

ABSTRACT

RICHARDS, ANDREW LATIMER. Experimental Investigations into the Targeted Delivery of Microspheres in Radioembolization Therapy. (Under the direction of Gregory D. Buckner).

The evolution of liver tumor treatment has enabled marked improvements in patient survivability and quality of life. In recent years, radioembolization (RE) procedures, in which radioactive microspheres are introduced directly into the hepatic vasculature, have provided promising results for the treatment of liver tumors. These procedures combine the efficacy of radiation therapy with the selective delivery of transarterial embolization. Despite the well documented advances, however, the lack of a direct tumor-targeting strategy has limited the widespread adoption of RE.

Recent work employing computational fluid-particle modeling of the hepatic arteries has suggested that optimized release locations exist which maximize ^{90}Y microsphere delivery to targeted artery branches. Studies examining this strategy have been primarily computational in nature; no experimental validation has been presented demonstrating directed microsphere delivery in a hepatic artery model. Furthermore, limited data has been published regarding the physical properties of clinical ^{90}Y RE microspheres most relevant to the simulation model.

This dissertation details experimental investigations examining the methods and materials used in this targeting strategy. First, a hepatic artery circulation model is constructed to match simulation-specified flow conditions. Controlled microsphere injections are performed at various locations within the model and the resulting branch distributions are compared with those predicated computationally. For the first time ever, selectable targeting

is demonstrated in a physical hepatic model. Achieving predictable targeting in a controlled environment lays the groundwork for adapting this technique for eventual clinical application.

Complete physical characterization of the treatment microspheres is required for accurate modeling and targeting in a clinical environment. For the first time in the more than 40 years of radioembolization therapy, independently measured data is reported for microsphere concentration, size distribution, and density values. The physical properties of the microspheres are characterized using multiple particle measurement techniques. Complete knowledge of the microsphere dose can only help to improve the treatment and analysis of patients receiving RE-based therapies.

Current RE procedures do not easily permit physicians to determine the delivered dose of microspheres (i.e. number of delivered microspheres), a necessary step when evaluating a targeting methodology. A novel micro-fluidics device utilizing liquid metal electrodes is developed which provides accurate microsphere detection in an easily-fabricated, inexpensive package. This device demonstrates a proof-of-concept which could be further developed for clinical integration.

© Copyright 2013 by Andrew Latimer Richards

All Rights Reserved

Experimental Investigations into the Targeted Delivery of
Microspheres in Radioembolization Therapy

by
Andrew Latimer Richards

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Mechanical Engineering

Raleigh, North Carolina

2013

APPROVED BY:

Gregory D. Buckner
Committee Chair

Michael Dickey

Ola Harrysson

Clement Kleinstreuer

DEDICATION

To my wife, Libby.

BIOGRAPHY

Andrew Richards was born and raised in Irmo, SC, a suburb of Columbia. A proud product of public education, Andrew graduated from Dutch Fork High School in 2000. He attended Clemson University from 2000-2004, graduating with a Bachelor of Science degree in Computer Engineering with a minor in International Engineering and Science. During his time at Clemson, Andrew participated in internships with Mettler-Toledo Inc. in Inman, SC, and with Sealevel Systems in Liberty, SC, and also studied abroad in Ulm, Germany. Following graduation, Andrew continued working with Sealevel Systems on a full-time basis, though his time there was short-lived.

In 2005, dissatisfied with the direction his career was headed, Andrew moved to Raleigh, NC to pursue a Master of Science in Biomedical Engineering, offered as a joint program by North Carolina State University and University of North Carolina Chapel Hill. Under the direction of Dr. Gregory Buckner, Andrew completed his M.S. in 2008 and was encouraged to continue with doctoral studies further investigating engineering solutions to medical problems.

ACKNOWLEDGMENTS

I am grateful to all members, past and present, of the Electromechanics Research Lab, who offered assistance, guidance, and friendship throughout my time here. I would especially like to thank Shaphan Jernigan, who was there since the beginning and on whom I could always rely for assistance or brainstorming sessions.

I'd like to thank Drs. Michael Dickey, Glenn Walker, and Thomas Ward for allowing me to conduct experiments in their labs. Their students, though already busy, never hesitated to offer assistance.

I am especially grateful to my advisor, Dr. Buckner, for always challenging me to do more and for helping me to stay on track when I strayed too far.

Most of all, I would like to thank my wife, Libby, and my parents, Varner and Lynne, for their unconditional support throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
Chapter 1 Introduction	1
1.1 Background and motivation	1
1.1.1 Radioembolization	5
1.1.2 Model-based targeting	8
1.2 Specific aims	10
1.3 Organization	11
Chapter 2 Experimental targeting	12
2.1 Introduction	12
2.2 Methods and materials	14
2.2.1 Model construction	14
2.2.2 Experimental parameter determination	20
2.2.3 Simulation design	22
2.2.4 Experimental microsphere injection	24
2.3 Results	26
2.4 Discussion and limitations	30
2.5 Conclusion	36
Chapter 3 Microsphere physical properties	37
3.1 Introduction	37
3.2 Methods and materials	39
3.2.1 Concentration	39
3.2.2 Size distribution	40
3.2.3 Density	41
3.3 Results	43
3.3.1 Concentration	43
3.3.2 Size distribution	45
3.3.3 Density	49
3.4 Discussion	50
3.5 Conclusion	55

Chapter 4 Micro-Coulter counter: Design and demonstration.....	56
4.1 Introduction	56
4.2 Methods and materials	60
4.2.1 Device design.....	60
4.2.2 Device fabrication.....	62
4.2.3 Measurement circuitry	65
4.2.4 Signal processing and analysis.....	67
4.3 Results	70
4.4 Discussion	73
4.5 Conclusion.....	77
 Chapter 5 Conclusions.....	 78
5.1 Future work	79
 REFERENCES.....	 82
 APPENDICES	 89
 Appendix A Injection tube location	 90
A.1 Introduction.....	90
A.2 Determining required wire length for desired position.....	91
A.3 Determining actual position from wire length	93
A.4 Considerations.....	94
 Appendix B Magnetic mixer design	 95
B.1 Introduction	95
B.2 Theory	95
B.3 Design	97
 Appendix C Particle injection rate determination	 99
C.1 Analysis.....	100
C.1.1 Inlet tube velocity.....	100
C.1.2 Syringe plunger velocity	101
C.2 Assumptions.....	102
 Appendix D Injection tip deflection	 103
D.1 Introduction.....	103

D.2 Bending analysis	104
D.2.1 Injection tube weight.....	105
D.2.2 Apparent force from flexible tube.....	112
D.3 Material specifications	117
D.3.1 Injection tube.....	118
D.3.2 Flexible tubing	119
D.4 Estimated tip deflection	121
D.5 Assumptions.....	123
Appendix E Image analysis: Particle detection and sizing	125
E.1 Introduction	125
E.2 Image capture	125
E.3 Image analysis	127
E.3.1 Preparation and filtering	128
E.3.2 Thresholding and initial identification	130
E.3.3 Centroid refinement.....	133
E.3.4 Invalid region removal.....	135
E.4 Results	136
E.5 Source code	138

LIST OF TABLES

Table 2.1 Published blood flow rates in patients with diseased livers.....	21
Table 2.2 Simulated and experimental flow distribution.....	24
Table 2.3 Microsphere distribution and injection location	27
Table 3.1 Statistical model results: vial concentration analysis	44
Table 3.2 Statistical model results and descriptive parameters: microsphere diameter analysis	48
Table 3.3 Descriptive statistical parameters: microsphere density analysis	50
Table 4.1 Microsphere count comparison.....	72
Table D.1 Symbolic parameters.....	105
Table D.2 Injection tube manufacturer specifications.....	118
Table D.3 Flexible supply tube manufacturer specifications.....	118
Table D.4 Physical parameters and variable definitions.....	121

LIST OF FIGURES

Figure 1.1 Simplified illustration of a transarterial embolization injection.....	3
Figure 1.2 Tumor localization via digital subtraction angiography; coronal view of a patient with metastatic colon cancer. The three images (A-C) were taken sequentially over 20 seconds during contrast injection in the proper hepatic artery. [14].....	4
Figure 1.3 Partial metastatic colorectal cancer response to ⁹⁰ Y RE treatment. Pre-treatment (left) and post-treatment (right) liver CT images are shown. Arrows denote the tumor locations. [5]	6
Figure 1.4 Illustration of RE injection and radiation coverage. Two possible microsphere deposition scenarios are shown: incomplete (A) and full tumor coverage (B).....	6
Figure 1.5 Illustrative comparison of non-targeted (left) and targeted (right) delivery of microspheres to a tumor site.	9
Figure 2.1 Directed microsphere targeting in hepatic artery simulation model. Image modified from original created by Christopher Basciano.....	13
Figure 2.2 Generalized hepatic artery model; CAD model showing major branches (left) and photograph of 4.4X model used for experiments (right).	15
Figure 2.3 Paddlewheel flowmeter with attached optical IR sensor.....	16
Figure 2.4 Feed-forward control topology used to regulate flow	17
Figure 2.5 Injection tube positioning system.....	18
Figure 2.6 Cross-section view of inlet pipe (grey) with inserted injection tube (red). A-C depict example release location positions (C also includes a rotation). The blue area in D denotes the feasible positioning region.	18
Figure 2.7 Custom computer-controlled syringe pump	19

Figure 2.8 Schematic of system components: supply reservoir (1), collection reservoir (2), gear pumps (3), flowmeter (4), syringe pump (5), positioning mounts (6), injection tube (7), injection plane (8), hepatic artery model (9), branch resistance valves (10), and microsphere filters (11). Arrows indicate the direction of flow.....	20
Figure 2.9 Inlet plane particle release map (right) with color coded regions depicting downstream exit locations.....	26
Figure 2.10 Combined simulation release map and experimental results. Pie charts depict microsphere distribution at each injection location.	28
Figure 2.11 Video frame depicting a line of microspheres exiting the injection tube and traveling past the PHA bifurcation during injection 10. The inset graphic indicates the location on the model. Arrows indicate the microsphere stream.....	29
Figure 2.12 Video frame during injection 10. Microspheres are shown traveling through the LHA to branch 1. The inset graphics indicate the location on the model and the injection location. Arrows indicate the microsphere stream.....	29
Figure 2.13 Video frame during injection 11. Microspheres are shown traveling through the RHA to branch 4. The inset graphics indicate the location on the model and the injection location. Arrows indicate the microsphere stream.....	30
Figure 2.14 Release map comparison after shifting and scaling the experimental injection coordinates.	31
Figure 2.15 Video frame showing the microsphere plum created by syringe plunger stick-slip. Arrows indicate microspheres.....	33
Figure 2.16 Graph of filter weight change over a 6-day period. Filters were wetted at hour 0 and left to open-air dry. The 15-filter average is shown in bold.....	34
Figure 3.1 SIR-Spheres delivery vial (without foil seal).	39
Figure 3.2 Beckman Coulter Z1 used for measuring sample concentrations.	40
Figure 3.3 Photographs taken during density measurement. Images show vacuum filtration (A), an oven dried sample (B) and a filled pycnometer (C).	43

Figure 3.4 Per-vial mean microsphere concentration. Vertical error bars denote the 95% confidence interval for each vial mean. The upper (green) and lower (blue) dashed horizontal lines denote the total estimated mean with and without the outlier measurement (vial 14), respectively. 44

Figure 3.5 Microsphere analysis sample image. Computed microsphere centroids and outlines are indicated in red. 45

Figure 3.6 Cumulative probability plot for all diameter measurements (in blue). A normal distribution and t-distribution are indicated by the dotted black and solid red lines, respectively. 46

Figure 3.7 Histogram of diameter measurements. The red vertical dashed line (red) denotes the mean diameter. The blue dotted curve denotes a non-standardized t-distribution fit to the data. 47

Figure 3.8 Per-vial mean diameter. Vertical error bars denote the 95% confidence interval for each vial distribution, based on a fitted non-standardized t-distribution. The blue dashed horizontal line denotes the total estimated mean diameter. 49

Figure 3.9 Measured density of all 26 vials. Vertical error bars display the 95% confidence interval for each density measurement. The blue dashed horizontal line denotes the total estimated mean density. 50

Figure 3.10 Composite micrograph depicting particulate matter present in some microsphere samples. Arrows indicate sphere fragments (images A and B) and non-spherical objects (images C and D). 53

Figure 3.11 Composite micrograph depicting the textured surface of resin ⁹⁰Y RE microspheres (A), and the smooth surface of bland resin training microspheres (B). 54

Figure 3.12 Scanning electron micrograph of ⁹⁰Y microsphere. Image courtesy of Dr. Andrew Kennedy (via personnel correspondence) 54

Figure 4.1 Illustration of the Coulter principle. Electrical resistance changes between two electrodes (in grey) are detected as particles (marked as red) pass through the sensing zone (marked in yellow). 57

Figure 4.2 Representative illustration of electric field lines across a fluid filled channel in a device with coplanar (left) and parallel cross-channel (right) electrodes.	58
Figure 4.3 Back-lit optical micrograph of a micro-Coulter counter after EGaIn injection. The image has been contrast enhanced to highlight the fluid channel.	61
Figure 4.4 Back-lit detail image of PDMS support posts, which prevent EGaIn penetration into the central fluid channel.	62
Figure 4.5 Completed photoresist-based master mold on Si wafer. Multiple devices can be fabricated simultaneously on one mold.	63
Figure 4.6 Image of completed device with secured electrode connections.	65
Figure 4.7 Active filter/amplifier circuit.	66
Figure 4.8 Illustration of Square law demodulation algorithm used to recover device response.	69
Figure 4.9 Six-frame depiction of a microsphere traversing the aperture channel. High speed camera images are shown above the corresponding normalized voltage change across the micro-counter, with the current value denoted by the vertical dashed line.	71
Figure 4.10 Filtered and post-processed device output displaying voltage response from five microspheres. Microsphere diameters obtained from high speed imagery are shown above each pulse and have an error of $\pm 0.9 \mu\text{m}$	71
Figure 4.11 Relationship between particle diameter and device response. The dotted line shows the theoretical response based on the aperture resistance change equation formulated by DeBlois et al. [59]. Diameter measurements have an error of $\pm 0.9 \mu\text{m}$	73
Figure 4.12 Filtered and post-processed device output. The voltage response due to a single large particle is indistinguishable from the response due to multiple coincident particles. Particle diameters obtained from high speed imagery are shown above each pulse and have an error of $\pm 0.9 \mu\text{m}$	74
Figure A.1 Injection tube position within the inlet pipe (left), and dimensions (right).	90
Figure A.2 Positioning mount with screw-driven adjustment mechanism.	91

Figure A.3 Geometric determination of wire length based on desired position.	92
Figure A.4 Geometric determination of position based on wire length.....	93
Figure B.1 Mixing mechanism. By toggling the current direction (I), the magnetic field direction (B) is flipped, which induces an alternating torque (τ) on the mixing rotor.	96
Figure B.2 Syringe mixing components. The stator is fits over a glass syringe. The star- shaped rotor is placed within the syringe before filling with the microsphere suspension. Quarter shown for size reference.....	98
Figure C.1 Parabolic laminar flow profile within a circular tube as a function of forward velocity and distance from the centerline.	100
Figure D.1 Results from experimental injections overlaid on the simulation injection map. Note the vertical offset in the actual vs expected distribution.	104
Figure D.2 Injection tube support mechanism (left) and dimensions (right).....	104
Figure D.3. Exaggerated beam deflection. Total deflection is shown (green line) along with the deflection caused by the weight of the beam (black line) and by the flexible tube attached to the left end of the beam (blue line). The stars indicate the location of the supports. The circle indicates the tip location.....	122
Figure D.4 Original release position results (left) and corrected positions based on injection tube tip deflection (right).	123
Figure E.1 Custom slide featuring image registration grid. Quarter shown for size reference.	127
Figure E.2 Micrograph of dense microsphere concentration. Arrows indicate overlapping, stacked microspheres.	127
Figure E.3 Initial micrograph with designated areas of interest. The typical analysis area is shown in blue. A smaller, example region is shown in red. Grid borders are visible as blurry black lines on the edges of the image.....	128
Figure E.4 Background subtraction process.	129

Figure E.5 Contrast enhanced image.	130
Figure E.6 Thresholded image before (left) and after (right) morphological smoothing.	131
Figure E.7 Binary image following removal of extreme-sized areas (left) and areas with poor roundness (right).	132
Figure E.8 Image with initial centroid estimates marked in blue. Note the erroneous centroid at the center of the clump indicated by the arrow.	133
Figure E.9 Inverted thresholded image with candidate regions filled before (left) and after (right) morphological reduction.	134
Figure E.10 Composite image subset showing the centroid refinement process, namely: initial centroid estimate (A, in blue), quadrant separation (B), closest pixel detection (C, colored by quadrant), minimization vectors (D), new centroid determination (E, in red), and diameter determination (F).	135
Figure E.11 Initial image overlaid with red circles located at the final. The erroneous region was successfully detect and is marked in green. The largest and smallest diameters found are marked in pink and yellow, respectively.	136
Figure E.12 Full analyzed micrograph. Detected microspheres are denoted in red.	137
Figure E.13 Sample micrograph after analysis. Detected microspheres are denoted in red.	137

Chapter 1

Introduction

1.1 Background and motivation

The evolution of liver tumor treatment has enabled marked improvements in patient survivability and quality of life. New multi-modal treatments utilize a combination of cancer-fighting approaches, such as surgical resection, ablation, chemotherapy, radiotherapy, and localized delivery methods (e.g. transarterial embolization). Despite these advances, the prognosis for patients presenting liver cancer is still poor [1].

The most common form of primary liver cancer (i.e. cancer which originates within the liver) is hepatocellular carcinoma (HCC), which accounts for an estimated 1 million new cases annually [2]. Although HCC is only the sixth most common cancer in the world, it is the third leading cause of cancer related death [3]. The liver is also the second most common organ involved in secondary metastatic tumors, which commonly originate from gastrointestinal, neuroendocrine, skin, breast, lung, and kidney cancers [1], [4] before then

spreading to the liver. 60% of patients with colorectal carcinoma, for example, eventually develop liver disease as the primary cancer site, causing an estimated 80,000 patient deaths annually [2], [5]. For most tumor types (colorectal, breast, lung, kidney, etc.), the presence of liver metastases leads to a poor prognosis and determines life expectancy [1].

The lethality of these liver diseases can be attributed to the tumors' resistance to conventional treatment methods. While surgical resection is the preferred and most effective treatment approach for primary and metastatic liver malignancies, 75 to 90% of patients are precluded from surgical intervention or ablation due to tumor size, location, or complexity [1], [5], [6]. Even after undergoing a successful resection, cancer recurrence rates reach 60%-70% [7]. Similarly, systemic (i.e. full-body) chemotherapy is often rendered ineffective; more than 50% of patients with solid tumors display full resistance to conventional chemotherapy treatments [8].

In contrast to chemotherapy, radiation therapy has consistently demonstrated efficacy for liver tumor treatment [9]. Normal, healthy liver tissue, however, is particularly radiosensitive, suffering radiation hepatitis at a 35 Gy radiation dose, approximately half the estimated 70 Gy dose required to destroy solid tumor tissue [10], [11]. Although external-based radiotherapies are extremely effective in destroying tumor cells, the collateral radiation damage to adjacent tissues and organs can be significant, limiting the adoption of the technology.

Transarterial approaches have been developed for localized delivery of various treatment agents, minimizing damage to the rest of the body. The foundation of these

treatments is transarterial embolization (TAE), a catheterized procedure where embolic agents (typically microspheres) are internally injected into the hepatic vasculature to implant permanently in the terminal vessels of tumors, occluding (i.e. embolizing) the tumor's arterial blood supply. The recommended microsphere diameter (25-35 μm) balances the objectives of enabling particle deposition near the tumor while not allowing passage through to venous circulation [12], [13].

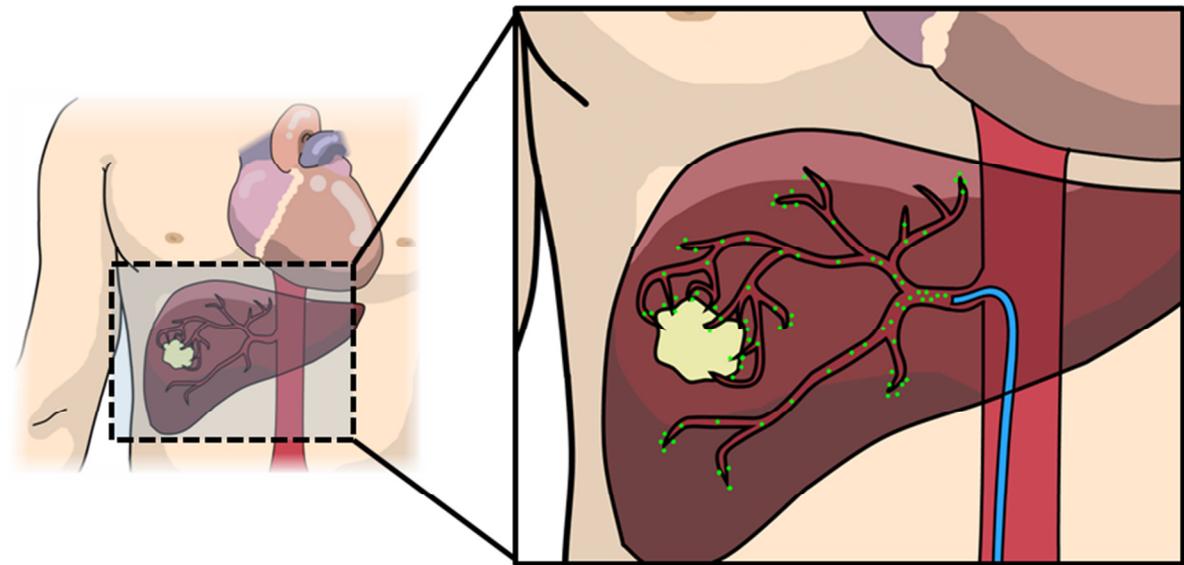


Figure 1.1 Simplified illustration of a transarterial embolization injection.

Transarterial techniques capitalize on the unique anatomic and pathologic features of liver tumors. A dual blood supply is present in the liver; portal venous blood supplies at least 75% of the normal liver requirements with the remaining 25% from the hepatic arterial vessels. Primary and metastatic tumors, on the other hand, receive 80-100% of their blood

supply from the oxygen-rich hepatic arteries. Furthermore, due to the highly angiogenic nature of tumors, there are a higher number of hepatic arterial vessels surrounding the tumor site compared to normal liver tissue (having generally 3:1 tumor-to-normal liver vessel ratio) [8]. Therefore, any particles released into the hepatic artery will preferentially reach the tumor as opposed to normal tissue. Additionally, larger sized particles ($>10\ \mu\text{m}$) will become permanently lodged in the vessels, as they are too large to pass through the capillary bed ($8\text{-}10\ \mu\text{m}$) and into circulation. Figure 1.4 shows angiographic images during a hepatic arterial contrast injection. Due to the increased vascularity and selective blood supply, high concentrations of contrast accumulate around the tumor and aid identification.



Figure 1.2 Tumor localization via digital subtraction angiography; coronal view of a patient with metastatic colon cancer. The three images (A-C) were taken sequentially over 20 seconds during contrast injection in the proper hepatic artery. [14]

An unfortunate consequence of the high vascularization displayed by liver tumors is that achieving 100% embolization is generally not practical. Investigations quickly turned to combination-based techniques, wherein a secondary treatment agent is delivered in addition

to embolic particles. With transarterial chemoembolization (TACE), for example, high dosages of chemotherapeutic agents are introduced into the hepatic arteries in conjunction with microspheres. By delivering the agents directly to the target area, high drug concentrations can be achieved with diminished side effects compared to full body (systemic) chemotherapies [15].

1.1.1 Radioembolization

In recent years, radioembolization (RE) procedures utilizing radioactive microspheres have provided promising results, combining the efficacy of radiation therapy with the selective delivery of TAE. Microspheres infused with a radioactive isotope are injected into the hepatic arteries and implant preferentially in the terminal arterioles of tumors [2], [8], [10], [16], [17]. Though a wide variety of radioisotopes have been investigated experimentally, current commercially-available products are solely focused on Yttrium-90 (^{90}Y) [5]. ^{90}Y particles emit doses of radiation with high energy, but low tissue penetration (2.5 mm mean depth), maximizing the effect on tumor tissue while leaving most normal liver parenchyma unharmed [8]. The 64.2 hr radioactive half-life offers a reasonable allowance for the manufacture and delivery of microspheres while maintaining sufficient radioactivity at treatment time.

Present radioembolization treatments utilize a manually-operated syringe and single-lumen microcatheter to release ^{90}Y microspheres into the hepatic arterial supply with the intent of targeting downstream tumor sites, irradiating and destroying the affected zones. Clearly, the success of this procedure requires that a critical mass of microspheres implant in

vessels at the tumor periphery. Insufficient deposition and coverage typically results in continued tumor growth (Figure 1.2 and Figure 1.3).



Figure 1.3 Partial metastatic colorectal cancer response to ^{90}Y RE treatment. Pre-treatment (left) and post-treatment (right) liver CT images are shown. Arrows denote the tumor locations. [5]

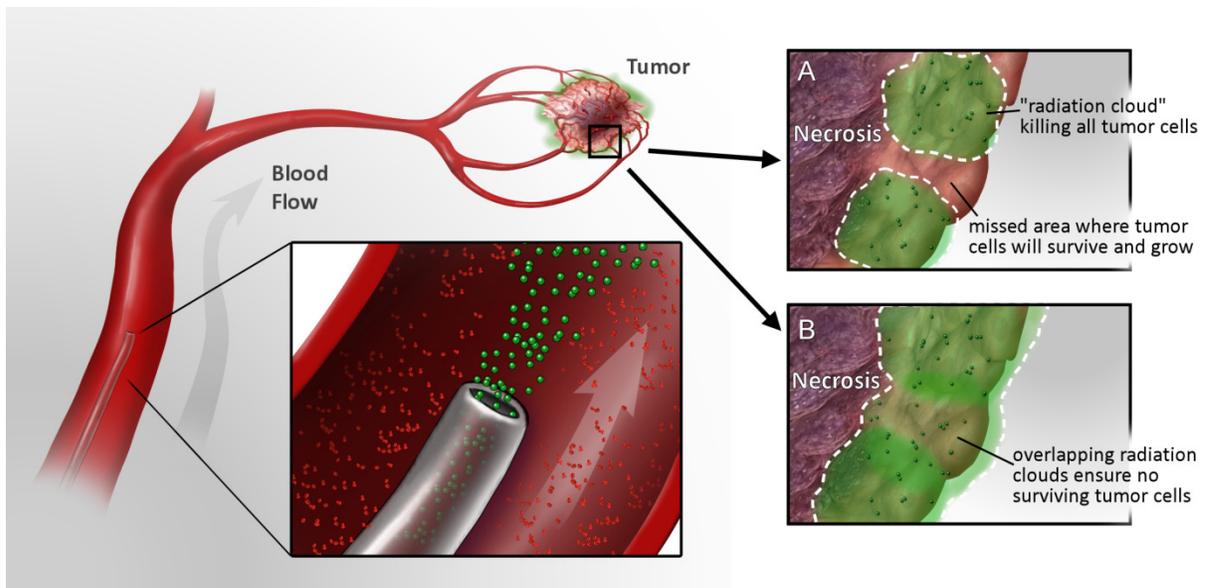


Figure 1.4 Illustration of RE injection and radiation coverage. Two possible microsphere deposition scenarios are shown: incomplete (A) and full tumor coverage (B).

Due to higher incidences of extrahepatic deposition when the catheter is inserted into the proper hepatic artery (PHA), multiple studies recommended placing the catheter as distally as possible into the left hepatic artery (LHA) and/or right hepatic artery (RHA) for either unilobar or bilobar tumor treatments [17], [18]. Anatomical size constraints, however, limit the extent to which the delivery catheter can be inserted into these arteries. The catheter requires periodic verification of placement during the procedure, as the catheter tip can dislodge from the targeted artery. Additionally, if blood flow into the delivery artery decreases to less than the microsphere injection rate, as can be the case when the terminal arterioles embolize with microspheres, the injected microspheres can reflux out of the desired artery and into neighboring branches of the common hepatic. Microspheres that do not travel to the tumor site have the potential to destroy healthy liver tissue via ischemia and/or radiation damage. Microspheres that reach radiation-sensitive nontarget organs, such as the stomach or duodenum, whether from abnormal vessel shunting, microsphere reflux, insufficient extrahepatic artery embolization, or improper catheter placement, can cause mucosal edema and ulceration [2], [17]. Common causes of improper deposition are excessive delivery volume, incorrect infusion speed, and catheter placement too close to gastric or gastroduodenal feeding vessels [5].

Radioembolization (RE) efficacy is clearly influenced by multiple factors, including microsphere radioactivity, penetration depth, and the number of spheres successfully implanted in the tumor periphery [11]. Despite the well documented advances in RE, the lack

of a direct targeting strategy has limited the widespread adoption of this procedure for the treatment of advanced liver tumors.

1.1.2 Model-based targeting

Recent work employing computational fluid-particle modeling of the hepatic arteries has identified a correlation between micro-particle release position and downstream branch distribution, suggesting direct tumor-targeting in RE procedures may be feasible. Kennedy et al. [19] first proposed the strategy by modeling a simplified hepatic geometry and tracking the flow of microspheres released from various upstream locations. The simulation results suggested that, by modeling all geometric and hemodynamic effects, optimized release locations could be found which maximize ^{90}Y delivery to targeted branches (see Figure 1.5). Basciano et al. [20] expanded the approach and investigated transient vs steady flow and particle release time influences. Both of these studies were primarily computational in nature; no experimental validation of directed targeting in a hepatic artery model was presented.

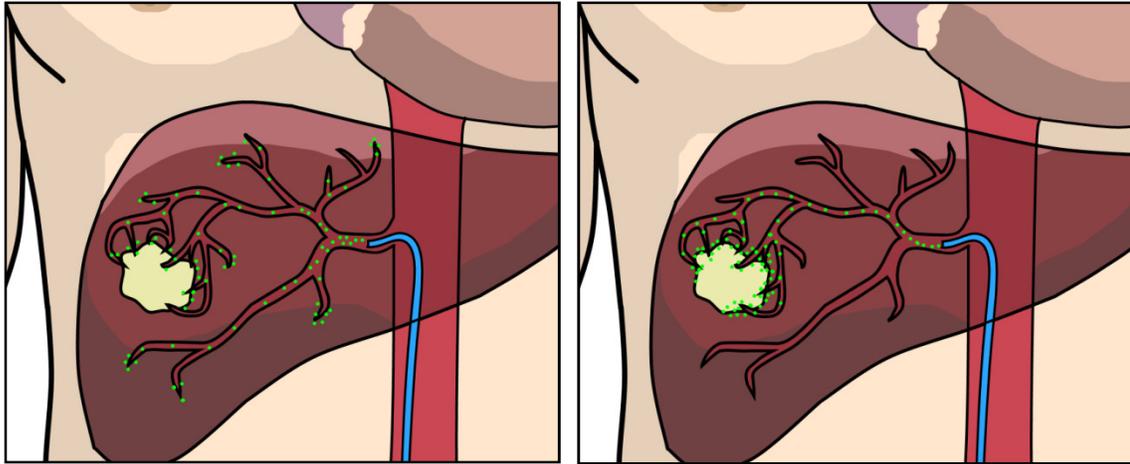


Figure 1.5 Illustrative comparison of non-targeted (left) and targeted (right) delivery of microspheres to a tumor site.

In order to apply these simulation strategies in a clinical treatment, complete and accurate data from all aspects of the procedure are required, including patient specific anatomy and hemodynamic conditions, and all physical properties of the microspheres. Patient specific data is straightforward to obtain, as patients routinely undergo comprehensive imaging to map the location of tumor sites. Similarly, hemodynamic data (flow, pressure) can often be obtained during imaging or with the use of clinical measurement transducers in the pre-treatment stage. Limited information exists, however, regarding the physical properties of the treatment microspheres, such as the microsphere concentration in solution, size distribution, and density.

1.2 Specific aims

The specific aims of this research seek to fill fundamental knowledge deficits regarding advanced RE treatment methodologies. Before new targeting strategies may be adopted clinically, experimental investigations are warranted examining methods and materials used in the procedure. The results presented here also have the potential to positively affect current clinical treatment methods.

The specific aims are as follows:

1. **Validate computational targeting strategy in a controlled laboratory environment.**

Before the CFD-based targeting strategy transitions to a clinically feasible methodology, the fundamental principle must first be validated in a laboratory environment. Achieving predictable targeting in a controlled environment lays the groundwork for adapting this technique for eventual clinical application.

2. **Characterize unknown physical properties of RE microspheres.**

Complete physical characterization of the treatment microspheres will be required for accurate modeling/targeting in a clinical environment. Additionally, having complete knowledge of the microsphere dose can only help to improve the treatment and analysis of patients receiving RE-based therapies.

3. **Develop potential method for clinical detection of microsphere dose.**

Current RE procedures do not easily permit physicians to determine the delivered dose of microspheres. Should microsphere physical properties prove to be highly

variable from vial-to-vial, a new method will be required to measure the dose in real-time.

1.3 Organization

Chapter 2 documents the validation of the simulation-based targeting theory and determines its feasibility for future clinical use. An experimental fluid circulation model is constructed to recreate flow conditions specified by the simulation protocol. A scaled model of a generalized hepatic system with a single inlet and five outlet branches is fabricated to replicate the hemodynamics of diseased livers. Microspheres are released from fixed locations within the model inlet cross-section and the resulting output distributions recorded. Computational fluid-particle transport simulations are conducted with identical parameters and the microsphere distributions compared with experimental-derived results.

Chapter 3 investigates resin microsphere physical properties. Resin ^{90}Y microsphere treatment vials, previously used in clinical RE treatments, are analyzed using multiple particle measurement techniques. Specifically, microsphere concentration, size distribution, and density are examined.

Chapter 4 describes the development of a low cost, simply fabricated micro-fluidics based sensor and demonstrates its use for microsphere detection in solution.

Chapter 2

Experimental targeting

2.1 Introduction

Continued favorable results over the last decade have propelled clinical adoption of radioembolization (RE) procedures for localized treatment of primary and secondary liver malignancies. Despite the well-documented successes of hepatic arterial particle delivery, the lack of effective targeting strategies has limited the widespread adoption of RE for advanced and early stage liver tumor treatments.

Recent work involving 3D computational fluid dynamic (CFD) modeling has investigated direct tumor-branch targeting. Kennedy et al. [19] developed a steady-state CFD model of hepatic blood flow and microsphere transport through a generalized hepatic geometry to determine optimal particle release positions for direct tumor targeting. This model was further refined by Basciano et al. [20]. Simulation results indicate that downstream microsphere branch distribution can be controlled by regulating the upstream

microsphere release location (within the cross section of the hepatic artery). An illustration of this directed targeting is depicted in Figure 2.1.

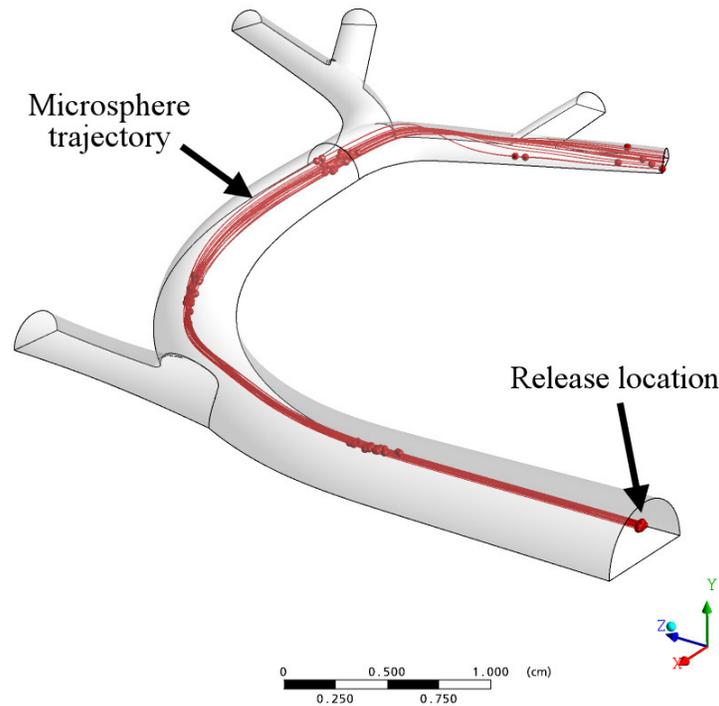


Figure 2.1 Directed microsphere targeting in hepatic artery simulation model. Image modified from original created by Christopher Basciano.

This chapter describes an experimental investigation seeking to verify the CFD approach for predicting microsphere distributions and examines whether such procedures may be feasible in clinical RE treatments. Multiple microsphere injections were made in a physical hepatic model to determine the effects of microsphere release location on downstream branch distributions, and to examine how the results compared to corresponding CFD model predictions.

2.2 Methods and materials

2.2.1 Model construction

A generalized hepatic artery model was constructed out of transparent urethane resin to facilitate the experimental aspects of this research. The model was first presented in Kennedy et al. [19] and later described in Basciano et al. [20], and was based on published dimensional data for common hepatic artery (CHA) and PHA diameters by Carlisle et al. [21] (6 mm and 4 mm, respectively) and RHA diameters by Han et al. [22] (2.68 mm). Approximations for other vessel diameters (daughter vessels of the GDA, RHA and LHA) were based on the clinical experience of a practicing radiation oncologist (Dr. Andrew S. Kennedy, Nashville, TN, USA). The full geometric details of the model can be found in Basciano et al. [20].

Hepatic artery structure varies widely from patient to patient. The model is a generalized hepatic artery configuration and consists of a single inlet and a series of bifurcations, resulting in five daughter vessel outlets that all lie in a single plane (Figure 2.2). In Basciano et al. [20], the model features are given labels representative of the CHA, PHA, gastroduodenal artery (GDA), LHA, RHA, and corresponding daughter vessels.

To aid in fabrication and experimental evaluations, all model dimensions were first scaled by a factor of 4.0 from their anatomic equivalents, and then scaled again so that the model inlet inner diameter (ID) matched the nearest Nominal Pipe Size standard (2" NPS PVC, 2.62 cm ID), resulting in a total scaling factor of approximately 4.4.

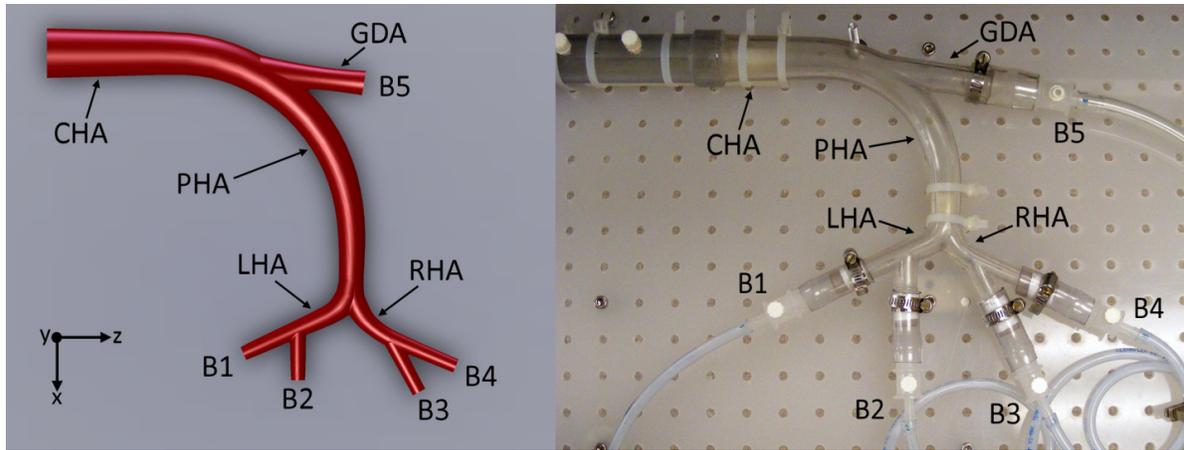


Figure 2.2 Generalized hepatic artery model; CAD model showing major branches (left) and photograph of 4.4X model used for experiments (right).

The model inlet was connected to 1.6 m of straight, transparent PVC pipe (2.62 cm ID) to ensure fully developed flow (i.e. Hagen-Poiseuille flow) at the inlet. An open supply fluid reservoir was connected to a computer-controlled gear pump (Greylor Co., Cape Coral, FL, USA), which, in combination with a custom paddlewheel-style flowmeter (Figure 2.3), was used to regulate steady flow rates through the model.

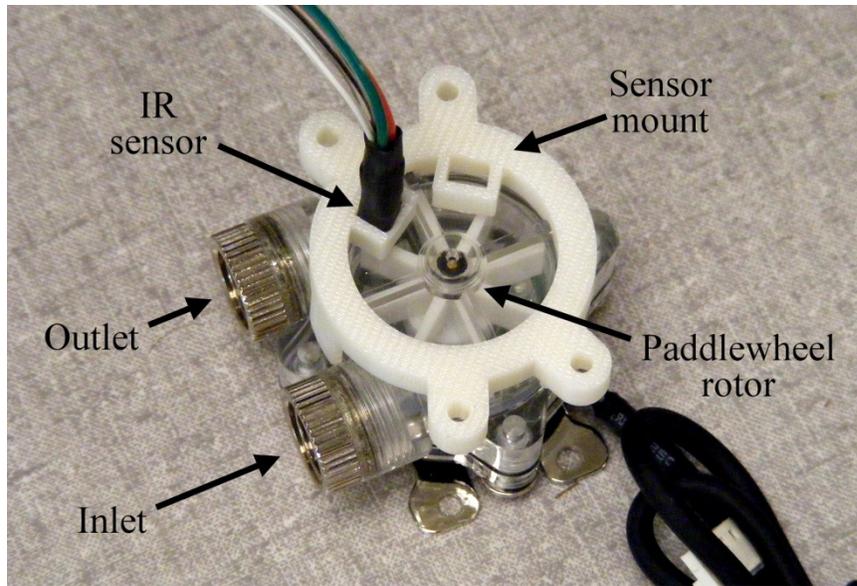


Figure 2.3 Paddlewheel flowmeter with attached optical IR sensor.

To achieve sufficient measurement resolution, a commercially-available, paddlewheel-style flowmeter (INS-FM16, Koolance, Auburn, WA, USA) was enhanced via the addition of an optical infrared (IR) sensor. The sensor was mounted to the flowmeter to sense each passing blade through the transparent casing, outputting a digital pulse (and increasing the measurement resolution by 8X). This optical sensor was connected to a microcontroller (Atmel ATTINY85), which implemented a closed-loop, feed-forward control strategy based on detected flowmeter pulse frequencies and target frequencies set by the host PC (via analog input). Target frequencies were calibrated by a series of steady flow experiments using a graduated cylinder and stopwatch. The PWM-based control output was connected to a half-H-bridge circuit which powered the gear pump. A diagram of this control strategy is shown in Figure 2.4.

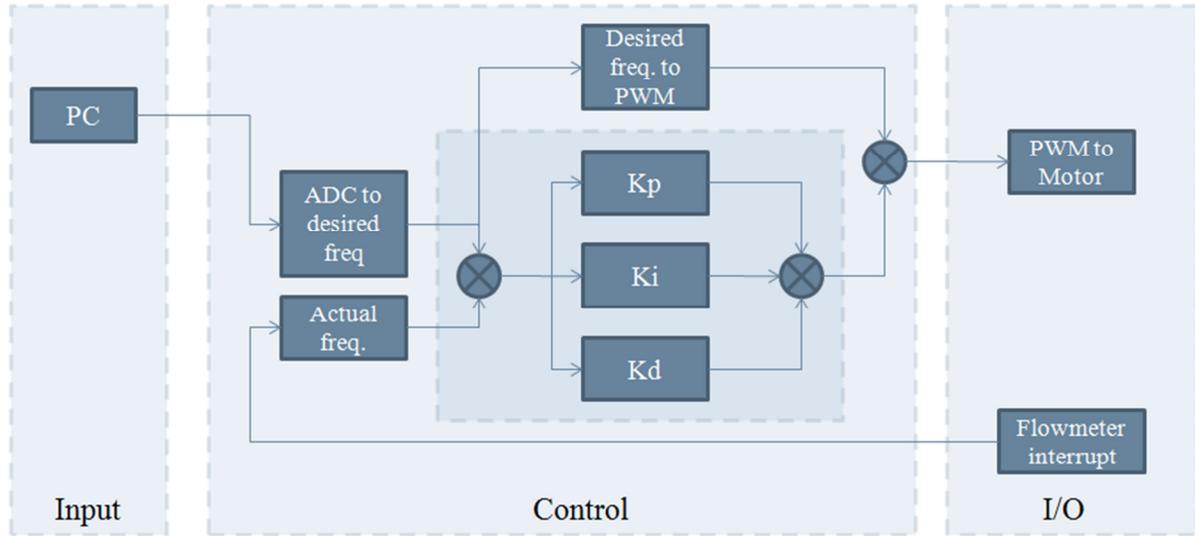


Figure 2.4 Feed-forward control topology used to regulate flow

To release microspheres at the proper location within the model inlet, a 72.5 cm section of hypodermic tubing (15 gauge, 1.37 mm ID) was mounted within the inlet pipe using two custom positioning mounts located 30.5 cm apart. This rigid tube was connected, via flexible tubing, to a computer-controlled syringe pump. Each positioning mount consisted of three stainless steel wires (0.1778 mm diameter) radially attached at 120 degree intervals to the injection tube. The distal portions of these positioning wires were routed through small holes in the inlet pipe, and terminated at a screw-driven positioning system (± 0.3175 mm resolution) (Figure 2.5). Vinyl patches were secured over each hole to prevent leakage, while still allowing movement of each positioning wire. Because the geometry of each mount was precisely known, any desired position in the injection plane could be achieved by rotating the inlet pipe and positioning system and appropriately adjusting the lengths of each of the six

support wires, as shown in Figure 2.6. For a detailed explanation of this positioning process, refer to Appendix A.

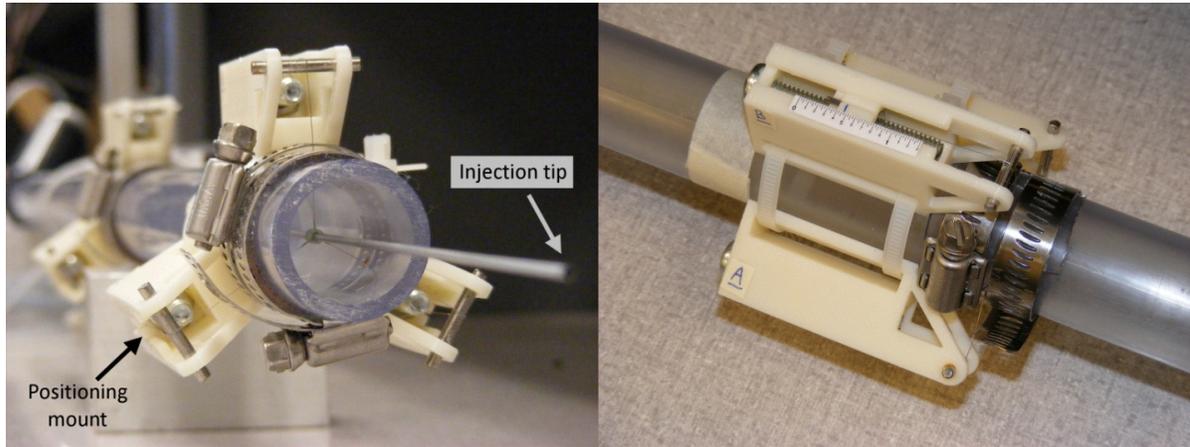


Figure 2.5 Injection tube positioning system.

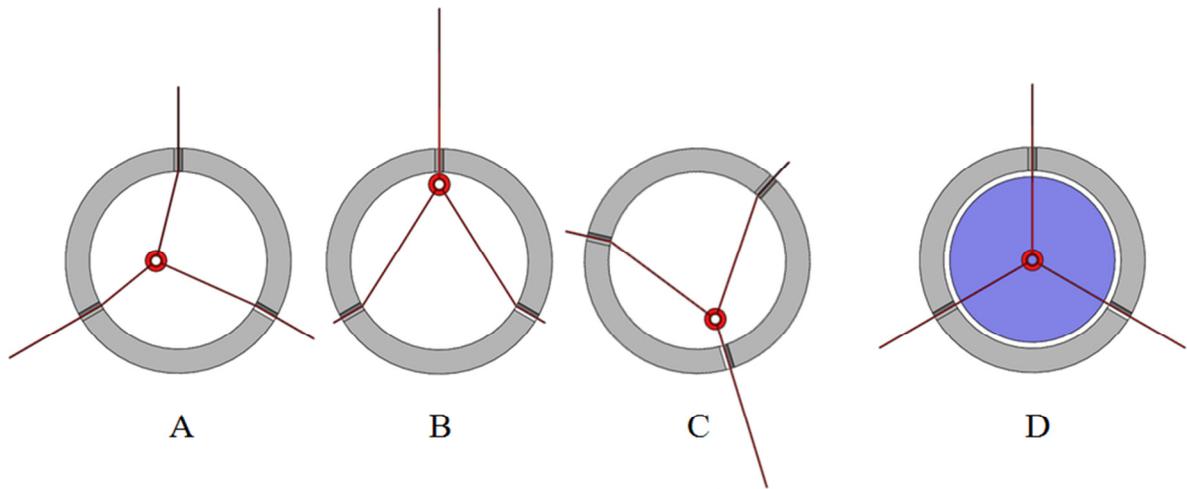


Figure 2.6 Cross-section view of inlet pipe (grey) with inserted injection tube (red). A-C depict example release location positions (C also includes a rotation). The blue area in D denotes the feasible positioning region.

The injection tube mechanism was connected via flexible tubing to a custom computer-controlled syringe pump (Figure 2.7). This microcontroller-based syringe pump allowed for precisely controlled injection profiles. To promote constant microsphere injection concentration and discourage microsphere settling, a magnetic based mixing mechanism was developed and attached to the injection syringe. The design and function of this magnetic mixing system is described in further detail in Appendix B.

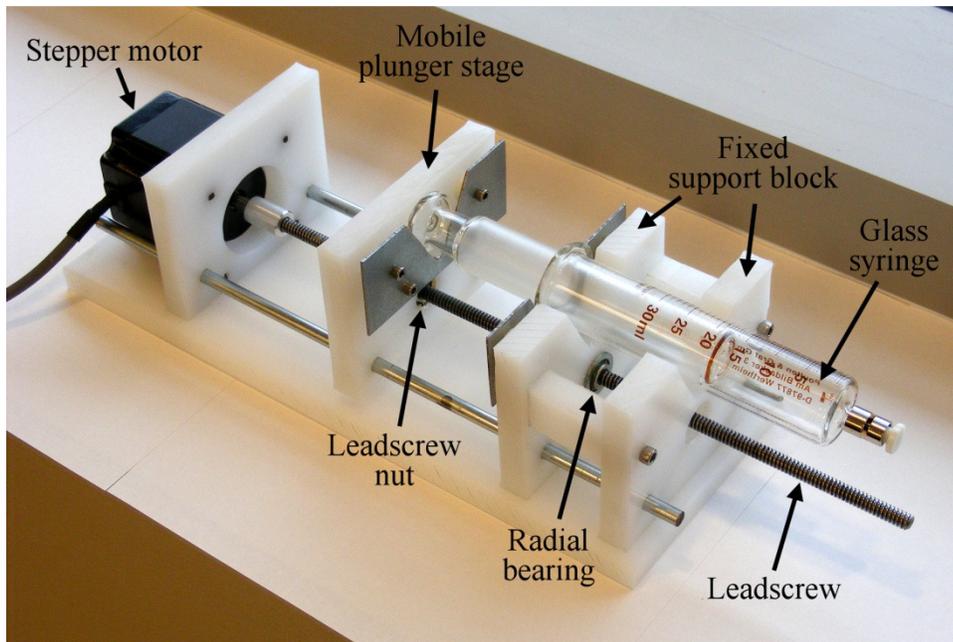


Figure 2.7 Custom computer-controlled syringe pump

Each model outlet was connected, via flexible tubing, to an open collection reservoir, where the fluid was recirculated back to the supply reservoir. External pinch valves allowed

the fluid resistance for each branch outlet to be precisely adjusted. A schematic of the entire system is shown in Figure 2.8.

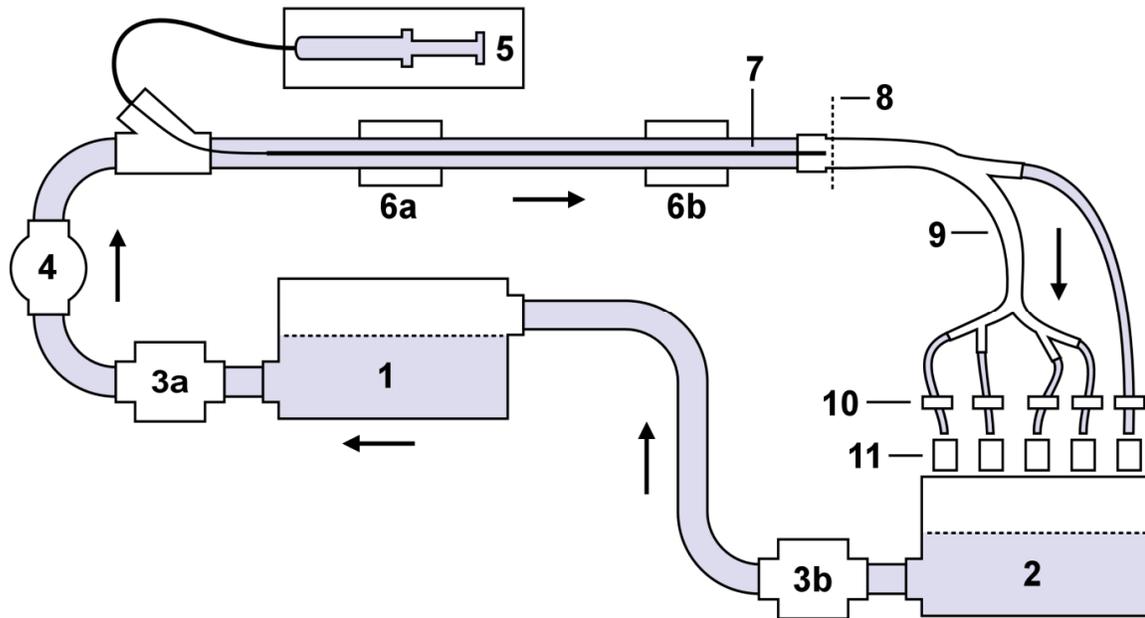


Figure 2.8 Schematic of system components: supply reservoir (1), collection reservoir (2), gear pumps (3), flowmeter (4), syringe pump (5), positioning mounts (6), injection tube (7), injection plane (8), hepatic artery model (9), branch resistance valves (10), and microspheres filters (11). Arrows indicate the direction of flow.

2.2.2 Experimental parameter determination

To scale the modeled flow conditions to a physiological equivalent, published CHA blood flow rates from patients with liver tumors were utilized and are shown in Table 2.1.

Table 2.1 Published blood flow rates in patients with diseased livers

No of Patients	Artery	Measurement type	Mean Flow (mL/s)	error (+-)	Source
26	CHA	Duplex Doppler Ultrasound	13.55	6.42	[23]
42	CHA	Duplex Doppler Ultrasound	10.58	6.07	[24]
19	CHA	Color Doppler Imaging	6.03	1.78	[25]
28	CHA	Doppler Ultrasound	5.72	1.78	[26]
Total number of patients			115		
Weighted Mean Flow (mL/s)			9.32		

The average physiological flow rate of Table 2.1 (9.31 mL/s) was scaled to match the experimental fluid conditions. This scaling utilized the Reynolds number, which is the dimensionless ratio of inertial and viscous forces within a fluid. Assuming a circular vessel cross section, the Reynolds number is,

$$Re = \frac{4\rho Q}{\pi\mu d} \quad (1.1)$$

where ρ , Q , μ , and d are the fluid density, volumetric flow rate, dynamic viscosity, and vessel diameter, respectively [27]. By ensuring the Reynolds number of the physical experiments (using water as the working fluid at 4.4x scale) matched the Reynolds number prescribed by physiological data (with blood at anatomic scale), fluid conditions within the model and its effects on particles within the flow were dynamically similar to those found within the body.

The representative Reynolds number for the CHA of diseased livers is 608, and was calculated using the weighted average mean flow in the CHA (9.32 mL/s) listed in Table 2.1,

the CHA vessel diameter of 6.0 mm reported by Carlisle et al. [21], a blood density of 1.06 g/cm³, and blood viscosity of 0.0345 g/cm/s). Actual Reynolds numbers for the physical experiments ranged from 600-700, due to variations in ambient temperature (and thus water density and viscosity) during each trial.

2.2.3 Simulation design

Using the computational model and methodology described previously [19], [20], simulations were performed using a commercial finite-volume software (ANSYS CFX v12.1, ANSYS Inc., Canonsburg, PA) with the same model dimensions and flow characteristics as the physical experiments.

For the CFD simulations, water was modeled as an incompressible Newtonian fluid, with an inlet Reynolds number of 650. At the inlet, a parabolic velocity profile was specified with a constant area-averaged velocity. At the terminating ends of outlet branches B1-B5, a specified mean pressure (100 mmHg) across the boundary plane was enforced. This pressure was chosen as an approximation of physiological conditions.

A spatial particle release map based on steady flow conditions was generated using a one-way coupled Eulerian-Lagrangian method. First, the Eulerian fluid model was applied to determine the flow field at all node locations within the computational domain. This flow field was then used to calculate the Lagrangian particle motion. Particle transport was modeled using Newton's second law of motion and included drag (F_D), fluid pressure (F_P), and gravitational forces (F_G). Following the justification given in Basciano et al. [20], all

other forces, including virtual mass forces, Saffmann lift forces, and Bassett history forces were neglected.

That is,

$$\begin{aligned}
 m_p \left(\frac{dv_p}{dt} \right) &= F_D + F_P + F_G \\
 F_D &= \frac{\pi}{8} \rho_f d_p^2 C_D (v_f - v_p) |v_f - v_p| \\
 F_P &= -\frac{\pi}{6} d_p^3 \nabla p \\
 F_G &= \frac{\pi}{6} d_p^3 (\rho_p - \rho_f) g \\
 C_D &= \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687}) \\
 Re_p &= \frac{|v_f - v_p| d_p \rho_f}{\mu_f}
 \end{aligned} \tag{1.2}$$

where $m_p, v_p, \rho_p, d_p, Re_p$ are a particle's mass, velocity, density, diameter, and Reynolds number, and v_f, ρ_f, μ_f are the fluid velocity, density, and dynamic viscosity. The drag coefficient, C_D , uses the Schiller-Naumann empirical correlation between viscous and inertial regimes.

Walls were modeled as smooth, rigid surfaces, and no-slip conditions were imposed. Neglecting inter-particle forces and the effects of the particle motion on the flow field has been shown to be a plausible simplification in dilute suspensions [28], [29], thus microspheres were modeled as non-interacting, neutrally buoyant, spherical particles.

To construct the release map, 10,000 particles (115 μm diameter, approximately 4.4 times the size used clinically) were uniformly distributed across the inlet plane of the model and introduced into the fluid with a zero-slip velocity. By tracking the paths of each particle, a graphical release map was constructed which visually correlates a particle's exit location to its initial release position in the injection plane.

The flow distribution at the outlet branches of the model, as computed by the Eulerian fluid model, was prescribed for the physical experiments, and is shown in Table 2.2.

Table 2.2 Simulated and experimental flow distribution

Branch	Simulated (Re=650)	Experimental* (Re=600-700)
B1	11.0%	10.4%
B2	8.4%	8.4%
B3	11.6%	11.7%
B4	9.7%	9.7%
B5	59.2%	60.1%

*Average flow distribution for experimental trials

2.2.4 Experimental microsphere injection

The hepatic artery model reservoir was filled with room temperature tap water (20-25°C).

Two drops of detergent were added to the circulating fluid to act as a surfactant to reduce microsphere hydrophobicity and prevent aggregation. A woven nylon mesh filter (88.9 μm opening) was installed in the supply reservoir to pre-filter any stray particles from the circulating fluid. Steady flow was established at a mean flow rate of 12.83 mL/s, which corresponds to a Reynolds number of 600-700 at the model inlet, depending on ambient

temperature. Individual branch resistances were adjusted until the flow distribution closely matched that specified by the fluid simulation (see Table 2.2). Because the flow rate and corresponding Reynolds number are dependent on the pressure gradient in the model (not absolute pressure), only the flow distribution was matched in the physical experiments. Time-averaged flow rates were verified by collecting branch outputs for 60 seconds using a graduated cylinder with a stated measurement error of ± 2.5 ml.

Polypropylene microspheres ranging in size from 106 μm to 125 μm with densities of 0.9-1.1 cm^3 (Cospheric, Santa Barbara, CA) were added to water to create a microsphere suspension of 3.33 mg/mL. 30 mL of the microsphere solution was then drawn into a 30 mL glass syringe and loaded into the computer-controlled syringe pump. The injection rate of the syringe pump was set so that the microspheres exited the injection tube with a velocity equal to, or slightly less than, the surrounding average fluid velocity at the injection tube outlet. For a mean flow of 12.83 mL/s, this corresponded to an average injection rate of 0.04 mL/s. For a detailed explanation of how these injection rates were calculated, refer to Appendix C. To ensure sufficient mixing and constant microsphere concentration, the syringe was agitated by magnetic mixer during the entire injection process. After the microsphere syringe was depleted, 20 mL of circulating fluid was injected in a similar manner to purge any remaining spheres from the injection line.

Prior to each injection, pre-weighed dry nylon mesh filters (88.9 μm opening) were inserted between each outlet and the collection reservoir to facilitate removal and quantification of microspheres in each branch. At the completion of each injection test, the

filters were removed and allowed to air dry for at least one day before weighing. Microsphere deposition in mg was calculated as the difference in filter weight before and after injection.

A total of 26 injections were made at various locations within the injection plane.

2.3 Results

The simulation-derived spatial particle release map is shown in Figure 2.9. Microspheres have been color coded to specify their exit location. Experimentally-derived microsphere distributions and corresponding injection locations are listed in Table 2.3.

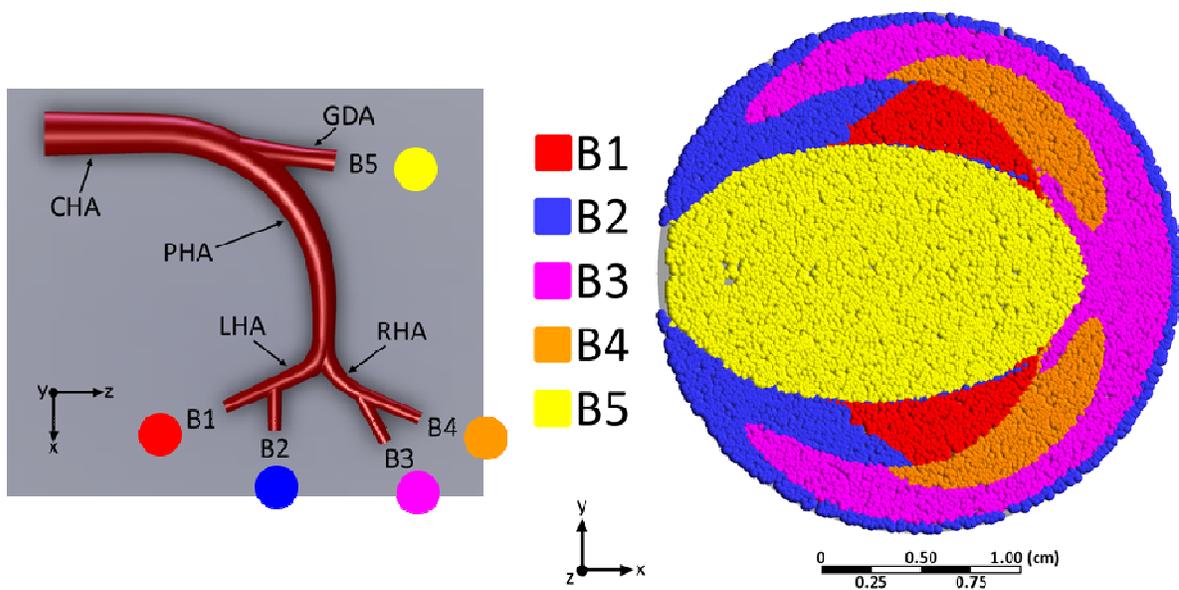


Figure 2.9 Inlet plane particle release map (right) with color coded regions depicting downstream exit locations.

Table 2.3 Microsphere distribution and injection location

Injection	Outlet Branch Microsphere Distribution					Release Location *		
	B1	B2	B3	B4	B5	Error ** (±)	X (mm)	Y (mm)
1	2.1%	0.0%	0.0%	1.0%	96.9%	3.1%	-0.10	5.08
2	0.0%	0.0%	0.0%	0.0%	100.0%	3.1%	-0.10	3.20
3	6.1%	1.2%	1.2%	3.7%	87.8%	3.7%	0.00	0.00
4	53.8%	4.4%	2.2%	0.0%	39.6%	3.3%	1.52	6.99
5	51.6%	0.0%	0.0%	0.0%	48.4%	3.3%	2.11	6.96
6	41.4%	0.0%	4.3%	51.4%	2.9%	4.3%	4.88	8.46
7	0.0%	0.0%	96.3%	3.7%	0.0%	3.7%	9.53	0.00
8	0.0%	71.1%	0.0%	0.0%	28.9%	7.9%	-4.93	7.67
9	0.0%	79.1%	6.0%	9.0%	6.0%	4.5%	-5.69	9.83
10	95.6%	0.0%	0.0%	0.0%	4.4%	4.4%	1.47	8.31
11	0.0%	0.0%	17.3%	82.7%	0.0%	3.7%	6.25	7.47
12	0.0%	0.0%	0.0%	0.0%	100.0%	3.6%	-7.62	0.00
13	0.0%	31.4%	0.0%	0.0%	68.6%	4.3%	-9.83	5.66
14	5.4%	71.6%	4.1%	4.1%	14.9%	4.1%	-1.98	11.18
15	7.3%	0.0%	27.3%	63.6%	1.8%	5.5%	1.55	-7.24
16	2.3%	0.0%	20.5%	77.3%	0.0%	6.8%	7.67	5.61
17	0.0%	1.4%	98.6%	0.0%	0.0%	4.1%	8.08	-4.34
18	0.0%	0.0%	90.0%	10.0%	0.0%	3.3%	9.07	-1.32
19	2.2%	1.1%	88.0%	8.7%	0.0%	3.3%	6.10	-6.83
20	6.3%	2.1%	33.7%	56.8%	1.1%	3.2%	5.59	-5.08
21	8.7%	1.9%	38.8%	50.5%	0.0%	2.9%	3.51	-6.71
22	5.4%	3.3%	29.3%	62.0%	0.0%	3.3%	1.02	-7.49
23	59.0%	1.0%	1.0%	39.0%	0.0%	3.0%	2.29	-4.57
24	87.5%	0.0%	1.0%	11.5%	0.0%	3.1%	0.61	-5.08
25	5.1%	88.6%	0.0%	0.0%	6.3%	3.8%	-3.76	8.76
26	0.0%	88.1%	10.9%	0.0%	1.0%	3.0%	-6.07	-4.09

*Location refers to the coordinates from the center of the inlet

**Measurement error for the microsphere distribution (see section 2.5 below)

Figure 2.10 graphically combines the simulation and experimental results, with pie charts (sized to reflect the injection tube OD) representing experimental microspheres distribution located at their appropriate injection location.

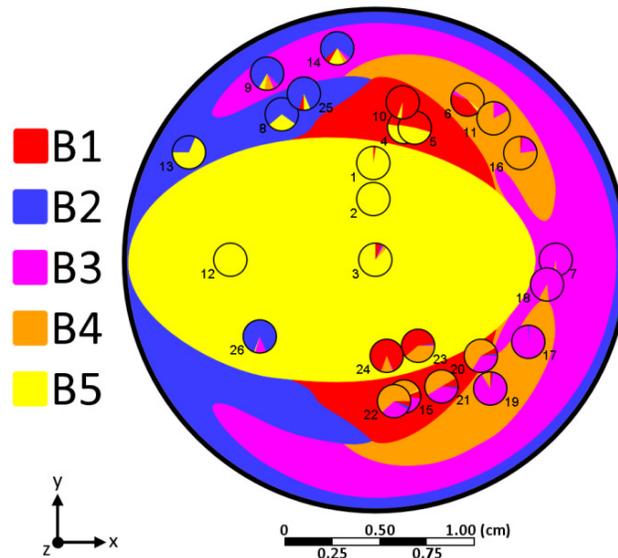


Figure 2.10 Combined simulation release map and experimental results. Pie charts depict microspheres distribution at each injection location.

During injections, a steady stream of microspheres could be observed leaving the injection tube tip and traveling through the model to an outlet (Figure 2.11). Figure 2.12 and Figure 2.13 depict still images from video footage of the LHA and RHA bifurcation, taken during injections 10 and 11, which were only 4.85 mm apart.

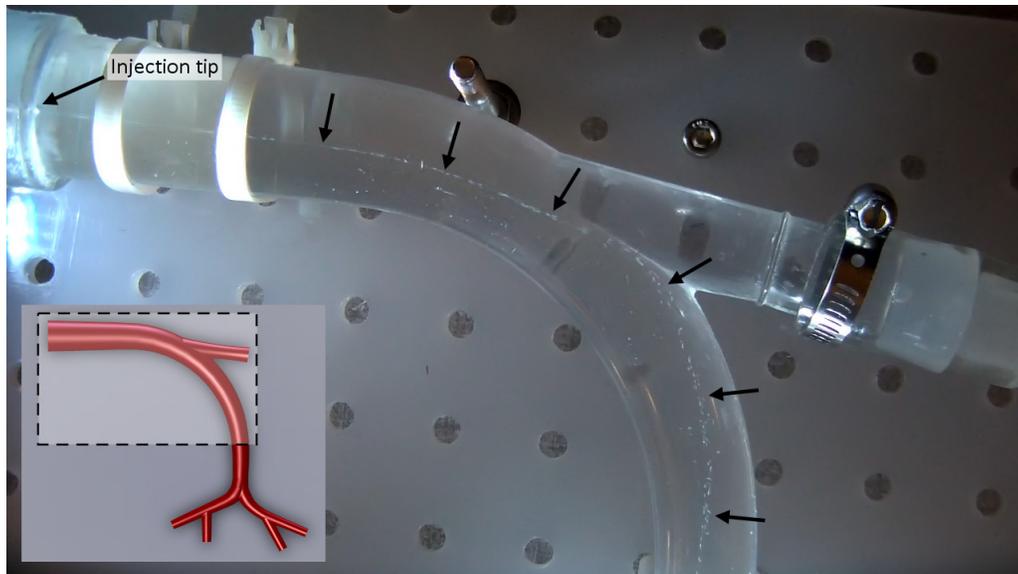


Figure 2.11 Video frame depicting a line of microspheres exiting the injection tube and traveling past the PHA bifurcation during injection 10. The inset graphic indicates the location on the model. Arrows indicate the microsphere stream.

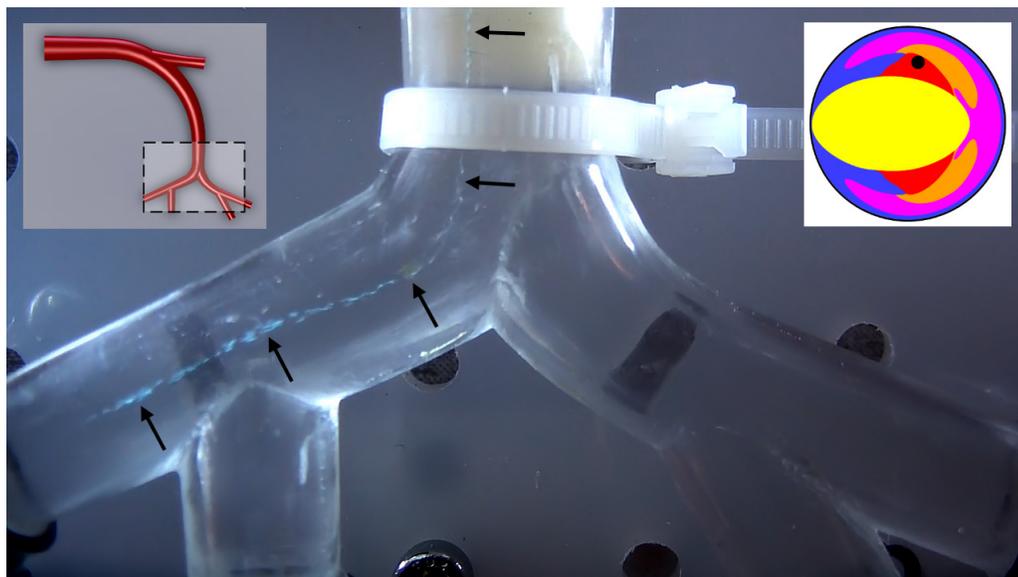


Figure 2.12 Video frame during injection 10. Microspheres are shown traveling through the LHA to branch 1. The inset graphics indicate the location on the model and the injection location. Arrows indicate the microsphere stream.

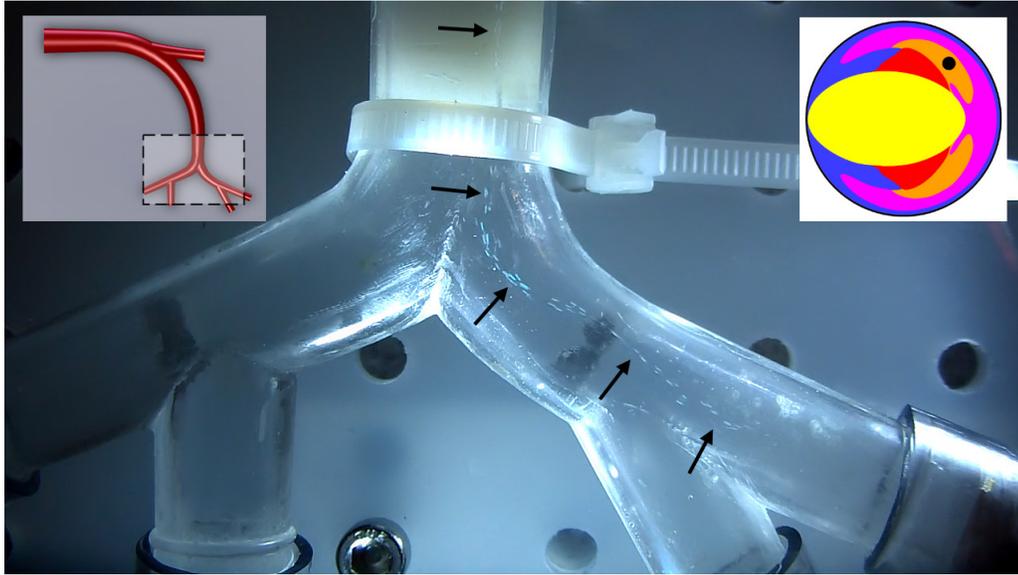


Figure 2.13 Video frame during injection 11. Microspheres are shown traveling through the RHA to branch 4. The inset graphics indicate the location on the model and the injection location. Arrows indicate the microsphere stream.

2.4 Discussion and limitations

For physiologically realistic conditions, the experimental work demonstrated that individual branch-targeting is possible. As shown in Figure 2.10, the experimental microsphere distribution exhibited a clear dependence on injection location that correlated very strongly with simulation results.

To facilitate comparison of experimental and simulation results, the pie charts of Figure 2.10 were rounded to single colors representing the dominant branch for each injection. Although Figure 2.14 shows that only 14 of 26 injection sites (54%) correlated with CFD predictions, a closer inspection reveals that the experimental results are offset

vertically (in the y-direction) and scaled differently than the CFD results. Adding a vertical offset of -1.57 mm, and proportionally scaling the x- and y-coordinates by 15% results in an accuracy of 88% with respect to CFD predictions (23 of 26 distributions matching), as shown in Figure 2.14. These offset and scale values were determined to optimize the release map correlation, and clearly illustrate the similarities between simulation and experimental results.

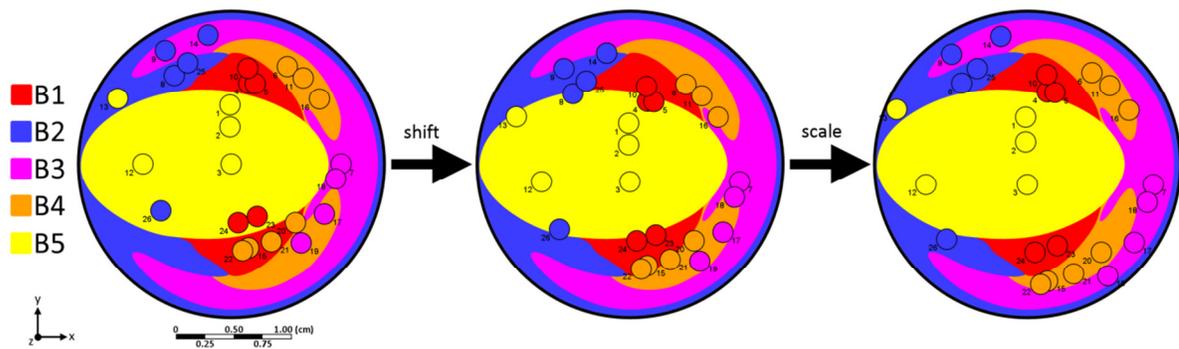


Figure 2.14 Release map comparison after shifting and scaling the experimental injection coordinates.

An explanation for the bias and scaling differences is bending of the injection tube, as it was only supported in two locations along its length. To estimate the potential tip deflection, a theoretical Eulerian beam bending analysis of the injection tube mounting system was performed. Experimentally-determined properties of the hypodermic injection tube (density, 5268 kg/m^3 ; Young's modulus, 140 GPa) were used along with an estimation of the force of the attached supply tubing (12.7 cm of flexible vinyl tubing). Including the effects of buoyancy, this method predicted a maximum tip deflection of -0.514 mm [30],

which is approximately 33% of the offset shown in Figure 2.14. The full analysis of theoretical tip deflection may be found in Appendix D.

A likely source of scaling error could lie in the dimensional tolerances and surface characteristics of the physical hepatic artery model. Although it was designed to feature perfectly circular arterial cross-sections, anisotropic shrinkage of the two-piece arterial model resulted in slightly elliptical cross-sections. Furthermore, the stereolithography process used to fabricate the two-piece model resulted in discrete joints and surface roughness characteristics not present in the computer model. An additional error source could be misalignment of the inlet tube with the model, as a flexible coupling was used to facilitate experimental testing. Nevertheless, zones were found experimentally for each of the five outlet branches that corresponded to >82% microsphere deposition down that branch. Nevertheless, zones were found experimentally for each of the five outlet branches that corresponded to >82% microsphere deposition down that branch.

Achieving a constant injection proved critical to ensuring smooth and consistent microsphere delivery. During the early experimental stages, pulsatile variations in microsphere delivery were observed and ultimately traced to the syringe. At the very low injection rates used in the experiment (approximately 0.04 mL/s), the common disposable plastic syringes that were used initially suffered from barely perceptible stick-slip motion in the rubber plunger, which caused intermittent bursts of higher velocity fluid to be injected from the syringe, and resulted in erratic plums of microspheres injected into the flow stream (Figure 2.15). This phenomenon has also been observed by neonatal infusion providers [31],

[32]. Precision glass syringes, which do not use a rubber plunger, did not suffer from this problem and were, therefore, used for all subsequent experiments.



Figure 2.15 Video frame showing the microspheres plume created by syringe plunger stick-slip. Arrows indicate microspheres.

No data have yet been published regarding the effects of catheter size on computer-simulated injection release maps, so efforts were made to reduce the size and complexity of the experimental injection tube to minimize flow disturbances. The presence of the injection tube itself impacts the flow profile and resulting injection map, and this could account for some of the observed variations between experimental and simulated microspheres distributions.

CFD simulations assumed the microspheres to be of uniform density. The manufacturer specified the particle density of the spheres used in the physical trials as a range of 0.9-1.1 g/cm³; however, it qualitatively appeared as though the density was skewed

to the denser end of this range. Any particles outside of the CFD-specified density would adversely affect comparison to simulation distributions.

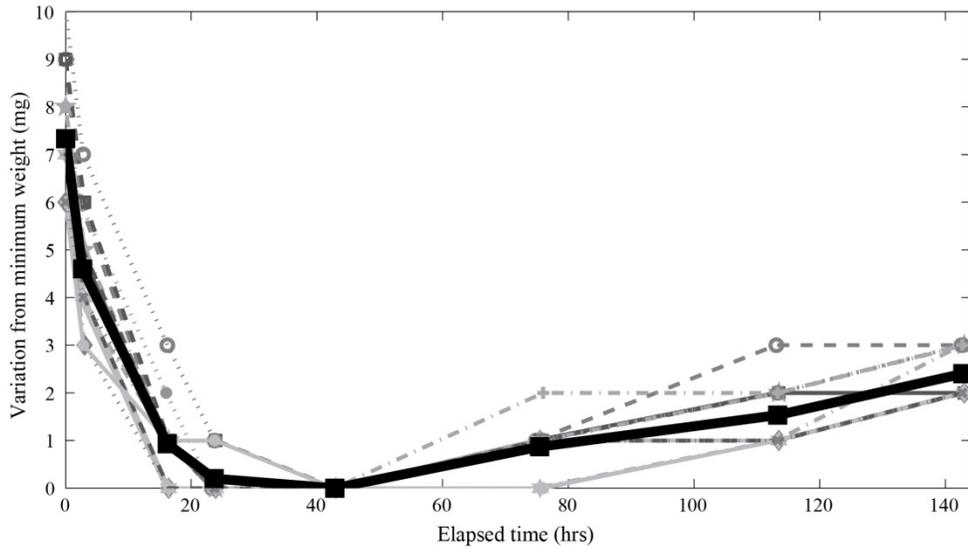


Figure 2.16 Graph of filter weight change over a 6-day period. Filters were wetted at hour 0 and left to open-air dry. The 15-filter average is shown in bold.

Small errors may have been introduced by the method used to quantify microsphere distributions. The scale used to weigh dry filters was accurate to 1 mg, approximately the mass of 1100 microspheres. There were cases where small depositions of spheres were visible on the mesh, but not enough to register on the scale. Additionally, variations in ambient temperature and humidity affected the open-air drying of collection filters. To investigate the effects of open-air drying, and to estimate errors associated with this method, 15 test filters were loaded with varying amounts of microspheres, wetted, then air dried. Filters were weighed regularly during a six day period. 80% of the filters reached their

minimum weight after 24 hours, with the remaining reaching a minimum by the next measurement (Figure 2.16). The maximum variance after the initial 24 hours was ± 3 mg; this magnitude was used as the measurement error in the collected data (see Table 2.3).

CFD simulations assumed the microspheres to be of uniform density (0.997 g/cm^3); however, the microsphere manufacturer specifies a range of particle densities ($0.9\text{-}1.1 \text{ g/cm}^3$) with a distribution that appeared to be skewed to the denser end of this range. CFD simulations with variable particle density reveal only minor variations in release map [20]. Future work will further examine these density effects.

Certainly, the experiments described represent a simplification of the conditions found in the clinical treatment. Flow inside hepatic arteries, which have elastic walls, is pulsatile in nature. Furthermore, the tests fail to capture the embolic effect that occurs during clinical treatments. Loading hepatic vessels with microspheres increases flow resistance, affecting the boundary conditions and associated flow distributions, thus altering the release map.

While an exact 1:1 correlation was not established, the experiments demonstrated that CFD predictions could be a valuable guide for initializing release map zone locations. Once the zones are identified, injections can be performed as dictated by the desired procedure. For the treatment of liver tumors via RE, this could mean targeting a specific branch where a tumor is located, or avoiding a specific branch that may lead to an undesirable location (such as the GDA). In the case of the idealized hepatic geometry, if B5 was indeed the GDA, and it could not be embolized via traditional methods for some reason, locations were still found

that completely avoided the branch – 10 of the injections resulted in 0% deposition down B5, even though B5 carried 60% of the total flow. Naturally, the boundary conditions described here are not valid for clinical cases. Rather, patient-specific data (arterial geometry, flow rates and pressures) would be required to construct corresponding clinical release maps. Additionally, the targeting strategy described requires precise positioning of the injection catheter within the injection plane, a difficult task with current clinical microcatheters.

2.5 Conclusion

The experimental results showed a strong positive correlation to CFD predictions and support the validity of the simulation approach as a technique for achieving controlled microsphere distributions in a known geometry with constant flow conditions. The release location within the inlet plane proved to be an accurate indicator of outlet distribution.

Chapter 3

Microsphere physical properties

3.1 Introduction

A simulation-based targeting strategy was demonstrated and experimentally validated, as detailed in Chapter 1. Because this technique utilizes computational particle-fluid dynamic modeling to determine optimal microsphere release locations, its accuracy is dependent on the accuracy of modeled parameters. This data must be obtained from all aspects of the treatment, which includes the properties of the treatment microspheres. The radioactive properties of commercially-available RE microspheres have been reported in the literature and are routinely measured during the clinical procedure [5], [14], [33]. The physical properties of these microspheres, however, are not accurately known. Critical physical properties referenced in the literature for resin-based microspheres vary depending on the source, and generally lack any citation. Comprehensive knowledge regarding the microsphere dose can only help to advance the RE-based treatment of liver tumors.

For commercially available resin-based RE microspheres (SIR-Spheres, Sirtex Medical Ltd., North Sydney, Australia), no specific particle concentration measurement (microspheres per mL) has been reported in the literature. The total microsphere count and radiation dose per vial has been consistently reported as 40-80 million spheres per 3 GBq, likely a reprint of manufacturer-supplied specifications [2], [6], [10], [34]; this implies a microsphere concentration of 8-16 million microspheres per mL (given ~5 mL solution per vial, SIR-Spheres package insert). Activity level per microsphere is similarly estimated as 50 Bq. Reported microsphere diameters include 35 μm , 30-35 μm , 29-35 μm , $32 \pm 10 \mu\text{m}$, $32.5 \pm 2.5 \mu\text{m}$, and 20-60 μm [5], [6], [34], [35], [36], with few, if any, descriptive shape parameters or characteristics for the distribution. A microsphere density of 1.6 g/mL has been reported, with no indication of the value's origin [5], [8], [10]. Additionally, no information exists regarding the variability of these physical properties between treatment vials.

Accurate knowledge of these parameters may provide valuable information for treatment planning or quantifying treatment efficacy, and may become increasingly more important as new clinical protocols are developed.

This chapter presents the first independent physical property analysis of commercially available resin-based RE microspheres: concentration in solution, size distribution, and density are investigated. This study aims to present a more comprehensive understanding of microsphere doses and may enable better-informed analysis of past results and decisions regarding future treatment plans.

3.2 Methods and materials

A total of 26 resin ^{90}Y microsphere delivery vials (SIR-Spheres, Sirtex Medical Ltd., North Sydney, Australia) were analyzed for microsphere concentration, size distribution, and density. For 25 of these 26 vials, a portion was withdrawn and utilized in a clinical RE treatment procedure prior to this testing. The delivery vial with remaining volume was placed in storage for 3 to 6 months in preparation for the analyses presented here. All vials contained at least 2.5 mL of microsphere suspension. Vial 4 represents an unused sample from a cancelled treatment, and thus was not previously opened.



Figure 3.1 SIR-Spheres delivery vial (without foil seal).

3.2.1 Concentration

The microsphere concentration (million microspheres per mL, hereafter designated as “Msph/mL”) of each vial was analyzed using a commercial coulter counter (Figure 3.2, Model Z1, Beckman Coulter Inc, Brea, CA, USA). After agitation by pipetting, 50 μL

samples were removed from each vial and added to 10 mL of diluent (ISOTON II, Beckman Coulter Inc, Brea, CA, USA). Each suspension was analyzed five times, using a 100 μm aperture, 20 μm threshold, and 0.5 mL metered volume. The sample was further agitated, and the Coulter machine flushed, between each analysis. This process was repeated at least three times per vial, yielding a minimum of 15 measurements per vial. To minimize evaporative losses, vials were tested within three days of removing the foil seal.



Figure 3.2 Beckman Coulter Z1 used for measuring sample concentrations.

3.2.2 Size distribution

Samples from each vial were analyzed via optical microscopy. First, a 200 μL sample from each vial was diluted with 800 μL deionized (DI) water. A 30 μL aliquot from the diluted sample was pipetted onto a glass slide and comprehensively imaged using a digital

microscope (5x combined optical magnification, 2580x1944 resolution, ProgRes C5, JENOPTIK Optical Systems GmbH, Jena, Germany). The captured micrographs were analyzed using image processing techniques to isolate and measure each microsphere diameter. Specifically, images were filtered to remove noise, thresholded to isolate the microspheres and provide approximate centroid location, followed by a least-squares minimization operation to further optimize the microsphere location and diameter. A full step-by-step breakdown of this image segmentation algorithm may be found in Appendix E.

The image analysis process was calibrated, and error quantified, using NIST-traceable microsphere standards (Duke Standards 4k-20, 4k-30, 4k-50, Thermo Scientific, Waltham, MA, USA). First a monodisperse microsphere suspension (50.2 μm diameter) was analyzed to calculate a calibration factor. The calibration factor was then verified through analysis of a polydisperse microsphere suspension (19.6, 29.96, and 50.2 μm diameters). Measurement error was calculated during the verification process and applied to all subsequent analyses.

3.2.3 Density

To determine microsphere density, the remaining contents of each vial were rinsed with DI water, vacuum filtered (Grade 540 filtration paper, Whatman, Maidstone, UK), then dried in a 60 degree C oven for at least 24 hours. The dry microsphere density per vial was then determined using a 2 mL pycnometer following a method similar to that defined in ASTM D854 [37]. The procedure for each sample was as follows: first, the empty, dry pycnometer weight, m_p , was measured and recorded using a laboratory balance (CPA 124S, Sartorius AG, Göttingen, Germany). Next, the dried microspheres were added to the pycnometer and

the total weight, m_{sp} , recorded. Distilled water was then added to the dry microspheres (while agitating the suspension to discourage air entrapment) before recording the filled pycnometer weight, m_{swp} . The pycnometer was then emptied, cleaned, and refilled with only distilled water and the weight, m_{wp} , recorded. The density of the microspheres was calculated as:

$$\rho_s = m_s * \frac{\rho_w}{m_w - m'_w} \quad (3.1)$$

where,

$$\begin{aligned} m_s &= m_{sp} - m_p \\ m_w &= m_{wp} - m_p \\ m'_w &= m_{swp} - m_p \end{aligned} \quad (3.2)$$

and ρ_s and m_s are the density and mass of the microsphere sample, m'_w is the mass of the water added to the microsphere sample, and ρ_w and m_w are the density and mass of the pure water sample. Water temperature (recorded at regular intervals during the testing) and standard pressure assumptions were factored into the water density calculation. Photographs from the density measurement process are shown in Figure 3.3.

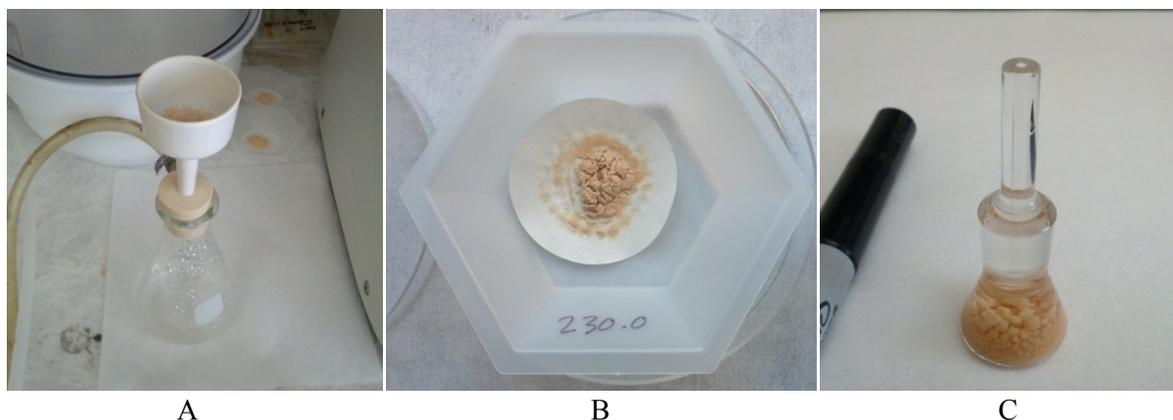


Figure 3.3 Photographs taken during density measurement. Images show vacuum filtration (A), an oven dried sample (B) and a filled pycnometer (C).

3.3 Results

3.3.1 Concentration

To examine sample variability, a random effects model was fit to the data using the restricted maximum likelihood method (REML). Model parameters included the random effects *vial* and *sample* (nested in *vial*). Analysis was performed using the statistical software package JMP (SAS Institute, Cary, NC, USA); the results are listed in Table 3.1. As one vial (vial 14) qualitatively appeared to be an outlier, analysis was also performed excluding that vial's samples. Vial 14 was thoroughly retested on multiple occasions, with consistent results between tests. The per-vial means and standard errors, as computed by the model, are shown in Figure 3.4.

Table 3.1 Statistical model results: vial concentration analysis

Model Parameter	Value (MspH/mL)[†]*
Mean	12.9 (13.4)
Standard Error	0.4 (0.7)
Standard Deviation	2.3 (3.5)
Degrees of Freedom	24.0 (25.2)

Random Effect	Variance Component*	% of Total*
Vial	4.80 (11.49)	87.0 (93.6)
Sample[Vial]	0.46 (0.52)	8.4 (4.2)
Residual	0.25 (0.27)	4.6 (2.2)
Total	5.51 (12.28)	100

* Values in parentheses denote results which include the outlier
[†] MspH/mL = million microspheres per mL

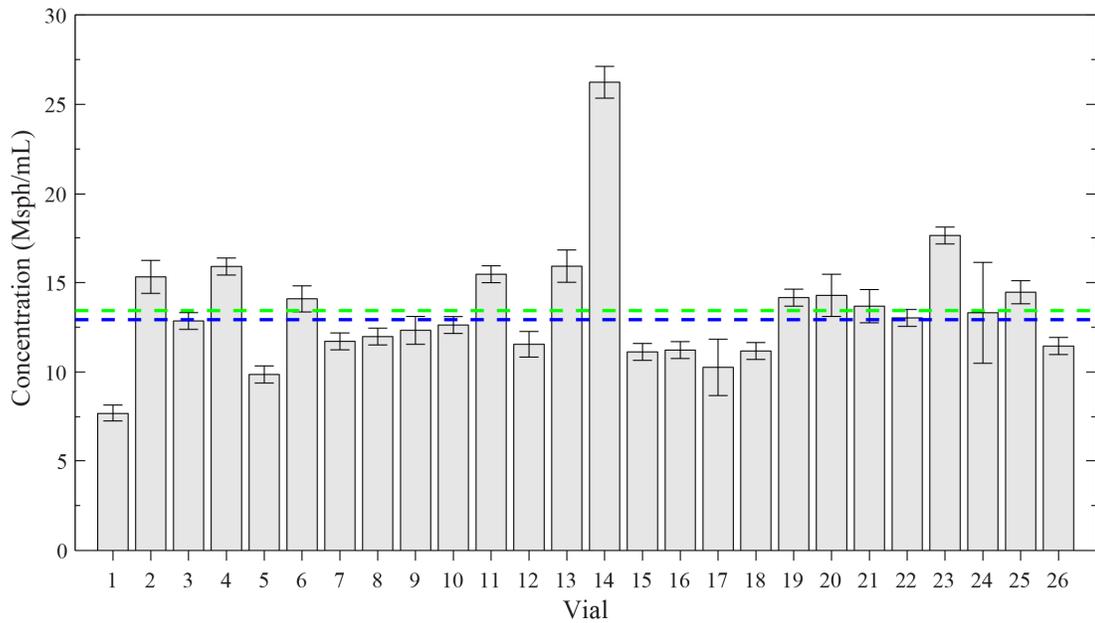


Figure 3.4 Per-vial mean microsphere concentration. Vertical error bars denote the 95% confidence interval for each vial mean. The upper (green) and lower (blue) dashed horizontal lines denote the total estimated mean with and without the outlier measurement (vial 14), respectively.

3.3.2 Size distribution

Approximately 30-70 thousand microsphere diameters were analyzed per vial, yielding over 1.25 million individual diameter measurements. A sample image from this process, demonstrating particle identification and diameter fitting, is shown in Figure 3.5.

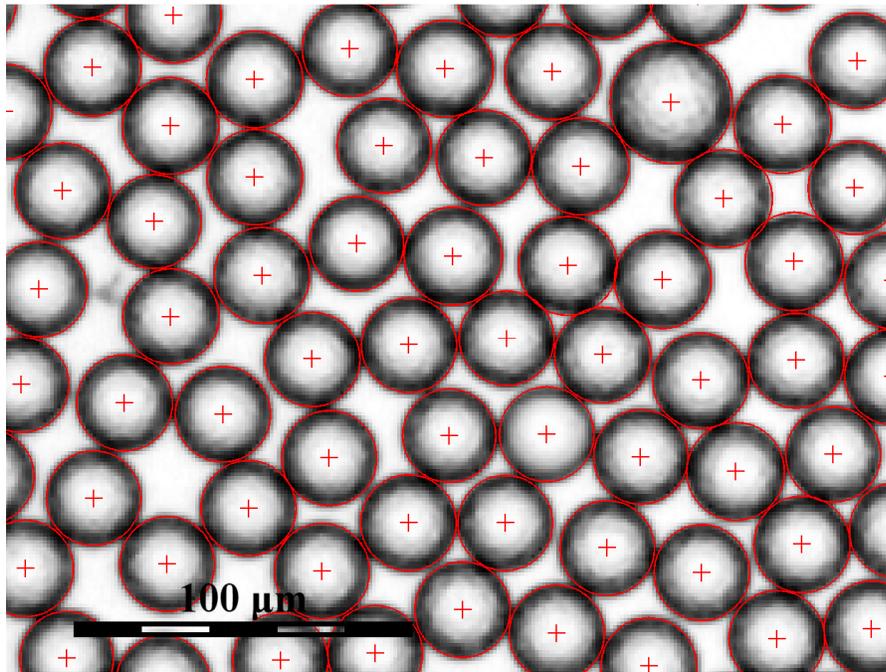


Figure 3.5 Microsphere analysis sample image. Computed microsphere centroids and outlines are indicated in red.

A cumulative probability density plot analysis for the entire dataset indicated a non-normal distribution (see Figure 3.6), being better described by a non-standardized t-distribution with location (μ), scale (σ), and shape (ν) parameters of 29.1, 0.666, and 3.11,

respectively. A histogram of all analyzed diameter samples is shown in Figure 3.7, with t-distribution overlaid.

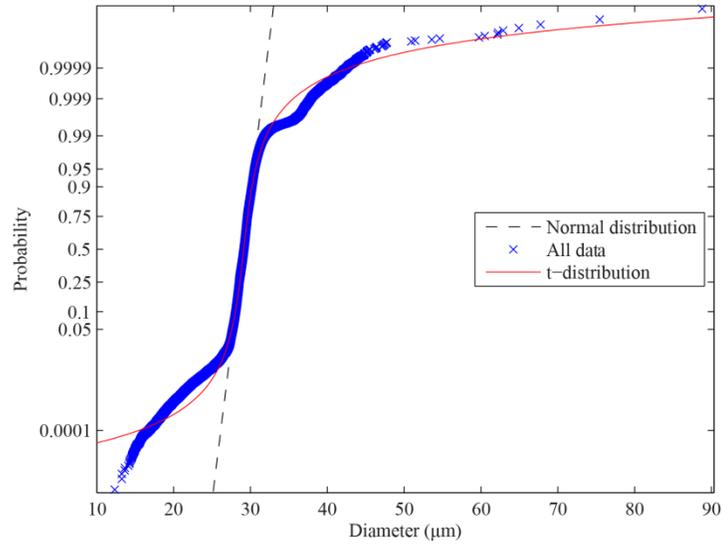


Figure 3.6 Cumulative probability plot for all diameter measurements (in blue). A normal distribution and t-distribution are indicated by the dotted black and solid red lines, respectively.

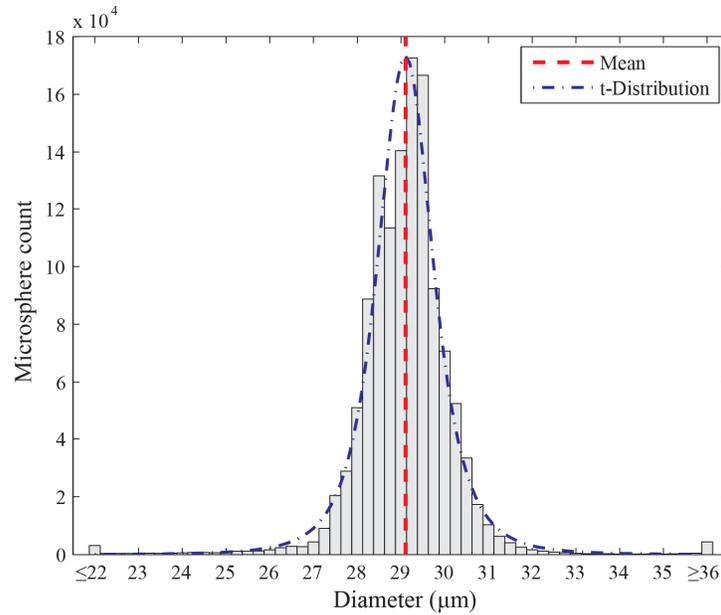


Figure 3.7 Histogram of diameter measurements. The red vertical dashed line (red) denotes the mean diameter. The blue dotted curve denotes a non-standardized t-distribution fit to the data.

As with concentration, a random effects model (using the single random effect *vial*) was fit to the collected data. The statistical model analysis results, along with observed descriptive statistical parameters, are listed in Table 3.2. The per-vial mean diameter and 95% confidence interval, as determined using the fitted t-distribution, are shown in Figure 3.8.

Table 3.2 Statistical model results and descriptive parameters: microsphere diameter analysis

Parameter	Value (µm)	
Mean*	29.1	
Standard Error*	0.8	
Standard Deviation*	1.2	
Degrees of Freedom*	25	
Median†	29.2	
Mode†	29.6	
Max†	88.8	
Min†	12.3	
Skewness†	0.364	
Kurtosis†	36.0	
Total microspheres	1,252,662	

Random Effect	Variance Component	% of Total
Vial	0.26	17.7
Residual	1.18	82.2
Total	1.44	100

* Inferred model parameter

† Descriptive parameter

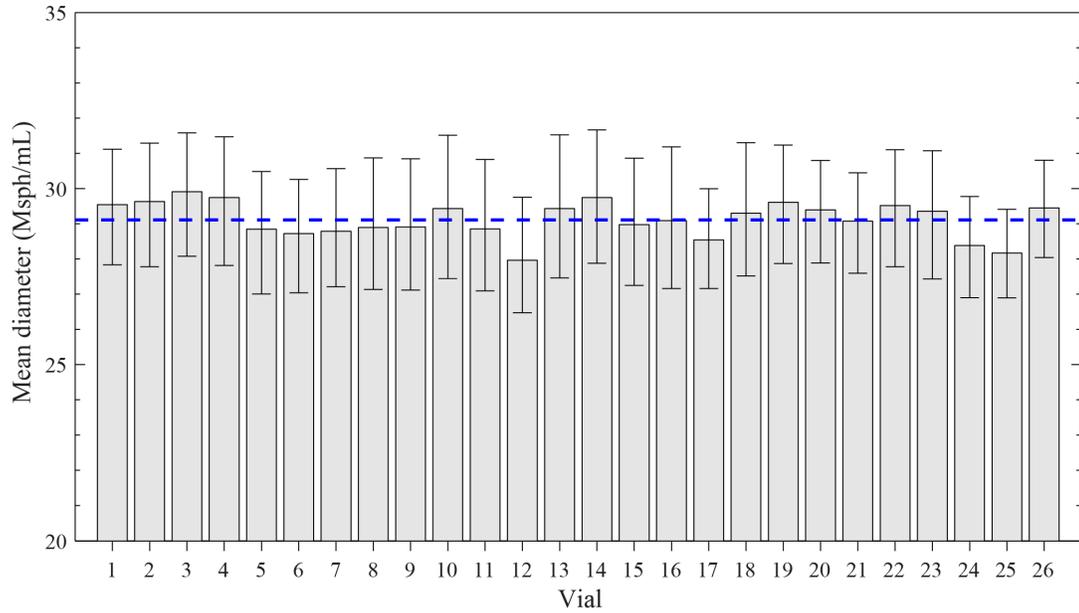


Figure 3.8 Per-vial mean diameter. Vertical error bars denote the 95% confidence interval for each vial distribution, based on a fitted non-standardized t-distribution. The blue dashed horizontal line denotes the total estimated mean diameter.

3.3.3 Density

To ensure sufficient accuracy, each density measurement utilized the entire contents of the corresponding vial, thus, only a single density measurement could be performed per vial. A Shapiro-Wilk normality test [38], using an alpha level of $\alpha=0.01$, indicated that the density measurements were normally distributed. Descriptive parameters for the measured vial densities are listed in Table 3.3, and individual vial measurements are shown in Figure 3.9. Due to the limited per-vial sample size ($n=1$), between-vial density variability could not be examined.

Table 3.3 Descriptive statistical parameters: microsphere density analysis

Parameter	Value (g/cm ³)
Mean	1.57
Standard Error	0.01
Standard Deviation	0.06

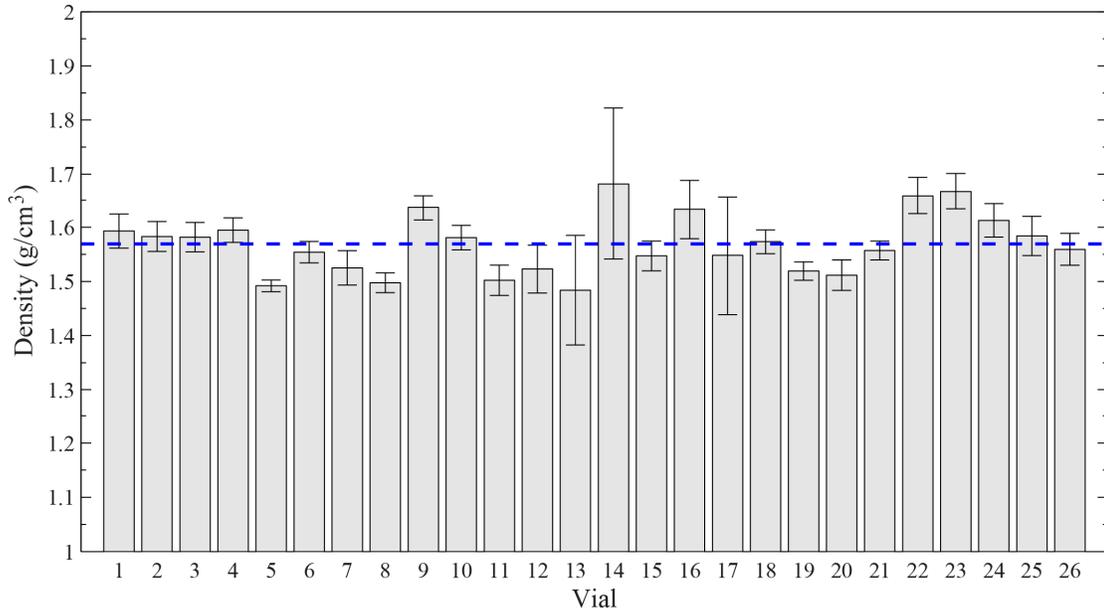


Figure 3.9 Measured density of all 26 vials. Vertical error bars display the 95% confidence interval for each density measurement. The blue dashed horizontal line denotes the total estimated mean density.

3.4 Discussion

Of the investigated physical characteristics, microsphere concentration displayed the greatest variability between vials, having a variance greatly exceeding the expected random statistical error (87.0% vs 4.6%). Even ignoring the outlier (vial 14), the large relative standard deviation of 18% suggests that the concentration of any specific vial cannot be accurately

estimated. This result should not be unexpected, given that microsphere concentration in solution is not a manufacturer specification. Instead, each vial is calibrated to contain a certain activity (3 GBq) at a specified time (typically 48 hrs) post-manufacture. Neither the volume, nor the number of microspheres, is accurately provided on a per-vial basis. Any variability in the manufacturing process prior to vial calibration would directly affect the microsphere concentration. In addition, it is important to consider that these analyses were performed following a clinical RE procedure, where a treatment dose was removed from each vial (apart from vial 4). Even though care was taken when withdrawing each dose, it is impossible to know the degree to which operator error (such as from insufficient mixing prior to sample withdrawal, improper handling during storage, etc.) could have affected the accuracy of these results. Furthermore, the 20 μm detection threshold precluded the counting of small particles, and thus, results may underestimate the true concentration. Based on the fitted t-distribution for sphere size, however, microspheres having a diameter less than 20 μm represent just 0.03% of all diameters predicted statistically.

Current RE procedures do not easily permit physicians to determine the delivered dose of microspheres (i.e. number of delivered microspheres). The problem is two-fold: first, in determining the concentration of the treatment dose, and second, in determining the proportion delivered to the patient. To address the former, commercial particle counters may be used to establish the total vial concentration upon reception; however, radiation contamination and waste management must be taken into consideration. Alternatively, as with this study, the concentration of the remaining dose may be analyzed after the sample has

adequately decayed, although this method precludes pre-treatment knowledge of concentration. Directly determining treatment dose volume delivered to the patient, however, is not currently possible. In the current SIR-Spheres delivery paradigm, clean water is introduced to the treatment vial, passively mixing the suspension and expelling a concentrated microsphere dose. With each injection, and hence, addition of more fluid to the treatment vial, the microsphere suspension becomes increasingly more dilute, impeding the ability to directly determine the delivered dose. If the treatment dose volume withdrawn from the delivery vial is accurately measured, however, the proportion delivered to the patient may be indirectly calculated from the delivered radioactivity percentage (i.e. the ratio of delivered activity to initial activity), a common metric calculated post-treatment. Clearly, changes in the current treatment approach are warranted if individual vial concentration or delivered microsphere dose is required prior to, or during, treatment.

Size distribution was analyzed using optical microscopy, which allowed for additional qualitative inspection of samples. Generally, the size distribution was representative of a $29.1 \pm 0.8 \mu\text{m}$ monodisperse sample with a narrow distribution ($1.2 \mu\text{m}$ standard deviation). The fitted t-distribution indicated a greater occurrence of extreme-sized particles (both large and small) than with a normal distribution (and was slightly skewed towards larger particles). While the largest sphere detected during imaging was $88.8 \mu\text{m}$, during concentration tests there were two occurrences where a large sphere completely obstructed the aperture, indicating a diameter greater than $100 \mu\text{m}$. These occurrences

represent only a small fraction of sizes predicted statistically, but in a treatment where millions of spheres are delivered, multiple large spheres are likely.

The image processing algorithms only considered circular (and thus, spherical) objects in each image; however, vial samples were observed to contain small non-spherical particles in the 10-30 μm range (some appearing as microsphere fragments), as shown in Figure 3.10. Although care was taken to prevent sample contamination, it is not possible to determine whether these particles were initially present in the delivery vials, or were a result of handling and storage prior to, and during, testing. Additionally, microspheres displayed a textured, “bumpy”, surface, which may be a result of ^{90}Y irradiation, as the texture was not observed in bland, untreated resin spheres (SIR-Spheres Training and Demonstration Samples, Sirtex, North Sydney, Australia), as shown Figure 3.11. Irregular texture observations are supported by scanning electron microscopy (SEM) studies undertaken by Kennedy et al. [39]. A preliminary image from this study is shown in Figure 3.12.

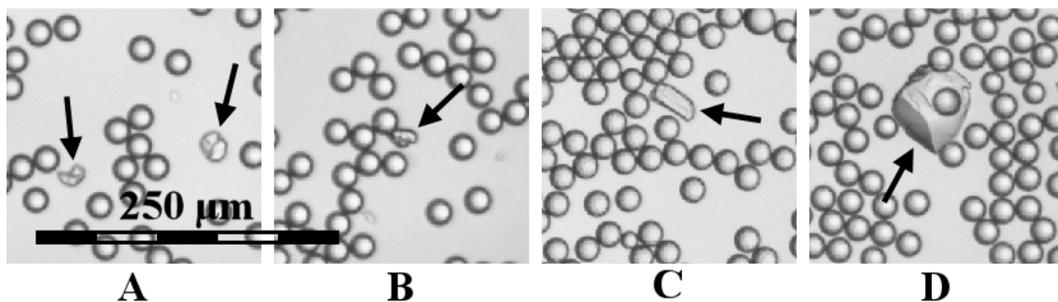


Figure 3.10 Composite micrograph depicting particulate matter present in some microsphere samples. Arrows indicate sphere fragments (images A and B) and non-spherical objects (images C and D).

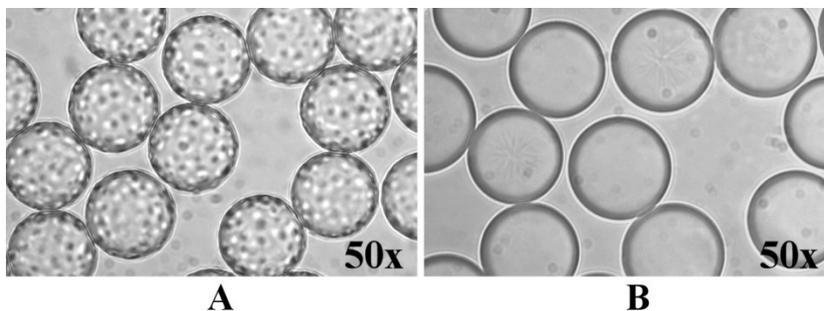


Figure 3.11 Composite micrograph depicting the textured surface of resin ^{90}Y RE microspheres (A), and the smooth surface of bland resin training microspheres (B).



Figure 3.12 Scanning electron micrograph of ^{90}Y microsphere. Image courtesy of Dr. Andrew Kennedy (via personnel correspondence)

Though between-vial density variability could not be formally tested, vials qualitatively displayed similar density characteristics. Per-vial density measurement error depended on the remaining mass of microspheres. As density was measured last, some vials

contained greater amounts of microspheres than others, and thus, for vials with little remaining sample, the associated measurement error was high.

It is not known whether radiation exposure affects the microsphere resin material, or whether material degradation occurs, which could potentially impact the size distribution and density results reported here.

3.5 Conclusion

The lack of specific knowledge regarding the physical properties of resin ^{90}Y RE treatment vials limits the ability to fully evaluate treatment efficacy and correlate outcomes between studies. For the first time in the more than 40 years of radioembolization therapy, data is now reported from independently measured resin ^{90}Y microsphere concentration, size distribution, and density values. Of these reported properties, microsphere concentration displayed the greatest variability between vials, suggesting that, if specific knowledge of microsphere concentration is clinically important, it must be determined on a per-vial basis. Complete data of ^{90}Y microspheres used in radioembolization is critical for optimizing current use, and just as importantly, in developing new and more effective treatment strategies.

Chapter 4

Micro-Coulter counter: Design and demonstration

4.1 Introduction

The analysis of treatment vial concentration detailed in Chapter 3 revealed a high degree of variability between vials. Within the current treatment paradigm, there exists no method for real-time determination of delivered microsphere dose. Clearly, changes to the current treatment approach are warranted if better control of patient microsphere dose is desired.

This chapter proposes and demonstrates a novel micro-counter, which could be adapted for real-time microsphere counting as part of a RE procedure.

The Coulter counter is a well-established tool for counting and sizing particles in solution. It consists of two electrolyte-filled reservoirs, separated by a small channel, also known as the aperture. As a particle passes through the aperture, the electrical resistance of

this channel temporarily increases (see Figure 4.1). This resistance change can be detected using a sensing electrode pair, one contacting the upstream reservoir and one contacting the downstream reservoir. Changes in electric potential associated with a constant aperture current can be correlated to particle position and size, yielding information on the number and size distribution of particles in the solution.

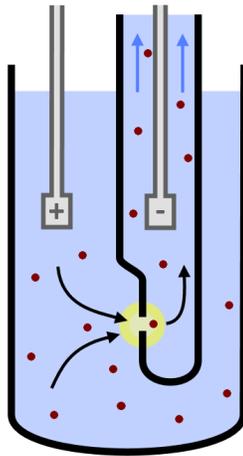


Figure 4.1 Illustration of the Coulter principle. Electrical resistance changes between two electrodes (in grey) are detected as particles (marked as red) pass through the sensing zone (marked in yellow).

Recent advancements in micro-fabrication techniques have improved the functionality and utility of micro-scale Coulter counters for detecting a wide array of micro-particles in compact devices [40], [41], [42], [43]. The majority of these designs rely on planar metal thin-film electrodes deposited onto a substrate, which is then bonded to a polydimethylsiloxane (PDMS) layer containing one or more micro-channels. Forming the electrodes requires sophisticated equipment and time-consuming fabrication processes for

metal vapor deposition (chemical or physical), and additional surface patterning steps to shape and etch the electrodes. Depending on the design, electrode substrates may require precise alignment with channel layers [44], [45]. Additionally, due to the coplanar nature of these electrodes, current density within the aperture is highly non-uniform and is concentrated along one side of the aperture channel [46]. Gawad et al. [47] and Sun et al. [48] investigated alternative electrode arrangements, and determined that devices with coplanar electrodes display decreased sensitivity in comparison to parallel cross-channel electrode arrangements, which feature parallel, but not coplanar, electrode faces, generally on either side of the primary channel (see Figure 4.2). Thick-film electrode fabrication techniques, which are based on screen printing practices and utilize a polymer-based conductive ink printed through a patterned sheet, can be used to construct full-height parallel electrodes, but these processes require multiple additional patterning and alignment steps for each device [49].

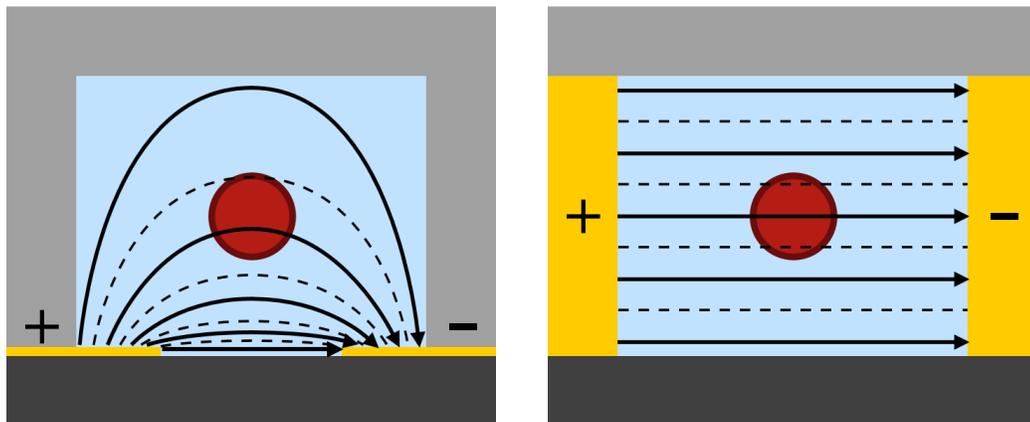


Figure 4.2 Representative illustration of electric field lines across a fluid filled channel in a device with coplanar (left) and parallel cross-channel (right) electrodes.

Micro-coulter counters can be classified according to their electrical excitation: AC or DC. While DC detection circuits are often easier to design and implement, performance is adversely affected by electrode polarization. As the device operates, ions in the electrolyte solution separate and a double layer resistance forms at the electrode-electrolyte interface which increases the total device resistance and reduces the ratio of aperture resistance to total device resistance (i.e. the aperture resistance ratio). Increasing the electrode-electrolyte contact surface area can reduce this double layer polarization effect, at the expense of device size and complexity. AC systems take advantage of the inverse proportionality of double layer impedance to excitation frequency to minimize its effects; however, stray capacitance effects are proportional to AC frequencies and can dominate the system's electrical response at high frequencies [50]. The AC frequency range which maximizes the aperture resistance ratio varies with device design, with reported values ranging from 10kHz to 10Mhz [43], [47], [51], [52], [53]. At high excitation frequencies, achieving sufficiently high sampling rates (well above the Nyquist limit) requires sophisticated measurement hardware. By increasing the electrode contact surface area, however, whether by using alternative electrode materials [such as with porous platinum black [52]], or by optimizing electrode design [43], improved counter performance can be achieved at more desirable, lower excitation frequencies (10 kHz and 30 kHz, respectively).

Recent investigations involving certain metal alloys, which are liquid at or slightly above room temperature and thus can be easily injected into pre-patterned channels, have found these materials to be suitable electrodes for micro-scale applications [54], [55], [56].

So and Dickey [57] investigated the use of eutectic gallium indium (EGaIn) as a micro-fluidic electrode in direct contact with NaCl solutions. EGaIn readily flows within the micro-channels once a critical injection pressure (dependent on channel dimensions) is exceeded. When the injection pressure is reduced below this threshold, an oxide skin (less than 1.0 nm thick) instantaneously forms, mechanically stabilizing the alloy for as long as the injection pressure remains below this threshold. The use of such electrodes offers a simple method for constructing stable devices while using low cost equipment and materials. Furthermore, fluid and electrodes channels can be fabricated simultaneously, producing inherently aligned features.

This chapter describes the design and demonstration of a novel, easily fabricated micro-Coulter counter using liquid metal electrodes. The design utilizes a full-channel parallel electrode arrangement and supports operation at low AC frequencies. Device fabrication requires only one photolithography step and uses standard soft-lithographic tools, eliminating the need for metal deposition and etching facilities. Particle detection is achieved using low-cost measurement equipment and straightforward, well-established post-processing techniques.

4.2 Methods and materials

4.2.1 Device design

As shown in Figure 4.3, the device consists of a primary fluid pathway (600 μm wide by 13.6 mm long) separated by a small aperture channel (80 μm wide by 300 μm long). Electrode

channels (700 μm wide), which are filled with a liquid metal alloy (EGaIn), flank either side of the aperture. A row of support posts (80 μm wide by 100 μm long, 250 μm spacing) separates each electrode channel from the central fluid channel (Figure 4.4). The spacing of these posts is optimized to prevent the liquid alloy from penetrating into the central primary fluid channel while maximizing electrode surface area. Furthermore, the “U”-shaped electrode channel provides a low pressure outlet and allows for manual injection of the EGaIn alloy without requiring precise control of injection volume or pressure. The spontaneous formation of a thin oxide skin on the metal stabilizes it mechanically within the channels. All features have height of 60 μm .

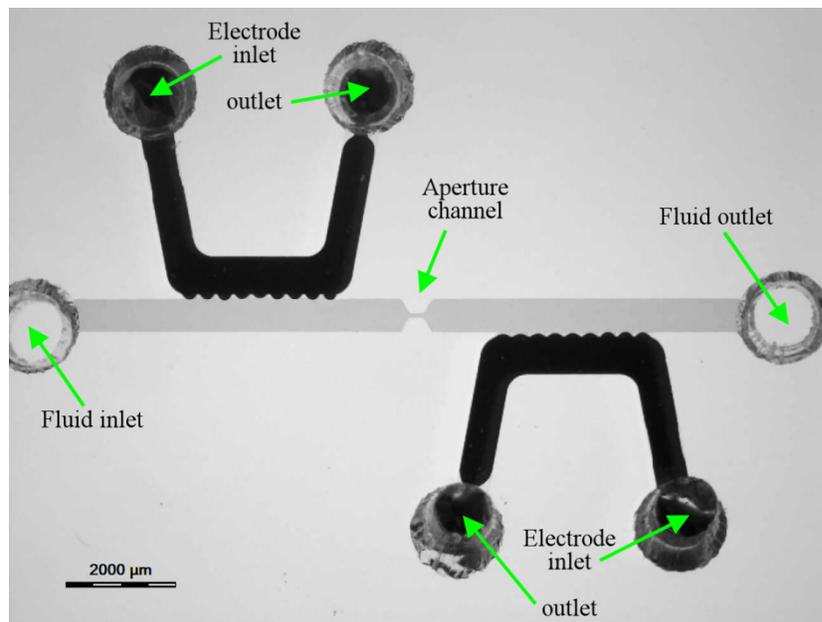


Figure 4.3 Back-lit optical micrograph of a micro-Coulter counter after EGaIn injection. The image has been contrast enhanced to highlight the fluid channel.

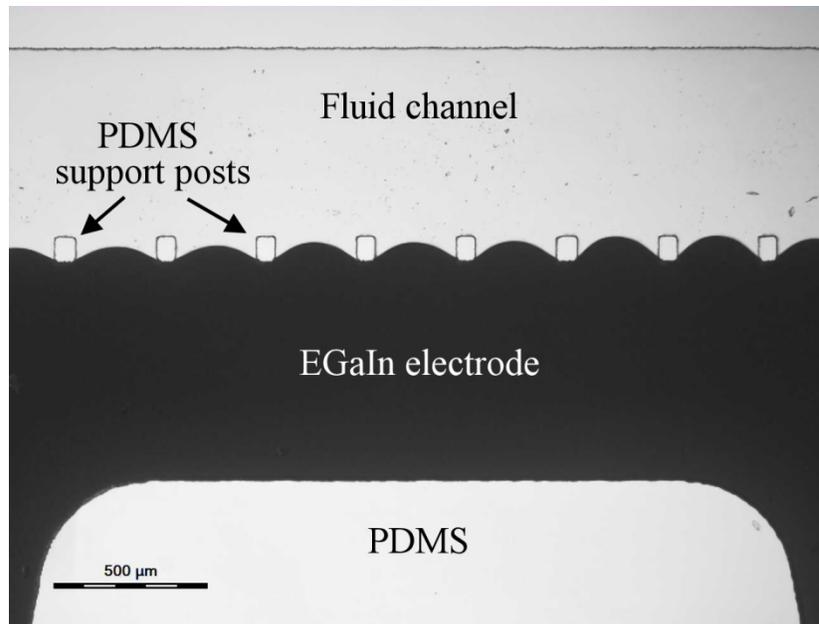


Figure 4.4 Back-lit detail image of PDMS support posts, which prevent EGaIn penetration into the central fluid channel.

4.2.2 Device fabrication

The device was fabricated from PDMS using established soft lithographic processes, wherein the channels were cast in PDMS from a negative master mold on a Si wafer, then bonded to a flat substrate, encapsulating the channels. Fluid and electrode channels were fabricated simultaneously in a single patterning step, eliminating the need for precise alignment or registration.

To fabricate the negative master mold, photoresist (SU-8 2050, Microchem Inc., Newton, MA, USA) was spin-coated on a silicon wafer at 2500 rpm for a target thickness of 60 μm. To allow irregularities in the photoresist surface to settle out, the coated wafer was

placed on a level surface for one hour before soft baking on a hotplate at 65 °C for 3 min and 95 °C for 9 min. A high-resolution printed photomask (Filmgraphics, Raleigh, NC, USA) was secured to the wafer before patterning the photoresist with ultraviolet light at target exposure energy of 200 mJ/cm² (Omniculture Series 1000, EXFO, Mississauga, ON, Canada). Following a post-exposure bake of 2 min at 65 °C and 7 min at 95 °C, the photoresist was agitated in SU-8 developer solution (Microchem Inc., Newton, MA, USA) for 10 minutes, then rinsed in isopropyl alcohol. Finally, to aid PDMS removal in later steps, the master mold was silanized using trichloro-(1H,1H,2H,2H-perfluorooctyl)-silane under vacuum for 6 hours. A picture of the completed master mold is shown in Figure 4.5.

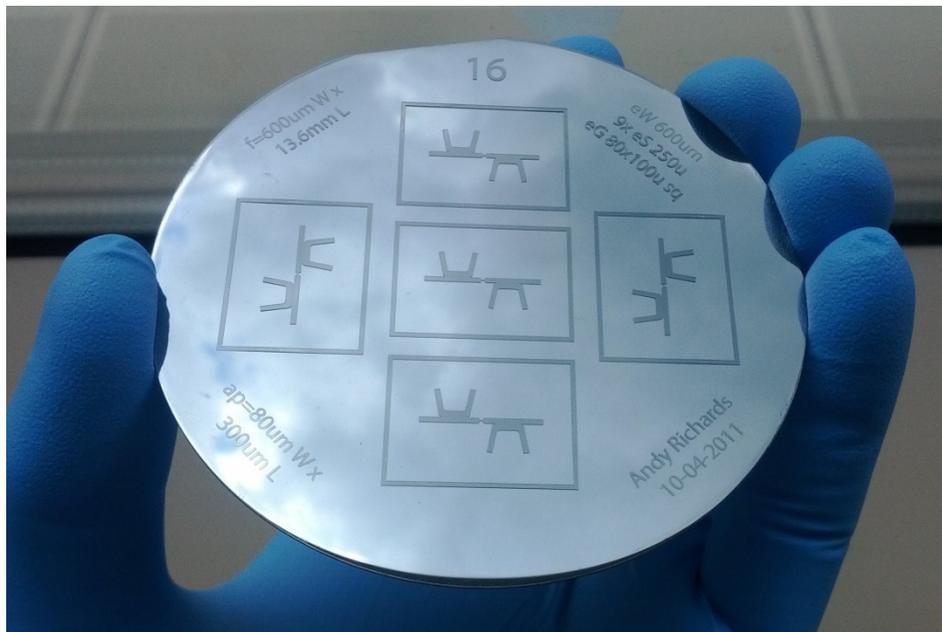


Figure 4.5 Completed photoresist-based master mold on Si wafer. Multiple devices can be fabricated simultaneously on one mold.

To create individual devices, 10:1 PDMS (Sylgard 184, Dow Corning, Midland, MI, USA) was poured over the master mold to a depth of approximately 3 mm, degassed in a vacuum chamber for 1 hour, and then cured at 60 °C for at least 2 hours. After removing the patterned PDMS from the Si wafer, a 1.5 mm hypodermic tube was used to create through-holes at the terminal ends of each inlet and outlet channel to enable fluid and electrode channel access. The PDMS layer and a glass slide were treated under oxygen plasma (20 s, 50% power, FEMTO, Diener Electronic, Ebhausen, Germany), then bonded together, thus encapsulating the channels. To form the electrodes, EGaIn ($\geq 99.99\%$, Sigma-Aldrich, St. Louis, MO, USA) was manually injected into each electrode inlet port using a 1 mL syringe connected via a short length of tubing (0.5 mm ID, 1.5 mm OD). Following EGaIn injection, 30 AWG wire was inserted into each EGaIn electrode and secured with a small amount of uncured PDMS. The completed device is shown in Figure 4.6. During experimental trials, tubing was inserted into the fluid inlet and outlet ports to facilitate particle introduction and removal.

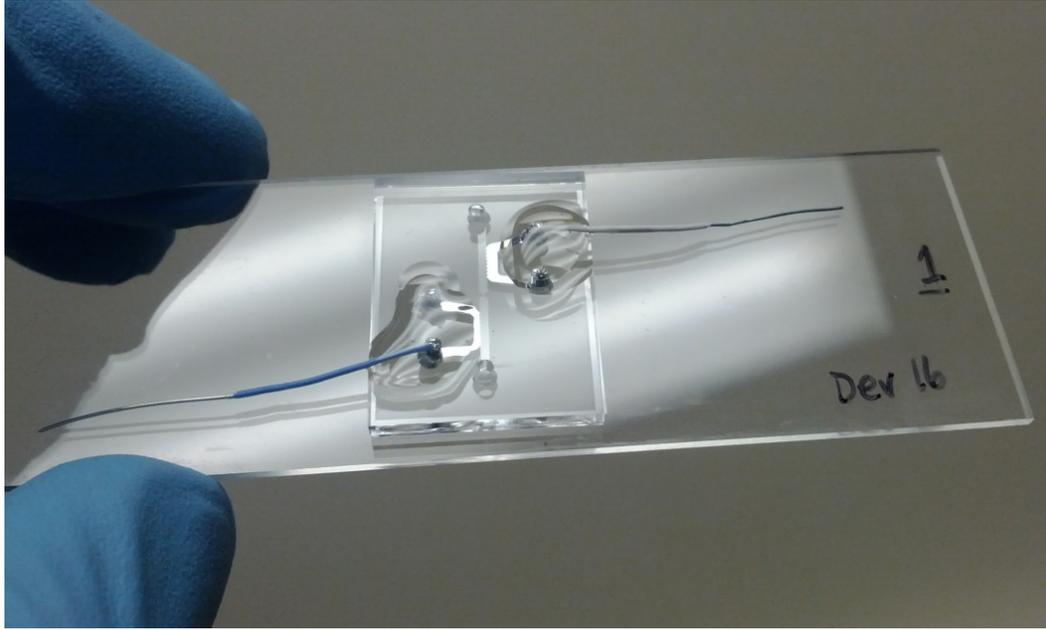


Figure 4.6 Image of completed device with secured electrode connections.

4.2.3 Measurement circuitry

To establish its functional characteristics, the device was filled with 0.9% saline (0.154 M NaCl) and a 1.0 V AC excitation potential was applied across its electrodes. The frequency was adjusted in discrete steps from 1 MHz to 10 Hz while monitoring the current through the device. This process identified 5 kHz to be the frequency which minimized the total device impedance (190 k Ω), while still remaining within the EGaIn electrode stability guidelines outlined by So and Dickey [57].

A polydisperse mixture of polymer resin microspheres, approximately 20-60 μm in diameter (SIR-Spheres Training and Demonstration Samples, Sirtex Medical Ltd., North

Sydney, Australia) was suspended in a 0.9% saline solution and introduced to the device via syringe pump at a rate of 10 $\mu\text{L}/\text{min}$.

To measure the resistance changes associated with particle transport through the aperture, a voltage divider circuit was fabricated using an added 150 k Ω series resistance. A signal generator (33220A, Agilent, Santa Clara, CA, USA) provided 1V AC excitation across the circuit. The device electrode potential was conditioned using a two stage active-filter and amplifier circuit, shown in Figure 4.7. Due to the relatively high resistance of the sensor, FET-input operational amplifiers were used to prevent any current draw from the amplifier circuit itself. After isolating the device from the remaining filter stage using a unity gain buffer, the signal was passed through a Sallen-Key 2nd order bandpass filter to eliminate environmental noise and interference.

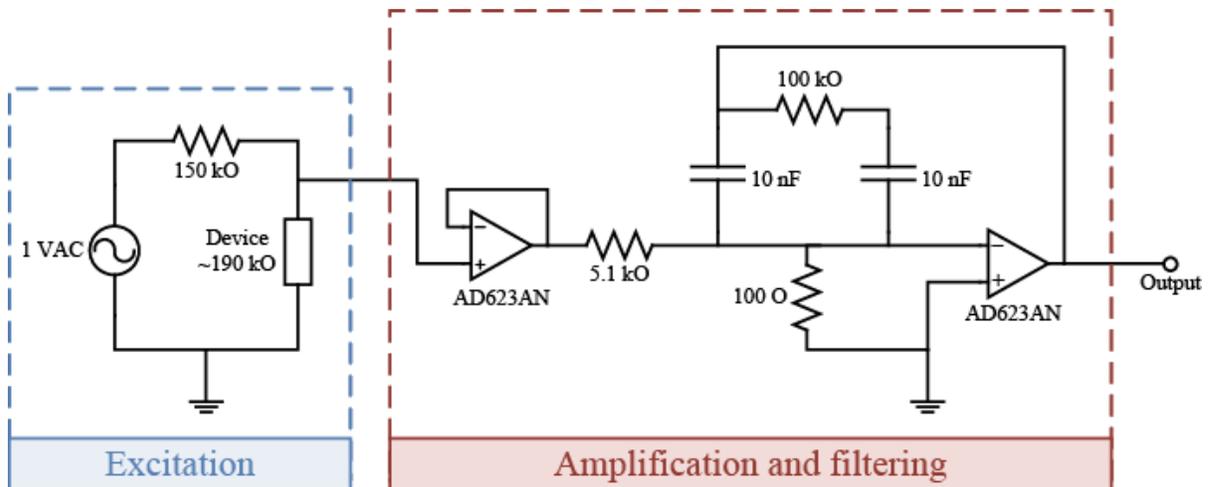


Figure 4.7 Active filter/amplifier circuit.

The bandpass filter was designed based on a 10 $\mu\text{L}/\text{min}$ fluid flow rate. Given the dimensions of the aperture, a particle transited the 300 μm aperture channel in no less than 8.6 ms, which corresponds to a maximum passage frequency of 115 Hz. The amplitude modulated device output, therefore, had a bandwidth of twice this frequency (i.e. 230 Hz) centered on the 5 kHz AC carrier frequency. The bandpass filter was constructed to preserve and amplify this signal, and exhibited an experimentally measured 4.97 kHz center frequency, 350 Hz 3dB bandwidth, and -8.8 V/V gain.

4.2.4 Signal processing and analysis

Voltage output from the filter stage was acquired at 200 kHz using a NI-6024E data acquisition card and a custom LabVIEW application (LabVIEW 2010, National Instruments, Austin, TX). By modeling the data as an amplitude modulated voltage signal, demodulation techniques could be used to separate the AC excitation wave from the aperture voltage signal associated with passing particles. Square-law demodulation, a well established demodulation technique [58], was implemented using custom MATLAB code (Mathworks, Natick, MA). To summarize, the output signal was squared to separate the signal into a high frequency component containing the AC carrier wave, and a low frequency component containing only the aperture voltage response. A 5th-order Butterworth low pass filter was used to isolate the aperture voltage signal. Finally, a custom bias-removal function was used to eliminate the DC offset, resulting in a signal representing the change in aperture voltage. The Square-law demodulation process can be represented mathematically as:

$$y(t) = (B + s(t)) * \cos(2\pi f_c t) \quad (4.1)$$

$$y(t)^2 = (B + s(t))^2 * \left(\frac{1}{2} + \frac{1}{2} \cos(4\pi f_c t) \right) \quad (4.2)$$

$$LPF[y(t)^2] \xrightarrow{yields} \frac{1}{2} (B + s(t))^2 \quad (4.3)$$

$$s(t) = \sqrt{2 * LPF[y(t)^2]} - B \quad (4.4)$$

where $y(t)$ is the amplitude modulated device output, $s(t)$ is the aperture voltage signal with DC offset B , and f_c is the AC excitation frequency. ‘ $LPF[]$ ’ represents the 5th-order low pass filter.

Finally, a peak-detection algorithm was used to identify local maxima within the reconstructed aperture signal, which correspond to temporary aperture obstructions. This filtering and analysis process is illustrated in Figure 4.8.

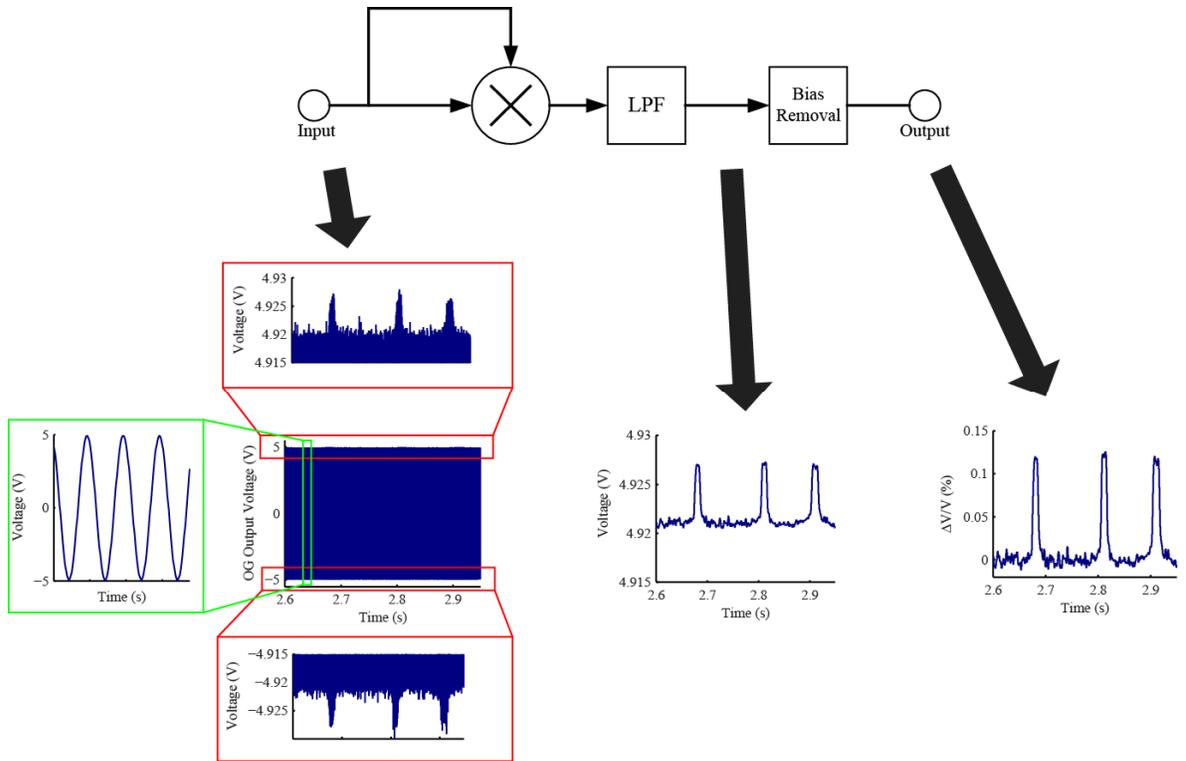


Figure 4.8 Illustration of Square law demodulation algorithm used to recover device response.

To evaluate the accuracy of this sensor, and to correlate particle sizes with output electrical responses, a high speed camera (1280x1024 resolution, HotShot Mega SC, NAC Image Technology, Simi Valley, CA) was connected to an inverted microscope (6.6X combined optical magnification, TC5400, Meiji Techno, Miyoshi, Saitama, Japan) to simultaneously record particles passing through the aperture channel at 500 fps. The inverted microscope was used to facilitate easy access to the device and prevent interference by the microscope objective.

The camera data was analyzed to determine the true number of “aperture obstructions” associated with particles passing through the device. A fundamental limitation

of all Coulter counters, referred to as “coincidence”, occurs when two or more particles exist in the aperture at the same time. An obstruction event was defined as one or more particles passing through the aperture in a single time period, thus coincidence occurrences were counted as a single obstruction.

Image processing was also performed on the captured camera data to determine the physical size of each particle for comparison with the device’s electrical response. For each identified microsphere, the measured diameter from three different camera frames was averaged to obtain the physical particle size. The theoretical relative aperture resistance change was calculated using the empirical formula presented in DeBlois et al. [59], which combines the upper and lower resistance change limits established by DeBlois and Bean [60], and Gregg and Steidley [61]. In order to calculate the relative change in voltage resulting from this resistance change, the baseline aperture resistance was estimated based on a saline conductivity of 14.4 mS/cm [62] and channel dimensions discussed previously.

4.3 Results

Figure 4.9 shows the simultaneous high speed camera images and sensor voltage responses (after filtering and peak detection) for a microsphere passing through the aperture. The sensor response associated with multiple microspheres is shown in Figure 4.10. The physical diameter measurement error, which was dependent on camera resolution and magnification, was determined to be $\pm 0.9 \mu\text{m}$.

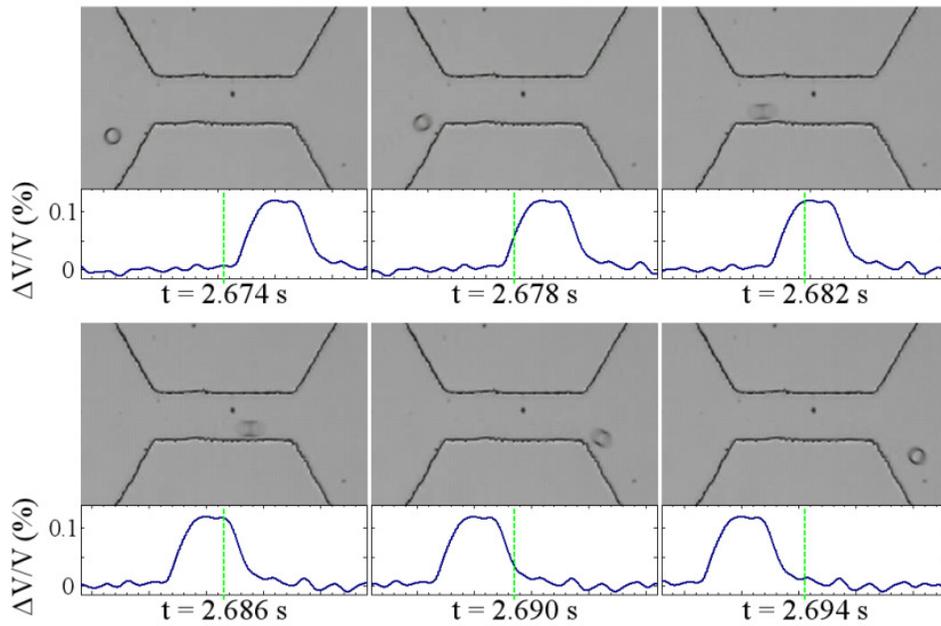


Figure 4.9 Six-frame depiction of a microsphere traversing the aperture channel. High speed camera images are shown above the corresponding normalized voltage change across the micro-counter, with the current value denoted by the vertical dashed line.

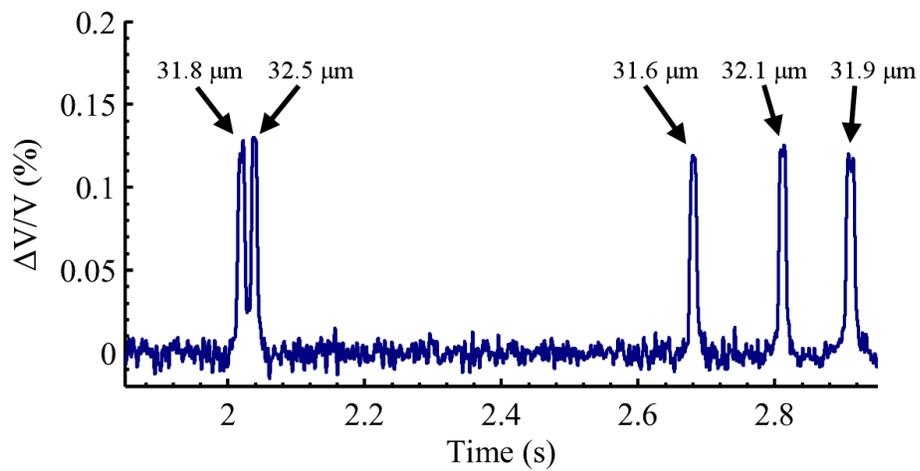


Figure 4.10 Filtered and post-processed device output displaying voltage response from five microspheres. Microsphere diameters obtained from high speed imagery are shown above each pulse and have an error of $\pm 0.9 \mu\text{m}$.

Table 4.1 shows a comparison between microsphere count determined by the high speed camera, and count derived from micro-counter pulses for six different samples. This comparison includes the total obstruction count, in which coincidence events are counted as a single obstruction, as well as the total microsphere count.

Table 4.1 Microsphere count comparison

Trial No.	Obstructions	Coincidence events (Spheres per event)	Actual spheres	Micro-counter pulses
1	15	1 (3)	17	15
2	28	1 (2)	29	28
3	29	5 (2,2,3,2,2)	35	29
4	32	7 (2,2,3,2,2,2,2)	40	32
5	32	5 (2,2,3,2,4)	39	32
6	11	2 (3,3)	15	11
Totals	147	21	175	147

The comparison between theoretical and actual device response, based on particle size, is shown in Figure 4.11 for a subset of experimental trials.

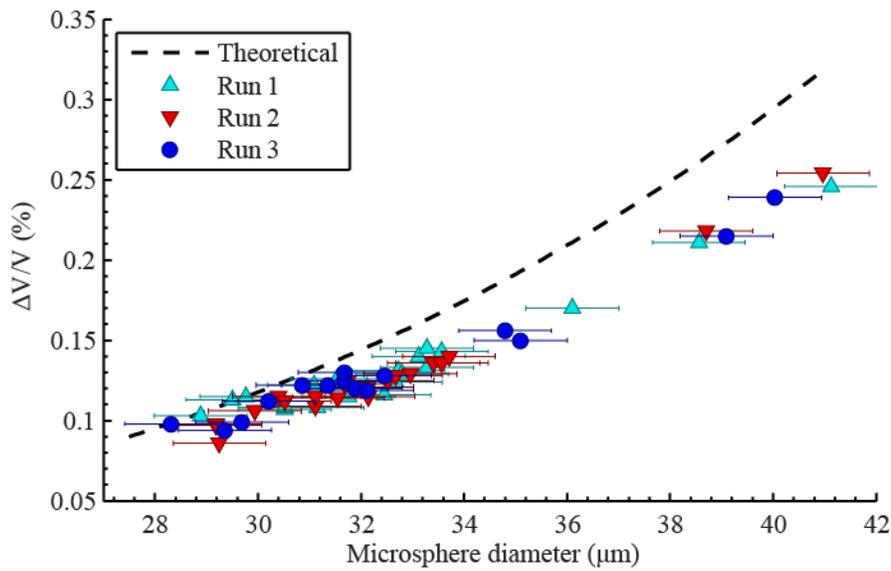


Figure 4.11 Relationship between particle diameter and device response. The dotted line shows the theoretical response based on the aperture resistance change equation formulated by DeBlois et al. [59]. Diameter measurements have an error of $\pm 0.9 \mu\text{m}$.

4.4 Discussion

The micro-counter successfully detected all aperture obstructions. As with all Coulter counters, coincidence events negatively affect the ability of the device to accurately count individual particles. With mono-disperse samples, coincidence can be compensated for based on the scaling of peak size with multiple particles; however, with polydisperse samples, Coulter counters lack the ability to distinguish between single large particles and multiple coincident smaller particles, as shown in Figure 4.12. While minimizing the aperture channel length would reduce coincidence occurrences, the high coincidence rate observed was predominantly due to insufficient particle mixing, as 95% of the coincidence events were the

result of aggregate particle groups. Adding a surfactant to the working fluid, along with sufficient agitation prior to injection, may minimize the coincidence rate for future studies.

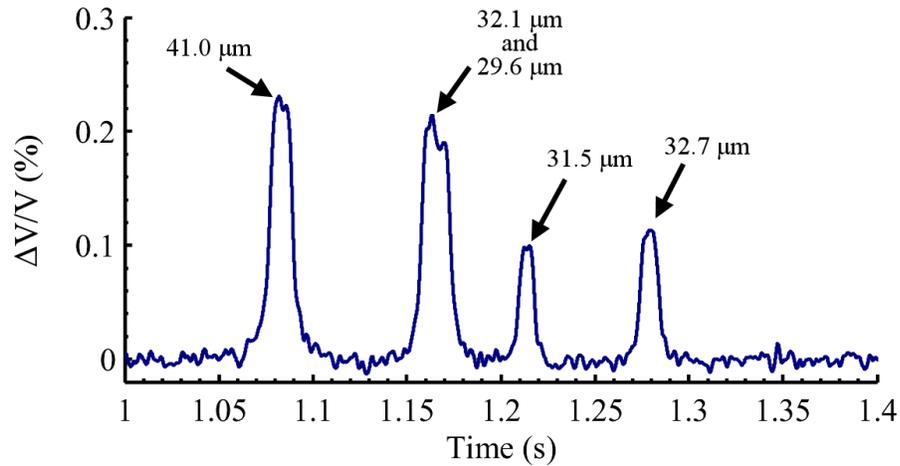


Figure 4.12 Filtered and post-processed device output. The voltage response due to a single large particle is indistinguishable from the response due to multiple coincident particles. Particle diameters obtained from high speed imagery are shown above each pulse and have an error of $\pm 0.9 \mu\text{m}$.

The device response was proportional to the size of the obstruction, as shown in Figure 4.11. As expected, the theoretical voltage response consistently overestimated the observed response, since the calculated resistance of the device was based solely on the electrical resistivity of the fluid (and neglected double layer resistance, wire lead resistance, stray capacitance effects, etc.). Without any further calibration, computed particle sizes based on the device output would underestimate the actual sizes. Improved particle sizing could be achieved through the use of microsphere size standards to create a calibration curve.

The device design allowed for AC operation at frequencies as low as 5 kHz. While robust particle detection was demonstrated, the presence of double layer effects indicates that optimal performance was not fully achieved. Future work will investigate increasing electrode contact surface area, or increasing the operating frequency to further reduce the double layer effects. Additionally, the performance of the sensor could be enhanced by integrating features from other published micro-counter designs, such as the use of hydrodynamic focusing to align the particles within the channel [51], and utilizing a multi-channel arrangement to increase throughput [53].

A limitation of all Coulter counter designs is the interaction of the electrodes with the fluid itself. If voltage amplitudes across the electrodes exceed a critical level, electrode degradation can occur as the electrolytic working fluid electrochemically reacts with the metal electrode. With EGaIn-based electrodes, the degradation of the stabilizing oxide skin of the metal can cause the electrode to retract into the electrode channel. No damage of the EGaIn electrodes was observed at AC voltage amplitudes less than 1 V.

Maximum device throughput is limited by the fluid pressure, due to the instability of the liquid EGaIn electrode at high pressures. Though the oxide skin stabilizes the electrode post-injection, it is still possible to rupture the skin and alter the electrode shape. The maximum flow rate can be approximated using the Darcy-Weisbach method for determining pressure loss within a channel. Based on the central fluid channel dimensions (and assuming no pressure loss beyond the outlet of the device), the fluid pressure exceeds the critical pressure required to disrupt the EGaIn electrode (12.9 kPa) at an approximate 540 $\mu\text{L}/\text{min}$

flow rate. At the documented flow rate of 10 $\mu\text{L}/\text{min}$, the theoretical pressure is only 0.07 kPa. In principle, the stability of the electrodes could be improved further by using alloys with melting points above room temperature such that the resulting electrodes are solid during operation.

While gallium compounds are routinely used in medical applications [63], the biocompatibility and toxicity of EGaIn has not been investigated, thus the effect on biological cells from using EGaIn-based electrodes in cytometric applications is unknown.

The use of liquid metal electrodes offers numerous fabrication and functionality benefits for micro-Coulter counters. The device can be fabricated using a single lithography step outside of a clean room and without the use of vapor deposition to define the electrodes. After a master mold has been fabricated, devices can be constructed without any additional patterning steps. Once injected, the electrode material fills a channel with the same height as the primary fluid pathway. By placing electrodes on opposite sides of the fluid aperture, a cross-channel parallel configuration is achieved, which has been shown to increase sensitivity compared to coplanar electrode configurations [47], [48]. This cross-channel electrode configuration is difficult to accomplish with traditional metal thin- and thick-film fabrication processes. By incorporating support posts, the effective EGaIn electrode surface area can be increased to reduce double layer effects at the electrode-electrolyte interface, and allow for AC operation at low frequencies.

The device utilized a flexible PDMS channel layer bonded to a rigid glass substrate. The use of glass was not essential for the device operation, but was chosen for ease of

handling during the course of the experiments. The use of an injectable liquid metal alloy such as EGaIn removes any planar constraint for the electrodes, and can enable non-planar or flexible device designs.

4.5 Conclusion

A micro-Coulter counter utilizing EGaIn electrodes was developed and demonstrated for the detection of particles in solution. The novel use of an injectable liquid metal alloy allowed for the simple fabrication of fluid and electrode channels patterned in a single step. The functionality of the completed device was tested using a polydisperse mixture of polymer-based microspheres and validated using high speed video. Operation was demonstrated at AC frequencies as low as 5 kHz using low-cost measurement equipment and simple post processing techniques. The device demonstrates a novel approach for the fabrication of functional micro-fluidic devices utilizing liquid metal electrodes.

Chapter 5

Conclusions

In Chapter 2, a scaled model of a generalized hepatic system was fabricated to replicate the fluid dynamics in diseased liver hepatic arteries. Neutrally buoyant microspheres were released from controlled locations within the inlet of the model and the resulting output distributions were recorded. Fluid and particle transport simulations were conducted with identical parameters. The resulting experimental- and simulation-derived microsphere distributions were compared.

The experimental microsphere distribution exhibited a clear dependence on injection location that correlated very strongly with the computationally predicted results. Individual branch targeting was possible for each of the five outputs. The experimental results validate the simulation methodology for achieving targeted microsphere distributions in a known geometry with constant flow conditions.

In Chapter 3, multiple particles analysis methods were used to determine the concentration, size distribution, and density of 26 resin ^{90}Y radioembolization microsphere delivery vials. Coulter counter analysis indicated a mean vial concentration of 12.9 ± 0.4 million microspheres per mL (Msph/mL), with a standard deviation of 2.3 Msph/mL. Optical

microsphere diameter analysis indicated a non-standardized t-distribution with mean microsphere diameter of $29.1 \pm 0.8 \mu\text{m}$ (standard deviation, $1.2 \mu\text{m}$). Density measurements indicated a mean microsphere density of $1.57 \pm 0.01 \text{ g/cm}^3$ (standard deviation, 0.06 g/cm^3). Based on statistical models fit to each dataset, microsphere concentration displayed the greatest variability between vials, suggesting that, if specific knowledge of concentration is clinically important, it must be determined on a per-vial basis. Complete microsphere data is critical for developing effective targeting strategies, and further enables better-informed clinical decisions and analyses.

Chapter 4 described the design and demonstration of a novel, easily fabricated micro-Coulter counter utilizing liquid metal electrodes. Fluid and electrode channels were fabricated simultaneously in a single lithographic patterning step. Eutectic Gallium-Indium (EGaIn) was injected into the device to form functional electrodes in a cross-channel parallel configuration. Functionality of the device was demonstrated at an AC excitation frequency of 5 kHz using a polymer microsphere suspension, and simple post processing techniques. The device successfully detected particles, exhibiting an output response proportional to particle size. EGaIn was demonstrated to be an effective micro-fluidic electrode material and provided a novel approach for the fabrication of a functional micro-Coulter counter.

5.1 Future work

The experimental validation of the CFD-based targeting strategy opens the door for continued research in the area of simulation-guided hepatic microsphere injections. By demonstrating that controlled targeting is feasible, the technique can be expanded to

encompass more parameters and geometries. As the tests described here only covered the case of steady flow, further tests are required to investigate transient/pulsatile flow scenarios. Future work will naturally address improving simulation and experimental parameters to better reflect the experimental environment and account for differences, enabling investigation of model variables such as microsphere density, gravity direction, and hepatic geometry.

Deviations from simulation ideals will always be present. Rather than relying completely on simulation results, an alternative approach could be utilized to achieve targeted microsphere delivery. Test injections, guided by prior knowledge of the approximate size and location of the branch zones from simplified CFD simulations, could be performed to define key areas of the experimental release map. Based on the distribution of these preliminary injections, the final location could be adjusted to increase the accuracy of the primary particle injection. In the experimental scenario, these test injections could consist of small concentrations of the same spheres used in the primary injection. A clinical correlation could be the use of temporary particles, such as ^{99m}Tc -labelled macroaggregated albumin (MAA), for a similar effect. This hybrid approach couples ideal simulation predictions with natural experimental instability, and may be a better fit for the clinical environment.

Future work will investigate pulsatile flow, intermittent injection profiles, flexible vessels, etc. as well as the development of an actuated catheter that can be used to regulate the release point in vivo. Additionally, this work will seek to better understand and minimize

discrepancies between simulated and experimental results. Alternative approaches will be investigated to enhance the efficacy of targeted microsphere delivery.

Though two brands of ^{90}Y microspheres are commonly used in RE treatments, only one brand's physical properties were examined in Chapter 3. While this data is valuable to physicians utilizing SIR-Spheres brand resin microspheres, analysis of all commercially available RE microspheres (e.g. TheraSpheres brand glass microspheres) would be prudent, as the targeting strategy is not dependant on one brand. Additional, uninvestigated properties exist that may prove useful in the future, such as fluorescence, magnetic permeability, sphericity, smoothness, etc.

For the micro-device developed in Chapter 4, only the basic functionality of the device was demonstrated. Further tests are required in varying flow scenarios. The tested flow rate through the device is too low to be connected directly in-line with current RE delivery systems, but it may be possible to divert partial flow through the micro-counter, and correlate the reading with the true value (based on calibration). Additionally, work is required to convert the basic proof-of-concept prototype into a form factor appropriate for a clinical setting.

REFERENCES

- [1] A. Kennedy, D. Coldwell, B. Sangro, H. Wasan, and R. Salem, "Integrating radioembolization ((90)Y microspheres) into current treatment options for liver tumors: introduction to the international working group report.," *Am J Clin Oncol*, vol. 35, no. 1, pp. 81–90, Feb. 2012.
- [2] R. Salem and K. G. Thurston, "Radioembolization with 90Yttrium microspheres: a state-of-the-art brachytherapy treatment for primary and secondary liver malignancies. Part 1: Technical and methodologic considerations.," *J Vasc Interv Radiol*, vol. 17, no. 8, pp. 1251–78, Aug. 2006.
- [3] D. M. Parkin, F. Bray, J. Ferlay, and P. Pisani, "Global Cancer Statistics, 2002," *CA: A Cancer Journal for Clinicians*, vol. 55, no. 2, pp. 74–108, Mar. 2005.
- [4] K. R. Hess, G. R. Varadhachary, S. H. Taylor, W. Wei, M. N. Raber, R. Lenzi, and J. L. Abbruzzese, "Metastatic patterns in adenocarcinoma.," *Cancer*, vol. 106, no. 7, pp. 1624–33, Apr. 2006.
- [5] J. S. Welsh, A. S. Kennedy, and B. Thomadsen, "Selective Internal Radiation Therapy (SIRT) for liver metastases secondary to colorectal adenocarcinoma.," *Int J Radiat Oncol Biol Phys*, vol. 66, no. 2 Suppl, pp. S62–73, Jan. 2006.
- [6] S. M. Ibrahim, P. Nikolaidis, F. H. Miller, R. J. Lewandowski, R. K. Ryu, K. T. Sato, S. Senthilnathan, A. Riaz, L. Kulik, M. F. Mulcahy, R. A. Omary, and R. Salem, "Radiologic findings following Y90 radioembolization for primary liver malignancies.," *Abdom Imaging*, vol. 34, no. 5, pp. 566–81, 2009.
- [7] H. Bismuth and P. E. Majno, "Hepatobiliary surgery," *J Hepatol*, vol. 32, no. null, pp. 208–224, Jan. 2000.
- [8] A. S. Kennedy, C. Nutting, D. Coldwell, J. Gaiser, and C. Drachenberg, "Pathologic response and microdosimetry of (90)Y microspheres in man: review of four explanted whole livers.," *Int J Radiat Oncol Biol Phys*, vol. 60, no. 5, pp. 1552–63, Dec. 2004.
- [9] K. K. Herfarth, J. Debus, F. Lohr, M. L. Bahner, B. Rhein, P. Fritz, A. Höss, W. Schlegel, and M. F. Wannemacher, "Stereotactic single-dose radiation therapy of liver tumors: results of a phase I/II trial.," *J Clin Oncol*, vol. 19, no. 1, pp. 164–70, Jan. 2001.

- [10] A. Kennedy, S. Nag, R. Salem, R. Murthy, A. J. McEwan, C. Nutting, A. Benson, J. Espot, J. I. Bilbao, R. A. Sharma, J. P. Thomas, and D. Coldwell, "Recommendations for radioembolization of hepatic malignancies using yttrium-90 microsphere brachytherapy: a consensus panel report from the radioembolization brachytherapy oncology consortium.," *Int J Radiat Oncol Biol Phys*, vol. 68, no. 1, pp. 13–23, May 2007.
- [11] A. M. Campbell, I. H. Bailey, and M. A. Burton, "Tumour dosimetry in human liver following hepatic yttrium-90 microsphere therapy.," *Phys Med Biol*, vol. 46, no. 2, pp. 487–98, Feb. 2001.
- [12] V. M. Meade, M. A. Burton, B. N. Gray, and G. W. Self, "Distribution of different sized microspheres in experimental hepatic tumours," *Eur J Cancer Clin Oncol*, vol. 23, no. 1, pp. 37–41, Jan. 1987.
- [13] R. Jirtle, K. H. Clifton, and J. H. G. Rankin, "Measurement of Mammary Tumor Blood Flow in Unanesthetized Rats," *J Natl Cancer Inst*, vol. 60, no. 4, pp. 881–886, Apr. 1978.
- [14] D. Coldwell, B. Sangro, H. Wasan, R. Salem, and A. Kennedy, "General selection criteria of patients for radioembolization of liver tumors: an international working group report.," *Am J Clin Oncol*, vol. 34, no. 3, pp. 337–41, Jun. 2011.
- [15] A. Rammohan, J. Sathyanesan, S. Ramaswami, A. Lakshmanan, P. Senthil-Kumar, U. P. Srinivasan, R. Ramasamy, and P. Ravichandran, "Embolization of liver tumors: Past, present and future.," *World J Radiol*, vol. 4, no. 9, pp. 405–12, Sep. 2012.
- [16] R. Salem and K. G. Thurston, "Radioembolization with yttrium-90 microspheres: a state-of-the-art brachytherapy treatment for primary and secondary liver malignancies: part 3: comprehensive literature review and future direction.," *J Vasc Interv Radiol*, vol. 17, no. 10, pp. 1571–93, Oct. 2006.
- [17] R. Salem and K. G. Thurston, "Radioembolization with 90yttrium microspheres: a state-of-the-art brachytherapy treatment for primary and secondary liver malignancies. Part 2: special topics.," *J Vasc Interv Radiol*, vol. 17, no. 9, pp. 1425–39, Sep. 2006.
- [18] M. W. Barentsz, M. A. D. Vente, M. G. E. H. Lam, M. L. J. Smits, J. F. W. Nijsen, B. a Seinstraa, C. E. N. M. Rosenbaum, H. M. Verkooijen, B. A. Zonnenberg, and M. A. A. J. Van den Bosch, "Technical Solutions to Ensure Safe Yttrium-90 Radioembolization in Patients With Initial Extrahepatic Deposition of (99m)Technetium-Albumin Macroaggregates.," *Cardiovasc Intervent Radiol*, vol. 34, no. 5, pp. 1074–1079, Dec. 2010.

- [19] A. S. Kennedy, C. Kleinstreuer, C. A. Basciano, and W. A. Dezarn, "Computer modeling of yttrium-90-microsphere transport in the hepatic arterial tree to improve clinical outcomes.," *Int J Radiat Oncol Biol Phys*, vol. 76, no. 2, pp. 631–7, Feb. 2010.
- [20] C. A. Basciano, C. Kleinstreuer, A. S. Kennedy, W. A. Dezarn, and E. Childress, "Computer modeling of controlled microsphere release and targeting in a representative hepatic artery system.," *Ann Biomed Eng*, vol. 38, no. 5, pp. 1862–79, May 2010.
- [21] K. M. Carlisle, M. Halliwell, A. E. Read, and P. N. Wells, "Estimation of total hepatic blood flow by duplex ultrasound.," *Gut*, vol. 33, no. 1, pp. 92–7, Jan. 1992.
- [22] S.-H. B. Han, S. Rice, S. M. Cohen, T. B. Reynolds, and T.-L. Fong, "Duplex Doppler ultrasound of the hepatic artery in patients with acute alcoholic hepatitis.," *J Clin Gastroenterol*, vol. 34, no. 5, pp. 573–7, 2002.
- [23] E. Leen, J. A. Goldberg, J. Robertson, G. R. Sutherland, D. M. Hemingway, T. G. Cooke, and C. S. McArdle, "Detection of hepatic metastases using duplex/color Doppler sonography.," *Ann Surg*, vol. 214, no. 5, pp. 599–604, Nov. 1991.
- [24] P. L. Roberson, R. K. Ten Haken, D. L. McShan, P. E. McKeever, and W. D. Ensminger, "Three-Dimensional Tumor Dosimetry for Hepatic Yttrium-90-Microsphere Therapy," *J. Nucl. Med.*, vol. 33, no. 5, pp. 735–738, 1992.
- [25] S. Guadagni, A. Pizzutilli, E. Mancini, A. Varrone, G. Palumbo, G. Amicucci, S. Perri, M. Deraco, and G. Fiorentini, "Significance of duplex/colour Doppler sonography in hepatic arterial chemotherapy for patients with liver metastases from colorectal carcinoma.," *Eur J Surg Oncol*, vol. 26, no. 4, pp. 381–6, Jun. 2000.
- [26] S. O. Oktar, C. Yucel, T. Demirogullari, A. Uner, M. Benekli, G. Erbas, and H. Ozdemir, "Doppler Sonographic Evaluation of Hemodynamic Changes in Colorectal Liver Metastases Relative to Liver Size," *J. Ultrasound Med.*, vol. 25, no. 5, pp. 575–582, 2006.
- [27] R. L. Mott, *Applied Fluid Mechanics*, Fifth. Upper Saddle River, New Jersey: Prentice Hall, 2000, pp. 224–233.
- [28] C. Kleinstreuer, *Two-phase flow: theory and applications*. Taylor & Francis, 2003, p. 454.
- [29] C. Kleinstreuer, *Biofluid dynamics: principles and selected applications*. CRC/Taylor & Francis, 2006, p. 492.

- [30] F. P. Beer, E. R. Johnston, and J. T. DeWolf, *Mechanics of Materials*, 3rd ed. New York: McGraw-Hill, 2002, pp. 530–606.
- [31] D. F. Capes, K. R. Dunster, V. B. Sunderland, D. McMillan, P. B. Colditz, and C. McDonald, “Fluctuations in syringe-pump infusions: association with blood pressure variations in infants.,” *Am J Health Syst Pharm*, vol. 52, no. 15, pp. 1646–53, Aug. 1995.
- [32] K. R. Dunster and P. B. Colditz, “Flow continuity of infusion systems at low flow rates.,” *Anaesth Intensive Care*, vol. 23, no. 5, pp. 605–9, Oct. 1995.
- [33] A. Kennedy, D. Coldwell, B. Sangro, H. Wasan, and R. Salem, “Radioembolization for the Treatment of Liver Tumors,” *Am J Clin Oncol*, vol. 35, no. 1, pp. 91–99, Feb. 2012.
- [34] C.-Y. O. Wong, M. Savin, K. M. Sherpa, F. Qing, J. Campbell, V. L. Gates, R. J. Lewandowski, V. Cheng, J. Thie, D. Fink-Bennett, C. Nagle, and R. Salem, “Regional yttrium-90 microsphere treatment of surgically unresectable and chemotherapy-refractory metastatic liver carcinoma.,” *Cancer Biother Radiopharm*, vol. 21, no. 4, pp. 305–13, Aug. 2006.
- [35] R. S. Stubbs, R. J. Cannan, and A. W. Mitchell, “Selective internal radiation therapy (SIRT) with 90Yttrium microspheres for extensive colorectal liver metastases.,” *Hepatogastroenterology*, vol. 48, no. 38, pp. 333–7, 2001.
- [36] SIR-Spheres package insert. Sirtex Medical Ltd., North Sydney, Australia, Feb 2010.
- [37] ASTM Standard D854, 2010, “Standard Test Methods for Specific Gravity of Soil Solids by Water Pycnometer”, ASTM International, West Conshohocken, PA, 2010, DOI: 10.1520/D0854-10, www.astm.org.
- [38] S. Shaphiro and M. Wilk, “An analysis of variance test for normality,” *Biometrika*, vol. 52, no. 3, pp. 591–611, 1965.
- [39] A. S. Kennedy, A. L. Richards, P. McNeille, A. Weiss, and W. A. Dezarn, “Electron Microscopy Analyses of 90Y Microspheres,” *Translational Cancer Research*. [in submission].
- [40] A. Carbonaro and L. L. Sohn, “A resistive-pulse sensor chip for multianalyte immunoassays.,” *Lab Chip*, vol. 5, no. 10, pp. 1155–60, Oct. 2005.

- [41] M. Korampally, J. D. Benson, Y. Wu, J. K. Critser, and M. Almasri, "MEMS based Coulter counter for cell sizing," in *SPIE 6886, Microfluidics, BioMEMS, and Medical Microsystems VI*, 2008, vol. 6886, p. 68860A.
- [42] J. Zhe, A. Jagtiani, P. Dutta, J. Hu, and J. Carletta, "A micromachined high throughput Coulter counter for bioparticle detection and counting," *Journal of Micromechanics and Microengineering*, vol. 17, no. 2, pp. 304–313, Feb. 2007.
- [43] A. V Jagtiani, J. Carletta, and J. Zhe, "An impedimetric approach for accurate particle sizing using a microfluidic Coulter counter," *Journal of Micromechanics and Microengineering*, vol. 21, no. 4, p. 045036, Apr. 2011.
- [44] A. L. McPherson and G. M. Walker, "A microfluidic passive pumping Coulter counter," *Microfluidics and Nanofluidics*, vol. 9, no. 4–5, pp. 897–904, Apr. 2010.
- [45] X. Wu, Y. Kang, Y.-N. Wang, D. Xu, D. Li, and D. Li, "Microfluidic differential resistive pulse sensors.," *Electrophoresis*, vol. 29, no. 13, pp. 2754–9, Jul. 2008.
- [46] L. I. Berge, T. Jossang, and J. Feder, "Off-axis response for particles passing through long apertures in Coulter-type counters," *Measurement Science and Technology*, vol. 1, no. 6, pp. 471–474, Jun. 1990.
- [47] S. Gawad, L. Schild, and P. H. Renaud, "Micromachined impedance spectroscopy flow cytometer for cell analysis and particle sizing.," *Lab Chip*, vol. 1, no. 1, pp. 76–82, Sep. 2001.
- [48] T. Sun, N. Green, S. Gawad, and H. Morgan, "Analytical electric field and sensitivity analysis for two microfluidic impedance cytometer designs," *IET Nanobiotechnology*, pp. 69–79, 2007.
- [49] X. Niu, M. Zhang, S. Peng, W. Wen, and P. Sheng, "Real-time detection, control, and sorting of microfluidic droplets.," *Biomicrofluidics*, vol. 1, no. 4, p. 44101, Jan. 2007.
- [50] S. Gawad, K. Cheung, U. Seger, A. Bertsch, and P. Renaud, "Dielectric spectroscopy in a micromachined flow cytometer: theoretical and practical considerations.," *Lab Chip*, vol. 4, no. 3, pp. 241–51, Jun. 2004.
- [51] R. Rodriguez-Trujillo, C. A. Mills, J. Samitier, and G. Gomila, "Low cost micro-Coulter counter with hydrodynamic focusing," *Microfluidics and Nanofluidics*, vol. 3, no. 2, pp. 171–176, Sep. 2006.

- [52] S. Zheng, M. Liu, and Y.-C. Tai, "Micro coulter counters with platinum black electroplated electrodes for human blood cell sensing.," *Biomed Microdevices*, vol. 10, no. 2, pp. 221–31, Apr. 2008.
- [53] A. V Jagtiani, J. Carletta, and J. Zhe, "A microfluidic multichannel resistive pulse sensor using frequency division multiplexing for high throughput counting of micro particles," *Journal of Micromechanics and Microengineering*, vol. 21, no. 6, p. 065004, Jun. 2011.
- [54] R. C. Chiechi, E. A. Weiss, M. D. Dickey, and G. M. Whitesides, "Eutectic gallium-indium (EgIn): a moldable liquid metal for electrical characterization of self-assembled monolayers.," *Angew Chem Int Ed Engl*, vol. 47, no. 1, pp. 142–4, Jan. 2008.
- [55] M. D. Dickey, R. C. Chiechi, R. J. Larsen, E. A. Weiss, D. A. Weitz, and G. M. Whitesides, "Eutectic Gallium-Indium (EgIn): A Liquid Metal Alloy for the Formation of Stable Structures in Microchannels at Room Temperature," *Advanced Functional Materials*, vol. 18, no. 7, pp. 1097–1104, Apr. 2008.
- [56] J.-H. So, J. Thelen, A. Qusba, G. J. Hayes, G. Lazzi, and M. D. Dickey, "Reversibly Deformable and Mechanically Tunable Fluidic Antennas," *Advanced Functional Materials*, vol. 19, no. 22, pp. 3632–3637, Nov. 2009.
- [57] J.-H. So and M. D. Dickey, "Inherently aligned microfluidic electrodes composed of liquid metal.," *Lab Chip*, vol. 11, no. 5, pp. 905–11, Mar. 2011.
- [58] Y. Shmaliy, *Signals and Communication Technology: Continuous-Time Signals*. Dordrecht: Springer, 2006, p. 155.
- [59] R. W. DeBlois, C. P. Bean, and R. K. . Wesley, "Electrokinetic measurements with submicron particles and pores by the resistive pulse technique," *Journal of Colloid and Interface Science*, vol. 61, no. 2, pp. 323–335, Sep. 1977.
- [60] R. W. DeBlois and C. P. Bean, "Counting and Sizing of Submicron Particles by the Resistive Pulse Technique," *Review of Scientific Instruments*, vol. 41, no. 7, p. 909, 1970.
- [61] E. C. Gregg and K. D. Steidley, "Electrical counting and sizing of mammalian cells in suspension.," *Biophys J*, vol. 5, no. 4, pp. 393–405, Jul. 1965.
- [62] R. C. Weast, *CRC Handbook of Chemistry and Physics*. Cleveland: CRC Press, 1974, p. D224.

- [63] C. R. Chitambar, "Medical applications and toxicities of gallium compounds.," *Int J Environ Res Public Health*, vol. 7, no. 5, pp. 2337–61, May 2010.

APPENDICES

Appendix A

Injection tube location

A.1 Introduction

For the design of the microsphere injection system presented in Chapter 2, three support wires, connected to the injection tube, were placed at 120 degree intervals around the inlet pipe, at two locations along the pipe as specified in Figure A.1. By adjusting the support wire lengths, the injection tube position within the inlet pipe could be controlled.

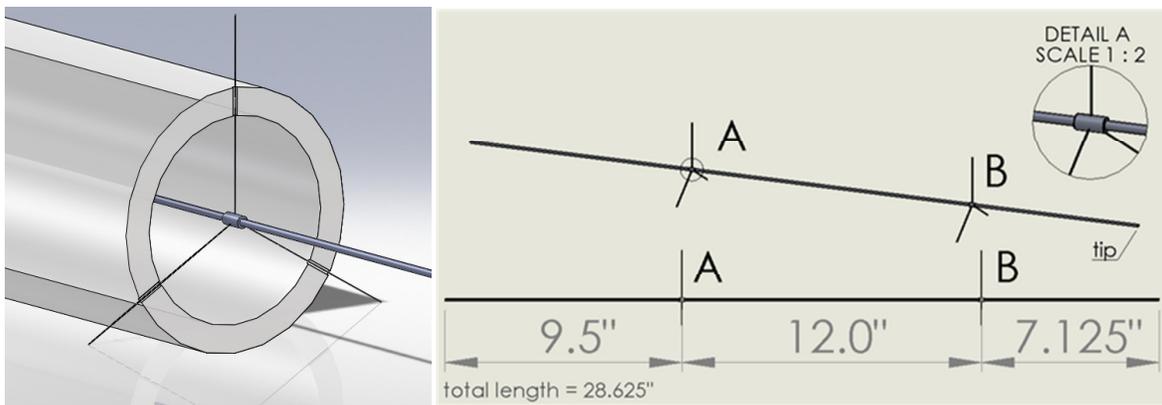


Figure A.1 Injection tube position within the inlet pipe (left), and dimensions (right).

3D-printed positioning mounts, based on a screw-driven mechanism, allowed for precise adjustment of the wire lengths, shown in Figure A.2.

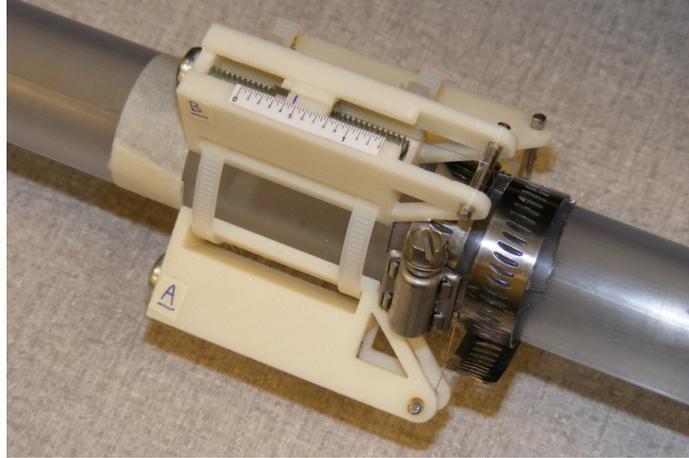


Figure A.2 Positioning mount with screw-driven adjustment mechanism.

A.2 Determining required wire length for desired position

It is assumed that the injection tube should always be colinear with the inlet pipe, thus the wire lengths need only be determined at one positioning mount and then duplicated at the second. Appropriate wire lengths could be determined using basic geometric principles.

Consider an inlet tube with radius r of the inlet tube, and a desired injection tube position $W = (x, y)$ from the center of the tube. As shown in Figure A.3, if W is represented as a vector h , from the center of the tube, the length of positioning wires (represented as vectors i, j, k) required to support the injection tube may be determined geometrically.

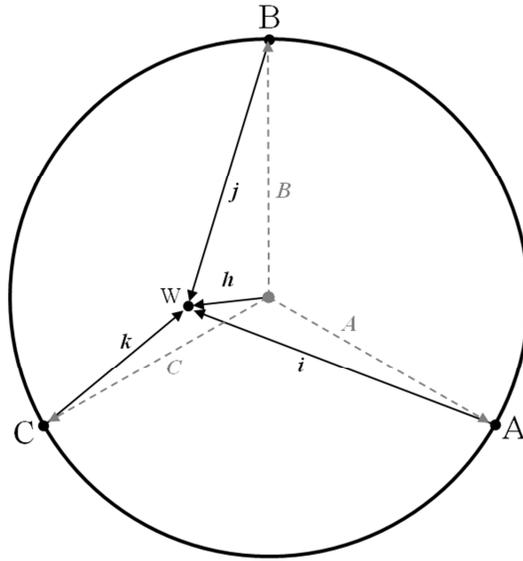


Figure A.3 Geometric determination of wire length based on desired position.

It stands, therefore, that,

$$\begin{aligned}
 \vec{A} &= (r \sin 60, -r \cos 60) \\
 \vec{B} &= (0, r) \\
 \vec{C} &= (-r \sin 60, -r \cos 60) \\
 \vec{i} &= \vec{h} - \vec{A} \\
 \vec{j} &= \vec{h} - \vec{B} \\
 \vec{k} &= \vec{h} - \vec{C}
 \end{aligned}
 \tag{A.1}$$

The positioning mounts may then be adjusted to achieve the desired wire lengths.

A.3 Determining actual position from wire length

Similarly, the actual position of the injection tube may be determined by a given wire length.

Radial symmetry constraints require only two of three wire lengths to obtain a solution.

Given two wires i, k , the injection tube position W may be determined by application of the law of cosines, as shown in Figure A.4.

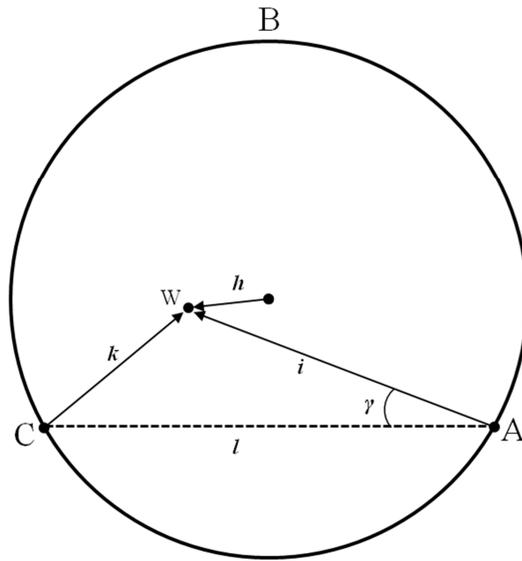


Figure A.4 Geometric determination of position based on wire length.

It stands therefore, that,

$$\begin{aligned}l &= 2r \sin 60 \\k^2 &= i^2 + l^2 - 2il \cos \gamma \\ \cos \gamma &= \frac{i^2 + l^2 - k^2}{2il}\end{aligned}\tag{A.2}$$

$$i_x = i \cos \gamma = i \left(\frac{i^2 + l^2 - k^2}{2il} \right)$$

$$i_y = \sqrt{i^2 - i_x^2}$$

$$\vec{i} = (-i_x, i_y)$$

$$\vec{h} = \vec{A} + \vec{i}$$

Ideally, the same result is obtained using the other two pairs of wires (j, k and j, i).

A.4 Considerations

Though the initial math is trivial, nonlinearities present within the system (slack, measurement irregularities) turn these calculations into an iterative process and more significantly complicate the determination of the actual injection tube position. In practice, desired wire lengths were calculated based on a desired position, then each wire was further adjusted until taut to remove any slack. The new wire lengths were then used to determine the actual injection tube position; each of the three wire pairs were used to calculate the position, then averaged together.

Appendix B

Magnetic mixer design

B.1 Introduction

To maintain microspheres in suspension during injection, a syringe mixing mechanism was constructed based on the principles used in magnetic stir plates. In short, a simple stator was wound and placed around the syringe. A permanent magnet rotor was inserted into the syringe before drawing in the microsphere suspension. By toggling the magnetic field direction of the stator, the permanent magnet-based rotor experienced a torque that tried to align the magnetic poles with the induced field, flipping the rotor back and forth, mixing the suspension.

B.2 Theory

Biot-Savart law for current-induced magnetic field of the stator:

$$B = \int \frac{\mu_0}{4\pi} \frac{Idl \times \hat{r}}{R^2} \quad (\text{B.1})$$

Or simply,

$$B = \frac{\mu_0 NI}{2R} \quad (\text{B.2})$$

where R is the radius of the coil, I is the current, N is the number of turns.

The torque applied to a magnetically permeable material in a magnetic field can be expressed as,

$$\tau = m \times B \quad (\text{B.3})$$

where m is the magnetic dipole moment (volume integral of the magnetization vector across a magnetized material) and is dependent on material.

Thus, alternating the direction of I causes the direction of B to toggle, which induces a similarly alternating torque on the rotor, flipping the disc around to align with B, as shown in Figure B.1.

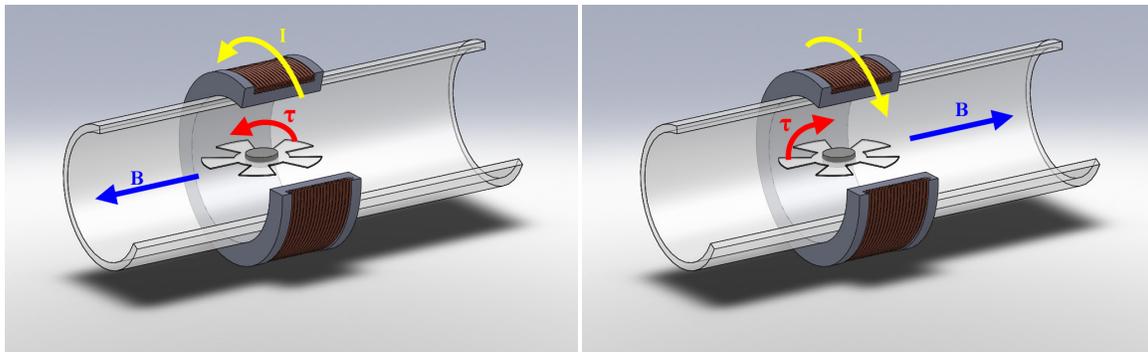


Figure B.1 Mixing mechanism. By toggling the current direction (I), the magnetic field direction (B) is flipped, which induces an alternating torque (τ) on the mixing rotor.

B.3 Design

A simple stator was wound by hand around a plastic frame. The stator consisted of 25 layers of 36 AWG magnet wire (0.005" diameter), with each layer composed of approximately 60 windings. The DC resistance of the stator was measured as 80 Ω .

The rotor was composed of a small NdFeB permanent magnet disc contained between two layers of cellophane tape. The tape was cut into a star pattern to facilitate mixing. The stator attached to a syringe, and associated rotor, are shown in Figure B.2.

A MOSFET-based H-bridge circuit, powered by a 5 VDC power supply, was constructed and connected to a microcontroller circuit. The microcontroller randomly toggled the current direction through the stator in the interval 0.5-3 seconds. Random intervals were used to discourage predictable fluid current patterns and instead mix the microsphere suspension in an erratic manner.

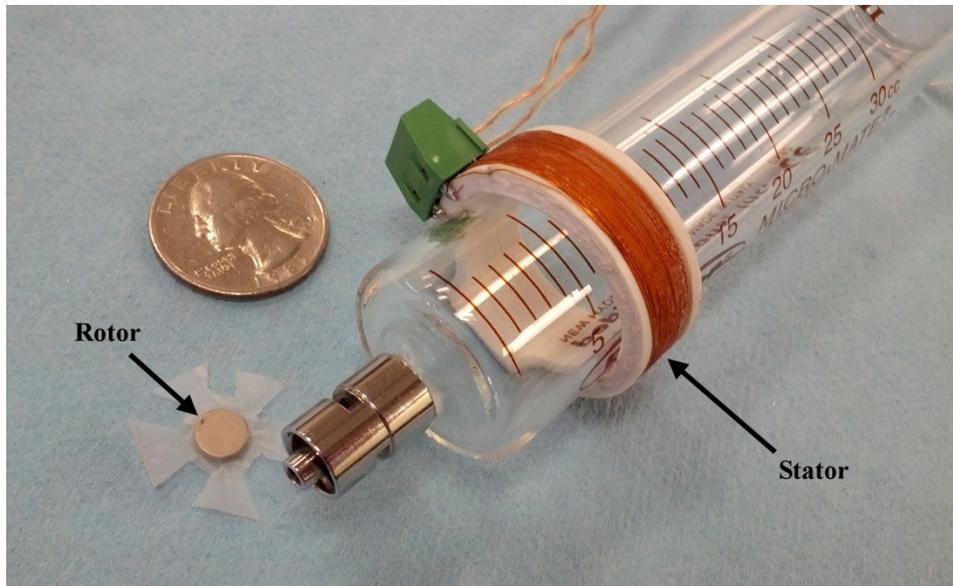


Figure B.2 Syringe mixing components. The stator is fits over a glass syringe. The star-shaped rotor is placed within the syringe before filling with the microsphere suspension. Quarter shown for size reference.

Appendix C

Particle injection rate determination

To minimize flow disturbances during microsphere injections, it was desired that the average fluid velocity exiting the injection tube should equal the velocity of the surrounding fluid in the inlet tube. This velocity value was not the same at all points in the cross section of the inlet tube, however. Instead, consideration was taken to match laminar flow conditions present in the experimental model.

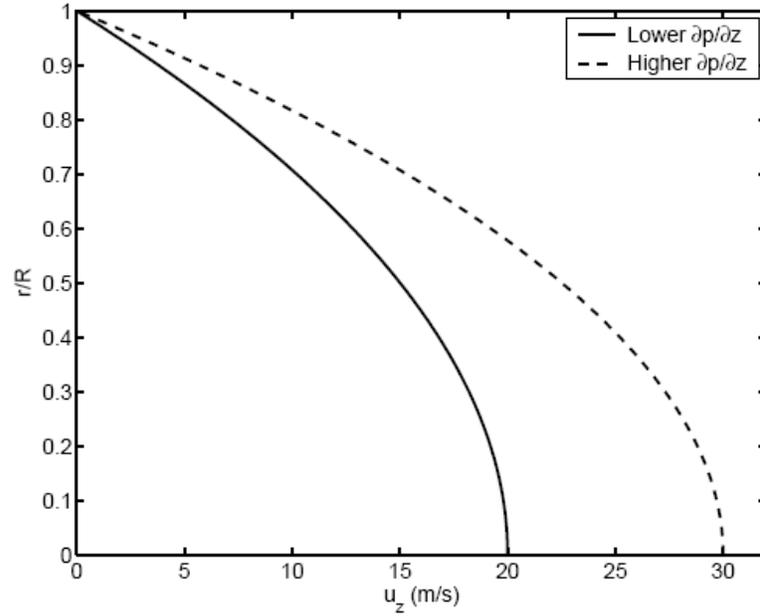


Figure C.1 Parabolic laminar flow profile within a circular tube as a function of forward velocity and distance from the centerline.

C.1 Analysis

To determine the correct syringe injection rate, and thus microsphere injection velocity, first the velocity at a specified location within the inlet tube must be found.

C.1.1 Inlet tube velocity

Consider the Hagen-Poiseuille equation for laminar, parabolic flow as it relates to the model inlet tube, as seen in Figure C.1,

$$u_z = -\frac{1}{4\mu} \frac{dp}{dz} (R^2 - r^2) \quad (\text{C.1})$$

where R is the radius of the inlet tube, and r is the radius to a point of interest.

Because the change in pressure relative to distance is not known, the velocity must instead be related to the mass flow rate of the inlet tube, Q_t . The relationship between flow rate and velocity is governed by the integral,

$$Q_t = \int_0^R u_z(2\pi r) dr \quad (\text{C.2})$$

substituting Equation C. 1 and integrating yields,

$$Q_t = -\frac{\pi R^4}{8\mu} \left(\frac{dp}{dz} \right) \quad (\text{C.3})$$

The target velocity at the injection point can now be written in terms of Q_t ,

$$u_z = \frac{2Q_t}{\pi R^4} (R^2 - r^2) \quad (\text{C.4})$$

C.1.2 Syringe plunger velocity

Now that the target injection velocity has been found, it must be related back to the syringe plunger velocity, V_s . The syringe velocity will depend on the cross sectional area of the syringe, as well as the injection tube. It has already been specified that the average fluid velocity exiting the injection tube, V_i , should equal the velocity at the injection location in the inlet tube, u_z . Furthermore, flow out the injection tube, Q_i , must equal flow from the syringe, Q_s .

$$\begin{aligned} u_z &= V_i \\ Q_i &= Q_s \end{aligned} \quad (\text{C.5})$$

The velocity of the syringe, therefore, is equal to the ratio of injection tube cross-sectional area to syringe cross-sectional area times the desired injection velocity,

$$V_s = \frac{Q_s}{A_s} = \frac{Q_i}{A_s} = \frac{V_i A_i}{A_s} \quad (\text{C.6})$$

Substituting the parabolic flow compensated velocity equation from Equation C. 1 yields the final relationship between injection tube location and required syringe velocity.

$$V_s = \frac{2Q_t A_i}{\pi R^4 A_s} (R^2 - r^2) \quad (\text{C.7})$$

C.2 Assumptions

The primary assumptions used in the analysis of laminar flow are as follows:

1. Steady laminar flow $\left(\frac{du}{dt} = 0\right)$, which also implies no radial or swirl velocity components ($u_r = u_\theta = 0$).
2. Axisymmetric flow $\left(\frac{du}{d\theta} = 0\right)$.
3. Fully developed flow $\left(\frac{du}{dz} = 0\right)$.

Appendix D

Injection tip deflection

D.1 Introduction

During analysis of experiment targeting results present in Chapter 3, a vertical (y-axis) offset was noticed when comparing with observed with expected (simulation) microsphere distributions (see Figure D.1). The injection tube was supported at two locations along its length (see Figure D.2), which still allowed first order bending, and could explain a downward tip deflection, depending on the location of the supports. Additionally, the mass of the flexible supply tube (connected to the back of the injection tube) imparted a downward force which would further emphasize a downward tip deflection. To determine whether or not this tip deflection was significant, Euler-Bernoulli beam theory was applied to the beam.

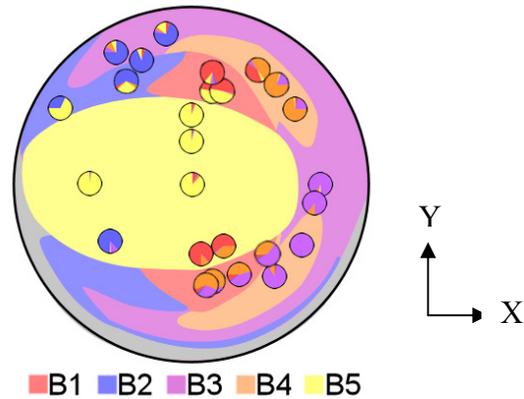


Figure D.1 Results from experimental injections overlaid on the simulation injection map. Note the vertical offset in the actual vs expected distribution.

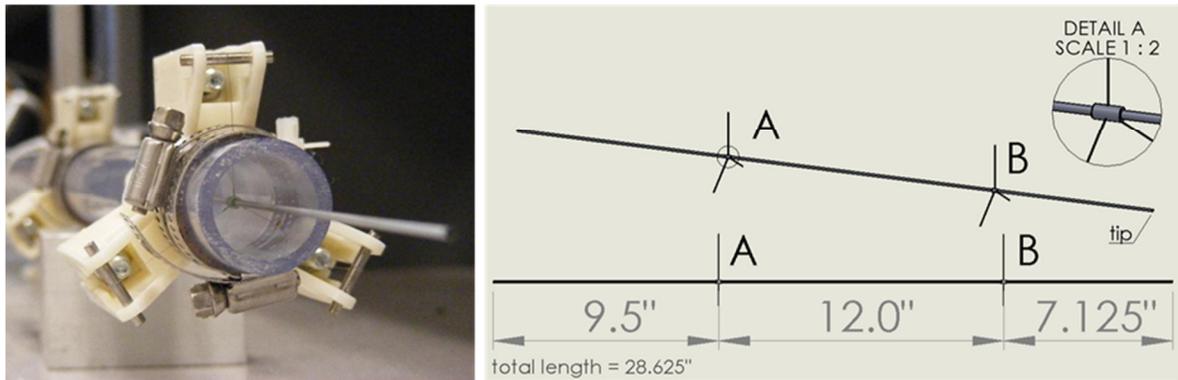


Figure D.2 Injection tube support mechanism (left) and dimensions (right).

D.2 Bending analysis

Using the principle of super-positioning, the analysis was split into two parts – the first part considered only the weight of the injection tube itself on the bending, and the second part considered only the weight of the applied force from the attached flexible tubing.

D.2.1 Injection tube weight

The injection tube weight can be considered a distributed load acting on a beam overhanging two supports, with unequal overhangs. Consider the following free-body diagram.

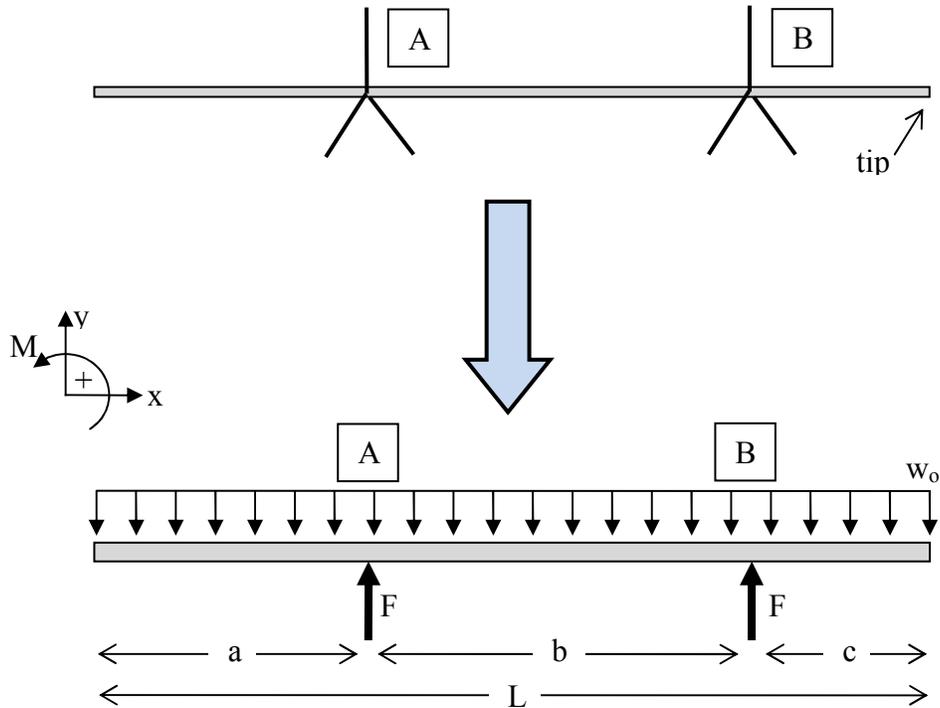


Table D.1 Symbolic parameters

Symbol	Unit	Description
w_0	N/m	Force distribution along tube
a	m	Distance from left end to A
b	m	Distance between A and B
c	m	Distance from B to right end
L	m	Total length of tube
F_A	N	Reaction force at A (unknown)
F_B	N	Reaction force at B (unknown)

First, the reaction forces at the two support locations, A and B, are found.

$$\Sigma F_y = 0 = F_A + F_B - w_0L$$

$$\Sigma M_z = 0 = F_A a + F_B(a + b) - w_0L\left(\frac{L}{2}\right)$$

Therefore,

$$F_A = \frac{wL(L - 2c)}{2b} \quad (D.1)$$

$$F_B = \frac{wL(L - 2a)}{2b} \quad (D.2)$$

Now the internal moments may be analyzed, starting from the left side. If the loading conditions change along the span of beam, there is corresponding change in moment equation. In the case of the injection tube, there are 3 different sections.

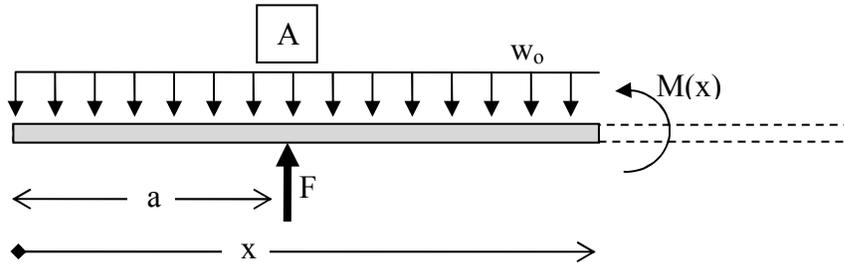
Section 1: $0 < x < a$



$$\Sigma M_z = 0 = M(x) + w_0x\left(\frac{x}{2}\right)$$

$$M(x) = -\frac{w_0x^2}{2} \quad (D.3)$$

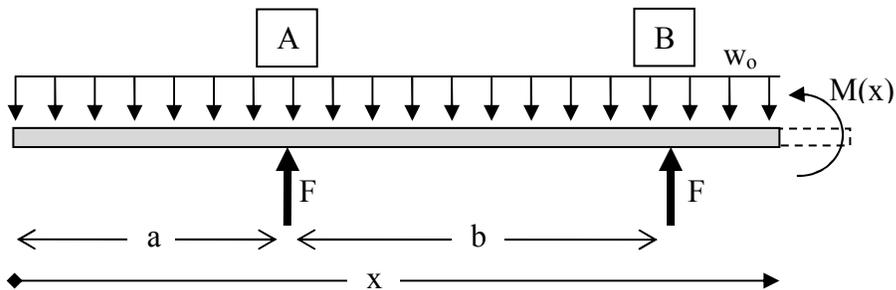
Section 2: $a < x < (a + b)$



$$\Sigma M_z = 0 = M(x) + w_0 x \left(\frac{x}{2} \right) - F_A(x - a)$$

$$\therefore M(x) = -\frac{w_0 x^2}{2} + xF_A - F_A a \quad (D.4)$$

Section 3: $(a + b) < x < L$



$$\Sigma M_z = 0 = M(x) + w_0 x \left(\frac{x}{2} \right) - F_A(x - a) - F_B(x - a - b)$$

$$\therefore M(x) = -\frac{w_0 x^2}{2} + x(F_A + F_B) - F_A a - F_B(a + b) \quad (D.5)$$

The relationship between bending moment and radius of curvature (ρ) is,

$$\frac{1}{\rho} = \frac{M(x)}{EI} \quad (D.6)$$

where E is the modulus of elasticity, I is the area moment of inertia. Generally, EI is considered the flexural rigidity.

From elementary calculus and geometry, we know

$$\frac{1}{\rho} = \frac{\frac{d^2y}{dx^2}}{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}}$$

where y is the deflection at any position, x, along the tube. For small deflections, the slope $\frac{dy}{dx}$ will be small, and $\left(\frac{dy}{dx}\right)^2$ will be negligible. Thus,

$$\frac{1}{\rho} \cong \frac{d^2y}{dx^2} \quad (\text{D.7})$$

And therefore, from (D.6) and (D.7),

$$EI \frac{d^2y}{dx^2} = EIy'' = M(x) \quad (\text{D.8})$$

$$EI \frac{dy}{dx} = EIy' = \int_0^x M(x) dx + C_1$$

$$EIy = \iint_0^x M(x) dx + C_1x + C_2$$

The constants C_1 and C_2 will be determined by the boundary conditions (B.C.'s) from each segment.

Now we apply (D.8) to our internal moment equations (D.3), (D.4), (D.5).

Section 1: $0 < x < a$

$$\begin{aligned}
EIy_1'' &= M_1(x) = -\frac{w_0x^2}{2} \\
EIy_1' &= -\frac{w_0x^3}{6} + C_{11} \\
EIy_1 &= -\frac{w_0x^4}{24} + xC_{11} + C_{12}
\end{aligned} \tag{D.9}$$

Section 2: $a < x < (a + b)$

$$\begin{aligned}
EIy_2'' &= -\frac{w_0x^2}{2} + xF_A - F_Aa \\
EIy_2' &= -\frac{w_0x^3}{6} + \frac{x^2}{2}F_A - xF_Aa + C_{21} \\
EIy_2 &= -\frac{w_0x^4}{24} + \frac{x^3}{6}F_A - \frac{x^2}{2}F_Aa + xC_{21} + C_{22}
\end{aligned} \tag{D.10}$$

Section 3: $(a + b) < x < L$

$$\begin{aligned}
EIy_3'' &= -\frac{w_0x^2}{2} + x(F_A + F_B) - F_Aa - F_B(a + b) \\
EIy_3' &= -\frac{w_0x^3}{6} + \frac{x^2}{2}(F_A + F_B) - x(F_Aa + F_B(a + b)) + C_{31} \\
EIy_3 &= -\frac{w_0x^4}{24} + \frac{x^3}{6}(F_A + F_B) - \frac{x^2}{2}(F_Aa + F_B(a + b)) + xC_{31} \\
&\quad + C_{32}
\end{aligned} \tag{D.11}$$

To solve for the constants, boundary conditions are applied. Because there are 6 unknowns, 6 equations are needed.

Due to the support wires, the following conditions are established,

$$\text{at } x = a, \quad \begin{cases} y_1 = 0 \\ y_2 = 0 \end{cases}$$

$$\text{at } x = (a + b), \quad \begin{cases} y_2 = 0 \\ y_3 = 0 \end{cases}$$

Continuity of slope requirement for the beam dictates that,

$$\text{at } x = a, \quad y'_1 = y'_2$$

$$\text{at } x = (a + b), \quad y'_2 = y'_3$$

With all 6 equations, a series of linear equations can be established.

Starting with section 2, since its B.C.'s are independent of the other sections,

$$\text{at } x = a, \quad y_{2_{x=a}} = 0 = -\frac{w_0 a^4}{24} + \frac{a^3}{6} F_A - \frac{a^2}{2} F_A a + a C_{21} + C_{22}$$

$$\therefore a C_{21} + C_{22} = \frac{w_0 a^4}{24} - \frac{a^3}{3} F_A$$

$$\text{at } x = (a + b), \quad y_{2_{x=(a+b)}} = 0$$

$$= -\frac{w_0 (a + b)^4}{24} + \frac{(a + b)^3}{6} F_A - \frac{(a + b)^2}{2} F_A a + (a + b) C_{21} + C_{22}$$

$$\therefore (a + b) C_{21} + C_{22} = \frac{w_0 (a + b)^4}{24} - \frac{(a + b)^3}{6} F_A + \frac{(a + b)^2}{2} F_A a$$

This can be rewritten in matrix form,

$$\begin{bmatrix} a & 1 \\ a+b & 1 \end{bmatrix} \begin{bmatrix} C_{21} \\ C_{22} \end{bmatrix} = \begin{bmatrix} \frac{w_0 a^4}{24} - \frac{a^3}{3} F_A \\ \frac{w_0 (a+b)^4}{24} - \frac{(a+b)^3}{6} F_A + \frac{(a+b)^2}{2} F_A a \end{bmatrix} \quad (\text{D.12})$$

C_{21} and C_{22} may be computed using a linear solver (e.g. using MATLAB *linsolve* function). Values for section 1 and section 3 may then be solved.

For section 1,

$$\begin{aligned} \text{at } x = a, \quad y_{1_{x=a}} = 0 &= -\frac{w_0 a^4}{24} + aC_{11} + C_{12} \\ \therefore aC_{11} + C_{12} &= \frac{w_0 a^4}{24} \end{aligned}$$

$$\text{at } x = a, \quad y'_{1_{x=a}} = y'_{2_{x=a}}$$

$$\begin{aligned} y'_{1_{x=a}} = -\frac{w_0 a^3}{6} + C_{11} &= -\frac{w_0 a^3}{6} + \frac{a^2}{2} F_A - aF_A a + C_{21} = y'_{2_{x=a}} \\ \therefore C_{11} &= -\frac{a^2}{2} F_A + C_{21} \end{aligned}$$

In matrix form,

$$\begin{bmatrix} a & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{12} \end{bmatrix} = \begin{bmatrix} \frac{w_0 a^4}{24} \\ -\frac{a^2}{2} F_A + C_{21} \end{bmatrix} \quad (\text{D.13})$$

For section 3,

$$\begin{aligned} \text{at } x = (a+b), \quad y_{3_{x=(a+b)}} &= 0 \\ &= -\frac{w_0 (a+b)^4}{24} + \frac{(a+b)^3}{6} (F_A + F_B) - \frac{(a+b)^2}{2} (F_A a + F_B (a+b)) + (a+b)C_{31} + C_{32} \end{aligned}$$

$$\therefore (a+b)C_{31} + C_{32} = \frac{w_0(a+b)^4}{24} - \frac{(a+b)^3}{6}(F_A + F_B) + \frac{(a+b)^2}{2}(F_A a + F_B(a+b))$$

$$\text{at } x = (a+b), \quad y'_{3x=a+b} = y'_{2x=a+b}$$

$$y'_{3x=a} = -\frac{w_0(a+b)^3}{6} + \frac{(a+b)^2}{2}(F_A + F_B) - (a+b)(F_A a + F_B(a+b)) + C_{31}$$

$$= -\frac{w_0(a+b)^3}{6} + \frac{(a+b)^2}{2}F_A - (a+b)F_A a + C_{21} = y'_{2x=a}$$

$$\therefore C_{31} = \frac{(a+b)^2}{2}F_B + C_{21}$$

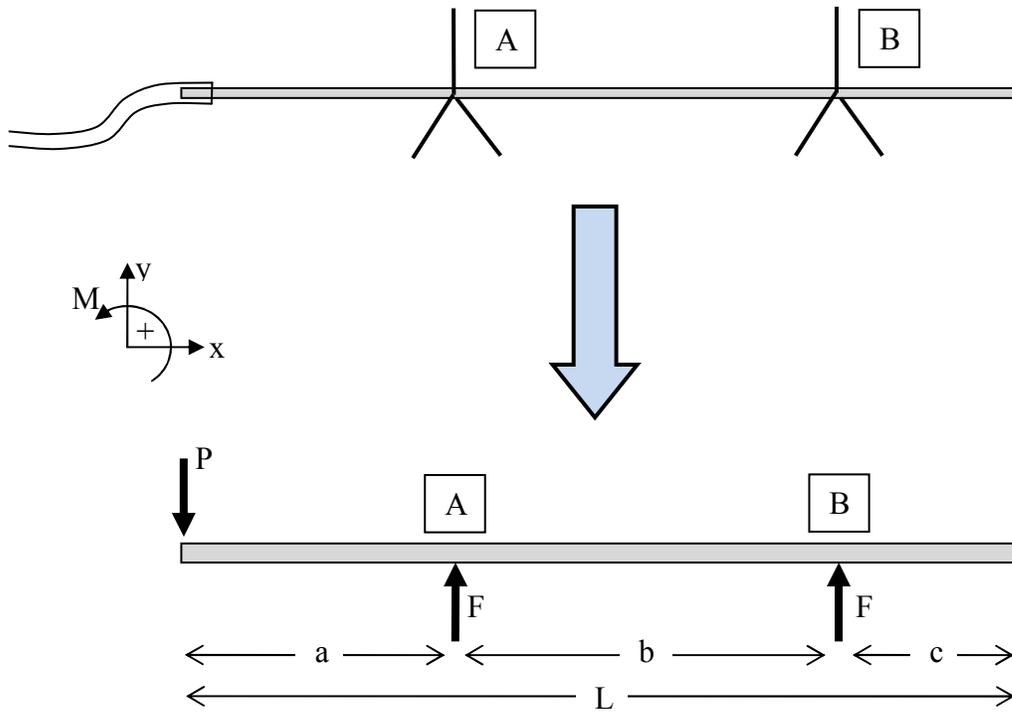
In matrix form,

$$\begin{bmatrix} a+b & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} C_{31} \\ C_{32} \end{bmatrix} = \begin{bmatrix} \frac{w_0(a+b)^4}{24} - \frac{(a+b)^3}{6}(F_A + F_B) + \frac{(a+b)^2}{2}(F_A a + F_B(a+b)) \\ \frac{(a+b)^2}{2}F_B + C_{21} \end{bmatrix} \quad (\text{D.14})$$

Equations (D.12), (D.13), and (D.14) define the constants. Using equations (D.9), (D.10), and (D.11), the deflection in each section can be found.

D.2.2 Apparent force from flexible tube

The same steps may be applied to determine the deflection from the flexible tube attached to one end of the injection tube (connecting to the syringe pump). Consider the following free-body diagram,



Where P is the apparent force from the attached flexible tube.

$$\Sigma F_y = 0 = F_A + F_B - P$$

$$\Sigma M_z = 0 = F_A a + F_B (a + b)$$

Therefore,

$$F_A = -\frac{Pa}{b} \tag{D.15}$$

$$F_B = P \left(\frac{a+b}{b} \right) \tag{D.16}$$

The internal moments, progressing from left to right, are as follows,

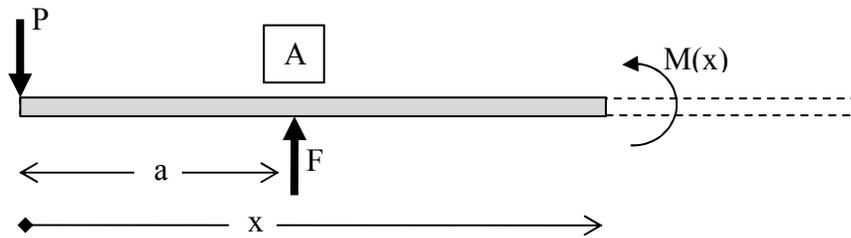
Section 1: $0 < x < a$



$$\Sigma M_z = 0 = M(x) + xP$$

$$\therefore M(x) = -xP \quad (D.17)$$

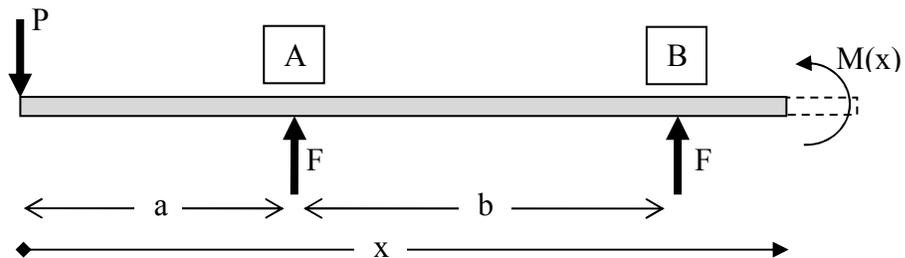
Section 2: $a < x < (a + b)$



$$\Sigma M_z = 0 = M(x) + xP - F_A(x - a)$$

$$\therefore M(x) = x(F_A - P) - F_A a \quad (D.18)$$

Section 3: $(a + b) < x < L$



$$\Sigma M_z = 0 = M(x) + xP - F_A(x - a) - F_B(x - a - b)$$

$$\therefore M(x) = 0 \quad (\text{D.19})$$

The constants C_1 and C_2 are set by the boundary conditions (B.C.'s) from each segment.

Equation (D.8) may be applied to the internal moment equations (D.17), (D.18), (D.19).

Section 1: $0 < x < a$

$$EIy_1'' = M_1(x) = -xP$$

$$EIy_1' = -\frac{x^2}{2}P + C_{11}$$

$$EIy_1 = -\frac{x^3}{6}P + xC_{11} + C_{12} \quad (\text{D.20})$$

Section 2: $a < x < (a + b)$

$$EIy_2'' = x(F_A - P) - F_A a$$

$$EIy_2' = \frac{x^2}{2}(F_A - P) - xF_A a + C_{21}$$

$$EIy_2 = \frac{x^3}{6}(F_A - P) - \frac{x^2}{2}F_A a + xC_{21} + C_{22} \quad (\text{D.21})$$

Section 3: $(a + b) < x < L$

$$EIy_3'' = 0$$

$$EIy_3' = C_{31}$$

$$EIy_3 = xC_{31} + C_{32} \quad (D.22)$$

Applying B.C.'s for the three segments,

For Section 2,

$$\text{at } x = a, \quad y_{2_{x=a}} = 0 = \frac{a^3}{6}(F_A - P) - \frac{a^2}{2}F_A a + aC_{21} + C_{22}$$

$$\therefore aC_{21} + C_{22} = -\frac{a^3}{6}(F_A - P) + \frac{a^2}{2}F_A a$$

$$\text{at } x = (a + b), \quad y_{2_{x=(a+b)}} = 0 = \frac{(a + b)^3}{6}(F_A - P) - \frac{(a + b)^2}{2}F_A a + (a + b)C_{21} + C_{22}$$

$$\therefore (a + b)C_{21} + C_{22} = -\frac{(a + b)^3}{6}(F_A - P) + \frac{(a + b)^2}{2}F_A a$$

Now in matrix form,

$$\begin{bmatrix} a & 1 \\ a + b & 1 \end{bmatrix} \begin{bmatrix} C_{21} \\ C_{22} \end{bmatrix} = \begin{bmatrix} -\frac{a^3}{6}(F_A - P) + \frac{a^2}{2}F_A a \\ -\frac{(a + b)^3}{6}(F_A - P) + \frac{(a + b)^2}{2}F_A a \end{bmatrix} \quad (D.23)$$

For section 1,

$$\text{at } x = a, \quad y_{1_{x=a}} = 0 = -\frac{a^3}{6}P + aC_{11} + C_{12}$$

$$\therefore aC_{11} + C_{12} = \frac{a^3}{6}P$$

$$\text{at } x = a, \quad y'_{1_{x=a}} = y'_{2_{x=a}}$$

$$y'_{1_{x=a}} = -\frac{a^2}{2}P + C_{11} = \frac{a^2}{2}(F_A - P) - aF_A a + C_{21} = y'_{2_{x=a}}$$

$$\therefore C_{11} = -\frac{a^2}{2}F_A + C_{21}$$

In matrix form,

$$\begin{bmatrix} a & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} C_{11} \\ C_{12} \end{bmatrix} = \begin{bmatrix} \frac{a^3}{6}P \\ -\frac{a^2}{2}F_A + C_{21} \end{bmatrix} \quad (\text{D.24})$$

For section 3,

$$\text{at } x = (a + b), \quad y_{3_{x=(a+b)}} = 0 = (a + b)C_{31} + C_{32}$$

$$\therefore (a + b)C_{31} + C_{32} = 0$$

$$\text{at } x = (a + b), \quad y'_{3_{x=a+b}} = y'_{2_{x=a+b}}$$

$$y'_{3_{x=a}} = C_{31} = \frac{(a + b)^2}{2}(F_A - P) - (a + b)F_A a + C_{21} = y'_{2_{x=a}}$$

$$\therefore C_{31} = \frac{(a + b)^2}{2}F_B + C_{21}$$

In matrix form,

$$\begin{bmatrix} a + b & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} C_{31} \\ C_{32} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{(a + b)^2}{2}F_B + C_{21} \end{bmatrix} \quad (\text{D.25})$$

Equations (D.23), (D.24), and (D.25) define the constants. Then, using Equations (D.20), (D.21), and (D.22), the deflection in each section can be found.

D.3 Material specifications

The injection tube was composed of a 28.625" length of 316 stainless steel (SS) hypodermic tubing (sourced via smallparts.com). The flexible supply tube was Tygon brand vinyl tubing. Manufacturer supplied material specifications for the materials used in the injection tube and

supply tube are listed in Table D.2 and D.3. Standard 316 SS material properties were used for values not supplied by the manufacturer.

Table D.2 Injection tube manufacturer specifications

Outer diameter	0.072 in
Inner Diameter	0.054 in
Wall thickness	0.009 in
Material	316 stainless steel
Density	8000 kg/m ³
Elastic modulus	193 GPa

Table D.3 Flexible supply tube manufacturer specifications

Outer diameter	0.125 in
Inner Diameter	0.0625 in
Wall thickness	0.03125 in
Material	Tygon B-44-3 vinyl
Specific gravity	1.20 (@ 73°F)
Max Length	5 in

To verify the specifications, and fill in any missing values, the properties of each material were also tested experimentally.

D.3.1 Injection tube

Using the volume density for 316 SS, the theoretical injection tube linear mass density was calculated to be 0.0919 g/cm, suggesting a total mass of 2.63 g. When measured with a scale, however, the injection tube weighed just 1.73 g, suggesting a linear mass density of 0.0605 g/cm. The experimentally measured ID and OD of the injection tube matched manufacturer specification, which suggested the mass density of the tube differed

from that of pure 316 SS. Further investigation revealed that the tubes were manufactured not by extrusion, but instead, the tube was drawn and welded and thus was not pure 316 SS. For the bending analysis, the experimentally derived linear mass density of 0.0605 g/cm was used.

The flexural rigidity (EI) of the tube was also determined experimentally using a specialized cantilever tool used to measure beam deflection. The tool contained a clamp at one end, and a dial micrometer at the other. A length of hypodermic tube was clamped and known masses were hung of the end. The vertical deflection due to each mass was recorded at a specified distances from the clamp. The process was repeated for 3 lengths of tubing and the resulting modules values averaged.

Given a concentrated load, P , applied at a distance a along a cantilevered beam, and given a vertical deflection of y measured at location x along the beam, the flexural rigidity, EI , may be calculated as,

$$EI = \frac{Px^2}{6y}(3a - x)$$

Following this process, the flexural rigidity of the tube was measured to be 0.05239 Pa·m⁴. Based on the dimensions of the tube, this corresponds to an elastic modulus, E , of 139 GPa.

D.3.2 Flexible tubing

To determine the load applied by the flexible tube, the mass had to be determined. Buoyancy effects also dictated the need for volume density. Unfortunately, the mass was not directly

measured before assembling the injection tube, so it was instead calculated based on linear mass density and length used.

Using the specifications found for Tygon B-44-3 (Table D.3), a theoretical volume density was first calculated based on the manufacturer supplied specific gravity.

$$\rho_{tygon} = s.g.* \rho_{water}$$

$$\rho_{water} = 997 \text{ kg/m}^3$$

$$\rho_{tygon} = 1.2 * 997 \text{ kg/m}^3 = 1196.4 \text{ kg/m}^3$$

Using this theoretical density, the mass of the flexible tube was calculated to be 0.071 g/cm.

The linear mass density was also measure experimentally using a length of identical Tygon tubing. A 49.125” length of Tygon tubing was weighed and found to be 8.734 g, which corresponded to a linear mass density of 0.07 g/cm. This showed a good agreement with what was measured experimentally, suggesting the manufacturer supplied specific gravity (and thus, density) value was correct.

The load, P , applied to the injection tube by the flexible tubing, including buoyancy effects, was calculated as,

$$P = (\rho_{tygon} - \rho_{water}) * Volume_{tube}$$

Where $Volume_{tube}$ was the material volume of the flexible tube in m^3 .

D.4 Estimated tip deflection

Using experimentally derived material properties and the previously describe bending analysis process, the injection tube tip deflection was calculated. Table D.4 lists all physical properties and variables definitions used in analysis.

Table D.4 Physical parameters and variable definitions

Symbol	Value	Unit	Description
a	0.2413	m	Distance from left end to A
b	0.3048	m	Distance between A and B
c	0.1810	m	Distance from B to right end
L	0.7271	m	Total length of tube
E	139	GPa	Elastic modulus for injection tube
ρ_{tube}	5268	kg/m ³	Measured mass density of injection tube
tube_{ID}	1.3716	mm	Inner diameter of injection tube
tube_{OD}	1.8288	mm	Outer diameter of injection tube
w₀	0.0594	N/m	Force distribution of injection tube
ρ_{tygon}	1196	kg/m ³	Density of flexible Tygon tube
tygon_{ID}	1.5875	mm	Inner diameter of flexible tube
tygon_{OD}	3.1750	mm	Outer diameter of flexible tube
tygon_{length}	0.13	m	Max length of Tygon tube hanging off end of injection tube**

*Approximated based on visual inspection

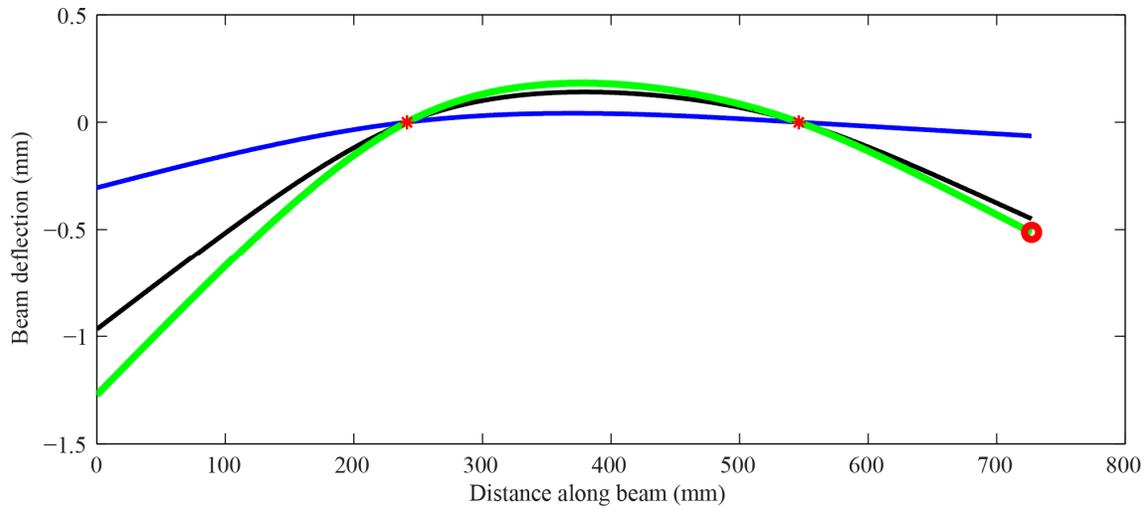


Figure D.3 Amplified injection tube deflection. Total deflection is shown (green line) along with the deflection caused by the weight of the beam (black line) and by the flexible tube attached to the left end of the beam (blue line). The stars indicate the location of the supports. The circle indicates the tip location.

As shown in Figure D.3, the total deflection at the tip of the injection tube was -0.514 mm. This offset can be applied to the injection locations of the original experimental data to correct for tip deflection positioning errors (see Figure D.4).

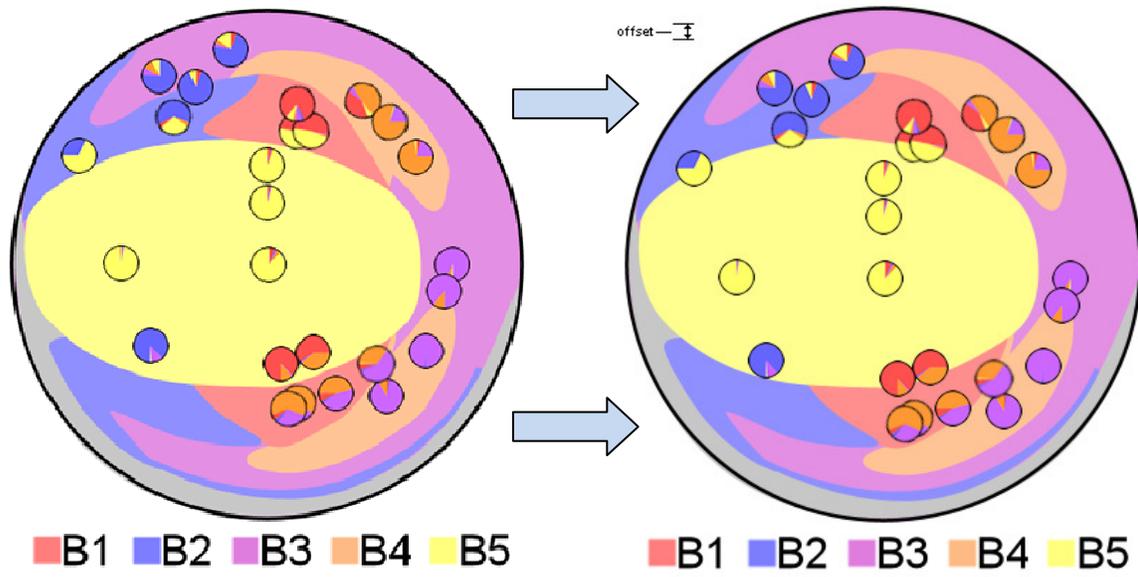


Figure D.4 Original release position results (left) and corrected positions based on injection tube tip deflection (right).

As discussed in Chapter 2, though compensating for tip deflection does aid comparison of experimental and simulation branch distributions, it does not explain all of the discrepancies.

D.5 Assumptions

Various assumptions were made during the calculation of the tip deflection. The first assumption was that pure bending exists and that the injection tube material is linearly elastic. To calculate tip deflection based on moment, the assumption was made that deflections were small and that the square of the slope of the tube was negligible compared to

unity, i.e. $\left[\left(\frac{dy}{dx}\right)^2 \ll 1\right]$. Lastly, the contribution of shear force to the deflection was assumed to be negligible.

Appendix E

Image analysis: Particle detection and sizing

E.1 Introduction

The size distribution analysis described Chapter 3 required unique diameters measurements from a large number of microspheres. An image processing routine was developed to automate the detection and measurement of closely located circular features within a set of micrographs. Optical microscopy was used to collect a large number of microsphere images, which could then be analyzed using the image processing routine. This technique was utilized for the microsphere diameter analysis presented in Chapter 3.

E.2 Image capture

Images were captured using standard bright field microscopy. To maximize the number of microspheres visible in each image, a 5X objective was used in combination with a ProgRes

C5 digital camera (2580x1944 resolution, JENOPTIK Optical Systems GmbH, Jena, Germany). The light level was adjusted for maximum contrast with LBD and ND6 filters enabled (generally around light power level 6). Plan achromatic objectives were used to minimize optical distortion and provide the flattest field of view.

To facilitate the imaging of unique samples, a special slide was developed containing a 5x5 rectangular grid (see Figure E.1). A grid was printed onto a clear transparency sheet and then sealed between two glass slides using PDMS. The grid sections were sized to fit within the microscope field of view (roughly 2.4 mm x 1.8 mm). A diluted microsphere suspension was pipetted onto this grid and covered with a cover slip. The grid area was systematically imaged, one grid section at a time, ensuring unique microspheres in each image. The grid lines provided easy image registration. 30 images were captured for each sample (25 from the inside grid, and 5 from locations outside the grid). A single, highly dense layer of microspheres was desired to maximize the number of spheres in each image. The 1:4 sphere-to-water dilution ratio described in Chapter 3 was chosen as the highest microsphere concentration which still minimized the occurrence of overlapping, or stacked, spheres (see Figure E.2).

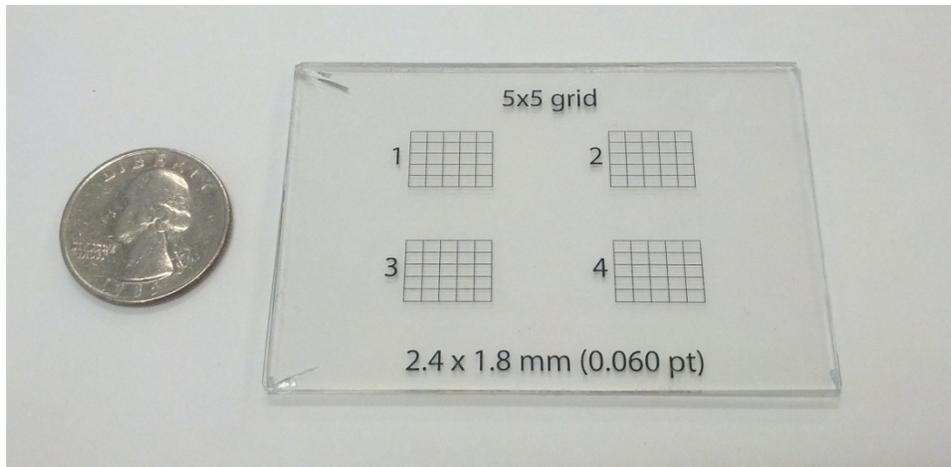


Figure E.1 Custom slide featuring image registration grid. Quarter shown for size reference.

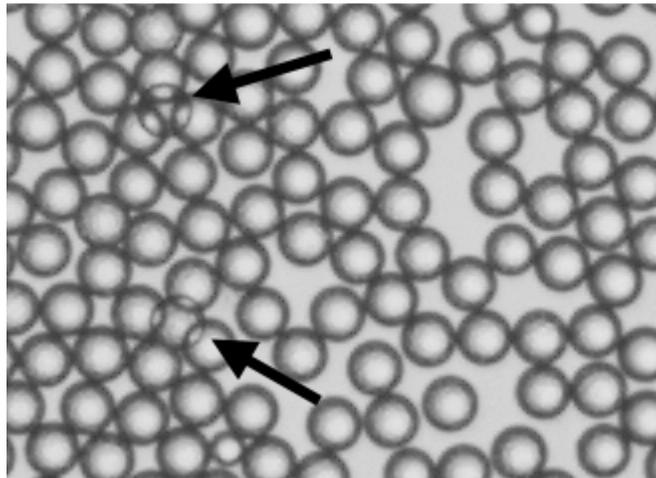


Figure E.2 Micrograph of dense microsphere concentration. Arrows indicate overlapping, stacked microspheres.

E.3 Image analysis

The image analysis method was optimized for translucent microspheres, as they feature optical properties advantageous for individual sphere segmentation in feature dense images.

Captured images of backlit translucent microspheres showed individual spheres with darkened peripheries and light centers. The microsphere acted as an additional lens, focusing light through one area of the sphere. In images with densely grouped microspheres, the bright center aided identification of individual spheres.

E.3.1 Preparation and filtering

An area of interest was defined within each image, isolating the desired microspheres and removing any extraneous areas of the image (such as the grid lines). For the purposes of this explanation, a smaller region will be utilized, as seen in Figure E.3.

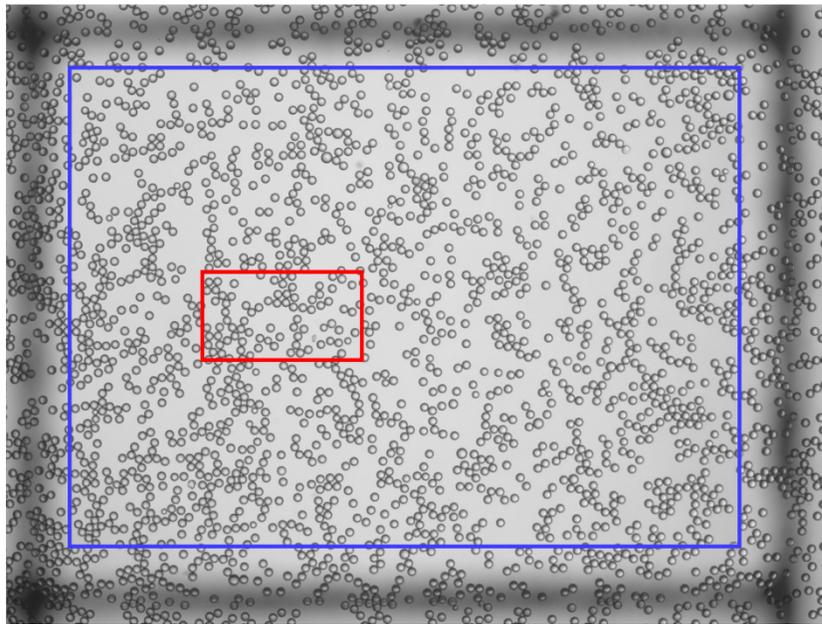


Figure E.3 Initial micrograph with designated areas of interest. The typical analysis area is shown in blue. A smaller, example region is shown in red. Grid borders are visible as blurry black lines on the edges of the image

Prior to images acquisition for each new sample, a blank image was taken with no microspheres. This blank image captured the background lighting conditions and was used to compensate images for variances in the lighting.

First the background image was smoothed using a gaussian filter (5x5, sigma of 0.75), then cropped, and finally subtracted from the microsphere image (Figure E.4). To increase contrast, the intensity curve of the image was adjusted such that the top and bottom 0.1% of pixels were saturated light and dark. The final prepared image is shown in Figure E.5.



Figure E.4 Background subtraction process.

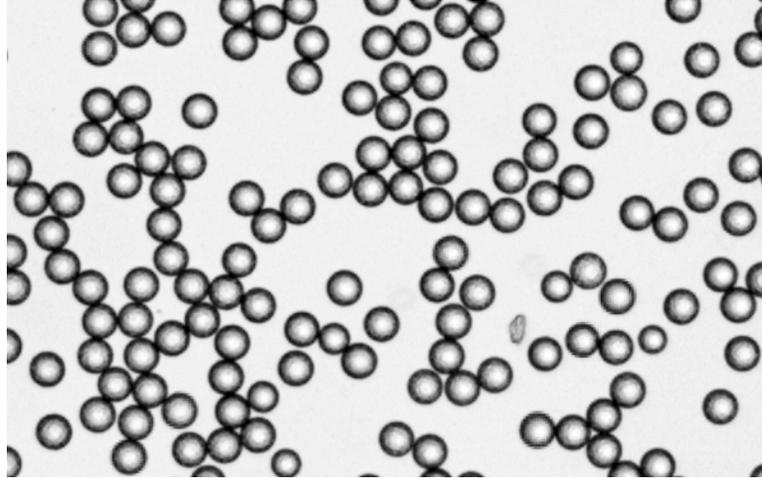


Figure E.5 Contrast enhanced image.

E.3.2 Thresholding and initial identification

A binary threshold was then applied to the image. Values above the threshold were set to white, values below were set to black. The threshold value was set such that the centers of the spheres were solid white areas with black edges. The empty space around spheres was white.

A majority morphological filter was then applied to the image. The value of each pixel was set to the majority value of all immediate neighbors, smoothing jagged edges and removing single pixel noise (see Figure E.6).

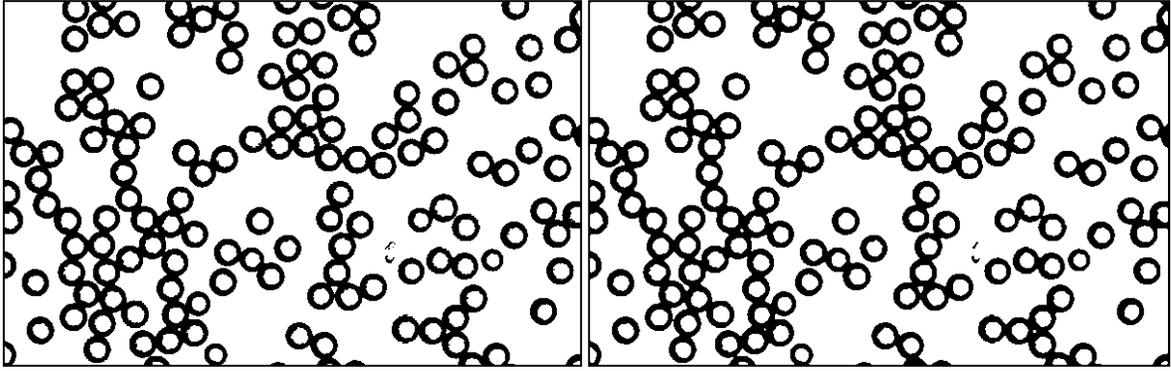


Figure E.6 Thresholded image before (left) and after (right) morphological smoothing.

Next, all connected, contiguous regions in the area were found and labeled. Regions with eliminate excessively large or small areas will filled in black. The remaining white areas were predominantly sphere centers, though some erroneous regions still remained.

Non-circular areas were then eliminated using a roundness criteria. Roundness was defined as the following,

$$roundness = \frac{4\pi * Area}{Perimeter^2}$$

Thus, in a perfect circle the roundness value is 1.

$$roundness = \frac{4\pi * \left(\frac{\pi d^2}{4}\right)}{(\pi d)^2} = 1$$

The roundness of each region was computed using its area and perimeter. Regions have roundness less than a specified value were rejected and filled black. Areas too close to the border of the image were also rejected.

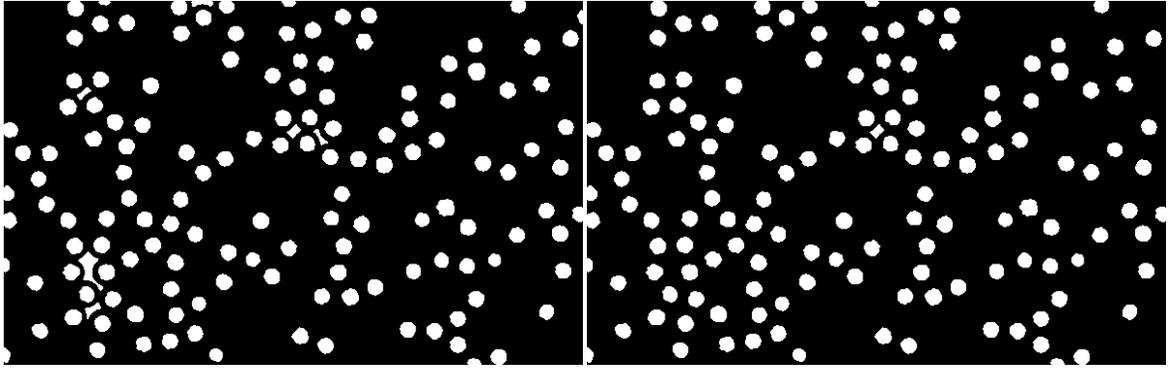


Figure E.7 Binary image following removal of extreme-sized areas (left) and areas with poor roundness (right).

The remaining regions represented potential candidates for sphere locations. The centroids of these regions were determined and served as initial estimates for the each sphere's centroid. Further refinement was necessary to determine whether the region was an actual microsphere, and also to identify the true centroid of the sphere (and not just the centroid of the light region within the sphere, which may be off-center due to the point source illumination).

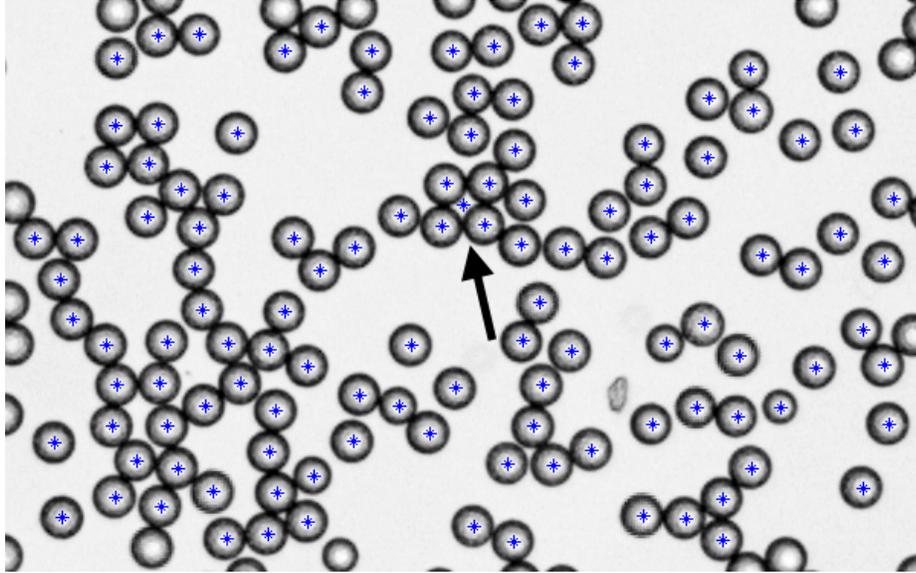


Figure E.8 Image with initial centroid estimates marked in blue. Note the erroneous centroid at the center of the clump indicated by the arrow.

E.3.3 Centroid refinement

The centroid of any circle can be calculated as the average of all pixels along the circumference of the sphere. To isolate the pixels bordering each sphere, the center regions identified in the previous step were subtracted from the initial thresholded image, then the image was reversed. This essentially filled in each sphere white, and left the space in between as black, along with any previously rejected regions.

Next, a morphological reduction was performed on the image, removing interior pixels from connected regions, leaving just the outline of each sphere (see Figure E.9).

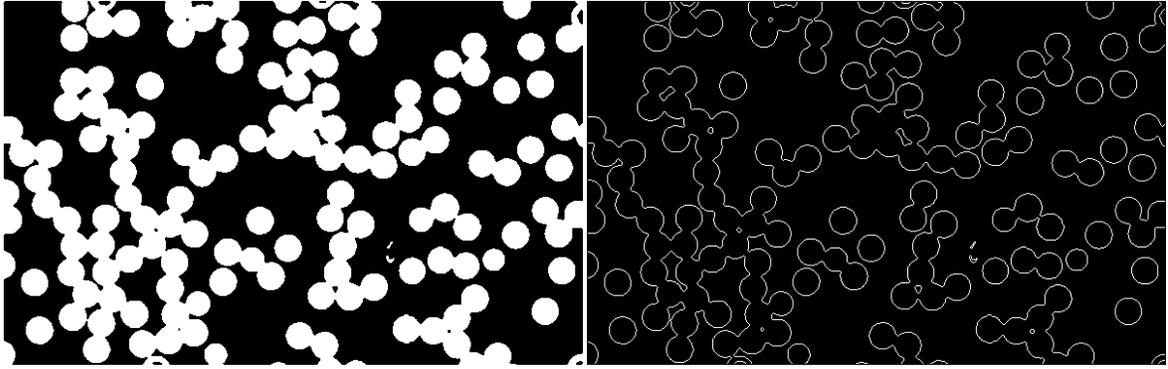


Figure E.9 Inverted thresholded image with candidate regions filled before (left) and after (right) morphological reduction.

For isolated spheres, not connected to any other, the centroid would represent the average of its border pixels, a simple calculation. Most spheres, however, were grouped together and shared border pixels, complicating centroid calculations.

To determine the true centroids, for each potential sphere location, a least-squares minimization function was determined to find the location that minimized the distance between each border pixel, starting at the initial centroid estimate. To avoid unevenly weighting the closest side of the border, the pixels surrounding each centroid were separated into quadrants (upper right, upper left, lower left, lower right). Only the five closest pixels from each quadrant were considered. Quadrants with excessively far pixels were disregarded.

After a sphere's true centroid was calculated, the average pixel diameter was computed based on the same border pixels. The total centroid refinement procedure, as performed on a single sphere's location, is shown in Figure E.10.

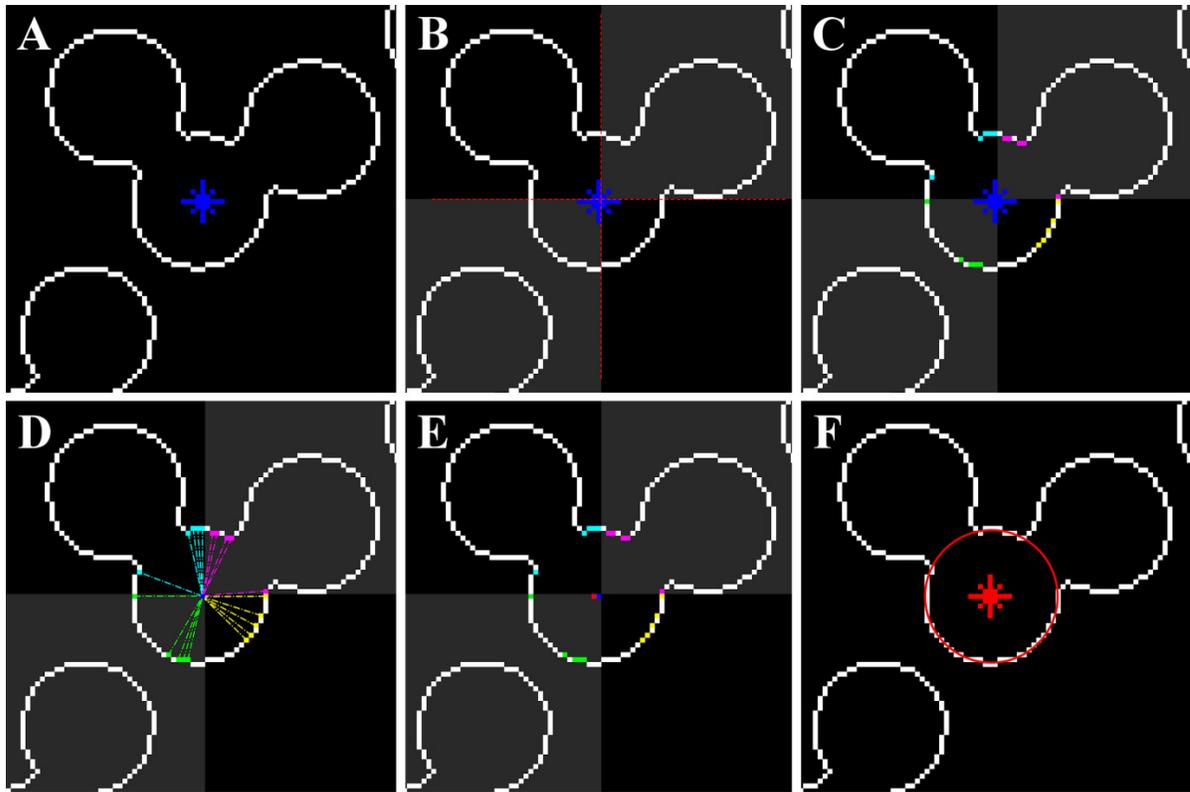


Figure E.10 Composite image subset showing the centroid refinement process, namely: initial centroid estimate (A, in blue), quadrant separation (B), closest pixel detection (C, colored by quadrant), minimization vectors (D), new centroid determination (E, in red), and diameter determination (F).

E.3.4 Invalid region removal

Once the true centroids and diameters were identified, each candidate regions was double checked to ensure that its diameter was not excessively large or small. Overlapping regions were identified by checking that the distance from each centroid to any other centroid was not less than the sum of their radii. Any invalid regions were eliminated.

E.4 Results

The effectiveness of the procedure can be visualized by drawing circles at the final centroid locations with calculated diameters, and overlaying these features on the initial microsphere image, as shown in Figure E.11. The measured diameters show a good fit to the microspheres.

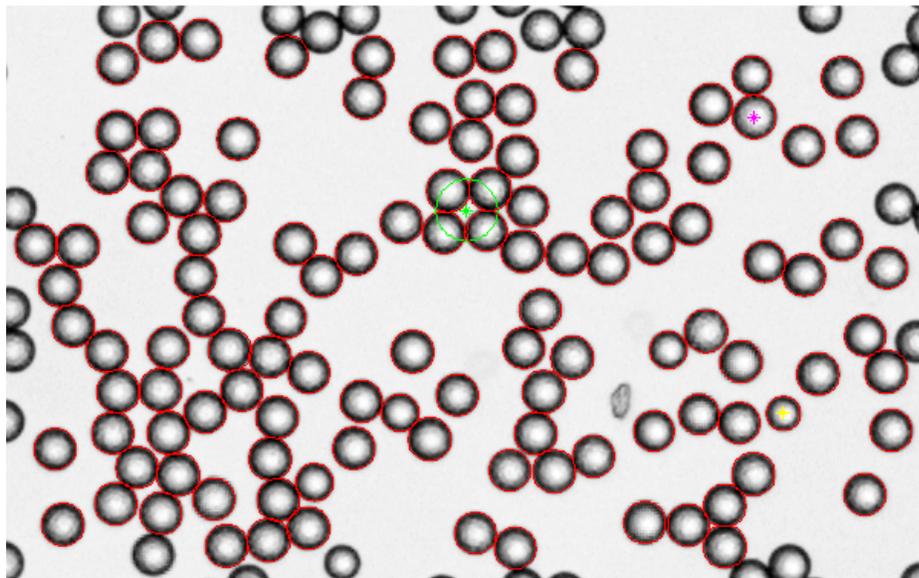


Figure E.11 Initial image overlaid with red circles located at the final. The erroneous region was successfully detect and is marked in green. The largest and smallest diameters found are marked in pink and yellow, respectively.

Analysis of the original full micrograph, with 1555 detected microspheres, is shown in Figure E.12. A total of 15 erroneous areas were detected and removed. Additional analysis results of an alternate micrograph are shown in Figure E.13, but with 2471 detected microspheres.

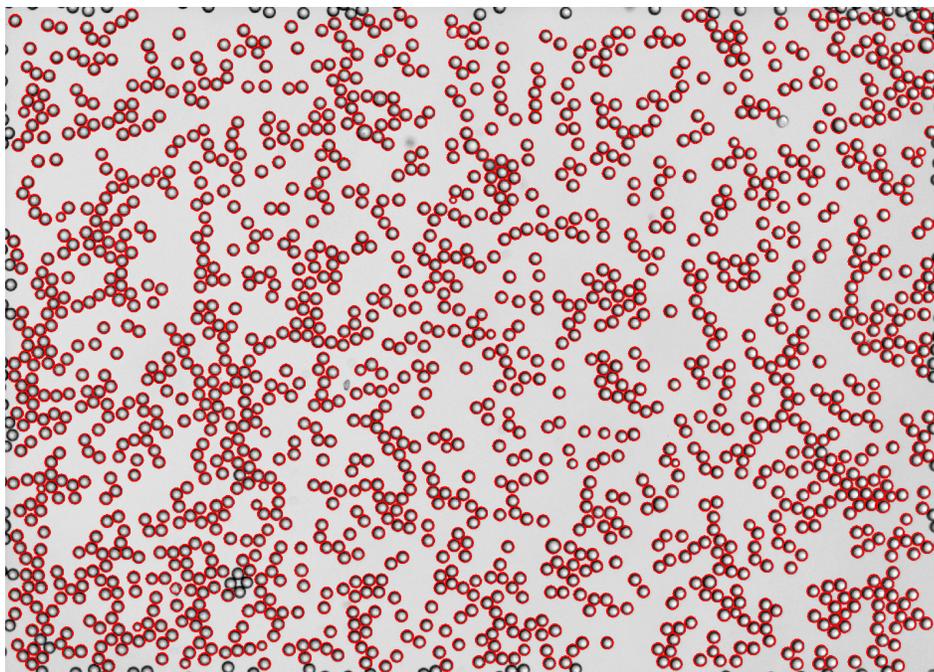


Figure E.12 Full analyzed micrograph. Detected microspheres are denoted in red.

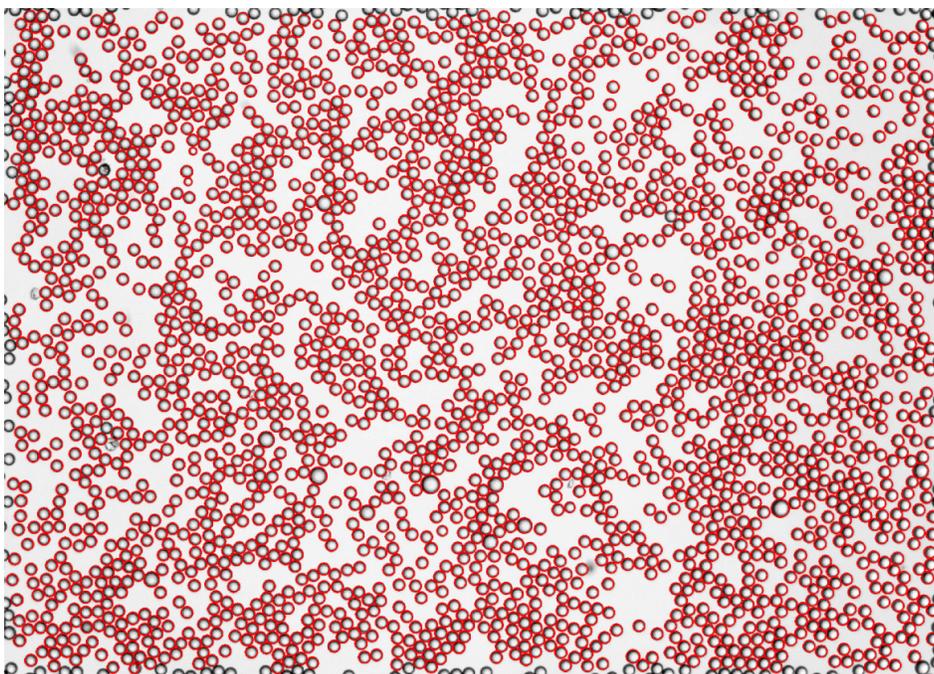


Figure E.13 Sample micrograph after analysis. Detected microspheres are denoted in red.

E.5 Source code

The following pages contain the MATLAB source code (version 7.12.0, R2011a) used to perform the image analysis. The program was designed to be as automated as possible after some preliminary parameters are defined. The user may specify how many images (from however many vials) to process.

The following source files are included:

1. **RunMe_SIR_Generic_Batch.m** – the main script which defines the operations to perform. This is the only script to run, as it calls all the other functions.
2. **AndySizeOptions.m** – a class file defining default analysis parameters and options.
3. **andyfunc_setDimensions.m** – a function used to define the area of interest in each image and write the dimension to file. This function must on each image be run before analysis can proceed.
4. **andyfunc_loadDimensions.m** – a function which loads either the previously defined area of interest dimensions from file, or the default dimension set.
5. **andyfunc_findSpheres_v10.m** – the main function which performs the image analysis, identifying spheres and determining the diameters.
6. **andyfunc_addFigureLegend.m** – a sub-function which draws a nicely formatted figure legend onto analyzed images.
7. **andyfunc_viewSingleSIR_v3.m** – a function which displays the largest and smallest spheres in an image for verification.

8. **andyfunc_combineTotalStatsSIR_v2.m** – a function which loads and combines saved diameter stats from file for all images of all vials.
9. **andyfunc_generateFigureHistogram_v1.m** – a function which displays a nicely formatted histogram and prints it to file , and writes histogram statistics to a file.

RunMe_SIR_Generic_Batch_v2.m

```
%This script configures and runs the microsphere sample analysis. Batch
% operations are supported. Multiple operations are selectable from this
% script, including defining the area of interest per image, running
% the sphere analysis, and viewing/printing multiple histogram types.

%Written by: Andy Richards
%Last modified: Sept 2013
%Department of Mechanical & Aerospace Engineering
%North Carolina State University, Raleigh, NC

clear all;
close all;

%Select which operations to perform
setDims = false;           %Define the area of interest for each image
                           % (and save to a txt file). This MUST be run at
                           % least once for each image.
processFiles = false;     %Search each image for spheres, measure the diam
verifyFiles = false;     %View the largest and smallest spheres to verify
                           % analysis and adjust/reject if needed
viewImageHistogram = false; %Generate histogram per image
viewVialHistogram = false; %Generate histogram per vial
viewFigureHistogram = true; %Generate TOTAL histogram for ALL vials

selectedVials = [1:5];    %Which vials to analyze to use? (which vials)
selectedImages = [1:30]; %Which images to process?
                       % (can be nonconsecutive, e.g. "[1:5,19,24]")

%Define the filepath for each vial's images.
%Filename/path Requirements:
%Images are saved as .png
%Images are numbered sequentially starting from *001.png
% (e.g. "image001.png", "image002.png", etc).
%Images for each vial are located in separate folders
% (e.g. "vial1\image001.png", "vial2\image001.png")
%The complete filepath for an individual image is formed by appending the
% image number to the path listed here (e.g. ".\vial3\image")
fpaths = {...
'..\228.1\228.1_020907_',... %vial 1
'..\228.9\228.9_020910_',... %vial 2
'..\230.0\230.0_020907_',... %3
'..\231.1\231.1_020910_',... %4
'..\232.1\232.1_020814_',... %5
'..\233.1\233.1_020814_',... %6
'..\235.1\235.1_020910_',... %7
'..\236.1\236.1_020910_',... %8
'..\237.1\237.1_020910_',... %9
'..\238.1\238.1_020907_',... %10
```

RunMe_SIR_Generic_Batch_v2.m (continued)

```

'..\239.1\239.1_020910_',...    %11
'..\240.1\240.1_020912_',...    %12
'..\241.1\241.1_020907_',...    %13
'..\242.1\242.1_020911_',...    %14
'..\242.2\242.2_020910_',...    %15
'..\243.1\243.1_020910_',...    %16
'..\243.2\243.2_020912_',...    %17
'..\244.1\244.1_020911_',...    %18
'..\245.1\245.1_020911_',...    %19
'..\245.2\245.2_020907_',...    %20
'..\246.1\246.1_020910_',...    %21
'..\246.2\246.2_020907_',...    %22
'..\246.3\246.3_020910_',...    %23
'..\247.1\247.1_020814_',...    %24
'..\247.2\247.2_020912_',...    %25
'..\247.3\247.3_020911_',...    %26
'..\247.1b\247.1_020912_'};    %vial 27

%Options and parameters (primarily used during image analysis)
%the "opts" class is only used for "processFiles", and "view*Histograms"
opts = AndySizeOptions; %Load default program variables/options
opts.writeStatsFile = 0; %Write csv file containing the diam stats,
    %   centroid locations and histogram stats
opts.printFigs = 0; %Print figures to file (png)
opts.showMainFigs = 1; %Show the primary figures during image analysis
opts.showAllFigs = 0; %Show EVERY figure - this is a LOT of figures!
opts.showHistogram = 0; %Show diameter histogram (if false, just display
    %   the stats)

if setDims
    %Set the crop dimensions for each file
    %This MUST be run at least once per image before analysis

    markDims = true; %When defining the area of interest, do you
        %   actually want to mark the area on each image,
        %   or just use the default area?

    for k=selectedVials
        filepath = fpaths{k};
        %set dimensions (define area of interest) for each file
        %if markDims = false, set default dimensions for all the images
        andyfunc_setDimensions(filepath,selectedImages,markDims);
    end
end

if processFiles
    %Run the image segmentation algorithm on each image
    for k=selectedVials
        filepath = fpaths{k};
        fprintf('\n***Beginning processing of %s...\n',filepath);
    end
end

```

RunMe_SIR_Generic_Batch_v2.m (continued)

```
%Load dimensions (area of interest) for each image from file
dims = andyfunc_loadDimensions(filepath,selectedImages);

for i=selectedImages
    fprintf('+++Processing image %d of %d\n',i, ...
           length(selectedImages));
    %Perform the primary image analysis
    andyfunc_findSpheres_v10([filepath sprintf('%03d.png',i)], ...
                             dims(i,2:end),opts);
    if (length(selectedImages)>1)
        close all;
    end
end
fprintf('All images processed.\n');
end

if verifyFiles
    %Check to make sure everything is legit.
    %Sometimes bad spheres are incorrectly ID'd. This generally occurs
    % with the largest and smallest spheres in an images, so check them
    % manually and eliminate any funky ones.
    %The revised centroid locations list will be written to file.

    numPointsVerify = 8; %Number of biggest and smallest spheres to
                        % highlight at a time during verification
    verifyByImage = true; %true = view images to select valid spheres
                        %false = blindly accept all spheres

    for k=selectedVials
        filepath = fpaths{k};
        andyfunc_viewSingleSIR_v3(filepath,selectedImages, ...
                                   numPointsVerify,[],verifyByImage);
    end
end

if viewFigureHistogram
    %view/print size histogram for ALL vials
    fprintf('\n***Combining total stats for all vials...\n');
    %combine all stats into one master list
    [finalDiam, imageID] = andyfunc_combineTotalStatsSIR_v2( ...
        fpaths(selectedVials),selectedImages);
    %make a pretty figure using the combined data
    hHistFig = andyfunc_generateFigureHistogram_v1(finalDiam,opts, ...
        sprintf('%d vials',length(selectedVials)), 'combined_');
end
```

AndySizeOptions.m

```
classdef AndySizeOptions
%This class defines default options and parameters used in sphere analysis

%Written by: Andy Richards
%Last modified: Sept 2013
%Department of Mechanical & Aerospace Engineering
%North Carolina State University, Raleigh, NC

    properties
        writeStatsFile = false; %Write csv file with the diameter stats,
                                % sphere size, and centroid location
        showMainFigs = true;    %Show the primary figures during analysis
        printFigs = false;     %Print main figures to file (png)
        showHistogram = true;  %Plot diameter histogram (if false, just
                                % display the stats

        showAllFigs = false;   %Show EVERY figure (including minor
                                % ones) - this is a LOT of figures!
        showGoodCentroids = false; %Mark valid centroid locations with
                                % with asterisks (red)
        showGoodSpheres = true; %Mark valid spheres with circle
                                % outlines (red)
        showBadCentroids = true; %Mark invalid centroid locations with
                                % with asterisks (green/cyan)
        showBadSpheres = true; %Mark invalid spheres with circle
                                % outlines (green/cyan)
        showApproxCentroids = false; %Mark the initial centroid estimate
                                % for all regions (blue asterisk)
        showMaxMinCentroids = true; %Mark the centroids of largest (pink)
                                % and smallest (yellow) spheres

        % Set up the thresholds and constants
        units = 'px'; %Which units to use for figures, micron or pixels?
                    %'\mum' or 'px' are the only valid choices
                    %If '\mum' is selected, the variable 'pixel_size'
                    % must be defined
        pixel_size = 50.2/47.769759; %um/px - Size of each pixel in real
                                    % units (microns).
                                    %UPDATED 11/20/12 using full
                                    % precision results.

        metric_thresh_down = 0.80; %Minimum roundness criteria
        size_threshold_down = 40; %px, Minimum valid area for a sphere
                                %Use '40-50' for SIRSpheres, and '20'
                                % for NIST standards
        size_threshold_up = 3500; %px, Maximum valid area for a sphere
        borderSize = 5; %Ignore any spheres whose edges are
                        % closer than this many pixels
    end
end
```

AndySizeOptions.m (continued)

```

window_side_length = 80;      % to the border
                               %Side length of the square window used
                               % for circle detection (should be
                               % greater than the largest sphere)
num_pixels_quadrant = 5;     %Number of pixels per quadrant to use
                               % during least squares refinement
                               % of centroid/radius
bw_thresh_manual = 0.60;    %Threshold override for converting
                               % grayscale to B&W

radius_percent = 0.925;     %For any two spheres, if the ratio of
                               % centroid-to-centroid distance over
                               % the sum of radii is less than this
                               % value, the spheres are in
                               % violation
num_radius_violations = 2;  %Max number of overlapped radii
                               % violoations before a sphere is
                               % deemed invalid and removed

initialMag = 25;            %Magnification for displaying each
                               % image (depends on your screen)
histogramBins = 22:0.25:36; %Where should hist bins be located?
                               %For NIST spheres, use 44:0.5:51 px

end

methods
end
end
```

andyfunc_setDimensions.m

```
function andyfunc_setDimensions(filepath,images,viewImages)
%This function either loads and displays each image and lets user select
% a rectangular area of interest to save to file, or just save the
% default rectangle dimensions without viewing the image

%Written by: Andy Richards
%Last modified: Sept 2013
%Department of Mechanical & Aerospace Engineering
%North Carolina State University, Raleigh, NC

%load previous dimensions if they exist, or load defaults
dims = andyfunc_loadDimensions(filepath,images);
hFig = 1;

if viewImages
    for i=images

        rectX = dims(i,2); %X
        rectY = dims(i,3); %Y
        rectWidth = dims(i,4); %width
        rectHeight = dims(i,5); %height

        %Open the image file
        img = imread([filepath sprintf('%03d.png',i)]);

        %Display the OG image
        figure(hFig);
        clf(hFig); %clear figure for next image
        imshow(img, 'InitialMagnification',25);

        cont = 'm';
        while (cont=='m')

            title(sprintf('%03d - [%d, %d, %d, %d]',i,rectX, rectY, ...
                rectWidth, rectHeight));
            hold on;
            %Draw a rectangle at the old dimensions (or the default)
            hRect = rectangle('Position',[rectX, rectY, ...
                rectWidth, rectHeight], 'EdgeColor','b');
            hold off;

            %Wait for user input
            cont = input(['Save and continue [Enter]? ', ...
                '[m]ark rectangle? [q]uit and save?>'], 's');

            if cont=='m'
                fprintf('Mark rectangle\n');
```

andyfunc_setDimensions.m (continued)

```
rect = getrect(hFig);
if(~isempty(rect))
    rectX = round(rect(1));
    rectY = round(rect(2));
    rectWidth = round(rect(3));
    rectHeight = round(rect(4));
end
fprintf('Dimensions updated to [%d, %d, %d, %d]\n', ...
    rectX, rectY, rectWidth, rectHeight);
end

delete(hRect);
end

if isempty(cont) %save and move to next image
    dims(i,:) = [i, rectX, rectY, rectWidth, rectHeight];
elseif cont == 'q'
    break;
end

end
close(hFig);
end

fprintf('Saving updated stats to file\n');
fid = fopen([filepath 'dimensions.txt'],'w+');
fprintf(fid, 'Image, RectX, RectY, RectWidth, RectHeight\r\n');
fprintf(fid, '%02d, %d, %d, %d, %d\r\n', dims');
fclose(fid);
end
```

andyfunc_loadDimensions.m

```
function [ dims ] = andyfunc_loadDimensions( filepath, images )
%This function loads previously saved area dimensions from file

%Written by: Andy Richards
%Last modified: Sept 2013
%Department of Mechanical & Aerospace Engineering
%North Carolina State University, Raleigh, NC

defDims = [200,200,2100,1500]; %Default dimensions
                                %These generally works for all images
dims = [(1:max(images))', defDims(1)*ones(max(images),1), ...
        defDims(2)*ones(max(images),1), defDims(3)*ones(max(images),1), ...
        defDims(4)*ones(max(images),1)];

fprintf('Loading dimensions file for %s images %d to %d\n', ...
        filepath, images(1),images(end));
fid = fopen([filepath 'dimensions.txt']);
if(fid<0) %does not exist
    fprintf('File not found, using default dimensions\n');
else
    fgetl(fid); %ignore first line (it's the header)
    %Read the image number, bottom left pixel, bot right, width, height
    data = fscanf(fid, '%d, %d, %d, %d, %d', [5,inf]);
    fclose(fid);
    data = data';

    %Check if the file contained more dimensions that you are asking for
    %If so, include them!
    if(data(end,1)>max(images))
        dims = [(1:data(end,1))', defDims(1)*ones(data(end,1),1), ...
                defDims(2)*ones(data(end,1),1), ...
                defDims(3)*ones(data(end,1),1), ...
                defDims(4)*ones(data(end,1),1)];
    end
    dims(data(:,1),:) = data(:,:);
    %fprintf('Data loaded.\n');
end
end
```

andyfunc_findSpheres_v10.m

```
function andyfunc_findSpheres_v10(image_filename,dimensions,opts)
%This function locates the centroid of all isolated circular features
%  within an image, measures the diameter, and rejects overlapping ones

%Written by: Andy Richards
%Last modified: Sept 2013
%Department of Mechanical & Aerospace Engineering
%North Carolina State University, Raleigh, NC

%Originally based on James' "histogram.m" file (but not really any more)
%Uses least squares minimization to determine the centroid
%Uses Andy's updated filtering and image segmentation
%  and Andy's optimized (least squares minimization) method for
%  refined centroid determination and more advanced radius determination
%Options and variables are defined in a separate class (AndySizeOptions.m)

if nargin>0
    [filePath,fileName,fileExten] = fileparts(image_filename);
    img_col_offset = dimensions(1); %X
    img_row_offset = dimensions(2); %Y
    img_col_useful = dimensions(3); %width
    img_row_useful = dimensions(4); %height
else
    %some defaults for testing this function with a single image
    close all;
    clear all;
    filePath = '..\232.1';
    fileName = '232.1_020814_025';
    fileExten = '.png';

    opts = AndySizeOptions;
    img_col_offset = 200; % column offset for crop      %200; 700; 1700
    img_row_offset = 200; % row offset for crop        %200; 600; 1200
    img_col_useful = 2100; % Desired width of the cropped image region
    img_row_useful = 1500; % Desired height of the cropped image region

    opts.writeStatsFile = false; %write csv file
    opts.showMainFigs = true; %show the final figure
    opts.printFigs = false; %print final figure and hist to file
    opts.showHistogram = true; %plot diameter histogram
    opts.showAllFigs = false; %show every figure (including minor ones)
end

fprintf('=====\n');
fprintf('Loading image file: %s\n',[filePath, '\', fileName, fileExten]);
fprintf('=====\n');

globalStartTime = tic;
```

andyfunc_findSpheres_v10.m (continued)

```
%Read the original image
img = imread([filePath, '\', fileName, fileExten]);

img_col_original = size(img,2); %Column number (width) of original image
img_row_original = size(img,1); %Row number (height) of original image

fprintf('Converting img to grayscale and smoothing noise (%d x %d).\n',...
        img_col_original,img_row_original);
%Convert the original image into grayscale image
grayscale = rgb2gray(img);
%Use filter to smooth image/noise
%Do this before cropping to prevent edge artifacts)
h = fspecial('gaussian', [3 3], 0.5); %gaussian filter to smooth noise
grayscale = imfilter(grayscale, h);

if opts.showAllFigs || opts.showMainFigs
    hOGFig = figure; imshow(grayscale,'InitialMagnification', ...
                            opts.initialMag);
    title(sprintf(['Original grayscale image, cropped region shown ', ...
                  '(%d,%d) (%d x %d)',img_col_offset,img_row_offset, ...
                  img_col_useful,img_row_useful]));
    rectangle('Position',[img_col_offset,img_row_offset, ...
                          img_col_useful,img_row_useful], 'EdgeColor', 'b', 'LineStyle', '-');
end

%crop image to workable region
gray_cropped = grayscale(img_row_offset:...
                        (img_row_offset+img_row_useful-1), ...
                        img_col_offset:(img_col_offset+img_col_useful-1),:);

fprintf('Adjusting for background variance and optimizing contrast.\n');
%Read the background variance image file
background = imread([filePath, '\blank.png']);
%Use filter to smooth image noise
%Do this before cropping to prevent edge artifacts
h = fspecial('gaussian', [5 5], 0.75); %gaussian filter to smooth noise
background = imfilter(background, h);
%crop to new size
background = background(img_row_offset:...
                        (img_row_offset+img_row_useful-1), ...
                        img_col_offset:(img_col_offset+img_col_useful-1),:);
background = rgb2gray(background);
background = imcomplement(background);

if opts.showAllFigs
    figure; imshow(background,'InitialMagnification',opts.initialMag);
    title('Background variance');
end
```

andyfunc_findSpheres_v10.m (continued)

```
%Subtract the background image from the grayscale image
%To avoid integer overflow here (cut off at 255), you'd better
%    convert to double first, then normalize back to uint8
uniform_gray = double(gray_cropped) + double(background);
%increase contrast to full range
uniform_gray = uniform_gray - min(min(uniform_gray));
%normalize (0-1) and convert back to uint8
uniform_gray = uniform_gray./max(max(uniform_gray));
uniform_gray = uint8(255*uniform_gray); %convert to 0-255 grayscale
if opts.showAllFigs
    figure; imshow(uniform_gray,'InitialMagnification',opts.initialMag);
    title('Background adjusted grayscale image - Andy');
end

%Increase the image contrast
%Sets top and bottom 0.1% to range limits (0, 255)
optim_gray= imadjust(uniform_gray,stretchlim(uniform_gray,[0.001 0.999]));
if opts.showAllFigs
    figure; imshow(optim_gray,'InitialMagnification',opts.initialMag);
    title('Contrast optimized grayscale image');
end
%optim_gray = uniform_gray;

fprintf('Applying gray level threshold (level = %0.2f).\n', ...
    opts.bw_thresh_manual);
%Threshold the image
level = opts.bw_thresh_manual; %manual threshold override
preblkwht = im2bw(optim_gray,level);

if opts.showAllFigs
    figure; imshow(preblkwht,'InitialMagnification',opts.initialMag);
    title(sprintf('Thresholded image: level = %0.2f',level));
end

fprintf('Smoothing BW image.\n');
%Morphologically filter the image to smooth jagged edges and noise pixels
nTimes = 1;
operation = 'majority'; %Sets pixel value based on majority of neighbors
blkwht = bwmorph(preblkwht,operation,nTimes);

if opts.showAllFigs
    figure; imshow(blkwht,'InitialMagnification',opts.initialMag);
    title(sprintf('Thresholded, smoothed image (oper=%s, n=%d)', ...
        operation,nTimes));
end
```

andyfunc_findSpheres_v10.m (continued)

```
%=====
% Eliminate large and non-circular regions
%=====
fprintf('Identifying connected regions...');
t1 = tic;
%find and label connected regions
cc = bwconncomp(blkwht); %this is faster than bwlabel, or bwboundaries
stats = regionprops(cc, 'Area', 'Centroid', 'Perimeter');
fprintf('%0.2f sec\n', toc(t1));
fprintf(['Removing large and non-circular regions ', ...
        '(circle thresh = %0.2f).\n'], opts.metric_thresh_down);
%First remove areas that are too small and too large. The remaining
% regions are ok size-wise, but need to be checked for roundness
idx_ok = find(([stats.Area]<opts.size_threshold_up) & ...
              ([stats.Area]>opts.size_threshold_down));

if opts.showAllFigs
    figure; imshow(ismember(labelmatrix(cc), idx_ok), ...
                  'InitialMagnification', opts.initialMag);
    title(sprintf('Extreme areas filled, size thresh = (%d,%d)', ...
                 opts.size_threshold_down, opts.size_threshold_up));
end

%Check remaining areas for roundness
% (compare area/perimeter to that of ideal circle)
testAreas = 4*pi*[stats(idx_ok).Area];
testPerims = [stats(idx_ok).Perimeter].^2;
metric_Arr = testAreas./testPerims;
%compare with our roundness threshold
idx_good = idx_ok(metric_Arr>opts.metric_thresh_down);

%keep the pixels labelled with the good indices (round regions)
blkwht_centroids = ismember(labelmatrix(cc), idx_good);
centroids = cat(1, stats(idx_good).Centroid); %save the centroid locations

if opts.showAllFigs
    figure; imshow(blkwht_centroids, ...
                  'InitialMagnification', opts.initialMag);
    title(sprintf('Non-circular areas filled, thresh = %0.2f', ...
                 opts.metric_thresh_down));
end

%Remove centroid regions too close to the edge
% (both left and right side, and top and bottom sides)
fprintf('Removing regions within %d pixels from the border.\n', 15);
goodBorderIndices = (centroids(:,1)>15) & ...
                   (centroids(:,1)<(img_col_useful-15)) & ...
                   (centroids(:,2)>15) & (centroids(:,2)<(img_row_useful-15));
```

andyfunc_findSpheres_v10.m (continued)

```
centroids2 = centroids(goodBorderIndices,:); %update the centroid list

fprintf(['--Initial centroid estimations complete ', ...
        '(%d centroids found).\n'],size(centroids2,1));
%Plot estimated centroids
if opts.showAllFigs
    figure; imshow(optim_gray,'InitialMagnification',opts.initialMag);
    title('Estimated centroids marked');
    hold on;
    plot(centroids2(:,1), centroids2(:,2), 'b*');
    hold off;
end

%=====
% Refine centroid estimate, then determine radius of spheres
%=====

fprintf('Refining centroid estimations and calculating sphere radii.\n');

%fill centroid regions, and complement the image
blkwht_for_radius = ~(blkwht & ~blkwht_centroids);
if opts.showAllFigs
    figure; imshow(blkwht_for_radius, ...
        'InitialMagnification',opts.initialMag);
    title(sprintf('Threshold, centroids filled'));
end

nTimes = 1;
%morphologically reduce the image to just the outline of each region
operation = 'remove';
bw_outline = bwmorph(blkwht_for_radius,operation,nTimes);

%set borders to black, this avoids long iterations on border regions
bw_outline(:,1) = 0;
bw_outline(:,end) = 0;
bw_outline(1,:) = 0;
bw_outline(end,:) = 0;

if opts.showAllFigs
    figure; imshow(bw_outline,'InitialMagnification',opts.initialMag);
    title(sprintf('Outlines (oper=%s, n=%d)',operation,nTimes));
    %if opts.showApproxCentroids
        hold on;
        %plot old centroid estimates
        plot(centroids2(:,1),centroids2(:,2), 'b+');
        hold off;
    %end
end
```

andyfunc_findSpheres_v10.m (continued)

```
%Prepare to step through each sphere and determine the actual centroid
% and the average radius
if (opts.showAllFigs || opts.showMainFigs)
    hFig = figure; imshow(optim_gray, ...
        'InitialMagnification',opts.initialMag);
    title('grayscale with spheres marked');
    hold on;
    if opts.showApproxCentroids
        %plot old centroid estimates
        plot(centroids2(:,1),centroids2(:,2), 'b+');
    end
end

radius = zeros(size(centroids2,1),1);
%prepare to store a new, more accurate, estimate for centroids
centroids3 = zeros(size(centroids2));
optim_iter_count = zeros(size(centroids2,1),1);
options = optimset('Display','off','Algorithm','levenberg-marquardt');

t1 = tic;
for sphere_num = 1:size(centroids2,1)

    %Grab the coordinates of each centroid from the previous step
    % (this is the approx centroid)
    centroid_x = centroids2(sphere_num,1);
    centroid_y = centroids2(sphere_num,2);

    %Determine the coordinates of a square window for testing
    window_left = round(centroid_x - 0.5*opts.window_side_length);
    window_right = window_left + opts.window_side_length;
    window_bottom = round(centroid_y-0.5*opts.window_side_length);
    window_top = window_bottom + opts.window_side_length;

    %Ensure the square doesnt exceed the boundary
    if(window_left<1)
        window_left=1;
    end
    if(window_right>img_col_useful)
        window_right=img_col_useful;
    end
    if(window_bottom<1)
        window_bottom=1;
    end
    if(window_top>img_row_useful)
        window_top=img_row_useful;
    end
end
```

andyfunc_findSpheres_v10.m (continued)

```
%Crop the image to just the square window
square = bw_outline(window_bottom:window_top, ...
                    window_left:window_right);

%Determine the location of the approx sphere centroid relative
% to a square window
centroid_x_in_square = centroid_x - (window_left) + 1; %column
centroid_y_in_square = centroid_y - (window_bottom) + 1; %row

%=====
%Determine the radius of this sphere
%=====

%Get the coordinates of the edge pixels in square
[row, col] = find(square == 1); %find all the white pixels
edgePixels_in_square = [col, row]; %coordinates of all white pixels,
% swapped for correct x and y

%Separate the pixels into 4 quadrants based on their location to the
% estimated centroid, then choose the 5 closest ones
% from each quadrant
topLeftPx = edgePixels_in_square( ...
    (edgePixels_in_square(:,1) <= centroid_x_in_square) & ...
    (edgePixels_in_square(:,2) <= centroid_y_in_square),:);
topRightPx = edgePixels_in_square( ...
    (edgePixels_in_square(:,1) > centroid_x_in_square) & ...
    (edgePixels_in_square(:,2) <= centroid_y_in_square),:);
botLeftPx = edgePixels_in_square( ...
    (edgePixels_in_square(:,1) <= centroid_x_in_square) & ...
    (edgePixels_in_square(:,2) > centroid_y_in_square),:);
botRightPx = edgePixels_in_square( ...
    (edgePixels_in_square(:,1) > centroid_x_in_square) & ...
    (edgePixels_in_square(:,2) > centroid_y_in_square),:);

minQuadSqDistArr = [Inf,Inf,Inf,Inf]; %preallocate space
%Determine the distances of edge pixels to the centroid of our sphere
%To save on computational time, just find the squared distance, dont
% do square root yet
if ~isempty(topLeftPx) %ensure there are at least some pixels
    distance_sq_TL = (topLeftPx(:,1)-centroid_x_in_square).^2 + ...
        (topLeftPx(:,2)-centroid_y_in_square).^2;
    %Grab the index of the closest points
    [distance_sq_sort_TL, ogIndexTL] = sort(distance_sq_TL);
    %Check if this quadrant has enough points
    if (length(distance_sq_sort_TL)<opts.num_pixels_quadrant)
        num_pixels_TL = length(distance_sq_sort_TL);
    else
        num_pixels_TL = opts.num_pixels_quadrant;
    end
end
```

andyfunc_findSpheres_v10.m (continued)

```

    %save the minimum distance
    minQuadSqDistArr(1) = distance_sq_sort_TL(1);
else
    distance_sq_sort_TL = [];
    num_pixels_TL = 0;
end

if ~isempty(topRightPx)
    distance_sq_TR = (topRightPx(:,1)-centroid_x_in_square).^2 + ...
        (topRightPx(:,2)-centroid_y_in_square).^2;
    %grab the index of the closest points
    [distance_sq_sort_TR, ogIndexTR] = sort(distance_sq_TR);
    %check if this quadrant has enough points
    if (length(distance_sq_sort_TR)<opts.num_pixels_quadrant)
        num_pixels_TR = length(distance_sq_sort_TR);
    else
        num_pixels_TR = opts.num_pixels_quadrant;
    end
    %save the minimum distance
    minQuadSqDistArr(2) = distance_sq_sort_TR(1);
else
    distance_sq_sort_TR = [];
    num_pixels_TR = 0;
end

if ~isempty(botLeftPx)
    distance_sq_BL = (botLeftPx(:,1)-centroid_x_in_square).^2 + ...
        (botLeftPx(:,2)-centroid_y_in_square).^2;
    %grab the index of the closest points
    [distance_sq_sort_BL, ogIndexBL] = sort(distance_sq_BL);
    %check if this quadrant has enough points
    if (length(distance_sq_sort_BL)<opts.num_pixels_quadrant)
        num_pixels_BL = length(distance_sq_sort_BL);
    else
        num_pixels_BL = opts.num_pixels_quadrant;
    end
    %save the minimum distance
    minQuadSqDistArr(3) = distance_sq_sort_BL(1);
else
    distance_sq_sort_BL = [];
    num_pixels_BL = 0;
end

if ~isempty(botRightPx)
    distance_sq_BR = (botRightPx(:,1)-centroid_x_in_square).^2 + ...
        (botRightPx(:,2)-centroid_y_in_square).^2;
    %grab the index of the closest points
    [distance_sq_sort_BR, ogIndexBR] = sort(distance_sq_BR);

```

andyfunc_findSpheres_v10.m (continued)

```
%check if this quadrant has enough points
if (length(distance_sq_sort_BR)<opts.num_pixels_quadrant)
    num_pixels_BR = length(distance_sq_sort_BR);
else
    num_pixels_BR = opts.num_pixels_quadrant;
end
%save the minimum distance
minQuadSqDistArr(4) = distance_sq_sort_BR(1);
else
    distance_sq_sort_BR = [];
    num_pixels_BR = 0;
end

%Be careful of cases where there are no reasonably close pixels
% in a quadrant
%Find the min distance from each quadrant, and disregard quadrants
% whose max pixels are excessively further than the min from
% all quadrants
minQuadSqDist = min(minQuadSqDistArr);

sub_edgePixels = [];
quadrantCount = 0;
%Ff the max pixel from this quadrant is more than 2X further ...
% (squared) from the min pixel, dont used this quadrant
if(num_pixels_TL>0 && ...
    (distance_sq_sort_TL(num_pixels_TL)/minQuadSqDist < 4))
    sub_edgePixels = [sub_edgePixels; ...
        topLeftPx(ogIndexTL(1:num_pixels_TL),:)];
    quadrantCount = quadrantCount+1;
end
if(num_pixels_TR>0 && ...
    (distance_sq_sort_TR(num_pixels_TR)/minQuadSqDist < 4))
    sub_edgePixels = [sub_edgePixels; ...
        topRightPx(ogIndexTR(1:num_pixels_TR),:)];
    quadrantCount = quadrantCount+1;
end
if(num_pixels_BL>0 && ...
    (distance_sq_sort_BL(num_pixels_BL)/minQuadSqDist < 4))
    sub_edgePixels = [sub_edgePixels; ...
        botLeftPx(ogIndexBL(1:num_pixels_BL),:)];
    quadrantCount = quadrantCount+1;
end
if(num_pixels_BR>0 && ...
    (distance_sq_sort_BR(num_pixels_BR)/minQuadSqDist < 4))
    sub_edgePixels = [sub_edgePixels; ...
        botRightPx(ogIndexBR(1:num_pixels_BR),:)];
    quadrantCount = quadrantCount+1;
end
```

andyfunc_findSpheres_v10.m (continued)

```
%Continue to search for the optimal centroid as long as one
% quadrant is represented
if(quadrantCount>0)
    %Define starting point for least squares minimization
    x0 = [centroid_x_in_square, centroid_y_in_square];
    %Define which function (pass the extra parameter)
    f = @(x)andyfunc_evalCentroid(x,sub_edgePixels);
    %perform least squares minimization to determine centroid
    [minX,~,~,~,output]=lsqnonlin(f,x0,[],[],options);
    %store how many iterations this took
    optim_iter_count(sphere_num) = output.iterations;

    %Since you've found a more accurate centroid, calculate the
    % average radius...again
    sub_dist = sqrt((sub_edgePixels(:,1)-minX(1)).^2 + ...
                    (sub_edgePixels(:,2)-minX(2)).^2);
    sub_avgDist= mean(sub_dist);

    %Check to make sure that this is a reasonable radius, sometimes,
    % if only 1 quadrant is represented, you can get some bad radii
    if(sub_avgDist > (2*sqrt(minQuadSqDist)))
        %Is your new guess more than 2x bigger than the old minimum???
        %It's probably a bad one, so mark it!
        %Use the old centroid and set the distance to an invalid
        % number - we'll check for it later
        sub_avgDist=0;
        minX = x0;
    end
else
    %There were no pixels within the window!! WTF!
    %Use the old centroid and set the distance to an invalid number
    sub_avgDist = 0;
    minX = x0;
end

%=====
%=====

%Add the new values to an updated list, convert back to the
% global position
newX = minX(1)+window_left-1;
newY = minX(2)+window_bottom-1;

centroids3(sphere_num,1) = newX;
centroids3(sphere_num,2) = newY;
radius(sphere_num)=sub_avgDist;

end
```

andyfunc_findSpheres_v10.m (continued)

```
fprintf(['--Radius determination completed in %0.2f sec. ', ...
        'Max iteration count = %d\n'],toc(t1),max(optim_iter_count));

%Check for overlapping radii
fprintf('Identifying and removing invalid regions...');
%For each questionable sphere, check the distance from its centroid to
%  all other centroids. If the distance to the closest neighbor is less
%  than the sum of the radii, remove it (after N violations).
badSphereIndex = [];
edgeSphereIndex = [];
t1 = tic;
for i=1:length(radius) %check ALL spheres!
    currIndex = i; %what's the index of the current large sphere you're on

    currX = centroids3(currIndex,1);
    currY = centroids3(currIndex,2);
    currRadius = radius(currIndex);

    %Skip this sphere if it's too close to the border (check again in
    %  case of bogus radius near the edge)
    if ((currX-currRadius)<opts.borderSize || ...
        (currX+currRadius)>(img_col_useful-opts.borderSize) || ...
        (currY-currRadius)<opts.borderSize || ...
        (currY+currRadius)>(img_row_useful-opts.borderSize))
        %sphere is too close to the border
        edgeSphereIndex = [edgeSphereIndex; currIndex];
    else
        %Check for spheres in the vicinity of the current sphere
        %Use the same window as previously
        windowSize = 0.5*opts.window_side_length; %2*radius(currIndex);
        %Logical array
        tempLogicalIndices = (centroids3(:,1)>(currX-windowSize)) & ...
            (centroids3(:,1)<(currX+windowSize)) & ...
            (centroids3(:,2)>(currY-windowSize)) & ...
            (centroids3(:,2)<(currY+windowSize));
        tempLogicalIndices(currIndex)=0; %remove the self reference
        windowIndices = find(tempLogicalIndices);
        %Check the distance from any spheres close to this large one
        %  and see if they are less than the sum of the radii
        sqDist = (centroids3(windowIndices,1) -
            centroids3(currIndex,1)).^2 + ...
            (centroids3(windowIndices,2) - centroids3(currIndex,2)).^2;

        %Check if the distance is less than some fraction of the radii
        %  sum (or if it's zero)
        %Generally, if two centroids are closer than ~95% of the sum of
        %  radii, they are invalid
        %Check for at least ~3 violation, otherwise you may get good ones
```

andyfunc_findSpheres_v10.m (continued)

```

if(currRadius==0 || (sum(sqDist<(opts.radius_percent * ...
    (currRadius+radius(windowIndices)).^2) > ...
    opts.num_radius_violations))
    %This is a bad location!!!!
    badSphereIndex = [badSphereIndex; currIndex];
    if ((opts.showAllFigs || opts.showMainFigs))
        if(currRadius > 0)
            if opts.showBadCentroids
                plot(currX,currY,'g*'); %mark this one!
            end
            if opts.showBadSpheres
                rectangle('Position',[currX-currRadius, ...
                    currY-currRadius,2*currRadius, ...
                    2*currRadius], ...
                    'Curvature',[1,1],'EdgeColor','g');
            end
        else
            if opts.showBadSpheres || opts.showBadCentroids
                %These have no outline - always plot the centroid
                %Plot the zero radii ones in a different color,
                % and don't draw the outline
                plot(currX,currY,'c*'); %mark this one!
            end
        end
    end
else
    %this is a good sphere! show it off!
    if (opts.showAllFigs || opts.showMainFigs)
        if opts.showGoodCentroids
            plot(currX,currY,'r+');
        end
        if opts.showGoodSpheres
            rectangle('Position',[currX-currRadius, ...
                currY-currRadius,2*currRadius,2*currRadius], ...
                'Curvature',[1,1],'EdgeColor','r');
        end
    end
end
end
end
end

finalCentroids = centroids3;
%remove bad/edge spheres
finalCentroids([edgeSphereIndex; badSphereIndex],:)=[];
finalDiam = radius;
finalDiam([edgeSphereIndex; badSphereIndex]) = [];
finalDiam = 2*finalDiam; %convert to diameter - keep in pixels!

fprintf('%0.2f sec\n',toc(t1));

```

andyfunc_findSpheres_v10.m (continued)

```
if (opts.showAllFigs || opts.showMainFigs)
    if opts.showMaxMinCentroids
        %hold on
        %Find the index of the smallest & largest radius
        maxIndex = find(finalDiam==max(finalDiam));
        minIndex = find(finalDiam==min(finalDiam));
        %Find approx centroid of that region
        plot(finalCentroids(maxIndex,1),finalCentroids(maxIndex,2),'m*');
        plot(finalCentroids(minIndex,1),finalCentroids(minIndex,2),'y*');
    end
    %add a nice figure legend
    andyfunc_addFigureLegend(hFig);

    hold off;
end

fprintf('--Finished. Total time = %0.2f sec\n',toc(globalStartTime));

numSpheres = length(finalDiam);
avgDiam = mean(finalDiam); %mean
stddevDiam = std(finalDiam); %standard deviation
minDiam = min(finalDiam);
maxDiam = max(finalDiam);
medianDiam = median(finalDiam);

fprintf('Spheres = %d (%d)\n',numSpheres,length(badSphereIndex));
fprintf('Mean = %0.1f %s\n',avgDiam,'px');
fprintf('Std dev = %0.2f\n',stddevDiam);
fprintf('Median = %0.1f\n',medianDiam);
fprintf('Max = %0.1f\n',maxDiam);
fprintf('Min = %0.1f\n',minDiam);

if opts.printFigs
    fprintf('Printing figures to file.\n');

    if (opts.showAllFigs || opts.showMainFigs)
        set(0,'CurrentFigure',hOGFig);
        set(hFig,'PaperPositionMode','auto');
        %print Main figure with spheres
        print('-dpng','-r300',[filePath,'\ ',fileName,'_crop.png']);

        set(0,'CurrentFigure',hFig);
        set(hFig,'PaperPositionMode','auto');
        print('-dpng','-r300',[filePath,'\ ',fileName,'_marked.png']);

    end
end
```

andyfunc_findSpheres_v10.m (continued)

```
if opts.writeStatsFile
    fprintf('Writing sphere diameter stats to file: %s.\n', ...
           [filePath, '\', fileName, '_stats_px.txt']);
    f = fopen([filePath, '\', fileName, '_stats_px.txt'],'w+');
    fprintf(f,['Num Spheres, Num Bad Spheres, Mean Diam, Std Dev, ', ...
              'Median Diam, Max Diam, Min Diam, Units, ', ...
              'TopX, TopY, Width, Height\r\n']);
    fprintf(f,['%d, %d, %0.4f, %0.4f, %0.3f, %0.3f, %0.3f, px, ', ...
              '%d, %d, %d, %d\r\n'], numSpheres,length(badSphereIndex), ...
            avgDiam,stddevDiam,medianDiam,maxDiam,minDiam, ...
            img_col_offset,img_row_offset,img_col_useful,img_row_useful);
    fprintf(f,'Diam, CentrX, CentrY\r\n');
    %sort by the diameter before saving
    totalData = sortrows([finalDiam, finalCentroids]);
    fprintf(f,'%f,%f,%f\r\n',totalData');
    fclose(f);

    %now that we've written the stats file, delete the old revised version
    if(exist([filePath, '\', fileName, '_stats_px_rev.txt'],'file')>0)
        fprintf('Deleting old revised stats file.\n');
        delete([filePath, '\', fileName, '_stats_px_rev.txt']);
    end
end
fprintf('Done\n\n');
end
```

andyfunc_addFigureLegend.m

```
function [ ] = andyfunc_addFigureLegend( hFig )
%This function draws a nice figure legend with all the correct symbols
% onto an existing figure

%Written by: Andy Richards
%Last modified: Sept 2013
%Department of Mechanical & Aerospace Engineering
%North Carolina State University, Raleigh, NC

set(0, 'CurrentFigure', hFig);
%widen the figure to accomodate the legend
hPos = get(hFig, 'Position');
ax1 = gca;
set(ax1, 'Units', 'pixels'); %dont let it autoadjust when we resize
ax1OutPos = get(ax1, 'OuterPosition');
%increase width by 100
set(hFig, 'Position', [hPos(1) hPos(2) hPos(3)+100 hPos(4)]);
set(ax1, 'Units', 'normalized'); %reenable autoresizing

ax2 = axes(); %add new axis for legend
set(ax2, 'Units', 'pixels');
%set the starting position the same
set(ax2, 'Position', [hPos(3)-50 0 100 hPos(4)]);
set(ax2, 'Units', 'normalized');
set(ax2, 'ActivePositionProperty', 'position');
set(ax2, 'PlotBoxAspectRatio', [0.4 1 1]);
xlim([0 1]);
ylim([0 1]);
axis off;
hold on;

initVal = 0.75; %top left y-value
yval = initVal;
space = 0.075; %how much space btw entries?

plot(0.1, yval, 'ro', 'MarkerSize', 10);
plot(0.1, yval, 'r+');
text(0.2, yval, 'Good');

yval = yval-space;
plot(0.1, yval, 'm*');
text(0.2, yval, 'Max');

yval = yval-space;
plot(0.1, yval, 'y*');
text(0.2, yval, 'Min');

yval = yval-space;
```

andyfunc_addFigureLegend.m (continued)

```
plot(0.1, yval, 'b+');
text(0.2, yval, 'Est centroid');

yval = yval-space;
plot(0.1, yval, 'go', 'MarkerSize', 10);
plot(0.1, yval, 'g*');
text(0.2, yval, 'Invalid');

yval = yval-space;
plot(0.1, yval, 'c*');
text(0.2, yval, 'Bad');

rectangle('Position', [0, yval-space, 1, initVal-yval+2*space], ...
          'FaceColor', [0.7 0.7 0.7]);
%flip the display order so the rect is at the bottom
set(gca, 'children', flipud(get(gca, 'children')));
hold off
end
```


andyfunc_viewSingleSIR_v3.m (continued)

```
if viewImages

    img = imread([filepath sprintf('%03d.png',i)]); %open image file
    imgc = img(img_row_offset:(img_row_offset+img_row_useful-1), ...
        img_col_offset:(img_col_offset+img_col_useful-1),:);

    figure(hFig);
    clf(hFig); %clear figure for next image
    imshow(imgc, 'InitialMagnification',38);

    cont = 'm';

    while (cont=='m')
        if(size(sorted,1)<(numPoints*2)) %are there enough to display?
            numPoints = floor(size(sorted,1)/2);
        end

        %get the smallest and largest ones
        currX = sorted([1:numPoints, (end-numPoints+1):end],2);
        currY = sorted([1:numPoints, (end-numPoints+1):end],3);
        currDiam = sorted([1:numPoints, (end-numPoints+1):end],1);

        title(sprintf('%03d - min: %0.1f, max: %0.1f',i, ...
            currDiam(1),currDiam(end)));

        %plot the smallest, see if they are valid
        hold on;
        %mark the smallest as different color
        hPlots(1) = plot(currX(1),currY(1),'y*');
        %mark the rest
        hPlots(2) = plot(currX(2:numPoints),currY(2:numPoints),'b+');
        for j=1:numPoints
            hRect(j) = rectangle('Position', ...
                [currX(j)-currDiam(j)/2,currY(j)-currDiam(j)/2, ...
                currDiam(j),currDiam(j)], ...
                'Curvature',[1,1], 'EdgeColor','r');
            %put the number next to it
            hText(j) =text(currX(j)-currDiam(j)/2,currY(j), ...
                sprintf('%d',j), 'HorizontalAlignment','Right', ...
                'Color','black','FontWeight','bold');
        end
        hold off

        %plot the largest 5, see if they are valid
        hold on;
        %mark the largest with different color
        hPlots(3) = plot(currX(end),currY(end),'m*');
```

andyfunc_viewSingleSIR_v3.m (continued)

```

%mark the rest
hPlots(4) = plot(currX((numPoints+1):(end-1)), ...
    currY((numPoints+1):(end-1)), 'b+');
for j=(numPoints+1):(2*numPoints)
    hRect(j) = rectangle('Position', ...
        [currX(j)-currDiam(j)/2,currY(j)-currDiam(j)/2, ...
        currDiam(j),currDiam(j)], ...
        'Curvature',[1,1], 'EdgeColor','g');
    hText(j) = text(currX(j)-currDiam(j)/2,currY(j), ...
        sprintf('%d',j), 'HorizontalAlignment','Right', ...
        'Color','black', 'FontWeight','bold');
end
hold off

%wait for user input.
cont = input(['Save and continue [Enter]? [m]ark points? ',...
    '[n]ext, no save? [q]uit no save?>'],'s');
if cont=='m'
    fprintf('Select spheres to remove\n');
    [ptsX,ptsY] = getpts(hFig);
    %ptsIndex = zeros(length(ptsX),1);
    ptsIndex = [];
    for j=1:length(ptsX)
        %find which points these are
        tempDiam = currDiam(currX>(ptsX(j)-buffer) & ...
            currX<(ptsX(j)+buffer) & ...
            currY>(ptsY(j)-buffer) & ...
            currY<(ptsY(j)+buffer));
        %what if there are no matches??
        %what if there are more than one??
        if ~isempty(tempDiam)
            %grab the indices
            tempIndex = find(sorted(:,1)==tempDiam(1));
            %Only add the first one
            %This will take care of multiples
            ptsIndex = [ptsIndex; tempIndex(1)];
        end
    end
    sorted(ptsIndex,:)=[]; %update the list
    %add these to the count of bad spheres
    tempBad = tempBad+length(ptsIndex);
    fprintf('%d spheres removed from image %d\n', ...
        length(ptsIndex),i);
end
delete(hPlots);
delete(hRect);
delete(hText);
end
end

```

andyfunc_viewSingleSIR_v3.m (continued)

```
if isempty(cont) %save and move to next image
    fprintf('Saving updated stats to file\n');
    fid2 = fopen([filepath sprintf('%03d_stats_px_rev.txt',i)], 'w+');
    fprintf(fid2, '%s\r\n', headerLine1);

    numSpheres = size(sorted,1);
    meanDiam = mean(sorted(:,1)); %mean
    stddevDiam = std(sorted(:,1)); %standard deviation
    minDiam = min(sorted(:,1));
    maxDiam = max(sorted(:,1));
    medianDiam = median(sorted(:,1));
    fprintf(fid2, ['%d, %d, %0.3f, %0.3f, %0.3f, %0.3f, %0.3f, ', ...
        'px, %d, %d, %d, %d\r\n'], numSpheres, tempBad, meanDiam, ...
        stddevDiam, medianDiam, maxDiam, minDiam, img_col_offset, ...
        img_row_offset, img_col_useful, img_row_useful);

    fprintf(fid2, '%s\r\n', headerLine3);
    fprintf(fid2, '%f,%f,%f\r\n', sorted');
    fclose(fid2);
elseif cont == 'q'
    break;
end
end

if viewImages
    close(hFig);
end
end
```

andyfunc_combineTotalStatsSIR_v2.m

```
function [finalDiam, imageID]= andyfunc_combineTotalStatsSIR_v2( ...
                                                filepath,images)
%This function loads and combines data for ALL images from ALL specified
%   vials

%Written by: Andy Richards
%Last modified: Sept 2013
%Department of Mechanical & Aerospace Engineering
%North Carolina State University, Raleigh, NC

%ALL UNITS ARE IN PIXELS SO FAR
%Filepaths should be a cell array
finalDiam = []; %global diameter array
imageID = zeros(0,2); %store the path and image number for each diam

for j=1:length(filepath)
    fprintf('Loading data from path %d/%d, %s\n',j, ...
           length(filepath),filepath{j});
    tempImageID = [];
    spheresPerVial = 0;
    for i=images
        fid = fopen([filepath{j} sprintf('%03d_stats_px_rev.txt',i)]);
        fgetl(fid); %skip the first line
        fgetl(fid); %read the number of spheres
        fgetl(fid); %skip the 3rd line
        data = fscanf(fid,'%f, %*f, %*f\r\n'); %read the radius data

        finalDiam = [finalDiam; data];
        spheresPerVial = spheresPerVial+length(data);
        %Maintain an array to track from which image number the data
        % comes from, in case you need to refer back to it
        tempImageID = [tempImageID; i*ones(size(data))];
        fclose(fid);
    end
    %save the images number mapping for this path
    imageID = [imageID; [j*ones(spheresPerVial,1), tempImageID]];
end

numSpheres = length(finalDiam); %determine the total number of spheres
avgDiam = mean(finalDiam); %mean
stddevDiam = std(finalDiam); %standard deviation
[minDiam,minIndex] = min(finalDiam);
[maxDiam,maxIndex] = max(finalDiam);
medianDiam = median(finalDiam);
```

andyfunc_combineTotalStatsSIR_v2.m (continued)

```
fprintf('--- Combined Pixel Stats ---\n');
fprintf('Spheres = %d\n', numSpheres);
fprintf('Mean = %0.4f px\n', avgDiam);
fprintf('Std dev = %0.4f\n', stddevDiam);
fprintf('Median = %0.2f\n', medianDiam);
fprintf('Max = %0.2f (path: %d, img: %d)\n', maxDiam, ...
        imageID(maxIndex,1), imageID(maxIndex,2));
fprintf('Min = %0.2f (path: %d, img: %d)\n', minDiam, ...
        imageID(minIndex,1), imageID(minIndex,2));
fprintf('Finished.\n');
end
```

andyfunc_generateFigureHistogram_v2.m

```
function [ hHistFig ] = andyfunc_generateFigureHistogram_v1( ...
                                                    finalDiam,opts,histTitle,filepath)
%This function generates a nicely formatted histogram figure and returns
% the figure handle. The figure may be printed if desired. The stats for
% the histogram may also be printed if desired.

%Written by: Andy Richards
%Last modified: Sept 2013
%Department of Mechanical & Aerospace Engineering
%North Carolina State University, Raleigh, NC

%convert the opts.units if um is desired
if strcmp(opts.units,'\mum') || strcmp(opts.units,'um')
    fprintf('Units converted to \mum\n');
    finalDiam = finalDiam*opts.pixel_size;
    opts.units = '\mum';
end

fprintf('Preparing diameter histogram.\n')

numSpheres = length(finalDiam);
avgDiam = mean(finalDiam);    %mean
stddevDiam = std(finalDiam);    %standard deviation
minDiam = min(finalDiam);
maxDiam = max(finalDiam);
medianDiam = median(finalDiam);
modeDiam = mode(finalDiam);

fprintf('--- Histogram Stats ---\n');
fprintf('Spheres = %d\n',numSpheres);
fprintf('Mean = %0.4f %s\n',avgDiam, opts.units);
fprintf('Std dev = %0.4f\n',stddevDiam);
fprintf('Median = %0.2f\n',medianDiam);
fprintf('Mode = %0.2f\n',modeDiam); %caution, this may be unreliable due
% to the continuous nature of the
% diameter measurements

fprintf('Max = %0.2f\n',maxDiam);
fprintf('Min = %0.2f\n',minDiam);

totalSkew = skewness(finalDiam);
totalKurt = kurtosis(finalDiam);
fprintf('Skewness = %0.3f\n',totalSkew);
fprintf('Kurtosis = %0.2f\n',totalKurt);

if opts.showHistogram
    [n,xbins] = hist(finalDiam,opts.histogramBins);
```

andyfunc_generateFigureHistogram_v2.m (continued)

```

hHistFig = figure(1);
bar(xbins,n, 'BarWidth',1, 'FaceColor',[0.9 0.9 0.9]);
xlabel(sprintf('Diameter (%s)',opts.units));
ylabel('Microsphere count');
set(gca, 'XLim',[21.5 36.5], 'XTick',[22:1:36]);
xlims = xlim;
ylims = ylim;
OGaxis = gca;
%title(sprintf('Diameter histogram - %s',histTitle));
axes %create new axis to overlay on top of histogram
pMean = plot(avgDiam*[1 1],[0 1], 'r--', 'LineWidth',2);
%hold on
%show the stdDev lines
%plot((avgDiam+stddevDiam)*[1 1],[0 1], 'g--', 'LineWidth',2);
%plot((avgDiam-stddevDiam)*[1 1],[0 1], 'g--', 'LineWidth',2);
%hold off
xlim(xlims);
ylim([0 1]);
axis off;

if 0 %dont show the text label
    axes; %create new axis for text labels
    xlim([0 1]);
    ylim([0 1]);
    axis off
    tx = { sprintf('Spheres = %d',numSpheres);...
          sprintf('Mean = %0.1f %s',avgDiam,opts.units); ...
          sprintf('Std. Dev. = %0.2f %s',stddevDiam,opts.units);...
          sprintf('Median = %0.1f %s',medianDiam,opts.units);...
          sprintf('Mode = %0.1f %s',modeDiam,opts.units);...
          sprintf('Max = %0.1f %s',maxDiam,opts.units);...
          sprintf('Min = %0.1f %s',minDiam,opts.units);...
          sprintf('Skewness = %0.3f %s',totalSkew);...
          sprintf('Kurtosis = %0.1f %s',totalKurt)};

    text('Units','normalized','Position',[0.05,0.95], 'String',tx, ...
         'VerticalAlignment','Top');%EdgeColor','black','Margin',10);
end

%overlay the t-distribution
tparams = mle(finalDiam, 'dist', 'tlocationscale');
tloc = tparams(1);
tscale = tparams(2);
tshape = tparams(3);
tloc = 29.1; %previously determined values
tscale = 0.666;
tshape = 3.11;

xtDist = xbins(1):0.1:xbins(end);

```

andyfunc_generateFigureHistogram_v2.m (continued)

```
t3 = pdf('tlocationscale',xtDist,tloc,tscale,tshape);
t4 = t3*0.25*numSpheres; %scale by the bin width and # of spheres
set(hHistFig, 'CurrentAxes',OGaxis);
hold on;
ptDist = plot(xtDist,t4, 'b-.', 'LineWidth',1.5);
hold off;
    %legend(sprintf('Mean'),sprintf('Mean \\\pm \\\sigma'));
legend([pMean ptDist], 'Mean', 't-Distribution');

if opts.printFigs
    fprintf('Saving histogram to file.\n');
    set(0, 'CurrentFigure',hHistFig);
    set(hHistFig, 'PaperPositionMode', 'auto');
    print('-dpng', '-r150', [filepath, 'figurehist.png']);
end

if opts.writeStatsFile
    fprintf('Saving histogram stats to file: %s.\n', ...
        [filepath, 'hist_stats.txt']);
    f = fopen([filepath, 'hist_stats_figure.txt'], 'w+');
    fprintf(f, ['Num Spheres, Mean Diam, Std Dev, Median Diam, ', ...
        'Max Diam, Min Diam, Units\r\n']);
    fprintf(f, '%d, %f, %f, %f, %f, %f, %s\r\n', numSpheres, avgDiam, ...
        stddevDiam, medianDiam, maxDiam, minDiam, opts.units);
    fclose(f);
end
end
end
```