NEW TECHNOLOGIES FOR FABRICATING BIOLOGICAL MICROARRAYS

By

Bradley James Larson

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

(MATERIALS SCIENCE)

at the

UNIVERSITY OF WISCONSIN – MADISON

2005

New technologies for fabricating biological microarrays

Bradley James Larson

Under the supervision of Professor Max G. Lagally

At the University of Wisconsin–Madison

Microarrays, composed of thousands of spots of different biomolecules attached to a solid substrate, have emerged as one of the most important tools in modern biological research. This dissertation contains the description of two technologies that we have developed to reduce the cost and improve the quality of spotted microarrays.

The first is a device, called a fluid microplotter, that uses ultrasonics to deposit spots with diameters of less than 5 μm. It consists of a dispenser, composed of a micropipette fastened to a piece of PZT piezoelectric, attached to a precision positioning system. A gentle pumping of fluid to the surface occurs when the micropipette is driven at specific frequencies. Spots or continuous lines can be deposited in this manner. The small fluid features conserve expensive and limited-quantity biological reagents. Additionally, the spots produced by the microplotter can be very regular, with coefficients of variability for their diameters of less than 5%.

We characterize the performance of the microplotter in depositing fluid and examine the theoretical underpinnings of its operation. We present an analytical expression for the diameter of a deposited spot as a function of droplet volume and wettability of a surface and compare it with experimental results. We also examine the resonant properties of the piezoelectric element used to drive the dispenser and relate that to the frequencies at which pumping occurs. Finally, we propose a mechanism to explain the pumping

behavior within the microplotter dispenser.

The second technology we present is a process that uses a cold plasma and a subsequent *in vacuo* vapor-phase reaction to terminate a variety of oxide surfaces with epoxide chemical groups. These epoxide groups can react with amine-containing biomolecules, such as proteins and modified oligonucleotides, to form strong covalent linkages between the biomolecules and the treated surface. The use of a plasma activation step followed by an *in vacuo* vapor-phase reaction allows for the precise control of surface functional groups, rather than the mixture of functionalities normally produced. By maintaining the samples under vacuum throughout the process, adsorption of contaminants is effectively eliminated. This process modifies a range of different oxide surfaces, is fast, consumes a minimal amount of reagents, and produces attachment densities for bound biomolecules that are comparable to or better than commercially available substrates.

We show applications of these two technologies in the fabrication of protein microarrays, enhancement of MALDI mass spectrometry, deposition of polymer electronics, directed growth of carbon nanotubes, and the chemical modification of carbon-containing materials.

Acknowledgements

I could not have performed any of the work that I describe here on my own. Therefore, I'd like to express my extreme gratitude to the following people for their help, in and out of the lab, during my time here at UW-Madison:

In this document, I describe the development of a couple of cold-plasma-based chemical treatments. This was work that was performed in an equal-parts collaboration with Professor Ferencz Denes, whose excitement about the field of cold plasmas is contagious. A then-student of his, Emilio Cruz-Barba, performed the plasma treatments on carbon-containing materials. Jason Helgren did much of the experimental work in the development of the plasma treatment of oxides. I was able to get more done working with Jason than with any other researcher I've met. I'd also like to thank Sorin Manolache for his advice and aid during the period that I worked on this and Albert Lau for performing analyses on some of our samples.

My work on the fluid microplotter grew out of an early collaboration with Professor Amit Lal and Chung Hoon Lee, and I would like to thank them for giving me the opportunity to learn about novel applications of ultrasonics. Anna Clausen and Albert Lau performed many of the experimental studies that allowed us to characterize the operation of the microplotter. Avery Frey conducted the experiments that tested for protein activity in microplotter-exposed solutions. I would also like to acknowledge Professor Robert Blick's research group for allowing me free access to their laboratories in order to perform the impedance-based studies of piezoelectric resonances.

As far as the other applications of these two technologies, all of the nanotube work was either performed by members of Professor Mark Eriksson's research group or was done with their assistance. Matt Marcus of the Eriksson Group and Todd Narkis of the Lagally Group did all of the research involving the spotting of nanotube catalyst. Jason Simmons and Matt Marcus provided the nanotubes used for the nanotube plasma functionalization experiments and helped with the experiments themselves. In addition to being excellent researchers, these guys could always make me laugh. The MALDI mass spectrometry experiments were done in collaboration with Professor David Barnes, who suggested the experiments to begin with, and Dr. Martha Vestling, who ran the MALDI equipment for these studies. The deposition of conducting polymers for the fabrication of LEDs was done in active collaboration with Professor Michael Winokur and his student, Hyunseok Cheun. I didn't always understand what Professor Winokur was talking about, but I knew it was important.

The folks over at Professor Lloyd Smith's research group were kind enough to let me use some of their equipment and take part in their weekly group meetings, where I learned quite a bit about surface chemistry. I also picked up quite a few useful tips and techniques from hanging around their labs.

You couldn't find a better group of people to surround yourself with than those in the Lagally Group (including our honorary member, Charles). Our group meetings could be brutal, but what doesn't kill you makes you stronger. I'd like to single out one of those people for special recognition. Susan Gillmor took me under her wing when I started graduate school and taught me most of the skills I would need. Not only that, she also introduced me to the research that would later become the basis for what I describe here. Thanks again, Susan.

I also would like to thank Diana Rhoads for all the ways that she has helped me (and other Materials Science Program students) over my time here. The MSP would fall apart if she wasn't there to hold it together.

The final person at the University of Wisconsin that I would like to thank is my advisor, Professor Max Lagally. He has given me the opportunity to do the research that I wanted to do, something that few others would allow. His criticisms have always been constructive and he has always fiercely defended me when he felt that I was getting the short end of the stick. Unfortunately, I still haven't learned how to be concise, with the size of this dissertation as proof.

It would take a document longer than the one in your hands to adequately describe what my family means to me. The work ethic and focus that I needed to make it this far I learned entirely from my parents. That, and I could always drive home and get them to do my laundry when I couldn't find the time to do it myself. My brother is now headed down this same road, so I wish him the best of luck in his own research even though I know he won't need it. I haven't thanked any of them nearly enough for what they have done, and continue to do, for me.

Thank you all again for making this possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Microarrays have become one of the most important tools available to the modern biological researcher, allowing hundreds to thousands of assays to be performed in a single experiment at comparatively low cost. In this dissertation, we describe the development and application of two technologies that show promise for the fabrication of lower-cost, higher-quality biological microarrays. The first technology, and the focus of the majority of this document, is a device called a fluid microplotter that uses ultrasonics to deposit picoliter volumes of fluid onto a surface. The second is a means of using cold plasmas to chemically treat the surfaces of a wide range of materials in order to covalently bind biomolecules to them. While the primary focus in the development of these technologies has been to improve microarray production, they both have a number of applications outside of this field.

1.1 Microarrays

A microarray is composed of many features or spots at which different probe biomolecules have been attached to a flat surface. When a probe biomolecule is attached to a substrate (in this dissertation, substrate refers to a solid support), the binding of a target to the probe may be detected using a variety of methods, commonly by means of a fluorescent tag on the target. When discrete spots of different probe biomolecules are placed on a solid surface, their known 2-D locations allow thousands of different probe-target interactions to be observed simultaneously.

To appreciate the value of microarrays in biological research and potentially in disease diagnosis, it helps to start at the central dogma of molecular biology. Within the nucleus of a cell, deoxyribonucleic acid (DNA) stores the information on how to manufacture the proteins that compose a living organism. To translate that information from DNA to proteins, messenger ribonucleic acid (mRNA) is first transcribed from DNA. This mRNA starts being transcribed at a specific region on a DNA strand, called a gene, that codes for a specific protein and stops at the end of that region (Figure 1.1a). The mRNA then travels out of the nucleus, where it provides the information to ribosomes necessary to start production of a protein (Figure 1.1b). Which genes are transcribed to mRNA, or expressed, and how many mRNA copies of a gene are transcribed is dictated by a complex set of interactions within a cell. This is referred to as gene expression, and the pattern of gene expression is what sets a muscle cell apart from a nerve cell, since within a single organism they all share the same DNA sequence.

In order to examine the gene expression pattern in a certain type of cell, an experimental method is needed to look at each of the thousands of genes that may be expressed or not in parallel. In the past, this required using 96- or 384-well plates, one well for each gene of interest. This was costly, due to the volumes of reagent and test solution needed, and time-consuming, due to the need to pipette reagents into each of these wells and run reactions within them.

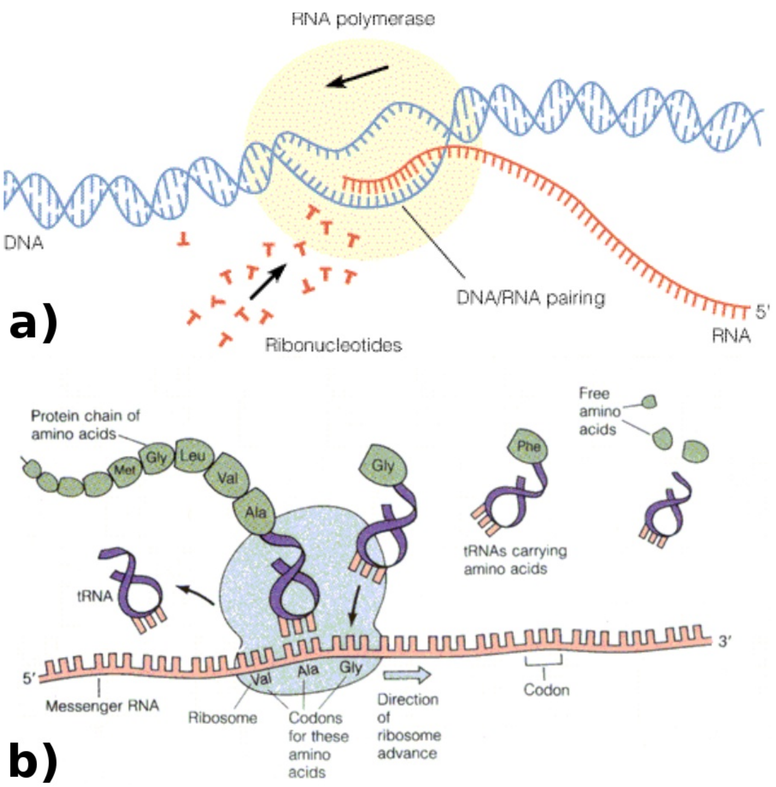

Figure 1.1: The central dogma of molecular biology. To produce proteins from DNA, a) mRNA is first transcribed from DNA and then b) transported outside of the cell nucleus to ribosomes, where proteins are produced from the sequence of the mRNA. Images are from Mathews, van Holde, and Ahern.[1]

1.1.1 DNA microarrays

DNA microarrays made this kind of gene expression analysis practical.[2, 3, 4] DNA microarrays are composed of discrete spots on a surface where specific sequences of single-stranded DNA are immobilized, as depicted in Figure 1.2a. These immobilized single strands of DNA, referred to as probes, recognize and bind with complementary target sequences of DNA in a process known as hybridization (Figure 1.2b). To detect when hybridization occurs, normally the target DNA strands are tagged with either a radioactive label or, more commonly, a fluorescent dye molecule. Fluorescence at a spot indicates that a specific sequence has hybridized to a probe and the intensity of that fluorescence is often proportional to the concentration of target present. By making an array of these spots of immobilized single-stranded DNA and keeping track of what sequence is immobilized where, thousands of sequences can be probed for at once on a single array. Microscale spots mean that this array can be shrunk down to the size of a standard glass microscope slide. These microarrays use a tiny fraction of the reagents used in the 96- or 384-well plate experiments and they enable experiments to be performed in the span of hours rather than days. They have been used to analyze cancers[5], examine toxicity of various compounds[6], identify pathogens[7], and even perform computing operations[8, 9].

There are two ways of producing a DNA microarray: synthesizing short DNA strands right off of the substrate surface itself, or depositing presynthesized or extracted DNA strands on a chemically modified surface on which they are immobilized. Both methods have their advantages and disadvantages and are used today to fabricate microarrays in use in industry and academia.

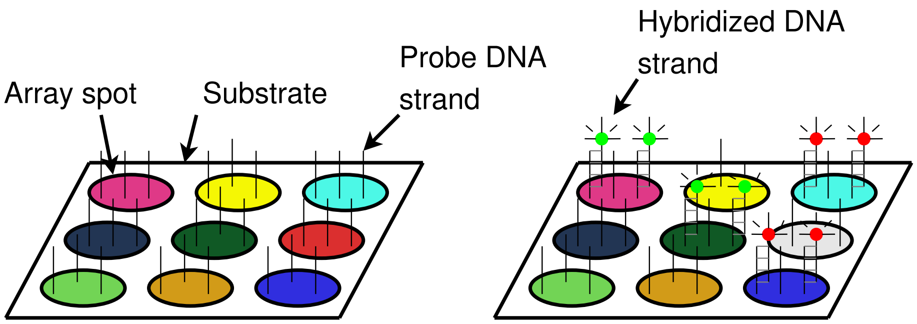

Figure 1.2: Schematic diagram of a DNA microarray. a) A DNA microarray is composed of many different spots where single strands of DNA with different sequences are immobilized. b) To detect a target sequence of DNA in solution, hybridization is used. Normally the target is tagged with a fluorescent dye, so that spots where hybridization have occurred fluoresce with an intensity proportional to the concentration of target present.

Fodor, *et al* first described a means of doing *in situ* DNA synthesis using techniques familiar to the semiconductor industry.[10, 11] First, a glass or silicon (with native oxide) surface is terminated with a photolabile protecting group. UV light is shone through a chrome mask, removing the protecting groups in the areas not covered by the chrome of the mask. A solution containing one of the nucleotide bases (adenosine, guanine, cytosine, or thymine) is then washed over the surface. The nucleotides in the solution attach only to the unprotected areas of the surface. The nucleotides themselves have photolabile protecting groups on the side of the molecule not bound to the surface. This exposure through a mask is repeated for the other three nucleotide bases, with a different mask each time, to pattern one step of the oligonucleotides (short strands of DNA) being synthesized on the surface. This process is repeated to synthesize the complete DNA sequences at each spot in the microarray. Very regular features as small as 11 μm may be produced this way. Figure 1.3 demonstrates this *in situ* synthesis.

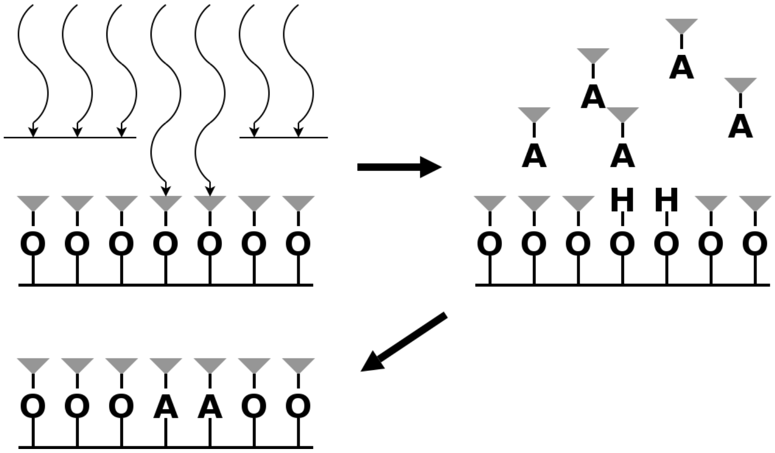

Figure 1.3: One step in the photolithographic DNA microarray synthesis process.[10, 11, 12] a) UV light is shone through a mask (or directed by an array of micromirrors) onto a surface that is terminated with photolabile protecting groups. b) These groups leave in only the area exposed to light. A solution containing one of the DNA bases is washed over the surface, and c) that base binds to the area deprotected by the light. The base has a photolabile protecting group attached, so the surface is once again terminated with blocking groups. This process is repeated for each of the four DNA bases to build a single level of the DNA strands coming off the surface.

This process has drawbacks, however. In its original design, four different photolithographic masks need to be made for each step in the length of DNA being synthesized on the surface (25 bases being the typical length of DNA synthesized using this method[13]) . These masks can be very expensive and time-consuming to fabricate individually, a serious problem when 100 different masks are needed to synthesize a given microarray. However, Singh-Gasson, *et al* developed a maskless array synthesis method that uses digital micromirror devices to direct light to the appropriate places on the surface at each attachment step.[12] Researchers at Agilent have also adapted their inkjet technology to the chemical synthesis of DNA microarrays[4]. Rather than use photolithography to deprotect certain areas and then flow a solution of a specific nucleotide base across the entire surface, they deprotect the entire surface and print nucleotide bases at only certain spots. Both of these techniques make practical the one-off *in situ* synthesis of custom DNA microarrays. Unfortunately, all of these synthesis methods are limited to creating short oligonucleotides, as errors build up at each step in the synthesis process (if each synthesis step has a yield of 99%, then only 78% of the DNA strands in an array of 25-mer oligonucleotides will have the right sequence). An even more significant limitation of *in-situ* synthesis is that, while it has been modified to assemble short polypeptides[14], it cannot be used for most proteins, polysaccharides, or for other biological molecules of interest.

The second major method of DNA microarray fabrication is the deposition of presynthesized or extracted DNA strands onto a chemically modified surface. To start with, a surface of some type is chemically modified to immobilize biomolecules (these chemical treatments will be discussed in detail in Section 1.3. Then, DNA-containing solutions from some source are loaded into a piece of fluid-deposition equipment that deposits microscale spots of the various DNA solutions in an array on the chemically treated surface. Several different technologies for doing so are described in Section 1.2. While depositing prepared biological solutions onto a treated surface makes possible the fabrication of microarrays with any type of biomolecule, to date the deposition technologies have not been able to achieve the feature sizes found on microarrays made using the *in situ* synthesis approach. There is also the problem that a solution of each of the different biomolecules to be deposited needs to be prepared ahead of time, usually at significant cost, while all the *in situ* methods need is to electronically store the DNA sequences to be synthesized.

Once a DNA microarray is created by either of these methods, it is ready for use in a gene expression experiment. Such an experiment starts with two separate cell cultures to be compared.[3] They may be both of the same cell type, from the same organism, but with one cell culture exposed to an outside stimulus, or be of two different cell types, with one being from cancerous tissue and the other from normal tissue, etc. mRNA is separately extracted from both of the cell cultures. It is then transcribed back to complementary DNA (cDNA) using an enzyme called reverse transcriptase. During or after the transcription process, these strands of cDNA are labeled with a fluorescent dye, with a different color being used for cDNA from either cell culture. Typically, Cy3 (absorption: 550 nm, emission: 570 nm) and Cy5 (absorption: 649 nm, emission: 670 nm) are the two dyes used for this. After labeling, these two different cDNA solutions are mixed together and spread across the surface of the microarray. Hybridization is allowed to occur and then the microarray is washed to remove any non-hybridized cDNA. The microarray is then read in a microarray scanner, a piece of equipment that takes a laser at a wavelength close to the absorption maximum of the dye to be detected and

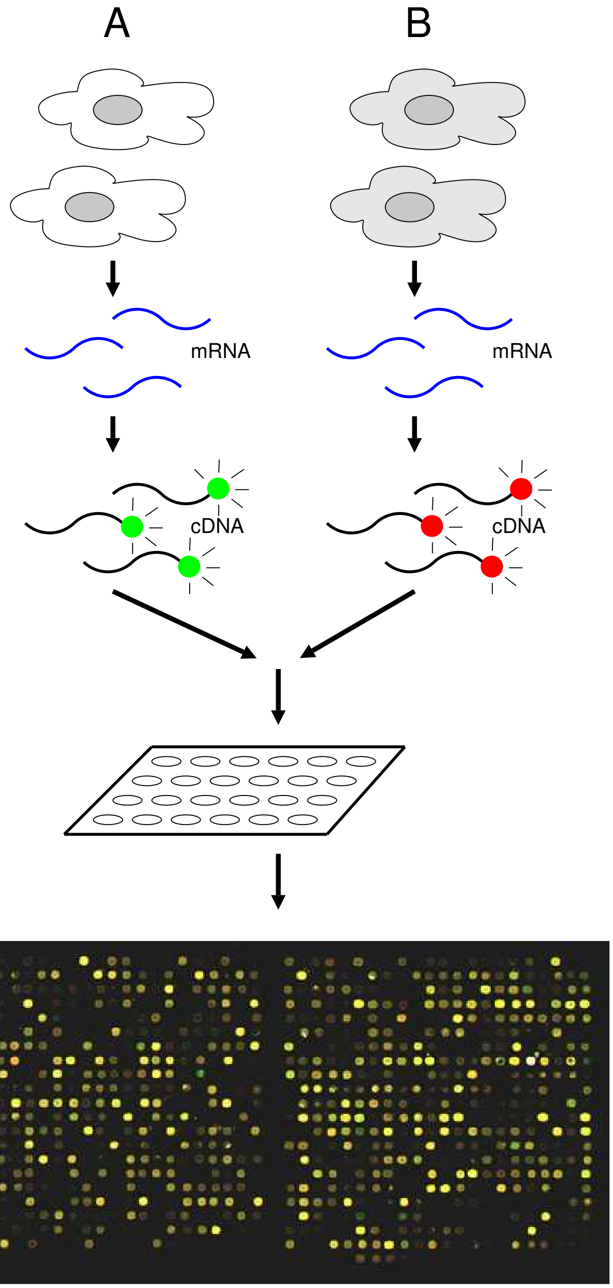

Figure 1.4: The steps in a DNA microarray experiment. mRNA is extracted from two different cell cultures, reverse transcribed to cDNA (while being tagged with fluorescent dyes), and this cDNA is flowed over the surface of a microarray. Finally, the microarray is washed and scanned to detect fluorescence from cDNA hybridized to the array. The image of a magnified portion of a scanned microarray is from *http://mphywww.tamu.edu/Microarray/scans_Human_7.5K_AA.html*.

scans it across the microarray's surface while a photomultiplier tube (behind a bandpass filter or similar that blocks out all light but that emitted by the dye) detects the intensity of fluorescence. This creates a 2-D image of the microarray surface, showing bright spots where target hybridized to probe and dark areas where target did not. The slide is scanned twice, once for each dye, and a multicolor image is generated by overlaying the two separate color images from the different scans. More of one color than the other in a microarray spot indicates that the gene with a DNA sequence complementary to the target in that spot was more strongly expressed in one of the cell cultures relative to the other. By studying these expression patterns, much can be determined about what is actually going on inside of the cells themselves. A flowchart of this process is present in Figure 1.4.

1.1.2 Protein microarrays

DNA microarrays were the first to be introduced, in large part because the stringency of nucleotide hybridization assures specific interactions between probe and target. DNA is also a reasonably robust biomolecule and one strand of DNA will behave almost exactly like another with a different sequence. However, the use of DNA microarrays in gene expression or even in diagnostic applications hinges on the assumption that the levels of mRNA present within a cell directly translate to higher levels of the protein that is coded for by a specific sequence of mRNA. It has been found that this is not always the case.[15, 16] Additionally, DNA microarrays cannot be used in drug development, where protein-small molecule interactions need to be measured, and in most diagnostic applications, where the detection of proteins present in serum is key. For these reasons, much effort is now being directed toward the reliable manufacture of protein microarrays.

Unlike DNA microarrays, which have one configuration (bound probe DNA hybridizes with fluorescently tagged target DNA of a complementary sequence), protein microarrays come in many different varieties based upon what the actual microarray experiment is testing for. Figure 1.5 is a schematic demonstrating the most popular formats for protein microarrays: protein expression profiling.[17], serum-based diagnostics, protein-protein interactions[18], and small molecule-protein binding (drug interactions, for example)[19]. As with DNA microarrays, the most common detection methodology for protein microarrays is fluorescence. Figure 1.6 shows three possible configurations that are used to detect the binding of probe to target using fluorescence. The first, and most widely used, method is simply to attach fluorescent tags directly to proteins in the solutions to be examined, similar to the process used for DNA. Fluorescent molecules with NHS functional groups (described later in Section 1.3) attach to free amine groups on the proteins in solution. Two different color dyes are used, again similar to the process for DNA microarrays. However this direct labeling does not attach to all proteins and many of those it does work for have their properties changed by the labeling process (this is especially true when dealing with small molecules, such as drug candidates). An alternate means of detection uses a fluorescently-labeled secondary antibody that ”sandwiches” a target biomolecule between it and the surface-bound probe. This sandwich assay avoids the problems of direct labeling, but it adds a step to the microarray experiment procedure and requires that a fluorescently tagged antibody exist for every target biomolecule.

As mentioned in the previous section, the *in situ* synthesis approach that has been

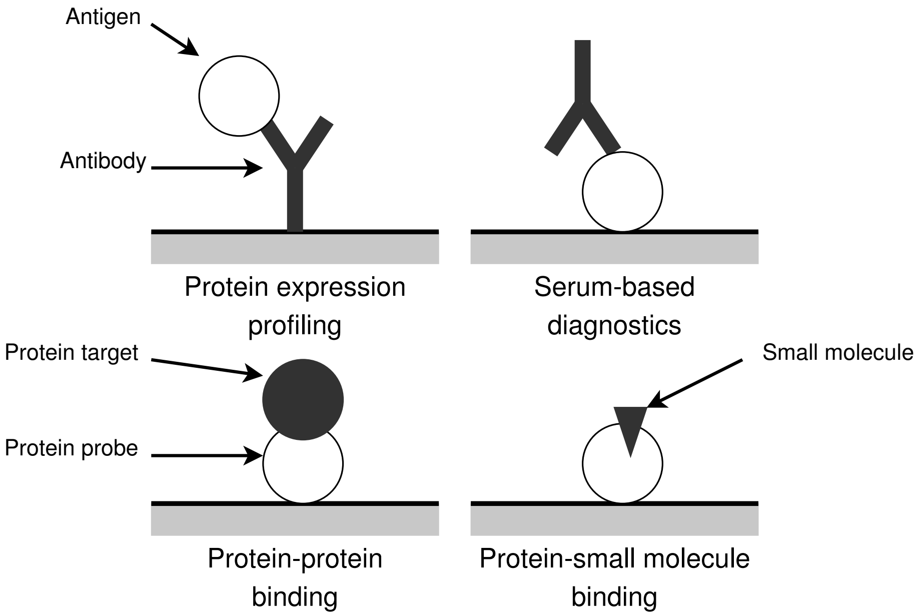

Figure 1.5: Types of protein microarrays. These are among the most popular configurations for protein microarray experiments. Schematic is adapted from Schena.[20]

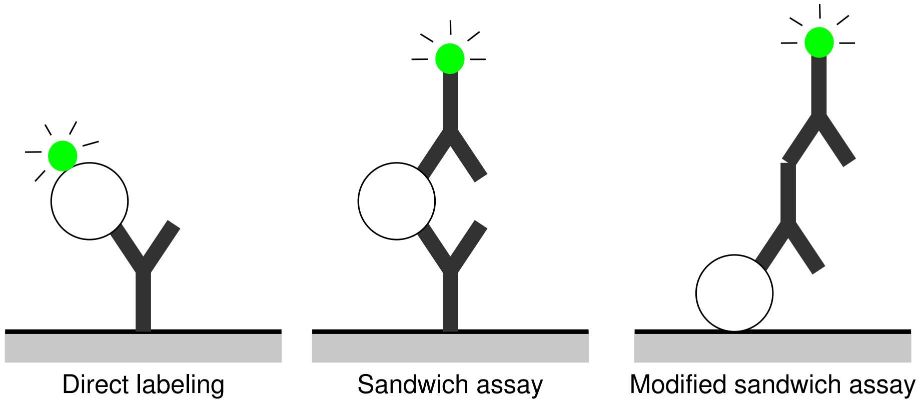

Figure 1.6: Means of using fluorescent detection in protein microarrays. These are three of the ways that fluorescent tags are used in protein microarray experiments. Although antibody-antigen binding is depicted in this figure, the same general principles can be applied to the other types of protein microarrays shown in Figure 1.5.

widely used in the fabrication of DNA microarrays cannot be used to make protein microarrays. While proteins, like DNA, are made of long strands of smaller molecules (amino acids in this case, instead of nucleotides) they present several problems when attempting to synthesize them. First, while there are only four different nucleotide bases in DNA, there are twenty different amino acids that go into proteins. This means that twenty different deprotection steps are required for every amino acid in the length of proteins being synthesized on a surface, not just the four needed for DNA. Second, most proteins are much longer than the DNA strands currently being synthesized. Most proteins are made of 100-4000 amino acids, and there are currently no reliable means of synthesizing strands this long. Finally, the true function of proteins comes not from the sequence of amino acids that makes them up, but the three-dimensional shape into which they fold. This folding process takes place either during or soon after the synthesis of the protein within a cell. ”Helper” structures within a cell often direct a protein to fold into a stable shape that it would not otherwise take. This folding process does not work very well when one end of a synthesized protein is tethered to a solid support, and without the presence of a natural cellular environment many proteins will never achieve their proper shape. It is for these reasons that the only method used to fabricate protein microarrays is the deposition of protein solutions onto a chemically treated surface.

However, the binding of prepared proteins to a surface is not without its share of problems. The protein microarray experiments shown in Figure 1.5 all rely on the specific binding of a protein to another molecule. This binding occurs at receptor sites on the protein. These receptor sites are located in specific places on a protein. When a protein is deposited onto a surface, this receptor site may be facing toward that surface. If the chemically treated surface binds the protein in that configuration, it will be unable to react with its target. Thus, with most protein binding chemistries some portion of the bound proteins are inactive. Chemistries have been developed to bind proteins only at specific portions of their structure, properly orienting them for microarray experiments, but these chemistries have to be custom-designed to fit the specific proteins being deposited and often require the proteins themselves to be reengineered. Aside from the issue of orientation is the problem that proteins do not interact well with solid surfaces, so the act of chemically binding them may disrupt their folded three-dimensional structure and deactivate them. To reduce the impact of this, a chemical ”tether” can be used to bind the proteins, while keeping them a sufficient distance from the solid support.

These issues, among others, make protein microarrays far more difficult to manufacture than DNA microarrays, but the potential applications for protein microarrays have such a far-reaching impact that many laboratories and companies are investing great amounts of money and time in their development.

1.2 Fluid deposition

The method of fabricating a microarray using presynthesized or extracted biomolecules requires a means of delivering biomolecule solutions to precise locations on a surface. In Section 1.1.1, we indicated that there are a few technologies currently employed for this purpose. Each of these fluid deposition devices has its advantages and disadvantages.

The goal of microarray fabrication methods is to create arrays that are as dense as possible, with small, regular spots. Dense arrays allow for greater numbers of analytes to be placed per unit area. Arrays of greater density allow for replication of detection in

Diameter (μm)	**Volume (pL)**
20	0.55
30	1.8
50	8.6
100	69
150	233
200	552
250	1078

Table 1.1: The relationship between spot diameter and droplet volume. The spot volumes were calculated from Equation 3.10 in Chapter 3, assuming the spots were from droplets of water deposited on glass.

multiply segregated locations on the surface, leading to greater data integrity and analytical security. Small, consistent spot sizes are required for dense arrays. More importantly, use of the smallest possible features (consistent with present-day detection capabilities) assures that the minimal volume of scarce and valuable biological solutions is consumed for each assay. Table 1.1 lists the volumes of solution consumed when producing various sizes of spots. For example, a 200 μm diameter spot consumes approximately 550 picoliters, whereas a 20 μm diameter spot consumes 0.55 picoliters. Hence, a 10-fold decrease in feature diameter leads to a 1000-fold reduction in reagent use. Moreover, consistency in spot morphology allows features to be placed more closely without risk of cross-contamination, and makes the results of a microarray experiment more reliable and much easier to analyze. [21, 22]

1.2.1 Existing technologies

Many different types of fluid deposition equipment exist, but the most popular are machines that use notched quill-pins and mechanical force to dislodge solution onto a surface, as shown in Figure 1.7.[23, 24] These quill-pin spotters can deposit spots as small

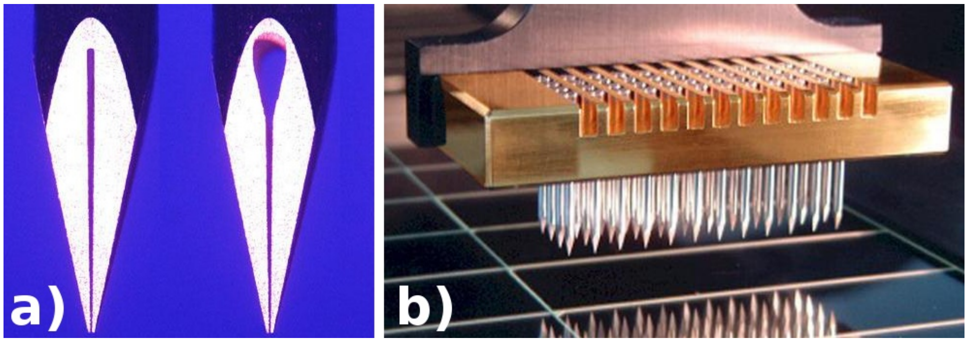

Figure 1.7: Examples of mechanical quill-pin spotters. a) Metal or ceramic pins with internal notches and sharp tips are dipped in solution to load the inside volume, then b) tapped gently on a surface to deposit small amounts of that solution in microscale spots. To improve throughput, pins are placed in parallel and used to deposit many subarrays at the same time. Images are of the TeleChem ArrayIt Stealth microarray spotting pins, from *http://arrayit.com/Products/Printing/Stealth/stealth.html*.

as 75μm in diameter, but more often create spots in the 150-200 μm size range. Quill-pin spotters are a well-established technology and step-by-step instructions for how to build a microarray spotter are even available on several research groups' web sites.[25]. Unfortunately, this technology has significant disadvantages. The notched tips can take up only enough solution to create roughly 100-150 spots on average and usually must return several times to reload with solution in order to complete a spotting run. The need for machining makes the individual tips expensive. They are also very sensitive to the solution being deposited, as certain aqueous solutions erode the alloy of the tips. Because quill pins are tapped on the surface, damage may occur to more fragile substrates and the tips will wear down over time. Finally, the spots produced by these tips are large and irregular, which means that to avoid spot-to-spot contamination they must be placed farther apart.

The quill-pin-based spotters use what is referred to as contact dispensing, due to the mechanical contact between pin and substrate required to dispense fluid, but there is an entire class of noncontact dispensers with their own special properties. First among these are the inkjets. Inkjet technology is familiar to many due to the widespread use of inkjet printers attached to personal computers. An inkjet print head is composed of a microscale cavity fabricated using techniques common in the semiconductor industry. At one end of this cavity is a nozzle from which droplets of fluid are dislodged by one of two means. The first technique, employed by printer manufacturers Hewlett-Packard and Canon, uses a small heater element to vaporize a small amount of the fluid being dispensed, forming a bubble that forces a droplet of fluid out of the end of the nozzle (Figure 1.8a). This is sometimes referred to as bubblejet printing. The second method, present in Epson printers, involves a piezoelectric crystal that compresses the fluid-containing cavity and ejects a droplet of fluid from the nozzle (Figure 1.8b). Through careful tuning of the inks used and the structure of the inkjet printhead, both types of inkjets can now deposit droplets that form spots less than 5 microns in diameter. The throughput of an inkjet printhead has also been the focus of much development; modern inkjets can deposit thousands of spots per second.

While such printing methods are fast, they are impractical for use in deposition of multiple previously prepared solutions. The print heads are designed for use with a limited number of solutions; use with many probe solutions for fabrication of arrays requires a large initial quantity of each analyte and incurs much waste. For these reasons, print-based methods in practice are confined to use in the chemical synthesis of DNA [26, 4] Rather than use photolithography to deprotect regions of a surface and then flow a nucleotide base across the entire surface, chemical synthesis means that the nucleotide bases are deposited only where needed and no light deprotection is needed.

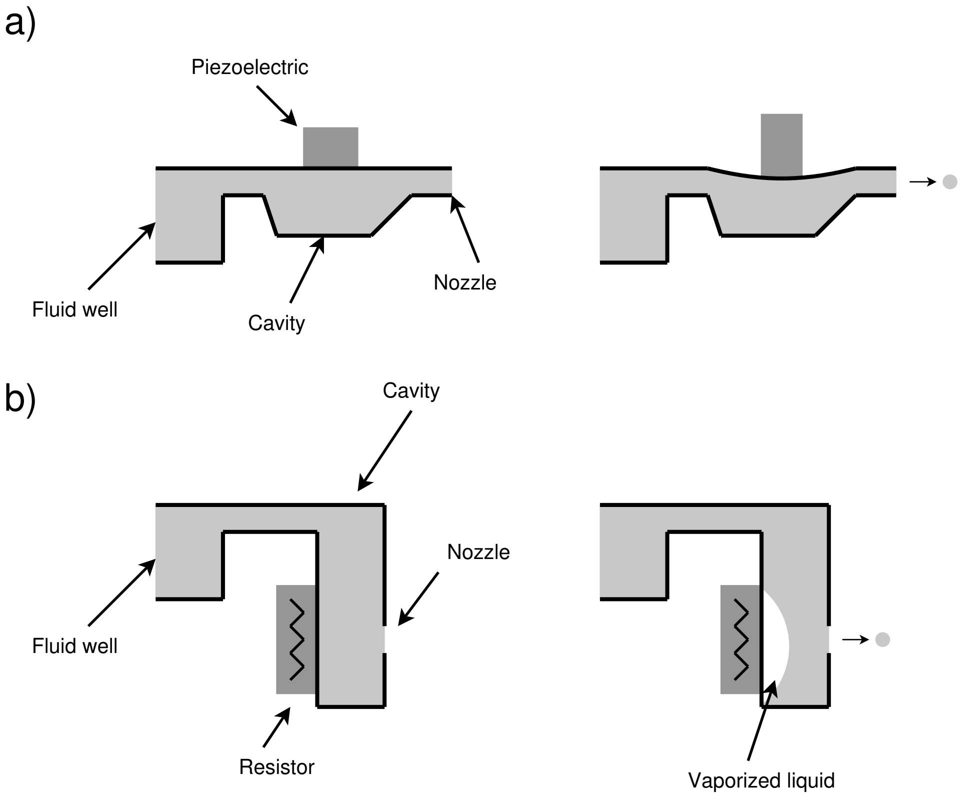

Figure 1.8: Schematics of two types of inkjets. a) One type of inkjet uses a piezoelectric element to depress a diaphragm or similar structure, compressing fluid within a cavity and ejecting a droplet from a nozzle. b) A second type of inkjet (also known as a bubblejet) relies on a resistor to heat the fluid in a cavity, creating a bubble that forces a droplet of fluid out of the nozzle.

Inkjets also suffer from the need to tune the fluid properties of each solution used in the printhead to get reliable deposition of small spots. Inkjets used in commercial inkjet printers can deposit sub-5-micron-sized spots only because the specific inks they use have been refined to produced these small spots (one reason why specific ink cartridges must be purchased for each manufacturer's printer model). The varying viscosities, rates of evaporation, etc. of many different biological solutions mean that only a small number of these solutions would be usable as is; most would require modification to produce small spot sizes. The bubble jets must also use solutions that are resistant to the heating introduced, in the case of biological solutions this may limit the number of biomolecules that can be deposited or may require some sort of stabilizing agent. Inkjets are also prone to clogging if solutions with viscosities, salt contents, etc. that are significantly different from their well-tuned inks are used.

A related technology relies on hollow capillaries that are surrounded by a ring of piezoelectric (Figure 1.9). When current is applied to the piezoelectric, it squeezes the capillary and ejects a droplet from one end. This is identical in function to one of the types of inkjet printers described above, but the use of a capillary means that smaller working volumes of fluid are required. The end of a capillary may be dipped in a solution well and fluid to be deposited may be aspirated up into the capillary. This eliminates the need for a large fluid well connected to the back end of the dispenser and allows for much smaller volumes of fluid to be worked with. Unfortunately, the squeezing action of the piezoelectric only dislodges a droplet if fluid has filled the capillary up to the point where the piezoelectric is wrapped around the capillary. In practice, this means that up to 10 microliters of the fluid to be dispensed need to be loaded into the dispenser before deposition can begin, meaning that a relatively large volume of fluid is wasted

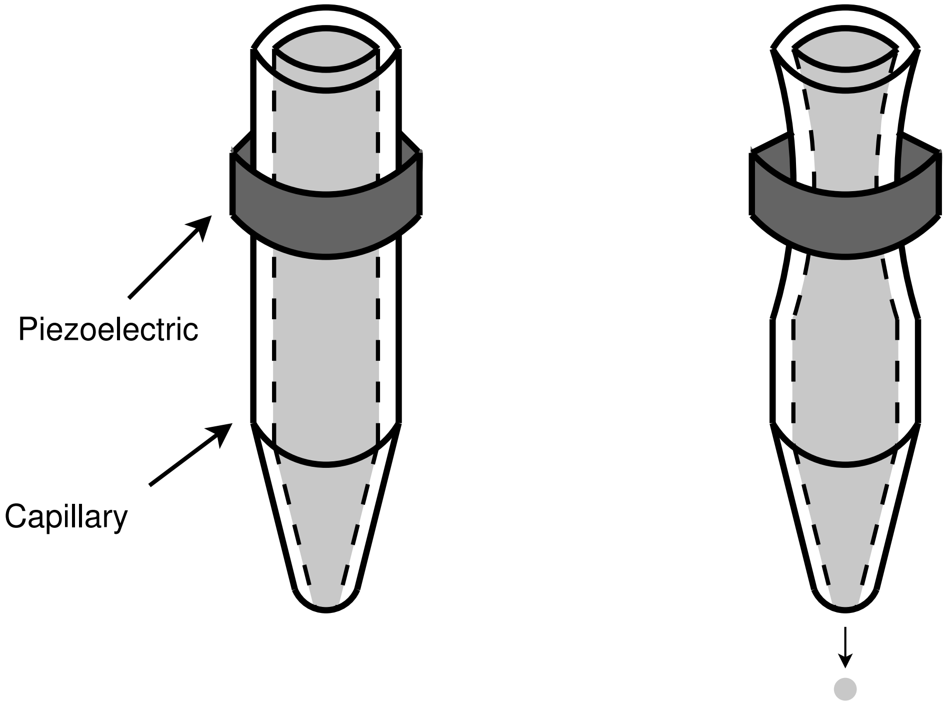

Figure 1.9: Schematic of a ringed-piezoelectric dispenser. A hollow capillary is surrounded by a ring of piezoelectric. When current is applied to the piezoelectric, it squeezes the capillary (squeezing motion is exaggerated in this diagram) and forces out a droplet of fluid.

when switching fluids. There have also been problems with inconsistent operation of these dispensers. Finally, the spot size deposited by current ringed-piezoelectric devices is on the order of 150-200 μm, which is still quite large.

A completely different means of fluid deposition has been developed by researchers at Northwestern, called dip-pen nanolithography.[27, 28] Shown in Figure 1.10, this technology uses an atomic force microscope (AFM) to draw spots and lines of nanoscale dimensions. AFMs are commonly used in a variety of research settings to image nanoscale features on a sample's surface. They operate much like a record player, with a very sharp pyramid placed on the end of a cantilever. When brought very close to a surface, the cantilever will deflect up and down as it is moved along the surface, tracing out its topography. These tips are very expensive to fabricate and have a limited life in routine use.

Dip-pen nanolithography is a technique in which the AFM tip is used to write features onto a surface instead of tracing its topography. To do this, the tip is first dipped into a solution. The solution will wick up the outside of the tip and coat it with solute. The tip is then moved to where a feature is to be deposited and brought in contact with the surface. Condensation from the atmosphere (or from a humidified chamber) causes a moisture bridge to form between the AFM tip and the surface. Solute will then be transferred from the tip, through this moisture bridge, to the surface.

The primary advantage to this technique is the small size of the features it can make (10-100 nanometers), but this approach offers significant disadvantages when it comes to fabricating microarrays. The first disadvantage is throughput. The loading of solute on the outside of an AFM tip is a slow process, taking minutes or even hours. Because commercial AFMs use only one tip for scanning, this makes fabricating an array from thousands of different biomolecules prohibitively time-consuming. Chips are being developed to use in the AFM that have many tips in parallel, but that approach may not be enough to fully address throughput concerns. Another disadvantage is the limited area over which an AFM tip can travel. Originally intended for imaging nanoscale features on a small area of a surface, the precision positioning systems of current AFMs can move over areas of up to only a centimeter on a side. This is far smaller than the half-meter by half-meter travel range needed to produce a commercial batch of microarrays (10-100) at a time, as is currently done using existing systems. Along with the limited travel range, there are problems with alignment of the AFM tips when moving over large

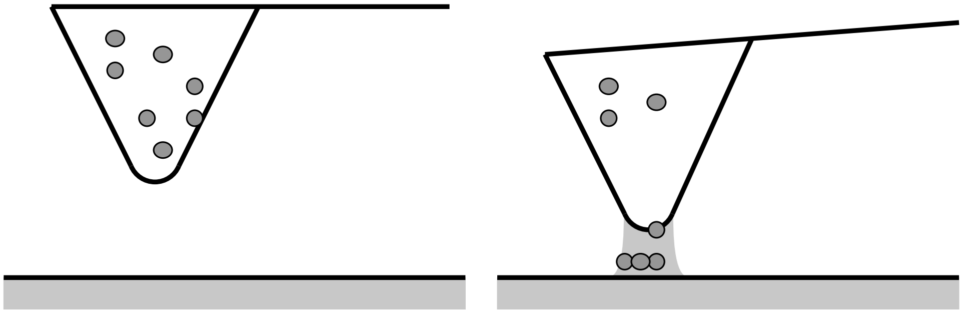

Figure 1.10: Schematic of dip-pen nanolithography. An atomic force microscope (AFM) tip is first coated with molecules to be deposited. When operated in a humid environment, water will bridge the gap between the surface and a tip brought close enough to it. The molecules on the outside of the AFM tip are then transferred to the surface.

distances. Finally, the features produced by dip-pen nanolithography are so small that they are below the detection ability of current microarray fluorescence scanners.

1.2.2 Non-microarray applications

While the work presented in this dissertation will focus on the application of fluid deposition and surface treatment technologies to improved biological microarray production, there are many applications outside of the microarray field, especially for fluid deposition technologies. In biology, picoliter-scale fluid dispensers are being used to perform high-throughput protein crystallization in order to determine protein structure using X-rays[29] as well as to deposit cells and polymer matrices in order to grow artificial tissues[30], among other applications.

Outside of the biological sciences, much interest has been generated by the printing of conducting polymers to form plastic microcircuits.[31, 32, 33] Conducting polymers combine the electrical properties of metals or semiconductors with the ease of processing and mechanical properties of standard polymers. These so-called π-conjugated polymers are long chains of monomers chosen such that these chains conduct electricity along their length. Certain conducting polymers can be dissolved in organic solvents in order to be loaded into a dispenser and deposited on a substrate. Using solutions of conducting, semiconducting, or insulating polymers all the elements of a microcircuit (wires and source, drain, and gate electrodes in a transistor) can be printed directly. Additionally, light-emitting diode (LED) structures can be made in this manner. Polymer microcircuits have the promise of being much less expensive to fabricate than silicon-based chips. Additionally, they can be used in applications that traditional microcircuits cannot, such as flexible circuits and displays.

To date, inkjets have been the primary fluid deposition technology used to print these polymer devices. As mentioned in the previous subsection, inkjets are very fast and well-adapted to depositing a limited number of different solutions, properties that match the needs of polymer microcircuitry fabrication. However, due to the fact that polymer solution inks are not as well tuned as the dye inks, the feature sizes produced by inkjets are larger than 50 μm. Also, continuous features such as lines or arcs must be made out of overlapping circular spots. These problems have required that photolithography be used to prepattern a surface with hydrophobic and hydrophilic regions before inkjet printing takes place, in order to define the sub-5-μm features required for functional microcircuits.[31]

1.3 Surface treatments

As hinted at in Section 1.1.1, a chemically treated surface is needed in order to attach presynthesized or extracted biomolecules to a substrate in order to form a microarray (except for in the photolithographic-synthesis-based approaches). The biomolecules attached in spots on the array need to stay there for the duration of an experiment, which usually requires that they remain throughout a couple of wash steps and the exposure of the surface to one or more target solutions. Ideally, these biomolecules would be attached so well that the number of them attached to the surface would not decrease appreciably even over multiple experiments using the same microarray. The best way to guarantee this is by chemically binding the biomolecules to the substrate. This chemical binding can be of several types, prime among those being electrostatic attachment and covalent attachment.

Electrostatic attachment relies on the attraction between a positively or negatively charged surface and biomolecules which possess the opposite charge. Usually the surface is positively charged since DNA with its negatively charged sugar-phosphate backbone and many proteins with their variously charged side-groups readily adhere to such a surface. Figure 1.11a shows an amine-terminated surface, the surface chemistry most often employed for electrostatic binding.[34, 35] These amine groups (-NH_2, but with an extra hydrogen and thus positively charged at neutral pH) can either be directly introduced to a surface or may come from the grafting of poly-*l*-lysine, an amine-containing polypeptide (protein), to a surface.

Surfaces for electrostatic attachment are relatively easy to create and use but they

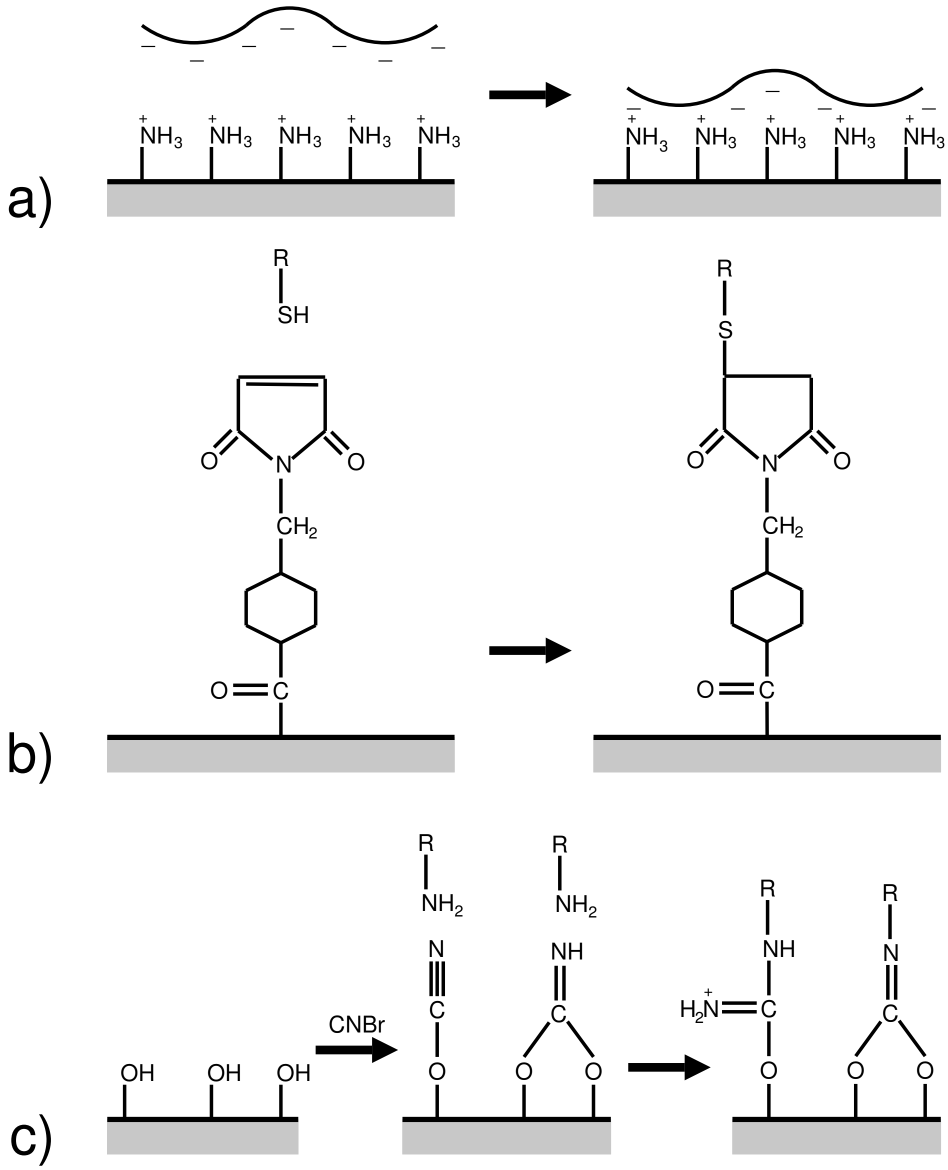

Figure 1.11: Commonly used surface chemistries for biomolecule attachment. The chemistries depicted are: a) amine, b) maleimide, and c) cyanogen bromide.

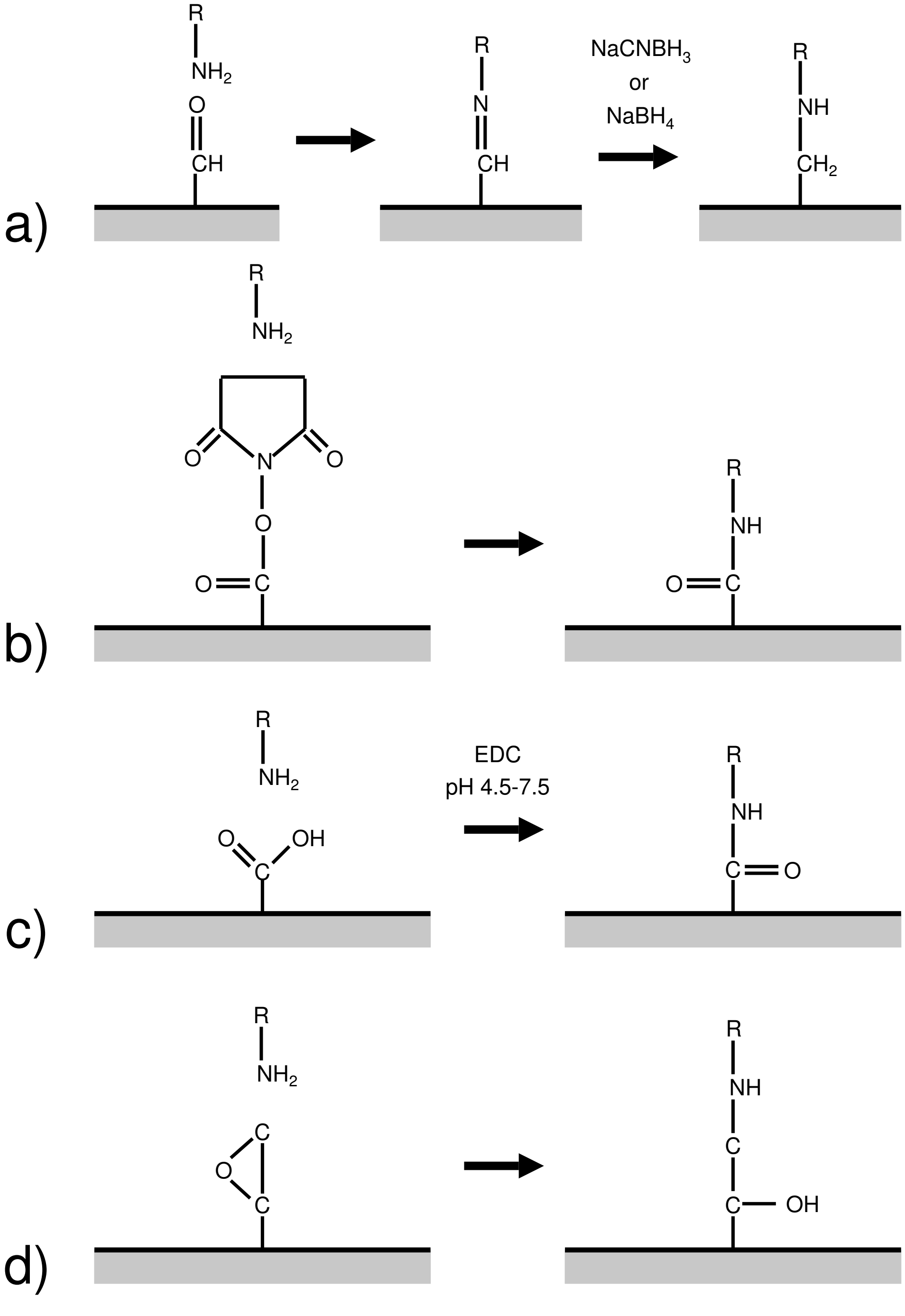

Figure 1.12: Additional commonly used surface chemistries for biomolecule attachment. The chemistries depicted are: a) aldehyde, b) NHS ester, c) carboxylic acid, and d) epoxide.

suffer from some disadvantages. Such surfaces can suffer from large amounts of nonspecific binding, where target biomolecules attach to the surface in the same fashion as probe biomolecules do. This leads to a decrease in sensitivity, as it becomes hard at low target molecule concentrations to distinguish between attachment of the target to a specific site on the microarray and the attachment of the target to open surface area. The primary problem with electrostatic attachment is the durability of that attachment. The probe biomolecules are not permanently attached to the surface, so a sizable percentage will be removed after a detection/wash cycle. This limits the number of uses for a surface using electrostatically attached biomolecules, often to a single use.

The technique most commonly used today to prevent probe biomolecules from leaving the surface over time is covalent attachment of the biomolecules to a substrate through wet chemical means. Typically a substrate is chemically treated so that functional groups are present on its surface that react with subsequently deposited biomolecules to form strong covalent linkages. After deposition of the biomolecules, any remaining reactive surface is blocked to prevent nonspecific interactions with target molecules. These covalent chemistries tend to react with and bind specific chemical groups that are present in different biomolecules. The most common functional group targeted by these chemistries is the amine group ($-NH_2$), present in nearly all proteins and easily introduced to oligonucleotides and many other molecules.

Another target is the sulfhydryl group (-SH), also known as the thiol group, which is present in certain proteins and can be added on to oligonucleotides. The sulfhydryl group is normally the target when attaching a protein via the amine group results in reduced activity. If a protein has a free sulfhydryl group (present in a cysteine amino acid residue), using it to bind the protein to the surface may result in greatly improved activity. However, it is an impractical functional group to use for binding oligonucleotides to the surface, as thiol-terminated oligos tend to form dithiol bonds (thiol groups on two separate oligos bind together) and it is a difficult process to cleave the dithiol bonds to yield lone sulfhydryl groups. One of the common chemistries used to bind sulfhydryl groups to a surface is by using maleimides. A particular chemical, sulfosuccinimidyl-4-(N-maleimidomethyl) cyclohexane-1-carboxylate (SSMCC), is commonly used as part of a multistep process in functionalizing gold, silicon, or diamond surfaces so that they can bind sulfhydryl groups.[36, 37, 38] This chemistry is shown in Figure 1.11b.

As stated above, amine groups are the most common targets for covalent linkages. One of the first chemistries to be developed for binding amine-containing biomolecules used cyanogen bromide (CNBr) to form cyanate esters (the left functional group formed in Figure 1.11c) on an alcohol-terminated (-OH) surface.[39] These cyanate esters react vigorously with amines to bind them to a surface. There are some drawbacks to this chemistry, however. Along with cyanate esters, cyclic imidocarbonates (the right functional group formed in Figure 1.11c) are formed as a side product of this reaction and they do not react nearly as well as the cyanate esters with the amines. The isourea bond formed with the surface via these two different functional groups is positively charged and so nonspecific binding of unwanted probe or target with the surface may result (in the same fashion as described for electrostatic attachment above). CNBr is a highly toxic reagent and must be handled with great care. Finally, a constant leakage of bound biomolecules off of the surface occurs with this chemistry.

Aldehydes are another amine-reactive functional group.[19] Figure 1.12a shows the chemistry by which aldehydes bind amine groups. Aldehydes react with amines to bind them to the surface, but the covalent bond formed is a Schiff's base, which is easily broken in even mildly basic conditions (such as the wash or hybridization buffers commonly used in microarray experiments). To convert this bond to a strong alkylamine bond, a process called reductive amination is used. A strong reducing agent, such as sodium borohydride ($NaBH_4$) or sodium cyanoborohydride ($NaCNBH_3$), is used to carry out this reaction. These agents are caustic ($NaBH_4$) or toxic ($NaCNBH_3$). In addition to requiring this extra linking step, a further disadvantage of the aldehyde chemistries is their relatively slow reaction rate with amine-containing biomolecules. Usually, several hours up to a couple of days are required for the aldehydes to fully react with deposited biomolecules. This reaction is also hindered by moisture and must be carried out under dry conditions ($<$10% humidity), making it not very compatible with protein microarrays using water-dependent proteins. However, aldehydes only react with primary amine groups and not secondary or tertiary amines, making them useful for binding amine-terminated oligonucleotides only at their tip, not at any of the secondary amines present in the nucleotide bases.

N-hydroxy succinimide (NHS) esters have also been used to bind amine-containing biomolecules.[40] In a reaction shown in Figure 1.12b, they react quickly with amine-containing biomolecules to form a strong covalent bond. NHS esters have been known in certain cases to retain higher activities of bound proteins and other biomolecules than the other amine-reactive chemistries listed here. However, these NHS esters also react with water, so the exposure to even atmospheric moisture can cause NHS-ester-terminated surfaces to lose their activity in a period of days or weeks.

Carboxylic acids are commonly used to bind proteins to surfaces.[41] By themselves, they are not amine-reactive, but using a solution of 1-ethyl-3-(3-dimethylaminopropyl) carbodiimide (EDC) and / or NHS, amine-containing biomolecules can be firmly attached. This chemistry is shown in Figure 1.12c.

Epoxide chemical groups, seen in Figure 1.12d, are among the most popular of the amine-binding functionalities.[42, 43] They react rapidly with amine groups and form stable covalent bonds to a surface in a reaction that takes on the order of 15 minutes to complete. No additional reagents or steps are necessary to complete the reaction. While epoxides will also react with water, they do so much less readily than the NHS esters, so storage in open laboratory conditions for prolonged periods (months) is possible. The primary drawback to these functional groups is the fact that they will react with and bind secondary amines (although much less vigorously than primary amines), so they will sometimes bind nucleotide bases in the middle of DNA strands. This can inhibit the hybridization necessary for proper microarray function, but this reduction in hybridization efficiency is not significant with short ($<$25 base) oligos.

As can be seen from this list, a large number of different chemical treatments have been developed to bind biomolecules to surfaces. This list is by no means exhaustive, but these are among the most commonly used types of surface chemistries currently employed. Almost all of the processes developed to attach the chemical groups listed rely on traditional liquid-phase (wet) chemistry. As an example of a typical wet-chemical functionalization process, Table 1.2 lists the steps in a process for introducing epoxide functionalities onto glass slides. As can be seen, hazardous (H_2SO_4 and H_2O_2) and relatively expensive (3-glycidoxypropyltrimethoxysilane) reagents are used in large enough quantities to bathe the slides being treated. Also, over 11 hours are required to treat a single batch of slides. These factors contribute to the relatively high cost of chemically treated substrates used in microarray fabrication.

Step	**Time required**
1. Immerse glass slides in ”piranha” solution (70% H_2SO_4, 30% H_2O_2)	$>$8 hours
2. Wash slides with distilled water to remove piranha solution	N/A
3. Rinse slides with 95% ethanol and dry with nitrogen	N/A
4. Treat slides with a solution of 95% ethanol, 16 mM acetic acid, and 1% 3-glycidoxypropyltrimethoxysilane	1 hour
5. Rinse slides with 95% ethanol and dry with nitrogen	N/A
6. Cure slides by baking in an oven at 150°C	$>$2 hours
7. Rinse slides with 95% ethanol and dry with nitrogen	N/A

Table 1.2: Steps in the epoxide functionalization of glass slides using a wet-chemical process.[20]

1.4 Cold plasmas

The cost, environmental impact, low throughput, and complexity of traditional wet chemical treatments have caused researchers to search out other means of functionalizing surfaces. Recently, the use of cold plasmas to chemically modify surfaces has been the subject of much research because of the potential ability of plasma-based treatments to address all of the listed problems with wet chemistry.

Plasmas are referred to as the fourth state of matter (in addition to the more common solids, liquids, and gases). A significant fraction of the particles within a plasma are ionized. These ionized particles can interact over long distances, in addition to the elastic and inelastic particle collisions experienced in gases and liquids. External electrical fields can shape and direct a plasma by imparting force to the charged particles within it. Molecules within the plasma can be excited, ionized, or broken apart by the collisions between high-energy electrons and nuclei. This leads to the formation of molecular fragments, ionized species, or free radicals that would not be seen in traditional chemical reactions.[44]

Two primary types of plasmas exist: hot and cold. Hot plasmas are ones where extremely high temperatures ($>$ 6000 K) cause electrons and nuclei to dissociate. Such plasmas are in local thermodynamic equilibrium and can be found in electric arcs, plasma jets, or in and around stars. Cold plasmas are created by the application of a strong electrical field that pulls the electrons off of atoms within the field. While the electrons will have significant energy and therefore be at a high temperature, the heavy particles (nuclei and uncharged atoms and molecules) have a much lower temperature. The overall temperature of the plasma can be only slightly higher than ambient temperature, thus the ”cold” in the name. This type of plasma is not in local thermodynamic equilibrium (cold plasmas are also known as non-equilibrium plasmas) and is much easier to generate and handle than hot plasmas.

As stated above, a strong electrical field must be applied to a gas in order to generate a cold plasma. Commonly, the plasma is generated inside of a chamber (an example of which is shown in Figure 1.13a) held at a pressure significantly lower than one atmosphere (760 Torr). A precursor gas is flowed through this chamber and a power supply drives a set of electrodes to produce a large enough electrical field to ionize the gas and form a plasma. Either a direct current (DC) or alternating current (AC) can be used to generate a plasma, but DC plasmas require very high voltages and electrodes that must be in direct contact with the plasma. For these reasons, radio-frequency (RF) AC discharges are more widely used to generate cold plasmas. Of the many different electrode configurations used in plasma chambers, the one most often seen in industrial systems consists of two parallel plates, one on the top and one on the bottom of the chamber, between which flows the precursor gas used in the plasma. One electrode, referred to

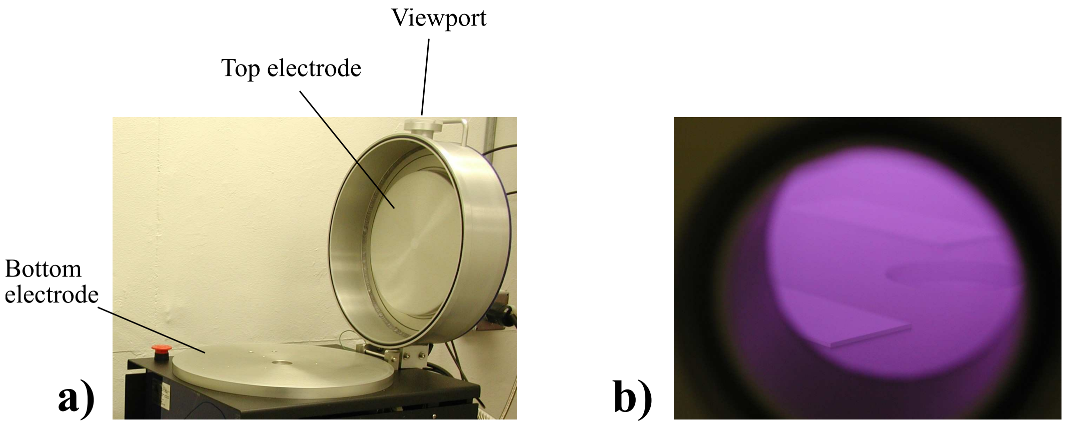

Figure 1.13: An example of a plasma chamber. The particular chamber shown in a) is a parallel-plate plasma etcher. An image taken through the viewport of this chamber during the generation of an argon plasma is shown in b).

as the ”active” electrode, is connected to a power supply, while the other is electrically grounded. Solid substrates that are to be exposed to the plasma are either placed directly on the lower electrode or are suspended a short distance off of it. If the lower electrode is active, the system is referred to as a reactive ion etcher (RIE) because positive ions from the plasma are strongly pulled to the substrates, causing anisotropic etching by ionic bombardment of exposed portions of their surfaces. If the upper electrode is active, the system is called a plasma etcher. The surfaces of substrates loaded in such a system experience a gentler, more isotropic exposure to ions from the plasma.

From the names of the two described plasma chamber configurations, it can be seen that one of the primary applications that have emerged for cold plasmas is in the etching of surfaces. Much of the modern semiconductor industry relies on cold plasmas to etch the small features present in microcircuitry. Another vital application of cold plasmas in semiconductor fabrication is plasma-enhanced chemical vapor deposition (PECVD), where the plasma assists in the process of building a film of material on a solid substrate from a vapor feedstock.

In addition to etching or film deposition, cold plasmas have the ability to modify the chemical properties of a surface. The first way they can do this is by cleaning adsorbed organic contaminants off of the surface of a substrate. Any solid material exposed to atmosphere builds up a thin layer of carbon-containing molecules on its surface. By exposing such a surface to a cold plasma of an gas (or a series of gases) such as oxygen or argon, these contaminants can be oxidized and ablated away. The oxidation and removal process is also referred to as ashing, and is widely used in the semiconductor industry to strip away layers of photoresist.[45] Atmospheric-pressure plasma cleans can also be used to sterilize surfaces in a rapid, inexpensive manner.[46]

A clean surface is vital for subsequent chemical modifications, which can also be performed using cold plasmas. Much research has been performed on the use of cold plasmas to modify the wetting and adhesion properties of polymers.[47, 48, 49, 50] By converting functional groups on the surface of hydrophobic polymers, such as polydimethylsiloxane (PDMS) or polytetrafluoroethylene (PTFE, Teflon), to hydrophilic functionalities, like hydroxyls or carboxylic acids, these polymers can be more readily integrated into microfluidic devices and can be adhered to other surfaces. Other simple chemistries, such as amines, have been formed on surfaces using plasmas, but there are problems with attaching more complex functionalities directly through a plasma.[51] When a precursor gas composed of a larger molecular species is converted to a plasma, the molecules within the gas are fragmented in several places. A mixture of functionalities on a treated surface results, rather than a single desired chemical moiety.[52] While this is a problem for direct functionalization of a surface, it has been used to plasma-polymerize films on various surfaces using monomer-containing precursor gases. [53]

Cold plasma reactions only consume gases and vapors, which means that they have the potential to use far lower amounts of reagents than equivalent wet chemical processes. As an example, the hazardous and relatively expensive sulfuric acid and hydrogen peroxide bath used in the process of Table 1.2 can be replaced by a 2-minute-long exposure to an oxygen plasma, followed by a 2-minute-long exposure to an argon plasma. The reduction in quantity of reagents consumed, and the containment of any hazardous chemicals within a sealed reactor at below atmospheric pressure, means that cold-plasma-based processes can be significantly less expensive, much safer, and more environmentally friendly than traditional wet chemistry. Additionally, the entire treatment can be automated and, due to the unique chemistries present in a plasma, may take much less time than a wet chemical equivalent.

1.5 Structure of the dissertation

The majority of this dissertation will focus on the development, application, and characterization of a fluid microplotter, but work involving plasma-based surface treatments will also be described. Chapter 2, originally published in the Review of Scientific Instruments[54], will introduce the fluid microplotter technology. This will be followed by Chapter 3, where the performance of the microplotter will be evaluated for tasks relevant to the production of microarrays. Chapter 4 will contain an analysis of the mechanism by which applied ultrasonics causes spraying or pumping from the microplotter dispenser. Chapter 5, a paper in press in Biosensors and Bioelectronics, will describe a plasma-based treatment process for the introduction of epoxide groups onto an oxide surface.[55] Finally, Chapter 6 will demonstrate applications of the microplotter and plasma treatment technologies. Appendix A is provided to describe the lattice gas fluid modeling program that we have developed to simulate fluid flow within the microplotter dispenser, and it contains the source code for this program.

Chapter 2

Controlled deposition of picoliter amounts of fluid using an ultrasonically driven micropipette

A fluid microplotter that uses ultrasonics to deposit small fluid features has been constructed. It consists of a dispenser, composed of a micropipette fastened to a piece of PZT piezoelectric, attached to a precision positioning system. When an electrical signal of the appropriate frequency and voltage is applied, solution in the tip of the micropipette wicks to the surface in a controlled fashion. The gentle pumping of fluid to the surface occurs when the micropipette is driven at frequencies in the range of 400-700 kHz. Spots with diameters smaller than several microns can be deposited in this manner. Continuous lines can also be produced. Several examples of deposited patterns and structures are described. This means of deposition represents a higher-resolution alternative to standard fluid deposition techniques in the fabrication of biological microarrays or polymer-based circuits.

2.1 Introduction

The dispensing of picoliter volumes of fluids onto precise locations on a surface is becoming increasingly important in a variety of venues. Deposited features may need to be continuous (lines or arcs), discrete (spots), or even three-dimensional (towers). A smaller size of these features is beneficial in a wide variety of applications, ranging from conductive-polymer-based electronic circuits to biological microarrays.

On one extreme, solution printing of polymer-based circuits shows potential as an inexpensive means of producing flexible electronics. Much effort has been directed toward adapting inkjet printers to deposit entire circuits. The end goal is to create a system in which circuit patterns could be defined in software and then rapidly printed onto a substrate. Inkjet printers have attracted interest for this application because they are inexpensive and deposit fluid rapidly. The production of microcircuits, however, requires the ability to draw continuous lines of a very small width. Inkjets, by their nature, cannot produce continuous lines but instead form lines from overlapping spots. They are currently limited to making features no smaller than $\sim$50 μm wide. To overcome this limitation, a photolithographic patterning step has been required to form dense microcircuits, significantly limiting the ability to custom-design circuits.[31]

On the opposite extreme, the fabrication of dense biological microarrays relies on the ability to attach many different biological molecules to a surface in a small area. There are three principal processes used to create biological microarrays: light-directed synthesis of short DNA strands on a surface, chemical synthesis of DNA on a surface, and the spotting and attachment of prepared biomolecules to a surface.

In the first technique, photolithographic methods familiar from the semiconductor industry are used to synthesize, base by base, short oligonucleotide strands with a specific sequence in each feature on a surface.[10, 11, 12] This light-directed *in situ* DNA synthesis can produce tightly packed small features ($\sim$ 20 μm wide) but is limited just to the production of DNA microarrays. Although some work has been done in patterning short polypeptides in this manner[14], practical protein or other biomolecule-based microarrays may never be created in this fashion because of length limitations and protein folding problems.

A similar chemical-based *in situ* DNA synthesis technique uses inkjets to build short strands of DNA on a surface by selectively depositing the appropriate chemicals to build specific DNA sequences at specific spots.[4, 26] The custom creation of DNA microarrays using inkjets is simpler than light-directed synthesis. Significantly less complex equipment is required as well. Arrays produced using inkjets are not as dense ($\sim$ 200 μm center-to-center spacing on spots) as those produced by the photolithographic techniques. Again, this approach is limited to only short strands of DNA.

The third technique differs from the other two in that previously prepared biological solutions are deposited. Finely machined quill pins are dipped in a biological solution and then tapped on a surface at a precise location to deposit a spot.[23] If this surface is chemically activated in a particular way, the biomolecules in solution will bind to the surface within the spot. A microarray can thus be fabricated by spotting multiple different biological solutions across the surface. The greatest advantage of this technique is that it can be used with a wide variety of prepared biological solutions, including proteins and carbohydrates.[18, 19, 56] These solutions can be very limited and expensive, so minimizing the use of solution per microarray produced is desirable. Existing quill pins for spotters are limited to producing spots roughly 75 μm in diameter. On average, 1 nL of solution is deposited per spot using existing systems.

We have developed a fluid microplotter that can deposit discrete features that are an order of magnitude smaller than those listed above, using many times less solution.[57]

In addition, it can easily draw continuous lines and arcs as well as deposit high-aspect-ratio towers. The microplotter is able to handle a wide range of fluid viscosities. This paper descibes the operation and use of this device.

2.2 Fluid microplotter design

The microplotter consists of a dispenser that is positioned over a substrate using a high-precision positioning system. This positioning system is controlled by custom software on a desktop PC that permits complex, synchronized movements and deposition patterns. The deposition process can be monitored in real time using an attached charge-coupled-device (CCD) camera. A schematic overview of the microplotter is shown in Fig. 2.1.

The dispenser head is composed of a micropipette attached to a piece of lead zirconate titanate (PZT) piezoelectric. When driven by an AC current, the PZT causes the micropipette to vibrate primarily along its axis. Under appropriate conditions (a driving frequency in the range 400-700 kHz, 1-4 V peak-to-peak driving voltage, tip IDs ranging from 1 μm - 100 μm), applying the ultrasonic field causes fluid in the tip to wick out onto the surface. Smaller micropipette IDs yield smaller spots. For a given micropipette, the size of the resulting fluid feature can be controlled by raising or lowering the amplitude of the peak-to-peak AC voltage. As we will show later, the viscosity of the deposited solution also affects spot size.

This design is fundamentally different from that of inkjet printers or other piezoelectric-based fluid dispensers. In those designs, the expansion of a piezoelectric element exerts force on fluid contained within a narrow channel, causing fluid to be ejected onto a

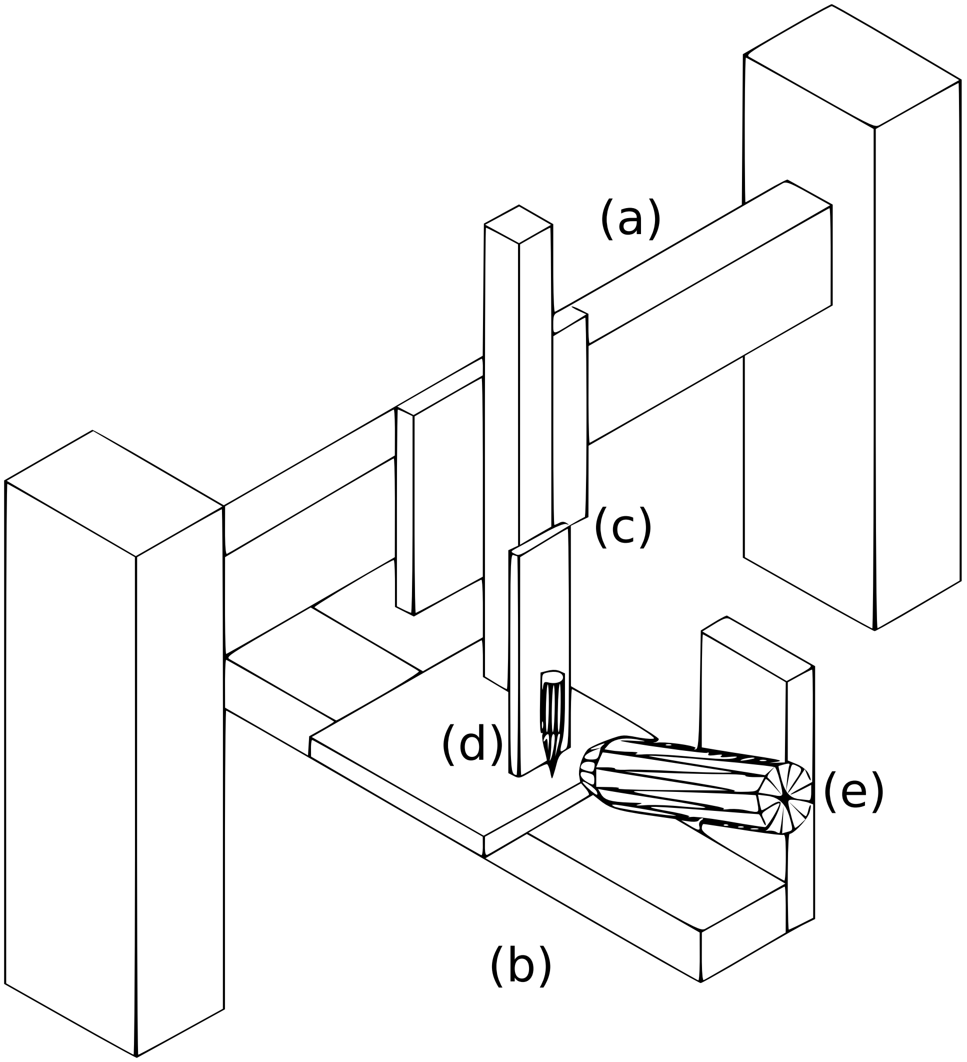

Figure 2.1: A schematic diagram of the fluid microplotter. It is composed of a precision positioning system with (a) X, (b) Y, and (c) Z positioning stages in the configuration shown and (d) a dispenser head composed of a piece of PZT piezoelectric attached to a micropipette, not shown to scale. The fluid deposition process is monitored by (e) a CCD camera.

surface. Other devices that use micropipettes use cylindrical piezoelectric elements surrounding the micropipette to squeeze fluid out of the micropipette tip and thus operate on the same principle of a pressure pulse.[58, 59] Our dispenser is much simpler and the principle of operation is different, leading to much better performance. As will be described in more detail later, ultrasonic vibrations cause fluid to be ejected from the tip of the micropipette, not pressure due to the contraction of the capillary tube, as in these other designs.

To achieve the deposition of fluid features at precise locations we use three positioning stages. Two stages move the dispenser head to give X-Z positioning, while a third stage moves the substrate in Y. These stages can position the substrate and micropipette tip relative to each other over a 50 cm x 50 cm x 10 cm volume, with a rated maximum lateral positioning error of 20 μm. By tuning the proportional-integral-derivative (PID) control loop used for positioning feedback and operating over a smaller portion of this volume (e.g. 5 cm x 5 cm x 2 cm) we can achieve much more accurate lateral positioning (error of < 5 μm). A motor controller directs the motion of these stages and it, in turn, is controlled by software on a desktop PC. This software allows for completely automated deposition of complex patterns and shapes on a substrate using multiple feed fluids.

2.3 Deposition process

The steps in depositing fluid onto a surface using the fluid microplotter are: loading the dispenser with fluid, positioning the dispenser over the starting coordinate for the first feature, drawing the first feature, repeating until all features of the current fluid are drawn, discharging any remaining fluid, washing the dispenser tip, and if necessary, loading the dispenser with the next fluid.

The loading of the micropipette with fluid can be accomplished by either attaching a supply of fluid to the wide end of the micropipette or by dipping the sharp end of the micropipette into a fluid well. A continuous fluid supply is useful for a large amount of deposition using a single fluid, whereas dipping the micropipette tip into fluid is useful for depositing multiple fluids (dip-pen mode). We concentrated on the use of the device in a dip-pen mode, as the ability to take up very small fluid volumes is useful when expensive or limited-quantity fluids are used, something that is often the case in the fabrication of biological microarrays. When the sharp tip of the micropipette is placed into a fluid well, capillary action causes fluid to wick up into the tip. This process can be controlled so as to allow sub-microliter control of the loaded fluid volume by controlling the time spent in the fluid well. A micropipette with a 1 μm inside diameter at the tip can take up a dilute aqueous solution at a controlled rate of $\sim$ 5 nanoliters per second.

To deposit a spot, the dispenser tip is lowered to the substrate at the desired X,Y position, a pulse at the correct frequency and peak-to-peak voltage is sent to the PZT for a certain duration, and the dispenser tip is raised. The same frequency pulse is used for all solutions deposited using a particular dispenser. If the feature is continuous, such as a line or arc, the tip is lowered until the fluid makes contact with the substrate. A continuous AC signal is supplied, the tip is backed off the surface slightly while maintaining fluid contact, and the continuous feature is drawn by coordinating the movement of all three stages. The AC signal is then shut off and the tip is raised. Satellite droplets, a concern for inkjets, are not a problem for the microplotter because the only fluid deposited is from contact of the fluid at the dispenser tip with the surface. The gentle nature of the interaction of the micropipette tip with the surface means that thousands of spots can be deposited by a dispenser without degradation of its tip.

Accurate Z-positioning is important in order to guarantee the deposition of regularly sized and shaped fluid features. If the dispenser is positioned too high above the surface when attempting to deposit a feature, fluid contact may not be made and no feature will be deposited. Software that calculates the correct surface height at each X,Y position is the primary means of precise Z-positioning control.

In the dip-pen mode, when all features for a particular solution have been drawn, the fluid remaining in the dispenser can be salvaged by returning the dispenser to rest over the original fluid well. An AC pulse of a higher peak-to-peak voltage than that used to deposit spots is sent to the PZT, causing the micropipette to spray any remaining solution back into the original well. Return of fluid is generally not necessary because of the precise loading characteristics of the device, which allow the fluid volume in the dispenser to be preadjusted to create the required number of features with no waste. The micropipette tip can be cleaned by loading a solvent, such as deionized water, and spraying it back out. This loading and spraying operation is repeated several times to reduce the possibility of cross-contamination.

Clogging by dust and other particles is a general problem with micropipettes that have less than 5 μm inside diameters at the tip. In our system clogging is overcome by two means. We can unclog a tip simply by using a medium-voltage AC pulse to dislodge the clogging particles. When the micropipette tip is in a solvent well, often the uptake of solvent from the sharp end of the micropipette will force the clogging particle loose and carry it away from the tip. When combining these two methods, clogs can be quickly removed, allowing operation even with micropipette inside diameters smaller than 1 μm.

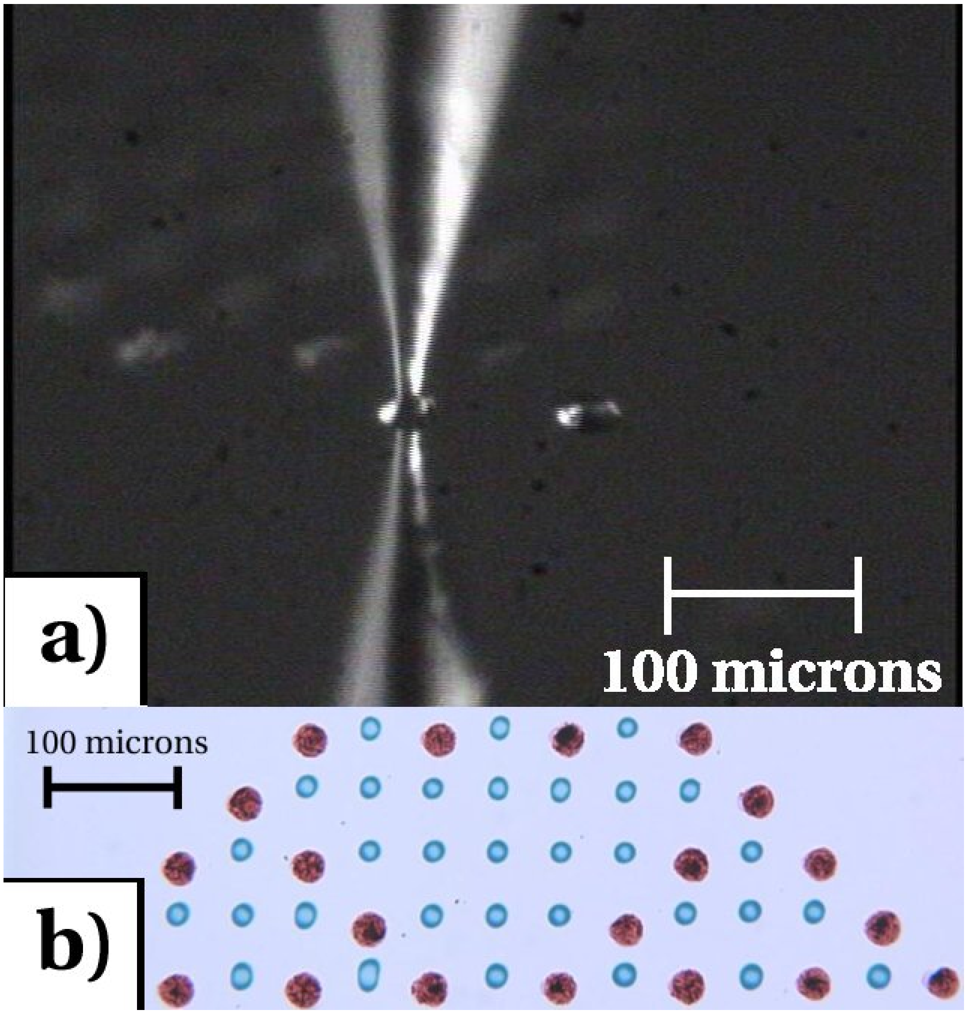

Figure 2.2: Microplotter performance. a) A closeup image of the micropipette tip in the process of depositing a pattern of spots. Several spots, spaced ∼100 μm apart, are shown. b) A pattern made from red and blue food coloring deposited on an untreated glass slide at the same dispenser voltage and frequency. The red spots are 28 μm in diameter, the blue spots are 20 μm in diameter, and the center-to-center spacing of the spots is 50 μm. The spot size differences are caused by the differing viscosities of the two solutions.

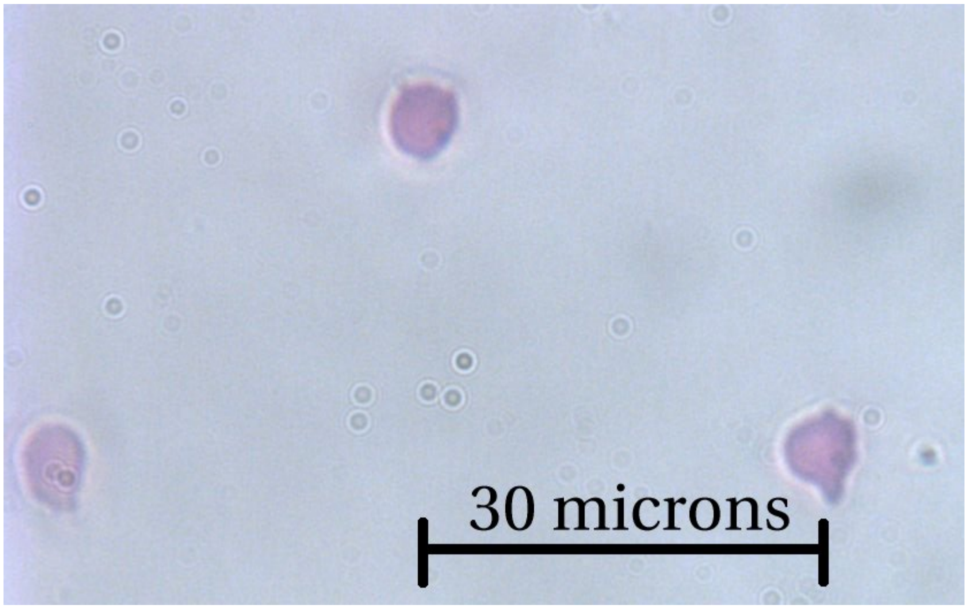

Figure 2.3: An example of small feature sizes: a triangular pattern of 6 μm diameter spots in red food coloring on an untreated glass slide. Even dilute, low-viscosity aqueous solutions deposited on a hydrophilic substrate can be controlled to give uniform spots of a small size.

2.4 Performance tests

We tested the capabilities of this system for small-volume fluid deposition in several ways. Initial testing was carried out using aqueous solutions of food coloring to allow for easy analysis of fluid deposition runs using standard optical microscopes. Figure 2.2 shows the micropipette tip in the process of depositing a spot as well as a pattern made from two solutions with quite different viscosities, red and blue food coloring. The red food coloring viscosity is close to that of water, while the blue food coloring is significantly more viscous. To demonstrate the effect of viscosity on deposited spots, the applied frequency and voltage were held constant for the two solutions. Different spot sizes resulted (28 μm diameter for the red spots, 20 μm diameter for the blue spots). Figure 2.3 shows that very small spots can be made, even with dilute aqueous solutions on a hydrophilic substrate. In both of these tests, the glass slides were not treated or cleaned.

For biological applications, small spots are desired because of the significantly lower volumes of solution deposited. To estimate how much solution is used per spot deposited by the microplotter, the dispenser head was loaded with a known volume of solution. The microplotter then was commanded to deposit as many spots as it could before it ran out of solution. By counting the resulting number of spots, it was determined that the microplotter uses approximately 16 pL per spot for 50 μm diameter spots and 3 pL per spot for 20 μm diameter spots. These values are lower than what is expected by modeling each spot as a hemisphere, but, as can be seen in Figure 2.2a, the deposited spots are not full hemispheres, so these values are reasonable.

The ability to draw continuous features, such as lines or arcs, is a unique capability

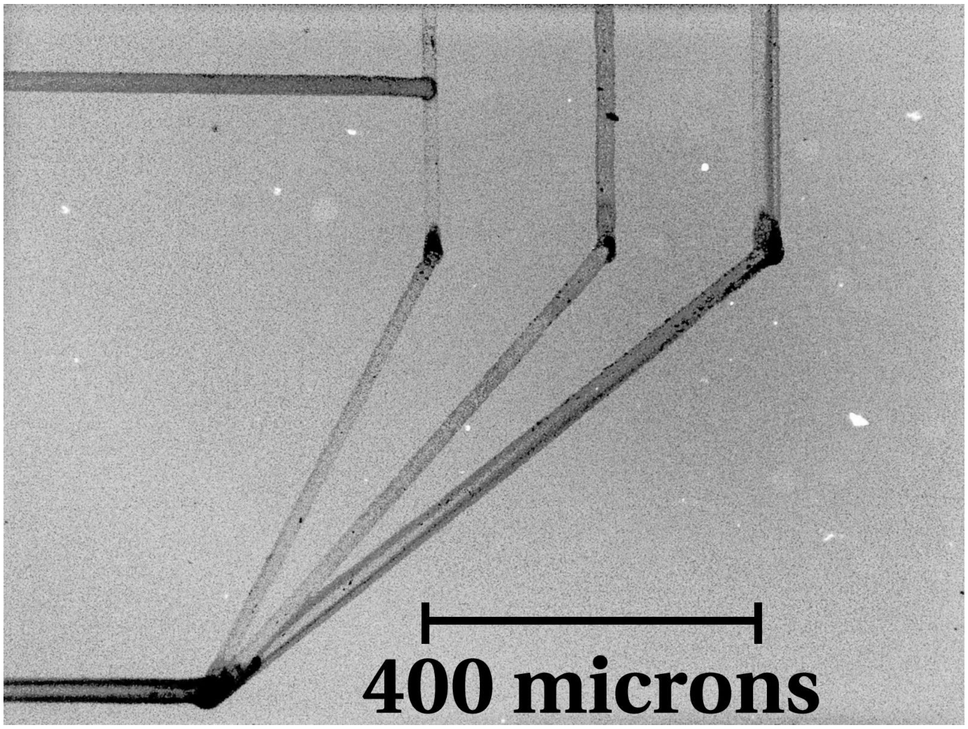

Figure 2.4: An example of continuous-feature deposition. A pattern of lines was deposited in blue food coloring onto an untreated glass slide. The lines are approximately 25 μm wide.

of the microplotter. To demonstrate this capability, a computer-aided design (CAD) file containing a pattern was loaded into the controlling software of the device and the pattern was then deposited on a glass slide using blue food coloring. A result is shown in Figure 2.4.

We have also tested this device with other solutions. One such test involved depositing DNA onto a glass slide to demonstrate the ability to fabricate biological microarrays. A solution containing double-stranded salmon sperm DNA at a concentration of 500 μg/mL was loaded into the dispenser and deposited in a pattern on a poly-*l*-lysine coated slide. That slide was then washed with DI water and ethanol, stained with an

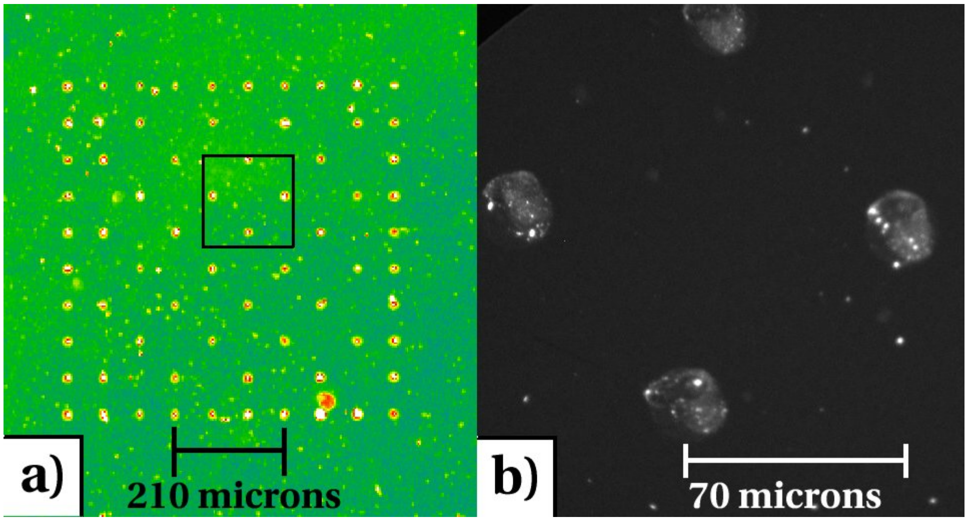

Figure 2.5: Image of a pattern deposited using a 500 μg/mL solution of double-stranded DNA, which was then fluorescently stained with an intercalating nucleic-acid dye. The specks of fluorescence surrounding the pattern are due to dust on the glass surface. a) The overall pattern was scanned using a commercial microarray scanner. b) To examine the spots, a laser confocal microscope was used. The spots are $\sim$25 μm in diameter.

intercalating dye that fluoresces at 488 nm, and washed again. The slide was scanned using a commercial microarray scanner. Figure 2.5a shows that the DNA was electrostatically attached to the surface in the specified pattern. The size of the deposited spots was measured in a scanning laser confocal microscope. The spots, shown in Figure 2.5b, are $\sim$25μm in diameter. The fluorescent specks in both images away from the pattern are not due to errant droplets from the plotter, but are believed to be due to dust particles on the slide surface.

Earlier, we showed continuous lines deposited in food coloring. As a demonstration of the small line widths the microplotter can draw, a dilute gold nanoparticle solution was deposited on a silicon wafer, yielding lines less than 5 μm wide. Figure 2.6 shows

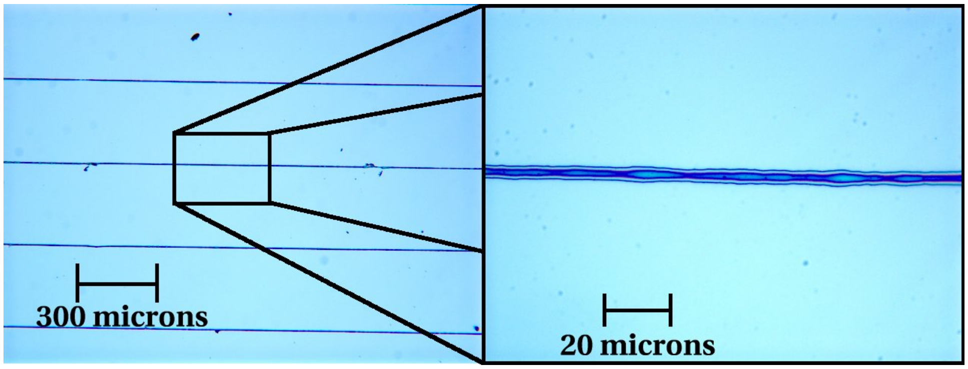

Figure 2.6: Lines of a dilute gold nanoparticle solution deposited on a silicon wafer. The lines are 5 μm wide and are continuous, except where broken by large particles present on the wafer.

that these lines are long and continuous, broken only where large particles were present (the wafer was not cleaned before fluid deposition).

Three-dimensional features can also be drawn using the microplotter, another unique capability. Figure 2.7 shows an example, a tower $\sim$30 μm wide and $\sim$370 μm tall deposited using a colloidal silver suspension. To create the tower, the micropipette, loaded with colloidal silver suspended in methanol, was brought in contact with the substrate (a silicon wafer) and ultrasonic vibrations were initiated. As the methanol evaporated at the micropipette tip, the suspended silver concentrated near the surface. By gradually drawing the dispenser away from the surface, silver was deposited in the shape of a tower.

2.5 Discussion

The process by which longitudinal ultrasonic vibrations cause fluid to wick out of a micropipette tip in a controllable manner is not understood. Ultrasonics has been used

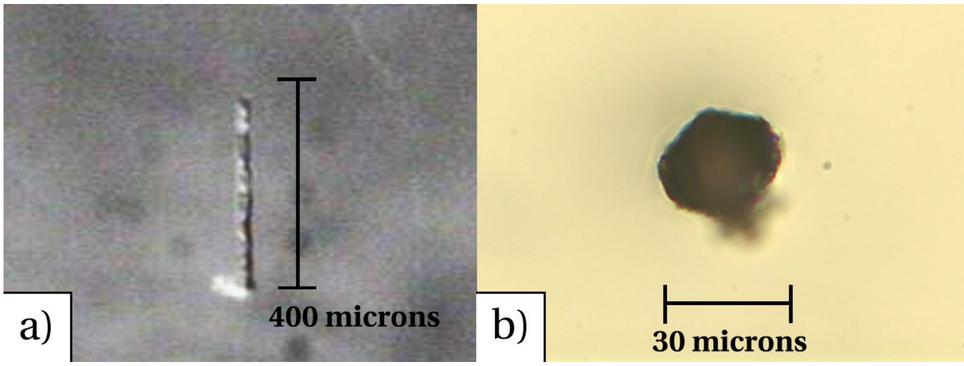

Figure 2.7: A tower of silver particles deposited from a methanol-based colloidal silver suspension. a) The height of the tower, $\sim$370 μm, can be seen in an image captured using the microplotter's attached CCD camera. b) A top view image of the tower was taken using a standard optical microscope to determine its width ($\sim$30μm).

in travelling-flexural-wave pumps for microfluidic systems.[60] Additionally, ultrasonically vibrating micromachined silicon needles have been explored for possible use as pumps.[61] These pump designs are more complex than our micropipette-based system and operate at different conditions. The fluid spraying mode of our micropipette tip appears to be similar to that reported previously[61, 62], but there have been no reports of controlled fluid ejection onto a surface as we show here and it seems unlikely that the same mechanisms are at play.

An apparent change in wetting of fluid on the micropipette occurs when it is ultrasonically driven. Fluid rises up the outside walls of the micropipette when it is placed in a fluid well and made to resonate. Figure 2.8 shows the difference in wetting behavior for a resting and an ultrasonically driven micropipette. The micropipette tip was intentionally sealed to demonstrate that this phenomenon is not caused by pumping from within the micropipette. Because we currently cannot predict the resonant frequency

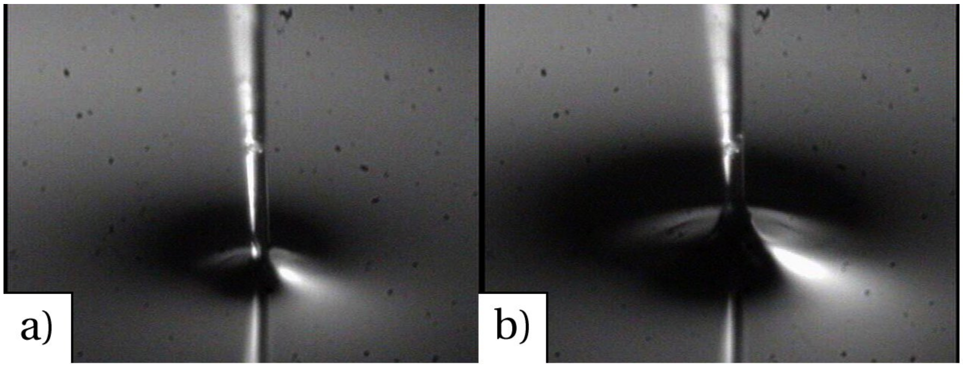

Figure 2.8: Apparent change in wetting behavior caused by an applied ultrasonic pulse. We placed a glass micropipette with a sealed tip in water and compared wetting behavior with a) no signal on the piezoelectric and b) a signal at the resonant frequency of the dispenser. At the resonant frequency, water rises up the side of the glass micropipette.

of a given dispenser, this change in wetting behavior is used to determine the resonant condition.

Several factors influence the sizes of fluid features produced by the fluid microplotter. The first is the size of the opening at the tip of the dispenser. The images we show illustrate fluid features produced by a variety of tip inside diameters. The smallest features (Fig. 2.3) were produced using an inside diameter of 1 μm, smaller than those of dispensers used for the other images. Second, fluid viscosity affects feature size. The more viscous the fluid, the less likely it is to spread from the dispenser tip. Third, the interaction of the surface with dispensed fluid affects feature size. If an aqueous solution is being dispensed onto a hydrophobic substrate, feature size is reduced. By adjusting these factors, it should be possible to create features smaller than 5 μm wide.

2.6 Acknowledgements

Funding for this research was provided by the University of Wisconsin-Madison's Office of University-Industry Relations. We thank Professor Dan van der Weide for the use of his confocal microscope.

Chapter 3

Theoretical and experimental studies of microplotter operation

In Chapter 2 we described a microplotter that is capable of depositing fluid in spots that are at least an order of magnitude smaller than those produced using existing technologies and we demonstrated proof-of-principle applications of the microplotter in the field of biological microarrays. The small spot size is generally due to the extremely small volume of fluid pumped to the surface to create a spot. However, the diameter of the spot varies with respect to several other parameters. First among these is the degree to which a fluid wets the surface it was deposited on. A higher degree of wetting leads to a larger spot for a constant deposited volume. After this is the role that viscosity plays in determining spot size. We have observed that a greater viscosity can cause both larger spots and smaller spots, depending upon the circumstances of fluid deposition. Finally, the volume of the spot itself may change with the settings of the microplotter dispenser. In this chapter, we present an analytical expression for deposited spot diameter as a function of surface wettability, as well as experimental studies on the effect of all the listed parameters on spot size.

Separate from the studies of spot size, we have also investigated the role that evaporation plays in the deposition of fluid at this scale. Due to the small fluid volumes contained within a needle and in a deposited spot, evaporation of solvent takes place rapidly. This can lead to a change in concentration for solution within a needle during the deposition of a series of spots or may cause a buildup of material as a needle is held over a spot for an extended period of time. The evaporation of spots is also important when dealing with proteins, as certain proteins lose their activity when dried out. These proteins need to be deposited under conditions that prevent evaporation. We have developed relationships for the rate of evaporation within the dispenser needle, at the needle tip, and from a spot on a surface.

3.1 Spot diameter as a function of surface wetting

3.1.1 Theory

As mentioned before, the way a fluid wets a surface plays a large role in the final size of a spot deposited on that surface. To explore the reasons behind this, we first start with the fundamentals of surface wetting. When a droplet is placed on a solid surface that is surrounded by a gas, as seen in Figure 3.1, three interfaces form. These three interfaces, liquid-gas, solid-gas, and solid-liquid, have three different values for the free energy of their surfaces: γ_{LG}, γ_{SG}, and γ_{SL}, respectively. Thomas Young assumed that these free energies could be represented by surface tensions and that at equilibrium they must be balanced. By setting the sum of the horizontal components equal to zero, he arrived at what's become known as Young's Equation[63]:

$$\gamma_{LG}\cos\theta = \gamma_{SG} - \gamma_{SL} \tag{3.1}$$

where θ is referred to as the contact angle. The contact angle is the interior angle that the edge of a droplet makes with a surface. The larger the contact angle, the less a fluid likes to wet a surface. Figure 3.2 shows droplets of distilled water placed on three

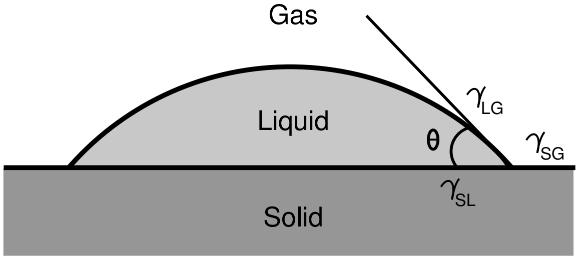

Figure 3.1: A cross-section of a liquid droplet on a solid substrate.

different materials: electrical grade Teflon, a silicon wafer (with native oxide present), and a cleaned glass microscope slide. As can be seen from the figure, water does not wet Teflon well, but it does wet glass, and the corresponding contact angles are significantly different.

Another observation that can be made from Figure 3.2 is that the droplets all have the shape of a portion of a sphere, also referred to as a spherical cap. A sphere has the least surface area per volume of any shape, so it is the shape with the lowest free energy for a free-floating liquid where $\gamma_{LG} > 0$. When liquid is on a surface, the interface energy interactions in Equation 3.1 determine the angle at which the edge of a droplet intersects a solid surface. However, the rest of the droplet tries to assume the shape of the remaining portion of the sphere. Figure 3.3a illustrates this more clearly, as a droplet on a surface can be seen as a sliced-away section of a larger sphere.

We are therefore able to use geometry to create a relationship between contact angle, volume, and diameter of contact area for a given droplet of fluid on a solid substrate. The first step is the calculation of the volume of the spherical cap. A sphere is composed

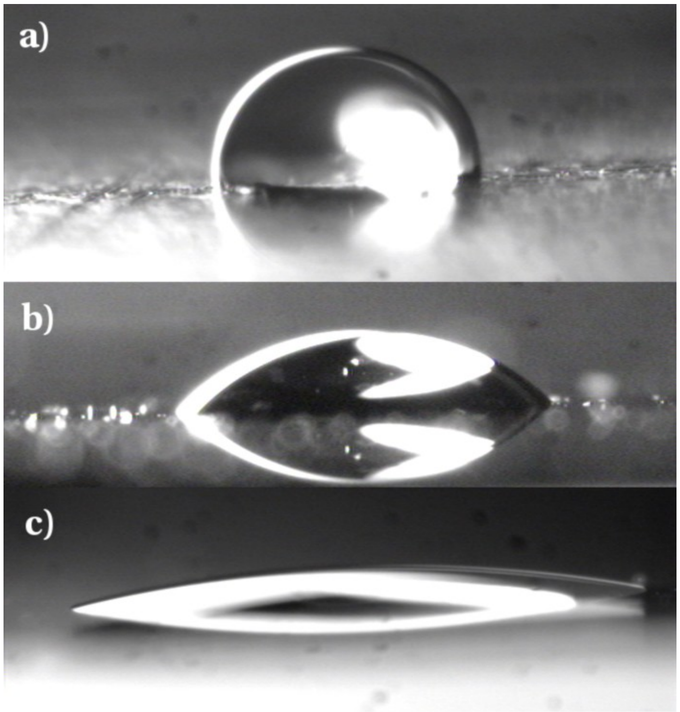

Figure 3.2: Wetting of three different materials by water. One microliter droplets of deionized water were deposited on three different substrate materials: a) a piece of electrical grade Teflon, b) a silicon wafer (with native oxide), and c) a cleaned glass microscope slide.

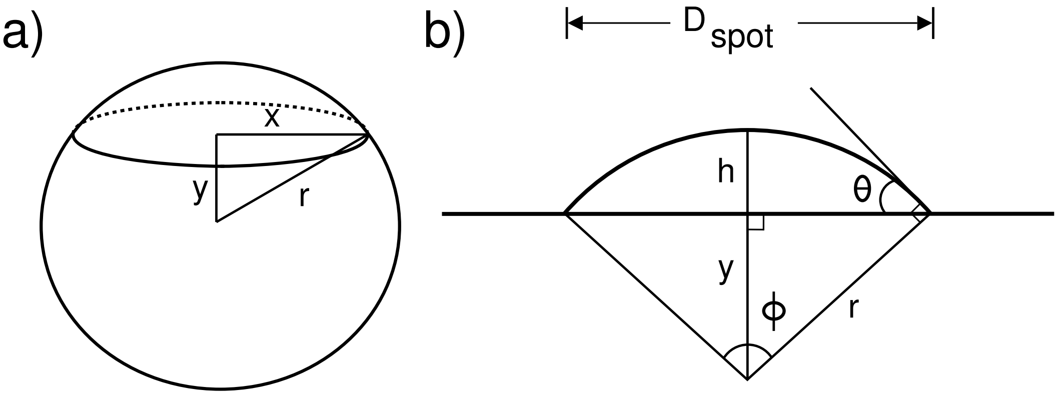

Figure 3.3: Representation of a liquid droplet as a spherical cap.

of an infinite number of circles following the equation

$$A_{circ} = \pi x^2 \tag{3.2}$$

where A_{circ} is the area of the circle and x is its radius. From Figure 3.3a, it can be seen that the Pythagorean identity

$$x^2 + y^2 = r^2 \tag{3.3}$$

relates the radius of a circular slice of a sphere to y, its distance from the center of the sphere, and r, the radius of the sphere.

The volume of a spherical cap can be found by integrating the area of these circular sections over a range of positions relative to the sphere's center. The integral produced from a combination of Equations 3.2 and 3.3 is

$$V_{cap} = \pi \int_{r-h}^{r} \left(r^2 - y^2\right) dy \tag{3.4}$$

where V_{cap} is the volume of the spherical cap and h is the height of the cap. The solution to this integral is

$$V_{cap} = \frac{\pi}{3} h^2 \left(3r - h\right) \tag{3.5}$$

Note that at the limit where $h = 2r$, this becomes the equation for the volume of a sphere, $V_{sphere} = \frac{4}{3}\pi r^3$.

Next, the appropriate variables in this expression need to be replaced to relate droplet volume to spot diameter and contact angle. Figure 3.3b shows a spot in profile, represented by an arc. The angle of the arc is ϕ, the contact angle of the droplet with the surface is θ, and the radius of the sphere that the droplet is cut out of is r. Because the tangent to a circle is perpendicular to its radius,

$$\theta = \frac{\phi}{2} \tag{3.6}$$

The height h of the drop is

$$\begin{aligned} h &= r - q \\ &= r - r\cos\left(\frac{\phi}{2}\right) \\ &= r - r\cos\theta \end{aligned} \tag{3.7}$$

Additionally, r can be replaced by d_{spot} using trigonometry:

$$r = \frac{d_{spot}}{2\sin\theta} \tag{3.8}$$

By substituting the last part of Equation 3.7 along with Equation 3.8 into Equation 3.5, the diameter of a spot can then be written with respect to V_{spot} and θ as

$$d_{spot} = 2\sin\theta\sqrt[3]{\frac{3V_{spot}}{\pi(1-\cos\theta)^2(2+\cos\theta)}} \tag{3.9}$$

or, in terms of commonly used units:

$$d_{spot}(\mu m) = 20\sin\theta\sqrt[3]{\frac{3V_{spot}(pL)}{\pi(1-\cos\theta)^2(2+\cos\theta)}} \tag{3.10}$$

This equation is plotted in Figure 3.4 for three spot volumes: 10 pL, 100 pL, and 1000 pL. As the contact angle approaches 180 degrees, the size of the area of the droplet in contact with the surface drops to zero. This is expected, as a 180 degree contact angle indicates complete non-wetting of a surface. Likewise, the diameter of a spot formed from a given volume increases asymptotically as the contact angle approaches zero. Finally, it can be seen from the functional form of Equations 3.9 and 3.10 that the diameter of a spot scales with the cube root of its volume.

3.1.2 Experimental results

To verify this relationship, droplets of a constant volume were deposited by both a microplotter and a hand pipette on a variety of different substrates and the resulting spot sizes were measured. First, sessile contact angles of deionized water with each of the substrate materials were taken using a DataPhysics Contact Angle System OCA Plus 15 goniometer. These values are listed in Table 3.1. Then, a dilute food coloring solution (assumed to have the same liquid-surface interaction and viscosity as deionized water) was hand-pipetted in a series of 0.5 μL droplets on each of the substrates. A single microplotter needle operating at a constant set of dispensing conditions (ultrasonic frequency and strength) was used to deposit 500 droplets of food coloring on each of the substrates. Finally, the spots were allowed to dry and images of the substrates were captured using a CCD camera attached to a standard optical microscope.

Using the NIH ImageJ analysis package[64], values for spot diameter were obtained. These values are plotted in Figures 3.5a and b for the hand-pipetted and microplotted spots, respectively, and are also present in Table 3.1. Curves with the functional form of Equation 3.10 have been fit to both sets of data. As can be seen, this analytical relationship models the observed spot sizes well, with deposited spot volume being the only fit parameter. The one outlying data point on both graphs is due to a single material that may have had an incorrect contact angle read from it. From these fit curves, values for spot volumes of 27.6 pL for the microplotted spots and 0.468 μL for the hand-pipetted spots were obtained. The analytically obtained spot volume for the hand-deposited spots is close to the volume specified on the hand-pipette and the $\sim$28 pL per

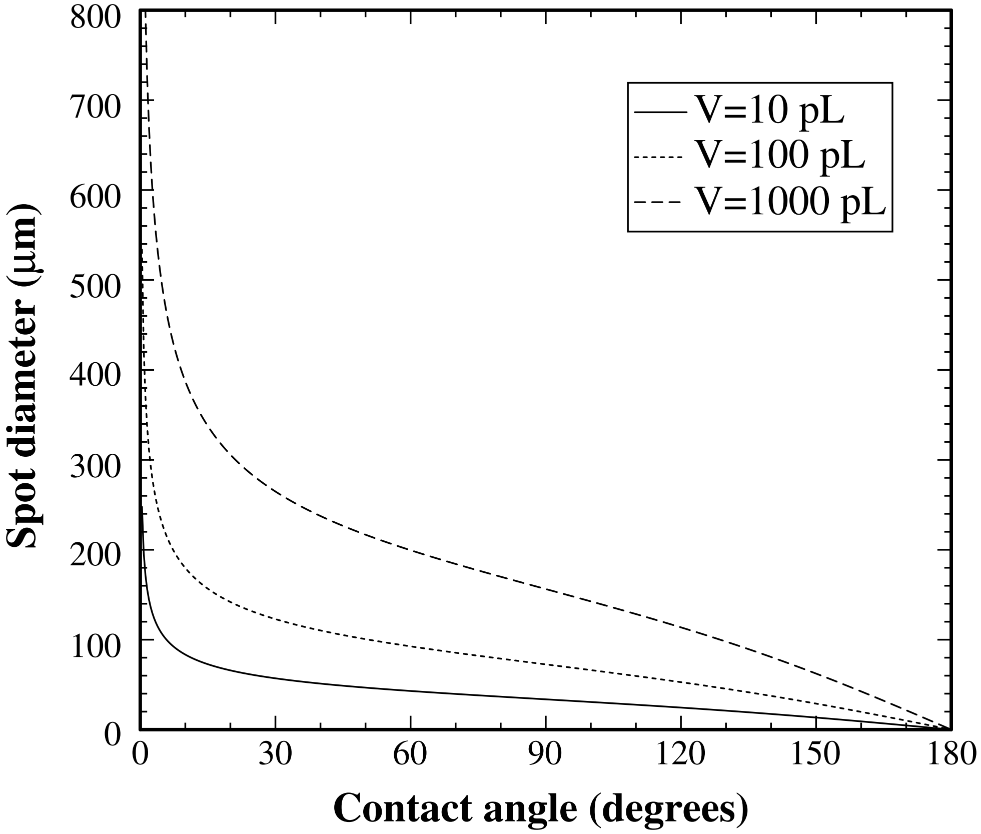

Figure 3.4: Form of the analytical expression for spot size and contact angle. Equation 3.10 is plotted for three values of spot volume: 10 pL, 100 pL, and 1000 pL.

Material	**Sessile contact angle (degrees)**	**Hand-pipetted spot diameter (μm)**	**Hand-pipetted spot CV (%)**	**Microplotted spot diameter (μm)**	**Microplotted spot CV (%)**
Glass	37.4 ± 2.3	1744	7.9	69.5	3.6
Silicon (with native oxide)	49.2 ± 1.8	2194	1.9	80.5	4.6
Gold-coated glass	58.3 ± 1.5	1712	28.3	59.4	5.0
Cast acrylic	70.6 ± 2.0	1295	3.2	57.2	7.3
Electrical grade Teflon	100 ± 3.4	1163	3.4	45.0	11.1

Table 3.1: Sessile contact angles of deionized water with various substrate materials.

microplotted spot is close to the ∼30 pL measured previously for 75 μm diameter spots on glass.

In addition to spot size, we also measured the regularity of the spot diameters. Coefficients of variation (CVs) of spot diameters were recorded for each of the separate spot size studies. More regular, circularly-shaped spots on a microarray greatly aid in the later analysis of experiments performed using that microarray.[21, 22] It can be seen that smooth surfaces, such as glass and silicon, have lower CVs than rough surfaces, perhaps due to a lower variability of the surface across the spotted region.

3.2 Spot diameter as a function of viscosity

In addition to the way a fluid interacts with a surface, viscosity is another fluid property that may affect the diameter of deposited spots. However, according to the above theory, viscosity plays no part in the ultimate shape of a droplet on a surface. If we assume that the droplets that are deposited reach their equilibrium shape, then the way that they flowed into that shape should not matter and differences in liquid viscosity should not

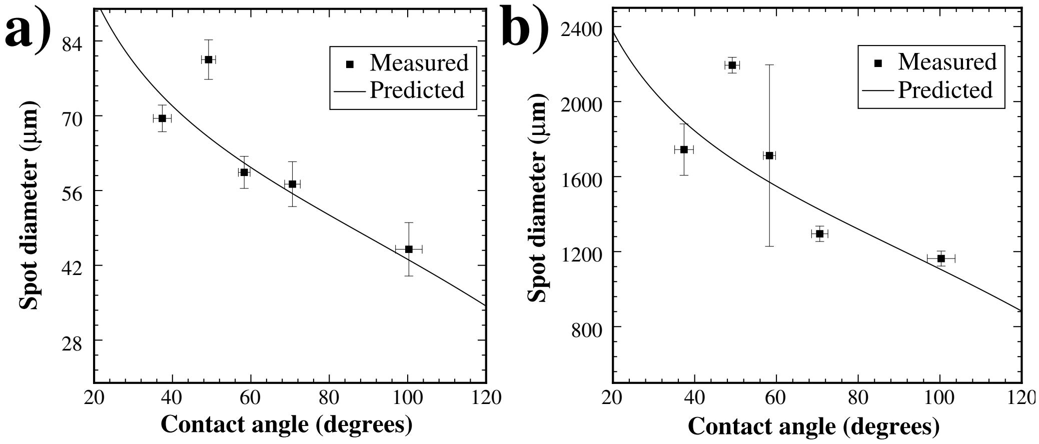

Figure 3.5: Measured spot sizes as a function of contact angle, compared with an analytical model. The model of Equation 3.10 was used to fit measured spot diameters from a) hand-pipetted spots and b) spots deposited using a microplotter.

produce different spot sizes.

We investigated this by depositing a series of fluids with different viscosities on the same substrate under the same conditions and then measuring the resulting spot sizes. To create solutions with different viscosities, a dilute food coloring solution (assumed to have approximately the same viscosity as water) was mixed with glycerol in different ratios. The viscosities of these various mixtures were known from previous studies.[65] These different dye solution / glycerol mixtures were deposited and analyzed as described above. Three sets of spots were deposited, under three different strengths of applied ultrasonics: 4 Vpp, 6 Vpp, and 8 Vpp. The results are shown in Figure 3.6.

Surprisingly, we found that the different solution viscosities led to significantly different spot sizes. In general, the higher viscosity the solution, the larger the diameter of spot deposited. This would seem to run counter to the earlier argument that viscosity does not affect spot size, however there may be an explanation for this in terms of the

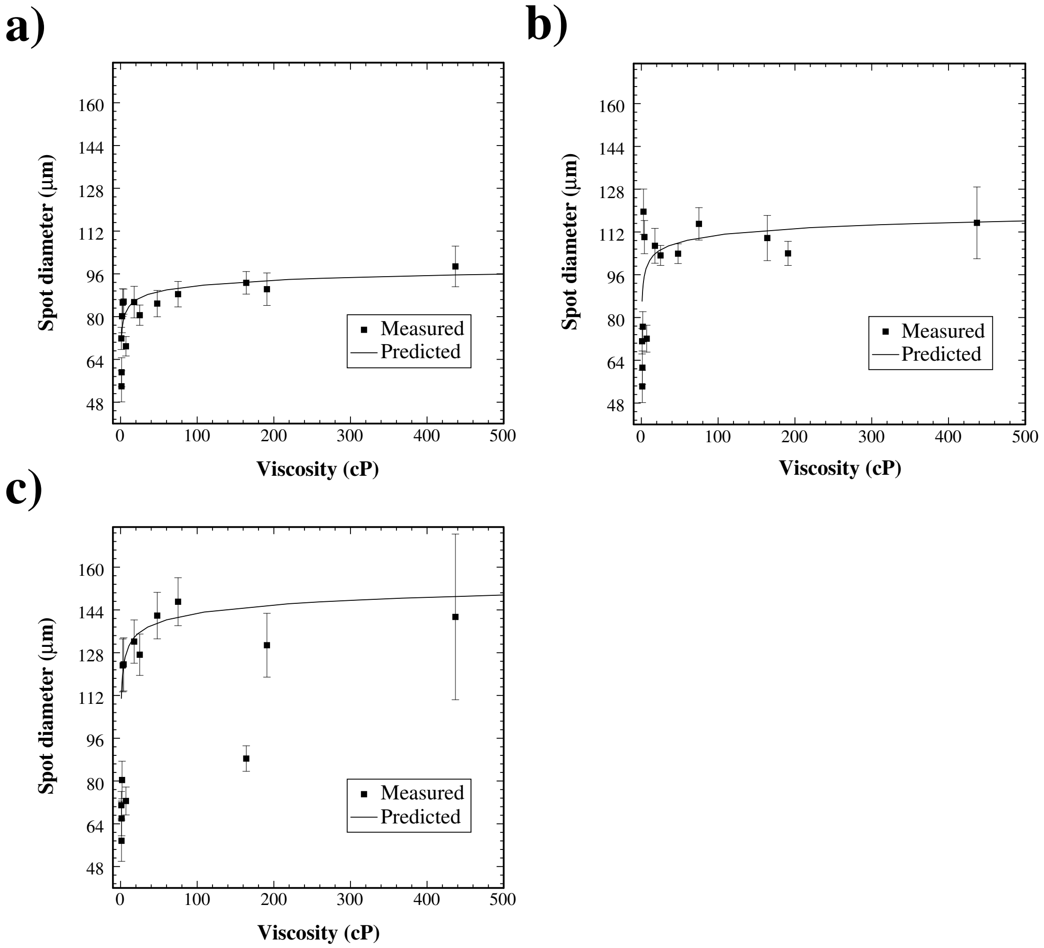

Figure 3.6: Measured spot sizes as a function of viscosity, at three different strengths of applied ultrasonics. Mixtures of glycerol and water were deposited using one dispenser that was driven at the same frequency, but three different amplitudes: a) 4 Vpp, b) 6 Vpp, and c) 8 Vpp.

wetting of the fluids on the surface. Glycerol does not wet a surface in the exact same way that water does, therefore each different glycerol / water mixture may have a different contact angle with the glass surface, leading to the different spot sizes observed. To determine whether this is the case, we assumed that the water / glycerol mixtures had contact angles that could be linearly interpolated from those for pure water or glycerol. We then assumed a contact angle for water of 37.4° and tried to fit Equation 3.10 to each of the sets of deposited spots. We found that for a glycerol contact angle of 15°, the shape of the curve of predicted spot diameters matches the increase in spot size with viscosity for each of the three sets, as can be seen in Figure 3.6.

3.3 Spot diameter as a function of dispenser settings

As was hinted at in the last section, changing the ultrasonic dispenser settings can result in different volumes of fluid being deposited on a surface. The microplotter dispenser relies on applied ultrasonics to cause the gentle pumping action that dispenses fluid. This pumping action will be examined in detail in Chapter 4, but for now we can say that the frequency and amplitude of the vibrations that the dispenser needle experiences can determine how much fluid is delivered to a surface. In the previous study, spots of food coloring solution were deposited at three different amplitudes of vibration. From the curve fits, the volume of the deposited spots were found to be 25 pL, 44 pL, and 94 pL for the 4 Vpp, 6 Vpp, and 8 Vpp strengths of ultrasonics, respectively. As will be described in more detail in Chapter 4, this is consistent with the expectation that higher amplitudes of vibration lead to larger volume droplets being deposited.

3.4 Evaporation

One of the great advantages of the fluid microplotter is its ability to work with nanoliter volumes of fluid, but at this size scale evaporation can be a serious problem. Two areas are affected by evaporation during the operation of the microplotter: droplets deposited on a surface and fluid contained within the microplotter dispenser. The rate of evaporation from small droplets is of importance when dealing with water-dependant proteins, because the drying of these proteins may cause them to be denatured. Also, water must usually be present for the coupling reaction between the biomolecules and the chemically treated surface to go to completion. Evaporation within the microplotter needle can cause problems such as an increase in concentration of the biomolecules within the needle during the time it takes to fully deposit a microarray or a batch of microarrays, leading to irregularity in bound probe concentrations within or between microarrays. In this section, we present relationships to estimate evaporation rates as functions of environmental parameters.

3.4.1 Evaporating droplets

As stated above, evaporation of droplets is a process that affects all microarray production, but may be most important when dealing with sensitive proteins. Additionally, we have observed that rapid evaporation within the small spots that the microplotter produces eliminates the ”coffee-ring effect” seen in larger spots (Figure 2.3, for example, shows uniform spots at a size of 5μm). It is therefore important to understand what factors determine the rate at which small droplets evaporate.

In Section 3.1.1 we described the shape of a droplet on a surface as being that of a

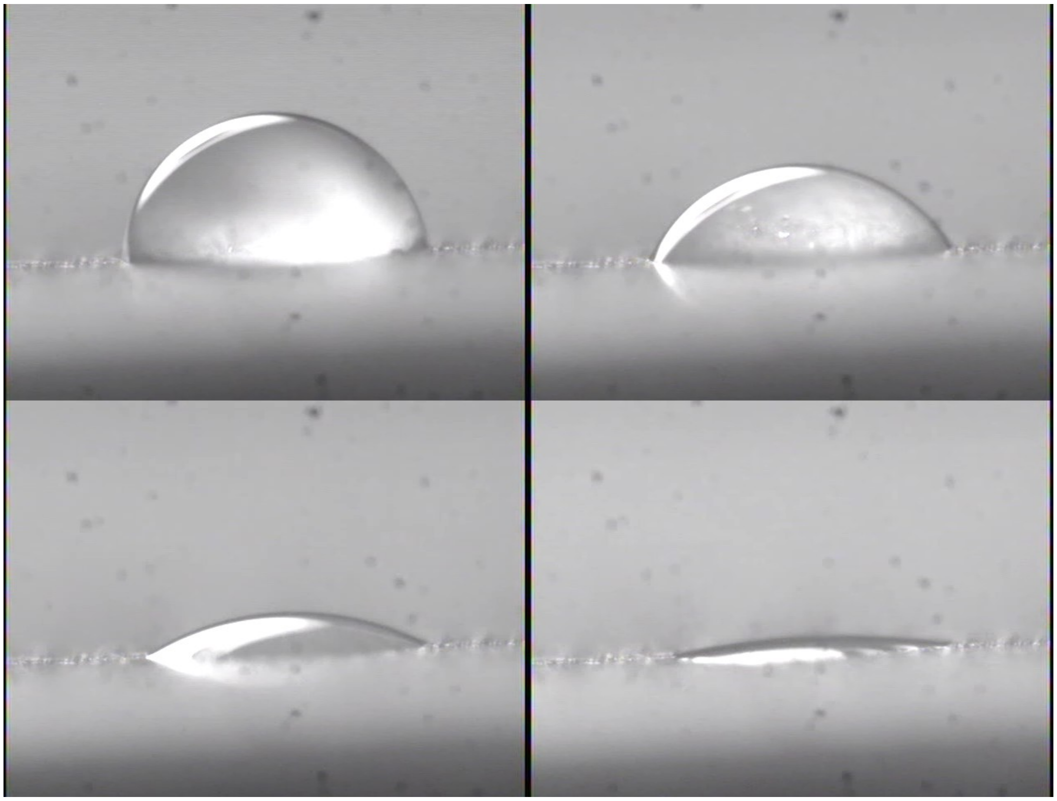

Figure 3.7: Images of an evaporating droplet. A 0.5 μL droplet of DI water was placed on Teflon and images were captured as it evaporated. Note that the droplet does not evaporate by shrinking in diameter, but by changing its contact angle with the surface.

spherical cap. It has been observed that droplets do not evaporate simply by reducing their volume while retaining their same shape, but that the diameter of a droplet remains constant while its contact angle decreases (Figure 3.7).

In his Ph.D. thesis, Yuri Popov modeled the behavior of such evaporating drops.[66] Assuming that droplet shape could be described in the same manner as presented in Section 3.1.1, particularly Equation 3.9, he derived the change in droplet mass over time as

$$\begin{aligned}\frac{dm}{dt} &= -\pi r_{spot} D_{AB}(n_s - n_\infty) \cdot \\ &\left[\frac{\sin\theta}{1+\cos\theta} + 4\int_0^\infty \frac{1+\cosh 2\theta\tau}{\sinh 2\pi\tau} \cdot \tanh\left[(\pi-\theta)\tau\right] d\tau\right] \qquad (3.11)\end{aligned}$$

where D_{AB} is the diffusivity of A (in this case, the vapor of the liquid in the droplet) in B (air), n_s is the density of saturated vapor above the liquid-air interface, n_∞ is the ambient vapor density, r_{spot} is the radius of the spot $\left(\frac{d_{spot}}{2}\right)$, and τ is an artifact of the integration used. This exact expression needs to be solved numerically for every θ, but at smaller contact angles ($\theta < 57°$) an approximation can be used to simplify this expression. Using this approximation, he determined that the time it takes for a droplet to completely evaporate is

$$t_f = \frac{\pi\rho r_{spot}^2 \theta_i}{16 D_{AB}(n_s - n_\infty)} \qquad (3.12)$$

where ρ is the density of the liquid and θ_i is the initial contact angle of the droplet. In the case of a water droplet in air, this can be modified into

$$t_f = \frac{\pi\rho r_{spot}^2 \theta_i}{16 D_{WA} n_s (1 - \frac{RH}{100})} \qquad (3.13)$$

where RH is the relative humidity (%) of the ambient air and D_{WA} is the diffusivity of water in air.

The density of saturated vapor above the liquid-air interface can be determined (assuming an ideal gas) from

$$n_s\left(\frac{g}{cm^3}\right) = M_W\left(\frac{g}{mole}\right) \cdot \frac{P^*(atm)}{R\left(\frac{atm\cdot cm^3}{mol\cdot K}\right) T(K)} \qquad (3.14)$$

where M_W is the molar mass of water (18 $\frac{g}{mole}$), P^* is the vapor pressure of water, R is the gas constant (82.06 $\frac{atm \cdot cm^3}{mol \cdot K}$), and T is the ambient temperature. The vapor pressure of water at a given temperature can be found using the Antoine Equation[67]:

$$\log_{10} 0.987 P^*(atm) = A - \frac{B}{T(K) + C} \tag{3.15}$$

The substance-specific variables are, for water, A = 5.40221, B = 1838.675, and C = -31.737. [68]

The diffusivity of water in air can be estimated by the following[67]:

$$D_{WA} = \frac{0.00143 T(K)^{1.75}}{P(atm)\left(\frac{2}{1/M_W + 1/M_A}\right)^{\frac{1}{2}} \left[(\Sigma_v)_W^{\frac{1}{3}} + (\Sigma_v)_A^{\frac{1}{3}}\right]^2} \tag{3.16}$$

where M_A is the molar mass of air (29 $\frac{g}{mole}$), $(\Sigma_v)_W$ = 13.1, and $(\Sigma_v)_A$ = 19.7.

At room temperature (T = 297 K), 1 atmosphere of pressure, a relative humidity of 30%, and a contact angle of water with glass (37°), Equations 3.13, 3.14, 3.15, and 3.16 simplify to

$$t_f(s) = 0.00008245 d_{spot}^2(\mu m) \tag{3.17}$$

Table 3.2 shows the time required for droplets of various sizes to evaporate under the conditions listed above. As can be seen, small droplets in the size range typically deposited by the microplotter evaporate almost instantly under these conditions. This flash evaporation reduces the ”coffee-ring” effect seen in larger spots, where material builds up at the edges of a spot during evaporation, because the solute does not have enough time to diffuse to the edge of the spot.[66, 69]

However, rapid evaporation like this may be a problem when dealing with moisture-dependent proteins. It make take up to 14 hours to print a batch of protein microarrays and the droplets of these proteins in solution must not evaporate within this time. Compounds, such as the glycerol used in the viscosity studies of Section 3.2, may be added to reduce the rate of droplet evaporation, but these chemicals can reduce the activity of sensitive proteins. The evaporation times of Table 3.2 are based on the assumption of certain environmental conditions, but altering these can provide significant help. The largest increase in evaporation time of a droplet can come from raising the relative humidity of the surrounding atmosphere. The term $\left(1 - \frac{RH}{100}\right)$ is present in the denominator of Equation 3.13, indicating that evaporation time will increase asymptotically as the relative humidity approaches 100%. For example, increasing the relative humidity from 30% to 90% increases evaporation time by a factor of 7. Going from 90% to 95% humidity further doubles that time. Finally, by lowering the temperature the rate of diffusion of the water from the droplet into the atmosphere is reduced, due to a suppressed water vapor pressure. Dropping the temperature from 24°C to 1°C increases the evaporation time by a factor of 5.7. An even greater benefit from lowered temperatures may come from the observation that sensitive proteins are less prone to denaturation in cooler conditions.

3.4.2 Evaporating solvent within a dispenser

Evaporation is also an issue within the microplotter dispenser itself, because the evaporation of solvent can lead to a buildup of solute in the dispenser. This may cause a steady increase in the concentration of biomolecule solutions as they are deposited across a microarray or a batch of microarrays, leading to undesirable inconsistencies in microarray fabrication. Clogging can also become a problem if too much solute builds up in the

Diameter (μm)	**Time to evaporate (s)**
5	0.002
20	0.033
30	0.074
50	0.21
100	0.82
200	3.3
1000	82

Table 3.2: Evaporation times for droplets of different diameters. The droplets were assumed to be water resting on glass, at a temperature of 24°C and a relative humidity of 30%. The times were calculated from Equation 3.17.

small aperture of the needle. This is a significant issue with the non-microarray application of fabricating polymer microcircuitry, as the organic solvents used tend to have high vapor pressures at room temperature (and low partial pressures in ambient atmosphere), therefore they evaporate much faster than water, and the polymers that are the solute tend to precipitate out of solution easily at high concentrations.

In a microplotter dispenser open at both ends (depicted in Figure 3.8), evaporation occurs at both the wide end and at the needle tip. We will examine both cases individually and combine them together to get the total evaporation rate within the dispenser. Figure 3.8 illustrates the configuration of the dispenser and highlights the evaporation that takes place at either end.

The evaporation from the wide end of the dispenser can be modeled as a simple case of one-dimensional unsteady-state diffusion into a semiinfinite medium. A solution to this case is as follows:[67]

$$\frac{dm}{dt} = -\frac{\pi d_{cyl}^2}{4}\left(n_s - n_\infty\right)\sqrt{\frac{D_{AB}}{\pi t}} \tag{3.18}$$

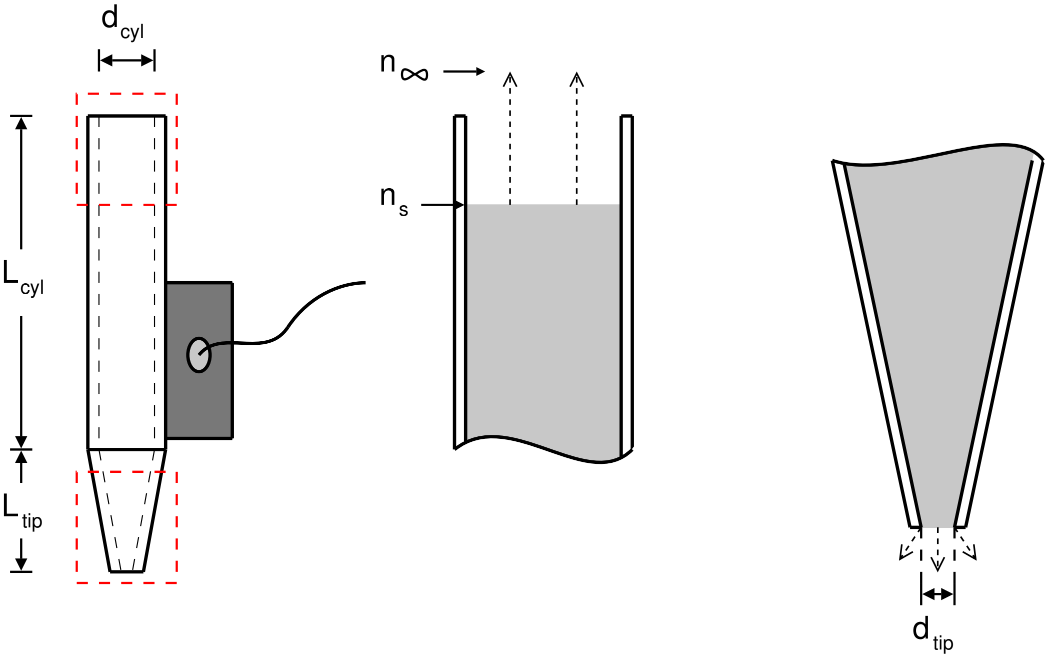

Figure 3.8: Schematic of evaporation within the microplotter dispenser. The entire dispenser is shown at left, along with magnified cross-sections of evaporation occurring at both the wide end of the dispenser as well as the needle tip.

where d_{cyl} is the inside diameter of the cylindrical portion of the microplotter dispenser. Integrating from time 0 to t gives the total mass (m) evaporated over this time period:

$$m = \frac{\pi}{2} d_{cyl}^2 \left(n_s - n_\infty\right) \sqrt{\frac{D_{AB} t}{\pi}} \tag{3.19}$$

Modeling the evaporation from the tip of the needle requires revisiting Equation 3.11. By solving that equation in the case where $\theta = 0$, the evaporation rate from a flat circle on a surface is determined. This is a good approximation of what happens at the circular opening of the needle tip. The evaporation rate is then

$$\frac{dm}{dt} = -2 d_{tip} D_{AB} \left(n_s - n_\infty\right) \tag{3.20}$$

where d_{tip} is the inside diameter of the tip of the needle. Like in the case of evaporation from the wide end of the needle, this expression can be integrated over time 0 to t to yield the total mass lost over that span:

$$m = 2 d_{tip} D_{AB} \left(n_s - n_\infty\right) t \tag{3.21}$$

Combining Equations 3.19 and 3.21 gives one expression for the mass of solvent lost over time t:

$$m = \left(n_s - n_\infty\right) \left(\frac{\pi}{2} d_{cyl}^2 \sqrt{\frac{D_{AB} t}{\pi}} + 2 d_{tip} D_{AB} t \right) \tag{3.22}$$

The volume inside a single dispenser can be calculated using the following formula:

$$V_{dispenser} = \frac{\pi L_{cyl} d_{cyl}^2}{4} + \frac{\pi L_{tip} d_{cyl}^2}{12} \tag{3.23}$$

where L_{cyl} is the length of the cylindrical portion of the needle and L_{tip} is the length of the conical tip (both dimensions are shown in Figure 3.8). A typical dispenser has L_{cyl}

= 1 cm, L_{tip} = 2 mm, d_{cyl} = 1 mm. Therefore, it has a loading capacity of 8.4 μL. Using the values presented in Section 3.4.1, Equation 3.22 indicates that it should take 150 hr for a fully-filled dispenser (with a 20μm I.D. tip) to evaporate at room temperature and 30% humidity. However, when depositing spots on a microarray, the microplotter dispenser is often loaded with much less solution than this. It can take as little as 38 seconds to evaporate 1 nL of solution from the dispenser, a volume that may be used to conserve reagents in the fabrication of microarrays. At such small volumes, evaporation of solvent can cause large changes in the concentration of solution contained within the dispenser. As shown earlier, humidifying the surrounding atmosphere can greatly reduce the rate of this evaporation.

3.5 Conclusion

In this chapter, we have characterized the operation of the fluid microplotter in regards to spot deposition. We presented an analytical expression to relate the diameter of a deposited spot as a function of the volume of a droplet and the wettability of a surface. We compared this expression to experimental results and found a good correlation. We tested the effect of fluid viscosity on spot size, and while we found the spot size appeared to increase with increasing viscosity, we argue that the larger spot size is in fact due to the glycerol in the high viscosity solutions wetting glass more readily. The effect of different ultrasonic settings on deposited fluid volume was evaluated. Finally, we explored evaporation that occurs on a deposited droplet and within the dispenser itself.

Chapter 4

Acoustic pumping and spraying

In the previous chapter, we examined the properties of the microplotter dispenser when it deposits spots of fluid during the fabrication of a microarray. This deposition of fluid relies on ultrasonic vibrations that cause a gentle pumping action within the microplotter dispenser. In this chapter, we will attempt to explain the process by which this pumping occurs, along with a description of the spraying that happens with higher-amplitude ultrasonic vibrations.

To do so, we will first observe the resonances of the PZT piezoelectric in an attempt to measure and/or predict at what frequencies pumping and spraying occur. These measurements will also provide some insight as to the type of vibrations produced by the piezoelectric. We will then briefly discuss the ultrasonics-induced spraying that occurs when the fluid outside the dispenser is different than that inside. Finally, we will explore the pumping behavior observed when the same fluid is present outside and inside the dispenser, analytically and with the aid of computer fluid models.

4.1 Measuring resonances

As described in Chapter 2, the dispenser at the core of the fluid microplotter is composed of a hollow, tapered glass needle attached to a rectangular piece of PZT piezoelectric. An example of such a dispenser is shown in Figure 4.1 (Figure 3.8 from Chapter 3 also shows a schematic of a dispenser). When an alternating current is supplied to the

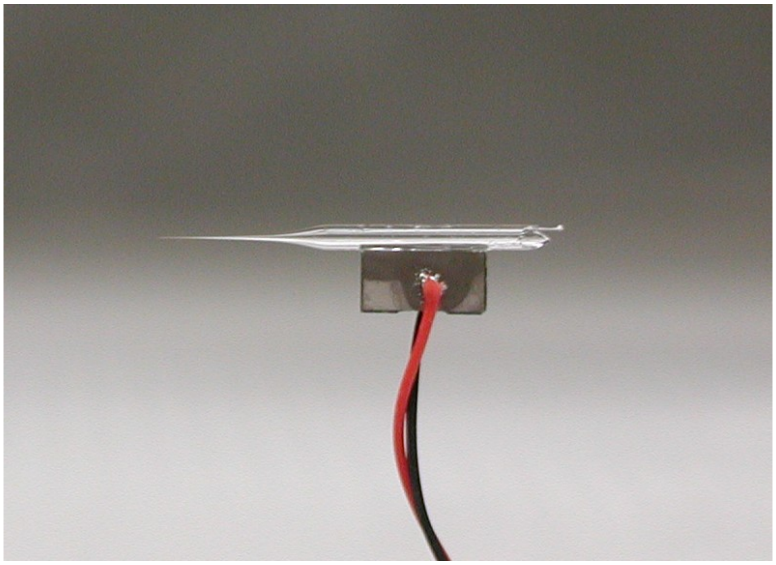

Figure 4.1: An example of a microplotter dispenser. A rectangular piece of PZT piezoelectric is bonded to the side of a pulled-glass micropipette. Electric leads are attached on both faces of the piezoelectric element.

piezoelectric, it vibrates. At certain frequencies, a pumping action occurs within the needle or, if the amplitudes of vibration are high enough, fluid is sprayed out of the end of the needle. A means of measuring or predicting the specific frequencies at which this happens is necessary for operation of such a dispenser to become practical.

It is assumed that these frequencies correspond to some sort of resonance within the dispenser, so we used impedance analysis to first look at resonances within the piezoelectric itself. A piezoelectric element can be modeled as a circuit, shown in Figure 4.2, with an inductance (L), a resistance (R), a parallel capacitance (C_p), and a series capacitance (C_s). This is called an LCR circuit and it represents an electrical resonator. C_p represents the electrostatic capacitance of the piezoelectric between the two electrical

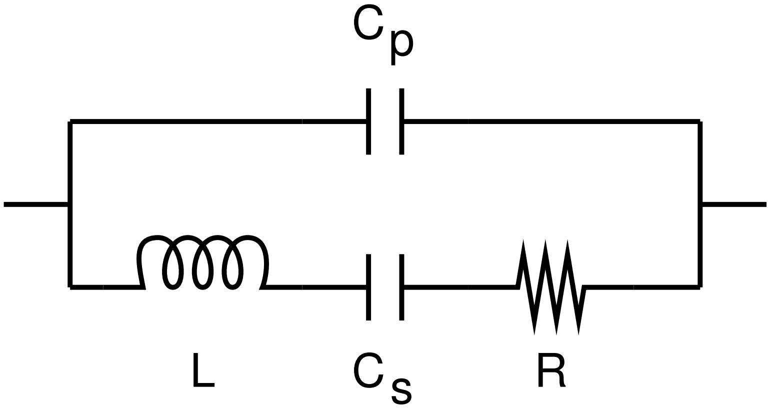

Figure 4.2: LCR circuit model of a piezoelectric element.

leads and the L, C$_s$, and R elements represent the mechanical properties of the piezoelectric. Such a circuit has both a resonance condition as well as an antiresonance. At resonance, C$_p$ is much larger than C$_s$. At antiresonance, there are low electromechanical losses. Antiresonance occurs at a slightly higher frequency than resonance.

Impedance can be used to measure the resonant properties of a piezoelectric element. Impedance is defined as ”the total apparent opposition a circuit offers to the flow of alternating current in an electrical circuit”.[70] By measuring impedance as a function of frequency, resonance and antiresonance can be directly observed. Figure 4.3 shows a sample impedance spectrum from a rectangular piece of PZT piezoelectric. Resonance occurs when the impedance drops sharply and antiresonance is the sharp rise in frequency that occurs at a slightly higher frequency.

Two resonances can be seen in this spectrum, one corresponding to the resonance along the width of the PZT rectangle, the other for resonance along its length. Figure 4.4 shows a schematic of the piezoelectric element used in the construction of the

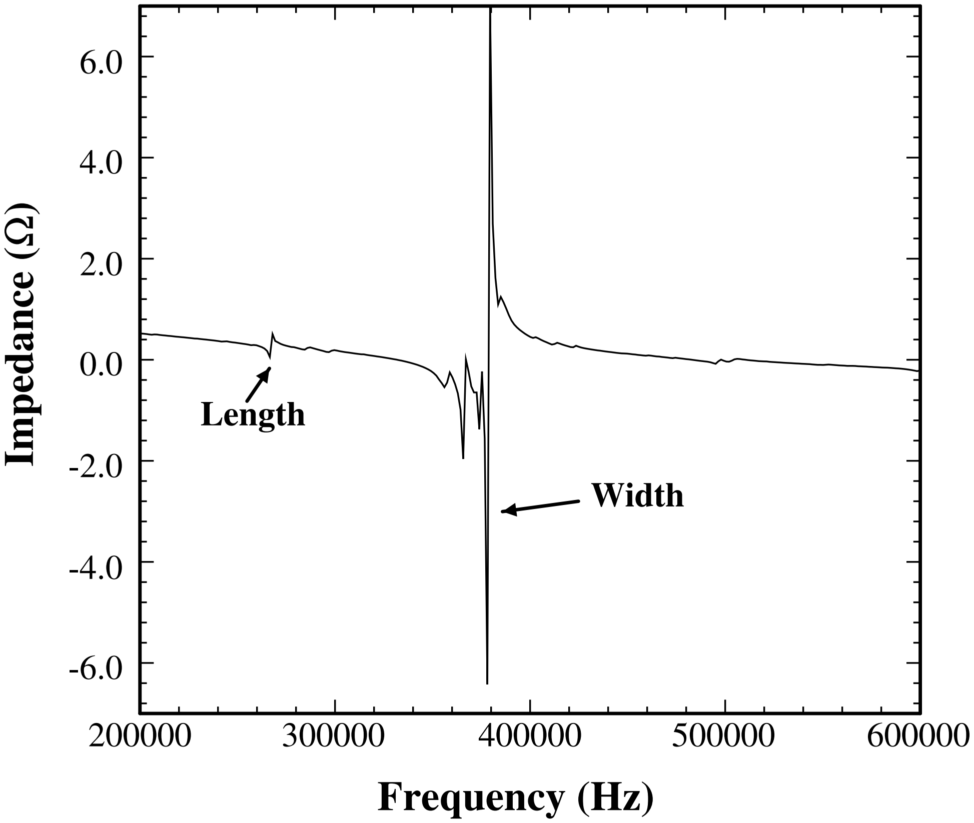

Figure 4.3: Example of an impedance spectrum for PZT. In this impedance spectrum two resonances can be seen: a small one that corresponds to the length mode of the piezoelectric, and a much larger one for the width mode.

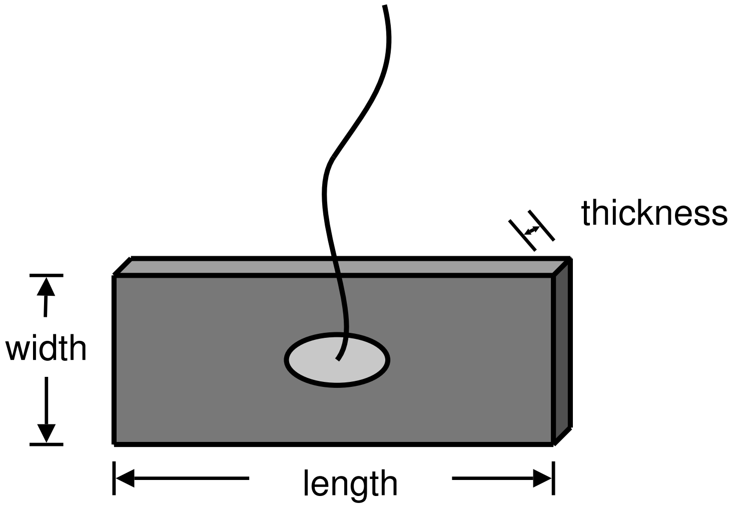

Figure 4.4: A schematic of the piezoelectric element used in a microplotter dispenser.

microplotter dispenser, where we define the dimensions of length, width, and thickness. The piezoelectric element is a rectangular slab of PZT bonded to a hollow glass needle along its length, and its width is perpendicular to the axis of the glass needle. Typical sizes for the length, width, and thickness are 4-6 mm, 2-4 mm, and 0.5 mm, respectively. We concentrated on the width and length resonances, because pumping and spraying were not observed at the much-higher-frequency resonances due to PZT thickness.

We examined the length and width resonant frequencies for a series of PZT rectangles with different dimensions. The resonant frequency for a given dimension (length, width, or thickness) in a rectangular piece of piezoelectric is simply the frequency at which a standing wave is set up along that dimension. These standing waves exist at the following condition:

$$f = \frac{cn}{2L} \tag{4.1}$$

where f is the resonant frequency, c is the speed of wave propagation in PZT, L is the appropriate dimension of the PZT (length, width, or thickness), and n is an integer. The first, or $\frac{\lambda}{2}$, resonance occurs at

$$f = \frac{c}{2L} \tag{4.2}$$

To determine the speed of wave propagation in the PZT used for the dispenser, and thus be able to predict resonances of the piezoelectric rectangle as a function of its length and width, we purchased a 0.5 mm thick PZT wafer from Stavely Sensors (piezoelectric type EBL#1) and diced it into rectangles of different lengths and widths. Impedance spectra were gathered for each of these pieces on an Agilent 4396B Impedance Analyzer. Due to the particular configuration of the test kit on this analyzer, we actually measured the reactance in order to determine the impedance, therefore the scale of the spectra shown is uniformly off. The length and width mode resonances were determined and are plotted as a function of inverse distance in Figures 4.6 and 4.5, respectively. The linear fits through those points, with y-intercepts of zero, provide values for the speed of wave propagation in PZT. These values can be seen in Table 4.1 The outliers on either plot were due to PZT pieces where the length and width dimensions were the same. It is assumed that vibrations from two directions interacted to significantly shift the resonance of the PZT.

One of the objectives of these measurements was to determine what type of vibrations were responsible for the observed resonances. There are two primary classes of vibrations: longitudinal and transverse. As depicted in Figure 4.7, transverse vibrations are perpendicular to the propagation direction of the wave, such as in the vibration of a string. Longitudinal vibrations are parallel to the propagation direction of the wave, an

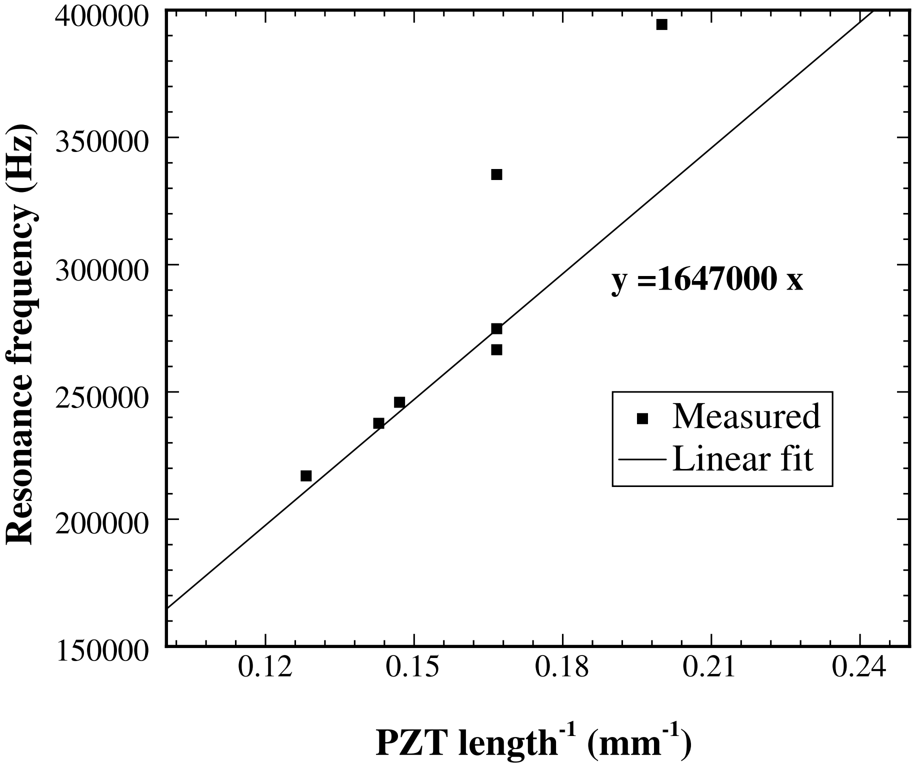

Figure 4.5: Length mode resonances in PZT. A linear increase in measured resonant frequency of the length of a PZT element as a function of inverse length is observed, consistent with Equation 4.2. The two outlier points are from PZT elements with identical widths and lengths.

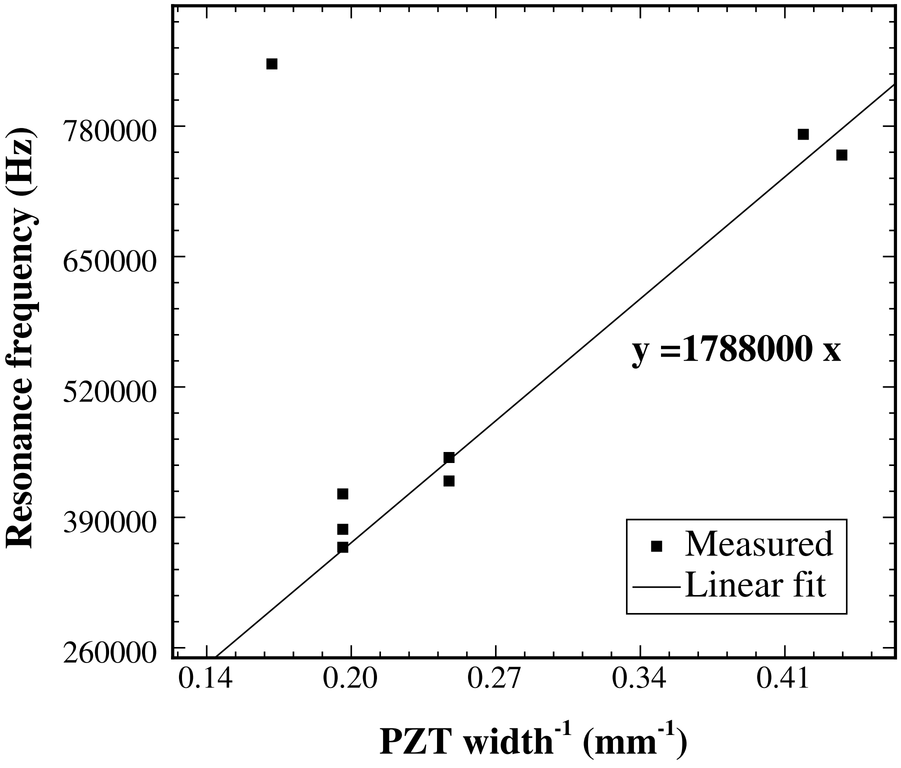

Figure 4.6: Width mode resonances in PZT. A linear increase in measured resonant frequency of the width of a PZT element as a function of inverse width is observed, again consistent with Equation 4.2. The two outlier points are from PZT elements with identical widths and lengths.

Source	Type	Speed of wave propagation (m/s)
Manufacturer		
	Longitudinal	4298
	Transverse	2642
	Average	3470
Experimental		
	Width	3576
	Length	3294
	Average	3435

Table 4.1: Measured speed of wave propagation in PZT, compared with values from supplier.

example being sound waves traveling through air. The manufacturer of the piezoelectric used specifies two different speeds for wave propagation through this type of PZT, depending on whether the PZT is vibrating in a transverse or longitudinal mode. Both of these values are in Table 4.1. It can be seen that the values for wave propagation in PZT found in our measurements do not agree with either of the manufacturer-specified values, but match their average very well. This seems to indicate that both longitudinal and transverse waves are propagating through the piezoelectric at resonance.

After characterizing the resonant properties of lone piezoelectric elements, we examined what happens when a hollow glass tube was added to the piezoelectric. Figure 4.8 shows impedance spectra gathered from both a lone piece of piezoelectric and that same piece of piezoelectric bonded to a 40 mm long hollow glass capillary. As can be seen, both the length and width mode resonances of the piezoelectric are damped, although only the length mode resonance is shifted in frequency. This could be due to the fact that the capillary is attached to the piezoelectric along its length, interfering the its ability to vibrate in that direction, but the piezoelectric is still free to vibrate along its width.

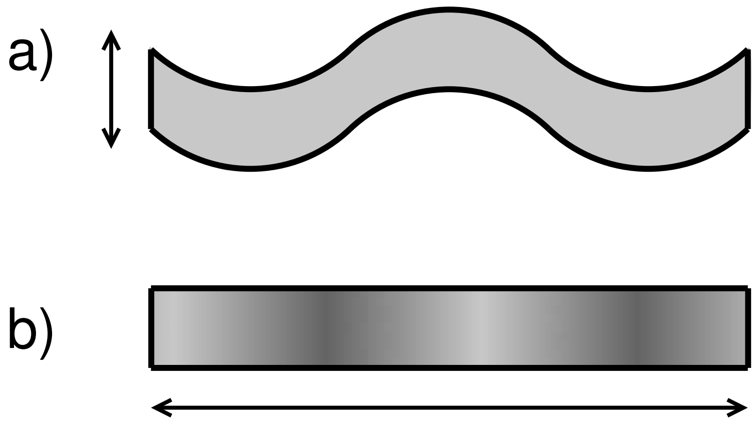

Figure 4.7: Types of vibrations. A plate is shown edge-on as it is subjected to a) transverse and b) longitudinal vibrations.

This hollow tube was then trimmed in length several times by removing sections of the tube on alternating sides of the PZT, with impedance spectra gathered after each cut. Table 4.2 lists the resonances found after each cut, along with the dimensions of the capillary. The measured resonances of the PZT did not shift by much with each cut, and there is no clear correlation between capillary length and the resonances.

To replicate the conditions in an actual dispenser, impedance spectra were measured in a PZT-capillary assembly before and after filling the interior of the capillary with water. The results are shown in Figure 4.9. The filling of the interior of the capillary with water shifted the frequencies of the two strongest resonances from 680 kHz and 770 kHz to 655 kHz and 751 kHz, respectively, and further reduced their strength. This indicates that it may be possible to design an electronic fill-level sensor for a microplotter dispenser using some variant of these impedance measurements. Additionally, the significant difference in resonant frequencies between empty and water-filled capillaries ($\sim$20 kHz) could be a problem when spraying out the contents of a dispenser, because

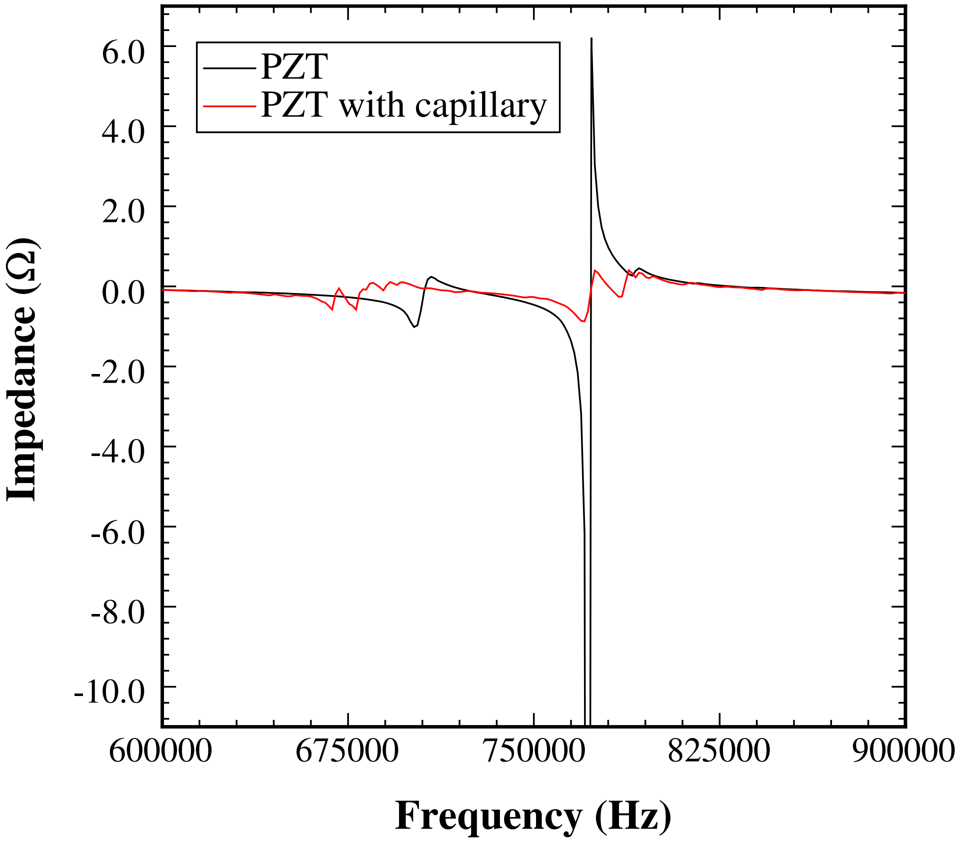

Figure 4.8: Damping of resonances in PZT with the addition of a capillary. By bonding a capillary to a piece of PZT, its length and width mode resonances are damped, but only the length mode resonance is shifted in frequency.

Total capillary length (mm)	**Capillary left of PZT (mm)**	**Capillary right of PZT (mm)**	**Resonance 1 (Hz)**	**Resonance 2 (Hz)**
40.3	15.5	18.0	681852	770087
38.3	15.5	16.0	677722	769712
34.0	11.2	16.0	680726	771214
30.2	11.2	12.2	680726	770088
27.1	8.1	12.2	683354	770088
22.8	8.1	7.9	682603	768210
19.7	5.0	7.9	684856	768961
18.0	5.0	6.2	679975	770088

Table 4.2: Measured resonances as a function of capillary length. A hollow glass capillary was bonded to a piece of PZT, then short segments of it were removed from alternating sides. Impedance spectra were gathered with each cut and the two largest resonances were found.

the driving frequency applied to a dispenser would need to be continually adjusted as fluid is emptied out of the needle in order to maintain a constant spray rate. This may be corrected for either by gradually changing the applied frequency as a dispenser is emptied or by using some sort of electronic feedback to keep the frequency locked on resonance.

Aside from the potential use in detecting the presence or absence of fluid within a dispenser, these impedance measurements are only important if they somehow correspond to the observed pumping and spraying actions of the microplotter dispensers. To test whether the resonance frequencies observed in the impedance spectra match with the frequencies at which pumping and spraying occur, we obtained impedance spectra from two dispensers (both filled with water): one with a hollow glass needle attached to a piece of piezoelectric, the other with a straight glass capillary bound to the piezoelectric. We determined the strongest resonances from these spectra along with their

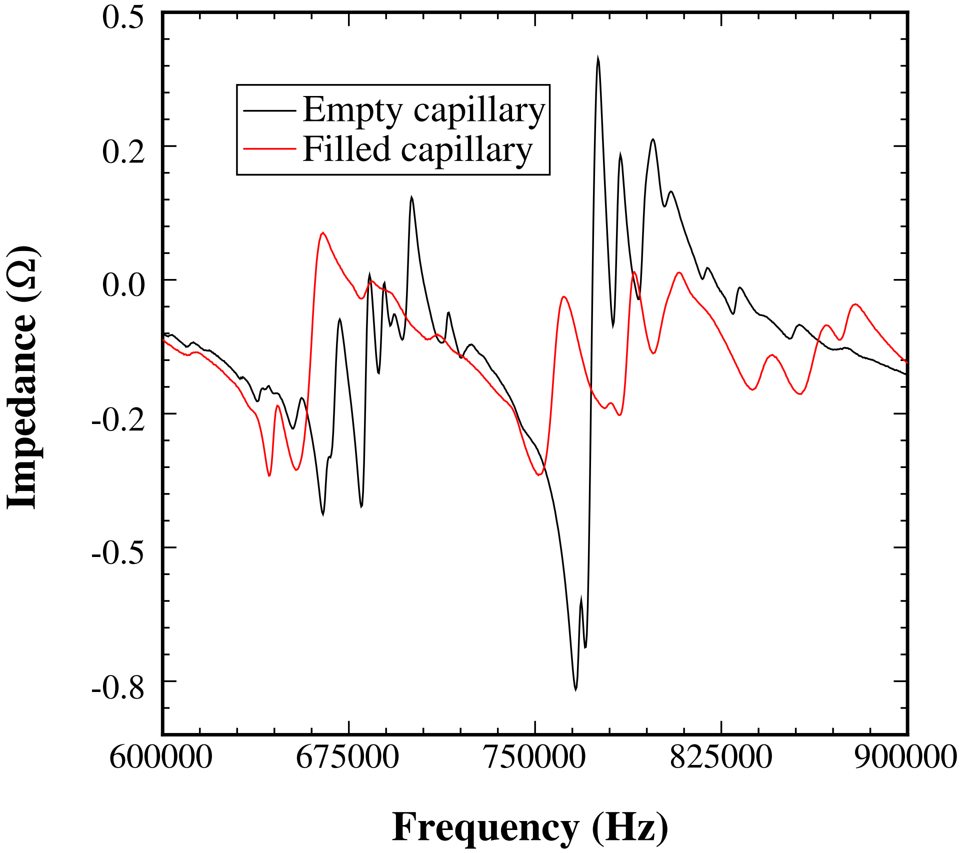

Figure 4.9: Damping of resonances in a PZT/capillary dispenser with the addition of water. A significant shift in resonance frequencies is observed after filling a capillary dispenser with water, as well as a damping of the resonances.

PZT Resonance (Hz)	Strength (Ω)	Spraying frequency (Hz)	Spray strength
259700	0.21	250000	Moderate
372600	0.0534		
398700	1.3259	395000	Strong
474500	0.0099		
625900	0.05	630000	Weak
857600	0.12	650000	Moderate
948100	0.02	940000	Weak

Table 4.3: Measured resonances vs. observed spraying in straight capillary.

PZT Resonance (Hz)	Strength (Ω)	Spraying frequency (Hz)	Spray strength
		270000	Weak
643200	0.46	642000	Strong
653700	0.44	654000	Strong
680000	0.03		
784000	0.27	778000	Moderate
797500	0.15		
		850000	Weak

Table 4.4: Measured resonances vs. observed spraying in a tapered needle.

intensities. These values are listed in Tables 4.3 and 4.4. We then determined the frequencies at which spraying from these dispensers occurred (pumping happens at the same frequencies as spraying, but spraying is easier to observe) and qualitatively determined the strength of this spraying. These values are also listed in Tables 4.3 and 4.4, and it can be seen that not only does spraying occur at the same frequencies as the resonances obtained from the impedance spectra, but the strength of spraying trends in the same direction as the strength of the measured resonances. Therefore, impedance measurements can be used to predict the exact frequencies at which the strongest pumping or spraying will occur, eliminating the trial-and-error determinations used to this point.

A resonant frequency is defined as ”the frequency of maximum forced amplitude”, that is, for a given vibrational force (or alternating electrical current, in this case) applied to the system, the largest possible displacement of the vibrating element occurs. The reason why pumping and spraying appear to occur only at the resonant frequencies of the piezoelectric may simply be due to the fact that the piezoelectric only vibrates with a great enough amplitude at resonance to cause these phenomena to be visible. The direct match between resonance measured using impedance and the frequency at which spraying occurs (as well as a match between the relative intensities of both), coupled with the measurements showing a minimal change in resonant frequency as a function of capillary length, is consistent with this. If resonance within the glass portion of the dispenser is important, we would expect that the measured resonance would change significantly with a change in length of the glass, but this does not happen. This seems to indicate that pumping and spraying may also occur at non-resonant frequencies if a higher-voltage AC signal were applied to the piezoelectric. In Section 4.3.1 we will show that resonance occurs within the conical portion of a pulled glass needle near 950 kHz, a significantly higher frequency than those where strong spraying behavior was observed in the above tapered needle dispenser. However, if the resonant frequency of the piezoelectric and the conical portion of the glass dispenser were made to overlap, a much higher intensity spraying or pumping action may be observed at that combined resonant frequency.

4.2 Spraying

In the previous section, we described a means of predicting at which frequencies pumping or spraying occurs, but we have not discussed the mechanisms responsible for these processes. We will first focus on the more well-understood effect, ultrasonics-induced spraying, an example of which can be seen in Figure 4.10. Spraying is a critical component in the operation of the microplotter because it allows for the rapid switching of fluids, a necessary function when depositing microarrays from thousands of different biological solutions. When deposition of a particular solution is finished, the microplotter dispenser can be moved over a well of that solution (to reuse the remaining solution in the needle) or a waste well (to dispose of the solution) and spraying can be activated until the dispenser is empty. The needle of the dispenser can then be loaded with a solvent, followed by the spraying of that solvent into a waste well. This last step is used to remove any remaining biomolecules adsorbed to the walls inside the needle and can be repeated a sufficient number of times to eliminate all cross-contamination of biomolecule solutions. However, this process can only be made practical if it is automated from beginning to end, and that requires an understanding of the cause of spraying from the microplotter.

It has been known for decades that the ultrasonic vibrations can cause droplets to be dislodged from the surface of a fluid, a process called atomization. This can also be referred to as nebulization if these droplets are only a few microns in diameter. Nebulization is used in humidifiers and in inhalation-based drug delivery devices for treating conditions such as asthma, among other applications.

Many different configurations of nebulizers exist, but they all work in one of two ways. First, an ultrasonic transducer may impart strong vibrations into the fluid that lead to cavitation (the formation and violent collapse of small bubbles) immediately below the fluid's surface. As shown in Figure 4.11a, these bubbles cause the ejection of droplets from the surface of the fluid. Second, applied ultrasonics may cause the

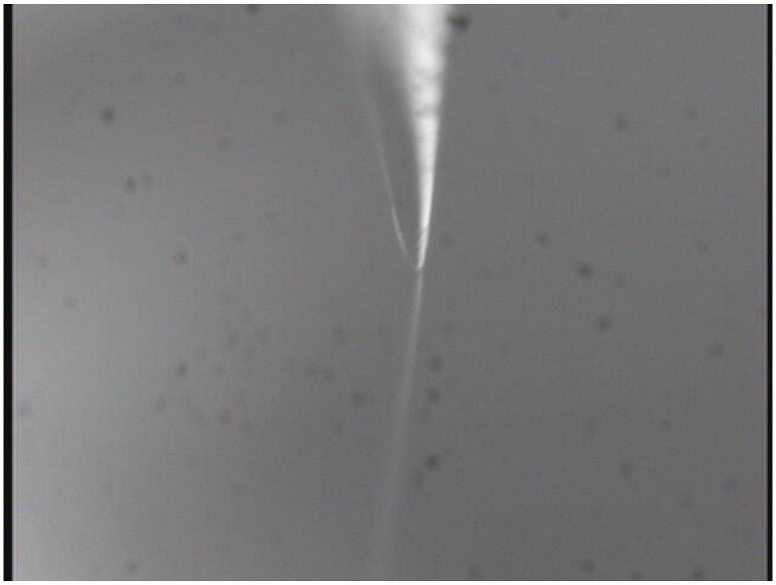

Figure 4.10: An image captured from the microplotter's CCD camera of ultrasonics-induced spraying.

formation of a surface wave, such as the one seen in Figure 4.11b. If the amplitude of this surface wave is large enough, then the crests of the wave becomes so thin and sharply curved that it is energetically favorable for them to pinch off into droplets that are then ejected from the surface.[71] The intensity of the ultrasonics used is what determines which of the two processes causes nebulization: cavitation typically takes place at the very high ultrasonic amplitudes characteristic of ultrasonic emulsifiers and cleaners, while surface wave breakup usually occurs with more gentle commercial ultrasonic sprayers.[72]

Surface wave breakup appears to be the most likely mechanism for the spraying observed in the microplotter dispenser. Using Kelvin's equation for the wavelength of capillary waves and the observation that the surface wave frequency is half of that for the exciting sound, Lang derived the following expression for the wavelength of capillary

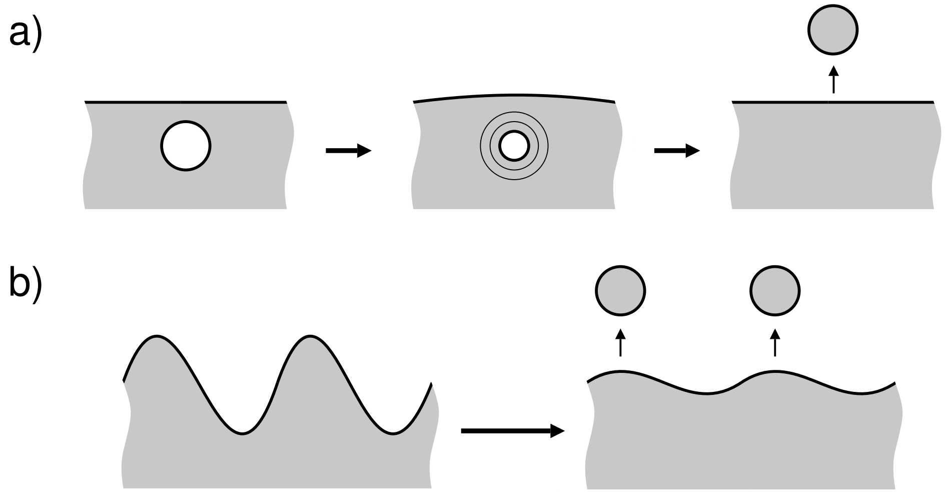

Figure 4.11: Two mechanisms for ultrasonics-induced spraying. a) Strong ultrasonics can cause cavitation in fluid just below the surface. The shockwaves produced by collapsing bubbles travel to the surface and lead to droplet ejection. b) Weaker ultrasonics cause the formation of surface waves. If the peaks of these waves become too narrow and sharply curved, it is energetically favorable for them to pinch off into droplets that are then ejected from the surface.

waves[73]:

$$\lambda = \left(\frac{8\pi\sigma}{\rho f^2}\right)^{\frac{1}{3}} \tag{4.3}$$

where λ is the capillary wavelength, σ is the surface tension of the fluid, ρ is the density of the fluid, and f is the frequency of the exciting sound. Experimentally, he found that a distribution of droplet sizes resulted, but that for a variety of fluids and sound frequencies the following relationship applies:

$$d_{droplet} = 0.34\lambda \tag{4.4}$$

where $d_{droplet}$ is the median diameter of the droplets ejected from the surface. For an applied ultrasonic frequency of 660 kHz and water as the fluid ($\rho = 1\frac{g}{cm^3}$, $\sigma = 73\frac{dyne}{cm^3}$), typical conditions for spraying from the microplotter, a droplet diameter of 16 μm is predicted. In a thorough study of ultrasonic spraying from a tapered-glass needle, Lee and Lal found that Equations 4.3 and 4.4 held until the needle outside diameter became less than $\frac{\lambda}{2}$.[74] In that case, a surface wave could not fully form but droplets were still being ejected. These droplets were found to have diameters approximately the same as that for the needle tip. In addition to measuring the diameter of ejected droplets, they found that the spraying rate (which is proportional to the droplet velocity) scaled linearly with the peak-to-peak voltage of the AC sent to the piezoelectric. Smaller needle tip diameters yielded higher droplet velocities due to the larger displacements of these tips when the needles were vibrated, something that will be touched on in the next section.

With further analysis of spraying from the tip of the microplotter dispenser, it should be possible to reduce the time it takes to manufacture microarrays. First, the automated process of switching solutions could be sped up if the rate of spray is known and / or controllable. Second, it may be possible to dispense in a completely non-contact mode using droplet ejection rather than gentle fluid pumping. The greatest drawback to the microplotter is its need to make fluid contact with a surface in order to deposit spots. If the surface wave breakup could be controlled well enough to be able to eject droplets on demand, then deposition of fluid could take place with no need to make any kind of contact with the surface. This could lead to deposition speeds comparable to inkjet printers or squeezed-capillary dispensers without the large spot sizes or dead volumes that hinder those technologies.

4.3 Acoustic pumping

The ejection of one fluid within the microplotter needle out into a second fluid (in the examples above, water into air) seems to be well explained by the breakup of surface waves, but we have also observed the pumping of a fluid into more of that same fluid. An example of this is shown in Figure 4.12. A dispenser was dipped into a well of a dilute silver colloid solution (to observe the fluid flow during pumping), with water as the solvent. Capillary action initially caused the solution to rise up within the dispenser, but as soon as the piezoelectric element was actuated at resonance with a high enough amplitude the fluid level dropped. A higher amplitude of vibration caused a faster lowering of the fluid level in the dispenser. Additionally, the application of ultrasonic vibrations seems to induce a change in the wetting properties on the outside of the dispenser, as indicated in Chapter 2.

Two types of vibrations within the glass micropipette may be responsible for the pumping action: longitudinal and transverse. These types of vibrations were described earlier in the context of vibrations within the piezoelectric. Longitudinal vibrations are

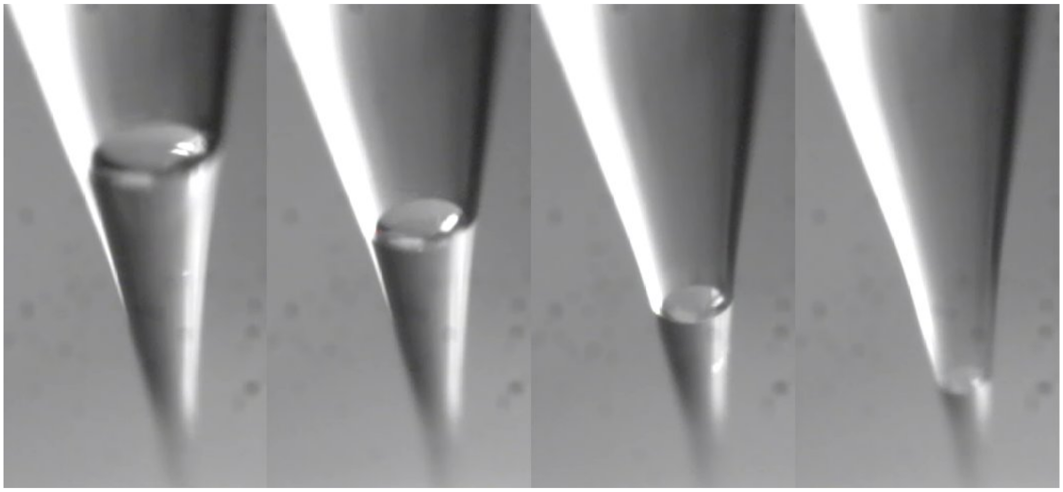

Figure 4.12: An example of a microplotter dispenser pumping liquid. A dispenser was dipped into a well containing DI water. Capillary action caused fluid to rise up in the glass needle portion of the dispenser, but as soon as ultrasonic vibrations of the correct frequency and strength were applied fluid began to be pumped out of the needle.

easier to model than are transverse vibrations, but, as shown in Section 4.1, the piezoelectric element that drives the micropipette appears to vibrate in both modes. Either or both of the types of vibrations may yield pumping behavior, and we will examine them separately.

4.3.1 Longitudinal vibrations

Longitudinal vibrations within the micropipette are ones in which the direction of vibration is parallel to the axis of the pipette. This class of vibrations is well-studied, especially in the case of simple systems. One such system, a vibrating plate, was thoroughly described by G.G. Stokes in 1850.[75] He provided analytical models for two cases, referred to as Stokes' First and Second Problems.[76]

Stokes' First Problem is the simpler of the two, dealing only with a plate that starts

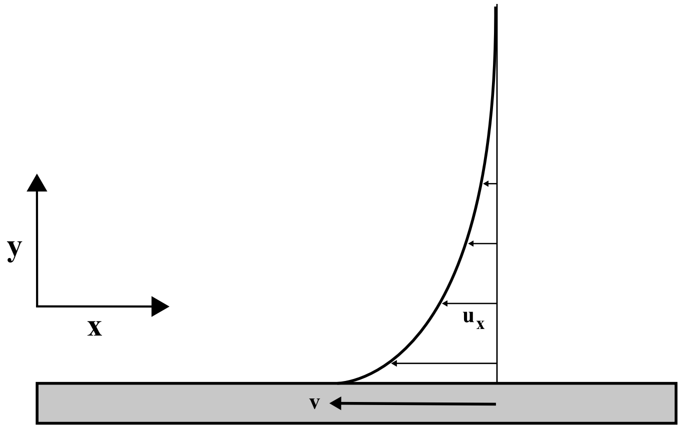

Figure 4.13: Stokes' First Problem. A flat plate starts moving at time $t = 0$, to the left in this example, dragging nearby fluid along. A distribution of fluid velocities develops, similar to the cross-section shown.

moving at a set velocity at time $t = 0$ (Figure 4.13. A gradient of fluid velocity develops, as the plate pulls the nearby fluid along at nearly the same velocity. A no-slip boundary condition is assumed, meaning that the fluid in direct contact with the plate is assumed to be moving at the same velocity as the plate. The bulk of the fluid is assumed to be at rest. If the x axis is parallel to the plate and the y axis is perpendicular to it, the following boundary conditions describe this problem:

$$u_x = v : y = 0, t > 0 \tag{4.5}$$

$$u_x = 0 : y \to \infty, t \geq 0 \tag{4.6}$$

$$u_x = 0 : 0 \leq y \leq \infty, t = 0 \tag{4.7}$$

The fade-off of fluid velocity from the plate outward is given by the following equation, which is the definition of viscosity:[77]

$$\frac{\delta u_x}{\delta t} = \nu \frac{\delta^2 u_x}{\delta y^2} \tag{4.8}$$

where $u_x(y,t)$ is the fluid velocity in the x direction at position y and time t and ν is the kinematic viscosity of the fluid. The kinematic viscosity is defined as $\nu = \frac{\mu}{\rho}$, where μ is the viscosity of the fluid and ρ is its density.

When this equation is solved with the boundary conditions above, the result is [76]

$$u_x(y,t) = v\left[1 - erf\left(\frac{y}{2\sqrt{\nu t}}\right)\right] \tag{4.9}$$

We employed a fluid modeling program (described in detail in Appendix A) to simulate the action of a fluid exposed to the conditions within a vibrating micropipette. In order to test this model, we performed a simulation of Stokes' First Problem and compared the results to that predicted by Equation 4.9. This comparison is shown in Figure 4.14 for three different times. The only fit parameter for the analytical curves was ν. Despite some noise in the results of the simulation, Figure 4.14 shows that the analytical curves match these results very closely, indicating that the fluid model correctly simulates this type of fluid behavior.

Stokes' Second Problem is much more relevant to the situation within a vibrating microplotter dispenser. In this case, a flat plate is vibrating with a maximum velocity amplitude of v_0 and an angular frequency of ω ($\omega = 2\pi f$), as illustrated in Figure 4.15. Equation 4.8 still governs the behavior of the fluid, but the boundary conditions for this problem now are

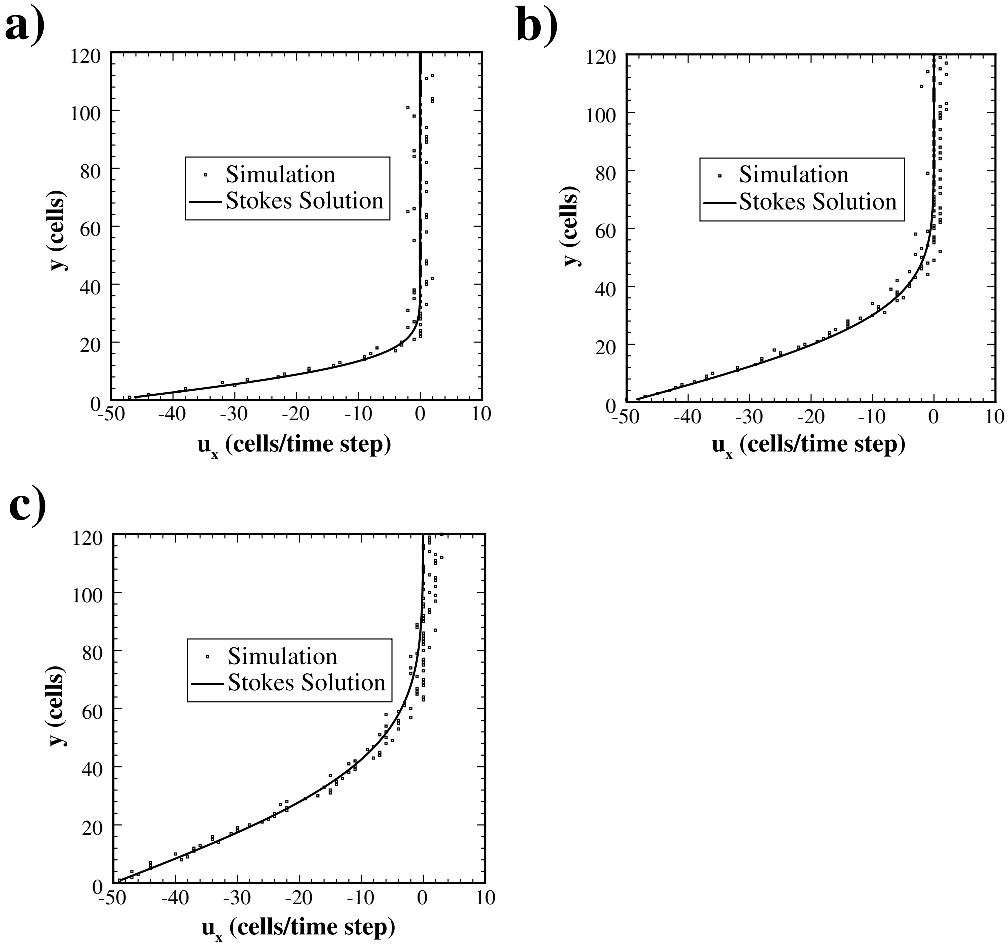

Figure 4.14: Analytical and computer modeling of Stokes' First Problem. A fluid modeling simulation of a plate moving at a constant velocity ($v = -50$ cells/time step) within a fluid was performed, with the results (dots) compared against an analytical solution to the problem (solid lines) for a) $t = 500$, b) $t = 2500$, and c) $t = 5000$ time steps. By fitting the analytical curves to the fluid model results, ν was found to be 0.11 cells^2/time step.

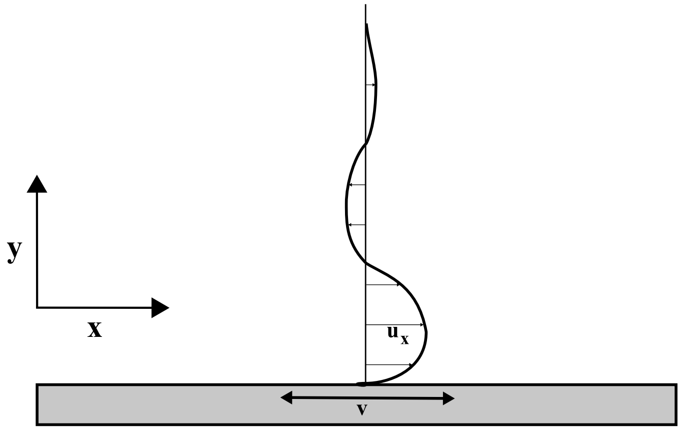

Figure 4.15: Stokes' Second Problem. A flat plate vibrates with amplitude v, inducing a velocity profile like the one shown.

$$u_x = v_0 \cos(\omega t) : y = 0, t > 0 \tag{4.10}$$

$$u_x \to 0 : y \to \infty, t > 0 \tag{4.11}$$

$$u_x = 0 : 0 \leq y \leq \infty, t = 0 \tag{4.12}$$

Solving Equation 4.8 with respect to these new conditions gives [76]

$$u_x(y,t) = v_0 e^{-\sqrt{\frac{\omega}{2\nu}}y} \cos\left(\omega t - \sqrt{\frac{\omega}{2\nu}}y\right) \tag{4.13}$$

Again, this was simulated using a fluid modeler and compared with the solution above. Figure 4.16 shows the results. Because ν was found in the curve fits for Stokes' First Problem, no fit parameters were used in these plots. Again, a very good match occurs between the fluid modeling simulation and the analytical prediction, seeming to indicate that the fluid modeler can handle vibrating walls.

There are some interesting aspects to this problem. First, it can be seen that the longitudinal vibration of the plate induces fluid movement only to a certain depth, due

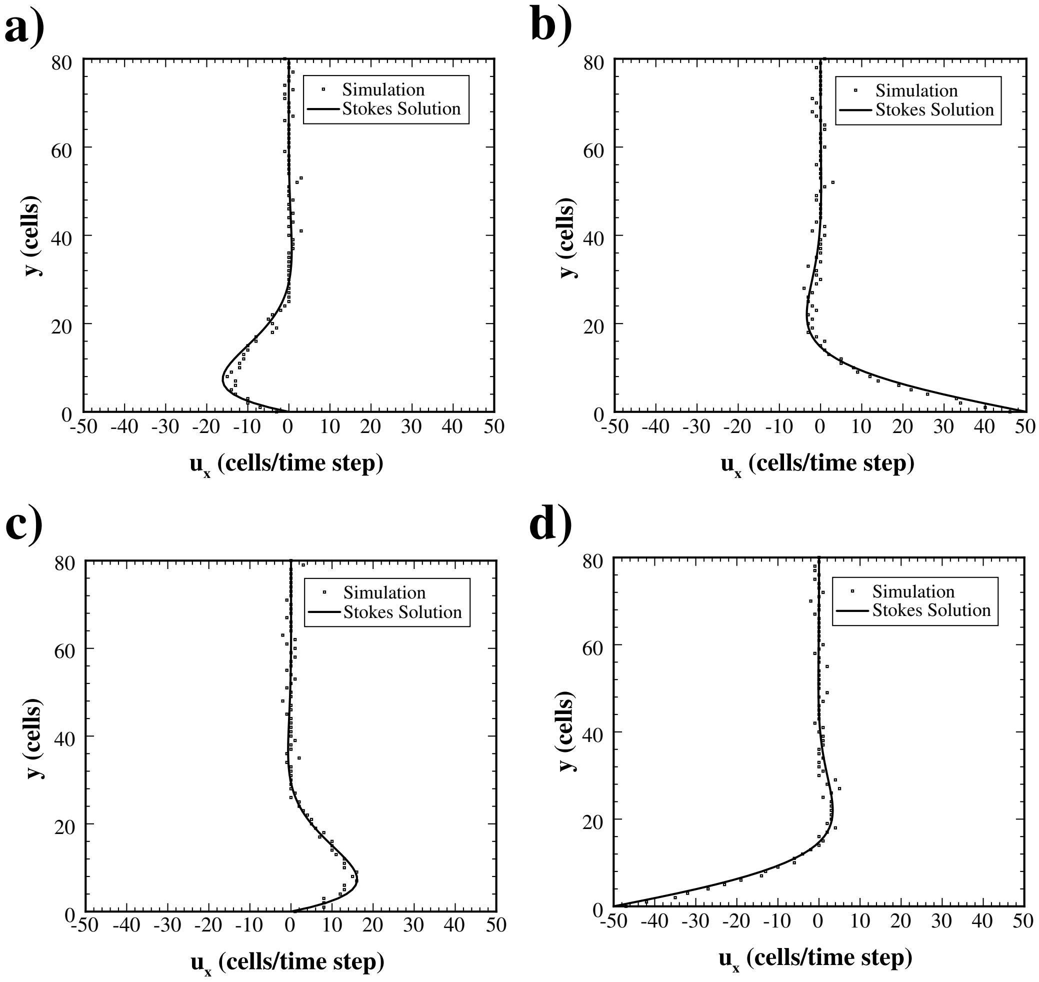

Figure 4.16: Analytical and computer modeling of Stokes' Second Problem. A fluid modeling simulation of a vibrating plate (having a maximum velocity amplitude of $v = 50$ cells/time step) within a fluid was performed, with the results (dots) compared against an analytical solution to the problem (solid lines) for a) $t = 13125$, b) $t = 13750$, c) $t = 14375$, and d) $t = 15000$ time steps. A value for ν of 0.11 cells^2/time step (the same as found earlier) and 0.0025 radians/time step for ω were used.

to the exponential fade-off term in Equation 4.13. Assuming the penetration depth of vibrations, also known as the Stokes layer, to be the distance away from the plate at which the vibrations have faded to 1% of their initial strength, this distance is given by

$$0.01 = e^{-\sqrt{\frac{\omega}{2\nu}}\delta} \tag{4.14}$$

$$\delta = \sqrt{\frac{42.4\nu}{\omega}} \tag{4.15}$$

In Figure 4.16, δ=43.1 cells. For a micropipette vibrating at 660 kHz with water as the fluid, δ=3.2 μm. Others have used a 1/e penetration depth[61], for which the solution is $\delta = \sqrt{\frac{2\nu}{\omega}}$, but the vibrating plate still has a strong effect on fluid velocities at that distance. For a given δ, decreasing the inside diameter of the micropipette tip increases the fraction of the fluid within that is affected by the wall vibrations, as shown in Figure 4.17. This fraction is given by

$$f_{vib} = \frac{2r_{tip}\delta - \delta^2}{r_{tip}^2} \tag{4.16}$$

where r_{tip} is the inside radius at the tip of the micropipette.

Another observation that can be made from the solution to Stokes' Second Problem is that a large amount of shear occurs in the fluid near a vibrating plate or wall. The shear rate within a fluid is defined as the local velocity gradient[77], and the following expression can be used to express the magnitude of that shear as a function of time and distance from the vibrating surface:

$$\begin{aligned} shear\ rate &= \left| \frac{du_x}{dy}\Big|_t \right| \\ &= \left| v_0\sqrt{\frac{\omega}{2\nu}} e^{-\sqrt{\frac{\omega}{2\nu}}y} \cos\left(\omega t - \sqrt{\frac{\omega}{2\nu}}y\right) \right| \end{aligned} \tag{4.17}$$

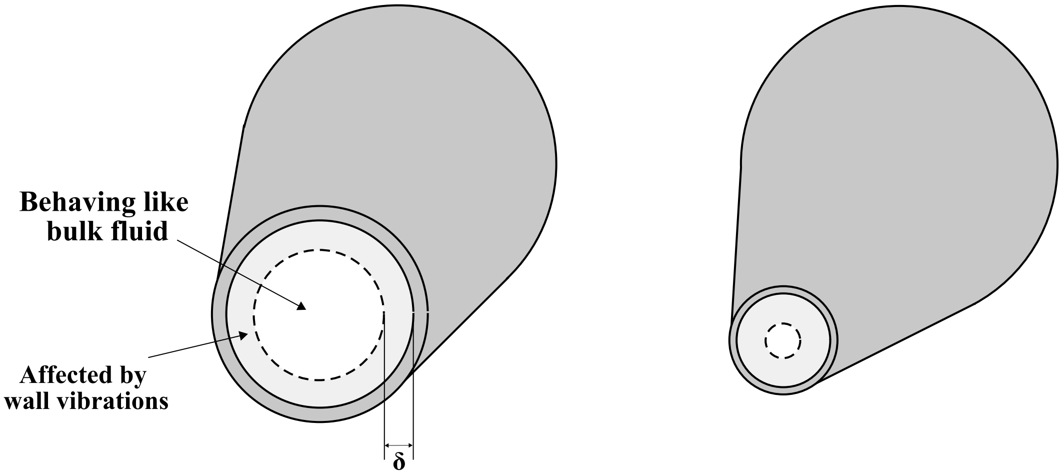

Figure 4.17: Schematic illustrating the penetration depth of vibrations. The tip of the microplotter needle is shown, where it can be seen that for a given penetration depth δ of longitudinal wall vibrations into the fluid, reducing the inside diameter of the tip increases the fraction of fluid that is affected by wall vibrations.

Figure 4.18 illustrates the calculated shear rate profile as a function of distance from a vibrating plate for conditions similar to those present in a microplotter dispenser. The velocity amplitude for longitudinal vibrations of 100 mm/s is a value that has been measured at the tip of a similar device.[74] Under these conditions, a shear rate as high as 144000 s^{-1} develops near the vibrating wall. As seen in Figure 4.17, a significant portion of the fluid passing through the tip of the vibrating needle could be subjected to this high shear.

Proteins, an important class of biomolecules that we would like to deposit with a microplotter dispenser, can have their three-dimensional structure (conformation) altered by high shear.[78, 79] Shear rates over 10000 s^{-1} can produce conformational changes that can deactivate more sensitive proteins[78], a rate that is present up to 0.9 μm away from the vibrating wall. While shear is not as well known for denaturing proteins as pH,

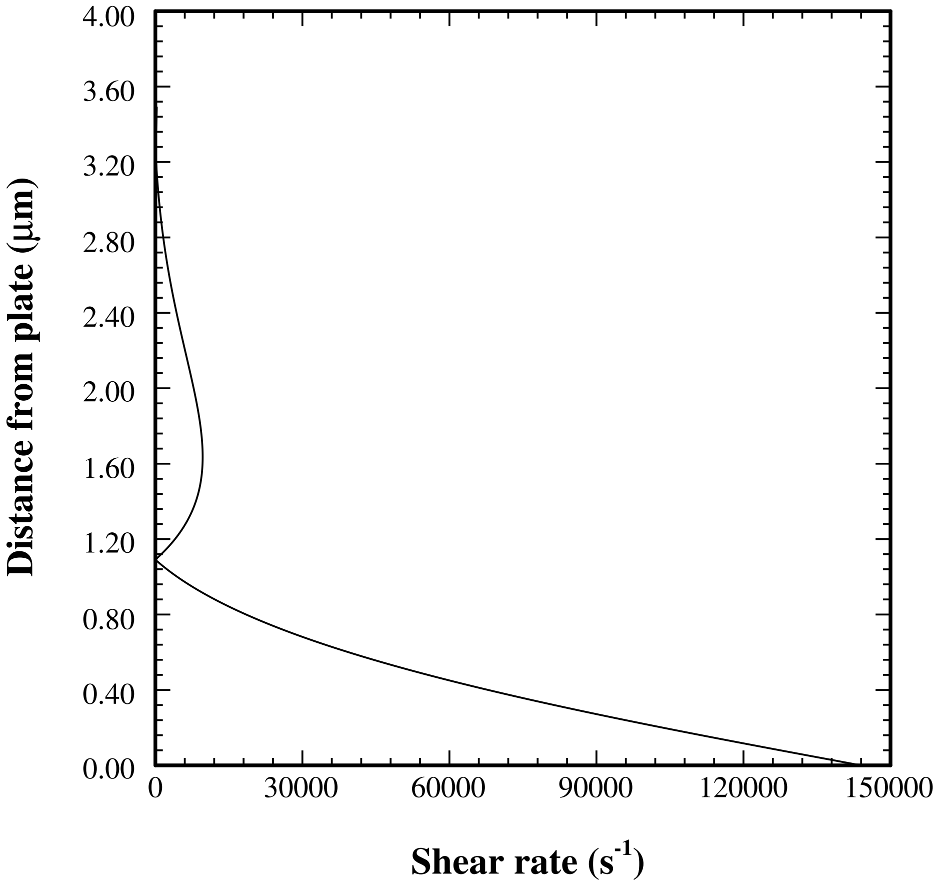

Figure 4.18: Shear rate due to longitudinal vibrations. This is a plot of Equation 4.17 for water as the fluid ($\nu = 1\text{x}10^{-6}\ m^2/s$ at 20°C) with a plate vibrating at 660000 Hz and a maximum amplitude of 100 mm/s, typical conditions for a microplotter dispenser. The shear rate profile varies throughout one half cycle of vibration, and this plot is a snapshot taken at the moment when the shear rate is strongest.

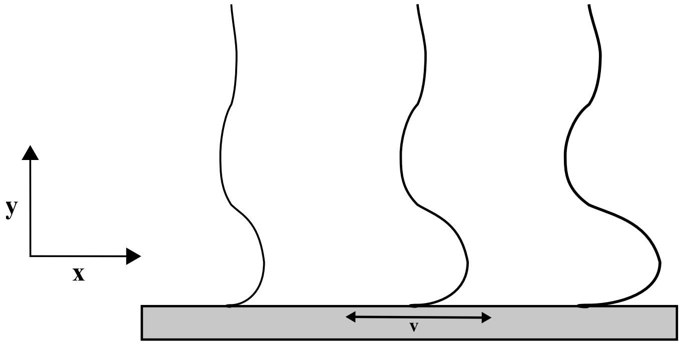

Figure 4.19: Schematic of the fluid modeling simulation of a gradient of vibrational amplitudes. A fluid modeling simulation was set up with a solid plate vibrating in the manner shown, with the amplitude of the vibration tapering off from the right end of the plate to the left end.

temperature, or dehydration, we will investigate any effects of shear on labile proteins in Chapter 6.

The solution to Stokes' Second Problem, while giving some insights into the behavior of a longitudinally vibrating wall, does not explain the pumping action we have observed within the microplotter dispenser. A plate that moves as a solid unit back and forth symmetrically imparts no net momentum to the fluid and thus cannot cause a directed pumping action. Even if a gradient of vibrational amplitude, such as that shown in Figure 4.19, is used we find that no pumping occurs. Figure 4.20b shows the results of a fluid modeling simulation where the vibrational amplitude was reduced linearly from the right end of the plate to the left ($v_0(x) = C_1x + C_2$). The plot is of the velocity profile of the fluid as a function of distance from the vibrating plate, averaged over four vibrational cycles. No net velocity of the fluid is noted.

In reality, the glass wall of the vibrating micropipette does not move as a solid unit. Vibrations propagate through the glass starting from the area where the piezoelectric

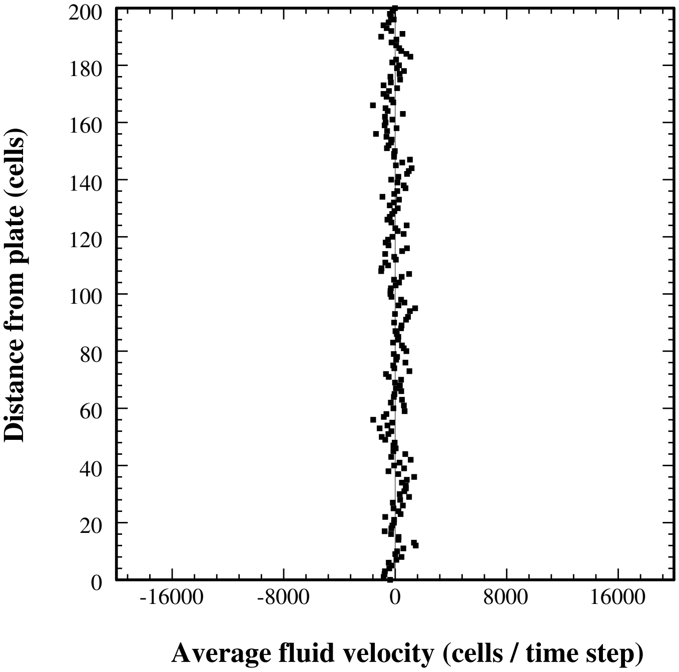

Figure 4.20: The results of the fluid modeling simulation for a plate vibrating with a gradient of vibrational amplitudes. The plot is of the fluid velocity in the x direction averaged over four vibrational cycles, showing that no net movement of the fluid occurs.

element is bound. To analytically model longitudinal waves travelling through the glass cone of the micropipette dispenser, we will used a modified form of the equation for longitudinal vibrations in a solid bar. The following derivation of that equation has been adapted from *Mechanical Vibrations*.[80]

When a beam is vibrating longitudinally, the displacement of each point x in the beam (along the x axis) at time t is given by the function $\xi(x, t)$. The tensile stress caused by the vibration is

$$E\frac{\partial \xi}{\partial x} \tag{4.18}$$

with E as the modulus of elasticity.

The forces around x are not balanced, and an excess is present in the positive x direction of

$$F = \frac{\partial}{\partial x}\left(EA(x)\frac{\partial \xi}{\partial x}\right)dx \tag{4.19}$$

where $A(x)$ is the cross-sectional area of the beam as a function of position.

The acceleration experienced by a segment dx of the vibrating beam is

$$a = \frac{\partial^2 \xi}{\partial t^2} \tag{4.20}$$

and the mass of that segment is

$$m = \rho A(x)dx \tag{4.21}$$

where ρ is the density of the beam material. Therefore, the force experienced by the segment dx is

$$F = \rho A(x)dx\frac{\partial^2 \xi}{\partial t^2} \tag{4.22}$$

By setting Equation 4.19 equal to Equation 4.22, we arrive at

$$\begin{aligned}\rho A(x) dx \frac{\partial^2 \xi}{\partial t^2} &= \frac{\partial}{\partial x}\left(EA(x)\frac{\partial \xi}{\partial x}\right) dx \\ \rho A(x) \frac{\partial^2 \xi}{\partial t^2} &= \frac{\partial}{\partial x}\left(EA(x)\frac{\partial \xi}{\partial x}\right) \qquad (4.23)\end{aligned}$$

which is the general form of the equation for a beam vibrating longitudinally. Normally, this is presented with $A(x)$ being constant, but that is not the case in the conical head of the microplotter dispenser. With some algebraic manipulation, this equation becomes

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial \ln A(x)}{\partial x}\frac{\partial \xi}{\partial x} = \frac{1}{c^2}\frac{\partial^2 \xi}{\partial t^2} \qquad (4.24)$$

with c, the speed of sound in the beam material, equal to $\sqrt{\frac{E}{\rho}}$. Both Equations 4.23 and 4.24 have been referred to as Webster's horn equation.[81]

An analytical solution for $\xi(x,t)$ in Equation 4.23 for a solid cone has been arrived at by several researchers, under the condition that the cone was in resonance.[82, 83] First, ξ can be expressed as a function of time by

$$\xi(x,t) = \xi(x) \sin \omega t \qquad (4.25)$$

with $\xi(x)$ being the amplitude of displacement at a position x along the axis of the cone. The physical meaning of x is shown in Figure 4.21, along with the other dimensions used in this solution. Using this substitution, Equation 4.24 becomes

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial \ln A(x)}{\partial x}\frac{\partial \xi}{\partial x} + \rho\left(\frac{\omega^2}{E}\right)\xi(x) = 0 \qquad (4.26)$$

The solution to this is expressed in terms of dimensionless units.[82] These are

$$T = 1 - \sqrt{\frac{A(L)}{A(0)}} \qquad (4.27)$$

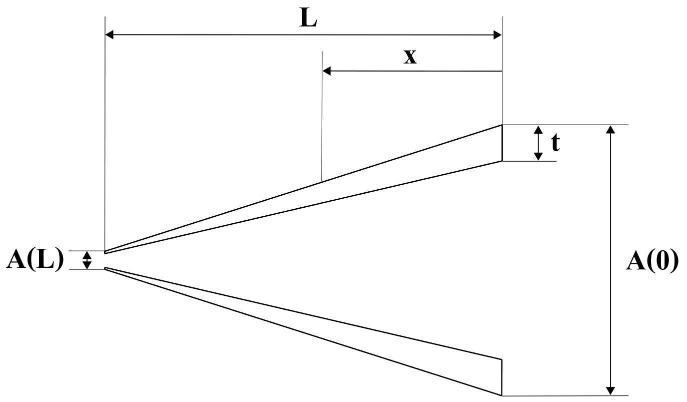

Figure 4.21: Schematic of a hollow cone. The dimensions referred to in the analytical solution to Equation 4.23 are shown, with the exception of $A(0)$ and $A(L)$, whose positions are only indicated.

$$\Omega = \omega L\sqrt{\frac{\rho}{E}} \tag{4.28}$$

$$X = \frac{x}{L} \tag{4.29}$$

where $A(0)$ is the cross-sectional area of the cone at its base, $A(L)$ is the cross-sectional area at its tip, and L is the length of the cone.

The amplitude of vibration as a function of position in the cone is then

$$\xi(x) = \frac{\xi(0)}{1-TX}\left[\cos(\Omega X) - \frac{T}{\Omega}\sin(\Omega X)\right] \tag{4.30}$$

with $\xi(0)$ being the amplitude of the vibrations applied at the base of the cone.

Even though Equation 4.30 was derived for a solid cone, it also accurately describes a hollow cone with walls that taper at the same rate as the overall cone. The condition that must be met for Equation 4.30 to be valid is

$$\sqrt{A(x)} = \sqrt{A(0)} - \frac{x\left[\sqrt{A(0)} - \sqrt{A(L)}\right]}{L} \tag{4.31}$$

The cross-sectional area of a slice of a hollow cone is

$$A = \pi \left(Dt - t^2 \right) \tag{4.32}$$

where D is the outside diameter of the circular slice and t is the wall thickness. If D and t scale at the same rate, the condition of Equation 4.31 is satisfied. This is the case in a hollow needle made by heating and pulling a glass capillary.

Figure 4.22 illustrates the distortion of the profile of a glass cone as it vibrates. The displacements were calculated based on the properties of borosilicate glass, $\rho = 2.23$ g/cm^3 and $E = 64$ GPa[84], a frequency of 954 kHz, a length of the cone of 4 mm, an outside diameter at the base of 1 mm and at the tip of 4 μm, and a wall thickness of 100 μm (typical dimensions for the pulled glass needles used to date). A $\xi(0)$ of 100 μm was used, resulting in a displacement far larger than the 0.5 μm previously seen in this type of needle[74], simply to make the distortion easily observable.

Note that $|\xi(L)| > |\xi(0)|$. The magnification of vibrational displacements from the base of the cone to the tip is at a maximum at the cone's resonant frequency. For a cone, the amplitude of vibrations at the tip can be up to 4.61 times that of the vibrations at its base.[82] Horns with non-conical shapes have been designed to achieve higher amplification factors.[85, 86]

A frequency of 954 kHz was used for this example because it is the resonant frequency of a cone with the above dimensions. At resonance, the cone obeys the following equation:

$$\tan \Omega = \frac{\Omega T^2}{\Omega^2 (1 - T) + T^2} \tag{4.33}$$

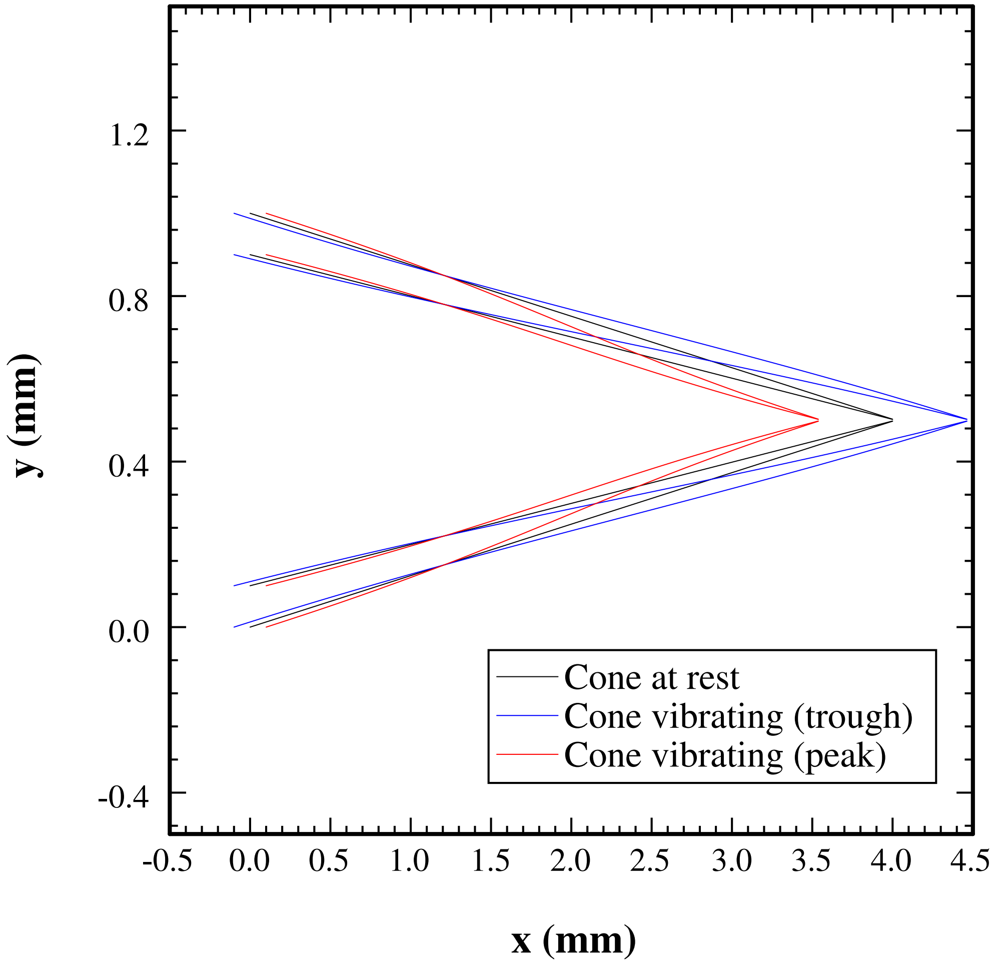

Figure 4.22: Cross-section of a longitudinally vibrating hollow cone. The original profile of a tapered cone, similar to one at the tip of a microplotter dispenser, is shown in black, and the profile displaced by longitudinal vibrations in red (peak) and blue (trough). The displacement shown is what occurs at longitudinal resonance, with a $\xi(0)$ of 100 μm.

Solved numerically, the resonant frequency can be calculated from

$$f \approx \frac{0.712}{L}\sqrt{\frac{E}{\rho}} \tag{4.34}$$

This holds as long as $A(0) \gg A(L)$.

954 kHz is a much higher frequency than the ∼660 kHz standard operating frequency for the microplotter dispensers. This may be further evidence that, as suggested in Section 4.1, the critical resonance frequencies are those for the piezoelectric elements. If the vibrational frequency is close to that for resonance within the cone, it will still distort in the fashion shown in Figure 4.22, only with a smaller magnification of vibrations from the base to the tip. The difference in frequency may also be due to the fact that the piezoelectric element does not directly contact the cone, but instead must pass vibrations through a short section of the cylindrical part of the needle. The resonance frequency of the combined cylinder/cone may be closer to 660 kHz.

We have described how the cone at the tip of a glass needle distorts when resonating longitudinally, but have not suggested how this could bring about pumping. A possible explanation comes from the change in volume within the cone caused by the change in shape shown in Figure 4.22. The volume within the cone at time t can be found by

$$\int_{\xi(0,t)}^{L+\xi(L,t)} \pi(r(x-\xi(x,t)))^2 dx \tag{4.35}$$

where r is the inside radius of the cone. It is of the form

$$r(x) = r(0) - x\frac{r(0)-r(L)}{L} \tag{4.36}$$

with $r(0)$ being the radius at the base of the cone, $r(L)$ being the radius at the tip, and x the distance away from the base.

Solving Equation 4.35 numerically for the glass cone of Figure 4.22 gives a difference in volume of the cone from peak to trough of 15.7 pL at an amplitude of incident vibrations of 0.01 μm and 157 pL at an amplitude of 0.1 μm. The change in volume scales linearly with the amplitude of incident vibrations. The difference in volume over a vibrational cycle is constant with changes in taper length, but the total volume of fluid in the cone (at rest) varies linearly with taper length. This means that a greater portion of the total fluid inside the cone is either forced out of, or drawn into, the cone as it vibrates. Finally, a larger volume change in proportion to the total volume of the cone occurs as the aperture at the tip of the cone gets smaller.

The reduction in volume that occurs as the incident vibrations go from trough to peak compresses the fluid within the cone, causing a buildup of pressure. The taper of the walls in the cone directs the fluid under pressure to be ejected from the aperture at the tip of the cone. During the other half of a stroke, when the incident vibrations go from peak to trough, the volume within the cone increases as it elongates. This leads to lower pressure within the cone, causing fluid to be pulled into it. However, it is not easy for fluid to return through the small opening at the tip, as the outside walls of the cone are tapered away from it. Fluid is more likely to be supplied from the wide end of the needle. A schematic of this process is shown in Figure 4.23.

Not all of the difference in volume results in ejected fluid. The calculated differences in volume per cycle, if they were translated to pumping rates, would be far larger than what we have observed in dispensers with this configuration. A large portion of the fluid under pressure during the compression of the cone may travel back out of the wide end of the cone. Some fluid may come back into the needle through the tip as the cone expands. However, enough of a net transport of fluid out through the tip of the needle

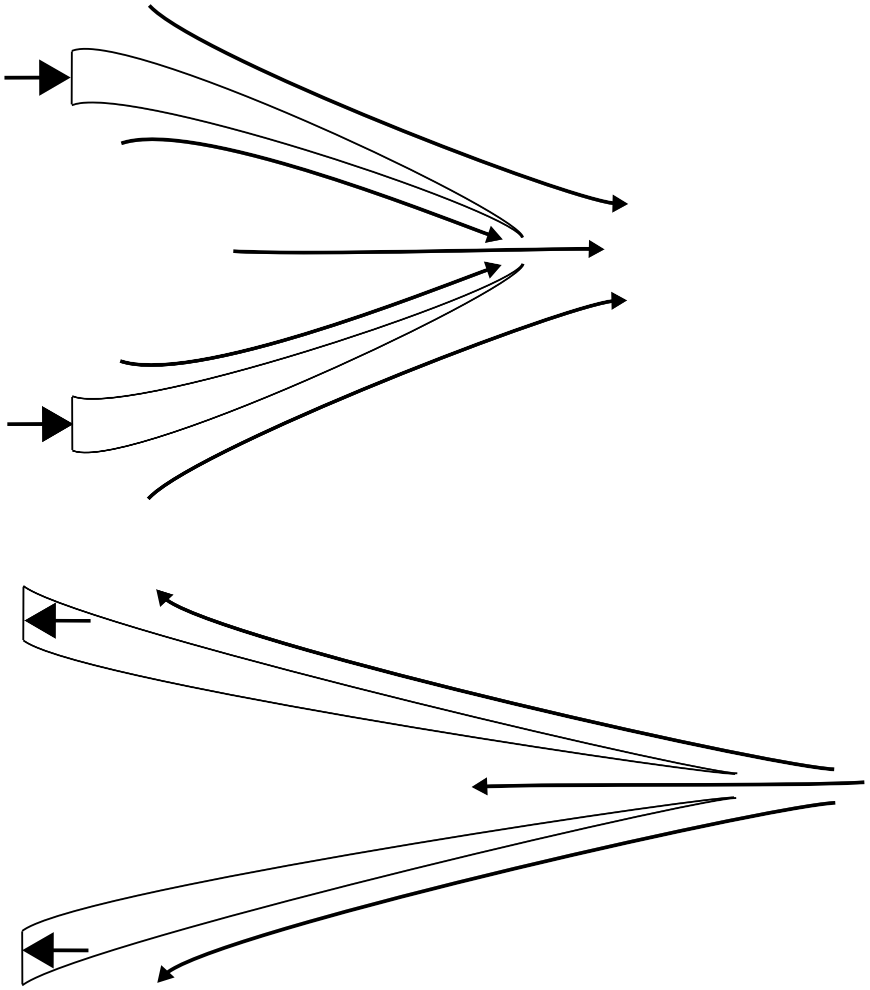

Figure 4.23: Fluid displacement caused by the change in cone shape with resonant vibrations. As the cone vibrates, it acts as a moving valve, forcing fluid out of the cone at the peak of incoming vibrations while only allowing a small amount back in at the trough.

occurs for a constant pumping action to be observed.

The ejection of fluid due to the changing volume of the cone as it vibrates was suggested by Lee and Lal in order to account for the spraying observed when needle tip inside diameters became smaller than half the wavelength of standing waves on the needle tip surface.[74] Indeed, the linear change in volume as a function of amplitude if incident vibrations matches their observation that the spraying rate scales linearly with the peak-to-peak voltage of the AC applied to the piezoelectric actuator. In Chapter 3, Section 3.3 we did not observe such a linear scaling with the volume of fluid in deposited spots, but spot volumes are not just determined by pumping rate and may be significantly affected by surface tension. Additionally, that study may not have had a large enough sample size. That a greater proportion of the cone volume changes with a shortening of the cone is consistent with our observations that a shorter taper on the pulled tip of a microplotter dispenser yields a stronger pumping action.

We believe that this is the most likely explanation for acoustic pumping due to longitudinal vibrations in the needle. Other pumping phenomena, such as acoustic streaming or peristaltic pumping have been observed in different systems, but we believe these are not primary actors in this system.[87, 88] In those cases, the wavelengths of the travelling vibrations are much shorter than the size of the system, so multiple periods of vibration occur along the dimension in the direction in which fluid movement occurs.

Work is ongoing to attempt to verify this pumping mechanism, both experimentally and through computer simulations.

4.3.2 Transverse vibrations

Above, we presented an analytical derivation for longitudinal vibrations in a hollow cone. Longitudinal vibrations appear to cause a change in volume of the cone by distorting its shape. The asymmetry of the cone leads to the net movement of fluid in one direction during a vibrational cycle. However, the needle of the microplotter does not just vibrate longitudinally. We have observed significant transverse vibrations in needles with very long tapers and very thin sidewalls. This is consistent with the resonance analysis we performed in Section 4.1, where we saw that transverse vibrations may be coming from the piezoelectric actuator in addition to longitudinal vibrations.

Transverse vibrations are more difficult to examine analytically than longitudinal ones. The equation governing transverse, or bending mode, vibrations in a beam is [80]

$$\frac{\partial^2}{\partial x^2}\left(EI(x)\frac{\partial^2 y}{\partial x^2}\right) = -\rho A(x)\frac{\partial^2 y}{\partial t^2} \tag{4.37}$$

where $I(x)$ is the moment of inertia of the slice of the beam at x.

Unfortunately, we know of no analytical solution to this equation for the case of a hollow cone. Numerical solutions have been found for the case of a solid cone, but those may not apply in this case.[89, 90]

With transverse vibrations, the same sort of volume change seen in the case of longitudinal vibrations does not occur. However, much as large waves can propel a boat forward, transverse waves in a solid wall can impart momentum to the contacting fluid. [60, 91] The impact of transverse vibrations on the pumping process warrants further investigation.

4.4 Conclusion

In this chapter, we examined the fluid pumping and spraying behavior of the microplotter dispenser. Impedance measurements were used to look at the resonances within the piezoelectric element used to drive the dispenser, and how those resonances shift with the bonding of a tapered needle to the piezoelectric and the filling of the needle with fluid. We observed that the piezoelectric resonances matched the frequencies at which fluid spraying occurs. Fluid spraying was indicated to be most likely from the breakup of capillary waves on the surface of fluid at the tip of the dispenser. Finally, we analytically described the distortion of the conical portion of the microplotter dispenser when it vibrates longitudinally at resonance. We demonstrated how this distortion could lead to a pumping action by forcing fluid out from inside the dispenser in one half of the vibration cycle and preventing its reuptake in the other.

Chapter 5

Cold-plasma modification of oxide surfaces for covalent biomolecule attachment

While many processes have been developed to modify the surface of glass and other oxides for biomolecule attachment, they rely primarily upon wet chemistry and are costly and time-consuming. We describe a process that uses a cold plasma and a subsequent *in vacuo* vapor-phase reaction to terminate a variety of oxide surfaces with epoxide chemical groups. These epoxide groups can react with amine-containing biomolecules, such as proteins and modified oligonucleotides, to form strong covalent linkages between the biomolecules and the treated surface. The use of a plasma activation step followed by an *in vacuo* vapor-phase reaction allows for the precise control of surface functional groups, rather than the mixture of functionalities normally produced. By maintaining the samples under vacuum throughout the process, adsorption of contaminants is effectively eliminated. This process modifies a range of different oxide surfaces, is fast, consumes a minimal amount of reagents, and produces attachment densities for bound biomolecules that are comparable to or better than commercially available substrates.

5.1 Introduction

Biological microarrays, composed of spots of different biomolecules attached to a substrate at known locations, are fabricated principally in two ways: synthesis directly on the substrate through photolithographic or chemical means [4, 10, 12] or deposition of presynthesized or extracted molecules onto a chemically treated substrate that binds them [23]. The latter way is required for those molecules that cannot currently be synthesized directly on a substrate, for example, many proteins [92] or carbohydrates [56].

Attachment to a substrate, either electrostatically or covalently, is critical to the production of microarrays using presynthesized or extracted biomolecules. Covalent attachment is preferred, because attached molecules remain bound to the substrate after exposure to target solutions, multiple wash steps, and even repeated use of the microarray. Surface chemical functionalization processes have been developed to bind biomolecules covalently to a range of substrate materials, including glass [93], silicon [37], and diamond [38]. Most of these processes rely on the use of wet chemistry, in many cases consuming significant volumes of rare, expensive, and/or environmentally unfriendly chemicals. Wet-chemical treatments may take anywhere from several hours to a day to produce a substrate ready for biomolecule attachment.[39, 94]

Cold plasmas can greatly reduce the complexity of preparing a chemically functionalized surface. A cold (non-equilibrium, near-room-temperature) plasma is an ionized gas produced by applying an electric field to create a discharge in a vapor. An ionized gas permits the modification of even the most chemically resistant surfaces. We describe here the use of cold plasmas to modify the surfaces of oxides. Oxides are of particular interest as substrate materials for biomolecule attachment because of the wide use of glass slides in existing microarrays and the reflectivity in the visible spectrum of metals with native oxides, which can enhance the fluorescent signal from a microarray. Specifically, we create amine-reactive epoxide chemical functionalities on oxide surfaces.

Several prior efforts have been made to use cold plasmas for functionalizing surfaces. The use of cold plasmas of reactive vapors to create specific functionalities on surfaces has not produced exceptional results, primarily because the mixture of different chemical fragments and charge states found in a plasma formed from reactive vapors can produce a mixture of different chemical terminations on the surface.[52] In one study, the attachment of different fragments on the surface was avoided by using the plasma only to activate the surface, which was then withdrawn from vacuum and exposed to a chemical bath to introduce the desired chemical functionalities.[95] This procedure, however, exposes the surface to atmosphere and the consequent contamination and deactivation of many of the active sites. The use of a chemical bath requires large quantities of chemicals and hinders the ability of the process to scale up. We resolve these issues by carrying out our entire process in vacuum, but using the cold plasma only to activate the surface and then using a *vapor* of the reactive species we want to attach on the surface.

5.2 Materials and methods

5.2.1 Plasma treatment

All of the steps in our treatment take place in a 15” diameter parallel-plate plasma reactor operated at a base pressure of 30 mTorr. The discharge is produced by a 30-kHz power supply. After each plasma treatment or vapor exposure, the chamber is again pumped down to 30 mTorr. The first step consists of a two-minute-long oxygen plasma clean at 200 mTorr and 200 W, followed by an argon plasma exposure for the same time at the same conditions. Oxygen and argon plasmas are commonly used in semiconductor wafer cleaning for oxidizing and stripping photoresist and other adsorbed organics off silicon wafer surfaces.[45] By not allowing the plasma-cleaned substrates to see atmosphere until after treatment is completed, chemical functionalization will be unaffected by atmospheric contaminants. Existing wet-chemical procedures require baths in strong acids or bases and strong oxidizers [96] for cleaning and run the risk of surface contamination when the cleaned substrates are exposed to atmosphere between cleaning and treatment steps. FisherBrand Precleaned microscope slides were used as our glass substrates and no additional chemical cleaning was performed on our substrates prior to the plasma clean.

Following this surface preparation step, the substrates are exposed to a plasma of a 2:1 water / oxygen mixture. The purpose of this step is to terminate the surface with hydroxyl functional groups. The water / oxygen plasma is formed at a pressure of 200 mTorr and at 200 W for two minutes.

As the final reaction step, a 1-Torr vapor of epichlorohydrin (with no plasma) is introduced to the substrates and allowed to react for at least 30 minutes. Epichlorohydrin (C_3H_5ClO) is a relatively volatile liquid (vapor pressure of $\sim$ 16 mmHg at 25 °C) composed of a single epoxide ring. A vapor is supplied to the chamber by opening a valve to a container of the liquid and allowing enough to evaporate to fill the chamber to a pressure of 1 Torr. As illustrated in Figure 5.1, the epichlorohydrin epoxide ring reacts with the nucleophilic oxygen species introduced to the oxide surfaces, causing the ring to open and binding the epichlorohydrin to the surface.[97] The ease with which

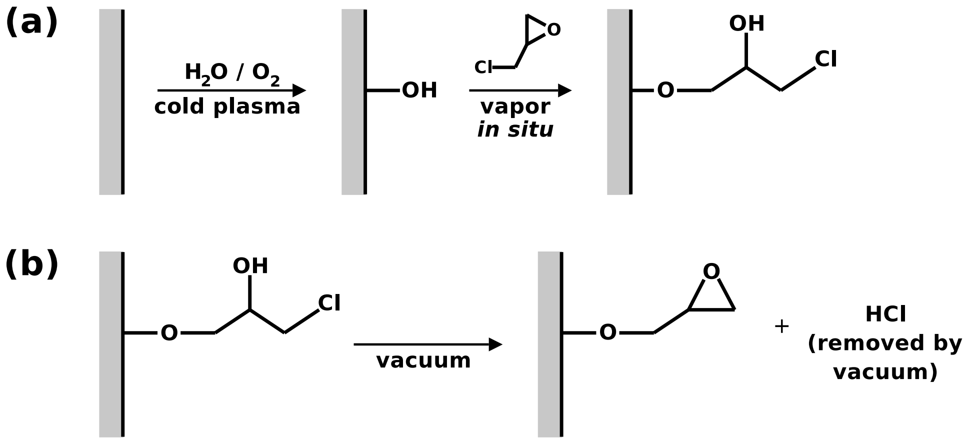

Figure 5.1: Proposed reaction mechanism for the described treatment process. a) First, the surface is terminated with hydroxyls by a water / oxygen plasma. The hydroxyls on the surface react with the epoxide ring of epichlorohydrin introduced as a vapor, covalently binding the epichlorohydrin to the surface. b) Finally, exposure to vacuum removes HCl and allows the reforming of the epoxide on the free end of the epichlorohydrin, which epoxide-terminates the surface.

the chlorine group can leave, a unique property of epichlorohydrin, allows for the epoxide ring of epichlorohydrin to reform on the free end of the bound molecule. In epoxy polymerization a strong base is normally used to draw off the HCl produced by the ring reformation.[98] In our process, we evacuate the epichlorohydrin and pump off the HCl in vacuum at room temperature for another 30 minutes. The epoxide-terminated substrates are then removed from the reactor and stored in dry conditions (<10% humidity), under which they retain their full activity for at least several months.

5.2.2 X-ray photoelectron spectroscopy

We used a PerkinElmer Physical Electronics 5400 Small Area System X-ray photoelectron spectroscopy (XPS) system with a Mg source at 15 kV and 300 W and a pass energy of 89.45 eV. In order to account for sample charging, we sputtered gold onto a portion of each sample and used the Au 4f$_{7/2}$ peak as a reference. We compared a plasma-cleaned glass slide (exposed to ambient air) and a glass slide completely treated with our plasma process with a commercially available epoxide-terminated glass slide (TeleChem International, Inc.).

5.2.3 Contact angles

We used a DataPhysics Contact Angle System OCA Plus 15 goniometer loaded with distilled water to measure sessile contact angles on treated, untreated, and commercially available substrates. Five locations were examined on each sample, with the average value being reported.

5.2.4 Atomic force microscopy

We use a Digital Instruments Multimode atomic force microscope (AFM) in intermittent-contact mode to image the surfaces of untreated, treated, and commercially available substrates, as well as determine the root-mean-square (RMS) roughnesses of those surfaces.

5.2.5 Oligonucleotide deposition, immobilization, and detection

16-mer oligonucleotides were C6-amino-modified on their 5’ end and fluorescein-modified on their 3’ end. All oligos used in these studies were synthesized by the University of Wisconsin Biotechnology Center. These oligos were diluted to a 200 μM concentration in distilled water and deposited in 0.5 μL spots on the treated substrates. The amine-modified oligos were allowed to bind for 2 hours in dry conditions, then a concentrated solution of 99% ethanolamine was spread evenly across the substrates underneath a coverslip and allowed to react for 30 minutes in order to block the remaining epoxide groups present on the surface. The substrates were rinsed with ethanol and distilled water and placed in a 2X SSPE buffer solution (20 mM NaH_2PO_4, 300 mM NaCl, 2 mM EDTA) for one hour in order to remove nonspecifically bound oligos. The substrates were then placed in a bath of distilled water for approximately one hour. Before analyzing them in a Genomic Solutions GeneTAC UC4x4 microarray scanner, we spread a 1 M Tris-HCl buffer solution across the surface of the substrates to enhance the fluorescence of the fluorescein dye. We quantified fluorescence from the spots of bound oligonucleotides by finding the average fluorescence within an entire spot using the NIH ImageJ [64] analysis software and then averaging that value across 12 spots on a slide.

5.2.6 Oligonucleotide hybridization

Two different 31-mer sequences were immobilized on a glass slide treated with our process, 5’ amino C6-T_{15}-GC TTA AGG AAG GTT CG-3’ (sequence A) and 5’ amino C6-T_{15}-GC TTA ACC ACC ATT CG-3’ (sequence B), and hybridized with their complements, 5’ (6-FAM)-CG AAC CTT CCT TAA GC-3’ (complement A) and 5’ (6-FAM)-CG AAT GGT GGT TAA GC-3’ (complement B). We deposited a set of spots of sequence A along with a set of spots of sequence B, both on the same treated substrate, which had been subjected to the previously described binding, blocking, and washing procedure. For hybridization, the complements were diluted to a concentration of 10 μM in a buffer solution of 2X SSPE / 0.2% SDS (20 mM NaH_2PO_4, 300 mM NaCl, 2 mM EDTA, 7mM sodium dodecylsulfate). The fluorescently tagged complements were allowed to hybridize with the surface-bound oligos at room temperature for one hour, then the substrates were washed for 15 minutes with 2X SSPE/0.2% SDS and dried. The substrates were scanned in the fashion described earlier.

5.3 Results and discussion

5.3.1 Surface characterization

We investigated the surface elemental composition of the treated substrates with XPS. High-resolution spectra of the C 1s peak and an elemental analysis are shown in Figure 5.2 and Table 5.1, respectively. Table 5.1 shows an increase in the relative amounts of Si and O in the glass surface at the expense of C after a plasma clean, indicating that adsorbed organics had been removed. The exposure to epichlorohydrin increases C on the surface and introduces a small amount of Cl, presumably from epichlorohydrin that has not yet reformed the epoxide ring during the 30 minute vacuum HCl removal. In the substrate exposed to the full plasma treatment the C-O component of the C 1s peak is greatly enhanced relative to the C-C component when compared with the substrate that was only plasma cleaned. The C-O peak is also much larger, relative to the C-C peak, on our plasma-treated substrate than it is on the commercial epoxide-terminated substrate. A possible explanation is that the compound used to epoxide-terminate the commercial surfaces (such as the commonly used 3-glycidoxypropyltrimethoxysilane, $C_9H_2O_5Si$) has more C-C bonds within its structure than does epichlorohydrin. Adsorbed organic contaminants may also be present on the commercial slides, introduced either before or after their chemical treatment. It is also possible that our process introduces a higher density of epoxide groups, which contain two C-O bonds each.

Another measure of the nature of a surface can be obtained using contact angles. Sessile contact angles of distilled water droplets measured on a series of treated substrates are shown in Table 5.1. The plasma-cleaned substrates as well as the substrates that were plasma-treated, but not exposed to epichlorohydrin, had contact angles too low

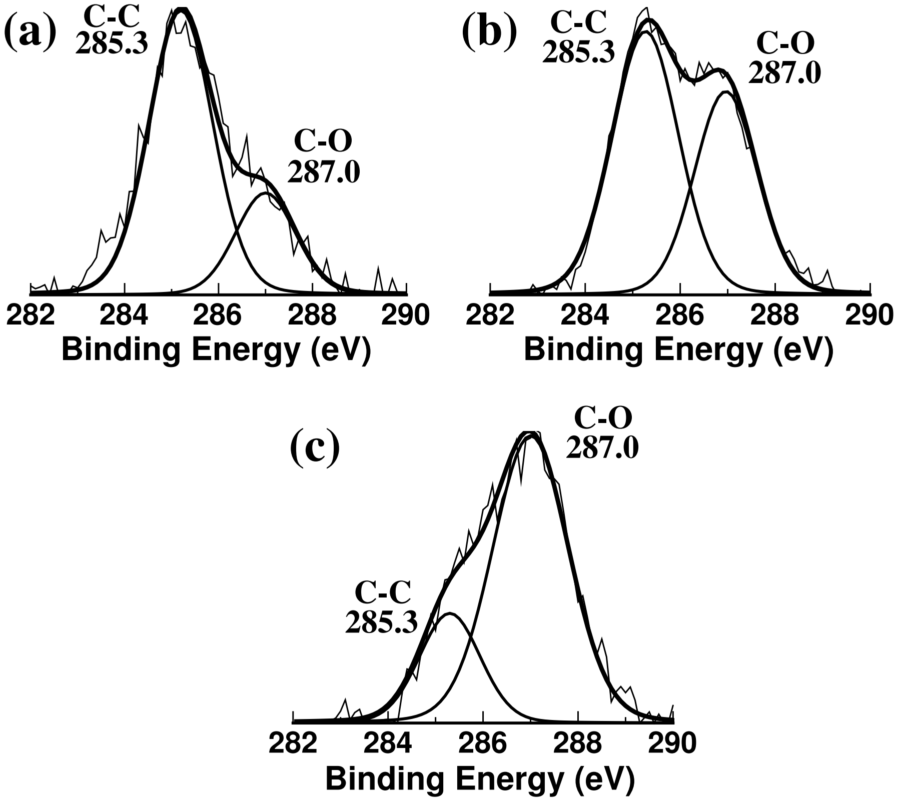

Figure 5.2: High-resolution XPS scans of the C 1s peak for treated and untreated glass substrates, from a) an untreated glass slide, b) a commercially available epoxide-terminated glass slide, and c) an epoxide-terminated glass slide prepared using our process.

Treatment	**% O**	**% Si**	**% C**	**% Cl**	**Sessile contact angle (degrees)**
Untreated	57	19	24	0	7.2 ± 3.5
Plasma-cleaned (exposed to atmosphere)	71	21	8	0	<2
Plasma-cleaned, water / oxygen plasma, no epichlorohydrin	75	16	9	0	<2
Plasma-cleaned, water / oxygen plasma, epichlorohydrin	70	17	10	3	66.5 ± 6.9
Commercial epoxide-terminated glass slide	58	26	16	0	46.1 ± 1.2

Table 5.1: XPS elemental analysis and sessile contact angles for a series of untreated, treated, and commercially available epoxide-terminated glass slides. XPS data obtained from high-resolution scans of the C 1s, Si 2p, O 1s, and Cl $2p_3$ peaks.

to detect, consistent with having a surface terminated by polar alcohol groups. After the surface is exposed to epichlorohydrin vapor, the contact angle becomes $\sim 67°$. This contact angle is consistent with the termination of the surface by epoxide chemical groups, as these groups are moderately hydrophobic and contact angles in this range have previously been observed on other epoxide-terminated surfaces.[42] The higher contact angle relative to the commercial slides is consistent with the larger C-O peaks in the XPS data from our surfaces. Further evidence for the epoxide termination of these substrates comes from the fact that exposure of the fully treated substrates to ethanolamine, an amine-containing compound that reacts with epoxide groups to form polar alcohol-containing functionalities, reduces the surface contact angles below the level that can be measured.

In addition to performing XPS and contact angle studies, we used an AFM to observe the removal of contaminants from the surface of glass slides by looking at nanoscale surface roughness. Figure 5.3 shows an untreated glass microscope slide, a commercial epoxide-terminated slide, a slide after the plasma-clean step, and a slide after the entire plasma treatment process (including the epoxide termination). The RMS roughnesses of these substrates are 1.9 nm, 2.4 nm, 0.9 nm, and 1.1 nm, respectively. The reduction in surface roughness shown in Figure 5.3a relative to that in Figure 5.3c is presumably due to the removal of adsorbed organics and dust particles. The roughness observed on the commercial slide does not appear to be caused by large particles present on the surface, but may be due to the morphology of the glass used.

5.3.2 Biomolecule binding

Epoxide chemical functionalities react with and covalently bind amine-containing biomolecules to a substrate. We tested the biomolecule binding properties of our freshly prepared substrates and compared them to commercially available epoxide-terminated glass substrates fresh out of a vacuum-packed box (which may, of course, have been stored for some period of time). We find that spots on the glass slides treated using the cold-plasma process had on average 10% higher fluorescence than spots on the commercial slides, however this increase was within the margin of error for the experiment (±20%). As controls, an untreated glass slide and a plasma-cleaned glass slide exposed to the same spotting and washing conditions as the treated substrates showed no fluorescence. A fluorescent image of spots deposited on a treated glass slide is shown in Figure 5.4a.

Because the fluorescently tagged oligos are covalently bound to the surface, the intensity of fluorescence within spots of bound oligos should be an indication of the density of biomolecule binding on the treated substrates. Our results suggest that our plasma-based treatment process provides a number of attachment sites on the substrate

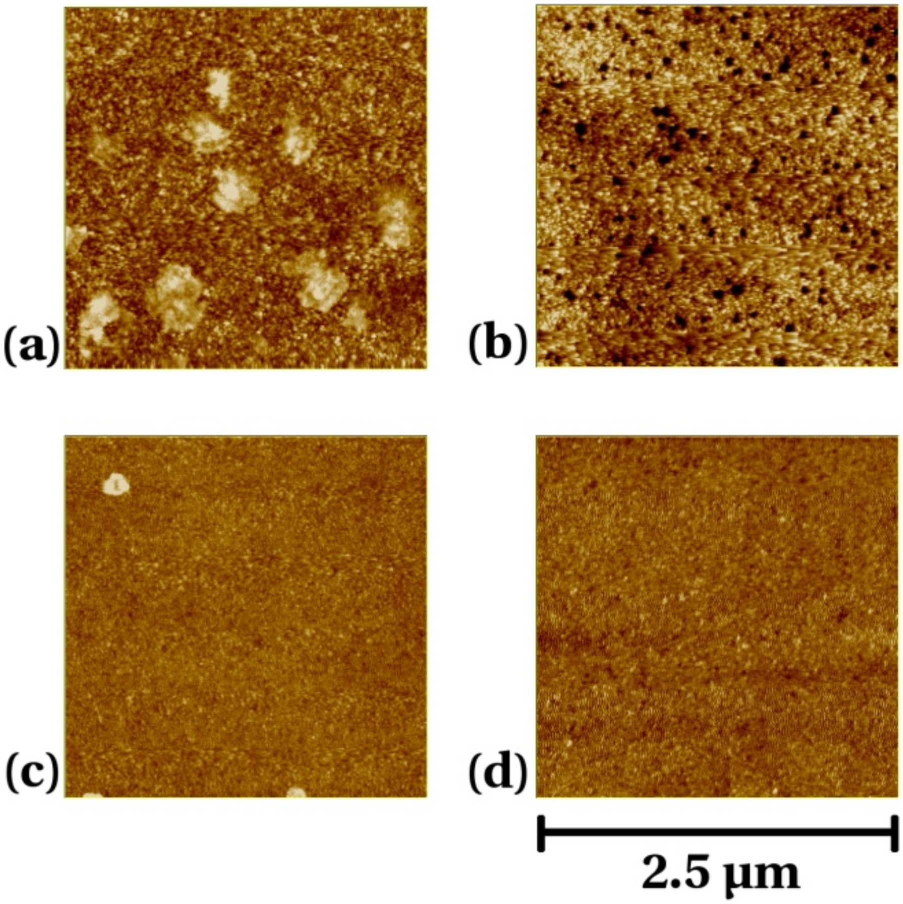

Figure 5.3: Atomic force microscopy images of a) an untreated glass slide, b) a commercial epoxide-terminated glass slide, c) a plasma cleaned glass slide, and d) a glass slide treated using our process. The height scale of each image is 10 nm. The surface roughness of the glass slide is reduced significantly after the plasma clean and storage in vacuum, presumably due to the removal of dust particles or other adsorbed contaminants. The cause of the observed roughness on the commercial epoxide-terminated slide is unknown.

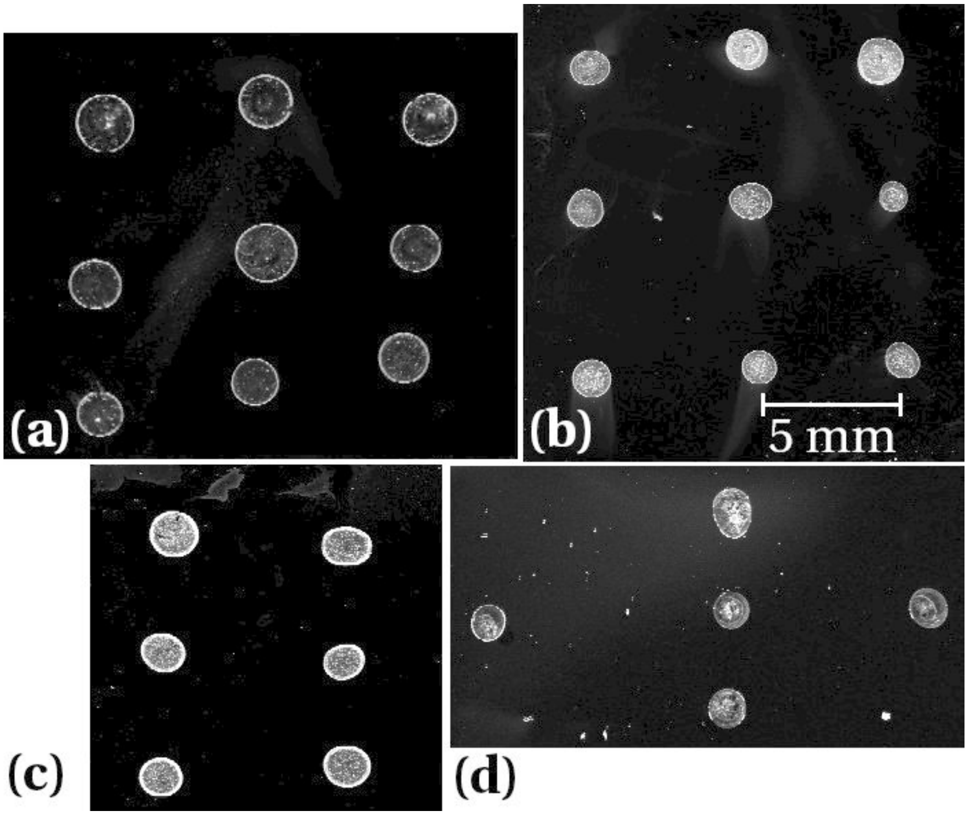

Figure 5.4: Fluorescence scans of amine-modified, fluorescein tagged oligonucleotides covalently bound on four different treated substrate materials. The four substrates are a) glass, b) fused silica, c) mirror-polished 304 stainless steel, and d) native oxide on silicon. All spots were $\sim$0.3 μL in volume, deposited by a hand pipette. The fluorescent intensities from spots on each substrate were at least as strong as on the commercial epoxide-terminated glass slides. All images are to the same scale.

surface at least comparable to that of the commercially available substrates, despite the fact that the plasma treatment has not been tuned for optimal binding site densities.

We tested our treatment process on several other oxide substrates, in addition to glass slides. Images of oligonucleotides chemically attached to fused silica, mirror-polished 304 stainless steel, and the native oxide on a silicon wafer are shown in Figure 5.4b-d. The best-performing of the various materials, functionalized mirror-polished stainless steel, has ∼10 times lower background fluorescence than glass slides plasma-functionalized at the same time and exhibits ∼2 times the fluorescent intensity in the spots of bound oligos (quantitated using the previously described procedure), possibly due to the reflective nature of the substrate. The same reduction in fluorescent background and increase in fluorescent signal is present when compared with commercial epoxide-terminated glass slides, suggesting that the functionalized mirror-polished stainless steel substrates may allow for more sensitive fluorescence-based assays than existing substrates.

5.3.3 Oligonucleotide hybridization

To verify the biological activity of molecules covalently bound to our plasma treated epoxide-terminated surfaces, we attached single-stranded oligos and allowed them to hybridize with their fluorescently tagged complements. Figure 5.5 shows an example of a glass slide on which spots of sequence A and sequence B were bound, then hybridized first with complement A followed by complement B. The fluorescence in only spots of sequence A with exposure to complement A and subsequent fluorescence of spots of B with exposure to complement B demonstrates that oligonucleotides bound to our plasma treated slides are capable of sequence-specific hybridization.

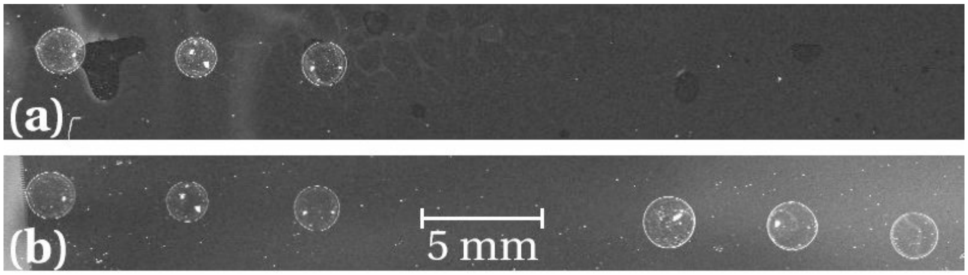

Figure 5.5: Fluorescence scans of fluorescently tagged oligos hybridized to their surface-bound complements. A glass slide was treated with our process, then spotted with $\sim 0.3\ \mu$L spots of sequence A and B in two 3-spot lines, sequence A on the left, sequence B on the right. These slides were then hybridized with a) complement A, followed by b) complement B.

5.3.4 Deposition of microscale spots

The high binding site density obtainable from our plasma functionalized substrates and increased fluorescent sensitivity from substrates such as stainless steel means that full advantage can be taken of recent technology that allows for the creation of much smaller, more uniform spots. One such device is called a fluid microplotter, which uses ultrasonics to deposit spots as small as 5 μm in diameter, consuming orders of magnitude less solution than current quill-pin-based spotters in the production of biological microarrays.[54] In addition, at these smaller spot sizes, irregularities, such as the coffee-ring effect noticed in Figs. 5.4 and 5.5, are greatly reduced, thereby increasing the reliability of fluorescent-intensity measurements from the spots. Figure 5.6a shows an image of a pattern of fluorescently tagged oligos deposited by the microplotter onto a glass slide treated with our process. The conditions for attachment are the same as those described previously. The spots in this image are 35 μm wide, with a 100 μm center-to-center spacing. Fluorescent intensity within the rectangular pattern of Figure 5.6a

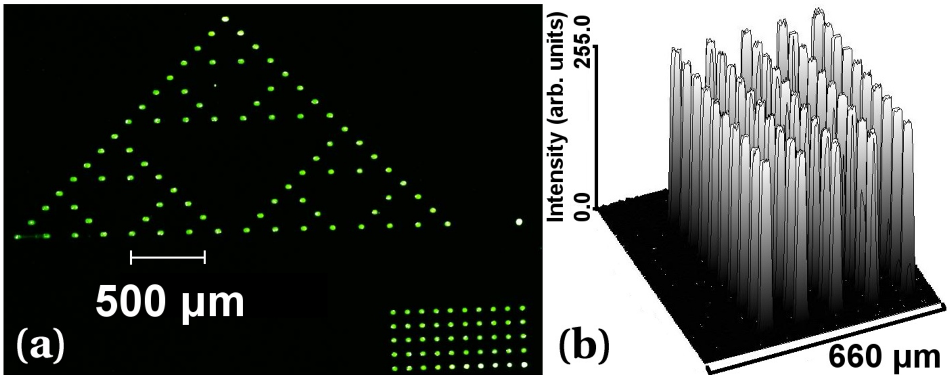

Figure 5.6: Fluorescence scans of amine-modified, fluorescein tagged oligos deposited by a fluid microplotter and bound to a glass slide functionalized with our process. The spots in these images are ∼35 μm in diameter, with a 100 μm center-to-center spacing. The two images are of a) patterns deposited by a microplotter and b) the fluorescent intensity of the spots in the rectangular spotted pattern, plotted in three dimensions. Even at this small a spot size, the bound oligonucleotides are easily detected over a minimal amount of substrate autofluorescence.

is plotted in Figure 5.6b, demonstrating a large signal-to-background ratio with these small spots.

5.4 Conclusion

In summary, we have used cold-plasma activation and an *in vacuo* vapor-phase reaction to terminate a range of oxide-containing surfaces with epoxide chemical groups. This procedure is much more efficient than prior methods. Minimal amounts of reagents are consumed when compared with standard wet-chemical processes and much less processing time is involved, implying a much lower cost of fabrication for these epoxide-terminated slides. We have so far made no effort to optimize the process, with respect to the plasma clean parameters, the epichlorohydrin exposure time, or the pump-out time. Despite that, the performance of our epoxide-terminated slides equals or exceeds that of commercial slides. In particular, the remaining Cl seen in the XPS data suggests that improvements are possible in the binding-site density we can produce on the surface.

5.5 Acknowledgements

Funding for this research was provided by Office of Naval Research and the University of Wisconsin-Madison's Office of University-Industry Relations. We thank Professors L. Smith and P. Nealey for permission to use facilities under their purview.

Chapter 6

Applications of a microplotter and cold plasma treatments

In the previous chapters, we described two technologies that have been developed to significantly improve the production of biological microarrays. We even demonstrated the application of these technologies in the fabrication of simple DNA microarrays. In this chapter, we will present a series of further applications of these technologies. These applications include functional protein microarrays used to detect specific targets in solution, a cold-plasma-based treatment process for the treatment of carbon-containing materials rather than oxides, the targeted growth of carbon nanotubes using small spots of deposited catalyst solution, the enhancement of matrix-assisted laser desorption / ionization mass spectrometry (MALDI) sensitivity using microplotter-deposited small spots, and polymer-based electronics deposited using a microplotter.

6.1 Protein microarrays

In Chapters 2 and 5, we showed some simple one- and two-component DNA microarrays deposited by a microplotter and / or bound to epoxide-terminated surfaces produced using our plasma-based treatment process. However, the true promise of these technologies lies in their ability to enhance the production of protein microarrays. We have verified the merit of both in the creation of a series of proof-of-principle protein microarrays, culminating in the fabrication of a working microarray capable of detecting the presence of a common chicken pathogen.

Before attempting to manufacture protein microarrays using the microplotter, it is necessary to determine if any denaturation of proteins (resulting in loss of activity) takes place when they are pumped out of the microplotter dispenser onto a surface. When the process of acoustic pumping was discussed in Chapter 4, we showed that a large amount of shear may develop in the fluid close to the vibrating wall of the glass needle. Additionally, the small aperture size of the microplotter needle can lead to shear by itself if fluid passes through it at a high enough rate. Shear has been known to denature certain sensitive proteins[78, 79], so deposition of these proteins using the microplotter may cause them to denature. As stated in Chapter 2, early results with long strands of DNA do not seem to indicate any adverse effects of deposition, but DNA is a fairly robust molecule. To investigate the effect of dispensing on protein activity, two enzymes, glucose oxidase and alkaline phosphatase, were placed in 30 μL solution wells. A microplotter dispenser with an aperture inside diameter of 10 μm was then used to pull up 1-2 μL of solution from a well and pump that solution back into a well. This process was repeated 30 times for each of the two wells. A colorimetric enzyme activity assay was then performed on these solutions, with enzyme solutions straight out of the bottle and enzyme solutions exposed to heat as controls. It was found that the microplotter-exposed samples had 99% of the activity of the unmodified enzymes, a difference well within the margin of error for this experiment, and that the heat-exposed controls had lost nearly all of their activity.[99]

As an initial test of protein deposition and binding, antibodies were deposited by a microplotter onto an epoxide-functionalized glass slide produced using our plasma treatment process. FITC-tagged anti-goat IgGs were deposited onto the slide, which was then blocked, washed and scanned to obtain the image in Figure 6.1, following a procedure similar to the one detailed in Section 5.2.5 of Chapter 5. The protein probe features in this microarray are approximately 40 μm wide. The contrast between fluorescent signal of the antibody spots and the slide background is not as high as in Figure 5.6 simply because a much lower concentration solution was used in spotting the antibodies. This provides a demonstration of how these two technologies can be used together to fabricate protein microarrays.

Next, we fabricated some simple capture-based protein microarrays using the microplotter. Anti-fluorescein goat IgGs (antibodies that selectively bind the fluorescent dye molecule fluorescein) were deposited and bound onto a poly-*l*-lysine-coated glass slide. Poly-lysine-coated glass was used as a substrate rather than epoxide-terminated glass simply because poly-lysine-coated glass is currently the most widely used substrate for protein microarrays and we were only concerned with demonstrating the use of the microplotter in depositing active protein microarrays. Four grids of spots made from different concentrations of antibody solution (from 1 to 200 μg/mL) were deposited onto a slide and allowed to react in humid conditions for 8 hours. This glass slide was then blocked with a 1% solution of bovine serum albumin (BSA) for 1 hour and then exposed to a uniform 1 μM concentration of fluorescein for 30 minutes. After the fluorescein exposure, the slide was rinsed and placed in a DI water bath for 2 hours. Figure 6.2 shows the result of a fluorescence scan of the surface of this slide. The concentration of deposited antibody increases from left to right, so it is clear that binding of fluorescein was occurring at the higher antibody concentration spots. The smearing in this image is due to weak electrostatic binding holding the antibodies on the poly-*l*-lysine-coated

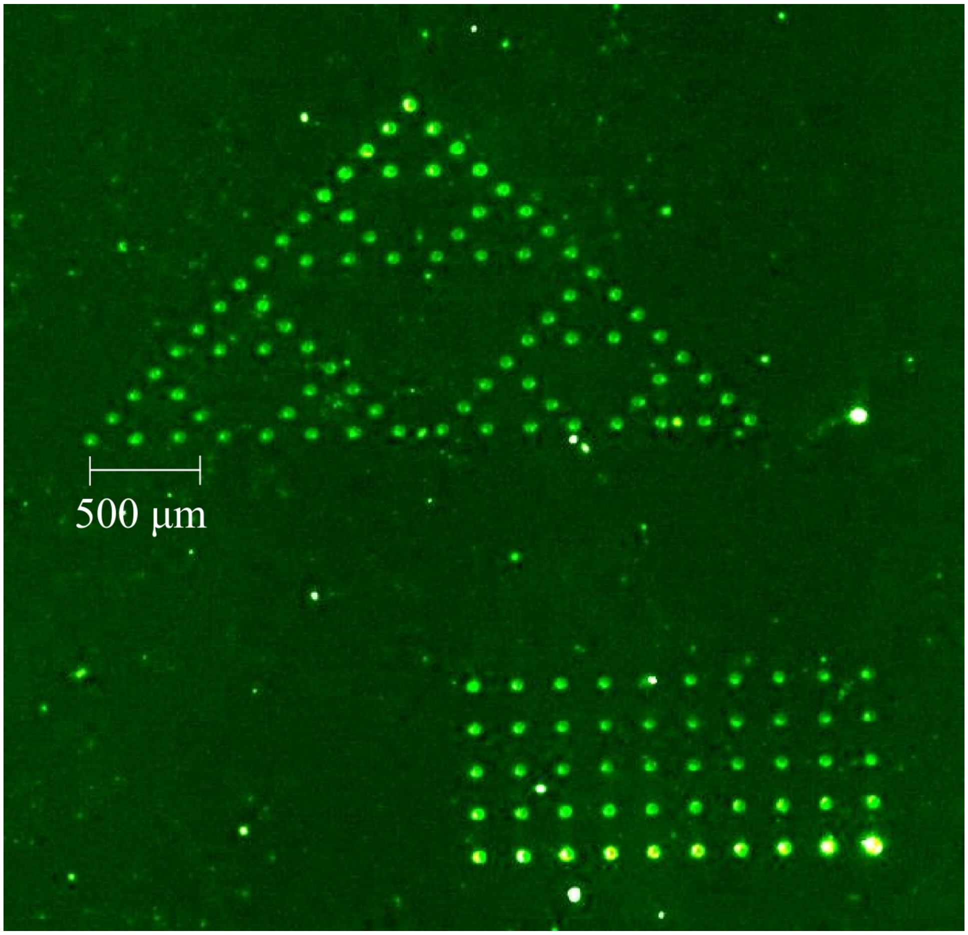

Figure 6.1: A fluorescence scan of FITC-tagged anti-goat IgGs bound to a glass slide functionalized with our plasma treatment process. The spots are 40 μm in diameter.

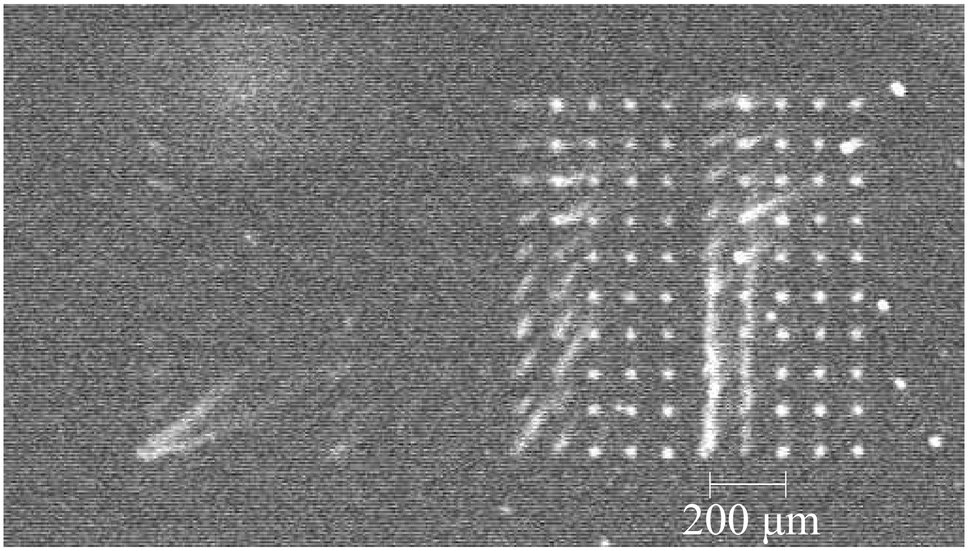

Figure 6.2: A microplotter-deposited assay for the detection of fluorescein. Grids of anti-fluorescein IgG spots were deposited on a poly-*l*-lysine-coated surface glass surface at varying concentrations and exposed to a uniform-concentration solution of fluorescein. The spots in the grids are 30 μm in diameter. The spots made from lower concentrations of IgG are not visible.

surface; the high background fluorescence is due to nonspecific binding of the fluorescein to the surface despite blocking. These artifacts illustrate why a covalent attachment scheme, such as the epoxide attachment used for our plasma treated slides, has significant advantages over the electrostatic attachment used with a poly-*l*-lysine-coated glass slide.

As a final test of the protein microarray fabrication capabilities of the microplotter, an enzyme-linked immunosorbent assay (ELISA)-type, or sandwich, microarray was fabricated to detect the presence of the Newcastle Disease Virus (NDV), a common avian pathogen. This type of assay is shown in the far right of Figure 1.6 in Chapter 1. NDV antigen solution was deposited using the microplotter in several grids on a poly-*l*-lysine-coated glass slide. As above, the slide was blocked with a 1% BSA solution for 1

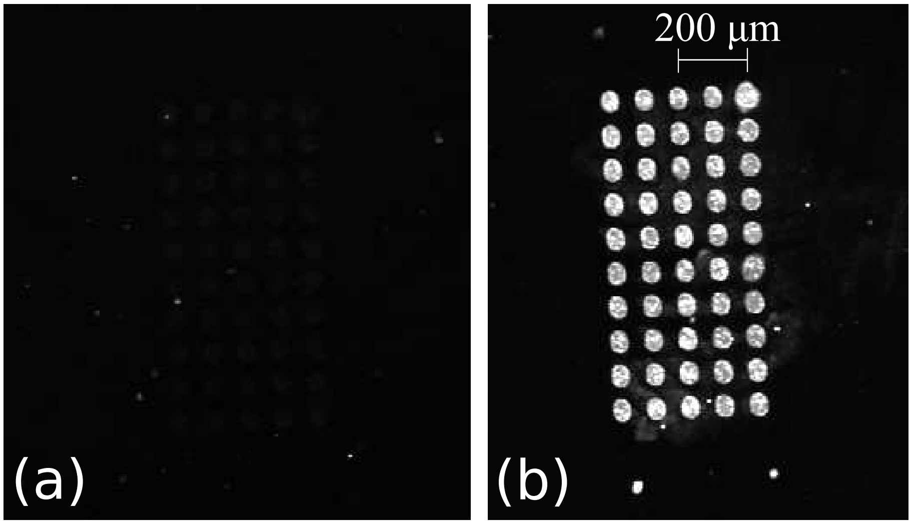

Figure 6.3: Fluorescent images of an ELISA-type assay in a microarray format fabricated using the fluid microplotter. Antigen for the Newcastle Disease virus was spotted in two grids on a surface, and then each of these grids was either exposed to a) NDV-negative sera or b) NDV-positive sera. Finally, a fluorescently tagged secondary antibody was introduced to both grids.

hour and then exposed to negative sera in one part and positive sera in another. Positive sera is simply solution containing antibodies for NDV, indicating an immune response to the presence of NDV in the avian host. These antibodies should bind to the probe antigen deposited on the glass slide surface. The slide was then rinsed, placed in a DI water bath for 1 hour, and then exposed to a fluorescently tagged anti-chicken IgY secondary antibody. The slide was rinsed again, placed in a DI water bath for 1 hour, dried, and scanned. The result is shown in Figure 6.3. Even at relatively low concentrations of positive sera, the grids of 70 μm spots fluoresce strongly (Fig. 6.3b), while only a very small amount of fluorescence (from nonspecific binding of the secondary antibody to the antigen spots) is seen in the negative controls (Fig. 6.3a).

6.2 Cold plasma functionalization of carbon-containing materials

In *Protein Microarrays*, it is stated that one of the chief problems with existing protein microarrays is background introduced by the nonspecific binding of target molecules on the array.[20] A microarray works because the specific interaction of probe and target leads to measurable changes at different locations on the microarray. If target is non-specifically attaching all over the microarray, then it becomes very hard to detect the specific binding of target at probe spots. It is therefore necessary to manufacture a microarray in such a way that proteins or other biomolecule targets will not stick to the surface and will only attach themselves to their probe complement. Glass is the most widely used substrate material for microarrays due in part to the existence of simple chemical processes for functionalizing its surface to bind probe biomolecules, but proteins will readily adsorb onto its surface. Teflon is an ideal nonstick surface, but it is so chemically resistant that few surface treatments are known and almost all of those are impractical for commercial-scale manufacturing. Other polymers, such as polyethylene, also have reduced levels of nonspecific adsorption or other desirable characteristics for microarray fabrication.

We have developed plasma-based processes, similar to those described in Chapter 5, that allow us to treat carbon-containing materials (such as polymers, graphite, diamond-like hard carbon, etc.) to introduce either aldehyde or epoxide functional groups. These processes share the advantages of the previously described oxide treatment, such as minimal reagent consumption and a short treatment time, and can work over a broader swath of materials due to the many different types of polymers with varying properties.

The basis for this work is a process developed by Alvarez-Blanco, *et al.*[100] They used an RF argon plasma to generate free radicals on the surface of a polymer, then turned off the plasma and flowed a vapor of ethylenediamine into their plasma chamber. Ethylenediamine has two amine groups in its chemical structure, and one of these amine groups can react with the free radicals present on the polymer surface, leaving the other amine free. This amine-terminates the polymer surface. They then flowed a vapor of oxalyl chloride over the surface. Oxalyl chloride has two chloracid groups, chemical groups that can react with amines to form a covalent bond. As with the ethylenediamine, one end of the oxalyl chloride reacts with the amine-terminated polymer surface, while the other is free. They were able to demonstrate the covalent binding of horseradish peroxidase to polyethylene surfaces treated in this fashion. Unfortunately, chloracid functionalities rapidly hydrolyze in the presence of moisture and will only retain their activity for 30 minutes in normal atmospheric conditions. This makes these functional groups impractical for use in fabricating biological microarrays.

6.2.1 Aldehyde-termination of polymer surfaces

To avoid the hydrolysis of the amine-reactive functional groups, we modified this process to aldehyde-terminate, rather than chloracid-terminate, a polymer surface. The reactor we employed was the same as the one described in Chapter 5, but with a 13.56 MHz power supply. The initial step in the process consisted of a two minute argon plasma at 200 mTorr and 200 W. The chamber was then evacuated and filled with ethylenediamine vapor at 1 Torr for 30 minutes. The chamber was pumped out again

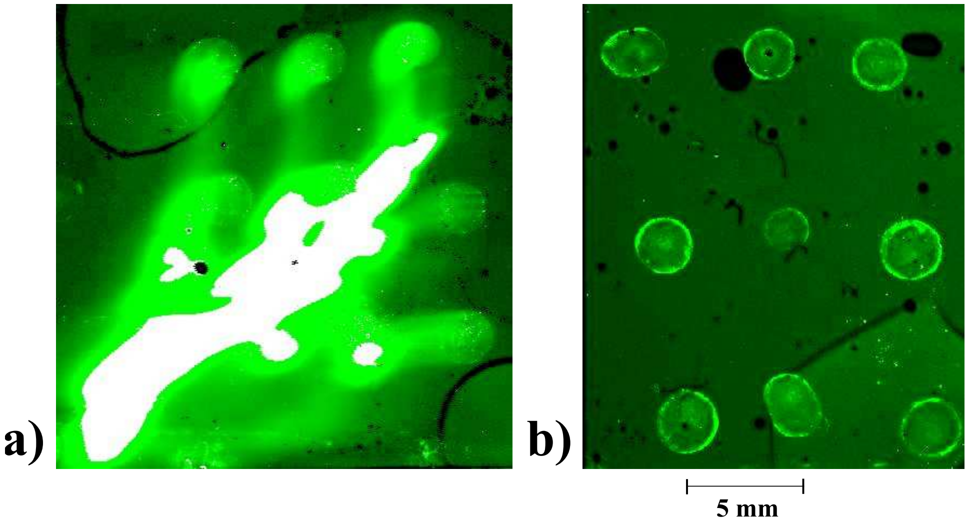

Figure 6.4: Fluorescence scans of fluorescein-tagged, amine-modified oligonucleotides bound onto aldehyde-terminated polycarbonate. The weak aldehyde-amine covalent bonds in a) were not reduced with $NaBH_4$, while those in b) were. The application of a weak base buffer to both caused oligonucleotides to be released from the surface of a), but the oligonucleotides in b) were firmly attached.

and, instead of the oxalyl chloride used by Alvarez-Blanco, *et al*, filled with a 1 Torr vapor of glutaraldehyde (a compound with two aldehyde groups) for 30 minutes. Amine-modified, fluorescently tagged oligos were deposited and bound to treated surfaces in the fashion stated in Section 5.2.5 of Chapter 5. Figure 6.4 shows two images of this: Figure 6.4a is a surface where no post-processing step was used after the deposition of oligonucleotides and Figure 6.4b is a surface that was exposed to a 15 mg/mL solution of $NaBH_4$ (a strong reducing agent) for 30 minutes. Both were washed thoroughly before being exposed to a weakly basic buffer and scanned. The basic buffer can be seen to hydrolyze the weak bonds with the surface and smear out the oligonucleotide spots in Figure 6.4a, while the stronger covalent bonds in Figure 6.4b survive.

The advantages and disadvantages of aldehyde groups are examined in detail in

Chapter 1, Section 1.3. Briefly, while aldehydes will covalently bind amine-containing biomolecules and do not suffer from rapid hydrolysis by atmospheric moisture, the reaction that binds amine-containing biomolecules is slow and requires post-processing in order to create a strong bond (as seen in the difference between Figure 6.4a and b). Aldehyde functional groups are most useful for immobilizing amine-modified oligonucleotides when epoxide groups are unsuitable due to an unacceptable level of binding between the epoxides and secondary or tertiary amines present on certain nucleotides. Epoxides react much faster and form stable bonds that need no postprocessing, so they are more widely used than aldehydes for all other covalent attachments of amine-containing biomolecules.

6.2.2 Epoxide-termination of polymer surfaces

A different process was used to epoxide-terminate the surface of various polymers. Again starting with a two minute argon plasma at 200 mTorr and 200 W, the chamber was then evacuated and filled with a 1 Torr vapor of epichlorohydrin for 30 minutes. After the vapor exposure, the chamber was pumped down and held at vacuum for an additional 30 minutes. This process is very similar to the one described in Chapter 5, with the differences being a lack of a plasma clean step and the use of an argon plasma, rather than a water / oxygen mixture plasma, to activate the surface prior to vapor exposure. Figure 6.5 shows a series of polymer materials that were exposed to this process, then to the previously described oligonucleotide attachment, blocking, and washing steps. All materials show covalent attachment of DNA, including the extremely chemically resistant Teflon.

Unfortunately, the density of bound oligonucleotides (used as a proxy for the density

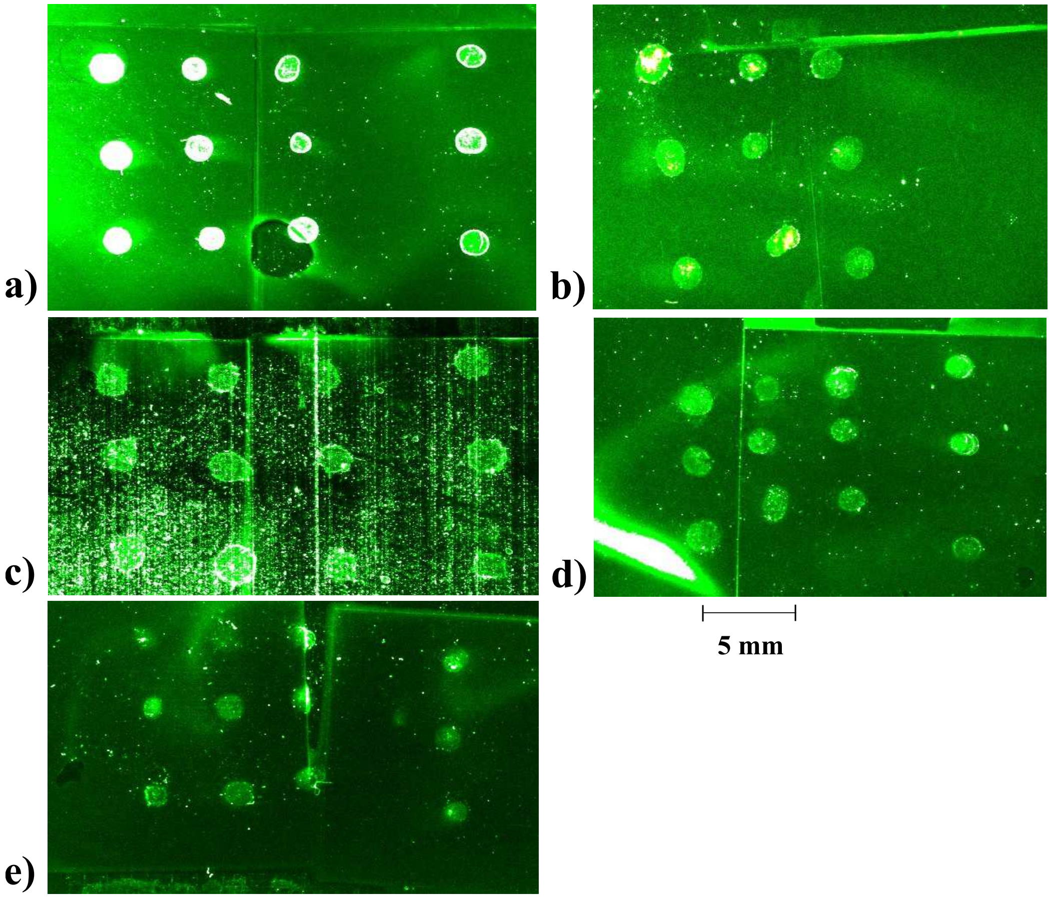

Figure 6.5: Fluorescence scans of fluorescein-tagged, amine-modified oligonucleotides bound onto various epoxide-terminated polymers. The polymers shown are a) Delrin (an acetal polymer), b) polyvinylchloride (PVC), c) polytetrafluoroethylene (PTFE or Teflon), d) high-density polyethylene (HDPE), and e) ultra-high-molecular-weight polyethylene (UHMWPE).

of epoxide functional groups on the surface) on these polymer surfaces is far less than that observed on the treated glass slides of Chapter 5. On average, the polymers have 5% of the surface density of epoxide functional groups that the plasma-treated glass slides do. The reason for this may have to do with the design of the process itself, because we still are not certain of the exact mechanism by which functionalization of the polymer surfaces with epoxides takes place. One possibility is that the initial argon plasma cleaves the carbon-carbon bond (as described above) and that the epoxide ring of epichlorohydrin is broken open by the free radicals on the surface, eventually leading to an epoxide-terminated surface. Another possibility is that the argon plasma, combined with a slow but noticeable air leak into the plasma chamber, oxidizes the top surface and introduces some hydroxyl groups. These hydroxyl groups are known, from the work described in Chapter 5, to react with epichlorohydrin and lead to an epoxide-terminated surface. If this is indeed the case, then the low densities of functional groups observed to date may be due to the use of argon, rather than a water / oxygen mixture, as the plasma. Argon, coupled with an air leak, would only produce a small number of hydroxy groups. Some slight evidence for this was found when a polyethylene slide was exposed to a water / oxygen plasma, then epichlorohydrin, and a significant improvement in biomolecule attachment over the argon-based process was noted.

Additionally, a significant amount of background fluorescence can be seen in Figures 6.4 and 6.5. We performed a survey of the background fluorescences of various polymer materials by first ordering each material as a 1 mm thick sheet from McMaster-Carr. These sheets were then cut into 75 mm x 25 mm (microscope-slide-sized) pieces and their surfaces cleaned thoroughly using a detergent solution. They were then scanned in the same microarray scanner used for the previously shown fluorescence images at two photomultiplier tube (PMT) gains and in three different dye channels: FITC (absorption: 488 nm, emission: 520 nm), Cy3 (absorption: 550 nm, emission: 570 nm), and Cy5 (absorption: 649 nm, emission: 670 nm). The high gain was used to differentiate between the low-autofluorescence materials and the low gain was used to examine the materials that maxed out the PMT at the higher gain. The average fluorescence measured within a large rectangular region of the surface free of obvious defects, such as scratches, was used to quantify the fluorescent background. Table 6.1 summarizes these results. Some trends can be seen. All materials that contain conjugated structures within their backbone (polycarbonate, garolite, PETG, PMMA, and ABS) have very high autofluorescences. Clear PVC has higher autofluorescence than grey PVC, translucent LDPE has a higher autofluorescence than opaque white HDPE or UHMWPE, and black Delrin has lower autofluorescence than white Delrin. The higher autofluorescence with translucent / transparent polymers may be due to the deeper penetration of the exciting laser from the microarray scanner into the bulk material, exciting fluorescent plasticizers or other impurities. The carbon black added to make the dark Delrin may have lower autofluorescence than bulk Delrin. We also noted that rough surfaces, such as can be seen in Figure 6.5c, have higher background fluorescence because of light scattering from surface defects.

It appears that the issue of high fluorescent background can be dealt with simply by choosing the correct starting material. Teflon, HDPE, and UHMWPE all show very low autofluorescences, in some cases lower than that of glass (especially in the Cy5 dye channel). When examined closely, we found that almost all of the registered autofluorescence for these materials was due to light scattering from a rough surface. Extruded

	40 % gain			70 % gain		
Material	**FITC**	**Cy3**	**Cy5**	**FITC**	**Cy3**	**Cy5**
Glass	0.3	1.1	1.1	17.2	9.1	75.6
Mechanical grade polytetrafluoroethylene (PTFE)	0.1	0.3	0.0	19.2	45.9	38.3
High-density polyethylene (HDPE)	0.7	5.2	0.3	26.5	64.4	20.4
Electrical grade Teflon	0.3	9.2	1.0	31.8	235.7	64.3
Ultra-high-molecular-weight polyethylene (UHMWPE)	0.4	2.3	0.9	32.2	89.4	23.0
Polyvinylchloride (PVC), gray	0.9	2.9	5.4	69.0	183.2	45.4
Delrin, black	1.0	8.1	0.1	82.2	255.0	28.1
Polypropylene	1.1	3.6	0.5	100.7	247.5	54.1
Low-density polyethylene (LDPE)	1.3	4.1	0.4	149.5	249.9	46.0
Acrylonitrile-butadiene-styrene (ABS)	2.9	5.5	0.0	220.4	255.0	36.6
Delrin, white	3.1	6.2	0.1	238.2	255.0	40.4
Extruded polymethylmethacrylate (PMMA)	7.6	68.8	109.2	255.0	255.0	255.0
Polyvinylchloride (PVC), clear	35.2	254.8	113.1	255.0	255.0	255.0
PETG	53.6	106.2	106.8	255.0	255.0	255.0
Polycarbonate	60.3	176.6	25.3	255.0	255.0	255.0
Garolite	246.9	255.0	52.6	255.0	255.0	255.0

Table 6.1: Measured autofluorescences of various polymers. The fluorescence values are in arbitrary units.

polymers, such as the PMMA, polycarbonate, and PVC that we tested, are smooth enough for our purposes, and it is possible to find polyethylene and processed in this manner. However, Teflon is a harder polymer to deal with, due in part to its much higher melting temperature. It may be a challenge to find flat enough Teflon sheets. Instead, it may be required to deposit a thin layer of Teflon prepolymer on an already-flat surface and allow polymerization to take place.

With the appropriate choice of starting materials and proper tuning of the plasma treatment process, it should be possible to create functionalized polymeric microarray substrates with significant advantages over glass-based substrates.

6.2.3 Epoxide-termination of carbon nanotubes

In addition to polymeric materials, we have treated more exotic carbon-containing materials, such as carbon nanotubes. Carbon nanotubes consist of individual sheets of graphitic carbon that have been wrapped into tubes with diameters on the order of 1-10 nanometers.[101] They can either exist as freestanding tubes with a single wall, referred to as single-walled nanotubes (SWNTs), or as tubes with multiple coaxial walls, referred to as multi-walled nanotubes (MWNTs). Nanotubes are fascinating structures for a number of reasons, such as the fact that nanotubes with a different chirality (or ”twist”) can be either semiconducting or metallic in character, or their extraordinarily high surface area to volume ratio.

One of the most interesting properties of SWNTs is the fact that they are almost entirely composed of surface area. Thus, interactions at the surface can have a significant influence on the bulk properties of the material. It has been shown that the adsorption of a particular gas species to the outside of a semiconducting nanotube causes a noticeable change in electrical conductivity.[102] A similar structure called a silicon nanowire has been used as a more specific biological and chemical sensor by chemically attaching receptors to the nanowire surface.[103]

These semiconducting nanostructures work as sensors in a manner similar the functioning of a field-effect transistor (FET). In an FET, charge applied to a gate electrode controls how much current flows from a source electrode to a drain electrode. In these nanostructures, metal contacts to either end of a nanotube or nanowire act as source and drain electrodes. When charged chemical groups attach to the surface of a nanotube or nanowire, they act as a gate and either help or hinder current flowing through the nanostructure. As has been demonstrated in the case of nanowires, when receptors for charged proteins or chemical groups are bound to the outside of a semiconducting wire that has a very large surface area to volume ratio, that semiconducting wire becomes an electronic sensor that responds selectively to one target.[103]

A means of covalently binding probe biomolecules to carbon nanotubes is necessary in order to use them for robust biosensors. Only a few chemical processes exist for adding reactive functionalities to the ends or sidewalls of carbon nanotubes, and those require aggressive wet chemical treatments for extended periods of time.[104] As an alternative to these treatments, we have applied the plasma treatment process described in the previous section to epoxide-terminate carbon nanotubes. In this case, the argon plasma should act to break the C-C bonds present in the sidewalls and at the ends of nanotubes, either leaving behind radicals or forming hydroxy groups due to a slight atmospheric leak in the plasma chamber. Epichlorohydrin is then bound to these sites on the nanotube, forming amine-reactive epoxide groups.

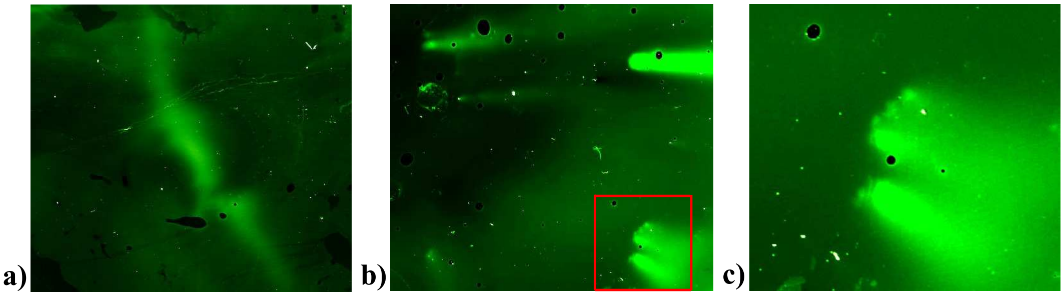

Figure 6.6: Fluorescence scans of fluorescein-tagged, amine-modified oligonucleotides bound onto epoxide-terminated carbon nanotubes. Large aggregates of carbon nanotubes were placed in a grid on two glass slides, with a) one unexposed slide as a control and b) the other slide exposed to the plasma treatment process. Oligonucleotide solutions were then washed across the faces of both slides. Three fluorescing aggregates of carbon nanotubes can be seen in the corners in b), with no such fluorescence (the large streak is an interference fringe) in a). A magnified image of one of the aggregates in b) is shown in c).

We tested this process by hand-pipetting large aggregates of nanotubes (so that fluorescence from bound oligos would be detectable in a microarray scanner) in a grid on two glass slides. One glass slide was exposed to the plasma treatment process, while the other was left untreated as a control. Both slides were exposed to amine-modified, fluorescein-tagged oligonucleotides uniformly across their entire surface for 1 hour, then blocked, washed, and scanned as described above. The results are shown in Figure 6.6, where it can be seen that three nanotube aggregates fluoresce on the treated sample (Fig. 6.6b), while no such spots are found on the control (Fig. 6.6a).

We then attempted to use this treatment in the fabrication of a simple nanotube-based biosensor. Carbon nanotubes were deposited between two gold electrodes that had been placed on a silicon wafer with a 200-nm-thick oxide layer. A relatively low resistance was measured between the gold electrodes, consistent with the presence of a large number of nanotubes bridging the cap between them. However, after exposure of this structure to the plasma treatment process, no current was found to flow between the electrodes. It appears that the argon ion bombardment of the nanotubes during the plasma treatment broke too many bonds in the nanotube sidewalls, leading to the disintegration of the tubes at some point along their length. A reduction in the amount of damage done to the nanotubes during this process should be achievable by altering the argon pressure or applied power during the plasma treatment step to reduce the degree of argon ion bombardment on the nanotubes.

6.3 Deposition of catalyst for the growth of carbon nanotubes

Aside from the use of cold plasmas to chemically functionalize carbon nanotubes, researchers in the Eriksson and Lagally research groups at the University of Wisconsin - Madison have used a microplotter to direct the growth of nanotubes at specific locations on a surface.[105, 106]

Nanotubes can be grown by placing an appropriate catalyst material within a flowing stream of a carbon feedstock gas inside of a heated chemical vapor deposition (CVD) chamber. [107] The catalyst, usually in the form of metal nanoparticles, induces carbon from the gas to be deposited at just the right rate to form a nanotube that grows out from the catalyst.

As mentioned in the previous section, the placement of carbon nanotubes at specific locations on a surface, either by growing them in the desired area or by depositing presynthesized nanotubes, is vital to the production of nanotube-based circuits or sensors. A microplotter can facilitate the growth of nanotubes in select areas on a surface by depositing microscale spots of catalyst material in precise locations. One of those

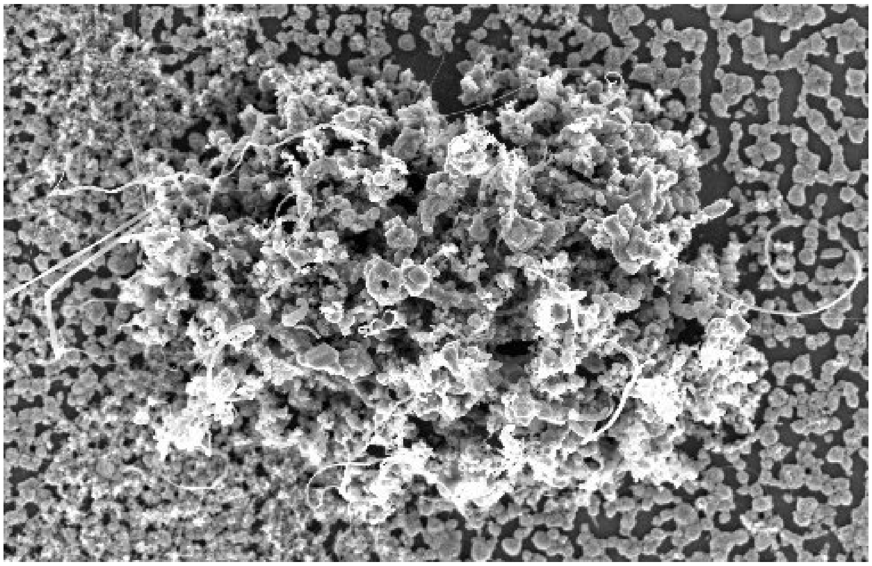

Figure 6.7: Microplotter-deposited spot of catalyst for the growth of nanotubes. This SEM micrograph of a 7 μm wide spot of catalyst was taken after exposure to the growth process for carbon nanotubes, so some small nanotubes can be seen coming off of the catalyst material. Image courtesy of Todd Narkis.

spots is shown in the scanning electron microscope (SEM) micrograph in Figure 6.7.

Simple nanotube-based devices composed of individual nanotubes or bundles of nanotubes bridging a gap between two electrical contacts were fabricated with the aid of a microplotter. The fabrication process began with a clean oxidized silicon wafer. This wafer may or may not have had some physical architecture pattered onto it (it was sometimes necessary to suspend the nanotubes in order to study them). The wafer was placed in a microplotter and aligned. A catalyst solution, composed of a methanol-based suspension of iron and molybdenum nanoparticles contained within porous alumina particles, was loaded into a dispenser. The microplotter then deposited spots of the catalyst at programmed locations that matched the patterning of the silicon wafer. An example of spots deposited on a patterned wafer is shown in Figure 6.8. Once the catalyst spots

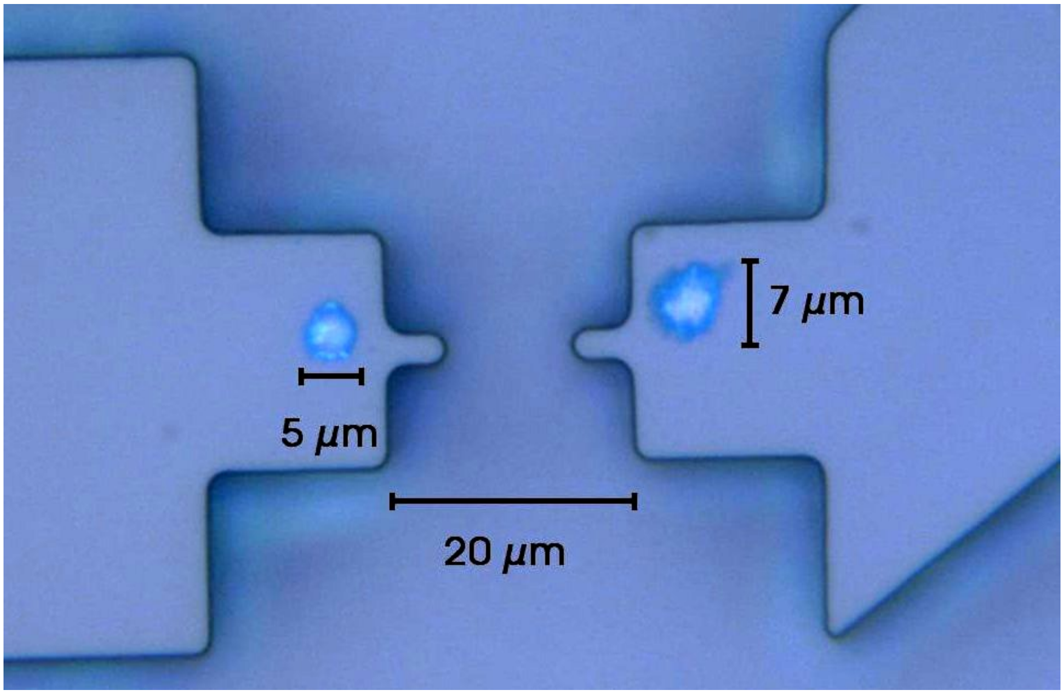

Figure 6.8: Locations of deposited catalyst for nanotube growth. In this test case, blue food coloring was deposited instead of catalyst solution at the locations where catalyst would be placed.

were all in place, the wafer was placed in a CVD chamber and exposed to the appropriate conditions for nanotube growth. The wafer was then removed from the chamber and metal electrical contacts were patterned on the wafer and wires were bonded to these contacts. A example of one of these devices, a single nanotube (or bundle of nanotubes), suspended in air between two electrical contacts, is shown in Figure 6.9.

Most researchers use photolithography to pattern spots of catalyst for the fabrication of such devices. Photolithography can be costly, time-consuming, and potentially damaging to the sample. Additionally, it was found that samples produced using the

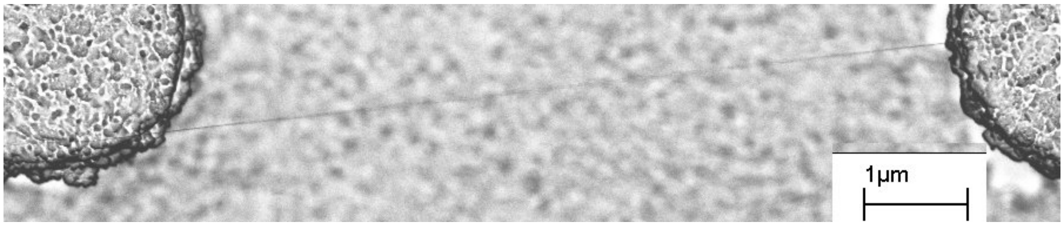

Figure 6.9: A grown bundle of carbon nanotubes bridging the gap between two electrical contacts. Image courtesy of Todd Narkis.

microplotter to deposit catalyst had a higher yield of functional devices than photolithographically patterned ones. This was possibly related to the observation that by using a more viscous nanotube catalyst solution with the microplotter, raised catalyst spots can be deposited. These raised catalyst spots appear to yield longer nanotubes, making it more likely that useful devices will be fabricated. The growth of longer nanotubes appears to be related to a disruption in the flow of carbon feedstock with the taller spots.[108]

The ability to produce significantly larger quantities of working devices in a shorter time and at a much lower cost enables research and applications which would otherwise be impractical, if not impossible. Devices using nanotubes grown at select locations could act as biosensors (as described above) [102, 109], photosensors[105, 106, 110], or even optical transmitters[111].

6.4 Enhancement of mass spectrometry sensitivity

Apart from the microarray-related applications, we have also used the microplotter to enhance the sensitivity of matrix-assisted laser desorption / ionization (MALDI) mass spectrometry.[112] MALDI mass spectrometry is a technique used to analyze the molecular weights of unknown chemicals or biomolecules, even when the molecules of interest weigh more than 10 kDa (10,000 molecular weight).[113, 114] To do this, analyte and matrix are deposited onto a metal plate, called a target. Analyte can either be mixed into matrix before deposition of both or analyte can be deposited and then coated by matrix. The target, with analyte and matrix on its surface, is then loaded into a vacuum chamber. A pulsed laser beam strikes the matrix at specific locations, transferring energy into the matrix and ejecting it from the surface. The matrix takes the analyte molecules with it and both become ionized, separating from each other before they reach the mass spectrometer. The spectrometer then reads the mass to charge ratio (m/z) for each incident molecule. This process is illustrated in Figure 6.10. The ability to precisely determine the molecular weight of analyte molecules allows for the identification of small molecules, proteins, or even DNA sequences and is widely used in the pharmaceutical industry.

The way in which analyte and matrix are put down on the MALDI target can affect the quality of the mass spectra gathered. Currently, the most popular means for placing both onto a surface is hand-spotting, which leaves behind inhomogeneous spots of irregular volume (see Figures 5.5 and 5.4 in Chapter 5 for examples of this). The variations in amount of analyte and matrix present at different locations on the surface makes quantitative analysis difficult, if not impossible. Additionally, the large spots waste material, because only the small area where the laser beam strikes (10-100 μm, typically) is used for gathering the mass spectrum.

Several of the fluid dispensing technologies described in Chapter 1 have been applied to the deposition of analyte and matrix on MALDI targets. Quill-pins, while widely used

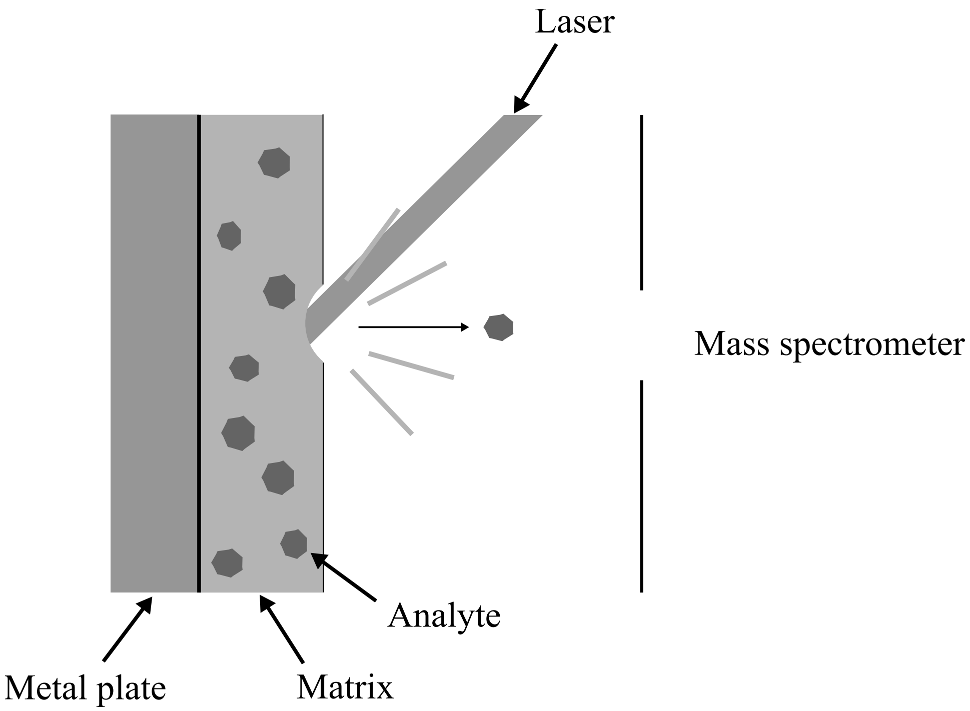

Figure 6.10: A schematic of MALDI mass spectrometry. An analyte is trapped within a matrix material. When the matrix is struck by a laser beam, enough energy is transferred into the material to eject it from the surface, taking the analyte with it. The matrix and analyte are ionized and separate as they travel to the mass spectrometer.

in microarray fabrication, are poorly suited for the deposition of matrix because the saturated matrix solutions tend to crystallize and clog the pins while dispensing. They are also limited to feature sizes larger than 75 μm. Inkjets have been used to deposit both analyte and matrix, but the smallest reported spot size is on the order of 180-200 μm.[115] They also have a problem with clogging when dealing with matrix solutions, so only certain matrix materials can be used with them. Finally, ringed-piezoelectric dispensers have been used to produce somewhat smaller spots, on the order of 100 μm, but they still suffer from many of the problems of inkjets.[116]

We have found that the small spots produced by the microplotter, along with its capability to codeposit analyte and matrix, create a noticeable enhancement in the sensitivity of MALDI mass spectrometry.[112] We attempted to detect the presence of cytochrome C, a 12 kDa protein, with α-cyano-4-hydroxycinnamic acid (α-CHCA) as the matrix material by depositing both in different fashions. We first loaded a solution of cytochrome C into the microplotter and dispensed grids of protein spots (with diameters ranging from 20 to 70 μ m) over one half of a stainless steel MALDI target. We then put down a continuous film of protein over the other half by drawing a series of overlapping lines. The dispenser was then emptied, cleaned, and loaded with α-CHCA dissolved in methanol. The same sort of half-and-half deposition was performed with the matrix, only the halves were rotated by 90 degrees relative to the dispensed protein. This formed four quadrants: spots of matrix deposited on spots of protein, spots of matrix on a continuous coat of protein, a continuous coat of matrix on spots of protein, and a continuous coat of matrix on a continuous coat of protein.

The MALDI target was then loaded into a mass spectrometry system. As described above, a laser pulse causes the ejection of matrix and analyte whereever it lands on the surface of the MALDI target. In our case, the laser beam spot size was 70 μm in diameter. For the spotted quadrants, the beam was lined up so that it was centered with a deposited spot before firing. Mass spectra were gathered from each of the quadrants, with samples shown in Figures 6.11 and 6.12. The peak on the far right of each spectrum corresponds to the z=1 charge state of the protein, with the next peak to the left corresponding to the z=2 charge state. Both the uniformly coated protein / uniformly coated matrix (Fig. 6.11a) and uniformly coated protein / spotted matrix (Fig. 6.11b) spectra show broad peaks for the protein. The spotted protein / uniformly deposited matrix spectrum (Fig. 6.11c) has noticeably sharper peaks, especially for the z=2 charge state. However, the most dramatic improvement comes from small spots of protein covered with spots of matrix. Figure 6.12a shows slightly sharper peaks for a 70 μm wide spot of matrix and protein, but the 20 μm wide spot of matrix and protein in Figure 6.12b has not only much sharper peaks than any of the other spectra, but the peaks are also more intense than any of the others.

This peak sharpening, combined with a much higher signal-to-noise ratio, indicates that identification of compounds with similar molecular weights can be performed more reliably with $<$20 μm wide microplotter-deposited spots than with traditional hand-spotting. Regular grids of consistent analyte and matrix volume should also aid in quantitation. Even though no enhancement in sensitivity is caused by the continuous coatings of matrix and protein deposited by the microplotter, the ability to make very uniform films of either could also help produce more consistent results from mass-spectrometry-based experiments.

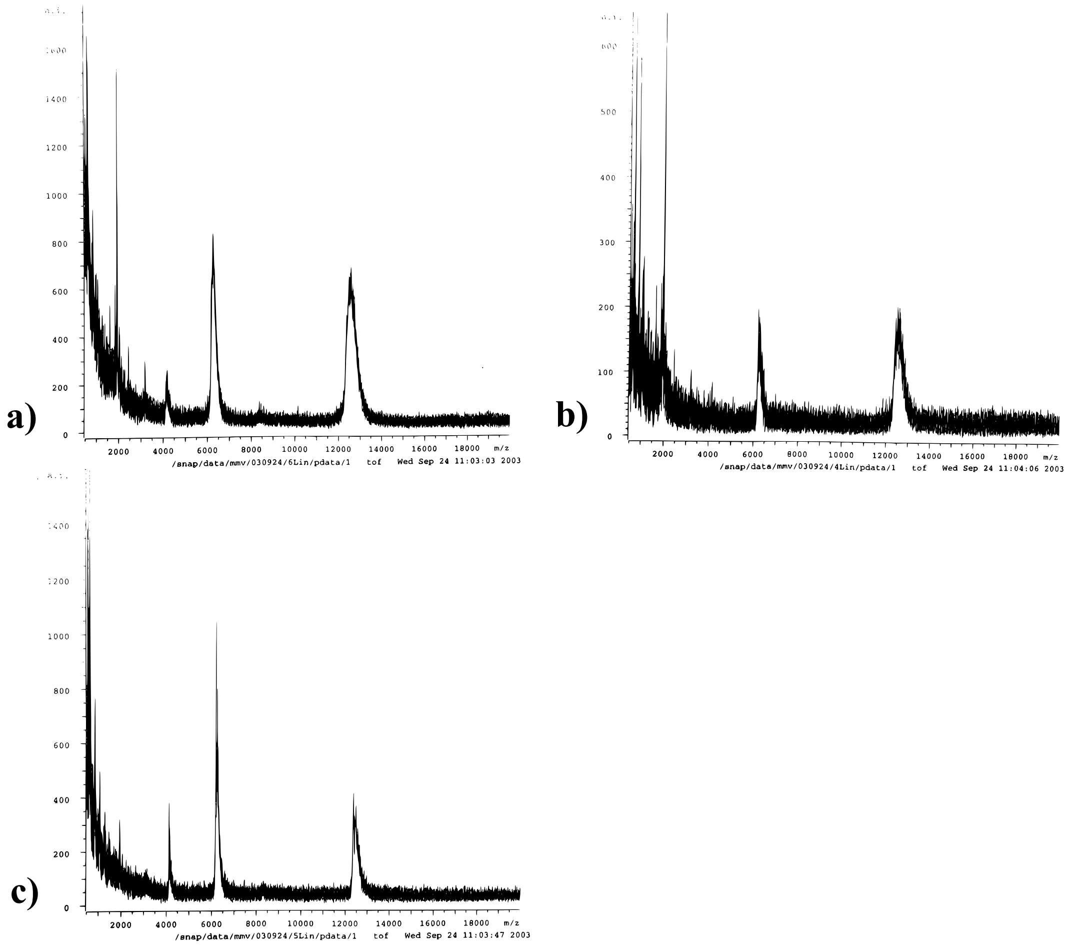

Figure 6.11: Mass spectra gathered from deposited cytochrome C. These spectra were gathered from areas where the microplotter deposited a) a continuous coating of protein covered by a continuous coating of matrix, b) a continuous coating of protein covered by a spotted grid of matrix, and c) a spotted grid of protein covered by a continuous coating of matrix.

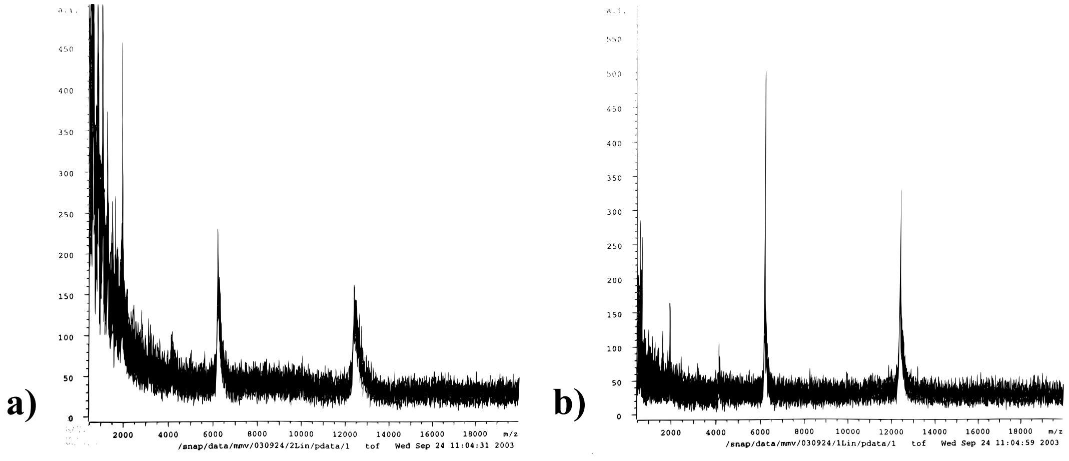

Figure 6.12: Additional mass spectra gathered from deposited cytochrome C. These spectra were gathered from the area of the MALDI target where spots of matrix were deposited on top of spots of protein. Two sizes of spots were examined: a) 70 μm spots exhibited a slight reduction in peak width, but b) 20 μm spots showed very narrow and intense peaks.

6.5 Polymer-based electronics

As mentioned in Chapters 1 and 2, the use of conducting or semiconducting polymers in the fabrication of flexible and/or inexpensive microelectronic circuits currently is an area of significant interest. Many of these polymers are capable of being dissolved in common organic solvents or even water, therefore they can be deposited using low-cost fluid dispensers, such as the microplotter technology described in this dissertation. Directly delivering these conducting polymers to specific locations on a surface would remove the need for prefabricated masks and allow for rapid prototyping of electronic circuits. Solution printing of polymer circuits would also be a low-cost manufacturing process ideally suited for a range of products, such as radio frequency identification (RFID) tags or polymer-based light-emitting diodes (LEDs).

We have fabricated a working polymer LED device, whose configuration is shown in Figure 6.13, using a microplotter and solution-suspended conjugated polymers. Our polymer LED device was constructed by first evaporating a set of indium tin oxide (ITO) stripes on a glass substrate to act as one set of electrical contacts. ITO was used due to its transparency and its ability to serve as a good anode. On top of the ITO, a conducting polymer layer of poly (3,4-ethylenedioxythiophene) (PEDOT) was directly drawn using a microplotter. PEDOT improves device performance by creating an ohmic contact with the LED polymer. To mask off all but a small area for use as the LED, poly(methylmethacrylate) (PMMA) was deposited by a microplotter on the ITO contact in two rectangular areas separated by a thin gap. PMMA is insulating and thus should prevent any charge from reaching the LED polymer. Poly(9,9'-dioctylfluorine) (PF8), a widely used blue-emitting polymer LED material[117], was then deposited over the area not covered by PMMA using a microplotter. It is critical that this film be as thin as possible in order to have the most efficient light emission. A microplotter can easily deposit a film thin enough to yield a very good device (film thicknesses below 50 nm have been measured). Calcium electrodes were evaporated on top of the polymer layers at a right angle to the ITO electrodes and a covering metal layer was evaporated on top of the calcium. Calcium was used because it is a good cathode material, but it needed to be covered since it can be oxidized quickly by atmosphere. With the application of a high enough electrical voltage between the two electrodes, light was emitted down through the ITO and glass side of the device (the metal on top of the calcium helped reflect light back that was emitted in the opposite direction) where the calcium and ITO electrodes cross to form an LED device.

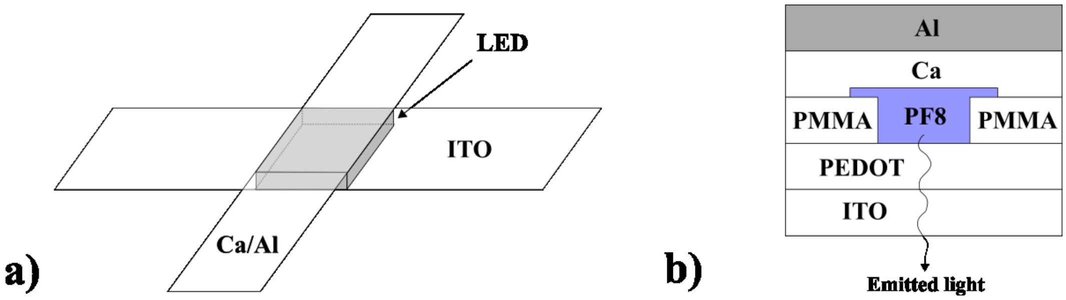

Figure 6.13: A schematic of a polymer LED fabricated using a microplotter. a) A view of the crossbar electrode structure, with the working LED sandwiched in between. b) A cross-section of the LED structure.

Figure 6.14a shows the device area with 0 V applied between the contacts, and Figure 6.14b shows the bright blue light given off with the application of 6 V. The PMMA layers did not fully insulate the LED polymer, so some light was generated in the area of the device that should have been masked off. The area of interest in this particular device is the thin line to the right, which is approximately 40 microns wide and is emitting a very bright blue. In this configuration with these particular materials the LED has a very short lifetime (1-2 days), which is far below that needed for commercial applications such as in flexible display screens. Increasing the lifetime of polymer microelectronic devices is currently the primary focus of researchers in the field, but the new materials or device configurations developed to address this problem can be deposited with a microplotter just as easily as in this particular LED.

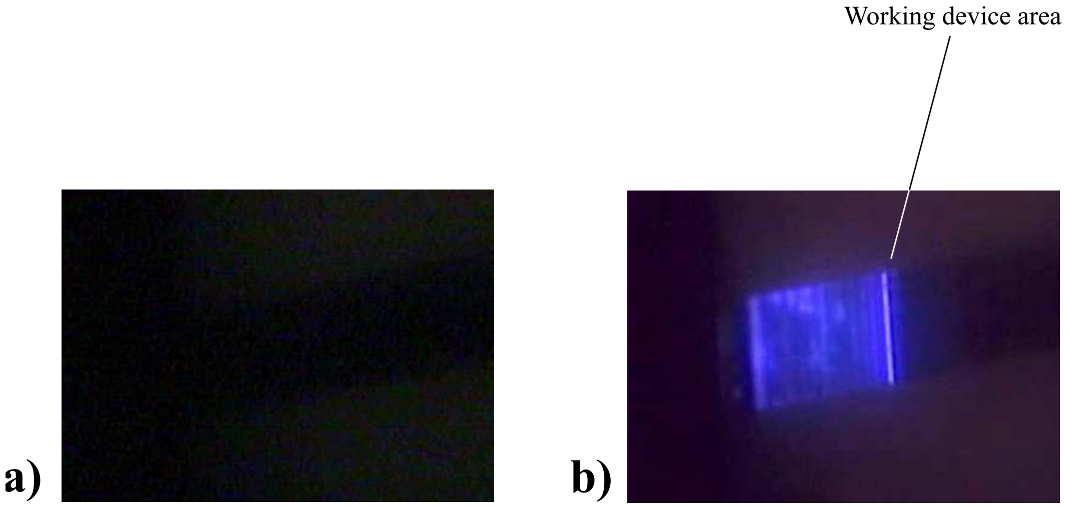

Figure 6.14: Images of a working polymer LED. A potential of a) 0 V and b) 6 V was applied to the previously described polymer LED. When on, blue light was emitted from a 40 μm wide strip (indicated in the image). The rest of the light in the image was due to scattering in the polymer layers or poor electrical insulation by the PMMA.

6.6 Conclusion

In summary, we have demonstrated several additional applications of the microplotter and cold-plasma-based treatment technologies. Proof-of-principle protein microarrays were constructed, including one capable of detecting a common chicken pathogen. Both aldehyde- and epoxide-terminated polymers were produced using cold plasmas and vapor-phase reactions similar to those described in Chapter 5. Carbon nanotubes, molecules with novel electrical and mechanical properties, were functionalized with epoxide groups using one of these plasma-based processes. Nanotubes were also grown by depositing small spots of catalyst at select locations on a surface. It was found that sub-20 μm diameter spots of protein and matrix greatly improved the sensitivity of MALDI mass spectrometry. Finally, we fabricated a working blue LED from conjugated polymer solutions deposited using a microplotter.

Appendix A

Lattice gas fluid modeler

A.1 Introduction

Several fluid modeling simulations were presented in Chapter 4. We performed these simulations using a custom-written fluid modeling program that uses lattice gas automata to simulate a two-dimensional fluid. When compared with traditional Navier-Stokes-based finite element simulations, lattice gas simulations can be performed much more rapidly on modern computing hardware and rely on none of the assumptions (such as an incompressible fluid) used to derive the Navier-Stokes equations. We describe here the lattice gas model used to create this computer program, provide some demonstrations of its capabilities, show how to operate the program, and finally detail the structure and source code of the program.

A.2 Lattice gas automata

Fluids are one of the most difficult physical phenomena to model. Analytical descriptions of fluid behavior are only available for the simplest of conditions and many of the equations used to predict fluid properties are empirical or semiempirical in origin. As available computational power has increased in recent years, finite element models have been used to handle most fluid simulations. Finite element models break down a simulated volume into small cells and then apply analytical expressions, or their stripped-down versions, within each of those cells. They allow for complex phenomena to be explored, even if no known solutions exist for the governing equations involved. Unfortunately, they also require a significant amount of processing power because these often complex equations must be run for each cell in the simulation grid. Additionally, these equations almost always require floating point math, computational operations that are very slow in off-the-shelf hardware. Finally, assumptions are often made in order to derive the equations used within each of the cells.

Invented by Frisch, Hasslacher, and Pomeau in 1986[118], hexagonal lattice gas automata provide a means of simulating fluids using simple, fast integer math while avoiding the limiting assumptions made in deriving the classical Navier-Stokes equations. Figure A.1 demonstrates the way in which a lattice gas automaton works. First, the area of fluid to be represented is turned into a hexagonal grid. At each point in the grid, up to six particles are placed, with unit velocities in one of the six possible directions. Only one particle at a time may have a specific velocity at a point in the grid. To proceed one unit forward in time, each of the particles moves a unit forward, according to its velocity. If two or more particles meet at a single point in the grid, collision rules are applied to determine their new velocities. It has been shown that if these rules are chosen to preserve mass and momentum, a fine enough hexagonal lattice will satisfy the Navier-Stokes equations while still being able to simulate compressible fluids.[119] The hexagonal grid is critical for accurate fluid modeling because square or other polygonal grids do not lead to sufficient isotropy to satisfy the Navier-Stokes equations.

The simplicity of this model leads to its computational advantages. Each point on the hexagonal lattice can have up to six particles at it, one for each direction, so the state of that point can be represented in memory by a single byte (eight bits). In contrast, finite element methods may require over ten times that amount of memory for each element in their grid. Bitwise arithmetic can be used to move particles from one cell to the next, computing operations that are among the fastest available on modern processors. There are only 64 collision rules, so a simple lookup table can be used to simply replace every particle collision with its end result, no calculations needed. Additionally, the calculations performed on a cell are independent of those performed in any other cell, so many cells can be computed in parallel during a given time step. Recently, commodity central processing units (CPUs) have gained the ability to perform parallel integer and bitwise arithmetic operations. Popular graphics cards have chips dedicated to just these sort of parallel operations. Thus, by taking advantage of this, parallel-processed lattice gas models may be able to run more than an order of magnitude faster than their serially-processed equivalent.

However, lattice gas models are not perfect. There is no known three-dimensional lattice that will satisfy the Navier-Stokes equations in the way that the hexagonal lattice does in two dimensions. As will be seen in the following section, the simulated fluids show the effects of Brownian motion, even when very fine meshes are used. Finally, the underlying discrete particle velocities can cause unusual behaviors in the larger, continuum scale for certain sets of conditions. To work around these, a lattice gas model using the Boltzmann equation to deal with distributions of particle velocities, rather than individual discrete ones, has been proposed.[120] This model is currently in wide use, but it has sacrificed the computational advantages of the simpler lattice gas model due to the need for floating point math and equations within each cell.

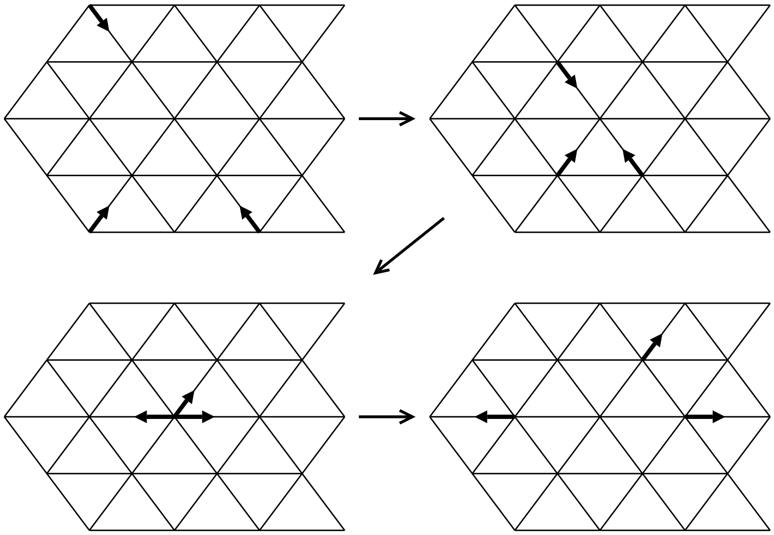

Figure A.1: An example of a lattice gas. Space is represented by a hexagonal grid, on which particles (the dark arrows) move with some discrete velocity. At each time step, the particles move one unit forward. If more than one particle meets at a single point in space, a collision occurs which conserves mass and momentum.

A.3 Demonstrations

As some simple demonstrations of the capabilities of this modeling program, two simulations of common fluid flow problems were set up and run. The first is flow past a flat plate. For this example, a wall was placed in the middle of the simulation grid. Particles were injected from the left side of the grid, inducing a flow that moved to the right. As this flow encountered the wall, it passed around it and turbulent eddies were formed. To visualize this, the program took the average velocity of 20 x 20 squares of cells and displayed vectors representing those averages. The result is shown in Figure A.2a and appears to qualitatively match real-world fluid behavior.

A simulation was also performed for one of the most common fluid flow problems in chemical engineering or even microfluidics, flow inside a pipe. Two horizontal walls

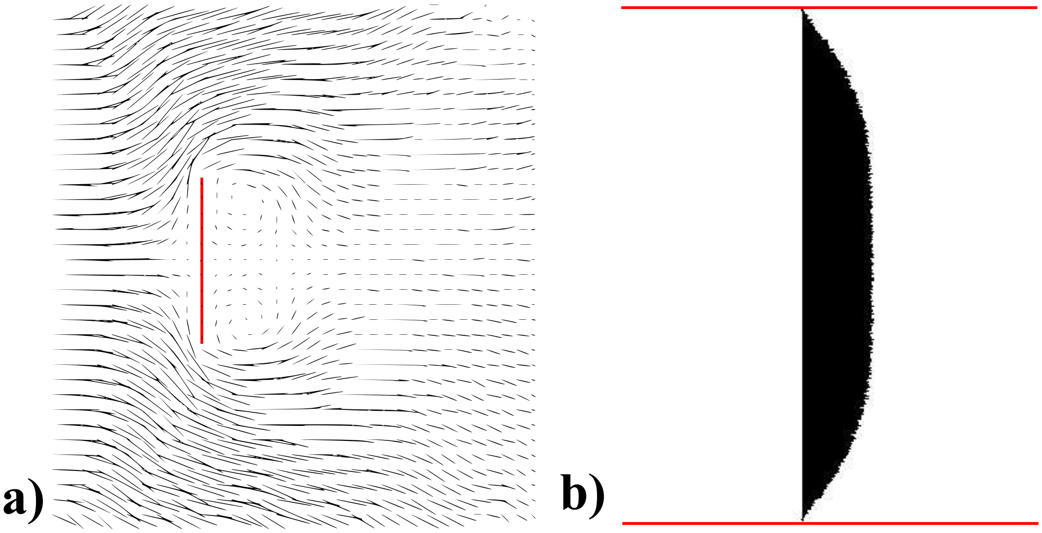

Figure A.2: Two demonstrations of lattice gas simulations. a) A constant stream of fluid is flown past a solid plate. b) A velocity profile is shown of fluid flowing from left to right through a pipe.

were placed across the length of the simulation grid and, as above, particles were injected from the left to induce the fluid to flow to the right. At set intervals, the velocity profile across a slice of the pipe was measured. This was done by averaging the x component of the fluid velocities across blocks of cells 40 cells wide and 1 cell tall, each cell along a vertical stripe that cut through the middle of the pipe. The output from this can be seen in Figure A.2b. This velocity profile is that of either turbulent (plug-like) flow through a pipe or laminar flow that has not fully developed. Since the size scale of the hexagonal grid is arbitrary, these sort of fluid simulations can be used for both macroscopic flow (pipes, air flow, etc.) as well as microfluidic applications (microchannels, mixers, etc.).

A.4 How to operate

A.4.1 Installation

This software is known to work under Linux, specifically, Gentoo Linux with a 2.6.8 kernel, but it should run on other Unices.

To compile from source on Linux, first make sure that the GD library version 2.0 or greater is installed. If not, you can get the latest versions from http://www.boutell.com/gd/. Next, obtain the latest source tarball of the fluid modeler from the author (who can be contacted at bjlarson2@students.wisc.edu). Then untar the downloaded archive using tar -jxvf latticegas-XXXX.tar.bz2 (where XXXX is the date stamp of the latest version). Change into the fluidmodeler directory that was created and type the following commands

```
make
make install
```

to compile and install the program.

A.4.2 Running the program

The fluid modeling program uses PNG bitmaps to input the starting conditions for the fluid model. Some examples of these are in the fluidmodeler/examples directory in the source code package. To run a simulation, you must first create a PNG bitmap using any available drawing program (the Gimp is highly recommended for this purpose). Then, you can simply run the simulation by typing

```
fluidmodeler [filename]
```

where [filename] is the name of your PNG simulation file.

The results of the simulation will be placed in the ./output directory.

A.5 Structure of program

The fluid modeler is written in C++ and is constructed from four files.

collisions.h contains the lookup table used to perform the particle collisions at points on the grid. It also has descriptions of the way that we have chosen to represent the particles at a point on the grid as bits in a byte.

main.cpp is simply the file that contains the wrapper code for running the simulation at the command line.

latticegas.cpp and *latticegas.h* are where the majority of this program is defined. These two files code for a class called LatticeGas that contains all the lattice gas data structures and methods for loading simulation conditions, running the simulation, and outputting the results.

A.6 C++ code

A.6.1 collisions.h

```
/***************************************************************************
                          collisions.h  -  description
                             -------------------
    begin                : Sun Jan 18 2004
    copyright            : (C) 2004 by Brad Larson
    email                : bjlarson2@students.wisc.edu
 ***************************************************************************/

/** An array defining the collision conditions for the existing fluid model.
    6-bits define velocities of particles at a lattice location:
                4     5
                 \   /
               3__\ /__0
                  / \
                 /   \
                2     1
  Byte(char): XX543210
  Where XX is a marker indicating whether this is a fluid source, a wall, or
      a normal cell
*/
```

```
/*   0-particle, no change
        0,
    One-particle, no change
        1,2,4,8,16,32,
    Two-particle cases where there’s no change
        3,5,6,10,12,17,20,24,33,34,40,48
    Two-particle cases where there’s a simple counterclockwise rotation (
        right shift)
    In:
        36,18,9
    Out:
        18,9,36
    Three-particle cases where there’s a simple counterclockwise rotation (
        right shift)
    In:
        21,42
    Out:
        42,21
    Three-particle cases where there’s a vertical flip
    In:
        22,11,37,50,25,44,13,38,19,41,52,26
    Out:
        13,38,19,41,52,26,22,11,37,50,25,44
    Three-particle cases where there’s no change
        7,14,28,35,49,56
    Four-particle cases where there’s a simple counterclockwise rotation (
        right shift)
    In:
        27,45,54
    Out:
        45,54,27
    Four-particle cases where there’s no change
        15,23,39,51,57,60,29,46,43,53,58,30
    Five-particle, no change
        31,47,55,59,61,62
    Six-particle, no change
        63
*/
unsigned char collisionTable[64] =
{
    0,1,2,3,4,5,6,7,8,
    36,10,38,12,22,14,15,16,17,9,
    37,20,42,13,23,24,52,44,45,
    28,29,30,31,32,33,34,35,18,
    19,11,39,40,50,21,43,26,54,
    46,47,48,49,41,51,25,53,27,
    55,56,57,58,59,60,61,62,63
};

/* The series of masks for grabbing bits from surrounding cells
    Bit organization: [0 0] <0 0 0 0 0 0>
    <> - velocity bits
    [] - type of cell:
```

```
    00 - fluid
    01 - solid wall  (>=64)
    10 - fluid source  (>=128)

    11 - vibrating wall (>=192)
    */

#define U   32
#define UL  16
#define L   8
#define BL  4
#define B   2
#define R   1

/* A set of arrays that contain all n-particle sites */
unsigned char oneParticle[6] =
{
    1,2,4,8,16,32
};

unsigned char twoParticle[15] =
{
        3,5,6,10,12,17,20,24,33,34,40,48,
        36,18,9
};

unsigned char threeParticle[20] =
{
        21,42,
        22,11,37,50,25,44,13,38,19,41,52,26,
        7,14,28,35,49,56
};

unsigned char fourParticle[15] =
{
        27,45,54,
        15,23,39,51,57,60,29,46,43,53,58,30
};

unsigned char fiveParticle[6] =
{
        31,47,55,59,61,62
};
```

A.6.2 latticegas.h

```
/***************************************************************************
                          latticegas.h  -  description
                             -------------------
    begin                : Thu Jan 20 2004
    copyright            : (C) 2004 by Brad Larson
    email                : bjlarson2@students.wisc.edu
 ***************************************************************************/
```

```
#ifndef LATTICEGAS_H
#define LATTICEGAS_H
#include  "gd.h"
#include <sys/time.h>

struct avgVelocity
{
    long xPos;
    long yPos;
};


/**This class defines a Lattice Gas for use in fluid modeling
  *@author Brad Larson
  */

class LatticeGas
{
public:
    LatticeGas();
    ˜LatticeGas();
    /** This function loops through processing all the frames needed */
    void processingLoop(long numLoops);
    /** This function does one step in the processing of the lattice gas, non
        -SSE2 implementation */
    void serialProcessingStep();
    /** This processes a bitmap file and fills out the internal array */
    void loadBitmap(string fileName);
    /** This inputs the values for the simulation from an XML file */
    void loadXML(string fileName);
    /** This writes the grid to a bitmap, in one of several styles */
    void writeBitmap(string fileName, int mode);
    /** This writes the grid to a scalable vector graphic (SVG) file, in one
        of several styles */
    void writeSVG(string fileName, int mode);
    /** This dumps raw data from the run, in one of several styles */
    void dumpRawData(string fileName, int mode);
    /** This processes a PNG bitmap file and fills out the internal array */
    int processPixel(gdImagePtr im, int x, int y);

    /** This returns an array of x,y pairs corresponding to the average
        velocity of a block of the width and height specified */
    avgVelocity * averageVelocities(int blockSize);
    /** This returns an array of average particle densities */
    long * averageDensities(int blockSize);
    /** This gets a velocity profile along a line */
    long * lineVelocities(int blockSize);

    /** This gives a random number within a range*/
    int boundedRandom(int range);

    /** The number of time steps to run the simulation for */
    long timeSteps;
```

```
/**This holds the array that represents the entire grid */
unsigned char *theArray;

/** This holds the array of particle velocities over time */
int *theArrayXVelocities;
short *theArrayYVelocities;

long xSize;
long ySize;
long arraySize;

/** These are values that store vibrational data */
bool isVibrating;
bool isSolidPlate;
double vibrationRadians;
short vibrationAmplitude;
short amplitudeMax;
int vibrationPeriod;
long currentStep;
char vibrationType;
int startVibration;
int stopVibration;
short *travelingVibrationState;
bool taperedVibration;
double vibrationSlope;
long vibrationIntercept;


/** Are we going to use a bitmap or a vector drawing for the pattern
    specification? */
bool isBitmap;
/** The name of the bitmap to use, if one exists */
string bitmapName;

/** Values for outputting a density map */
bool outputDensity;
int densityCellSize;
long densityInterval;

/** Values for outputting a velocity map */
bool outputVelocity;
int velocityCellSize;
long velocityInterval;

/** Values for outputting a raw bitmap */
bool outputRawBitmap;
long rawBitmapInterval;

/** Values for outputting a line profile */
bool outputLineProfile;
int lineProfileCellSize;
long lineProfileInterval;
long lineProfileLocation;
```

```
    bool lineProfileAutoscale;
    bool lineProfileDumpRawData;

};

#endif
```

A.6.3 latticegas.cpp

```
/***************************************************************************
                          latticegas.cpp  -  description
                             -------------------
    begin                : Thu Jan 20 2004
    copyright            : (C) 2004 by Brad Larson
    email                : bjlarson2@students.wisc.edu
 ***************************************************************************/


#include <iostream>
#include <fstream>
using namespace std;

#include <stdlib.h>
#include "latticegas.h"
#include "collisions.h"

// Converts integer values to a string
string intToString(long val)
{
    if (val < 0)
    {
        val = val * (-1);
        return "-" + intToString (val);
    }
    else if (val < 10 )
    {
        switch (val)
        {
            case 0:  return "0"; break;
            case 1:  return "1"; break;
            case 2:  return "2"; break;
            case 3:  return "3"; break;
            case 4:  return "4"; break;
            case 5:  return "5"; break;
            case 6:  return "6"; break;
            case 7:  return "7"; break;
            case 8:  return "8"; break;
            case 9:  return "9"; break;
        }
    }
    else
    {
        return intToString(val / 10) + intToString (val % 10);
    }
    return "error";
}

LatticeGas::LatticeGas()
{
#if wxUSE_LIBPNG
  wxImage::AddHandler( new wxPNGHandler );
#endif
}

LatticeGas::˜LatticeGas()
{
    delete[] theArray;
    delete[] theArrayXVelocities;
    delete[] theArrayYVelocities;
    if (travelingVibrationState != NULL)
        delete[] travelingVibrationState;
}

void LatticeGas::processingLoop(long numLoops)
{
    struct timezone the_zone;
    struct timeval beforeSerial, afterSerial;
    struct timeval duringSerial;
    long serialtimediff = 0;

    string fileName;
    long counterino = 10000;

    // Get the first timestamp
    gettimeofday(&beforeSerial, &the_zone);
    // Seed random number generator with the time of day
    srand(beforeSerial.tv_usec);

    writeBitmap("output/initialconditions.png",0);

    /** These are values that store vibrational data */
    if (isVibrating)
    {
        if (!isSolidPlate)
        {
            travelingVibrationState = new short [stopVibration -
                startVibration];
            for (int I=0; I < (stopVibration - startVibration); I++)
            {
                travelingVibrationState[I] = 0;
            }
        }
        if (taperedVibration)
        {
            short vibrationAmplitudeMin = 0;
            vibrationSlope = ((double)(vibrationAmplitude-
                vibrationAmplitudeMin))/((double)(stopVibration-
                startVibration));
            vibrationIntercept = vibrationAmplitudeMin - (startVibration * (
                vibrationAmplitude - vibrationAmplitudeMin)) / (stopVibration
                 - startVibration);
        }

    }

    for (long I=0; I<numLoops; I++)
    {
        // Run one processing step, using non-parallel computations
        serialProcessingStep();

        currentStep++;
        // Output the relevant snapshots of the simulation
        if(outputDensity)
        {

            if((I % densityInterval) == 0)
            {
                fileName = "output/densities" + intToString(counterino) + ".
                    png";
                writeBitmap(fileName,1);
            }
        }
        if(outputVelocity)
        {

            if((I % velocityInterval) == 0)
            {
                fileName = "output/velocities" + intToString(counterino) + ".
                    png";
                writeBitmap(fileName,2);
                fileName = "output/velocities" + intToString(counterino) + ".
                    svg";
                writeSVG(fileName,2);
            }
        }
        if(outputRawBitmap)
        {
            if((I % rawBitmapInterval) == 0)
            {
                fileName = "output/rawbitmap" + intToString(counterino) + ".
                    png";
                writeBitmap(fileName,0);
            }
        }
        if(outputLineProfile)
        {
            if((I % lineProfileInterval) == 0)
            {
                fileName = "output/lineprofile" + intToString(counterino) + "
                    .png";
                writeBitmap(fileName,4);
                if (lineProfileDumpRawData)
                {
                    fileName = "output/lineprofile" + intToString(counterino)
                         + ".csv";
                    dumpRawData(fileName,0);
                }
            }
        }

        cout << (100*I)/numLoops;
        cout << "%␣|␣Frame:␣" << I;
        gettimeofday(&duringSerial, &the_zone);
        cout << "␣|␣Time␣running:␣" << duringSerial.tv_sec-beforeSerial.
            tv_sec << "s";
        cout << "␣|␣Time␣remaining:␣" << (duringSerial.tv_sec-beforeSerial.
            tv_sec)*(numLoops-I)/currentStep << "s\r";
        counterino++;

    }
    // Get the last timestamp
    gettimeofday(&afterSerial, &the_zone);
    serialtimediff = (afterSerial.tv_sec - beforeSerial.tv_sec)*1000000+
        afterSerial.tv_usec - beforeSerial.tv_usec;
    cout << "\nProcessing␣time:␣" << serialtimediff <<"\n";

    // Dump the state at the end of the simulation
    fileName = "output/finalconditions.png";
    writeBitmap(fileName,0);
}

void LatticeGas::serialProcessingStep()
{
    // New array resulting from the one-step transformation of the old array
    unsigned char *theNewArray = new unsigned char[arraySize];
    // Rows need to alternate in order to map hexagonal lattice to square
        arrays
    bool rowAlignment;

    long upperLeftPos;
    long upperPos;
    long bottomLeftPos;
    long bottomPos;

    short vibFrac;
    bool positiveVibDirection;

    // Need this for averages of velocities
    unsigned short averageCounter = 4000;
    unsigned short averageCounter2 = 0;
```

```
if (isVibrating)
{
    // Calculate vibrational value, in radians
    vibrationRadians = 2.0 * 3.14159 * (double)currentStep / (double)
       vibrationPeriod;

    if (isSolidPlate)
    {
        // Switched to cosine to match Stokes' analysis
        vibFrac = (short)(vibrationAmplitude * cos(vibrationRadians));
    }
    else
    {
        // Wave propagates by one cell
        for (int I=1; I < (stopVibration - startVibration); I++)
        {
            travelingVibrationState[(stopVibration - startVibration) - I
                ] = travelingVibrationState[(stopVibration -
                startVibration) - I - 1];
        }
        // Inject new vibration state
        travelingVibrationState[0] = (short)(vibrationAmplitude * cos(
           vibrationRadians));
    }
}

// Do special case for first row (there are no cells above this one)
// Don't process top row, bottom row, and left and right columns, simply
   blank them out in new grid
for (long J=0;J<xSize;J++)
{
    theNewArray[J] = theArray[J];
    theNewArray[(ySize-1)*xSize + J] = theArray[(ySize-1)*xSize + J];
}
rowAlignment = false;
// Process middle rows
for (long J=1;J<(ySize-1);J++)
{
    // In order to maintain hexagonal grid within a linear array, need to
         alternate way rows interact
    if (rowAlignment)
    {
        upperLeftPos = xSize + 1;
        upperPos = xSize;
        bottomLeftPos = xSize - 1;
        bottomPos = xSize;
    }
    else
    {
        upperLeftPos = xSize;
        upperPos = xSize - 1;
        bottomLeftPos = xSize;
        bottomPos = xSize + 1;
    }
```

```
rowAlignment = !rowAlignment;
// Do left-to-right periodic boundary conditions
theNewArray[J*xSize] = theArray[((J+1)*xSize)-2];
theNewArray[((J+1)*xSize)-1] = theArray[J*xSize+1];

long xPositionHolder = 0;
// Middle rows/columns
for (long I=(J*xSize)+1; I<(((J+1)*xSize)-1); I++)
{
    // Deal with a vibrating wall
    if (theArray[I] >= 192)
    {
        // Movement
        unsigned char holder = 0;
        unsigned char holder2 = 0;
        unsigned char holder3 = 0;
        // Mask off all necessary bits to grab velocities from
            surrounding cells
        holder = theArray[I - upperLeftPos] & B;
        holder = holder | (theArray[I - upperPos] & BL);
        holder = holder | (theArray[I - 1] & R);
        holder = holder | (theArray[I + 1] & L);
        holder = holder | (theArray[I + bottomLeftPos] & U);
        holder = holder | (theArray[I + bottomPos] & UL);
        // Check if supposed to move, based on sine-wave
            probabilities

        if (taperedVibration)
        {
            // Calculate the amplitude of vibration as a function of
                X position
            if(isSolidPlate)
            {
                vibFrac = (long)((vibrationSlope * xPositionHolder +
                    vibrationIntercept) * cos(vibrationRadians));
            }
            else
            {
                // Check to make sure this vibrating element is
                    between the bounds
                if ( (startVibration < xPositionHolder) || (
                    stopVibration >= xPositionHolder))
                {
                    vibFrac = 0;
                }
                else
                {
                    vibFrac = (long)(((vibrationSlope *
                        xPositionHolder + vibrationIntercept) *
                        travelingVibrationState[xPositionHolder -
                        startVibration]) / vibrationAmplitude);
                }
            }
        }
```

```
else if (!isSolidPlate)
{
    // Check to make sure this vibrating element is between
        the bounds
    if ( (startVibration < xPositionHolder) || (stopVibration
         >= xPositionHolder))
    {
        vibFrac = 0;
    }
    else
    {
        vibFrac = travelingVibrationState[xPositionHolder -
            startVibration];
    }
}

if (vibFrac < 0)
{
    vibFrac = -vibFrac;
    positiveVibDirection = false;
}
else positiveVibDirection = true;

if (boundedRandom(amplitudeMax) < vibFrac)
{
    if (positiveVibDirection) // Right to left, up to down
    {
        // Mask off most significant bits, shift them 3 to
            the right
        holder2 = (holder & 56) >> 3;
        // Mask off least significant bits, shift them 3 to
            the left and add them in
        holder2 = holder2 | ((holder & 7) << 3);
        holder3 = holder3 | ((U & holder2) & ((UL & holder2)
             << 1)); // Cell OR (UL AND UR)
        holder3 = holder3 | ((UL & holder2) | ((U & holder2)
             >> 1)); // Cell OR (UL OR UR)
        holder3 = holder3 | ((BL & holder2) | ((B & holder2)
             << 1)); // Cell OR (LL OR LR)
        holder3 = holder3 | ((B & holder2) & ((BL & holder2)
             >> 1)); // Cell OR (LL AND LR)
        holder3 = holder3 | (9 & holder2); // Cell OR (L+R);
        holder3 = holder3 | 192; // Maintain vibrational
            status
        theNewArray[I] = holder3;
    }
    else // Left to right, down to up
    {
        // Mask off most significant bits, shift them 3 to
            the right
        holder2 = (holder & 56) >> 3;
        // Mask off least significant bits, shift them 3 to
            the left and add them in
        holder2 = holder2 | ((holder & 7) << 3);
```

```
            holder3 = holder3 | ((U & holder2) | ((UL & holder2)
                 << 1)); // Cell OR (UL AND UR)
            holder3 = holder3 | ((UL & holder2) & ((U & holder2)
                 >> 1)); // Cell OR (UL OR UR)
            holder3 = holder3 | ((BL & holder2) & ((B & holder2)
                 << 1)); // Cell OR (LL OR LR)
            holder3 = holder3 | ((B & holder2) | ((BL & holder2)
                 >> 1)); // Cell OR (LL AND LR)
            holder3 = holder3 | (9 & holder2); // Cell OR (L+R);
            holder3 = holder3 | 192; // Maintain vibrational
                status
            theNewArray[I] = holder3;
        }
    }
    else // Just behave like a normal wall
    {
        // Collisions
        // Bounce surrounding particles off wall by shifting all
             3 bits to the right
        // Mask off most significant bits, shift them 3 to the
            right
        holder2 = (holder & 56) >> 3;
        // Mask off least significant bits, shift them 3 to the
            left and add them in
        holder2 = holder2 | ((holder & 7) << 3);
        // Preserve bits indicating solid element
        holder2 = holder2 | 192;
        theNewArray[I] = holder2;
    }
}
// Deal with a fluid source
else if (theArray[I] >= 128)
{
    // Keep supplying particles from the cell
    theNewArray[I] = theArray[I];
}
// Deal with a solid wall
else if (theArray[I] >= 64)
{
    // Movement
    unsigned char holder = 0;
    unsigned char holder2 = 0;
    // Mask off all necessary bits to grab velocities from
        surrounding cells
    holder = theArray[I - upperLeftPos] & B;
    holder = holder | (theArray[I - upperPos] & BL);
    holder = holder | (theArray[I - 1] & R);
    holder = holder | (theArray[I + 1] & L);
    holder = holder | (theArray[I + bottomLeftPos] & U);
    holder = holder | (theArray[I + bottomPos] & UL);
    // Collisions
    // Bounce surrounding particles off wall by shifting all 3
        bits to the right
    // Mask off most significant bits, shift them 3 to the right
```

```
                holder2 = (holder & 56) >> 3;
                // Mask off least significant bits, shift them 3 to the left
                    and add them in
                holder2 = holder2 | ((holder & 7) << 3);
                // Preserve bits indicating solid element
                holder2 = holder2 | 64;
                theNewArray[I] = holder2;
            }
            else
            {
                // Movement
                unsigned char holder = 0;
                // Mask off all necessary bits to grab velocities from
                    surrounding cells
                holder = theArray[I - upperLeftPos] & B;
                holder = holder | (theArray[I - upperPos] & BL);
                holder = holder | (theArray[I - 1] & R);
                holder = holder | (theArray[I + 1] & L);
                holder = holder | (theArray[I + bottomLeftPos] & U);
                holder = holder | (theArray[I + bottomPos] & UL);

                // Collisions
                theNewArray[I] = collisionTable[holder];
            }

            // Measure the velocities of the cell
            if (averageCounter > xSize)
            {
                unsigned char holder2;
                holder2 = theNewArray[I] << 2;
                theArrayXVelocities[averageCounter2] += holder2 >> 7;
                holder2 = theNewArray[I] << 3;
                theArrayXVelocities[averageCounter2] -= holder2 >> 7;
                holder2 = theNewArray[I] << 4;
                theArrayXVelocities[averageCounter2] -= (holder2 >> 7)*2;
                holder2 = theNewArray[I] << 5;
                theArrayXVelocities[averageCounter2] -= holder2 >> 7;
                holder2 = theNewArray[I] << 6;
                theArrayXVelocities[averageCounter2] += holder2 >> 7;
                holder2 = theNewArray[I] << 7;
                theArrayXVelocities[averageCounter2] += (holder2 >> 7)*2;
                averageCounter = 0;
                averageCounter2++;
            }
            averageCounter++;
            xPositionHolder++;
        }
    }

    delete[] theArray;
    theArray = theNewArray;
    theNewArray = NULL;
}
```

```
/** This processes a PNG bitmap file and fills out the internal array */
void LatticeGas::loadBitmap(string fileName)
{
    gdImagePtr im = 0;
    FILE *pngin;

    // Load image from file
    pngin = fopen (fileName.c_str(), "rb");
    if (!pngin) // PNG file missing
    {
        cout << "Error:␣" << fileName.c_str() << "␣does␣not␣exist.␣␣Please␣
            see␣to␣this.\n";
        im = 0;
    }

    im = gdImageCreateFromPng (pngin);
    fclose (pngin);

    xSize = gdImageSX(im);
    ySize = gdImageSY(im);
    // Add border on left, right, up, and down
    xSize = xSize + 2;
    ySize = ySize + 2;
    arraySize = xSize * ySize;
    theArray = new unsigned char[arraySize];

    // Clone first row
    for (long J=1; J<(xSize-1); J++)
    {
        theArray[J] = processPixel(im, J-1, 0);
    }
    for (long I=1; I<(ySize-1); I++)
    {
        // Clone first column
        theArray[I*xSize] = processPixel(im, 0, I-1);
        for (long J=1; J<(xSize-1); J++)
        {
            theArray[I*xSize + J] = processPixel(im, J-1, I-1);
        }
        // Clone last column
        theArray[((I+1)*xSize-1)] = processPixel(im, xSize-3, I-1);
    }

    // Clone last row
    for (long J=1; J<(xSize-1); J++)
    {
        theArray[(ySize-1)*xSize + J] = processPixel(im, J-1, ySize-3);
    }

    // Create and blank out running tallies of velocities
    theArrayXVelocities = new int[ySize];
    for (long I=0; I<ySize; I++)
    {
        theArrayXVelocities[I] = 0;
    }
    gdImageDestroy(im);

}

/** This handles an individual pixel and determines what value to output for
    it */
int LatticeGas::processPixel(gdImagePtr im, int x, int y)
{
    int redColor, blueColor, greenColor;
    int lowVal, highCount;
    double particleAverage;
    redColor = gdImageRed(im, gdImageGetPixel(im, x, y));
    blueColor = gdImageBlue(im, gdImageGetPixel(im, x, y));
    greenColor = gdImageGreen(im, gdImageGetPixel(im, x, y));

    if (blueColor != redColor)  // If not a gray pixel, process it
        differently
    {
        if (redColor == 0) // Blue pixels set average particle densities
        {
            particleAverage = ((double)blueColor)*6.0/255.0;
            lowVal = (int)particleAverage;
            // Set up fraction of high density sites needed, out of a
                thousand
            highCount = (int)((particleAverage-(double)lowVal)*1000.0);
            // Take a random number between 0 and 1000, compare to fraction
                of high-density sites
            if (boundedRandom(1000) < highCount) // High-density site
            {
                switch (lowVal+1)
                {
                    case 0: return 0;
                    case 1: return oneParticle[boundedRandom(6)];
                    case 2: return twoParticle[boundedRandom(15)];
                    case 3: return threeParticle[boundedRandom(20)];
                    case 4: return fourParticle[boundedRandom(15)];
                    case 5: return fiveParticle[boundedRandom(6)];
                    case 6: return 63;
                }

            }
            else // Low-density site
            {
                switch (lowVal)
                {
                    case 0: return 0;
                    case 1: return oneParticle[boundedRandom(6)];
                    case 2: return twoParticle[boundedRandom(15)];
                    case 3: return threeParticle[boundedRandom(20)];
                    case 4: return fourParticle[boundedRandom(15)];
                    case 5: return fiveParticle[boundedRandom(6)];
                    case 6: return 63;
                }
            }
        }
        else
        {
        }
    }
    else if (redColor != greenColor) // Do walls with a set average particle
       density in them
    {
        particleAverage = ((double)greenColor)*6.0/255.0;
        lowVal = (int)particleAverage;
        // Set up fraction of high density sites needed, out of a thousand
        highCount = (int)((particleAverage-(double)lowVal)*1000.0);
        // Take a random number between 0 and 1000, compare to fraction of
            high-density sites
        if (boundedRandom(1000) < highCount) // High-density site
        {
            switch (lowVal+1)
            {
                case 0: return (0+64);
                case 1: return (oneParticle[boundedRandom(6)]+64);
                case 2: return (twoParticle[boundedRandom(15)]+64);
                case 3: return (threeParticle[boundedRandom(20)]+64);
                case 4: return (fourParticle[boundedRandom(15)]+64);
                case 5: return (fiveParticle[boundedRandom(6)]+64);
                case 6: return (63+64);
            }

        }
        else // Low-density site
        {
            switch (lowVal)
            {
                case 0: return (0+64);
                case 1: return (oneParticle[boundedRandom(6)]+64);
                case 2: return (twoParticle[boundedRandom(15)]+64);
                case 3: return (threeParticle[boundedRandom(20)]+64);
                case 4: return (fourParticle[boundedRandom(15)]+64);
                case 5: return (fiveParticle[boundedRandom(6)]+64);
                case 6: return (63+64);
            }
        }

    }
    else // Take value of pixel and use it as literal value for grid
    {
        return redColor;
    }
    return 0;
}

/** This writes the grid to a bitmap, in one of several styles */
void LatticeGas::writeBitmap(string fileName, int mode)
{
```

```
gdImagePtr im;
FILE *pngout;

switch (mode)
{
    case 0:  // Draw each grid element as a pixel, greyshade signifying
        number value
    {
        cout << "xSize:␣" << xSize << "\n";
        cout << "ySize:␣" << ySize << "\n";
        im = gdImageCreateTrueColor((xSize-2), (ySize-2));
        long counter2=xSize+1;
        for (int K=0; K < (ySize-2); K++)
        {
            for (int M=0; M < (xSize-2); M++)
            {
                gdImageSetPixel(im, M, K, gdImageColorAllocate(im,
                    theArray[counter2], theArray[counter2], theArray[
                    counter2]));
                counter2++;
            }
            counter2 += 2;
        }


    }; break;
    case 1:  // Draw the average density over a certain block size as a
        color value
    {
        long counter2=0;
        long * averages = averageDensities(densityCellSize);
        im = gdImageCreateTrueColor((xSize/densityCellSize)*
            densityCellSize, (ySize/densityCellSize)*densityCellSize);
        for (int K=0; K < averages[1]; K++)
        {
            for (int M=0; M < averages[0]; M++)
            {
                gdImageFilledRectangle(im, M*densityCellSize, K*
                    densityCellSize, (M+1)*densityCellSize, (K+1)*
                    densityCellSize, gdImageColorAllocate(im, averages[
                    counter2+2], averages[counter2+2], averages[counter2
                    +2]) );
                counter2++;
            }
        }
    }; break;
    case 2:  // Draw the average velocities of a block as a line
    {
        long counter2=0;
        avgVelocity * averages = averageVelocities(velocityCellSize);
        im = gdImageCreateTrueColor((xSize/velocityCellSize)*
            velocityCellSize, (ySize/velocityCellSize)*velocityCellSize);
        gdImageFill(im, 1, 1, gdImageColorAllocate(im, 255, 255, 255) );
        long black=gdImageColorAllocate(im, 0, 0, 0);
        for (int K=0; K < averages[0].yPos; K++)
        {
            for (int M=0; M < averages[0].xPos; M++)
            {
                gdImageLine(im, M*velocityCellSize, K*velocityCellSize, M
                    *velocityCellSize + averages[counter2].xPos, K*
                    velocityCellSize - averages[counter2].yPos, black);
                counter2++;
            }
        }

    }; break;
    case 3:  // Draw the average velocities of a block as an arrow
    {
        long counter2=0;
        long xHolder, yHolder;
        avgVelocity * averages = averageVelocities(velocityCellSize);

        im = gdImageCreateTrueColor((xSize/velocityCellSize)*
            velocityCellSize, (ySize/velocityCellSize)*velocityCellSize);
        gdImageFill(im, 1, 1, gdImageColorAllocate(im, 255, 255, 255) );
        long black=gdImageColorAllocate(im, 0, 0, 0);
        for (int K=0; K < averages[0].yPos; K++)
        {
            for (int M=0; M < averages[0].xPos; M++)
            {
                gdImageLine(im, M*velocityCellSize, K*velocityCellSize, M
                    *velocityCellSize + averages[counter2].xPos, K*
                    velocityCellSize - averages[counter2].yPos, black);
                xHolder = (4*averages[counter2].xPos)/5 - averages[
                    counter2].yPos/5;
                yHolder = (4*averages[counter2].yPos)/5 + averages[
                    counter2].xPos/5;
                gdImageLine(im, M*velocityCellSize + averages[counter2].
                    xPos, K*velocityCellSize - averages[counter2].yPos, M
                    *velocityCellSize + xHolder, K*velocityCellSize -
                    yHolder, black);
                counter2++;
            }
        }

    }; break;
    case 4:  // Do a velocity profile along a line using average
        velocities
    {
        // Use scaling to smooth out the line profile
        int scaling = 64;

        lineProfileCellSize *= scaling;
        long * averages = lineVelocities(lineProfileCellSize);
        im = gdImageCreateTrueColor((lineProfileCellSize*4)/scaling,
            ySize ); // Max velocity range along x is -2*cells..2*cells
        gdImageFill(im, 1, 1, gdImageColorAllocate(im, 255, 255, 255) );
        long black=gdImageColorAllocate(im, 0, 0, 0);
```

```
            for (int K=0; K < ySize; K++)
            {
                gdImageLine(im,(lineProfileCellSize*2)/scaling,K,(
                    lineProfileCellSize*2+averages[K])/scaling,K,black);
            }
        }; break;
        case 5:
        {
        }; break;
        case 6:
        {
        }; break;
    };

    pngout = fopen(fileName.c_str(), "wb");
    gdImagePng(im, pngout);
    fclose(pngout);
    gdImageDestroy(im);
}

/** This writes the grid to a scalable vector graphic (SVG) file, in one of
    several styles */
void LatticeGas::writeSVG(string fileName, int mode)
{
    ofstream dataFile(fileName.data());
    dataFile << "<?xml␣version=\"1.0\"␣encoding=\"UTF-8\"␣standalone=\"no
        \"?>\n";
    dataFile << "<!DOCTYPE␣svg␣PUBLIC␣\"-//W3C//DTD␣SVG␣1.0//EN\"\n";
    dataFile << "\"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd
        \">\n";
    dataFile << "<!--␣Created␣with␣LatticeGas␣-->\n";
    dataFile << "<svg\n";
    dataFile << "\txmlns=\"http://www.w3.org/2000/svg\"\n";
    dataFile << "\txmlns:xlink=\"http://www.w3.org/1999/xlink\"\n";
    dataFile << "\tx=\"0.0000000\"\n";
    dataFile << "\ty=\"0.0000000\"\n";


    switch (mode)
    {
        case 0:  // Draw each grid element as a pixel, greyshade signifying
            number value
        {
        }; break;
        case 1:  // Draw the average density over a certain block size as a
            color value
        {
        }; break;
        case 2:  // Draw the average velocities of a block as a line
        {
            dataFile << "\twidth=\"" << (xSize/velocityCellSize)*
                velocityCellSize << "\"\n";
            dataFile << "\theight=\"" << (ySize/velocityCellSize)*
                velocityCellSize << "\">\n";
            long counter2=0;
            avgVelocity * averages = averageVelocities(velocityCellSize);
            for (int K=0; K < averages[0].yPos; K++)
            {
                for (int M=0; M < averages[0].xPos; M++)
                {
                    dataFile << "\t<line␣x1=\"" << M*velocityCellSize << "\"␣
                        y1=\"" << K*velocityCellSize << "\"␣x2=\"" << M*
                        velocityCellSize + averages[counter2].xPos << "\"␣y2
                        =\"" << K*velocityCellSize - averages[counter2].yPos
                         << "\"␣stroke=\"#000000\"␣stroke-width=\"0.100\"␣/>\
                        n";
                    counter2++;
                }
            }

        }; break;
        case 3:  // Draw the average velocities of a block as an arrow
        {
        }; break;
        case 4:  // Do a velocity profile along a line using average
            velocities
        {
        }; break;
        case 5:
        {
        }; break;
        case 6:
        {
        }; break;
    };
    dataFile << "</svg>\n";
    dataFile.close();
}


/** This writes the grid to a comma-separated-value file, in one of several
    styles */
void LatticeGas::dumpRawData(string fileName, int mode)
{
    ofstream dataFile(fileName.data());
    switch (mode)
    {
        case 0: // Comma-separated-value list of fluid velocities along a
            line
        {
            int blockSize = 100;
            int scaling = 64;
            blockSize *= scaling;
            long * averages = lineVelocities(blockSize);
            for (int K=0; K < ySize; K++)
            {
                dataFile << K << "," << (averages[K]/scaling) << "\n";
            }
        }
        case 1: // Comma-separated-value list of time-averaged fluid
            velocities along a line
        {
            for (int K=0; K < ySize; K++)
            {
                dataFile << K << "," << theArrayXVelocities[K] << "\n";
            }
        }
    }

    dataFile.close();
}

/** This returns an array of x,y pairs corresponding to the average velocity
    of a block of the width and height specified */
avgVelocity * LatticeGas::averageVelocities(int blockSize)
{
    avgVelocity * averages = new avgVelocity[(xSize/blockSize) * (ySize/
        blockSize) + 1];
    avgVelocity holder;
    holder.xPos = xSize/blockSize;
    holder.yPos = ySize/blockSize;
    averages[0] = holder;

    // Split the grid up into blocks
    for (int I=0; I<(ySize/blockSize); I++)
    {
        for (int J=0; J<(xSize/blockSize); J++)
        {
            // Store tallies for number of particles with each velocity
                vector
            long left=0, right=0, upperright=0, upperleft=0, lowerright=0,
                lowerleft=0;
            // Within blocks, tally up velocity vectors of all elements
            for (int M=(I*blockSize); M<((I+1)*blockSize); M++)
            {
                for (int N=(J*blockSize); N<((J+1)*blockSize); N++)
                {
                    // Using bit-shift operators, pick out single velocity
                        bits and add them to a running tally
                    unsigned char holder2;
                    holder2 = theArray[M*xSize+N] << 2;
                    upperright += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 3;
                    upperleft += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 4;
                    left += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 5;
                    lowerleft += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 6;
                    lowerright += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 7;
                    right += holder2 >> 7;
                }
            }
            // Translate velocities to the (x,y) coordinate system
            holder.xPos = (right + (5 * (upperright + lowerright))/10) - (
                left + (5 * (upperleft + lowerleft))/10);
            holder.yPos = ((866 * (upperright + upperleft))/1000) - ((866 * (
                lowerleft + lowerright))/1000);

            holder.xPos = (holder.xPos / blockSize)*2;
            holder.yPos = (holder.yPos / blockSize)*2;
            averages[(I*(xSize/blockSize) + J)+1] = holder;
        }
    }
    return averages;
}

/** This returns an array of average particle densities */
long * LatticeGas::averageDensities(int blockSize)
{
    long * averages = new long[(xSize/blockSize) * (ySize/blockSize) + 2];
    long maximumDensity;
    // Figure the maximum density, use this to normalize the averages to 255
    maximumDensity = blockSize * blockSize * 6;
    averages[0] = xSize/blockSize;
    averages[1] = ySize/blockSize;

    // Split the grid up into blocks
    for (int I=0; I<(ySize/blockSize); I++)
    {
        for (int J=0; J<(xSize/blockSize); J++)
        {
            // Store tallies for number of particles
            long particlesInBlock=0;
            // Within blocks, tally particles in all elements
            for (int M=(I*blockSize); M<((I+1)*blockSize); M++)
            {
                for (int N=(J*blockSize); N<((J+1)*blockSize); N++)
                {
                    // Using bit-shift operators, pick out single velocity
                        bits and add them to a running tally
                    unsigned char holder2;
                    holder2 = theArray[M*xSize+N] << 2;
                    particlesInBlock += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 3;
                    particlesInBlock += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 4;
                    particlesInBlock += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 5;
                    particlesInBlock += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 6;
                    particlesInBlock += holder2 >> 7;
                    holder2 = theArray[M*xSize+N] << 7;
                    particlesInBlock += holder2 >> 7;
                }
            }
            averages[(I*(xSize/blockSize) + J)+2] = 255 - ((particlesInBlock
                 * 255) / maximumDensity);
        }
    }
    return averages;
}

/** This gets a velocity profile along a line */
long * LatticeGas::lineVelocities(int blockSize)
{
    long xHolder;
    long xLinePosition;
    // Take half of the block to the left of the line, half to the right
    xLinePosition=lineProfileLocation-blockSize/2;

    long * averages = new long[ySize * blockSize];
    // Store tallies for number of particles with each velocity vector

    for (int M=0; M<ySize; M++)
    {
        // Within blocks, tally up velocity vectors of all elements
        long left=0, right=0, upperright=0, upperleft=0, lowerright=0,
            lowerleft=0;
        for (int N=xLinePosition; N<(xLinePosition+blockSize); N++)
        {
            // Using bit-shift operators, pick out single velocity bits and
                add them to a running tally
            unsigned char holder2;
            holder2 = theArray[M*xSize+N] << 2;
            upperright += holder2 >> 7;
            holder2 = theArray[M*xSize+N] << 3;
            upperleft += holder2 >> 7;
            holder2 = theArray[M*xSize+N] << 4;
            left += holder2 >> 7;
            holder2 = theArray[M*xSize+N] << 5;
            lowerleft += holder2 >> 7;
            holder2 = theArray[M*xSize+N] << 6;
            lowerright += holder2 >> 7;
            holder2 = theArray[M*xSize+N] << 7;
            right += holder2 >> 7;
        }

        // Pull out the x component of the velocities
        xHolder = (right + (5 * (upperright + lowerright))/10) - (left
             + (5 * (upperleft + lowerleft))/10);
        averages[M] = xHolder;
    }

    return averages;

}

int LatticeGas::boundedRandom(int range)
```

```
{
  double  rawResult;
  rawResult = (double) rand() / RAND_MAX;
  return (int) (rawResult * range);
}

/** This processes an XML file and fills out the internal array */
void LatticeGas::loadXML(string fileName)
{
/*  XML tags:
<fluidmodeler>:
    <bitmap>[filename]</bitmap>: Loads a simulation grid from a bitmap file
    <timesteps>[number]</timesteps>: Specifies the number of time steps of
        the simulation to run
    <currentstep>[number]</currentstep>: If resuming a simulation from a
        previous run, lets you set what the current timestep is
    <vibration>: Enables the simulation of vibrations
        <period>[number]</period>: Specify the period of vibrations, in time
            units
        <amplitude>[number]</amplitude>: Specify the amplitude of vibrations
            , out of a maximum of 10000
        <solidplate>[1 or 0]</solidplate>: Either plate vibrates as a solid
            unit or vibrations travel through the plate
        <vibrationtype>[0, 1]</vibrationtype>: 0 - longitudinal vibrations
            , 1 - transverse
        <startvibration>[number]</startvibration>: If a traveling and/or
            tapered vibration, where vibration starts from
        <stopvibration>[number]</stopvibration>: If a traveling and/or
            tapered vibration, where vibration stops
        <tapered>[1 or 0]</tapered>: Either the vibration amplitude changes
            with x-position or not
    </vibration>

    <output>: This section controls the output of data from the run
        <density>: Output a density map
            <cellsize>[number]<cellsize>: Get density by averaging across
                cells that are [number] units on a side
            <interval>[number]</interval>: The interval, in time steps, at
                which to output each new image
        </density>
        <velocity>: Output a velocity map
            <cellsize>[number]<cellsize>: Get velocity by averaging across
                cells that are [number] units on a side
            <interval>[number]</interval>: The interval, in time steps, at
                which to output each new image
        </velocity>
        <rawbitmap>: Dump the current state of the grid as a bitmap
            <interval>[number]</interval>: The interval, in time steps, at
                which to output each new image
        </rawbitmap>
        <lineprofile>
            <cellsize>[number]</cellsize>: Width of area to average
                velocities across
            <interval>[number]</interval>: The interval, in time steps, at
                which to output each new image
            <location>[number]</location>: X value where the line is located
            <autoscale>[1 or 0]</autoscale>: Auto-scales
            <dumprawdata>[1 or 0]</dumprawdata>: Dumps raw values in a CSV
                file
        </lineprofile>
    </output>
</fluidmodeler>
*/

    ifstream xmlFile(fileName.data());
    string tagHolder;
    string betweenTags;
    bool foundStart, notDoneProcessing, processingUnit;
    foundStart = false;

    int cellSize = 0;
    int interval = 0;

    vibrationRadians = 0;
    vibrationAmplitude = 0;
    vibrationPeriod = 0;
    amplitudeMax = 10000;
    currentStep = 0;

    taperedVibration = false;

    isVibrating = false;
    outputDensity = false;
    outputVelocity = false;
    outputRawBitmap = false;
    outputLineProfile = false;
    lineProfileAutoscale = false;
    lineProfileDumpRawData = false;

    // Find the start tag
    while (!foundStart)
    {
        if (xmlFile.eof())
        {
            cout << "Error:␣this␣is␣not␣a␣supported␣XML␣pattern␣file.\n";
            foundStart = true;
        }
        else
        {
            getline(xmlFile, betweenTags, '<'); // Junk before <
            getline(xmlFile, tagHolder, '>'); // Actual tag
            if (tagHolder == "fluidmodeler")
            {
                 foundStart = true;
            }
         }
    }
```

```
notDoneProcessing = true;
processingUnit = false;
// Process all tags
while (notDoneProcessing)
{
    if (xmlFile.eof()) // Something’s wrong, there’s no </fluidmodeler>
        end tag
    {
         cout << "Warning:␣no␣terminating␣</fluidmodeler>␣tag.\n";
         notDoneProcessing = false;
    }
    else // Not end of file, keep processing
    {
        getline(xmlFile, betweenTags, ’<’); // Junk before <
        getline(xmlFile, tagHolder, ’>’); // Actual tag
        if (tagHolder == "/fluidmodeler")
        {
             notDoneProcessing = false;
        }
        else if (tagHolder == "/bitmap")
        {
            bitmapName = betweenTags;
        }
        else if (tagHolder == "/timesteps")
        {
             timeSteps = atol(betweenTags.data());
        }
        else if (tagHolder == "/currentstep")
        {
             currentStep = atol(betweenTags.data());
        }
        else if (tagHolder == "/vibration")
        {
            isVibrating = true;
        }
        else if (tagHolder == "/solidplate")
        {
            isSolidPlate = atoi(betweenTags.data());
        }
        else if (tagHolder == "/period")
        {
             vibrationPeriod = atol(betweenTags.data());
        }
        else if (tagHolder == "/amplitude")
        {
             vibrationAmplitude = atol(betweenTags.data());
        }
        else if (tagHolder == "density")
        {
             processingUnit = true;
        }
        else if (tagHolder == "/density")
        {
```

```
     processingUnit = false;
     densityCellSize = cellSize;
     densityInterval = interval;
     outputDensity = true;
}
else if (tagHolder == "velocity")
{
     processingUnit = true;
}
else if (tagHolder == "/velocity")
{
     processingUnit = false;
     velocityCellSize = cellSize;
     velocityInterval = interval;
     outputVelocity = true;
}
else if (tagHolder == "rawbitmap")
{
     processingUnit = true;
}
else if (tagHolder == "/rawbitmap")
{
     processingUnit = false;
     rawBitmapInterval = interval;
     outputRawBitmap = true;
}
else if (tagHolder == "lineprofile")
{
     processingUnit = true;
}
else if (tagHolder == "/lineprofile")
{
     processingUnit = false;
     lineProfileCellSize = cellSize;
     lineProfileInterval = interval;
     outputLineProfile = true;
}
else if (tagHolder == "/location")
{
     lineProfileLocation = atol(betweenTags.data());
}
else if (tagHolder == "/autoscale")
{
    lineProfileAutoscale = atoi(betweenTags.data());
}
else if (tagHolder == "/dumprawdata")
{
    lineProfileDumpRawData = atoi(betweenTags.data());
}

else if (tagHolder == "/cellsize")
{
     cellSize = atoi(betweenTags.data());
}
```

```
            else if (tagHolder == "/interval")
            {
                 interval = atol(betweenTags.data());
            }
            else if (tagHolder == "/vibrationtype")
            {
                 vibrationType = atoi(betweenTags.data());
            }
            else if (tagHolder == "/startvibration")
            {
                 startVibration = atoi(betweenTags.data());
            }
            else if (tagHolder == "/stopvibration")
            {
                 stopVibration = atoi(betweenTags.data());
            }
            else if (tagHolder == "/tapered")
            {
                taperedVibration = atoi(betweenTags.data());
            }
        }
    }

    xmlFile.close();

    // Start a simulation run based on parameters from the XML file
    loadBitmap(bitmapName);

    processingLoop(timeSteps);

}
```

A.6.4 main.cpp

```
/***************************************************************************
                          main.cpp  -  description
                             -------------------
    begin                : Wed Oct 22 20:57:42 CDT 2003
    author               : Brad Larson
    email                : blarson@cae.wisc.edu
 ***************************************************************************/

#include <iostream>
using namespace std;

#include <stdlib.h>

#include "latticegas.h"

int main(int argc, char *argv[])
{
    if (argc == 2) // fluidmodeler [filename]: Inputs specification for fluid
         modeling simulation from an XML file
    {
            LatticeGas *theLattice = new LatticeGas();
            theLattice->loadXML(argv[1]);
            delete theLattice;
    }
    else
    {
        cout<<"Proper␣usage:\n";
        cout<<"./fluidmodeler␣[filename]\n";
        cout<<"[filename]:␣An␣XML␣file␣that␣contains␣the␣specifications␣for␣
            the␣simulation␣run\n";

    }

    return EXIT_SUCCESS;
}
```

Bibliography

[1] C. K. Mathews, K. E. vanHolde, and K. G. Ahern. *Biochemistry*. Addison Wesley Longman, San Francisco, CA. (2000).

[2] D. J. Lockhart, H. L. Dong, M. C. Bryne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. W. Wang, M. Kobayashi, H. Horton, and E. L. Brown. Expression monitoring by hybridization to high density oligonucleotide arrays. *Nature Biotechnology* **14**, 1675–1680 (1996).

[3] D. J. Duggan, M. Bittner, Y. Chen, P. Meltzer, and J. M. Trent. Expression profiling using cDNA microarrays. *Nature Genetics Supplement* **21**, 10–14 (1999).

[4] T. R. Hughes, M. Mao, A. R. Jones, J. Burchard, M. J. Marton, K. W. Shannon, S. M. Lefkowitz, M. Ziman, J. M. Schelter, M. R. Meyer, S. Kobayashi, C. Davis, H. Dai, Y. D. He, S. B. Stephaniants, G. Cavet, W. L. Walker, A. West, E. Coffey, D. Shoemaker, R. Stoughton, A. P. Blanchard, S. H. Friend, and P. S. Linsley. Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer. *Nature Biotechnology* **19**, 342–347 (2001).

[5] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. G. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. M. Yang, G. E. Marti, T. Moore, J. Hudson, L. S. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt.

Distinct types of diffuse large b-cell lymphoma identified by gene expression analysis. *Nature* **403**, 503–511 (2000).

[6] C. M. Henry. Dna microarrays in toxicology. *Analytical Chemistry* **71**, 462A–464A (1999).

[7] S. S. Iqbal, M. W. Mayo, J. G. Bruno, B. V. Bronk, C. A. Batt, and J. P. Chambers. A review of molecular recognition technologies for detection of biological threat agents. *Biosensors and Bioelectronics* **15**, 549–578 (2000).

[8] L. M. Smith, R. M. Corn, A. E. Condon, M. G. Lagally, A. G. Frutos, Q. Liu, and A. J. Thiel. A surface-based approach to DNA computation. *Journal of Computational Biology* **5**, 255 (1998).

[9] A. G. Frutos, Q. Liu, A. J. Thiel, A. M. W. Sanner, A. E. Condon, L. M. Smith, and R. M. Corn. Demonstration of a word design strategy for DNA computing on surfaces. *Nucleic Acids Research* **25**, 4747–4757 (1997).

[10] S. P. A. Fodor, J. L. Read, M. C. Pirrung, L. Stryer, L. A. Tsai, and D. Solas. Light-directed, spatially addressable parallel chemical synthesis. *Science* **251**, 767–773 (1991).

[11] M. Chee, R. Yang, E. Hubbell, A. Berno, X. C. Huang, D. Stern, J. Winkler, D. J. Lockhart, M. S. Morris, and S. P. A. Fodor. Accessing genetic information with high-density DNA arrays. *Science* **274**, 610 (1996).

[12] S. Singh-Gasson, R. D. Green, Y. Yue, C. Nelson, F. Blattner, M. R. Sussman, and

F. Cerrina. Maskless fabrication of light-directed oligonucleotide microarrays using a digital micromirror array. *Nature Biotechnology* **17**, 974–978 (1999).

[13] Affymetrix - CustomExpress Array Program. http://www.affymetrix.com/products/arrays/specific/cexpress.affx.

[14] K. N. Lee, D. S. Shin, Y. S. Lee, and Y. K. Kim. Protein patterning by virtual mask photolithography using a micromirror array. *Journal of Micromechanics and Microengineering* **13**, 18–25 (2003).

[15] S. P. Gygi, Y. Rochon, B. R. Franza, and R. Aebersold. Correlation between protein and mRNA abundance in yeast. *Molecular and Cellular Biology* **19**, 1720–1730 (1999).

[16] G. Chen, T. G. Gharib, C.-C. Huang, J. M. G. Taylor, D. E. Misek, S. L. R. Kardia, T. J. Giordano, M. D. Iannettoni, M. B. Orringer, S. M. Hanash, and D. G. Beer. Discordant protein and mRNA expression in lung adenocarcinomas. *Molecular and Cellular Proteomics* **1**, 304–313 (2002).

[17] A. Sreekumar, M. K. Nyati, S. Varambally, T. R. Barrette, D. Ghosh, T. S. Lawrence, and A. M. Chinnaiyan. Profiling of cancer cells using protein microarrays: discovery of novel radiation-regulated proteins. *Cancer Research* **61**, 7585–7593 (2001).

[18] H. Zhu, M. Bilgin, R. Bangharm, D. Hall, A. Casamayor, P. Bertone, N. Lan, R. Jansen, S. Bidlingmaier, T. Houfek, T. Mitchell, P. Miller, R. A. Dean, M. Gerstein, and M. Snyder. Global analysis of protein activities using proteome chips. *Science* **293**, 2101–2106 (2001).

[19] G. MacBeath and S. L. Schreiber. Printing proteins as microarrays for high-throughput function determination. *Science* **289**, 1760–1764 (2000).

[20] M. Schena. *Protein Microarrays*. Jones and Bartlett, Sudbury, MA. (2004).

[21] A. Watson, A. Mazumder, M. Stewart, and S. Balasubramanian. Technology for microarray analysis of gene expression. *Current Opinion in Biotechnology* **9**, 609–614 (1998).

[22] Y. Balagurunathan, N. Wang, E. R. Dougherty, D. Nguyen, Y. Chen, M. L. Bittner, J. Trent, and R. Carroll. Noise factor analysis for cDNA microarrays. *Journal of Biomedical Optics* **9**, 663–678 (2004).

[23] M. Schena, D. Shalon, R. Davis, and P. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* **270**, 467–470 (1995).

[24] V. G. Cheung, M. Morley, F. Aguilar, A. Massimi, R. Kucherlapati, and G. Childs. Making and reading microarrays. *Nature Genetics Supplement* **21**, 15–19 (1999).

[25] Build your own arrayer. http://cmgm.stanford.edu/pbrown/mguide/.

[26] A. P. Blanchard, R. J. Kaiser, and L. E. Hood. High-density oligonucleotide arrays. *Biosensors and Bioelectronics* **11**, 687–690 (1996).

[27] R. Piner, J. Zhu, F. Xu, S. Hong, and C. A. Mirkin. Dip-pen nanolithography. *Science* **283**, 661–663 (1999).

[28] K. B. Lee, S. J. Park, C. A. Mirkin, J. C. Smith, and M. Mrksich. Protein nanoarray generated by dip-pen nanolithography. *Science* **295**, 1702–1706 (2002).

[29] D. Filmore. Crystallography on drugs. *Today's Chemist at Work* **13**, 32–36 (2004).

[30] A. Park, B. Wu, and L. Griffith. Integration of surface modification and 3D fabrication techniques to prepare patterned poly(L-lactide) substrates allowing regionally selective cell adhesion. *J. Biomater. Sci., Polymer Ed.* **9**, 89–118 (1998).

[31] H. Sirringhaus, T. Kawase, R. H. Friend, T. Shimoda, M. Inbasekaran, W. Wu, and E. P. Woo. High-resolution inkjet printing of all-polymer transistor circuits. *Science* **290**, 2123–2126 (2000).

[32] T. R. Hebner, C. C. Wu, D. Marcy, M. H. Lu, and J. C. Sturm. Ink-jet printing of doped polymers for organic light emitting devices. *Applied Physics Letters* **72**, 518–521 (1998).

[33] S. R. Forrest. The path to ubiquitous and low-cost organic electronic appliances on plastic. *Nature* **428**, 911–918 (2004).

[34] F. Diehl, S. Grahlmann, M. Beier, and J. D. Hoheisel. Manufacturing DNA microarrays of high spot homogeneity and reduced background signal. *Nucleic Acids Research* **29**, e38 (2001).

[35] UWBC: Slide Coating (polylysine treatment). http://www.biotech.wisc.edu/GEC/protocols/GEC_Slide_Coating.asp.

[36] S. D. Gillmor, A. J. Thiel, T. C. Strother, L. M. Smith, and M. G. Lagally. Hydrophilic/hydrophobic patterned surfaces and templates for DNA arrays. *Langmuir* **16**, 7223–7228 (2000).

[37] T. C. Strother, W. Cai, X. S. Zhao, R. J. Hamars, and L. M. Smith. Characterization of DNA-modified silicon (111) surfaces. *Journal of the Americal Chemical Society* **122**, 1205–1209 (2000).

[38] W. Yang, O. Auciello, J. E. Butler, W. Cai, J. A. Carlisle, J. E. Gerbi, D. M. Gruen, T. Knickerbocker, T. L. Lasseter, J. J N Russell, L. M. Smith, and R. J. Hamers. DNA-modified nanocrystalline diamond thin-films as stable, biologically active substrates. *Nature Materials* **1**, 253–257 (2002).

[39] G. T. Hermanson, A. K. Mallia, and P. K. Smith. *Immobilized Affinity Ligand Techniques*. Academic Press, San Diego, CA. (1992).

[40] M. Yan, S. X. Cai, M. N. Wybourne, and J. F. Keana. N-hydroxysuccinimide ester functionalized perfluorophenyl azides as novel photoactive heterobifunctional cross-linking reagents. The covalent immobilization of biomolecules to polymer surfaces. *Bioconjugate Chemistry* **5**, 151–157 (1994).

[41] A Härtl and E Schmich and J A Garrido and J Hernando and S C R Catharino and S Walter and P Feulner and A Kromka and D Steinmüller and M Stutzmann. Protein-modified nanocrystalline diamond thin films for biosensor applications. *Nature Materials* **3**, 736–742 (2004).

[42] S. Taylor, S. Smith, B. Windle, and A. Guiseppi-Elie. Impact of surface chemistry and blocking strategies on DNA microarrays. *Nucleic Acids Research* **31**, e87 (2003).

[43] C. Preininger and U. Sauer. Quality control of chip manufacture and chip analysis using epoxy-chips as a model. *Sensors and Actuators B* **90**, 98–103 (2003).

[44] A. Grill. *Cold Plasma in Materials Fabrication*. IEEE Press, New York, NY. (1993).

[45] J. L. Vossen and W. Kern. *Thin Film Processes*. Academic Press, San Diego, CA. (1978).

[46] J. M. Helgren. Applications of cold plasma chemistry to surface functionalization for biomolecule attachment and disinfection of teat cup liners. Master's thesis, University of Wisconsin-Madison, August (2004).

[47] M. J. Owen and P. J. Smith. Plasma Treatment of Polydimethylsiloxane. *J. Adhesion Sci Technology.* **8**, 1063–1075 (1994).

[48] S. D. Gillmor. *Studies of patterned surfaces for biological microarrays*. PhD thesis, University of Wisconsin-Madison, July (2002).

[49] S. D. Gillmor, B. J. Larson, J. M. Braun, C. E. Mason, L. E. Cruz-Barba, F. Denes, and M. G. Lagally. Low-contact-angle polydimethyl siloxane (pdms) membranes for fabricating micro-bioarrays. In *Second Annual International IEEE-EMBS Special Topic Conference on Microtechnology in Medicine and Biology,* A. Dittmar and D. Beebe, editors, 51–56. IEEE, IEEE, May (2002).

[50] S. D. Gillmor†, B. J. Larson†, J. M. Braun, L. E. Cruz-Barba, D. E. Savage, C. E. Mason, F. Denes, and M. G. Lagally. Long-term reduction in poly(dimethylsiloxane) surface hydrophobicity via cold-plasma treatments. , in preparation.

[51] M. Mason, K. P. Vercruysse, K. R. Kirker, R. Frisch, D. M. Marecak, G. D.

Prestwich, and W. G. Pitt. Attachment of hyaluronic acid to polypropylene, polystyrene, and polytetrafluoroethylene. *Biomaterials* **21**, 31–36 (2000).

[52] F. Denes, R. A. Young, and M. Sarmadi. Surface functionalization of polymers under cold plasma conditions - a mechanistic approach. *Journal of Photopolymer Science and Technology* **10**, 91–112 (1997).

[53] F. S. Denes and S. Manolache. Macromolecular plasma-chemistry: an emerging field of polymer science. *Progress in Polymer Science* **29**, 815–885 (2004).

[54] B. J. Larson, S. D. Gillmor, and M. G. Lagally. Controlled deposition of picoliter amounts of fluid using an ultrasonically driven micropipette. *Review of Scientific Instruments* **75**, 832–836 (2004).

[55] B. J. Larson, J. M. Helgren, S. O. Manolache, A. Y. Lau, M. G. Lagally, and F. S. Denes. Cold-plasma modification of oxide surfaces for covalent biomolecule attachment. *Biosensors and Bioelectronics*, in press.

[56] D. Wang, S. Liu, B. J. Trummer, C. Deng, and A. Wang. Carbohydrate microarrays for the recognition of cross-reactive molecular markers of microbes and host cells. *Nature Biotechnology* **20**, 275–281 (2002).

[57] B. J. Larson, C. H. Lee, A. Lal, and M. G. Lagally. Methods and apparata for precisely dispensing microvolumes of fluids. US Patent: 6,874,699. (2005).

[58] R. C. Gamble, T. P. Theriault, and J. Baldeschwieler. Jet droplet device. US Patent: 6,001,309. (1999).

[59] G. C. Sasaki. Fluid dispenser and dispensing methods. US Patent: 6,296,811. (2001).

[60] R. M. Moroney, R. M. White, and R. T. Howe. Ultrasonically induced microtransport. *Proceedings Transducers 1991* , 931–934 (1991).

[61] A. Lal. *Micromachined silicon ultrasonic longitudinal mode actuators: theory and applications to surgery, pumping, and atomization*. PhD thesis, University of California, Berkeley, December (1996).

[62] M. Kurosawa, T. Watanabe, A. Futami, and T. Higuchi. Surface acoustic wave atomizer. *Sensors and Actuators A* **50**, 69–74 (1995).

[63] T. Young. An Essay on the Cohesion of Fluids. *Philosophical Transactions of the Royal Society* **95**, 65–87 (1805).

[64] W. S. Rasband. ImageJ. National Institutes of Health, Bethesda, MD. (1997-2004).

[65] DOW Glycerine: Viscosity. http://www.dow.com/glycerine/resources/table18.htm.

[66] Y. O. Popov. *Singularities, universality, and scaling in evaporative deposition patterns*. PhD thesis, University of Chicago, December (2003).

[67] J. D. Seader and E. J. Henley. *Separation Process Principles*. John Wiley and Sons, New York, NY. (1998).

[68] NIST Webbook - Water. http://webbook.nist.gov/cgi/cbook.cgi?ID=C7732185 &Units=SI&Mask=4#Thermo-Phase.

[69] R. D. Deegan, O. Bakajin, T. F. Dupont, G. Huber, S. R. Nagel, and T. A. Witten. Capillary flow as the cause of ring stains from dried liquid drops. *Nature* **389**, 827–829 (1997).

[70] Glossary to Electrical Engineering Terms. http://ece.wpi.edu/ frontier/glossary.html.

[71] R. F. Probstein. *Physicochemical hydrodynamics: an introduction*. Butterworths, Boston, MA. (1989).

[72] H. L. Berger. *Ultrasonic Liquid Atomization*. Partridge Hill, Hyde Park, NY. (1998).

[73] R. J. Lang. Ultrasonic atomization of liquids. *Journal of the Acoustical Society of America* **34**, 6–8 (1962).

[74] C. H. Lee and A. Lal. Single microdroplet ejection using an ultrasonic longitudinal mode with a PZT / tapered glass capillary. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* **51**, 1514–1522 (2004).

[75] G. G. Stokes. On the effect of the internal friction of fluids on the motion of pendulums. *Transactions of the Cambridge Philosophical Society* **9**, 8 (1850).

[76] T. C. Papanastasiou, G. C. Georgiou, and A. N. Alexandrou. *Viscous Fluid Flow*. CRC Press, New York, NY. (2000).

[77] W. McCabe, J. C. Smith, and P. Harriott. *Unit Operations of Chemical Engineering*. McGraw-Hill, New York, NY. (1985).

[78] Y.-F. Maa and C. C. Hsu. Effect of high shear on proteins. *Biotechnology and Bioengineering* **51**, 458–465 (1996).

[79] M. J. Rieder, R. Carmona, J. E. Krieger, J. K A Pritchard, and A. S. Greene. Suppression of angiotensin-converting enzyme expression and activity by shear stress. *Circulation Research* **80**, 312–319 (1997).

[80] J. P. D. Hartog. *Mechanical Vibrations*. Dover, New York, NY. (1985).

[81] A. G. Webster. Acoustical impedance and the theory of horns and of the phonograph. *Proc. Natl. Acad. Sci.* **5**, 275–282 (1919).

[82] E. Eisner. Resonant vibrations of a cone. *Journal of the Acoustical Society of America* **36**, 309–312 (1964).

[83] D. Ensminger. Solid cone in longitudinal half-wave resonance . *Journal of the Acoustical Society of America* **32**, 194–196 (1959).

[84] Corning Pyrex Properties. http://65.108.128.37/pub/prodsvc/corning-pyrex-properties.php.

[85] C. H. Lee. *Ultrasonic actuators: strain measurements and applications to microfluidics*. PhD thesis, University of Wisconsin-Madison, August (2002).

[86] E. Eisner. Design of sonic amplitude transformers for high magnification. *Journal of the Acoustical Society of America* **35**, 1367–1377 (1963).

[87] W. L. Nyborg. *Physical Acoustics*, volume 2B, chapter Acoustic Streaming. Academic, New York, NY (1965).

[88] M. Y. Jaffrin and A. H. Shapiro. Peristaltic pumping. *Annual Review of Fluid Mechanics* **3**, 13–36 (1971).

[89] S. Naguleswaran. A direct solution for the transverse vibration of Euler-Bernoulli wedge and cone beams. *Journal of Sound and Vibration* **172**, 289–304 (1994).

[90] S. Abrate. Vibration of non-uniform rods and beams. *Journal of Sound and Vibration* **185**, 703–716 (1995).

[91] Y. Hashimoto, Y. Koike, and S. Ueha. Transporting objects without contact using flexural traveling waves. *Journal of Acoustical Society of America* **103**, 3230–3233 (1998).

[92] M. F. Templin, D. Stoll, J. M. Schwenk, O. Potz, S. Kramer, and T. O. Jobs. Protein microarrays: Promising tools for proteomic research. *Proteomics* **3**, 2155–2166 (2003).

[93] S. D. Conzone and C. G. Pantano. Glass slides to DNA microarrays. *Materials Today* **7**, 20–26 (2004).

[94] Z. Guo, R. A. Guilfoyle, A. J. Thiel, R. Wang, and L. M. Smith. Direct fluorescence analysis of genetic polymorphisms by hybridization with oligonucleotide arrays on glass supports. *Nucleic Acids Research* **22**, 5456–5465 (1994).

[95] K. Yamada, A. Ito, N. Iwamoto, T. Haraguchi, and T. Kajiyama. Plasma-grafting of polymers with pendant oligo(oxyethylene) chains onto the surface of a polyaniline electrode. *Polymer Journal* **32**, 222–227 (2000).

[96] C. M. Halliwell and A. E. G. Cass. A factorial analysis of silanization conditions for the immobilization of oligonucleotides on glass surfaces. *Analytical Chemistry* **73**, 2476–2483 (2001).

[97] G. N. Merrill. The gas-phase reactivity of epichlorohydrin with hydroxide. *Journal of Physical Organic Chemistry* **17**, 241–248 (2004).

[98] M. Chanda and S. K. Roy. *Plastics Technology Handbook.* Marcel Dekker, Inc., New York, NY. (1998).

[99] A. Frey. Unpublished data.

[100] S. Alvarez-Blanco, S. Manolache, and F. Denes. A novel plasma-enhanced way for functionalization of polymeric substrates. *Polymer Bulletin* **47**, 329–336 (2001).

[101] S. Iijima. Helical microtubules of graphitic carbon. *Nature* **354**, 56–58 (1991).

[102] J. Kong, N. R. Franklin, C. Zhou, M. G. Chapline, S. Peng, K. Cho, and H. Dai. Nanotube molecular wires as chemical sensors. *Science* **287**, 622–625 (2000).

[103] Y. Cui, Q. Wei, H. Park, and C. M. Lieber. Nanowire nanosensors for highly sensitive and selective detection of biological and chemical species. *Science* **293**, 1289–1292 (2001).

[104] S. E. Baker, W. Cai, T. L. Lasseter, K. P. Weidkamp, and R. J. Hamers. Covalently bonded adducts of deoxyribonucleic acid (DNA) oligonucleotides with single-wall carbon nanotubes: synthesis and hybridization. *Nano Letters* **2**, 1413–1417 (2002).

[105] M. S. Marcus, T. R. Narkis, J. M. Simmons, M. G. Lagally, and M. A. Eriksson. Carbon nanotube photodetector devices. In *Annual APS March Meeting*. American Physical Society, March (2003).

[106] T. R. Narkis, B. Barnes, M. S. Marcus, M. A. Eriksson, and M. G. Lagally. Photoconductivity of individual single-walled carbon nanotubes illuminated with synchrotron radiation. In *Annual APS March Meeting*. American Physical Society, March (2003).

[107] H. Dai. *Carbon Nanotubes: Synthesis, Structure, Properties and Application*, volume 80 of *Topics in Applied Physics*, chapter Nanotube Growth and Characterization. Springer-Verlag, New York, NY (2001).

[108] B. Yang, M. S. Marcus, D. G. Keppel, P. P. Zhang, Z. W. Li, B. J. Larson, D. E. Savage, J. M. Simmons, O. M. Castellini, M. A. Eriksson, and M. G. Lagally. Template-directed carbon nanotube network using self-organized Si nanocrystals. *Applied Physics Letters* **86**, 263107 (2005).

[109] K. Besteman, J. O. Lee, F. G. M. Wiertz, H. A. Heering, and C. Dekker. Enzyme-coated carbon nanotubes as single-molecule biosensors. *Nano Letters* **3**, 727–730 (2003).

[110] M. Freitag, Y. Martin, J. A. Misewich, R. Martel, and P. Avouris. Photoconductivity of single carbon nanotubes. *Nano Letters* **3**, 1067–1071 (2003).

[111] J. A. Misewich, R. Martel, P. Avouris, J. C. Tsang, S. Heinze, and J. Tersoff. Electrically induced optical emission from a carbon nanotube FET. *Science* **300**, 783–786 (2003).

[112] D. Barnes, M. Vestling, B. J. Larson, and M. G. Lagally. Deposition of samples and sample matrix for enhancing the sensitivity of matrix-assisted laser desorption / ionization mass spectrometry. USPTO Application # 11/025,349.

[113] M. Karas, D. Bachmann, U. Bahr, and F. Hillenkamp. Matrix-assisted ultraviolet-laser desorption of nonvolatile compounds. *International Journal of Mass Spectrometry and Ion Processes* **78**, 53–68 (1987).

[114] K. Tanaka, H. Waki, Y. Ido, S. Akita, Y. Yoshida, and T. Yoshida. Protein and polymer analyses up to m/z 100,000 by laser ionization time-of-flight mass spectrometry. *Rapid Communications in Mass Spectrometry* **2**, 151–153 (1988).

[115] M. A. R. Meier, B. deGans, A. M. J. van denBerg, and U. S. Schubert. Automated multiple-layer spotting for matrix-assisted laser desorption/ionization time-of-flight mass spectrometry of synthetic polymers utilizing ink-jet printing technology. *Rapid Communications in Mass Spectrometry* **17**, 2349–2352 (2003).

[116] T. Milliotis, S. Kjellstrom, J. Nilsson, T. Laurell, L. E. Edholm, and G. Marko-Varga. Capillary liquid chromatography interfaced to matrix-assisted laser desorption/ionization time-of-flight mass spectrometry using an on-line coupled piezoelectric flow-through microdispenser. *Journal of Mass Spectrometry* **35**, 369–377 (2000).

[117] H. Cheun, B. Tanto, W. Chunwaschirasiri, B. Larson, and M. J. Winokur. Near-term aging and thermal behavior of polyfluorene in various aggregation states. *Applied Physics Letters* **84**, 22–24 (2004).

[118] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the Navier-Stokes equation. *Physical Review Letters* **56**, 1505–1508 (1986).

[119] S. Wolfram. Cellular automaton fluids 1: basic theory. *Journal of Statistical Physics* **45**, 471–526 (1986).

[120] G. R. McNamara and G. Zanetti. Use of the Boltzmann equation to simulate lattice-gas automata. *Physical Review Letters* **61**, 2332–2335 (1988).