University of Wisconsin-Madison

Department of Biomedical Engineering

BME 400 – Design

Neural Bioreactor

Final Report

Maria Estevez – BWIG
Jeff Groskopf – Co-Leader
Tyler Klann – Co-Leader
Lisa Kohli – BSAC
Ian Linsmeier – Communicator

Dr. Tracy Puccinelli – Advisor
Dr. Krishanu Saha – Client

December 12, 2012

Abstract

Reprogramming adult cells into induced pluripotent stem cells (iPSCs), as well as their subsequent expansion and differentiation, is normally completed in adherent cell cultures. Recently, it has been proven that iPSCs can be derived and expanded in suspension cultures using a stirred suspension bioreactor. These reactors establish stable cell culture conditions by controlling the temperature as well as the level of nutrients (media), CO_2 (pH), O_2, and other soluble factors. The suspension components are uniformly distributed within the reactor fluid through various mixing techniques, most commonly an impeller. The process of adult cell reprogramming and iPSC expansion and differentiation can be scaled up and automated using bioreactor stirred suspension cultures.

Dr. Saha has asked our team to design a bioreactor that maximizes the production of neural progenitor cells from mouse embryonic fibroblasts in stirred suspension cultures. The project involves designing culture processes and optimizing culture conditions to reprogram adult cells to iPSCs and differentiate those iPSCs to neural progenitors. The team then determined three impeller design alternatives: a pitched blade impeller, a whisk-shaped impeller, or a flat blade impeller. We designed the pitch blade to maximize mixing with radial and axial flow while simultaneously minimizing shear stress endured by the stem cells. The bioreactor will be designed to function in an incubator hood. It will have a 100 mL glass vessel, a motor-powered non-cytotoxic pitch blade impeller, probes to monitor temperature and gas concentrations, and a yet to be designed media exchange system. Future work will be conducted to continue construction of the impeller, to finalize and construct the design of the rest of the bioreactor, and to proceed with cell testing for each step of construction.

Contents

Background:

Stem cells are characterized by their ability to remain pluripotent and differentiate into multiple adult cell types. In 1998, Dr. James Thomson was able to isolate cells from the inner cell mass of human embryos and developed the first human embryonic stem (hES) cell lines. These cells hold tremendous promise for advances in biomedical sciences and regenerative medicine; however, this has remained as a very controversial topic ever since the first hESCs were derived. Recently, induced pluripotent stem cells (iPSCs) were generated by genetically modifying differentiated cells to overexpress four genes that make the adult cells revert to a pluripotent state. iPS cells also have a distinct advantage over ES cells as they exhibit key characteristics of ES cells without the ethical dilemma of embryo destruction.

Stem cell cultures are mostly done in adherent culture conditions. This system involves growing cells as monolayers on different substrates which allows for easy visual inspection under the microscope. However, in adherent culture, cell growth is limited by surface area, and thus, it requires periodic passaging utilizing enzymes to lift the cells off the surface. This system is extensively used for research applications but is not ideal for the large-scale production of therapeutic cells. Approximately 10^9 cells would be required for any relevant clinical application and it would be very expensive and unfeasible to expand this number of cells in a monolayer system [14]. Stirred-suspension bioreactors are an alternative culture system where cell growth is no longer limited by surface area but by the concentration of cells in the media. Stirred-suspension bioreactors have the capacity of producing from 10^6 to 10^7 cells per milliliter, thus, solving the issue of scalability [5]. We would initially like to design and develop a 100ml bioreactor that could potentially yield an estimate of 10^8 cells, which theoretically would be enough for a clinical-scale procedure for a single patient.

Our client, Dr. Krishanu Saha, is interested in developing a bioreactor where adult cells can be reprogrammed to an immature, embryonic state, and then differentiated to neural progenitors. We will use secondary mouse embryonic fibroblasts (MEFs) that have already been encoded with the four transcription factors - Oct 4, Sox 2, c-Myc and Klf4-, necessary for reprogramming. In order to deliver these transcription factors to the cells, the DNA that encodes their production must be introduced into the genome of the adult somatic cell using retroviral vectors such as lentiviruses. These retroviral vectors deliver the transgenes encoded for the four reprogramming factors into the cells, and if all of them are successfully integrated to the adult cell's genome, the cell will start expressing the four transgenes as functional proteins. Addition of Doxycycline to the cell culture medium results in transcriptional activation of each of the transcription factors, thereby inducing the expression of genes that are exclusively expressed in pluripotent stem cells [18]. Figure 1 shows the reprogramming process and how the overexpression of the four transcription factors causes the adult fibroblasts to revert to a pluripotent like state. Our reprogramming experiments will be mostly based on recently published articles that successfully reprogrammed MEFs in stirred suspension bioreactors [2, 13]. The differentiation experiments will be also conducted following protocols that have successfully transformed iPSCs into more specialized cells.

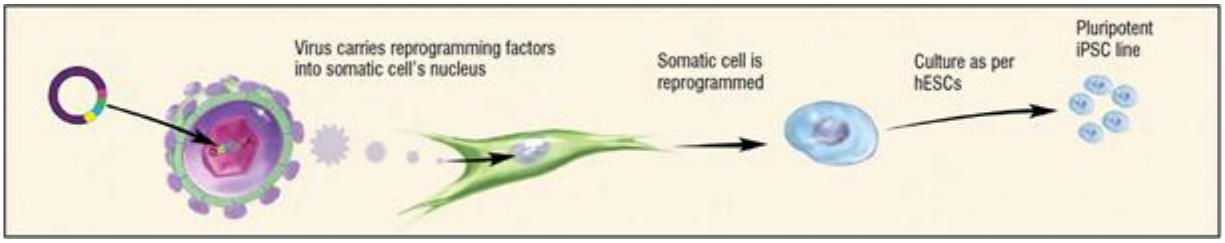

Virus carries reprogramming factors into somatic cell's nucleus
Somatic cell is reprogrammed
Culture as per hESCs
Pluripotent iPSC line

Figure 1: Generation of Induced Pluripotent Stem Cells [6]

Motivation:

Induced pluripotent stem cells have the potential to become a powerful research tool to understand and model diseases. In theory, skin fibroblasts from a patient could be obtained and reprogrammed in order to recapitulate the patient's disease in a research laboratory. These cells could be further differentiated to the cell types affected by the disease and serve as a study model [6]. Induced pluripotent stem cells have many potential clinical applications; however, it is imperative to overcome existing challenges before stem cell technologies can be translated to the clinic. One of the many current challenges is to find cost-effective ways of producing therapeutic cells. Although stirred-suspension bioreactors also present some challenges, it appears to be a suitable system to the expansion, reprogramming, and differentiation of large amounts of cells for clinical-grade stem cell therapeutics.

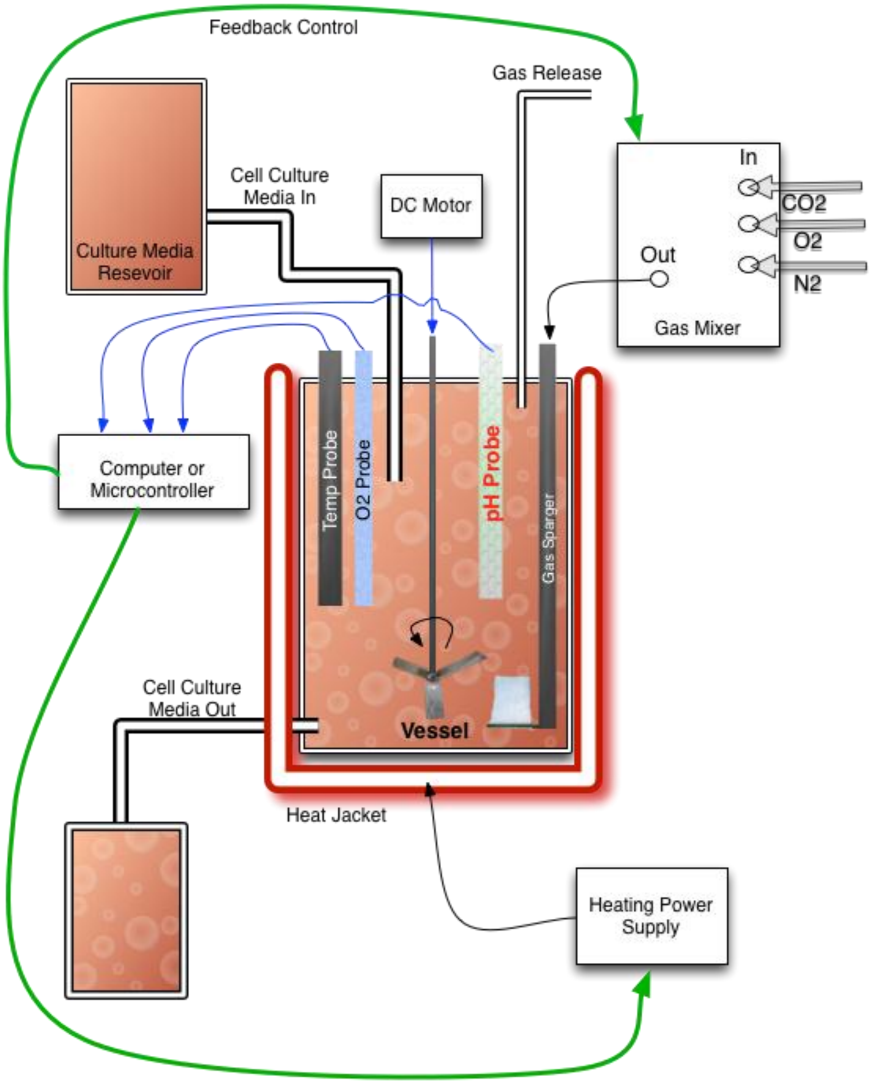


Figure 2: Stirred Suspension Bioreactor Flow Diagram

Overall Bioreactor Design:

Our system will consist of multiple components as illustrated in Figure 2. First, the main component of the device will be the vessel. Our vessel will be a cylinder shape and the volume of liquid will be determined by how many cells need to be produced. Initially, though, our vessel size will be designed to hold 125 mL of liquid including the impeller. Our impeller will be powered by a variable speed DC motor. The user of the device will determine the rate at which the impeller will rotate. The function of the impeller is to mix gas, nutrients, and cells evenly throughout the volume, as well as to break up large clumps of cells that may form.

Additionally, there will be three different sensors, pH, dissolved oxygen, and temperature, within the vessel to monitor the conditions the cells are exposed to. The pH sensor will measure indirectly how much dissolved CO_2 is in the cell culture media using the equations found in Figure 3. The dissolved oxygen probe will measure how much oxygen is in the system, since the device will be closed to the outside environment. The temperature sensor will detect the temperature of the cell culture media. All three sensors will be connected to a computer or microprocessor, which will record the readings of all three sensors over the course of an experiment. The computer will display the current readings of the sensor. Also, the temperature sensor will provide a feedback loop to the temperature controller on whether to keep the heater on or off.

$$P_{CO_2} = \frac{\%CO_2}{100\%} * 1Atm$$

$$[CO_2(aq)] = K_{CO_2} P_{CO_2}$$

$$K_{A1} = \frac{[H^+][HCO_3^-]}{[CO_2(aq)]} = \frac{[H^+]^2}{[CO_2(aq)]}$$

$$pH = -\log([H^+])$$

Figure 3: Equations Used For Calculating pH from Carbon Dioxide Concentration

Since the device is a closed system, there will not be exposure to atmospheric air. To provide a correct level of gas levels, our system will utilize a gas mixer. This mixer will be connected to carbon dioxide, oxygen, and nitrogen sources. Through changing input nitrogen and oxygen concentrations, the user will determine the oxygen concentrations to be either normal atmospheric or hypoxic conditions.. The carbon dioxide levels should be maintained at a constant 5% of total gas volume.

Cell culture media will need to be changed over the course of an experiment. For initial testing, this will be performed manually, but the final product should have a system which does this automatically. This will require a vessel that would act as a media reservoir, and another vessel where old media would be moved. Whether the system would filter out cells or just remove cells with old media has yet to be determined. If cells would be occluded from old media removal, a filter would be used to prevent cells from escaping.

Relating to whether the cells will be removed with old media is the idea of whether cells will be passaged in batch, or if they will be gradually removed with old media. For batch passage, the cells would be contained to the vessel with old media removed over time and new media replenished. The cells would eventually replicate and expand to fill the volume of the media. Once this occurs the entire contents of the vessel would need to be removed and cells would be separated from the media. The cells would then be seeded to a new vessel at the desired initial concentration and the process of growth would repeat.

The other option to passaging would be to remove cells with old media and separate those cells from the removed media. This would require the correct rate of media removal and replenishment so the expansion rate of the cells would be the same or lower than the rate of cell removal. Both options need to be research more and a final decision on one method over the other, or a combination of the two needs to be made.

Existing Devices:

Today there are hundreds of bioreactors on the market. These bioreactors are used for many different applications including the growth of cells or tissues in a suspension culture. Many companies offer complex bioreactors systems such as the ones offered by Eppendorf; however, these bioreactors are not applicable to our client for two main reasons. The first reason is that these bioreactors are designed for generalized or specific cell and tissue growth; unfortunately, there are no bioreactors specific to stem cells. Therefore, if our client were to purchase a bioreactor, this bioreactor would then need to be specialized for the process of reprogramming adult cells to iPSCs and then differentiating them into neural progenitors.

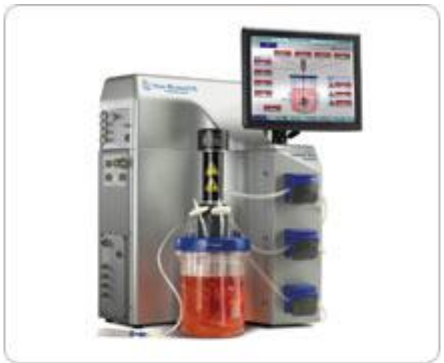


Figure 4: Eppendorf CelliGen BLU single-use bioreactor [15].

The second reason is that these bioreactors are simply out of our client's budget. For instance, the CelliGen BLU single-use bioreactor, Figure 4, offered by Eppendorf costs $32,000 with an additional cost of $700 per reaction vessel [4]. This is especially expensive when these bioreactors will need further customization, leading to an additional cost, for them to be useful for our client. At this point, it would be more cost effective and efficient for a bioreactor to be designed and developed for this specific process.

Another option to be used in the development of a bioreactor is to purchase a spinner flask for use as the vessel. Spinner flasks come in a variety of sizes and can be either reusable or disposable. A triangular shaped impeller connected to a stir bar along with baffles produces the agitation necessary for bioreactors. Several companies sell spinner flask, including Corning. Corning offers a 125 mL reusable glass spinner flask that cost $365 [3]. Spinner flask could be a viable option depending on the shear stresses that the flat blade impeller would produce. These shear stresses are analyzed and discussed in later sections.

Client Requirements:

Our client has requested a few parameters be met, but ultimately has decided to allow us to make most of the design decisions. The biggest design consideration that must be accomplished is to produce the required amount of cells. For tissue engineering applications, the amount of cells required would be on the order of magnitude of 10^9 differentiated cells. To expand the amount of cells you input to the reactor, basic cell culture conditions must be met, such as 21% oxygen, 5% carbon dioxide, and a temperature of 37˚ Celsius.

Additionally, our client would like the bioreactor system to be an automated closed system. For the final design, you would be able to take a skin biopsy and inoculate it to the reactor to reprogram those cells into induced pluripotent stem cells. Then those reprogrammed cells would need to be expanded to a larger number of cells and then reprogrammed into the desired cell type. There should be minimal interaction and maintenance needed on the bioreactor between steps to simplify the production of the desired type of cells.

Finally, the cell culture media should be serum free. This is because using serum obtained from animal sources introduces unknown factors which can affect the cells produced. If using human cells, growing those cells in fetal bovine serum could contaminate the cells with infection of bovine disease or cause immune reaction to foreign proteins. If the cells are grown in defined conditions, these possible contaminations can be avoided.

Impeller Design Overview:

All stirred suspension bioreactors require an agitation system to control fluid flow, shear stresses, mass and heat transportation. For this bioreactor design, the agitation system will consist of a motor that will rotate a shaft that an impeller is attached to. The motor will rotate the impeller at a rate of 75 revolutions per minute, or RPM, as this was found to produce the best cellular growth [10]. This type of turbulence will cause the bioreactor's media to observe different fluid dynamics depending on the impeller design. The impeller design could cause axial or radial flow or a combination of both to occur within the bioreactor. Axial flow is the movement of the fluid along the axis of the impeller while radial flow pushes the fluid away from the impeller axis towards the vessels wall. Each flow type can be seen in Figure 5.

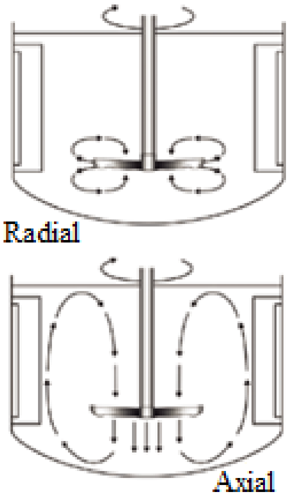


Figure 5: Graphical representations of axial and radial flow [11]

The type of flow the impeller design creates will affect the other parameters that the agitation system controls. The impeller design must be able to efficiently mix the media within the bioreactor. The mixing affects both the mass and heat transportation. Having efficient mixing will allow the heat to disperse uniformly throughout the media, causing the bioreactor to remain at a relatively constant temperature of 37° C. Also, the mixing will allow for carbon dioxide, oxygen and nutrient mass transportation to occur. This is important to ensuring that the bioreactor is at the correct conditions to allow for the cells to proliferate, reprogram and differentiate.

The impeller design must be able to produce shear stresses on the cells within an optimal range. This optimal shear stress range is specific to the cell line being used in the bioreactor, initially in this case mouse 3T3 cells. The optimal range means that the shear stress is low enough to avoid causing cell damage but high enough that the stress is able to break apart cell aggregates. A favorable shear stress value for mouse embryonic stem cells was 0.61 Pascal and 0.21 Pascal for mammary epithelial stem cells which gave an approximate goal for the impeller designs' shear stress [16]. These values were estimated by assuming that the maximum shear stress in bioreactors are caused by Kolmogorov eddies. Therefore, the estimated values were calculated by Youn et. al using the equation $\tau_{max} = 5.33\rho(\epsilon v)$, where ϵ is the power dissipated per unit mass, v is the kinematic viscosity and ρ is the fluid density [20]. To determine the optimal shear stress range and which impeller will achieve this value, fluid molding of the velocity profiles was performed using Solidworks. These velocities were then translated into shear stresses and plotted using a Matlab program. This analysis of shear stresses produced by the impeller is explained in further detail in fluid modeling and shear stress analysis sections.

Using this information about impellers, three impeller designs were developed to meet the above parameters. Unfortunately, an impeller cannot optimize mixing and have low shear stress at the same time. To keep shear stress low, there is a loss in the mixing efficiency. Therefore, the best impeller design will create low shear stress while still keeping an appropriate mixing efficiency for mass and heat transportation.

Pitched Blade Impeller:

The first design is referred to as the Pitched Blade Impeller. This design is a common impeller used for many applications including bioreactors. The impeller consists of 3 flat blades, equally spaced apart, and attached to a shaft at a 45° angle. Both axial and radial flows are supposed to be produced by this type of impeller. Because it produces both types of flow, this impeller type will have better overall mixing for mass and heat transportation than just axial or radial flow type impellers [11]. Also, the flow patterns of this impeller cause gentle mixing of the media. This means that the shear stress acting on the cells will be low and should cause little cell damage. Pitched blade impellers have been used

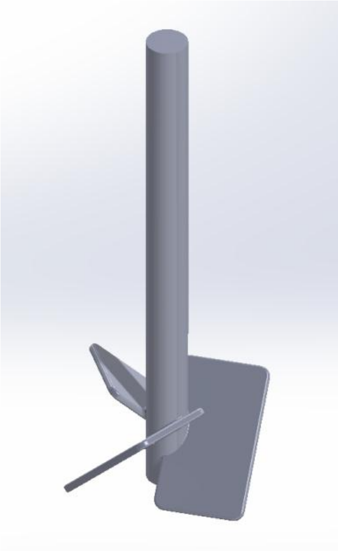


Figure 6: SolidWorks model of the pitched blade impeller. The dimensions of this design are shown in Figures B1, B2 in Appendix B.

on many mammalian cell lines and specifically should be compatible with the mouse 3T3 cell line [13].

Our pitched blade impeller design follows the general form of having 3 blades attached to the shaft at a 45° angle as Figure 6 shows. The impeller blade dimensions are 17.5 mm by 30 mm by 1.5 mm. Fluid modeling was performed for the pitch blade impeller. The overall impeller geometry was rotated at 75 rpm (7.85 rad/s) within a vessel 80 mm tall and 50 mm in diameter. The vessel was filled completely with water. The fluid model was under the influence of gravity and was analyzed over a period of time. Through the flow analysis in SolidWorks, Figure 7, it was determined that this impeller design does produce both axial and radial flow simultaneously. The fluid velocities of the flow analysis are mostly within the range of 0.060 - 0.099 m/s throughout the entire vessel with some peaking around 0.119 m/s near the blades. This shows that the entire vessel is undergoing moderate mixing while keeping the shear stresses inside the vessel low.

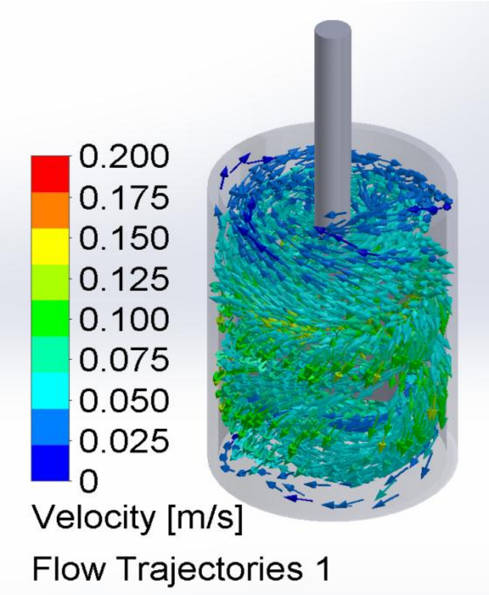


Figure 7: Solidworks flow analysis model ofor the pitched blade impeller.

From the qualitative analysis, it was determined that the shear stresses created by the impeller should not cause cell damage and should be able to break up the cell aggregates. Also, this impeller design should be able to spread the heat throughout the entire vessel generating an approximately uniform temperature throughout the bioreactor. The mixing created by the pitched blade impeller should have the capability for carbon dioxide, oxygen and nutrient mass transportation to occur. The pitched blade impeller design meets the necessary requirements of a bioreactor to help proliferate, reprogram and differentiate cells.

Whisk Impeller:

The second impeller configuration considered for the bioreactor is the whisk impeller (Figure 8), shaped much like its kitchen namesake. This impeller features 4 flat blades evenly spaced about the central rod. Each blade has a removed interior region that extends from the central rod towards the periphery of the blade. The whisk impeller blades extend the height of the bioreactor for fluid flow generation throughout the entire vessel space. The dimensions of this design are as follows: the overall diameter is 30 mm, the blade height is 70 mm, the blade width is 1.5 mm, the removed region of each blade extends from the central rod to 5 mm from the blade perimeter in each

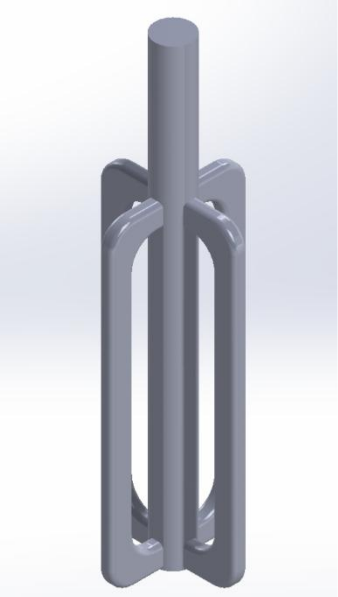


Figure 8: Solidworks model of the whisk impeller. The dimensions of this design are shown in Figure B3 in Appendix B.

direction, and the central rod has a diameter of 8 mm; the design is shown in Figure 8 with its dimensions displayed in Figure B3 of Appendix B.

Fluid Modeling was performed for this design to determine its efficacy for the parameters discussed in the Impeller Design Overview section. The modeling was performed under the same conditions stated for the pitch blade impeller. A snapshot of the resulting velocity profile can be seen in Figure 9. Through qualitative analysis of the resulting velocity profile, it can be seen that this design creates mainly radial flow, with very little axial flow. Towards the perimeter of the vessel there are regions of static fluid, indicated by dark blue velocity vectors, which is a product of poor mixing. Additionally, there are large velocity gradients, designated by a drastic change in velocity vector color within the gaps of the blades. Large velocity gradients indicate high shear stresses within that region. This is alarming since the cells may undergo damage or death under the influence of high shear stresses.

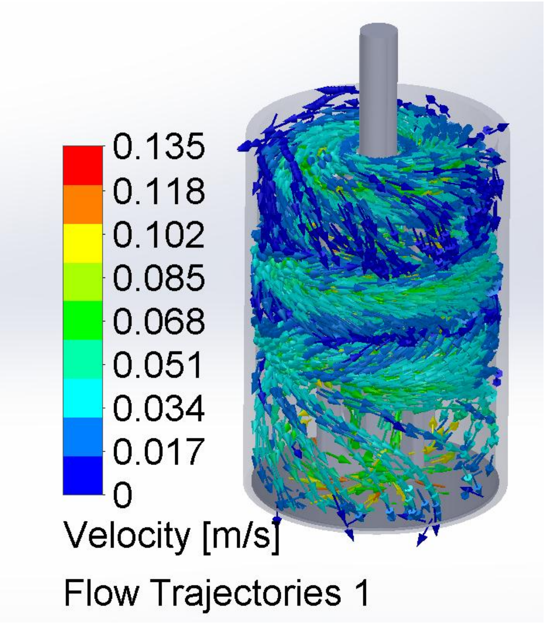


Figure 9: Whisk Impeller Fluid Model. This is a snapshot of the resulting velocity profile for the fluid simulation of the whisk impeller. The legend on the left shows the magnitudes of the relative velocities (meters/second) within the vessel.

Flat Blade:

The third design considered for the bioreactor is the flat blade impeller (Figure 10). This design was modeled after the impeller used for spinner flasks marketed by various companies. Spinner flasks have been proven as viable small-scale bioreactors for the production mouse induced pluripotent stem cells [5] [17]. This impeller has two triangular blades protruding from the central rod. These blades have a slight inward taper as they extend the entire height of the bioreactor vessel. The large base diameter of this design is intended to create uniform flow throughout the entire fluid region. The dimensions of this impeller are as follows: the overall diameter is 43 mm, the blade width is 1.5 mm, the blade height is 75 mm,

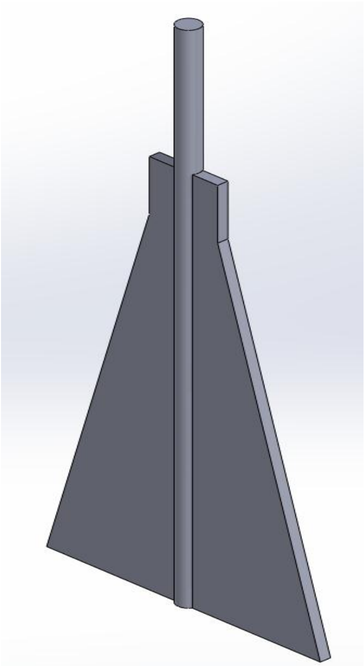


Figure 10: SolidWorks model of the Flat Blade Impeller. The dimensions for this design are shown in Figure B4, B5 in Appendix B.

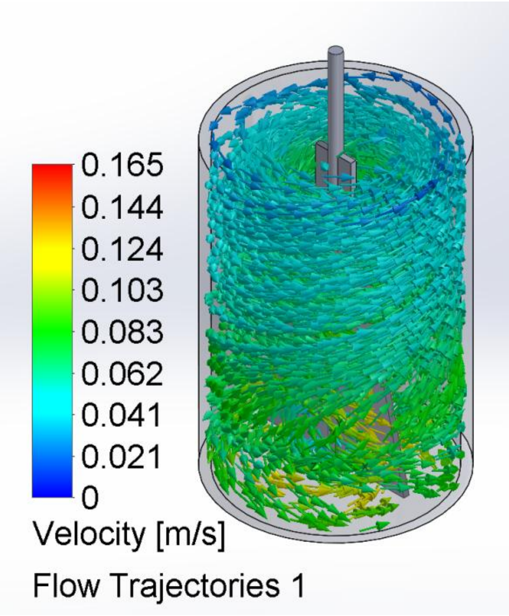


Figure 11: Flat Blade Fluid Model. This is a snapshot of the resulting velocity profile for the fluid simulation of the flat blade impeller. The legend on the left shows the magnitudes of the relative velocities (meters/second) within the vessel.

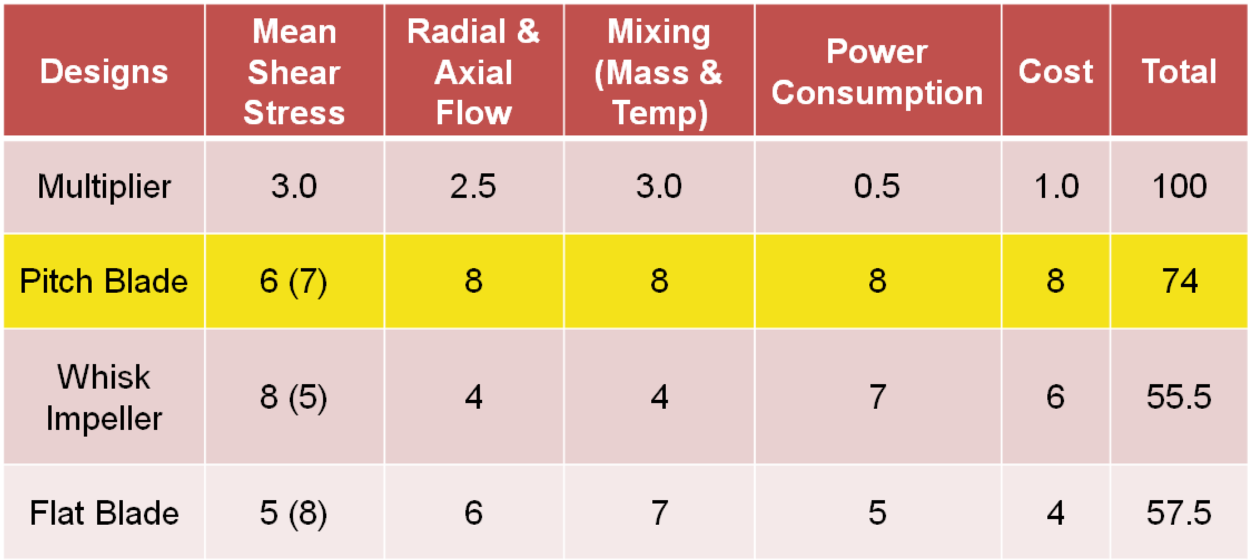
and the central rod is 3 mm in diameter; these dimensions are shown in in Figure 10.

Fluid modeling was performed for the flat blade impeller under the same conditions stated for the pitch blade impeller. However, the rotating geometry was changed to match the geometry of the flat blade impeller. A snapshot of the resulting velocity vector field is shown in Figure 11. After examining the fluid model qualitatively, it can be seen that the flat blade impeller creates mean radial flow, indicating low shear stress, except for near the base of the impeller. However, this design fails to generate axial flow or substantial mixing; this was concluded from the uniform flow in the radial direction and overall lack of turbulence.

Design Matrix:

The design matrix (Table 1) used to evaluate the three impeller designs was based upon five categories: mean shear stress, radial and axial flow, mixing, power consumption and cost. The three impeller designs were rated from 1-10 for each category. The individual category scores were multiplied by the corresponding multiplier, accumulating to a maximum total score of 100 for a single design.

Table 1. Design Matrix. This matrix contains the five categories that each impeller design was rated against. The pitch blade impeller (highlighted in yellow) scored the highest and was chosen for the final design. Values in parentheses are initial mid-semester values.



Designs	Mean Shear Stress	Radial & Axial Flow	Mixing (Mass & Temp)	Power Consumption	Cost	Total
Multiplier	3.0	2.5	3.0	0.5	1.0	100
Pitch Blade	6 (7)	8	8	8	8	74
Whisk Impeller	8 (5)	4	4	7	6	55.5
Flat Blade	5 (8)	6	7	5	4	57.5

Initially, the three designs were rate against the five categories based upon the qualitative analysis of the resulting fluid flow profiles as well as information gathered from literature. After creating the Matlab shear stress calculation code, the fluid flow simulation results were used to calculate the maximum shear stresses across two planes of fluid flow, as described by the general process of calculation in the Shear Stress section below. The resulting data somewhat contradicted our qualitative analysis of mean shear stress against the three preliminary designs. Our initial scores can be seen in parentheses in the design matrix above. However, the matrix has since been updated to account for the newly acquired data, and the new totals are shown above. As you can see, alteration of our scoring did not affect the final design choice, but instead made the pitch blade impeller the clear favorite for the three designs. The shear stress data used to score mean shear stress for the three designs can be seen in Table 2.

Table 2. Average maximum shear stress calculations for the three preliminary impeller designs. The calculation methods to obtain this data are outlined in the shear stress section below.

	Shear Stress Calculations for Preliminary Impeller Designs					
Impeller Type	Pitch Blade		Flat Blade		Whisk Blade	
Calculation Plane	Front	Top	Front	Top	Front	Top
Max Average Shear Stress (Pa)	0.01999	0.067173	0.051893	0.063978	0.026857	0.034815
Note: The shear stress calculations were determined at a rotational speed of 75 rpm						

Amongst the design matrix categories, mean shear stress and mixing were the two categories that were weighted the highest with a multiplier of 3 for each. These two design parameters are most important for creating viable cell culture conditions within the bioreactor; the rationale for their importance is outlined in the Impeller Design Overview section. A single design cannot simultaneously optimize both of these features since there is a trade-off from one another. This means that improving one will adversely affect the other and vice versa. Therefore, the ideal design will achieve mean shear stress within an acceptable range, while creating enough mixing to provide nutrients and diffused gas to the cells within the uniformly heated media.

Mean shear stress is a function of the velocity gradients within the flowing fluid. This was rated for each design based upon the calculated maximum average shear stress, which can be seen in Table 2. The design with the lowest mean shear stress was the whisk impeller due to its lower velocity flows within the fluid space.

The degree of mixing generated by each impeller was also determined through qualitative analysis of the fluid modeling. For this category the fluid velocity animation videos were observed. The level of mixing was based upon the change in both the magnitude and direction of velocity throughout the entire fluid space. The flat blade impeller created flow throughout the entire fluid volume, but failed to create changes in the magnitude or direction of the flow velocity. The whisk blade impeller generated a high level of mixing near the center of the vessel; however, the presence of dead space severely hurt its evaluation for this category. The pitch blade impeller provided the highest level of mixing due to the constantly changing velocity directions within the fluid. The high level of mixing observed for the pitch blade impeller was further supported by computational fluid modeling and testing performed by Kumaresan et. al [8].

The generation of radial and axial flow was assigned the next highest multiplier at 2.5. Qualitative analysis of the fluid modeling was again used to evaluate this category. The generation of flow in the radial and axial directions within the vessel is more self-explanatory from the velocity profiles. It can be seen that the pitch blade impeller best generated flow in both directions. This flow generation is a product of the 45^o angled impeller blades that equally create flow along the two axes.

The proof of concept impeller design will be 3D printed using the Viper si2 SLA printer which uses an epoxy material called Accura 60. Thus, the cost of each design was estimated based on the volume of Accura 60 required to print each design. Since each impeller is fairly small and relatively the same volume, cost was only assigned a multiplier of 1 in the design matrix. It was estimated that the flat blade would cost the most, followed by the whisk impeller, and lastly the pitch blade impeller. Since the pitch blade was the cheapest, it scored the highest in the design matrix for this category.

Classical impeller design is based around the amount of power necessary to generate flow within a reaction vessel [8]. Although the impeller designs presented for the bioreactor are small enough where power consumption should not be an issue, it was deemed necessary to include it in the evaluation of each design. Since the bioreactor is being designed with the intention of scale-up in the future, power use would become a more important factor and thus should be given at least some consideration. For this reason power consumption was allotted a 0.5 multiplier within the design matrix. The level of consumed power for each design was estimated based upon the surface area of the blades normal to the rotational direction of the impeller. This is the effective area that generates the fluid flow. Generally, increasing this surface area will increase the amount of power consumed by the impeller [8]. The pitch blade impeller scored the highest due to its small three blades, which consumed the least amount of power. The whisk impeller was a close second for this category.

Based upon the aforementioned design considerations, the overall scores were summed for each design and they are as follows: the pitch blade impeller scored a 74/100, the whisk impeller scored a 55.5/100, and the flat blade impeller scored a 57.5/100. As the highest scoring design, the pitch blade impeller was chosen as the final design. The flat blade impeller scored the second highest, with the whisk impeller a close third. also a notable design. Since the flat blade impeller has been proven effective in recent spinner flask experiments [5] [17], it will used as a control against our final design in future cell culture experiments.

Fluid Modeling

Fluid modeling was performed in SolidWorks for each of the the impeller designs to generate quantitative fluid velocity profiles. These profiles were initially used for qualitative analysis of the flow characteristics, shear stress within the fluid, and overall mixing. Later, point parameter arrays were generated across 2D velocity profile planes to obtain instantanous velocity data at specific points within the fluid. From these data, shear stress estimations were calculated between adjacent point parameters. An in depth description of the fluid modeling methodology is described in Appendix C. The shear stress calculations are explained further in the next section.

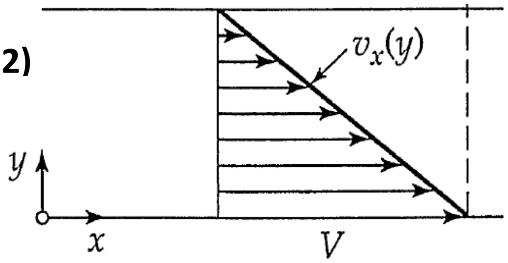
Shear Stress

To calculate shear stress within a fluid, one must employ Newton's Law of viscosity (Figure 12). This fluid law states that the shear stress between layers of fluid (laminar flow) is proportional to the viscosity of the fluid as well as the negative velocity gradient between the layers separated by infinitesimal distances. For the specific 2D representation of newton's law of viscosity shown in Figure 12-2, 3, shear stress is equal to the force in the x-direction on a unit area perpendicular to the y-direction. Or stated another way, the shear stress is equivalent to the flux of x-momentum in the positive y-direction.

(1) $$\sum_i \sum_j \tau_{ij} = -\mu \frac{\partial v_i}{\partial x_j}$$

i, j=1, 2, 3

(2)



(3) $$\tau_{xy} \cong -\mu \frac{\Delta v_x}{\Delta y}$$

Figure 12. Newton's Law of viscosity. 1.) Newton's Law of viscosity accounting for all 9 components of the shear stress tensor. τ_{ij} is the shear stress, μ is the fluid viscosity, and $\frac{\partial v_i}{\partial x_j}$ is the velocity gradient. I & j are equal to 1, 2, and 3 corresponding to the x, y, and z components of the Cartesian coordinate system, and successively summing to all 9 components. 2.) 2D representation of a velocity gradient. 3.) Shear stress approximation corresponding to the velocity gradient in 2.) for changes in velocity over a finite distance.

Shear stress is a 3D fluid tensor quantity, meaning it has 9-components that fully describe its magnitude and direction (Figure 12-1). However, shear stress can be broken down into its component forms and calculated in 2D before extending the quantity to 3D. Whilst in 2D, one is able to more easily make estimate calculations from real-world data, since there isn't a standard method of calculating fluid shear stress in 3D.

Using our fluid modeling results, we used the 2D estimation method to calculate various components of shear stress within the fluid for different impellers at various rotational speeds. First, a velocity profile was generated using the aforementioned fluid modeling method (Figure 13-1). Then, the velocity was plotted across a chosen plane within the model (Figure 13-2) and an array of point parameters was generated over that surface (Figure 13-3). We chose to use the front plane (x-y plane), that bisected the impeller longitudinally and extends along the entire height of the bioreactor vessel for each fluid model. Additionally, point parameters were plotted across the top plane (x-z plane), which bisects the impeller blades latitudinally and extends across the entire circular cross section of the bioreactor vessel. The point parameters were each assigned their relative position within the coordinate axis (x, y, z) in meters, as well as the instantaneous velocity components (v_x, v_y, v_z) in m/s at that point in space. The maximum number of point parameters was plotted across each plane as permitted by the software; this means that the points were spaced as close together as allowed by the software. Most models had a minimum distance between the point parameters around ~850µm. From this data, a MATLAB program calculated

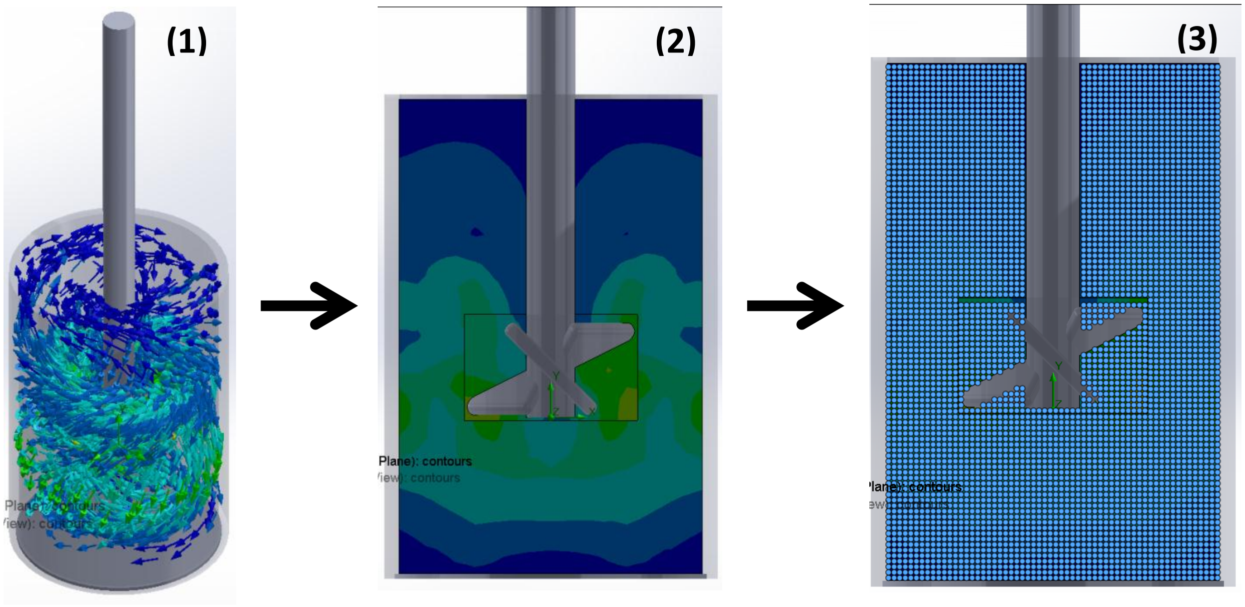


Figure 13. Fluid Modeling Process: 1.) Generation of a velocity profile at a certain rotational speed – 75 rpm shown above 2.) 2D plot of velocity magnitude across a plane – Front plane shown above (x-y plane) 3.) Insert an array of point parameters over the 2D velocity plot with the smallest possible distance between the points. Each point has v_x, v_y, and v_z components of velocity associated with its specific point in space.

the change in velocity components between adjacent points in space both across the rows and down the columns of the point parameter array. From a single plane, 6 components of the shear stress tensor can be calculated using the equation from Figure 12-3. For example, using the Front plane (x-y), one can calculate τ_{xx}, τ_{yx}, τ_{yz}, from the change in v_x, $v_{y,}$ and v_z over the fixed distance Δx (Figure 14) across each of the rows in the plane. Additionally, one can calculate τ_{xy}, τ_{yy} τ_{zy} from the change in v_x, $v_{y,}$ and v_z over the fixed distance Δy (Figure 14) down each of the columns for the same front plane (x-y plane). An additional plane would be needed to calculate the remaining three shear stress components (τ_{xz}, τ_{yz} τ_{zz}). However, calculating the overall magnitude of the shear stress seemed superfluous since it would be representative of a volume of space 850µm x 850µm x 850µm. Although this seems like a very small space, we are calculating the shear stresses that will be acting on cells. 850µm is very large in comparison to a single cell and would not be a representative value. Thus, we calculated and evaluated shear stress solely in its component form for a single point in space. Maximum shear stress values were calculated in Pascals for each component of velocity. This was performed iteratively across the aforementioned planes for various impellers over a range of speeds. Data was collected was the preliminary impeller models and can be seen in Table 2. Additionally, maximum shear stress values were calculated for the final pitch blade impeller over a range of rotational speeds to determine the optimal rotational speed; this data can be seen in Table 3. From this data, one can see that the

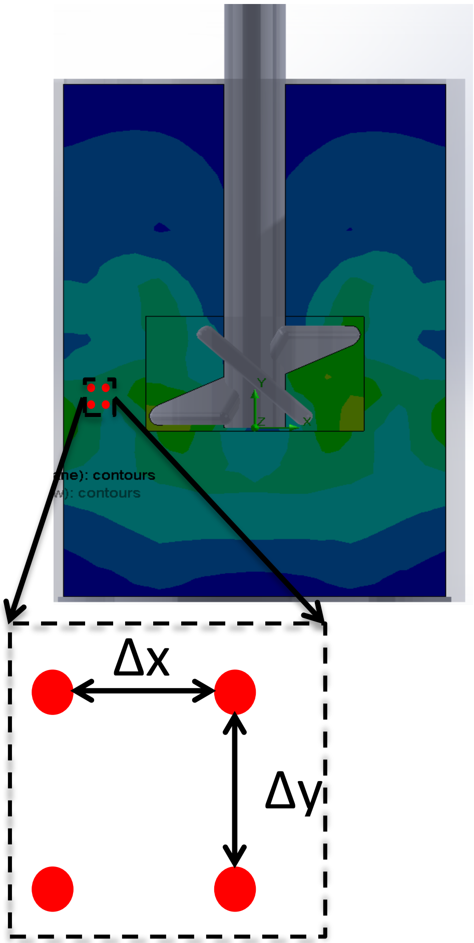


Figure 14. Minimum distance between adjacent point parameters: shown is an enlarged representation of the fixed distances between point parameters in the array shown in Figure 13-3. Using the data from adjacent points in conjunction with the equation shown in Figure 12-3 allows one to calculate various components of shear stress.

optimal speed for minimizing shear stress in both the front and top plane is about 100 rpm. Interestingly, 75 rpm seems to maximize the shear stress in the top plane near the impeller blades. This verifies that out preliminary fluid models, which were run at 75 rpm, were testing the worst case scenario for the shear stresses imposed upon the cells.

It was decided to calculate maximum shear stresses within a plane instead of average values for two reasons. First, the majority of the shear stresses within the fluid are very low and the average of them calculates to be a relatively small number; this makes it difficult to pinpoint design flaws, since only a small region of high shear stress can cause cell death. Thus, we decided to focus on maximum values or average maximum values based upon multiple components to obtain values for the highest possible shear that the cells may experience.

Table 3. Shear stress calculation data for the final pitch blade impeller across a range of rotational speeds from 65-115 rpm. Maximum shear stress values were calculated in Pascals for both the front and top plane.

Final Pitch Blade Impeller Shear Stress vs. RPM

Rotational Speed (rpm)	Front Plane Max Shear (Pa)	Top Plane Max Shear (Pa)
65	0.022439	0.064748
75	0.01999	0.067173
85	0.01999	0.059951
95	0.020939	0.060168
105	0.022309	0.060105
115	0.024683	0.060022

As stated earlier, velocity profiles of thousands of points were collected using SolidWorks. Because the software available to the team did not have shear stress calculating capabilities compatible with our impeller models, a MATLAB program was written to analyze the points and velocity data, which can be found in Appendix D. The program opens a file, determines what plane it is observing based on the file name, and then makes shear stress calculations between the given points in a matter appropriate for that plane. To calculate the shear stress between points, the program uses a single flat plane of data centered on the impeller. Then, it finds two points in the same row then finds the product of the dynamic viscosity and the velocity difference between the points divided by the horizontal distance between the points, creating what is labeled as 'shearx' in the program. A similar method is then used to find two points in the same column then to find the product of the dynamic viscosity and the velocity difference between the points divided by the vertical distance between points. The estimated dynamic viscosity of the culture at 40°C was assigned the value of 0.0006553 [3]. The program then found the magnitude of shearx and sheary at a point, using this to calculate the maximum and minimum shear stresses at a given point, which are then displayed for each file analyzed. For bird's eye views (top plane) of the modeled bioreactor, the program displayed a plot of discrete points with color indicating the shear stress magnitudes, one for each x, y, and z velocity. For frontal plane views of the modeled bioreactor, the program returns a set of plots: a plot of discrete points with color indicating the shear stress magnitudes (Figure 15-1), a meshed plot of shear in the x direction (Figure 15-2), a meshed plot of shear in the y direction of the plane. A set would then be displayed once for shear based on the x velocity, y velocity, and z velocity.

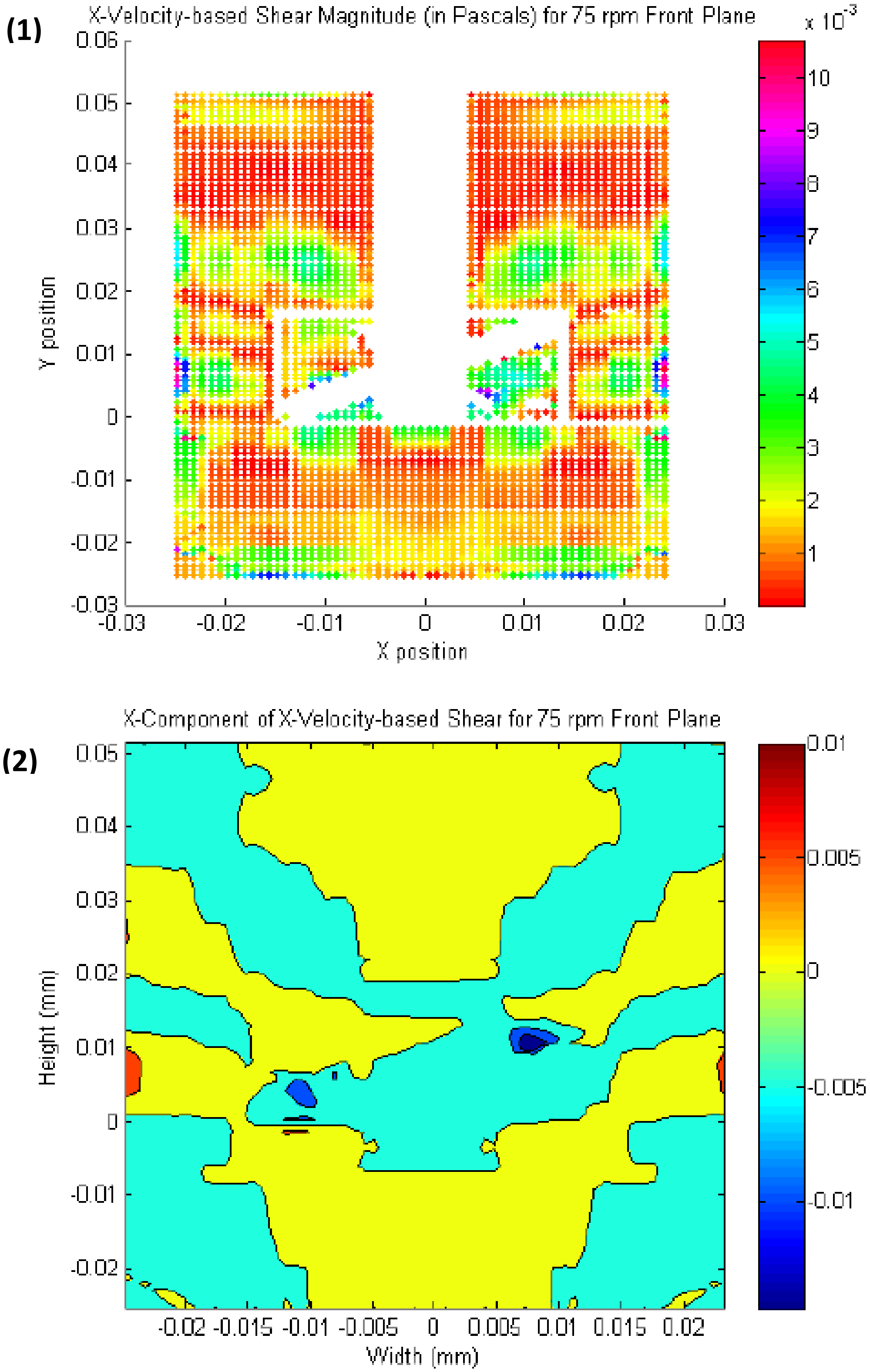


Figure 15. Shear Stress plots. 1.) Discrete points of shear stress with the magnitudes represented by the displayed color 2.) Meshed shear stress plot with the shear stress in certain regions represented by the displayed color.

Final Design:

Impeller

Based on the scoring of the design matrix of the potential impeller designs, the pitch blade impeller design was chosen for this project. The pitch blade was initially chosen for its predicted success of mixing, both radially and axially. Good mixing of vessel contents should maximize the efficacy of the heating and media exchange systems by ensuring the most uniform culture possible. Impeller speed was determined from shear stresses calculated from SolidWorks velocity profiles and from qualitative mixing tests with the Accura60 plastic impeller pitch blade. From the results of the shear stress calculations and mixing time testing, the desired rotational speed is near 100 rpm, which agrees with the research performed by Shafa et. al [17]. The CAD model of the impeller can be seen in Figure 16, with the precise dimensions displayed in Figures B6-B9 in Appendix B. The 3D-printed Accura60 impeller in the 100 mL bottle can be seen in Figure 17.

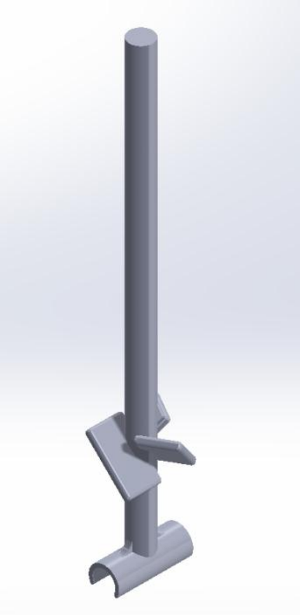


Figure 16. CAD model of the final impeller design. This CAD drawing was used to 3D print the part in Accura60. The dimensions are shown in Figures B6-B9 in Appendix B.

Initial Impeller Manufacturing Results

The designed impeller was manufactured using the Viper si2 SLA 3D printer from a photopolymerizable epoxy called Accura60. This method was used for its resolution and ease of construction. To test whether the Accura60 was toxic to cells, we performed the experiment outlined in Appendix G

For the coated samples used, we took small (approximately 3 mm x 1 cm x 1 cm) samples of the epoxy material and coated it in polystyrene (PS). To coat the samples we dissolved solid PS in toluene at a 1 to 3 ratio by heating and mixing. The small epoxy pieces were then dip coated in the dissolved PS. Two different methods were used to dip coat. The first method, which we call oven dried, involved submerging the epoxy sample in the dissolved PS, taking the sample out and placing it in an oven at 60˚ C overnight. The second method, called heat dried, involved dipping the sample in the dissolved PS once, then using a heat gun, quickly drying the surface of the sample for 2-5 minutes. Next the sample was dipped into the submerged PS again and heat dried again. After two dip coats were applied, the samples were placed in an oven at 60˚ C overnight.

In addition to coating samples in polystyrene, a coating of polyethylene glycol (PEG) was created on the surface of the sample. This procedure is outlined in Appendix F and was performed by Jin Sha, a visiting graduate student in the Ashton Lab.

After obtaining all the coated samples as well as uncoated samples, the experiment was performed, culturing cells with the materials in 6 well plates (outlined in Appendix G). After day 1, the cells in the wells of the oven dried seemed to have survived the most, with some cells in the heat dried well. Most cells seemed to have died in the PEG coated and uncoated wells. However, upon day 2, the cells in every well had died. We believe this was due to the Accura60 material leaching through the coatings into the cell culture media and killing the cells. This is possibly due to the thin layer of polystyrene and PEG coated as well as possible surface defects on the coatings of the samples. Going forward, we are going to have to find a non-cytotoxic material to manufacture the impeller from.

Vessel

The vessel of the bioreactor is a cylindrical 100 mL glass bottle with about 115 mL of culture liquid. The relative abundance of cylindrical bottles provides a good starting point as the price is favorable and they are readily available. Also, the cylinder is a shape that is convenient for axial mixing without any corners or extreme angles to complicate the circulation of the cells and media in the vessel. As with most bottles, the opening is on the top circular face of the cylinder, from where the impellers and probes can enter the bioreactor and form a top lid to create a closed system. As it is made of glass, the vessel is autoclavable and reusable for the bioreactor.

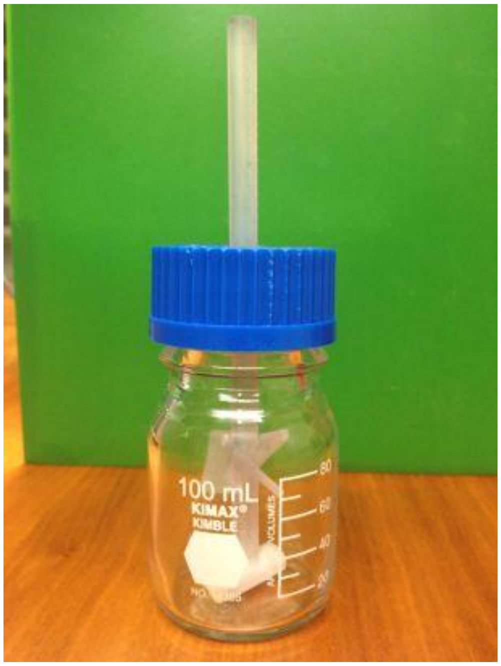


Figure 17: Photo of Accura60 impeller with magnetic stir bar in 100 mL bottle

Incubator Use

The environment in which the bioreactor is to be designed dictates the components necessary to maintain a constant, controlled environment. For instance if the bioreactor is to be placed on a lab bench, it would be designed as a stand-alone unit as opposed to being placed within an incubator hood and designed accordingly.

As a stand-alone closed system unit, the bioreactor itself would be responsible for the maintenance of specific pH, oxygen, and temperature levels. This would require the purchase of multiple probes to monitor levels and an expensive gas mixer to deliver the appropriate levels of oxygen, carbon dioxide, and air to the vessel. Since the temperature and gas controls are taken care of by the incubator, a bioreactor designed for the incubator would be significantly cheaper because it would not require the purchase of a costly pH probe, dissolved oxygen probe, or gas mixer. If part of a bioreactor's maintenance system were to fail or require maintenance attention, it might be more difficult to troubleshoot and fix the issue for a stand-alone bioreactor since it would be a custom-built system rather than relying on a more common system such as an incubator.

At the same time, a stand-alone unit offers its own advantages. For instance, the unit would not require the help of an incubator so would not be restricted by a hood size when scaling up and might make transportation simpler if the bioreactor vessel contains cells without dependency on an incubator hood. Creating an affordable stand-alone bioreactor unit would be a tremendous achievement, but given the time and financial limitations, a bioreactor designed for use within an incubator is significantly more feasible. For this reason, the team decided to construct a bioreactor designed for

If the team had decided to design a closed-system independent bioreactor, additional steps would have been necessary to maintain proper temperature and gas levels within the bioreactor. To manage heating, a temperature probe would detect the temperature and send its measurements to a microcontroller, which would then turn on or off a solid state relay connected to a nichrome wire wrapped around the vessel, and electrically insulated with Kapton tape. Nichrome was chosen due to its widespread use as a heating element and low oxidation when heated. Nichrome's resistance increases as its temperature rises [19]; thus, to achieve a relatively low heated temperature of 37°C the Nichrome wire would require very little voltage. The Kapton tape, a polyimide film with silicone adhesive [9], would electrically insulate the nichrome wire since they are compatible with a wide temperature range from -269°C to 400°C, and thus is at lower risk of incurring damage from the heated nichrome. The microcontroller would dictate whether the heating nichrome wire is actively heating depending on whether the feedback data indicates if more heating is necessary to maintain the specified goal temperature of 37°C.

However, the team has ultimately decided that the bioreactor should be first designed to be in an incubator hood because of financial and time constraints. Therefore, the temperature system simply should be responsible for monitoring the temperature of its contents rather than controlling the temperature. Also, the decision to use an incubator also means that the incubator itself, not a separate gas mixer, will control the gases that the culture is exposed to. The bioreactor culture will require a maintained level of about 5% CO2 to control pH and a hypoxic O2 concentration if the incubator allows since stem cells have been shown to thrive under hypoxic conditions [7].

The bioreactor will also require a media exchange system to refresh the cell media as necessary as will be discussed in the "Future Works" section.

Temperature Probe

The temperature of a bioreactor must remain at approximately 37°C to have the correct culture conditions for cell growth. If the temperature of the media fluctuate more than a couple of degrees from 37°C, cell death can occur ruining the efficiency of the bioreactor. Therefore, it is necessary for the temperature within the bioreactor to be monitored. A Miniature Pt100 Resistance Temperature Detector, RTD, sensor was purchased from Auber Instruments Incorporated. This sensor, Figure 18, is 10 mm in length with a 2.0 mm diameter that is made out of stainless steel and a perfluoroalkoxy, PFA, wire [2]. Both of the materials are non-cytotoxic which allows the probe to be

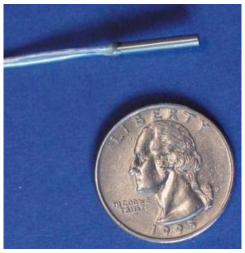


Figure 18: Image of a Miniature Pt100 RTD sensor [2].

used in within a bioreactor and not affect the cells. The Pt100 RTD sensor has ±0.23°C accuracy at 40°C which allows for accurate temperature monitoring to be achieved.

An RTD temperature sensor works by changing their resistance depending of the temperature. For the Pt100 RTD, the resistance is 100 ohms when the temperature is at 0°C and a resistance of 119.40 ohms at 40°C. A complete temperature versus resistance table for the Pt100 RTD sensor can be found in Appendix E. In order for the temperature of the RTD sensor to be detected, the resistance of the sensor is converted to voltage using a voltage divider. The voltage divider in Figure 19 would create an output voltage related to the RTD by equation of $V_{out} = \frac{V_{CC}*RTD}{RTD+R}$, where V_{cc} is the input voltage and for our voltage divider R is 1 kilo-ohm [12].

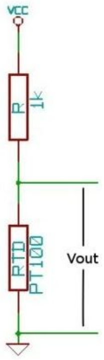


Figure 19: A circuit diagram for a voltage divider where Vcc is the input voltage and R is 1k [12].

For the voltages to be converted from voltages into temperatures, a microcontroller is necessary. The microcontroller will read in the voltage as an analog input and then using a program convert the input into temperature that can be outputted. The microcontroller we decided to use is the Arduino Uno, Figure 20. We selected a temperature range of 0°C - 100°C, or 0.4545 – 0.6083 volts, for the Arduino program to function for.

Figure 20: Image of an Arduino Uno microcontroller [1].

The Arduino Uno has an internal ADC reference voltage of 1.1 volts [1]. This means that 0 to 1.1 volts is divided by 1023 divisions; therefore, form 0 to 0.0978 volts, the Arduino represents this voltage as an analog input equal to 1. With our temperature range, the Arduino allows for an accuracy of 0.1°C if the max voltage matches the ADC reference voltage. The max voltage for our range of temperature is 0.6083 volts which does not match up to the internal ADC reference voltage. This creates a need for the voltage to be amplified to allow the two voltages to match up. A differential amplifier can be used to ramp up the voltage and also allows 0 volts to represent the minimum temperature of 0°C. This is done because a differential amplifier, Figure 21, has an output voltage given by the equation of $V_{out} = \frac{RB}{RA}(V_2 - V_1)$. V1 is set to be the minimum temperature voltage of 0.4545 volts by using a voltage divider with a 1KΩ and a 100Ω resistors and a 5 volt input. V2 is the output from the RTD voltage divider. In between the voltage dividers and differential amplifier, a voltage buffer is necessary to allow the voltage dividers and the differential amplifier to correctly perform the desired operation.

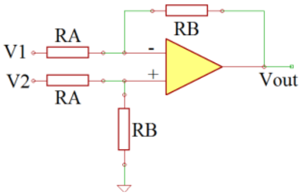


Figure 21: Circuit diagram of a differential amplifier [10].

The resistors in the differential amplifier were determined by solving for the necessary gain, RB/RA, to bring the max temperature voltage to approximately 1.1 volts. The voltage difference between the max and min temperature is 0.1537 volts; therefore, the gain is 7.155, 1.1 divided by 0.1537. We then selected to resistors that would approximately give us this value. RA and RB were selected to be 47kΩ and 330kΩ respectively. This values lead to a gain of 7.02 which allows for the voltage to be amplified to 1.0795 volts. The entire circuit diagram for the RTD sensor to Arduino interface can be seen in Appendix E. This circuit includes a low pass filter and a 220kΩ resistor to reduce noise.

An Arduino program was also created to read in an analog input, convert it to temperature and finally output that temperature. The program reads in the voltage which is converted to a number between 0-1023. This is converted back into a voltage by dividing the input by 1023 and multiplying this by 1.0795. Then the voltage can be converted to temperature by dividing the voltage by 1.0795 and multiplying it by 100. Simpler, the analog input can be converting into temperature by multiply the input by 100 and dividing by 1023. The code for this program can be seen in Appendix E.

Testing:

Time of Mixing

Qualitative testing was performed on the printed pitched blade impeller to assess its true mixing ability. The test consisted of viewing the impeller's mixing ability of a red dye at various RPM. The results obtained were compared to a spinner flask and a magnetic stir bar that underwent the same testing procedure. The stir bar was used as a control to compare with the two impellers.

To perform the mixing tests, 115 mL of water were measured out in a graduated cylinder and placed in the 100 mL bottle, or 125 mL corning spinner flask. The bottle or flask was then placed on a stir plate. The test was performed at 4 different speeds of 65, 95, 125 and 350 RPM. At each speed, three trials were run per mixing element type. When the impeller being tested reached the proper speed, a micropipette was used to inject 20 µL of dye into the fluid. A video of each trial was recorded to determine qualitative mixing time and to view how the dye was mixed throughout the vessel.

The results of the mixing testing can be seen in Table 4 or graphically in Figure 22. The data shows that the spinner flasks mixing time was the faster; however, this is not completely accurate or truly representative as the spinners flask's container was wider and included baffles that help to increase its mixing efficiency. Analysis of the videos for the custom pitched blade impeller and the spinner flask indicated that the custom impeller provided better axial flow.

Comparing the custom impeller to the stir bar control gave a better comparison for mixing times as they took place in the same vessel. The data proves that the custom impeller mixes the dye faster than the stir bar at the lower RPM. At the higher RPM, the stir bar and custom impellers mixing times start to approach similar times. The mixing times and analysis of the videos showed that the custom impeller provided better uniform mixing than the magnetic stir bar and greater axial flow than the spinner flask. This verified our selection of the pitched blade impeller to be implemented in the final design of the bioreactor.

Table 4: Collection of mixing time data at various RPMs for custom pitch blade impeller, spinner flask and stir bar.

Mixing Time vs. Impeller RPM

	100 mL bottle with Custom Impeller			125 mL Corning Spinner Flask			100 mL bottle with magnetic stir bar		
RPM	Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3	Trial 1	Trial 2	Trial 3
65	10.7	10.9	10.5	3.1	3.1	3.1	22.9	19.5	19.6
95	9.1	9.7	7.3	2.4	3.2	2.4	17.6	17.6	15.6
125	5.6	3.4	3.5	1.1	1.2	0.9	7.9	8.3	7.5
350	2.4	2	2.8	0.3	0.4	0.5	3.8	3.8	3.5

Note: Mixing times are reported in seconds

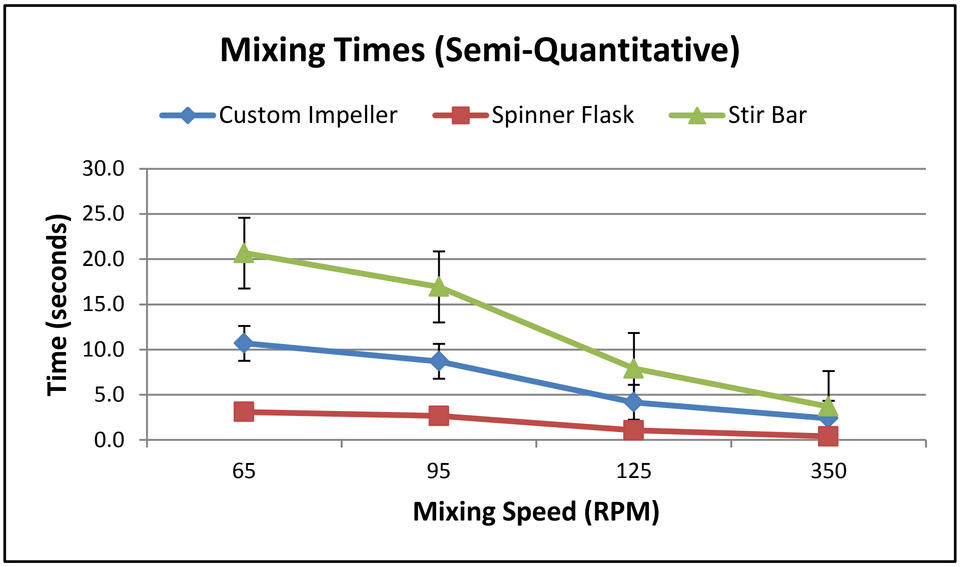


Figure 22: Plot of Mixing Time vs. RPM for the custom pitched blade impeller, spinner flask and stir bar.

Ethical Considerations:

There is a great deal of ethical controversy that surrounds stem cell research. The majority of these ethical concerns arise from the derivation of human embryonic stem cells. However, the bioreactor design described above is intended to culture induced pluripotent stem cells, which are derived in a vastly different manner then embryonic stem cells. Instead of extraction from an embryo, induced pluripotent stem cells are derived from adult cells. Once the adult cells are obtained from a human donor, reprogramming factors are introduce, which reverts them to their pluripotent state. Since no embryos are destroyed in this process, there should be little to no ethical concerns about the bioreactor's use with stem cells.

There may also be ethical and safety concerns related to the mass production of patient-specific cells for tissue engineered therapies. Our bioreactor is intended to culture these patient-specific cells en mass, but utilizing the resulting cells for medical therapies is still a long way off. A possible concern as patient-specific, tissue-engineered therapies emerge is the increased need for human clinical trials. Initial clinical trials may cause more harm to the patients than help; thus, performing a thorough risk-benefit analysis prior to enrollment of the patient into clinical trials will be necessary. The patients that participate in these initial trials should be suffering from preexisting ailments and have no other alternatives. Further, the idea of risk-benefit analysis will need to be extended into clinical practice once tissue engineered therapies have been approved. This will again prevent the imposition of more harm than help.

Our device doesn't infringe on the intellectual property of the existing bioreactors discussed in the "Existing Devices" section due its fundamentally different design. First and foremost, our bioreactor is designed for use within an incubator, whereas full bioreactor designs, such as those from Eppendorf, come equipped with all of the associated hardware. Second, our impeller design was tailored to minimize shear stresses on the cells; this distinguishing factor of our design relates to the intended use of our bioreactor, which is the final factor that separates this from the competition. This bioreactor is designed specifically for the reprogramming and differentiation of human induced pluripotent stem cells, which are sensitive to shear stresses in suspension culture. No bioreactor exists on the market for this purpose, and as our design is iteratively improved through cell culture testing, the final design will diverge further from currently available bioreactors.

The bioreactor as currently designed should not present a significant safety risk given the following considerations. The motor and impeller components must be secure and designed in such a manner to minimize risk of causing injury to the bioreactor operator, such as having a simple means to stop the impeller. The contents of the bioreactor are heated by the incubator air rather than the bioreactor itself, which should eliminate risk of burns or other heating element related injuries. In the future, if the bioreactor is redesigned to become a closed system, over-pressurization would be a potential hazard to guard against. However, since the bioreactor is currently designed to be an open system, there is little to no risk of gas build up or dangerous pressure changes for our design.

Future Work:

Several components will be added to the bioreactor that each introduce several new variables to consider and will require modifications to the existing design. Therefore, it was deemed appropriate that the system be designed component by component to avoid introducing several new variables to test simultaneously. Introducing each component individually would enhance the ability to trace the origins of any unforeseen issues.

The client did not set a specific budget limit. Budget was therefore determined on a component-by-component basis to determine whether the item's value to the project justified the cost. Items already purchased are outlined in Table 5.

Table 5: Project Purchases

Part Category	Specific Part	Cost
Vessel	GL 45 100 mL glass bottles (case of 10)	$33.32
Vessel	Corning Spinner Flask	$365.51
Temperature	Arduino Microcontroller	$22.95
Temperature	Auber PT 100MN Temperature probe	$28.60
Impeller	Magnetic Stir Bar (2 bars)	$13.24
Impeller	3D Accura60 Printed Impeller	$74.00
Circuit Components	LM741 Op-Amp (x3)	$3.87
Circuit Components	100Ω, 1K (x2), 47K (x2), 330K (x2), 220K Resistors	$1.91
Circuit Components	100 µF capacitors	$1.39
Total		$544.79

Future purchases include new impeller material to be milled, currently estimated at $40. A DC motor also will be purchased likely for about $150. The team may choose to purchase another vessel with a wide cap or a shape more conducive to monitoring by probes. Cell culture media will likely be the most significant addition to project cost with rough estimates detailed in Appendix G.

A major component yet to be designed for the bioreactor is the media exchange system. Future work includes the design, construction, and testing of this component.

Impeller Manufacturing

The final impeller design must be constructed from a material which can be used multiple times as well as non-cytotoxic. We discovered that the SLA epoxy material, Accura60, was cytotoxic, even after coating with polystyrene and polyethylene glycol. Because this material will not be compatible with our system, we will choose another material to manufacture the final impeller design. One option would be to utilize an automated 5-axis mill to produce our design. This type of mill gives greater flexibility in creating complex designs by allowing the mill tooling to be moved in more than 3 axes. This also would allow the impeller to be manufactured in a more robust material, such as stainless steel, which would also be non-cytotoxic. If manufactured from stainless steel, the device would be able to be used in an autoclave for sterilization.

Alternatively, the impeller could be manufactured using an investment casting process (Figure 23). This involves creating a sacrificial model of the impeller out of wax. Next, the wax model coated by a refractory material and surrounded by coarse ceramic particle slurry until a desired thickness is achieved. The wax model is then burned off and a mold of the model remains. Finally, a material, such as steel, is poured into the mold and allowed to cure. After the material has solidified, the mold is broken off and the final part is retrieved. To use this method, we would need to find a material that is suitable for casting as well as non-cytotoxic. This method requires more materials and time compared with using an automated mill, so we plan to pursue using the automated 5-axis mill instead of casting our impeller.

Media Transfer

A bioreactor can perform cell cultures by either using a batch method or having a continuous cultivation. The batch method is when media is brought to the appropriate cell culture conditions and the cells will go through expansion and then the cells will be passaged and placed in a new batch for reprogramming. Once the reprogramming batch has finished, the cells are passaged again and put into another batch for differentiation. Batch cultures have a limited amount of nutrients and are able to only produce a set amount of product. After the max amount of cells is reached, the cells are removed and the bioreactor must undergo sterilization before the vessel can be used again. Due to multiple passagings and batches being required, batch cultures require a need for continually skilled labor. This labor causes batch cultures to cost more to produce the cells.

On the contrary, continuous cultures have media flow in and out of the bioreactor to allow for unlimited amounts of nutrients while maintaining cell culture conditions. Continuous cultures can produce large amounts of cells while keeping labor cost low because the process can be automated. However, it requires a design that allows for an inflow and outflow of media without damaging or losing large amounts of cells.

We need to determine if we are going to use a batch or continuous culture method. Currently the final design is a batch method but could be changed to a continuous culture by designing an automated process for media exchange. A possible way to do this would be by designing the bioreactor vessel to have an inlet and outlet to allow media to flow in and out at a controlled rate. The outlet would require a way to filter out the wastes while keeping the cells within the bioreactor.

Vessel

Currently we are planning to use 100 mL glass bottles as out bioreactor vessel. However, there are several constraints that would require the vessel to be redesigned. As stated above, the final vessel design is affected by the type of culture is chosen and how the media will be transferred. The vessel could be designed with an inlet and outlet for continuous culture. Also, the vessel has several other design options. It can to be designed to be with or without baffles along with being disposable or reusable. These are design options that we will need to determine.

The vessel needs to also be designed to hold and keep the probes that will monitor and insure the cell culture conditions are correct out of the way of the impeller. These probes consist of the designed temperature probe along with a pH probe to monitor dissolved CO_2 and a dissolved oxygen probe. To do this, a cap could be designed that would contain holes to hold the probes above the impeller blades but still under the media. A second option would be to have the probes enter on the side of the vessel and hold the probes to side of the impeller blades.

The impeller must be held at the correct height within the vessel and prevent the impeller from wobbling during rotation. A cap could be designed to house the impeller shaft within a cylinder to allow for rotation without it moving from side to side. This cylinder could also be designed to allow for the impeller shaft to be connected to an incubator safe motor that would rotate the impeller at the proper RPM.

Lastly, the cap and vessel must be designed to allow for proper gas diffusion into the media. This could be done several different ways and we will have to develop one. One design idea would be to design a cap that has a filter to allow the gas into the bioreactor where it could then diffuse into the media. Calculations and tests would need to be performed to insure that proper mass transfer of the gases occur to produce the proper cell culture conditions.

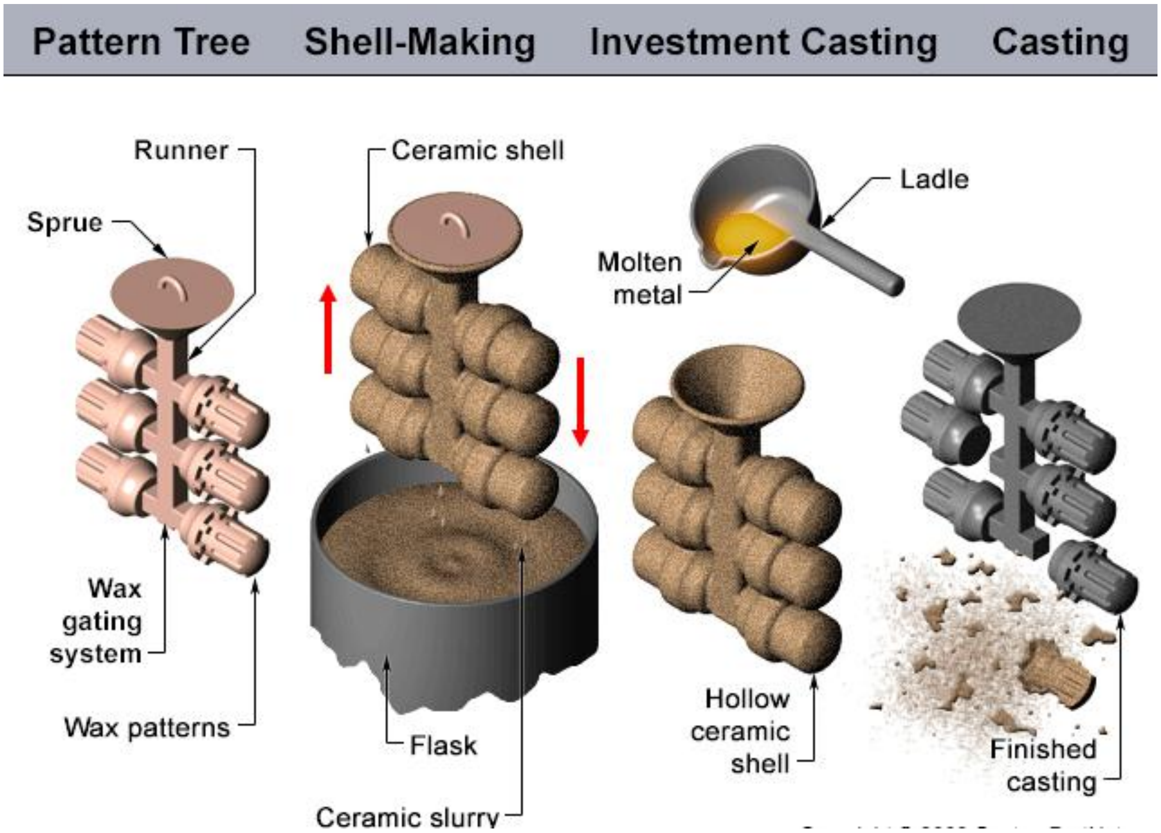


Figure 23: Investment Casting Process [7]

Testing With Cells

Once the bioreactor is assembled, several cell experiments will be conducted in order to determine the success of our design. The first experiment will test the survival and expansion of 3T3 cells in suspension culture. The 3T3 cell line was originally established from mouse embryonic fibroblast cells and will be used for the initial testing of the bioreactor since it has become a standard fibroblast cell line that is easily obtained. The goal of this first experiment will be to determine an optimal range of revolutions per minute velocities at which 3T3 cells survive and expand in suspension culture.

This experiment will be performed inside a cell culture incubator and will be carried out for eight days. Cells will initially be cultured under adherent conditions, and on day 0 they will be dissociated with accutase and seeded at a density of 50,000 cells per milliliter of media. We will take samples of media (2.0 ml) from the bioreactors every other day from day 2 to 8 and cell aggregates will be dissociated with 0.25% trypsin-EDTA. We will then proceed to calculate the total cell numbers and viabilities using a hemocytometer combined with trypan blue staining. The data will be analyzed at the end of the experiment to determine the expansion of cells in comparison to a spinner flask which will serve as a positive control throughout the experiment. The experiment will be done three times in order to test three different velocities.

Based on literature review and fluid modeling analysis, the first velocity that will be tested is 100 revolutions per minute, and depending on the results obtained from this experiment, two other velocities will be determined and tested. Statistical analysis will be conducted to conclude the optimal velocity for the survival and expansion of 3T3 cells in suspension culture. We will then proceed to test the optimal velocity with the secondary MEFs that will be used for reprogramming.

After showing that cells can proliferate in suspension culture, we will start to reprogram mouse embryonic fibroblast (MEFs) to induced pluripotent stem cells (iPSCs) in the bioreactor. Secondary mouse embryonic fibroblasts that have already been encoded with the four transcription factors - Oct 4, Sox 2, c-Myc and Klf4- will be used for this experiment. According to Fluri et al. (2012), the suspension reprogramming process can be done either in serum containing mouse embryonic stem cell medium or in serum-free medium. The mouse ESC medium consists of DMEM supplemented with 15% (v/v) FBS (Wisent), 0.1 mM β-mercaptoethanol (BME, Sigma), 1 mM sodium pyruvate, 0.1 mM MEM non-essential amino acid (NEAA, Gibco), 2 mM Glutamax, 1% (v/v) penicillin-streptomycin and 1,000 U ml−1 of leukemia inhibitory factor (LIF) (Millipore). The serum-free ESC medium consists of DMEM with F12 (Gibco) and Neurobasal (Gibco)-based medium supplemented with N2 (Gibco), B27 (Gibco), 0.05% (w/v) BSA, 2 mM Glutamax, 1% (v/v) penicillin and streptomycin, 1.5 × 10−4 M monothioglycerol, 1000 U ml−1 LIF, and 10 ng ml−1 BMP4.

Secondary MEFs in adherent culture will be induced with 1 microgram per milliliter of doxycycline and after 8 hours the cells will be trypsinized and seeded into the bioreactor. One third of the medium culture will be replaced every day and to prevent possible cell differentiation because of aggregates size. All the cells will be passed through 100 micrometer cell strainers. We will also collect samples of suspension-cultured cells undergoing reprogramming at different time points after doxycycline induction. These cells will be analyzed for pluripotency markers using flow cytometry and/or immunocytochemistry in order to determine when to stop the reprogramming process. If our bioreactor is successful at reprogramming secondary mouse embryonic fibroblasts, we will proceed to sort the induced pluripotent stem cells and test neural differentiation protocols on the iPSCs.

References:

1. Arduino. *Language Reference*. Arduino. <arduino.cc/en/Reference/HomePage>
2. Auber Instruments Incorporated. *Miniature RTD Sensor.* Auber Instruments. http://www.auberins.com/index.php?main_page=product_info&cPath=20_15&products_id=38
3. Bird, R. Byron, Warren E. Stewart, and Edwin N. Lightfoot. *Transport Phenomena*. New York: J. Wiley, 2007.
4. Corning Incorporated. 125mL Proculture® Glass Spinner Flask with 70mm Flat center Cap and 2 Angled Sidearms. *Corning Life Sciences*. < http://catalog2.corning.com/Lifesciences/en-US/Shopping/index.aspx>
5. Fluri et al. Derivation, expansion and differentiation of induced pluripotent stem cells in continuous suspension cultures. Nature Methods. January 2012.
6. Goldthwaite, Charles A., Jr., Ph.D. "The Promise of Induced Pluripotent Stem Cells (iPSCs)." *Stem Cell Information*. National Institute of Health, Dec. 2011. Web. Oct. 2012. <http://stemcells.nih.gov/StemCells/Templates/StemCellContentPage.aspx?NRMODE=Published>.
7. Investment Casting. (n.d.). *Manufacturing Cost Estimation*. Retrieved December 11, 2012, from http://www.custompartnet.com/wu/investment-casting
8. Joshi, Jyeshtharaj B., Kumaresan, T. Effect of impeller design on the flow pattern and mixing in stirred tanks. Chemical Engineering Journal. ELSEVIER. April 2005.
9. "Kapton Tapes." *Kapton Tapes*. DuPont, 2012. http://www.kaptontape.com
10. Kehoe, Daniel E., Jing, Donghui, Lock, Lye T., and Tzanakakis, Emmanuel S. Scalable Stirred-Suspension Bioreactor Culture of Human Pluripotent Stem Cells. *Tissue Engineering*. Vol. 16-2. 2010. <http://www.ncbi.nlm.nih.gov/pubmed/19739936>
11. Kumaresan, T., Jyeshtgaraj, B.J. Effect of Impeller Design on the Flow Pattern and Mixing in Stirred Tanks. *Chemical Engineering Journal.* Vol. 115-3. 15 January 2006. <http://www.sciencedirect.com/science/article/pii/51385894705003645>.
12. Lea, Trystan. *4X Multiplexed RTD Temperature Sensor Module.* OpenEnergyMonitor. Nov 30, 2010. <openenergymonitor.org/emon/node/75>
13. Mirro, Rich, and Kevin Voll. Which Impeller Is Right for Your Cell Line?. *BioProcess International*. Jan. 2009. Accessed 18 Sept. 2012. <http://www.bioprocessintl.com/journal/2009/January/Which-Impeller-Is-Right-for-Your-Cell-Line-183538>.
14. Murry CE, Reinecke H, Pabon LM. 2006. Regeneration gaps: observations on stem cells and cardiac repair. J. Am. Coll. Cardiol. 47:1777–85
15. New Brunswick. Cell Culture Systems. *New Brunswick an Eppendorf company.*2007. <http://www.nbsc.com/cellculturebioreactors.aspx>
16. Rodrigues et al. Stem Cell Cultivation in Bioreactors. *Biotechnology Advances*. ELSEVIER. June 2011. <berg.ist.utl.pt/scbl/BADV_Rodrigues.2011.pdf>
17. Shafa et al. Derivation of iPSCs in stirred suspension bioreactors. Nature Methods. March 2012.
18. "Stemolecule Doxycycline Hyclate." *Miltenyi Biotec*. Miltenyi Biotec, Web. 22 Oct. 2012. <http://www.miltenyibiotec.com/en/PG_890_1224_Stemolecule_trade_Doxycycline_hyclate.aspx>.
19. "Wire: Nichrome 80 (tm) & Other Resistance Alloys - Tech Data." *Wire: Nichrome 80 (tm) & Other Resistance Alloys - Tech Data*. WireTronic, 2012. < http://www.wiretron.com/nicrdat.html>
20. Youn BS, Sen A, Kallos MS, Behie LA, Girgis-Gabardo A, Kurpios N, Barcelon M, Hassell JA: Large-scale expansion of mammary epithelial stem cell aggregates in suspension bioreactors. Biotechnol Prog 2005, 21:984-993.

Appendix A: Product Design Specifications

Neural Bioreactor

Team Members: Maria Estevez, Jeff Groskopf, Tyler Klann, Lisa Kohli, Ian Linsmeier

Date: 12/12/12

Function: Reprogramming adult cells into induced pluripotent stem cells (iPSCs), as well as their subsequent expansion and differentiation, is normally completed in adherent cell cultures. Recently, it has been proven that iPSCs can be derived and expanded in suspension cultures using a stirred suspension bioreactor. These reactors establish stable cell culture conditions by controlling the temperature as well as the level of nutrients (media), CO_2 (pH), O_2, and other soluble factors. The suspension components are uniformly distributed within the reactor fluid through various mixing techniques, most commonly an impeller. The process of adult cell reprogramming and iPSC expansion and differentiation can be scaled up and automated using bioreactor stirred suspension cultures.

Dr. Saha has asked our team to design a bioreactor that maximizes the production of neural progenitor cells from mouse embryonic fibroblasts in stirred suspension cultures. The project involves designing culture processes and optimizing culture conditions to reprogram adult cells to induced pluripotent stem cells (iPSCs) and differentiate those iPSCs to neural progenitors.

Client Requirements

- Stirred suspension culture
- Use mouse embryonic fibroblasts (MEFs)
- Reprogram MEFs into iPSCs
- Culture Environment: 37° C , 5% CO_2

Design Requirements

1) **Physical and Operational Characteristics**
    a. ***Performance Requirements:*** The bioreactor must be able maintain 37° C and 5% CO_2 for multiple weeks at a time. Most components of the bioreactor will be reusable. The bioreactor must provide an environment conducive to cell culturing and reprogramming.
    b. ***Safety:*** The bioreactor will incorporate a heating element that will heat the culture to 37° C, but a malfunction in the heating regulation system could lead to much higher temperatures that could damage the cells, microscope, or even the lab technician.
    c. ***Accuracy and Reliability:*** The bioreactor must maintain an internal temperature of 37±1°C and a CO_2 concentration of 5±.5%. The bioreactor must allow for accurate and reproducible conditions.

d. ***Life in Service:*** The bioreactor will be autoclavable. It would be autoclaved after one use or iteration of reprogramming secondary MEFs into iPSCs, and culturing them to their desired states.

e. ***Operating Environment:*** The device will be used in a cell culture hood by a skilled lab researcher or technician. The device will only be exposed to the lab environment, which will be well controlled.

f. ***Ergonomics:*** The bioreactor must be simple to use.

g. ***Size:*** The bioreactor will be a stand-alone unit, meaning it will operate independently from other equipment within the lab environment. It will use a 100 mL vessel volume to contain the cell culture. All of the components of the bioreactor should be able to be transported from building to building.

h. ***Weight:*** The weight should be light enough so that one person can lift the bioreactor.

i. ***Materials:*** The materials on the inside of the bioreactor must not be cytotoxic. The bioreactor vessel and elements exposed to the cell culture must be cytophobic or coated in a material that prevents cell adhesion. The material must be impermeable to small molecules and gas to create a closed system.

2) Production Characteristics

a. ***Quantity:*** 1

b. ***Target Product Cost:*** Indeterminate

3) Miscellaneous

a. ***Customer:*** The customer would like this to be eventually used for human adult cells to be reprogrammed into iPSCs. However, we are initially going to design the bioreactor to use MEFs due to the existing protocols being readily available in literature.

b. ***Competition:*** Bioreactors exist in the market for specific applications, but the commercially available bioreactors are not tailored to the specific needs to reprogram cells into iPSCs or for later differentiation of those cells.

Appendix B: Preliminary & Final Design Dimensions

Preliminary Impeller Designs

<u>Pitch Blade Impeller Dimensions</u>

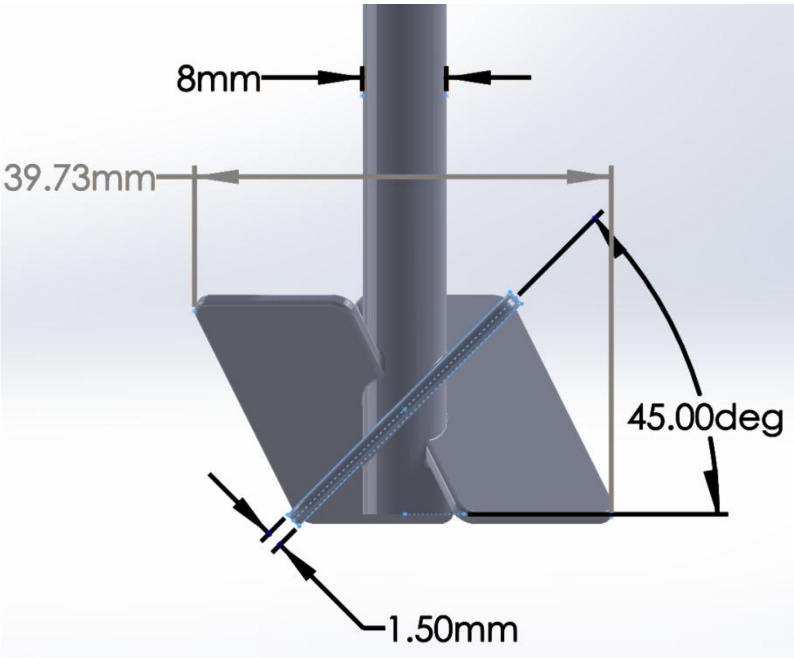


Figure B1: Preliminary Pitch Blade Impeller Dimensions: Front View.

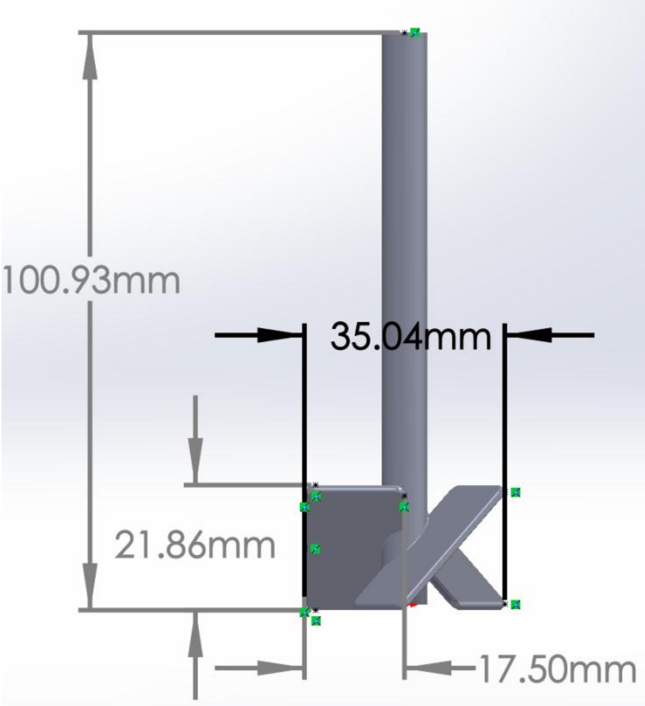


Figure B2: Preliminary Pitch Blade Impeller Dimensions: Side View

Whisk Impeller

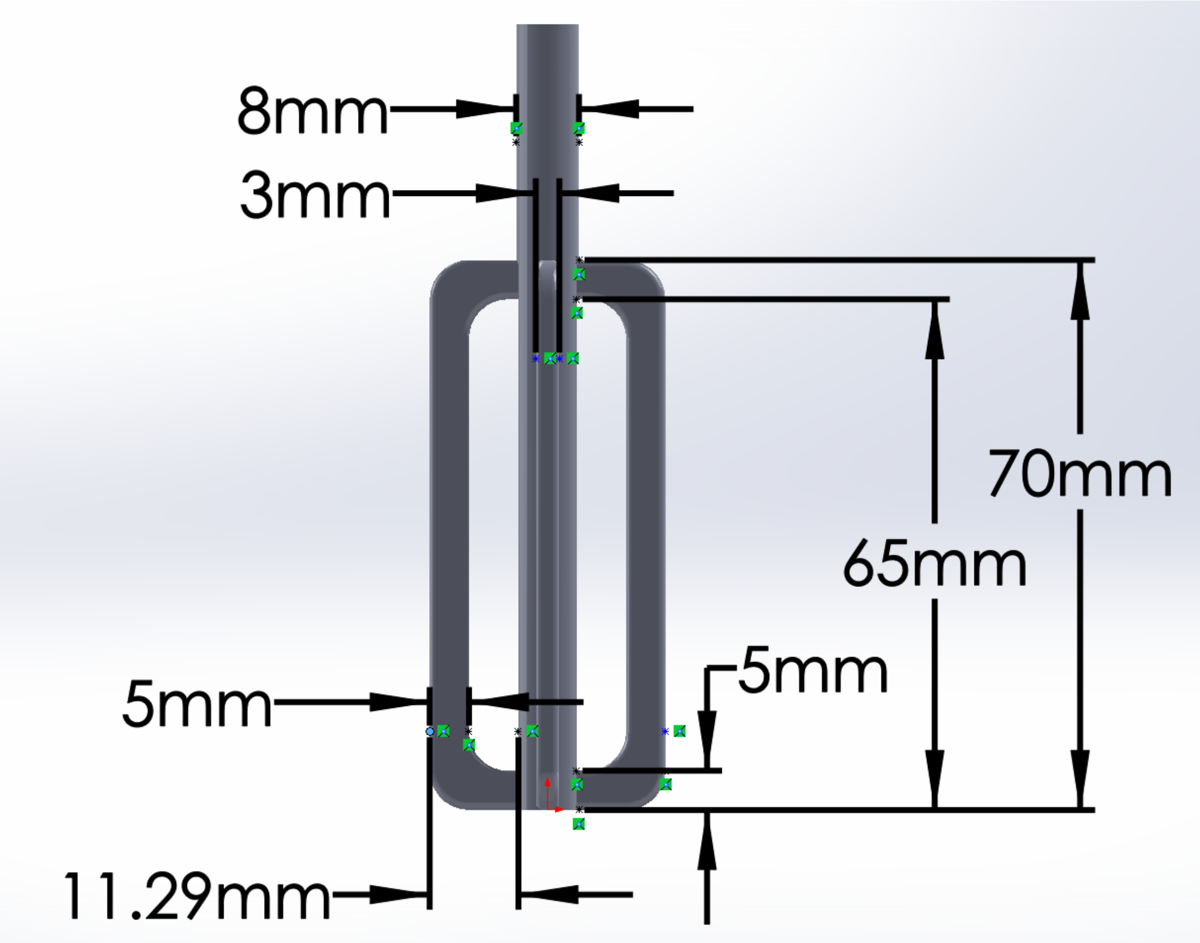


Figure B3: Whisk Impeller Dimensions: Front View.

Flat Blade Impeller

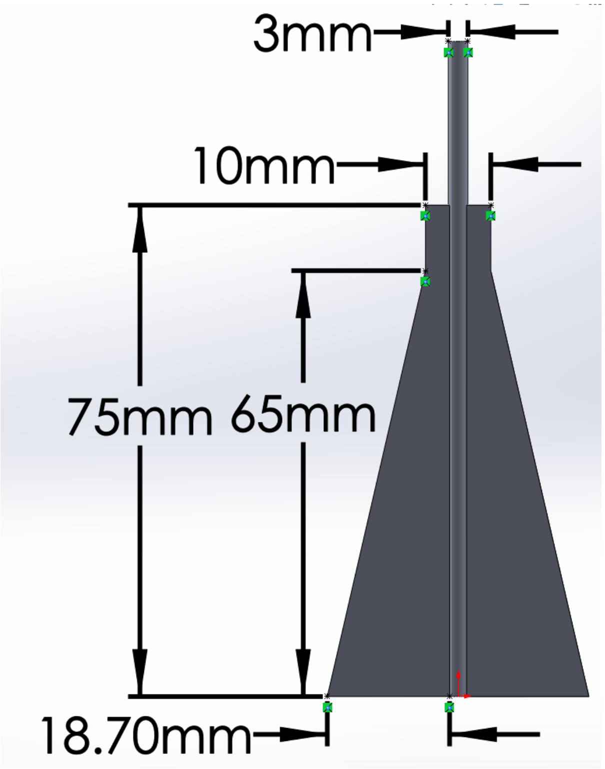


Figure B4: Flat Blade Impeller Dimensions: Front View.

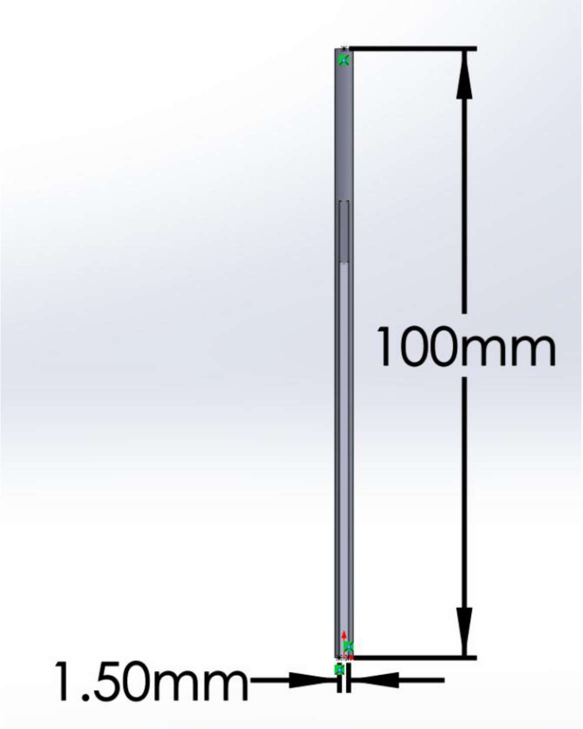


Figure B5: Flat Blade Impeller Dimensions: Side View.

Final Impeller Design: Pitch Blade

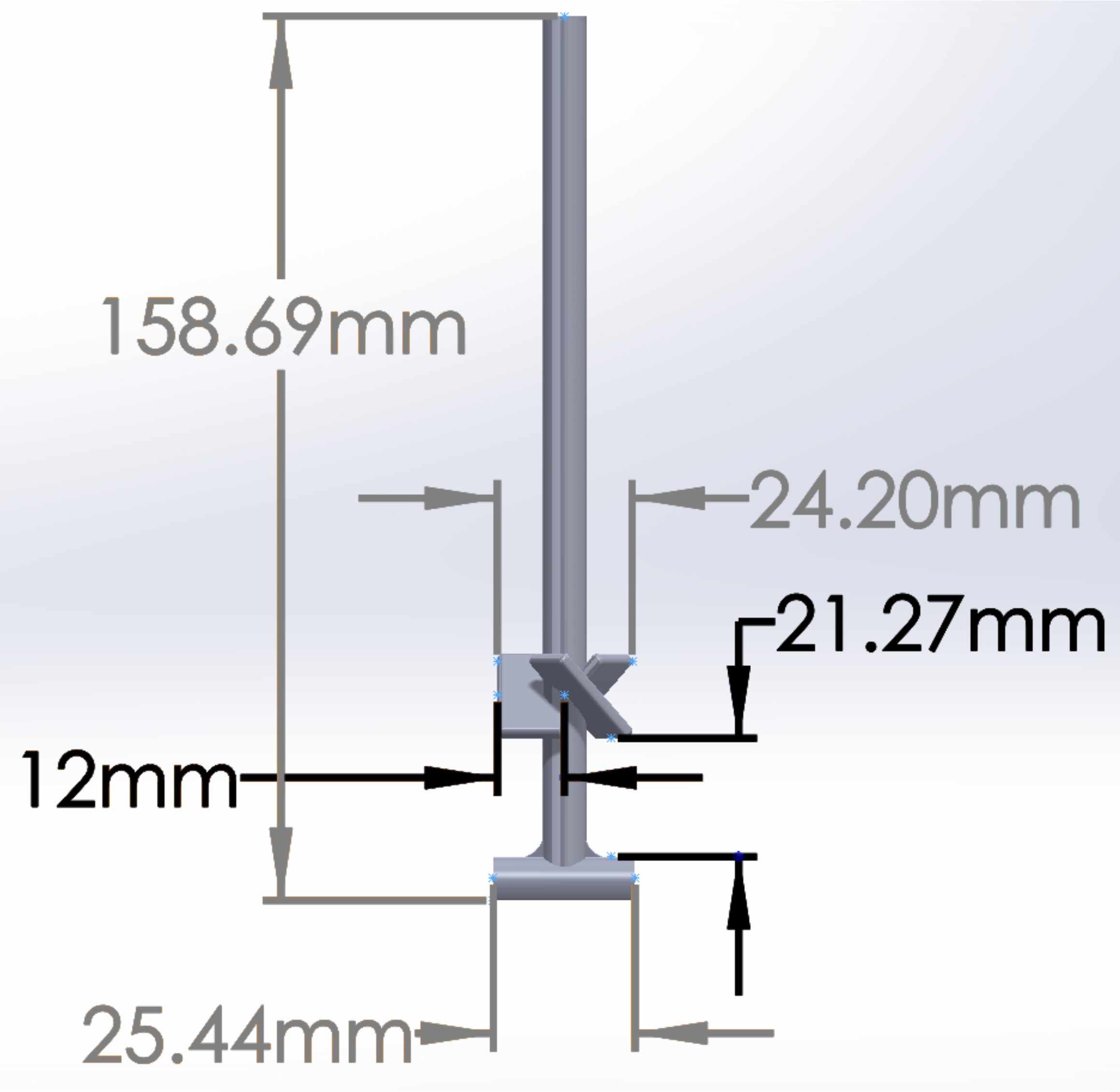


Figure B6: Final Pitch Blade Impeller Dimensions: Overall Side View.

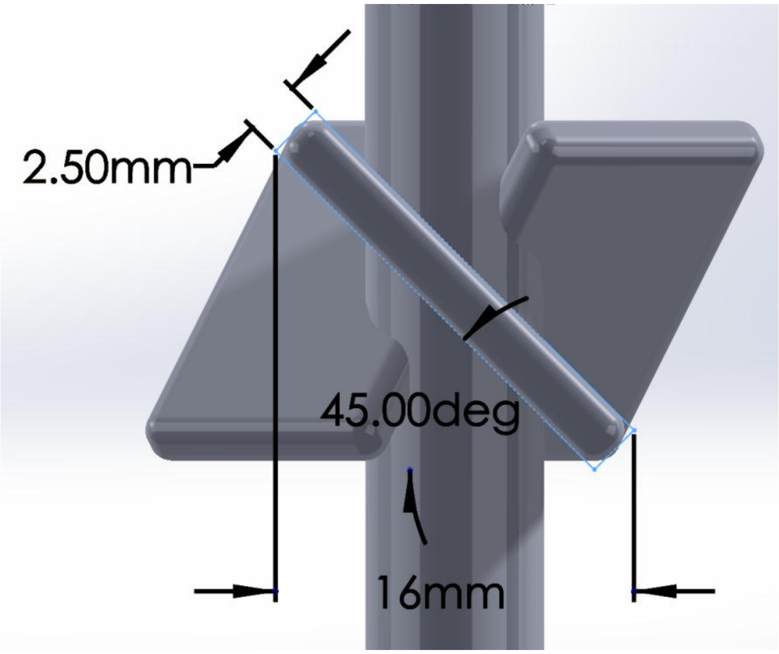


Figure B7: Final Pitch Blade Impeller Dimensions: Front view of impeller blades.

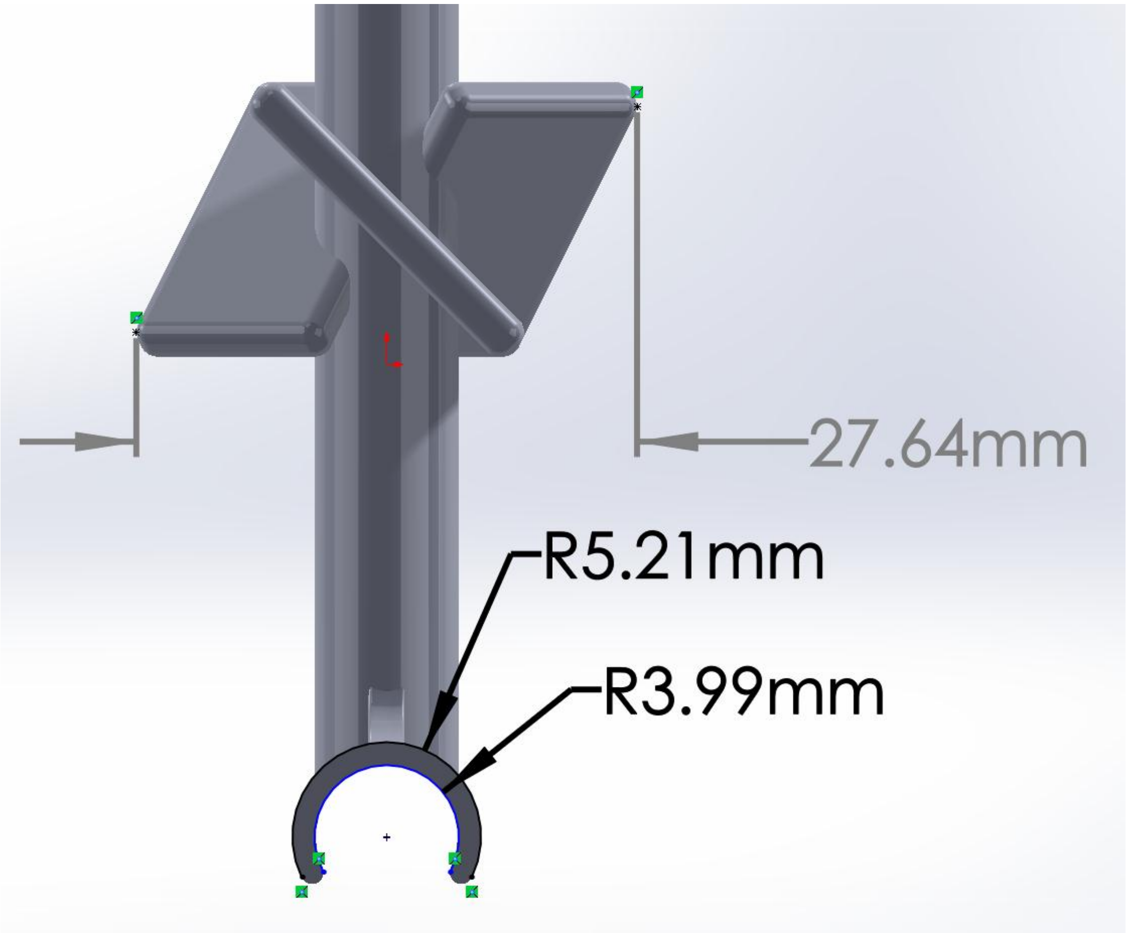


Figure B8: Final Pitch Blade Impeller Dimensions: Front view of impeller blades and magnetic stir bar clasp.

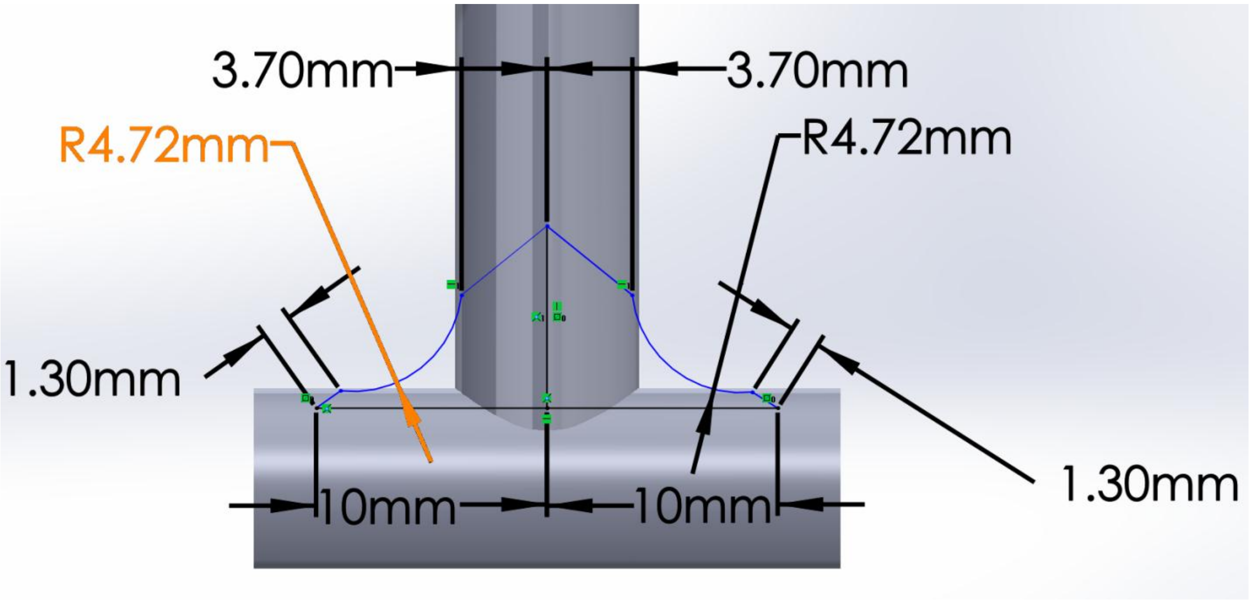


Figure B9: Final Pitch Blade Impeller Dimensions: Magnetic stir bar clasp side view. The blue outline depicts the support structures that connect the magnetic stir bar clasp to the central rod of the impeller.

Appendix C: Fluid Modeling

Each impeller design was created in SolidWorks based off of the measured dimensional contraints of a 100mL beaker, which will act as the bioreactor vessel – the 100mL beaker has a height of 80mm and a inner diameter of 50mm. After creating an impeller to the desired dimensions, additional geometries were added to facilitate fluid modeling.

First the blades of the impeller must be encompassed within a region that will be rotated (Figure C1). A geometry that fully surrounds the blades was created, and at least 1mm of room was allowed between the edge of the blades and the faces of the constructed geometry. The resulting 3D geometry to be rotated was not merged with the impeller; this was accomplished by unchecking the "Merge Result" box when extruding the 2D sketch into 3D.

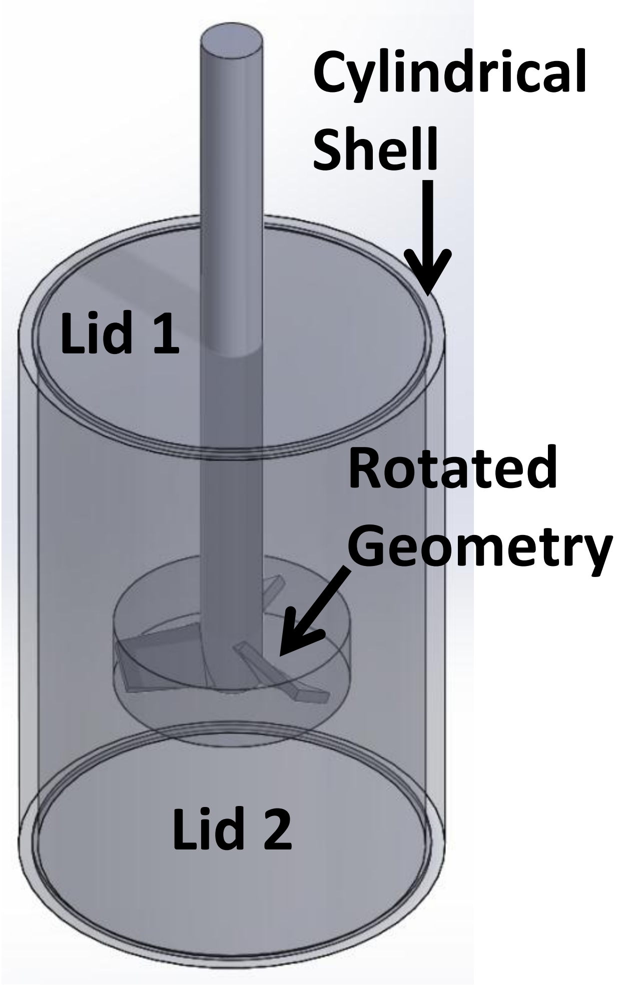


Figure C3: Fluid Model Configuration. Labeled above is the geometry to be rotated surrounding the impeller blades as well as the cylindrical shell that constitutes the vessel boundary. Additionally, the lids that create the closed systems are shown. Lid 1 is the top lid of the vessel, and Lid 2 is the bottom lid; both are required to create a closed system.

After creating the geometry to be rotated, the bioreactor vessel was created. The first step in creating a closed bioreactor vessel is the construction of a cylindrical shell (Figure C1). Two concentric circles were drawn on a sketch plane perpendicular to the central axis of the impeller. The inner circle diameter was created to be equivalent to the inner diameter of the vessel. For the bioreactor fluid modeling, an inner diameter of 50mm was used. The outer circle diameter is arbitrary since it will have no effect on the fluid model. Extrude the annulus, such that the impeller is positioned at the desired location within the vessel. For the preliminary fluid modeling, the impeller was located at one third of the total vessel height from the bottom of the vessel – for a vessel height of 80mm, the bottom of the impeller was positioned 26.67mm above the bottom of the vessel. To achieve this, a sketch was created on the bottom face of the central impeller rod, which is level with the bottom edges of the impeller blades. The two concentric circles were then drawn on this sketch. The 2D sketch was extruded upwards 53.33mm and downwards 26.67mm – this positions the impeller 26.67mm above the vessel bottom, or one third of the total height. Again, the resulting cylindrical geometry wasn't merged with the impeller when extruding the 2D sketch into 3D.

Once these supplemental geometries were created, the fluid model could be defined. First, the SolidWorks Flow Simulation Add-On was initialized by going to Tools → Add-Ins and selecting the checkbox next to “SolidWorks Flow Simulation.”

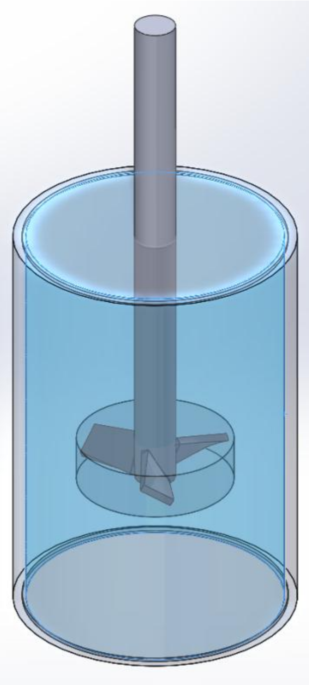


Figure C4: Fluid Subdomain. Depicted in blue is the fluid subdomain. This fluid volume is defined by selecting the inner face of the cylindrical shell.

The simulation wizard was selected to start a new fluid model. Within the wizard, one can define each of the global settings, step-by-step. The recommended settings are “Internal” for “Analysis Type,” with the options “Exclude cavities without fluid flow” and “Exclude Internal Space” unselected. The “Time Dependent,” “Gravity,” and “Rotation” options were selected for the physical features of the fluid model. Under the fluids setting tab, any fluid that will be simulated can be added into the model; water was used as the fluid for the neural bioreactor simulations. The remaining settings were left at their defualt values and selections.

Next a closed vessel was defined by creating lids on both the top and bottom faces of the cylindircal shell (Figure C1). Clicking the “Lids” feature in the Flow Simulation upper tab and then selecting both the top and bottom face of the cylindrical shell automatically creates two lids and a closed vessel space. The closed vessel space allows the fluid domain to be defined.

Next, a fluid subdomain (Figure C2) was inserted into the bioreactor vessel. The inner face of the cylindrical shell was selected to define the fluid space within the vessel. The fluid subdomain should match the desired fluid for the simulation. If the aforementioned steps have been followed correctly, the fluid domain should correctly initialize; any deviation from the prior steps may disallow the user from creating a fluid subdomain. It may be necessary to hide certain features from view or change their transparency to be able to select the inner face of the bioreactor vessel. Also, the SolidWorks Fluid Simulation software cannot model the liquid-gas interface; therefore, the entire vessel space must be filled with liquid.

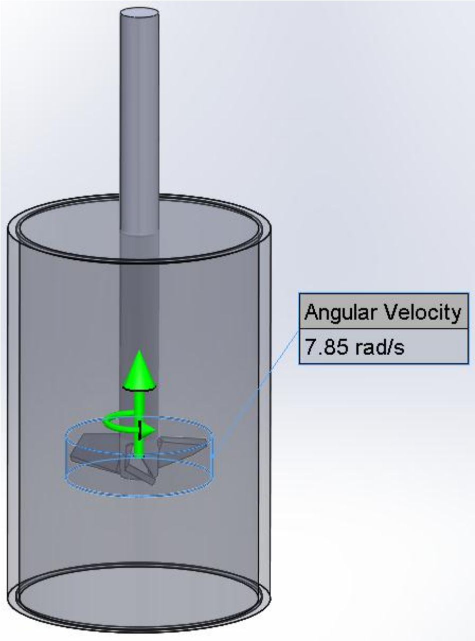


Figure C5: Rotated Geometry. Selecting the geometry surrounding the impeller blades and assigning a rotational speed (rad/s) will simulate rotation within this region during the fluid simulation. The curved arrow denotes the rotational direction; this can be reversed by introducing a negative sign in front of the rotational speed.

Following the insertion of a fluid domain, the rotation region was defined (Figure C3). Selecting the Rotation option in the Physical Features tab of the fluid simulation wizard will initialize the rotation region option. This feature was selected and applied to the region encompassing the impeller blades. An arrow will appear denoting the rotational direction for the geometry. The rotational speed was entered in radians/second and the direction of rotation was determined by including or excluding a negative sign. The

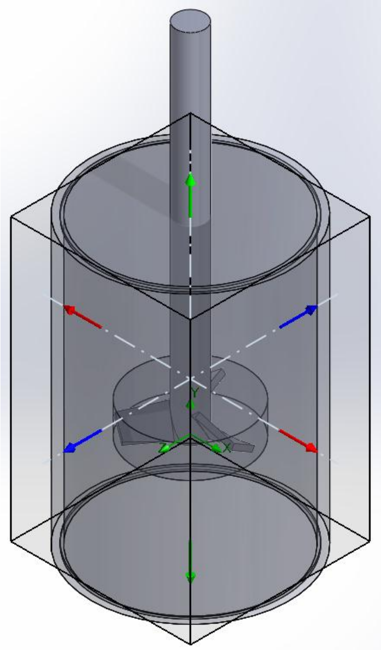


Figure C6: Computational Domain. The wireframe rectangle defines the computational domain. The fluid simulation software should correctly define this volume to the inner diameter of the cylindrical vessel, as shown above.

inclusion of a negative sign will switch the direction of rotation that was initially defined by the software.

The last step before running the simulation is defining the global computational domain (Figure C4). The software should have correcty defined the computational domain to the inner faces of the cylindrical vessel; however, if this is not the case, the computational domain can be adjusted to include only the vessel space containing the fluid. It should be noted that the computational domain is a rectangular region and will include some space beyond the cylindrical geometry.

Lastly, the fluid simulation was run, causing the solver to create its mesh and calculate the results. After the solver finishes, a variety of fluid parameters can be plotted over the model. For the bioreactor simulations velocity profiles were plotted for each of the impellers (Figure C5).

The aforementioned steps cover the minimum requirements to model a rotating impeller within a fluid space. The model can have additional constraints or initial conditions added to it by selecting the appropriate features or adjusting the settings.

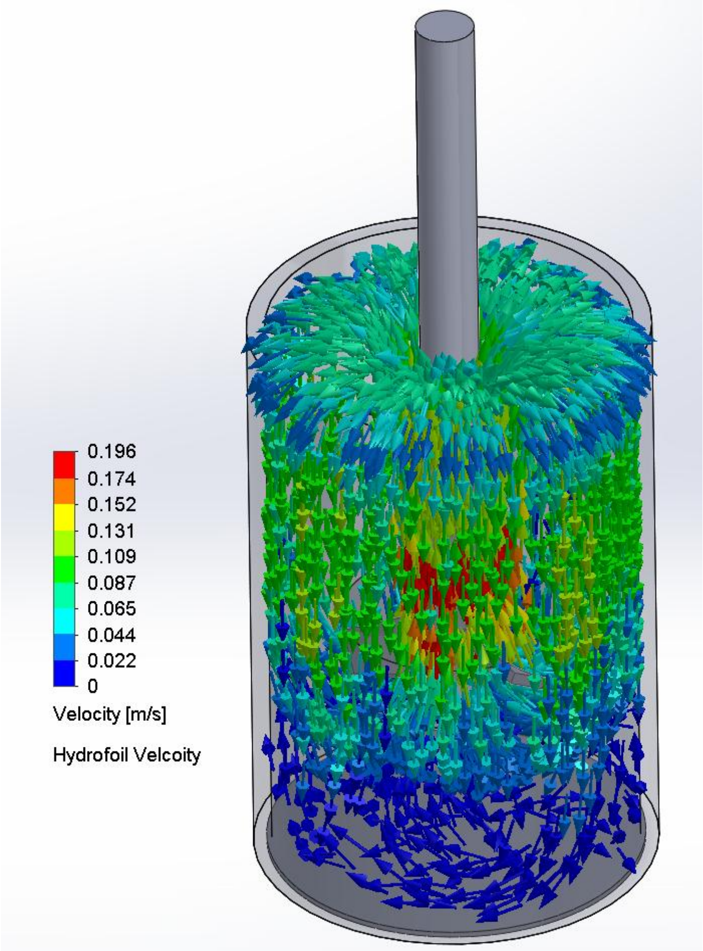


Figure C7: Resultant Velocity Profile. After running the fluid model simulation, flow trajectories can be defined on the inner surface of the cylindrical shell, similar to the definition of the fluid subdomain. One such flow trajectory is the fluid velocity profile, shown above.

Appendix D: MATLAB Code

Shear Stress Calculations

Program imports data files, then uses speed and position information to calculate x and y direction shear stress at each position.

Consists of main program that calls 7 functions: ShearStressCalculation.m, decideAction.m, analyzeFrontPlaneWithXVelocity.m, analyzeFrontPlaneWithYVelocity.m, analyzeFrontPlaneWithZVelocity.m, analyzeTopPlaneWithXVelocity.m, analyzeTopPlaneWithYVelocity.m, and analyzeTopPlaneWithZVelocity.m

```
close all
clear all
% Find the folder
folder = uigetdir;
% Get the names of all files. dirListing is a struct array.
dirListing = dir(folder);
dirListingLength = length(dirListing); % Number of files in folder
for newFileIteration = 1:dirListingLength
    fileName = dirListing(newFileIteration).name;
    action = decideAction(fileName);
    if action == 1
        analyzeFrontPlaneWithXVelocity(fileName)
        analyzeFrontPlaneWithYVelocity(fileName)
        analyzeFrontPlaneWithZVelocity(fileName)
    elseif action == 2
        analyzeTopPlaneWithXVelocity(fileName)
        analyzeTopPlaneWithYVelocity(fileName)
        analyzeTopPlaneWithZVelocity(fileName)
    end
    clear fileName;
end
```

```
function [ action ] = decideAction( fileName )
% Return an action number based on the type of plane indicated by
% the file name (searches through string for key words)

if strfind(fileName,'front')
    action = 1;
elseif strfind(fileName,'fp')
    action = 1;
elseif strfind(fileName,'Front')
```

```
    action = 1;
elseif strfind(fileName,'top')
    action = 2;
elseif strfind(fileName,'tp')
    action = 2;
elseif strfind(fileName,'Top')
    action = 2;
elseif strfind(fileName,'Plane 1')
    action = 2;
elseif strfind(fileName,'Plane 2')
    action = 2;
elseif strfind(fileName,'Plane 3')
    action = 2;
else action = 0;

end
```

```
function [  ] = analyzeFrontPlaneWithXVelocity( fileName )
% Calculates shear stresses from frontal plane/"coronal" view of the bioreactor
% Uses X Velocity

[fnR, fnC]=size(fileName);
fileNameShort = fileName(1,1:fnC-4); % Removes .csv extension of string
data=load(fileName);
mu = 0.000653;              % Pa*s @40 degrees C
[m,n] = size(data);         % Determine dimensions of file
shearx = zeros(m-1,4);
sheary = zeros(m-1,4);
A = sortrows(data,[2,1]);
numOfFilledRowsX = -1;
for i=1:m-1
   if A(i+1,2) == A(i,2)
    if abs(A(i+1,1)-A(i,1))<0.00085
       shearx(i,1)=A(i,1);
       shearx(i,2)=A(i,2);
       shearx(i,3)=A(i,3);
       shearx(i,4)=((A(i+1,4)-A(i,4))/(A(i+1,1)-A(i,1)))*-mu;
       numOfFilledRowsX = numOfFilledRowsX + 1;
    end
   end
end
r=1;
for k=1:m-1
    if shearx(k,1)~=0
        filteredShearx(r,[1 2 3 4])=shearx(k,[1 2 3 4]);
        r=r+1;
    end
end
```

```
B = sortrows(data,[1,2]);
numOfFilledRowsY = 0;
for i=1:m-1
   if B(i+1,1) == B(i,1)
    if abs(B(i+1,2)-B(i,2))<0.00085 % Ignore points at irrelevant locations
       sheary(i,1)=B(i,1);
       sheary(i,2)=B(i,2);
       sheary(i,3)=B(i,3);
       sheary(i,4)=((B(i+1,6)-B(i,6))/(B(i+1,2)-B(i,2)))*-mu;
       numOfFilledRowsY = numOfFilledRowsY + 1;
    end
   end
end
r=1;
for k=1:m-1
    if sheary(k,1)~=0
        filteredSheary(r,[1 2 3 4])=sheary(k,[1 2 3 4]);
        r=r+1;
    end
end


x_max = max(filteredShearx(:,1));
x_min = min(filteredShearx(:,1));
y_max = max(filteredShearx(:,2));
y_min = min(filteredShearx(:,2));


%Create the X, Y, and shear variables for shearx to plot
Xx= filteredShearx(:,1);
Yx= filteredShearx(:,2);
ShearInX = filteredShearx(:,4);

%Create the X, Y, and shear variables for sheary to plot
Xy= filteredSheary(:,1);
Yy= filteredSheary(:,2);
ShearInY = filteredSheary(:,4);

[totalRowsOfX, cx]=size(filteredShearx);
[totalRowsOfY, cy]=size(filteredSheary);

i = 1;
for rx = 1:totalRowsOfX
    for ry = 1:totalRowsOfY
        if filteredShearx(rx,1)==filteredSheary(ry,1)& ...
                filteredShearx(rx,2)==filteredSheary(ry,2)

                shearMagnitude(i, [1 3])= [filteredShearx(rx,1) ...
                    filteredShearx(rx,2)];
                shearMagnitude(i, [2 4])=[filteredSheary(ry,1) ...
                    filteredSheary(ry,2)];
                shearMagnitude(i, 5)=sqrt(filteredShearx(rx,4)^2+ ...
                    filteredSheary(ry,4)^2);
                i= i+1;
        end
    end
end

shear_max = max(shearMagnitude(:,5)); % Find overall maximum (magnitude)
shear_min = min(shearMagnitude(:,5)); % Find overall minimum (magnitude)

disp(['Impeller: ', fileNameShort]);
disp('Frontal Plane View, Stress Based on X Velocity:');
disp(['The maximum shear magnitude was ', num2str(shear_max), ' Pascal.']);
disp(['The minimum shear magnitude was ', num2str(shear_min), ' Pascal.']);
disp(' ');

figure;
scatter(shearMagnitude(:,1), shearMagnitude(:,3), 100, ...
    shearMagnitude(:,5), '.');
colormap(hsv)
colorbar
title(['X-Velocity-based Shear Magnitude (in Pascals) for ', fileNameShort]);
xlabel('X position');
ylabel('Y position');

[X,Y] = meshgrid(x_min:0.0008:x_max,y_min:0.0008:y_max);

Z=griddata(Xx,Yx,ShearInX,X,Y,'cubic'); % Use values from shearx array

figure;
contourf(X,Y,Z);
title(['X-Component of X-Velocity-based Shear for ', fileNameShort]);
xlabel('Width (mm)');
ylabel('Height (mm)');
colorbar;

x_max_2 = max(Xy);
x_min_2 = min(Xy);
y_max_2 = max(Yy);
y_min_2 = min(Yy);

[X2,Z2] = meshgrid(x_min_2:0.0008:x_max_2,y_min_2:0.0008:y_max_2);

Y=griddata(Xy,Yy,ShearInY,X2,Z2,'cubic'); % Use values from sheary

figure;
contourf(X2,Z2,Y);
title(['Y-Component of X-Velocity-based Shear for ', fileNameShort]);
xlabel('Width (mm)');
ylabel('Height (mm)');
colorbar;

end
```

```
function [  ] = analyzeFrontPlaneWithYVelocity( fileName )
% Calculates shear stresses from frontal plane/"coronal" view of the bioreactor
% Uses Y Velocity

[fnR, fnC]=size(fileName);
fileNameShort = fileName(1,1:fnC-4); % Removes .csv extension of string
data=load(fileName);
mu = 0.000653;              % Pa*s @40 degrees C
[m,n] = size(data);         % Determine dimensions of file
shearx = zeros(m-1,4);
sheary = zeros(m-1,4);
A = sortrows(data,[2,1]);
numOfFilledRowsX = -1;
for i=1:m-1
   if A(i+1,2) == A(i,2)
    if abs(A(i+1,1)-A(i,1))<0.00085
       shearx(i,1)=A(i,1);
       shearx(i,2)=A(i,2);
       shearx(i,3)=A(i,3);
       shearx(i,4)=((A(i+1,5)-A(i,5))/(A(i+1,1)-A(i,1)))*-mu;
       numOfFilledRowsX = numOfFilledRowsX + 1;
    end
   end
end
r=1;
for k=1:m-1
    if shearx(k,1)~=0
        filteredShearx(r,[1 2 3 4])=shearx(k,[1 2 3 4]);
        r=r+1;
    end
end

B = sortrows(data,[1,2]);
numOfFilledRowsY = 0;
for i=1:m-1
   if B(i+1,1) == B(i,1)
    if abs(B(i+1,2)-B(i,2))<0.00085 % Ignore points at irrelevant locations
       sheary(i,1)=B(i,1);
       sheary(i,2)=B(i,2);
       sheary(i,3)=B(i,3);
       sheary(i,4)=((B(i+1,6)-B(i,6))/(B(i+1,2)-B(i,2)))*-mu;
       numOfFilledRowsY = numOfFilledRowsY + 1;
    end
   end
end
r=1;
for k=1:m-1
    if sheary(k,1)~=0
        filteredSheary(r,[1 2 3 4])=sheary(k,[1 2 3 4]);
        r=r+1;
    end
end
```

```
x_max = max(filteredShearx(:,1));
x_min = min(filteredShearx(:,1));
y_max = max(filteredShearx(:,2));
y_min = min(filteredShearx(:,2));


%Create the X, Y, and shear variables for shearx to plot
Xx= filteredShearx(:,1);
Yx= filteredShearx(:,2);
ShearInX = filteredShearx(:,4);

%Create the X, Y, and shear variables for sheary to plot
Xy= filteredSheary(:,1);
Yy= filteredSheary(:,2);
ShearInY = filteredSheary(:,4);

[totalRowsOfX, cx]=size(filteredShearx);
[totalRowsOfY, cy]=size(filteredSheary);

i = 1;
for rx = 1:totalRowsOfX
    for ry = 1:totalRowsOfY
        if filteredShearx(rx,1)==filteredSheary(ry,1)& ...
                filteredShearx(rx,2)==filteredSheary(ry,2)

                shearMagnitude(i, [1 3])= [filteredShearx(rx,1) ...
                    filteredShearx(rx,2)];
                shearMagnitude(i, [2 4])=[filteredSheary(ry,1) ...
                    filteredSheary(ry,2)];
                shearMagnitude(i, 5)=sqrt(filteredShearx(rx,4)^2+ ...
                    filteredSheary(ry,4)^2);
                i= i+1;
        end
    end
end

shear_max = max(shearMagnitude(:,5)); % Find overall maximum (magnitude)
shear_min = min(shearMagnitude(:,5)); % Find overall minimum (magnitude)

disp(['Impeller: ', fileNameShort]);
disp('Frontal Plane View, Stress Based on Y Velocity:');
disp(['The maximum shear magnitude was ', num2str(shear_max), ' Pascal.']);
disp(['The minimum shear magnitude was ', num2str(shear_min), ' Pascal.']);
disp(' ');

figure;
scatter(shearMagnitude(:,1), shearMagnitude(:,3), 100, ...
    shearMagnitude(:,5), '.');
colormap(hsv)
colorbar
title(['Y-Velocity-based Shear Magnitude (in Pascals) for ', fileNameShort]);
```

```
xlabel('X position');
ylabel('Y position');

[X,Y] = meshgrid(x_min:0.0008:x_max,y_min:0.0008:y_max);

Z=griddata(Xx,Yx,ShearInX,X,Y,'cubic'); % Use values from shearx array

figure;
contourf(X,Y,Z);
title(['X-Component of Y-Velocity-based Shear for ', fileNameShort]);
xlabel('Width (mm)');
ylabel('Height (mm)');
colorbar;

x_max_2 = max(Xy);
x_min_2 = min(Xy);
y_max_2 = max(Yy);
y_min_2 = min(Yy);

[X2,Z2] = meshgrid(x_min_2:0.0008:x_max_2,y_min_2:0.0008:y_max_2);

Y=griddata(Xy,Yy,ShearInY,X2,Z2,'cubic'); % Use values from sheary

figure;
contourf(X2,Z2,Y);
title(['Y-Component of Y-Velocity-based Shear for ', fileNameShort]);
xlabel('Width (mm)');
ylabel('Height (mm)');
colorbar;

end
```

```
function [  ] = analyzeFrontPlaneWithZVelocity( fileName )
% Calculates shear stresses from frontal plane/"coronal" view of the bioreactor
% Uses Z Velocity

[fnR, fnC]=size(fileName);
fileNameShort = fileName(1,1:fnC-4); % Removes .csv extension of string
data=load(fileName);
mu = 0.000653;              % Pa*s @40 degrees C
[m,n] = size(data);         % Determine dimensions of file
shearx = zeros(m-1,4);
sheary = zeros(m-1,4);
A = sortrows(data,[2,1]);
numOfFilledRowsX = -1;
for i=1:m-1
   if A(i+1,2) == A(i,2)
    if abs(A(i+1,1)-A(i,1))<0.00085
       shearx(i,1)=A(i,1);
       shearx(i,2)=A(i,2);
       shearx(i,3)=A(i,3);
       shearx(i,4)=((A(i+1,6)-A(i,6))/(A(i+1,1)-A(i,1)))*-mu;
       numOfFilledRowsX = numOfFilledRowsX + 1;
    end
   end
end
r=1;
for k=1:m-1
    if shearx(k,1)~=0
        filteredShearx(r,[1 2 3 4])=shearx(k,[1 2 3 4]);
        r=r+1;
    end
end

B = sortrows(data,[1,2]);
numOfFilledRowsY = 0;
for i=1:m-1
   if B(i+1,1) == B(i,1)
    if abs(B(i+1,2)-B(i,2))<0.00085 % Ignore points at irrelevant locations
       sheary(i,1)=B(i,1);
       sheary(i,2)=B(i,2);
       sheary(i,3)=B(i,3);
       sheary(i,4)=((B(i+1,6)-B(i,6))/(B(i+1,2)-B(i,2)))*-mu;
       numOfFilledRowsY = numOfFilledRowsY + 1;
    end
   end
end
r=1;
for k=1:m-1
    if sheary(k,1)~=0
        filteredSheary(r,[1 2 3 4])=sheary(k,[1 2 3 4]);
        r=r+1;
    end
end


x_max = max(filteredShearx(:,1));
x_min = min(filteredShearx(:,1));
y_max = max(filteredShearx(:,2));
y_min = min(filteredShearx(:,2));


%Create the X, Y, and shear variables for shearx to plot
Xx= filteredShearx(:,1);
Yx= filteredShearx(:,2);
ShearInX = filteredShearx(:,4);

%Create the X, Y, and shear variables for sheary to plot
Xy= filteredSheary(:,1);
Yy= filteredSheary(:,2);
ShearInY = filteredSheary(:,4);
```

```
[totalRowsOfX, cx]=size(filteredShearx);
[totalRowsOfY, cy]=size(filteredSheary);

i = 1;
for rx = 1:totalRowsOfX
    for ry = 1:totalRowsOfY
        if filteredShearx(rx,1)==filteredSheary(ry,1)& ...
                filteredShearx(rx,2)==filteredSheary(ry,2)

                shearMagnitude(i, [1 3])= [filteredShearx(rx,1) ...
                    filteredShearx(rx,2)];
                shearMagnitude(i, [2 4])=[filteredSheary(ry,1) ...
                    filteredSheary(ry,2)];
                shearMagnitude(i, 5)=sqrt(filteredShearx(rx,4)^2+ ...
                    filteredSheary(ry,4)^2);
                i= i+1;
        end
    end
end

shear_max = max(shearMagnitude(:,5)); % Find overall maximum (magnitude)
shear_min = min(shearMagnitude(:,5)); % Find overall minimum (magnitude)

disp(['Impeller: ', fileNameShort]);
disp('Frontal Plane View, Stress Based on Z Velocity:');
disp(['The maximum shear magnitude was ', num2str(shear_max), ' Pascal.']);
disp(['The minimum shear magnitude was ', num2str(shear_min), ' Pascal.']);
disp(' ');

figure;
scatter(shearMagnitude(:,1), shearMagnitude(:,3), 100, ...
    shearMagnitude(:,5), '.');
colormap(hsv)
colorbar
title(['Z-Velocity-based Shear Magnitude (in Pascals) for ', fileNameShort]);
xlabel('X position');
ylabel('Y position');

[X,Y] = meshgrid(x_min:0.0008:x_max,y_min:0.0008:y_max);

Z=griddata(Xx,Yx,ShearInX,X,Y,'cubic'); % Use values from shearx array

figure;
contourf(X,Y,Z);
title(['X-Component of Z-Velocity-based  Shear for ', fileNameShort]);
xlabel('Width (mm)');
ylabel('Height (mm)');
colorbar;

x_max_2 = max(Xy);
x_min_2 = min(Xy);
y_max_2 = max(Yy);
y_min_2 = min(Yy);
```

```
[X2,Z2] = meshgrid(x_min_2:0.0008:x_max_2,y_min_2:0.0008:y_max_2);

Y=griddata(Xy,Yy,ShearInY,X2,Z2,'cubic'); % Use values from sheary

figure;
contourf(X2,Z2,Y);
title(['Y-Component of Z-Velocity-based Shear for ', fileNameShort]);
xlabel('Width (mm)');
ylabel('Height (mm)');
colorbar;

end
```

```
function []= analyzeTopPlaneWithXVelocity( fileName )
% Calculates shear stresses from bird's eye view of the bioreactor
% Uses X Velocity

[fnR, fnC]=size(fileName);
fileNameShort = fileName(1,1:fnC-4); % Removes .csv extension of string
data=load(fileName);
mu = 0.000653;              % Pa*s @40 degrees C
[m,n] = size(data);         % Determine dimensions of file
shearx = zeros(m-1,4);
shearz = zeros(m-1,4);
A = sortrows(data,[3,1]);
numOfFilledRowsX = -1;
for i=1:m-1
   if A(i+1,3) == A(i,3)
    %if abs(A(i+1,1)-A(i,1))<0.00085
       shearx(i,1)=A(i,1);
       shearx(i,2)=A(i,2);
       shearx(i,3)=A(i,3);
       shearx(i,4)=((A(i+1,4)-A(i,4))/(A(i+1,1)-A(i,1)))*-mu;
       numOfFilledRowsX = numOfFilledRowsX + 1;
    %end
   end
end
r=1;
for k=1:m-1
    if shearx(k,1)~=0
        filteredShearx(r,[1 2 3 4])=shearx(k,[1 2 3 4]);
        r=r+1;
    end
end

B = sortrows(data,[1,3]);
numOfFilledRowsZ = 0;
for i=1:m-1
```

```
   if B(i+1,1) == B(i,1)
    %if abs(B(i+1,3)-B(i,3))<0.00085 % Ignore points at irrelevant locations
       shearz(i,1)=B(i,1);
       shearz(i,2)=B(i,2);
       shearz(i,3)=B(i,3);
       shearz(i,4)=((B(i+1,5)-B(i,5))/(B(i+1,3)-B(i,3)))*-mu;
       numOfFilledRowsZ = numOfFilledRowsZ + 1;
    %end
   end
end
r=1;
for k=1:m-1
    if shearz(k,1)~=0
        filteredShearz(r,[1 2 3 4])=shearz(k,[1 2 3 4]);
        r=r+1;
    end
end


x_max = max(filteredShearx(:,1));
x_min = min(filteredShearx(:,1));
z_max = max(filteredShearx(:,2));
z_min = min(filteredShearx(:,2));


%Create the X, Y, and shear variables for shearx to plot
Xx= filteredShearx(:,1);
Zx= filteredShearx(:,3);
ShearInX = filteredShearx(:,4);

%Create the X, Y, and shear variables for sheary to plot
Xz= filteredShearz(:,1);
Zz= filteredShearz(:,3);
ShearInZ = filteredShearz(:,4);

[totalRowsOfX, cx]=size(filteredShearx);
[totalRowsOfZ, cz]=size(filteredShearz);


i = 1;
for rx = 1:totalRowsOfX
    for rz = 1:totalRowsOfZ
        if filteredShearx(rx,1)==filteredShearz(rz,1)& ...
                filteredShearx(rx,3)==filteredShearz(rz,3)

                shearMagnitude(i, [1 3])= [filteredShearx(rx,1) ...
                    filteredShearx(rx,3)];
                shearMagnitude(i, [2 4])=[filteredShearz(rz,1) ...
                    filteredShearz(rz,3)];
                shearMagnitude(i, 5)=sqrt(filteredShearx(rx,4)^2+ ...
                    filteredShearz(rz,4)^2);
                i= i+1;
        end
    end
end

shear_max = max(shearMagnitude(:,5)); % Find overall maximum (magnitude)
shear_min = min(shearMagnitude(:,5)); % Find overall minimum (magnitude)

disp(['Impeller: ', fileNameShort]);
disp('Top View, Stress Based on X Velocity:');
disp(['The maximum shear magnitude was ', num2str(shear_max), ' Pascal.']);
disp(['The minimum shear magnitude was ', num2str(shear_min), ' Pascal.']);
disp(' ');

figure;
scatter(shearMagnitude(:,1), shearMagnitude(:,3), 100, ...
    shearMagnitude(:,5), '.');
colormap(hsv)
colorbar
title(['X-Velocity-Based Shear Magnitude (in Pascals) for ', fileNameShort]);
xlabel('X position');
ylabel('Z position');

end
```

```
function []= analyzeTopPlaneWithYVelocity( fileName )
% Calculates shear stresses from bird's eye view of the bioreactor
% Uses Y Velocity

[fnR, fnC]=size(fileName);
fileNameShort = fileName(1,1:fnC-4); % Removes .csv extension of string
data=load(fileName);
mu = 0.000653;              % Pa*s @40 degrees C
[m,n] = size(data);         % Determine dimensions of file
shearx = zeros(m-1,4);
shearz = zeros(m-1,4);
A = sortrows(data,[3,1]);
numOfFilledRowsX = -1;
for i=1:m-1
   if A(i+1,3) == A(i,3)
    %if abs(A(i+1,1)-A(i,1))<0.00085
       shearx(i,1)=A(i,1);
       shearx(i,2)=A(i,2);
       shearx(i,3)=A(i,3);
       shearx(i,4)=((A(i+1,5)-A(i,5))/(A(i+1,1)-A(i,1)))*-mu;
       numOfFilledRowsX = numOfFilledRowsX + 1;
    %end
   end
end
r=1;
for k=1:m-1
```

```
    if shearx(k,1)~=0
        filteredShearx(r,[1 2 3 4])=shearx(k,[1 2 3 4]);
        r=r+1;
    end
end

B = sortrows(data,[1,3]);
numOfFilledRowsZ = 0;
for i=1:m-1
   if B(i+1,1) == B(i,1)
    %if abs(B(i+1,3)-B(i,3))<0.00085 % Ignore points at irrelevant locations
       shearz(i,1)=B(i,1);
       shearz(i,2)=B(i,2);
       shearz(i,3)=B(i,3);
       shearz(i,4)=((B(i+1,5)-B(i,5))/(B(i+1,3)-B(i,3)))*-mu;
       numOfFilledRowsZ = numOfFilledRowsZ + 1;
    %end
   end
end
r=1;
for k=1:m-1
    if shearz(k,1)~=0
        filteredShearz(r,[1 2 3 4])=shearz(k,[1 2 3 4]);
        r=r+1;
    end
end


x_max = max(filteredShearx(:,1));
x_min = min(filteredShearx(:,1));
z_max = max(filteredShearx(:,2));
z_min = min(filteredShearx(:,2));


%Create the X, Y, and shear variables for shearx to plot
Xx= filteredShearx(:,1);
Zx= filteredShearx(:,3);
ShearInX = filteredShearx(:,4);

%Create the X, Y, and shear variables for sheary to plot
Xz= filteredShearz(:,1);
Zz= filteredShearz(:,3);
ShearInZ = filteredShearz(:,4);

[totalRowsOfX, cx]=size(filteredShearx);
[totalRowsOfZ, cz]=size(filteredShearz);


i = 1;
for rx = 1:totalRowsOfX
    for rz = 1:totalRowsOfZ
        if filteredShearx(rx,1)==filteredShearz(rz,1)& ...
                filteredShearx(rx,3)==filteredShearz(rz,3)
                shearMagnitude(i, [1 3])= [filteredShearx(rx,1) ...
                    filteredShearx(rx,3)];
                shearMagnitude(i, [2 4])=[filteredShearz(rz,1) ...
                    filteredShearz(rz,3)];
                shearMagnitude(i, 5)=sqrt(filteredShearx(rx,4)^2+ ...
                    filteredShearz(rz,4)^2);
                i= i+1;
        end
    end
end

shear_max = max(shearMagnitude(:,5)); % Find overall maximum (magnitude)
shear_min = min(shearMagnitude(:,5)); % Find overall minimum (magnitude)

disp(['Impeller: ', fileNameShort]);
disp('Top View, Stress Based on Y Velocity:');
disp(['The maximum shear magnitude was ', num2str(shear_max), ' Pascal.']);
disp(['The minimum shear magnitude was ', num2str(shear_min), ' Pascal.']);
disp(' ');

figure;
scatter(shearMagnitude(:,1), shearMagnitude(:,3), 100, ...
    shearMagnitude(:,5), '.');
colormap(hsv)
colorbar
title(['Y-Velocity-Based Shear Magnitude (in Pascals) for ', fileNameShort]);
xlabel('X position');
ylabel('Z position');

end
```

```
function []= analyzeTopPlaneWithZVelocity( fileName )
% Calculates shear stresses from bird's eye view of the bioreactor
% Uses Z Velocity

[fnR, fnC]=size(fileName);
fileNameShort = fileName(1,1:fnC-4); % Removes .csv extension of string
data=load(fileName);
mu = 0.000653;              % Pa*s @40 degrees C
[m,n] = size(data);         % Determine dimensions of file
shearx = zeros(m-1,4);
shearz = zeros(m-1,4);
A = sortrows(data,[3,1]);
numOfFilledRowsX = -1;
for i=1:m-1
   if A(i+1,3) == A(i,3)
    %if abs(A(i+1,1)-A(i,1))<0.00085
       shearx(i,1)=A(i,1);
       shearx(i,2)=A(i,2);
       shearx(i,3)=A(i,3);
       shearx(i,4)=((A(i+1,4)-A(i,4))/(A(i+1,1)-A(i,1)))*-mu;
       numOfFilledRowsX = numOfFilledRowsX + 1;
    %end
   end
end
r=1;
for k=1:m-1
    if shearx(k,1)~=0
        filteredShearx(r,[1 2 3 4])=shearx(k,[1 2 3 4]);
        r=r+1;
    end
end

B = sortrows(data,[1,3]);
numOfFilledRowsZ = 0;
for i=1:m-1
   if B(i+1,1) == B(i,1)
    %if abs(B(i+1,3)-B(i,3))<0.00085 % Ignore points at irrelevant locations
       shearz(i,1)=B(i,1);
       shearz(i,2)=B(i,2);
       shearz(i,3)=B(i,3);
       shearz(i,4)=((B(i+1,5)-B(i,5))/(B(i+1,3)-B(i,3)))*-mu;
       numOfFilledRowsZ = numOfFilledRowsZ + 1;
    %end
   end
end
r=1;
for k=1:m-1
    if shearz(k,1)~=0
        filteredShearz(r,[1 2 3 4])=shearz(k,[1 2 3 4]);
        r=r+1;
    end
end


x_max = max(filteredShearx(:,1));
x_min = min(filteredShearx(:,1));
z_max = max(filteredShearx(:,2));
z_min = min(filteredShearx(:,2));


%Create the X, Y, and shear variables for shearx to plot
Xx= filteredShearx(:,1);
Zx= filteredShearx(:,3);
ShearInX = filteredShearx(:,4);

%Create the X, Y, and shear variables for sheary to plot
Xz= filteredShearz(:,1);
Zz= filteredShearz(:,3);
ShearInZ = filteredShearz(:,4);
```

```
[totalRowsOfX, cx]=size(filteredShearx);
[totalRowsOfZ, cz]=size(filteredShearz);


i = 1;
for rx = 1:totalRowsOfX
    for rz = 1:totalRowsOfZ
        if filteredShearx(rx,1)==filteredShearz(rz,1)& ...
                filteredShearx(rx,3)==filteredShearz(rz,3)

                shearMagnitude(i, [1 3])= [filteredShearx(rx,1) ...
                    filteredShearx(rx,3)];
                shearMagnitude(i, [2 4])=[filteredShearz(rz,1) ...
                    filteredShearz(rz,3)];
                shearMagnitude(i, 5)=sqrt(filteredShearx(rx,4)^2+ ...
                    filteredShearz(rz,4)^2);
                i= i+1;
        end
    end
end

shear_max = max(shearMagnitude(:,5)); % Find overall maximum (magnitude)
shear_min = min(shearMagnitude(:,5)); % Find overall minimum (magnitude)

disp(['Impeller: ', fileNameShort]);
disp('Top View, Stress Based on Z Velocity:');
disp(['The maximum shear magnitude was ', num2str(shear_max), ' Pascal.']);
disp(['The minimum shear magnitude was ', num2str(shear_min), ' Pascal.']);
disp(' ');

figure;
scatter(shearMagnitude(:,1), shearMagnitude(:,3), 100, ...
    shearMagnitude(:,5), '.');
colormap(hsv)
colorbar
title(['Z-Velocity-Based Shear Magnitude (in Pascals) for ', fileNameShort]);
xlabel('X position');
ylabel('Z position');

end
```

Appendix E: Temperature Probe Circuit Diagram and Code

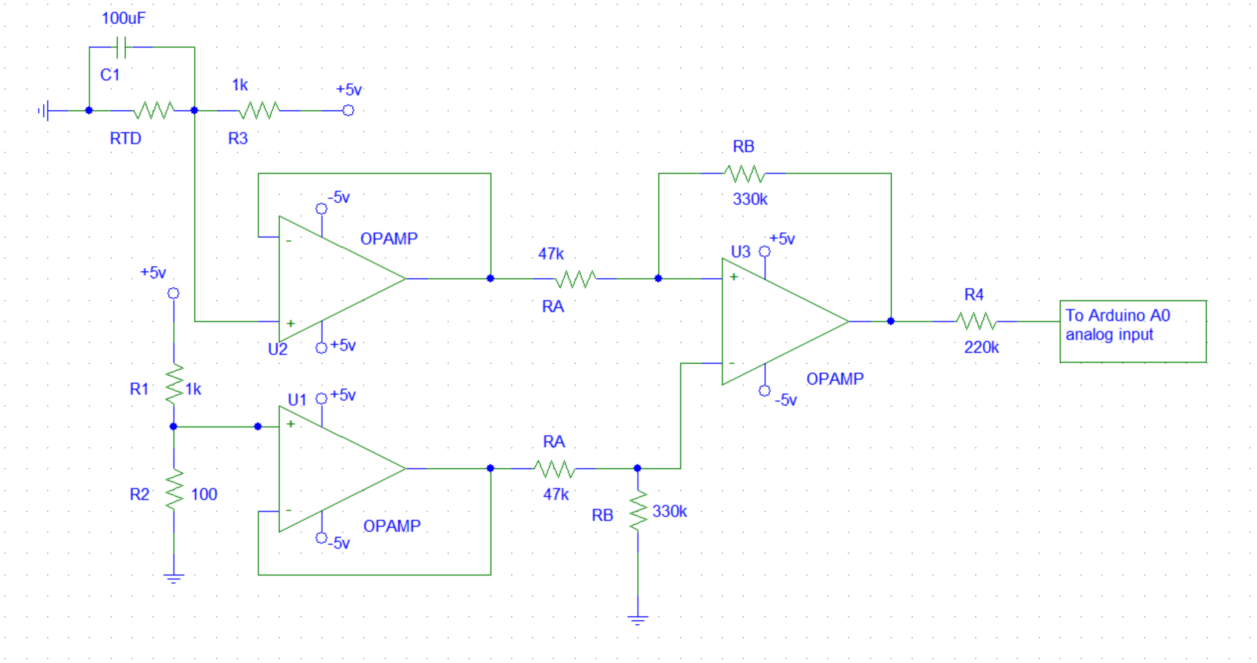
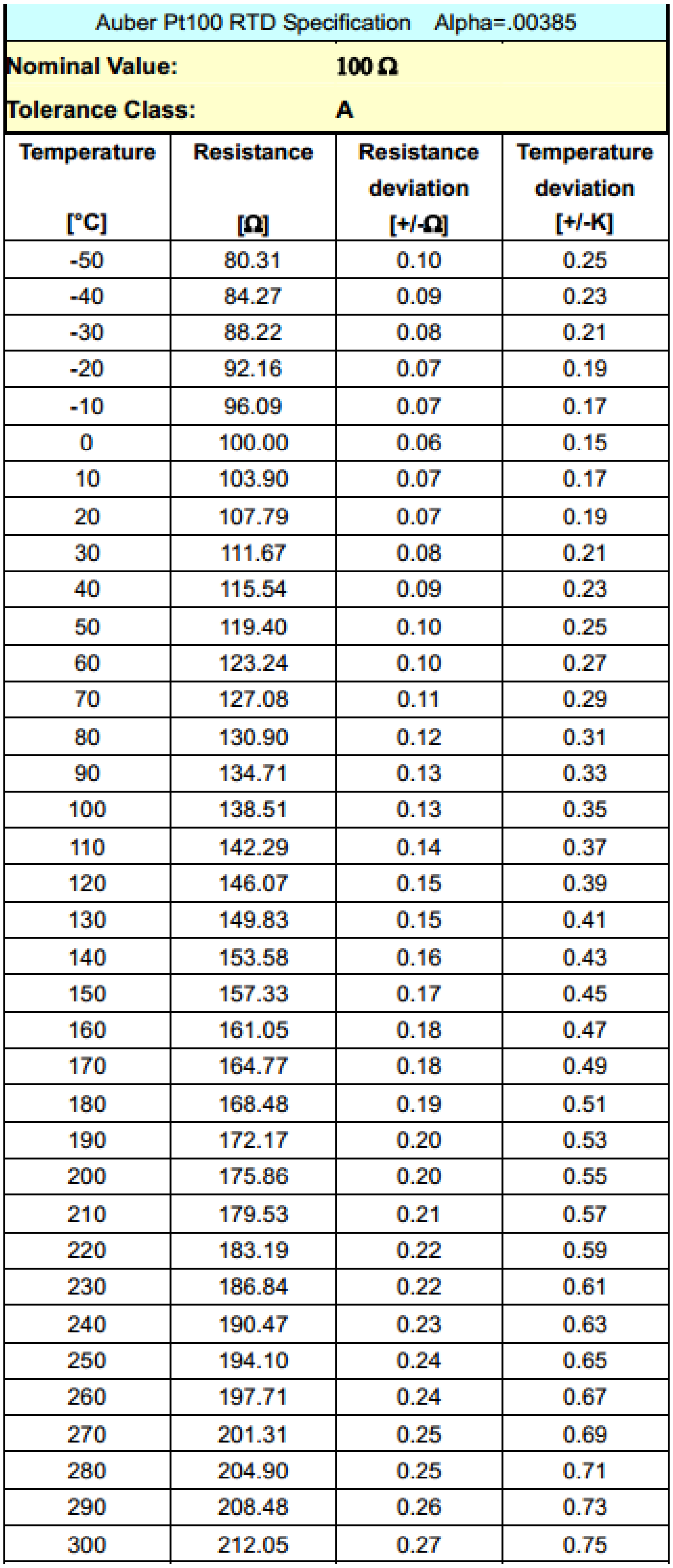


Figure C1: Circuit digram for the interface of the Pt100 RTD tempeature probe to the Arduino microcontroller.

Arduino program code:

```
int analogPin = 0; //Our analog pin
float analogInput = 0; //Stores incoming analog value
float voltage = 0; //Stores what input is in voltage
float temp = 0; //Stores what input is in degress Celcius
void setup()
{
 Serial.begin(9600);
 analogReference(INTERNAL); //Set analog reference voltage to 1.1V
}
void loop()
{
 delay(50); //Samples input every 50 milliseconds
 analogInput = analogRead(analogPin);
 voltage = (analogInput/1023)*1.0795; //1.0795V is max voltage
 Serial.print("Voltage: ");
 Serial.println(voltage);
 temp = (voltage / 1.0795) * 100; //Needs to be checked/calibrated
 Serial.print("Temperature: ");
 Serial.println(temp);
}
```

Table C1: Resistance vs Temperature tabel for the Pt100 RTD sensor [X4].



Auber Pt100 RTD Specification Alpha=.00385

Nominal Value:	100 Ω
Tolerance Class:	A

Temperature	Resistance	Resistance	Temperature
		deviation	deviation
[°C]	[Ω]	[+/-Ω]	[+/-K]
-50	80.31	0.10	0.25
-40	84.27	0.09	0.23
-30	88.22	0.08	0.21
-20	92.16	0.07	0.19
-10	96.09	0.07	0.17
0	100.00	0.06	0.15
10	103.90	0.07	0.17
20	107.79	0.07	0.19
30	111.67	0.08	0.21
40	115.54	0.09	0.23
50	119.40	0.10	0.25
60	123.24	0.10	0.27
70	127.08	0.11	0.29
80	130.90	0.12	0.31
90	134.71	0.13	0.33
100	138.51	0.13	0.35
110	142.29	0.14	0.37
120	146.07	0.15	0.39
130	149.83	0.15	0.41
140	153.58	0.16	0.43
150	157.33	0.17	0.45
160	161.05	0.18	0.47
170	164.77	0.18	0.49
180	168.48	0.19	0.51
190	172.17	0.20	0.53
200	175.86	0.20	0.55
210	179.53	0.21	0.57
220	183.19	0.22	0.59
230	186.84	0.22	0.61
240	190.47	0.23	0.63
250	194.10	0.24	0.65
260	197.71	0.24	0.67
270	201.31	0.25	0.69
280	204.90	0.25	0.71
290	208.48	0.26	0.73
300	212.05	0.27	0.75

Appendix F: Cell Protocols

i. Reprogramming protocol

To reprogram in suspension conditions, secondary inducible 6C MEFs were trypsinized 8 h after doxycycline induction and seeded either into Sigmacote (Sigma)-treated spinner flasks (Integra Biosciences) at 0.5 × 105 to 1 × 105 cells ml−1 or in low-cell-binding plates (Nunc). Primary mouse fibroblasts were transduced with viral preparations 24 h and 36 h after seeding. Cultures were supplemented with doxycycline 24 h after the last viral transduction. Eight to twelve hours after induction (doxycycline addition), cells were trypsinized and seeded into spinner flasks at 2 × 105 cells ml−1. Culture volumes were between 30–50 ml, cultured with a constant stirring speed of 65 r.p.m. One-third of the culture medium was replaced every day. Spinner flasks were replaced every 6 d to prevent sticking of cells to vessel walls. To remove large aggregates from high-density cultures, cells were passed through 100-µm cell strainers (BD Biosciences). All adherent cultures, spinner flasks and low-cell-binding plates were incubated in a humidified 5% (v/v) CO2 air environment at 37 °C [5].

ii. Immunocytochemistry and flow cytometry protocols

All surface stainings for flow cytometry were performed in the presence of 7AAD (Molecular Probes), and populations were gated on live cells. For cell sorting, cells isolated from the thymus were first blocked for 10 min on ice with mouse CD16/CD32 Fc block (clone 2.4G2, BD 553142). The cells were subsequently labeled with the conjugated antibodies above for 20 min on ice to sort for CD4−CD8−CD25−CD44+ DN1 T cells using a FACSAria flow cytometer (BD Biosciences). For intracellular staining, cells were fixed with PBS containing 4% formaldehyde and then permeabilized with methanol. Analysis was performed on FACSDiva (BD Biosciences) as well as FlowJo (Tree Star).

Immunocytochemistry stainings were performed by fixing cells in PBS containing 4% (v/v) formaldehyde. Cells were permeabilized in PBS containing 0.1% (v/v) Triton X-100 and subsequently blocked in PBS containing 10% (v/v) donkey serum. Samples were incubated with primary and secondary antibodies in PBS containing 1% (w/v) BSA and imaged using a confocal microscope (FV1000 laser scanning confocal; Olympus) with 5 µm optical sections. Images represent the *z*-stack projection of five to ten confocal optical sections. Annexin V (Invitrogen) staining was carried out according to the manufacturer’s protocol. EdU cell proliferation assays were performed according to the manufacturer’s protocol (Invitrogen). Calcein-AM and ethidium homodimer I staining (Sigma) was performed as indicated in the manufacturer’s protocol [5].

Appendix G: Cell Culture Budget

Cell Culture Budget (Tentative Estimates)

Item	vendor	catalog #	unit price	Quantity**	total cost*
DMEM/F12	Invitrogen	11330	255.78	1	255.78
MEM Non-Essential Amino Acids	Invitrogen	2700845	15.71	1	15.71
L-Glutamine-200 mM (100X), Liquid	Invitrogen	25030081	22.17	1	22.17
LIF	Millipore	LIF1010	262.2	1	262.2
B27	Invitrogen	17504044	79.65	1	79.65
N2	Invitrogen	17502-048	65.33	1	65.33
Penicillin-Streptomycin	Sigma	P4333-20ML	10	1	10
Glutamax supplement	Invitrogen	10564-011	36.67	1	36.67
1 mM sodium pyruvate	Sigma	S8636	9.1	1	9.1
FBS (500mL)	Invitrogen	26140-079	280	1	280
6-well plates, case of 100	Fisher	07-200-80	108.14	1	108.14
E8 media	Invitrogen	A14666SA	195.98	1	195.98
accutase	Invitrogen	A1110501	34.67	1	34.67
Doxycycline	Sigma	D9891-1G	38	1	38
				Total estimate	1413.4

* Immunocytochemistry and flow cytometry materials are not included here

** Quanties may vary in the future depending on number of experiments performed

Appendix H: Protocols

Standard Polydopamine Deposition Procedure

1) Tris-Burffer 0.1M @ PH=8.5 (1L): 500ml DI water in beaker and add 15.8g Tris-HCl. Adjust PH value with 1M NaOH aqueous solution. Store the buffer solution at ambient temperature.

2) Dopamine (2 mg/mL) was dissolved in 10 mM Tris-HCl (pH 8.5), and substrates were dipped into the solution. pH-induced oxidation changes the solution color to dark brown. Stirring and/or vertical sample orientation were necessary to prevent non-specific microparticle deposition on surfaces. The coated surfaces were rinsed with ultrapure water and dried by N_2 gas before storage.

Notation: Incubation of dopamine solution at room temperature for several days (>3days) prior to immersion of substrates did not produce surface discoloration (to dark-brown) typical of polydopamine coatings, indicating that the coating did not occur or was too thin to observe visually.

Standard Polydopamine Initiator Deposition Procedure

Dopamine (Dopamine hydrochloride 239 mg,1.26 mmol)was placed in a flask which was degassed by purging with dry N_2 for 5 min. To this flaskwas added N,N'-dimethylformamide (DMF, 12 mL), 2-bromoisobutyryl bromide (BIBB, 78µL, 0.63 mmol), and triethylamine (90µL,0.63 mmol) under dry N_2. After stirring under dry N_2 at room temperature for 3 h, this mixture was transferred to a glass bottle to which tris(hydroxymethyl) aminomethane (TRIS) (288 mg,2.4 mmol, add 24 ml TRIS Buffer) and deionized water (36ml)were added. continuously magnetically stirred at a speed of 200 rpm while open to the air. Polydopamine initiator-coated substrates were removed from the solution after various deposition times washed with deionised water and dried with compressed air. Samples with varying BIBB:dopamine ratios were prepared as above, but with the BIBB and triethylamine concentrations being altered according to the desired ratio.

Standard SI-ARGET ATRP with PEGMEMA using CuBr(II) and Byp and starting reaction by adding L-ascorbic acid.

SI-ARGET ATRP Procedure

General protocols of SI-ARGET ATRP for PEGMEMA brushes used to synthesize brushes matrix as follow: 2.75ml deionized water, 3.75ml methanol and PEGMEMA (16.2mmol, 8g), Copper(II) Bromide (0.08mmol, 17.9mg) and 2'2-Bipyridine (0.24mmol, 37.5mg) were mixed in schlenk flask and degassed with three freeze-vacuum-thaw cycles. Then the degassed mixture was transferred into the vacuumed reaction flask containing modified dopamine coating samples by syringe. To start the reaction, ascorbic acid (0.8mmol, 140.9mg) in 1ml deionized was purged into the flask, a clear color change from bright blue to brown was observed. The reaction was performed for 16h in purpose of high thickness brushes film.

Appendix I: Poly Styrene Coating Protocol

Testing polystyrene as a coating material for the bioreactor's impeller

Purpose: To manufacture an impeller for our Stirred Suspension Bioreactor, we will initially be using a 3D printer to produce our design due to its ease, accuracy, and relatively low cost. The material used in the SLA printing process is Accura60, which in previous tests has been found to be cytotoxic.

To prevent cell death in our initial testing with 3T3 cells we will coat the Accura60 material with Polystyrene. This test will determine whether coating the SLA material will be suitable for use in suspension culture.

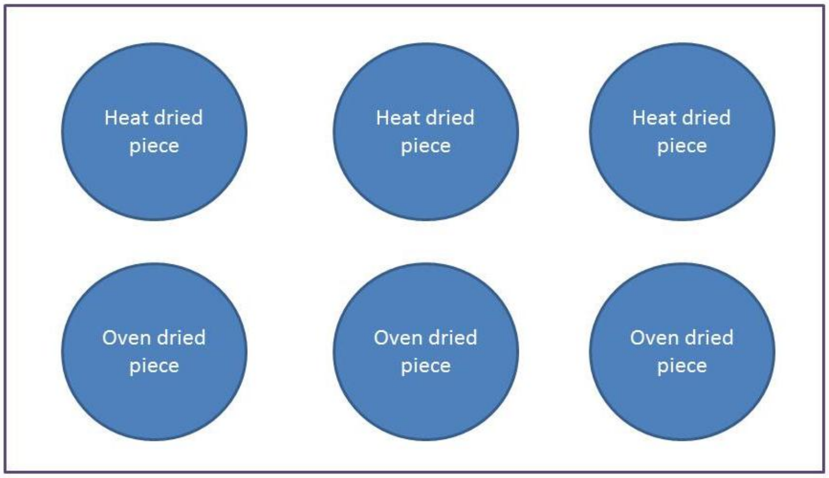
Process:

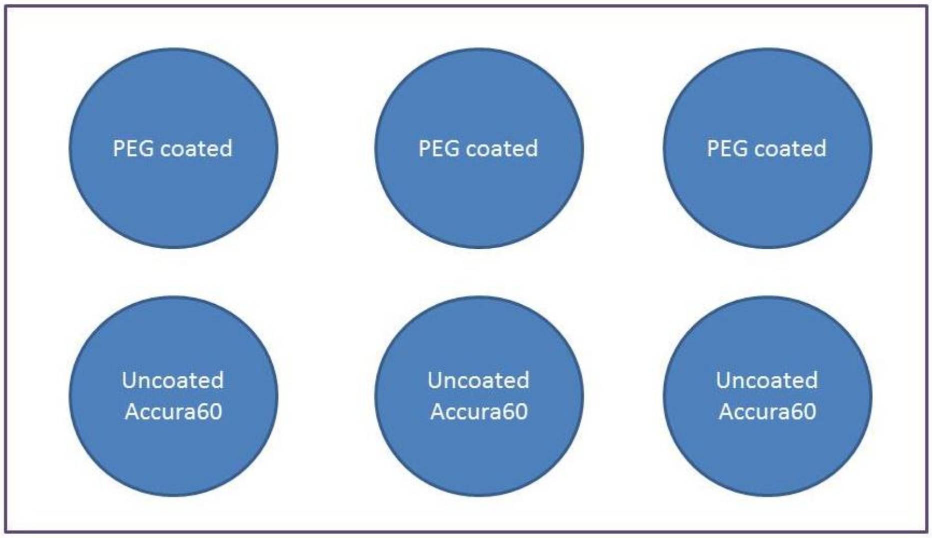
1. Passage H9 human embryonic stem cells and place the piece of material in the well
2. Maintain the cells until next passage is needed.
3. Singularize cells
4. Stain dead cells with Trypan Blue
5. Count and compare to control wells.

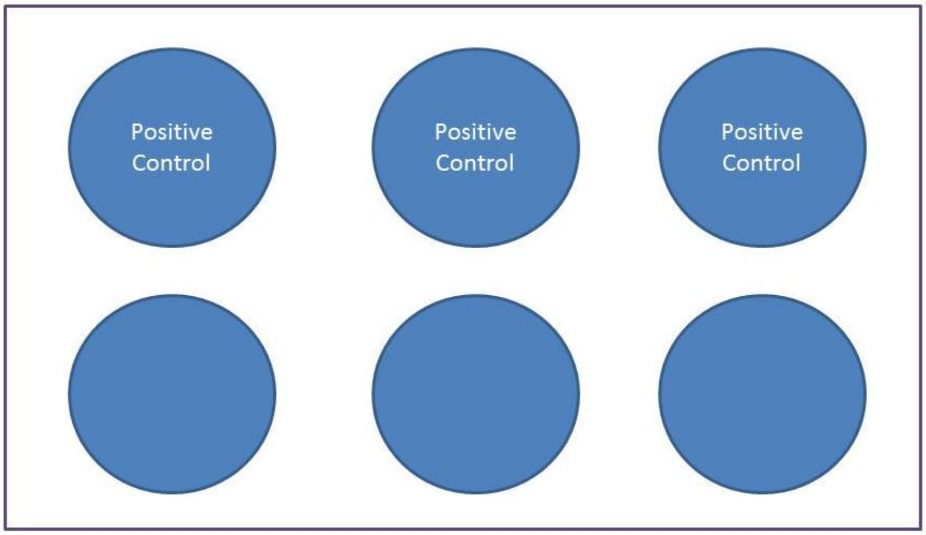
Materials

- E8 media
- EDTA
- 6-well plates
- Trypan blue
- Accutase

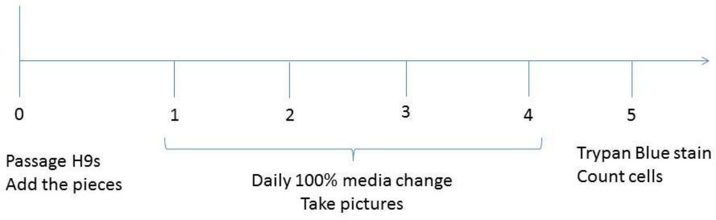
Design







Timeline



Protocols

- Cleaning Coated Pieces

For PS coated pieces

1. Rinse with 100% Ethanol 3X

2. Rinse with PBS 3X

For PEG coated pieces

1. Rinse with 100% Ethanol 2X for 15 seconds

2. Soak in 100% Ethanol for 2 hours on a shaker

3. Rinse with PBS 3X

Trypan Blue stain:

1. Place 0.5 ml of a suitable cell suspension (dilute cells in complete medium without serum to an approximate concentration of 1 x 10^5 to 2 x 10^5cells per ml) in a screw cap test tube.

2. Add 0.1 ml of 0.4% Trypan Blue Stain. Mix thoroughly.

3. Allow to stand 5 min at 15 to 30°C (room temperature).

4. Fill a hemocytometer as for cell counting.

5. Under a microscope, observe if non-viable are stained and viable cells excluded the stain.

Results

- After day 2 of cell culture, all wells containing a sample (coated and uncoated) had all dead cells, at which point we ended the experiment
- Future work will have to look into different coating methods or finding other ways of manufacturing the impeller