Nutrient Diffusion through a 3-D Bio-printed Cellular Matrix

Michael Hu - Fikret Yalcinbas - Zeyang Shen - Ed Kantz

Table of Contents

1. Introduction

Presently, organ transplants are the most effective treatments, and in many cases, the only viable treatments, for end stage organ failure[15]. However, increased demand for organ donations has far outstripped the supply. Between 1995 and 2005, the waiting list for organ donations grew 19% annually, while the annual increase in the number of donors only reached 3.7%[15]. This trend has changed minimally, and the result is that there are now over 100,000 individuals in the US awaiting organ transplants[1], with 10,000 being added each year, with less than 25% of them receiving transplants[5]. The situation is poor enough in fact, that several European nations have considered providing monetary incentive to increase the number of willing donors[4].

Consequently, much work has been dedicated towards investigating the possibility of using engineered cellular and tissue scaffolds as a substitute for biologically-derived organs[6]. While there has been a significant amount of success in engineering skin grafts and dermal implants, generating viable tissue scaffolds for more advanced organs has been much more difficult[6]. A major reason for this is the limited or lack of vascularization in many of these engineered constructs. In vitro vascularization has historically been difficult due to the structural and organizational complexity involved. However, it is also essential, as the limited diffusivity of most essential nutrients and metabolites mean that cells cannot survive more than 100 um from a nutrient source[12]. Because of this, non-vascularized scaffolds have extremely limited thickness and overall size.

With the development of 3D bioprinting however, the difficulties of vascularization have been greatly mitigated, as the printers allow for construction of relatively complex and vascularized 3D structures while minimally compromising the quality of the scaffold. The 3D bioprinting process is, on the whole, relatively simple. In general, only 3 major components are needed: a matrix material, a material for the vascular channels, and a cell type[10]. The matrix material is typically a biocompatible polymer (i.e. collagen, gelatin, etc.), and is used to encapsulate the desired cell type. The 3D printer is then used to extrude the cell-matrix mix to form a base of predetermined dimensions. On top of the base, the printer will then extrude the vascular channel material to form the vascular pattern desired. This will subsequently be enclosed by layering the cell-matrix mix on top. The vascular material is then evacuated to open the vascular channel, which will then be perfused with nutrient media to ensure cell survival (**Figure 1**).

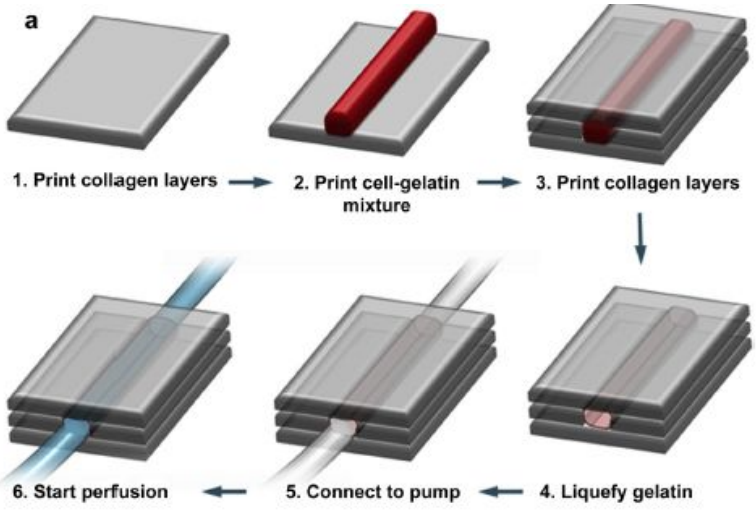

Figure 1: procedure for 3D printing biomatrices[9]

While the procedure is relatively straightforward, there are many parameters that need to be optimized, including the type of material used for the matrix, the dimensions of the matrix, and the cell seeding density[8]. Experimentally, this can become extremely time-consuming, as the flexibility of most 3D printers allows for a huge number of customizable parameters to be combined. As such, a mathematical model predicting the diffusive distribution of essential nutrients in a cellularized matrix may be extremely relevant and helpful.

2. Problem Statement

2.1 General Goal

The goal of the project is to create a mathematical model that can be used to characterize diffusion of a specified nutrient through a 3D bio-printed matrix, while forecasting the survival of the seeded cells utilizing nutrient consumption rates per cell and cell proliferation rates obtained from literature.

2.2 System Setup

A diagram of the system is shown in **Figure 2**, consisting of a single vascular channel running through the center of a bio-printed, cell-seeded matrix inside a sealed container. The nutrient of interest is glucose, which is known to be essential for cell survival as a carbon source[7]. Specifically, a solution of Dulbecco's Modified Eagle Medium (DMEM), a common cell culture medium, with standard glucose concentration is run through the vascular structure, from which it can continuously diffuse through the matrix in the directions indicated by arrows. Simultaneously, as the glucose diffuses throughout the matrix, it will also be consumed by the cells.

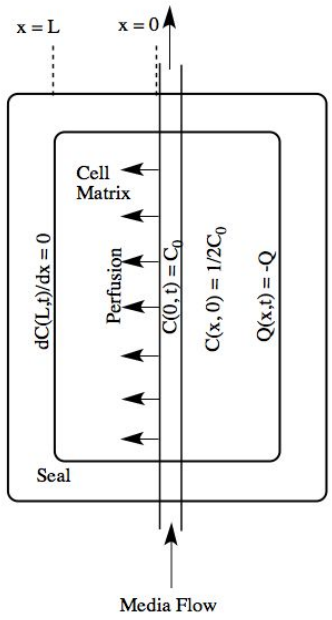

Figure 2: Initial diffusion-system setup

2.3 Model Setup

Overall, glucose concentration can be modeled by the one-dimensional diffusion equation shown below.

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} + Q(x,t)$$

Here, C(x,t) represents glucose concentration across the matrix as a function of both space and time. For simplicity, we initially assume that the concentration only changes in the x dimension, with a multidimensional model described later in Section 6. In our scenario, the matrix will be seeded with HepG2 cells, which are a carcinoma cell line frequently used to model hepatocytes. Because the cells seeded in the matrix consume glucose, we add the forcing term Q(x,t). Throughout this report, Q(x,t) will be modified such that it gradually becomes more representative of an actual experimental scenario. The diffusion coefficient of glucose is represented by “D”, and is dependent on the material of the matrix. While a large variety of materials can be used, we will limit ourselves to two biopolymers that are most frequently used in this context, collagen and alginate[2].

Notably, the matrix is symmetric about the vascular channel, meaning both halves should have identical diffusion properties. As such, we only model one half of the matrix, assuming that the behavior of the other half will be identical.

With regards to boundary conditions, the location of the vascular structure is set as x = 0, and the edge of the matrix as x = L. In our model, L is given a practical value of 2 cm, which is applicable in many printing systems[8]. Because media with a set glucose concentration constantly flows through vascular structure, a constant value boundary condition exists at x = 0 such that the glucose concentration is equal to that of DMEM, designated as C(0, t) = C_0 (C_0 =

24.75 mM[7]). By existing protocols, the matrix is typically sealed on all sides to prevent contamination[8], meaning a zero-flux boundary condition exists at x = L. Additionally, based on existing protocols, matrices are typically formulated with a 1:1 dilution of media[8]. Thus, initial glucose concentration throughout the matrix should be half that of the DMEM, and C(x, 0) = $C_0/2$. The initial condition and boundary conditions are summarized below, and remain relevant in all models described in this report unless otherwise stated.

Initial Condition Boundary Conditions

$$C(x,0) = \frac{C_0}{2} \qquad C(0,t) = C_0 \quad \frac{\partial C}{\partial x}(L,t) = 0$$

3. 1-D Glucose Diffusion in a Collagen Matrix with Stable Cell Density

As the simplest case, we assume that the cells seeded in a collagen matrix neither proliferate nor die, while consuming glucose at a constant rate. The rate of glucose consumption ($-Q_0$) can be calculated by multiplying the glucose uptake rate per cell (U) by the seeding density (ρ_0). The rate of hepatocyte glucose consumption (U = 6.48 x 10^{-11} mmol/cell/hour) was obtained from literature[3], and cell seeding density was set at a standard of 5 x 10^8 cells/L[9]. Additionally, the diffusion constant (D) of glucose through collagen is reported as being D = 1.3 x 10^{-6} cm^2/s[14] This scenario is characterized in **Figure 1**, and can be represented by the partial differential equation below.

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} - Q_0$$

Because no time-dependence exists in either the boundary or forcing terms, this PDE can be solved using the “Poison-Tooth Extraction Method”.

The particular solution $C_p(x,t)$ can be calculated by making a steady state assumption to remove the time-dependent term, and then satisfying the boundary conditions.

$$\frac{\partial C_p}{\partial t} = 0 = D\frac{\partial^2 C_p}{\partial x^2} - Q_0$$

$$C_p(x) = \frac{Q_0}{2D}x^2 - \frac{Q_0 L}{D}x + C_0$$

We then solve for the homogeneous equation in which the forcing term is removed, and boundary conditions are set as zero.

$$\frac{\partial C_H(x,t)}{\partial t} = D\frac{\partial^2 C_H(x,t)}{\partial x^2} \qquad C_H(0,t) = 0 \qquad \frac{\partial C_H}{\partial x}(L,t) = 0$$

Using the separation of variables technique, we can calculate the homogeneous solution as a Fourier Series, shown below.

$$C_H(x,t) = \sum_{n=0}^{\infty} B_n \sin\left(\frac{2n+1}{2L}\pi x\right) e^{-D\left(\frac{2n+1}{2L}\pi\right)^2 t}$$

The full solution is obtained by summing the particular and homogeneous solutions, and then applying initial conditions to solve for the orthogonal Fourier coefficient B_n.

$$B_n = \frac{2}{L}\int_0^L \left(-\frac{Q_0}{2D}x^2 + \frac{Q_0 L}{D}x - \frac{C_0}{2}\right)\sin\left(\frac{2n+1}{2L}\pi x\right)dx$$

$$= \frac{16Q_0L^2}{D(2n+1)^3\pi^3} - \frac{2C_0}{(2n+1)\pi}$$

Detailed steps are listed in *Appendix A*.

The solution is displayed in **Figure 3a** over a time span of 960 hours, representing 4 weeks of cell culturing. An equivalent numerical solution can be obtained using the MATLAB function pdepe, and is shown in **Figure 3b**.

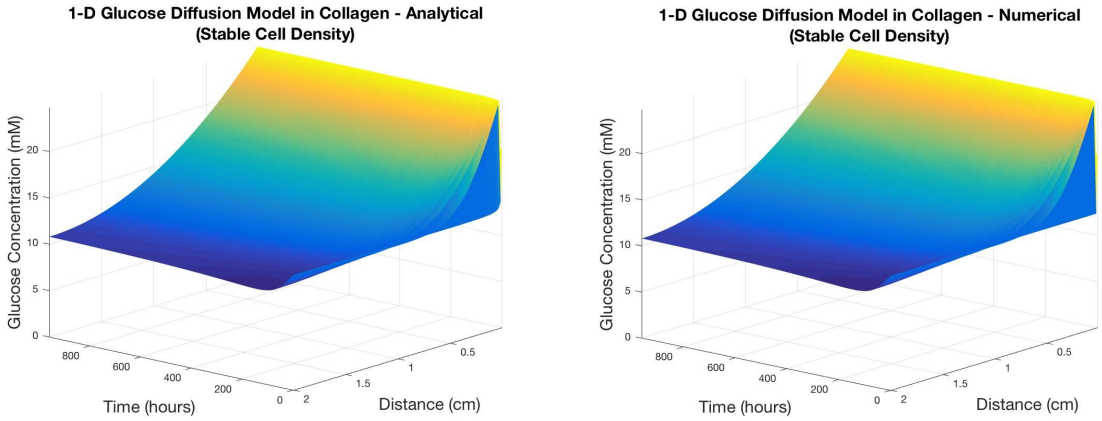

Figure 3: Analytical and numerical results of 1-D glucose diffusion model in collagen with stable cell density.

Several prominent features can be noted in the graphs. At x = 0, the concentration remains C_0, indicating the presence of the glucose-containing DMEM. From there, the glucose concentration rapidly drops off, as perfusion of glucose to further regions is diffusion-dependent. At t = 0, the concentration is consistently $C_0/2$ along x, as the matrix is formulated to be half-composed of DMEM. As time goes on, the glucose concentration decreases at all locations before stabilizing, marking the balance between diffusion from the source, and consumption by the cells. In both cases, even the furthest locations from the vascular channel maintain glucose concentrations above zero, indicating that the system is adequate to maintain a cell population at a 5 x 10^8 cells/L density.

4. 1-D Glucose Diffusion in a Collagen Matrix with Exponential Cell Growth

The stable cell density scenario presented in the previous section illustrates the simplest form of the problem being considered, and provides an adequate representation when the seeded cells are not likely to proliferate, as is the case with select cell types and conditions. However, the goal when creating a 3D cellular matrix, is typically for the cells to proliferate such that they are able to remodel the matrix in a manner that allows it to be transplanted in place of damaged tissue[6].

In most cases, cells will proliferate as long as they are provided with conditions required for survival, and the particular condition examined here is adequate supply of glucose. Here, we make several assumptions to model this proliferation.

Because mitosis divides a single cell into two, the number of cells in a growing culture should effectively double with each round of division. This suggests that cell proliferation could be accurately modeled as exponential. With increased cell number however, there should also be increased glucose consumption. Thus, the driving force (Q) in the model will now be function of time. Notably, the glucose consumption rate will still be calculated as a constant uptake rate per cell (U) multiplied by cell density (ρ), but rather than being constant, cell density will grow exponentially, as summarized below.

$$Q = U\rho_0 \rightarrow Q(t) = U\rho(t)$$
$$\rho(t) = \rho_0 2^{kt} \rightarrow -Q(t) = -U\rho_0 2^{kt}$$

As was the case with the previous model, Q is the rate of glucose consumption in mM/hour and is the forcing term of the system, while U is the glucose uptake rate per cell in mmol/hour/cell. Additionally, $\rho(t)$ is the time-dependent cell density (cells/L), which is based on exponential growth of the seeding density at time t = 0, ρ_0 (cells/L). Notably, the actual exponential model is specified by the exponential growth constant k, which is derived from literature[13].

The revised differential equation with the new driving force term becomes:

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} + Q(t) = D\frac{\partial^2 C(x,t)}{\partial x^2} - U\rho_0 2^{kt}$$

Because the driving force is now a function of time, the equation must be solved analytically using Green's Function. In particular, Green's Function was specified by the value-flux boundary conditions described in Section 2.2.

$$G(x,t,x_0,t_0) = \sum_{n=odd}^{\infty} \frac{2}{L}\sin\left(\frac{n\pi x_0}{2L}\right)\sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t-t_0)}$$

The equation is then solved by applying the Green's Function in 4 different terms; one for the initial condition, one for the driving force, one for the constant-value boundary at x = 0 and one

for the flux boundary at x = L. The function for the concentration in terms of x and t is then the summation of these four terms. This is shown below.

C(x,t) =

Initial Condition Term

$$\int_0^L g(x_0)G(x,t,x_0,t_0=0)dx_0$$

Driving force term

$$+\int_0^L\int_0^t Q(x_0,t_0)G(x,t,x_0,t_0)dx_0$$

Value boundary at x = 0 term

$$+\int_0^t C_0(t_0)D\frac{\partial}{\partial x_0}G(x,t,x_0=0,t_0)dt_0$$

Flux boundary at x = L term

$$+\int_0^t DC'_L(t_0)G(x,t,x_0=L,t_0)dt_0$$

Each of these terms is then solved analytically, and the individual solutions are summed to arrive at the full solution to the partial differential equation.

The detailed solution is included as an appendix to this report. But the full solution is provided below.

$$C(x,t)=\frac{C_0}{2}\sum_{n=odd}^{\infty}\frac{4}{n\pi}sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t)}-U\rho_0\frac{16L^2}{n\pi}\sum_{n=odd}^{\infty}sin\left(\frac{n\pi x}{2L}\right)\frac{2^{kt}-e^{-D\left(\frac{n\pi}{2L}\right)^2 t}}{D(n\pi)^2+kL^2 4\,ln(2)}$$

$$+C_0\sum_{n=odd}^{\infty}\frac{4}{n\pi}sin\left(\frac{n\pi x}{2L}\right)\left(1-e^{-D\left(\frac{n\pi}{2L}\right)^2(t)}\right)$$

This solution and its numerical equivalent, generated using the MATLAB pdepe function, are plotted below (**Figure 4**). Because of the unique nature of the Green's Function solution as an infinite summation series, the x = 0 term of the expression is calculated incorrectly as being zero, and as such, the constant-value boundary is cut off from the graph.

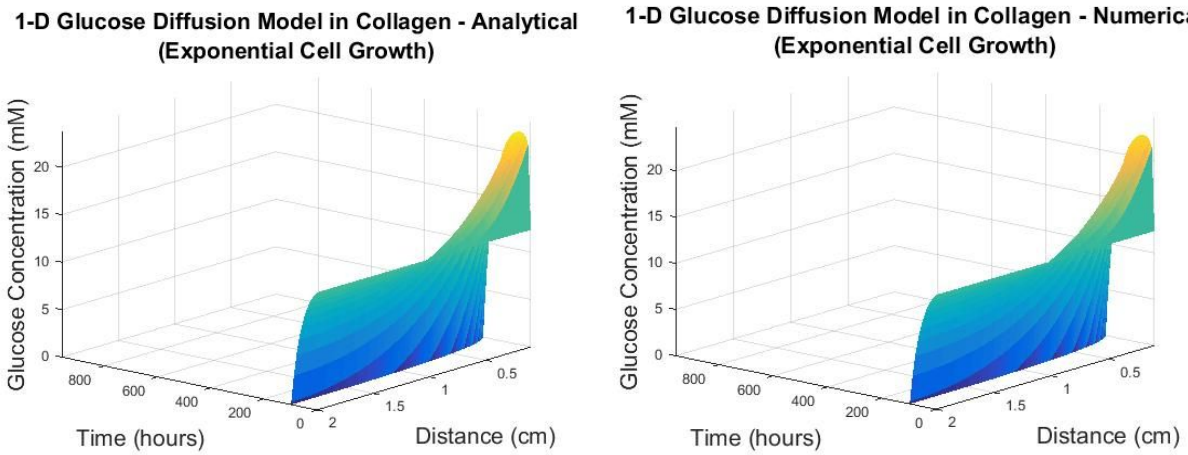

Figure 4: Analytical and numerical results of 1-D glucose diffusion in collagen with exponential cell growth.

As expected, the analytical and numerical methods produced similar results, which can also be compared with the results of the first scenario (shown again below for convenience).

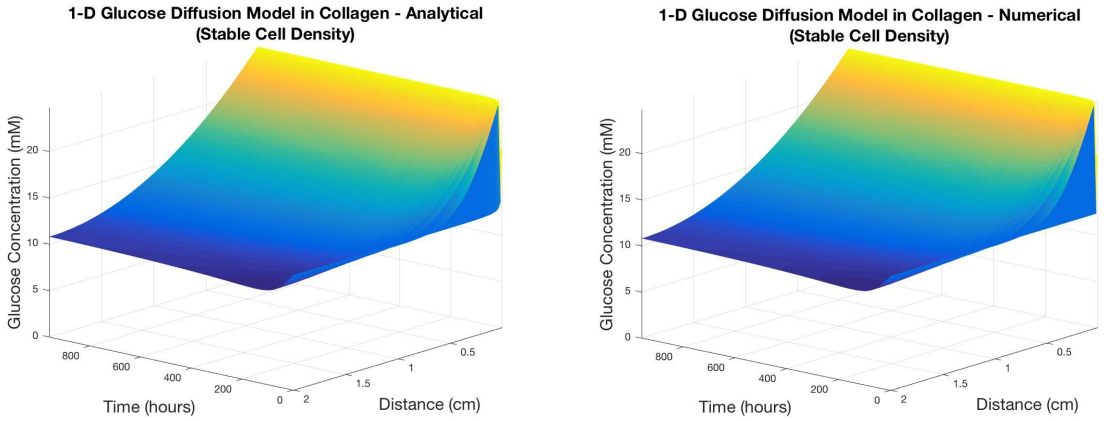

Figure 5: Analytical and numerical results of 1-D glucose diffusion model in collagen with stable cell density (identical to Figure 3).

Clearly, the addition of exponential cell growth resulted in rapid depletion of the glucose concentration, even in the portion of the matrix near the glucose source ($x \cong 0$). A clearer surface from the exponential growth scenario (zoomed in on timescale) is shown in **Figure 6**.

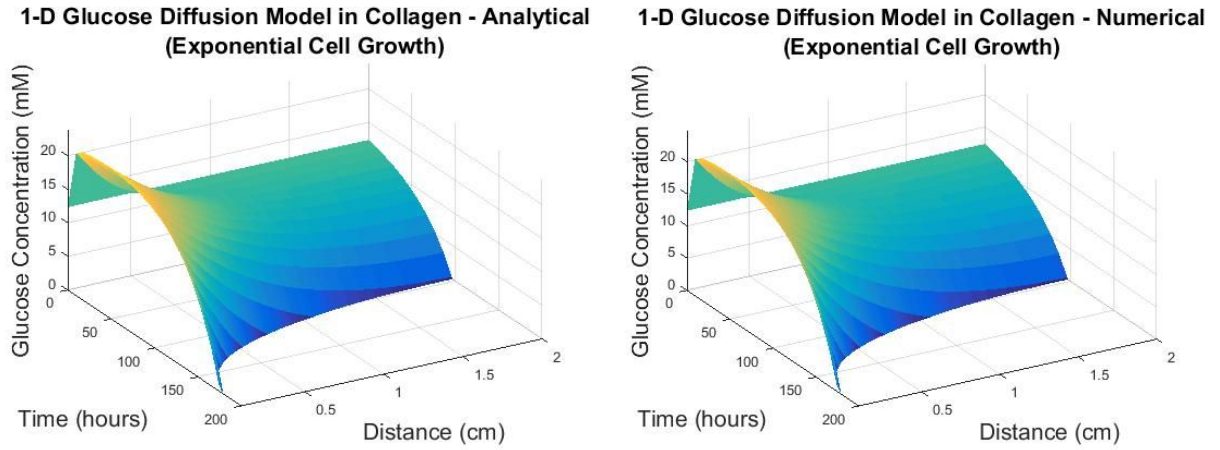

Figure 6: Analytical and numerical results of 1-D glucose diffusion in collagen with exponential cell growth

Note that the axis orientation and time span of these plots has been altered from the preceding plots to improve the clarity of the results. The plots were again generated just positive of x = 0 to remove distortion associated with the boundary. These plots even more clearly illustrate the rapid depletion of the glucose concentration which reaches 0 in the portion of the matrix right next to the glucose source in approximately 150 hours.

This result was not unexpected. The exponential growth assumption was an over-simplification that generated cell densities that were much greater than what would likely occur in a real matrix in a very short period of time. This is due to the fact that, while cells do proliferate in a way that can be approximated exponentially, this pattern of proliferation has limits. The issues with the exponential growth assumption are discussed in the following section.

5. 1-D Glucose Diffusion in Collagen and Alginate Matrices with Sigmoidal Cell Growth

Fundamentally, while exponential cell growth does add time-dependence to the system, it should be noted that it does not properly model cell growth long term. The reason for this is that cells do not divide infinitely as an exponential model suggests, since the matrix that the cells are encapsulated in has a limited amount of space.

A more accurate time-dependent growth model would be a sigmoidal curve, in which cells initially grow at an exponential rate, but subsequently level off at a maximum. It is difficult to determine exactly what the maximum growth level is for any group of cells, since it is affected by the cell type, the available space, and the cell morphology, but for this model, we will assume a maximum at a 10 x 10^9 cells/L density. Using previously-characterized growth data for the HepG2 cell line[13], a sigmoidal growth curve can be modeled as follows:

$$P(t) = \frac{P_{max}}{1 + e^{-k\left(t - \frac{\ln(P_{max} - P_0)}{k}\right)}} \rightarrow P(t) = \frac{10^7}{1 + e^{-0.036\left(t - \frac{\ln(9)}{0.036}\right)}}$$

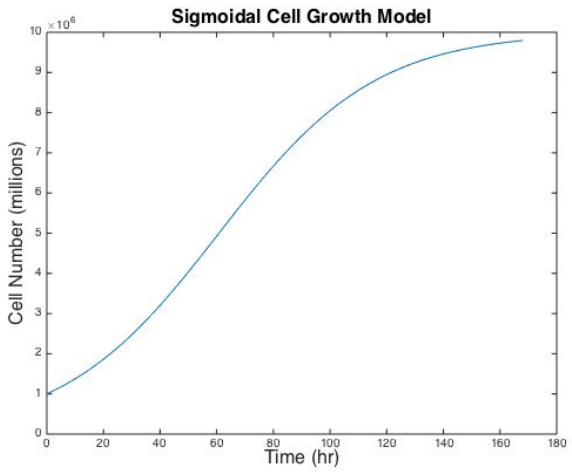

Figure 7: Sigmoidal Cell Growth model.

When applied to the diffusion model, all parameters are identical to those of the exponential growth scenario, with the exception of the forcing term having dependence on a sigmoidal, rather than exponential, growth rate. To reiterate, ρ_{max} represents the maximum population density of 10×10^9 cells/L (10 million cells/mL), while ρ_0 represents the initial cell seeding density of 5×10^8 cells/L (0.5 million cells/mL), and k is a constant derived from a previously determined HepG2 cell growth rate (approx. 0.03615).

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{dx^2} - U\frac{\rho_{max}}{1+e^{-k\left(t-\frac{\ln(\rho_{max}-\rho_0)}{k}\right)}}$$

Because of the difficulties associated with integrating the term representing sigmoidal growth and consumption, the system was solely solved numerically, with the output shown in **Figure 8**.

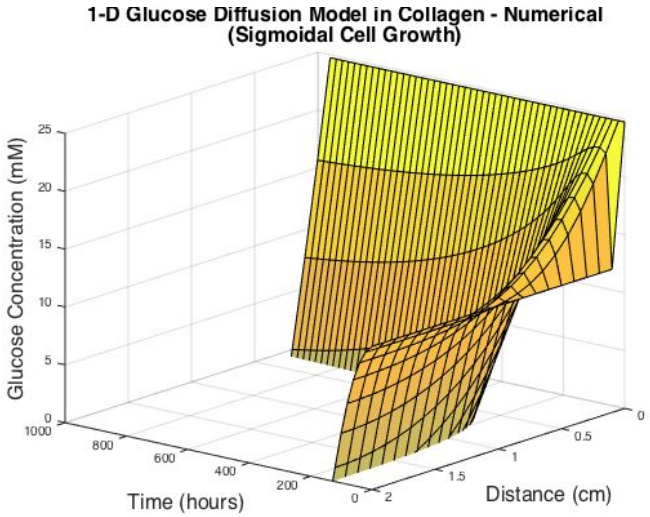

Figure 8: Numerical results of 1-D glucose diffusion model in collagen with sigmoidal cell growth

Like the exponential growth model, the concentration of glucose rapidly falls to zero. This indicates that the system is not suitable for sustaining a cell density of 10 million cells/mL. In an experimental scenario, there are several potential methods for resolving this, assuming that the

desired cell density would ideally remain uncompromised. The first is to increase the nutrient concentration flowing through the channel, which would increase the value of C_0 at x = 0. Alternatively, the material can be changed to one that allows for increased levels of diffusion.

For instance, the diffusion constant of glucose through collagen gels is ~1.3 x 10^{-6} cm^2/s, while the diffusion constant of glucose through alginate gels is nearly 6 times greater at ~7.3 x 10^{-6} cm^2/s. Since alginate gels are also biocompatible and have been used as encapsulation materials for HepG2 cells[2], we model a system using alginate gels. Again, all parameters of the system are identical to the previous case, with the exception of the increased the diffusion constant such that D = 1.3 x 10^{-6} cm^2/s → D = 7.3 x 10^{-6} cm^2/s. Results are shown in **Figure 9**.

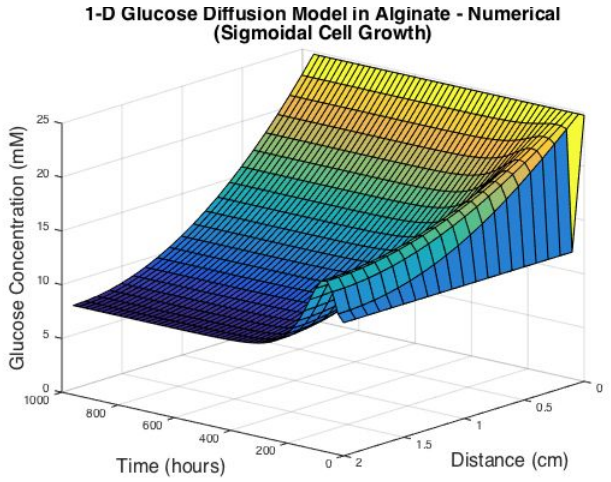

Figure 9: Numerical results of 1-D glucose diffusion model in alginate with sigmoidal cell growth

Notably, the substitution of collagen with alginate greatly mitigates the glucose shortage in the system, with the concentration stabilizing at approximately 8 mM across the matrix after 4 weeks. In this manner, the model demonstrates usefulness in assessing the potential impact of different matrix materials on cell viability, as a simple change in just the diffusion parameter was able to produce a large modification of the behavior of the entire system.

6. 2-D Glucose Diffusion in Alginate Matrices with Sigmoidal Cell Growth

Thus far, the diffusion model has been limited to one dimension for simplicity. However, in order to visualize the solution over the entire area of the matrix and explore further applications, the model can be expanded into two dimensions. Specifically, the model will be expanded to incorporate the y-dimension, running parallel to the length of the vascular channel, as modeled in **Figure 10**.

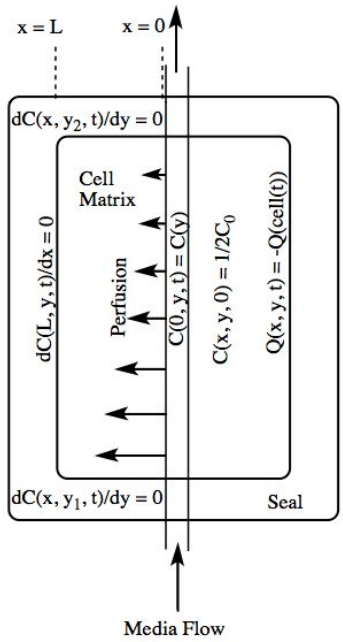

Figure 10: Initial diffusion-system setup

The equations for the two dimensional diffusion model is given below. Initially, all conditions are equivalent to the one dimensional sigmoidal growth model, with the addition of two zero flux boundary conditions to represent the sealing of the system in the y-boundaries.

$$\frac{\partial \mathrm{C}(\mathrm{x},\mathrm{y},\mathrm{t})}{\partial \mathrm{t}} = \mathrm{D}\,\frac{\partial^2 \mathrm{C}(\mathrm{x},\mathrm{y},\mathrm{t})}{\partial \mathrm{x}^2} + \mathrm{D}\,\frac{\partial^2 \mathrm{C}(\mathrm{x},\mathrm{y},\mathrm{t})}{\partial \mathrm{y}^2} + Q(t)$$

$$\frac{\partial C(x, y_1, t)}{dy} = 0 \qquad \frac{\partial C(x, y_2, t)}{dy} = 0$$

Just as was the case with the 1-Dimensional sigmoidal cell growth model, the forcing term Q(t) can be described as a product of the glucose consumption rate per cell U, and the cell seeding density ρ(t). Simplified numbers are also shown for easing approximation.

$$Q(t) = U\frac{\rho_{max}}{1 + e^{-k\left(t - \frac{\ln(\rho_{max} - \rho_0)}{k}\right)}} \rightarrow Q(t) = \frac{0.216}{1 + e^{-0.0362t - 5.65}}$$

Again, because of the integration-difficulties brought about by the forcing term, we will solely use numerical methods for finding a solution to the system, this time with the explicit two dimensional finite differences method (shown below), since the MATLAB pdepe function is not suited for more than one dimension.

$$C_{i,j}^{n+1} = C_{i,j}^{n} + \frac{D\Delta t}{\Delta x^2}\left(C_{i,j+1}^{n} - C_{i,j}^{n} + C_{i,j-1}^{n}\right) + \frac{D\Delta t}{\Delta y^2}\left(C_{i+1,j}^{n} - 2C_{i,j}^{n} + C_{i-1,j}^{n}\right) + Q_{i,j}^{n}\Delta t$$

Notably, an implicit finite differences method was initially utilized to estimate the model, which was accomplished through the adaptation of an open source MATLAB script (Copyright (c) 2012, Suraj Shankar). However, due to concerns about the time interval, this was switched with the explicit method shown above. The script still remained in use, and the final version is thus still a modification of existing code.

6.1 2-D Model of Collagen matrix with constant concentration along channel

In order to confirm that the behavior of the 2D model aligned with that of the 1D model described previously, we initially modeled glucose diffusion through both a collagen and alginate matrix with parameters identical to those of the sigmoidal model described in Section 5. That is, a constant concentration C_0 = 24.75 mM existed at x = 0, while an initial concentration $C_0/2$ = 12.375 mM existed throughout the matrix at t = 0, and zero flux conditions were present at all other x and y boundaries. A model of the collagen matrix is shown first (**Figure 11**). Because the model plots the concentration on an x-y plane in real-time, several frames are shown below, spanning from 0 - 288 hours.

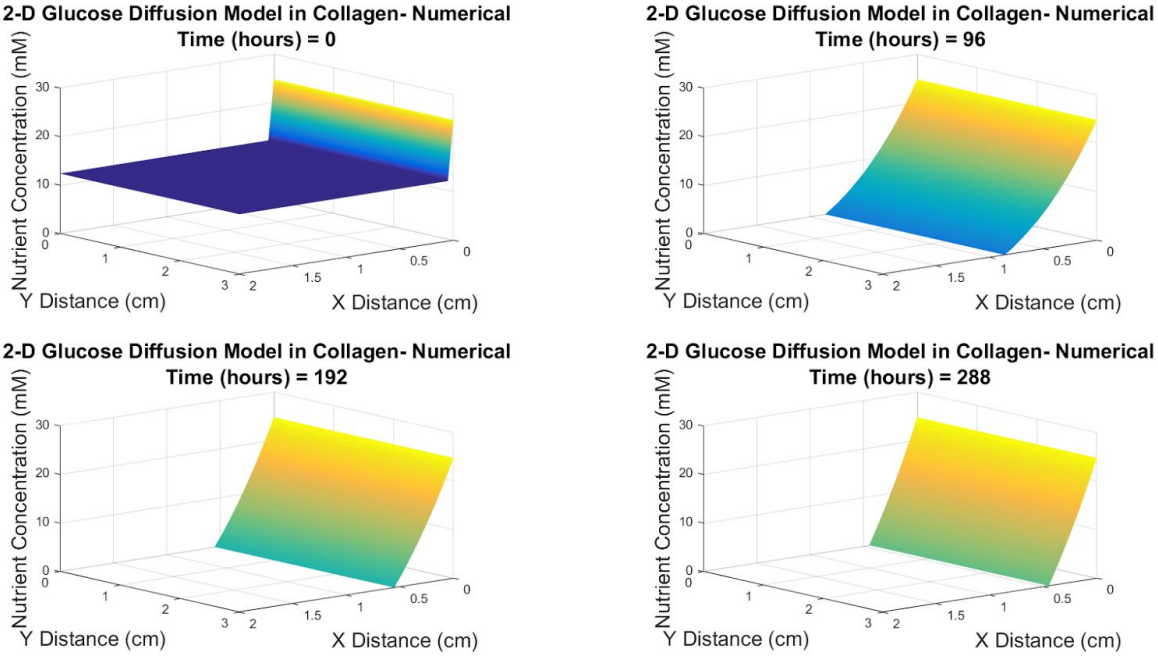

Figure 11: Numerical results of 2-D glucose diffusion model in collagen with sigmoidal cell growth at various timepoints

The results match those of the 1D model (**Figure 8**) very well, with the glucose consumption rate overpowering the rate at which diffusion from the vascular channel is able to resupply it, resulting in only the cells closest to the channel (x = 0) having an adequate glucose supply. Additionally, because the channel runs entirely in the y-direction, the glucose concentration is consistent across the entirety of the y-dimension.

6.2 2-D Model of Alginate matrix with constant concentration along channel

As described previously, the second application of the model uses alginate as a medium, and has otherwise identical conditions to the first model. The only difference, again, is that the diffusion constant (D) is changed from 1.3 x 10^{-6} cm^2/s (collagen) to 7.3 x 10^{-6} cm^2/s (alginate). Again, frames of the results from 0 to 288 hours are shown below (**Figure 12**).

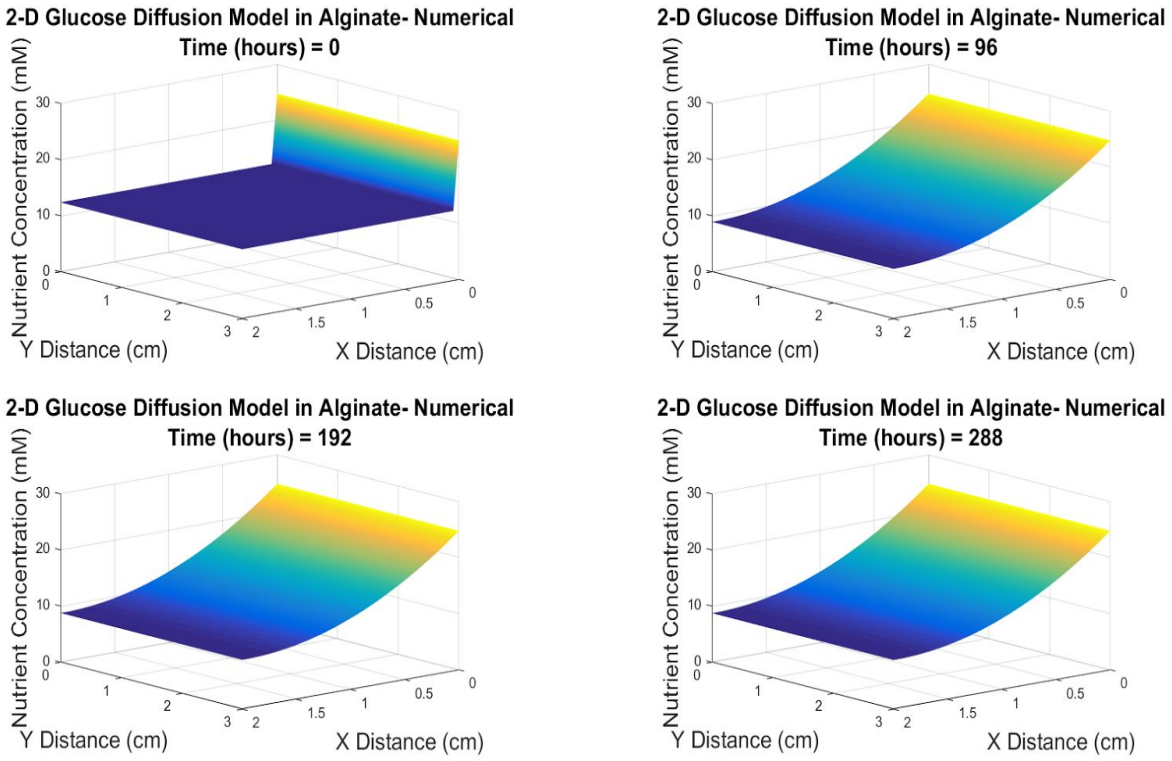

Figure 12: Numerical results of 2-D glucose diffusion model in alginate with sigmoidal cell growth at various timepoints

Just like with the 1D model (**Figure 9**) the increased diffusivity provided by the alginate matrix is able to balance the rate of glucose consumption by the sigmoidally-increasing number of cells, and a stable equilibrium is eventually reached in both the x and y dimensions.

6.3 2-D Model of Alginate matrix with linearly decreasing concentration along channel

Up until now, we have assumed that the concentration of glucose in the channel (x = 0) remains constant. In reality though, as glucose from earlier points in the channel (closer to y = 0) diffuse into the matrix, less should remain in the channel. Thus, as position in the y-dimension increases, glucose concentration in the channel should decrease. We approximate this behavior using a model that is identical to the second, with the exception of having a linearly decreasing function replacing the constant term for the concentration along the vascular channel at x = 0.

$$C(0, y, t) = C_0 \rightarrow C(0, y, t) = C_0 - 3y$$

Notably, unlike the rest of the model, the linearly decreasing function is not based on existing published results, and is was chosen arbitrarily simply to model the potential behavior. The results of this change are shown in **Figure 13**.

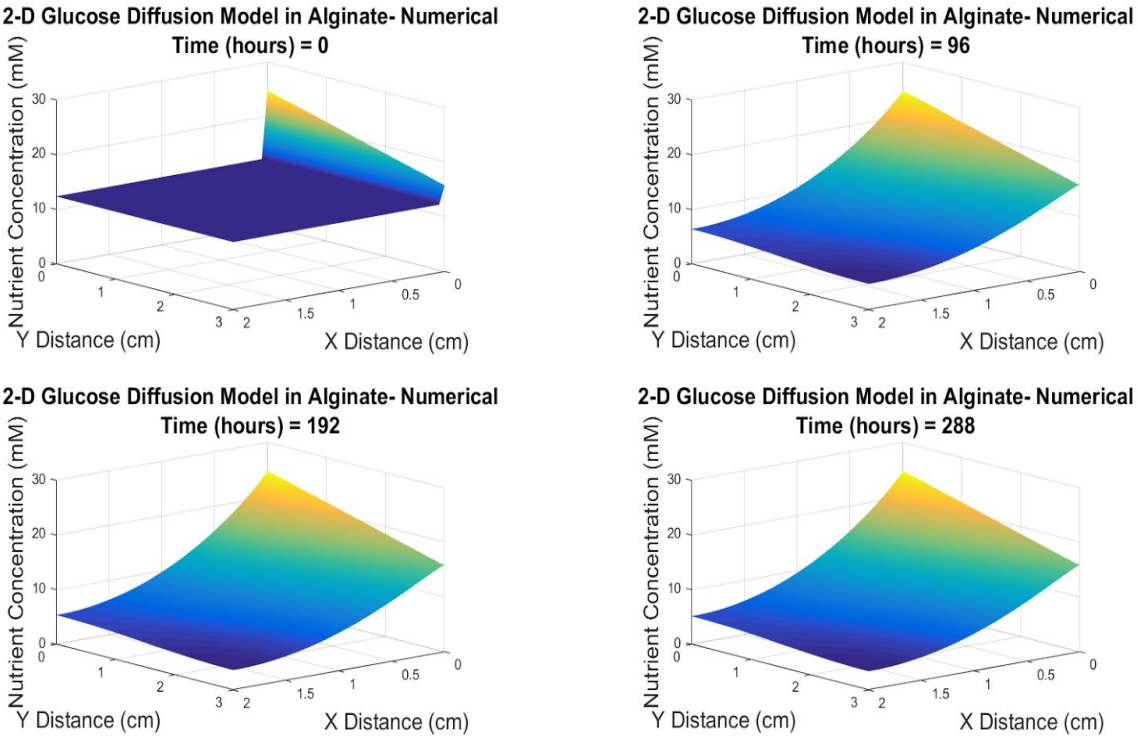

Figure 13: Numerical results of 2-D glucose diffusion model in alginate with sigmoidal cell growth and space-dependent vascular channel glucose concentration at various timepoints

The results are similar to those of the previous case. However, a decreasing glucose concentration is clearly visible along x = 0, and the result is that positions further along the y-dimension receive significantly less glucose. This results in a lower overall glucose concentration at these positions. This model only shows results up to t = 288 hours, but with increased perfusion time, it's possible that the glucose concentration at these locations may drop to zero, indicating that while cells would be able to survive with constant glucose concentration in the vascular channel, the same may not be true if the concentration decreases along the channel length.

6.3 2-D Model of Alginate matrix with linearly decreasing concentration along channel and open boundaries

The fourth and last two-dimensional model changes the model significantly by altering the zero-flux boundary conditions to zero-value boundary conditions. Thus far, we have modeled in a controlled scenario in which the matrix is perfused in a sealed container (described previously in Section 2). However, the end goal of these bioprinted vascularized scaffolds is to be implanted in tissue, where seals will not be present (**Figure 14**). Typically when implanting, the scaffold will simply be sutured in the location of interest, with the scaffold channel linked to existing blood vessels so perfusion is then maintained by the body of the host.

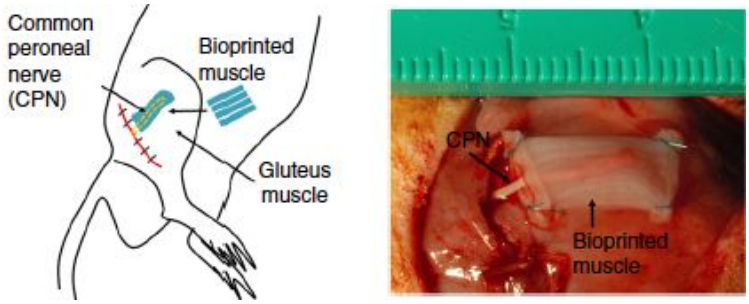

Figure 14: Implanted vascularized muscle scaffold in rat[9]

In this scenario, all existing zero-flux boundaries would be converted into zero-value boundaries, while the x = 0 boundary condition where the vascular channel exists would remain unchanged.

$$C(0,y,t) = C_0 - 3y \qquad C(L,y,t) = 0$$
$$C(x,y_1,t) = 0 \qquad C(x,y_2,t) = 0$$

As was the case with previous iterations of the model, frames are graphed spanning from 0 to 288 hours in **Figure 15**.

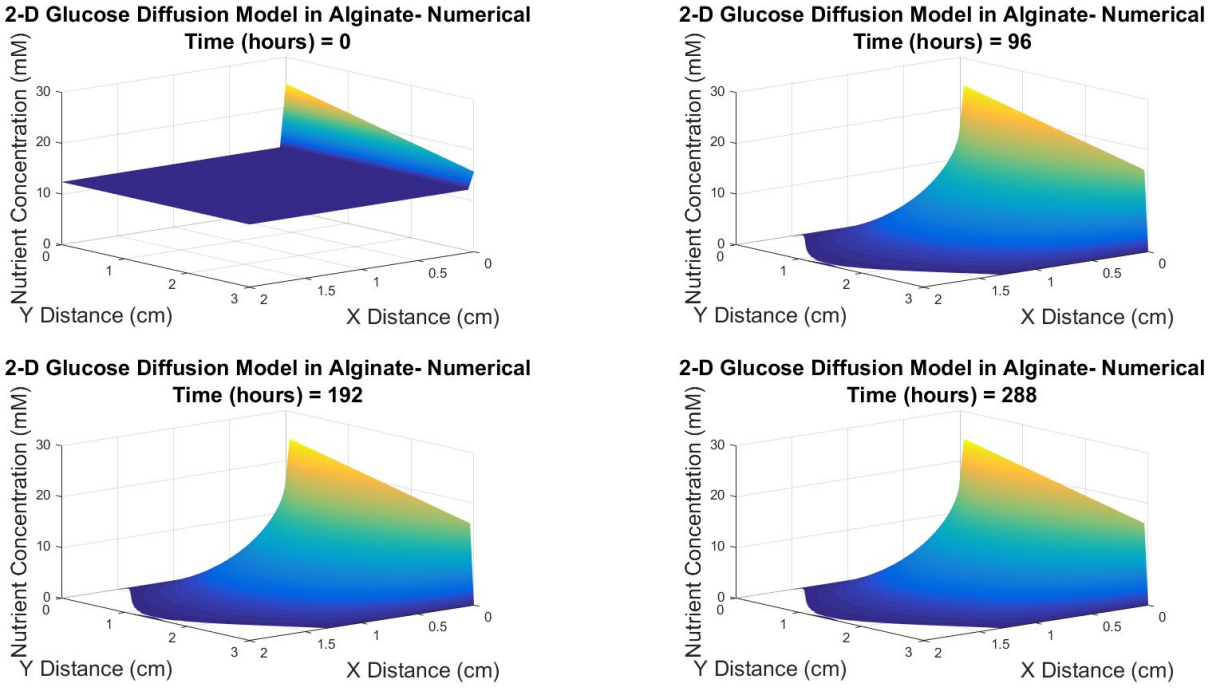

Figure 15: Numerical results of 2-D glucose diffusion model in alginate with sigmoidal cell growth, space-dependent vascular channel glucose concentration, and zero-value boundary conditions at various timepoints

As expected, several major changes are visible. At the boundaries, the rate of diffusion is not quick enough to counter the combined loss of glucose through consumption and leakage, preventing the channel from completely sustaining the layer of cells. While this may appear dire, it should be noted that although we assumed zero-value boundary conditions in this case, most tissues do not have zero glucose concentration, and as such, it's possible that the surrounding tissue itself may be able to help sustain glucose levels. However, that will be left for future modeling.

7. Conclusions and Future Considerations

It should be reiterated that the goal of this endeavor was simply to generate a model that could predict the effect of various factors on the presence of a specific nutrient in the cell. In that sense, this model could potentially be an experimental tool for researchers in the area. Thus far, we have illustrated that the model is able to incorporate a variety of parameters including

- Different rates of cell growth, resulting in different rates of consumption.
- Different matrix materials used, resulting in different diffusion constants.
- Multiple dimensions.
- Different environments, resulting in different boundary conditions.

However, in addition to these parameters explored, there are many that have not been varied. For instance, this report examines only a single cell type, the HepG2 cell line. However, there are many different cell types for which there are applications that would require the creation of seeded matrices, including myoblasts, osteoblasts, progenitors, stem cells, fibroblasts, etc., each of which would likely produce a different glucose consumption rate and growth rate. Additionally, the only nutrient explored was glucose, but in many cell types, a different nutrient, such as amino acids or vitamins, may be the limiting factor. In such cases, the diffusion constant through any given material would be altered.

Beyond this, there is also the fact that matrix thickness was not incorporated into the model, and that the model as a whole was extremely simple, consisting of a linear vascular channel through a rectangular matrix. In reality, there can be many geometries incorporated, such as multiple parallel vascular structures, or sinusoidal vascular structures[8]. Both 3D modeling and more advanced vascular structures would likely be difficult to solve for analytically, but with numerical methods, it may be possible to explore the dynamics associated with them as well.

8. References

1. Brown, Carlos V. R., Kelli H. Foulkrod, Sarah Dworaczyk, Kit Thompson, Eric Elliot, Hassie Cooper, and Ben Coopwood. “Barriers to Obtaining Family Consent for Potential Organ Donors:” *The Journal of Trauma: Injury, Infection, and Critical Care* 68, no. 2 (February 2010): 447–51. doi:10.1097/TA.0b013e3181caab8f.
2. Capone, Stephanie H., Murielle Dufresne, Mathias Rechel, Marie-José Fleury, Anne-Virginie Salsac, Patrick Paullier, Martine Daujat-Chavanieu, and Cecile Legallais. “Impact of Alginate Composition: From Bead Mechanical Properties to Encapsulated HepG2/C3A Cell Activities for In Vivo Implantation.” Edited by Thomas Claudepierre. *PLoS ONE* 8, no. 4 (April 25, 2013): e62032. doi:10.1371/journal.pone.0062032.
3. Casciari, Joseph J., Stratis V. Sotirchos, and Robert M. Sutherland. “Variations in Tumor Cell Growth Rates and Metabolism with Oxygen Concentration, Glucose Concentration, and Extracellular pH.” *Journal of Cellular Physiology* 151, no. 2 (1992): 386–94.
4. Eyting, Markus, Arne Hosemann, and Magnus Johannesson. “Can Monetary Incentives Increase Organ Donations?” *Economics Letters* 142 (May 2016): 56–58. doi:10.1016/j.econlet.2016.03.005.
5. Fukumitsu, K., H. Yagi, and A. Soto-Gutierrez. “Bioengineering in Organ Transplantation: Targeting the Liver.” *Transplantation Proceedings* 43, no. 6 (July 2011): 2137–38. doi:10.1016/j.transproceed.2011.05.014.
6. Griffith, Linda, and Gail Naughton. “Tissue Engineering- Current Challenges and Expanding Opportunities.” *Science* 295 (February 8, 2002): 1009–14.
7. Harry, I. “Nutrition Needs of Mammali Cells in Tissue Culture,” 1955. http://www.ehu.eus/biofisica/juanma/mbb_old/articulos_cientificos/eagle.pdf.
8. Kolesky, David B., Kimberly A. Homan, Mark A. Skylar-Scott, and Jennifer A. Lewis. “Three-Dimensional Bioprinting of Thick Vascularized Tissues.” *Proceedings of the National Academy of Sciences* 113, no. 12 (March 22, 2016): 3179–84. doi:10.1073/pnas.1521342113.
9. Lee, Vivian K., Diana Y. Kim, Haygan Ngo, Young Lee, Lan Seo, Seung-Schik Yoo, Peter A. Vincent, and Guohao Dai. “Creating Perfused Functional Vascular Channels Using 3D Bio-Printing Technology.” *Biomaterials* 35, no. 28 (September 2014): 8092–8102. doi:10.1016/j.biomaterials.2014.05.083.
10. Murphy, Sean V, and Anthony Atala. “3D Bioprinting of Tissues and Organs.” *Nature Biotechnology* 32, no. 8 (August 5, 2014): 773–85. doi:10.1038/nbt.2958.
11. Ohno, M., K. Motojima, T. Okano, and A. Taniguchi. “Maturation of the Extracellular Matrix and Cell Adhesion Molecules in Layered Co-Cultures of HepG2 and Endothelial Cells.” *Journal of Biochemistry* 145, no. 5 (May 1, 2009): 591–97. doi:10.1093/jb/mvp019.
12. Ott, Harald C, Thomas S Matthiesen, Saik-Kia Goh, Lauren D Black, Stefan M Kren, Theoden I Netoff, and Doris A Taylor. “Perfusion-Decellularized Matrix: Using Nature’s Platform to Engineer a Bioartificial Heart.” *Nature Medicine* 14, no. 2 (February 2008): 213–21. doi:10.1038/nm1684.
13. Ozeki, Tsuneo, and Tsuneo Natori. “The Specific Inhibition of HepG2 Cells Proliferation by Apoptosis Induced by Gabexatemesilate.” *Int. J. Clin. Exp. Pathol* 3 (2010): 710–17.
14. Rong, Zimei, Umber Cheema, and Pankaj Vadgama. “Needle Enzyme Electrode Based Glucose Diffusive Transport Measurement in a Collagen Gel and Validation of a Simulation Model.” *The Analyst* 131, no. 7 (2006): 816. doi:10.1039/b600334f.
15. Salim, Ali, Matthew Martin, Carlos Brown, Peter Rhee, Demetrios Demetriades, and Howard Belzberg. “The Effect of a Protocol of Aggressive Donor Management: Implications for the National Organ Donor Shortage:” *The Journal of Trauma: Injury, Infection, and Critical Care* 61, no. 2 (August 2006): 429–35. doi:10.1097/01.ta.0000228968.63652.c1.
16. Stanners, C. P., G. L. Eliceiri, and H. Green. “Two Types of Ribosome in Mouse–hamster Hybrid Cells.” *Nature* 230, no. 10 (1971): 52–54.
17. https://www.mathworks.com/matlabcentral/fileexchange/38088-diffusion-in-1d-and-2d

Appendix A Detailed Analytical Solution to Stable Cell Density Differential Equation

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} - Q_0$$

$$Q_0 = U\rho_0$$

Initial Condition Boundary Conditions

$$C(x,0) = \frac{C_0}{2} \qquad C(0,t) = C_0 \quad \frac{\partial C}{\partial x}(L,t) = 0$$

The partial differential equation is solved by finding a particular solution C_p and its homogeneous solution C_H. We find the particular solution C_p assuming steady state.

$$\frac{\partial C_p}{\partial t} = 0 = D\frac{\partial^2 C_p}{\partial x^2} - Q_0$$

C_p is required to satisfy the boundary conditions below.

$$C_p(0,t) = C_0 \quad \frac{dC_p}{dx}(L,t) = 0$$

Rearranging the above equation gives us

$$\frac{\partial^2 C_p}{\partial x^2} = \frac{Q_0}{D}$$

C_p is in the form as below with a and b as presuming coefficients.

$$C_p = \frac{Q_0}{2D}x^2 + ax + b$$

By applying boundary conditions to the expression of C_p, we are able to solve for a and b.

$$\begin{cases} C_p(0) = C_0 = b \\ \frac{dC_p}{dx}(L,t) = 0 = \frac{Q_0 L}{D} + a \end{cases}$$

$$\begin{cases} b = C_0 \\ a = -\frac{Q_0 L}{D} \end{cases}$$

Therefore, our steady state solution as well as particular solution is

$$C_p(x) = \frac{Q_0}{2D}x^2 - \frac{Q_0 L}{D}x + C_0$$

Now, we are going to solve for the homogeneous equation shown below.

$$\frac{\partial C_H(x,t)}{\partial t} = D\frac{\partial^2 C_H(x,t)}{\partial x^2}$$

Boundary conditions for the above equation are illustrated below.

$$C_H(0,t) = 0 \qquad \frac{\partial C_H}{\partial x}(L,t) = 0$$

By doing variable separation, we find the following expression of C_H that satisfies both boundary conditions. B_n is coefficient of the n^{th} term.

$$C_H(x,t) = \sum_{n=0}^{\infty} B_n \sin\left(\frac{2n+1}{2L}\pi x\right) e^{-D\left(\frac{2n+1}{2L}\pi\right)^2 t}$$

Full solution is expressed by the addition of C_p and C_H, which is equivalent to the following.

$$C(x,t) = C_p(x) + C_H(x,t)$$

$$C(x,t) = \frac{Q_0}{2D}x^2 - \frac{Q_0 L}{D}x + C_0 + \sum_{n=0}^{\infty} B_n \sin\left(\frac{2n+1}{2L}\pi x\right) e^{-D\left(\frac{2n+1}{2L}\pi\right)^2 t}$$

In order to find the values of B_n, the initial condition is applied to the above expression of C.

$$C(x,0) = \frac{C_0}{2} = \frac{Q_0}{2D}x^2 - \frac{Q_0 L}{D}x + C_0 + \sum_{n=0}^{\infty} B_n \sin\left(\frac{2n+1}{2L}\pi x\right)$$

$$\sum_{n=0}^{\infty} B_n \sin\left(\frac{2n+1}{2L}\pi x\right) = -\frac{Q_0}{2D}x^2 + \frac{Q_0 L}{D}x - \frac{C_0}{2}$$

B_n can be expressed as

$$B_n = \frac{2}{L}\int_0^L \left(-\frac{Q_0}{2D}x^2 + \frac{Q_0 L}{D}x - \frac{C_0}{2}\right)\sin\left(\frac{2n+1}{2L}\pi x\right)dx$$

$$= \frac{2Q_0}{LD}\left[-\int_0^L \frac{1}{2}x^2 \sin\left(\frac{2n+1}{2L}\pi x\right)dx + L\int_0^L x\sin\left(\frac{2n+1}{2L}\pi x\right)dx\right] - \frac{C_0}{L}\int_0^L \sin\left(\frac{2n+1}{2L}\pi x\right)dx$$

$$= \frac{2Q_0}{LD}\left\{\frac{4L^3}{(2n+1)^2\pi^2}\left[-\sin\left(\frac{2n+1}{2}\pi\right) + \frac{2}{(2n+1)\pi}\right] + L\left[\frac{2L}{(2n+1)\pi}\right]^2 \sin\left(\frac{2n+1}{2}\pi\right)\right\} - \frac{C_0}{L}\frac{2L}{(2n+1)\pi}$$

$$= \frac{16Q_0L^2}{D(2n+1)^3\pi^3} - \frac{2C_0}{(2n+1)\pi}$$

Therefore, the final full solution is shown as below.

$$C(x,t) = \frac{Q_0}{2D}x^2 - \frac{Q_0 L}{D}x + C_0 + \sum_{n=0}^{\infty}\left[\frac{16Q_0L^2}{D(2n+1)^3\pi^3} - \frac{2C_0}{(2n+1)\pi}\right]\sin\left(\frac{2n+1}{2L}\pi x\right) e^{-D\left(\frac{2n+1}{2L}\pi\right)^2 t}$$

Appendix B Detailed Analytical Solution to Exponential Growth Differential Equation

$$\frac{\partial C(x,t)}{\partial t} = D\frac{\partial^2 C(x,t)}{\partial x^2} + Q(t) = D\frac{\partial^2 C(x,t)}{\partial x^2} - U\rho_0 2^{kt}$$

Green's Function for the Value - Flux case:

$$G(x,t,x_0,t_0) = \sum_{n=odd}^{\infty} \frac{2}{L}\sin\left(\frac{n\pi x_0}{2L}\right)\sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t-t_0)}$$

In succession, we solve for:

- Initial Condition Term

$$\int_0^L g(x_0)G(x,t,x_0,t_0=0)dx_0$$

- Driving force term

$$+\int_0^L\int_0^t Q(x_0,t_0)G(x,t,x_0,t_0)dx_0$$

- Value boundary at x = 0 term

$$+\int_0^t C_0(t_0)D\frac{\partial}{\partial x_0}G(x,t,x_0=0,t_0)dt_0$$

- Flux boundary at x = L term

$$+\int_0^t DC'_L(t_0)G(x,t,x_0=L,t_0)dt_0$$

Solving for the Initial condition term

$$\int_0^L g(x_0)G(x,t,x_0,t_0=0)dx_0$$

$$\int_0^L \frac{C_0}{2}\sum_{n=odd}^{\infty}\frac{2}{L}\sin\left(\frac{n\pi x_0}{2L}\right)\sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t-t_0)}dx_0$$

$$\frac{C_0}{2}\sum_{n=odd}^{\infty}\frac{2}{L}\int_0^L \sin\left(\frac{n\pi x_0}{2L}\right)dx_0\,\sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t-t_0)}$$

Because $t_0 = 0$ in this case, we simplify to

$$\frac{C_0}{2}\sum_{n=odd}^{\infty}\frac{4}{n\pi}\left(1-cos\left(\frac{n\pi}{2}\right)\right)sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t)}$$

Here, $\cos(n\pi/2) = 0$ when n is odd, and as a result, the cosine term can be removed

$$\frac{C_0}{2}\sum_{n=odd}^{\infty}\frac{4}{n\pi}sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t)}$$

Solving for the Driving force term

$$\int_0^L\int_0^t -U\rho_0 2^{kt_0}\sum_{n=odd}^{\infty}\frac{2}{L}sin\left(\frac{n\pi x_0}{2L}\right)sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t-t_0)}\,dx_0dt_0$$

$$-U\rho_0\frac{2}{L}\sum_{n=odd}^{\infty}sin\left(\frac{n\pi x}{2L}\right)\int_0^L sin\left(\frac{n\pi x}{2L}\right)dx_0\int_0^t 2^{kt_0}\,e^{-D\left(\frac{n\pi}{2L}\right)^2(t-t_0)}dt_0$$

$$-U\rho_0\frac{2}{L}\sum_{n=odd}^{\infty}sin\left(\frac{n\pi x}{2L}\right)\left(\frac{2L}{n\pi}\right)\left(1-\cos\left(\frac{n\pi}{2}\right)\right)4L^2\frac{2^{kt}-e^{-D\left(\frac{n\pi}{2L}\right)^2t}}{D(n\pi)^2+kL^2 4\ln(2)}$$

$$-U\rho_0\frac{16L^2}{n\pi}\sum_{n=odd}^{\infty}sin\left(\frac{n\pi x}{2L}\right)\frac{2^{kt}-e^{-D\left(\frac{n\pi}{2L}\right)^2t}}{D(n\pi)^2+kL^2 4\ln(2)}$$

Solving for the Value boundary at x = 0

$$\int_0^t C_0 D\frac{\partial}{\partial x_0}\sum_{n=odd}^{\infty}\frac{2}{L}sin\left(\frac{n\pi x_0}{2L}\right)sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t-t_0)}\,dt_0$$

$$C_0 D\frac{2}{L}\sum_{n=odd}^{\infty}\frac{\partial}{\partial x_0}sin\left(\frac{n\pi x_0}{2L}\right)sin\left(\frac{n\pi x}{2L}\right)\int_0^t e^{-D\left(\frac{n\pi}{2L}\right)^2(t-t_0)}\,dt_0$$

$$C_0 D\frac{2}{L}\sum_{n=odd}^{\infty}\frac{n\pi}{2L}sin\left(\frac{n\pi x}{2L}\right)\left(\frac{2L}{n\pi}\right)^2\left(\frac{1}{D}\right)\left(1-e^{-D\left(\frac{n\pi}{2L}\right)^2(t)}\right)$$

$$C_0\sum_{n=odd}^{\infty}\frac{4}{n\pi}sin\left(\frac{n\pi x}{2L}\right)\left(1-e^{-D\left(\frac{n\pi}{2L}\right)^2(t)}\right)$$

Solving for the Flux boundary at x = L

This term goes to 0 because the flux at x = L, shown as C'_L is 0.

$$+\int_0^t DC'_L(t_0)G(x,t,x_0=L,t_0)dt_0$$

Full solution (Summation of all previous terms)

$$C(x,t)=\frac{C_0}{2}\sum_{n=odd}^{\infty}\frac{4}{n\pi}sin\left(\frac{n\pi x}{2L}\right)e^{-D\left(\frac{n\pi}{2L}\right)^2(t)}-U\rho_0\frac{16L^2}{n\pi}\sum_{n=odd}^{\infty}sin\left(\frac{n\pi x}{2L}\right)\frac{2^{kt}-e^{-D\left(\frac{n\pi}{2L}\right)^2 t}}{D(n\pi)^2+kL^2 4\,ln(2)}$$

$$+C_0\sum_{n=odd}^{\infty}\frac{4}{n\pi}sin\left(\frac{n\pi x}{2L}\right)\left(1-e^{-D\left(\frac{n\pi}{2L}\right)^2(t)}\right)$$

Appendix C MATLAB Code for 1-D Stable Cell Density Scenario

```
%% Basic Coefficients
T = 960;
L = 2;

global D C0 R k
D = 1.3e-6*3600;
C0 = 24.75;
R = 2.16e-8 * 1.5e6;
k = 0.03615;

% domain
dx = 0.02; % step size in x dimension
dt = 9.6;  % step size in t dimension

xmesh = 0:dx:L; % domain in x
tmesh = 0:dt:T;  % domain in t

nx = length(xmesh); % number of points in x dimension
nt = length(tmesh); % number of points in t dimension

%% PDEPE Function
sol_pde = pdepe(0,@nogrowth,@icexp,@bcexp,xmesh,tmesh);

figure(1)
surf(tmesh,xmesh(1:nx),sol_pde(:,1:nx)','EdgeColor','none')
axis([0,T,0.01,2,0,inf])
title({'1-D Glucose Diffusion Model in Collagen - Numerical','(Stable Cell
Density)'},'Fontsize',16)
xlabel('Time (hours)','Fontsize',16)
ylabel('Distance (cm)','Fontsize',16)
zlabel('Glucose Concentration (mM)','Fontsize',16)
%% Analytical Solution
sol_anal = zeros(nt,nx);
N = 200;

for t=1:nt
    for x=1:nx
        sol_anal(t,x) = R/2/D*xmesh(x)^2 - R*L/D*xmesh(x) + C0;
        for n=0:N-1
                                                sol_anal(t,x)    =    sol_anal(t,x)    +
(16*R*L^2/(D*(2*n+1)^3*pi^3)-2*C0/((2*n+1)*pi))*...
                sin((2*n+1)*pi*xmesh(x)/2/L)*exp(-D*((2*n+1)*pi/2/L)^2*tmesh(t));
        end
    end
end

figure(2)
surf(tmesh,xmesh(1:nx),sol_anal(:,1:nx)', 'EdgeColor','none')
axis([0,T,0.01,2,0,inf])
title({'1-D Glucose Diffusion Model in Collagen - Analytical','(Stable Cell
Density)'},'Fontsize',16)
xlabel('Time (hours)','Fontsize',16)
ylabel('Distance (cm)','Fontsize',16)
zlabel('Glucose Concentration (mM)','Fontsize',16)
```

Appendix D MATLAB Code for 1-D Exponential Cell Growth Scenario

```
%%% Exponential Growth Scenario

%% Define constants
global D L tmax C_0 C_m R k
% Initial glucose concentration in matrix
C_0_su = 12.375; % mmol/L
C_0 = C_0_su * .001; % mmol/cm^3 - converted units
% Glucose concentration in perfused media
C_m_su = 24.75; %mmol/L
C_m = C_m_su * .001; % mmol/cm^3 - converted units

% Diffusion of glucose in collagen gell
D_su = 1.3 * 10^(-6); % cm^2/sec
D = D_su * 60 * 60; %cm^2/hr - unit conversion
% Distance from vascular channel to wall
L = 2; % cm
tmax = 960; % hours
% Seeding density
S = 0.5 * 10^6; %cells/cm^3
% uptake rate per cell
Upc_su = 1.8 * 10^-17; % mol/cell/s
Upc = Upc_su * 3.6*10^6; %mmol/cell/hr
% initial uptake rate per vol
R = S * Upc; %mmol/cm^3/hr
% Exponent rate constant in R*2^(k*t)
k = 0.03615;

% Domain
dx = L/100; % step size in x dimension
dt = tmax/100; % step size in t dimension
xmesh = 0:dx:L; % domain in x
tmesh = 0:dt:tmax; % domain in t
nx = length(xmesh); % number of points in x dimension
nt = length(tmesh); % number of points in t dimension
[xx,tt] = meshgrid(xmesh, tmesh); % mesh for calculation comfort
% Analytical solution

% % plot exponential model
% figure
% plot(tmesh,Upc * 2 .^ (k*tmesh))

sol_anal = zeros(size(xx));

% Non-zero odd terms
count_max = 1000;

for count = 0:count_max
    n = 2*count + 1;
  icterm = C_0 * 4/(n*pi()).* sin(n*pi().*xx/(2*L)) .* exp((-1)*D*(n*pi()/(2*L))^2 .*
tt);
  drterm = (-1) * R * 16 * L^2 / (n*pi()) * ...
  sin(n*pi()*xx/(2*L)).* (2.^(k*tt) - exp((-1)*D*(n*pi()/(2*L))^2 .* tt)) / ...
  (D*(n*pi())^2+k*L^2*log(16));
  uLterm = C_m * (4/(n*pi())) * ...
  sin(n*pi()*xx/(2*L)) .* (1-exp((-1)*D*(n*pi()/(2*L))^2.* tt));
  sol_anal = sol_anal + icterm + drterm + uLterm;
```

```
end

% plot analytical solution
figure ()
surf(tmesh,xmesh,sol_anal'*1000, 'EdgeColor','none')
% axis([0,tmax,0.1,2,0,inf])
% view(-140,20)
axis([0,200,0.1,2,0,inf])
view(60,40)
caxis([0,inf])
tit_An = {'1-D Glucose Diffusion Model in Collagen - Analytical', '(Exponential Cell
Growth)'};
title(tit_An,'fontsize',16)
xlabel('Time (hours)','fontsize',16)
ylabel('Distance (cm)','fontsize',16)
zlabel('Glucose Concentration (mM)','fontsize',16)

% solution using Matlab's built in "pdepe"
sol_pdepe = pdepe(0,@pdefun,@ic,@bc,xmesh,tmesh);
figure()
surf(tmesh,xmesh,sol_pdepe'*1000,'EdgeColor','none')
% axis([0,tmax,0.1,2,0,inf])
% view(-140,20)
axis([0,200,0.1,2,0,inf])
view(60,40)
caxis([0,inf])
tit_Num = {'1-D Glucose Diffusion Model in Collagen - Numerical', '(Exponential Cell
Growth)'};
title(tit_Num,'fontsize',16)
xlabel('Time (hours)','fontsize',16)
ylabel('Distance (cm)','fontsize',16)
zlabel('Glucose Concentration (mM)','fontsize',16)

% ASSOCIATED FUNCTIONS
function [c, f, s] = pdefun(x, t, u, DuDx)
% PDE coefficients functions
global D R k
c = 1; % leave as 1 - could be 1/D if D not included in f
f = D * DuDx; % diffusion
s = (-1) * R*2^(k*t); % driving term
end
function u0 = ic(x)
% Initial conditions function
global C_0
u0 = C_0;
end
function [pL, qL, pR, qR] = bc(xL, uL, xR, uR, t)
% Boundary conditions function
% In the form of general mixed BC (Robin BC):
% Left: pL + qL * f(x,t,u,du/dx) = 0
% Right: pR + qR * f(x,t,u,du/dx) = 0
global C_m
% left boundary condition constant value at media concentration
pL = uL - C_m;
qL = 0; % if flux term, enter 1, else 0
% right boundary condition flux = 0
pR = 0;
qR = 1; % if flux term, enter 1, else 0
end
```

Appendix E MATLAB Code for 1-D Sigmoidal Cell Growth Scenario

```
function sigmoidal_model_II

% diffusion constant
global D
D = 7.3*10^-6*3600; % Diffusion Constant in cm^2/hr
%%% TURN TO -140 and 20 %%%

% domain
xmesh = 0:0.1:2; % domain in x (cm)
tmesh = 0:20:960;  % domain in t (hr)

% solution using Matlab's built in "pdepe"
sol_pdepe = pdepe(0,@pdefun,@ic,@bc,xmesh,tmesh);

figure(2)
surf(tmesh,xmesh,sol_pdepe')
title({'1-D Glucose Diffusion Model in Alginate - Numerical'; '(Sigmoidal Cell
Growth)'}, 'fontsize', 16)
xlabel('Time (hours)', 'fontsize', 16)
ylabel('Distance (cm)', 'fontsize', 16)
zlabel('Glucose Concentration (mM)', 'fontsize', 16)
zlim([0 25]);

% function definitions for pdepe:
% --------------------------------------------------------------
function [c, f, s] = pdefun(x, t, u, DuDx)
% PDE coefficients functions

global D
c = 1;
f = D * DuDx; % diffusion

C0 = 1*10^6;
Cn = 10*10^6;
k = 0.03615;
a = log(Cn/C0-1)/k;

s = -6.48*10^-11/(3/1000)*(Cn/(1+exp(-k*(t-a)))); % nonzero driving term

% --------------------------------------------------------------
function u0 = ic(x)
% Initial conditions function

u0 = 12.375; % Initial Condition Uniformity

% --------------------------------------------------------------
function [pl, ql, pr, qr] = bc(xl, ul, xr, ur, t)
% Boundary conditions function

pl = ul-24.75; % FLUX of ZERO at Left Boundary
ql = 0;  % Indicates Flux at Left Boundary Exists
pr = 0; % Value of 5.5 at Right Boundary
qr = 1;  % Indicates no Flux Condition at Right Boundary
```

Appendix F MATLAB Codes for 2-D Sigmoidal Cell Growth Scenario

Section 1: Original open source script used as foundation for model


```
% Simulating the 2-D Diffusion equation by the Finite Difference
...Method
% Numerical scheme used is a first order upwind in time and a second
...order central difference in space (Implicit and Explicit)

%%
%Specifying parameters
nx=40;                          %Number of steps in space(x)
ny=50;                          %Number of steps in space(y)
nt=30;                          %Number of time steps
dt=0.01;                        %Width of each time step
dx=2/(nx-1);                    %Width of space step(x)
dy=2/(ny-1);                    %Width of space step(y)
x=0:dx:2;                       %Range of x(0,2) and specifying the grid points
y=0:dy:2;                       %Range of y(0,2) and specifying the grid points
u=zeros(nx,ny);                 %Preallocating u
```

```
un=zeros(nx,ny);                    %Preallocating un
vis=0.1;                            %Diffusion coefficient/viscocity
UW=0;                               %x=0 Dirichlet B.C
UE=0;                               %x=L Dirichlet B.C
US=0;                               %y=0 Dirichlet B.C
UN=0;                               %y=L Dirichlet B.C
UnW=0;                              %x=0 Neumann B.C (du/dn=UnW)
UnE=0;                              %x=L Neumann B.C (du/dn=UnE)
UnS=0;                              %y=0 Neumann B.C (du/dn=UnS)
UnN=0;                              %y=L Neumann B.C (du/dn=UnN)

%%
%Initial Conditions
for i=1:nx
      for j=1:ny
      if ((1<=y(j))&&(y(j)<=1.5)&&(1<=x(i))&&(x(i)<=1.5))
            u(i,j)=2;
      else
            u(i,j)=0;
      end
      end
end

%%
%B.C vector
bc=zeros(nx-2,ny-2);
bc(1,:)=UW/dx^2; bc(nx-2,:)=UE/dx^2;  %Dirichlet B.Cs
bc(:,1)=US/dy^2; bc(:,ny-2)=UN/dy^2;  %Dirichlet B.Cs
%bc(1,:)=-UnW/dx; bc(nx-2,:)=UnE/dx;  %Neumann B.Cs
%bc(:,1)=-UnS/dy; bc(:,nx-2)=UnN/dy;  %Neumann B.Cs
%B.Cs at the corners:
bc(1,1)=UW/dx^2+US/dy^2; bc(nx-2,1)=UE/dx^2+US/dy^2;
bc(1,ny-2)=UW/dx^2+UN/dy^2; bc(nx-2,ny-2)=UE/dx^2+UN/dy^2;
bc=vis*dt*bc;

%Calculating the coefficient matrix for the implicit scheme
Ex=sparse(2:nx-2,1:nx-3,1,nx-2,nx-2);
Ax=Ex+Ex'-2*speye(nx-2);          %Dirichlet B.Cs
%Ax(1,1)=-1; Ax(nx-2,nx-2)=-1;  %Neumann B.Cs
Ey=sparse(2:ny-2,1:ny-3,1,ny-2,ny-2);
Ay=Ey+Ey'-2*speye(ny-2);          %Dirichlet B.Cs
%Ay(1,1)=-1; Ay(ny-2,ny-2)=-1;  %Neumann B.Cs
A=kron(Ay/dy^2,speye(nx-2))+kron(speye(ny-2),Ax/dx^2);
D=speye((nx-2)*(ny-2))-vis*dt*A;

%%
%Calculating the field variable for each time step
i=2:nx-1;
j=2:ny-1;
for it=0:nt
      un=u;
      h=surf(x,y,u','EdgeColor','none');      %plotting the field variable
      shading interp
      axis ([0 2 0 2 0 2])
      title({['2-D   Diffusion   with   {\nu}   =   ',num2str(vis)];['time   (\itt)   =
',num2str(it*dt)]})
      xlabel('Spatial co-ordinate (x) \rightarrow')
      ylabel('{\leftarrow} Spatial co-ordinate (y)')
      zlabel('Transport property profile (u) \rightarrow')
      drawnow;
      refreshdata(h)
```

```
        %Uncomment as necessary
        %Implicit method:
        U=un;U(1,:)=[];U(end,:)=[];U(:,1)=[];U(:,end)=[];
        U=reshape(U+bc,[],1);
        U=D\U;
        U=reshape(U,nx-2,ny-2);
        u(2:nx-1,2:ny-1)=U;
        %Boundary conditions
        %Dirichlet:
        u(1,:)=UW;
        u(nx,:)=UE;
        u(:,1)=US;
        u(:,ny)=UN;
        %Neumann:
        %u(1,:)=u(2,:)-UnW*dx;
        %u(nx,:)=u(nx-1,:)+UnE*dx;
        %u(:,1)=u(:,2)-UnS*dy;
        %u(:,ny)=u(:,ny-1)+UnN*dy;
        %}
        %Explicit method:
        %{

u(i,j)=un(i,j)+(vis*dt*(un(i+1,j)-2*un(i,j)+un(i-1,j))/(dx*dx))+(vis*dt*(un(i,j+1)-2*u
n(i,j)+un(i,j-1))/(dy*dy));
        %Boundary conditions
        %Dirichlet:
        u(1,:)=UW;
        u(nx,:)=UE;
        u(:,1)=US;
        u(:,ny)=UN;
        %Neumann:
        %u(1,:)=u(2,:)-UnW*dx;
        %u(nx,:)=u(nx-1,:)+UnE*dx;
        %u(:,1)=u(:,2)-UnS*dy;
        %u(:,ny)=u(:,ny-1)+UnN*dy;
        %}
End
```

Section 2: Modified script based on section 1, much of it is identical

```
% Modified script
clear
close all
%%
%Specifying parameters
nx=40;                          %Number of steps in space(x)
ny=50;                          %Number of steps in space(y)
nt=1440;                        %Number of time steps
dt=0.2;                         %Width of each time step
dx=2/(nx-1);                    %Width of space step(x)
dy=3/(ny-1);                    %Width of space step(y)
x=0:dx:2;                       %Range of x(0,2) and specifying the grid points
y=0:dy:3;                       %Range of y(0,2) and specifying the grid points
u=zeros(nx,ny);                 %Preallocating u
un=zeros(nx,ny);                %Preallocating un
vis=1.3 * 10^-6*3600;                       %Diffusion coefficient/viscocity
%vis=7.3 * 10^-6*3600;

UW=24.75;                       %x=0 Dirichlet B.C
UE=0;                           %x=L Dirichlet B.C
US=0;                           %y=0 Dirichlet B.C
UN=0;                           %y=L Dirichlet B.C
UnW=0;                          %x=0 Neumann B.C (du/dn=UnW)
UnE=0;                          %x=L Neumann B.C (du/dn=UnE)
UnS=0;                          %y=0 Neumann B.C (du/dn=UnS)
UnN=0;                          %y=L Neumann B.C (du/dn=UnN)

%%
%Initial Conditions
for i=1:nx
       for j=1:ny

              u(i,j)=12.375;

       end
end
u(1,:)=24.75;
%u(1,:)=24.75-y*3;
%u(1,:)=24.75*exp(-y/3);


%B.C vector
bc=zeros(nx-2,ny-2);
bc(1,:)=UW/dx^2; bc(nx-2,:)=UE/dx^2;  %Dirichlet B.Cs
bc(:,1)=US/dy^2; bc(:,ny-2)=UN/dy^2;  %Dirichlet B.Cs
%bc(1,:)=-UnW/dx; bc(nx-2,:)=UnE/dx;  %Neumann B.Cs
%bc(:,1)=-UnS/dy; bc(:,nx-2)=UnN/dy;  %Neumann B.Cs
%B.Cs at the corners:
bc(1,1)=UW/dx^2+US/dy^2; bc(nx-2,1)=UE/dx^2+US/dy^2;
bc(1,ny-2)=UW/dx^2+UN/dy^2; bc(nx-2,ny-2)=UE/dx^2+UN/dy^2;
bc=vis*dt*bc;

%Calculating the field variable for each time step
i=2:nx-1;
j=2:ny-1;

v = VideoWriter('Alginatesuccessnew.avi');

open(v)
```

```
for it=0:nt
       un=u;
       h=surf(x,y,u','EdgeColor','none');      %plotting the field variable
       shading interp
       view(135,30)
       axis ([0 2 0 3 0 30])

       title({['2-D      Diffusion      with      collagen'];['time      (\itt)      =
',num2str(floor(it*dt))]})

       xlabel('Spatial co-ordinate (x) \rightarrow')
       ylabel('{\leftarrow} Spatial co-ordinate (y)')
       zlabel('Transport property profile (u) \rightarrow')
       drawnow;
       refreshdata(h)
       F(it+1) = getframe(gcf);
       if rem(it,8)==0
       writeVideo(v,F(it+1))
       end
       %Explicit method:


u(i,j)=un(i,j)+((vis*dt*(un(i+1,j)-2*un(i,j)+un(i-1,j)))/(dx*dx))-dt*(6.48*10^-11/0.00
3)*(10*10^6/(1+exp(-0.03615*it*dt-log(9/0.03615))))+((vis*dt*(un(i,j+1)-2*un(i,j)+un(i
,j-1)))/(dy*dy));
       %Boundary conditions
       %Dirichlet:
       %u(1,:)=24.75-y*3;
       %u(1,:)=24.75*exp(-y/4);
       u(1,:)=UW;
       %u(nx,:)=UE;
       %u(:,1)=US;
       %u(:,ny)=UN;
       %Neumann:
       %u(1,:)=u(2,:)-UnW*dx;
       u(nx,:)=u(nx-1,:)+UnE*dx;
       u(:,1)=u(:,2)-UnS*dy;
       u(:,ny)=u(:,ny-1)+UnN*dy;

end
close(v)
```