

Model of chromium poisoning in the cathode of a solid oxide fuel cell using the lattice Boltzmann method

Gayle Marie Kestell

Thesis Submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Mechanical Engineering

Committee Members:
Michael R. von Spakovsky, Chair
Michael W. Ellis
P. Kathy Lu
William T. Reynolds

Friday, February 12, 2010
Blacksburg, Virginia

Keywords: Fuel Cell, SOFC, Kinetic Theory, Lattice Boltzmann Method, chromium poisoning

Model of chromium poisoning in the cathode of a solid oxide fuel cell using the lattice Boltzmann method

Gayle Marie Kestell

Abstract

The metallic interconnect of a solid oxide fuel cell (SOFC) contains chromium in order to protect the metal from the corrosive environment in the fuel cell. Unfortunately, the chromium introduces chemical instability in the cathode as it migrates from the interconnect to the pores in the cathode. A model was developed previously in Asinari et al. [1] and Kasula et al [2] to model the flow of particles in a fuel cell electrode. To learn more about the migration of the chromium, the previous code is modified in this thesis work to include the effects of the chromium. The model uses Kinetic Theory to simulate the fuel cell at a mesoscopic scale. The discretized form of the Lattice Boltzmann equation is modified for enhanced performance and for use on a parallel processing system.

With the new model, the migration of the chromium in the cathode and the performance degradation of the fuel cell are predicted.

Acknowledgements:

I would like to thank Dr. von Spakovsky for choosing me work on this project and guiding me through its entirety. I would like to thank Dr. Pietro Asinari for answering my many questions. I would like to thank Stephen Edwards for teaching me Linux and helping me debug code. I would also like to thank Lucas Tucker for helping me better understand the chemistry in my simulation. Finally, I would like to thank my thesis committee, Dr. Michael Ellis, Dr. Kathy Lu and Dr. William Reynolds, for reviewing my thesis.

Dedication

This work is dedicated to my family who encouraged me to pursue my education, especially my husband who supported me every step of the way.

Table of Contents

List of Figures	vii
List of Tables	xi
Chapter 1: Introduction	1
1.1 Fuel Cell Overview	3
1.2 Solid Oxide Fuel Cells	5
1.3 Chromium Poisoning	6
1.4 Fuel Cell Modeling	8
1.5 Research Objective	9
Chapter 2: Literature Review	11
2.1 Fuel Cell Modeling	11
2.1.1 Macroscopic Continuum Models	12
2.1.2 Mesoscopic Non-continuum Models	17
2.1.3 3D Microstructure Reconstruction	20
2.2 Chromium Poisoning	21
2.3 Conclusion	23
Chapter 3: Mathematical Model	25
3.1 Kinetic Theory	25
3.2 The BGK Master Equation	31
3.2.1 The BGK Model	32
3.2.2 Discrete and Discretized forms of the Boltzmann and BGK Equations	34
3.2.3 The Lattice Boltzmann Method [18]	36
3.2.4 3D Cathode Reacting Mixture Model	43
3.3 Porous Medium 3D Reconstruction	46
3.3.1 One-Point Statistics (Granulometry Law) [2]	46
3.3.2 Two-point Statistics	49
3.3.3 Connectivity Calculations	55
Chapter 4: Physical Properties and Geometric Considerations	56
4.1 Physical Properties	56
4.1.1 System Properties and Initial Conditions	56
4.1.2 Diffusivity	58
4.1.3 Viscosity	61
4.2 Geometry	63
4.2.1 Porosity	63
4.2.2 TPBs	63
4.3 Boundary Conditions	64
4.3.1 Inlet and Outlet B.C.s	64
4.3.2 Electrochemical B.C.s	65
4.4 Relaxation Frequencies	69
4.4.1 Relaxation Frequency Related to Diffusivity	70
4.4.2 Relaxation Frequencies Related to Viscosity	71
Chapter 5: Results and Discussion	73
5.1 Validation, H ₂ O and H ₂ mixture	75
5.1.1 2 Species Mixture for Geometries A and B	76

5.1.2 2 Species Mixture for Geometries C, D and E.....	81
5.1.3 2 Species Mixture for Geometries F and G	84
5.2 Without Chromium Poisoning.....	86
5.2.1 3 Species Mixture for Geometries A and B.....	86
5.2.2 3 Species Mixture for Geometries C, D and E.....	90
5.2.3 3 Species Mixture for Geometries F and G	93
5.3 With Chromium Poisoning.....	97
5.3.1 Mixture without Blocking of the TPBs for Geometries A and B	97
5.3.2 Mixture with Blocking of the TPBs for Geometry B	100
5.3.3 Mixture without Blocking of the TPBs for Geometry C	104
5.3.4 Mixture with Blocking of the TPBs for Geometry C	105
5.3.5 Mixture without Blocking of the TPBs for Geometries D and E	107
5.3.6 Mixture without Blocking of the TPBs for Geometry F.....	108
5.3.7 Mixture with Blocking of the TPBs for Geometry F.....	111
5.3.8 Mixture without Blocking of the TPBs for Geometry G	111
Chapter 6: Conclusions and Recommendations	113
6.1 Conclusions – Problem, Models, and Methods	113
6.2 Conclusions - Code.....	114
6.3 Conclusions and Recommendations - Simulation Results.....	115
References.....	117
Appendix A: Porous Geometry Reconstruction Code	124
Appendix B: ASC1 SOFC Data Sheets	162
Appendix C: LBM Input Parameters	165
Appendix D: Post Processing Code	166
Appendix E: Results Graphs and Contour Plots.....	178
Appendix F: LBM Code Function Tree.....	267
Appendix G Refinement Factor 8 Preliminary Results	275
Appendix H Copyright Permission Letters.....	286

List of Figures

Figure 1.1. SOFC fuel cell schematic [4].	2
Figure 1.2. Fuel cells in series [5].	3
Figure 1.3. SOFC with interconnects.	5
Figure 1.4. Fuel cell degradation due to chromium poisoning [7].	7
Figure 3.1. Lattice structure diagram.	34
Figure 3.2. Two dimensional, five node (D2Q5) lattice.	37
Figure 3.3. Two dimensional, nine node (D2Q9) lattice.	37
Figure 3.4. Two dimensional, seven node (D2Q7) lattice.	37
Figure 3.5. Three dimensional, nineteen node (D3Q19) lattice.	38
Figure 3.6. Pre-collision lattice.	42
Figure 3.7. Post-collision lattice.	43
Figure 3.8. Post-streaming lattice.	43
Figure 3.9. SOFC cathode active layer: original image (SEM-BSE).	46
Figure 3.10. SOFC cathode active layer: threshold image, for one-point statistics; black indicates pores, dark grey LSM, light grey YSZ.	47
Figure 3.11. One-point statistics function containing pore information, $f(d_i)$.	47
Figure 3.12. SOFC cathode active layer: 3D reconstructed image using one-point statistics.	49
Figure 3.13. SOFC cathode active layer: threshold image (YSZ only), two-point statistics; white indicates YSZ and black pore and LSM.	49
Figure 3.14. SOFC cathode active layer: threshold image (LSM only), two-point statistics; white indicates LSM and black pore and YSZ.	49
Figure 3.15. Two point statistics algorithm, black: solid, white: pores; (a) zoomed-in portion of the digital image at an individual pixel level, (b) step 1, (c) step 3, (d) step 4.	50
Figure 3.16. Sampled two-point correlation functions, $S_2(r)$, and target two-point correlation function, $\hat{S}_2(r)$.	52
Figure 3.17. SOFC cathode active layer: 3D reconstructed image, two-point statistics.	53
Figure 3.18. Reconstructed geometry with a porosity of 50%; (a) inlet plane; (b) intermediate plane.	54
Figure 3.19. Mass flux contours for previously published geometry with a porosity of 50%; (a) inlet plane; (b) intermediate plane [2].	54
Figure 4.1. SEM SOFC cathode image with a TPB site and its electron and ion path marked.	64
Figure 5.1. Comparison of H_2 and H_2O species τ variations as a function of depth using Geometry B for (a) the results from this thesis work; (b) the results from Kasula et al. [2] ⁴ .	77
Figure 5.2. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O and H_2 mixture for (a) H_2O and (b) H_2 .	78
Figure 5.3. Normalized surface-averaged density variations from the results from Asinari et al. [1] as a function of depth using Geometry B for the H_2O and H_2 mixture.	78
Figure 5.4. Comparison of H_2 and H_2O species normalized surface-averaged mass flux variations as a function of depth using Geometry B for the H_2O and H_2 mixture for the results from (a) this thesis work and (b) Asinari et al. [1] ⁶ .	79
Figure 5.5. Comparison of H_2O mass flux contours at the inlet/outlet plane using Geometry B for the H_2O and H_2 mixture for the results from (a) this thesis work and (b) Kasula et al. [2] ⁴ .	80

Figure 5.6. Comparison of H ₂ O mass flux contours at an intermediate plane using Geometry B for the H ₂ O and H ₂ mixture for the results from (a) this thesis work and (b) Kasula et al. [2] ⁴ ...	80
Figure 5.7. Comparison of H ₂ mass flux contours at the inlet/outlet plane using Geometry B for the H ₂ O and H ₂ mixture for the results from (a) this thesis work and (b) Kasula et al. [2] ⁴	81
Figure 5.8. Comparison of H ₂ mass flux contours at an intermediate plane using Geometry B for the H ₂ O and H ₂ mixture for the results from (a) this thesis work and (b) Kasula et al. [2] ⁴	81
Figure 5.9. Results of this thesis work for variations as a function of depth using Geometry C for the H ₂ O and H ₂ mixture for (a) τ and (b) normalized surface-averaged mass flux.	82
Figure 5.10. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry C for the H ₂ O and H ₂ mixture for (a) H ₂ O and (b) H ₂	83
Figure 5.11. Results of this thesis work for variations as a function of depth using Geometry F for the H ₂ O and H ₂ mixture for (a) τ and (b) normalized surface-averaged mass flux.	84
Figure 5.12. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H ₂ O and H ₂ mixture for (a) H ₂ O and (b) H ₂	85
Figure 5.13. Results of this thesis work for variations as a function of depth using Geometry B for the H ₂ O, O ₂ and N ₂ mixture for (a) τ and (b) normalized surface-averaged mass flux.	87
Figure 5.14. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H ₂ O, O ₂ and N ₂ mixture for (a) H ₂ O and (b) O ₂	88
Figure 5.15. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H ₂ O, O ₂ and N ₂ mixture for N ₂	88
Figure 5.16. Normalized mass flux contours for H ₂ O using Geometry B for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	89
Figure 5.17. Normalized mass flux contours for O ₂ using Geometry B for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	89
Figure 5.18. Normalized mass flux contours for N ₂ using Geometry B for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	89
Figure 5.19. Results of this thesis work for variations as a function of depth using Geometry C for the H ₂ O, O ₂ and N ₂ mixture for (a) τ for H ₂ O and O ₂ , (b) τ for N ₂ and (c) normalized surface-averaged mass flux.	90
Figure 5.20. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H ₂ O, O ₂ and N ₂ mixture for (a) H ₂ O and (b) O ₂	91
Figure 5.21. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H ₂ O, O ₂ and N ₂ mixture for N ₂	91
Figure 5.22. Normalized mass flux contours for H ₂ O using Geometry C for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	92
Figure 5.23. Normalized mass flux contours for O ₂ using Geometry C for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	92
Figure 5.24. Normalized mass flux contours for N ₂ using Geometry C for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	92
Figure 5.25. Results of this thesis work for variations as a function of depth using Geometry F for the H ₂ O, O ₂ and N ₂ mixture for (a) τ for H ₂ O and O ₂ , (b) τ for N ₂ and (c) normalized surface-averaged mass flux.	94

Figure 5.26. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H ₂ O, O ₂ and N ₂ mixture for (a) H ₂ O and (b) O ₂ .	95
Figure 5.27. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H ₂ O, O ₂ and N ₂ mixture for N ₂ .	95
Figure 5.28. Normalized mass flux contours for H ₂ O using Geometry F for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	95
Figure 5.29. Normalized mass flux contours for O ₂ using Geometry F for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	96
Figure 5.30. Normalized mass flux contours for N ₂ using Geometry F for the H ₂ O, O ₂ and N ₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.	96
Figure 5.31. Results of this thesis work for variations as a function of depth using Geometry B for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for (a) τ and (b) normalized surface-averaged mass flux.	98
Figure 5.32. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for (a) H ₂ O and (b) O ₂ .	98
Figure 5.33. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for (a) N ₂ and (b) CrO ₃ .	99
Figure 5.34. Comparison of the normalized mass flux contours for O ₂ using Geometry F for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for (a) plane 50 and (b) plane 75.	99
Figure 5.35. Comparison of the normalized mass flux contours for N ₂ using Geometry F for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for (a) plane 50 and (b) plane 75.	100
Figure 5.36. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for O ₂ after (a) 10 collisions, (b) 50 collisions, (c) 500 collisions, (d) 1000 collisions, (e) 3000 collisions.	101
Figure 5.37. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for CrO ₃ after (a) 10 collisions, (b) 50 collisions, (c) 500 collisions, (d) 1000 collisions, (e) 3000 collisions.	102
Figure 5.38. Results of this thesis work for variations as a function of depth using Geometry C for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for (a) τ and (b) normalized surface-averaged mass flux.	104
Figure 5.39. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for (a) H ₂ O and (b) O ₂ .	104
Figure 5.40. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for (a) N ₂ and (b) CrO ₃ .	105
Figure 5.41. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for O ₂ after (a) 10 collisions, (b) 50 collisions, (c) 500 collisions, (d) 1000 collisions.	106
Figure 5.42. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture for CrO ₃ after (a) 10 collisions, (b) 50 collisions, (c) 500 collisions, (d) 1000 collisions.	107

Figure 5.43. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) τ and (b) normalized surface-averaged mass flux. 108

Figure 5.44. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) H₂O and (b) O₂. 109

Figure 5.45. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) N₂ and (b) CrO₃. 109

Figure 5.46. Normalized mass flux contours for H₂O using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) the inlet/outlet plane, (b) an intermediate plane. 110

Figure 5.47. Normalized mass flux contours for O₂ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) the inlet/outlet plane, (b) an intermediate plane. 110

Figure 5.48. Normalized mass flux contours for N₂ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) the inlet/outlet plane, (b) an intermediate plane. 110

Figure 5.49. Normalized mass flux contours for CrO₃ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) the inlet/outlet plane, (b) an intermediate plane. 111

List of Tables

Table 1.1. Fuel cell overview [3].	4
Table 2.1. Recknagle, Williford, Chick, Rector and Khaleel [38].	12
Table 2.2. Ferguson, Fiard and Herbin [38].	13
Table 2.3. Ho, Kosinski, Hoffmann and Vik [39].	14
Table 2.4. Suwanwarangkul, Croiset, Entchev, Charojrochkul, Pritzker, Fowler, Douglas, Chewathanakup and Mahaudom [40].	14
Table 2.5. Hall and Colclaser [41].	15
Table 2.6. Kenny and Karan [44].	15
Table 2.7. Chan, Chen and Khor [45].	16
Table 2.8. Costamagna, Costa and Antonucci [46].	17
Table 2.9. Asinari, Quaglia, von Spakovsky and Kasula [1].	18
Table 2.10. Chen, Doolen and Eggert [48].	18
Table 2.11. Asinari [50].	19
Table 2.12. Békri, Vizika, Thovert and Adler [53].	19
Table 2.13. Kasula, Mercado, Asinari and von Spakovsky [2].	21
Table 2.14. Jiao, Stillinger and Torquato [19].	21
Table 2.15. Badwal, Deller, Foger, Ramprakash and Zhang [6].	22
Table 2.16. Stanislawski, Wessel, Hilpert, Markus and Singheiser [8].	22
Table 2.17. Taniguchi, Kadowaki, Kawamura, Yasuo, Akiyama, Miyake and Saitoh [7].	23
Table 2.18. Paulson and Birss [57].	23
Table 2.19. Stanislawski, Froitzheim, Niewolak, Quadackers, Hilpert, Markus and Singheiser [9].	24
Table 3.1. D2Q9 lattice speeds and global equilibrium distributions.	39
Table 4.1. Summary of the φ_σ for the water, oxygen and nitrogen mixture [70].	57
Table 4.2. Summary of the φ_σ for the water, oxygen, nitrogen and chromium oxide vapor mixture [70].	57
Table 4.3. Initial normalized densities or mass fractions for the water, oxygen and nitrogen mixture [8].	58
Table 4.4. Initial normalized densities or mass fractions for the water, oxygen, nitrogen and chromium oxide vapor mixture [8].	58
Table 4.5. Relevant properties for calculating mixture diffusivity [74].	59
Table 4.6. Relevant properties for calculating the chromium oxide vapor characteristic properties [75].	59
Table 4.7. Diffusivities for the water, oxygen and nitrogen mixture.	61
Table 4.8. Diffusivities for the water, oxygen, nitrogen and chromium oxide vapor mixture.	61
Table 4.9. Properties for calculating the mixture viscosity.	61
Table 4.10. Kinematic viscosities for the water, oxygen and nitrogen mixture [1, 74].	62
Table 4.11. Kinematic viscosities for the water, oxygen, nitrogen and chromium oxide vapor mixture [1, 74].	63
Table 4.12. Cathode active layer properties.	63
Table 4.13. IOBCs for the oxygen, nitrogen and water mixture.	64
Table 4.14. IOBCs for the oxygen, nitrogen, water and chromium oxide vapor mixture.	64
Table 4.15. Electrochemical reaction parameters for the water, oxygen and nitrogen mixture. ..	69

Table 4.16. Electrochemical reaction parameters for the water, oxygen, nitrogen and chromium oxide vapor mixture when the oxygen reacts always.	69
Table 4.17. Electrochemical reaction parameters for the water, oxygen, nitrogen and chromium oxide vapor mixture with the chromium oxide vapor reacting.	69
Table 4.18. Dimensionless diffusivity and $\hat{\lambda}_m^I$ for the water, oxygen and nitrogen mixture.	71
Table 4.19. Dimensionless diffusivity and $\hat{\lambda}_m^I$ for the water, oxygen, nitrogen and chromium oxide vapor mixture.	71
Table 4.20. Dimensionless effective viscosity and $\hat{\lambda}_{m1}^{II}$ for the water, oxygen and nitrogen mixture.	72
Table 4.21. Dimensionless effective viscosity and $\hat{\lambda}_{m1}^{II}$ for the water, oxygen, nitrogen and chromium oxide vapor mixture.	72
Table 5.1. Descriptions of the geometries used in this thesis work.	75
Table 5.2. Volume-averaged property summary for the results from this thesis work using Geometry A and the H ₂ O and H ₂ mixture.	76
Table 5.3. Volume-averaged property summary for the results from Kasula et al. [2] using Geometry A and the H ₂ O and H ₂ mixture.	76
Table 5.4. Volume-averaged property summary for the H ₂ O and H ₂ mixture using Geometry D.	83
Table 5.5. Volume-averaged property summary for the H ₂ O and H ₂ mixture using Geometry E.	83
Table 5.6. Volume-averaged property summary for the H ₂ O and H ₂ mixture using Geometry G.	86
Table 5.7. Volume-averaged property summary for the results from this thesis work using Geometry A and the H ₂ O, O ₂ and N ₂ mixture.	87
Table 5.8. Volume-averaged property summary for the results from this thesis work using Geometry D and the H ₂ O, O ₂ and N ₂ mixture.	93
Table 5.9. Volume-averaged property summary for the results from this thesis work using Geometry E and the H ₂ O, O ₂ and N ₂ mixture.	93
Table 5.10. Volume-averaged property summary for the results from this thesis work using Geometry G and the H ₂ O, O ₂ and N ₂ mixture.	96
Table 5.11. Volume-averaged property summary for the results from this thesis work using Geometry A and the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture.	97
Table 5.12. Volume-averaged property summary for the results from this thesis work using Geometry D and the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture.	108
Table 5.13. Volume-averaged property summary for the results from this thesis work using Geometry E and the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture.	108
Table 5.14. Volume-averaged property summary for the results from this thesis work using Geometry G and the H ₂ O, O ₂ , N ₂ and CrO ₃ mixture.	111

Chapter 1: Introduction

As interest in renewable energy and emissions reduction grows, interest in fuel cells grows as well. Fuel cells are currently under development for use in vehicles, power centers for buildings, cell phones and portable computers. Some fuel cell models (alkaline fuel cells or AFCs for example) were used by NASA in manned spacecraft. Fuel cells are particularly interesting in the areas of combined heat and power systems and mobile power systems. They have features that make them or have the potential for making them desirable. They combine hydrogen (with the exception of the direct methanol fuel cell (DMFC) which uses methanol) and oxygen in an electrochemical reaction to produce power with water as a byproduct which is a clean emission. Obtaining hydrogen, however, can be inefficient (as in the case of electrolysis) or release pollutants (as in the case of reforming natural gases), however, at levels typically below those of conventional combustion processes. Another attribute that makes fuel cells attractive is that they are generally more efficient than combustion engines [3]. Excluding the balance of plant (BOP), fuel cells are also very simple, requiring no moving parts, which translates to easier maintenance than in other types of systems. This is a somewhat moot point in that the BOP may consist of fuel reactors, turbo machinery, etc. for fuel refining and thermal management. Nonetheless, fuel cell operation is comparatively quiet.

One important disadvantage preventing the wide-spread use of fuel is the cost [3]. Expensive materials are required for high temperature fuel cells such as in the solid oxide fuel cell (SOFC) and the molten carbonate fuel cell (MCFC) because of the high thermal stresses and reactive environment. Other fuel cells, such as AFCs and proton exchange membrane fuel cells (PEMFCs) operate at lower temperatures but as a result require expensive catalysts to aid the electrochemical reaction.

Fuel cells create power by combining oxygen and hydrogen (pure or from a natural gas) via an electrochemical reaction. Using the reaction mechanism of a solid oxide fuel cell as an example (different types of fuel cells have different reaction equations), oxygen undergoes the following half reaction in the cathode:



The oxygen ions are conducted to the anode through the electrolyte where the hydrogen is converted to water via:



Power is created by the movement of electrons due to the electrochemical reactions releasing electrons on one side of the fuel cell and consuming them on the other side. See Figure 1.1.

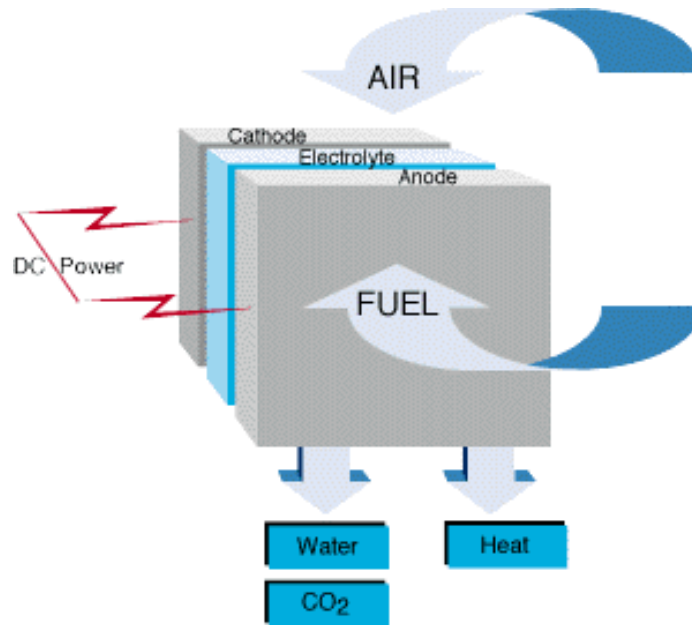


Figure 1.1. SOFC fuel cell schematic [4]¹.

A single fuel cell produces little voltage, on the order of 0.7 V [3], which is insufficient for most applications. In order to increase the voltage, fuel cells are connected in series electrically (Figure 1.2) using interconnects. An interconnect is made of a highly electrically conductive material that has separate channels for the oxygen and hydrogen containing fluids for the adjoining cells.

In a solid oxide fuel cell, the interconnect experiences high temperatures and a chemically aggressive environment. The cathode side is particularly corrosive as it contains oxygen and often water. Because of the oxygen inlet flow and the high temperature SOFC interconnects containing chromium release a chromium oxide vapor(s) (CrO_3 and/or $\text{Cr}(\text{OH})_2\text{O}_2$) into the cathode which severely degrades the performance of the fuel cell. The behavior of the chromium oxide vapor within the cathode is not completely understood. A microscopic continuum mathematical model that is capable of modeling the fluid flow within the cathode

¹ Reprinted from Fuel Cell Systems Explained, Second Edition, Fuel cell information guide, with permission from USACE.

would help researchers better understand the changes in the cathode that lead to its degradation. Such a model is developed in the thesis work.

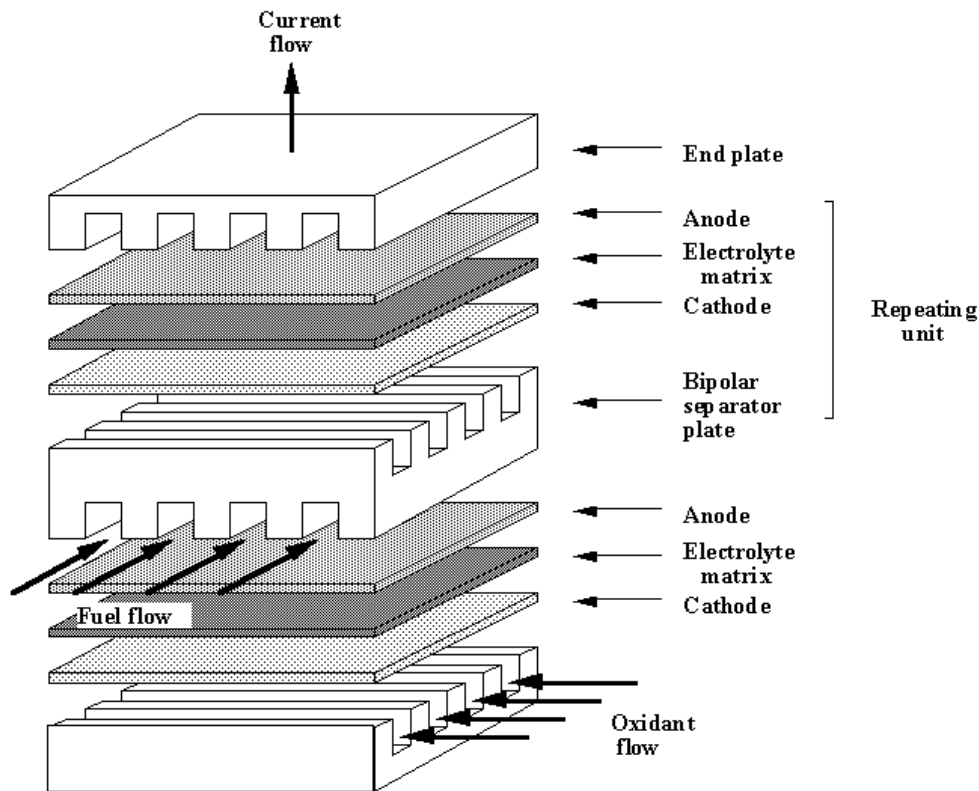


Figure 1.2. Fuel cells in series [5]².

1.1 Fuel Cell Overview

There are six common types of fuel cells each with different strengths and weaknesses. Table 1.1 shows a summary of some of the properties of the fuel cells.

AFCs use an alkaline solution (e.g. potassium hydroxide) as an electrolyte, which is a very low cost material. The electrodes also do not need expensive precious metals or exotic materials. One significant disadvantage of AFCs is that they are very susceptible to carbon dioxide poisoning and so the inlet flows must be carefully cleaned.

PEMFCs use a polymer that contains mobile H^+ ions called a proton exchange membrane (PEM) as the electrolyte. The PEM can be very thin, thereby allowing PEMFCs to be compact. PEMFCs operate at low temperatures that make the start-up and shut-down processes faster than

² Copied by permission from the "Electrochemical Science and Technology Information Resource" (ESTIR) (<http://electrochem.cwru.edu/estir/>) on "02/17/2010." The original material is subject to periodical changes and updates.

those of a higher temperature fuel cell. PEMFCs can work in any orientation and require no hazardous fluids. The electrodes of PEMFCs can be very expensive because the low temperatures require them to contain sophisticated catalysts to speed the electrochemical reaction rate. PEMFCs also require complicated water management and cooling.

Table 1.1. Fuel cell overview [3].

Fuel cell type	Mobile ion	Operating temperature range	Power output (W)	Applications
Alkaline fuel cell (AFC)	OH ⁻	50-200 °C	500-10k	Spacecraft, vehicles,
Proton exchange membrane fuel cell (PEMFC)	H ⁺	30-100 °C	1-100k	Mobil applications, spacecraft, vehicle, combined heat and power systems
Direct methanol fuel cell (DMFC)	H ⁺	20-90 °C	1-100	Portable, low power electronics, applications requiring long running times
Phosphoric acid fuel cell (PAFC)	H ⁺	~ 220 °C	10k-1M	combined heat and power systems
Molten carbonate fuel cell (MCFC)	CO ₃ ⁻²	~ 650 °C	100k-10M	Distributed power generation, combined heat and power systems, medium to large scale
Solid oxide fuel cell (SOFC)	O ⁻²	500-1000 °C	1k-10M	combined heat and power systems

DMFCs are typically a type of PEMFC but use methanol instead of pure hydrogen as a fuel. This eliminates the need for reforming the fuel before it can be used in the cell. The methanol, however, can pass through the PEM and react at the cathode which reduces the power output of the cell. DMFCs are very low power as seen in Table 1.1 and so are suitable only for smaller applications, but they may become an alternative to the lithium ion battery [3].

PAFCs use phosphoric acid as an electrolyte. Phosphoric acid is relatively stable and is tolerant to carbon dioxide. As in PEMFCs and DMFCs, PAFCs require expensive catalysts in the electrodes to increase electrochemical reaction rates. Field trials have proven PAFCs to produce reliable, high quality power [3].

The electrolyte of an MCFC is mixture of molten alkali metal carbonates which adds to the complexity of the system. The molten carbonates conduct CO₃⁻² ions from the cathode to the anode. MCFCs are the only fuel cell that requires CO₂ in the electrochemical reactions. Overall

the amount of CO₂ in the system remains constant as it is consumed in the cathode but the same amount is formed again in the anode. The catalysts in the electrodes can be made of cheaper materials than those in the PEMFC, DMFC, and PAFC due to the high operation temperature.

Finally, SOFCs use solid ion conducting materials as electrolytes. As with the MCFC the high operating temperature of the SOFC eliminates the need for expensive catalyst materials in the electrodes. Unfortunately, the high temperatures require all of the materials in the cell to be specialized materials that can withstand the thermal stresses. An attractive feature of SOFCs is that they are capable of internal reforming, eliminating the need for external processes to reform the fuel. As SOFCs are the fuel cell studied in the research, they will be discussed in more detail in the next section.

1.2 Solid Oxide Fuel Cells

SOFCs are among the simplest of fuel cells as they are completely solid-state. They are made of layers of solid materials (as opposed to more delicate membranes as in the PEMFC) and do not require delicate electrolyte balancing as in PAFCs and MCFCs. SOFCs operate at high temperatures which eliminates the need for expensive catalysts required in low temperature fuel cells (see Table 1.1). The high temperatures in SOFCs makes selecting suitable materials difficult and expensive [3].

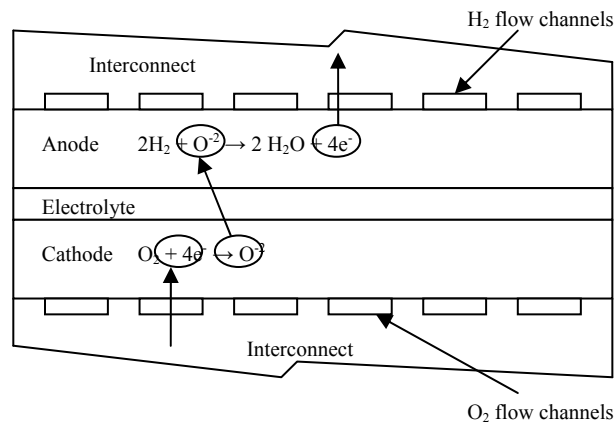


Figure 1.3. SOFC with interconnects.

The cathode of an SOFC (see Figure 1.3) is comprised of a porous material or compound of materials that conducts electrons and ions. Oxygen in the cathode undergoes an electrochemical reaction at so-called three-phase boundaries (TPB), which are sites where, in addition to the reaction, there is connectivity to the interconnect (i.e., an electron path) and the

electrolyte (i.e., an ion path). The electrochemical reactions in the cathode absorb electrons and release oxygen ions. The oxygen ions are conducted through the solid electrolyte from the cathode to the anode. The electrolyte is a barrier to fluid flow from one side of the fuel cell to the other. The anode (see Figure 1.3) is an ion conducting porous ceramic and electron conducting nickel composite. When the hydrogen in the pores interacts with a location that contains both the ceramic connected to the electrolyte and the nickel connected to the interconnect, i.e., this location is a TPB on the anode side, water and electrons are released and ions are absorbed. The electrons released are conducted to the load or through the interconnect to the next cell.

The interconnect must be able to resist corrosion at high temperatures while maintaining its ability to conduct electrons and it must have a thermal expansion coefficient similar to those for the anode, electrolyte and cathode materials. In order to increase the corrosion resistance of the materials used for the interconnect, additives are introduced. One of the common additives is chromium. Chromium is used in many manufacturing processes in addition to those for high temperature fuel cell materials. For example, chromium is used in making stainless steel to protect the iron from rusting by forming a hard protective oxide layer. Chromium oxide is, however, susceptible to degradation in a high temperature environment such as exists in an SOFC [6]. The degradation of chromium oxide in an SOFC will be discussed in the following section.

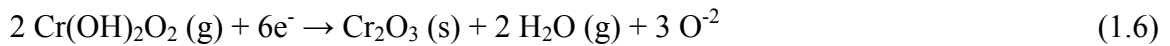
1.3 Chromium Poisoning

As mentioned above, the interconnect of an SOFC is subjected to harsh environments: high temperatures, oxygen, hydrogen and water. High temperatures combined with fluids that contain oxygen cause the interconnect to corrode and become electrically resistant, which severely degrades the performance of the fuel cell. Any decrease in the conductivity of the interconnect directly impacts fuel cell performance. Chromium is added to the interconnect because its properties help prevent the corrosion of the interconnect while maintaining good electrical conductivity. Chromium prevents metals from oxidation by oxidizing faster than the metal and forming a protective chromium oxide layer. The oxide layer is tenacious enough to resist further oxidation while preventing the metal from oxidizing as well. Research shows that at high temperatures, however, the chromium oxide, Cr_2O_3 , is reactive with oxygen or with oxygen and water (both of which are typically in the inlet flow on the cathode

side of an SOFC) and forms chromium oxide vapors. Two of which are CrO_3 or $\text{Cr}(\text{OH})_2\text{O}_2$ gases which are formed via the following reactions:



The CrO_3 and $\text{Cr}(\text{OH})_2\text{O}_2$ vapors enter the cathode's pores and react at the electrochemically reactive sites (similar to oxygen under ideal operation). As a result, the oxygen in CrO_3 is converted to ions at the active site and Cr_2O_3 solid is deposited on the electrochemically reactive site, according to the reactions:



The solid chromium oxide stays on the active site and blocks it from further use as a TPB. Blocking the reaction sites degrades the performance of the fuel cell since this reduces the surface area on which the electrochemical reactions can occur. Eventually the reactions completely stop because there are no more sites at which the electrochemical reactions occur and the fuel cell shuts down [6].

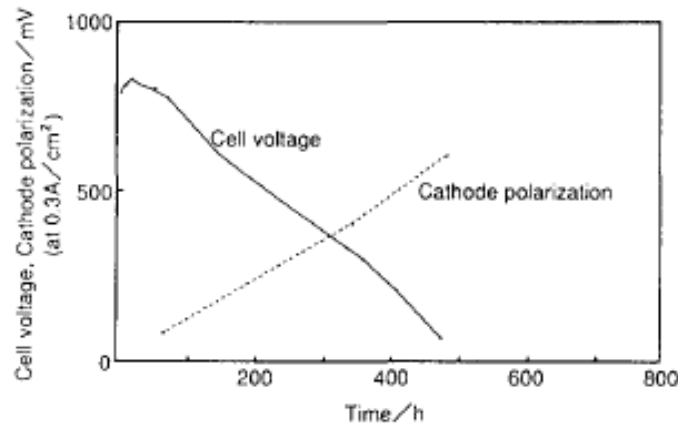


Figure 1.4. Fuel cell degradation due to chromium poisoning [7]³.

Figure 1.4 shows the degradation of a fuel cell operating with a constant current density of 0.3 A/cm^2 at a temperature of $1000 \text{ }^\circ\text{C}$ using air in the SOFC on the cathode side. Taniguchi et al. [7] present further results to verify that chromium oxide vapor is causing significant degradation in the voltage output of the cell. After only about 500 hr the cell they tested

³ Reprinted from Journal of Power Sources, 55, S. Taniguchi, M. Kadowaki, H. Kawamura, T. Yasuo, Y. Akiyama, Y. Miyake and T. Saitoh, Degradation phenomena in the cathode of a solid oxide fuel cell with an alloy separator, 74, Copyright (1995), with permission from Elsevier.

produced only a tenth of the original voltage. Further hours of operation quickly render the fuel cell useless. Clearly this is unacceptable.

In an attempt to lessen the amount of chromium released into the cathode, new interconnect materials and coatings are being developed and tested [8, 9]. In [8], Stanislawski et al. tested different interconnect materials and coatings in order to find which releases the lowest amount of chromium oxide vapor, i.e. the material and/or coating that will have the slowest TPB blocking rate. They found that sputtered ceramic chromium coatings did not reduce the amount of chromium oxide vapor released by the interconnect by a significant amount. The interconnects with metallic coatings, however, released only 1% of the chromium oxide vapor of that of uncoated interconnects. The coatings tested need to be examined further in order to determine long term performance and durability. The interconnect materials (without coatings) tested in [8] and [9] were all found to release too much chromium oxide vapor, thereby causing an unacceptable cell degradation rate. Further research is needed to develop new materials and/or coatings that do not release chromium and are sufficiently durable.

SOFCS are composed of very different components and, thus, place extreme demands on materials to perform in severe environments involving high temperatures and chemically aggressive conditions [10-13]. The most challenging problems are those associated with interfacial reactions and diffusion, which lead to performance degradation between the metal interconnect and the oxygen electrode in SOFCs [14-16]. In particular, migration of Cr-containing species from the metallic interconnect to the oxygen electrode causes serious chemical and microstructural stability problems. These issues are often exacerbated by long term operation (>40,000 hrs), high temperatures (750-900°C), and corrosive atmospheres (wet oxidizing), which frequently drive materials into regimes where conventional understanding of material behaviors and transport processes is not sufficient.

1.4 Fuel Cell Modeling

In the chemically reactive environments of SOFCs, high temperatures can have devastating effects on materials by accelerating degradation kinetics and causing significant interdiffusion, reactions, and/or microstructural changes. Proper design of materials to withstand these extremes requires a clear understanding of how ions, electrons, and defects move and how much nano-/meso scale interactions lead to composition and microstructure changes. Unfortunately, from a theoretical standpoint, fundamental understanding of these phenomena is

very limited. Thus, there is a critical need to establish a sound framework and knowledge base for material predictions at interfaces in SOFCs. This requires understanding the non-equilibrium thermodynamics and chemical kinetics at the nano-/meso-scales that describe the complex set of interrelated chemical and physical phenomena taking place. Such an understanding is crucial for further development of SOFCs, particularly if the goal of prolonged life times and reduced costs is to be achieved without significant reductions in energy conversion efficiencies. This requires revolutionary advances in electrode and interconnect materials and how they interface.

To model the flow of reactants from the flow channels and the interconnect into the porous fuel cell electrode in order to capture the effects of chromium poisoning on fuel cell performance, microscopic as opposed to macroscopic continuum models are needed. The latter, unlike the former, are unable to capture the details of microstructural effects or the placement of the individual TPBs which are being blocked. However, microscopic models have the drawback of being much more computationally demanding per unit volume and, thus, require significantly more computational resources. Nonetheless, to gain an insight in the performance degradation due to chromium poisoning, this level of modeling is required.

The type of microscopic (or mesoscopic) modeling approach which will be used in this thesis work is a non-continuum approach which comes from Kinetic Theory [17, 18]. This statistical mechanically based theory looks at the effects of the collisions of particles to predict the flow of a gas or a mixture (non-reacting or reacting) through a flow structure. Implicitly satisfied in solving this theory's statistically based master equations of motion (e.g. the BGK, the Fokker-Planck etc. [17, 18]) are conservation of mass, momentum and energy. In fact, taking moments of the set of microscopic master equations utilized results in the macroscopic continuum conservation equations of mass, momentum and energy; and, thus, there is a direct link between the time constant at the microscopic level and the experimentally based transport properties at the continuum level (e.g. viscosity, diffusivity, thermal and electrical conductivities, etc.). It is this microscopic approach which will be used here to meet the object of this research outlined in the following section.

1.5 Research Objective

As previously stated, chromium poisoning from the interconnect to the cathode electrode is a significant problem facing the development of SOFC's. Thus, the objective of this thesis work is to gain a better understanding via a parametric simulation of the movement of chromium

oxide vapor particles in the cathode. To that end a microscopic geometric model of the cathode porous microstructure as well as a microscopic mathematical model of reactant mixture flow will be developed based on previous work done at the Center for Energy Systems Research [1, 2]. The previous work which was based on two-species reactant mixture flows in an SOFC anode will be expanded upon to include more species and flow in the cathode where the reactant mixture consists of oxygen, nitrogen, water and other species (e.g. chromium oxides). The microscopic mathematical model of reactant mixture flows is based on the BGK [17, 18] equation of Kinetic Theory and is solved numerically using the lattice Boltzmann method [17, 18]. To accomplish this objective, the following set of tasks is envisioned:

- Examine the literature to ascertain the cause of chromium poisoning.
- Understand Kinetic Theory and derive the set of governing equations which determine reactant mixture flows in porous media.
- Understand and learn how to apply the LB Method.
- Examine and understand in detail the LBM code partially developed and used previously at the Center for Energy Systems Research in collaboration with Dr. Pietro Asinari of the Politecnico di Torino.
- Modify the code to incorporate the new flow mode equations and appropriate boundary and initial conditions.
- Obtain scanning electron microscope (SEM) images of an SOFC cathode.
- Write code based on two-point statistics [2, 19, 20] to recreate 3D geometries of the cathode based on the SEM images.
- Validate both geometric and LBM codes with the previous anode work [1, 2] as well as with the literature.
- Do a parametric study of the effects of chromium poisoning on fuel cell performance.

To accomplish this last task, the results from a cathode subjected to chromium poisoning will be compared to the ideal case in which there is no chromium poisoning. Parameters which will be examined include porosity, the size of region, the number and placement of TPBs, the threshold level for chromium poisoning, the time to fuel cell shutdown and the number of species in the simulation.

Chapter 2: Literature Review

This literature review gives an overview of some of the techniques available for modeling fuel cells. They range from the macroscopic to mesoscopic level of descriptions and from continuum to non-continuum models. The mesoscopic non-continuum models (Asinari et al. [1] for example) require a 3D reconstruction of the fuel cell electrodes and some methods for 3D microstructure reconstruction are discussed.

The focus of this thesis work is the effects of chromium poisoning on the cathode of an SOFC. This literature review discusses the causes of chromium poisoning and research being done to reduce and understand chromium poisoning.

2.1 Fuel Cell Modeling

Mathematical models of the phenomena, which occur in the fuel cells, are useful for understanding how new materials or operating conditions affect the performance of a fuel cell or one of its components without having to build an expensive prototype. These models range from so-called lump-parameter models typically used for system-level modeling to one-, two- and three-dimensional models which predict the behavior of a given component or subcomponent using a macroscopic (> 1 mm), mesoscopic ($1 \mu\text{m}$ to 1 mm), or atomistic ($< 1 \mu\text{m}$) level description. The macroscopic description involves a continuum approach solved analytically or numerically using, for example, a finite element [21, 22, 23, 24], finite difference [23, 24, 25] or finite volume [26, 27, 28] method. The mesoscopic description typically involves a non-continuum statistical mechanical approach such as that provided by Kinetic Theory and the Boltzmann equation and its derivative master equations (the BGK and Fokker-Plank equations [17, 18]). These are always solved numerically using methods such as the Lattice Boltzmann Method (LBM). Of course, sometimes a continuum approach can be used even at the mesoscopic level provided the Knudsen number is in a favorable range (e.g., > 0.1 but < 3 or 4). For the atomistic description, a variety of approaches from the classical (e.g., molecular dynamics (MD) [29]) to the statistical (e.g., Kinetic Monte Carlo [30-32]) the quantum mechanical (e.g., quantum chemistry, density functional theory, dissipative quantum mechanics [33-37]) are used. A more recent approach at this level of description in quantum

thermodynamics [37] which provides both an equilibrium as well as a non-equilibrium description if the phenomena present,

The focus in this literature survey is only on the macroscopic description as it relates to SOFC operation and performance and on the mesoscopic description as is relates to SOFC operation and performance and to flow in porous media. Thus, a sampling of lumped-parameter and 1D to 3D continuum models is examined along with 1D to 3D non-continuum models.

2.1.1 Macroscopic Continuum Models

In Recknagle et al. [38], the behavior of a planar SOFC stack is modeled with the purpose of finding the flow configuration with the least amount of thermal stress. The authors use a finite volume computational fluid dynamics (CFD) code coupled with electrochemical equations. Recknagle et al. find that the thermal stresses are lowest in the co-flow case because the temperature gradient across the cell is lower than that of the other flow cases. This study is summarized briefly in Table 2.1.

Table 2.1. Recknagle, Williford, Chick, Rector and Khaleel [38].

Paper	Three-dimensional thermo-fluid electrochemical modeling of planar SOFC stacks (2003).
Authors	K. P. Recknagle, R. E. Williford, L. A. Chick, D. R. Rector and M. A. Khaleel
Model Summary	Commercial computational fluid dynamics simulation code (finite volume) combined with electrochemistry calculations
Region	A single fuel cell
Dimensionality	3D macroscopic
Assumptions	Cyclic boundary conditions; adiabatic lateral walls; current density calculation based on temperature; hydrogen reaction calculation based on current; flow channels are one large flow channel
Phenomena Investigated	Predict flow and concentration of cathode and anode gases; predict temperature fuel use and current distribution; lower temperature gradients to lower stress; compare cross-, co-, and counter flows
Main Conclusions	Co-flow yields the smallest temperature gradient and, thus, has thermo-structural advantages over the other cases
Limits	Must have a pre-calibrated electrochemistry model, macroscopic

In the next paper, Ferguson, Fiard and Herbin [38] use the conservation equations for thermal energy, mass and electric current in a finite volumes CFD code to predict the performance of a number of SOFC geometries. Both temperature and electric potential are

predicted. Both planar and tubular geometries for various and configurations (co-flow, counter-flow, etc.) are investigated. The authors find that the counter flow configuration is the most efficient. The results correspond well with International Energy Agency (IEA) benchmark models with the exception of the electric potential which is somewhat lower which is probably due to the fact that the model takes diffusion effects into account more accurately than the IEA models. This study is summarized briefly in Table 2.2.

Table 2.2. Ferguson, Fiard and Herbin [38].

Paper	Three-dimensional numerical simulation for various geometries of solid oxide fuel cells (1996).
Authors	J. R. Ferguson, J. M. Fiard and R. Herbin
Model Summary	Conservation of thermal energy, electric current, and mass; finite volume discretization
Region	A single fuel cell
Dimensionality	3D macroscopic
Assumptions	Electrochemical reactions occur only at the anode/electrolyte and cathode/electrolyte boundaries; no chemical reactions in the air channels; shift reactions occur in the fuel channel and anode domain; no radiation effects; continuous electric current
Phenomena Investigated	Local electrical potential; temperature; local species concentrations; most efficient flow configuration; most efficient anode thickness
Main Conclusions	Model confirmed by IEA benchmark models and numerical expressions; counter flow is most efficient
Limits	No radiation effects; porosity not taken into account

Ho et al. [39] use a commercial finite volume CFD package to solve equations for the chemical and electrochemical reactions present in an SOFC as well as those for the conservation of energy, momentum and mass. This study is summarized in Table 2.3. The fluid concentrations, temperatures and current densities are obtained for an anode supported SOFC unit. The model produces reasonable results, but more experiments must be performed to fully validate the model.

Next, Suwanwarangkul et al. [40] implement a mechanistic transport model along with appropriate electrochemical and reaction kinetics in a finite elements CFD code and apply it to an SOFC button cell geometry. The mechanistic transport model includes momentum, mass and charge transport equations. Commercial software is used to solve the system equations. Suwanwarangkul et al. are interested in carbon deposit formation and the influence of the deposits on operation. The model agrees well with experimental results for 900 °C but not for

800 °C. Further investigation is required to understand why the model does not produce accurate predictions at 800 °C. The study is summarized in Table 2.4.

Table 2.3. Ho, Kosinski, Hoffmann and Vik [39].

Paper	Numerical modeling of solid oxide fuel cells (2008).
Authors	T. X. Ho, P. Kosinski, A. C. Hoffmann and A. Vik
Model Summary	Commercial CFD package augmented with subroutines
Region	A single fuel cell
Dimensionality	3D macroscopic
Assumptions	Electric potential is uniformly distributed; external forces are neglected; gases obey Dalton's laws; homogeneous medium and viscosity; ideal, incompressible, laminar flow
Phenomena Investigated	Analysis of the phenomena in the electrodes: concentration, temperature, current densities
Main Conclusions	Reasonable results for concentrations, temperature and current densities for an anode-supported planar SOFC were obtained; derived activation overpotential found to be low
Limits	Thermal effects on mass diffusion not included

Table 2.4. Suwanwarangkul, Croiset, Entchev, Charojrochkul, Pritzker, Fowler, Douglas, Chewathanakup and Mahaudom [40].

Paper	Experimental and modeling study of solid oxide fuel cell operating with syngas fuel (2006).
Authors	R. Suwanwarangkul, E. Croiset, E. Entchev, S. Charojrochkul, M. D. Pritzker, M. W. Fowler, P. L. Douglas, S. Chewathanakup and H. Mahaudom
Model Summary	Mechanistic transport model coupled with electrochemical and chemical reaction kinetics; equations solved by a commercial finite element CFD software
Region	A single cell
Dimensionality	2D macroscopic isothermal button cell
Assumptions	Mechanistic based; electrode resistance approximated with experimental results; isothermal; constant gas density; mass transfer and ohmic resistances neglected; reactions occur only at the electrode/electrolyte interface
Phenomena Investigated	Carbon formation; influence of carbon deposits on operation
Main Conclusions	Model agrees well with the experimental results, especially at 900 °C
Limits	Model not accurate at 800 °C

The study of Hall and Colclaser [41] to model the transient behavior of a tubular SOFC is summarized in Table 2.5. The model includes the effects of independently verified thermal and electrochemical finite volume models. The authors investigate changes in system properties due

to changes in electrical load. The results for the transient model follow the same trends as those for a previously published model [43].

Table 2.5. Hall and Colclaser [41].

Paper	Transient modeling and simulation of a tubular SOFC (1999).
Authors	D. L. Hall and R. G. Colclaser
Model Summary	Electrochemical, thermal and mass flow equations solved using a finite volume CFD approach
Region	A single fuel cell
Dimensionality	3D macroscopic
Assumptions	Axial conduction neglected, constant fuel utilization percentage
Phenomena Investigated	Transient fuel cell performance
Main Conclusions	Shows similar results to previously published SOFC models
Limits	Model shows transient behavior only between known steady state points

Table 2.6. Kenny and Karan [44].

Paper	Mathematical micro-model of a solid oxide fuel cell composite cathode (2004).
Authors	B. Kenny and K Karan
Model Summary	Ohm's law, Butler-Volmer equation, Stefan-Maxwell equation applied to a charge balance
Region	Cathode
Dimensionality	1D mesoscopic
Assumptions	Cathode reactions govern cell output
Phenomena Investigated	Performance of a fuel cell under different porosities
Main Conclusions	Model followed the same trend as the experimental data but had higher performance
Limits	Not accurate for higher overpotentials

Kenny and Karan [44] model the electrochemical losses in the cathode of a SOFC since the cathode dominates the electrochemical polarization in the fuel cell. They use a charge balance with Ohm's law, the Butler-Volmer equation and the Stefan-Maxwell equation to model the cathode in one dimension. The characteristics of the cathode are included in parameters such as the effective resistivity and the TPB line length, which is related to the probability of percolation for each phase. Results show good agreement to experimental results for low overpotentials. The results include a parametric study for various compositions of the cathode species, including the porosity for which the ideal composition is found. This study is summarized briefly in Table 2.6.

Chan, Chen and Khor [45] perform a parametric study on the thickness, porosity and other parameters of the cathode of an SOFC. The model is a lumped-parameter model that uses

reaction mechanism equations (oxygen reduction and reactions at the cathode/electrolyte interface) and an equivalent electrical circuit to predict the behavior of the cathode. The optimum configuration for the cathode is found and the results agree with those experimental data.

Table 2.7. Chan, Chen and Khor [45].

Paper	Cathode micromodel of solid oxide fuel cell (2004).
Authors	Chan, S. H., Chen, X. J., Khor, K. A
Model Summary	Reaction mechanism equations, effective circuit,
Region	Cathode
Dimensionality	Lumped-parameter equivalent circuit macroscopic model
Assumptions	Assumed rate determining step of reaction mechanism (all other steps are in equilibrium); uniform specific active area, catalyst activity, temperature and constant resistivity throughout; cathode parameters from the literature
Phenomena Investigated	Effect of microstructure, transport phenomena, and electrochemical processes on cathode over potential in different operating conditions
Main Conclusions	Found the optimum particle size and electrode thickness for maximum electronic and ionic resistivity and minimum overpotential; results in agreement with previous papers; larger particles require a thicker electrode to lower overpotential; operating conditions did not effect the ideal electrode thickness but did effect the optimum particle size
Limits	Does not allow for variations along the geometry perpendicular to the fluid flow

Costamagna, Costa and Antonucci [46] use a one-dimensional model to predict the performance of the electrodes of a fuel cell in order to find the optimum electrode thickness. The model uses electronic and ionic transport equations along with the electrochemical reaction equations in order to predict the behavior of the electrodes. The equations are solved analytically along with charge balance and mass transport equations. The model uses information about the porosity of a fuel cell in the form of the percolation threshold which is a parameter based on the volume concentration of a species in the electrode material and the size of clusters created by those particles. The model results are in agreement with experimental data in literature. The study is briefly summarized in Table 2.8.

The problem with the macroscopic models presented above is that they contain no detailed information of the electrode microstructures and TPBs to be able to use them to model

the effects of chromium poisoning in an SOFC cathode. For that, a 3D mesoscopic approach is required. Applications of this type of approach are discussed next.

Table 2.8. Costamagna, Costa and Antonucci [46].

Paper	Micro-modelling of solid oxide fuel cell electrodes (1997).
Authors	P. Costamagna, P. Costa and V. Antonucci
Model Summary	Electronic and ionic transport equations; electrochemical reaction equations
Region	Electrodes
Dimensionality	1D mesoscopic
Assumptions	Steady state; uniform temperature and pressure throughout; hydrogen and oxygen are reacting electrochemically; conducting phases are continuous and homogeneous
Phenomena Investigated	Optimal electrode thickness
Main Conclusions	Found approximate maximum performance for electrode thicknesses
Limits	Maximum performance is difficult to predict due to errors in percolation theory

2.1.2 Mesoscopic Non-continuum Models

The lattice Boltzmann method described in Asinari et al. [1] is a mesoscopic approach which utilizes the microstructure itself and the locations of the TPBs to describe reacting mixture flow through the porous structure of an SOFC anode. This approach develops a system of governing equations based on the BGK master equation of Kinetic Theory to describe the time evolution of a set of species particle probability density distributions from which the species densities, mass fluxes and other properties can be obtained. The equations implemented in a research code using the LB method are applied to a 3D reconstruction of the geometry of the porous anode from a 2D SEM image of the electrode; measured values of porosity and the percentages of electron and ion conducting materials present; and one- and two-point statistics reconstruction techniques [2, 19, 20]. The model is used to predict the performance of the SOFC anode structure and TPB locations. A brief outline of this study is provided in Table 2.9.

Chen, Doolen and Eggert [48] use governing equations derived from the Boltzmann equation theory to predict the results of the phase separation of two immiscible fluids. In a related work, the authors work with an oil company and in a companion article [49] to predict the behavior of immiscible fluids (water and oil) flowing through a porous sandstone geometry. The surface tensions of the fluids are included in the collision operator. These mesoscopic models

are solved using the LB Method. They also discuss other general applications of their governing equations and the LB method. Their results compare well with those of experiments for flow patterns and permeabilities. The study is briefly summarized in Table 2.10.

Table 2.9. Asinari, Quaglia, von Spakovsky and Kasula [1].

Paper	Direct numerical calculation of the kinematic tortuosity of reactive mixture flow in the anode layer of solid oxide fuel cells by the lattice Boltzmann method (2007).
Authors	P. Asinari, M. C. Quaglia, M. R. von Spakovsky and B. V. Kasula
Model Summary	LB Method
Region	Anode
Dimensionality	3D mesoscopic
Assumptions	Ideal gas; two species; Butler-Volmer equations for boundary conditions; reconstructed medium uses particle shape and placement simplifications; low Reynolds number flow
Phenomena Investigated	Electronic output, fluid particle flow and density
Main Conclusions	Two-point statistics creates a more accurate 3D geometry; tortuosity values correspond well to those in literature [47]
Limits	Model too detailed to model entire fuel cell at the same time, ion and electron dynamics only implicitly included

Table 2.10. Chen, Doolen and Eggert [48].

Paper	Lattice Boltzmann fluid dynamics a versatile tool for multiphase and other complicated flows (1994).
Authors	S. Chen, G. D. Doolen and K. G. Eggert
Model Summary	LB Method
Region	Porous medium
Dimensionality	2D mesoscopic
Assumptions	Incompressible fluids; momenta of interacting particles is transferred at a constant rate towards equilibrium
Phenomena Investigated	Fluid dynamics of two immiscible fluids; geometry permeability; flow patterns
Main Conclusions	Results compare well with experimental results
Limits	Must have low Reynolds number; lack of Galilean invariance; unphysical equations of state

Asinari [50] looks at the flow of two species through a randomly generated porous medium. He shows the asymptotic analysis of the Hamel model [51, 52] and the LB method and shows that the lattice Boltzmann equations for a mixture reduce to known flow equations. The simulations produce results that are in good agreement with the theoretical predictions. The

results yield fluid flow profiles and diffusivity and viscosity data. This study is a precursor to the work done in Asinari et al. [1] and is briefly summarized in Table 2.11.

Table 2.11. Asinari [50].

Paper	Viscous coupling based lattice Boltzmann model for binary mixtures (2005).
Authors	P. Asinari
Model Summary	LB method
Region	Randomly generated porous medium
Dimensionality	1D, 2D mesoscopic
Assumptions	Ideal gas; isothermal
Phenomena Investigated	Two arbitrary fluid velocities in a randomly generated porous medium
Main Conclusions	Good agreement between theoretical predictions and simulation results for diffusivity and viscosity
Limits	Very computationally demanding

Békri, et al. [53] investigate the fluid vaporization and pressure evolution in a porous medium. They use a series of structures and conditions focusing on the phase change in order to plot the phase distribution and macroscopic properties (e.g. phase distribution). The research is limited by the time of computation due to the significant numerical effort. Further work can be done in improving the computational time and other areas to extend the model's value for predicting fluid flows. The model is briefly summarized in Table 2.12.

Table 2.12. Békri, Vizika, Thovert and Adler [53].

Paper	Binary two-phase flow with phase change in porous media (2001).
Authors	S. Békri, O. Vizika, J.-F. Thovert and P. M. Adler
Model Summary	Immiscible LB Method
Region	Porous medium
Dimensionality	2D, 3D mesoscopic
Assumptions	Equations of state; domain dimensions much larger than pore scale; phases well mixed; uniform temperature; local equilibrium
Phenomena Investigated	Fluid vaporization; pressure evolution; phase distribution
Main Conclusions	Reasonable results; results compare well with those for theoretical Poiseuille flow
Limits	Long computational time

Of the mesoscopic approaches just presented, the one most applicable for modeling the effects of chromium migration in the cathode of an SOFC is that used in Ref. [1], i.e., Kinetic Theory and the LB Method. This approach will be used to simulate the TPBs being blocked due to chromium deposits. The research code used in Ref. [1] is available and must be modified

from the two species (H_2 and H_2O) that it currently simulates to the four species (H_2O , O_2 , N_2 and CrO_3 or $Cr(OH)_2O_2$) seen in the cathode. It must also be changed to simulate the blocking of the TPB sites. However, in order to apply this approach, a technique for digitally reconstructing the 3D microstructure is needed. The two approaches used in this thesis work are briefly discussed below. A detailed discussion is found in Chapter 3.

2.1.3 3D Microstructure Reconstruction

There are several techniques available for digitally reconstructing 3D porous microstructures. One is done using SEM/TEM images of a series of slices of the electrode structure. However, this is an expensive, delicate and time consuming process beyond the current resources of this thesis work. Until recently models were also limited by the fact that it is impossible to obtain a resolution of more than 10 μm between slices [2, 54] but new technology allows a resolution closer to 50 nm [55]. Another method is 3D computer tomography which uses non-destructive x-ray technology to create a 3D pore space image. Again, however, the resolution (about one micron) is not sufficient for modeling the electrodes of an SOFC [2, 56]. An alternative is to construct the geometry statistically based on a single SEM/TEM image. Kasula et al [2] discuss two such methods: one-point statistics based on the granulometry law and two-point statistics.

Two-point statistics provides a more reliable reconstruction since it takes into account not only porosity, but also uses an auto-correlation function which provides information about the pore connectivity in the porous medium. Kasula et al. [2] and Asinari et al. [1] use commercial software from Engineering Simulation and Scientific Software Ltda. called IMAGOTM in conjunction with an in house code to apply the two point statistics to create the digital reconstruction of the 3D geometry. The one-point statistics is applied using an in-house code. The techniques are summarized in Table 2.13.

Jiao, Stillinger and Torquato [19] create an algorithm for two-point statistics reconstruction. Their goal is to be able to model and characterize a porous medium efficiently and accurately. The paper below and its companion paper [20] discuss techniques and algorithms used as well as real-world examples. Since the IMAGOTM software is no longer available to us, an in-house code based on Ref. [19, 20] was written in this thesis work to digitally recreate the cathode geometries needed in this thesis work (see Chapter 3). A summary of the authors' two-point statistics work is given in Table 2.14.

Table 2.13. Kasula, Mercado, Asinari and von Spakovsky [2].

Paper	3D microstructure reconstructions of solid oxide fuel cell electrodes with applications to numerical simulations of reacting mixture flows using LBM (2007).
Authors	B. V. Kasula, L. Mercado, P. Asinari and M. R. von Spakovsky
Model Summary	One- and two-point statistics using the IMAGO™ commercial software and some in-house codes
Region	Anode
Assumptions	One-point statistics: all particles are uniformly shaped; isotropic material
Main Conclusions	Two 3D reconstructed geometries from a single SEM/TM image; two point statistics is more accurate

Table 2.14. Jiao, Stillinger and Torquato [19].

Paper	Modeling heterogeneous materials via two-point correlation functions: Basic principles. (2007).
Authors	Y. Jiao, F. H. Stillinger and S. Torquato
Model Summary	Two-point statistics uses porosity data and an auto-correlation function which provides information about pore connectivity
Region	Porous medium
Assumptions	Isotropic material
Main Conclusions	A geometry can be reconstructed from an image or a distribution equation using the two-point statistics

2.2 Chromium Poisoning

Badwal et al. [6] experimentally investigate the performance of an SOFC as the chromium migrates into the porous cathode. They show that the performance decays rapidly. They also look at images of the cathode after the experiment to see where the chromium poisoning occurs. They find that chromium poisoning happens only when current is applied to the system and that the performance decay is related to current density. A brief summary of this study is given in Table 2.15.

The work of Stanislawski et al. [8] is based on the knowledge that the vaporization of chromium from the interconnect causes the poisoning. They measure the vaporization of chromium in humid air from various alloys and coatings. They find that the alloys and coatings that release the least amount of chromium into the air. Stanislawski et al. [8] also estimate an SOFC's behavior for the various amounts of chromium vaporization associated with the different alloys and coatings. A summary appears in Table 2.16.

Table 2.15. Badwal, Deller, Foger, Ramprakash and Zhang [6].

Paper	Interaction between chromia forming alloy interconnects and air electrode of solid oxide fuel cells (1997).
Authors	S.P.S. Badwal, R. Deller, K. Foger, Y. Ramprakash, and J.P. Zhang
Phenomena Investigated	Interactions between chromium rich alloy and a doped LaMnO ₃ ; materials and interfaces microscopically examined after experiments
Main Conclusions	Performance of electrodes in contact with a chromium containing alloy decays very rapidly; during the experiments crystals rich in Cr and Mn are formed on and in the electrode; speed of degradation and crystal formation was based on current density

Table 2.16. Stanislawski, Wessel, Hilpert, Markus and Singheiser [8].

Paper	Chromium Vaporization from High Temperature Alloys I. Chromia-Forming Steels and The Influence of Outer Oxide Layers. (2007).
Authors	M. Stanislawski, E. Wessel, K. Hilpert, T. Markus and L. Singheiser
Phenomena Investigated	Equilibrium vaporization measurement for Cr ₂ O ₃ in humid air; Cr vaporization in various alloys; estimate of fuel cell degradation rates
Main Conclusions	The partial pressures of chromium oxide vapors measured agree with results in a previously published paper [57]; the alloy with the least chromium vaporization was found; the advantages of coatings were also found

Taniguchi et al. [7] performed early experiments on the degradation of an SOFC with chromium migration into the cathode. Their reasoning for why the degradation occurs is inconsistent with the research and conclusions in Badwal et al. [6], Stanislawski et al. [8], and Paulson and Birss [58]. Taniguchi et al. may have come to incorrect conclusions because they did not use the microscopic imaging employed by the other authors. Their paper is nonetheless useful, because the experiments performed show that the degradation is linear. The deduction from the fuel cell degradation results is that the TPB site blocking is based only on the concentration of the chromium oxide vapor. Therefore, the chromium oxide vapor always reacts when at a TPB instead of the oxygen in the air. This work is summarized in Table 2.17.

Paulson and Birss [58] experimentally look at the distribution of chromium deposits in the electrode. They do a parametric study for different operating conditions to investigate the effects on current. The SEM images taken after the experiment show that the chromium deposits

are formed on the TPBs. The type of deposit depends on the electrochemical reaction that occurs at the site. This study is summarized in Table 2.18.

Table 2.17. Taniguchi, Kadowaki, Kawamura, Yasuo, Akiyama, Miyake and Saitoh [7].

Paper	Degradation phenomena in the cathode of a solid oxide fuel cell with an alloy separator. (1995)
Authors	S. Taniguchi, M. Kadowaki, H. Kawamura, T. Yasuo, Y. Akiyama, Y. Miyake and T. Saitoh
Phenomena Investigated	Cathode degradation in an SOFC using an alloy separator; degradation was investigated for different alloys and operating conditions
Main Conclusions	Cathode polarization increases due to the filling of the pores by chromium which decreases the oxygen flow

Table 2.18. Paulson and Birss [58].

Paper	Chromium Poisoning of LSM-YSZ SOFC Cathodes. I. Detailed Study of the Distribution of Chromium Species at a Porous Single-Phase Cathode. (2004)
Authors	S. C. Paulson and V. I. Birss
Phenomena Investigated	Distribution of chromium deposits in a cathode after being poisoned; current of the SOFC for different operating conditions
Main Conclusions	The chromium deposits form in and around three TPBs which alters the structure of the TPB; electrochemical reactions affect the types of chromium species found throughout the cathode

Stanislowski et al. [9] run a series of experiments on different interconnect materials and coatings in order to determine the configuration with the least amount of chromium oxide vapor released. The materials are tested by heating the sample and submitting it to a constant flow of humid air. The air is collected and chemically analyzed to determine the amount of chromium oxide vapor released. All of the uncoated alloys tested released an unacceptable amount of chromium oxide vapor. The coatings substantially reduced the amount of chromium oxide vapor released, especially those based on Co, Cu and Ni. Table 2.19 summarizes this work.

2.3 Conclusion

The literature review reveals that Kinetic Theory and the LB Method are well-suited to modeling the effects of chromium poisoning since this approach is capable of modeling the actual movement of chromium oxide vapor particles in a porous medium. A major cause of rapid performance decay in the cathode of an SOFC is found to be the chromium oxide vapor entering the cathode and blocking the TPB. In addition, the review discusses the construction of

a 3D digital porous medium from an SEM/TEM image of the porous medium. Two-point statistics is found to be an accurate and simple method for reconstructing the geometry.

Table 2.19. Stanislawski, Froitzheim, Niewolak, Quadackers, Hilpert, Markus and Singheiser [9].

Paper	Reduction of Chromium Vaporization from SOFC Interconnectors by Highly Effective Coatings. (2007)
Authors	M. Stanislawski, J. Froitzheim, L. Niewolak, W. J. Quadackers, K. Hilpert, T. Markus and L. Singheiser
Phenomena Investigated	Amount of chromium oxide vapor released for different interconnect materials and coatings
Main Conclusions	The best metal alloys still release an unacceptable amount of chromium oxide vapor; the metal based coatings tested reduced chromium oxide vapor release by 99%; further work must be done to test the durability of the coatings

Chapter 3: Mathematical Model

Mathematical models are used to predict the behavior of countless systems. They provide developers a way to predict the outcome of an experiment without having to go through the expense of creating a prototype. Parameters can be changed with simple code changes as opposed to expensive material changes until the model predicts a desirable outcome. The draw to modeling as opposed to experimentation is no different for fuel cells. For example, the materials in fuel cells must be resistant to high temperatures and corrosive environments and as a result are specialized and expensive. An accurate model that describes the performance of the fuel cell would be invaluable to researchers: allowing them to iterate different fuel cell models without the expense of building them.

Kinetic Theory and the LB method are used here to predict the effect on performance of reacting mixture flow in the electrodes of an SOFC. This, of course, requires a digital description of the porous medium in the electrodes to which the fluid flow model equations are applied. A presentation of the reacting mixture flow and geometric models developed as well as the numerical approach used to solve the flow model are given in the following sections.

3.1 Kinetic Theory

The derivation of the Boltzmann equation from the Liouville (or Liouville-Hamiltonian) equation is taken from Harris [17]. Systems on a molecular level have a large number of degrees of freedom due to the large amount of particles involved. The system is simplified by assuming each particle has no mass, which eliminates the effects due to vibration and molecular orientation and structure. The most convenient method to arrange the equations of motion that describe the system is in terms of the Hamiltonian, i.e.

$$H(\{\bar{q}\}, \{\bar{p}\}) = E(\{\bar{q}\}, \{\bar{p}\}) \quad (3.1)$$

$E(\{\bar{q}\}, \{\bar{p}\})$ is the total (potential and kinetic) energy of the system and $\{\bar{q}\}$ and $\{\bar{p}\}$ are the collection of spatial coordinates and momenta, respectively. The molecular or microscopic state of the system, once described, can then be used to find the intensive properties of the macroscopic system. A macroscopic property is obtained by calculating the time averaged value of the appropriate microscopic property, which is impossible using equation (3.1) as the

information at a given time is not known and the number of degrees of freedom to solve for is immense.

A way around this is to replace the time averaging with ensemble averaging at a given instance of time. In this approach, the state of the system is described using a probability density function, $F_N \equiv F_N(\{\bar{q}\}, \{\bar{p}\}, t)$, where t is the time and F_N describes the distribution of states of an ensemble of identical systems (N is the number of systems). Knowing F_N at a given instance of time, the ensemble average of a macroscopic property, $G(\{\bar{q}\}, \{\bar{p}\})$, can be found by integrating over the phase space formed by $\prod_{i=1}^N d\bar{q}_i d\bar{p}_i$. However, the assumed equivalency between the time and ensemble averaging, the so-called Ergodic Principle, is not a general principle and must, therefore, be verified with experimental results.

Thus, the details of the distribution functions, F_N , are key to solving for the properties of state of the system. The evolution in state of the system is then described from a statistical mechanical viewpoint by the Liouville equation which tracks the time evolution of the distribution functions, such that

$$\frac{dF_N}{dt} = \frac{\partial F_N}{\partial t} + \sum_{i=1}^N \frac{\partial F_N}{\partial \bar{q}_i} \cdot d\bar{q}_i + \sum_{i=1}^N \frac{\partial F_N}{\partial \bar{p}_i} \cdot d\bar{p}_i = 0 \quad (3.2)$$

where $\partial F_N / \partial t$ describes the local change in time, $\sum_{i=1}^N \partial F_N / \partial \bar{q}_i \cdot d\bar{q}_i$ is a convective term and $\sum_{i=1}^N \partial F_N / \partial \bar{p}_i \cdot d\bar{p}_i$ a forcing term. Equation (3.2) is reversible but still contains an impossibly large amount of information and number of degrees of freedom.

To address this, one first notes that the macroscopic properties that are physically relevant do not depend on averages with respect to the probability density functions, F_N , but rather on those with respect to the reduced probability density functions,

$$F_R(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_R, t) \equiv \int F_N(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_R, \dots, \bar{x}_N, t) d\bar{x}_{R+1} \dots d\bar{x}_N. \quad (3.3)$$

The first two of these are

$$F_1(\bar{x}_1, t) = \int F_N(\bar{x}_1, \dots, \bar{x}_N, t) d\bar{x}_2 \dots d\bar{x}_N \quad (3.4)$$

$$\text{and } F_2(\bar{x}_1, \bar{x}_2, t) \equiv \int F_N(\bar{x}_1, \dots, \bar{x}_N, t) d\bar{x}_3 \dots d\bar{x}_N. \quad (3.5)$$

Now, in order to transform the Liouville equation from one in terms of F_N to one in terms of F_R , the equation is first rewritten using $d\bar{q}_i = \bar{p}_i/m$ and $d\bar{p}_i = \bar{F}_i$ to yield

$$\frac{\partial F_N}{\partial t} + \sum_{i=1}^N \frac{\bar{p}_i}{m} \frac{\partial F_N}{\partial \bar{q}_i} + \sum_{i=1}^N \bar{F}_i \frac{\partial F_N}{\partial \bar{p}_i} = 0, \quad (3.6)$$

$$\text{where } \bar{F}_i = - \sum_{j=1 \neq i}^N \frac{\partial \phi_{ij}}{\partial \bar{q}_i}. \quad (3.7)$$

In this equation, ϕ_{ij} is a two-particle potential function represented by, for example, a hard sphere model, the Sutherland potential, the Leonard-Jones potential or one of the many other possible potential functions [17]. Now, integrating equation (3.6) over the phases $d\bar{x}_{R+1} \dots d\bar{x}_N$ and after considerable algebraic manipulation, the BBGKY (Bogoliubov, Born, H. S. Green, Kirkwood and Yvon) equations are found such that

$$\frac{\partial F_R}{\partial t} + \sum_{i=1}^R \frac{\bar{p}_i}{m} \frac{\partial F_R}{\partial \bar{q}_i} - \sum_{i,j=1}^R \frac{\partial \phi_{ij}}{\partial \bar{q}_i} \frac{\partial F_R}{\partial \bar{p}_i} = (N-R) \int \sum_{i=1}^R \frac{\partial \phi_{i,R+1}}{\partial \bar{q}_i} \frac{\partial F_{R+1}}{\partial \bar{p}_i} d\bar{x}_{R+1} \quad (3.8)$$

Equation (3.8) in fact represents a system of R equations and unlike the Liouville equation, the BBGKY equations are not reversible due to the loss of information in certain simplifications made in the derivation, namely using a hard sphere model, equation (3.7). Again note that the $\partial F_R / \partial t$ term describes the local change in time, $\sum_{i=1}^R \bar{p}_i / m \cdot \partial F_R / \partial \bar{q}_i$ the convective change and $\sum_{i,j=1}^R \partial \phi_{ij} / \partial \bar{q}_i \cdot \partial F_R / \partial \bar{p}_i$ the change due to acceleration. Although the system of equations, equation (3.8), are a simplification over the Liouville equation, they are still not solvable as they are not closed, i.e. the equation in F_R depends on F_{R+1} .

To address this, the BBGKY hierarchy of equations must be derived in such a way that we arrive at an equation in F_R which does not depend on F_{R+1} . Furthermore, of the set of R possible equations, we are interested in particular in a F_1 equation which does not depend on F_2 since this is the equation which has the greatest physical relevance. A method for arriving at this result in the method of Grad [17] which begins by replacing the reduced probability density function (equation (5.3)) with the reduced probability density function given by

$$F_1^\sigma(\bar{y}_1, t) = \int_{D_1} F_N d\bar{y}_2 \dots d\bar{y}_N \quad (3.9)$$

$$F_2^\sigma(\bar{y}_1, \bar{y}_2, t) = \int_{D_2} F_N d\bar{y}_3 \dots d\bar{y}_N \quad (3.10)$$

:

where the dependency of F_R on the phase space coordinates $\bar{x}_R = \{\bar{q}_R, \bar{p}_R\}$ has been replaced with the velocity space coordinates $\bar{y}_R = \{\bar{q}_R, \bar{v}_R\}$ to arrive at the truncated function $F_R^\sigma(\bar{y}_1, \dots, \bar{y}_R, t)$. Thus, the momenta, \bar{p}_R , have been substituted with the individual particle velocities, \bar{v}_R . In addition, D_i is that portion of physical space where no particle is within σ of particle 1 while D_2 is that portion in which $|\bar{q}_2 - \bar{q}_1| \geq \sigma$.

Now, in order to proceed further with the derivation, a set of assumptions called the Boltzmann Gas Limit (BGL) assumptions must be made. They are as follows:

1. Boltzmann Assumptions:
 - a. The fluid density is sufficiently low so that only binary collisions, or collisions involving only two particles, occur.
 - b. Collisions can be thought of as localized in space owing to a sufficiently slow spatial dependence of the gas properties.
 - c. The interparticle potential has a sufficiently short range so that statement 1a. is meaningful.
2. N (the number of particles) goes to infinity.
3. The particle mass, m goes to zero, i.e., particles are assumed to be point masses.
4. The parameter, σ , characterizing the range of intermolecular forces goes to zero.
5. $N\sigma^2$ is constant ($N\sigma^2$ is the mean free path).
6. Nm is constant.

Applying these assumptions, Grad's First equation, i.e., the BBGKY equation for F_1^σ , is developed by integrating the Liouville equation over the space for which no particle is within σ of particle 1. The resulting equation is expressed as

$$\frac{\partial F_1^\sigma}{\partial t} + \bar{v}_1 \frac{\partial F_1^\sigma}{\partial \bar{q}_1} + (N-1) \oint_{S_2} d\bar{v}_2 d\bar{S}_2 (\bar{v}_1 - \bar{v}_2) F_2^\sigma - (N-1) \frac{\partial}{\partial \bar{v}_1} \int_{|\bar{q}_1 - \bar{q}_2| > \sigma} d\bar{y}_2 \phi'_{12} F_2^\sigma = 0 \quad (3.11)$$

where \bar{S}_2 is the surface of the sphere $|\bar{q}_1 - \bar{q}_2| = \sigma$ that describes the influence of particle 1 on particle 2. Furthermore, for $|\bar{q}_1 - \bar{q}_2| > \sigma$, no intermolecular forces are in play and, therefore: $\phi_{12} = \phi'_{12} = 0$. Thus, the last term in equation (3.11) drops out.

Now, introducing a coordinate system situated in a plane perpendicular to the vector $\vec{V} = \bar{v}_2 - \bar{v}_1$, with its origin at \bar{q}_1 and with polar coordinates, r and ε such that r varies between 0

and σ and ε between 0 and 2π , the surface integration in equation (3.11) can be transformed so that this equation becomes

$$\frac{\partial F_1}{\partial t} + \bar{v}_i \cdot \frac{\partial F_1}{\partial \bar{q}_1} = N \int V [F_2(\bar{y}_1, \bar{y}_2^+, t) - F_2(\bar{y}_1, \bar{y}_2^-, t)] d\omega d\bar{v}_2 \quad (3.12)$$

In equation (3.11) $d\omega = r dr d\varepsilon$, $\bar{y}_2^\pm = \{\bar{q}_2^\pm, \bar{y}_2\}$ and $\bar{q}_2^\pm = q_2^\pm(r, \varepsilon)$. Note that F_1^σ and F_2^σ have been replaced with N since the latter is very large and, thus, $N \cong N - 1$.

Finally, using the stosszahlansatz, i.e., the assumption (which in the BGL can be shown to be exact) that

$$F_2(\bar{y}_1, \bar{y}_2, t) = F_1(\bar{y}_1, t) F_1(\bar{y}_2, t) \quad (3.13)$$

when \bar{y}_1 and \bar{y}_2 represent molecules which have not yet collided, and the additional reasonable assumption that F_1 does not vary appreciably over distances of order σ , equation (3.12) can be recast as a closed equation for $f = F_1$ such that

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial \bar{q}_1} (\bar{v}_1 f) = \frac{1}{m} \int \int_{\bar{v}_2 \omega} V [f(\bar{q}_1, \bar{v}_1, t) f(\bar{q}_1, \bar{v}_2, t) - f(\bar{q}_1, \bar{v}_1, t) f(\bar{q}_1, \bar{v}_2, t)] d\omega d\bar{v}_2 \quad (3.14)$$

This is the Boltzmann equation in which \bar{v}_1 and \bar{v}_2 are the pre-collision velocities and \bar{v}_1 and \bar{v}_2 are the post-collision velocities (the collisions are assumed to be binary only). The Boltzmann equation is a non-linear integral-differential equation and as such is very complicated and difficult to solve. In fact, it took researchers 50 years to find the first approximate solution.

The right hand side of the Boltzmann equation is the so-called Boltzmann collision integral,

$$J(f) = \frac{1}{m} \int \int_{\bar{v}_2 \omega} V [\bar{f}_1 \bar{f}_2 - f_1 f_2] d\omega d\bar{v}_2 \quad (3.15)$$

where $\bar{f}_i \equiv f'_i \equiv f(\bar{q}_i, \bar{v}_i, t)$ and $f_i \equiv f(\bar{q}_i, \bar{v}_i, t)$. Recall that ω is the disk in the sphere that represents the influence of a particle and has two coordinates, r and ε , associated with it. Furthermore, using the fact that r is a function of θ (the third spherical coordinate not on the disk), that energy and momentum are conserved, and that the pre- and post-collision velocity states are linked, the collision integral can be rewritten as

$$J(f) = \frac{1}{m} \int \int_{\bar{v}_2 \theta, \varepsilon} B(\theta, V) [f'_1 f'_2 - f_1 f_2] d\theta d\varepsilon d\bar{v}_2 \quad (3.16)$$

$$\text{where } B(\theta, V) = V r \left| \frac{\partial r(\theta, V)}{\partial \theta} \right| \quad (3.17)$$

Note that equation (3.16) (not (3.15)) is restricted to purely repulsive potentials.

The form of the Boltzmann equation in equation (3.14) is only valid for a single fluid species. When multiple species are present, a cross collision term must be introduced. For the case of a binary mixture of species A and B , the Boltzmann equations are now written with both cross and self collision terms such that

$$\frac{\partial f_A}{\partial t} + \bar{v} \frac{\partial f_A}{\partial \bar{q}} = J_{AA}(f) + J_{AB}(f) \quad (3.18)$$

$$\text{and } \frac{\partial f_B}{\partial t} + \bar{v} \frac{\partial f_B}{\partial \bar{q}} = J_{BB}(f) + J_{BA}(f) \quad (3.19)$$

Both types of collision integrals can then be written as

$$J_{ij}(f) = \frac{1}{m} \int \int_{\bar{v}_2, \theta, \varepsilon} B_{ij}(\theta, V) [f'_{i1} f'_{j2} - f_{i1} f_{j2}] d\theta d\varepsilon d\bar{v}_2 \quad (3.20)$$

Note that $i, j = A, B$ and $B_{AB}(\theta, V) = B_{BA}(\theta, V)$.

Although the Boltzmann equation is complicated, it can be used to obtain useful information about a system. By taking the first five moments of f in equation (3.14) the macroscopic fluid mass density (ρ), macroscopic fluid velocity (\bar{u}), internal energy density (ξ), pressure tensor ($\bar{\bar{p}}$) and heat flux vector (\bar{Q}), respectively, can be found. Thus the zeroth moment is expressed as

$$\rho(\bar{q}, t) = \int_{\bar{v}_1} f(\bar{q}, \bar{v}_1, t) d\bar{v}_1 \quad (3.21)$$

the first as

$$\bar{u}(\bar{q}, t) = \frac{1}{\rho(\bar{q}, t)} \int_{\bar{v}_1} \bar{v}_1 f(\bar{q}, \bar{v}_1, t) d\bar{v}_1 \quad (3.22)$$

the second moment as

$$\xi = \frac{1}{2\rho(\bar{q}, t)} \int_{\bar{v}_1} v_0^2 f(\bar{q}, \bar{v}_1, t) d\bar{v}_1 = \frac{3}{2} RT, \quad (3.23)$$

where \bar{v}_0 is the peculiar velocity, $\bar{v}_0 = \bar{v}_1 - \bar{u}(\bar{q}, t)$,

the third as

$$\bar{\bar{P}} = \int_{\bar{v}_1} \bar{v}_0 \otimes \bar{v}_0 f(\bar{q}, \bar{v}_1, t) d\bar{v}_1 = \rho RT \quad (3.24)$$

and finally the fourth as

$$\bar{Q} = \int_{\bar{v}_1} \frac{v_0^2}{2} \bar{v}_0 f(\bar{q}, \bar{v}_1, t) d\bar{v}_1. \quad (3.25)$$

In addition to macroscopic thermofluid properties, the macroscopic conservation equations for mass, momentum and energy can be found from the Boltzmann equation via moments. The balances for mass, momentum and energy conservation found in this way are, respectively,

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho \bar{u})}{\partial \bar{q}} = 0 \quad (3.26)$$

$$\frac{\partial(\rho \bar{u})}{\partial t} + \frac{\partial}{\partial \bar{q}} (\rho \bar{u} \bar{u} + \bar{\bar{P}}) = 0 \quad (3.27)$$

$$\text{and } \frac{\partial}{\partial t} \left[\rho \left(\xi + \frac{u^2}{2} \right) \right] + \frac{\partial}{\partial \bar{q}} \left[\bar{Q} + \bar{u} \cdot \bar{\bar{P}} + \rho \bar{u} \left(\xi + \frac{u^2}{2} \right) \right] = 0 \quad (3.28)$$

Although the Boltzmann equation is useful for calculating the macroscopic properties of fluids, the equation's limitations make it not applicable for dense fluids and Knudsen gases due to the BGL assumptions. Furthermore, many difficulties are encountered in applying this equation to practical flow problems. These difficulties can become manageable when the system is near equilibrium so that the linearized version of the Boltzmann equation (see Harris [17]) can be used. However, even then, a great number of difficulties occur in obtaining solutions for practical problems of interest. To minimize these difficulties, Bhatnager, Gross and Krook [59] and, independently, Weylander [60] introduced a simplified model of the Boltzmann equation for practical flows. The result, the BGK master equation of Kinetic Theory, is discussed in the following section.

3.2 The BGK Master Equation

In an attempt to solve the Boltzmann equation, a number of methods have been proposed. One, the Chapman-Enskog Method [17, 61, 62, 63, 64, 65, 66], uses a direct expansion of the equations for the moments of f (equation (3.14)). The method also introduces an ansatz, which assumes that the time dependence of f is solely through the macroscopic properties ρ , \bar{u} and T .

Another method, Grad's Moment Method [17, 67, 68], is based on expanding f in a set of n -dimensional Hermite tensor polynomials. Both the Chapman-Enskog and Grad's Moment methods as well as the other methods listed in Harris [17] are either not general or not particularly suited for the treatment of generalized flows or fairly complicated in working out the attendant details. In an attempt to create a usable, accurate solution to the Boltzmann equation the BGK master equation was developed. The outline of the derivation of this equation given in the next section is based on Harris [17].

3.2.1 The BGK Model

The BGK model starts with the Boltzmann equation as expressed by

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial \bar{q}_1} (\bar{v}_1 f) = \frac{1}{m} \int \int_{\bar{v}_2, \theta, \varepsilon} B(\theta, V) [f'_1 f'_2 - f_1 f_2] d\theta d\varepsilon d\bar{v}_2 \quad (3.29)$$

and assumes that the system is not too far from equilibrium so that the pre-collision f' 's can be replaced with the local equilibrium (local Maxwellian) terms, i.e.,

$$f'_1 f'_2 = f'_{1LM} f'_{2LM} \quad (3.30)$$

Thus, the collision integral in equation (3.29) can be rewritten as

$$J(f) = \frac{f_{1LM}}{m} \int \int_{\bar{v}_2, \theta, \varepsilon} B(\theta, V) f_{2LM} d\theta d\varepsilon d\bar{v}_2 - \frac{f_1}{m} \int \int_{\bar{v}_2, \theta, \varepsilon} B(\theta, V) f_2 d\theta d\varepsilon d\bar{v}_2 \quad (3.31)$$

This last expression for $J(f)$ can be reduced further down to two important properties of f_{2LM} . The first is that the first five moments of f_{2LM} are equal to those of f_2 and if one imagines the remaining parts of the integrands in equation (3.31) to be expanded in powers of \bar{v}_2 , then those integrals become no more than the sum of moments over f_{2LM} and f_2 . Thus, the first five terms are equal by definition. The second important property of f_{2LM} is that the rest of the terms diminish close to equilibrium so that

$$\int \int_{\bar{v}_2, \theta, \varepsilon} B(\theta, V) [f_{2LM} - f_2] d\theta d\varepsilon d\bar{v}_2 = 0 \quad (3.32)$$

Therefore equation, (3.31) reduces to

$$J(f) = \frac{1}{m} \int \int_{\bar{v}_2, \theta, \varepsilon} B(\theta, V) f_{2LM} d\theta d\varepsilon d\bar{v}_2 (f_{1LM} - f_1) = \nu (f_{LM} - f) \quad (3.33)$$

The subscripts “1” are dropped for convenience and ν , although a function of \bar{v}_1 is assumed to be a constant called the collision frequency which is inversely proportional to the relaxation time for a given process. Finally, the BGK master equation is written as

$$\frac{\partial f}{\partial t} + \bar{v} \frac{\partial f}{\partial \bar{q}} = \nu (f_{LM} - f) \quad (3.34)$$

The local Maxwellian distribution, f_{LM} , can be expressed in terms of initial values for the macroscopic properties ρ , \bar{u} and T (i.e., ρ_o , \bar{u}_o and T_o) as well as perturbations of these values (i.e. $\bar{\rho}$, \bar{u} and \bar{T}). Thus,

$$f_{LM} = \frac{\rho_o}{(2\pi RT_o)^{3/2}} \exp\left(-\frac{\bar{v}^2}{2RT_o}\right) \left(1 + \bar{\rho} + \frac{\bar{v}_o \bar{u}}{RT_o} + \bar{T} \left(\frac{v^2}{2RT_o} - \frac{3}{2}\right)\right) \quad (3.35)$$

Plugging equation (3.35) into equation (3.34) yields

$$\frac{\partial f}{\partial t} + \bar{v} \frac{\partial f}{\partial \bar{q}} = \nu \left(-f + \bar{\rho} + \frac{\bar{v}_o \bar{u}}{RT_o} + \bar{T} \left(\frac{v^2}{2RT_o} - \frac{3}{2}\right)\right). \quad (3.36)$$

As with the Boltzmann equation, the BGK master equation can be expanded to deal with multiple species so that, for example, for a binary mixture, it is written as

$$\frac{\partial f_A}{\partial t} + \bar{v} \frac{\partial f_A}{\partial \bar{q}} = \nu_A (f_{LM,A} - f_A) + \nu_{AB} (f_{LM,AB} - f_A) \quad (3.37)$$

$$\text{and } \frac{\partial f_B}{\partial t} + \bar{v} \frac{\partial f_B}{\partial \bar{q}} = \nu_B (f_{LM,B} - f_B) + \nu_{BA} (f_{LM,BA} - f_B) \quad (3.38)$$

Equations (3.37) and (3.38) have three adjustable parameters: ν_A , ν_B , and $\nu_{AB} = n_B/n_A \cdot \nu_{BA}$ where n_A and n_B are the number densities of species A and B , respectively. The equations for $f_{LM,AB}$ and $f_{LM,BA}$, namely,

$$f_{LM,AB} = n_A \left(\frac{1}{2\pi R_A T_{AB}}\right)^{3/2} \exp\left(-\frac{(v - u_{AB})^2}{2R_A T_{AB}}\right) \quad (3.39)$$

$$\text{and } f_{LM,BA} = n_B \left(\frac{1}{2\pi R_B T_{BA}}\right)^{3/2} \exp\left(-\frac{(v - u_{BA})^2}{2R_B T_{BA}}\right) \quad (3.40)$$

introduce four additional adjustable parameters that are not related to physical parameters: T_{AB} , T_{BA} , \bar{u}_{AB} and \bar{u}_{BA} . These parameters are chosen to improve the accuracy of the model. A reasonable method by Morse [69] for choosing these is to derive the momentum and energy

relaxation equations from both the Boltzmann and BGK equations, resulting in equations for T_{AB} , T_{BA} , \bar{u}_{AB} and \bar{u}_{BA} , i.e.,

$$\bar{u}_{AB} = \bar{u}_{BA} = \frac{m_A u_A + m_B u_B}{m_A + m_B} \quad (3.41)$$

$$T_{AB} = T_A + \frac{2m_A m_B}{(m_A + m_B)^2} \left[(T_B - T_A) + \frac{m_B}{6k} (u_B - u_A)^2 \right] \quad (3.42)$$

$$T_{BA} = T_B + \frac{2m_A m_B}{(m_A + m_B)^2} \left[(T_A - T_B) + \frac{m_A}{6k} (u_B - u_A)^2 \right] \quad (3.43)$$

3.2.2 Discrete and Discretized forms of the Boltzmann and BGK Equations

The discretized form of the Boltzmann equation, i.e.,

$$\frac{\partial f}{\partial t} + \bar{v} \frac{\partial f}{\partial \bar{q}} = J(f) \quad (3.44)$$

is called the lattice Boltzmann equations which are based on the kinetic equations of the form

$$n_i(\bar{x} + \bar{c}_i \Delta t, t + \Delta t) = n_i(\bar{x}, t) + \Delta_i \quad (3.45)$$

In this equation, Δ_i is a collision term, n_i in the occupation number for node j which takes on values of 0 and 1, Δt is a time step, i is the lattice direction, \bar{x} is the position vector of each node j , and \bar{c}_i is the lattice velocity vector connecting the nearest neighbors (see Figure 3.1). As should be evident, equation (3.45) is simply a relation between a function at time step t and time step $t + \Delta t$.

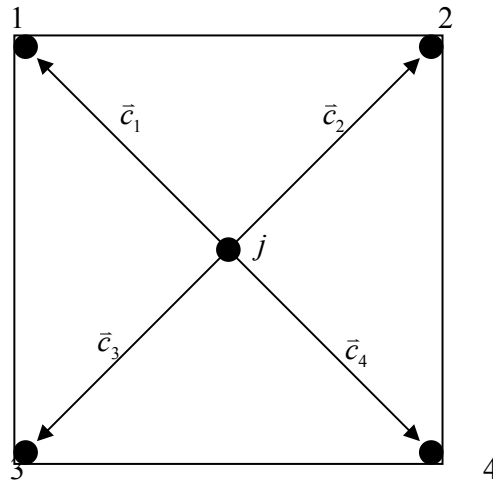


Figure 3.1. Lattice structure diagram.

Showing the correspondence between equation (3.45) and (3.44) can be done by expanding the left hand side of equation (3.45) using a Taylor series expansion, resulting in

$$n_i(\bar{x}, t) + \Delta t \frac{\partial n_i}{\partial t} + \bar{c}_i \Delta t \nabla n_i + O((\Delta t)^2) = n_i(\bar{x}, t) + \Delta_i \quad (3.46)$$

Ignoring the higher order terms and reducing this equation yields the discrete form of the Boltzmann equation, i.e.,

$$\frac{\partial n_i}{\partial t} + \bar{c}_i \nabla n_i = \frac{\Delta_i}{\Delta t} \quad (3.47)$$

which upon substitution of f for n_i , \bar{v} for \bar{c}_i and $J(f)$ for $\Delta_i/\Delta t$ yields the Boltzmann equation.

In a similar fashion, the discrete form of the BGK master equation is found by substituting f_i for f and \bar{v}_i for \bar{v} such that

$$\frac{\partial f_i}{\partial t} + \bar{v}_i \nabla f_i = -\frac{1}{\tau} (f_i - f_i^{eq}) \quad (3.48)$$

Note the changes in notation: $f_{LM} \equiv f^{eq}$, $\nu \equiv 1/\tau$ and $\partial f/\partial \bar{q} \equiv \nabla f$. This equation can be nondimensionalized by using the following characteristic properties: length L , reference speed U , reference number density n_r and time between particle collisions t_c . The resulting nondimensional values are then $\bar{c}_i = \bar{v}_i/U$, $\hat{\nabla} = L\nabla$, $\hat{t} = tU/L$, $\hat{\tau} = \tau/t_c$, $F_i = f_i/n_r$ and $\varepsilon = t_c U/L$. This last value is recognizable as the ratio of collision time to flow time or the Knudsen number (i.e., the mean free path to the characteristic length). Substituting these values into equation (3.48) results in

$$\frac{\partial F_i}{\partial \hat{t}} + \bar{c}_i \hat{\nabla} F_i = -\frac{1}{\hat{\tau} \varepsilon} (F_i - F_i^{eq}) \quad (3.49)$$

Now, using a finite forward difference formula, the discrete nondimensional BGK master equation, equation (3.49), can be discretized so that it is in the form needed by the Lattice Boltzmann Method which is discussed in the next section. This discretization results in the following equation:

$$\begin{aligned} & \frac{F_i(\hat{x}, \hat{t} + \Delta \hat{t}) - F_i(\hat{x}, \hat{t})}{\Delta \hat{t}} + \bar{c}_{ix} \frac{F_i(\hat{x} + \Delta \hat{x}, \hat{t} + \Delta \hat{t}) - F_i(\hat{x}, \hat{t} + \Delta \hat{t})}{\Delta \hat{x}} \\ & + \bar{c}_{iy} \frac{F_i(\hat{x} + \Delta \hat{y}, \hat{t} + \Delta \hat{t}) - F_i(\hat{x}, \hat{t} + \Delta \hat{t})}{\Delta \hat{y}} + \bar{c}_{iz} \frac{F_i(\hat{x} + \Delta \hat{z}, \hat{t} + \Delta \hat{t}) - F_i(\hat{x}, \hat{t} + \Delta \hat{t})}{\Delta \hat{z}} \end{aligned} \quad (3.50)$$

$$= -\frac{1}{\hat{\tau}\mathcal{E}}(F_i - F_i^{eq}) \quad (3.50)$$

where $\Delta\hat{t} = \Delta t U / L$. Equation (3.50) can be simplified to the lattice Boltzmann (BGK master) equation using the definition of the lattice spacing $\Delta\hat{x}/\Delta\hat{t} = \hat{c}_i$, and setting $\Delta t = t_c$. The result is

$$F_i(\hat{x} + \hat{c}_i\Delta\hat{t}, \hat{t} + \Delta\hat{t}) - F_i(\hat{x}, \hat{t}) = -\frac{\Delta\hat{t}}{\hat{\tau}}(F_i - F_i^{eq}) \quad (3.51)$$

3.2.3 The Lattice Boltzmann Method [18]

The lattice BGK master equation is applied over a 1D, 2D or 3D lattice structure that must be appropriately chosen for the problem of interest. Figures 3.2 to 3.4 show commonly used 2D lattices. In Figure 3.2, a D2Q5 lattice is shown with five nodes, hence the designation Q5. As can be seen from this figure, the square geometry and the placement of nodes 1 to 4 signifies that the distances from node 0 to the other nodes, e.g., $\Delta\hat{x}_i = \hat{c}_{i_x}\Delta\hat{t}$ from 0 to 1 or $\Delta\hat{x}_i = \hat{c}_{i_x}\Delta\hat{t}$ from 0 to 3, are all the same. In contrast, Figure 3.3, which shows a 2D lattice with nine nodes in which the distance varies depending on whether it is on the vertical or horizontal or on the diagonal of the square. The distances along the diagonals are, of course, all equal and given by $\Delta\hat{x}_i = \hat{c}_i\Delta\hat{t}/\sqrt{2}$. Figure 3.4 depicts a non-square lattice (D2Q7) in which, similar to the D2Q5, all the distances between nodes are equal.

For 3D applications, a typical lattice structure used is the 3DQ19 lattice shown in Figure 3.5. This is, in fact, the lattice structure used in this thesis work. As can be seen, there are 18 nodes surrounding a single node (P, 0) at the center of the 3D cube. The letter designations next to the node numbers of the simply refer to the directions of the compass and the locations (top, bottom or center (P)) on the cube. For example, SW is southwest, while WT is west-top. As is evident from the figure, there are two possible distances between the central or principal node (P, 0) and the other nodes, e.g., that between P, 0 and E, 1 or P, 0 and ET, 9. The lattice structure actually chosen is based on the medium that is being analyzed and the required details of the solution. Obviously, the more nodes in the system, the more detailed the solution can be.

The three main elements of the LB Method are the lattice speeds, the local Maxwellian equilibrium distributions and the BGK approximation. Equation (3.51) is the BGK approximation and the lattice velocities are $\Delta\hat{x}/\Delta\hat{t} = \hat{c}_i$ (see Table 3.1 for more detail).

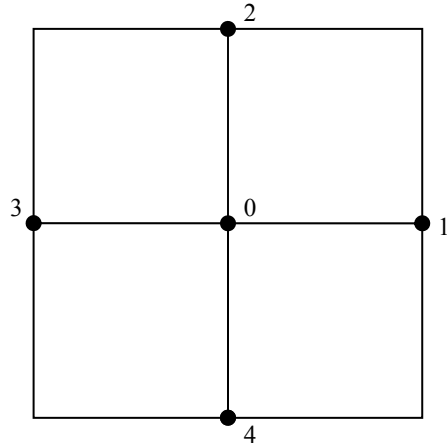


Figure 3.2. Two dimensional, five node (D2Q5) lattice.

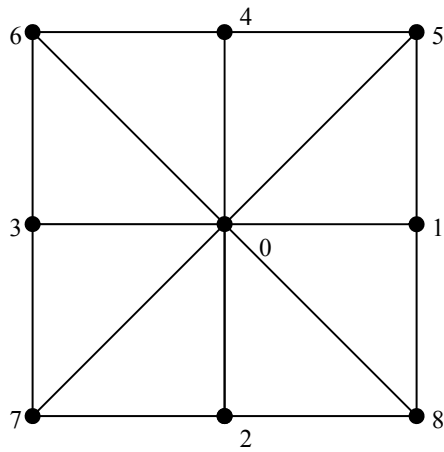


Figure 3.3. Two dimensional, nine node (D2Q9) lattice.

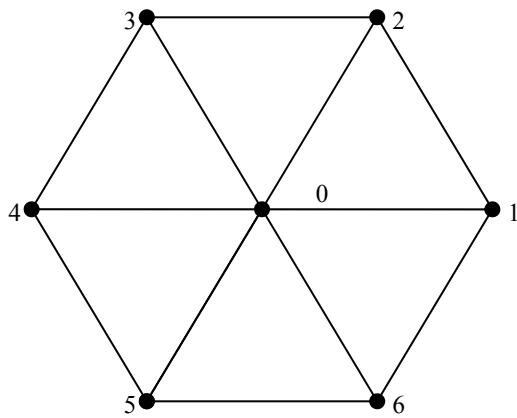


Figure 3.4. Two dimensional, seven node (D2Q7) lattice.

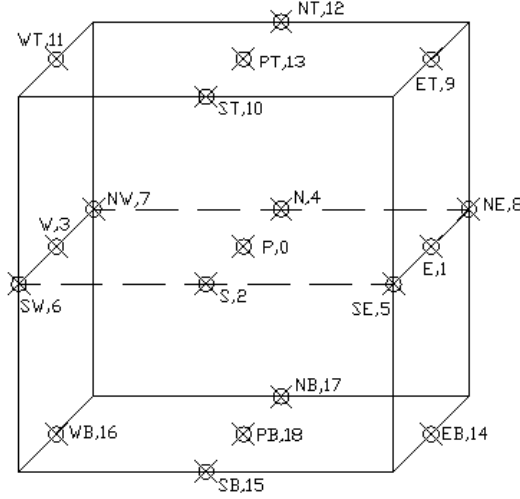


Figure 3.5. Three dimensional, nineteen node (D3Q19) lattice.

In order to find the local Maxwellian equilibrium distributions, F_i^{eq} in equation (3.51), the first two moments of the lattice Boltzmann (BGK master) equation, $F_i(\bar{x}, t)$, are written in terms of the discrete distribution functions (equations (3.21) and (3.22) show the moments written in terms of the Boltzmann equations), i.e.,

$$\rho(\bar{x}, t) = \sum_i F_i(\bar{x}, t) \quad (3.52)$$

$$\text{and } \bar{j}(\bar{x}, t) = \rho(\bar{x}, t) \bar{u}(\bar{x}, t) = \sum_i \bar{c}_i F_i(\bar{x}, t) \quad (3.53)$$

where \bar{c}_i is the particle velocity as before. For a fluid at rest, a global equilibrium distribution, W_i is defined such that, in the vicinity of equilibrium,

$$F_i(\bar{x}, t) = W_i + f_i(\bar{x}, t) \quad (3.54)$$

where $f_i(\bar{x}, t)$ are small perturbations such that $W_i \gg |f_i(\bar{x}, t)|$ and $W_i > 0$ to assure positive $\rho(\bar{x}, t)$. The lattice velocity moments in terms of the Maxwell equilibrium distribution, i.e.,

$$W_B(v) = \rho_o \left(\frac{m}{2\pi kT} \right)^{D/2} \exp \left[\frac{-mv^2}{2kT} \right] \quad (3.55)$$

where m is the particle mass and v is the particle speed, are

$$\sum_i W_i = \int_{\bar{v}} W_B(v) d\bar{v} = \rho_o \quad (3.56)$$

$$\sum_i W_i c_{i,\alpha} = 0 \quad (3.57)$$

$$\sum_i W_i c_{i,\alpha} c_{i,\beta} = \int_{\bar{v}} W_B(v) v_\alpha v_\beta d\bar{v} = \rho_o \frac{kT}{m} \delta_{\alpha\beta} \quad (3.58)$$

$$\sum_i W_i c_{i,\alpha} c_{i,\beta} c_{i,\gamma} = 0 \quad (3.59)$$

$$\begin{aligned} \sum_i W_i c_{i,\alpha} c_{i,\beta} c_{i,\gamma} c_{i,\delta} &= \int_{\bar{v}} W_B(v) v_\alpha v_\beta v_\gamma v_\delta d\bar{v} \\ &= \rho_o \left(\frac{kT}{m} \right)^2 (\delta_{\alpha\beta} \delta_{\gamma\delta} + \delta_{\alpha\gamma} \delta_{\beta\delta} + \delta_{\alpha\delta} \delta_{\beta\gamma}) \end{aligned} \quad (3.60)$$

where $\delta_{i,j}$ is the Kronecker delta and the subscripts, α , β , γ and δ , coincide with coordinate directions x or y . Now, using the D2Q9 lattice structure (Figure 3.3) as an example, the different lattice speeds and corresponding W_i are shown in Table 3.1.

Table 3.1. D2Q9 lattice speeds and global equilibrium distributions.

Lattice nodes	\bar{c}_i	c_i^2	Number of Lattice Nodes	W_i
0	(0,0)	0	1	W_0
1, 3	($\pm c, 0$)	1	2	W_1
2, 4	(0, $\pm c$)	1	2	W_1
5, 6, 7, 8	($\pm c, \pm c$)	2	4	W_2

Applying the data in Table 3.1 to equation (3.56) gives

$$\begin{aligned} \sum_i W_i &= W_0 + W_1 + W_1 + W_1 + W_1 + W_2 + W_2 + W_2 + W_2 \\ &= W_0 + 4W_1 + 4W_2 = \rho_o \end{aligned} \quad (3.61)$$

Expanding equation (3.58) in a similar manner: $\sum_i W_i c_{i,\alpha} c_{i,\beta} = 0$ if $\alpha \neq \beta$ because of the

Kronecker delta so if $\alpha = \beta = x$, equation (3.58) is expanded as

$$\begin{aligned} \sum_i W_i c_{i,\alpha} c_{i,\beta} &= W_0 (c_{0,x} c_{0,x}) + W_1 [(c_{1,x} c_{1,x})_1 + (c_{1,x} c_{1,x})_2 + (c_{1,x} c_{1,x})_3 + (c_{1,x} c_{1,x})_4] \\ &\quad + W_2 [(c_{2,x} c_{2,x})_5 + (c_{2,x} c_{2,x})_6 + (c_{2,x} c_{2,x})_7 + (c_{2,x} c_{2,x})_8] \\ &= W_0 (0 \cdot 0) + W_1 (c^2 + 0^2 + (-c)^2 + 0^2) + W_2 (c^2 + (-c)^2 + c^2 + (-c)^2) \\ &= 2c^2 W_1 + 4c^2 W_2 = \rho_o \frac{kT}{m} \end{aligned} \quad (3.62)$$

Expanding the remaining equations, (3.57), (3.59) and (3.60) and solving for the unknowns, W_i and kT/m , in terms of ρ_o and c yields

$$\frac{W_0}{\rho_o} = \frac{4}{9} \quad (3.63)$$

$$\frac{W_1}{\rho_o} = \frac{1}{9} \quad (3.64)$$

$$\frac{W_2}{\rho_o} = \frac{1}{36} \quad (3.65)$$

$$\frac{kT}{m} = \frac{c^2}{3} \quad (3.66)$$

Now, returning to the lattice Boltzmann equation (equation (3.51)), the local Maxwellian, F_i^{eq} , must be formulated in terms of the local values of mass and momentum densities, i.e., $F_i^{eq} = F_i^{eq}(\rho(\vec{x}, t), \vec{j}(\vec{x}, t))$. This is done by applying the maximum entropy principle in a constrained optimization where mass and momentum conservation are the constraints. Define the relative entropy as in Koelman [70] such that

$$S(\rho, \vec{j}) = -\frac{k}{m} \sum_i F_i^{eq}(\rho, \vec{j}) \ln \frac{F_i^{eq}(\rho, \vec{j})}{W_i} \quad (3.67)$$

Maximize S with respect to F_i^{eq} subject to mass and momentum conservation, i.e.,

$$\rho = \sum_i F_i^{eq}(\rho, \vec{j}) \quad (3.68)$$

$$\text{and } \vec{j} = \sum_i \vec{c}_i F_i^{eq}(\rho, \vec{j}) \quad (3.69)$$

Using the Lagrange method of undetermined multipliers, transform the constrained optimization into an unconstrained one such that

$$\begin{aligned} \mathcal{L} &= S(\rho, \vec{j}) + \tilde{A}\rho + \tilde{\vec{B}} \cdot \vec{j} \\ &= -\frac{k}{m} \sum_i F_i^{eq}(\rho, \vec{j}) \ln \frac{F_i^{eq}(\rho, \vec{j})}{W_i} + \tilde{A} \sum_i F_i^{eq}(\rho, \vec{j}) + \tilde{\vec{B}} \sum_i \vec{c}_i F_i^{eq}(\rho, \vec{j}) \end{aligned} \quad (3.70)$$

where \mathcal{L} is the Lagrange multiplier and \tilde{A} and $\tilde{\vec{B}}$ are Lagrange multipliers. To find the maximum, set the partial derivative of \mathcal{L} with respect to F_i^{eq} equal to zero, i.e.,

$$\frac{\partial \mathcal{L}}{\partial F_i^{eq}} = -\frac{k}{m} \left[\ln \left(\frac{F_i^{eq}}{W_i} \right) + 1 \right] + \tilde{A} + \tilde{\vec{B}} \cdot \vec{c}_i = 0 \quad (3.71)$$

Solving equation (3.71) for F_i^{eq}

$$F_i^{eq} = W_i \exp(A(\rho, \bar{j}) + \bar{B}(\rho, \bar{j}) \cdot \bar{c}_i) \quad (3.72)$$

where $A(\rho, \bar{j}) = (m/k)\tilde{A} - 1$ and $\bar{B}(\rho, \bar{j}) = (m/k)\tilde{B}$. A and B can be determined by a Taylor series expansion of F_i^{eq} about $\bar{j} = 0$, using the assumptions that A is quadratic in \bar{j} and \bar{B} is linear in \bar{j} , i.e.,

$$A(\rho, \bar{j}) = A_0(\rho) + A_2(\rho)\bar{j}^2 + O(\bar{j}^4).. \quad (3.73)$$

$$\bar{B}(\rho, \bar{j}) = B_1(\rho)\bar{j} + O(\bar{j}^3).. \quad (3.74)$$

The Taylor series expansion of F_i^{eq} is then

$$F_i^{eq} = W_i \exp(A_0) \left(1 + \frac{\partial F_i^{eq}}{\partial \bar{j}_\alpha} \bar{j}_\alpha + \frac{1}{2} \frac{\partial^2 F_i^{eq}}{\partial \bar{j}_\alpha^2} \bar{j}_\alpha^2 + \frac{\partial^2 F_i^{eq}}{\partial \bar{j}_\alpha \partial \bar{j}_\beta} \bar{j}_\alpha \bar{j}_\beta + O(\bar{j}^3) \right) \quad (3.75)$$

Substituting the partial derivatives and recognizing that $\bar{j}_\alpha \cdot \bar{j}_\beta = 0$ because \bar{j}_α and \bar{j}_β are orthogonal, equation (3.75) becomes

$$F_i^{eq} = W_i \exp(A_0) \left(1 + B_1 \bar{c}_i \cdot \bar{j} + \left(\frac{B_1}{2} \right)^2 (\bar{c}_i \cdot \bar{j})^2 + A_2 \bar{j}^2 \right) \quad (3.76)$$

A_0 , A_2 and B_1 are still unknown so substitute equation (3.76) into the mass and momentum conservation equations (equations (3.68) and (3.69)) and solve for A_2 and B_1 such that

$$B_1 = \frac{1}{\exp(A_0)} \frac{m}{\rho_o k T} \quad (3.77)$$

$$A_2 = \frac{-B_1^2 k T}{2 m} + \frac{1}{\bar{j}^2} \left(\frac{\rho}{\rho_o \exp(A_0)} - 1 \right) \quad (3.78)$$

However, assuming $A_2 = A_2(\rho)$,

$$\frac{\rho}{\rho_o \exp(A_0)} - 1 = 0 \quad (3.79)$$

$$\text{which leads to } \exp(A_0) = \frac{\rho}{\rho_o} \quad (3.80)$$

$$\text{so that } B_1 = \frac{m}{\rho k T} \quad (3.81)$$

$$\text{and } A_2 = \frac{-1}{2\rho^2} \frac{m}{k T} \quad (3.82)$$

Substituting A_0 , A_2 , B_1 (equations (3.80) through (3.82)), W_i (equations (3.63) through (3.65)) and kT/m (equation (3.66)) into equation (3.76) the local equilibrium distributions for each of the lattice positions become

$$F_0^{eq} = \frac{4}{9} \rho \left[1 - \frac{3 \bar{u}^2}{2 c^2} \right] \quad (3.83)$$

$$F_{1,2,3,4}^{eq} = \frac{1}{9} \rho \left[1 + 3 \frac{\bar{c}_i \cdot \bar{u}}{c^2} + \frac{9 (\bar{c}_i \cdot \bar{u})^2}{2 c^4} - \frac{3 \bar{u}^2}{2 c^2} \right] \quad (3.84)$$

$$F_{5,6,7,8}^{eq} = \frac{1}{36} \rho \left[1 + 3 \frac{\bar{c}_i \cdot \bar{u}}{c^2} + \frac{9 (\bar{c}_i \cdot \bar{u})^2}{2 c^4} - \frac{3 \bar{u}^2}{2 c^2} \right] \quad (3.85)$$

Equations (3.83) through (3.85) are the local equilibrium distributions for the D2Q9 lattice structure only and must be developed separately for any other lattice structure. Now the lattice Boltzmann (BGK master) equation can be applied using equations (3.83) through (3.85) via the following algorithm:

1. Initial conditions or previous time step (Figure 3.6).
2. Collision Step: Solve the lattice Boltzmann (BGK master) equation (equation 3.51) for each node and each lattice direction (Figure 3.7).
3. Streaming Step: “Move” distribution functions to the appropriate neighboring node (Figure 3.8).
4. Calculate the moments of the lattice Boltzmann equations to get the macroscopic system properties (equations (5.52) and (5.53)).
5. Repeat steps 1 through 4 for each time step

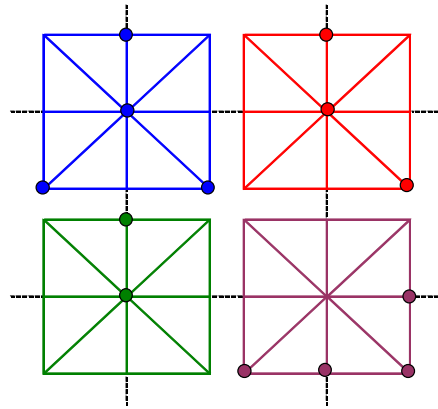


Figure 3.6. Pre-collision lattice.

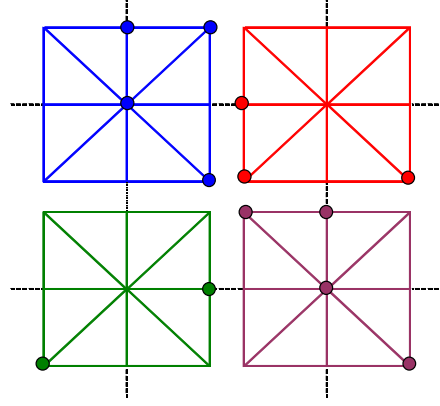


Figure 3.7. Post-collision lattice.

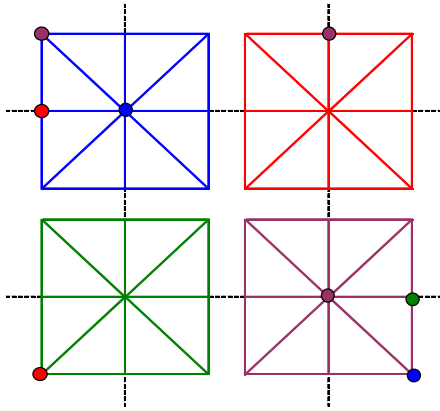


Figure 3.8. Post-streaming lattice.

3.2.4 3D Cathode Reacting Mixture Model

The LB method can be reformulated in matrix form to include multiple relaxation times which improve the stability of the system and allow for easier parallelization. The derivations for the multiple-relaxation-time lattice Boltzmann model can be seen in more detail in Asinari [71] and Asinari et al. [1]. Starting with equation (3.48) in vector form,

$$\frac{\partial \bar{f}_\sigma}{\partial t} + \mathbf{V} \cdot \nabla \bar{f}_\sigma = \mathbf{A}_m (\bar{f}_{\sigma(m)}^e - \bar{f}_\sigma), \quad (3.86)$$

where \mathbf{V} is a $Q \times D$ matrix (e.g., for a D3Q19 lattice the matrix size is 19×3) that contains the lattice velocity components. The notation in this equation is that “ e ” refers to the equilibrium, “ σ ” to the species and “ m ” to the Maxwellian. The species groups that will be considered here are H_2 and H_2O ; O_2 , N_2 and H_2O ; and O_2 , N_2 , H_2O and CrO_3 . Furthermore, \mathbf{A}_m is a diagonal matrix that contains the collision or relaxation frequencies $\lambda_m \equiv \nu_m \equiv 1/\tau_m$, i.e., $\mathbf{A}_m = \lambda_m \mathbf{I}$. In order to increase the number of relaxation frequencies, matrix \mathbf{A}_m can be reformulated as

$$\mathbf{A}_m = \mathbf{M}_D^{-1} \mathbf{D}_m \mathbf{M}_D \quad (3.87)$$

where \mathbf{M}_D is an orthonormal basis for the lattice (see [71] for the development), and \mathbf{D}_m is a diagonal matrix containing the relaxation frequencies. For the D3Q19 lattice, \mathbf{D}_m is

$$\text{diag}(\mathbf{D}_m) = [0, \lambda_m^I, \lambda_m^I, \lambda_m^I, \lambda_{m1}^{II}, \lambda_{m1}^{II}, \lambda_{m1}^{II}, \lambda_{m1}^{II}, \lambda_{m1}^{II}, \lambda_{m2}^{II}, \lambda_m^{III}, \lambda_m^{III}, \lambda_m^{III}, \lambda_m^{III}, \lambda_m^{III}, \lambda_m^{III}, \lambda_m^{IV}, \lambda_m^{IV}, \lambda_m^{IV}] \quad (3.88)$$

The relaxation frequencies are not arbitrary. The λ_m^I are related to the molecular diffusivity, λ_{m1}^{II} and λ_{m2}^{II} to the mixture kinematic and bulk viscosity, respectively, and λ_m^{III} and λ_m^{IV} to the free parameters that affect the model's stability and are usually both set equal to one. See Chapter 4, Section 4.4 on details for finding values for the dimensionless λ_m^I , λ_{m1}^{II} and λ_{m2}^{II} , i.e., $\hat{\lambda}_m^I$, $\hat{\lambda}_{m1}^{II}$ and $\hat{\lambda}_{m2}^{II}$ which are used in equation (3.89) below.

Equation (3.86) is now discretized with respect to time and written in terms of dimensionless quantities as before (see equation (3.51)) so that

$$\bar{f}_\sigma(\hat{t}, \hat{\mathbf{X}}) - \bar{f}_\sigma(\hat{t} - 1, \hat{\mathbf{X}} - \hat{\mathbf{V}}) = \hat{\mathbf{A}}_m [\bar{f}_{\sigma(m)}^e(\hat{t}, \hat{\mathbf{X}}) - \bar{f}_\sigma(\hat{t}, \hat{\mathbf{X}}) + \bar{k}_b(\hat{t}, \hat{\mathbf{X}})] \quad (3.89)$$

The dimensionless quantities are defined using a characteristic length L and two characteristic times t_C and t_S so that: $\hat{t} = t/t_S$, $\hat{\mathbf{X}} = \mathbf{X}/L$ and $\hat{\mathbf{A}}_m = t_C \mathbf{A}_m$. Note that \mathbf{X} , like \mathbf{V} , is a $Q \times D$ matrix that contains the lattice position components and the -1 multiplying \mathbf{V} is the nondimensional time increment, $\Delta \hat{t}$. Likewise, the -1 added to \hat{t} in the same parentheses as $\Delta \hat{t}$. In addition, $\bar{k}_b(\hat{t}, \hat{\mathbf{X}})$ is a proper forcing term that is introduced to satisfy the continuity equations up to the second order in space. Introducing $\bar{k}_b(\hat{t}, \hat{\mathbf{X}})$ improves the stability of the scheme but requires implicit formulation. Fortunately, the quadratic velocity terms do not affect the stability much and, thus, a semi-implicit formulation is possible. See Asinari [71] and Asinari et al. [1] for more details on these formulations.

The equilibrium probability density function, vector $\bar{f}_{\sigma(m)}^e$, can be separated into terms that do not depend, depend linearly or depend quadratically on the mixture velocity. The quadratic term of the equilibrium probability density function, $\bar{f}_{\sigma(m)}^{e2}$, can then be expressed as

$$\bar{f}_{\sigma(m)}^{e2} = \bar{f}_{\sigma(m)}^e - \bar{f}_{\sigma(m)}^{e0} - \bar{f}_{\sigma(m)}^{e1}. \quad (3.90)$$

$$\text{Note that } \bar{f}_{\sigma(m)}^{e0} = \mathbf{M}_{e0} \bar{f}_\sigma \quad (3.91)$$

$$\text{and } \bar{f}_{\sigma(m)}^{e1} = x_\sigma \mathbf{M}_{e1} \sum_{\sigma} \bar{f}_\sigma \quad (3.92)$$

where \mathbf{M}_{e0} and \mathbf{M}_{e1} are the appropriate linear mapping terms that, when multiplied to the equilibrium probability density function, $\tilde{f}_{\sigma(m)}^e$, yields only the terms that do not depend on or linearly depend on, respectively, the mixture velocity. x_σ in equation (3.92) is the species mass fraction. The barycentric probability density function vector (i.e., that based on the barycentric mixture velocity) is then formed by combining equations (3.89) through (3.92) and summing over the species, i.e.,

$$\sum_{\sigma} \tilde{f}_{\sigma}(\hat{t}, \hat{\mathbf{X}}) = \mathbf{B}_S \sum_{\sigma} \tilde{f}_{\sigma}(\hat{t}-1, \hat{\mathbf{X}} - \hat{\mathbf{V}}) + \mathbf{B}_Q \sum_{\sigma} \tilde{f}_{\sigma(m)}^{e2}(\hat{t}-1, \hat{\mathbf{X}}), \quad (3.93)$$

$$\text{where } \mathbf{B}_S = [\mathbf{I} - \hat{\mathbf{A}}_m (\mathbf{M}_{e0} - \mathbf{I}) - \hat{\mathbf{A}}_m \mathbf{M}_{e1}]^{-1} \quad (3.94)$$

$$\text{and } \mathbf{B}_Q = \mathbf{B}_S \hat{\mathbf{A}}_m \quad (3.95)$$

Now that the barycentric probability density function vector, $\sum_{\sigma} \tilde{f}_{\sigma}(\hat{t}, \hat{\mathbf{X}})$, is known, equation (3.93) can be solved for the species probability density functions, $\tilde{f}_{\sigma}(\hat{t}, \hat{\mathbf{X}})$, such that

$$\tilde{f}_{\sigma}(\hat{t}, \hat{\mathbf{X}}) = \mathbf{S}_S \tilde{f}_{\sigma}(\hat{t}-1, \hat{\mathbf{X}} - \hat{\mathbf{V}}) + \mathbf{S}_Q \tilde{f}_{\sigma(m)}^{e2}(\hat{t}-1, \hat{\mathbf{X}}) + x_{\sigma}(\hat{t}, \hat{\mathbf{X}}) \mathbf{S}_C \sum_{\sigma} \tilde{f}_{\sigma}(\hat{t}, \hat{\mathbf{X}}), \quad (3.96)$$

$$\text{where } \mathbf{S}_S = \left[\mathbf{I} - \hat{\mathbf{A}}_m (\mathbf{M}_{e0} - \mathbf{I}) + \frac{\lambda_m^I}{2 - \lambda_m^I} \hat{\mathbf{A}}_m \mathbf{M}_{e1} \right]^{-1} \quad (3.97)$$

$$\mathbf{S}_C = \frac{2}{2 - \lambda_m^I} \mathbf{S}_S \hat{\mathbf{A}}_m \mathbf{M}_{e1} \quad (3.98)$$

$$\text{and } \mathbf{S}_Q = \mathbf{S}_S \hat{\mathbf{A}}_m \quad (3.99)$$

All of the coefficient matrices in equations (3.93) and (3.96) (\mathbf{B}_S , \mathbf{B}_Q , \mathbf{S}_S , \mathbf{S}_Q and \mathbf{S}_C) are dependent only on the dimensionless relaxation frequencies and, thus, need only be calculated once at the beginning of the solution. Equations (3.93) and (3.96) can be applied in the lattice Boltzmann (BGK master) equation algorithm mentioned above in section 3.2.3 in step 2 instead of equation (3.51).

As can be seen from the algorithm described earlier in section 3.2.3, the LB method is very well suited to parallel processing. The collision step for each node is independent of any other nodes and so the LB equations for the nodes can be calculated in parallel. The LB code [52] modified and used in this thesis work uses Message Passing Interface (MPI) for the parallel processing. The LB code is run on System X at Virginia Tech using 32 nodes. System X consists of 1,100 Apple PowerMac G5 computers each consisting of Dual 2.3 GHz PowerPC

970FX processors. The processors are divided among three OSX head nodes and one Linux head node [72].

3.3 Porous Medium 3D Reconstruction

To run the LB code requires a 3D reconstruction of the porous medium of a portion of the SOFC cathode. As indicated in Chapter 2, using a series of SEM/TEM images from slices of the electrode to reconstruct the cathode in 3D is difficult to do and expensive. An alternative approach is to use a statistically based reconstruction of the medium. Two methods are investigated in this research. The first, called one-point statistics is based on the granulometry law, and uses 2D SEM/TEM, which it converts into a digital image by counting different grain sizes relative to a set of shade thresholds. The method then uses the grain-size statistics to reconstruct the porous medium. The second method, two-point statistics, also converts the digital image into a set of shade thresholds. The method then looks at each pixel within a certain threshold and counts the number of pixels around it that are also in that threshold at specified distances. The information about the distances of the pixels in each threshold not only provides information about the porosity of the media but also the connectivity of the pixels. The distance statistics are then used to reconstruct a porous medium. Both methods use phase functions that contain information as to whether or not a space is a pore or a solid.

3.3.1 One-Point Statistics (Granulometry Law) [2]

One-point statistics makes use of the first moment of the phase function only, i.e. the porosity only. A 2D SEM image (Figure 3.9) of the medium is obtained and the phases of the media are separated using a shade threshold, i.e., each phase is given a range of brightness intensity and any pixel in the image that falls within that range is marked as being that phase. In the SOFC cathode active layer the phases are pores, LSM (electron conductor) and YSZ (ion conductor)). Figure 3.10 shows the image in Figure 3.9 after the thresholding process.

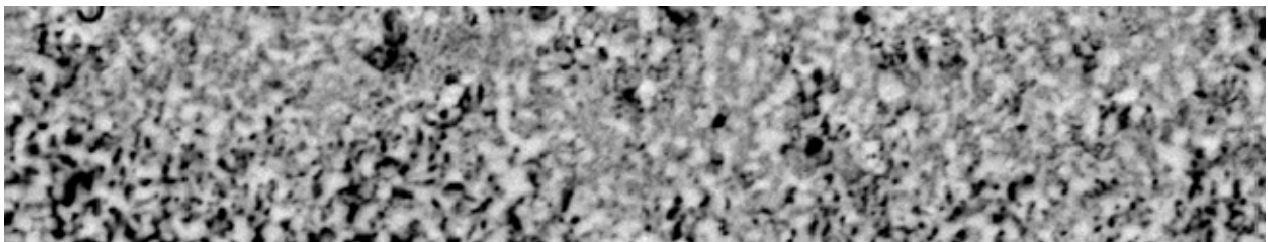


Figure 3.9. SOFC cathode active layer: original image (SEM-BSE).

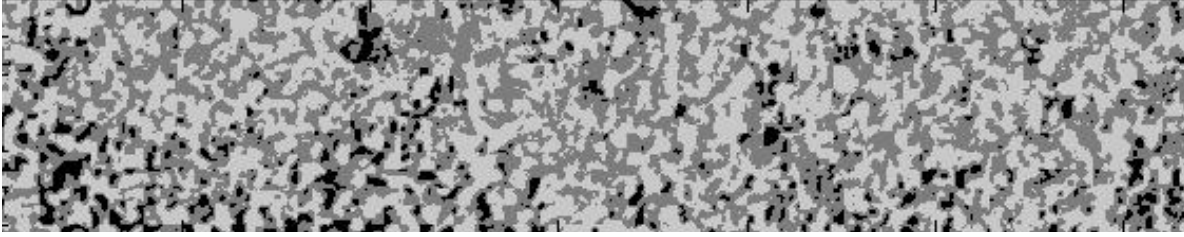


Figure 3.10. SOFC cathode active layer: threshold image, for one-point statistics; black indicates pores, dark grey LSM, light grey YSZ.

Figure 3.10 is visually examined for an appropriate number for the largest grain size. Then following algorithm is applied to the new image (i.e., Figure 3.10):

1. The image is checked for the largest grain size by simply looking at a block of pixels the size of the largest grain and checking the phases within it. If the block consists completely of a single phase, it is counted as a grain for that phase.
2. The grains found are converted to neutral cells (in order to prevent them from being counted again)
3. The total number of grains for each grain size is counted.
4. Steps 1 through 3 are repeated for each subsequent smaller grain size until the grain size is one pixel square.
5. Steps 1 through 4 are repeated for each phase.

From the data produced by application of this algorithm forms a function, $f(d_i)$, that contains the information about the numbers and sizes (d_i) of the grains of all phases (including the pores) (Figure 3.11).

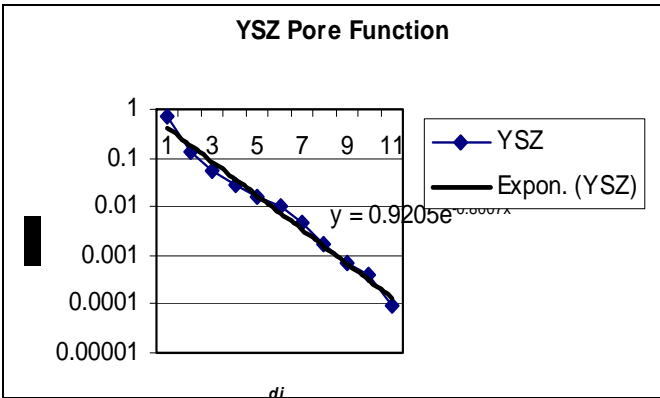


Figure 3.11. One-point statistics function containing pore information, $f(d_i)$.

It is assumed that the 3D reconstructed medium has the same porosity as the 2D image so that the 2D function can be related to the 3D reconstruction. The porosities are defined as

$$\pi_{2D} = \pi_{3D} = \frac{L^3 - V_0}{L^3} \quad (3.100)$$

where L is the size of the 3D medium being reconstructed and V_0 is the total volume of the solid in the 3D reconstruction. Note that V_0 is represented as

$$V_0 = \sum_i d_i^3 N_{i3D} \quad (3.101)$$

where, as above, the d_i are the sizes of the grains and the N_{i3D} are the number of particles of each grain size. N_{i3D} can be related to the function containing information about the grains, $f(d_i)$, by dividing N_{i3D} by the total number of grains, N_{i3D} , i.e.,

$$\frac{N_{i3D}}{N_{i3D}} = f(d_i) \quad (3.102)$$

Combining equations (3.102) and (3.102) yields

$$V_0 = N_{i3D} \sum_i d_i^3 f(d_i) = N_{i3D} K \quad (3.103)$$

Plugging equation (3.103) into equation (3.100) results in

$$\pi_{3D} = \frac{L^3 - N_{i3D} K}{L^3}. \quad (3.104)$$

Solving equation (3.104) for the total number of grains, N_{i3D} , gives

$$N_{i3D} = \frac{1 - \pi_{3D}}{(K / L^3)}. \quad (3.105)$$

N_{i3D} , N_{i3D} and $f(d_i)$, can now be used to produce the 3D reconstruction using the following algorithm:

1. The correct number of grains (using N_{i3D}) of the largest grain size is randomly placed in the 3D domain of interest.
2. Step 1 is repeated for each subsequent smaller grain size until the grain size is one pixel square.
3. Steps 1 and 2 are repeated for each species.

Through trial and error, the one-point statistics method has been found to not be effective for porous media with porosities less than about 22%. The method fails because the porosity is too low to allow all the grains to be placed in the geometry while maintaining the correct porosity for the reconstructed geometry. Figure 12 is a 3D reconstructed image of the active layer of a cathode in an SOFC (see program in Appendix A). As the porosity is very low (about

7%), the image is created using a cut-off so that most of the grains are placed. In other words, the image is only as accurate as the method allows. The porosity of the cathode active layer in this case is taken from a commercial fuel cell made by H. C. Starck (see Appendix B). It is lower than a typical cathode porosity which is about 30% to 40% [73, 74].

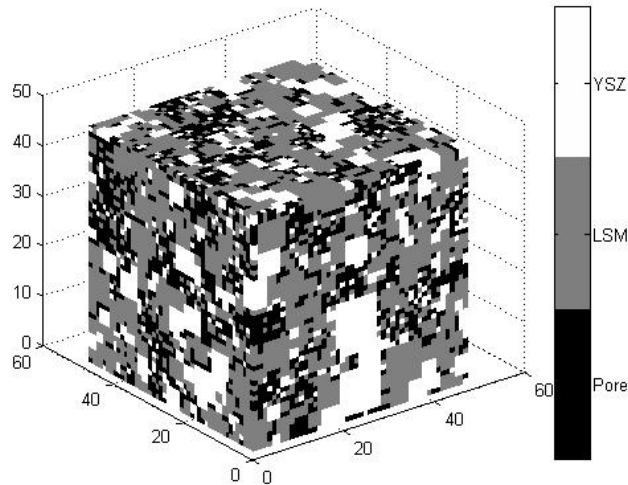


Figure 3.12. SOFC cathode active layer: 3D reconstructed image using one-point statistics.

3.3.2 Two-point Statistics

Two-point statistics [2, 19, 20] makes use of the porosity of the SEM/TEM image as well as a two-point correlation function to reconstruct a 3D geometry. The phase function in this case contains information about how many pixels of a species are a specific distance (and a series of decreasing distances) from pixels of the same species.

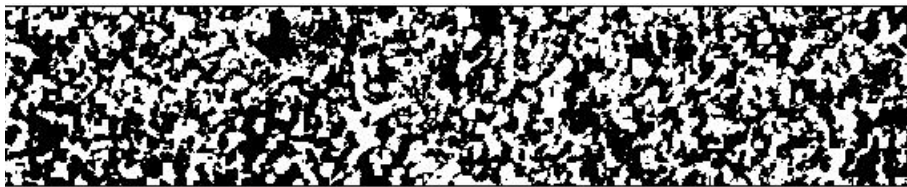


Figure 3.13. SOFC cathode active layer: threshold image (YSZ only), two-point statistics; white indicates YSZ and black pore and LSM.



Figure 3.14. SOFC cathode active layer: threshold image (LSM only), two-point statistics; white indicates LSM and black pore and YSZ.

As with the one-point statistics method, a digital image must be obtained (Figure 3.9) and the species separated using shading thresholds but it must be applied separately for each species, considering all other species as pores (Figures 3.13 and 3.14).

The new digital images are then independently treated with the following algorithm which is explained using the zoomed-in portion of a digital image (e.g., Figures 3.13 or 3.14) at an individual pixel level:

1. Start with the first occupied pixel (red square in Figure 3.15 (b)).
2. Count the number of occupied pixels at a chosen largest distance between pixels along the x- and y-axis of the image. In this thesis work the largest distance is half the pixel dimension of the short edge of the digital image. In Figure 3.15 (b) the largest distance is three pixels. The circled square in Figure 3.15 (b) is the only one counted for the initial occupied pixel.
3. Steps 1 and 2 are repeated for the next and each subsequent occupied pixel (going from left to right, top to bottom of the digital image). Figure 3.15 (c) shows an example of an intermediate pixel where the two circled pixels are counted and the other two directions contain unoccupied pixels.
4. Steps 1 through 3 are repeated for each subsequent smaller distance between pixels until the distance between counted pixels is one pixel (Figure 3.15 (d)).

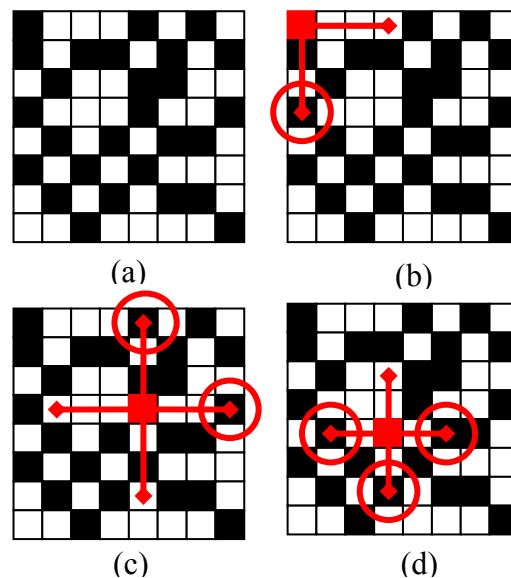


Figure 3.15. Two point statistics algorithm, black: solid, white: pores; (a) zoomed-in portion of the digital image at an individual pixel level, (b) step 1, (c) step 3, (d) step 4.

Applying the algorithm to the digital image yields a distribution that contains the number of solid pixels that are one pixel away from the next solid pixel, the number that are two pixels away, and so on. This algorithm takes only the x- and y- axis directions into account but other algorithms can take into account particles on diagonal lines but this makes the distributions more complicated [19, 20]. The distribution created using the algorithm is the so-called target two-point correlation function, $\hat{S}_2(r)$, where the r is the set of distances between pixels used in the algorithm.

The reconstructed 3D medium is initialized by randomly placing black pixels so that the percent of occupied space is the same as that of the species in the 2D digital image. The so-called sampled two-point correlation function, $S_2(r_i)$, is found for the initial reconstructed medium and is determined from

$$S_2(r) = \text{ave} \left(\frac{B_N(r_i)}{S_N(r_i)} \right) \quad (3.106)$$

where $B_N(r_i)$, obtained using the algorithm described above, contains the number of occupied pixels that are separated from an occupied pixel site, i , by distance r_i and $S_N(r_i)$ are the total possible number of sites that can be occupied at a distance r_i from the reference site, i.e., for the occupied pixel marked in red in Figure 3.15 (b) the total possible occupied sites is 2 whereas for the occupied pixel marked in red in Figure 3.15 (c) the total possible occupied sites is 4. $B_N(r_i)/S_N(r_i)$ is found for each occupied pixel and the average for each distance between pixels, r , is found resulting in $S_2(r)$. The sampled two-point correlation function, $S_2(r)$, is compared to the target two-point correlation function, $\hat{S}_2(r)$, and an error determined,

$$E = \sum_i [S_2(r) - \hat{S}_2(r)]^2 \quad (3.107)$$

The error threshold, or final error, is usually chosen to be very small, $O(10^{-4})$. Figure 3.16 shows the evolution of the sampled two-point correlation function, $S_2(r)$ (in magenta), towards the target two-point correlation function, $\hat{S}_2(r)$ (in blue), where $S_2(r)$ for the initial reconstructed 3D geometry is the left-most and bottom-most line in magenta.

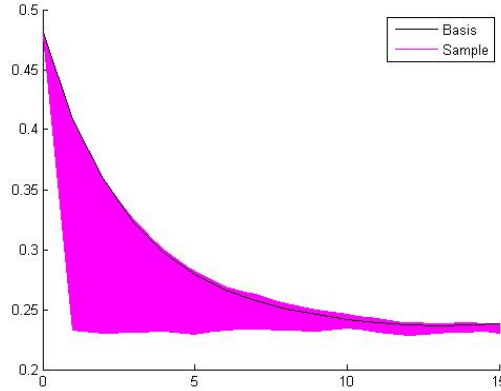


Figure 3.16. Sampled two-point correlation functions, $S_2(r)$, and target two-point correlation function, $\hat{S}_2(r)$.

The following algorithm is used for modifying the reconstructed 3D geometry, starting with the initialized 3D geometry, so that the sampled two-point correlation becomes similar to the target two-point correlation, i.e., so that the error calculated in equation (3.107) is lower than the error before the 3D geometry is modified until the error is below the error threshold.

1. Calculate the sampled two-point correlation function (equation (3.106)) and the error (equation (3.107)) of the initial geometry.
2. Exchange the position of an occupied pixel with that of a non-occupied pixel.
3. Recalculate the sampled two-point correlation function (equation (3.106)) and the error (equation (3.107)).
4. Accept a given geometry when ΔE , the difference between the sampled two-point correlation functions of current and previous geometries, is < 0 or when $\Delta E \geq 0$ and a randomly generated probability, P_R (a number between 0 and 1), is less than or equal to the acceptance probability, $\exp(-\Delta E/T)$. If $\Delta E \geq 0$ and the randomly generated probability is greater than $\exp(-\Delta E/T)$, the geometry is rejected. The acceptance criterion is summarized in equation form as

$$P_A(\text{accept } E_{new}) = \begin{cases} 1, & \Delta E < 0, \\ \exp(-\Delta E/T), & \Delta E \geq 0, \end{cases} \quad (3.108)$$

The basis for choosing the expression $\exp(-\Delta E/T)$ as the acceptance probability, P_A , is that it is an established technique for simulated annealing, i.e., slowly converging towards equilibrium [75]. In this operation, T is a so-called “cooling rate”

and is determined once at the beginning and is set so that initially $\exp(-\Delta E/T) \approx 0.5$. The purpose of the inequality in equation (3.108) is to slow the rate that the error (equation (3.107)) converges toward the threshold to prevent trapping the system in a metastable state [19].

5. Steps 2 through 4 are repeated until the calculated error is less than the error threshold, i.e. $E \leq O(10^{-4})$ (equation (3.107)).

The algorithm above is applied for the digital image with only YSZ (Figure 3.13) and the digital image with only LSM (Figure 3.14). The two 3D reconstructed geometries are then combined to create the 3D reconstructed geometry (Figure 3.17) containing the three phases present in the active layer of an SOFC cathode: pores, LSM and YSZ (see program in Appendix A).

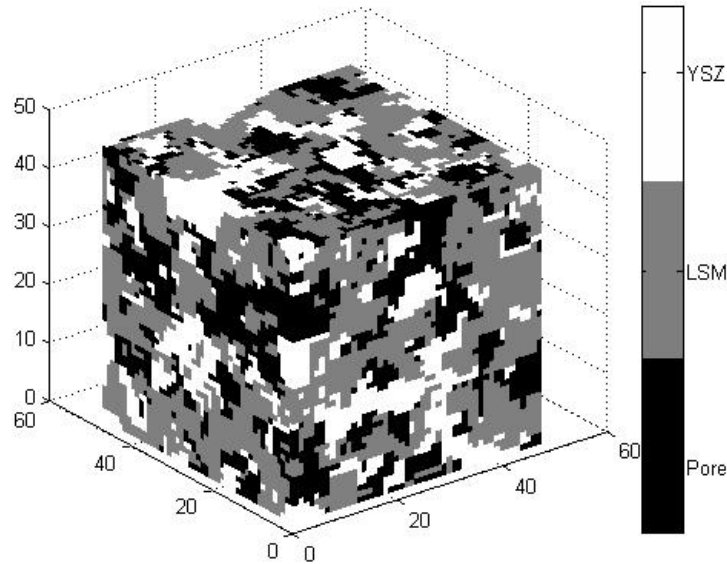


Figure 3.17. SOFC cathode active layer: 3D reconstructed image, two-point statistics.

The two-point statistics method can also be used for generating geometries that are associated with an equation. Instead of finding $\hat{S}_2(r)$ from an image, a specified equation is used. Two-point statistics can also be used for generating sandstone or other porous geological mediums for research.

To validate the two-point statistics 3D reconstruction software developed in this thesis work, 2D slices of the reconstructed 3D geometry shown in Figure 3.17 are visually compared to the geometry created in Kasula et al. [2] (Figure 3.19) in order to verify that the new geometry is reasonable. The anode two-point statistics geometry created by Kasula et al. [2] using the commercial software IMAGOTM used by the petroleum industry has a porosity of 50%, and so a

new anode geometry with a porosity of 50% is created using the program created here to have a geometry with comparable attributes. Figure 3.18 shows 2D slices (one of the inlet/outlet plane and another at an intermediate plane) of the new geometry after the connectivity calculation (see Section 3.3.3).

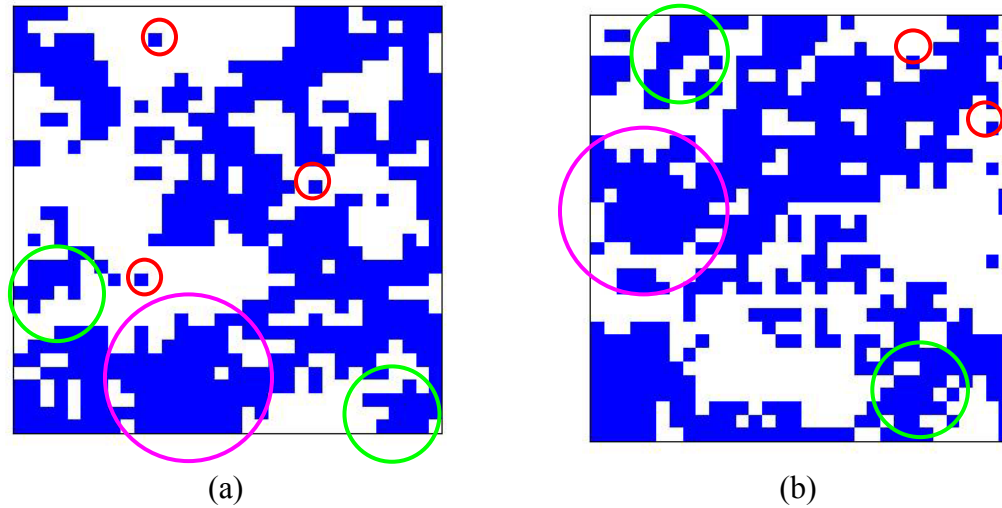


Figure 3.18. Reconstructed geometry with a porosity of 50%; (a) inlet plane; (b) intermediate plane.

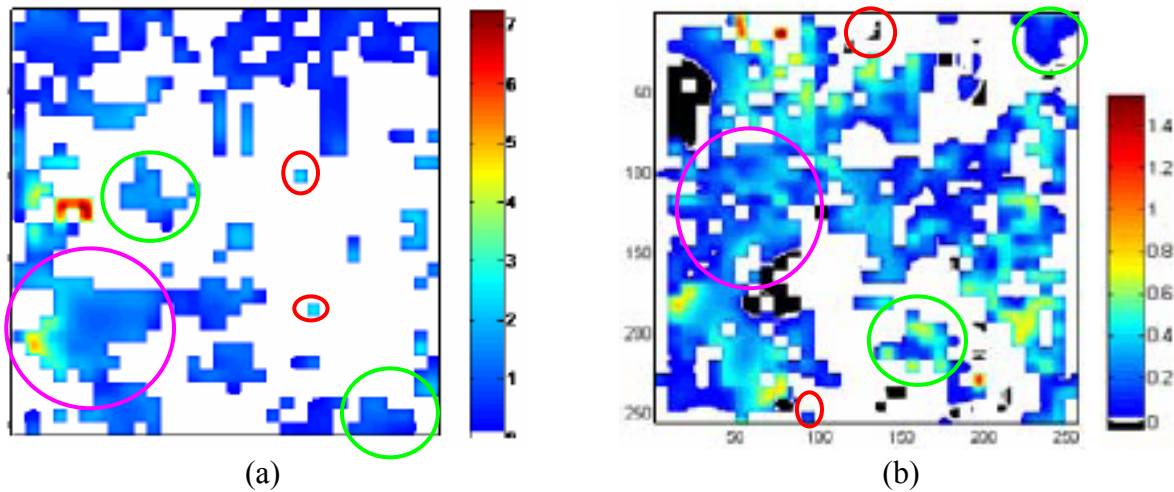


Figure 3.19. Mass flux contours for previously published geometry with a porosity of 50%; (a) inlet plane; (b) intermediate plane [2]⁴.

In Figures 3.18 and 3.19, the white represents solid material while the colored areas represent pores. Although the geometries are different structures and have different pore distributions they

⁴ Reprinted from Proceedings of IMECE 2007, B. V. Kasula, L. Mercado, P. Asinari and M. R. von Spakovsky, 3D microstructure reconstructions of solid oxide fuel cell electrodes with applications to numerical simulations of reacting mixture flows using LBM, 9-10, Copyright (2007), with permission from Dr. Michael von Spakovsky.

have similar characteristics, i.e., pore sizes and shapes. There are areas of small pores (the red circles in Figures 3.18 and 3.19), areas of intermediate sized pores (the green circles in Figures 3.18 and 3.19) and areas of larger pores (the magenta circles in Figures 3.18 and 3.19). The shapes of the pores are also similar in that they are not regularly shaped (such as rectangles or squares) but rather have a seemingly disorganized structure. Note that the different colors in Figure 3.19 represent different values for the mass fluxes of H_2 at the inlet/outlet plane (Figure 3.19 (a)) and an intermediate plane (Figure 3.19 (b)).

3.3.3 Connectivity Calculations

A model containing the information as to statistically where each of the species are does not contain the correct information. The LB code requires a geometric model which contains only the pores, the solid cells and the TPB. A TPB exists only where there is a pore connected to the inlet stream, an electron conductor connected to the interconnect, and an ion conductor connected to the electrolyte. To find pores that are connected to the inlet stream, the pores along the surface of the geometry that are directly exposed to the inlet stream are marked. Then pores in the next layer of the geometry that are connected to the marked pores are marked. The process is repeated throughout the geometry. The same process is used to find the connectivity of the electron and ion conducting materials. The electron connectivity process is started at the plane on the interconnect side of the geometry, and the ion connectivity is started at the plane on the electrolyte side of the geometry. Now that the connected sites for each of the pores, electron conductors and ion conductors are marked independently from those that are not connected the TPB locations can be marked. The connected pore sites that come in contact with connected electron conductor sites and connected ion conductor sites are marked as TPB.

As stated previously, the LB code requires a geometry containing only the fluid (pore) cells, solid cells and TPB so the geometry must be cleaned up. All of the ion and electron conductor cells (both connected and unconnected) are converted to solid cells. The fluid cells not marked as TPB are all (connected and unconnected) converted to simply fluid cells. The bottom plane is marked as all solid cells because the fluid can not flow through the electrolyte. The inlet three phase boundaries and fluid cells are marked separately as inlet cells so that the code can apply the appropriate boundary conditions. The result is a .csv file that is read into the LB code (see program in Appendix A).

Chapter 4: Physical Properties and Geometric Considerations

As with every model, the LB model requires physical parameters that need to be calculated and/or set; and these include the boundary conditions, the relaxation constants, the physical properties of the fluid and the properties of the geometry. The calculation and selection of these parameters are discussed in this chapter.

4.1 Physical Properties

In this section, the calculation and selection of the physical properties are discussed, including the system properties, initial conditions, fluid diffusivity and fluid viscosity. The system properties include typical thermodynamic parameters, such as temperature and pressure and parameters that are specific to the LB method. The initial conditions are the initial fluid densities and velocities.

4.1.1 System Properties and Initial Conditions

The system properties are those which stay constant throughout the simulation. The first of these is pressure. The pressure used in the simulations is standard atmospheric pressure, 101,325 kPa. Next, the temperature is fixed at 1073 K or 800 °C and is in the range of standard operating temperatures for SOFCs that have zirconia in the electrodes [3]. The next property that does not change is φ , which is the ratio of molecular weights that is calculated for each species and is used for tuning the mixture speed of sound, cs . Starting with the ideal gas equation of state for each species, σ , and dividing both sides of this equation by a correction factor, c^2 , yields

$$\frac{p_\sigma}{c^2} = \frac{\rho_\sigma R_\sigma T}{c^2} \quad (4.1)$$

Usually, because the speed of sound is a lattice constant [76],

$$\frac{R_\sigma T_\sigma}{c^2} = \frac{cs^2}{c^2} = \frac{1}{3} \quad (4.2)$$

where cs^2 is the speed of sound squared. In order to tune the mixture speed of sound, the left hand side of equation (4.2) is set to

$$\frac{R_\sigma T_\sigma}{c^2} = \frac{cs^2}{c^2} = \frac{\varphi_\sigma}{3} \quad (4.3)$$

The correction parameter, c , is then chosen such that the φ_σ of the lightest species (by molecular weight) is equal to one. For example, in the case of oxygen, nitrogen and water, water is the lightest species with a molecular weight of 18, and, therefore set $\varphi_{H_2O} = 1$. To find φ_σ for the other species, c is first found from equation (4.3) for water and then this value is used for all of the other species, i.e.,

$$c = \sqrt{\frac{3 R T}{M_{H_2O} \varphi_{H_2O}}} = 1218.95 \text{ m/s} \quad (4.4)$$

$$\varphi_{O_2} = \frac{3 R T}{M_{O_2} c^2} = \frac{3 R T}{M_{O_2}} \frac{M_{H_2O} \varphi_{H_2O}}{3 R T} = \frac{M_{H_2O}}{M_{O_2}} \quad (4.5)$$

Similarly, for nitrogen

$$\varphi_{N_2} = \frac{3 R T}{M_{N_2} c^2} = \frac{3 R T}{M_{N_2}} \frac{M_{H_2O} \varphi_{H_2O}}{3 R T} = \frac{M_{H_2O}}{M_{N_2}}. \quad (4.6)$$

In summary, the φ_σ for a system containing oxygen, nitrogen and water are found in Table 4.1

Table 4.1. Summary of the φ_σ for the water, oxygen and nitrogen mixture [71].

σ	Species	φ_σ
0	Water	1
1	Oxygen	0.563
2	Nitrogen	0.6431

In the same manner, the φ_σ for a system containing oxygen, nitrogen, water and chromium oxide vapor, CrO_3 are summarized in Table 4.2.

Table 4.2. Summary of the φ_σ for the water, oxygen, nitrogen and chromium oxide vapor mixture [71].

σ	Species	φ_σ
0	Water	1
1	Oxygen	0.563
2	Nitrogen	0.6431
3	Chromium oxide vapor	0.1802

There are also geometric parameters such as the size of the geometry. Two cases are considered: a cube 32 pixels by 32 pixels by 32 pixels and another 48 pixels by pixels 48 by 48 pixels where a pixel is 0.0367 μm which is the size of the pixels in the original SOFC cathode

SEM image (see Section 3.3.1, Figure 3.9 and Appendix B). In addition, the porosity, π , takes on three values: 0.07, 0.3 and 0.5 (see Section 4.2.1), and the width, Δx , of a pixel in the SEM image as previously mentioned is 0.0367 μm (see Section 4.4.1). Also needed are the values of an input file which include the refinement (i.e. the number of computational cells used to represent one cell in the geometry) and the reconstructed 3D geometry file name (see Appendix C). Finally, the number of time steps, or collisions, is required. The simulation must run long enough for the system to reach steady state. This is set from run to run.

The initial conditions of the system include the initial densities and fluid velocities. Initially the fluids are at rest relative to their velocities entering the porous medium and, thus, are set to zero. The initial densities, $(\rho_\sigma)_o$, are based on the initial mass fractions, $(\chi_\sigma)_o = (\rho_\sigma)_o / \rho_o$ where ρ_o is the initial mixture density. For nitrogen, the mass fraction is chosen to be that of nitrogen in air. The water is chosen to be that of the humidity of air found in Stanislawski, et al. [8] since this study deals with chromium oxide vapor as well. When the latter is also present in the system, its mass fraction is that found in [8]. The oxygen concentration is simply the remaining fraction. Tables 4.3 and 4.4 summarize these initial mass fractions. Now, given the initial conditions and the system properties, the mixture diffusivity and mixture viscosity can be estimated.

Table 4.3. Initial normalized densities or mass fractions for the water, oxygen and nitrogen mixture [8].

σ	Species	$(\chi_\sigma)_o = (\rho_\sigma)_o / \rho_o$
0	Water	0.011831
1	Oxygen	0.224896
2	Nitrogen	0.763273

Table 4.4. Initial normalized densities or mass fractions for the water, oxygen, nitrogen and chromium oxide vapor mixture [8].

σ	Species	$(\chi_\sigma)_o = (\rho_\sigma)_o / \rho_o$
0	Water	0.0118283
1	Oxygen	0.2249157
2	Nitrogen	0.7632385
3	Chromium oxide vapor	0.0000175

4.1.2 Diffusivity

The mixture diffusivity is calculated based on the individual species diffusivities. The relevant properties of the species are found, beginning with the characteristic lengths of water,

oxygen and nitrogen characterizing the range of intermolecular forces [77], and the characteristic energies / k (k is the Boltzmann's constant equal to 1.3805×10^{-23} J/K) for the same species [77]. It is important that these values are obtained from the same source as the same result can be obtained using a different pair of characteristic lengths and energies / k . These values appear in Table 4.5.

Table 4.5. Relevant properties for calculating mixture diffusivity [77].

σ	Species	$\sigma_{\sigma} (\text{Å})$	$\varepsilon_0/k (K)$
0	Water	2.641	809.1
1	Oxygen	3.467	106.7
2	Nitrogen	3.798	71.4

These characteristic properties for CrO_3 are, however, not so readily available and must thus be estimated. The properties of Cr and O_3 are found individually using the critical temperature, $T_{c,\sigma}$, and pressure, $P_{c,\sigma}$, of each species as well as their acentric factors, ω_{σ} [78]. These are shown in Table 4.6.

Table 4.6. Relevant properties for calculating the chromium oxide vapor characteristic properties [78].

Species	$T_{c,\sigma} (K)$	$P_{c,\sigma} (\text{atm})$	$\omega_{\sigma} (-)$
O_3	261.00	55.0012	0.227
Cr	8560.93	5708.46	-0.2

The characteristic lengths and energies / k of Cr and O_3 can now be found and using the arithmetic and geometric averages, respectively, the characteristic lengths and energies / k of CrO_3 can be found. This method is used in most studies as it is simple and there are no better alternatives (pg 11.9 in Ref. [77] and pg 267 in Ref. [79]). Experimental results show that this method is valid (pg 11.12 in Ref. [77]). For the characteristic lengths, which represent the characteristic Lennard-Jones lengths [77],

$$\sigma_{Cr} = (2.3551 - 0.087\omega_{Cr}) \left(\frac{T_{cCr}}{P_{cCr}} \right)^{1/3} = 2.71565 \text{ Å} \quad (4.7)$$

$$\sigma_{O_3} = (2.3551 - 0.087\omega_{O_3}) \left(\frac{T_{cO_3}}{P_{cO_3}} \right)^{1/3} = 3.92441 \text{ Å} \quad (4.8)$$

$$\sigma_3 = \frac{\sigma_{Cr} + \sigma_{O3}}{2} = 3.32 \text{ \AA} \quad (4.9)$$

and the characteristic energies / k , which represent the characteristic Lennard-Jones energies are [77]

$$\frac{\varepsilon_{Cr}}{k} = (0.7915 - 0.1693\omega_{Cr})T_{cCr} = 6486.1 \text{ K} \quad (4.10)$$

$$\frac{\varepsilon_{O3}}{k} = (0.7915 - 0.1693\omega_{O3})T_{cO3} = 197.744 \text{ K} \quad (4.11)$$

$$\frac{\varepsilon_3}{k} = \sqrt{\frac{\varepsilon_{Cr}}{k} \frac{\varepsilon_{O3}}{k}} = 1132.51 \text{ K} \quad (4.12)$$

Equations (4.9) and (4.12) are the standard equations for handling the characteristic Lennard-Jones lengths and energies, respectively, between two dissimilar molecules [77, 79]. Having found the relevant species properties, the individual diffusivities can be found. The diffusivity equation is [77]

$$D_{ij} = 1.858 * 10^{-3} T^{3/2} \frac{[(M_i + M_j) / M_i M_j]^{1/2}}{P \sigma_{ij}^2 \Omega} \quad (4.13)$$

$$\text{where } \sigma_{ij} = \frac{\sigma_i + \sigma_j}{2} \quad (4.14)$$

and Ω , an accurate relation for the diffusion collision integral [77], is expressed

$$\Omega = \frac{1.06036}{(T^*)^{0.1561}} + \frac{0.193}{\exp(0.47635T^*)} + \frac{1.03587}{\exp(1.52996T^*)} + \frac{1.76474}{\exp(3.89411T^*)}, \quad (4.15)$$

In this last equation

$$T^* = kT / \varepsilon_{ij} \quad (4.16)$$

$$\text{and } \varepsilon_{ij} = \sqrt{\varepsilon_i \varepsilon_j} \quad (4.17)$$

The effective diffusivity can now be calculated using [77]

$$D_{eff} = \left(\sum_{j \neq i} \frac{x_j}{D_{ij}} \right)^{-1} \quad (4.18)$$

The diffusivity values for each mixture and its species are summarized in Table 4.7. For the case of the water, oxygen and nitrogen mixture the indices are 0 for water, 1 for oxygen and 2 for nitrogen. For the case of the water, oxygen, nitrogen and chromium oxide vapor the indices are

0 for water, 1 for oxygen, 2 for nitrogen and 3 for chromium oxide vapor. The effective diffusivities, D_{eff} , are used in the LB code and to calculate the relaxation frequency, λ'_m (see Section 4.4.1).

Table 4.7. Diffusivities for the water, oxygen and nitrogen mixture.

D_{20}	0.0002204 m^2/s
D_{21}	0.0001786 m^2/s
D_{eff}	0.0007616 m^2/s

Table 4.8. Diffusivities for the water, oxygen, nitrogen and chromium oxide vapor mixture.

D_{20}	0.0002204 m^2/s
D_{21}	0.0001786 m^2/s
D_{23}	0.0001229 m^2/s
D_{eff}	0.0007615 m^2/s

4.1.3 Viscosity

The mixture viscosity is calculated in a fashion similar to that of the mixture diffusivity, i.e., the species viscosities are calculated and combined to get the effective mixture viscosity. The densities and dynamic viscosities for N_2 (index 2) and O_2 (index 1) are found from [80], the density and dynamic viscosity for H_2O (index 0) from [81] and [82], respectively, and the density and dynamic viscosity for CrO_3 (index 3) are calculated, the former from the ideal gas equation of state.

Table 4.9. Properties for calculating the mixture viscosity.

Index	Density (kg/m^3)	Dynamic Viscosity ($Pa-s$)
0	0.2021	$382 * 10^{-7}$
1	0.3641	$50.617 * 10^{-6}$
2	0.3188	$42.068 * 10^{-6}$
3	$\frac{P * 101.325 * M_3}{R * T} = 1.13553$	$48.0436 * 10^{-6}$

The dynamic viscosity for CrO_3 is determined as follows [77]:

$$\mu_3 = 26.69 * 10^{-7} \frac{\sqrt{M_3 T}}{\sigma_3^2 \Omega} \quad (4.19)$$

$$\text{where } \Omega = \frac{1.16145}{(T^*)^{0.14874}} + \frac{0.52487}{\exp(0.7732T^*)} + \frac{2.16178}{\exp(2.43787T^*)} \quad (4.20)$$

$$\text{and } T^* = kT / \varepsilon_3 \quad (4.21)$$

In this last equation, T is the system temperature (see Section 4.1.1) and ε_3 / k is the characteristic energy / k found in Section 4.1.2. Table 4.9 summarizes the species densities and kinematic viscosities.

The species kinematic viscosities can now be determined from [77]

$$\nu_i = \frac{\mu_i}{\rho_i} \text{ m}^2 / \text{s} \quad (4.22)$$

so that the mixture kinematic viscosity can be found from [1, 77]

$$\nu_{eff} = \sum_i \frac{x_i \nu_i}{1 + \sum_{j \neq i} \phi_{ij} \frac{x_j M_i}{x_i M_j}} \quad (4.23)$$

In this last equation,

$$\phi_{ij} = \frac{\left[1 + (\mu_i / \mu_j)^{1/2} (M_j / M_i)^{1/4}\right]^2}{\left[8(1 + M_i / M_j)\right]^{1/2}} \quad (4.24)$$

$$\text{and } \phi_{ji} = \frac{\mu_j M_i}{\mu_i M_j} \phi_{ij} \quad (4.25)$$

Tables 4.10 and 4.11 summarize the mixture kinematic viscosities. The indices for the species are the same as used previously for the two mixtures of interest here. The effective (or mixture) kinematic viscosities are those used by the LB code and used to calculate the relaxation frequency, λ_{m1}'' (see Section 4.4.2).

Table 4.10. Kinematic viscosities for the water, oxygen and nitrogen mixture [1, 77].

ν_0	0.0001890 m^2/s
ν_1	0.0001390 m^2/s
ν_2	0.0001320 m^2/s
ν_{eff}	0.00008544 m^2/s

Table 4.11. Kinematic viscosities for the water, oxygen, nitrogen and chromium oxide vapor mixture [1, 77].

ν_0	0.0001890 m^2/s
ν_1	0.0001390 m^2/s
ν_2	0.0001320 m^2/s
ν_3	0.00004265 m^2/s
ν_{eff}	0.00008544 m^2/s

4.2 Geometry

It is important to have as realistic a representation of the 3D geometry of the porous electrode as possible. The reconstruction of this geometry is discussed in detail in Section 3.3 of Chapter 3. It is also important to as realistically as possible determine the number of TPBs and their locations in the geometry consistent with the experimental data. The geometry used in this thesis work is based on the cathode of the ASC1 model SOFC from the high tech materials company, H. C. Starck (see Appendix B).

4.2.1 Porosity

The porosity of the electrode is a defining feature of the reconstruction, which is obtained from the data sheets of the electrode (see Appendix B). Table 4.12 shows the percentages of the electron conductor, porosity and ion conductor of the cathode modeled in this thesis work as well as the electronic and ionic grain sizes. The values in Table 4.12 are used to define the thresholds discussed in Section 3.3 of Chapter 3 so that the reconstructed 3D geometry has the correct percentage of each component of the electrode.

Table 4.12. Cathode active layer properties.

Porosity (%)	7
Electronic Phase (%)	48
Ionic Phase (%)	45
Electronic phase grain size (μm)	0.5
Ionic phase grain size (μm)	0.6

4.2.2 TPBs

In addition to the porosity, the locations of the TPBs are critical features. These sites require an electron conductor with a current path to the interconnect, an ion conductor with a path to the electrolyte and a pore with a path to the inlet flow. As can be seen in Figure 4.1, a TPB is marked at the intersection of an ion path and an electron path at a pore.

Before the lattice Boltzmann simulation can be run the TPB must be marked in the geometry. The marked sites are recognized in the lattice Boltzmann code as sites to apply the electrochemical boundary condition (see Section 4.3.2).

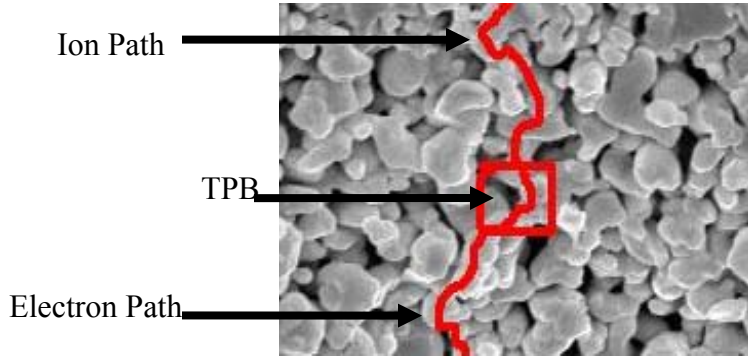


Figure 4.1. SEM SOFC cathode image with a TPB site and its electron and ion path marked.

4.3 Boundary Conditions

The streaming step (Section 3.2.3 of Chapter 3) includes the boundary conditions (B.C.s) that define the movement of the species when they hit a wall. These B.C.s are the same modified bounce-back conditions used in Asinari et al. [1]. The inlet/outlet flow B.C.s and electrochemical B.C.s are applied in the collision step (Section 3.2.3 of Chapter 3).

4.3.1 Inlet and Outlet B.C.s

The flow through the electrode in the LB model is driven by differences in concentrations. The inlet/outlet B.C. (IOBC) is simply a matter of applying a constant normalized concentration for each species at the inlet/outlet plane. The IOBC for the two types of mixtures considered are summarized in Tables 4.13 and 4.14 and repeat the values found in Tables 4.3 and 4.4.

Table 4.13. IOBCs for the oxygen, nitrogen and water mixture.

σ	Species	IOBC $_{\sigma}$
0	Water	0.011831
1	Oxygen	0.224896
2	Nitrogen	0.763273

Table 4.14. IOBCs for the oxygen, nitrogen, water and chromium oxide vapor mixture.

σ	Species	IOBC $_{\sigma}$
0	Water	0.0118283
1	Oxygen	0.2249157
2	Nitrogen	0.7632385
3	Chromium oxide vapor	0.0000175

4.3.2 Electrochemical B.C.s

The electrochemical B.C.s are more complicated than those for the IOBC. The Butler-Volmer relation can be used to describe the microscopic dynamics of the electrochemical reactions. However, the model in this thesis work does not account for the ion and electron dynamics, and, thus, the activation overpotentials used in the Butler-Volmer relation cannot be found. Therefore, the LB code does not take into account the details of the surface electrochemistry in the formulation of the electrochemical B.C.s but simply assumes that the reactions takes place and cause a change in the concentration of the species at each TPB. Thus, the details of the TPB surface events and the cathode material properties are not modeled, i.e., the cathode surface events are assumed to not be the limiting step. Instead, equations (4.26), (4.39) and (4.46) below are applied to account for the concentration changes at each TPB. In addition, the assumption is made that if the concentration of chromium oxide vapor at a given TPB is above a certain threshold, it and not the oxygen species reacts, resulting in the immediate and permanent blockage of the TPB.

Now, the method for accounting for the electrochemical reactions is to modify the local concentrations of the species at the TPBs using tunable parameters and no reaction rates directly [1]. The B.C. equations used at the electrochemical boundaries for the species concentrations (densities), ρ_σ^R , are

$$\rho_{O_2}^R = \rho_{O_2}^0 - \delta_{\rho D}, \quad (4.26)$$

$$\rho_{H_2O}^R = \rho_{H_2O}^0 + \frac{M_{H_2O}}{M_{O_2}} (\delta_{\rho M}) \quad (4.27)$$

$$\text{and } \rho_{N_2}^R = \rho_{N_2}^0 + \frac{M_{N_2}}{M_{O_2}} (\delta_{\rho M}) \quad (4.28)$$

$\delta_{\rho D}$ and $\delta_{\rho M}$ are the freely tunable parameters and the ρ_σ^0 are the species concentrations set by the IOBC (see Table 4.13).

Equations (4.26) through (4.28) are for the case of the water, oxygen and nitrogen mixture. Water and nitrogen are not reacting but still have concentration changes at the TPBs due to the oxygen undergoing the electrochemical reaction at these sites. The relationship between $\delta_{\rho D}$ and $\delta_{\rho M}$ and the volume-averaged mass fluxes are

$$\langle\langle \rho_{O_2} u_{O_2} \rangle\rangle - \chi_{O_2}^0 \langle\langle \rho u \rangle\rangle = k_D \delta_{\rho D} \quad (4.29)$$

$$\text{and } \langle\langle \rho u \rangle\rangle = -k_M \delta_{\rho M} \quad (4.30)$$

where $\chi_{O_2}^0$ is the inlet/outlet mass fraction for oxygen and k_D and k_M are calculated parameters. The volume-averaged mass flux of the oxygen can also be related to the current density, J , of the fuel cell by

$$\langle\langle \rho_{O_2} u_{O_2} \rangle\rangle = M_{O_2} \frac{J}{zF} \quad (4.31)$$

where F is Faraday's constant equal to 96500 C/mol, and z is the number of electrons produced in the electrochemical reaction. For oxygen, $z = 4$, i.e.,



Equations (4.29) and (4.30) can be expanded using equation (4.31) and the fact that the mixture volume-averaged mass flux, $\langle\langle \rho u \rangle\rangle$, is the sum of the species mass fluxes, $\langle\langle \rho_\sigma u_\sigma \rangle\rangle$.

Thus,

$$\langle\langle \rho_{O_2} u_{O_2} \rangle\rangle = \chi_{O_2}^0 \langle\langle \rho u \rangle\rangle + k_D \delta_{\rho D} = M_{O_2} \frac{J}{4F} \quad (4.33)$$

$$\langle\langle \rho u \rangle\rangle = -k_M \delta_{\rho M} = \langle\langle \rho_{H_2O} u_{H_2O} \rangle\rangle + \langle\langle \rho_{O_2} u_{O_2} \rangle\rangle + \langle\langle \rho_{N_2} u_{N_2} \rangle\rangle \quad (4.34)$$

Substitution of equation (4.31) into equation (4.34) results in

$$\langle\langle \rho u \rangle\rangle = -k_M \delta_{\rho M} = M_{O_2} \frac{J}{4F} + \langle\langle \rho_{H_2O} u_{H_2O} \rangle\rangle + \langle\langle \rho_{N_2} u_{N_2} \rangle\rangle \quad (4.35)$$

Now, substitution of equation (4.35) into equation (4.33) yields

$$\chi_{O_2}^0 \left(M_{O_2} \frac{J}{4F} + \langle\langle \rho_{H_2O} u_{H_2O} \rangle\rangle + \langle\langle \rho_{N_2} u_{N_2} \rangle\rangle \right) + k_D \delta_{\rho D} = M_{O_2} \frac{J}{4F} \quad (4.36)$$

and rearranging gives

$$k_D \delta_{\rho D} = (1 - \chi_{O_2}^0) M_{O_2} \frac{J}{4F} - \chi_{O_2}^0 \left(\langle\langle \rho_{H_2O} u_{H_2O} \rangle\rangle + \langle\langle \rho_{N_2} u_{N_2} \rangle\rangle \right) \quad (4.37)$$

Finally, in matrix form, the following system of equations results:

$$\begin{bmatrix} k_D & 0 \\ 0 & -k_M \end{bmatrix} \begin{bmatrix} \delta_{\rho D} \\ \delta_{\rho M} \end{bmatrix} = \begin{bmatrix} (1 - \chi_{O_2}^0) M_{O_2} \frac{J}{4F} - \chi_{O_2}^0 \left(\langle\langle \rho_{H_2O} u_{H_2O} \rangle\rangle + \langle\langle \rho_{N_2} u_{N_2} \rangle\rangle \right) \\ M_{O_2} \frac{J}{4F} + \langle\langle \rho_{H_2O} u_{H_2O} \rangle\rangle + \langle\langle \rho_{N_2} u_{N_2} \rangle\rangle \end{bmatrix} \quad (4.38)$$

In order to find the correct δ_{ρ_D} and δ_{ρ_M} for a specific set of parameters and boundary conditions, values for δ_{ρ_D} and δ_{ρ_M} are guessed and the LB simulation is run. The results from this simulation are then used to find the current density, J , using equation (4.31) and the values of k_D and k_M using equations (4.38). These values for k_D and k_M and a value for J of interest in this research ($J = 0.4 \text{ A / cm}^2$ [1]) are used to find calculated values for δ_{ρ_D} and δ_{ρ_M} using equations (4.38). The new values for δ_{ρ_D} and δ_{ρ_M} are used in the code for further simulations. This type of approach for replacing the Butler-Volmer equation as the B.C. is justified on the basis that the species changes at the TPBs can be reasonably approximated by locally modifying the species concentrations via linear relations such as the ones proposed here (i.e., equations (3.26) through (3.28)).

The electrochemical B.C.s for the case of the water, oxygen, nitrogen and chromium oxide vapor mixture are slightly different as both the oxygen and the chromium oxide vapor can react but only one at a time, i.e., if chromium oxide vapor is present at a TPB and its concentration is above a certain threshold, it reacts and the oxygen does not, otherwise the oxygen reacts at the site. This means, of course, that there must be two sets of B.C.s at each TPB: one for when just the oxygen reacts and one for when just the chromium oxide vapor reacts. The threshold for the chromium oxide vapor's reaction is set to 0.0000170 which is just below the initial condition and IOBC value. Setting the threshold at the IOBC value results in the chromium oxide vapor not reacting because $\rho_{CrO_3} \geq 0.0000175$ happens only at the IOBC locations. Thresholds lower than 0.0000170 yield results with more anomalies in the mass flux curves.

The reaction equations for the mixture with chromium oxide vapor and the oxygen as the reactant are very similar to those of the previous mixture. They are

$$\rho_{O_2}^R = \rho_{O_2}^0 - \delta_{\rho_D} \quad (4.39)$$

$$\rho_{H_2O}^R = \rho_{H_2O}^0 + \frac{M_{H_2O}}{M_{O_2}} (\delta_{\rho_M}) \quad (4.40)$$

$$\rho_{N_2}^R = \rho_{N_2}^0 + \frac{M_{N_2}}{M_{O_2}} (\delta_{\rho_M}) \quad (4.41)$$

$$\text{and } \rho_{CrO_3}^R = \rho_{CrO_3}^0 + \frac{M_{CrO_3}}{M_{O_2}} (\delta_{\rho_M}) \quad (4.42)$$

The equations for the reaction parameters are developed in the same manner as described above for the three species case. The process for refining δ_{ρ_D} and δ_{ρ_M} is also the same but the following system of equations is used instead:

$$\begin{bmatrix} k_D & 0 \\ 0 & -k_M \end{bmatrix} \begin{bmatrix} \delta_{\rho_D} \\ \delta_{\rho_M} \end{bmatrix} = \begin{bmatrix} (1-\chi_{O_2}^0)M_{O_2}\frac{J}{4F}-\chi_{O_2}^0(\langle\langle\rho_{H_2O}u_{H_2O}\rangle\rangle+\langle\langle\rho_{N_2}u_{N_2}\rangle\rangle+\langle\langle\rho_{CrO_3}u_{CrO_3}\rangle\rangle) \\ M_{O_2}\frac{J}{4F}+\langle\langle\rho_{H_2O}u_{H_2O}\rangle\rangle+\langle\langle\rho_{N_2}u_{N_2}\rangle\rangle+\langle\langle\rho_{CrO_3}u_{CrO_3}\rangle\rangle \end{bmatrix} \quad (4.43)$$

The reaction equation for the chromium oxide vapor is



Because the reaction in equation (4.44) both absorbs electrons and releases oxygen ions just as oxygen reaction would at the TPB, the B.C. equations for chromium oxide vapor reacting can be developed in a manner similar to those of oxygen reacting. Note that the chromium oxide product, Cr_2O_3 , is no longer a vapor but a solid which means that once this reaction takes place at a given TPB site, that site is blocked from any further reactions with CrO_3 or with O_2 . Now, for the reaction of equation (4.44), z in equation (4.31) for the chromium oxide vapor is 3, and thus

$$\langle\langle\rho_{CrO_3}u_{CrO_3}\rangle\rangle = M_{CrO_3} \frac{J}{3F} \quad (4.45)$$

The corresponding reaction equations are

$$\rho_{CrO_3}^R = \rho_{CrO_3}^0 - \delta_{\rho_D} \quad (4.46)$$

$$\rho_{O_2}^R = \rho_{O_2}^0 + \frac{M_{O_2}}{M_{CrO_3}} (\delta_{\rho_M}) \quad (4.47)$$

$$\rho_{N_2}^R = \rho_{N_2}^0 + \frac{M_{N_2}}{M_{CrO_3}} (\delta_{\rho_M}) \quad (4.48)$$

$$\text{and } \rho_{H_2O}^R = \rho_{H_2O}^0 + \frac{M_{H_2O}}{M_{CrO_3}} (\delta_{\rho_M}) \quad (4.49)$$

The equations for refining δ_{ρ_D} and δ_{ρ_M} then become

$$\begin{bmatrix} k_D & 0 \\ 0 & -k_M \end{bmatrix} \begin{bmatrix} \delta_{\rho D} \\ \delta_{\rho M} \end{bmatrix} = \begin{bmatrix} (1-\chi_{CrO_3}^0)M_{CrO_3} \frac{J}{3F} - \chi_{CrO_3}^0 (\langle\langle \rho_{O_2} u_{O_2} \rangle\rangle + \langle\langle \rho_{H_2O} u_{H_2O} \rangle\rangle + \langle\langle \rho_{N_2} u_{N_2} \rangle\rangle) \\ M_{CrO_3} \frac{J}{3F} + \langle\langle \rho_{O_2} u_{O_2} \rangle\rangle + \langle\langle \rho_{H_2O} u_{H_2O} \rangle\rangle + \langle\langle \rho_{N_2} u_{N_2} \rangle\rangle \end{bmatrix} \quad (4.50)$$

To find the values for $\delta_{\rho D}$ and $\delta_{\rho M}$ for the separate cases of either chromium oxide vapor reacting or oxygen reacting, the LB simulation must be run twice: once when the chromium oxide vapor always reacts and once when the oxygen always reacts.

The values for $\delta_{\rho D}$ and $\delta_{\rho M}$ for the case where there is no chromium oxide vapor present and for the cases where there is chromium oxide vapor present are given in Tables 4.15 through 4.17.

Table 4.15. Electrochemical reaction parameters for the water, oxygen and nitrogen mixture.

$\delta_{\rho D}$	0.04655
$\delta_{\rho M}$	0.0136

Table 4.16. Electrochemical reaction parameters for the water, oxygen, nitrogen and chromium oxide vapor mixture when the oxygen reacts always.

$\delta_{\rho D}$	0.04647
$\delta_{\rho M}$	0.0000003416

Table 4.17. Electrochemical reaction parameters for the water, oxygen, nitrogen and chromium oxide vapor mixture with the chromium oxide vapor reacting.

$\delta_{\rho D}$	0.000006114
$\delta_{\rho M}$	0.2055

4.4 Relaxation Frequencies

The final case-specific values to be calculated are the non-dimensional relaxation frequencies, $\hat{\lambda}'_m$, $\hat{\lambda}''_{m1}$ and $\hat{\lambda}''_{m2}$, that appear in the lattice Boltzmann model developed in Section 3.2.4 of Chapter 3 for equations (3.93) through (3.99). $\hat{\lambda}'_m$ is related to the diffusivity calculated in Section 4.1.2 and $\hat{\lambda}''_{m1}$ is related to the kinematic viscosity found in Section 4.1.3. $\hat{\lambda}''_{m2}$ is related to the bulk viscosity and is equal to 1 because, although in the low Mach number limit the physical bulk viscosity is meaningless, the bulk viscosity effects the stability of the LB method

and thus is set between the variability range, i.e., $0 < \hat{\lambda} < 2$ [83]. $\hat{\lambda}_m^I$, $\hat{\lambda}_{m1}^{II}$ and $\hat{\lambda}_{m2}^{II}$ are critical for the stability of the system of LB equations.

4.4.1 Relaxation Frequency Related to Diffusivity

The calculations for $\hat{\lambda}_m^I$ require the effective diffusivities in Tables 4.7 and 4.8 to be dimensionless. A characteristic time scale is defined based on the size of the SEM image used to construct the 3D geometry. dx is the smallest size of each particle, which is also the physical size that each pixel in the SEM image represents. From Section 4.1.2

$$dx = 0.0367 \cdot 10^{-6} \text{ m} \quad (4.51)$$

The characteristic time scale, t_c , is then

$$t_c = \frac{dx}{c} = 3.0108 \cdot 10^{-11} \text{ s} \quad (4.52)$$

where c is given by equation (4.4). The dimensionless effective diffusivity can, thus, be written as [1]

$$\hat{D}_{eff} = \frac{D_{eff}}{c^2 t_c} \quad (4.53)$$

To find the relation between the dimensionless effective diffusivity and the relaxation frequency, $\hat{\lambda}_m^I$, the definition of the effective diffusivity given in equation (4.18) is used and for the case without chromium oxide vapor in the system, this diffusivity becomes

$$D_{eff} = \left(\frac{\chi_0}{D_{20}} + \frac{\chi_1}{D_{21}} \right)^{-1} \quad (4.54)$$

Dividing this last equation by $c^2 t_c$ yields

$$\frac{D_{eff}}{c^2 t_c} = \left(\frac{\chi_0}{D_{20}} + \frac{\chi_1}{D_{21}} \right)^{-1} / c^2 t_c \quad (4.55)$$

which in turn can be written as

$$\hat{D}_{eff} = \left(\frac{\chi_0}{\hat{D}_{20}} + \frac{\chi_1}{\hat{D}_{21}} \right)^{-1} \quad (4.56)$$

From [1], the relationship between the individual dimensionless diffusivities and the relaxation frequency, $\hat{\lambda}_m^I$, is

$$\hat{D}_{ij} = \frac{1}{3\hat{\lambda}_m^I} \frac{\chi_i M_i + \chi_j M_j}{M_j} \quad (4.57)$$

Substitution of equation (4.57) into (4.56) results in

$$\hat{D}_{eff} = \left(3\hat{\lambda}_m^I \frac{\chi_0 M_0}{(\chi_2 M_2 + \chi_0 M_0)} + 3\hat{\lambda}_m^I \frac{\chi_1 M_1}{(\chi_2 M_2 + \chi_1 M_1)} \right)^{-1} \quad (4.58)$$

After algebraic manipulation, the expression for $\hat{\lambda}_m^I$ becomes

$$\hat{\lambda}_m^I = \left(\frac{\chi_0 M_0}{(\chi_2 M_2 + \chi_0 M_0)} + \frac{\chi_1 M_1}{(\chi_2 M_2 + \chi_1 M_1)} \right)^{-1} / (3\hat{D}_{eff}) \quad (4.59)$$

The equations for the case when the chromium oxide vapor is present follow the same steps as for equations (4.54) through (4.59). The equation for $\hat{\lambda}_m^I$ in this case is

$$\hat{\lambda}_m^I = \left(\frac{\chi_0 M_0}{(\chi_2 M_2 + \chi_0 M_0)} + \frac{\chi_1 M_1}{(\chi_2 M_2 + \chi_1 M_1)} + \frac{\chi_3 M_3}{(\chi_2 M_2 + \chi_3 M_3)} \right)^{-1} / (3\hat{D}_{eff}) \quad (4.60)$$

Tables 4.18 and 4.19 provide the values found for the dimensionless diffusivity using equations (4.51) through (4.53) and Tables 4.7 and 4.8. The results for the relaxation frequencies using equations (4.59) and (4.60) also appear in these tables.

Table 4.18. Dimensionless diffusivity and $\hat{\lambda}_m^I$ for the water, oxygen and nitrogen mixture.

\hat{D}_{eff}	17.0236
$\hat{\lambda}_m^I$	0.07483

Table 4.19. Dimensionless diffusivity and $\hat{\lambda}_m^I$ for the water, oxygen, nitrogen and chromium oxide vapor mixture.

\hat{D}_{eff}	17.0205
$\hat{\lambda}_m^I$	0.074809

4.4.2 Relaxation Frequencies Related to Viscosity

The process for finding the relaxation constant related to the viscosity, $\hat{\lambda}_{m1}^II$, is similar to that of $\hat{\lambda}_m^I$. The dimensionless effective viscosity is first found from [1]

$$\hat{\nu}_{eff} = \frac{\nu_{eff}}{c^2 t_c} \quad (4.61)$$

where the values of c and t_c are as given above. The values for the effective viscosity, ν_{eff} , can be found in Tables 4.10 and 4.11 for the two mixture cases. The relationship between the dimensionless effective viscosity and the relaxation frequency is [1]

$$\hat{\lambda}_{m1}'' = \frac{1}{3\hat{\nu}_{eff} - 1/2} \quad (4.62)$$

The values found for the dimensionless effective viscosity and the relaxation frequency for the two mixture cases are given in Tables 4.20 and 4.21:

Table 4.20. Dimensionless effective viscosity and $\hat{\lambda}_{m1}''$ for the water, oxygen and nitrogen mixture.

$\hat{\nu}_{eff}$	1.9099
$\hat{\lambda}_{m1}''$	0.1912

Table 4.21. Dimensionless effective viscosity and $\hat{\lambda}_{m1}''$ for the water, oxygen, nitrogen and chromium oxide vapor mixture.

$\hat{\nu}_{eff}$	1.9098
$\hat{\lambda}_{m1}''$	0.19122

Chapter 5: Results and Discussion

The LBM, as applied in this thesis work, solves for the time evolution of the distribution functions for the fluid species in the porous cathode of an SOFC. The macroscopic properties, e.g., the density and the fluid velocities, can be found using the distribution functions as discussed in Sections 3.1 and 3.2 in Chapter 3. The output of the LB code consists of a density .csv file and three mass flux .csv files (one for each spatial dimension: x, y and z) for each fluid and each node used in the computation. The files are combined using a code that creates a matrix with the data placed so that the geometry matches the original geometry (see Appendix D). The size of the matrix and, thus, the lattice structure is simply the product of the refinement factor and the size of the geometry. For example, for a geometry with a size of 32^3 cells and a refinement factor of 4, the lattice structure and as a consequence the matrix containing the results has 128^3 cells or matrix elements. The choice of a refinement factor of 4 as opposed to 6 or 8 [1] is based on the assumption that the trade off between the computational error introduced and the computational time is an acceptable one. This assumption is first based on the results given in Asinari et al. [1] that show that the difference in refinement factors of 4 and 8 is less than 3.5% for all properties and all species. Secondly, initial results at the beginning of this research after the code had been debugged showed that for the two species mixtures the error is small. Thirdly, the increase in computational time is substantial in that a simulation with a refinement factor of 4 takes about an hour to run and post-process, whereas a simulation with a refinement factor of 8 takes about 2 hr to run and post-process. This becomes a significant time difference when, with the debugging and final results, the number of simulations is on the order of 2000. Therefore, in this thesis work, it was assumed that the gain in computational time that permitted running a much larger number of cases due to a refinement factor of 4 instead of 6 or 8 sufficiently offset the maximum computational error which is less than 5% for all properties and all species with the exception of the volume-averaged N_2 mass flux in the 3 and 4 species mixtures. In this case, the error with a refinement factor of 4 turned out to be significantly higher (see Appendix G for preliminary results using a refinement factor of 8). However and unfortunately, this problem was not discovered until the end of this thesis work and will, thus, have to be left for future work to be dealt with.

Looking at the density and mass flux data shows what is happening within the cathode. The density and mass fluxes can also be used to calculate the tortuosity, which is a characteristic property of the fluid dynamics [1]⁵ that helps validate the results. The tortuosity is calculated using the ratio of a reference diffusivity, D , and the fluid diffusivity, D_e [1, 84, 85]

$$\tau = \frac{\pi D}{D_e} = \frac{1/\hat{\lambda}_m^I}{1/\langle \hat{\lambda}_m^I \rangle} = \frac{\langle \hat{\lambda}_m^I \rangle}{\hat{\lambda}_m^I} \quad (5.1)$$

where τ is the tortuosity, $\hat{\lambda}_m^I$ is the relaxation frequency calculated in Section 4.4.1 of Chapter 4 and tabulated in Tables 4.18 and 4.19 and $\langle \hat{\lambda}_m^I \rangle$ is defined as

$$\langle \hat{\lambda}_m^I \rangle = \frac{-dp_\sigma/dz}{\langle \rho_\sigma u_\sigma \rangle - \langle \rho_\sigma u \rangle} \quad (5.2)$$

dp_σ/dz is the change in pressure over the depth of the geometry and is calculated using the derivative of the polynomial that fits the density data, the φ_σ terms found Tables 4.1 and 4.2, and the following expression taken from Asinari et al. [1]:

$$\frac{dp_\sigma}{dz} = \frac{d\rho_\sigma}{dz} \frac{\varphi_\sigma}{3} \quad (5.3)$$

In addition, u_σ in the denominator of equation (5.2) is the species fluid velocity so $\langle \rho_\sigma u_\sigma \rangle$ is the species surface-averaged mass flux ($\langle \rangle$ indicates a surface-averaged property). Following the same notation, $\langle \rho u \rangle$ is the mixture surface-averaged mass flux. The denominator of (5.2) can be rewritten as

$$\langle \rho_\sigma u_\sigma \rangle - \langle \rho_\sigma u \rangle = \langle \rho_\sigma u_\sigma \rangle - \chi_\sigma \langle \rho u \rangle, \quad (5.4)$$

where χ_σ is the mass fraction, i.e.,

$$\chi_\sigma = \frac{\rho_\sigma}{\rho}. \quad (5.5)$$

The mixture properties, $\langle \rho u \rangle$ and ρ , are found by simply summing the respective species properties.

⁵ The tortuosity is a fluid-dynamic (as opposed to a strictly geometric) parameter which depends on the flow path of a particular species and, thus, is an indicator of how the species interacts with a given porous medium (see pp 373 of Asinari et al. [1]).

In order to verify the results, several different simulations are run with a refinement factor of 4 for 3000 collisions each (with the exception of Geometry B which is run for 100000 collisions). The latter allows all of the cases in this parametric study to reach steady state. Simulations are run with 3 different mixture cases (water and hydrogen; water, oxygen and nitrogen; and water, oxygen, nitrogen and chromium oxide vapor) and the 7 different geometries listed in Table 5.1.

Table 5.1. Descriptions of the geometries used in this thesis work.

Geometry	Description
Geometry A	one-point statistics, 50% porosity, 32^3 geometry developed in Asinari et al. [1] and Kasula et al. [2]
Geometry B	two-point statistics, 50% porosity, 32^3 geometry developed in Asinari et al. [1] and Kasula et al. [2]
Geometry C	7% porosity, 32^3 geometry developed in this thesis work
Geometry D	30% porosity, 32^3 geometry developed in this thesis work
Geometry E	50% porosity, 32^3 geometry developed in this thesis work
Geometry F	7% porosity, 48^3 geometry developed in this thesis work
Geometry G	30% porosity, 48^3 geometry developed in this thesis work

The hydrogen and water mixture simulations are compared to the results published in Asinari et al. [1] and Kasula et al. [2] in order to verify the validity of the geometries reconstructed in this thesis work. Next, the results for several cathode geometries for the mixture case with three species, i.e., water, oxygen and nitrogen, are presented and discussed and compared to the results using the hydrogen and water mixture. Then the results that include the chromium oxide vapor in the mixture for the different cathode geometries are presented, discussed and compared to the mixture cases without the chromium oxide vapor. Finally, the effects of chromium poisoning are examined in a series of density profiles at different time steps.

5.1 Validation, H₂O and H₂ mixture

In order to help validate the modifications to the LB code made in this thesis work, the simulation results presented in Asinari et al. [1] and Kasula et al. [2] are reproduced here using the modified code. Asinari et al. model the anode of an SOFC with a mixture of H₂ and H₂O. H₂ is consumed in the electrochemical reactions and H₂O is released. The initial and inlet/outlet plane mass fractions are 10% H₂ and 90% H₂O. Kasula et al. [2] digitally recreate an SEM

image of an SOFC anode and then, using a one-point statistics in-house code as well as a two-point statistics commercial code called IMAGOTM, reconstruct two separate 3D geometries of the SOFC anode. The results of both reconstructions are presented in Kasula et al. [2] and Asinari et al. [1] and are digitally available for this thesis work.

5.1.1 2 Species Mixture for Geometries A and B

Table 5.2 provides a summary of the volume-averaged results using Geometry A (see Table 5.1). The mass fluxes are normalized using the inlet mixture density $\rho_0 = 1 \text{ kg m}^{-3}$ and the correction factor, c , defined in equation (4.4) in Chapter 4. The densities are normalized using the inlet mixture density ρ_0 . The depth of the electrode (the x-axis of the surface averaged tortuosity, density and mass flux plots) is represented in lattice units, or the dimensionless units discussed in Section 3.2.4 in Chapter 3. The normalization factors are the same for all graphs and contour plots shown in this chapter and Appendix E. The tortuosities listed in the tables include the range of the tortuosity which incorporates the spikes that are typical to the tortuosity curves (for example see Figure 5.1 (a)).

Table 5.2. Volume-averaged property summary for the results from this thesis work using Geometry A and the H₂O and H₂ mixture.

Property	H ₂ O	H ₂
$\langle\langle\rho_\sigma u_\sigma\rangle\rangle/\rho_0 c_0$	-7.6575e-6	2.0726e-6
$\langle\langle\rho_\sigma\rangle\rangle/\rho_0$	0.9058	0.09944
τ	4.5777 +/- 22.1	1.5132 +/- 0.828

Table 5.3. Volume-averaged property summary for the results from Kasula et al. [2] using Geometry A and the H₂O and H₂ mixture.

Property	H ₂ O	H ₂
$\langle\langle\rho_\sigma u_\sigma\rangle\rangle/\rho_0 c_0$	-10.5e-6	3.92e-6
τ	2.2733	

The values for the volume-averaged mass fluxes from the results of this thesis work (Table 5.2) are similar to those reported in Kasula et al. [2] (Table 5.3). The tortuosity in Table 5.3 is the average of that for both species. The average of both species tortuosities in Table 5.2 is 1.9537. The difference in the species averages tortuosities in Table 5.2 and Table 5.3 are due to the fact that some of the model parameters discussed in Chapter 4 are unavailable for this thesis work,

e.g., Δx (Section 4.1.1 of Chapter 4), $\hat{\lambda}_m^I$, $\hat{\lambda}_{m1}^{II}$ and $\hat{\lambda}_{m2}^{II}$ (Section 4.4 of Chapter 4). The value for Δx used in this thesis work to obtain results using Geometries A and B (Table 5.1) was taken from previous versions of the LB code and the values for $\hat{\lambda}_m^I$, $\hat{\lambda}_{m1}^{II}$ and $\hat{\lambda}_{m2}^{II}$ were calculated based on the operating conditions (temperature and pressure) used in this thesis work. See Appendix E for density, mass flux and tortuosity graphs (Figures E.1 through E.8) for the hydrogen and water mixture in Geometry A.

Figures 5.1 (a), 5.2 and 5.4 (a) show the results from this thesis work along the thickness of the electrode (the x-axis) for the H₂ and H₂O mixture Geometry B (Table 1.1). The first four planes on the left side of the graphs are not shown as they represent the inlet/outlet plane and have a zero mass flux condition imposed. The last four planes on the right-hand side of the graphs are also not shown as they are a solid plane representing the electrolyte boundary.

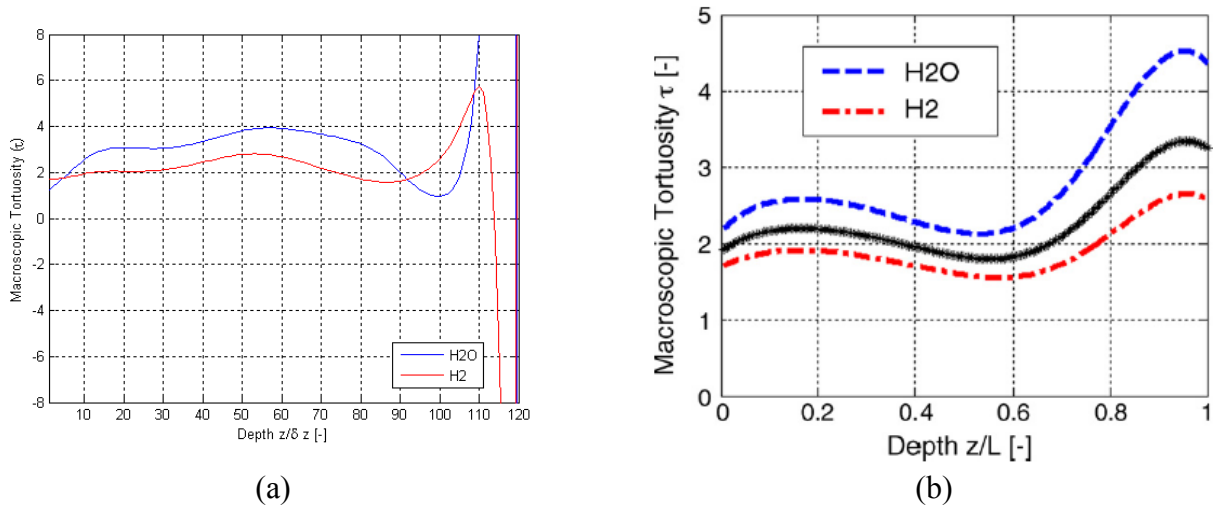


Figure 5.1. Comparison of H₂ and H₂O species τ variations as a function of depth using Geometry B for (a) the results from this thesis work; (b) the results from Kasula et al. [2]⁴.

The tortuosities from this thesis work (Figure 5.1 (a)) (the average of the species volume-averaged tortuosity is 3.9028) show similar trends to those reported in Asinari et al. [1] (Figure 5.1 (b)) (the average of the species volume-averaged tortuosity is 2.2733). Removing the spikes from Figure 5.1 (a) in the same manner as in Figure 5.1 (b) the volume-averaged, species-averaged tortuosity is 2.563 which is very similar to that from Figure 5.1 (b). The black line shown in Figure 5.1 (b) is the average of the species tortuosities. Both plots show the water above the hydrogen and tortuosities around 2. Any differences in the plots are due to the different model parameters as mentioned previously. The tortuosity plots in Figure 5.1 (a) also

show more of the depth of the electrode whereas the tortuosity plots in Asinari et al. [1] (Figure 5.1 (b)) show only a portion of the depth of the electrode. The negative tortuosities are due to the fact that they are not purely a geometric property in this thesis work but depend on the flow of the fluids. For example, the tortuosity will be affected by changes in the slope of the density. Also the spikes in the tortuosity plots are due to the definition of the tortuosity in this thesis work (equation (5.1)) which is no longer useful as the mass fluxes approach zero. In other words, since the denominator of the tortuosity calculation is related to the mass flux and as the mass flux approaches zero, the tortuosity tends to infinity.

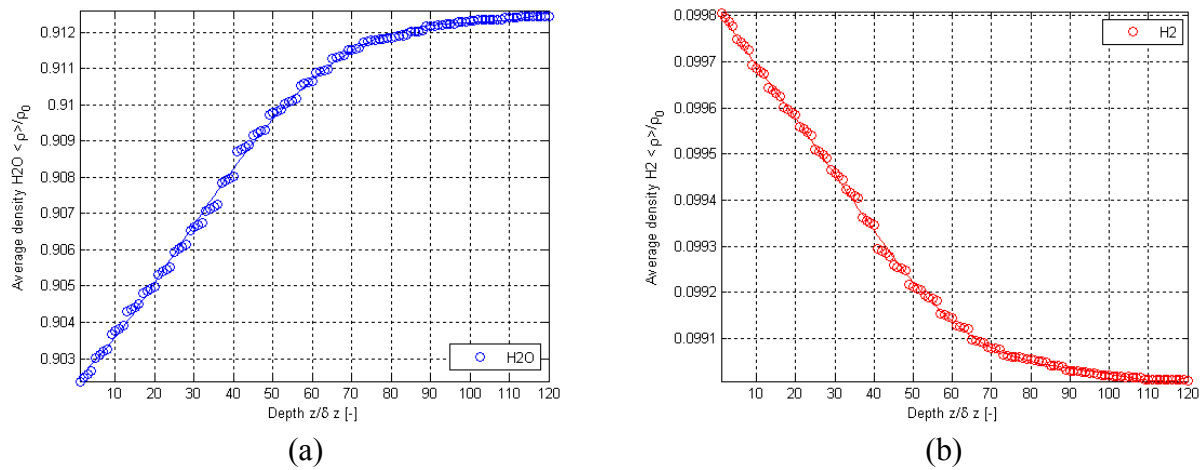


Figure 5.2. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O and H₂ mixture for (a) H₂O and (b) H₂.

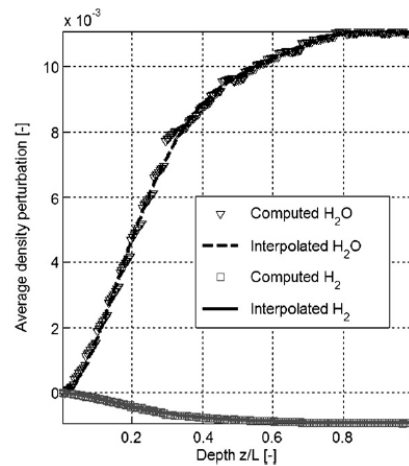


Figure 5.3. Normalized surface-averaged density variations from the results from Asinari et al. [1] as a function of depth using Geometry B for the H₂O and H₂ mixture⁶.

⁶ Reprinted from Journal of Power Sources, 170, Author(s), P. Asinari, M. C. Quaglia, M. R. von Spakovsky and B. V. Kasula, 372, Copyright (2007), with permission from Dr. Michael von Spakovsky.

The density (Figures 5.2 (a) and 5.2 (b)) and mass flux (Figure 5.4 (a)) graphs from this thesis work show similar trends to those from Asinari et al. [1] (Figures 5.3 and 5.4 (b), respectively). The densities shown in Figure 5.3 are the so-called density perturbations i.e., the change in the density from the inlet/outlet plane density throughout the depth of the electrode. Note that in Figure 5.2 (a) the scale is larger than that in Figure 5.2 (b) indicating a greater change in the density throughout the depth of the electrode which corresponds to the results found in Figure 5.3 from Asinari et al. [1]. The increase in density of the water into the electrode shown in Figure 5.2 (a) is caused by water being released by the electrochemical reactions. Likewise, the decrease in density of the hydrogen shown in Figure 5.2 (b) is due to hydrogen being consumed in the electrochemical reactions.

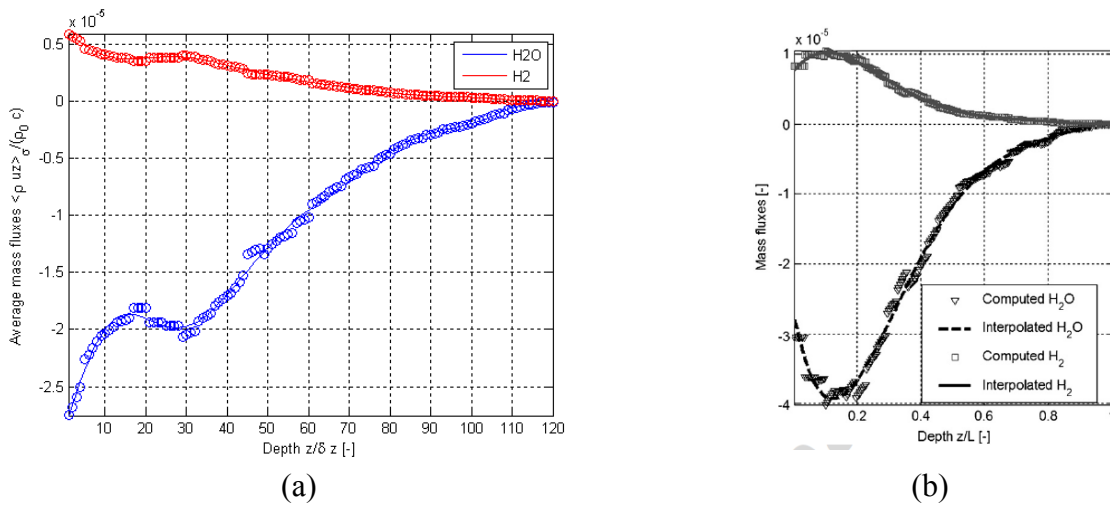


Figure 5.4. Comparison of H₂ and H₂O species normalized surface-averaged mass flux variations as a function of depth using Geometry B for the H₂O and H₂ mixture for the results from (a) this thesis work and (b) Asinari et al. [1]⁶.

In Figure 5.4, water has a negative mass flux because it is being created in the electrochemical reactions and so has a general outward flow. Hydrogen has a positive mass flux because hydrogen is driven into the electrode by the concentration changes due to the electrochemical reactions. The mass flux trends for the results from this thesis work (Figure 5.4 (a)) and the results from Asinari et al [1] (Figure 5.4 (b)) are very similar, especially after the first 20 planes of the results from this thesis work (Figure 5.4 (a)).

Figures 5.5 to 5.8 show comparisons of the results in this thesis work and in Kasula et al. [2] for the mass flux contours for H₂O and H₂ at two different planes. The fluxes are normalized in the same manner as in the previous mass flux plots. The mass flux contour results from

Kasula et al. [2] are normalized with an addition factor of 10^6 . White represents the solid areas in the geometry. Red indicates a more positive flow, i.e., flowing faster into the electrode. Negative mass flux values indicate flow out of the electrode. The high flux areas in the inlet/outlet planes, such as around (20, 70) in Figures 5.5 (a) and 5.7 (a), correspond to high flux areas seen in Kasula et al. [2] (Figures 5.5 (b) and 5.7 (b), respectively). Note that for the results for H_2O in this thesis work the scale is mostly negative, therefore the higher mass flux areas are more green, whereas the results for H_2 in this thesis work are mostly positive so higher mass flux areas are more red.

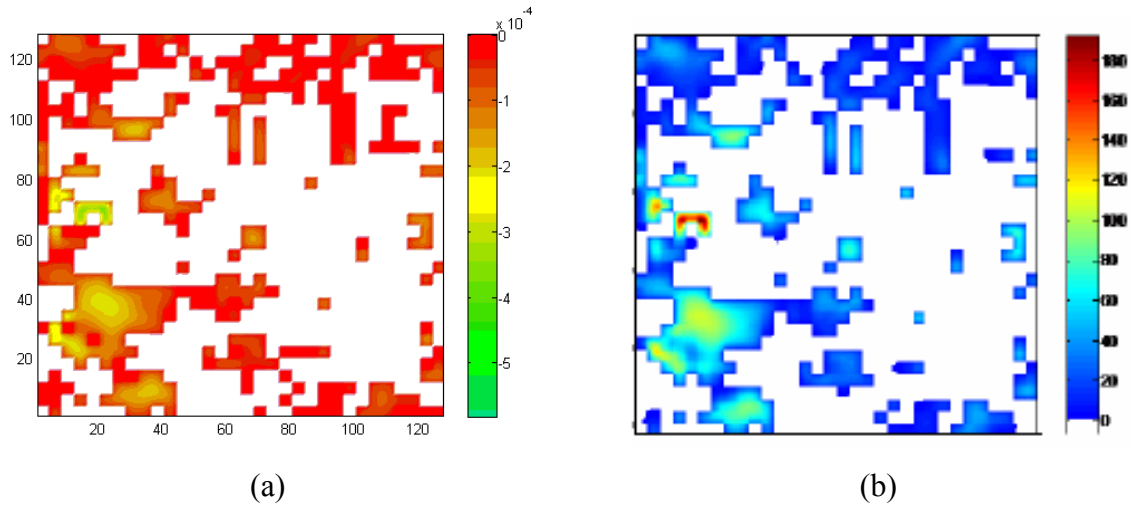


Figure 5.5. Comparison of H_2O mass flux contours at the inlet/outlet plane using Geometry B for the H_2O and H_2 mixture for the results from (a) this thesis work and (b) Kasula et al. [2]⁴.

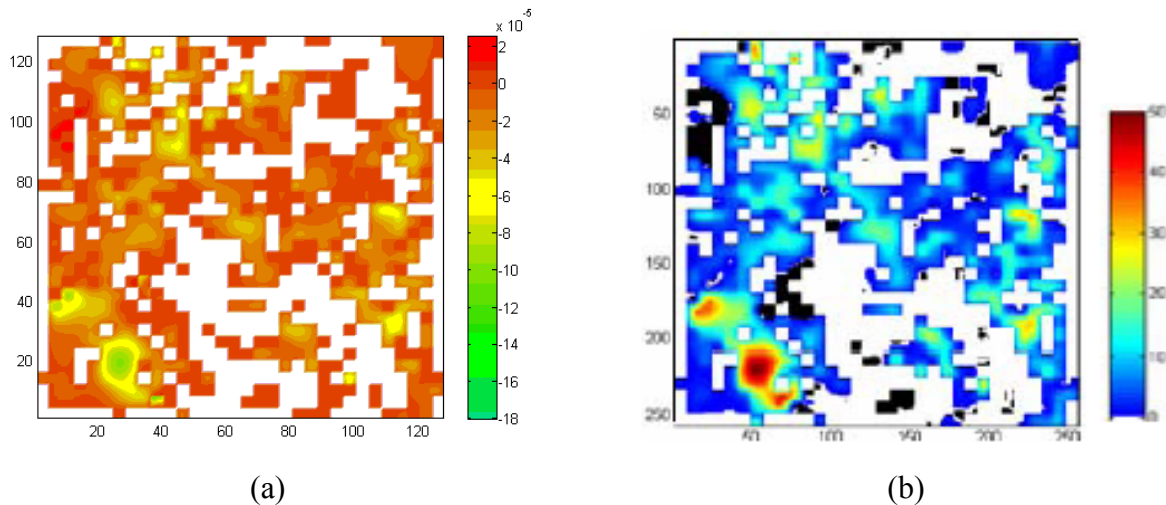


Figure 5.6. Comparison of H_2O mass flux contours at an intermediate plane using Geometry B for the H_2O and H_2 mixture for the results from (a) this thesis work and (b) Kasula et al. [2]⁴.

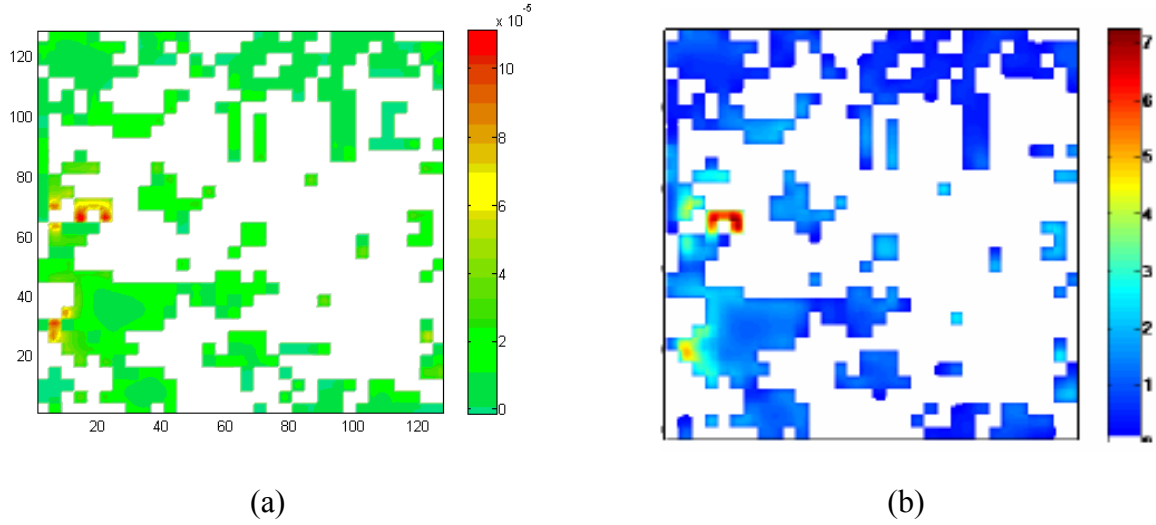


Figure 5.7. Comparison of H₂ mass flux contours at the inlet/outlet plane using Geometry B for the H₂O and H₂ mixture for the results from (a) this thesis work and (b) Kasula et al. [2]⁴.

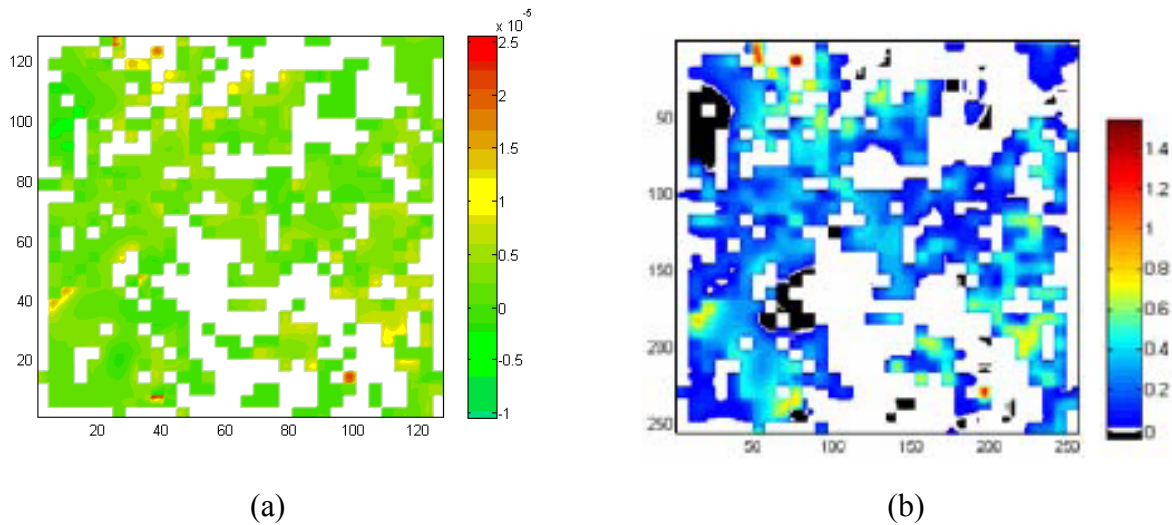


Figure 5.8. Comparison of H₂ mass flux contours at an intermediate plane using Geometry B for the H₂O and H₂ mixture for the results from (a) this thesis work and (b) Kasula et al. [2]⁴.

5.1.2.2 Species Mixture for Geometries C, D and E

The next geometry examined is Geometry C (Table 5.1). Because the porosity is so small, the pores close to the electrolyte do not have connectivity to the inlet flow; and, therefore, there are no TPBs in that region. The results in that region are not, thus, shown since they are not relevant. Figures 5.9 and 5.10 show the results of the water and hydrogen mixture simulation using Geometry C.

Figure 5.9 (a) shows the tortuosity of this simulation. The first 10 planes of the tortuosity in Figure 5.9 (a) are similar to that in Figure 5.1 (a) (the tortuosity results for the LB simulation

for the water and hydrogen mixture using Geometry B). The deviation in the following planes is due to the difference in the density and mass flux trends between Geometry C and Geometry B results as the tortuosity is based on both properties (see equations (5.1) through (5.5)). The mass flux (Figure 5.9 (b)) tends very quickly toward zero causing the tortuosity calculations, after about 12 planes into the electrode to become less useful as discussed for the tortuosity for Geometry B in Section 5.1.1. Because the tortuosity curve is not useful after about 12 planes, Figure 5.9 (a) shows only the first 12 planes.

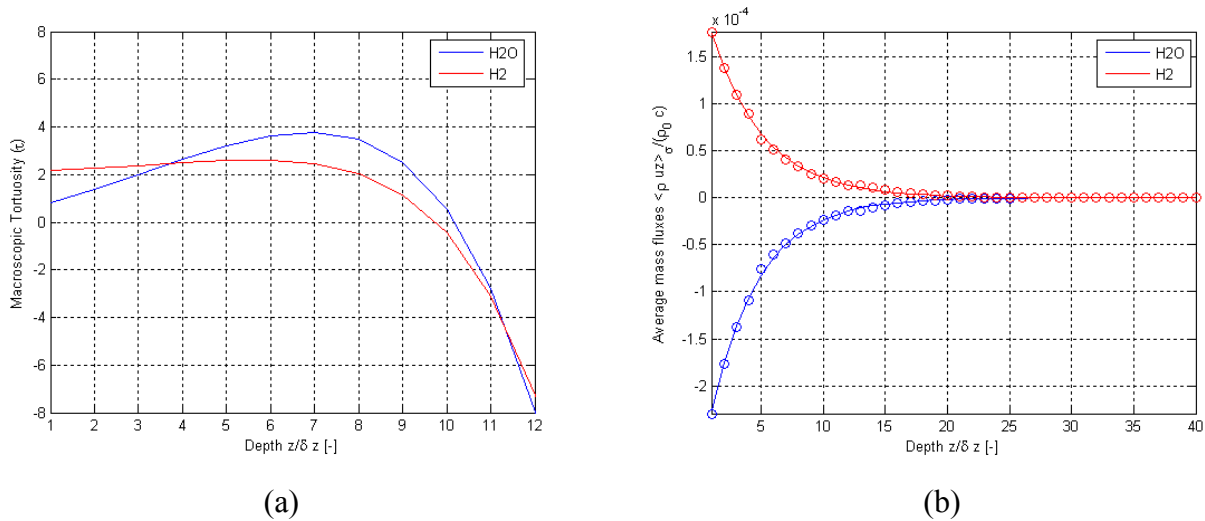


Figure 5.9. Results of this thesis work for variations as a function of depth using Geometry C for the H_2O and H_2 mixture for (a) τ and (b) normalized surface-averaged mass flux.

The surface-averaged mass fluxes (Figure 5.9 (b)) drop more quickly toward zero than what is seen in Figure 5.4 (a) because the physical size of the geometries reconstructed in this thesis work (of which Geometry C is one) is much smaller than that of the geometries developed in Asinari et al. [1] and Kasula et al. [2], i.e. one cell in the geometries created for this work has a physical size of $0.0367 \mu\text{m}$ whereas in the geometries created in Asinari et al. [1] and Kasula et al. [2] (Geometries A and B) one cell has a physical size $0.25 \mu\text{m}$. This affects the results because the pores in Geometries C, D, E, F and G are smaller than those for Geometries A and B even for geometries with the same porosity (i.e., Geometries A, B and E), thereby, causing the fluid particles to slow more as they travel toward the electrode.

Figures 5.10 (a) and 5.10 (b) show the density profiles for water and hydrogen, respectively. They have a stair-step pattern because the porosity is so low that the surface-averaged density cannot change much between the planes in the results that represent the original

geometry plane since for the refinement of 4 used in this thesis work, each cell in the geometry is represented by 4^3 cell in the results. The density profiles for the current thesis results (Figures 5.10 (a) and 5.10 (b)) show the same trend as the respective profiles in the results from this thesis work using Geometry B for the first 10 planes (Figures 5.2 (a) and 5.2 (b)). The succeeding planes contain fewer TPBs so less hydrogen is consumed and less water produced and, thus, the densities tend toward the IOBCs.

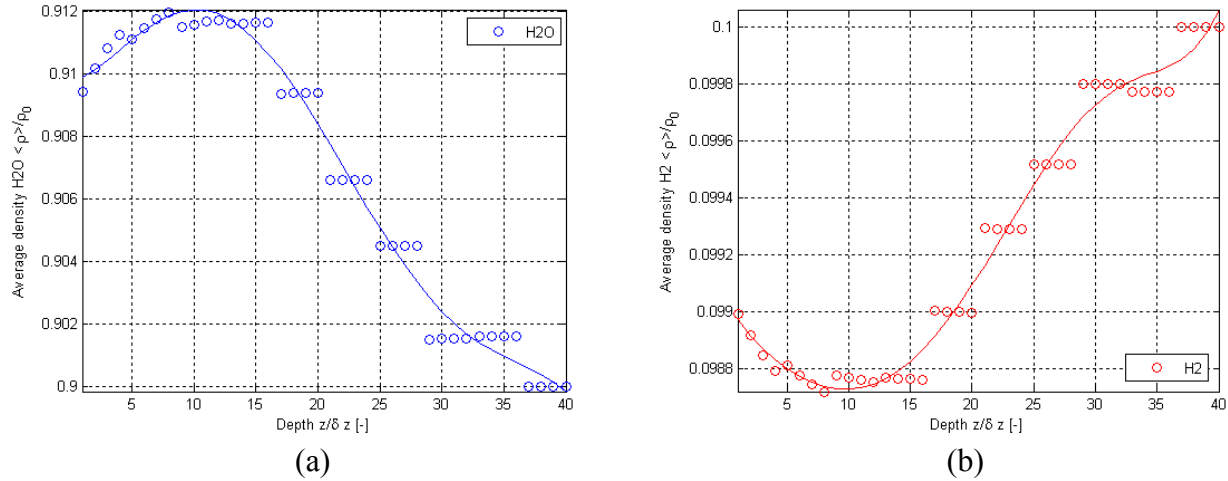


Figure 5.10. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry C for the H_2O and H_2 mixture for (a) H_2O and (b) H_2 .

Simulations for water and hydrogen were also run for Geometries D and E (Table 5.1). The graphs for these cases appear in Appendix E (Figures E.21 through E.28). As in the results from this thesis work for Geometry B, the first and last four planes are removed. Tables 5.4 and 5.5 show the volume-averaged results from this thesis work for Geometries D and E.

Table 5.4. Volume-averaged property summary for the H_2O and H_2 mixture using Geometry D.

Property	H_2O	H_2
$\langle\langle \rho_\sigma u_\sigma \rangle\rangle / \rho_0 c_0$	-3.358e-6	2.073e-6
$\langle\langle \rho_\sigma \rangle\rangle / \rho_0$	0.9094	0.09924
τ	5.610 +/- 2.76	4.387 +/- 2.36

Table 5.5. Volume-averaged property summary for the H_2O and H_2 mixture using Geometry E.

Property	H_2O	H_2
$\langle\langle \rho_\sigma u_\sigma \rangle\rangle / \rho_0 c_0$	-2.698e-6	2.071e-6
$\langle\langle \rho_\sigma \rangle\rangle / \rho_0$	0.9053	0.09955
τ	4.721 +/- 7.36	5.475 +/- 4.63

The trends in the density curves from this thesis work for the water and hydrogen mixture in Geometries D and E (Figures E.22, E.23, E.26 and E.27) are similar to those of the results from this thesis work for Geometry B (Figures 5.2 (a) and 5.2 (b)). As with the surface-averaged mass fluxes found in this thesis work Geometry C (Figure 5.9 (a)), the surface-averaged mass fluxes (Figures E.24 and E.28) drop off quickly because of the small cell size. This causes the volume-averaged water mass flux in both Tables 5.4 and 5.5 to be lower than the volume-averaged water mass flux seen in the results from this thesis for Geometry A (Table 5.2). The surface-averaged H₂ mass flux results from this thesis work in Geometries D and E show similar trends in the surface-averaged mass flux curves for H₂ in Geometry A. The volume-averaged densities for Geometries D and E (Tables 5.4 and 5.5) are nearly identical to those for Geometry A (Table 5.2). The high range in the tortuosity for H₂ in Geometry E are due to the many spikes in the tortuosity graph (Figure E.25) which are due to the fact that the mass-flux curves (Figure E.28) tend quickly toward zero.

5.1.3 2 Species Mixture for Geometries F and G

Figures 5.11 and 5.12 show the results for Geometry F (Table 5.1). The 64 planes closest to the electrolyte are not shown as the pores do not have connectivity to the inlet flow plane and the data is irrelevant.

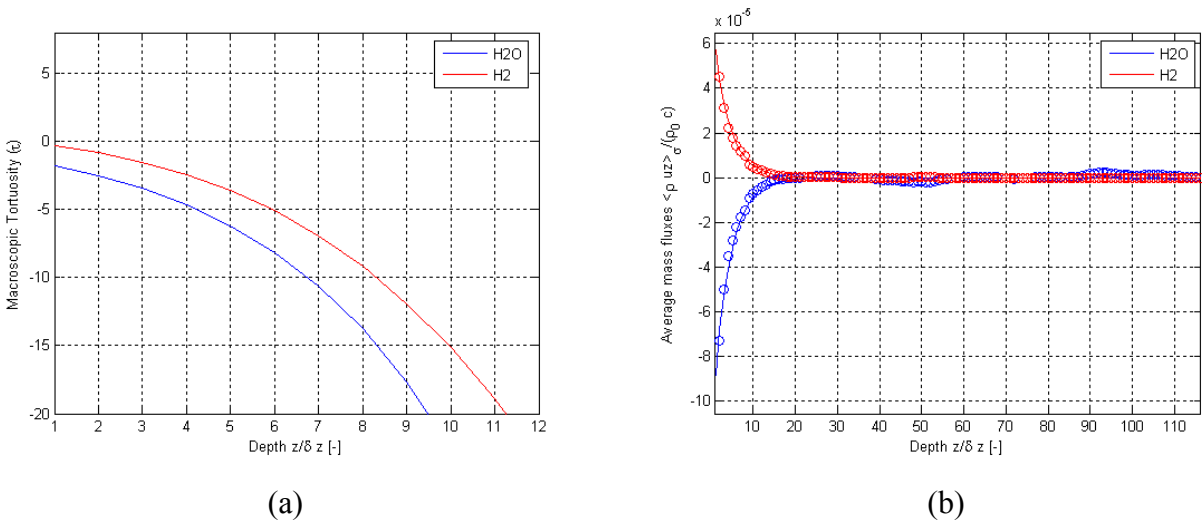


Figure 5.11. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O and H₂ mixture for (a) τ and (b) normalized surface-averaged mass flux.

The tortuosities in Figure 5.11 (a) are the inverse of what is seen in the in the results for this thesis work for Geometry C because the density curves of the first 12 planes have the

opposite slopes i.e., taking the H₂O density curves as an example, Figure 5.12 (a) shows the downward slope of the density curve, whereas for the H₂O density curve for Geometry C (Figure 5.10 (a)), the initial slope is positive. Again, only the first 12 planes of the tortuosity are shown because the mass fluxes (Figure 5.11 (b)) drop to zero after about 12 planes into the geometry. The mass flux and density trends (Figures 5.11 (b) and 5.12) are similar to those for Geometry C (Figures 5.9 (b) through 5.10). As with the Geometry C, the low porosity does not allow for significant density changes between the planes that represent the same physical geometry causing the flat spots in the density curves (Figures 5.12 (a) and 5.12 (b)).

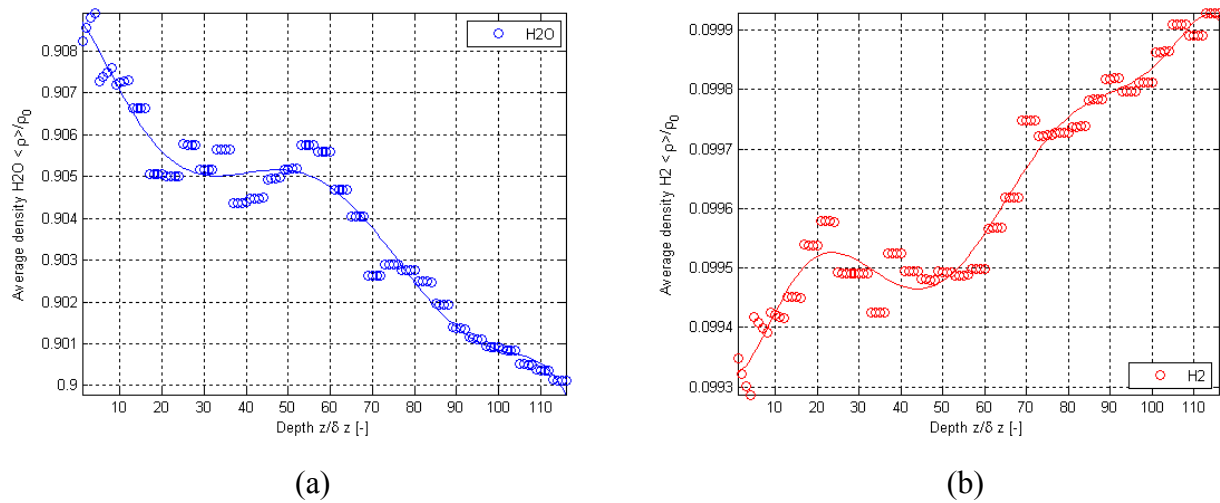


Figure 5.12. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O and H₂ mixture for (a) H₂O and (b) H₂.

The volume-averaged tortuosity, mass flux and density results for Geometry G are shown in Table 5.6. See Appendix E for the corresponding graphs (Figures E.33 through E.36). The trends for the mass flux and densities (Figures E.33 through E.35) are similar to those for Geometries D and E (Figures E.22 through E.24 and E.26 through E.28). The geometry has TPBs throughout the geometry and, as such, requires only the last four planes removed from the results. The values for all properties and species reported in Table 5.6 are within an order of magnitude of those reported in Tables 5.4 and 5.5, and are especially close to those in Table 5.5, the results for this thesis work using Geometry D. Geometry G yields results closer to those of Geometry E because, due to the reproduction technique, the average pore sizes in Geometry G are closer to those of Geometry E than Geometry D.

Note that the average reproduced pore sizes of Geometry D are larger than those of geometry E because the shade threshold used to separate the phases in the geometry is shifted to incorporate more “dark” pixels as pores. The average reproduced pore size in geometries F and G are larger than the average reconstructed pore sizes of geometries C and D, respectively, even though they have the same porosities. This is due to the fact that geometries F and G represent a larger portion of the actual geometry than geometries C and D.

Table 5.6. Volume-averaged property summary for the H₂O and H₂ mixture using Geometry G.

Property	H ₂ O	H ₂
$\langle\langle\rho_{\sigma}u_{\sigma}\rangle\rangle/\rho_0c_0$	-3.66597e-6	2.07302e-6
$\langle\langle\rho_{\sigma}\rangle\rangle/\rho_0$	0.907825	0.099376
τ	5.135 +/- 9.12	5.712 +/- 3.12

5.2 Without Chromium Poisoning

In order to understand the effects that chromium poisoning has on the cathode of an SOFC, a reference or base must be examined and verified first. Air containing nitrogen, oxygen and water enters the cathode where oxygen undergoes an electrochemical reaction and the nitrogen and water are inert. See Chapter 4 for the details of the model parameters.

5.2.1 3 Species Mixture for Geometries A and B

The three species simulation is run using Geometries A and B in order to verify the validity of the code modifications required to incorporate the third species. The results of the simulation using Geometry A are tabulated in Table 5.7 and the corresponding graphs are shown in Appendix E (Figures E.37 through E.41). The density profile for the oxygen (Figure E.39) is nearly identical to that for hydrogen in the case of the anode (the hydrogen and water mixture simulation in Section 5.1.1) for the same geometry (Figure E.3). This is a very important similarity as both oxygen in the cathode and hydrogen in the anode are the species consumed in the electrochemical reactions. The other species in the cathode do not show similar trends to those of the water in the anode as these other species are inert as opposed to the water, which is formed in the electrochemical reactions. The tortuosities for the water and oxygen in Table 5.7 are, however, of the same order of magnitude as those in Table 5.2, the anode results using the same geometry. Nitrogen has a lower tortuosity because of significant negative spikes in the tortuosity curve (Figure E.37) and because it has a higher mass flux than the other species

causing the difference in the species and mixture mass fluxes in equation 5.2 to be negative. The volume-averaged densities are near the IOBC and initial condition values which signifies there are no significant anomalies in the surface-averaged density curves.

Table 5.7. Volume-averaged property summary for the results from this thesis work using Geometry A and the H₂O, O₂ and N₂ mixture.

Property	H ₂ O	O ₂	N ₂
$\langle\langle \rho_\sigma u_\sigma \rangle\rangle / \rho_0 c_0$	1.306e-5	2.683e-4	8.599e-4
$\langle\langle \rho_\sigma \rangle\rangle / \rho_0$	0.01138	0.2084	0.7607
τ	1.950 +/- 2.41	1.014 +/- 8.5	0.2259 +/- 9.74

Figures 5.13 through 5.15 show the results for the simulation with the water, oxygen and nitrogen mixture using Geometry B. The tortuosities (Figure 5.13 (a)) are on the same order of magnitude as those for the water and hydrogen case with the exception of nitrogen. The spikes, as discussed in Section 5.1.1 are due to the near-zero mass flux near the right-hand side of Figure 5.13 (b). The mass fluxes (Figure 5.13 (b)) are all positive, showing flow only into the electrode because there are no concentration gradients driving the fluid outwards.

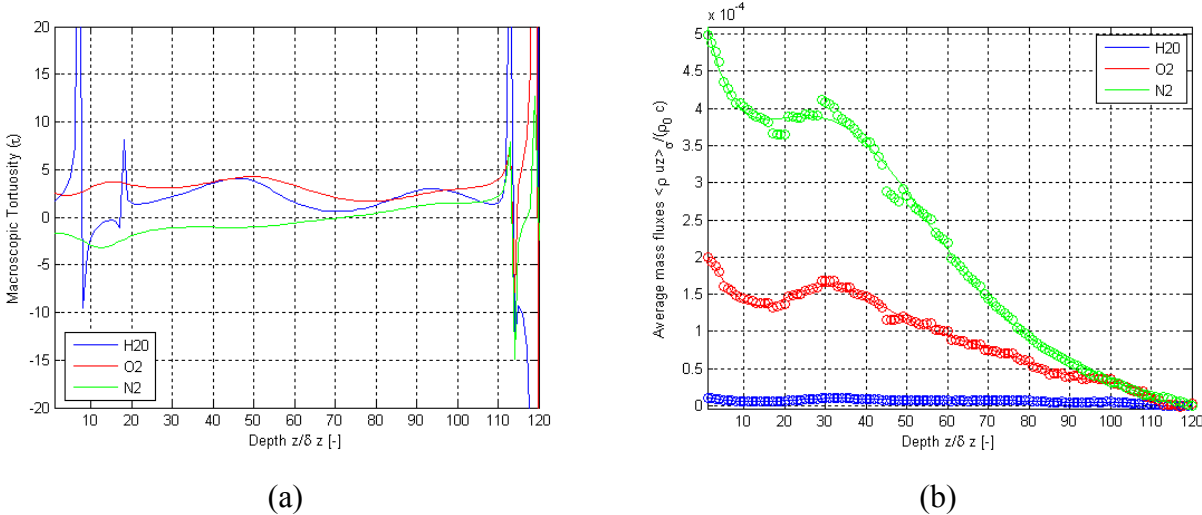


Figure 5.13. Results of this thesis work for variations as a function of depth using Geometry B for the H₂O, O₂ and N₂ mixture for (a) τ and (b) normalized surface-averaged mass flux.

As with Geometry A, the reacting species, oxygen (Figure 5.17), shows very similar trends in the density profile to that of hydrogen (Figure 5.2 (b)) reacting in the same geometry. The water density profile (Figure 5.14 (a)) follows the oxygen profile (Figure 5.14 (b)) (on a lower scale as the percentage of water in the system is less than that of oxygen) because it is

heavily influenced by the concentration gradient due to the electrochemical reaction because its density is so low. The density of the water varies on the $O(10^{-4})$ where as the oxygen varies on the $O(10^{-2})$ and the nitrogen varies on the $O(10^{-3})$. Oxygen varies the most as it is the reacting species. Water varies the least as there is the least amount of it in the mixture. The changes in the densities of each species also have different magnitudes for the remaining geometries and the four species mixture cases.

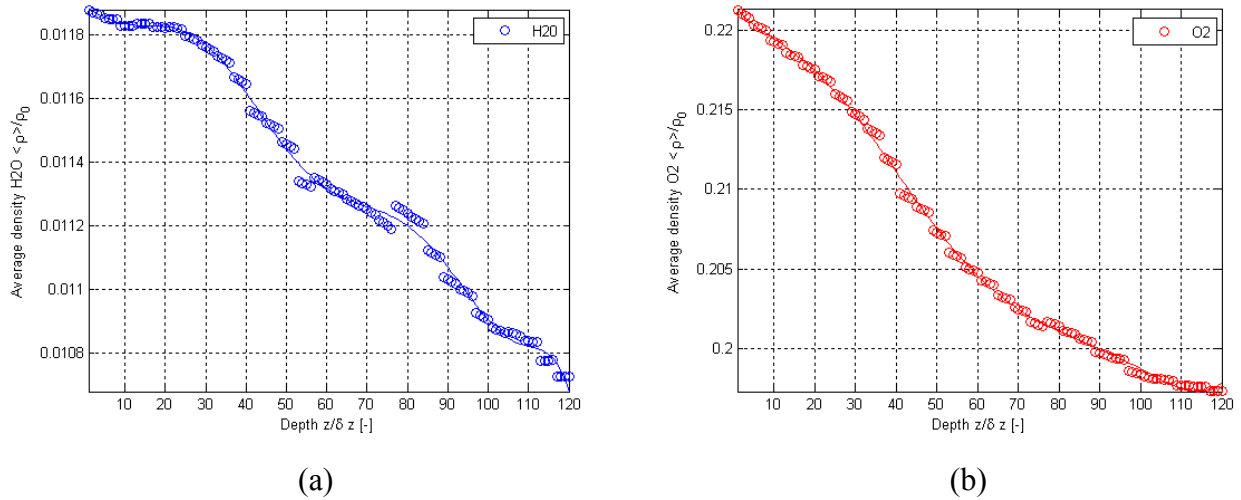


Figure 5.14. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂ and N₂ mixture for (a) H₂O and (b) O₂.

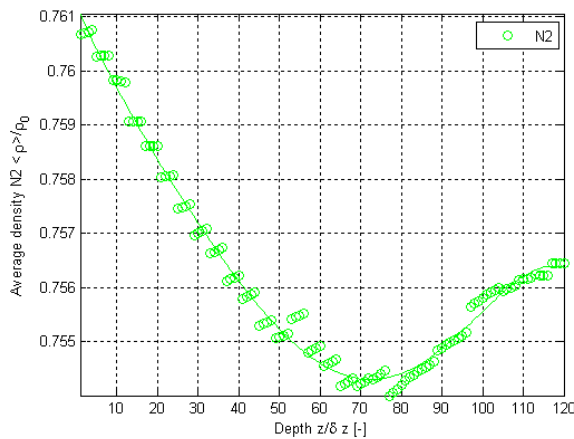


Figure 5.15. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂ and N₂ mixture for N₂.

Figures 5.16 through 5.18 show the mass flux contours for Geometry B. They show similar high flux regions to Figures 5.5 through 5.8 (the mass flux contours for the H₂O and H₂ mixture in Geometry B) for both the inlet/outlet planes and the intermediate planes.

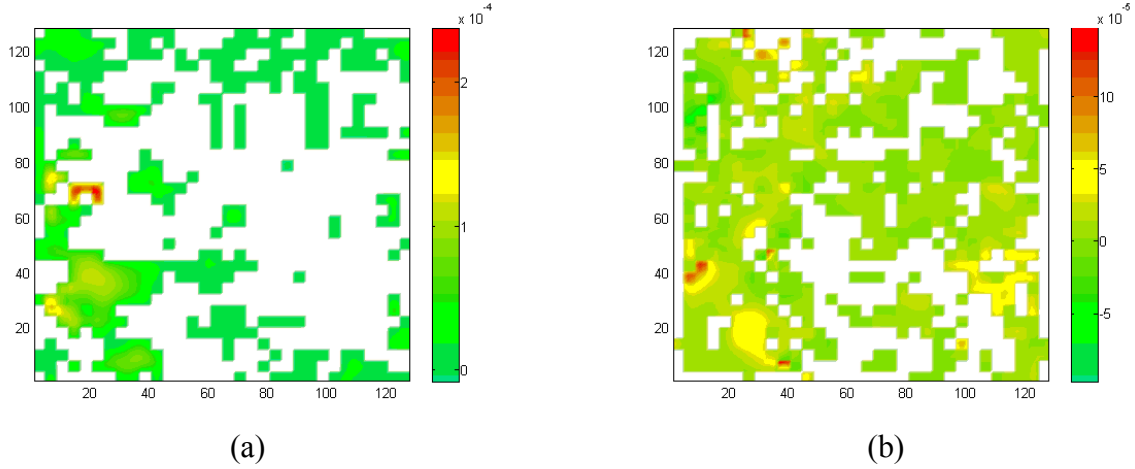


Figure 5.16. Normalized mass flux contours for H₂O using Geometry B for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

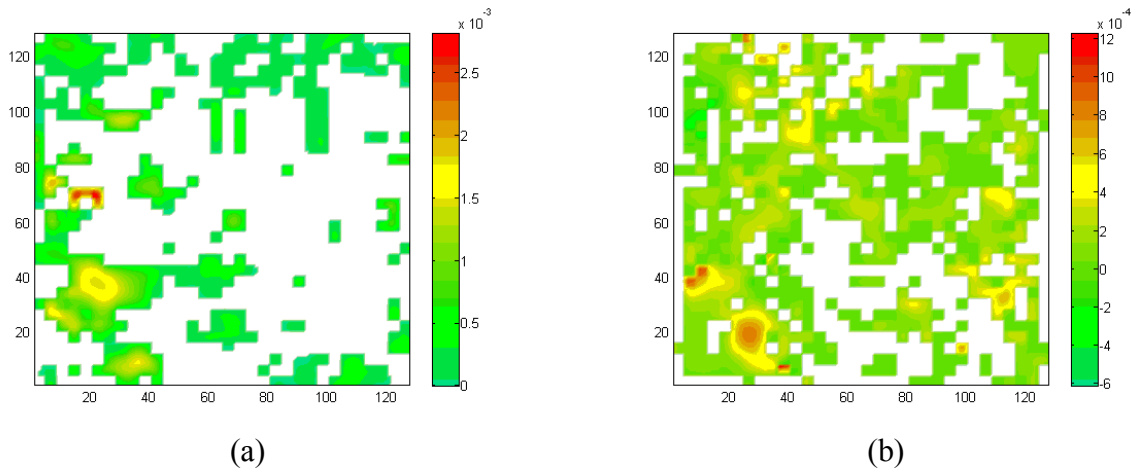


Figure 5.17. Normalized mass flux contours for O₂ using Geometry B for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

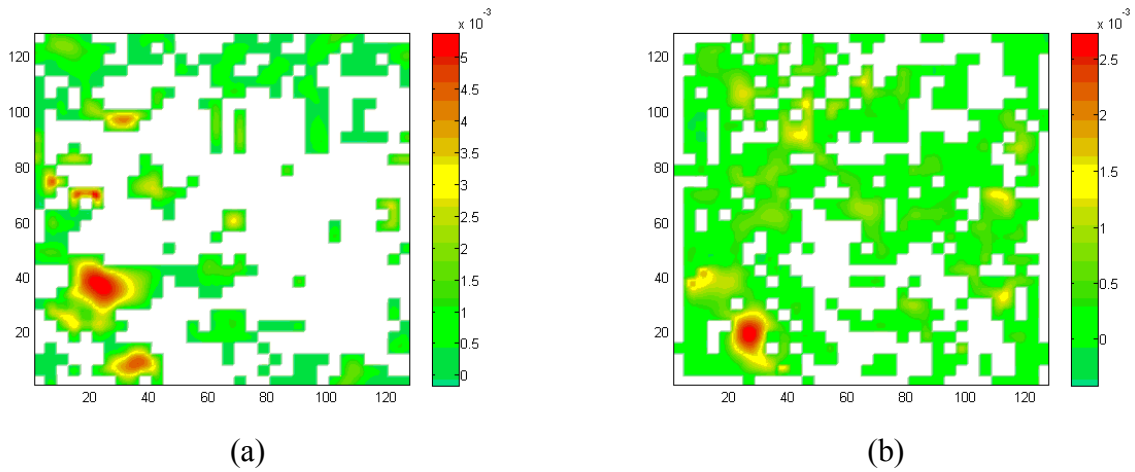


Figure 5.18. Normalized mass flux contours for N₂ using Geometry B for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

5.2.2 3 Species Mixture for Geometries C, D and E

The results for the simulation using a water, oxygen and nitrogen mixture and Geometry C are shown in Figures 5.19 through 5.24. The tortuosities (Figures 5.19 (a) and (b)) show behaviors consistent with the tortuosities seen using Geometry B (Figure 5.13 (a)): water and oxygen tortuosities around a value of 2 and a partially negative tortuosity with sharp negative spikes for the nitrogen. The surface-averaged mass flux profiles (Figure 5.19 (c)) show a rapid slowing of the particles as they enter the system due to the small pore sizes as discussed in section 5.1.2. All the profiles are positive as there are no constituents being formed in the system.

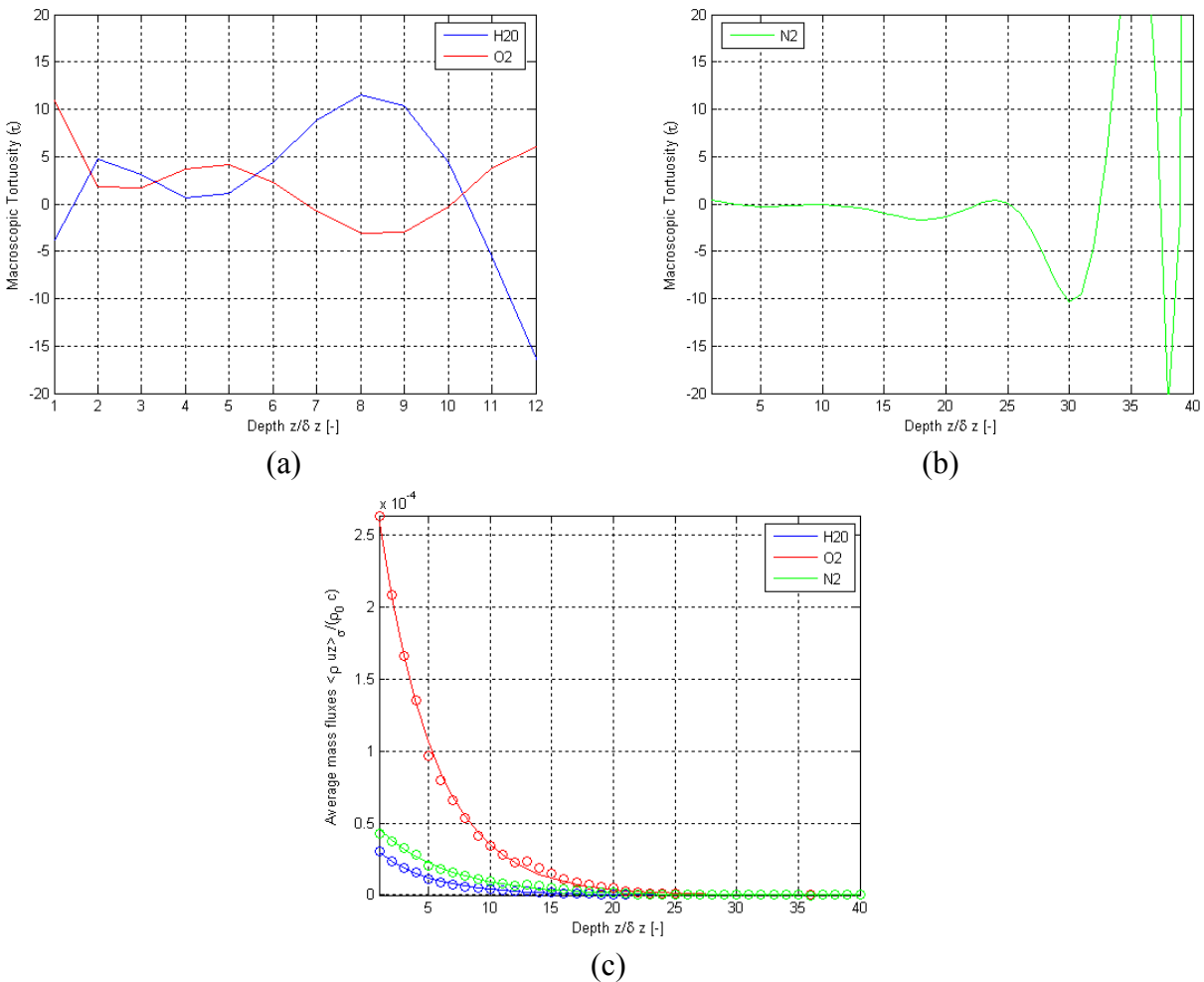


Figure 5.19. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for (a) τ for H₂O and O₂, (b) τ for N₂ and (c) normalized surface-averaged mass flux.

Again, the similarity in the oxygen surface-averaged density profile (Figure 5.20 (b)) and the hydrogen surface-averaged density profile (Figure 5.10 (b)) for the same geometry must be noted. The surface-averaged density for water follows that for oxygen as in the results for Geometry B in the three species mixture case. The surface-averaged density curve for nitrogen has similar trends to that of nitrogen in Geometry B.

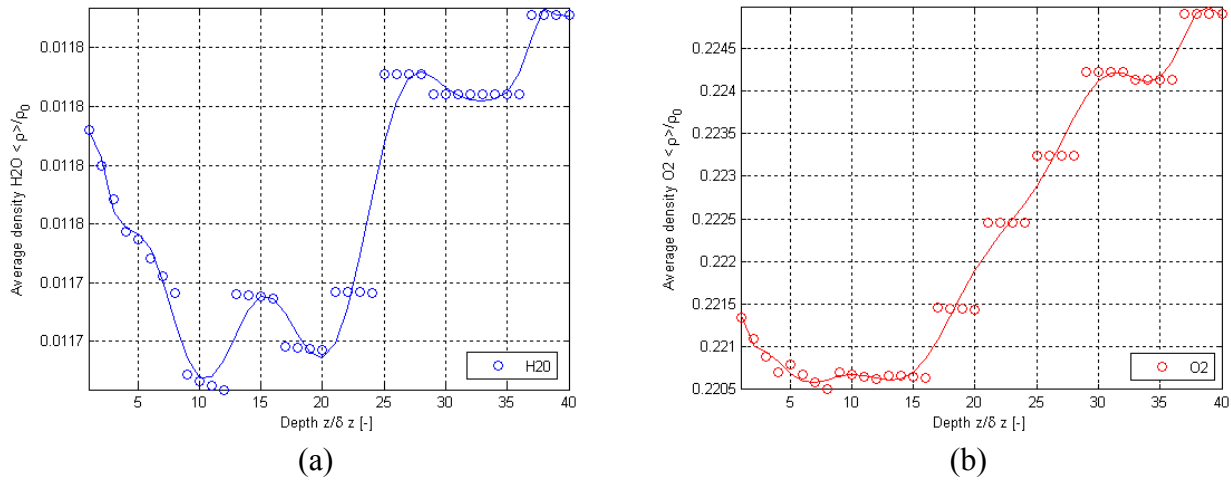


Figure 5.20. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for (a) H₂O and (b) O₂.

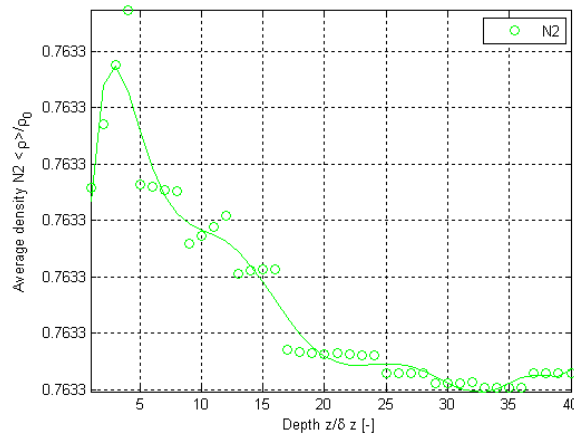


Figure 5.21. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for N₂.

Figures 5.22 through 5.24 show the normalized mass flux contour plots for the constituents for the inlet/outlet plane and an intermediate plane. As in Figures 5.5 through 5.8 the white indicates solid regions of the electrode. It is easy to see, especially at the intermediate plane, that the geometry is very dense allowing very few TPBs. Figures 5.22 through 5.24 all show positive mass flux with consistent high flux areas for all the species, which is expected,

since the oxygen concentrations gradient (due to the electrochemical reactions) cause concentration gradients in the water and nitrogen.

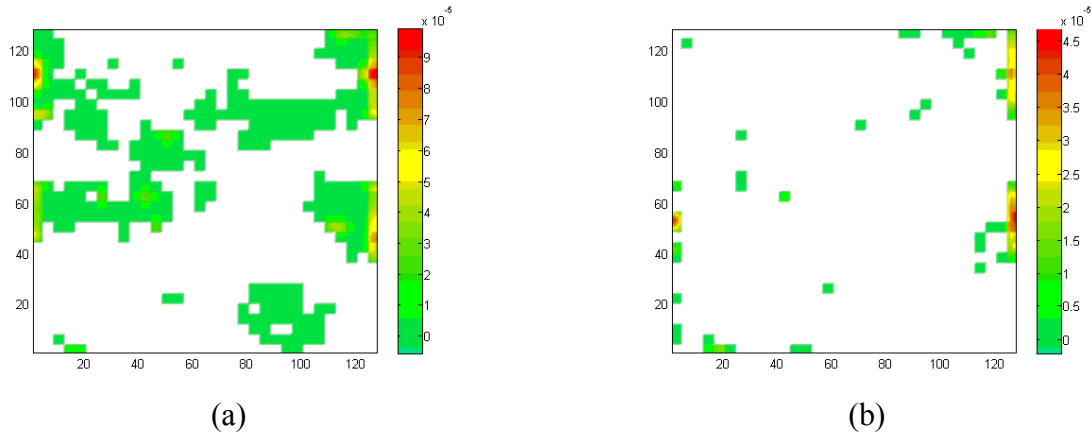


Figure 5.22. Normalized mass flux contours for H₂O using Geometry C for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

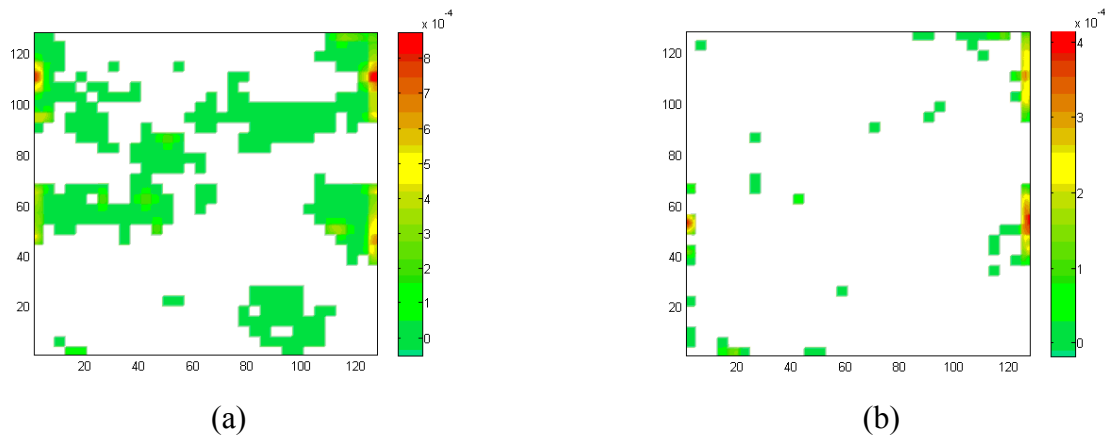


Figure 5.23. Normalized mass flux contours for O₂ using Geometry C for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

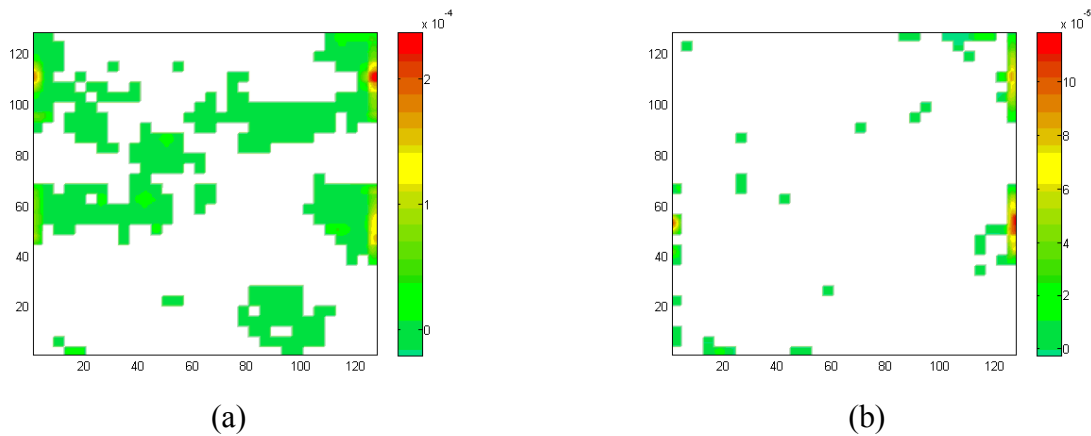


Figure 5.24. Normalized mass flux contours for N₂ using Geometry C for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

Tables 5.8 and 5.9 show the volume-averaged properties for Geometries D and E, respectively. The corresponding graphs are shown in Appendix E (Figures E.64 through E.73). The surface-averaged mass flux trends (Figures E.68 and E.73) are similar to those for the 3 species mixture in Geometry C (Figure 5.13 (b)) and the oxygen surface-averaged density trends (Figures E.66 and E.71) are similar to those of hydrogen for Geometries D and E, respectively (Figures E.23 and E.27) as before. The volume-averaged mass flux values in Tables 5.8 and 5.9 are an order of magnitude lower than those in Table 5.7 (the properties for the water, oxygen and nitrogen mixture in Geometry A) because the size of each cell in the geometry is an order of magnitude smaller as discussed in section 5.1.2. The volume-averaged density values in Tables 5.8 and 5.9 are similar to those in Table 5.7. The tortuosities in Tables 5.8 and 5.9 are similar to those seen in the cases of the water and hydrogen mixture for the same geometries (Tables 5.4 and 5.5).

Table 5.8. Volume-averaged property summary for the results from this thesis work using Geometry D and the H₂O, O₂ and N₂ mixture.

Property	H ₂ O	O ₂	N ₂
$\langle\langle\rho_\sigma u_\sigma\rangle\rangle/\rho_0 c_0$	4.1566e-6	3.606e-5	6.750e-6
$\langle\langle\rho_\sigma\rangle\rangle/\rho_0$	0.01089	0.1953	0.7633
τ	5.488 +/- 6.35	5.037 +/- 2.25	-3.317 +/- 4.10

Table 5.9. Volume-averaged property summary for the results from this thesis work using Geometry E and the H₂O, O₂ and N₂ mixture.

Property	H ₂ O	O ₂	N ₂
$\langle\langle\rho_\sigma u_\sigma\rangle\rangle/\rho_0 c_0$	4.168e-6	3.691e-5	9.957e-6
$\langle\langle\rho_\sigma\rangle\rangle/\rho_0$	0.01099	0.2071	0.76323
τ	10.08 +/- 3.53	7.625 +/- 4.25	-2.693 +/- 3.87

5.2.3 3 Species Mixture for Geometries F and G

The results for the 3 species mixture simulation using Geometry F are shown in Figures 5.25 through 5.27. The results show behavior similar to the 3 species simulations with Geometry C, i.e., the tortuosity (Figures 5.25 (b)) for nitrogen is generally more negative and the surface-averaged mass flux (Figure 5.25 (c)) and surface-averaged density curves (Figures 5.26 and 5.27) have similar trends to those in Figure 5.19 (c) and Figures 5.20 and 5.21, respectively. In addition, note that the oxygen surface-averaged density curve (Figure 5.26 (b)) matches that of

the hydrogen in the two species simulation (Figure 5.12 (b)) and the surface-averaged density curves (Figures 5.26 and 5.27) have flat spots due to low porosity. The tortuosities for water and oxygen, however, (Figure 5.25 (a)) are different due to the difference in the initial slope of the density curves as discussed in the water and hydrogen mixture case in Section 5.1.3.

Figures 5.28 through 5.30 show the normalized mass flux contour plots for the constituents at the inlet/outlet plane and an intermediate plane. Note that the mass flux ranges in the contour plots for the intermediate planes (Figures 5.28 (b), 5.29 (b) and 5.30 (b)) are an order of magnitude lower those for the inlet/outlet planes (Figures 5.28 (a), 5.29 (a) and 5.30 (a)), which matches the results seen in Figure 5.25 (c). Again, in general, there appear to be no regions or areas of negative mass fluxes.

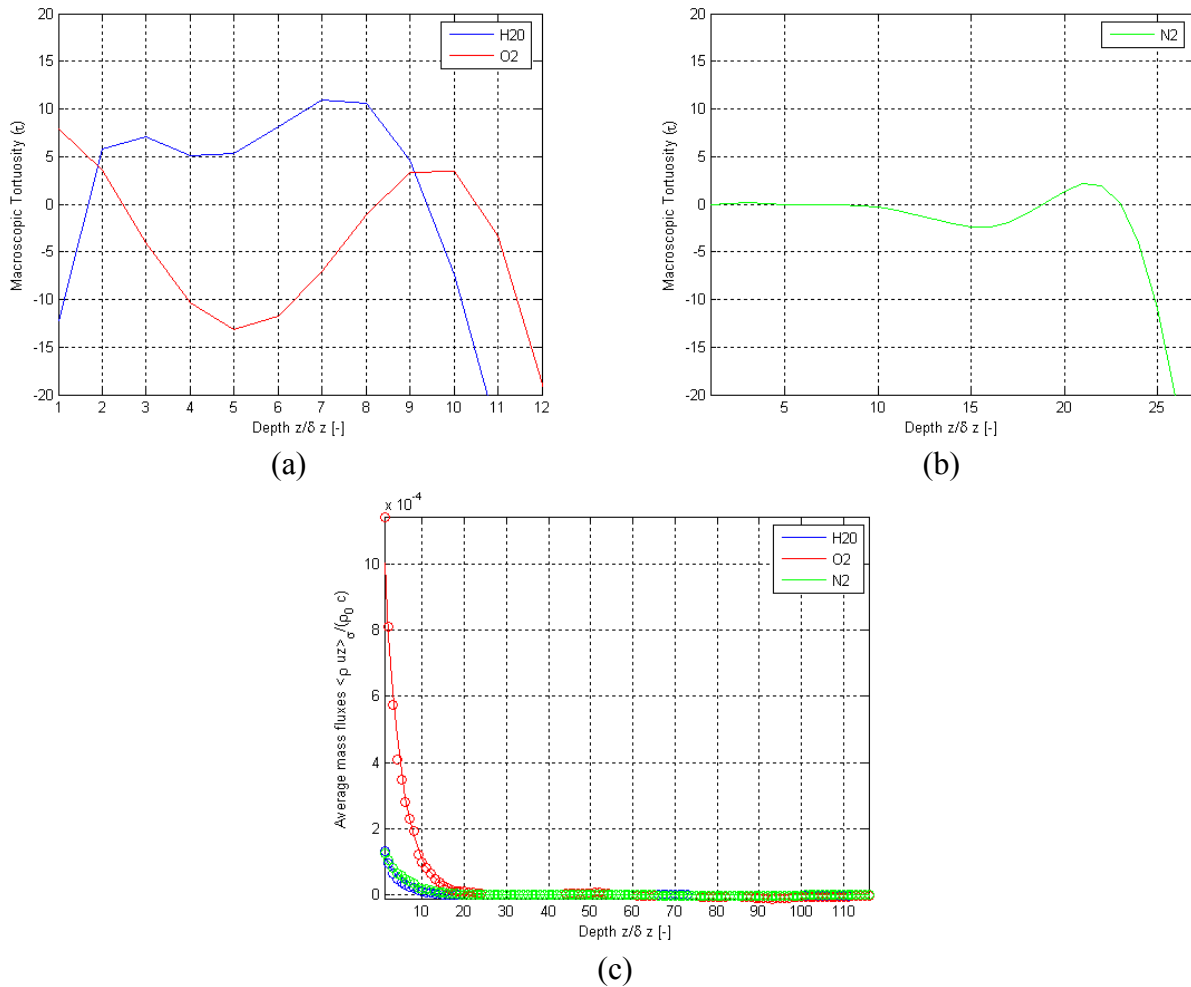
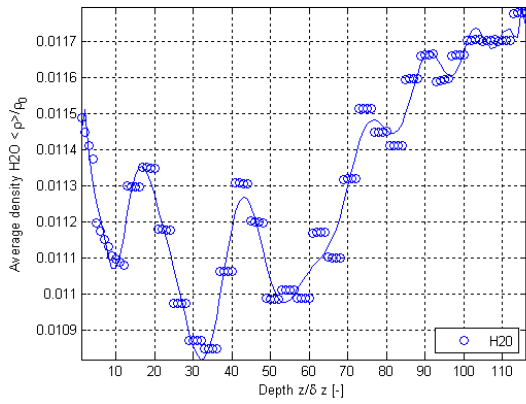
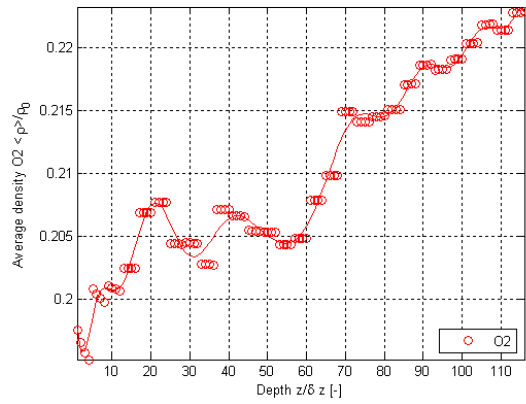


Figure 5.25. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for (a) τ for H₂O and O₂, (b) τ for N₂ and (c) normalized surface-averaged mass flux.



(a)



(b)

Figure 5.26. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for (a) H₂O and (b) O₂.

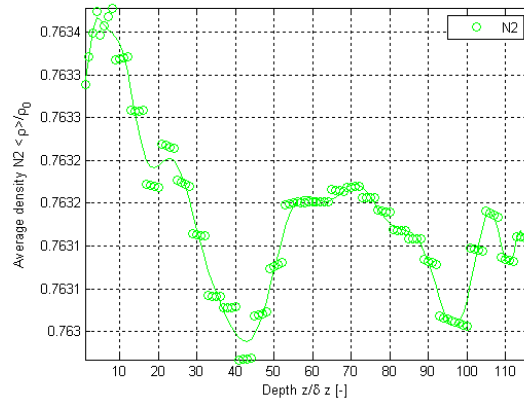
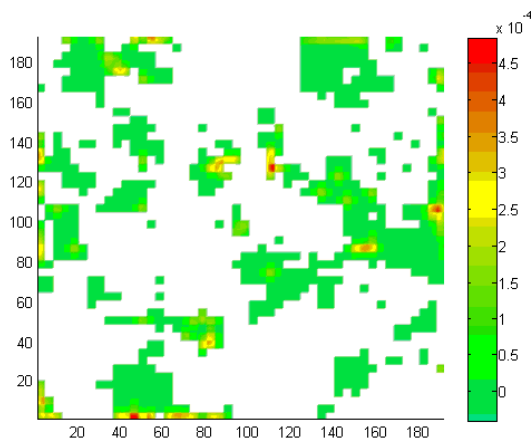
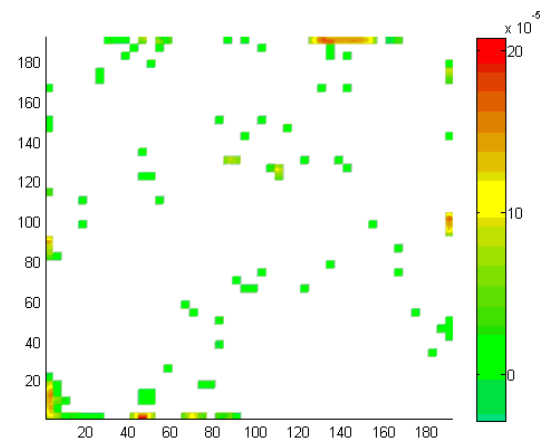


Figure 5.27. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for N₂.



(a)



(b)

Figure 5.28. Normalized mass flux contours for H₂O using Geometry F for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

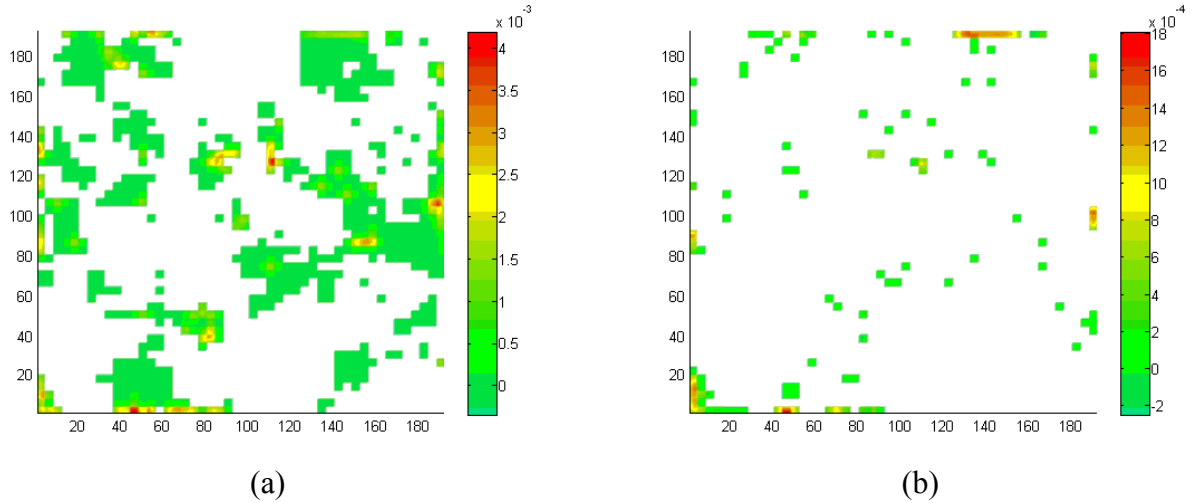


Figure 5.29. Normalized mass flux contours for O₂ using Geometry F for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

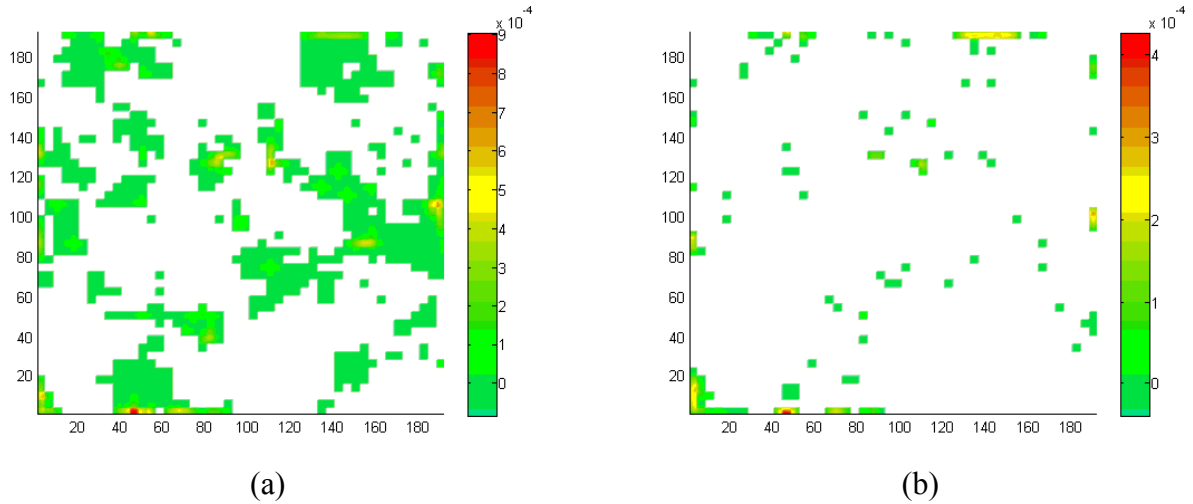


Figure 5.30. Normalized mass flux contours for N₂ using Geometry F for the H₂O, O₂ and N₂ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

Table 5.10. Volume-averaged property summary for the results from this thesis work using Geometry G and the H₂O, O₂ and N₂ mixture.

Property	H ₂ O	O ₂	N ₂
$\langle\langle \rho_{\sigma} u_{\sigma} \rangle\rangle / \rho_0 c_0$	2.819e-6	2.494e-5	6.914e-6
$\langle\langle \rho_{\sigma} \rangle\rangle / \rho_0$	0.01132	0.2083	0.7633
τ	-8.995 +/- 19.0	6.832 +/- 6.20	0.4726 +/- 6.51

Table 5.10 shows the volume-averaged results for Geometry G. The corresponding graphs can be seen in Appendix E (Figures E.85 through E.89). The results show trends closest to those of Geometry E (Figures E.69 through E.73) because they have similar average

reconstructed pore sizes as mentioned in Section 5.1.3. The summary in Table 5.10 shows that the results are similar to those for Geometry E (Table 5.9) as well. The tortuosities, however, are lower because the mass fluxes have a sharper drop which causes the tortuosities to have a negative spike nearer to the inlet/outlet plane.

5.3 With Chromium Poisoning

The LB code is first run with 4 species mixture of water, oxygen, nitrogen and chromium oxide vapor, functionality for blocking the TPBs disabled in order to establish reasonable results are obtained for the 4 constituents mixture in the simulation. Then, in order to establish when the electrode becomes completely disabled, the density profiles for the reacting species are examined at different time steps. Because the flow is concentration driven due to the TPB B.C.s (see Section 4.3.2 in Chapter 4), all of the TPBs are assumed to be blocked when the density profiles of the reacting species, i.e., oxygen and chromium oxide vapor, become constant throughout the geometry. Function of the SOFC is then assumed to be halted.

5.3.1 Mixture without Blocking of the TPBs for Geometries A and B

The results for the 4 species mixture in the cathode using Geometry A are shown in Appendix E (Figures E.90 through E.95) and summarized in Table 5.11. The results show similar trends to those seen in the three species case (Figures E.37 through E.41) with the exception of some anomalies due to the chromium oxide vapor added to the mixture. The volume-averaged mass fluxes in Table 5.11 are a little lower than those seen in Table 5.7 (the three species case for Geometry A) which is due to the chromium oxide vapor also moving through the electrode.

Table 5.11. Volume-averaged property summary for the results from this thesis work using Geometry A and the H₂O, O₂, N₂ and CrO₃ mixture.

Property	H ₂ O	O ₂	N ₂	CrO ₃
$\langle\langle\rho_\sigma u_\sigma\rangle\rangle/\rho_0 c_0$	1.452e-6	3.815e-5	1.291e-4	4.564e-9
$\langle\langle\rho_\sigma\rangle\rangle/\rho_0$	0.01170	0.2213	0.7630	1.470e-5
τ	1.476 +/- 5.90	2.636 +/- 21.8	-11.11 +/- 149.	6.928 +/- 22.5

Figures 5.31 through 5.33 show the results for the four species mixture in Geometry B. The tortuosity graph (Figure 5.31 (a)) shows more spikes than previous curves for tortuosity. This is because the fluid path or tortuosity is affected by the forces and molecular weights of the

other species in the mixture in the diffusivity equations [1, 77]. As discussed in Section 4.1.2 in Chapter 4, the diffusivity is calculated using the collision integral that contains characteristic lengths and energies used in modeling interactions between species. As mentioned previously, the spikes and negative tortuosities are due to the fact that the tortuosity is not purely a geometric parameter but is calculated using the fluid properties.

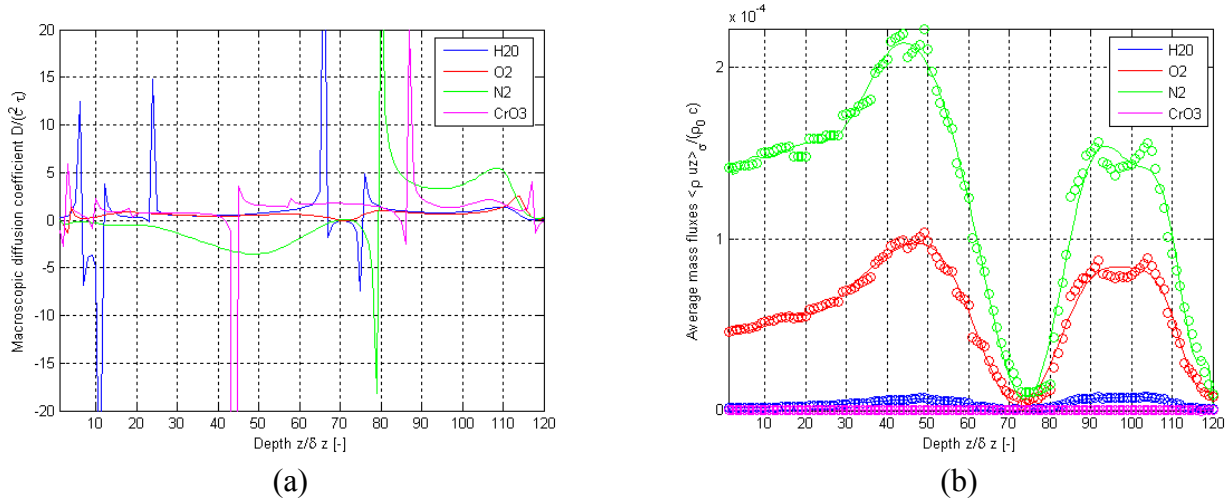


Figure 5.31. Results of this thesis work for variations as a function of depth using Geometry B for the H₂O, O₂, N₂ and CrO₃ mixture for (a) τ and (b) normalized surface-averaged mass flux.

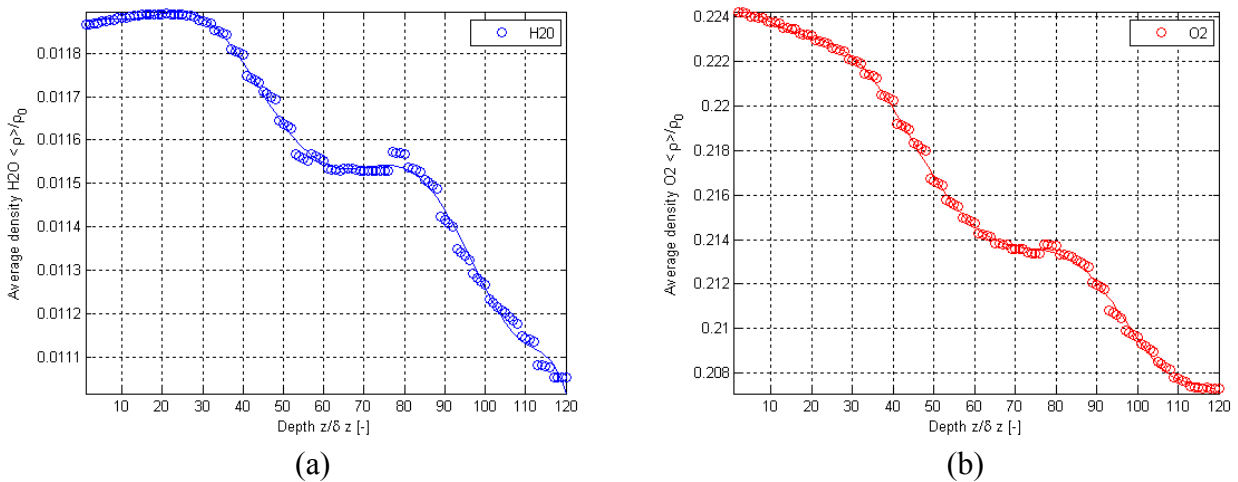


Figure 5.32. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂, N₂ and CrO₃ mixture for (a) H₂O and (b) O₂.

The density curves for water, oxygen and nitrogen (Figures 5.32 and 5.33) are similar to the three species density trends seen for the same geometry in Figures 5.14 and 5.15. The surface-averaged mass flux curves (Figure 5.31 (b)) show a significant dip between planes 70 and 80 that

corresponds to a dip in the chromium oxide vapor density (Figure 5.33 (b)), i.e., the chromium oxide vapor reactions become more significant causing a dip in its density and more flow of the other species to that area resulting in the mass flux average over those planes to be lower. Since the chromium oxide vapor is so large compared to the other species, its consumption in the electrochemical reactions causes a large concentration driven flow of the other species in order to “fill the space.” This can cause negative mass fluxes or back flows as there is not enough of the other constituents flowing into the system from the inlet/outlet plane to make up the difference. The negative mass fluxes cause a drop in the surfaced-averaged mass fluxes but may or may not be “negative” enough to cause the surfaced-averaged mass fluxes to be negative.

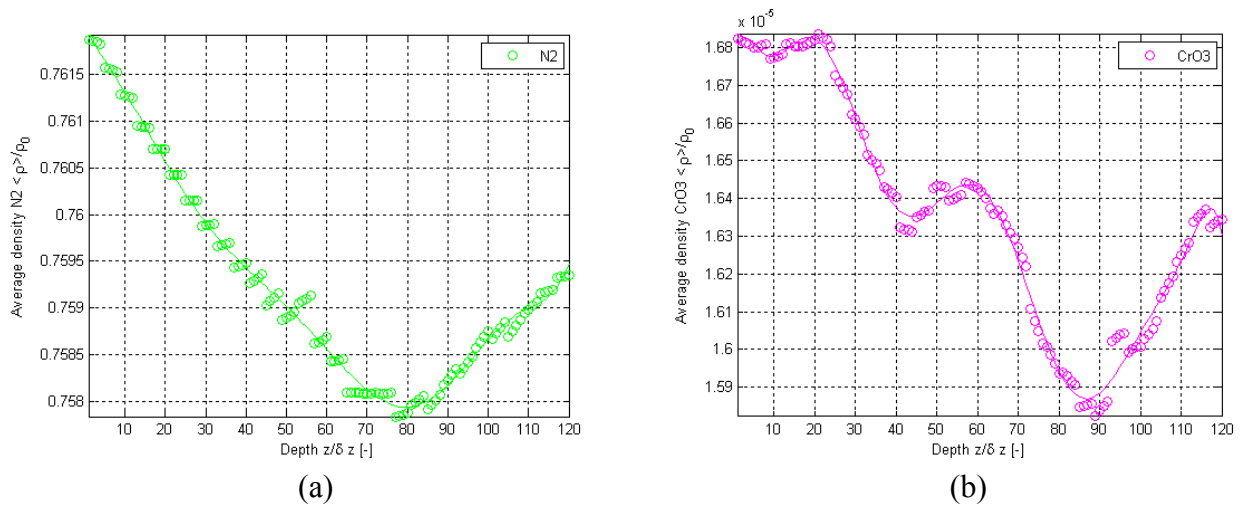


Figure 5.33. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) N_2 and (b) CrO_3 .

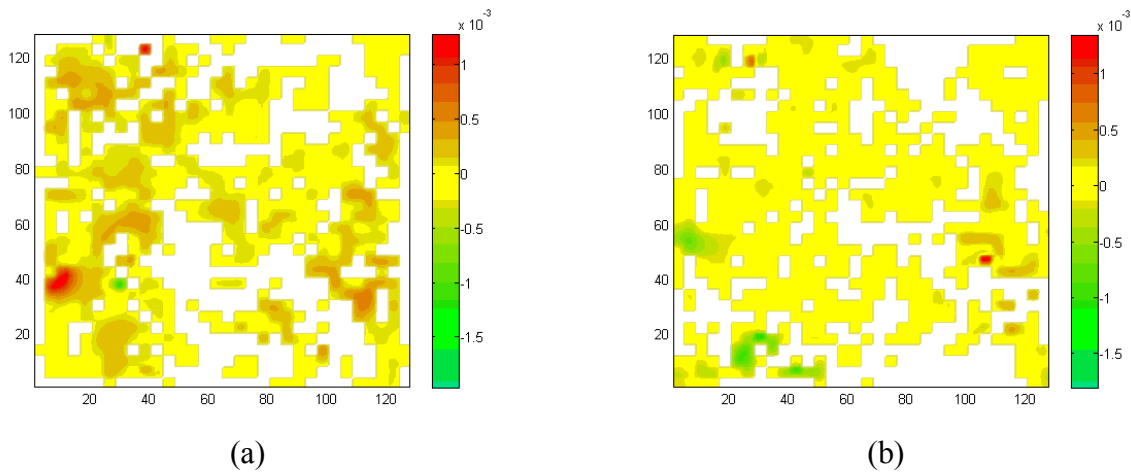


Figure 5.34. Comparison of the normalized mass flux contours for O_2 using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) plane 50 and (b) plane 75.

The normalized mass flux contours in Figures 5.34 and 5.35 illustrate phenomenon. Figures 3.34 (a) and 3.35 (a) correspond to plane 50 in Figure 5.31 (b) where the surface-averaged mass fluxes for oxygen and nitrogen are near their peak. They show more a positive flow than that in Figures 3.34 (b) and 3.35 (b) that correspond to plane 75, near the lowest peak in Figure 5.31 (b). Oxygen is often the most affected by the electrochemical reactions of the chromium oxide vapor (see equations (4.46) through (4.49) in Chapter 4) because its molecular weight is higher than that of either water or nitrogen.

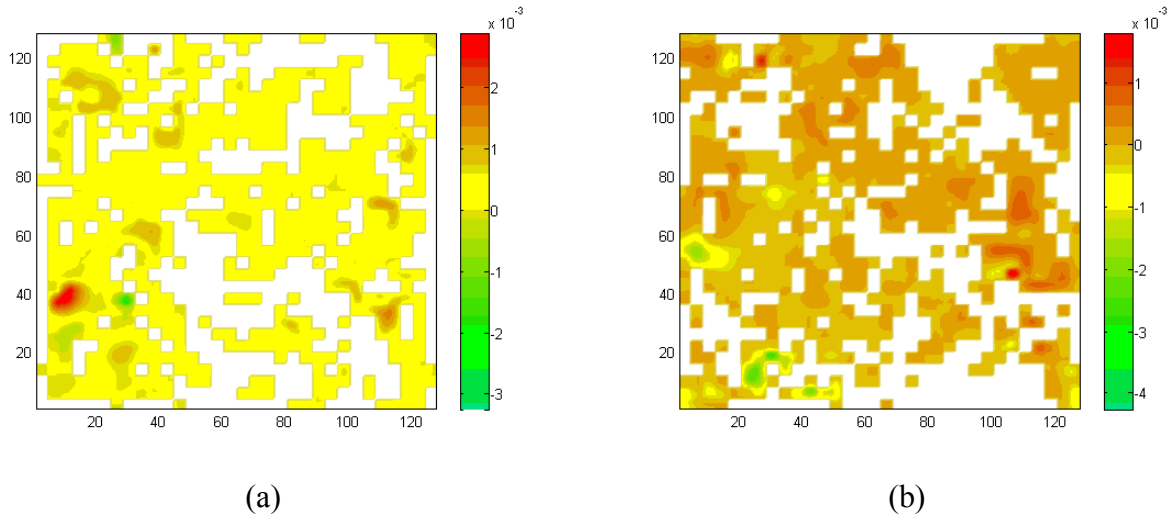
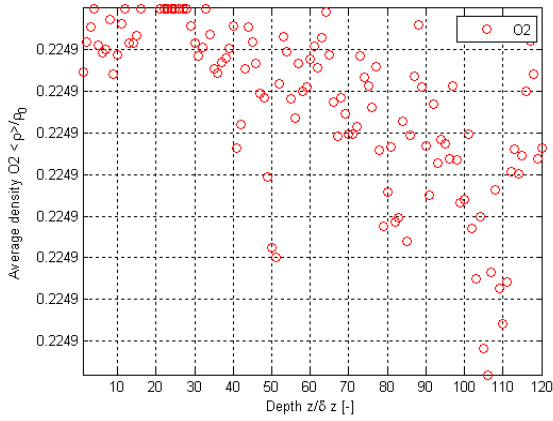


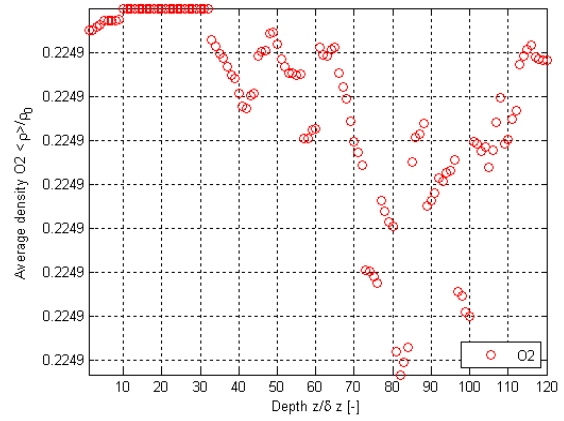
Figure 5.35. Comparison of the normalized mass flux contours for N_2 using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) plane 50 and (b) plane 75.

5.3.2 Mixture with Blocking of the TPBs for Geometry B

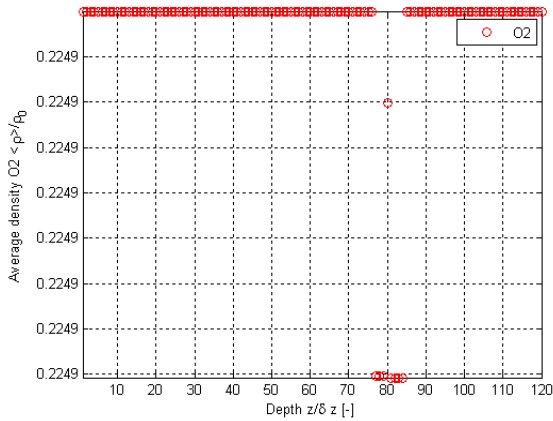
Figures 5.36 and 5.37 show a series of density profiles for the oxygen and chromium oxide vapor for a series of time steps, or collisions. To model chromium poisoning the assumption is made that the TPB site is immediately and permanently blocked if the concentration of the chromium oxide vapor is above a given threshold resulting in it and not the oxygen species reacting (see Section 4.3.2 in Chapter 4). When the density for both species is constant throughout the geometry, all of the TPBs are blocked because the flow is driven by concentration changes which no longer occur when all the TPBs are blocked. After 1000 collisions the oxygen is no longer reacting (Figure 5.36 (d)). The chromium oxide vapor continues to react, but ceases reacting after 3000 collisions (Figure 5.37 (e)).



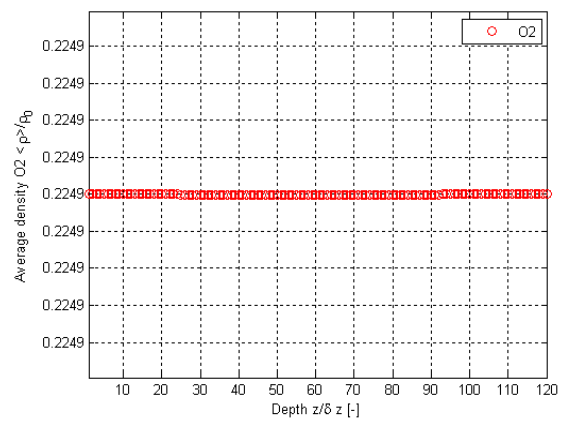
(a)



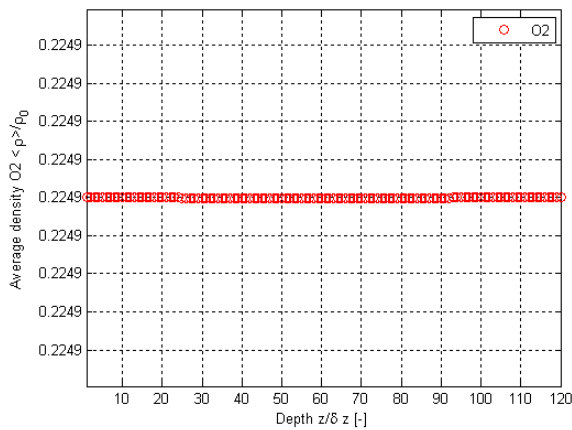
(b)



(c)



(d)



(e)

Figure 5.36. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after (a) 10 collisions, (b) 50 collisions, (c) 500 collisions, (d) 1000 collisions, (e) 3000 collisions.

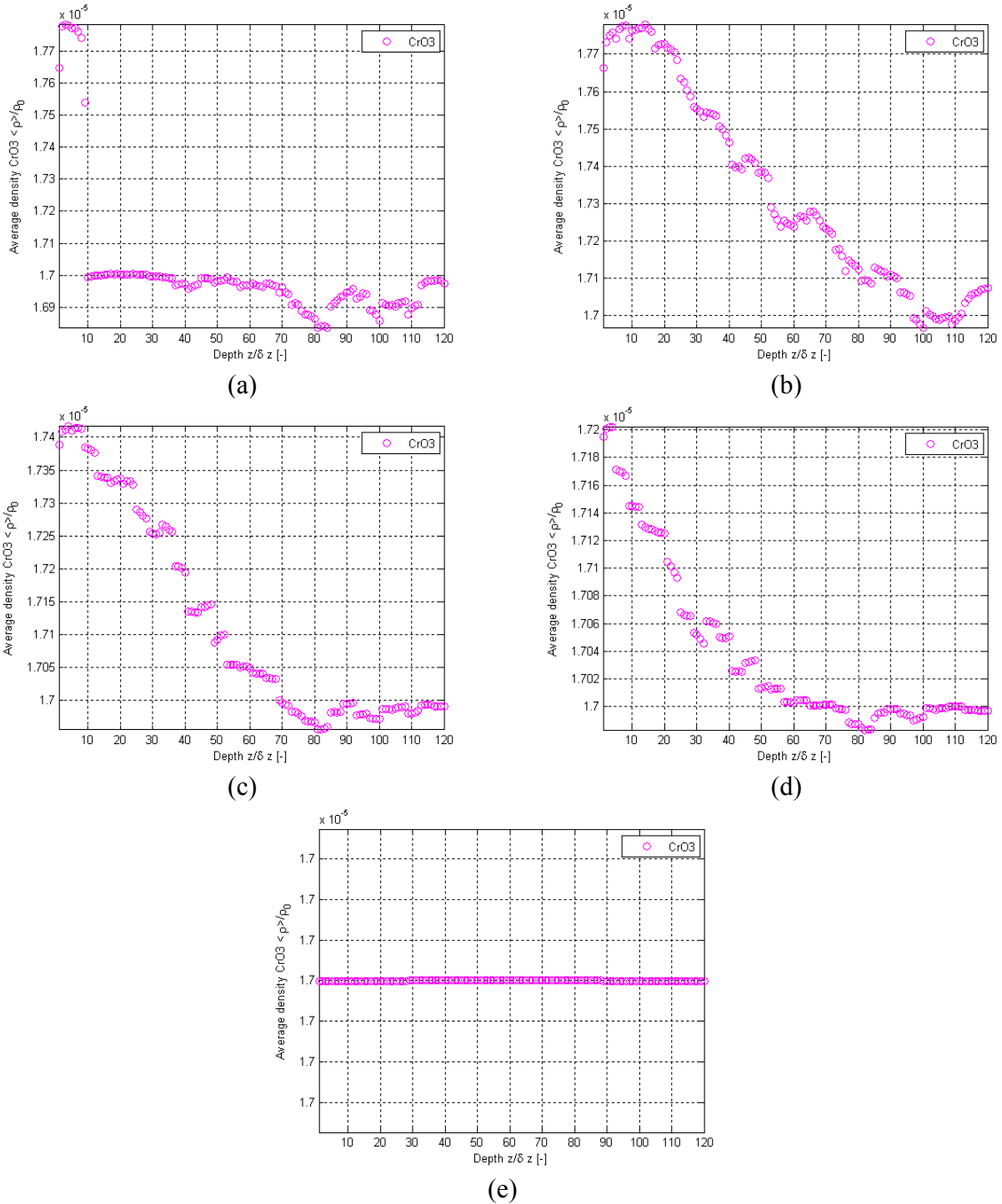


Figure 5.37. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after (a) 10 collisions, (b) 50 collisions, (c) 500 collisions, (d) 1000 collisions, (e) 3000 collisions.

The number of collisions to shutdown is a dimensionless time \hat{t} introduced in Section 3.2.4 in Chapter 3 and is related to time t by

$$\hat{t} = \frac{t}{t_s} \quad (5.6)$$

where t_s is a characteristic time that describes the slow fluid dynamics and is in turn related to the porosity, δ , of the geometry using

$$\frac{t_c}{t_s} = \delta^2 \quad (5.7)$$

t_s in equation (5.7) is a characteristic time that describes the collision phenomena and is defined in equation (4.52) and repeated here for convenience

$$t_c = \frac{dx}{c} \quad (5.8)$$

where dx is defined in equation (4.51) and c in equation (4.4). Combining equations (5.6) through (5.8) and solving for time t

$$t = \frac{\hat{t} dx}{c \delta^2} \quad (5.9)$$

When \hat{t} is the number of collisions required for the density curves for oxygen and chromium oxide vapor to become flat due to the TPB blocking, t is the actual time to shutdown for the portion of the cathode represented in the model. This time can be scaled up linearly to estimate the time to shutdown of the entire cathode assuming that the geometry of the entire cathode is made of a series of geometries that exactly match the reconstructed geometry B, i.e.,

$$\frac{\hat{t}_{portion}}{V_{portion}} = \frac{\hat{t}_{cathode}}{V_{cathode}} \quad (5.10)$$

The volume $V_{cathode}$ of the active layer of the cathode in the ASC1 model SOFC from H. C. Starck is 200 mm x 200 mm x 4.6 μm (see Appendix B). The time to shutdown for Geometry B is 1.64 hr.

The geometries created in this thesis work are representative of a small portion of the entire cathode active layer. Each geometry has different characteristics (even those with the same porosity as the connectivity of the geometry is also taken into account; see Section 3.3.2 in Chapter 3) and as such is different from the actual geometry. The assumption in equation (5.10) is that the entire cathode layer is made up of a series of geometries that are the same as the reconstructed representative geometry. Therefore, in order to get a true picture of the time to shutdown for a specific cathode active layer, a series of different representative geometries must be generated in order to accurately represent the whole cathode geometry.

5.3.3 Mixture without Blocking of the TPBs for Geometry C

The results for the four species simulation without TPB blocking are shown in Figures 5.38 through 5.40. The tortuosity curves in Figure 5.38 (a) are similar to those seen in the three species mixture case for Geometry C (Figures 5.19 (a) and (b)). The tortuosity of the chromium oxide vapor shows large spikes almost immediately as its surface-averaged mass flux stays near zero. As mentioned in Section 5.1.2, the tortuosity after 15 planes is not shown as it is not meaningful because the mass fluxes are near zero (see Figure 5.38 (b)).

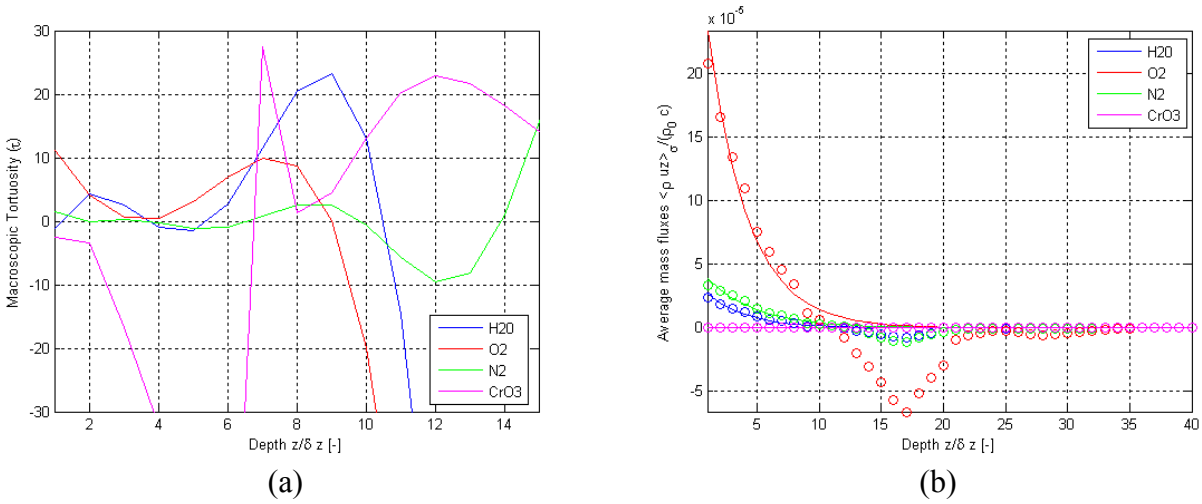


Figure 5.38. Results of this thesis work for variations as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) τ and (b) normalized surface-averaged mass flux.

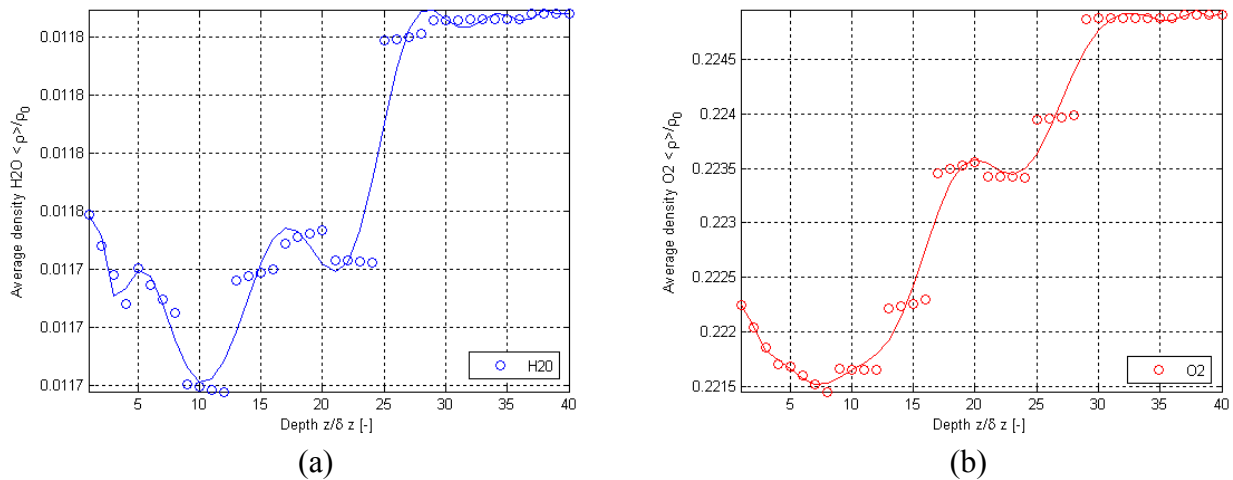


Figure 5.39. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) H_2O and (b) O_2 .

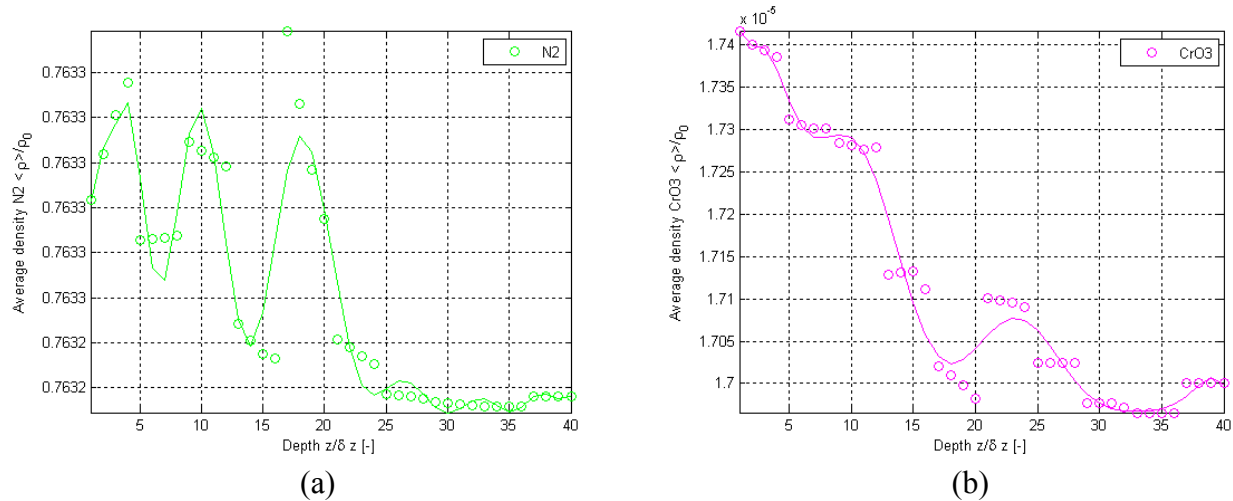
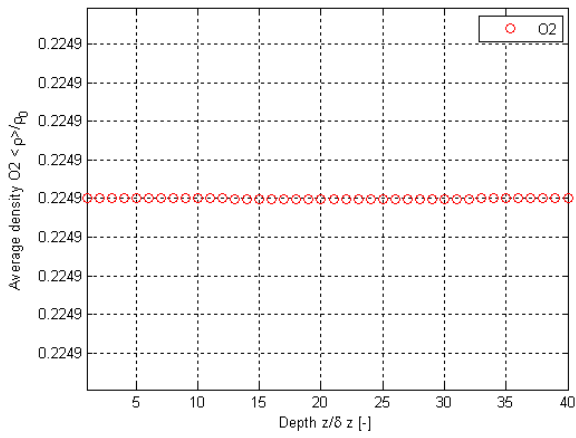


Figure 5.40. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) N_2 and (b) CrO_3 .

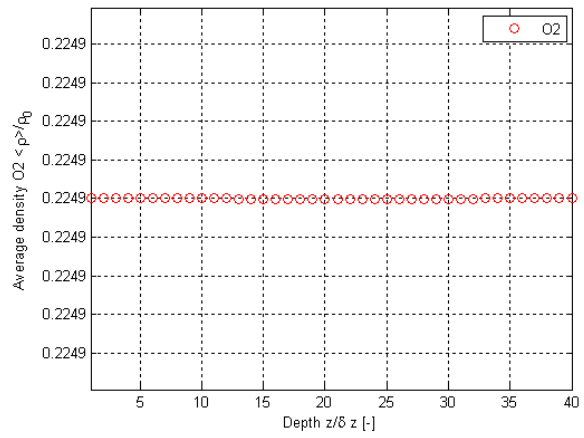
The surface-averaged density trends (Figures 5.39 and 5.40) are the same as in the three species case (Figures 5.19 through 5.21). The surface-averaged mass flux trends (Figure 5.38 (b)) are similar with the exception of a dip in the oxygen mass flux due to chromium oxide vapor reacting as mentioned previously. The mass flux contours can be seen in Appendix E (Figures E.118 through E.125). The sharp nature of the density profiles, especially for nitrogen, are due to the low porosity and the fact that the nitrogen (as well as the oxygen and water) fluid path is affected by the intermolecular forces from the other species as discussed in Section 5.3.1, especially the heavier chromium oxide vapor.

5.3.4 Mixture with Blocking of the TPBs for Geometry C

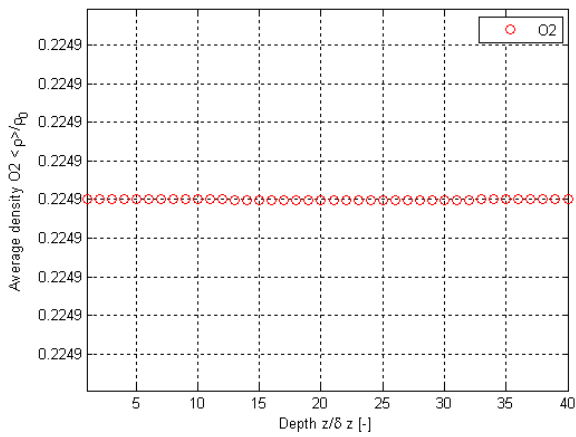
Figures 5.41 and 5.42 show the time evolution of the density profiles for oxygen and chromium oxide vapor. The oxygen does not react from the start. In contrast to the oxygen, the chromium oxide vapor density profile is constant after 1000 collisions throughout the geometry with the exception of the first few planes (Figure 5.42 (d)). This is because density of the chromium oxide vapor in the geometry is not the same as at the inlet/outlet plane and, thus, the first few planes are a transition between the inlet/outlet plane density and the density deeper in the geometry. Using the method and equations discussed in Section 5.3.2, the time to shutdown that corresponds to 1000 collisions for geometry C is 169.3 hr.



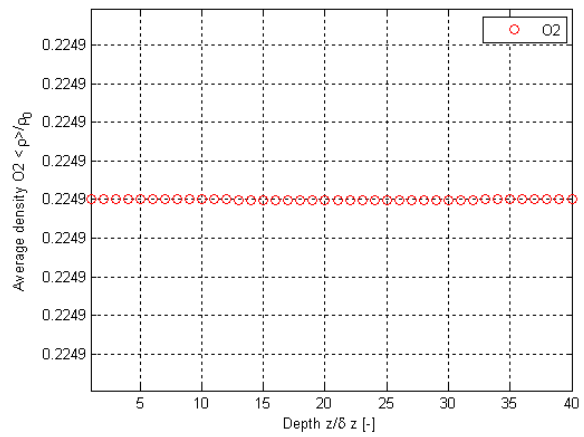
(a)



(b)



(c)



(d)

Figure 5.41. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after (a) 10 collisions, (b) 50 collisions, (c) 500 collisions, (d) 1000 collisions.

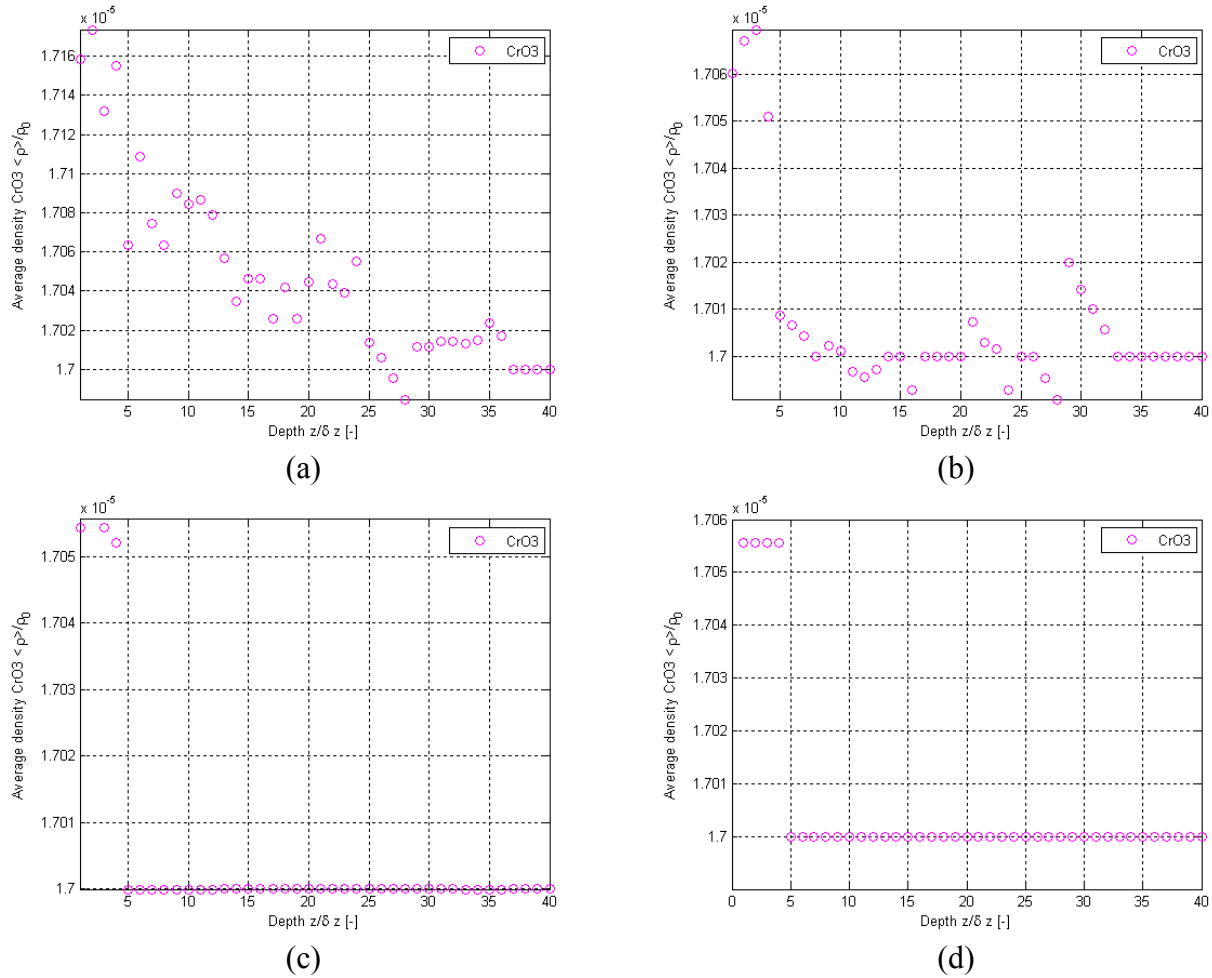


Figure 5.42. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for CrO₃ after (a) 10 collisions, (b) 50 collisions, (c) 500 collisions, (d) 1000 collisions.

5.3.5 Mixture without Blocking of the TPBs for Geometries D and E

Tables 5.12 and 5.13 show the volume-averaged property summaries for four species in Geometries D and E, respectively. As with the four species in Geometry C, the surface-averaged density profiles and surface-averaged mass fluxes are similar to the three species with the exception of the anomalies due to chromium oxide vapor undergoing an electrochemical reaction. These property graphs can be seen in Appendix E (Figures E.136 through E.147). Tables 5.12 and 5.13 show mass fluxes of the same order of magnitude as the three species case of Tables 5.8 and 5.9 (Geometries D and E, respectively). This is different than what is seen in Table 5.11 (the four species mixture in Geometry A) because the chromium oxide vapor is not moving as fast here as in Geometry A due to the smaller cell size in the representative geometry as discussed in Section 5.1.2.

Table 5.12. Volume-averaged property summary for the results from this thesis work using Geometry D and the H₂O, O₂, N₂ and CrO₃ mixture.

Property	H ₂ O	O ₂	N ₂	CrO ₃
$\langle\langle \rho_\sigma u_\sigma \rangle\rangle / \rho_0 c_0$	1.386e-6	1.196e-5	1.831e-6	9.023e-10
$\langle\langle \rho_\sigma \rangle\rangle / \rho_0$	0.01130	0.2086	0.7632	1.602e-5
τ	0.7546 +/- 14.3	-4.411 +/- 20.0	-2.253 +/- 3.50	101.5 +/- 255.

Table 5.13. Volume-averaged property summary for the results from this thesis work using Geometry E and the H₂O, O₂, N₂ and CrO₃ mixture.

Property	H ₂ O	O ₂	N ₂	CrO ₃
$\langle\langle \rho_\sigma u_\sigma \rangle\rangle / \rho_0 c_0$	2.490e-6	2.200e-5	5.591e-6	5.391e-10
$\langle\langle \rho_\sigma \rangle\rangle / \rho_0$	0.01141	0.2167	0.7633	1.633e-5
τ	-1.902 +/- 8.20	-22.02 +/- 35.2	0.2277 +/- 0.728	-3669. +/- 24239.

5.3.6 Mixture without Blocking of the TPBs for Geometry F

Figures 5.43 through 5.45 show the results of four species in Geometry F with no TPB blocking. The tortuosity (Figure 4.43 (a)) is, on average similar to the tortuosity for the 3 species mixture in Geometry F (Figures 5.25 (a) and (b)) but has more spikes due to the fact that the fluid path of each species is affected by the forces of the other species, especially the heavier chromium oxide vapor (see Section 5.3.1). The surface-averaged mass fluxes between the 3 and 4 species mixture cases for Geometry F (Figures 5.25 (c) and 5.43 (b), respectively) are also similar with the exception of the dip in the oxygen surface-averaged mass flux in the 4 species mixture case.

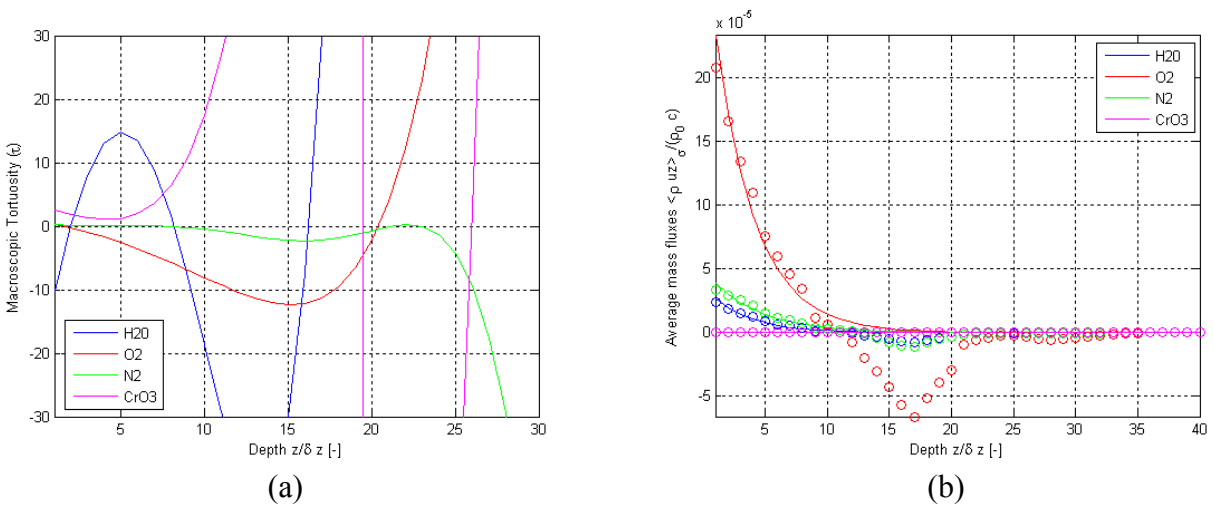


Figure 5.43. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) τ and (b) normalized surface-averaged mass flux.

The density profiles (Figures 5.44 and 5.45) are similar to both the 3 species simulation with the same geometry (Figures 5.26 through 5.27) and the 4 species simulation in Geometry C (Figures 5.39 through 5.40).

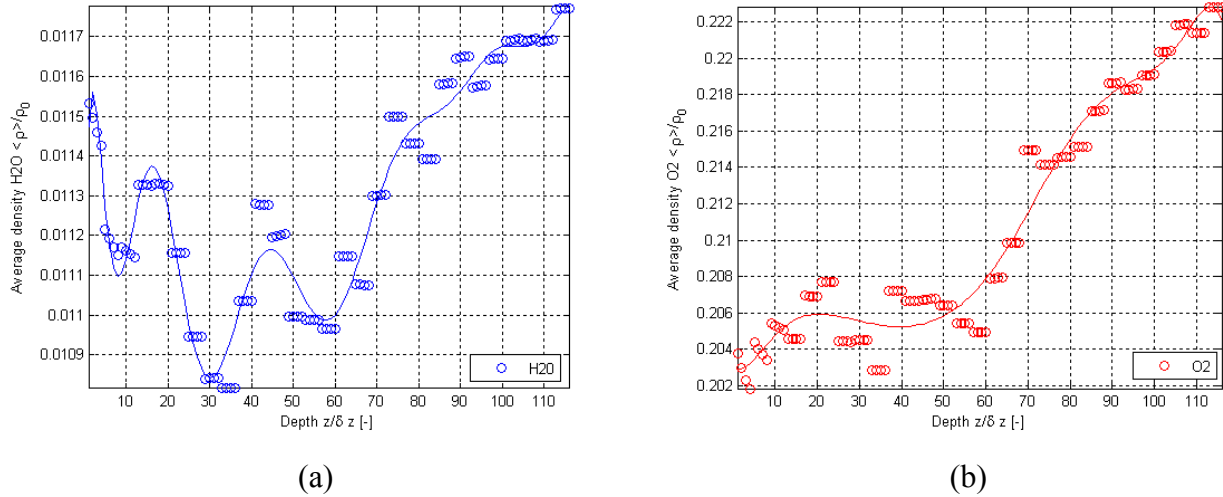


Figure 5.44. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) H_2O and (b) O_2 .

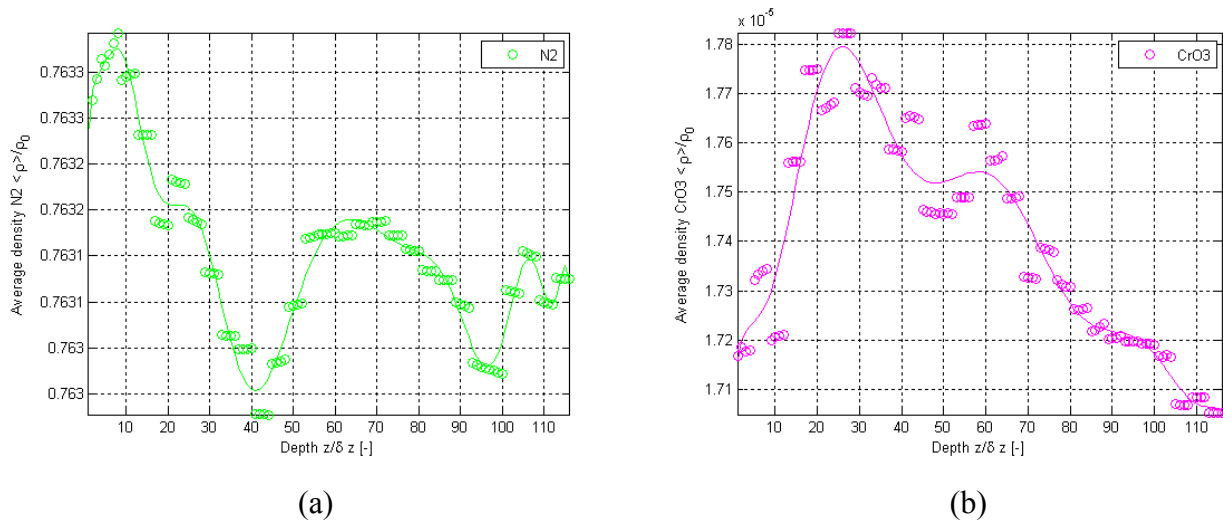


Figure 5.45. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) N_2 and (b) CrO_3 .

Figures 5.47 through 5.49 show the normalized mass flux contour plots. The inlet/outlet and intermediate planes for all species show similar areas of high mass flux. Figures 5.47 and

5.49 show areas of negative mass flux due to the concentration gradients from the oxygen and chromium oxide vapor electrochemically reacting.

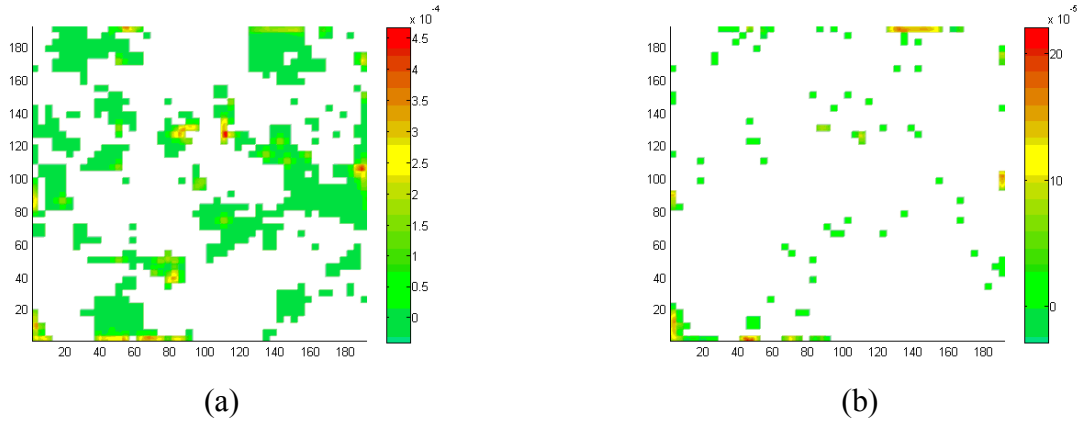


Figure 5.46. Normalized mass flux contours for H₂O using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

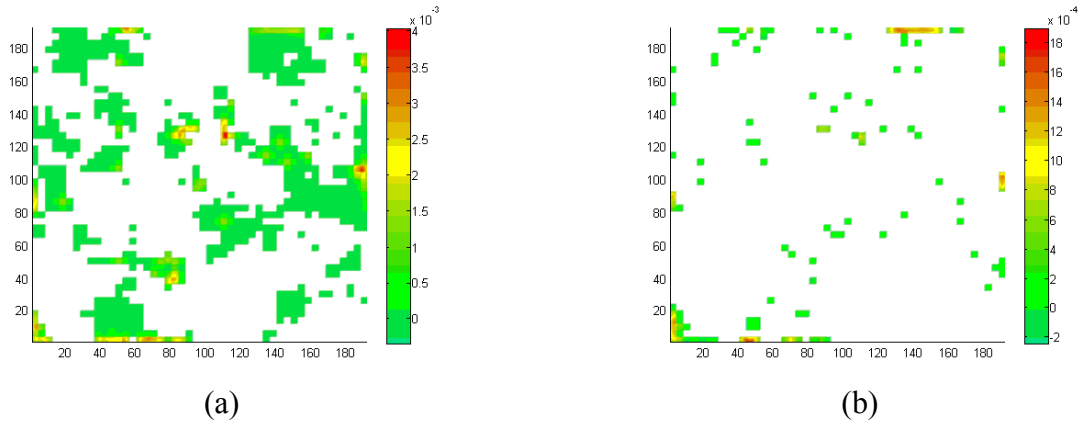


Figure 5.47. Normalized mass flux contours for O₂ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

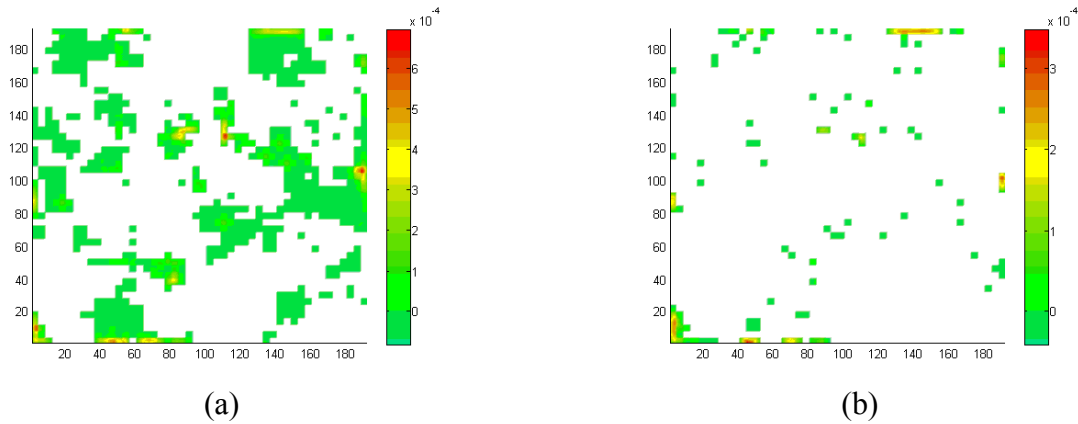


Figure 5.48. Normalized mass flux contours for N₂ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

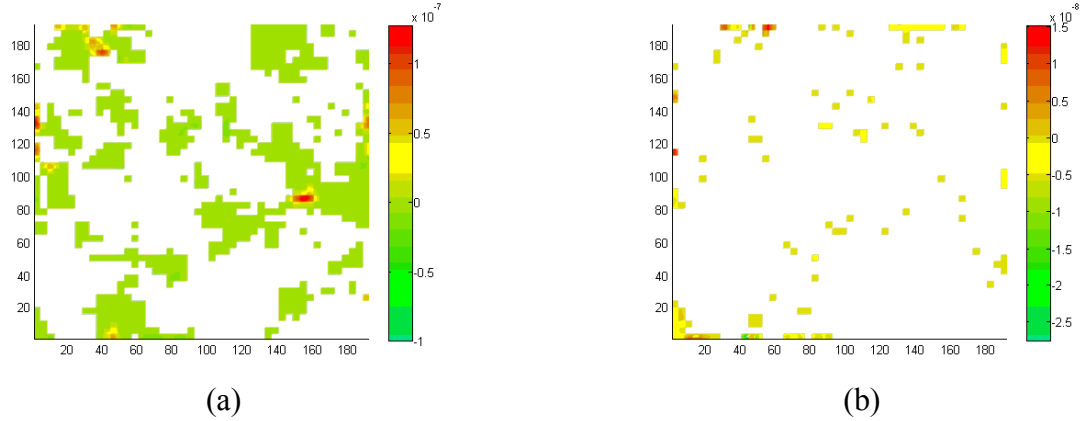


Figure 5.49. Normalized mass flux contours for CrO_3 using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for (a) the inlet/outlet plane, (b) an intermediate plane.

5.3.7 Mixture with Blocking of the TPBs for Geometry F

The results for the 4 species simulation with TPB blocking are in Appendix E (Figures E.162 through E.171). They show that TPBs are completely blocked after 30,000 collisions. It takes many more collisions than Geometries B and C (Figures 5.36, 5.37, 5.41 and 5.42, respectively) to block all the TPBs in this case because there are more TPBs than in the other cases examined, i.e., Geometries B and C. Using the method and equations discussed in Section 5.3.2, the time to shutdown that corresponds to 30,000 collisions for Geometry F is 1504.6 hr. The times to shutdown for Geometries C (169.3 hr) and F bracket the time shown in the experimental results of Figure 1.4 [7] where the cathode is ineffective after about 500 hr.

5.3.8 Mixture without Blocking of the TPBs for Geometry G

Table 5.14 shows the volume-averaged properties for Geometry G with the 4 species mixture. The graphs can be seen in Appendix E (Figures E.172 through E.177).

Table 5.14. Volume-averaged property summary for the results from this thesis work using Geometry G and the H_2O , O_2 , N_2 and CrO_3 mixture.

Property	H_2O	O_2	N_2	CrO_3
$\langle\langle \rho_\sigma u_\sigma \rangle\rangle / \rho_0 c_0$	1.587e-6	1.401e-5	4.244e-6	8.867e-10
$\langle\langle \rho_\sigma \rangle\rangle / \rho_0$	0.01150	0.2148	0.7633	1.579e-5
τ	-1.173 +/- 6.40	3.754 +/- 8.16	-1.680 +/- 1.58	29.41 +/- 63.0

The surface-averaged mass flux profiles are similar to the 3 species simulation with the exception of anomalies the due to chromium oxide vapor reacting. The water and nitrogen

density profiles of the 4 species simulation are similar to those of the 3 species case. The oxygen profiles, however, show an opposite trend. In the 3 species case, the oxygen density tends to decrease into the geometry; but in the 4 species case, it increases into the geometry. This is because the chromium oxide vapor is reacting heavily (as seen by a large decrease in its density) and creating a concentration gradient requiring oxygen to “fill the space.” Table 5.14 shows values the same order of magnitude as in Tables 5.12 and 5.13, results for the 4 species mixture in Geometries D and E.

Chapter 6: Conclusions and Recommendations

6.1 Conclusions – Problem, Models, and Methods

This thesis work includes a description of chromium poisoning and how it affects the cathode of a SOFC; the development of a system of BGK master equations to model the poisoning process; the implementation of the LB solution, the development and application of one- and two-point statistics reconstruction methods, and the validation and application of these models, methods and solutions to the cathode of a SOFC. Attempts at preventing chromium poisoning have been made by experimenting with alternate materials and coatings. Although interconnects with coatings show improvements with regards to the leaching of chromium, there is still a small amount of chromium oxide vapor released. The coatings also may or may not be durable enough to withstand use for an acceptable length of time. Rather than testing each new coating for effectiveness and durability with a fuel cell, the interconnect is tested separately, measuring the amount of chromium oxide vapor released over time. A reacting mixture flow model of the cathode of a SOFC can take this data to predict the effects of the chromium oxide vapor and the time it takes to shutdown all the TPBs and this can be done without ruining an expensive fuel cell for each new coating.

In this thesis work, the BGK master equations are developed from Kinetic Theory in order to model the reacting mixture flow in the cathode. The BGK master equations allow the modeling of these types of flows at the mesoscopic level provided the Boltzmann Gas Limit assumptions hold true. The BGK master equations are well suited to parallel processing as the distribution functions which are tracked can be solved for each node independently from the surrounding nodes. This decreases the computational time drastically which is important as the computational time for the BGK master equations is large and restricts the size of the geometry that can be used in the simulations. The LB method for solving the BGK master equations is well-suited for modeling flow in a porous medium as it requires a grid of nodes overlaying the media which can easily contain information about each fluid and the solid cells. Unfortunately, the LB method does not reduce the computational size of the BGK master equations.

In order to solve the BGK master equations, a 3D digital representation of the porous electrode must be obtained. A relatively simple and inexpensive method to do so is to

stochastically reconstruct it from a microscopic 2D SEM/TEM image of the electrode. Two common methods are the one- and two-point statistics methods. One-point statistics uses only information about the pores to reconstruct the medium and assumes the pores have fixed sizes and regular shapes throughout the medium. It is not practical for media with porosities less than 22%. On the other hand applying one-point statistics is simple and can reconstruct more than two phases (e.g., pores, LSM and YSZ) at the same time and, thus, is faster than the two-point statistics method. Two-point statistics can only handle two phases (pore and one solid phase) in the geometry at a time and must be run multiple times to create a 3D geometry containing more than two phases. The two-point statistics, however, is a more accurate reconstruction method as it retains the exact porosity of the microscopic 2D image and uses both pore and pore connectivity information to reconstruct the 3D geometry.

6.2 Conclusions - Code

The model found in Asinari et al. [1] was developed for a two-fluid mixture in which both species are involved in the electrochemical reactions. The distribution function equations used in this thesis work required no modification to include the additional species required as they are the same for all species, but the code had to first be understood and then be carefully checked to modify all lines that applied to only the two species mixture. The boundary conditions needed to be reformulated to incorporate the additional non-reacting species and the TPB blocking. Initially, the third and fourth species in the mixture made the code unstable. The problem was related to the barycentric φ (discussion in Chapter 4 surrounding equations (4.3), (4.5) and (4.6)). The barycentric φ provides coupling between the mixture and species mass fluxes. In order to eliminate the instability, this value was set constant for the cells with B.C.s (i.e., inlet/outlet B.C.s or TPB B.C.s) where it is not required and left to be calculated for the fluid cells. The TPB blocking B.C. was implemented by simply choosing whether O_2 or CrO_3 was going to react based on a threshold value; and if the CrO_3 reacted, the indicator for that node was changed from TPB to fluid cell. A threshold was required because the LB code requires all four species to be present at each node even though research shows that if there is any CrO_3 at the TPB it will react and not the O_2 .

The choice of a refinement factor of 4 as opposed to 6 or 8 [1] for this thesis is based on the assumption that the tradeoff between the computational error introduced and the

computational time is an acceptable one. This assumption is based on the results given in Asinari et al. [1], initial results at the beginning of this research, and the substantial increase in computational time. Therefore, in this thesis work, it was assumed that the gain in computational time that permitted running a much larger number of cases due to a refinement factor of 4 instead of 6 or 8 sufficiently offset the maximum computational which is less than 5% for all properties and all species with the exception of the volume-averaged N_2 mass flux in the 3 and 4 species mixtures. In this case, the error with a refinement factor of 4 turned out to be significantly higher. See Appendix G for preliminary results with a refinement factor of 8.

6.3 Conclusions and Recommendations - Simulation Results

The parametric studies in this thesis work were used to verify the geometries and types of mixtures considered and helped provide insight into the phenomena present in SOFC electrodes. Below is a list of important conclusions made in this thesis work:

- The number of species in the mixture affects the tortuosities of the species. The more species, the larger and more frequent the spikes in the tortuosity curves. The fluid paths are affected by the forces of the other species in the diffusivity equation. The diffusivity equation uses a collision integral and characteristic length in order to model the interactions between the species.
- The physical size of one cell heavily affects the surface-averaged mass flux contours, i.e., the fluids slow down much more quickly in the geometries created in this thesis work than in the geometries developed in Asinari et al. [1] and Kasula et al. [2] because the physical size of the cells are $0.0367 \mu\text{m}$ versus $0.25 \mu\text{m}$, respectively.
- The average size of the reconstructed pores also affects the results which can be seen in a case where in two different geometries (e.g., 32^3 versus 48^3) with different porosities (e.g., 50% versus 30%), the species flow results have very similar trends because the average size of the reconstructed pores is similar. Note that the actual pore size of the electrode is not different but due to the reconstruction technique the reconstructed geometries mentioned appear to have similar pore sizes.
- The porosity of the geometry especially affects the results because when it is low, the fluid cells near the electrolyte do not have connectivity and, thus, there are no TPBs in that region so that the region itself is irrelevant to the performance of the electrode.

- The number of the TPBs does not affect the results for the fluid properties when compared to the other geometry properties but it does significantly affect the number of time steps required to “shutdown” the electrode. The more TPBs in the geometry, the longer it lasts before it shuts down. Geometry F takes the longest time to shutdown as it has the largest number of TPBs. The predicted time to shutdown for the active layer of the cathode for geometries C and F are significantly different because the geometries created from the 2D image have different average pore sizes as the area represented in geometry F is larger than that in geometry C.

The first recommended improvement to the LB code results would be to rerun all of the simulations with a higher refinement factor. This will not change the property trends seen in the results in this thesis work, but it will improve the accuracy of all the species property values slightly with the exception of the mass fluxes for N_2 in which case the accuracy will be improved significantly. The next improvement would be to refine the CrO_3 versus O_2 electrochemical reaction threshold by adjusting it so that the results match experimental data for the specific cathode modeled.

A final recommendation would be to use the model in conjunction with experimental research in order to improve the performance of a SOFC by changing the electrode and interconnect configuration to reduce chromium poisoning. The model would allow researchers to simulate the affects of different amounts of chromium oxide vapor released from different interconnects without running expensive fuel cell experiments and ruining the SOFC.

References

1. P. Asinari, M. C. Quaglia, M. R. von Spakovsky and B. V. Kasula, "Direct numerical calculation of the kinematic tortuosity of reactive mixture flow in the anode layer of solid oxide fuel cells by the lattice Boltzmann method," *Journal of Power Sources*, 170, 359-375, 2007.
2. B. V. Kasula, L. Mercado, P. Asinari and M. R. von Spakovsky, "3D microstructure reconstructions of solid oxide fuel cell electrodes with applications to numerical simulations of reacting mixture flows using LBM," *Proceedings of IMECE 2007*, Seattle, Washington, USA, November 11-15, 2007.
3. A. Dicks and J. Larminie, *Fuel Cell Systems Explained*, Second Edition, John Wiley and Sons, West Sussex, England, 2003.
4. Fuel cell information guide, retrieved from <http://dodfuelcell.cecer.army.mil/fcdescriptions.php>.
5. K. Kinoshita, "Electrochemical uses of carbon," *Electrochemistry Encyclopedia*, retrieved from <http://electrochem.cwru.edu/estir/>
6. S.P.S. Badwal, R. Deller, K. Foger, Y. Ramprakash and J.P. Zhang, "Interaction Between Chromia Forming Alloy Interconnects and air Electrode of Solid Oxide Fuel Cells," *Solid State Ionics*, 99, 297-310, 1997.
7. S. Taniguchi, M. Kadowaki, H. Kawamura, T. Yasuo, Y. Akiyama, Y. Miyake and T. Saitoh, "Degradation phenomena in the cathode of a solid oxide fuel cell with an alloy separator," *Journal of Power Sources*, 55, 73-79, 1995.
8. M. Stanislawski, E. Wessel, K. Hilpert, T. Markus and L. Singheiser, "Chromium Vaporization from High Temperature Alloys I. Chromia-Forming Steels and The Influence of Outer Oxide Layers," *Journal of the Electrochemical Society*, 154 (4), A295-A306, 2007.
9. M. Stanislawski, J. Froitzheim, L. Niewolak, W. J. Quadackers, K. Hilpert, T. Markus and L. Singheiser, "Reduction of Chromium Vaporization from SOFC Interconnectors by Highly Effective Coatings," *Journal of Power Sources*, 164, 578-589, 2007.

10. W. Schafer, A. Koch, U. Herold-Schmidt, and D. Stolten, "Materials, interfaces and production techniques for planar solid oxide fuel cells," *Solid State Ionics*, 86-88, 1235-1239, 1996.
11. J. Akikusa, K. Adachi, K. Hoshino, T. Ishihara, and Y. Takita, "Development of a low temperature operation solid oxide fuel cell," *Journal of the Electrochemical Society*, 148(11), A1275-A1278, 2001.
12. P. Singh and N. Q. Minh, "Solid oxide fuel cells: technology status," *International Journal of Applied Ceramic Technology*, 1(1), 5-15, 2004.
13. J. W. Fergus, "Effect of cathode and electrolyte transport properties on chromium poisoning in solid oxide fuel cells," *International Journal of Hydrogen Energy*, 32, 3664-3671, 2007.
14. H. Yokokawa, T. Horita, N. Sakai, K. Yamaji, M. E. Brito, Y. P. Xiong, and H. Kishimoto, "Thermodynamic considerations on Cr poisoning in SOFC cathodes," *Solid State Ionics*, 177, 3193-3198, 2006.
15. S. P. Jiang and Y. D. Zhen, "Mechanism of Cr deposition and its application in the development of Cr-tolerant cathodes of solid oxide fuel cells," *Solid State Ionics*, 179, 1459-1464, 2007.
16. Y. D. Zhen, J. Li, and S. P. Jiang, "Oxygen reduction on strontium-doped LaMnO₃ cathodes in the absence and presence of an iron-chromium alloy interconnect," *Journal of Power Sources*, 162(2), 1043-1052, 2006.
17. S. Harris, *An Introduction to the Theory of the Boltzmann Equation*, Dover Publications, Inc., Mineola, New York, 1999.
18. D. A. Wolf-Gladrow, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models, An Introduction*, Springer-Verlag Berlin Heidelberg, Germany, 2000.
19. Y. Jiao, F. H. Stillinger and S. Torquato, "Modeling heterogeneous materials via two-point correlation functions: Basic principles," *Physical Review E*, 76, 031110, 2007.
20. Y. Jiao, F. H. Stillinger and S. Torquato, "Modeling heterogeneous materials via two-point correlation functions. II. Algorithmic details and applications," *Physical Review E*, 77, 031135, 2008.
21. P. Tong and J. N. Rossettos, *Finite Element Method, Basic Technique and Implementation*, The MIT Press, Cambridge, Massachusetts, 1977.

22. S. V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, USA, 1980.
23. J. N. Reddy and D. K. Gartling, The Finite Element Method in Heat Transfer and Fluid Dynamics, Second Edition, CRC Press LLC, Boca Raton, Florida, 2001.
24. W. F. Ames, Numerical Methods for Partial Differential Equations, Second Edition, Academic Press, Inc., New York, New York, 1977.
25. M. L. James, G. M. Smith and J. C. Wolford, Applied Numerical Methods for Digital Computation with FORTRAN, International Textbook Company, Scranton, Pennsylvania, 1967.
26. H. K. Versteeg and W. Malalasekera, An Introduction to Computational Fluid Dynamics: the Finite Volume Method, Addison-Wesley, London, 1996.
27. R. J. LeVeque, Finite-Volume Methods for Hyperbolic Problems, Cambridge University Press, Cambridge, 2002.
28. R. Li, Z. Chen and W. Wu, Generalized Difference Methods for Differential Equations: Numerical Analysis of Finite Volume Methods, M. Dekked, New York, 2000.
29. T. Kreibich, R. van Leeuwen, and E. K. U. Gross, "Time-dependent variational approach to molecules in strong laser fields," Chemical Physics, 304(1-2), 183-202, 2004.
30. A. Chatterjee and D. G. Vlachos, "An overview of spatial microscopic and accelerated kinetic Monte Carlo methods," Journal of Computer-Aided Materials Design, 14, 253-308, 2007.
31. A. Van der Ven and G. Ceder, "First principles calculation of the interdiffusion coefficient in binary alloys," Physical Review Letters, 94, 045901, 2005.
32. A. Van der Ven, G. Ceder, M. Asta and P. D. Tapesch, "First-principles theory of ionic diffusion with nondilute carriers," Physical Review B, 64, 184307, 2001.
33. R. Car and M. Parrinello, "Unified approach for molecular-dynamics and density-functional theory," Physical Review Letters, 55(22), 2471-2474, 1985.
34. M. Thiele, E. K. U. Gross, and S. Kummel, "Adiabatic approximation in nonperturbative time-dependent density-functional theory," Physical Review Letters, 100(15), 153004.1-153004.4, 2008.

35. Q. Cui, M. Elstner, E. Kaxiras, T. Frauenheim, and M. Karplus, "A QM/MM implementation of the self-consistent charge density functional tight binding (SCC-DFTB) method," *Journal of Physical Chemistry B*, 105(2), 569-585, 2001.
36. J. M. Soler, E. Artacho, J. D. Gale, A. Garcia, J. Junquera, P. Ordejon, and D. Sanchez-Portal, "The SIESTA method for ab initio order-N materials simulation," *Journal of Physics-Condensed Matter*, 14(11), 2745-2779, 2002.
37. P. Ordejon, D. A. Drabold, M. P. Grumbach, and R. M. Martin, "Unconstrained minimization approach for electronic computations that scales linearly with system size," *Physical Review B*, 48(19), 14646-14649, 1993.
38. K. P. Recknagle, R. E. Williford, L. A. Chick, D. R. Rector and M. A. Khaleel, "Three-dimensional thermo-fluid electrochemical modeling of planar SOFC stacks," *Journal of Power Sources*, 113, 109-114, 2003.
39. J. R. Ferguson, J. M. Fiard and R. Herbin, "Three-dimensional numerical simulation for various geometries of solid oxide fuel cells," *Journal of Power Sources*, 58, 109-122, 1996.
40. T. X. Ho, P. Kosinski, A. C. Hoffmann and A. Vik, "Numerical modeling of solid oxide fuel cells," *Chemical Engineering Science*, 63, 5356-5365, 2008.
41. R. Suwanwarangkul, E. Croiset, E. Entchev, S. Charojrochkul, M. D. Pritzker, M. W. Fowler, P. L. Douglas, S. Chewathanakup and H. Mahaudom, "Experimental and modeling study of solid oxide fuel cell operating with syngas fuel," *Journal of Power Sources*, 161, 308-322, 2006.
42. D. L. Hall and R. G. Colclaser, "Transient modeling and simulation of a tubular solid oxide fuel cell," *IEEE Transactions on Energy Conversion*, 14 (3), 749-753, 1999.
43. E. Achenbach, "Three-dimensional and time-dependant simulation of a planar solid oxide fuel cell stack." *Journal of Power Sources*. Vol. 49, pp. 333-348, 1994.
44. B. Kenny and K. Karan, "Mathematical micro-model of a solid oxide fuel cell composite cathode," *Canadian Hydrogen and Fuel Cells conference*, September, 2004.
45. S. H. Chan, X. J. Chen and K. A. Khor, "Cathode micromodel of solid oxide fuel cell," *Journal of the Electrochemical Society*, 152 (1), A164-A172, 2004.
46. P. Costamagna, P. Costa and V. Antonucci, "Micro-modelling of solid oxide fuel cell electrodes," *Electrochimica Acta*, 43 (3-4), 375-394, 1997.

47. T. Ackmann, L.G.J. Haart, W. Lehnert and F. Thom, Proceedings of the Fourth European Solid Oxide Fuel Cell Forum, Lucerne, Switzerland, pp. 431-438, 2000.
48. S. Chen, G. D. Doolen and K. G. Eggert, "Lattice Boltzmann fluid dynamics a versatile tool for multiphase and other complicated flows," Los Alamos Science, 22, 98-111, 1994.
49. J. J. Buckles, R. D. Hazlett, S. Chen, K. G. Eggert, D. W. Grunau and W. E. Soll, "Toward improved prediction of reservoir flow performance-simulating oil and water flows at the pore scale," Los Alamos Science, 22, 112-121, 1994.
50. P. Asinari, "Viscous coupling based lattice Boltzmann model for binary mixtures," Physics of Fluids, 17, 067102, 2005.
51. B. B. Hamel, "Kinetic model of binary mixtures," Physics of Fluids, 8, 3, 418-425, 1965.
52. P. Asinari, Ph. D. Dissertation, Politecnico di Torino, 2005
53. S. Békri, O. Vizika, J.-F. Thovert and P. M. Adler, "Binary two-phase flow with phase change in porous media," International Journal of Multiphase Flow, 27, 477-526, 2001.
54. F. A. L. Dullien, "Porous media: Fluid transport and pore structure," Academic Press, San Diego, 1992.
55. J. R. Wilson, W. Kobsiriphat, R. Mendoza, H. Chen, J. M. Hiller, D. J. Miller, K. Thornton, P. W. Voorhees, S. B. Adler and S. A. Barnett, "Three-dimensional reconstruction of a solid-oxide fuel-cell anode," Nature Materials, 5, 541-544, July 2006.
56. J. H. Dunsmuir, S. R. Ferguson, K. L. D'Amico and J. P. Stokes, "X-ray microtomography- A new tool for the characterization of porous media," SPE 22860, Proceedings of the Annual Technical Conference, Dallas, Texas, October 6-9, 423-430, 1991.
57. E. J. Opila, N. S. Jacobson, D. L. Meyers and E. H. Copland, "Predicting oxide stability in high-temperature water vapor," JOM, 58, 22, 2006.
58. S. C. Paulson and V. I. Birss, "Chromium Poisoning of LSM-YSZ SOFC Cathodes. I. Detailed Study of the Distribution of Chromium Species at a Porous Single-Phase Cathode," Journal of the Electrochemical Society, 151 (11), A1961-A1968, 2004.
59. P. L. Bhatnager, E. P. Gross and M. Krook, "A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems," Physics Review, 94, 511, 1954.
60. P. Welander, "On the temperature jump in a rarefied gas," Arkiv Fysik, 7, 507, 1954.

61. S. Chapman, "On the law of distribution of molecular velocities, and on the theory of viscosity and thermal conduction, in a non-uniform simple monatomic gas," *Phil. Trans. Royal Soc., A*, 216, 279, 1916.
62. S. Chapman, "On the Kinetic Theory of a gas. Part II: A composite monatomic gas: diffusion, viscosity, and thermal conduction," *Phil. Trans. Royal Soc., A*, 217, 115, 1917.
63. D. Enskog, Doctoral dissertation, Uppsala University, Uppsala, Sweden, 1917.
64. D. Burnett, "The distribution of velocities in a slightly non-uniform gas," *Proc. London Math Soc.*, 39, 385, 1935.
65. D. Burnett, "The distribution of molecular velocities and the mean motion in a non-uniform gas," *Proc. London Math Soc.*, 40, 382, 1936.
66. S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases*, Cambridge University Press, Cambridge, England, 1990.
67. H. Grad, "Note on n-dimensional hermite polynomials," *Commun. Pure and Appl. Math.*, 2, 325, 1949.
68. H. Grad, "On the kinetic theory of rarefied gases," *Commun. Pure and Appl. Math.*, 2, 331, 1949.
69. T. Morse, "Kinetic model equations for a gas mixture," *Phys. Fluids*, 7, 12, 2012, 1964.
70. J. M. V. A. Koelman, "A simple lattice Boltzmann scheme for Navier-Stokes fluid flow," *Europhysics Letters*, 15, 6, 603-607, 1991.
71. P. Asinari, "Semi-implicit-linearized multiple-relaxation-time formulation of lattice Boltzmann schemes for mixture modeling," *Physical Review E*, 73, 056705, 2006.
72. Virginia Tech Advanced Research Computing, System X, Retrieved from <http://www.arc.vt.edu/arc/SystemX/>, 2007.
73. I. Kivi, P. Möller, H. Kurig, S. Kallip, G. Nurk and E. Lust, "Development of porous cathode powders for SOFC and influence if cathode structure on the oxygen elecroreduction kinetics," *Electrochemistry Communications*, 10, 1455-1458, 2008.
74. K. Yuan, Y. Ji and J. N. Chung, "Physics-based modeling of a low-temperature solid oxide fuel cell with consideration of microstructure and interfacial effects," *Journal of Power Sources*, 194, 908-919, 2009.
75. S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science, New Series*, 220 4598, 671-680, May 13, 1983.

76. J. Latt and O. Malaspinas, "Choice of units in lattice Boltzmann simulations," Creative Commons Attribution-Share Alike 3.0 License, www.lbmmethod.org, 2008.
77. R. C. Reid, J. M. Prausnitz and T. K. Sherwood, *Properties of Gases and Liquids* (3rd ed.), McGraw-Hill, Inc., United States of America, 1977.
78. C. L. Yaws, *Chemical Properties Handbook*. McGraw-Hill, 1999. Online version available at: http://knovel.com/web/portal/browse/display?_EXT_KNOVEL_DISPLAY_bookid=49&VerticalID=0
79. C. L. Tien and J. H. Lienhard, *Statistical Thermodynamics*, Hemisphere Publishing Corporation, Washington, 1979.
80. W. M. Kays, M. E. Crawford and B. Weigand, *Convective Heat and Mass Transfer* (4th ed.), McGraw-Hill, Inc., New York, NY, 2005.
81. D. R. Lide, *CRC Handbook of Chemistry and Physics* (89th ed.), CRC Press, Boca Raton, FL, 2008. Online version available at: <http://www.hbcpnetbase.com/>
82. I. F. Golubev, *Viscosity of Gases and Gas Mixtures: A Hand Book*, Program for Scientific Translations, Jerusalem, Israel, 1970.
83. P. Asinari and I. V. Karlin, "Quasiequilibrium lattice Boltzmann models with tunable bulk viscosity for enhancing stability," *Physical Review E*, 81, 016702.
84. T. C. Zhang and P. L. Bishop, "Evaluation of tortuosity factors and effective diffusivities in biofilms," *Water Resources*, 28, 11, 2279-2287, 1994.
85. J. M. Zalc, S. C. Reyes and E. Iglesia, "The effects of diffusion mechanism and void structure on transport rates and tortuosity factors in complex porous structures," *Chemical Engineering Science*, 59, 2947-2960, 2004.

Appendix A: Porous Geometry Reconstruction Code

One-point Statistics

Image_to_ObstrctNum.m

```
% This program will take an SEM of the cathode of a SOFC and convert it to
% a binary image. It will then count the number of grains of each of
% twenty sizes
clc
clear all

% Load cathode SEM
M=imread('SEM_BSE_SOFC_cath_act.png');

% Convert image from 3D (because of 8-bit) to 2D
Mmod = zeros(size(M,1),size(M,2));
for i=1:1:size(M,1)
    for j=1:1:size(M,2)
        Mmod(i,j) = mean(M(i,j,:));
    end
end

thrsh_bot = 57;
thrsh_top = 156;
% Convert image to 3D binary
MRGB = zeros(size(M,1),size(M,2),3,'uint8');
for i=1:1:size(M,1)
    for j=1:1:size(M,2)
        if (Mmod(i,j)<thrsh_bot) % Pore
            MRGB(i,j,1) = uint8(0);
            MRGB(i,j,2) = uint8(0);
            MRGB(i,j,3) = uint8(0);
        % LSM, catalyst/electron conductor
        elseif (Mmod(i,j)>=thrsh_bot && Mmod(i,j)<thrsh_top)
            MRGB(i,j,1) = uint8(125);
            MRGB(i,j,2) = uint8(125);
            MRGB(i,j,3) = uint8(125);
        % YSZ, electrolyte/ion conductor
        elseif (Mmod(i,j)>=thrsh_top)
            MRGB(i,j,1) = uint8(200);
            MRGB(i,j,2) = uint8(200);
            MRGB(i,j,3) = uint8(200);
        end
    end
end

% convert 3D binary to 2D and set up the 2D matrix for LBM
TempGeometry = zeros(size(M,1),size(M,2));
for i=1:1:size(M,1)
    for j=1:1:size(M,2)
```



```

    if MRGB(i,j,1) == uint8(0)
        TempGeometry(i,j) = 1; % Pore
    elseif MRGB(i,j,1) == uint8(125)
        TempGeometry(i,j) = 2; % LSM, catalyst/electron conductor
    else
        TempGeometry(i,j) = 3; % YSZ, electrolyte/ion conductor
    end
end
end

% Output original and binary images
figure(1)
image(M)
figure(2)
colormap('default')
image(MRGB)

% Calculation the number of grains in each grain size
NumXCells=size(TempGeometry,1);
NumYCells=size(TempGeometry,2);

Num_o_sizes = 25;
PDFL = zeros(Num_o_sizes,1);
PDFY = zeros(Num_o_sizes,1);

% Count the number of LSM cells of each size
fprintf('Count LSM cells \n')
for d=Num_o_sizes:-1:1;
    fprintf(' %i',d);

    for Main_i=1:1:(NumXCells-d+1);
        for Main_j=1:1:(NumYCells-d+1);
            IsGrain=2;

            % If the space or the ones around it is a fluid or YSZ cell
            % set IsGrain to 0
            for i=Main_i:1:(Main_i+d-1);
                for j=Main_j:1:(Main_j+d-1);
                    if TempGeometry(i,j)==1 || TempGeometry(i,j)==3
                        IsGrain=1;
                    end
                end
            end

            % If the cell is a LSM and the ones around it are as
            % well change the elements to zero and count the cell
            if IsGrain==2
                for i=Main_i:1:(Main_i+d-1);
                    for j=Main_j:1:(Main_j+d-1);
                        TempGeometry(i,j) = 1;
                    end
                end
                PDFL(d) = PDFL(d) + 1;
            end
        end
    end
end
end

```

```

end
end

% Count the number of YSZ cells of each size
fprintf('\nCount YSZ cells \n')
for d=Num_o_sizes:-1:1;
    fprintf(' %i',d);

    for Main_i=1:1:(NumXCells-d+1);
        for Main_j=1:1:(NumYCells-d+1);
            IsGrain=3;

            % If the space or the ones around it is a fluid cell
            % set IsGrain to 0
            for i=Main_i:1:(Main_i+d-1);
                for j=Main_j:1:(Main_j+d-1);
                    if TempGeometry(i,j)==1
                        IsGrain=1;
                    end
                end
            end
            end

            % If the cell is a YSZ and the ones around it are as
            % well change the elements to zero and count the cell
            if IsGrain==3
                for i=Main_i:1:(Main_i+d-1);
                    for j=Main_j:1:(Main_j+d-1);
                        TempGeometry(i,j)=1;
                    end
                end
                PDFY(d) = PDFY(d) + 1;
            end
        end
    end
end

end

% figure(3)
% semilogy(PDFL)
% grid on
% hold on
% axis tight

LSM_num = 0;
YSZ_num = 0;
for i = 1 : Num_o_sizes
    LSM_num = LSM_num + PDFL(i)*i^2;
    YSZ_num = YSZ_num + PDFY(i)*i^2;
end
LSM_frac = LSM_num / (NumXCells * NumYCells);
YSZ_frac = YSZ_num / (NumXCells * NumYCells);
pore_frac = 1 - (YSZ_num + LSM_num) / (NumXCells * NumYCells);
fprintf('\nLSM fraction: %f\n',LSM_frac)
fprintf('YSZ fraction: %f\n',YSZ_frac)
fprintf('Pore fraction: %f\n',pore_frac)

```

```
TotalObstructions=0;
for i=1:1:Num_o_sizes
    TotalObstructions=TotalObstructions + PDFL(i)*i^2 + PDFY(i)*i^2;
end
fprintf('\nTotal Obstructions: %i\n',TotalObstructions)
```

GranulometryLaw-LSM-YSZ.xls

grain size ratio = actual size [µm]/ pixel size of image, using image SEM_BSE_SOFC_cath_act.png:
 grain size ratio: 0.1504 image size (pixels): 123 x 643

"PDFL" from Image_to_ObstrctNum.m

LSM	Pixel Size	Grain Size (µm)	Surface (µm ²)	% Surface	LaSrMnO3/Nt	Volume	%volume
10217	1	0.15037594	231.036237	26.911629	0.750146843	0.00255	9.142157
2121	2	0.30075188	191.848041	22.346898	0.155726872	0.00424	15.18294
706	3	0.45112782	143.682515	16.736468	0.051835536	0.00476	17.05665
343	4	0.60150376	124.099723	14.455419	0.025183554	0.00548	19.64262
137	5	0.7518797	77.4492623	9.0214671	0.010058737	0.00428	15.32342
70	6	0.90225564	56.9845667	6.6376926	0.005139501	0.00377	13.52935
16	7	1.05263158	17.7285319	2.0650599	0.001174743	0.00137	4.910654
8	8	1.20300752	11.5778167	1.3486106	0.000587372	0.00102	3.665095
1	9	1.35338346	1.83164679	0.2133544	7.34214E-05	0.00018	0.652308
1	10	1.5037594	2.26129233	0.2634005	7.34214E-05	0.00025	0.894799
0	11	1.65413534	0	0	0	0	0
0	12	1.80451128	0	0	0	0	0
0	13	1.95488722	0	0	0	0	0
0	14	2.10526316	0	0	0	0	0
0	15	2.2556391	0	0	0	0	0
0	16	2.40601504	0	0	0	0	0
0	17	2.55639098	0	0	0	0	0
0	18	2.70676692	0	0	0	0	0
0	19	2.85714286	0	0	0	0	0
0	20	3.0075188	0	0	0	0	0
0	21	3.15789474	0	0	0	0	0
0	22	3.30827068	0	0	0	0	0
0	23	3.45864662	0	0	0	0	0
0	24	3.60902256	0	0	0	0	0
0	25	3.7593985	0	0	0	0	0
Totals:	13620		858.499633	100	1	0.0279	100

"PDFY" from Image_to_ObstrctNum.m

YSZ	Pixel Size	Grain Size (μm)	Surface (μm ²)	% Surface	YSZ/Nt	Volume	%volume
7810	1	0.15037594	176.606931	22.156028	0.752263533	0.00256	5.968028
1402	2	0.30075188	126.813274	15.90922	0.135041418	0.00367	8.57073
543	3	0.45112782	110.509356	13.86383	0.052302061	0.0048	11.20323
282	4	0.60150376	102.02951	12.8	0.027162396	0.00591	13.79142
162	5	0.7518797	91.5823393	11.489362	0.01560393	0.00663	15.47408
103	6	0.90225564	83.8487195	10.519149	0.009921017	0.00729	17.00086
51	7	1.05263158	56.5096953	7.0893617	0.004912348	0.00573	13.36731
17	8	1.20300752	24.6028605	3.0865248	0.001637449	0.00285	6.65118
7	9	1.35338346	12.8215275	1.6085106	0.000674244	0.00167	3.899468
4	10	1.5037594	9.04516931	1.1347518	0.000385282	0.00131	3.056608
1	11	1.65413534	2.73616372	0.3432624	9.63206E-05	0.00044	1.017086
0	12	1.80451128	0	0	0	0	0
0	13	1.95488722	0	0	0	0	0
0	14	2.10526316	0	0	0	0	0
0	15	2.2556391	0	0	0	0	0
0	16	2.40601504	0	0	0	0	0
0	17	2.55639098	0	0	0	0	0
0	18	2.70676692	0	0	0	0	0
0	19	2.85714286	0	0	0	0	0
0	20	3.0075188	0	0	0	0	0
0	21	3.15789474	0	0	0	0	0
0	22	3.30827068	0	0	0	0	0
0	23	3.45864662	0	0	0	0	0
0	24	3.60902256	0	0	0	0	0
0	25	3.7593985	0	0	0	0	0
Totals:	10382		797.105546	100	1	0.04286	100

CreatePorousSOFC_Cathode.m

```
% This program takes the Porosity and GranulometryLaw as found in
% GranulometryLaw-LaSrMnO3.xls and forms a 3D porous geometry based on that
% information using the Granulometry Law. It outputs a .dat file of the
% Geometry matrix

clc
clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% Input %%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Enter the desired size of the 3D output
Num = 32;
NumXCells=Num;
NumYCells=Num;
NumZCells=Num;

% Enter Porosity and GranulometryLaw as found in
% GranulometryLaw-LaSrMnO3_v2.xls where the shade cutoff is at 117
Porosity=0.30515;
GranulometryLaw=[0.739385848,0.142056075,0.052069426,0.028971963,...
    0.016555407,0.010413885,0.00493992,0.00293725,0.001869159,0.000267023...
    ,0.000267023,0.000133511,0.000133511,0,0,0,0,0,0,0,0,0,0,0,0];
% f(d)

NumClasses=size(GranulometryLaw,2);
ParticlesSize=[1:1:NumClasses].^3; %d_i^3
TotalVolume=NumXCells*NumYCells*NumZCells; % L^3
NormConstant=((1-Porosity)*TotalVolume)/sum(GranulometryLaw.*ParticlesSize);
% N_t3D
% sum(GranulometryLaw.*ParticlesSize)=K
PoreDistribution=round(NormConstant.*GranulometryLaw); %N_i3D

Geometry=ones(NumXCells,NumYCells,NumZCells);

for (d = NumClasses : -1 : 1)
    min_i = 1;
    max_i = NumXCells - (d-1);
    min_j = 1;
    max_j = NumYCells - (d-1);
    min_k = 1;
    max_k = NumZCells - (d-1);

    WhileFlag = PoreDistribution(d);% * ParticlesSize(d); %N_i3D*d_i^3
    fprintf('d=%i, WhileFlag=%i\n',d,WhileFlag);

    InitialWhileFlag(d) = WhileFlag;
    m(d) = 1;
    %test2(d) = 1;

    while WhileFlag && m(d) <= ((Num - d) ^ 3 + 1000)
        % generate a random integer for i,j,k between max_ and min_
```

```

i = min_i - 1 + (ceil(rand * (max_i - min_i + 1)));
j = min_j - 1 + (ceil(rand * (max_j - min_j + 1)));
k = min_k - 1 + (ceil(rand * (max_k - min_k + 1)));

m(d) = m(d) + 1;

FreeFlag = 1;
for in = i : 1 : (i + (d-1))
    for jn = j : 1 : (j + (d-1))
        for kn = k : 1 : (k + (d-1))
            % If any part of the ParticlesSize sized, randomly
            % placed chunk is solid, ie ~= 1, set FreeFlag=0, ie
            % do nothing for the rest of the While loop
            if(Geometry(in,jn,kn) ~= 1)
                FreeFlag = 0;
            end
        end
    end
end

% If no part of the ParticleSize sized, randomly placed chunk is
% solid, ie == 1, put zeros in the chunk and count down WhileFlag
if FreeFlag == 1
    for in = i : 1 : (i + (d-1))
        for jn = j : 1 : (j + (d-1))
            for kn = k : 1 : (k + (d-1))
                Geometry(in,jn,kn) = 0; % LaSrMnO3
            end
        end
    end
    %test2(d) = test2(d) + 1;
    WhileFlag = WhileFlag - 1;
end % if (FreeFlag == 1)

end % while(WhileFlag)
FinalWhileFlag(d) = WhileFlag;
end % for (d = NumClasses : -1 : 1)

% Count the number of Solid and Fuel cells and calculate the constructed
% porosity
SolidCells=0;
FluidCells=0;

for i=1:1:NumXCells
    for j=1:1:NumYCells
        for k=1:1:NumZCells

            if Geometry(i,j,k)==0
                SolidCells=SolidCells+1;
            else
                FluidCells=FluidCells+1;
            end
        end
    end
end
end

```

```

end
CalcPorosity=FluidCells/(SolidCells+FluidCells);
fprintf('The original porosity is: %f\n',Porosity);
fprintf('The porosity of the constructed geometry is: %f\n',CalcPorosity);

% Plot constructed geometry
colormap(gray);
figure(1)
[x,y,z] = meshgrid(1:1:NumXCells,1:1:NumXCells,1:1:NumXCells);
xslice = round(NumXCells/2); yslice = round(NumXCells/2); zslice = ...
    round(NumXCells/2);
h=slice(x,y,z,Geometry,xslice,yslice,zslice);
for p=1:1:size(h,1)
    set(h(p),'LineStyle','none')
end

% writes Geometry3D_SOFC.dat from Geometry
for i = 1:1:NumXCells
    for j = 1:1:NumYCells
        for k = 1:1:NumZCells
            if Geometry(i,j,k)==1
                Geometry3D_SOFC(i,j,k) = 1; % '1' represents pore or fluid;
            elseif Geometry(i,j,k)== 0
                Geometry3D_SOFC(i,j,k) = 2; %'2' represents conducting
            end
        end
    end
end
end

csvwrite('Geometry3D_SOFC_cathode_test.dat',Geometry3D_SOFC);

```


Two Point Statistics

SOFC_cathode3D.m

```
% TWO POINT CORRELATION RECONSTRUCTION
% This program creates a random image and then matches it to a target
% distribution by exchanging two randomly selected pixels of opposite
% species. The new distribution is only accepted if it is closer to the
% target distribution.

clc
clear

% input variables
iter_max = 100000;
energy_threshold = 0.0001;
e_t_txt = num2str(energy_threshold);
x = 48; % size of reconstructed image in "x"
y = x; % size of reconstructed image in "y"
z = x; % size of reconstructed image in "z"
r_end = floor(x/2); % max distance between pixels of the same species must
    % be less than half of "x" and "y"
Temp = 7 * 10 ^ (-6); % "cooling" rate

% Load image
M=imread('SEM_BSE_SOFC_cath_act.png');
% figure(1)
% image(M)
% title('Original Image')

% Convert image to 3D binary
Mmod = zeros(size(M,1),size(M,2),3);
numPore = 0;
numSolid = 0;

% thrsh_bot = 57; % 7% porosity
% thrsh_top = 156; % 7% porosity
thrsh_bot = 128; % 30% porosity
thrsh_top = 168; % 30% porosity
% thrsh_bot = 151; % 50% porosity
% thrsh_top = 178; % 50% porosity

%%%%%%%%% YSZ only model %%%%%%%%%%
L1 = 'Pore, LSM';
L2 = 'YSZ';
T = 'YSZ';
for i=1:1:size(M,1)
    for j=1:1:size(M,2)
        % Pore
        if mean(M(i,j,:)) < thrsh_top
            Mmod(i,j,:) = 0;
            numPore = numPore + 1;
        % YSM, ion conductor
        elseif mean(M(i,j,:)) >= thrsh_top
```

```

        Mmod(i,j,:) = 1;
        numSolid = numSolid + 1;
    end
end
end

%%%%%%%%% LSM Only Model %%%%%%%%%%
% L1 = 'Pore, YSZ';
% L2 = 'LSM';
% T = 'LSM';
% for i=1:1:size(M,1)
%     for j=1:1:size(M,2)
%         % Pore
%         if mean(M(i,j,:)) < thrsh_bot
%             Mmod(i,j,:) = 0;
%             numPore = numPore + 1;
%         % LSM, catalyst/electron conductor
%         elseif mean(M(i,j,:)) >= thrsh_bot && mean(M(i,j,:)) < thrsh_top
%             Mmod(i,j,:) = 1;
%             numSolid = numSolid + 1;
%         % Pore
%         elseif mean(M(i,j,:)) >= thrsh_top
%             Mmod(i,j,:) = 0;
%             numPore = numPore + 1;
%         end
%     end
% end

% figure(2)
% image(Mmod)
% title([T,' Only Image where black is ',L1,' and white is ',L2])

prct = numSolid/(numPore+numSolid);
fprintf('The percentage of %s is: %f\n',L2,prct)

clear M

% Create initial basis 2-pnt correlation function
phi1 = numSolid / (numSolid + numPore); % ones, black
phi2 = numPore / (numSolid + numPore); % zeros, white

% Find the two-pnt correlation function of the original image

% Count the number of black pixels that have another black pixel a distance
% r from it, average the amount for all pixels, repeat for each r
BN = zeros(size(Mmod,1),size(Mmod,2),r_end + 1);
SN = zeros(size(Mmod,1),size(Mmod,2),r_end + 1);
S_hat = zeros(1,r_end + 1);

for r = r_end : -1 : 0 % step thru distances
    n = 1;
    S2_temp = zeros(1,size(Mmod,1)*size(Mmod,2));

    xstart = 1;
    xend = size(Mmod,1);

```

```

ystart = 1;
yend = size(Mmod,2);

BN(xstart:xend,ystart:yend,r + 1) = ...
    countBN(Mmod(:, :, 1), r, xstart, xend, ystart, yend);

for i = 1 : size(Mmod,1)
    for j = 1 : size(Mmod,2)
        % the only pixels "r" away are in positive i
        if i <= r
            SN(i,j,r + 1) = SN(i,j,r + 1) + 1;
            % the only pixels "r" away are in negative i
        elseif i > size(Mmod,1) - r
            SN(i,j,r + 1) = SN(i,j,r + 1) + 1;
        else
            SN(i,j,r + 1) = SN(i,j,r + 1) + 2;
        end

        % the only pixels "r" away are in positive j
        if j <= r
            SN(i,j,r + 1) = SN(i,j,r + 1) + 1;
            % the only pixels "r" away are in negative j
        elseif j > size(Mmod,2) - r
            SN(i,j,r + 1) = SN(i,j,r + 1) + 1;
        else
            SN(i,j,r + 1) = SN(i,j,r + 1) + 2;
        end
        % occupied spaces for each pixel
        S2_temp(n) = BN(i,j,r + 1) / SN(i,j,r + 1);
        n = n + 1;
    end
end

% save percent # of occupied spaces for each "r"
S_hat(r + 1) = mean(S2_temp(1:n-1));
end

% Create the initial structure
M_restrct = zeros(x,y,z,3);
n = 0;
while n <= phil * x * y * z
    x1 = 0 + ceil((x-0).*rand);
    y1 = 0 + ceil((y-0).*rand);
    z1 = 0 + ceil((z-0).*rand);
    if M_restrct(x1,y1,z1,1) == 0
        M_restrct(x1,y1,z1,:) = 1;
        n = n + 1;
    end
end

% Plot 3D slices
figure(3)
[x2,y2,z2] = meshgrid(1:1:x,1:1:y,1:1:z);
xslice = round(x/2);
yslice = round(y/2);
zslice = round(z/2);

```

```

h=slice(x2,y2,z2,M_restrct(:,:, :,1),xslice,yslice,zslice);
colormap(copper(3))
%colorbar('YTickLabel',{'Pore','LSM','YSM'},'ALim',[0 1],...
% 'YTick',[.1666,.5,.83333]);
for p=1:1:size(h,1)
    set(h(p),'LineStyle','none')
end
title('Initial Structure')

% Count the number of black pixels that have another black pixel a distance
% r from it, average the amount for all pixels, repeat for each r
BN = zeros(x,y,z,r_end + 1);
SN = zeros(x,y,z,r_end + 1);
S2 = zeros(1,r_end + 1);

for r = r_end : -1 : 0 % step thru distances
    n = 1;
    S2_temp = zeros(1,x*y*z);

    xstart = 1;
    xend = x;
    ystart = 1;
    yend = y;
    zstart = 1;
    zend = z;

    BN(xstart:xend,ystart:yend,zstart:zend,r + 1) = ...
        countBN3D(M_restrct(:,:, :,1),r,xstart,xend,ystart,yend,zstart,zend);

    for i = 1 : x
        for j = 1 : y
            for k = 1 : z
                % the only pixels "r" away are in positive i
                if i <= r
                    SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 1;
                    % the only pixels "r" away are in negative i
                elseif i > size(M_restrct,1) - r
                    SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 1;
                else
                    SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 2;
                end

                % the only pixels "r" away are in positive j
                if j <= r
                    SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 1;
                    % the only pixels "r" away are in negative j
                elseif j > size(M_restrct,2) - r
                    SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 1;
                else
                    SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 2;
                end

                % the only pixels "r" away are in positive k
                if k <= r
                    SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 1;

```

```

        % the only pixels "r" away are in negative k
    elseif k > size(M_restrct,3) - r
        SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 1;
    else
        SN(i,j,k,r + 1) = SN(i,j,k,r + 1) + 2;
    end

    % occupied spaces for each pixel
    S2_temp(n) = BN(i,j,k,r + 1) / SN(i,j,k,r + 1);
    n = n + 1;
end
end
end

% save percent # of occupied spaces for each "r"
S2(r + 1) = mean(S2_temp(1:n-1));
end

%Calculate the initial energy (the sum of the square of the differences of
%each target and sampled correlation function)
E = sum((S2-S_hat).^2);
E_old = E;

fprintf('Finished Initializing\n')

% Change the image pixel by pixel so that each move lowers the energy of
% the system until it is below the required threshold
p = 1;
q = 1;
S2_store = zeros(iter_max,r_end+1);
while E >= energy_threshold && p <= iter_max

    % exchange a randomly selcted black with a randomly selcted white pixel
    flag = 1;
    while flag
        x1 = 0 + ceil(((x)-0).*(rand));
        y1 = 0 + ceil(((y)-0).*(rand));
        z1 = 0 + ceil(((z)-0).*(rand));
        x2 = 0 + ceil(((x)-0).*(rand));
        y2 = 0 + ceil(((y)-0).*(rand));
        z2 = 0 + ceil(((z)-0).*(rand));
        if M_restrct(x1,y1,z1,1) == 1 && M_restrct(x2,y2,z2,1) == 0
            M_restrct(x1,y1,z1,:) = 0;
            M_restrct(x2,y2,z2,:) = 1;
            flag = 0;
        end
    end
end

% Store the old distribution
BN_old = BN;

% Calculate the sampling 2-pnt correlation
% Count the number of black pixels that have another black pixel a
distance
%r from it, average the amount for all pixels, repeat for each r

```

```

S2 = zeros(1,r_end + 1);
for r = r_end : -1 : 0 % step thru distances
    n = 1;
    S2_temp = zeros(1,x*y*z);

    % Recalculate the BN for pixels "r" away from x1,y1,z1
    xstart = x1 - r;
    xend = x1 + r;
    ystart = y1 - r;
    yend = y1 + r;
    zstart = z1 - r;
    zend = z1 + r;
    if xstart <= 0
        xstart = 1;
    end
    if xend > x
        xend = x;
    end
    if ystart <= 0
        ystart = 1;
    end
    if yend > y
        yend = y;
    end
    if zstart <= 0
        zstart = 1;
    end
    if zend > z
        zend = z;
    end
    end

    BN(xstart:xend,ystart:yend,zstart:zend,r + 1) = ...

countBN3D(M_restrct(:, :, :, 1), r, xstart, xend, ystart, yend, zstart, zend);

    % Recalculate the BN for pixels "r" away from x2,y2,z2
    xstart = x2 - r;
    xend = x2 + r;
    ystart = y2 - r;
    yend = y2 + r;
    zstart = z2 - r;
    zend = z2 + r;
    if xstart <= 0
        xstart = 1;
    end
    if xend > x
        xend = x;
    end
    if ystart <= 0
        ystart = 1;
    end
    if yend > y
        yend = y;
    end
    if zstart <= 0
        zstart = 1;
    end

```

```

end
if zend > z
    zend = z;
end

BN(xstart:zend,ystart:yend,zstart:zend,r + 1) = ...

countBN3D(M_restrct(:, :, :, 1), r, xstart, xend, ystart, yend, zstart, zend);

% Record occupied spaces for each pixel
for i = 1 : x
    for j = 1 : y
        for k = 1 : z
            S2_temp(n) = BN(i, j, k, r + 1) / SN(i, j, k, r + 1);
            n = n + 1;
        end
    end
end
end
% ave percent # of occupied spaces for each "r"
S2(r + 1) = mean(S2_temp(1:n-1));
% fprintf('Finished r = %i\n', r)
end

% Calculate the new energy
E = sum((S2-S_hat).^2);

% if the energy of the system is not reduced, restore it to the
% original state as per Metropolis algorithm
delE = E - E_old;
if delE >= 0
    M_restrct(x1,y1,z1,:) = 1;
    M_restrct(x2,y2,z2,:) = 0;
    BN = BN_old;
else
    Prob = exp(-delE/Temp); % acceptance probability
    R = rand;
    if R > Prob
        M_restrct(x1,y1,z1,:) = 1;
        M_restrct(x2,y2,z2,:) = 0;
        BN = BN_old;
    else
        E_old = E;
        if rem(p,10) == 0 || p == 1
            S2_store(q,:) = S2;
            q = q + 1;
            fprintf('p = %i, E = %f \n', p, E)
        end
        p = p + 1;
    end
end
end % while loop

% Plot the correlation functions
figure(4)
hold on
%plot(0:r_end,S2_image,'r')

```

```

%plot(0:r_end,S_hat0,'b')
plot(0:r_end,S_hat,'k')
for i = 1 : q - 1
    plot(0:r_end,S2_store(i,:), 'm')
end
% plot(0:r_end,S_hat,'k') % this plot is repeated so that the legend will
% % work and so that it will show on top
% legend('Image','UnMOD','Basis','Sample')
legend('Basis','Sample')

% Plot the final structure
% Plot 3D slices
figure(5)
[x2,y2,z2] = meshgrid(1:1:x,1:1:y,1:1:z);
xslice = round(x/2);
yslice = round(y/2);
zslice = round(z/2);
h=slice(x2,y2,z2,M_restrct(:,:,,1),xslice,yslice,zslice);
colormap(copper(3))
for p=1:1:size(h,1)
    set(h(p), 'LineStyle', 'none')
end
title('Final Structure')

% Write the Geometry matrix to file
% csvwrite('LSM_32x32x32.txt',M_restrct(:,:,,1));
M_restrct = M_restrct(:,:,,1);
save([L2, '_r',int2str(r_end), '_err0_',e_t_txt(1,3:end), '__',int2str(x), 'x', ...
    int2str(y), 'x',int2str(z), '_30pore'], 'M_restrct');

```


countBN3D.m

```
function [BN] = countBN3D(M,r,xstart,xend,ystart,yend,zstart,zend)

BN = zeros(xend-xstart+1,yend-ystart+1,zend-zstart+1);

i_count = 1;
for i = xstart : xend
    j_count = 1;
    for j = ystart : yend
        k_count = 1;
        for k = zstart : zend
            if M(i,j,k) == 1
                % the only pixels "r" away are in positive i
                if i <= r
                    if M(i + r,j,k) == 1
                        BN(i_count,j_count,k_count) ...
                            = BN(i_count,j_count,k_count) + 1;
                    end
                % the only pixels "r" away are in negative i
                elseif i > size(M,1) - r
                    if M(i - r,j,k) == 1
                        BN(i_count,j_count,k_count) ...
                            = BN(i_count,j_count,k_count) + 1;
                    end
                else
                    if M(i + r,j,k) == 1
                        BN(i_count,j_count,k_count) ...
                            = BN(i_count,j_count,k_count) + 1;
                    end
                    if M(i - r,j,k) == 1
                        BN(i_count,j_count,k_count) ...
                            = BN(i_count,j_count,k_count) + 1;
                    end
                end
            end

            % the only pixels "r" away are in positive j
            if j <= r
                if M(i,j + r,k) == 1
                    BN(i_count,j_count,k_count) = ...
                        BN(i_count,j_count,k_count) + 1;
                end
            % the only pixels "r" away are in negative j
            elseif j > size(M,2) - r
                if M(i,j - r,k) == 1
                    BN(i_count,j_count,k_count) ...
                        = BN(i_count,j_count,k_count) + 1;
                end
            else
                if M(i,j + r,k) == 1
                    BN(i_count,j_count,k_count) ...
                        = BN(i_count,j_count,k_count) + 1;
                end
                if M(i,j - r,k) == 1
                    BN(i_count,j_count,k_count) ...
                        = BN(i_count,j_count,k_count) + 1;
                end
            end
        end
    end
end
```

```

        end
    end

    % the only pixels "r" away are in positive k
    if k <= r
        if M(i,j,k + r) == 1
            BN(i_count,j_count,k_count) ...
                = BN(i_count,j_count,k_count) + 1;
        end
    % the only pixels "r" away are in negative k
    elseif k > size(M,3) - r
        if M(i,j,k - r) == 1
            BN(i_count,j_count,k_count) ...
                = BN(i_count,j_count,k_count) + 1;
        end
    else
        if M(i,j,k + r) == 1
            BN(i_count,j_count,k_count) ...
                = BN(i_count,j_count,k_count) + 1;
        end
        if M(i,j,k - r) == 1
            BN(i_count,j_count,k_count) ...
                = BN(i_count,j_count,k_count) + 1;
        end
    end
    end
    k_count = k_count + 1;
end
    j_count = j_count + 1;
end
    i_count = i_count + 1;
end

```

Combine_SOFC_Data.m

```
% This program combines the LSM and YSZ only structures created by the Two  
% Point Correlation Reconstruction, SOFC_cathode3D.m
```

```
clear all;
```

```
clc;
```

```
LSM = load('LSM_r24_err0_0001_48x48x48_30pore.mat');
```

```
YSZ = load('YSZ_r24_err0_0001_48x48x48_30pore.mat');
```

```
Geometry3D_LSM = LSM.M_restrct(:,:,,1);
```

```
Geometry3D_YSZ = YSZ.M_restrct(:,:,,1);
```

```
N = size(Geometry3D_LSM,1);
```

```
% Calculate the Original Volume Fraction of YSZ
```

```
Pore = 0;
```

```
Solid = 0;
```

```
for i = 1 : N
```

```
    for j = 1 : N
```

```
        for k = 1 : N
```

```
            if Geometry3D_YSZ(i,j,k) == 1
```

```
                Solid = Solid + 1;
```

```
            else
```

```
                Pore = Pore + 1;
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
Actual_YSZVolumeFraction = Solid / (Solid + Pore);
```

```
% Convert solid pixels of LSM from 1 to 2 and solid pixels of YSZ from 1
```

```
% to 3
```

```
% 1 --> Pore
```

```
% 2 --> LSM
```

```
% 3 --> YSZ
```

```
for k = 1:1:N
```

```
    for i = 1:1:N
```

```
        for j = 1:1:N
```

```
            if Geometry3D_LSM(i,j,k) == 1
```

```
                Geometry3D_LSM(i,j,k) = 2;
```

```
            end
```

```
            if Geometry3D_YSZ(i,j,k) == 1
```

```
                Geometry3D_YSZ(i,j,k) = 3;
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
% set Geometry3D matrix to 1
```

```
Geometry3D = ones(N,N,N);
```

```

PoreCells = 0;
LSMCells = 0;
YSZCells = 0;

for k = 1 : 1 : N
    for i = 1 : 1 : N
        for j = 1 : 1 : N
            % add YSZ then LSM to the Geometry3D matrix
            if Geometry3D_YSZ(i,j,k) == 3
                Geometry3D(i,j,k) = 3;
            end
            if Geometry3D_LSM(i,j,k) == 2
                Geometry3D(i,j,k) = 2;
            end
            % Count the number of cells of each species
            if Geometry3D(i,j,k) == 1
                PoreCells = PoreCells + 1;
            elseif Geometry3D(i,j,k) == 2
                LSMCells = LSMCells + 1;
            elseif Geometry3D(i,j,k) == 3
                YSZCells = YSZCells + 1;
            end
        end
    end
end

% find volume fractions
LSMVolumeFraction = LSMCells / (PoreCells + LSMCells + YSZCells);
YSZVolumeFraction = YSZCells / (PoreCells + LSMCells + YSZCells);

% Convert some pores to YSZ if there is a surrounding YSZ cell until the
% YSZ Volume fraction is correct
while YSZVolumeFraction < Actual_YSZVolumeFraction

    i = ceil(rand*N);
    j = ceil(rand*N);
    k = ceil(rand*N);

    if i-1 >= 1 && i+1 <= N && j-1 >= 1 && j+1 <= N && k-1 >= 1 && ...
        k+1 <= N
        if Geometry3D(i,j,k) == 1 && (Geometry3D(i+1,j,k) == 3 || ...
            Geometry3D(i-1,j,k) == 3 || Geometry3D(i,j+1,k) == 3 || ...
            Geometry3D(i,j-1,k) == 3 || Geometry3D(i,j,k+1) == 3 || ...
            Geometry3D(i,j,k-1) == 3)

            Geometry3D(i,j,k) = 3;

            YSZCells = YSZCells + 1;
            PoreCells = PoreCells - 1;
            YSZVolumeFraction = YSZCells / (PoreCells + LSMCells + YSZCells);
        end
    end
end
% Find Porosity

```

```

Porosity = PoreCells / (PoreCells + LSMCells + YSZCells);

% show Geometry3D figure, take slice from middle of matrix
figure(1)
[x,y,z] = meshgrid(1:1:N,1:1:N,1:1:N);
xslice = round(1); yslice = round(1); zslice = round(N);
h=slice(x,y,z,Geometry3D,xslice,yslice,zslice);
colormap(gray(3))
colorbar('YTickLabel',{'Pore','LSM','YSZ'},'ALim',[1 3],...
        'YTick',[1.25,2,2.75]);
for p=1:1:size(h,1)
    set(h(p),'LineStyle','none')
end
title('3D Image')

% take 2D slice at k=25 and plot
k = 25;
MGeometry = zeros(N,N,3);
for i=1:1:N
    for j=1:1:N
        if (Geometry3D(i,j,k)==3) % Pore
            MGeometry(i,j,1) = 1;
            MGeometry(i,j,2) = 1;
            MGeometry(i,j,3) = 1;
        elseif (Geometry3D(i,j,k)==2) % LSM
            MGeometry(i,j,1) = 0.5;
            MGeometry(i,j,2) = 0.5;
            MGeometry(i,j,3) = 0.5;
        elseif (Geometry3D(i,j,k)==1) % YSZ
            MGeometry(i,j,1) = 0;
            MGeometry(i,j,2) = 0;
            MGeometry(i,j,3) = 0;
        end
    end
end

figure(2)
image(MGeometry)
colormap(gray(3))
colorbar('YTickLabel',{'Pore','LSM','YSZ'},'ALim',[1 3],...
        'YTick',[0,1,9]);
title('2D Slice at k = 25')

csvwrite('Geometry3D_48x48x48_SOFC_2pnt_SEM_BSE1_30pore.dat',Geometry3D);

```

Connectivity Calculation

StdAfx.h

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#ifdef AFX_STDAFX_H__55DE2844_8D84_417D_A722_FB46FE8F1929__INCLUDED_
#define AFX_STDAFX_H__55DE2844_8D84_417D_A722_FB46FE8F1929__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

// TODO: reference additional headers your program requires here
// general library headers
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <math.h>
#include <ctype.h>

// program related headers
#include "Macros.h"
#include "FunctionPrototypes.h"

// I/O headers
#include <iostream>
using namespace std;
#include <fstream>
#include <iomanip>
#include <string>

#endif // AFX_STDAFX_H__55DE2844_8D84_417D_A722_FB46FE8F1929__INCLUDED_ //
```

Macros.h

```
// Definition of macros and constants like maximum vector and
// matrices dimensions

#define NumXCells 48
#define NumYCells 48
#define NumZCells 48
```

FunctionPrototypes.h

```
// Definition of function prototypes

extern double round(double);
extern int Infection3D(int [NumXCells*NumYCells*NumZCells+1][3],int *,
                      int
                      [NumXCells+1][NumYCells+1][NumZCells+1],int,int,int,int,int *);
```

CreatePorousCPP.cpp

```
// CreatePorousCPP.cpp : Defines the entry point for the console application.
// This program takes a 3D matrix (.dat file) and finds the active sites for
the
// reaction to happen. It then outputs the matrix with the active sites
marked
// in a .csv file.

#include "Stdafx.h"

int main(int argc, char* argv[])
{
    // Variables
    //*****
    int i,j,k,in,jn,kn;
    int const MAX_RAND=NumXCells*NumYCells*NumZCells; // =32*32*32
    int NumInfected=0;
    int NumCellsFluidPath=0, Count=0;
    int CooCellsFluidPath[NumXCells*NumYCells*NumZCells+1][3];
    double ActualPorosity,LSMFract,YSZFract;
    int quit;

    int c[19][3]={ {0,0,0}, // 0
                  { 1, 0, 0}, // 1
                  { 0, 1, 0}, // 2
                  {-1, 0, 0}, // 3
                  { 0,-1, 0}, // 4
                  { 0, 0, 1}, // 5
                  { 0, 0,-1}, // 6
                  { 1, 1, 0}, // 7
                  {-1, 1, 0}, // 8
                  {-1,-1, 0}, // 9
                  { 1,-1, 0}, // 10
                  { 1, 0, 1}, // 11
                  { 0, 1, 1}, // 12
                  {-1, 0, 1}, // 13
                  { 0,-1, 1}, // 14
                  { 1, 0,-1}, // 15
                  { 0, 1,-1}, // 16
                  {-1, 0,-1}, // 17
                  { 0,-1,-1}}; // 18
```

```

FILE *fpt;
char p;
int Geometry[NumXCells+1][NumYCells+1][NumZCells+1];
int GeometrySize = 48;

fpt = fopen("Geometry3D_48x48x48_SOFC_2pnt_SEM_BSE1_30pore.dat", "r");

if (fpt==NULL)
{
    perror ("Error opening file");
}
else
{
    i=1;
    j=1;
    k=1;

    p = getc(fpt);

    while (p != EOF)
    {
        while ( k <= (GeometrySize) )
        {
            j = 1;
            while ( j <= (GeometrySize) )
            {
                i = 1;
                while ( i <= (GeometrySize) )
                {
                    if ( (p != ',') & (p != '\n') )
                    {
                        if (p == '1')
                        {
                            Geometry[i][j][k] = 1;
                        }
                        else if (p == '2')
                        {
                            Geometry[i][j][k] = 2;
                        }
                        else if (p == '3')
                        {
                            Geometry[i][j][k] = 3;
                        }
                    }

                    p = getc(fpt);
                }
                i = i + 1;
            }
            p = getc(fpt);
            j = j + 1;
        }
        k = k + 1;
    }
}

```



```

    }
    fclose(fpt);
}

// Print the array Geometry

printf("\n Geometry \n\n");
for (k = 1; k <= GeometrySize; k++)
{
    for (j = 1; j <= GeometrySize; j++)
    {
        for (i = 1; i <= GeometrySize; i++)
        {
            printf(" %i",Geometry[i][j][k]);
        }
        printf("\n");
    }
    printf("\n\n\n");
}

ActualPorosity = 0.0;
LSMFract = 0.0;
YSZFract = 0.0;
for (k = 1; k <= NumZCells; k++)
{
    for (i = 1; i <= NumXCells; i++)
    {
        for (j = 1; j <= NumYCells; j++)
        {
            if (Geometry[i][j][k] == 1)
            {
                ActualPorosity = ActualPorosity + 1;
            }
            else if (Geometry[i][j][k] == 2)
            {
                LSMFract = LSMFract + 1;
            }
            else if (Geometry[i][j][k] == 3)
            {
                YSZFract = YSZFract + 1;
            }
        }
    }
}
ActualPorosity = ActualPorosity / (NumXCells * NumYCells * NumZCells);
LSMFract = LSMFract / (NumXCells * NumYCells * NumZCells);
YSZFract = YSZFract / (NumXCells * NumYCells * NumZCells);

// Marking Accessible Cells
//*****

// Accessible Fluid Cells
NumCellsFluidPath = 0;
for (i=1;i<=NumXCells;i++)
{

```

```

    for (j=1;j<=NumYCells;j++)
    {
        if (Geometry[i][j][1]==1)
        {
            Geometry[i][j][1]=4; // Mark fluid cells at k=1
            NumCellsFluidPath=NumCellsFluidPath+1;
            CooCellsFluidPath[NumCellsFluidPath][0]=i;
            CooCellsFluidPath[NumCellsFluidPath][1]=j;
            CooCellsFluidPath[NumCellsFluidPath][2]=1;
        }
    }
}
// Mark all fluid cells "infected" by the k=1 cell
Count=0;
while (Count<NumCellsFluidPath)
{
    Count=Count+1;
    printf("\n 1 Count=%i",Count);

    Infection3D(CooCellsFluidPath,&NumCellsFluidPath,Geometry,

    CooCellsFluidPath[Count][0],CooCellsFluidPath[Count][1],CooCellsFluidPa
th[Count][2],1,
                &NumInfected);
}

// Accessible Electron Cells
NumCellsFluidPath=0;
for (i=1;i<=NumXCells;i++)
{
    for (j=1;j<=NumYCells;j++)
    {
        if (Geometry[i][j][1]==2)
        {
            Geometry[i][j][1]=5; // Mark electron cells at k=1
            NumCellsFluidPath=NumCellsFluidPath+1;
            CooCellsFluidPath[NumCellsFluidPath][0]=i;
            CooCellsFluidPath[NumCellsFluidPath][1]=j;
            CooCellsFluidPath[NumCellsFluidPath][2]=1;
        }
    }
}
// Mark all electron cells "infected" by the k=1 cell
Count=0;
while (Count<NumCellsFluidPath)
{
    Count=Count+1;
    printf("\n 2 Count=%i",Count);

    Infection3D(CooCellsFluidPath,&NumCellsFluidPath,Geometry,

    CooCellsFluidPath[Count][0],CooCellsFluidPath[Count][1],CooCellsFluidPa
th[Count][2],2,
                &NumInfected);
}

// Accessible Ion Cells

```

```

NumCellsFluidPath=0;
for (i=1;i<=NumXCells;i++)
{
    for (j=1;j<=NumYCells;j++)
    {
        if (Geometry[i][j][NumZCells]==3)
        {
            Geometry[i][j][NumZCells]=6; // Mark ion cells at
k=NumZCells
            NumCellsFluidPath=NumCellsFluidPath+1;
            CooCellsFluidPath[NumCellsFluidPath][0]=i;
            CooCellsFluidPath[NumCellsFluidPath][1]=j;
            CooCellsFluidPath[NumCellsFluidPath][2]=NumZCells;
        }
    }
}
// Mark all ion cells "infected" by the k=NumZCells cell
Count=0;
while (Count<NumCellsFluidPath)
{
    Count=Count+1;
    printf("\n 2 Count=%i",Count);

    Infection3D(CooCellsFluidPath,&NumCellsFluidPath,Geometry,
CooCellsFluidPath[Count][0],CooCellsFluidPath[Count][1],CooCellsFluidPa
th[Count][2],3,
    &NumInfected);
}

// Search for Three Phase Boundaries
//*****

int FluidFlag,ElectronFlag,IonFlag;
int iThreePhase,jThreePhase,kThreePhase;
int iEle,jEle,kEle;
int iIon,jIon,kIon;

// Frame oriented along x(i)
for (k=1;k<=(NumZCells-1);k++)
{
    for (i=1;i<=(NumXCells);i++)
    {
        for (j=1;j<=(NumYCells-1);j++)
        {
            FluidFlag = 0;
            ElectronFlag = 0;
            IonFlag = 0;

            for (kn=0;kn<=1;kn++)
            {
                for (jn=0;jn<=1;jn++)
                {
                    if (Geometry[i+0][j+jn][k+kn]==4)
                    {
                        FluidFlag = 1;
                    }
                }
            }
        }
    }
}

```

```

        iThreePhase = i + 0;
        jThreePhase = j + jn;
        kThreePhase = k + kn;
    }
    if (Geometry[i+0][j+jn][k+kn]==5)
    {
        ElectronFlag = 1;
        iEle = i + 0;
        jEle = j + jn;
        kEle = k + kn;
    }
    if (Geometry[i+0][j+jn][k+kn]==6)
    {
        IonFlag = 1;
        iIon = i + 0;
        jIon = j + jn;
        kIon = k + kn;
    }
}
}

if ((FluidFlag==1) & (ElectronFlag==1) &
(IonFlag==1))
{
    Geometry[iThreePhase][jThreePhase][kThreePhase]
= 9; // Three Phase Boundaries
}
}
}

// Frame oriented along y(j)
for (k=1;k<=(NumZCells-1);k++)
{
    for (i=1;i<=(NumXCells-1);i++)
    {
        for (j=1;j<=(NumYCells);j++)
        {
            FluidFlag = 0;
            ElectronFlag = 0;
            IonFlag = 0;

            for (kn=0;kn<=1;kn++)
            {
                for (in=0;in<=1;in++)
                {
                    if (Geometry[i+in][j+0][k+kn]==4) //
                    fluid cell
                    {
                        FluidFlag = 1;
                        iThreePhase = i + in;
                        jThreePhase = j + 0;
                        kThreePhase = k + kn;
                    }
                    if (Geometry[i+in][j+0][k+kn]==5) //
                    electron cell
                    {

```



```

printf("\n Mark active surfaces");

for (k=1;k<=(NumZCells);k++)
{
    for (i=1;i<=(NumXCells);i++)
    {
        for (j=1;j<=(NumYCells);j++)
        {
            for (m=1;m<=18;m++)
            {
                TargetI = i+c[m][0];
                TargetJ = j+c[m][1];
                TargetK = k+c[m][2];

                if ( (TargetI>=1) & (TargetI<=NumXCells)
&
                    (TargetJ>=1) & (TargetJ<=NumYCells)
&
                    (TargetK>=1) & (TargetK<=NumZCells)
)
                {
                    if
(Geometry[TargetI][TargetJ][TargetK]==5) // electron cells
                    {
                        ActiveAreaFlag=1;
                    }

                    if
(Geometry[TargetI][TargetJ][TargetK]==9) //TPB
                    {
                        ThreePhaseFlag=1;
                    }
                }

                if (ActiveAreaFlag==1)
                {
                    Geometry[i][j][k]=7; // contact
                }
                if ((ThreePhaseFlag==1) & (ActiveAreaFlag==1))
                {
                    Geometry[i][j][k]=-8; // active surface
                }
            }
        }
    }
}

printf("\n Count active surfaces");

int NumActiveSurfaces=0;
int ActiveSurfaceSize[NumXCells*NumYCells*NumZCells+1];
for (k=1;k<=(NumZCells);k++)
{

```

```

for (i=1;i<=(NumXCells);i++)
{
    for (j=1;j<=(NumYCells);j++)
    {
        if (Geometry[i][j][k]==-8)
        {
            Geometry[i][j][k]=8;
            CooCellsFluidPath[1][0]=i;
            CooCellsFluidPath[1][1]=j;
            CooCellsFluidPath[1][2]=k;
            NumCellsFluidPath=1;

            NumActiveSurfaces = NumActiveSurfaces+1;
            Count=0;
            while (Count<NumCellsFluidPath)
            {
                Count=Count+1;

                Infection3D(CooCellsFluidPath,&NumCellsFluidPath,Geometry,

                CooCellsFluidPath[Count][0],CooCellsFluidPath[Count][1],CooCellsFluidPa
th[Count][2],7,

                &NumInfected);
            }
            NumCellsFluidPath=1;
            Count=0;
            while (Count<NumCellsFluidPath)
            {
                Count=Count+1;

                Infection3D(CooCellsFluidPath,&NumCellsFluidPath,Geometry,

                CooCellsFluidPath[Count][0],CooCellsFluidPath[Count][1],CooCellsFluidPa
th[Count][2],-8,

                &NumInfected);
            }
            ActiveSurfaceSize[NumActiveSurfaces] =
            ActiveSurfaceSize[NumActiveSurfaces]+NumCellsFluidPath-1;
        }
    }
}

// Print Geometry
//*****
/*
for (k=1;k<=NumZCells;k++)
{
    printf("\n k=%i",k);
    for (i=1;i<=NumXCells;i++)
    {
        printf("\n");
    }
}

```



```

        for (j=1;j<=NumYCells;j++)
        {
            printf(" %i ",Geometry[i][j][k]);
        }
    }
}
*/

// Print to File
//*****

ofstream GeometryOutput("SOFCGeometry_48x48x48_30pore.csv",ios::out);
if( !GeometryOutput )
{
    cerr << "\n Error saving" << endl;
}

// Write matrix
for (k=1;k<=NumZCells;k++)
{
    for (i=1;i<=NumXCells;i++)
    {
        for (j=1;j<=NumYCells;j++)
        {
            GeometryOutput.setf(ios::fixed);
            GeometryOutput << Geometry[i][j][k];

            if (j != NumYCells)
            {
                GeometryOutput << ",";
            }
        }
        GeometryOutput << endl;
    }
    GeometryOutput << "*" << endl;
}
GeometryOutput.close();

ofstream GeometryOutput2("SOFCGeometry_48x48x48_30pore.csv",ios::out);
if( !GeometryOutput2 )
{
    cerr << "\n Error saving" << endl;
}

// Write matrix
for (k=1;k<=NumZCells;k++)
{
    for (i=1;i<=NumXCells;i++)
    {
        for (j=1;j<=NumYCells;j++)
        {
            if (k==NumZCells)
            {
                Geometry[i][j][k]=0;
            }
            if (Geometry[i][j][k]==2)
            {

```

```

        Geometry[i][j][k]=0;
    }
    if (Geometry[i][j][k]==3)
    {
        Geometry[i][j][k]=0;
    }
    if (Geometry[i][j][k]==4)
    {
        Geometry[i][j][k]=1;
    }
    if (Geometry[i][j][k]==5)
    {
        Geometry[i][j][k]=0;
    }
    if (Geometry[i][j][k]==6)
    {
        Geometry[i][j][k]=0;
    }
    if (Geometry[i][j][k]==7)
    {
        Geometry[i][j][k]=0;
    }
    if (Geometry[i][j][k]==8)
    {
        Geometry[i][j][k]=0;
    }
    if ( (k==1) & ((Geometry[i][j][k]==1) |
(Geometry[i][j][k]==9)) )
    {
        Geometry[i][j][k]=8;
    }
    //if (Geometry[i][j][k]==9)
    //{
    //    Geometry[i][j][k]=1;
    //}

    GeometryOutput2.setf(ios::fixed);
    GeometryOutput2 << Geometry[i][j][k];

    if (j != NumYCells)
    {
        GeometryOutput2 << ", ";
    }
    }
    GeometryOutput2 << endl;
}
GeometryOutput2 << "*" << endl;
}
GeometryOutput2.close();

//*****

int MaxSize=0;
int MinSize=1000;

printf("\n");
for (i=1;i<=NumActiveSurfaces;i++)

```

```

{
    if (ActiveSurfaceSize[i]>MaxSize)
    {
        MaxSize = ActiveSurfaceSize[i];
    }
    if (ActiveSurfaceSize[i]<MinSize)
    {
        MinSize = ActiveSurfaceSize[i];
    }
    //printf("\n %i",ActiveSurfaceSize[i]);
}
printf("\n");

printf("\n Actual porosity = %1.4f",ActualPorosity);
printf("\n LSMFract = %1.4f",LSMFract);
printf("\n YSZFract = %1.4f",YSZFract);
printf("\n Number of three phase boundaries (points) =
%i",NumThreePhaseBoundary);
printf("\n Number of three phase boundaries (surfaces) =
%i",NumActiveSurfaces);
printf("\n Active surfaces MaxSize = %i",MaxSize);
printf("\n Active surfaces MinSize = %i",MinSize);
printf("\n");

printf("\n Done !!\n");
cout << "Enter a number to quit";
cin >> quit;
return 0;
}

```

Infection3D.cpp

```
#include "Stdafx.h"

int Infection3D(int CooCellsFluidPath[NumXCells*NumYCells*NumZCells+1][3],int
*NumCellsFluidPath,
               int
Geometry[NumXCells+1][NumYCells+1][NumZCells+1],int i,int j,int k,
               int Type,int *NumInfected)
{
    int m;
    int TargetI,TargetJ,TargetK;

    int c[19][3]={0,0,0}, // 0
                 { 1, 0, 0}, // 1
                 { 0, 1, 0}, // 2
                 {-1, 0, 0}, // 3
                 { 0,-1, 0}, // 4
                 { 0, 0, 1}, // 5
                 { 0, 0,-1}, // 6
                 { 1, 1, 0}, // 7
                 {-1, 1, 0}, // 8
                 {-1,-1, 0}, // 9
                 { 1,-1, 0}, // 10
                 { 1, 0, 1}, // 11
                 { 0, 1, 1}, // 12
                 {-1, 0, 1}, // 13
                 { 0,-1, 1}, // 14
                 { 1, 0,-1}, // 15
                 { 0, 1,-1}, // 16
                 {-1, 0,-1}, // 17
                 { 0,-1,-1}}; // 18

    int NumI = NumXCells;
    int NumJ = NumYCells;
    int NumK = NumZCells;

    *NumInfected = 0;
    for (m=1;m<=6;m++)
    {
        TargetI = i+c[m][0];
        TargetJ = j+c[m][1];
        TargetK = k+c[m][2];

        if ((TargetI>=1) & (TargetI<=NumI) & (TargetJ>=1) & (TargetJ<=NumJ)
& (TargetK>=1) & (TargetK<=NumK))
        {
            if (Geometry[TargetI][TargetJ][TargetK]==Type)
            {
                Geometry[TargetI][TargetJ][TargetK] =
Geometry[i][j][k];

                *NumCellsFluidPath = *NumCellsFluidPath+1;
                *NumInfected = *NumInfected+1;
            }
        }
    }
}
```

```

        CooCellsFluidPath[*NumCellsFluidPath][0]=TargetI;
        CooCellsFluidPath[*NumCellsFluidPath][1]=TargetJ;
        CooCellsFluidPath[*NumCellsFluidPath][2]=TargetK;
    }
}

// success
return 0;
}

```

round.cpp

```

#include "Stdafx.h"

double round(double value)
{
    double fractpart, intpart;


    fractpart=modf(value,&intpart);

    if (value>=0)
    {
        if (fabs(fractpart)>=0.5)
        {
            return ceil(intpart+1);
        }
        else
        {
            return ceil(intpart);
        }
    }
    else
    {
        if (fabs(fractpart)>=0.5)
        {
            return floor(intpart-1);
        }
        else
        {
            return floor(intpart);
        }
    }
}
}

```

Appendix B: ASC1 SOFC Data Sheets⁷

Product Information



InDEC B.V.

ASC1

Anode Supported Cell, type 1

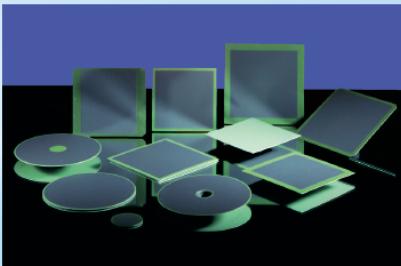
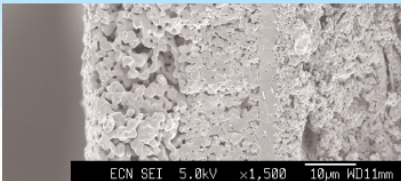
Anode Support	Porous NiO/8YSZ	520 – 600 μm
Anode	Porous NiO/8YSZ	5 – 10 μm
Electrolyte	Dense 8YSZ	4 – 6 μm
Cathode	Porous 8YSZ/LSM-LSM double layer	30 – 40 μm

Typical geometrical qualification

Maximum Size	200 × 200 mm
Lateral Size Tolerance	± 0.2 mm
Total cell thickness	0.55 – 0.63 mm

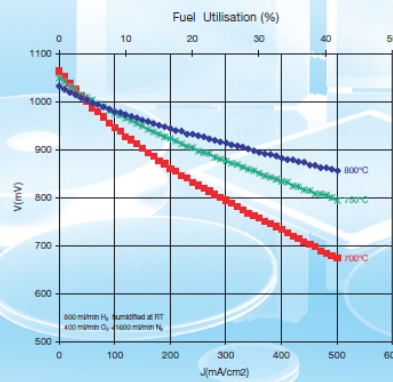
Features

- SOFC cell suitable for intermediate temperature operation, 750 – 850 °C

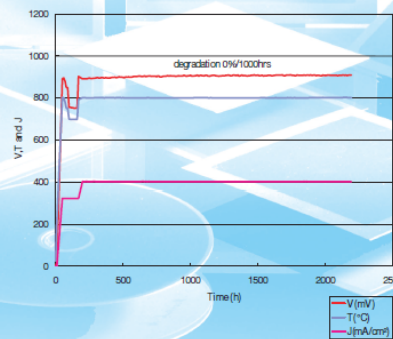



ECN SEI 5.0kV ×1,500 10μm WD11mm

Typical electrochemical performance*



800 ml. min⁻¹ Hydrogen humidified at R.T.
2000 ml. min⁻¹ Air



***Operating Conditions:**

- Cell active area 100 cm²
- Ceramic single cell housing:
Anode current collector Ni-grid
Cathode current collector Pt-grid
- Non sealed cell housing

ASR (Ω.cm²)

500 hrs	0.29
1000 hrs	0.29
1500 hrs	0.28
2000 hrs	0.27

Abbreviations

ASC1	Anode Supported Cell, type 1	ASR	Area Specific Resistance
8YSZ	8 mol% Y ₂ O ₃ doped ZrO ₂		
LSM	Lanthanum Strontium Manganese Oxide		

⁷ Reprinted from http://www.hcstarck.com/index.php?page_id=2614, ASC 1 Product Information, with permission from H. C. Starck.



The conditions of your use and application of our products, technical assistance and information (whether verbal, written or by way of production evaluations), including any suggested formulations and recommendations, are beyond our control. Therefore, it is imperative that you test our products, technical assistance and information to determine to your own satisfaction whether they are suitable for your intended uses and applications. This application-specific analysis at least must include testing to determine suitability from a technical as well as health, safety, and environmental standpoint. Such testing has not necessarily been done by H.C. Starck. All information is given without warranty or guarantee. It is expressly understood and agreed that the customer assumes and hereby expressly releases H.C. Starck from all liability, in tort, contract or otherwise, incurred in connection with the use of our products, technical assistance and information. Any statement or recommendation not contained herein is unauthorized and shall not bind H.C. Starck. Nothing herein shall be construed as a recommendation to use any product in conflict with patents covering any material or its use. No license is implied or in fact granted under the claims of any patent. Properties of the products referred to herein shall as general rule not be classed as information on the properties of the item for sale. In case of order please refer to issue number of the respective product data sheet. All deliveries are based on the latest issue of the product data sheet and the latest version of our General Conditions of Sale and Delivery.

Europe

H.C. Starck GmbH

Im Schleeke 78-91
38642 Goslar
Germany
Phone: +49/53 21/751-0
Fax: +49/53 21/751-6192

H.C. Starck Ceramics GmbH & Co. KG

Lorenz-Hutschenreuther-Str. 81
95100 Selb
Germany
Phone: +49/92 87/8 07-0
Fax: +49/92 87/8 07-4 77

InDEC Pilot Production B.V.

Westerduinweg 3
1755 LE Petten
Netherlands
Phone: +31/2 24/56-48 88
Fax: +31/2 24/56-86 15

North America

H.C. Starck Inc.

45 Industrial Place
Newton, MA 02461-1951
USA
Tel.: +1(617)630-58 00
Fax: +1(617)630-58 79

Asia

H.C. Starck-V TECH Ltd.

Ohmori NM Bldg.
1-1, Ohmori Kita 2-chome
Ohta-ku, Tokyo 143-0016
Japan
Tel.: +81/3/57 62/20 38
Fax: +81/3/57 62/20 21

Bayer Korea Ltd.

Division H.C. Starck
Samsung Boramae Omni Tower 395-62
Shindaebang-dong
Dongjak-ku, Seoul 156-712/Korea
Tel.: +82/2/829/66 77
Fax: +82/2/847/30 62

HEADQUARTERS:**H.C. Starck GmbH**

Im Schleeke 78-91
38642 Goslar
Germany

Tel.: +49/53 21/7 51-0
Fax: +49/53 21/7 51-61 92

www.hcstarck.com
info@hcstarck.com

A  Bayer MaterialScience Company

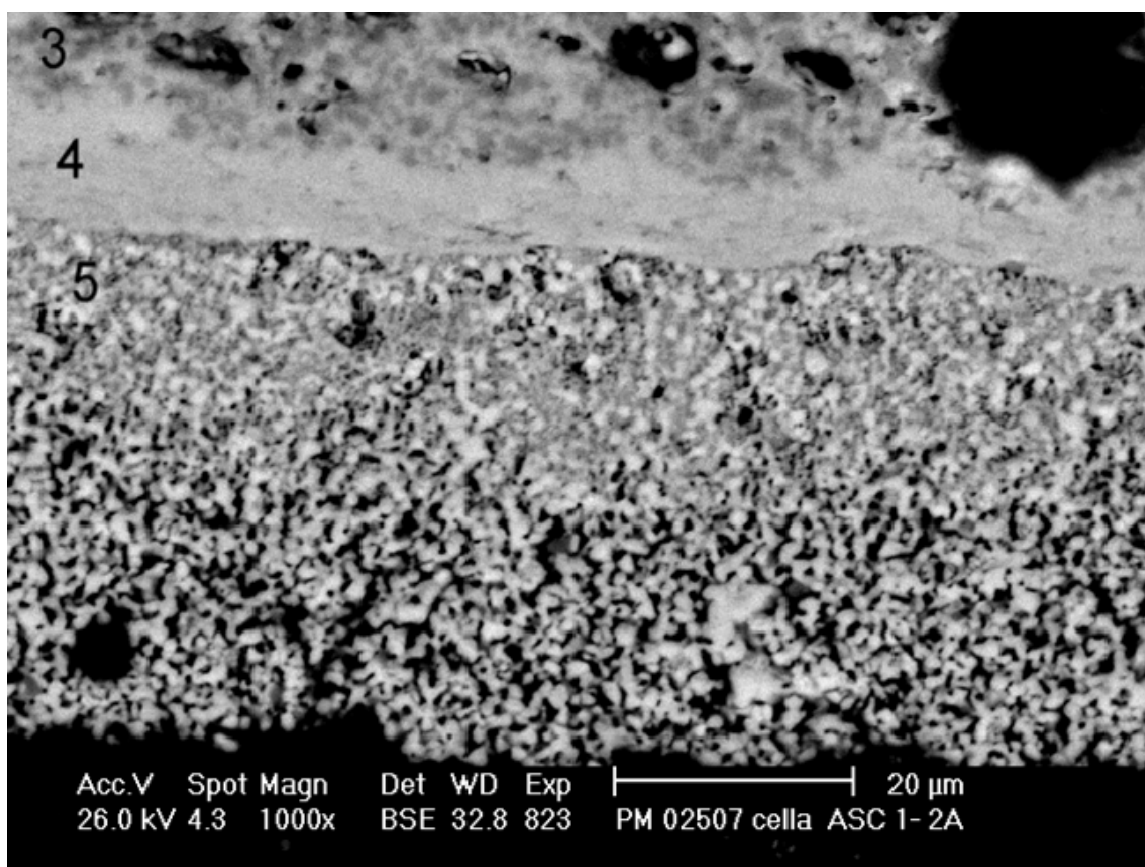
H.C. Starck 
Engineered Material Solutions

2mw 10/04

IMAGE ANALYSIS - MICROSTRUCTURAL RESULTS

IMAGE TYPE	POROSITY [%]	VOL. Electronic Phase* [%]	VOL. Ionic Phase** [%]	GRAIN SIZE Electronic Ph.* [μm]	GRAIN SIZE Ionic Ph.** [μm]	Thickness [μm]
ASC1 SEM-BSE / ANODE SUPPORT	32	34	34	0,5	0,8	520
ASC1 SEM-BSE / CATHODE ACTIVE LAYER	7	48	45	0,5	0,6	-
ASC1 SEM-BSE / CATHODE COLLECTOR LAYER	30	70	0	1	-	18

IMAGE TYPE	Weigth Electronic Phase [%]	Weigth Ionic Phase [%]
ASC1 SEM-BSE / ANODE SUPPORT	50	20
ASC1 SEM-BSE / CATHODE ACTIVE LAYER	50	50
ASC1 SEM-BSE / CATHODE COLLECTOR LAYER	100	0



SOFC cathode SEM-BSE

Appendix C: LBM Input Parameters

Contents of “input.log”

```
8
4
1
0.5
1
0
1
SOFCGeometry
```

Remove this text before running code:

Entry description:

```
-WorkingMode
-Refinement Factors
-Number of Void Fractions to analyze
-Void fraction value
-Number of pressure differences
-pressure differences (Pa)
-Number of geometries to analyze
-geometry file name
```

Appendix D: Post Processing Code

PostProcessing.m

```
clear all
close all
clc

CaseName = 'SOFCGeometry_48x48x48_07pore';
f = 0;
TotalNodes=32;
SizeX=96;
SizeY=48;
SizeZ=48;
% SizeX=64;
% SizeY=32;
% SizeZ=32;

PostProcessing_Density
% delete *Density.csv
delete *Density.mat
clear Density DensityTOT

comp='X';
% PostProcessing_Velocity
% delete(['*Velocity',comp, '.csv'])
% delete(['*Velocity',comp, '.mat'])
% clear Velocity VelocityTOT
% clear(['Velocity',comp, 'TOT'])
%
comp='Y';
% PostProcessing_Velocity
% delete(['*Velocity',comp, '.csv'])
% delete(['*Velocity',comp, '.mat'])
% clear Velocity VelocityTOT
% clear(['Velocity',comp, 'TOT'])

comp='Z';
PostProcessing_Velocity
% delete(['*Velocity',comp, '.csv'])
delete(['*Velocity',comp, '.mat'])
clear Velocity VelocityTOT
clear(['Velocity',comp, 'TOT'])
```

PostProcessing_Density.m

```
    for n=0:1:(TotalNodes-1);

FileName=[CaseName, '_t', num2str(TotalNodes), 'n', num2str(n), '_x4f', num2str(f),
'_Density.csv']

FileNameM=[CaseName, '_t', num2str(TotalNodes), 'n', num2str(n), '_x4f', num2str(f)
, '_Density.mat'];

    Density = zeros(SizeX,SizeY,SizeZ);
    for k=1:1:SizeZ;
%       fprintf(' %i',k);
        start=(k-1)*(SizeX+1)+1;
        M=csvread(FileName,start-1,1-1,[start-1,1-1,start+SizeX-2,SizeY-
1]);

        for i=1:1:SizeX;
            for j=1:1:SizeY;
                Density(i,j,k)=M(i,j);
            end
        end
    end
    save(FileNameM, 'Density');
end

nX=0;
nY=1;
nZ=1;

    for n=0:1:(TotalNodes-1);

FileNameM=[CaseName, '_t', num2str(TotalNodes), 'n', num2str(n), '_x4f', num2str(f)
, '_Density.mat'];
        load(FileNameM);

        nX=nX+1;
        if (nX>2)
            nX=1;
            nY=nY+1;
            if (nY>4)
                nY=1;
                nZ=nZ+1;
            end
        end
    end
    fprintf('\n %i %i %i %i',n,nX,nY,nZ);

    for i=1:1:SizeX;
        for j=1:1:SizeY;
            for k=1:1:SizeZ;
                iT=(nX-1)*(SizeX)+i;
                jT=(nY-1)*(SizeY)+j;
                kT=(nZ-1)*(SizeZ)+k;
                DensityTOT(iT,jT,kT)=Density(i,j,k);
            end
        end
    end
```

```
        end
    end
end
```

```
FileNameM=[CaseName, '_t', num2str(TotalNodes), '_x4f', num2str(f), '_Density_TOT.  
mat'];  
    save(FileNameM, 'DensityTOT');
```

PostProcessing_Velocity.m

```
    for n=0:1:(TotalNodes-1);

FileName=[CaseName, '_t', num2str(TotalNodes), 'n', num2str(n), '_x4f', num2str(f),
'_Velocity', comp, '.csv']

FileNameM=[CaseName, '_t', num2str(TotalNodes), 'n', num2str(n), '_x4f', num2str(f)
, '_Velocity', comp, '.mat'];

    Velocity = zeros(SizeX,SizeY,SizeZ);
    for k=1:1:SizeZ;
%       fprintf(' %i',k);
        start=(k-1)*(SizeX+1)+1;
        M=csvread(FileName,start-1,1-1,[start-1,1-1,start+SizeX-2,SizeY-
1]);

        for i=1:1:SizeX;
            for j=1:1:SizeY;
                Velocity(i,j,k)=M(i,j);
            end
        end
    end
    save(FileNameM, 'Velocity');
end

nX=0;
nY=1;
nZ=1;

    for n=0:1:(TotalNodes-1);

FileNameM=[CaseName, '_t', num2str(TotalNodes), 'n', num2str(n), '_x4f', num2str(f)
, '_Velocity', comp, '.mat'];
    load(FileNameM);

    nX=nX+1;
    if (nX>2)
        nX=1;
        nY=nY+1;
        if (nY>4)
            nY=1;
            nZ=nZ+1;
        end
    end
    fprintf('\n %i %i %i %i',n,nX,nY,nZ);

    for i=1:1:SizeX;
        for j=1:1:SizeY;
            for k=1:1:SizeZ;
                iT=(nX-1)*(SizeX)+i;
                jT=(nY-1)*(SizeY)+j;
                kT=(nZ-1)*(SizeZ)+k;
                VelocityTOT(iT,jT,kT)=Velocity(i,j,k);
            end
        end
    end
end
```

```
        end
    end
end

assignin('base', ['Velocity', comp, 'TOT'], VelocityTOT);

FileNameM=[CaseName, '_t', num2str(TotalNodes), '_x4f', num2str(f), '_Velocity', comp, 'TOT.mat'];
save(FileNameM, ['Velocity', comp, 'TOT']);
```

MeanVelEvolution.m

```
% This code takes the Output.log file from the LBM and graphs the mean
% velocity evolution over time for each of 3 fluids: H2O, O2, N2
% Note: you must convert the .log file to .xls in Excel

clc
clear
close all

M = xlsread('SOFCGeometry_48x48x48_07pore_t32_x4Output','c21:i12770');
coll_num = 3000;
MeanVel_H2O = zeros(1,coll_num);
MeanVel_O2 = zeros(1,coll_num);
MeanVel_N2 = zeros(1,coll_num);
MeanVel_CrO3 = zeros(1,coll_num);

for i = 1 : size(M,1)
    if isnan(M(i,7)) == 0 && M(i,7) ~= 0.0
        if M(i,4) == 0
            MeanVel_H2O(M(i,1)) = M(i,7);
        elseif M(i,4) == 1
            MeanVel_O2(M(i,1)) = M(i,7);
        elseif M(i,4) == 2
            MeanVel_N2(M(i,1)) = M(i,7);
        else
            MeanVel_CrO3(M(i,1)) = M(i,7);
        end
    end
    if rem(M(i,1),100) == 0 && M(i,4) == 1
        fprintf('Collision: %i\n',M(i,1))
    end
end

hold on
plot(1:length(MeanVel_H2O),MeanVel_H2O,'b')
plot(1:length(MeanVel_O2),MeanVel_O2,'r')
plot(1:length(MeanVel_N2),MeanVel_N2,'g')
plot(1:length(MeanVel_CrO3),MeanVel_CrO3,'m')
xlabel('Collision Number')
ylabel('Mean Velocity')
legend('H2O','O2','N2','CrO3')
```

Results_Analysis.m

```
% This will show the density, velocity and other interesting graphs
clc
clear
close all

lambda = 0.074826; % first relaxation constant
refine = 192-120;
x = '4';
base = 'SOFCGeometry_48x48x48_07pore';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load the density data and average over the plane
load ([base, '_t32_x', x, 'f0_Density_TOT.mat'])
Z_Size = size(DensityTOT,3) - refine;
AveDensity0 = Averaging_Func(DensityTOT);
clear DensityTOT
load ([base, '_t32_x', x, 'f1_Density_TOT.mat'])
AveDensity1 = Averaging_Func(DensityTOT);
clear DensityTOT
load ([base, '_t32_x', x, 'f2_Density_TOT.mat'])
AveDensity2 = Averaging_Func(DensityTOT);
clear DensityTOT
load ([base, '_t32_x', x, 'f3_Density_TOT.mat'])
AveDensity3 = Averaging_Func(DensityTOT);
clear DensityTOT

% resize to eliminate NaN elements
AveDensity0 = AveDensity0(str2double(x)+1:Z_Size);
AveDensity1 = AveDensity1(str2double(x)+1:Z_Size);
AveDensity2 = AveDensity2(str2double(x)+1:Z_Size);
AveDensity3 = AveDensity3(str2double(x)+1:Z_Size);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load the velocityZ data and average over the plane
load ([base, '_t32_x', x, 'f0_VelocityZTOT.mat'])
AveVelocityZ0 = Averaging_Func(VelocityZTOT);
clear VelocityZTOT
load ([base, '_t32_x', x, 'f1_VelocityZTOT.mat'])
AveVelocityZ1 = Averaging_Func(VelocityZTOT);
clear VelocityZTOT
load ([base, '_t32_x', x, 'f2_VelocityZTOT.mat'])
AveVelocityZ2 = Averaging_Func(VelocityZTOT);
clear VelocityZTOT
load ([base, '_t32_x', x, 'f3_VelocityZTOT.mat'])
AveVelocityZ3 = Averaging_Func(VelocityZTOT);
clear VelocityZTOT

% resize to eliminate NaN elements
AveVelocityZ0 = AveVelocityZ0(str2double(x)+1:Z_Size) * 1e-6;
AveVelocityZ1 = AveVelocityZ1(str2double(x)+1:Z_Size) * 1e-6;
AveVelocityZ2 = AveVelocityZ2(str2double(x)+1:Z_Size) * 1e-6;
AveVelocityZ3 = AveVelocityZ3(str2double(x)+1:Z_Size) * 1e-6;
```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fit the data with a interp_rho vel or interp_rho degree polynomial

% Velocity polynomials
p=polyfit(1:size(AveVelocityZ0,2),AveVelocityZ0,13);
InterpVelZ0=polyval(p,1:size(AveVelocityZ0,2));

p=polyfit(1:size(AveVelocityZ1,2),AveVelocityZ1,20);
InterpVelZ1=polyval(p,1:size(AveVelocityZ1,2));

p=polyfit(1:size(AveVelocityZ2,2),AveVelocityZ2,10);
InterpVelZ2=polyval(p,1:size(AveVelocityZ2,2));

p=polyfit(1:size(AveVelocityZ3,2),AveVelocityZ3,12);
InterpVelZ3=polyval(p,1:size(AveVelocityZ3,2));

InterpVelZ=InterpVelZ0+InterpVelZ1+InterpVelZ2+InterpVelZ3;

% Density Polynomials
interp_rho = 23; % fit data to degree this
p=polyfit(1:size(AveDensity0,2),AveDensity0,interp_rho);
InterpRho0=polyval(p,1:size(AveDensity0,2));
devp = (interp_rho:-1:1) .* p(1:interp_rho);
DerInterpRho0 = polyval(devp,1:size(AveDensity0,2)); % = d(InterpRho0)/dz

interp_rho = 22; % fit data to degree this
p=polyfit(1:size(AveDensity1,2),AveDensity1,interp_rho);
InterpRho1=polyval(p,1:size(AveDensity1,2));
devp = (interp_rho:-1:1) .* p(1:interp_rho);
DerInterpRho1 = polyval(devp,1:size(AveDensity1,2)); % = d(InterpRho1)/dz

interp_rho = 24; % fit data to degree this
p=polyfit(1:size(AveDensity2,2),AveDensity2,interp_rho);
InterpRho2=polyval(p,1:size(AveDensity2,2));
devp = (interp_rho:-1:1) .* p(1:interp_rho);
DerInterpRho2 = polyval(devp,1:size(AveDensity2,2)); % = d(InterpRho2)/dz

interp_rho = 9; % fit data to degree this
p=polyfit(1:size(AveDensity3,2),AveDensity3,interp_rho);
InterpRho3=polyval(p,1:size(AveDensity3,2));
devp = (interp_rho:-1:1) .* p(1:interp_rho);
DerInterpRho3 = polyval(devp,1:size(AveDensity3,2)); % = d(InterpRho3)/dz

InterpRho=InterpRho0+InterpRho1+InterpRho2+InterpRho3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot Density and Velocity Graphs

figure(12)
plot(InterpVelZ0,'b'),grid on,hold on
plot(InterpVelZ1,'r'),grid on,hold on
plot(InterpVelZ2,'g'),grid on,hold on
plot(InterpVelZ3,'m'),grid on,hold on
plot(InterpVelZ,'k'),grid on,hold on

```

```

plot(AveVelocityZ0,'bo--'),grid on,hold on
plot(AveVelocityZ1,'ro--'),grid on,hold on
plot(AveVelocityZ2,'go--'),grid on,hold on
plot(AveVelocityZ3,'mo--'),grid on,hold on
% plot(AveVelocityZ,'ko'),grid on,hold on
axis ([0 120 -1e-5 1e-5])%tight
legend('H2O','O2','N2','CrO3','Mixture')
ylabel('Average mass fluxes < \rho uz>_{\sigma}/(\rho_0 c)')
xlabel('Depth z/\delta z [-]')

figure(13)
plot(AveDensity0,'bo--'),grid on,hold on
plot(InterpRho0,'b'),grid on,hold on
axis tight
legend('H2O')
ylabel('Average density H2O < \rho>/\rho_0')
xlabel('Depth z/\delta z [-]')

figure(14)
plot(AveDensity1,'ro--'),grid on,hold on
plot(InterpRho1,'r'),grid on,hold on
axis tight
legend('O2')
ylabel('Average density O2 < \rho>/\rho_0')
xlabel('Depth z/\delta z [-]')

figure(15)
plot(AveDensity2,'go--'),grid on,hold on
plot(InterpRho2,'g'),grid on,hold on
axis tight
legend('N2')
ylabel('Average density N2 < \rho>/\rho_0')
xlabel('Depth z/\delta z [-]')

figure(16)
plot(AveDensity3,'mo--'),grid on,hold on
plot(InterpRho3,'m'),grid on,hold on
axis tight
legend('CrO3')
ylabel('Average density CrO3 < \rho>/\rho_0')
xlabel('Depth z/\delta z [-]')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Other important quantities
figure(17)
x0 = InterpRho0./InterpRho;
x1 = InterpRho1./InterpRho;
x2 = InterpRho2./InterpRho;
x3 = InterpRho3./InterpRho;
InterpRhoW0 = InterpVelZ0-x0.*InterpVelZ;
InterpRhoW1 = InterpVelZ1-x1.*InterpVelZ;
InterpRhoW2 = InterpVelZ2-x2.*InterpVelZ;
InterpRhoW3 = InterpVelZ3-x3.*InterpVelZ;
plot(InterpRhoW0,'b'),grid on,hold on
plot(InterpRhoW1,'r'),grid on,hold on
plot(InterpRhoW2,'g'),grid on,hold on

```

```

plot(InterpRhoW3, 'm'), grid on, hold on
axis tight
legend('H2O', 'O2', 'N2', 'CrO3')
ylabel('Interp mass diffusion fluxes ( $\rho_{uz} \sigma - x_{\sigma} \rho_{uz} \sigma$ ) / ( $\rho_0 c$ )')
xlabel('Depth  $z / \Delta z$  [-]')

figure(18)
DerInterpPres0 = DerInterpRho0 ./ (3/1);
DerInterpPres1 = DerInterpRho1 ./ (3/0.5630);
DerInterpPres2 = DerInterpRho2 ./ (3/0.6431);
DerInterpPres3 = DerInterpRho3 ./ (3/0.1802);
DerInterpPres =
DerInterpPres0 + DerInterpPres1 + DerInterpPres2 + DerInterpPres3;
plot(DerInterpPres0, 'b'), grid on, hold on
plot(DerInterpPres1, 'r'), grid on, hold on
plot(DerInterpPres2, 'g'), grid on, hold on
plot(DerInterpPres3, 'm'), grid on, hold on
axis tight
legend('H2O', 'O2', 'N2', 'CrO3')
ylabel('Interp partial pressure gradients  $\nabla p_{\sigma} / (c^2 \rho_0)$ ')
xlabel('Depth  $z / \Delta z$  [-]')

figure(19)
D0 = (InterpVelZ0 - x0 .* InterpVelZ) ./ (-DerInterpPres0);
D1 = (InterpVelZ1 - x1 .* InterpVelZ) ./ (-DerInterpPres1);
D2 = (InterpVelZ2 - x2 .* InterpVelZ) ./ (-DerInterpPres2);
D3 = (InterpVelZ3 - x3 .* InterpVelZ) ./ (-DerInterpPres3);
D = mean([D0; D1; D2; D3]);
plot(D0, 'b'), grid on, hold on
plot(D1, 'r'), grid on, hold on
plot(D2, 'g'), grid on, hold on
plot(D3, 'm'), grid on, hold on
% plot(D, 'k'), grid on, hold on
axis tight
legend('H2O', 'O2', 'N2', 'CrO3')%, 'Mean')
ylabel('Macroscopic diffusion coefficient  $D / (c^2 \tau)$ ')
xlabel('Depth  $z / \Delta z$  [-]')

figure(110)
x = 1:size(D, 2);
plot(lambda.* D0, 'b'), grid on, hold on
plot(lambda.* D1, 'r'), grid on, hold on
plot(lambda.* D2, 'g'), grid on, hold on
plot(lambda.* D3, 'm'), grid on, hold on
% plot(lambda.* D, 'k'), grid on, hold on
axis tight
legend('H2O', 'O2', 'N2', 'CrO3')%, 'Mean')
ylabel('Macroscopic Tortuosity ( $\tau$ )')
xlabel('Depth  $z / \Delta z$  [-]')

% figure(11)
% Reduce the data range to 1/10th : 3/4 of original
D0_red = D0; %(round((Z_Size+refine)/10):round((Z_Size+refine)*3/4));
D1_red = D1; %(round((Z_Size+refine)/10):round((Z_Size+refine)*3/4));
D2_red = D2; %(round((Z_Size+refine)/10):round((Z_Size+refine)*3/4));

```

```
D3_red = D3; %(round((Z_Size+refine)/10):round((Z_Size+refine)*3/4));
D_red = mean([D0_red;D1_red;D2_red;D3_red]);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Results Summary
```

```
fprintf('\n Molecular Diffusivity %f',1/lambda);
fprintf('\n Porous Diffusivity %0.10f',mean(D_red));
Tortuosity = lambda .* D_red;
Tortuosity0 = lambda .* D0_red;
Tortuosity1 = lambda .* D1_red;
Tortuosity2 = lambda .* D2_red;
Tortuosity3 = lambda .* D3_red;
fprintf('\n Mean Tortuosity %0f',mean(Tortuosity));
fprintf('\n STD Tortuosity %0.10f',sqrt(var((Tortuosity))));
fprintf('\n STD/Mean Tortuosity
%0.10f',sqrt(var(Tortuosity))/mean(Tortuosity));
fprintf('\n\n Tortuosity H2O %0.10f +/-
%f',mean(Tortuosity0),sqrt(var(Tortuosity0)));
fprintf('\n Tortuosity O2 %0.10f +/-
%f',mean(Tortuosity1),sqrt(var(Tortuosity1)));
fprintf('\n Tortuosity N2 %0.10f +/-
%f',mean(Tortuosity2),sqrt(var(Tortuosity2)));
fprintf('\n Tortuosity CrO3 %0.10f +/-
%f',mean(Tortuosity3),sqrt(var(Tortuosity3)));
fprintf('\n Tortuosity %0.10f +/-
%f',mean(Tortuosity),sqrt(var(Tortuosity)));
fprintf('\n\n');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Values for finding DELTA_rho values
```

```
meanRho0 = mean(AveDensity0);
meanJz0 = mean(AveVelocityZ0);
meanRho1 = mean(AveDensity1);
meanJz1 = mean(AveVelocityZ1);
meanRho2 = mean(AveDensity2);
meanJz2 = mean(AveVelocityZ2);
meanRho3 = mean(AveDensity3);
meanJz3 = mean(AveVelocityZ3);

fprintf('\n\n Surfaced Averaged values')
fprintf('\n ----- Rho Jz')
fprintf('\n H2O %10.6f %8.5e',meanRho0,meanJz0)
fprintf('\n O2 %10.6f %8.5e',meanRho1,meanJz1)
fprintf('\n N2 %10.6f %8.5e',meanRho2,meanJz2)
fprintf('\n CrO3 %10.6e %8.5e',meanRho3,meanJz3)
fprintf('\n\n');
```

Averaging_Func.m

```
function [Average] = Averaging_Func(TOT)

Average = zeros(1,size(TOT,3));
for k = 1 : size(TOT,3)
    count = 0;
    for i = 1 : size(TOT,1)
        for j = 1 : size(TOT,2)
            if TOT(i,j,k) ~= -1
                Average(k) = Average(k) + TOT(i,j,k);
                count = count + 1;
            end
        end
    end
    Average(k) = Average(k) / count;
end
```

Appendix E: Results Graphs and Contour Plots

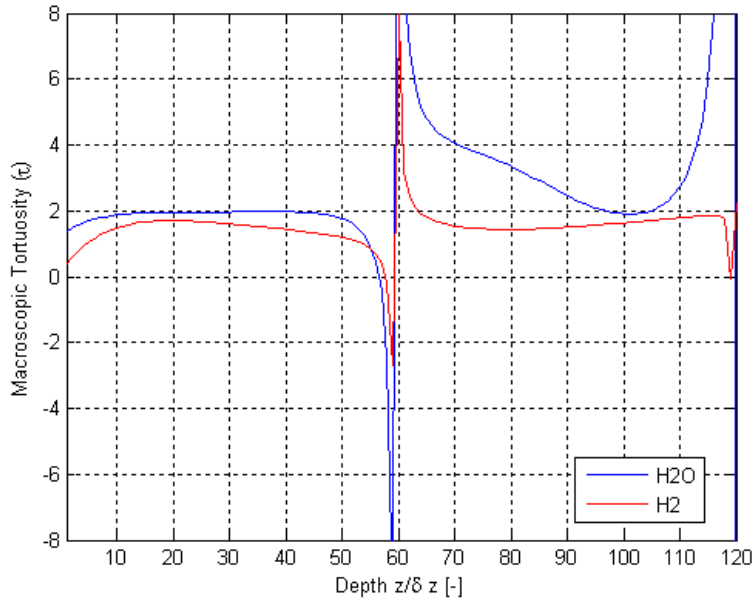


Figure E.1. Results of this thesis work for variations as a function of depth using Geometry A for the H₂O and H₂ mixture for τ .

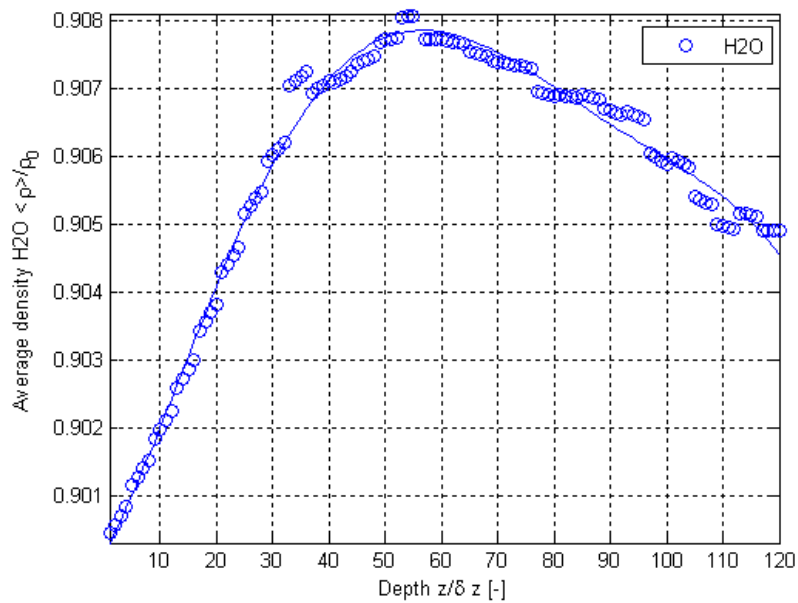


Figure E.2. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry A for the H₂O and H₂ mixture for H₂O.

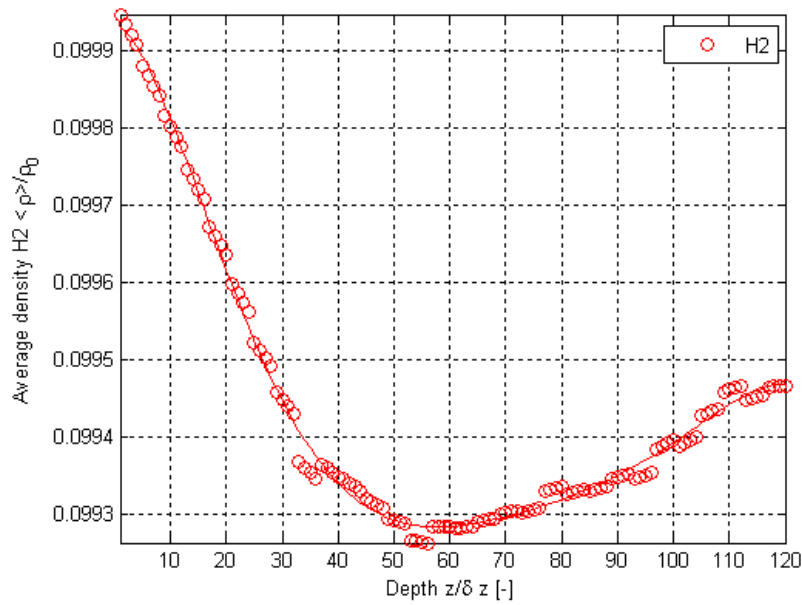


Figure E.3. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry A for the H_2O and H_2 mixture for H_2 .

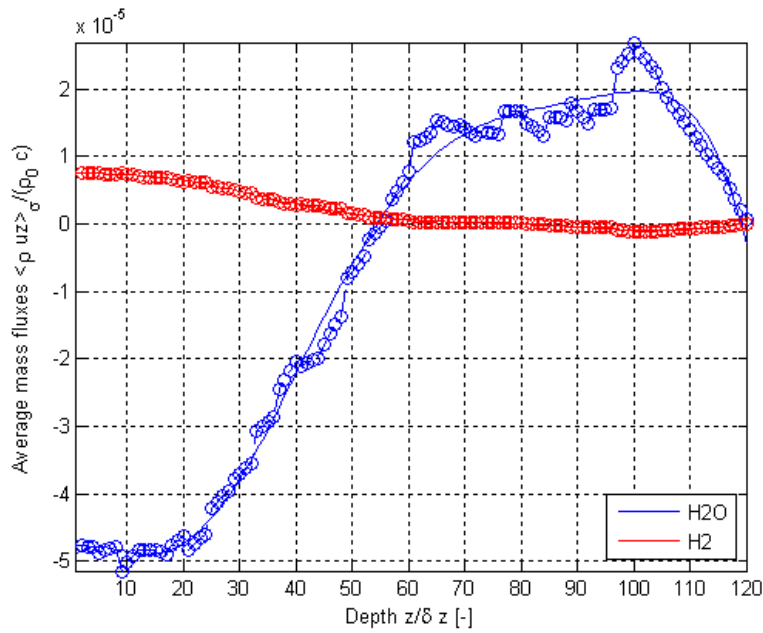


Figure E.4. Results of this thesis work for variations as a function of depth using Geometry A for the H_2O and H_2 mixture for normalized surface-averaged mass flux.

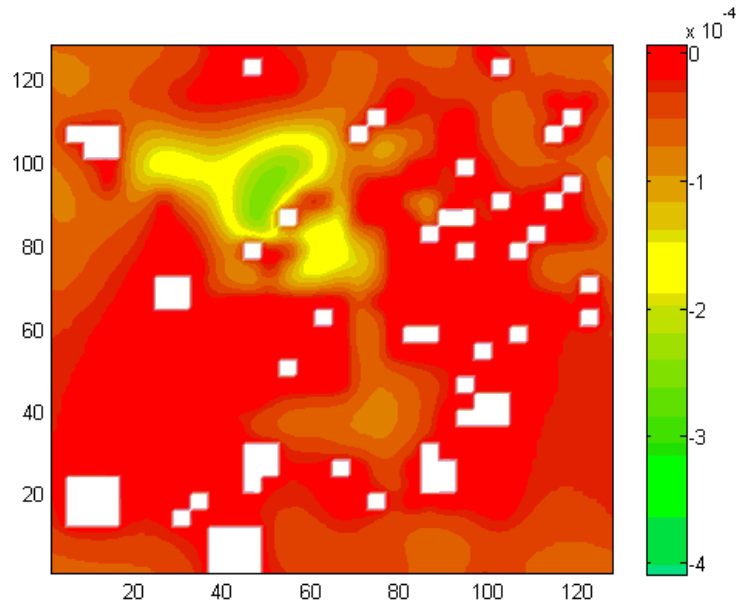


Figure E.5. Normalized mass flux contours for H₂O using Geometry A for the H₂O and H₂ mixture for the inlet/outlet plane.

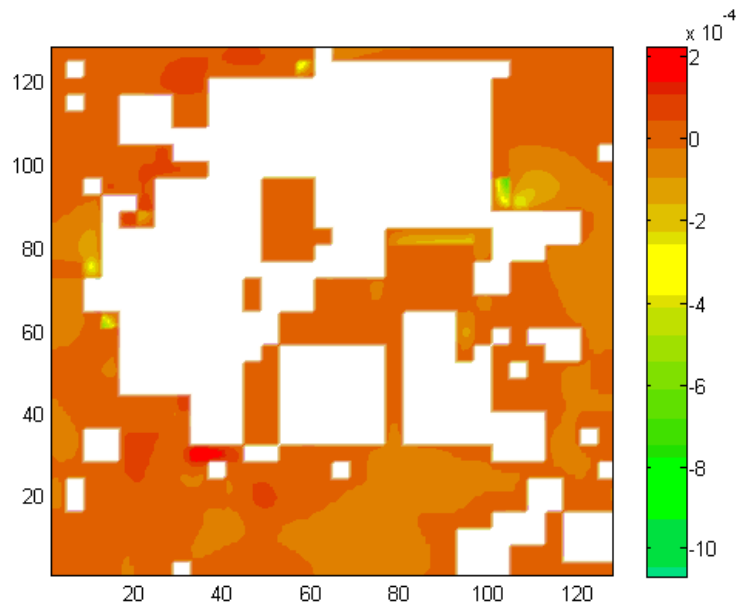


Figure E.6. Normalized mass flux contours for H₂O using Geometry A for the H₂O and H₂ mixture for an intermediate plane.

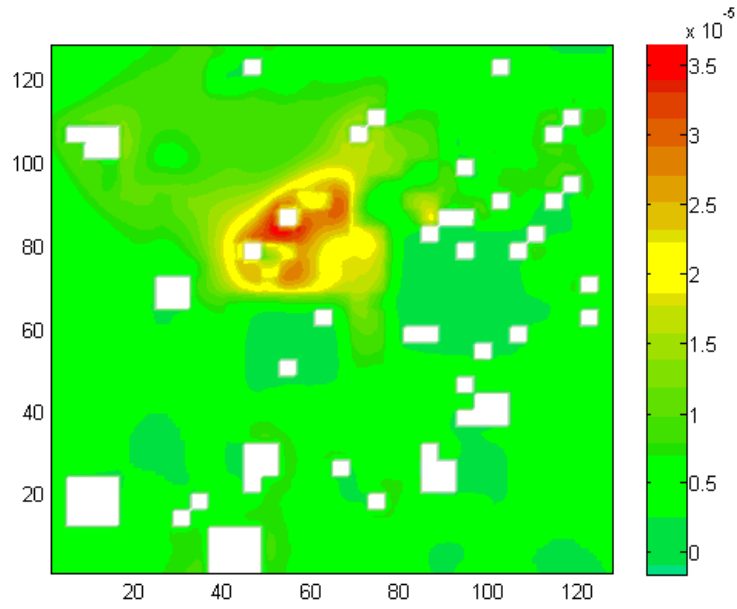


Figure E.7. Normalized mass flux contours for H₂ using Geometry A for the H₂O and H₂ mixture for the inlet/outlet plane.

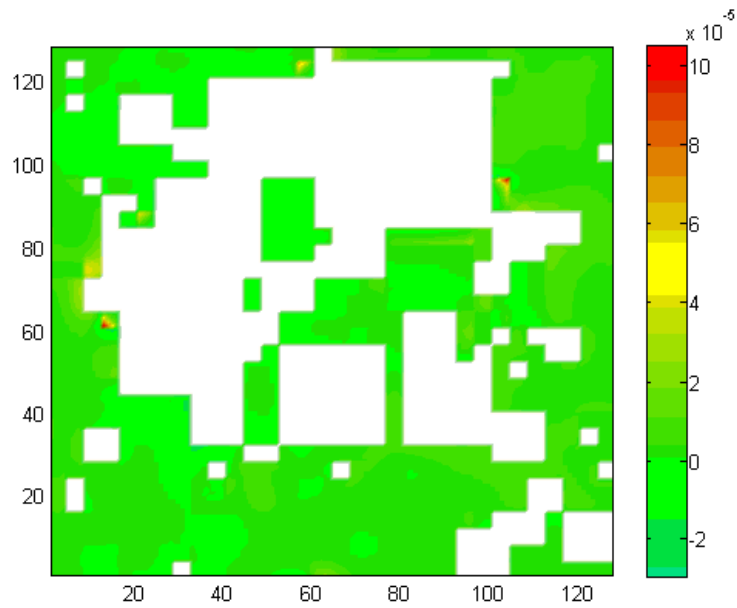


Figure E.8. Normalized mass flux contours for H₂ using Geometry A for the H₂O and H₂ mixture for an intermediate plane.

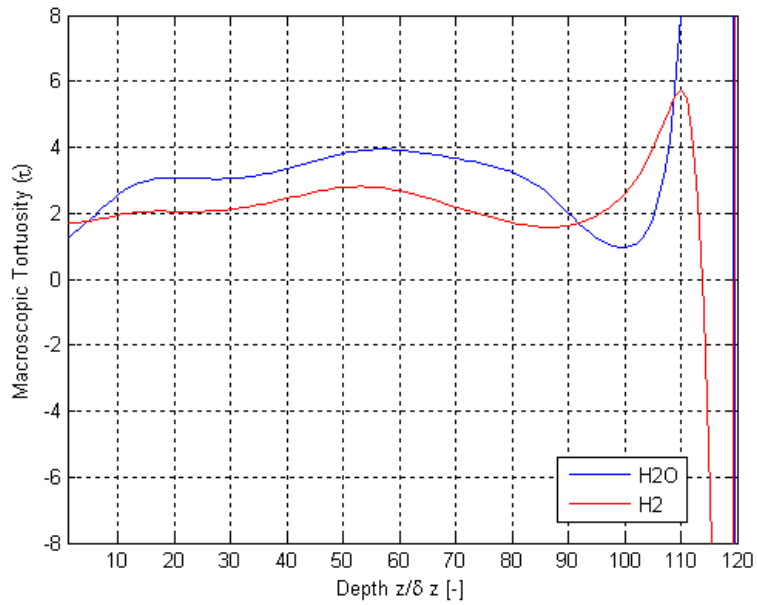


Figure E.9. Results of this thesis work for variations as a function of depth using Geometry B for the H₂O and H₂ mixture for τ .

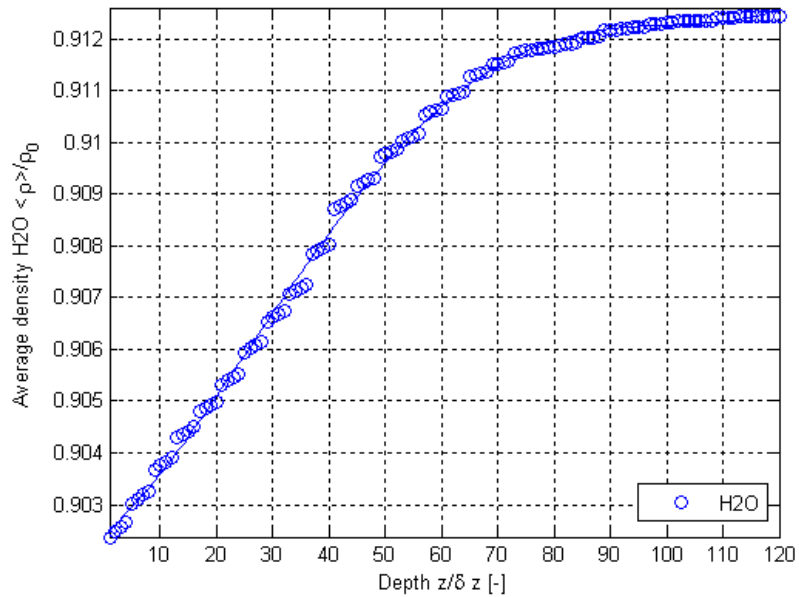


Figure E.10. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry B for the H₂O and H₂ mixture for H₂O.

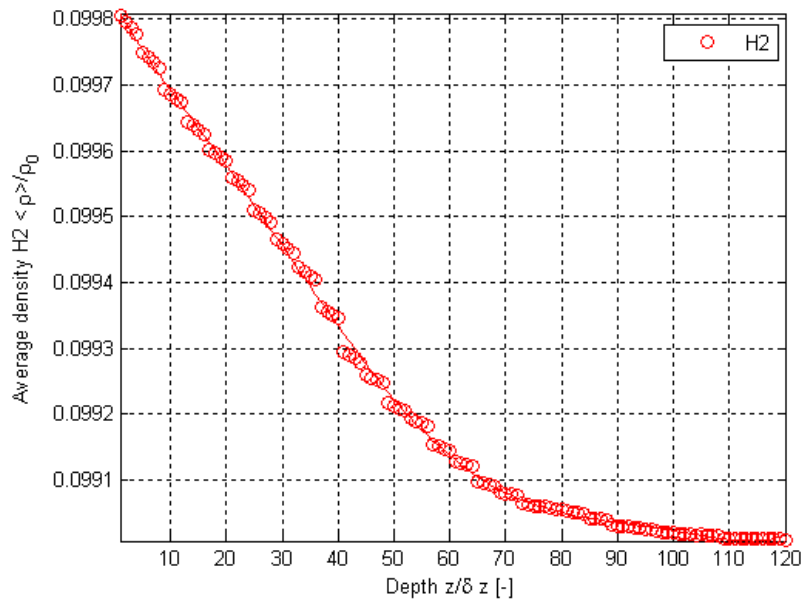


Figure E.11. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry B for the H₂O and H₂ mixture for H₂.

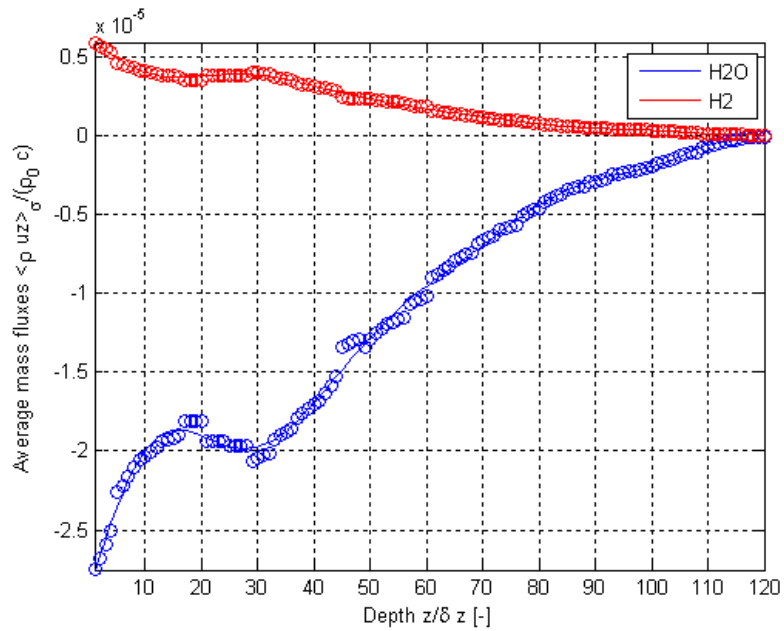


Figure E.12. Results of this thesis work for variations as a function of depth using Geometry B for the H₂O and H₂ mixture for normalized surface-averaged mass flux.

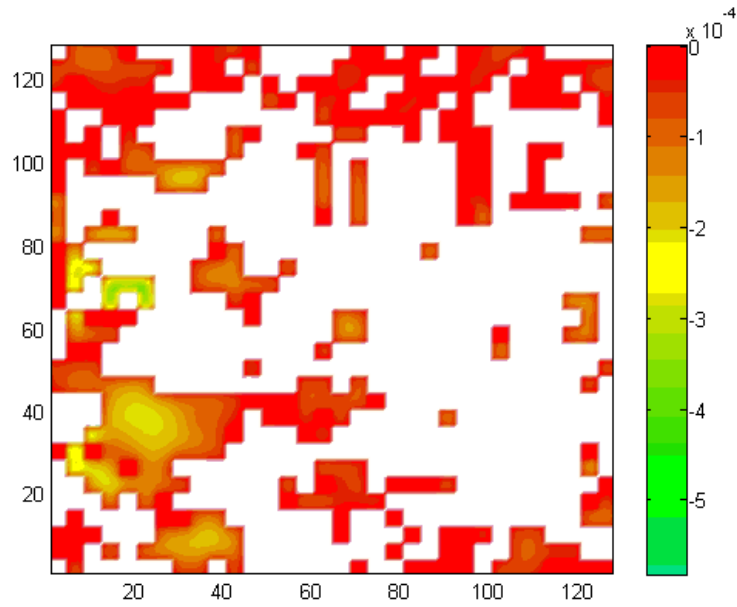


Figure E.13. Normalized mass flux contours for H₂O using Geometry B for the H₂O and H₂ mixture for the inlet/outlet plane.

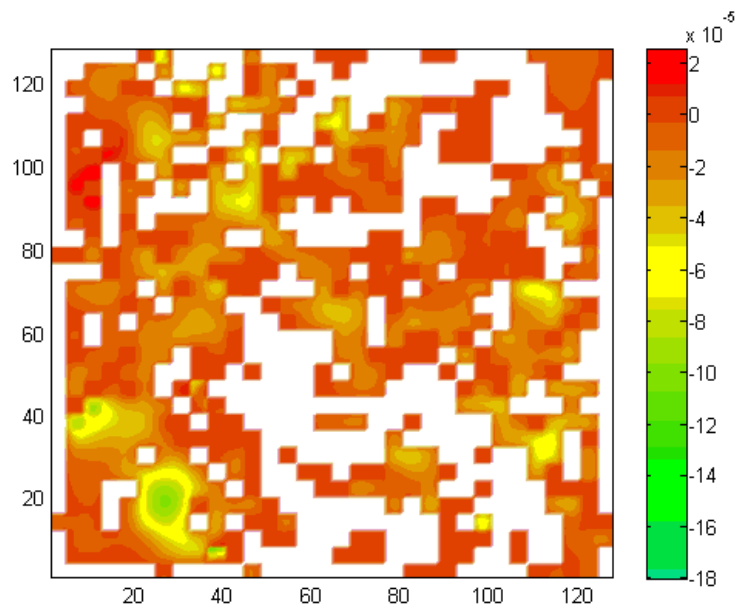


Figure E.14. Normalized mass flux contours for H₂O using Geometry B for the H₂O and H₂ mixture for an intermediate plane.

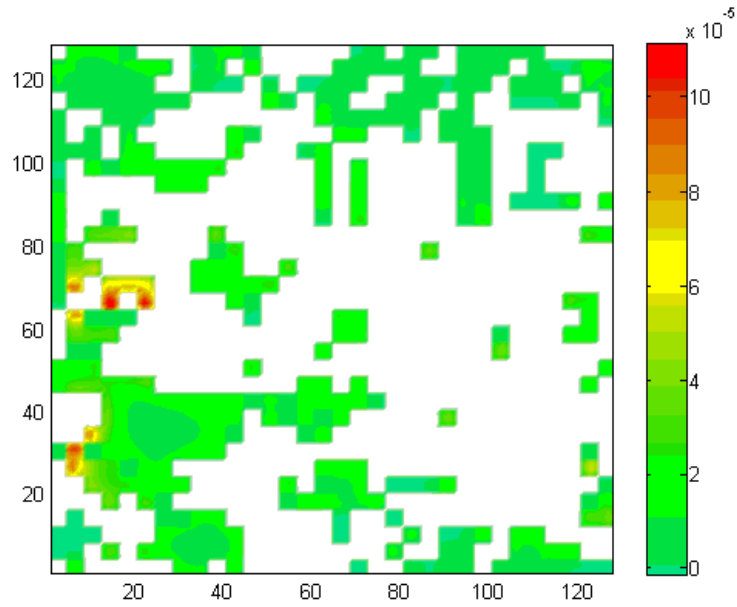


Figure E.15. Normalized mass flux contours for H₂ using Geometry B for the H₂O and H₂ mixture for the inlet/outlet plane.

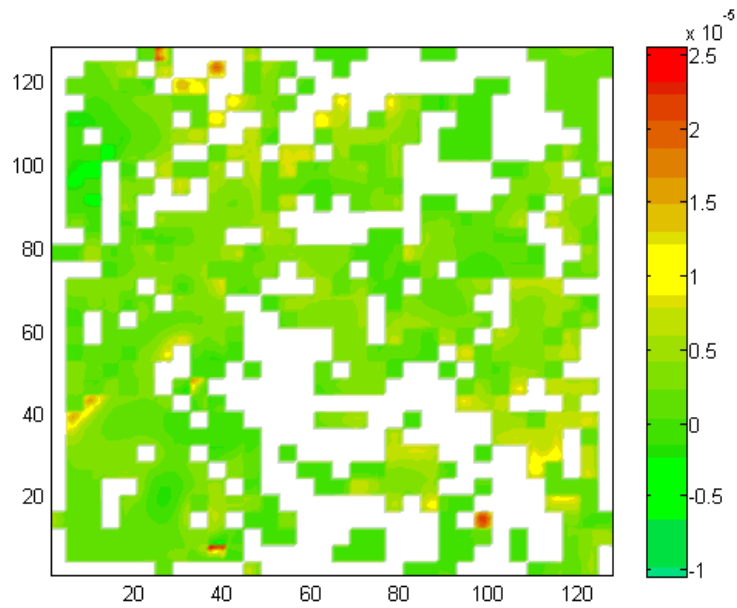


Figure E.16. Normalized mass flux contours for H₂ using Geometry B for the H₂O and H₂ mixture for an intermediate plane.

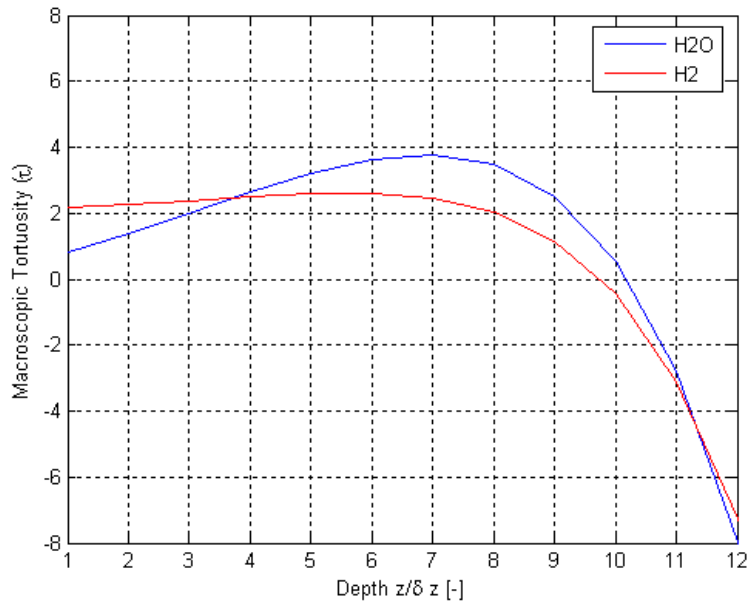


Figure E.17. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O and H₂ mixture for τ .

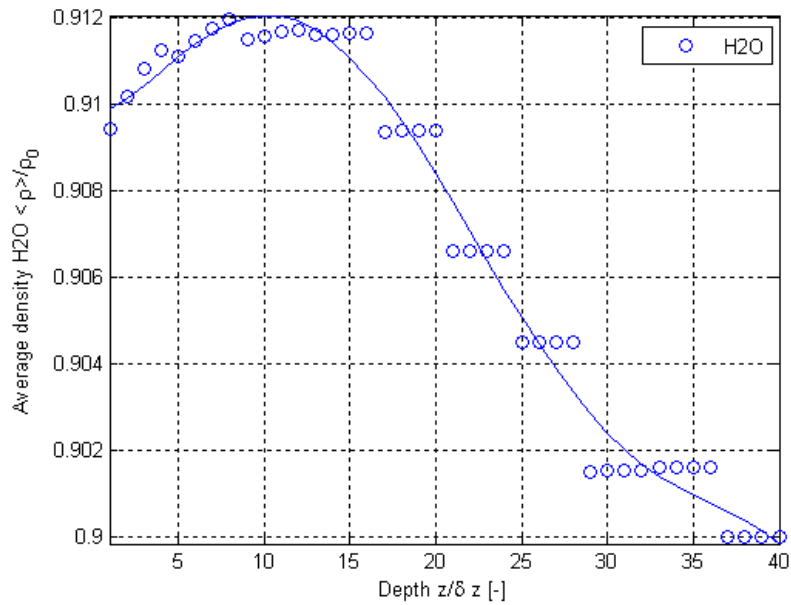


Figure E.18. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry C for the H₂O and H₂ mixture for H₂O.

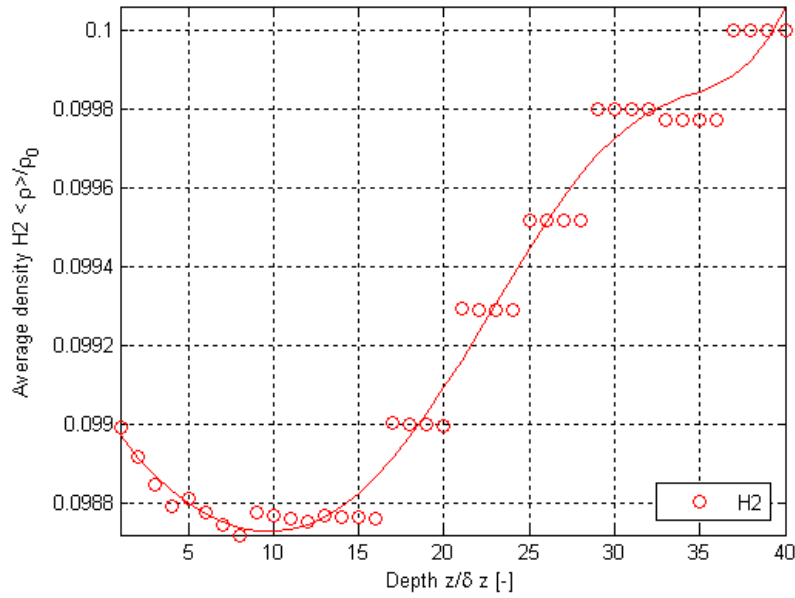


Figure E.19. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry C for the H₂O and H₂ mixture for H₂.

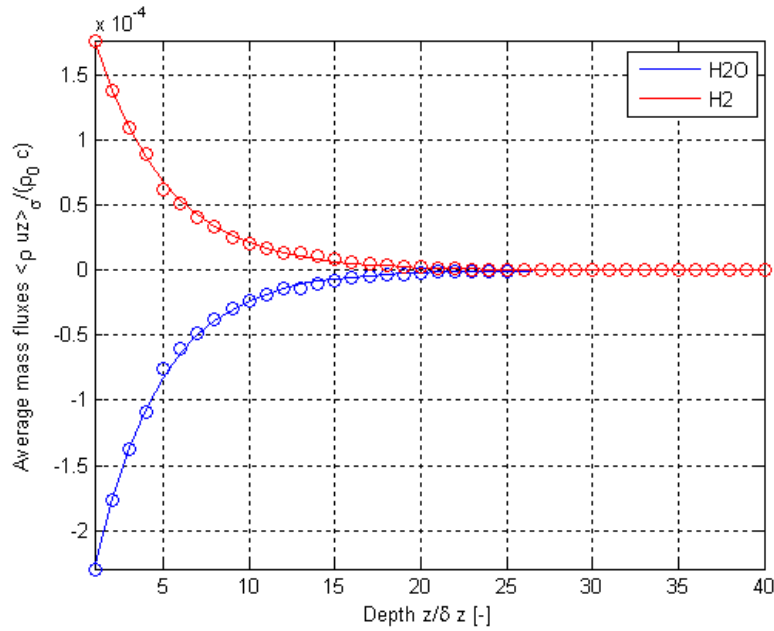


Figure E.20. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O and H₂ mixture for normalized surface-averaged mass flux.

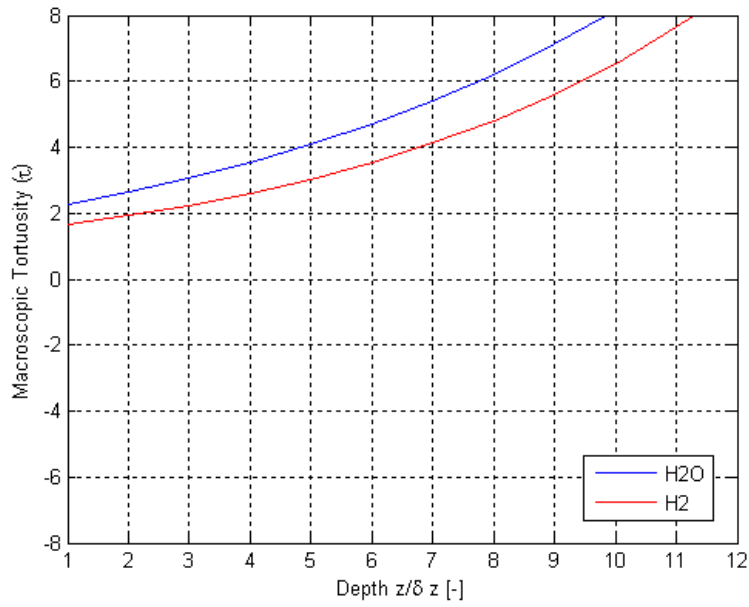


Figure E.21. Results of this thesis work for variations as a function of depth using Geometry D for the H₂O and H₂ mixture for τ .

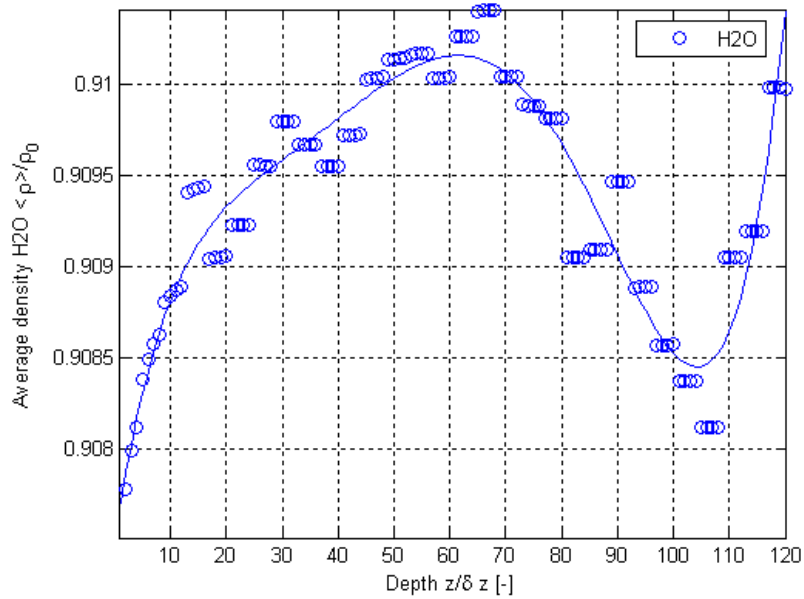


Figure E.22. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry D for the H₂O and H₂ mixture for H₂O.

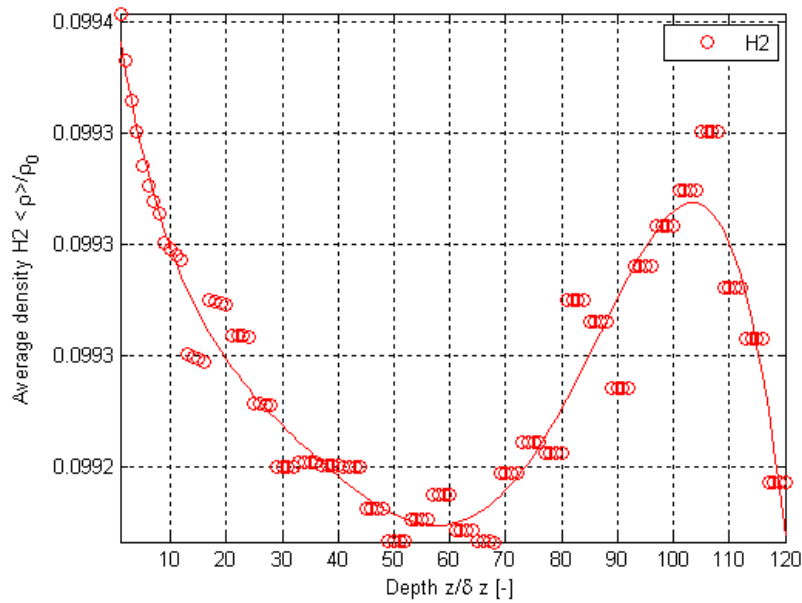


Figure E.23. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry D for the H_2O and H_2 mixture for H_2 .

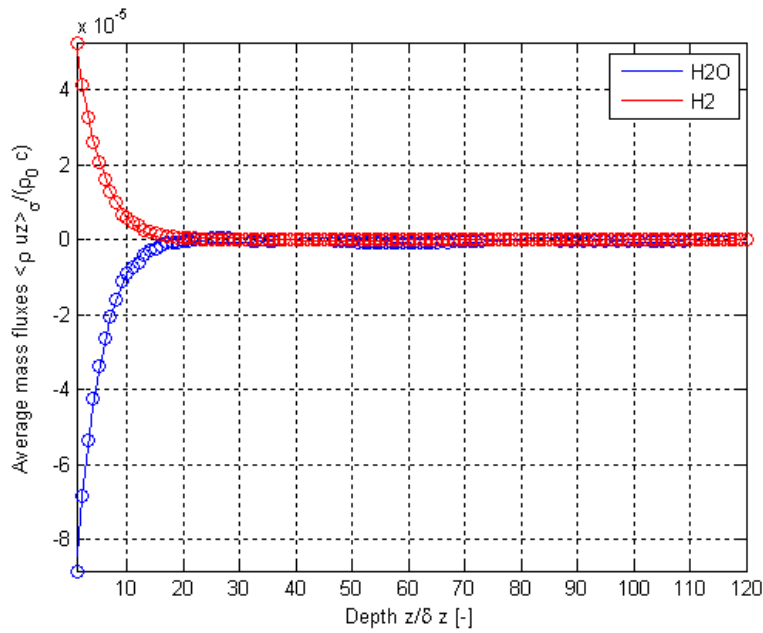


Figure E.24. Results of this thesis work for variations as a function of depth using Geometry D for the H_2O and H_2 mixture for normalized surface-averaged mass flux.

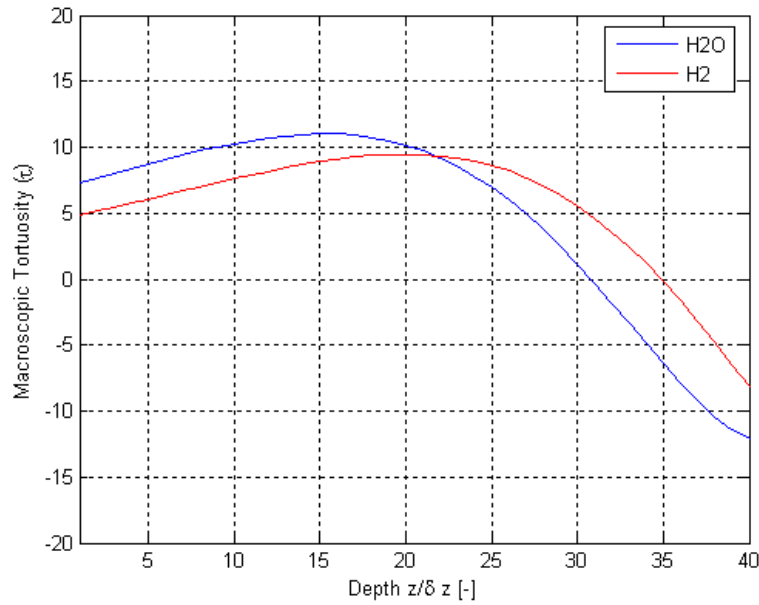


Figure E.25. Results of this thesis work for variations as a function of depth using Geometry E for the H₂O and H₂ mixture for τ .

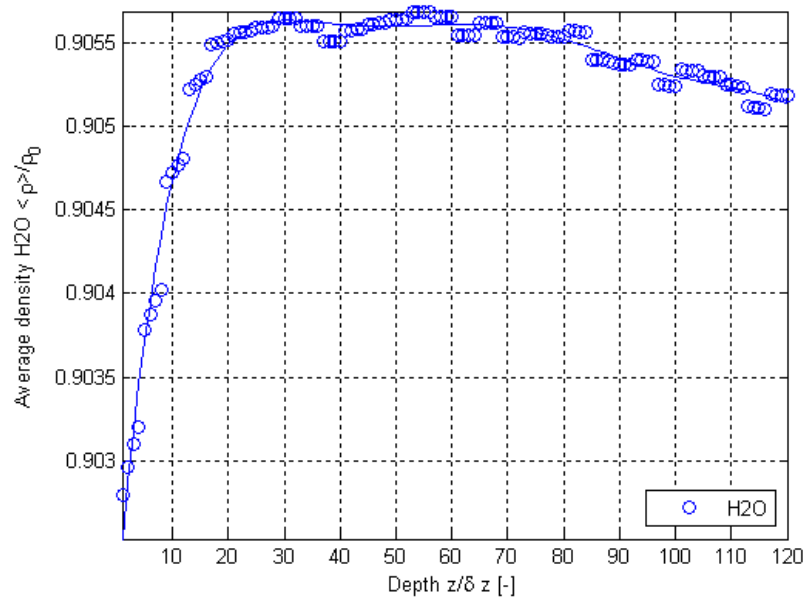


Figure E.26. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry E for the H₂O and H₂ mixture for H₂O.

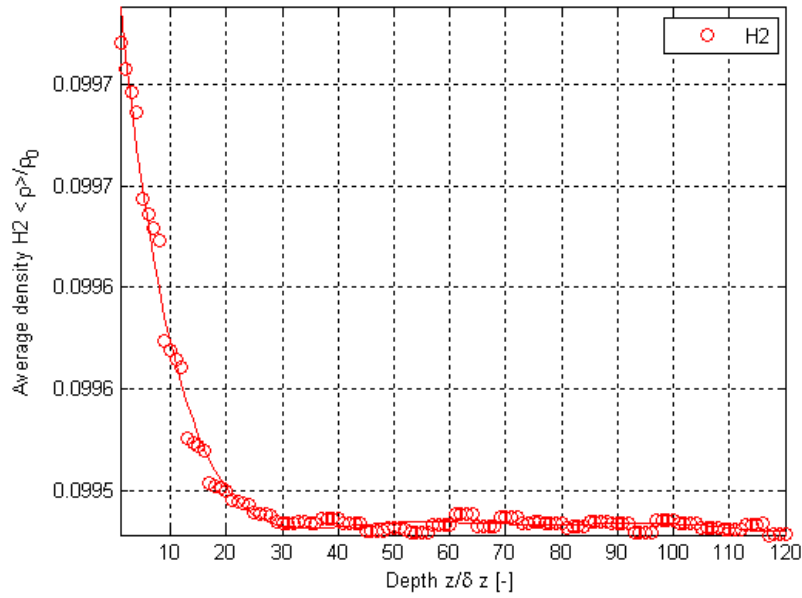


Figure E.27. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry E for the H₂O and H₂ mixture for H₂.

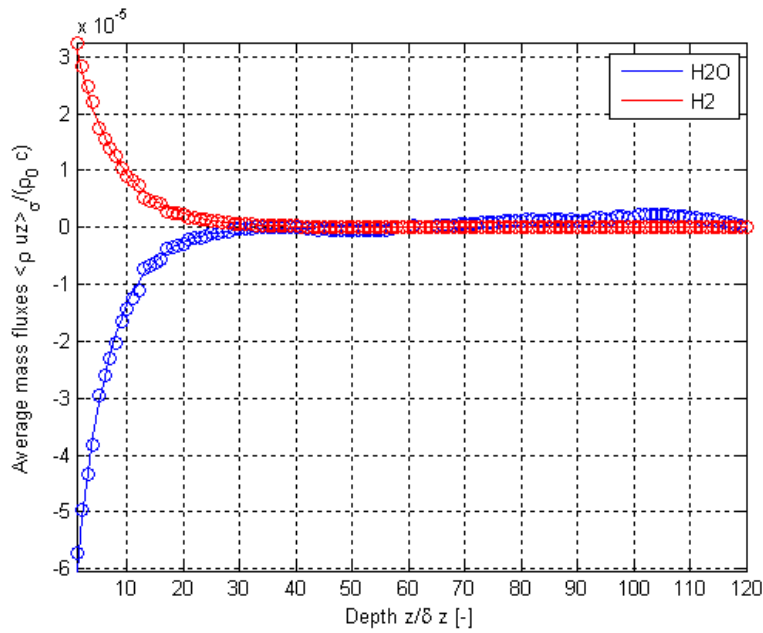


Figure E.28. Results of this thesis work for variations as a function of depth using Geometry E for the H₂O and H₂ mixture for normalized surface-averaged mass flux.

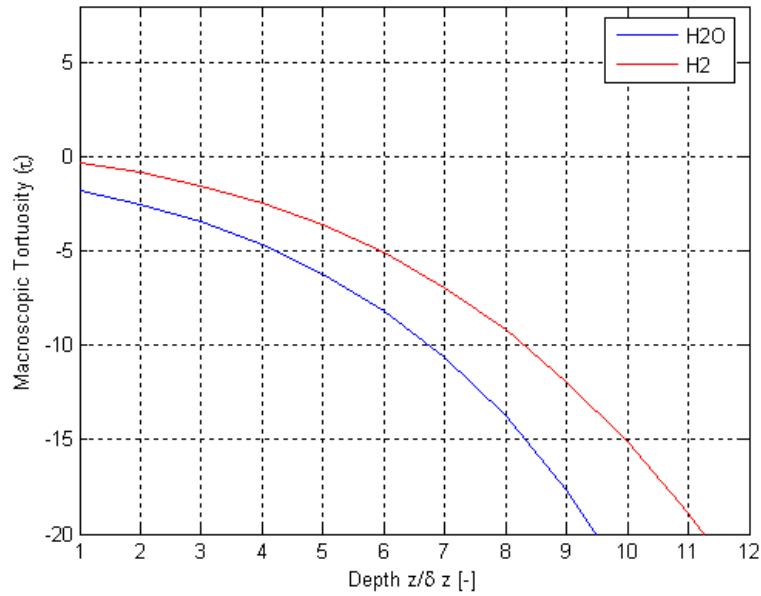


Figure E.29. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O and H₂ mixture for τ .

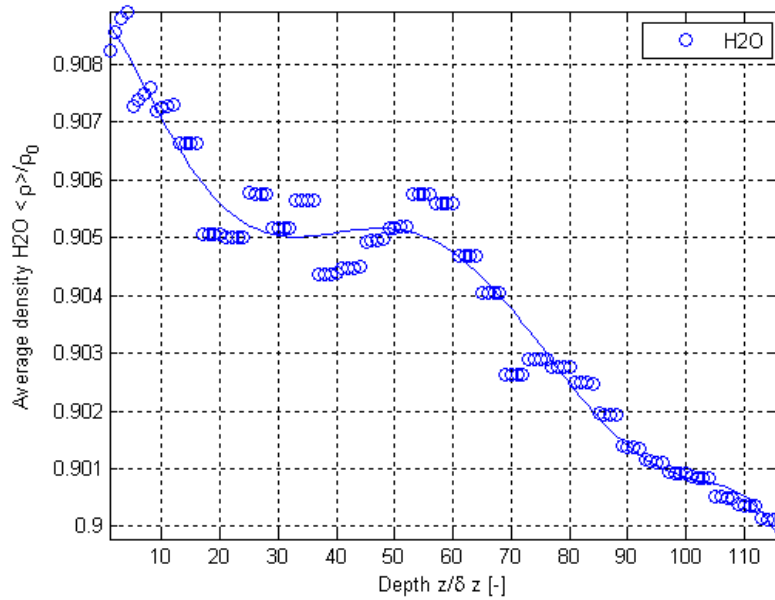


Figure E.30. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry F for the H₂O and H₂ mixture for H₂O.

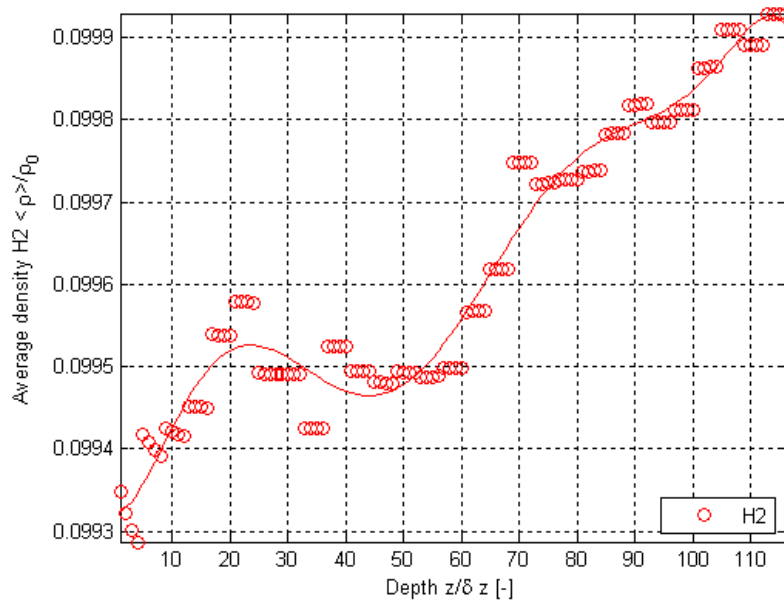


Figure E.31. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry F for the H₂O and H₂ mixture for H₂.

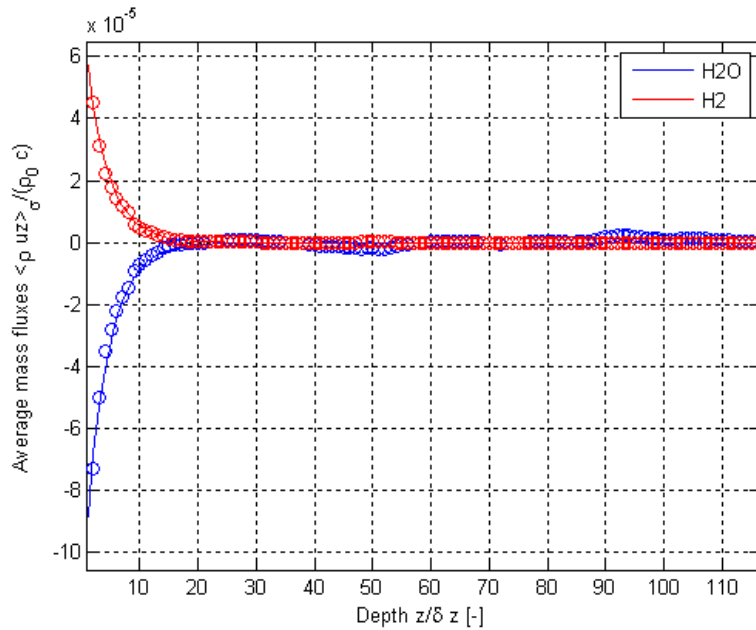


Figure E.32. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O and H₂ mixture for normalized surface-averaged mass flux.

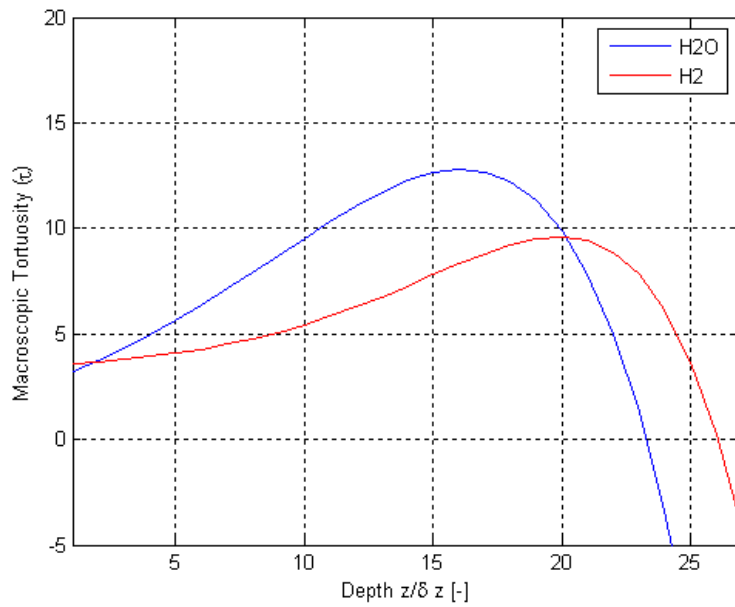


Figure E.33. Results of this thesis work for variations as a function of depth using Geometry G for the H₂O and H₂ mixture for τ .

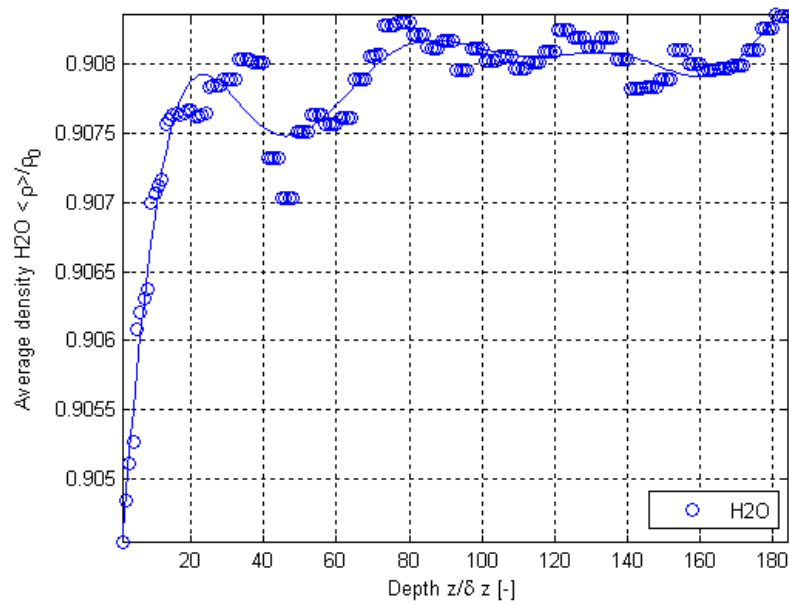


Figure E.34. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry G for the H₂O and H₂ mixture for H₂O.

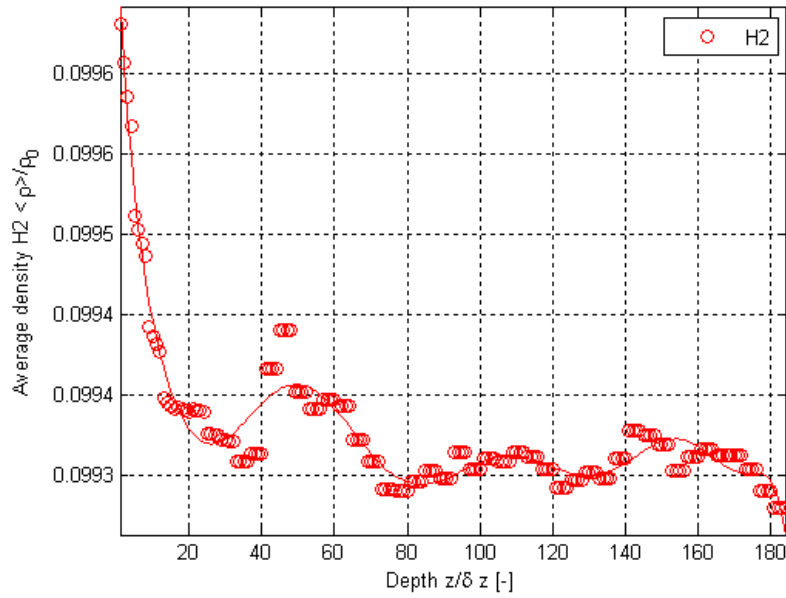


Figure E.35. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using the Geometry G for the H₂O and H₂ mixture for H₂.

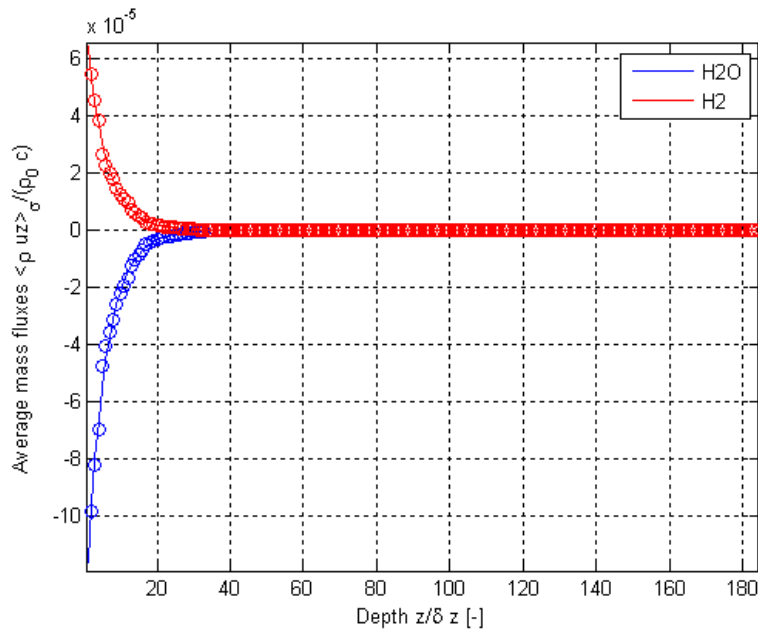


Figure E.36. Results of this thesis work for variations as a function of depth using Geometry G for the H₂O and H₂ mixture for normalized surface-averaged mass flux.

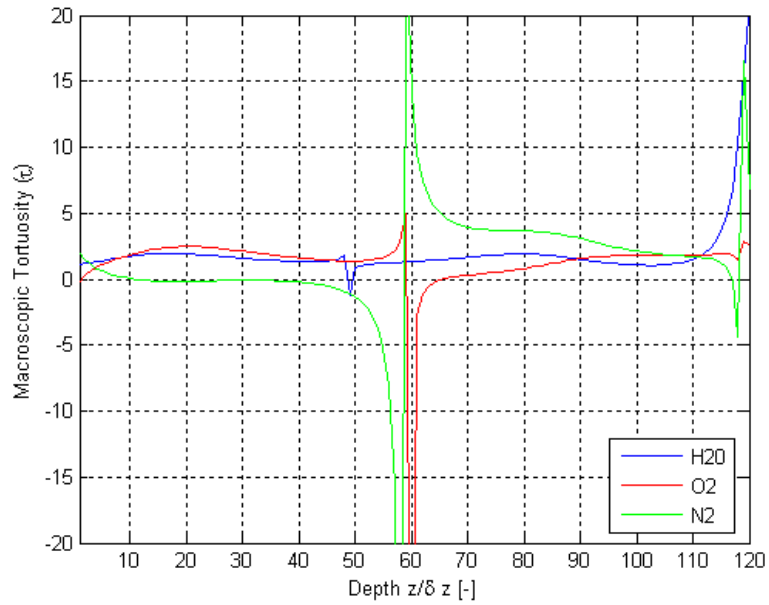


Figure E.37. Results of this thesis work for variations as a function of depth using Geometry A for the H₂O, O₂ and N₂ mixture for τ .

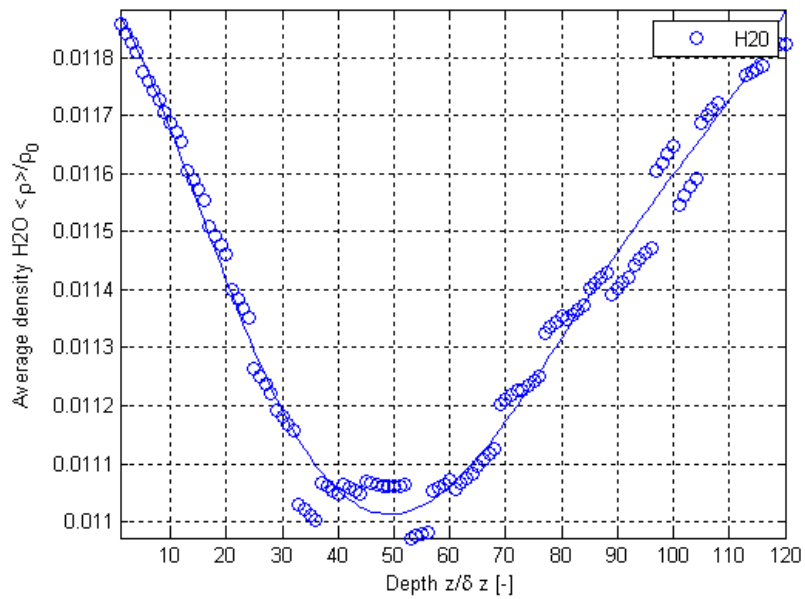


Figure E.38. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry A for the H₂O, O₂ and N₂ mixture for H₂O.

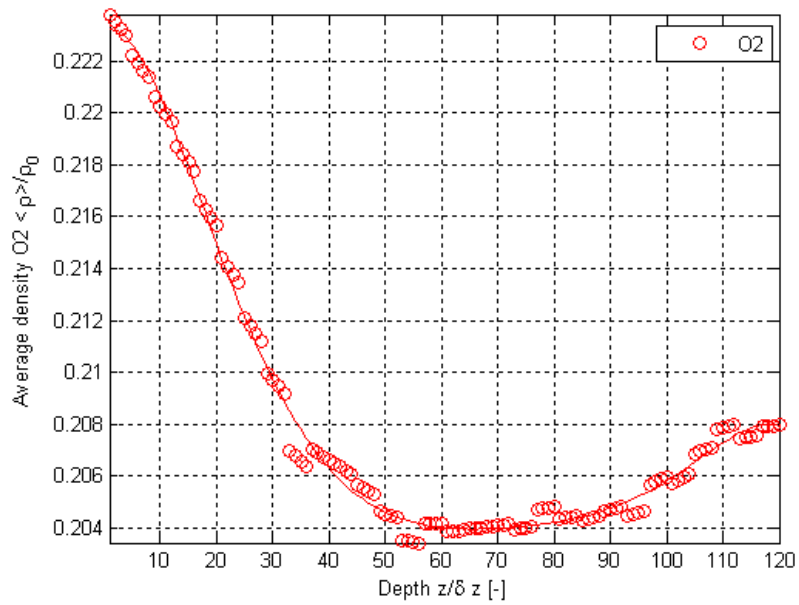


Figure E.39. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry A for the H_2O , O_2 and N_2 mixture for O_2 .

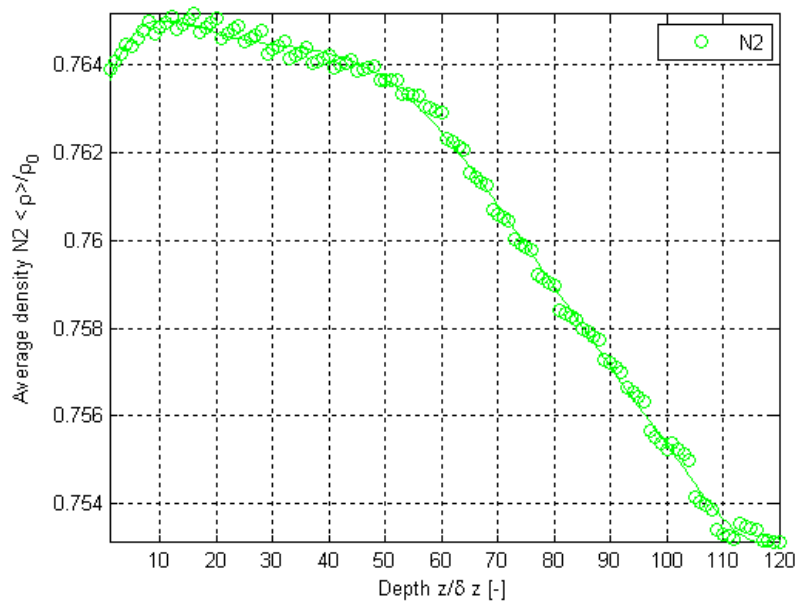


Figure E.40. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry A for the H_2O , O_2 and N_2 mixture for N_2 .

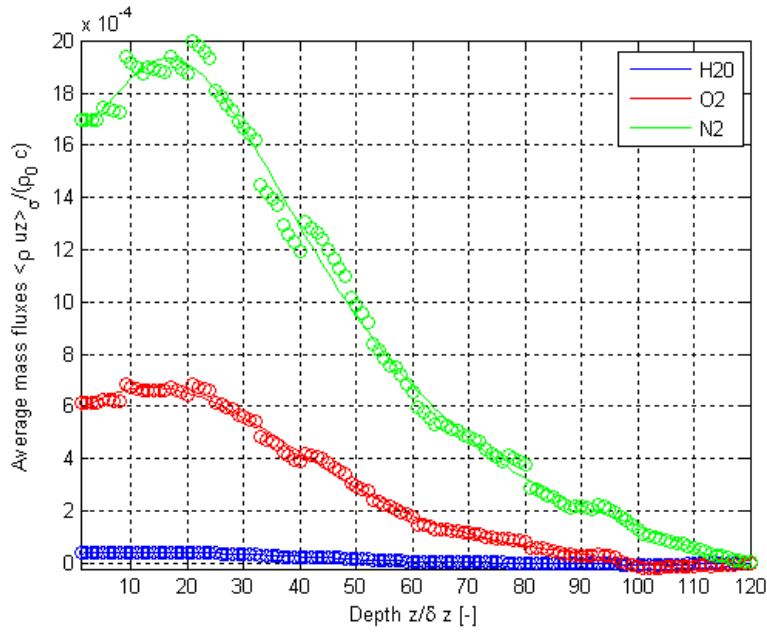


Figure E.41. Results of this thesis work for variations as a function of depth using Geometry A for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

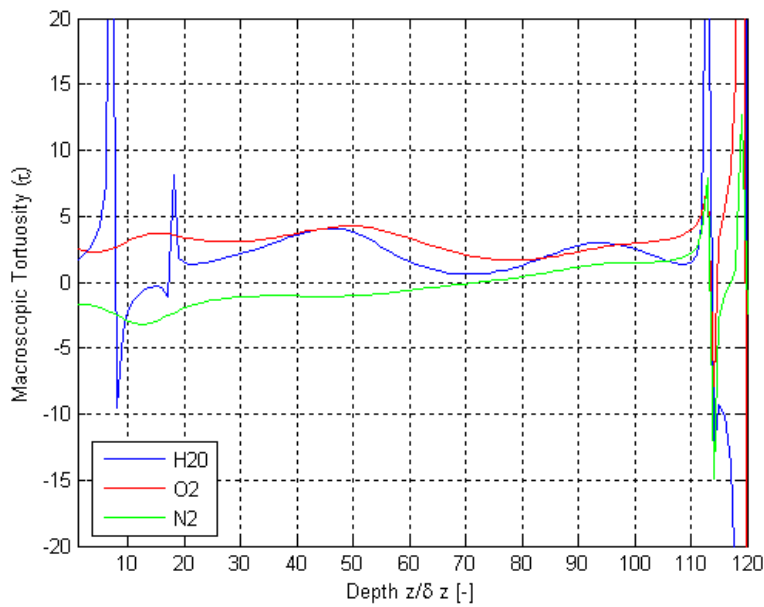


Figure E.42. Results of this thesis work for variations as a function of depth using Geometry B for the H₂O, O₂ and N₂ mixture for τ .

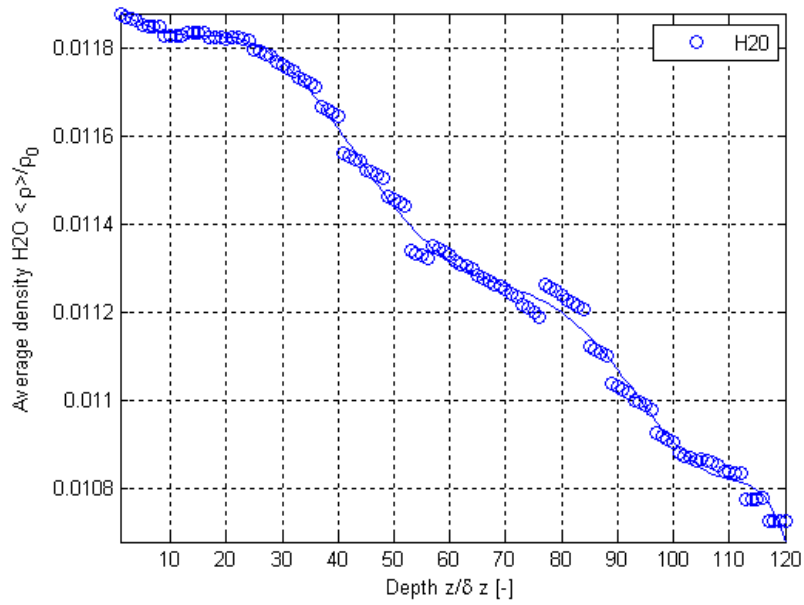


Figure E.43. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂ and N₂ mixture for H₂O.

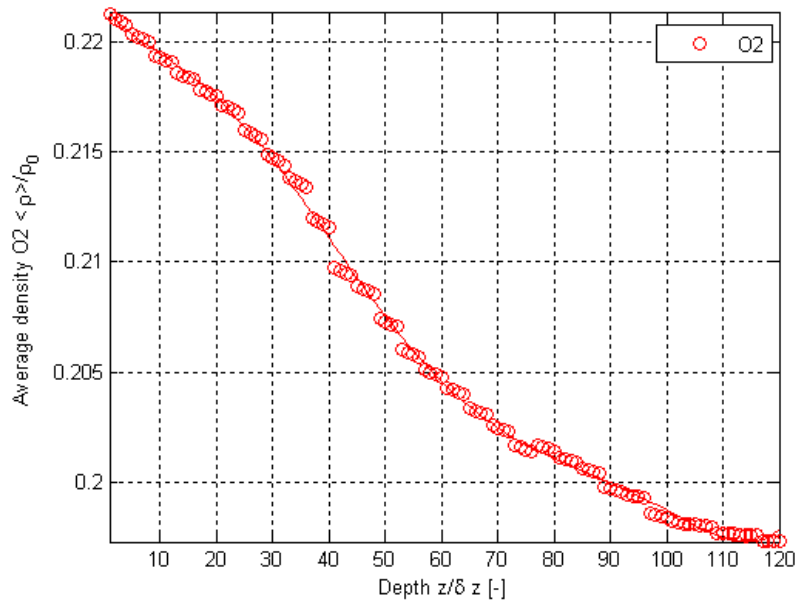


Figure E.44. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂ and N₂ mixture for O₂.

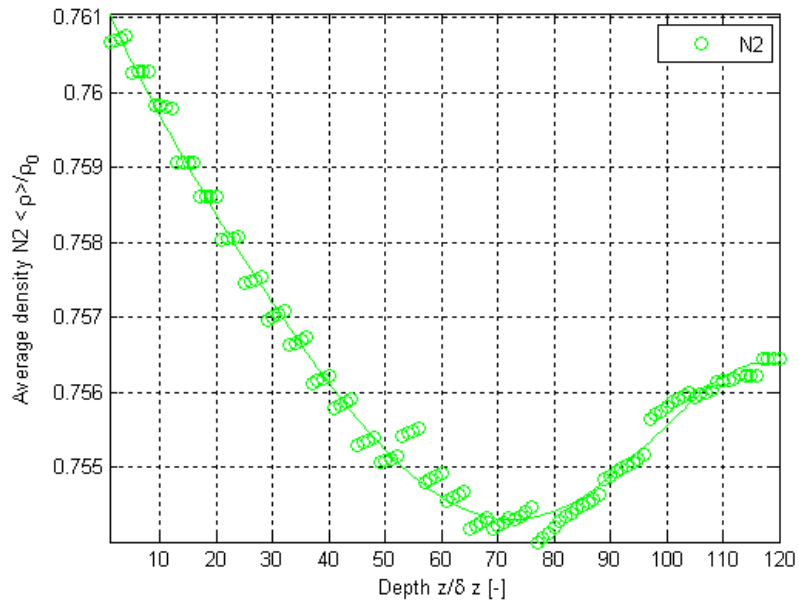


Figure E.45. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂ and N₂ mixture for N₂.

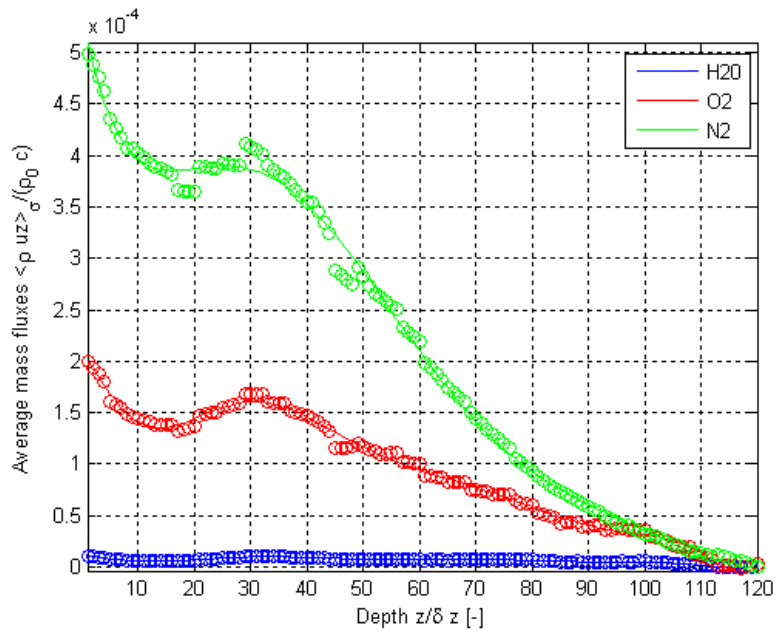


Figure E.46. Results of this thesis work for variations as a function of depth using Geometry B for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

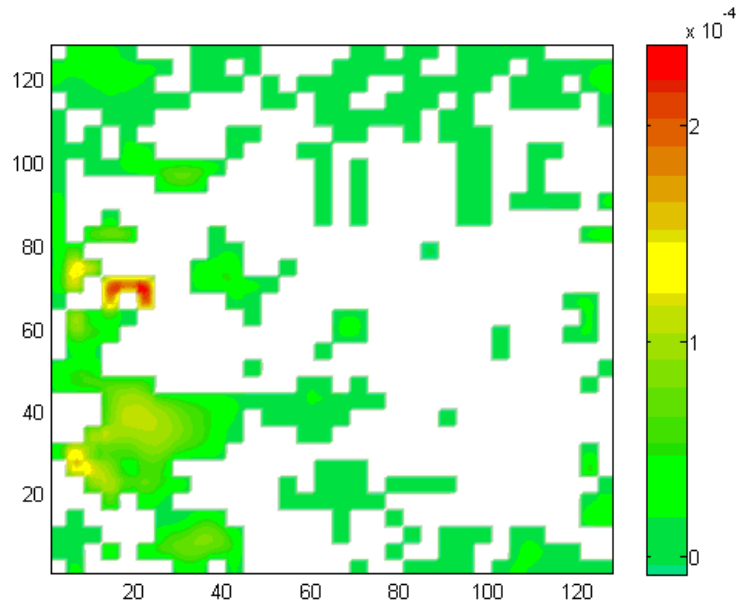


Figure E.47. Normalized mass flux contours for H₂O using Geometry B for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

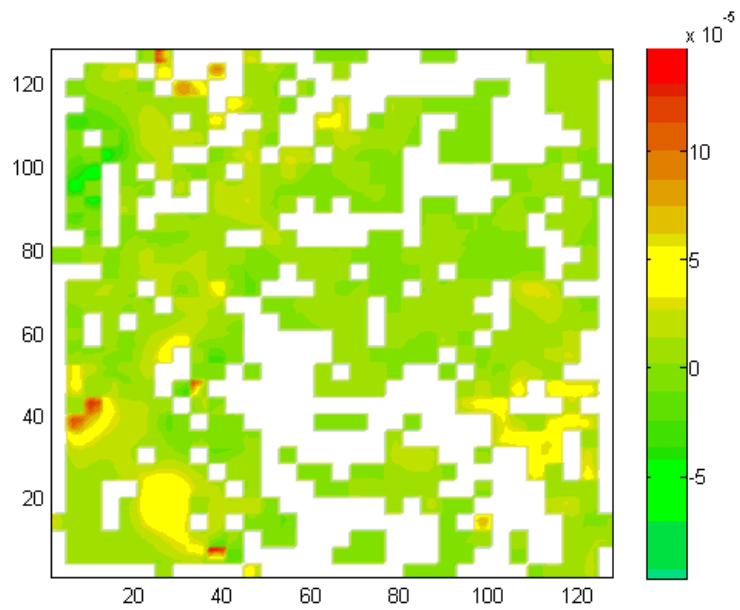


Figure E.48. Normalized mass flux contours for H₂O using Geometry B for the H₂O, O₂ and N₂ mixture for an intermediate plane.

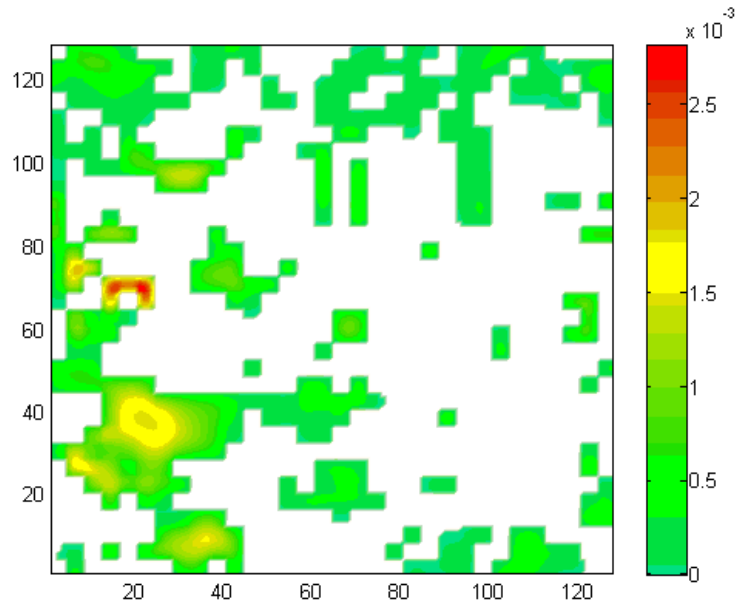


Figure E.49. Normalized mass flux contours for O₂ using Geometry B for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

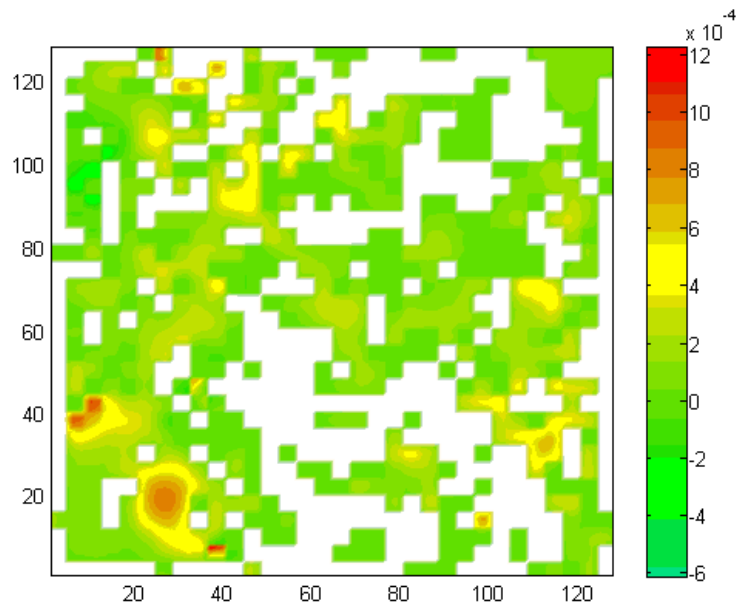


Figure E.50. Normalized mass flux contours for O₂ using Geometry B for the H₂O, O₂ and N₂ mixture for an intermediate plane.

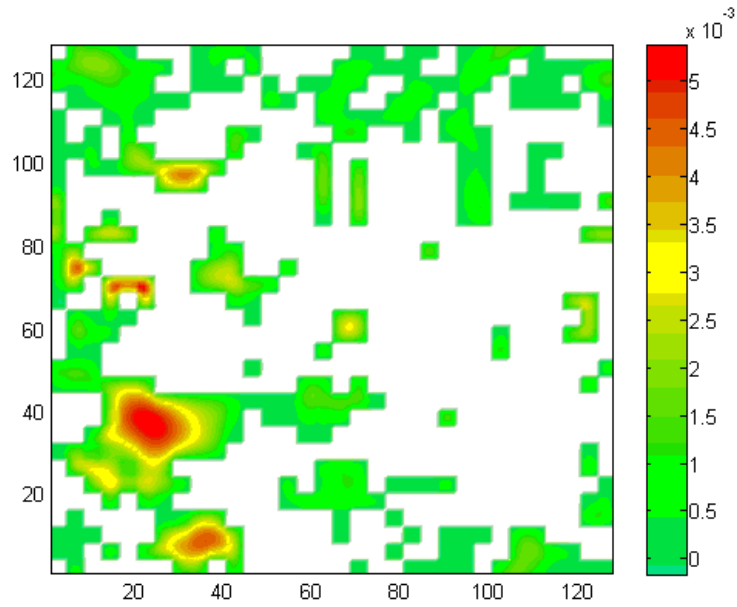


Figure E.51. Normalized mass flux contours for N₂ using Geometry B for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

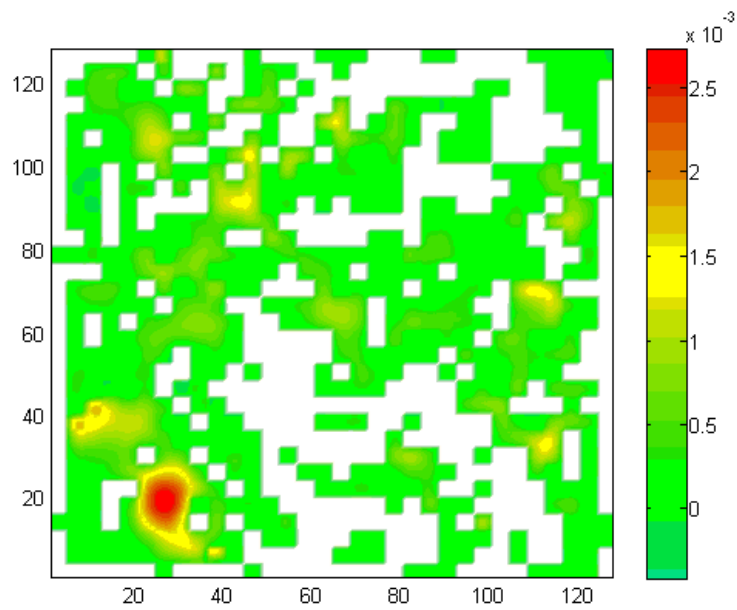


Figure E.52. Normalized mass flux contours for N₂ using Geometry B for the H₂O, O₂ and N₂ mixture for an intermediate plane.

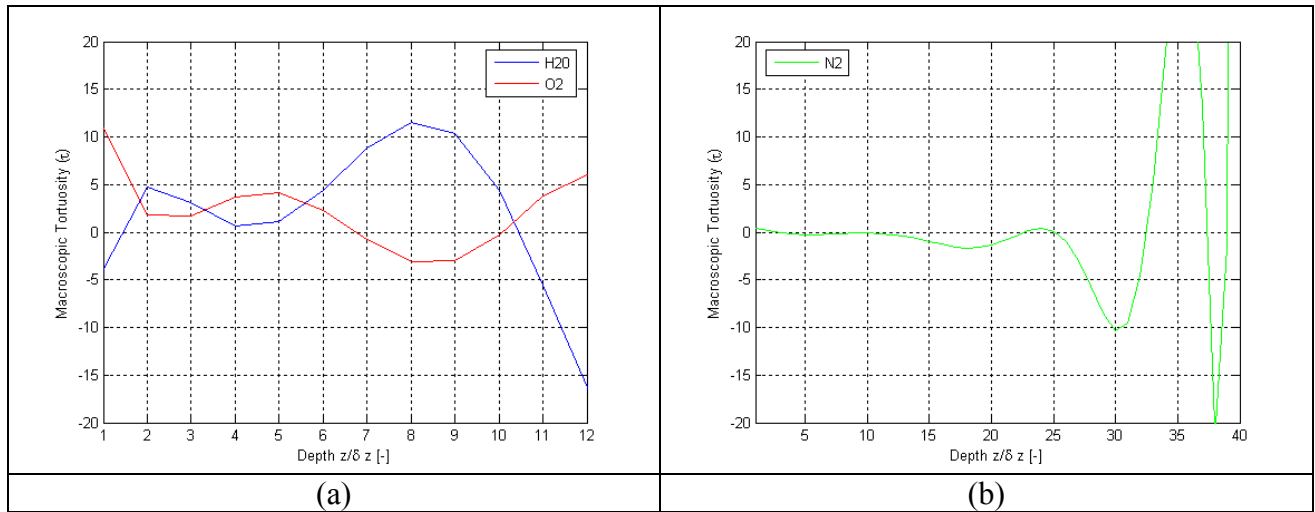


Figure E.53. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for τ for (a) H₂O and O₂ and (b) N₂.

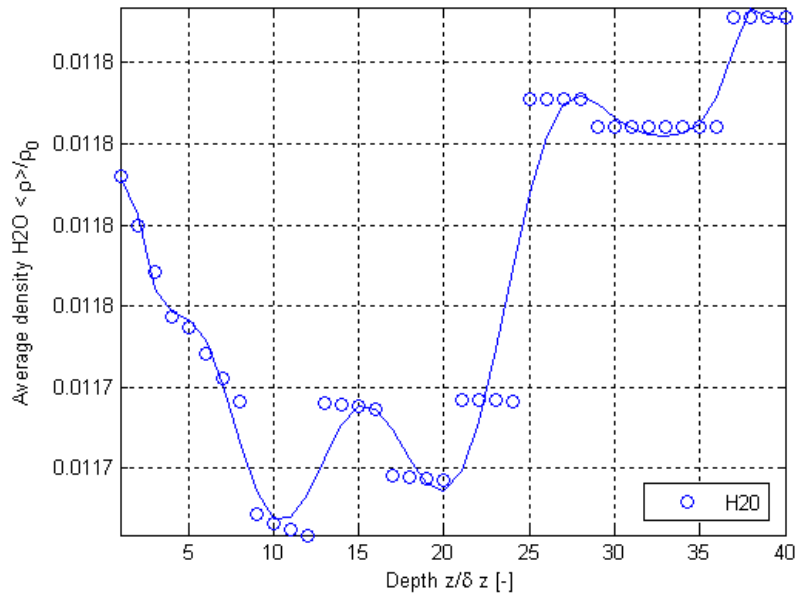


Figure E.54. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for H₂O.

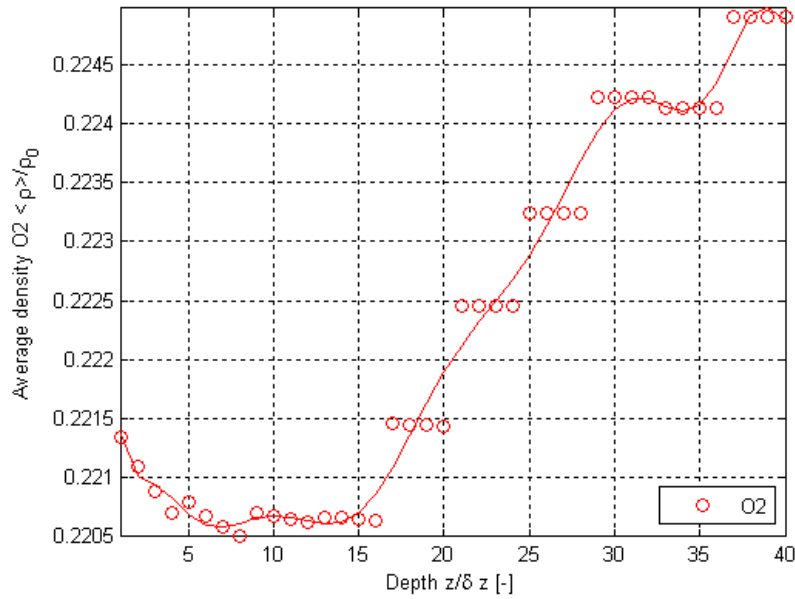


Figure E.55. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 and N_2 mixture for O_2 .

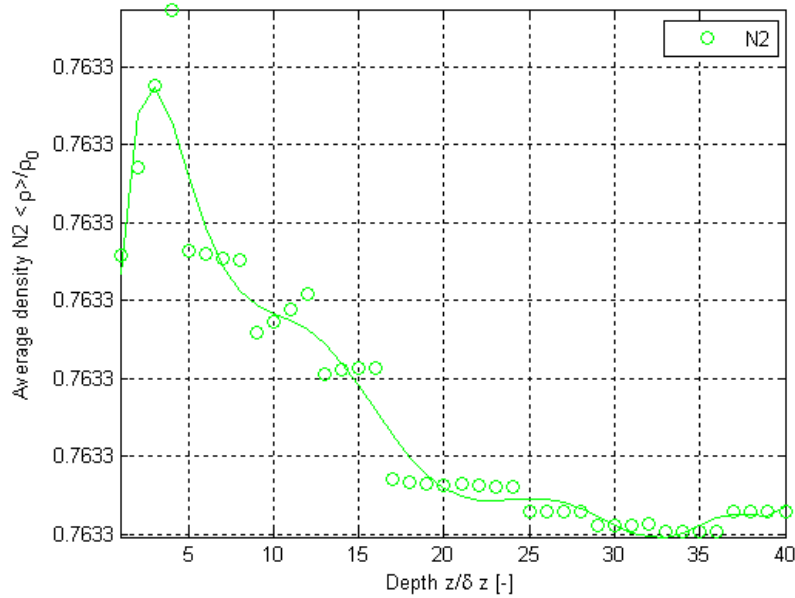


Figure E.56. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 and N_2 mixture for N_2 .

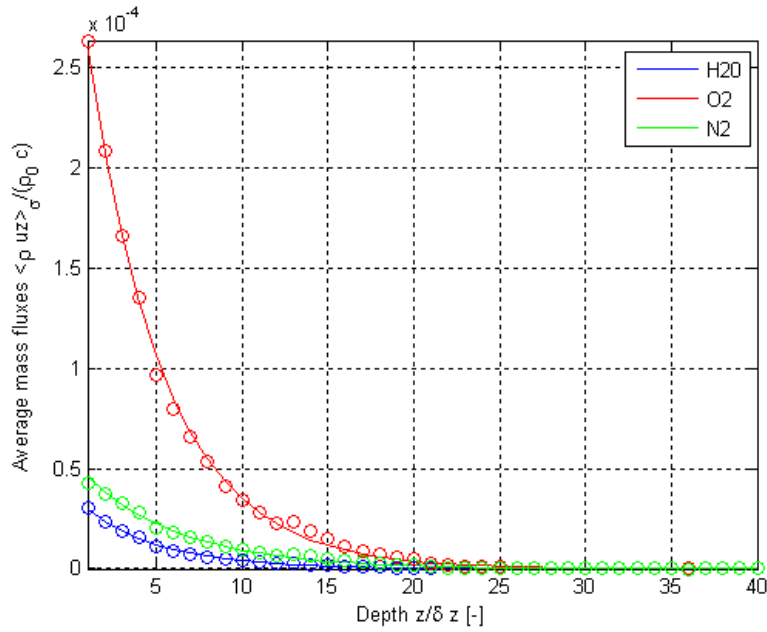


Figure E.57. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

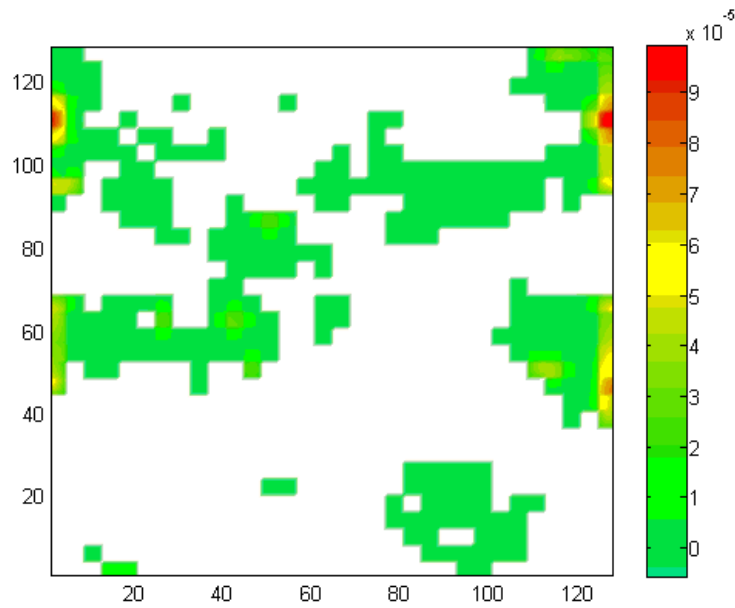


Figure E.58. Normalized mass flux contours for H₂O using Geometry C for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

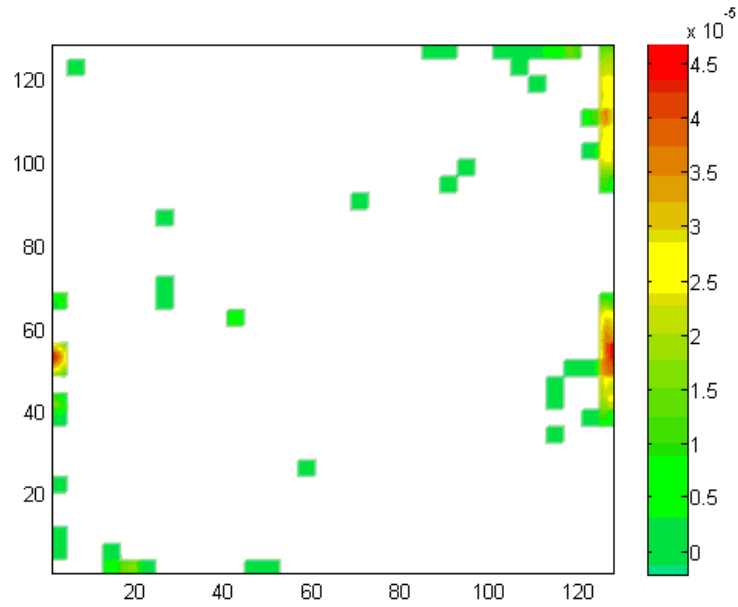


Figure E.59. Normalized mass flux contours for H₂O using Geometry C for the H₂O, O₂ and N₂ mixture for an intermediate plane.

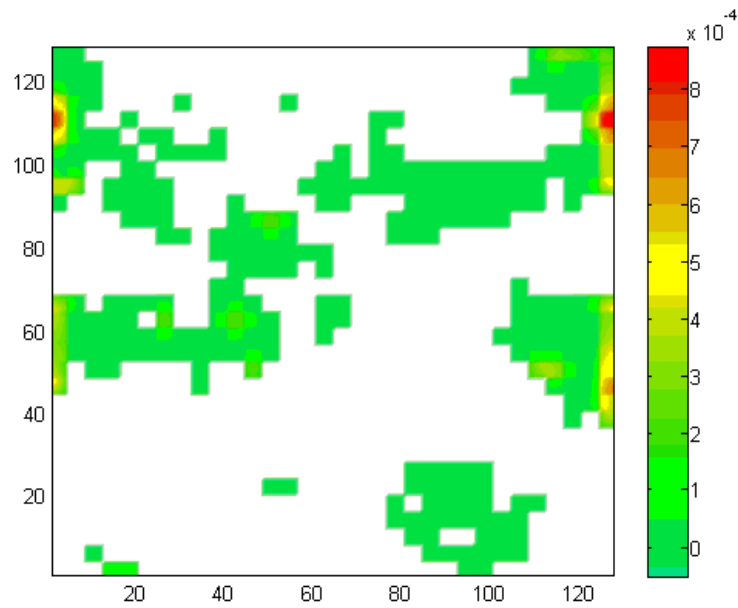


Figure E.60. Normalized mass flux contours for O₂ using Geometry C for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

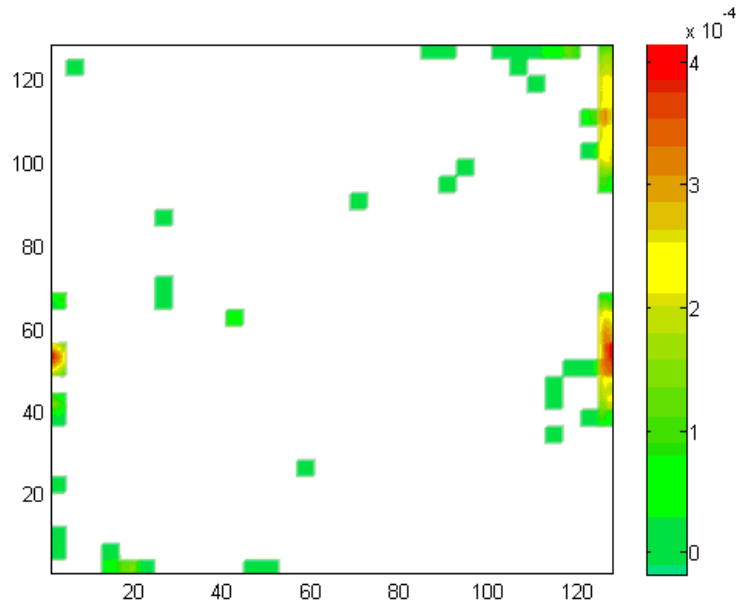


Figure E.61. Normalized mass flux contours for O₂ using Geometry C for the H₂O, O₂ and N₂ mixture for an intermediate plane.

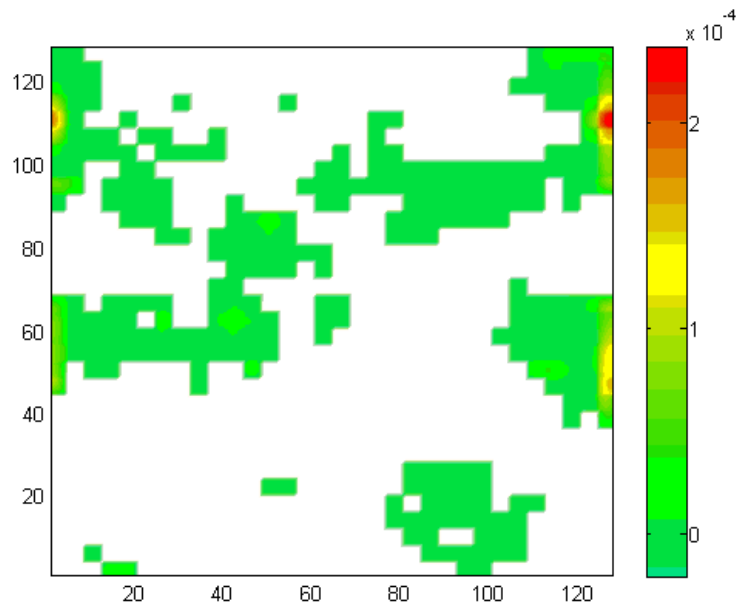


Figure E.62. Normalized mass flux contours for N₂ using Geometry C for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

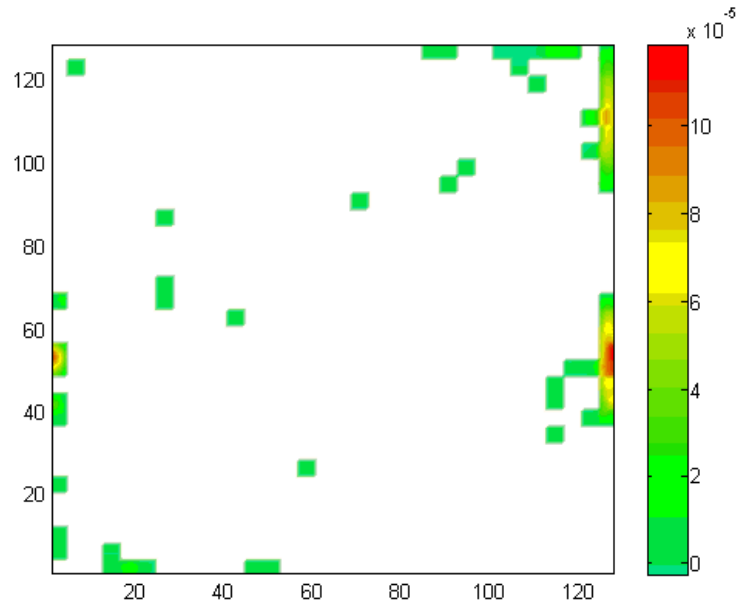


Figure E.63. Normalized mass flux contours for N_2 using Geometry C for the H_2O , O_2 and N_2 mixture for an intermediate plane.

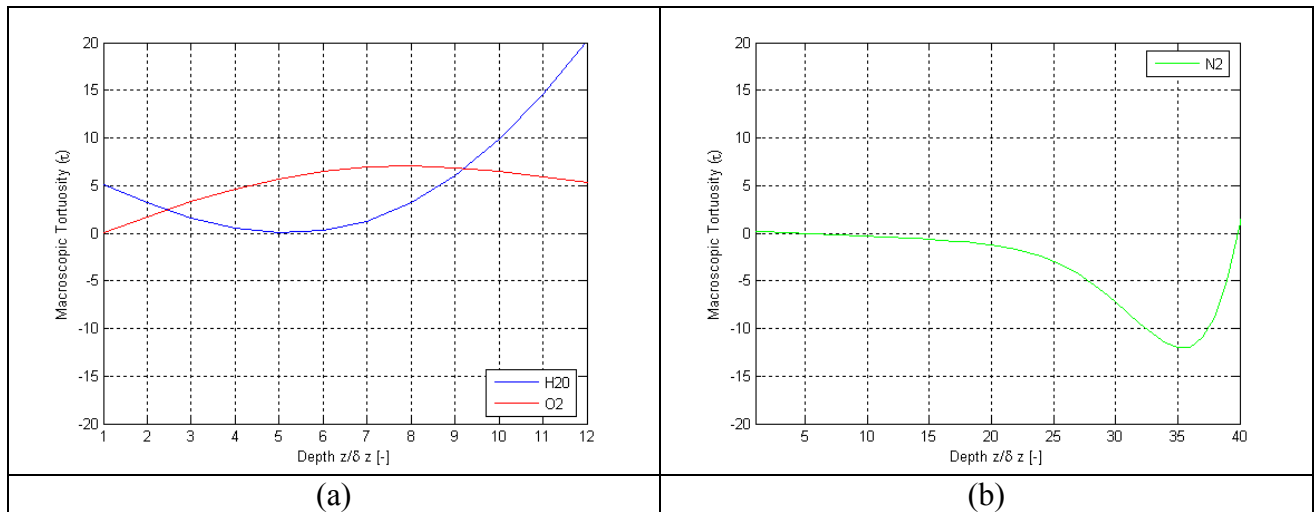


Figure E.64. Results of this thesis work for variations as a function of depth using Geometry D for the H_2O , O_2 and N_2 mixture for τ for (a) H_2O and O_2 and (b) N_2 .

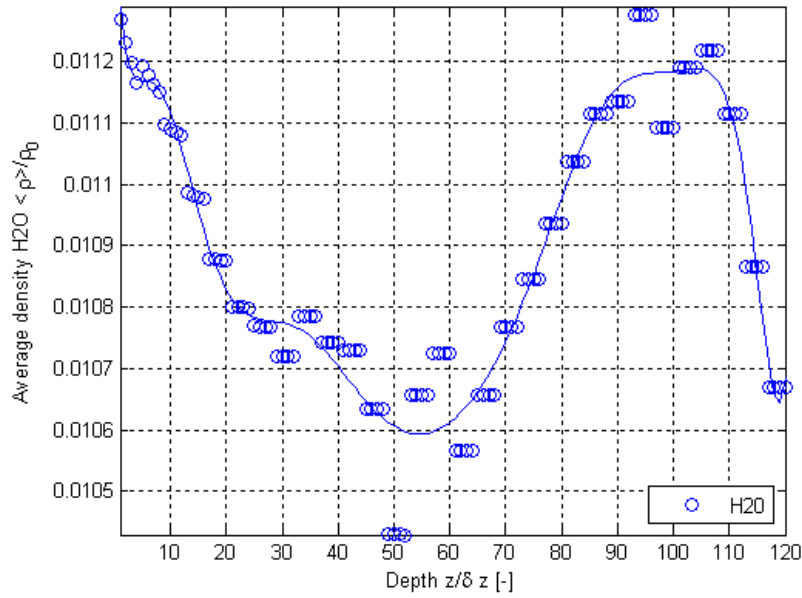


Figure E.65. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for H₂O.

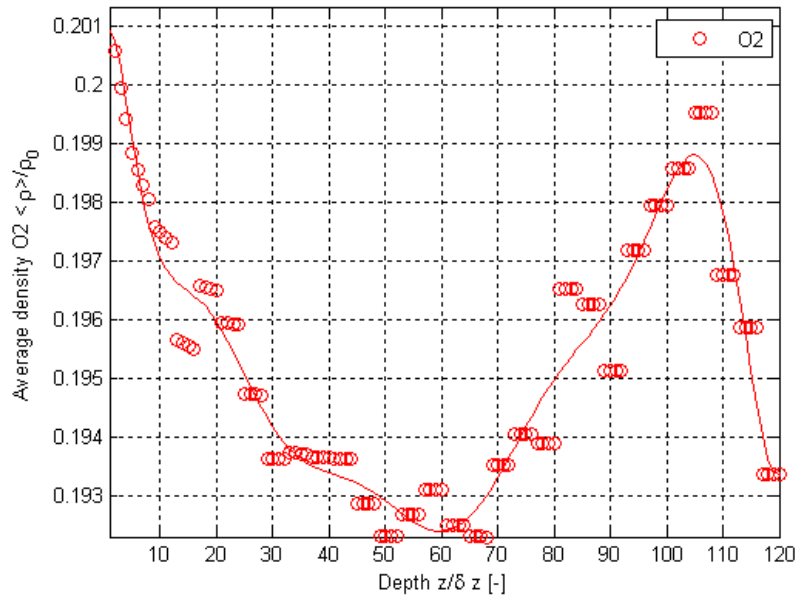


Figure E.66. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for O₂.

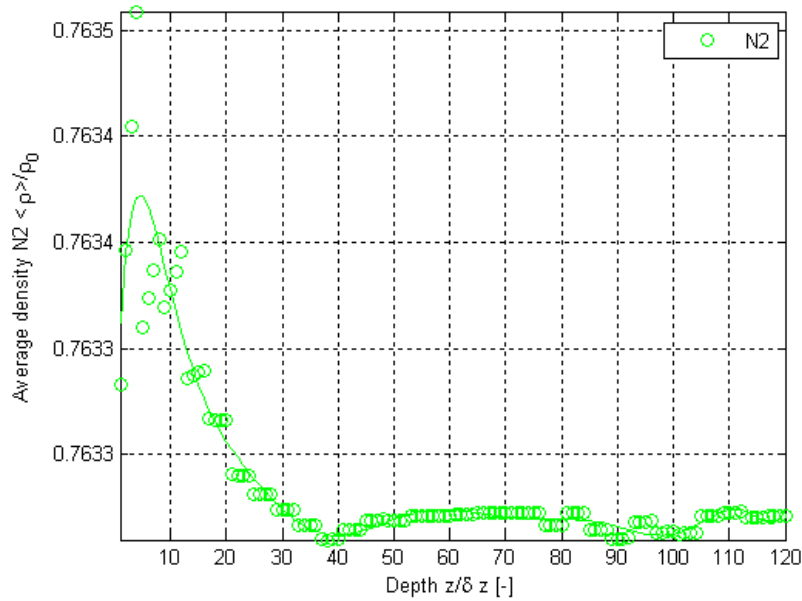


Figure E.67. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for N₂.

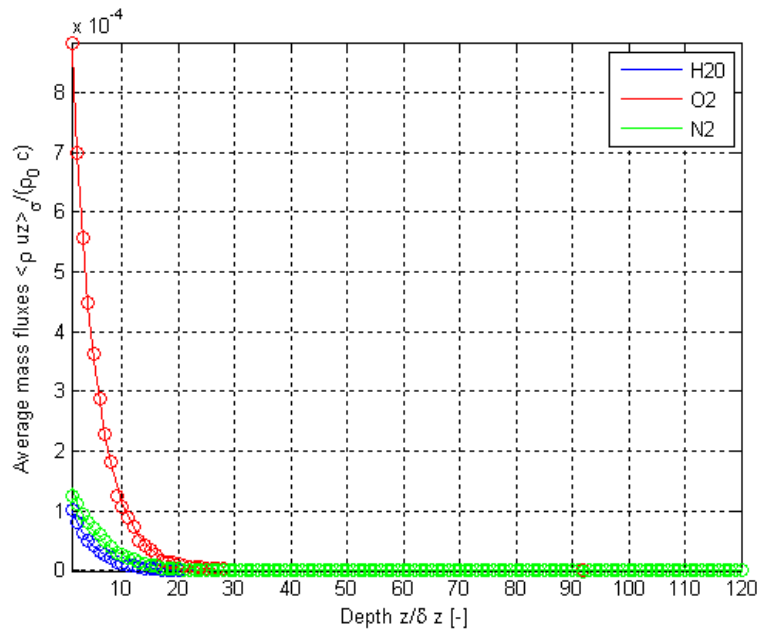


Figure E.68. Results of this thesis work for variations as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

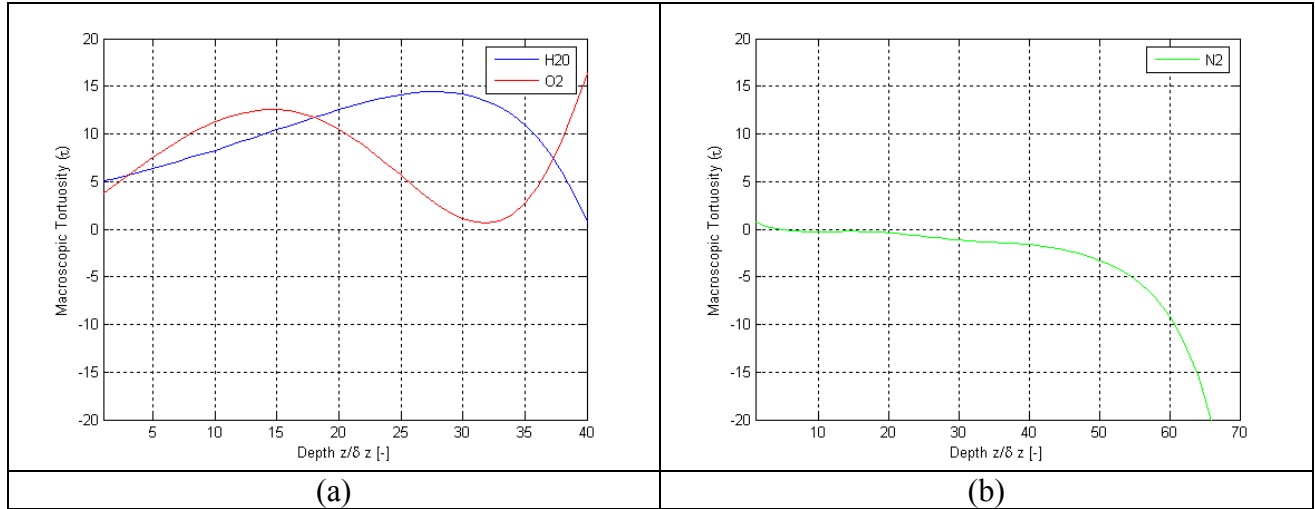


Figure E.69. Results of this thesis work for variations as a function of depth using Geometry E for the H₂O, O₂ and N₂ mixture for τ for (a) H₂O and O₂ and (b) N₂.

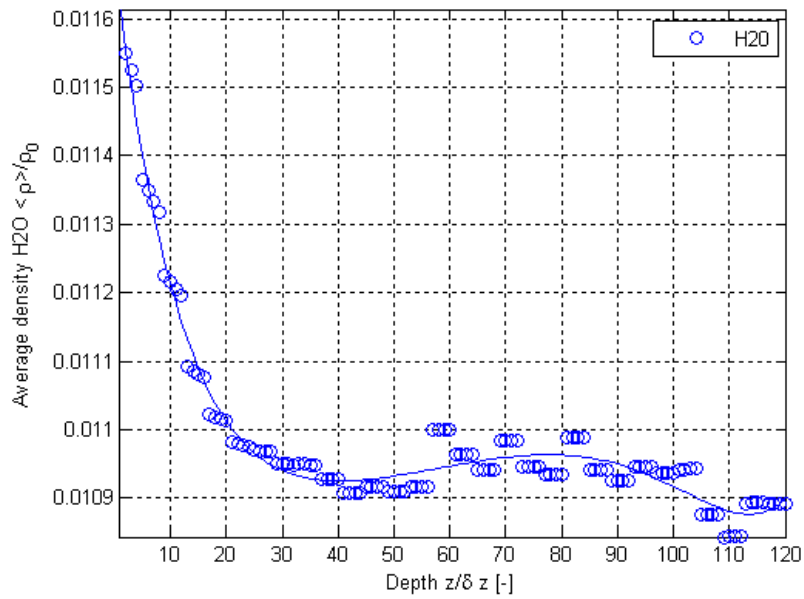


Figure E.70. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry E for the H₂O, O₂ and N₂ mixture for H₂O.

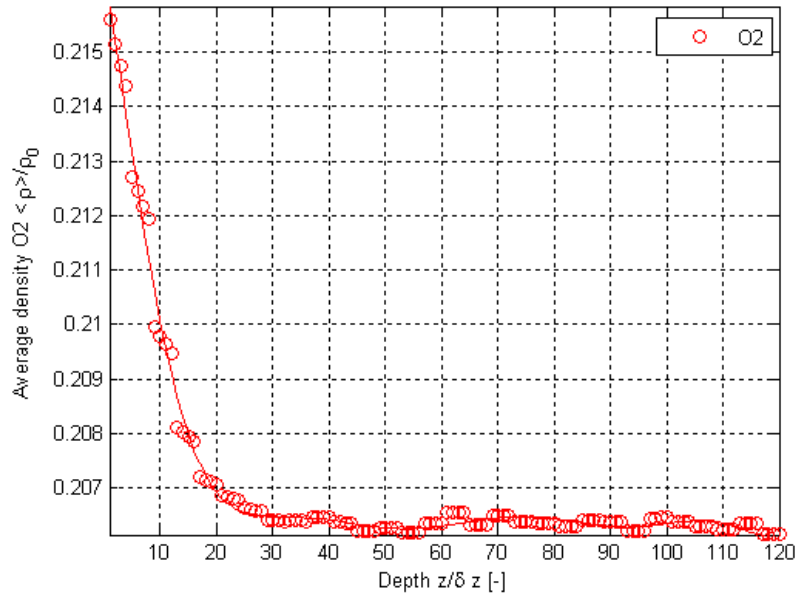


Figure E.71. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry E for the H_2O , O_2 and N_2 mixture for O_2 .

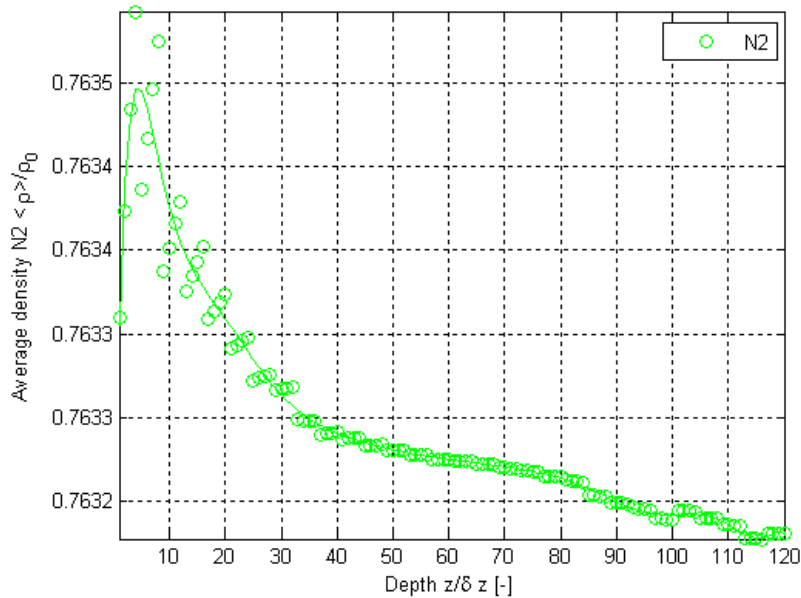


Figure E.72. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry E for the H_2O , O_2 and N_2 mixture for N_2 .

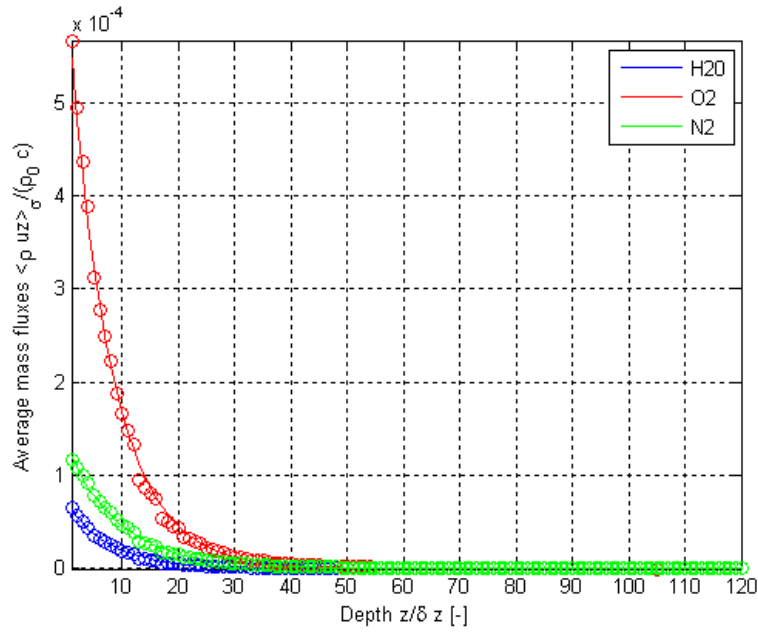


Figure E.73. Results of this thesis work for variations as a function of depth using Geometry E for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

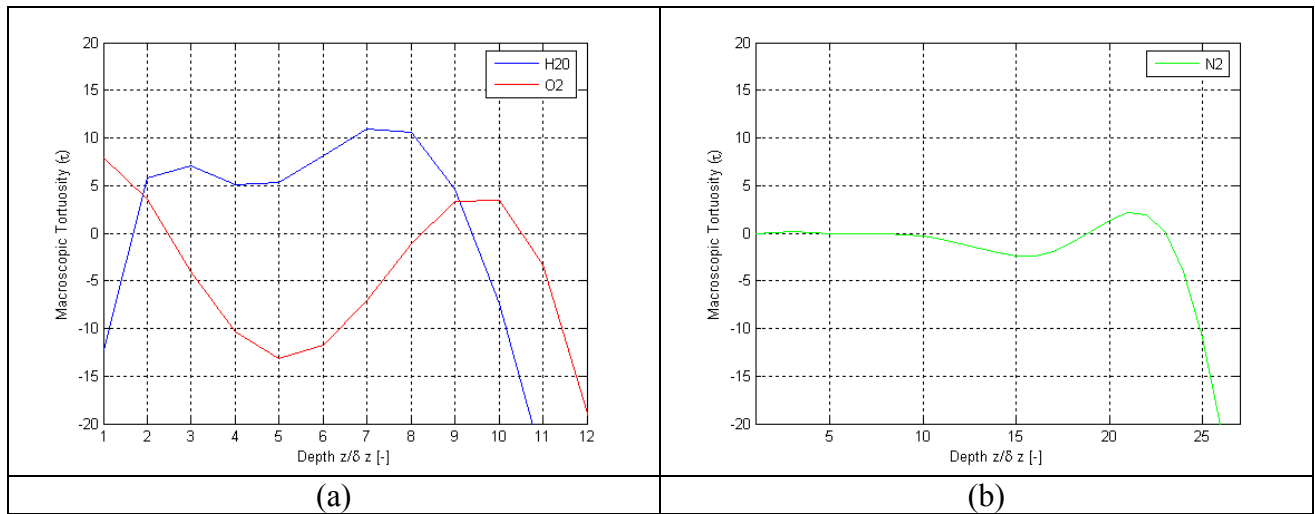


Figure E.74. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for τ for (a) H₂O and O₂ and (b) N₂.

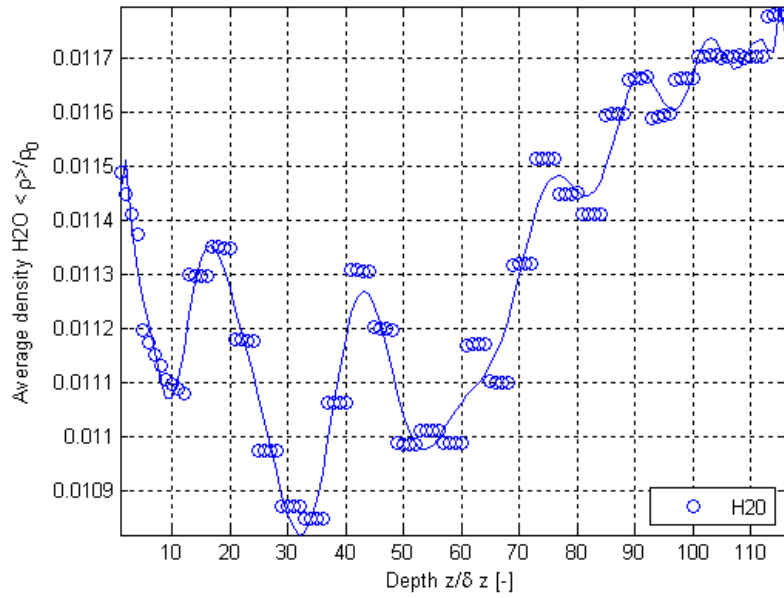


Figure E.75. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for H₂O.

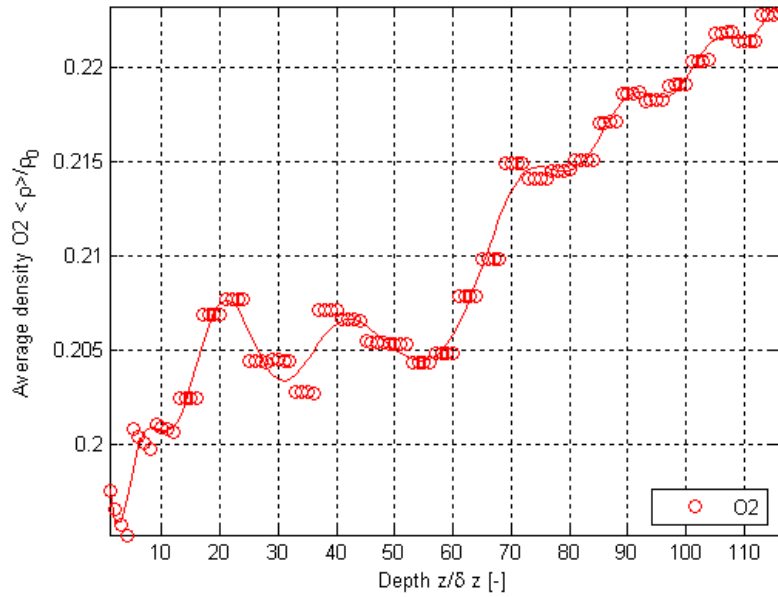


Figure E.76. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for O₂.

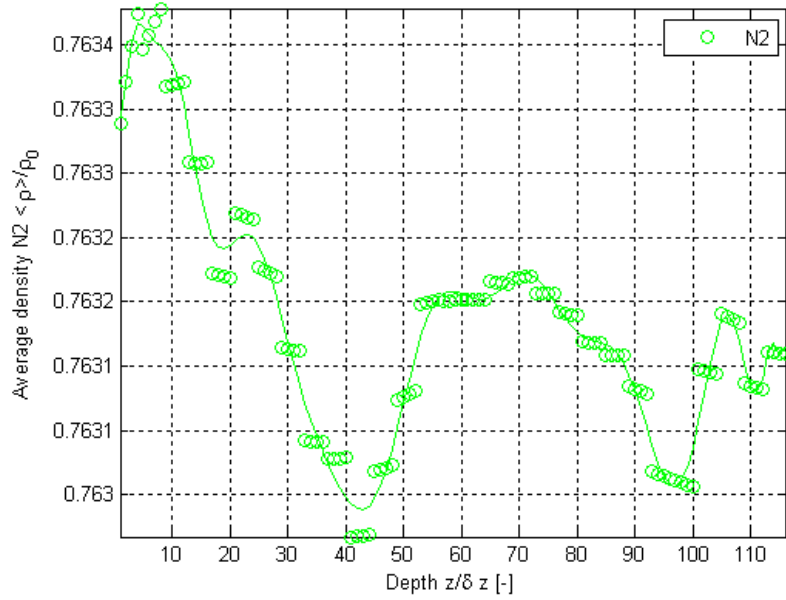


Figure E.77. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for N₂.

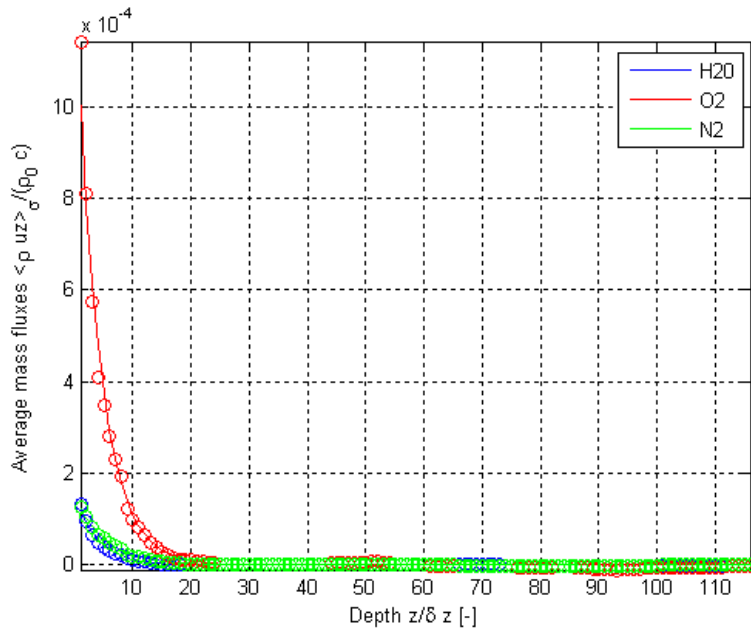


Figure E.78. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

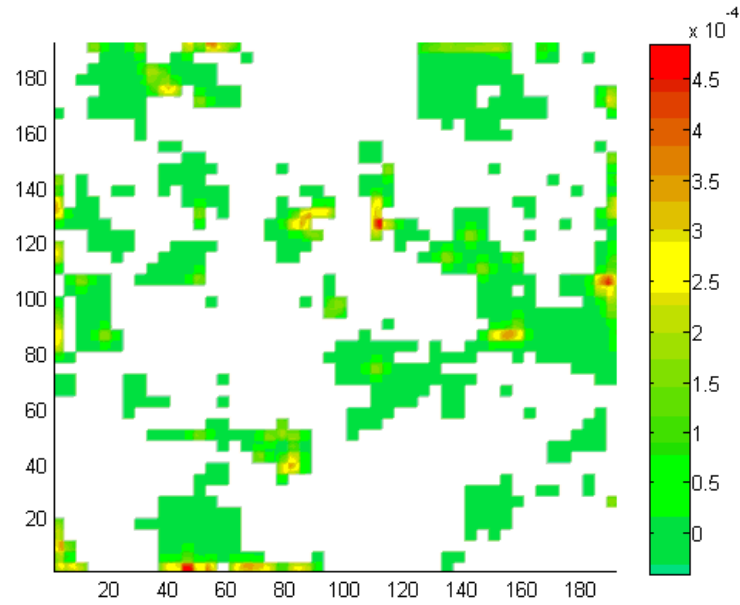


Figure E.79. Normalized mass flux contours for H₂O using Geometry F for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

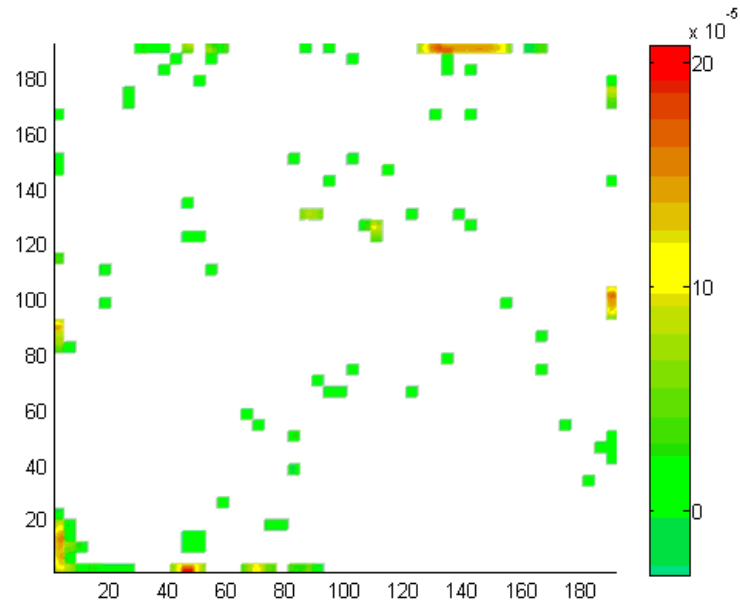


Figure E.80. Normalized mass flux contours for H₂O using Geometry F for the H₂O, O₂ and N₂ mixture for an intermediate plane.

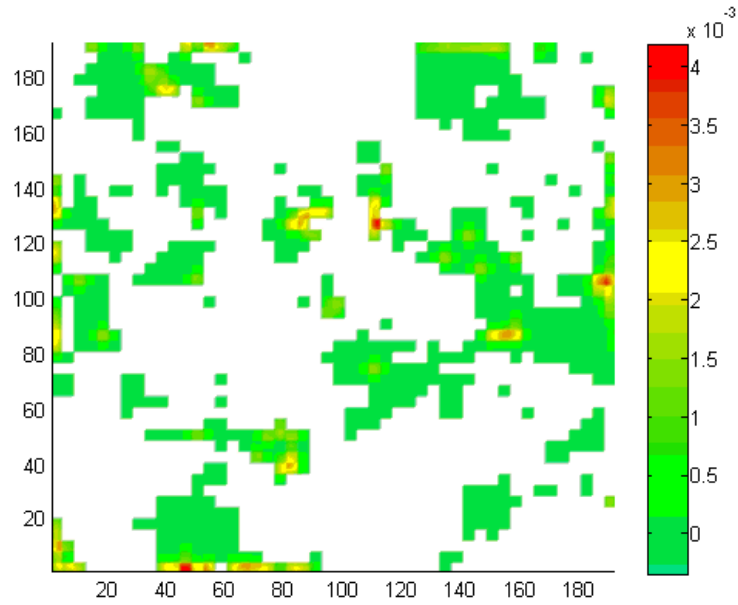


Figure E.81. Normalized mass flux contours for O₂ using Geometry F for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

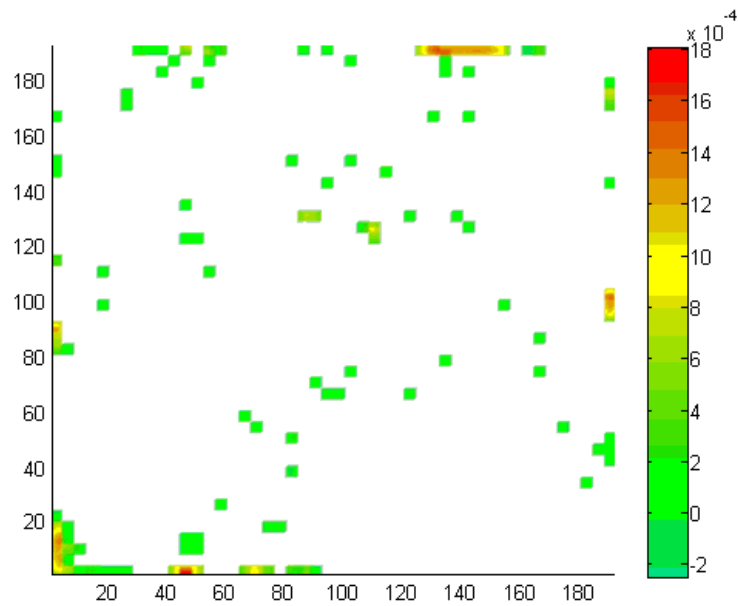


Figure E.82. Normalized mass flux contours for O₂ using Geometry F for the H₂O, O₂ and N₂ mixture for an intermediate plane.

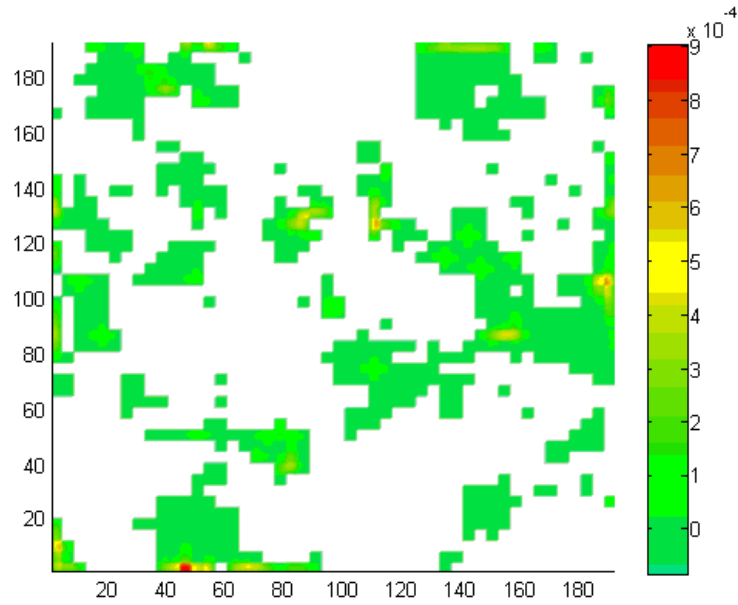


Figure E.83. Normalized mass flux contours for N₂ using Geometry F for the H₂O, O₂ and N₂ mixture for the inlet/outlet plane.

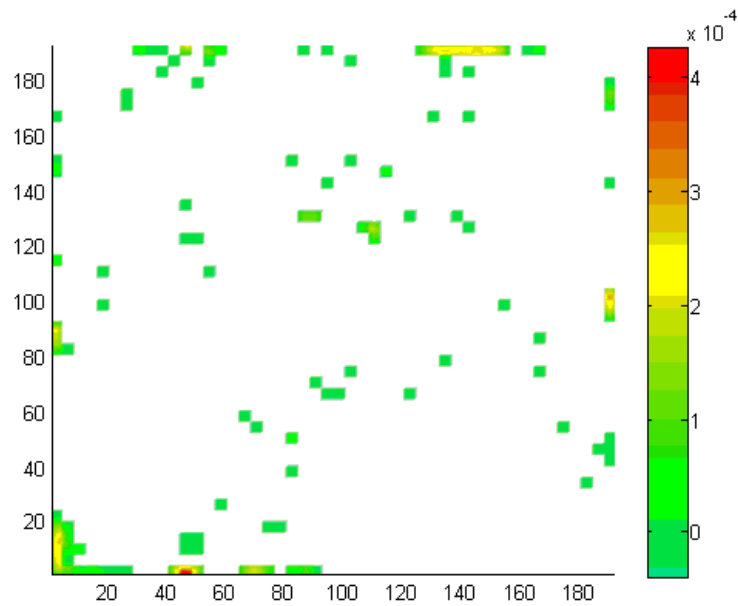


Figure E.84. Normalized mass flux contours for N₂ using Geometry F for the H₂O, O₂ and N₂ mixture for an intermediate plane.

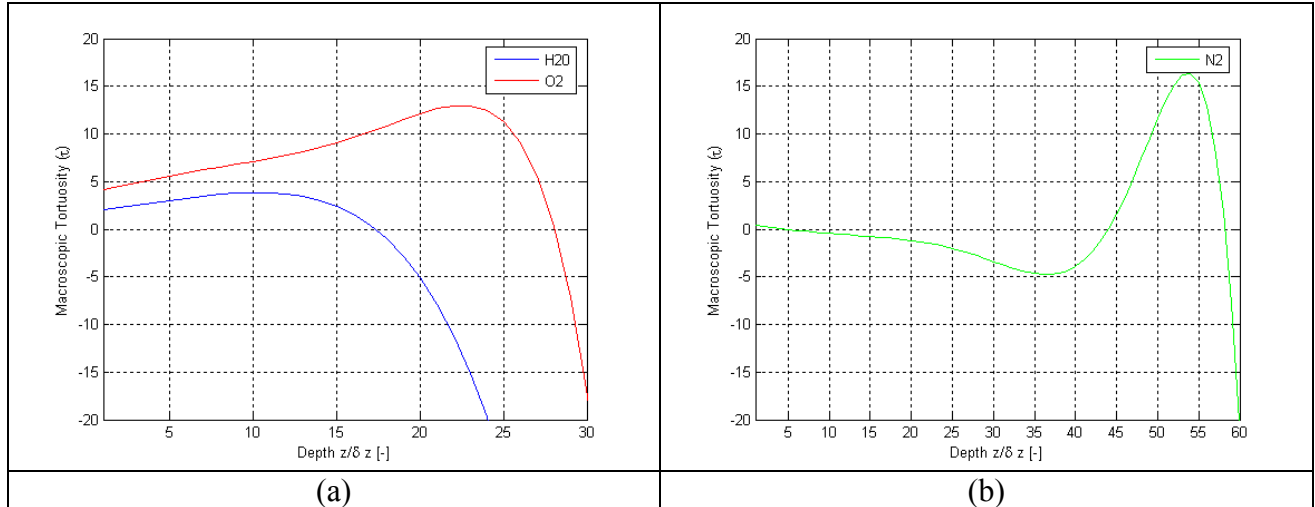


Figure E.85. Results of this thesis work for variations as a function of depth using Geometry G for the H₂O, O₂ and N₂ mixture for τ for (a) H₂O and O₂ and (b) N₂.

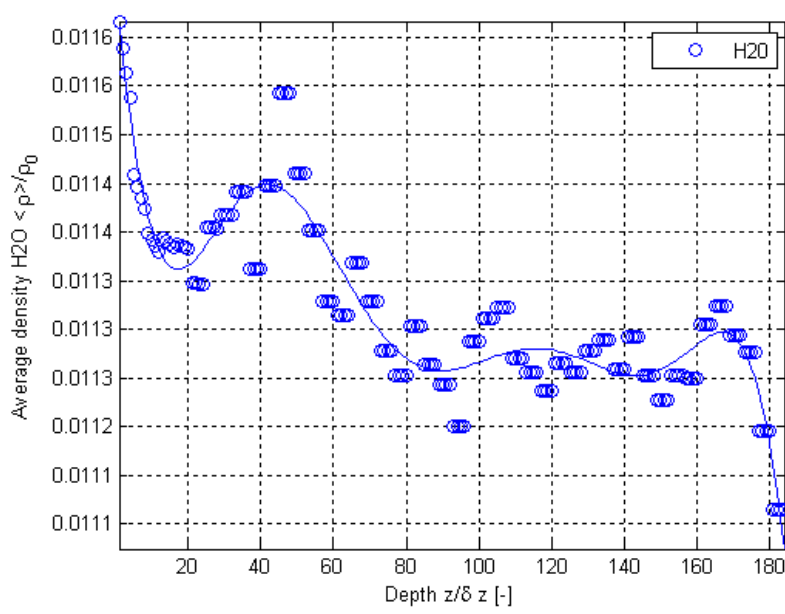


Figure E.86. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry G for the H₂O, O₂ and N₂ mixture for H₂O.

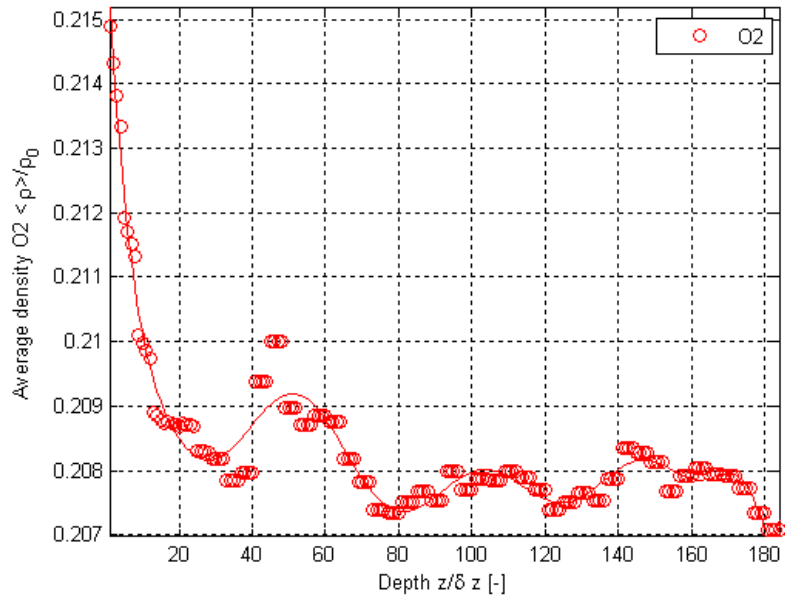


Figure E.87. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry G for the H_2O , O_2 and N_2 mixture for O_2 .

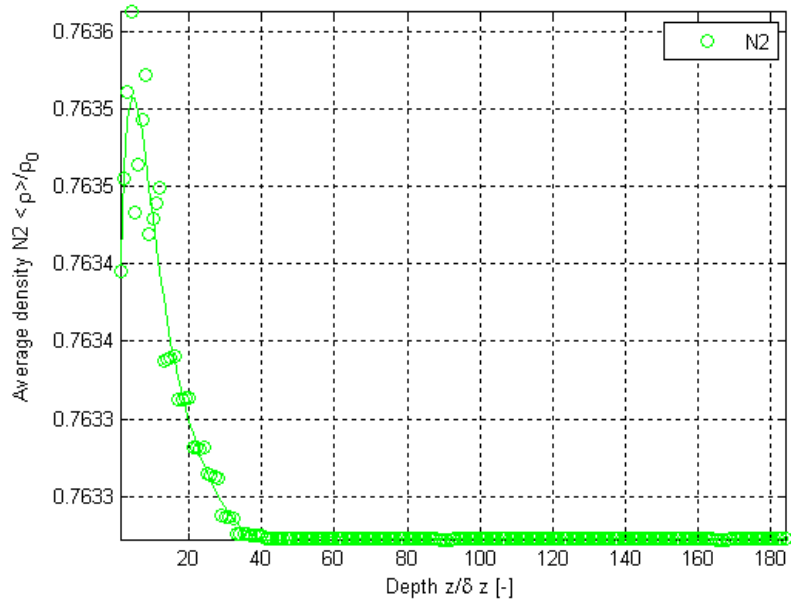


Figure E.88. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry G for the H_2O , O_2 and N_2 mixture for N_2 .

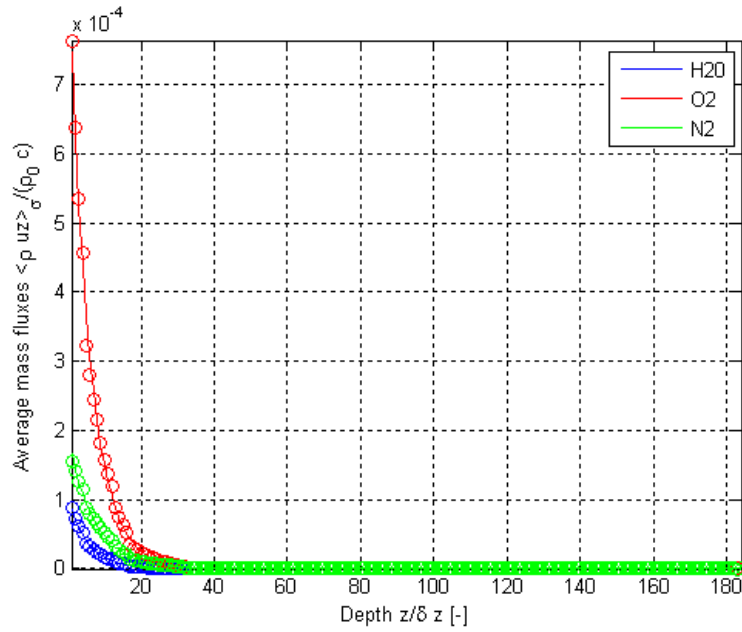


Figure E.89. Results of this thesis work for variations as a function of depth using Geometry G for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

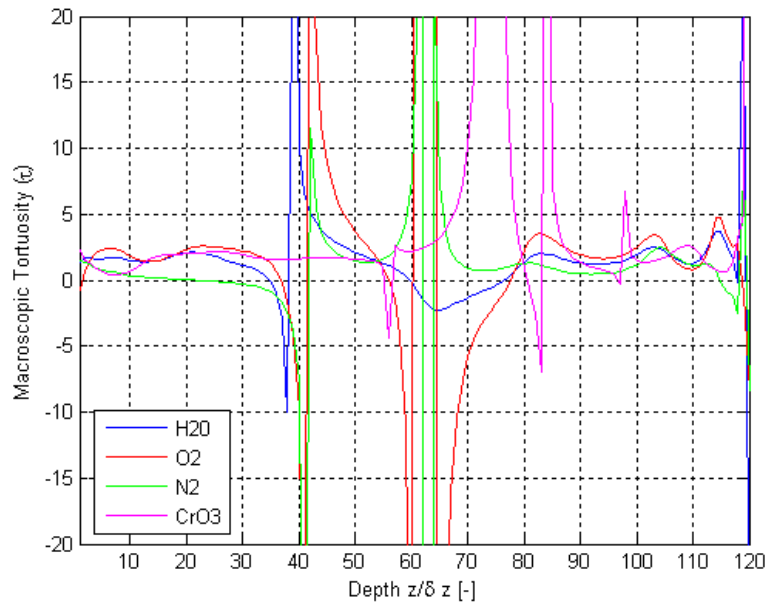


Figure E.90. Results of this thesis work for variations as a function of depth using Geometry A for the H₂O, O₂, N₂ and CrO₃ mixture for τ .

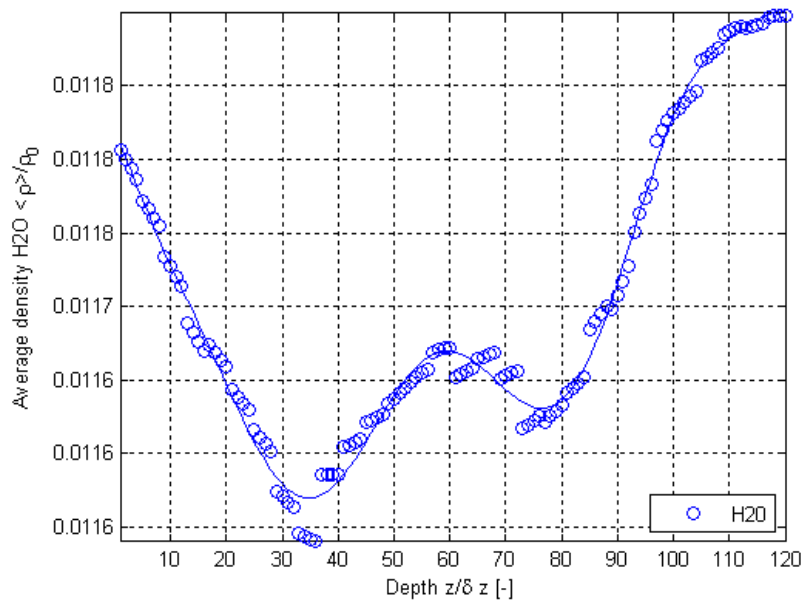


Figure E.91. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry A for the H₂O, O₂, N₂ and CrO₃ mixture for H₂O.

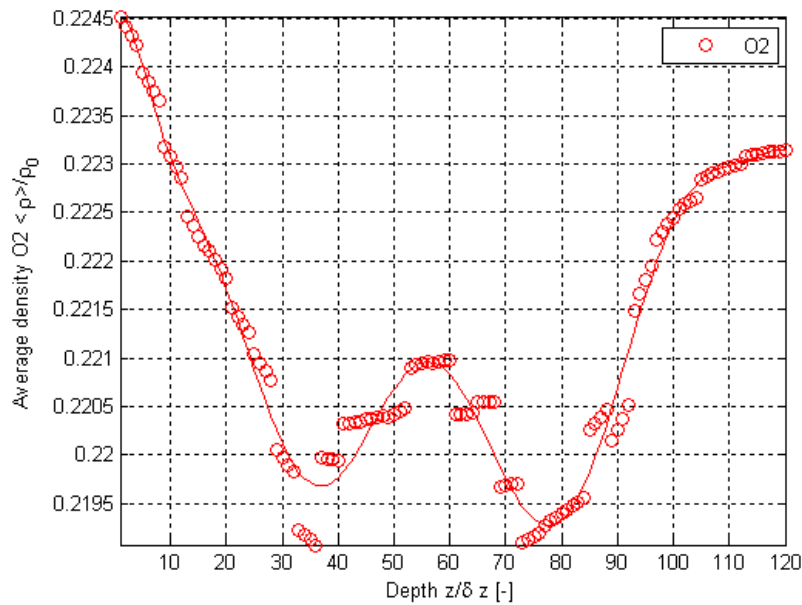


Figure E.92. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry A for the H₂O, O₂, N₂ and CrO₃ mixture for O₂.

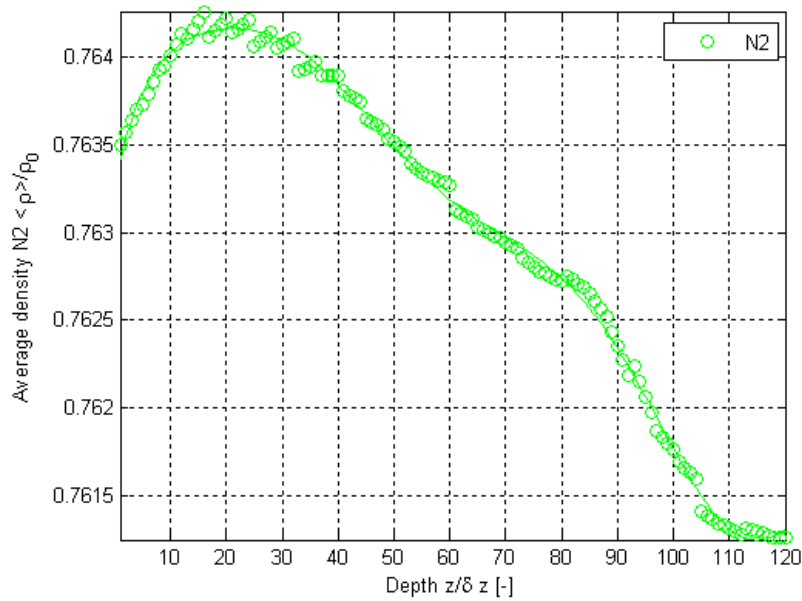


Figure E.93. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry A for the H_2O , O_2 , N_2 and CrO_3 mixture for N_2 .

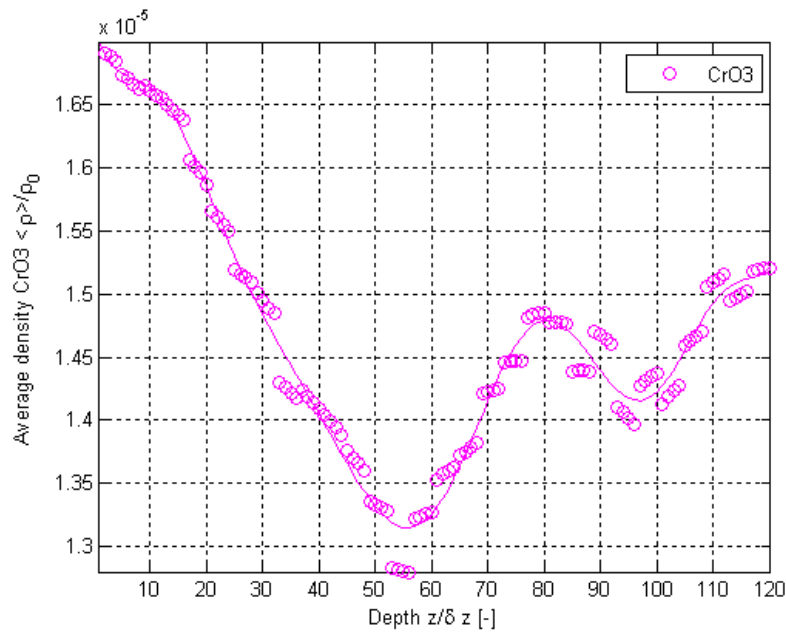


Figure E.94. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry A for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 .

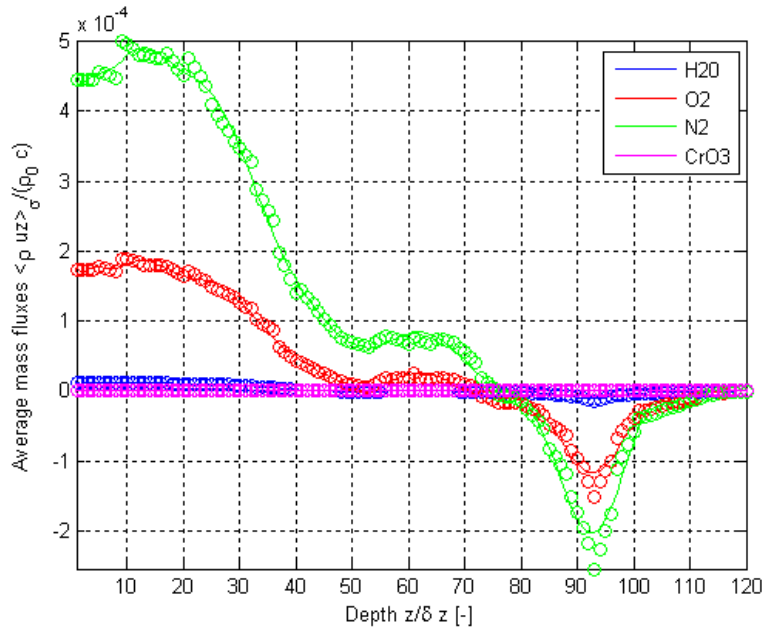


Figure E.95. Results of this thesis work for variations as a function of depth using Geometry A for the H_2O , O_2 , N_2 and CrO_3 mixture for the normalized surface-averaged mass flux.

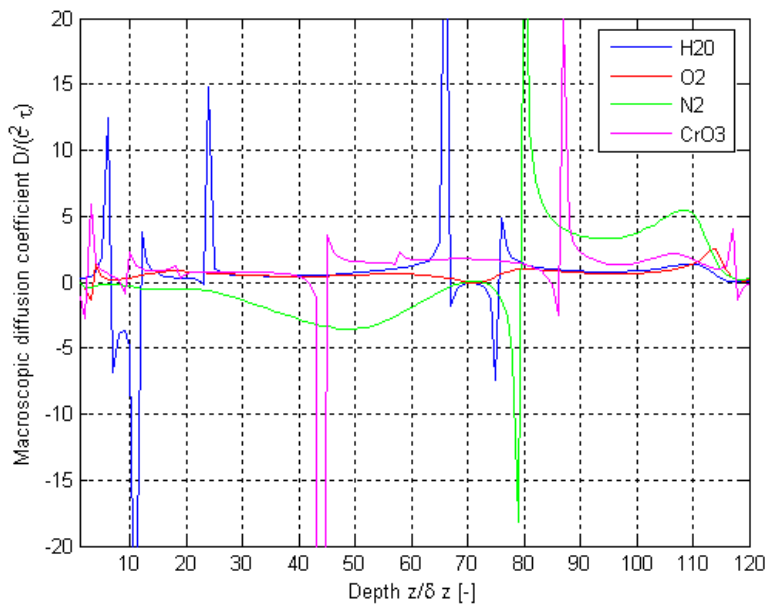


Figure E.96. Results of this thesis work for variations as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for τ .

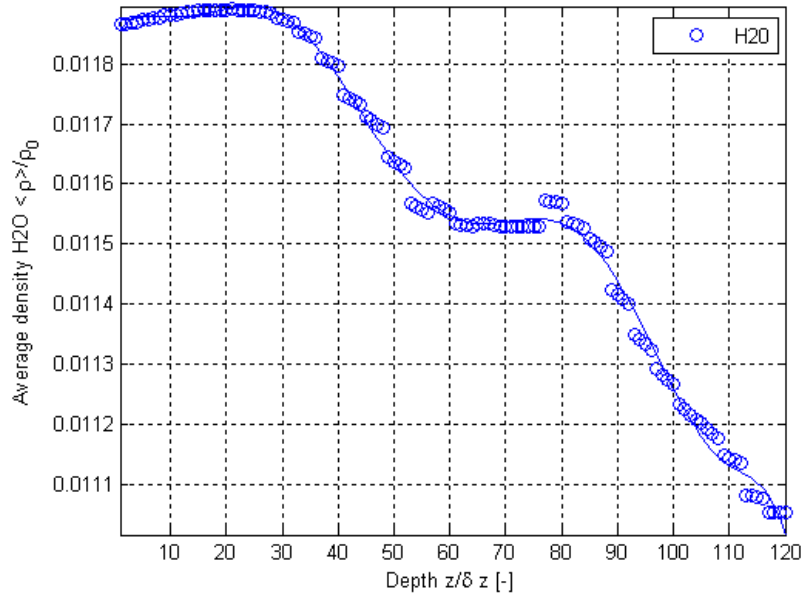


Figure E.97. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂, N₂ and CrO₃ mixture for H₂O.

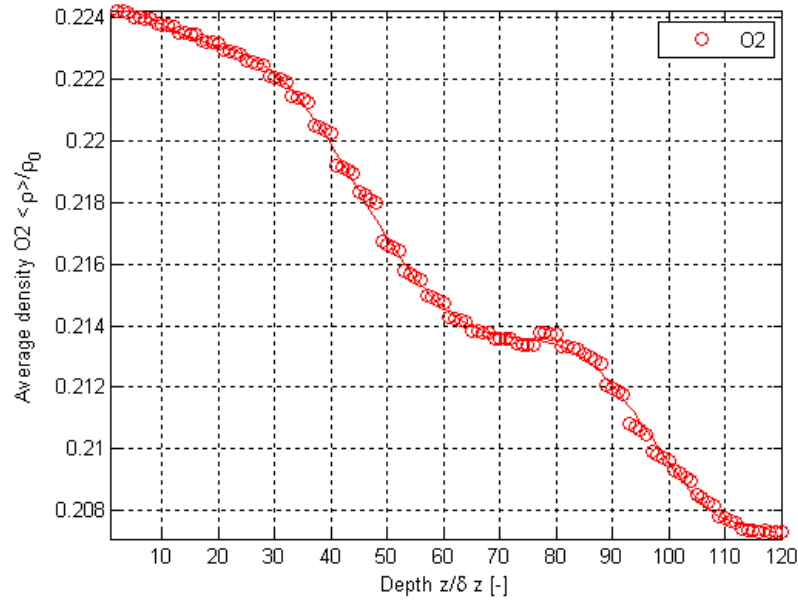


Figure E.98. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂, N₂ and CrO₃ mixture for O₂.

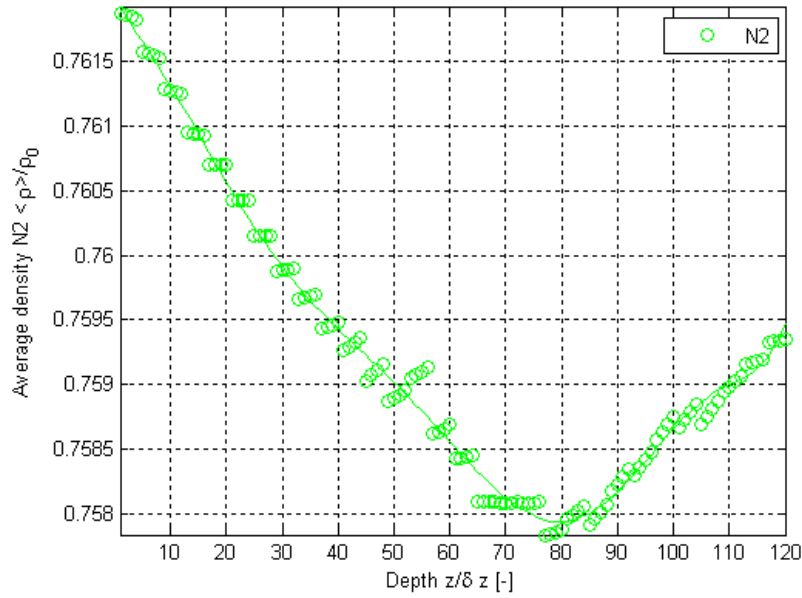


Figure E.99. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for N_2 .

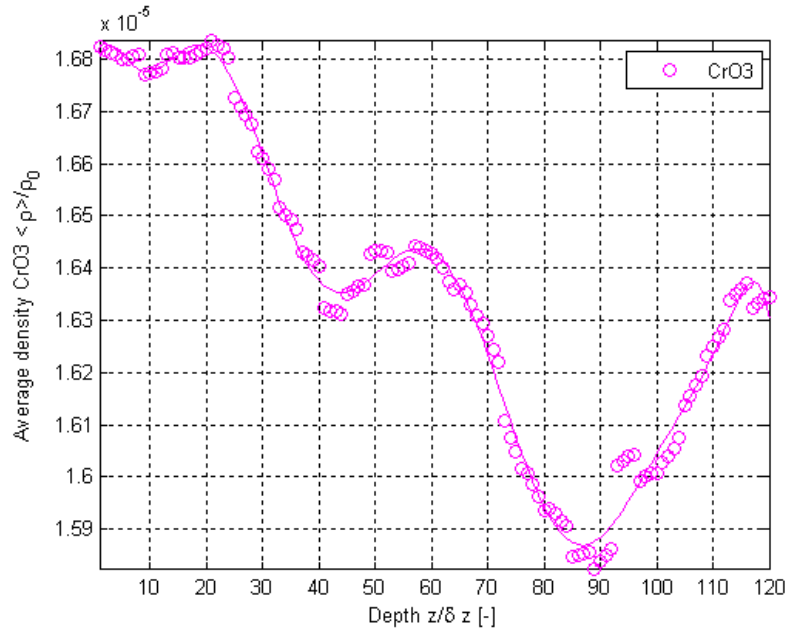


Figure E.100. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 .

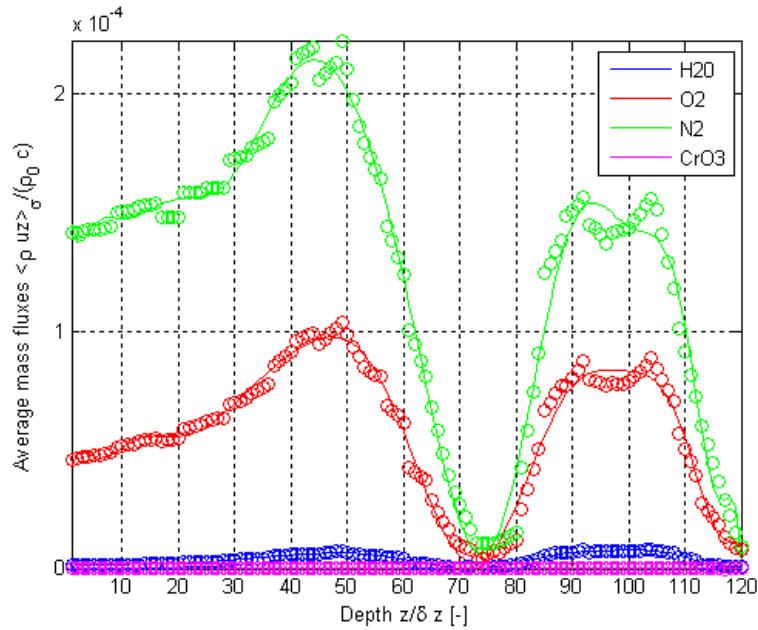


Figure E.101. Results of this thesis work for variations as a function of depth using Geometry B for the H₂O, O₂, N₂ and CrO₃ mixture for the normalized surface-averaged mass flux.

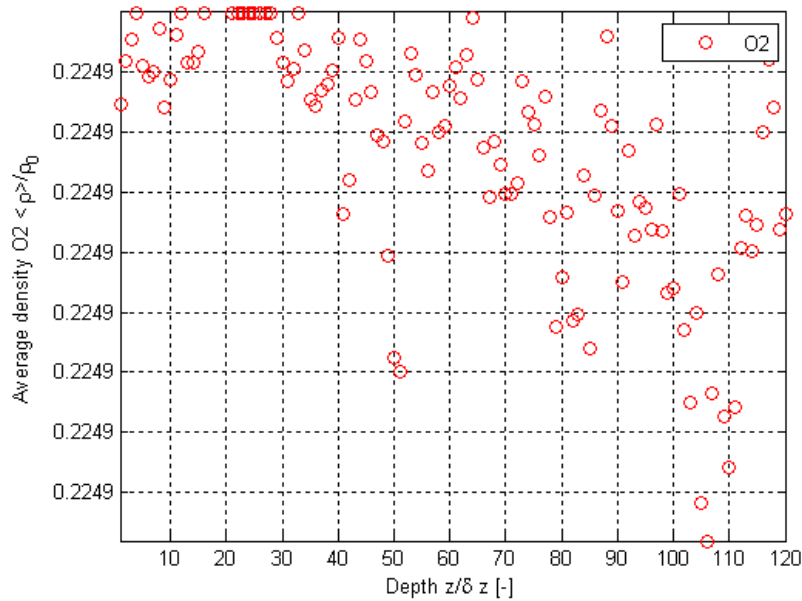


Figure E.102. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂, N₂ and CrO₃ mixture for O₂ after 10 collisions.

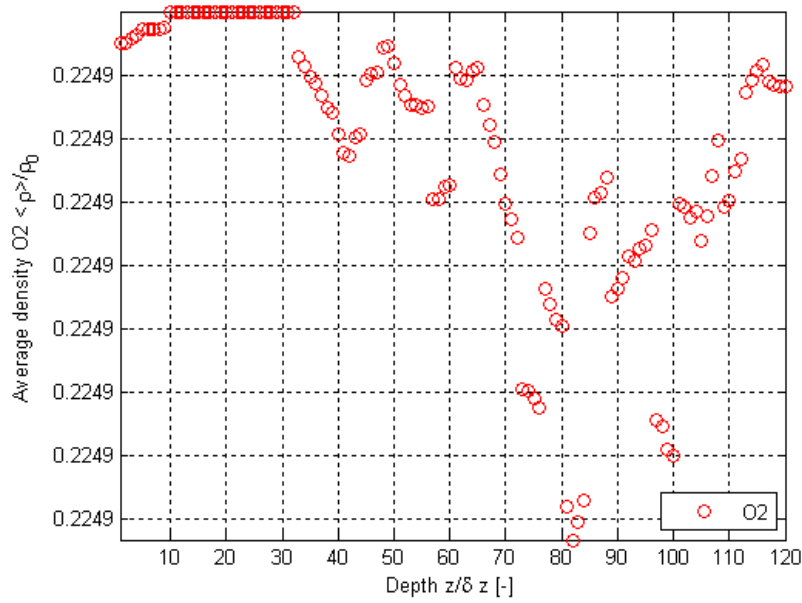


Figure E.103. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after 50 collisions.

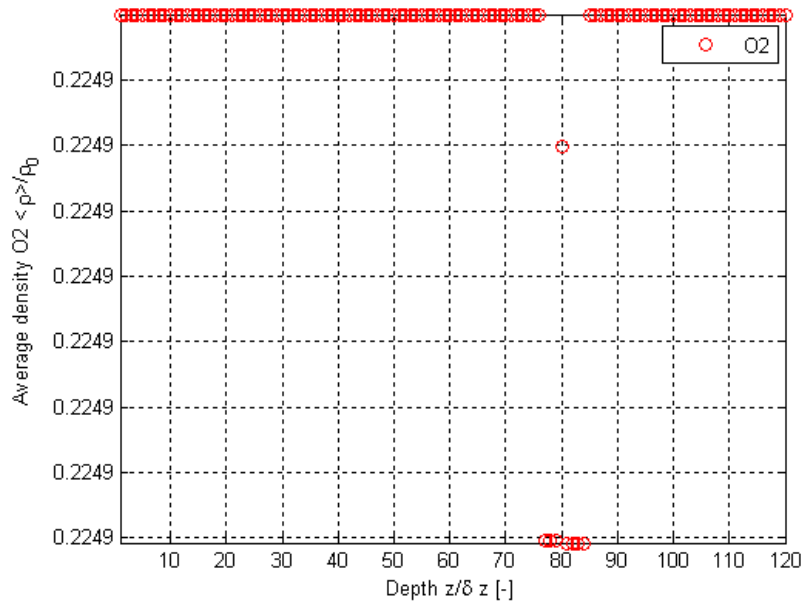


Figure E.104. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after 500 collisions.

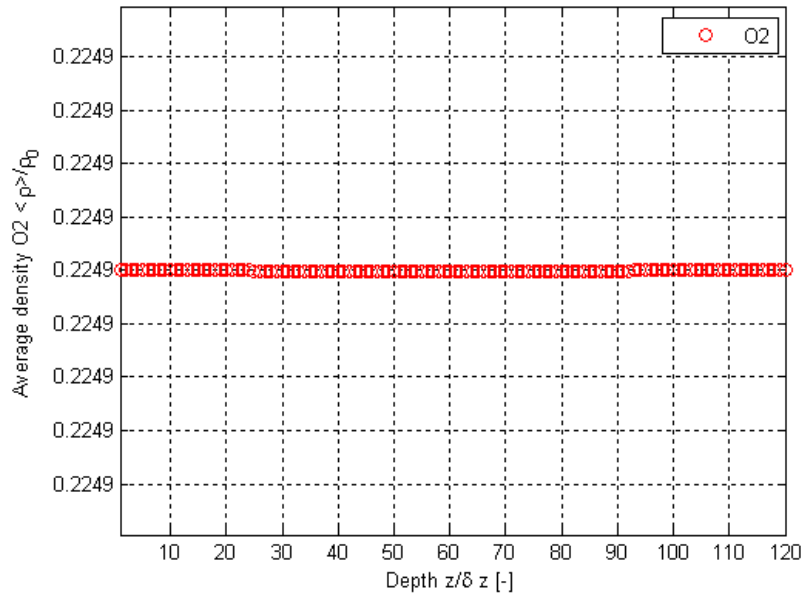


Figure E.105. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂, N₂ and CrO₃ mixture for O₂ after 1000 collisions.

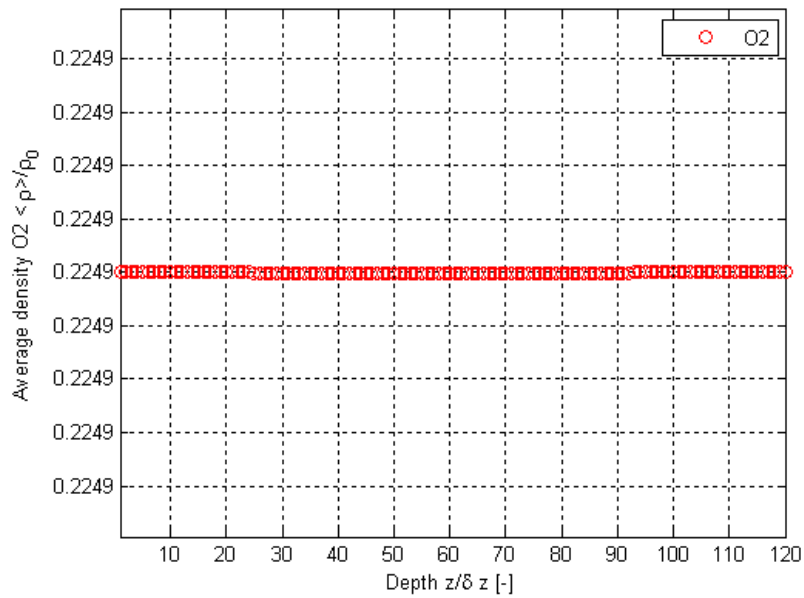


Figure E.106. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H₂O, O₂, N₂ and CrO₃ mixture for O₂ after 3000 collisions.

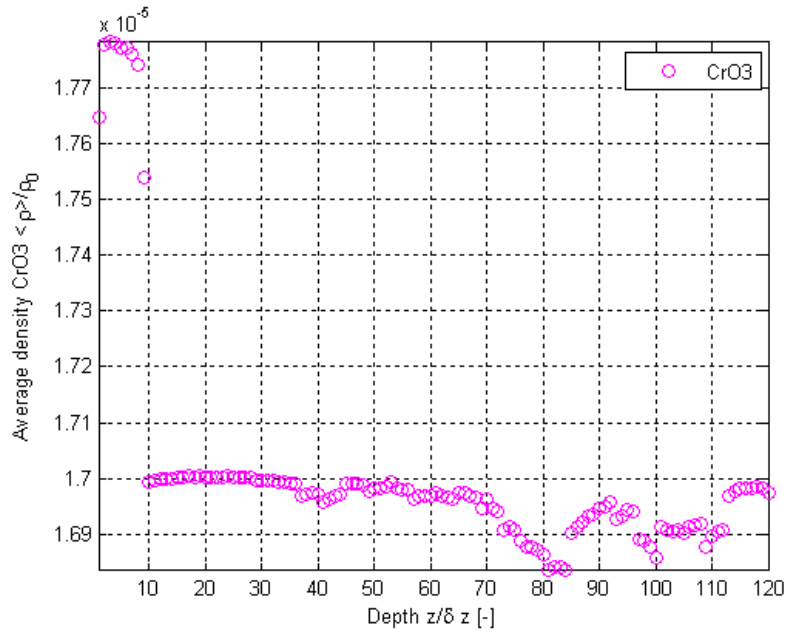


Figure E.107. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 10 collisions.

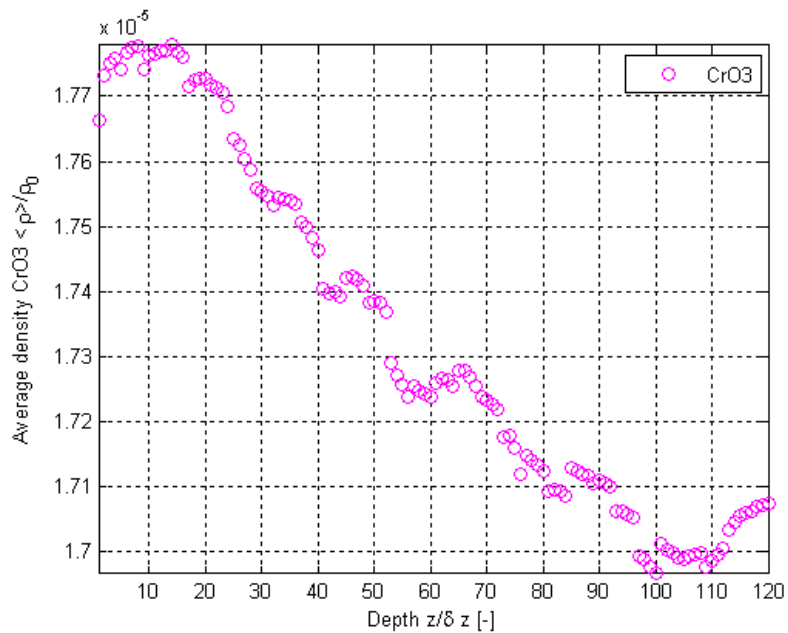


Figure E.108. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 50 collisions.

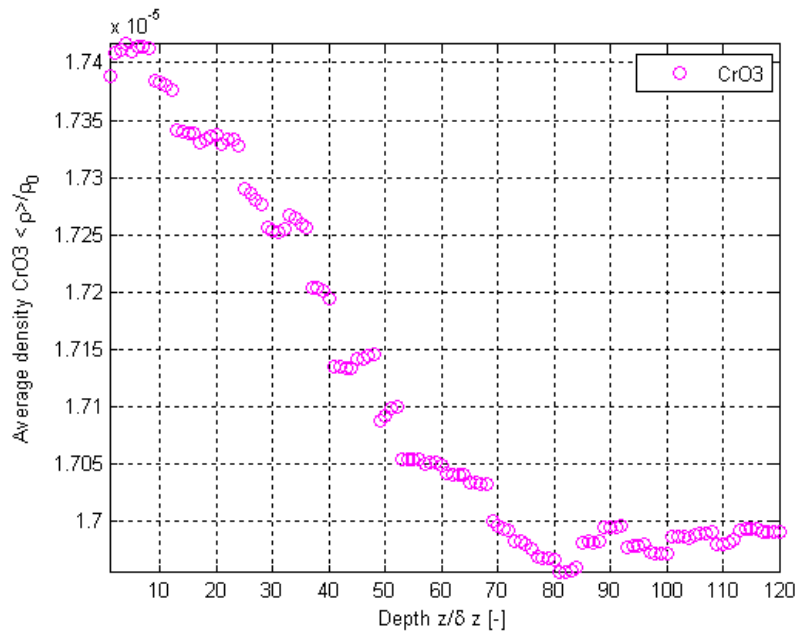


Figure E.109. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 500 collisions.

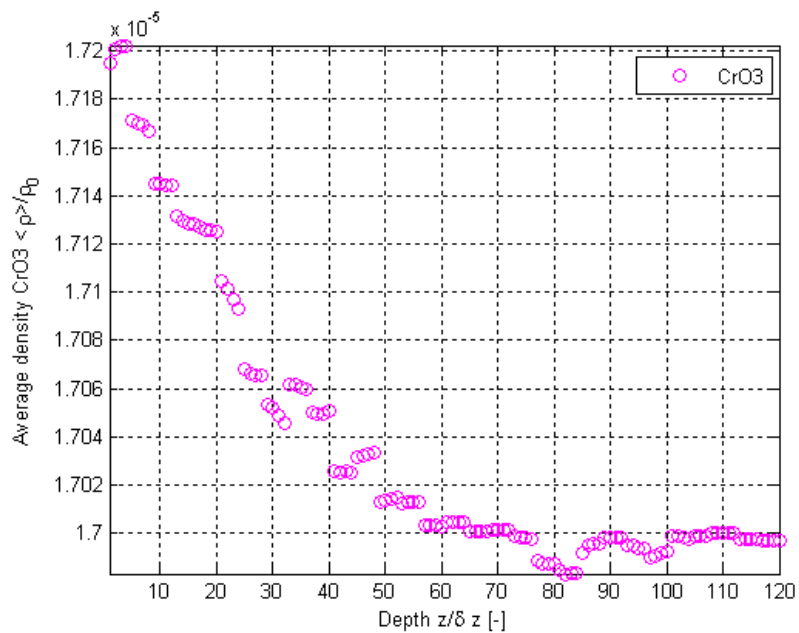


Figure E.110. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 1000 collisions.

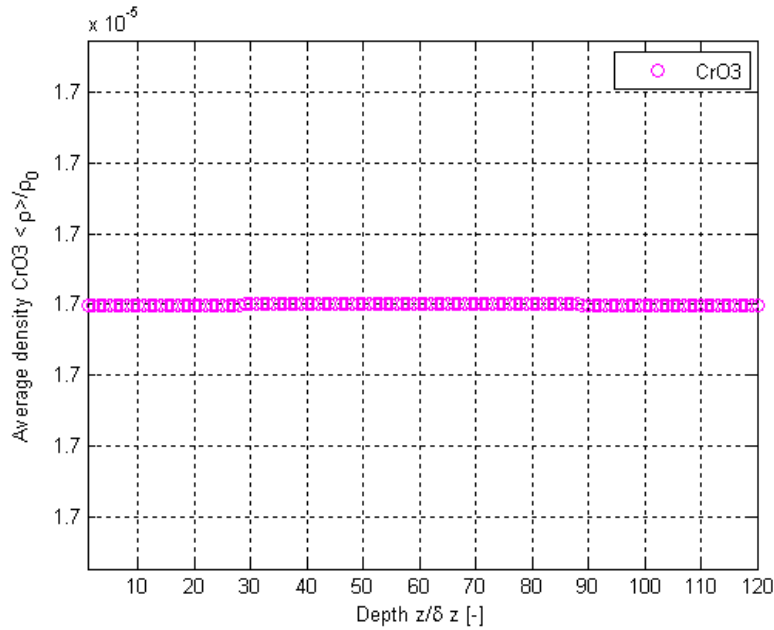


Figure E.111. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry B for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 3000 collisions.

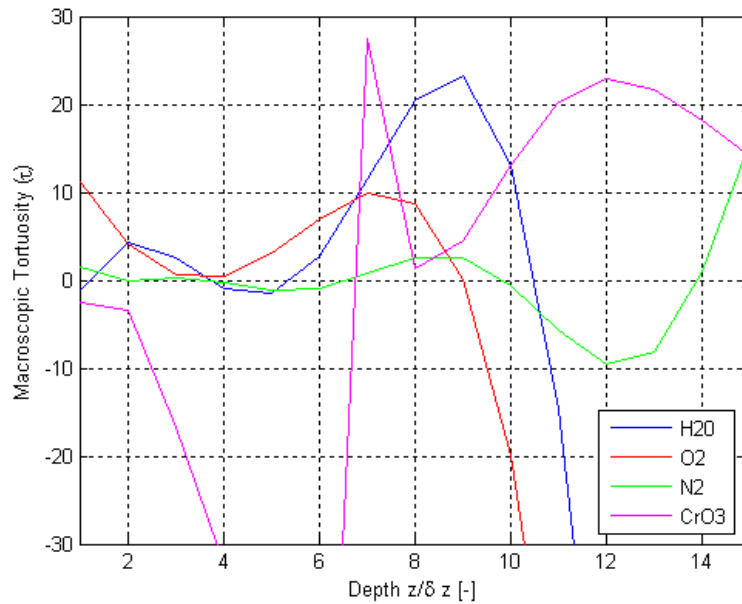


Figure E.112. Results of this thesis work for variations as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for τ .

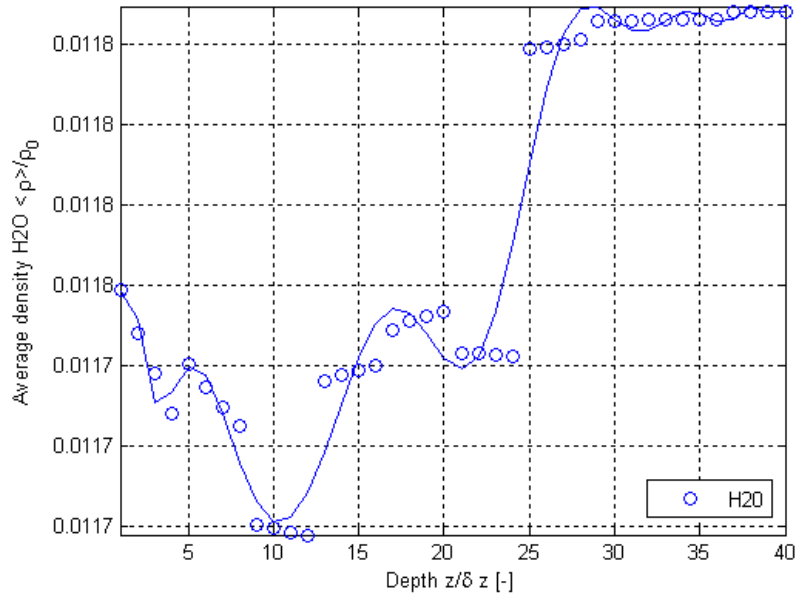


Figure E.113. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for H₂O.

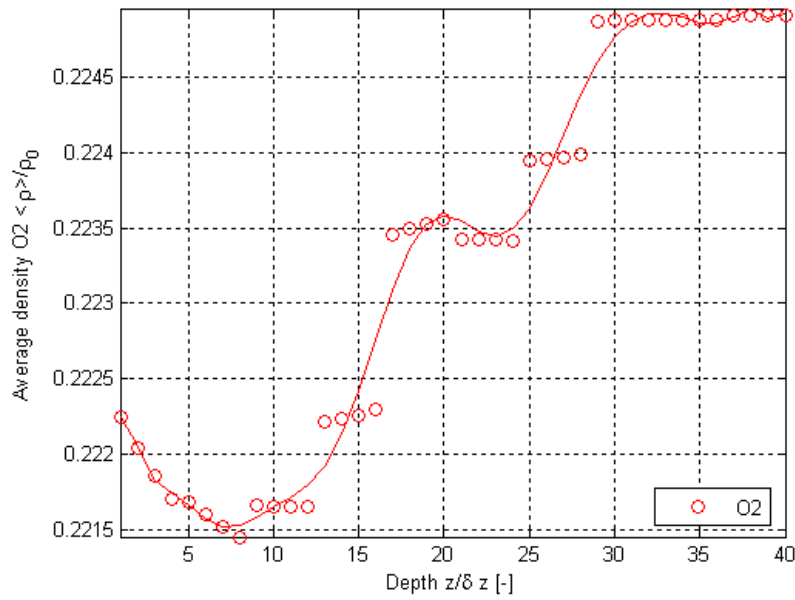


Figure E.114. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for O₂.

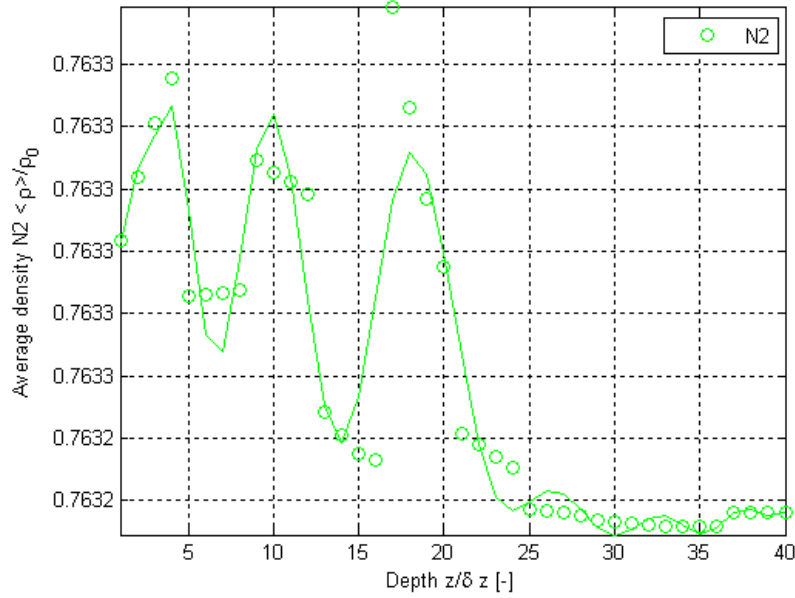


Figure E.115. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for N₂.

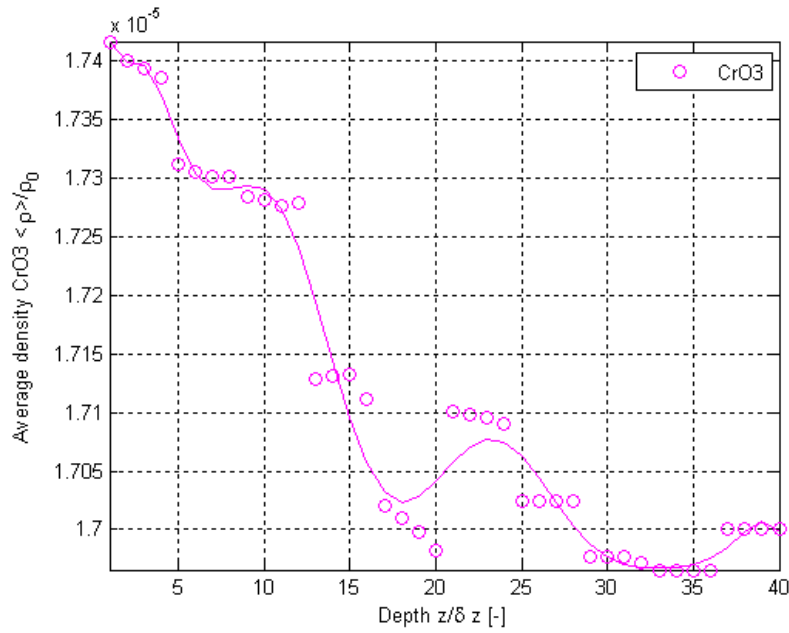


Figure E.116. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for CrO₃.

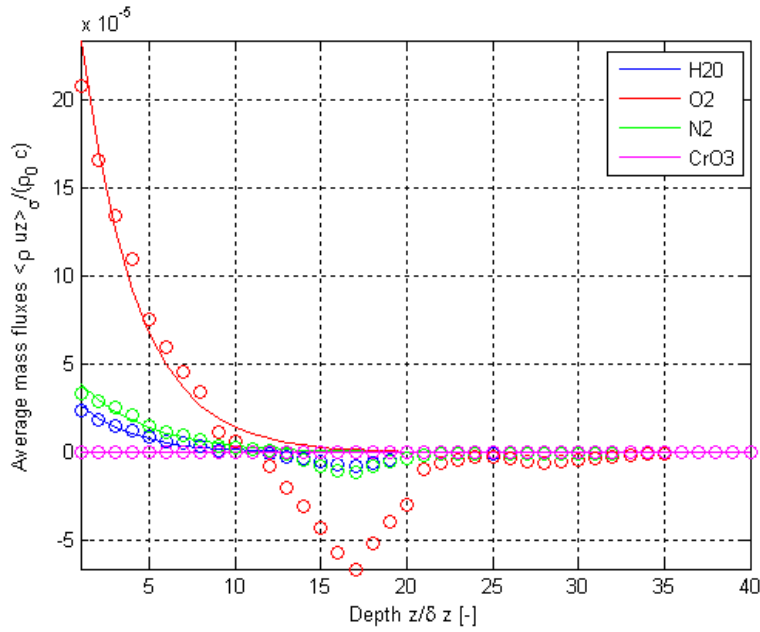


Figure E.117. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for the normalized surface-averaged mass flux.

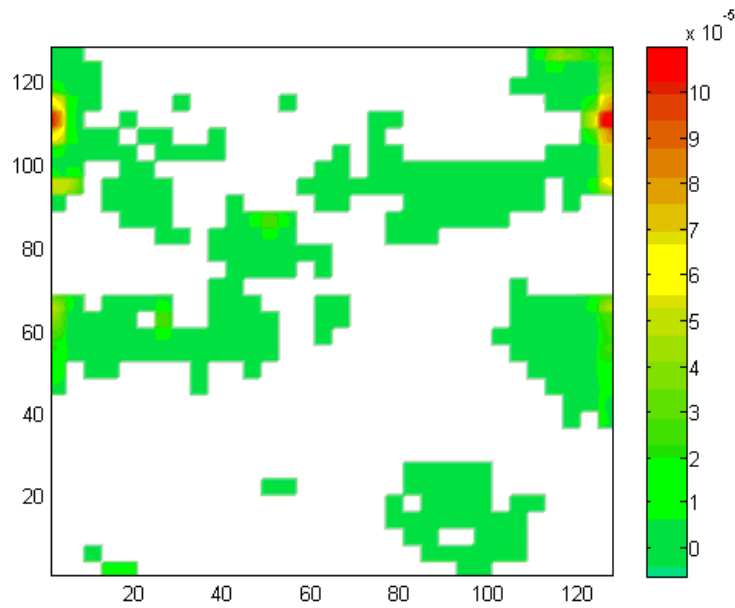


Figure E.118. Normalized mass flux contours for H₂O using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for the inlet/outlet plane.

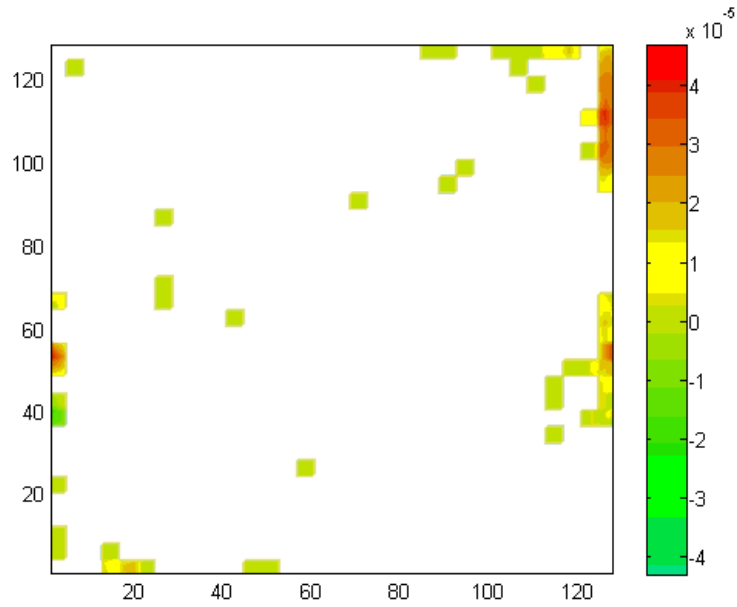


Figure E.119. Normalized mass flux contours for H₂O using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for an intermediate plane.

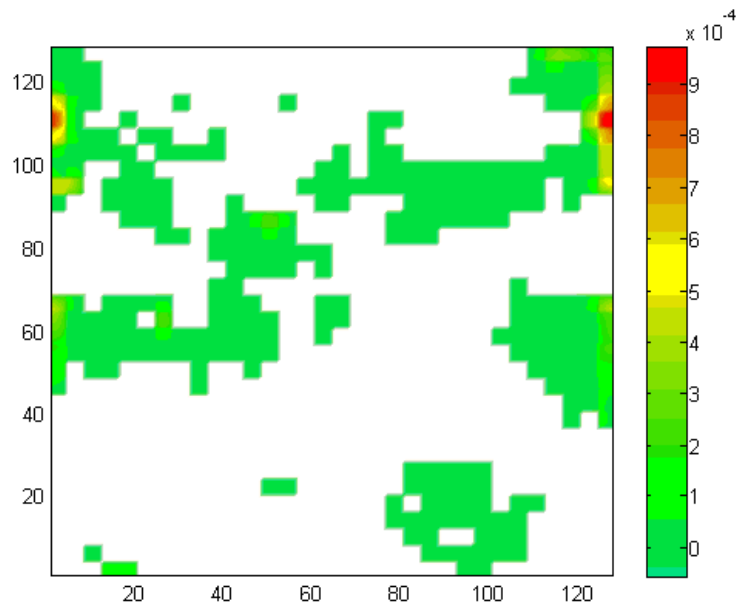


Figure E.120. Normalized mass flux contours for O₂ using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for the inlet/outlet plane.

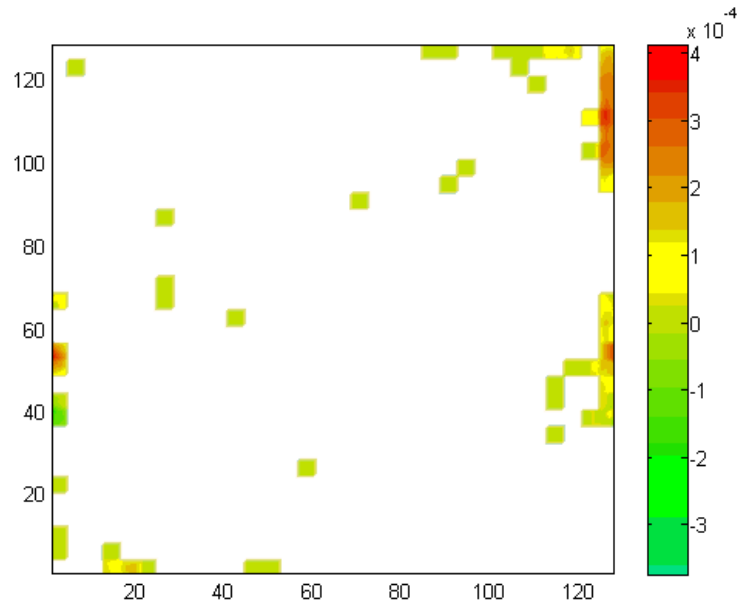


Figure E.121. Normalized mass flux contours for O₂ using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for an intermediate plane.

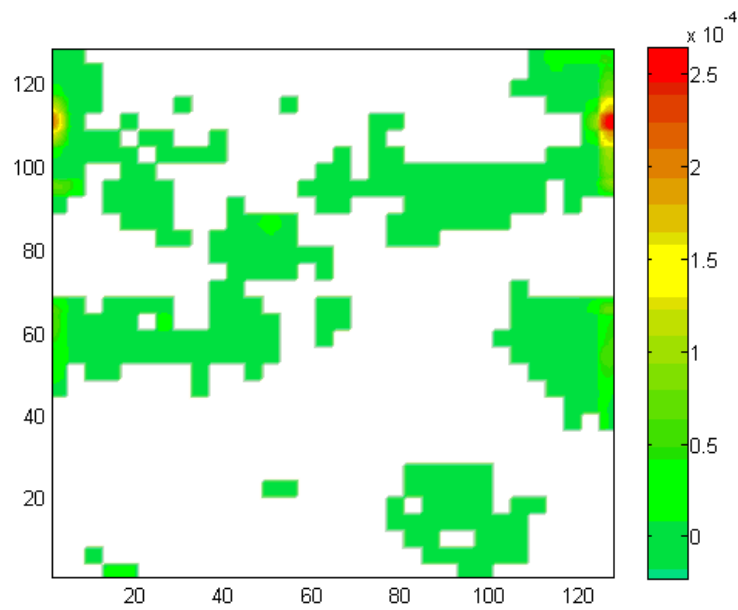


Figure E.122. Normalized mass flux contours for N₂ using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for the inlet/outlet plane.

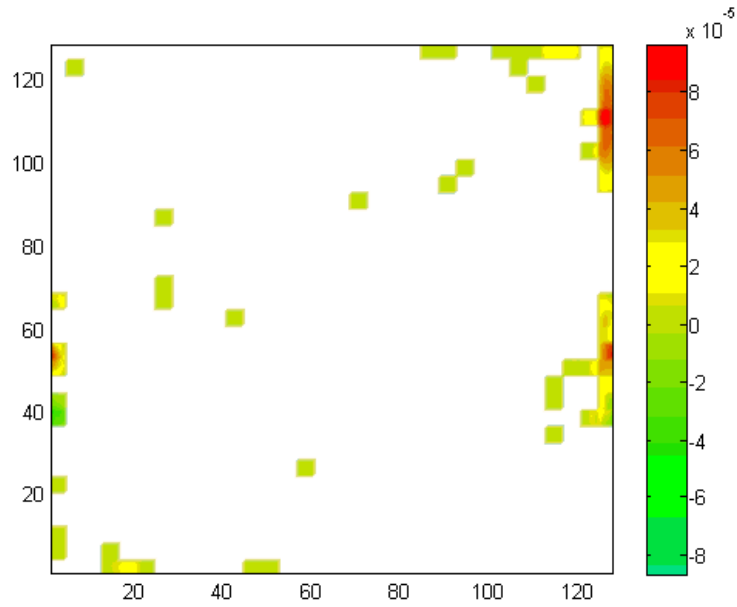


Figure E.123. Normalized mass flux contours for N₂ using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for an intermediate plane.

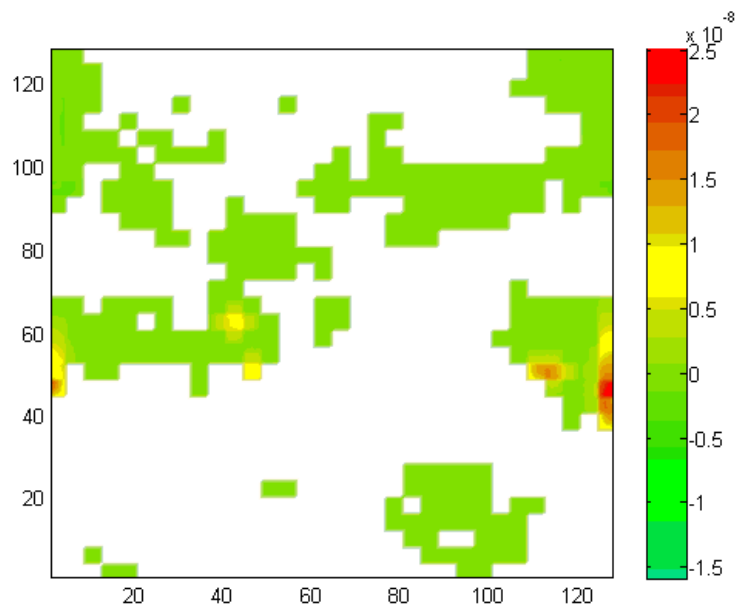


Figure E.124. Normalized mass flux contours for CrO₃ using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for the inlet/outlet plane.

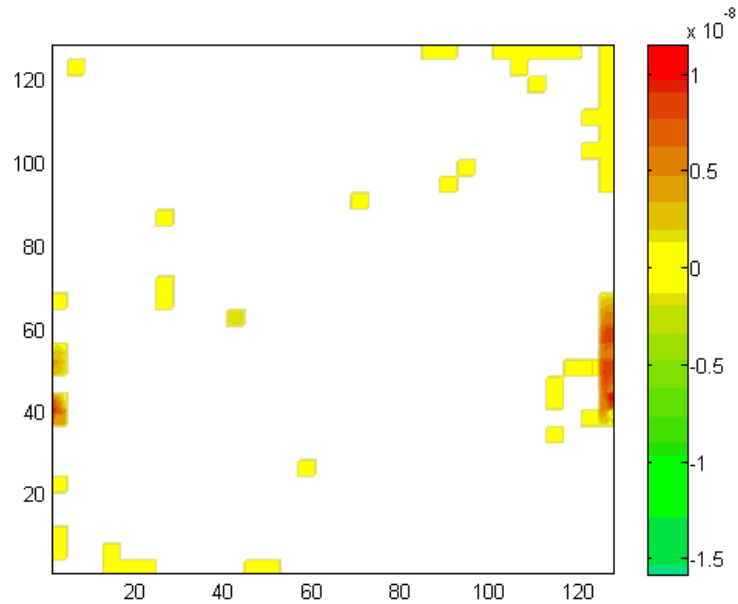


Figure E.125. Normalized mass flux contours for CrO_3 using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for an intermediate plane.

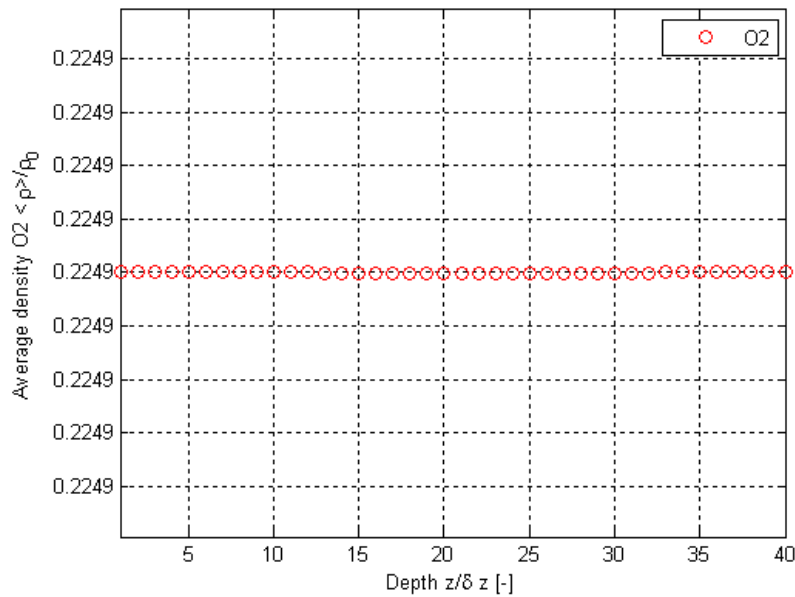


Figure E.126. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after 10 collisions.

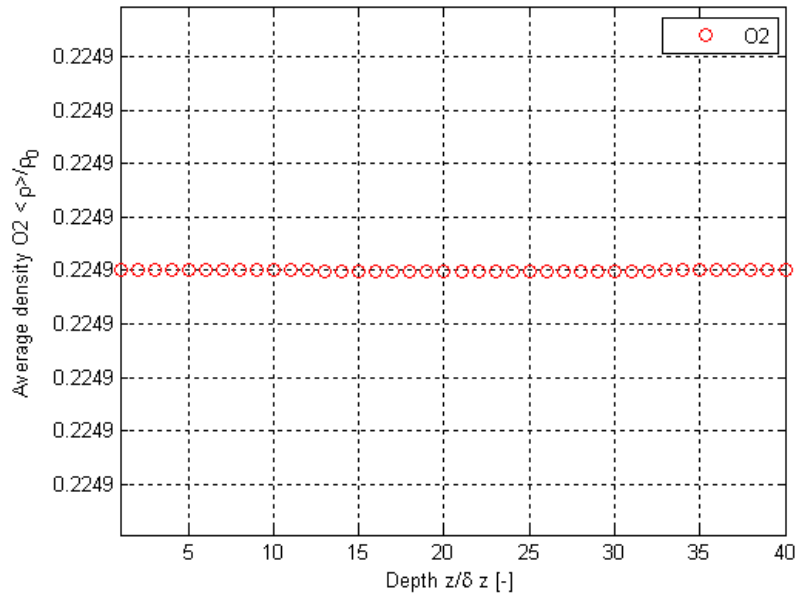


Figure E.127. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after 50 collisions.

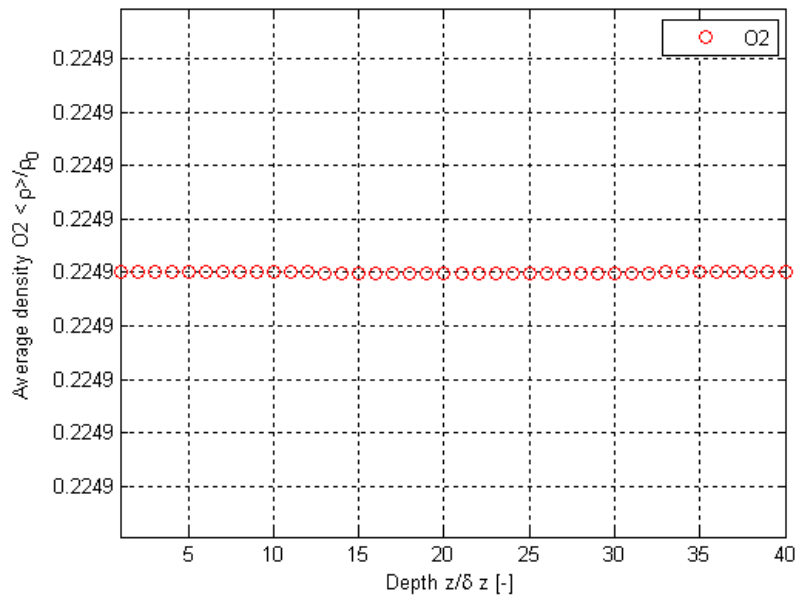


Figure E.128. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after 500 collisions.

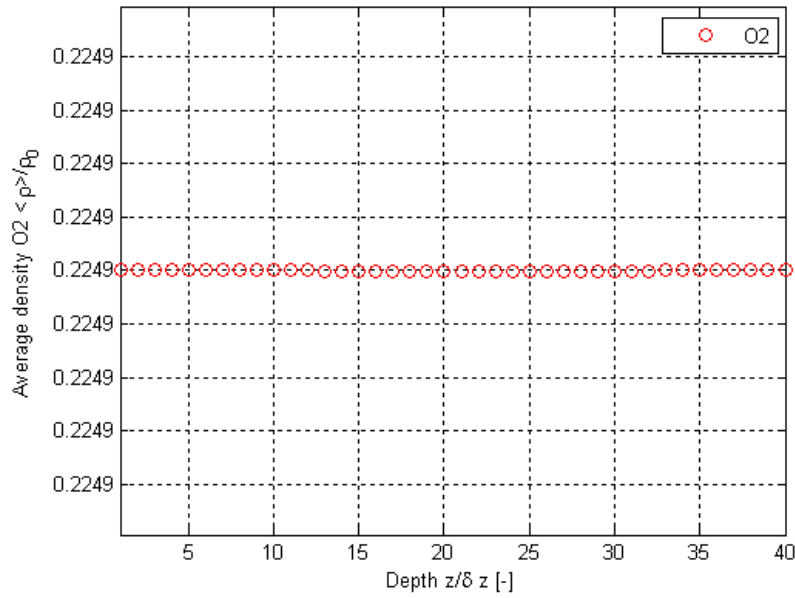


Figure E.129. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after 1000 collisions.

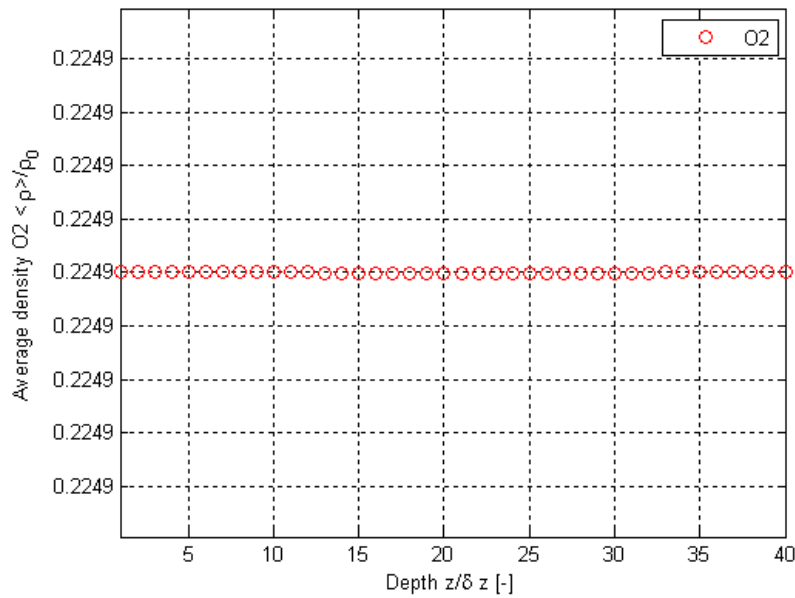


Figure E.130. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after 3000 collisions.

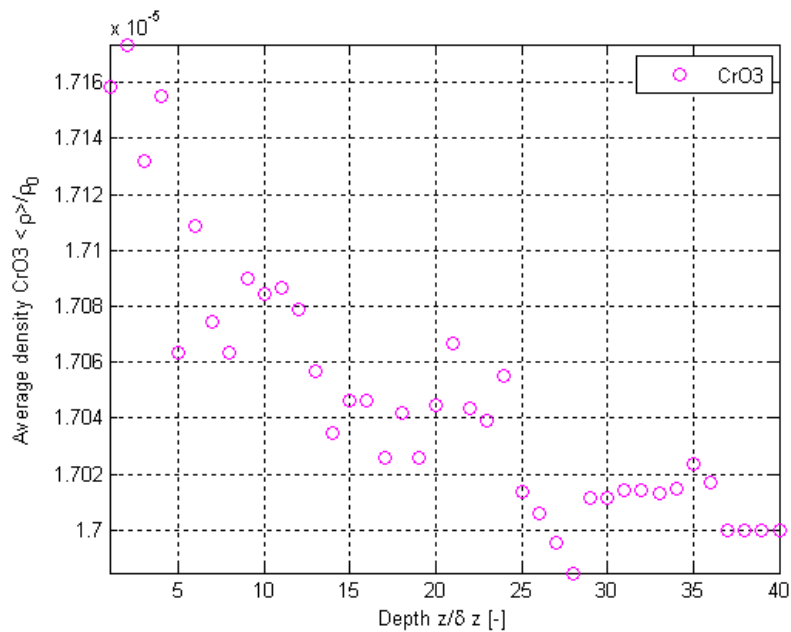


Figure E.131. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for CrO₃ after 10 collisions.

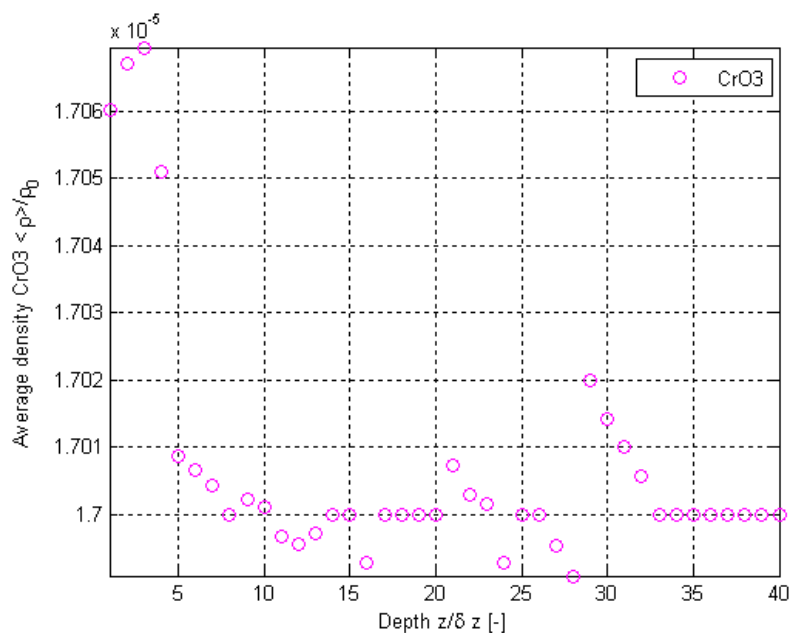


Figure E.132. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for CrO₃ after 50 collisions.

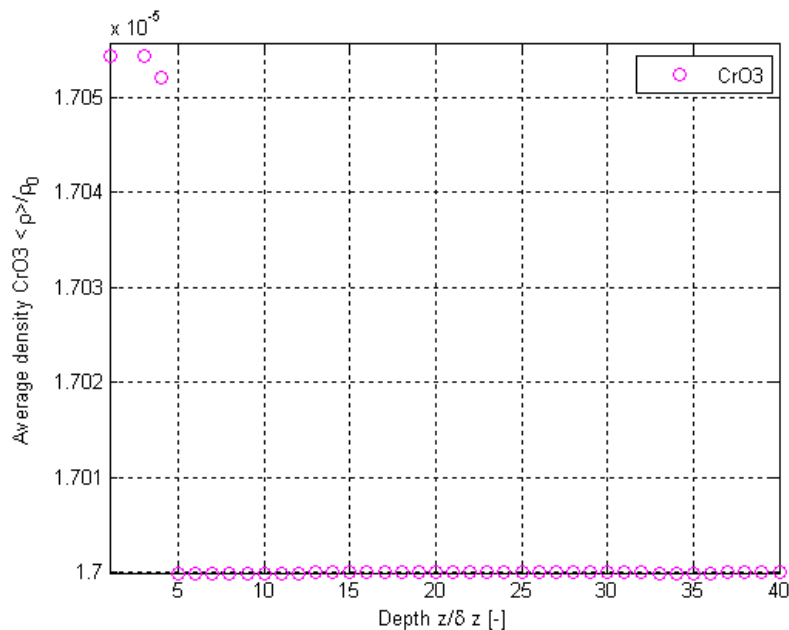


Figure E.133. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 500 collisions.

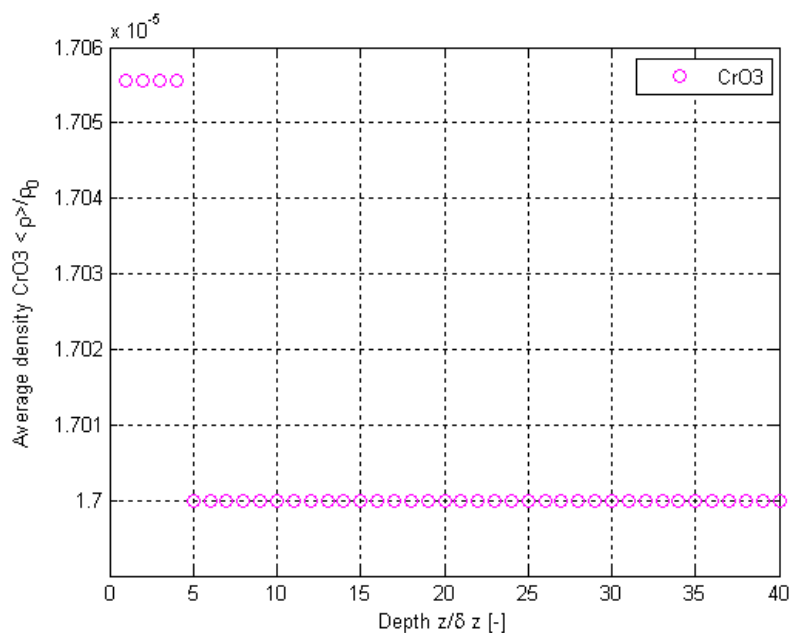


Figure E.134. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 1000 collisions.

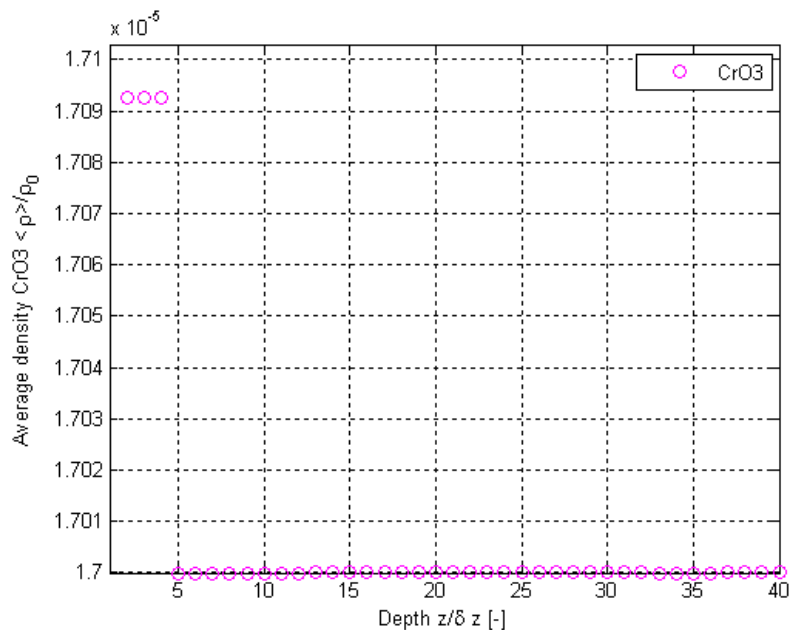


Figure E.135. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 3000 collisions.

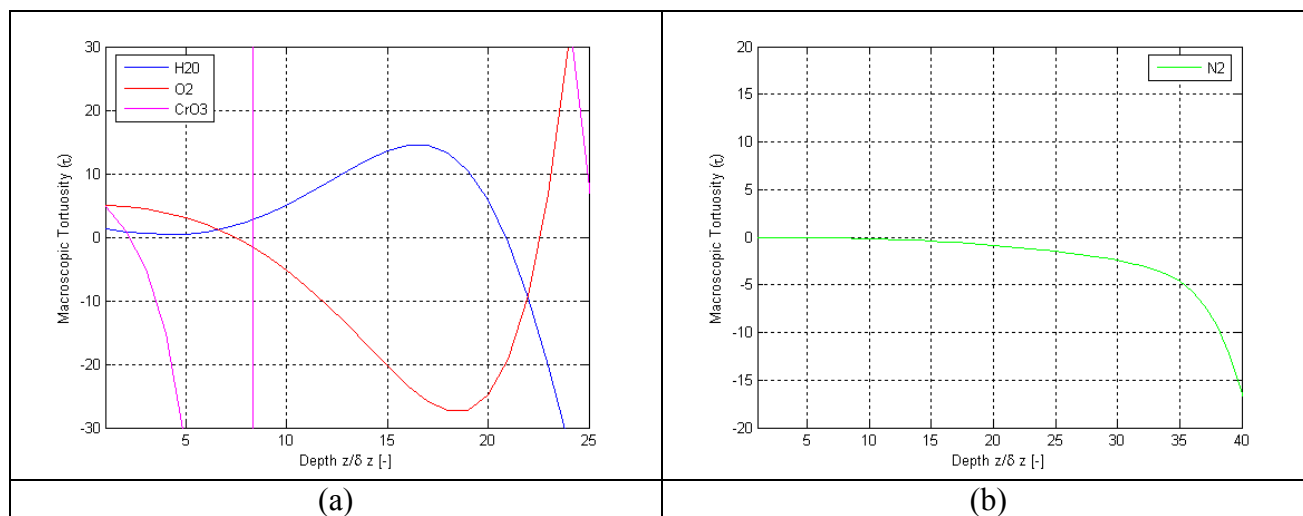


Figure E.136. Results of this thesis work for variations as a function of depth using Geometry D for the H_2O , O_2 , N_2 and CrO_3 mixture for τ for (a) H_2O , O_2 and CrO_3 (b) N_2 .

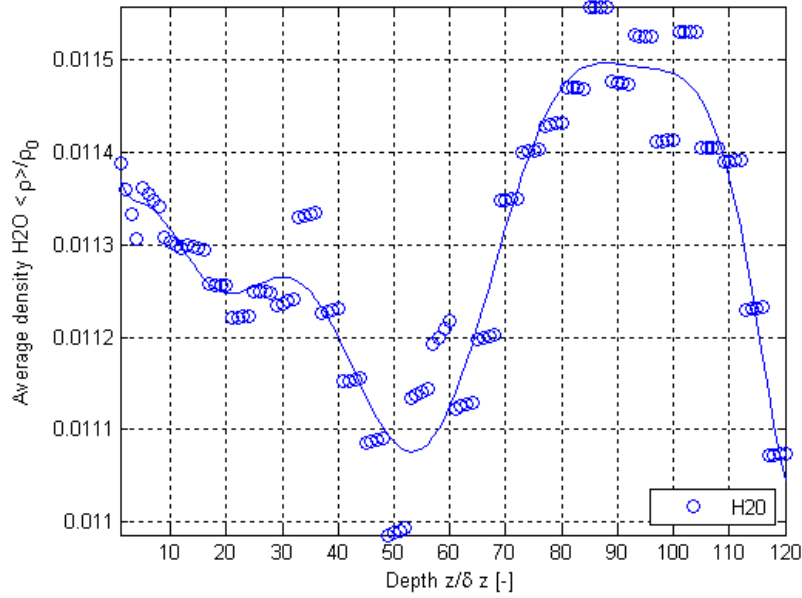


Figure E.137. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H₂O, O₂, N₂ and CrO₃ mixture for H₂O.

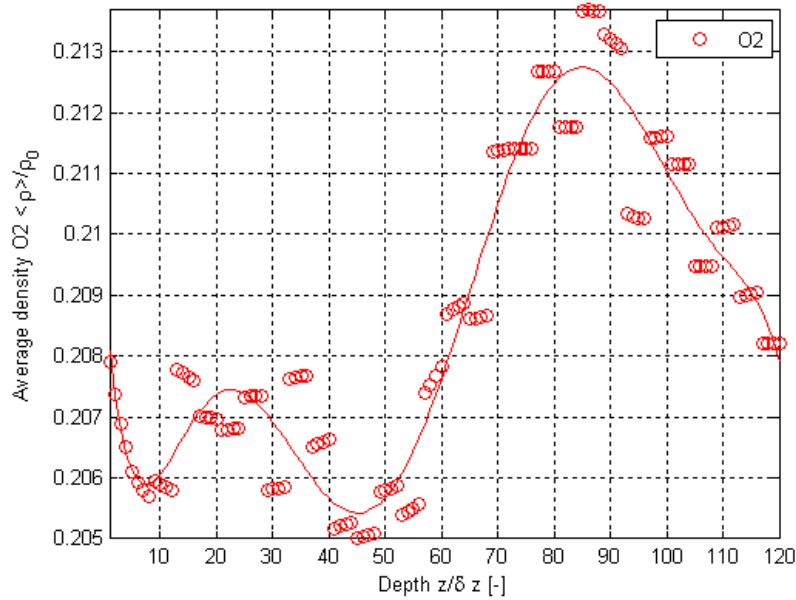


Figure E.138. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H₂O, O₂, N₂ and CrO₃ mixture for O₂.

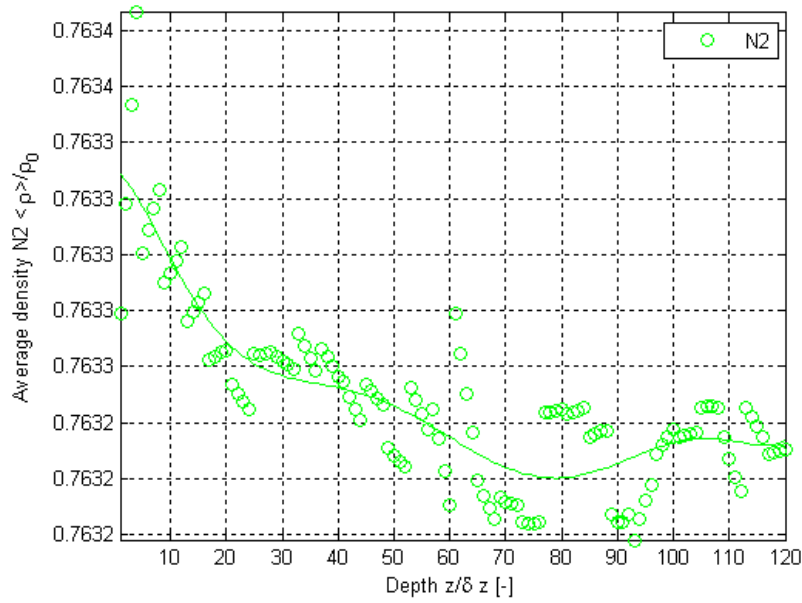


Figure E.139. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H_2O , O_2 , N_2 and CrO_3 mixture for N_2 .

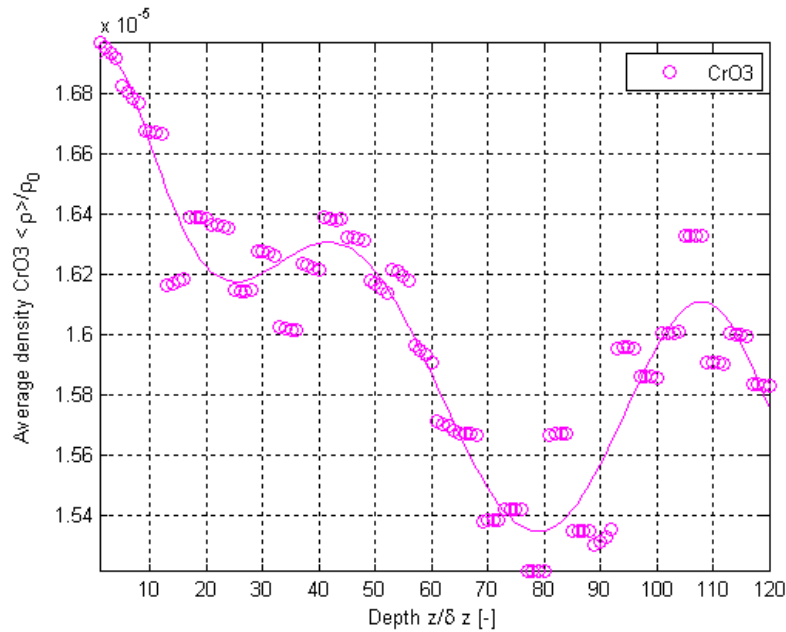


Figure E.140. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 .

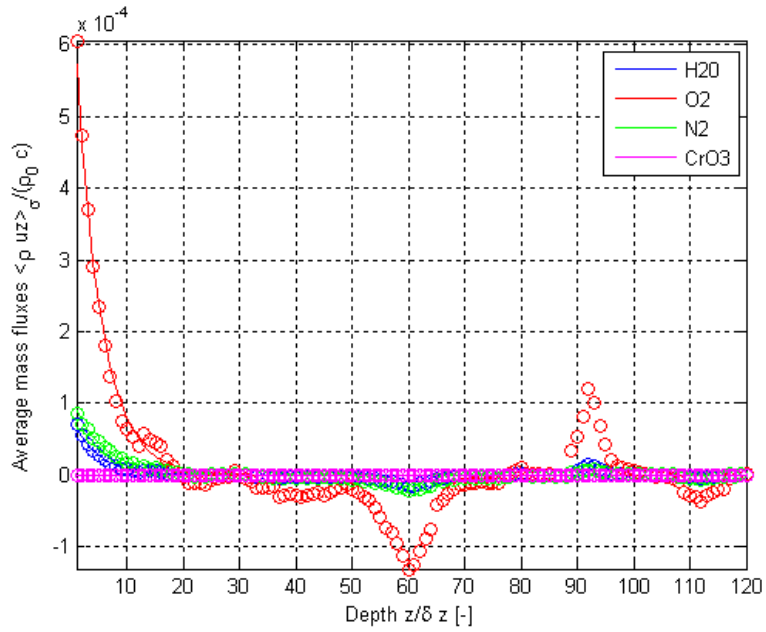


Figure E.141. Results of this thesis work for variations as a function of depth using Geometry D for the H_2O , O_2 , N_2 and CrO_3 mixture for the normalized surface-averaged mass flux.

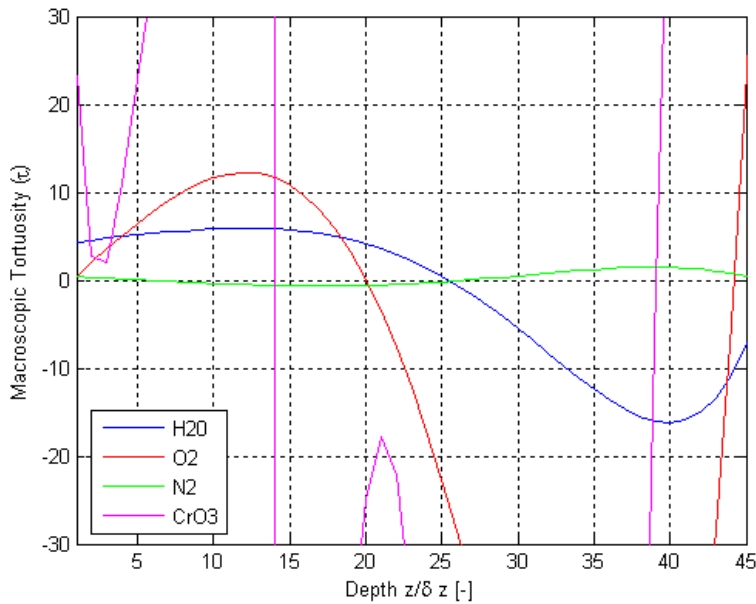


Figure E.142. Results of this thesis work for variations as a function of depth using Geometry E for the H_2O , O_2 , N_2 and CrO_3 mixture for τ .

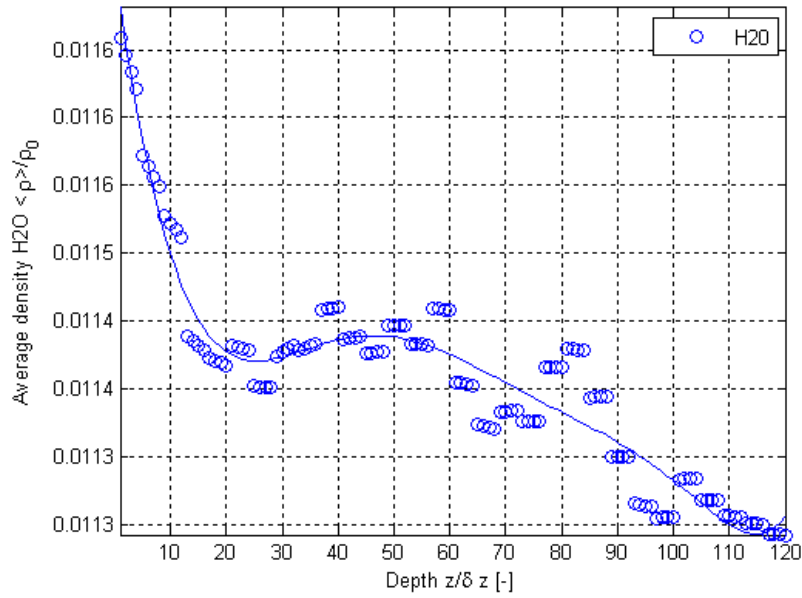


Figure E.143. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry E for the H_2O , O_2 , N_2 and CrO_3 mixture for H_2O .

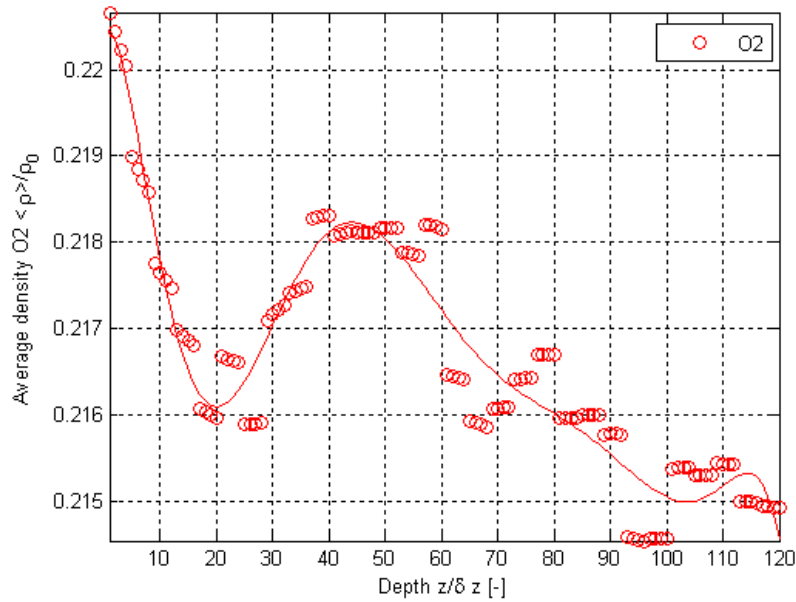


Figure E.144. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry E for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 .

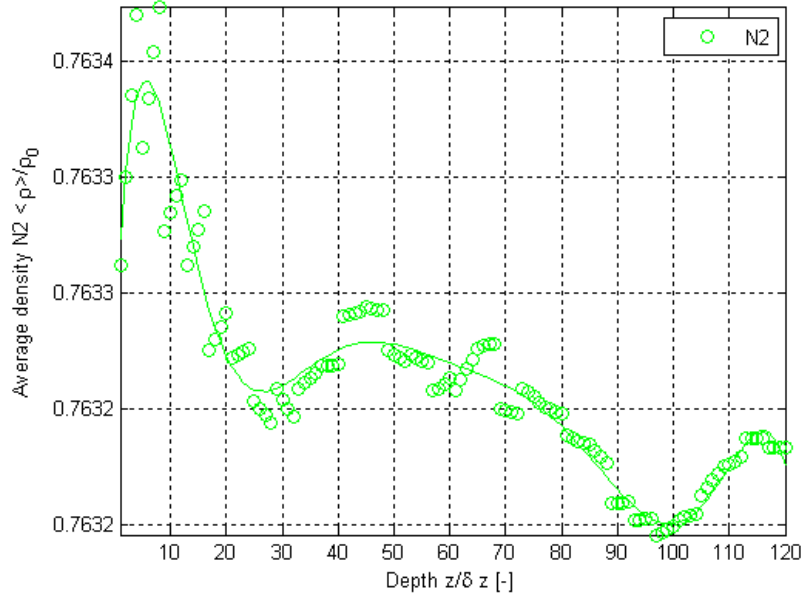


Figure E.145 Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry E for the H_2O , O_2 , N_2 and CrO_3 mixture for N_2 .

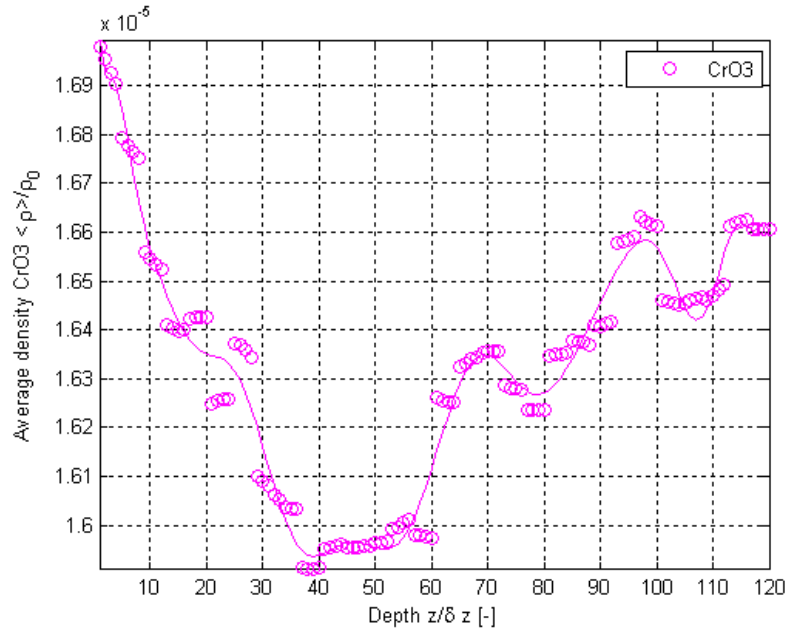


Figure E.146. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry E for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 .

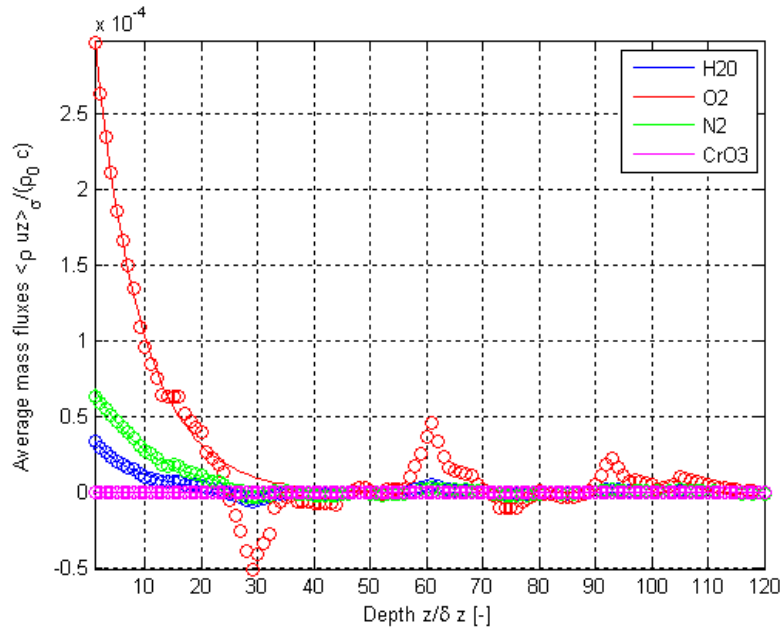


Figure E.147. Results of this thesis work for variations as a function of depth using Geometry E for the H_2O , O_2 , N_2 and CrO_3 mixture for the normalized surface-averaged mass flux.

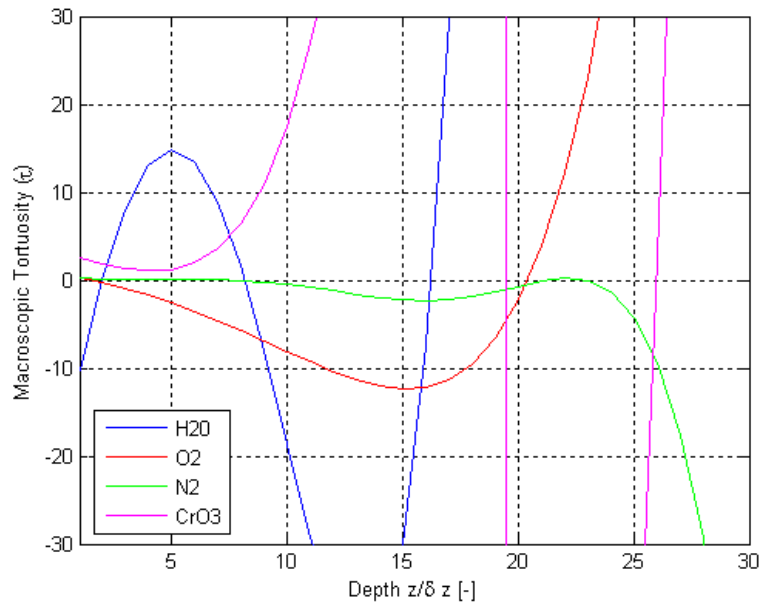


Figure E.148. Results of this thesis work for variations as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for τ .

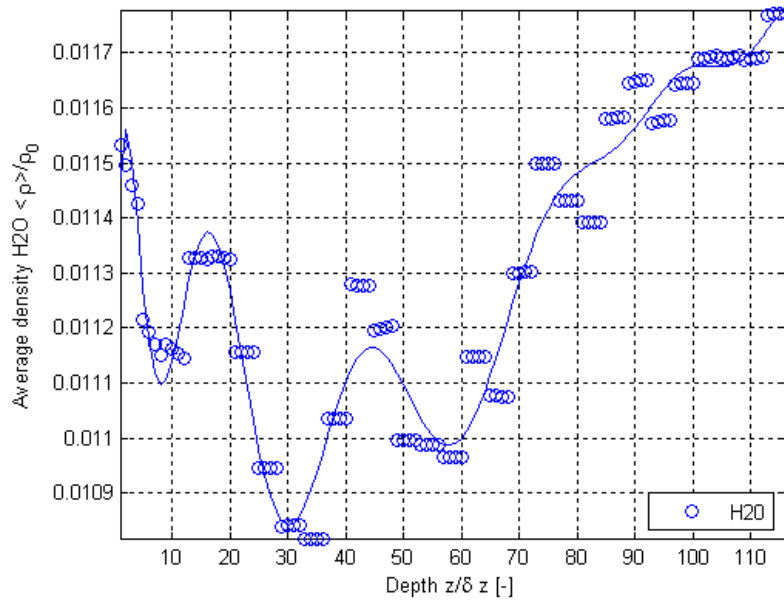


Figure E.149. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for H₂O.

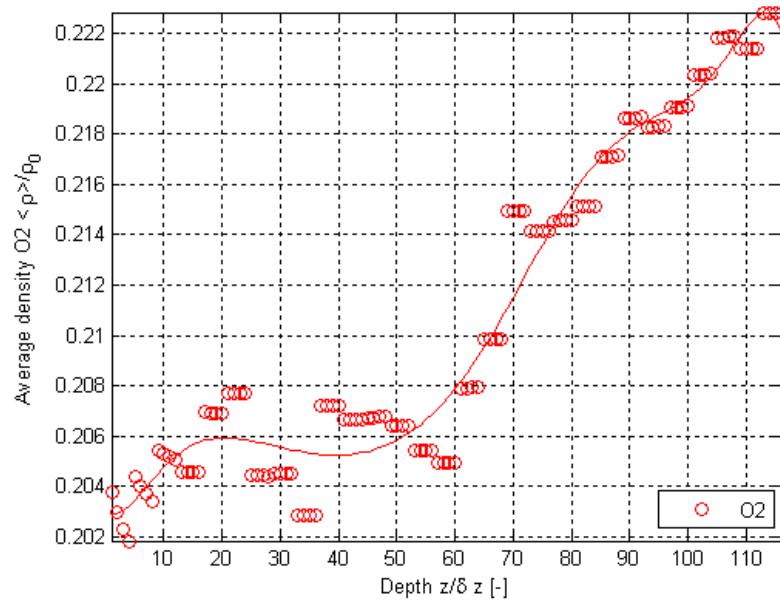


Figure E.150. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for O₂.

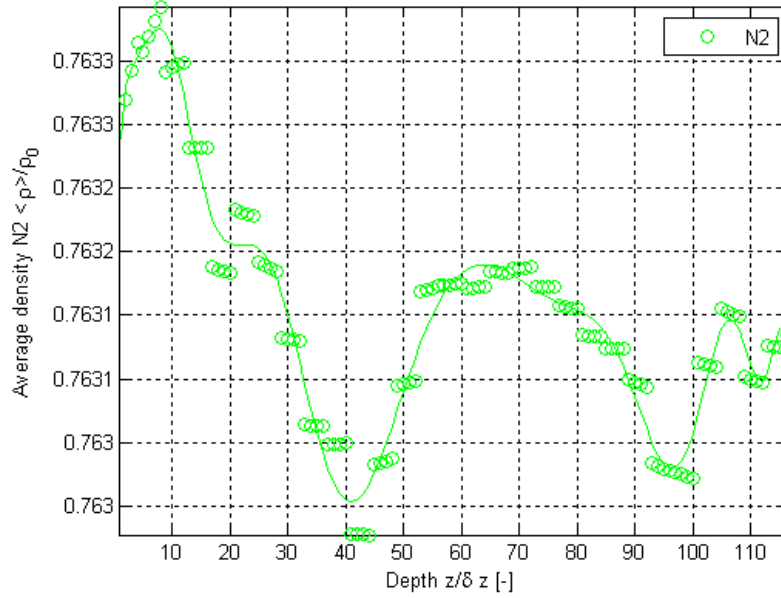


Figure E.151. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for N_2 .

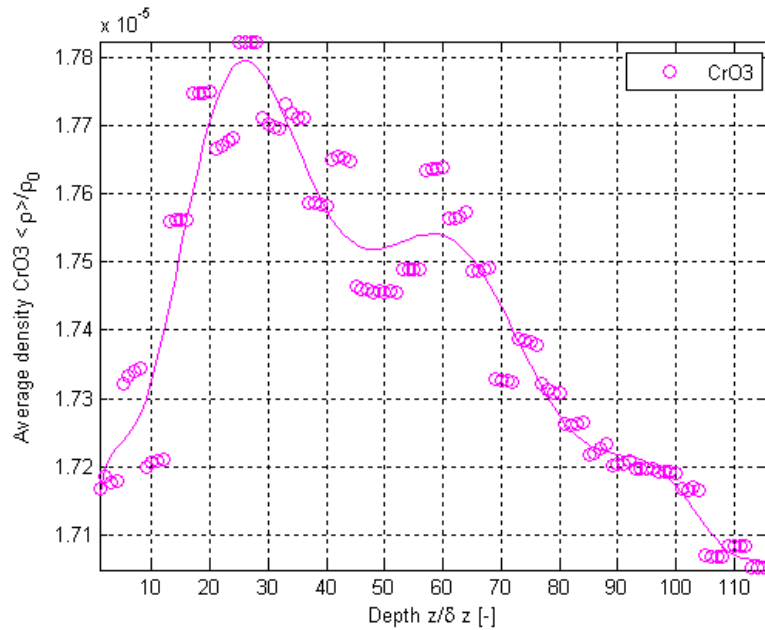


Figure E.152. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 .

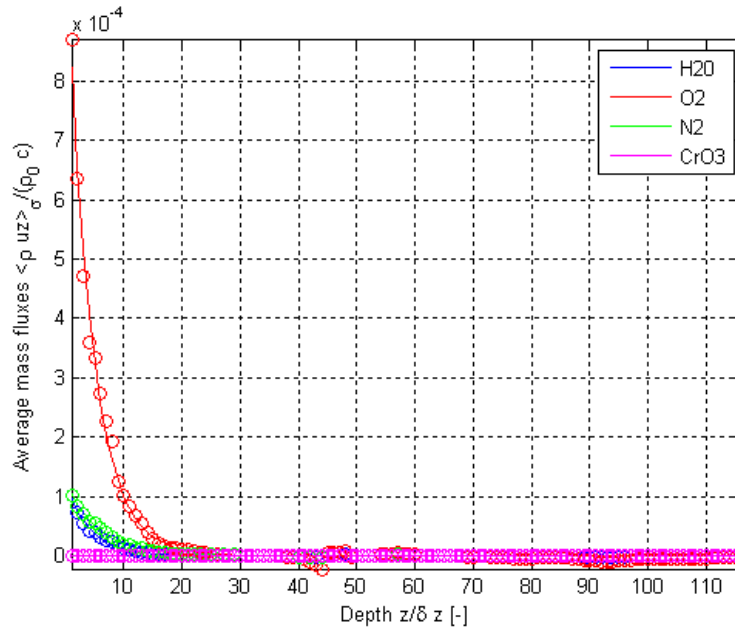


Figure E.153. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for the normalized surface-averaged mass flux.

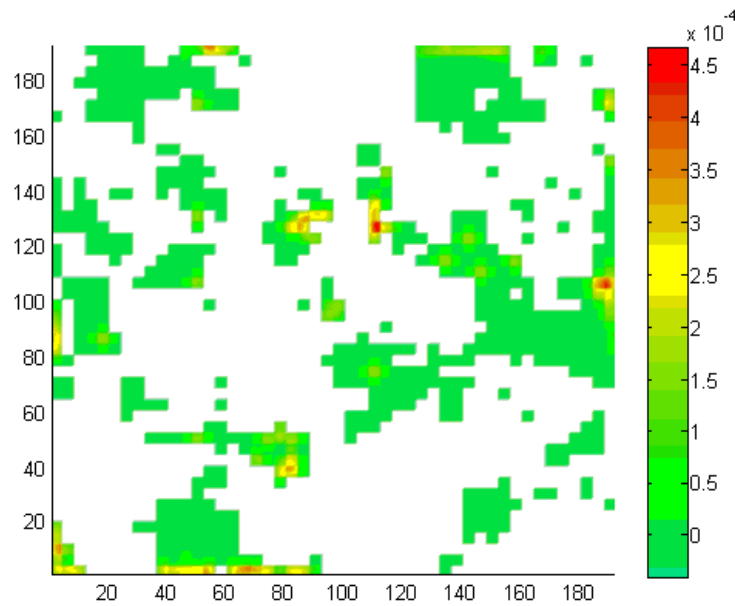


Figure E.154. Normalized mass flux contours for H₂O using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for the inlet/outlet plane.

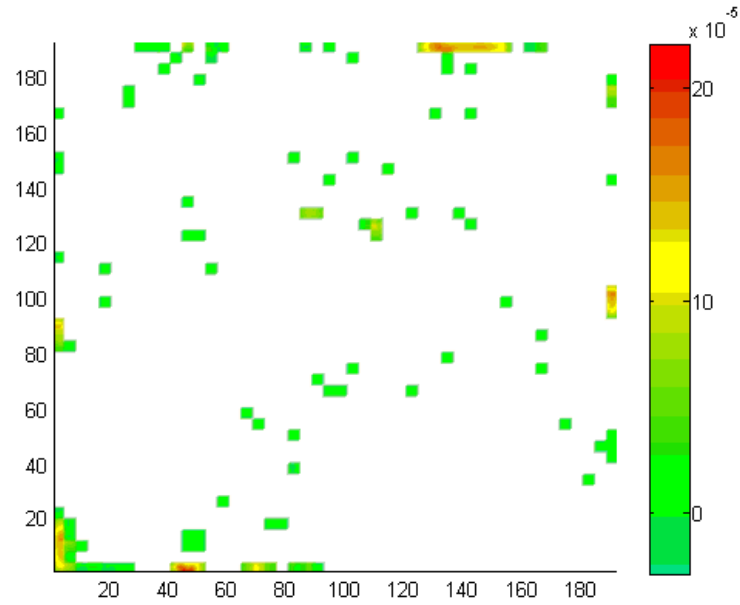


Figure E.155. Normalized mass flux contours for H₂O using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for an intermediate plane.

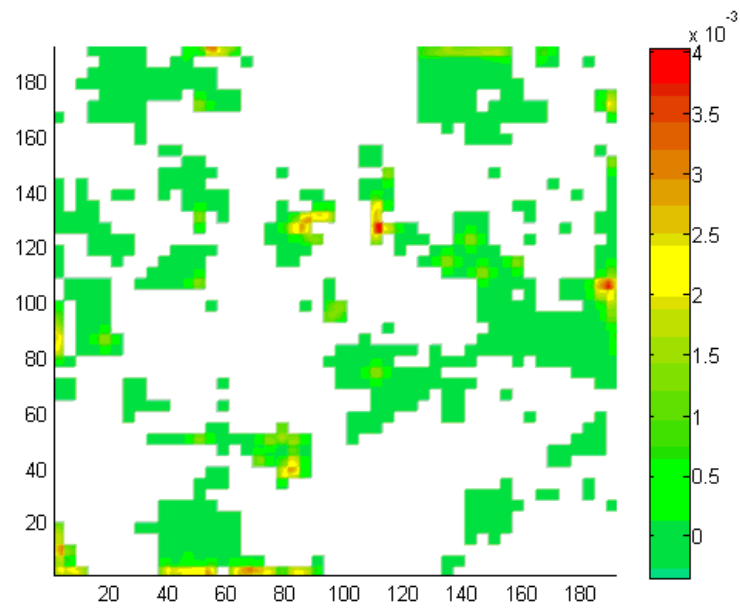


Figure E.156. Normalized mass flux contours for O₂ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for the inlet/outlet plane.

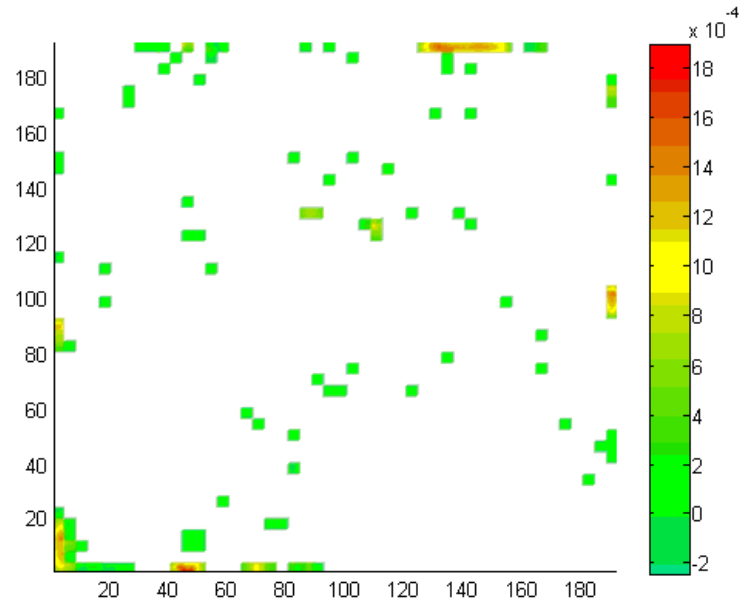


Figure E.157. Normalized mass flux contours for O₂ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for an intermediate plane.

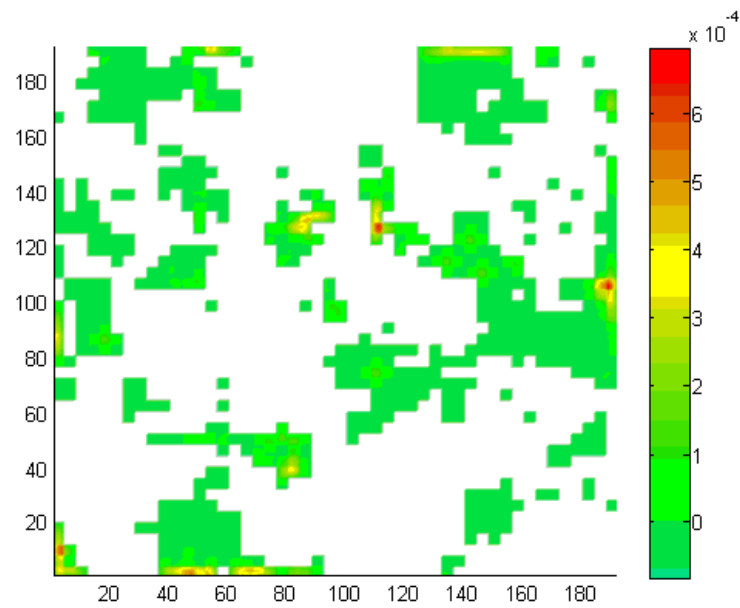


Figure E.158. Normalized mass flux contours for N₂ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for the inlet/outlet plane.

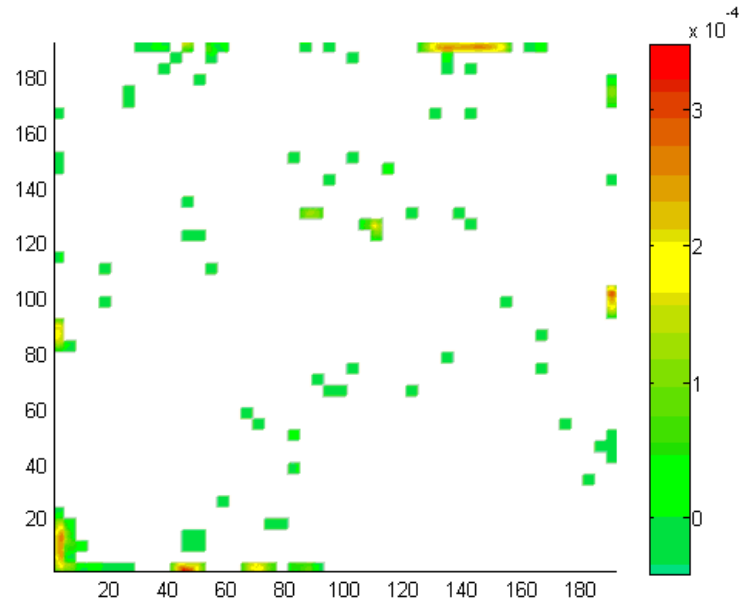


Figure E.159. Normalized mass flux contours for N₂ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for an intermediate plane.

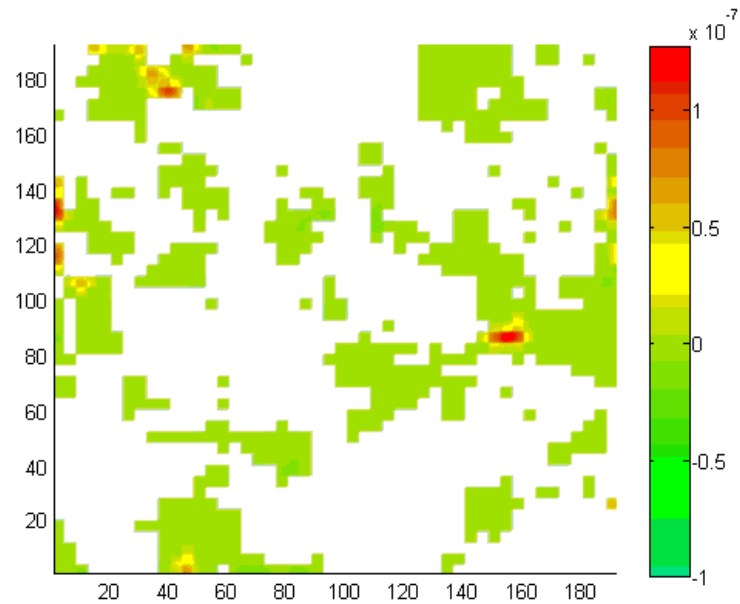


Figure E.160. Normalized mass flux contours for CrO₃ using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for the inlet/outlet plane.

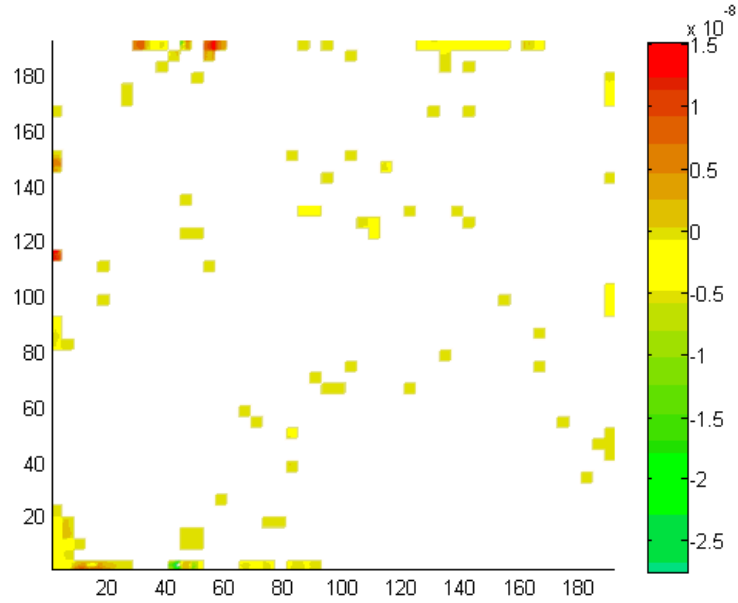


Figure E.161. Normalized mass flux contours for CrO_3 using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for an intermediate plane.

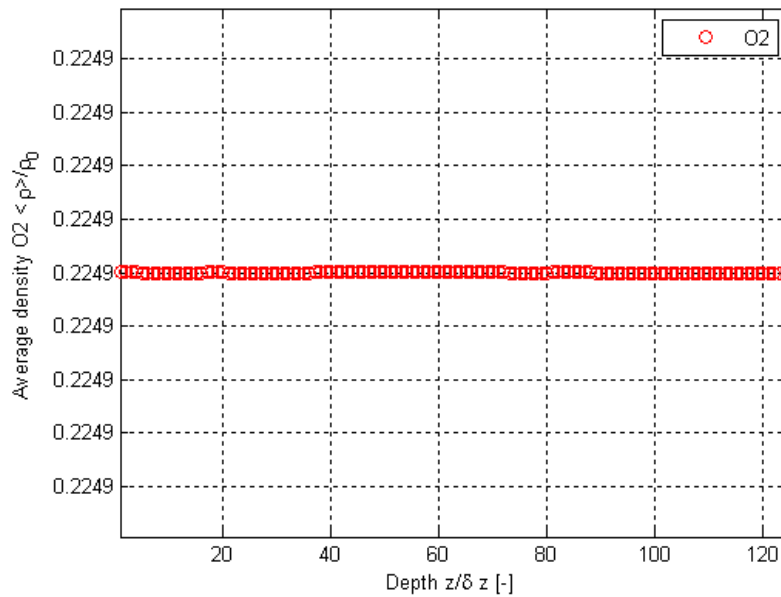


Figure E.162. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 after 10 collisions.

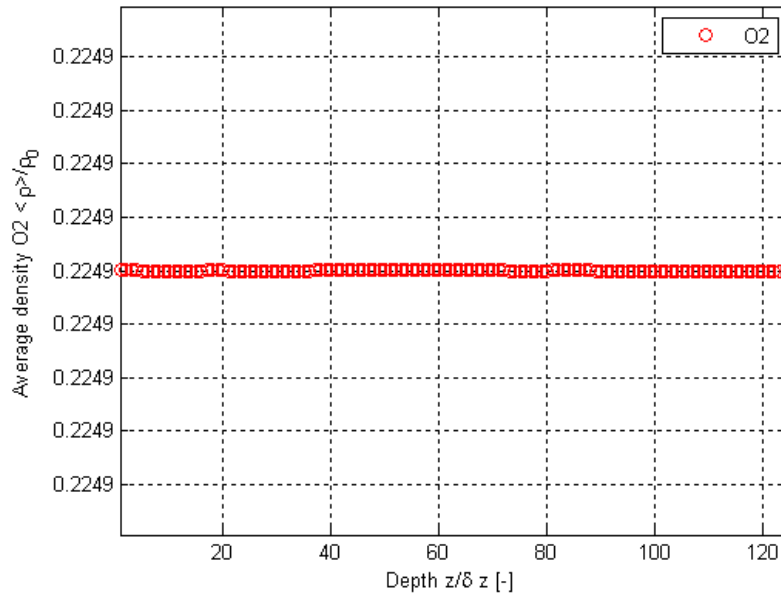


Figure E.163. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for O₂ after 1000 collisions.

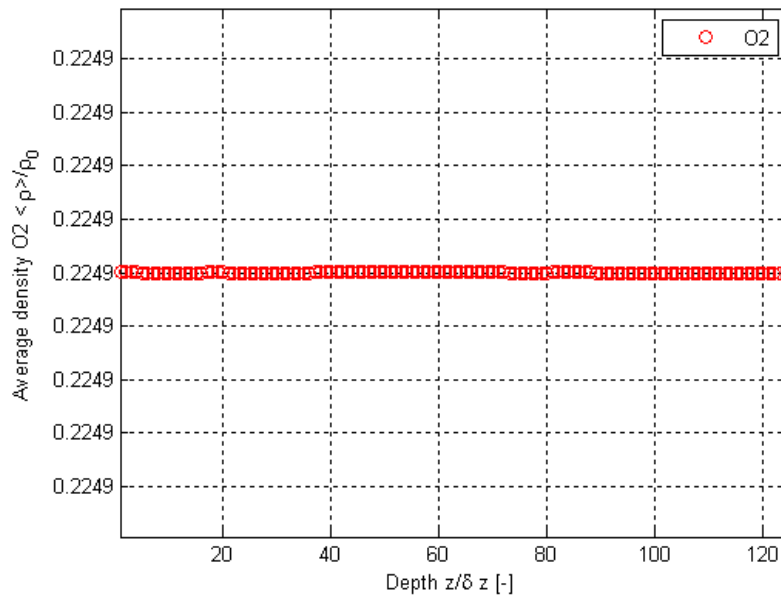


Figure E.164. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for O₂ after 10000 collisions.

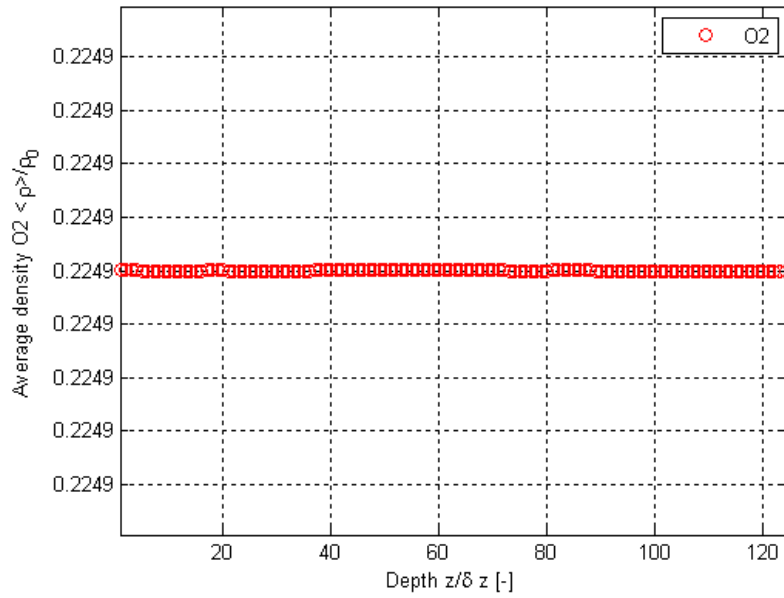


Figure E.165. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for O₂ after 20000 collisions.

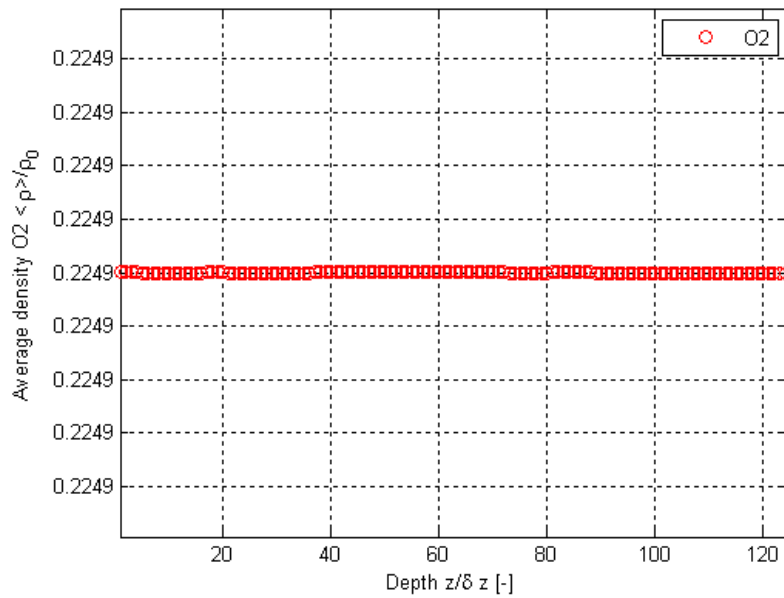


Figure E.166. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂, N₂ and CrO₃ mixture for O₂ after 30000 collisions.

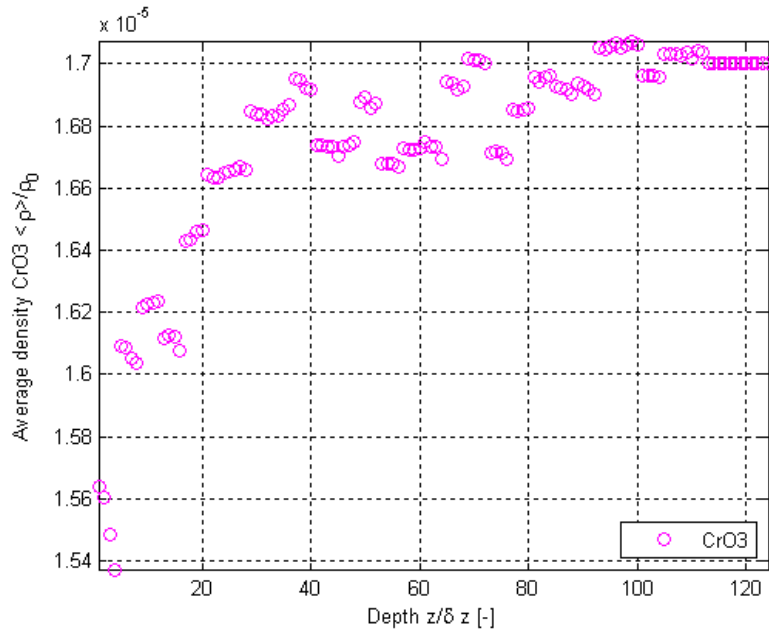


Figure E.167. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 10 collisions.

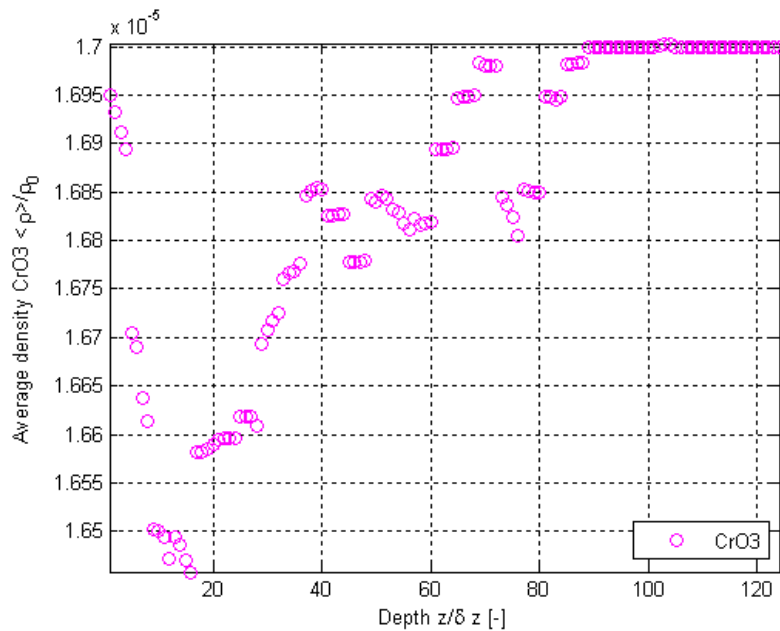


Figure E.168. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 1000 collisions.

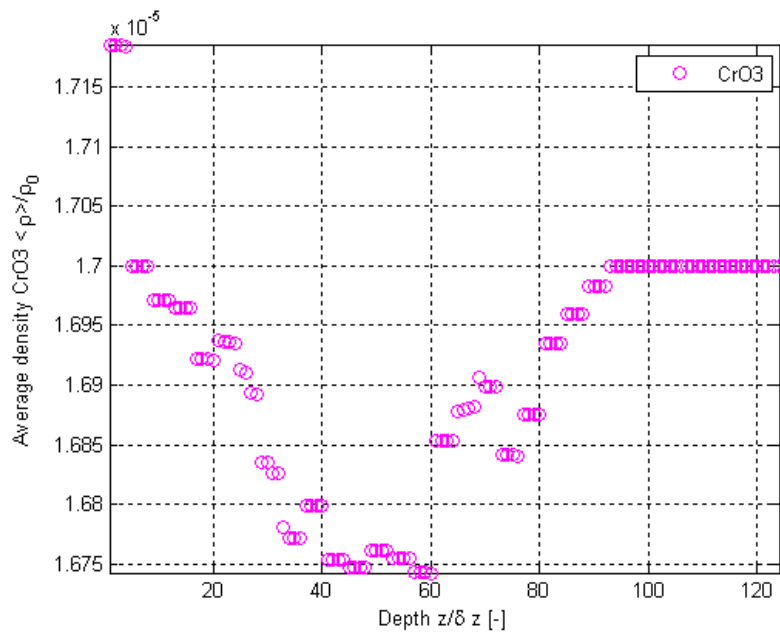


Figure E.169. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 10000 collisions.

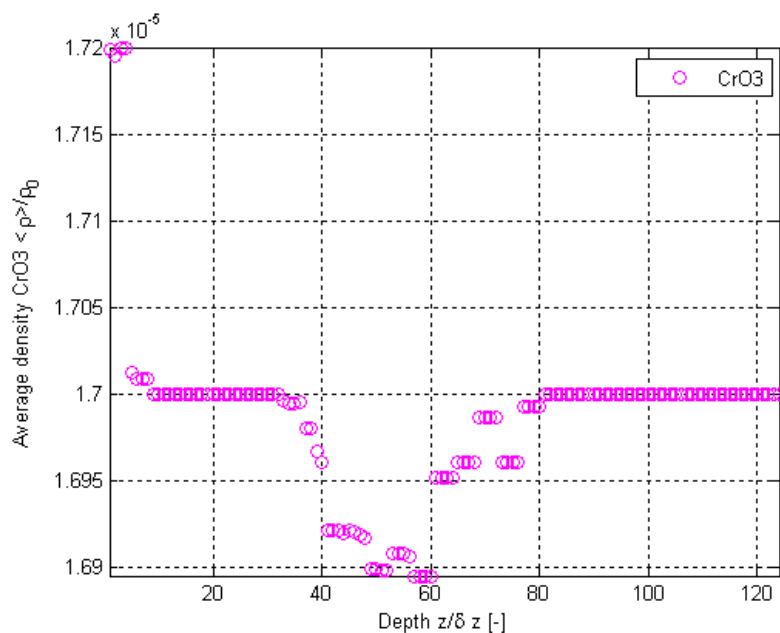


Figure E.170. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 20000 collisions.

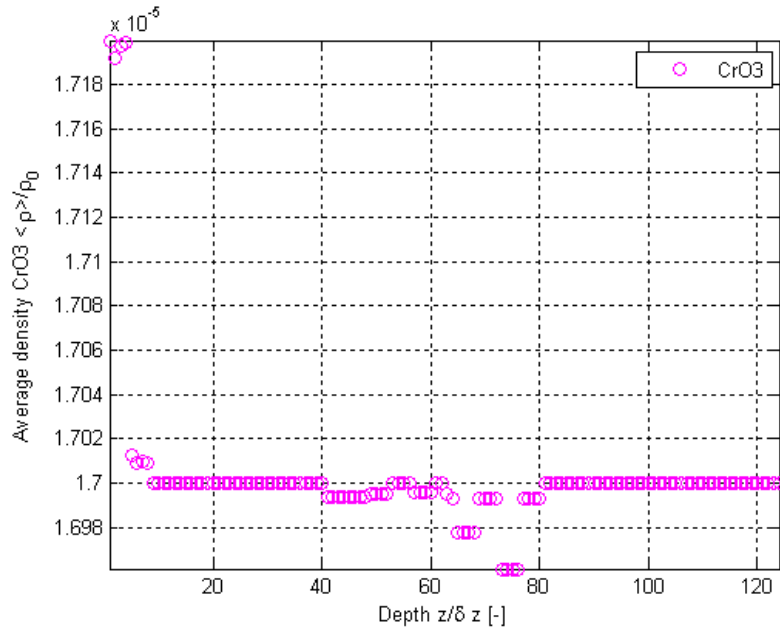


Figure E.171. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 after 30000 collisions.

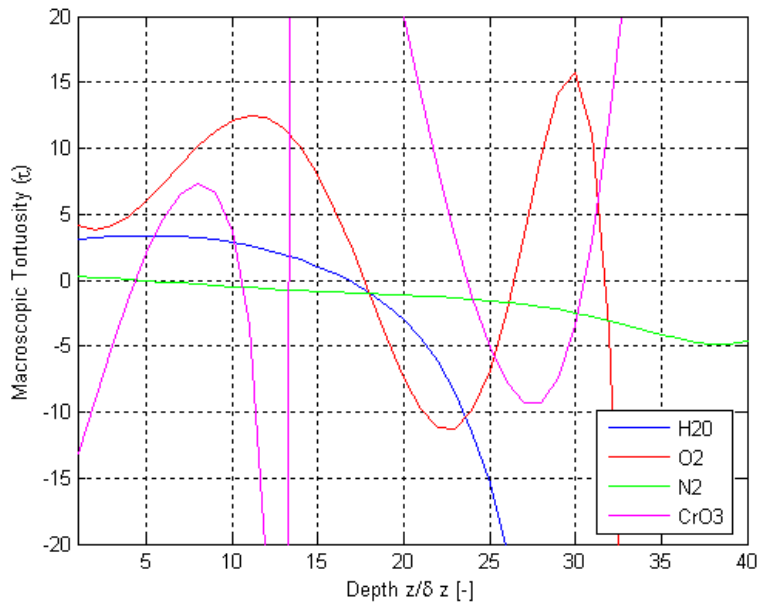


Figure E.172. Results of this thesis work for variations as a function of depth using Geometry G for the H_2O , O_2 , N_2 and CrO_3 mixture for τ .

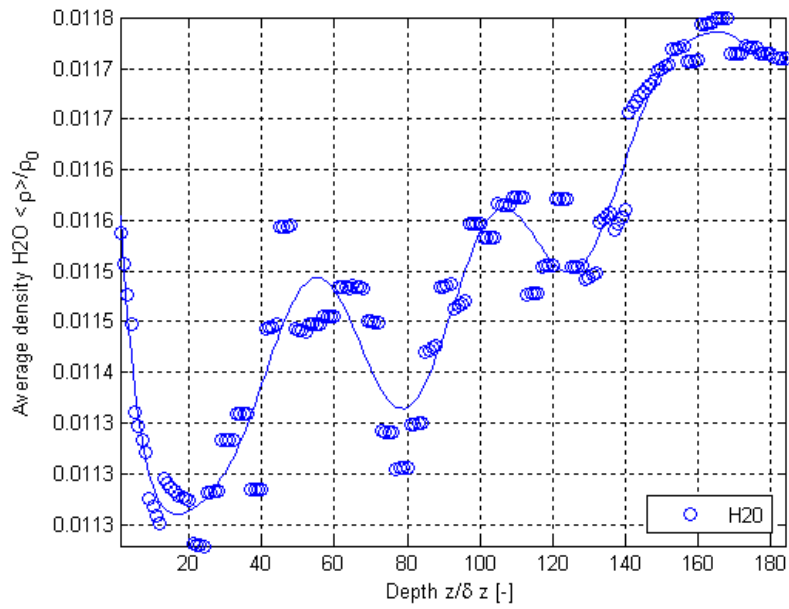


Figure E.173. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry G for the H_2O , O_2 , N_2 and CrO_3 mixture for H_2O .

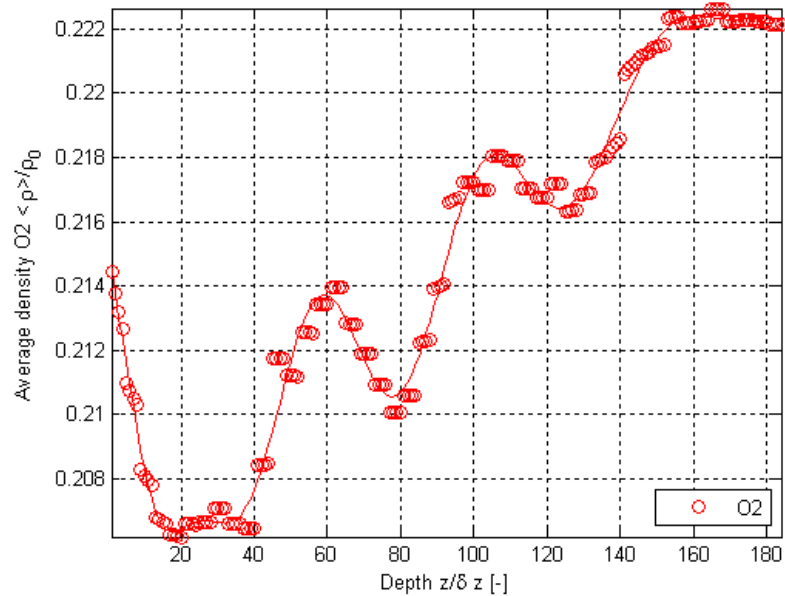


Figure E.174. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry G for the H_2O , O_2 , N_2 and CrO_3 mixture for O_2 .

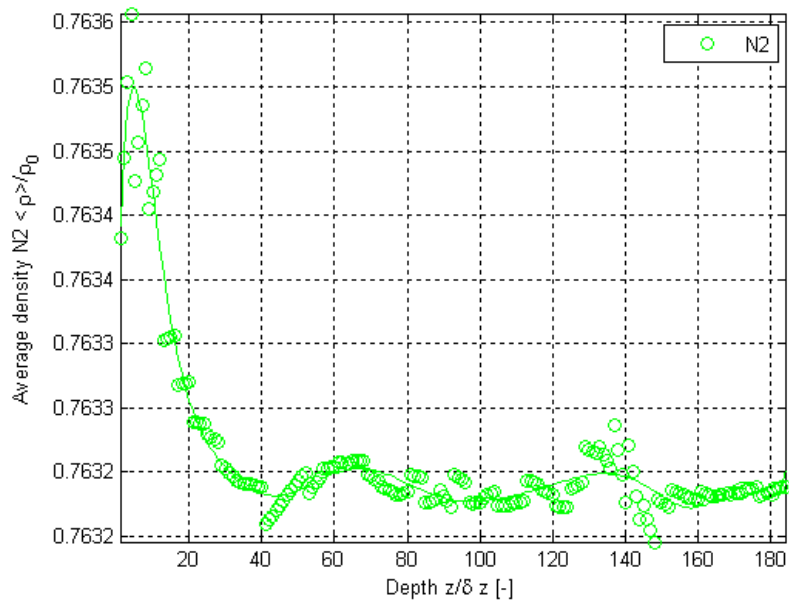


Figure E.175. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry G for the H₂O, O₂, N₂ and CrO₃ mixture for N₂.

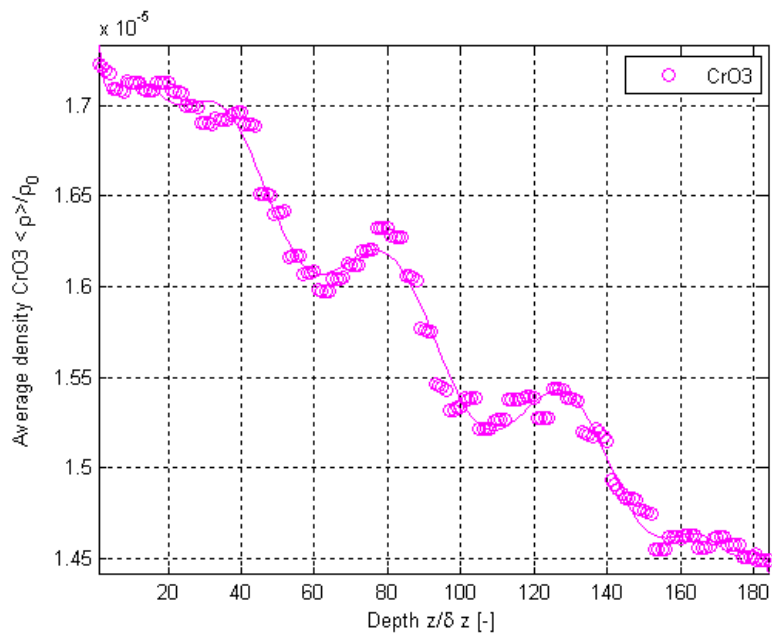


Figure E.176. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry G for the H₂O, O₂, N₂ and CrO₃ mixture for CrO₃.

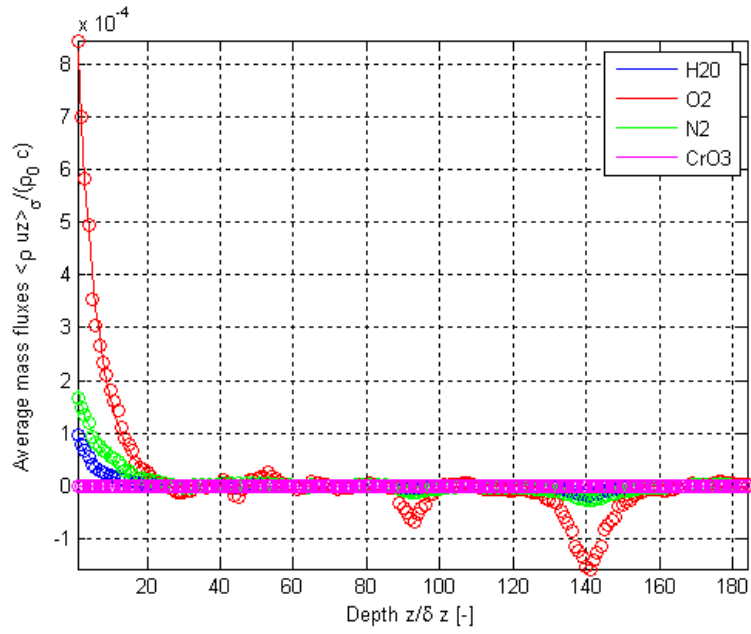



Figure E.177. Results of this thesis work for variations as a function of depth using Geometry G for the H₂O, O₂, N₂ and CrO₃ mixture for the normalized surface-averaged mass flux.

Appendix F: LBM Code Function Tree

LBM Program Hierarchy and Functionality

 indicates file needs modification for cathode analysis

Porous D3Q19.cpp

Defines entry point for the console operation

include

stdafx.h

include file for standard system include files, or project specific include files that are used or changed infrequently

include

DynStructures.h

creates 4 dynamic matrices and 3 dynamic cubes

initialize Dmatrix

call

DMatrix.cpp

store data in, resize, output Dmatrix

initialize DMatrixInt

call **DMatrixInt.cpp**

store data in, resize, output DMatrixInt

initialize DMatrixDouble

call **DMatrixDouble.cpp**

store data in, resize, output DMatrixDouble

initialize DMatrixFloat

call **DMatrixFloat.cpp**

store data in, resize, output DMatrixFloat

initialize Dcube

call **DCube.cpp**

store data in, resize, output Dcube

initialize DCubeInt

call **DCubeInt.cpp**

store data in, resize, output DCubeInt

initialize DCubeDouble

call **Dcube Double.cpp**

store data in, resize, output DCubeDouble

include **Macros.h**

Definition of macros and constants like maximum vector and matrices' dimensions

include

DataStructures.h

Initialize data structures: ControlFlags, GridData, PhysData, ResultsData, LatticeData

include **FunctionPrototypes.h**

Defines function prototypes, ie specifies the interface of the functions

call **InitStructs.cpp**

*initializes structures to null values or values specified in **Macros.h***

if MyNode=0 'MyNode' is the processor number

call **InputData.cpp**

```

if FRONTEND_INPUT in Macros.h = 1
    gets user input for simulation type
    Gets user input for loading test files, refinement values, void fractions, pressure
    differences, # of geometries, gmtry file names, Geometry.csv, as appropriate
if FRONTEND_INPUT in Macros.h = 0
    call LoadInput.cpp
        opens test reference data: Input.log
    call InputDataDefault.cpp
        Accepts user input and initialises relative structure fields
    call InputDataCalculated.cpp
        calculate single or fluid mixture properties
        calculate time constants and dimensionless frequencies for species
        calculate time constants and dimensionless frequencies for diffusion
Sending Master --> Slaves (150, 1-103)
Send value of working mode to all slave nodes
Send data structures with loaded values to all slave nodes
call SendStructs.cpp
    initializes structures to null values or values specified in Macros.h
    receive working mode
    Update local data structures with user input given to node 0
else
    Receiving Slaves <-- Master (150, 1-103)
    call
    ReceiveStructs.cpp
        initializes structures to null values or values specified in Macros.h
Begin cycle on porosity: select current void fraction
Begin cycle on pressure gradients: calc current lattice data and pressure gradient
    call InitMixture.cpp to initialize mixture data
        Function that initializes PhysData for use with single or multi fluid mixtures
    Begin cycle on geometry (real solution loop)
        get/set data for current calculation
        if MyNode=0 (Master Node - Geometry Generation, Partition and Distribution)
            call DCubeInt.cpp to define BaseGeometry and Geometry
                generate base geometry
                if FileNames[1] == 48
                    call GetGeometry.cpp
                        coordinates the geometry creation phase
                    call CreateGeometry.cpp
                        structures geometry information in Topology matrix
                        generate uniformly porous mtl if desired
                    call PlaceParticles.cpp
                        structures geometry information in Topology matrix
                        distributes solid particles into geometry matrix
                    generate non-isotropic mtl if desired
                    Calculation of the number of particles to be placed in
                    the 3 regions
                    Particle placement in the first region (uniform porosity)
                    call PlaceParticles.cpp
                    Particle placement in the second region (linearly varying porosity)
                    call PlaceParticles.cpp
                    Particle placement in the third region (uniform porosity)

```



```

                                call PlaceParticles.cpp
    call FindPath.cpp
        finds a path of empty cells
        check to make sure the geometry has a valid fluid path
    else
        call LoadTest.cpp if user wants to run reference test case
            loads test reference geometry stores in variable "Geometry"
call FindPath.cpp
    finds a path of empty cells
Save data before refinement
call RefineMesh.cpp
    performs mesh refinement as per defined refinement factor
call UpdateData.cpp
    updates grid data during mesh refinement if required
Sending Master --> Slaves (1-103)
    Send data structures with loaded values to all slave nodes
call SendStructs.cpp
    initializes structures to null values or values specified in Macros.h
call ShowGeometry.cpp if in Debug mode
call FindOptDecomp3D.cpp
    get best domain decomposition
Splitting of refined Geometry for all nodes
    Receiving Master <-- Slaves (400-401) if SlabDecomposition==1
    call PartGeometry3D.cpp
        define row/column/layer start/finish indexes
        add boundry buffer
        return partition dimension
    call DCubeInt.cpp to define NodeGeometry
        structure that holds geometry buffer
        Sending Master --> Slaves (200-203)
    Sending Master --> Slaves (500) if SlabDecomposition==1
call PartGeometry3D.cpp
    define row/column/layer start/finish indexes
    add boundry buffer
    return partition dimension
call DCubeInt.cpp to define RootGeometry
    structure that holds geometry buffer
call ShowGeometry.cpp if in Debug mode
set up Output.log
call FindOptDecomp3D.cpp
    get best domain decomposition
call CreateTopology3D.cpp
    structures topology information in Topology matrix
Wait for all the processes to synchronize
call SolverLBM.cpp (Solver 0, solver type defined in Macros.h as a constant)
    description need to be added, skiped b/c solver=1
if NumFluids==1 call SolverLBM3D.cpp (Solver 1, solver type defined in
                                Macros.h as a constant)

Initialization
    initialize local variables

```

call **InitLattice.cpp**
*Accepts user input and initialises relative structure fields
 modifies variable LatticeData with direction values*

call **CreateInternalBoundaryOld.cpp**
*gets number of cells generated by **CreateTopology3d.cpp**
 creates InternalLayer, InternalBoundary, and InternalCondition matrices*

call **NeighbourProcess.cpp**
*determine the neighbours of the point
 initialize Increment variable
 call **FindOptDecomp3D.cpp**
 call **PartGeometry3D.cpp***

call **IntQuickSort.cpp** for InternalBoundary
*sorts matrices in specified region, between indices
 initialize "SendBlock" and "SendBlockPointer" variables*

call **CreateExternalBoundaryOld.cpp**
*gets number of cells generated by **CreateTopology3d.cpp**
 creates ExternalLayer, ExternalBoundary, and ExternalCondition
 matrices*

call **NeighbourProcess.cpp**
*determine the neighbours of the point
 initialize Increment variable
 call **FindOptDecomp3D.cpp**
 call **PartGeometry3D.cpp***

call **IntQuickSort.cpp** for ExternalBoundary
*sorts matrices in specified region, between indices
 initialize "RecvBlock" and "RecvBlockPointer" variables
 set up **Output.log** ???
 find and save calculation start time*

run collision loop
 Sending Slave --> Slave
 Wait for all the processes to synchronize
 Receiving Slave <-- Slave
 Wait for all the processes to synchronize

call **DoubleBubbleSort.cpp**
sorts matrix with a target column and a frozen column

Propagation: call **StreamingStep3D.cpp**
*Accepts user input and initialises relative structure fields
 iterate on fluid cells
 iterate on lattice velocities not at rest
 particles at rest
 Reflection*

Collision: call **CollisionStep3D.cpp**
*Accepts user input and initialises relative structure fields
 define variables and constants
 collect data
 common sub-expression elimination (CSE)
 hydrodynamic moments [8/3]
 code macroscopic variables
 kinetic moments [33/16]
 equilibrium moments [2/8]*

```

    relaxation [6/6]
    backtransformation [86/52]
Show Results
Wait for all the processes to synchronize
if MyNode != 0 Slave Node (Geometry Generation, partition, and distribution)
    Sending Slaves --> Master (800-900)
else Master Node (Geometry Generation, partition, and distribution)
    Receiving Master <-- Slaves (800-900)
Convergence Check
if MyNode != 0 Master Node (Geometry Generation, partition,
    and distribution)
    Sending Slaves --> Master (900)
else Slave Node (Geometry Generation, partition, and distribution)
    Receiving Master <-- Slaves (900)
Wait for all the processes to synchronize
Record time
Calculation of output data
Front end results plot
Wait for all the processes to synchronize
if NumFluids > 1 call SolverLBMmix3D.cpp (Solver 1, solver type defined in
    Macros.h as a constant)
Initialization
    initialize local variables
    call InitLatticeMix.cpp
        Accepts user input and initialises relative structure fields
        modifies variable LatticeData with direction values
        save Output.log
    call RandomDivergenceFree.cpp
        initialize
        random field
        orthogonalization
        call round.cpp
            returns a correctly rounded value
        Inverse transformation
        set up LatticeData for species and mixture types
    save Output.log, indicate start time
    get onion enter and exit boundary
        call CreateOnion.cpp
    get internal boundry if PRE_TIMING defined
    call CreateInternalBoundry.cpp
        initialize "SendBlock" and "SendBlockPointer" variables
        gets number of cells generated by CreateTopology3d.cpp
        creates InternalLayer, InternalBoundary, and InternalCondition matrices
    call NeighbourProcess.cpp
        detirmine the neighbours of the point
        initialize Increment variable
        call FindOptDecomp3D.cpp
        call PartGeometry3D.cpp
    call IntQuickSort.cpp for InternalBoundary
        sorts matrices in specified region, between indices

```

```

get pre extrenal boundry if PRE_TIMING defined
call CreateExternalBoundary.cpp
    initialize "SendBlock" and "SendBlockPointer" variables
    gets number of cells generated by CreateTopology3d.cpp
    creates ExternalLayer, ExternalBoundry, and ExternalCondition
    matrices
call NeighbourProcess.cpp
    detirmine the neighbours of the point
    initialize Increment variable
    call FindOptDecomp3D.cpp
    call PartGeometry3D.cpp
call IntQuickSort.cpp for ExternalBoundry
    sorts matrices in specified region, between indices
save Output.log, indicate start time
for t = 1 : NumCollisions Main Loop
    Sending Slave --> Slave
    Receiving Slave <-- Slave
        display node entering sending step and loop #
        display node and tag numbers
    Propagation (core)
        display node entering streaming step and loop #
    call StreamingStepMix3D.cpp
        initialize local variables
        detirmine direction based on particle class, position
    Collision (core)
        display node entering collision step and loop #
    call CollisionStepMix3D.cpp
        initialize local variables
        BaryCentric Collision
            collect data: inlet/outlet and electrochemical boundry condition
            common sub-expression elimination (CSE) [33/3]
            hydrodynamic moments [8/3]
            redefine code macroscopic variables
            kinetic moments [33/16]
            equilibrium moments [2/8]
            relaxation [6/6]
            backtransformation [86/52]
        Species collision
            collect data: inlet/outlet and electrochemical boundry condition
            common sub-expression elimination (CSE) [33/3]
            hydrodynamic moments [8/3]
            redefine code macroscopic variables
            kinetic moments [33/16]
            equilibrium moments [2/8]
            relaxation [6/6]
            backtransformation [86/52]

Synchronize
Correction
    display node entering correction step and loop #
    display node and tag numbers

```

```

    for loop, purpose ???
    Propagation (Boundary)
        display node entering streaming step and loop #
        call StreamingStepMix3D.cpp
    Collision (Boundary)
        display node entering collision step and loop #
        call CollisionStepMix3D.cpp
        Invert (mix_phi_prop) with (mix_phi)
        Invert (phi_prop) with (phi)
    Show Results
    call TurbulenceData.cpp if INCLUDE_FFTW is defined
        ???
    if MyNode != 0 Sending Slaves --> Master (800)
    else Receiving Master <-- Slaves (800)
    Covergence Check
    if MyNode != 0 Sending Master --> Slaves (900)
    else Receiving Slaves <-- Master (900)
    display end time
    Calculation of output data
    Frontend results plot
    call SolverLBM.cpp (Solver 2, solver type defined in Macros.h as a constant)
        description need to be added, skipped b/c solver=1
    call PackageResults.cpp
        packages analysis results on disk
        Write Velocity.csv, Density.csv, VelocityX.csv, VelocityY.csv,
            and VelocityZ.csv for wall and fluid cells
    call SaveResults.cpp
        saves analysis results on disk
        Save Velocity.csv, Density.csv, VelocityX.csv, VelocityY.csv,
            and VelocityZ.csv
        Receiving Master <--Slaves (1000-1002)
    call SaveResults.cpp
    else (Slave nodes- Geometyery Generation, Partition, and Distribution)
        Receiving Slaves <--Master (1-103)
        call ReceiveStructs.cpp
            initializes structures to null values or values specified in Macros.h
        if SlabDecomposition == 1 Sending Slaves --> Master (400-401)
        Receiving Slaves <--Master (200-203)
        if SlabDecomposition == 1 Receiving Slaves <-- Master (500)
    call ShowGeometry.cpp if in Debug mode
    call FindOptDecomp3D.cpp
        get best domain decomposition
    call PartGeometry3D.cpp
        define row/column/layer start/fininsh idexes
        add boundry buffer
        return partition dimension
    call CreateTopology3D.cpp
        structures topology information in Topology matrix
    call SolverLBM.cpp (Solver 0, solver type defined in Macros.h as a constant)
        description need to be added, skipped b/c solver=1

```

*if NumFluids=1 call SolverLBM3D.cpp (Solver 1, solver type defined in
Macros.h as a constant)*
*if NumFluids > 1 call SolverLBMmix3D.cpp (Solver 1, solver type defined in
Macros.h as a constant)*
call **SolverLBM.cpp** (Solver 2, solver type defined in **Macros.h** as a constant)
description need to be added, skipped b/c solver=1
Sending Slaves --> Master (1000-1002)
Call **PackageResults.cpp**

End if

End cycles

Display end node

Appendix G Refinement Factor 8 Preliminary Results

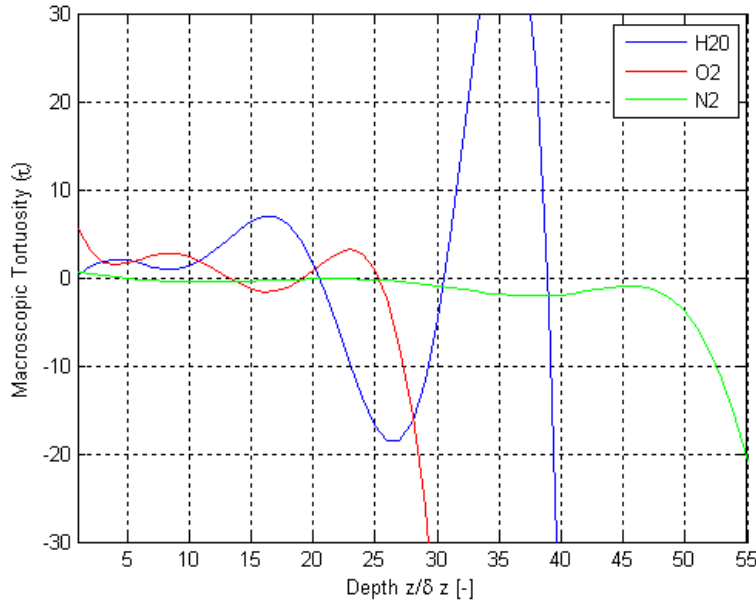


Figure G.1. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for τ .

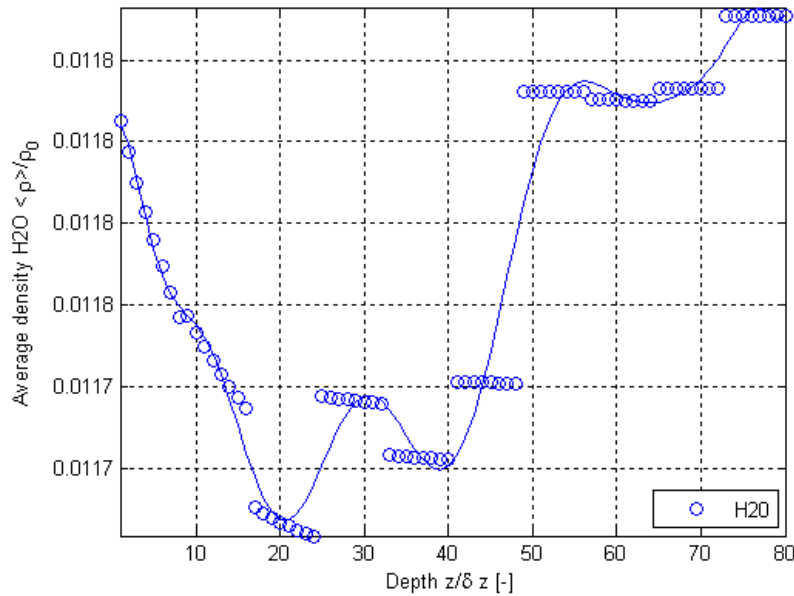


Figure G.2. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for H₂O.

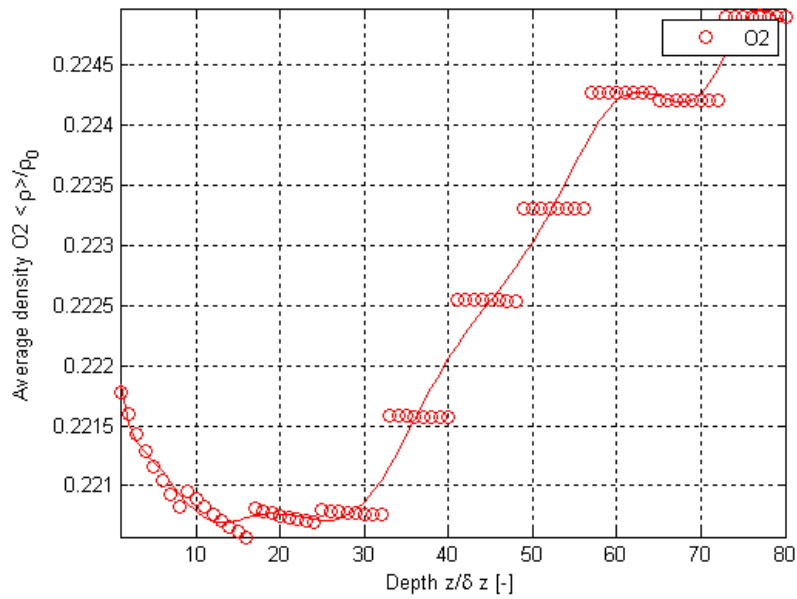


Figure G.3. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 and N_2 mixture for O_2 .

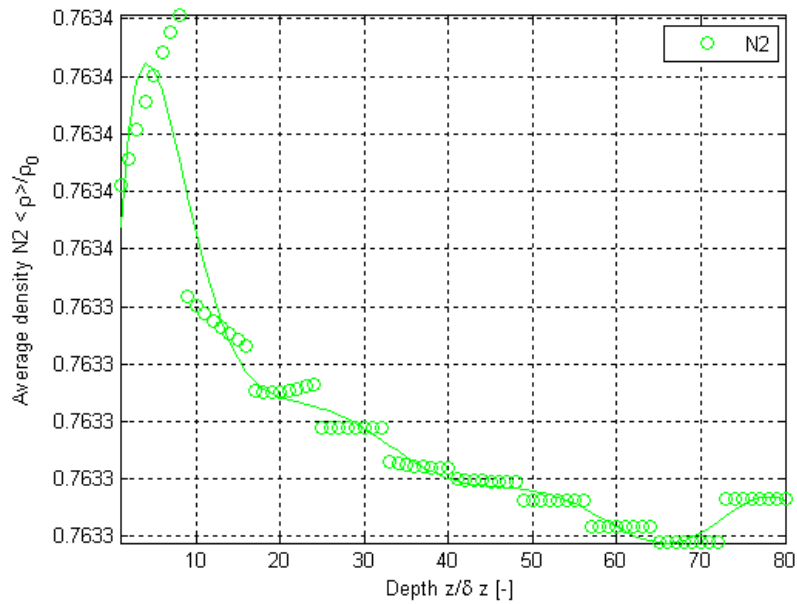


Figure G.4. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 and N_2 mixture for N_2 .

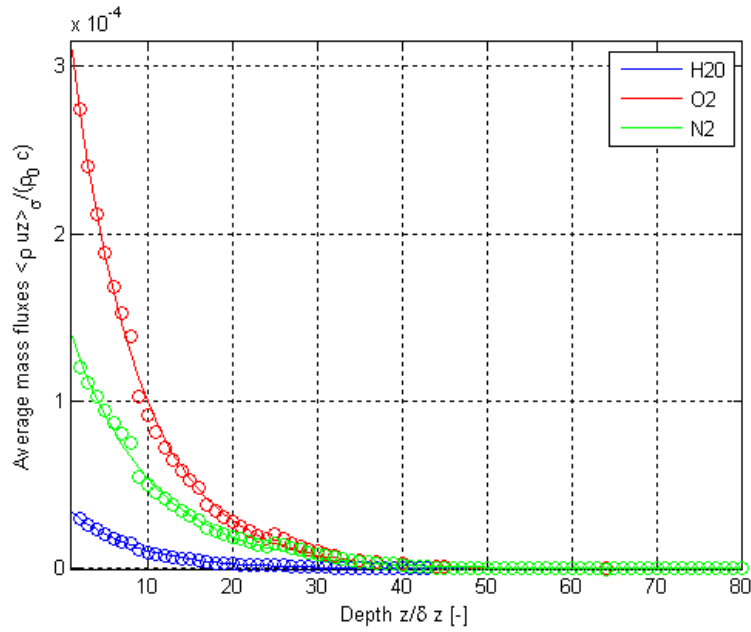


Figure G.5. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

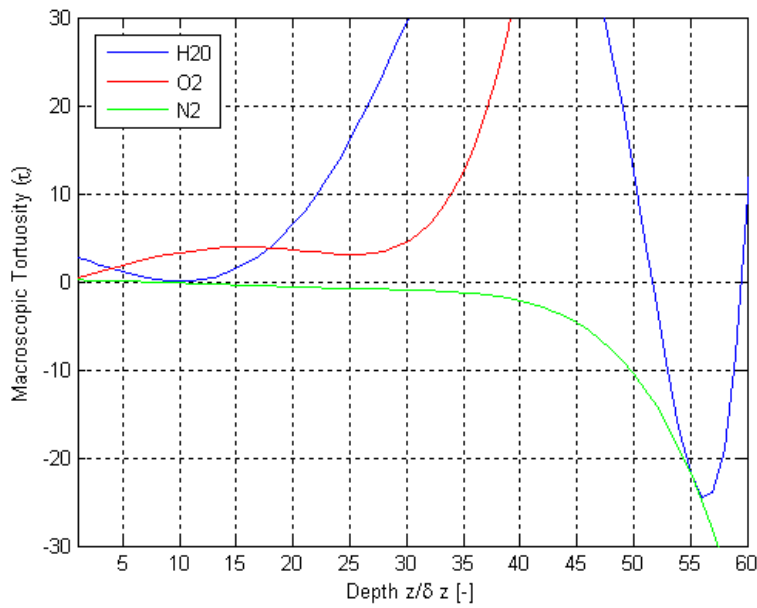


Figure G.6. Results of this thesis work for variations as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for τ .

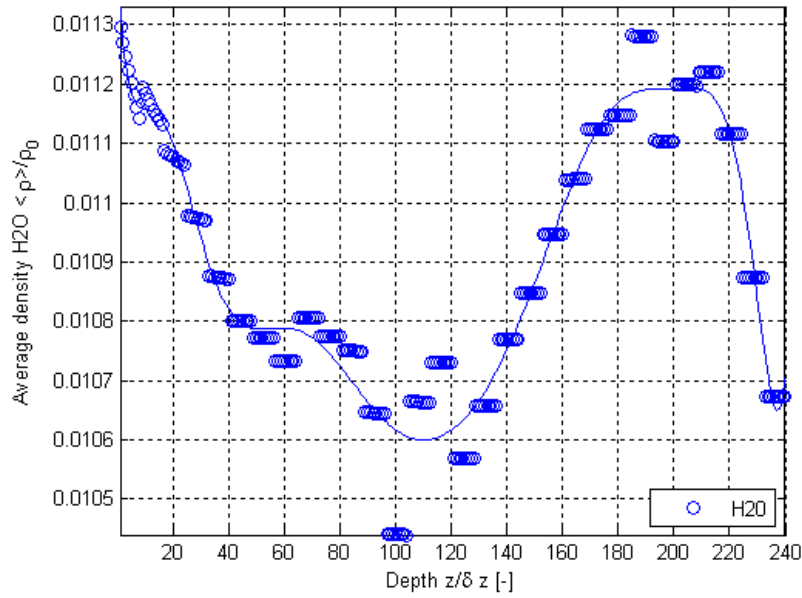


Figure G.7. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for H₂O.

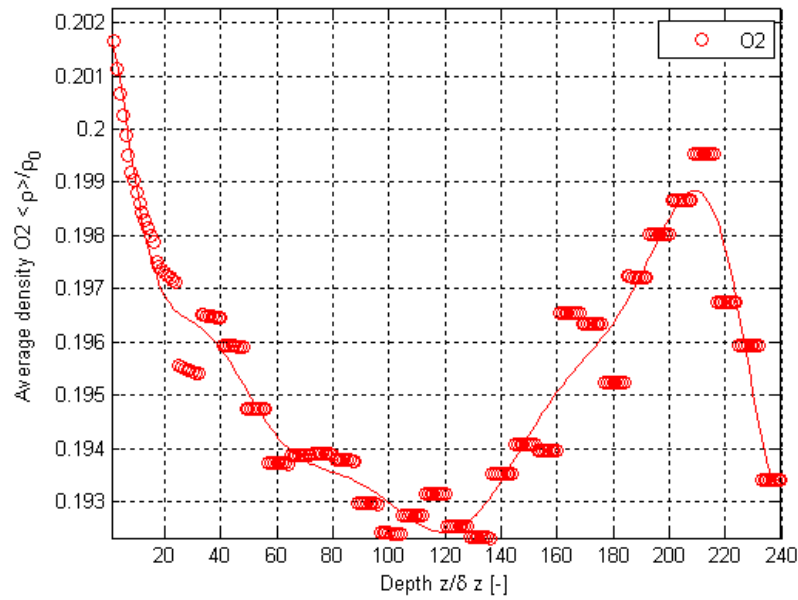


Figure G.8. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for O₂.

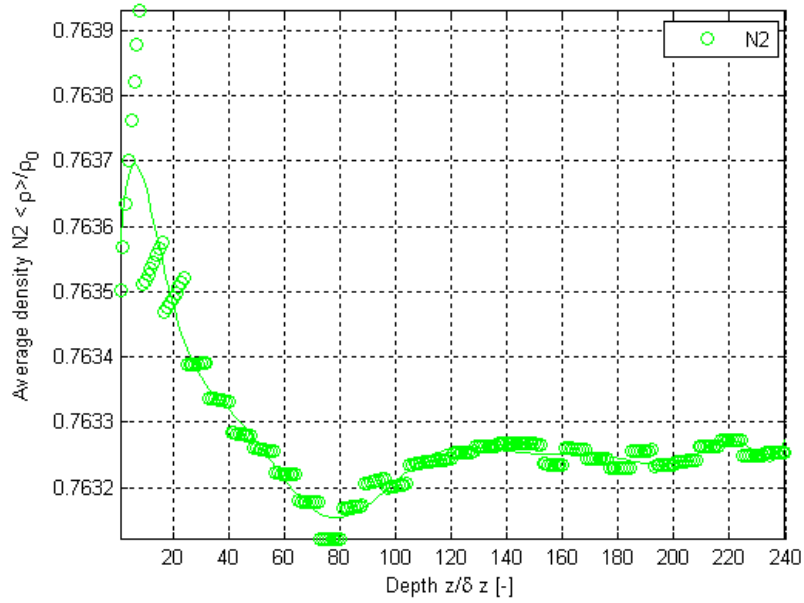


Figure G.9. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for N₂.

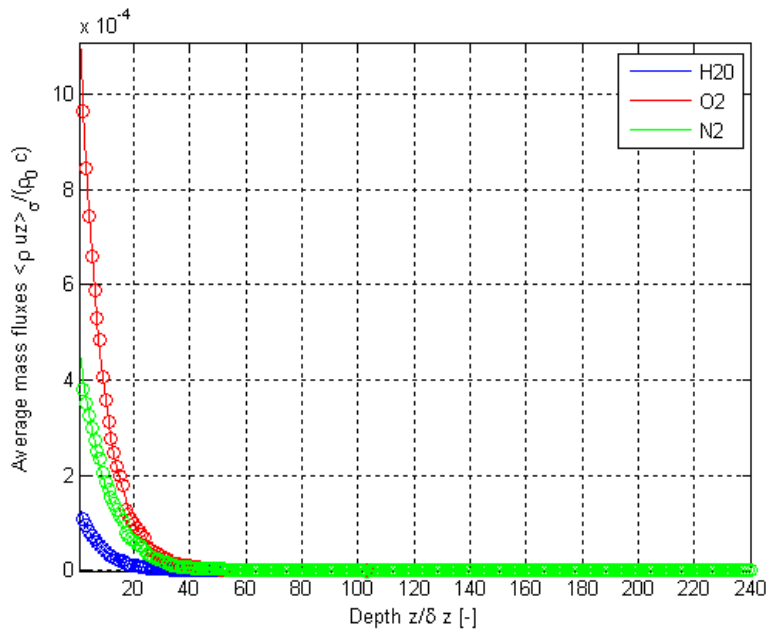


Figure G.10. Results of this thesis work for variations as a function of depth using Geometry D for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

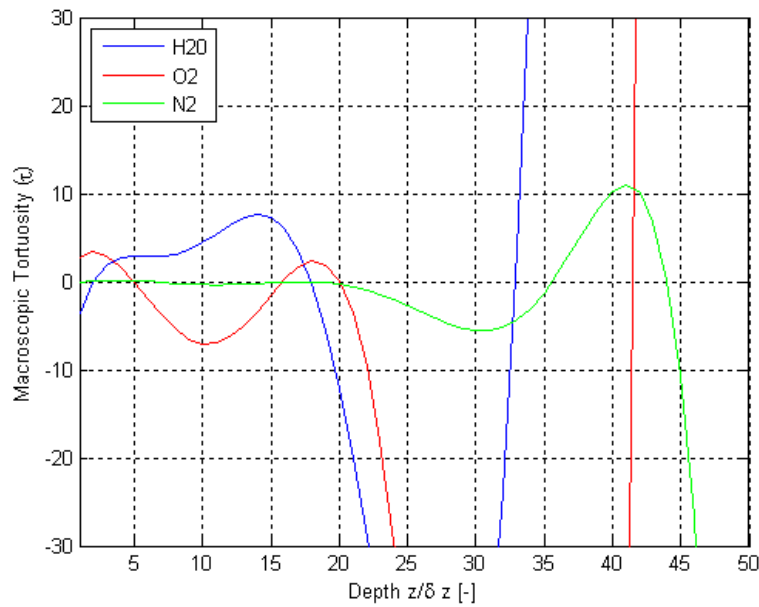


Figure G.11. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for τ .

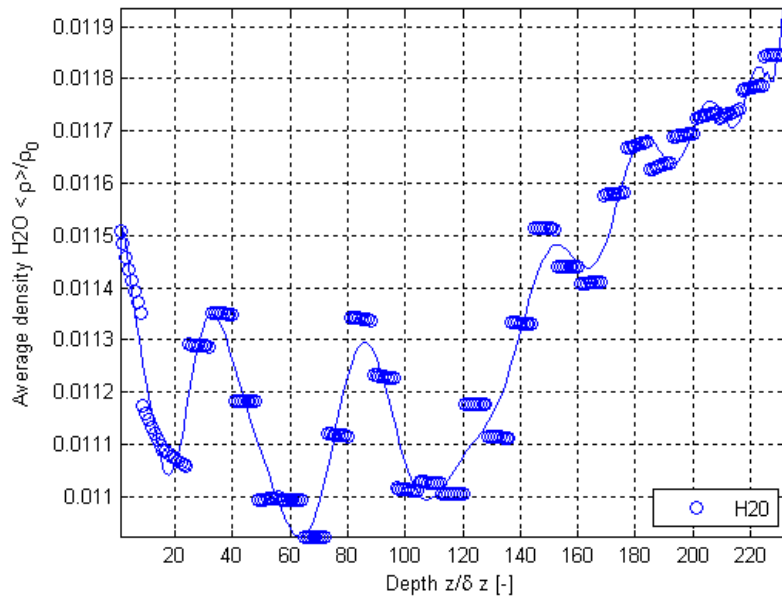


Figure G.12. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for H₂O.

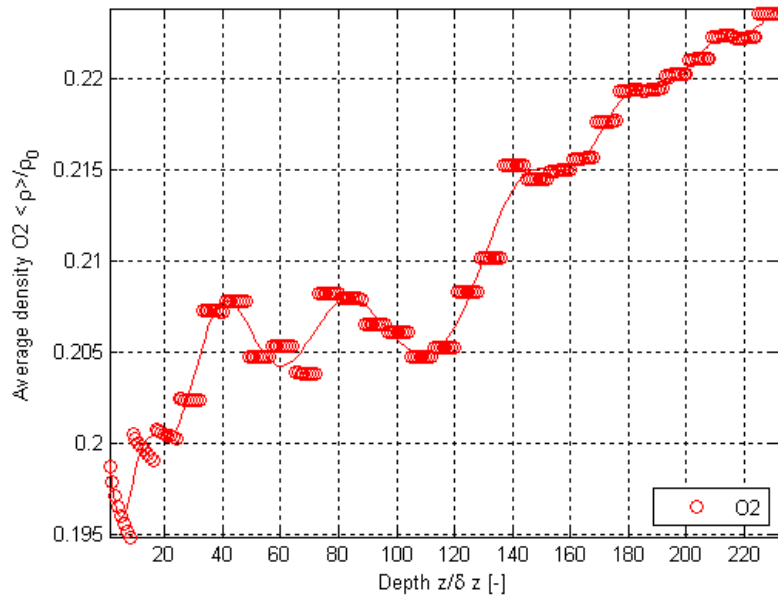


Figure G.13. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 and N_2 mixture for O_2 .

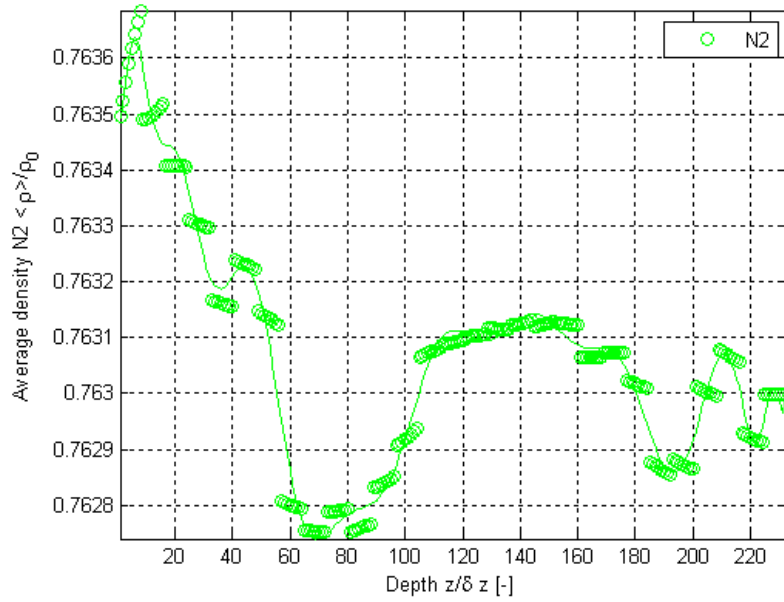


Figure G.14. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry F for the H_2O , O_2 and N_2 mixture for N_2 .

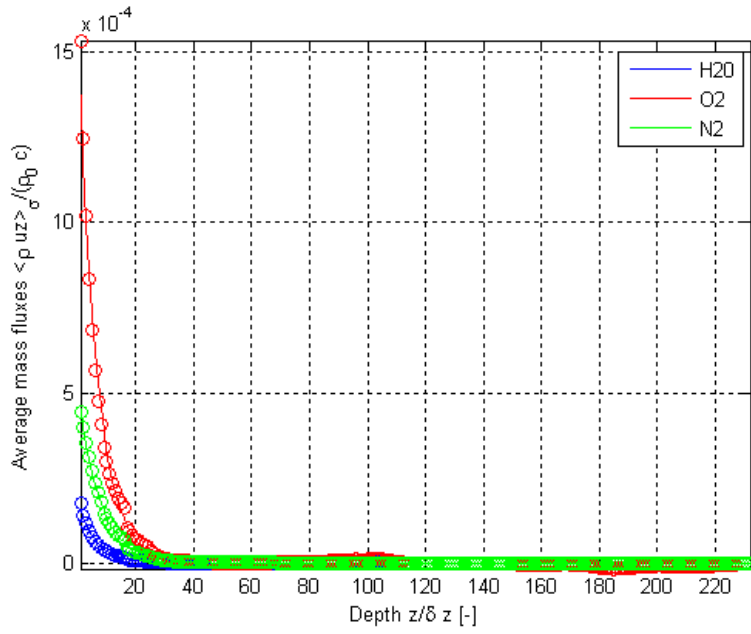


Figure G.15. Results of this thesis work for variations as a function of depth using Geometry F for the H₂O, O₂ and N₂ mixture for normalized surface-averaged mass flux.

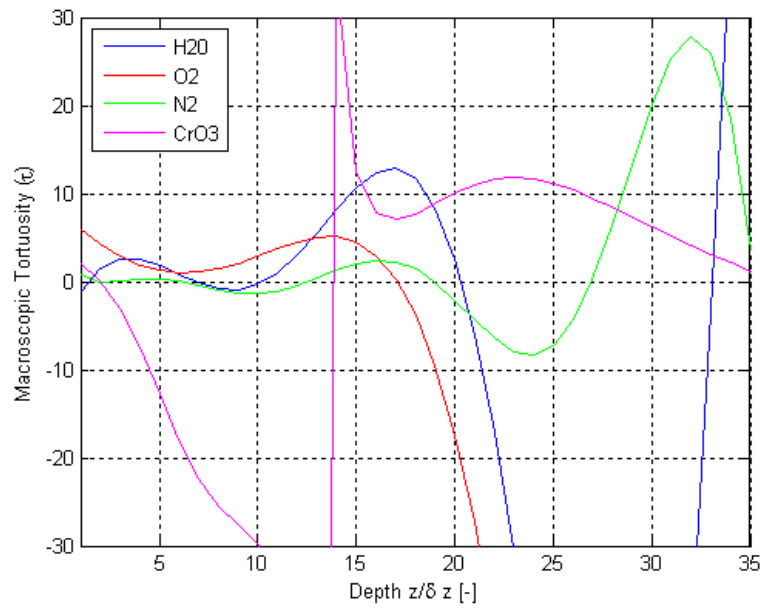


Figure G.16. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for τ .

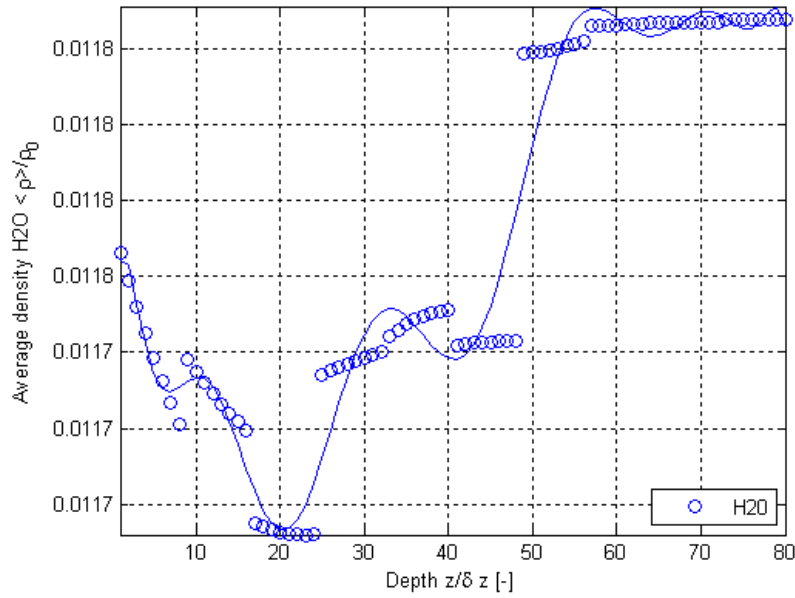


Figure G.17. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for H₂O.

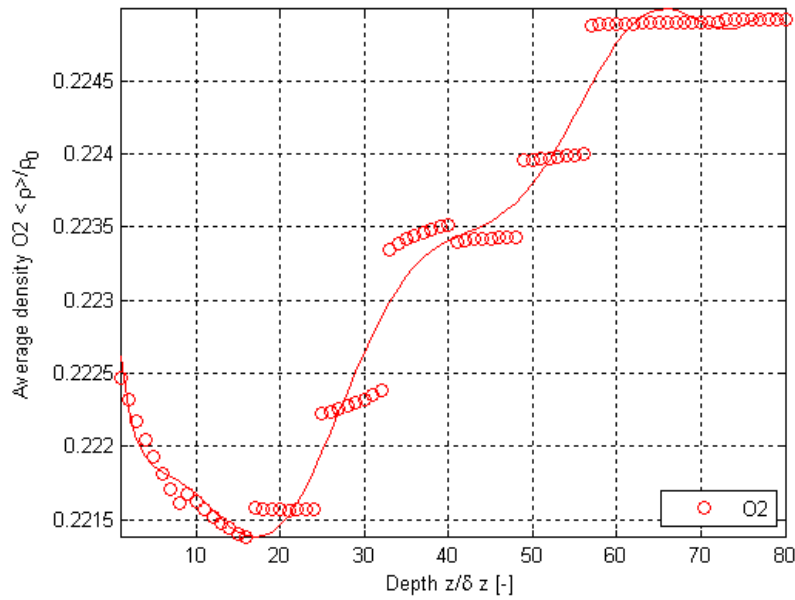


Figure G.18. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for O₂.

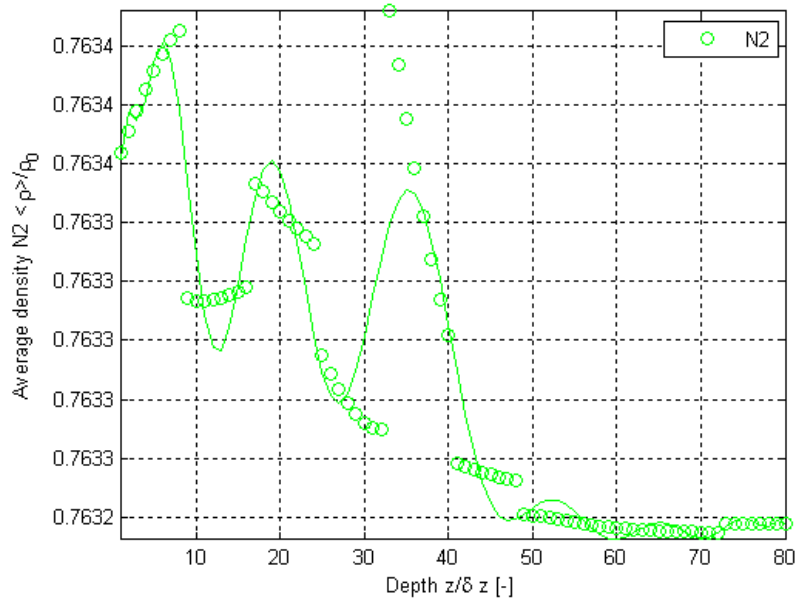


Figure G.19. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for N_2 .

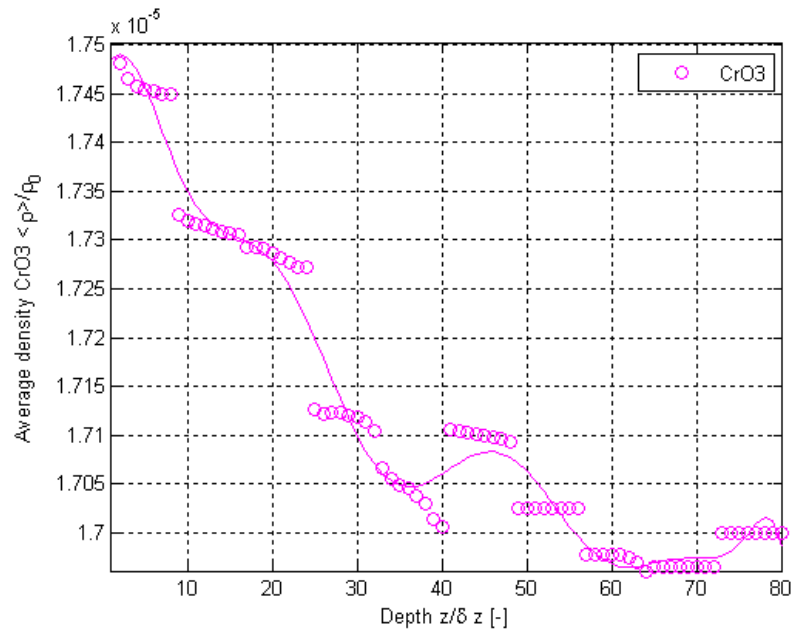


Figure G.20. Normalized surface-averaged density variations from the results of this thesis work as a function of depth using Geometry C for the H_2O , O_2 , N_2 and CrO_3 mixture for CrO_3 .

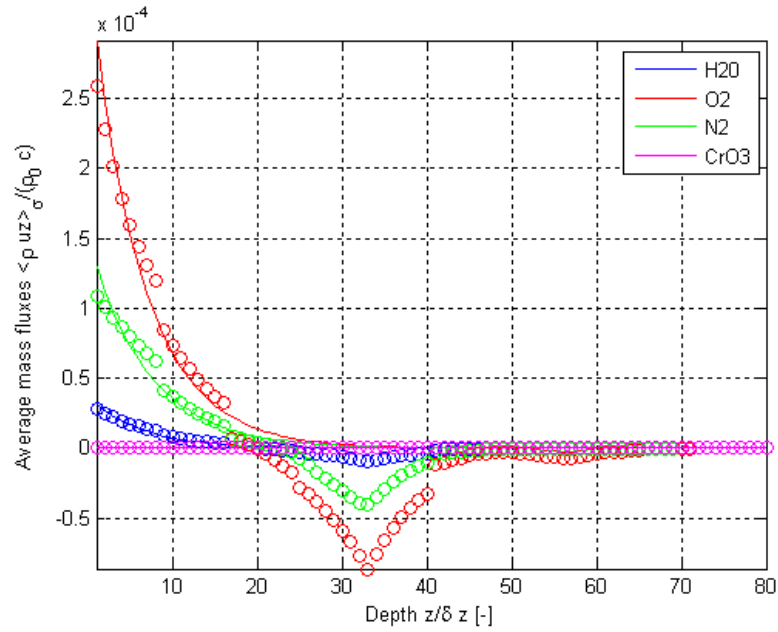


Figure G.21. Results of this thesis work for variations as a function of depth using Geometry C for the H₂O, O₂, N₂ and CrO₃ mixture for the normalized surface-averaged mass flux.

Appendix H Copyright Permission Letters

Gayle,

I hereby grant you my permission to reproduce these figures in your thesis.

Dr. von Spakovsky

Dr. Michael von Spakovsky, Ph.D.
Editor-in-chief
International Journal of Thermodynamics
Associate Editor
ASME Journal of Fuel Cell Science and Technology
Professor and
Director
Center for Energy Systems Research
Mechanical Engineering Department 0238
Virginia Tech
Blacksburg, VA 24061
tel: (540) 231-6684
fax: (540) 231-9100
e-mail: Michael.von.Spakovsky@vt.edu

-----Original Message-----

From: Gayle Kestell [mailto:gkestell@vt.edu]
Sent: Tuesday, February 09, 2010 1:36 PM
To: Michael von Spakovsky
Subject: request for reproduction permission

Dr. von Spakovsky,

I would like to use figures 8 and 9 from the electronic form of:

P. Asinari, M. C. Quaglia, M. R. von Spakovsky and B. V. Kasula, "Direct numerical calculation of the kinematic tortuosity of reactive mixture flow in the anode layer of solid oxide fuel cells by the lattice Boltzmann method," Journal of Power Sources, 170, 359-375, 2007.

and figures 22, 24 and 25 (b) from the electronic form of:

B. V. Kasula, L. Mercado, P. Asinari and M. R. von Spakovsky, "3D microstructure reconstructions of solid oxide fuel cell electrodes with applications to numerical simulations of reacting mixture flows using LBM," Proceedings of IMECE 2007, Seattle, Washington, USA, November 11-15, 2007.

in my thesis. May I please have permission to reproduce these figures.

Thank you,
Gayle Kestell

**ELSEVIER LICENSE
TERMS AND CONDITIONS**

Feb 09, 2010

This is a License Agreement between Gayle M Kestell ("You") and Elsevier ("Elsevier") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

Supplier	Elsevier Limited The Boulevard, Langford Lane Kidlington, Oxford, OX5 1GB, UK
Registered Company Number	1982084
Customer name	Gayle M Kestell
Customer address	Dept of Mechanical Engineering Blacksburg, VA 24061
License Number	2364871249520
License date	Feb 09, 2010
Licensed content publisher	Elsevier
Licensed content publication	Journal of Power Sources
Licensed content title	Degradation phenomena in the cathode of a solid oxide fuel cell with an alloy separator
Licensed content author	S. Taniguchi, M. Kadowaki, H. Kawamura, T. Yasuo, Y. Akiyama, Y. Miyake, T. Saitoh
Licensed content date	May 1995
Volume number	55
Issue number	1
Pages	7
Type of Use	Thesis / Dissertation
Portion	Figures/tables/illustrations
Number of Figures/tables/illustrations	1
Format	Electronic
You are an author of the Elsevier article	No
Are you translating?	No
Order Reference Number	

<https://s100.copyright.com/App/PrintableLicenseFrame.jsp?publisherID=70&licenseID=20...> 2/9/2010

H. C. Starck:

General Conditions of Use

Access to and use of this website are subject to the following conditions. Please do not use this website unless you agree with these conditions. This website has been developed by H.C. Starck Ceramics GmbH & Co. KG (hereinafter to be referred to as "H.C. Starck") and is administrated by the same. We reserve the right to discontinue or to make partial or complete modifications to this website. Please note that we may make such changes at our own discretion and without prior announcement.

Use of this website

All details, documents and illustrations published on this website are the sole property of H.C. Starck. Any permission to use the same is granted on the proviso that the relevant copyright note of H.C. Starck is displayed on all copies, that such details are only used for personal purposes, that they are not exploited commercially, that the details are not modified in any way and that all illustrations gained from the website are only used in conjunction with the accompanying text.

Trademarks

All product names in upper-case letters or marked in some other way on this website are trademarks of H.C. Starck or its affiliates, unless otherwise noted or in any other way perceivable as third party rights. Any unauthorized use or abuse of these trademarks or other materials is expressly prohibited and constitutes a violation of copyright, trademark law or other industrial property rights.

Limited liability

H.C. Starck has compiled the detailed information provided on this website from internal and external sources to the best of its knowledge and belief, using professional diligence. We endeavour to expand and update this range of information on an ongoing basis. The information on this website is purely for the purpose of presenting H.C. Starck and its products and services. However, no representation is made or warranty given, either expressly or tacitly, for the completeness or correctness of the information on this website. In particular, we must ask you to be aware that this information may no longer be up to date. We therefore recommend that you check any information you obtain from this website prior to using it in whatever form.

Advice given on this website does not exempt you from conducting your own checks on our latest advice - particularly our product datasheets and technical specifications - and on our products, with a view to their suitability for the intended processes and purposes. Should you require any advice or instructions concerning our products or services, please contact us directly. Users of this website declare that they agree to access the website and its content at their own risk. Neither H.C. Starck nor third parties involved in the writing, production or transmission of this website can be held liable for damage or injury resulting from access or the impossibility of access or from the use or impossibility of use of this website or from the fact that you have relied on information given on this website.

Websites of third-party vendors/links

In accordance with § 7 Telemedia Law (TMG) H.C. STARCK as provider of contents is responsible for its "own contents" made available for usage in accordance with the general law. From this own contents references ("Links") to any contents made available by other providers have to be differentiated. In this respect H.C. STARCK makes available for usage "third parties' contents" which is marked correspondingly. Links are always "living" (i.e. dynamic) references. Upon the first linking H.C. STARCK has checked the third parties' contents as to whether it might trigger any possible civil or criminal liability, but it does not constantly check the contents to which it refers in its offer with regard to modifications which could later on lead to a liability. In case H.C. STARCK notices or is made aware of by others that a certain offer to which a link is made available will trigger a civil or criminal liability, it will cancel the link to the offer concerned. This declaration is applicable to all indicated links and to the complete contents of pages to which are leading any possibly installed banners and links.

Details supplied by yourself

The user of this website is fully responsible for the content and correctness of details he or she sends to H.C. Starck as well as for the non-violation of any third-party rights that may be involved in such details.

International users

This website is checked, operated and updated by H.C. Starck at Selb. It is exclusively intended for world wide use. H.C. Starck gives no guarantee that the details presented on this website are also correct in all places, and, in particular, that products and services will be available with the same appearance, in the same sizes or on the same conditions. Should you call up this website, please note that it is your own responsibility to ensure that you act in compliance with local legislation applicable in that place.

Products mentioned on this website may come in different packaging, in different package sizes, or with different lettering or markings, depending on the country.

Inclusion of information about H.C. Starck products on this website is not intended to announce product availability anywhere in the world. Especially US-customers shall consult H.C. Starck GmbH or its local affiliate to learn where products can be sold.

Sale of H.C. Starck Products

Our products are sold in accordance with the current version of our General Conditions of Sale and Delivery.

For sites with login: access rights

Please note that you are obliged to protect your access rights from unauthorized third-party use and that you must ensure that they cannot be used by others without your authorization.

You must notify H.C. Starck immediately if you learn that security has been violated in respect of any details available on the H.C. Starck website or if unauthorized persons have obtained your access rights or if you have found indications that such access might become possible.

Forward-Looking Statements

This internet site contains forward-looking statements. These statements use words like “believes,” “assumes,” “expects” or similar formulations. Various known and unknown risks, uncertainties and other factors could lead to material differences between the actual future results, financial situation, development or performance of our company and those either expressed or implied by these statements. These factors include, among other things:

- downturns in the business cycle of the industries in which we compete;
- new regulations, or changes to existing regulations, that increase our operating costs or otherwise reduce our profitability;
- increases in the prices of our raw materials, especially if we are unable to pass these costs along to customers;
- loss or reduction of patent protection for our products;
- liabilities, especially those incurred as a result of environmental laws or product liability litigation;
- fluctuation in international currency exchange rates as well as changes in the general economic climate; and
- other factors identified in this internet site.

In view of these uncertainties, we caution readers not to place undue reliance on these forward-looking statements. H.C. Starck assumes no liability whatsoever to update these forward-looking statements or to conform them to future events or developments.

Amendment of this General Conditions of Use

For legal and/or organisational reasons amendments and/or adjustments of these General Conditions of Use will become necessary. Therefore please observe the respectively applicable version of our General Conditions of Use.

Privacy and Security Notification
ATTENTION!

1. CorpsWeb is provided as a public service by the U.S. Army Corps of Engineers (USACE).
2. Information presented on CorpsWeb public web sites is considered public information and may be distributed or copied. Use of appropriate byline/photo/image credits is requested.
3. For site management, information collected at CorpsWeb public web sites is for statistical purposes. These government computer systems use software programs to create summary statistics, which are used for such purposes as assessing what information is of most and least interest, determining technical design specifications, and identifying system performance or problem areas.
4. For site security purposes and to ensure that this service remains available to all users, these government computer systems employ software programs to monitor network traffic to identify unauthorized attempts to upload or change information, or otherwise cause damage.
5. Except for authorized law enforcement investigations, no other attempts are made to identify individual users or their usage habits. Raw data logs are used for no other purposes and are scheduled for regular destruction in accordance with AR 25-400-2 Army Records Information Management System (ARIMS) and National Archives and Records Administration General Records Schedule 20. All data collection activities are in strict accordance with Department of Defense Directive 5240.1.
6. Unauthorized attempts to upload information or change information on these services are strictly prohibited and may be punishable under the Computer Fraud and Abuse Act of 1987 and the National Information Infrastructure Protection Act.
7. If you have any questions or comments about this notification, please forward them to Joy Renfro, Information Information Assurance Program Manager, United States Army Corps of Engineers, E-Mail: Joy.Renfro@usace.army.mil.

Copyright Notice

Permission is hereby granted, without any fee, to use, copy, modify, and distribute the pages/files contained in the Websites named below for any legitimate, not-for-profit, non-commercial use. However, an identification of the original source and a direct link to it is required as shown below. No fee or any charge can ever be requested for the use of the information contained in these Websites.

In no event shall the maintainer/editor of, or any contributor to, these Websites, or any institution/organization supporting these Websites be liable for any cost, loss, or damage incurred due to the use, or misuse of these Websites or due to any inaccuracy of the information provided in these Websites.

The following **source identifications and links** are requested on any copy of the pages/files of the Websites.

Copied by permission from the "*Electrochemical Science and Technology Information Resource*" (ESTIR) (<http://electrochem.cwru.edu/estir/>) on "month/day/year." The original material is subject to periodical changes and updates.

or

Copied by permission from the "*Electrochemistry Encyclopedia*" (<http://electrochem.cwru.edu/encycl/>) on "month/day/year." The original material is subject to periodical changes and updates.

or

Copied by permission from the "*Electrochemistry Dictionary*" (<http://electrochem.cwru.edu/ed/dict.htm>) on "month/day/year." The original material is subject to periodical changes and updates.