DIPLOMA THESIS

NUMERICAL SIMULATION OF AIR FLOW AND THERMAL EFFECTS IN PLASMA REACTORS, USED FOR SEWAGE SLUDGE TREATMENT

Rention S. Pasolari

Student’s Number: 247019

Main supervisor: Polycarpos K. Papadopoulos, Assistant Professor

Diploma’s thesis submitted to the Department of Mechanical and Aeronautics of the University of Patras

PATRAS, JULY 2019
This diploma thesis presented by:

Rention S. Pasolari

Student’s Number: 247019

Dated: July 11, 2019
The approval of the present diploma thesis does not imply acceptance of the author's opinions. Writing complied with the principles of academic ethics.
Foremost, I would like to express my sincere gratitude to my main supervisor Prof. Polycarpos Papadopoulos for the patient guidance and advice he has provided in order to complete the present Diploma thesis.

Besides my main supervisor, I would like to acknowledge Prof. Panagiotis Svarnas from the Department of Electrical Engineering and Computers Science who gave me the chance to contribute via this thesis, to his experimental analysis. Moreover, I would like to thank the staff of the High Voltage Laboratory (HVL) of the Department of Electrical Engineering and Computers Science, University of Patras, for the collaboration on the specific topic.

Last but not least, I must express my very profound gratitude my parents, Spartak and Kozeta Pashollari, and my brother Erando for their encouragement throughout my time as a post-graduate student.
The aim of the present thesis is to study the fluid dynamic and the thermal effects of an Dielectric Barrier Discharge (DBD) plasma reactor which is used in order to decontaminate sewage sludge. It is an engineer’s duty to study technologies that are environmentally friendly, since such technologies can save not only Earth’s life, but also humans’ life. The decontamination of sewage sludge is such a technology due to the fact that the sludges in our country are unusable, which is a large waste, since we can exploit them in agricultural purposes. This thesis deals with the numerical analysis of the mentioned reactor, and the aims are to study the fluid dynamic characteristics of the flow and the thermal effects. First of all, we present the theory behind plasma and the way that cold plasma contribute to the treatment of contaminated soil and sewage sludge. Afterwards, a description of the DBD plasma reactor follows, i.e the apparatus that is used for the plasma generation and the decontamination of the sewage sludge. A significant part of the thesis, deals with the mesh generation. The geometry of the reactor is relatively complex, so a lot of tests were necessary for an appropriate mesh to be achieved. An appropriate mesh, is a mesh dense enough to converge to the correct solution, but without an excessive computational cost, at the same time. The next large part of the thesis has to do with the fluid dynamic and the thermal analysis of the problem. A new solver, based on OpenFOAM’s standard solvers, had to be developed for the purpose of solving the specific problem. The final section of the present thesis, considers the comparison of the numerical results with the experimental data.

Key words: [cold plasma, thermal analysis, fluid dynamic analysis, sewage sludge, OpenFOAM]
Τίτλος εργασίας: Υπολογιστική προσομοίωση ροής αέра και θερμικών φαινομένων σε αντιδραστήρες πλάσματος, με σκοπό τον καθαρισμό στερεών λυμάτων

Ο σκοπός της παρούσας διπλωματικής εργασίας, είναι η μελέτη της ρευστοδυναμικής ροής και των θερμικών φαινομένων σε αντιδραστήρα ψυχρού πλάσματος τύπου εκκενώσεως διηλεκτρικού φράγματος. Είναι χρήση κάθε μηχανικού, να ερευνά νέες τεχνολογίες οι οποίες θα προστατεύουν το περιβάλλον το οποίο καθημερινά υποδοθείται. Έτσι, και η συγκεκριμένη τεχνολογία, έχει ως στόχο της, τον καθαρισμό στερεών λυμάτων βιολογικού καθαρισμού, έτσι ώστε αυτά να είναι επαναχρησιμοποιήσιμα. Η ανάλυση που περιγράφεται παρακάτω, γίνεται ώστε να προσδιοριστούν τα ρευστοδυναμικά και θερμικά αποτελέσματα στο εσωτερικό ενός τέτοιου αντιδραστήρα, μιας και οι παράμετροι αυτές είναι ιδιαίτερα σημαντικές. Αρχικά γίνεται μια σύντομη παρουσίαση της θεωρίας που διέπει το πλάσμα, καθώς και τον μηχανισμό με τους οποίους το ψυχρό πλάσμα μπορεί να καθαρίσει τα στερεά λύματα. Στη συνέχεια, παρουσιάζεται η κατασκευή του υπολογιστικού πλέγματος, το οποίο είναι βασικό κομμάτι της προσομοίωσης. Ύστερα, γίνεται αναφορά του μαθηματικού και του υπολογιστικού μοντέλου που χρησιμοποιείται στην παρούσα προσομοίωση. Τέλος, παρουσιάζονται οι παράμετροι και τα αποτελέσματα της προσομοίωσης, καθώς και η σύγκριση τους με τα αντίστοιχα πειραματικά.

Λέξεις κλειδία: [ρευστοδυναμική ανάλυση, θερμική ανάλυση, ψυχρό πλάσμα, στερεά λύματα, OpenFOAM]
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Feature refinement level</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>Main refinement level</td>
<td>16</td>
</tr>
<tr>
<td>3.3</td>
<td>Snapping control parameters</td>
<td>18</td>
</tr>
<tr>
<td>3.4</td>
<td>Cell information for each region</td>
<td>20</td>
</tr>
<tr>
<td>5.1</td>
<td>Velocity boundary conditions-Fluid</td>
<td>46</td>
</tr>
<tr>
<td>5.2</td>
<td>Pressure boundary conditions-Fluid</td>
<td>46</td>
</tr>
<tr>
<td>5.3</td>
<td>Temperature boundary conditions-Fluid</td>
<td>47</td>
</tr>
<tr>
<td>5.4</td>
<td>Temperature boundary conditions-Solid</td>
<td>47</td>
</tr>
<tr>
<td>6.1</td>
<td>Solid materials properties</td>
<td>52</td>
</tr>
<tr>
<td>6.2</td>
<td>Solid matrices properties</td>
<td>54</td>
</tr>
<tr>
<td>6.3</td>
<td>Power</td>
<td>54</td>
</tr>
<tr>
<td>6.4</td>
<td>Energy distribution</td>
<td>56</td>
</tr>
</tbody>
</table>
List of figures

1.1 Cathode-directed streamer. ........................................... 3
2.1 The DBD reactor ...................................................... 8
2.2 The driven electrodes with the dielectric tubes, and the configuration of the suspender. ........................................... 8
2.3 The grounded electrode with its support. .......................... 9
3.1 Triangulated surfaces illustrated through \textit{paraview}. .......... 13
3.2 The background mesh .................................................. 15
3.3 \texttt{snappyHexMeshDict-castellatedMesh} controls ............... 17
3.4 \texttt{snappyHexMeshDict-snapping} controls ...................... 18
3.5 The \texttt{checkMesh} output ........................................... 19
3.6 The outer geometry after (a) the castellated mesh control (b) the snapping control. ........................................... 20
3.7 The cylinders after (a) the castellated mesh control (b) the snapping control. ........................................... 21
3.8 The (a) upper solid’s (b) driven electrodes (c) lower solid’s mesh .......................... 21
3.9 The outer cylinder mesh .............................................. 22
3.10 Internal mesh .......................................................... 22
3.11 \texttt{topoSetDict} for patches refinement .......................... 23
3.12 \texttt{refineMeshDict} ................................................... 24
3.13 \texttt{createPatchDict} .................................................. 24
3.14 The inlet patches ..................................................... 25
3.15 The outlet patch ...................................................... 25
4.1 grounded electrode (green), sewage sludge (red), thin air layer(blue) in y-normal cross-sectional plane .......................... 28
4.2 2D cross-section of a porous medium .................................. 32
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Regions definition-topoSet</td>
<td>38</td>
</tr>
<tr>
<td>5.2</td>
<td>The different regions in paraView</td>
<td>39</td>
</tr>
<tr>
<td>5.3</td>
<td>Porous zones definition-topoSet</td>
<td>40</td>
</tr>
<tr>
<td>5.4</td>
<td>Fluid region, blue-fluid, red-sewage sludge, green-grounded electrode</td>
<td>40</td>
</tr>
<tr>
<td>5.5</td>
<td>Solid parts</td>
<td>41</td>
</tr>
<tr>
<td>5.6</td>
<td>Momentum equation in OpenFOAM</td>
<td>42</td>
</tr>
<tr>
<td>5.7</td>
<td>fvOptions for porous media</td>
<td>43</td>
</tr>
<tr>
<td>5.8</td>
<td>Energy equation for the solid regions</td>
<td>44</td>
</tr>
<tr>
<td>5.9</td>
<td>Energy equation for the fluid region</td>
<td>44</td>
</tr>
<tr>
<td>5.10</td>
<td>setFields</td>
<td>45</td>
</tr>
<tr>
<td>5.11</td>
<td>Variable properties</td>
<td>45</td>
</tr>
<tr>
<td>5.12</td>
<td>Heat source definition</td>
<td>45</td>
</tr>
<tr>
<td>5.13</td>
<td>fvShcemes files</td>
<td>48</td>
</tr>
<tr>
<td>5.14</td>
<td>fvSolution files</td>
<td>49</td>
</tr>
<tr>
<td>5.15</td>
<td>controlDict file</td>
<td>50</td>
</tr>
<tr>
<td>6.1</td>
<td>Power fitting of the total power given to the reactor</td>
<td>55</td>
</tr>
<tr>
<td>6.2</td>
<td>Parallel circular plates</td>
<td>57</td>
</tr>
<tr>
<td>6.3</td>
<td>Pressure field at y-normal cross-sectional plane</td>
<td>58</td>
</tr>
<tr>
<td>6.4</td>
<td>Pressure drop due to the porous media</td>
<td>59</td>
</tr>
<tr>
<td>6.5</td>
<td>The velocity field at the x-normal and y-normal cross-sectional planes</td>
<td>60</td>
</tr>
<tr>
<td>6.6</td>
<td>z-normal slice</td>
<td>60</td>
</tr>
<tr>
<td>6.7</td>
<td>Velocity field in the plasma and porous zones vicinity</td>
<td>61</td>
</tr>
<tr>
<td>6.8</td>
<td>Temperature distribution for different moments</td>
<td>61</td>
</tr>
<tr>
<td>6.9</td>
<td>The temperature distribution in the plasma region at (a) x-normal and (b) y-normal cross-sectional planes</td>
<td>62</td>
</tr>
<tr>
<td>6.10</td>
<td>The temperature distribution in the solid regions</td>
<td>62</td>
</tr>
<tr>
<td>6.11</td>
<td>Temperature field at 20 minutes</td>
<td>63</td>
</tr>
<tr>
<td>6.12</td>
<td>Temperature curve</td>
<td>63</td>
</tr>
<tr>
<td>6.13</td>
<td>The points that the temperature was measured</td>
<td>64</td>
</tr>
<tr>
<td>6.14</td>
<td>Comparison between computational and experimental results</td>
<td>65</td>
</tr>
<tr>
<td>A.1</td>
<td>The outer geometry of the reactor</td>
<td>68</td>
</tr>
<tr>
<td>A.2</td>
<td>Regions</td>
<td>69</td>
</tr>
<tr>
<td>A.3</td>
<td>Regions</td>
<td>70</td>
</tr>
<tr>
<td>B.1</td>
<td>Three dimensional control volume</td>
<td>74</td>
</tr>
<tr>
<td>B.2</td>
<td>One dimensional control volume</td>
<td>75</td>
</tr>
<tr>
<td>C.1</td>
<td>The PISO algorithm flowchart</td>
<td>81</td>
</tr>
</tbody>
</table>


Contents

Acknowledgment ix
Abstract xi
Περίληψη xiii
List of tables xv
List of figures xvii
Contents xix
Nomenclature xxiii

1 Introduction 1
  1.1 How are plasmas made? ................................................. 2
  1.2 The concept of streamers ............................................... 3
    1.2.1 How does an avalanche transforms into a positive streamer? 3
  1.3 Applications of plasma .............................................. 4

2 Cold Plasma for Treatment 5
  2.1 Soil and sewage sludge contamination ................................. 5
  2.2 How does plasma decontaminate sewage sludge? ...................... 7
  2.3 The Dielectric Barrier Discharge reactor .............................. 7

3 Mesh Generation 11
  3.1 snappyHexMesh ...................................................... 12
    3.1.1 Triangulated surfaces .......................................... 12
    3.1.2 Background mesh ................................................ 14
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.3</td>
<td>snappyHexMeshDict</td>
<td>14</td>
</tr>
<tr>
<td>3.2</td>
<td>Mesh quality controls</td>
<td>19</td>
</tr>
<tr>
<td>3.3</td>
<td>Results</td>
<td>20</td>
</tr>
<tr>
<td>3.4</td>
<td>Inlet and outlet definition</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>Mathematical Modeling</td>
<td>27</td>
</tr>
<tr>
<td>4.1</td>
<td>Description of physics of the problem</td>
<td>27</td>
</tr>
<tr>
<td>4.2</td>
<td>Fluid region</td>
<td>29</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Continuity equation</td>
<td>29</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Momentum equation</td>
<td>30</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Energy equation</td>
<td>30</td>
</tr>
<tr>
<td>4.3</td>
<td>Solid regions</td>
<td>31</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Energy equation</td>
<td>31</td>
</tr>
<tr>
<td>4.4</td>
<td>Porous media</td>
<td>32</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Momentum equation</td>
<td>32</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Direct modeling approach</td>
<td>33</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Continuum modeling</td>
<td>33</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Energy equation</td>
<td>34</td>
</tr>
<tr>
<td>4.5</td>
<td>Evaporation of water</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>Computational Modeling</td>
<td>37</td>
</tr>
<tr>
<td>5.1</td>
<td>Definition of regions</td>
<td>37</td>
</tr>
<tr>
<td>5.2</td>
<td>Definition of different materials in a region</td>
<td>39</td>
</tr>
<tr>
<td>5.3</td>
<td>Solving mathematical equations with OpenFOAM</td>
<td>41</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Fluid dynamics problem</td>
<td>41</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Energy equation</td>
<td>43</td>
</tr>
<tr>
<td>5.4</td>
<td>Boundary conditions</td>
<td>46</td>
</tr>
<tr>
<td>5.5</td>
<td>Solution</td>
<td>47</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Discretization</td>
<td>48</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Algorithm control</td>
<td>48</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Control</td>
<td>49</td>
</tr>
<tr>
<td>6</td>
<td>Problem parameters and results</td>
<td>51</td>
</tr>
<tr>
<td>6.1</td>
<td>Problem parameters</td>
<td>51</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Material properties</td>
<td>52</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Power</td>
<td>54</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Assumptions</td>
<td>56</td>
</tr>
<tr>
<td>6.2</td>
<td>Results</td>
<td>58</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Pressure field</td>
<td>58</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Velocity field</td>
<td>59</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Temperature field</td>
<td>61</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>6.3 Comparison with the experimental results</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>6.4 Conclusions</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>6.5 Future work</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>A Designs</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>A.1 Reactor</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>A.2 Upper solid configuration</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>A.3 Lower solid configuration</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>B Finite volumes method</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>B.1 Grid topology</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>B.2 Discretization</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>C PISO Algorithm</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>Bibliography</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>
Symbols

\( A \) \quad \text{area}
\( c \) \quad \text{speed of sound}
\( c_p \) \quad \text{specific heat capacity}
\( Co \) \quad \text{Courant number}
\( F \) \quad \text{field forces}
\( F_r \) \quad \text{Forschheimer coefficient}
\( F_{1\to2} \) \quad \text{view factor from surface 1 to surface 2}
\( g \) \quad \text{acceleration of gravity}
\( h_{fg} \) \quad \text{specific enthalpy of phase change (fluid to gas)}
\( k \) \quad \text{thermal conductivity}
\( K \) \quad \text{permeability coefficient}
\( M \) \quad \text{molar mass}
\( Ma \) \quad \text{Mach number}
\( p \) \quad \text{pressure}
\( \dot{q} \) \quad \text{heat power per unit volume}
\( \dot{Q} \) \quad \text{heat power}
\( R \) \quad \text{universal gas constant}
\( S_m \) \quad \text{source term derived from Darcy-Brinkman-Stokes law}
\( t \) \quad \text{time}
\( T \) \quad \text{temperature}
\( V \) \quad \text{velocity}

Greek symbols

\( \alpha \) \quad \text{thermal diffusivity}
\( \beta \) \quad \text{coefficient of thermal expansion}
\( \epsilon \) \quad \text{porosity of the medium}
$\epsilon_a$  emissivity factor
$\mu$  dynamic viscosity
$\nu$  kinematic viscosity
$\rho$  density
$\sigma$  Stefan-Boltzmann constant
$\tilde{\tau}$  viscous shear tensor

**Special symbols**

$\partial$  partial derivative
$D$  substantive derivative
$\nabla$  nabla operator
$\nabla^2$  Laplace operator

**Abbreviations**

CFD  computational fluid dynamics
DBD  dielectric barrier discharge
fvm  finite volume method
fvc  finite volume calculus
N-S  Navier-Stokes
OBJ  wavefront object format
PISO  pressure implicit with splitting of operator
REV  representative elementary volume
STL  stereolithography format
SIMPLE  semi-implicit method for pressure linked equations
Chapter 1

Introduction

It is generally known that when a solid is heated enough, the thermal motion of the atoms, which constitute it, break the crystal lattice structure apart, leading to the formation of a liquid. Moreover, when the thermal energy is received by a liquid, the atoms obtain high amount of kinetic energy and as a result a gas is formed. The aforementioned states of matter, are not the only existing. There is a fourth one, which is formed when a gas is heated enough, that the atoms collide with each other and knock their electrons off. This state of matter is called plasma. Plasma is an ionized gas. In the bibliography, one can encounter the definition below:

"Plasma is a macroscopically neutral ionized gas, which consists of charged and neutral particles."

The charged particles are mainly electrons, anions and cations. However, in several cases, depending on the gas nature and the plasma generation method, photons and other excited particles can be observed. Although, plasma is macroscopically neutral, due to the fact that the positive charge is equal to the negative one, it is not microscopically neutral. This implies that locally, plasma may be charged positively of
negatively. This property makes plasmas a quasi-neutral gas. In addition, plasma has collective behavior. That derives from its composition. The differently charged particles behave in a collective way because of attractive and repelling electric forces. Plasma is highly electrically conductive to the point that long-range electric and magnetic fields dominate the behavior of the matter.

Even though plasma was first described relatively recently (specifically in the 1920s by the chemist Irving Langmuir), it dominates the world. More than 90% of the universe consists of plasma, e.g., the Sun, the stars and the ionosphere.

Plasma is divided into three categories, depending on the temperature of the most heavy particles:

- Cold (or non-thermal) plasma, where ions and neutral particles are preserved into low temperatures, while electrons obtain high temperatures. Cold plasma is used for several operations, such as the engraving of materials, the decontamination of soil and the sterilization of medical devices.

- Thermic plasma, which reaches high temperatures (i.e., more than 3,000K), and is used primarily for metal welding and cutting.

- Thermal plasma, which reaches temperatures more than 1,000,000K. This type of plasma is used for thermonuclear fusion.

1.1 How are plasmas made?

Heating a gas is not the only way to create plasma. Specifically, in most cases this way is not feasible due to the fact that the temperatures that must be reached are excessively high, resulting in barriers considering its storage. The main methods that are used instead of heating are:

- Driving an electric current through a gas.

- Subjecting a gas to electromagnetic waves at the frequencies of radio waves.

These means of plasma formation give energy directly to electrons. After that, the collisions between electrons and atoms release more electrons, leading to the generation of plasma. Electrons are often in higher temperatures than the ions, since electrons carry the electrical current or absorb the radio waves.
1.2 The concept of streamers

When a gas is placed between two electrodes, in a strong electric field, there are electrons that escape the positively nuclei hold. Due to the electric field, these electrons are accelerated and collide with the neutral molecules of the gas. Thus, more electrons and positive ions rise. This phenomenon continues, creating the so-called primary avalanche. When an avalanche reaches sufficiently high amplification, the avalanche transforms into a streamer.

The streamer is a weakly ionized thin channel, and constitutes the precursor of a spark discharge in the gap. It can grow, in one or both directions, toward the electrodes. When the direction is from the anode to the cathode, the streamer is called cathode-directed or positive. Otherwise an anode-directed or negative streamer is formed.

1.2.1 How does an avalanche transforms into a positive streamer?

During the collisions of the electrons with the neutral molecules in the primary avalanche, energetic photons are emitted. These photons lead to an ionization in the primary avalanche, producing, in this way, more free electrons, and a secondary avalanche. The electrons, are pulled due to the strong electric field, to the anode. In their path, they collide with other neutrals creating more electrons and positively charged ions, which are pulled from the cathode. Meanwhile, a channel is being formed with a positively charged head. This head pulls new electrons, the phenomenon is repeated and the streamer grows. This process is shown in Figure(1.1).

The formation criterion of streamer, is that the field of the avalanche’s space charge, must reach a value of the order of the external field.

The same process takes place at the anode-directed streamers. The only difference is that the electrons are accelerated out of the channel, owing to the fact that the head of the channel is negatively charged.
1.3 Applications of plasma

Plasma is used in a large extent nowadays. It has unique properties, which can be utilized in several areas of technology. Some of the usages of plasma are cited below:

1. Nuclear fusion.
2. Propulsion of spacecrafts.
3. Medical purposes.
5. Ozone production.
7. Solid surfaces treatment (e.g., polymers and metals).
8. Decontamination purposes (e.g., water and soil).
Cold Plasma for Treatment

It is obvious that nowadays, the environment has put into danger. Hence, human has turned his attention to finding new technologies that are environment friendly. The present thesis, deals with such a technology which focuses on the remediation of soil and sewage sludge (wastes). Specifically, the main axis is the treatment of sewage sludge, but the same method can be applied for soils.

2.1 Soil and sewage sludge contamination

The problem of soil contamination by organic pollutants is at very serious levels. Whereas the contamination of the atmosphere and the water can be easily observed, the soil contamination needs a lot of study to be detected. This situation makes the soil pollution highly dangerous for the human health.

Moreover, in most of cases, the sewage sludge which is produced from the biological treatment systems are useless. This is also a damage for the environment, due to the
fact that sewage sludges are polluted, but also because they are potential environment-friendly fertilizers. Thus, if there is an easy and low-power mean of decontaminating sewage sludge, it is sure that it will become one of the most useful fertilizers in agriculture.

Many studies have been conducted on this topic, and have concluded that soil and wastes are contaminated by a variety of elements, the most important of them being:

- Organic contaminants.
- Heavy metals.
- Pesticides.
- Polycyclic aromatic hydrocarbons (PAH’s).
- Dichlorodiphenyltrichloroethane (DDT).

A wide range of human activities lead to the pollution of soil and wastes. Most of them can be grouped together in order to describe sufficiently the main causes of the contamination. These groups are shown below:

1. Contamination from agricultural activities. In this category, chemical pesticides and fertilizers play the most important role to the soil contamination.

2. Contamination from industrial processes. Most of the industrial areas worldwide, suffer from atmospheric, water and soil pollution, owing to the uncontrolled emission of several industrial contaminants.

3. Contamination from atmospheric deposition. This is a crucial problem, mainly in large cities, and secondarily in rural areas. The most important emission sources of the contamination, are the increasing traffic exhaust flumes and the fuel burning.

4. Contamination of sewage sludges owing to the detergents that are used immoderately.

It is obvious that drastic measures must be taken in order to eliminate this urgent issue. There is a variety of existing technologies for soil and sewage sludge remediation, such as physical remediation, bioremediation and chemical remediation.

In recent years a new technology has been adopted. This technology is the remediation using cold plasma. The philosophy behind this way of remediation, is that plasma consists of highly energetic electrons. These electrons collide with the molecules of the gas that is used for producing the plasma, breaking down their bonds and so, a high number of energetic and chemically reactive species is produced (e.g., excited atoms, ions, free radicals). After that, the produced species are used for an extensive number of chemical reactions, for the decontamination of the solid (soil or wastes).

Cold plasma (or non-thermal plasma) can be generated by different methods. These methods are:
Corona discharge.

Non-thermal gliding arc fluidized bed.

Dielectric barrier discharge (DBD) reactor.

The present thesis deals with the DBD method of generating plasma. An extensive description of the reactor is given in the section 2.3.

2.2 How does plasma decontaminate sewage sludge?

The plasma generation process, leads to the generation of chemically reactive species. In the DBD reactor that was described above, the gas that is used is atmospheric air. Thus, the produced reactive species are highly energetic electrons, $O_3$ (ozone) molecules, OH radicals, $NO_x$ etc. Each of these species, plays a significant role on the decontamination of the soil, by reacting with the contaminants. For instance, $O_3$ oxidizes the organics directly and generates OH radicals by reacting with other active species. It has been observed, that even the most persistent soil pollutants can be oxidized with this method.

2.3 The Dielectric Barrier Discharge reactor

Dielectric barrier discharge (DBD) is one of the main ways that non-thermal plasma can be generated. It is also known as ozone production discharge, due to the fact that large amounts of ozone ($O_3$) are produced during the plasma generation. The philosophy behind this reactor, is that a discharge takes place between two electrodes, separated by one (or more) dielectric barriers. The dielectric’s purpose is to prevent the transition to an electric arc. Specifically, the existence of the dielectric restricts the conductivity current between the two electrodes. The most common materials used for the dielectric barrier are glass, quartz or ceramic.

The apparatus used in the analysis which follows, is shown below:

The main body of the reactor is cylindrical. In the perimeter of the main body there are four windows, in order to observe the phenomenon, and also for taking measurements during the plasma generation and the decontamination process.

The gas that is used for this process is atmospheric air. The air is inserted through four symmetric inlets, located at the top of the reactor. On the other hand, at the bottom of the reactor there is an outlet, which is the only way that gas can escape.
In the figures that are following, one can see the reactor’s interior geometry characteristics.

Figure 2.2 shows the driven electrodes, the dielectric tubes and the configuration of the suspender. The driven electrodes, are eight INOX cylinders, placed one beside the other, in order to approach a plate electrode setup. The driven electrodes are surrounded by dielectric tubes, made of quartz. The grounded electrode is placed eight milimeters (8mm) under the dielectric,(figure 2.3). The grounded electrode is an INOX plate with 840 holes of 1-1.1mm diameter. The sewage sludge that must be treated, is located at the top of the grounded electrode, having a thickness of 4mm. An alternating voltage in order of 15-20 kV is imposed on the driven electrodes, to initiate the plasma generation process.
In the appendix A, one can see the full design of the reactor.

The DBD process for the treatment of soil (and sewage sludge consequently), has several advantages which make it one of the most efficient methods. First of all, the power consumption of the process is small, comparably with the energy consumption of a small light bulb. Moreover, experiments have revealed that, 20-30 minutes of the aforementioned process, are able to eliminate the most resistant pollutants. Finally, this reactor can be easily scaled up in order to be used for commercial purposes.
Chapter 3

Mesh Generation

This chapter aims to the generation of the computational mesh. The mesh is an integral part of the Computational Fluid Dynamics science, and the computational physics in general. Without a high quality mesh, an approach to the correct solution is not feasible.

The geometry of the specific reactor, is complex enough, hence special attention was paid in order to generate a high quality mesh. Due to the fact of its complexity, a grid independence study was very difficult to be held. For that reason, the mesh that was selected, is dense enough. However, it was taken into account, that the problem must be solved by the available computers, into sensible time periods.

The main characteristics that make the geometry to be complex are the following:

- The outer cylindrical geometry.
- The four cylindrical windows in the perimeter of the main body.
- The driven electrodes and the dielectric tubes geometry.

Whereas, the first problem was easy to get over by constructing the mesh via a
cylindrical mesh generator code, the rest were not too easy to be overcome. On account of this, the mesh was generated by an utility of OpenFOAM, appropriate for constructing complex geometries, called snappyHexMesh. The main barrier was the meshing of the driven electrodes and the dielectric tubes setup. There is a little space of 2mm between each dielectric tube, and for this reason, the specific area of the domain had to be such a lot dense, in order to achieve at least one cell of air between the tubes. The way that such a problem can be exceeded is by using local densification.

3.1 snappyHexMesh

The snappyHexMesh utility generates three-dimensional (3D) meshes, which consist of hexahedra and split-hexahedra, automatically from triangulated surface geometries, in Stereolithography (STL) or Wavefront Object (OBJ) format. The mesh conforms to the given geometries via an iterative process on an initial mesh that is called background mesh. In order to use snappyHexMesh, the user must add the following files:

1. The triangulated surfaces in Stereolithography (STL) or Wavefront Object (OBJ) format, located in the constant/triSurfaces sub-directory of the case.
2. An initial background mesh that defines the size of the computational domain and a first mesh density.
3. The snappyHexMesh dictionary, which contains all the necessary settings so as the snappyHexMesh to run. This dictionary must be located in the system directory of the case.

In the next paragraphs, a detailed description of the aforementioned files for the specific problem follows.

3.1.1 Triangulated surfaces

With the aim of achieving the exact geometry of the reactor, every part of it was designed in AutoCAD. Having exported them to STL format, their import into OpenFOAM is an easy part. In the figures below, the designs are illustrated through paraView (an open-source application for data analysis and visualization), grouped in six categories:
• The outer cylinder geometry.
• The cylinders that contain the driven electrodes and the dielectric tubes.
• The configuration that the driven electrodes are suspended.
• The lower solid geometry which is, in essence, the support of the grounded electrode.
• The grounded electrode.
• The sewage sludge.

Figure 3.1: Triangulated surfaces illustrated through \textit{paraView}.
It should be noted, that whereas all the cylinders are depicted into one group (Figure 3.1b), there is a design file for each of them. This will be more clear in the section 3.1.3.

### 3.1.2 Background mesh

As a first attempt, the background mesh was chosen to be cylindrical. Despite the fact that such a mesh gives a high resolution considering the outer cylinder, is unsuitable for the inner characteristics of the reactor. Thus, a cuboid background mesh was finally selected. There are three points that the generated background mesh must fulfill in order to run correctly and give better results. These three points are mentioned below:

- This mesh must be comprised purely of hexahedras (hexes).
- The cell aspect ratio must be approximately 1, notably at the areas that the snappyHexMesh is about to operate.
- There must be at least one intersection between this mesh and the trisurface geometry.

Another key issue considering the background mesh, is that there is no limitation which indicates that all the cells composing it, must be the same. Thus, one can dense or disperse the mesh. For the needs of the present thesis, a highly dense mesh was used at the area of the driven electrodes. The background mesh which used is shown in the Figure 3.2.

### 3.1.3 snappyHexMeshDict

As it was mentioned above, snappyHexMeshDict is the dictionary responsible for all the settings that snappyHexMesh needs to run. It is divided into three steps:
1. Castellated mesh controls.

2. Snapping process.

3. Adding layers process.

Now, is following a description of the parameters that were used in order to generate the final mesh for the present thesis.

**Castellated mesh controls**

Castellated mesh controls, is a sub-dictionary that deals with the level of refinement for each STL surface, such as the removal of the unused cells. Before the main refinement, another dictionary is used, called `surfaceFeatureExtractDict`, which specifies a level for any cell intersected by its edges. In the table below, the feature refinement level for each imported surface is depicted.

<table>
<thead>
<tr>
<th>Surface</th>
<th>Feature refinement level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylinders group</td>
<td>level 2</td>
</tr>
<tr>
<td>Grid</td>
<td>level 1</td>
</tr>
<tr>
<td>Porosity</td>
<td>level 1</td>
</tr>
<tr>
<td>Outer cylinder</td>
<td>level 1</td>
</tr>
<tr>
<td>Configuration of the suspender</td>
<td>level 1</td>
</tr>
<tr>
<td>Lower solid</td>
<td>level 1</td>
</tr>
</tbody>
</table>
Consequently the main refinement level of each surface is defined. After a few attempts, it was concluded that the levels that are mentioned below, lead to the optimal mesh.

<table>
<thead>
<tr>
<th>Surface</th>
<th>Main refinement level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylinders group</td>
<td>level (1 3)</td>
</tr>
<tr>
<td>Grid</td>
<td>level (1 1)</td>
</tr>
<tr>
<td>Porosity</td>
<td>level (1 1)</td>
</tr>
<tr>
<td>Outer cylinder</td>
<td>level (0 0)</td>
</tr>
<tr>
<td>Configuration of the suspender</td>
<td>level (1 1)</td>
</tr>
<tr>
<td>Lower solid</td>
<td>level (1 1)</td>
</tr>
</tbody>
</table>

The two numbers indicate the lower and the higher level of refinement respectively. It is obvious that cylinders need the higher refinement level due to their complexity.

The final parameter to be defined, is the locationInMesh. All the refinement surfaces are kept into patches. The locationInMesh specifies a point into the domain, where the patch reachable from this point is kept. In the present problem this point is the (0 0 0.05). The structure of the castellated mesh controls file, is illustrated in the Figure 3.3.
Figure 3.3: snappyHexMeshDict-castellatedMesh controls
Snapping controls

Having removed the unused cells, the next step is to make the the geometry smoother. This can be done via the snapControls sub-dictionary. There are some parameters which control the iterative process behind the snapping. The selection of them was made after some attempts and comparing the results.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>nSmoothPatch</td>
<td>5</td>
</tr>
<tr>
<td>tolerance</td>
<td>2.0</td>
</tr>
<tr>
<td>nSolveIter</td>
<td>100</td>
</tr>
<tr>
<td>nRelaxIter</td>
<td>10</td>
</tr>
<tr>
<td>nFeatureSnapIter</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 3.3: Snapping control parameters

In the figure below, the way that the snapping controls sub-dictionary is structured, is shown:

```csharp
// Settings for the snapping.
ssnapControls {
    // Number of patch smoothing iterations before finding correspondence
    // to surface.
    nSmoothPatch 5; //3
    // Relative distance for points to be attracted by surface feature point
    // or edge. True distance is this factor times local maximum edge length.
    tolerance 2.0;
    // Number of mesh displacement relaxation iterations.
    nSolveIter 100;
    // Maximum number of snapping relaxation iterations. Should stop
    // before upon reaching a correct mesh.
    nRelaxIter 10;  //5
    // Highly experimental and wip: number of feature edge snapping
    // iterations. Leave out altogether to disable.
    // Of limited use in this case since faceZone faces not handled.
    nFeatureSnapIter 8; //4
    // Detect (geometric only) features by sampling the surface
    // (default=false).
    implicitFeatureSnap false;
    // Use castellatedMeshControls::features (default = true)
    explicitFeatureSnap true;
    // Detect points on multiple surfaces (only for explicitFeatureSnap)
    multipleRegionFeatureSnap false; //true
}
```

Figure 3.4: snappyHexMeshDict-snapping controls
**Adding layers**

The final sub-dictionary is the addLayers. This is used commonly for adding cell layers near the surfaces in order to get better results, e.g., when a boundary layer study is necessary. In the specific problem there is no such a need, so this step is omitted.

### 3.2 Mesh quality controls

Running the problem, despite the fact that the mesh seemed to be acceptable, the check mesh control failed. Specifically, there were a lot of skewed faces, that may ruin the simulation. Investigating closer the snappyHexMesh, it was observed that the mesh quality limits that this process sets, are different to the OpenFOAM’s. Hence, setting the `maxBoundarySkewness` and the `maxInternalSkewness` to 2 and 1 correspondingly, this problem was overcome. In the Figure 3.5, the final check mesh output is illustrated.

![Figure 3.5: The checkMesh output](image)

As it is observed, the generated mesh complies with the requirements, and each check seems to be correct. Apart from the quality of the mesh, the checkMesh command gives information about the total cells in the computational domain, as well as the type of these cells. In table 3.4, the information about the cells of each domain are depicted.

It must be mentioned that the fluid region contains also the sewage sludge and the grounded electrode. Moreover, the upper solid region consists of the driven electrodes, the dielectric and the configuration of the suspender. Finally, the lower solid region is the solid that support the grounded electrode. It is obvious that there is large amount of cells in the upper solid region. However, this is the minimum number of cells in order to achieve the specific geometry.
Table 3.4: Cell information for each region

<table>
<thead>
<tr>
<th>Region</th>
<th>No of cells</th>
<th>Hexahedra</th>
<th>Prisms</th>
<th>Tet wedges</th>
<th>Polyhedra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluid</td>
<td>1185695</td>
<td>1074450</td>
<td>12411</td>
<td>208</td>
<td>98626</td>
</tr>
<tr>
<td>Upper Solid</td>
<td>1121243</td>
<td>986548</td>
<td>-</td>
<td>-</td>
<td>134695</td>
</tr>
<tr>
<td>Lower Solid</td>
<td>200897</td>
<td>181431</td>
<td>1172</td>
<td>40</td>
<td>18254</td>
</tr>
<tr>
<td>Total</td>
<td>2507835</td>
<td>2242429</td>
<td>13583</td>
<td>248</td>
<td>251575</td>
</tr>
</tbody>
</table>

3.3 Results

In this section, the results of the process that was described above, are illustrated. First of all, the outer cylinder and the driven electrode geometry that derived from the castellated and the snapping process respectively, are shown.

![Figure 3.6: The outer geometry after (a) the castellated mesh control (b) the snapping control.](image)

Figure 3.6: The outer geometry after (a) the castellated mesh control (b) the snapping control.
Figure 3.7: The cylinders after (a) the castellated mesh control (b) the snapping control.

The effect of each process is obvious. Castellated mesh control, achieves a first approach of the surface, while snapping makes it smooth.

After that, the cells of the upper solid, the lower solid and the electrodes are depicted.

Figure 3.8: The (a) upper solid’s (b) driven electrodes (c) lower solid’s mesh
Finally, in the Figure 3.9, the mesh of the outer cylinder is depicted, while in Figure 3.10 the internal mesh of the fluid region and the area where plasma is generated, which has also the most computational interest, are illustrated.

![Figure 3.9: The outer cylinder mesh](image)

![Figure 3.10: Internal mesh](image)

### 3.4 Inlet and outlet definition

The final step, before the computational domain is completed, is the definition of the inlet and outlet patches. The specific DBD plasma reactor has four cyclic inlet in the perimeter of the top, and one cyclic outlet in the bottom of the central axis. In order to achieve the shape of these patches, the vicinity of them must be refined. First of all,
using the `topoSet` utility, the areas that have to be refined, are defined. `TopoSet`, is a utility of OpenFOAM, where one can group cells or faces of the computational domain into sets or zones. The `topoSet` dictionary, is shown below:

```
--- C 47.3d ---
\ And multiplication
Foamtite
{
    version 2.0;
    format ascii;
    class dictionary;
    object topoSetDict;

    // actions
    {
        name refinedPatches;
        type cellSet;
        action new;
        source cylinderToCell;
        sourceInfo
        {
            p1 (-0.038 0.038 0.21);
            p2 (-0.038 0.038 0.2148);
            radius 0.0035;
        }
    }

    // actions
    {
        name refinedPatches;
        type cellSet;
        action add;
        source cylinderToCell;
        sourceInfo
        {
            p1 (-0.038 -0.038 0.21);
            p2 (-0.038 -0.038 0.2148);
            radius 0.0035;
        }
    }

    // actions
    {
        name refinedPatches;
        type cellSet;
        action add;
        source cylinderToCell;
        sourceInfo
        {
            p1 (0 0 0);
            p2 (0 0 0.002);
            radius 0.0035;
        }
    }
}
```  

![Figure 3.11: topoSetDict for patches refinement](image)

Having defined, the refined areas, the next step is to make the refinement. This can be achieved, using the `refineMesh` utility. As one can see in the Figure 3.12, the `refineMesh` dictionary, gives user the option to refine in any of the directions in needed. Here, the refinement takes place in all directions.

If a further refinement level is needed, the aforementioned process should be repeated. For the needs of this simulation, the process was repeated twice. After that, the `topoSet` dictionary is used again, for the purpose of defining the exact dimensions of the inlet and outlet patches.

The final step of the patches definition, is to run the `createPatch` command. It is obvious that this utility creates patches of the cellSets that was imported in the `topoSet` dictionary. In the Figure 3.13 the `createPatch` dictionary is depicted.

This is where the mesh generation process is completed. In the next figures (3.14 and 3.15), the inlet and the outlet patches are illustrated.
Figure 3.12: \textit{refineMeshDict} \\

Figure 3.13: \textit{createPatchDict}
Figure 3.14: The inlet patches

(a) The four inlet patches (red color)
(b) Mesh of the four inlets

Figure 3.15: The outlet patch

(a) The outlet patch (red color)
(b) Mesh of the outlet patch
In the previous chapter, the computational domain of the problem was presented. This chapter deals with the differential equations that must be solved in this domain. These are the main equations that one will encounter into a computational fluid dynamic problem, e.g., the continuity equation, the Navier-Stokes or momentum equations and finally the energy equation. It must be noted that the computational mesh is divided into three parts, each of them to be governed by different form of the aforementioned equations. The three regions are described below.

4.1 Description of physics of the problem

First of all, the fluid dynamic problem deals with an air flow, in which air is inserted in the reactor from four circular inlets, passes through the sewage sludge and the grounded electrode and finally it exerts from the outlet which is located at the bottom of the reactor (the location of the sewage sludge and the grounded electrode are illustrated in figure
4.1). It is obvious that solving the fluid dynamic problem requires the solution of the governing equations, which are the continuity equation and the Navier-Stokes equations. However, the existence of the porous media (both the sewage sludge and the grounded electrode, which is actually a perforated plate, are treated as porous media), entails the solution of specific equations that govern flows in porous media. As it is described below, this equation is a combination of the N-S equation and the Darcy’s law. The aforementioned equations complete the fluid dynamic problem.

![Grounded electrode, sewage sludge, thin air layer in y-normal cross-sectional plane](image)

**Figure 4.1**: grounded electrode (green), sewage sludge (red), thin air layer (blue) in y-normal cross-sectional plane

On the other hand, the thermal problem is more complicated. There are several parameters which, one has to take into account. Nevertheless, there are not enough data, in order to approach the entire problem with the exact phenomena that occur in the reactor. Hence, the approach that is used, is based, at a large proportion, on the experimental data, taking into account the natural laws. The effect of the plasma is approached as heat sources in the energy equations. The first point to be discussed, is that the way that the supplied power is distributed into the reactor, is not known. The approach that is used, is that the energy used for chemical reactions, and the heat that escapes to the atmosphere can be neglected. This assumption, rest on experimental measurements. Moreover, experiments have shown that the initial state of the sewage sludge differs significantly to the final one, due to the fact that the vast percentage of the water that comprises it, has been evaporated. The mechanism of the evaporation in porous media is difficult to be modeled in CFD, because it involves complex mass and heat transfer mechanisms. When the water on the surface of the sludge gets evaporated due to the action of the plasma, the water that is inside the sludge is diffused and replaces it. In that way, the evaporation procedure continues. It is very important to mention that the air which is used is dry, and it reaches high temperatures, thus it can keep large amounts of water as moisture inside it. The way that this phenomenon is included here, is by computing the overall energy that the specific amount of water needs to change its phase, and subtracting it from the total power. In other words, observing the power curve through time, one can see the time period where the most of
the water is evaporated. Integrating the power, the energy that is provided in the specific time period is derived. Hence, it is easy to calculate the ratio of the energy needed for evaporation to the total energy, and so find the energy that is given to the reactor. This means that the evaporation curve, follows the power curve, so the evaporation rate is not constant. This model seems to be close to the real mechanism of the evaporation. Finally, another point to be noted is that the energy used for increasing the temperature in the reactor, is not imposed to the driven electrodes. Previous researches [9] have shown that the heat is distributed in different parts of the DBD reactor in a complex way, affected by the velocity of the gas, the nature of the gas etc. Here, the experimental observations, as well as several different simulations, showed that the most appropriate way to distribute the heat in the reactor is:

- 52.5% of the power to the dielectric tubes
- 47.5% of the power to a thin air layer above the sewage sludge

This air layer is shown again in figure 4.1. It has to be mentioned that these values refer to the power after subtracting the evaporation energy.

4.2 Fluid region

The first region is the fluid, i.e., the vast percentage of the space inside the reactor. In the fluid domain the continuity equation, the Navier-Stokes set of equations and the energy equation were solved. The form of the equations is given below:

4.2.1 Continuity equation

The general form of the continuity equation is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V})$$  \hspace{1cm} (4.1)

where $\mathbf{V}$ is the velocity vector, and $\rho$ the density of the fluid.

It will be shown in chapter 6, the maximum Mach number is: $Ma_{max} = 2.36 \cdot 10^{-3}$. Hence, the flow can be considered incompressible. For an incompressible flow the continuity equation is written as:

$$\nabla \cdot \mathbf{V} = 0$$  \hspace{1cm} (4.2)
4.2.2 Momentum equation

The momentum equation (or Navier-Stokes equation), in its general form, can be applied in every flow and for every fluid. As it mentioned above, this specific problem manages an incompressible flow, thus the momentum equation can be written as follows:

\[
\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{V} + \frac{1}{\rho} \mathbf{F} \tag{4.3}
\]

where \( p \) is the pressure field, \( \nu \) the kinematic viscosity and \( \mathbf{F} \) the field forces acting on the fluid.

On the left hand side, the first term is the time derivative, and the second one represents the advection term. On the right hand side the first term is the pressure gradient, the second one is the diffusion term, which describes the viscous forces, and finally the third term defines the field forces. The field forces term consists of various forces, such as gravitational, buoyancy, electromagnetic etc.

For the flow of this problem, the field forces that act on the fluid are the gravitational and the buoyancy due to temperature differences. However, as will be discussed in chapter 6, the velocities that are used, are high enough to set these field forces negligible.

Taking into consideration the above, the Navier-Stokes equations can be rewritten in this form:

\[
\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla) \mathbf{V} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{V} \tag{4.4}
\]

4.2.3 Energy equation

One of the most important information outcomes of this analysis, is the temperature field. In this respect, the energy equation has to be solved. The energy equation can be written in different ways, such as in terms of enthalpy, internal energy or temperature. Here, it is presented in terms of temperature. The general form of the energy equation is

\[
\rho c_p \frac{DT}{Dt} = \mathbf{\nabla} \cdot (k \mathbf{\nabla} T) + \beta T \frac{Dp}{Dt} + (\mathbf{\tau} : \mathbf{\nabla} \mathbf{V}) + \dot{q} \tag{4.5}
\]

where \( \rho \) is the fluid’s density, \( c_p \) the specific heat capacity, \( \beta \) the coefficient of thermal expansion and \( \mathbf{\tau} \) the viscous stress tensor.

The temperature increase in the reactor is small, hence density is considered independent from the temperature. Moreover the fluid flow is incompressible and the velocities gradients are in general low. Under these assumptions, equation 4.5 is simplified to
\[
\rho c_p \frac{DT}{Dt} = \nabla \cdot (k \nabla T) + \mu (\nabla^2 T) + \dot{q} \quad (4.6)
\]

or,
\[
\rho c_p \frac{\partial T}{\partial t} + \rho c_p \mathbf{V} \cdot (\nabla T) = \nabla \cdot (k \nabla T) + \mu (\nabla^2 T) + \dot{q} \quad (4.7)
\]

Furthermore, the term that includes the viscosity of the fluid, is negligible compared to the other terms of the equation and can be omitted. Finally, the thermal conductivity \( k \) is considered constant. Taking these into account, the final form of the energy equation is
\[
\rho c_p \frac{\partial T}{\partial t} = k (\nabla^2 T) + \dot{q} \quad (4.8)
\]

In the equation 4.8, on the left hand side, the first term is the time derivative and the second one is the convection term. On the right hand side, the first term expresses the diffusion and the final one is the heat source term.

### 4.3 Solid regions

#### 4.3.1 Energy equation

There are two different solid regions in the domain, i.e., the upper solid which consists of the driven electrodes, the dielectric tubes and the configuration of the suspender, and the lower solid which supports the grounded electrode. For these regions there is one and only equation that has to be solved. This is the energy equation. The energy equation for the solids is an reduced form of the energy equation presented for the fluid, as there are no fluxes through the solids. Therefore the energy equation for the solid regions is
\[
\rho c_p \frac{\partial T}{\partial t} = k (\nabla^2 T) + \dot{q} \quad (4.9)
\]

Dividing by the term \( \rho c_p \) which is assumed constant, the equation can be rewritten as
\[
\frac{\partial T}{\partial t} = \alpha (\nabla^2 T) + \frac{\dot{q}}{\rho c_p} \quad (4.10)
\]

where \( \alpha \) is the thermal diffusivity of the corresponding material.
4.4 Porous media

4.4.1 Momentum equation

In porous media, the equation that governs the fluid flow is the Darcy’s law. This law formulated by Henry Darcy, when he conducted experiments on the flow of water through beds of sand. Darcy’s law indicates that through a porous media the filtration velocity (or Darcy flux) is equal to the product of the pressure gradient and the permeability of the medium, divided by the dynamic viscosity of the fluid. This equation is shown below:

\[ u = -\frac{K}{\mu} \nabla p \]  

(4.11)

where \( u \) is the Darcy flux, \( K \) the permeability of the medium, \( \mu \) the dynamic viscosity of the fluid and \( \nabla p \) the pressure gradient.

In order to solve a problem combining fluid mechanics and porous media physics, one must combine the equations of these fields. For many years, scientists were trying to find the missing link between them. Today, this is feasible, and such problems are easier to be solved.

Generally speaking, there are two different approaches that one can solve a flow through porous media problem. These approaches are described briefly in the following sections. Let figure 4.2 to be an 2D cross-section of a porous medium. One can detect the voids that the fluid passes through (white color), and the solid region too (black color).

Figure 4.2: 2D cross-section of a porous medium

In the fluid region the continuity and the momentum equations are applied correctly. These equations are shown below:

\[ \nabla \cdot V_f = 0 \]

\[ \frac{\partial V_f}{\partial t} + (V_f \cdot \nabla)V_f = -\frac{1}{\rho_f} \nabla p_f + \nu_f \nabla^2 V_f \]  

(4.12)

where \( V_f \) is the velocity vector in the pore space, \( \nabla p_f \) the pressure drop, \( \rho_f \) and \( \nu_f \) the density and the kinematic viscosity of the fluid respectively.
4.4.2 Direct modeling approach

In this approach, the equations 4.12 has to be solved directly for the fluid region, imposing no-slip boundary conditions at the fluid-solid interface. Such a method gives results that are very close to the reality, but in many cases is frustrating. Whereas, the only inputs that must be imported are the geometry and the fluid properties, sometimes the first input is not feasible. Specific measuring techniques must be applied in order to export the full pore-scale geometry. Furthermore, this method can not be applied in large species, due to the fact that there is no computer potential to solve so many equations, for such a dense mesh.

4.4.3 Continuum modeling

The continuum modeling (or macroscale modeling) deals not with the pore-scale, but with larger domains of investigation. Control volumes are defined all over the domain, and quantities averaged over these volumes are considered. The control volume can not be arbitrary. It needs to be a Representative Elementary Volume (REV). In this approach, each point of the domain contains both solid and fluid in the averaged sense. Thus, the equations of fluid mechanics are no longer valid. The averaged quantities are governed by Darcy’s law.

\[
\nabla \cdot \langle V_f \rangle = 0
\]

\[
\langle V_f \rangle = -\frac{K}{\mu_f}(\nabla \langle p_f \rangle - \rho_f g)
\] (4.13)

The bracket notations are used to emphasize the difference between this method and the direct modeling approach. The quantity \( \mu_f \) is the dynamic viscosity of the fluid, while \( K \) is the permeability of the porous medium. Permeability is a measure of the ability of a porous material to transmit fluids.

Assuming known the equations above, and following a procedure called averaging one can conclude to the following equation, which is named Darcy -Brinkman- -Stokes equation

\[
0 = -\nabla \langle p_f \rangle + \rho_f g + \mu_f^* \nabla^2 \langle V_f \rangle - \mu_f K^{-1} \langle v_\beta \rangle
\] (4.14)

From a fluid dynamic point of view, the porous medium effect in the equations is a additional source terms, that represents the drag force due to the viscous friction of the fluid with the walls of the solid structure. Taking into account this assumption, the momentum equation in the porous zone can be written

\[
\frac{\partial V}{\partial t} + (V \cdot \nabla)V = -\frac{1}{\rho} \nabla p + \nu \nabla^2 V + \frac{1}{\rho} S_m
\] (4.15)
where $S_m$ is the aforementioned additional source term. Darcy’s law dictates that this term is equal to

$$S_m = -\frac{\mu}{K}V$$

(4.16)

However, when the velocity field is greater increasing the particle-diameter-based Reynolds number, this term must be corrected. The correction suggested by Forschheimer, and the new law, named Forschheimer-Extended Darcy’s law is

$$S_m = -\frac{\mu}{K}V - \frac{\rho F_r}{\sqrt{K}}V^2$$

(4.17)

In this chapter, the continuum modeling is used, with the equation 4.16 to represent the additional source term.

### 4.4.4 Energy equation

In porous media, there are in general two models that describe the heat transfer. The first one is the equilibrium model, which assumes that the fluid and the solid phase have the same temperature, and the non-equilibrium model, which obviously suggests that the two phases have different temperatures. In the present thesis, the model that will be used is the equilibrium thermal model. It is assumed that inside the sewage sludge, the fluid has numerous paths that can follow, so as the heat transfer between the hot fluid and the solid matrix is amplified, and the two phases have the same temperature. The same thermal model is taken into account for the grounded electrode. In this case the validity of the model can be justified by the fact that the fluid issuing from the sewage sludge follows complex flowlines exchanging large amounts of heat with the electrode.

**The equilibrium thermal model**

As it was mentioned, this model assumes that the two phases are in thermal equilibrium, so they have the same temperature for every moment. The energy equation that governs the two combined phases is

$$\left(\rho c_p\right)_m \frac{\partial T}{\partial t} + \left(\rho c_p\right)_f V \cdot \nabla T = k_m \nabla^2 T + \dot{q}$$

(4.18)

or

$$\left(\frac{\rho c_p}{\rho c_p}\right)_m \frac{\partial T}{\partial t} + \frac{V \cdot \nabla T}{\left(\rho c_p\right)_f} = \frac{k_m}{\left(\rho c_p\right)_f} \nabla^2 T + \frac{1}{\left(\rho c_p\right)_f} \dot{q}$$

(4.19)
where m, f and s refer to the medium, fluid phase and solid phase respectively. The source term \( \dot{q} \) represents every heat generation or heat absorption that takes place to the problem.

There is a wide range of expressions relating the thermophysical properties of the solid and the fluid phases with those of the medium. The simplest correlations for specific heat and thermal conductivity of the porous region are given by the rules of mixture

\[
(\rho C)_m = \epsilon(\rho C)_f + (1 - \epsilon)(\rho C)_s
\]
\[
k_m = \epsilon k_f + (1 - \epsilon)k_s
\]

where m, f and s refer to the medium, fluid phase and solid phase respectively. Here, \( \epsilon \) is the porosity of the porous medium.

### 4.5 Evaporation of water

The energy that is consumed in the procedure of the phase change, is computed in the following way. Having computed the mass difference of the sewage sludge before and after the treatment, the amount of water which is evaporated can be derived. After that, the required energy is computed, using this formula

\[
\Delta Q = \Delta m \cdot h_{fg}
\]

where \( \Delta m \) is the evaporated mass and \( h_{fg} \) the specific enthalpy of phase change.
The simulation of the present thesis was executed in OpenFOAM ("Open source Field Operation and Manipulation") environment. OpenFOAM is a C++ toolbox for the development of customized numerical solvers, and pre-/post-processing utilities for the solution of continuum mechanics problems, including Computational Fluid Dynamics (CFD). Having generated the computational mesh, and having presented the equations that govern the physical problem, in this section, the pre-processing and the development of a new solver are described.

5.1 Definition of regions

Although the mesh has been generated, the regions that it is divided into, have not been defined. This process is necessary in order to assign different equations for different parts of the computational domain. After the generation of the mesh, presented in chapter 3, each of the parts that took place in this procedure, was grouped into
Diploma Thesis

Rention S. Pasolari

cellZones. Thus, as a first step, these parts must be gathered together into larger cell zones. This can be done using the topoSetDict as was described in section 3.4. As one can see in the topoSetDict file illustrated in figure, there are three regions that the whole domain is separated into:

- The fluid region, that includes the porous regions.
- The upper solid region that comprises the driven electrodes, the dielectric tubes and the configuration of the suspender.
- The lower solid region that supports the grounded electrode.

The final step is to divide the defined cell zones. This can be done typing `splitMeshRegions -cellZones -overwrite`. At the same time, this command creates folders for each region, that every file concerning them must be included in these folders.

```plaintext
// Air is all the other region
{
    name airCellSet;
    type cellSet;
    action new;
    source zoneToCell;
    sourceInfo
    {
        name UpSolid;
    }`
}

actions
{
    // LowerSolid
    {
    name DownSolid;
    type cellZoneSet;
    action new;
    source setToCellZone;
    sourceInfo
    {
        set DownSolidCellSet;
    }
    }`
}

// UpperSolid
{
    name UpSolid;
    type cellZoneSet;
    action new;
    source setToCellZone;
    sourceInfo
    {
        set UpSolidCellSet;
    }
}

Figure 5.1: Regions definition-topoSet

Department of Mechanical Engineering and Aeronautics
After this process, the computational domain has three separate regions which interact with each other only by their boundary conditions. The aforementioned regions are depicted in the figure below.

![Figure 5.2: The different regions in paraView](image)

The blue region is the lower solid, the green one is the upper solid and finally the region that is surrounded by the red boundary is the fluid region.

### 5.2 Definition of different materials in a region

Despite the fact that the whole domain was divided, there is another need concerning each region. Different materials, so different properties, comprise each region. For instance, the dielectric tubes and the driven electrodes belong into the same region, yet there are made of quartz and inox steel respectively. Moreover, the fluid region includes also the porous zones, hence this must be defined. The definition of the different materials into one region can be done again using `topoSet` utility. Another big advantage of this approach is that it is feasible to solve the porous media and the fluid equations, simultaneously, using only one equation. Here is the topoSet dictionary for the definition of the sewage sludge (porous blockage) and the grounded electrode (grid) in the fluid region.
The following figure depicts the fluid region with the sewage sludge and the grounded electrode.

Figure 5.3: Porous zones definition-topoSet

![Diagram of porous zones definition-topoSet](image)

Figure 5.4: Fluid region, blue-fluid, red-sewage sludge, green-grounded electrode

The next figures illustrate the different materials in the upper and the lower solid.
5.3 Solving mathematical equations with OpenFOAM

In chapter 4, the equations that govern the entire problem were presented. Now, the way that these equations are inserted into OpenFOAM, is discussed. Generally speaking, OpenFOAM includes several solvers which can be applied in a great range of computational fluid dynamics problems. However, in some cases these solvers have to be edited, in order to solve specific cases. For the need of the present thesis, the pisoFoam solver was used. PisoFoam is an incompressible, transient solver which has the capability to solve both laminar and turbulent flows. Moreover, pisoFoam includes the fvOptions module, which is a powerful and very useful framework that allows user to select any physics that can be represented as sources or constraints on the governing equations, e.g. porous media and heat sources. Furthermore, it is worth noting, that pisoFoam, uses the PISO (Pressure-Implicit with Splitting of Operators) algorithm in order to correct the velocity and pressure fields for the satisfaction of the continuity equation. A detailed description of this algorithm is cited in the appendix C.

5.3.1 Fluid dynamics problem

After a few modifications to the pisoFoam, that concern the assumptions for the flow and the fluid which were discussed in chapter 4, the momentum equation in the OpenFOAM environment is
Diploma Thesis Rention S. Pasolari

Figure 5.6: Momentum equation in OpenFOAM

where \( \text{ddt}(U) \) is the time derivative of the velocity \( V \), \( \text{div}(\phi, U) \) the convection term, \( \text{MRF.DDt}(U) \) a term concerning problems with rotating systems, \( \text{laplacian}(\nu, U) \) the diffusion term and \( \text{fvOptions}(U) \) the fvOptions module that was mentioned above.

OpenFOAM divides its finite volume method into two main namespaces due to the fact that the solution method for an implicit equation differs significantly from the solution method for an explicit equation. The two namespaces are:

- fvm (finite volume method) for implicit equations.
- fvc (finite volume calculus) for explicit equations.

Implicit equations must be solved iteratively rather that producing an immediate solution (like the fvc namespaces). Hence, fvm namespace produces a fvMatrix object. This object can simply be solved using a great variety of methods. Since an explicit equation can be solved immediately, the fvc namespace does so. In other words, given an operation on a volume field, the fvc namespace produces another volume field. A number of different discretization schemes are available for each operation.

It should be noted, that every field that is used in the equations, must firstly be defined in a `createFields` file.

Since pisoFoam supports the fvOptions module, the porous zones effect the fluid dynamic problem will be inserted in this way, so an fvOptions dictionary must be included in the `constant` folder of the case. The user has to define the porous regions by importing the corresponding cell zones, as well as some properties of the porous media, that will be discussed in chapter 6. For the specific problem the fvOptions file is the following:

```plaintext
fvVectorMatrix UEqn
{
    fvm::ddt(U) + fvm::div(phi, U)
    + MRF.DDt(U)
    -fvm::laplacian(v,U)
    == fvOptions(U);
UEqn.relax();
fvOptions.constrain(UEqn);
if (piso.momentumPredictor())
{
    solve(UEqn == -fvc::grad(p));
    fvOptions.correct(U);
}
```

42 Department of Mechanical Engineering and Aeronautics
As one can see in the figure 5.7, this is an explicit way of adding porous media in OpenFOAM.

5.3.2 Energy equation

A detailed description of the energy equation that governs the specific problem, and the way that are treated in order to approach the real solution was presented in sections 4.2.3, 4.3.1 and 4.4.4.

Solid regions

The general equation that is solved for the solid regions is:
where \( \text{ddt}(T_{\text{UpSolid}}) \), \( \text{ddt}(T_{\text{DownSolid}}) \) are the time derivatives of the temperature field, \( \text{laplacian}(\alpha \, T_{\text{UpSolid}}, T_{\text{UpSolid}}) \) and \( \text{laplacian}(\alpha \, T_{\text{DownSolid}}, T_{\text{DownSolid}}) \) the diffusion terms and \( q_{\text{UpSolid}} \) represents the source term for the upper solid region.

### Fluid and porous regions

The new solver that was generated in the present thesis, gives the chance to the user to solve in one equation, the energy equations for both the fluid and the porous media. The code that was developed is shown below.

![Energy equation for the fluid region](image)

\[
\text{fvScalarMatrix} \, \text{TEqn} \\
\text{fvm::ddt}(\rho_{\text{Cp}}/\rho_{\text{CpFluid}}, T) + \text{fvm::div}(\phi, T) - \text{fvm::laplacian}((kappa/\rho_{\text{CpFluid}}), T) == (q*Qf/(1.8786*1e-06*(\rho_{\text{CpFluid}}/(rhouInt*cpInt))))
\]

\text{TEqn.relax();}

where \( \text{ddt()} \) represents the time derivative term, \( \text{div()} \) is the advection term, \( \text{laplacian()} \) is the diffusion term and the final term represents the heat source. The Fluid index, differentiates the properties that refer to the fluid from the properties of the porous zone. In order to solve the equation for both regions simultaneously, different values of the properties must be defined at each region. For that purpose, the setFields utility is used. The following figure shows the way that the different properties can be defined for specific zones of the fluid region. This form of the equation, along with the file presented in figure 5.10, make the simultaneous solution of the equations feasible. Specifically, figure 5.11, shows the way that air properties are computed according to the temperature, as well as the definition of the mean properties of the porous zones. It is important to mention that in the same way, the heat generation term is defined in the appropriate zones. Figure 5.12, illustrates the definition of the power that is imposed in the reactor.
```c
defaultFieldValues
{
  volScalarFieldValue epsilon 1
  volScalarFieldValue kappaM 15
  volScalarFieldValue rhoCpM 3.9e06
  volScalarFieldValue Qf 0
};
regions
{
  zoneToCell
  {
    name grid;
    fieldValues
    {
      volScalarFieldValue epsilon 0.27
      volScalarFieldValue kappaM 15
      volScalarFieldValue rhoCpM 3.9e06
      volScalarFieldValue Qf 0
    };
  }
  zoneToCell
  {
    name alrayer;
    fieldValues
    {
      volScalarFieldValue Qf 1
    };
  }
  zoneToCell
  {
    name porousBlockage;
    fieldValues
    {
      volScalarFieldValue epsilon 0.4
      volScalarFieldValue kappaM 0.364
      volScalarFieldValue rhoCpM 4.392e06
      volScalarFieldValue Qf 0
    };
  }
}
```

**Figure 5.10:** setFields

```c
cpFluid = (((0.165400e-0.264807e-0.01)*T/UnitT*0.701681e-0.9*UnitT*0.497676e-0.9*UnitT*0.317792e-0.9*UnitT*0.217792e-0.9*UnitT*0.169792e-0.9*UnitT)/UnitT);
kappaFluid = (((-0.227655e-0.312596e-0.01)*T/UnitT*0.1815233e-0.9*UnitT*0.1235564e-0.9*UnitT*0.1235564e-0.9*UnitT*0.1235564e-0.9*UnitT)/UnitT);
mu = (((-0.8681e-0.1e+0.898125e-0.5*UnitT*0.1736557e-0.5*UnitT*0.791209e-0.5*UnitT)/UnitT*0.1736557e-0.5*UnitT*0.791209e-0.5*UnitT))/UnitT;
rhoFluid = (((p/UnitT)*101325)/(287.058*UnitT))*
```

**Figure 5.11:** Variable properties

```c
if (runTime.time().value()<=5000)
{
  qa=0.88653+3.34079*(runTime.time().value()/(60))-0.00946*pow((runTime.time().value()/(60)),2)
  +0.99280*pow((runTime.time().value()/(60)),3)-0.00421*pow((runTime.time().value()/(60)),4)
  +7.17e-05*pow((runTime.time().value()/(60)),5))*10^0.156;
}
else if (qa=0.88653+3.34079*(runTime.time().value()/(60))-0.00946*pow((runTime.time().value()/(60)),2)
```

**Figure 5.12:** Heat source definition
5.4 Boundary conditions

Having presented the equations that govern the problem and the way that these are inserted into OpenFOAM, the implementation of the boundary conditions follows.

Fluid

For the fluid region the user must include boundary conditions for the following three fields:

1. Velocity $V$.
2. Pressure $p$.
3. Temperature $T$.

For the velocity field the boundary conditions that are imposed are:

<table>
<thead>
<tr>
<th>$1^{st}$ region</th>
<th>$2^{nd}$ region</th>
<th>Boundary condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluid $\rightarrow$</td>
<td>Outer walls</td>
<td>noSlip</td>
</tr>
<tr>
<td>Fluid $\rightarrow$</td>
<td>Inlet</td>
<td>fixedValue</td>
</tr>
<tr>
<td>Fluid $\rightarrow$</td>
<td>Outlet</td>
<td>zeroGradient</td>
</tr>
<tr>
<td>Fluid $\rightarrow$</td>
<td>Solid regions</td>
<td>noSlip</td>
</tr>
</tbody>
</table>

The value that is imported in the fixedValue boundary condition is presented in the next chapter, where all the parameters of the problem are defined.

For the pressure field the boundary conditions are:

<table>
<thead>
<tr>
<th>$1^{st}$ region</th>
<th>$2^{nd}$ region</th>
<th>Boundary condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluid $\rightarrow$</td>
<td>Outer walls</td>
<td>zeroGradient</td>
</tr>
<tr>
<td>Fluid $\rightarrow$</td>
<td>Inlet</td>
<td>zeroGradient</td>
</tr>
<tr>
<td>Fluid $\rightarrow$</td>
<td>Outlet</td>
<td>fixedValue</td>
</tr>
<tr>
<td>Fluid $\rightarrow$</td>
<td>Solid regions</td>
<td>zeroGradient</td>
</tr>
</tbody>
</table>
For the temperature field:

**Table 5.3: Temperature boundary conditions-Fluid**

<table>
<thead>
<tr>
<th>1&lt;sup&gt;st&lt;/sup&gt; region</th>
<th>2&lt;sup&gt;nd&lt;/sup&gt; region</th>
<th>Boundary condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fluid → Outer walls</td>
<td></td>
<td>zeroGradient</td>
</tr>
<tr>
<td>Fluid → Inlet</td>
<td></td>
<td>fixedValue</td>
</tr>
<tr>
<td>Fluid → Outlet</td>
<td></td>
<td>zeroGradient</td>
</tr>
<tr>
<td>Fluid → Solid regions</td>
<td></td>
<td>compressible::turbulentTemperatureCoupledBaffleMixed</td>
</tr>
</tbody>
</table>

The *compressible::turbulentTemperatureCoupledBaffleMixed* boundary condition indicates the heat transfer between the fluid and the solid region, taking into account the thermal conductivity.

**Solids**

The solid regions of the domain contact only the fluid region and the outer walls. The only field that should be solved is the energy equation in terms of temperature. The same boundary conditions are imposed for both the upper and the lower solid region. These boundary conditions shown in the table 5.4.

**Table 5.4: Temperature boundary conditions-Solid**

<table>
<thead>
<tr>
<th>1&lt;sup&gt;st&lt;/sup&gt; region</th>
<th>2&lt;sup&gt;nd&lt;/sup&gt; region</th>
<th>Boundary condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid → Fluid</td>
<td></td>
<td>compressible::turbulentTemperatureCoupledBaffleMixed</td>
</tr>
<tr>
<td>Solid → Outer walls</td>
<td></td>
<td>zeroGradient</td>
</tr>
</tbody>
</table>

**5.5 Solution**

Having defined the boundary conditions, the final steps are to define the discretization of the governing equations, the solving method of the resultant system and some parameters essential for the creation of the database. In the next subsections these topics are presented:
5.5.1 Discretization

OpenFOAM uses the finite volume method for the discretization of the equations (more information in appendix C). The following figures illustrate the discretization method used for each region.

(a) fvSchemes for fluid

(b) fvSchemes for solid

Figure 5.13: fvSchemes files

5.5.2 Algorithm control

After the discretization process, an algebraic system arises. There are several methods for solving these systems. The methods that were used in the present problem are shown below:
Having completed the above steps, the final one is to set the time step of the solution, as well as some parameters for the creation of the database. This information is included in the controDict file of the case, which is illustrated in figure 5.15. The time step can not be arbitrary. Specifically, it must comply with the Courant’s criterion. This criterion implies there is a relationship between the time step, the spatial step and the velocity in the computational domain, and it is given below:

\[
\frac{u \Delta t}{\Delta x} \leq 1 \implies Co \leq 1
\]  

(5.1)
It is important to mention that the simulation, and the snappyHexMesh utility, run in parallel in 72 cores, and it was completed about 24 hours after the start time.

**Figure 5.15:** controlDict file
Chapter 6

Problem parameters and results

Having presented the problem, the equations that govern it and all the procedures that where followed in OpenFOAM for constructing the mesh and pre-processing the problem, this final chapter deals with the results of the simulation. First of all, the parameters of the entire problem, e.g., material properties, air properties, assessment of the assumptions considering the flow and the heat transfer etc., are presented. These are followed by the results of the simulation for the pressure, the velocity and the temperature field. It is very important to mention that there is a comparison with the experimental data. Moreover, the conclusions that derive from these results will be discussed. Finally, future aims and improvements of the model are presented.

6.1 Problem parameters

Although the problem has been presented analytically, there has been no mention to the numerical parameters that are used. This will be done in this section.
6.1.1 Material properties

First of all, the properties of the materials, the porous media and the air are specified.

Solid properties

There are three different materials which comprise the solid parts of the reactor. These materials are:

- **Quartz** for the dielectric tubes.
- **Macor** for some parts of the lower solid, showed in figure 5.5.
- **Inox steel**, which is the material of the rest solids of the reactor.

It is assumed that the properties of the solids are independant of the temperature. This assumption is generally acceptable for temperature differences of order 100K. In the table below, the properties of the materials are defined.

<table>
<thead>
<tr>
<th>Material</th>
<th>$k \left( \frac{W}{m\cdot K} \right)$</th>
<th>$c_p \left( \frac{J}{kg\cdot K} \right)$</th>
<th>$\rho \left( \frac{kg}{m^3} \right)$</th>
<th>$\alpha \left( \frac{m^2}{s} \right)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quartz</td>
<td>12</td>
<td>660</td>
<td>2200</td>
<td>8.2640 · 10^{-6}</td>
</tr>
<tr>
<td>Macor</td>
<td>1.46</td>
<td>774</td>
<td>2520</td>
<td>7.4853 · 10^{-7}</td>
</tr>
<tr>
<td>Inox</td>
<td>15</td>
<td>500</td>
<td>7800</td>
<td>3.8460 · 10^{-6}</td>
</tr>
</tbody>
</table>

Air properties

The properties of dry air can not be assumed as invariant of the temperature. There is a correlation between its properties and the temperature. The functions that are used in order to compute the air properties each time step are the following.

- **Density**

  The air is considered as an ideal gas, hence the density is calculated from the ideal gases equation

  $$\rho = \frac{p}{RT}$$

  where $\rho$ is the density, $p$ the pressure, $T$ the temperature and $R$ the specific gas constant for air that is 287.058$\frac{J}{kg\cdot K}$.
• **Thermal conductivity**

The thermal conductivity of air is approximated as:

\[
k = -2.276501 \cdot 10^{-3} + 1.259845 \cdot 10^{-4}T - 1.4815235 \cdot 10^{-7}T^2 + 1.73550646 \cdot 10^{-10}T^3 - 1.066657 \cdot 10^{-13}T^4 + 2.47663035 \cdot 10^{-17}T^5
\]

for \(250K < T < 1050K\) in \((W/k\cdot\text{K})\).

• **Specific heat capacity**

The specific heat capacity of air is approximated as:

\[
c_p = 0.103409 \cdot 10^1 - 0.2848870 \cdot 10^{-3}T + 0.7816818 \cdot 10^{-6}T^2 - 0.4970786 \cdot 10^{-9}T^3 + 0.1077024 \cdot 10^{-12}T^{-4}
\]

with dimensions \((kJ/kg\cdot\text{K})\).

• **Dynamic Viscosity**

The dynamic viscosity of air is approximated as:

\[
\mu = -9.8601 \cdot 10^{-2} + 9.08125 \cdot 10^{-2}T - 1.17635575 \cdot 10^{-4}T^2 + 1.234903 \cdot 10^{-7}T^3 - 5.7971299 \cdot 10^{-11}T^4
\]

in \((Ns/m^2 \cdot 10^6)\).

**Porous media**

In sections 4.4.1 and 4.4.4, the model used for the porous regions was discussed. In those sections, was mentioned that the properties in porous zones are the mean values between the air and the solid part of the material, and they depend on the porosity. As it was described above, the properties of the air are not constant, hence the properties of the porous media are not constant too.

The porosity of the grounded electrode was computed as follows

\[
\epsilon = \frac{\text{volume of pores}}{\text{total volume}} \Rightarrow \epsilon = \frac{840\pi d^2}{4\pi D^2} \Rightarrow \epsilon = 840 \left(\frac{d}{D}\right)^2 = 840 \left(\frac{0.00105}{0.059}\right)^2 \Rightarrow
\]

\[
\Rightarrow \epsilon = 0.267
\]
The porosity of the sewage sludge is specified as 0.4. This is a mean value of the initial and the final porosity of the sewage sludge i.e. before and after the evaporation of the water.

Equations 4.20 represent the formulas that give the mean values of the properties for the porous zones. These values are functions of the properties of air, of the solid matrix and the porosity of the medium. The solid matrices have the following properties (including the porosity and the permeability coefficient $K$)

<table>
<thead>
<tr>
<th>Table 6.2: Solid matrices properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Material</strong></td>
</tr>
<tr>
<td>Sewage sludge solid matrix</td>
</tr>
<tr>
<td>Grounded electrode solid matrix (Inox)</td>
</tr>
</tbody>
</table>

### 6.1.2 Power

The experiments showed that the overall power that was provided to the driven electrodes was not constant. In other words, while the sewage sludge has a large amount of water the power is increasing and reaches a peak. After that, while the evaporation of water takes place, the power is sharply decreased and reaches a very low value for the last minutes of the experiment. In the table 6.4, the power for different times is presented.

<table>
<thead>
<tr>
<th>Table 6.3: Power</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time (min)</strong></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

Using a polynomial interpolation (power fitting), the following function determines the power through time.
where $t$ is expressed in minutes.

The next figure illustrates the power distribution through time.

![Experimental and fitting of the total power](image)

**Figure 6.1:** Power fitting of the total power given to the reactor

However, as it was described in section 4.1, there is a large percentage of this power that is consumed by the evaporation of the water. The experiments have shown that the initial amount of water was 78% water mass to total mass. During the decontamination procedure, water is evaporated from the surface of the sewage sludge. After 20 minutes of the experiment, the water percentage has been reduced to 30%. It is reasonable to assume that a proportion of the provided energy is consumed to this process. The temperature of the water while it is evaporated is not constant. However, the enthalpy of phase change does not vary significantly in the temperature range between $25^\circ C$ and $100^\circ C$. Specifically, a value of $2500 \frac{kJ}{kg}$ is acceptable.

Having calculated the initial and the final mass of the sewage sludge, their difference is the mass of water that was evaporated. This difference is $\Delta m = 4.785 \cdot 10^{-3} kg$. Thus the energy that is required for this, is

$$\Delta Q = \Delta m \cdot h_{fg} = 4.785 \cdot 10^{-3} \cdot 2500 \cdot 10^3 J \Rightarrow \Delta Q = 11962 J$$

As one can see in figure 6.1, the overall power has been reduced a lot, within the first 10 minutes of the experiment. Hence, it is reasonable to assume that the evaporation occurs during this time period. Taking what has been mentioned here and in section 4.1 into account, the proportions of the total energy that is distributed as follows.
Table 6.4: Energy distribution

<table>
<thead>
<tr>
<th>Time period (min)</th>
<th>Zone</th>
<th>Percentage of total energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-10</td>
<td>Dielectric tubes</td>
<td>18 %</td>
</tr>
<tr>
<td>0-10</td>
<td>Thin air layer</td>
<td>17 %</td>
</tr>
<tr>
<td>10-20</td>
<td>Dielectric tubes</td>
<td>18 %</td>
</tr>
<tr>
<td>10-20</td>
<td>Thin air layer</td>
<td>15.8 %</td>
</tr>
</tbody>
</table>

6.1.3 Assumptions

It has already been mentioned, that some assumptions have been made in the specific model. Here, these assumptions are reported again, and they are justified via experimental and computational results.

- The flow is considered laminar.
- The fluid is incompressible.
- The reactor is insulated.
- The heat loss by chemical reactions is not considered.
- The radiation effects can be neglected.

Incompressibility

The Mach number is given by the following expression

\[ Ma = \frac{u}{c} \]  \hspace{1cm} (6.2)

where \( u \) is the velocity of the air and \( c \) is the sound velocity in it. Sound velocity \( c \) is a function of the gas and the temperature, and specifically \( c = \sqrt{\frac{\gamma RT}{M}} \), where \( \gamma = \frac{c_p}{c_v} \), \( R \) the universal gas constant, \( T \) the temperature \( M \) the molar mass of the specific gas. Taking into consideration the maximum velocity and the maximum temperature that are developed in the flow (depicted in section 6.2.2), the maximum Mach number is

\[ Ma = \frac{0.94}{\sqrt{\frac{1.4 \cdot 8314 \cdot 400}{28.97}}} \Rightarrow Ma = 2.36 \cdot 10^{-3} < 0.3 \]

Thus, the assumption of the incompressibility is valid.
Energy consumed to chemical reactions and exchange with the environment

Experimental observations have proven that the heat loss through the reactor’s boundary walls is negligible.

The power that is consumed for the chemical reactions which take place in the sewage sludge is not easy to be computed, but it is certain that it is very low compared to the total imposed energy, thus it can be neglected.

Radiation

The assumption that the radiation is negligible can be justified by investigating the most extreme occasion, where the dielectric tubes have increased their temperature to 400K while the sewage sludge is at 300K. Moreover, the dielectric tubes’ emissivity is considered 0.9. The radiation exchange between dielectric tubes and sewage sludge is approximated as radiation exchange between parallel circular plates, as it is shown if figure 6.2.

Figure 6.2: Parallel circular plates

The view factor between two parallel circular plates is:

\[ F_{1 \rightarrow 2} = \frac{1}{2} \left[ X - \sqrt{X^2 - 4 \left( \frac{R_2}{R_1} \right)^2} \right] \]

where \( R_1 = \frac{r_1}{h}, \) \( R_2 = \frac{r_2}{h} \) and \( X = 1 + \frac{1 + R_2^2}{R_1^2}. \) Here \( r_1 = r_2 = 0.025mm, \) and \( h = 0.002mm. \)

Thus, \( F_{1 \rightarrow 2} = 0.9232 \)

So, in the most extreme occasion, the heat transferred by radiation from the dielectric tubes to the sewage sludge is:

\[ \dot{Q} = F_{1 \rightarrow 2} \sigma A \epsilon_\alpha (T_1^4 - T_2^4) = 0.9232 \cdot 5.67 \cdot 10^{-8} \cdot \pi 0.025^2 \cdot 0.9 \cdot (400^4 - 300^4) \Rightarrow \]

\[ \Rightarrow \dot{Q} = 1.617W \]
In the previous equation $\sigma$ is the Stefan-Boltzmann constant, $A$ the area of each plate, $\epsilon_\alpha$ the emissivity of the dielectric surface, $T_1$ the temperature of the dielectric surface and $T_2$ the temperature of the sewage sludge.

6.2 Results

This section deals with the results of the simulation, as well as the conclusions that arise from each result. The results that are illustrated concern the pressure, velocity and temperature fields.

6.2.1 Pressure field

The following figure shows the pressure field at the y-normal cross-sectional plane.

Figure 6.3: Pressure field at y-normal cross-sectional plane
Conclusions

It is obvious that the porous media lead to a pressure drop, due to the Darcy term that was included in the Navier-Stokes equation. The pressure drop is proportional to the permeability coefficient. This can be observed from figure 6.4 which depicts the distribution of pressure on the z-axis, from the plasma vicinity to the end of the grounded electrode. One can see that there is a large pressure drop at the place that the sewage sludge is located, and a negligible one at the grounded electrode, which has much lower permeability coefficient.

6.2.2 Velocity field

Figures 6.5a and 6.5b, illustrate the velocity field at the x-normal and y-normal cross-sectional plane respectively, is illustrated. Moreover, figure 6.6 shows the velocity field in a slice at the porous zone in z-normal cross-sectional plane. Finally, in figure 6.7, one can see the velocity field, at the area where the plasma generation takes place.

Conclusions

The velocity field is the expected. The air follows the only possible path for escaping from the reactor. The velocity magnitude seems to increase when air passes from the space between the driven electrode and the sewage sludge due to the narrowing of the geometry. In addition, in figure 6.7 one can see that the gap between the dielectric tubes
does not allow the flow of the air between them. This is an important point, due to the fact that if a less dense grid is needed, this geometric characteristic can be neglected. Finally, it is concluded that the maximum magnitude of the velocity is at the outlet, where velocity has a magnitude $U = 0.8102 \frac{m}{s}$.
6.2.3 Temperature field

The temperature field is the most important part of the specific simulation. Figure 6.8 represents snapshots of the temperature profile in the middle cross-section of the fluid region at several time instances.

Figure 6.7: Velocity field in the plasma and porous zones vicinity

Figure 6.8: Temperature distribution for different moments
The next figures illustrate the maximum temperature in the reactor, in plasma vicinity.

**Figure 6.9:** The temperature distribution in the plasma region at (a) x-normal and (b) y-normal cross-sectional planes

Now, the maximum temperature of the upper and the lower solid are presented

**Figure 6.10:** The temperature distribution in the solid regions
Here, the temperature when the simulation stops (20 minutes) is illustrated

![Temperature field at 20 minutes](image)

**Figure 6.11:** Temperature field at 20 minutes

In the next figure, one can see the temperature curve, from the plasma vicinity to the outlet.

![Temperature curve](image)

**Figure 6.12:** Temperature curve
Conclusions

The temperature distribution in the reactor shows that the convection terms dominate the flow. Moreover, near the configuration of the suspender of the driven electrodes, one can observe that the thermal energy is diffused.

The maximum temperature is reached in about the 6th minute of the simulation, in the thin air layer above the sewage sludge. This temperature is 397K. After this moment, the temperature is reduced, due to the fact that the input power is decreased.

The maximum temperatures in the upper and the lower solid are 361K and 325K, respectively.

6.3 Comparison with the experimental results

The results that were presented above are validated via comparison with experimental data. The comparison was carried out by measuring the development of the temperature for two specific points of the reactor. These points are:

- Grounded electrode.
- Driven electrode vicinity.

The specific place of the points that the temperature was measured are shown in the following figure.

![Figure 6.13: The points that the temperature was measured.](image-url)
6.4 Conclusions

The numerical results seem to fit well enough to the corresponding experimental results. Specifically, the temperatures that were computed through the simulation are a little larger than the experimental. It can be assumed that this little difference is owing to the assumptions with regard to the the omission of the energy consumption for chemical reactions. Moreover, the evaporation model that token into account may need to be closer to the real mechanisms. However, the results satisfy the aims of the specific thesis. The maximum temperature in the reactor is about $400K$ or $130^\circ C$, which seems to be appropriate for the decontamination of the sewage sludge.
6.5 Future work

The simulation that was held and that was presented above, has potential to be more completed. A possible improvement, is the modeling of the evaporation of the water. The exact way that this phenomenon takes place, will give more exact results. Furthermore, some chemical reactions can be included in the model, in order to include in the model the way that plasma acts for the treatment of sewage sludge. Moreover, it is very important to investigate the exact effect of plasma to the fluid dynamic flow. This can be done by adding the equations that govern plasma flows. The resolution of plasma kinetics however, would require an extremely dense mesh and a very small time step, which render the simulation very demanding in terms of the computational power.
In this appendix, the designs of the outer and inner characteristics of the reactor are presented. Specifically, one can see below, the geometry of the exterior of the reactor, the driven electrodes with the dielectric tubes and their suspender, as well as the grounded electrode and the different parts that comprise its support.

A.1 Reactor
Figure A.1: The outer geometry of the reactor
A.2 Upper solid configuration

Figure A.2: Regions
A.3 Lower solid configuration

(a) Stages of assembly of the lower solid

(b) Inox base

(c) Macor 1

Figure A.3: Regions
Figure A.3 (continued): Triangulated surfaces illustrated through *paraView*. 
Finite volumes method

There are three available methods of discretization in computational fluid dynamics. These methods are:

1. Finite differences method (FDM).
2. Finite volumes method (FVM).
3. Finite elements method (FEM).

The method that is the most common and appropriate for solving CFD problems is the finite volume method. The method can be applied in complex geometries, and in contrast to the finite differences, it can be applied in unstructured grids.

The computational domain is divides into a number of control volumes (or cells) where all the computations take place. Generally, the variables are calculated in the center of the cell and the fluxes on the faces. However in specific cases, some variables and the fluxes may be computed in different location. The computational procedure that was described above concerns discretization with cell-centered collocated grid. There is a variety of control volumes that differ to their shape. The most common control volume is
the cuboid (hexahedron). Nevertheless, a complex geometry is not feasible to be consisted purely of hexes.

The basic idea of the method is that the equations that must be written in a conservative form, are integrated within the control volume, in opposition to finite differences method, where the equation terms are differentiated between the nodes. This makes FVM much more complex that FDM, but also the most suitable method for CFD.

### B.1 Grid topology

Finite volumes method can solve one, two and three dimension problems. In the general situation, the 3D cuboid control volume is shown below

![Figure B.1: Three dimensional control volume](image)

The control volume has a center where the variables are computed, and six faces where the fluxes are calculated. The central node is defined by the letter P (polar), the upper node is N (north), S is the bottom node (south), E the right node (east), W the left node (west) and finally T and B are the front and the back nodes respectively. It is observed that nodes are represented with capital letters. For each capital letter there is an small letter which defines the corresponding face of the cell.
B.2 Discretization

Depending of the dimensions of the problem that must be solved, the discretization process differs. Obviously the 3D problems are the most complex. However, the philosophy behind each discretization procedure is the same. For example, the discretization of the diffusion and the advection term of the Navier-Stokes equation in one dimension is presented.

Let $\phi$ be one of the velocities $u$, $v$ or $w$. The Navier-Stokes equation in one dimension (assuming $x$) is

$$\frac{\partial \phi}{\partial t} + \frac{\partial (u\phi)}{\partial x} = -\frac{\partial (p/\rho)}{\partial x} + \frac{\partial}{\partial x} \left( \nu \frac{\partial \phi}{\partial x} \right)$$

This equation must be discretized in the following 1D control volume

![Figure B.2: One dimensional control volume](image)

**Diffusion term**

As it has been mentioned, the terms of the equations must be integrated within the control volume. Here, the integration takes places from the face $w$ to the face $e$. So the mathematical expression is

$$D = \int_{w}^{e} \frac{\partial}{\partial x} \left( \nu \frac{\partial \phi}{\partial x} \right) dx$$

The quantity inside the integral can be integrated directly since it is written in a conservative form. The integration gives

$$D = \left( \nu \frac{\partial \phi}{\partial x} \right)_{e} - \left( \nu \frac{\partial \phi}{\partial x} \right)_{w}$$

The quantity $\phi$ is computed at the nodes, thus the above terms are expanded to

$$\left( \frac{\partial \phi}{\partial x} \right)_{e} = \frac{\phi_{E} - \phi_{P}}{\delta x_{e}} , \quad \left( \frac{\partial \phi}{\partial x} \right)_{w} = \frac{\phi_{P} - \phi_{W}}{\delta x_{w}}$$
Substituting these quantities into the Equation B.3, the discretized diffusion term will be

\[ D = \alpha_D^W \phi_W + \alpha_D^P \phi_P + \alpha_D^E \phi_E \]  

where

\[ \alpha_D^W = \frac{\nu}{\delta x_w}, \quad \alpha_D^P = -\nu \left( \frac{1}{\delta x_e} + \frac{1}{\delta x_w} \right), \quad \alpha_D^E = \frac{\nu}{\delta x_e} \]

### Advection term

The advection term is more complex to be discretized owing to its non-linearity. There are several techniques of overcoming this problem. The most common is the lagging technique, which suggests that the flux term can be approximated by the velocity at the previous time step. For instance, in an implicit method of solving partial differential equations the lagging technique is

\[ F = \frac{\partial (u \phi)}{\partial x} = \frac{\partial (u^n \phi^{n+1})}{\partial x} \]  

The finite volume method, without taking into account the time step, suggests that the advection terms is integrated inside the control volume as shown below

\[ F = \int_{w}^{e} \frac{\partial (u \phi)}{\partial x} \, dx = (u \phi)_e - (u \phi)_w \]

The variables again must be computed at the nodes. For this reason, a wide range of interpolation schemes is used in order to calculate them in the faces. Some of this schemes is the upwind, the central difference and the QUICK. Here the upwind difference scheme (UDS) is presented.

UDS suggests that the value of the variable at a face, is equal to the value of the variable in the previous or the following node, depending on the sign of the flux. Specifically, the UDS expression is:

- If \( u_e > 0 \) and \( u_w > 0 \) then
  \[ \phi_e = \phi_P + O(\Delta x) \]
  \[ \phi_w = \phi_W + O(\Delta x) \]  
  In this case the advection term will become
  \[ F = u_e \phi_P - u_w \phi_W \]  

- If \( u_e < 0 \) and \( u_w < 0 \) then
  \[ \phi_e = \phi_E + O(\Delta x) \]
  \[ \phi_w = \phi_P + O(\Delta x) \]
In this case the advection term will become

\[ F = u_e \phi_E - u_w \phi_P \]  \hspace{1cm} (B.10)

Same techniques are used for 2D and 3D discretization. It must be noted that in these situations, the Green’s theorem and the Gauss’s theorem must be used for 2D and 3D cells respectively, since surface and volume integrals must be computed.
When the Navier-Stokes equations are solved, the obtained solution does not satisfy the continuity equation. This can be proven by observing the divergence error, which is not equal to zero. Thus, there is a need of a method that will lead this error to zero.

One of the most important algorithms used for this purpose is PISO (Pressure Implicit with Splitting of Operators) algorithm, which was introduced by Issa in 1985. The main advantage of this algorithm is that can be applied in collocated grids, in contrast to the SIMPLE family of algorithms which need a staggered grid.

The concept of PISO is that a Poisson equation is used to calculate the pressure, in order to obtain a second guess of the velocity that leads to a convergence error closer to zero. The procedure is repeated enough times, until the corrected velocity and pressure fields, satisfy both Navier-Stokes and continuity equations.

Now the steps that PISO algorithm follows are presented. In addition, a flowchart can be found in figure C.1:

The Navier-Stokes set of equations for a 3D problem, is given below.
\[
\frac{\partial V}{\partial t} + (V \cdot \nabla)V = -\frac{1}{\rho} \nabla p + \nu \nabla^2 V + \frac{1}{\rho} F
\]  

(C.1)

1. Solving the equation above, a first estimation of the velocity field is obtained. This estimation is expressed using an asterisk \( V^* \).

2. Using the velocity field \( V^* \), the flux term is defined as:

\[
\nabla \cdot \left[ H^* \frac{A}{A} \right] = \text{(flux term)}
\]  

(C.2)

where \( H^* \) is a quantity which depends on the velocity field \( V^* \).

The pressure \( p^* \) is estimated from a Poisson equation, which is derived from the continuity equation.

\[
\nabla \cdot \left[ \frac{1}{A} \nabla \cdot p^* \right] = \nabla \cdot \left[ \frac{H^*}{A} \right]
\]  

(C.3)

3. Substituting the calculated pressure \( p^* \) into the Navier-Stokes, a better estimation of the velocity is obtained \( V^{**} \).

4. If the new estimation of the velocity satisfies the continuity equation

\[
\nabla \cdot V^{**} = 0
\]  

(C.4)

then \( V^{n+1} = V^{**} \) and \( p^{n+1} = p^* \) and the algorithm is completed for the current time step.

If not, then after defining \( V^* = V^{**} \) and \( p = p^* \), steps 2-4 are repeated.

5. The next time step starts.
Navier-Stokes

\[ \frac{\partial V}{\partial t} + (V \cdot \nabla)V = -\nabla p + \frac{1}{Re} \nabla^2 V \]

\[ V^* \]

\[ H^*, A \]

Poisson Equation

\[ \frac{1}{A} \int_\Sigma \nabla^2 p^* \, dS = \nabla \cdot \left[ \frac{H^*}{A} \right] \]

\[ V^* = V^{**}, \ p = p^* \]

\[ p^* \]

\[ V^{**} = \frac{H^*}{A} - \frac{1}{A} \int_\Sigma \nabla p^* \, dS \]

\[ V^{**} \]

no

\[ \nabla \cdot V^{**} = 0 \]

yes

\[ V^{n+1} = V^{**}, \ p^{n+1} = p^* \]

**Figure C.1:** The PISO algorithm flowchart
Bibliography


