variation under different cell voltages [77]. Permission obtained.

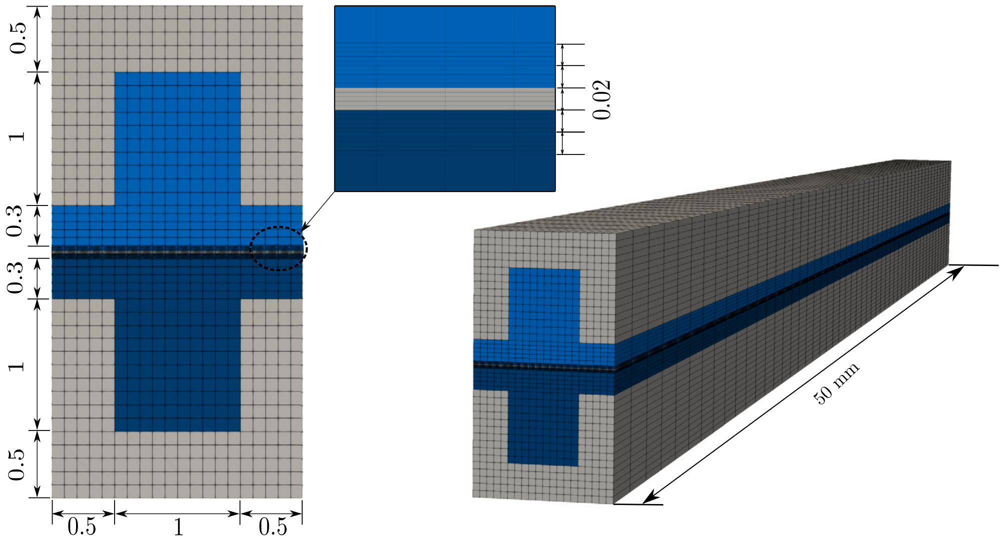


Fig. 8. Geometry and mesh of a case for a single straight channel. All dimensions are in millimeters.

density in the land region is much lower compared to the channel region due to the issue of oxygen diffusion in the air in the GDLs. The slightly decreasing values in the channel center possibly result from lower water content in the membrane, as less liquid water accumulates in this zone. The results of the numerical predictions are quite promising. There exist slight deviations between the experimental and numerical results, e.g., the values and locations of extrema, as the compression of the GDL and contact resistance (both thermal and electric) are not taken into account by the present simulations.

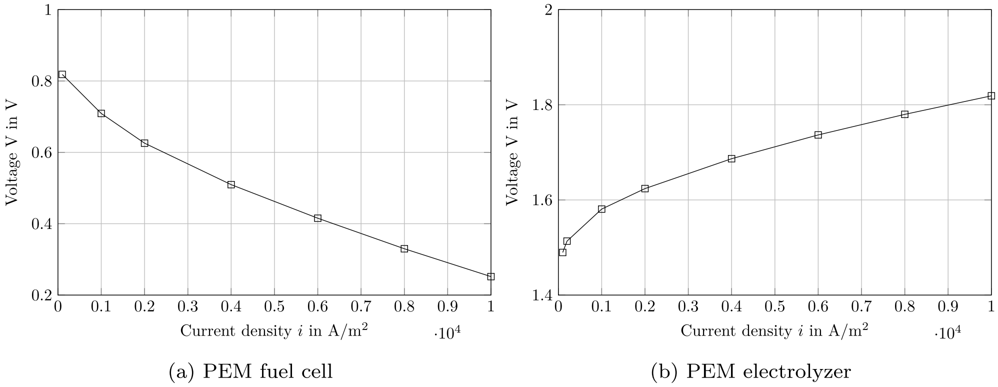
6.2. Low-temperature PEMFC/PEMEC: single channel

In this subsection, numerical simulations of a low-temperature PEMFC and PEMEL are presented. Fig. 8 exhibits the computational domain and mesh for both applications. These represent a simple example of one straight channel for both the anode and cathode sides in a co-flow configuration. The corresponding tutorial cases provided within the `openFuelCell2` framework consist of a similar structure with different case-dependent settings. As it is also carried out in the tutorial cases, a structured hexahedral mesh is created using an `OpenFOAM` utility, `blockMesh`. General instructions to establish similar cases together with a collection of all the required data are provided in Appendix B. As both the fuel cell and electrolyzer operate under atmospheric pressure and temperatures of around 353.15 K, the two-phase flow regime consists of gaseous and liquid phases. Humidified air and hydrogen (RH $= 0.5$) are supplied in the fuel cell application, while pure liquid water is injected on both sides for the electrolyzer case. The fuel cell operates under constant stoichiometric factors ($\lambda = 5/5$) of air and hydrogen, whereas for the electrolyzer case, the stoichiometric factor of water is significantly higher. Fig. 9 shows the polarization curves of the PEMFC and PEMEC. It can be seen that the current density decreases/increases due to the activation and ohmic overpotentials, for the fuel cell/electrolyzer, respectively.

It is usually important to have sufficient humidification on both sides to ensure high ionic conductivity in the membrane, as it is a strong function of the water content. The water content in the membrane is constrained by the water concentration differences of both sides and the electroosmotic drag due to the proton transfers as Eq. (25) reveals. The water content dissolved in the membrane is represented in Fig. 10 for different current loads, $i = 0.4\,\mathrm{A\,cm^{-2}}$, $i = 0.6\,\mathrm{A\,cm^{-2}}$, and $i = 1.0\,\mathrm{A\,cm^{-2}}$, in fuel cell operations. It can be observed that the water content increases between the inlet (left) and outlet (right). This is because the water is produced in the liquid phase and accumulates in the down flow. The overall water content also increases with higher current densities as more water is generated due to the electrochemical reaction.

The gaseous species distributions, oxygen, and hydrogen, can be seen in Fig. 11 for various cross-section positions from inlet to outlet. Both oxygen and hydrogen mass fractions decrease from inlet to outlet due to the consumption of reactants in the electrochemical reactions. It can also be seen that the mass fractions are lower in the GDLs than in the channels, due to the electrochemical reactions in the adjacent catalyst layer and diffusion of the reactants through the different porous layers.

Another important requirement that has to be fulfilled by the applied model is mass conservation of reactants within the cell. Table 1 shows the values of the total mass flow rate entering the cathode in-



(a) PEM fuel cell

(b) PEM electrolyzer

Fig. 9. Polarization curves for a low-temperature PEM fuel cell and electrolyzer, respectively.

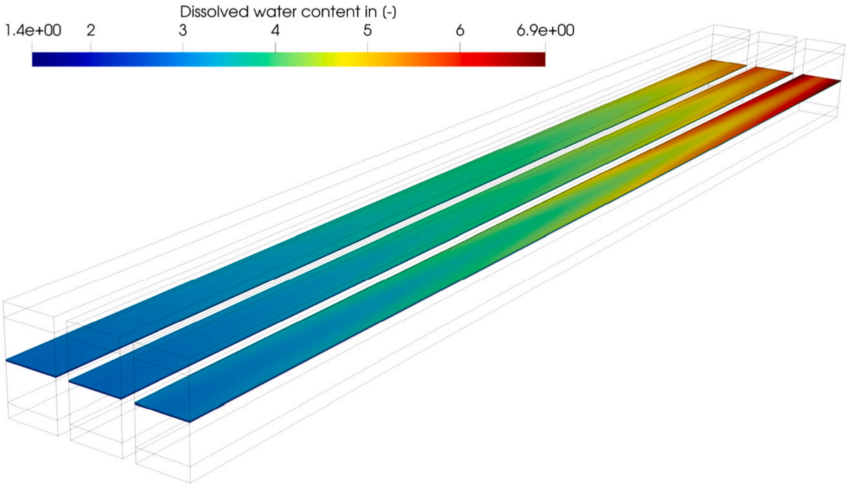


Fig. 10. Dissolved water content in the membrane for different current loads, $i = 0.4\mathrm{A\,cm^{-2}}$, $i = 0.6\mathrm{A\,cm^{-2}}$, and $i = 1\mathrm{A\,cm^{-2}}$ (from left to right).

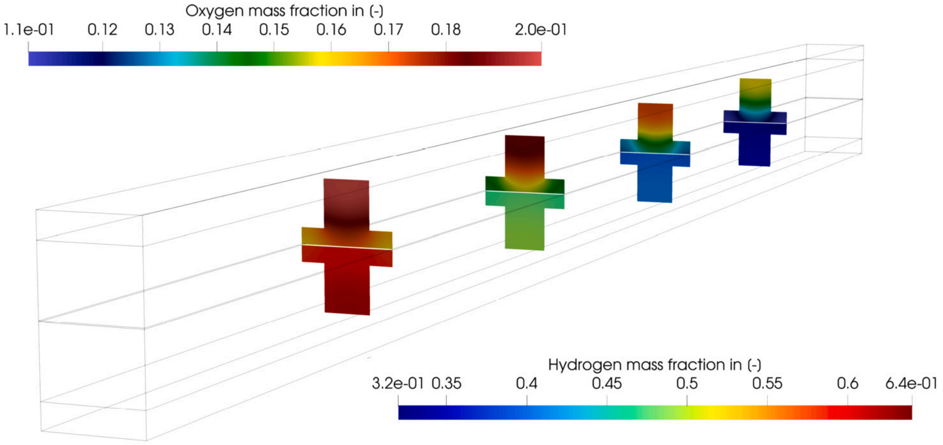


Fig. 11. Mass fraction distributions of oxygen and hydrogen on the cross-sections at different positions, $i = 0.8\mathrm{A\,cm^{-2}}$. Flow direction, left to right.

let, consumption due to the reaction, water transport, and the outlet flux, which is the sum of the flux of the gas mixture and the liquid water. These values are compared to those predicted by Faraday's law for the reaction, for example. It can be readily observed that the consumption of reactants and dissolved water shows good agreement between the numerical simulation and theoretical calculations. However, a small difference of around 1.9% arises for the outgoing mass flow rate. In order to obtain a more detailed overview of the different contributions of the ratio of species to total mass flux, readers can refer to Table A.3 for each species.

In the electrolyzer operations, liquid water is fed into both channels. In such a case, the membrane could be readily humidified to provide high electric conductivity. The liquid water saturation inside the cell impacts directly on the exchange current density and the diffusion coefficients of each species in the gas phase; therefore, it influences the overall performance of the electrolyzer. Fig. 12 displays the liquid water saturation in the fluid regions on both sides. Since oxygen and hydrogen are produced at the anode/upper and cathode/lower catalyst electrodes, respectively, lower liquid water saturation can be seen in the porous zones than in the channel zones. It could also be found

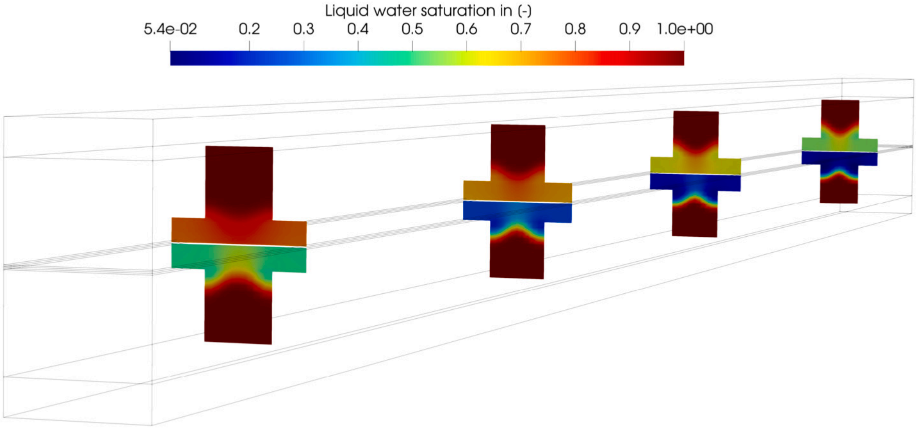


Fig. 12. Liquid water saturation distributions on cross-sections at different positions, for both anode and cathode sides, $i = -0.8\mathrm{A\,cm}^{-2}$.

Table 1
Mass conservation on the cathode side. In the upper part of the table, the mass flow rates $\dot{m}_i$ in kg/s expected from theory and gained from the simulation are listed.

	Inlet	Reactions	Drag water	Outlet
Theory	1.644e-06	8.357e-09	-3.580e-08	1.653e-06
`openFuelCell2`	1.643e-06	8.357e-09	-3.580e-08	1.587e-06
Rel. error in %	5.576e-02	1.051e-03	0	1.86

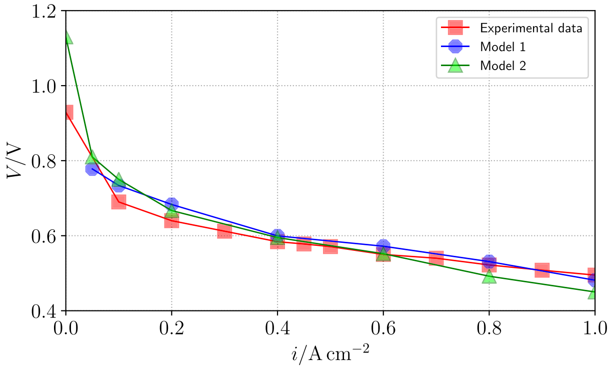


Fig. 13. Comparison of polarization curves. Model 1: `openFuelCell` [15], Model 2: `openFuelCell2`. Permission obtained from [79].

that more gases accumulate in the porous zones near the flow outlets and the amount of hydrogen gas exceeds the amount of oxygen on the anode side because the stoichiometric factor is doubled.

6.3. High-temperature PEMFC

In this section, numerical simulations were conducted for a single repeating unit of a short in-house designed high-temperature PEMFC stack [78,23]. The simulation results were then compared with the experimental measurements and numerical results [23] predicted by the `openFuelCell` model [15]. Additional comparisons were presented in a previously published paper [79]. The comparisons of polarization curves and local current density distributions are shown below. Further details regarding the operation of stack/repeating unit can be found in the work of Kvesić et al. [78]. Readers can refer to the work of Zhang et al. [23] for more detailed information.

Fig. 13 shows a comparison of polarization curves based on different methods. It can be seen that the voltages predicted by the present model, `openFuelCell2`, exhibit good agreement with the experimental data and the predictions of the previous model, `openFuelCell` [15]. The deviation only becomes larger near the open circuit voltage. As has been discussed in the work of Reimer et al. [80], a range of values of transfer coefficient, ohmic resistance, and exchange current density can provide essentially similar polarization curves. A comparison of local current density distributions is more illuminating. This is displayed in Fig. 14. It is clear that the results from `openFuelCell2` and `openFuelCell` fit quite well with the experimental measurement.

6.4. Scaling study

The parallel efficiency of the `openFuelCell2` code was demonstrated by means of a refined and elongated single channel for the low-temperature PEMFC case (compare Fig. 8) from the previous section 6.1. Therefore, it considers the two-phase flow together with the inter-phase heat and mass transfer using the `interfaceCompositionPhaseChangeTwoPhaseSystem` phase system. To illustrate the scalability, a `strong` scaling study was performed, that is, the number of computational finite volume cells was fixed, and the number of CPU cores employed was changed, in contrast to a `weak` scaling study, in which the load for each core was kept constant, the load for one CPU core changed in the strong scaling study.

For an ideal scaling algorithm, the computing time should scale linearly in relation to the number of CPU cores employed. This means that if the core count is doubled, the execution time should only be half. The computational domain consisting of 20,889,600 cells in total and 6,144,000 cells for each fluid region was decomposed with the `manual` scheme along the channel length to ensure that each core held at least one cell of each region. This is, for now, one of the requirements of the scheme. The lowest number of cores used was 48. As the simulation of the given problem with one computational core would have taken disproportionately long, a perfect scaling up to 48 cores was assumed at this point, i.e. $t_{1C} \approx 48 \times t_{48C}$. Here t_{1C} and t_{48C} describe the time needed for the simulation using one core or 48 cores, respectively. The core count is increased up to 960 cores where its number is always doubled from simulation to simulation.

The simulations were carried out on the CLAIX supercomputer from the RWTH University Aachen. Each compute node consists of two Intel Xeon Platinum 8160 Processors “SkyLake” (2.1 GHz, 24 cores each) and 192 GB main memory per node. The computational resources were provided by the Jülich Aachen Research Alliance (JARA-HPC).

Figs. 15a and 15b show computing time and speed-up in relation to one CPU core. In addition to this, Table 2 offers detailed data on the cells per CPU core. As a rule-of-thumb for finite volume methods using standard iterative solution methodologies, achieving good parallel efficiency requires approximately 20×10^3 to 50×10^3 cells per CPU core [81].

In this particular case, linear scaling is reached up to a core count of about 192, which corresponds to an overall number of cells of 108,800 per CPU core, as can be seen in Table 2. This value is obviously much

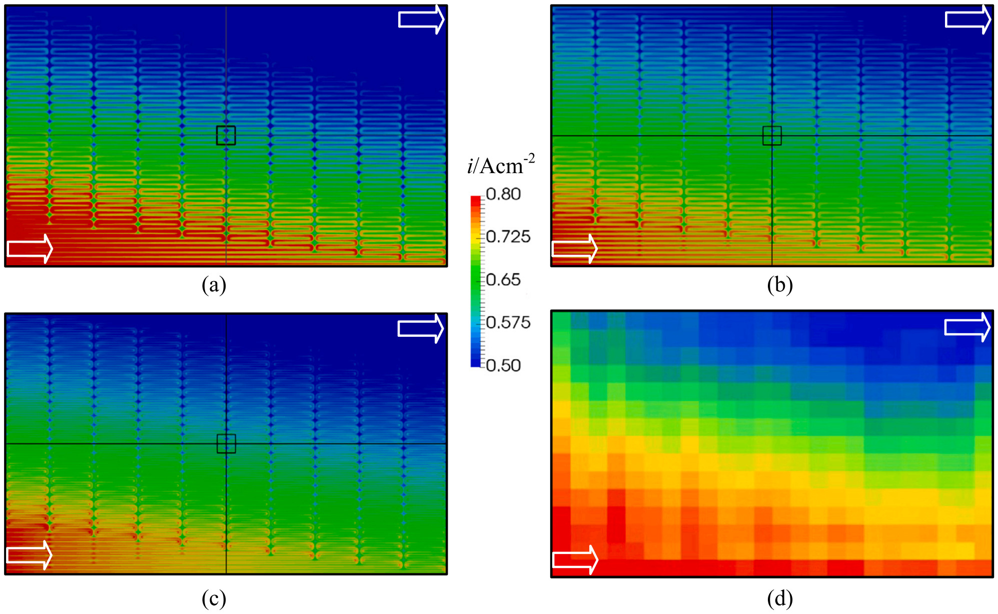
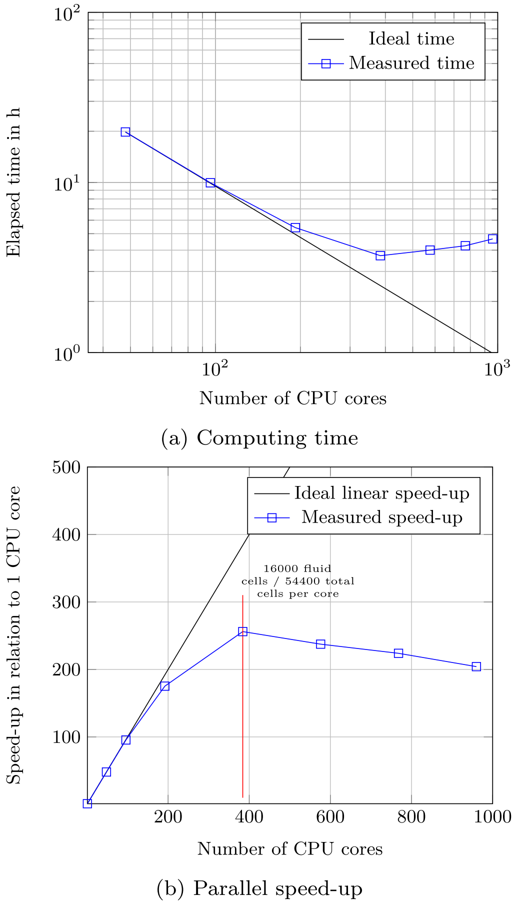


Fig. 14. Comparison of local current density distributions in (a) electrode (`openFuelCell`), (b) electrode (`openFuelCell2`), (c) current collector (`openFuelCell2`), and (d) experiment (S++), for $\bar{i} = 0.5\,\mathrm{A\,cm^{-2}}$, and $\lambda = 2/2$ for anode/cathode sides. Permission obtained from [79].



(a) Computing time

(b) Parallel speed-up

Fig. 15. Strong scaling study for the single channel low-temperature (two-phase system) PEM fuel cell case.

Table 2
Parallel performance.

Number of CPU cores	Computing time in h	Cells per CPU core	Speed-up
48	19.41	435,200	48
96	9.95	217,600	95.46
192	5.42	108,800	175.20
384	3.71	54,400	256.08
576	4.00	36,266	237.43
768	4.25	27,200	223.87
960	4.66	21,760	204.13

higher than the expected number of cells per core proposed by the mentioned rule of thumb. Here it needs to be considered that the fuel cell case is subdivided into multiple regions in which different sets of equations need to be solved. Accordingly, the effort required to solve those equations significantly differs as well. For instance, the computational demand of the semi-implicit pressure-velocity coupling, implemented in a segregated manner, and the incorporation of electrochemistry within the fluid regions is significantly higher compared to solving the energy conservation equation represented by a Laplace/Poisson equation. Taking this into account, the previously mentioned linear scaling point was reached for approximately 32,000 cells in each fluid region and 108,000 cells in the global region per CPU core. With an increasing number of cores and a reduced number of finite volume cells, the actual workload for each CPU core decreases, whereas the amount of communication between the cores, e.g., due to region-region mapping, is increasing. This is the main reason why the simulation might take longer despite using a higher number of CPU cores. The communication, mapping procedures, and pressure-velocity coupling together with the semi-implicit interfacial terms, e.g. drag, were found to be the main bottlenecks for the scalability of the `openFuelCell2` solver.

7. Practical advice

It is suggested that users start with the tutorials (see Appendix B) and make changes accordingly. Starting a new run from scratch, novel users will likely encounter challenges, such as how to create meshes, create zones, split meshes, and define new boundaries, etc. The tutorials employ a built-in method, `blockMesh`, to generate the computational mesh that is the global mesh. The sub-meshes consist of fluid (`air`, `fuel`), solid (`electrolyte`, `interconnect`) and electric (`phiEA`, `phiEC`, `phiI`), as well as sub-zones within themselves. All these steps can be simultaneously expedited by executing the script/command `make mesh`. The region names can be self-renamed, without duplication. If an external mesh system is used, it must include corresponding sub-zones used by the sub-meshes. In addition, the region/zone names have to be correctly set up in the shell file, `preprocessing.sh`.

It is also important to account for the initial and boundary conditions, physical models, and related parameters. The user has to obtain the operating conditions and material properties, determine the flow types/regimes, select the appropriate physical models, and fill in the necessary parameters. An operating condition indicates the initial and boundary conditions of temperature, velocity, pressure, species concentrations, and cell voltages/current density, etc. The physical models employed in the applications refer to the corresponding application in the tutorials.

8. Summary and outlook

This article introduces `openFuelCell2`, a new open-source toolbox designed for simulating electrochemical devices using the finite volume method (FVM). The comprehensive discussion covers theoretical foundations, code development principles, and practical applications. Leveraging the robust open-source library `OpenFOAM`, the toolbox incorporates built-in features like mesh generation, domain decomposition, and parallelization; it is able to simulate key physical processes, including single and two-phase flow, heat and mass transfer, charge transfer, and electrochemical reactions across multiple regions. The program may be employed for numerical simulations of various electrochemical devices, such as fuel cells, electrolyzers, and hydrogen pumps. Released under the GPLv3 licence, the source code is freely available.

To expand its capabilities in multi-physical, multi-region scenarios, the code will undergo further enhancement by integrating additional physics models. For instance,

- The applicability of the Euler-Euler two-phase flow algorithm may vary. Considerations for alternatives like drift-flux [82] or volume-of-fluid [35] procedures will be explored.
- The inclusion of liquid electrolytes and relevant physical models is imperative [83] for simulating lithium batteries.
- Addressing mechanical stresses/strains may involve integrating a stress analysis toolbox such as `solids4foam` [81].

Although these extensions are beyond the immediate scope of the present software development, they constitute promising directions for further cooperative activities, to promote code improvement and optimization. Immediate plans and ongoing testing by the authors include:

- Enhanced performance/speed: A coupled approach for solving matrices in multi-region scenarios, offering a faster alternative to the current segregated approach [84].
- Incorporating the option to model electrochemical devices with thin electrodes, is particularly beneficial for scenarios where resolving electrode thickness is unnecessary.
- Improving code parallel performance.
- Implementing an alkaline water electrolyzer model with a focus on the impact of liquid electrolyte, incorporating various definitions of the Nernst-Planck equations [85], [86].
- Updating and enhancing `twoPhaseSystem`, particularly in terms of stabilization and convergence rate, possibly through the introduction of the partial elimination algorithm [38], [87], [88].
- Enabling the use of `openFuelCell2` with the latest `OpenFOAM` version through multiple forks.

These planned improvements are crucial for expanding the range of potential applications and will be incorporated into future versions of the code.

CRediT authorship contribution statement

Shidong Zhang: Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Steffen Hess:** Software, Writing – original draft. **Holger Marschall:** Investigation, Software, Writing – review & editing. **Uwe Reimer:** Formal analysis, Methodology, Writing – original draft. **Steven Beale:** Conceptualization, Formal analysis, Methodology, Supervision, Validation, Writing – original draft, Writing – review & editing. **Werner Lehnert:** Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgement

This work was partially supported and funded by the AI Data Analytics and Scalable Simulations (AIDAS). Calculations were performed on the HPC hardware of JARA, under grant JARA0070.

Appendix A. Parameters and results for the LT-PEMFC single channel case

Table A.3
Detailed mass flow rates [kg/s] cathode side for $i = 8000$ Am^{-2}.

	`openFuelCell2`	Theory	Rel. error in %
$\dot{m}_{\mathrm{O2,in}}$	3.316e-07	3.316e-07	0
$\dot{m}_{\mathrm{H2O,in}}$	2.625e-07	2.625e-07	0
$\dot{m}_{\mathrm{N2,in}}$	1.050e-06	1.050e-06	0
$\dot{m}_{\mathrm{tot,in}}$	1.644e-06	1.644e-06	0
$\dot{m}_{\mathrm{O2,reac}}$	-6.633e-08	-6.633e-08	2.087e-04
$\dot{m}_{\mathrm{H2O,reac}}$	7.469e-08	7.469e-08	3.030e-04
$\dot{m}_{\mathrm{tot,reac}}$	8.357e-09	8.357e-09	1.051e-03
$\dot{m}_{\mathrm{H2O,diss}}$	-3.580e-08	-3.580e-08	0
$\dot{m}_{\mathrm{O2,out}}$	2.653e-07	2.604e-07	1.854e+00
$\dot{m}_{\mathrm{H2O,out}}$	3.014e-07	2.984e-07	9.832e-01
$\dot{m}_{\mathrm{N2,out}}$	1.050e-06	1.028e-06	2.103e+00
$\dot{m}_{\mathrm{tot,out}}$	1.617e-06	1.587e-06	1.860e+00

Appendix B. Tutorials: usage and structure

The tutorials are designed to help the users familiarize themselves with the functionalities and structures of `openFuelCell2`. They are categorized into different types of applications for fuel cell/electrolyzer that can be simulated with the present code. They include the single-phase cases `HTPEMFC`, `HTPEMEC`, `SOFC`, `SOEC`, and `hydrogenPump`, and the two-phase cases represented by `PEMFC` and `PEMEC`. Together with their operating conditions, they are listed in Table B.4.

Within the tutorial directory, an additional `README.md` file explains the requirements and steps to use the code together with the instructions to run the simulation. For all applications employed with the code, the geometry consists of one straight channel for both the air and the fuel side, as shown in Fig. B.16.

This geometry and mesh are divided into multiple regions, alias `regionType` objects, where each of them stores different quantities depending on the equations that have to be solved. All the created regions and the corresponding cell zones are shown in Table B.5.

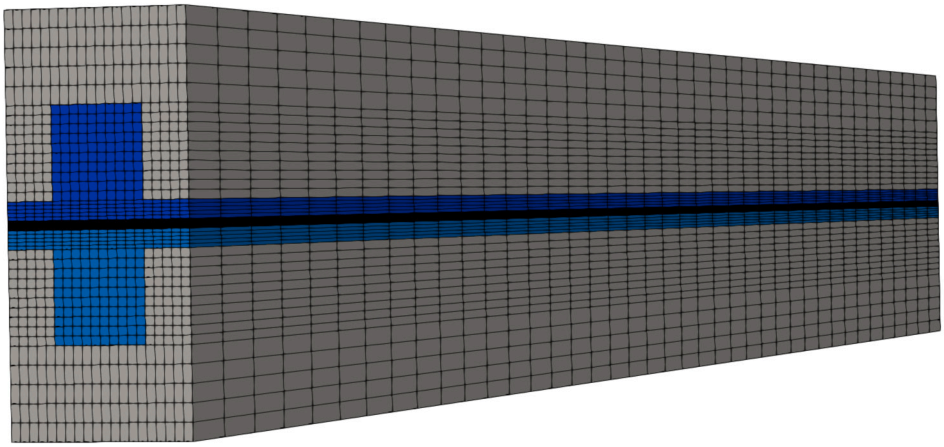


Fig. B.16. Mesh single channel.

In addition to this, Table B.6 depicts the dimensions of the regions and cell zones like the various porous layers.

In the following, the main characteristics and quantities that are needed and read in from diverse input files are at focus. Since there are numerous comments describing the individual quantities in the dictionaries within the tutorial cases, they are only listed but not described here in more detail.

Table B.4
Basic information of the tutorials.

Tutorials	i (A m^{-2})	Air	Fuel	p (bar)	T (K)
CHT	—	One-phase	One-phase	1	973/273
PEMFC	8000	Two-phase	One-phase	1	373/373
PEMEC	-8000	Two-phase	One-phase	1	313/313
HT-PEMFC	8000	One-phase	One-phase	1	443/443
HT-PEMEC	-8000	One-phase	One-phase	1	443/443
SOFC	5000	One-phase	One-phase	1	973/973
SOEC	-5000	One-phase	One-phase	1	973/973

Note: CHT refers to conjugate heat transfer. For a heat exchanger, the comparison of the present code to the standard chtMultiRegionSimpleFoam is prepared.

Table B.5
Regions and cell zones created for the simulations.

Region	Region type	Cell zones
air	fluid	air, cathode, mpl, ccl
fuel	fluid	fuel, anode, mpl, acl
interconnect	solid	-
electrolyte	solid	-
phiEC	electric	bpp, cathode, ccl
phiEC	electric	bpp, anode, acl
phiI	electric	membrane, acl, ccl

Table B.6
Regions and cell Zones created for the simulations.

Parameter	Dimensions
Channel length	40 mm
Rib width	1 mm
Fluid channel width	1 mm
Fluid channel height	1 mm
Membrane thickness	20 μm
Catalyst thickness	20 μm
MPL thickness	20 μm
GDL thickness	200 μm

constant/regionProperties defines the sub-meshes and their types. The information is read and stored in a hash table. Duplication is avoided since the keywords, fluid, solid, and electric are unique. The values are presented in the form of string/word lists, for example, `(air fuel)`. It is not necessary to instantiate all types of sub-meshes in this hash table, i.e. for example, electric regions may be excluded. A typical example of an electrochemical device is shown below:

```
// Key word of the hash table
regions
(
    // keyword: fluid; value: region names:
    air/fuel
    fluid (air fuel)
    // keyword: solid; value: region names:
    electrolyte/interconnect
    solid (electrolyte interconnect)
    // keyword: electric; value: region names:
    phiEA/phiEC/phiI
    electric (phiEA phiEC phiI)
);
```

constant/cellProperties: defines controls for parallel simulations.

```
nx 2; // Division in x direction
ny 2; // Division in y direction
```

constant/fluid_region_name/regionProperties is an important file for the fluid regions. It defines the flow type in either single-phase or two-phase flow. Depending on the exact two-phase system that has been chosen, a higher number of closure models for the drag force, the mass and heat transfer of the interface, etc., need to be defined. The other important entry is the selection of the type of the `phaseModel` for each phase. It defines among other things which physical aspects take place within this phase and which equations are solved.

constant/fluid_region_name/combustionProperties.phase is the key file of the electrochemical reaction. Regarding the file name, the current electrochemical reaction class is constructed from the basic combustion model. In this dictionary the main required entries are

- Type of the combustion model, alias reaction model
- Type of the activation overpotential model
- Type of the Nernst model

constant/electric_region_name/regionProperties: This represents the only file needed for the electric region. It defines the electric conductivity, electric controls, and boundary conditions for potentials and currents. The ion-conducting region provides the control of dissolved water transfer through the membrane. The reference voltage is usually fixed at one boundary, while the voltage of the other side refers to the output value. The operational cell can be simulated under galvanostatic conditions as well as under potentiostatic conditions. By default, it is recommended to use the `galvanostatic` approach.

References

[1] A.Z. Weber, R.L. Borup, R.M. Darling, P.K. Das, T.J. Dursch, W. Gu, D. Harvey, A. Kusoglu, S. Litster, M.M. Mench, R. Mukundan, J.P. Owejan, J.G. Pharoah, M. Secanell, I.V. Zenyuk, A critical review of modeling transport phenomena in polymer-electrolyte fuel cells, J. Electrochem. Soc. 161 (2014) F1254–F1299.

[2] M. Andersson, S.B. Beale, M. Espinoza, Z. Wu, W. Lehnert, A review of cell-scale multiphase flow modeling, including water management, in polymer electrolyte fuel cells, Appl. Energy 180 (2016) 757–778.

[3] S.B. Beale, M. Andersson, C. Boigues-Muñoz, H.L. Frandsen, Z. Lin, S.J. McPhail, M. Ni, B. Sundén, A. Weber, A.Z. Weber, Continuum scale modelling and complementary experimentation of solid oxide cells, Prog. Energy Combust. Sci. 85 (2021) 100902.

[4] S. Beale, R. Jerome, A. Ginolin, M. Perry, D. Ghosh, Towards a virtual reality prototype for fuel cells, PHOENICS J. Comput. Fluid Dyn. Appl. 13 (3) (2000) 287–295.

[5] D.H. Schwarz, S.B. Beale, Calculations of transport phenomena and reaction distribution in a polymer electrolyte membrane fuel cell, Int. J. Heat Mass Transf. 52 (2009) 4074–4081.

[6] M. Andersson, J. Yuan, B. Sundún, Review on modeling development for multiscale chemical reactions coupled transport phenomena in solid oxide fuel cells, Appl. Energy 87 (5) (2010) 1461–1476.

[7] A. Li, C. Song, Z. Lin, A multiphysics fully coupled modeling tool for the design and operation analysis of planar solid oxide fuel cell stacks, Appl. Energy 190 (2017) 1234–1244.

[8] G. Zhang, K. Jiao, Multi-phase models for water and thermal management of proton exchange membrane fuel cell: a review, J. Power Sources 391 (2018) 120–133.

[9] Y. Su, Z. Zhong, Z. Jiao, A novel multi-physics coupled heterogeneous single-cell numerical model for solid oxide fuel cell based on 3D microstructure reconstructions, Energy Environ. Sci. (2022).

[10] D. Hu, J. Liu, F. Yi, Q. Yang, J. Zhou, Enhancing heat dissipation to improve efficiency of two-stage electric air compressor for fuel cell vehicle, Energy Convers. Manag. 251 (2022) 115007.

[11] M. Secanell, A. Putz, P. Wardlaw, V. Zingan, M. Bhaiya, M. Moore, J. Zhou, C. Balen, K. Domican, OpenFCST: an open-source mathematical modelling software for polymer electrolyte fuel cells, ECS Trans. 64 (3) (2014) 655.

[12] R. Vetter, J.O. Schumacher, Free open reference implementation of a two-phase PEM fuel cell model, Comput. Phys. Commun. 234 (2019) 223–234.

[13] N. Weber, L. Knüpfer, S.B. Beale, W. Lehnert, U. Reimer, S. Zhang, P. Ferreira-Aparicio, A.M. Chaparro, Open-source computational model for polymer electrolyte fuel cells, OpenFOAM® J. 3 (2023) 26–48.

[14] S. Beale, W. Lehnert, Electrochemical Cell Calculations with OpenFOAM, Springer, 2022.

[15] S.B. Beale, H.-W. Choi, J.G. Pharoah, H.K. Roth, H. Jasak, D.H. Jeon, Open-source computational model of a solid oxide fuel cell, Comput. Phys. Commun. 200 (2016) 15–26.

[16] R.T. Nishida, S.B. Beale, J.G. Pharoah, Comprehensive computational fluid dynamics model of solid oxide fuel cell stacks, Int. J. Hydrog. Energy 41 (2016) 20592–20605.

[17] R. Nishida, S. Beale, J. Pharoah, L. de Haart, L. Blum, Three-dimensional computational fluid dynamics modelling and experimental validation of the Jülich Mark-F solid oxide fuel cell stack, J. Power Sources 373 (2018) 203–210.

[18] S. Zhang, R. Peters, B.A. Varghese, R. Deja, N. Kruse, S.B. Beale, L. Blum, R. Peters, Modeling of reversible solid oxide cell stacks with an open-source library, ECS Trans. 103 (1) (2021) 569.

[19] S. Zhang, R. Peters, B. Varghese, R. Deja, N. Kruse, S.B. Beale, L. Blum, R. Peters, R.-A. Eichel, Modeling of reversible solid oxide cell stacks with an open-source library, J. Electrochem. Soc. (2022).

[20] S. Yu, S. Zhang, D. Schäfer, R. Peters, F. Kunz, R.-A. Eichel, Numerical modelling and simulation of solid oxide cells and metal interconnect oxidation with OpenFOAM, Energies (2023), https://doi.org/10.3390/en16093827.

[21] S. Zhang, S. Yu, R. Peters, S.B. Beale, H. Marschall, F. Kunz, R.-A. Eichel, Development of a region-to-region coupling procedure for numerical calculations in electrochemical cells, Electrochimical Acta (2023), https://doi.org/10.1016/j.electacta.2023.143275.

[22] S. Zhang, S.B. Beale, U. Reimer, R.T. Nishida, M. Andersson, J.G. Pharoah, W. Lehnert, Simple and complex polymer electrolyte fuel cell stack models: a comparison, ECS Trans. 86 (13) (2018) 287–300.

[23] S. Zhang, U. Reimer, S.B. Beale, W. Lehnert, D. Stolten, Modeling polymer electrolyte fuel cells: a high precision analysis, Appl. Energy 233–234 (2019) 1094–1103.

[24] S. Zhang, U. Reimer, Y. Rahim, S.B. Beale, W. Lehnert, Numerical modeling of polymer electrolyte fuel cells with analytical and experimental validation, J. Electrochem. Energy Convers. Storage 16 (3) (2019) 031002.

[25] S. Zhang, S.B. Beale, Y. Shi, H. Janßen, U. Reimer, W. Lehnert, Development of an open-source solver for polymer electrolyte fuel cells, ECS Trans. 98 (9) (2020) 317.

[26] W. Zhao, V.J. Pinfield, H. Wang, J. Xuan, Z. Niu, An open source framework for advanced multi-physics and multiscale modelling of solid oxide fuel cells, Energy Convers. Manag. 280 (2023) 116791.

[27] H.-J. Bungartz, F. Lindner, B. Gatzhammer, M. Mehl, K. Scheufele, A. Shukaev, B. Uekermann, Precice – a fully parallel library for multi-physics surface coupling, Comput. Fluids 141 (2016) 250–258, advances in Fluid-Structure Interaction.

[28] G. Dhondt, The Finite Element Method for Three-Dimensional Thermomechanical Applications, John Wiley & Sons, 2004.

[29] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, D. Wells, The deal.II finite element library: design, features, and insights, Comput. Math. Appl. 81 (2021) 407–422, https://doi.org/10.1016/j.camwa.2020.02.022, https://arxiv.org/abs/1910.13247.

[30] openFuelCell, https://openfuelcell.sourceforge.io/. (Accessed 25 April 2023).

[31] C. Siegel, Review of computational heat and mass transfer modeling in polymer-electrolyte-membrane (PEM) fuel cells, Energy 33 (9) (2008) 1331–1352.

[32] K. Jiao, X. Li, Water transport in polymer electrolyte membrane fuel cells, Prog. Energy Combust. Sci. 37 (2011) 221–291.

[33] G. Zhang, K. Jiao, Multi-phase models for water and thermal management of proton exchange membrane fuel cell: a review, J. Power Sources 391 (2018) 120–133.

[34] Y. Zhang, G. Agravante, T. Kadyk, M.H. Eikerling, Modeling water phenomena in the cathode side of polymer electrolyte fuel cells, Electrochim. Acta 452 (2023) 142228.

[35] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1) (1981) 201–225.

[36] R.B. Ferreira, D. Falcão, V. Oliveira, A. Pinto, Numerical simulations of two-phase flow in proton exchange membrane fuel cells using the volume of fluid method – a review, J. Power Sources 277 (2015) 329–342.

[37] F.H. Harlow, A.A. Amsden, Numerical calculation of multiphase fluid flow, J. Comput. Phys. 17 (1) (1975) 19–52.

[38] D.B. Spalding, Numerical Computation of Multi-Phase Fluid Flow and Heat Transfer, vol. 1, Recent Advances in Numerical Methods in Fluids, 1980.

[39] H. Rusche, Computational fluid dynamics of dispersed two-phase flows at high phase fractions, Ph.D. thesis, Imperial College London (University of London), 2003.

[40] M. Ishii, T. Hibiki, Thermo-Fluid Dynamics of Two-Phase Flow, Springer Science & Business Media, 2010.

[41] H. Marschall, Towards the numerical simulation of multi-scale two-phase flows, Ph.D. thesis, Technische Universität München, 2011.

[42] H. Marschall, R. Mornhinweg, A. Kossmann, S. Oberhauser, K. Langbein, O. Hinrichsen, Numerical simulation of dispersed gas/liquid flows in bubble columns at high phase fractions using OpenFOAM®. Part II – numerical simulations and results, Chem. Eng. Technol. 34 (8) (2011) 1321–1327.

[43] H. Marschall, R. Mornhinweg, A. Kossmann, S. Oberhauser, K. Langbein, O. Hinrichsen, Numerical simulation of dispersed gas/liquid flows in bubble columns at high phase fractions using OpenFOAM®. Part I – modeling basics, Chem. Eng. Technol. 34 (8) (2011) 1311–1320.

[44] V. Gurau, J. Zawodzinski, A. Thomas, J. Mann, J. Adin, Two-phase transport in PEM fuel cell cathodes, J. Fuel Cell Sci. Technol. 5 (2) (04 2008) 021009.

[45] G. He, P. Ming, Z. Zhao, A. Abudula, Y. Xiao, A two-fluid model for two-phase flow in PEMFCs, J. Power Sources 163 (2007) 864–873.

[46] G. Zhang, K. Jiao, Three-dimensional multi-phase simulation of pemfc at high current density utilizing eulerian-eulerian model and two-fluid model, Energy Convers. Manag. 176 (2018) 409–421.

[47] D. Froning, J. Yu, G. Gaiselmann, U. Reimer, I. Manke, V. Schmidt, W. Lehnert, Impact of compression on gas transport in non-woven gas diffusion layers of high temperature polymer electrolyte fuel cells, J. Power Sources 318 (2016) 26–34.

[48] J. Yu, D. Froning, U. Reimer, W. Lehnert, Apparent contact angles of liquid water droplet breaking through a gas diffusion layer of polymer electrolyte membrane fuel cell, Int. J. Hydrog. Energy 43 (12) (2018) 6318–6330.

[49] J. Yu, D. Froning, U. Reimer, W. Lehnert, Liquid water breakthrough location distances on a gas diffusion layer of polymer electrolyte membrane fuel cells, J. Power Sources 389 (2018) 56–60.

[50] M. Andersson, S.B. Beale, U. Reimer, W. Lehnert, D. Stolten, Interface resolving two-phase flow simulations in gas channels relevant for polymer electrolyte fuel cells using the volume of fluid approach, Int. J. Hydrog. Energy 43 (5) (2018) 2961–2976.

[51] M. Andersson, V. Vukčević, S. Zhang, Y. Qi, H. Jasak, S.B. Beale, W. Lehnert, Modeling of droplet detachment using dynamic contact angles in polymer electrolyte fuel cell gas channels, Int. J. Hydrog. Energy 44 (21) (2019) 11088–11096.

[52] Y. Xu, G. Zhang, L. Wu, Z. Bao, B. Zu, K. Jiao, A 3-D multiphase model of proton exchange membrane electrolyzer based on open-source CFD, Digit. Chem. Eng. 1 (2021) 100004.

[53] M. Andersson, A. Mularczyk, A. Lamibrac, S.B. Beale, J. Eller, W. Lehnert, F.N. Büchi, Modeling and synchrotron imaging of droplet detachment in gas channels of polymer electrolyte fuel cells, J. Power Sources 404 (2018) 159–171.

[54] O. Panchenko, L. Giesenberg, E. Borgardt, W. Zwaygardt, N. Kardjilov, H. Markötter, T. Arlt, I. Manke, M. Müller, D. Stolten, W. Lehnert, Influence of stoichiometry on the two-phase flow behavior of proton exchange membrane electrolyzers, Energies 12 (3) (2019) 350.

[55] S. Zhang, Modeling and simulation of polymer electrolyte fuel cells, 493, RWTH Aachen, Dissertation, Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag,

Jülich, 2020, FZJ-2020-02318, 978-3-95806-472-0, Schriften des Forschungszentrums Jülich Reihe Energie & Umwelt / Energy & Environment, p. 4, xii, 214 S. https://juser.fz-juelich.de/record/877603.

[56] Q. Fang, L. Blum, R. Peters, M. Peksen, P. Batfalsky, D. Stolten, SOFC stack performance under high fuel utilization, Int. J. Hydrog. Energy 40 (2) (2015) 1128–1136.

[57] L. Schiller, A drag coefficient correlation, Z. Ver. Dtsch. Ing. 77 (1933) 318–320.

[58] S.B. Beale, A simple, effective viscosity formulation for turbulent flow and heat transfer in compact heat exchangers, Heat Transf. Eng. 33 (1) (2012) 4–11.

[59] N. Zamel, X. Li, Effective transport properties for polymer electrolyte membrane fuel cells – with a focus on the gas diffusion layer, Prog. Energy Combust. Sci. 39 (1) (2013) 111–146.

[60] S.B. Beale, D.H. Schwarz, M.R. Malin, D.B. Spalding, Two-phase flow and mass transfer within the diffusion layer of a polymer electrolyte membrane fuel cell, Comput. Therm. Sci. Int. J. 1 (2009) 105–120.

[61] M. Leverett, Capillary behavior in porous solids, Trans. AIME 142 (01) (1941) 152–169.

[62] K.S. Udell, Heat transfer in porous media considering phase change and capillarity—the heat pipe effect, Int. J. Heat Mass Transf. 28 (2) (1985) 485–495.

[63] U. Pasaogullari, C.Y. Wang, Liquid water transport in gas diffusion layer of polymer electrolyte fuel cells, J. Electrochem. Soc. 151 (3) (2004) A399–A406.

[64] X. Liu, F. Peng, G. Lou, Z. Wen, Liquid water transport characteristics of porous diffusion media in polymer electrolyte membrane fuel cells: a review, J. Power Sources 299 (2015) 85–96, https://doi.org/10.1016/j.jpowsour.2015.08.092.

[65] X.-D. Wang, Y.-L. Wang, Y. Chen, C. Si, A. Su, D.-J. Lee, Proton exchange membrane fuel cell modeling with diffusion layer-based and sands-based capillary pressure correlations: comparative study, J. Taiwan Inst. Chem. Eng. 45 (4) (2014) 1532–1541, https://doi.org/10.1016/j.jtice.2014.03.011.

[66] S.B. Beale, Mass transfer formulation for polymer electrolyte membrane fuel cell cathode, Int. J. Hydrog. Energy 40 (2015) 11641–11650.

[67] N. Djilali, T. Berning, A 3D, multiphase, multicomponent model of the cathode and anode of a PEM fuel cell, J. Electrochem. Soc. 150 (12) (2003) A1589–A1598.

[68] L. Qingfeng, H. Hjuler, N. Bjerrum, Phosphoric acid doped polybenzimidazole membranes: physiochemical characterization and fuel cell applications, J. Appl. Electrochem. 31 (2001) 773–779.

[69] T.E. Springer, T.A. Zawodzinski, S. Gottesfeld, Polymer electrolyte fuel cell model, J. Electrochem. Soc. 138 (8) (1991) 2334–2342.

[70] S.B. Beale, S. Zhang, M. Andersson, R.T. Nishida, J.G. Pharoah, W. Lehnert, Heat and mass transfer in fuel cells and stacks, in: A. Runchal (Ed.), 50 Years of CFD in Engineering Sciences: A Commemorative Volume in Memory of D. Brian Spalding, Springer, 2020, pp. 485–511.

[71] Sourcetrail, https://github.com/CoatiSoftware/Sourcetrail. (Accessed 24 September 2022), maintained till 14.12.2021.

[72] OpenFOAM, https://openfoam.org. (Accessed 25 April 2023).

[73] OpenFOAM, https://openfoam.com. (Accessed 25 April 2023).

[74] ParaView, https://www.paraview.org. (Accessed 25 April 2023).

[75] woPhaseEulerFoam: Added experimental face-based momentum equation fo… · OpenFOAM/OpenFOAM-dev@16f03f8 (2022-09-24).

[76] T. Holzmann, Mathematics, Numerics, Derivations and OpenFOAM®, Holzmann CFD, Loeben, Germany, 2016.

[77] I.A. Schneider, S. von Dahlen, A. Wokaun, G.G. Scherer, A segmented microstructured flow field approach for submillimeter resolved local current measurement in channel and land areas of a PEFC, J. Electrochem. Soc. 157 (3) (2010) B338–B341.

[78] M. Kvesić, U. Reimer, D. Froning, L. Lüke, W. Lehnert, D. Stolten, 3D modeling of a 200cm2 HT-PEFC short stack, Int. J. Hydrog. Energy 37 (2012) 2430–2439.

[79] S. Zhang, S. Beale, U. Reimer, M. Andersson, W. Lehnert, Polymer electrolyte fuel cell modeling - a comparison of two models with different levels of complexity, Int. J. Hydrog. Energy 45 (38) (2020) 19761–19777.

[80] U. Reimer, W. Lehnert, Y. Holade, B. Kokoh, Irreversible losses in fuel cells, in: V. Hacker, S. Mitsushima (Eds.), Fuel Cells and Hydrogen - from Fundamentals to Applied Research, Elsevier, 2018, pp. 15–40.

[81] P. Cardiff, A. Karač, P. De Jaeger, H. Jasak, J. Nagy, A. Ivanković, v. Tuković, An open-source finite volume toolbox for solid mechanics and fluid-solid interaction simulations (2018-09-03), arXiv:1808.10736.

[82] D. Brennan, The numerical simulation of two phase flows in settling tanks, phdthesis, Imperial College London (University of London), 2001.

[83] N. Weber, C. Duczek, G.M. Horstmann, S. Landgraf, M. Nimtz, P. Personnettaz, T. Weier, D.R. Sadoway, Cell voltage model for Li-Bi liquid metal batteries, Appl. Energy 309 (2022) 118331.

[84] S. Zhang, S. Yu, R. Peters, S.B. Beale, H. Marschall, F. Kunz, R.-A. Eichel, A new procedure for rapid convergence in numerical performance calculations of electrochemical cells 472 143275, https://doi.org/10.1016/j.electacta.2023.143275, https://www.sciencedirect.com/science/article/pii/S0013468623014470.

[85] J. Newman, Electrochemical Systems, 2004, p. 669.

[86] F. Pimenta, M. Alves, rheotool, https://github.com/fppimenta/rheoTool, 2016.

[87] S. Radman, A coarse-mesh methodology for the analysis of one and two-phase nuclear reactor thermal-hydraulics in a multi-physics context, phdthesis, Lausanne, EPFL, LRS, 2021.

[88] A. Cubero, A. Sánchez-Insa, N. Fueyo, A consistent momentum interpolation method for steady and unsteady multiphase flows, Comput. Chem. Eng. 62 (2014) 96–107.