Development of a microfluidic redox flow battery

Youssef Kharchouf

DEVELOPMENT OF A MICROFLUIDIC REDOX FLOW BATTERY

SUBMITTED BY YOUSSEF KHARCHOUF

PhD Thesis in Physics-Chemistry

Sorbonne Université - Université Abdelmalek Essaadi

16 December 2023

Jury Members:

Cristina IoJoiu	Research Director	Université de Grenoble	Reviewer
Amine Moussaoui	Professor	Université Mohammed Premier	Reviewer
Vincent Vivier	Research Director	Sorbonne Université	Examiner
Mustapha Diani	Professor	Université Abdelmalek Essaâdi FSTT	Co-supervisor
Adil Chahboun	Professor	Université Abdelmalek Essaâdi FSTT	Co-director
Mireille Turmine	Associate Professor HDR	Sorbonne Université	Supervisor

The ultimate, hidden truth of the world is that it is something that we make, and could just as easily make differently.

— David Graeber

ABSTRACT

This work aims to advance the integration of microfluidic redox flow batteries and computational modeling to develop energy conversion technologies. A flow-through microfluidic cell is developed using porous electrodes for increased active surface area and reactant utilization. Due to their potential as electrolytes, the Emim[TFSI] ionic liquid is used in an Iron(II)/Quinone system due to the wide electrochemical window and its viscosity that ensures a colaminar flow within the channels of the device. This minimizes reactant crossover and leads to higher performance. The flow-through architecture used consists of forcing the electrolytes to flow within the electrode pores before reaching the outlet. The cell is compared to a similar flow-by planar electrode cell and shows a significant increase in power output.

Furthermore, a Lattice Boltzmann based model is developed to simulate fluid flow and mass transport within the cell. Electrochemical reactions are modelled using Butler-Volmer kinetics and are coupled with mass transport through the source term of the convection-diffusion equation. The model is validated with flow and transport problems with well-known analytical solutions. For the sake of including porous electrodes, a stochastic method for the generation of synthetic porous media for use with this model is also presented and the effects of the porosity on the current across a porous electrode is studied.

RESUMÉ

Ce travail vise à promouvoir l'intégration des batteries à flux redox microfluidiques et de la modélisation numérique pour développer leurs performances. Une cellule microfluidique "flow-through" est élaborée en utilisant des électrodes poreuses afin d'augmenter la surface active et l'utilisation des réactifs. Le liquide ionique Emim[TFSI] est utilisé dans un système Fer(II) / Quinone en raison de sa fenêtre électrochimique large et de sa viscosité qui assure un écoulement colaminaire dans les canaux du dispositif. Le régime laminaire minimise la contamination des réactifs et conduit à une performance accrue. La géométrie

"flow-through" utilisée consiste à forcer les électrolyte à s'écouler à travers les pores des électrodes avant d'atteindre la sortie. Cette cellule est comparée à une cellule similaire à électrodes planes et à écoulement parallèle, et montre une augmentation significative de la puissance générée.

De plus, un modèle basé sur la méthode de Boltzmann sur réseau est développé pour simuler l'écoulement des fluides et le transport de masse à l'intérieur de la cellule. Les réactions électrochimiques sont modélisées en utilisant la cinétique de Butler-Volmer et sont couplées avec le transport de masse à travers le terme source de l'équation de convection-diffusion. Le modèle est validé avec des problèmes d'écoulement et de transport ayant des solutions analytiques bien connues. Dans le but d'inclure des électrodes poreuses, une méthode stochastique pour la génération de milieux poreux synthétiques utilisables avec ce modèle est également présentée, et les effets de la porosité sur le courant à travers une électrode poreuse sont étudiés.

PUBLICATIONS AND CONFERENCES

[1] Youssef Kharchouf. "Development of a Microfluidic Redox Flow Battery: Combining theoretical and experimental methods." In: *74th Annual Meeting of the International Society of Electrochemistry*. International Society of Electrochemistry. Lyon, France, 2023.

[2] Youssef Kharchouf, Rachid Herbazi, and Adil Chahboun. "Parameter's extraction of solar photovoltaic models using an improved differential evolution algorithm." In: *Energy Conversion and Management* 251 (2022), p. 114972.

[3] Rachid Herbazi, Youssef Kharchouf, Khalid Amechnoue, Ahmed Khouya, and Adil Chahboun. "Solar photovoltaic cell parameters extraction using differential evolution algorithm." In: *Proceedings*. Vol. 63. 1. MDPI. 2020, p. 43.

ACKNOWLEDGMENTS

I would like to thank the Partenariat Hubert Curien Toubkal for their financial support of this joint project *PHC Toubkal/21/129 : 45925ZG*.

I first extend my sincere appreciation to the jury members for their time and diligent efforts in reviewing this work.
Special acknowledgment is due to Mr Soufiane Derfoufi, Mme Cristina IoJoiu, M. Amine Moussaoui for their commitment to evaluating and assessing my thesis.

I extend my profound gratitude to M. Vincent Vivier for his invaluable contributions to this thesis. His readiness to engage in scientific discussions significantly enriched my research and provided crucial support at every stage of this project.
Heartfelt thanks to M. Kieu Ngo for his generous availability and assistance. His consistently kind and supportive demeanor was helpful throughout the course of my stays in Paris.
I am indebted to M. Adil Chahboun, my co-director in Morocco, whose guidance and unwavering support were vital to my work and progress. His mentorship played a crucial role in the achievement of this thesis.
Acknowledging the efforts of M. Mustapha Diani, I appreciate his crucial role in efficiently orchestrating key aspects of this project in Morocco, ensuring its smooth and successful completion.
To my supervisor, Mme Mireille Turmine, I owe a debt of gratitude for her exceptional guidance, patience, and clarity in elucidating complex concepts. Her meticulous assistance in all facets of this project was key to its realization.

I express gratitude to my colleagues at the LRS in Paris and the CMN team in Tangier for creating a nurturing and collaborative environment. Their collective wisdom and camaraderie have greatly enriched my journey. A special mention for my colleagues at 326 – Albert, Ana, Walid, and Fadoua – whose camaraderie made my time at LRS a memorable experience.
Lastly, my deepest gratitude is for my parents and my favorite trio Kawtar, Mouna and Hajar, as well as Fatima and Aymane, whose unwavering love and support have been the bedrock of

my strength. Additionally, a heartfelt thank you to Farah for her enduring encouragement and understanding, which have been invaluable throughout this journey.

CONTENTS

1 Chapter 1: Literature Review 17
1.1 Redox Flow Batteries 18
1.1.1 Redox Flow Battery Chemistry 20
1.1.2 Redox Flow Batteries with Ionic Liquids 23
1.2 Microfluidics 28
1.2.1 Microfluidics as an extension of MEMS 28
1.2.2 The advent of microfluidic fuel cells and batteries 31
1.3 the Lattice Boltzmann Method for Fluid Dynamics and Mass Transport 35
1.4 Concluding remarks 38
2 Chapter 2: Experimental Prototyping 41
2.1 Performance evaluation through polarization curves 42
2.2 Materials 44
2.2.1 Electrolytes 45
2.2.2 Resin printing 49
2.3 Channel Design and geometry 50
2.4 Fabrication and assembly process 53
2.5 Experimental Setup 57
2.6 Results 59
2.7 Conclusion 64
3 Chapter 3: Theoretical Framework 65
3.1 Navier-Stokes equations 66
3.2 the Lattice Boltzmann Equation at the macroscopic limit 67
3.2.1 Boundary Conditions 75
3.3 LBM for advection-diffusion 77
3.4 Electrochemical model 79
3.5 Coupling Navier-Stokes, advection-diffusion and electrochemistry 81
3.6 Summary 82
4 Chapter 4: Implementation and Results 85
4.1 General remarks about implementation 85
4.2 Validation with analytical solutions 86
4.2.1 Cottrell Experiment 86
4.2.2 The Graetz problem 91
4.3 Synthetic porous geometries 94
4.3.1 Tortuosity 99
4.3.2 Anisotropy 101

4.3.3 Cyclic Voltammetry in LBM 105
4.4 Summary 110
5 Conclusion and Future Work 113

I Appendix
A Appendix: LBM implementation in Python 117
A.1 Standard Navier-Stokes Solver with torch 117
A.2 Convection-Diffusion equation solver 121
A.3 Cyclic voltammetry cycling in LBM 125
A.4 Cottrell current calculation 132
A.5 Graetz problem and the Leveque approximation 138
A.6 Various utility functions 144
B Appendix: Synthetic pore microstructure generation 151
B.1 Stochastic pore microstructure generation 151
B.2 Pressure drop calculation across pore space 153
B.3 Tortuosity and streamlines 156
B.4 CAD technical drawings 159

Bibliography 161

LIST OF FIGURES

Figure 1 Schéma d'une cellule "flow-through" avec des électrodes poreuses 4

Figure 2 Modèles 3D des pièces de la cellule. (a) Pièce avec la géométrie des canaux. (b) Pièce de couverture contenant des fentes pour les électrodes et des trous d'entrée et de sortie d'électrolytes. 4

Figure 3 Montage expérimental utilisé pour l'opération de la batterie. 5

Figure 4 Montage électrique de la batterie redox flow RFB pour la mesure de la courbe de polarisation. Le paramètre variable est la résistance de charge R. Le courant et la tension sont mesurés avec des multimètres connectés en série et en parallèle respectivement. 6

Figure 5 Interface laminaire bien définie entre les deux électrolytes dans la cellule "flow-through" observée dans le canal principal de la cellule. 6

Figure 6 La courbe de polarisation pour une cellule flow-through avec des electrodes poreuses. 7

Figure 7 Comparaison de la puissance maximale obtenue expérimentalement pour la cellule à électrodes planes et les deux cellules "flow-through" avec différentes distances de séparation des électrodes. 8

Figure 8 Résultats de la simulation du problème de diffusion de Cottrell. (a) Variation du courant à travers l'interface en fonction du temps. (b) Erreur relative entre la solution analytique et la solution numérique. 10

Figure 9 (a) Exemple d'une structure poreuse générée avec la méthode stochastique montrant l'interface active entre solide et fluide. (b) Domaine de simulation utilisé avec Lattice Boltzmann pour représenter une cellule électrochimique avec une électrode poreuse. 10
Figure 10 (a) Courbes de voltamétrie cyclique effectuée avec le modèle Lattice Boltzmann pour différentes porosités. (b) Courant maximal atteint en fonction de la porosité. 11
Figure 1.1 Diagram of a conventional Redox Flow Battery. 18
Figure 1.2 A typical $90KWh$-rated vanadium redox flow battery system (Sumimoto Electric Industries) [2]. The cell stacks on the left are composed of 100 cells in series, while the electrolyte storage tanks store 4000 L of electrolyte. 19
Figure 1.3 Examples of common ionic liquid cations and anions. 24
Figure 1.4 Schematic of a macroscale membraneless biphasic redox flow battery. (a) Schematic representation showing an upper aqueous phase and a lower ionic liquid phase (b) Schematic of a redox flow battery with a horizontal design based on immiscible electrolytes (From [65]). 27
Figure 1.5 Different scales of fluid dynamics, from [69] 28
Figure 1.6 Different flow types for low and high Reynolds numbers. (From [67]) 30
Figure 1.7 Schematic of a Y shaped membraneless microfluidic redox flow battery. (from [12]) 32
Figure 1.8 Schematic of the membraneless microfluidic fuel cell by Ferrigno et al. (2002) [71] 33
Figure 1.9 Membraneless microfluidic redox flow battery using ionic liquids by Chaabene et al. (2023) [13] 34

Figure 2.1 The electrical measurement setup for a polarization and power curve of a redox flow battery. The RFB is the power source, while the varying parameter is the load resistance R. Current and Voltage are measured using multimeters connected in series and in parallel, respectively. 42
Figure 2.2 A typical polarization curve for a redox flow battery showcasing the different limiting regimes 43
Figure 2.3 Picture of the carbon felt sheets that are used in this work. The sheets are 6 mm thick (uncompressed) and can be manually cut in the desired shape according to the device geometry. 45
Figure 2.4 Sketch of the planar electrode PDMS/-glass microfluidic redox flow battery 51
Figure 2.5 Sketch of the flow through design architecture with porous electrodes 52
Figure 2.6 Computer-aided Design (CAD) model of the piece containing the flow channels 53
Figure 2.7 CAD model of the cover piece 55
Figure 2.8 Cover piece with electrodes inserted into their respective slots. (a) The printed part with the copper contacts and carbon felt electrode. (b) CAD model of the same piece with electrodes shown in black. Note the slight extrusion of the electrodes above the main plane. 56
Figure 2.9 The full device with the two pieces glued together. 57
Figure 2.10 Device setup with the pressure based flow controller 58
Figure 2.11 Experimental setup with the device interfaced with the pressure based flow controller through capillary tubes and the external circuit 59
Figure 2.12 The laminar interface between the two reactants 60
Figure 2.13 Comparison between the polarization curves and the power output of the flow-through device and the planar flow-by device. 61

Figure 2.14 Polarization curve of the flow through device with the maximum performance that was obtained. 62
Figure 2.15 Comparison of the maximum power output obtained experimentally for the flow-by device with planar electrodes and the two flow through devices with different electrode separation distances. 63
Figure 3.1 LBM Lattices in one, two, or three dimension: (a) D1Q3, (b) D2Q5, (c) D2Q9 and (d) D3Q15 (*Note: rest velocities not shown*). Refer to [125] for a more comprehensive list of lattice velocity sets 70
Figure 3.2 State of populations neighboring a central node before streaming (left). State of the same populations after streaming to the central node, and right after collision (right) 75
Figure 3.3 Illustration of the bounce-back rule in the lattice Boltzmann method. The filled circles represent solid nodes, the dotted line represents the wall location, and f_i and f_i^* are the populations before and after bounce-back, respectively 76
Figure 3.4 (a) Overview of the numerical model that integrates fluid flow, species transport and electrokinetics. (b) Breakdown of the loop that constitutes the Navier-Stokes solver. It is executed until a predefined maximum simulation time t_{max} is reached, which is chosen such that the flow is at a steady-state. 83
Figure 4.1 Geometry of the Cottrell experiment. The electrode is shown in blue, while the white region represents the bulk electrolyte solution and black represents the walls of the cell. 88
Figure 4.2 Comparison between the analytical solution for the Cottrell current and the Lattice Boltzmann model 89
Figure 4.3 Concentration profiles at different times for the Cottrell experiment. 89

Figure 4.4 Ratio of the Lattice Boltzmann calculated current over the analytical solution $I_{\text{LBM}} / I_{\text{Cottrell}}$ as a function of time 90
Figure 4.5 Evolution of the error between the Lattice Boltzmann calculated concentration field and the analytical solution given by equation 4.3 90
Figure 4.6 Geometry of the Graetz problem. The red line represents the boundary layer where the velocity profile is linear. 92
Figure 4.7 Solution of the Graetz problem with Lattice Boltzmann Method (LBM) and comparison with the Leveque approximation 93
Figure 4.8 Relative error between the LBM calculated current density and the analytical solution 94
Figure 4.9 Example of a 200×200 matrix of random numbers sampled from a normal gaussian distribution 95
Figure 4.10 (a) Diagram of the resulting matrix after Gaussian smoothing and (b) boolean matrix after cutoff 96
Figure 4.11 Generated porous geometry with a blue color code for electrode surface 97
Figure 4.12 Generated sample structures for a fixed porosity of $\epsilon = 0.4$ in (a), (b) and (c) and porosity of $\epsilon = 0.7$ in (d), (e) and (f) 97
Figure 4.13 Influence of the smoothing kernel standard deviation σ and the porosity ϵ on the surface area. The surface area $S*$ is the sum total of all interface nodes and is normalized by the maximum value obtained in this parameter space. Surface area is proportional to σ^{-1} up to a constant C as shown by the dashed line 98
Figure 4.14 Streamlines of the flow in (a) a Poiseuille flow and (b) a generated porous geometry with $\epsilon = 0.7$ 100

Figure 4.15 Different kernels used in the gaussian smoothing step by varying the covariance matrix Σ and the rotation angle θ. The resulting microstructures are shown in the bottom row. (a) The original kernel with scalar standard deviation creates an isotropic geometry (b) A kernel with a larger standard deviation along the horizontal axis adds a preferred horizontal direction (c) A kernel with a larger standard deviation along the horizontal axis is rotated by $\frac{\pi}{4}$ radians creates a preferred direction along that angle. 102
Figure 4.16 Pressure drop and lattice velocity across a channel for a (a) vertically and (b) horizontally oriented microstructure. 104
Figure 4.17 Pressure drop across a channel for different orientations of the microstructure. 105
Figure 4.18 Hydraulic tortuosity of the microstructure as a function of the kernel angle θ across multiple measurements for each angle. 105
Figure 4.19 Overall simulation geometry of the porous electrode. The active surface area nodes are distinguished by a blue color coding, while the bounce-back nodes of LBM are represented in black. 106
Figure 4.20 Generated pore structures with different porosities ρ ranging from $\rho = 0$, to a porosity of $\rho = 0.9$. The interface between the solid and fluid regions is indicated in blue. 108
Figure 4.21 Cyclic voltammetry plots for the porous electrode at different levels of porosity. 109
Figure 4.22 Maximum current generated as a function of porosity during cyclic voltammetry runs on the porous electrode. 110

LIST OF TABLES

Table 1.1 Comparison of different water-based RFB systems 21
Table 2.1 Peak potential drift and standard potentials of the redox species of Quinone and Iron(II) Chloride in Emim[TFSI] [13]. 49

CODE LISTINGS

gfx/code/navierstokeslattice.py 117
gfx/code/depletion_lattice.py 121
gfx/code/echemlattice.py 125
gfx/code/cottrell.py 132
gfx/code/leveque.py 138
gfx/code/util.py 144
gfx/code/poregen_gauss.py 151
gfx/code/p_drop_calc.py 153
gfx/code/tortuosity_measurement.py 156
gfx/code/streamlines.py 159

ACRONYMS

CAD Computer-aided Design
CE Coulombic Efficiency
CFD Computational Fluid Dynamics
IEM Ion Exchange Membrane
IL Ionic Liquid
LBM Lattice Boltzmann Method
MD Molecular Dynamics
MEMS MicroElectroMechanical Systems

MMRFB Membraneless Microfluidic Redox Flow Battery

PDMS Poly(dimethylsiloxane)

RFB Redox Flow Battery

SLA Stereolithography

RÉSUMÉ ÉTENDU EN FRANÇAIS

Dans le cadre des problématiques environnementales actuelles, le développement de technologies de stockage d'énergie propres et efficaces est l'un des sujets principaux liés au domaine de l'électrochimie [1]. Par conséquent, les piles à combustible, les batteries rechargeables et les super-condensateurs sont au cœur des recherches visant à apporter des solutions à la crise énergétique en permettant une utilisation flexible et efficiente de l'énergie électrique. Les études sur les batteries conventionnelles sont principalement axées sur les systèmes mobiles et de transport, où la densité d'énergie et la compacité du système constituent des préoccupations majeures [2]. Cependant, il existe un besoin croissant de systèmes de stockage d'énergie plus orientés vers les applications stationnaires et le stockage sur le réseau pour lesquels la densité d'énergie et la taille sont des paramètres bien moins critiques [3]. Ainsi, les batteries redox flow se sont avérées être une technologie prometteuse pour le stockage d'énergie à grande échelle, grâce à leur flexibilité et à leur excellente cyclabilité. Le meilleur exemple étant la batterie redox flow à base de vanadium qui a connu un succès commercial notable [4]. Néanmoins, ces systèmes souffrent encore d'une faible densité d'énergie et de la contamination à travers la membrane qui sert à séparer l'anolyte du catholyte tout en étant perméable aux ions [5]. Un autre inconvénient commun aux batteries redox flow aqueuses est la fenêtre de stabilité électrochimique de l'eau qui limite la tension circuit ouvert du système [6]. Une solution possible à ce problème consiste à utiliser des électrolytes organiques ou non aqueux, qui présentent une fenêtre électrochimique plus large, mais qui introduisent d'autres défis tels que les risques d'inflammation et les pertes par évaporation [7]. À cet égard, un certain nombre de chercheurs ont envisagé l'utilisation de liquides ioniques comme électrolytes pour ces batteries. En raison du grand nombre de combinaisons possibles d'anions et de cations, les liquides ioniques peuvent être mis au point pour une large gamme de propriétés comme la non-volatilité et une large fenêtre électrochimique [8].

Parallèlement la recherche d'électrolytes adaptés, un autre axe de recherche est l'optimiser les batteries via leur membrane, qui est un composant cher et souvent limitant en termes de perfor-

mance [9, 10]. Face aux contraintes très exigeantes imposées aux membranes, une partie de la littérature a été consacrée à l'étude de conceptions de batteries sans membrane [11]. Ces concepts de batteries reposent sur différentes approches pour prévenir la contamination au sein de la cellule, telles que la séparation de phase ou, plus immédiatement pertinent pour ce travail, l'utilisation de flux colaminaires à des échelles microfluidiques [12]. Dans cette approche, la nature laminaire de l'écoulement des fluides à petite échelle est exploitée pour minimiser la contamination et éviter le besoin d'une membrane. Cependant, cette approche modifie également les applications potentielles de la batterie, car les tailles requises pour les canaux microfluidiques afin de maintenir un écoulement laminaire ne conviennent évidemment plus pour le stockage d'énergie à grande échelle. En effet, ces cellules seront plus adaptées à des applications mobiles et portables.

Dans ce contexte, l'objectif de ce projet est de poursuivre les travaux de Chaabene et al. [13] dans l'étude de l'utilisation de liquides ioniques en tant qu'électrolyte pour les batteries redox flow microfluidiques, en intégrant notamment des techniques issues du domaine des piles à combustible, comme l'utilisation d'électrodes poreuses pour augmenter la surface de contact avec l'électrolyte et l'utilisation des réactifs. Ceci afin de fabriquer un prototype d'un batterie redox flow microfluidique à base de liquide ionique. Le deuxième axe de ce projet est de développer un modèle numérique pour ces dispositifs afin d'accompagner les travaux expérimentaux et de guider l'optimisation des conceptions de batteries futures. En tenant compte de ces deux axes, ce manuscrit est organisé comme suit :

- *Chapitre I: Etude bibliographique*

Ce chapitre est divisé en trois parties. La première concerne l'état de l'art des batteries redox flow conventionnelles, le principe de fonctionnement, les reactions et les électrolytes utilisés, avec une partie consacrée aux batteries utilisant des liquides ioniques et les concepts sans membrane. La deuxième partie aborde le domaine de la microfluidique, et le principe d'écoulement laminaire que les dispositifs microfluidiques permettent d'atteindre, ainsi que l'avantage que ce regime d'écoulement apporte aux batteries redox flow, à savoir la minimisation de la contamination entre l'anolyte et le catholyte. Une revue bibliographique des travaux existants sur les batteries redox flow microfluidiques est également présentée. La troisième et dernière partie est consacrée aux modèles numériques et aux méthodes de

simulation utilisées dans la littérature pour simuler les systèmes électrochimiques.

- Chapitre II: Prototype expérimental

Ce chapitre concerne la partie expérimentale de ce travail. On commence avec la présentation des matériaux et méthodes utilisées pour la fabrication du prototype de batterie redox flow microfluidique. Dans la continuité de la cellule redox flow de Chaabene et al [13], le liquide ionique Emim[TFSI] a été utilisé pour sa conductivité élevée, sa large fenêtre électrochimique ainsi que sa viscosité relativement faible qui permet d'assurer un écoulement laminaire dans les canaux de la cellule. Pour controller la pureté, la teneur en eau et en général la reproductibilité du travail experimental, on a choisi de le synthétiser. Une section de ce chapitre est donc consacrée à la synthèse du liquide ionique. En ce qui concerne les electrodes, on a utilisé des électrodes en feutre de carbone de faible coût. Contrairement à la cellule de Chaabene et al. qui comprend des electrodes planes en or déposées sur un substrat en verre, nous avons opté pour des electrodes poreuses pour augmenter la surface active disponible pour les réactions électrochimiques, ce qui permettrait d'augmenter la performance de la cellule.

La structure du dispositif est construite en impression 3D (stéréolithographie), qui permet de fabriquer des prototypes à bas coût et avec une résolution de 25 μm, ce qui est largement suffisant pour notre application. En effet, puisque nous utilisons un liquide ionique de viscosité élevée, les canaux peuvent être relativement grands (à l'échelle de 1 mm) tout en conservant un écoulement laminaire qui evite le mélange des électrolytes. La géométrie utilisée est une cellule "flow-through" (figure 1) qui permet d'obtenir une densité de courant élevée, et qui a montré de bonnes performances pour les batteries redox flow aqueuses et les piles à combustible.

Pour fabriquer une cellule avec cette géométrie. Le dispositif est composé de deux pièces dont les modèles 3D sont présentés dans la figure 2. La première pièce contient les canaux avec la géométrie flow-through et des rainures pour distribuer l'écoulement de manière uniforme le long des electrodes. La deuxième pièce est une pièce de couverture qui contient des fentes pour les électrodes et des trous d'entrée et de sortie d'électrolytes. Les deux pièces sont assemblées avec de la même résine utilisée en impression 3D, grâce à un traitement avec une lampe UV pour la solidifier.

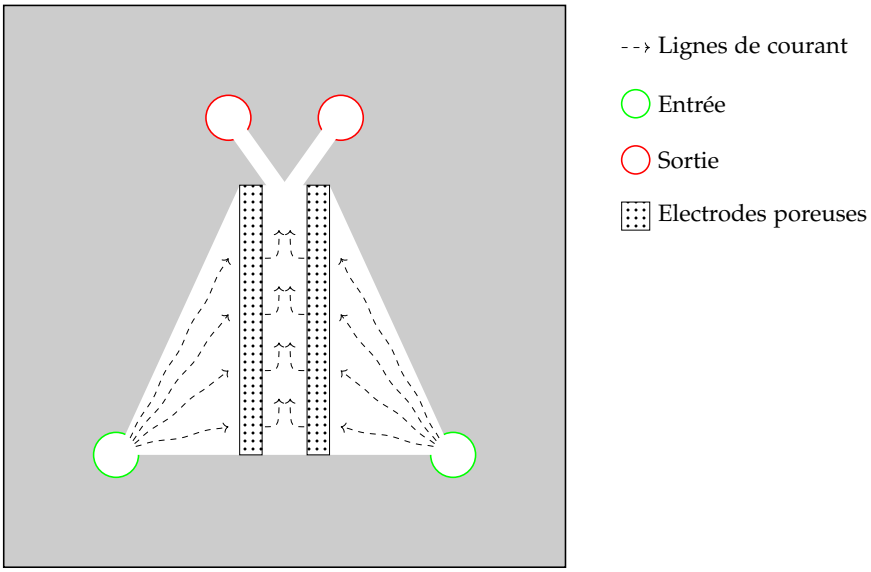
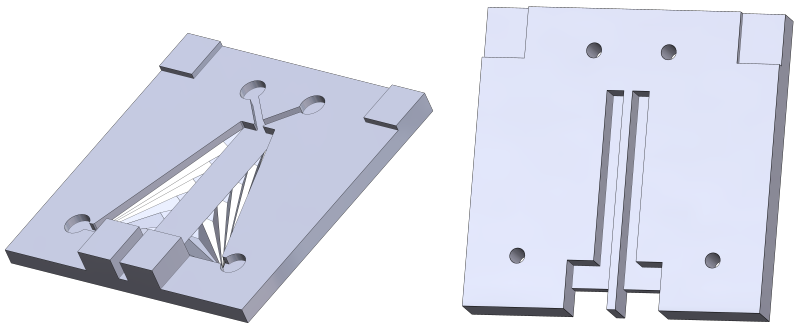


Figure 1: Schéma d'une cellule "flow-through" avec des électrodes poreuses



(a) (b)

Figure 2: Modèles 3D des pièces de la cellule. (a) Pièce avec la géométrie des canaux. (b) Pièce de couverture contenant des fentes pour les électrodes et des trous d'entrée et de sortie d'électrolytes.

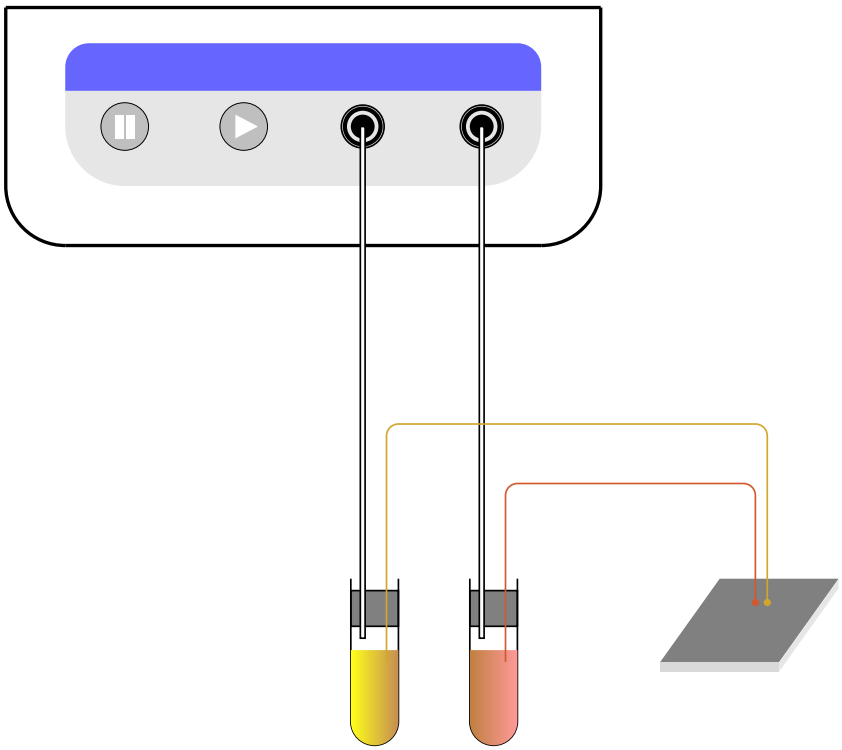


Figure 3: Montage expérimental utilisé pour l'opération de la batterie.

Une fois la cellule assemblée, on procède au montage expérimental qui permet d'évaluer ses performances en traçant sa courbe de polarisation. La cellule est connectée avec des tubes capillaires à des reservoirs contenant les électrolytes. Ces réservoirs sont eux-mêmes connectés à un contrôleur microfluidique qui permet de contrôler le débit des électrolytes en appliquant une pression constante sur les réservoirs. La figure 3 montre le montage microfluidique utilisé pour les tests de la cellule. Les contacts électriques de la cellule sont connectés à un voltmètre en parallèle et un ampèremètre en série pour mesurer la tension et le courant à travers la cellule. Une résistance variable est connectée en série avec la cellule pour tracer la courbe de polarisation comme le montre la figure 2.1.

La section 2.6 décrit les résultats obtenus avec ce montage expérimental pour les cellules flow-through pour un système $FeCl_2$/Quinone avec des concentrations de 0.1 M et 0.3 M respectivement. On a pu mettre en évidence l'interface colaminaire entre les deux électrolytes, ce qui confirme que l'écoulement dans les canaux est effectivement en régime laminaire (figure 5).

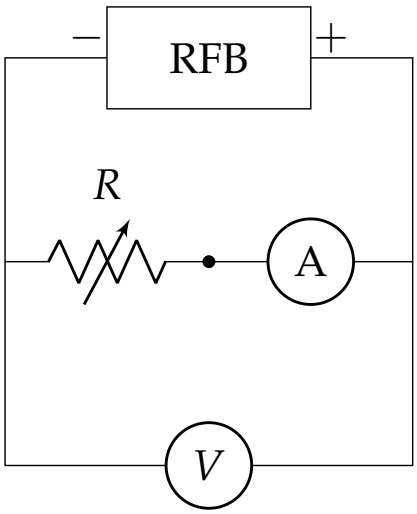


Figure 4: Montage électrique de la batterie redox flow RFB pour la mesure de la courbe de polarisation. Le paramètre variable est la résistance de charge R. Le courant et la tension sont mesurés avec des multimètres connectés en série et en parallèle respectivement.

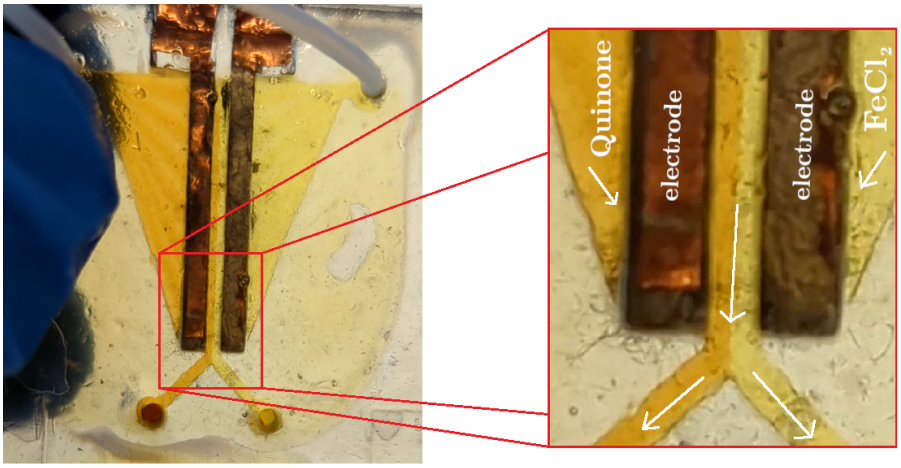


Figure 5: Interface laminaire bien définie entre les deux électrolytes dans la cellule "flow-through" observée dans le canal principal de la cellule.

En raison du caractère manuel du processus d'assemblage, il est difficile de garantir une performance constante entre les différentes cellules. La figure 6 montre la courbe de polarisation d'un cellule optimisée dont la distance entre les électrodes est de $1,5\ mm$. On observe que la tension circuit ouvert et le courant court-circuit atteignent $0,3\ V$ et $360\ \mu A$, respectivement. La puissance maximale obtenue est d'environ $36\ \mu W$.

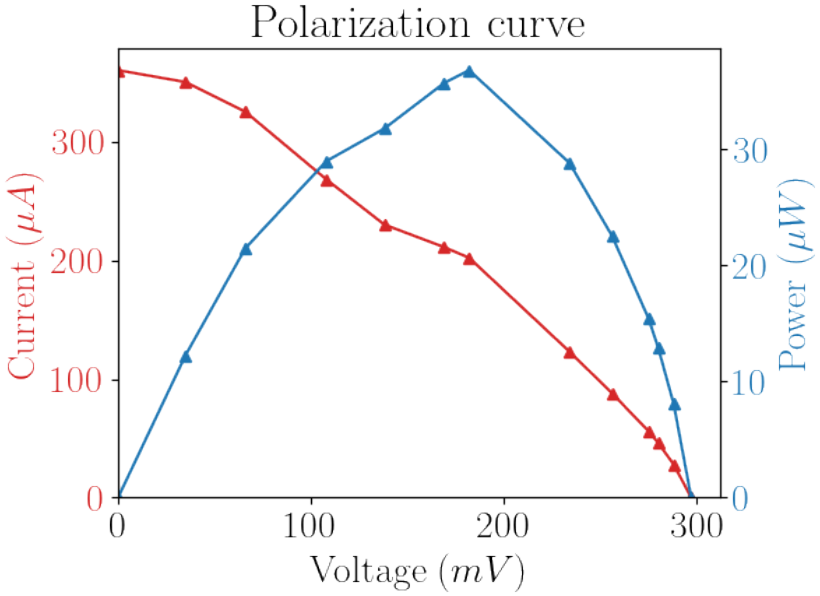


Figure 6: La courbe de polarisation pour une cellule flow-through avec des electrodes poreuses.

La figure 7 compare la performance d'une cellule avec des électrodes planes avec deux cellules "flow-through" avec des distances entre les électrodes de 2 mm et 1,5 mm. On observe que, dans des conditions identiques, la cellule avec des électrodes poreuses a une puissance maximale considérablement plus élevée que la cellule avec des electrodes planes. De même, la cellule dont la distance entre les électrodes est réduite a une puissance maximale plus élevée. Cela est dû au fait que la résistance interne de la cellule engendrée par la résistance de l'électrolyte a diminué. Cependant, il est difficile de rapprocher davantage les électrodes avec un assemblage manuel à cause du risque de court-circuit. Un problème fréquent qui survient lors de l'assemblage est la présence de fibres égarées qui se prolongent hors de l'électrode et offrent un chemin à faible résistance entre les deux électrodes. Cela provoque souvent un court-circuit à travers la cellule et conduit à une réduction significative des performances.

Il est donc évident grace à ces prototypes que l'utilisation des électrodes poreuses est possible aux échelles millimétriques en utilisant un liquide ionique comme électrolyte et tout en gardant un régime laminaire dans les canaux de la cellule. Cependant, l'assemblage manuel des cellules est un processus fastidieux et peu reproductible. Ce prototype non-optimisé peut donc

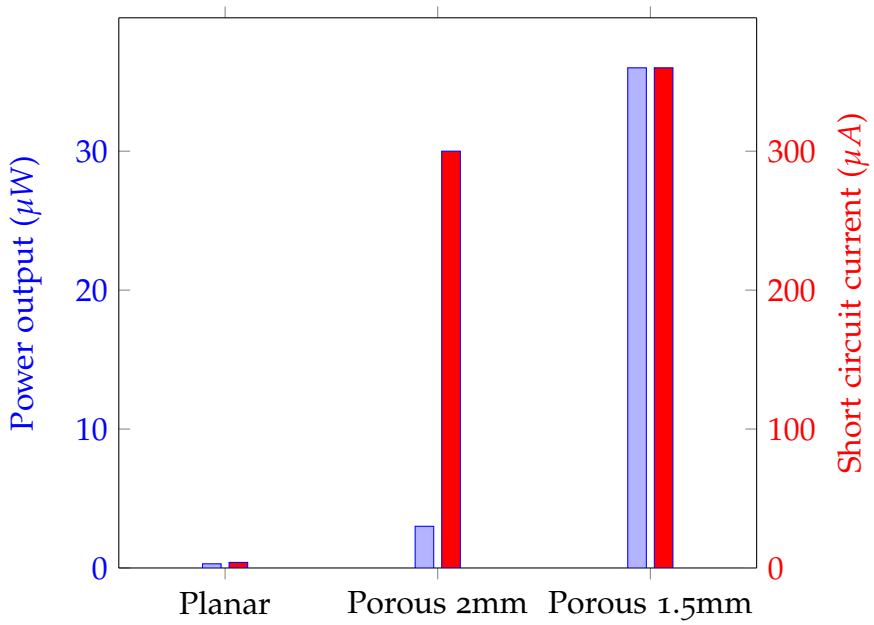


Figure 7: Comparaison de la puissance maximale obtenue expérimentalement pour la cellule à électrodes planes et les deux cellules "flow-through" avec différentes distances de séparation des électrodes.

servir comme point de départ pour des travaux futurs plus systématiques pour optimiser l'électrolyte, les couples redox utilisés ainsi que la géométrie de la cellule.

- *Chapitre III: Fondements théoriques du modèle*

Ce chapitre est consacré à la présentation du modèle numérique développé dans ce travail. On commence par la présentation des équations de Navier-Stokes qui sont le noyau du modèle dans la section 3.1. Ensuite, on montre, dans la section 3.2, comment ces équations peuvent être retrouvées à la limite macroscopique de l'équation de Boltzmann avec l'analyse de Chapman-Enskog, ce qui fait de la méthode Lattice Boltzmann un solveur des equations Navier-Stokes.

A la suite de cela, on montre comment il est possible de modifier la méthode Lattice Boltzmann pour résoudre l'équation de convection-diffusion dans la section 3.3, pour établir un modèle de transport de la matière. Ceci est accompli en faisant l'analogie entre les équations de Navier-Stokes et l'équation de convection-diffusion, en considérant Navier-Stokes comme une équation de convection-diffusion de la quantité de mouvement $\rho\mathbf{u}$.

La section 3.4 est dédiée au modèle électrochimique qui est basé sur la cinétique de Butler-Volmer, et qui est couplée avec le modèle de transport de la matière à travers le terme source de l'équation de convection-diffusion. Le potentiel d'équilibre est calculé à partir de l'équation de Nernst et correspond au cas où les courants anodique et cathodique s'annulent. Il est important de noter que le transport de matière dans ce cas inclut la diffusion et la convection, mais pas la migration électrique. En effet, la concentration faible des réactifs ($\sim 0,1\ M$) par rapport à l'électrolyte support, qui est dans ce cas le liquide ionique agissant comme solvant et électrolyte à la fois, fait que la migration dûe au champ électrique est négligeable, ce qui simplifie le modèle considérablement.

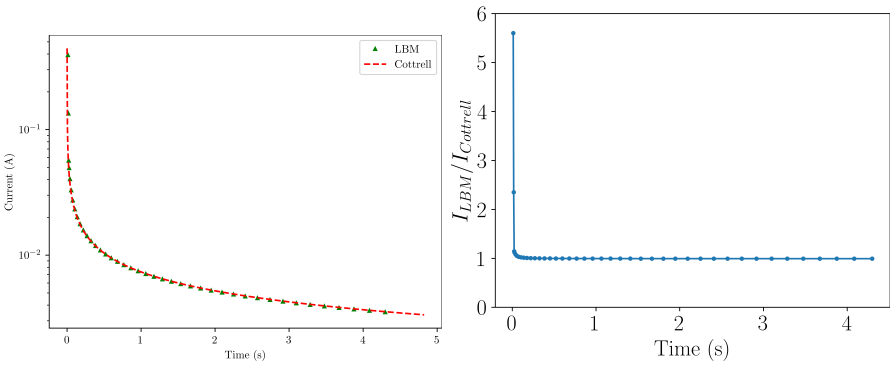
Enfin, la section 3.5 présente la méthode de couplage entre les différents éléments de ce modèle avec Lattice Boltzmann: Navier-Stokes, convection-diffusion et électrochimie.

- *Chapitre IV: Implémentation et Résultats*

Ce dernier chapitre est consacré à l'implémentation du modèle numérique présenté dans le chapitre précédent. On commence avec une discussion de quelques considérations pratiques pour l'implémentation du modèle dans la section 4.1. Ensuite, on présente les résultats de la validation du modèle avec des problèmes d'écoulement et de transport avec des solutions analytiques connues dans la section 4.2. La figure 8 montre le résultat de la simulation d'un problème de diffusion de Cottrell, qui est un problème classique de diffusion dans un milieu semi-infini avec une concentration initiale uniforme et une concentration à l'interface qui varie avec le temps selon la loi de Cottrell. On observe que le modèle reproduit bien la solution analytique.

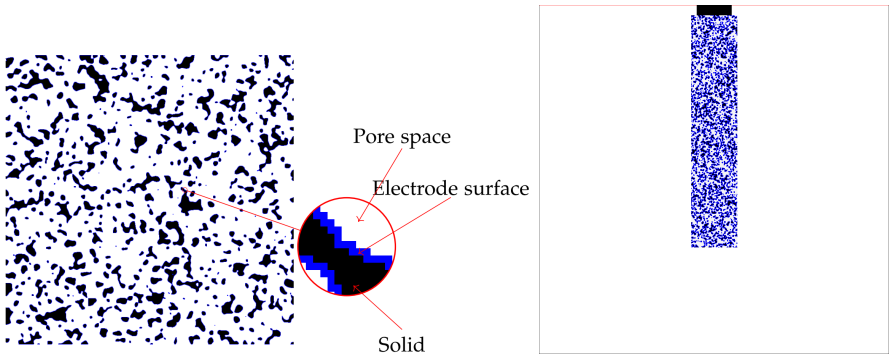
Enfin, on présente les résultats de l'application du modèle à des problèmes plus complexes avec des électrodes poreuses dans la section 4.3. Dans cette section, on introduit une méthode stochastique qui permet de générer des microstructures poreuses en deux dimensions avec une porosité spécifiée, une taille de pore variable ainsi qu'une orientation préférée qui affecte la tortuosité. Ces structures sont utilisées pour représenter les electrodes poreuses sur Lattice Boltzmann, et l'interface entre solide et fluide (figure 9) et considérée comme surface active où les réactions électrochimiques ont lieu selon Butler-Volmer.

En utilisant le domaine de simulation présenté dans la figure 9.b, on effectue une simulation de voltamétrie cyclique en utilisant le modèle. Ceci est accompli en imposant un potentiel à l'électrode et en enregistrant la réponse du système. La porosité



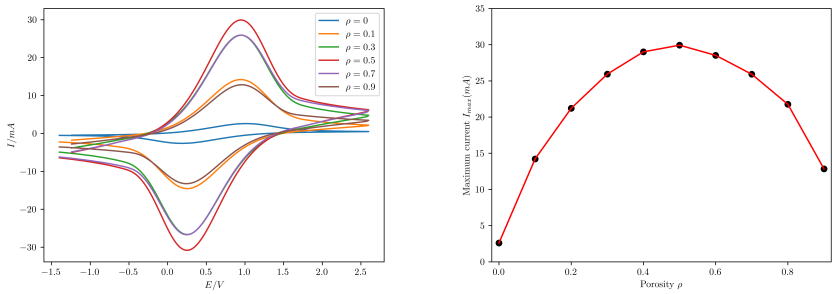
(a) (b)

Figure 8: Résultats de la simulation du problème de diffusion de Cottrell. (a) Variation du courant à travers l'interface en fonction du temps. (b) Erreur relative entre la solution analytique et la solution numérique.



(a) (b)

Figure 9: (a) Exemple d'une structure poreuse générée avec la méthode stochastique montrant l'interface active entre solide et fluide. (b) Domaine de simulation utilisé avec Lattice Boltzmann pour représenter une cellule électrochimique avec une électrode poreuse.



(a) (b)

Figure 10: (a) Courbes de voltamétrie cyclique effectuée avec le modèle Lattice Boltzmann pour différentes porosités. (b) Courant maximal atteint en fonction de la porosité.

est variée à chaque essai entre $\epsilon = 0$ (électrode plane) et $\epsilon = 0,9$ et le courant résultant est tracé sur la figure 10.a.

Si nous sélectionnons le courant de pic dans chaque voltammogramme et le représentons en fonction de la porosité, nous obtenons le graphique de la figure 10.b. On observe que le courant maximal augmente avec la porosité jusqu'à un certain point, puis commence à diminuer. Cela est dû au fait que la porosité est directement liée à la surface de l'électrode, et de même, le courant maximal est également lié à cette surface. En effet, à mesure que la porosité augmente, l'espace poreux devient plus prédominant et les régions solides sont réduites, ce qui diminue la surface active disponible pour la réaction de transfert d'électrons. C'est pourquoi le courant maximal commence à diminuer pour $\epsilon \geq 0,5$.

Il est donc possible d'utiliser ce modèle pour évaluer l'influence de la porosité sur le système, ainsi que de réaliser des méthodes électrochimiques tel que la voltammetrie cyclique dans la simulation. Cependant, il est important de noter que le modèle pourrait être amélioré davantage en incluant par exemple la migration, des simulations 3D ou encore des simulations à double électrodes qui permettraient de simuler le dispositif entier d'une batterie redox flow.

- Conclusion

Ce projet de thèse a conjugué les domaines des batteries redox flow, de la microfluidique et de la modélisation numérique en employant la méthode Lattice Boltzmann. Nous avons exploré deux axes principaux : la réalisation d'un prototype de batterie

redox flow microfluidique à base de liquides ioniques et comprenant des électrodes poreuses d'un côté, et le développement d'un modèle numérique pour la simulation et l'optimisation de ces dispositifs de l'autre, en particulier ceux utilisant des électrodes poreuses.

S'inspirant des avancées dans les piles à combustible avec des électrodes poreuses, notre travail a adapté ce concept pour une batterie non aqueuses. En remplaçant les cellules à électrodes planes par des électrodes en feutre de carbone, nous avons significativement amélioré la puissance et le courant de la cellule. Cette amélioration a été réalisée à des échelles relativement supérieures à celles typiques de la microfluidique, facilitant l'utilisation de techniques de fabrication économiques comme l'impression 3D.

Parallèlement, nous avons développé un modèle numérique versatile pour représenter les milieux poreux dans les dispositifs électrochimiques. Au lieu d'adopter une approche de continuum homogène avec des paramètres effectifs, notre modèle intègre l'écoulement des fluides, le transport de masse et les réactions électrochimiques dans des géométries complexes, grâce à la méthode Lattice Boltzmann. Cette méthode permet d'inclure la cinétique des réactions via l'équation de Butler-Volmer et de simuler la voltammétrie cyclique sur des géométries poreuses générées de manière stochastique. Ce modèle représente un outil intéressant pour la conception et l'optimisation de dispositifs utilisant des milieux poreux, ouvrant la voie à des études plus approfondies sur les performances des dispositifs dans des conditions opérationnelles variées.

GENERAL INTRODUCTION

In the context of environmental problems, developing clean and efficient energy storage technologies is one of the main topics of electrochemistry [1]. Fuel cells, rechargeable batteries and supercapacitors have therefore been a central focus of researchers in order to develop solutions to the energy crisis by enabling versatile and efficient use of electrical energy.

Research on conventional batteries has primarily focused on mobile and transport systems, where energy density and compactness are critical concerns [2]. In this regard, Lithium-ion batteries have proven to be a highly successful technology with applications ranging from portable electronics to electric vehicles [14]. However, there is a growing need for energy storage systems that are more optimized for stationary applications and grid storage, where energy density and size are far less critical parameters [3]. Consequently, Redox Flow Batteries have emerged as a promising technology for large-scale energy storage due to their flexibility and excellent cycling performance, with the all-vanadium redox flow battery being a notable commercial success [4]. Nonetheless, these systems still suffer from low energy density and cross-contamination through the membrane [5], which separates the anolyte and catholyte while being permeable to ions. Notably, the membrane is a relatively expensive component that accounts for a significant portion of the overall cost of these systems [10]. Another common drawback in the widely used water-based redox flow batteries is the electrochemical stability window of water, which limits the cell voltage that can be achieved [6]. A possible solution to this problem is to use organic or non-aqueous electrolytes, which have a wider electrochemical window and enable higher cell voltages but introduce other challenges, such as inflammation risks and evaporative losses [7]. In light of this, a number of researchers have considered using ionic liquids as electrolytes for these batteries. Due to the large number of possible anion-cation combinations, ionic liquids can be tailored for a wide range of properties and, in this case, can be tuned for non-volatility and a wide electrochemical window [8]. On the one hand, the use of optimal electrolytes has been a focus of research in this field. On the other hand, another axis of research has been to optimize

the batteries through their membrane [9], which is an expensive component that is often limiting in terms of performance [10]. Due to the highly demanding constraints that membranes are subjected to, a subset of literature has been dedicated to the investigation of membraneless designs for flow batteries [11]. These battery concepts rely on different approaches to prevent crossover within the cell such as phase separation or, more immediately relevant to this work, through the use of colaminar flows at microfluidic scales [12]. In this approach, the laminar nature of fluid flow at small scales is leveraged to minimize crossover and bypass the need for a membrane entirely. However, this approach also changes the potential applications of the battery, as the sizes required for microfluidic channels to maintain laminar flow are unsuitable for large-scale energy storage. Instead, these cells will be more suited for small, mobile and portable applications.

In this context, the objective of this project is to go a few steps beyond the original work of Chaabene et al. [13] with the investigation of using ionic liquids with microfluidic redox flow batteries, namely by integrating techniques from the field of flow cells such as the use of porous electrodes for increased surface area and reactant utilization.

Another facet of this project is to develop a numerical model for these devices in order to accompany the experimental work and guide the optimization of future battery designs.

Electrochemical energy storage relies on the transfer of electrons during ion charge or discharge processes in an electrode/-electrolyte interface [15]. Therefore, enhancing electrochemical storage device performance relies heavily on advancing our understanding of electrochemical phenomena. Consequently, the employment of numerical and computational techniques for the modeling and simulation of these phenomena is crucial. These models span a wide range of length scales, from the small and short time scales of individual electron transfer events to the broader kinetics of electrodes, encompassing the materials and structural characteristics of the device as a whole.

At the molecular scale, models are valuable to elucidate the fundamental mechanisms of electrochemical reactions and facilitate identifying novel catalysts. These models rely predominantly on molecular dynamics or Density Function Theory and include various trade-offs for an accurate simulation of the electrochemical interfaces. On the other hand, continuum models have proven immensely valuable for the overall design of elec-

trochemical devices such as batteries, fuel cells, capacitors and electrolyzers [16–18]. These large scale models are typically used to guide device design and characteristics, identify potential optimization strategies, address integration issues, and predict overall system behavior. One desirable performance metric for flow batteries and fuel cells is achieving high current densities. In this regard, porous electrodes have been demonstrated to be superior to planar electrodes due to their increased electroactive surface area [19]. Atomic-scale models are applicable for porous electrodes and can help elucidate the underlying mechanisms and the influence of pores on charge transfer and surface interactions. However, scaling them up to the device level for which the influence of structural properties and geometry is significant remains challenging due to their time complexity. Continuum models, on the other hand, often treat porous regions of the device as homogeneous media and employ bulk effective parameters without explicitly modeling the interfacial interactions between the pores [20]. Mesoscale methods serve as intermediaries between the macroscopic and microscopic scales. There are two primary types: mesh-based computational fluid dynamics (CFD) techniques, including finite volume and finite element methods, which are integrated with meshes detailed enough to model the microstructure of porous materials, and particle-based approaches, namely the Lattice Boltzmann Method (LBM), dissipative particle dynamics, and smoothed particle hydrodynamics, which simulate fluids through interactions of discrete particles [21–23].

In this work, we will develop a numerical model based on the Lattice Boltzmann Method that can be used to simulate the electrochemical reactions and transport phenomena in microfluidic redox flow batteries. The model uses the Lattice Boltzmann Method due to its handling of complex geometries and its ability to solve both the Navier-Stokes equations for fluid flow as well as the convection-diffusion equation for mass transport. The model will be used to investigate the influence of the porous electrode geometry on the performance of the battery and to guide the design of future devices.

This manuscript is split into four chapters, which are organized as follows:

The first chapter concerns the state of the art of conventional redox flow batteries, and developments concerning the use of ionic liquids as electrolytes for these cells. In this context, we introduce the concept of membraneless redox flow batteries,

specifically microfluidic devices, and the current state of research in this subfield. We conclude the chapter with a discussion of the various numerical methods and models that are used in literature to simulate electrochemical systems.

The second chapter is dedicated to the experimental work that was carried out in this project. We start by introducing the materials and methods that were used in the construction of the prototype microfluidic redox flow battery. We then present the results of the performance evaluation of the battery and finish the chapter with a discussion of the results and their comparison with previous work.

The third chapter delves into the theoretical underpinnings of the numerical Lattice-Boltzmann based model that was developed in this project. First, we introduce the Lattice Boltzmann Method with its kinetic origins and explain how it is used to recover the incompressible Navier-Stokes equations macroscopically. Then, we explore how it can be extended analogically to solve the convection-diffusion equation for mass transport. We then explain how electrochemical kinetics are incorporated into the model. We finish the chapter with an overview of how these different components are coupled and integrated into a single model.

The fourth and final chapter is devoted to unpacking the different results that were obtained from the numerical model. We start by validating the model against problems with known analytical solutions, and then introduce the scheme used to generate porous microstructures representing the electrodes in the simulation domain. Finally, we present the results of the simulations and discuss the influence of the porous electrode geometries on various aspects of the battery.

1

CHAPTER 1: LITERATURE REVIEW

In this inaugural chapter of this manuscript, a broad overview of the existing literature surrounding the field of microfluidic redox flow batteries is conducted. In the first part, an overview of conventional redox flow batteries is presented, along with their significance in the broader context of electrochemical energy storage technologies and their main commercial success in stationary applications. Then we examine literature concerning the use of ionic liquids for energy systems due to their potential benefits and advantages as electrolytes over the more common water-based redox flow batteries. Then we turn our attention to the state-of-the-art of microfluidic redox flow batteries, which is an emergent field that presents a derivative concept of a redox flow battery aiming to miniaturize the device through the incorporation of techniques from the field of microfluidics and MicroElectroMechanical Systems (MEMS) or Lab-on-Chip devices. Integrating microfluidics techniques into the field of redox flow batteries opens up a new range of potential applications that are distinct from the conventional stationary applications, into more portable and mobile applications.

Analyzing these different fields, namely conventional redox flow batteries, ionic liquid electrolytes, and finally, microfluidics, will allow us to identify the main parameters that we can manipulate in order to optimize the performance of our prototype redox flow battery.

The final section of this chapter is dedicated to numerical modeling approaches that apply to electrochemical devices, with a particular focus on the Membraneless Microfluidic Redox Flow Battery (MMRFB). Within this framework, where flow and mass transport phenomena in complex geometries are involved, the applicability and utility of the Lattice Boltzmann Method are discussed for the simulation and modeling of such electrochemical devices.

1.1 REDOX FLOW BATTERIES

Amidst the ongoing energy problem and the global transition towards environmentally friendly energy sources, there remains a critical need for energy storage solutions which are a significant bottleneck facing the widespread adoption of renewables [24]. In this context, Redox Flow Batteries stand out as a promising candidate energy storage system, particularly for intermittent sources such as wind or solar power [2].

The operational mechanism of Redox Flow Batteries is predicated on storing electrical energy using two redox species dissolved in electrolyte tanks. The two solutions are then pumped through a reactor where charge transfer reactions occur on the surface of the electrodes, while keeping the two components separate using an Ion Exchange Membrane. Figure 1.1 shows the main components of a Redox Flow Battery (RFB). The main distinctive characteristic of RFBs that presents an advantage in many applications is the decoupling of power output and energy. On the one hand, power output depends on the size of the stack, where larger electrode surface area allows higher current densities. On the other hand, the energy storage capacity depends on the size of the electrolyte storage tanks and the concentration of the redox-active species.

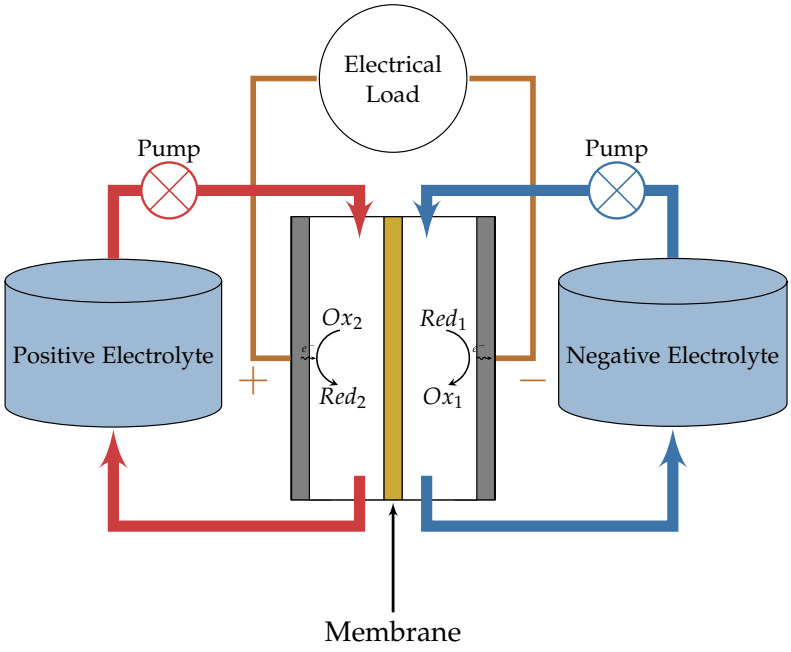


Figure 1.1: Diagram of a conventional Redox Flow Battery.

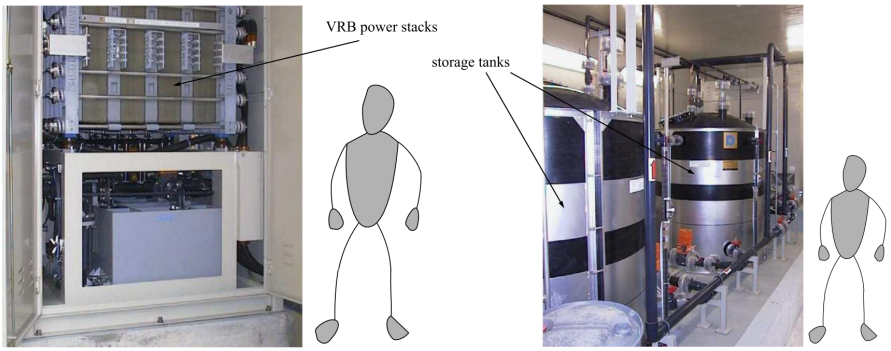


Figure 1.2: A typical 90KWh-rated vanadium redox flow battery system (Sumimoto Electric Industries) [2]. The cell stacks on the left are composed of 100 cells in series, while the electrolyte storage tanks store 4000 L of electrolyte.

This decoupling of power and energy has a number of advantages: increasing the capacity of a cell can be achieved by simply replacing electrolyte tanks with larger ones, without modifying the reactor. Similarly, it can be refilled quickly by replacing the electrolyte. On the other hand, the power output of the device depends on the size of the electrodes, and consequently, the size of the reactor. Therefore, increasing power output can be achieved by using larger electrodes that comprise a larger active surface area. This flexibility of RFBs is a key point in the potential commercial success of this technology, as it can be adapted for specific applications regardless of which battery chemistry is considered optimal [2, 25, 26].

Due to the relatively lower energy density of redox flow batteries compared to the more common rechargeable intercalation systems such as Lithium ion batteries [27], the main commercial success of redox flow batteries was a result of the flexibility offered by the energy-power decoupling. This has led to their adoption in large-scale energy storage applications such as peak-shaving and load balancing at the grid level, where energy density and the total weight of the system are not decisive parameters. Rather, in these applications, the main criteria are long cycling life, scalability and flexible design [3, 27]. In fact, it is currently the application where this technology has been most widely used. An example of a large-scale system is shown in figure 1.2, which is a vanadium based system rated at 90KWh.

Notably, the increased adoption of intermittent renewable energy sources such as wind and solar power has led to an increased need for storage systems that can mitigate this intermittency. Meanwhile, energy demand itself is not constant and

the grid infrastructure that comprises end users with little to no storage capacity is required to conform to these fluctuations. This is where redox flow batteries have emerged as one of the best technologies to address this issue in the last few decades [28]. To further understand this technology, the next section explores the chemistries of the most notable redox flow systems in literature.

1.1.1 *Redox Flow Battery Chemistry*

The electrochemical reactions describing the two half-cells of a Redox Flow Battery are shown in equations as follows:

$$Ox_1 + ne^- \rightleftharpoons Red_1 \qquad Red_2 \rightleftharpoons Ox_2 + ne^- \tag{1.1}$$

The overall reaction is given by:

$$Ox_1 + Red_2 \rightleftharpoons Ox_2 + Red_1 \tag{1.2}$$

This reaction is reversible and is what drives the current through the external circuit. Historically, RFB development is considered to have began in the 1970s in NASA, with an Iron/Chromium system for a photovoltaic array application [29, 30] which used an Ion Exchange Membrane (IEM) and low cost carbon felt electrodes. It was based on the following reactions:

$$Cr^{3+} + e^- \rightleftharpoons Cr^{2+}, \quad E^0 = -0.41 \text{ V vs. RHE} \tag{1.3}$$
$$Fe^{2+} \rightleftharpoons Fe^{3+} + e^-, \quad E^0 = 0.77 \text{ V vs. RHE} \tag{1.4}$$

The system had an open circuit voltage between 0.9 and 1.2 V but suffered from slow kinetics on the Chromium side [2] as well as low coulombic efficiency due to the hydrogen evolution side reaction that is caused by the low redox potential of the Cr^{3+}/Cr^{2+} couple.

Since this initial system, a considerable number of different RFBs have been developed since then, with different operating conditions, electrolytes and electrode materials. Some notable systems are presented in Table 1.1. However, the system that has arguably garnered the most attention and research is the all-Vanadium RFB. First proposed by Skyllas-Kazacos et al. in the 1980s [31], the system uses Vanadium in four different oxidation states (V(III), V(II), V(IV) & V(V)) in the following reaction:

Table 1.1: Comparison of different water-based RFB systems

System	Electrolyte	⊖ pole	⊕ pole	$E(V)$	Ref.
Fe/Cr	Hydrochloric Acid	$Cr^{2+} \rightleftharpoons Cr^{3+} + e^-$	$Fe^{3+} + 2e^- \rightleftharpoons Fe^{2+}$	$0.9 \sim 1.2$	[30]
Br/S	NaBr & NaS_2	$2S_2{}^{2-} \rightleftharpoons S_4{}^{2-} + 2e^-$	$Br_3{}^- + 2e^- \rightleftharpoons 3Br^-$	$1.7 \sim 2.1$	[35]
Zn/Br	$ZnBr_2$	$Zn \rightleftharpoons Zn^{2+} + 2e^-$	$Br_2 + 2e^- \rightleftharpoons 2Br$	1.85	[36]
V(V)/ V(II)	H_2SO_4	$V^{2+} \rightleftharpoons V^{3+} + e^-$	$VO_2{}^+ + 2H^+ + e^- \rightleftharpoons VO^{2+} + H_2O$	1.26	[31]

$$V^{3+} + e^- \rightleftharpoons V^{2+}, \quad E^0 = -0.26 \text{ V vs. RHE} \tag{1.5}$$

$$VO^{2+} + H_2O \rightleftharpoons VO_2{}^+ + 2H^+ + e^-, \quad E^0 = 1.0 \text{ V vs. RHE} \tag{1.6}$$

This presents a significant advantage (which facilitated the successful commercialization the Vanadium RFB [32]) due to the mitigation of contamination issues from cross over that cause continuous degradation on top of the efficiency loss [2]. It should be noted however that capacity and efficiency fade due to crossover are still an issue because of the varying rates of diffusion of the four Vanadium species across Nafion membranes. However, it was shown by Li et al. (2014) that the problem of capacity decay can be mitigated through careful management of the pressure in the electrolyte tanks [33]. Furthermore, according to Luo et al. (2013) [34], periodic total remixing of the electrolytes can also mitigate this issue and enable long-term cycling of the system. The Vanadium RFB is therefore considered today to be the most mature and commercially viable RFB technology [2].

While RFBs can be classified according to the redox active species used, one can also categorize them in terms of the nature of the electrolyte. In fact, depending on whether water is used as a solvent, RFBs can be classified as aqueous or non-aqueous cells. Most of the systems reported in literature, and indeed all of those presented in Table 1.1 use water-based electrolytes. The maximum potential of these systems is therefore limited by the electrochemical window of water electrolysis which limits their energy density. This window varies with pH and the nature of the electrode, but is typically 1.23 V at 25°C [37]. Non-aqueous systems are therefore promising because they provide a wider electrochemical window and offer more flexibility in terms of possible redox couples. Non-aqueous RFBs have therefore garnered increasing research interest in recent years despite some ongoing issues

such as low solubility of reactants and low cycling efficiency [26]. For instance, Li et al. [38] introduced the all-organic RFB which uses a 2,2,6,6-tetramethyl-1-piperidinyloxy (TEMPO) and N-Methylphthalimide system. Both reactant species were dissolved at a concentration of $0.1M$ in $1M$ $NaClO_4$/Acetonitrile as the electrolyte. The authors mainly used cyclic voltammetry to evaluate the diffusion coefficient of the redox species in the electrolyte, and were also able to show stable charge-discharge cycling for the first 20 cycles with a Coulombic Efficiency of 90% at a current density of 0.35 mA/cm. Similarly, Brushett et al. [39] showcased a device using 2,5-Di-tert-butyl-1,4-bis(2-methoxyethoxy)benzene (DBBB) along with an array of quinoxaline derived molecules as the active species. The concentration of both reactants was 0.05 M in an electrolyte consisting of 0.2 M $LiBF_4$ in propylene carbonate. The battery achieves a peak of 79% Coulombic Efficiency (CE) at a charge/discharge current of 0.0625 mA/cm^2. More recently, Yichao et al. [40] showed a proof of concept study of an all-organic cell using thioether-substituted cyclopropenium derivatives as the electrolytes, and were able to achieve a cell voltage of 3.2 V when paired with a phthalimide derivative. Despite the very high cell potential displayed by this particular device, the concentrations used were 0.05 M for the methylthioether-substituted cyclopropenium and 0.1 M for the N-alkylphthalimide that was paired with it. This system was based on a supporting electrolyte of 0.5 M $TBAPF_6$ in acetonitrile. In operating conditions, the battery was able to achieve a Coulombic efficiency of 83% in the first 15 cycles. However, after about 17 cycles, there was the onset of a rapid capacity loss due to anolyte decay. It is worthy of note that carbon felt electrodes at a thickness of 400 μm were combined with graphite places for current collection in this device. Likewise, an all-iron non-aqueous RFB was reported in 2020 by Zhen et al. [41], using iron acetylacetonate (Fe(acac)3) anolyte and N-(ferrocenylmethyl)-N,N-dimethyl -N-ethylammonium bis(trifluoromethane-sulfonyl)imide (Fc1N112-TFSI) catholyte with a cell voltage of 1.32 V. In this case, acetonitrile was used as the solvent and [TEA][TFSI] as the supporting electrolyte with a concentration of 0.1 M for the active iron species in 0.5 M TEATFSI/MeCN. This particular device displayed high charge/discharge performance with a coulombic efficiency of 98.7% over 100 cycles at a current density of 10 mA/cm^2. Despite the very high cycling performance, electrolyte crossover

and evaporation still contribute significantly to the long-term capacity fade of this battery.

It is therefore apparent that low solubility remains an challenge for non-aqueous redox flow batteries. Furthermore, The use of organic solvents introduces evaporative losses and inflammation risks [7]. Ideally, electrolytes that are non-volatile while still preserving the advantage of a wide electrochemical window, as well as having the flexibility to potentially enable higher reactant solubilities are required. Ionic liquids are a relatively new and promising candidate for this purpose, and are the focus of the next section.

1.1.2 *Redox Flow Batteries with Ionic Liquids*

Ionic liquids stand as a distinct category of solvents with unique physico-chemical properties [42, 43]. They are essentially salts that exist in liquid state at room temperature (they are also commonly referred to as Room Temperature Ionic Liquids), or have a melting point below 100°C. Unlike more conventional salts in which the electrostatic forces between the ions are strong enough to maintain a solid state at well above room temperature, ionic liquids typically comprised of large, unorganized ions with low symmetry which weakens these forces enough to allow a liquid state at room temperature.

Generally, they are composed of an organic cation such as imidazolium, pyrrolidinium, or quaternary ammonium ions, and either a small inorganic anion such as chloride or bromide, or a complex anion such as bis(trifluoromethylsulfonyl)imide (TFSI) (Figure 1.3). This wide variety of possible cations and anions allows for a large number of possible combinations, each with its own unique properties (Estimated number of possibilities is between 10^{12} to 10^{18} [44, 45]). This makes ionic liquids highly tuneable and versatile, and therefore suitable for a wide range of applications.

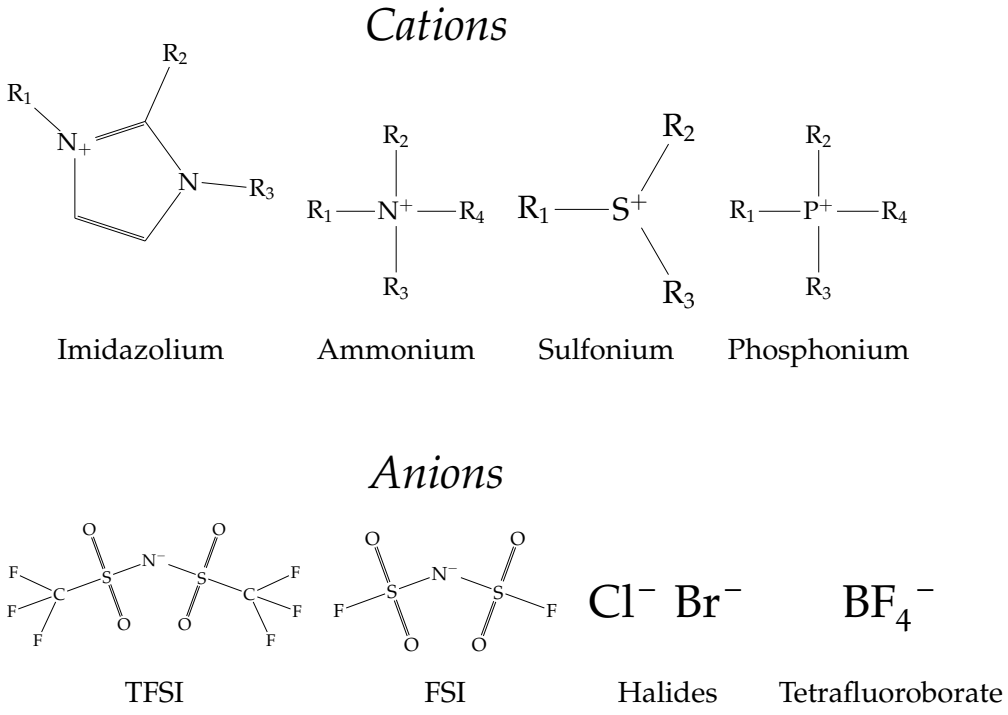


Figure 1.3: Examples of common ionic liquid cations and anions.

Historically, Paul Walden was the first to report the properties of ethylammonium nitrate, the first ionic liquid to be discovered in 1914 [46–48]. It is not, however, until the 1970s that active research on ionic liquids picked up in the United States [49].

Due to their tunability, they are subject of investigation and research especially in the energy sector [8]. For electrochemical applications, and specifically for Redox Flow Batteries, properties such as low volatility and wide electrochemical window make them a very attractive candidate as new electrolytes.

Among early attempts to evaluate Ionic Liquids as supporting electrolytes for Redox Flow Batteries is that of Zhang et al. (2012) [50] which used tetrabutylammonium hexafluorophosphate ($TBAPF_6$) and 1-ethyl-3-methylimidazolium hexafluorophosphate ($EmimPF_6$) with acetonitrile as the supporting electrolyte for vanadium acetylacetonate. The ionic liquids were tested with various concentrations in acetonitrile between 0.05 M and 0.5 M and were shown to be stable between -2.5 V and 1.5 V. The stability window of this system is limited by the ionic liquids because acetonitrile is known to have an electrochemical window of around 5 V [51]. The authors of this work focused on the diffusion coefficients of the active species at different concentrations of the ionic liquids. Nonetheless, the Ionic Liquid (IL)s enabled cell potentials of around 2.3 V, which would not be

possible in aqueous systems. In this work, the active vanadium species was dissolved at 0.01 M in both supporting electrolytes.

Similarly, tetraethylammonium bis(trifluoromethylsulfonyl)imide ([TEA][TFSI]) was also used in all-organic RFBs in multiple works. For instance, in the system of Wei et al. [52], despite the fact that the solubility limit of the used redox species (FL: FL: 9-Fluorenone and DBMMB: 2,5-di-tert-butyl-1-methoxy-4-[2'-methoxyethoxy]benzene) is at 0.9 M in 1.2 M [TEA][TFSI]/acetonitrile, the authors used a concentration of 0.1M for both species. The cell voltage achieved was about 2.3 V but performance was hindered by the decay of the FL and DBMMB radicals through parasitic reactions, which results in capacity fade not only with number of cycles but also with storage time. Similarly, Wang et al. (2014) [53] used the same [TEA][TFSI] ionic liquid with a redox couple of Benzophenone and 1,4-di-tert-butyl-2,5-dimethoxybenzene (DBB) to achieve an even higher cell voltage of 2.95 V. The concentrations of the active species were relatively low at 0.01 M for both species in 0.1 M [TEA][TFSI]/acetonitrile. The coulombic efficiency was measure at around 97% during 50 charge-discharge cycles.

Interestingly, ILs have also been used in aqueous RFBs by Chen et al. (2016) [54], who used an ionic liquid which is soluble in water, namely 1-butyl-3-methylimidazolium chloride ([Bmim]Cl). In this particular case, the IL was used to shift the potential of side reactions (hydrogen and oxygen evolution), improving the round-trip efficiency of the aqueous system. Increasing the IL concentration was found to affect the width of the stability window due to the suppression of water electrolysis reactions. These electrolytes were tested at a molality of 10 m (BmimCl accounts for 60% of the total volume of the aqueous electrolyte at 10 m) which results in a electrochemical stability window of 3 V and using Zn/Ce active species at concentrations of 0.04 M for a cell voltage of 2.1 V.

In conjunction with these works, several preliminary investigations into the kinetics and electrochemical performance of different redox species in ILs were conducted. It was shown for instance that Vanadium acetylacetonate is quasi-reversible in [Emim][TFSI] [55]. In this system, the concentration used for the active vanadium species was about 0.01 M, which is lower than the 1 M that can be achieved with acetonitrile as a solvent at room temperature [56]. On the other hand, redox kinetics have been reported to be low due to solubility issues in [57], which tested [Bmim][TFSI] in the presence of an iron com-

plex with the same anion ([Fe(bpy)][TFSI]), despite the reported quasi-reversibility of Fe(II)/Fe(III) in this medium.

Another interesting battery concept is that of Shaltin et al. (2016) [58], which uses a copper based ionic liquid that functions as solvent and electrolyte simultaneously. The IL in question is the [Cu(MeCN)4][TFSI], and allows for very high copper ion concentrations that enable impressive energy densities of 75 WhL^{-1} for a RFB.

In light of the results obtained of these previous works, it is clear that ionic liquids are a promising candidate electrolyte for redox flow batteries. The ensemble of the works presented above show that they indeed enable higher cell voltages than aqueous systems. Nevertheless, apart from very few exceptions, the concentrations of the active species that can be achieved remain low compared to aqueous electrolytes, which limits the energy density of the device.

With these considerations in mind, in all these systems, Ion exchange membranes (IEMs) remain a limiting component of redox flow batteries in terms of cost, durability and efficiency [2]. An IEM is used to prevent crossover, which is the migration of the redox active species to the other half cell across the membrane, leading to self-discharge, irreversible reactions or low coulombic efficiency [59]. In fact, a significant number of works have addressed membrane-related challenges as improvements in this area can have a significant impact on the performance of the device [11, 60]. However, it remains an expensive component of the redox flow battery and represents between 20 and 40 % of the total cost of a RFB cell [6, 10]. In addition, ionic conductivity remains a limiting factor in most membranes [61].The widely used Nafion membrane for aqueous systems, for instance, suffers from crossover despite having decent ionic conductivity [2]. While it is true that reactant crossover can be reduced by increasing membrane thickness, this comes at the cost of an increased ohmic resistance of the cell [62].

In this regard, researchers have investigated the possibility of alternative solutions for the mitigation of reactant crossover. One possible solution that was explored in literature is to rely on the biphasic separation of the anolyte and catholyte, which are selected for their mutual immiscibility. A schematic of such a device is shown in figure 1.4. In fact, the first work to examine such a cell is that of Gong et al. (2016) [63], which comprised a Zn^{2+}/Zn couple in the aqueous phase and a Fc^{+}/Fc couple in the organic phase that consists of butyl acetate and an ionic

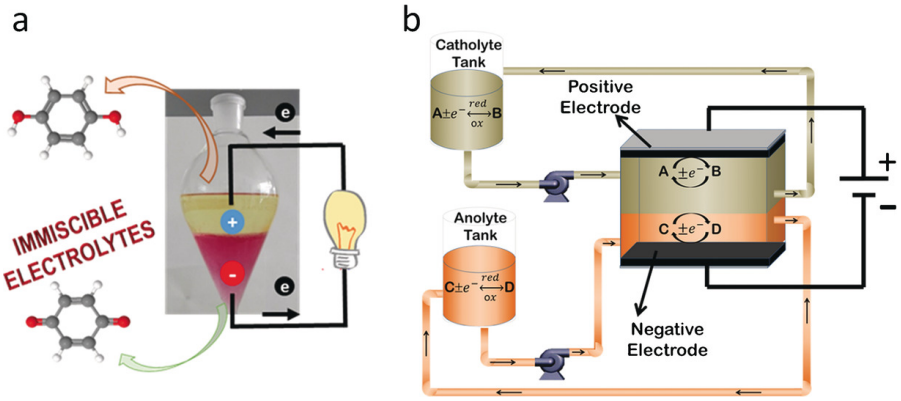


Figure 1.4: Schematic of a macroscale membraneless biphasic redox flow battery. (a) Schematic representation showing an upper aqueous phase and a lower ionic liquid phase (b) Schematic of a redox flow battery with a horizontal design based on immiscible electrolytes (From [65]).

liquid composed of a mixture of tricaprylmethylammonium chloride and trioctylmethylammonium chloride [64]. This introduced a proof-of-concept of a conventional macroscale redox flow battery without a membrane altogether.

Similarly, Navolpotro et al. [65] showcased a device that used para-benzoquinone (pBQ) in the ionic liquid 1-butyl-1-methylpyrrolidinium bis(trifluoromethylsulfonyl)imide ([PYR_{14}][TFSI]) which is hydrophobic, alongside an acidic solution of hydroquinone. The battery is able to operate in a reversible manner at a cell voltage of 1.4 V with a coulombic efficiency of 100%. The performance of the system was evaluated at an active species concentration of 0.1 M for an energy density of 2.4 WhL^{-1}, but authors claim that the solubility of the reactant species in each electrolyte can be increased up to 0.7 M for a high theoretical energy density of 22.5 WhL^{-1}. Another instance of devices leveraging biphasic separation is the device of Bamgbopa et al. (2018) [11]. In this device, the aqueous anolyte consists of Iron(II) Sulfate ($FeSO_4$) as the active species, with the added property of being immiscible in the ethyl-acetate based catholyte, which is supported by the ionic liquid [$Pyrr_{14}$][TFSI] and contains iron(III) acetylacetonate ($Fe(acac)_3$). The cell was tested with active species concentrations of 0.1 M and showed high capacity retention with only 1.6% fade per cycle over 25 cycles. However, these biphasic separation systems remain a niche subset of the literature surrounding RFBs, and are still in their infancy. On the other hand, a much more popular approach to the elimina-

tion of the membrane is the miniaturization of the device using microfluidics.

However, before listing some of the systems that have tried to eliminate the membrane using this strategy, the next section will introduce the field of microfluidics and its applications.

1.2 MICROFLUIDICS

1.2.1 *Microfluidics as an extension of MEMS*

In the 1970s, advancements in science and technology have enabled the miniaturization of electromechanical systems down to micrometric scales, leading to the development of the field of MEMS. In the following decades, and particularly since the 1990s, the field of MEMS has expanded to encompass an array of devices that manipulate and process small quantities of fluids. In practice, microfluidic scales range from 10^{-9} to 10^{-18} L, or 100 *nm* to 1 *mm* [66, 67]. While microfluidics includes different subdomains depending on the specific scale (Figure 1.5) (A common subdivision of the field is into nanofluidics, microfluidics and millifluidics), throughout this manuscript we will use microfluidics as an all-encompassing term, as well as for millifluidics, which becomes the focus later on. A contributing factor to the widespread adoption of microfluidics techniques in many different scientific fields is the introduction of the low-cost and convenient soft lithography micro-fabrication protocol, especially for Poly(dimethylsiloxane) (PDMS) [68].

The mechanics of fluids at these scales depends on the Knudsen number which is defined as the ratio of the mean free path over the characteristic length scale of the system:

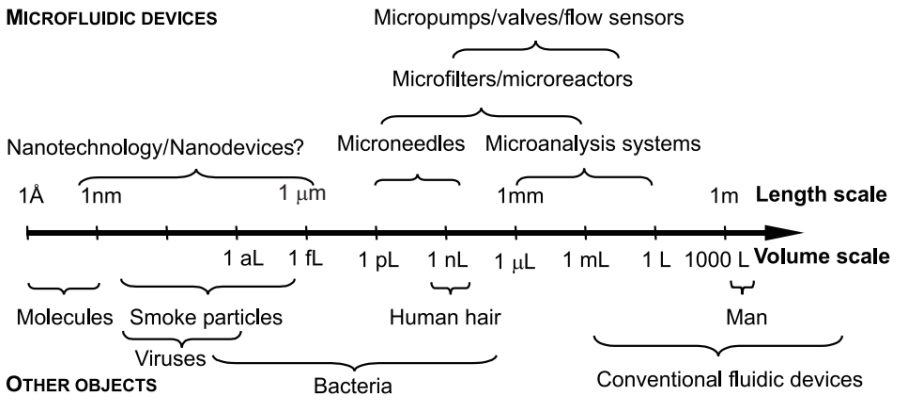


Figure 1.5: Different scales of fluid dynamics, from [69]

$$Kn = \frac{\lambda}{L} \tag{1.7}$$

Where:

- λ is the mean free path of the fluid particles in m.
- L is the characteristic length scale in m.

The Knudsen number defines three regimes:

- $Kn < 10^{-3}$: the continuum approximation holds and the Navier-Stokes equations apply.
- $10^{-3} < Kn < 10^{-1}$: transitional flow regime. Commonly divided into two sub-regimes: Slip flow and Transition flow.
- $10 < Kn$: the free molecular regime. The Navier-Stokes equations do not apply, and statistical mechanics must be used.

However, even in very small devices, and especially for liquids, the Knudsen number is very low and the continuum approximation still holds. In later chapters, our device will be on the millimetric scale so the Navier-Stokes equations are used. As an example, given a microfluidic channel with a characteristic length scale of 100 μm and a mean free path of 0.3 nm (commonly used as the mean free path for water [70]), the Knudsen number is Kn = 3×10^{-6}, which is well within the continuum regime.

At these small scales, gravitational and inertial forces become increasingly negligible compared to capillary forces. A critical parameter in establishing the nature of these flows is the Reynolds Number:

$$Re = \frac{\rho u L}{\mu} = \frac{u L}{\nu} \tag{1.8}$$

where:

- ρ is the fluid density in kg/m^3
- u is the characteristic fluid velocity in m/s
- L is the characteristic length scale in m

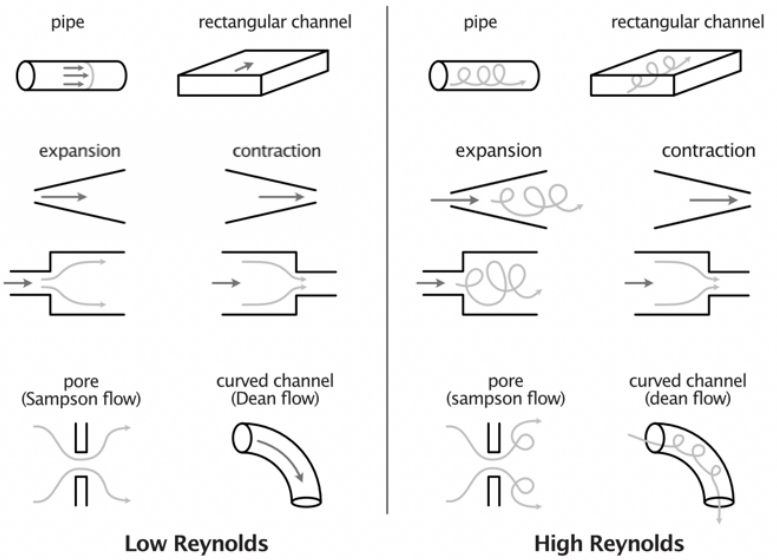


Figure 1.6: Different flow types for low and high Reynolds numbers. (From [67])

- μ is the dynamic viscosity in $Pa.s$
- ν is the kinematic viscosity in m^2/s

One can interpret this as the ratio of Inertial forces to viscous forces, and can be arrived at by adimensionalizing the Navier-Stokes equations or using the π-theorem. Given that flow velocities in microfluidics do not exceed 1 cm/s, and that characteristic length scales are on the order of 1 mm, the Reynolds number of a microfluidic device is typically low, but exceptions with higher Re do exist however, see [67]. Different flow configurations for low and high Reynolds numbers are shown in Figure 1.6.

It is generally considered that flows with a Reynolds number above the range of $2300 - 4000$ are turbulent due to the domination of inertial forces. This wide range of possible critical values of the Reynolds number is due to the fact that the transition from laminar to turbulent flow depends on the geometry of the system, which dictates the onset of shear forces that lead to instabilities in the flow, and the transfer of energy of larger scale bulk laminar flow, into smaller scale eddies. In microfluidics, if we take an upper bound of characteristic length scales of about $5mm$ and a velocity of $1cm/s$, the reynolds number for the flow of water, which has a kinematic viscosity of around $1 \times 10^{-6} m^2/s$, in these conditions works out to be $Re = 50$, which is considered to be in the laminar regime. It should be

noted however that typical microfluidic devices have characteristic length scales on the order of $100\mu m$ and velocities on the order of $1mm/s$, which results in a Reynolds number of $Re < 1$, which is well within the laminar regime.

In these conditions, the Navier-Stokes equations can be simplified to the Stokes equations due to the dominance of viscous forces over the inertial forces:

$$\mu\nabla^2\mathbf{u} + f = \nabla p \tag{1.9}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{1.10}$$

where:

- f is the body force per unit volume in N/m^3
- p is the pressure in Pa

The working principle behind microfluidic devices is leveraging flow channels to manipulate and control fluids at small scales. In the case of redox flow batteries, the key idea is to use the laminar nature of the flow inside microchannels to avoid mixing reactants and minimize crossover. This is due to the fact that streamlines in laminar flow are parallel and do not induce convective mixing currents. For example, considering a parallel and laminar flow inside a microchannel transporting a given species along the horizontal direction, any mass transport that occurs in the vertical direction is due purely to diffusion due to concentration gradients. This is not the case in turbulent flow, where the vertical component of the velocity field at a given point is not necessarily zero, and is often fluctuating with time.

1.2.2 *The advent of microfluidic fuel cells and batteries*

The simplest concept of a microfluidic redox flow battery is a device that consists of two microchannels through which the electrolytes are introduce, and subsequently converge into a main central channel that plays the role of the reactor as shown in figure 1.7. In this case, the interdiffusion of the species depends on the channel length. The central channel contains the electrodes, and if the flow is laminar, the two components are separated by a diffusion layer without requiring a membrane:

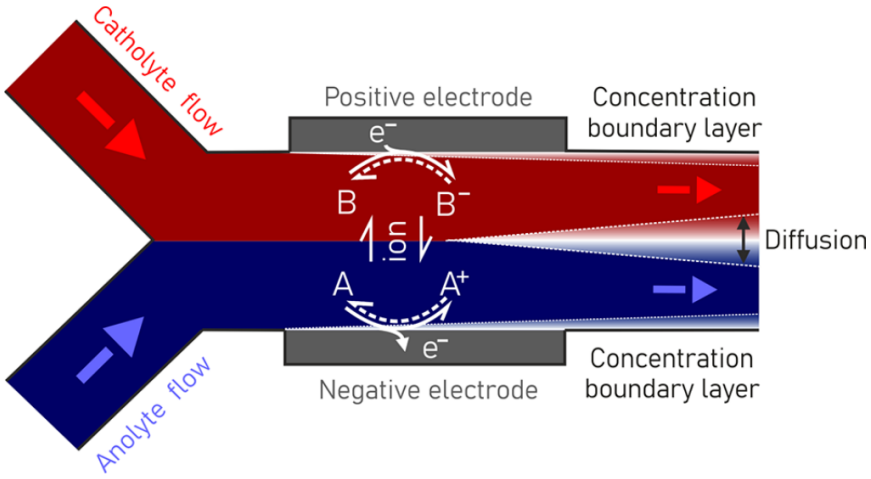


Figure 1.7: Schematic of a Y shaped membraneless microfluidic redox flow battery. (from [12])

In their seminal work of 2002, Ferrigno et al. [71] presented the first example of a microfluidic fuel cell. The device used the advantages of microfluidics to mitigate some of the issues that conventional devices were facing. The nature of the flow in a micro-channel allowed for better control of the reactants thanks to the parallel streamlines of laminar flow, and the absence of turbulent convective mixing currents. Furthermore, the membrane of a conventional device constituted a significan portion of its overall cost [60], and contributed to the ohmic losses of the system.

Ferrigno's device used planar graphite electrodes and comprised a main 2 mm wide central channel at a height between 50 to 200 μm with two inlets leading the two components of V(II) and V(V) at a concentration of 1 M in H_2SO_4 into it (Figure 1.8). The electrodes were situated at the bottom of the channel and had a width of 0.5 mm and length of 17 mm with a 1 mm gap between them.

The Ferrigno et al. cell was constructed using PDMS with the soft lithography protocol. In fact, much of later research on microfluidic redox flow batteries continued with PDMS for ease of manufacturing and its compatibility with most electrolytes [72].

Similar Y shaped designs using various electrolytes and reactants were showcased in the following years, and most were fuel cell designs [73–75]. Despite the fact the most research in this field focused on single pass devices operating in discharge mode. Most devices feature a single output slot, which leads to the confluence of the reactants and precludes the possibility

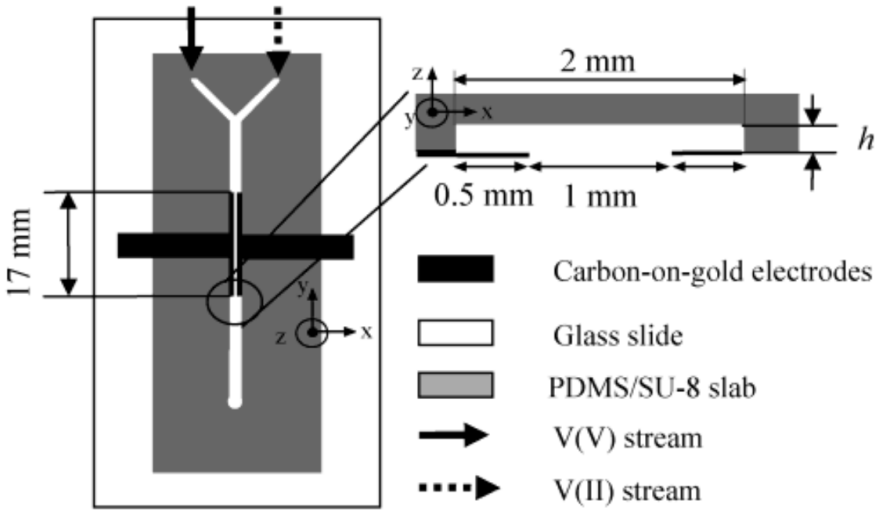


Figure 1.8: Schematic of the membraneless microfluidic fuel cell by Ferrigno et al. (2002) [71]

of recirculation. Notably, one of the first devices to showcase a microfluidic redox flow battery design with an investigation into its performance in roundtrip charging and recharging cycles is Braff et al. hydrogen bromine cell [76]. Inspired by air-breathing fuel cells, the device uses a micro-porous hydrogen anode and an aqueous hydrogen bromide solution as a flow separating electrolyte. The device was able to achieve a roundtrip voltage efficiency of 90% which is comparable to conventional vanadium RFBs.

A more immediately relevant approach to this thesis project is the work of Chaabene et al. [13] which used ionic liquids as the electrolyte and solvent in a MMRFB with planar electrodes and is shown in figure 1.9. In this work, the construction of the device was also done using PDMS and soft lithography, and was a proof of concept for a membraneless device that uses ionic liquids in a device that is geometrically similar to the previously cited work of Ferrigno et al. The active species that were used were mainly Iron-Quinone in $Emim[TFSI]$ at a concentration of 0.1 M and 0.35 M respectively, as well as Tempo-Alizarine at a concentration of 2.5 M and 0.1 M respectively. However, the device suffered from low-reactant utilization due to thinness of the diffusion layer, and relatively low power densities of around 40 μWcm^{-2} for the Fe/Q system and 140 μWcm^{-2} for the Tempo/Alizarine system, both at flow rate of 20 $\mu l \text{min}^{-1}$. This is attributed to the limited surface area of the planar electrodes as

well as the suboptimal positioning of the electrodes which are not located directly opposite one another.

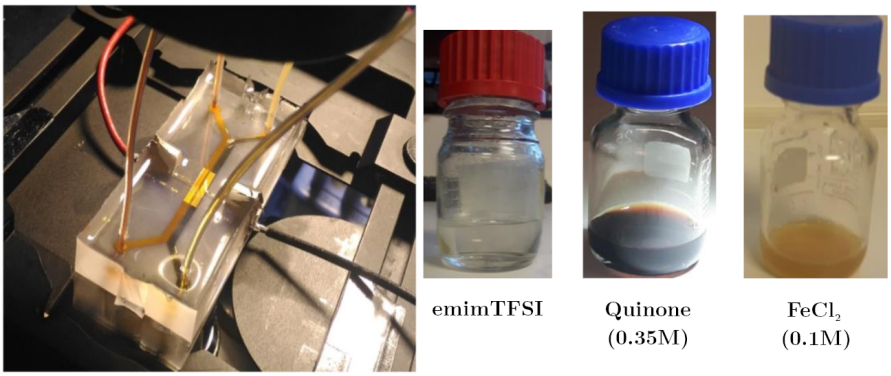


Figure 1.9: Membraneless microfluidic redox flow battery using ionic liquids by Chaabene et al. (2023) [13]

A particularly interesting advancement in the adjacent field of fuel cells was the cell architecture introduced by Kjeang et al. [77]. They demonstrated a cell with flow through porous electrodes in an aqueous Vanadium redox system, and achieved high energy conversion efficiency and fuel utilization, with a power density of 131 $mW\ cm^{-2}$. In fact the flow cell that displayed high power output in literature do use the flow through porous electrode design [78]. A particularly high power output cell was reported by Goulet et al. (2017) [79] achieves a power density of 2.01 $W\ cm^{-2}$. They were able to achieve this high value by decreasing the distance between the porous electrodes (down to 1 mm) as well as by using a flow deposition method in which carbon nanotubes are introduced along with the electrolyte during cell operation, which enhances the active surface area of the electrodes. However, it should be noted that the power density values that are reported in literature for porous electrodes are not comparable to those of planar electrodes. A few common metrics that are used in an attempt to standardize and compare different cell designs are the concept of the *electrochemical chamber*, which divides the power output by the volume of the reactor portion of the cell, including the electrodes, as well as the *cross sectional area* of the flow channels, in which the area considered is that of the cross section of the porous electrodes that is normal to the reactant flow direction. The power densities measured in this manner do take into account the size of the system, but comparison with planar electrode designs remains relatively ambiguous [80].

Furthermore, in contrast to fuel cells, redox flow batteries have the additional constraint of operation in both charge and discharge mode, though refilling the electrolyte tanks between operation cycles is possible. In fact, a fuel cell can be considered as a battery operating purely in discharge mode, while fuel regeneration is performed off-chip [80]. Moreover, scaling up microfluidic devices has remained a challenging issue in the field. Due to the fact that most devices rely on the laminar nature of the flow in their micro-channels to avoid crossover, and most systems have a voltage output on the order of $1V$, most approaches have focused on different stacking strategies [81–84].

In conclusion, ionic liquids have been demonstrated as an effective electrolyte for microfluidic redox flow batteries, while literature from fuel cells has inspired the use of porous electrodes in microfluidic batteries in order to increase the active surface area, and consequently, the power output of the device. A first axis of investigation aims to combine these two ideas in order to evaluate the use of porous electrodes with ionic liquids in a membraneless microfluidic redox flow battery. The second facet of this work that will be elaborated in the next section concerns accompanying this experimental work with a computational model that can be used to guide the development and optimization of these systems at the device level.

1.3 THE LATTICE BOLTZMANN METHOD FOR FLUID DYNAMICS AND MASS TRANSPORT

Models at the small scale of molecules, electron transfer events and electronic structure are useful to understand the underlying mechanisms of electrochemical reactions and discover new catalysts. They have mostly relied on Molecular Dynamics (MD) methods and Density Functional Theory (DFT). The review of Schwarz et al. discusses various approaches and trade-offs that different models use in order to accurately describe electrochemical interfaces, ranging from classical MD to Ab Initio MD or Wave function and Quantum Monte Carlo methods [85]. On the other side of the length scale, continuum models have been exceptionally beneficial to the design of efficient electrochemical devices such as batteries, flow cells and electrolyzers [16–18, 86]. Models at this scale are used mostly to guide the overall design of these devices, find optimal structural characteristics, solve integration problems and predict the behavior of these systems.

In the case of flow batteries and fuel cells, performance depends on achieving high current densities. This is achievable using porous electrodes that have been shown to be superior to planar electrodes due to their increased electroactive surface area [19, 87]. Accurate modelling of porous media is therefore crucial for the understanding and optimization of such devices. While atomic-scale models are applicable for porous electrodes and can help understand underlying mechanisms of charge transfer and surface interactions, they remain difficult to scale up to the device level where its structural properties and geometry have significant effects. On the other hand, at the other end of the scale, continuum models often consider porous regions of the device as homogeneous media and use bulk effective parameters without explicitly modeling the interfacial interactions in the pores [23]. Bridging these two extremes are the so-called "mesoscale" models, which are divided into two categories: fine-scale Computational Fluid Dynamics (CFD) methods such as the finite volume and finite element methods, coupled with meshes representing the porous micro-structure, or particle based methods such as the Lattice Boltzmann Method [21, 22], dissipative particle dynamics and smoothed-particle hydrodynamics [23].

The lattice Boltzmann method has been shown in literature to be a powerful tool for modelling of porous media, due to its amenability to complex fluid solid boundary conditions [88, 89]. Fu et al. (2022) [90] used LBM at the pore scale to simulate the gas diffusion layer of a flow cell and study wetting phenomena by reconstructing the porous regions in three dimensions as cylindrical fibers that are randomly positioned and oriented. Concerning redox flow batteries, Ma et al. (2019) used a similar porous geometry reconstruction technique for a multiple relaxation time based lattice boltzmann simulation with convection, diffusion and migration. Their results show that decreasing the porosity leads to increased fuel utilization even at similar pumping power [22]. Furthermore, in 2020, Zhang et al. [21] studied the influence of pore microstructures on the performance of redox flow batteries using LBM. Their analysis of pressure drop showed that carbon cloth which has a dominant pore size peak near 10 μm with fewer larger pores between $60 - 160$ μm, presents the lowest pressure drop when compared to Freudenberg and SGL paper, which minimizes the operational cost due to the pumping power requirements in RFBs.

In terms of pore microstructure representation, a few approaches are possible. By far the most common is three dimen-

sional imaging techniques through X-ray computed tomography (XCT) [91]. In this method, the three dimensional image is voxellized or converted into a format that can be integrated into the numerical solver [92, 93]. This is also the case with focused beam ion-scanning electron microscopy [94], which is a technique that slices the sample with an ion beam while imaging successive planes of the sample. This technique is destructive due to the high energy of the beam compared to x-rays, but it can be used when XCT does not provide enough contrast between the features of a given sample [23]. Another possible approach is that of stochastic reconstruction [20], which allows one to generate microstructures with a wide range of properties and compositions efficiently and without the constraints and limits of experimental imaging methods. Possible stochastic techniques include Monte Carlo based methods, in which microstructure regeneration is based on some probability distribution, and particles are generated and continuously placed in the domain according to this distribution until a specified volume fraction is attained [95]. Similarly, Dynamic Particle Packing and Simulated Annealing rely on defining a certain form of *energy* that is a function of the configuration of the system and iteratively evolving it until equilibrium is reached [96–98].

A general and efficient model with a focus on membraneless redox flow batteries, which uses microfluidics and laminar flow along with porous electrodes, is needed. MMRFB performance is heavily dependent on the structure and geometry of the cell. Therefore, in this work, LBM is used to simulate the flow battery at the pore-scale. Unlike most LBM models, such as that of Mukherjee et al. [99], which are a description of flooding/wetting phenomena in order to quantify mass transport losses, the hydrodynamic and electroactive species transport equations are coupled for a more complete description of the device. Previous publications that have used LBM to model redox flow batteries with Ion Exchange Membranes (IEMs) [21, 22, 90] have used a full 3 dimensional scheme to solve the hydrodynamic, advection-diffusion and Poisson equation for the electric field in the entire computational domain. However, membraneless redox flow batteries have mostly been modeled using standard CFD methods using commercial software and cconsist of planar or 3D electrodes with simple geometries [75, 100–103]. For instance, Bazylak et al. used a finite volume based CFD model to simulate a T-shaped fuel cell with planar electrodes [100]. They optimized the aspect ratio of the central microchannel

and proposed a tapered electrode design to improve fuel utilization up to 50%, which is high for a planar electrode design. Marschewski et al. (2017) used a finite element based commercial CFD package to simulate a membraneless redox flow battery including herringbone-shaped flow promoters in order to improve mass transfer towards the planar electrode [102]. In cases where porous electrodes are considered [104–106], the porous media are modeled as macro-homogenious regions with darcy flow as previously mentioned. Krishnamurthy et al. (2011), for example, used a finite element based model to simulate a redox flow battery with porous electrodes [106]. In the porous regions of their simulation domain, Darcy's law is used to dictate fluid flow, the carbon paper electrodes are therefore modeled as homogeneous porous media with a given permeability. Similarly, Li et al. (2016) also use the finite element method to simulate their device, but use the Brinkman equation instead of the Navier-Stokes equations in the porous regions [104].

In light of the above, there is a gap in literature concerning the modeling of membraneless redox flow batteries with porous electrodes. In fact, given the fact that flow and transport phenomena in microfluidic devices tend to occur mostly in the two dimensional plane of the chip, it should be possible to capture the relevant kinetic and mass transport phenomena in two dimensions. Furthermore, due to the use of highly concentrated supporting electrolytes when compared to the active redox species, the migration term can be neglected and the dilute solution approximation can be used. In this work we will demonstrate that an efficient two dimensional model with these simplifications can adequatly describe these devices with a greatly simplified model and reduced computation time. Furthermore, LBM can be modified and extended to include mass transport phenomena and electrochemical reactions [107, 108]. Therefore performance characteristics such as generated current and overpotential can be implemented using a Butler-Volmer model on the electrode-electrolyte interface where electron transfer is assumed to occur. These electrochemical reactions are coupled to the bulk advection-diffusion equation using source terms that ensure mass and charge conservation.

1.4 CONCLUDING REMARKS

In this chapter, we have introduced the field of redox flow batteries, which have been shown to be a successful technology for

stationary and large-scale energy storage. Different categories of RFBs were discussed according to their redox active species or the supporting electrolyte. For instance, non-aqueous electrolytes such as ionic liquids have been used in literature to overcome the electrochemical stability window of water-based systems. Furthermore, various avenues have been explored in order to optimize the performance of redox flow batteries, such as the use of porous electrodes, as well as mitigating membrane associated issues. In particular, researchers have attempted to implement membraneless designs in order to reduce material and maintenance costs, as well as improve the ohmic resistance of the battery. In this regard, microfluidics has been shown to be a promising avenue of research, as it allows for the manipulation of fluids at small scales, which can be used to leverage laminar flow to prevent species crossover without a membrane. However, it should be noted that such microfluidic devices do change the potential domain of application of redox flow batteries. Instead of large-scale stationary storage, microfluidic redox flow batteries will be more concerned with portable and mobile applications. The objective of this work is to combine different aspects of the literature that have been discussed in this chapter, namely the use of ionic liquids as electrolytes, porous electrodes, as well as microfluidics, to develop a prototype microfluidic redox flow battery with high power output. In parallel, a computational model is developed alongside for the simulation of such devices. For this task, the Lattice Boltzmann Method is used due to its ability to model complex geometries, which is important for devices comprising porous geometries. Futhermore, Lattice Boltzmann models can be modified to include electrochemical reactions, as well as mass transport, which is necessary for the simulation of redox flow batteries.

2

CHAPTER 2: EXPERIMENTAL PROTOTYPING

The objective of this project is to explore the optimization paths that are available to membraneless microfluidic redox flow batteries. While the scope of this work is to focus more on achieving this through modelling and simulations using numerical methods, it is important to accompany it with appropriate experimental works in order to get a measure of the parameters that influence the device and consequently have more potential for unlocking higher performances. In this chapter, we describe the methods, design choices, experimental setups and techniques that were used to a construct the device and evaluate its performance.

First, a discussion about how the performance of the cell is measured and evaluated. This is carried out using the polarization curves of the cell, which give insight into the power output of the device, as well as the limiting phenomena that are present in given operation conditions. Secondly, we describe the various materials that were used for manufacturing of the device, namely the resin that was used as the main structural material, the ionic liquid that was used as the supporting electrolyte, and the redox species that were used as the active species. We then elaborate on the fabrication techniques that were used, which in our case is 3D printing (specifically 'Stereolithography (SLA)' or 'resin printing'), which is a flexible and cheap fabrication method that is less common for microfluidic devices. We demonstrate that the resolutions that can be achieved with stereolithography are sufficient due to the relatively large scale of the microfluidic device, as laminar flow can be achieved due to the viscosity of the used electrolytes. We delve thereafter into the in-house synthesis procedure for the ionic liquid, which is chosen for its high conductivity and wide electrochemical stability window. We then describe the design choices that we made for the geometry and architecture of the device, specifically for the flow channel geometry and the electrode design. Followed by a description of the experimental setup that was used to measure the polarization curve of the device, and finally we present the results and discuss them in the context of the design choices that were made.

2.1 PERFORMANCE EVALUATION THROUGH POLARIZATION CURVES

Evaluating the performance of a battery device consists of measuring its ability to provide electrical energy to an external load. In this regard, a method that can be used to gain insight into the overall performance of the device is through recording the current that the battery is able to generate for a given external resistive load, or equivalently, the voltage drop across it. Performing these measurements in a range of different values of the load resistance allows one to obtain the polarization curve, which is a common metric that is used in literature. The electrical setup used for this purpose is shown in figure 2.1.

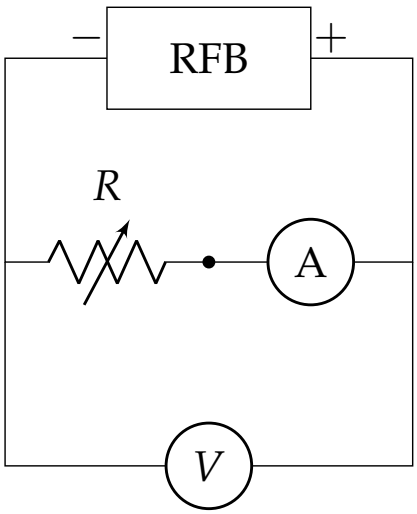


Figure 2.1: The electrical measurement setup for a polarization and power curve of a redox flow battery. The RFB is the power source, while the varying parameter is the load resistance R. Current and Voltage are measured using multimeters connected in series and in parallel, respectively.

To elaborate on this point, the polarization curve is a plot of the voltage drop across the device as a function of the current that is generated by it, therefore it can also be used as a measure of the power output of the device. During the operation of the cell, the operating point is a particular spot on the polarization curve and is dictated by the load resistance. In more graphical terms, the operating point is defined as the intersection of the polarization curve with the load resistance line that goes through the origin, as shown in figure 2.2. Considering Ohm's law $V =$

RI, which is a linear function of the current, the resistance R is the *slope* of the line.

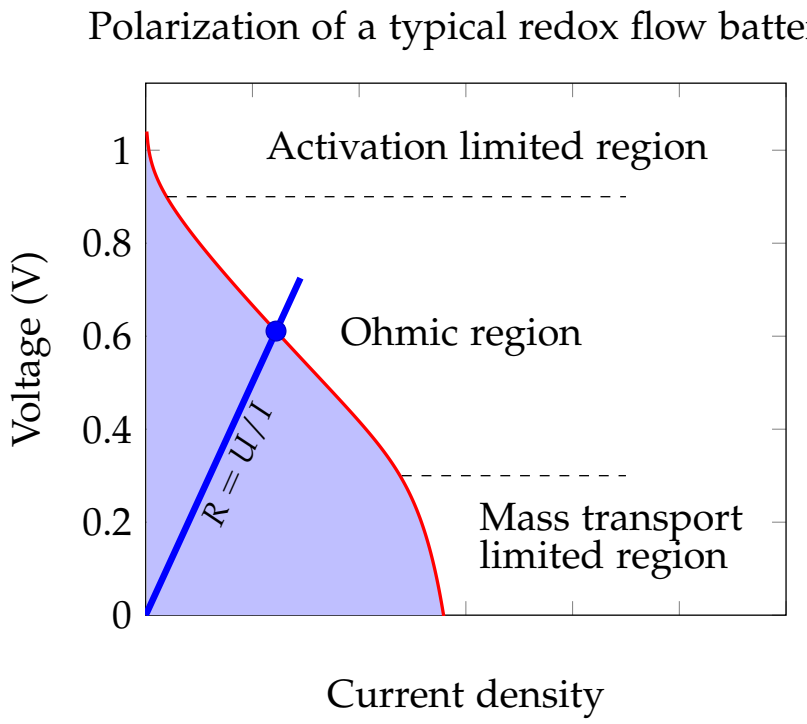


Figure 2.2: A typical polarization curve for a redox flow battery showcasing the different limiting regimes

As previously mentioned, the polarization curve is usually accompanied by a power curve plotting the power output of the device as a function of voltage. The characteristics that can be extracted from the polarization curve are the open circuit voltage, the short circuit current, as well as the maximum power point. These three parameters are typically associated with the different operation regimes of the device: As is shown in figure 2.2, we can distinguish three regions in the plot:

1. *The open circuit region*: In this region, the load resistance is considered infinite and no current flows through the device. Consequently, the voltage drop is at its maximum value, the slope of the curve is very negative and the voltage drops rapidly as current begins to flow. This is mainly due to the system being limited by the electrode activation losses. In fact, at low currents, activation overpotentials are the main limiting parameter of the device [3].

2. *The ohmic region*: As the current increases, the voltage becomes a linear function of the current. In this region, the limiting parameter is the internal resistance of the device, which typically includes the membrane, but in membraneless designs is mostly dictated by the ionic resistance of the electrolyte. Moreover, the slope of the tangent line to the curve in this region can be used to obtain the internal resistance of the device. Additionally, concentration overpotential can also affect the behavior of the system in this region. Nonetheless, this operational regime usually corresponds to the maximum power output point, and is the preferred operating point for the device [109].

3. *The short circuit region*: In this region, the load resistance is considered zero and the current is at its maximum value. The voltage drop across the device is therefore also zero. Due to the high current, the limiting factor in this region is the concentration of the reactants which are being consumed at the electrode surface. The performance of the device is consequently limited by mass transport of the reactants to the electrode surface, either through diffusion or convection [110].

Now that the main performance metric has been defined, along with the electrical circuit setup used to evaluate it, we proceed to discuss the materials that were used for the construction of the device, as well as the experimental setup used to operate the redox flow battery.

2.2 MATERIALS

The main guiding principle of the materials that were used in this project is to ensure low cost and ease of use and manufacturing. Consequently, we opted to use commercially available carbon felt sheets as the electrode material, and Formlabs' *ClearV4* resin which is supplied for their *Form 3* SLA 3D printers for fabrication of the device. The carbon felt electrodes are *6mm* thick cheap off-the-shelf carbon felt sheets (shown in figure 2.3) that are manually cut into the required dimensions and glued to copper tape contacts using conductive glue.

Figure 2.3: Picture of the carbon felt sheets that are used in this work. The sheets are 6 *mm* thick (uncompressed) and can be manually cut in the desired shape according to the device geometry.

2.2.1 *Electrolytes*

In terms of electrolytes, what one finds in literature is mostly aqueous or water-based systems. The combination of Ionic Liquids with membraneless redox flow batteries was explored in [13]. The attractiveness of ionic liquids lies mainly in the typically wide electrochemical window, as well as the relatively high shear viscosity. Having a wide electrochemical window is important in order to ensure that the electrolyte is stable at the operating potentials of the device and does not undergo any side reactions at the electrode surface, which would lead to a degradation of the electrolyte and consequently a loss of performance. Moreover, the large width of the electrochemical stability window of ionic liquids allows for more combinations of redox species to be used, as the standard potential difference between the two species can be larger. This, in turn, leads to a higher open circuit voltage for the device which makes achieving higher power outputs more feasible. Admittedly, the relatively high viscosity of ionic liquids is usually considered a drawback, as it leads to higher pumping losses and consequently lower energy efficiency. It is also the case that higher viscosities tend to correlate with lower ionic conductivities and low diffusion coefficients of active species. However, in our case, higher viscosity should have a positive effect due to the fact that we are not working in the typical microfluidic length scale of a few μms, but at a relatively larger scale of a few millimeters. The increased viscosity is therefore important to ensure that we

remain in a laminar regime with a small Reynolds, that is at the order of $Re \leq 1$.

In this work, the IL that used in this project is mainly ***emimTFSI!*** ***(emimTFSI!)***. This choice has mainly been informed by the work in [13, 111]. The main result concerning conductivity from Khalil et al. (2020) in [111] is that the conductivity decreases progressively as the alkyl chain increases. Therefore, Emim[TFSI] exhibits the highest conductivity compared to similar imidazolium-based ionic liquids such as Bmim[TFSI] or Dmim[TFSI]. Therefore, in the following section is a description the process I used to synthesize said Ionic Liquid in the lab. This synthesis process is suitable for batches of around $200mL$ of ionic liquid, which is enough for an extended run of polarization experiments for microfluidic systems that manipulate small amounts of liquid.

2.2.1.1 *Emim[TFSI] synthesis*

Due to the wide variety of possible combinations of anions and cations, there is a wide range of synthesis techniques that are used for ionic liquids [112]. As previously mentioned, Emim[TFSI] was chosen due to its wide electrochemical stability windows, a relatively high viscosity that ensures laminar flow and limits crossover, as well as its high conductivity. An added benefit is that fact that it shows good wettability in the carbon felt electrodes that are used in this work. This is beneficial as it ensures that the electrolyte is able to penetrate the porous electrode and reach the active sites on the surface for electrode transfer. In fact, before assembling the cell, the electrodes are submerged in the ionic liquid in a vacuum chamber to extract air bubbles as described in section 2.5. Moreover, good wettability reduces pumping power required to circulate the electrolyte through the device.

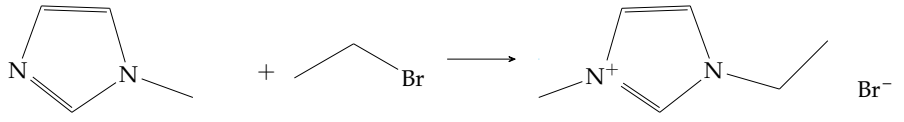
The synthesis of Emim[TFSI] was performed in-house is for reasons of purity and reproducibility. Through the synthesis procedure, we are able to control the water content of the ionic liquid, which is significant as water impurities lead to a reduction in its performance as a solvent and electrolyte. Furthermore, in-house synthesis allows us to ensure reproducibility of our results as we use the same procedure for each batch of ionic liquid that is used in the experiments.

Emim[TFSI] is an imidazolium-based aprotic ionic liquid, and its synthesis is divided into two steps:

1. Quaternization of 1-methylimidazole

2. Anion exchange

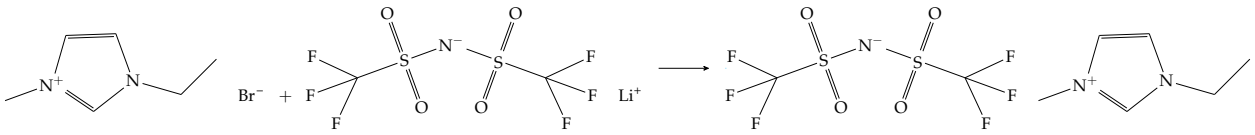
First, the quaternization reaction in the following reaction of the 1-methylimidazole is performed using the methylimidazole with an excess of the bromoethane with a molar ratio of 1 : 1.1, in order to produce the imidazolium salt EmimBr.



The quaternization reaction: methylimidazole + bromoethane → EmimBr

This is performed in the presence of ethyl acetate (around $50 \sim 100\ mL$ of ethyl acetate) and is agitated at a temperature of 75°C for 24 hours. The resulting EmimBr salt typically crystallizes at room temperature. The remaining ethyl acetate is filtered out with a vacuum pump as well as by heating to a temperature above 70°C.

The second step of this procedure, which consists of exchanging the anion, is performed as follows: The two components are mixed with a molar ratio of 1:1, but with a slight deficit of LiTFSI, and are agitated at ambient temperature for a minimum of 24 hours. The mixture separates into a phase containing the ionic liquid, and an aqueous phase containing the highly hygroscopic Lithium Bromide (LiBr).



Anion exchange reaction: EmimBr + LiTFSI ⟶ EmimTFSI

The next steps consist of progressively extracting the remaining IL from the aqueous phase using an organic phase such as dichloromethane, followed by a washing procedure to remove the remaining traces of LiBr from the IL phase. This last step is achieved using the hygroscopic nature of LiBr to extract it from the IL phase, with consecutive extraction steps and periodically testing the aqueous phase for its LiBr content using silver nitrate.

The last step is to remove the organic phase from the IL under agitation at about 50°C by using a vacuum pump and liquid nitrogen cooled trap for the dichloromethane.

Once the organic phase is completely removed, we further reduce the water content by freeze drying the sample and testing it with Karl Fischer titration. If the water content is less than 50 ppm, the ionic liquid is ready to be used as an electrolyte for the redox flow battery.

2.2.1.2 *Active species*

As was mentioned in the introduction of this work, redox flow batteries rely on the electrochemical reactions of two redox species that are dissolved in the electrolyte. The main criteria of that dictate the choice of an active species are as follows:

- *High redox standard potential difference*: The standard redox potentials of the two species should be as widely spaced as possible within the electrochemical stability window of the underlying electrolyte, in order to maximize the open circuit voltage of the device and consequently the power output.
- *High solubility in the electrolyte*: The solubility of the species in the electrolyte should be as high as possible in order to maximize the energy density of the device.
- *Reversibility*: The redox reactions should be as reversible as possible in order to minimize losses due to a high required overpotential.
- *Low cost*: The cost of the species should be as low as possible in order to minimize the cost of the device.

In literature, it is common to pair organic or ionic liquid based electrolytes with metallic compounds such as [80]. However, they tend to be expensive, toxic and not very soluble in non-aqueous environments. In fact, most metal oxydes such as $V(acac)_3, MnO_2$ and VO_4 have a solubility limit of the order of a few millimoles per liter in ionic liquids [13]. Organic redox species are therefore a more attractive alternative due to their low cost and relatively high solubility. In light of the work of Chaabene et al., and in order to ensure similar conditions to enable a consistent comparison of different cell geometries, we combine in this work the organic compound of Quinone with a metallic ion, the Iron(II) Chloride, $FeCl_2$. The corresponding redox reactions are:

$$Q^0 + 2e^- \rightleftharpoons Q^{2-} \tag{2.1}$$

$$Fe^{2+} \rightleftharpoons Fe^{3+} + e^- \tag{2.2}$$

The Quinone reaction 2.1 has been shown to occur in two distinct steps through an intermediary radical anion $Q^{\bullet -}$ [113], but in emim[TFSI], it occurs in a single step. Table 2.1 shows the peak potential drifts for varying CV sweep rates as well as the standard potential of the active species in emim[TFSI].

Redox species	Maximum concentration (M)	Standard potential E°(V) vs Fc^+/Fc
Quinone	0.3	0.51
$FeCl_2$	0.1	-0.11

Table 2.1: Peak potential drift and standard potentials of the redox species of Quinone and Iron(II) Chloride in Emim[TFSI] [13].

Given this information, the maximum theoretical cell voltage expected of this system is around 620 mV. However, the actual cell voltage is expected to be lower due to the activation overpotential of the electrodes, as well as possible shunt currents due to the presence of defects and stray carbon fibers between the electrodes. Another added benefit of this system is the contrast between the colors of the two electrolytes, which will allow for a visual confirmation of the formation of the laminar interface in the flow channel.

2.2.2 *Resin printing*

As it was mentioned in the previous chapter, the main fabrication technique that is used in microfluidics is soft-lithography, which requires the creation of a mold that can be replicated using a polymer [114]. The process of photolithography typically requires a clean-room environment, which implies significant costs. However, recent work has shown that 3D printing is a promising alternative to the widely used soft-lithography process [115]. In fact, the main challenge facing 3D printing is the

resolution of the printed parts. In microfluidics, microchannels with a width of 5 μms are common, whereas the highest resolution 3D printers have a resolution above 10 μm[115].

In fact due to our use of the Emim[TFSI] ionic liquid (see section 2.2.1.1), which has a kinematic viscosity of $\nu = 2 \times 10^{-5}\ m^2/s$ [13], the width of the channels that can be used is therefore much larger than typical microfluidic devices and is at the millimeter scale. To elucidate this point further, given an upper bound for flow velocity of 1 cm/s in a channel that is 1 mm wide, the Reynolds number is therefore:

$$Re = \frac{0.01\ ms^{-1} \times 10^{-3}\ m}{2 \times 10^{-5}\ m^2 s^{-1}} = 0.5 \tag{2.3}$$

This value of the Reynolds number is still well in the laminar regime despite having a relatively large characteristic length scale of 1 mm. It is therefore possible to use 3D printing to manufacture the prototype device. In this work we use a proprietary 3D printer from Formlabs which is based on stereolithography. The working principle of this technique is using a laser to selectively cure a photosensitive resin polymer layer by layer. The commercially available *Form 2* 3D printer that we use in this project is able to achieve printing resolutions of 25 μm in the XY plane, and is therefore sufficient to directly print the device with its channel geometry.

The mechanical properties of the resulting prints after UV curing for the commercial *Clear V2*^{TM} resin that we use are discussed specifically in more detail in [116] and [117], but the main guideline is to keep curing temperature below 45°. We should note however that the main concern in our application is less the mechanical strength of the prints, but rather the deformation of parts during the post-curing process that may lead to misalignment between the different pieces of the device, and consequently to potential leaks, which can pose a considerable challenge.

2.3 CHANNEL DESIGN AND GEOMETRY

This project is the successor of the work previously done by Chaabene et al. [13] which evaluated the use of ionic liquids in microfluidic redox flow batteries. The channel design that was used is similar to the one initially introduced in the seminal paper of Ferrigno et al. [71] using a Y-shaped channel geometry

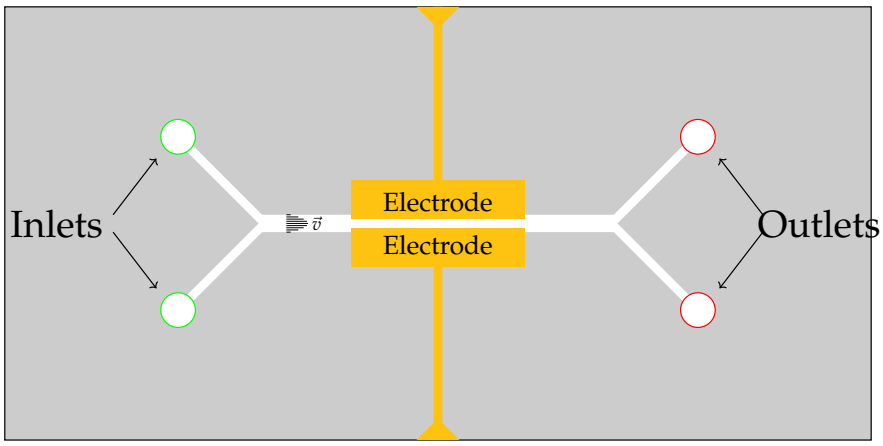


Figure 2.4: Sketch of the planar electrode PDMS/glass microfluidic redox flow battery

with laminar flow to eliminate the need of a membrane as is shown in figure 2.4. Ferrigno's device, however, was water-based and used a sulfuric acid electrolyte. The main idea is to introduce the reactive species in the inlets, so as to form the laminar interface in a main central channel at the junction of the two branches of the 'Y'. The floor of this channel contains planar two-dimensional electrodes which are integrated or printed on the substrate. The reactive species need to reach the electrode surface for electron transfer to occur. This approach is very common in the literature, and is used partly due to available fabrication techniques that are common in the field of microfluidic fuel cells, specifically photolithography and soft lithography techniques. The diagram in figure 2.4 is a top view of the device and show all the features on the plane of the glass substrate, with electrodes being printed on top of it using lithographic techniques. The PDMS component is sealed on top of it to form the flow channels. Hence this design is known in literature as a monolithic "flow-by" or "flow-over" device.

There are two main design limitations of this approach: The first is the limited surface area of the electrodes, which is equal to the geometric area of the electrodes that are printed on the substrate. The second limitation is low reactant utilization: Microfluidic devices typically experience flows with high Peclet numbers, which implies the formation of a thin diffusion layer in front of the electrode surface. Consequently, a significant portion of the reactants that flow through the channel is unable to reach the electrode and exits the device without undergoing

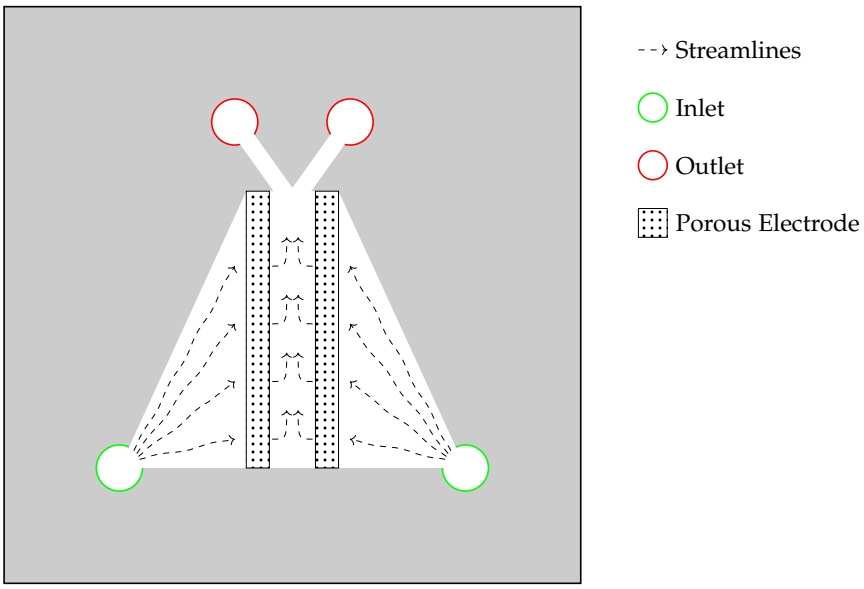


Figure 2.5: Sketch of the flow through design architecture with porous electrodes

a reaction and is therefore wasted. This reduces the reactant utilization and significantly limits the current that can be generated by the device. The nature of the flow and the geometry of the planar channel electrodes also result in a significant portion of the generated current density occurring at the leading edge of the electrode where active species can react first. The formation of the diffusion layer next to the electrode therefore means that the remaining area of the electrode does not contribute as much to the current density.

In order to mitigate some of these limitations, in this work we opted for a flow-through design which is shown in figure 2.5.

In this design, the planar electrodes are replaced with by three dimensional porous electrodes that occupy a volume within the flow channels. An immediately apparent advantage of this concept is that the available surface area for charge transfer is much greater due to pore microstructure. The reactants flow through the pore space of the electrode which spans the full vertical extent of the flow channel. Consequently, the likelihood of the reactants encountering a reaction site on the surface of the electrode is enhanced, obviating the necessity of reactants to travel all the way to the channel floor for reaction to occur. Fuel utilization is therefore improved significantly in this design.

A possible concern that one may have with this concept is the issue of wetting and trapped air bubbles reducing the effective

surface area of the electrode. Both of these problems will be addressed in the experimental setup section (see the experimental setup section 2.5) as these issues have more to do with the operational side of our work and can be effectively avoided by carefully setting up the polarization measurement experiment.

2.4 FABRICATION AND ASSEMBLY PROCESS

In order to construct a cell with a flow through design using SLA, we use an approach that is similar to the PDMS-based microfluidic devices that are common in literature [114]. A widely used strategy consists of sealing the molded PDMS part to a substrate which is typically either also PDMS or glass.

Similarly, using the stereolithography approach, the device is composed of two different printed pieces: The first one contains the channel geometry as was shown in figure 2.5, while the second one acts as a substrate and contains grooves and slots for the electrodes and contacts.

These pieces first have to be modelled in CAD software and are shown in figures 2.6 and 2.7.

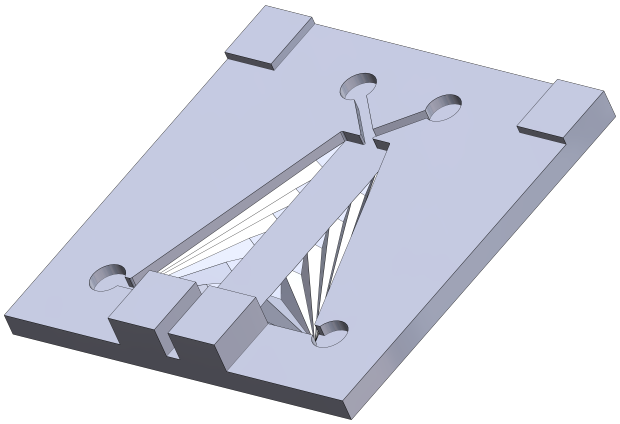


Figure 2.6: CAD model of the piece containing the flow channels

The channel piece in 2.6 contains four notches that are extruding from the main plane that acts as the interface of the two pieces. These act as alignment indicators during manual assembly, as well as enclosing the electrodes while simultaneously allowing the electrical contacts to protrude out of the device without any gaps or leaks. There are four circular areas for two inlets and two outlets that are cut or recessed into the surface. These are intended to interface with the holes that are designed in the cover piece that will be directly linked to input

or output reservoirs through capillary tubes. The cut extrude length corresponds to the height of the flow channel and is uniform throughout the entire device. In fact, the trapezoidal area between the inlets and outlets is recessed into the surface by the same length. This area is split into three zones:

1. A pre-electrode zone that progressively expands from the inlet channel into the full length of the electrode and leads the reactant flow into the next area. The shape of this zone is designed to ensure uniform flow distribution and deliver reactants at a constant rate across the length of the electrode. The grooves that are added to the surface also contribute in this regard, but with the added benefit of guiding any trapped air bubbles towards the inlet and out of the device. This can be performed by running the flow through the device in reverse and allowing the air bubbles to escape through the inlet.

2. An active zone occupied by the porous electrode where the redox reactions occur. Given the fact that the electrodes span the full height of the channel, the reactants are therefore forced to flow through the electrode pore space before reaching the main channel. It is therefore important to ensure that there are no leaks in the channels that allow the reactants to bypass the electrodes, as this would significantly reduce reactant utilization, and consequently hinder the of the battery.

3. A main channel where the interface between the two components of the device is formed and that leads the flow into the two outlet channels. It this region, there are two main considerations: First is keeping the flow laminar and the streamlines parallel to limit reactant crossover, and second is to reduce the separation distance between the electrodes as much as possible to reduce the internal resistance of the device due to the ionic resistance of the electrolyte. Due to the fact that the device is manually assembled and consists of porous carbon felt electrodes, it is difficult to reduce this distance below 1.5 mm as this would lead to a significant risk of fibers from the electrodes protruding out and short-circuiting the device. This is a limitation that is inherent to the assembly process and is difficult to overcome without using more advanced fabrication techniques.

The pre-electrode zone connects the inlets to the electrodes and contains linear and periodic extrusions and cuts in order to

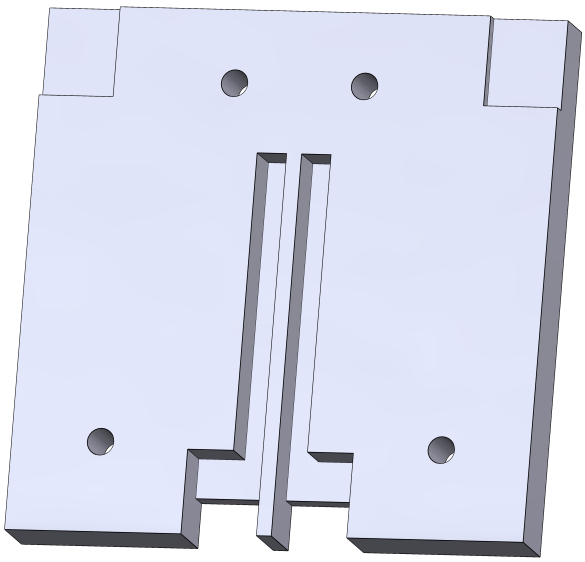


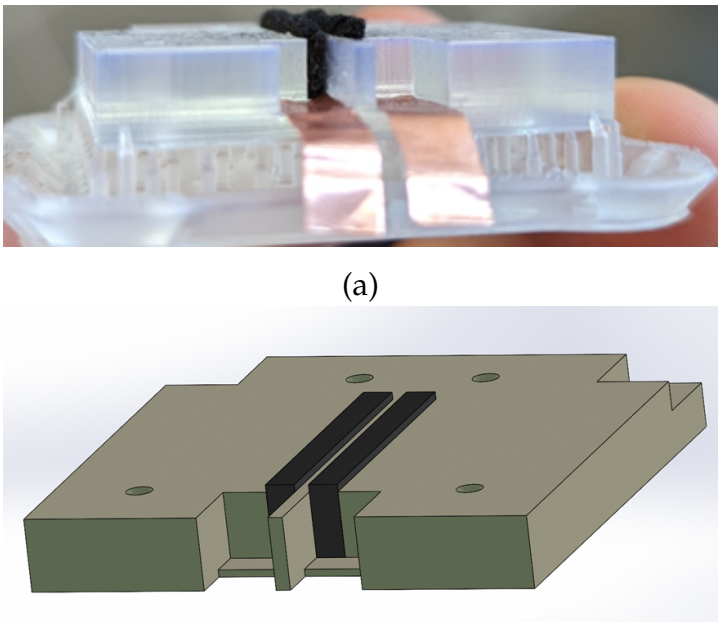
Figure 2.7: CAD model of the cover piece

uniformly distribute the flow across the length of the electrode. In this configuration, we can avoid preferential flow along the left/right walls of the expanding channel. This is important especially if the electrodes do not seal off the main channel, thereby allowing reactants to leak through the device without reacting. The main interfacial channel is located between the two electrodes and is where ion exchange occurs in a colaminar flow configuration. The two electrodes have to be far enough apart to allow such an interface to form and avoid short-circuit contact, but narrow enough to reduce the battery's internal resistance which is directly proportional to electrode separation distance and the electrolyte conductivity [79] . This flow-through approach is inspired by literature from the field of fuel cells such as Kjeang et al. [77] and Goulet et al. [79], but is novel for ionic liquid based redox flow batteries.

Appendix A. Supplementary material of [79]

We use copper tape as electrical contacts which are glued to the electrodes using conductive epoxy. The copper tape is inserted between the cover piece and electrodes and is held in place by the slots that are designed and cut from cover piece. The thickness of the copper tape is negligible compared to the height of the electrodes, and the resulting compression force is sufficient to ensure good electrical contact between the electrodes and the copper tape, as well as avoiding any leaks. Figure 2.8 shows the cover piece with the electrodes and copper tape in place. It should be noted that the electrodes extrude slightly from the main interfacial plane of the cover piece. This

extrusion length is slightly larger than the height of the channels that are cut into the channel piece. This is to ensure that the electrodes span the full height of the flow channel.



(a)

(b)

Figure 2.8: Cover piece with electrodes inserted into their respective slots. (a) The printed part with the copper contacts and carbon felt electrode. (b) CAD model of the same piece with electrodes shown in black. Note the slight extrusion of the electrodes above the main plane.

Both printed parts are then cleaned using isopropyl alcohol and are individually cured in a UV oven for 30 minutes at room temperature. The UV lamp used is commercially available from *FormLabs* and uses multi-directional LEDs at 39*W* with possible operating temperatures up to 80°C. Since there are no specific mechanical requirements for the device apart from preserving the integrity of the device during operation, the prints are not heated during the curing process to avoid deformation of the parts that may lead to misalignment and leaks. Consequently, curing at above room temperature is performed after the assembly.

Once the electrodes and contacts are inserted into place, the two pieces are subsequently glued together by coating the interfaces with a thin layer of resin followed by a second round of curing in the UV oven. The resulting device is shown in figure 2.9 and is now ready to be used in the experimental setup.

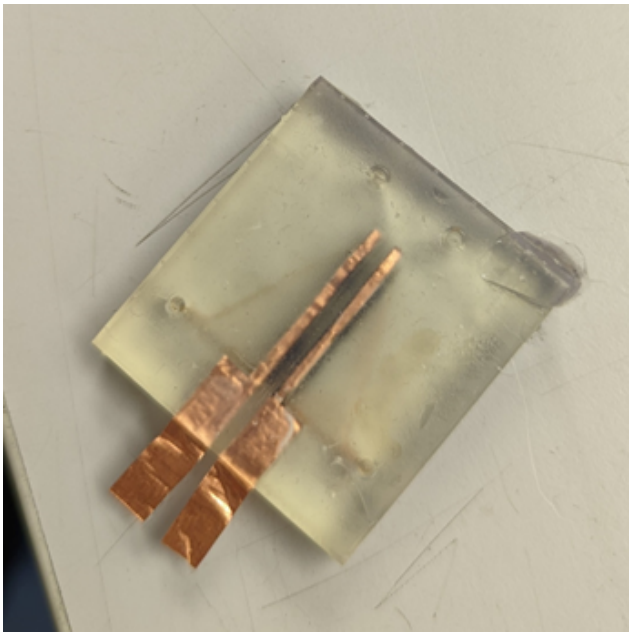


Figure 2.9: The full device with the two pieces glued together.

2.5 EXPERIMENTAL SETUP

The main performance metric that is used to test a microfluidic battery is the current/voltage (I/V) or polarization curve. This is performed either by measuring the current that is generated by the device as a function of the applied voltage or by connecting the device to an external variable resistive load and measuring the current drawn from the device as a function of the voltage drop across it. In this section we describe the experimental setup that we use to measure the polarization curve of the device.

In order to operate the device, a constant flow of the reactants being introduced through the inlets is required. Historically, this was achieved by using syringe pumps to drive the flow due to their ease of use and availability. They do however have the disadvantage of slow response times and flow oscillations from the stepper motor. It is also possible to utilize gravity to drive the fluid flow across the device by placing the inlet reservoirs at a higher elevation than the outlets, but it is more challenging to precisely control the flow using this method.

In this work, we use a pressure based flow controller instead. These devices enable the flow by applying a constant pressure to the fluid reservoirs which are air tight and are connected to the device inlets through capillary tubes as shown in figure 2.10. The inlet pressures are controlled using a computer through software. In our specific case, we use the *Fluigent MFCS-EZ* controller

which is capable of controlling up to 4 different channels and is operational on a wide range of pressures from 0 to 2100 mbar, which is particularly useful for our application due to the relatively high viscosity - and consequent pressure drop - of the ionic liquid that we use.

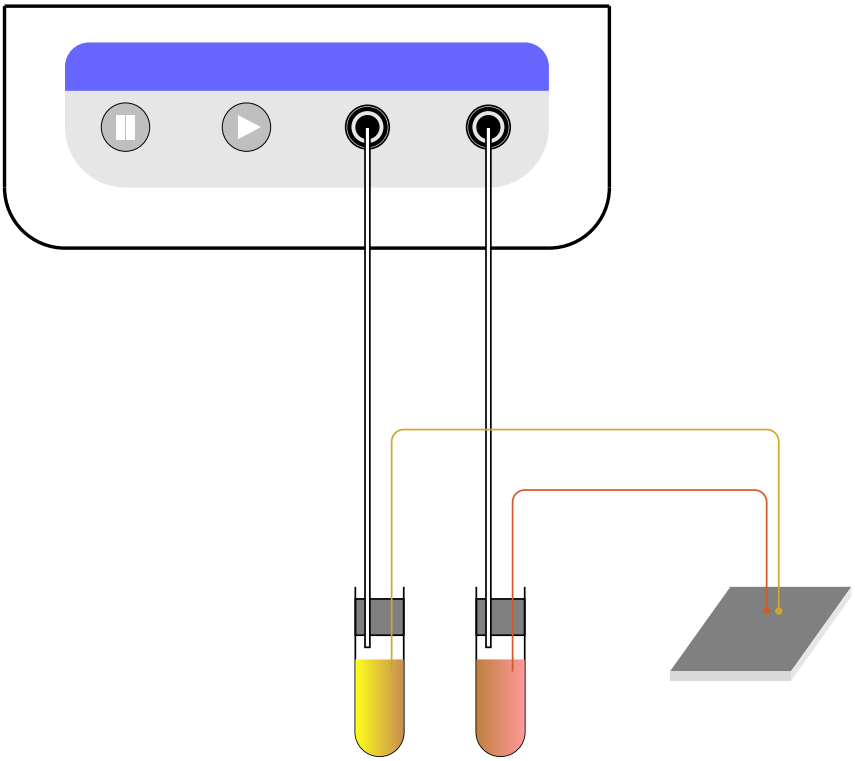


Figure 2.10: Device setup with the pressure based flow controller

Once the device is interfaced with the flow controller on the inlets, and the outlets are connected to the waste reservoirs, the device is cycled with the ionic liquid in order to remove any air bubbles that may be trapped in the flow channels. Due to the fact that the capillary tubes are connected manually, it is possible that some air bubbles may be introduced into the device. Bubbles in the central channel will eventually flow downstream and exit the device, but bubbles that are in the pre-electrode zone may get trapped as they are unable to traverse the electrodes. This is problematic as it reduces the effective surface area of the electrode and therefore the current density that can be generated by the device. It is therefore important to ensure that any such bubbles are removed, and this is achieved by cycling ionic liquid backwards through the device so that the bubbles are pushed out through the inlets.

Finally, when the flow channels are free of air bubbles, the device electrical contacts are connected to a the external circuit in order to initiate the polarization measurement as shown in figure 2.11.

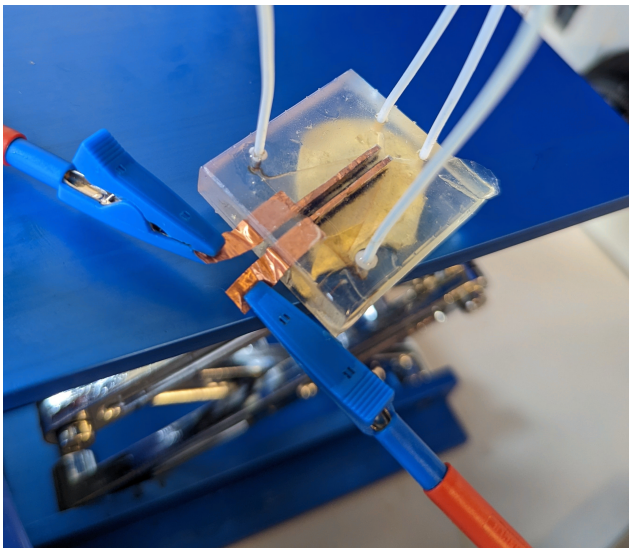


Figure 2.11: Experimental setup with the device interfaced with the pressure based flow controller through capillary tubes and the external circuit

2.6 RESULTS

First, one initially needs to run the device for a few minutes in order to ensure that the ionic liquid is fully wetting the electrode surface and flow channels. Once we are confident that the flow field is fully developed and that the device is operating in a steady state, we can start to take measurements for different values of the load resistance. Given that the Clear $V4^{tm}$ resin that we use is semi-transparent, we can confirm the formation of the laminar interface between the two reactants by visual inspection. The interface is clearly visible as a sharp boundary between the two liquids in figure 2.12.

The reactants used in this experiment are the Iron (II) Chloride redox couple and Quinone (Q/Q^{2-}) which have different colors when dissolved in ionic liquid and allow us to visually confirm the formation of the interface as is shown in figure 2.12. We are therefore able to confirm that the fluid flow does penetrate the pores of the electrodes and subsequently form a laminar regime in the main channel. Hence there is no mixing or leaks occurring between the two components, which is crucial for membraneless

designs. Furthermore, this geometry includes two outlets that allow for the anolyte and catholyte to be recovered separately, which is important for electrolyte reuse and regeneration.

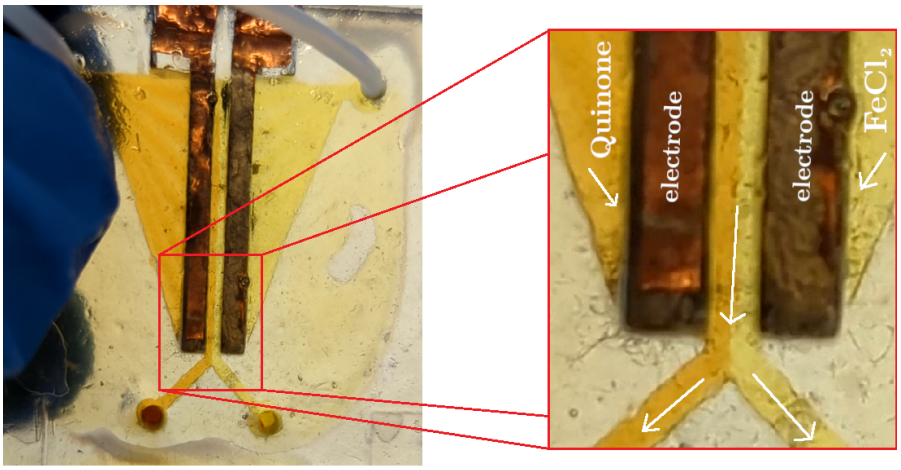


Figure 2.12: The laminar interface between the two reactants

The maximum concentrations that we were able to achieve for these two species are $0.3M$ for Quinone and $0.1M$ for $FeCl_2$. As previously mentioned (Refer to the active redox species section 2.2.1.2), these two species present a potential difference of $\sim 620mV$ which sets an upper bound on the open circuit voltage that we can expect from the device. The polarization curve that we obtained is shown in figure 2.13, in comparison to a planar electrode cell that was tested in identical conditions. We can see a clear sizable increase in performance for the flow through design in terms of power output and short circuit current. The flow through device is able to provide up to $3\mu W$ while the planar electrode cell is only able to provide less than $0.3\mu W$. The short circuit current is also increased by three orders of magnitude with the flow through cell outputting $300\mu A$ in these conditions as opposed to $4\mu A$ by the planar flow-by cell. It should be noted however, that the polarization curve obtained in figure 2.13 is for an average representative cell with an electrode separation distance of about $2mm$ and a relatively high internal resistance of 300Ω.

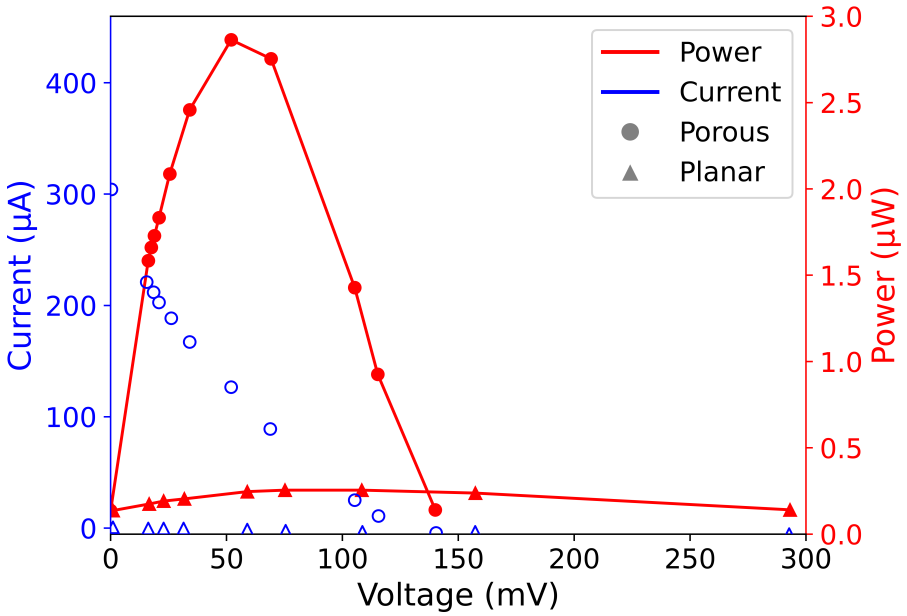


Figure 2.13: Comparison between the polarization curves and the power output of the flow-through device and the planar flow-by device.

Due to the manual nature of the assembly process, it is difficult to ensure a consistent performance across different devices. Four main issues have been identified that may contribute to this variability:

1. *Electrode alignment*: The electrodes are cut and inserted manually and are therefore not perfectly aligned in the flow channels. This leads to a variation in the electrode separation distance along the length of the main interfacial channel. It should also be noted that the active surface area of each electrode is likely to slightly vary from one device to another, and even within the same device.

2. *Shunt currents*: Due to the low separation distance between the two electrodes and the fibrous composition of carbon felt electrodes, a common issue that arises during assembly is the presence of stray fibers that extrude from the device and provide a low resistance path between the two electrodes. This often short circuits the device and leads to a significant reduction in performance.

3. *Contact resistance*: The copper tape contacts are also manually glued to the carbon felt electrodes and imperfect contact may lead to a significant increase in the internal resistance of the device.

4. *Electrolyte leaks*: The carbon felt sheets have an uncompressed thickness of about 6 mm, while the channel height is 5 mm. The rationale behind this is that the electrodes are slightly compressed within the channels in order to avoid leaks. However, imperfections during the cutting and assembly process can lead to leaks in the flow channels where the electrolyte can bypass the electrodes and flow directly through the device without reacting, leading to reduced reactant utilization.

Despite these challenges, the objective is to develop a proof-of-concept device that can be optimized in the future. This can be achieved through more comprehensive and systematic experimental works, as well as with the support of the numerical simulation model that will be presented in the following chapters.

Through trial and error, the maximum performance that we were able to obtain with this design using only manual assembly is shown in figure 2.14:

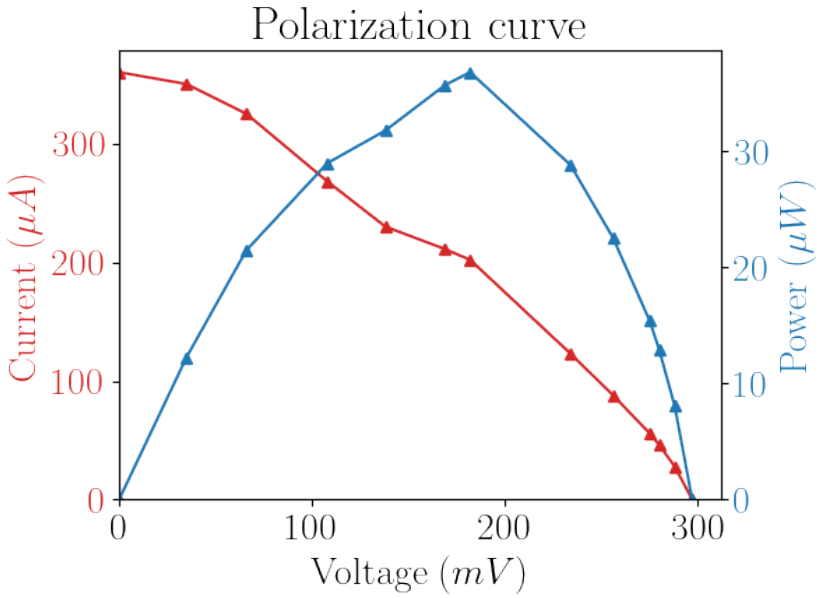


Figure 2.14: Polarization curve of the flow through device with the maximum performance that was obtained.

This device has an internal resistance of about 120Ω and a reduced electrode separation distance of 1.5mm. The open circuit voltage and short circuit current are 0.3V and 360μA,

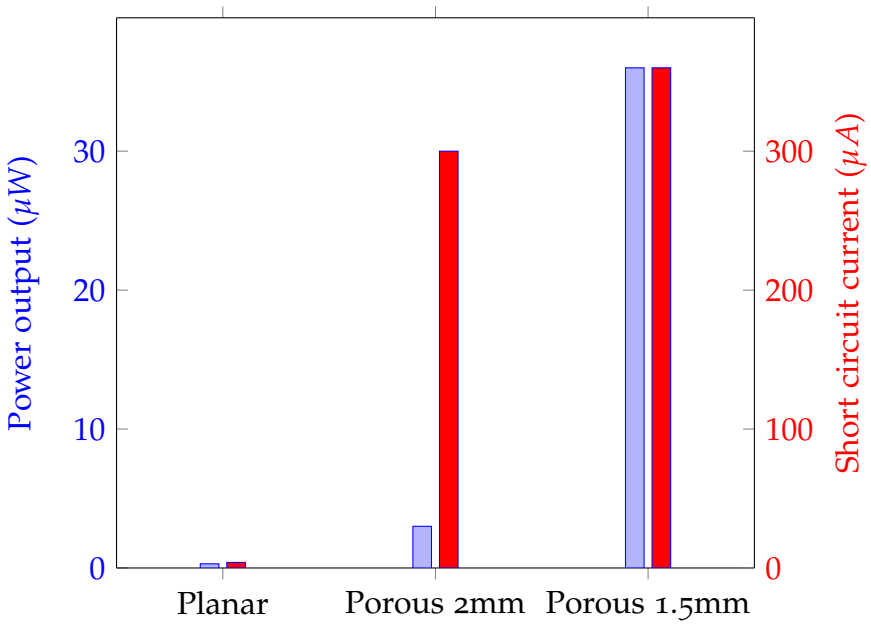


Figure 2.15: Comparison of the maximum power output obtained experimentally for the flow-by device with planar electrodes and the two flow through devices with different electrode separation distances.

respectively, for a maximum power output of $36\mu W$. The performance of this device is still significantly higher than the planar electrode cell, but is still far from the theoretical maximum performance that can be achieved with this design due to the relatively low open circuit voltage of the device given the redox species that are used.

Nevertheless, the flow through cell architecture is able to significantly improve the performance of a device given identical chemistry and operation conditions. Figure 2.15 shows a comparison between the maximum power output and short circuit current of the planar electrode cell and the two flow through devices that were tested. The flow through devices are able to provide a significant increase in power output and short circuit current compared to the planar electrode cell. It is also important to note that the separation distance between the two electrodes significantly influences the performance, as smaller distances lead to reduced ionic resistance within the electrolyte. However, this distance cannot be arbitrarily reduced as it is limited by the risk of short circuiting the device due to the inconsistencies engendered by manual assembly.

2.7 CONCLUSION

In this chapter we have presented the experimental setup that was used to measure the polarization curve of the device. We have also presented the results of the polarization curve and compared it to a planar electrode cell that was tested in identical conditions. We have shown that the flow through design is able to provide a significant increase in power output and short circuit current compared to the planar electrode cell. This is due to the fact that the flow through design is able to provide a larger surface area for charge transfer to occur, as well as improved reactant utilization due to the fact that the reactants are able to flow through the porous electrode and are not required to migrate to a two dimensional surface at the bottom of the channel. Improved reactant utilization is evident from the increased current output in all regimes for identical flow rates and concentrations of redox species. The flow through design is therefore a promising candidate for membraneless microfluidic redox flow batteries, and we have been able to show a proof of concept for this design, at relatively large scales of a few millimeters, and with Ionic Liquids as the supporting electrolyte. These results are a proof-of-concept that can be a stepping stone towards more comprehensive and systematic studies of even more optimized ionic liquids and the corresponding redox active species that are able to simultaneously provide a high open circuit voltage for the device, as well as a high solubility for increased energy density.

3

CHAPTER 3: THEORETICAL FRAMEWORK

The objective of this chapter is to construct a numerical model for microfluidic redox flow batteries at the device level in order to evaluate the effects of kinetics and mass transport limitations on their overall performances. We therefore begin by describing the relevant governing equations at this scale. First, the main governing equations for fluid flow are the Navier-Stokes equations [67, 80]. However, solving the governing equations of fluid mechanics is notoriously challenging [118]. They are a set of coupled non-linear partial differential equations for which analytical solutions are rare and can be found only for very basic geometries or simplified flows such as Poiseuille or Couette flow. Numerical methods are therefore required to resolve the velocity field at arbitrary geometries and boundary conditions. In fact, electrochemical devices are often characterized by complex geometries and boundary conditions [119]. Indeed, maximizing the performance of an electrochemical system typically hinges on managing and mitigating various limiting phenomena for charge transfer across an electrode-electrolyte interface, hence maximizing the extent of this interface means increasing the available active surface area of the electrode. Consequently, it is common to find porous electrodes in these systems as they greatly enhance the surface area compared to conventional planar electrodes [19, 120]. Nonetheless, numerical methods for fluid flow are themselves difficult to manage and implementing them in a computationally efficient way while resolving the physical phenomena that are relevant to a given problem is not a trivial task. In this work, we opt to use the LBM specifically due to its ability to handle complex geometries better than most direct Navier-Stokes solvers. This is due to the local nature of operations that are performed in the LBM algorithm, which do not require the determination of derivative approximations from neighboring nodes which is the source of a significant part of the complexity of direct solvers. In LBM, this complexity is transferred towards local operations, which gives it distinct advantages (and disadvantages) such as compatibility with parallel computing architectures [107].

Beyond the fluid dynamics, for electrochemical devices, the transport of chemical reactants and their production and consumption also need to be included. In this case, we will also use the Lattice Boltzmann Method to solve the transport problem. The second part of this chapter is therefore dedicated to describing exactly how this can be achieved while taking into consideration reaction kinetics and mass transport by extending and modifying the Lattice Boltzmann Method.

All things considered, the model should be a useful platform for the analysis and understanding of electrochemical devices and their limiting phenomena, as well as a predictive tool for their performances.

3.1 NAVIER-STOKES EQUATIONS

The first part of the model concerns the flow of electrolyte within the channels. Given the fact that we are working at the device level, which is at the millimetric scale, a continuum model is therefore required, hence the flow field, $\mathbf{u}$, is the solution of the incompressible Navier-Stokes equations:

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \eta \Delta \mathbf{u} + \mathbf{F} \tag{3.1}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{3.2}$$

where :

- $\Delta = \nabla \cdot \nabla = \partial_\alpha \partial^\alpha$ is the Laplacian operator
- $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$ is the material or convective derivative
- ρ is the density of the fluid
- p is the pressure
- η is the shear viscosity of the fluid
- $\mathbf{F}$ is a source term representing a body force acting on the fluid

In the incompressible limit, the continuity equation reduces to

$$\nabla \cdot \mathbf{u} = 0 \tag{3.3}$$

The shear stress term for the general Navier-Stokes equations reduces to the Laplacian $\eta\Delta\mathbf{u}$, which makes equation 3.1 a vector convection-diffusion equation for momentum. This will become important later for solving the transport problem with the Lattice Boltzmann Method.

In index notation, the Navier-Stokes equations can be written as:

$$\frac{\partial(\rho u_\alpha)}{\partial t} + \frac{\partial(\rho u_\alpha u_\beta)}{\partial x_\beta} = -\frac{\partial p}{\partial x_\alpha} + \frac{\partial}{\partial x_\beta}\left[\eta\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right)\right] + F_\alpha \tag{3.4}$$

Index notation is used for ease of comparison with the macroscopic limit of the Lattice Boltzmann Method which is shown later. When this equation is taken along with the continuity equation, the mathematical system is closed and can be solved for a given set of boundary and initial conditions.

Our objective is now to numerically solve these equations for the flow field, $\mathbf{u}$, in the channels of the device. Given that the LBM is a particle based mesoscopic numerical method that is founded on principles from statistical mechanics, the next section will explore the kinetic roots of LBM and how it can be used to solve the Navier-Stokes equations.

3.2 THE LATTICE BOLTZMANN EQUATION AT THE MACROSCOPIC LIMIT

The fundamental object of kinetic theory is the distribution function $f(\mathbf{x}, \mathbf{v}, t)$. It can be considered to be the generalized analog of the scalar density $\rho(x, t)$ -which is defined on physical space- into velocity space. The distribution function defines the density of particles at position $\mathbf{x} = (x, y, z)$ and time t with velocity $\mathbf{v} = (v_x, v_y, v_z)$. One can recover the density (which is the zeroth moment of the distribution function) by integrating the distribution function over the velocity space:

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t) d^3v \tag{3.5}$$

The macroscopic velocity $\mathbf{u}$ is defined as the first moment of the distribution function:

$$\rho \mathbf{u} = \int \mathbf{v} f d^3 v \tag{3.6}$$

Where $\mathbf{u}$ is the macroscopic velocity and $\mathbf{v}$ the microscopic velocity.

Given the large number of collisions, and the sensitivity of scattering events between particles to the initial conditions, it is assumed that these particles can be described by an isotropic distribution in velocity space around the mean velocity $\mathbf{u}$. Consequently, the time evolution of the distribution function is going to be towards an equilibrium distribution $f_{eq}(\mathbf{x}, \mathbf{v}, t)$.

By taking mass, momentum and energy conservation into account, one can derive the form of the equilibrium distribution function [121]:

$$f_{eq}(\mathbf{x}, \mathbf{v}, t) = \rho(\mathbf{x}, t) \left(\frac{1}{2\pi RT} \right)^{3/2} \exp\left(-\frac{(\mathbf{v} - \mathbf{u})^2}{2RT} \right) \tag{3.7}$$

To discretize the equilibrium distribution function for numerical use, we expand f_{eq} in a taylor series with respect to $\mathbf{u}$ up to second order in u:

$$f_{eq} = \frac{\rho}{(2\pi RT)^{3/2}} e^{-\frac{\mathbf{v}\cdot\mathbf{v}}{2RT}} \left[1 - \frac{-2\mathbf{v}\cdot\mathbf{u} + \mathbf{u}\cdot\mathbf{u}}{2RT} + \frac{(\mathbf{c}\cdot\mathbf{u})^2}{2(RT)^2} \right] \tag{3.8}$$

It can be shown through Chapman-Enskog expansion that this form of the equilibrium distribution function is sufficient to recover the Navier-Stokes equations in the macroscopic limit [122]. It is therefore computationally efficient to neglect any higher order terms in the expansion.

We make the following identifications: $RT = c_s^2$, $\left(\frac{1}{2\pi RT} \right)^{3/2} e^{-\frac{\mathbf{v}\cdot\mathbf{v}}{2RT}} = W(\mathbf{v})$, where c_s is the system speed of sound and $W(\mathbf{v})$ a velocity dependent weight function. For a given lattice scheme, we therefore have this form for the equilibrium distribution function along a discrete direction i:

$$f_i^{eq} = \rho W(\mathbf{v}) \left[1 + \frac{2\mathbf{c}\cdot\mathbf{u} - \mathbf{u}\cdot\mathbf{u}}{2c_s^2} + \frac{(\mathbf{c}\cdot\mathbf{u})^2}{2c_s^4} \right] \tag{3.9}$$

Furthermore, applying the total derivative of the distribution function $f(\mathbf{x}, \mathbf{v}, t)$ with respect to time, one can derive the Boltzmann equation using the the chain rule:

$$\frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial \mathbf{v}}\frac{d\mathbf{v}}{dt} = \mathbf{v} \cdot \nabla f + \mathbf{a} \cdot \frac{\partial f}{\partial \mathbf{v}} \tag{3.10}$$

We therefore obtain:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \mathbf{v}} = \Omega(f) \tag{3.11}$$

This can be interpreted as an advection equation for the quantity f along a velocity field $\mathbf{v}$, with a source term $\Omega(f)$ that accounts for collisions between particles. The source term is usually expressed as a double integral over the velocity space and consider all possible outcomes of a collision between two particles, which is fairly complicated to solve. Refer to [123] for a more detailed derivation of the Boltzmann equation and the collision term. For the purposes of this work, we can move directly to a simplification of the collision operator that turns out to have practical applications, especially for LBM, namely the Bhatnagar-Gross-Krook (BGK) approximation [124]:

$$\Omega(f) = -\frac{1}{\tau}(f - f_{eq}) \tag{3.12}$$

This approximation is based on replacing the complex collision term with a relaxation of the distribution functions towards equilibrium $f_{eq}(\mathbf{x}, \mathbf{v}, t)$ with a relaxation time τ. In fact, taking $f \simeq f_{eq}$ in the Boltzmann equation yields the Euler equations, which are simplifications or limiting cases of the Navier-Stokes equations in the absence viscosity. Importantly, this reveals that the phenomena associated with fluid viscosity are encapsulated within the non-equilibrium part of the distribution function $f - f_{eq}$. Hence, the BGK approximation is a way to model viscous flows in a way that is computationally efficient. The BGK operator is also sometimes referred to as the Single Relaxation Time operator, mainly in contrast to other Multiple Relaxation schemes, that comprise a multitude of relaxation times instead. The use of multiple relaxation times is to mitigate the BGK operator's issue of viscosity dependent accuracy [107] as well as offering an increased number of degrees of freedom that one can use to tune and stabilize the simulations.

The Boltzmann Equation is discretized both in velocity as well as physical space in order to be used in numerical simulations. The time and space discretization are fairly straightforward and analogous to other numerical methods, where time t is divided

into discrete steps Δt and x into a lattice spacing Δx. The 'lattice' in LBM has more structure than these two discretization steps, and this structure arises from discretizing the velocity space into a set of velocities $\mathbf{c}_i$, each with a corresponding weighting factor w_i. Examples of discrete velocity sets are shown in figure 3.1. The convention is that each set is named with a $DnQm$ scheme where n refers to the lattice dimensions, and m the number of directions in the set.

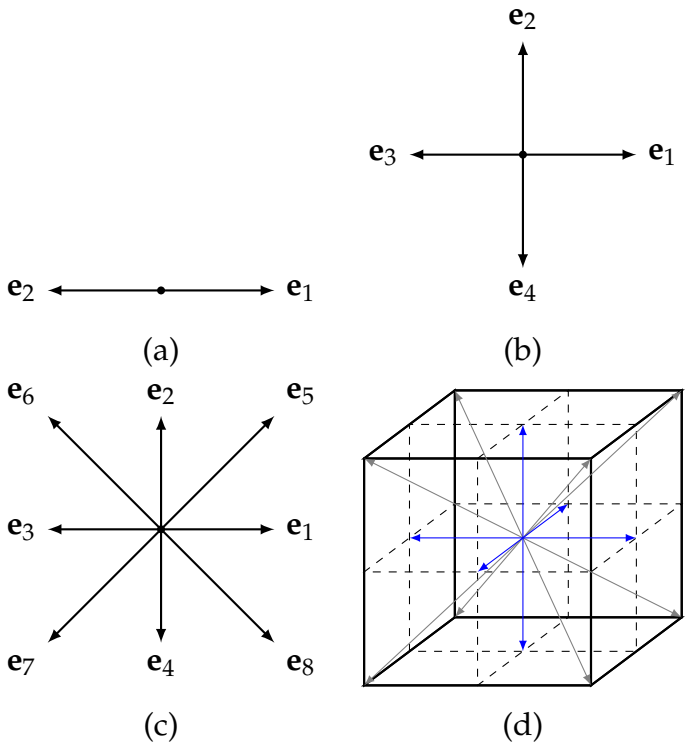


Figure 3.1: LBM Lattices in one, two, or three dimension: (a) D1Q3, (b) D2Q5, (c) D2Q9 and (d) D3Q15 (*Note: rest velocities not shown*). Refer to [125] for a more comprehensive list of lattice velocity sets

Numerous lattice velocity sets exist, as they can be constructed using a fairly straightforward set of rules relating to isotropy (equations 3.19, 3.20, and 3.21). However, increasing the number of directions in a set will increase the computational cost of the simulations later on, as more populations will need to be tracked and calculated. The choice of a velocity set should therefore be informed by the isotropy requirements of the problem at hand, as well as the computational efficiency. For instance, the $D2Q9$ set shown in figure 3.1.c is by far the most widely used velocity set because it's simultaneously isotropic enough to resolve the Navier-Stokes flows and computationally efficient

[107, 108]. If, however, one needs to resolve the heat equation in two dimensions for example, which is a diffusion equation, then a set such as D2Q5 is sufficient. In fact, the isotropy requirements for each problem are informed by the multiscale Chapman-Enskog analysis and include different moments of the weighting factor w_i. For the Navier-Stokes equation, isotropy is required for all moments up to fifth order [126]:

$$\sum_i w_i = 1 \tag{3.13}$$

$$\sum_i w_i c_{i\alpha} = 0 \tag{3.14}$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} = c_s^2 \delta_{\alpha\beta} \tag{3.15}$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} = 0 \tag{3.16}$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\mu} = c_s^4 (\delta_{\alpha\beta}\delta_{\gamma\mu} + \delta_{\alpha\gamma}\delta_{\beta\mu} + \delta_{\alpha\mu}\delta_{\beta\gamma}) \tag{3.17}$$

$$\sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\mu} c_{i\nu} = 0 \tag{3.18}$$

$D2Q9$ is the smallest velocity set that satisfies these constraints up to fifth order in two dimensions, which is why it is very popular in literature for simulating incompressible flows. However, there are additional constraints that apply on these velocity sets for isotropy, mass, energy and momentum conservation to be satisfied, and which all velocity sets have to verify. Given a speed of sound c_s, the constraints are the following:

$$\sum_i c_i = 0 \tag{3.19}$$

$$\sum_i c_{\alpha i} c_{\beta i} = c_s^2 \delta_{\alpha\beta} \tag{3.20}$$

$$\sum_i w_i c_{\alpha i} = 0 \tag{3.21}$$

Another constraint that is used for the components of $\mathbf{c}_i$ are integer multiples of $\Delta x/\Delta t$:

$$c_{\alpha i} = n\frac{\Delta x}{\Delta t}$$

Such an assumption guarantees that a mesoscopic population at $\mathbf{x}$ traveling along one of these velocities will end up exactly at a neighboring lattice site $\mathbf{x} + \mathbf{c_i}\Delta t$ at the next time step.

Using the discrete set of velocities, we integrate equation 3.11 with the BGK operator formally along a time step Δt, and by neglecting terms of order $\mathcal{O}(\Delta t^2)$ (refer to [127] for a full derivation), and retrieve the discrete Lattice Boltzmann Equation:

$$f_i(\mathbf{x} + \mathbf{c}_i\Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau}\left(f_i(\mathbf{x}, t) - f_{eq}(\mathbf{x}, t)\right) \quad (3.22)$$

A way to investigate the previously mentioned non-equilibrium part of the distribution function is to expand it in a perturbation series around the equilibrium distribution function f_{eq} in terms of a small parameter ϵ:

$$f = f_{eq}^{(0)} + \epsilon f^{(1)} + \epsilon^2 f^{(2)} + \dots \quad (3.23)$$

ϵ is interpreted as being a parameter that is directly linked to the Knudsen number. The first term leads the Euler equation of momentum, with subsequent higher order terms adding non-equilibrium corrections to the overall system. By substituting this expansion into the discretized Lattice Boltzmann Equation with the BGK operator, this leads to the following equations [107]:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_\alpha)}{\partial x_\alpha} = 0 \quad (3.24)$$

$$\partial_t(\rho u_\alpha) + \partial_\beta(\rho u_\alpha u_\beta) = -\partial_\alpha \rho c_s^2 + \partial_\beta\left[\rho c_s^2\left(\tau - \frac{\Delta t}{2}\right)(\partial_\beta u_\alpha + \partial_\alpha u_\beta)\right] \quad (3.25)$$

Where:

- $\rho c_s^2 = p$ is the pressure
- $\rho c_s^2\left(\tau - \frac{\Delta t}{2}\right) = \eta$ is the shear viscosity

It is therefore clear from equation 3.25 that $\tau - \frac{\Delta t}{2} > 0$ is a necessary stability condition for LBM to avoid negative viscosity.

We have therefore seen that at the macroscopic limit, the Lattice Boltzmann Method recovers the Navier-Stokes equations,

showing that the it is a valid method for simulating fluid flows. As a final note on this section, we recover the macroscopic quantities ρ and $\mathbf{u}$ from the microscopic populations f_i by summing over the discrete velocities, analogously to equation 3.5 and 3.6:

$$\rho = \sum_i f_i \tag{3.26}$$

$$\rho \mathbf{u} = \sum_i \mathbf{c}_i f_i \tag{3.27}$$

Moreover, the link between the relaxation time and the kinematic viscosity of the flow is give by:

$$\nu = c_s^2 \left(\tau - \frac{\Delta t}{2} \right) \tag{3.28}$$

Another collision operator that we use in this work is the Two-Relaxation rate operator (TRT) that was introduced by Ginzburg et al. [128]. With the TRT operator, two relaxation times are used as the name suggests; one for the symmetric part of the distribution function, and another for the antisymmetric part:

$$f_i^+ = \frac{1}{2} \left(f_i + f_{\bar{i}} \right) \tag{3.29}$$

$$f_i^- = \frac{1}{2} \left(f_i - f_{\bar{i}} \right) \tag{3.30}$$

where $\bar{i}$ is the index of the opposite direction of i ($\mathbf{c_i} = -\mathbf{c_{\bar{i}}}$). The original distribution functions can be recovered by summing or subtracting the symmetric and antisymmetric parts:

$$f_i = f_i^+ + f_i^- \tag{3.31}$$

$$f_{\bar{i}} = f_i^+ - f_i^- \tag{3.32}$$

This defines two different relaxation coefficients ω^+ and ω^-, for the symmetric and antisymmetric distribution functions f^+ and f^-. With this operator, the symmetric relaxation coefficient determines the viscosity:

$$\omega_+ = \frac{1}{\tau_+} = \frac{1}{3\nu + 0.5} \tag{3.33}$$

In contrast to the BGK operator where the error is proportional to $(\tau - \frac{\Delta t}{2})^2$, the error with the TRT operator is dependent on a combination of ω^+ and ω^-, which are the relaxation coefficients for the symmetric and antisymmetric parts respectively. The truncation error is characterized with this Λ parameter:

$$\Lambda = \left(\frac{1}{\omega^+ \Delta t} - \frac{1}{2}\right)\left(\frac{1}{\omega^- \Delta t} - \frac{1}{2}\right) \tag{3.34}$$

The LBE with the TRT operator becomes:

$$f_i(\mathbf{x} + \mathbf{c_i}\Delta t, t + \Delta t) = f_i - \omega^+ \Delta t \left(f_i^+ - f_i^{eq+}\right) - \omega^- \Delta t \left(f_i^- - f_i^{eq-}\right) \tag{3.35}$$

For a given viscosity, we can tune the ω^- parameter to minimize the error for a given problem. According to Kuzmin et al. (2011), setting $\Lambda = 1/4$ provides the most stable simulations in general [129]. Moreover, researchers have identified other values of the Λ parameter that cancel specific terms in the spatial error. For instance, $\Lambda = \frac{1}{12}$ cancels out the third-order spatial error, which is optimal for diffusion dominated flows. On the other hand, $\Lambda = \frac{1}{6}$ cancels the fourth-order spatial error, leading to optimal calculations for pure diffusion [107].

The two main algorithmic operations that can be derived from a numerical implementation of the Lattice Boltzmann Equation are the streaming and collision steps. The collision operation is performed as described above using the BGK or TRT operators, while the streaming step consists simply of moving the populations along their corresponding lattice velocities. The two steps are demonstrated in figure 3.2.

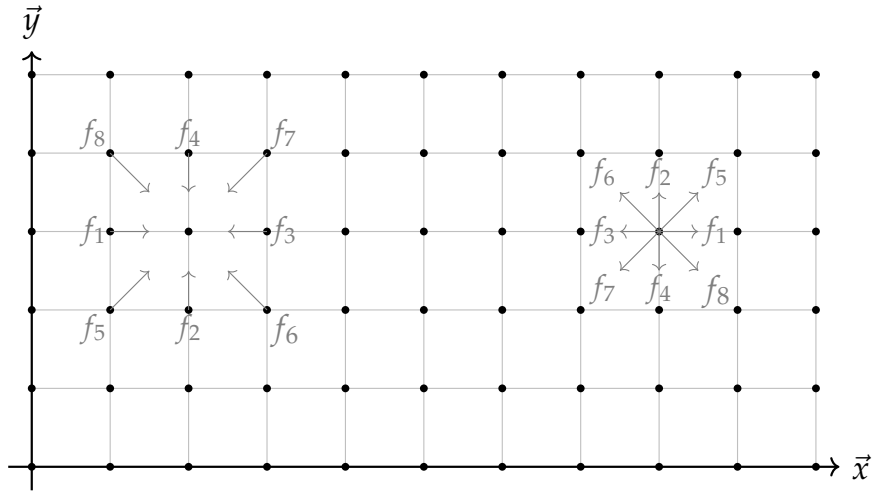


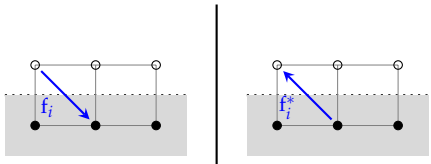
Figure 3.2: State of populations neighboring a central node before streaming (left). State of the same populations after streaming to the central node, and right after collision (right)

3.2.1 *Boundary Conditions*

The treatment of boundary conditions in LBM is also distinctive from other traditional CFD methods. These methods are based of partial differential equations, and the boundary conditions are imposed directly on the relevant quantities (ρ and u). In LBM, the boundary conditions are instead imposed on the populations f_i, which introduces an issue of non-uniqueness. In fact, while the macroscopic quantities can be determined uniquely given a set of populations, there are multiple sets of populations that can give rise to the same macroscopic quantities. Consequently, the problem is under-constrained and one can find many different boundary conditions schemes in literature that give rise to the same macroscopic behavior. In order to limit the scope of this work, we will only present the boundary conditions that were used in this work for the incompressible flow and advection-diffusion problem, which are the bounce-back and the Zou-He conditions.

One of the most common boundary conditions in hydrodynamics is the no-slip condition, which applies to solid walls where the velocity of the fluid is stated to be zero. The most common and straightforward method to implement this condition in LBM is the bounce-back rule. In this method, the populations that are streamed towards a node which is considered to be a solid wall are simply reflected back to the node where they came from. It should be noted that in this case, the

wall is considered to be located roughly halfway between the solid node and the fluid node. This is illustrated in figure 3.3.



(a) t (b) $t + \Delta t$

Figure 3.3: Illustration of the bounce-back rule in the lattice Boltzmann method. The filled circles represent solid nodes, the dotted line represents the wall location, and f_i and f_i^* are the populations before and after bounce-back, respectively

Given that opposing velocities are $c_{\bar{i}} = -c_i$, the formula for the bounce-back rule is as follows:

$$f_{\bar{i}}(\mathbf{x}, t + \Delta t) = f_i^*(\mathbf{x}, t) \tag{3.36}$$

This numerical scheme has a few advantages apart from being relatively straightforward to implement. It has been shown to be stable and the reflection of populations guarantees mass and momentum conservation, which is not always the case with other boundary condition schemes. Nevertheless, the bounce-back rule is not without its drawbacks. The fact that it is a stair-cased representation of the geometry, making it only first-order in acccuracy in cases where the wall orientation is not exactly aligned with the lattice. However, the bounce-back rule remains a popular choice for boundary conditions in LBM due to its simplicity. In this work, we will use the bounce-back rule for all solid walls in the simulations. Solid nodes that have at least one fluid neighbor are considered to be boundary nodes, and are subject to the bounce-back rule.

Another case of boundary conditions that is required is the inlet and outlet conditions. At each new time step, some incoming populations on boundary nodes are unknown and need to be specified. A common required boundary condition is a dirichlet condition on the velocity, which is usually the case for inlets. The Zou-He (also referred to as the non-equilibrium bounceback) method is a popular method to impose this condition [108]. The simplest way to enforce a dirichlet condition is to set the boundary node populations to the equilibrium distribution functions $f_i^{eq}(\rho, u_d)$ where u_d is the specified velocity at the boundary. It

should be noted, however, that it is only second-order accurate in cases where $\tau/\Delta t = 1$ [107]. The Zou-He method, on the other hand, can be third-order accurate as it is based on the idea of using the bounce-back rule, but only on the non-equilibrium part of the populations:

$$f_i(\mathbf{x},t) - f_i^{eq}(\mathbf{x},t) = f_{\bar{i}}(\mathbf{x},t) - f_{\bar{i}}^{eq}(\mathbf{x},t) \tag{3.37}$$

For instance, in the D2Q9 lattice, using this condition along with the well-known formulas for the macroscopic quantities, one can derive the following formulas for the unknown populations at a supposed left inlet boundary:

$$\rho_i = \frac{1}{1-u_{x,i}}\left(\sum_{i=0,2,4} f_i + 2\sum_{i=3,6,7} f_i\right) \tag{3.38}$$

$$f_1 = f_3 + \frac{2}{3}\rho_i u_i \tag{3.39}$$

$$f_5 = f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_i u_{x,i} - \frac{1}{2}\rho_i u_{y,i} \tag{3.40}$$

$$f_8 = f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_i u_{x,i} - \frac{1}{2}\rho_i u_{y,i} \tag{3.41}$$

In the above, $u_{x,i}$ and $u_{y,i}$ are the two components of the velocity at the inlet, and ρ_i is the density. the velocities c_1, c_5, and c_8 are the velocities that are pointing towards the bulk of the geometry, and consequently have a positive x-component.

3.3 LBM FOR ADVECTION-DIFFUSION

In this section, we will show how we can modify LBM to solve the chemical species transport problem, which is governed by the advection-diffusion equation:

$$\frac{\partial c}{\partial t} + \mathbf{u}\cdot\nabla c = \nabla\cdot(D\nabla c) + s \tag{3.42}$$

where:

- c is the concentration of a reactant species.
- $\mathbf{u}$ is the bulk fluid velocity.
- D is the diffusion coefficient of the species.

- s is the source term with $s > 0$ indicating production and $s < 0$ consumption of the species.

In order to use LBM to solve this equation, we can make the observation that the Navier-Stokes equations are in fact a convection diffusion equation for the vector momentum quantity $\rho\mathbf{u}$:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u}\mathbf{u} + p\mathbf{I}) = \nu\nabla^2(\rho\mathbf{u}) + \mathbf{F} \tag{3.43}$$

where $\mathbf{I}$ is the identity matrix and $\nu = \eta/\rho$. Making the following identifications [107]:

$$\rho\mathbf{u} \to c, \quad \rho\mathbf{u}\mathbf{u} + p\mathbf{I} \to c\mathbf{u}, \quad \mathbf{F} \to s, \quad \nu \to D \tag{3.44}$$

As a result of this substitution, it is apparent that the analog of viscosity in the advection-diffusion system is the diffusion coefficient, while c is the only conserved quantity. The velocity vector $\mathbf{u}$ is considered to be an input of this system instead of a solution, and it is usually obtained as the steady-state solution of the Navier-Stokes equations that are solve beforehand using a different Navier-Stokes solver. In fact, LBM itself can be used as this solver as described in earlier sections. The macroscopic quantity c is recovered from the microscopic populations in a similar way to the Navier-Stokes equations:

$$c = \sum_i g_i \tag{3.45}$$

The same equilibrium distribution function from 3.9 can be used for this system. However, due to relaxed constraints, one can use a simplified expression that still recovers advection-diffusion:

$$f_i^{eq} = w_i c \left[1 + \frac{\mathbf{c_i} \cdot \mathbf{u}}{c_s^2}\right] \tag{3.46}$$

Using another Chapman-Enskog procedure for advection-diffusion, the source term s is implemented in LBM using the following scheme:

$$Q_i = \left(1 - \frac{1}{2\tau}\right) w_i s \tag{3.47}$$

$$c = \sum_i f_i + \frac{Q_i \Delta t}{2} \tag{3.48}$$

The source term is therefore implemented as a source population Q_i that is added to the distribution function at each time step during the calculation of the concentration. The source population is then relaxed towards equilibrium with a relaxation time τ that is related to the diffusion coefficient D, analogously to the incompressible flow problem:

$$\tau = c_s^2 \left(\tau - \frac{\Delta t}{2} \right) \tag{3.49}$$

Before concluding this section, it should be noted that the bounce-back rule that was described previously for hydrodynamics problems is used analogously for mass transport as well. In fact, the reflection of populations at the boundary nodes can be used to enforce the no-flux condition at solid walls. Similarly, the Zou-He method can be used to enforce a dirichlet condition for the concentration at the inlets.

Thus we have introduced all the particularities of the LBM extension for advection-diffusion problems. All other parameters that were not included in this section are identical to the Navier-Stokes incompressible flow problem in the preceding section.

3.4 ELECTROCHEMICAL MODEL

The main focus of modelling in electrochemistry is concerned with studying the different types of ion transport phenomena in electrolyte. A general method to approach the problem at the continuum scale is through a Poisson-Nernst-Planck system which is a conservation law for the ion flux $\mathbf{J}$:

$$\frac{\partial c_i}{\partial t} + \nabla \cdot \mathbf{J_i} = 0 \tag{3.50}$$

$$\mathbf{u} c_i - D_i \nabla c_i + \frac{D_i z e}{k_B T} c \mathbf{E} = \mathbf{J} \tag{3.51}$$

where:

- z and e are the valence of the ion and the elementary charge respectively.

- k_B and T are the Boltzmann constant and temperature respectively.
- $\mathbf{E}$ is the electric field.

The flux $\mathbf{J}$ is divided into three components: a convective term $\mathbf{u}c_i$, a diffusive term $-D_i \nabla c_i$ and a migration term $\frac{D_i z e}{k_B T} c\mathbf{E}$. In the dilute solution limit, which is used in this project due to the relatively low reactant concentrations used, the migration term can be neglected due to the dominance of the supporting electrolyte in the flow. If the migration term is considered, one needs to also solve the Poisson equation for the electric potential ϕ in order to calculate the electric field $\mathbf{E} = -\nabla\phi$. The Poisson-Nernst-Planck system can subsequently be simplified to the advection-diffusion system described in the previous section.

Electrochemical reactions occur just outside a thin double layer near the electrode surface [130]. The thickness of this layer is on the nanometer scale [131] and is therefore not resolved in our LBM lattices. If double layer related phenomena do need to be considered, it should be possible to introduce a double layer capacitance to the impedance of the system. Otherwise, it is sufficient to use a mask or indicator function that labels lattice sites according to whether they are part of an electrochemical interface or not. The source terms s in the advection-diffusion system are only non-zero in this region, and depend on a locally determined reaction rate. Away from the interface, the source terms are zero and do not affect the advection-diffusion system.

The reaction rate depends on a number of parameters related to the nature of the electrolyte solution, the nature and the potential of the electrode, etc. In this work, we use a general Butler-Volmer model to describe the reaction kinetics [131]:

$$j = j_0 \left\{ \frac{c_O^s}{c_O^b} \exp\left(\frac{\alpha_a n F}{RT}\eta\right) - \frac{c_R^s}{c_R^b} \exp\left(-\frac{\alpha_c n F}{RT}\eta\right) \right\} \tag{3.52}$$

where:

- η is the overpotential, defined as $\eta = E - E^0$ where E is the electrode potential and E^0 is the thermodynamic standard potential.
- j_0 is the exchange current density.
- n is the number of electrons transferred in the reaction.

- α_a and α_c are the anodic and cathodic transfer coefficients respectively.
- R and T are the universal gas constant and temperature resp.
- c_O^s and c_R^s are the surface concentrations of the oxidized and reduced species respectively.
- c_O^b and c_R^b are the bulk concentrations of the oxidized and reduced species respectively.

The equilibrium potential is defined as the potential at which the overall reaction rate is zero which corresponds to the case where the anodic and cathodic currents cancel out. It is given by the Nernst equation:

$$E_{eq} = E^0 - \frac{RT}{nF} \ln \left[\frac{c_O^s}{c_R^s} \right] \tag{3.53}$$

The principles of mass and charge conservation impose that the source term for the consumed species has to be the additive inverse of the source term of the produced species: $s_r = -s_p$

3.5 COUPLING NAVIER-STOKES, ADVECTION-DIFFUSION AND ELECTROCHEMISTRY

The coupling of the Navier-Stokes equations, electrochemical reactions and the advection-diffusion system is shown in figure 3.4. We proceed in the following manner:

We set up the lattice that will solve for bulk fluid flow (NS Lattice in figure 3.4.b). In our case it's a D2Q9 lattice and we iterate it until we reach a maximum iteration number t_{max} or the variation of the velocity field falls below a certain tolerance parameter ϵ_c. The velocity field that results from this process is taken as our steady-state velocity field, and is used as an input for the advection-diffusion system (transport lattice in 3.4.a). After every streaming step in the transport lattice, we calculate the electrochemical quantities η and j that are used in the source terms which in turn are calculated through equation 3.48. In fact, the source terms can either be accounted for in the concentration calculation 3.48 or in the collision step [108].

The transport lattice is also iterated until a maximum iteration number, which varies depending on the specific simulation. In

the case of the Navier-Stokes solver, iteration is stopped when the variation of the velocity field falls below a certain tolerance parameter ϵ_c. In the case of the transport problem, the total number of iterations N_{it} is selected given a specific physical time t_{ph} is chosen for the simulation using the time step Δt:

$$N_{it} = \frac{t_{ph}}{\Delta t} \tag{3.54}$$

$$\Delta t = c_s^2 \left(\tau - \frac{1}{2}\right) \frac{\Delta x^2}{D} \tag{3.55}$$

The concentration field is periodically saved to disk for post-processing using an iteration interval. Selecting a lower interval will result in more fine-grained data in time being saved to disk, but tends to slow down the simulation despite the fact that this operation is performed on a parallel thread. It is therefore important to select a good compromise that allows for sufficiently resolved output data in time, while also keeping disk usage and computational time at reasonable levels.

3.6 SUMMARY

In this chapter we described the theoretical underpinnings of the model that we use in this work, and whose results will be shown in the next chapter. We first set our initial objective to be the solution of the Navier-Stokes equations. To achieve that numerically, we described how the Lattice Boltzmann Method relies on principles of kinetic theory, especially the concept of distribution functions, to recover Navier-Stokes macroscopically. These distribution functions are themselves governed by an advection equation with a right hand side collision term. Then the BGK operator is introduced as an approximation of the collision process through a relaxation towards equilibrium with a single relaxation parameter. We briefly described a similar two relaxation time parameter due to its use in the transport problem. Then a discretization step is performed on the physical and phase spaces of the system, wherein time and space are discretized along with the velocity space using specific velocity sets that obey certain isotropy relations. Thus, we obtain the discrete Lattice Boltzmann Equation, which when expanded in a perturbative series around the equilibrium distribution function, recovers the Navier-Stokes equations macroscopically.

The second part of this chapter shows how by considering the Navier-Stokes equations as advection-diffusion of the vec-

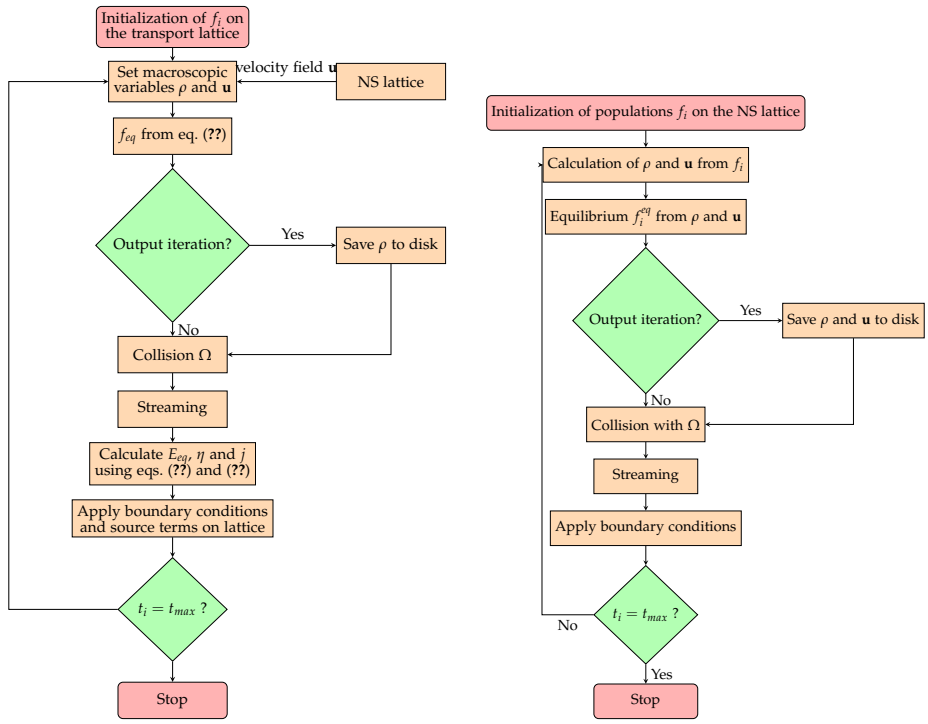


Figure 3.4: (a) Overview of the numerical model that integrates fluid flow, species transport and electrokinetics. (b) Breakdown of the loop that constitutes the Navier-Stokes solver. It is executed until a predefined maximum simulation time t_{max} is reached, which is chosen such that the flow is at a steady-state.

tor momentum quantity, we can analogously derive a Lattice Boltzmann extension that solves for the transport of chemical species. We then show how electrochemical reactions are linked to source terms in these transport equations in order to include reaction kinetics. Finally, we show how the Navier-Stokes, advection-diffusion and electrochemical systems are coupled together in a single numerical model that is used to solve the entire simulation of an electrochemical system at the device scale .

4

CHAPTER 4: IMPLEMENTATION AND RESULTS

This chapter dives into the implementation details of the Lattice Boltzmann Method for fluid flow and its extension for mass transport based on the theoretical framework that was laid down in the previous chapter. The amenability of LBM for complex geometries is taken advantage of to represent complex microporous geometries that are typically encountered in flow through MMRFBs and can also be extended to other electrochemical energy conversion devices where surface area is strongly linked to performance and power output. However, it is not obvious how one can obtain the complex porous geometry of an electrode for use with LBM. In this work, we use a stochastic method to synthetically generate two dimensional porous geometries with a controlled porosity and tortuosity.

After validating the model with well known problems with analytical solutions, we use the LBM model to study the effect of porosity on the performance of a porous electrode in electrochemical devices at the millimeter scale. A section about mass transport limitations in a MMRFB is also shown for a given porosity and tortuosity.

4.1 GENERAL REMARKS ABOUT IMPLEMENTATION

Throughout this project, the implementation of the code was done with pragmatic considerations. The objective is to simultaneously model a complex electrochemical system with reasonable computation time, so as to be able to control various parameters of the system in the duration of a thesis project. While it is usually more optimal to use a compiled language for higher performance, the code was written in Python for rapid development and familiarity. As we mentioned in previous chapters, the Lattice Boltzmann Method is amenable to parallel computing, but parallelizing the code on the CPU is not optimal due to the limited number of cores available in most computers (A modern high-end consumer CPU has at most 24 cores). Modern GPUs comprise thousands of cores and are therefore more suitable for this task. However, due to the fact that GPU programming with languages such as CUDA [132] or OpenCL [133] offers more

fine grained control over the hardware, it's necessary to perform memory management tasks and thread synchronization that can render the code development process more complex and prone to bugs and issues than CPU code. For this reason, instead of developing the code in raw CUDA or OpenCL, throughout this project we made the compromise of using taking advantage of Python's accessibility, debugging tools and rapid development, but pair it with Pytorch's [134] CUDA backend for GPU acceleration. Pytorch is a machine learning framework that offers many scientific computing tools for tensor manipulation in the Python programming language [135]. Due to its wide use in the machine learning field, it is well supported and offers a CUDA backend that performs the tensor operations directly on the GPU. This allows us to abstract away the memory management, thread synchronization aspects of GPU programming and focus on the algorithmic implementation of the numerical model while still benefiting from the performance boost of GPU acceleration. Other performance optimizations that are applicable to python code are used, when possible, such as avoiding explicit loops and using vectorized operations instead.

4.2 VALIDATION WITH ANALYTICAL SOLUTIONS

In this section, and before we proceed to study of porous complex geometries, we validate our implementation of the model that was described in the previous chapter. Validation of Navier-Stokes equations for different collision operators and lattice sizes was already performed in literature [107, 108, 136, 137]. Consequently, we will focus on the validation of the coupled Navier-Stokes and mass transport systems. The first test case is the calculation of the Cottrell current for a planar electrode geometry with pure diffusion. The second test case is the calculation of a limiting current at a planar electrode in a channel with Poiseuille flow, where convection and diffusion are both present.

4.2.1 *Cottrell Experiment*

The Cottrell Experiment is a well known chronoamperometry experiment where a potential step signal is imposed at a planar electrode in an electrolyte solution. This potential is taken such that the reactant species kinetics for reduction are considered to be fast and the concentration at the surface drops to zero.

This corresponds to the case where there is no kinetic limitation, and only the mass transport of the reactant from the bulk of the solution to the electrode surface limits the current. The current across the electrode is flowing to reduce all the reactant species near the surface where:

$$Ox + e^- \rightarrow Red \tag{4.1}$$

The consumption of the reactant creates a concentration gradient that leads to a diffusion flux towards the electrode surface. The thickness of the diffusion layer that emerges continues to grow, reducing the slope of the concentration at the electrode, and consequently the resulting current as well. The current is therefore a function of time in this diffusion-limited regime. Using Fick's second law of diffusion, one can devise the formula for the time evolution of the current [138]:

$$i(t) = \frac{FA\sqrt{D_{Ox}}C_{Ox}}{\sqrt{\pi t}} \tag{4.2}$$

where:

- F is the Faraday constant
- A is the electrode surface area
- D_{Ox} is the diffusion coefficient of the reactant
- C_{Ox} is the bulk concentration of the reactant

The current is therefore inversely proportional to the square root of time and the tends towards zero as $t \rightarrow \infty$. This is known as the *Cottrell current* and is a good description of fast electrochemical processes at the electrode surface. Interestingly, in the case of very slow processes (or processes that occur in a long time scale), this formula is often experimentally invalid as it does not take into account natural convection in the solution due to density gradients in the solution [139, 140] which limit the thickness of the diffusion layer.

The following is a diagram of the geometry of used for the Lattice Boltzmann simulation:

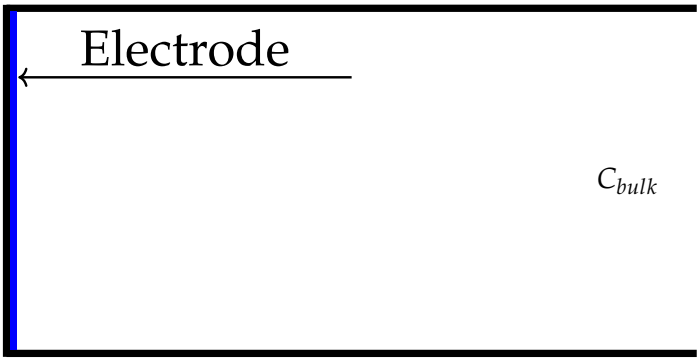


Figure 4.1: Geometry of the Cottrell experiment. The electrode is shown in blue, while the white region represents the bulk electrolyte solution and black represents the walls of the cell.

Since diffusion occurs purely along the horizontal axis x, we can plot the distribution of the concentration profile where the electrode is located at $x = 0$ as:

$$C(x,t) = C_{bulk}\ \text{erf}\left[\frac{x}{2\sqrt{(D_O t)}}\right] \tag{4.3}$$

In our model, the potential imposed at the electrode is sufficiently large to effectively consume almost all available reactants in the nodes immediately next to the electrode surface. A comparison between the lattice boltzmann solution and the analytical solution for the current in a cell with a reactant species with a diffusion coefficient of $D_{Ox} = 7.6 \times 10^{-9}\ m^2/s$ is shown in figure 4.2. The physical concentration used in this case is $C_{bulk} = 0.1\ M$, however, as diffusion is the only rate determining step, this solution can be scaled in a straightforward manner for other concentrations or diffusion coefficients with the law of similarity. Similarly, figure 4.3 delineates the analytical solutions and the lattice Boltzmann method (LBM) computed concentration profiles as functions of the spatial distance from the electrode at various temporal instances.

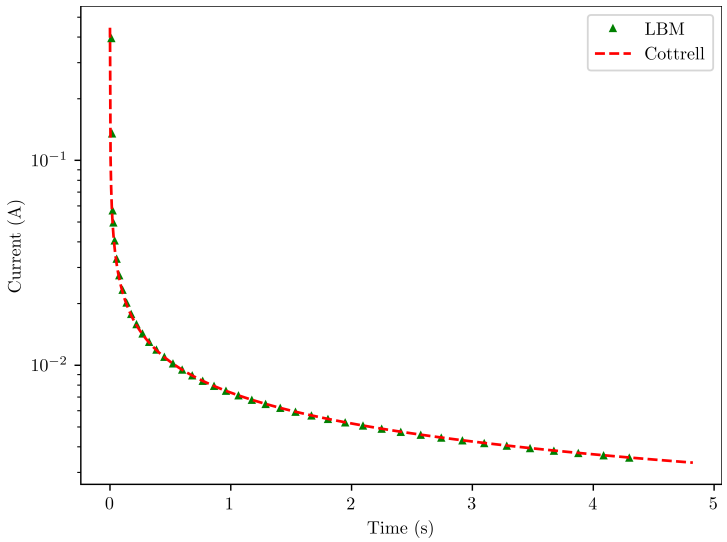
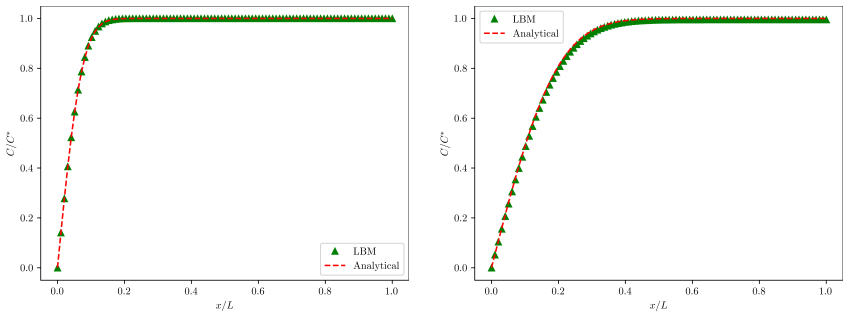


Figure 4.2: Comparison between the analytical solution for the Cottrell current and the Lattice Boltzmann model



(a) $t = 0.54s$ (b) $t = 3.83s$

Figure 4.3: Concentration profiles at different times for the Cottrell experiment.

We can therefore clearly see that the Lattice Boltzmann model is able to reproduce the analytical solution for the Cottrell current. As shown in the previous chapter, our model uses source terms to represent reactions in the system. However, despite the fact that the analytical solution for the Cottrell current is based on the concentration gradient $\frac{\partial C_O}{\partial x}\Big|_{Electrode}$ and not source terms in the diffusion equation, they are mathematically equivalent and provide the same solution.

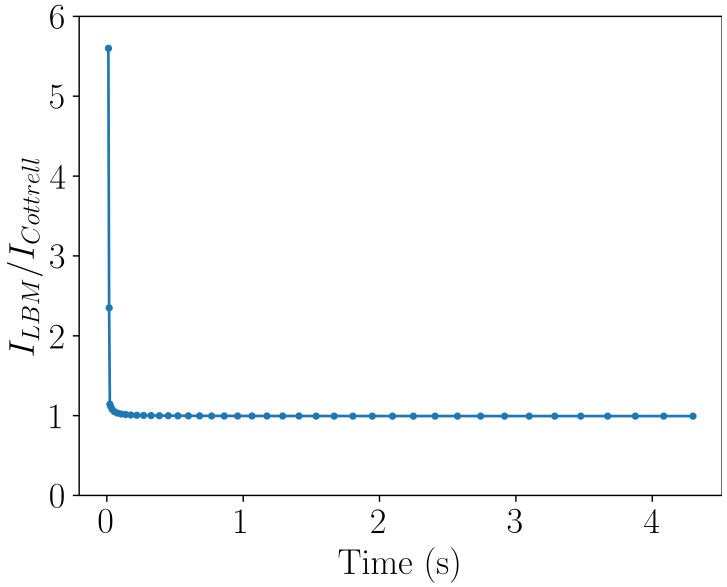


Figure 4.4: Ratio of the Lattice Boltzmann calculated current over the analytical solution $I_{\text{LBM}}/I_{\text{Cottrell}}$ as a function of time

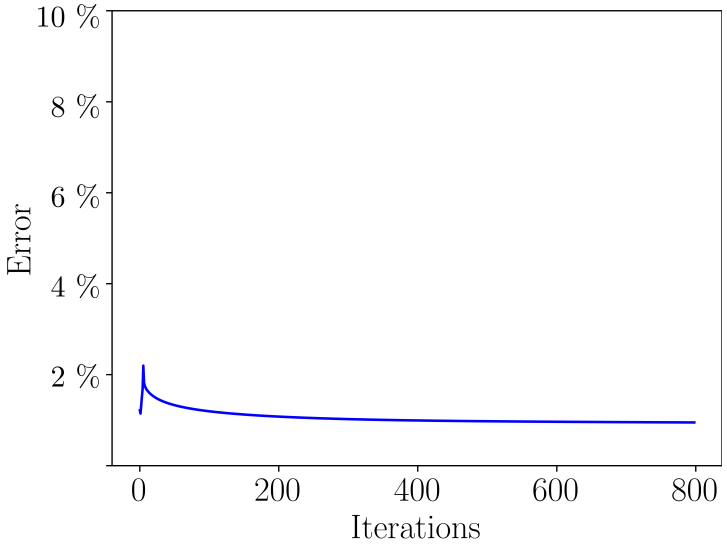


Figure 4.5: Evolution of the error between the Lattice Boltzmann calculated concentration field and the analytical solution given by equation 4.3

Figure 4.4, we analyze the ratio of the current computed via Lattice Boltzmann methods I_{LBM} to that obtained from the analytical Cottrell equation I_{Cottrell}, plotted as a function of time. This ratio serves as an indicator of the discrepancy between the numerical simulation and the analytical model, with a value of 1 representing a perfect match. The plot reveals a marked deviation from unity at the initial time steps, which is attributed to

the lattice discretization errors inherent in the Lattice Boltzmann method. Similarly, the error of the LBM calculated concentration field, which is measured as the *Root Mean Squared Error* is shown in figure 4.5 and is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\left(c_{i,LBM} - c_{i,\text{analytical}}\right)} \tag{4.4}$$

These errors are pronounced at the outset due to the coarse-grained approximation of the continuous space by a discrete lattice. However, the ratio quickly converges towards the analytical solution as the simulation progresses, with the ratio approaching unity within a few iterations for the current. The RMSE of the concentration field also follows a similar evolution with the error decreasing and stabilizing at around 1% after a few iterations. It should be noted however, that for high diffusion coefficients or simply long simulation times, the error will start to rise as the semi-infinite approximation for diffusion in equation 4.3 no longer holds due to the finite size of the simulation domain used and the dirichlet boundary condition that is imposed at the right edge.

Solving this problem for different diffusion coefficients gives scaled but identical solutions to the ones shown above. Since the only determining factor of the time scale of the problem is the diffusion coefficient, varying it only scales the time axis and is therefore equivalent to the above solution in terms of accuracy. The next step is to validate the model for a slightly more involved problem where convection and diffusion are both present.

4.2.2 *The Graetz problem*

In this section we discuss the solution of the Graetz problem using our LBM based model. Originally devised for a problem about heat transfer, the Graetz problem is also a well known problem in fluid mechanics [131, 141]. It's geometry is shown in figure 4.6 and consists of a flow inside a channel with a Poiseuille-type velocity profile:

$$v_x = 2\langle v_x \rangle \left(1 - \left(\frac{y}{H}\right)^2\right) \tag{4.5}$$

where y is the vertical coordinate, H is the height of the channel and $\langle v_x \rangle$ is the average velocity of the flow along the x axis.

The flow is assumed to be fully developed and at temperature T_{in} and encounters a channel wall with a constant temperature T_{wall} Assuming the diffusion coefficient is small, the temperature profile of the fluid domain is constant except for a thin boundary layer near the wall. The Leveque approximation therefore consists of assuming that the variation of the fluid velocity profile near the wall is linear [141]. In fact, the parabolic Poiseuille velocity profile is approximated by a linear profile near the wall.

For our purposes which deal with electrochemical applications, we will use the equivalent mass transport formulation of the problem. Instead of an imposed temperature, we consider an inlet reactant concentration C_{in} and a concentration of zero at the electrode surface which replaces the wall in the heat transfer version of the problem.

It should also be noted that this problem has been treated in previous works in literature. Namely, the work of Tsuru et al. [142], who studied the anodic dissolution of iron using a channel flow double electrode as a substitute for a rotating disk electrode, in a geometry similar to the Graetz problem. Similarly, Amatore et al. [143], numerically analyzed an identical geometry of a microelectrode in a microchannel with laminar flow using commercial finite element based software.

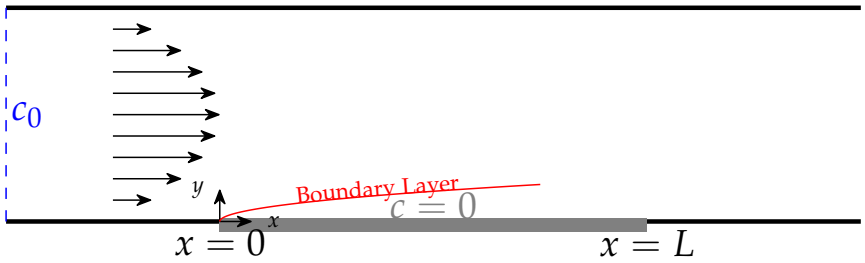
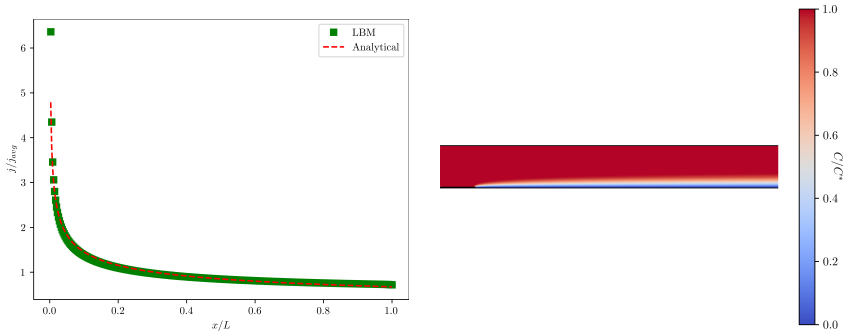


Figure 4.6: Geometry of the Graetz problem. The red line represents the boundary layer where the velocity profile is linear.

Given that the concentration imposed at the electrode surface is zero, similarly to the Cottrell experiment, the current is diffusion limited but varies with the position x at the electrode surface due to reactant depletion downstream of the flow. The current density is therefore a function of the position and is analytically approximated in steady state as [144]:

$$j(x) = 0.9783 \left(\frac{nFDC^*}{A} \right) \left(\frac{\langle v \rangle}{hDx} \right)^{1/3} \tag{4.6}$$

where:



(a) Current density profile (b) Concentration profile

Figure 4.7: Solution of the Graetz problem with LBM and comparison with the Leveque approximation

- C^* is the bulk concentration.
- A the surface area of the electrode.
- x varies between 0 and L where L is the length of the electrode.

In figure 4.7 we compare the analytical solution achieved by the Leveque approximation with the LBM calculated current density for a channel with an aspect ratio of 400 / 50 = 8. The channel height is 6 mm and an electrode length of 4.4 cm. The bulk concentration is 1 M and the diffusion coefficient is 7.6×10^{-7} m^2/s. The peak inlet velocity is 1 cm/s.

Note that despite the Leveque solution containing a singularity at the leading edge of the electrode, the numerical LBM solution is in good agreement with the analytical formula beyond the singular point at $x = 0$. The relative error between the LBM calculated current density and the analytical solution is shown in figure 4.8. We can see that the error is large at the leading edge of the electrode where $x = 0$ due to the discretization not matching the analytical solution which diverges to infinity at that point. The error is calculated as:

$$\text{Error} = \frac{j_{LBM} - j_{\text{analytical}}}{j_{\text{analytical}}} \tag{4.7}$$

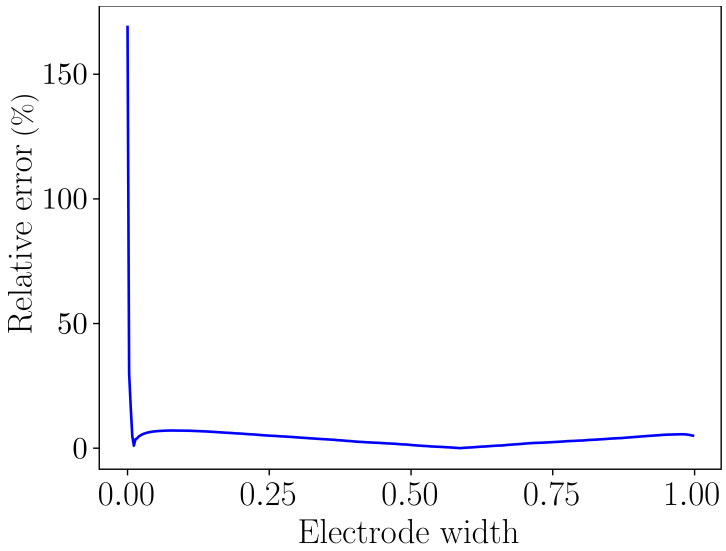


Figure 4.8: Relative error between the LBM calculated current density and the analytical solution

It should be noted that in terms of the Lattice Boltzmann implementation, due to the high Peclet number imposed by the low diffusion coefficient of the reactant (In this case we chose representative values of the diffusion coefficient from [138, 145]), the time scale is determined by convective transport. Thus, in order to keep the lattice velocity below the value of 0.1 (Lattice velocity u should be significantly smaller than the lattice speed of sound c_s to remain in the incompressible limit), one needs to set a relaxation time of the BGK operator that is close to the critical value of $1/2$. This leads to simulations that are prone to numerical instabilities and a small time step dt. We therefore make use of the TRT operator in this case, with a Λ parameter of $1/12$ for optimal results in advection dominated flows such as the Graetz problem.

4.3 SYNTHETIC POROUS GEOMETRIES

After validating our model using problems with known analytical solutions, we now proceed to study the effect of porosity on the performance of porous electrodes that are commonly used in MMRFBs. However, in order to do so, a method to generate and represent complex porous geometries in a way that is compatible with the Lattice Boltzmann Method is needed. Ideally, in order to study the effect of porosity, one would need to be able to generate porous geometries with a controlled porosity and tortuosity. One of the methods that we find in litterature

for modelling of batteries is to use pseudorandom or regular packed assemblies of spheres [146]. However, in order to have more fine grained control over the the pore structure, we use the following stochastic scheme:

We first generate a matrix of random numbers r_{ij} from a gaussian distribution of G mean $\mu = 0$ and variance $\sigma^2 = 1$ (see Figure 4.9). Then we apply a Gaussian Filtering (Figure 4.10.a) to generate a smooth matrix of elements $s(i,j)$ by convolving the original matrix with a discretized Gaussian kernel using the following formula:

$$s(i,j) = (r * G)(i,j) = \sum_{u=-\frac{m}{2}}^{\frac{m}{2}} \sum_{v=-\frac{n}{2}}^{\frac{n}{2}} s(i-u, j-v) G(u,v) \quad (4.8)$$

In the next step, we select a cutoff value c based on the target porosity ϵ using the probit function which is the percent point function of the normal distribution:

$$c = \text{probit}(\epsilon), \quad \epsilon \in [0,1] \quad (4.9)$$

Finally, we use the cutoff value c to generate a boolean matrix where 1 represents boundaries and 0 represents pore space as shown in Figure 4.10.b. In this figure, a 2D representation of the porous microstructure is shown. The black elements represent the solid electrode and the white domain represents the fluid electrolute/pore space. Since the microstructure corresponds to a single electrode, we assume that all the black elements are electrically connected.

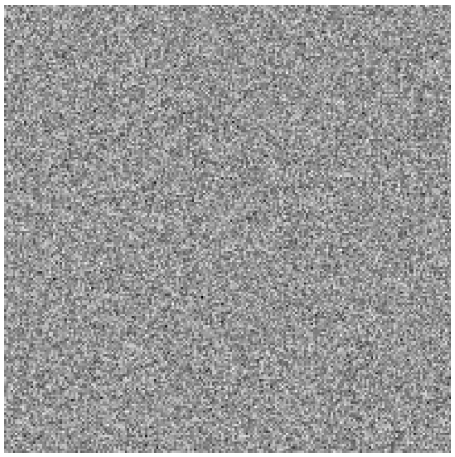
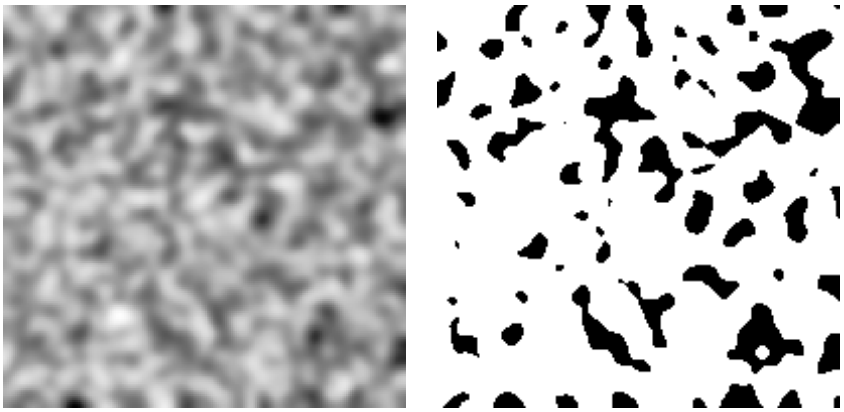


Figure 4.9: Example of a 200×200 matrix of random numbers sampled from a normal gaussian distribution



(a) Gaussian smoothing (b) Thresholding

Figure 4.10: (a) Diagram of the resulting matrix after Gaussian smoothing and (b) boolean matrix after cutoff

Figure 4.10.a shows the result of convolving the original matrix r_{ij} with the gaussian kernel. While in image processing this is performed for purposes as a lowpass spatial frequency filter, in our case it is used to separate contiguous regions of neighboring values that will, in the next step, represent solid elements or pore space. This structure is subsequently inserted into the overall simulation geometry will be shown in later sections. It should also be noted that we use another boolean matrix in order to indicate the interface between the solid and fluid nodes. Any node that is adjacent to a solid node and is not in the bulk of the electrode is considered an active surface area node. This interface corresponds to the location where electron transfer occurs and therefore the nodes where the Butler-Volmer relationship is applied with source terms in the Convection-Diffusion equation. As previously mentioned, the double layer capacitance is not taken into account in these simulations. For easier of the sketch, we use in this work blue as a color-code for these nodes as is shown in Figure 4.11.

The threshold or cutoff parameter defined by equation 4.9 is used to select the desired porosity of our geometry. However, a free parameter of the pore generation scheme is the standard deviation of the smoothing kernel. Indeed, setting large values for this parameter allows for larger features in the pore structure, and vice versa. It can therefore be used as a tool to change the pore size and tortuosity of the resulting structure, which affects the extent to which the pore structure opposes convection-driven transport phenomena [147].

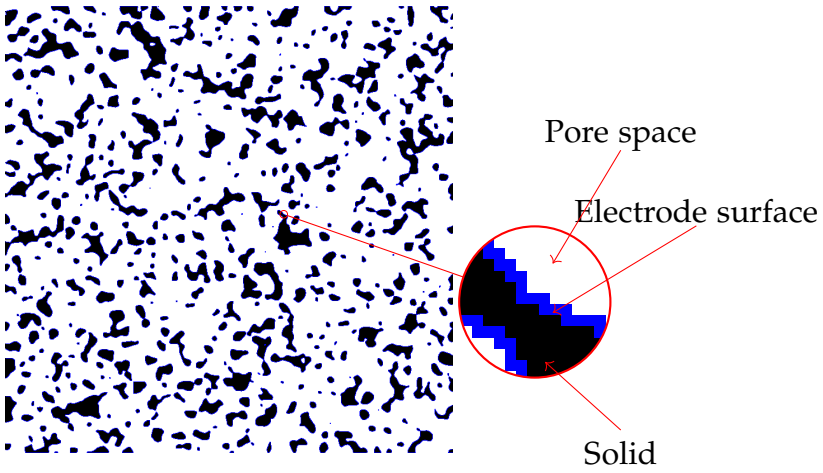
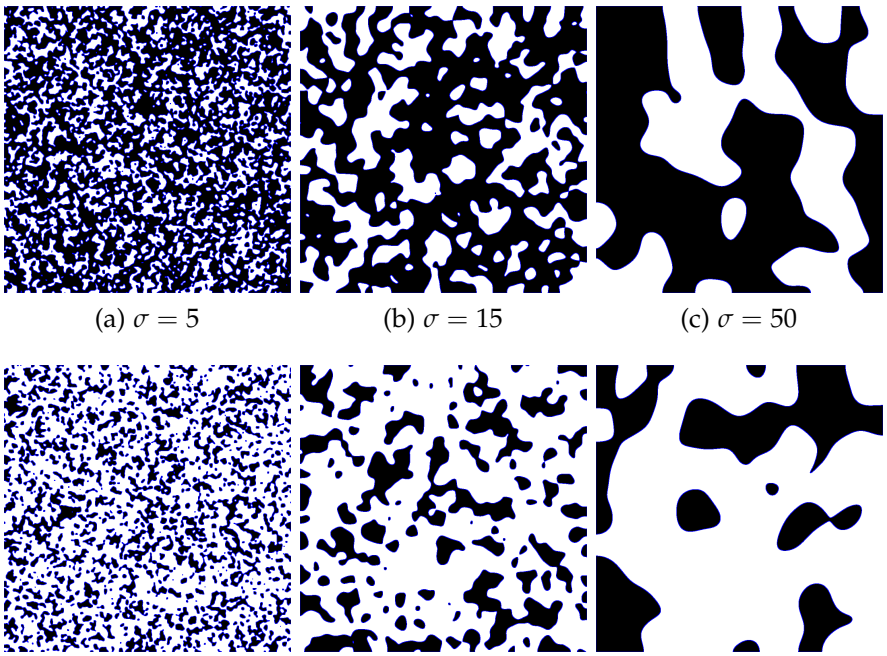


Figure 4.11: Generated porous geometry with a blue color code for electrode surface

If the porosity parameter is fixed, varying the kernel gaussian standard deviation can also affect the surface area between the walls and pore space, as is shown in Figure 4.12. The standard deviation parameter is related to the feature size in the microstructure: A larger value of σ increases the average size of the solid elements and vice versa.



(a) $\sigma = 5$ (b) $\sigma = 15$ (c) $\sigma = 50$

(d) $\sigma = 5$ (e) $\sigma = 15$ (f) $\sigma = 50$

Figure 4.12: Generated sample structures for a fixed porosity of $\epsilon = 0.4$ in (a), (b) and (c) and porosity of $\epsilon = 0.7$ in (d), (e) and (f)

If we approximate the electrode features with discs or 2-spheres with radius r in this 2D geometry, the surface-to-volume ratio is proportional to r^{-1}. Consequently, for an equal solid volume fraction, having a large number of smaller 2-spheres provides more surface area than a small number of large spheres, thanks to their high surface-to-volume ratio. As a general rule, we can say that that the number of pores in the structure decreases as their size increases, and that finer features allow for larger surface area.

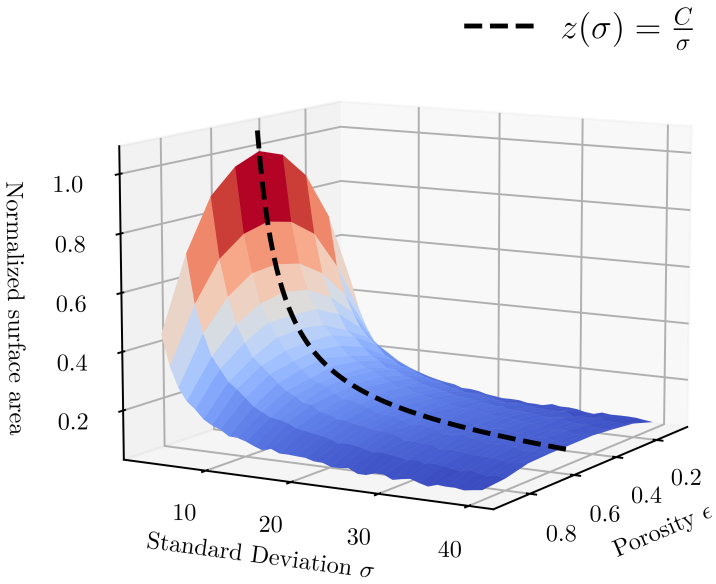


Figure 4.13: Influence of the smoothing kernel standard deviation σ and the porosity ϵ on the surface area. The surface area $S*$ is the sum total of all interface nodes and is normalized by the maximum value obtained in this parameter space. Surface area is proportional to σ^{-1} up to a constant C as shown by the dashed line

Figure 4.13 shows the surface area between pores and walls as a function of both the kernel σ and the porosity ϵ. The surface area, which is the set of interface sites between solid elements (black) and the pore space (white), is measured for each generated microstructure given a porosity $0.1 \leq \epsilon \leq 0.9$ and standard deviation $5 \leq \sigma \leq 40$. We observe that increasing the smoothing gaussian standard deviation decreases the normalized surface area available in a given generated pore structure, and that the surface area is proportional to σ^{-1}. Which further confirms that it is indeed directly linked to the feature size in the generated structures.

4.3.1 *Tortuosity*

The tortuosity is defined in our case as the ratio between the mean length of the hydraulic paths through the geometry and the length of a straight line between inlet and outlet and along the direction of flow:

$$\theta = \frac{\langle \lambda \rangle}{L_x} \tag{4.10}$$

In order to measure the tortuosity of each generated structure, we use the velocity field obtained from the solution of the Lattice Boltzmann Method. We then integrate the velocity field in a cross-section perpendicular to the direction of flow to essentially obtain the streamlines of the flow, and the flux-weighted average length is taken as $\langle \lambda \rangle$. L_x is the length of the domain in the direction of flow.

The flux weighted average is obtained through Matyka et al.'s formula [148]:

$$\theta = \frac{1}{L_x} \frac{\int_A \lambda(\mathbf{x}) u_\perp(\mathbf{x}) d^2\mathbf{x}}{\int_A v(\mathbf{x}) d^2\mathbf{x}} \tag{4.11}$$

where:

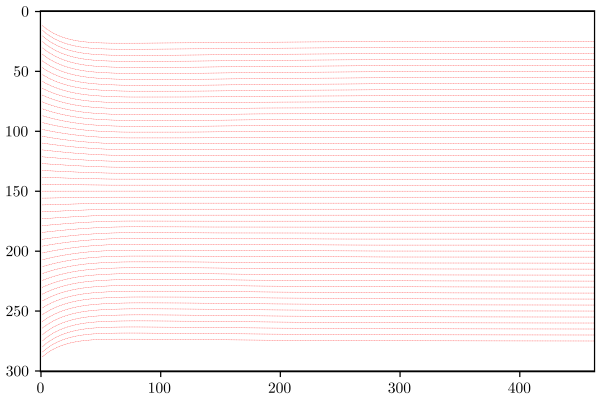
- A is an arbitrary two dimensional surface perpendicular to the flow direction
- $\lambda(\mathbf{x})$ is the streamline length at $\mathbf{x}$
- $u_\perp(\mathbf{x})$ is the velocity component perpendicular to the surface A

This surface integral can be transformed into a volume integral [149] which yields a particularly simple form of the tortuosity that can be calculated directly from the LBM velocity field:

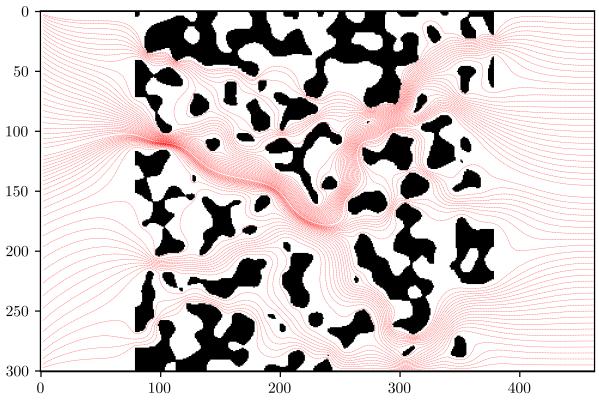
$$\theta = \frac{1}{L_x} \frac{\int_A \lambda(\mathbf{x}) u_\perp(\mathbf{x}) d^2\mathbf{x}}{\int_A u_\perp(\mathbf{x}) d^2\mathbf{x}} = \frac{\langle \|\mathbf{u}\| \rangle}{\langle \|u_x\| \rangle} \tag{4.12}$$

From a practical point of view, we sample a set of initial points at the outlet along the y-axis and integrate with an Euler scheme the velocity field backwards through the flow up to the inlet. This is to avoid taking into consideration the streamlines that do

not cross the entire domain or terminate within a dead end pore or circular loop. Alternatively, we use the formula in equation 4.12 to calculate the tortuosity from the LBM resolved velocity field directly. The resulting streamlines are shown in figure 4.14 for a standard Poiseuille geometry and a porous geometry.



(a) Poiseuille flow



(b) $\epsilon = 0.7$

Figure 4.14: Streamlines of the flow in (a) a Poiseuille flow and (b) a generated porous geometry with $\epsilon = 0.7$

As a sanity check, the streamlines in figure 4.14.a are parallel to the direction of flow as expected for a laminar pipe flow. Therefore, the distance travelled by a fluid particle between the inlet and the outlet is equal to the horizontal length of the simulation domain. Consequently, the tortuosity is $\tau = 1$. This

is not the case for the porous microstructure, as the fluid has to travel through the pore space and around the solid elements, resulting in a longer travel distance. We can also observe in figure 4.14.b that the density of the streamlines increases in some regions of the domain. This is due to the fact that the flow velocity and therefore the local flux is higher in these regions. Measuring the tortuosity of the structure using equation 4.12 takes this into account as the length of the streamlines in each surface element A is weighted by the local flux through this surface which is the term $u_{\perp}(\mathbf{x})$.

4.3.2 *Anisotropy*

The gaussian kernel that was used previously is characterized by a scalar standard deviation σ. This means that the smoothing kernel is isotropic, and consequently the resulting pore structure is also isotropic. We can therefore use a multivariate normal distribution to generate an anisotropic pore structure with a specific standard deviation along each axis, by using the covariance matrix as shown in Equation 4.15:

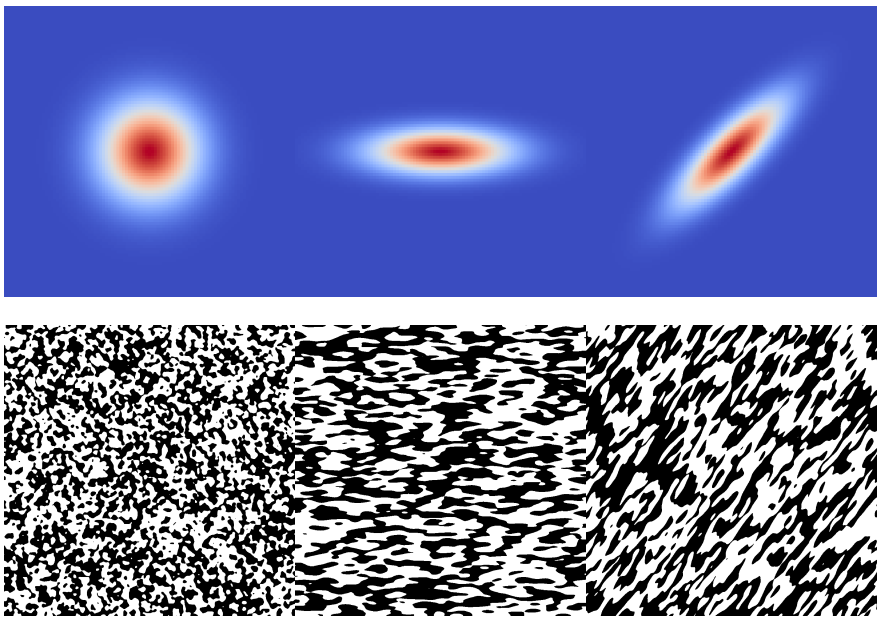
$$\Sigma = \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \tag{4.13}$$

It is also possible to have specific standard deviations along any arbitrary angle θ by applying a simple rotation matrix to the covariance matrix Σ:

$$R_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4.14}$$

The multivariate normal distribution is then defined as:

$$f(x) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}} e^{-\frac{1}{2}\left(x^T\left(R_\theta \Sigma R_\theta^T\right)^{-1}x\right)} \tag{4.15}$$



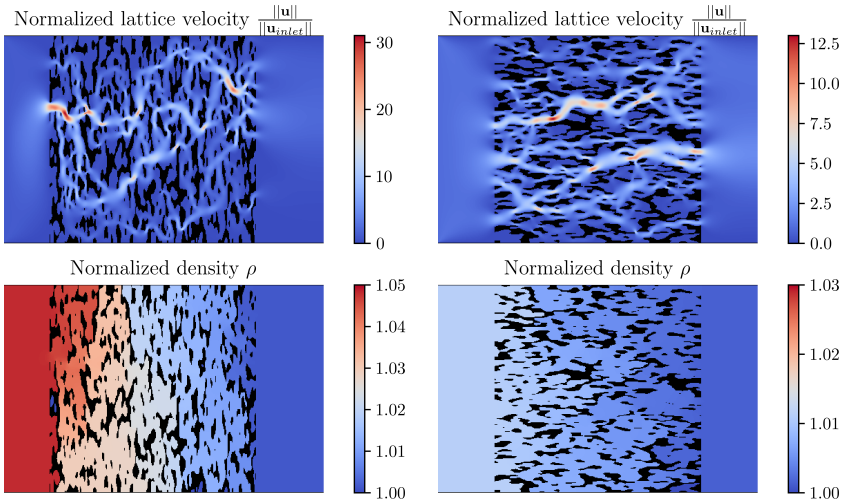
(a) $\Sigma_0 = I_d$, $\theta = 0$ (b) $\Sigma_b = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$, $\theta = 0$ (c) $\Sigma_a = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$, $\theta = \frac{\pi}{4}$

Figure 4.15: Different kernels used in the gaussian smoothing step by varying the covariance matrix Σ and the rotation angle θ. The resulting microstructures are shown in the bottom row. (a) The original kernel with scalar standard deviation creates an isotropic geometry (b) A kernel with a larger standard deviation along the horizontal axis adds a preferred horizontal direction (c) A kernel with a larger standard deviation along the horizontal axis is rotated by $\frac{\pi}{4}$ radians creates a preferred direction along that angle.

In Figure 4.15, we present a visual comparison of the effects that different kernels have on the gaussian smoothing process, achieved by altering the covariance matrix, Σ, and the rotation angle, θ. The top row of the figure delineates the three distinct kernel shapes employed in the smoothing step, each designed to introduce a specific anisotropy into the system. Panel (a) depicts the original gaussian kernel, characterized by a scalar standard deviation. This kernel is isotropic, meaning it smoothes the microstructure uniformly in all directions, thus preserving the original geometry without introducing directional bias. In contrast, panel (b) illustrates a kernel modified to have an elongated standard deviation along the horizontal axis. This modification imprints an anisotropic effect, resulting in a smoothing that is more pronounced in the horizontal direction. The outcome is a microstructure with a preferred orientation, which could be useful in systems where directional properties are under

examination. Panel (c) advances this concept by rotating the anisotropic kernel from panel (b) by $\frac{\pi}{4}$ radians. The rotation alters the direction of anisotropy, aligning it along the new axis defined by the angle of rotation. The bottom row showcases the resulting microstructures, highlighting how the orientation of the kernel influences the directional properties of the smoothed microstructures. These variations can be pivotal in tailoring the material properties for specific applications where directional characteristics are essential.

In this regard, the orientation of the generated microstructure can influence the pressure drop across the simulation domain. In figure 4.16, we illustrate the influence of microstructure orientation on the fluid dynamics within a channel, specifically focusing on the pressure drop and lattice velocity profiles. The figure is split into two panels, each representing a distinct orientation of the microstructure and its corresponding effect on fluid flow. Panel (a) shows a microstructure oriented perpendicular to the fluid flow in the channel. The pressure drop across this microstructure is mapped, indicating how the fluid experiences resistance as it moves from one side of the channel to the other. The lattice velocity profile in this configuration reveals how the velocity of the fluid varies within the interstices of the microstructure, which can be indicative of the flow permeability and the potential for turbulent flow regions. In panel (b), the microstructure is oriented in parallel to the flow. The resulting pressure drop is notably lower than the vertically orientated geometry. The comparison between the two orientations in this figure is crucial for understanding how microstructure alignment can be exploited to optimize fluid flow within engineered systems. For instance, a microstructure perpendicular to the flow may increase the time a fluid particle spends within the pores, which can be beneficial for reactant utilization. On the other hand, a microstructure parallel to the flow may reduce the pressure drop across the system, which can be advantageous for reducing pumping costs.



(a) $\Sigma = \begin{bmatrix} \sigma & 0 \\ 0 & 10\sigma \end{bmatrix}$ (b) $\Sigma = \begin{bmatrix} 10\sigma & 0 \\ 0 & \sigma \end{bmatrix}$

Figure 4.16: Pressure drop and lattice velocity across a channel for a (a) vertically and (b) horizontally oriented microstructure.

This is further corroborated in Figure 4.16, where we plot the pressure drop across a channel for a given orientation of the microstructure. Due to the stochastic nature of the pore generation scheme, the pressure drop is not deterministic and the simulation is repeated multiple times for each orientation. The figure shows the mean and standard deviation of the pressure drop across the channel for each orientation.

The orientation of the kernel also affects the hydraulic tortuosity of the microstructure, which is shown in figure 4.18 as a function of the kernel angle θ. It is clear that the more aligned the microstructure is with the flow, the lower the hydraulic tortuosity. It should also be noted the for both the pressure drop and the hydraulic tortuosity, the variation between different runs on the same microstructure orientation is inversely proportional to the angle, as geometries that are oriented along the flow direction result in more consistent pressure drops and toruosities. This can be explained by the fact that as the geometry increasingly constricts the flow due to the misalignment between the flow direction and the angle θ, the influence of individual pores on the overall flow becomes more significant, resulting in a higher variation between runs.

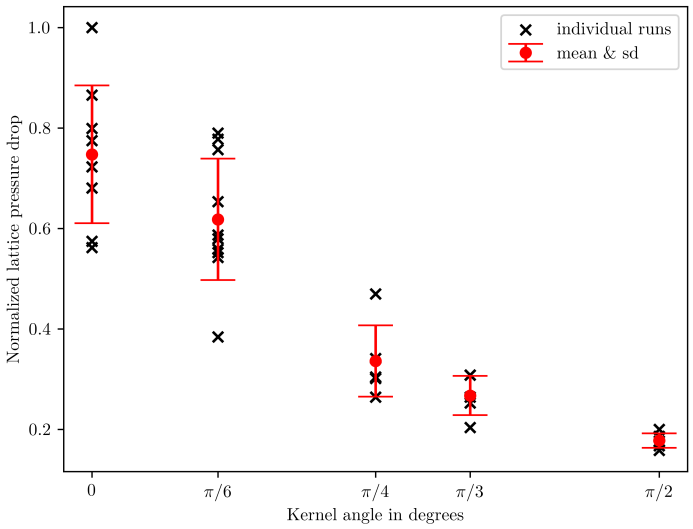


Figure 4.17: Pressure drop across a channel for different orientations of the microstructure.

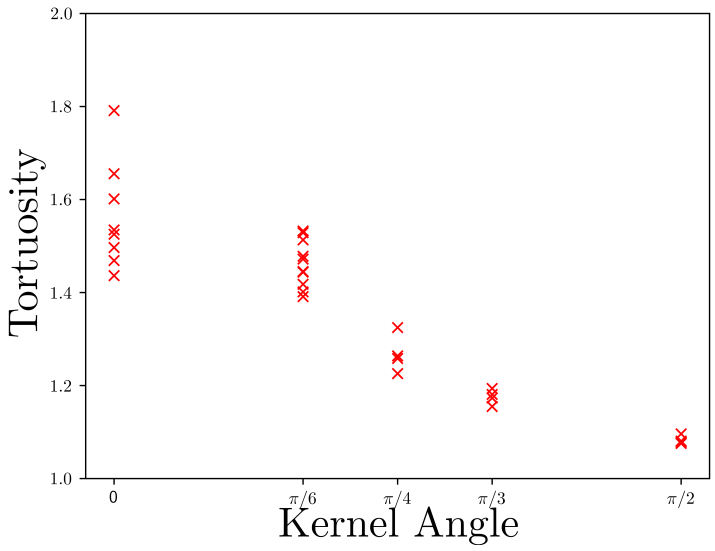


Figure 4.18: Hydraulic tortuosity of the microstructure as a function of the kernel angle θ across multiple measurements for each angle.

4.3.3 *Cyclic Voltammetry in LBM*

In this section we will investigate the relationship between the porosity of a working electrode and the current in operation.

The porosity of the electrode, ϵ, represents the fraction of the electrode volume that is composed of open pore space. In general, high porosity tends to correlate with a large active surface area and consequently high currents are generated, so it is expected that the maximum generated current should depend on the porosity up to a point.

The current LBM model is single phase, which accounts for the electrolyte phase only in the Navier-Stokes solver. This corresponds to the assumption that there is full-wetting of the electrode and that there is no second phase (air) that could affect the active surface area of the electrode. The densities in the Advection-Diffusion solver represent the concentrations of the reactants throughout the electrolyte phase. Consequently, the reactants are able to diffuse into the pores and reach the electrode surface in order to undergo electron transfer.

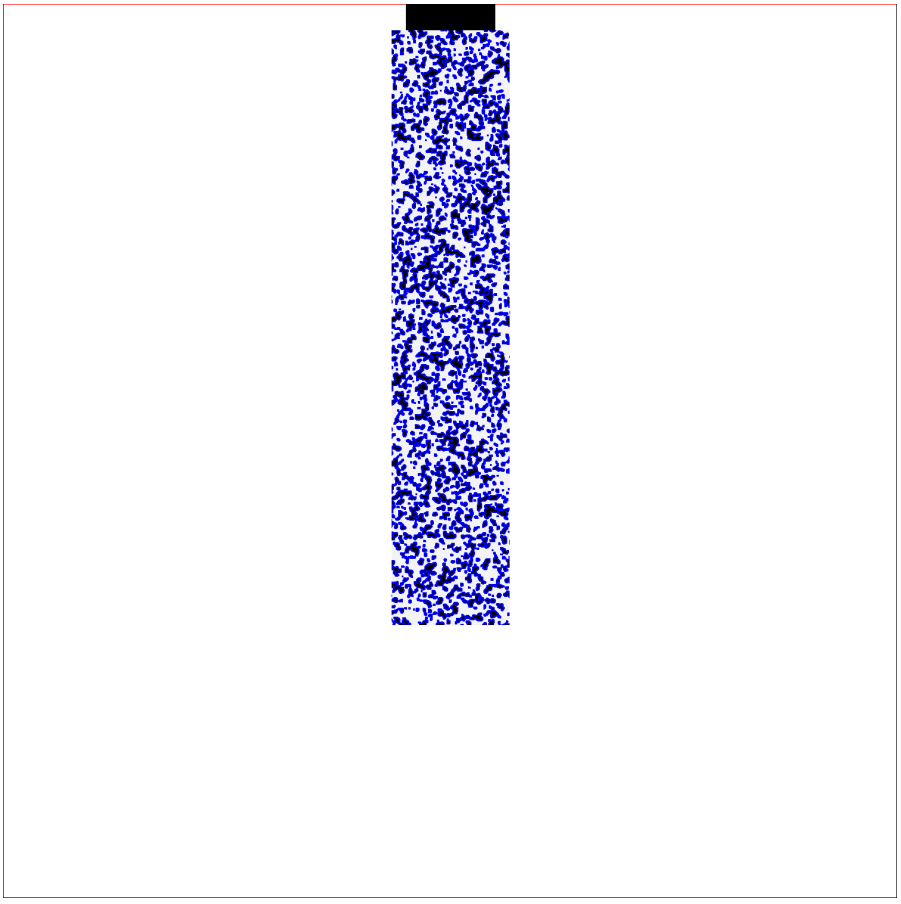
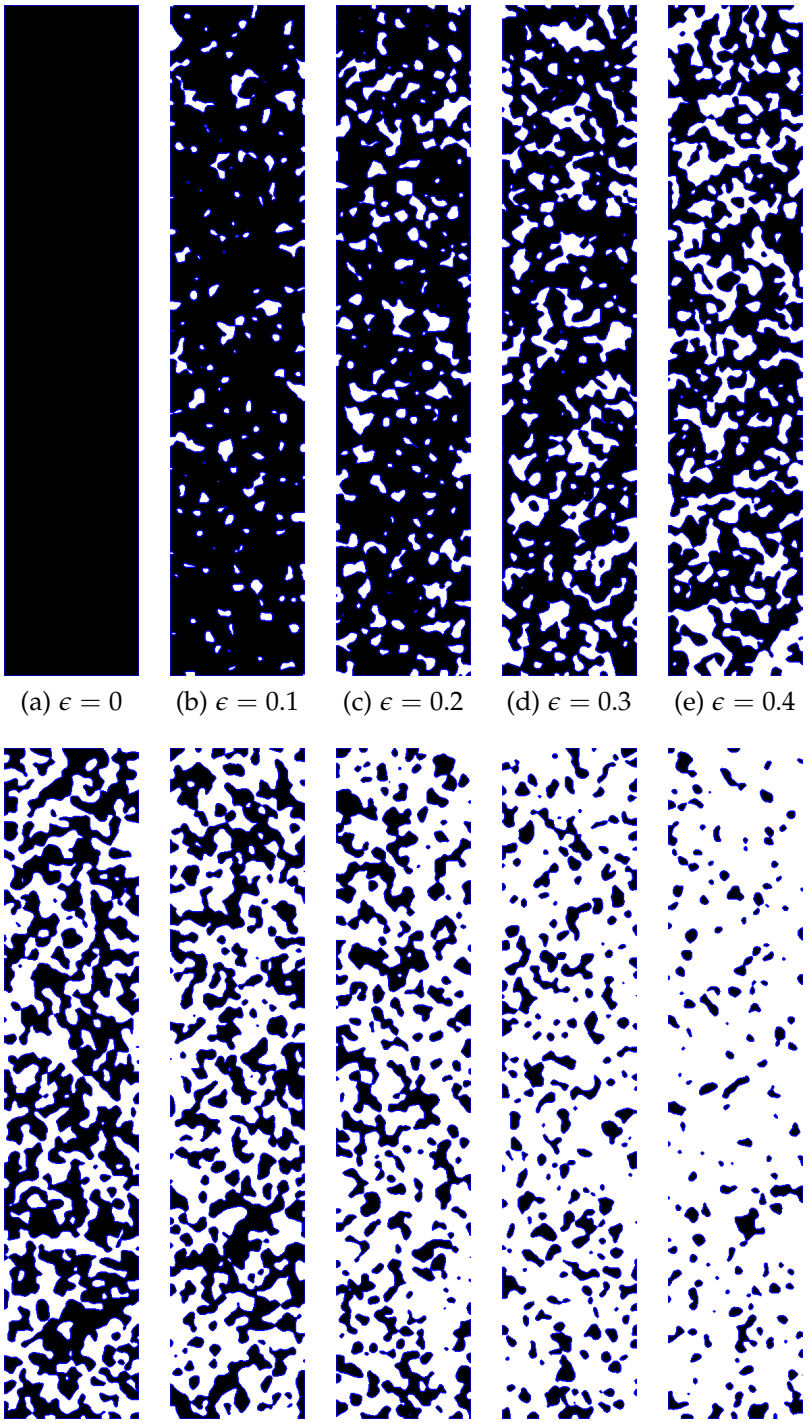


Figure 4.19: Overall simulation geometry of the porous electrode. The active surface area nodes are distinguished by a blue color coding, while the bounce-back nodes of LBM are represented in black.

The simulation conditions for this study were as follows: a redox species with the half reaction $Ox + e^- \rightleftharpoons Red$ and a standard potential of 0.6 V (vs. an arbitrary reference potential that is taken as 0 V in the simulation) underwent a cyclic voltammetry (CV) cycle between a maximum potential of $E_{max} = 2.6$ V and a minimum potential of $E_{min} = -1.4$ V. The simulation domain is shown in Figure 4.19 and corresponds to an electrochemical cell. The domain has bounceback boundary conditions on all the edges of the geometry and the fluid phase corresponding to the electrolyte solution is initialized with a uniform concentration.

To investigate the relationship between porosity and the maximum current generated during CV, a porous electrode with varying porosity was generated and used. A total of 10 structures with different porosities were generated. Figure 4.20 shows the pore structures generated using the previously described scheme that involves the generation of a matrix of random numbers, Gaussian filtering for smoothing, selection of a cutoff based on the desired porosity, and the use of a boolean matrix to indicate the porous regions of the electrode. Each different structure is generated by varying the cutoff value c using the probit function described previously in order to obtain a porosity ranging from $\epsilon = 0$ to $\rho = 0.9$. For a porosity of zero as shown in Figure 4.20a, the electrode is completely solid and the entire simulation domain reserved for the electrode is considered solid, except for the edges of the domain which are considered active surface area in a blue color, representing a planar electrode. For very low porosities ($\epsilon = 0.1$, $\epsilon = 0.2$) we can see that the solid regions are still largely dominant in the simulation domain, and the pore space is reduced to a few small and mostly disconnected regions. At this range of porosities, most of the available active surface area is still on the edge of the domain. On the other hand, as we increase the porosity beyond $\epsilon = 0.7$, the pore space (shown in white) is becoming increasingly more dominant, reducing the size of the solid regions (shown in black). This is expected as the cutoff value c is increased, the probability of a random matrix element being higher than c is decreased, making it more likely to be considered a pore space element instead of a solid node element.



(a) $\epsilon = 0$ (b) $\epsilon = 0.1$ (c) $\epsilon = 0.2$ (d) $\epsilon = 0.3$ (e) $\epsilon = 0.4$

(f) $\epsilon = 0.5$ (g) $\epsilon = 0.6$ (h) $\epsilon = 0.7$ (i) $\epsilon = 0.8$ (j) $\epsilon = 0.9$

Figure 4.20: Generated pore structures with different porosities ρ ranging from $\rho = 0$, to a porosity of $\rho = 0.9$. The interface between the solid and fluid regions is indicated in blue.

Furthermore, the available area for electron transfer is reduced significantly with $\epsilon = 0.8$ and $\epsilon = 0.9$. When the porosity

parameter is in the medium range ($\epsilon = 0.4$, $\epsilon = 0.5$ and $\epsilon = 0.6$), the pore space and solid regions are relatively matched and neither is dominant, making the pore space relatively well connected and the active surface area is distributed throughout the simulation domain.

The potential at the electrode is cycled at a scan rate of 100 mV/s and the current response is shown in figure 4.21 for different porosities. Each plot captures the current response of the electrode as a function of the applied voltage, revealing characteristic redox peaks whose magnitudes are indicative of the electrochemical activity within the electrode. The magnitude of the CV waves is initially low for low porosities, and increases up to a peak value at a porosity of $\epsilon = 0.5$, and subsequently drops off.

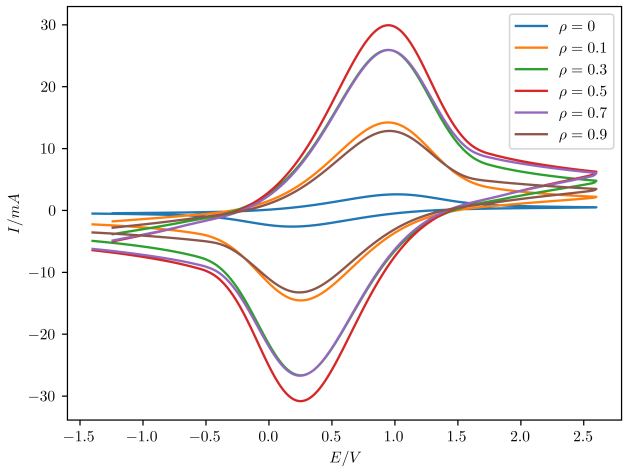


Figure 4.21: Cyclic voltammetry plots for the porous electrode at different levels of porosity.

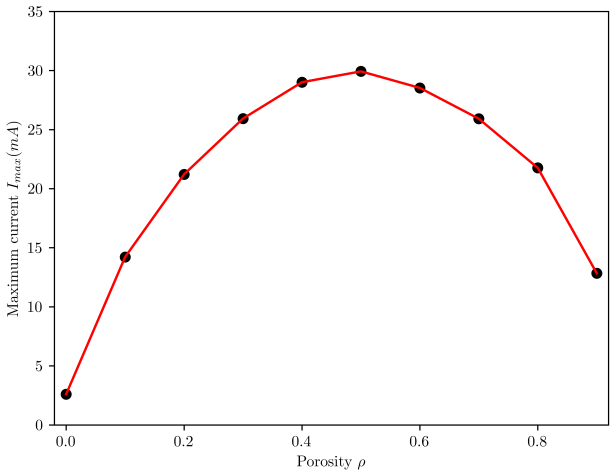


Figure 4.22: Maximum current generated as a function of porosity during cyclic voltammetry runs on the porous electrode.

If we select out the peak current in each cyclic voltammogram and plot it as a function of the porosity, we obtain the plot in figure 4.22. We can see that the maximum current increases with the porosity up to a point, and then starts to decrease. This is due to the fact that the porosity is directly linked to the surface area of the electrode, and therefore the maximum current is also linked to the surface area. To elucidate, as the porosity increases, the pore space becomes more dominant and the solid regions are reduced, which reduces the available active surface area for the electron transfer reaction. This is why the maximum current starts to decrease for $\epsilon \geq 0.5$.

4.4 SUMMARY

In this chapter we delved into the practical details of the implementation of the electrochemistry, convection-diffusion and fluid flow problem with the Lattice Boltzmann Method. The development of the code for the project was carried out using the Python machine learning library *PyTorch* in order to take advantage of Python's convenient development environment while using the library's *CUDA* backend make use of the GPU's parallel architecture to enhance performance. This configuration circumvents tedious implementation details and bug-prone GPU programming so that we can focus more on the algorithmic aspects of the model while still performing most computationally

intensive tasks such as streaming or collision operations on the GPU.

The second part of this chapter was dedicated to comparing the numerical results of the LBM-based model with well-known problems with analytical solutions. First, we used the Cottrell experiment, which is a pure linear 1D diffusion problem, to calculate the diffusion limited current at a planar electrode. Secondly, the Leveque approximation in the Graetz problem was used for mass transport in a channel with a fully developed parabolic velocity profile. In this case, linearizing this profile near the electrode for cases where the Peclet number is high enough to render the diffusion layer thickness negligible compared to the channel height allows us to obtain an analytical formula that we can benchmark the LBM-based model on.

In the third and final part of the chapter, we introduced a stochastic scheme that allows us to generate porous geometries that we can use in our Lattice Boltzmann simulations. The method allows us to select a target porosity for our synthetic microstructures, as well as tune the average feature size, pressure drop and tortuosity. By using asymmetric mutilvariable normal distribution kernels that we rotate at any arbitrary angle, we are able to introduce anisotropy into our geometry. Furthermore, we used these synthetic geometries to evaluate the effect of the porosity on the maximum current generated during a cyclic voltammetry experiment that we perform using our model. We found that the maximum current increases with porosity up to a maximum value at a porosity of 0.5, and then subsequently decreases due to the dominance of the pore space and consequent reduction in the surface area available for charge transfer.

Given the fact that many electrochemical energy conversion devices rely on the significant surface area of porous electrodes to achieve high current densities and power outputs, the ability to generate synthetic porous geometries with tunable porosity and feature size is a useful tool for the design and optimization of these devices. In its current state, the model we developped is two dimensional and is therefore limited to devices in which mass transport phenomena occur mostly in a two dimensional plane. It is therefore applicable to microfluidic devices such as MMRFBs due to the fact that concentration gradients are negligible in the z axis direction that is perpendicular to the device's main plane, especially flow through cells where the channel height is an order of magnitude lower than the channel lengths. However, extending this model to three dimensions is a straight-

forward task that can be achieved by using a three dimensional lattice configuration and modifying the streaming and collision operators accordingly to account for populations with a z-axis component. However, further research is required to determine if the added computational cost of a three dimensional model would be justified for the study of MMRFBs.

5

CONCLUSION AND FUTURE WORK

This thesis project aimed to integrate the domains of redox flow batteries, microfluidics and computational modelling via the Lattice Boltzmann Method. The research was split into two main axes: On one hand, we continued the investigation of the usability of Ionic Liquids as supporting electrolytes in microfluidic redox flow batteries. On the other hand, we developed a numerical model along with tools that can be used to analyze, simulate and model electrochemical energy conversion devices including microfluidic redox flow batteries.

Based on the state-of-the art of aqueous RFBs, where remarkable performance gains have been achieved through the adoption of high surface area porous electrodes with flow through designs, we opted to follow a similar path the non-aqueous redox flow battery. Part of this work was therefore dedicated to adapting this concept for a non-aqueous application. In comparison to the previous proof-of-concept cell by Chaabene et al. which used a flow-by planar electrode cell, the flow through cell architecture with porous electrodes manifested an appreciable increase power and current output in identical working conditions thanks to the large surface area provided by the carbo felt electrodes. Furthermore, a colaminar flow interface was achieved within the main channel downstream of the porous electrodes, which is crucial for minimizing reactant crossover. Importantly, due to the high viscosity of the used ionic liquid, this was accomplished at relatively large scales compared to typical microfluidic micrometric scales. Working at these scales allows for use of low-cost materials and fabrication techniques such as direct 3D stereolithography, which was a key requirement for this project.

The second axis of our work was to develop a numerical model for microfluidic or other electrochemical energy conversion devices that utilize porous electrodes. The most widely used approach throughout a range of different numerical methods is to represent the porous domain as a bulk continuum with effective flow and mass transport properties that depend on characterizing parameters such as the porosity or the permeability. In this light, the objective was to build a model that can

integrate fluid flow along with mass transport and electrochemical reactions in complex geometries. For this purpose, the Lattice Boltzmann Method was chosen due to its amenability to include arbitrary geometries as long as they can be represented in a voxellized structure. LBM is relatively new numerical method that solves the Navier-Stokes equations indirectly by tracking the evolution of mesoscopic populations. Due to the fact that it can be modified and extended to include other multiphysics phenomena, we were able to use in this project as a solver for the mass transport problem in addition to the fluid flow. After including reaction kinetics using the Butler-Volmer equation, we were able to compare results of the model with analytical solutions for simple geometries. In order to use LBM's compatibility with complex geometries, we showcased a stochastic method for generating porous media with tailored porosity, feature size and anisotropy. The generated porous microstructures were subsequently used in order to evaluate the effect of porosity on a given electrode in a standard electrochemical cell. This was achieved by performing cyclic voltammetry simulations directly on the model by cycling the potential at a certain scan rate and measuring the current response. The model is therefore a useful tool for the design and optimization of devices that use porous media without requiring the use of bulk effective parameters or continuum assumptions. Nevertheless, ideally the model would be able to predict device performance given a set of operating conditions and geometry. However, this would require the inclusion of two-electrode systems and coupling the two half-cell models along with an external circuit through which discharge occurs, which is non-trivial. A potential next step would be to incorporate the third dimension into the model, which would certainly increase the computational cost but could potentially allow modelling of a more general set of batteries and electrochemical devices.

Part I

APPENDIX

A

APPENDIX: LBM IMPLEMENTATION IN PYTHON

This first appendix contains the python code that implements the various parts of the numerical model.

A.1 STANDARD NAVIER-STOKES SOLVER WITH TORCH

This is the standard incompressible Navier-Stokes equations solver with a D2Q9 lattice and the BGK collision operator:

```
import matplotlib.pyplot as plt
import torch
import threading
import queue
from PIL import Image
import numpy as np
from alive_progress import alive_bar
import time
from util import *
# To Generate ffmpeg video from images
# ffmpeg -f image2 -framerate 30 -i %05d.png -s 1080x720 -pix_fmt yuv420p output.mp4
# to speed up playback to real time use
# ffmpeg -i input.mp4 -filter:v "setpts=0.5*PTS" output.mp4
# instead of 0.5 use 1/(current_playback_length / target_playback_length)

"""Simulation parameters"""
u_ph = 0.01  # m/s ~ 5mm/s
visc_ph = 1.0035e-6  # m^2/s water at 25C
inlet_width_ph = 0.00382  # m = .23cm
re_ph = u_ph * inlet_width_ph / visc_ph  # Reynolds number
cell_length_ph = 3e-2  # 3cm

# Create obstacle tensor from numpy array‘
obstacle = generate_obstacle_tensor('input/levequev2.png')
obstacle = obstacle.clone().to(device)
nx, ny = obstacle.shape  # Number of nodes in x and y directions
omega_l = 1.
```

```
re, dx, dt, ulb = convert_from_physical_params_ns(
    cell_length_ph, inlet_width_ph, u_ph, visc_ph, nx,
    omega_l)
input("Press enter to continue...")


def equilibrium():
    global feq
    # Calculate equilibrium populations (Kruger et al., page
        64)
    usqr = 3 / 2 * (u[0] ** 2 + u[1] ** 2)
    cu = 3 * torch.einsum('ixy,ji->jxy', u, c)  # previously
        ijk,li->ljk
    feq = rho * w.view(9, 1, 1) * (1 + cu + 0.5 * cu ** 2 -
        usqr)


# Initialize macroscopic variables
rho = torch.ones((nx, ny), device=device).float()
u = torch.zeros((2, nx, ny), device=device).float()
last_u = torch.zeros((2, nx, ny), device=device).float()
du = []

# Initialize populations
feq = torch.zeros((9, nx, ny), device=device).float()
equilibrium()  # Initialize equilibrium populations
fin = feq.clone()  # Initialize incoming populations (pre-
    collision)
fout = feq.clone()  # Initialize outgoing populations (post-
    collision)


def step():
    global fin, fout, rho, u, last_u, du
    # Perform one LBM step
    # Outlet BC
    # Doing this first is more stable for some reason
    fin[left_col, -1, :] = fin[left_col, -2, :]
    macroscopic()  # Calculate macroscopic variables
    # Calculate velocity difference
    du.append(torch.norm(u - last_u).cpu().item())
    last_u = u.clone()
    # Impose conditions on macroscopic variables
    # u[0, 0, :] = ulb * torch.ones(ny, device=device).float
        ()
    u[0, 0, :] = 0
```

```
        u[0, 0, 3:-3] = poiseuille_inlet(ulb, ny - 6)
        rho[0, :] = 1 / (1 - u[0, 0, :]) * (torch.sum(fin[
            center_col, 0, :], dim=0) +
                                                    2 * torch.
                                                        sum(
                                                        fin[
                                                        left_col
                                                        , 0,
                                                        :],
                                                        dim=0)
                                                        )

71      # Equilibrium
        equilibrium()

        # Boundary conditions on populations
        # Zou-He BC Fin = Feq + Fin(op) - Feq(op)
76      fin[right_col, 0, :] = feq[right_col, 0, :] + fin[
            left_col, 0, :] - feq[left_col, 0, :]

        # BGK collision
        fout = fin - omega_l * (fin - feq)

81      # Bounce-back
        fout[:, obstacle] = fin[c_op][:, obstacle]

        # Streaming
        stream()
86


    def run(iterations: int, save_to_disk: bool = True, interval
        : int = 100, continue_last: bool = False):
        # Launches LBM simulation and a parallel thread for
            saving data to disk
        global rho, u, fin, fout, dx, dt
91      print(f"Simulating {iterations * dt} seconds")

        if continue_last:  # Continue last computation
            rho = torch.from_numpy(np.load("output/
                BaseLattice_last_rho.npy")).to(device)
            u = torch.from_numpy(np.load("output/
                BaseLattice_last_u.npy")).to(device)
96          equilibrium()
            fin = feq.clone()  # Initialize incoming populations
                (pre-collision)
```

```
            fout = feq.clone()  # Initialize outgoing
                populations (post-collision)

        if save_to_disk:
            # Create queue for saving data to disk
            q = queue.Queue()
            # Create thread for saving data
            t = threading.Thread(target=save_data, args=(q,
                obstacle))
            t.start()

        # Run LBM for specified number of iterations
        with alive_bar(iterations) as bar:
            start = time.time()
            counter = 0
            for i in range(iterations):
                step()  # Perform one LBM step
                if i % interval == 0:
                    # Calculate MLUPS by dividing number of
                        nodes by time in seconds
                    delta_t = time.time() - start
                    mlups = nx * ny * counter / (delta_t * 1e6)
                    if save_to_disk:
                        # push data to queue
                        velocity = convert_to_physical_velocity(
                            u, dx, dt)
                        q.put(((velocity, rho), f"output/{i //
                            interval:05}.png"))  # Five digit
                            filename
                    # Reset timer and counter
                    start = time.time()
                    counter = 0

                counter += 1
                bar.text(f"MLUPS: {mlups:.2f}, du: {du[-1]:.5e}"
                    )
                bar()

        # Save final data to numpy files
        np.save(f"output/BaseLattice_last_u.npy", u.cpu().numpy
            ())
        np.save(f"output/BaseLattice_last_rho.npy", rho.cpu().
            numpy())
        fig, ax = plt.subplots()
        ax.semilogy(np.asarray(du[2:]))
        plt.show()
```

```
    if save_to_disk:
        # Stop thread for saving data
        q.put((None, None))
        t.join()


if __name__ == '__main__':
    print("Using device: ", device)
    run(10000, save_to_disk=True, interval=100,
        continue_last=False)
```

A.2 CONVECTION-DIFFUSION EQUATION SOLVER

The following code solves for the convection-diffusion equation in a microfluidic flow-through cell in the mass transport limited regime. This requires a precalculated velocity field in steady state which is the result of the solver in the previous section

```
import matplotlib.pyplot as plt
import torch

from util import *
import threading
import queue
from alive_progress import alive_bar
import time
import numpy as np

# Physical dimensions
cell_length_ph = 5e-2  # 2.5cm or 0.025m
channel_width_ph = 5e-3  # 5mm or 0.005m
depth_ph = 5e-3  # 5mm
diff_ph = 0.76e-9  # m^2/s (From Allen , Bard appendix for
    Ferrocyanide, page 831)
vel_ph = 0.01  # m/s
Pe = vel_ph * channel_width_ph / diff_ph  # Peclet number
concentration_ph = 100  # mol/m^3 ~ 0.1M

input_image = "input/mmrfbs/depletion_130w.png"

obstacle = generate_obstacle_tensor(input_image)
fluid = ~obstacle
electrode1 = generate_electrode_tensor(input_image, BLUE)
# TODO: properly treat velocity (should be below 0.1)
```

```
v_field = torch.from_numpy(np.load("output/BaseLattice_last_u.npy"))
v_field = v_field.to(device)

# Electrode lengths for current densities
electrode1_size = torch.sum(electrode1)

# inlets have different color coding to electrodes
inlet_bottom = generate_electrode_tensor(input_image, GREEN)
inlet_top = generate_electrode_tensor(input_image, YELLOW)
outlet = generate_electrode_tensor(input_image, RED)


# Diffusion constant
nx, ny = obstacle.shape
omega_l = 1.99
pe, dx, dt, d_l, ulb = convert_from_physical_params_diff(cell_length_ph, channel_width_ph, vel_ph, diff_ph, nx, omega_l)
input("Press enter to continue...")
v_field = v_field / torch.max(v_field) * ulb

# Initialize
rho_ox_1 = torch.zeros((nx, ny), dtype=torch.float64, device=device)
feq_ox_1 = torch.zeros((9, nx, ny), device=device)
rho_ox_1[:, 900:] = 1
rho_ox_1[obstacle] = 0
outlet_c = 0
outlet_hist = []


def equilibrium():
    global feq_ox_1
    cu = torch.einsum('ixy,ji->jxy', v_field, c)
    feq_ox_1 = w.view(9, 1, 1) * rho_ox_1 * (1 + 3 * cu)


equilibrium()

fin_ox_1 = feq_ox_1.clone()
fout_ox_1 = feq_ox_1.clone()


def macroscopic():
    global rho_ox_1
```

```
    # With source term correction as shown in Kruger 310
    rho_ox_1 = torch.clamp(torch.sum(fin_ox_1, 0), 0, 10)


def step(i):
    global fin_ox_1, fout_ox_1, outlet_c, outlet_hist

    fin_ox_1[left_col, -1, :] = fin_ox_1[left_col, -2, :]

    macroscopic()
    # Inlet concentrations
    rho_ox_1[inlet_bottom] = 1
    rho_ox_1[inlet_top] = 0
    rho_ox_1[electrode1] = 0
    rho_ox_1[obstacle] = 0

    # inlet flux
    inlet_flux = torch.sum(-v_field[1, inlet_bottom] *
        rho_ox_1[inlet_bottom])

    outlet_flux = torch.sum(v_field[0, outlet] * rho_ox_1[
        outlet])

    ratio = outlet_flux / inlet_flux
    outlet_hist.append(ratio.cpu().numpy())

    equilibrium()

    # Zhou He BC
    fin_ox_1[right_col, 0, :] = feq_ox_1[right_col, 0, :] +
        fin_ox_1[left_col, 0, :] - feq_ox_1[left_col, 0, :]

    fout_ox_1 = fin_ox_1 - omega_l * (fin_ox_1 - feq_ox_1)

    # Bounce Back
    fout_ox_1[:, obstacle] = fin_ox_1[c_op][:, obstacle]
    # Streaming
    stream(fin_ox_1, fout_ox_1)


def run(iterations: int, save_to_disk: bool = True, interval
    : int = 100, continue_last: bool = False):
    global rho_ox_1, fin_ox_1

    print(f"Simulating {iterations * dt} seconds")
```

```
    if continue_last:
        rho_ox_1 = torch.from_numpy(np.load("output/
            Depletion_last_rho_ox_1.npy")).to(device)

        equilibrium()

        fin_ox_1 = feq_ox_1.clone()

    if save_to_disk:
        q = queue.Queue()
        t = threading.Thread(target=save_data, args=(q,))
        t.start()

    with alive_bar(iterations, force_tty=True) as bar:
        start = time.time()
        counter = 0
        for i in range(iterations):
            step(i)
            if i % interval == 0:
                delta_t = time.time() - start
                mlups = nx * ny * counter / (delta_t * 1e6)
                if save_to_disk:
                    q.put((rho_ox_1.clone(), f"output/{i //
                        interval:05}.png"))
                start = time.time()
                counter = 0
            if i % 1000 == 0:
                # periodically save the state
                np.save(f"output/temp/
                    Depletion_last_rho_ox_1.npy", rho_ox_1.
                    cpu().numpy())
            counter += 1
            bar.text(f"MLUPS: {mlups:.2f}, outlet: {
                outlet_hist[-1]:.5e}")
            bar()

    # save final state
    np.save(f"output/Depletion_last_rho_ox_1.npy", rho_ox_1.
        cpu().numpy())
    fig, ax = plt.subplots()
    ax.plot(np.asarray(outlet_hist[2:]))
    plt.show()

    if save_to_disk:
        q.put((None, None))
        t.join()
```

```
151
def save_data(q: queue.Queue):
    while True:
        data, filename = q.get()
        if data is None:
156             break
        plt.clf()
        plt.axis('off')
        data[obstacle] = np.nan
        plt.imshow(data.cpu().numpy().transpose(), cmap=cmap
            )
161     plt.colorbar()
        plt.savefig(filename, bbox_inches='tight',
            pad_inches=0, dpi=500)
        plt.close()


166 if __name__ == "__main__":
    run(20000, save_to_disk=True, interval=1000,
        continue_last=False)
```

A.3 CYCLIC VOLTAMMETRY CYCLING IN LBM

This code performs the cyclic voltammetry experiment in LBM by coupling the convection-diffusion equation from the previous section with source terms governed by Butler-Volmer kinetics.

```
import matplotlib.pyplot as plt
import numpy as np
3 import torch
from PIL import Image
from util import *
import threading
import queue
8 from alive_progress import alive_bar
import time
from scipy.signal import sawtooth

"""
13 Solves the convective-diffusion equation for a species in an
     electrochemical system and performs cyclic voltammetry
```

```
# Converting from physical to lattice units we dx_l = 1,
    dt_l = 1, rho_l = 1, therefore our conversion factors
    are:
C_rho = concetration_ph
C_t = dt
C_length = dx
in order to convert between two systems we use
physical_quantity = lattice_quantity * C_quantity (Kruger
    page 272)
for example the conversion factor for velocity C_u is
    C_length / C_t = dx / dt
"""

# Physical constants
cell_size_ph = 5e-2  # 5cm or 0.05m
cell_depth_ph = 5e-3  # 5mm or 0.005m
concentration_ph = 100  # mol/m^3 or 0.1M

z = 1  # Number of electrons transferred
E_0 = 0.6  # Standard potential
d_ph = 0.76e-9  # m^2/s Diffusion coefficient (Bard page
    1013)
j_0 = 10  # Exchange current density in A/m^2
j_0_ph = 5 * 10000  # A/m2 from lvov page 129 for FerriFerro

electrode = generate_electrode_tensor("input/echem_cells/
    planar_electrode.png")
obstacle = generate_obstacle_tensor("input/echem_cells/
    planar_electrode.png")
nx, ny = obstacle.shape
v_field = torch.zeros((2, nx, ny), device=device)

"""Simulation parameters"""
# Diffusion coefficient
omega_l = 1.98
pe, dx, dt, d_l = convert_from_physical_params_diff(
    cell_size_ph, cell_size_ph, 0, d_ph, nx, omega_l)
tau = 1 / omega_l
# j0_l = j0 / dx * cell_depth_ph  # A/lattice_site
input("Press enter to start...")

"""Initialization"""
# Initialize scalar field for species concentration
rho_ox = torch.ones((nx, ny), dtype=torch.float64, device=
    device)
```

```
rho_red = torch.ones((nx, ny), dtype=torch.float64, device=
    device)
rho_ox[:, 1] = 1  # Inlet concentration
rho_red[:, 1] = 1  # Inlet concentration

feq_ox = torch.zeros((9, nx, ny), dtype=torch.float64,
    device=device)
feq_red = torch.zeros((9, nx, ny), dtype=torch.float64,
    device=device)


def equilibrium():
    global feq_ox, feq_red
    # Calculate equilibrium populations (Kruger page 304)
    cu = torch.einsum('ixy,ji->jxy', v_field, c)  # TODO:
        Precalculate if v_field is constant
    feq_ox = w.view(9, 1, 1) * rho_ox * (1 + 3 * cu)
    feq_red = w.view(9, 1, 1) * rho_red * (1 + 3 * cu)


equilibrium()

fin_ox = feq_ox.clone()
fin_red = feq_red.clone()

fout_ox = feq_ox.clone()
fout_red = feq_red.clone()

source_ox = torch.zeros_like(rho_ox, dtype=torch.float64,
    device=device)
source_red = torch.zeros_like(rho_red, dtype=torch.float64,
    device=device)

# Set electrode potential
e = E_0 * torch.ones(10000, device=device)
# Nernst potential on electrode verify if needed R * T / (z
    * F)
e_nernst = torch.ones_like(electrode, dtype=torch.float64,
    device=device) * E_0
# Current density
j = torch.zeros_like(e_nernst, dtype=torch.float64, device=
    device)
j_log = torch.empty(10000, dtype=torch.float64, device=
    device)

"""LBM operations"""
```

```
88
    def macroscopic():
        global rho_ox, rho_red, source_ox, source_red
        # rho_ox = torch.clamp(fin_ox.sum(0) + source_ox / 2, 0,
            2)
        rho_ox = fin_ox.sum(0) + source_ox / 2
93      # rho_red = torch.clamp(fin_red.sum(0) + source_red / 2,
            0, 2)
        rho_red = fin_red.sum(0) + source_red / 2

    def step(i, rho_ox_log, src_log):
        global fin_ox, fin_red, fout_ox, fout_red, source_ox,
            source_red, rho_ox, rho_red, e_nernst, j, e
98      # Perform one LBM step
        # Outlet BC
        # Equiv. to neumann BC on concentration (null flux)
        fin_ox[left_col, -1, :] = fin_ox[left_col, -2, :]
        fin_red[left_col, -1, :] = fin_red[left_col, -2, :]
103
        macroscopic()
        rho_ox_log[i] = torch.mean(rho_ox[electrode])

        # Inlet BC
108     rho_ox[:, ny - 2] = 1  # Inlet concentration
        rho_red[:, ny - 2] = 1  # Inlet concentration

        equilibrium()

113     # Zhou He BC
        fin_ox[right_col, 0, :] = feq_ox[right_col, 0, :] +
            fin_ox[left_col, 0, :] - feq_ox[left_col, 0, :]
        fin_red[right_col, 0, :] = feq_red[right_col, 0, :] +
            fin_red[left_col, 0, :] - feq_red[left_col, 0, :]
        if i == 2900:
            print("reached peak")
118     # Electrode BC
        # TODO: Use Tafel in logarithmic space to avoid
            instabilities
        e_nernst = E_0 - (R * T) / (z * F) * torch.log(torch.
            mean(rho_red[electrode]) / torch.mean(rho_ox[
            electrode]))
        # e_nernst = torch.tensor(E_0, device=device)
        # avoid large exponents
123     exponent = (.5 * F / (R * T)) * (e[i] - e_nernst)
        # Current density in amps per m^2 in each lbm box
```

```
        j = j_0 * concentration_ph * (torch.mean(rho_ox[
            electrode]) ** .5) * (torch.mean(rho_red[electrode])
             ** .5) * \
            (torch.exp(exponent) - torch.exp(-exponent))
        # current in amps per lbm box
128     current = torch.nan_to_num(j * dx * cell_depth_ph, nan
            =0.0)

        charge = current * dt  # Charge per site (C / site)
        matter = charge / (z * F)  # substance created/consumed
            per site (mol / site)
        c_ph = matter / (dx**2 * cell_depth_ph)  # substance in
            mol per lbm box
133     # substance in lattice units source_ph = C x
            source_lattice where C is C_rho / C_t
        # Source term units are mol litre^-1 s^-1
        c_l = c_ph * dt / concentration_ph
        # c_l[c_l > rho_ox[electrode]] = 1 * rho_ox[electrode][
            c_l > rho_ox[electrode]]
        # c_l[-c_l > rho_red[electrode]] = -1 * rho_red[
            electrode][-c_l > rho_red[electrode]]
138
        # current = c_l * (concentration_ph * dx**2 *
            cell_depth_ph * z * F) / (dt**2)
        src_log[i] = torch.mean(c_l)

        source_ox[electrode] = c_l
143     source_red[electrode] = -c_l
        j_log[i] = torch.sum(current)  # Log current density

        # BGK collision
        fout_ox = fin_ox - omega_l * (fin_ox - feq_ox) + (1 - 1
            / (2 * tau)) * torch.einsum('i,jk->ijk', w,
            source_ox)
148     fout_red = fin_red - omega_l * (fin_red - feq_red) + (1
            - 1 / (2 * tau)) * torch.einsum('i,jk->ijk', w,
            source_red)

        # Bounce-back
        fout_ox[:, obstacle] = fin_ox[c_op][:, obstacle]
        fout_red[:, obstacle] = fin_red[c_op][:, obstacle]
153
        # Streaming
        stream(fin_ox, fout_ox)
        stream(fin_red, fout_red)
```

```
158
    def run(iterations: int, save_to_disk: bool = True, interval
        : int = 100, continue_last: bool = False):
        # Launches LBM simulation and a parallel thread for
            saving data to disk
        global rho_ox, rho_red, fin_ox, fin_red, fout_ox,
            fout_red, j_log, e, dt

163     print(f"Simulating {iterations * dt} seconds")

        j_log = torch.zeros(iterations, dtype=torch.float64,
            device=device)
        rho_ox_log = torch.zeros(iterations, dtype=torch.float64
            , device=device)
        src_log = torch.zeros(iterations, dtype=torch.float64,
            device=device)
168
        buffer_time = 400  # Buffer time for lbm stabilization
        # Set up electrode potential
        t = np.linspace(0, 1, iterations - buffer_time)
        # phase_shift = 0.0397885 * np.pi  # 4 pi freq
173     # phase_shift = 0.099472 * np.pi  # 8 pi fre
        phase_shift = np.pi * 0.0799  # 2 pi freq
        signal = torch.from_numpy(sawtooth(2 * np.pi * (t +
            phase_shift), 0.5)).to(device)
        max_e = 1.4 * E_0
        e = signal * (max_e / 2) * torch.ones(iterations -
            buffer_time, device=device) + E_0
178     e = torch.cat((E_0 * torch.ones(buffer_time, device=
            device), e), dim=0)
        plt.plot(e.cpu().numpy())
        plt.ylabel("Potential (V)")
        plt.xlabel("Iteration")
        plt.grid()
183     plt.show()
        plt.close()
        scan_rate = torch.abs((e[-2] - e[-1]) / dt)
        print(f"CV Scan rate is: {scan_rate * 1000:.3f}mV/s")
        input("Continue..")
188     # input("Continue?")
        # print(e[0])
        if continue_last:  # Continue last computation
            rho_ox = torch.from_numpy(np.load("output/
                Electrochemical_last_rho_ox.npy")).to(device)
            rho_red = torch.from_numpy(np.load("output/
                Electrochemical_last_rho_red.npy")).to(device)
```

```
193         equilibrium()
            fin_ox = feq_ox.clone()  # Initialize incoming
                populations (pre-collision)
            fin_red = feq_red.clone()  # Initialize incoming
                populations (pre-collision)
            fout_ox = feq_ox.clone()  # Initialize outgoing
                populations (post-collision)
            fout_red = feq_red.clone()  # Initialize outgoing
                populations (post-collision)
198
        if save_to_disk:
            # Create queue for saving data to disk
            q = queue.Queue()
            # Create thread for saving data
203         t = threading.Thread(target=save_data, args=(q,))
            t.start()

        # Run LBM for specified number of iterations
        with alive_bar(iterations, force_tty=True) as bar:
208         start = time.time()
            counter = 0
            for i in range(iterations):
                step(i, rho_ox_log, src_log)  # Perform one LBM
                    step
                if i % interval == 0:
213                 # Calculate MLUPS by dividing number of
                        nodes by time in seconds
                    delta_t = time.time() - start
                    mlups = nx * ny * counter / (delta_t * 1e6)
                    if save_to_disk:
                        # push data to queue
218                     q.put((rho_ox.detach().clone(), f"output
                            /{i // interval:05}.png"))  # Five
                            digit filename
                    # Reset timer and counter
                    start = time.time()
                    counter = 0

223             counter += 1
                bar.text(f"MLUPS: {mlups:.2f} | Electrode
                    density {rho_ox[electrode].mean().cpu().
                    numpy():.5f} | "
                         f"Potential {e[i]:.5f} | Nernst
                             Potential {torch.mean(e_nernst).cpu
                             ().numpy():.3f}")
                bar()
```

```
    # Save final data to numpy files
    np.save(f"output/Electrochemical_last_rho_ox.npy",
        rho_ox.cpu().numpy())
    np.save(f"output/Electrochemical_last_rho_red.npy",
        rho_red.cpu().numpy())

    # Plot current density
    plt.show()
    fig, ax = plt.subplots()
    ax.plot(e[buffer_time:].cpu().numpy(), j_log[buffer_time
        :].cpu().numpy())
    plt.show()

    if save_to_disk:
        # Stop thread for saving data
        q.put((None, None))
        t.join()


if __name__ == '__main__':
    print(f"omega: {omega_l}")
    run(2 * 4963, save_to_disk=True, interval=100,
        continue_last=False)
```

A.4 COTTRELL CURRENT CALCULATION

The following code calculated the cottrell current using LBM and compares the numerical result with the analytical expression.

```
import matplotlib.pyplot as plt
import numpy as np
import torch
from PIL import Image
from util import *
import threading
import queue
from alive_progress import alive_bar
import time
from scipy.signal import sawtooth

"""
Solves the convective-diffusion equation for a species in an
    electrochemical system and performs cyclic voltammetry

```

```
# Converting from physical to lattice units we dx_l = 1,
    dt_l = 1, rho_l = 1, therefore our conversion factors
    are:
C_rho = concetration_ph
C_t = dt
C_length = dx
in order to convert between two systems we use
physical_quantity = lattice_quantity * C_quantity (Kruger
    page 272)
for example the conversion factor for velocity C_u is
    C_length / C_t = dx / dt
"""

# Physical constants
cell_size_ph = 5e-2  # 5cm or 0.05m
cell_depth_ph = 2e-3  # 2mm or 0.002m
concentration_ph = 100  # mol/m^3 or 0.1M

z = 1  # Number of electrons transferred
d_ph = 0.76e-9  # m^2/s Diffusion coefficient (Bard page
    813)


electrode = generate_electrode_tensor("input/cottrell.png")
obstacle = generate_obstacle_tensor("input/cottrell.png")
nx, ny = obstacle.shape

"""Simulation parameters"""
delay = 5  # Delay before applying voltage
total_iterations = 400
# Diffusion coefficient
omega_l = 1.
fo, dx, dt, d_l = convert_from_physical_params_pure_diff(
    cell_size_ph, d_ph, nx, omega_l, total_iterations)
w_e = omega_l
lambda_trt = 1/4
# lambda_trt = (1/w_e - 0.5) * (1/w_o - 0.5)
w_o = lambda_trt / (1/(w_e) - 0.5) + 0.5
# w_o = .8
print(f"lambda_trt: {lambda_trt}")
print(f"omega_e: {w_e}, omega_o: {w_o}")
tau = 1 / omega_l
j_log = torch.zeros(1, dtype=torch.float64, device=device)
input("Press enter to start...")

"""Initialization"""
```

```
# Initialize scalar field for species concentration
rho_ox = torch.zeros((nx, ny), dtype=torch.float64, device=
    device)
rho_red = torch.ones((nx, ny), dtype=torch.float64, device=
    device)
rho_ox[obstacle] = 0
rho_red[obstacle] = 0

feq_ox = torch.zeros((9, nx, ny), dtype=torch.float64,
    device=device)
feq_red = torch.zeros((9, nx, ny), dtype=torch.float64,
    device=device)


equilibrium()

fin_ox = feq_ox.clone()
fin_red = feq_red.clone()

fout_ox = feq_ox.clone()
fout_red = feq_red.clone()


def step(i):
    global fin_ox, fin_red, fout_ox, fout_red, rho_ox,
        rho_red
    # Perform one LBM step
    # Outlet BC
    # Equiv. to neumann BC on concentration (null flux)
    fin_ox[left_col, -1, :] = fin_ox[left_col, -2, :]
    fin_red[left_col, -1, :] = fin_red[left_col, -2, :]

    macroscopic()

    if i > delay:
        # Calculate generated current
        # Charge
        q = F * (rho_red[electrode] * concentration_ph) * (
            dx ** 2) * cell_depth_ph
        total_q = q.sum()
        # Current
        j_log[i - delay] = total_q / dt

        # Electrode BC
        rho_ox[electrode] = 1
```

```
            rho_red[electrode] = 0
        equilibrium()

        # Zhou He BC
        fin_ox[right_col, 0, :] = feq_ox[right_col, 0, :] +
            fin_ox[left_col, 0, :] - feq_ox[left_col, 0, :]
        fin_red[right_col, 0, :] = feq_red[right_col, 0, :] +
            fin_red[left_col, 0, :] - feq_red[left_col, 0, :]

        # BGK collision
        # Collision
        f_plus = .5 * (fin_ox + fin_ox[c_op])
        f_minus = .5 * (fin_ox - fin_ox[c_op])

        feq_plus = .5 * (feq_ox + feq_ox[c_op])
        feq_minus = .5 * (feq_ox - feq_ox[c_op])

        fout_ox = fin_ox - w_o * (f_plus - feq_plus) - w_e * (
            f_minus - feq_minus)

        g_plus = .5 * (fin_red + fin_red[c_op])
        g_minus = .5 * (fin_red - fin_red[c_op])

        geq_plus = .5 * (feq_red + feq_red[c_op])
        geq_minus = .5 * (feq_red - feq_red[c_op])

        fout_red = fin_red - w_o * (g_plus - geq_plus) - w_e * (
            g_minus - geq_minus)

        # Bounce-back
        fout_ox[:, obstacle] = fin_ox[c_op][:, obstacle]
        fout_red[:, obstacle] = fin_red[c_op][:, obstacle]

        # Streaming
        stream(fin_ox, fout_ox)
        stream(fin_red, fout_red)


def run(iterations: int, save_to_disk: bool = True, interval
    : int = 100, continue_last: bool = False):
    # Launches LBM simulation and a parallel thread for
        saving data to disk
    global rho_ox, rho_red, fin_ox, fin_red, fout_ox,
        fout_red, j_log, dt

    print(f"Simulating {iterations * dt} seconds")
```

```
134
        j_log = torch.zeros(iterations, dtype=torch.float64,
            device=device)

        if continue_last:  # Continue last computation
            rho_ox = torch.from_numpy(np.load("output/
                Electrochemical_last_rho_ox.npy")).to(device)
139         rho_red = torch.from_numpy(np.load("output/
                Electrochemical_last_rho_red.npy")).to(device)
            equilibrium()
            fin_ox = feq_ox.clone()  # Initialize incoming
                populations (pre-collision)
            fin_red = feq_red.clone()  # Initialize incoming
                populations (pre-collision)
            fout_ox = feq_ox.clone()  # Initialize outgoing
                populations (post-collision)
144         fout_red = feq_red.clone()  # Initialize outgoing
                populations (post-collision)

        if save_to_disk:
            # Create queue for saving data to disk
            q = queue.Queue()
149         # Create thread for saving data
            t = threading.Thread(target=save_data, args=(q,))
            t.start()

        # Run LBM for specified number of iterations
154     with alive_bar(iterations, force_tty=True) as bar:
            start = time.time()
            counter = 0
            for i in range(iterations):
                step(i)  # Perform one LBM step
159             if i % interval == 0:
                    # Calculate MLUPS by dividing number of
                        nodes by time in seconds
                    delta_t = time.time() - start
                    mlups = nx * ny * counter / (delta_t * 1e6)
                    if save_to_disk:
164                     # push data to queue
                        q.put((rho_red.detach().clone(), f"
                            output/{i // interval:05}.png"))  #
                            Five digit filename
                    # Reset timer and counter
                    start = time.time()
                    counter = 0
169
```

```
            counter += 1
            bar.text(f"MLUPS: {mlups:.2f} | Electrode
                density {rho_ox[electrode].mean().cpu().
                numpy():.5f}")
            bar()

    # Save final data to numpy files
    np.save(f"output/Electrochemical_last_rho_ox.npy",
        rho_ox.cpu().numpy())
    np.save(f"output/Electrochemical_last_rho_red.npy",
        rho_red.cpu().numpy())

    # Plot current density
    plt.show()
    fig, ax = plt.subplots()
    area = electrode.sum() * dx * cell_depth_ph  # Electrode
         area (only works with planar electrodes)
    ph_time = np.arange(1, iterations) * dt
    cot_time = np.linspace(dt / 20, iterations * dt * 1.1,
        1000)
    cott = (F * float(area.cpu()) * concentration_ph * np.
        sqrt(d_ph)) / (np.sqrt(np.pi * cot_time))
    indices = get_indices_to_keep(iterations - 1, start_fine
        =0.005)
    ax.semilogy(ph_time[indices], j_log[:-1].cpu().numpy()[
        indices], "g^", label="LBM", markersize=3)
    ax.semilogy(cot_time, cott, "r—", label="Cottrell")
    ax.set_xlabel("Time (s)")
    ax.set_ylabel("Current (A)")
    ax.legend()
    plt.savefig("output/cottrell_current.png", bbox_inches='
        tight', pad_inches=0, dpi=900)
    # plt.show()
    # fig, ax = plt.subplots()
    # rel_err = j_log[:-1].cpu().numpy() / cott
    # ax.plot(ph_time, rel_err, '-g')
    # ax.set_xlabel("Time (s)")
    # ax.set_ylabel("$I / I_{Cottrell}$")
    # ax.grid()
    # ax.set_ylim(0.8, 2.5)
    # plt.show()

    if save_to_disk:
        # Stop thread for saving data
        q.put((None, None))
        t.join()
```

```
def get_indices_to_keep(length, start_fine=0.2):
    """
    Returns indices to keep for plotting.

    Parameters:
    - length: Total number of data points.
    - start_fine: Fraction of the data points at the start
        where we don't skip any points.

    Returns:
    - List of indices to keep.
    """
    start_indices = int(length * start_fine)
    remaining = length - start_indices

    # Calculate the required total sum to achieve the
        end_skip
    skip_value = np.arange(remaining)
    indices = np.concatenate((np.arange(start_indices), np.
        cumsum(skip_value) + start_indices))
    indices = indices[indices < length]
    return indices


if __name__ == '__main__':
    print(f"omega: {omega_l}")
    run(total_iterations, save_to_disk=True, interval=100,
        continue_last=False)
```

A.5 GRAETZ PROBLEM AND THE LEVEQUE APPROXIMATION

Similar to the previous code, this compares the numerical current density with the analytical expression from Leveque's approximation.

```
import matplotlib.pyplot as plt
import numpy as np
import torch
from PIL import Image
from util import *
import threading
import queue
from alive_progress import alive_bar
```

```
import time

# Physical constants
cell_size_ph = 5e-2  # 5cm or 0.05m
cell_depth_ph = 2e-3  # 2mm or 0.002m
concentration_ph = 100  # mol/m^3 or 0.1M
vel_ph = 0.01  # m/s

z = 1  # Number of electrons transferred
d_ph = 0.76e-7  # m^2/s Diffusion coefficient (Bard page 813)


electrode = generate_electrode_tensor("input/levequev2.png")
obstacle = generate_obstacle_tensor("input/levequev2.png")
nx, ny = obstacle.shape
v_field = torch.from_numpy(np.load("output/BaseLattice_last_u.npy")).to(device)

"""Simulation parameters"""
delay = 200  # Delay before applying voltage
total_iterations = 50000
# Diffusion coefficient
omega_l = 1.96
w_e = omega_l
lambda_trt = 1/12
w_o = 1 / (lambda_trt / (1/w_e - 0.5) + 0.5)
# w_o = 0.6
# lambda_trt = (1/w_e - 0.5) * (1/w_o - 0.5)
pe, dx, dt, d_l, u_l = convert_from_physical_params_diff(cell_size_ph, ny - 2, vel_ph, d_ph, nx, omega_l)
print(f"lambda_trt: {lambda_trt}")
print(f"omega_e: {w_e}, omega_o: {w_o}")
tau = 1 / omega_l
j_log = torch.zeros(1, dtype=torch.float64, device=device)
v_field = v_field / torch.max(v_field) * u_l
input("Press enter to start...")

"""Initialization"""
# Initialize scalar field for species concentration
rho_ox = torch.zeros((nx, ny), dtype=torch.float64, device=device)
rho_red = torch.ones((nx, ny), dtype=torch.float64, device=device)
rho_ox[obstacle] = 0
rho_red[obstacle] = 0
```

```
feq_ox = torch.zeros((9, nx, ny), dtype=torch.float64,
    device=device)
feq_red = torch.zeros((9, nx, ny), dtype=torch.float64,
    device=device)


def equilibrium():
    global feq_ox, feq_red
    cu = torch.einsum('ixy,ji->jxy', v_field, c)
    # Calculate equilibrium populations (Kruger page 304)
    feq_ox = w.view(9, 1, 1) * rho_ox * (1 + 3 * cu)
    feq_red = w.view(9, 1, 1) * rho_red * (1 + 3 * cu)


equilibrium()

fin_ox = feq_ox.clone()
fin_red = feq_red.clone()

fout_ox = feq_ox.clone()
fout_red = feq_red.clone()


def step(i):
    global fin_ox, fin_red, fout_ox, fout_red, rho_ox,
        rho_red
    # Perform one LBM step
    # Outlet BC
    # Equiv. to neumann BC on concentration (null flux)
    fin_ox[left_col, -1, :] = fin_ox[left_col, -2, :]
    fin_red[left_col, -1, :] = fin_red[left_col, -2, :]

    macroscopic()

    if i > delay:
        # Calculate generated current
        # Charge
        q = F * (rho_red[electrode] * concentration_ph) * (
            dx ** 2) * cell_depth_ph
        # Current
        j_log[i - delay] = q

        # Electrode BC
        rho_ox[electrode] = 1
        rho_red[electrode] = 0
```

```
        equilibrium()

        # Zhou He BC
        fin_ox[right_col, 0, :] = feq_ox[right_col, 0, :] +
            fin_ox[left_col, 0, :] - feq_ox[left_col, 0, :]
        fin_red[right_col, 0, :] = feq_red[right_col, 0, :] +
            fin_red[left_col, 0, :] - feq_red[left_col, 0, :]

        # BGK collision
        # Collision
        f_plus = .5 * (fin_ox + fin_ox[c_op])
        f_minus = .5 * (fin_ox - fin_ox[c_op])

        feq_plus = .5 * (feq_ox + feq_ox[c_op])
        feq_minus = .5 * (feq_ox - feq_ox[c_op])

        fout_ox = fin_ox - w_o * (f_plus - feq_plus) - w_e * (
            f_minus - feq_minus)

        g_plus = .5 * (fin_red + fin_red[c_op])
        g_minus = .5 * (fin_red - fin_red[c_op])

        geq_plus = .5 * (feq_red + feq_red[c_op])
        geq_minus = .5 * (feq_red - feq_red[c_op])

        fout_red = fin_red - w_o * (g_plus - geq_plus) - w_e * (
            g_minus - geq_minus)

        # Bounce-back
        fout_ox[:, obstacle] = fin_ox[c_op][:, obstacle]
        fout_red[:, obstacle] = fin_red[c_op][:, obstacle]

        # Streaming
        stream(fin_ox, fout_ox)
        stream(fin_red, fout_red)


    def run(iterations: int, save_to_disk: bool = True, interval
        : int = 100, continue_last: bool = False):
        # Launches LBM simulation and a parallel thread for
            saving data to disk
        global rho_ox, rho_red, fin_ox, fin_red, fout_ox,
            fout_red, j_log, dt

        print(f"Simulating {iterations * dt} seconds")
```

```
    j_log = torch.zeros((iterations, electrode.sum()), dtype
        =torch.float64, device=device)

    if continue_last:  # Continue last computation
        rho_ox = torch.from_numpy(np.load("output/
            Electrochemical_last_rho_ox.npy")).to(device)
        rho_red = torch.from_numpy(np.load("output/
            Electrochemical_last_rho_red.npy")).to(device)
        equilibrium()
        fin_ox = feq_ox.clone()  # Initialize incoming
            populations (pre-collision)
        fin_red = feq_red.clone()  # Initialize incoming
            populations (pre-collision)
        fout_ox = feq_ox.clone()  # Initialize outgoing
            populations (post-collision)
        fout_red = feq_red.clone()  # Initialize outgoing
            populations (post-collision)

    if save_to_disk:
        # Create queue for saving data to disk
        q = queue.Queue()
        # Create thread for saving data
        t = threading.Thread(target=save_data, args=(q,))
        t.start()

    # Run LBM for specified number of iterations
    with alive_bar(iterations, force_tty=True) as bar:
        start = time.time()
        counter = 0
        for i in range(iterations):
            step(i)  # Perform one LBM step
            if i % interval == 0:
                # Calculate MLUPS by dividing number of
                    nodes by time in seconds
                delta_t = time.time() - start
                mlups = nx * ny * counter / (delta_t * 1e6)
                if save_to_disk:
                    # push data to queue
                    q.put((rho_red.detach().clone(), f"
                        output/{i // interval:05}.png"))  #
                        Five digit filename
                # Reset timer and counter
                start = time.time()
                counter = 0

            counter += 1
```

```
            bar.text(f"MLUPS: {mlups:.2f} | Electrode
                density {rho_ox[electrode].mean().cpu().
                numpy():.5f}")
            bar()

    # Save final data to numpy files
    np.save(f"output/Electrochemical_last_rho_ox.npy",
        rho_ox.cpu().numpy())
    np.save(f"output/Electrochemical_last_rho_red.npy",
        rho_red.cpu().numpy())

    # Plot current density
    plt.show()
    fig, ax = plt.subplots()
    # x = np.linspace(1e-1, electrode.sum().cpu().numpy(),
        1000)
    # lev = 0.67 * (d_l**(2/3)) * (u_l/(ny*x))**(1/3)
    x = np.arange(1, electrode.sum().cpu().numpy() + 1)
    sfc = cell_depth_ph * electrode.sum().cpu().numpy() * dx
    avg_vel = torch.mean(v_field).cpu().numpy()
    h = (ny - 3) * dx
    lev = 0.9783 * (F * d_ph * concentration_ph / sfc) * (
        avg_vel * vel_ph / (h * d_ph * x * dx)) ** (1 / 3)
    lbm_sol = F * (j_log[iterations - delay - 1] *
        concentration_ph) * dx / dt
    plt.plot(x/x.size, lbm_sol.cpu().numpy() / np.mean(
        lbm_sol.cpu().numpy()), 'gs', label="LBM")
    plt.plot(x/x.size, lev / np.mean(lev), 'r--', label='
        Analytical')
    ax.set_ylabel(r"$j / j_{avg}$")
    ax.set_xlabel(r"$x / L$")
    # plt.xlim([0, 1])
    # plt.ylim([0, torch.max(j_log[iterations - delay - 1]).
        cpu().numpy() * 1.1])
    ax.legend()
    plt.savefig("output/leveque_current.png", bbox_inches='
        tight', pad_inches=0, dpi=900)
    plt.show()

    if save_to_disk:
        # Stop thread for saving data
        q.put((None, None))
        t.join()


def save_data(q: queue.Queue):
```

```
    while True:
        data, filename = q.get()
        if data is None:
            break
        plt.clf()
        plt.axis('off')
        data[obstacle] = np.nan
        plt.imshow(data.cpu().numpy().transpose(), cmap=cmap
            , vmin=0, vmax=1)
        plt.colorbar()
        plt.savefig(filename, bbox_inches='tight',
            pad_inches=0, dpi=500)
        plt.close()


if __name__ == '__main__':
    print(f"omega: {omega_l}")
    run(total_iterations, save_to_disk=True, interval=1000,
        continue_last=False)
```

A.6 VARIOUS UTILITY FUNCTIONS

The following are a set of functions and variables that were useful throughout the codebase. These functions range from tools to interface between different parts of the model, to function that perform unit conversion between lattice units and physical units, as well as repetitive tasks such as performing common routines for data vizualisation and LBM-specific operations such as collision, calculating equilibrium distributions or streaming.

```
import numpy as np
import torch
from PIL import Image
import matplotlib.pyplot as plt
import queue

plt.rcParams.update({
    "text.usetex": True,
    "font.family": "Computer Modern"
})

BLACK = np.asarray([0, 0, 0])
WHITE = np.asarray([255, 255, 255])
RED = np.asarray([255, 0, 0])
```

```
GREEN = np.asarray([0, 255, 0])
BLUE = np.asarray([0, 0, 255])
YELLOW = np.asarray([255, 255, 0])
CYAN = np.asarray([0, 255, 255])
MAGENTA = np.asarray([255, 0, 255])

device = torch.device("cuda" if torch.cuda.is_available()
    else "cpu")
print(f"using device: {device}")

cmap = plt.get_cmap('coolwarm')
cmap.set_bad((0, 0, 0, 1))

# Physical constants
T = 298  # Kelvin
R = 8.3145  # J/mol.K
F = 96485  # C/mol

"""
Outgoing directions:
6---2---5
| \ | / |
3---0---1
| / | \ |
7---4---8
Lattice parameters:
"""
c = torch.tensor([[0, 0], [1, 0], [0, 1], [-1, 0], [0, -1],
    [1, 1], [-1, 1], [-1, -1], [1, -1]],
                 device=device).float()
c_op = torch.tensor([0, 3, 4, 1, 2, 7, 8, 5, 6], device=
    device)  # Opposite directions indices
w = torch.tensor([4 / 9, 1 / 9, 1 / 9, 1 / 9, 1 / 9, 1 / 36,
     1 / 36, 1 / 36, 1 / 36], device=device)  # weights
right_col = [1, 5, 8]  # Right column of velocities
left_col = [3, 7, 6]  # Left column of velocities (order is
    important, see line 85 equilibrium function in inlet)
center_col = [0, 2, 4]  # Center column of velocities

top_row = [2, 6, 5]
center_row = [0, 1, 3]
bottom_row = [4, 8, 7]


def generate_obstacle_tensor(file):
    # Generate obstacle tensor from image file
```

```
    img_array = np.asarray(Image.open(file))
    # Black pixels are True, white pixels are False
    if img_array.shape[2] == 4:
        img_array = img_array[:, :, :3]
    obstacle_solid = (img_array == BLACK).all(axis=2).T
    obstacle_electrode = (img_array == BLUE).all(axis=2).T
    obstacle = torch.tensor(obstacle_solid, dtype=torch.bool
        ).to(device)
    return obstacle


def generate_electrode_tensor(file, color=BLUE):
    # Generate electrode tensor from image file
    img_array = np.asarray(Image.open(file))
    # Black pixels are True, white pixels are False
    if img_array.shape[2] == 4:
        img_array = img_array[:, :, :3]
    electrode = (img_array == color).all(axis=2).T
    electrode = torch.tensor(electrode, dtype=torch.bool).to
        (device)
    return electrode


def save_data(q: queue.Queue, obstacle):
    # Save data to disk by running a separate thread that
        gets data from a queue
    while True:
        data, filename = q.get()
        if data is None:
            break

        # Preprocessing before plotting
        velocity = torch.sqrt(data[0][0] ** 2 + data[0][1]
            ** 2)  # module of velocity
        # velocity /= inlet_vel  # normalize
        density = data[1]
        velocity[obstacle] = np.nan
        density[obstacle] = np.nan

        # Plot both macroscopic variables
        fig, (ax0, ax1) = plt.subplots(2, 1)
        cax0 = ax0.imshow(velocity.cpu().numpy().transpose()
            , cmap=cmap)
        cax1 = ax1.imshow(density.cpu().numpy().transpose(),
             cmap=cmap)
        ax0.set_title(r"lattice velocity $u$")
```

```
        ax1.set_title(r"density $\rho$")
        ax0.axis("off")
        ax1.axis("off")
        fig.colorbar(cax0, ax=ax0)
        fig.colorbar(cax1, ax=ax1)
        plt.savefig(filename, bbox_inches='tight',
            pad_inches=0, dpi=600)
        plt.close(fig)


def convert_to_physical_velocity(velocity_array, dx, dt):
    # Convert lattice velocity to physical velocity
    # C_factor * lattice_velocity = physical_velocity
    conversion_factor = dx / dt  # lattice velocity to
        physical velocity
    return velocity_array * conversion_factor


def convert_to_lattice_velocity(velocity_array, dx, dt):
    return velocity_array / (dx / dt)


def get_lattice_viscosity_from_tau_l(tau):
    """Kruger page 272"""
    return (1 / 3) * (tau - .5)


def convert_from_physical_params_ns(total_length_ph,
    channel_width_ph, char_velocity_ph, viscosity_ph,
                                        lattice_size, omega_l):
    # Kruger page 283 example
    re = channel_width_ph * char_velocity_ph / viscosity_ph
    dx = total_length_ph / lattice_size
    inlet_width_l = channel_width_ph / dx

    tau_l = 1 / omega_l
    nu_l = get_lattice_viscosity_from_tau_l(tau_l)

    dt = (1 / 3) * ((tau_l - 0.5) * dx ** 2) / viscosity_ph
    ulb = char_velocity_ph / (dx / dt)  # C_factor *
        lattice_velocity = physical_velocity
    assert np.abs(ulb - (re * nu_l / inlet_width_l)) < 1e-5
    print("Simulation parameters (Navier-Stokes):")
    print(f"Re: {re}")
    print(f"dt: {dt}")
    print(f"dx: {dx}")
```

```
138     print(f"ulb: {ulb}")
        print(f"omega_l: {omega_l}")
        print(f"nu_l: {nu_l}")
        return re, dx, dt, ulb


143
    def convert_from_physical_params_diff(total_length_ph,
        channel_width_ph, char_velocity_ph, diffusion_coeff_ph,
                                              lattice_size, omega_l)
                                                  :
        pe = char_velocity_ph * channel_width_ph /
            diffusion_coeff_ph
        dx = total_length_ph / lattice_size
148     inlet_width_l = channel_width_ph / dx

        tau_l = 1 / omega_l
        dt = (1 / 3) * ((tau_l - 0.5) * dx ** 2) /
            diffusion_coeff_ph
        u_l = char_velocity_ph / (dx / dt)
153     d_l = get_lattice_viscosity_from_tau_l(tau_l)

        print("Simulation parameters (Diffusion):")
        print(f"Pe: {pe}")
        print(f"dt: {dt}")
158     print(f"dx: {dx}")
        print(f"u_l: {u_l}")
        print(f"omega_l: {omega_l}")
        print(f"d_l: {d_l}")
        return pe, dx, dt, d_l, u_l
163

    def convert_from_physical_params_pure_diff(total_length_ph,
        diffusion_coeff_ph, lattice_size, omega_l,
                                                 total_iterations)
                                                     :
        dx = total_length_ph / lattice_size
168

        tau_l = 1 / omega_l
        dt = (1 / 3) * ((tau_l - 0.5) * dx ** 2) /
            diffusion_coeff_ph
        d_l = get_lattice_viscosity_from_tau_l(tau_l)

173     fo = d_l * total_iterations / lattice_size  # Fourier
            number
        print("Simulation parameters (Diffusion):")
        print(f"Fo: {fo}")
```

```
    print(f"dt: {dt}")
    print(f"dx: {dx}")
    print(f"omega_l: {omega_l}")
    print(f"d_l: {d_l}")
    return fo, dx, dt, d_l


def poiseuille_inlet(vmax, n):
    y = torch.arange(n, device=device).float()
    v = vmax * (4 / n) * y * (1 - y / n)
    return v


def stream(fin, fout):
    global nx, ny
    fin[1, 1:, :] = fout[1, :nx - 1, :]  # vel 1 increases x
    # fin[1, 0, :] = fout[1, -1, :]  # wrap
    fin[3, :nx - 1, :] = fout[3, 1:, :]  # vel 3 decreases x
    # fin[3, -1, :] = fout[3, 0, :]  # wrap

    fin[2, :, 1:] = fout[2, :, :ny - 1]  # vel 2 increases y
    # fin[2, :, 0] = fout[2, :, -1]  # wrap
    fin[4, :, :ny - 1] = fout[4, :, 1:]  # vel 4 decreases y
    # fin[4, :, -1] = fout[4, :, 0]  # wrap

    # vel 5 increases x and y simultaneously
    fin[5, 1:, 1:] = fout[5, :nx - 1, :ny - 1]
    # fin[5, 0, :] = fout[5, -1, :]  # wrap right
    # fin[5, :, 0] = fout[5, :, -1]  # wrap top
    # vel 7 decreases x and y simultaneously
    fin[7, :nx - 1, :ny - 1] = fout[7, 1:, 1:]
    # fin[7, -1, :] = fout[7, 0, :]  # wrap left
    # fin[7, :, -1] = fout[7, :, 0]  # wrap bottom

    # vel 6 decreases x and increases y
    fin[6, :nx - 1, 1:] = fout[6, 1:, :ny - 1]
    # fin[6, -1, :] = fout[6, 0, :]  # wrap left
    # fin[6, :, 0] = fout[6, :, -1]  # wrap top
    # vel 8 increases x and decreases y
    fin[8, 1:, :ny - 1] = fout[8, :nx - 1, 1:]
    # fin[8, 0, :] = fout[8, -1, :]  # wrap right
    # fin[8, :, -1] = fout[8, :, 0]  # wrap bottom

    fin[0, :, :] = fout[0, :, :]
```

B

APPENDIX: SYNTHETIC PORE MICROSTRUCTURE GENERATION

In this appendix, we lay out the source of the code that was used to generate the synthetic microstructures that were used to model porous electrodes:

B.1 STOCHASTIC PORE MICROSTRUCTURE GENERATION

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
# from scipy.ndimage import gaussian_filter
from scipy.stats import norm
from scipy.stats import multivariate_normal
from scipy.signal import convolve2d
from PIL import Image
from util import *

cmap = matplotlib.colormaps['Greys']
display_interface = False  # Colors Electrodes sfc in blue
    for identification in LBM obstacle

nx, ny = 1344, 130  # domain dimensions
target_rho = 0.8


# Custom gaussian kernel with multivariate normal
    distribution and custom covariance matrix
def gaussian_kernel(size: int, mean: float, cov: np.ndarray,
     angle: float = 0) -> np.ndarray:
    """Creates a 2D Gaussian kernel with given parameters.
        """
    # Create 2D coordinates
    x = np.linspace(-3, 3, size)
    y = np.linspace(-3, 3, size)
    x, y = np.meshgrid(x, y)

    # Mean vector
    mean_vec = np.array([0, 0])
```

```
    # Create the rotation matrix
    rotation_matrix = np.array([[np.cos(angle), -np.sin(
        angle)],
                                [np.sin(angle),  np.cos(
                                    angle)]])

    # Rotate the covariance matrix
    cov_rotated = rotation_matrix @ cov @ rotation_matrix.T

    # Create multivariate normal distribution
    dist = multivariate_normal(mean_vec, cov_rotated)

    # Evaluate the PDF on the grid and reshape it into 2D
    kernel = dist.pdf(np.dstack([x, y]))

    return kernel


# kernel parameters
size = 101  # Size of the kernel. Should be odd, to have a
    center pixel
mean = 0   # Mean. Should be 0 for a centered kernel
cov = np.asarray([[1, 0],
                  [0, 1]])  # Covariance matrix

# Generate the kernel
kernel = gaussian_kernel(size, mean, .1 * cov, angle=torch.
    pi / 2)

plt.imshow(kernel, interpolation='none')
plt.axis('off')
# plt.savefig(f'output/gaussian_kernel_piby2.png',
    bbox_inches='tight', pad_inches=0, dpi=600)
plt.show()
# exit()
# Generate random noise matrix from normal distribution
noise = np.random.randn(nx, ny)

print("Applying Gaussian Smoothing..")
# Smooth it with gaussian kernel
smooth = convolve2d(noise, kernel, mode='same', boundary='
    symm')

# Calculate cutoff for given target porosity using percent
    point function
```

```
cutoff = norm.ppf(target_rho, loc=np.mean(smooth), scale=np.
    std(smooth))

# Generate boolean matrix
final = smooth > cutoff
print(f'Porosity: {1 - np.sum(final) / (nx * ny)}')

# plt.imshow(final.T, cmap=cmap, interpolation='none')
# plt.axis('off')
# plt.savefig(f'output/electrode.png', bbox_inches='tight',
    pad_inches=0, dpi=800)
# plt.show()

interface = np.full_like(final, False)
# # identify interface
print("Finding interface..")
for i in range(1, nx - 1):
    for j in range(1, ny - 1):
        neighborhood = final[i - 1: i + 1, j - 1: j + 1]
        if not (neighborhood.all() or not neighborhood.any()
            ):
            interface[i, j] = True

# Obstacles on Boundaries are considered electrode surface
# interface[0, final[0]] = True
# interface[-1, final[-1]] = True
interface[final[:, 0], 0] = True
interface[final[:, -1], -1] = True

print("Drawing Image..")
rgb_array = 255 * np.ones((nx, ny, 3))
rgb_array[final, :] = np.asarray(BLACK)
rgb_array[interface, :] = np.asarray(BLUE)
image = Image.fromarray(rgb_array.transpose(1, 0, 2).astype(
    np.uint8), mode='RGB')
image.save(f'output/temp.png')
print("Done.")
```

B.2 PRESSURE DROP CALCULATION ACROSS PORE SPACE

```
import matplotlib.pyplot as plt
import torch
import matplotlib as mpl
import threading
import queue
```

```
from PIL import Image
import numpy as np
from alive_progress import alive_bar
import time
from util import *
# To Generate ffmpeg video from images
# ffmpeg -f image2 -framerate 30 -i %05d.png -s 1080x720 -pix_fmt yuv420p output.mp4


"""Simulation parameters"""
# Create obstacle tensor from numpy array
obstacle = generate_obstacle_tensor('input/permeability/rho_0_75_sig_0_01.png')
obstacle = obstacle.clone().to(device)
nx, ny = obstacle.shape  # Number of nodes in x and y directions
re = 20  # Reynolds number
ulb = 0.001  # characteristic velocity (inlet)
nulb = ulb * ny / re  # kinematic viscosity
print("nulb:", nulb)
omega = 1 / (3 * nulb + 0.5)  # relaxation parameter


# Initialize macroscopic variables
rho = torch.ones((nx, ny), device=device).float()
u = torch.zeros((2, nx, ny), device=device).float()

# Initialize populations
feq = torch.zeros((9, nx, ny), device=device).float()
equilibrium()  # Initialize equilibrium populations
fin = feq.clone()  # Initialize incoming populations (pre-collision)
fout = feq.clone()  # Initialize outgoing populations (post-collision)


def run(iterations: int, save_to_disk: bool = True, interval: int = 100, continue_last: bool = False):
    # Launches LBM simulation and a parallel thread for saving data to disk

    global rho, u, fin, fout
    if continue_last:  # Continue last computation
        rho = torch.from_numpy(np.load("output/PdropLattice_last_rho.npy")).to(device)
```

```
        u = torch.from_numpy(np.load("output/
            PdropLattice_last_u.npy")).to(device)
        equilibrium()
        fin = feq.clone()  # Initialize incoming populations
             (pre-collision)
        fout = feq.clone()  # Initialize outgoing
            populations (post-collision)

    if save_to_disk:
        # Create queue for saving data to disk
        q = queue.Queue()
        # Create thread for saving data
        t = threading.Thread(target=save_data, args=(q,
            obstacle))
        t.start()

    # Run LBM for specified number of iterations
    with alive_bar(iterations, force_tty=True) as bar:
        start = time.time()
        counter = 0
        for i in range(iterations):
            step()  # Perform one LBM step
            if i % interval == 0:
                # Calculate MLUPS by dividing number of
                    nodes by time in seconds
                dt = time.time() - start
                mlups = nx * ny * counter / (dt * 1e6)
                if save_to_disk:
                    # push data to queue
                    q.put(((u, rho), f"output/{i // interval
                        :05}.png"))  # Five digit filename
                # Reset timer and counter
                start = time.time()
                counter = 0

            counter += 1
            ddp[i] = torch.mean(rho[1, :]) - torch.mean(rho
                [-2, :])
            bar.text(f"MLUPS: {mlups:.2f}, DDP: {ddp[i]:.7f}
                ")
            bar()

    # Save final data to numpy files
    np.save(f"output/PdropLattice_last_u.npy", u.cpu().numpy
        ())
```

```
        np.save(f"output/PdropLattice_last_rho.npy", rho.cpu().
            numpy())

    if save_to_disk:
        # Stop thread for saving data
        q.put((None, None))
        t.join()


if __name__ == '__main__':
    print(f"Device: {device}")
    print(f"omega: {omega}")
    iterations = 25000
    ddp = torch.zeros(iterations, device=device)  # Keep
        track of pressure drop evolution
    run(iterations, save_to_disk=True, interval=1000,
        continue_last=False)

    plt.clf()
    plt.plot(np.arange(iterations), ddp.cpu().numpy(), label
        ="Pressure drop")  # Plot pressure drop evolution
    plt.show()
    print("Pressure drop:")
    print(ddp[-1].item())

    # Calculate permeability
    # mean velocity at outlet
    L = 300
    u_mean = torch.mean(u[0, -1, :])
    k = u_mean * nulb * L / ddp[-1].item()
    print(f"Permeability: {k.cpu().numpy()}")
```

B.3 TORTUOSITY AND STREAMLINES

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.linalg import norm
from scipy.interpolate import RegularGridInterpolator
from util import *

v = np.load("output/PdropLattice_last_u.npy")
obstacle = generate_obstacle_tensor("input/permeability/rho
    0_75 sig 0_01.png").cpu().numpy()
left_margin = 10
v[:, obstacle] = 0
```

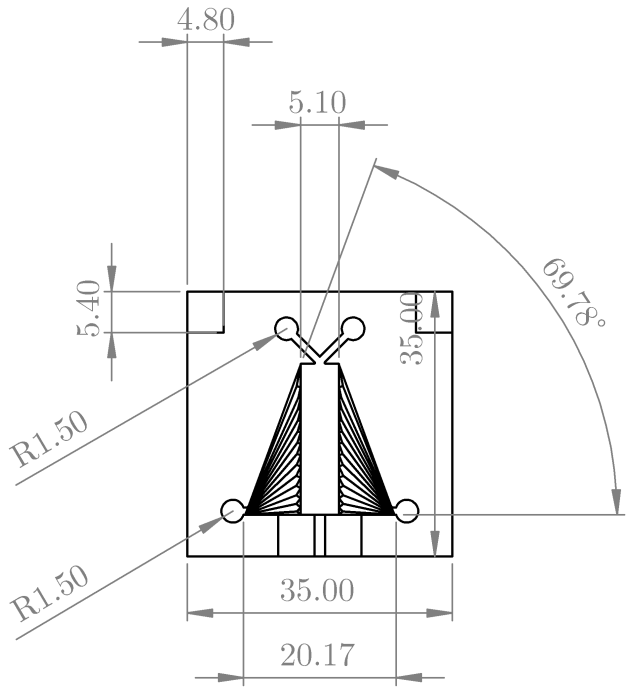
```
U = v[0]
V = v[1]
x = np.arange(v.shape[1])
y = np.arange(v.shape[2])

# Interpolation of vector field 244 74
U_interp = RegularGridInterpolator((x, y), U)
V_interp = RegularGridInterpolator((x, y), V)


def func(point):
    return [U_interp(point)[0], V_interp(point)[0]]


# plotting
fig, ax = plt.subplots()
# ax.streamplot(x, y, U.T, V.T)

paths = []
for i in range(0, v.shape[2], 5):
    # initial seed
    y0 = np.asarray([v.shape[1] - 10, i])  # Start from last
        element on the right (outflow) for each y

    # initialize path
    path = [y0]

    while True:
        t = -100  # step backwards
        # Calculate next step from previous step using euler
            method
        try:
            next_step = path[-1] + np.asarray(func(path[-1])
                ) * t
        except ValueError:
            print("ValueError")
            break
        path.append(next_step)
        if path[-1][0] < left_margin:  # if reached left
            boundary
            paths.append(path)
            # print("reached end")
            break
        if np.abs(path[-1][1] - 0) < 1 or np.abs(path[-1][1]
            - v.shape[2]) < 1:  # if reached top boundary
            print("reached top/bot boundary")
```

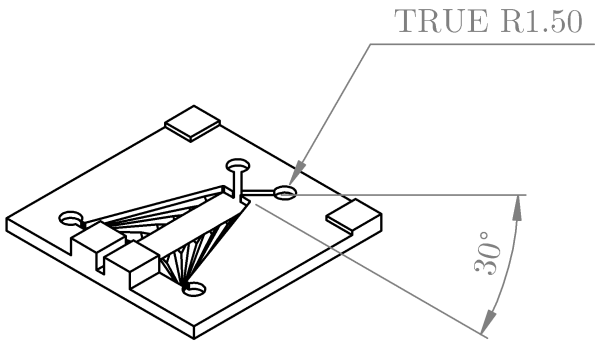
```
            break
        if norm(path[-1] - path[-2]) < 1e-16:  # If reached
            end of streamline (v = 0)
            print("reached velocity ~zero")
            break
        if len(path) > 100000:
            print("reached max length")
            break
        print(f"Progress: {int(i * 100 / v.shape[2])}%, \
            tCurrent x: {path[-1][0]}\tCurrent y: {path
            [-1][1]}", sep='', end="\r", flush=True)

# Calculate lengths of streamlines
print("Calculating lengths...")
lengths = []
for path in paths:
    path_array = np.asarray(path)
    dxdt = np.gradient(path_array[:, 0])
    dydt = np.gradient(path_array[:, 1])
    t_path = np.linspace(0, v.shape[1], len(path))
    length = np.sum(np.sqrt(dxdt ** 2 + dydt ** 2))
    lengths.append(length)

for path in paths:
    ax.plot(np.asarray(path)[:, 0], np.asarray(path)[:, 1],
        'r--', linewidth=0.2)

plt.imshow(np.invert(obstacle.T), cmap='gray')
plt.savefig("output/Streamlines_rho_0_75_sig_0_01.png", dpi
    =1200)

lengths = np.asarray(lengths)
tortuosity = np.mean(lengths) / (v.shape[1] - left_margin -
    1)  # -1 for right margin

print(f"Measured Tortuosity: {tortuosity}")

# flux averaged streamlines tortuosity
# calculate average module of velocity, excluding obstacle
v_pores = v[:, ~obstacle]
v_pores_module = np.sqrt(v_pores[0] ** 2 + v_pores[1] ** 2)
# average velocity in pores
v_mean = np.mean(v_pores_module)
# average x component of velocity
v_x_mean = np.mean(v_pores[0])
```

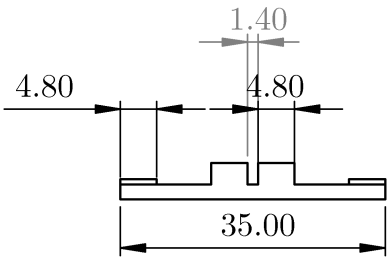
```
flux_avg_tortuosity = v_mean / v_x_mean
print(f"Flux averaged tortuosity: {flux_avg_tortuosity}")
```

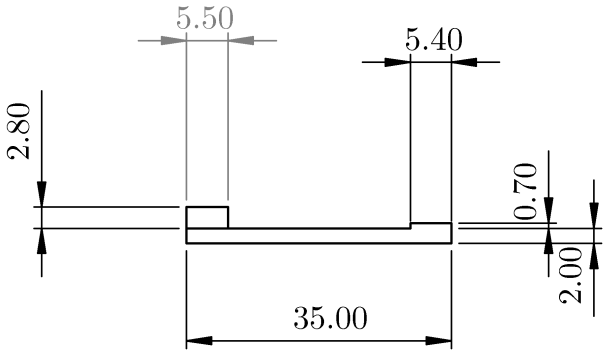
```
import numpy as np
import matplotlib.pyplot as plt
from util import *

v = np.load("output/BaseLattice_last_u.npy")
obstacle = generate_obstacle_tensor("input/tortuosity/
    pdrop_08sig.png").cpu().numpy()
v[:, obstacle] = 0

d, nx, ny = v.shape
x = np.arange(nx)
y = np.arange(ny)
X, Y = np.meshgrid(x, y)
#
U = np.transpose(v[0, :, :])
V = np.transpose(v[1, :, :])
M = np.hypot(U, V)
magnitude = np.sqrt(U ** 2 + V ** 2)
n = 5
plt.imshow(obstacle.T, cmap='Greys')
plt.streamplot(X, Y, U, V, density=2, linewidth=.2,
    arrowstyle='-', color=magnitude, cmap=cmap)

plt.savefig('output/test.png', dpi=1500)
# plt.show()
```

B.4 CAD TECHNICAL DRAWINGS









UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS TOLERANCES: 1mm			DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION	10
	NAME	SIGNATURE	DATE		TITLE: channel piece with grooves and low profile aligners		
DRAWN	Youssef Kharchouf		06/10/23				
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL: CLEAR V4	DWG NO. 1		A4
					SCALE:1:1	SHEET 1 OF 1	

BIBLIOGRAPHY

[1] Satyanarayana Maddukuri, David Malka, Munseok S. Chae, Yuval Elias, Shalom Luski, and Doron Aurbach. "On the challenge of large energy storage by electrochemical devices." In: *Electrochimica Acta* 354 (Sept. 2020), p. 136771. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2020.136771.

[2] Adam Z. Weber, Matthew M. Mench, Jeremy P. Meyers, Philip N. Ross, Jeffrey T. Gostick, and Qinghua Liu. "Redox flow batteries: A review." In: *Journal of Applied Electrochemistry* 41 (10 Sept. 2011). 2011 conventional RFB review
, pp. 1137–1164. ISSN: 0021891X. DOI: 10.1007/S10800-011-0348-2/FIGURES/15. URL: https://link.springer.com/article/10.1007/s10800-011-0348-2.

[3] Piergiorgio Alotto, Massimo Guarnieri, and Federico Moro. "Redox flow batteries for the storage of renewable energy: A review." In: (2013). DOI: 10.1016/j.rser.2013.08.001. URL: http://dx.doi.org/10.1016/j.rser.2013.08.001.

[4] Grigorii L. Soloveichik. "Flow Batteries: Current Status and Trends." In: *Chemical Reviews* 115 (20 Oct. 2015), pp. 11533–11558. ISSN: 15206890. DOI: 10.1021/CR500720T/ASSET/CR500720T.FP.PNG_V03. URL: https://pubs.acs.org/doi/abs/10.1021/cr500720t.

[5] R Ye, D Henkensmeier, SJ Yoon Journal of . . ., and undefined 2018. "Redox flow batteries for energy storage: a technology review." In: *asmedigitalcollection.asme.orgR Ye, D Henkensmeier, SJ Yoon, Z Huang, DK Kim, Z Chang, S Kim, R ChenJournal of Electrochemical Energy Conversion and Storage, 2018•asmedigitalcollection.asme.org* (2018). URL: https://asmedigitalcollection.asme.org/electrochemical/article-abstract/15/1/010801/444047.

[6] SH Shin, SH Yun, SH Moon Rsc Advances, and undefined 2013. "A review of current developments in non-aqueous redox flow batteries: characterization of their membranes for design perspective." In: *pubs.rsc.orgSH Shin, SH Yun, SH MoonRsc Advances, 2013•pubs.rsc.org* 3 (2013), p. 9095.

DOI: 10.1039/c3ra00115f. URL: https://pubs.rsc.org/en/content/articlehtml/2013/ra/c3ra00115f.

[7] V. M. Ortiz-Martínez, L. Gómez-Coma, G. Pérez, A. Ortiz, and I. Ortiz. "The roles of ionic liquids as new electrolytes in redox flow batteries." In: *Separation and Purification Technology* 252 (Dec. 2020), p. 117436. ISSN: 1383-5866. DOI: 10.1016/J.SEPPUR.2020.117436.

[8] Douglas R. Macfarlane, Naoki Tachikawa, Maria Forsyth, Jennifer M. Pringle, Patrick C. Howlett, Gloria D. Elliott, James H. Davis, Masayoshi Watanabe, Patrice Simon, and C. Austen Angell. "Energy applications of ionic liquids." In: *Energy and Environmental Science* 7 (1 Dec. 2013), pp. 232–250. ISSN: 1754-5706. DOI: 10.1039/C3EE42099J. URL: https://pubs.rsc.org/en/content/articlehtml/2014/ee/c3ee42099jhttps://pubs.rsc.org/en/content/articlelanding/2014/ee/c3ee42099j.

[9] Helen Prifti, Aishwarya Parasuraman, Suminto Winardi, Tuti Mariana Lim, and Maria Skyllas-Kazacos. "Membranes for Redox Flow Battery Applications." In: *Membranes 2012, Vol. 2, Pages 275-306* 2 (2 June 2012), pp. 275–306. ISSN: 2077-0375. DOI: 10.3390/MEMBRANES2020275. URL: https://www.mdpi.com/2077-0375/2/2/275/htmhttps://www.mdpi.com/2077-0375/2/2/275.

[10] Gareth Kear, Akeel A. Shah, and Frank C. Walsh. "Development of the all-vanadium redox flow battery for energy storage: A review of technological, Financial and policy aspects." In: *International Journal of Energy Research* 36 (11 Sept. 2012), pp. 1105–1120. ISSN: 0363907X. DOI: 10.1002/er.1863.

[11] Musbaudeen O. Bamgbopa, Saif Almheiri, and Hong Sun. "Prospects of recently developed membraneless cell designs for redox flow batteries." In: *Renewable and Sustainable Energy Reviews* 70 (Apr. 2017), pp. 506–518. ISSN: 1364-0321. DOI: 10.1016/J.RSER.2016.11.234.

[12] Omar A. Ibrahim, Marina Navarro-Segarra, Pardis Sadeghi, Neus Sabaté, Juan Pablo Esquivel, and Erik Kjeang. *Microfluidics for Electrochemical Energy Conversion*. Apr. 2022. DOI: 10.1021/acs.chemrev.1c00499.

[13] Nesrine Chaabene, Kieu Ngo, Mireille Turmine, and Vincent Vivier. "Ionic liquid redox flow membraneless battery in microfluidic system." In: *Journal of Energy Storage* 57 (Jan. 2023), p. 106270. ISSN: 2352-152X. DOI: 10.1016/J.EST.2022.106270.

[14] Taehoon Kim, Wentao Song, Dae Yong Son, Luis K. Ono, and Yabing Qi. "Lithium-ion batteries: outlook on present, future, and hybridized technologies." In: *Journal of Materials Chemistry A* 7 (7 Feb. 2019), pp. 2942–2964. ISSN: 2050-7496. DOI: 10.1039/C8TA10513H. URL: https://pubs.rsc.org/en/content/articlehtml/2019/ta/c8ta10513hhttps://pubs.rsc.org/en/content/articlelanding/2019/ta/c8ta10513h.

[15] Chang Liu, Feng Li, Ma Lai-Peng, and Hui Mmg Cheng. "Advanced Materials for Energy Storage." In: *Advanced Materials* 22 (8 Feb. 2010), E28–E62. ISSN: 1521-4095. DOI: 10.1002/ADMA.200903328. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/adma.200903328https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.200903328https://onlinelibrary.wiley.com/doi/10.1002/adma.200903328.

[16] Marcelo Carmo, David L. Fritz, Jürgen Mergel, and Detlef Stolten. "A comprehensive review on PEM water electrolysis." In: *International Journal of Hydrogen Energy* 38 (12 Apr. 2013), pp. 4901–4934. ISSN: 0360-3199. DOI: 10.1016/J.IJHYDENE.2013.01.151.

[17] M. H. Shojaeefard, G. R. Molaeimanesh, M. Nazemian, and M. R. Moqaddari. "A review on microstructure reconstruction of PEM fuel cells porous electrodes for pore scale simulation." In: *International Journal of Hydrogen Energy* 41 (44 Nov. 2016), pp. 20276–20293. ISSN: 0360-3199. DOI: 10.1016/J.IJHYDENE.2016.08.179.

[18] Adam Z. Weber and John Newman. "Modeling transport in polymer-electrolyte fuel cells." In: *Chemical Reviews* 104 (10 Oct. 2004), pp. 4679–4726. ISSN: 00092665. DOI: 10.1021/CR020729L/ASSET/IMAGES/MEDIUM/CR020729LF00014.GIF. URL: https://pubs.acs.org/doi/full/10.1021/cr020729l.

[19] John S. Newman and Charles W. Tobias. "Theoretical Analysis of Current Distribution in Porous Electrodes." In: *Journal of The Electrochemical Society* 109 (12 Dec. 1962),

p. 1183. ISSN: 1945-7111. DOI: 10.1149/1.2425269. URL: https://iopscience.iop.org/article/10.1149/1.2425269https://iopscience.iop.org/article/10.1149/1.2425269/meta.

[20] Aashutosh N. Mistry, Kandler Smith, and Partha P. Mukherjee. "Secondary-Phase Stochastics in Lithium-Ion Battery Electrodes." In: *ACS Applied Materials and Interfaces* 10 (7 Feb. 2018), pp. 6317–6326. ISSN: 19448252. DOI: 10.1021/ACSAMI.7B17771/ASSET/IMAGES/LARGE/AM-2017-17771V_0005.JPEG. URL: https://pubs.acs.org/doi/full/10.1021/acsami.7b17771.

[21] Duo Zhang, Antoni Forner-Cuenca, Oluwadamilola O. Taiwo, Vladimir Yufit, Fikile R. Brushett, Nigel P. Brandon, Sai Gu, and Qiong Cai. "Understanding the role of the porous electrode microstructure in redox flow battery performance using an experimentally validated 3D pore-scale lattice Boltzmann model." In: *Journal of Power Sources* 447 (Jan. 2020). ISSN: 03787753. DOI: 10.1016/J.JPOWSOUR.2019.227249.

[22] Qiang Ma, Qian Xu, Qing Chen, Zhenqian Chen, Huaneng Su, and Weiqi Zhang. "Lattice Boltzmann model for complex transfer behaviors in porous electrode of all copper redox flow battery with deep eutectic solvent electrolyte." In: *Applied Thermal Engineering* 160 (Sept. 2019), p. 114015. ISSN: 1359-4311. DOI: 10.1016/J.APPLTHERMALENG.2019.114015.

[23] Emily M. Ryan and Partha P. Mukherjee. "Mesoscale modeling in electrochemical devices—A critical perspective." In: *Progress in Energy and Combustion Science* 71 (Mar. 2019), pp. 118–142. ISSN: 0360-1285. DOI: 10.1016/J.PECS.2018.11.002.

[24] S. Koohi-Fayegh and M. A. Rosen. "A review of energy storage types, applications and recent developments." In: *Journal of Energy Storage* 27 (Feb. 2020), p. 101047. ISSN: 2352-152X. DOI: 10.1016/J.EST.2019.101047.

[25] Petr Vanýsek and Vítězslav Novák. "Redox flow batteries as the means for energy storage." In: *Journal of Energy Storage* 13 (Oct. 2017), pp. 435–441. ISSN: 2352-152X. DOI: 10.1016/J.EST.2017.07.028.

[26] Wei Wang, Qingtao Luo, Bin Li, Xiaoliang Wei, Liyu Li, and Zhenguo Yang. "Recent progress in redox flow battery research and development." In: *Advanced Functional Materials* 23 (8 Feb. 2013), pp. 970–986. ISSN: 1616301X. DOI: 10.1002/ADFM.201200694.

[27] Changkun Zhang, Leyuan Zhang, Yu Ding, Sangshan Peng, Xuelin Guo, Yu Zhao, Gaohong He, and Guihua Yu. "Progress and prospects of next-generation redox flow batteries." In: *Energy Storage Materials* 15 (Nov. 2018), pp. 324–350. ISSN: 2405-8297. DOI: 10.1016/J.ENSM.2018.06.008.

[28] Puiki Leung, Xiaohong Li, Carlos Ponce De León, Leonard Berlouis, C T John Low, and Frank C Walsh. "Progress in redox flow batteries, remaining challenges and their applications in energy storage." In: *RSC Advances* (2012), pp. 10125–10156. DOI: 10.1039/c2ra21342g. URL: www.rsc.org/advances.

[29] L. H. Thaller. *Electrically rechargeable redox flow cells*. Patent for Fe/Cr cell. 1974.

[30] L. H. Thaller. *Redox flow cell energy storage systems*. Patent for Fe/Cr cell. Feb. 1979.

[31] M. Skyllas-Kazacos and F. Grossmith. "Efficient Vanadium Redox Flow Cell." In: *Journal of The Electrochemical Society* 134 (12 Dec. 1987). original vanadium rfb
, pp. 2950–2953. ISSN: 0013-4651. DOI: 10.1149/1.2100321/XML. URL: https://iopscience.iop.org/article/10.1149/1.2100321https://iopscience.iop.org/article/10.1149/1.2100321/meta.

[32] Chris Menictas and Maria Skyllas-Kazacos. "Performance of vanadium-oxygen redox fuel cell." In: *Journal of Applied Electrochemistry* 41 (10 Oct. 2011). commercial success of VRFB, pp. 1223–1232. ISSN: 0021891X. DOI: 10.1007/S10800-011-0342-8/METRICS. URL: https://link.springer.com/article/10.1007/s10800-011-0342-8.

[33] Bin Li, Qingtao Luo, Xiaoliang Wei, Zimin Nie, Edwin Thomsen, Baowei Chen, Vincent Sprenkle, and Wei Wang. "Capacity decay mechanism of microporous separator-based all-vanadium redox flow batteries and its recovery." In: *ChemSusChem* 7 (2 Feb. 2014), pp. 577–584. ISSN: 1864-564X. DOI: 10.1002/CSSC.201300706. URL: https://pubmed.ncbi.nlm.nih.gov/24488680/.

[34] Qingtao Luo, Liyu Li, Wei Wang, Zimin Nie, Xiaoliang Wei, Bin Li, Baowei Chen, Zhenguo Yang, and Vincent Sprenkle. "Capacity Decay and Remediation of Nafion-based All-Vanadium Redox Flow Batteries." In: *ChemSusChem* 6 (2 Feb. 2013), pp. 268–274. ISSN: 1864-564X. DOI: 10.1002/CSSC.201200730. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/cssc.201200730https://onlinelibrary.wiley.com/doi/abs/10.1002/cssc.201200730https://chemistry-europe.onlinelibrary.wiley.com/doi/pdf/10.1002/cssc.201200730?casa_token=zCcwyEhWlyEAAAAA%3Aq0pisf6BHdjWcZqeR6hUtgeK6cL3X4FWODZRc9I4huCvoYXGM6iAeTq3FGRu71vNqefl7R9Vjhvfd9A.

[35] Robert Remick J and Ang G. P. Peter. *Electrically rechargeable anionically active reduction-oxidation electrical storage-supply system*. Patent for Br/S cell
. Nov. 1984.

[36] K. J. Cathro, K. Cedzynska, and D. C. Constable. "Preparation and performance of plastic-bonded-carbon bromine electrodes." In: *Journal of Power Sources* 19 (4 Apr. 1987). Z/Br cells
, pp. 337–356. ISSN: 0378-7753. DOI: 10.1016/0378-7753(87)87009-X.

[37] Ruben-Simon Kühnel, David Reber, and Corsin Battaglia. "Perspective—Electrochemical Stability of Water-in-Salt Electrolytes." In: *Journal of The Electrochemical Society* 167 (7 Mar. 2020), p. 070544. ISSN: 1945-7111. DOI: 10.1149/1945-7111/AB7C6F. URL: https://iopscience.iop.org/article/10.1149/1945-7111/ab7c6fhttps://iopscience.iop.org/article/10.1149/1945-7111/ab7c6f/meta.

[38] Zhen Li, Sha Li, Suqin Liu, Kelong Huang, Dong Fang, Fengchao Wang, and Sui Peng. "Electrochemical Properties of an All-Organic Redox Flow Battery Using 2,2,6,6-Tetramethyl-1-Piperidinyloxy and N-Methylphthalimide." In: *Electrochemical and Solid State Letters* 14 (12 Nov. 2011). ISSN: 10990062. DOI: 10.1149/2.012112ESL.

[39] Fikile R. Brushett, John T. Vaughey, and Andrew N. Jansen. "An All-Organic Non-aqueous Lithium-Ion Redox Flow Battery." In: *Advanced Energy Materials* 2 (11 Nov. 2012), pp. 1390–1396. ISSN: 1614-6840. DOI: 10.1002/AENM.201200322. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/aenm.201200322https://onlinelibrary.wiley.com/doi/abs/10.1002/aenm.

201200322https://onlinelibrary.wiley.com/doi/10.1002/aenm.201200322.

[40] Yichao Yan, Sophia G. Robinson, Matthew S. Sigman, and Melanie S. Sanford. "Mechanism-Based Design of a High-Potential Catholyte Enables a 3.2 v All-Organic Nonaqueous Redox Flow Battery." In: *Journal of the American Chemical Society* 141 (38 Sept. 2019), pp. 15301–15306. ISSN: 15205126. DOI: 10.1021/JACS.9B07345. URL: https://pubs.acs.org/sharingguidelines.

[41] Yihan Zhen, Cuijuan Zhang, Jiashu Yuan, Yicheng Zhao, and Yongdan Li. "A high-performance all-iron non-aqueous redox flow battery." In: *Journal of Power Sources* 445 (Jan. 2020), p. 227331. ISSN: 0378-7753. DOI: 10.1016/J.JPOWSOUR.2019.227331.

[42] Khashayar Ghandi. "A Review of Ionic Liquids, Their Limits and Applications." In: *Green and Sustainable Chemistry* 2014 (01 Jan. 2014), pp. 44–53. ISSN: 2160-696X. DOI: 10.4236/GSC.2014.41008. URL: http://www.scirp.org/journal/PaperInformation.aspx?PaperID=43349http://dx.doi.org/10.4236/gsc.2014.41008.

[43] J. D. Holbrey and K. R. Seddon. "Ionic Liquids." In: *Clean Products and Processes 1999 1:4* 1 (4 Dec. 1999), pp. 223–236. ISSN: 1618-954X. DOI: 10.1007/S100980050036. URL: https://link.springer.com/article/10.1007/s100980050036.

[44] Alan R. Katritzky, Ritu Jain, Andre Lomaka, Ruslan Petrukhin, Mati Karelson, Ann E. Visser, and Robin D. Rogers. "Correlation of the melting points of potential ionic liquids (imidazolium bromides and benzimidazolium bromides) using the CODESSA program." In: *Journal of chemical information and computer sciences* 42 (2 Mar. 2002), pp. 225–231. ISSN: 0095-2338. DOI: 10.1021/CI0100494. URL: https://pubmed.ncbi.nlm.nih.gov/11911691/.

[45] John M Slattery, Corinne Daguenet, Paul J Dyson, Thomas J S Schubert, Ingo Krossing, J M Slattery, I Krossing, C Daguenet, J Dyson, and T J S Schubert. "How to Predict the Physical Properties of Ionic Liquids: A Volume-Based Approach." In: *Angewandte Chemie* 119 (28 July 2007), pp. 5480–5484. ISSN: 1521-3757. DOI: 10.1002/ANGE.200700941. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/ange.200700941https://

onlinelibrary.wiley.com/doi/abs/10.1002/ange.200700941https://onlinelibrary.wiley.com/doi/10.1002/ange.200700941.

[46] Paul Walden. "Molecular weights and electrical conductivity of several fused salts." In: *Bull. Acad. Imper. Sci.(St. Petersburg)* 1800 (1914).

[47] Philippe Hapiot and Corinne Lagrost. "Electrochemical reactivity in room-temperature ionic liquids." In: *Chemical Reviews* 108 (7 July 2008), pp. 2238–2264. ISSN: 00092665. DOI: 10.1021/CR0680686. URL: https://pubs.acs.org/sharingguidelines.

[48] Natalia V. Plechkova and Kenneth R. Seddon. "Applications of ionic liquids in the chemical industry." In: *Chemical Society Reviews* 37 (1 2008), pp. 123–150. ISSN: 03060012. DOI: 10.1039/B006677J.

[49] John S. Wilkes. "A short history of ionic liquids—from molten salts to neoteric solvents." In: *Green Chemistry* 4 (2 Apr. 2002), pp. 73–80. ISSN: 14639262. DOI: 10.1039/B110838G. URL: https://pubs.rsc.org/en/content/articlehtml/2002/gc/b110838ghttps://pubs.rsc.org/en/content/articlelanding/2002/gc/b110838g.

[50] Dapeng Zhang, Qinghua Liu, Xiaosong Shi, and Yongdan Li. "Tetrabutylammonium hexafluorophosphate and 1-ethyl-3-methyl imidazolium hexafluorophosphate ionic liquids as supporting electrolytes for non-aqueous vanadium redox flow batteries." In: *Journal of Power Sources* 203 (Apr. 2012), pp. 201–205. ISSN: 03787753. DOI: 10.1016/J.JPOWSOUR.2011.10.136.

[51] M. H. Chakrabarti, R. A.W. Dryfe, and E. P.L. Roberts. "Evaluation of electrolytes for redox flow battery applications." In: *Electrochimica Acta* 52 (5 Jan. 2007), pp. 2189–2195. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2006.08.052.

[52] Xiaoliang Wei et al. "Radical Compatibility with Nonaqueous Electrolytes and Its Impact on an All-Organic Redox Flow Battery." In: *Angewandte Chemie (International ed. in English)* 54 (30 July 2015), pp. 8684–8687. ISSN: 1521-3773. DOI: 10.1002/ANIE.201501443. URL: https://pubmed.ncbi.nlm.nih.gov/25891480/.

[53] Xiang Wang, Xueqi Xing, Yongjie Huo, Yicheng Zhao, Yongdan Li, and Hong Chen. "Study of tetraethylammonium bis(trifluoromethylsulfonyl)imide as a supporting electrolyte for an all-organic redox flow battery using Benzophenone and 1,4-di-tert-butyl-2,5-dimethoxybenzene as active species." In: *International Journal of Electrochemical Science* 13 (7 July 2018), pp. 6676–6683. ISSN: 14523981. DOI: 10.20964/2018.07.56.

[54] Ruiyong Chen and Rolf Hempelmann. "Ionic liquid-mediated aqueous redox flow batteries for high voltage applications." In: (2016). DOI: 10.1016/j.elecom.2016.07.003. URL: http://dx.doi.org/10.1016/j.elecom.2016.07.003.

[55] Andinet Ejigu, Peter A. Greatorex-Davies, and Darren A. Walsh. "Room temperature ionic liquid electrolytes for redox flow batteries." In: *Electrochemistry Communications* 54 (May 2015), pp. 55–59. ISSN: 1388-2481. DOI: 10.1016/J.ELECOM.2015.01.016.

[56] Qinghua Liu, Alice E.S. Sleightholme, Aaron A. Shinkle, Yongdan Li, and Levi T. Thompson. "Non-aqueous vanadium acetylacetonate electrolyte for redox flow batteries." In: *Electrochemistry Communications* 11 (12 Dec. 2009), pp. 2312–2315. ISSN: 1388-2481. DOI: 10.1016/J.ELECOM.2009.10.006.

[57] Kazunori Teramoto, Toshikazu Nishide, Shin Okumura, Koichiro Takao, and Yasuhisa Ikeda. "Studies on Metal Complexes as Active Materials in Redox-flow Battery Using Ionic Liquids as Electrolyte: Electrochemical Properties of [Fe(L)x][Tf2N]2 (L: Multidentate Ligands, x = 2 or 3) in 1-Butyl-3-methylimidazolium Bis(Trifluoromethylsulfonyl)Imide, [BMI][Tf2N]." In: *Electrochemistry* 82 (7 July 2014), pp. 566–572. ISSN: 1344-3542. DOI: 10.5796/ELECTROCHEMISTRY.82.566. URL: http://dx.doi.org/10.5796/electrochemistry.82.566.

[58] Stijn Schaltin, Yun Li, Neil R. Brooks, Jeroen Sniekers, Ivo F.J. Vankelecom, Koen Binnemans, and Jan Fransaer. "Towards an all-copper redox flow battery based on a copper-containing ionic liquid." In: *Chemical Communications* 52 (2 Dec. 2015), pp. 414–417. ISSN: 1364-548X. DOI: 10.1039/C5CC06774J. URL: https://pubs.rsc.org/en/content/articlehtml/2016/cc/c5cc06774jhttps:

//pubs.rsc.org/en/content/articlelanding/2016/cc/c5cc06774j.

[59] Yanxin Yao, Jiafeng Lei, Yang Shi, Fei Ai, and Yi Chun Lu. "Assessment methods and performance metrics for redox flow batteries." In: *Nature Energy 2021 6:6* 6 (6 Feb. 2021), pp. 582–588. ISSN: 2058-7546. DOI: 10.1038/s41560-020-00772-8. URL: https://www.nature.com/articles/s41560-020-00772-8.

[60] Musbaudeen O. Bamgbopa, Yang Shao-Horn, Raed Hashaikeh, and Saif Almheiri. "Cyclable membraneless redox flow batteries based on immiscible liquid electrolytes: Demonstration with all-iron redox chemistry." In: *Electrochimica Acta* 267 (Mar. 2018), pp. 41–50. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2018.02.063.

[61] Yuyue Zhao, Zhizhang Yuan, Wenjing Lu, Xianfeng Li, and Huamin Zhang. "The porous membrane with tunable performance for vanadium flow battery: The effect of charge." In: *Journal of Power Sources* 342 (Feb. 2017), pp. 327–334. ISSN: 0378-7753. DOI: 10.1016/J.JPOWSOUR.2016.12.058.

[62] Dongyang Chen, Michael A. Hickner, Ertan Agar, and E. Caglan Kumbur. "Optimizing membrane thickness for vanadium redox flow batteries." In: *Journal of Membrane Science* 437 (June 2013), pp. 108–113. ISSN: 0376-7388. DOI: 10.1016/J.MEMSCI.2013.02.007.

[63] K Gong. *Study of novel redox flow batteries based on double-membrane, single-membrane, and membrane-less cell configurations*. 2016. URL: https://search.proquest.com/openview/e2ae7acfa8fff80b332569ee3f38e405/1?pq-origsite=gscholar&cbl=18750.

[64] Andrés F. Molina-Osorio, Alonso Gamero-Quijano, Pekka Peljo, and Micheál D. Scanlon. "Membraneless energy conversion and storage using immiscible electrolyte solutions." In: *Current Opinion in Electrochemistry* 21 (June 2020), pp. 100–108. ISSN: 2451-9103. DOI: 10.1016/J.COELEC.2020.01.013.

[65] Paula Navalpotro, Jesus Palma, Marc Anderson, and Rebeca Marcilla. "A Membrane-Free Redox Flow Battery with Two Immiscible Redox Electrolytes." In: *Angewandte Chemie International Edition* 56 (41 Oct. 2017), pp. 12460–12465. ISSN: 1521-3773. DOI: 10.1002/ANIE.201704318.

URL: https://onlinelibrary.wiley.com/doi/full/10.1002/anie.201704318https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201704318https://onlinelibrary.wiley.com/doi/10.1002/anie.201704318.

[66] George M. Whitesides. "The origins and the future of microfluidics." In: *Nature 2006 442:7101* 442 (7101 July 2006), pp. 368–373. ISSN: 1476-4687. DOI: 10.1038/nature05058. URL: https://www.nature.com/articles/nature05058.

[67] P. Tabeling. *Introduction to microfluidics*. Oxford University Press, 2005, p. 301. ISBN: 0199588163. URL: https://books.google.com/books/about/Introduction_to_Microfluidics.html?hl=fr&id=wysKAwAAQBAJ.

[68] Younan Xia and George M. Whitesides. "Soft lithography." In: *Annual Review of Materials Science* 28 (1 1998), pp. 153–184. ISSN: 00846600. DOI: 10.1146/ANNUREV.MATSCI.28.1.153.

[69] Nam-Trung Nguyen, Steven T Wereley, and Seyed Ali Mousavi Shaegh. *Fundamentals and applications of microfluidics*. Artech house, 2019.

[70] Bastian E. Rapp. "Microfluidics: Modeling, mechanics and mathematics." In: *Microfluidics: Modeling, Mechanics and Mathematics* (Dec. 2016), pp. 1–766. DOI: 10.1016/C2012-0-02230-2. URL: http://www.sciencedirect.com:5070/book/9781455731411/microfluidics-modeling-mechanics-and-mathematics.

[71] Rosaria Ferrigno, Abraham D. Stroock, Thomas D. Clark, Michael Mayer, and George M. Whitesides. "Membraneless vanadium redox fuel cell using laminar flow." In: *Journal of the American Chemical Society* 124 (44 Nov. 2002), pp. 12930–12931. ISSN: 00027863. DOI: 10.1021/JA020812Q/SUPPL_FILE/JA020812Q_S.PDF. URL: https://pubs.acs.org/doi/pdf/10.1021/ja020812q.

[72] Jessamine Ng Lee, Cheolmin Park, and George M. Whitesides. "Solvent Compatibility of Poly(dimethylsiloxane)-Based Microfluidic Devices." In: *Analytical Chemistry* 75 (23 Dec. 2003). Compatibility between PDMS and most solvents, pp. 6544–6554. ISSN: 00032700. DOI: 10.1021/AC0346712. URL: https://pubs.acs.org/sharingguidelines.

[73] Eric R Choban, Larry J Markoski, Andrzej Wieckowski, and Paul J A Kenis. "Microfluidic fuel cell based on laminar flow." In: *Journal of Power Sources* 128 (2004), pp. 54–60. DOI: 10.1016/j.jpowsour.2003.11.052.

[74] E. R. Choban, J. S. Spendelow, L. Gancs, A. Wieckowski, and P. J.A. Kenis. "Membraneless laminar flow-based micro fuel cells operating in alkaline, acidic, and acidic/alkaline media." In: *Electrochimica Acta* 50 (27 Sept. 2005), pp. 5390–5398. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2005.03.019.

[75] Falin Chen, Min Hsing Chang, and Chia Wei Hsu. "Analysis of membraneless microfuel cell using decomposition of hydrogen peroxide in a Y-shaped microchannel." In: *Electrochimica Acta* 52 (25 Sept. 2007), pp. 7270–7277. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2007.05.072.

[76] William A. Braff, Martin Z. Bazant, and Cullen R. Buie. "Membrane-less hydrogen bromine flow battery." In: *Nature Communications 2013 4:1* 4 (1 Aug. 2013), pp. 1–6. ISSN: 2041-1723. DOI: 10.1038/ncomms3346. URL: https://www.nature.com/articles/ncomms3346.

[77] Erik Kjeang, Raphaelle Michel, David A. Harrington, Ned Djilali, and David Sinton. "A microfluidic fuel cell with flow-through porous electrodes." In: *Journal of the American Chemical Society* 130 (12 Mar. 2008), pp. 4000–4006. ISSN: 00027863. DOI: 10.1021/JA078248C. URL: https://pubs.acs.org/sharingguidelines.

[78] Sean Moore, David Sinton, and David Erickson. "A plate-frame flow-through microfluidic fuel cell stack." In: *Journal of Power Sources* 196 (2011), pp. 9481–9487. DOI: 10.1016/j.jpowsour.2011.07.024.

[79] Marc Antoni Goulet, Omar A. Ibrahim, Will H.J. Kim, and Erik Kjeang. "Maximizing the power density of aqueous electrochemical flow cells with in operando deposition." In: *Journal of Power Sources* 339 (Jan. 2017), pp. 80–85. ISSN: 0378-7753. DOI: 10.1016/J.JPOWSOUR.2016.11.053.

[80] Erik Kjeang. *Microfluidic Fuel Cells and Batteries*. Springer International Publishing, 2014. ISBN: 978-3-319-06345-4. DOI: 10.1007/978-3-319-06346-1. URL: https://link.springer.com/10.1007/978-3-319-06346-1.

[81] Li Li, Shaoyi Bei, Ranran Liu, Qiang Xu, Keqing Zheng, Yiyi She, and Yun He. "Design of a radial vanadium redox microfluidic fuel cell: A new way to break the size limitation." In: *International Journal of Energy Research* 43 (7 June 2019), pp. 3028–3037. ISSN: 1099-114X. DOI: 10.1002/ER.4473. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/er.4473https://onlinelibrary.wiley.com/doi/abs/10.1002/er.4473https://onlinelibrary.wiley.com/doi/10.1002/er.4473.

[82] Kamil S. Salloum and Jonathan D. Posner. "A membraneless microfluidic fuel cell stack." In: *Journal of Power Sources* 196 (3 Feb. 2011), pp. 1229–1234. ISSN: 03787753. DOI: 10.1016/j.jpowsour.2010.08.069.

[83] Yifei Wang and Dennis Y.C. Leung. "A high-performance aluminum-feed microfluidic fuel cell stack." In: *Journal of Power Sources* 336 (Dec. 2016), pp. 427–436. ISSN: 0378-7753. DOI: 10.1016/J.JPOWSOUR.2016.11.009.

[84] Seoung Hwan Lee and Yoomin Ahn. "A laminar flow-based single stack of flow-over planar microfluidic fuel cells." In: *Journal of Power Sources* 351 (May 2017), pp. 67–73. ISSN: 0378-7753. DOI: 10.1016/J.JPOWSOUR.2017.03.102.

[85] Ravishankar Sundararaman, Derek Vigil-Fowler, and Kathleen Schwarz. "Improving the Accuracy of Atomistic Simulations of the Electrochemical Interface." In: *Chemical Reviews* 122 (12 June 2022), pp. 10651–10674. ISSN: 15206890. DOI: 10.1021/ACS.CHEMREV.1C00800/ASSET/IMAGES/LARGE/CR1C00800_0016.JPEG. URL: https://pubs.acs.org/doi/full/10.1021/acs.chemrev.1c00800.

[86] Chengxiang Xiang et al. "Modeling, Simulation, and Implementation of Solar-Driven Water-Splitting Devices." In: *Angewandte Chemie International Edition* 55 (42 Oct. 2016), pp. 12974–12988. ISSN: 1521-3773. DOI: 10.1002/ANIE.201510463. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/anie.201510463https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201510463https://onlinelibrary.wiley.com/doi/10.1002/anie.201510463.

[87] Sehkyu Park, Jong Won Lee, and Branko N. Popov. "A review of gas diffusion layer in PEM fuel cells: Materials and designs." In: *International Journal of Hydrogen Energy* 37 (7 Apr. 2012), pp. 5850–5865. ISSN: 0360-3199. DOI: 10.1016/J.IJHYDENE.2011.12.148.

[88] Li Chen, Qinjun Kang, and Wenquan Tao. "Pore-scale study of reactive transport processes in catalyst layer agglomerates of proton exchange membrane fuel cells." In: *Electrochimica Acta* 306 (May 2019), pp. 454–465. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2019.03.158.

[89] Duo Zhang, Qiong Cai, Oluwadamilola O. Taiwo, Vladimir Yufit, Nigel P. Brandon, and Sai Gu. "The effect of wetting area in carbon paper electrode on the performance of vanadium redox flow batteries: A three-dimensional lattice Boltzmann study." In: *Electrochimica Acta* 283 (Sept. 2018), pp. 1806–1819. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2018.07.027. URL: https://openresearch.surrey.ac.uk/esploro/outputs/journalArticle/The-effect-of-wetting-area-in/99513697202346.

[90] Ya lu Fu, Biao Zhang, Xun Zhu, Ding ding Ye, Pang Chieh Sui, Ned Djilali, and Qiang Liao. "Pore-scale modeling of mass transport in the air-breathing cathode of membraneless microfluidic fuel cells." In: *International Journal of Heat and Mass Transfer* 188 (June 2022), p. 122590. ISSN: 0017-9310. DOI: 10.1016/J.IJHEATMASSTRANSFER.2022.122590.

[91] Alex P. Cocco, George J. Nelson, William M. Harris, Arata Nakajo, Timothy D. Myles, Andrew M. Kiss, Jeffrey J. Lombardo, and Wilson K.S. Chiu. "Three-dimensional microstructural imaging methods for energy materials." In: *Physical Chemistry Chemical Physics* 15 (39 Sept. 2013), pp. 16377–16407. ISSN: 1463-9084. DOI: 10.1039/C3CP52356J. URL: https://pubs.rsc.org/en/content/articlehtml/2013/cp/c3cp52356jhttps://pubs.rsc.org/en/content/articlelanding/2013/cp/c3cp52356j.

[92] Yu Chen Karen Chen-Wiegart, Paul Shearing, Qingxi Yuan, Andrei Tkachuk, and Jun Wang. "3D morphological evolution of Li-ion battery negative electrode LiVO 2 during oxidation using X-ray nano-tomography." In: *Electrochemistry Communications* 21 (1 July 2012), pp. 58–61. ISSN: 13882481. DOI: 10.1016/J.ELECOM.2012.04.033.

[93] Philip J. Withers. "X-ray nanotomography." In: *Materials Today* 10 (12 Dec. 2007), pp. 26–34. ISSN: 13697021. DOI: 10.1016/S1369-7021(07)70305-X. URL: http://dx.doi.org/10.1016/S1369-7021(07)70305-X.

[94] Severin Vierrath, Lukas Zielke, Riko Moroni, Andrew Mondon, Dean R Wheeler, Roland Zengerle, and Simon Thiele. "Morphology of nanoporous carbon binder domains in Li-ion batteries-A FIB-SEM study." In: (2015).

[95] Y. B. Yi, C. W. Wang, and A. M. Sastry. "Compression of Packed Particulate Systems: Simulations and Experiments in Graphitic Li-ion Anodes." In: *Journal of Engineering Materials and Technology* 128 (1 Jan. 2006), pp. 73–80. ISSN: 0094-4289. DOI: 10.1115/1.2130733. URL: https://dx.doi.org/10.1115/1.2130733.

[96] David E. Stephenson, Bryce C. Walker, Cole B. Skelton, Edward P. Gorzkowski, David J. Rowenhorst, and Dean R. Wheeler. "Modeling 3D Microstructure and Ion Transport in Porous Li-Ion Battery Electrodes." In: *Journal of The Electrochemical Society* 158 (7 2011), A781. ISSN: 00134651. DOI: 10.1149/1.3579996.

[97] S Kirkpatrick, ; C D Gelatt, and ; M P Vecchi. "Optimization by Simulated Annealing." In: *New Series* 220 (4598 1983), pp. 671–680.

[98] Wei Wu and Fangming Jiang. "Simulated annealing reconstruction and characterization of the three-dimensional microstructure of a LiCoO2 Lithium-ion battery cathode." In: *Materials Characterization* 80 (2013), pp. 62–68. ISSN: 10445803. DOI: 10.1016/J.MATCHAR.2013.03.011.

[99] Partha P. Mukherjee, Chao Yang Wang, and Qinjun Kang. "Mesoscopic modeling of two-phase behavior and flooding phenomena in polymer electrolyte fuel cells." In: *Electrochimica Acta* 54 (27 Nov. 2009), pp. 6861–6875. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2009.06.066.

[100] Aimy Bazylak, David Sinton, and Ned Djilali. "Improved fuel utilization in microfluidic fuel cells: A computational study." In: *Journal of Power Sources* 143 (1-2 Apr. 2005), pp. 57–66. ISSN: 03787753. DOI: 10.1016/J.JPOWSOUR.2004.11.029.

[101] Lin Wei, Xianxia Yuan, and Fangming Jiang. "A three-dimensional non-isothermal model for a membraneless direct methanol redox fuel cell." In: *Journal of Power Sources* 385 (May 2018), pp. 130–140. ISSN: 0378-7753. DOI: 10.1016/J.JPOWSOUR.2018.03.037.

[102] Julian Marschewski, Patrick Ruch, Neil Ebejer, Omar Huerta Kanan, Gaspard Lhermitte, Quentin Cabrol, Bruno Michel, and Dimos Poulikakos. "On the mass transfer performance enhancement of membraneless redox flow cells with mixing promoters." In: *International Journal of Heat and Mass Transfer* 106 (Mar. 2017), pp. 884–894. ISSN: 0017-9310. DOI: 10.1016/J.IJHEATMASSTRANSFER.2016.10.030.

[103] Min-Hsing Chang, Falin Chen, and Nai-Siang Fang. "Analysis of membraneless fuel cell using laminar flow in a Y-shaped microchannel." In: *Journal of Power Sources* 159 (2006), pp. 810–816. DOI: 10.1016/j.jpowsour.2005.11.066.

[104] Li Li, Georgios Nikiforidis, Michael K.H. Leung, and Walid A. Daoud. "Vanadium microfluidic fuel cell with novel multi-layer flow-through porous electrodes: Model, simulations and experiments." In: *Applied Energy* 177 (Sept. 2016), pp. 729–739. ISSN: 0306-2619. DOI: 10.1016/J.APENERGY.2016.05.072.

[105] Falin Chen, Min-Hsing Chang, and Mu-Kun Lin. "Analysis of membraneless formic acid microfuel cell using a planar microchannel." In: *Electrochimica Acta* 52 (2007), pp. 2506–2514. DOI: 10.1016/j.electacta.2006.09.011.

[106] Deepak Krishnamurthy, Erik O. Johansson, Jin Wook Lee, and Erik Kjeang. "Computational modeling of microfluidic fuel cells with flow-through porous electrodes." In: *Journal of Power Sources* 196 (23 Dec. 2011), pp. 10019–10031. ISSN: 0378-7753. DOI: 10.1016/J.JPOWSOUR.2011.08.024.

[107] Timm Krüger, Halim Kusumaatmaja, Alexandr Kuzmin, Orest Shardt, Goncalo Silva, and Erlend Magnus Viggen. "The Lattice Boltzmann Method." In: (2017). DOI: 10.1007/978-3-319-44649-3. URL: http://link.springer.com/10.1007/978-3-319-44649-3.

[108] A. A. Mohamad. "Lattice Boltzmann Method." In: *Lattice Boltzmann Method* (2019). DOI: 10.1007/978-1-4471-7423-3.

[109] Jay B Benziger, M Barclay Satterfield, Warren H J Hogarth, James P Nehlsen, and Ioannis G Kevrekidis. "The power performance curve for engineering analysis of fuel cells." In: *Journal of Power Sources* 155 (2006). polarization regimes (activation, ohmic, mass transfer limited), pp. 272–285. DOI: 10.1016/j.jpowsour.2005.05.049.

[110] Robert K. Emmett and Mark E. Roberts. "Recent developments in alternative aqueous redox flow batteries for grid-scale energy storage." In: *Journal of Power Sources* 506 (Sept. 2021), p. 230087. ISSN: 0378-7753. DOI: 10.1016/J.JPOWSOUR.2021.230087.

[111] Rita Khalil, Nesrine Chaabene, Mirella Azar, Ibrahim Bou Malham, and Mireille Turmine. "Effect of the chain lengthening on transport properties of imidazolium-based ionic liquids." In: *Fluid Phase Equilibria* 503 (Jan. 2020). longer alkyl chain means lower conductivity, p. 112316. ISSN: 0378-3812. DOI: 10.1016/J.FLUID.2019.112316.

[112] Spyridon Koutsoukos, Julian Becker, Ana Dobre, Zhijie Fan, Farhana Othman, Frederik Philippi, Gavin J. Smith, and Tom Welton. "Synthesis of aprotic ionic liquids." In: *Nature Reviews Methods Primers 2022 2:1* 2 (1 June 2022). Ionic liquids (ILs) are typically a salt in the liquid state. They have been intensively investigated for many applications, including as solvents, electrolytes, catalysts, energy storage materials and lubricants, owing to their synthetic flexibility; there is a large number of different possible combinations of anions and cations, which can lead to ILs with very distinct properties. However, there is no standard methodology for the synthesis of ILs, with the quality of these being based solely on the internal experience of each research group. This Primer provides a guide to IL synthetic practices, outlining considerations when choosing the synthetic route as well as evaluating the advantages and disadvantages of each route. The Primer serves as a reference for both new and experienced researchers, to help determine whether the IL from a specific route will be suitable for their intended application, be it ultra-pure, fast or cheap., pp. 1–18. ISSN: 2662-8449. DOI: 10.1038/s43586-022-00129-3.

[113] Viktoriya A. Nikitina, Renat R. Nazmutdinov, and Galina A. Tsirlina. "Quinones electrochemistry in room-temperature ionic liquids." In: *Journal of Physical Chemistry B* 115 (4 Feb. 2011), pp. 668–677. ISSN: 15205207. DOI: 10.1021/JP1095807/SUPPL_FILE/JP1095807_SI_001.PDF. URL: https://pubs.acs.org/doi/abs/10.1021/jp1095807.

[114] Hoang Tuan Nguyen, Ha Thach, Emmanuel Roy, Khon Huynh, and Cecile Mong Tu Perrault. "Low-Cost, Accessible Fabrication Methods for Microfluidics Research in Low-Resource Settings." In: *Micromachines 2018, Vol. 9, Page 461* 9 (9 Sept. 2018), p. 461. ISSN: 2072-666X. DOI: 10.3390/MI9090461. URL: https://www.mdpi.com/2072-666X/9/9/461/htmhttps://www.mdpi.com/2072-666X/9/9/461.

[115] Nirveek Bhattacharjee, Arturo Urrios, Shawn Kang, and Albert Folch. "The upcoming 3D-printing revolution in microfluidics." In: *Lab on a Chip* 16 (10 May 2016), pp. 1720–1742. ISSN: 1473-0189. DOI: 10.1039/C6LC00163G. URL: https://pubs.rsc.org/en/content/articlehtml/2016/lc/c6lc00163ghttps://pubs.rsc.org/en/content/articlelanding/2016/lc/c6lc00163g.

[116] Zachary Zguris. "How mechanical properties of stereolithography 3D prints are affected by UV curing." In: *Formlabs White Paper* (2016). how uv curing affects the mechanical properties of sla resin, pp. 1–11.

[117] Francesca Cosmi and Alberto Dal Maso. "A mechanical characterization of SLA 3D-printed specimens for low-budget applications." In: *Materials Today: Proceedings* 32 (2020). mechanical properties of SLA resin, pp. 194–201. DOI: 10.1016/j.matpr.2020.04.602. URL: https://doi.org/10.1016/j.matpr.2020.04.602.

[118] Lev Davidovich Landau and Evgenii Mikhailovich Lifshitz. *Fluid Mechanics: Landau and Lifshitz: Course of Theoretical Physics*. Vol. 6. Elsevier, 2013.

[119] Dieter Britz and Jörg Strutwolf. "Digital Simulation in Electrochemistry." In: (2016). DOI: 10.1007/978-3-319-30292-8. URL: http://link.springer.com/10.1007/978-3-319-30292-8.

[120] H Yamakawa, H ; R W Yamakawa, B H Kilb ; Zimm, G M Roe, L F Epstein ; Zimmerman, M C Williams ; Zwanzig, R ; J Kiefer, and G H Weiss. "Porous-electrode theory with battery applications." In: *AIChE Journal* 21 (1 Jan. 1975), pp. 25–41. ISSN: 1547-5905. DOI: 10.1002/AIC.690210103. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/aic.690210103https://onlinelibrary.wiley.com/doi/abs/10.1002/aic.690210103https://aiche.onlinelibrary.wiley.com/doi/10.1002/aic.690210103.

[121] TI Gombosi. *Gaskinetic theory*. 1994. URL: https://books.google.com/books?hl=en&lr=&id=fC52SR5yxRwC&oi=fnd&pg=PA1&dq=T.I.+Gombosi,+Gaskinetic+Theory+(Cambridge+University+Press,+Cambridge,+1994)&ots=hmLWbDrTCD&sig=Pqv3dO7jFws5dBfn-TuqUlcXUg0.

[122] Li-Shi Luo. "Lattice Boltzmann Model for the Incompressible Navier-Stokes Equation." In: *Article in Journal of Statistical Physics* 88 (1997), p. 1997. DOI: 10.1023/B%3AJOSS.0000015179.12689.e4. URL: https://www.researchgate.net/publication/304425201.

[123] Sydney Chapman and Thomas George Cowling. *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press, 1990.

[124] P. L. Bhatnagar, E. P. Gross, and M. Krook. "A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems." In: *Physical Review* 94 (3 May 1954), p. 511. ISSN: 0031899X. DOI: 10.1103/PhysRev.94.511. URL: https://journals.aps.org/pr/abstract/10.1103/PhysRev.94.511.

[125] Dieter A Wolf-Gladrow, Springer Berlin, Heidelberg Newyork, Hong Kong, London Milan, and Paris Tokyo. "Lattice-Gas Cellular Automata and Lattice Boltzmann Models-An Introduction." In: (2005).

[126] G Silva, V Semiao Journal of Computational Physics, and undefined 2014. "Truncation errors and the rotational invariance of three-dimensional lattice models in the lattice Boltzmann method." In: *Elsevier* 269 (2014), pp. 259–279. DOI: 10.1016/j.jcp.2014.03.027. URL: https://www.sciencedirect.com/science/article/pii/S0021999114002083.

[127] Xiaoyi He and Li-Shi Luo. "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation." In: (1997).

[128] Irina Ginzburg, Ginzburg, and Irina. "Equilibrium-type and link-type lattice Boltzmann models for generic advection and anisotropic-dispersion equation." In: *AdWR* 28 (11 Nov. 2005), pp. 1171–1195. ISSN: 0309-1708. DOI: 10.1016/J.ADVWATRES.2005.03.004. URL: https://ui.adsabs.harvard.edu/abs/2005AdWR...28.1171G/abstract.

[129] A Kuzmin, I Ginzburg, and A A Mohamad. "The role of the kinetic parameter in the stability of two-relaxation-time advection-diffusion lattice Boltzmann schemes." In: *Computers and Mathematics with Applications* 61 (2011), pp. 3417–3442. DOI: 10.1016/j.camwa.2010.07.036. URL: www.elsevier.com/locate/camwa.

[130] Wolfgang Schmickler. "Double layer theory." In: *Journal of Solid State Electrochemistry* 24 (9 Sept. 2020), pp. 2175–2176. ISSN: 14330768. DOI: 10.1007/S10008-020-04597-Z/METRICS. URL: https://link.springer.com/article/10.1007/s10008-020-04597-z.

[131] Xiaoyi He and Ning Li. "Lattice Boltzmann simulation of electrochemical systems." In: *Computer Physics Communications* 129 (1-3 July 2000), pp. 158–166. ISSN: 0010-4655. DOI: 10.1016/S0010-4655(00)00103-X.

[132] NVIDIA. *Compute Unified Device Architecture (CUDA)*. Nov. 2023. URL: https://developer.nvidia.com/cuda-zone.

[133] Inc. The Khronos Group. *Open Computing Language (OpenCL)*. Nov. 2023. URL: https://www.khronos.org/opencl/.

[134] Linux Foundation. *PyTorch*. 2023. URL: https://pytorch.org/https://github.com/pytorch/pytorch.

[135] Nikhil Ketkar and Jojo Moolayil. "Introduction to PyTorch." In: *Deep Learning with Python* (2021), pp. 27–91. DOI: 10.1007/978-1-4842-5364-9_2. URL: https://link.springer.com/chapter/10.1007/978-1-4842-5364-9_2.

[136] Takaji Inamuro, Masato Yoshino, and Fumimaru Ogino. "A non-slip boundary condition for lattice Boltzmann simulations." In: *Physics of Fluids* (1995).

[137] Qisu Zou and Xiaoyi He. "On pressure and velocity flow boundary conditions and bounceback for the lattice Boltzmann BGK model." In: (1996).

[138] AJ Bard, LR Faulkner, and HS White. *Electrochemical methods: fundamentals and applications*. 2022. URL: https://books.google.com/books?hl=en&lr=&id=Sct6EAAAQBAJ&oi=fnd&pg=PR21&dq=bard+and+faulkner&ots=QUboxgFXNR&sig=IUsV1zTXRqk6eofxsjU1dcBf-5w.

[139] Christian Amatore, Cécile Pebay, Laurent Thouin, and Aifang Wang. "Cyclic voltammetry at microelectrodes. Influence of natural convection on diffusion layers as characterized by in situ mapping of concentration profiles." In: *Electrochemistry Communications* 11 (6 June 2009), pp. 1269–1272. ISSN: 1388-2481. DOI: 10.1016/J.ELECOM.2009.04.018.

[140] Christian Amatore, Cécile Pebay, Laurent Thouin, Aifang Wang, and J. S. Warkocz. "Difference between ultramicroelectrodes and microelectrodes: Influence of natural convection." In: *Analytical Chemistry* 82 (16 Aug. 2010), pp. 6933–6939. ISSN: 00032700. DOI: 10.1021/AC101210R/ASSET/IMAGES/LARGE/AC-2010-01210R_0003.JPEG. URL: https://pubs.acs.org/doi/full/10.1021/ac101210r.

[141] André Lévêque. *Les Lois de la transmission de chaleur par convection*. Dunod. 1928.

[142] Tooru Tsuru, Toshiyasu Nishimura, and Shiro Haruyama. "Anodic Dissolution of Iron as Studied with Channel Flow Double Electrode." In: *Materials Science Forum* 8 (Jan. 1986), pp. 429–438. ISSN: 1662-9752. DOI: 10.4028/WWW.SCIENTIFIC.NET/MSF.8.429. URL: https://www.scientific.net/MSF.8.429.

[143] Christian Amatore, Nicolas Da Mota, Catherine Sella, and Laurent Thouin. "Theory and Experiments of Transport at Channel Microband Electrodes under Laminar Flows. 1. Steady-State Regimes at a Single Electrode." In: *Analytical Chemistry* 79 (22 Nov. 2007), pp. 8502–8510. ISSN: 00032700. DOI: 10.1021/AC070971Y. URL: https://pubs.acs.org/doi/full/10.1021/ac070971y.

[144] J Newman and NP Balsara. *Electrochemical systems*. 2021. URL: https://books.google.com/books?hl=en&lr=&id=JlYMEAAAQBAJ&oi=fnd&pg=PP15&dq=newman+electrochemical+systems&ots=el7MysyY_6&sig=YhmJaC0Ekvk38AI7E2um3tWMCfw.

[145] SN Lvov. *Introduction to Electrochemical Science and Engineering*. 2021. URL: https://books.google.com/books?hl=en&lr=&id=S71IEAAAQBAJ&oi=fnd&pg=PP1&dq=lvov+electrochemical&ots=hlfoFOcDeN&sig=tat0NjkS6UYlrf91EyB7zdiLa2c.

[146] Graham M. Goldin, Andrew M. Colclasure, Andreas H. Wiedemann, and Robert J. Kee. "Three-dimensional particle-resolved models of Li-ion batteries to assist the evaluation of empirical parameters in one-dimensional models." In: *Electrochimica Acta* 64 (Mar. 2012), pp. 118–129. ISSN: 0013-4686. DOI: 10.1016/J.ELECTACTA.2011.12.119.

[147] Norman Epstein. "On tortuosity and the tortuosity factor in flow and diffusion through porous media." In: *Chemical Engineering Science* 44 (3 Jan. 1989), pp. 777–779. ISSN: 0009-2509. DOI: 10.1016/0009-2509(89)85053-5.

[148] MacIej Matyka, Arzhang Khalili, and Zbigniew Koza. "Tortuosity-porosity relation in porous media flow." In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 78 (2 Aug. 2008). ISSN: 15393755. DOI: 10.1103/PHYSREVE.78.026306.

[149] Artur Duda, Zbigniew Koza, and Maciej Matyka. "Hydraulic tortuosity in arbitrary porous media flow." In: *PHYSICAL REVIEW E* 84 (2011), p. 36319. DOI: 10.1103/PhysRevE.84.036319.

DECLARATION

Put your declaration here.

Paris - Tanger, 16 December 2023

Youssef Kharchouf

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić.

Final Version as of March 15, 2024 (Pre-submission draft).