

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

Adil Mardinoglu

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

Submitted to the Higher Education and Training Awards Council,
November, 2009.

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

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	61
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

B.2.1.7	T file	102
B.2.1.8	U blood velocity file	103
B.2.2	constant	104
B.2.2.1	Polymesh File (blockMeshDict) file	104
B.2.2.2	physical properties file	106
B.2.2.3	transport properties file	107
B.2.2.4	turbulence properties file	107
B.2.3	system	108
B.2.3.1	controlDict file	108
B.2.3.2	fvSchemes file	108
B.2.3.3	fvSolution file	109
B.2.3.4	sampleDict file	110
B.2.3.5	twoRegionDict file	111
C	Mesh Generator for Stent Model	112
C.1	Mesh Generator	112
C.2	BlockMeshDict.py	116
C.3	Vector.py	118
D	OpenFOAM Code for Stent Model	120
D.1	Applications (Solver) for Stent Implant	120
D.1.1	createFields.H file	120
D.1.2	createFields-analytic.H file	121
D.1.3	readtwoRegionDict.H file	124
D.1.4	stent.C file	125
D.2	Run (Case) for Stent Implant	136
D.2.1	0 (initial conditions file)	136
D.2.1.1	epsilon file	136
D.2.1.2	k file	138
D.2.1.3	nuTilda file	139
D.2.1.4	p file	141
D.2.1.5	R file	142
D.2.1.6	T file	143
D.2.1.7	Blood velocity file, U	145
D.2.1.8	Magnetic Field, H	146
D.2.1.9	Uniform Field in the Model	148
D.2.1.10	Modification to the Magnetic Flux Density	149
D.2.1.11	Magnetisation in Avilés Model	150
D.2.1.12	Magnetic Moment in the Model	152
D.2.2	constant	153
D.2.2.1	Polymesh File (blockMeshDict) file	153
D.2.2.2	physical properties file	154
D.2.2.3	transport properties file	155
D.2.2.4	turbulence properties file	156
D.2.3	system	156
D.2.3.1	controlDict file	156
D.2.3.2	fvSchemes file	157

D.2.3.3	fvSolution file	158
D.2.3.4	sampleDict file	160
D.2.3.5	twoRegionDict file	161
E	Publications	162

List of Figures

1.1	Experimental and theoretical CE results of Avilés et al. (source: [8]). . . .	10
2.1	Schematic representation of the B - H curve of a ferromagnetic material [45].	15
2.2	Schematic representation of the M - H curve of a typical ferromagnetic material [2].	16
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].	19
3.1	Schematic diagram of the control volume representing a capillary bed. . . .	27
3.2	Schematic diagram of the control volume representing a vessel with stent. .	28
4.1	General problem of object (region 1 = implant) embedded in a space region (region 2 = space) of constant permeability [76].	38
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]).	40
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, \vec{H}_0	47
5.2	Capture cross section, λ_c , plotted as a function of the applied magnetic field strength calculated using the Langevin function and following Avilés et al.	48
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.	54
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.	55
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.	56
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.	57

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.	58
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.	59
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.	65
5.10	The collection efficiency (CE) of the system plotted as a function of the blood velocity at the applied field $\mu_0 H_0 =$ of 0.17 T.	66
B.1	Structure of the OpenFOAM program	81

List of Symbols

Roman Symbols

A_c	Area of the electric current loop	13
\vec{B}	Magnetic flux density or magnetic induction	11
\vec{B}_{total_n}	Total magnetic flux density acting on MDCP n	49
B_r	Remanent magnetic induction	16
C	Curie constant	17
D	Initial interparticle distance between MDCPs	52
\vec{E}	Electric field strength	12
\vec{F}_{hyd_n}	Force due to the hydrodynamic interaction which acts on MDCP n ...	50
\vec{F}_i	Inertial forces	29
\vec{F}_{i_n}	Inertial forces of MDCP n	50
\vec{F}_{int_n}	Modified magnetic force of MDCP n	50
\vec{F}_m	Magnetic forces	29
\vec{F}_s	Stokes drag	29
\vec{F}_{s_n}	Stokes drag of MDCP n	50
f_0	Frequency pre-factor	23
f_w	The magnetic force density	80
\vec{H}	Magnetic field strength	11
H_c	Coercive field	16
\vec{H}_0	Externally applied magnetic field	29
h	Distance between the centres of the stent wires.	28
i	Current in the circuit	13
K_a	anisotropy constant	23
k_B	Boltzmann's constant	18
L	length of coiled stent	61

\vec{M}	(Volume) Magnetisation	13
\vec{m}	Magnetic moment	13
\vec{m}_n	Total magnetic moment of MDCP n	49
$M_{\text{fm,p}}$	Magnetisation of the ferromagnetic material in the MDCP	77
$M_{\text{fm,p,s}}$	Saturation magnetisation of the ferromagnetic material in the MDCP	76
M_{implant}	Magnetisation of the implant	77
$M_{\text{implant,s}}$	Saturation magnetisation of the implant	77
$M_{\mathbf{r}}$	Remanent magnetisation	16
M_s	Spontaneous magnetisation	18
M_{sat}	Saturation magnetisation	19
$N_{\mathbf{Eu}}$	Euler numbers	32
N_l	Number of loops in stent	61
$N_{\mathbf{Re}}$	Reynolds numbers	32
\hat{n}	The normal vector of the interface	39
$n_{\mathbf{a}}$	The number of atoms per unit volume	18
P	Pressure	30
P_0	Pressure at the outlet of the CV	31
R_{implant}	Radius of the implant	33
$R_{\mathbf{p}}$	Radius of the MDCP	29
$R_{\mathbf{p}_i}$	Radius of the MDCP i	51
$R_{\mathbf{p}_n}$	Radius of the MDCP n	50
R_{vessel}	Radius of vessel	28
R_{wire}	Radius of the wire (implant)	28
\vec{r}	Represents an arbitrary point in space	49
\vec{r}_n	Position of the MDCP n	49
T	Temperature	17
T_B	Blocking temperature	23
T_c	Curie temperature	17

\hat{t}	The tangential vector of the interface	39
t_r	Relaxation time	23
u_0	Inlet (average) velocity	27
$x_{\text{fm,p}}$	Weight fraction	46
V	Volume	13
V_B	Blocking volume	23
V_p	Volume of the MDCP	46
V_{p_n}	Volume of MDCP n	49
\vec{v}_b	Blood velocity	27
\vec{v}_p	Velocity of the MDCP	29
\vec{v}_{p_n}	Velocity of MDCP n	50

Greek Symbols

$\alpha_{\text{fm,p}}$	Demagnetising factor for a ferromagnetic material	76
α_{implant}	Demagnetising factor for the implant	77
β	Langevin argument	49
χ	Magnetic susceptibility	14
$\chi_{\text{fm,p}}$	Volumetric magnetic susceptibility of the ferromagnetic material in the MDCP with field	77
$\chi_{\text{fm,p},0}$	Volumetric magnetic susceptibility of the ferromagnetic material in the MDCP with zero field	76
χ_{implant}	Volumetric magnetic susceptibility of the implant	77
$\chi_{\text{implant},0}$	Volumetric magnetic susceptibility of the implant with zero field	77
η	Viscosity of the fluid	36
η_b	Viscosity of the blood	29
η_{eff}	Kinematic viscosity of the fluid	37
λ_c	Capture radius	27
λ_c^*	The resulting boundary of the reference capture cross section	54

μ	Permeability of the medium	12
μ_{implant}	Relative permeability of implant	43
μ_0	Permeability of free space	12
μ_r	Relative permeability	14
μ_{space}	Relative permeability of space	77
$\omega_{\text{fm,p}}$	Volume fraction of ferromagnetic material	46
ϕ	Magnetic scalar potential (Reduced–reduced)	30
ψ	Magnetic scalar potential (Total–total)	77
ψ_s	Stream function	33
ρ	Density of the fluid	36
ρ_b	Density of the blood	30
$\rho_{\text{fm,p}}$	Density of the ferromagnetic material in the MDCPs	46
$\rho_{\text{pol,p}}$	Density of the polymer material in the MDCPs.	46
$\vec{\tau}$	Torque on a magnetic dipole	13
τ_{max}	Maximum torque on a magnetic dipole	13
φ	Magnetic field orientation	30

List of Acronyms

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

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 [71] or radionuclides [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 [46]. To begin with, we discuss the use of magnetic nanoparticles in three particular applications:

- magnetic drug targeting,
- magnetic resonance imaging (MRI) contrast enhancement [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 [49, 66]. In contrast enhancement, the metastatic lymph nodes absorb the nanoparticles more than inflamed nodes and this is detectable with MRI [59, 74]. Hyperthermic treatment involves heating, via radio-frequency fields, within the range 41° – 46° C, damaging cells but also significantly assisting radiotherapy of tumour cells [65].

Although, biomedical applications of magnetic nanoparticles have been proposed since the 1950s [34, 38], recent advances in nanotechnology have meant that many of the technical problems associated with production and biocompatibility [50, 51, 52] are now being solved. Significantly, in March 2005, the US Food and Drug Administration gave approval to Advanced Magnetics [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 [53] (CEO Dr Andreas Jordan, see reference [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 [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 [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 [11, 36]. To overcome this problem several authors [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 [5, 6, 7, 8, 9, 10, 22, 66, 68] firstly, we consider a seed as the implant with single domain magnetic nanoparticles as MDCPs [28, 29] and secondly, we consider the *stent* as the implant, with MDCPs containing magnetic single domain nanoparticles [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. [8] 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 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 *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.

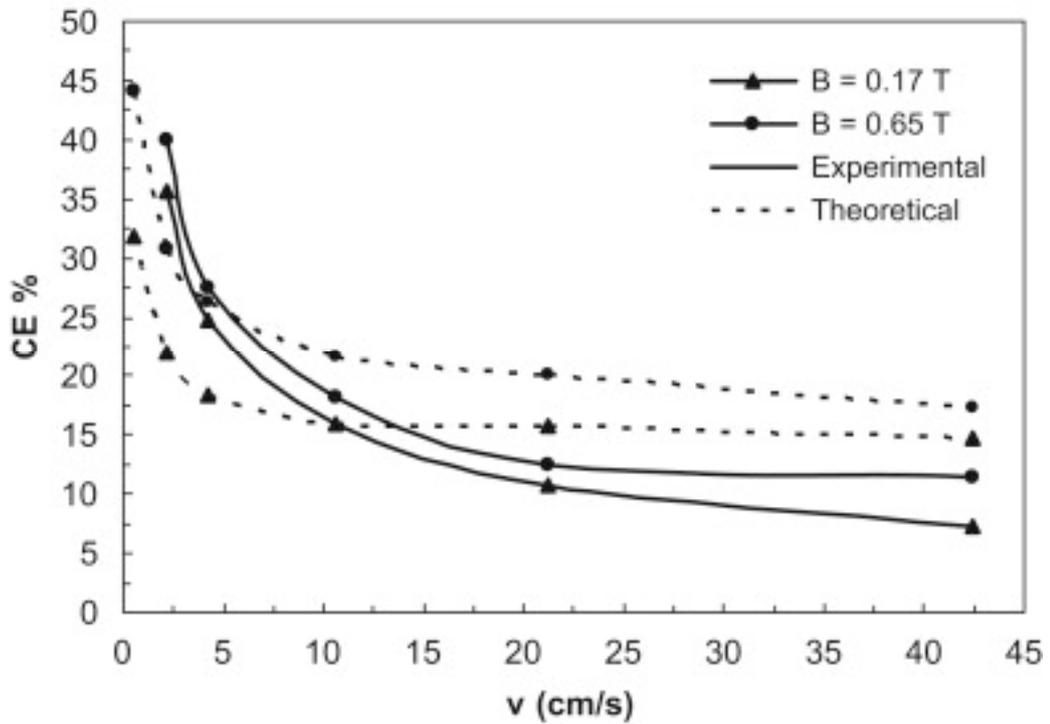
Recently, Mikkelsen et al. [21, 58] calculated the magnetic dipole-dipole and hydrodynamic interactions between magnetic beads under low magnetic fields (0.05 T) in microfluidic systems. Firstly, in their model, as a result of dipole-dipole interaction, magnetic beads (which behave as small implants) 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

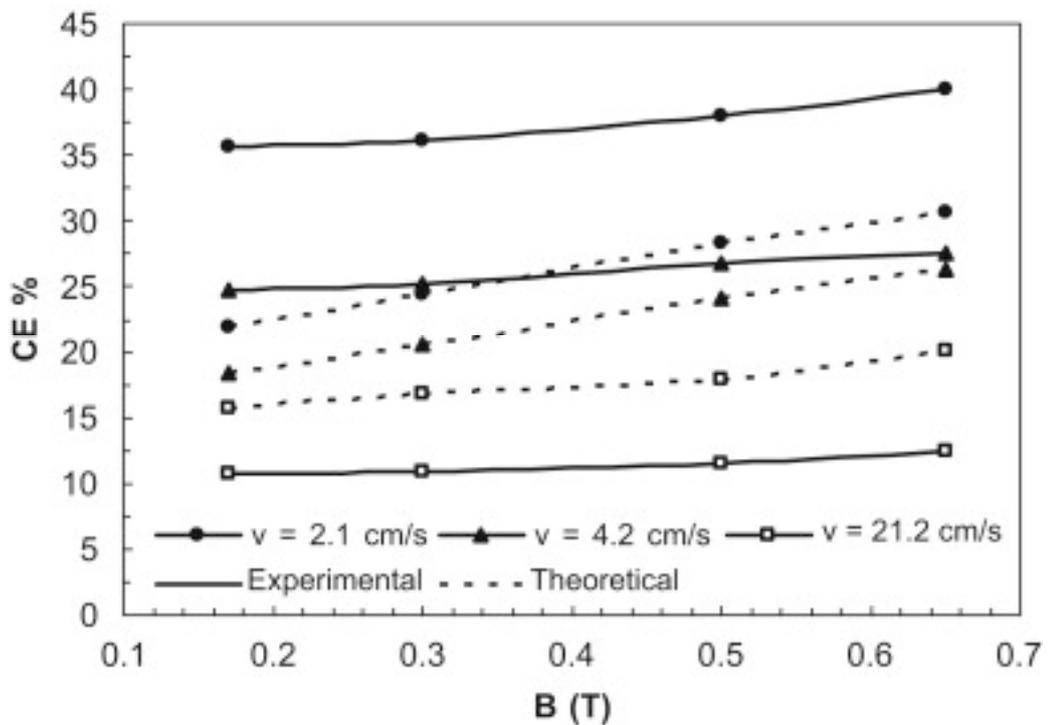
Specifically, in this work, we have developed the models of Ritter's group, in particular the stent model to incorporate dipole-dipole and hydrodynamic interactions between the MDCPs. This is with a view to account for the recent results of Avilés et al. [8]. These show higher CE than predicted with 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 Lübke 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.



(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.

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

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 current¹.

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/m²) 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 μ_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, μ , through,

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

where μ 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,

$$m = i A_c, \quad (2.3)$$

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 [45]

$$\vec{\tau} = \vec{m} \times \vec{B}, \quad (2.4)$$

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

$$m = \frac{\tau_{\max}}{B}, \quad (2.5)$$

and the unit of magnetic moment is ampere metres-squared (Am^2).

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

$$\vec{M} = \frac{\vec{m}}{V}. \quad (2.6)$$

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

$$\vec{B} = \mu_0 (\vec{H} + \vec{M}), \quad (2.7)$$

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 [45]

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

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

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

Magnetic susceptibility denoted by χ , 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} .

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.

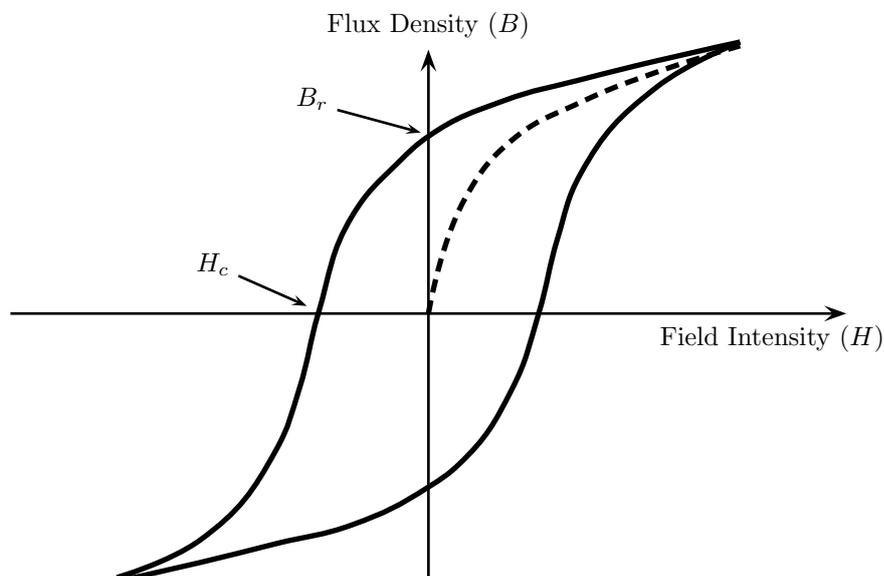


Figure 2.1: Schematic representation of the B - H curve of a ferromagnetic material [45].

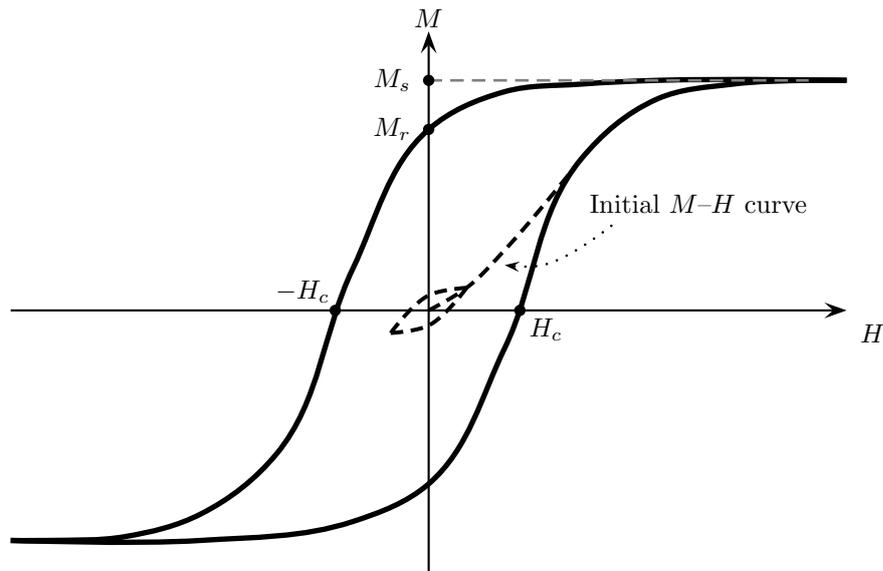


Figure 2.2: Schematic representation of the M - H curve of a typical ferromagnetic material [2].

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 [45] 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.

2.1.5 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} [45], 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],

$$\frac{M}{n_a m} = \coth\left(\frac{\mu_0 m H}{k_B T}\right) - \left(\frac{k_B T}{\mu_0 m H}\right), \quad (2.13)$$

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].

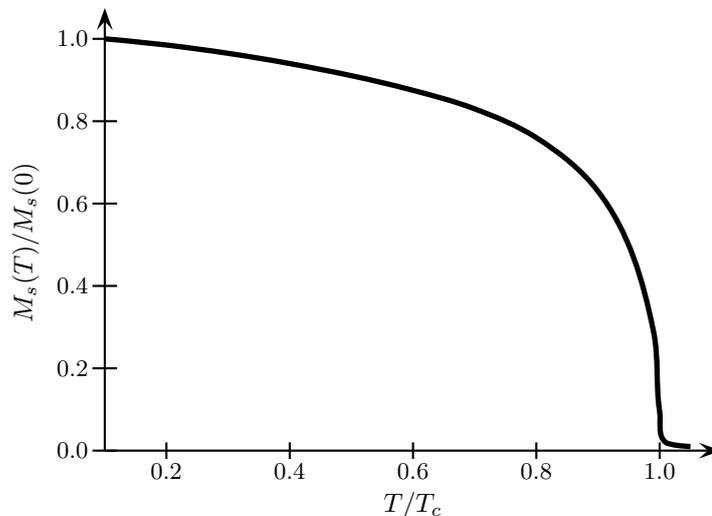


Figure 2.3: Spontaneous magnetisation, M_s , of a ferromagnet as a function of the temperature, T , normalised to the Curie Temperature, T_c . The applied field is assumed to be small, but finite, as it is in real measurements [2].

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_{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 ($< 20 \mu\text{m}$) 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 superparamagnetic (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,

$$\frac{1}{t_r} = f_0 \exp\left(\frac{-K_a V}{k_B T}\right) \quad (2.14)$$

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 $K_a V < 25k_B T$.

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 *in vivo* distribution of the nanoparticles. The contrast agents should have the same *in vivo* localisation as the beneficial nanoparticles because of their similar size and charge. Research is continuing in imaging the *in vivo* distribution of the contrast agents by MRI to optimise the size and surface properties for targeting tumours.

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.

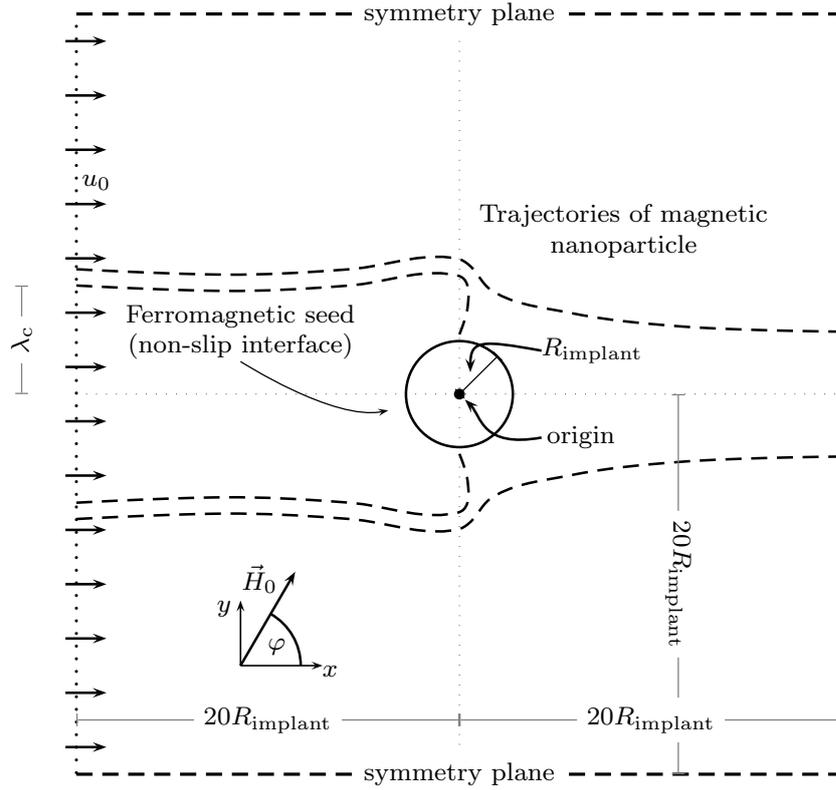


Figure 3.1: Schematic diagram of the control volume, CV, used in determining the capture radius, λ_c , of the magnetic nanoparticles.

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

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

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.

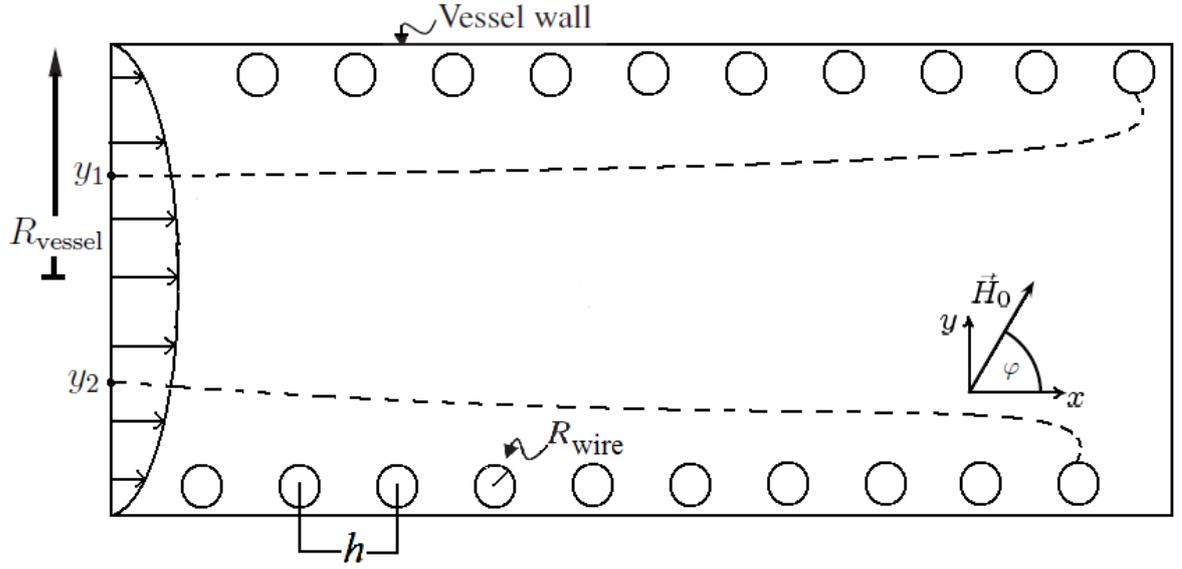


Figure 3.2: Schematic diagram of the control volume, CV, used for developing the 2D model of the stent-base IA-MDT system.

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_{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

$$\vec{v}_b = \begin{pmatrix} 1.5 u_0 \left(1 - \left(\frac{y}{R_{\text{vessel}}} \right)^2 \right) \\ 0 \end{pmatrix}, \quad (3.2)$$

where u_0 is the average inlet blood velocity and R_{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

$$\vec{F}_s + \vec{F}_m = \vec{F}_i \quad (3.3)$$

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

$$\vec{F}_s = 6\pi \eta_b R_p (\vec{v}_b - \vec{v}_p), \quad (3.4)$$

where η_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

$$\vec{F}_m = (\vec{m} \cdot \nabla) \vec{B}, \quad (3.5)$$

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

$$6\pi \eta_b R_p (\vec{v}_b - \vec{v}_p) + (\vec{m} \cdot \nabla) \vec{B} = 0. \quad (3.6)$$

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

$$\vec{v}_p = \vec{v}_b + \frac{1}{6\pi \eta_b R_p} (\vec{m} \cdot \nabla) \vec{B}. \quad (3.7)$$

where, \vec{B} is the total magnetic flux density acting on the MDCPs. The magnetic flux

density, \vec{B} is given by

$$\vec{B} = \mu_0 \vec{H}, \quad (3.8)$$

and in the space around the circular implant \vec{H} is given by,

$$\vec{H} = \vec{H}_0 - \nabla \phi. \quad (3.9)$$

where ϕ 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 φ , as in figures 3.1 & 3.2, the magnitude of the total magnetic field strength can be written in Cartesian coordinates as

$$H = \sqrt{\left(H_0 \cos \varphi - \frac{\partial \phi}{\partial x}\right)^2 + \left(H_0 \sin \varphi - \frac{\partial \phi}{\partial y}\right)^2}. \quad (3.10)$$

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

$$\nabla \cdot \vec{v}_b = 0, \quad (3.11)$$

and

$$\rho_b [(\vec{v}_b \cdot \nabla) \vec{v}_b] = -\nabla P + \eta_b \nabla^2 \vec{v}_b, \quad (3.12)$$

where ρ_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,

$$\tilde{x} = \frac{x}{R_{\text{wire}}}, \quad \tilde{y} = \frac{y}{R_{\text{wire}}}, \quad \tilde{v}_{b,x} = \frac{v_{b,x}}{u_0}, \quad \tilde{v}_{b,y} = \frac{v_{b,y}}{u_0}, \quad \tilde{P} = \frac{P}{P_0}. \quad (3.13)$$

where \vec{v}_b is the blood velocity, u_0 is the average inlet velocity, \tilde{v}_b is the scaled blood velocity, R_{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],

$$\frac{\partial \tilde{v}_{b,x}}{\partial \tilde{x}} + \frac{\partial \tilde{v}_{b,y}}{\partial \tilde{y}} = 0 \quad (3.14)$$

and

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

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

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

$$N_{\text{Eu}} = \frac{\rho_b u_0^2}{P_0}, \quad N_{\text{Re}} = \frac{2\rho_b u_0 R_{\text{wire}}}{\eta_b}. \quad (3.17)$$

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

$$\tilde{P} = 1. \quad (3.18)$$

Also, a non-slip boundary condition $\tilde{\vec{v}}_b = 0$ 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

$$\nabla \cdot \vec{v}_b = 0, \quad (3.19)$$

and

$$-\frac{2N_{\text{Eu}}}{N_{\text{Re}}} \nabla^2 \vec{v}_b + N_{\text{Eu}} (\vec{v}_b \cdot \nabla \vec{v}_b) + \nabla P = 0. \quad (3.20)$$

On rearranging the above equations we obtain

$$\nabla \cdot \vec{v}_b = 0, \quad (3.21)$$

and

$$N_{\text{Eu}} (\vec{v}_b \cdot \nabla \vec{v}_b) = -\nabla P + \frac{2N_{\text{Eu}}}{N_{\text{Re}}} \nabla^2 \vec{v}_b. \quad (3.22)$$

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

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

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

where ψ_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), λ_c , defined as

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

where R_{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

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

where R_{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 [63]

$$\frac{\partial \rho \vec{v}}{\partial t} + \nabla \cdot K \vec{v} - \nabla^2 \eta \vec{v} = -\nabla P \quad (4.1)$$

is represented by the code block

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

where ρ is the density, η is the viscosity, P is the pressure and \vec{v} is the velocity of fluid. (Elsewhere in OpenFOAM, ρ , η , 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 [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 [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

$$\frac{\partial \rho \vec{Q}}{\partial t} + \nabla \cdot (\rho \vec{v} \otimes \vec{Q}) - \nabla \cdot \rho D \nabla \vec{Q} = S_p \vec{Q} + S_q, \quad (4.2)$$

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

$$\nabla \cdot \vec{v} = 0, \quad (4.3)$$

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

$$\frac{\partial \vec{v}}{\partial t} + \nabla \cdot (\vec{v} \otimes \vec{v}) - \nabla \cdot 2\eta_{\text{eff}} \vec{D} = -\frac{1}{\rho} \nabla P, \quad (4.4)$$

where

$$\vec{D} = \frac{1}{2} (\nabla \vec{v} + \nabla \vec{v}^T), \quad (4.5)$$

where η_{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,

$$\nabla \cdot \vec{v} = 0, \quad (4.6)$$

$$-\eta_{\text{eff}} \nabla^2 \vec{v} + \nabla \cdot (\vec{v} \otimes \vec{v}) + \nabla P = 0. \quad (4.7)$$

where η_{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 η/ρ for η_{eff} . For calculating the blood velocity in **SimpleFOAM**, η_{eff} is specified as η_b/ρ_b

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

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

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

$$\vec{B} = \mu \vec{H}, \quad (4.9)$$

where μ is the permeability of the medium as

$$\mu = \mu_r \mu_0, \quad (4.10)$$

and \vec{H} is given by,

$$\vec{H} = \vec{H}_0 - \nabla \phi. \quad (4.11)$$

where ϕ is the magnetic scalar potential¹.

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

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

and

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

¹Technically, ϕ 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.

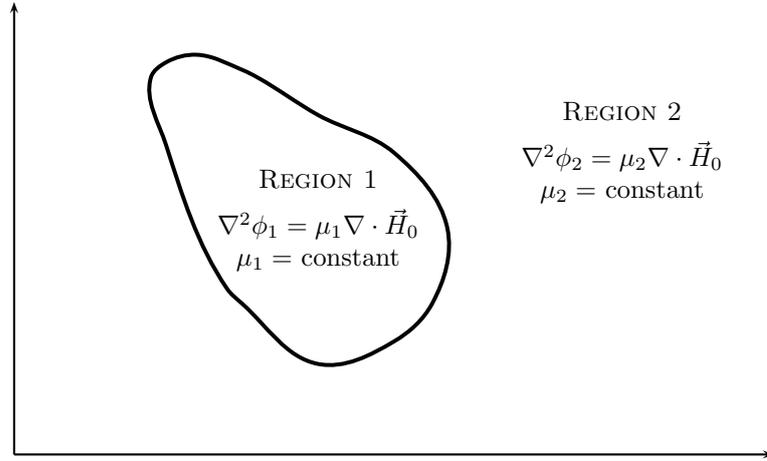


Figure 4.1: General problem of object (region 1 = implant) embedded in a space region (region 2 = space) of constant permeability [76].

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),

ϕ_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, θ) 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

$$\nabla = \frac{\partial}{\partial r} \hat{e}_r + \frac{1}{r} \frac{\partial}{\partial \theta} \hat{e}_\theta. \quad (4.14)$$

Hence

$$\nabla \cdot \vec{v} = \frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta}, \quad (4.15)$$

and

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}. \quad (4.16)$$

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

$$\vec{H}_1 = \vec{H}_0 - \nabla \phi_1, \quad \vec{B}_1 = \mu_1 \vec{H}_1, \quad \nabla^2 \phi_1 = \mu_1 \nabla \cdot \vec{H}_0. \quad (4.17)$$

and in region 2 (space)

$$\vec{H}_2 = \vec{H}_0 - \nabla \phi_2, \quad \vec{B}_2 = \mu_2 \vec{H}_2, \quad \nabla^2 \phi_2 = \mu_2 \nabla \cdot \vec{H}_0. \quad (4.18)$$

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)

$$\hat{n} \cdot \vec{B}_1 = \hat{n} \cdot \vec{B}_2 \quad \implies \quad \mu_1 \frac{\partial \phi_1}{\partial r} = \mu_2 \frac{\partial \phi_2}{\partial r} + (\mu_1 - \mu_2) \vec{H}_0 \cdot \hat{n}, \quad (4.19)$$

and

$$\hat{t} \cdot \vec{H}_1 = \hat{t} \cdot \vec{H}_2 \quad \implies \quad \frac{\partial \phi_1}{\partial \theta} = \frac{\partial \phi_2}{\partial \theta}, \quad (4.20)$$

and satisfy the boundary condition

$$\vec{H}_1 \text{ bounded}, \quad (4.21)$$

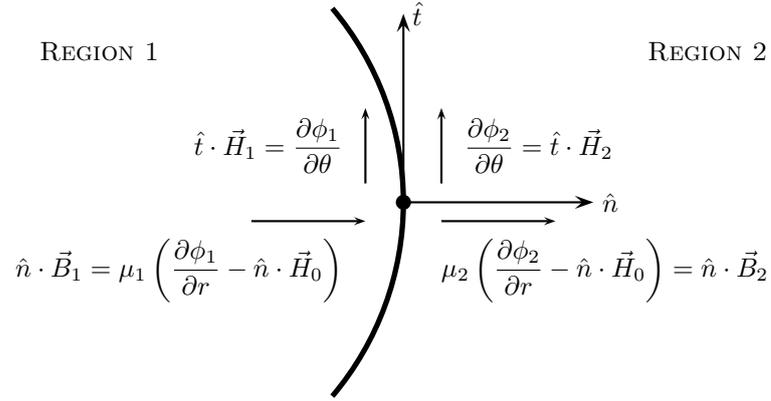


Figure 4.2: Interface conditions across the boundary of regions with different permeability.— the normal component of \vec{B} and the tangential component of \vec{H} are continuous [76]).

and

$$\vec{H}_2 \rightarrow \vec{H}_0 \text{ as } r \rightarrow \infty \implies \phi_2 \rightarrow 0 \text{ as } r \rightarrow \infty. \quad (4.22)$$

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

The Poisson equations for the scalar potentials both have solution

$$\phi_j = (a_j r + b_j r^{-1}) \cos \theta, \quad (4.23)$$

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

$$\phi_1 = (a_1 r + b_1 r^{-1}) \cos \theta, \quad (4.24)$$

and for region 2 (space)

$$\phi_2 = (a_2 r + b_2 r^{-1}) \cos \theta. \quad (4.25)$$

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

$$\lim_{r \rightarrow r_\infty} \phi_2 = 0 \implies \text{then } \lim_{r \rightarrow r_\infty} (a_2 r + b_2 r^{-1}) \cos \theta = 0, \quad (4.26)$$

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 $r = 1$,

$$-\mu_1(a_1 - b_1) \cos \theta + \mu_1 H_0 \cos \theta = -\mu_2(a_2 - b_2) \cos \theta + \mu_2 H_0 \cos \theta, \quad (4.27)$$

and on applying $a_2 = 0$ and $b_1 = 0$, this reduces to

$$-\mu_1 a_1 + \mu_1 H_0 = \mu_2 b_2 + \mu_2 H_0. \quad (4.28)$$

The interface condition (4.13) implies that

$$\frac{\partial \phi_1}{\partial \theta} = \frac{\partial \phi_2}{\partial \theta} \implies (a_1 + b_1) \cos \theta = (a_2 + b_2) \cos \theta, \quad (4.29)$$

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

$$-\mu_1 a_1 + \mu_1 H_0 = \mu_2 a_1 + \mu_2 H_0, \quad (4.30)$$

and so

$$a_1 = b_2 = \left(\frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0. \quad (4.31)$$

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

$$\phi_1 = a_1 r \cos \theta = \left(\frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0 r \cos \theta, \quad (4.32)$$

and for region 2 (space) is

$$\phi_2 = b_2 r^{-1} \cos \theta = \left(\frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0 r^{-1} \cos \theta. \quad (4.33)$$

We can rewrite these potentials in Cartesian coordinates as follows:

$$\phi_1 = \left(\frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0 x, \quad (4.34)$$

and

$$\phi_2 = \left(\frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) H_0 \frac{x}{x^2 + y^2}. \quad (4.35)$$

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

$$\phi = H_0 R_{\text{implant}}^2 \frac{\mu_{\text{implant}} - 1}{\mu_{\text{implant}} + 1} \frac{x \cos \varphi + y \sin \varphi}{x^2 + y^2}, \quad (4.36)$$

where R_{implant} is the radius of the implant (seed, stent wire), x and y are the coordinates measured from the centre of the implant and μ_{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 Adrielle 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.

5.1 *Inclusion of the Langevin Function*

5.1.1 *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 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]

$$L(\beta) = \coth(\beta) - \frac{1}{\beta}, \quad (5.1)$$

with Langevin argument

$$\beta = \frac{\omega_{\text{fm,p}} V_{\text{p}} M_{\text{fm,p,s}} B}{kT}, \quad (5.2)$$

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 [66]

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

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 (3.6)

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

The velocity of a MDCP, \vec{v}_p , can be written as in (3.7)

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

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 ≈ 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 $R_{\text{seed}} = 1$ μm .

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.

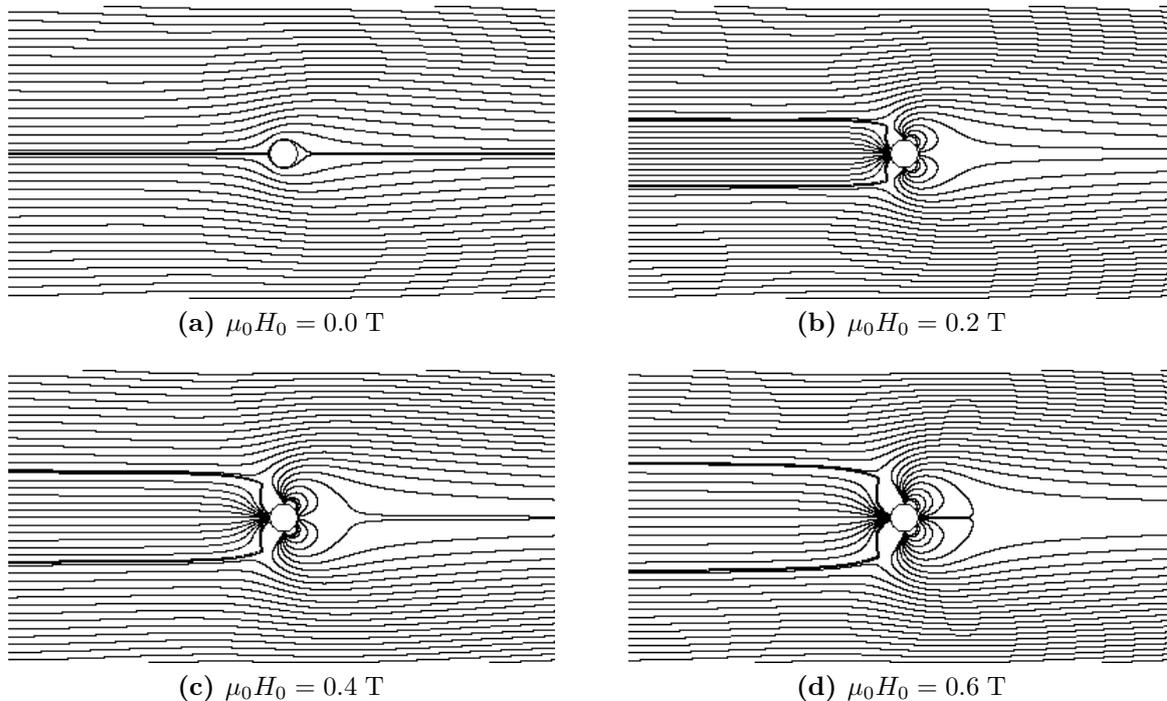


Figure 5.1: Streamlines indicating the trajectories of the single domain nanoparticles, calculated using OpenFOAM, as they traverse the control volume for different magnitudes of the externally applied magnetic field, \vec{H}_0 .

The resulting CCS, λ_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.

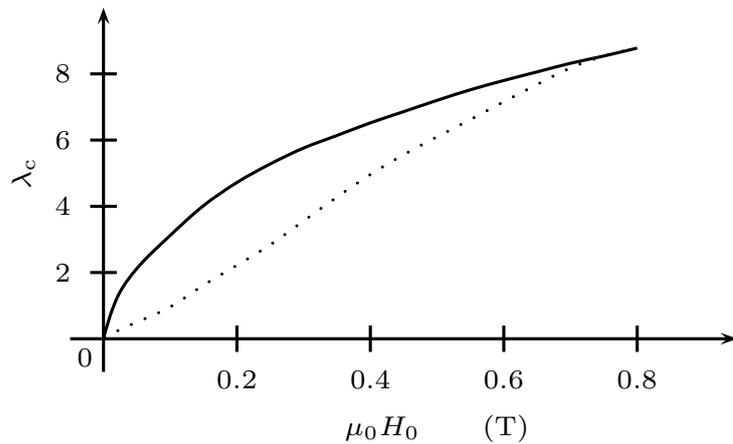


Figure 5.2: Capture cross section, λ_c , plotted as a function of the applied magnetic field strength, $\mu_0 H_0$, calculated using (—) the Langevin function as appropriate for single domain nanoparticles and (····) following Avilés et al. for multiple domain particles.

Property	Value	SI Unit	Property	Value	SI Unit
ρ_b	1040.0	kg/m ³	$\chi_{seed,0}$	1 000	—
η_b	0.002	kg/ms	$M_{seed,s}$	1 397 000	A/m
u_0	0.001	m/s	$M_{fm,p,s}$	1 735 000	A/m
$\mu_0 H_0$	0.0–0.8	kg/s ² A	R_{seed}	1.0×10^{-6}	m
$x_{fm,p}$	0.4	—	R_p	50×10^{-9}	m
$\rho_{fm,p}$	7 850	kg/m ³	$\rho_{pol,p}$	950	kg/m ³
$\chi_{fm,p,0}$	1 000	—			

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-

tion 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

$$\vec{F}_{\text{int}_n} = (\vec{m}_n \cdot \nabla) \vec{B}_{\text{total}_n} \quad (5.7)$$

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

$$\vec{m}_n = \omega_{\text{fm,p}} V_{p_n} M_{\text{fm,p,s}} L(\beta) \frac{\vec{B}_{\text{total}_n}}{B_{\text{total}_n}}, \quad (5.8)$$

where V_{p_n} 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}_{total_n} is the total magnetic flux acting on MDCP n . \vec{B}_{total_n} is taken as

$$\vec{B}_{\text{total}_n} = \vec{B} + d\vec{B}_1 + \dots + d\vec{B}_{(n-1)} + d\vec{B}_{(n+1)} \dots + d\vec{B}_N \quad (5.9)$$

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

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

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,

\vec{F}_{i_n} , as

$$\vec{F}_{s_n} + \vec{F}_{\text{int}_n} = \vec{F}_{i_n}. \quad (5.11)$$

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

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

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

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

where η_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}_{hyd_n} , which acts on MDCP n due to presence of other $(N - 1)$ MDCPs, can be written as [58],

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

where ξ_{ni} is the modification due to the hydrodynamic interaction given by

$$\xi_{ni} = -6\pi \eta_b R_{p_n} \frac{3 R_{p_i}}{4 |\vec{r}_n - \vec{r}_i|} \left(\mathbf{1} + \frac{(\vec{r}_n - \vec{r}_i) \otimes (\vec{r}_n - \vec{r}_i)}{|\vec{r}_n - \vec{r}_i|^2} \right) \quad (5.15)$$

where R_{p_i} 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}_{i_n} , as

$$\vec{F}_{s_n} + \vec{F}_{\text{hyd}_n} + \vec{F}_{\text{int}_n} = \vec{F}_{i_n}. \quad (5.16)$$

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

$$6\pi\eta_b R_{p_n} (\vec{v}_b - \vec{v}_{p_n}) + \sum_{\substack{i=1 \\ i \neq n}}^N \xi_{ni} \cdot (\vec{v}_b - \vec{v}_{p_i}) + (\vec{m}_n \cdot \nabla) \vec{B}_{\text{total}_n} = 0. \quad (5.17)$$

Hence, we can obtain \vec{v}_{p_n} 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 [21]. 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, λ_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

- 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).

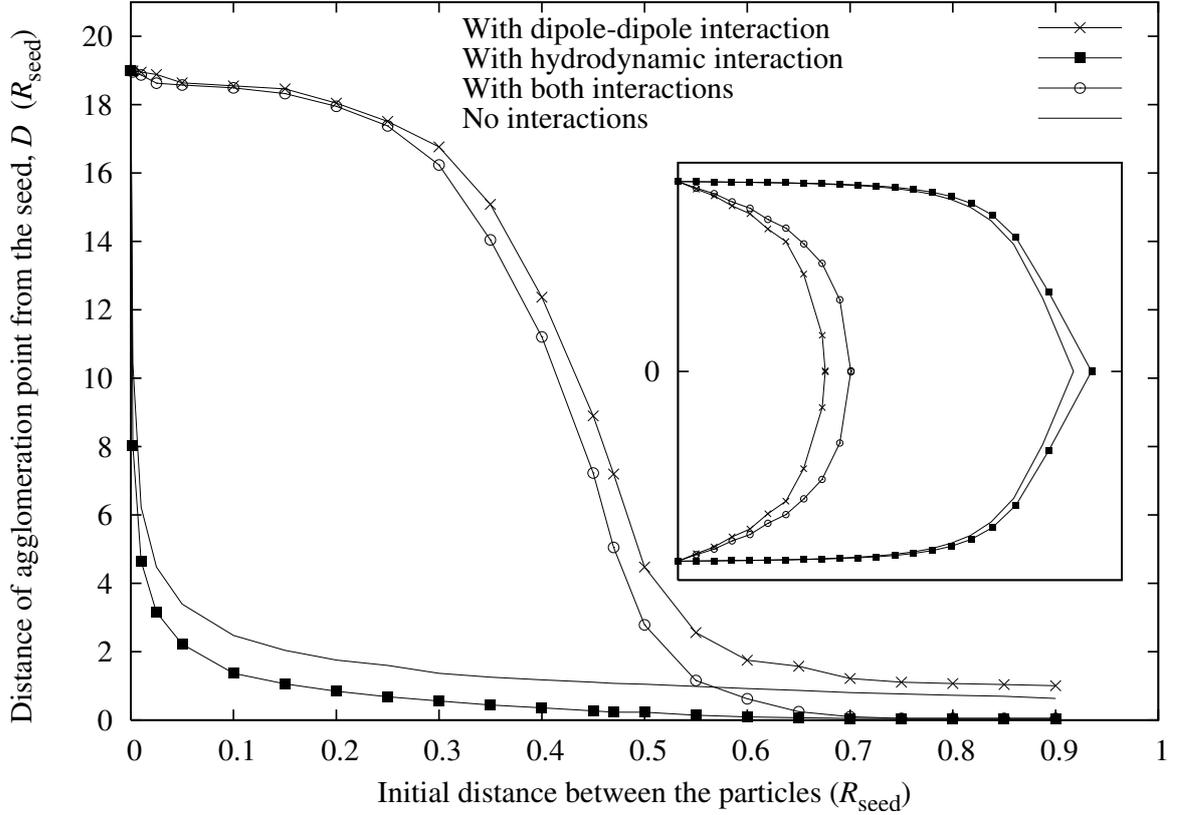


Figure 5.3: Distance of agglomeration point from the seed plotted against initial distance between the MDCPs, D , with (a) dipole-dipole magnetic interaction only, (b) hydrodynamic interaction only, (c) both interactions and (d) no interactions between the MDCPs. All other conditions are as the reference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

5.4.2 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, λ_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 λ_c^* , corresponding to the initial position $(-20, \lambda_c^*)$ where λ_c^* is 4.47. This critical value λ_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 λ_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 $0.25 R_{\text{seed}}$ corresponding to a $\approx 6\%$ absolute increase in λ_c .

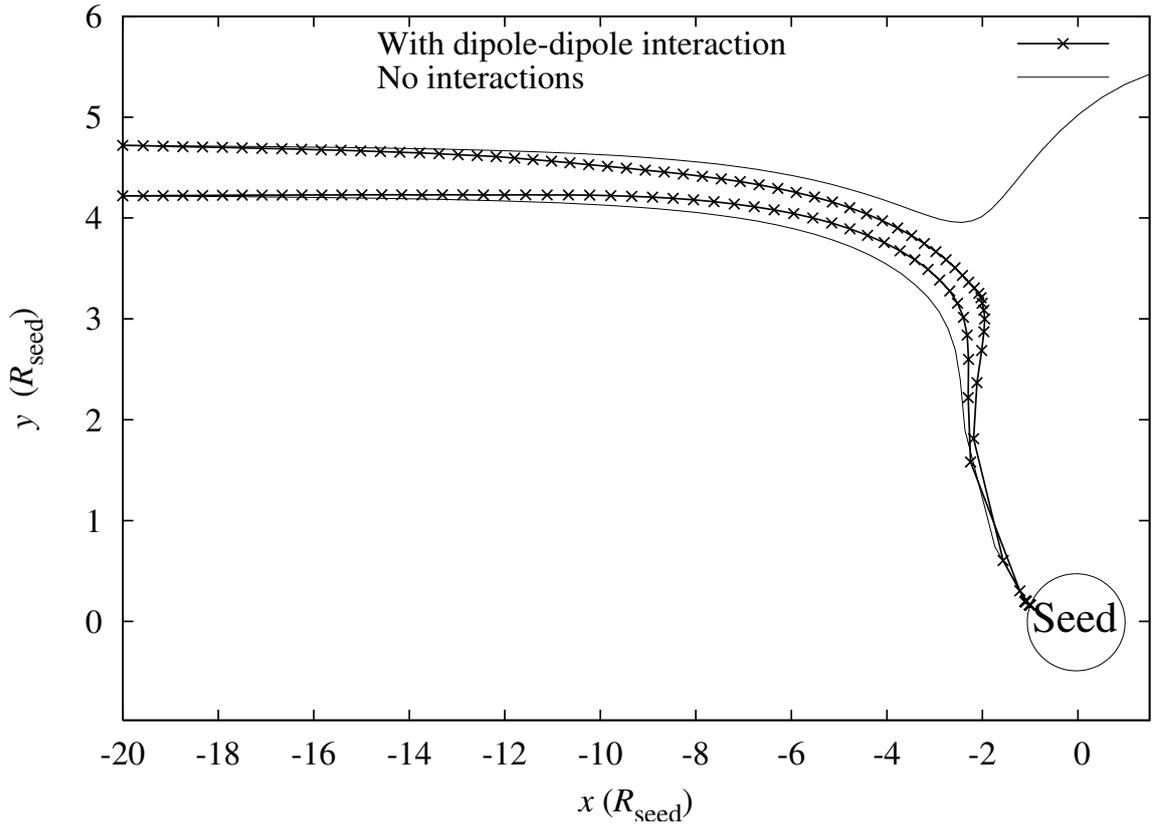


Figure 5.4: The trajectories of the MDCPs are presented with the magnetic dipole-dipole interaction and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda_c^* + 0.25)$ & $(-20, \lambda_c^* - 0.25)$. With the magnetic dipole-dipole interaction both MDCPs are captured.

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 λ_c .

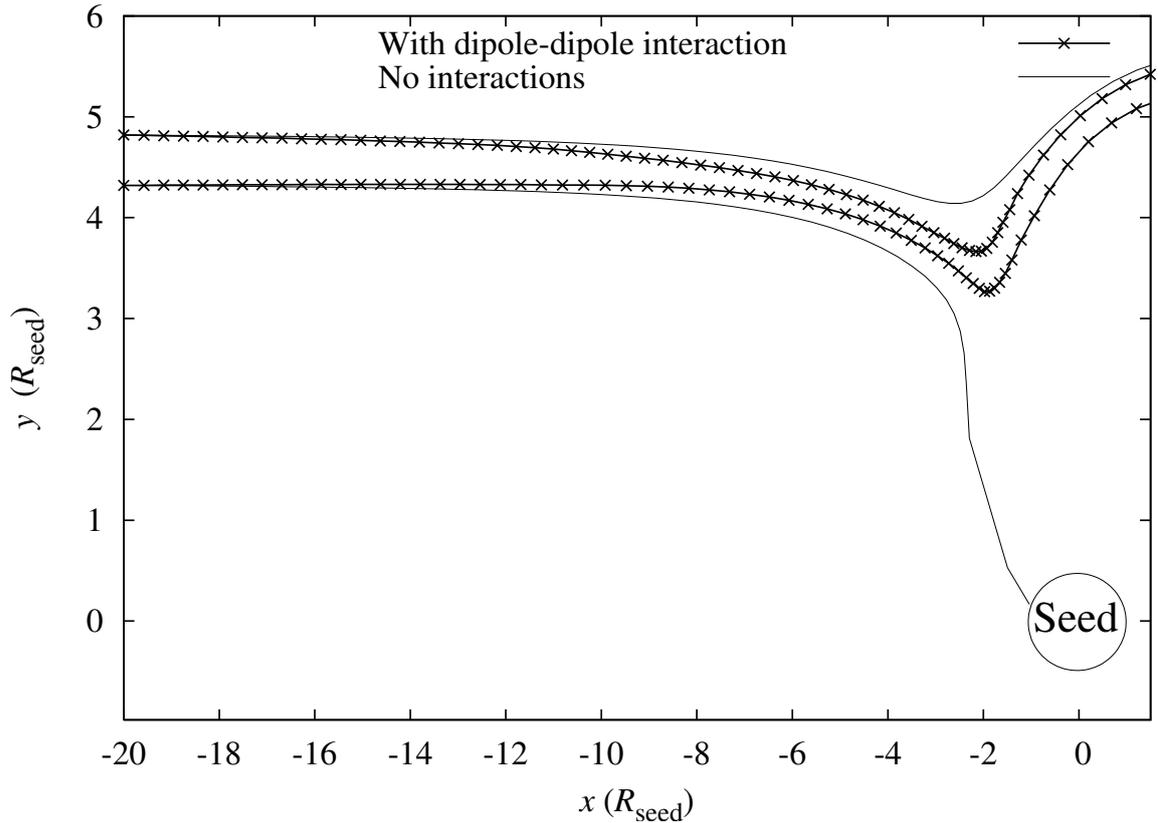


Figure 5.5: The trajectories of the MDCPs are presented with the magnetic dipole-dipole interaction and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda_c^* + 0.35)$ & $(-20, \lambda_c^* - 0.15)$. With the magnetic dipole-dipole interaction, neither MDCP is captured.

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 λ_c .

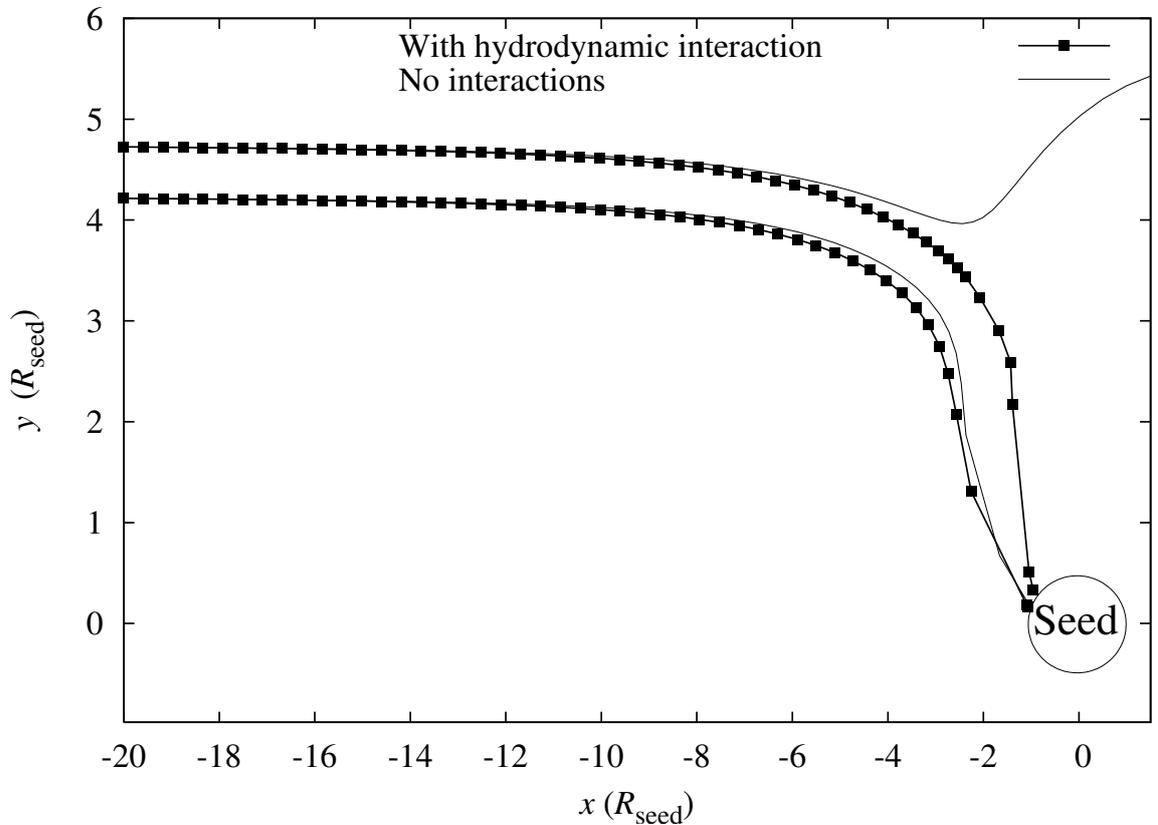


Figure 5.6: The trajectories of the MDCPs are presented with the hydrodynamic interaction and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda_c^* + 0.255)$ & $(-20, \lambda_c^* - 0.255)$. With the hydrodynamic interaction both MDCPs are captured.

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 λ_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.

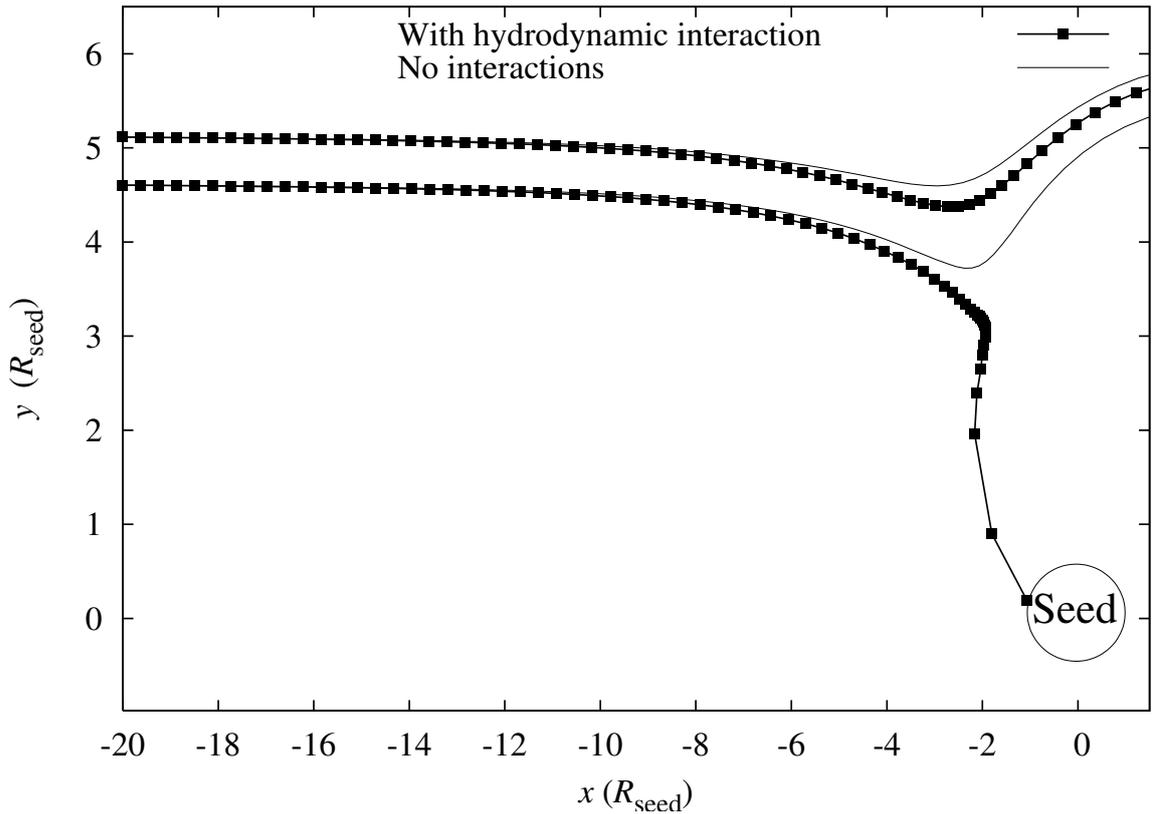


Figure 5.7: The trajectories of the MDCPs are presented with the hydrodynamic interaction and without any interaction. Initial position of MDCP 1 & 2 are $(-20, \lambda_c^* + 0.644)$ & $(-20, \lambda_c^* + 0.134)$. With hydrodynamic interaction, MDCP 2 is now captured.

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 λ_c . In this case, the magnetic dipole-dipole and hydrodynamic interactions both have a positive effect on the CCS of the system.

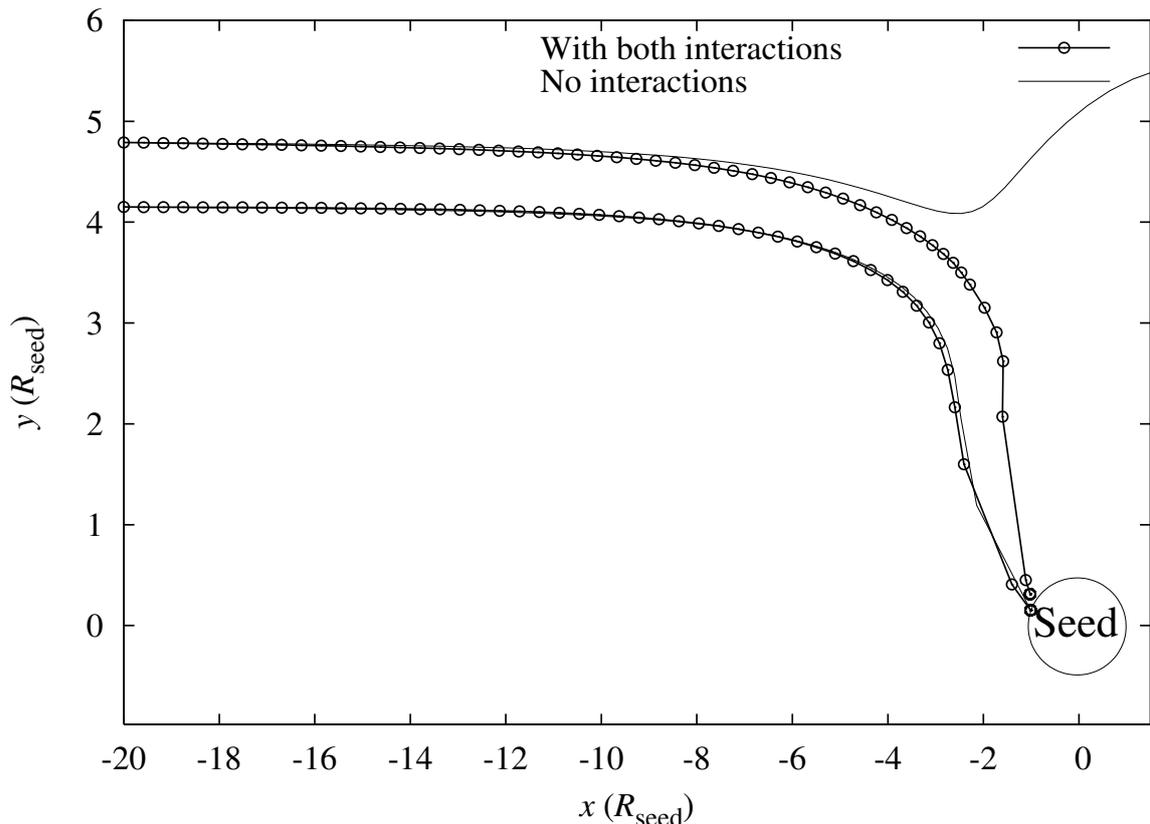


Figure 5.8: The trajectories of the MDCPs are presented with both interactions and no interactions. Initial position of MDCP 1 & 2 are $(-20, \lambda_c^* + 0.32)$ & $(-20, \lambda_c^* - 0.32)$. With both interactions both MDCPs captured.

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

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 effects 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 $0.16 R_{\text{seed}}$, 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 $0.138 R_{\text{seed}}$, relative to the case with hydrodynamic interaction only when the value of D is 0.40. These apparent imbalances we attribute to the inherent 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 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 nanoparticles 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 nanoparticles, 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 altered 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_{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 CEs for identical MDCPs with initial radius $R_p = 0.435 \mu\text{m}$ 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 $(4/3 \pi r^3)$ [3]. 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 λ_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, λ_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 λ_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 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 5.9.

In figures 5.9 and 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

Properties	Symbol	Units	Values	Data type
<i>MDCPs Properties</i>				
Polymer material	-	-	P(S/V-COOH)Mag	Physical
Radius	R_p	μm	0.435	Physical
Saturation magnetisation	$M_{p,s}$	kA/m	22.4	Measured
<i>MDCPs Magnetic Material Properties</i>				
Material	-	-	Magnetite	Physical
Weight content	$x_{\text{fm},p}$	wt%	25	Physical
Volume content	$\omega_{\text{fm},p}$	-	6.4	Measured
Saturation magnetisation	$M_{\text{fm},p,s}$	kA/m	351.9	Measured
Magnetic moment	$m_{\text{fm},p}$	Am^2	2.03×10^{-19}	Measured
Radius	$R_{\text{fm},p}$	nm	5.18	Calculated
<i>Physical Properties</i>				
Temperature	T	K	300	Physical
Boltzmann's constant	k_B	J/K	1.38×10^{-23}	Physical
Permeability of vacuum	μ_0	Tm/A	$4\pi \times 10^{-7}$	Physical
<i>Applied Field Properties</i>				
Magnitude	$\mu_0 H_0$	T	0.0–0.7	Physical
Angle of field direction	φ	-	$\pi/2$	Physical
<i>Stent Properties</i>				
Material	-	-	SS 430	Physical
Wire radius	R_{wire}	μm	62.5	Physical
Loop separation	h	mm	2	Physical
Number of loops	N_l	-	10	Physical
Coil length	L	cm	2	Physical
Saturation magnetisation	$M_{\text{implant},s}$	kA/m	1261	Measured
Magnetic susceptibility	$\chi_{\text{implant},0}$	-	1000	Physical
<i>Blood & Vessel Properties</i>				
Velocity	u_0	cm/s	2.1, 4.2, 10.6, 21.2, 42.4	Physical
Volume	V_{blood}	ml	10	Physical
Density	ρ_b	kg/m^3	1000	Physical
Viscosity	η_b	kg/ms	1.0×10^{-3}	Physical
Vessel radius	R_{vessel}	mm	0.5	Physical

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

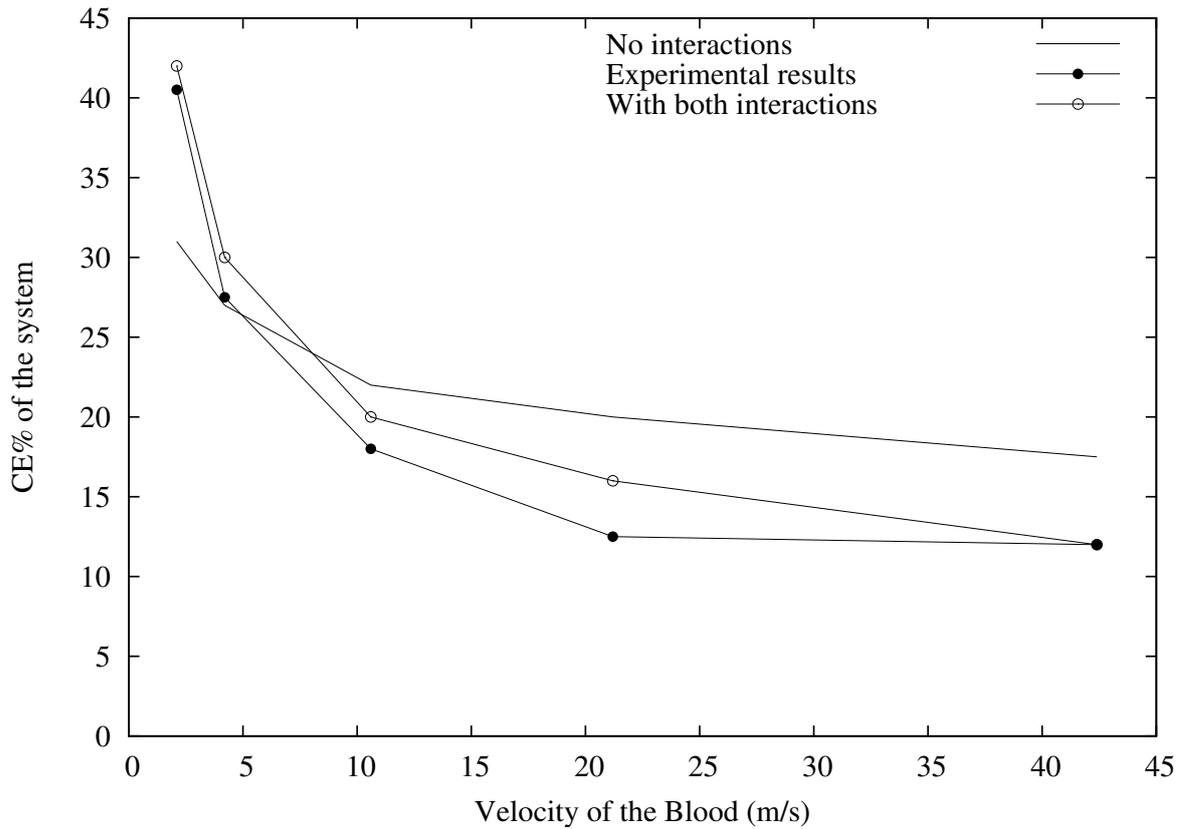


Figure 5.9: The collection efficiency (CE) of the system plotted as a function of the blood velocity at the applied field $\mu_0 H_0 = 0.65$ T.

(≤ 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.

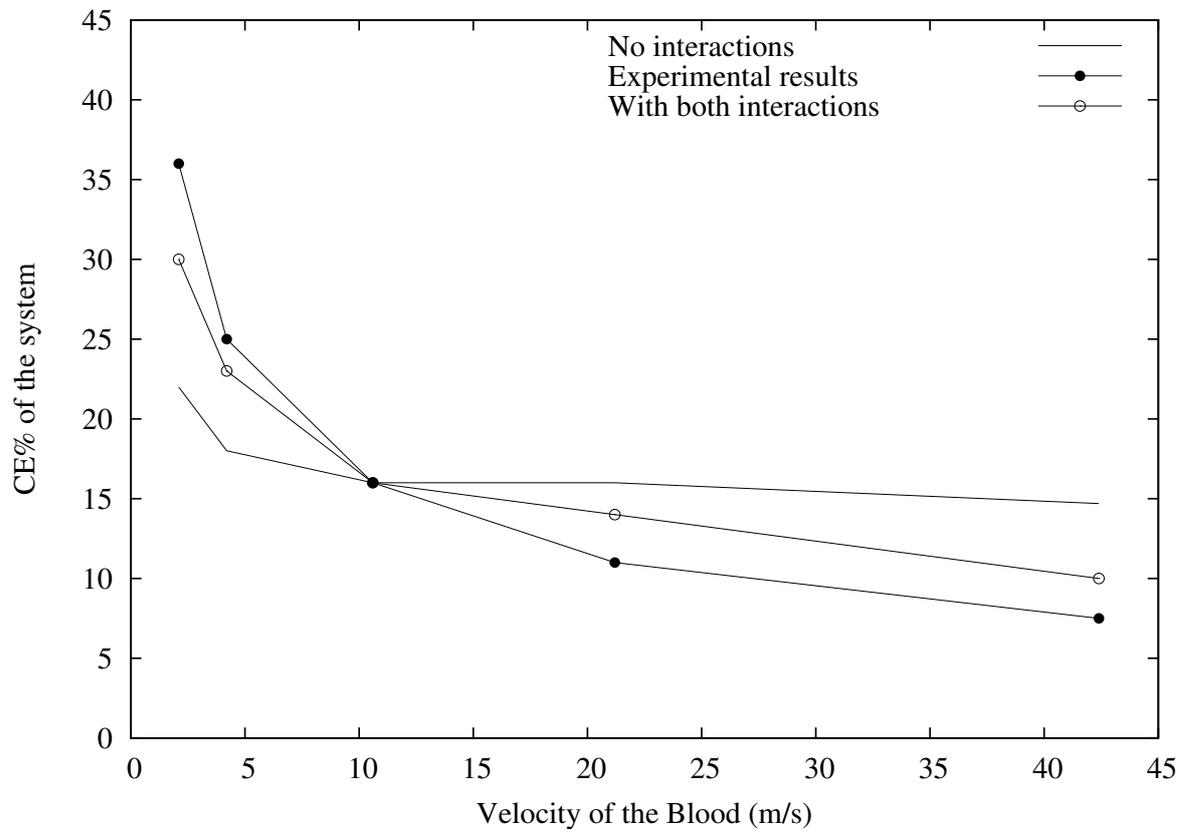


Figure 5.10: The collection efficiency (CE) of the system plotted as a function of the blood velocity at the applied field $\mu_0 H_0 =$ of 0.17 T.

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

- [1] AdvancedMagnetics (2009). *Advanced Magnetics*.
<http://www.advancedmagnetics.com/doc/prod/combindex.htm>.
- [2] Aharoni, A. (1996). *Introduction to the Theory of Ferromagnetism*. Oxford University Press.
- [3] Allen, E., Smith, P., and Henshaw, J. (2001). A review of particles agglomeration. *US Department of Energy*.
- [4] Ampère, A. M. (1958). *Theorie mathematique des phenomenes electrodynamiques uniquement deduite de l'experience. reprinted by Blanchard, Paris*.
- [5] Avilés, M. O., Ebner, A. D., Chen, H., Rosengart, A. J., Kaminski, M. D., and Ritter, J. A. (2005). Theoretical analysis of a transdermal ferromagnetic implant for retention of magnetic drug carrier particles. *J. Magn. Magn. Mater.*, **293**, 605–615.
- [6] Avilés, M. O., Ebner, A. D., and Ritter, J. A. (2007a). Ferromagnetic seeding for the magnetic targeting of drugs and radiation in capillary beds. *J. Magn. Magn. Mater.*, **310**, 131–144.
- [7] Avilés, M. O., Chen, H., Ebner, A. D., Rosengart, A. J., Kaminski, M. D., and Ritter, J. A. (2007b). In vitro study of ferromagnetic stents for implant assisted-magnetic drug targeting. *J. Magn. Magn. Mater.*, **311**, 306–311.
- [8] Avilés, M. O., Ebner, A. D., and Ritter, J. A. (2008a). Implant assisted-magnetic drug targeting: comparison of in vitro experiments with theory. *J. Magn. Magn. Mater.*, **320**, 2704–2713.
- [9] Avilés, M. O., Ebner, A. D., and Ritter, J. A. (2008b). In vitro study of magnetic particle seeding for implanted assisted-magnetic drug targeting. *J. Magn. Magn. Mater.*, **320**, 2640–2646.

- [10] Avilés, M. O., Ebner, A. D., Mangual, J. O., and Ritter, J. A. (2008c). Isolated swine heart ventricle perfusion model for implanted assisted-magnetic drug targeting. *Int. J. Pharm.*, **361**, 202–208.
- [11] Babincova, M., Leszczynska, D., Sourivong, P., and Babinec, P. (2001). Lysis of photosensitized erythrocytes in an alternating magnetic field. *J. Magn. Magn. Mater.*, **225**, 194–196.
- [12] Bean, C. P. and Livingston, J. D. (1959). Superparamagnetism. *J. Appl. Phys.*, **30**, 120S–129S.
- [13] Becker, R. (1964). *Electromagnetic Fields and Interactions*. Dover Publications Inc., New York.
- [14] Berry, C. C. (2005). Possible exploitation of magnetic nanoparticle-cell interaction for biomedical application. *J. Mater. Chem.*, **15**, 543.
- [15] Berry, C. C. and Curtis, A. S. G. (2003). Functionalisation of magnetic nanoparticles for applications in biomedicine. *J. Appl. Phys.*, **36**, R198–R206.
- [16] Brajdic-Mitidieri, P. (2005). *Advanced Modelling of Elastohydrodynamic Lubrication*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [17] Brennan, D. (2001). *The Numerical Simulation of Two-Phase Flows in Settling Tanks*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [18] Brown, W. F. (1963). Thermal fluctuations of a single domain ferromagnetic particle. *Physical Review*, **130**, 1677.
- [19] Brown, W. F. (1966). *Magnetostatic Principles in Ferromagnetism*. Springer, New York.
- [20] Bryant, H. C., Sergatskov, D. A., Lovato, D., Adolphi, N. L., Larson, R. S., and Flynn, E. R. (2007). Magnetic needles and superparamagnetic cells. *Phys. Med. Biol.*, **52**, 4009–4025.

- [21] C. I. Mikkelsen, M. F. Hansen, H. B. (2005). Theoretical comparison of magnetic and hydrodynamic interactions between magnetically tagged particles in microfluidic systems. *J. Magn. Magn. Mater.*, **293**, 578–583.
- [22] Chen, H., Ebner, A. D., Kaminski, M. D., Rosengart, A. J., and Ritter, J. A. (2005). Analysis of magnetic drug carrier particle capture by a magnetizable intravascular stent: Parametric study with multi-wire two-dimensional model. *J. Magn. Magn. Mater.*, **293**, 616–632.
- [23] Christman, J. R. (1988). *Fundamentals of Solid State Physics*. John Wiley.
- [24] Coffey, W. T., Cregg, P. J., and Kalmykov, Y. (1993). On the theory of Debye and Néel relaxation of single domain ferromagnetic particles. *Adv. Chem. Phys.*, **83**, 263–464.
- [25] CRANN (2009). *CRANN*. <http://www.crann.tcd.ie>.
- [26] Cregg, P. J. and Bessais, L. (1999). Series expansions for the magnetisation of a solid superparamagnetic system of non-interacting particles with anisotropy. *J. Magn. Magn. Mater.*, **202**, 554–564.
- [27] Cregg, P. J., Crothers, D. S. F., and Wickstead, A. W. (1994). An approximate formula for the relaxation time of a single domain ferromagnetic particle with uniaxial anisotropy and collinear field. *J. Appl. Phys.*, **76**, 4900.
- [28] Cregg, P. J., Murphy, K., and Mardinoglu, A. (2008). Calculation of nanoparticle capture efficiency in magnetic drug targeting. *J. Magn. Magn. Mater.*, **320**, 3272–3275.
- [29] Cregg, P. J., Murphy, K., and Mardinoglu, A. (2009a). Inclusion of magnetic dipole-dipole and hydrodynamic interactions in implant assisted magnetic drug targeting. *Accepted by J. Magn. Magn. Mater.*
- [30] Cregg, P. J., Murphy, K., and Mardinoglu, A. (2009b). Inclusion of many particle interactions in implant (stent) assisted magnetic drug targeting. *For submission to J. Magn. Magn. Mater.*

- [31] Crommelin, D. J. A., Scherphof, G., and Storm, G. (1995). Active targeting with particulate carrier systems in the blood compartment. *Adv. Drug Deliv. Rev.*, **17**, 49–60.
- [32] Frenkel, J. and Dorfman, J. (1930). Spontaneous and induced magnetization in ferromagnetic bodies. *Nature*, **126**, 274.
- [33] Furlani, E. J. and Furlani, E. P. (2007). A model for predicting magnetic targeting of multifunctional particles in the microvasculature. *J. Magn. Magn. Mater.*, **312**, 187–193.
- [34] Gilchrist, R. K. (1957). Selective inductive heating of lymph nodes. *Ann. Surg.*, **146**, 596.
- [35] Goodwin, S., Peterson, C., Hoh, C., and Bittner, C. (1999). Targeting and retention of magnetic targeted carriers (mtcs) enhancing intra-arterial chemotherapy. *J. Magn. Magn. Mater.*, **194**, 132.
- [36] Grief, A. D. and Richardson, G. (2005). Mathematical modelling of magnetically targeted drug delivery. *J. Magn. Magn. Mater.*, **293**, 455–463.
- [37] Häfeli, U. (2001). Radiolabeling of magnetic particles with rhenium-188 for cancer therapy. *J. Magn. Magn. Mater.*, **225**, 73.
- [38] Häfeli, U. (2004). Magnetically modulated therapeutic systems. *Int. J. Pharm.*, **277**, 19.
- [39] Hill, D. P. (1998). *The Computer Simulation of Dispersed Two-phase Flows*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [40] Hutchings, J. K. (2000). *On Modelling the Mass of Arctic Sea Ice*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.

- [41] Iacob, G., Rotariu, O., Strachan, N. J. C., and Häfeli, U. O. (2004). Magnetizable needles and wires - modeling an efficient way to target magnetic microspheres in vivo. *Biorheology*, **41**, 599–612.
- [42] Jakubovics, J. P. (1987). *Magnetism and Magnetic Materials*. The Institute of Metals.
- [43] Jasak, H. (1996). *Error Analyses and Estimation for the Finite Volume Method with Applications to Fluid Flows*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [44] Jasak, H., Jemcov, A., and Tukovic, Z. (2007). Openfoam: A C++ library for complex physics simulations. *International Workshop on Coupled Methods in Numerical Dynamics, IUC, Dubrovnik, Croatia, September 19th-21st*.
- [45] Jiles, D. (1991). *Introduction to Magnetism and Magnetic Materials*. Chapman and Hall, USA.
- [46] Jordan, A., Scholz, R., Maier-Hauff, K., Johannsen, M., Wust, P., Nadobny, J., Schirra, H., Schmidt, H., Deger, S., Loening, S., Lanksch, W., and Felix, R. (2001). Presentation of a new magnetic field therapy system for the treatment of human solid tumors with magnetic field hyperthermia. *J. Magn. Magn. Mater.*, **225**, 118–126.
- [47] Juretic, F. (2004). *Error Analysis in Finite Volume CFD*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [48] Karac, A. (2003). *Drop impact of fluid filled polyethylene containers*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [49] Leach, J. H. (2003). *Magnetic Targeted Drug Delivery*. MSc Thesis, Virginia Polytechnic and State University.
- [50] Lemke, A. J., von Pilsach, M.-I. S., Lübke, A. S., Bergemann, C., Riess, H., and Felix, R. (2004). MRI after magnetic drug targeting in patients with advanced solid malignant tumors. *Eur. Radiol.*, **14**, 1949–1955.

- [51] Lübbe, A. S., Bergemann, C., Brock, J., and McClure, D. G. (1999). Physiological aspects in magnetic drug-targeting. *J. Magn. Magn. Mater.*, **194**, 149–155.
- [52] Lübbe, A. S., Aleciou, C., and Bergemann, C. (2001). Clinical applications of magnetic drug targeting. *J. Surg. Res.*, **95**, 200–206.
- [53] Magforce (2008). *Magforce*. <http://www.magforce.de>.
- [54] Mangani, L. (2008). *Development and Validation of an Object Oriented CFD Solver for Heat Transfer and Combustion Modeling in Turbomachinery Applications*. Ph.D. Thesis, Universita degli studi di Firenze, Italy.
- [55] Mattis, D. C. (1981). *The Theory of Magnetism 1, Statics and Dynamics*. Springer-Verlag Berlin Heidelberg, New York.
- [56] Mattis, D. C. (2006). *The Theory of Magnetism Made Simple: an introduction to physical concepts and to some useful mathematical concepts*. World Scientific Publishing.
- [57] Mehasni, R., Feliachi, M., and Latreche, M. (2007). Effect of the magnetic dipole dipole interaction on the capture efficiency in open gradient magnetic separation. *IEEE Trans. Magn.*, **43**, 3488.
- [58] Mikkelsen, C. I. (2005). *Magnetic Separation and Hydrodynamic interactions in Microfluidic Systems*. Ph.D. Thesis, Department of Micro and NanoTechnology, Technical University of Denmark.
- [59] Morales, M. P., Bomati-Miguel, O., de Alejo, R. P., Ruiz-Cabello, J., Veintemillas-Verdaguer, S., and O’Grady, K. (2003). Contrast agents for MRI based on iron oxide nanoparticles prepared by laser pyrolysis. *J. Magn. Magn. Mater.*, **266**, 102–109.
- [60] Neuberger, T., Schöpf, B., Hofmann, H., Hofmann, M., and von Rechenberg, B. (2005). Superparamagnetic nanoparticles for biomedical applications: Possibilities and limitations of a new drug delivery system. *J. Magn. Magn. Mater.*, **293**, 483–496.

- [61] Nilsson, H. and Gyllenram, W. (2007). *Experiences with OpenFOAM for water turbine applications*. Proceedings of the 1st OpenFOAM International Conference, 26-27 November 2007, Beaumont House, Old Windsor, United Kingdom.
- [62] Oersted, H. C. (1820). Experiment on the effects of a current on the magnetic needle. *Annals of Philosophy*, **16**.
- [63] OpenFOAM (2009a). *OpenFOAM: Programmer's Guide*.
- [64] OpenFOAM (2009b). *OpenFOAM: User Guide*.
- [65] Pankhurst, Q. A., Connolly, J., Jones, S. K., and Dobson, J. (2003). Applications of magnetic nanoparticles in biomedicine. *J. Appl. Phys.*, **36**, R167–R181.
- [66] Ritter, J. A., Ebner, A. D., Daniel, K. D., and Stewart, K. L. (2004). Application of high gradient magnetic separation principles to magnetic drug targeting. *J. Magn. Magn. Mater.*, **280**(2-3), 184–201.
- [67] Rivola, V. (2007). *Comparative Study of the CFD codes Mistral and OpenFOAM*. Ph.D. Thesis, Hotel d'Entreprises, Parc Technologique Cap Delta 09340 Verniolle, France.
- [68] Rosengart, A. J., Kaminski, M. D., Chen, H., Caviness, P. L., Ebner, A. D., and Ritter, J. A. (2005). Magnetizable implants and functionalised magnetic carriers: A novel approach for non-invasive yet targeted drug delivery. *J. Magn. Magn. Mater.*, **293**, 633–638.
- [69] Rusche, H. (2002). *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [70] Savelyev, I. V. (1980). *Physics A General Course, Electricity and Magnetism Waves Optics*. MIR Publishers.
- [71] Scherer, F., Anton, M., Schillinger, U., Henke, J., Bergemann, C., Krüger, A., Gsbacher, B., and Plank, C. (2002). Magnetofection: enhancing and targeting gene delivery by magnetic force in vitro and in vivo. *Gene Therapy*, **9**, 102.

- [72] Schütt, W., Grüttner, C., Teller, J., Westphal, F., Häfeli, U., Paulke, B., Goetz, P., and Finck, W. (1999). Biocompatible magnetic polymer carriers for in vivo radionuclide delivery. *Artif Organs*, **23**, 98.
- [73] Senyei, A., Widder, K., and Czerlinski, G. (1978). Magnetic guidance of drug-carrying microspheres. *J. Magn. Magn. Mater.*, **49**, 3578.
- [74] Shieh, D., Cheng, F., Su, C., Yeh, C., Wu, M., Wu, Y., Tsai, C., Wu, C., Chen, D., and Chou, C. (2005). Aqueous dispersions of magnetite nanoparticles with NH₃ surfaces for magnetic manipulations of biomolecules and mri contrast agents. *Biomaterials*, **26**, 7183.
- [75] Shliomis, M. I. (1974). Magnetic fluids. *Sov. Phys. Usp.*, **17**, 153–169.
- [76] Simkin, J. and Trowbridge, C. W. (1979). On the use of the total scalar potential in the numerical solution of field problems in electromagnetics. *I JNME*, **14**, 432–440.
- [77] Torchilin, W. P. (2000). Drug targeting. *J. Pharm. Sci.*, **11**, S81–S91.
- [78] Tropsa, V. (2001). *Predicting Residual Stresses due to solidification in cast Plastic Plates*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [79] Ubbink, O. (1997). *Numerical prediction of two fluid systems with sharp interfaces*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [80] Villiers, E. (2006). *The Potential of Large Eddy Simulation for the Modeling of Wall Bounded Flows*. Ph.D. Thesis, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London.
- [81] W. F. Brown, J. (1963). *Micromagnetics*. Interscience, New York.
- [82] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *American Institute of Physics*.

- [83] Yellen, B. B., Forbes, Z. G., Halverson, D. S., Fridman, G., Barbee, K. A., Chorny, M., Levy, R., and Friedman, G. (2005). Targeted drug delivery to magnetic implants for therapeutic applications. *J. Magn. Magn. Mater.*, **293**, 647–654.

Appendix A

Scalar Potential Formulation

When implementating 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, $\alpha_{\text{fm,p}}$, for a ferromagnetic material in the MDCP as,

$$\alpha_{\text{fm,p}} = \min \left(\frac{\chi_{\text{fm,p,o}}}{3 + \chi_{\text{fm,p,0}}}, \frac{M_{\text{fm,p,s}}}{3H_0} \right) \quad (\text{A.1})$$

where $\chi_{\text{fm,p,0}}$ is the (volumetric) magnetic susceptibilities of the ferromagnetic material in the MDCP with zero field ($H_0 = 0$) and $M_{\text{fm,p,s}}$ 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 the induced magnetisation of the ferromagnetic material are obtained as below,

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

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

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

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

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, χ_{implant} , and the induced magnetisation of the implant, M_{implant} , in the 2D case are written as,

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

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

For calculating the relative permeability of implant, μ_{implant} , we use the formula $\mu_{\text{implant}} = 1 + \chi_{\text{implant}}$ and for the relative permeability of space, μ_{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. [6] the same approach is taken for the implant and for the MDCPs and so

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

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

A.2 Solution Using Total-Total Potential Formula-tion

The associated total-total magnetic scalar potential, ψ_0 , satisfies

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

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

$$\psi_0(x, y) = -H_0 x. \quad (\text{A.9})$$

In cylindrical polar coordinates the scalar potential can be written as

$$\psi_0(r, \theta) = -H_0 r \cos \theta \quad (\text{A.10})$$

where θ 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

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

and in region 2 (space)

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

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

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

and

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

and satisfy the boundary conditions

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

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

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

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

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

and

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

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

$$b_1 = 0. \quad (\text{A.19a})$$

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

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

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

$$a_2 = \lim_{r_\infty \rightarrow \infty} (-H_0 - b_2 / r_\infty^2) = -H_0$$

Using interface condition (A.13), we have at $r = 1$

$$\begin{aligned}\mu_1 \frac{\partial \psi_1}{\partial r} = \mu_2 \frac{\partial \psi_2}{\partial r} &\implies \mu_1 (a_1 - b_1) \cos \theta = \mu_2 (a_2 - b_2) \cos \theta \\ &\implies \mu_1 a_1 = \mu_2 (a_2 - b_2)\end{aligned}\quad (\text{A.19c})$$

and applying interface condition (A.14), we have

$$\begin{aligned}\frac{\partial \psi_1}{\partial \theta} = \frac{\partial \psi_2}{\partial \theta} &\implies (a_1 + b_1) \cos \theta = (a_2 + b_2) \cos \theta \\ &\implies a_1 = a_2 + b_2.\end{aligned}\quad (\text{A.19d})$$

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

$$a_2 = -H_0 \left[\frac{r_\infty^2 (\mu_1 + \mu_2)}{-(\mu_1 - \mu_2) + r_\infty^2 (\mu_1 + \mu_2)} \right] = -H_0 \left[1 - \frac{1}{r_\infty^2} \left(\frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} \right) \right]^{-1} \quad (\text{A.20a})$$

$$\text{and} \quad a_1 = \frac{2\mu_2}{\mu_1 + \mu_2} a_2, \quad b_1 = 0, \quad b_2 = -\frac{\mu_1 - \mu_2}{\mu_1 + \mu_2} a_2. \quad (\text{A.20b})$$

If we take the limit as $\mu_1 \rightarrow \mu_2$ we expect $\psi_1 = \psi_2 = \psi_0$ and consequently $\vec{H}_1 = \vec{H}_2 = \vec{H}_0$. Thus, in this limit we find

$$\lim_{\mu_1 \rightarrow \mu_2} a_2 = -H_0, \quad \lim_{\mu_1 \rightarrow \mu_2} a_1 = -H_0, \quad \lim_{\mu_1 \rightarrow \mu_2} b_2 = 0. \quad (\text{A.21})$$

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

$$\psi_1 = (a_1 r + b_1 r^{-1}) \cos \theta \quad (\text{A.22})$$

and

$$\phi_2 = (a_2 r + b_2 r^{-1}) \cos \theta. \quad (\text{A.23})$$

Applying these conditions we obtain

$$a_1 = \frac{2\mu_2 a_2 - 2\mu_2 H_0}{\mu_1 + \mu_2}, \quad b_1 = 0, \quad (\text{A.24})$$

and

$$a_2 = -\frac{b_2}{r_\infty^2} = -H_0 \left[1 - \frac{1}{r_\infty^2} \frac{\mu_1 + \mu_2}{\mu_1 - \mu_2} \right]^{-1}, \quad b_2 = \frac{-(\mu_1 - \mu_2) a_2 + (\mu_1 - \mu_2) H_0}{\mu_1 + \mu_2} \quad (\text{A.25})$$

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]

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

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

$$\vec{H} = \vec{H}_0 - \nabla \phi \quad (\text{A.27})$$

and we have

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

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

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

The resulting f_w is

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

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.

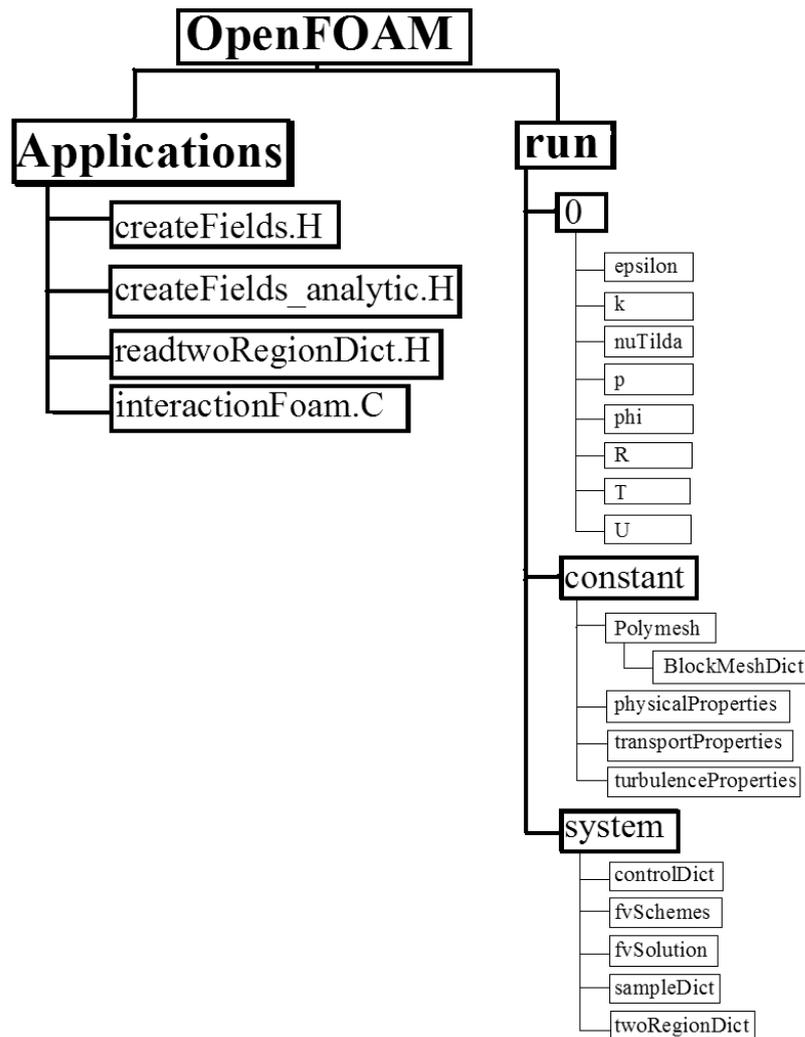


Figure B.1: Structure of the OpenFOAM program

B.1 Applications (Solver) for Seed Implant

B.1.1 createFields.H file

```

/*-----*/
=====
// \ F ield      | OpenFOAM: The Open Source CFD Toolbox
// \ O peration  |
// \ A nd        | Copyright (C) 1991-2009 OpenCFD Ltd.
// \ M anipulation |
-----

```

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.

10

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.

20

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.

30

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

40

File Name: createFields.H
 Author: mardinoghuyahoo.com

```

*/
Info << "Reading field p\n" << endl;
volScalarField p(
    IOobject(
        "p",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh

```

50

```

    );
Info << "Reading field U\n" << endl;
volVectorField U(
    IOobject(
        "U",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
# 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:    mardinogluyahoo.com
 */
Info<< "region: Reading field T\n" << endl;
volScalarField T(
    IOobject(
        "T",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    ),
    mesh
);
Info<< "region:Reading physicalProperties\n" << endl;
IOdictionary physicalProperties(
    IOobject(
        "physicalProperties",
        runTime.constant(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    )
);
Info<< "Reading parameter - correct_beta dimensions" << endl;
dimensionedScalar correct_beta (physicalProperties.lookup("correct_beta"));
Info<< "Correct_beta dimensions is" << correct_beta <<endl <<endl;

```

```

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

Info<< "Reading parameter - correct_H in OpenFOAM dimensions" << endl;
dimensionedScalar correct_H_open (physicalProperties.lookup("correct_H_open"));
Info<< "Correct_H_open dimensions is" << correct_H_open <<endl <<endl;

Info<< "Reading parameter - correct_fw in OpenFOAM dimensions" << endl;
dimensionedScalar correct_fi_open (physicalProperties.lookup("correct_fi_open"));
Info<< "Correct_fi_open dimensions is" << correct_fi_open <<endl <<endl;

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

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

Info <<"\nCalculation - (mu_0 H_0) is " <<mu_0*H_0 << endl << endl;

Info<< "\nReading parameter -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 <<endl;

Info<< "Reading parameter - saturation magnetization of the implant (M_i_s)" << endl;
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;
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;
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 <<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 <<endl;

Info<< "Reading parameter -Magnetic susceptility of Material at H_0=0<< endl;
dimensionedScalar chi_fm_p_0 (physicalProperties.lookup("chi_fm_p_0"));
Info<< "Magnetic susceptility of Material at H_0=0 is " <<chi_fm_p_0<< 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<< "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 <<endl;          90

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 <<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 <<endl;

Info<< "Reading parameter – Magnetic susceptility of Medium (chi_m)" << endl;          100
dimensionedScalar chi_m (physicalProperties.lookup("chi_m"));
Info<< "Magnetic susceptility of Medium is " <<chi_m << endl << endl;

Info<< "Reading parameter – density of the fer. 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;          110

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 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 <<endl;

Info<< "Calculating parameter –volume fraction of the fer. in the MDCP" << endl;          120
dimensionedScalar omega_fm_p = rho_p*(x_fm_p/rho_fm_p);
Info<< "Volume fraction of the ferromagnetic in the MDCP is ="<<omega_fm_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 << 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 << endl;          130

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 (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;          140
dimensionedScalar Rpv_Rw = R_pv/R_i;

```

```

Info<< "Ratio of Rpv/Rw (Rpv_Rw) is " <<Rpv_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; 150

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; 160
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; 170
Info<< "Magnetic velocity (velocity_m) is =" <<velocity_m <<endl <<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_s* 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); 180

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) 190
);
space_H_0.write();

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

```

```

        IObject::AUTO_WRITE),
        -fvc::grad(T_0)
    );
    space_H_0_cal.write();

```

200

B.1.3 *readtwoRegionDict.H file*

```

/**
 * File Name: readtwoRegionDict.H
 * Author:    mardinogluyahoo.com
 */
Info <<"\n\n@ Reading twoRegionDict\n" << endl;
IOdictionary twoRegionDict
    (
        IObject
        (
            "twoRegionDict",
            runTime.system(),
            runTime,
            IObject::MUST_READ,
            IObject::NO_WRITE)
        );
Info<<"Reading parameter - - space_mu" << endl;
scalar space_mu = readScalar(twoRegionDict.lookup("space_mu"));
Info <<"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 <<"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;

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;

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;

```

10

20

30

40

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 << "\nEvaluating analytical solution for Aviles Case" << endl;
volVectorField centres1 = T.mesh().C();
volScalarField magCentres1 = mag(centres1);
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))*(0.000001))));
// 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 Totalscalar_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)))
    );
volVectorField H_space_an_total_hand(
    IOobject ( "H_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))/(radius1*radius1*radius1*radius1)))
*vector(1,0,0))+((2*const_b2*centres1.component(vector::X)*
centres1.component(vector::Y))/(radius1*radius1*radius1*radius1))*
vector(0,1,0)))
);
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)* ((4.0*((centres1.component(vector::X)*
  (const_a2*centres1.component(vector::X)*centres1.component(vector::X)-
  3.0*const_a2*centres1.component(vector::Y)*centres1.component(vector::Y)-
  const_b2))/(radius1*radius1*radius1*radius1*radius1*radius1))*
  vector(1,0,0))+4.0*((centres1.component(vector::Y)*(3.0*const_a2*
  centres1.component(vector::X)*centres1.component(vector::X)-const_a2*
  centres1.component(vector::Y)*centres1.component(vector::Y)-const_b2))/
  (radius1*radius1*radius1*radius1*radius1*radius1))*vector(0,1,0)))
);
Info << "\n 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 << "\n 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*f_i_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)
);

```

```

    );
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))
    /(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*f_i_space_total_analytical*R_i*
    (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)
    /(mu_0*mag_H_space_an_total_hand))
);
Info << "\n\nwrite. . . .END" << endl;
Info << "PART1\n" << endl;
// *****
ofstream myfile;
myfile.open("Pjinteraction.txt");

```

```

// For Blood velocity

Info<< "\nStarting time loop\n" << endl;
    for (runTime++; !runTime.end(); runTime++) 170
    {
        Info<< "Time = " << runTime.timeName() << nl << endl;
#        include "readSIMPLEControls.H"
        p.storePrevIter();
        // Pressure-velocity SIMPLE corrector
        {
            // Momentum predictor
            tmp<fvVectorMatrix> UEqn
            (
                fvm::div(phi, U) 180
                + 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); 190
            // 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) 200
                {
                    phi -= pEqn.flux();
                }
            }
#        include "continuityErrs.H"
        // Explicitly relax pressure for momentum corrector
        p.relax();
        // Momentum corrector
        U -= fvc::grad(p)/AU;
        U.correctBoundaryConditions(); 210
    }
    turbulence->correct();
    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 220
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;
    230

// Info <<< "\n'Evaluating solution for magnetic interaction on Mikkelsen's paper" <<< endl;
// In this part the effect of dipole interaction is calculated
#include "Time.H"
#include "IOstreams.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";
    240

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]);
    250

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]);
    260

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]);
    270

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]);
    270

```

```

volTensorField gradH=fvc::grad(H_space_an_total_hand);
                                                                    280

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);
                                                                    290

//Info<< "Calculating- - distance between the center of particles" << endl;
Info <<"Parameter - - (r_1_vector)= " << (r_1_vector) <<endl;
Info <<"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();
                                                                    300

//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)
                                                                    310
);
//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))
                                                                    320
);
//Info<< "Creating vector field- - r_2 constant field for particle 2 position " << endl;
volVectorField r_2_constant_field(
    IOobject("r_2",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE),
    mesh,
    dimensionedVector("0", dimless, r_2_vector)
                                                                    330
);
//Info<< "Creating vector field- - B_r_2 constant field for particle 2 magnetic Flux " << endl;
volVectorField B_r_2_constant_field(

```

```

    IOobject("B_r_2",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::AUTO_WRITE),
    mesh,
    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*M_fm_p_s*Langevin_Cregg/(mu_0*mag_H_space_an_total_hand))*
    pow(R_p_Cregg,3)/pow(mag(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)
    );
Info<< "\nCalculating parameter - Langevin Factor for the Langevin Function" <<endl;
volScalarField Langevin_Cregg_par_one(
    IOobject(
    "Langevin_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 magmoment_Cregg_par_one(
    IOobject ( "magmoment_Cregg_one",
    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_term11))/((mu_0*mag_H_space_an_total_hand)+
    mag(F_int_term11))
    );
volVectorField F_int_par_one_new(
    IOobject ( "F_int_1_new",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::AUTO_WRITE),

```

```

        (fvc::grad((magmoment_Cregg_par_one)&(mu_0*H_space_an_total_hand+F_int_term1))) 390
    );
    //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 << "\n'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),
        (0.33333)*((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<< "\nCalculating 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 ( "magmoment_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),

```

```

        (fvc::grad((magmoment_Cregg_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_Cregg.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_Cregg.value()*u_0)));

    // 0.005 was the time step
    // 0.0025 was the time step
    // 0.00125 is the time step
    // 0.000625 was 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();

x_1=particle1_ccs.x();
y_1=particle1_ccs.y();
x_2=particle2_ccs.x();
y_2=particle2_ccs.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

```

/*-----*\
|=====|
| \ / 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       epsilon;
}
// ***** //
dimensions      [0 2 -3 0 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;
    }
}

```

```
} 50
```

```
// ***** //
```

B.2.1.2 *k* file

```
FoamFile
{
  version      2.0;
  format       ascii;
  root         "";
  case         "";
  instance     "";
  local        "";
  class        volScalarField;
  object       k; 10
}
// ***** //
dimensions    [0 2 -2 0 0 0 0];
internalField uniform 0.375;
boundaryField
{
  inlet
  {
    type      fixedValue;
    value     uniform 0.375; 20
  }
  outlet
  {
    type      zeroGradient;
  }
  top
  {
    type      zeroGradient;
  }
  bottom     30
  {
    type      zeroGradient;
  }
  seed_0
  {
    type      zeroGradient;
  }
  defaultFaces
  {
    type      empty; 40
  }
}
// ***** //
```

B.2.1.3 *nuTilda file*

```

FoamFile
{
  version      2.0;
  format       ascii;
  root         "";
  case         "";
  instance     "";
  local        "";
  class        volScalarField;
  object       nuTilda;
}
// ***** //
dimensions    [0 2 -1 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;
  }
}
// ***** //

```

B.2.1.4 *p file*

```

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

```

```

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;

```

```

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];
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;
}
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;
    }
}

```

```

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];
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);
    }
}

```

```

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)

```

```

( 20.000000 1.414214 0.000000)
( 20.000000 21.000000 0.000000)
(-20.000000 -21.000000 1.000000)
(-20.000000 -1.414214 1.000000)
(-20.000000 1.414214 1.000000)
(-20.000000 21.000000 1.000000)
( -1.414214 -21.000000 1.000000)
( -1.414214 -1.414214 1.000000)
( -1.414214 1.414214 1.000000)
( -1.414214 21.000000 1.000000)
( -0.707107 -0.707107 1.000000)
( -0.707107 0.707107 1.000000)
( 0.707107 -0.707107 1.000000)
( 0.707107 0.707107 1.000000)
( 1.414214 -21.000000 1.000000)
( 1.414214 -1.414214 1.000000)
( 1.414214 1.414214 1.000000)
( 1.414214 21.000000 1.000000)
( 20.000000 -21.000000 1.000000)
( 20.000000 -1.414214 1.000000)
( 20.000000 1.414214 1.000000)
( 20.000000 21.000000 1.000000)
);
edges
(
  arc 5 13 ( -0.000000 -2.000000 0.000000)
  arc 13 14 ( 2.000000 -0.000000 0.000000)
  arc 14 6 ( 0.000000 2.000000 0.000000)
  arc 6 5 ( -2.000000 0.000000 0.000000)
  arc 8 10 ( -0.000000 -1.000000 0.000000)
  arc 10 11 ( 1.000000 -0.000000 0.000000)
  arc 11 9 ( 0.000000 1.000000 0.000000)
  arc 9 8 ( -1.000000 0.000000 0.000000)
  arc 25 33 ( -0.000000 -2.000000 1.000000)
  arc 33 34 ( 2.000000 -0.000000 1.000000)
  arc 34 26 ( 0.000000 2.000000 1.000000)
  arc 26 25 ( -2.000000 0.000000 1.000000)
  arc 28 30 ( -0.000000 -1.000000 1.000000)
  arc 30 31 ( 1.000000 -0.000000 1.000000)
  arc 31 29 ( 0.000000 1.000000 1.000000)
  arc 29 28 ( -1.000000 0.000000 1.000000)
);
blocks
(
  hex ( 0 4 5 1 20 24 25 21) (32 32 1) simpleGrading (1.000000 1.000000 1)
  hex ( 1 5 6 2 21 25 26 22) (32 16 1) simpleGrading (1.000000 1.000000 1)
  hex ( 2 6 7 3 22 26 27 23) (32 32 1) simpleGrading (1.000000 1.000000 1)
  hex ( 4 12 13 5 24 32 33 25) (16 32 1) simpleGrading (1.000000 1.000000 1)
  hex ( 5 13 10 8 25 33 30 28) (16 16 1) simpleGrading (1.000000 1.000000 1)
  hex (10 13 14 11 30 33 34 31) (16 16 1) simpleGrading (1.000000 1.000000 1)
  hex (11 14 6 9 31 34 26 29) (16 16 1) simpleGrading (1.000000 1.000000 1)
  hex ( 9 6 5 8 29 26 25 28) (16 16 1) simpleGrading (1.000000 1.000000 1)
  hex ( 6 14 15 7 26 34 35 27) (16 32 1) simpleGrading (1.000000 1.000000 1)
  hex (12 16 17 13 32 36 37 33) (32 32 1) simpleGrading (1.000000 1.000000 1)
  hex (13 17 18 14 33 37 38 34) (32 16 1) simpleGrading (1.000000 1.000000 1)
  hex (14 18 19 15 34 38 39 35) (32 32 1) simpleGrading (1.000000 1.000000 1)
);

```

```
);
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_0 H_0 [ 0 -1 0 0 0 1 0] 1591549.4302045; // magnitude of H_m field for mu_0*H_0=2.0T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 1432394.4871841; // magnitude of H_m field for mu_0*H_0=1.8T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 1273239.5441636; // magnitude of H_m field for mu_0*H_0=1.6T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 1114084.6011432; // magnitude of H_m field for mu_0*H_0=1.4T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 954929.6581227; // magnitude of H_m field for mu_0*H_0=1.2T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 795774.7151023; // magnitude of H_m field for mu_0*H_0=1.0T 20
//H_0 H_0 [ 0 -1 0 0 0 1 0] 636619.7720818; // magnitude of H_m field for mu_0*H_0=0.8T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 596830.935; // magnitude of H_m field for mu_0*H_0=0.75T
H_0 H_0 [ 0 -1 0 0 0 1 0] 557042.2000000; // magnitude of H_m field for mu_0*H_0=0.7
//H_0 H_0 [ 0 -1 0 0 0 1 0] 517253.564; // magnitude of H_m field for mu_0*H_0=0.65T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 477464.8290614; // magnitude of H_m field for mu_0*H_0=0.6T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 437676.092; // magnitude of H_m field for mu_0*H_0=0.55T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 397887.3575511; // magnitude of H_m field for mu_0*H_0=0.5T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 358098.62; // magnitude of H_m field for mu_0*H_0=0.45T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 318309.8860409; // magnitude of H_m field for mu_0*H_0=0.4T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 278521.13501; // magnitude of H_m field for mu_0*H_0=0.35T 30
//H_0 H_0 [ 0 -1 0 0 0 1 0] 238732.4; // magnitude of H_m field for mu_0*H_0=0.3T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 198943.6780305; // magnitude of H_m field for mu_0*H_0=0.25T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 159154.9430205; // magnitude of H_m field for mu_0*H_0=0.2T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 119366.2072653; // magnitude of H_m field for mu_0*H_0=0.15T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 79577.4715102; // magnitude of H_m field for mu_0*H_0=0.1T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 39788.73501; // magnitude of H_m field for mu_0*H_0=0.05T
//H_0 H_0 [ 0 -1 0 0 0 1 0] 19894.36750; // magnitude of H_m field for mu_0*H_0=0.025T
//H_0 H_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] 0.000001256637062; // magnetic permabailty of free space
R_i R_i [ 0 1 0 0 0 0 0] 1000.0e-09; // implant radius 40
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] 1735000; // Saturation magnetization of the mat. in MDCP
rho_beta rho_beta [ 1 -3 0 0 0 0 0] 1040; // blood density
```

```

rho_fm_p    rho_fm_p [ 1 -3 0 0 0 0 0] 7850; // density of in MDCP
rho_pol_p   rho_pol_p [ 1 -3 0 0 0 0 0] 950;
x_fm_p      x_fm_p [ 0 0 0 0 0 0 0] 0.4; // mass fraction of material in MDCP
R_p         R_p [ 0 1 0 0 0 0 0] 50.0e-9; // MDCP radius
R_pv        R_pv [ 0 1 0 0 0 0 0] 0.000021; // parent blood vessel radius
u_0         u_0 [ 0 1 -1 0 0 0 0] 0.001; // average inlet velocity
eta_beta    eta_beta [ 1 -1 -1 0 0 0 0] 0.002; // blood viscosity (Pa s)
epsilon_p    epsilon_p [ 0 0 0 0 0 0 0] 0; // porosity of the material!!!!
R_p-Cregg   R_p-Cregg [ 0 1 0 0 0 0 0] 50.0e-9; // MDCP radius

```

B.2.2.3 transport properties file

```

FoamFile
{
  version      2.0;
  format       ascii;
  root         "";
  case         "";
  instance     "";
  local        "";
  class        dictionary;
  object       transportProperties;
}
// ***** //
transportModel Newtonian;
nu             nu [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
{
  Cmu          Cmu          [0 0 0 0 0 0 0] 0.09;
  C1           C1           [0 0 0 0 0 0 0] 1.44;
  C2           C2           [0 0 0 0 0 0 0] 1.92;
}

```

```

    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;
// * * * * * //

```

10

20

B.2.3.2 fvSchemes file

```

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

```

10

```

    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 "";
}

```

```

    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;
}
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;
}
// ***** //

```

B.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
);
// ***** //

```

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("\tConvertToMeters factor (%s) : " %
        (convertToMeters) )).strip() or convertToMeters
grading = (raw_input("\tMesh grading (%s) : " % (grading) )).strip() or grading
R_s = (raw_input("\tSeed radius (%s) : " % (R_s) )).strip() or R_s
n = (raw_input("\tNumber of seeds (%s) : " % (n) )).strip() or n
delta = (raw_input("\tDistance between (%s) : " % (delta) )).strip() or delta
# a = max(1.0, delta*0.2)
a = (raw_input("\tWidth of annulus around seed (%s) : " % (a) )).strip() or a
os.system("./build_cv %s %s %s %s %s %s > blockMeshDict" %
        (convertToMeters, grading, R_s, n, delta, a))
sys.exit(0)

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])
    if len(sys.argv)==10: # also moving origin (for adil meshes which are wire centered)
        origin = Vector(float(sys.argv[7]), float(sys.argv[8]), float(sys.argv[9]))
else:
    print "// Argument count was ", len(sys.argv)
```

```

    print "// Argument list ", sys.argv
    print "// Using default settings"

DEBUG=0

print "// Parameters :"
print "// \t convertToMeters (convertToMeters) : ", convertToMeters           50
print "// \t grading (grading)           : ", grading
print "// \t Radius of seed (R_s)       : ", R_s
print "// \t Number of seeds (n)        : ", n
print "// \t Width between seeds (delta) : ", delta
print "// \t Width of annulus around seed (a) : ", a
print "// \t Mesh origin (origin)       : ", origin
print "// "
# 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) )

```

```

vertices.append( Vector(x_1-(1+a)*tx, +8.0-2*(1+a)*ty) )
vertices.append( Vector(x_1-(1+a)*tx, +8.0) )
vertices.append( Vector(x_1-1*tx, y_1-1*ty) )
vertices.append( Vector(x_1-1*tx, y_1+1*ty) )
vertices.append( Vector(x_1+1*tx, y_1-1*ty) )
vertices.append( Vector(x_1+1*tx, y_1+1*ty) )
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) )
vertices.append( Vector(x_1+(1+a)*tx, +8.0-2*(1+a)*ty) )
vertices.append( Vector(x_1+(1+a)*tx, +8.0) )
x = 20
vertices.append( Vector(x, 0) )
vertices.append( Vector(x, +8.0-2*(1+a)*ty) )
vertices.append( Vector(x, +8.0) )
vertices.append( Vector(x, -8.0) )
vertices.append( Vector(x, -8.0+2*(1+a)*ty) )

# 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+13; B = k+11; 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) # 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+20; 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+24; B_1 = k+26; edges.append ( [A_1,B_1, (vertices[A_1]-C_1).rotate(45) + C_1] )
    A_1 = k+26; B_1 = k+27; edges.append ( [A_1,B_1, (vertices[A_1]-C_1).rotate(45) + C_1] )
    A_1 = k+27; B_1 = k+25; edges.append ( [A_1,B_1, (vertices[A_1]-C_1).rotate(45) + C_1] )
    A_1 = k+25; B_1 = k+24; 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 = []
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+16, k+21, k+22, k+17, 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, gB, 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)
front_patch = []
back_patch = []

def printHeader():
    global convertToMeters
    print r
    """/*-----*\
| ===== |

```

```

| \ \   / F i e l d       | OpenFOAM: The Open Source CFD Toolbox |
| \ \   / O p e r a t i o n | Version: 1.4                               |
| \ \ /   A n d           | Web:    http://www.openfoam.org       |
|  \ \ /   M a n i p u l a t i o n |                               |
\*-----*/
20

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

convertToMeters %s;
""" % (convertToMeters)
def printVertices():
    global vertices
    print "vertices\n%d (" % (2*len(vertices))
    for vertex in vertices: print "\t%s" % (vertex + back_offset)
    for vertex in vertices: print "\t%s" % (vertex + front_offset)
    print ");\n"
40
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"
50
def buildBlock(face):
    global vertices
    V = len(vertices)
    # grading values
    gx, gy = 1, 1
    front_patch.append("4(%s %s %s %s)" % (face[3], face[2], face[1], face[0]))
    back_patch.append("4(%s %s %s %s)" % (face[0]+V, face[1]+V, face[2]+V, face[3]+V))
    if grading>=2 and len(face)>4: gx, gy = face[4:6]
    sx, sy = 1, 1
    if grading>=2 and len(face)>6: sx, sy = face[6:]
    return ("\thex (%2d %2d %2d %2d %2d %2d %2d) (%d %d 1) simpleGrading (%f %f 1)" %
            (face[0], face[1], face[2], face[3],
             face[0]+V, face[1]+V, face[2]+V, face[3]+V, gx*grading, gy*grading, sx, sy))
60
def printBlocks():
    global blocks, vertices
    print "blocks\n("
    for block in blocks: print buildBlock(block)
    print ");\n"
70
def buildPatch (patch):
    global vertices

```

```

V = len(vertices)
(type, name, corners) = patch
result = "\t%s %s\n\t(\n" % (type, name)
for f in range(len(corners)-1):
result += "\t\t4(%2d %2d %2d %2d)\n" % (corners[f], corners[f+1], corners[f+1]
+V, corners[f]+V)
result += "\t)"
return result
def printPatches():
print "patches\n("
for patch in patches: print buildPatch(patch)
print "\tempty front\n\t("
for p in front_patch: print "\t\t%s" %p
print "\t)"
print "\tempty back\n\t("
for p in back_patch: print "\t\t%s" %p
print "\t)"
print ");\n"

```

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 "(% 10.6f % 10.6f % 10.6f)" % (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
```

40

50

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

```
/*-----*\
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
);
Info << "Reading field U\n" << endl;
volVectorField U(
    IOobject(
        "U",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
# 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

```

```

    ),
    mesh
  );
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 permability 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 << 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 <<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 <<endl;

Info<< "\nReading parameter -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<< "Reading parameter - saturation magnetization of the implant"<<endl;
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"<<endl;
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;
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;
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_0_vector = vector(H_0.value()*cos(convertToRad*theta).value(),
    H_0.value()*sin(convertToRad*theta).value(), 0.0);
Info<< " End of createDiels_analytic" <<endl;

```

D.1.3 *readtwoRegionDict.H file*

```

// file: readtwoRegionDict.H
//
// Read info from twoRegionDict
//*****
//*****
Info <<"\n\n@ Reading twoRegionDict\n" << 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 <<"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: mardinoghuyahoo.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
(
    IObject
    (
        "Zero",
        runTime.timeName(),
        mesh,
        IObject::NO_READ,
        IObject::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(
        IObject ( "Centres_seeds",
        runTime.timeName(),
        mesh,
        IObject::NO_READ,
        IObject::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_fm_p_s*Langevin_Aviles*H)/mag(H);

volVectorField fw_langevin(
IOobject ( "fw_langevin",
runTime.timeName(),
mesh,

```

```

    IOobject::NO_READ,
    IOobject::AUTO_WRITE),
    (1.0/(2.0*R_i_scale))*mu_0*fvc::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_scale))*fvc::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*volume_p*M_fm_p_s*Langevin_Aviles*(mu_0*H)/(mu_0*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_scale))*(fvc::grad((magmoment_Cregg)&(mu_0*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 \n " <<setw(4)
    <<" " <<setw(9) <<setprecision(4) <<min(mag(U))
    <<" \n" <<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))
    <<" ";

```

140

150

160

170

180

190

```

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))
    <<" ";
200

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))
    <<" ";
210

Info<<"\n# ANALYTICAL SOLUTION. . .END" << endl;

    ofstream myfile;
    ofstream langfile;
    myfile.open("stentinter.txt");
    langfile.open("stentintlang.txt");
220

/**
 * 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"
230        p.storePrevIter();
        // Pressure-velocity SIMPLE corrector
        {
            // Momentum predictor
            tmp<fvVectorMatrix> UEqn
            (
                fvm::div(phi, U)
                + turbulence->divR(U)
            );
            UEqn().relax();
240            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 arrays for storing the datas
 **/

#define Npar 20
//Store the initial and current positions of particles.
vector r_n[Npar];
//Store the new positions of particles.
vector r_n_new[Npar];
//Store the temp positions of particles (used for swaping the positions of particles).
vector r_n_temp[Npar];
//Store the distance between the particles.
vector r_np[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.
**/
for(int k=0; k<Npar; k++){
    r_n[k].x()=((k+1)%5)*x_2_original+x_1_original;
    r_n[k].y()=((k/5)*y_2_original)+y_1_original;
    r_n[k].z()=0;
// Save the initial positions of particles.
    myfile<<r_n[k].x()<<"\t " <<r_n[k].y()<<"\t " <<r_n[k].z()<<"\n";
// Initial velocity_hyd_int is set to zero.
    velocity_hyd_int[k]=vector(0,0,0);
// Initial particles have same radius.
    Radius_Particle[k]=R_p.value();
    Info <<"Radius initials[" << k << "]"<<Radius_Particle[k]<<endl;
}
myfile.close();

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

/**
* Agglomeration of particles is considered.
* Once the particles agglomerate, the position of the particle_1 will be the
* (r_1+r_2)/2 and particle_2 will be out of the space.
* 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++){
        r_np[i][j]=r_n[i]-r_n[j];
Info <<"Distance between particles between [" << i << "]" and [" << j <<"]="
    << mag(r_np[i][j])<<"\t COMPARE 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].x())>19.0){
            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())){

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[j]*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=0;

for(int i=0; i<Npar; i++){

//save the current positions of each particles
langfile<<runTime.timeName()<<"\t " <<r_n[0].x()<<"\t " <<r_n[0].y()<<"\t " <<
r_n[1].x()<<"\t " <<r_n[1].y()<<"\t " <<r_n[2].x()<<"\t " <<r_n[2].y()<<"\t " <<
r_n[3].x()<<"\t " <<r_n[3].y()<<"\t " <<r_n[4].x()<<"\t " <<r_n[4].y()<<"\t " <<
r_n[5].x()<<"\t " <<r_n[5].y()<<"\t " <<r_n[6].x()<<"\t " <<r_n[6].y()<<"\t " <<
r_n[7].x()<<"\t " <<r_n[7].y()<<"\t " <<r_n[8].x()<<"\t " <<r_n[8].y()<<"\t " <<
r_n[9].x()<<"\t " <<r_n[9].y()<<"\t " << "\n";

//Calculates the values of each particles
vector r_p_vector=vector(r_n[i].x(),r_n[i].y(),0);
// if the particle is in the space region we calculate the values for it.
if (r_p_vector.x()<19.0){
//Finds the blood velocity of each particles
vector probePoint(r_p_vector);
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_vector);
// 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_vector_field(
    IOobject("r_p_field",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::NO_WRITE),
    mesh,
    dimensionedVector("0", dimless, r_p_vector)
);

```

```

// Creates a constant vectorfield by using the applied field on particle
volVectorField B_r_p_vector_field(
  IOobject("B_r_p",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::NO_WRITE),
  mesh,
  dimensionedVector("0", dimless, (mu_0.value()*H_part_space))
);
420

// 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=vector(r_n[j].x(),r_n[j].y(),0);
430

  if((mag(r_n_vector-r_p_vector))>((Radius_Particle[i]+Radius_Particle[j])/
    R_i_scale.value()) && r_n_vector.x()<19){
    //Note that(Radius_Particle[i]+Radius_Particle[j])is the distance between the particles
    vector probePoint_n(r_n_vector);
    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]);
440

// Creates a constant vectorfield by using the position of particle
volVectorField r_n_vector_field(
  IOobject("r_n_field",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::NO_WRITE),
  mesh,
  dimensionedVector("0", dimless, r_n_vector)
);
450

// Creates a constant vectorfield by using the applied field on the particle
volVectorField B_r_n_vector_field(
  IOobject("B_r_n",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::NO_WRITE),
  mesh,
  dimensionedVector("0", dimless, (mu_0.value()*H_part_space_n))
);
460

// Calculate the forces acting on the particle due to the other particles
volVectorField F_int_p_n(
  IOobject ("F_int_p",
    runTime.timeName(),
    mesh,
    IOobject::NO_READ,
    IOobject::AUTO_WRITE),
  (1.0/3.0)*((mu_0*M_fm_p_s*Langevin_Aviles/(mu_0*mag(H)))*pow(Radius_Particle[j],3)/
  pow(mag(r_p_vector-r_n_vector),3))*((3.0*(B_r_n_vector_field&(r_positions-r_n_vector_field))*
470

```

```

        (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;
    }
}
480

// 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)
    );
490

// 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)
    );
500

// 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[i]*Radius_Particle[i]) *M_fm_p_s*Langevin_Aviles_p*
        ((mu_0*H)+modif_B))/mag((mu_0*H)+modif_B)
    );
510

// Fix the boundaries of the magnetic moment for each particle
    moment_Cregg=magmoment_Cregg_p;
520

// Calculates the modified magnetic forces for each particle due to the stent and other particles
    volVectorField F_int_overall_p(
    IOobject ( "F_int_1_new",
        runTime.timeName(),
        mesh,

```

```

IOobject::NO_READ,
IOobject::NO_WRITE),
(1.0/(R_i_scale))*(fvc::grad((moment_Cregg)&(mu_0*H+modif_B)))
);
530

//vector probePoint_p(x_p,y_p,0);
//label probeCell_one = mesh.findCell(probePoint_p);
volTensorField gradF_int_overall_p= fvc::grad(F_int_overall_p);
//vector cellCentre_one = mesh.C()[probeCell_one];
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;
}
550

// 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(1,0,0,0,1,0,0,0,1) +(normalized*normalized));
F_hyd_temp=F_hyd_temp+(hyd_force & velocity_hyd_int[j]);
}
}
F_hyd[i]=F_hyd_temp;
F_hyd_temp=vector(0,0,0);
}
else{
F_hyd[i]=vector(0,0,0);
}
}
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

```

```

Info <<"Velocity_hyd+dip[" << i << "]" << 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[j]-seed_centre)<=1.0 && r_n_new[j].x()< 19){
            r_n[j]=r_n_temp[j];
            N_count_1=N_count_1+1;
            break;
        }
        else{
            r_n[j]=r_n_new[j];
        }
    }
}
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

```

/*-----*\
|=====|

```

```

| \\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O p e r a t i o n | Version: 1.4 |
| \\      / A n d      | Web: http://www.openfoam.org |
| \\      / M a n i p u l a t i o n |
| *-----* |
FoamFile
{
  version      2.0;
  format       ascii;
  root         "";
  case        "";
  instance     "";
  local       "";
  class        volScalarField;
  object       epsilon;
}
// *****
dimensions    [0 2 -3 0 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
  {

```

```

        type          zeroGradient;
    }
    front
    {
        type          empty;
    }
    back
    {
        type          empty;
    }
}
// ***** //

```

D.2.1.2 k file

```

/*-----*\
|=====|
| \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O p e r a t i o n | Version: 1.4 |
| \ \ / / A n d | Web: http://www.openfoam.org |
| \ \ / / M a n i p u l a t i o n | |
\*-----*/
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;
    }
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

```

/*-----* 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       nuTilda;
}

```


D.2.1.4 p file

```

/*-----* C++ -*-----*\
|=====|
| \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O p e r a t i o n | Version: 1.4 |
| \ \ / / A n d | Web: http://www.openfoam.org |
| \ \ / / M a n i p u l a t i o n | |
\*-----*/

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    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;
    }
    seedtop_0
    {
        type zeroGradient;
    }
    seedbot_0
    {
        type zeroGradient;
    }
    ....
    ....
    ....
    ....
    seedtop_9
}

```

```

    {
        type            zeroGradient;
    }
seedbot_9
    {
        type            zeroGradient;
    }
front
    {
        type            empty;
    }
back
    {
        type            empty;
    }
}
// ***** //

```

D.2.1.5 R file

```

/*-----* 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      volTensorField;
    object     R;
}
// ***** //
dimensions    [0 2 -2 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;
    }
}

```

```

}
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

```

/*-----* 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    T;
}
// ***** //
dimensions [0 0 0 1 0 0 0];
internalField uniform 0;
boundaryField
{
  inlet
  {
    type      zeroGradient;
  }
  outlet
  {
    type      zeroGradient;
  }
  top
  {
    type      zeroGradient;
  }
  bottom
  {
    type      zeroGradient;
  }
  seedtop_0
  {
    type      fixedGradient;
    gradient  uniform 0;
  }
  seedbot_0
  {
    type      fixedGradient;
    gradient  uniform 0;
  }
  ....
  ....
  ....
  ....
  seedtop_9
  {
    type      fixedGradient;
    gradient  uniform 0;
  }
  seedbot_9
  {
    type      fixedGradient;
    gradient  uniform 0;
  }
  front
  {

```

```

        type          empty;
    }
    back
    {
        type          empty;
    }
}

```

D.2.1.7 Blood velocity file, U

```

/*-----* C++ -*-----*\
|=====|
| \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O p e r a t i o n | Version: 1.4 |
| \ \ / / A n d | Web: http://www.openfoam.org |
| \ \ / / M a n i p u l a t i o n |
\*-----*/

FoamFile
{
    version 2.0;
    format ascii;
    root "";
    case "";
    instance "";
    local "";
    class volVectorField;
    object U;
}
// ***** //
dimensions [0 1 -1 0 0 0];
internalField uniform (0 0 0);
boundaryField
{
    inlet
    {
        type parabolicVelocity;
        maxValue 1.5;
        n (1 0 0);
        y (0 1 0);
        value nonuniform List<vector>
96
    (
    (0.0217952 0 0)
    (0.0649072 0 0)
    (0.0649072 0 0)
    .
    .
    .
    (0.0649072 0 0)
    (0.0649072 0 0)
    (0.0217952 0 0)
    );
}
}

```

```

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

```

/*-----*\
|=====|
| \\    / Field | OpenFOAM: The Open Source CFD Toolbox |
| \\    / Operation | Version: 1.4 |

```

```
|  \ \ /  A nd      | Web:   http://www.openfoam.org      |
|  \ \ /  M anipulation |
\*-----*/
```

```
FoamFile
{
    version      2.0;
    format       ascii;
    root         "";
    case         "";
    instance     "";
    local        "";
    class        volVectorField;
    object       H;
}
// ***** //
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
    {
        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;
    }
  }

```

70

D.2.1.9 Uniform Field in the Model

```

/*-----*/
|=====| | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / Field | Version: 1.4 |
| \ \ / / Operation | Web: http://www.openfoam.org |
| \ \ / / And |
| \ \ / / Manipulation |
/*-----*/

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

```

10

20

30

40

```

{
  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;
}
}

```

50

60

70

D.2.1.10 Modification to the Magnetic Flux Density

```

/*-----*\
|=====|
| \ \ / / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O peration | Version: 1.4 |
| \ \ / / A nd | Web: http://www.openfoam.org |
| \ \ / / M anipulation | |
\*-----*/

```

```

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);

```

10

20

```

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

```

/*-----*\
|=====|
| \\    / Field | OpenFOAM: The Open Source CFD Toolbox |

```

```

|  \ \ / /  O peration   |  Version:  1.4   |  |
|  \ \ / /  A nd         |  Web:      http://www.openfoam.org |  |
|  \ \ / /  M anipulation |  |
|  *-----* /
FoamFile
{
  version      2.0;
  format       ascii;
  root         "";
  case         "";
  instance     "";
  local        "";
  class        volVectorField;
  object       M_aviles;
}
// *****
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
  {
    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.2.1.12 Magnetic Moment in the Model

```

/*-----*\
=====
| \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O p e r a t i o n | Version: 1.4 |
| \ \ / / A n d | Web: http://www.openfoam.org |
| \ \ / / M a n i p u l a t i o n | |
/*-----*/

```

```

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

  class        volVectorField;
  object       moment_Creg;
}
// *****
dimensions    [0 2 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
{
    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.2.2 constant

D.2.2.1 Polymesh File (blockMeshDict) file

```

// 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
//
/*-----*\
|=====|
| \ / \ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \ / \ / O peration | Version: 1.4 |
| \ / \ / A nd | Web: http://www.openfoam.org |
| \ / \ / M anipulation | |
/*-----*/

```

```

FoamFile
{

```

```

    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

```

/*-----*\
|=====|
| \ \ / / 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        dictionary;
    object       physicalProperties;
}
// ***** //
//                               kg m s K ml A cd                               //

```

```

theta      theta      [ 0 0 0 0 0 0 0]90.0;           // H_m field angle (in degrees)
B_0        B_0        [ 1 0 -2 0 0 -1 0]0.65;         // magnitude of applied B
mu_0       mu_0       [ 1 1 -2 0 0 -2 0]12.56637062e-07; // magnetic permability of free space
R_i        R_i        [ 0 1 0 0 0 0 0]62.5e-06; // implant radius
R_i_scale  R_i_scale  [ 0 0 0 0 0 0 0]62.5e-06; // implant radius
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]1261000; // Saturation magnetization of implant
M_fm_p_s   M_fm_p_s   [ 0 -1 0 0 0 1 0]351900; // Saturation magnetization of mat. in MDCP
moment_fm_p_s moment_fm_p_s [ 0 2 0 0 0 1 0]2.03e-19; // magnetic moment of the mat. in MDCP
rho_beta   rho_beta   [ 1 -3 0 0 0 0 0]1000;         // blood density
rho_fm_p   rho_fm_p   [ 1 -3 0 0 0 0 0]5050;         // density of in MDCP
rho_pol_p  rho_pol_p  [ 1 -3 0 0 0 0 0]950;           // density of pol. mat. in MDCP
x_fm_p     x_fm_p     [ 0 0 0 0 0 0 0]0.25;         // mass fraction of fer. mat. in MDCP
w_fm_p     w_fm_p     [ 0 0 0 0 0 0 0]0.064;         // volime fraction of fer. mat. in MDCP
R_p        R_p        [ 0 1 0 0 0 0 0]435.0e-9; // MDCP radius
R_pv       R_pv       [ 0 1 0 0 0 0 0]0.5e-3; // blood vessel radius
u_0        u_0        [ 0 1 -1 0 0 0 0]0.021; // average inlet velocity
eta_beta   eta_beta   [ 1 -1 -1 0 0 0 0]0.001; // blood viscosity (Pa s)
epsilon_p  epsilon_p  [ 0 0 0 0 0 0 0]0;           // porosity of the material
k_B        k_B        [ 1 2 -2 -1 0 0 0]1.38e-23; //porosity of the material
T_L        T_L        [ 0 0 0 1 0 0 0]300.0;

```

D.2.2.3 transport properties file

```

/*-----*\
|=====|
| \ \ / / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O peration | Version: 1.4 |
| \ \ / / A nd | Web: http://www.openfoam.org |
| \ \ / / M anipulation | |
\*-----*/
FoamFile
{
  version      2.0;
  format       ascii;
  root         "";
  case        "";
  instance     "";
  local       "";
  class        dictionary;
  object       transportProperties;
}
// *****

transportModel      Newtonian;
nu                  nu      [0 2 -1 0 0 0 0] 1e-06;
CrossPowerLawCoeffs
{
  nu0               nu0     [0 2 -1 0 0 0 0] 1e-06;
  nuInf             nuInf   [0 2 -1 0 0 0 0] 1e-06;
  m                 m       [0 0 1 0 0 0 0] 1;
  n                 n       [0 0 0 0 0 0 0] 1;
}
BirdCarreauCoeffs

```

```

{
    nu0          nu0    [0 2 -1 0 0 0 0] 1e-06;
    nuInf        nuInf  [0 2 -1 0 0 0 0] 1e-06;
    k            k      [0 0 1 0 0 0 0] 0;
    n            n      [0 0 0 0 0 0 0] 1;
}
// ***** //

```

D.2.2.4 turbulence properties file

```

/*-----*\
|=====|
| \ \ / / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O peration | Version: 1.4 |
| \ \ / / A nd | Web: http://www.openfoam.org |
| \ \ / / M anipulation |
/*-----*/

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

turbulenceModel laminar;
turbulence off;
laminarCoeffs
{
}

```

D.2.3 system

D.2.3.1 controlDict file

```

/*-----*\
|=====|
| \ \ / / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O peration | Version: 1.4 |
| \ \ / / A nd | Web: http://www.openfoam.org |
| \ \ / / M anipulation |
/*-----*/

FoamFile
{
    version          2.0;
    format           ascii;
    root             "";
}

```

```

    case      "";
    instance  "";
    local    "";
    class     dictionary;
    object    controlDict;
}
// ***** //
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

```

/*-----*\
|=====|
|  \ \ /  /  F ield   | OpenFOAM: The Open Source CFD Toolbox |
|  \ \ /  /  O peration | Version: 1.4                |
|  \ \ /  /  A nd       | Web:    http://www.openfoam.org   |
|  \ \ /  /  M anipulation |                               |
\*-----*/

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

ddtSchemes
{
    default steadyState;
}
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(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

```

/*-----*\
|=====|
|  \  /  /  Field      | OpenFOAM: The Open Source CFD Toolbox |
|  \  /  /  Operation   | Version: 1.4                      |
|  \  /  /  And         | Web:    http://www.openfoam.org     |
|  \  /  /  Manipulation|                                     |
\*-----*/

```

```

FoamFile
{
    version      2.0;
    format       ascii;
}

```



```

U          0.7;
k          0.7;
epsilon    0.7;
R          0.7;
nuTilda   0.7;
}
// ***** //

```

D.2.3.4 sampleDict file

```

/*-----*\
|=====|
| \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O peration | Version: 1.4 |
| \ \ / / A nd | Web: http://www.openfoam.org |
| \ \ / / M anipulation |
/*-----*/

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.2.3.5 *twoRegionDict* file

```

/*-----*/
|=====| | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / Field | | Version: 1.4 |
| \ \ / Operation | | Web: http://www.openfoam.org |
| \ \ / And | |
| \ \ / Manipulation |
/*-----*/
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

```

10

20

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:

- P. J. Cregg, K. Murphy and A. Mardinoglu, *Calculation of nanoparticle capture efficiency in magnetic drug targeting*, *Journal of Magnetism and Magnetic Materials*, **320**, 3272–3275, 2008. (Page 166)
- P. J. Cregg, K. Murphy and A. Mardinoglu, *Inclusion of magnetic dipole-dipole and hydrodynamic interactions in implant assisted magnetic drug targeting*, *Journal of Magnetism and Magnetic Materials*, **321**, 3893–3898, 2009. (Page 170)
- P. J. Cregg, K. Murphy, A. Mardinoglu and Adriele Prina-Mello, *Many particle magnetic dipole-dipole and hydrodynamic interactions in magnetisable stent assisted magnetic drug targeting*, Submitted to *Journal of Magnetism and Magnetic Materials*. (Page 176)
- P. J. Cregg, K. Murphy and A. Mardinoglu, *Inclusion of interparticle interactions in the modelling of magnetic drug targeting*, 6th International Scientific and Clinical Applications of Magnetic Carriers, May 17th – 20th, 2006, Krems, Austria. (Page 163)
- P. J. Cregg, K. Murphy and A. Mardinoglu, *Calculation of dipole interactions in magnetic drug targeting*, 7th International Scientific and Clinical Applications of Magnetic Carriers, May 21st – 24th, 2008, Vancouver, Canada. (Page 164)
- P. J. Cregg, K. Murphy and A. Mardinoglu, *Inclusion of magnetic dipole-dipole interaction in implant assisted magnetic drug targeting*, Joint European Magnetic Symposia (JEMS08), September 14th – 19th, 2008, Dublin, Ireland. (Page 165)



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: pjcregg@wit.ie, Phone:+353-51-302631, Fax:+353-51-302666

MCP

Abstract

The magnetic targeted drug delivery system of Ritter and co-workers, which uses ferromagnetic implants and high gradient magnetic separation principles to target within the body, is considered. In that model, collection (CE) and diversion efficiencies (DE) are defined and used to evaluate system performance. There, microparticles were considered and the benefit of particle agglomeration on CEs was assessed by considering the larger agglomerated particles. Berry has pointed out the biomedical advantages of nanoparticles (<50 nm). Here, we follow the model of Ritter *et al.* 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.

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 [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, 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] but investigate the CEs for superparamagnetic nanoparticles.
- We note the significant beneficial role that agglomeration near the wire might play in increasing the magnetic force and subsequent de-agglomeration [5].
- We take steps to model this for the mathematically simplest case of (short range) inter-particle exchange interaction.
- As the particles are small and undergo significant thermal fluctuations, we use the Langevin function to describe the magnetic moment of each particle.
- We also consider the limit of large exchange interaction, described by appropriate Langevin functions, which is likely to occur between small nanoparticles in contact or near contact [8,9].

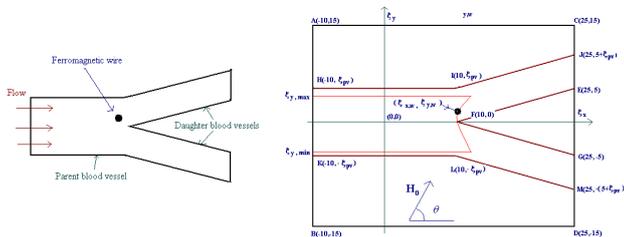


FIGURE 1: (a) Ferromagnetic wire placed asymmetrically at branch point in blood vessels. (b) The co-ordinate system of Ritter *et al.* The values $\xi_{y,max}$ and $\xi_{y,min}$ are significant in defining the efficiencies CE and DE.

Stokes Drag and Magnetic Force

Collection Efficiency (CE) is determined following Ritter by considering the behaviour of the magnetic nanoparticles under the influence of the two forces of Stokes drag and the magnetic force.

$$\mathbf{F}_s = 6\pi\eta_b R_p (\mathbf{v}_b - \mathbf{v}_p)$$

where η_b is the viscosity of the blood, R_p the radius of the particle, \mathbf{v}_b and \mathbf{v}_p , the velocities of the blood and the particle respectively. The blood velocity, \mathbf{v}_b , is determined by solving the appropriate Navier-Stokes equations.

$$\mathbf{F}_m = (\mathbf{m} \cdot \nabla) \mathbf{B}$$

where \mathbf{B} is the resulting magnetic flux density (due to external \mathbf{H} and the presence of the wire) and \mathbf{m} is the magnetic moment of the particle. Ritter *et al.* considered microparticles where the axis of the moment \mathbf{m} lay along that of \mathbf{B} , and the magnetisation increased linearly with H until saturated. 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].

$$\mathbf{m} = \omega_{fm,p} V_p M_s L(\beta) \frac{\mathbf{B}}{|\mathbf{B}|}$$

where

$$L(\beta) = \coth(\beta) - \frac{1}{\beta} \quad \text{and} \quad \beta = \frac{\mu_0 \omega_{fm,p} V_p M_s H}{kT}$$

Here $\omega_{fm,p}$ 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 \mathbf{B} field is calculated from solving the Laplacian as indicated by Ritter, ensuring continuity of flux and potential, across the wire-blood interface.

Collection Efficiency (CE)

Particles, which in equilibrium under the influence of these forces, are found to have the coordinates within the region indicated by the min and max y points as in Figure 1(b) and as in the following equation

$$CE = \frac{\xi_{y,max} - \xi_{y,min}}{2\xi_{pw}} \times 100\%$$

are deemed to have been collected.

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 Z 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 $\langle \mathbf{m} \rangle$ 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

$$M_{R,0}(\beta) = 2L(\beta)$$

Whereas, two identical particles in the limit of very large exchange interaction ($J \rightarrow \infty$) behave as one moment with twice the magnitude *viz.*

$$M_{R,J \rightarrow \infty}(\beta) = 2L(2\beta)$$

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 $\beta_{ex} = J(V_p M_s)^2 / kT = 5$.

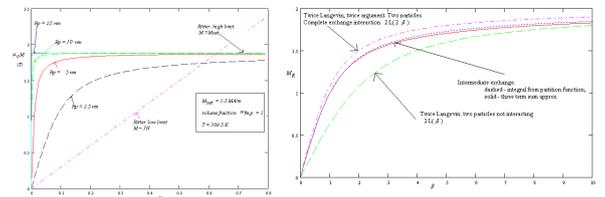


FIGURE 2: (a) Magnetisation $\mu_0 M_{fm,p}$ per unit volume as calculated by Ritter, and by the Langevin function. (b) Reduced magnetisation for two identical nanoparticles against Langevin parameter $\beta_2 = \mu_0 V_p M_s H_0 / (kT)$ for external field \mathbf{H}_0 . — non interacting, — limit of infinite exchange interaction, — finite exchange $\beta_{ex} = 5$.

Conclusions

- The per 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

- [1] Q.A. Pankhurst, J. Connolly, S.K. Jones, J. Dobson, *J. Phys. D: Appl. Phys.* 36 (2003) R167.
- [2] A.D. Grief, G. Richardson, *J. Magn. Magn. Mater.*, 293 (2005) 455.
- [3] Gh. Iacob, O. Rotariu, N.J.C. Strachan, U.O. Häfeli, *Biorheology*, 41 (2004) 599.
- [4] J.A. Ritter, A.D. Ebner, K.D. Daniel, K.L. Stewart, *J. Magn. Magn. Mater.*, 280 (2004) 184.
- [5] M.O. Aviles, A.D. Ebner, H. Chen, A.J. Rosengart, M.D. Kaminski, J.A. Ritter, *J. Magn. Magn. Mater.*, 293 (2005) 605.
- [6] H. Chen, A.D. Ebner, M.D. Kaminski, A.J. Rosengart, J.A. Ritter, *J. Magn. Magn. Mater.*, 293 (2005) 616.
- [7] C.C. Berry, A.S.G. Curtis, *J. Phys. D: Appl. Phys.*, 36 (2003) R198.
- [8] Denis G. Rancourt, *Magnetism of Earth, Planetary and Environmental Nanomaterials*, in: *Nanoparticles in the Environment* (Banfield, J.F. and A. Navrotsky, eds.) Rev. Mineral. Geochem., 44 (2001) 217.
- [9] D. Kechrakos, K.N. Trohidou, *J. Magn. Magn. Mater.*, 262 (2003) 107.
- [10] M.I. Shliomis, *Sov. Phys. Usp.*, 17 (1974) 153.



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 Group, SEAM Centre, Waterford Institute of Technology, Ireland.

E-mail: pjcregg@wit.ie, Phone:+353-51-302631, Fax:+353-51-302666

MCP

Abstract

The magnetic targeted drug delivery system of Aviles, Ebner 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 here 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 a 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 [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 [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 [4-6,11] 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.

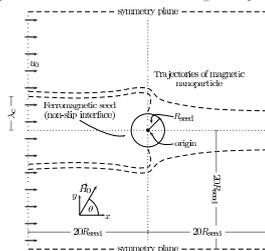


FIGURE 1: Schematic diagram of the control volume, CV, used in determining the capture radius, λ_c , of the magnetic nanoparticles.

Stokes Drag, Magnetic Force and Magnetic (Dipole) Interaction Force

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:

$$F_s = 6\pi\eta_b R_p (\mathbf{v}_b - \mathbf{v}_p)$$

where η_b is the viscosity of the blood, R_p the radius of the particle, \mathbf{v}_b and \mathbf{v}_p , the velocities of the blood and the particle respectively. The blood velocity, \mathbf{v}_b , is determined by solving the appropriate Navier-Stokes equations.

Magnetic force:

$$F_m = (\mathbf{m} \cdot \nabla) \mathbf{B}$$

where \mathbf{B} is the resulting magnetic flux density (due to external \mathbf{H}_0 magnetic field and the presence of the implant (seed)) and \mathbf{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 \mathbf{B} can be calculated from the Langevin function [2,15,16].

$$L(\beta) = \coth(\beta) - \frac{1}{\beta}, \quad (1)$$

with Langevin argument

$$\beta = \frac{\omega_{fm,p} V_p M_{fm,p,s} B}{kT}, \quad (2)$$

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

$$\mathbf{m} = \omega_{fm,p} V_p M_{fm,p,s} L(\beta) \frac{\mathbf{B}}{B}. \quad (3)$$

For calculation of \mathbf{B} field, Laplace's equation for the scalar potential is solved analytically by separation of variables [13,14].

Magnetic (Dipole) Interaction Force:

Including the magnetic effect of particle 2 on particle 1, the magnetic force can be augmented to

$$F_{int} = ((\mathbf{m}_1 + d\mathbf{m}_1) \cdot \nabla) (\mathbf{B} + d\mathbf{B}_2)$$

where $d\mathbf{B}_2$ is the modification of the resulting magnetic flux density due to particle 2 at \mathbf{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

$$d\mathbf{B}_2(\mathbf{r}) = \frac{1}{3} \left(\mu_0 M_{fm,p,s} \frac{L(\beta)}{B} \right) \frac{a^3}{|\mathbf{r}_1 - \mathbf{r}_2|^3} \left(3 \frac{(\mathbf{B}(\mathbf{r}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2))}{|\mathbf{r}_1 - \mathbf{r}_2|^2} (\mathbf{r}_1 - \mathbf{r}_2) - \mathbf{B}(\mathbf{r}_2) \right)$$

where \mathbf{r}_1 and \mathbf{r}_2 are the positions of the particles and $d\mathbf{m}_1$ is the change in the magnetic moment of the particle 1, given by

$$d\mathbf{m}_1 = \frac{1}{3} \left(\mu_0 M_{fm,p,s} \frac{L(\beta)}{B} \right)^2 \frac{a^3}{|\mathbf{r}_1 - \mathbf{r}_2|^3} \left(3 \frac{(\mathbf{B}(\mathbf{r}_2) \cdot (\mathbf{r}_1 - \mathbf{r}_2))}{|\mathbf{r}_1 - \mathbf{r}_2|^2} (\mathbf{r}_1 - \mathbf{r}_2) - \mathbf{B}(\mathbf{r}_2) \right)$$

Capture Cross Section (CCS)

The particle trajectories are obtained from evaluating the streamline function

$$\frac{\partial \psi}{\partial y} = -v_{p,x}$$

and

$$\frac{\partial \psi}{\partial x} = v_{p,y}$$

where ψ is the stream function, and $v_{p,x}$ and $v_{p,y}$ are the components of particle velocity \mathbf{v}_p . The system performance of this model is calculated in terms of the capture cross section, λ_c , defined as (see Fig.1)

$$\lambda_c = \frac{y_c}{R_{seed}},$$

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.

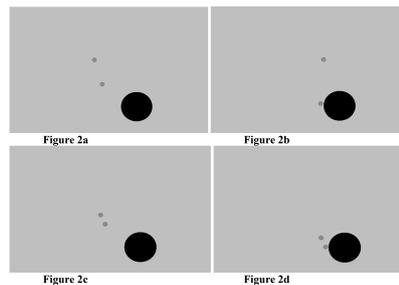


FIGURE 2: Figures 2a and 2b are the output of the OpenFOAM program and show the particle behaviour without dipole interactions. Figures 2c and 2d show the effect of the dipole interaction. Iron (50 nm particle radius) for the magnetic carrier particles and SS 409 (1000 nm seed radius) as the seed ferromagnetic materials is considered under high magnetic field.

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

- [1] Q.A. Pankhurst, J. Connolly, S.K. Jones, J. Dobson, *J. Phys. D: Appl. Phys.* 36 (2003) R167.
- [2] A.D. Grief, G. Richardson, *J. Magn. Magn. Mater.*, 293 (2005) 455.
- [3] Gh. Iacob, O. Rotariu, N.J.C. Strachan, U.O. Häfeli, *Biorheology*, 41 (2004) 599.
- [4] J.A. Ritter, A.D. Ebner, K.D. Daniel, K.L. Stewart, *J. Magn. Magn. Mater.*, 280 (2004) 184.
- [5] M.O. Aviles, A.D. Ebner, H. Chen, A.J. Rosengart, M.D. Kaminski, J.A. Ritter, *J. Magn. Magn. Mater.*, 293 (2005) 605.
- [6] H. Chen, A.D. Ebner, M.D. Kaminski, A.J. Rosengart, J.A. Ritter, *J. Magn. Magn. Mater.*, 293 (2005) 616.
- [7] C.C. Berry, A.S.G. Curtis, *J. Phys. D: Appl. Phys.*, 36 (2003) R198.
- [8] Denis G. Rancourt, *Magnetism of Earth, Planetary and Environmental Nanomaterials*, in: *Nanoparticles in the Environment* (Banfield, J.F. and A. Navrotsky, eds.) Rev. Mineral. Geochem., 44 (2001) 217.
- [9] D. Kechrakos, K.N. Trohidou, *J. Magn. Magn. Mater.*, 262 (2003) 107.
- [10] M.I. Shliomis, *Sov. Phys. Usp.*, 17 (1974) 153.
- [11] M.O. Aviles, A.D. Ebner, J.A. Ritter, *J. Magn. Magn. Mater.*, 310 (2007) 131.
- [12] C. Mikkelsen, M. F. Hansen, H. Bruus, *J. Magn. Magn. Mater.*, 293 (2005) 578.
- [13] K. J. Binns and P. J. Lawrenson and C. W. Trowbridge, *The Analytical and Numerical Solution of Electric and Magnetic Fields*, 1992, Wiley, Sec 3.2.2.
- [14] P. J. Cregg, K. Murphy, A. Mardinoglu, *Submitted to J. Magn. Magn. Mater.*, (2008).
- [15] P. J. Cregg and L. Bessais, *Calculation of nanoparticle capture efficiency in magnetic drug targeting J. Magn. Magn. Mater.*, 202 (1999) 554.
- [16] H. C. Bryant and D. A. Sergatskov and D. Lovato and N. L. Adolph and R. S. Larson and E. R. Flynn, *Phys. Med. Biol.*, 52 (2007) 4009.



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 conference organisers for free registration.

7th International Conference on the Scientific and Clinical Applications of Magnetic Carriers
May 21st – 24th 2008, Vancouver, Canada.



INCLUSION OF MAGNETIC DIPOLE-DIPOLE INTERACTION IN IMPLANT ASSISTED MAGNETIC DRUG TARGETING

P. J. Cregg*, Kieran Murphy & Adil Mardinoglu

Materials Characterisation & Processing Group, SEAM Centre, Waterford Institute of Technology, Ireland.

E-mail: pjcregg@wit.ie, Phone:+353-51-302631, Fax:+353-51-302666

MCP

Abstract

The magnetic targeted drug delivery system of Aviles, Ebner and Ritter [8], which uses SS 409 as the seed ferromagnetic material and iron for the magnetic drug carrier particles, is considered. Here in order to model agglomeration of particles is known to occur in this system and the magnetic dipole-dipole interaction is included. Interactions were calculated previously by Mikkelsen *et al.* [9] under low magnetic fields (0.05 T) and dipole-dipole interaction of two particles by using a higher magnetic field (0.7 T) is calculated in our submitted paper [11]. Here, dipole-dipole interaction of N multiple 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. Different initial positions are considered and the system performance is assessed in terms capture cross section [8]. In the simulations agglomeration is seen to occur leading to larger and smaller capture cross section according to the positions of the particles.

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 [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 [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].
- Mikkelsen *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 three nanoparticles with arbitrary initial positions.

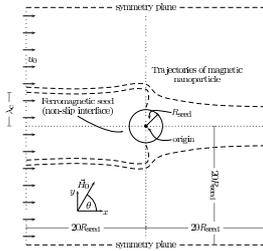


FIGURE 1: Schematic diagram of the control volume, CV, used in determining the capture radius, λ_c , of the magnetic nanoparticles.

Stokes Drag, Magnetic Force and Magnetic (Dipole-Dipole) Interaction Force

Capture Cross Section (CCS) is determined following Aviles *et al.* [8] by considering the behaviour of the magnetic nanoparticles under the influence of Stokes drag and the magnetic forces.

Stokes drag:
$$\mathbf{F}_s = 6\pi\eta_b R_p (\mathbf{v}_b - \mathbf{v}_p)$$

where η_b is the viscosity of the blood, R_p the radius of the particle, \mathbf{v}_b and \mathbf{v}_p , the velocities of the blood and the particle respectively. The blood velocity, \mathbf{v}_b , is determined by solving the appropriate Navier-Stokes equations.

Magnetic force:
$$\mathbf{F}_{in} = (\mathbf{m} \cdot \nabla) \mathbf{B}_{total}$$

where \mathbf{B}_{total} is the total magnetic flux density (due to external \mathbf{H}_0 magnetic field, the presence of the implant (seed) and the presence of the magnetic nanoparticles) and \mathbf{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 \mathbf{B} can be calculated from the Langevin function [2,12,13].

$$L(\beta) = \coth(\beta) - \frac{1}{\beta}$$

with Langevin argument

$$\beta = \frac{\omega_{fm,p} V_p M_{fm,p,s} B_{total}}{kT}$$

where $\omega_{fm,p}$ is the volume fraction of ferromagnetic material in the particle, V_p is the particle volume, $M_{fm,p,s}$ the (volume) saturation magnetisation, B_{total} is the magnitude of \mathbf{B}_{total} , k is Boltzmann's constant and T is the absolute temperature, so that the average projection of \mathbf{m} the magnetic moment of a nanoparticle in the direction of \mathbf{B}_{total} can be written as

$$\mathbf{m} = \omega_{fm,p} V_p M_{fm,p,s} L(\beta) \frac{\mathbf{B}_{total}}{B_{total}}$$

For calculation of \mathbf{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

$$(\mathbf{F}_{int})_n = (\mathbf{m}_n \cdot \nabla) (\mathbf{B}_{total})_n$$

where $(\mathbf{B}_{total})_n$ is the total magnetic flux acting on particle n (for particle n , $(\mathbf{B}_{total})_n$ is taken as $(\mathbf{B}_{total})_n = \mathbf{B} + d\mathbf{B}_1 + \dots + d\mathbf{B}_{(n-1)} + d\mathbf{B}_{(n+1)} + \dots + d\mathbf{B}_N$) and $d\mathbf{B}_n$ is the modification of the resulting magnetic flux density due to particle n at \mathbf{r} . 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

$$d\mathbf{B}_n(\mathbf{r}) = \frac{1}{3} \left(\mu_0 M_{fm,p,s} \frac{L(\beta)}{B} \right) \frac{a^3}{|\mathbf{r} - \mathbf{r}_n|^3} \left(3 \frac{(\mathbf{B}(\mathbf{r}_n) \cdot (\mathbf{r} - \mathbf{r}_n))}{|\mathbf{r} - \mathbf{r}_n|^2} (\mathbf{r} - \mathbf{r}_n) - \mathbf{B}(\mathbf{r}_n) \right)$$

where \mathbf{r} represents an arbitrary point in space and \mathbf{r}_n is the position of the particle n . The reader should note that it is the *gradient* of the flux density (including the above modification) which is required for the calculation of the magnetic force.

Capture Cross Section (CCS)

The particle trajectories are obtained from evaluating the streamline function

$$\frac{\partial \psi}{\partial y} = -v_{p,x}$$

and

$$\frac{\partial \psi}{\partial x} = v_{p,y}$$

where ψ is the stream function, and $v_{p,x}$ and $v_{p,y}$ are the components of particle velocity \mathbf{v}_p . The system performance of this model is calculated in terms of the capture cross section, λ_c , defined as (see Fig 1)

$$\lambda_c = \frac{y_c}{R_{seed}}$$

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.

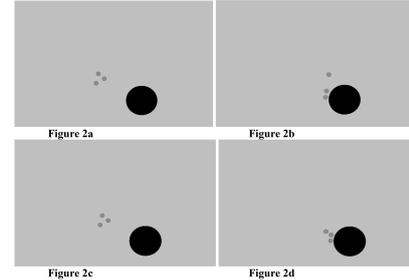


FIGURE 2: Figures 2a and 2b are the output of the OpenFOAM program and show the particle behaviour without dipole interactions. Figures 2c and 2d show the effect of the dipole interaction. Iron (50 nm particle radius) for the magnetic carrier particles and SS 409 (1000 nm seed radius) as the seed ferromagnetic materials is considered under high magnetic field.

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 Aviles *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.

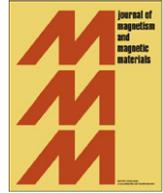
References

- [1] Q.A. Pankhurst, J. Connolly, S.K. Jones, J. Dobson, *J. Phys. D: Appl. Phys.* 36 (2003) R167.
- [2] A.D. Grief, G. Richardson, *J. Magn. Magn. Mater.*, 293 (2005) 455.
- [3] Gh. Iacob, O. Rotariu, N.J.C. Strachan, U.O. Häfeli, *Biorheology*, 41 (2004) 599.
- [4] J.A. Ritter, A.D. Ebner, K.D. Daniel, K.L. Stewart, *J. Magn. Magn. Mater.*, 280 (2004) 184.
- [5] M.O. Aviles, A.D. Ebner, H. Chen, A.J. Rosengart, M.D. Kaminski, J.A. Ritter, *J. Magn. Magn. Mater.*, 293 (2005) 605.
- [6] H. Chen, A.D. Ebner, M.D. Kaminski, A.J. Rosengart, J.A. Ritter, *J. Magn. Magn. Mater.*, 293 (2005) 616.
- [7] C.C. Berry, A.S.G. Curtis, *J. Phys. D: Appl. Phys.*, 36 (2003) R198.
- [8] M.O. Aviles, A.D. Ebner, J.A. Ritter, *J. Magn. Magn. Mater.*, 310 (2007) 131.
- [9] C. Mikkelsen, M. F. Hansen, H. Bruus, *J. Magn. Magn. Mater.*, 293 (2005) 578.
- [10] P. J. Cregg, K. Murphy, A. Mardinoglu, *Calculation of nanoparticle capture efficiency in magnetic drug targeting*, *J. Magn. Magn. Mater.*, doi:10.1016/j.jmmm.2008.06.21.
- [11] P. J. Cregg, K. Murphy, A. Mardinoglu, *Inclusion of magnetic dipole-dipole interaction in implant assisted magnetic drug targeting*, Submitted to *J. Magn. Magn. Mater.*, (2008).
- [12] P. J. Cregg and L. Bessais, *J. Magn. Magn. Mater.*, 202 (1999) 554.
- [13] H. C. Bryant and D. A. Sergatskov and D. Lovato and N. L. Adolph and R. S. Larson and E. R. Flynn, *Phys. Med. Biol.*, 52 (2007) 4009.



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.

Joint European Magnetic Symposia (JEMS08), September 14th – 19th, 2008, Dublin, Ireland.



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

ARTICLE INFO

Article history:

Received 17 December 2007

Received in revised form

12 May 2008

Available online 24 June 2008

PACS:

47.63.mh

47.63.-b

87.85.gf

Keywords:

Magnetic drug targeting

High gradient magnetic separation (HGMS)

Magnetic nanoparticle

Simulation

Magnetic seed

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 ≈ 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 μm , 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

$$\vec{F}_s = 6\pi\eta_b R_p (\vec{v}_b - \vec{v}_p), \quad (1)$$

where η_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

$$\vec{F}_m = (\vec{m} \cdot \nabla) \vec{B}, \quad (2)$$

where \vec{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}_{seed} ,

* Corresponding author. Tel.: +353 51 302631.
E-mail address: pjcregg@wit.ie (P.J. Cregg).

forces acting upon it. Hence we obtain

$$\vec{v}_p = \vec{v}_b + \frac{1}{2} v_m \frac{R_{seed}}{M_{seed,s} H} \nabla(\vec{H} \cdot \vec{H}). \quad (14)$$

Here \vec{H} is the total magnetic field at the location of the magnetic drug carrier particle. The field \vec{H} relates to \vec{B} via

$$\vec{B} = \mu_0 \vec{H}, \quad (15)$$

and in the space around the seed is given by, from Eq. (11),

$$\vec{H} = \vec{H}_0 - \nabla\phi. \quad (16)$$

Therefore, the magnitude of the total magnetic field is

$$H = \sqrt{\left(H_0 \cos \theta - \frac{\partial\phi}{\partial x}\right)^2 + \left(H_0 \sin \theta - \frac{\partial\phi}{\partial y}\right)^2}, \quad (17)$$

and v_m , the magnitude of the magnetic velocity, is given by

$$v_m = \frac{2}{9} \frac{R_p^2}{R_{seed}} \frac{\mu_0}{\eta_b} \omega_{fm,p} M_{seed,s} M_{fm,p,s} L(\beta). \quad (18)$$

The volume fraction of ferromagnetic material $\omega_{fm,p}$ in the magnetic drug carrier particle is related to its weight fraction $x_{fm,p}$ through [5]

$$\omega_{fm,p} = \frac{x_{fm,p}}{x_{fm,p} + (1 - x_{fm,p}) \rho_{fm,p} / \rho_{pol,p}}, \quad (19)$$

where $\rho_{fm,p}$ is the density of the ferromagnetic material in the magnetic drug carrier particle and $\rho_{pol,p}$ is the density of the polymer material in the magnetic drug carrier particle.

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

$$\frac{\partial\psi}{\partial y} = -v_{p,x}, \quad (20)$$

$$\frac{\partial\psi}{\partial x} = v_{p,y}, \quad (21)$$

where ψ is the stream function, and $v_{p,x}$ and $v_{p,y}$ are the components of \vec{v}_p from Eq. (14). The system performance of this model is calculated in terms of the capture cross section, λ_c ,

defined as

$$\lambda_c = \frac{y_c}{R_{seed}}, \quad (22)$$

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 $1 \mu\text{m}$ radius. The streamline functions for the capture of nanoparticles are presented in Fig. 2 for particle radius $R_p = 50 \text{ nm}$, containing 40 wt% iron ($x_{fm,p} = 0.4$), 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, λ_c , is

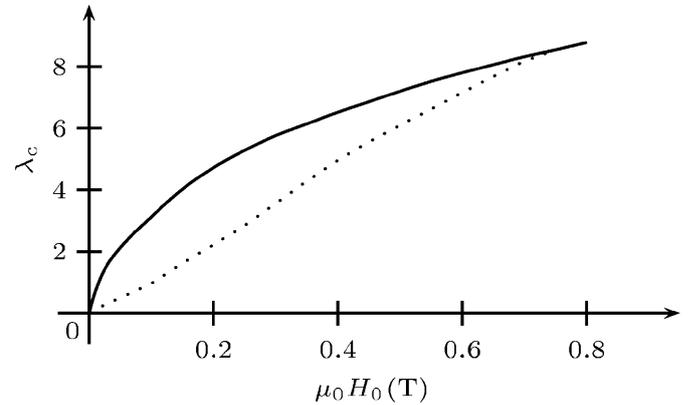


Fig. 3. Capture cross section, λ_c , plotted as a function of the applied magnetic field strength, $\mu_0 H_0$, calculated using (—) the Langevin function as appropriate for single domain particles and (.....) without Langevin function as appropriate for multiple domain particles.

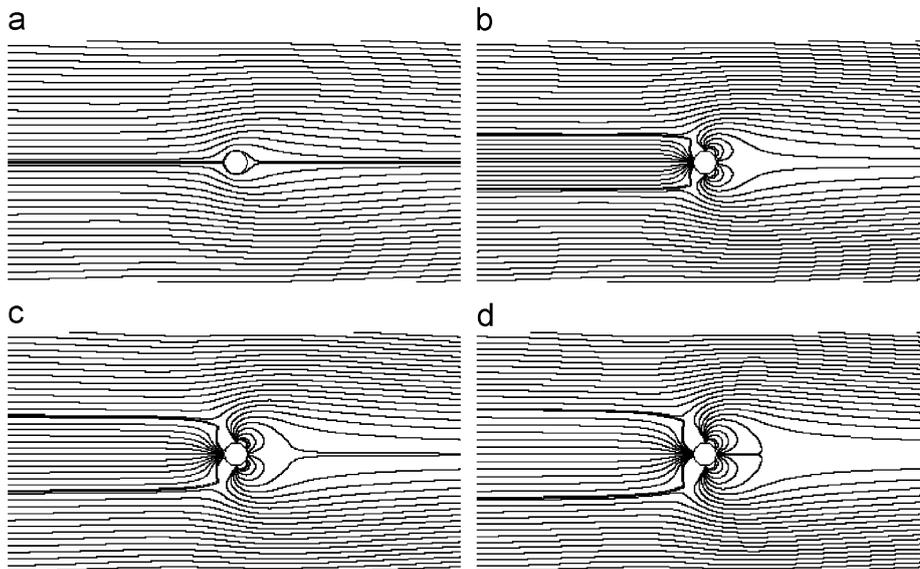


Fig. 2. Streamlines indicating the trajectories of the single domain nanoparticles, calculated using OpenFOAM, as they traverse the control volume for different magnitudes of the externally applied magnetic field, H_0 . (a) $\mu_0 H_0 = 0.0 \text{ T}$; (b) $\mu_0 H_0 = 0.2 \text{ T}$; (c) $\mu_0 H_0 = 0.4 \text{ T}$; (d) $\mu_0 H_0 = 0.6 \text{ T}$.

Table 1
Values of system and material parameters used in the simulation

Property	Value	SI Unit
ρ_b	1040.0	kg m^{-3}
η_b	0.002	$\text{kg m}^{-1} \text{s}^{-1}$
u_0	0.001	m s^{-1}
$\mu_0 H_0$	0.0–0.8	$\text{kg s}^{-2} \text{A}^{-1}$
$\chi_{\text{fm,p}}$	0.4	
$\rho_{\text{fm,p}}$	7850	kg m^{-3}
$\chi_{\text{seed,0}}$	1000	
$M_{\text{seed,s}}$	1397000	A m^{-1}
$M_{\text{fm,p,s}}$	1735000	A m^{-1}
R_{seed}	1.0×10^{-6}	m
R_p	50×10^{-9}	m
$\rho_{\text{pol,p}}$	950	kg m^{-3}

calculated and presented in Fig. 3 for 50 nm particles, as a function of the magnetic field strength $\mu_0 H_0$. 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 Avilés et al. Beyond a field of ≈ 0.7 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.

7. Conclusions

The model of Avilés, 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 Avilés for helpful conversations at Krems.

References

- [1] Q.A. Pankhurst, J. Connolly, S.K. Jones, J. Dobson, *J. Phys. D: Appl. Phys.* 36 (2003) R167.
- [2] C.C. Berry, A.S.G. Curtis, *J. Phys. D: Appl. Phys.* 36 (2003) R198.
- [3] A.D. Grief, G. Richardson, *J. Magn. Magn. Mater.* 293 (1) (2005) 455.
- [4] G. Jacob, O. Rotariu, N.J.C. Strachan, U.O. Häfeli, *Biorheology* 41 (2004) 599.
- [5] J.A. Ritter, A.D. Ebner, K.D. Daniel, K.L. Stewart, *J. Magn. Magn. Mater.* 280 (2–3) (2004) 184.
- [6] M.O. Avilés, A.D. Ebner, H. Chen, A.J. Rosengart, M.D. Kaminski, J.A. Ritter, *J. Magn. Magn. Mater.* 293 (1) (2005) 605.
- [7] H. Chen, A.D. Ebner, M.D. Kaminski, A.J. Rosengart, J.A. Ritter, *J. Magn. Magn. Mater.* 293 (1) (2005) 616.
- [8] B.B. Yellen, Z.G. Forbes, D.S. Halverson, G. Fridman, K.A. Barbee, M. Chorny, R. Levy, G. Friedman, *J. Magn. Magn. Mater.* 293 (1) (2005) 647.
- [9] E.J. Furlani, E.P. Furlani, *J. Magn. Magn. Mater.* 312 (1) (2007) 187.
- [10] M.O. Avilés, A.D. Ebner, J.A. Ritter, *J. Magn. Magn. Mater.* 310 (1) (2007) 131.
- [11] P.J. Cregg, L. Bessais, *J. Magn. Magn. Mater.* 202 (2–3) (1999) 554.
- [12] M.I. Shliomis, *Sov. Phys. Usp.* 17 (3) (1974) 153.
- [13] H.C. Bryant, D.A. Sergatskov, D. Lovato, N.L. Adolphi, R.S. Larson, E.R. Flynn, *Phys. Med. Biol.* 52 (14) (2007) 4009.
- [14] K.J. Binns, P.J. Lawrenson, C.W. Trowbridge, *The Analytical and Numerical Solution of Electric and Magnetic Fields*, Wiley, New York, 1992 Section 3.2.2.
- [15] OpenCFD Ltd, OpenFOAM 1.4 (<http://www.opencfd.co.uk>) (2007).



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

ARTICLE INFO

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

ABSTRACT

Mathematical modelling of the implant-assisted magnetic drug targeting system of Avilés, 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 Avilés 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

$$\vec{F}_{s_1} = 6\pi\eta_b R_p (\vec{v}_b - \vec{v}_{p_1}) \quad (1)$$

where η_b is the viscosity of the blood, R_p the radius of the nanoparticle, \vec{v}_b and \vec{v}_{p_1} the velocities of the blood and Particle 1, respectively. A similar expression applies for Particle 2. The blood velocity, \vec{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],

$$\vec{F}_{\text{hyd}_1} = -\frac{9\pi\eta_b R_p^2}{2|\vec{r}_1 - \vec{r}_2|} \left[\mathbf{1} + \frac{(\vec{r}_1 - \vec{r}_2) \otimes (\vec{r}_1 - \vec{r}_2)}{|\vec{r}_1 - \vec{r}_2|^2} \right] \cdot (\vec{v}_b - \vec{v}_{p_2}) \quad (2)$$

where $\mathbf{1}$ is the unit tensor, \otimes the vector tensor product (outer product), \vec{r}_1 and \vec{r}_2 the positions of Particle 1 and Particle 2, respectively and \vec{v}_{p_2} 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).

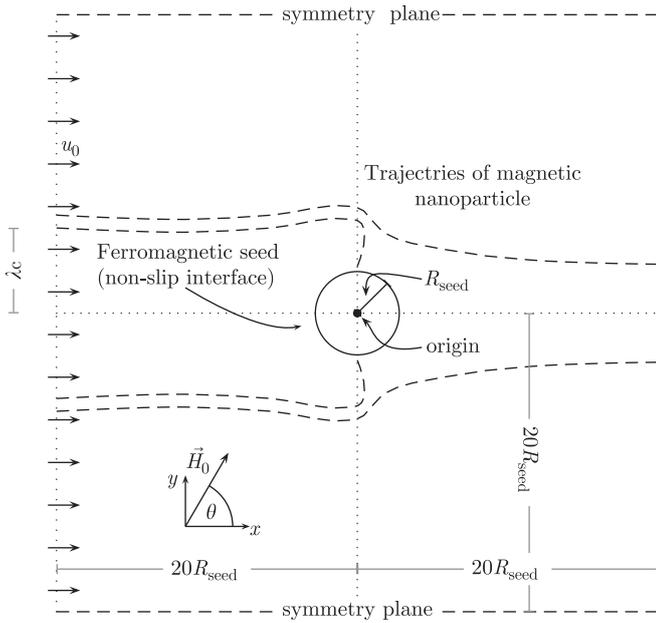


Fig. 1. Schematic diagram of the control volume, used in determining the capture cross section, λ_c , of the magnetic nanoparticles.

In general the magnetic force acting on a magnetic moment is determined by

$$\vec{F}_m = (\vec{m} \cdot \nabla) \vec{B}_{\text{total}} \quad (3)$$

where \vec{m} is the magnetic moment and \vec{B}_{total} is the total magnetic flux density. Two 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. Thus for Particle 1 we can write

$$\vec{F}_{\text{int}_1} = (\vec{m}_1 \cdot \nabla) (\vec{B} + d\vec{B}_2) \quad (4)$$

where \vec{m}_1 is the total magnetic moment of Particle 1, \vec{B} the magnetic flux density due to the external field and the seed and $d\vec{B}_2$ the modification of the resulting magnetic flux density at Particle 1 at \vec{r}_1 due to the presence of Particle 2 at \vec{r}_2 . (An equivalent formula applies for Particle 2 with \vec{m}_2 and $\vec{B} + d\vec{B}_1$.) The value of \vec{B} required to calculate the magnetic force as given by Eqs. (4) and (12), is calculated from the scalar magnetic potential due to the seed, which satisfies the Laplace equation over two con-jointed regions: inside the seed and outside the seed. Thus for outside the seed we have magnetic flux density given by

$$\vec{B} = \mu_0 (\vec{H}_0 - \nabla \phi) \quad (5)$$

where μ_0 is the magnetic permeability of free space, and \vec{H}_0 is the external magnetic field. Taking the seed centre as the origin, as in Fig. 1, the reduced scalar potential, ϕ in the region outside the seed is given by [13,16]

$$\phi = H_0 R_{\text{seed}}^2 \alpha_{\text{seed}} \frac{x \cos \theta + y \sin \theta}{x^2 + y^2} \quad (6)$$

where H_0 is the magnitude of the applied field, R_{seed} radius of the seed and α_{seed} the demagnetising factor which for an infinitely long cylinder in a perpendicular field can be taken as [9]

$$\alpha_{\text{seed}} = \min \left(\frac{\chi_{\text{seed},0}}{2 + \chi_{\text{seed},0}}, \frac{M_{\text{seed},s}}{2H_0} \right) \quad (7)$$

where $\chi_{\text{seed},0}$ and $M_{\text{seed},s}$ are the zero field susceptibility and saturation magnetisation of the seed, respectively, and \vec{H}_0 can be

written

$$\vec{H}_0 = \begin{pmatrix} H_0 \cos \theta \\ H_0 \sin \theta \end{pmatrix} \quad (8)$$

where θ is the angle of the field from the positive x -axis in Fig. 1.

The average projection of \vec{m} the magnetic moment of a nanoparticle in the direction of \vec{B}_{total} (which for Particle 1 is taken as $\vec{B} + d\vec{B}_2$ and equivalently for Particle 2 would be $\vec{B} + d\vec{B}_1$) can be calculated from the Langevin function [3,11,13,17–19]

$$L(\beta) = \coth(\beta) - \frac{1}{\beta} \quad (9)$$

with argument

$$\beta = \frac{\omega_p V_p M_{p,s} B_{\text{total}}}{kT} \quad (10)$$

where V_p is the nanoparticle volume, B_{total} the magnitude of \vec{B}_{total} , k Boltzmann's constant, T the absolute temperature, $M_{p,s}$ the saturation magnetisation of the nanoparticle, and ω_p the volume fraction of ferromagnetic material in the nanoparticle which relates to its weight fraction x_p through [6]

$$\omega_p = \frac{x_p}{x_p + (1 - x_p)(\rho_p / \rho_{\text{pol},p})} \quad (11)$$

where ρ_p is the density of the ferromagnetic material and $\rho_{\text{pol},p}$ is the density of the polymer material in the nanoparticle. Thus \vec{m} (for either particle) can be written

$$\vec{m} = \omega_p V_p M_{p,s} L(\beta) \frac{\vec{B}_{\text{total}}}{B_{\text{total}}} \quad (12)$$

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]

$$d\vec{B}_2(\vec{r}) = \frac{1}{3} \left(\mu_0 M_{p,s} \frac{L(\omega_p V_p M_{p,s} |\vec{B}(\vec{r}_2)|)}{|\vec{B}(\vec{r}_2)|} \right) \frac{R_p^3}{|\vec{r} - \vec{r}_2|^3} \times \left(\frac{3(\vec{B}(\vec{r}_2) \cdot (\vec{r} - \vec{r}_2))}{|\vec{r} - \vec{r}_2|^2} (\vec{r} - \vec{r}_2) - \vec{B}(\vec{r}_2) \right) \quad (13)$$

where \vec{r} represents an arbitrary point in space and $\vec{B}(\vec{r}_2)$ is flux density at \vec{r}_2 due to the external field and seed. The reader should note that it is the *gradient* of this field, $d\vec{B}_2$, at \vec{r}_1 (and equivalently at \vec{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

$$\nabla \cdot \vec{v}_b = 0 \quad (14)$$

and the Navier–Stokes equation

$$\rho_b [(\vec{v}_b \cdot \nabla) \vec{v}_b] = -\nabla P + \eta_b \nabla^2 \vec{v}_b \quad (15)$$

where ρ_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

$$\vec{v}_b = \begin{pmatrix} u_0 \\ 0 \end{pmatrix} \quad (16)$$

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

$$\vec{v}_{p1} = \vec{v}_b + \frac{\vec{F}_{int1} + \vec{F}_{hyd1}}{6\pi\eta_b R_p} \quad (17)$$

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, λ_c , defined as

$$\lambda_c = \frac{y_c}{R_{seed}} \quad (18)$$

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 $1\ \mu\text{m}$. Results are presented by generating streamlines for two identical iron nanoparticles with radius $R_p = 50\ \text{nm}$, containing 40 wt% iron ($x_p = 0.4$), under the influence of homogeneous magnetic field oriented parallel to the flow ($\theta = 0$) with magnitude $\mu_0 H_0 = 0.7\ \text{T}$. 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, λ_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_{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 1
Values of system and material parameters used in the simulation.

Property	Value	SI Unit
ρ_b	1040	kg/m ³
η_b	0.002	kg/ms
U_0	0.001	m/s
$\mu_0 H_0$	0.7	Tesla (T)
x_p	0.4	–
ρ_p	7850	kg/m ³
$\chi_{seed,0}$	1000	–
$M_{seed,s}$	1397000	A/m
$M_{p,s}$	1735000	A/m
R_s	1.0×10^{-6}	m
R_p	50×10^{-9}	m
$\rho_{pol,p}$	950	kg/m ³
T	300	K
k	1.38×10^{-23}	J/K

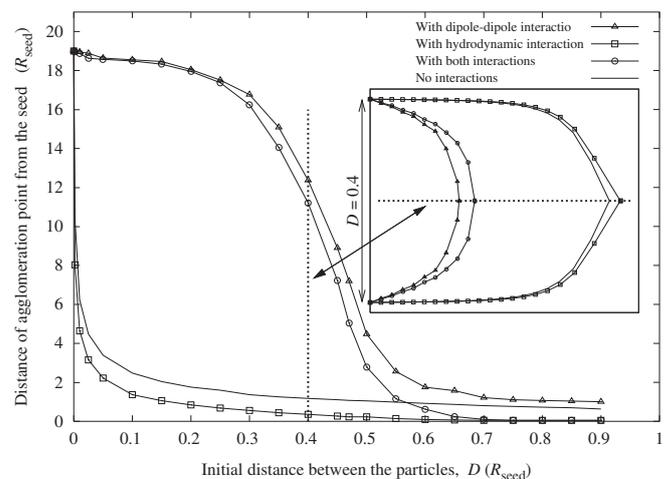


Fig. 2. Distance of agglomeration point from the seed plotted against initial distance between the particles, D , with (a) dipole-dipole magnetic interaction only, (b) hydrodynamic interaction only, (c) both interactions and (d) no interactions between the particles. All other conditions are the reference case as in Table 1. Inset represents the trajectories as the particles move towards to the seed.

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 agglomerate earlier for all initial particle distances up to $D \sim 1$ (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 $|\vec{r}_1 - \vec{r}_2|^{-3}$ and $|\vec{r}_1 - \vec{r}_2|^{-1}$, 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 $D \sim 0.56$. 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

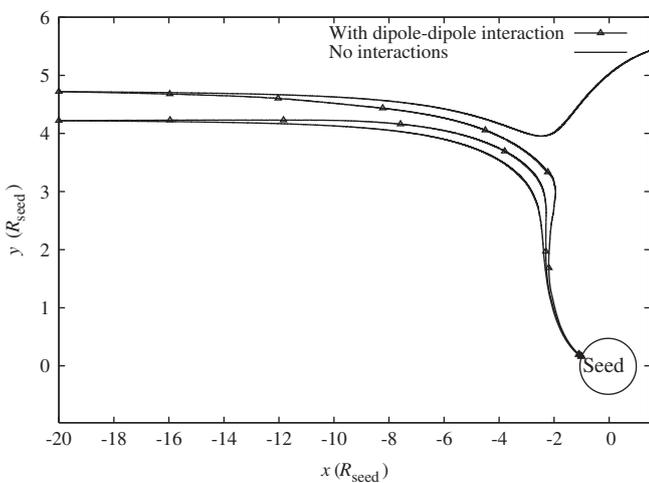


Fig. 3. The trajectories of the particles are presented with the magnetic dipole–dipole interaction and no interactions. Initial position of Particle 1 and 2 are $(-20, \lambda_c^*+0.25)$ and $(-20, \lambda_c^*-0.25)$. With the magnetic dipole–dipole interaction both particles are captured.

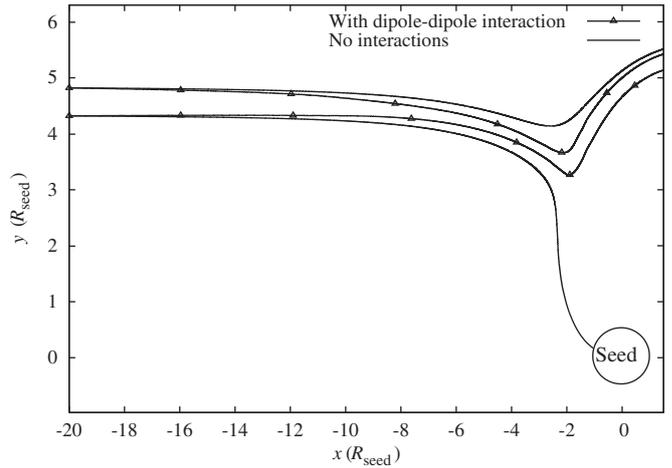


Fig. 4. The trajectories of the particles are presented with the magnetic dipole–dipole interaction and no interactions. Initial position of Particle 1 and 2 are $(-20, \lambda_c^*+0.35)$ and $(-20, \lambda_c^*-0.15)$. With the magnetic dipole–dipole interaction, neither particle is captured.

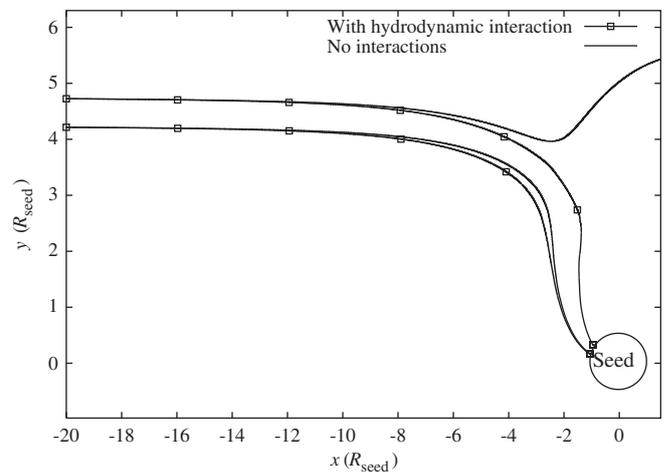


Fig. 5. The trajectories of the particles are presented with the hydrodynamic interaction and no interactions. Initial position of Particle 1 and 2 are $(-20, \lambda_c^*+0.255)$ and $(-20, \lambda_c^*-0.255)$. With the hydrodynamic interaction both particles are captured.

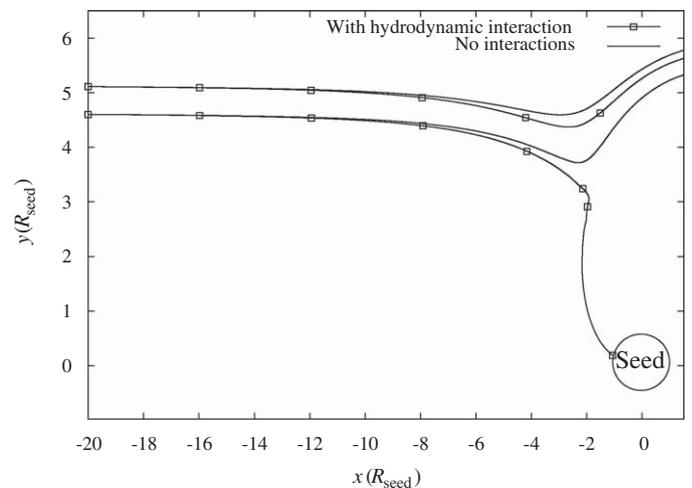


Fig. 6. The trajectories of the particles are presented with the hydrodynamic interaction and without any interaction. Initial position of Particle 1 and 2 are $(-20, \lambda_c^*+0.644)$ and $(-20, \lambda_c^*+0.134)$. With hydrodynamic interaction, Particle 2 is now captured.

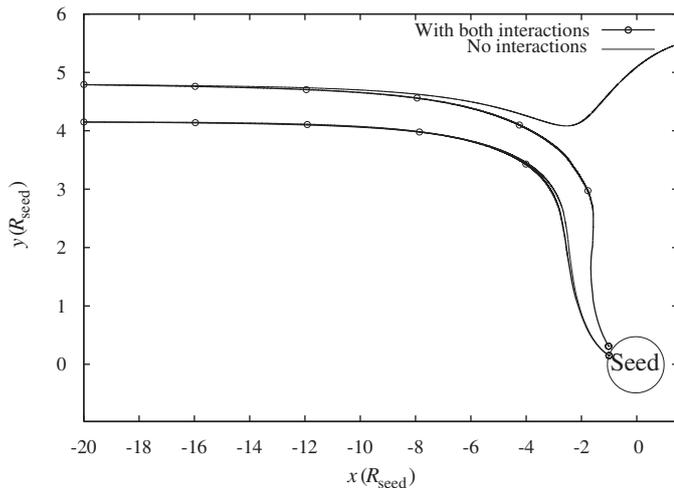


Fig. 7. The trajectories of the particles are presented with both interactions and no interactions. Initial position of Particle 1 and 2 are $(-20, \lambda_c^*+0.32)$ and $(-20, \lambda_c^*-0.32)$. With both interactions both particles captured.

cross section, λ_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 λ_c^* , corresponding to the initial position $(-20, \lambda_c^*)$ where λ_c^* is 4.47. This critical value, λ_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 λ_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 $0.25 R_{\text{seed}}$ corresponding to a $\sim 6\%$ (absolute) increase in λ_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 $D = 0.40$. 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 following is observed. As might be expected, 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 now lower than for the non-interacting case. Thus, Particle 1 has caused the “non-capture” of Particle 2. Fig. 4 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 $\sim 4\%$ (absolute) decrease in λ_c .

5.2.2. Hydrodynamic interaction

In the case with (only) hydrodynamic interaction, we find that the maximum value of D for which both particles are now captured is 0.41 which is slightly larger than in the case with (only) magnetic dipole–dipole interaction. Here, the upper particle (Particle 1) repels the lower particle (Particle 2) and the lower particle attracts the upper particle due to their velocities relative to the velocity of blood. In Fig. 5, 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 $\sim 6\%$ (absolute) increase in λ_c .

To explore further the effect of hydrodynamic 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, upper 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 still 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 $0.134 R_{\text{seed}}$, which corresponds to a $\sim 3\%$ (absolute) increase in λ_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 particles relative to the fluid is an important factor.

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 $0.32 R_{\text{seed}}$ corresponding to a $\sim 7\%$ (absolute) increase in λ_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 $0.16 R_{\text{seed}}$, 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 $0.138 R_{\text{seed}}$, 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 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 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 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-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 Avilés 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

- [1] Q.A. Pankhurst, J. Connolly, S.K. Jones, J. Dobson, *J. Phys. D: Appl. Phys.* 36 (2003) R167.
- [2] C.C. Berry, A.S.G. Curtis, *J. Phys. D: Appl. Phys.* 36 (2003) R198.
- [3] A.D. Grief, G. Richardson, *J. Magn. Magn. Mater.* 293 (2005) 455.
- [4] Gh. Iacob, O. Rotariu, N.J.C. Strachan, U.O. Häfeli, *Biorheology* 41 (2004) 599.
- [5] B.B. Yellen, Z.G. Forbes, D.S. Halverson, G. Fridman, K.A. Barbee, M. Chorny, R. Levy, G. Friedman, *J. Magn. Magn. Mater.* 293 (2005) 647.
- [6] J.A. Ritter, A.D. Ebner, K.D. Daniel, K.L. Stewart, *J. Magn. Magn. Mater.* 280 (2004) 184.
- [7] M.O. Avilés, A.D. Ebner, H. Chen, A.J. Rosengart, M.D. Kaminski, J.A. Ritter, *J. Magn. Magn. Mater.* 293 (2005) 605.
- [8] H. Chen, A.D. Ebner, M.D. Kaminski, A.J. Rosengart, J.A. Ritter, *J. Magn. Magn. Mater.* 293 (2005) 616.
- [9] M.O. Avilés, A.D. Ebner, J.A. Ritter, *J. Magn. Magn. Mater.* 310 (2007) 131.
- [10] M.O. Avilés, A.D. Ebner, J.A. Ritter, *J. Magn. Magn. Mater.* 320 (2008) 2640.
- [11] M.O. Avilés, A.D. Ebner, J.A. Ritter, *J. Magn. Magn. Mater.* 320 (2008) 2704.
- [12] M.O. Avilés, J.O. Mangual, A.D. Ebner, J.A. Ritter, *Int. J. Pharm.* 361 (2008) 202.
- [13] P.J. Cregg, K. Murphy, A. Mardinoglu, *J. Magn. Magn. Mater.* 320 (2008) 3272.
- [14] C. Mikkelsen, M.F. Hansen, H. Bruus, *J. Magn. Magn. Mater.* 293 (2005) 578.
- [15] R. Mehasni, M. Feliachi, M. Latreche, *IEEE Trans. Magn.* 43 (2007) 3488.
- [16] K.J. Binns, P.J. Lawrenson, C.W. Trowbridge, *The Analytical and Numerical Solution of Electric and Magnetic Fields*, Section 3.2.2, Wiley, New York, 1992.
- [17] P.J. Cregg, L. Bessais, *J. Magn. Magn. Mater.* 202 (1999) 554.
- [18] M.I. Shliomis, *Magnetic fluids*, *Sov. Phys. Usp.* 17 (1974) 153.
- [19] H.C. Bryant, D.A. Sergatskov, D. Lovato, N.L. Adolphi, R.S. Larson, E.R. Flynn, *Phys. Med. Biol.* 52 (2007) 4009.
- [20] J.D. Jackson, *Classical Electrodynamics*, Section 5.11, Wiley, New York, 1999.
- [21] OpenCFD Ltd, OpenFOAM 1.4, <<http://www.open CFD.co.uk>>, 2009.

Many particle magnetic dipole-dipole and hydrodynamic interactions in magnetisable stent assisted magnetic drug targeting

P.J. Cregg^{a,*}, Kieran Murphy^a, Adil Mardinoglu^a and
Adriele Prina-Mello^{b,*}

^a*SEAM Centre, Materials Characterisation and Processing Group,
Waterford Institute of Technology, Waterford, Ireland*

^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 (*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 *in vitro* experimental results and with presented literature. The implications in Nanotechnology and Nanomedicine are based on the prediction of the particle

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 (Adriale 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 $0.86\ \mu\text{m}$ 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 $62.5\ \mu\text{m}$ 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 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

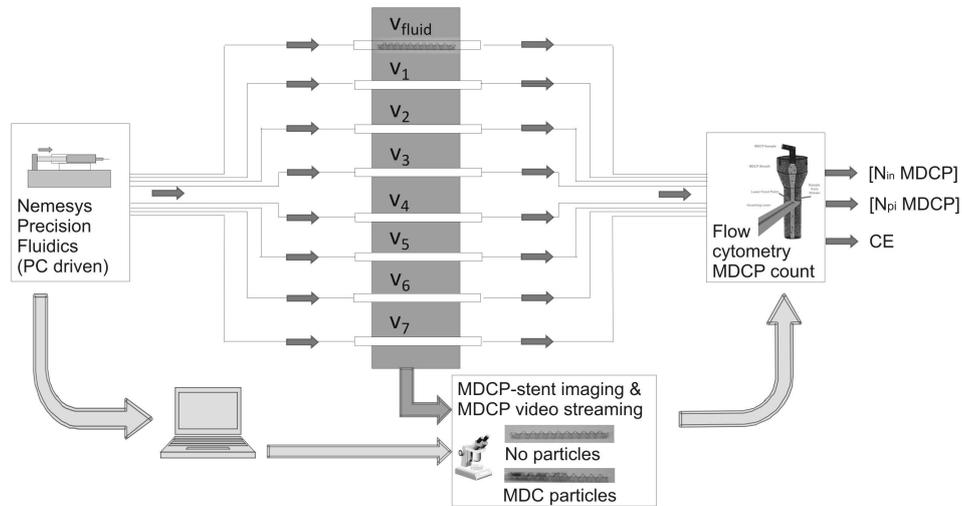


Fig. 1. Schematic diagram of the *in vitro* experimental setup used to study a stent-based IA-MDT system.

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 were 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

software (NCH Software, USA).

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 4×10^{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 11.2×10^{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 were 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_{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

$$\vec{F}_{s_n} = 6\pi \eta_f R_{p_n} (\vec{v}_f - \vec{v}_{p_n}), \quad (1)$$

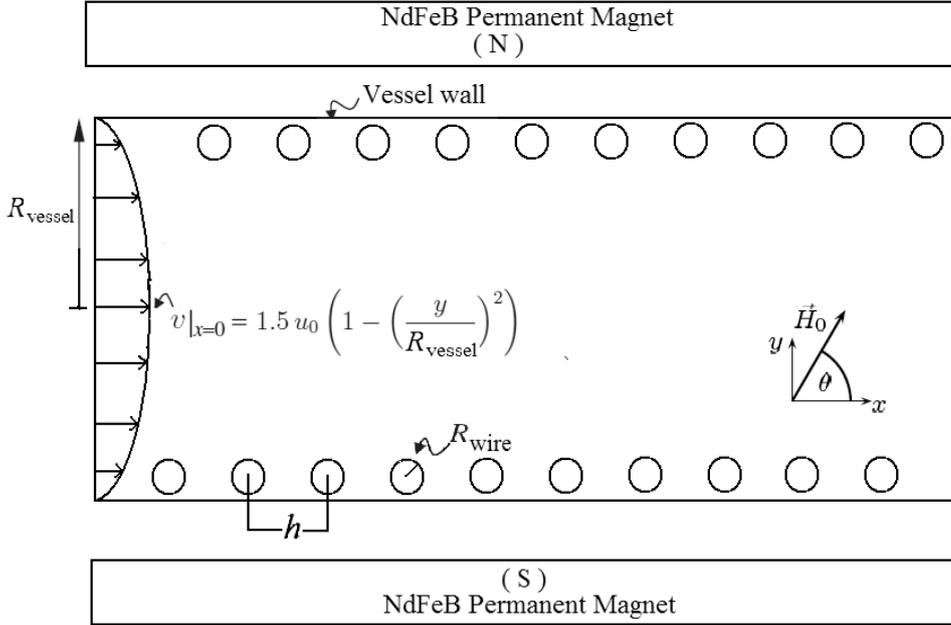


Fig. 2. Schematic of the control volume (CV) used for determining the magnetisable stent collection efficiency (CE) through analysis of the corresponding MDCP trajectories. The CV has dimensions of 2 cm and 0.05 cm and encompasses a ten-loop stent within an expanded vessel. The MDCPs enter the CV from the left with a reduced average velocity defined by a parabolic profile and unexpanded average blood vessel velocity.

where η_f is the viscosity of the fluid, R_{p_n} is the radius of MDCP n , and \vec{v}_f and \vec{v}_{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}_{hyd_n} , which acts on MDCP n due to the presence of other

$(N - 1)$ MDCPs, can be written as,

$$\vec{F}_{hyd_n} = \sum_{\substack{i=1 \\ i \neq n}}^N \xi_{ni} \cdot (\vec{v}_f - \vec{v}_{p_i}) \quad (2)$$

where ξ_{ni} is the modification due to the hydrodynamic interaction given by

$$\xi_{ni} = -6\pi \eta_f R_{p_n} \frac{3 R_{p_i}}{4 |\vec{r}_n - \vec{r}_i|} \left(\mathbf{1} + \frac{(\vec{r}_n - \vec{r}_i) \otimes (\vec{r}_n - \vec{r}_i)}{|\vec{r}_n - \vec{r}_i|^2} \right) \quad (3)$$

where R_{p_i} 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 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

$$\vec{F}_m = (\vec{m} \cdot \nabla) \vec{B}_{total}, \quad (4)$$

where \vec{m} is the magnetic moment and \vec{B}_{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_{p_n} 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

$$\vec{F}_{mm_n} = (\vec{m}_n \cdot \nabla) \vec{B}_{total_n} \quad (5)$$

where \vec{m}_n is the total magnetic moment of MDCP n , \vec{B}_{total_n} is the total magnetic flux acting on MDCP n . It can be taken as

$$\vec{B}_{total_n} = \vec{B} + \sum_{\substack{i=1 \\ i \neq n}}^N d\vec{B}_i \quad (6)$$

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

$$d\vec{B}_n(\vec{r}) = \frac{1}{3} \left(\mu_0 M_{f_{m,p,s}} \frac{L(\beta)}{B} \right) \frac{R_{pn}^3}{|\vec{r} - \vec{r}_n|^3} \left(\frac{3 \left(\vec{B}(\vec{r}_n) \cdot (\vec{r} - \vec{r}_n) \right)}{|\vec{r} - \vec{r}_n|^2} (\vec{r} - \vec{r}_n) - \vec{B}(\vec{r}_n) \right) \quad (7)$$

where μ_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 $M_{f_{m,p,s}}$ 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 con-joined regions: inside and outside the stent wires. Thus for outside the stent wires regions we have magnetic flux given by [8–13, 16]

$$\vec{B} = \mu_0(\vec{H}_0 - \nabla\phi) \quad (8)$$

where \vec{H}_0 is the applied homogeneous magnetic field as in Fig. 2 and ϕ represents the reduced magnetic scalar potential which in the region outside the stent wires is given by [19–21]

$$\phi = H_0 R_{wire}^2 \alpha_{wire} \frac{x \cos \theta + y \sin \theta}{x^2 + y^2}, \quad (9)$$

where R_{wire} is the radius of the stent wire implant, α_{wire} is the demagnetising factor of the stent wire (given by Eq. (11)). The induced magnetisation of the wire, \vec{M}_{wire} , is taken to be parallel to the external magnetic field, \vec{H}_0 , and can be calculated from

$$\vec{M}_{wire} = 2\alpha_{wire}\vec{H}_0, \quad (10)$$

where α_{wire} is the demagnetising factor for an infinitely long cylinder in a perpendicular field taken as

$$\alpha_{wire} = \min\left(\frac{\chi_{wire,0}}{2 + \chi_{wire,0}}, \frac{M_{wire,s}}{2H_0}\right), \quad (11)$$

where $\chi_{wire,0}$ and $M_{wire,s}$ are the zero field susceptibility and saturation magnetisation of the ferromagnetic wire respectively and \vec{H}_0 can be written

$$\vec{H}_0 = (H)_0 \cos\theta \hat{x} + H_0 \sin\theta \hat{y}, \quad (12)$$

where H_0 is the magnitude of the applied field and θ 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}_{total} can be calculated from the Langevin function [6, 15, 22–24]

$$L(\beta) = \coth(\beta) - \frac{1}{\beta}, \quad (13)$$

with Langevin argument

$$\beta = \frac{m_{fm,p} B_{total}}{kT}, \quad (14)$$

where B_{total} is the magnitude of \vec{B}_{total} , k is Boltzmann's constant, T is the absolute temperature and $m_{fm,p}$ is the magnitude of the magnetic moment of the magnetite in the MDCPs. The magnetic moment of each magnetite nanoparticle within the MDCP, $\vec{m}_{fm,p}$, can be written as

$$\vec{m}_{fm,p} = V_{fm,p} M_{fm,p,s} \frac{\vec{B}}{B} \quad (15)$$

where $V_{fm,p}$ is the spherical volume of a single domain magnetite nanoparticle and $M_{fm,p,s}$ is the (volume) saturation magnetisation of the magnetite inside the MDCPs. Note that $M_{fm,p,s}$ and $m_{fm,p,s}$ 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

$$\vec{m} = \omega_{fm,p} V_p M_{fm,p,s} L(\beta) \frac{\vec{B}}{B} \quad (16)$$

where V_p is the MDCP volume and $\omega_{fm,p}$ is the volume fraction of ferromagnetic material in the MDCP, related to its weight fraction $x_{fm,p}$ through [16]

$$\omega_{fm,p} = \frac{x_{fm,p}}{x_{fm,p} + (1 - x_{fm,p})\rho_{fm,p}/\rho_{pol,p}} \quad , \quad (17)$$

where $\rho_{fm,p}$ is the density of the ferromagnetic material in the MDCP and $\rho_{pol,p}$ is the density of the polymer material in the MDCP. In this model the value of $\omega_{fm,p}$ 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

$$\nabla \cdot \vec{v}_f = 0, \quad (18)$$

and the Navier-Stokes equation

$$\rho_f [(\vec{v}_f \cdot \nabla) \vec{v}_f] = \nabla P + \eta_f \nabla^2 \vec{v}_f, \quad (19)$$

where ρ_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

$$v_{f,x}|_{x=0} = 1.5 u_0 \left(1 - \left(\frac{y}{R_{vessel}} \right)^2 \right), \quad (20)$$

$$v_{f,y}|_{x=0} = 0 \quad (21)$$

where u_0 is the average inlet fluid velocity and R_{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}_{i_n} , as

$$\vec{F}_{s_n} + \vec{F}_{hyd_n} + \vec{F}_{mm_n} = \vec{F}_{i_n}. \quad (22)$$

For MDCP n , by ignoring the inertial forces, \vec{F}_{i_n} , we rewrite Eq. (22) as

$$6\pi\eta_f R_{p_n} (\vec{v}_f - \vec{v}_{p_n}) + \sum_{\substack{i=1 \\ i \neq n}}^N \xi_{ni} \cdot (\vec{v}_f - \vec{v}_{p_i}) + (\vec{m}_n \cdot \nabla) (\vec{B}_{total})_n = 0. \quad (23)$$

Hence, we can obtain \vec{v}_{p_n} 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

$$CE = \frac{2R_{vessel} - y_1 + y_2}{2R_{vessel}} 100, \quad (24)$$

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), λ_c^* 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 11.2×10^{10} MDCPs per litre and the MDCP concentration of our experimental setup is 4×10^{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 λ_c^* . For this, we place the centre of the particle cluster on the λ_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 ($\mu_0 H_0 = 0.17 \text{ T}$) and we increase the blood velocity from $u_0 = 2.1 \text{ cm/s}$ to $u_0 = 42.4 \text{ cm/s}$. The resulting CEs for these simulations are shown in Fig. 3.

Secondly, using the same methodology we applied $\mu_0 H_0 = 0.65$ 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.

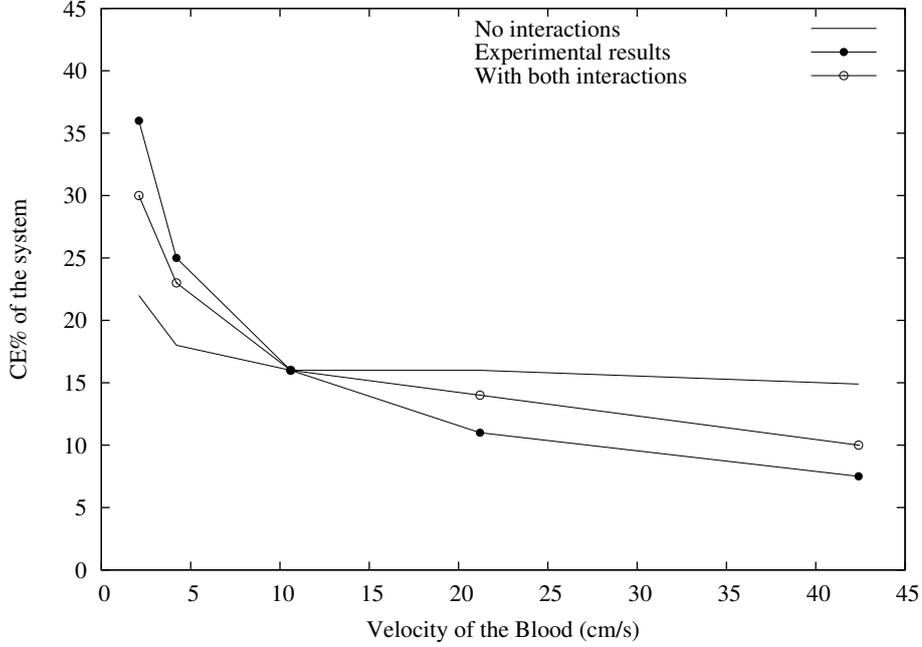


Fig. 3. The collection efficiency (CE) of the system plotted as a function of the blood velocity (2.1, 4.2, 10.6, 21.2, 42.4 cm/s) at the applied field $\mu_0 H_0 = 0.17$ T.

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 (≤ 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 predicts lower collection than the results of Avilés model without interactions. This is due to the effect of

Properties	Symbol	Units	Values	Data type
<i>MDCPs Properties</i>				
Polymer material	-	-	P(S/V-COOH)Mag	Physical
Radius	R_p	μm	0.435, 0.43	Physical
Saturation magnetisation	$M_{p,s}$	kA/m	22.4	Measured
<i>MDCPs Magnetic Material Properties</i>				
Material	-	-	Magnetite	Physical
Weight content	$x_{fm,p}$	wt%	25, 45.8	Physical
Volume content	$\omega_{fm,p}$	-	6.4	Measured
Saturation magnetisation	$M_{fm,p,s}$	kA/m	351.9	Measured
Magnetic moment	$m_{fm,p}$	Am^2	2.03×10^{-19}	Measured
Radius	$R_{fm,p}$	nm	5.18	Calculated
<i>Physical Properties</i>				
Number of Particles	-	particle/L	11.2×10^{10} , 4×10^{10}	Physical
Temperature	T	K	300	Physical
Boltzmann's constant	k_B	J/K	1.38×10^{-23}	Physical
Permeability of vacuum	μ_0	Tm/A	$4\pi \times 10^{-7}$	Physical
<i>Applied Field Properties</i>				
Magnitude	$\mu_0 H_0$	T	0.0–0.7	Physical
Angle of field direction	θ	-	$\pi/2$	Physical
<i>Stent Properties</i>				
Material	-	-	SS 430	Physical
Wire radius	R_{wire}	μm	62.5	Physical
Loop separation	h	cm	0.2	Physical
Number of loops	N_l	-	10	Physical
Coil length	L	cm	2	Physical
Saturation magnetisation	$M_{implant,s}$	kA/m	1261	Measured
Magnetic susceptibility	$\chi_{implant,0}$	-	1000	Physical
<i>Blood & Vessel Properties</i>				
Velocity	u_0	cm/s	2.1, 4.2, 10.6, 21.2, 42.4	Physical
	u_0	cm/s	0.58, 1.17, 2.34, 4.68, 11.7, 23.4, 52.6	Physical
Volume	V_{blood}	ml	10	Physical
Density	ρ_b	kg/m^3	1000	Physical
Viscosity	η_b	kg/ms	1.0×10^{-3}	Physical
Vessel radius	R_{vessel}	cm	0.05, 0.04	Physical

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].

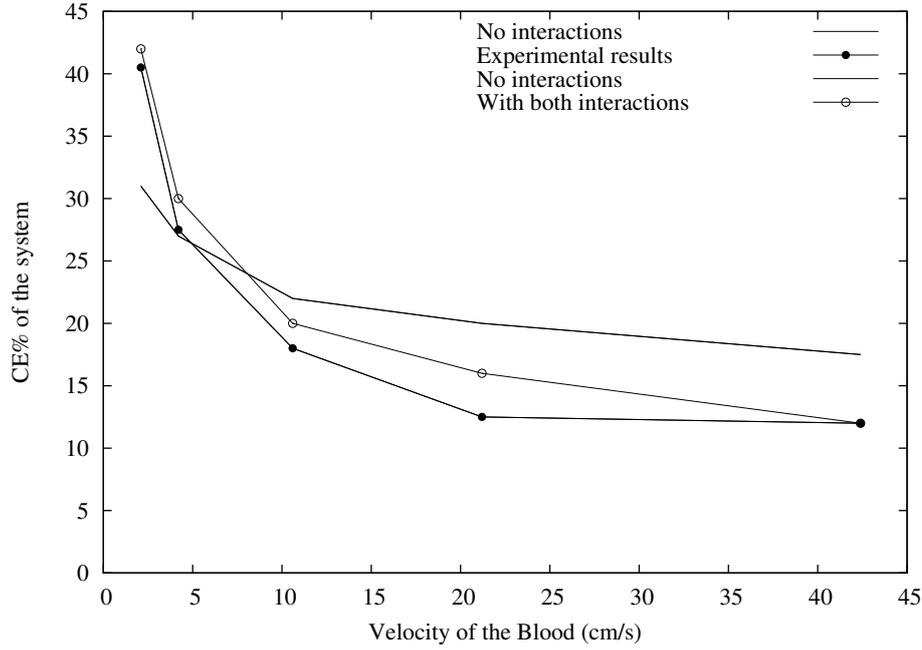


Fig. 4. The collection efficiency (CE) of the system plotted as a function of the blood velocity (2.1, 4.2, 10.6, 21.2, 42.4 cm/s) at the applied field $\mu_0 H_0 = 0.65$ T. hydrodynamic interactions on the velocity of MDCPs and so the trajectories of the MDCPs.

6.3 Comparison of the Mathematical Model and Experimental Results

Next, we compare the results of the mathematical model and *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.

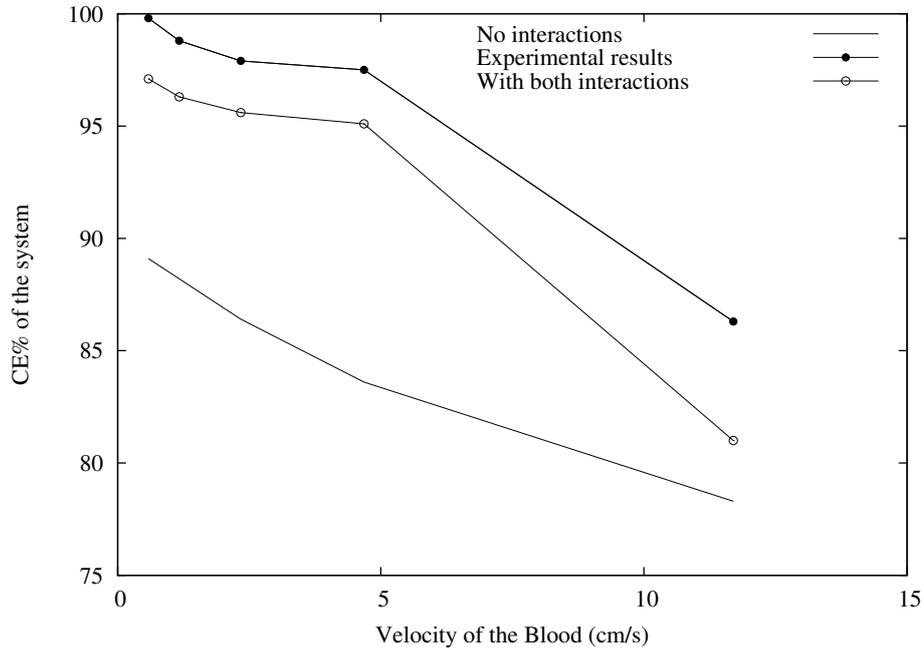


Fig. 5. The collection efficiency (CE) of the system plotted as a function of the blood velocity (0.58, 1.17, 2.34, 4.68, 11.7 cm/s) at the applied field $\mu_0 H_0 = 0.15$ T.

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

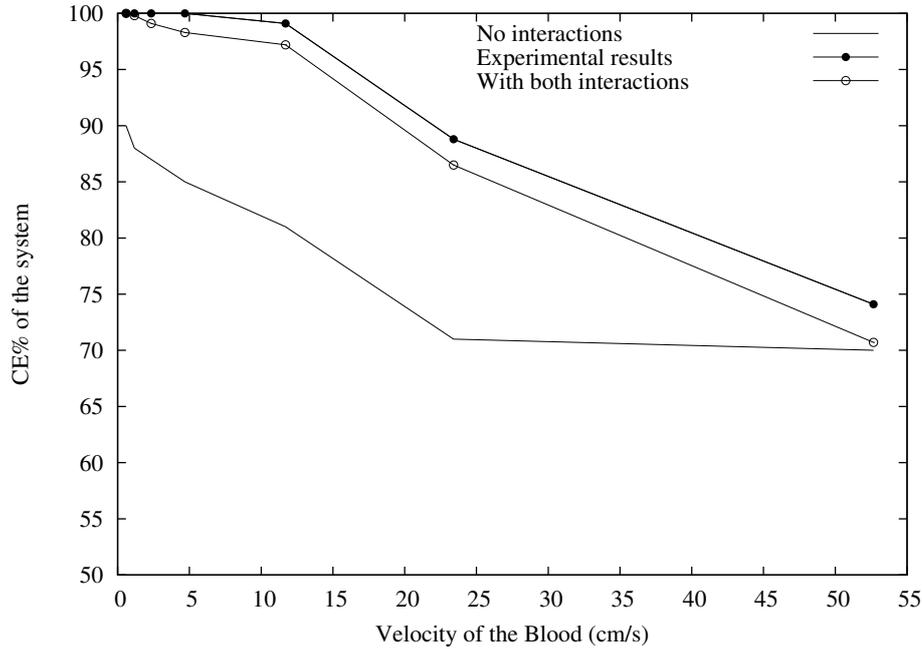


Fig. 6. The collection efficiency (CE) of the system plotted as a function of the blood velocity (2.34, 4.68, 11.7, 23.4, 52.6 cm/s) at the applied field $\mu_0 H_0 = 0.60$ T.

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	Avilés <i>et al.</i> model	Cregg <i>et al.</i> model
Vessel radius (cm)	0.05	0.04
Velocity Range (cm/s)	2.1–42.4	0.58–52.6
Magnetic Field (T)	0.17, 0.65	0.15, 0.60
Number of Repeats	-	10

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

- [1] W. P. Torchilin, Drug targeting, *J. Pharm. Sci.* 11 (2000) S81–S91.
- [2] D. J. A. Crommelin, G. Scherphof, G. Storm, Active targeting with particulate carrier systems in the blood compartment, *Adv. Drug Deliv. Rev.* 17 (1995) 49–60.
- [3] A. S. Lübbe, C. Bergemann, J. Brock, D. G. McClure, Physiological aspects in magnetic drug-targeting, *J. Magn. Magn. Mater.* 194 (1999) 149–155.
- [4] A. S. Lübbe, C. Aleciou, C. Bergemann, Clinical applications of magnetic drug targeting, *J. Surg. Res.* 95 (2001) 200–206.
- [5] A. J. Lemke, M.-I. S. von Pilsach, A. S. Lübbe, C. Bergemann, H. Riess, R. Felix, MRI after magnetic drug targeting in patients with advanced solid malignant tumors., *Eur. Radiol.* 14 (2004) 1949–1955.
- [6] A. D. Grief, G. Richardson, Mathematical modelling of magnetically targeted drug delivery, *J. Magn. Magn. Mater.* 293 (2005) 455–463.
- [7] M. O. Avilés, A. D. Ebner, J. O. Mangual, J. A. Ritter, Isolated swine

- heart ventricle perfusion model for implanted assisted-magnetic drug targeting, *Int. J. Pharm.* 361 (2008) 202–208.
- [8] M. O. Avilés, A. D. Ebner, J. A. Ritter, In vitro study of magnetic particle seeding for implant assisted-magnetic drug targeting, *J. Magn. Magn. Mater.* 320 (2008) 2640–2646.
- [9] M. O. Avilés, A. D. Ebner, J. A. Ritter, Implant assisted-magnetic drug targeting: Comparison of in vitro experiments with theory, *J. Magn. Magn. Mater.* 320 (2008) 2704–2713.
- [10] M. O. Avilés, A. D. Ebner, J. A. Ritter, Ferromagnetic seeding for the magnetic targeting of drugs and radiation in capillary beds, *J. Magn. Magn. Mater.* 310 (2007) 131–144.
- [11] M. O. Avilés, A. D. Ebner, H. Chen, A. J. Rosengart, M. D. Kaminski, J. A. Ritter, Theoretical analysis of a transdermal ferromagnetic implant for retention of magnetic drug carrier particles, *J. Magn. Magn. Mater.* 293 (2005) 605–615.
- [12] A. J. Rosengart, M. D. Kaminski, H. Chen, P. L. Caviness, A. D. Ebner, J. A. Ritter, Magnetizable implants and functionalised magnetic carriers: A novel approach for non-invasive yet targeted drug delivery, *J. Magn. Magn. Mater.* 293 (2005) 633–638.
- [13] H. Chen, A. D. Ebner, M. D. Kaminski, A. J. Rosengart, J. A. Ritter, Analysis of magnetic drug carrier particle capture by a magnetizable intravascular stent: Parametric study with multi-wire two-dimensional model, *J. Magn. Magn. Mater.* 293 (2005) 616–632.
- [14] G. Iacob, O. Rotariu, N. J. C. Strachan, U. O. Häfeli, Magnetizable needles and wires - modeling an efficient way to target magnetic microspheres in vivo, *Biorheology* 41 (2004) 599–612.
- [15] B. B. Yellen, Z. G. Forbes, D. S. Halverson, G. Fridman, K. A. Barbee,

- M. Chorny, R. Levy, G. Friedman, Targeted drug delivery to magnetic implants for therapeutic applications, *J. Magn. Magn. Mater.* 293 (2005) 647–654.
- [16] J. A. Ritter, A. D. Ebner, K. D. Daniel, K. L. Stewart, Application of high gradient magnetic separation principles to magnetic drug targeting, *J. Magn. Magn. Mater.* 280 (2004) 184–201.
- [17] H. B. C. I. Mikkelsen, M. F. Hansen, Theoretical comparison of magnetic and hydrodynamic interactions between magnetically tagged particles in microfluidic systems, *J. Magn. Magn. Mater.* 293 (2005) 578–583.
- [18] R. Mehasni, M. Feliachi, M. Latreche, Effect of the magnetic dipole-dipole interaction on the capture efficiency in open gradient magnetic separation, *IEEE Trans. Magn.* 43 (2007) 3488.
- [19] P. J. Cregg, K. Murphy, A. Mardinoglu, Inclusion of magnetic dipole-dipole and hydrodynamic interactions in implant assisted magnetic drug targeting, *J. Magn. Magn. Mater.* 321 (2009) 3893–3898.
- [20] P. J. Cregg, K. Murphy, A. Mardinoglu, Calculation of nanoparticle capture efficiency in magnetic drug targeting, *J. Magn. Magn. Mater.* 320 (2008) 3272–3275.
- [21] K. J. Binns, P. J. Lawrenson, C. W. Trowbridge, *The analytical and numerical solution of electric and magnetic Fields*, Wiley, 1992, sec 3.2.2.
- [22] H. C. Bryant, D. A. Sergatskov, D. Lovato, N. L. Adolphi, R. S. Larson, E. R. Flynn, Magnetic needles and superparamagnetic cells, *Phys. Med. Biol.* 52 (2007) 4009–4025.
- [23] M. I. Shliomis, *Magnetic fluids*, *Sov. Phys. Usp.* 17 (1974) 153–169.
- [24] P. J. Cregg, L. Bessais, Series expansions for the magnetisation of a solid superparamagnetic system of non-interacting particles with anisotropy, *J. Magn. Magn. Mater.* 202 (1999) 554–564.

- [25] OpenCFD Ltd, OpenFOAM 1.4, <http://www.opencfd.co.uk> (2007).
- [26] E. Allen, P. Smith, J. Henshaw, A review of particles agglomeration, US Department of Energy.
- [27] W. Dzwiniel, K. Boryczko, D. A. Yuen, A discrete-particle model of blood dynamics in capillary vessels, *Journal of Colloid and Interface Science* 258 (2003) 163–173.