Inclusion of Interactions in Mathematical Modelling of Implant Assisted Magnetic Drug Targeting

Adil Mardinoglu

PhD 2009
Inclusion of Interactions in Mathematical Modelling of Implant Assisted Magnetic Drug Targeting

Adil Mardinoglu

PhD

School of Engineering
Waterford Institute of Technology

Supervisors: Dr. P J Cregg
Dr. Kieran Murphy

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy in Engineering is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: ___________________________  ID No: 20010510

Date: 30 November, 2009
Acknowledgements

I wish to express my gratitude to my supervisors, Dr. P J Cregg and Dr. Kieran Murphy, for the incredible amount they have taught me during the course of this research. It has been an exceedingly rewarding experience.

I want to thank Enterprise Ireland for support for this work under the Applied Research Enhancement (ARE) programme as part of the South Eastern Applied Materials (SEAM) Research Centre at Waterford Institute of Technology. I am also grateful to Cancer Research Ireland (Irish Cancer Society) for an Oncology Scholars Travel Award to attend the 6th International Scientific and Clinical Applications of Magnetic Carriers, May 17th – 20th, 2006, Krems, Austria.

I would like to thank Dr. Venie Martin and the other members of International Office for their support and to the Erasmus Programme for facilitating the completion my Istanbul Technical University Bachelor of Engineering degree studies at WIT in 2004-2005.

Finally, I would like to thank Michael Brennan and the members of the South Eastern Applied Materials (SEAM) Centre, in particular Dr. John O’Dwyer, for their helpful conversations during the research.

I want to take the opportunity to thank my family and friends for their support and interest throughout the research.
To my family.
Abstract

Drug delivery technologies are an important area within biomedicine. Targeted drug delivery aims to reduce the undesired side effects of drug usage by directing or capturing the active agents near a desired site within the body. This is particularly beneficial in, for instance, cancer chemotherapy, where the side effects of general (systemic) drug administration can be severe.

One approach to targeted drug delivery uses magnetic nanoparticles as the constituents of carriers for the desired active agent. Once injected into the body, the behaviour of these magnetic carriers can be influenced and controlled by magnetic fields. In implant assisted magnetic drug targeting systems a magnetic implant, typically a stent, wire or spherical seed can be used to target sites deep within the body as the implant acts as a focus for the resulting magnetic force. This can be easily understood as the force depends on the gradient of the magnetic field and the gradient near the implant is large.

In designing such a system many factors need to be considered including physical factors such as the size and nature of the implants and carriers, and the fields required. Moreover the range of applicability of these systems in terms of the regions of the vasculature system, from low blood velocity environments, such as capillary beds to higher velocity arteries, must be considered. Furthermore, assessment criteria for these systems are needed. Mathematical modelling and simulation has a valuable role to play in informing in vitro and in vivo experiments, leading to practical system design.

Specifically, the implant assisted magnetic drug targeting systems of Avilés, Ebner and Ritter are considered within this work, and two dimensional mathematical modelling is performed using the open source C++ finite volume library OpenFOAM. In the first system treated, a large ferromagnetic particle is implanted into a capillary bed as a seed to aid collection of single domain nanoparticles (radius 20-100 nm). The Langevin function is used to calculate the magnetic moment of the particles, and the model is further adapted to treat the agglomeration of particles known to occur in these systems. This agglomeration can be attributed to interparticle interactions and here the magnetic dipole-dipole and hydrodynamic interactions for two mutually interacting nanoparticles are modelled, following Mikkelsen et al. who treated two particle interactions in microfluidic systems, with low magnetic field (0.05 T). The resulting predicted performance is found to both increase and decrease significantly depending on initial positions of the particles. Secondly, a ferromagnetic, coiled wire stent is implanted in a large arterial vessel. The magnetic dipole-dipole and hydrodynamic interactions for multiple (N < 20) particles are included. Different initial positions are considered and the system performance is assessed. Inclusion of these interactions yields predictions that are in closer agreement with the experimental results of Avilés et al.. We conclude that the discrepancies between the non-interacting theoretical predictions and the corresponding experimental results can (as suggested by Avilés et al.) be largely attributed to interparticle interactions and the consequent agglomeration.
Contents

List of Figures v
List of Symbols vii
List of Acronyms xi

1 Introduction to Magnetic Drug Targeting 1
  1.1 Magnetic Targeted Nanoparticles in Biomedicine ................................. 1
  1.2 Review of Mathematical Modelling of Magnetic Drug Targeting ................ 4

2 Magnetic Fundamentals 11
  2.1 Magnetic Properties of Ferromagnets .................................................. 11
      2.1.1 Fundamental Quantities, $\vec{H}$, $\vec{B}$ and $\vec{M}$ ..................... 11
      2.1.2 Permeability and Susceptibility ................................................. 14
      2.1.3 Hysteresis and Saturation Magnetisation .................................... 15
      2.1.4 Remanence and Coercivity ....................................................... 16
      2.1.5 Curie Temperature ...................................................................... 17
      2.1.6 Hard and Soft Ferromagnetic Materials .................................... 17
  2.2 Paramagnetism and the Langevin Function ............................................. 17
  2.3 Single Domain Ferromagnetic Particles and Superparamagnetism ................ 18
      2.3.1 Weiss Domain Theory of Ferromagnetism ..................................... 18
      2.3.2 Magnetic Anisotropy .................................................................. 20
          2.3.2.1 Magnetocrystalline (Crystal Structure) Anisotropy ............... 20
          2.3.2.2 Shape Anisotropy ............................................................... 21
      2.3.3 Single Domain Particles ............................................................... 21
      2.3.4 Pseudo-Single Domain ................................................................. 22
      2.3.5 Superparamagnetism .................................................................. 22
      2.3.6 Hysteresis Properties of Different Size Particles ......................... 23
  2.4 Nanoparticles ......................................................................................... 24
  2.5 Magnetic Nanoparticles ........................................................................ 24
      2.5.1 Other Applications of Magnetic Nanoparticles in Biomedicine .......... 24
          2.5.1.1 Magnetic Resonance Imaging Contrast Agents for Monitoring Drug Delivery ................................................................................................. 24
          2.5.1.2 Magnetic Fluid Hyperthermia .................................................. 25

3 The Basic Mathematical Model 26
  3.1 Introduction ............................................................................................ 26
  3.2 Physical Domains ................................................................................... 26
      3.2.1 The Capillary Bed used in the Seed Model ................................... 26
      3.2.2 The Single Vessel used in the Stent Model .................................... 28
  3.3 Derivation of Particle Velocity ............................................................... 29
  3.4 Calculation of Blood Velocity ............................................................... 30
      3.4.1 Introduction to Navier-Stokes Equations ..................................... 30
CONTENTS

3.4.2 Navier-Stokes Equations in Dimensionless Form ............... 31
3.5 Derivation of Streamlines, Capture Cross Section and Collection Efficiency 33

4 Implementation in OpenFOAM 34
4.1 Introduction to OpenFOAM ........................................ 34
4.2 Navier-Stokes Equations in OpenFOAM .................................. 36
4.3 Calculation of Magnetic Force ........................................ 37
4.3.1 Calculation of Magnetic Scalar Potential ............................ 38
4.3.1.1 2D models: Circular Implant and Polar Coordinates ........... 39
4.3.1.2 Background Source Field ....................................... 40
4.3.1.3 Analytic Solution of Magnetic Scalar Potential ................. 40

5 Development of Model to Include Interactions and Results 44
5.1 Inclusion of the Langevin Function .................................... 45
5.1.1 Theory .................................................................. 45
5.1.2 Results .................................................................. 46
5.2 Inclusion of Interparticle Dipole-Dipole Interaction in the Model ............... 48
5.3 Inclusion of Hydrodynamic Interaction in the Model .......................... 50
5.4 Inclusion of Magnetic Dipole-Dipole and Hydrodynamic Interactions for Two MDCPs — Seed Model .......................................................... 51
5.4.1 Effect of Interactions on the Agglomeration of MDCPs ............ 52
5.4.2 Effect of Interactions on the Capture Cross Section of the System 54
5.5 Inclusion of Magnetic Dipole-Dipole and Hydrodynamic Interactions for Multiple MDCPs - Stent Model ....................................................... 56

6 Conclusions 67

A Scalar Potential Formulation 76
A.1 Calculation of Permeability of the Implant ................................. 76
A.2 Solution Using Total-Total Potential Formulation ...................... 77
A.3 Solution Using Total–Reduced Potential Formulation .......................... 79
A.4 Derivation of Magnetic Force Density .................................... 80

B OpenFOAM Code for Seed Model 81
B.1 Applications (Solver) for Seed Implant ..................................... 81
B.1.1 createFields.H file ................................................. 81
B.1.2 createFields-analytic.H file ....................................... 83
B.1.3 readtwoRegionDict.H file ....................................... 87
B.1.4 interactionFoam.C file ............................................ 88
B.2 Run (Case) for Seed Implant ................................................ 97
B.2.1 0 (initial conditions file) ............................................ 97
B.2.1.1 epsilon file ................................................ 97
B.2.1.2 k file .................................................... 98
B.2.1.3 nuTilda file ................................................ 99
B.2.1.4 p file .................................................... 99
B.2.1.5 phi file ................................................ 100
B.2.1.6 R file ................................................ 101
List of Figures

1.1 Experimental and theoretical CE results of Avilés et al. (source: \cite{8}).

2.1 Schematic representation of the $B-H$ curve of a ferromagnetic material \cite{45}.

2.2 Schematic representation of the $M-H$ curve of a typical ferromagnetic material \cite{2}.

2.3 Spontaneous magnetisation, $M_s$, of a ferromagnet as a function of the temperature, $T$, normalised to the Curie Temperature, $T_c$. The applied field is assumed to be small, but finite, as it is in real measurements \cite{2}.

3.1 Schematic diagram of the control volume representing a capillary bed.

3.2 Schematic diagram of the control volume representing a vessel with stent.

4.1 General problem of object (region 1 = implant) embedded in a space region (region 2 = space) of constant permeability \cite{76}.

4.2 Interface conditions across the boundary of regions with different permeability.

5.1 Streamlines indicating the trajectories of the single domain nanoparticles, calculated using OpenFOAM, as they traverse the control volume for different magnitudes of the externally applied magnetic field, $H_0$.

5.2 Capture cross section, $\lambda_c$, plotted as a function of the applied magnetic field strength calculated using the Langevin function and following Avilés et al.

5.3 Distance of agglomeration point from the seed plotted against initial distance between the MDCPs, $D$, with (a) dipole-dipole magnetic interaction only, (b) hydrodynamic interaction only, (c) both interactions and (d) no interactions between the MDCPs. All other conditions are as the reference case condition in table 5.1.

5.4 The trajectories of the MDCPs are presented with the magnetic dipole-dipole interaction and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda_* + 0.25)$ & $(-20, \lambda_* - 0.25)$. With the magnetic dipole-dipole interaction both MDCPs are captured.

5.5 The trajectories of the MDCPs are presented with the magnetic dipole-dipole interaction and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda_* + 0.35)$ & $(-20, \lambda_* - 0.15)$. With the magnetic dipole-dipole interaction, neither MDCP is captured.

5.6 The trajectories of the MDCPs are presented with the hydrodynamic interaction and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda_* + 0.255)$ & $(-20, \lambda_* - 0.255)$. With the hydrodynamic interaction both MDCPs are captured.
5.7 The trajectories of the MDCPs are presented with the hydrodynamic interaction and without any interaction. Initial position of MDCP 1 & 2 are $(-20, \lambda^* + 0.644)$ & $(-20, \lambda^* + 0.134)$. With hydrodynamic interaction, MDCP 2 is now captured.

5.8 The trajectories of the MDCPs are presented with both interactions and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda^* + 0.32)$ & $(-20, \lambda^* - 0.32)$. With both interactions both MDCPs captured.

5.9 The collection efficiency (CE) of the system plotted as a function of the blood velocity at the applied field $\mu_0 H_0 = 0.65$ T.

5.10 The collection efficiency (CE) of the system plotted as a function of the blood velocity at the applied field $\mu_0 H_0 = 0.17$ T.

B.1 Structure of the OpenFOAM program
# List of Symbols

## Roman Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A_c)</td>
<td>Area of the electric current loop</td>
<td>13</td>
</tr>
<tr>
<td>(\vec{B})</td>
<td>Magnetic flux density or magnetic induction</td>
<td>11</td>
</tr>
<tr>
<td>(\vec{B}_{\text{total}})</td>
<td>Total magnetic flux density acting on MDCP (n)</td>
<td>49</td>
</tr>
<tr>
<td>(B_r)</td>
<td>Remanent magnetic induction</td>
<td>16</td>
</tr>
<tr>
<td>(C)</td>
<td>Curie constant</td>
<td>17</td>
</tr>
<tr>
<td>(D)</td>
<td>Initial interparticle distance between MDCPs</td>
<td>52</td>
</tr>
<tr>
<td>(\vec{E})</td>
<td>Electric field strength</td>
<td>12</td>
</tr>
<tr>
<td>(\vec{F}_{\text{hyd}})</td>
<td>Force due to the hydrodynamic interaction which acts on MDCP (n)</td>
<td>50</td>
</tr>
<tr>
<td>(\vec{F}_i)</td>
<td>Inertial forces</td>
<td>29</td>
</tr>
<tr>
<td>(\vec{F}_n)</td>
<td>Inertial forces of MDCP (n)</td>
<td>50</td>
</tr>
<tr>
<td>(\vec{F}_{\text{int}})</td>
<td>Modified magnetic force of MDCP (n)</td>
<td>50</td>
</tr>
<tr>
<td>(\vec{F}_m)</td>
<td>Magnetic forces</td>
<td>29</td>
</tr>
<tr>
<td>(\vec{F}_s)</td>
<td>Stokes drag</td>
<td>20</td>
</tr>
<tr>
<td>(\vec{F}_{\text{sn}})</td>
<td>Stokes drag of MDCP (n)</td>
<td>50</td>
</tr>
<tr>
<td>(f_0)</td>
<td>Frequency pre-factor</td>
<td>23</td>
</tr>
<tr>
<td>(f_w)</td>
<td>The magnetic force density</td>
<td>80</td>
</tr>
<tr>
<td>(\vec{H})</td>
<td>Magnetic field strength</td>
<td>11</td>
</tr>
<tr>
<td>(H_c)</td>
<td>Coercive field</td>
<td>16</td>
</tr>
<tr>
<td>(\vec{H}_0)</td>
<td>Externally applied magnetic field</td>
<td>29</td>
</tr>
<tr>
<td>(h)</td>
<td>Distance between the centres of the stent wires</td>
<td>28</td>
</tr>
<tr>
<td>(i)</td>
<td>Current in the circuit</td>
<td>13</td>
</tr>
<tr>
<td>(K_a)</td>
<td>Anisotropy constant</td>
<td>23</td>
</tr>
<tr>
<td>(k_B)</td>
<td>Boltzmann’s constant</td>
<td>18</td>
</tr>
<tr>
<td>(L)</td>
<td>Length of coiled stent</td>
<td>61</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>$\vec{M}$</td>
<td>(Volume) Magnetisation</td>
<td>13</td>
</tr>
<tr>
<td>$\vec{m}$</td>
<td>Magnetic moment</td>
<td>13</td>
</tr>
<tr>
<td>$\vec{m}_n$</td>
<td>Total magnetic moment of MDCP $n$</td>
<td>49</td>
</tr>
<tr>
<td>$M_{fm,p}$</td>
<td>Magnetisation of the ferromagnetic material in the MDCP</td>
<td>77</td>
</tr>
<tr>
<td>$M_{fm,p,s}$</td>
<td>Saturation magnetisation of the ferromagnetic material in the MDCP</td>
<td>76</td>
</tr>
<tr>
<td>$M_{\text{implant}}$</td>
<td>Magnetisation of the implant</td>
<td>77</td>
</tr>
<tr>
<td>$M_{\text{implant},s}$</td>
<td>Saturation magnetisation of the implant</td>
<td>77</td>
</tr>
<tr>
<td>$M_r$</td>
<td>Remanent magnetisation</td>
<td>16</td>
</tr>
<tr>
<td>$M_s$</td>
<td>Spontaneous magnetisation</td>
<td>18</td>
</tr>
<tr>
<td>$M_{\text{sat}}$</td>
<td>Saturation magnetisation</td>
<td>19</td>
</tr>
<tr>
<td>$N_{\text{Eu}}$</td>
<td>Euler numbers</td>
<td>32</td>
</tr>
<tr>
<td>$N_l$</td>
<td>Number of loops in stent</td>
<td>61</td>
</tr>
<tr>
<td>$N_{\text{Re}}$</td>
<td>Reynolds numbers</td>
<td>32</td>
</tr>
<tr>
<td>$\hat{n}$</td>
<td>The normal vector of the interface</td>
<td>39</td>
</tr>
<tr>
<td>$n_a$</td>
<td>The number of atoms per unit volume</td>
<td>18</td>
</tr>
<tr>
<td>$P$</td>
<td>Pressure</td>
<td>30</td>
</tr>
<tr>
<td>$P_0$</td>
<td>Pressure at the outlet of the CV</td>
<td>31</td>
</tr>
<tr>
<td>$R_{\text{implant}}$</td>
<td>Radius of the implant</td>
<td>33</td>
</tr>
<tr>
<td>$R_p$</td>
<td>Radius of the MDCP</td>
<td>29</td>
</tr>
<tr>
<td>$R_{p,i}$</td>
<td>Radius of the MDCP $i$</td>
<td>51</td>
</tr>
<tr>
<td>$R_{p,n}$</td>
<td>Radius of the MDCP $n$</td>
<td>50</td>
</tr>
<tr>
<td>$R_{\text{vessel}}$</td>
<td>Radius of vessel</td>
<td>28</td>
</tr>
<tr>
<td>$R_{\text{wire}}$</td>
<td>Radius of the wire (implant)</td>
<td>28</td>
</tr>
<tr>
<td>$\vec{r}$</td>
<td>Represents an arbitrary point in space</td>
<td>49</td>
</tr>
<tr>
<td>$\vec{r}_n$</td>
<td>Position of the MDCP $n$</td>
<td>49</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
<td>17</td>
</tr>
<tr>
<td>$T_B$</td>
<td>Blocking temperature</td>
<td>29</td>
</tr>
<tr>
<td>$T_c$</td>
<td>Curie temperature</td>
<td>17</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>$\hat{t}$</td>
<td>The tangential vector of the interface</td>
<td>39</td>
</tr>
<tr>
<td>$t_r$</td>
<td>Relaxation time</td>
<td>23</td>
</tr>
<tr>
<td>$u_0$</td>
<td>Inlet (average) velocity</td>
<td>27</td>
</tr>
<tr>
<td>$x_{fm,p}$</td>
<td>Weight fraction</td>
<td>46</td>
</tr>
<tr>
<td>$V$</td>
<td>Volume</td>
<td>13</td>
</tr>
<tr>
<td>$V_B$</td>
<td>Blocking volume</td>
<td>23</td>
</tr>
<tr>
<td>$V_p$</td>
<td>Volume of the MDCP</td>
<td>46</td>
</tr>
<tr>
<td>$V_{p_n}$</td>
<td>Volume of MDCP $n$</td>
<td>49</td>
</tr>
<tr>
<td>$\vec{v}_b$</td>
<td>Blood velocity</td>
<td>27</td>
</tr>
<tr>
<td>$\vec{v}_p$</td>
<td>Velocity of the MDCP</td>
<td>29</td>
</tr>
<tr>
<td>$\vec{v}_{p_n}$</td>
<td>Velocity of MDCP $n$</td>
<td>50</td>
</tr>
</tbody>
</table>

**Greek Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{fm,p}$</td>
<td>Demagnetising factor for a ferromagnetic material</td>
<td>76</td>
</tr>
<tr>
<td>$\alpha_{implant}$</td>
<td>Demagnetising factor for the implant</td>
<td>77</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Langevin argument</td>
<td>49</td>
</tr>
<tr>
<td>$\chi$</td>
<td>Magnetic susceptibility</td>
<td>14</td>
</tr>
<tr>
<td>$\chi_{fm,p}$</td>
<td>Volumetric magnetic susceptibility of the ferromagnetic material in the MDCP with field</td>
<td>77</td>
</tr>
<tr>
<td>$\chi_{fm,p,0}$</td>
<td>Volumetric magnetic susceptibility of the ferromagnetic material in the MDCP with zero field</td>
<td>76</td>
</tr>
<tr>
<td>$\chi_{implant}$</td>
<td>Volumetric magnetic susceptibility of the implant</td>
<td>77</td>
</tr>
<tr>
<td>$\chi_{implant,0}$</td>
<td>Volumetric magnetic susceptibility of the implant with zero field</td>
<td>77</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Viscosity of the fluid</td>
<td>36</td>
</tr>
<tr>
<td>$\eta_b$</td>
<td>Viscosity of the blood</td>
<td>29</td>
</tr>
<tr>
<td>$\eta_{eff}$</td>
<td>Kinematic viscosity of the fluid</td>
<td>37</td>
</tr>
<tr>
<td>$\lambda_c$</td>
<td>Capture radius</td>
<td>27</td>
</tr>
<tr>
<td>$\lambda_c^*$</td>
<td>The resulting boundary of the reference capture cross section</td>
<td>54</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Permeability of the medium</td>
<td>12</td>
</tr>
<tr>
<td>$\mu_{\text{implant}}$</td>
<td>Relative permeability of implant</td>
<td>43</td>
</tr>
<tr>
<td>$\mu_0$</td>
<td>Permeability of free space</td>
<td>12</td>
</tr>
<tr>
<td>$\mu_r$</td>
<td>Relative permeability</td>
<td>14</td>
</tr>
<tr>
<td>$\mu_{\text{space}}$</td>
<td>Relative permeability of space</td>
<td>77</td>
</tr>
<tr>
<td>$\omega_{\text{fm,p}}$</td>
<td>Volume fraction of ferromagnetic material</td>
<td>46</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Magnetic scalar potential (Reduced–reduced)</td>
<td>30</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Magnetic scalar potential (Total–total)</td>
<td>77</td>
</tr>
<tr>
<td>$\psi_s$</td>
<td>Stream function</td>
<td>33</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density of the fluid</td>
<td>36</td>
</tr>
<tr>
<td>$\rho_b$</td>
<td>Density of the blood</td>
<td>30</td>
</tr>
<tr>
<td>$\rho_{\text{fm,p}}$</td>
<td>Density of the ferromagnetic material in the MDCPs</td>
<td>46</td>
</tr>
<tr>
<td>$\rho_{\text{pol,p}}$</td>
<td>Density of the polymer material in the MDCPs</td>
<td>46</td>
</tr>
<tr>
<td>$\bar{\tau}$</td>
<td>Torque on a magnetic dipole</td>
<td>13</td>
</tr>
<tr>
<td>$\tau_{\text{max}}$</td>
<td>Maximum torque on a magnetic dipole</td>
<td>13</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>Magnetic field orientation</td>
<td>30</td>
</tr>
</tbody>
</table>
**List of Acronyms**

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM</td>
<td>Computational Continuum Mechanics</td>
<td>34</td>
</tr>
<tr>
<td>CCS</td>
<td>Capture Cross Section</td>
<td>6</td>
</tr>
<tr>
<td>CE</td>
<td>Collection Efficiency</td>
<td>4</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
<td>34</td>
</tr>
<tr>
<td>CV</td>
<td>Control Volume</td>
<td>6</td>
</tr>
<tr>
<td>DE</td>
<td>Diversion Efficiency</td>
<td>5</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite Element Method</td>
<td>34</td>
</tr>
<tr>
<td>FVM</td>
<td>Finite Volume Method</td>
<td>34</td>
</tr>
<tr>
<td>HGMS</td>
<td>High Gradient Magnetic Separation</td>
<td>5</td>
</tr>
<tr>
<td>IA-MDT</td>
<td>Implant Assisted Magnetic Drug Targeting</td>
<td>3</td>
</tr>
<tr>
<td>MD</td>
<td>Multiple Domain</td>
<td>21</td>
</tr>
<tr>
<td>MDCP</td>
<td>Magnetic Drug Carrier Particle</td>
<td>2</td>
</tr>
<tr>
<td>MDT</td>
<td>Magnetic Drug Targeting</td>
<td>1</td>
</tr>
<tr>
<td>MFH</td>
<td>Magnetic Fluid Hyperthermia</td>
<td>24</td>
</tr>
<tr>
<td>MIS</td>
<td>Magnetisable Intravascular Stent</td>
<td>7</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic Resonance Imaging</td>
<td>2</td>
</tr>
<tr>
<td>OpenFOAM</td>
<td>Open Field Operation and Manipulation</td>
<td>1</td>
</tr>
<tr>
<td>PSD</td>
<td>Pseudo-Single Domain</td>
<td>21</td>
</tr>
<tr>
<td>SD</td>
<td>Single Domain</td>
<td>21</td>
</tr>
<tr>
<td>SPM</td>
<td>Superparamagnetic</td>
<td>21</td>
</tr>
<tr>
<td>SS</td>
<td>Stainless Steel</td>
<td>7</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction to Magnetic Drug Targeting

1.1 Magnetic Targeted Nanoparticles in Biomedicine

In this introductory chapter, we present some general information concerning magnetic drug targeting (MDT) and the motivation for this work. Also, a brief history of MDT and related applications is given. This is followed by a chapter on the magnetic fundamentals relevant to MDT. The third chapter deals with the fluid dynamics, specifically the Navier-Stokes equations, and the basic MDT model which is considered. Next, we give some general information about OpenFOAM (Open Field Operation and Manipulation) a finite volume simulation C++ library, used in this work. The fifth chapter outlines the results of the Ph.D. and the conclusions are presented in the last chapter. The analytical solution of magnetic scalar potential, C++ finite volume library code for seed model, mesh generator, C++ finite volume library code for the stent model and associated publications are given in appendices A to E.

The notable properties of magnetic nanoparticles [24, 26, 27] have been exploited to good effect in many applications, particularly in magnetic recording. These particles can be used in many potential applications in biomedicine as a result of their particular physical properties [14, 15, 60, 65]. To begin with, for biological applications nanoparticles can be prepared with sizes comparable to a cell (10–100 µm), a virus (20–450 nm), a protein (5–50 nm), or a gene (2 nm wide and 10–100 nm long). Furthermore, they can be coated with biological molecules to facilitate biocompatibility in the body. In addition, if the nanoparticles are magnetic they can be controlled by an external magnetic field. This allows magnetic nanoparticles to transport therapeutic agents such as anticancer
drugs, genes \cite{71} or radionuclides \cite{37,72} to a targeted site in the body, such as a tumour. Moreover, the magnetic nanoparticles can react to a time-varying magnetic field. Thus, they can be heated and used as hyperthermic agents to deliver a toxic amount of thermal energy to the targeted site of the body, such as a tumour \cite{46}. To begin with, we discuss the use of magnetic nanoparticles in three particular applications:

- magnetic drug targeting,
- magnetic resonance imaging (MRI) contrast enhancement \cite{59,74},
- hyperthermia treatments.

Targeted delivery and vascular treatment use the particles as carriers for the appropriate therapeutic agents which are manipulated under the control of magnetic fields \cite{49,66}. In contrast enhancement, the metastatic lymph nodes absorb the nanoparticles more than inflamed nodes and this is detectable with MRI \cite{59,74}. Hyperthermic treatment involves heating, via radio-frequency fields, within the range $41^\circ - 46^\circ$ C, damaging cells but also significantly assisting radiotherapy of tumour cells \cite{65}.

Although, biomedical applications of magnetic nanoparticles have been proposed since the 1950s \cite{34,38}, recent advances in nanotechnology have meant that many of the technical problems associated with production and biocompatibility \cite{50,51,52} are now being solved. Significantly, in March 2005, the US Food and Drug Administration gave approval to Advanced Magnetics \cite{1} for the use of Combidex, an ultra small superparamagnetic compound, for human injection as an MRI contrast agent. Similar progress in the other applications has followed, with Magforce \cite{53} (CEO Dr Andreas Jordan, see reference \cite{46}) producing drug delivery compounds and systems.

The development of more effective drug treatment methodologies is an area of much research. In most drug delivery systems much of any drug administered to patients does not reach its target site. The aim of drug targeting is to decrease the amount of drug delivered to healthy tissue, while maintaining the therapeutic action at the site. One such approach is MDT. For instance, magnetic nanoparticles can be employed as carriers in a cancer treatment, thereby avoiding the side effects of conventional chemotherapy \cite{31,77}. MDT typically uses an external magnetic field source to capture and retain magnetic drug carrier particles (MDCPs) at a specific site after being injected into the body. Studies
have shown that MDT is a relatively safe and effective methodology for targeting drugs to a specific site in the body \cite{50, 51, 52}. However, there are some significant limitations of MDT. One limitation associated with MDT is the gradient problem, that is the magnetic force requires a magnetic field gradient; thus it can be difficult using external magnets only to target areas deep within the body, without targeting the surface more strongly \cite{11, 36}. To overcome this problem several authors \cite{5, 6, 8, 9, 10, 22, 41, 66, 68, 83} have proposed implanting ferromagnetic materials such as wires, seeds and stents within the body. In a homogeneous magnetic field these implants create strong localised gradients, and this approach is known as implant assisted MDT (IA-MDT). This IA-MDT approach consists of three components. First, it uses standard magnets that provide a long-range, low-gradient magnetic field. Second, it uses an implant that creates a localised high-gradient magnetic field when it is magnetised by a low-gradient magnetic field. Third, it uses MDCPs designed to aggregate and/or collect only when and where they come across a high-gradient magnetic field. Of the various IA-MDT implants suggested by Ebner, Ritter and co-workers \cite{5, 6, 7, 8, 9, 10, 22, 66, 68} firstly, we consider a seed as the implant with single domain magnetic nanoparticles as MDCPs \cite{28, 29} and secondly, we consider the stent as the implant, with MDCPs containing magnetic single domain nanoparticles \cite{30}.

Stent technology is well advanced in Ireland with many companies such as Abbott, Clearstream, Stryker, Cordis Corporation and Boston Scientific involved. Their success in cardiovascular treatment is well known, with many people in Ireland having stents in their bodies. Patients need only spend a day in hospital to have a stent fitted, compared to more than a week following a bypass operation. In the USA, more than 500,000 heart stents are placed each year and in the UK, around 70,000 patients receive heart stents each year. In this work a magnetisable stent is chosen as one of the proposed implant.

Avilés et al. \cite{8} compared the (non-interacting) particle model of this stent system with \textit{in vitro} experimental arrangement using a ferromagnetic stent made in the shape of a coil. Their results indicated that at low fluid velocity more particles were collected than predicted. Furthermore, they suggested that particle agglomeration (due to interparticle interactions) might explain this. With this in mind, we have developed their mathematical model to include both dipole-dipole and hydrodynamic interactions between many particles. These theoretical results are presented here and are compared with the \textit{in vitro}
experiments of Avilés et al.. We note that Mikkelsen et al. [21] have included both the hydrodynamic and dipole-dipole interactions for the case of low magnetic fields by considering high gradient magnetic separation. Also, Mehasni et al. have considered the effect of magnetic dipole-dipole interaction on the performance of high gradient magnetic separation systems [57]. In this work, we calculate the effect of interactions of many particles on the collection efficiency (CE) of the system allowing the agglomeration of particles. Simulations are obtained using OpenFOAM a finite volume simulation C++ library.

1.2 Review of Mathematical Modelling of Magnetic Drug Targeting

The work presented in this thesis concerns MDT. One of the basic requirements in drug management is that sufficient quantities of the drug reach the desired site in the body. Taking a systemic approach, in order to have a sufficient amount of drug at the intended organ or disease site, large doses of drug have to be taken into the body. These high doses can harm the non-target organs and cells of the body. To avoid this, methods are being developed to target the desired site and to decrease the amount of drug at non-target sites. In MDT, this can be achieved by attaching the drugs to magnetic nanoparticles to produce MDCPs and controlling these by means of external magnetic fields. The traditional way of applying magnetic fields to target drugs is to locate the permanent magnets directly over the affected site in the body. The magnet creates a magnetic field and gradients that are theoretically strong enough to collect MDCPs.

Specifically, the work presented here concerns the theoretical modelling of the behaviour of MDCPs in the cardiovascular system. Historically, the pioneering work of Senyai et al. [73] showed that it should be possible to attract the particles within the human body to specific locations with the use of external magnetic fields. Their model considered the basic physical laws of the behaviour of one particle flowing in the presence of an externally applied magnetic field gradient and undergoing Stokes drag. They considered a broad range of flow rates, from 0.05 cm/s to 10 cm/s that occur throughout the human cardiovascular system. Predictions from this model were used to inform in vitro and later in vivo experiments. Inherent in their model was the assumption of smooth
vessel walls and the in vitro experiments used a continuous flow rate as opposed to the pulsatile flow that is a dominant effect near the heart. The cellular constituents of the blood were not considered, and the problems arising from junctions in the vessel network were not considered. Grief and Richardson advanced the theoretical models of this hydrodynamic problem by incorporating the diffusive effects of particle interactions with the red blood cells and the problem of junctions [36]. They performed numerical calculations for a two dimensional network based on this model. Significantly their results suggested limitations in the control of the particles which could be affected by the use of external magnets. In particular, they suggest that it is not possible to target interior regions of the body (deeper than 2 cm [66]) without targeting some of the surrounding regions of the body more strongly. A related problem is that the blood velocity in large arteries is 50-100 times larger than the blood velocity in capillaries. Therefore a large external magnetic field is required to collect the MDCPs in large arteries [66]. This magnetic force problem has been addressed by Babincova et al. [11] who showed that increasing the strength of the magnetic field is not the only way to increase the collection of MDCPs. The force acting on a magnetic particle is directly proportional to both the strength and the gradient of the magnetic field. A larger gradient of the magnetic field results in a greater force on the MDCPs. One way to locally increase the gradient of the magnetic field is to place a ferromagnetic implant in the region of the magnetic field and this approach is known as implant assisted magnetic drug targeting (IA-MDT). Taking this approach, James Ritter and co-workers at the University of South Carolina have proposed the use of a wire [66], a seed [6] and a stent [7, 22, 68] as the implant.

In particular, Ritter et al. [66] studied, in FEMLAB simulations, a theoretical MDT system using high gradient magnetic separation (HGMS) principles. In this, their original model, they used a ferromagnetic wire placed inside the blood vessel as an implant. They applied an external magnetic field to control the MDCPs in the system and to keep them at the target site of the system. They defined collection (CE) and diversion (DE) efficiencies to assess the system performance. In particular, the CE is defined as the percentage of the MDCPs captured by the magnetised implant. In their system the changes in CE and DE are studied as the following vary:

- the strength of the applied magnetic field (0.3–2.0 T),
• the amount of ferromagnetic material (iron) in the MDCP (20–100 wt%),

• the size of the MDCPs (1–10 μm radius),

• average inlet velocity (0.1–0.8 m/s),

• the size of the wire (50–250 mm radius),

• the ratio (4–10) of the parent vessel radius to the wire radius (0.25–1.25 mm radius).

It was seen that the effect of the applied magnetic field direction on CE and DE was small. Under the above conditions, 70% CEs and 30% DEs were achieved, and when the MDCPs were allowed to agglomerate, 100% CEs were achieved. (The treatment of agglomeration here amounted to viewing the agglomerated cluster as a larger particle with significant porosity.) Ritter et al. conclude that their MDT system for collecting MDCPs at the target site is promising. Their proposed system to divert MDCPs through the circulatory system also shows promise but is in their view more limited. The Ritter et al. [66] study suggested that MDT using HGMS principles had significant promise with many potential applications.

Ritter’s group (Avilés et al. [6]) later developed an IA-MDT model which uses ferromagnetic particles with a seed as the implant for collecting MDCPs at the target site in the body, specifically in a capillary bed near a tumour. Here, they used a capture cross section approach, to assess the system performance. Capture cross section (CCS) is the size of the capture radius expressed as a multiple of the seed (implant) radius, where the capture radius is defined by the location of streamline at the entrance to the control volume (CV) of the last MDCP captured by the seed (implant) (see figure 3.1). In their 2D mathematical model, the changes in CCS were studied in FEMLAB for different values of:

• the magnetic field strength,

• MDCP radius,

• MDCP ferromagnetic material weight content,

• average blood velocity,
seed radius, number of seeds and seed separation.

Avilés et al. used different magnetic materials such as iron, stainless steel (SS) 409, magnetite, and SS 304 for the MDCP and seed. Increasing the magnetic field strength, MDCP size, seed size, MDCP ferromagnetic material content, or MDCP or seed saturation magnetisation significantly increased the system performance while, as expected, increasing the average blood velocity decreased the performance. The number of seeds and the seed separation had relatively small changes in the system performance. The study of Avilés et al. [6] indicates that using seeds as implants for IA-MDT has significant effect in targeting drugs in capillary beds.

Ritter’s group (Chen et al. [22]) also developed a 2D mathematical model and, in FEMLAB simulations, studied the collection of MDCPs by a magnetisable intravascular stent (MIS) implant and used CE to assess the system performance. In their system the changes in CE are studied as the following parameters vary over a wide range of realistic conditions:

- the blood flow rate,
- magnetic field strength and direction,
- MDCP properties,
- stent design parameters such as MIS radius, wire radius, number of MIS loops, wire loop spacing and MIS ferromagnetic material.

Chen et al. [22] show that MDT using an MIS has significant promise. Furthermore it is worth noting that stent technology is already well established for the treatment of many cardiovascular conditions.

More recently Ritter’s group (Avilés et al. [7]) studied IA-MDT in vitro using a coiled ferromagnetic wire stent made from SS 430 or 304, and MDCPs which were made using polystyrene and 20 wt% magnetite. They employed CE to assess the system performance. In their system the changes in CE are studied as the following parameters vary:

- the fluid velocity,
- MDCP concentration,
• magnetic field strength,

• stent material.

It was seen that all the above parameters are important for the quantity of the MDCPs captured. This study confirms *in vitro* that MDT using an MIS is effective in attracting and collecting MDCPs at the desired site.

Elsewhere, Furlani and Furlani [33] studied a mathematical model of IA-MDT. They considered the dominant magnetic and fluidic forces on an MDCP and derived an analytic expression for predicting its trajectory in a microvessel. Also, their model allows parametric analysis of magnetic targeting as a function of key variables:

• size of the carrier particle,

• the properties and volume fraction of the magnetic nanoparticles,

• the properties of the magnet, the microvessel and the blood properties.

Furthermore, their results show that magnetic targeting can be achieved using submicron carrier particles when the tumour is within a few centimetres of the surface of the body.

A significant effect at the target site is that agglomeration of the particles can occur. This is due to the interparticle magnetic dipole-dipole interaction, which becomes significant as the magnetic field brings the particles closer together, overcoming the designed repulsion inherent in the ferrofluid state. This can lead to partial or total vessel occlusion. Also, this leads to changes in the particle trajectories and should be incorporated in the modelling [41]. Furthermore, past the target site de-agglomeration should occur and has been observed [35] but this phenomenon had not been successfully modelled. A primary motivation for the work in this thesis was to achieve realistic modelling of these phenomena.

Recently, Mikkelsen et al. [21, 58] calculated the magnetic dipole-dipole and hydrodynamic interactions between magnetic beads under low magnetic fields (0.05 T) in microfluidic systems. Firstly, in their model, as a result of dipole-dipole interaction, magnetic beads (which behave as small implants) increase the gradient of the magnetic field acting on the other magnetic beads. Therefore the magnetic force acting on the magnetic beads is also increased and they interact with each other. Also, the total magnetic force
is increased because of the change in magnetisation. Secondly, the movement of each magnetic beads affects the fluid flow which in turn affects the other beads, leading to a hydrodynamic interaction. In conclusion, in their model, they highlighted the importance of hydrodynamic interactions during the capturing of the magnetic beads. They showed that the effect of hydrodynamic interactions on bead capturing cannot be ignored when treating agglomeration, particularly for large particles.

Specifically, in this work, we have developed the models of Ritter’s group, in particular the stent model to incorporate dipole-dipole and hydrodynamic interactions between the MDCPs. This is with a view to account for the recent results of Avilés et al. [8]. These show higher CE than predicted with low fluid velocity (≤ 15 cm/s) and lower CE than predicted for higher blood velocity (see figure 1.1). These they consider to be due to interparticle interaction resulting in agglomeration and shearing force effects.

Finally, we note in terms of future work, the complexity of the problems leaves many factors neglected in existing models. As outlined by Lübbe et al. [52]

Physiological as well as pharmacological parameters in magnetically controlled drug targeting warrant further investigation. This is because the efficacy of in vivo drug targeting with ferrofluids critically depends on physiological parameters. To understand this new form of pharmacological application as well as the mechanism of action of the concentrated drug in the tissue at the microcirculatory level one must consider not only the ferrofluids’ parameters (particle size, surface characteristics of the particle, concentration of the fluid, volume of the fluid, reversibility and strength of the drug/ferrofluid binding, desorption characteristics), but also access to the organism (infusion route/duration/rate of the injection/infusion time), geometry and strength of the magnetic field, and duration of the magnetic field application.
1. Introduction to MDT

1.2. Review of Mathematical Modelling

Figure 1.1: Experimental and theoretical CE results of Avilés et al. (source: [8]).

(a) Experimental and theoretical CE is plotted as a function of fluid velocity for 0.17 T and 0.65 T.

(b) Experimental and theoretical CE is plotted as a function of the applied magnetic field for the fluid velocities of 2.1 cm/s, 4.2 cm/s and 21.2 cm/s.
Chapter 2

Magnetic Fundamentals

A major aspect of this work involves the calculation and prediction of the behaviour of magnetic nanoparticles in the presence of magnetic fields. In order to perform these calculations it is beneficial to discuss some basics of magnetism and the relevant magnetic quantities. In this chapter, the different forms of magnetism are discussed and classified; in particular we consider ferromagnetism, paramagnetism and superparamagnetism with a view to understanding the nature of magnetic nanoparticles and their applications.

2.1 Magnetic Properties of Ferromagnets

2.1.1 Fundamental Quantities, $\vec{H}$, $\vec{B}$ and $\vec{M}$

In this section we clarify the terms magnetic field, magnetic flux density (or magnetic induction) and magnetisation. We begin by stating that any region of space which exhibits an influence on a magnet, for instance a compass needle, can be said to possess a magnetic field. The source of this magnetic field can be understood ultimately to be due to the presence of electric currents. In the case of electromagnets, the currents are clearly those circulating in the coils. In the case of magnetic materials, the source is attributed to uncompensated orbital or spin motion of the electrons within the atoms, which in the case of ferromagnets orient collectively through exchange.

As first observed by Oersted in 1819, a magnetic field can be created by current carrying conductor [62]. Oersted discovered that the direction of the current carried on a wire can determine a compass needle’s direction.

We next distinguish between the quantities magnetic field strength, $\vec{H}$, and the magnetic flux density, $\vec{B}$, also termed magnetic induction. The magnetic field strength created by an electric current can be calculated from the Biot-Savart law or from Ampère’s law [4] [45]. In the SI unit system the strength of magnetic field, $H$ is measured in amperes...
per metre (A/m) which indicates the relation of this quantity to the electric current.

On the other hand, magnetic flux density, $\vec{B}$, is the response of the medium to the magnetic field. It can be understood as the density of magnetic lines of force, or magnetic flux lines, passing through a particular area. The movement of a compass needle (a magnetic dipole) is clearly due to the applied torque on the compass needle. The strength of this torque is in turn determined by the strength of the magnetic induction, $\vec{B}$. Thus we note that $\vec{B}$, not $\vec{H}$, plays the role of the physical observable in magnetism, in the same way that the electric field strength, $\vec{E}$, does in electrostatics. In the SI unit system $\vec{B}$ is measured in webers per metre squared (Wb/m$^2$) and is equivalent to a magnetic induction of one tesla (T).

In free space the relation between magnetic field and magnetic flux density is simple and magnetic flux density is proportional to magnetic field strength,

$$\vec{B} = \mu_0 \vec{H}, \quad (2.1)$$

where $\mu_0$ is the permeability of free space and has value $\mu_0 = 4\pi \times 10^{-7}$H/m. On the other hand, for different media, magnetic flux density is not in general a linear function of magnetic field. However they can still be related in terms of the permeability of the medium, $\mu$, through,

$$\vec{B} = \mu \vec{H}, \quad (2.2)$$

where $\mu$ is in general not a constant and furthermore can be multivalued, as is the case with hysteresis.

Magnetisation relates to the contribution of the magnetic material to the magnetic flux density, $\vec{B}$, when a field is applied to the material. Magnetisation depends on the magnetic characteristic of the material. One expects larger magnetisation for ferromagnets than paramagnets or diamagnets. Magnetisation results from two sources: orbital motion of electrons around the nucleus and the spinning of electrons on their own axes. Both the electron and spin motions contribute the magnetic dipole moment of the atom although in most magnetic materials, the magnetic moment is due to spin motion. The magnetisation, and the related quantity magnetic dipole moment, are useful in understanding the

---

1 Throughout this work the magnitude of a vector quantity, $\vec{v}$, is represented by $v$. 
response of these materials to the field. The magnetic dipole moment is defined here for the simplest magnetic field creator circuit which is a circular loop of conductor carrying an electric current as,

\[ m = i A_c, \]  
\[ (2.3) \]

where \( A_c \) is the area of the electric current loop and \( i \) is the current in the circuit.

Magnetic flux density, \( \vec{B} \), results in a torque on the moment which can lead to the moment aligning with the magnetic flux density. Hence, the magnetic moment, \( \vec{m} \), can be defined as a vector relating the aligning torque on the magnetic dipole. The relationship is given by

\[ \vec{\tau} = \vec{m} \times \vec{B}, \]
\[ (2.4) \]

where \( \vec{\tau} \) is the torque on a magnetic dipole and so \( m \) can be determined from the maximum torque, \( \tau_{\text{max}} \), through

\[ m = \frac{\tau_{\text{max}}}{B}, \]
\[ (2.5) \]

and the unit of magnetic moment is ampere metres-squared (Am²).

Furthermore, the (volume) magnetisation, \( \vec{M} \), is defined as the sum of the magnetic dipole moment per unit volume of a solid via,

\[ \vec{M} = \frac{\vec{m}}{V}, \]
\[ (2.6) \]

where \( V \) is the sample volume and \( \vec{M} \) is measured in ampere per metre (A/m). Finally the relationship between the fundamental quantities \( \vec{H}, \vec{M} \) and \( \vec{B} \) for a linear material can be written as

\[ \vec{B} = \mu_0 \left( \vec{H} + \vec{M} \right), \]
\[ (2.7) \]

where \( \vec{B} \) is in tesla (T) and \( \vec{H} \) and \( \vec{M} \) are in amperes per metre (A/m). The reader should note the existence of other unit system such as the CGS and the Imperial unit systems [13]. Furthermore we note the fundamental equations can differ, depending on the unit system.
2.1.2 Permeability and Susceptibility

Permeability is an important distinguishing property of ferromagnets. It is the indication of the magnetic induction, $\vec{B}$, arising due to an applied magnetic field, $\vec{H}$. Whilst the permeability of a vacuum is constant, in general for magnetic material permeability is not constant but depends on the value of magnetic flux density, $\vec{B}$, for each $\vec{H}$ value as indicated by the hysteresis loop in figure 2.1. Consistent with (2.2) the magnetic permeability is defined through

$$\vec{B} = \mu \vec{H}. \quad (2.8)$$

A related concept, the relative permeability, denoted by $\mu_r$, also used in the SI unit system is defined as

$$\mu_r = \frac{\mu}{\mu_0}. \quad (2.9)$$

Magnetic susceptibility denoted by $\chi$, is closely related to the relative permeability through

$$\chi = \mu_r - 1 \quad (2.10)$$

and to $M$ and $H$ through

$$M = \chi H. \quad (2.11)$$

The magnetic susceptibility of a material can be positive or negative (unlike the analogous electric susceptibilities). The major types of magnetic material are classified as diamagnetic, paramagnetic or ferromagnetic according to their magnetic susceptibilities [23] where:

**diamagnetic** materials have small and negative susceptibilities because of the opposing nature of the magnetisation, $\vec{M}$, with respect to applied field, $\vec{H}$,

**paramagnetic** materials have small and positive susceptibilities because of their weak magnetisation, $\vec{M}$, in the applied field, $\vec{H}$, both of which are in the same direction,

**ferromagnetic** material whose magnetic susceptibilities are positive and large due to their strong magnetisation, $\vec{M}$.  

14
2.1.3 

**Hysteresis and Saturation Magnetisation**

The magnetic properties of ferromagnetic materials are commonly represented by a $B-H$ curve which is a plot of magnetic flux density, $B$, against magnetic field, $H$ or by a $M-H$ curve which is a plot of magnetisation, $M$, against magnetic field, $H$. In most cases, these curves involve hysteresis loops. Typical hysteresis loops are shown in figures 2.1 and 2.2 for ferromagnetic materials. The relationship between $H$ and $B$ is highly nonlinear and multivalued due to the presence of hysteresis and the hysteresis loop.

The hysteresis loop can be understood by first considering an unmagnetised sample. Thus in figures 2.1 and 2.2 it can be seen that initially the sample is not magnetised so that $M$ and $B$ are both zero for $H = 0$. In figure 2.2 as the magnetic field is increased the magnetisation increases and ultimately it reaches saturation. The magnetisation upper limit of a ferromagnetic material is called saturation magnetisation [70]. We note, that in figure 2.1 a saturation induction does not exist because $B$ continues to increase with $H$ owing to the non-zero permeability of free space.

![Figure 2.1: Schematic representation of the $B-H$ curve of a ferromagnetic material [45].](image-url)
2.1.4 Remanence and Coercivity

Ferromagnets can be magnetised in the presence of an applied magnetic field and once they are magnetised they can retain their magnetisation even if the applied magnetic field is removed. This magnetic property of ferromagnetic materials is called retentivity and distinguishes ferromagnets from paramagnets, as paramagnets do not retain their magnetisation after the applied field is removed.

Returning to figure 2.2, having reached saturation, if the magnetic field strength is now reduced the magnetisation decreases as indicated. Eventually when the magnetic field is reduced to zero, the term remanence is used to describe the remaining magnetisation and it is shown as $M_r$ in figure 2.2. In the $B$-$H$ curve if the magnetic field is removed after a certain time, the remaining magnetic induction is called the remanent magnetic induction, $B_r$ as in figure 2.1 and $B_r = \mu_0 M_r$.

In order to reduce the magnetisation, $M$, and (zero field) magnetic induction $B$ to zero, a reverse magnetic field of strength, called coercive field, $H_c$, should be applied to the ferromagnetic sample. This characteristic property of ferromagnetic material is known as coercivity.
2. Magnetic Fundamentals

2.1.5 Curie Temperature

Thermal properties of magnetic materials were first studied by Pierre Curie (1859–1906), who demonstrated that there is a temperature dependent relationship between magnetisation $\vec{M}$ and applied magnetic field $\vec{H}$, leading to the Curie law for susceptibility,

$$\chi = \frac{C}{T}, \quad (2.12)$$

where $T$ is the temperature in Kelvin and $C$ is the Curie constant. Curie also considered the effect of temperature of the magnetisation which a ferromagnetic sample exhibits in the absence of a field, termed the spontaneous magnetisation. He observed that the spontaneous magnetisation, $M_s$, decreases rapidly as the temperature approaches a critical value, known as the Curie temperature, $T_c$. At this point, thermal agitation energy overcomes the exchange forces, the spontaneous magnetisation disappears and the material loses its large magnetisation. Above the Curie temperature, ferromagnets behave essentially like paramagnetic materials [55]. Thus, at this critical point the permeability of the material drops suddenly and both coercivity and remanence become zero (see figure 2.3).

2.1.6 Hard and Soft Ferromagnetic Materials

The ferromagnetic materials are classified, according to their coercivity, as hard and soft magnetic materials, where the coercivity of soft magnetic materials is smaller than the coercivity of hard magnetic materials. The hysteresis loops of the hard magnetic materials are wider than the hysteresis of the soft magnetic materials. Therefore, more energy is required to magnetise the hard magnetic materials.

2.2 Paramagnetism and the Langevin Function

The theory of paramagnetism is simpler than that of ferromagnetism. Paramagnetic materials contain atoms each with a permanent magnetic dipole moment [56]. The simplest theory assumes a collection of such moments which do not interact with each other [45]. In the absence of an applied magnetic field, paramagnets do not retain any magnetisation.

and the orientation of individual magnetic moments point in random directions. When a magnetic field, $\vec{H}$ is applied there is partial alignment of the moments with the field. This is due to the thermal energy which is large enough to cause random disruption of the alignment of magnetic moments. In thermal equilibrium the average alignment of the moments was considered by Langevin leading to the Langevin function for the magnetisation given by [24],

$$\frac{M}{n_a m} = \coth \left( \frac{\mu_0 m H}{k_B T} \right) - \left( \frac{k_B T}{\mu_0 m H} \right),$$

where $n_a$ is the number of atoms per unit volume, $m$ is the magnetic moment per atom and $k_B$ is Boltzmann’s constant. This is consistent with the Curie law which indicates that it is more difficult to align a dipole at higher temperatures [42].

2.3 Single Domain Ferromagnetic Particles and Superparamagnetism

2.3.1 Weiss Domain Theory of Ferromagnetism

We have seen that the hysteresis curves in figures 2.1 and 2.2 can be described by the terms coercive field/coercivity and spontaneous magnetisation/saturation magnetisation for minor/major loops. The saturation magnetisation is a distinguishing property of a ferromagnetic material. It differs as the temperature of the material sample changes. Spontaneous magnetisation, $M_s$, of a ferromagnet is drawn as a function of the temperature $T$ in figure 2.3 where the temperature of the sample is normalised by the Curie temperature $T_c$. It is seen that all ferromagnetic materials behave like a paramagnetic material beyond the Curie temperature. Beyond this point the curve does not reduce to zero but reduces accordingly to the Curie law for paramagnets. Thus any theory which attempts explain ferromagnetism must explain both the existence of the hysteresis loop, and the reduction of this loop with increasing temperature with its disappearance above the Curie temperature. It was Weiss in 1907 who first explained both hysteresis and temperature dependence of the magnetisation in one theory. In his model, he assumed that there is an internal energy to align the dipoles of the atoms within regions called domains and that this allows an explanation for both the constant magnetisation below
the Curie temperature and the hysteresis loop [2].

![Figure 2.3: Spontaneous magnetisation, $M_s$, of a ferromagnet as a function of the temperature, $T$, normalised to the Curie Temperature, $T_c$. The applied field is assumed to be small, but finite, as it is in real measurements [2].](image)

Weiss wished to explain the unusual field dependence in figure 2.2. In his model, he assumed that ferromagnets are made up of many domains and each domain is magnetised to the saturation value as in figure 2.3 but these domains have different magnetisation direction. Thus, the value of the magnetisation is determined by the average of the magnetisation over these domains. It may be zero or non zero according to the direction of the domains. If the magnitude of the applied magnetic field is large enough to rotate all the domains in the direction of applied field then the average magnetisation becomes the saturation magnetisation, $M_{\text{sat}}$. This explains the unusual field dependence of the hysteresis loop in figure 2.2.

The existence of domains has been demonstrated by experimental work [2] and the origin of the molecular field is now known as an approximation to coupling forces between spins, termed exchange. Finally we note, whilst Weiss’s initial assumption was to allow random orientation of the domains, observation in many materials show the existence of a domain ordering or structure, which can be explained through micromagnetics [2, 19, 81].
2.3.2 Magnetic Anisotropy

The term anisotropy is used to describe situations where properties are dependent on direction. Thus, magnetic anisotropy is used when the magnetic properties of materials depend on the direction. A magnetically anisotropic material’s moment tends align to an easy axis which refers to the energetically favourable direction of the moment in the material. Magnetic anisotropy affects the shape of hysteresis loops and changes the values of coercivity and remanence. Hence, magnetic anisotropy is an important practical property in designing a magnetic material. There are different types of anisotropy depending on the crystal structure, shape of grains and applied or residual stresses.

2.3.2.1 Magnetocrystalline (Crystal Structure) Anisotropy

Magnetocrystalline anisotropy which is the most common anisotropy is caused by the spin magnetic moment and crystal lattice (spin-orbit coupling) interaction [2]. Crystals can be magnetised in some directions more easily than other directions.

Magnetocrystalline anisotropy energy is the energy which moves the magnetic moment in a single crystal from the direction of the hard axis. Although the magnetocrystalline energy has very small magnitude compared to the exchange energy, it is of importance in determining the direction of magnetisation. Both exchange energy and magnetocrystalline energy try to align all spins parallel to crystallographic direction, where specifically exchange energy tries to align all the spins parallel to each other and magnetocrystalline energy tries to align them in a definite crystallographic direction.

The anisotropy in hexagonal crystals which is referred as an uniaxial anisotropy is defined by the angle between the direction of the magnetisation and the easy axis. In most hexagonal crystals, the minimum magnetisation lies in the crystalline c-axis which is the easy axis. There are also some hexagonal crystals whose c-axis is in the hard axis where aligning the magnetisation along the c-axis is extremely difficult. Hexagonal crystals usually cannot easily reach their saturation where other (say cubic) crystals can. This is a key feature of hexagonal crystals.

Ferromagnets can shrink or expand in the direction of magnetisation whilst being magnetised, a phenomenon known as magnetostriction. Equally, by changing the shape of a ferromagnet, the value of magnetisation and dimensions of the domains can vary.
This form of anisotropy is also called stress anisotropy.

### 2.3.2.2 Shape Anisotropy

Another type of anisotropy considered is due to the shape of a mineral grain which is, in turn, due to magnetostatic properties. A magnetised material produces magnetic charges or poles at the surface. This surface charge distribution is another source of magnetic field. It is called the demagnetising field and acts in opposition to the magnetising field. For instance, consider a long thin needle shaped grain; the demagnetising field is weaker along the long axis than along the short axes. This produces an easy axis of magnetisation along the long axis.

Shape anisotropy is the most important form of anisotropy for smaller particles (< 20 µm) whereas it is less important than magnetocrystalline anisotropy for larger particles. Shape anisotropy is not important if the saturation magnetisation is low.

### 2.3.3 Single Domain Particles

We recall Weiss’s theory that ferromagnets are composed of domains. Within this theory each domain’s magnetisation reaches saturation but the direction of magnetisation differs from domain to domain. In an unmagnetised sample, all of these domains produce a net total magnetisation vector which is almost zero. In this model, the applied magnetic field can either alter the domain direction or through domain wall motion can increase the size of the domains in the direction of applied field. Both of these tend to increase the magnetisation.

Some magnetic properties of ferromagnets, like coercivity and remanence vary with grain size and the magnetic behaviour of ferromagnets can be subdivided on the basis of grain size into four ranges as:

- multiple domain (MD),
- single domain (SD), including superparamagnetic (SPM),
- pseudo-single domain (PSD).


A multiple domain (MD) sample contains many domains. The reason for this is that it reduces the magnetostatic energy associated with the surface charges. However, the domains must be separated by domain walls, that is small regions in which the moments have different directions. To be maintained, these walls also require energy, determined by the exchange and magnetocrystalline energies. Thus, for given sample size, balancing these energies a critical number of domain is reached. As predicted by Frenkel and Dorfman [32], if the size of the grain is reduced, a critical point is reached beyond which it can no longer provide a wall. It then contains a single domain that is uniformly magnetised [12]. The critical size for grains varies depending on the saturation magnetisation and the shape of grain (For magnetite, the critical size is about 80 nm). The magnetisation of an SD grain can be changed only by rotating the magnetisation, which can be energetically difficult process. Hence, single domain grains have high coercivity and remanence and thus they are magnetically hard materials. On the other hand, changing the magnetisation of a MD grain can be done by translating the domain wall, which requires a lower field. Hence some multiple domain grains can have lower coercivity and remanence, and these result in magnetically soft materials. Materials have their maximum coercivity within their single domain range, and coercivity decreases as the larger grain sizes subdivide into domains.

2.3.4 Pseudo-Single Domain

Typically SD and MD particles have different magnetic properties. Nevertheless, some MD particles have high remanence like SD particles and low coercivity like MD particles. This magnetic behaviour is called Pseudo-Single Domain (PSD). (For magnetite, this behaviour occurs in the size range between 0.1–20.0 µm in natural samples).

2.3.5 Superparamagnetism

As the grain size continues to decrease within the SD range, another critical point is reached where remanence and coercivity reduce to zero. The particle becomes super-paramagnetic (SPM) at this critical point [12]. A SD particle with the volume, $V$, has a uniform magnetisation along its easy axis. If $V$ is small enough or the temperature is high enough, thermal energy, $k_B T$, is sufficient to overcome the anisotropy energy. The
average of magnetic moment vector of a SPM particle in zero field and at $T > 0$ K is zero. However, in the presence of an applied field, there is a net alignment of magnetic moments. The resulting behaviour thus resembles paramagnetism. However, the much larger moments involved mean that superparamagnetism offers a much higher initial susceptibility value than simple paramagnetism. Néel’s treatment of this phenomenon resulted in an equation for a characteristic relaxation time, $t_r$. This was subsequently developed by Brown and others [18, 27] and can be expressed as,

$$\frac{1}{t_r} = f_0 \exp \left( \frac{-K_a V}{k_B T} \right)$$

(2.14)

where $f_0$ is the frequency pre-factor, (typically $10^9$ s$^{-1}$), $K_a$ is the anisotropy constant, $V$ is the particle volume. From this expression it is possible to define a blocking temperature, $T_B$ (at constant volume), or blocking volume $V_B$, (at constant temperature) at which the magnetisation goes from an unstable (SPM) condition to a stable condition. Furthermore, from (2.14), taking the standard benchmark for superparamagnetism to be zero remanence after 100 s we can obtain the approximate condition for superparamagnetism to be $K_a V < 25 k_B T$.

### 2.3.6 Hysteresis Properties of Different Size Particles

The shape of a hysteresis loop is determined by the domain state. Hysteresis loops of SD particles are wider than loops for MD materials because of the higher coercivity and remanence in SD material. Thus, the hysteresis loop parameters are useful in distinguishing domain state.

For SD particles, remanent magnetisation, $M_r$, can be calculated and depends on the type of anisotropy. On the other hand for MD or PSD particles, experimental results are used for the hysteresis loop because of the difficulty of theoretical prediction and thus calculation of $M_r$ and $H_c$ ratios.

For SPM particles, the shape of the hysteresis loop is extremely thin because of the very low remanence and coercivity. In the presence of an applied field, SPM particles have a steep initial rise in magnetisation followed by a gradual increase to saturation as described by the Langevin function. We note that SPM and MD particles can have the
similar hysteresis properties (low hysteresis) at room temperature, but cooling the sample down to very low temperatures can help distinguish between samples.

2.4 Nanoparticles

A nanoparticle is a particle whose size is of the order of nanometres. Strictly, it can be defined as a particle with at least one dimension is in nanometre range which is smaller than 200 nm. In practice, their sizes range from 10 nm to 1000 nm.

The properties of materials can change significantly as the size of the sample reduces. In particular on the nanoscale, the increased surface area to volume ratio is significant. This can alter the optical, electrical, or magnetic properties and also affect the mechanical properties, such as flexibility or elasticity in materials.

2.5 Magnetic Nanoparticles

Magnetic nanoparticles are magnetic systems whose dimensions are on the nanometre range. They show many new features such as slow relaxation at low temperatures accompanied by hysteretic magnetisation with high coercivity. Also nanoparticles become SPM beyond the blocking temperatures.

2.5.1 Other Applications of Magnetic Nanoparticles in Biomedicine

Magnetic nanoparticles can be used in numerous fields, including MDT which is the subject of this thesis but also in magnetic resonance imaging (MRI) and magnetic fluid hyperthermia (MFH) treatment [65].

2.5.1.1 Magnetic Resonance Imaging Contrast Agents for Monitoring Drug Delivery

Magnetic resonance imaging (MRI) can provide detailed images of the structure and functioning of the body. MRI produces images of all organs and this is useful in analysis and during the course of therapy [59, 74]. In contrast enhancement, human injections
of agents, such as Combidex, work as the metastatic lymph nodes absorb the particles more than inflamed nodes and this is detectable with MRI [59]. MRI is mainly used for imaging the brain and cancer cells.

MRI can be used in conjunction with MDT where real time imaging can monitor the \textit{in vivo} distribution of the nanoparticles. The contrast agents should have the same \textit{in vivo} localisation as the beneficial nanoparticles because of their similar size and charge. Research is continuing in imaging the \textit{in vivo} distribution of the contrast agents by MRI to optimise the size and surface properties for targeting tumours.

\subsection*{2.5.1.2 Magnetic Fluid Hyperthermia}

Magnetic Fluid Hyperthermia (MFH) is a promising cancer treatment that uses magnetic nanoparticles to heat the cancerous tissue to appropriate temperatures. This can be achieved by localising the magnetic nanoparticles in the cancerous tissue through applying an external magnetic field to the desired tissue. The temperature rise in the tissue during MFH depends on the structure of the particles, quantity of the particles and, the amplitude and frequency of the magnetic field. MFH can be performed in conjunction with radiotherapy.
Chapter 3

The Basic Mathematical Model

3.1 Introduction

In this work, we develop a number of related IA-MDT models and we begin with outlining the basic mathematical model used. This model is based on that of Ritter et al. [66], and the presentation here reflects this.

Since different types implants (e.g. seeds, stents) are suitable for different regions of the vascular system (e.g. capillary beds, arteries), we need to consider these different physical domains. The physical domains used in this work and accompanying boundary and initial conditions are discussed in section 3.2. In this basic mathematical model, we consider no interactions; in effect we simply treat the dynamics of a single MDCP. The forces governing its dynamics are discussed in section 3.3. The outline of the model concludes with the performance metrics of MDCP capture.

3.2 Physical Domains

In this work, we use two different 2D models, which are the seed model of Avilés et al. [6] (see figure 3.1) and the stent model of Avilés et al. [8] (see figure 3.2). In these models we consider the effect of these magnetisable implants placed in the blood flow as indicated in figures 3.1 and 3.2. In the 2D model both these can be represented ultimately in terms of a circular implant.

3.2.1 The Capillary Bed used in the Seed Model

A capillary bed is a dense network of tiny blood vessels. A simple and effective approach to modelling the flow here is to treat the region as homogeneous. Embedded in this region is a spherical seed and the boundaries of the regions are assumed to be far from the seed.
By considering a slice through the centre of the seed, this domain can effectively reduce to 2D. We point out that this in fact corresponds to flow in a rectangular box with a cylindrical wire, both of infinite extent.

In this context the natural velocity profile is uniform, thus, a uniform inlet velocity profile is assumed at the inlet control volume (CV) in Cartesian coordinate this can be expressed as

\[ \vec{v}_b = \begin{pmatrix} u_0 \\ 0 \end{pmatrix}, \] (3.1)

where \( \vec{v}_b \) is the blood velocity, \( u_0 \) is the inlet blood velocity. Non-slip boundary conditions are applied at the seed-blood interface. In addition, symmetry boundary conditions are applied at the upper and lower CV boundaries to maintain the constant flow profile. Atmospheric pressure is assumed at the outlet of the CV to satisfy the boundary condition on pressure.
3.2.2 The Single Vessel used in the Stent Model

The stent model is based on a ferromagnetic coiled wire stent placed next to the walls of cylindrical vessel (tube). In order to reduce this 3D geometry to 2D, a slice is taken through the centre of the vessel. Thus, the coiled stent is modelled as a series of circular cross sections of infinitely long wires with radii of $R_{\text{wire}}$ located at the upper and lower boundaries of the walls offset from each other and with centres separated by a distance, $h$.

For the single vessel used in the stent model, a parabolic velocity profile is assumed at the inlet CV such that

$$
\vec{v}_b = \begin{pmatrix} 1.5 u_0 \left(1 - \left(\frac{y}{R_{\text{vessel}}}\right)^2\right) \\ 0 \end{pmatrix},
$$

where $u_0$ is the average inlet blood velocity and $R_{\text{vessel}}$ is the vessel radius. Non-slip boundary conditions ($\vec{v}_b = 0$) are applied at the stent-blood interface and at the upper and lower CV boundaries. Atmospheric pressure is assumed at the outlet of the CV to satisfy the boundary condition on pressure.
3.3 Derivation of Particle Velocity

In this section, we derive the total velocity of a MDCP resulting from the forces which act on it. The force balance on a MDCP can be written as

\[ \vec{F}_s + \vec{F}_m = \vec{F}_i \]  

(3.3)

where \( \vec{F}_s \), \( \vec{F}_m \), and \( \vec{F}_i \) are the Stokes drag, magnetic and inertial forces, respectively.

Firstly, the Stokes drag is given by

\[ \vec{F}_s = 6\pi \eta_b R_p (\vec{v}_b - \vec{v}_p), \]  

(3.4)

where \( \eta_b \) is the viscosity of the blood, \( R_p \) the radius of the MDCP, and \( \vec{v}_b \) and \( \vec{v}_p \) are the velocities of the blood and the MDCP respectively. The blood velocity, \( \vec{v}_b \), is determined by solving the appropriate Navier-Stokes equations as in section 3.4. Secondly, the magnetic force is determined by

\[ \vec{F}_m = (\vec{m} \cdot \nabla) \vec{B}, \]  

(3.5)

where \( \vec{B} \) is the magnetic flux density due to the externally applied magnetic field, \( \vec{H}_0 \), and the presence of the circular implant (seed and stent), and \( \vec{m} \) is the magnetic moment of the MDCP.

Neglecting the inertial forces, the MDCPs are under the influence of Stokes drag and magnetic force as given in (3.4) and (3.5) respectively so that

\[ 6\pi \eta_b R_p (\vec{v}_b - \vec{v}_p) + (\vec{m} \cdot \nabla) \vec{B} = 0. \]  

(3.6)

The velocity of a MDCP, \( \vec{v}_p \), can be obtained from (3.6). Hence, we obtain

\[ \vec{v}_p = \vec{v}_b + \frac{1}{6\pi \eta_b R_p} (\vec{m} \cdot \nabla) \vec{B}. \]  

(3.7)

where, \( \vec{B} \) is the total magnetic flux density acting on the MDCPs. The magnetic flux
density, \( \vec{B} \) is given by
\[
\vec{B} = \mu_0 \vec{H},
\] (3.8)
and in the space around the circular implant \( \vec{H} \) is given by,
\[
\vec{H} = \vec{H}_0 - \nabla \phi.
\] (3.9)
where \( \phi \) is the magnetic scalar potential. In this work, an analytic solution of magnetic scalar potential is derived and this derivation is outlined in section 4.3.1.

Considering field orientation, given by \( \varphi \), as in figures 3.1 & 3.2, the magnitude of the total magnetic field strength can be written in Cartesian coordinates as
\[
H = \sqrt{ \left( H_0 \cos \varphi - \frac{\partial \phi}{\partial x} \right)^2 + \left( H_0 \sin \varphi - \frac{\partial \phi}{\partial y} \right)^2 }.
\] (3.10)

### 3.4 Calculation of Blood Velocity

In this section, calculation of the blood velocity is shown following Avilés et al. [6, 8]. The blood is treated as an incompressible, Newtonian, isothermal, single-phase fluid with velocity, \( \vec{v}_b \), and pressure \( P \) for steady state flow. The Navier-Stokes equations consist of the continuity equation
\[
\nabla \cdot \vec{v}_b = 0,
\] (3.11)
and
\[
\rho_b [(\vec{v}_b \cdot \nabla \vec{v}_b)] = -\nabla P + \eta_b \nabla^2 \vec{v}_b,
\] (3.12)
where \( \rho_b \) is the density of the blood.

#### 3.4.1 Introduction to Navier-Stokes Equations

The Navier-Stokes equations are the fundamental partial differential equations that describe the flow of fluids. Before outlining the Navier-Stokes equations, we define some fundamental concepts and then express the Navier-Stokes equations for 2D fluid flow using the Cartesian coordinate system.
• **Viscosity** Viscosity is the measure of the resistance of a liquid to flow. If a fluid is flowing over a surface, the molecules next to the surface have zero speed. As we get further away from the surface, the speed of the molecules increases. The friction of the liquid is due to the difference in speed of the molecules. Viscosity determines the amount of friction and thus the amount of energy absorbed by the flow.

• **Laminar Flow** Laminar flow (streamline flow) occurs when a fluid flows in parallel layers. Fluid elements or particles appear to slide over each other in layers. Although there is molecular agitation and diffusion, there is no large scale mixing between the layers.

• **Incompressible Flow** Incompressible flows are those for which the density of fluid is constant on particle paths. For an incompressible flow the divergence of fluid velocity is zero.

### 3.4.2 Navier-Stokes Equations in Dimensionless Form

Here we begin with the form of the Navier-Stokes equations as given in the wire implant model of Ritter et al. [66], and show how this can be rewritten in the form used in the subsequent seed and stent models [6, 8]. The Cartesian coordinate system is used with the following assumptions: isothermal behaviour, incompressible Newtonian fluid, and single phase flow. After dimensionless analysis, the Navier-Stokes equations are obtained with the following dimensionless variables,

\[
\tilde{x} = \frac{x}{R_{\text{wire}}}, \quad \tilde{y} = \frac{y}{R_{\text{wire}}}, \quad \tilde{v}_{b,x} = \frac{v_{b,x}}{u_0}, \quad \tilde{v}_{b,y} = \frac{v_{b,y}}{u_0}, \quad \tilde{P} = \frac{P}{P_0}.
\]

(3.13)

where \( \tilde{v}_b \) is the blood velocity, \( u_0 \) is the average inlet velocity, \( \tilde{v}_b \) is the scaled blood velocity, \( R_{\text{wire}} \) is the radius of the wire implant, \( P \) is the blood pressure, \( P_0 \) is the blood pressure at the outlet of the CV and \( \tilde{P} \) is the scaled blood pressure. By using these variables, the continuity and 2D Navier-Stokes equations are written as [66],

\[
\frac{\partial \tilde{v}_{b,x}}{\partial \tilde{x}} + \frac{\partial \tilde{v}_{b,y}}{\partial \tilde{y}} = 0
\]

(3.14)
3. The Basic Mathematical Model

3.4. Calculation of Blood Velocity

and

\[
- \frac{2N_{Eu}}{N_{Re}} \left( \frac{\partial^2 \tilde{v}_{b,x}}{\partial x^2} + \frac{\partial^2 \tilde{v}_{b,x}}{\partial y^2} \right) + N_{Eu} \left( \tilde{v}_{b,x} \frac{\partial \tilde{v}_{b,x}}{\partial x} + \tilde{v}_{b,y} \frac{\partial \tilde{v}_{b,x}}{\partial y} \right) + \frac{\partial \tilde{P}}{\partial \tilde{x}} = 0 \tag{3.15}
\]

\[
- \frac{2N_{Eu}}{N_{Re}} \left( \frac{\partial^2 \tilde{v}_{b,y}}{\partial x^2} + \frac{\partial^2 \tilde{v}_{b,y}}{\partial y^2} \right) + N_{Eu} \left( \tilde{v}_{b,x} \frac{\partial \tilde{v}_{b,y}}{\partial x} + \tilde{v}_{b,y} \frac{\partial \tilde{v}_{b,y}}{\partial y} \right) + \frac{\partial \tilde{P}}{\partial \tilde{y}} = 0 \tag{3.16}
\]

where \(N_{Eu}\) and \(N_{Re}\) are Euler and Reynolds numbers defined by

\[
N_{Eu} = \frac{\rho_b u_0^2}{P_0}, \quad N_{Re} = \frac{2\rho_b u_0 R_{wire}}{\eta_b}. \tag{3.17}
\]

At the outlet of the vessel, the blood pressure boundary condition is

\[
\tilde{P} = 1. \tag{3.18}
\]

Also, a non-slip boundary condition \(\tilde{v}_b = 0\) is applied to every interface in contact with the bloodstream. Equations (3.14), (3.15) and (3.16) can be rewritten in vector notation and dropping the tildas we obtain the form of Avilés et al. \[6,8\] given below

\[
\nabla \cdot \vec{v}_b = 0, \tag{3.19}
\]

and

\[
- \frac{2N_{Eu}}{N_{Re}} \nabla^2 \vec{v}_b + N_{Eu}(\vec{v}_b \cdot \nabla \vec{v}_b) + \nabla \tilde{P} = 0. \tag{3.20}
\]

On rearranging the above equations we obtain

\[
\nabla \cdot \vec{v}_b = 0, \tag{3.21}
\]

and

\[
N_{Eu}(\vec{v}_b \cdot \nabla \vec{v}_b) = -\nabla \tilde{P} + \frac{2N_{Eu}}{N_{Re}} \nabla^2 \vec{v}_b. \tag{3.22}
\]

This is the form of the Navier-Stokes equations used in the seed and stent models.
3.5 Derivation of Streamlines, Capture Cross Section and Collection Efficiency

Finally, the MDCP trajectories are obtained from evaluating the streamline function

$$\frac{\partial \psi_s}{\partial y} = -v_{p,x},$$  \hfill (3.23)

$$\frac{\partial \psi_s}{\partial x} = v_{p,y},$$  \hfill (3.24)

where $\psi_s$ is the stream function, and $v_{p,x}$ and $v_{p,y}$ are the components of $\vec{v}_p$ from (3.7).

The system performance of this model is calculated in terms of the capture cross section (CCS), $\lambda_c$, defined as

$$\lambda_c = \frac{y_c}{R_{\text{implant}}},$$  \hfill (3.25)

where $R_{\text{implant}}$ is the radius of the implant and $y_c$ is the capture radius of the ferromagnetic implant. The capture radius, $y_c$, is defined by the location of the streamline at the entrance to the CV of the last MDCP captured to the implant (see figure 3.1).

The system performance of the stent-based mathematical model is calculated in terms of collection efficiency, CE, defined as

$$\text{CE} = \frac{2 R_{\text{vessel}} - y_1 + y_2}{2 R_{\text{vessel}}} \times 100\%,$$  \hfill (3.26)

where $R_{\text{vessel}}$ is the radius of the vessel and $y_1$ and $y_2$ are defined by the location of the streamline at the entrance to the CV of the last MDCPs captured to the stent wires (see figure 3.2).
Chapter 4

Implementation in OpenFOAM

In this chapter, we outline the open-source, finite volume library OpenFOAM. OpenFOAM stands for Open Field Operations And Manipulation and in section 4.1, general information is given followed by the treatment of Navier-Stokes equations in OpenFOAM. Finally the calculation of magnetic flux density, $\vec{B}$, is implemented using an analytic solution for the magnetic scalar potential, the derivation of which is presented.

4.1 Introduction to OpenFOAM

OpenFOAM is a C++ toolbox consisting of pre-written numerical solvers for Computational Continuum Mechanics (CCM) and Computational Fluid Dynamics (CFD) problems and an extensible class library to allow development of new models. CFD is a branch of CCM and covers compressible, incompressible, multiphase and free surface flows as well as flows involving further physics such as chemical reactions and electromagnetic effects [82]. These can be combined to create solvers and utilities, or additional functionality can be introduced through new libraries or new modules. The library provides Finite Volume and Finite Element methods in operator form and with polyhedral mesh support. Structural analysis is treated by the Finite Element Method (FEM), while fluid flow is handled using the Finite Volume Method (FVM).

OpenFOAM allows the user to employ third party pre- and post-processing utilities, such as paraFOAM, for visualisation of solution data and meshes. OpenFOAM itself provides an efficient solution framework, including geometry handling, mesh generation, solution, post processing and data analysis while implementing a large number of numerical and physical models. While OpenFOAM has its own mesh generator, it also allows the importing of a wide range of mesh converters from a number of leading commercial packages [44].

OpenFOAM is produced by the UK company, OpenCFD Ltd. and is released open
source under General Public License. OpenFOAM software source code is freely available and it permits users to study, change, and improve the code through the user’s own modification. Its development began in the late 1980s at Imperial College, London, in efforts to find a more powerful and flexible general simulation platform than Fortran. Since then it has used the latest advanced features of the C++ language, and it has been re-written several times. OpenFOAM is designed to make it as easy as possible to develop reliable and efficient CCM codes, by making the syntax of the code closer to standard mathematical notation.

OpenFOAM has been pioneering in a number of ways. Readable descriptions of partial differential equations make it an understandable programming language for physical simulations and it is the first major general-purpose CFD package to use polyhedral cells [64]. It is also the most capable general purpose CFD package which is released under an open-source licence. OpenFOAM is designed for CCM problems but it is easy to generate multi-physics simulations as well. Standard Solvers have been developed for problems in a number of areas including [64]:

- Basic CFD
- Incompressible flows
- Compressible flows
- Multiphase flows
- Combustion
- Heat transfer
- Electromagnetic
- Solid dynamics
- Finance

One of the key features of OpenFOAM is that solver applications can be created easily. OpenFOAM uses syntax that closely resembles the standard mathematical descriptions
of differential equations. For example, the equation \ref{eq:navier_stokes}

\[
\frac{\partial \rho \vec{v}}{\partial t} + \nabla \cdot K \vec{v} - \nabla^2 \eta \vec{v} = -\nabla P
\]  

(4.1)

is represented by the code block

```cpp
solve (  
    fvm::ddt(rho, v)  
    =  
    + fvm::div(K, v)  
    - fvm::laplacian(eta, v)  
    =  
    - fvc::grad(P)  
);
```

where \( \rho \) is the density, \( \eta \) is the viscosity, \( P \) is the pressure and \( \vec{v} \) is the velocity of fluid. (Elsewhere in OpenFOAM, \( \rho \), \( \eta \), \( P \) and \( K \) are defined as scalar quantities and \( v \) as a vector quantity.)

Since 1980 considerable effort has been directed towards development of OpenFOAM as a scientific numerical modelling package \cite{16, 17, 39, 40, 43, 47, 48, 54, 67, 69, 78, 79, 80}. In these publications, OpenFOAM is compared with other CFD packages. OpenFOAM results are almost identical to those of the CFX-5 CFD code and show the same trend as the results using the Fluent CFD code. OpenFOAM also gives similar results as the CALC-PMB in-house CFD code that was developed specifically for water turbine applications \cite{61}.

### 4.2 Navier-Stokes Equations in OpenFOAM

In OpenFOAM different types of flows can be described by systems of linked partial differential equations of the form

\[
\frac{\partial \rho \vec{Q}}{\partial t} + \nabla \cdot (\rho \vec{v} \otimes \vec{Q}) - \nabla \cdot \rho D \nabla \vec{Q} = S_p \vec{Q} + S_q,  
\]  

(4.2)

where $\vec{Q}$ is any tensor valued property of the flow. These equations involve time derivatives, $\partial(\rho\vec{Q})/\partial t$, convective terms, $\nabla \cdot (\rho\vec{v} \otimes \vec{Q})$, diffusive terms, $\nabla \cdot \rho D \nabla \vec{Q}$ and source terms, $S_p\vec{Q}$ and $S_q$. A simple example is that of incompressible flow as described by the Navier-Stokes equations. Navier-Stokes equations representing incompressible flow can be written by substituting $\vec{Q} = 1$ in (4.2) to get continuity equation

$$\nabla \cdot \vec{v} = 0,$$

and by substituting $\vec{Q} = \vec{v}$ in (4.2) we get

$$\frac{\partial \vec{v}}{\partial t} + \nabla \cdot (\vec{v} \otimes \vec{v}) - \nabla \cdot 2\eta_{\text{eff}} \bar{D} = -\frac{1}{\rho} \nabla P,$$

where

$$\bar{D} = \frac{1}{2} + (\nabla \vec{v} + (\nabla \vec{v})^T),$$

where $\eta_{\text{eff}}$ is the kinematic viscosity.

In OpenFOAM to solve (3.11) and (3.12), we use the SimpleFOAM solver which is a steady state solver for incompressible, laminar and turbulent flow of Newtonian fluids. Before solving our equations we explain the SimpleFOAM solver which is specifically designed for solving the system,

$$\nabla \cdot \vec{v} = 0,$$

$$- \eta_{\text{eff}} \nabla^2 \vec{v} + \nabla \cdot (\vec{v} \otimes \vec{v}) + \nabla P = 0,$$

where $\eta_{\text{eff}}$ is the kinematic viscosity of the fluid. The Navier-Stokes equations as formulated in (3.11) and (3.12) can be readily solved by the SimpleFOAM solver upon specifying the value of $\eta/\rho$ for $\eta_{\text{eff}}$. For calculating the blood velocity in SimpleFOAM, $\eta_{\text{eff}}$ is specified as $\eta_b/\rho_b$. 

37
4.3 Calculation of Magnetic Force

In this section, we outline the derivation of magnetic scalar potential and its implementation in OpenFOAM. The magnetic force acting on a MDCP is determined by

$$\vec{F}_m = (\vec{m} \cdot \nabla) \vec{B},$$  \hspace{1cm} (4.8)

where $\vec{B}$ is the magnetic flux density due to the externally applied magnetic field, $\vec{H}_0$, and the presence of the implant, and $\vec{m}$ is the magnetic moment of the particle. The magnetic flux density, $\vec{B}$, relates to the total magnetic field, $\vec{H}$, via

$$\vec{B} = \mu \vec{H},$$  \hspace{1cm} (4.9)

where $\mu$ is the permeability of the medium as

$$\mu = \mu_r \mu_0,$$  \hspace{1cm} (4.10)

and $\vec{H}$ is given by,

$$\vec{H} = \vec{H}_0 - \nabla \phi.$$  \hspace{1cm} (4.11)

where $\phi$ is the magnetic scalar potential.

4.3.1 Calculation of Magnetic Scalar Potential

We consider the general problem of determining the magnetic scalar potential over multiple regions where within each region the permeability is constant (see figure 4.1). This is resolved using the Biot-Savart law \[ 76 \], which requires the normal component of $\vec{B}$ and the tangential component of $\vec{H}$ to be continuous across the interface (see figure 4.2). That is

$$\hat{n} \cdot \vec{B}_1 = \hat{n} \cdot \vec{B}_2,$$  \hspace{1cm} (4.12)

and

$$\hat{t} \cdot \vec{H}_1 = \hat{t} \cdot \vec{H}_2,$$  \hspace{1cm} (4.13)

Technically, $\phi$ is actually the reduced magnetic scalar potential rather than the total magnetic scalar potential. The reason for our use of the reduced magnetic scalar potential is discussed in appendix A.
4. Implementation in OpenFOAM 4.3. Calculation of Magnetic Force

\[ \nabla^2 \phi_1 = \mu_1 \nabla \cdot \vec{H}_0 \]
\[ \mu_1 = \text{constant} \]

\[ \nabla^2 \phi_2 = \mu_2 \nabla \cdot \vec{H}_0 \]
\[ \mu_2 = \text{constant} \]

**Region 1**

\[ \nabla^2 \phi_1 = \mu_1 \nabla \cdot \vec{H}_0 \]
\[ \mu_1 = \text{constant} \]

**Region 2**

\[ \nabla^2 \phi_2 = \mu_2 \nabla \cdot \vec{H}_0 \]
\[ \mu_2 = \text{constant} \]

![Diagram](image)

**Figure 4.1:** General problem of object (region 1 = implant) embedded in a space region (region 2 = space) of constant permeability [76].

where \( \hat{n} \) is the normal vector and \( \hat{t} \) is the tangential vector of the interface.

To distinguish between the two regions, we use the notation listed below,

- \( j \) Region index — implant \( j = 1 \), space \( j = 2 \) (subscript is dropped when region independent),
- \( \phi_j \) Magnetic scalar potential in region \( j \),
- \( \vec{H}_j \) Resulting magnetic field due to external magnet and implant in region \( j \),
- \( a_j \) and \( b_j \) are the constants to be determined in region \( j \).

### 4.3.1.1 2D models: Circular Implant and Polar Coordinates

All the implants considered whether spherical seeds, cylindrical wires or coiled stents can be described in two dimensions in terms of circular implants as discussed in section 3.2.

Considering the resulting (or in the case of a stent, any one of the resulting) circular implants the physical domain is more naturally represented in terms of polar coordinates \((r, \theta)\) with origin coincident with the implant centre. In polar coordinates the problem of determining the magnetic scalar potential reduces to a standard separation of variables problem as outlined in the following sections.

In polar coordinates the differential operator is

\[ \nabla = \frac{\partial}{\partial r} \hat{e}_r + \frac{1}{r} \frac{\partial}{\partial \theta} \hat{e}_\theta. \]  \( (4.14) \)
4. Implementation in OpenFOAM

4.3. Calculation of Magnetic Force

Hence

\[ \nabla \cdot \vec{v} = \frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta}, \]  

(4.15)

and

\[ \nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}. \]  

(4.16)

4.3.1.2 Background Source Field

In order to simplify the derivation of magnetic force, we consider the background source field magnitude of \( H_0 \) parallel to the \( x \)-axis. Hence, we have \( \vec{H}_0 = H_0 \hat{e}_x \).

4.3.1.3 Analytic Solution of Magnetic Scalar Potential

The magnetic field, the magnetic flux density and the magnetic scalar potential in region 1 (implant) are related by

\[ \vec{H}_1 = \vec{H}_0 - \nabla \phi_1, \quad \vec{B}_1 = \mu_1 \vec{H}_1, \quad \nabla^2 \phi_1 = \mu_1 \nabla \cdot \vec{H}_0. \]  

(4.17)

and in region 2 (space)

\[ \vec{H}_2 = \vec{H}_0 - \nabla \phi_2, \quad \vec{B}_2 = \mu_2 \vec{H}_2, \quad \nabla^2 \phi_2 = \mu_2 \nabla \cdot \vec{H}_0. \]  

(4.18)

The normal component of the magnetic flux density and the tangential component of magnetic field are both assumed to be continuous across the implant-space interface. These are linked through the interface conditions (figure 4.2)

\[ \hat{n} \cdot \vec{B}_1 = \hat{n} \cdot \vec{B}_2 \implies \mu_1 \frac{\partial \phi_1}{\partial r} = \mu_2 \frac{\partial \phi_2}{\partial r} + (\mu_1 - \mu_2) \vec{H}_0 \cdot \hat{n}, \]  

(4.19)

and

\[ \hat{t} \cdot \vec{H}_1 = \hat{t} \cdot \vec{H}_2 \implies \frac{\partial \phi_1}{\partial \theta} = \frac{\partial \phi_2}{\partial \theta}, \]  

(4.20)

and satisfy the boundary condition

\[ \vec{H}_1 \text{ bounded}, \]  

(4.21)
\( \hat{n} \cdot \vec{B}_1 = \mu_1 \left( \frac{\partial \phi_1}{\partial r} - \hat{n} \cdot \vec{H}_0 \right) \)

\( \mu_2 \left( \frac{\partial \phi_2}{\partial r} - \hat{n} \cdot \vec{H}_0 \right) = \hat{n} \cdot \vec{B}_2 \)

**Figure 4.2:** Interface conditions across the boundary of regions with different permeability.— the normal component of \( \vec{B} \) and the tangential component of \( \vec{H} \) are continuous [76].

\[
\hat{t} \cdot \vec{H}_1 = \frac{\partial \phi_1}{\partial \theta} \quad \frac{\partial \phi_2}{\partial \theta} = \hat{t} \cdot \vec{H}_2
\]

\[
\hat{n} \cdot \vec{B}_1 = \mu_1 \left( \frac{\partial \phi_1}{\partial r} - \hat{n} \cdot \vec{H}_0 \right)
\mu_2 \left( \frac{\partial \phi_2}{\partial r} - \hat{n} \cdot \vec{H}_0 \right) = \hat{n} \cdot \vec{B}_2
\]

and

\[
\vec{H}_2 \rightarrow \vec{H}_0 \text{ as } r \rightarrow \infty \implies \phi_2 \rightarrow 0 \text{ as } r \rightarrow \infty.
\]

(4.22)

Here, we need the finite version of this condition \( \vec{H}_2 \rightarrow \vec{H}_0 \text{ as } r \rightarrow r_\infty \) and \( \vec{H}_1 \) bounded.

The Poisson equations for the scalar potentials both have solution

\[
\phi_j = \left( a_j r + b_j r^{-1} \right) \cos \theta,
\]

(4.23)

for some undetermined coefficients \( a_j \) and \( b_j \). Hence, for region 1 (implant) we have

\[
\phi_1 = \left( a_1 r + b_1 r^{-1} \right) \cos \theta,
\]

(4.24)

and for region 2 (space)

\[
\phi_2 = \left( a_2 r + b_2 r^{-1} \right) \cos \theta.
\]

(4.25)

Far from the implant, the magnetic scalar potential should tend towards zero and applying the boundary conditions we have

\[
\lim_{r \rightarrow r_\infty} \phi_2 = 0 \implies \lim_{r \rightarrow r_\infty} \left( a_2 r + b_2 r^{-1} \right) \cos \theta = 0,
\]

(4.26)

and \( a_2 = 0 \). Furthermore the solution in both regions must be bounded, hence \( b_1 = 0 \).
Using the interface condition (4.12), we rewrite (4.19) at \( r = 1 \),

\[
- \mu_1(a_1 - b_1) \cos \theta + \mu_1 H_0 \cos \theta = -\mu_2(a_2 - b_2) \cos \theta + \mu_2 H_0 \cos \theta,
\]

(4.27)

and on applying \( a_2 = 0 \) and \( b_1 = 0 \), this reduces to

\[- \mu_1 a_1 + \mu_1 H_0 = \mu_2 b_2 + \mu_2 H_0.\]

(4.28)

The interface condition (4.13) implies that

\[
\frac{\partial \phi_1}{\partial \theta} = \frac{\partial \phi_2}{\partial \theta} \implies (a_1 + b_1) \cos \theta = (a_2 + b_2) \cos \theta,
\]

(4.29)

and on applying \( a_2 = 0 \) and \( b_1 = 0 \), the condition \( a_1 = b_2 \) is obtained. Equation (4.28) can now be expressed in terms of \( a_1 \) only as follows

\[- \mu_1 a_1 + \mu_1 H_0 = \mu_2 a_1 + \mu_2 H_0,\]

(4.30)

and so

\[a_1 = b_2 = \left( \frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0.\]

(4.31)

Hence, the scalar potential for regions 1 (implant) is

\[\phi_1 = a_1 r \cos \theta = \left( \frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0 r \cos \theta,\]

(4.32)

and for region 2 (space) is

\[\phi_2 = b_2 r^{-1} \cos \theta = \left( \frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0 r^{-1} \cos \theta.\]

(4.33)

We can rewrite these potentials in Cartesian coordinates as follows:

\[\phi_1 = \left( \frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0 x,\]

(4.34)

and

\[\phi_2 = \left( \frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0 \frac{x}{x^2 + y^2}.\]

(4.35)
This result is readily generalised to treat applied fields with an arbitrary field direction, \( \varphi \), to obtain for the scalar potential in the region outside the implant

\[
\phi = H_0 R_{\text{implant}}^2 \frac{\mu_{\text{implant}} - 1}{\mu_{\text{implant}} + 1} \frac{x \cos \varphi + y \sin \varphi}{x^2 + y^2},
\]

where \( R_{\text{implant}} \) is the radius of the implant (seed, stent wire), \( x \) and \( y \) are the coordinates measured from the centre of the implant and \( \mu_{\text{implant}} \) is the relative permeability of the ferromagnetic implant.

In the stent model, the overall magnetic scalar potential in the space due to the stent is calculated through the sum of the individual magnetic scalar potentials of each stent wires. This analytic solution of magnetic scalar potential is implemented in OpenFOAM directly and the magnetic field is calculated through the numerical gradient of the magnetic scalar potential.
Chapter 5

Development of Model to Include Interactions and Results

In this work, we try to develop more realistic models of IA-MDT. Firstly when single domain magnetic nanoparticles (radius in the range 20-100 nm) are used as the MDCPs, the Langevin function is used to describe the magnetisation of the MDCPs. The results of our simulations for the seed model indicate that use of the Langevin function predicts greater collection efficiency (CE) than might be otherwise expected. The results of this work have been presented in the *Journal of Magnetism and Magnetic Materials* [28].

Secondly, with a view to modelling experimentally observed agglomeration in IA-MTD [6, 8, 9, 66], we adapt and extend the current approaches to model two mutually interacting MDCPs with larger field strength and a seed implant. The effect of the dipole-dipole and hydrodynamic interactions between two MDCPs on the calculated magnetic force in the IA-MDT system of Avilés et al. [6] is considered. In these simulations, depending on the initial configuration of the MDCPs, both increases and decreases of up to 7% in absolute terms, can be observed in the CCS of the model. The results of this work have been presented in the second paper accepted by the *Journal of Magnetism and Magnetic Materials* [29].

We extend these approaches to model dipole-dipole and hydrodynamic interactions for multiple MDCPs in further implant arrangements. In particular we model the stent arrangement proposed and studied in Avilés et al. [8, 9], where multiple particle agglomeration can be expected to contribute significantly to increase in the capture of MDCPs (containing single domain nanoparticles) reported therein. The results of this model show closer agreement with the experimental results of Avilés et al.. The results of this work are to be presented in a third paper [30].

In order to check the validity of the research programme *in vitro* experiments were
performed with Dr Adriele Prina-Mello of CRANN Research Centre, Trinity College Dublin \[25\]. In these the predicted and real trajectories of the MDCPs can be compared.

In this chapter we investigate the behaviour of MDCPs under the influence of Stokes drag, the force due to hydrodynamic interaction and a magnetic force that incorporates the mutual magnetic dipole-dipole interaction \[21\] whilst ignoring other effects such as inertia and gravity. First, we calculate the magnetic moment of magnetic MDCPs from the Langevin function as indicated in section 5.1. Next, we include the effect of the mutual magnetic dipole-dipole interaction in the magnetic force equation (3.5) and we also calculate the effect of hydrodynamic interaction and the Stokes drag as described in sections 5.2 and 5.3 respectively.

\section{5.1 \textit{Inclusion of the Langevin Function}}

\subsection{5.1.1 \textit{Theory}}

Using the seed model of Avilés et al. \[6\], we consider single domain magnetic nanoparticles as the MDCPs. Furthermore, in the stent model of Avilés et al. \[8\] the MDCPs are microparticles containing single domain magnetic nanoparticles. In the original seed model of Avilés et al. \[6\], the carriers were microparticles and in order to calculate the magnetic moment of each carrier, the axis of the moment $\vec{m}$ of each carrier was taken to lie along that of $\vec{B}$, and the magnetisation was taken to increase with applied field, after accounting for demagnetising as given by (A.7). In contrast, a nanoparticle of diameter $< 100 \text{ nm}$ is typically a superparamagnetic single domain. As a result of thermal agitation, the magnetic moment of such a particle does not, in general, align with the external field. However, the average projection of the moment in the direction of $\vec{B}$ can be calculated from the Langevin function \[20, 26, 36, 75, 83\]

\begin{equation}
L (\beta) = \coth (\beta) - \frac{1}{\beta},
\end{equation}

with Langevin argument

\begin{equation}
\beta = \frac{\omega_{m,p} V_p M_{m,p,s} B}{kT},
\end{equation}
5. Development of Model

5.1. Inclusion of the Langevin Function

where $\omega_{\text{fm,p}}$ is the volume fraction of ferromagnetic material in the MDCP, $V_p$ is the MDCP volume, $M_{\text{fm,p,s}}$ the (volume) saturation magnetisation, $B$ is the magnitude of $\vec{B}$, $k$ is Boltzmann’s constant and $T$ is the absolute temperature, so that magnetic moment, $\vec{m}$, can be written as

$$\vec{m} = \omega_{\text{fm,p}} V_p M_{\text{fm,p,s}} L(\beta) \frac{\vec{B}}{B}. \quad (5.3)$$

The volume fraction of ferromagnetic material, $\omega_{\text{fm,p}}$, in the MDCP is related to its weight fraction, $x_{\text{fm,p}}$, through

$$\omega_{\text{fm,p}} = \frac{x_{\text{fm,p}}}{x_{\text{fm,p}} + (1 - x_{\text{fm,p}}) \rho_{\text{fm,p}}/\rho_{\text{pol,p}}}, \quad (5.4)$$

where $\rho_{\text{fm,p}}$ and $\rho_{\text{pol,p}}$ are the densities of the ferromagnetic material and polymer material respectively in the MDCPs.

Neglecting the inertial forces, the MDCPs are under the influence of Stokes drag and magnetic force as in

$$6\pi \eta_b R_p (\vec{v}_b - \vec{v}_p) + (\vec{m} \cdot \nabla) \vec{B} = 0. \quad (5.5)$$

The velocity of a MDCP, $\vec{v}_p$, can be written as

$$\vec{v}_p = \vec{v}_b + \frac{1}{6\pi \eta_b R_p} (\vec{m} \cdot \nabla) \vec{B}. \quad (5.6)$$

5.1.2 Results

Avilés, Ebner and Ritter [6] suggested a 2D model which uses large ferromagnetic particles as seeds to aid collection of multiple domain particles (radius $\approx 200$ nm). Here, single domain magnetic nanoparticles (radius in the range 20–100 nm) are considered and the Langevin function is used to describe the magnetisation. In our simulations iron nanoparticles with radius, $R_p = 50$ nm, containing 40 wt% iron ($x_{\text{fm,p}} = 0.4$) were taken as the MDCPs and SS 409 as the seed material with seed radius $R_{\text{seed}} = 1$ $\mu$m.

As described in section 5.1.1, the magnetic moment of the individual nanoparticles is taken as the average value given by the Langevin function. The streamline functions for the capture of nanoparticles are presented in figure 5.1 under the influence of homogeneous
magnetic field $\mu_0 H_0$ oriented parallel to the flow ($\varphi = 0$) with magnitudes of 0.0 to 0.6 T. The relevant blood flow properties and the properties of the ferromagnetic material that are used in the MDCPs and for the seeds are given in table 5.1.

![Streamlines indicating the trajectories of the single domain nanoparticles, calculated using OpenFOAM, as they traverse the control volume for different magnitudes of the externally applied magnetic field, $H_0$.](image)

**Figure 5.1:** Streamlines indicating the trajectories of the single domain nanoparticles, calculated using OpenFOAM, as they traverse the control volume for different magnitudes of the externally applied magnetic field, $H_0$.

The resulting CCS, $\lambda_c$, is calculated and presented in figure 5.2 for 50 nm nanoparticles, as a function of the magnetic field strength $\mu_0 H_0$ with magnitudes of 0.0 to 0.8 T. The simulations indicate that through the use of the Langevin function greater CE is predicted than the approach taken by Avilés et al.. Beyond 0.7 T, the MDCP magnetisation is saturated and both models agree as expected.
5. Development of Model

5.2 Inclusion of Dipole-Dipole Interaction

Figure 5.2: Capture cross section, $\lambda_c$, plotted as a function of the applied magnetic field strength, $\mu_0 H_0$, calculated using (---) the Langevin function as appropriate for single domain nanoparticles and (---) following Avilés et al. for multiple domain particles.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>SI Unit</th>
<th>Property</th>
<th>Value</th>
<th>SI Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_b$</td>
<td>1040.0</td>
<td>kg/m$^3$</td>
<td>$\chi_{\text{seed},0}$</td>
<td>1000</td>
<td>–</td>
</tr>
<tr>
<td>$\eta_b$</td>
<td>0.002</td>
<td>kg/ms</td>
<td>$M_{\text{seed},s}$</td>
<td>1397000</td>
<td>A/m</td>
</tr>
<tr>
<td>$u_0$</td>
<td>0.001</td>
<td>m/s</td>
<td>$M_{\text{fm,p,s}}$</td>
<td>1735000</td>
<td>A/m</td>
</tr>
<tr>
<td>$\mu_0 H_0$</td>
<td>0.0–0.8</td>
<td>kg/s$^2$A</td>
<td>$R_{\text{seed}}$</td>
<td>$1.0 \times 10^{-6}$</td>
<td>m</td>
</tr>
<tr>
<td>$x_{\text{fm,p}}$</td>
<td>0.4</td>
<td>–</td>
<td>$R_{\text{p}}$</td>
<td>$50 \times 10^{-9}$</td>
<td>m</td>
</tr>
<tr>
<td>$\rho_{\text{fm,p}}$</td>
<td>7850</td>
<td>kg/m$^3$</td>
<td>$\rho_{\text{pol,p}}$</td>
<td>950</td>
<td>kg/m$^3$</td>
</tr>
<tr>
<td>$\chi_{\text{fm,p,0}}$</td>
<td>1000</td>
<td>–</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Values of the system and material parameters used in the simulation for seed model.

5.2 Inclusion of Interparticle Dipole-Dipole Interaction in the Model

Of interest here is the dipole-dipole interaction between a number of identical magnetic particles. Magnetic dipole-dipole interaction refers to direct interaction between the magnetic dipoles. While dipole-dipole interaction is ubiquitous in magnetic systems, it is relatively weak in comparison to exchange interaction. However, for superparamagnetic nanoparticles exchange interaction between the nanoparticles can be ignored, leaving dipole-dipole interaction as the primary magnetic interaction.

Magnetic dipoles exert a force on each other, which can be included in the magnetic force equation by considering (i) the modified magnetic flux density and (ii) the modifica-
5. Development of Model

5.2. Inclusion of Dipole-Dipole Interaction

Inclusion of Dipole-Dipole Interaction in the magnetic moment resulting from this modified flux density. With regard to the magnetic dipole-dipole interaction between \( N \) MDCPs, each MDCP is taken as spherical with radius, \( R_p \), and sufficiently small to have homogeneous magnetic flux throughout the MDCP. Hence, in order to include the magnetic effect on MDCP \( n \) of the other \((N - 1)\) MDCPs, the magnetic force can be written as

\[
\vec{F}_{\text{int}} = (\vec{m}_n \cdot \nabla) \vec{B}_{\text{total}n}
\]  
(5.7)

where \( \vec{m}_n \) is the total magnetic moment of MDCP \( n \), and it can be written for MDCP \( n \) as,

\[
\vec{m}_n = \omega_{\text{fm,p}} V_{pn} M_{\text{fm,p,s}} L(\beta) \frac{\vec{B}_{\text{total}n}}{B_{\text{total}n}}
\]  
(5.8)

where \( V_{pn} \) is the volume of MDCP \( n \), \( M_{\text{fm,p,s}} \) the (volume) saturation magnetisation of the ferromagnetic particles in the MDCPs, \( L(\beta) \) is the Langevin function, \( \omega_{\text{fm,p}} \) is the volume fraction of ferromagnetic material in the MDCPs and \( \vec{B}_{\text{total}n} \) is the total magnetic flux acting on MDCP \( n \). \( \vec{B}_{\text{total}n} \) is taken as

\[
\vec{B}_{\text{total}n} = \vec{B} + d\vec{B}_1 + \ldots + d\vec{B}_{(n-1)} + d\vec{B}_{(n+1)} \ldots + d\vec{B}_N
\]  
(5.9)

where \( d\vec{B}_n \) is the modification of the resulting magnetic flux density due to MDCP \( n \) at \( \vec{r} \). The modification to the magnetic flux density is thus

\[
d\vec{B}_n(\vec{r}) = \frac{1}{3} \left( \mu_0 M_{\text{fm,p,s}} L(\beta) \right) \frac{R_p^3}{|\vec{r} - \vec{r}_n|^3} \left( \frac{3}{|\vec{r} - \vec{r}_n|^2} (\vec{B}(\vec{r}_n) \cdot (\vec{r} - \vec{r}_n)) (\vec{r} - \vec{r}_n) - \vec{B}(\vec{r}_n) \right)
\]  
(5.10)

where \( \vec{r} \) represents an arbitrary point in space, \( \vec{r}_n \) is the position of the MDCP \( n \) and \( \vec{B}(\vec{r}_n) \) is the flux density at \( \vec{r}_n \). The value of \( \vec{B} \) required to calculate the magnetic force is calculated from the scalar magnetic potential due to the implant, which satisfies the Laplace equation over two con-joined regions: inside and outside the implant as outlined previously in section 4.3.

The velocity of MDCP \( n \) can be obtained by summing the Stokes drag and the modified magnetic force, as given in equations (5.13), and (5.7) respectively with inertial forces,
5. Development of Model

5.3 Inclusion of Hydrodynamic Interaction

\[ \vec{F}_{in} = \vec{F}_{sn} + \vec{F}_{int} = \vec{F}_{in}, \quad (5.11) \]

For MDCP \( n \), ignoring the inertial forces, \( \vec{F}_{in} \), we rewrite (5.11) as

\[ 6\pi\eta_b R_{p_n} (\vec{v}_b - \vec{v}_{p_n}) + (\vec{m}_n \cdot \nabla) \vec{B}_{\text{total}} = 0. \quad (5.12) \]

Hence, we can obtain \( \vec{v}_{p_n} \) by solving (5.12) numerically in each time step.

5.3 Inclusion of Hydrodynamic Interaction in the Model

The Stokes drag for MDCP \( n \) is

\[ \vec{F}_{sn} = 6\pi\eta_b R_{p_n} (\vec{v}_b - \vec{v}_{p_n}), \quad (5.13) \]

where \( \eta_b \) is the viscosity of the blood, \( R_{p_n} \) is the radius of the MDCP \( n \), and \( \vec{v}_b \) and \( \vec{v}_{p_n} \) are the velocities of the blood and MDCP \( n \), respectively. Once more, the blood velocity, \( \vec{v}_b \), is determined by solving the appropriate Navier-Stokes equations as in section 3.4.

The motion of a MDCP through a viscous fluid creates a disturbance to the fluid flow, which will be felt by all other MDCPs. As a result, these MDCPs experience a force which is said to result from hydrodynamic interaction with the original MDCP. By considering \( N \) MDCPs, the force due to the hydrodynamic interaction, \( \vec{F}_{\text{hyd}_n} \), which acts on MDCP \( n \) due to presence of other \((N-1) \) MDCPs, can be written as [58],

\[ \vec{F}_{\text{hyd}_n} = \sum_{(i=1)_{i \neq n}}^{N} \xi_{ni} \cdot (\vec{v}_b - \vec{v}_{p_i}) \quad (5.14) \]

where \( \xi_{ni} \) is the modification due to the hydrodynamic interaction given by

\[ \xi_{ni} = -6\pi\eta_b R_{p_n} \frac{3 R_{p_i}}{4 |\vec{r}_n - \vec{r}_i|^2} \left( 1 + \frac{\vec{r}_n - \vec{r}_i}{|\vec{r}_n - \vec{r}_i|^2} \otimes (\vec{r}_n - \vec{r}_i) \right) \quad (5.15) \]
where $R_{pi}$ is the radius of the MDCP $i$, $\mathbf{1}$ is the unit tensor, $\otimes$ is the vector tensor product (outer product), $\vec{r}_n$ and $\vec{r}_i$ are the positions of MDCP $n$ and MDCP $i$, respectively. Initially MDCPs have the same radius but after agglomeration, MDCPs of different radii are possible, as each agglomeration is viewed as a new MDCP of increased radius.

The velocity of MDCP $n$ can be obtained by summing the Stokes drag, the force due to hydrodynamic interaction and the modified magnetic force, as given in (5.13), (5.14) and (5.7) respectively with inertial forces, $\vec{F}_{in}$, as

$$\vec{F}_{sn} + \vec{F}_{hyd,n} + \vec{F}_{int,n} = \vec{F}_{in}. \quad (5.16)$$

For MDCP $n$, ignoring the inertial forces, $\vec{F}_{in}$, we rewrite (5.16) as

$$6\pi \eta b R_{pn} (\vec{v}_b - \vec{v}_{pn}) + \sum_{i \neq n}^{N} \xi_{ni} \cdot (\vec{v}_b - \vec{v}_{pi}) + (\vec{m}_n \cdot \nabla) \vec{B}_{total,n} = 0. \quad (5.17)$$

Hence, we can obtain $\vec{v}_{pn}$ by solving (5.17) numerically in each time step.

### 5.4 Inclusion of Magnetic Dipole-Dipole and Hydrodynamic Interactions for Two MDCPs

— Seed Model

The strength of forces due to dipole-dipole and hydrodynamic interactions depends on many factors including:

- the magnitude of the applied external magnetic field,
- the initial distance between the MDCPs,
- the relative position of the MDCPs to each other,
- the size of the MDCPs,
- the size of the magnetic implant.
Moreover, the strength of the forces due to hydrodynamic interaction depends on the velocities of MDCPs relative to the blood velocity. For two MDCPs we focus on varying the initial distance between the MDCPs and present the results in terms of agglomeration and the altered capture cross section of the system.

In these simulations stainless steel (SS) 409 is taken as the seed material with a seed radius of 1 µm. Results are presented by generating streamlines for two identical iron nanoparticles with radius $R_p = 50$ nm, containing 40 wt% iron, under the influence of homogeneous magnetic field oriented parallel to the flow ($\varphi = 0$) with magnitude $\mu_0 H_0 = 0.7$ T. The relevant blood flow properties and the properties of the ferromagnetic material, used in the MDCPs and for the seeds, are given in table 5.1.

In order to describe the effect of the interactions we consider two different simulation configurations. The first configuration is intended to illustrate the dependence of the agglomeration point on the interparticle distance for MDCPs that originate within the reference capture cross section (CCS) area. Agglomeration is taken to occur where the (surface-to-surface) interparticle distance reaches zero. The second simulation configuration is intended to examine the effects of interactions on the trajectories of MDCPs near the boundary of the reference CCS and the resulting changes in the CCS. The boundary of the reference CCS, $\lambda_c^*$, is the trajectory of the last MDCP, which would be captured by the seed in the non-interacting case. In these two MDCP simulations, the behaviour of the MDCPs after agglomeration is not considered. The MDCPs are taken to have the same initial $x$-coordinate with an initial interparticle distance, $D$. Initial interparticle distance is defined as the distance between the surfaces of the MDCPs. These initial conditions serve to illustrate the effect of the interparticle distance on behaviour. The coordinates and nanoparticle dimensions used are scaled in terms of $R_{seed}$ and hence the scaled nanoparticle radius is 0.05.

### 5.4.1 Effect of Interactions on the Agglomeration of MDCPs

Of interest is the relationship between initial interparticle distance, $D$, and the resulting position of the agglomeration point as measured from the surface of the seed. This relationship is shown in figure 5.3 with
5. Development of Model

5.4. Inclusion of Interactions for Two MDCPs

- dipole-dipole magnetic interaction only,
- hydrodynamic interaction only,
- both interactions
- no interactions.

As expected, in all cases, the distance between the agglomeration point and centre of the seed is seen to decrease as $D$ increases. In these simulations, two MDCPs, labelled MDCP 1 and MDCP 2, are placed at scaled positions $(-20, D/2 + R_p/R_{seed})$ and $(-20, -D/2 - R_p/R_{seed})$ for a range of values of $D$. The inset shows the initial position of the MDCPs and their trajectories for all cases for a typical value of $D$ ($D = 0.40$).

On comparing the agglomeration point for the MDCPs with only magnetic dipole-dipole interaction to that for the MDCPs with no interaction, we find that the MDCPs with magnetic interaction agglomerate earlier for all initial MDCP distances up to $D=1$ (see figure [5.3]). Also in figure [5.3] with the inclusion of hydrodynamic interaction only, the two MDCPs are seen to repel each other due to their velocities relative to the blood, inhibiting agglomeration. It is worth noting that the relative velocities are solely due to the magnetic velocities resulting from the presence of the seed gradient. In the inset it is seen that in the case with (only) hydrodynamic interaction the MDCPs agglomerate after the agglomeration point expected without any interactions. With the study of the combined effect of magnetic dipole-dipole and hydrodynamic interactions, we observe that, as expected, at short range the magnetic effects dominate, and at longer range the hydrodynamic are dominant. This is consistent with the forces being dependent on $|\vec{r}_1 - \vec{r}_2|^{-3}$ and $|\vec{r}_1 - \vec{r}_2|^{-1}$ respectively. From figure [5.3] a critical value of $D$ can be observed at the intersection of the curves with both interactions and no interactions at $D \approx 0.56$. Below this critical value of $D$, the two MDCPs are seen to agglomerate before the agglomeration point expected without interactions. For initial distances larger than this critical value of $D$, (repulsive) hydrodynamic forces dominate and the MDCPs agglomerate after the agglomeration point expected without interactions (i.e. closer to the seed).
5. Development of Model

5.4. Inclusion of Interactions for Two MDCPs

5.4.2 Effect of Interactions on the Capture Cross Section of the System

In figures 5.4 - 5.8 trajectories are presented and the effect of the inclusion of interactions on the CCS of the system is studied. The trajectories of two MDCPs are calculated again with

- dipole-dipole magnetic interaction only,
- hydrodynamic interaction only,
- both interactions.

In all three interaction cases, the trajectories of MDCPs without any interactions and the resulting boundary of the reference CCS, \( \lambda^*_c \) are used as reference points. Thus, two
different trajectories are generated for each MDCP in each figure.

MDCPs are placed equidistant and symmetric about $\lambda_c^*$, corresponding to the initial position ($-20, \lambda_c^*$) where $\lambda_c^*$ is 4.47. This critical value $\lambda_c^*$ depends on the model parameters used (see table 5.1). In each of first three cases, a maximum value of $D$, whereby two MDCPs are still captured by the seed is determined. For each interaction case, the separate effect on $\lambda_c$ for this maximum value of $D$ is calculated and compared.

In the case with (only) magnetic dipole-dipole interaction, we find that the maximum value of $D$ for which both MDCPs are now captured is 0.40. In figure 5.4, the trajectories for this case are presented. Thus, for this specific initial arrangement, the calculated capture radius can be said to increase by 0.25 $R_{seed}$ corresponding to a $\approx 6\%$ absolute increase in $\lambda_c$.

**Figure 5.4:** The trajectories of the MDCPs are presented with the magnetic dipole-dipole interaction and no interactions. Initial position of MDCP 1 & 2 are ($-20, \lambda_c^* + 0.25$) & ($-20, \lambda_c^* - 0.25$). With the magnetic dipole-dipole interaction both MDCPs are captured.
To explore further the effect of magnetic dipole-dipole interaction on the CCS of the system, the initial position of both MDCPs is translated vertically whilst maintaining a fixed interparticle distance of $D = 0.40$. In the first case, by moving the MDCPs downwards, as expected both MDCPs continue to be captured by the seed, and thus the CCS of system is unchanged. In the second case by moving both MDCPs upwards the following is observed. As might be expected, the upper MDCP (MDCP 1) is no longer captured by the seed. By further moving the two MDCPs upwards we next observe that the initial position at which the lower MDCP (MDCP 2) ceases to be captured by the seed is now lower than for the non-interacting case. Thus, MDCP 1 has caused the *non capture* of MDCP 2. Figure 5.5 illustrates this undesirable effect on the capture radius of the system where it is decreased by $0.16\, R_{\text{seed}}$, which corresponds to a $\approx 4\%$ absolute decrease in $\lambda_c$.

![Graph](image)

**Figure 5.5:** The trajectories of the MDCPs are presented with the magnetic dipole-dipole interaction and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda_c^* + 0.35)$ & $(-20, \lambda_c^* - 0.15)$. With the magnetic dipole-dipole interaction, neither MDCP is captured.
In the case with (only) hydrodynamic interaction, we find that the maximum value of $D$ for which both MDCPs are now captured is 0.41 which is slightly larger than in the case with (only) magnetic dipole-dipole interaction. Here, the upper MDCP (MDCP 1) repels the lower MDCP (MDCP 2) and the lower MDCP attracts the upper MDCP due to their velocities relative to the velocity of blood. In figure 5.6, the trajectories for this case are presented. Thus, for this specific initial arrangement, the calculated capture radius can be said to increase by 0.255 $R_{\text{seed}}$ corresponding to a $\approx 6\%$ absolute increase in $\lambda_c$.

![Figure 5.6: The trajectories of the MDCPs are presented with the hydrodynamic interaction and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda^*_c+0.255)$ & $(-20, \lambda^*_c-0.255)$. With the hydrodynamic interaction both MDCPs are captured.](image)

To explore further the effect of hydrodynamic interaction on the CCS of the system, the initial position of both MDCPs is translated vertically whilst maintaining a fixed interparticle distance of $D = 0.41$. In the first case, by moving the MDCPs downwards, as expected both MDCPs continue to be captured by the seed, and thus the CCS of
system is unchanged. In the second case, by moving both MDCPs upwards, the upper MDCP (MDCP 1) is no longer captured by the seed as expected. By further moving the two MDCPs upwards, the initial position at which the lower MDCP (MDCP 2) ceases to be captured by the seed is still higher than for the non-interacting case. Thus, MDCP 1 has caused the capture of MDCP 2 by pushing it towards the seed. Figure 5.7 illustrates this positive effect on the capture radius of the system where it is increased by 0.134 $R_{\text{seed}}$, which corresponds to a $\approx 3\%$ absolute increase in $\lambda_c$. For this specific case, if the value of $D$ is decreased to 0.40 as in the case with (only) magnetic dipole-dipole interaction, the capture radius of the system increases by 0.138 $R_{\text{seed}}$. It should be noted that for hydrodynamic interaction, that the direction of velocity of MDCPs relative to the fluid is an important factor.

![Figure 5.7: The trajectories of the MDCPs are presented with the hydrodynamic interaction and without any interaction. Initial position of MDCP 1 & 2 are $(-20, \lambda^*_c + 0.644)$ & $(-20, \lambda^*_c + 0.134)$. With hydrodynamic interaction, MDCP 2 is now captured.](image-url)
With the inclusion of both interactions, we find that the maximum value of $D$ for which both MDCPs are now captured is 0.54. In figure 5.8, the trajectories for this case are presented. For this initial arrangement, the calculated capture radius can be said to increase by $0.32 \, R_{\text{seed}}$ corresponding to a $\approx 7\%$ absolute increase in $\lambda_c$. In this case, the magnetic dipole-dipole and hydrodynamic interactions both have a positive effect on the CCS of the system.

To study the combined effect of both interactions, we include the hydrodynamic interaction to the case with (only) magnetic dipole-dipole interaction. Thus, the simulations are repeated with a fixed interparticle distance of $D = 0.40$ and the CCS of the system is calculated. In the first case, by moving the MDCPs downwards, both MDCPs continue to be captured by the seed, and thus the CCS of system is unchanged. In the second case by moving both MDCPs upwards the following is observed. Again, the upper MDCP (MDCP

![Figure 5.8: The trajectories of the MDCPs are presented with both interactions and no interactions. Initial position of MDCP 1 & 2 are ($-20, \lambda^*_c + 0.32$) & ($-20, \lambda^*_c - 0.32$). With both interactions both MDCPs captured.](image-url)
1) is no longer captured by the seed. By further moving the two MDCPs upwards we
next observe that the initial position at which the lower MDCP (MDCP 2) ceases to be
captured by the seed is the same as the non-interacting case. When the value of $D$ is 0.40,
we find that inclusion of both interactions does not affect the CCS of the system as the ef-
fec ts of magnetic dipole-dipole interaction and hydrodynamic interaction on CCS balance
each other. Thus, inclusion of hydrodynamic interaction has caused the increase of the
capture radius by 0.16 $R_{seed}$, relative to the case with magnetic dipole-dipole interaction
only. Similarly, inclusion of magnetic dipole-dipole interaction has caused the decrease of
the capture radius by 0.138 $R_{seed}$, relative to the case with hydrodynamic interaction only
when the value of $D$ is 0.40. These apparent imbalances we attribute to the inherent the
nonlinearity and cross dependence of the two interactions. Furthermore, for this specific
case, if we decrease the value of $D$, the magnetic dipole-dipole interaction becomes dom-
ninant and if we increase the value of $D$, the hydrodynamic interaction dominates again
consistent with the $|\vec{r}_1 - \vec{r}_2|^{-3}$ and $|\vec{r}_1 - \vec{r}_2|^{-1}$ dependence. Specifically, in our model with
inclusion of both interactions, the effect of magnetic dipole-dipole interaction on the CCS
of the system is larger than the effect of the hydrodynamic interaction when the value of $D$ is less than 0.40.

The effect of the dipole-dipole and hydrodynamic interactions between two nanopar-
ticles on the calculated magnetic force in the IA-MDT seed model of Aviles et al. is
considered. In these simulations, depending on the initial configuration of the nanoparti-
cles, both increases and decreases can be observed in the CCS of the modified model. It
is observed that, both dipole-dipole and hydrodynamic interactions should be considered
to calculate the CCS of the IA-MTD system due to comparable size of both interactions.
Inclusion of both interactions was seen to alter the CCS of the system by up to 7% in
absolute terms. We note that the relative positions of the MDCPs and the velocities of
MDCPs relative to blood flow are important factors during the calculation of the effect of
hydrodynamic interaction on the capture radius of the system. Also, we note that if two
MDCPs can agglomerate and start moving together it might be expected that their al-
tered hydrodynamic volume would reduce the effective Stokes drag allowing both MDCPs
to be more easily captured by the seed and thus leading to an additional CCS increase.
5.5 *Inclusion of Magnetic Dipole-Dipole and Hydrodynamic Interactions for Multiple MDCPs* — *Stent Model*

In this part of the work, we include the effect of both magnetic dipole-dipole and hydrodynamic interactions for multiple MDCPs in the stent based mathematical model of Avilés et al. [8]. We focus on varying the initial positions of \( N \) (\( N < 20 \)) MDCPs at the entrance of the CV and present the results in terms of the CE of the system considering the agglomeration of MDCPs.

Of interest is the relationship between the velocity of the blood and the field strength on the CE of the system. This relationship is shown in figures 5.9 and 5.10 with

- both dipole-dipole magnetic and hydrodynamic interactions,
- experimental results,
- no interactions.

The experimental results presented are those of Avilés et al. [8]. In our simulations, with larger field and lower blood velocity, MDCPs agglomerated to create a cluster of larger volume. In the experiment of Avilés et al. [8] stainless steel (SS) 430 is taken as the wire material for the stent with a 62.5 \( \mu \text{m} \) radius. The stent is prepared by looping a length of wire (\( L \)) into a 2 cm long coil having a 0.5 mm radius containing 10 loops (\( N_l \)) with 0.2 cm between each loop. This stent is placed in a tube with radius of 0.5 mm. In order to effectively model this system, the 3D geometry of the stent and tube is reduced to 2-D slice through the centre of the tube (See figure 3.2). The coiled stent was modelled as a series of circular cross sections of an infinite wire with radius of \( R_{\text{wire}} \) located at the upper and lower boundaries of the walls. At each wall the wires are separated by a distance, \( h \), between their centres, and the upper and lower sections are offset by \( h/2 \) as shown in figure 3.2. It should be noted that physically this corresponds to a 2-D description of flow with a parabolic profile in a rectangular box with transverse cylindrical wires, all of infinite extent.
Results are presented by calculating the CE’s for identical MDCPs with initial radius $R_p = 0.435 \mu m$ containing 25 wt% magnetite, under the influence of homogeneous magnetic field oriented perpendicularly to the flow ($\varphi = \pi/2$) with magnitudes of 0.17 T to 0.65 T. In the model the magnetisation of the individual MDCPs is taken as the average value given by the Langevin function due to the single domain magnetic nanoparticles inside them. The relevant fluid flow properties and the properties of the ferromagnetic material, used in the MDCP and for the stent wire, are given in table 5.2.

In this 2D model, the behaviour of the MDCPs after agglomeration is also considered. We assume that the MDCPs create a cluster during their agglomeration as a result of both interactions. The volume of the cluster is calculated by summing the volume of the MDCPs agglomerated and the radius of the cluster is calculated using the general volume formulation $\left(\frac{4}{3}\pi r^3\right)$.

Whilst this assumption does not account fully for the resulting hydrodynamic volume, the effect of this assumption should not significantly affect our results.

The rationale for the simulations is as follows. Given infinite computing power, one might consider randomly distributing, in the form of a cloud, a very large number ($>10,000$) of MDCPs and allow interactions between all of these. With finite computing resources, one is forced to reduce this. We do this in two ways. Firstly, by limiting the regions of initial positions that we consider and secondly by limiting the number of MDCPs that we allow to mutually interact. Thus we consider only those parts of the simulation which are likely to contribute to any alteration in the CE. For instance, in those parts of the capture cross section closest to the vessel walls, one can expect no improvement in the CE. In fact it is only where the initial positions are close to the border between the collection and no collection region, that is around $\lambda^*_c$, that we start to see altered trajectories due to interactions. Secondly, the mutual interparticle interaction would not be expected to have infinite extent. One can postulate a number $P$ of MDCPs in the model where the predicted difference in performance between modelling $P$ and $P+1$ becomes arbitrarily small/insignificant. We point out that the computational effort required to model interactions scales with $N^2$, where $N$ is the number of MDCPs interacting. Simulations were performed for increasing $N$, and the results indicate that there is no significant change to the system performance metrics beyond twenty MDCPs.
We consider a particular, arbitrarily distributed cloud of $N$ MDCPs. This cloud is to be placed with its centre on the line of the reference CCS. We associate a scaling distance with this cloud and increase this distance until for a given field and fluid velocity, this cloud still results in all MDCPs agglomerating into a single cluster within the simulation. Such a scaled cloud is then obtained for each field considered using the lowest (non-zero) fluid velocity considered. In our simulations we choose this reference velocity ($u_0 = 2.1$ cm/s) as this was the lowest considered in the experiments of Avilés et al.. This scaled cloud is then used as the starting point for simulations.

Thus, in order to describe the effect of the both interactions we consider two different simulation configurations, similar to those used previously for the inclusion of interactions for two MDCPs. The first configuration is intended to illustrate the agglomeration of the MDCPs within the reference CCS area. In this configuration all of the MDCPs are captured, as expected and the resulting CE of the system for this situation is unaltered.

The second simulation configuration is intended to examine the effects of interactions on the CE of the system near the boundary of the reference capture cross section. The boundary of the reference CCS, $\lambda^*_c$ is the trajectory of the last MDCP, which would be captured by the stent wires in the non-interacting case. For this, we place the cloud centre on the $\lambda^*_c$ for a given velocity and record changes in CE through following the MDCP trajectories in the normal way. We then translate this cloud up and down, and again record changes in CE. This approach is repeated for each increased fluid velocity, using, for a given field, the same scaled cloud.

For the configurations outlined above, we keep the the applied field constant ($\mu_0 H_0 = 0.17$ T) and we increase the blood velocity up to $u_0 = 42.4$ cm/s. The resulting CEs for these simulations are shown in figure \ref{fig:5.10}.

Secondly, using the same methodology we obtain a reference cloud with the applied field $\mu_0 H_0 = 0.65$ T and low fluid velocity ($u_0 = 2.1$ cm/s). Again, we increase the blood velocity up to $u_0 = 42.4$ cm/s for fixed field $\mu_0 H_0 = 0.65$ T. The resulting CEs are given in figure \ref{fig:5.9}.

In figures \ref{fig:5.9} and \ref{fig:5.10}, the results of the model with the interactions show closer agreement with experimental results of Avilés et al. with low fluid velocity. This is due to the interaction and agglomeration of MDCPs in our model. With low fluid velocity
<table>
<thead>
<tr>
<th>Properties</th>
<th>Symbol</th>
<th>Units</th>
<th>Values</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MDCPs Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polymer material</td>
<td>-</td>
<td>-</td>
<td>P(S/V-COOH)Mag</td>
<td>Physical</td>
</tr>
<tr>
<td>Radius</td>
<td>( R_p )</td>
<td>( \mu m )</td>
<td>0.435</td>
<td>Physical</td>
</tr>
<tr>
<td>Saturation magnetisation</td>
<td>( M_{p,s} )</td>
<td>kA/m</td>
<td>22.4</td>
<td>Measured</td>
</tr>
<tr>
<td><strong>MDCPs Magnetic Material Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>-</td>
<td>Magnetite</td>
<td></td>
<td>Physical</td>
</tr>
<tr>
<td>Weight content</td>
<td>( x_{im,p} )</td>
<td>wt%</td>
<td>25</td>
<td>Physical</td>
</tr>
<tr>
<td>Volume content</td>
<td>( \omega_{im,p} )</td>
<td>-</td>
<td>6.4</td>
<td>Measured</td>
</tr>
<tr>
<td>Saturation magnetisation</td>
<td>( M_{im,p,s} )</td>
<td>kA/m</td>
<td>351.9</td>
<td>Measured</td>
</tr>
<tr>
<td>Magnetic moment</td>
<td>( m_{im,p} )</td>
<td>Am²</td>
<td>( 2.03 \times 10^{-19} )</td>
<td>Measured</td>
</tr>
<tr>
<td>Radius</td>
<td>( R_{im,p} )</td>
<td>nm</td>
<td>5.18</td>
<td>Calculated</td>
</tr>
<tr>
<td><strong>Physical Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>( T )</td>
<td>K</td>
<td>300</td>
<td>Physical</td>
</tr>
<tr>
<td>Boltzmann’s constant</td>
<td>( k_B )</td>
<td>J/K</td>
<td>( 1.38 \times 10^{-23} )</td>
<td>Physical</td>
</tr>
<tr>
<td>Permeability of vacuum</td>
<td>( \mu_0 )</td>
<td>Tm/A</td>
<td>( 4\pi \times 10^{-7} )</td>
<td>Physical</td>
</tr>
<tr>
<td><strong>Applied Field Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magnitude</td>
<td>( \mu_0 H_0 )</td>
<td>T</td>
<td>0.0–0.7</td>
<td>Physical</td>
</tr>
<tr>
<td>Angle of field direction</td>
<td>( \varphi )</td>
<td>-</td>
<td>( \pi/2 )</td>
<td>Physical</td>
</tr>
<tr>
<td><strong>Stent Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>-</td>
<td>-</td>
<td>SS 430</td>
<td>Physical</td>
</tr>
<tr>
<td>Wire radius</td>
<td>( R_{wire} )</td>
<td>( \mu m )</td>
<td>62.5</td>
<td>Physical</td>
</tr>
<tr>
<td>Loop separation</td>
<td>( h )</td>
<td>mm</td>
<td>2</td>
<td>Physical</td>
</tr>
<tr>
<td>Number of loops</td>
<td>( N_l )</td>
<td>-</td>
<td>10</td>
<td>Physical</td>
</tr>
<tr>
<td>Coil length</td>
<td>( L )</td>
<td>cm</td>
<td>2</td>
<td>Physical</td>
</tr>
<tr>
<td>Saturation magnetisation</td>
<td>( M_{implant,s} )</td>
<td>kA/m</td>
<td>1261</td>
<td>Measured</td>
</tr>
<tr>
<td>Magnetic susceptibility</td>
<td>( \chi_{implant,0} )</td>
<td>-</td>
<td>1000</td>
<td>Physical</td>
</tr>
<tr>
<td><strong>Blood &amp; Vessel Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velocity</td>
<td>( u_0 )</td>
<td>cm/s</td>
<td>2.1, 4.2, 10.6, 21.2, 42.4</td>
<td>Physical</td>
</tr>
<tr>
<td>Volume</td>
<td>( V_{blood} )</td>
<td>ml</td>
<td>10</td>
<td>Physical</td>
</tr>
<tr>
<td>Density</td>
<td>( \rho_b )</td>
<td>kg/m³</td>
<td>1000</td>
<td>Physical</td>
</tr>
<tr>
<td>Viscosity</td>
<td>( \eta_b )</td>
<td>kg/ms</td>
<td>( 1.0 \times 10^{-3} )</td>
<td>Physical</td>
</tr>
<tr>
<td>Vessel radius</td>
<td>( R_{vessel} )</td>
<td>mm</td>
<td>0.5</td>
<td>Physical</td>
</tr>
</tbody>
</table>

*Table 5.2: Values of system and material parameters used in the stent based simulation.*
5. Development of Model  

5.5. Inclusion of Interactions for Multiple MDCPs

Figure 5.9: The collection efficiency (CE) of the system plotted as a function of the blood velocity at the applied field $\mu_0 H_0 = 0.65$ T.

($\leq 10$ cm/s) and higher applied field ($\mu_0 H_0 = 0.65$ T) MDCPs create a larger volume of cluster more easily than with the lower applied field ($\mu_0 H_0 = 0.17$ T). When we increase the fluid velocity the likelihood of the agglomeration of the MDCPs starts to decrease. For higher fluid velocity the CE of the IA-MDT system gives smaller results than the results of Avilés model without interactions. This is due to the effect of interactions on the velocity of MDCPs and so the trajectories of the MDCPs.
Figure 5.10: The collection efficiency (CE) of the system plotted as a function of the blood velocity at the applied field $\mu_0H_0=0.17$ T.
Chapter 6

Conclusions

Firstly, the model of Avilés, Ebner and Ritter is considered for collecting single domain MDCPs. The Langevin function is used to calculate the expected value of the nanoparticle magnetisation. Magnetic flux density is calculated analytically by using the separation of variable solution and the the blood velocity is obtained from the Navier-Stokes equations using the finite volume library OpenFOAM. The simulations indicate that use of the Langevin function predicts greater collection efficiency than might be otherwise expected.

Secondly, the effect of the dipole-dipole and hydrodynamic interactions between two nanoparticles on the calculated magnetic force in the IA-MTD system of Avilés et al. is considered. In these simulations, depending on the initial configuration of the nanoparticles, both increases and decreases can be observed in the capture cross section of the modified model. It is observed that both dipole-dipole and hydrodynamic interactions should be considered to calculate the capture cross section of the IA-MTD system due to comparable size of both interactions. Inclusion of both interactions was seen to alter the capture cross section of the system by up to 7% in absolute terms. We note that the relative positions of the particles and the relative velocities of particles to blood flow are important factors during the calculation of the effect of interactions on the capture radius of the system. Also, we note that if two particles can agglomerate and start moving together it might be expected that their altered hydrodynamic volume would reduce the effective Stokes drag allowing both particles to be more easily captured by the seed and thus leading to an additional capture cross section increase.

Finally, we have presented an interaction model applied to IA-MTD. This model considered the agglomeration of particles known to occur in such systems [5, 8, 9]. We include the effects of both the dipole-dipole and hydrodynamic interactions for multiple particles in stent implant arrangements. The resulting collection efficiencies from this model are in closer agreement with the experimental results of Avilés et al..
Bibliography


75
Appendix A
Scalar Potential Formulation

When implementing the basic model we initially calculated the magnetic scalar potential using a numerical solver for the Laplace equation. However, this approach was not satisfactory due to the time needed for convergence. Numerical difficulties when calculating the scalar potential are not unusual and one approach to address these difficulties is to make use of the total scalar potential in addition to the reduced scalar potential. During the investigation of this, it was realised that an analytic expression for the scalar potential was obtainable using separation of variables. Calculation of scalar potential is done in three different ways and these derivations are based on the formulations:

- Using total scalar potential in both regions, which we term the total-total scalar potential formulation.
- Using reduced scalar potential in both regions, termed the reduced-reduced scalar potential formulation. In general, this formulation has convergence problems when used in a numerical scheme. This formulation has also non-uniqueness difficulties when determining the potential in the region containing the source (space).
- Using total scalar potential in the source free region (implant) and the reduced scalar potential in region with the source (space), labelled the total-reduced scalar potential formulation.

We note the analytic solution does not depend on the total/reduced scalar potential formulation used in the derivation.

Firstly, in section A.1 the permeability of the implant required for the scalar potential formulations is considered. Reduced-reduced scalar potential has been presented in section 4.3.1. Here, the total-total and total-reduced scalar potentials are presented. In the last section, the magnetic force density is derived.

A.1 Calculation of Permeability of the Implant

In order to calculate the magnetic field arising from placing an implant (wire, seed, stent) within a space region of constant permeability, the permeability of the implant is required. This can be obtained from the relationship between the (relative) permeability and susceptibility of a material given by \( \mu_r = 1 + \chi \). In this section, we obtain the permeability of the implant, from the expressions for susceptibility in Ritter et al. [66]. Ritter et al. represented the demagnetising factor, \( \alpha_{fm,p} \), for a ferromagnetic material in the MDCP as,

\[
\alpha_{fm,p} = \min \left( \frac{\chi_{fm,p,0}}{3 + \chi_{fm,p,0}}, \frac{M_{fm,p,s}}{3H_0} \right)
\]  

(A.1)

where \( \chi_{fm,p,0} \) is the (volumetric) magnetic susceptibilities of the ferromagnetic material in the MDCP with zero field \( (H_0 = 0) \) and \( M_{fm,p,s} \) saturation magnetisation of the ferromagnetic material in the MDCP. From the first part of (A.1), Ritter et al. calculated
the magnetic susceptibility of the material, $\chi_{\text{fm,p}}$, and using the second part of (A.1), the induced magnetisation of the ferromagnetic material in the MDCP, $M_{\text{fm,p}}$, is calculated. Rearranging the first and second parts of (A.1), the (volumetric) magnetic susceptibility of the material, $\chi_{\text{fm,p}}$, and and the induced magnetisation of the ferromagnetic material are obtained as below,

$$\chi_{\text{fm,p}} = 3 \frac{\alpha_{\text{fm,p}}}{1 - \alpha_{\text{fm,p}}}$$  \hspace{1cm} (A.2)  

$$M_{\text{fm,p}} = 3 \alpha_{\text{fm,p}} H_0.$$  \hspace{1cm} (A.3)  

Using the same methodology, we derive the susceptibility and the permeability of the implant. In Ritter et al. [66], the demagnetising factor of the implant, $\alpha_{\text{implant}}$, is given as,

$$\alpha_{\text{implant}} = \min \left( \frac{\chi_{\text{implant,o}}}{2 + \chi_{\text{implant,o}}} \frac{M_{\text{implant,s}}}{2H_0} \right)$$  \hspace{1cm} (A.4)  

where $\chi_{\text{implant,o}}$ is the volumetric magnetic susceptibility of the implant (wire, seed, stent) with zero field and $M_{\text{implant,s}}$ is the saturation magnetisation of the implant. The factor 3 appears for the sphere and the factor 2 for the cylinder. Therefore the magnetic susceptibility, $\chi_{\text{implant}}$, and the induced magnetisation of the implant, $M_{\text{implant}}$, in the 2D case are written as,

$$\chi_{\text{implant}} = 2 \frac{\alpha_{\text{implant}}}{1 - \alpha_{\text{implant}}}$$  \hspace{1cm} (A.5)  

$$M_{\text{implant}} = 2 \alpha_{\text{implant}} H_0.$$  \hspace{1cm} (A.6)  

For calculating the relative permeability of implant, $\mu_{\text{implant}}$, we use the formula $\mu_{\text{implant}} = 1 + \chi_{\text{implant}}$ and for the relative permeability of space, $\mu_{\text{space}}$, we use $\mu_{\text{space}} = 1$ since the susceptibility of free space is zero.

We note that in the models of Ritter et al. [66] and Avilés et al. [8] the same approach is taken for the implant and for the MDCPs and so

$$\vec{m} = \omega_{\text{fm,p}} V_p M_{\text{fm,p}} \vec{B} \frac{\vec{B}}{B}$$  \hspace{1cm} (A.7)  

where, $\omega_{\text{fm,p}}$ is the volume fraction of ferromagnetic material as in (5.4).

\textbf{A.2 Solution Using Total-Total Potential Formulation}

The associated total-total magnetic scalar potential, $\psi_0$, satisfies

$$\vec{H}_0 = - \nabla \psi_0 \text{ hence } (H_0, 0) = (-\partial_x \psi_0, -\partial_y \psi_0).$$  \hspace{1cm} (A.8)  

Integrating (A.8) we obtain the scalar potential as,

$$\psi_0(x, y) = -H_0 x.$$  \hspace{1cm} (A.9)
In cylindrical polar coordinates the scalar potential can be written as

$$\psi_0(r, \theta) = -H_0 r \cos \theta$$  \hspace{1cm} (A.10)

where $\theta$ is the angle from the positive $x$-axis, as in figure 3.1. In region 1 (implant) the magnetic field, the magnetic flux density and the total–total magnetic scalar potential are related by [76] to

$$\vec{H}_1 = -\nabla \psi_1, \quad \vec{B}_1 = \mu_1 \vec{H}_1, \quad \nabla^2 \psi_1 = 0$$  \hspace{1cm} (A.11)

and in region 2 (space)

$$\vec{H}_2 = -\nabla \psi_2, \quad \vec{B}_2 = \mu_2 \vec{H}_2, \quad \nabla^2 \psi_2 = 0. \quad \hspace{1cm} (A.12)$$

The normal component of $\vec{B}$ and the tangential component of $\vec{H}$ are continuous across the boundary of regions with different permeability. Eqs. (A.11) and (A.12) are linked with the interface conditions

$$\vec{n} \cdot \vec{B}_1 = \vec{n} \cdot \vec{B}_2, \quad \mu_1 \frac{\partial \psi_1}{\partial r} = \mu_2 \frac{\partial \psi_2}{\partial r}$$  \hspace{1cm} (A.13)

and

$$\hat{t} \cdot \vec{H}_1 = \hat{t} \cdot \vec{H}_2, \quad \frac{\partial \psi_1}{\partial \theta} = \frac{\partial \psi_2}{\partial \theta} \quad \hspace{1cm} (A.14)$$

and satisfy the boundary conditions

$$\vec{H}_2 \rightarrow \vec{H}_0 \text{ as } r \rightarrow r_{\infty} \quad \hspace{1cm} (A.15)$$

and $\vec{H}_1$ is bounded. Poisson equation (A.13)(c) and (A.14)(c) have solution

$$\psi_j = (a_j r + b_j r^{-1}) \cos \theta \quad \hspace{1cm} (A.16)$$

for some undetermined coefficients $a_j$ and $b_j$. Hence, for each region we have

$$\psi_1 = (a_1 r + b_1 r^{-1}) \cos \theta \quad \hspace{1cm} (A.17)$$

and

$$\psi_2 = (a_2 r + b_2 r^{-1}) \cos \theta. \quad \hspace{1cm} (A.18)$$

Firstly, the solution in both regions must be bounded, hence

$$b_1 = 0. \quad \hspace{1cm} (A.19a)$$

Applying the boundary condition at $r = r_{\infty}$

$$\lim_{r \rightarrow r_{\infty}} \psi_2 = \psi_0 \implies a_2 r_{\infty} + b_2 r_{\infty}^{-1} = -H_0 r_{\infty} \implies a_2 = -H_0 - b_2 r_{\infty}^2. \quad \hspace{1cm} (A.19b)$$

Alternatively, as $r_{\infty} \rightarrow \infty$ we obtain condition

$$a_2 = \lim_{r_{\infty} \rightarrow \infty} (-H_0 - b_2 / r_{\infty}^2) = -H_0$$
A.3. Total–Reduced Potential Formulation

Using interface condition (A.13), we have at 
\[ r = 1 \]
\[ \mu_1 \frac{\partial \psi_1}{\partial r} = \mu_2 \frac{\partial \psi_2}{\partial r} \implies \mu_1 (a_1 - b_1) \cos \theta = \mu_2 (a_2 - b_2) \cos \theta \]
\[ \implies \mu_1 a_1 = \mu_2 (a_2 - b_2) \tag{A.19c} \]
and applying interface condition (A.14), we have
\[ \frac{\partial \psi_1}{\partial \theta} = \frac{\partial \psi_2}{\partial \theta} \implies (a_1 + b_1) \cos \theta = (a_2 + b_2) \cos \theta \]
\[ \implies a_1 = a_2 + b_2. \tag{A.19d} \]

The solution to the system of (A.19a) to (A.19d) for \( a_1, a_2, b_1, \) and \( b_2 \) is written in terms of \( a_2 \) as
\[ a_2 = -H_0 \left[ \frac{r_\infty^2 (\mu_1 + \mu_2)}{-(\mu_1 - \mu_2) + r_\infty^2 (\mu_1 + \mu_2)} \right] = -H_0 \left[ 1 - \frac{1}{r_\infty^2} \left( \frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) \right]^{-1} \tag{A.20a} \]
and
\[ a_1 = \frac{2\mu_2}{\mu_1 + \mu_2} a_2, \quad b_1 = 0, \quad b_2 = -\frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} a_2. \tag{A.20b} \]

If we take the limit as \( \mu_1 \to \mu_2 \) we expect \( \psi_1 = \psi_2 = \psi_0 \) and consequently \( H_1 = H_2 = H_0 \).

Thus, in this limit we find
\[ \lim_{\mu_1 \to \mu_2} a_2 = -H_0, \quad \lim_{\mu_1 \to \mu_2} a_1 = -H_0, \quad \lim_{\mu_1 \to \mu_2} b_2 = 0. \tag{A.21} \]

A.3 Solution Using Total–Reduced Potential Formulation

Calculation of magnetic scalar potential is presented using total–reduced potential formulation. In region 1 (implant) the magnetic field, the magnetic flux density and the total–reduced scalar potentials are related by
\[ \psi_1 = (a_1 r + b_1 r^{-1}) \cos \theta \tag{A.22} \]
and
\[ \phi_2 = (a_2 r + b_2 r^{-1}) \cos \theta. \tag{A.23} \]

Applying these conditions we obtain
\[ a_1 = \frac{2\mu_2 a_2 - 2\mu_2 H_0}{\mu_1 + \mu_2}, \quad b_1 = 0, \tag{A.24} \]
and
\[ a_2 = -\frac{b_2}{r_\infty^2} = -H_0 \left[ 1 - \frac{1}{r_\infty^2} \frac{\mu_1 + \mu_2}{\mu_1 - \mu_2} \right]^{-1}, \quad b_2 = -\frac{(\mu_1 - \mu_2) a_2 + (\mu_1 - \mu_2) H_0}{\mu_1 + \mu_2} \tag{A.25} \]
A.4 Derivation of Magnetic Force Density

In this section, in order to compare the results of magnetic flux density, $\vec{B}$, generated by OpenFOAM with the results of the Ritter et al. [66], a simple term magnetic force density, $f_w$, is derived. $f_w$ is given as [6, 66]

$$f_w = \left| \mu_0 \nabla (\vec{H} \cdot \vec{H}) \right| = \left| \mu_0 \nabla H^2 \right|$$  \hspace{1cm} (A.26)

where $H$ is the magnitude of the total magnetic field from $H = B/\mu$.

Here, $f_w$ is derived through the magnetic scalar potential formulation. We consider

$$\vec{H} = \vec{H}_0 - \nabla \phi$$  \hspace{1cm} (A.27)

and we have

$$\phi_j = (a_j r + b_j r^{-1}) \cos \theta.$$  \hspace{1cm} (A.28)

For derivation of $\vec{H}$, we take the gradient of the above equation and $\vec{H}$ can be written for region $j$ as

$$\vec{H}_j(r, \theta) = -\nabla \phi_j = (-a_j + b_j r^{-2}) \cos \theta, (a_j + b_j r^{-2}) \sin \theta + \vec{H}_0.$$  \hspace{1cm} (A.29)

The resulting $f_w$ is

$$f_w = \frac{8b_j \sqrt{(a_j r^2 + b_j)} - 4(a_j - H_0)r^2 \cos^2 \theta + (-2a_j r^4 - 2b_j r^2 + r^4)H_0}{r^5}$$  \hspace{1cm} (A.30)

The results of $f_w$ generated by OpenFOAM is in agreement with the results of Ritter et al.
Appendix B

OpenFOAM Code for Seed Model

In this chapter, we present the structure of the OpenFOAM program (see figure B.1) and C++ code which was generated for the calculation of the dipole-dipole and hydrodynamic interactions for two MDCPs with a seed implant.

![Diagram of OpenFOAM program structure]

**Figure B.1:** Structure of the OpenFOAM program

### B.1 Applications (Solver) for Seed Implant

#### B.1.1 `createFields.H` file
B.1. Applications (Solver) for Seed Implant

The OpenFOAM (Open Field Operation and Manipulation) CFD Toolbox can simulate anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics, electromagnetics and the pricing of financial options. OpenFOAM is produced by OpenCFD Ltd and is freely available and open source, licensed under the GNU General Public Licence.

The core technology of OpenFOAM is a flexible set of efficient C++ modules. These are used to build a wealth of: solvers, to simulate specific problems in engineering mechanics; utilities, to perform pre- and post-processing tasks ranging from simple data manipulations to visualisation and mesh processing; libraries, to create toolboxes that are accessible to the solvers/utilities, such as libraries of physical models.

OpenFOAM is supplied with numerous pre-configured solvers, utilities and libraries and so can be used like any typical simulation package. However, it is open, not only in terms of source code, but also in its structure and hierarchical design, so that its solvers, utilities and libraries are fully extensible.

OpenFOAM uses finite volume numerics to solve systems of partial differential equations ascribed on any 3D unstructured mesh of polyhedral cells. The fluid flow solvers are developed within a robust, implicit, pressure-velocity, iterative solution framework, although alternative techniques are applied to other continuum mechanics solvers. Domain decomposition parallelism is fundamental to the design of OpenFOAM and integrated at a low level so that solvers can generally be developed without the need for any parallel-specific coding.

This file is part of OpenFOAM. OpenFOAM is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

You should have received a copy of the GNU General Public License along with OpenFOAM; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

File Name: createFields.H
Author: mardinoglu@yahoo.com
*/
Info "<"Reading field p"n" << endl;
volScalarField p

    IOobject(
        "p",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
B.1. Applications (Solver) for Seed Implant

volVectorField U

Info "Reading field U

volVectorField U

# include "createPhi.H"

label pRefCell = 0;
scalar pRefValue = 0.0;
setRefCell(p, mesh.solutionDict().subDict("SIMPLE"), pRefCell, pRefValue);

singlePhaseTransportModel laminarTransport(U, phi);
autoPtr<turbulenceModel> turbulence(turbulenceModel::New(U, phi, laminarTransport))

B.1.2 createFields-analytic.H file

/**
 * File Name: createFields_analytic.H
 * Author: mardinoglu@yahoo.com
 */

Info << "region: Reading field T

volScalarField T

Info << "region:Reading physicalProperties

IOdictionary physicalProperties(

Info << "Reading parameter - correct_beta dimensions"

dimensionedScalar correct_beta (physicalProperties.lookup("correct_beta"));
Info << "Correct_beta dimensions is" << correct_beta << endl <<endl;

84
Info<> "Reading parameter - correct fi_r dimensions" << endl; 30
dimensionedScalar correct_fi_r (physicalProperties.lookup("correct_fi_r"));
Info<> "Correct f_i_r dimensions is" << correct_fi_r <<endl;

Info<> "Reading parameter - correct H in OpenFOAM dimensions" << endl; 40
dimensionedScalar correct_H_open (physicalProperties.lookup("correct_H_open"));
Info<> "Correct H_open dimensions is" << correct_H_open <<endl;

Info<> "Reading parameter - correct_fi_open in OpenFOAM dimensions" << endl; 50
dimensionedScalar correct_fi_open (physicalProperties.lookup("correct_fi_open"));
Info<> "Correct fi_open dimensions is" << correct_fi_open <<endl;

Info<> "Reading parameter - magnetic permeability of free space (mu_0)" << endl; 60
dimensionedScalar mu_0 (physicalProperties.lookup("mu_0"));
Info<> "Magnetic permeability of free space (mu_0) is " << mu_0 <<endl;

Info<> "Reading parameter - H_m field angle (theta)" << endl; 70
dimensionedScalar theta (physicalProperties.lookup("theta"));
Info<> "H_m field angle (theta) is " << theta <<endl;

Info<> "Reading parameter - H_m field magnitude (H_0)" << endl; 80
dimensionedScalar H_0 (physicalProperties.lookup("H_0"));
Info <>"H_m field magnitude (H_0) is " <<H_0 <<endl;

Info <>"\nCalculation - (mu_0 H_0) is " <<mu_0*H_0 <<endl;
Info <>"\nReading parameter -magnetic susceptibility of implant at H_0=0" << endl; 90
dimensionedScalar chi_i_0 (physicalProperties.lookup("chi_i_0"));
Info <> "Magnetic susceptibility of implant at H_0=0" <<chi_i_0 <<endl;

Info <> "Reading parameter - saturation magnetization of the implant (M_i_s)" << endl; 100
dimensionedScalar M_i_s (physicalProperties.lookup("M_i_s"));
Info <> "Saturation magnetization of the implant (M_i_s) is ="<<M_i_s<<endl;

Info <> "Calculating parameter - demagnetizing factor of the implant (alpha_i)" "<<endl; 110
dimensionedScalar alpha_i = Foam::min(chi_i_0/(2.0+chi_i_0), M_i_s/(2.0*H_0));
Info <> "Demagnetizing factor (implant_alpha_i) is = " <<alpha_i<<endl;

Info <> "Calculating parameter - susceptibility of implant (chi_i)" <<endl; 120
dimensionedScalar chi_i = 2*(alpha_i/(1.0-alpha_i));
Info <> "susceptibility of implant(implant_chi_i) is = " <<chi_i<<endl;

Info <>"Calculating parameter- - implant_mu" "<< endl;
scalar implant_mu = 1.0+chi_i.value();
Info <>"implant_mu is = " "implant_mu" <<endl;

Info <> "Calculating parameter - Magnetization of the implant (M_i)" "<< endl; 130
dimensionedScalar M_i = 2*alpha_i*H_0;
Info <> "Magnetization of the implant (M_i) is = " "M_i" <<endl;

Info <> "Reading parameter -Magnetic susceptibility of Material at H_0=0" "<< endl; 140
dimensionedScalar chi_fm_p_0 (physicalProperties.lookup("chi_fm_p_0"));
Info <> "Magnetic susceptibility of Material at H_0=0 is " "chi_fm_p_0" <<endl;

Info <> "Reading parameter - saturation magnetization of the material in MDCP" "<< endl; 150
dimensionedScalar M_fm_p_s (physicalProperties.lookup("M_fm_p_s"));
B.1. Applications (Solver) for Seed Implant

Info<< "Saturation magnetization of the material in MDCP is " <<M_fm_p_s << endl;

Info<< "Calculating parameter — demagnetizing factor (alpha_fm_p)" << endl;
dimensionedScalar alpha_fm_p = Foam::min(chi_fm_p_0/(3.0+chi_fm_p_0), M_fm_p_s/(3.0*H_0));
Info<< "Demagnetizing factor (alpha_fm_p) is " <<alpha_fm_p << endl;

Info<< "Calculating parameter — susceptibility of material (chi_fm_p)" << endl;
dimensionedScalar chi_fm_p = 3*(alpha_fm_p/(1.0-alpha_fm_p));
Info<< "Demagnetizing factor (material chi_fm_p) is " <<chi_fm_p << endl;

Info<< "Calculating parameter — Magnetization of the material (M_fm_p)" << endl;
dimensionedScalar M_fm_p = 3*alpha_fm_p*H_0;
Info<< "Magnetization of the material (M_fm_p) is " <<M_fm_p << endl;

Info<< "Reading parameter — Magnetic susceptility of Medium (chi_m)" << endl;
dimensionedScalar chi_m (physicalProperties.lookup("chi_m"));
Info<< "Magnetic susceptility of Medium is " <<chi_m<< endl;

Info<< "Reading parameter — density of the ferromagnetic material in the MDCP is " <<rho_fm_p << endl;

Info<< "Reading parameter — mass fraction of the ferromagnetic material in the MDCP is " <<x_fm_p << endl;

Info<< "Calculating parameter — MDCP density (rho_p)" << endl;
dimensionedScalar rho_p = 1/((x_fm_p/rho_fm_p)+((1-x_fm_p)/rho_pol_p));
Info<< "MDCP density (rho_p) is " <<rho_p<< endl;

Info<< "Reading parameter — radius of implant (R_i)" << endl;
dimensionedScalar R_i (physicalProperties.lookup("R_i"));
Info<< "Radius of implant (R_i) is " <<R_i<< endl;

Info<< "Reading parameter — radius of MDCP (R_p)" << endl;
dimensionedScalar R_p (physicalProperties.lookup("R_p"));
Info<< "Radius of MDCP (R_p) is " <<R_p<< endl;

Info<< "Calculating parameter — volume of MDCP (volume_p)" << endl;
dimensionedScalar volume_p=4.0/3.0 * mathematicalConstant::pi * R_p * R_p* R_p;
Info<< "Volume of MDCP (volume_p) is " <<volume_p<< endl;

Info<< "Reading parameter — parent blood vessel radius (R_pv)" << endl;
dimensionedScalar R_pv (physicalProperties.lookup("R_pv"));
Info<< "Parent blood vessel radius (R_pv) is " <<R_pv<< endl;

Info<< "Calculating parameter — Ratio of Rpv/Rw (Rpv_Rw)" << endl;
dimensionedScalar Rpv_Rw = R_pv/R_i;

86
B.1. Applications (Solver) for Seed Implant

Info "Ratio of Rp_Rw (Rp_Rw) is " << Rp_Rw << endl; Info "Reading parameter — blood viscosity (eta_beta)" << endl; 
dimensionedScalar eta_beta (physicalProperties.lookup("eta_beta")); Info "Blood viscosity (eta_beta) is " << eta_beta << endl; Info "Reading parameter — blood density (rho_beta)" << endl; 
dimensionedScalar rho_beta (physicalProperties.lookup("rho_beta")); Info "Blood density (rho_beta) is " << rho_beta << endl; Info "Reading parameter — Porosity of a cluster of MDCP (epsilon_p)" << endl; 
dimensionedScalar epsilon_p (physicalProperties.lookup("epsilon_p")); Info "Porosity of a cluster of MDCP (epsilon_p) is " << epsilon_p << endl; Info "Reading parameter — radius of MDCP for Cregg (R_p_Cregg)" << endl; 
dimensionedScalar R_p_Cregg (physicalProperties.lookup("R_p_Cregg")); Info "Radius of MDCP for Cregg (R_p_Cregg) is " << R_p_Cregg << endl; Info "Calculating parameter — volume of MDCP for Cregg (volume_p_Cregg)" << endl; 
dimensionedScalar volume_p_Cregg=4.0/3.0 * mathematicalConstant::pi * pow(R_p_Cregg,3); Info "Radius of MDCP for Cregg (volume_p_Cregg) is " << volume_p_Cregg << endl; Info "Reading parameter — average inlet velocity (u_0)" << endl; 
dimensionedScalar u_0 (physicalProperties.lookup("u_0")); Info "Average inlet velocity (u_0) is " << u_0 << endl; Info "Calculating parameter — Magnetic velocity" << endl; 
dimensionedScalar velocity_m = (2.0/9.0)*(R_p*R_p/R_i)*(mu_0/eta_beta) 
* (1-epsilon_p)*omega_fm_p* M_fm_p* M_i; Info "Magnetic velocity (velocity_m) is = " << velocity_m << endl; Info "Calculating parameter — Magnetic velocity for Cregg" << endl; 
dimensionedScalar velocity_m_Cregg = (2.0/9.0)*(R_p_Cregg*R_p_Cregg/R_i) 
* (mu_0/eta_beta)* (1-epsilon_p)*omega_fm_p* M_fm_p* M_i; Info "Magnetic velocity for Cregg is = " << velocity_m_Cregg << endl; vector H_0_vector = vector(H_0.value()*cos(convertToRad*theta).value(), 
H_0.value()*sin(convertToRad*theta).value(), 0.0); Info "\n\n@ Creating fields for space_region.\n" << endl; 
volVectorField space_H_0( 
 IOobject("H_0", runTime.constant()), 
 mesh, 
 IOobject::NO_READ, 
 IOobject::AUTO_WRITE), 
 mesh, 
 dimensionedVector("0", dimless, H_0_vector) ); space_H_0.write(); 

volVectorField space_H_0_cal ( 
 IOobject("H_0_cal", runTime.constant()), 
 mesh, IOobject::NO_READ,

87
B.1. Applications (Solver) for Seed Implant

IOobject::AUTO_WRITE),
-fvc::grad(T_0)
);

space_H_0_cal.write();

B.1.3 readtwoRegionDict.H file

/**
 * File Name: readtwoRegionDict.H
 * Author: mardinoglu@yahoo.com
 */
Info <<"\n\n@ Reading twoRegionDict
" <<endl;
IOdictionary twoRegionDict
(IOobject
("twoRegionDict",
runTime.system(),
runTime,
IOobject::MUST_READ,
IOobject::NO_WRITE)
);

Info<<"Reading parameter - - space_mu" << endl;
scalar space_mu = readScalar(twoRegionDict.lookup("space_mu"));
Info <<"Space_mu = " <<space_mu <<endl;

20
Info<<"Reading parameter - - distance between center and point on X axis " << endl;
scalar shift_x = readScalar(twoRegionDict.lookup("shift_x"));
Info <<"Distance between center and point on X axis = " <<shift_x<<endl;

Info<<"Reading parameter - - distance between center and point on Y axis" << endl;
scalar shift_y = readScalar(twoRegionDict.lookup("shift_y"));
Info <<"Distance between center and point on Y axis = " <<shift_y<<endl;

30
Info<<"Reading parameter - - position of particle one on X axis " << endl;
scalar x_1_original = readScalar(twoRegionDict.lookup("x_1_original"));
Info <<"Position of particle one on X axis= " <<x_1_original<<endl;

Info<<"Reading parameter - - position of particle one on Y axis " << endl;
scalar y_1_original = readScalar(twoRegionDict.lookup("y_1_original"));
Info <<"Position of particle one on Y axis= " <<y_1_original<<endl;

40
Info<<"Reading parameter - - position of particle two(reference) on X axis " << endl;
scalar x_2 = readScalar(twoRegionDict.lookup("x_2"));
Info <<"Position of particle two on X axis= " <<x_2<<endl;

Info<<"Reading parameter - - position of particle two(reference) on Y axis " << endl;
scalar y_2 = readScalar(twoRegionDict.lookup("y_2"));
Info <<"Position of particle one on Y axis= " <<y_2<<endl;
B.1.4 interactionFoam.C file

/**
 * File Name: interactionFoam.C
 * Author: mardinoglu@yahoo.com
*/
#include "fvCFD.H"
#include "incompressible/singlePhaseTransportModel/singlePhaseTransportModel.H"
#include "incompressible/turbulenceModel/turbulenceModel.H"
#include "typeInfo.H"
#include "OFstream.H"
#include "IOmanip.H"
#include "mathematicalConstants.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
int main(int argc, char *argv[])
{
    const scalar convertToRad = mathematicalConstant::pi/180.0;

    # include "setRootCase.H"
    # include "createTime.H"
    # include "createMesh.H"
    # include "createFields.H"
    # include "createFields_analytic.H"
    # include "initContinuityErrs.H"
    # include "readtwoRegionDict.H"

    Info << "Evaluating analytical solution for Aviles Case" << endl;
    volVectorField centres1 = T.mesh().C();
    volScalarField radius1 = sqrt((centres1.component(vector::X)*centres1.component(vector::X))
        +(centres1.component(vector::Y)*centres1.component(vector::Y)));

    scalar const_a2 = (-H_0.value()*(1/(1-((implant_mu-space_mu)/(space_mu+implant_mu))*(0.0000001))));
    // 0.000000001 will be changed according to the blockmesh file.
    scalar const_b2 = -((implant_mu-space_mu)/(space_mu+implant_mu))*const_a2;

    volScalarField theta1 = acos((centres1 & vector(1,0,0))/magCentres1);
    volScalarField Tcls_total_space(   
        IOobject(    
            "T_an_total",
            runTime.timeName(),
            mesh,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE),
        (const_a2*centres1.component(vector::X)+const_b2*centres1.component(vector::X)/
         (radius1*radius1));
    volScalarField T_an_total_hand(    
        IOobject("T_an_total_hand",    
            runTime.timeName(),
            mesh,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE),
        correct_H_open*((-const_a2+(const_b2*centres1.component(vector::X)*
           centres1.component(vector::X)−centres1.component(vector::Y))*
           centres1.component(vector::Y)−centres1.component(vector::Y)*
           centres1.component(vector::Y));
centre1.component(vector::Y)/(radius1*radius1*radius1*radius1)) *vector(1.0,0,0)+(2*const_b2*centre1.component(vector::X)*
centre1.component(vector::Y)/(radius1*radius1*radius1*radius1))
);

volScalarField mag_H_space_an_total_hand(
    IOobject("H_mag_an_total_hand",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::AUTO_WRITE),
    mag(H_space_an_total_hand)
);

volVectorField f_i_space_total_analytical(
    IOobject("f_i_vector_analytical",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::AUTO_WRITE),
    (correct_fi_open*(3.0/(2.0*R_i.value()))*(1.0/4.0)*alpha_fm_p*mu_0*
const_b2* 
(3.0*const_a2*centre1.component(vector::X)*centre1.component(vector::Y)−
3.0*const_a2*centre1.component(vector::X)*centre1.component(vector::Y)−
const_b2)/((radius1*radius1*radius1*radius1)*vector(1.0,0)+(4.0*((centre1.component(vector::Y)*(3.0*const_a2*
centre1.component(vector::X))*centre1.component(vector::Y)−centre1.component(vector::Y))*centre1.component(vector::Y)−
const_b2))/
(radius1*radius1*radius1*radius1)*vector(0,1,0))
);

Info << "Calculating the magnitude of f_i_analytical in space region" << endl;
volScalarField mag_f_i_space_total_analytical(
    IOobject("f_i_mag_analytical",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::AUTO_WRITE),
    mag(f_i_space_total_analytical)
);

Info << "Calculating the velocity of particle by using the analytical fw" << endl;
volVectorField velocity_particle_analytical(
    IOobject("velocity_p_an",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::AUTO_WRITE),
    velocity_m_fi_space_total_analytical*R_i*(1/(M_i*mu_0*H_0*u_0))*(4/(3*alpha_fm_p))
);

volScalarField mag_velocity_particle_analytical(
    IOobject("mag_velocity_p_an",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::AUTO_WRITE),
    mag(velocity_particle_analytical)
B.1. Applications (Solver) for Seed Implant

Info \<\<"\nCALCULATING THE VELOCITY OF PARTICLE WITH LANGEVIN FUNCTIONS"\< endl;
Info\< "\nCalculating parameter - beta for the Langevin Function" \<endl;
volScalarField beta_Cregg(
    IOobject(
        "beta_Cregg",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE),
    correct_beta*(omega_fm_p*volume_p_Cregg*M_fm_p_s*mag(mu_0*H_space_an_total_hand)) 120
    /(1.38e-23 * 309.5)
    );
Info\< "\nCalculating parameter - Langevin Factor for the Langevin Function" \<endl;
volScalarField Langevin_Cregg(
    IOobject(
        "Langevin_Cregg",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE),
    (1.0/Foam::tanh(beta_Cregg) - 1.0/beta_Cregg)
    );
Info \<\<"\nCalculating the velocity of particle by using hand calculation for H " \<endl;
volVectorField velocity_particle_an_Cregg(
    IOobject ( "velocity_p_an_Cregg",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE),
    velocity_m_Cregg*Langevin_Cregg*L_i_space_total_analytical*R_i* 140
    (1/(M_i*mu_0*H_0*u_0))*(4/(3*alpha_fm_p))
    );
volScalarField mag_velocity_particle_an_Cregg(
    IOobject ( "mag_velocity_p_an_Cregg",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE),
    mag(velocity_particle_an_Cregg)
    );
volVectorField magmoment_Cregg(
    IOobject ( "magmoment_Cregg",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE),
    (omega_fm_p*volume_p_Cregg*M_fm_p_s*Langevin_Cregg*(mu_0*H_space_an_total_hand) 160
    /(mu_0*mag_H_space_an_total_hand))
    );
Info\< "\n\nwrite. . . . . .END" \<endl;
Info\< "\nPART1\n" \<endl;
// ************************************************************************* //
ofstream myfile;
myfile.open("Pjinteraction.txt");
B.1. Applications (Solver) for Seed Implant

// For Blood velocity
Info<< "\nStarting time loop\n" << endl;
for (runTime++; !runTime.end(); runTime++)
{
    Info<< "Time = " << runTime.timeName() << nl << endl;
    // include "readSIMPLEControls.H"
    p.storePrevIter();
    // Pressure-velocity SIMPLE corrector
    {
        // Momentum predictor
        tmp<fvVectorMatrix> UEqn
        {
            fvm::div(phi, U)
            + turbulence->divR(U)
        };
        UEqn().relax();
        solve(UEqn() == -fvc::grad(p));
        p.boundaryField().updateCoeffs();
        volScalarField AU = UEqn().A();
        U = UEqn().H()/AU;
        UEqn.clear();
        phi = fvc::interpolate(U) & mesh.Sf();
        adjustPhi(phi, U, p);
        // Non-orthogonal pressure corrector loop
        for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
        {
            fvScalarMatrix pEqn
            {
                fvm::laplacian(1.0/AU, p) == fvc::div(phi)
            };
            pEqn.setReference(pRefCell, pRefValue);
            pEqn.solve();
            if (nonOrth == nNonOrthCorr)
            {
                phi -= pEqn.flux();
            }
        }
    }
    // Explicitly relax pressure for momentum corrector
    p.relax();
    // Momentum corrector
    U -= fvc::grad(p)/AU;
    U.correctBoundaryConditions();
}

runTime.write();
Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
<< " ClockTime = " << runTime.elapsedClockTime() << " s"
<< nl << endl;
// For particle Tracking
// For position of the particle see tworegionDict file.
scalar distance= mag(y_2-y_1_original); //distance between the particles
scalar distance_original= mag(y_2-y_1_original); //distance between the particles
scalar x_1=x_1_original;
scalar y_1=y_1_original;
scalar x_2_original=x_2;
scalar y_2_original=y_2;

if (runTime.time.value() > 3.0)
{
    for (runTime++; !runTime.end(); runTime++)
    {
        Info<< "Time = " << runTime.timeName() << nl << endl;
    }
}

// Info << "\n\n\nEvaluating solution for magnetic interaction on Mikkelsen’s paper" << endl;
// In this part the effect of dipole interaction is calculated
#include "Time.H"
#include "I0streams.H"
// save the output of the file
myfile<<runTime.timeName()<<"\t"<<x_1<<"\t"<<y_1<<"\t"<<x_2<<"\t"<<y_2<<"\t"
<<x_1_original<<"\t"<<y_1_original<<"\t"<<x_2_original<<"\t"<<y_2_original
<<"\t"<<distance<<"\t"<<distance_original<<"\n";

vector probePoint_one_original(x_1_original,y_1_original,0);
label probeCell_one_original = mesh.findCell(probePoint_one_original);
volTensorField gradU = fvc::grad(U);
vector cellCentre_one_original = mesh.C()[probeCell_one_original];
vector U_one_original_int = U[probeCell_one_original]+((probePoint_one_original
-cellCentre_one_original)&gradU[probeCell_one_original]);

vector probePoint_one(x_1,y_1,0);
label probeCell_one = mesh.findCell(probePoint_one);
vector cellCentre_one = mesh.C()[probeCell_one];
vector U_one_int = U[probeCell_one] + ((probePoint_one
-cellCentre_one) & gradU[probeCell_one]);

vector probePoint_two_original(x_2_original,y_2_original,0);
label probeCell_two_original = mesh.findCell(probePoint_two_original);
vector cellCentre_two_original = mesh.C()[probeCell_two_original];
vector U_two_original_int = U[probeCell_two_original]+((probePoint_two_original
-cellCentre_two_original)&gradU[probeCell_two_original]);

vector probePoint_two(x_2,y_2,0);
label probeCell_two = mesh.findCell(probePoint_two);
vector cellCentre_two = mesh.C()[probeCell_two];
vector U_two_int=U[probeCell_two]+((probePoint_two-cellCentre_two)&gradU[probeCell_two]);

volTensorField gradU_cregg = fvc::grad(velocity_particle_an_Cregg);
vector U_cregg_one_original_int=velocity_particle_an_Cregg[probeCell_one_original]
+((probePoint_one_original-cellCentre_one_original)&gradU_cregg[probeCell_one_original]);

vector U_cregg_one_int = velocity_particle_an_Cregg[probeCell_one]+
((probePoint_one-cellCentre_one) & gradU_cregg[probeCell_one]);

vector U_cregg_two_original_int=velocity_particle_an_Cregg[probeCell_two_original]
+((probePoint_two_original-cellCentre_two_original)&gradU_cregg[probeCell_two_original]);

vector U_cregg_two_int=velocity_particle_an_Cregg[probeCell_two]+
((probePoint_two-cellCentre_two) & gradU_cregg[probeCell_two]);
volTensorField gradH=fvc::grad(H(space_an_total_hand));

vector H_r1_vector_int=H(space_an_total_hand)[probeCell_one]+((probePoint_one-cellCentre_one)&gradH[probeCell_one]);
vector H_r2_vector_int=H(space_an_total_hand)[probeCell_two]+((probePoint_two-cellCentre_two)&gradH[probeCell_two]);

//Info<< "Creating vector- - r vectors for each particle " << endl;
vector r_1_vector = vector(x_1,y_1,0);
vector r_1_original_vector = vector(x_1_original,y_1_original,0);
vector r_2_vector = vector(x_2,y_2,0);
vector r_2_original_vector = vector(x_2_original,y_2_original,0);

//Info<< "Calculating- - distance between the center of particles" << endl;
Info <<"Parameter - - (r_1_vector)= " << (r_1_vector) <<" Parameter - - (r_2_vector)= " << (r_2_vector) <<endl;
distance=mag(r_2_vector-r_1_vector);
distance_original=mag(r_2_original_vector-r_1_original_vector);
Info <<"Parameter -DISTANCE BETWEEN THE PARTICLES >> 0.1 = " <<distance <<endl;

//Info<< "Creating vector- - r position vectors for all space region " << endl;
volVectorField r_positions= T.mesh().C();

//Info<< "Creating vector field- - r_1 constant field for particle 1 position " << endl;
volVectorField r_1_constant_field(IOobject("r_1",
runTime.timeName()),
  mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
  mesh,
dimensionedVector("0", dimless, r_1_vector));

//Info<< "Creating vector field- - B_r_1 constant field for particle one magnetic Flux " << endl;
volVectorField B_r_1_constant_field(IOobject("B_r_1",
runTime.timeName()),
  mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
  mesh,
dimensionedVector("0", dimless, (mu_0.value()*H_r1_vector_int)));
B.1. Applications (Solver) for Seed Implant

```
IOobject("B_r_2",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
dimensionedVector("0", dimless, (mu_0.value()*H_r2.vector.int))
);

//Info << "/n'Evaluating solution for particle one because of particle 2; called (dB_2)" << endl;
volVectorField F_int_term11(  
IOobject ( "F_int_1_dB2",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
(1.0/3.0)*((mu_0*mu_0*M_fm_p_s*Langevin_Cregg/(mu_0*mag.H_space_an_total_hand))*
pow(R_p_Cregg,3)/pow((r_1_vector−r_2_vector).3)*((3.0*(B_r_2_constant_field&
(r_positions−r_2_constant_field)((r_positions−r_2_constant_field))/
pow(mag(r_positions−r_2_constant_field),2))−(B_r_2_constant_field))*pow((1.0/R_i),3))
);

volScalarField beta_Cregg_par_one(  
IOobject(  
"beta_Cregg_one",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
correct_beta*(omega_fm_p*volume_p_Cregg*M_fm_p_s*(mag(mu_0*H_space_an_total_hand)  
+mag(F_int_term11))/(1.38e−23 * 309.5))
);

volVectorField magmoment_Cregg_par_one(  
IOobject ( "magmoment_Cregg_one",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
(1.0/Foam::tanh(beta_Cregg_par_one)−1.0/beta_Cregg_par_one)
);

volVectorField F_int_par_one_new(  
IOobject ( "F_int_1_new",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
```

95
B.1. Applications (Solver) for Seed Implant

```cpp
(fvc::grad((magmoment_Cregg_par_one)&(mu_0*H_space_an_total_hand+F_int_term11)))

);  //vector probePoint_one(x_1,y_1,0);
//label probeCell_one = mesh.findCell(probePoint_one);
volTensorField gradF_int_one = fvc::grad(F_int_par_one_new);
//vector cellCentre_one = mesh.C()[probeCell_one];
vector F_int_par_one_vector_new = F_int_par_one_new[probeCell_one] +
((probePoint_one-cellCentre_one)&gradF_int_one[probeCell_one]);
//Info << "Evaluating solution for particle two because of particle 1; called (dB_1)" << endl;
volVectorField F_int_term21(  
IOobject ("F_int_2_dB1",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
(0.3333333)*(mu_0*M_fm_p*s*Langevin_Cregg/(mu_0*mag_H_space_an_total_hand)*)
pow(R_p_Cregg_3)/pow(mag(r_2_vector-r_1_vector),3)*((3.0*(B_r_1_constant_field&
(r_positions-r_1_constant_field))*(r_positions-r_1_constant_field)/
pow(mag(r_positions-r_1_constant_field),2))-(B_r_1_constant_field))*pow((1.0/R_i),3)
);
volScalarField beta_Cregg_par_two(  
IOobject("beta_Cregg_two",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
correct_beta*(omega_fm_p*volume_p_Cregg*M_fm_p_s*(mag(mu_0*H_space_an_total_hand) +mag(F_int_term21)))/(1.38e-23 * 309.5)
);
Info<< "Calculating parameter - Langevin Factor for the Langevin Function" << endl;
volScalarField Langevin_Cregg_par_two(  
IOobject("Langevin_Cregg_two",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
(1.0/Foam::tanh(beta_Cregg_par_two) - 1.0/beta_Cregg_par_two)
);
volVectorField magmoment_Cregg_par_two(  
IOobject("magemoment_Cregg_two",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
(omega_fm_p*volume_p_Cregg*M_fm_p_s*Langevin_Cregg_par_one*
((mu_0*H_space_an_total_hand)+F_int_term21))/((mu_0*mag_H_space_an_total_hand) +mag(F_int_term21))
);
volVectorField F_int_par_two_new(  
IOobject("F_int_2_new",
runTime.timeName(),
mesh,
IOobject::NO_READ,
IOobject::AUTO_WRITE),
```
B.1. Applications (Solver) for Seed Implant

(fvc::grad((magmoment_Criegg_par_two)&(mu_0*H_space_an_total_hand+F_int_term21)))
);
//vector probePoint_two(x_2,y_2,0);
//label probeCell_two = mesh.findCell(probePoint_two);
volTensorField gradF_int_two = fvc::grad(F_int_par_two_new);
//vector cellCentre_two = mesh.C()[probeCell_two];
vector F_int_par_two_vector_new = F_int_par_two_new[probeCell_two] +
((probePoint_two - cellCentre_two)&gradF_int_two[probeCell_two]);

//Info << "Calculating magnetic velocities with interaction for particle one and two " << endl;
dimensionedVector velocity_int_par_one_new=(F_int_par_one_vector_new*(1.0/(6.0*R_i.value() *
mathematicalConstant::pi*eta_beta*R_p_Criegg.value())*u_0)));
dimensionedVector velocity_int_par_two_new=(F_int_par_two_vector_new*(1.0/(6.0*R_i.value() *
mathematicalConstant::pi*eta_beta*R_p_Criegg.value())*u_0)));

// 0.005 was the time step
// 0.0025 was the time step
// 0.00125 is the time step
vector particle1_ccs_original=vector(x_1_original,y_1_original,0)+0.001*
(U_cregg_one_original_int+U_one_original_int); //with the seed effect
vector particle2_ccs_original=vector(x_2_original,y_2_original,0)+0.001*
(U_cregg_two_original_int+U_two_original_int); //with the seed effect

vector particle1_ccs=vector(x_1,y_1,0)+0.001*(velocity_int_par_one_new.value())+U_one_int);
vector particle2_ccs=vector(x_2,y_2,0)+0.001*(velocity_int_par_two_new.value())+U_two_int);

// Calculates the new positions of the particles(Creating streamlines)
x_1_original=particle1_ccs_original.x();
y_1_original=particle1_ccs_original.y();
x_2_original=particle2_ccs_original.x();
y_2_original=particle2_ccs_original.y();

runTime.write();
Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
<< " ClockTime = " << runTime.elapsedClockTime() << " s"
<< nl << endl;
}
}
myfile.close();
Info<< "End\n" << endl;

return(0);
B.2 Run (Case) for Seed Implant

B.2.1 0 (initial conditions file)

B.2.1.1 epsilon file

```plaintext
FoamFile
{
    version     2.0;
    format      ascii;
    root         "";
    case         "";
    instance     "";
    local        "";

    class        volScalarField;
    object       epsilon;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions                      [0 2 -3 0 0 0];
internalField                  uniform 14.855;
boundaryField
{
    inlet
    {
        type       fixedValue;
        value      uniform 14.855;
    }
    outlet
    {
        type       zeroGradient;
    }
    top
    {
        type       zeroGradient;
    }
    bottom
    {
        type       zeroGradient;
    }
    seed_0
    {
        type       zeroGradient;
    }
    defaultFaces
    {
        type       empty;
    }
}```
B.2. Run (Case) for Seed Implant

// ************************************************************************* //

B.2.1.2 k file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class volScalarField;
    object k;
}
// ************************************************************************* //

dimensions [0 2 -2 0 0 0 0];
internalField uniform 0.375;
boundaryField
{
    inlet
    {
        type fixedValue;
        value uniform 0.375;
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type zeroGradient;
    }
    bottom
    {
        type zeroGradient;
    }
    seed_0
    {
        type zeroGradient;
    }
    defaultFaces
    {
        type empty;
    }
}
// ************************************************************************* //
\textbf{B.2.1.3 \textit{nuTilda file}}

\begin{verbatim}
FoamFile
{

    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class volScalarField;
    object nuTilda;
}

// ************************************************************************* //

dimensions [0 2 -1 0 0 0 0];
internalField uniform 0;
boundaryField
{
    inlet
    {
        type fixedValue;
        value uniform 0;
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type zeroGradient;
    }
    bottom
    {
        type zeroGradient;
    }
    seed_0
    {
        type zeroGradient;
    }
    defaultFaces
    {
        type empty;
    }
}
// ************************************************************************* //
\end{verbatim}

\textbf{B.2.1.4 \textit{p file}}

\begin{verbatim}
FoamFile
{

    version 2.0;
    format ascii;
    root "";
    case "";
}

100
\end{verbatim}
B.2. Run (Case) for Seed Implant

instance "";
local "";
class volScalarField;
object p;

// ************************************************************************* //
dimensions [0 2 -2 0 0 0 0];
internalField uniform 0;
boundaryField
{
  inlet
  {
    type zeroGradient;
  }
  outlet
  {
    type fixedValue;
    value uniform 0;
  }
  top
  {
    type zeroGradient;
  }
  bottom
  {
    type zeroGradient;
  }
  seed_0
  {
    type zeroGradient;
  }
defaultFaces
  {
    type empty;
  }
}
// ************************************************************************* //

B.2.1.5 phi file

FoamFile
{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
class surfaceScalarField;
object phi;
}
// ************************************************************************* //
dimensions [0 3 -1 0 0 0 0];
internalField uniform 0;
B.2. Run (Case) for Seed Implant

```
boundaryField
{
    inlet
    {
        type calculated;
        value uniform 90;
    }
    outlet
    {
        type calculated;
        value uniform 90;
    }
    top
    {
        type calculated;
        value uniform 90;
    }
    bottom
    {
        type calculated;
        value uniform 90;
    }
    seed_0
    {
        type calculated;
        value uniform 90;
    }
    defaultFaces
    {
        type empty;
    }
}

// ************************************************************************* //

B.2.1.6 R file

```

```
FoamFile
{
    version 2.0;
    format ascii;
    root ";
    case ";
    instance ";
    local ";
    class volTensorField;
    object R;
}

dimensions [0 2 -2 0 0 0 0 0 0];
internalField uniform (0 0 0 0 0 0 0 0 0);
boundaryField
{
    inlet
    {
```
B.2. Run (Case) for Seed Implant

```plaintext
type fixedValue;
value uniform (0 0 0 0 0 0 0 0 0);
}
outlet {
  type zeroGradient;
}
top {
  type zeroGradient;
}
bottom {
  type zeroGradient;
}
seed_0 {
  type zeroGradient;
}
defaultFaces {
  type empty;
}

// ************************************************************************* //

B.2.1.7 T file

FoamFile
{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class volScalarField;
  object T;
}
// ************************************************************************* //

dimensions [0 0 0 1 0 0 0];
internalField uniform 0;
boundaryField {
  inlet {
    type fixedValue;
    value uniform 0.0;
  }
  outlet {
    type fixedValue;
    value uniform 0.0;
  }
}
B.2. Run (Case) for Seed Implant

top
{
    type fixedValue;
    value uniform 0.0;
}
bottom
{
    type fixedValue;
    value uniform 0.0;
}
seed_0
{
    type fixedGradient;
    gradient uniform 0.0;
}
defaultFaces
{
    type empty;
}

// ************************************************************************* //

B.2.1.8 U blood velocity file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class volVectorField;
    object U;
}

// ************************************************************************* //

dimensions [0 1 -1 0 0 0 0];
internalField uniform (0 0 0);
boundaryField
{
    inlet
    {
        type fixedValue;
        value uniform (1 0 0);
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type fixedValue;
        value uniform (0 0 0);
    }
}
B.2. Run (Case) for Seed Implant

bottom
{
    type fixedValue;
    value uniform (0 0 0);
}
seed_0
{
    type fixedValue;
    value uniform (0 0 0);
}
defaultFaces
{
    type empty;
}

// *************************************************************** //

B.2.2 constant

B.2.2.1 Polymesh File (blockMeshDict) file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class dictionary;
    object blockMeshDict;
}

// *************************************************** //
convertToMeters 1.0;
vertices 40 {
    (−20.000000 −21.000000 0.000000)
    (−20.000000 −1.414214 0.000000)
    (−20.000000 1.414214 0.000000)
    (−20.000000 21.000000 0.000000)
    (−1.414214 −21.000000 0.000000)
    (−1.414214 −1.414214 0.000000)
    (−1.414214 1.414214 0.000000)
    (−1.414214 21.000000 0.000000)
    (0.707107 −0.707107 0.000000)
    (0.707107 −0.707107 0.000000)
    (0.707107 0.707107 0.000000)
    (0.707107 0.707107 0.000000)
    (1.414214 −21.000000 0.000000)
    (1.414214 −1.414214 0.000000)
    (1.414214 1.414214 0.000000)
    (1.414214 21.000000 0.000000)
    (20.000000 −21.000000 0.000000)
    (20.000000 −1.414214 0.000000)
}
B.2. Run (Case) for Seed Implant

patches
(
);

B.2.2.2 physical properties file

FoamFile
{
  version 2.0;
  format ascii;
  root ""
  case ""
  instance ""
  local ""
  class dictionary;
  object physicalProperties;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
// kg m s K ml A cd
theta theta [ 0 0 0 0 0 0 0] 0.0;       // H_m field angle (in degrees)
//H H_0 0 [ 0 -1 0 0 0 1 0] 159154.4302045; // magnitude of H_m field for mu_0*H_0=2.0T
//H H_0 0 [ 0 -1 0 0 0 1 0] 143239.5441636; // magnitude of H_m field for mu_0*H_0=1.8T
//H H_0 0 [ 0 -1 0 0 0 1 0] 111408.4611432; // magnitude of H_m field for mu_0*H_0=1.6T
//H H_0 0 [ 0 -1 0 0 0 1 0] 954929.6581227; // magnitude of H_m field for mu_0*H_0=1.2T
//H H_0 0 [ 0 -1 0 0 0 1 0] 795774.7151023; // magnitude of H_m field for mu_0*H_0=1.0T
//H H_0 0 [ 0 -1 0 0 0 1 0] 636619.7720818; // magnitude of H_m field for mu_0*H_0=0.8T
//H H_0 0 [ 0 -1 0 0 0 1 0] 596830.935;     // magnitude of H_m field for mu_0*H_0=0.75T
//H H_0 0 [ 0 -1 0 0 0 1 0] 557042.2000000; // magnitude of H_m field for mu_0*H_0=0.7T
//H H_0 0 [ 0 -1 0 0 0 1 0] 517253.564;     // magnitude of H_m field for mu_0*H_0=0.65T
//H H_0 0 [ 0 -1 0 0 0 1 0] 477164.8290614; // magnitude of H_m field for mu_0*H_0=0.6T
//H H_0 0 [ 0 -1 0 0 0 1 0] 437676.092;     // magnitude of H_m field for mu_0*H_0=0.55T
//H H_0 0 [ 0 -1 0 0 0 1 0] 397887.355511; // magnitude of H_m field for mu_0*H_0=0.5T
//H H_0 0 [ 0 -1 0 0 0 1 0] 358098.62;      // magnitude of H_m field for mu_0*H_0=0.45T
//H H_0 0 [ 0 -1 0 0 0 1 0] 318309.8660109; // magnitude of H_m field for mu_0*H_0=0.4T
//H H_0 0 [ 0 -1 0 0 0 1 0] 278521.13501;  // magnitude of H_m field for mu_0*H_0=0.35T
//H H_0 0 [ 0 -1 0 0 0 1 0] 238732.4;       // magnitude of H_m field for mu_0*H_0=0.3T
//H H_0 0 [ 0 -1 0 0 0 1 0] 198943.7280305; // magnitude of H_m field for mu_0*H_0=0.25T
//H H_0 0 [ 0 -1 0 0 0 1 0] 159154.94130205; // magnitude of H_m field for mu_0*H_0=0.2T
//H H_0 0 [ 0 -1 0 0 0 1 0] 119366.2072653; // magnitude of H_m field for mu_0*H_0=0.15T
//H H_0 0 [ 0 -1 0 0 0 1 0] 795777.4175102; // magnitude of H_m field for mu_0*H_0=0.1T
//H H_0 0 [ 0 -1 0 0 0 1 0] 397887.355511; // magnitude of H_m field for mu_0*H_0=0.05T
//H H_0 0 [ 0 -1 0 0 0 1 0] 198943.7280305; // magnitude of H_m field for mu_0*H_0=0.025T
//H H_0 0 [ 0 -1 0 0 0 1 0] 0.0;            // magnitude of H_m field for mu_0*H_0=0.0T
mu_0 mu_0 [ 1 1 -2 0 0 -2 0] 0.0000001256637062; // magnetic permeability of free space
R_i R_i [ 0 1 0 0 0 0 0] 1000.00e-09; // implant radius
chi_fm_p_0 chi_fm_p_0 [ 0 0 0 0 0 0 0] 1000; // magnetic susceptibility of material at H_0=0
chi_i_0 chi_i_0 [ 0 0 0 0 0 0 0] 1000; // magnetic susceptibility of implant at H_0=0
chi_m chi_m [ 0 0 0 0 0 0 0] 0; // magnetic susceptibility of medium
M_i_s M_i_s [ 0 -1 0 0 0 1 0] 1397000; // Saturation magnetization of the implant
M_fm_p_s M_fm_p_s [ 0 -1 0 0 0 1 0] 17350000; // Saturation magnetization of the mat. in MDCP
rho_beta rho_beta [ 1 -3 0 0 0 0 0] 1040; // blood density
B.2. Run (Case) for Seed Implant

\[
\begin{align*}
\text{rho}_{\text{fm}} & \quad \text{rho}_{\text{fm}} \quad [1 -3 0 0 0 0 0] \quad 7850; \quad // \text{density of in MDCP} \\
\text{rho}_{\text{pol}} & \quad \text{rho}_{\text{pol}} \quad [1 -3 0 0 0 0 0] \quad 950; \\
x_{\text{fm}} & \quad x_{\text{fm}} \quad [0 0 0 0 0 0 0] \quad 0.4; \quad // \text{mass fraction of material in MDCP} \\
R_p & \quad R_p \quad [0 1 0 0 0 0 0] \quad 50.0e^{-9}; \quad // \text{MDCP radius} \\
R_{pv} & \quad R_{pv} \quad [0 1 0 0 0 0 0] \quad 0.000021; \quad // \text{parent blood vessel radius} \\
u_0 & \quad u_0 \quad [0 1 -1 0 0 0 0] \quad 0.001; \quad // \text{average inlet velocity} \\
\eta_{\beta} & \quad \eta_{\beta} \quad [1 -1 -1 0 0 0 0] \quad 0.002; \quad // \text{blood viscosity (Pa s)} \\
\epsilon & \quad \epsilon \quad [0 0 0 0 0 0 0] \quad 0; \quad // \text{blood viscosity (Pa s)} \\
R_{p,Cregg} & \quad R_{p,Cregg} \quad [0 1 0 0 0 0 0] \quad 50.0e^{-9}; \quad // \text{MDCP radius}
\end{align*}
\]

### B.2.2.3 transport properties file

FoamFile
\{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class dictionary;
    object transportProperties;
\}

// ************************************************************************* //
transportModel Newtonian;
mu \[0 2 -1 0 0 0 0\] 1.9231e−06;
// ************************************************************************* //

### B.2.2.4 turbulence properties file

FoamFile
\{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class dictionary;
    object turbulenceProperties;
\}

// ************************************************************************* //
turbulenceModel laminar;
turbulence off;
laminarCoeffs
\{
\}
kEpsilonCoeffs
\{
    C_{mu} \quad 0.09;
    \quad C_1 \quad 1.44;
    \quad C_2 \quad 1.92;
\}

108
B.2. Run (Case) for Seed Implant

alphaEps alphaEps [0 0 0 0 0 0] 0.76923;

B.2.3 system

B.2.3.1 controlDict file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class dictionary;
    object controlDict;
}

startFrom startTime;
startTime 0;
stopAt endTime;
endTime 100.0;
deltaT 0.001;
writeControl timeStep;
writeInterval 1500;
purgeWrite 0;
writeFormat ascii;
writePrecision 6;
writeCompression compressed;
timeFormat general;
timePrecision 6;
runTimeModifiable yes;

B.2.3.2 fvSchemes file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class dictionary;
    object fvSchemes;
}

ddtSchemes
{
B.2. Run (Case) for Seed Implant

default Euler;

}  

gradSchemes
{
    default Gauss linear;
    grad(p) Gauss linear;
    grad(U) Gauss linear;
}

divSchemes
{
    default none;
    div(phi,U) Gauss upwind;
    div(phi,k) Gauss upwind;
    div(phi,epsilon) Gauss upwind;
    div(phi,R) Gauss upwind;
    div(R) Gauss linear;
    div(phi,nuTilda) Gauss upwind;
    div((nuEff*dev(grad(U).T()))) Gauss linear;
}

laplacianSchemes
{
    default none;
    laplacian(nuEff,U) Gauss linear corrected;
    laplacian(1|A(U),p) Gauss linear corrected;
    laplacian(1|A(U),p) Gauss linear limited 1;
    laplacian(DkEff,k) Gauss linear corrected;
    laplacian(DepsilonEff,epsilon) Gauss linear corrected;
    laplacian(DREff,R) Gauss linear corrected;
    laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
}

interpolationSchemes
{
    default linear;
    interpolate(U) linear;
}

snGradSchemes
{
    default corrected;
}

fluxRequired
{
    default no;
    p;
}  

B.2.3.3  fvSolution file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
}
B.2. Run (Case) for Seed Implant

```plaintext
local "";
class dictionary;
object fvSolution;

// **********************************************
solvers {
p   ICCG 1e−06 0.01;
U   BICCG 1e−05 0.1;
k   BICCG 1e−05 0.1;
epsilon BICCG 1e−05 0.1;
R   BICCG 1e−05 0.1;
nuTilda BICCG 1e−05 0.1;
} 20

SIMPLE {
  nNonOrthogonalCorrectors 0;
pRefCell 0;
pRefValue 0;
}
relaxationFactors {
  p  0.3;
U   0.7;
k   0.7;
epsilon 0.7;
R   0.7;
nuTilda 0.7;
} 30

// ************************************************************************* //

B.2.3.4 sampleDict file

FoamFile {
  version 2.0;
  format ascii;
  root "";
case "";
in stance "";
local "";
class dictionary;
object sampleDict;
}

// **********************************************

interpolationScheme cellPoint;
writeFormat raw;
sampleSets {
  uniform {
    name leftPatch;
    axis y;
  } 20
```
B.2. Run (Case) for Seed Implant

```
start (0 0.5 0.25);
end (0 2 0.25);
nPoints 100;
}

fields {
  sigmaxx
};

// ************************************************************************* //

B.2.3.5 twoRegionDict file

FoamFile {
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class dictionary;
  object twoRegiondict;
}

// ************************************************************************* //

space mu 1.0;
shift_x 0.0; // distance between center and point on X axis
shift_y 0.0; // distance between center and point on Y axis
x_1_original -19.999; // position of particle 1 on X axis
y_1_original 4.16; // position of particle 1 on Y axis
x_2 -19.999; // position of the particle 2 on X axis
y_2 -4.14; // position of the particle 2 on Y axis

// ************************************************************************* //

```
Appendix C
Mesh Generator for Stent Model

In this chapter, a mesh generator to create a 2D stent model is presented.

C.1 Mesh Generator

#!/usr/bin/python
DOC = "Build a mesh comparable to that used in Ritter's 2007a paper. Mesh consists of a rectangular domain containing n circular seeds, The North and South boundaries are symmetry boundaries and blood flow is from West (inflow) to East (outflow) boundary."
from vector import *
import blockMeshDict, os

# default parameter settings
origin = Vector(0, 0, 0) # location of origin
convertToMeters = 1.0
grading = 8
n = 10
R_s = 1
delta = 30
a = 0.979898987

import sys
if len(sys.argv)==1: # no parameters => in interactive/help mode
    print sys.argv[0], "\t - \t", DOC
print "Enter parameter data ... 
convertToMeters = (raw_input("\t\tConvertToMeters factor (%s) : " % (convertToMeters ))).strip() or convertToMeters
grading = (raw_input("\t\tMesh grading (%s) : " % (grading ))).strip() or grading
R_s = (raw_input("\t\tSeed radius (%s) : " % (R_s ))).strip() or R_s
n = (raw_input("\t\tNumber of seeds (%s) : " % (n ))).strip() or n
delta = (raw_input("\t\tDistance between (%s) : " % (delta ))).strip() or delta
a = max(1.0, delta*0.2)

if len(sys.argv)==7 or len(sys.argv)==10 :
    convertToMeters = float(sys.argv[1])
grading = int(sys.argv[2])
R_w = float(sys.argv[3])
n = int(sys.argv[4])
delta = float(sys.argv[5])
a = float(sys.argv[6])
else:
    origin = Vector(float(sys.argv[7]), float(sys.argv[8]), float(sys.argv[9]))

os.system("./build_cv %s %s %s %s %s %s > blockMeshDict" % (convertToMeters, grading, R_s, n, delta, a))
sys.exit(0)
print "\t Width of annulus around seed (a) : ", convertToMeters(a) 50
print "\t Number of seeds (n) : ", n 60
print "\t convertToMeters (convertToMeters) : ", convertToMeters 70
print "\t grading (grading) : ", grading 80
print "\t Radius of seed (R_s) : ", R_s 90
print "\t grading (grading) : ", grading 100
print "\t Width between seeds (delta) : ", delta
print "\t Width of annulus around seed (a) : ", a 110
print "\t Mesh origin (origin) : ", origin 120
print "\t Mesh origin (origin) : ", origin
print "\t Mesh origin (origin) : ", origin
print "\t Mesh origin (origin) : ", origin

# START
#t = 1.0/sqrt(2.0)
ty=1.0/sqrt(2.0) #ty=0.707
tx=1.0/sqrt(2.0) #tx=0.707
W = Vector(0, 0)
vertices = []

# west edge
x = -19 - (n-1)*(2+delta) - 1
y = 8.0 - (1+a)*ty
y_1 = -y
vertices.append( Vector(x, -8.0) )
vertices.append( Vector(x, -8.0+2*(1+a)*ty) )
vertices.append( Vector(x, 0) )
vertices.append( Vector(x, +8.0-2*(1+a)*ty) )
vertices.append( Vector(x, +8.0) )

# seed for seed in range(n,0,-1):
x = -(2+delta)*(seed-1) # centre of wire x
y = 8.0 - (1+a)*ty # centre of wire y
x_1 = -(2+delta)*(seed-1)+16.0 # centre of wire 2 x
y_1 = -y # centre of wire 2 y
vertices.append( Vector(x-(1+a)*tx, -8.0) )
vertices.append( Vector(x-(1+a)*tx, -8.0+2*(1+a)*ty) )
vertices.append( Vector(x-(1+a)*tx, 0) )
vertices.append( Vector(x-(1+a)*tx, +8.0-2*(1+a)*ty) )
vertices.append( Vector(x-(1+a)*tx, +8.0) )
vertices.append( Vector(x-1*tx, y-1*ty) )
vertices.append( Vector(x-1*tx, y+1*ty) )
vertices.append( Vector(x+1*tx, y-1*ty) )
vertices.append( Vector(x+1*tx, y+1*ty) )
vertices.append( Vector(x+(1+a)*tx, -8.0) )
vertices.append( Vector(x+(1+a)*tx, -8.0+2*(1+a)*ty) )
vertices.append( Vector(x+(1+a)*tx, 0) )
vertices.append( Vector(x+(1+a)*tx, +8.0-2*(1+a)*ty) )
vertices.append( Vector(x+(1+a)*tx, +8.0) )
vertices.append( Vector(x_1-(1+a)*tx, -8.0) )
vertices.append( Vector(x_1-(1+a)*tx, -8.0+2*(1+a)*ty) )
vertices.append( Vector(x_1-(1+a)*tx, 0) )
C.1. Mesh Generator

for blocks
    blocks.append( Vector(x, 0) )
    blocks.append( Vector(x, +8.0) )
    blocks.append( Vector(x, y, y) )
    for move points relative to origin
        vertices[vertices].append( Vector(x, y) )
        vertices[vertices].append( Vector(x, y) )
        vertices[vertices].append( Vector(x, y) )
        vertices[vertices].append( Vector(x, y) )

    # scale lengths by R_s
    for k in range(len(vertices)): vertices[k] = vertices[k].scale(R_s)

    # move points relative to origin
    for k in range(len(vertices)): vertices[k] = vertices[k] - origin

V = len(vertices)

# EDGES
edges = []
for seed in range(n, 0, -1):
    C = Vector(−(2+delta)∗(seed−1)y)  # centre of wire
    k = (n−seed)∗28
    A = k+ 8; B = k+17; edges.append( [A, B, (vertices[A]−C).rotate(45) + C] )
    A = k+17; B = k+18; edges.append( [A, B, (vertices[A]−C).rotate(45) + C] )
    A = k+ 9; B = k+ 8; edges.append( [A, B, (vertices[A]−C).rotate(45) + C] )
    A = k+10; B = k+12; edges.append( [A, B, (vertices[A]−C).rotate(45) + C] )
    A = k+12; B = k+13; edges.append( [A, B, (vertices[A]−C).rotate(45) + C] )
    A = k+11; B = k+10; edges.append( [A, B, (vertices[A]−C).rotate(45) + C] )
    C_1 = Vector(16.0−(2+delta)∗(seed−1)y, 1.0)  # centre of wire 2
    A_1 = k+28; B_1 = k+29; edges.append( [A_1, B_1, (vertices[A_1]−C_1).rotate(45) + C_1] )
    A_1 = k+29; B_1 = k+30; edges.append( [A_1, B_1, (vertices[A_1]−C_1).rotate(45) + C_1] )
    A_1 = k+20; B_1 = k+19; edges.append( [A_1, B_1, (vertices[A_1]−C_1).rotate(45) + C_1] )
    A_1 = k+19; B_1 = k+25; edges.append( [A_1, B_1, (vertices[A_1]−C_1).rotate(45) + C_1] )
    A_1 = k+17; B_1 = k+18; edges.append( [A_1, B_1, (vertices[A_1]−C_1).rotate(45) + C_1] )

    gA = 1
    gB = 2  # x block 0... and 2...
    gC = 2  # y block 0... and 2...

    blocks.append( [ 0, 5, 6, 1, gA, gB] )
    blocks.append( [ 1, 6, 7, 2, gA, gA] )
    blocks.append( [ 2, 7, 8, 3, gA, gA] )
blocks.append([3, 8, 9, 4, gA, gB])

for seed in range(0,n):
    k = 0+seed*28
    # seed column
    blocks.append([k+5, k+14, k+15, k+6, gB, gB])
    blocks.append([k+6, k+15, k+16, k+7, gB, gA])
    blocks.append([k+7, k+16, k+17, k+8, gB, gA])
    blocks.append([k+8, k+17, k+12, k+10, gB, gB])
    blocks.append([k+12, k+17, k+18, k+13, gB, gB])
    blocks.append([k+13, k+18, k+9, k+11, gB, gB])
    blocks.append([k+11, k+9, k+8, k+10, gB, gB])
    blocks.append([k+14, k+19, k+20, k+15, gA, gB])
    blocks.append([k+15, k+20, k+21, k+16, gA, gA])
    blocks.append([k+17, k+22, k+23, k+18, gA, gB])
    blocks.append([k+19, k+28, k+26, k+24, gB, gB])
    blocks.append([k+26, k+28, k+29, k+27, gB, gB])
    blocks.append([k+27, k+29, k+20, k+25, gB, gB])
    blocks.append([k+25, k+20, k+19, k+24, gB, gB])
    blocks.append([k+20, k+29, k+30, k+21, gA, gA])
    blocks.append([k+21, k+30, k+31, k+22, gB, gA])
    blocks.append([k+22, k+31, k+32, k+23, gB, gB])

    if seed<n-1:
        blocks.append([k+28, k+33, k+34, k+29, gA, gB])
        blocks.append([k+29, k+34, k+35, k+30, gA, gA])
        blocks.append([k+30, k+35, k+36, k+31, gA, gA])
        blocks.append([k+31, k+36, k+37, k+32, gA, gB])

    # right hand side column
    k = 0 + (n-1)*28
    blocks.append([k+28, k+36, k+37, k+29, gA, gB])
    blocks.append([k+29, k+37, k+33, k+30, gA, gA])
    blocks.append([k+30, k+33, k+34, k+31, gA, gA])
    blocks.append([k+31, k+34, k+35, k+32, gA, gB])

    # PATCHES
    patches = []
    patches.append(['patch', 'inlet', [0, 1, 2, 3, 4]])
    k = 5+n*28
    patches.append(['patch', 'outlet', [k+3,k+4,k,k+1,k+2]])

    tmp = [4]
    for seed in range(0,n):
        tmp.append(9+28*seed)
        tmp.append(18+28*seed)
        tmp.append(23+28*seed)
        tmp.append(32+28*seed)
        tmp.append(35+28*(n-1))
patches.append( ["wall", "top", tmp] )

tmp = [0]
for seed in range(0,n):
    tmp.append( 5+28*seed)
    tmp.append(14+28*seed)
    tmp.append(19+28*seed)
    tmp.append(28+28*seed)
    tmp.append(36+28*(n-1))
patches.append( ["wall", "bottom", tmp] )

for seed in range(0,n):
    k = 10 + seed*28
    patches.append( ["wall", ("seedtop_%d" % seed), [k, k+2, k+3, k+1, k]] )
    k_1=24 + seed*28
    patches.append( ["wall", ("seedbot_%d" % seed), [k_1, k_1+2, k_1+3, k_1+1, k_1]] )

# publish data to blockMeshDict.routines
blockMeshDict.convertToMeters = convertToMeters
blockMeshDict.grading = grading
blockMeshDict.vertices = vertices
blockMeshDict.edges = edges
blockMeshDict.blocks = blocks
blockMeshDict.patches = patches

blockMeshDict.printHeader()
blockMeshDict.printVertices()
blockMeshDict.printEdges()
blockMeshDict.printBlocks()
blockMeshDict.printPatches()

---

C.2 BlockMeshDict.py

# blockMeshdict.py
# python utility library to generate blockMeshDict

from vector import *
from math import *
back_offset = Vector(0, 0, 0)
front_offset = Vector(0, 0, 1)
frontPatch = []
backPatch = []

def printHeader():
    global convertToMeters
    print r"""/**********************************************************************************\"
    | ==-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*\"
    | ==-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*\"
```
FoamFile
{
    version 2.0;
    format ascii;
    root "
    case "
    instance "
    local "
    class dictionary;
    object blockMeshDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
convertToMeters %s;
// *** % (convertToMeters)
def printVertices():
global vertices
print "vertices\n" % (2*len(vertices))
for vertex in vertices: print "\t" % (vertex + back_offset)
for vertex in vertices: print "\t" % (vertex + front_offset)
print "\n"
def buildEdge(edge, d, zOffset):
    start, end, point = edge
    return "\tarc %2d %2d %s" % (start+d, end+d, point+zOffset)
def printEdges():
global edges, vertices
V = len(vertices)
print "edges\n"
for edge in edges: print buildEdge (edge, 0, back_offset)
for edge in edges: print buildEdge (edge, V, front_offset)
print "\n"
def buildBlock(face):
global vertices
V = len(vertices)
# grading values
gx, gy = 1, 1
front_patch.append("4(\%s  \%s  \%s  \%s)\%s\%s\%s\%s" % (face[3], face[2], face[1], face[0]))
if grading>=2 and len(face)>6: gx, gy = face[4:6]
if grading>=2 and len(face)>6: sx, sy = face[6:]
return "\thex (\%d  \%d  \%d  \%d  \%d  \%d  \%d  \%d) (\%d  \%d) simpleGrading (\%d \%d 1)\%s\%s\%s\%s\%s\%s\%s\%s" % 
    (face[0], face[1], face[2], face[3],
def printBlocks():
global blocks, vertices
print "blocks\n"
for block in blocks: print buildBlock(block)
print "\n"
def buildPatch (patch):
    global vertices
```

C.3 Vector.py

from math import *

class Vector:
    def __init__(self, x, y, z=0):
        self.x = x
        self.y = y
        self.z = z
    def __add__(self, other):
        return Vector(self.x + other.x, self.y + other.y, self.z + other.z)
    def __sub__(self, other):
        return Vector(self.x - other.x, self.y - other.y, self.z - other.z)
    def mask(self, other):
        return Vector(self.x * other.x, self.y * other.y, self.z * other.z)
    def dot(self, other):
        return (self.x * other.x + self.y * other.y + self.z * other.z)
    def norm(self):
        return sqrt(self.x * self.x + self.y * self.y + self.z * self.z)
    def scale(self, factor):
        return Vector(factor * self.x, factor * self.y, factor * self.z)
    def unit(self):
        f = 1 / self.norm()
        return Vector(f * self.x, f * self.y, f * self.z)
    def translate(self, other):
        return self + other
    def rotate(self, angle, inRad=0):
        if not inRad: angle = angle * pi / 180.0
        return Vector(
            self.x * cos(angle) - self.y * sin(angle),
            self.x * sin(angle) + self.y * cos(angle),
            self.z)
    def __str__(self):
        return "%.6f %.6f %.6f" % (self.x, self.y, self.z)

if __name__ == '__main__':
print "Testing norm"
a = Vector(3,4,0)
b = Vector(0,0,0)
print "Vector a=%s has norm %d " % (a, a.norm())
print "Vector b=%s has norm %d " % (b, b.norm())

print "Testing operations"
a = Vector(3,4,1)
b = Vector(-5,2,50)
print "%s+%s = %s " % (a, b, a+b)
print "%s-%s = %s " % (a, b, a-b)
print "%s.%s = %s " % (a, b, a.dot(b))
print "%s.%s = %s (test %s) " % (a,a, a.dot(a), a.norm()**2)

print "Testing transformations"
a = Vector(1,0,1)
print a
a = a.rotate(90)
print a
a = a.rotate(90)
print a
a = a.rotate(90)
print a
a = a.rotate(90)
print a
print a
Appendix D

OpenFOAM Code for Stent Model

In this chapter, we present the OpenFOAM C++ code which was generated for the calculation of the dipole-dipole and hydrodynamic interactions for multiple MDCPs with stent implant. In this stent model, the agglomeration of the MDCPs is considered.

D.1 Applications (Solver) for Stent Implant

D.1.1 createFields.H file

```c++
#include "createPhi.H"

label pRefCell = 0;
scalar pRefValue = 0.0;
setRefCell(p, mesh.solutionDict().subDict("SIMPLE"), pRefCell, pRefValue);

singlePhaseTransportModel laminarTransport(U, phi);
autoPtr<turbulenceModel> turbulence(
  turbulenceModel::New(U, phi, laminarTransport)
);
```
D.1.2 createFields-analytic.H file

/** *
 * File Name: createFields_analytic.H
 * Author: mardinoglu@yahoo.com
 * Description:
 */
Info<< "region: Reading field T\n" << endl;
volScalarField T(
    IOobject(
        "$T$",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
volVectorField H(
    IOobject(
        "$H$",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
volVectorField space_H_0(
    IOobject(
        "$space_H_0$",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
volVectorField M_aviles(
    IOobject(
        "$M_aviles$",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
volVectorField moment_Cregg(
    IOobject(
        "$moment_Cregg$",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    )
);
D.1. Applications (Solver) for Stent Implant

volVectorField modif_B(
    IOobject(
        "modif_B",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);

Info<< "region:Reading physicalProperties\n" << endl;
IODictionary physicalProperties(
    IOobject(
        "physicalProperties",
        runTime.constant(),
        mesh,  
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    )
);

Info<< "Reading parameter - magnetic permeability of free space" << endl;
dimensionedScalar mu_0 (physicalProperties.lookup("mu_0"));
Info<< "Magnetic permeability of free space (mu_0) is " << mu_0 << endl;

Info<< "Reading parameter - H_m field angle (theta)" << endl;
dimensionedScalar theta (physicalProperties.lookup("theta"));
Info<< "H_m field angle (theta) is " << theta << endl;

Info<< "Reading parameter - B applied field magnitude (B_0)" << endl;
dimensionedScalar B_0 (physicalProperties.lookup("B_0"));
Info<< "B applied field magnitude (B_0) is " << B_0 << endl;

Info<< "Calculating parameter - H_m field magnitude (H_0)" << endl;
dimensionedScalar H_0 = B_0/mu_0;
Info<< "H_m field magnitude (H_0) is = " << H_0 << endl;

Info<< "Magnetic susceptibility of implant at H_0=0" << endl;
dimensionedScalar chi_i_0 (physicalProperties.lookup("chi_i_0"));
Info<< "Magnetic susceptibility of implant at H_0=0 (chi_i_0) is " << chi_i_0 << endl;

Info<< "Saturation magnetization of implant (M_i_s) is = " << M_i_s << endl;

Info<< "Demagnetizing factor (implant_alpha_i) is = " << alpha_i << endl;

Info<< "Susceptibility of implant(implant_chi_i) is = " << chi_i << endl;
Info<< "Calculating parameter- - implant_mu" << endl;
scalar implant_mu = 1.0+chi_i.value();
Info "Implant mu is " <<implant_mu<<endl;

Info "Calculating parameter - Magnetization of the implant (M_i)"<<endl;
dimensionedScalar M_i = 2*alpha_i*H_0;
Info "Magnetization of the implant (M_i) is "<<M_i<<endl;

Info "Reading parameter - saturation magnetization of the material in MDCP"<<endl;
dimensionedScalar M_fm_p_s (physicalProperties.lookup("M_fm_p_s"));
Info "Saturation magnetization of the material in MDCP is " <<M_fm_p_s<<endl;

Info "Reading parameter - volumetric magnetic susceptibility of Medium"<<endl;
dimensionedScalar chi_m (physicalProperties.lookup("chi_m"));
Info "Magnetic susceptibility of Medium is " <<chi_m << endl << endl;

Info "Reading parameter-density of the ferromagnetic material in the MDCP"<<endl;
dimensionedScalar rho_fm_p (physicalProperties.lookup("rho_fm_p"));
Info "Density of the ferromagnetic material in the MDCP is " <<rho_fm_p<<endl;

Info "Reading parameter-density of the polymer and/or drug in the MDCP"<<endl;
dimensionedScalar rho_pol_p (physicalProperties.lookup("rho_pol_p"));
Info "Density of the polymer and/or drug in the MDCP is " <<rho_pol_p<<endl;

Info "Reading parameter-mass fraction of the fer. material in the MDCP"<<endl;
dimensionedScalar x_fm_p (physicalProperties.lookup("x_fm_p"));
Info "Mass fraction of the fer. material in the MDCP is " <<x_fm_p<<endl;

Info "Reading parameter-magnetic moment of the single domain particles"<<endl;
dimensionedScalar moment_fm_p_s (physicalProperties.lookup("moment_fm_p_s"));
Info "Magnetic moment of the single domain particle is " <<moment_fm_p_s<<endl;

Info "Reading parameter - Boltzmann constant (k_B)"<<endl;
dimensionedScalar k_B (physicalProperties.lookup("k_B"));
Info "Boltzmann constant is " <<k_B<<endl;

Info "Reading parameter - Temperature (T)"<<endl;
dimensionedScalar T_L (physicalProperties.lookup("T_L"));
Info "Temperature (T_L) is " <<T_L<<endl;

Info "Reading parameter-volume fraction of the fer. material in the MDCP"<<endl;
dimensionedScalar w_fm_p (physicalProperties.lookup("w_fm_p"));
Info "Volume fraction of the fer. material in the MDCP is "<<w_fm_p<<endl;

Info "Calculating parameter -volume fraction of the fer. in the MDCP"<<endl;
dimensionedScalar omega_fm_p = (w_fm_p);
Info "Volume fraction of the fer. in the MDCP is " <<omega_fm_p<<endl;

Info "Reading parameter - radius of implant"<<endl;
dimensionedScalar R_i (physicalProperties.lookup("R_i"));
Info "Radius of implant is " <<R_i<<endl;

Info "Reading parameter - radius of implant (R_i_scale)"<<endl;
dimensionedScalar R_i_scale (physicalProperties.lookup("R_i_scale"));
Info "Radius of implant is " <<R_i_scale<<endl;

Info "Reading parameter - radius of MDCP (R_p)"<<endl;
dimensionedScalar \( R_p \) (physicalProperties.lookup("R_p"));
Info<< "Radius of MDCP is " <<R_p<< endl;

Info<< "Calculating parameter - volume of MDCP (volume_p)"<<endl;
dimensionedScalar volume_p=4.0/3.0*mathematicalConstant::pi*R_p*R_p*R_p;
Info<< "Radius of MDCP is " <<volume_p<< endl;

Info<< "Reading parameter - parent blood vessel radius (R_pv)"<<endl;
dimensionedScalar R_pv (physicalProperties.lookup("R_pv"));
Info<< "Parent blood vessel radius is " <<R_pv<< endl;

Info<< "Reading parameter - blood viscosity" <<endl;
dimensionedScalar eta_beta (physicalProperties.lookup("eta_beta"));
Info<< "Blood viscosity (eta_beta) is " <<eta_beta<< endl;

Info<< "Reading parameter - blood density" <<endl;
dimensionedScalar rho_beta (physicalProperties.lookup("rho_beta"));
Info<< "Blood density is " <<rho_beta<< endl;

Info<< "Reading parameter - Porosity of a cluster of MDCP" <<endl;
dimensionedScalar epsilon_p (physicalProperties.lookup("epsilon_p"));
Info<< "Porosity of a cluster of MDCP is " <<epsilon_p<< endl;

Info<< "Reading parameter - average inlet velocity (u_0)"<<endl;
dimensionedScalar u_0 (physicalProperties.lookup("u_0"));
Info<< "Parameter - average inlet velocity (u_0) is " <<u_0<< endl;

Info<< "Calculating parameter -Magnetic velocity (velocity_m)"<<endl;
dimensionedScalar velocity_m = (2.0/9.0)*((R_p*R_p/R_pv)*(mu_0/eta_beta))
* (1-epsilon_p)*omega_fm_p*M_i*M_fm_p_s;
Info<< "Magnetic velocity is =" <<velocity_m<< endl;

vector H_0vector = vector(H_0.value()*cos(convertToRad*theta).value(),
H_0.value()*sin(convertToRad*theta).value(), 0.0);
Info<< " End of createFields_analytic" <<endl;

\section{readtwoRegionDict.H file}

// file: readtwoRegionDict.H
//
// Read info from twoRegionDict
//******************************************************************************************
//******************************************************************************************

Info <<"\n\n@ Reading twoRegionDict\n" << end;

IOdictionary twoRegionDict
(IObject
 ( "twoRegionDict",
 runTime.system(),
 runTime,
 IObject::MUST_READ,
 IObject::NO_WRITE)
D.1. Applications (Solver) for Stent Implant

Info << "Reading parameter -- space_mu" << endl;
scalar space_mu = readScalar(twoRegionDict.lookup("space_mu"));
Info << "Parameter -- space_mu = " << space_mu << endl;

Info << "Reading parameter -- distance between center and point on X axis " << endl;
scalar shift_x = readScalar(twoRegionDict.lookup("shift_x"));
Info << "Parameter -- distance between center and point on X axis = " << shift_x << endl;

Info << "Reading parameter -- distance between center and point on Y axis" << endl;
scalar shift_y = readScalar(twoRegionDict.lookup("shift_y"));
Info << "Parameter -- distance between center and point on Y axis = " << shift_y << endl;

Info << "Reading parameter -- position of particle one on X axis" << endl;
scalar x_1_original = readScalar(twoRegionDict.lookup("x_1_original"));
Info << "Parameter -- position of particle one on X axis= " << x_1_original << endl;

Info << "Reading parameter -- position of particle one on Y axis" << endl;
scalar y_1_original = readScalar(twoRegionDict.lookup("y_1_original"));
Info << "Parameter -- position of particle one on Y axis= " << y_1_original << endl;

Info << "Reading parameter -- position of particle two(reference) on X axis" << endl;
scalar x_2_original = readScalar(twoRegionDict.lookup("x_2_original"));
Info << "Parameter -- position of particle two on X axis= " << x_2_original << endl;

Info << "Reading parameter -- position of particle two(reference) on Y axis" << endl;
scalar y_2_original = readScalar(twoRegionDict.lookup("y_2_original"));
Info << "Parameter -- position of particle one on Y axis= " << y_2_original << endl;

D.1.4 stent.C file

/**
 * File Name: hydstent.C
 * Author: mardinogl@yahoo.com
 * Here magnetic and hydrodynamic interaction effect is calculated by considering the agglomeration of multiple particles
 */
#include "fvCFD.H"
#include "incompressible/singlePhaseTransportModel/singlePhaseTransportModel.H"
#include "incompressible/turbulenceModel/turbulenceModel.H"
#include "typeInfo.H"
#include "OFstream.H"
#include "IOmanip.H"
#include "mathematicalConstants.H"

// ************************************************************

int main(int argc, char *argv[])
{
    const scalar convertToRad = mathematicalConstant::pi/180.0;

    # include "setRootCase.H"
    # include "createTime.H"
    # include "createMesh.H"
    # include "createFields.H"
}
# include "createFields_analytic.H"
# include "initContinuityErrs.H"
# include "readtwoRegionDict.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Info << "\nEVALUATING ANALYTICAL SOLUTION..." << endl;
/**
* Scalar potential and the resulting magnetic field is calculated analytically.
* We consider 20 wires in the space region.
*
**/
#define Nseed 10  
  // For resetting the value of any vectorfield
  volVectorField Zero
    
      IOobject
        
          ("Zero",
            runTime.timeName(),
            mesh,
            IOobject::NO_READ,
            IOobject::NO_WRITE
          ),
        mesh,
        ,dimensionedVector("0", dimless, vector(0,0,0))
    
    // H_0 is calculated for the space region
    space_H_0=dimensionedVector("0", dimless, H_0_vector);

    volVectorField H_space_total=space_H_0;

    scalar const_a2 = (-H_0.value());
    scalar const_b2 = -((implant_mu-space_mu)/(space_mu+implant_mu))*const_a2 ;

    volVectorField centres_org = T.mesh().C();
    volScalarField mag_centres_org = mag(centres_org);

    for(int i=0; i<2*Nseed; i++)
      
        //Info << “ Calculating the i-th seed = ” << i << endl;
        int i_x=1-i;
        int i_y=pow(-1,i);

        //Info << “ Parameter i_x= ” << i_x << endl;
        //Info << “ Parameter i_y= ” << i_y << endl;

        volVectorField centres
          IOobject ("Centres_seeds",
            runTime.timeName(),
            mesh,
            IOobject::NO_READ,
            IOobject::AUTO_WRITE),
            (centres_org.component(vector::X)-(i_x*shift_x)*vector(1,0,0)+(centres_org.component(vector::Y)-(i_y*shift_y))*vector(0,1,0)+(centres_org.component(vector::Z))*vector(0,0,1)
          );

        volScalarField mag_centres = mag(centres);
volScalarField radius = sqrt((centres.component(vector::X)*centres.component(vector::X))
+(centres.component(vector::Y)*centres.component(vector::Y)));
volScalarField theta = acos((centres & vector(1,0,0))/mag_centres);

T=(const_a2*centres.component(vector::Y)+const_b2*centres.component(vector::Y)/(radius*radius));
volVectorField H_space_ninty(IOobject("H_ninty", runTime.timeName(), mesh, IOobject::NO_READ, IOobject::AUTO_WRITE),
(((2*const_b2*centres.component(vector::X)*centres.component(vector::Y))
/(radius*radius*radius*radius))*vector(1,0,0))+((-const_a2−(const_b2*
(centres.component(vector::X)*centres.component(vector::X)−centres.component(vector::Y)*
centres.component(vector::Y))/(radius*radius*radius*radius)))*vector(0,1,0));

H_space_total=H_space_total+H_space_ninty;

// Total magnetic field in the space is calculated for the stent
H=H_space_total;

/**
beta_Aviles, Langevin_Aviles, M_aviles, fw_langevin, velocity_particle_an_Cregg,
magmoment_Cregg, F_Cregg and F_Cregg_velocity are created to check
the accuracy of the code and formulas. Magnetic velocity is calculated by two different formula.
**/
volScalarField beta_Aviles(IOobject("beta_Aviles", runTime.timeName(), mesh, IOobject::NO_READ, IOobject::AUTO_WRITE),
(moment_fm_p*s*mu_0*mag(H))/(k_B*T_L));

//Langevin Factor is calculated for the langevin function
volScalarField Langevin_Aviles(IOobject("Langevin_Aviles", runTime.timeName(), mesh, IOobject::NO_READ, IOobject::AUTO_WRITE),
(1.0/Foam::tanh(beta_Aviles))−(1.0/beta_Aviles));

//Total magnetisation due to the stent is calculated
M_aviles=(M_fin_p*s*Langevin_Aviles*H)/mag(H);

volVectorField fw_langevin(IOobject("fw_langevin", runTime.timeName(), mesh, IOobject::NO_READ, IOobject::AUTO_WRITE),
(moment_fm_p*s*mu_0*mag(H))/(k_B*T_L));
\( \text{IOobject::NO\ READ,} \)
\( \text{IOobject::AUTO\ WRITE,} \)
\( (1.0/(2.0*R_i\textunderscore\scale))\*\mu_0\*fvc\\textunderscore\grad((M\_aviles&H)) \);

//Velocity of particle due to the stent is calculated
volVectorField velocity\_particle\_an\_Cregg( 
  IOobject( "velocity\_p\_an\_Cregg", 
    runTime.timeName(), 
    mesh, 
    IOobject::NO\ READ, 
    IOobject::AUTO\ WRITE), 
  (2.0/9.0)*R_p\*R_p\*\mu_0\*\omega_{fm\_p}\*(1.0/((\eta_{beta}\*u_0*R_i\textunderscore\scale)))*fvc\\textunderscore\grad((M\_aviles&H)) 
);

// Magnetic moment of the particle due to the stent is calculated
volVectorField magmoment\_Cregg( 
  IOobject( "magmoment\_Cregg", 
    runTime.timeName(), 
    mesh, 
    IOobject::NO\ READ, 
    IOobject::AUTO\ WRITE), 
  (\omega_{fm\_p}\*\text{-}volume\_p\*M\_fm\_p\text{-}s\_Langevin\_Aviles\*(\mu_0\text{-}H)/(\mu_0\text{-}mag(H))) 
);

//Magnetic force due to stent is calculated
volVectorField F\_Cregg( 
  IOobject( "F\_Cregg", 
    runTime.timeName(), 
    mesh, 
    IOobject::NO\ READ, 
    IOobject::AUTO\ WRITE), 
  (1.0/(R_i\textunderscore\scale))\*(fvc\\textunderscore\grad((magmoment\_Cregg)\&(\mu_0\text{-}H})) 
);

/**
  * Velocity due to the magnetic force is calculated
  * F\_Cregg\_velocity is compared with velocity\_particle\_an\_Cregg and they generate same results.
  *
  **/
volVectorField F\_Cregg\_velocity( 
  IOobject( "F\_Cregg\_velocity", 
    runTime.timeName(), 
    mesh, 
    IOobject::NO\ READ, 
    IOobject::AUTO\ WRITE),
  F\_Cregg\*(1.0/(6.0*\mathematicalConstant::pi*\eta_{beta}\*R_p*u_0)) 
);

Info <<"\n# Maximum U in Space 
 " <<setw(4)
  <<" " <<setw(9) <<setprecision(4) <<min(mag(U))
  <<" \\
  " <<setw(9) <<setprecision(4) <<max(mag(U))
  << "\n";

Info <<"\n# beta,Langevin\_velocity of particle for Cregg\n " <<setw(4)
  <<" " <<setw(9) <<setprecision(4) <<min(mag(velocity\_particle\_an\_Cregg))
  <<" \\
  " <<setw(9) <<setprecision(4) <<max(mag(velocity\_particle\_an\_Cregg))
  << "\n";

129
Info << "\n# mu_0*H general\n" << setw(4)
<< "" << setw(9) << setprecision(4) << min(mag(mu_0*H))
<< "" << setw(9) << setprecision(4) << max(mag(mu_0*H))
<< "";
Info << "\n# fw_langevin \n" << setw(4)
<< "" << setw(9) << setprecision(4) << min(mag(fw_langevin))
<< "" << setw(9) << setprecision(4) << max(mag(fw_langevin))
<< "";
Info << "\n# magmoment_Cregg\n" << setw(4)
<< "" << setw(9) << setprecision(4) << min(mag(magmoment_Cregg))
<< "" << setw(9) << setprecision(4) << max(mag(magmoment_Cregg))
<< "";
Info << "\n# F_Cregg\n" << setw(4)
<< "" << setw(9) << setprecision(4) << min(mag(F_Cregg))
<< "" << setw(9) << setprecision(4) << max(mag(F_Cregg))
<< "";
Info << "\n# F_Cregg_velocity\n" << setw(4)
<< "" << setw(9) << setprecision(4) << min(mag(F_Cregg_velocity))
<< "" << setw(9) << setprecision(4) << max(mag(F_Cregg_velocity))
<< "";
Info<< "\n# ANALYTICAL SOLUTION. . . .END" << endl;

ofstream myfile;
ofstream langfile;
myfile.open("stentinter.txt");
langfile.open("stentintlang.txt");

/**
* Blood Velocity is calculated by solving the Navier-Stokes equation
*/

Info<< "\nStarting time loop\n" << endl;
for (runTime++; !runTime.end(); runTime++)
{
    Info<< "Time = " << runTime.timeName() << nl << endl;
    # include "readSIMPLEControls.H"
    p.storePrevIter();
    // Pressure-velocity SIMPLE corrector
    
    // Momentum predictor
    tmp<fvVectorMatrix> UEqn
    
    (   
        fvm::div(phii, U) 
        + turbulence->divR(U) 
    );
    UEqn().relax();
    solve(UEqn() == -fvc::grad(p));
    p.boundaryField().updateCoeffs();
    volScalarField AU = UEqn().A();
    U = UEqn().H()/AU;
    UEqn.clear();
    phi = fvc::interpolate(U) & mesh.Sf();
adjustPhi(\phi, U, p);

// Non-orthogonal pressure corrector loop
for (int nonOrth=0; nonOrth<nNonOrthCorr; nonOrth++)
{
    fvScalarMatrix pEqn
    {
        fvm::laplacian(1.0/AU, p) == fvc::div(phi);
    }
    pEqn.setReference(pRefCell, pRefValue);
    pEqn.solve();
    if (nonOrth == nNonOrthCorr)
    {
        phi -= pEqn.flux();
    }
}

#include "continuityErrs.H"

//Explicitly relax pressure for momentum corrector
p.relax();
//Momentum corrector
U -= fvc::grad(p)/AU;
U.correctBoundaryConditions();
}
turbulence->correct();
runTime.write();
Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
   " ClockTime = " << runTime.elapsedClockTime() << " s"
   " nl << endl;

/**
 * Here we will calculate the dipole-dipole and hydrodynamic interactions effect.
 * We define the number of particles and define the array for storing the datas
 **/  
#define Npar 20
//Store the initial and current positions of particles.
vector r_ini[Npar];
//Store the new positions of particles.
vector r_n_new[Npar];
//Store the temp positions of particles (used for swapping the positions of particles).
vector r_n_temp[Npar];
//Store the distance between the particles.
vector r_up[Npar][Npar];
//Store the velocity of particles due to the stant and dipole dipole interaction effect.
vector velocity_dip_int[Npar];
//Store the velocity of particles due to the hydrodynamic interaction.
vector velocity_hyd_int[Npar];
//Store the velocity of particles due to the blood.
vector U_velocity_blood[Npar];
//Store the forces due to hydrodynamic interaction.
vector F_hyd[Npar];
//Store the radius of each particles.
scalar Radius_Particle[Npar];
//Move the particles out of the space region.
vector particle_out=vector(19.8, 0, 0);
/**
* We define the initial positions of particles.

```cpp
/* Initial velocity
   * Radius of particle
   * Agglomeration of particles is considered.
   * Radius of particle_1 will be recalculated.
   * Radius of particle_2 will be very very small`zero
   */

double pow(double base, double exponent);

for(int i=0; i<Npar; i++){
    for(int j=0; j<Npar; j++){
        Info <<"Distance between particles between [" << i << "] and [" << j << "] =" << mag(r_np[i][j])<<"\tCOMPARE WITH \t"<< ((Radius_Particle[i]+Radius_Particle[j])/R_i_scale.value())<endl;
        if (mag(r_np[i][j])>=0 && mag(r_np[i][j])<(Radius_Particle[i]+Radius_Particle[j])/R_i_scale.value()) {
            r_n[i]=r_n[i]+r_n[j]/2.0;
            r_n[j]=particle_out;
            if ((r_n[i]>.190){
                r_n[i]=particle_out;
            }
            Info <<r_n["<< i << "]=" <<r_n[i]<<endl;
            Info <<r_n["<< j << "]=" <<r_n[j]<<endl;
            if (Radius_Particle[i]>(R_p.value())){
                Info <<"Radius_Particle["<< i << "]=" <<Radius_Particle[i]<<endl;
            }
            scalar times_i=pow((Radius_Particle[i]/R_p.value()),3);
            scalar times_j=pow((Radius_Particle[j]/R_p.value()),3);
            scalar times_new=times_i+times_j;
            Info <<"times_i="<<times_i<<endl;
            Info <<"times_j="<<times_j<<endl;
            Info <<"times_new="<<times_new<<endl;
            Radius_Particle[i]=pow(times_new,(1.0/3.0))*R_p.value();
            Radius_Particle[j]=Radius_Particle[i]*0.00000001;
            Info <<"Radius_Particle["<< i << "]=" <<Radius_Particle[i]<<endl;
        }
```
Info <<"Radius_Particle["<< j "<<"]="<<Radius_Particle[j]<<<endl;
  }
}

// Output the new radius of the particles
for(int i=0; i<Npar; i++){
  Info <<"Current position r_n["<< i "<< "]=" <<r_n[i]<<<endl;
  Info <<"Radius_Particle["<< i "<< "]="<<Radius_Particle[i]<<<endl;
}

// Info<< “\nCalculating the magnetic dipole-dipole interaction\n” << endl;
volVectorField r_positions= T.mesh().C();

 scalar hyd_constant=(6.0*mathematicalConstant::pi*eta_beta.value());
 scalar N_count=1;

 for(int i=0; i<Npar; i++){
  // save the current positions of each particles
  langfile<<runTime.timeName()<<"r_n[0].x()"<<endl;
  r_n[0].x()<"r_n[0].y()"<<endl;
  r_n[2].x()<"r_n[2].y()"<<endl;
  r_n[4].x()<"r_n[4].y()"<<endl;
  r_n[6].x()<"r_n[6].y()"<<endl;
  r_n[8].x()<"r_n[8].y()"<<endl;
  r_n[9].x()<"r_n[9].y()"<<endl;

  // Calculates the values of each particles
  vector r_p=vector(r_n[i].x(),r_n[i].y(),0);
  if (r_p.x()<19.0){
    // Finds the blood velocity of each particles
    vector probePoint(r_p);
    label probeCell = mesh.findCell(probePoint);
    volTensorField gradU = fvc::grad(U);
    vector cellCentre = mesh.C()[probeCell];
    U.velocity_blood[i]= U[probeCell] + ((probePoint− cellCentre) & gradU[probeCell]);
  }

  // Finds the applied magnetic field on each particles due to the stent
  vector probePoint(r_p);
  label probeCell = mesh.findCell(probePoint);
  volTensorField gradH = fvc::grad(H);
  vector cellCentre = mesh.C()[probeCell];
  vector H_part_space = H[probeCell] + ((probePoint− cellCentre) & gradH[probeCell]);

  // Creates a constant vectorfield by using the position of particle
  volVectorField r_p_field;
  IOobject("r_p_field",
           runTime.timeName(),
           mesh,
           IOobject::NO_READ,
           IOobject::NO_WRITE),
           mesh,
           dimensionedVector("0", dimless, r_p));
D.1. Applications (Solver) for Stent Implant

// Creates a constant vectorfield by using the applied field on particle
volVectorField B_r_p = IOobject("B_r_p", runTime.timeName(), mesh, IOobject::NO_READ, IOobject::NO_WRITE), mesh,
dimensionedVector("0", dimless, (mu_0.value()*H_part_space));

// Calculates the dipole interaction effect of each particle
volVectorField F_int_p_total = Zero;
for(int j=0; j<Npar; j++){
    vector r_n = vector(r_n[j].x(),r_n[j].y(),0);
    if((mag(r_n−r_p))>((Radius_Particle[i]+Radius_Particle[j])/R_i.scale.value()) && r_n.x()<19)
        //Note that(Radius_Particle[i]+Radius_Particle[j])is the distance between the particles
        vector probePoint_n = mesh.findCell(probePoint_n);
        label probeCell_n = mesh.findCell(probePoint_n);
        //volTensorField gradH_n = fvc::grad(H);
        vector cellCentre_n = mesh.C[probeCell_n];
        vector H_part_space_n = H[probeCell_n] + ((probePoint_n− cellCentre_n) & gradH[probeCell_n]);

    // Creates a constant vectorfield by using the position of particle
    volVectorField r_n_field = IOobject("r_n_field", runTime.timeName(), mesh, IOobject::NO_READ, IOobject::NO_WRITE), mesh,
    dimensionedVector("0", dimless, r_n_vector);
}
(r_positions−r_n_vector_field)/pow(mag(r_positions−r_n_vector_field),2))−
(B_r_n_vector_field))*pow((1.0/R_i),3)
);

// Calculate the total forces acting on the particle due to the other particles
F_int_p_total=F_int_p_total+F_int_p_n;
}

else{
F_int_p_total=Zero;
}

// Calculates the modification to the B due to the other particles
modif_B=F_int_p_total;

volScalarField beta_Cregg_p(IOobject(  
"beta_Cregg_p",  
runTime.timeName(),  
mesh,  
IOobject::NO_READ,  
IOobject::AUTO_WRITE),  
(moment_fm_p_s*(mag((mu_0*H)+modif_B)))/(k_B*T_L)
);

// Calculates the Langevin factor for each particle by using the modified B
volScalarField Langevin_Aviles_p(IOobject(  
"Langevin_Aviles_p",  
runTime.timeName(),  
mesh,  
IOobject::NO_READ,  
IOobject::NO_WRITE),  
(1.0/Foam::tanh(beta_Cregg_p) − 1.0/beta_Cregg_p)
);

// Calculates the magnetic moment for each particle
volVectorField magmoment_Cregg_p(IOobject (  
"magmoment_Cregg_p",  
runTime.timeName(),  
mesh,  
IOobject::NO_READ,  
IOobject::AUTO_WRITE),  
(omega_fm_p*(4.0/3.0*mathematicalConstant::pi*Radius_Particle[i]*
Radius_Particle[j]*Radius_Particle[k]) *M_fm_p_s*Langevin_Aviles_p*  
((mu_0*H)+modif_B))/mag((mu_0*H)+modif_B)
);

// Fix the boundaries of the magnetic moment for each particle
moment_Cregg=magmoment_Cregg_p;

// Calculates the modified magnetic forces for each particle due to the stent and other particles
volVectorField F_int_overall_p(IOobject (  
"F_int_overall_p",  
runTime.timeName(),  
mesh,
D.1. Applications (Solver) for Stent Implant

IOobject::NO_READ,
IOobject::NO_WRITE,
(1.0/(R_i_scale))*(fvc:grad((moment_Cregg)&(mu_0*H+modif_B)))
);

//vector probePoint_p(x_p,y_p,p);

IOobject::NO_READ,
IOobject::NO_WRITE)
(1.0/(R scale))*(fvc:grad((moment_Cregg)&(mu_0*H+modif_B)));

530

//vector probePoint_p(x_p,y_p,0);

//label probeCell = mesh.findCell(probePoint_p);

volTensorField gradF_int_overall_p= fvc:grad(F_int_overall_p);

//vector cellCentre = mesh.C()[probeCell];

vector F_int_overall_p_vector = F_int_overall_p[probeCell] + ((probePoint-cellCentre)&
gradF_int_overall_p[probeCell]);

// Calculate the magnetic velocity of particles due to the stent and other particles
velocity_dip_int[i]=(F_int_overall_p_vector*(1.0/(hyd_constant*Radius_Particle[i]*u_0.value())));

540

else{
// if the particle is outside the space region we set the velocity to zero.
velocity_dip_int[i]=vector(0,0,0);
}

// Outputs the velocity of particles (Just for check)
for(int i=0; i<Npar; i++){
Info <<"Velocity_dipole[i]=" <<velocity_dip_int[i]<<endl;
}

// Calculate the hydrodynamic interactions
for(int k=0; k<10; k++){
vector F_hyd_temp=vector(0,0,0);

for(int i=0; i<Npar; i++){
if (r_n[i].x()<19.0){
for(int j=0; j<Npar; j++){
if((mag(r_np[i][j]))>((Radius_Particle[i]+Radius_Particle[j])/R_i_scale.value()) &
r_n[j].x()<19){

560
vector normalized=(r_np[i][j])/mag(r_np[i][j]);
tensor hyd_force=(hyd_constant*Radius_Particle[i]*3.0*Radius_Particle[j]*(1.0/((4.0*mag(r_np[i][j])*R_i.value())))*
tensor(1.0,0,0,0,1,0,0,0,1)+(normalized*normalized));
F_hyd_temp=F_hyd_temp+(hyd_force & velocity_hyd_int[i]);
}
}
F_hyd[i]=F_hyd_temp;
F_hyd_temp=vector(0,0,0);
}
else{
F_hyd[i]=vector(0,0,0);
}
}

// Calculate the new positions of the particles
for(int i=0; i<Npar; i++){
velocity_hyd_int[i]=(hyd_constant*Radius_Particle[i]*velocity_dip_int[i]+F_hyd[i])/
(hyd_constant*Radius_Particle[i]);
}

570

// Calculate the new positions of the particles
for(int i=0; i<Npar; i++){

580

136
D.2 Run (Case) for Stent Implant

D.2.1 0 (initial conditions file)

D.2.1.1 epsilon file

```cpp
/*—————————————————————————*
|        |        |
---------------------
  137

Info << "Velocity_hyd+dip[v << i << v]=" << velocity_hyd_int[i] << endl;
r_n_temp[i]=r_n[i];

if(r_n[i].x()<19){
r_n_new[i]=r_n[i]+0.1*(U_velocity_blood[i]+velocity_hyd_int[i]);
}
else{
   // Particles dont move if thy are outside the space region
   r_n_new[i]=r_n[i];
}

// Update the positions of the particles
for(int j=0; j<Npar; j++){
   for(int i=0; i<2*Nseed; i++){
      int i_x=i+1;
      int i_y=-pow(1, i);
      vector seed_centre=vector((32.0−(16.0*i_x), (6.7*i_y), 0);
      Info<< "seed_centre_" <<i<<"seed_centre_" << endl;
      if(mag(r_n_new[i]−seed_centre)<1.0 && r_n_new[i].x() < 19){
         r_n[i]=r_n_temp[i];
         N_count_1=N_count_1+1;
         break;
      }
      else{
         r_n[i]=r_n_new[i];
      }
   }
   Info<< "N_captured = " << N_count_1 << endl;
}
runTime.write();
Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
   << " ClockTime = " << runTime.elapsedClockTime() << " s"
   << nl << endl;
}
langfile.close();
Info<< "End\n" << endl;

return(0);
```

---

**D.2 Run (Case) for Stent Implant**

**D.2.1 0 (initial conditions file)**

**D.2.1.1 epsilon file**

```cpp
/*—————————————————————————*
|        |        |
---------------------
  137

Info << "Velocity_hyd+dip[v << i << v]=" << velocity_hyd_int[i] << endl;
r_n_temp[i]=r_n[i];

if(r_n[i].x()<19){
r_n_new[i]=r_n[i]+0.1*(U_velocity_blood[i]+velocity_hyd_int[i]);
}
else{
   // Particles dont move if thy are outside the space region
   r_n_new[i]=r_n[i];
}

// Update the positions of the particles
for(int j=0; j<Npar; j++){
   for(int i=0; i<2*Nseed; i++){
      int i_x=i+1;
      int i_y=-pow(1, i);
      vector seed_centre=vector((32.0−(16.0*i_x), (6.7*i_y), 0);
      Info<< "seed_centre_" <<i<<"seed_centre_" << endl;
      if(mag(r_n_new[i]−seed_centre)<1.0 && r_n_new[i].x() < 19){
         r_n[i]=r_n_temp[i];
         N_count_1=N_count_1+1;
         break;
      }
      else{
         r_n[i]=r_n_new[i];
      }
   }
   Info<< "N_captured = " << N_count_1 << endl;
}
runTime.write();
Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
   << " ClockTime = " << runTime.elapsedClockTime() << " s"
   << nl << endl;
}
langfile.close();
Info<< "End\n" << endl;

return(0);
```

---

**D.2 Run (Case) for Stent Implant**

**D.2.1 0 (initial conditions file)**

**D.2.1.1 epsilon file**

```cpp
/*—————————————————————————*
|        |        |
---------------------
  137

Info << "Velocity_hyd+dip[v << i << v]=" << velocity_hyd_int[i] << endl;
r_n_temp[i]=r_n[i];

if(r_n[i].x()<19){
r_n_new[i]=r_n[i]+0.1*(U_velocity_blood[i]+velocity_hyd_int[i]);
}
else{
   // Particles dont move if thy are outside the space region
   r_n_new[i]=r_n[i];
}

// Update the positions of the particles
for(int j=0; j<Npar; j++){
   for(int i=0; i<2*Nseed; i++){
      int i_x=i+1;
      int i_y=-pow(1, i);
      vector seed_centre=vector((32.0−(16.0*i_x), (6.7*i_y), 0);
      Info<< "seed_centre_" <<i<<"seed_centre_" << endl;
      if(mag(r_n_new[i]−seed_centre)<1.0 && r_n_new[i].x() < 19){
         r_n[i]=r_n_temp[i];
         N_count_1=N_count_1+1;
         break;
      }
      else{
         r_n[i]=r_n_new[i];
      }
   }
   Info<< "N_captured = " << N_count_1 << endl;
}
runTime.write();
Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
   << " ClockTime = " << runTime.elapsedClockTime() << " s"
   << nl << endl;
}
langfile.close();
Info<< "End\n" << endl;

return(0);
```
FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class volScalarField;
    object epsilon;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 2 -3 0 0 0];
internalField uniform 14.855;
boundaryField
{
    inlet
    {
        type fixedValue;
        value uniform 14.855;
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type zeroGradient;
    }
    bottom
    {
        type zeroGradient;
    }
    seedtop_0
    {
        type zeroGradient;
    }
    seedbot_0
    {
        type zeroGradient;
    }
    .
    .
    .
    .
    .
    .
    seedtop_9
    {
        type zeroGradient;
    }
    seedbot_9
    {

D.2. Run (Case) for Stent Implant

```plaintext
D.2.1.2 k file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class volScalarField;
    object k;
}

dimensions [0 2 -2 0 0 0 0];
internalField uniform 0.375;
boundaryField
{
    inlet
    {
        type fixedValue;
        value uniform 0.375;
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type zeroGradient;
    }
    bottom
    {
```
D.2. Run (Case) for Stent Implant

```plaintext
  type zeroGradient;
}
seedtop_0
{
  type zeroGradient;
}
seedbot_0
{
  type zeroGradient;
}
....
....
....
....
seedtop_9
{
  type zeroGradient;
}
seedbot_9
{
  type zeroGradient;
}
front
{
  type empty;
}
back
{
  type empty;
}

// ************************************************************************* //

D.2.1.3 nuTilda file

FoamFile
{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class volScalarField;
  object nuTilda;
}
```

---

D.2.1.3 nuTilda file

```plaintext
/*------------------*- C++ -*-------------------*/

FoamFile
{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class volScalarField;
  object nuTilda;
}
```
D.2. Run (Case) for Stent Implant

dimensions [0 2 -1 0 0 0 0];
internalField uniform 0;
boundaryField
{
    inlet
    {
        type fixedValue;
        value uniform 0;
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type zeroGradient;
    }
    bottom
    {
        type zeroGradient;
    }
    seedtop_0
    {
        type zeroGradient;
    }
    seedbot_0
    {
        type zeroGradient;
    }
    ....
    ....
    ....
    ....
    seedtop_9
    {
        type zeroGradient;
    }
    seedbot_9
    {
        type zeroGradient;
    }
    front
    {
        type empty;
    }
    back
    {
        type empty;
    }
}
D.2. Run (Case) for Stent Implant

D.2.1.4 p file

```cpp
/*---------------------------------------------* C++ ---------------------------------------------*/
|
| Field OpenFOAM: The Open Source CFD Toolbox |
| Operation Version: 1.4 |
| And Web: http://www.openfoam.org |
| Manipulation |

`FoamFile`{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class volScalarField;
  object p;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 2 -2 0 0 0];
internalField uniform 0;
boundaryField{
  inlet{
    type zeroGradient;
  }
  outlet{
    type fixedValue;
    value uniform 0;
  }
  top{
    type zeroGradient;
  }
  bottom{
    type zeroGradient;
  }
  seedtop_0{
    type zeroGradient;
  }
  seedbot_0{
    type zeroGradient;
  }
  ....
  ....
  ....
  seedtop_9
```
D.2. Run (Case) for Stent Implant

D.2.1.5 R file

```foam
FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";

    class volTensorField;
    object R;
}

// ************************************************************************* //
dimensions [0 2 -2 0 0 0 0 0];
internalField uniform (0 0 0 0 0 0 0 0);
boundaryField
{
    inlet
    {
        type fixedValue;
        value uniform (0 0 0 0 0 0 0 0);
    }
    outlet
    {
        type zeroGradient;
    }
```
D.2. Run (Case) for Stent Implant

```
} top
{ type zeroGradient;
} bottom
{ type zeroGradient;
} seedtop_0
{ type zeroGradient;
} seedbot_0
{ type zeroGradient;
} .... .... ....
} seedtop_9
{ type zeroGradient;
} seedbot_9
{ type zeroGradient;
} front
{ type empty;
} back
{ type empty;
}

// ************************************************************************* //

D.2.1.6 T file

FoamFile{

version 2.0;
format ascii;
```
root **""**;
case **""**;
instance **""**;
local **""**;

class volScalarField;
object T;

}  

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 0 0 1 0 0 0];
internalField uniform 0;
boundaryField
{
  inlet
  {
  }
  outlet
  {
  }
  top
  {
  }
  bottom
  {
  }
  seedtop_0
  {
  }
  seedbot_0
  {
  }
  seedtop_9
  {
  }
  seedbot_9
  {
  }
  front
  {

}
D.2. Run (Case) for Stent Implant

D.2.1.7 Blood velocity file, U

```c++
FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class volVectorField;
    object U;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 1 -1 0 0 0 0];
internalField uniform (0 0 0);
boundaryField
{
    inlet
    {
        type parabolicVelocity;
        maxValue 1.5;
        n (1 0 0);
        y (0 0 0);
        value nonuniform List<vector>
        (0.0217952 0 0)
        (0.0649072 0 0)
        (0.0649072 0 0)
        ...
        (0.0649072 0 0)
        (0.0649072 0 0)
        (0.0217952 0 0)
    );
}
```
D.2. Run (Case) for Stent Implant

```plaintext
outlet
{
    type zeroGradient;
}
top
{
    type fixedValue;
    value uniform (0 0 0);
}
bottom
{
    type fixedValue;
    value uniform (0 0 0);
}
seedtop_0
{
    type fixedValue;
    value uniform (0 0 0);
}
seedbot_0
{
    type fixedValue;
    value uniform (0 0 0);
}
seedtop_9
{
    type fixedValue;
    value uniform (0 0 0);
}
seedbot_9
{
    type fixedValue;
    value uniform (0 0 0);
}
front
{
    type empty;
}
back
{
    type empty;
}
}

D.2.1.8 Magnetic Field, H

/*—————————————————————————*
 |                                  |
 | ==========                      |
 | / F ield                         |
 | OpenFOAM: The Open Source CFD Toolbox |
 | / O peration                     |
 | Version: 1.4                    |
*/
FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class volVectorField;
    object H;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 -1 0 0 1 0];
internalField uniform (0 0 0);
boundaryField
{
inlet
{
    type zeroGradient;
}
outlet
{
    type zeroGradient;
}
top
{
    type zeroGradient;
}
bottom
{
    type zeroGradient;
}
seedtop_0
{
    type fixedGradient;
    gradient uniform (0 0 0);
}
seedbot_0
{
    type fixedGradient;
    gradient uniform (0 0 0);
}
....
....
....
....
seedtop_9
{
    type fixedGradient;
    gradient uniform (0 0 0);
}
seedbot_9
D.2. Run (Case) for Stent Implant

```cpp
{
    type fixedGradient;
    gradient uniform (0 0 0);
}

defaultFaces
{
    type empty;
}
```

D.2.1.9 Uniform Field in the Model

```cpp
FoamFile
{
    version 2.0;
    format ascii;
    root ""
    case ""
    instance ""
    local ""
    class volVectorField;
    object space_H_0;
}
```

```cpp
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 -1 0 0 0 1 0];
internalField uniform (0 0 0);
boundaryField
{
    inlet
    {
        type zeroGradient;
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type zeroGradient;
    }
    bottom
    {
        type zeroGradient;
    }
    seedtop_0
```
D.2. Run (Case) for Stent Implant

{  
  type fixedGradient;
  gradient uniform (0 0 0);
}

seedbot_0
{  
  type fixedGradient;
  gradient uniform (0 0 0);
}

seedtop_9
{  
  type fixedGradient;
  gradient uniform (0 0 0);
}

seedbot_9
{  
  type fixedGradient;
  gradient uniform (0 0 0);
}

defaultFaces
{  
  type empty;
}

D.2.1.10 Modification to the Magnetic Flux Density

FoamFile
{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class volVectorField;
  object modif_B;
}

dimensions [ 1 0 -2 0 0 -1 0];

internalField uniform (0 0 0);
D.2.2. Run (Case) for Stent Implant

```latex
boundaryField
{
    inlet
    {
        type zeroGradient;
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type zeroGradient;
    }
    bottom
    {
        type zeroGradient;
    }
    seedtop_0
    {
        type fixedGradient;
        gradient uniform (0 0 0);
    }
    seedbot_0
    {
        type fixedGradient;
        gradient uniform (0 0 0);
    }
    seedtop_9
    {
        type fixedGradient;
        gradient uniform (0 0 0);
    }
    seedbot_9
    {
        type fixedGradient;
        gradient uniform (0 0 0);
    }
    defaultFaces
    {
        type empty;
    }
}
```

D.2.1.11 Magnetisation in Avilés Model

```latex
/\*
  \begin{align*}
  & \text{Field} \\
  & \text{OpenFOAM: The Open Source CFD Toolbox}
\end{align*}
*/
```
D.2. Run (Case) for Stent Implant

FoamFile
{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class volVectorField;
  object M_aviles;
}

/// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 -1 0 0 1 0];
internalField uniform (0 0 0);
boundaryField
{
inlet
{
  type zeroGradient;
}
outlet
{
  type zeroGradient;
}
top
{
  type zeroGradient;
}
bottom
{
  type zeroGradient;
}
seedtop_0
{
  type fixedGradient;
  gradient uniform (0 0 0);
}
seedbot_0
{
  type fixedGradient;
  gradient uniform (0 0 0);
}....
....
....
seedtop_9
{
  type fixedGradient;
  gradient uniform (0 0 0);
}
D.2. Run (Case) for Stent Implant

seedbot.9
{
    type fixedGradient;
    gradient uniform (0 0 0);
}

defaultFaces
{
    type empty;
}

D.2.1.12 Magnetic Moment in the Model

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";

class volVectorField;
object moment_Cregg;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions [0 2 0 0 1 0];
internalField uniform (0 0 0);
boundaryField
{
    inlet
    {
        type zeroGradient;
    }
    outlet
    {
        type zeroGradient;
    }
    top
    {
        type zeroGradient;
    }
    bottom
    {
        type zeroGradient;
    }
D.2. Run (Case) for Stent Implant

D.2.2 constant

D.2.2.1 Polymesh File (blockMeshDict) file

```plaintext
// Parameters :
//       convertToMeters (convertToMeters) : 1.0
//       grading (grading) : 16
//       Radius of seed (R_s) : 1
//       Number of seeds (n) : 10
//       Width between seeds (delta) : 30.0
//       Width of annulus around seed (a) : 0.838477631

FoamFile
{

}
D.2. Run (Case) for Stent Implant

version 2.0;
format ascii;
root "";
case "";
instance "";
local "";
class dictionary;
object blockMeshDict;

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
Please see the mesh generator in Appendix C.

convertToMeters 1.0;

vertices 580 {
};
edges {
};
blocks {
};
patches {
};

D.2.2.2 physical properties file

FoamFile {
version 2.0;
format ascii;
root "";
case "";
instance "";
local "";
class dictionary;
object physicalProperties;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
kg m s K ml A cd
D.2 Run (Case) for Stent Implant

theta [ 0 0 0 0 0 0 0 90.0;  // H-field angle (in degrees)
B_0 [ 1 0 -2 0 0 0 0 0.06;  // magnitude of applied B
mu_0 [ 1 1 -2 0 0 0 0 0.256637062 0e-07;  // magnetic permeability of free space
R_i [ 0 1 0 0 0 0 0 62.5e-06;  // implant radius
R_i_scale [ 0 0 0 0 0 0 0 0;  // implant radius
chi_i_0 [ 0 0 0 0 0 0 0 100;  // magnetic susceptibility of implant H_0=0
chi_i [ 0 0 0 0 0 0 0 0;  // magnetic susceptibility of medium
M_i_s [ 1 1 -2 0 0 0 0 1.261 0 0;  // Saturation magnetization of implant
M_fm_p_s [ 0 0 0 0 0 0 0 1.3519e00;  // Saturation magnetization of mat. in MDCP
moment_fm_p_s [ 0 2 0 0 0 1 0 2.03e-19;  // magnetic moment of the mat. in MDCP
rho_beta [ 1 -3 0 0 0 0 0 1000;  // blood density
rho_fm_p [ 1 -3 0 0 0 0 0 5050;  // density of in MDCP
rho_pol_p [ 1 -3 0 0 0 0 0 950;  // density of in MDCP
x_FM_p [ 0 0 0 0 0 0 0 0.25;  // mass fraction of fer. mat. in MDCP
w_FM_p [ 0 0 0 0 0 0 0 0.064;  // volume fraction of fer. mat. in MDCP
R_p [ 0 1 0 0 0 0 0 435.0e-9;  // MDCP radius
R_pv [ 0 1 0 0 0 0 0 0.5e-3;  // blood vessel radius
u_0 [ 0 1 -3 0 0 0 0 0.021;  // average inlet velocity
eta_beta [ 1 -1 1 0 0 0 0 0.001;  // blood viscosity (Pa s)
epsilon_p [ 0 0 0 0 0 0 0 0;  // porosity of the material
k_B [ 1 2 -2 0 0 0 0 1.38e-23; // porosity of the material
T_L [ 0 0 0 1 0 0 0 300.0; }

D.2.2.3 transport properties file

FoamFile
{
version 2.0;
format ascii;
root "";
case "";
instance "";
local "";
class dictionary;
object transportProperties;
}
transportModel Newtonian;
mu [ 0 2 -1 0 0 0 0 0.06;  // CrossPowerLawCoeffs

CrossPowerLawCoeffs
{

mu0
[ 0 2 -1 0 0 0 0 0.06;

muInf
[ 0 2 -1 0 0 0 0 0.06;

m
[ 0 1 0 0 0 0 1;

n
[ 0 0 0 0 0 0 1; }

BirdCarreauCoeffs

/* —————————————————————————— */

FoamFile
{
version 2.0;
format ascii;
root "";
case "";
instance "";
local "";
class dictionary;
object transportProperties;
}
transportModel Newtonian;
mu = [ 0 2 -1 0 0 0 0 0.06;  // CrossPowerLawCoeffs

CrossPowerLawCoeffs
{

mu0
[ 0 2 -1 0 0 0 0 0.06;

muInf
[ 0 2 -1 0 0 0 0 0.06;

m
[ 0 1 0 0 0 0 1;

n
[ 0 0 0 0 0 0 1; }

BirdCarreauCoeffs

156
D.2. Run (Case) for Stent Implant

\{
  \nu0 \quad \nu0 \quad \begin{bmatrix} 0 & 2 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \times 10^{-6};
  \nuInf \quad \nuInf \quad \begin{bmatrix} 0 & 2 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \times 10^{-6};
  k \quad k \quad \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times 0;
  n \quad n \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times 1;
\}

// ************************************************************************* //

D.2.2.4 turbulence properties file

D.2.3 system

D.2.3.1 controlDict file

FoamFile
{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class dictionary;
  object turbulenceProperties;
}

// ************************************************************************* //

FoamFile
{
  version 2.0;
  format ascii;
  root "";
  case "";
  instance "";
  local "";
  class dictionary;
  object turbulenceProperties;
}

// ************************************************************************* //
D.2. Run (Case) for Stent Implant

```cpp
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
20
startFrom startTime;
startTime 3;
stopAt endTime;
endTime 60;
deltaT 0.001;
writeControl timeStep;
writeInterval 1500;
purgeWrite 0;
writeFormat ascii;
writePrecision 6;
writeCompression uncompressed;
timeFormat general;
timePrecision 6;
runTimeModifiable yes;
libs ("libmyBCs.so");
// ************************************************************************* //

D.2.3.2 fvSchemes file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class dictionary;
    object fvSchemes;
}
// *********************************************** //

ddtSchemes
{
    default steadyState;
}
gradSchemes
{
    default Gauss linear;
}
```
D.2. Run (Case) for Stent Implant

grad(p) Gauss linear;
grad(U) Gauss linear;
}
divSchemes
{
default none;
div(phi,U) Gauss upwind;
div(phi,k) Gauss upwind;
div(phi,epsilon) Gauss upwind;
div(phi,R) Gauss upwind;
div(R) Gauss linear;
div((nuEff*dev(grad(U).*T()))) Gauss linear;
}
laplacianSchemes
{
default none;
laplacian(nuEff,U) Gauss linear corrected;
laplacian((1|A(U)).p) Gauss linear corrected;
laplacian(DkEff,k) Gauss linear corrected;
laplacian(DepsilonEff,epsilon) Gauss linear corrected;
laplacian(DREff,R) Gauss linear corrected;
laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
}
interpolationSchemes
{
default linear;
interpolate(U) linear;
}
snGradSchemes
{
default corrected;
}
fluxRequired
{
default no;
p;

// ************************************************** */

D.2.3.3 fvSolution file

/* ============================================================== */
| ================== | OpenFOAM: The Open Source CFD Toolbox |
| \ / F ield         | Version: 1.4 |
| \ / O peration     | Web: http://www.openfoam.org |
| \ / A nd           | |
| \ / M anipulation  | |
| \*-----------------------------------------------------------------*/

FoamFile
{
  version 2.0;
  format ascii;
}
root "";
case "";
instance "";
local "";
class dictionary;
object fvSolution;
}

// ............................................................ //
solvers {
P PCG {
    preconditioner DIC;
tolerance 1e−06;
    relTol 0.01;
};
P BiCG {
    preconditioner DILU;
tolerance 1e−05;
    relTol 0.1;
};
k BiCG {
    preconditioner DILU;
tolerance 1e−05;
    relTol 0.1;
};
epsilon BiCG {
    preconditioner DILU;
tolerance 1e−05;
    relTol 0.1;
};
R BiCG {
    preconditioner DILU;
tolerance 1e−05;
    relTol 0.1;
};
uTilda BiCG {
    preconditioner DILU;
tolerance 1e−05;
    relTol 0.1;
};
}
SIMPLE {
    nNonOrthogonalCorrectors 1;
}

relaxationFactors {
p 0.3;
}
D.2. Run (Case) for Stent Implant

U 0.7;
k 0.7;
epsilon 0.7;
R 0.7;
muTilda 0.7;

// ************************************************************************* //

D.2.3.4 sampleDict file

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class dictionary;
    object sampleDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

interpolationScheme cellPoint;
writeFormat raw;
sampleSets
{
    uniform
    {
        name leftPatch;
        axis y;
        start (0 0.5 0.25);
        end (0 2 0.25);
        nPoints 100;
    }
};
fields
{
    sigmaxx

D.2. Run (Case) for Stent Implant

D.2.3.5 twoRegionDict file

```
FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class dictionary;
    object twoRegionDict;
}

// ************************************************************************* //
space
mu 1.0;
shift_x 6.0; // half of the distance between the center of seeds on X
shift_y 6.7; // Position of the center of seed on Y
x_1_original -307.90; // position of particle 1 on X axis
y_1_original 5.75; // position of particle 1 on Y axis
x_2_original 0.15; // position of the particle 2 on X axis
y_2_original 0.20; // position of the particle 2 on Y axis
```
Appendix E

Publications

To date this work has resulted in two peer reviewed journal papers and a third paper (in collaboration with CRANN, Trinity College Dublin) is under review with the *Journal of Magnetism and Magnetic Materials* (JMMM), and three international conference poster presentations, details of which are given below:


Inclusion of Interparticle Interactions in the Modelling of Magnetic Drug Targeting

P. J. Cregg*, Kieran Murphy & Adil Mardinoglu
Materials Characterisation & Processing Group, Waterford Institute of Technology, Waterford, Ireland.

E-mail: pjcgregg@wit.ie Phone: +353-51-302651 Fax: +353-51-302660

Introduction

Magnetic nanoparticles continue to offer much promise as carriers in drug targeting systems [1].

That the force exerted on an individual particle is determined by the gradient of the field and not simply the field is well known [2-6].

As has been pointed out by several authors [2-6] this may inhibit the targeting, by means of external magnets, of areas deep within the body.

With this in mind, the implications of ferromagnetic materials, such as wires, in blood vessels, in order to create large localised gradients within the vessels, has been proposed by some authors [2-6].

Berry [7] has suggested that magnetic nanoparticles <50nm may have advantages as drug carriers.

Here we follow the model of Ritter et al. [4] but consider nanoparticles and use the Langevin function to describe the average magnetic moment of each superparamagnetic nanoparticle. We consider the case of interparticle exchange interaction and model this in the limit of high exchange by appropriate Langevin functions. With these changes, the simulations of Ritter et al. are then followed using OpenFOAM.

Keywords: Magnetic drug targeting; High gradient magnetic separation; Exchange interaction; Simulation.

Collection Efficiency (CE)

Collection Efficiency (CE) is determined following Ritter by considering the behaviour of the magnetic Stokes drag:

\[ F = \frac{8}{3} \pi \eta \frac{p}{s} v \]

where \( \eta \) is the viscosity of the blood, \( p \) the radius of the particle, \( v \) the velocities of the blood and the particle respectively. The blood velocity, \( v \), is determined by solving the Navier-Stokes-Stokes equations.

Magnetic force:

\[ F_m = (m \cdot \nabla) B \]

where \( B \) is the resulting magnetic flux density (due to external \( H \) and the presence of the wire) and \( m \) is the magnetic moment of the particle. Ritter et al. considered microparticles where the axis of the moment lay along that of \( B \), and the magnetisation increased linearly with \( H \) until saturation. Whereas, nanoparticles are saturated single domains, typically superparamagnetic and experience thermal agitation, so that the magnetisation is given by the Langevin equation for magnetic fluids [2,10].

\[ m = \omega_{L,ex} V_p M_s L(\beta) \]

where

\[ L(\beta) = \frac{1}{1 + 3 \beta} \]

and

\[ J = \frac{\mu_0 V_p M_s H}{k_B T} \]

Here \( \omega_{L,ex} \) is volume fraction of ferromagnetic material in the particle, \( V_p \) is the particle volume, \( M_s \) the saturation magnetisation. The two models for the magnetisation are given in Figure 2(a).

The \( H \) field is calculated from solving the Laplacian as indicated by Ritter, ensuring continuity of flux and potential, across the wire-blood interface.

Inclusion of Exchange Interactions

Rancourt [8] refers to the role of inter-particle exchange bridges between nanoparticles. From the point of view of altered magnetisation, this interaction can be treated as follows.

The Langevin equation is derived from the partition function \( Z \). Recently one of the authors has considered the effect of two identical particles interacting via exchange and has reduced \( Z \) from a quadruple integral to an infinite sum of known functions (expressible in terms of the Langevin function), from which \( m \) can be calculated. It is hoped to present this elsewhere in the very near future.

However, at this stage it is possible to consider in terms of simple Langevin functions the effect of very large exchange interaction. For two identical particles not interacting, the appropriate reduced magnetisation is

\[ M_{0,ex}(\beta) = 1 - 4 \beta \]

Whereas, two identical particles in the limit of very large exchange interaction \( (J \rightarrow \infty) \) behave as one moment with twice the magnitude \( m \).

\[ M_{0,ex,J=\infty}(\beta) = 2(1 - 4 \beta) \]

These are shown on Figure 2(b), as well as a three term approximation obtained by others, which shows close agreement to the exact partition function integral value for \( \beta = J V_p M_s^2 k_B T = 1 \).

Conclusions

• The per unit volume magnetisation predicted by the Langevin function for magnetic fluids is larger than that of the microparticle model used by Ritter et al.

• Hence, the calculated collection efficiency in the modified model is increased for nanoparticles.

References


This work was funded by Enterprise Ireland under the Applied Research Enhancement (ARE) program as part of the South Eastern Applied Materials (SEAM) Research Centre. AM is grateful to the Cancer Research Ireland (Irish Cancer Society) for an Oncology Scholars Travel Award and to the conference organisers for free registration.

6th International Conference on the Scientific and Clinical Applications of Magnetic Carriers

May 17th – 20th 2006, Krems, Austria.
CALCULATION OF DIPOLE INTERACTIONS IN MAGNETIC DRUG TARGETING

P. J. Cregg*, Kieran Murphy & Adil Mardinoglu
Materials Characterisation & Processing S.E.A.M, Waterford Institute of Technology, Ireland.
E-mail: pjcregg@wit.ie Phone:+353-51-843210 Fax:+353-51-843666

Abstract

The magnetic targeted drug delivery system of Aviles, Elner and Ritter [11], which uses SS 409 as the seed ferromagnetic material and iron for the magnetic drug carrier particles, is considered. Agglomeration of the particles is known to occur in such systems, and the effect of magnetic (dipole) interactions between the particles is included. Hydrodynamic and dipole interactions were calculated previously by Mikkelsen et al. [12] under low magnetic fields. Here, for higher magnetic fields, the effect of the magnetic interactions between the two particles is calculated using reference particle and particle tracking. The calculations were performed with the open source software OpenFOAM. The system performance is assessed in terms capture cross section [11]. In the simulations agglomeration is seen to occur leading to larger capture cross section.

Keywords: Magnetic drug targeting, High gradient magnetic separation, Dipole interaction, Simulation.

Introduction

- Magnetic nanoparticles continue to offer much promise as carriers in drug targeting systems [3].
- That the force exerted on an individual particle is determined by the gradient of the field and not simply the field is well known [4-6].
- As has been pointed out by several authors [2-6] this may inhibit the targeting, by means of external magnetic, of areas deep within the body.
- With this in mind, the implanting of ferromagnetic materials, such as wires, seeds, in blood vessels, in order to create large localised gradients within the vessels, has been proposed by some authors [3-6].
- Berry [7] has suggested that magnetic nanoparticles <50 nm may have advantages as drug carriers.
- Here we follow the model of Ritter and co-workers [4-6] and investigate the Capture Cross Sections for nanoparticles.
- We note the significant beneficial role that agglomeration near the seed might play in increasing the magnetic force and subsequent de-agglomeration [5].
- As the particles are small and undergo significant thermal fluctuations, we use the Langevin function to describe the magnetic moment of each particle.
- Previously Mikkelsen et al. included the dipole interactions for the case of low magnetic fields [12]. Here we adapt and extend their approach to model two interacting nanoparticles, with arbitrary field strength.

![Diagram of the system](image)

**Figure 1:** Schematic diagram of the control volume, CV, used in determining the capture radius, λ, of the magnetic nanoparticles.

Capture Cross Section (CCS) is determined following Aviles et al. [11] by considering the behaviour of the magnetic nanoparticles under the influence of Stokes drag and the magnetic forces.

**Stokes drag:**

\[ F_s = 6 \pi \eta R \left( v_s - v_p \right) \]

where \( \eta \) is the viscosity of the blood, \( R \) is the radius of the particle, \( v_s \) and \( v_p \) are the velocities of the blood and the particle respectively. The blood velocity \( v_s \) is determined by solving the appropriate Navier-Stokes equations.

**Magnetic force:**

\[ F_m = \left( \mu_0 m \right) \cdot B \]

where \( \mu_0 \) is the magnetic flux density (due to external H, magnetic field and the presence of the implant (seed)) and \( m \) is the magnetic moment of the particle. As a result of thermal agitation, the magnetic moment of such a particle does not, in general, align with the external field. However, the average projection of the moment in the direction of \( B \) can be calculated from the Langevin function [2,15,16].

\[ \langle m \rangle = \left( \mu_0 m \cdot B \right) B \]

\[ \langle m \rangle = \left( \mu_0 m \cdot B \right) B \]

\[ B = \frac{\mu_0 v_p M_{1110,0,0} B}{kT} \]

where \( \omega_m \) is the volume fraction of ferromagnetic material in the particle, \( v_p \) is the particle volume, \( M_{1110,0,0} \) (the volume) saturation magnetisation, \( B \) is the magnetic field, \( k \) is Boltzmann’s constant and \( T \) is the absolute temperature, so that \( m \) can be written as

\[ m = \omega_m v_p M_{1110,0,0} B \]

\[ F_{m} = \left( \mu_0 m \right) \cdot B \]

For calculation of \( B \), field, Laplace’s equation for the scalar potential is solved analytically by separation of variables [13,14].

**Dipole (Dipole) Interaction Force:**

Including the magnetic effect of particle 2 on particle 1, the magnetic force can be augmented to

\[ F_{m} = \left( \mu_0 m \right) \cdot B + \left( \mu_0 m \right) \cdot B \]

where \( m_B \) is the modification of the resulting magnetic flux density due to particle 2 at \( r_2 \). The nanoparticles are taken as spherical with radius \( a \) and sufficiently small so that the magnetic flux can be taken as homogeneous over the particle. The modification to the magnetic flux density is thus taken as

\[ m_B = \frac{1}{4 \pi} \left( \frac{\mu_0 m_2 B}{r_2} \right) \]

\[ \frac{m_2 B}{r_2} \]

where \( r_1 \) and \( r_2 \) are the positions of the particles and \( m_B \) is the change in the magnetic moment of the particle 1 given by

\[ m_B = \frac{1}{4 \pi} \left( \frac{\mu_0 m_2 B}{r_2} \right) \]

\[ \frac{m_2 B}{r_2} \]

Capture Cross Section (CCS)

The particle trajectories are obtained from evaluating the streamline function

\[ \frac{\partial m}{\partial x} = -v_B \]

\[ m_B \]

\[ \frac{\partial m}{\partial y} = v_B \]

\[ \frac{\partial m}{\partial z} = v_B \]

where \( x \) is the stream function, and \( v_B \) and \( v_B \) are the components of particle velocity \( v_B \). The system performance of this model is calculated in terms of the capture cross section, \( \lambda_c \), defined as (see Fig.1)

\[ \lambda_c = \frac{\rho_B}{\rho_m} \]

where \( \rho_B \) is the capture radius of the ferromagnetic seed. The capture radius, \( \rho_B \), is defined by the location of the streamline at the entrance to the CV of the last magnetic drug carrier particle captured to the seed.

Conclusions

- The effect of the dipole interaction between two nanoparticles on the calculated magnetic force in the implant assisted magnetic drug targeting system of Aviles et al. is considered.
- Increased capture cross section in the modified model is observed in simulations.
- It should be possible to extend this approach to model dipole interactions in further implant arrangements such as stents.

References

INCLUSION OF MAGNETIC DIPOLE-DIPOLE INTERACTION IN IMPLANT ASSISTED MAGNETIC DRUG TARGETING

P. J. Cregg*, Kieran Murphy & Adil Maridinogho
Materials Characterisation & Processing Group, SEAMC Centre, Waterford Institute of Technology, Ireland.
E-mail: pjcregg@wits.ie Phone: +353-51-846231 Fax: +353-51-832666

Abstract

The magnetic targeted drug delivery system of Arico et al. [6], which uses SS 409 as the seed ferromagnetic material and iron for the magnetic drug carrier particles, is considered. Here in order to model of agglomeration of particles is known to occur in the system and the magnetic dipole-dipole interaction is included. Interactions were calculated perfectly using Mladenoff et al. [7] under low magnetic fields (0.1 T) and dipole-dipole interaction of two particles by using a high magnetic field (0.7 T) is calculated in our calculated paper [8]. Here, dipole-dipole interaction of N multiple magnetic nanoparticles with a seed implant using same magnetic field (0.7 T) is calculated by particle tracking. The calculations were performed with the open source software OpenFoam. Initial model parameters and the system performance is assessed in terms capture cross section [9]. In the simulations agglomeration is seen to occur leading to larger and smaller capture cross section according to the position of the particles.

Keywords: Magnetic drug targeting, High gradient magnetic separation, Dipole-dipole, Simulation.

Introduction

- Magnetic nanoparticles continue to offer much promise as carriers in drug targeting systems [4].
- That the force exerted on an individual particle is determined by the gradient of the field and not simply the field is well known [1, 6].
- As has been pointed out by several authors [2, 6] this may inhibit the targeting, by means of external magnets, of areas deep within the body.
- With this in mind, the implanting of ferromagnetic materials, such as wires, seeds, in blood vessels, in order to create large localised gradients within the vessels, has been proposed by some authors [3, 6].
- Berry [7] has suggested that magnetic nanoparticles ≤ 50 nm may have advantages as drug carriers.
- Here we follow the model of Ritter and co-workers [4-6.8] and investigate the Capture Cross Sections for multiple nanoparticles.
- We note the significant beneficial role that agglomeration near the seed might play in increasing the magnetic force and subsequent de-agglomeration [5].
- As the particles are small and undergo significant thermal fluctuations, we use the Langevin function to describe the magnetic moment of each particle [10].
- Mladenoff et al. included the magnetic dipole–dipole interactions for the case of low magnetic fields [9].
- Previously in our recently submitted paper Cregg et al. [11] included the dipole interactions for two particles. Here we extend our approach to model N interacting nanoparticles, beginning here with the presentation of results for those nanoparticles with arbitrary initial positions.

Magnetic (Dipole-Dipole) Interaction Force:

\[ F_{dd} = \frac{BM_{d}}{2} \frac{V_{p}}{r^3} \]

where \( B \) is the magnetic field, \( M_{d} \) is the total magnetic flux density due to external \( H \) magnetic field, the presence of the implant (seed) and the presence of the magnetic nanoparticles and \( r \) is the magnetic moment of the particle. As a result of thermal agitation, the magnetic moment of such a particle does not, in general, align with the external field. However, the average projection of the moment in the direction of \( B \) can be calculated from the Langevin function [2, 12, 13]:

\[ L(\beta) = \text{coth}(\beta) \frac{1}{\beta} \]

with Langevin argument

\[ \beta = \frac{M_{d} V_{p} B}{k_{b} T} \]

where \( M_{d} \) is the volume fraction of ferromagnetic material in the particle \( V_{p} \) is the particle volume, \( M_{d} V_{p} B \) is the (volume) saturation magnetisation, \( B \) is the Boltzmann's constant and \( T \) is the absolute temperature, so that the average projection of the magnetic moment of a nanoparticle in the direction of the magnetic field can be written as

\[ m = \frac{M_{d} V_{p} B}{k_{b} T} L(\beta) \]

For calculation of \( B \) field, Laplace’s equation for the scalar potential is solved analytically by separation of variables [10].

Magnetic (Dipole-Dipole) Interaction Force:

Including the magnetic effect on particle n of the other \( N-1 \) particles, the magnetic force can be augmented to

\[ F_{dd}(n) = (m_{n} \times \nabla) \left( \frac{B_{m}}{2} \right) \sum_{j=1}^{N} \frac{V_{j}}{r_{j}^3} \]

where \( B_{m} \) is the total magnetic flux acting on particle n (for particle n, \( B_{m} \) is taken as \( B_{m} = B + B_{d} \)), \( B_{d} = -\frac{1}{2} \sum_{j} B_{m}|_{\nabla \times} \left( \sum_{i \neq j} \frac{m_{i} V_{i}}{r_{ij}^3} \right) \), and \( B_{d} \) is the modification of the resulting magnetic flux density due to particle n at \( r \). The nanoparticles are taken as spherical with radius \( a \) and sufficiently small so that the magnetic flux can be taken as homogenous over the particle. The modification to the magnetic flux density is thus taken as

\[ B_{d}(r) = \frac{1}{r} \sum_{n} \left( \frac{m_{n} \times \nabla}{r_{n}} \right) \frac{V_{n}}{r_{n}^3} \]

Figure 1: Schematic diagram of the control volume, \( C \), used in determining the capture radius, \( \lambda_{c} \), of the magnetic nanoparticles.

Figure 2a: Capture Cross Section (CCS)

Figure 2b: Capture Cross Section (CCS)

Figure 2c: Capture Cross Section (CCS)

Figure 2d: Capture Cross Section (CCS)

Figure 2e: Capture Cross Section (CCS)

Figure 2f: Capture Cross Section (CCS)

Figure 2g: Capture Cross Section (CCS)

Figure 2h: Capture Cross Section (CCS)

Figure 2i: Capture Cross Section (CCS)

Conclusions

- The effect of the dipole interaction between N multiple nanoparticles on the calculated magnetic force in the implant assisted magnetic drug targeting system of Arico et al. is considered.
- Depending on the initial positions of the particles increased and decreased capture cross section in the modified model is observed in simulations.
- It should be possible to extend this approach to model dipole-dipole interactions in further implant arrangements such as stents.

References


Joint European Magnetic Symposia (JEMS08), September 14th – 19th, 2008, Dublin, Ireland.

This work was funded by Enterprise Ireland under the Applied Research Enhancement (ARE) programme as part of the South Eastern Applied Materials (SEAM) Research Centre.
Current Perspectives
Calculation of nanoparticle capture efficiency in magnetic drug targeting
P.J. Cregg*, Kieran Murphy, Adil Mardinoglu
SEAM Research Centre, Materials Characterisation and Processing Group, Waterford Institute of Technology, Waterford, Ireland

Abstract
The implant assisted magnetic targeted drug delivery system of Avilés, Ebner and Ritter, which uses high gradient magnetic separation (HGMS) is considered. In this 2D model large ferromagnetic particles are implanted as seeds to aid collection of multiple domain nanoparticles (radius \(\sim 200\) nm). Here, in contrast, single domain magnetic nanoparticles (radius in 20–100 nm) are considered and the Langevin function is used to describe the magnetization. Simulations based on this model were performed using the open source C++ finite volume library OpenFOAM. The simulations indicate that use of the Langevin function predicts greater collection efficiency than might be otherwise expected.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction
Magnetic nanoparticles continue to offer much promise as carriers in drug targeting systems [1,2]. That the force exerted on an individual particle is determined by the gradient of the field and not simply the field is well known [1,3–8]. As has been pointed out by several authors [3–8] this may inhibit the targeting, solely by means of external permanent magnets, of areas deep within the body. With this in mind, the implanting of ferromagnetic materials, such as wires, seeds or stents, in blood vessels has been proposed by some authors [4–7], in order to create large localised gradients within the vessels. Berry [2] has suggested that magnetic nanoparticles with radii of the order of 50 nm may have advantages as drug carriers, and here these are taken as the carriers. For a related problem, Furlani and Furlani [9] have developed a model for which it was possible to obtain an analytical expression for the behaviour of multifunctional particles. In contrast, the approach taken here is largely numerical in that while the magnetic field is obtained from an analytical expression both the fluid flow and resulting particle trajectories are obtained using OpenFOAM a finite volume simulation C++ library.

2. Outline of model
Ebner, Ritter and co-workers [5,6,10] have proposed various implant systems. Here, we consider the system which employs a spherical ferromagnetic implant with radius of order 1 \(\mu\)m, which they term a seed [10]. We follow their 2D model which represents a slice through the centre of the seed. It should be noted that physically this corresponds to a 2D description of flow in a rectangular box with a transverse cylindrical wire, both of infinite extent.

The model treats the behaviour of magnetic particles under the influence of Stokes drag and the magnetic force. Other forces such as inertia and gravity are ignored. The Stokes drag is given by

\[
F_s = 6\pi\eta_b R_p (\vec{v}_b - \vec{v}_p),
\]

where \(\eta_b\) is the viscosity of the blood, \(R_p\) the radius of the particle, and \(\vec{v}_b\) and \(\vec{v}_p\) are the velocities of the blood and the particle, respectively. The blood velocity, \(\vec{v}_b\), is determined by solving the appropriate Navier–Stokes equations. The magnetic force is determined by

\[
F_m = (\vec{m} \cdot \nabla)B,
\]

where \(B\) is the magnetic flux density and \(\vec{m}\) is the magnetic moment of the particle. We follow Avilés et al. [10] and consider the effect of a magnetisable seed placed in the blood flow as indicated in Fig. 1. The resulting magnetisation of the seed, \(\vec{M}_{\text{seed}}\).
is parallel to the externally applied magnetic field, $H_0$, and can be calculated from

$$M_{\text{seed}} = 2a_{\text{seed}} H_0,$$

(3)

where $a_{\text{seed}}$ is the demagnetising factor for an infinitely long cylinder in a perpendicular field taken as

$$a_{\text{seed}} = \min \left( \frac{Z_{\text{seed,0}}}{Z_{\text{seed,0}} + \frac{M_{\text{seed,1}}}{2H_0}} \right),$$

(4)

where $Z_{\text{seed,0}}$ and $M_{\text{seed,1}}$ are the zero field susceptibility and saturation magnetisation of the ferromagnetic seed, respectively, and $H_0$ is the magnitude of the external magnetic field. In the model of Avilés et al. microparticles were considered, where the axis of the moment $m$ along that of $B$, and the magnetisation increased with applied field, after accounting for demagnetising. In contrast, a nanoparticle of diameter <100 nm is typically a superparamagnetic single domain. As a result of thermal agitation, the magnetic moment of such a particle does not, in general, align with the external field. However, the average projection of the moment in the direction of $B$ can be calculated from the Langevin function [3,8,11–13]

$$L(\beta) = \coth(\beta) - \frac{1}{\beta},$$

(5)

with Langevin argument

$$\beta = \frac{\omega_{\text{FMP}} V_p M_{\text{FMP,1}} B}{k T},$$

(6)

where $\omega_{\text{FMP}}$ is the volume fraction of ferromagnetic material in the particle, $V_p$ is the particle volume, $M_{\text{FMP,1}}$ the (volume) saturation magnetisation, $B$ is the magnitude of $B$, $k$ is Boltzmann’s constant and $T$ is the absolute temperature, so that $m$ can be written as

$$m = \omega_{\text{FMP}} V_p M_{\text{FMP,1}} L(\beta) \frac{B}{B}. $$

(7)

The value of $\beta$, required to calculate the magnetic force as given by Eqs. (2) and (7) is calculated from solving the Laplace equation as outlined in Section 4.

3. Blood flow—the Navier–Stokes equations

The blood is treated as an incompressible, Newtonian, isothermal, single-phase fluid with velocity $\mathbf{v}_b$ and pressure $P$ at steady state flow. We have the continuity equation

$$\nabla \cdot \mathbf{v}_b = 0, $$

(8)

and the Navier–Stokes equation

$$\rho_b (\mathbf{v}_b \cdot \nabla \mathbf{v}_b) = \nabla P + \eta_b \nabla^2 \mathbf{v}_b, $$

(9)

where $\rho_b$ is the density of the blood. To solve Eqs. (8) and (9) a uniform inlet velocity profile is assumed at the inlet control volume (CV) such that

$$\mathbf{v}_b = \left( \begin{array}{c} \frac{\mu_0}{\rho} \\ 0 \end{array} \right),$$

(10)

where $\mu_0$ is the inlet blood velocity. Non-slip boundary conditions are applied at the seed-blood interface. In addition, symmetry boundary conditions are applied at the upper and lower CV boundaries to maintain the constant flow profile. Atmospheric pressure is assumed at the outlet of the CV to satisfy the boundary condition on pressure.

4. The magnetic force—the scalar magnetic potential

The second part of this model involves the scalar magnetic potential, which satisfies the Laplace equation over two conjoined regions: inside the seed and outside the seed. From the scalar potential, we can obtain the magnetic flux density. Thus for the two regions, within the seed and outside the seed we have magnetic flux given by

$$B = \begin{cases} \mu_0 (\mathbf{M}_{\text{seed}} + H_0 - \nabla \phi) & \text{within the seed}, \\ \mu_0 (H_0 - \nabla \phi) & \text{outside the seed}. \end{cases}$$

(11)

where $\mu_0$ is the magnetic permeability of free space, and $\phi$ represents the reduced scalar magnetic potential. Here $\mathbf{M}_{\text{seed}}$, the induced magnetisation of the seed, is obtained through Eq. (3) and $H_0$ can be written

$$H_0 = \begin{pmatrix} H_0 \cos \theta \\ H_0 \sin \theta \end{pmatrix},$$

(12)

where $\theta$ is the angle from the positive x-axis, as in Fig. 1.

Laplace’s equation for the scalar potential is solved analytically by separation of variables. Firstly, the normal component of the magnetic flux and potential are both assumed to be continuous across the seed–blood interface. Secondly, far away from the seed, the scalar potential should tend towards zero. The required analytical solution for the reduced scalar potential defined in the region outside the seed is [14]

$$\phi = \frac{H_0 R_{\text{seed}}^2}{\mu_1 + 1} \frac{1 - \frac{x \cos \theta + y \sin \theta}{\mu_1 + 1} - \sqrt{x^2 + y^2}}{x^2 + y^2},$$

(13)

where $R_{\text{seed}}$ is the radius of the seed implant, $\mu_1$ is the relative permeability of the ferromagnetic seed (calculated through $(\mu_1 - 1)/\mu_1 - 1 = a_{\text{seed}}$, from Eq. (4)), and the origin is taken as the centre of the seed.

5. Velocity equations, streamlines and capture cross section

The nanoparticles are under the influence of Stokes drag and magnetic force as given in Eqs. (1) and (2), respectively. The velocity of a nanoparticle, $\mathbf{v}_p$, can be obtained by summing the
forces acting upon it. Hence we obtain
\[ \mathbf{v}_p = \mathbf{v}_b + \frac{1}{2} \mathbf{v}_m \frac{R_{\text{seed}}}{M_{\text{seed}};sH} \nabla(\mathbf{H} \cdot \mathbf{H}). \] (14)

Here \( \mathbf{H} \) is the total magnetic field at the location of the magnetic drug carrier particle. The field \( \mathbf{H} \) relates to \( \mathbf{B} \) via
\[ \mathbf{B} = \mu_0 \mathbf{H}, \] (15)
and in the space around the seed is given by, from Eq. (11),
\[ H = H_0 - \nabla \phi. \] (16)

Therefore, the magnitude of the total magnetic field is
\[ H = \sqrt{\left(H_0 \cos \theta - \frac{\partial \phi}{\partial x}\right)^2 + \left(H_0 \sin \theta - \frac{\partial \phi}{\partial y}\right)^2}. \] (17)

and \( v_m \), the magnitude of the magnetic velocity, is given by
\[ v_m = \frac{2 R_{\text{seed}}^2 \mu_0}{9 \rho_{\text{seed}}} \omega_{\text{lim.p}} M_{\text{seed},sH} L(f). \] (18)

The volume fraction of ferromagnetic material \( \omega_{\text{lim.p}} \) in the magnetic drug carrier particle is related to its weight fraction \( x_{\text{lim.p}} \) through [5]
\[ \omega_{\text{lim.p}} = \frac{x_{\text{lim.p}}}{x_{\text{lim.p}} + (1 - x_{\text{lim.p}}) \rho_{\text{pol.p}} / \rho_{\text{pol.p}}}. \] (19)

where \( \rho_{\text{lim.p}} \) is the density of the ferromagnetic material in the magnetic drug carrier particle and \( \rho_{\text{pol.p}} \) is the density of the polymer material in the magnetic drug carrier particle.

Finally, the particle trajectories are obtained from evaluating the streamline function
\[ \frac{\partial \psi}{\partial y} = -v_{p,x}, \] (20)
\[ \frac{\partial \psi}{\partial x} = v_{p,y}, \] (21)
where \( \psi \) is the stream function, and \( v_{p,x} \) and \( v_{p,y} \) are the components of \( \mathbf{v}_p \) from Eq. (14). The system performance of this model is calculated in terms of the capture cross section, \( \lambda_c \), defined as
\[ \lambda_c = \frac{y_c}{R_{\text{seed}}}, \] (22)
where \( y_c \) is the capture radius of the ferromagnetic seed. The capture radius, \( y_c \), is defined by the location of the streamline at the entrance to the CV of the last magnetic drug carrier particle captured to the seed (Fig. 1). All calculations were performed using the open-source software finite volume library OpenFOAM [15].

6. Results and discussions

In this simulation iron was taken as the magnetic drug carrier particle and SS 409 as the seed material with 1 mm radius. The streamline functions for the capture of nanoparticles are presented in Fig. 2 for particle radius \( R_p = 50 \) nm, containing 1 wt% iron \( (x_{\text{lim.p}} = 0.4) \), under the influence of homogenous magnetic field \( \mu_0 H_0 \) oriented perpendicularly to the flow \( (\theta = \pi/2) \) with magnitudes of 0.0–0.6 T. The resulting capture cross-section, \( \lambda_c \), is

Fig. 2. Streamlines indicating the trajectories of the single domain nanoparticles, calculated using OpenFOAM, as they traverse the control volume for different magnitudes of the externally applied magnetic field, \( H_0 \). (a) \( \mu_0 H_0 = 0.0 \) T; (b) \( \mu_0 H_0 = 0.2 \) T; (c) \( \mu_0 H_0 = 0.4 \) T; (d) \( \mu_0 H_0 = 0.6 \) T.

Fig. 3. Capture cross section, \( \lambda_c \), plotted as a function of the applied magnetic field strength, \( \mu_0 H_0 \), calculated using (---) the Langevin function as appropriate for single domain particles and (...) without Langevin function as appropriate for multiple domain particles.
calculated and presented in Fig. 3 for 50 nm particles, as a function of the magnetic field strength $m_0H_0$. In the model the magnetisation of the individual nanoparticles is taken as the average value given by the Langevin function. The values of the capture cross-section predicted through use of the Langevin function are significantly larger (see Fig. 3) than would result from the large particle approach taken by Aviles et al. Beyond a field of $0.7\,\text{T}$, for the material used in this simulation, the carrier particle magnetisation is saturated for both models, leading to identical results. The relevant blood flow properties and the properties of the ferromagnetic material that are used in the magnetic drug carrier particles and for the seeds are given in Table 1.

### Table 1

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>SI Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_b$</td>
<td>1040.0</td>
<td>kg m$^{-3}$</td>
</tr>
<tr>
<td>$\eta_b$</td>
<td>0.002</td>
<td>kg m$^{-1}$ s$^{-1}$</td>
</tr>
<tr>
<td>$l_0$</td>
<td>0.001</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>$\mu_0H_0$</td>
<td>0.0–0.8</td>
<td>kg s$^{-2}$ A$^{-1}$</td>
</tr>
<tr>
<td>$\chi_{\text{sat}}$</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>$\rho_{\text{mag}}$</td>
<td>7850</td>
<td>kg m$^{-3}$</td>
</tr>
<tr>
<td>$Z_{\text{cell}}$</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>$M_{\text{sat}}$</td>
<td>1397000</td>
<td>A m$^{-1}$</td>
</tr>
<tr>
<td>$M_{\text{vec}}$</td>
<td>1735000</td>
<td>A m$^{-1}$</td>
</tr>
<tr>
<td>$R_{\text{seed}}$</td>
<td>$1.0 \times 10^{-6}$</td>
<td>m</td>
</tr>
<tr>
<td>$R_s$</td>
<td>$50 \times 10^{-9}$</td>
<td>m</td>
</tr>
<tr>
<td>$\rho_{\text{pol}}$</td>
<td>950</td>
<td>kg m$^{-3}$</td>
</tr>
</tbody>
</table>

### 7. Conclusions

The model of Aviles, Ebner and Ritter has been considered for collecting single domain magnetic drug carrier nanoparticles. Here the Langevin function is used to calculate the expected value of the nanoparticle magnetisation. Magnetic flux density $\vec{B}$ is calculated analytically by using the separation of variable solution and the blood velocity $\vec{v}_b$ is obtained from the Navier–Stokes equation using the finite volume library OpenFOAM. The simulations indicate that use of the Langevin function predicts greater collection efficiency than might be otherwise expected.

### Acknowledgements

This work was funded by Enterprise Ireland under the Applied Research Enhancement (ARE) program as part of the South Eastern Applied Materials (SEAM) Research Centre at Waterford Institute of Technology. AM is grateful to Cancer Research Ireland (Irish Cancer Society) for an Oncology Scholars Travel Award to attend the 6th International Conference on the Scientific and Clinical Applications of Magnetic Carriers, May 2006 in Krems, Austria. PJC and AM thank Armin Ebner, Axel Rosengart & Misael Aviles for helpful conversations at Krems.

### References

Inclusion of magnetic dipole–dipole and hydrodynamic interactions in implant-assisted magnetic drug targeting

P.J. Cregg *, Kieran Murphy, Adil Mardinoglu

SEAM Centre, Materials Characterisation and Processing Group, Waterford Institute of Technology, Waterford, Ireland

A R T I C L E   I N F O

Article history:
Received 18 March 2009
Received in revised form 17 June 2009
Available online 3 August 2009

Keywords:
Magnetic drug targeting
High-gradient magnetic separation (HGMS)
Magnetic nanoparticle
Magnetic seed
Dipole–dipole interaction
Hydrodynamic interaction

A B S T R A C T

Mathematical modelling of the implant-assisted magnetic drug targeting system of Avilès, Ebner and Ritter is performed. In order to model the agglomeration of particles known to occur in this system, the magnetic dipole–dipole and hydrodynamic interactions are included. Such interactions were calculated previously by Mikkelsen et al. under low magnetic fields (~0.05 T) in microfluidic systems. Here, a higher magnetic field (0.7 T) is considered and the effect of interactions on two nanoparticles with a seed implant is calculated. The calculations were performed with the open-source software OpenFOAM. Different initial positions are considered and the system performance is assessed in terms of capture cross section. Inclusion of both interactions was seen to alter the capture cross section of the system by up to 7% in absolute terms.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Magnetic nanoparticles continue to offer promise in biomedicine [1,2]. Their use in magnetic drug targeting (MDT) however is limited somewhat by the fact that the magnetic force depends specifically on the gradient of the field [3]. To overcome this problem several authors [4–12] have proposed implanting ferromagnetic materials such as wires, seeds and stents within the body. In a homogeneous magnetic field these implants create strong localised gradients, and this approach is known as implant-assisted MDT (IA-MDT). Of the various IA-MDT implants suggested by Ebner, Ritter and co-workers [6–12] we consider the micron sized spherical ferromagnetic implant, which they term a seed [9,13], with magnetic nanoparticles (50 nm radius) as the drug carriers. Considering miniaturised high-gradient magnetic separation, Mikkelsen et al. [14] have included both the hydrodynamic and dipole–dipole interactions for the case of low magnetic fields. Also, Mehasni et al. have considered the effect of magnetic dipole–dipole interaction on the performance of high-gradient magnetic separation systems [15]. Here, with a view to modelling experimentally observed agglomeration in IA-MDT [6.9–11], we adapt and extend these approaches to model two mutually interacting nanoparticles with larger field strength and a seed implant.

2. Outline of model

We consider the effect of a magnetisable seed placed in the blood flow as indicated in Fig. 1. Following Avilès et al. [9] we take the 2D approximation of a slice through the seed centre, noting that this corresponds to 2D flow in a rectangular box with a cylindrical wire, both of infinite extent. We model the behaviour of two magnetic nanoparticles under the influence of Stokes drag, a force due to hydrodynamic interaction, and a magnetic force, modified to incorporate the mutual magnetic dipole–dipole interaction [14]. The Stokes drag for Particle 1 is

$$ F_s = 6\pi \eta_b R_p (v_b - \bar{v}_{p1}) $$

where \( \eta_b \) is the viscosity of the blood, \( R_p \) the radius of the nanoparticle, \( \bar{v}_b \) and \( \bar{v}_{p1} \) the velocities of the blood and Particle 1, respectively. A similar expression applies for Particle 2. The blood velocity, \( \bar{v}_b \), is determined by solving the appropriate Navier–Stokes equations. The force due to the hydrodynamic interaction that acts on Particle 1 due to presence of Particle 2, can be written as [14],

$$ F_{hyd1} = -\frac{9\pi \eta_b R_p^2}{2} \left( \frac{p_1 - p_2}{|p_1 - p_2|^2} \right) \cdot (v_b - \bar{v}_{p1}) $$

where \( \mathbf{I} \) is the unit tensor, \( \otimes \) the vector tensor product (outer product), \( \bar{r}_1 \) and \( \bar{r}_2 \) the positions of Particle 1 and Particle 2, respectively and \( \bar{v}_{p1} \) the velocity of Particle 2. An equivalent formula applies for Particle 2 due to the presence of Particle 1.

* Corresponding author. Tel.: +353 51 302631; fax: +353 51 302666.
E-mail address: pjcregg@wit.ie (P.J. Cregg).

0304-8853/$ - see front matter \& 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.jmmm.2009.07.056
In general the magnetic force acting on a magnetic moment is determined by
\[ F_m = \frac{\mathbf{m}}{c^2} \frac{1}{r} \mathbf{B}_{\text{total}} \]
where \( \mathbf{m} \) is the magnetic moment and \( \mathbf{B}_{\text{total}} \) is the total magnetic field. Two magnetic dipoles exert a force on each other, which can be included in the magnetic force equation by considering (i) the modified magnetic field density and (ii) the modification in the magnetic moment resulting from this modified flux density. Thus for Particle 1 we can write
\[ F_m = (\mathbf{m} \cdot \nabla) \mathbf{B}_{\text{total}} \]
where \( \mathbf{m} \) is the total magnetic moment of Particle 1, \( \mathbf{B} \) the magnetic flux density due to the external field and the seed and \( \mathbf{d} \mathbf{B}_2 \) the modification of the resulting magnetic flux density at Particle 1 at \( \mathbf{r}_1 \) due to the presence of Particle 2 at \( \mathbf{r}_2 \). (An equivalent formula applies for Particle 2 with \( \mathbf{m}_2 \) and \( \mathbf{B} + \mathbf{d} \mathbf{B}_1 \).)

The value of \( \mathbf{B} \) required to calculate the magnetic force as given by Eqs. (4) and (12), is calculated from the scalar magnetic potential \( \phi \) using the Laplace equation over two conjoined regions: inside the seed and outside the seed. Thus for outside the seed we have magnetic flux density given by
\[ \mathbf{B} = \mu_0(\mathbf{H}_0 - \nabla \phi) \]
where \( \mu_0 \) is the magnetic permeability of free space, and \( \mathbf{H}_0 \) is the external magnetic field. Taking the seed centre as the origin, in Fig. 1, the reduced scalar potential, \( \phi \) in the region outside the seed is given by [13,16]
\[ \phi = \frac{H_0 R_{\text{seed}}^2 x_{\text{seed}}}{x^2 + y^2} \]
where \( H_0 \) is the magnitude of the applied field, \( R_{\text{seed}} \) radius of the seed and \( x_{\text{seed}} \) the demagnetising factor which for an infinitely long cylinder in a perpendicular field can be taken as [9]
\[ x_{\text{seed}} = \min \left( \frac{x_{\text{seed},0}}{2}, \frac{M_{\text{seed}}}{2H_0} \right) \]
where \( x_{\text{seed},0} \) and \( M_{\text{seed}} \) are the zero field susceptibility and saturation magnetisation of the seed, respectively, and \( H_0 \) can be written
\[ \mathbf{H}_0 = \left( \frac{H_0 \cos \theta}{H_0 \sin \theta} \right) \]
where \( \theta \) is the angle of the field from the positive x-axis in Fig. 1.

The average projection of \( \mathbf{m} \) the magnetic moment of a nanoparticle in the direction of \( \mathbf{B}_{\text{total}} \) (which for Particle 1 is taken as \( \mathbf{B} + \mathbf{d} \mathbf{B}_2 \) and equivalently for Particle 2 would be \( \mathbf{B} + \mathbf{d} \mathbf{B}_1 \)) can be calculated from the Langevin function [3,11,13,17–19]
\[ \mathbf{L}(\beta) = \coth(\beta) - \frac{1}{\beta} \]
with argument
\[ \beta = \frac{\omega_p V_p M_{\text{pol,p}}}{kT} \]
where \( V_p \) is the nanoparticle volume, \( M_{\text{pol,p}} \) the magnitude of \( \mathbf{B}_{\text{total}} \), \( k \) Boltzmann’s constant, \( T \) the absolute temperature, \( M_{\text{pol}} \) the saturation magnetisation of the nanoparticle, and \( \omega_p \) the volume fraction of ferromagnetic material in the nanoparticle which relates to its weight fraction \( x_p \) through [6]
\[ \omega_p = \frac{x_p}{x_p + (1 - x_p)(\rho_p/\rho_{\text{pol,p}})} \]
where \( \rho_p \) is the density of the ferromagnetic material and \( \rho_{\text{pol,p}} \) is the density of the polymer material in the nanoparticle. Thus \( \mathbf{m} \) (for either particle) can be written
\[ \mathbf{m} = \omega_p V_p M_{\text{pol,p}} \mathbf{L}(\beta) \frac{\mathbf{B}_{\text{total}}}{B_{\text{total}}} \]  

With regard to the magnetic interaction, each nanoparticle is taken as spherical and sufficiently small so that its magnetic flux can be taken as homogeneous over the particle. The dipole field of Particle 2 then leads to a modification of the magnetic flux density (at any point in space), taken as [14,20]
\[ d\mathbf{B}_2(\mathbf{r}) = \frac{1}{3} \left( \frac{\mu_0 M_{\text{pol}} L(\omega_p V_p M_{\text{pol}} B(\mathbf{r}_2))}{B(\mathbf{r}_2)} \right) \frac{R_{\text{seed}}^3}{|\mathbf{r} - \mathbf{r}_2|^3} \times \frac{3((\mathbf{B}(\mathbf{r}_2) - \mathbf{B}(\mathbf{r}_2)) \cdot (\mathbf{r} - \mathbf{r}_2)^2}{|\mathbf{r} - \mathbf{r}_2|^4}(\mathbf{r} - \mathbf{r}_2) - \mathbf{B}(\mathbf{r}_2) \right) \]
where \( \mathbf{r} \) represents an arbitrary point in space and \( \mathbf{B}(\mathbf{r}_2) \) is flux density at \( \mathbf{r}_2 \) due to the external field and seed. The reader should note that it is the gradient of this field, \( d\mathbf{B}_2 \), at \( \mathbf{r}_1 \) (and equivalently at \( \mathbf{r}_2 \) for Particle 2) which is required for the calculation of Eq. (4).

3. Blood flow—the Navier–Stokes equations

The blood is treated as an incompressible, Newtonian, isothermal, single-phase fluid with steady state flow [9]. Thus, we have the continuity equation
\[ \nabla \cdot \mathbf{v}_b = 0 \]
and the Navier–Stokes equation
\[ \rho_b(\nabla \mathbf{v}_b + (\mathbf{v}_b \cdot \nabla) \mathbf{v}_b) = -\nabla P + \eta_b \nabla^2 \mathbf{v}_b \]
where \( \rho_b \) is the density of the blood and \( P \) is the pressure. To solve Eqs. (14) and (15), a uniform inlet velocity profile is assumed at the inlet control volume (CV) such that
\[ \mathbf{v}_b = \begin{pmatrix} u_0 \\ 0 \end{pmatrix} \]
where \( u_0 \) is the inlet blood velocity. Non-slip boundary conditions are applied at the seed-blood interface. In addition, symmetry boundary conditions are applied at the upper and lower CV boundaries to maintain the constant flow profile. Atmospheric pressure is assumed at the outlet of the CV to satisfy the boundary condition on pressure.

4. Velocity equations, streamlines and capture cross section

The velocity of a particle can be obtained by summing the Stokes drag, the force due to hydrodynamic interaction and the modified magnetic force, as given in Eqs. (1), (2) and (4), respectively, while ignoring inertia. Hence, for Particle 1 we obtain

\[
\dot{y}_{p1} = v_b + \frac{\vec{F}_{\text{int}} + \vec{F}_{\text{hyd}}}{6\eta_0 R_p} \tag{17}
\]

Finally, the trajectories of each particle can be obtained from evaluating the streamline functions [6,13]. The system performance of this model is calculated in terms of capture cross section, \( \lambda_c \), defined as

\[
\lambda_c = \frac{y_c}{R_{\text{seed}}} \tag{18}
\]

where \( y_c \) is the capture radius of the ferromagnetic seed. The capture radius, \( y_c \), is defined by the location of the streamline at the entrance to the CV of the last magnetic drug carrier particle captured to the seed (see Fig. 1). All calculations were performed using the open-source software finite volume library OpenFOAM [21].

5. Results and discussions

Clearly the strength of forces due to dipole–dipole and hydrodynamic interactions depends on many factors including: the magnitude of the applied external magnetic field, the initial distance between the particles, relative position of the particles to each other, the size of the ferromagnetic drug carrier particles and of the ferromagnetic seed. Moreover, the strength of the forces due to hydrodynamic interaction depends on the velocities of particles relative to the blood velocity [14]. In this paper we focus on varying the initial distance between the particles and present the results in terms of agglomeration and the altered capture cross section of the system.

In these simulations stainless steel (SS 409) is taken as the seed material with a seed radius of 1 \( \mu \)m. Results are presented by generating streamlines for two identical iron nanoparticles with radius \( R_p = 50 \) nm, containing 40 wt% iron (\( \chi_p = 0.4 \)), under the influence of homogeneous magnetic field oriented parallel to the flow (\( \theta = 0 \)) with magnitude \( \mu_0 H_0 = 0.7 \) T. The relevant blood flow properties and the properties of the ferromagnetic material, used in the magnetic drug carrier particles and for the seeds, are given in Table 1.

In order to describe the effect of the interactions we consider two different simulation configurations. The first configuration is intended to illustrate the dependence of the agglomeration point on the interparticle distance for particles that originate within the reference capture cross-section area. Agglomeration is taken to occur where the (surface-to-surface) interparticle distance reaches zero. The second simulation configuration is intended to examine the effects of interactions on the trajectories of particles near the boundary of the reference capture cross section and the resulting changes in the capture cross section. The boundary of the reference capture cross section, \( \lambda_c^* \), is the trajectory of the last particle, which would be captured by the seed in the non-interacting case. In all simulations, the behaviour of the particles after agglomeration is not considered. Throughout, the particles are taken to have the same initial x-coordinate with an initial interparticle distance, \( D \). Throughout, interparticle distance is defined as the distance between the surfaces of the particles. These initial conditions serve to illustrate the effect of the interparticle distance on behaviour. The coordinates and nanoparticle dimensions used are scaled in terms of \( R_{\text{seed}} \) and hence the scaled particle radius is 0.05.

5.1. Effect of interactions on the agglomeration of particles

Of interest is the relationship between initial interparticle distance, \( D \), and the resulting position of the agglomeration point as measured from the surface of the seed. This relationship is shown in Fig. 2 with (a) dipole–dipole magnetic interaction only, (b) hydrodynamic interaction only, (c) both interactions and (d) no interactions. As expected, in all cases, the distance between the agglomeration point and centre of the seed is seen to decrease

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>SI Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho_0 )</td>
<td>1040</td>
<td>kg/m³</td>
</tr>
<tr>
<td>( \eta_0 )</td>
<td>0.002</td>
<td>kg/ms</td>
</tr>
<tr>
<td>( U_0 )</td>
<td>0.001</td>
<td>m/s</td>
</tr>
<tr>
<td>( \mu_0 H_0 )</td>
<td>0.7</td>
<td>(T)</td>
</tr>
<tr>
<td>( \chi_p )</td>
<td>0.4</td>
<td>–</td>
</tr>
<tr>
<td>( \rho_p )</td>
<td>7850</td>
<td>kg/m³</td>
</tr>
<tr>
<td>( f_{\text{seed}} )</td>
<td>1000</td>
<td>A/m</td>
</tr>
<tr>
<td>( M_{\text{seed},A} )</td>
<td>1397000</td>
<td>A/m</td>
</tr>
<tr>
<td>( M_{\text{seed,S}} )</td>
<td>1735000</td>
<td>A/m</td>
</tr>
<tr>
<td>( R_s )</td>
<td>1.0 ( \times ) 10⁻⁶</td>
<td>m</td>
</tr>
<tr>
<td>( R_p )</td>
<td>5.0 ( \times ) 10⁻⁹</td>
<td>m</td>
</tr>
<tr>
<td>( \rho_{\text{pol,p}} )</td>
<td>950</td>
<td>kg/m³</td>
</tr>
<tr>
<td>( T )</td>
<td>300</td>
<td>K</td>
</tr>
<tr>
<td>( k )</td>
<td>1.38 ( \times ) 10⁻²³</td>
<td>J/K</td>
</tr>
</tbody>
</table>

![Fig. 2. Distance of agglomeration point from the seed plotted against initial distance between the particles, D, with (a) dipole–dipole magnetic interaction only, (b) hydrodynamic interaction only, (c) both interactions and (d) no interactions between the particles. All other conditions are the reference case as in Table 1. Inset represents the trajectories as the particles move towards the seed.](image-url)
as $D$ increases. In these simulations, two particles, labelled Particle 1 and Particle 2, are placed at scaled positions $(-20, D/2+R_{p}/R_{seed})$ and $(-20, -D/2-R_{p}/R_{seed})$ for a range of values of $D$. The initial $x$-coordinate value corresponds to left border of the CV. The inset indicates the initial position of the particles and their trajectories for cases (a) to (d) for a typical value of $D (D = 0.40)$ as they approach the seed.

On comparing the agglomeration point for the particles with only magnetic dipole–dipole interaction to that for the particles with no interaction, we find that the particles with magnetic interaction agglomeration earlier for all initial particle distances up to $D=1$ (see Fig. 2). Also in Fig. 2, with the inclusion of hydrodynamic interaction only, the two particles are seen to repel each other due to their velocities relative to the blood, inhibiting agglomeration. It is worth noting that the relative velocities are solely due to the magnetic velocities resulting from the presence of the seed gradient. In the inset it is seen that in the case with (only) hydrodynamic interaction the particles agglomerate after the agglomeration point expected without any interactions.

With the study of the combined effect of magnetic dipole–dipole and hydrodynamic interactions, we observe that, as expected, at short range the magnetic effects dominate, and at longer range the hydrodynamic are dominant. This is consistent with the forces being dependent on $|\mathbf{r}_1-\mathbf{r}_2|^{-3}$ and $|\mathbf{r}_1-\mathbf{r}_2|^{-1}$, respectively. From Fig. 2 a critical value of $D$ can be observed at the intersection of the curves with both interactions and no interactions at $D=0.56$. Below this critical value of $D$, the two particles are seen to agglomerate before the agglomeration point expected without interactions. For initial distances larger than this critical value of $D$, (repulsive) hydrodynamic forces dominate and the particles agglomerate after the agglomeration point expected without interactions (i.e. closer to the seed).

### 5.2. Effect of interactions on the capture cross section of the system

In Figs. 3–7 trajectories are presented and the effect of the inclusion of interactions on the capture cross section of the system is studied. The trajectories of two particles are calculated again with (a) dipole–dipole magnetic interaction only, (b) hydrodynamic interaction only, and (c) both interactions. In all three cases the trajectories of particles without any interactions and the resulting boundary of the reference capture...
cross section, \( \lambda_{c}^{*} \), are used as the reference case. Thus, two different trajectories are generated for each particle in each figure.

Particles are placed equidistant and symmetric about the \( \lambda_{c} \), corresponding to the initial position \((-20, \lambda_{c}^{*}=0.32)\) where \( \lambda_{c}^{*} \) is 4.47. This critical value, \( \lambda_{c}^{*} \), depends on the model parameters used (see Table 1). In each of three cases (a) to (c), a maximum value of \( D \), whereby two particles are still captured by the seed is determined. For each interaction case, the separate effect on \( \lambda_{c} \) for this maximum value of \( D \) is calculated and compared.

### 5.2.1. Magnetic dipole–dipole interaction

In the case with (only) magnetic dipole–dipole interaction, we find that the maximum value of \( D \) for which both particles are now captured is 0.40. In Fig. 3, the trajectories for this case are presented. Thus, for this specific initial arrangement, the calculated capture radius can be said to increase by 0.25 \( R_{seed} \) corresponding to a \(-6\%\) (absolute) increase in \( \lambda_{c} \).

To explore further the effect of magnetic dipole–dipole interaction on the capture cross section of the system, the initial position of both particles is translated vertically whilst maintaining a fixed interparticle distance of \( D = 0.41 \). In the first case, by moving the particles downwards, as expected both particles continue to be captured by the seed, and thus the capture cross section of system is unchanged. In the second case, by moving both particles upwards, the initial position at which the lower particle (Particle 1) is no longer captured by the seed as expected. By further moving the two particles upwards, the initial position at which the lower particle (Particle 2) ceases to be captured by the seed is now higher than for the non-interacting case. Thus, Particle 1 has caused the “capture” of Particle 2 by pushing it towards the seed. Fig. 6 illustrates this positive effect on the capture radius of the system where it is increased by 0.138 \( R_{seed} \). Which corresponds to a \(-3\%\) (absolute) increase in \( \lambda_{c} \).

5.2.3. Magnetic dipole–dipole and hydrodynamic interactions

With the inclusion of both interactions, we find that the maximum value of \( D \) for which both particles are now captured is 0.54. In Fig. 7, the trajectories for this case are presented. For this initial arrangement, the calculated capture radius can be said to increase by 0.32 \( R_{seed} \) corresponding to a \(-7\%\) (absolute) increase in \( \lambda_{c} \). In this case, the magnetic dipole–dipole and hydrodynamic interactions both have a positive effect on the capture cross section of the system.

To study the combined effect of both interactions, we include the hydrodynamic interaction to the case with (only) magnetic dipole–dipole interaction. Thus, the simulations are repeated with a fixed interparticle distance of \( D = 0.40 \) and the capture cross section of the system is calculated. In the first case, by moving the particles downwards, both particles continue to be captured by the seed, and thus the capture cross section of system is unchanged. In the second case by moving both particles upwards the following is observed. Again, the upper particle (Particle 1) is no longer captured by the seed. By further moving the two particles upwards we next observe that the initial position at which the lower particle (Particle 2) ceases to be captured by the seed is the same as the non-interacting case. When the value of \( D \) is 0.40, we find that inclusion of both interactions does not affect the capture cross section of the system as the effects of magnetic dipole–dipole interaction and hydrodynamic interaction on capture cross section balance each other. Thus, inclusion of hydrodynamic interaction has caused the increase of the capture radius by 0.16 \( R_{seed} \) relative to the case with magnetic dipole–dipole interaction only. Similarly, inclusion of magnetic dipole–dipole interaction has caused the decrease of the capture radius by 0.138 \( R_{seed} \) relative to the case with hydrodynamic interaction only when the value of \( D \) is 0.40. These apparent imbalances we attribute to the inherent the nonlinearity and cross dependence of the two interactions. Furthermore, for this specific case, if we decrease the value of \( D \), the magnetic dipole–dipole interaction becomes dominant and if we increase the value of \( D \), the hydrodynamic interaction dominates again consistent with \(|F_{1} - F_{2}|^{-3}\) and \(|F_{1} - F_{2}|^{-1}\) dependence. Specifically, in our model with inclusion of both interactions, the effect of magnetic dipole–dipole interaction on the capture cross section of the
system is larger than the effect of the hydrodynamic interaction when the value of $D$ is less than 0.40.

6. Conclusions

The effect of the dipole–dipole and hydrodynamic interactions between two nanoparticles on the calculated magnetic force in the implant-assisted magnetic drug targeting system of Avilès et al. is considered. In these simulations, depending on the initial configuration of the nanoparticles, both increases and decreases can be observed in the capture cross section of the modified model. It is observed that, both dipole–dipole and hydrodynamic interactions should be considered to calculate the capture cross section of the IA-MDT system due to comparable size of both interactions. Inclusion of both interactions was seen to alter the capture cross section of the system by up to 7% in absolute terms. We note that the relative positions of the particles and the relative velocities of particles to blood flow are important factors during the calculation of the effect of hydrodynamic interaction on the capture radius of the system. Also, we note that if two particles can agglomerate and start moving together it might be expected that their altered hydrodynamic volume would reduce the effective Stokes drag allowing both particles to be more easily captured by the seed and thus leading to an additional capture cross section increase.

We have presented an interaction model applied to IA-MDT. This model should be capable of treating agglomeration of particles known to occur in such systems [7,10,11]. It should be possible to extend this approach to model dipole–dipole and hydrodynamic interactions for multiple particles in further implant arrangements. In particular we intend to model the stent arrangement proposed and studied in Avilès et al. [10,11], where multiple particle agglomeration can be expected to contribute significantly to increase in the capture of particles reported therein.

Acknowledgements

This work was funded by Enterprise Ireland under the Applied Research Enhancement (ARE) program as part of the South Eastern Applied Materials (SEAM) Research Centre. Dr Adriele Prina-Mello of CRANN, Trinity College Dublin is thanked for assistance at the 7th International Conference on the Scientific and Clinical Applications of Magnetic Carriers, May 2008 in Vancouver. AM is grateful to the conference organisers for free registration.

References

Many particle magnetic dipole-dipole and hydrodynamic interactions in magnetisable stent assisted magnetic drug targeting

P.J. Cregg\textsuperscript{a,}\textsuperscript{*}, Kieran Murphy\textsuperscript{a}, Adil Mardinoglu\textsuperscript{a} and Adriele Prina-Mello\textsuperscript{b,}\textsuperscript{*}

\textsuperscript{a}SEAM Centre, Materials Characterisation and Processing Group, Waterford Institute of Technology, Waterford, Ireland

\textsuperscript{b}CRANN and School of Physics, Trinity College, Dublin 2, Ireland

Abstract

The implant assisted magnetic targeted drug delivery system of Avilés, Ebner and Ritter is considered both experimentally (\textit{in vitro}) and theoretically. The results of a 2D mathematical model are compared with 3D experimental results for a magnetisable wire stent. In this experiment a ferromagnetic, coiled wire stent is implanted to aid collection of particles which consist of single domain magnetic nanoparticles (radius \(\approx 10\, \text{nm}\)). In order to model the agglomeration of particles known to occur in this system, the magnetic dipole-dipole and hydrodynamic interactions for multiple, \(N\), particles are included. Simulations based on this mathematical model were performed using the open source C++ finite volume library OpenFOAM. Different initial positions are considered and the system performance is assessed in terms of collection efficiency. The results of this model show closer agreement with the measured \textit{in vitro} experimental results and with presented literature. The implications in Nanotechnology and Nanomedicine are based on the prediction of the particle

\textit{Preprint submitted to Elsevier}  
\textit{10 November 2009}
efficiency, in conjunction with the magnetisable stent, for the targeted drug delivery.

Key words: magnetic drug targeting, high gradient magnetic separation (HGMS), magnetic nanoparticles, simulation, dipole-dipole interaction, hydrodynamic interaction, magnetisable stent.

PACS: 47.63.mh, 47.63.-b, 87.85.gf

1 Introduction

The development of more effective drug treatment methodologies is an area of much research. In most drug delivery systems much of any drug administered to patients does not reach its target site. The aim of the drug targeting is to decrease the amount of drug delivered to healthy tissue, while maintaining the therapeutic action at the desired site. One such approach is magnetic drug targeting (MDT). For instance magnetic particles can be employed as carriers in a cancer treatment, thereby avoiding the side effects of conventional chemotherapy [1, 2]. MDT typically uses an external magnetic field source to capture and retain magnetic drug carrier particles (MDCPs) at a specific site after being injected into the body. Studies have shown that MDT is a relatively safe and effective methodology for targeting drugs to a specific site in the body [3–5]. However, there are some significant limitations of MDT. One limitation associated with MDT is the gradient problem, that is the magnetic force requires a magnetic field gradient. Specifically it can be difficult using external magnets only to target areas deep within the body, without

* Corresponding authors.

Email addresses: pjcregg@wit.ie (P.J. Cregg), prinamea@tcd.ie (Adriele Prina-Mello).
targeting the surface more strongly [6]. To overcome this problem several authors [7–16] have proposed implanting ferromagnetic materials such as wires, seeds and stents within the body. Of the various IA-MTD implants suggested by Ebner, Ritter and co-workers [7–13, 16], we consider a magnetisable stent as the implant, with MDCPs containing magnetic single domain nanoparticles. Previously, by considering high gradient magnetic separation, Mikkelsen et al. [17] have included both the hydrodynamic and dipole-dipole interactions for the case of low magnetic fields. Also, Mehasni et al. have considered the effect of magnetic dipole-dipole interaction on the performance of high gradient magnetic separation systems [18]. Some of the present authors have previously considered the effect of the interactions for two MDCPs on the agglomeration of the MDCPs [19]. Here, we calculate the effect of interactions of many particles on the collection efficiency of the system leading to the agglomeration of particles. Avilés et al. [9] compared the (non interacting) particle model of this stent system with in vitro experimental arrangement using a ferromagnetic stent made in the shape of a coil. Their results indicated that at low fluid velocity more particles were collected than predicted. Furthermore, they suggested that particle agglomeration (due to interparticle interactions) might explain this. With this in mind, we have further developed their mathematical model to include both dipole-dipole and hydrodynamic interactions between many MDCPs. These theoretical results are presented here and are compared with the experimental results of Avilés et al. [9] and new in vitro experiments. Simulations are performed using OpenFOAM a finite volume simulation C++ library.
2 Experimental Setup

In this experiment ferromagnetic particles with diameter of 0.86 µm containing 45.8 wt% magnetite are used as the MDCPs (Polysciences Europe GmbH). Stainless steel (SS) 430 (California Fine Wire Co.) is taken as the wire material for the stent with a 62.5 µm radius following Avilés et al. [9]. The stent is prepared by looping a length of wire, $L$, into a 2 cm long coil having a 0.04 cm radius containing 10 loops, $N_l$, with 0.2 cm between each loop. Between use, each stent wire is cleaned by a 30 minute sonication in ethanol. A set of 15 identical coil stents are made and cleaned for the full MDT experimental testing.

The stent is firmly positioned within a borosilicate glass capillary tube by interference adhesion against the inner surface of the tube (radius of 0.04 cm). Controlled thickness capillary tubing is used to maximise the contrast between stent and glass curvature for real time video imaging and particle detection. Furthermore, this is also eliminates any turbulence caused by the irregular glass surface roughness. In this experiment we use a capillary glass tube (0.04 cm radius) and particle size proportionally similar to Avilés et al. [9].

The experimental setup is shown in Fig. 1. It consists of a capillary glass tube with a regularly spaced coil stent, an equally spaced pair of single NdFeB permanent magnets (in opposition), connected by tygon tubing to a 2.5 ml syringe where one end is connected to a high precision syringe pump to supply the suspension of MDCPs and the other end is connected to a collection system for collection efficiency measurements. The setup also comprises an
inverted microscope connected to a CCD camera for high resolution imaging (QI Micropublisher, USA) and video acquisition. Magnetic field strength is measured by a Hall probe gaussmeter (Lake shore, USA). The particle, pre- and post- wash buffer solution where precisely injected by using 2.5 ml syringes connected to a high precision syringe pump system and software where it is possible to control injection direction, volume injected, flow rate in relation to the fluid solution injected (Nemesys system, Cetoni Gmbh, Germany). For each solution injected the total concentration is measured, pre- and post-experiment, by flow cytometry technique (Accuri, C6 Flow Cytometer and CFlow plus software, UK). Thus, each experiment had the same initial volume of solution.

Microscopy imaging is carried out using an Olympus microscope (Olympus, Japan) connected to a QI micropublisher camera driven by ImagePro software (Media Cybernetics, UK). Real-time streaming is carried out using Debut
An homogeneous particle solution is prepared with the use of full cell culture media (RPMI, Gibco, UK) with the addition of 5% bovine serum albumin (BSA) to make up to a similar viscosity. The concentration of the MDCP solution used here is $4 \times 10^{10}$ per litre, a lower concentration than that used in the experiment of Avilés et al. [9]. There the concentration was 50 mg/litre which corresponds to $11.2 \times 10^{10}$ per litre. These concentrations are calculated from the mass of one MDCP. In both concentration the particles agglomerated and they create clusters. In this study, we use lower concentration of MDCP due to the higher magnetite load single MDCP containing 45.8 wt% magnetite whereas Avilés et al. [9] uses MDCP containing 25 wt% magnetite. To model the behaviour of the MDCPs, we use smaller number of the MDCPs for lower concentration to match the experimental setup of Avilés et al. [9].

Once the MDT system is set up, control runs are carried out, with and without magnetic field to calibrate the system and monitor the particle trajectory and agglomeration in the absence of the stent.

The coil stent is then inserted into the tube and two homogeneous magnetic field strengths $\mu_0 H_0 = 0.15$ T and $\mu_0 H_0 = 0.60$ T are applied for different fluid velocities ranging between 0.58 cm/s and 52.6 cm/s. Once the magnetic field is applied the MDCPs were seen to agglomerate and create clusters. Different flow rates where chosen similar to those Avilés et al. [9]. For $\mu_0 H_0 = 0.15$ T magnetic field strength 0.05, 0.1, 0.2, 0.4, 1.0 cm/s injection velocities and for $\mu_0 H_0 = 0.60$ T magnetic field strength 0.2, 0.4, 1.0, 2.0, 4.5 cm/s injection velocities were used.

The amount of the MDCPs collected by the stent is measured by the differ-
ential between the MDCP concentration in the collection tube and the known initial particle concentration. Both solutions are measured by flow cytometry in triplicate counts.

After each particle solution injection the magnetic gradient was removed to demagnetise the superparamagnetic particles and to account for the mechanically-bound particle residuals (always < 1% of the overall injected volume).

3 Outline of Model

In order to effectively model this system, the 3D geometry of the stent and tube is reduced to 2D slice through the centre of the tube (See Fig. 2). Thus the coiled stent is modelled as a series of circular cross sections of an infinite wire with radius of $R_{\text{wire}}$ located at the upper and lower boundaries of the walls. At each wall the wires are separated by a distance, $h$, between their centres, and the upper and lower sections are offset by $h/2$ as shown in Fig. 2. It should be noted that physically this corresponds to a 2D description of flow with a parabolic profile in a rectangular box with transverse cylindrical wires, all of infinite extent. We model the behaviour of $N$ ($N < 25$) MDCPs under the influence of Stokes drag, a force due to hydrodynamic interaction, and a magnetic force, modified to incorporate the mutual magnetic dipole-dipole interaction. Other forces such as inertia and gravity are ignored. The Stokes drag for MDCP $n$ is

$$\vec{F}_{sn} = 6\pi \eta f R_{pn} (\vec{v}_f - \vec{v}_{pn}), \quad (1)$$
Fig. 2. Schematic of the control volume (CV) used for determining the magnetisable stent collection efficiency (CE) through analysis of the corresponding MDCP trajectories. The CV has dimensions of 2 cm and 0.05 cm and encompasses a ten-loop stent within an expanded vessel. The MDCPs enter the CV from the left with a reduced average velocity defined by a parabolic profile and unexpanded average blood vessel velocity.

where $\eta_f$ is the viscosity of the fluid, $R_{\text{p}_n}$ is the radius of MDCP $n$, and $\vec{v}_f$ and $\vec{v}_{\text{p}_n}$ are the velocities of the fluid and MDCP $n$ respectively. The fluid velocity, $\vec{v}_f$, is determined by solving the appropriate Navier-Stokes equations. The motion of a MDCP through a viscous fluid creates a disturbance to the fluid flow, which will be felt by all other MDCPs. As a result, the other MDCPs experience a force which is said to result from hydrodynamic interaction with the original MDCP. By considering $N$ MDCPs, the force due to the hydrodynamic interaction, $\vec{F}_{\text{hyd}_n}$, which acts on MDCP $n$ due to the presence of other
\((N - 1)\) MDCPs, can be written as,

\[
\vec{F}_{\text{hyd}_n} = \sum_{\left(\begin{array}{c} i=1 \\ i \neq n \end{array}\right)}^{N} \xi_{ni} \cdot (\vec{v}_f - \vec{v}_p)
\]

where \(\xi_{ni}\) is the modification due to the hydrodynamic interaction given by

\[
\xi_{ni} = -6\pi \eta f R_{pn} \frac{3 R_{pn}}{4 |\vec{r}_n - \vec{r}_i|} \left( 1 + \frac{(\vec{r}_n - \vec{r}_i) \otimes (\vec{r}_n - \vec{r}_i)}{|\vec{r}_n - \vec{r}_i|^2} \right)
\]

where \(R_{pn}\) is the radius of the MDCP \(i\), \(1\) is the unit tensor, \(\otimes\) is the vector tensor product (outer product), \(\vec{r}_n\) and \(\vec{r}_i\) are the positions of MDCP \(n\) and MDCP \(i\), respectively. Initially all MDCPs are taken to have the same radius but after agglomeration, MDCPs of different radius are possible, as each agglomeration is viewed as a new MDCP of increased radius.

In general the magnetic force acting on a magnetic moment is determined by

\[
\vec{F}_m = (\vec{m} \cdot \nabla) \vec{B}_{\text{total}}
\]

where \(\vec{m}\) is the magnetic moment and \(\vec{B}_{\text{total}}\) is the total magnetic flux density. Magnetic dipoles exert a force on each other, which can be included in the magnetic force equation by considering (i) the modified magnetic flux density and (ii) the modification in the magnetic moment resulting from this modified flux density. With regard to the magnetic dipole-dipole interaction between \(N\) number of MDCPs, each MDCP is taken as spherical with radius \(R_{pn}\) and sufficiently small to have homogeneous magnetic flux throughout the MDCPs. Hence, in order to include the magnetic effect on MDCP \(n\) of the other \((N - 1)\) MDCPs, the modified magnetic force, \(\vec{F}_{mm_n}\), can be written as

\[
\vec{F}_{mm_n} = (\vec{m}_n \cdot \nabla) \vec{B}_{\text{total}_n}
\]
where \( \vec{m}_n \) is the total magnetic moment of MDCP \( n \), \( \vec{B}_{\text{total}n} \) is the total magnetic flux acting on MDCP \( n \). It can be taken as

\[
\vec{B}_{\text{total}n} = \vec{B} + \sum_{i=1}^{N} d\vec{B}_i
\]  

(6)

where \( \vec{B} \) is the magnetic flux density due to the external field, \( d\vec{B}_n \) is the modification of the resulting magnetic flux density due to MDCP \( n \) at \( \vec{r} \). The modification to the magnetic flux density is thus taken as

\[
d\vec{B}_n(\vec{r}) = \frac{1}{3} \left( \mu_0 M_{f \text{m,p,s}} \frac{L(\beta)}{B} \right) \frac{R_{\text{wire}}^3}{|\vec{r} - \vec{r}_n|^3} \left( \frac{3}{|\vec{r} - \vec{r}_n|^2} - \frac{3}{|\vec{r} - \vec{r}_n|^2} \right) (\vec{r} - \vec{r}_n) - \vec{B}(\vec{r}_n)
\]  

(7)

where \( \mu_0 \) is the magnetic permeability of free space, \( \vec{r} \) represents an arbitrary point in space, \( \vec{B}(\vec{r}_n) \) is the flux density at \( \vec{r}_n \) and \( M_{f \text{m,p,s}} \) is the saturation magnetisation of the ferromagnetic material in the MDCP. The value of \( \vec{B} \) required to calculate the magnetic force as given by Eqs. (5) and \( (16) \), is calculated from the scalar magnetic potential due to the stent wires, which satisfies the Laplace equation over two conjoined regions: inside and outside the stent wires. Thus for outside the stent wires regions we have magnetic flux given by [8–13, 16]

\[
\vec{B} = \mu_0 (\vec{H}_0 - \nabla \phi)
\]  

(8)

where \( \vec{H}_0 \) is the applied homogeneous magnetic field as in Fig. 2 and \( \phi \) represents the reduced magnetic scalar potential which in the region outside the stent wires is given by [19–21]

\[
\phi = H_0 R_{\text{wire}}^2 \alpha_{\text{wire}} \frac{x \cos \theta + y \sin \theta}{x^2 + y^2},
\]  

(9)
where \( R_{\text{wire}} \) is the radius of the stent wire implant, \( \alpha_{\text{wire}} \) is the demagnetising factor of the stent wire (given by Eq. (11)). The induced magnetisation of the wire, \( \vec{M}_{\text{wire}} \), is taken to be parallel to the external magnetic field, \( \vec{H}_{0} \), and can be calculated from

\[
\vec{M}_{\text{wire}} = 2\alpha_{\text{wire}}\vec{H}_{0},
\]

(10)

where \( \alpha_{\text{wire}} \) is the demagnetising factor for an infinitely long cylinder in a perpendicular field taken as

\[
\alpha_{\text{wire}} = \min \left( \frac{\chi_{\text{wire},0}}{2 + \chi_{\text{wire},0}}, \frac{M_{\text{wire},s}}{2H_{0}} \right),
\]

(11)

where \( \chi_{\text{wire},0} \) and \( M_{\text{wire},s} \) are the zero field susceptibility and saturation magnetisation of the ferromagnetic wire respectively and \( \vec{H}_{0} \) can be written

\[
\vec{H}_{0} = (H_{0})_{0}\cos \theta H_{0}\sin \theta,
\]

(12)

where \( H_{0} \) is the magnitude of the applied field and \( \theta \) is the direction of the applied magnetic field with respect to the \( x \)-axis, as in Fig. 2.

It is assumed that the ferromagnetic material in each MDCP consists of smaller single domain spherical nanoparticles. Thus, the average projection of \( \vec{m} \) the moment in the direction of \( \vec{B}_{\text{total}} \) can be calculated from the Langevin function [6, 15, 22–24]

\[
L(\beta) = \coth(\beta) - \frac{1}{\beta},
\]

(13)

with Langevin argument

\[
\beta = \frac{m_{\text{fm,p}}B_{\text{total}}}{kT},
\]

(14)
where $B_{\text{total}}$ is the magnitude of $\vec{B}_{\text{total}}$, $k$ is Boltzmann’s constant, $T$ is the absolute temperature and $m_{f_{m,p}}$ is the magnitude of the magnetic moment of the magnetite in the MDCPs. The magnetic moment of each magnetite nanoparticle within the MDCP, $\vec{m}_{f_{m,p}}$, can be written as

$$\vec{m}_{f_{m,p}} = V_{f_{m,p}} M_{f_{m,p,s}} \vec{B} \frac{\beta}{B}$$

(15)

where $V_{f_{m,p}}$ is the spherical volume of a single domain magnetite nanoparticle and $M_{f_{m,p,s}}$ is the (volume) saturation magnetisation of the magnetite inside the MDCPs. Note that $M_{f_{m,p,s}}$ and $m_{f_{m,p,s}}$ are fitting parameters in this model, obtained by Avilés et al. through characterisation of the magnetic fluid [9].

Thus, the magnetic moment of the MDCP, $\vec{m}$, can be written as

$$\vec{m} = \omega_{f_{m,p}} V_p M_{f_{m,p,s}} L(\beta) \frac{\vec{B}}{B}$$

(16)

where $V_p$ is the MDCP volume and $\omega_{f_{m,p}}$ is the volume fraction of ferromagnetic material in the MDCP, related to its weight fraction $x_{f_{m,p}}$ through [16]

$$\omega_{f_{m,p}} = \frac{x_{f_{m,p}}}{x_{f_{m,p}} + (1 - x_{f_{m,p}}) \rho_{f_{m,p}} / \rho_{p_{m,p}}},$$

(17)

where $\rho_{f_{m,p}}$ is the density of the ferromagnetic material in the MDCP and $\rho_{p_{m,p}}$ is the density of the polymer material in the MDCP. In this model the value of $\omega_{f_{m,p}}$ is measured through the experiment.
4 Fluid flow — the Navier-Stokes equations

The fluid is treated as an incompressible, Newtonian, isothermal, single-phase fluid with velocity \( \vec{v}_f \) and pressure \( P \) at steady state flow. We have the continuity equation

\[
\nabla \cdot \vec{v}_f = 0,
\]

and the Navier-Stokes equation

\[
\rho_f [ (\vec{v}_f \cdot \nabla \vec{v}_f) ] = \nabla P + \eta_f \nabla^2 \vec{v}_f,
\]

where \( \rho_f \) is the density of the fluid. To solve Eqs. (18) and (19), a parabolic velocity profile is assumed at the inlet control volume (CV) such that

\[
v_{f,x}|_{x=0} = 1.5 u_0 \left( 1 - \left( \frac{y}{R_{\text{vessel}}} \right)^2 \right),
\]

\[
v_{f,y}|_{x=0} = 0
\]

where \( u_0 \) is the average inlet fluid velocity and \( R_{\text{vessel}} \) is the vessel (tube) radius. Furthermore, non-slip boundary conditions \( (\vec{v}_f = 0) \) are applied at the wire-fluid interface and at the upper and lower CV boundaries. Atmospheric pressure is assumed at the outlet of the CV to satisfy the boundary condition on pressure.

5 Velocity equations, Streamlines and Capture Cross Section

The velocity of a MDCP \( n \) can be obtained by summing the Stokes drag, the force due to hydrodynamic interaction and the modified magnetic force, as
given in Eqs. (1), (2) and (5) respectively with inertial forces, $\vec{F}_{in}$, as

$$\vec{F}_{sn} + \vec{F}_{hyd,n} + \vec{F}_{mm,n} = \vec{F}_{in}.$$  

(22)

For MDCP $n$, by ignoring the inertial forces, $\vec{F}_{in}$, we rewrite Eq. (22) as

$$6\pi \eta f R_{pn} (\vec{v}_f - \vec{v}_{pn}) + \sum_{i=1, i\neq n}^{N} \xi_{ni} \cdot (\vec{v}_f - \vec{v}_{pi}) + (\vec{m}_n \cdot \nabla) (\vec{B}_{total})_n = 0.$$  

(23)

Hence, we can obtain $\vec{v}_{pn}$ by solving Eq. (23) numerically in each time step.

Finally, the trajectories of each MDCP can be obtained from evaluating the streamline functions [6,13]. The system performance of this mathematical model is calculated in terms of collection efficiency, $CE$, defined as

$$CE = \frac{2R_{vessel} - y_1 + y_2}{2R_{vessel}} 100,$$  

(24)

where $y_1$ and $y_2$ are defined by the location of the streamline at the entrance to the CV of the last MDCPs captured by the stent wires (Fig. 2). All calculations were performed using the open-source software finite volume library OpenFOAM [25].

6 Results and Discussions

In this paper, we include the effect of both magnetic dipole-dipole and hydrodynamic interactions for multiple MDCPs in the stent based mathematical model of Avilés et al. [9]. We focus on varying the initial positions of $N$ ($N < 25$) MDCPs at the entrance of the CV and present the results in terms of the CE of the system considering the agglomeration of MDCPs.
Of interest is the effect of the velocity of the blood and the field strength on the CE of the system. This is shown in Figs. 3–6 with both dipole-dipole magnetic and hydrodynamic interactions, experimental results and without any particle interactions.

In the 2D model, the behaviour of the MDCPs after agglomeration is also considered. It is seen that the MDCPs create a cluster during their agglomeration as a result of both interactions. The volume of the cluster is calculated by summing the volume of the MDCPs agglomerated and the radius of the cluster is calculated using the general volume formulation \((4/3 \pi r^3)\) [26]. Whilst this assumption does not account fully for the resulting hydrodynamic volume, the effect of this assumption should not significantly affect our results.

6.1 Mathematical Model Explanation and Details

The rationale for the simulations is as follows. Given sufficient computing power, one might consider randomly distributing, particle in the form of a cluster, a very large number (> 10,000) of MDCPs and allow interactions between all of these. With limited computing resources, one is forced to reduce this. We do this in two ways. Firstly, by limiting the regions of initial positions that we consider and secondly by limiting the number of MDCPs that we allow to mutually interact. Thus we consider only those parts of the simulation which are likely to contribute to any alteration in the CE. For instance, in those parts of the capture cross section closest to the vessel walls, one can expect no improvement in the CE. In fact it is only where the initial positions are close to the border between the collection and no collection region, that is around the boundary of the reference capture cross section that we start
to see altered trajectories due to interactions. The boundary of the reference capture cross section (CCS), $\lambda^*$, is the trajectory of the last MDCP, which would be captured by the stent wires in the non-interacting case. Secondly, the mutual interparticle interaction would not be expected to have infinite extent. One can postulate a number $N^*$ of MDCPs in the model where the predicted difference in performance between modelling $N^*$ and $N^*+1$ becomes arbitrarily small. We point out that the computational effort required to model interactions scales with $N^2$, where $N$ is the number of MDCPs interacting. Simulations were performed for increasing $N$, and the results indicate that there is no significant change to the system performance metrics beyond twenty five MDCPs.

In light of these factors, we consider a particular, homogeneously distributed cluster of $N$ MDCPs. The MDCP concentration of the Avilés et al. system is 50 mg/l which corresponds to $11.2 \times 10^{10}$ MDCPs per litre and the MDCP concentration of our experimental setup is $4 \times 10^{10}$ per litre. The effective initial distance between the MDCPs in the CV is calculated using the concentration of the MDCPs in the glass tube. Initial distance is taken as the cube root of the MDCPs amount per litre ($(dm)^3$) and we created a homogeneous rectangular cluster of particles which mimic the experimental particle concentration flowing through the stent during the video streaming.

In order to describe the effect of both interactions we consider two different simulation configurations, similar to those used in a previous paper for the inclusion of interactions between the two MDCPs and between the MDCPs and the fluid [19]. The first configuration is intended to illustrate the agglomeration of the MDCPs within the reference CCS region. In this configuration all of the MDCPs are captured, as expected and the resulting CE of the system
for this situation is unaltered.

The second simulation configuration is intended to examine the effects of interactions on the CE of the system near the $\lambda_c^*$. For this, we place the centre of the particle cluster on the $\lambda_c^*$ for a given velocity and record changes in CE through following the MDCP trajectories in the normal way. We then shift the particle cluster up and down, and again record changes in CE. This approach is repeated for each increased fluid velocity, using, for a given field, the same particle cluster.

6.2 Comparison of the Mathematical Model Results and Literature

Initially, the results of our mathematical model and the experimental result of Avilés et al. are compared. Results are presented by calculating the CEs for identical MDCPs with initial radius $R_p = 0.435\, \mu m$ containing 25 wt% magnetite, under the influence of homogeneous magnetic field oriented perpendicularly to the flow ($\theta = \pi/2$) with magnitudes of $0.17\, T$ to $0.65\, T$. The glass tube radius size is taken as $0.05\, cm$ as in the experiment of Avilés et al.. In the model the magnetisation of the individual MDCPs is taken as the average value given by the Langevin function due to the single domain magnetic nanoparticles within. The relevant fluid flow properties and the properties of the ferromagnetic materials used in the MDCPs and for the stent wire, are given in Table 1.

For the configurations outlined above, we keep the applied field constant ($\mu_0 H_0 = 0.17\, T$) and we increase the blood velocity from $u_0 = 2.1\, cm/s$ to $u_0 = 42.4\, cm/s$. The resulting CEs for these simulations are shown in Fig. 3.
Secondly, using the same methodology we applied $\mu_0H_0 = 0.65$ T and vary the fluid velocity between $u_0 = 2.1$ cm/s and $u_0 = 42.4$ cm/s. The resulting CEs are given in Fig. 4.

Fig. 3. The collection efficiency (CE) of the system plotted as a function of the blood velocity ($2.1, 4.2, 10.6, 21.2, 42.4$ cm/s) at the applied field $\mu_0H_0 = 0.17$ T.

In Figs. 3 and 4, the results of the mathematical model with the interactions show closer agreement with experimental results of Avilés et al. with low fluid velocity. This is due to the interaction and agglomeration of MDCPs in our model. With low fluid velocity ($\leq 10$ cm/s) and higher applied field ($\mu_0H_0 = 0.65$ T) MDCPs create a larger volume of cluster more easily than with the lower applied field ($\mu_0H_0 = 0.17$ T). When we increase the fluid velocity the likelihood of the agglomeration of the MDCPs starts to decrease. For higher fluid velocity the CE of the IA-MDT system predicts lower collection than the results of Avilés model without interactions. This is due to the effect of
### Properties

<table>
<thead>
<tr>
<th>Properties</th>
<th>Symbol</th>
<th>Units</th>
<th>Values</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MDCPs Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polymer material</td>
<td>-</td>
<td>-</td>
<td>P(S/V-COOH)Mag</td>
<td>Physical</td>
</tr>
<tr>
<td>Radius</td>
<td>( R_p )</td>
<td>( \mu m )</td>
<td>0.435, 0.43</td>
<td>Physical</td>
</tr>
<tr>
<td>Saturation magnetisation</td>
<td>( M_{p,s} )</td>
<td>kA/m</td>
<td>22.4</td>
<td>Measured</td>
</tr>
<tr>
<td><strong>MDCPs Magnetic Material Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>-</td>
<td>-</td>
<td>Magnetite</td>
<td>Physical</td>
</tr>
<tr>
<td>Weight content</td>
<td>( x_{fm,p} )</td>
<td>wt%</td>
<td>25, 45.8</td>
<td>Physical</td>
</tr>
<tr>
<td>Volume content</td>
<td>( \omega_{fm,p} )</td>
<td>-</td>
<td>6.4</td>
<td>Physical</td>
</tr>
<tr>
<td>Saturation magnetisation</td>
<td>( M_{fm,p,s} )</td>
<td>kA/m</td>
<td>351.9</td>
<td>Measured</td>
</tr>
<tr>
<td>Magnetic moment</td>
<td>( m_{fm,p} )</td>
<td>Am²</td>
<td>2.03 × 10⁻¹⁹</td>
<td>Measured</td>
</tr>
<tr>
<td>Radius</td>
<td>( R_{fm,p} )</td>
<td>nm</td>
<td>5.18</td>
<td>Calculated</td>
</tr>
<tr>
<td><strong>Physical Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Particles</td>
<td>-</td>
<td>particle/L</td>
<td>11.2 × 10¹⁰, 4 × 10¹⁰</td>
<td>Physical</td>
</tr>
<tr>
<td>Temperature</td>
<td>( T )</td>
<td>K</td>
<td>300</td>
<td>Physical</td>
</tr>
<tr>
<td>Boltzmann’s constant</td>
<td>( k_B )</td>
<td>J/K</td>
<td>1.38 × 10⁻²³</td>
<td>Physical</td>
</tr>
<tr>
<td>Permeability of vacuum</td>
<td>( \mu_0 )</td>
<td>Tm/A</td>
<td>4π × 10⁻⁷</td>
<td>Physical</td>
</tr>
<tr>
<td><strong>Applied Field Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magnitude</td>
<td>( \mu_0 H_0 )</td>
<td>T</td>
<td>0.0–0.7</td>
<td>Physical</td>
</tr>
<tr>
<td>Angle of field direction</td>
<td>( \theta )</td>
<td>-</td>
<td>( \pi/2 )</td>
<td>Physical</td>
</tr>
<tr>
<td><strong>Stent Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td>-</td>
<td>-</td>
<td>SS 430</td>
<td>Physical</td>
</tr>
<tr>
<td>Wire radius</td>
<td>( R_{wire} )</td>
<td>( \mu m )</td>
<td>62.5</td>
<td>Physical</td>
</tr>
<tr>
<td>Loop separation</td>
<td>( h )</td>
<td>cm</td>
<td>0.2</td>
<td>Physical</td>
</tr>
<tr>
<td>Number of loops</td>
<td>( N_L )</td>
<td>-</td>
<td>10</td>
<td>Physical</td>
</tr>
<tr>
<td>Coil length</td>
<td>( L )</td>
<td>cm</td>
<td>2</td>
<td>Physical</td>
</tr>
<tr>
<td>Saturation magnetisation</td>
<td>( M_{implant,s} )</td>
<td>kA/m</td>
<td>1261</td>
<td>Measured</td>
</tr>
<tr>
<td>Magnetic susceptibility</td>
<td>( \chi_{implant,0} )</td>
<td>-</td>
<td>1000</td>
<td>Physical</td>
</tr>
<tr>
<td><strong>Blood &amp; Vessel Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velocity</td>
<td>( u_0 )</td>
<td>cm/s</td>
<td>2.1, 4.2, 10.6, 21.2, 42.4</td>
<td>Physical</td>
</tr>
<tr>
<td>Volume</td>
<td>( V_{blood} )</td>
<td>ml</td>
<td>0.58, 1.17, 2.34, 4.68, 11.7, 23.4, 52.6</td>
<td>Physical</td>
</tr>
<tr>
<td>Density</td>
<td>( \rho_b )</td>
<td>kg/m³</td>
<td>1000</td>
<td>Physical</td>
</tr>
<tr>
<td>Viscosity</td>
<td>( \eta_b )</td>
<td>kg/ms</td>
<td>1.0 × 10⁻³</td>
<td>Physical</td>
</tr>
<tr>
<td>Vessel radius</td>
<td>( R_{vessel} )</td>
<td>cm</td>
<td>0.05, 0.04</td>
<td>Physical</td>
</tr>
</tbody>
</table>

**Table 1**

Experimental values of fluidic setup and material parameters used in the mathematical model of the stent based simulation. Bold values are used in our experiment. Some material parameters are in agreement with Avilés and coworker study [9].
6.3 Comparison of the Mathematical Model and Experimental Results

Next, we compare the results of the mathematical model and \textit{in vitro} experiments undertaken at CRANN TCD. Results are presented by calculating the CEs for identical MDCPs with initial radius $R_p = 0.43 \, \mu m$ containing 45.8 wt\% magnetite, under the influence of homogeneous magnetic field oriented perpendicularly to the flow ($\theta = \pi/2$) with magnitudes of $0.15 \, T$ and $0.60 \, T$. The glass tube radius size is 0.04 cm in our experiments. This was done to achieve a better image contrast between the particle layers aggregating on the stent during the experimental testing which is also increased by the smaller
capillary diameter when compared to Avilés et al. [9] model.

In the model the magnetisation of the individual MDCPs is taken as the average value given by the Langevin function due to the single domain magnetic nanoparticles within. The relevant fluid flow properties and the properties of the ferromagnetic material used in the MDCP and for the stent wire, are given in Table 1.

![Graph showing collection efficiency (CE) plotted as a function of blood velocity.](image)

Fig. 5. The collection efficiency (CE) of the system plotted as a function of the blood velocity (0.58, 1.17, 2.34, 4.68, 11.7 cm/s) at the applied field $\mu_0 H_0 = 0.15$ T.

For the configurations outlined above, we keep the applied field constant ($\mu_0 H_0 = 0.15$ T) and we increase the blood velocity up to $u_0 = 11.7$ cm/s. The resulting CEs for these simulations are shown in Fig. 5. Secondly, we apply $\mu_0 H_0 = 0.60$ T and increase the fluid velocity up to $u_0 = 52.6$ cm/s. The resulting CEs are given in Fig. 6. In Figs. 5 and 6, the results of the model...
Fig. 6. The collection efficiency (CE) of the system plotted as a function of the blood velocity (2.34, 4.68, 11.7, 23.4, 52.6 cm/s) at the applied field $\mu_0 H_0 = 0.60$ T.

with the interactions show closer agreement with the measured experimental results. The results shown also highlight how a 0.01 cm reduction in the capillary radius can affect the collection efficiency. This leads to speculation over a higher efficacy of the MDCT technique at the level of peripheral circulatory capillary vessels. On the other hand, this increased CE efficiency also increases the risk of vessels clotting and thrombolytic effect especially when also accounting for the presence of the solid part of the blood [27].

Collection Efficiency is a key parameter for the modelling validation of the experimental testing. Differences between Avilés et al. and our experimental model (Cregg et al.) are shown in Table 2.
### Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Avilés et al. model</th>
<th>Cregg et al. model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vessel radius (cm)</td>
<td>0.05</td>
<td>0.04</td>
</tr>
<tr>
<td>Velocity Range (cm/s)</td>
<td>2.1–42.4</td>
<td>0.58–52.6</td>
</tr>
<tr>
<td>Magnetic Field (T)</td>
<td>0.17, 0.65</td>
<td>0.15, 0.60</td>
</tr>
<tr>
<td>Number of Repeats</td>
<td>-</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 2

Differences between Avilés et al. and Cregg et al. experimental model.

### 7 Conclusions

We have presented an interaction model applied to IA-MTD. This model considered the agglomeration of particles known to occur in such systems [8, 9, 11]. We include the effects of both the dipole-dipole and hydrodynamic interactions for multiple particles in stent implant arrangements. The resulting collection efficiencies derived from the mathematical model are in closer agreement with our latest experimental results and those presented by Avilés et al. Furthermore, the mathematical model presented in this work represents a useful analytical tool for the prediction of the efficacy of targeted drug delivery by superparamagnetic particles. The implications in the Nanotechnology and Nanomedicine research area are based on the efficiency in delivering the drug coated particles within the magnetisable stent length.
8 Acknowledgements

This work was funded by Enterprise Ireland under the Applied Research Enhancement (ARE) program as part of the South Eastern Applied Materials (SEAM) Research Centre at Waterford Institute of Technology. AM would like to thank Maurice Curtin for his help during the experiment in CRANN. Dr Prina-Mello would also like to thank Science Foundation Ireland (SFI), and CRANN for the financial support and Cetoni Gmbh for their technical support.

References


