Improved Methods for Agent-Based Modeling of Infarct Healing

A

Thesis

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

---

in partial fulfillment
of the requirements for the degree

Master of Science

by

Arlynn Celeste Baker

December 2020

APPROVAL SHEET

This

Thesis

is submitted in partial fulfillment of the requirements
for the degree of

Master of Science

Author: Arlynn Celeste Baker

This Thesis has been read and approved by the examing committee:

Advisor: Jeffrey Holmes

Advisor:

Committee Member: Shayn Peirce-Cottler

Committee Member: Matthew Panzer

Committee Member: Christopher Highley

Committee Member:

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

December 2020

IMPROVED METHODS FOR AGENT-BASED MODELING OF INFARCT HEALING

A Dissertation
Presented to
The Academic Faculty

By

Arlynn C. Baker

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Engineering
Department of Biomedical Engineering

University of Virginia

Oct 29th 2020

IMPROVED METHODS FOR AGENT-BASED MODELING OF INFARCT HEALING

Thesis committee:

Dr. Jeffrey Holmes
Department of Biomedical
Engineering
University of Virginia

Dr. Christopher Highley
Department of Biomedical
Engineering
University of Virginia

Dr. Shayn Peirce-Cottler
Department of Biomedical
Engineering
University of Virginia

Dr. Matthew Panzer
Department of Mechanical
and Aerospace Engineering
University of Virginia

Date approved: Oct 29th, 2020

ACKNOWLEDGMENTS

I would like to thank the members of my thesis committee for their help in the preparation of this work. Particularly to my advisor Jeff - thank you for this opportunity and for helping me pull this thesis together at the last minute.

I would also like to thank Ms. Michaela Rikard for allowing me to adapt her finite difference diffusion method implemented in MATLAB, which is used in the third chapter of this text.

Special thanks are due to the friends and colleagues who made this work possible. To my lab mates, whose constant banter and camaraderie kept me laughing for the last two years, thank you. To my roommate, Grace, thank you for every late-night cookie dough infused study session. To my parents, thank you for prioritizing my education before I was able to. Finally, to my fiance, Blake, thank you for your unending support and love and for challenging me to do my best always.

The author gratefully acknowledges the support for this work by the National Heart, Lung, and Blood Institute R01 HL-116449 and R01 HL-137755.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ABM agent-based model

ECM extracellular matrix

FDM finite difference method

FEM finite element method

GUI graphical user interface

IRF impulse response filter

LV left ventricle

MAD Mean absolute difference

MI myocardial infarction

MVA mean vector angle

MVL mean vector length

PDE partial differential equation

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Significance of Myocardial Infarction

Over one million people experience a heart attack each year in the US. The development of numerous treatments for acute myocardial infarction (MI), such as the use of thrombolytics and stenting, have pushed the survival rate of patients that experience MI to 95%. Despite this high survival rate, post-infarction scar tissue decreases cardiac function, producing long-term health impacts in survivors. As a result, research has turned towards investigating the formation and maturation of infarct scar [1]. However, one of the biggest obstacles to understanding and perturbing scar formation is the lack of relevant models. *In vivo* models are time-consuming, expensive, and often difficult to control between subjects making them less ideal for exploratory research. On the other hand, *in vitro* models which are easier to standardize and may require fewer resources, often fail to recapitulate the dynamics and complexity of infarct healing [2]. Computational models, which are fast, inexpensive, highly controlled, and offer tunable complexity, are uniquely positioned to fill the gap in models of scar formation. Therefore, the following research seeks to advance our understanding of scar formation and ability to assess treatments which address MI scar, by building an efficient and flexible computational model incorporating the most up to date experimental data on infarct scar collagen alignment.

1.2 Background

1.2.1 Myocardial Infarction *In Vivo*

Myocardial infarction occurs when the blood supply to a part of the heart is obstructed, starving myocytes of oxygen, and eventually causing their death. This kick starts the inflammatory phase of infarct healing, lasting approximately one week, where cell death signals inflammatory cells to enter the infarct where they resorb necrotic material and form granulation tissue which temporarily maintains the infarct's structural integrity [3]. Immune cells involved in this acute response also release a host of cytokines, chemokines, growth factors, and hormones [4]. Resident fibroblasts in the surrounding tissue respond to these inflammatory signaling molecules by migrating into the infarct where they begin the fibrotic phase of healing. Once inside, fibroblasts proliferate and differentiate into various fibroblast sub-types, the most abundant of which being myofibroblasts marked by increased expression of α-smooth muscle actin. These myofibroblasts begin aggressively depositing extracellular matrix (ECM) to form scar tissue over the next few weeks [5]. After infarct scar collagen density stabilizes, the final remodeling phase of healing can begin. During this months-long phase, myofibroblast populations recede slightly and begin to cross-link collagens and incorporate them into fibrils [6].

At the organ level, acute infarction divides the heart into two mechanically distinct regions: actively contracting healthy myocardium and passively stretching non-contractile infarcted tissue. This loss of contractile tissue results in significantly reduced pump function [7]. Within seconds to minutes after MI, myocyte contractility, heart rate, and arterial resistance change to help return cardiac output and blood pressure to baseline. Over time, the demand that these compensation mechanisms put on the heart cause it to undergo remodeling. This remodeling often leads to eccentric hypertrophy, a form of heart failure characterized by a progressive thinning and dilation of one or more of the chambers of the heart. Current systemic treatments for MI induced heart failure, such as beta-adrenergic blockers and angiotensin-converting enzyme inhibitors, work to modulate the body's response to decreased cardiac function and are an important part of treatments to prevent and delay heart failure [8, 9]. However, these treatments address the body's compensation mechanisms for mechanical dysfunction rather than the cause, the infarct itself.

1.2.2 Infarct Mechanics and Matrix Alignment

Fibroblasts create and remodel ECM via production and reorganization of collagen. Fibroblasts have an elongated morphology and deposit collagen fibrils parallel to their major axis [10, 11].

Thus, the alignment of fibroblasts drives the alignment of the ECM they form. Our group has shown that in infarct scar, fibroblasts on average align in the direction of the greatest stretch. The wall of the heart is stretched in both the circumferential and longitudinal directions by the pressure within the heart cavity. The balance of these two stretches, which varies across the heart, determines the direction of the greatest stretch and therefore the infarct alignment. For example, infarcts near the equator of the left ventricle (LV) are aligned to match primarily circumferential stretches, whereas infarcts near the apex of the LV are aligned randomly due to equal stretches in both directions [12]. Our research group has also shown that scar collagen orientation impacts the function of the heart after MI [13].

1.2.3 Agent-Based Model of Infarct Healing

Agent-based models (ABMs) provide a unique platform to investigate the orientation of infarct collagen during scar formation. ABMs have been used in the literature for many years to model similar biological phenomena including wound healing [14], tumor growth [15], and developmental processes like structural patterning [16]. One of the primary benefits of ABMs, is that can be built, tuned, and validated using existing data. For example, agent parameters and interactions are built from information obtained from *in vitro and ex vivo* experiments about how a particular cell type behaves and the characteristics of their environment. Once built, these models can be validated with *in vivo* data.

Our group previously built an ABM simulating MI scar formation [17]. Briefly, this ABM simulates fibroblasts as circular agents in a 2D environment representing a mid-wall section of the LV. During model initiation, fibroblast agents are seeded with a random alignment and location throughout the simulation space. The model updates every thirty minutes of simulated time allowing each agent individually to migrate; undergo apoptosis or mitosis; and deposit, degrade, and rotate collagen fibers according to external cues. These external factors include the structural alignment of the collagen and other matrix components, mechanical stretch created by the beating of the heart, and chemokines produced by the necrotic infarcted tissue and the body's immune response.

To simulate infarction, all agents within a central radius of the simulation space are deleted, and the mechanical and chemokine cues in and surrounding the infarct are altered to reflect the presence of a non-contractile and inflamed infarcted area.

This infarct healing ABM can match data from several infarct studies our group has conducted. The weights of the mechanical and chemokine cues were tuned to fit the mean vector length (MVL), mean vector angle (MVA), and area fraction of scar collagen from cryoinfarcts created at the apex and equator of the LV in adult rats at three weeks post-infarction [12]. The model predicted that longitudinal loading would produce longitudinally aligned collagen at six weeks after infarction. Our group confirmed this finding, by creating longitudinally loaded infarcts via directional reinforcement with a Dacron patch. While the ABM's predicted collagen alignment matched that of the patched infarcts at six weeks, it failed to capture the aligned longitudinal collagen seen at one, two, and three weeks post infarction [13]. Additionally, further inspection of infarcts in rats revealed that, under biaxial loading conditions, while average collagen alignment across many infarcts is essentially uniform, within individual infarcts average collagen alignment is in a seemingly random direction. Locally, within smaller subsections of the infarct, collagen is highly aligned[18]. The infarct ABM has not been able to generate the random average alignment or the local heterogeneous collagen alignment in biaxially loaded infarcts.

Besides, the model simulates a scaled down infarct about 200 microns in diameter with only hundreds of agents. A single six week simulation takes almost an hour to run on a standard laptop computer. As the simulation space and the number of agents increases, so does the solving time. Therefore, it would be extremely computationally expensive to simulate whole infarcts, which are on the scale of millimeters to centimeters and contain millions to billions of cells (dependent on the animal model), using the current ABM framework.

1.2.4 Objectives

This thesis seeks to overcome these predictive and computational obstacles providing a more useful tool for the understanding of scar formation and exploration of therapies for MI by doing the following. In chapter 2 we will transfer the infarct ABM to an object-oriented programming platform and incorporate a discrete migration scheme (as opposed to the continuous scheme utilized in the existing model) to improve the computational efficiency and flexibility of the model. In chapter 3 we will develop a computationally efficient method of simulating diffusing signaling molecules compatible with an ABM framework. In chapter 4 we will match early collagen alignment of the ABM to experimental data by evaluating new and existing model features. Finally, In chapter 5 we will determine if modifying cell-cell interactions in the ABM can produce alignment heterogeneity consistent with what we see in biaxially loaded infarcts.

CHAPTER 2

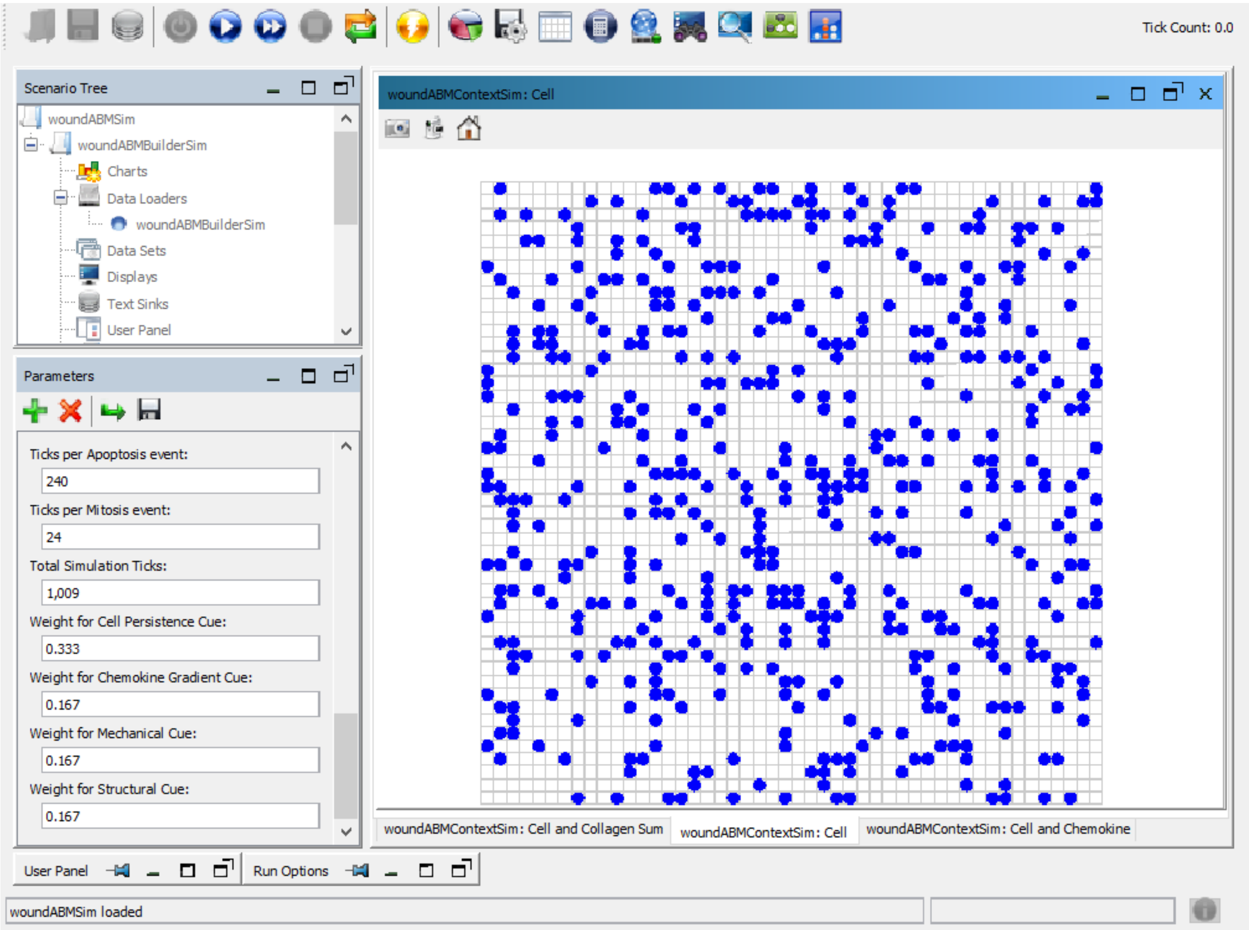
TRANSFER OF THE ROUILLARD INFARCT ABM TO REPAST

2.1 Introduction

Agent-based models (ABMs) of infarct healing have helped us to better understand how infarct scar is created and what factors contribute most to its characteristics. However, further development of the infarct healing ABM published by Rouillard et al. in 2012 has been slow due to structural difficulties with the model [17]. This ABM is written in a single MATLAB script that is over 1000 lines with many long nested for loops. This structure encourages mistakes when editing and makes debugging difficult. Additionally, to take advantage of the matrix computation power of MATLAB, the lion's share of the data is stored as vectors or matrices. This format results in a number of complex indexing schemes where referencing is difficult to understand. This complexity is particularly true for agent-associated values, which are stored separately from the actual agents and retrieved using an agent id index, which makes adding and removing agents from the simulation computationally and conceptually cumbersome.

The simplest approach to make the code more user-friendly would be to restructure it in MATLAB by moving well-defined agent behaviors and model initiation actions to separate functions and use data structures to attach information to agents. Unfortunately, this would do little to solve the obstacle of the model's computation time. A single six week infarct healing model runs in just under an hour on a standard desktop computer. While this may not seem very long, when you consider that the model already simulates a scaled-down infarct with only a few hundred agents and that multiple model runs must be averaged to eliminate stochastic effects, one hour is quite a long time. Therefore, improving the computational efficiency of the model is a significant factor in the building of subsequent iterations of the model.

One option to improve the usability and speed of the model is to move it to a more appropriate software platform. Repast Simphony is a java-based modeling toolkit widely used in the ABM community. It is already designed to separate model initiation from the objects in the model, namely the environment and the agents. Because Repast is written in java, an object-oriented language, it compiles code and stores data for object-oriented programs, such as ABMs, in a much more efficient and customizable manner. Plus it already includes a wealth of ABM focused functions. One of Repast's most useful features is the graphical user interface (GUI) (pictured in Figure 2.1) which allows the user to input all of the model parameters, visualize the simulation in real-time, and start and stop the model easily. Repast also has a built-in framework for batch running models which has a similar interface. Here, the user can set parameters as constant across all model runs, as a range of numbers, as a random integer, or as a space-separated list. The batch



Ticks per Apoptosis event: 240

Ticks per Mitosis event: 24

Total Simulation Ticks: 1,009

Weight for Cell Persistence Cue: 0.333

Weight for Chemokine Gradient Cue: 0.167

Weight for Mechanical Cue: 0.167

Weight for Structural Cue: 0.167

Figure 2.1: Repast graphical user interface with parameter specification tab and simulation visualization with agents (blue dots) before the infarct has been introduced

runner automatically makes all possible unique combinations of the parameters and runs each in a temporary directory. After all the models have been executed, predefined output file patterns are moved to a previously determined directory. Running models as a batch not only simplifies the process of running multiple models with a specific pattern but also saves time, because the simulation visualizations are not generated and the user can parallelize the models by adding local and remote hosts.

Additional computational benefits may be achieved by discretizing the agent locations in the ABM. This would allow static environmental parameters, such as the chemokine and mechanical cues, to be pre-calculated by location and stored in a lookup table. Finally, reducing the spatial resolution of the collagen data will also provide computational savings, however, it is unclear what losses in the predictive power of the model this might cause.

Therefore, the goals of this chapter are to transfer the Rouillard 2012 ABM to Repast Symphony; improve the model's usability by breaking the code down into relevant classes and methods; discretize the fibroblast agent locations to improve the computation speed; incorporate a flexible spatial scale, allowing users to prioritize the speed or resolution of their output; and validate this new ABM against the previous ABM.

2.2 Methods

2.2.1 Code Reorganization in Repast

The infarct ABM was constructed in Repast Symphony 2.6 using three classes. The woundABMBuilderSim class builds the context for the model, adds the initial fibroblast agents, and terminates the model after the desired simulation time has been achieved.

The woundABMContextSim class uses scheduled methods (blocks of code that are run at a certain tick or interval of ticks during the model execution) to build the environment that the fibroblast agents exist in. This environment consists of several grid value layers. Grid value layers are data storage objects provided by a built-in Repast class, which allow data of any type to be stored and accessed by coordinates. Structural information is stored in lists of grid value layers.

Thirty-six separate layers represent the oriented collagen fibers stored in five-degree angle bins from negative ninety degrees to ninety degrees. Fibrin is stored in the same way. Because cell migration is discrete in this implementation of the ABM, the mechanical and chemokine cues are now calculated for each location in the simulation space and stored in individual grid value layers. In addition to creating and managing the model environment, the woundABMContextSim class also calculates and writes output data.

The cellAgentSim class is used to create and run the behaviors of all of the fibroblast agents in the model. The cellAgentSim constructor (the portion of the class that creates the cell agent objects) imbues each fibroblast agent with a set of modifiable parameters including apoptosisTime (lifetime of the agent), apoptosisAge (age of the agent), mitosisTime (ticks between replication cycles); mitosisAge (ticks since the last replication event); depositionTime (ticks required to deposit collagen); depositionCounter (ticks since the previous deposition event); degradationTime (ticks required to degrade collagen); degradationCounter (ticks since the degradation event); angleSelection (angle of orientation of the agent); pastMigration (record of whether the agent moved during the last tick). These parameters, which the agents carry around with them, inform the agent decision tree carried out in the step method of the cellAgentSim class which is scheduled to run at each 0.5 tick (once every thirty minutes of simulation time). The agent decision tree is shown in Figure 2.2. Interested readers can find the code for the step method in Appendix Listing A.1.

2.2.2 Discretizing Migration

Aside from structural differences, the Repast ABM is functionally identical to its predecessor except in one aspect: the fibroblast agents do not move in a continuous space. Instead, the agents move between the center points of a discrete grid. This grid is the same size as the grid value layers which store the environmental data. This migration is the default for Repast, so no additional modifications to the code were necessary to implement it.

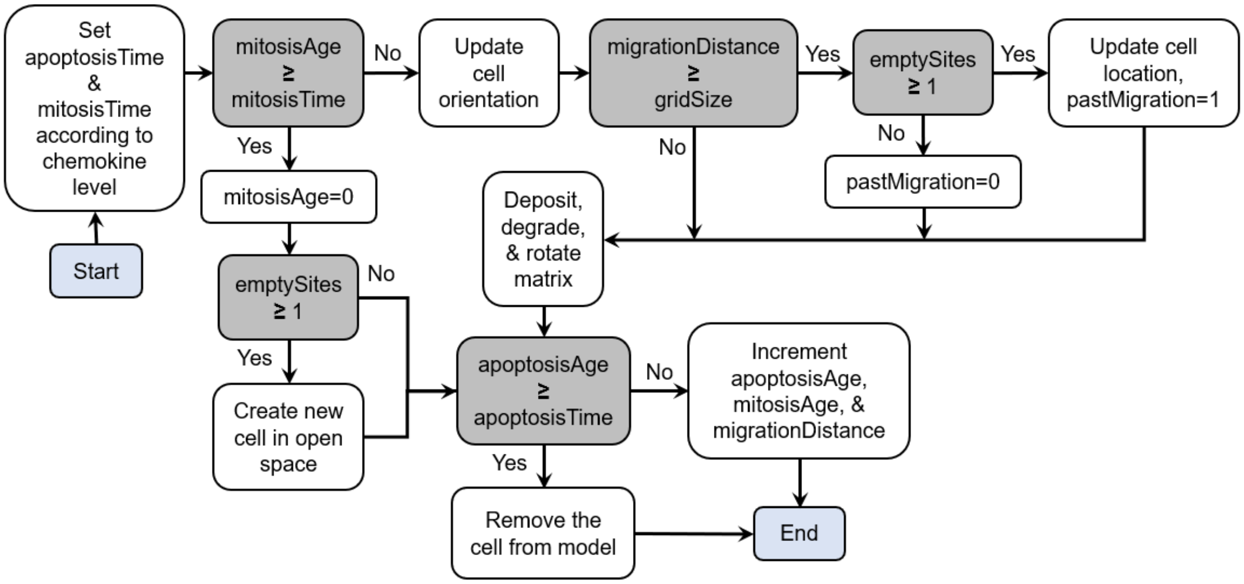


Figure 2.2: Agent decision tree run for each fibroblast agent at each 0.5 tick in the model. Start and end points (blue), if statements (gray).

2.2.3 Flexible Spatial Resolution

This ABM also permits the user to specify the spatial discretization (grid size), allowing the user to prioritize computational efficiency or high spatial resolution. To do this, a user-specified variable called gridUnitSize was added to the model to replace all instances where the 2.5X2.5 µm grid was included in a calculation. Other parameters in the model which were related to grid size were made a function of the gridUnitSize parameter.

2.3 Results

All Repast ABM variants discussed in this chapter were run with the parameter values in Table 2.1 entered into the Repast GUI. The continuous ABM written in MATLAB and the discrete ABM written in Repast were both run with three different infarct loading conditions: circumferential, longitudinal, and biaxial. All ABM results were averaged across four model runs to reduce stochastic effects. Figure 2.3 compares the continuous ABM to the discrete ABM. A 2.5X2.5 µm grid size was used in the discrete model to ensure a reasonable comparison to the continuous ABM which stores collagen on the same sized grid. The discrete ABM was also run with two additional grid sizes (10X10 µm and 5X5 µm). Figure 2.4 shows the effect of grid size on the discrete ABM. The average computation time for each ABM variant is shown in Figure 2.5.

parameter	**value**	**parameter**	**value**
Sample Width (µm)	480	Degradation Time (h)	0.5
Sample Height (µm)	480	Deposition Time (h)	0.5
Initial Fiber Distribution	Circumferential	Apoptosis Time (h)	240
Deposition Type	Aligned	Total Simulation Time (h)	1009
Persistence Cue Weight	0.333	Time to Mitosis (h)	240
Structural Cue Weight	0.167	Time step (h)	0.5
Mechanical Cue Weight	0.167	Initial Cell Count	576
Chemokine Cue Weight	0.167	Initial Collagen (%)	3.0

Table 2.1: Input parameters for the Repast GUI to create a standard model run.

Experimental measures of collagen MVL (measure of strength of alignment that ranges from 0 for randomly oriented collagen to 1 for perfectly aligned collagen), MVA (direction of alignment), and area fraction are also displayed in Figure 2.3 and Figure 2.4 for reference. Data for circumferentially loaded infarcts was taken from circular cryoinfarcts located at the equator of the LV, where strains are primarily in the circumferential direction. Data for biaxially loaded infarcts was taken from circular cryoinfarcts located at the apex of the LV, where strains are approximately equal in both directions [12]. Data for longitudinally loaded infarcts was taken from infarcts surgically reinforced in the circumferential direction, creating a longitudinally loaded infarct [13]. It should be noted that the model collagen area fraction was originally fit to data from coronary ligation induced rat infarcts at three weeks. These particular infarcts showed lower collagen area fraction than cryoinfarcts and patched infarcts. While we do not intend to address this discrepancy in this text, we chose to use the most consistent and up to date data.

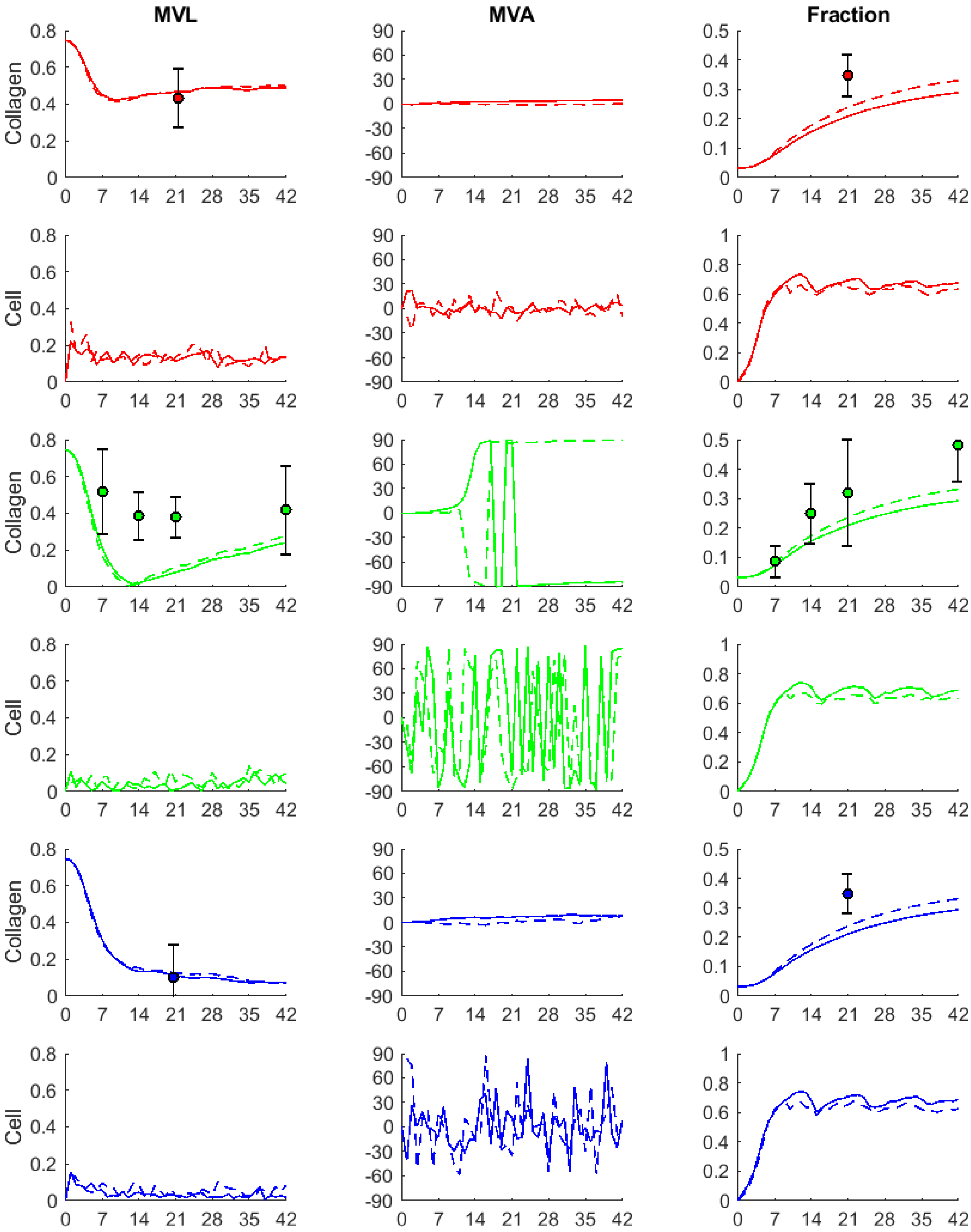


Figure 2.3: Continuous (solid line) and discrete (dashed line) infarct ABM implementations with circumferential (red), longitudinal (green), and biaxial (blue) loading conditions plotted with experimental data (error bars).

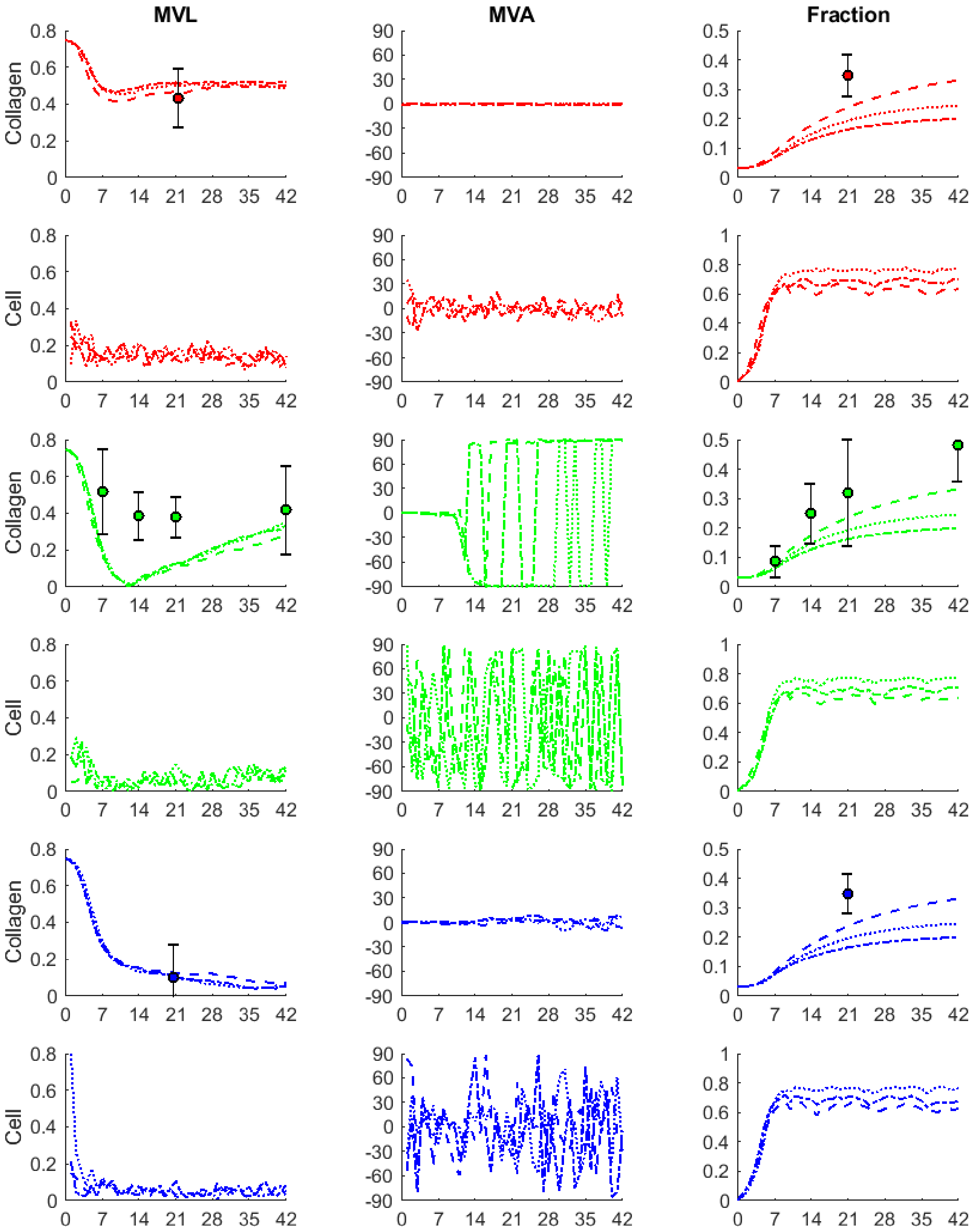


Figure 2.4: Discrete infarct ABM with 10X10 µm (dotted), 5X5 µm (dash-dotted), and 2.5X2.5 µm (dashed) grid sizes and circumferential (red), longitudinal (green), and biaxial (blue) loading conditions plotted with experimental data (error bars).

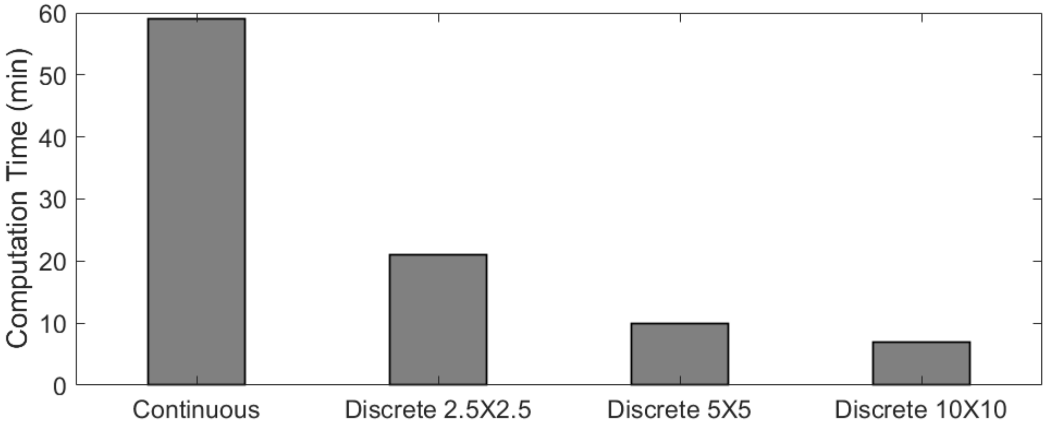


Figure 2.5: Computation time of continuous ABM implemented in MATLAB and various grid sizes of the discrete ABM implemented in Repast (numbers in um).

2.4 Discussion and Conclusions

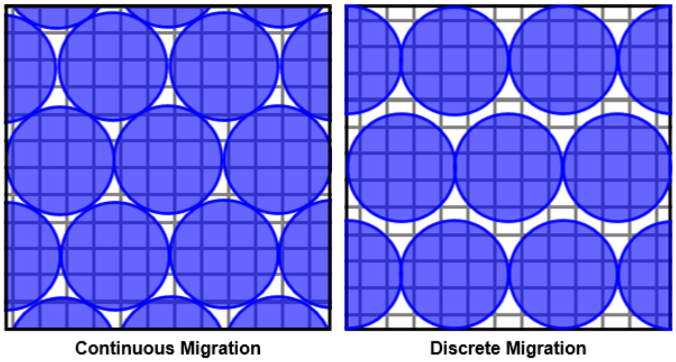


Figure 2.6: Tightest packing allowed by continuous migration and discrete migration on a 2.5X2.5 µm grid

In each panel of Figure 2.3 we can see good agreement between the two ABMs on cell and collagen MVL and MVA in all three loading conditions. It should be noted that what appears to be oscillatory MVA in the longitudinally loaded infarcts, actually only represents a slight shift in the collagen alignment which appears oscillatory because a collagen fiber aligned at 90 degrees is the same as a collagen fiber aligned at -90 degrees. The cell fraction of the discrete ABM is marginally depressed compared to that of the continuous ABM. This is due to the reduced packing factor imposed on the fibroblast agents by the discrete grid as shown in Figure 2.6.

The collagen fraction in the discrete model is greater than that of the continuous model. This difference has to do with the total area of collagen-containing grids each fibroblast agent has access to. In the continuous ABM the agents move over the discrete collagen grid. This varies the total area of grid space which they have access to, in this case between twelve and fourteen grids (examples can be seen in the first three panels of Figure 2.7). The grid space accessible to a cell is important because the amount of collagen deposited by a single cell is constant. Therefore, when the collagen deposition is spread out over a larger grid space, the amount of collagen which can be deposited into a single grid during a single deposition event decreases. This reduces the maximum amount of collagen which can accumulate in each grid, and also across the whole simulation space. We also see this effect in the collagen fraction of different grid sizes of the discrete model in Figure 2.4. The 5X5 µm grid size produces the lowest collagen fraction because it has the highest cell grid coverage. The 10X10 µm grid size produces a higher collagen fraction because it has a lower cell grid coverage. Finally, the 2.5X2.5 µm grid size produces the highest collagen fraction because it has the lowest cell grid coverage. The slight variations in the collagen MVL and MVA may be attributed to the aforementioned differences in collagen fraction.

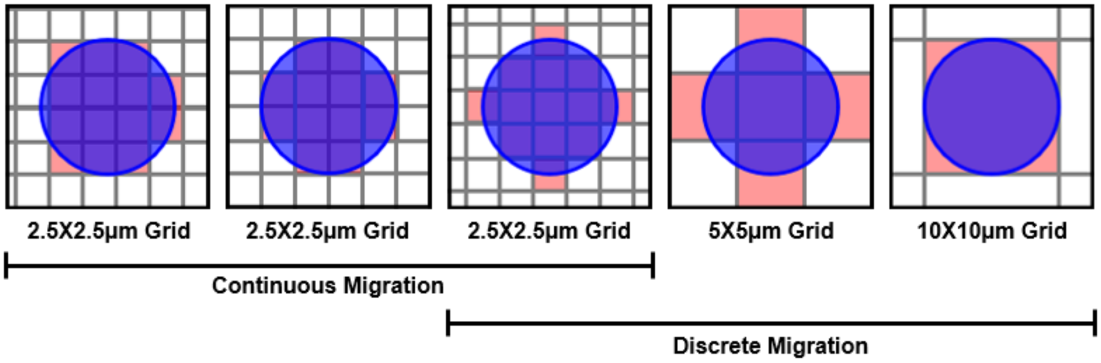


Figure 2.7: Agent grid coverage for continuous and discrete migration schemes

We can see additional effects of the grid size on the Repast model in Figure 2.4. Cell fraction plateaus at a higher level for a 5X5 µm grid and even higher for a 10X10 µm grid although the packing factor remains the same. This phenomenon is a result of how grid size affects the probability of a mitosis event occurring. As grid size increases, the number of grids that a cell occupies decreases, increasing the probability that enough adjacent grids are available for it to replicate. Figure 2.8 gives a simple example of this phenomenon which is also consistent with other findings in the literature [19].

In total, the results show that our discrete implementation of the infarct healing ABM in Repast

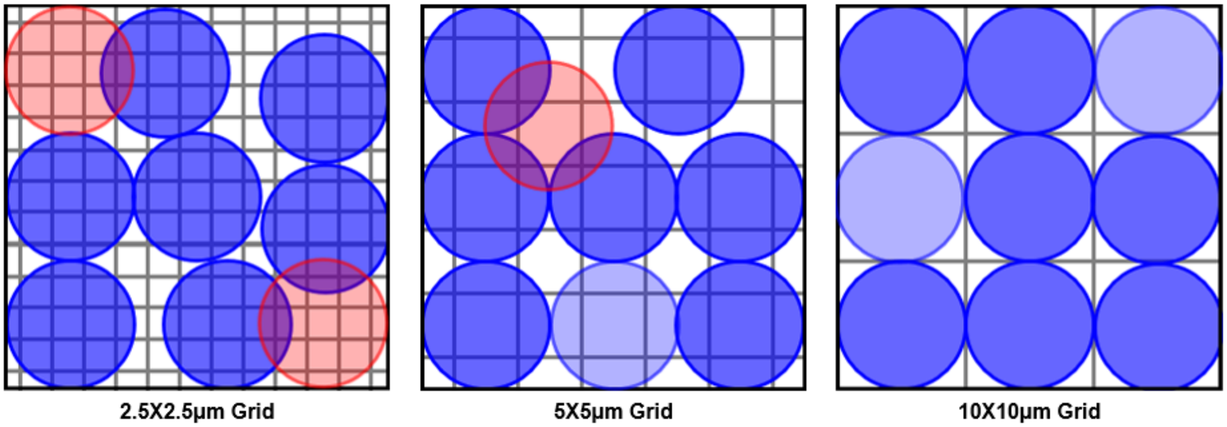


Figure 2.8: Replication scenarios for each grid size. All of these simulation areas have the same agent density, however, only the 10X10 µm grid allows two additional agents to be added through replication. Clearly, the probability of new agents being successfully added to the simulation decreases as the grid size gets smaller

is sufficiently similar to the original continuous version in MATLAB. The discrete implementation at a comparable spatial scale offers an approximately 50% reduction in computation time from the continuous implementation. Furthermore, the flexible spatial scale offered by the discrete ABM allows additional computational savings useful for efficient exploratory modeling while maintaining a good fit to experimental data.

This model is available for download through the Cardiac Biomechanics Group GitHub at https://github.com/cardiacbiomechanicsgroup/RepastABM. To access this specific version, check out the validatedDiscreteABM commit.

CHAPTER 3

AN IMAGE PROCESSING APPROACH TO MODELING DIFFUSION

3.1 Introduction

Cell signaling mediated by the diffusion of cytokines and other molecules may be useful or necessary in future variants of the infarct ABM, especially as we consider adding additional cell types like macrophages. However, solving the partial differential equations (PDEs) that govern diffusion in such multi-cell problems can be particularly challenging within the framework of an ABM.

Analytical solutions for diffusion are impractical given the geometric complexity and evolutionary nature of the problems we solve using the ABM. Numerical solutions, such as the finite element method (FEM) and finite difference method (FDM), can provide solutions to these PDEs. Unfortunately, most software packages designed to implement ABMs do not contain built-in FEMs, which are difficult to construct by hand. FDMs are much easier to implement, but require small time steps (fractions of a second) to generate a stable solution at a cellular spatial scale (tens of microns), making them computationally expensive. Therefore, the goal of this research is to establish a computationally efficient and easily implemented numerical method based on image processing techniques for approximating the diffusion of molecules using multiple sources and sinks.

3.2 Methods

This chapter will explore numerical solutions to two-dimensional dynamic diffusion equations of the following form,

$$\frac{dC}{dt} = D\nabla^2 C - k_{deg}C + k_{gen} \tag{3.1}$$

where C is the concentration, D is the diffusion rate, k_{deg} is the degradation rate and, and k_{gen} is the generation rate. Assuming that D is constant, then the diffusion segment of the equation $(dC/dt = D\nabla^2 C)$ is both linear and shift-invariant. Therefore, the transfer function of this system is equal to its impulse response, which is given by the analytic solution of a point source of a singular quantity of substance ($M = 1$) diffusing in an infinite two-dimensional plane [20].

$$C = \frac{M}{4\pi Dt} \exp\left(-\frac{r^2}{4Dt}\right) \tag{3.2}$$

Where M is the amount of substance diffusing, t is the time, and r is the radial distance from the source. An impulse response filter (IRF) can be generated from this analytic solution by sampling on a grid centered about the point source. IRFs may be calculated for any value of time (Δt) and at any spatial scale (Δx, Δy). When an IRF is convolved with a concentration profile, the result approximates that profile after diffusing over the time period of Δt.

$$C_{t+\Delta\mathrm{t}}\left[x, y\right] = C_t\left[x, y\right] * IRF_{\Delta\mathrm{t}}\left[x, y\right] \tag{3.3}$$

Convolution with an IRF can be performed iteratively to obtain the approximate concentration profile after multiple time steps. Degradation and generation can be incorporated between convolutions via addition and subtraction, as shown in Figure 3.1. Computation was performed in MATLAB R2018b (see Appendix Listing B.1 for code examples of the IRF method and FDM). Mean absolute difference (MAD) was used to assess the accuracy of the concentration profiles generated by the IRF method and the FDM. The solution given by the MATLAB PDE solver (which implements an FEM) was used as the gold standard solution.

3.3 Results

All three methods were used to determine the concentration profile of four representative biochemicals (D = 1000, 100, 10, and 1 µm^2/s) being constantly generated (k_{gen} = 0.01 nmol/s) from randomly sized and located circular areas centered in a 1x1cm plane (Δx = Δy = 20 µm), and

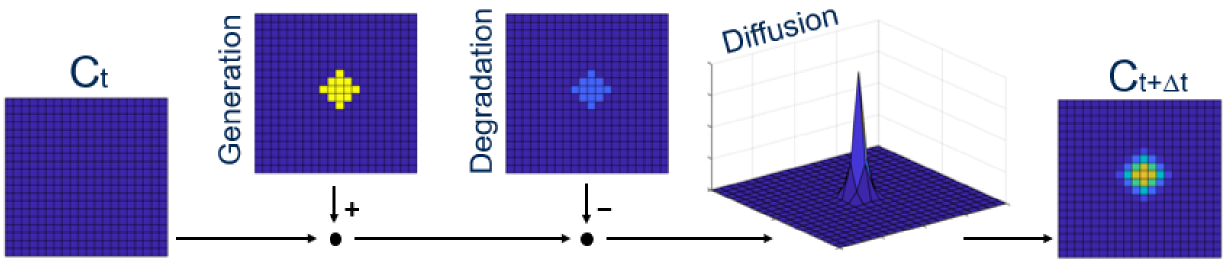


Figure 3.1: IRF method diagram (from left to right): blank concentration profile is input, generation from a cluster of cells is added, uniform degradation is subtracted, diffusion is calculated by convolution with an impulse response filter, and the concentration after Δt is output.

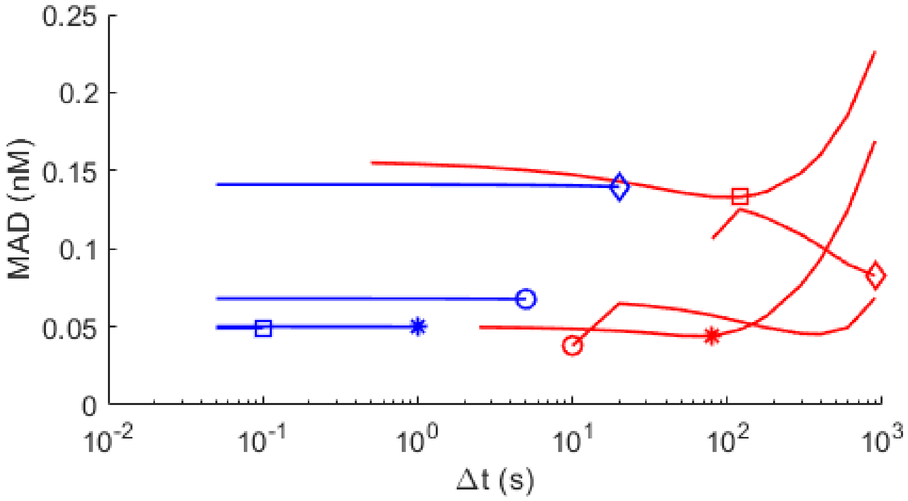


Figure 3.2: Mean absolute difference of the FDM (blue) and IRF method (red) compared to the FEM. The IRF method solution with the lowest MAD and the FDM solution with the highest Δt for each diffusion constant (D = 1000, 100, 10, and 1) are denoted by markers (□, $*$, ○, and ⋄, respectively).

uniformly degraded (k_{deg} = 0.001/s). The FDM and IRF method were both implemented over a range of Δt (0.1-900 s), for a total simulated time of one hour (t_{total} = 3600 s). The MADs of both methods compared to the FEM for each diffusion coefficient and value of Δt that generated stable solutions are shown in Figure 3.2.

We can see that the MAD of the FDM is dependant on D, but independent of Δt (creating the horizontal blue lines). However, the MAD of of the IRF is dependant on D and Δt. This produces the curved red lines which increase exponentially towards the right side of the graph. This behavior is a result of the increased ability of the filter to capture the diffusion as Δt increases and the increasing accuracy of the degradation and generation as Δt decreases. We would like to note, that we would expect the MAD of solutions with a diffusion coefficient of D=1000 to increase in an exponential manner at higher values of Δt, matching the trend seen for lower diffusion coefficients.

The range of Δt which produce stable solutions in the FDM is lower for higher values of D. In other words, as the diffusion coefficient increases, spreading the chemical mass faster, the time step used in the calculation must decrease to accurately approximate the diffusion profile. This is why the blue lines lengthen and D decreases. We see the opposite effect for the IRF. The red lines cover a shorter range of Δt as D increases. The range of stable solutions for the IRF start at different values of Δt because the filter has difficulty capturing steep diffusion profiles. Therefore, the IRF method produces unstable solutions for small diffusion coefficients with relatively short time steps.

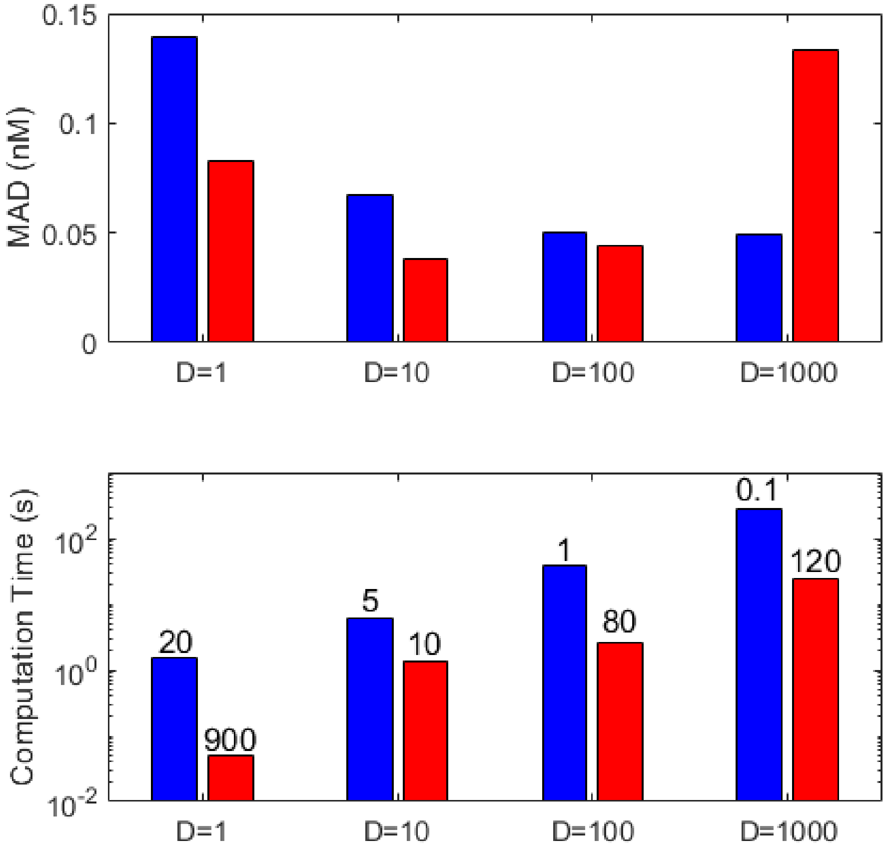


Figure 3.3: (top) MAD for marked solutions from Figure 3.2. (bottom) Computation time for marked solutions. The Δt for each solution is displayed above the corresponding bar.

The solutions from each method with the lowest MAD (most accurate) or highest Δt (most efficient, where multiple solutions had the same MAD) are denoted by markers in Figure 3.2; their MAD and computational efficiency are compared in Figure 3.3.

In the top of Figure 3.3, we see that for the FDM the MAD of the best solution for each diffusion constant goes down as D increases. However the bottom panel shows that computation time for the best FDM solutions increases as the Diffusion coefficient increases and the maximum allow step size (to remain stable) is forces to get smaller. The MAD of the IRF method's best solutions for each diffusion coefficient are not related to D alone. Furthermore, the computation time for the best IRF solutions does not appear to be directly related to Δt either.

3.4 Discussion

The results show that there are significant benefits to using the IRF method over the traditional FDM. For each diffusion coefficient tested, the IRF method was able to produce valid solutions in about an order of magnitude less time than the most computationally efficient FDM solution. Furthermore, the IRF method resulted in solutions with reduced mean absolute difference for all diffusion coefficients, with the exception of $D = 1000\,\mu\text{m}^2/\text{s}$. This last result is due to a difference between the boundary conditions of the different models. Both the FDM and FEM have a no-flux boundary condition, as opposed to the IRF method, which allows flux at the boundary because it is constructed from the analytic solution of a point source in an infinite 2-D plane. This difference in boundary conditions was not apparent for the other values of D because they were not high enough to transport sufficient chemical mass to the boundaries. This limitation can be offset in practice by padding the simulation mesh with additional elements beyond the region of interest and/or specifying appropriate consumption rates at the border nodes.

Also of note is that the relationships between Δt, D, and computation time are different for the FDM and IRF method. The computation time for the FDM is directly related to Δt. For a given t_{total}, as Δt decreases, the number of iterations performed increases, driving up the computation time. The computation time of the IRF method is related to the product of D and Δt. As $D *$

Δt increases, the filter extent must increase to capture the spread of the diffusion. As filter size increases, the computation time for an individual iteration of the method also increases. This relationship accounts for the results in Figure 3.3, where the IRF method solution for D = 1000 has a higher computation time than that of D = 100 and D = 10, despite the fact that the latter two have lower Δt than the former.

Overall, the iterative impulse response filter method introduced in this chapter appears to be a useful tool for generating accurate concentration profiles of diffusing molecules with relatively large time steps, facilitating computationally efficient implementation even within larger models where built-in PDE solvers are not available.

The research was presented at the Summer Biomechanics, Bioengineering, and Biotransport Conference, June 17-20, 2020.

CHAPTER 4

IMPROVING MODEL PREDICTIONS OF EARLY COLLAGEN REMODELING

4.1 Introduction

The original infarct healing ABM published by our group in 2012 [17], matched the measured collagen content in healing rat infarcts at one, two, three, and six weeks. The collagen alignment predicted by the model was first evaluated against the most similar experimental data available at the time: cryoinfarct scar harvested at three weeks. We replicated the infarct size, shape, and location (loading condition) of different cryoinfarcts, and were able to match the collagen alignment at three weeks. The model was also able explain some aspects of the transmural distribution of scar collagen alignment reported in pig infarcts at three weeks. However, for biaxially loaded infarcts the ABM predicted a slow transition from the highly aligned collagen structure present before infarction to the uniformly aligned structure seen in the data at three weeks. This was not consistent with earlier findings, which showed that scars from ligation-induced infarcts in rats (which are biaxially loaded during healing) are mechanically and structurally isotropic as early as one week. [21].

Additionally, when a longitudinal loading condition was prescribed in the original infarct ABM, the model predicted that a longitudinally aligned collagen matrix would develop. Caggiano et al. directly tested this prediction experimentally by sewing a Dacron patch, which allowed strain only in the longitudinal direction, onto the epicardium of adult rats immediately following infarction via coronary ligation. The six-week collagen alignment of these patched infarcts matched that predicted by the ABM. However, the ABM predicted that collagen at the mid-wall would initially align with the circumferentially oriented matrix that survived the initial infarction, then gradually switch to longitudinal alignment over several weeks as fibroblasts deposited new longitudinally oriented collagen. Instead, the experiments clearly showed that collagen was already strongly aligned in the longitudinal direction at one week, and maintained that alignment as it continued to accumulate through six weeks (Figure 4.1) [13]. These discrepancies suggest that our understanding of what guides collagen alignment in the early stages of healing is flawed, which is important because novel therapies such as polymer injection are designed to be deployed during this early phase of healing [22].

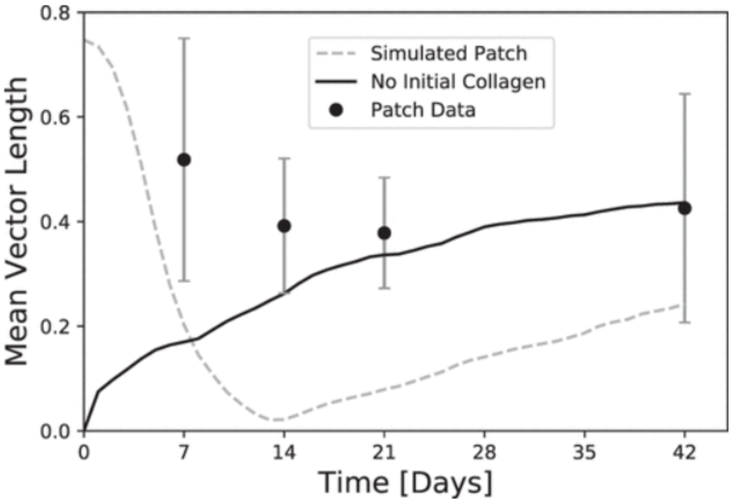


Figure 4.1: Mean vector length of longitudinally loaded infarct simulations under normal assumptions (dashed line) and with initial collagen removed from the wound area (solid line). Error bars show data collected from patched infarcts.

Note. Reprinted from "Surgical reinforcement alters collagen alignment and turnover in healing myocardial infarcts", by Caggiano, L. R., 2018, *American Journal of Physiology-Heart and Circulatory Physiology*, *315*, p. H1047.

In the same study, Caggiano et al. attempted to remedy this discrepancy by testing the assumption that the initial structure, including collagen and non-collagen components, was completely disrupted during infarction. The rational behind removing the collagen, was to prevent the fibroblasts from having to remodel or overwhelm the pre-existing 2-3% area fraction of circumferentially aligned collagen during the first week of healing. Similarly, the non-collagen components, which were intended to represent the physical constraints imposed by dead myocytes before they are resorbed and other aligned matrix components, were removed to prevent the initial structural cue (which would otherwise be circumferentially aligned), from conflicting with the longitudinal mechanical cue. While eliminating the pre-existing structure improved the fit of the model at the two and three week time points, the model still failed to replicate the high longitudinal alignment seen in patched animals at one week.

Removal of the collagen and non-collagen structural components, was not only intended to improve the results, but also reflected a genuine question as to the state of the structure after infarction and before fibroblast invasion. Previously we had assumed that the structure was acutely unchanged by infarction. However, it may be reasonable to assume that some percentage of the preexisting aligned collagen is disrupted by the inflammation, edema, early white blood cell invasion, etc. Furthermore, the contact guidance cues, and even the physical boundaries, offered by the myocytes in the infarct are likely to be disrupted to some extent during necrosis.

Another realistic change to model's assumptions about the initial infarct structure, alternative to removing the structure entirely, is to simulate the non-collagen structure with a uniform alignment. This may be a more realistic representation of the disrupted structure following infarction.

Adding new assumptions may also make the model more biologically relevant and change the collagen alignment time course. Previously, the MVL (a measure of alignment) of the structure was used to scale the strength of the structural cue on fibroblast alignment. In other words, a uniformly aligned structure would not contribute to the alignment of agents but a perfectly aligned structure would contribute maximally to the alignment of agents. However, under this scheme a perfectly aligned structure with 1% collagen area fraction would contribute just as much to the alignment of agents as a perfectly aligned structure with 30% collagen area fraction.

One other feature of the model that has been brought into question, is the use of collagen rotation by fibroblasts. While collagen rotation has been shown to contribute to matrix alignment in unloaded collagen gels as they compact [23], its existence *in vivo* has yet to be confirmed. Furthermore, without the use of collagen rotation, the current model is unable to produce aligned collagen in any loading condition. This suggests that other factors contributing the alignment of fibroblasts, particularly the mechanical cue, may need to be reevaluated.

The overall goal of this chapter is to evaluate the biological relevance of structure related features in the infarct healing ABM developed in chapter 2 by removing or changing said features

- alone or in combination - and observing the affect on early collagen alignment under different loading scenarios.

4.2 Methods

The following model changes are activated or suppressed by individual toggles added to the GUI. These toggles open corresponding if gates in the model which route to newly written code chunks that produce the desired effect of the toggle.

When the wound collagen toggle is turned on, the collagen grid value layers in the model are initialized normally. When the wound collagen toggle is turned off, the grids within all 36 collagen layers corresponding to locations within the wound area are populated by zeros.

When the non-collagen toggle is turned on, the non-collagen grid value layers in the model are initialized normally. When the non-collagen toggle is turned off, the non-collagen grid value layer is populated with zeros.

To vary the orientation of the non collagen structure, a non-collagen distribution parameter was added to the GUI. This parameter determines the distribution type (circumferential, longitudinal, or uniform) given to a new, non-collagen specific, distribution vector in the method which populates the structural grid value layers. This method can be found in Appendix Listing C.1.

When the scale structural cue toggle is on, the collagen from all 36 layers beneath the cell is totaled up and normalized by the maximum possible collagen that could exist in that area. This value is then used to scale the structural cue (W_S) when various orientation cues are integrated to direct overall cell behavior. When the scale structural cue toggle is off, the structural cue normalization factor is set to 1.

Listing 4.1: Scale structural cue gate and calculations in the guidanceCue method

```
// Calculate structural guidance cue normalization factors based on
   maximum values
double Ms = 1;
if (scaleStructuralCue == true) {
```

```
  final GridValueLayer colTotal = (GridValueLayer)
     woundabmspace.getValueLayer("Collagen Sum");
  final GridValueLayer nonColTotal = (GridValueLayer)
     woundabmspace.getValueLayer("Non Collagen Sum");
  final List<GridPoint> coveredSites = cellCoverage(x, y);
  double colSum = 0;
  double nonColSum = 0;
  for (GridPoint site : coveredSites ) {
    final int siteX = site.getX();
    final int siteY = site.getY();
    colSum = colSum+colTotal.get(siteX, siteY);
    nonColSum = nonColSum+nonColTotal.get(siteX, siteY);
  }
  Ms = colSum/(kColGenMax/kColDegMax);
}
```

When the collagen rotation toggle is turned on, the collagen rotation method is run as a part of the agent behaviors during the step method. When the collagen rotation toggle is turned off, the rotation method is skipped during execution of the step method.

Listing 4.2: Collagen rotation gate in the step method

```
// Check for rotation, degradation, deposition
if (colRotation == true) {
  collagenRotation(siteX, siteY, chemokine, collagenLayers);
}
collagenDeposition(siteX, siteY, chemokine, collagenLayers,
   depositionTime);
matrixDegradation(siteX, siteY, chemokine, collagenLayers,
   nonColLayers, degradationTime);
```

Mean vector length MVL (measure of strength of alignment that ranges from 0 for randomly oriented collagen to 1 for perfectly aligned collagen) is calculated using the following equation, where $collagen_x$ is the mean x component of all the collagen in the wound and $collagen_y$ is the mean y component of all the collagen in the wound.

$$MVL = \sqrt{collagen_x^2 + collagen_y^2}$$

4.3 Results

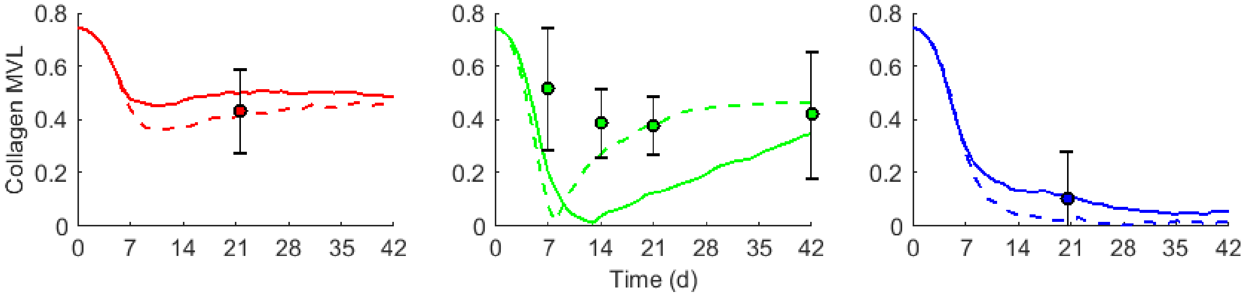
All of the following exploratory models were conducted on a 10X10 µm grid. The best fitting version of the model was also averaged across four model runs to reduce stochastic effects. The full model results including the cell and collagen alignment and area faction time courses is shown in Figure 4.6.

First, all of the individual feature changes were run separately in the ABM. Circumferential, longitudinal, and biaxial loading conditions were simulated for all cases to ensure that model changes that improved fits to our data from longitudinally loaded infarcts did not simultaneously degrade our ability to fit prior three-week data from circumferentially and biaxially loaded infarcts. The time courses of predicted strength of alignment for these models are shown in Figure 4.2.

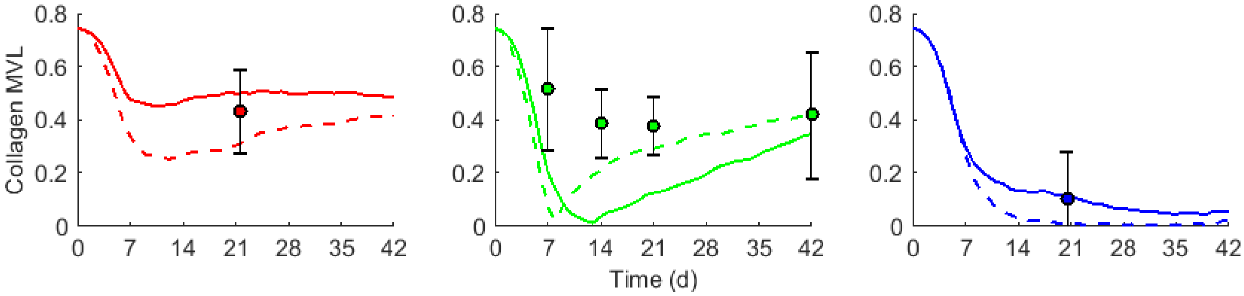
Removing the initial non-collagen structure reduces the influence of the original structure on cell behavior. In the longitudinal loading condition, the switch to longitudinal orientation occurs more quickly but the shape of the curve remains similar. The circumferential and biaxial loading conditions show decreased collagen alignment at earlier time points, but appear to still reach the same steady state value towards week six.

Setting the initial alignment of the non-collagen structure to be uniform, had a similar affect to removing the non-collagen structure entirely. However, the overall alignment across all loading conditions and time points is lower with a uniform distribution of non-collagen than without the non-collagen structure.

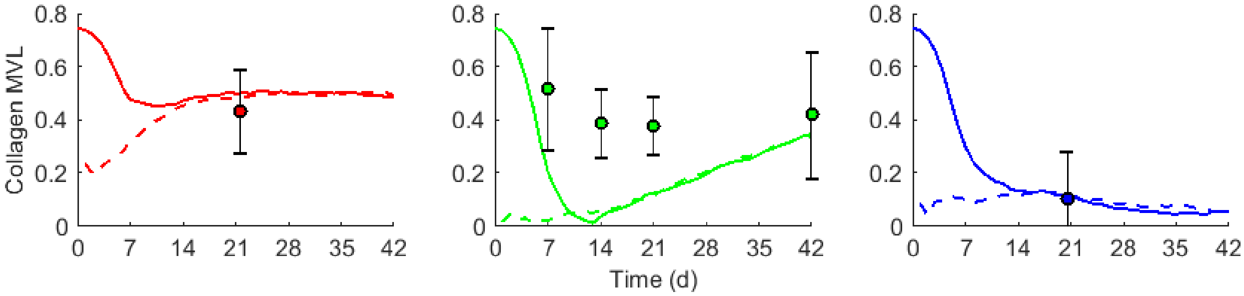
Eliminating the initial collagen allowed the mean collagen direction to be gradually determined



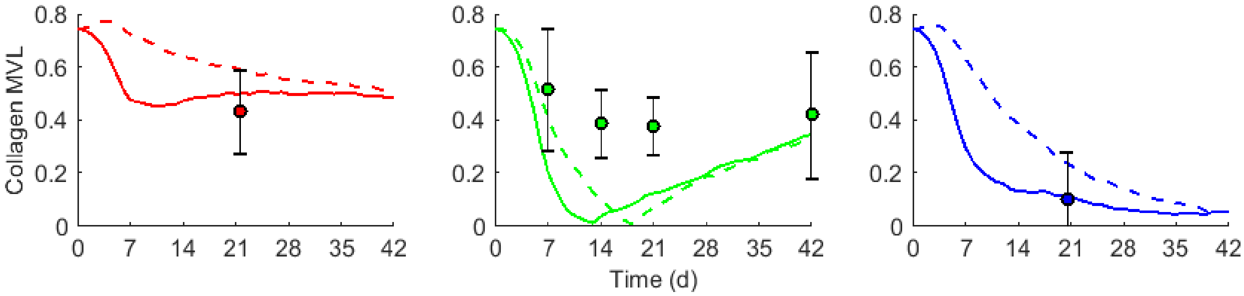
(a) No Non Collagen



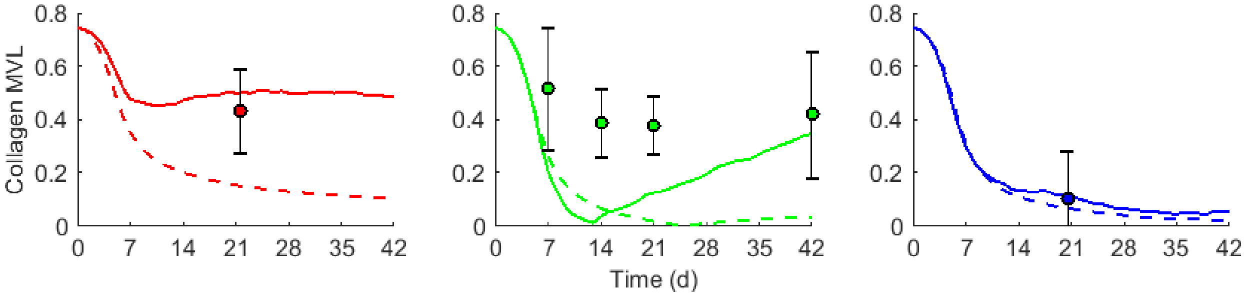
(b) Uniform Non Collagen



(c) No Initial Collagen



(d) Scale W_s



(e) No Rotation

Figure 4.2: Collagen mean vector length of the standard (solid line) and modified (dashed line) ABM with experimental data (error bars). Circumferential (red), longitudinal (green), and biaxial (blue) results shown.

by deposition rather than switching from an initial to a final orientation, so that the MVL curve rises monotonically to a steady state rather than dipping and then rising again. While the alignment of all loading conditions is low during the first 2 weeks of simulation, the alignment after two weeks is virtually unchanged.

Scaling the structural cue in proportion to the content of the structure increases the effect of all the other cues at early time-points. Particularly, the cell's persistence which is already weighted higher than the other guidance cues. Because the initial structure is circumferentially aligned, cells will invade slightly faster from the left and right sides of the infarct, creating the bump in alignment we see in the circumferential and biaxial loading cases. Heightened persistence smooths and slows changes in all of the loading conditions. Regardless, all models appear to still reach the same steady state value toward week six.

Removing collagen rotation slowed changes to the collagen structure and drastically reduced overall alignment in all loading conditions and all model variants.

No individual change to the model was sufficient to produce high alignment in longitudinally loaded infarcts and uniform in biaxially loaded infarcts at week one. Even so, only one of the changes (No Rotation) pushes the circumferential and biaxial simulations outside of the error bars of the data, so any of the other variants could be considered equally successful in terms of their predictive capability when compared to the original model.

Next, all the possible combinations of two (Table 4.1), three, and four (Table 4.2) of these changes were simulated (MVL results are shown in Appendix Figure D.1, Figure D.2, and Figure D.3). Combinations including both the removal of the initial non-collagen structure and a uniform initial non-collagen structure were omitted because of their incompatibility. Model change combinations were assessed visually. Two promising combinations were selected for further review based on their potential to fit the MVL time course for all of the loading conditions.

	No Non Collagen	Uniform Non Collagen	No Initial Collagen	Scale W_s	No Rotation
No Non Collagen	Figure 4.2a	X	Figure D.1a	Figure D.1b	Figure D.1c
Uniform Non Collagen	X	Figure 4.2b	Figure D.1d	Figure D.1e	Figure D.1f
No Initial Collagen	X	X	Figure 4.2c	Figure D.1h	Figure D.1g
Scale W_s	X	X	X	Figure 4.2d	Figure D.1i
No Rotation	X	X	X	X	Figure 4.2e

Table 4.1: First and second order change combinations. X's indicate redundant or incompatible change combinations.

No Non Collagen + No Initial Collagen + Scale W_s	Figure 4.3
No Non Collagen + No Initial Collagen + No Rotation	Figure D.2b
No Non Collagen + Scale W_s + No Rotation	Figure D.2c
Uniform Non Collagen + No Initial Collagen + Scale W_s	Figure D.2d
Uniform Non Collagen + No Initial Collagen + No Rotation	Figure D.2e
Uniform Non Collagen + Scale W_s + No Rotation	Figure D.2f
No Initial Collagen + Scale W_s + No Rotation	Figure D.2g
No Non Collagen + No Initial Collagen + Scale W_s + No Rotation	Figure 4.4
Uniform Non Collagen + No Initial Collagen + Scale W_s + No Rotation	Figure D.3b

Table 4.2: Second and third order change combinations.

The only change combination that was able reach the week one error bars for longitudinal loading was the model with no non-collagen, no initial collagen, and a scaled structural cue. This version also produces uniform collagen in the first week of biaxially loaded simulations and maintains a good fit to the three week data in the circumferential loading scenario. The MVL time courses for this model are shown in Figure 4.3.

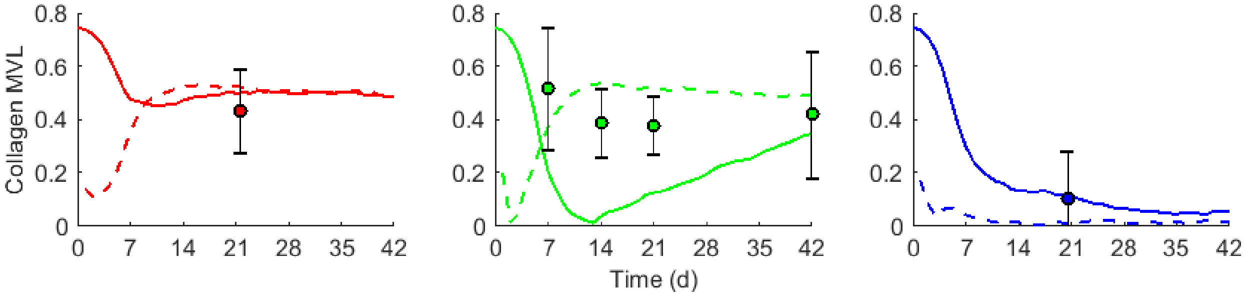


Figure 4.3: Collagen mean vector length of the ABM with no non-collagen, no initial collagen and a scaled structural cue (dashed line) plotted with standard results (solid line) and experimental data (error bars). Circumferential (red), longitudinal (green), and biaxial (blue) results shown.

One other combination of changes showed promising results. The model with no non-collagen, no initial collagen, a scaled structural cue, and no rotation did not match any of the data for circumferential or longitudinal infarcts. However, while the alignment for these loading scenarios was low across the entire time course, the overall trend of alignment (Figure 4.5) looked as if it would fit the data nicely if the cells were made to respond more highly to the direction of stretch.

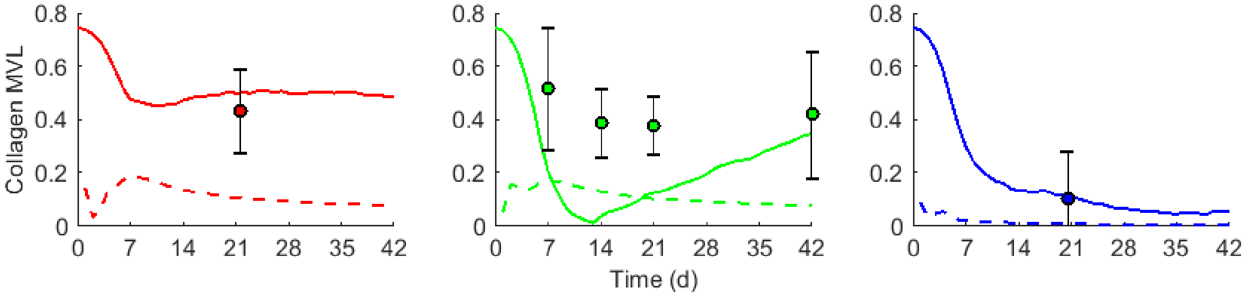


Figure 4.4: Collagen mean vector length of the ABM with no non-collagen, no initial collagen, a scaled structural cue, and no rotation (dashed line) plotted with standard results (solid line) and experimental data (error bars). Circumferential (red), longitudinal (green), and biaxial (blue) results shown.

To test this theory, the mechanical cue of the model with no non-collagen, no initial collagen, a scaled structural cue, and no rotation was adjusted to better fit the early collagen alignment time-course. Mechanical cues from 0.2 to 0.9 at intervals of 0.1 were simulated. The three best mechanical cue weights are shown in Figure 4.5.

This version of the model, with mechanical cue weight (W_m) set to 0.6, was able to fit the circumferential alignment at three weeks, produce uniform alignment in the biaxial loading con-

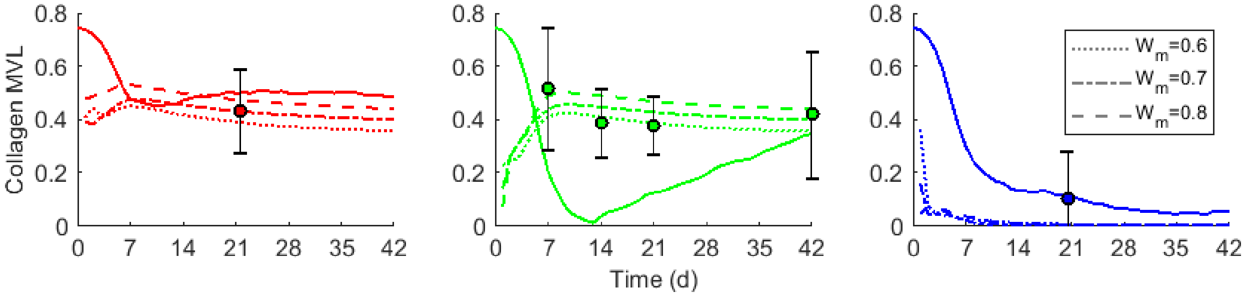


Figure 4.5: Collagen mean vector length of the ABM with no non-collagen, no initial collagen, a scaled structural cue, and no rotation with various mechanical cue weights (dashed, dotted, and dash-dotted) plotted with standard results (solid line) and experimental data (error bars). Circumferential (red), longitudinal (green), and biaxial (blue) results shown.

dition by week one, and fit within the error bars of all the time points for the longitudinal loading condition. It is in this latter accomplishment that this version of the ABM outperforms the other promising, but more conservative, set of changes. The version with no non-collagen, no initial collagen, and a scaled structural cue does not fall within the error bars of the longitudinal data at weeks two and three.

Therefore, the model with no non-collagen, no initial collagen, a scaled structural cue, and no rotation with a mechanical cue weight of 0.6 provides the best fit to the data across all time points and loading conditions. Comprehensive results for this version of the model are shown in Figure 4.6. Here, we can see that cell MVL is markedly higher in loading conditions with a strong mechanical cue. Cell and collagen MVA are consistent with previous results for the longitudinal and circumferential loading conditions and random for the biaxial loading condition. Collagen fraction is lower during the first two to three weeks of collagen accumulation due to the removal of the initial collagen matrix, but fits the week one collagen fraction from patched infarcts and still approaches the same final value. Cell fraction remains unchanged.

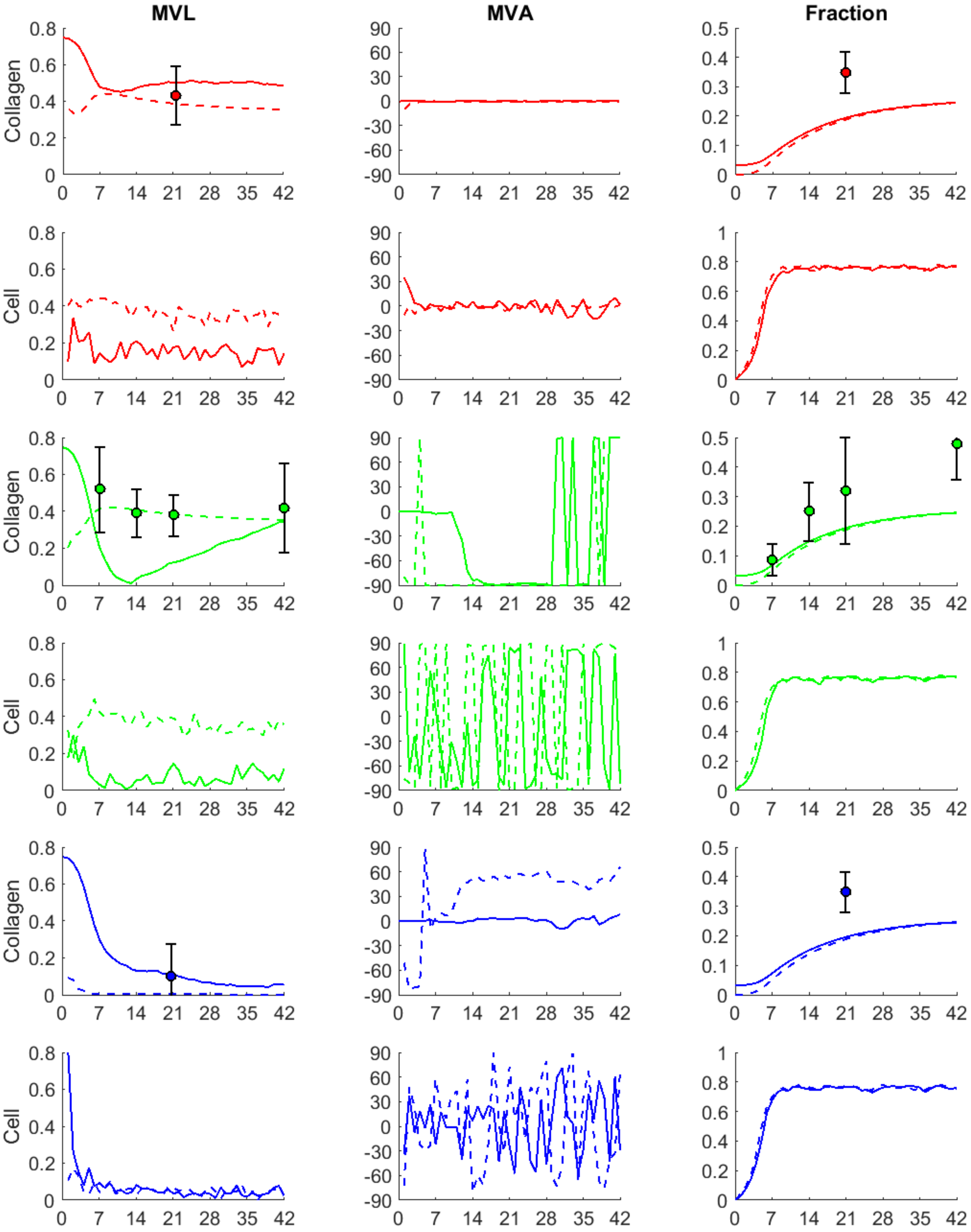


Figure 4.6: The best fitting combination of changes (No Non Collagen + No Initial Collagen + Scale W_s + No Rotation) to the ABM with a tuned mechanical cue weight (Wm=0.6) (dashed) compared to the standard model (solid) and experimental data (error bars). Circumferential (red), longitudinal (green), and biaxial (blue) results shown.

4.4 Discussion and Conclusions

Some combinations of modifications we considered were able to better match our new data on the time course of collagen alignment during longitudinal loading while still fitting data used to validate the original ABM. The three key changes that improved the match to the data were eliminating initial structure (both collagen and non-collagen components), scaling structural guidance in proportion with collagen content, and eliminating rotation.

It has been found that pro-inflammatory cytokines, including Tumor Necrosis Factor-α and IL-1β, released after MI enhance matrix metalloproteinase (MMP) expression and activity promoting degradation of collagen and other matrix proteins which further enhance the inflammatory cascade [24] [25]. This biologically supports our findings that removing the initial structure from the model is necessary to generate highly aligned collagen at early time points.

Fibroblasts interact with matrix proteins through surface proteins, usually integrins [26]. Therefore, it makes sense that the as the amount of matrix surrounding a cell increases, the integrins on the cell's surface are more likely to bind to matrix components, allowing the cell to better sense its structural surroundings. This was the rational we employed in scaling the structural cue magnitude proportional to the collagen content. Because the structural cue is less aligned than the mechanical cue, as collagen is deposited the structural cue increases in magnitude. This causes the cell alignment to drop creating the plateau in longitudinal MVL time course which continues through week six. This subtle sift in cue weighting is obscured when collagen rotation is allowed because the initial collagen, which is more weakly aligned, is remodeled before there is enough content for it stabilize and slightly lower the later collagen alignment.

The simulations with no collagen rotation clearly show that the model was reliant on rotation in order to produce aligned collagen. Considering that there is currently no *in vivo* data supporting the rotation of collagen fibers by fibroblasts, removing this feature of the model and attempting to fit the data through, other, more biologically relevant means, was important.

Even with all of these biologically plausible modifications in collagen turnover and structural guidance cues, the relative weight of the mechanical cue on cell orientation had to be changed in order to fit the data. The original ABM conservatively set the mechanical cue weight equal to the structural cue weight (which was fitted to published measurements of fibroblast behavior). Now that we have data from additional loading conditions [13] we can actually fit the mechanical cue weight. Increasing the mechanical cue weight to almost four times it's original value, seems drastic. However, we had already established in previous research that collagen alignment is ultimately determined by mechanics [17], therefore is makes sense that the mechanical cue weight would be higher than the other cues.

The natural result of increasing the mechanical cue weight by so much, is that the average cell angle is much more highly aligned than we have seen in previous versions of the model. However, this is also consistent with cell alignment data that our group has collected, which shows that infarct fibroblasts are aligned with a similar strength to local and global collagen alignment [18]. This finding reinforces our decision to increase the weight of the mechanical cue.

While circumferential and biaxial alignment time courses in Figure 4.6 look fairly different than before, they still fit the experimental data well. In fact, the trends in the biaxial simulations show promise for better fitting the biaxial data. Namely, the removal of the initial non-collagen and collagen structures, has allowed the collagen to align in a random direction consistent with what we have seen for whole biaxially loaded infarcts.

As a bonus, these model modifications also simplify the model setup and agent behaviors and have a positive effect on computational efficiency.

To conclude, this chapter details the review of various changes to assumptions in the infarct ABM. By removing the initial wound collagen and non-collagen, scaling the structural cue strength to the content of the structure, removing rotation of collagen fibers by fibroblast agents, and tuning the mechanical cue weight, we were able to match the collagen alignment time course of longitudinally loaded infarcts while maintaining a good fit to circumferentially and biaxially loaded infarcts.

This model is available for download through the Cardiac Biomechanics Group GitHub at https://github.com/cardiacbiomechanicsgroup/RepastABM. To access this specific version, check out the collagenAlignmentTimeCourseTuning commit.

CHAPTER 5

COLLISION GUIDANCE AND COLLAGEN ORIENTATION HETEROGENEITY

5.1 Introduction

In the previous chapter we focused on fitting our infarct ABM to new data in patched longitudinally loaded infarcts. The changes we made in chapter 4 also improved the match to the predicted average orientation of ligation-induced biaxial infarcts as well. Previously, the ABM predicted that these infarcts would slowly transition from the circumferentially aligned collagen matrix present before infarction to a virtually uniform matrix with an overall preference for the circumferential direction over a period of six weeks. Removing the initial structure, both collagen and non-collagen elements, allowed biaxial loading simulations to develop uniformly aligned and spatially heterogeneous collagen scar with random average alignment as early as one week after infarction (Figure 5.1). This matches what we see in apical (biaxially loaded) rat infarcts much better than before.

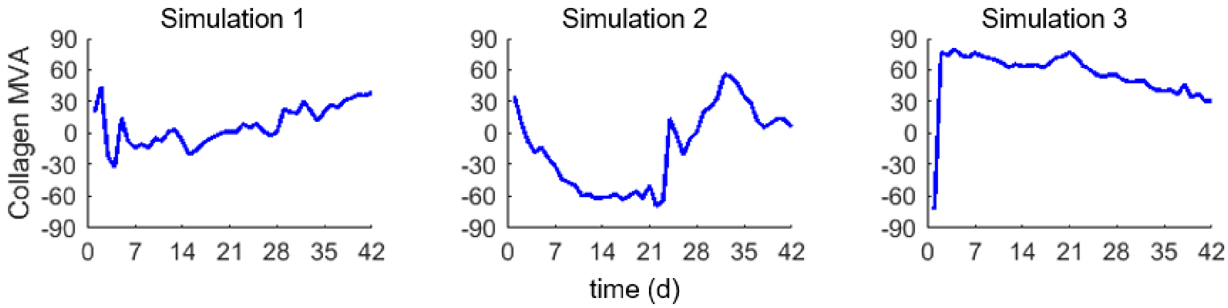


Figure 5.1: Collagen alignment direction differs between identical biaxial simulations.

While biaxial loading leads to low alignment of collagen in a random preferred direction across whole infarcts, local alignment of infarct sub-regions is quite high and persists over time [18]. We can see an example of this in Figure 5.2 which shows the global and local alignment of an apical infarct stained for collagen with picrosirius red. Quantification of the collagen heterogeneity revealed that the relative alignment of two fibers is related to their distance from one another.

So far, the Rouillard infarct ABM [17] has only been able to produce early heterogeneity under somewhat tenuous assumptions. Richardson et al. found that increasing the distance at which cells can sense or deposit collagen in the ABM, produced locally aligned collagen after one week of remodeling. However, the authors concluded that collagen deposition and sensing distances required to produce heterogeneity are not biologically realistic based on our current understanding of fibrillogenesis or structural sensing, respectively [18].

Understanding what drives these heterogeneous alignment patterns could provide insight for developing new methods to control scar alignment. Additionally, the emergence of such patterns are important to characterize because they may be mechanically disadvantageous and pose an ongoing threat to regenerative therapies, which require aligned tissue in order to generate functional, aligned myocytes [27].

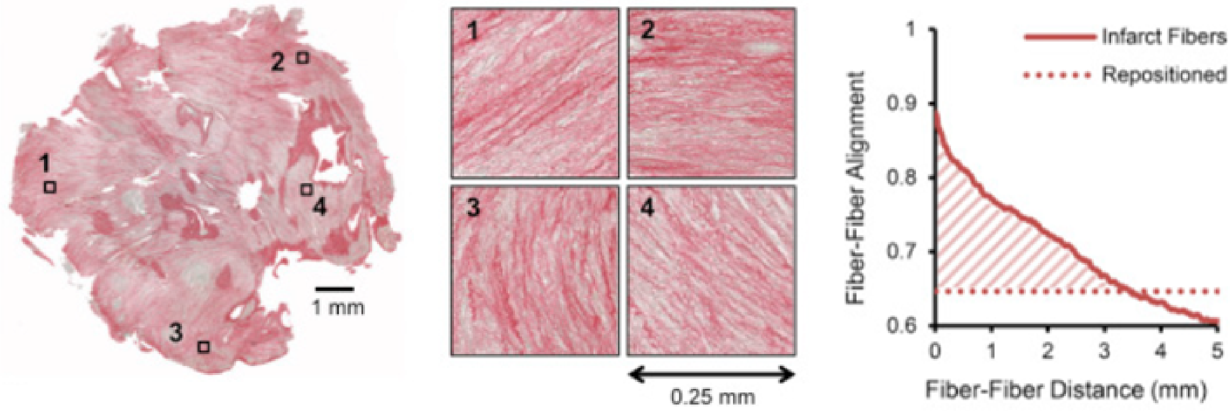


Figure 5.2: Mid-wall apical rat infarct section stained for collagen (picrosirius red). (left) Whole infarct histology shows low global alignment, (middle) sub-regions of aligned collagen, (right) fiber-fiber alignment verses distance quantifies regional heterogeneity.

Note. Adapted from “Emergence of Collagen Orientation Heterogeneity in Healing Infarcts and an Agent-Based Model”, by Richardson, W. J., 2016, *Biophysical Journal*, *110*, p. 2268.

Collision guidance, a phenomenon where cells become more aligned after running into one another, has been proposed as means of alignment. Recently, Park et al. found that different degrees of matrix alignment, from highly aligned across large tissue sections, to heterogeneously aligned across smaller tissue sections, to uniformly aligned, could be produced *in vitro* and *in silico* with high, medium, and low levels of collision guidance, respectively [28].

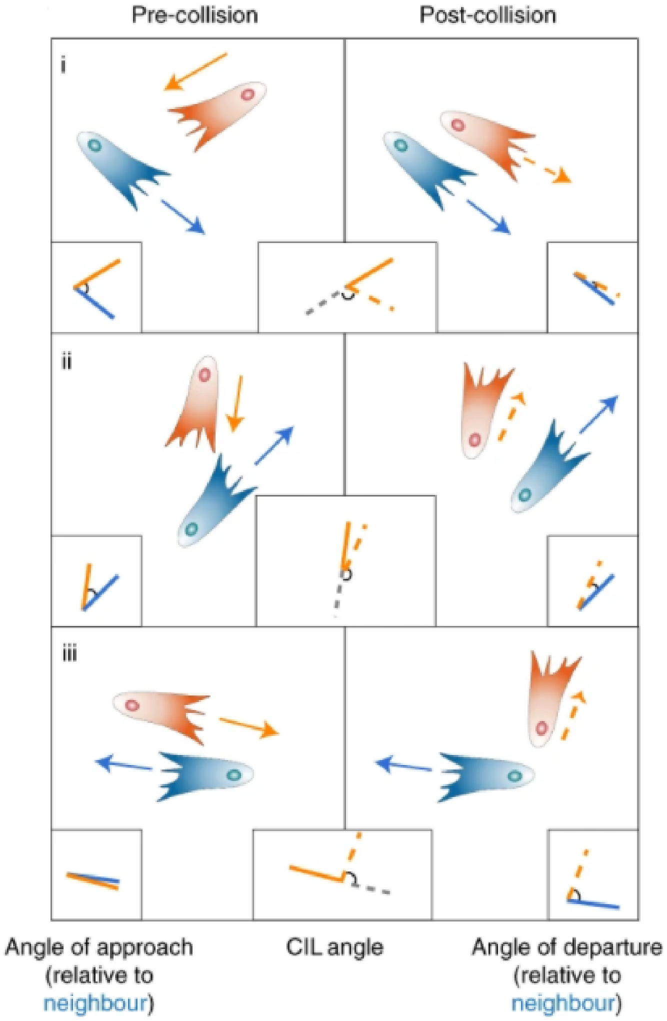


Figure 5.3: Three cell collision guidance interaction examples with representative CIL (contact inhibition of locomotion) angles.

Note. Adapted from "Extracellular matrix anisotropy is determined by TFAP2C-dependent regulation of cell collisions", by Park, D., 2019, *Nature Materials*, *19*, p. 231.

Well defined cellular mechanisms make a compelling case for the biological relevance of collision guidance mediated alignment. The transcription factor TFAP2C regulates cell collision guidance by controlling the expression of RND3 which downregulates actomyosin activity at cell-cell collision zones causing shifts in cell orientation. Without TFAP2C, cell collision guidance is disrupted, leading to uniform matrix generation [28].

Therefore, the goal of this chapter is to establish whether or not collision guidance, within the framework of the infarct ABM developed in chapters 2 and 4, can produce a tunable level of heterogeneity in biaxially loaded infarcts.

5.2 Methods

Collision guidance is implemented as a part of the migration method in the ABM. This separates it from the mechanical, chemical, and structural cues, allowing it only to change the cell's current orientation without affecting the balance or weighting of the other alignment cues. The collision guidance formula (adapated from [28]) used in this chapter is shown below.

$$\theta_i(t+1) = \tan^{-1}\left(\frac{Y_i}{X_i}\right)$$

$$X_i = \frac{w_1 X_{p_i} + \delta_{c_i} w_2 X_{c_i}}{w_1 + \delta_{c_i} w_2}, \qquad Y_i = \frac{w_1 Y_{p_i} + \delta_{c_i} w_2 Y_{c_i}}{w_1 + \delta_{c_i} w_2}$$

$$X_{p_i} = cos(\theta_i(t) + \eta), \qquad Y_{p_i} = sin(\theta_i(t) + \eta)$$

$$X_{c_i} = \delta_{c_i} \sum_{j=1}^{N_i} \frac{cos(\theta_j(t)}{N_i}, \qquad Y_{c_i} = \delta_{c_i} \sum_{j=1}^{N_i} \frac{sin(\theta_j(t)}{N_i}$$

$$\delta_{c_i} = \begin{cases} 1, & N_i \neq 0 \\ 0, & N_i = 0 \end{cases}$$

Where θ_i is the angle of the migratory cell, η is a Gaussian distributed random variable with mean 0 and variance Var(η) which simulates a persistent random walk, θ_j is the angle of a non-migratory cell involved in the collision, N_i is the number of non-migratory cell involved in the collision, and t is the time in ticks. This equation ultimately adds a weighted fraction of the stationary cell orientation vector/s to the colliding cell orientation vector, which brings the colliding cell into closer alignment with the stationary cell/s.

A parameter called collisionGuidance was also added to the model GUI. When collisionGuidance is given a value greater than zero and less than or equal to one, a new if gate in the migration method is opened. Once opened, if any cell attempts to migrate and encounters one or more cells during that attempt, the angles of all cells involved are added to the collision guidance calculation and the migratory cell's angle is updated based on the results. The value of the collisionGuidance parameter is also used in the collision guidance calculation as w_2, the weight of the cell angles being collided with. Because our model incorporates a persistent random walk when calculating the cell angle based on the structural, mechanical, and chemokine cues, the value of η was set to 0. The full migration method, with collision guidance implemented, is listed in appendix Listing E.1.

Regional heterogeneity was assessed by calculating the dot product of each grid's collagen orientation vector with every other grid's collagen orientation vector and plotting those dot products versus the distance between each pair of vectors.

5.3 Results

The ABM was run with a biaxial loading condition, 10X10 µm grid size, and various levels of collision guidance (w_2=0.05, 0.1, 0.2, 0.4, 0.8). A representative collagen angle heat map at six weeks of simulation for each level of collision guidance is shown in Figure 5.4. Fiber-fiber alignment quantification was performed to determine whether locally aligned areas of collagen were present or varied with the level of collision guidance. To reduce noise, multiple models were run for each condition and aggregated into one alignment distance quantification.

Figure 5.4 shows circumferentially aligned collagen outside the infarct and heterogeneous collagen alignment within the infarct for all values of collisionGuidance. Aligned sub-regions of collagen are not visually apparent in any of the heat maps.

Fiber-fiber alignment (Figure 5.5) is low when fibers are close together, or moderately far apart. Fiber-fiber alignment is high when the distance between them is greatest. This is an artifact of the board zone of the infarct, in which fibroblasts are always aligned perpendicular to the edge.

Fiber-fiber alignment quantification was also was performed at two weeks to determine if heterogeneity was transient (Figure 5.6). Fiber-fiber alignment did not vary with fiber-fiber distance at week two. Alignment at the boarder zone has only just begun to form.

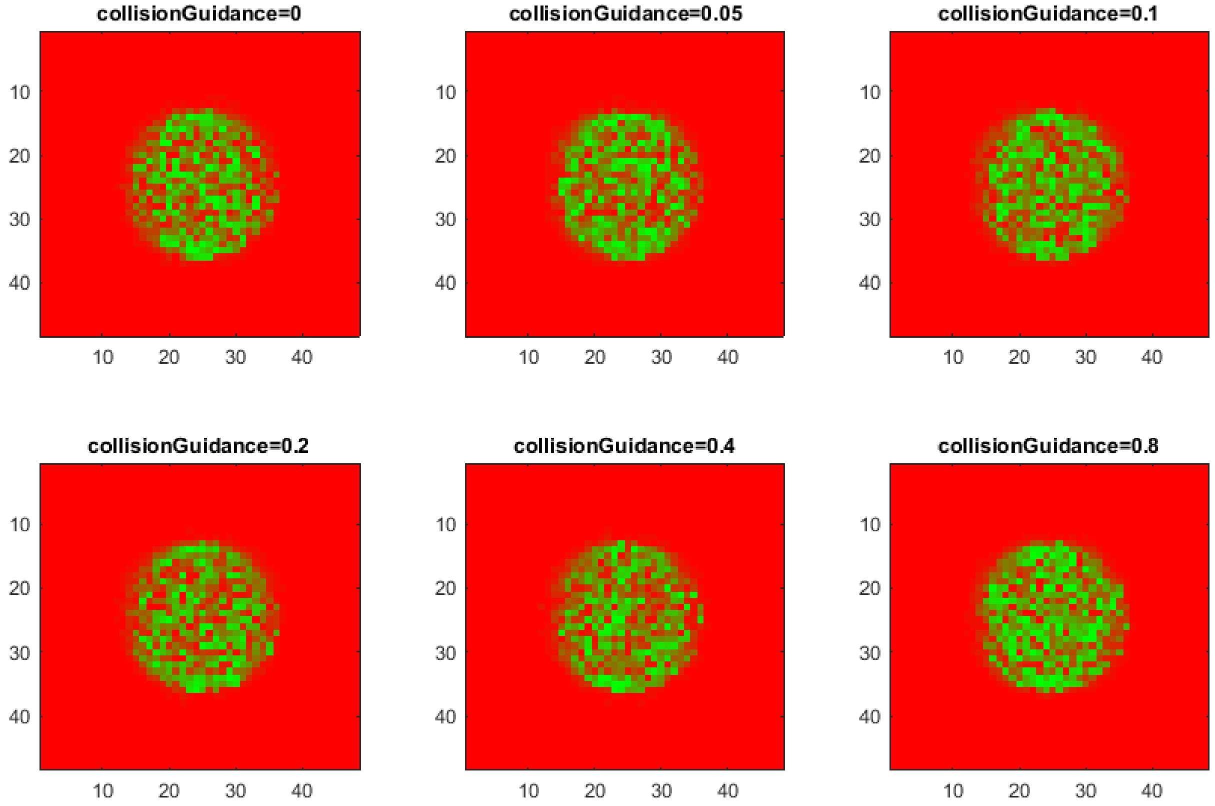


Figure 5.4: Collagen angle heat maps of biaxially loaded infarct simulations with various levels of collision guidance. Circumferentially aligned collagen (red), longitudinally aligned collagen (green).

5.4 Discussion and Conclusions

Visual and mathematical quantification of collagen alignment confirms that collagen patches in near proximity are not more likely to be aligned similarly. Therefore, collision guidance alone was not able to create locally aligned areas of collagen in biaxially loaded infarct simulations at early or late time points.

The most likely reason that we were not able to generate alignment using collision guidance, is that our model contains a separate cell angle calculation factoring other guidance cues in. Although our implementation of collision guidance modifies this guidance cue calculation by directly changing the persistent angle of the cell, the computation of a new cell angle with influence from other cues, obscures any effect of collision guidance.

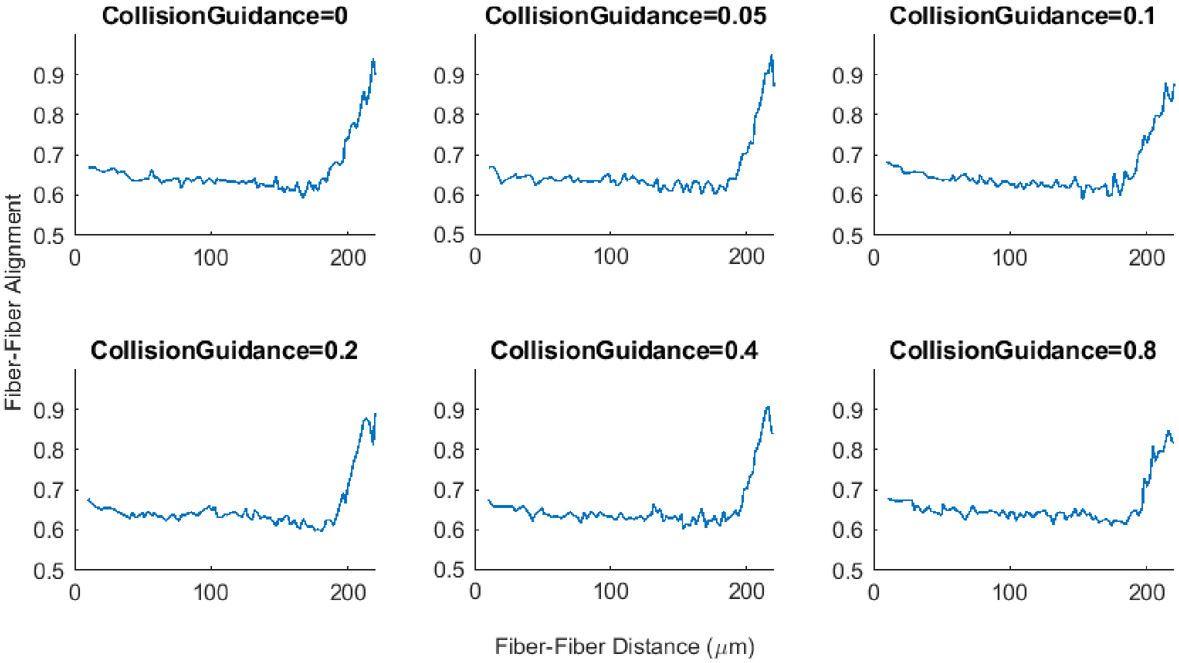


Figure 5.5: 6 week fiber-fiber alignment vs fiber-fiber distance of biaxially loaded infarct simulations with various levels of collision guidance

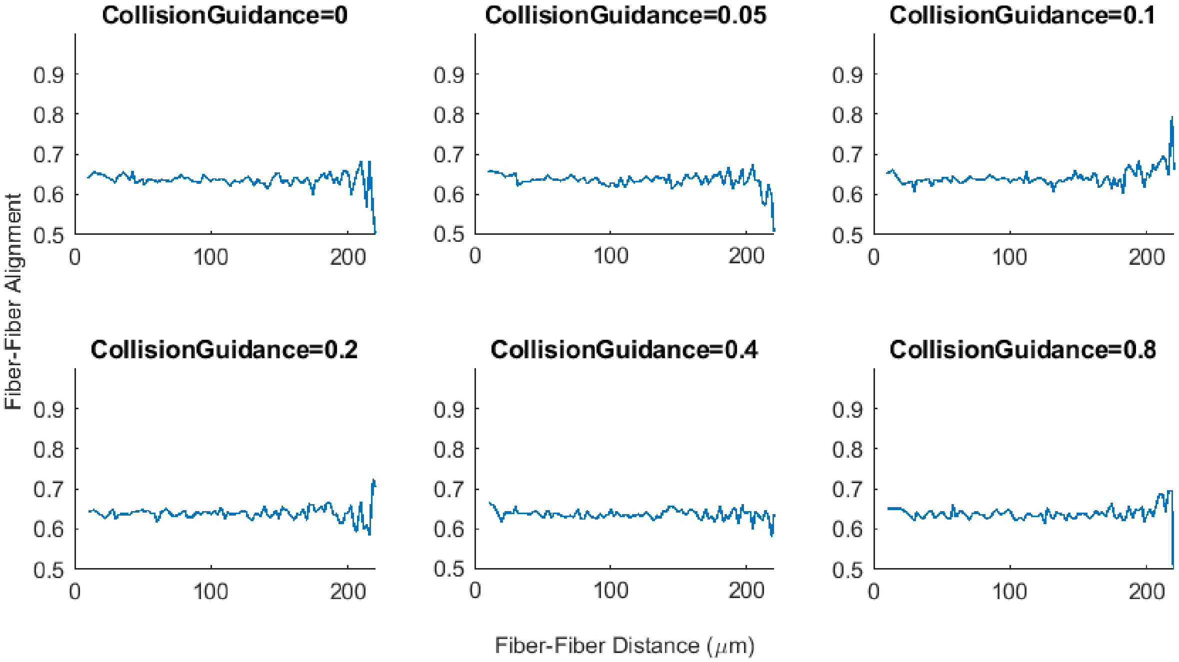


Figure 5.6: 2 week fiber-fiber alignment vs fiber-fiber distance of biaxially loaded infarct simulations with various levels of collision guidance

Another possible reason that we are unable to create local alignment, is that collision guidance only comes into play when cell density is high enough to cause collisions. Because our model no longer includes collagen rotation, and cell density does not peak until week one, a collagen structure without the influence of collision guidance may develop fully enough to prevent local alignment via collision guidance later on.

Peripheral invasion of the infarct, further limits the effects of collision guidance at early timepoints because cells are moving collectively at the same rate towards the center of the infarct, rather than against one another. Incorporating transmural migration of fibroblasts into the infarct might cause more collisions earlier between cells invading from the epicardium and infarct boarder. By entering the infarct with an alignment perpendicular to the simulation plane, transmurally migrating cells may also be able to produce local alignment without the aid of collision guidance.

This model is available for download through the Cardiac Biomechanics Group GitHub at https://github.com/cardiacbiomechanicsgroup/RepastABM. To access this specific version, check out the collisionGuidanceAdded commit.

CHAPTER 6
CONCLUSION AND DIRECTIONS FOR FUTURE WORK

6.1 Conclusion

This research outlines how we built a more computationally efficient ABM of infarct healing which predicts new and previously unmatched infarct data.

In chapter two, we showed that our new infarct healing ABM with discrete migration is not only more computationally efficient than the previous versions, but also fits experimental data just as well, even with lower spatial resolution.

In chapter three we developed a novel image processing based numerical method for calculating the diffusion profiles of signaling molecules suitable for implementation in an ABM framework and stable at a spatial and temporal scale similar to that of our ABM. We also showed that this method was more computationally efficient and similarly accurate to solutions generated by traditional methods.

Chapter four improves the ability of the model to match a wider range of experimental data by re-examining the biological plausibility of multiple features of the ABM as originally formulated. Here, we concluded that previous version of the ABM relied heavily on collagen rotation to produce collagen alignment, despite the fact that large rotations of collagen fibers seem unlikely to occur in dense, mechanically loaded infarcts in vivo. By removing rotation and increasing the influence of the mechanical cue on the cell alignment, were able to match new data on the time course of collagen alignment in longitudinally loaded infarcts without sacrificing the ability to match other data used for validation of the original ABM. The changes introduced in this chapter also produced higher cell alignments, consistent with previous studies relating matrix and cell alignment.

Finally, in chapter 5 we determined that collision guidance alone was not sufficient to produce locally aligned regions of collagen in biaxially loaded infarcts. We also discussed some features of the model which might be changed to produce the patterns of heterogeneity seen in biaxially loaded infarct with, or without, the inclusion of collision guidance.

In conclusion, in this work we created a computationally efficient and flexible infarct healing ABM and an associated diffusion method that better matches the full range of available data on collagen alignment in variously loaded healing infarcts than our previously published models.

6.2 Directions for Future Work

Based on this work, further research should be considered in the following areas.

6.2.1 Multiresolution Agent-Based Model

ABMs, including the one discussed in this text, are typically constructed on a single discrete grid. However, different biologic processes involved in infarct healing operate on different spatial scales, limiting the ability of ABMs to efficiently and accurately represent them all on a single grid. For instance, collagen fiber alignment in healing scar varies at a scale of 100-200 microns [18], while migration occurs on the scale of microns per hour. There is currently no methodology for representing multiple processes on separate scales within one ABM despite the potential benefits to accuracy and efficiency.

6.2.2 Coupling to a Finite Element Model

Finite-element models FEMs have been used extensively in the study of cardiac biomechanics to determine how mechanical forces are distributed throughout the heart during the cardiac cycle. These models have also been adapted to simulate infarction [29] and treatments like injectable biomaterials [22] and reinforcing patches [30]. Unfortunately, such models only offer us discrete snapshots of these disease states. In other words, they tell us the strains that the tissue undergoes, but not how the tissue responds to those strains. Recently, FEMs have been extended to included volumetric strain-based growth laws, giving elements the ability to geometrically respond to mechanical loading over time [31]. However, these growth laws are incapable of capturing how strains cause structural remodeling which changes the material properties of elements. FEMs operate optimally (in that they predict strain reliably and represent complex organic geometry well, while also being stable and efficient) at larger scales (elements on the order of $mm^2 - cm^2$). FEMs also follow a continuum assumption, in which each element is made of a continuous substance as opposed to individual constituent parts. Spatial scale and continuum assumption together, mean that FEMs are unable to simulate the creation of heterogeneity, which is produced through stochastic processes at the cellular level.

ABMs and FEMs are suited to coupling for the purpose of investigating structural-mechanical feedback. Our group has already coupled an ABM and FEM to investigate this feedback phenomena. However, only collagen content (which was used to determine tissue stiffness in an isotropic material model) was fed from the ABM to the FEM [32]. Creating a similar coupled model with structural-mechanical feedback which includes collagen alignment and distribution information may help us to predict the collagen properties of infarct scar treated with various spatially targeted therapies.

6.2.3 Simulating Non-Invasive Therapies Using the Agent-Based Model

Our recent patch reinforcement study showed that scars with longitudinally aligned collagen fibers could be created [13], and other studies from our group have shown that anisotropic infarct scar that is selectively stiff in the longitudinal direction improves pump function [12] [30]. However, this method of aligning collagen is invasive, and is not a treatment which can be adjusted over time as the heart changes. “Less invasive” methods for aligning collagen have been studied widely in orthopedic medicine, primarily for the purpose of tendon repair. Magnetic fields have been used to direct fibroblast alignment in vitro, resulting in aligned collagen production [33]. Electric fields have been noted to direct cell alignment. Lui et al. created aligned fibroblast cultures with biphasic pulse direct current electric fields [34]. Similar electrical stimulation has even been proven to be possible in a large animal model of MI. While this study did not characterize collagen orientation, they did note significant reduction in ventricular dilation in pigs with MI who received infarct pulse electrical stimulation therapy [35]. Ultrasound stimulation, perhaps the most studied in tendon fibroblasts, also produces changes in fibroblast behavior and collagen production. Fu et al. noted that ultrasound stimulation increases collagen alignment in injured tendons when administered at early time points. Ultrasound stimulation also significantly increases fibroblast collagen synthesis and proliferation when administered in acute models of tendon injury both in vitro [36] and in vivo [37].

Now that the infarct ABM has been streamlined and moved to a more flexible platform, it is feasible to incorporate new cell alignment cues, including those mentioned above, to investigate new treatments which address infarct scar collagen alignment.

Appendices

APPENDIX A

KEY METHODS IN THE REPAST ABM

Listing A.1: Step Method: details the behaviors of each fibroblast agent at each tick of the model

```
@ScheduledMethod(start = 0.5, interval = 0.5, priority = 1)
public void step() {
  final woundABMContextSim woundABMSpace = (woundABMContextSim)
     ContextUtils.getContext(this);
  final Grid<?> grid = (Grid<?>) woundABMSpace.getProjection("Cell
     Grid");
  final GridValueLayer chemokine = (GridValueLayer)
     woundABMSpace.getValueLayer("Chemokine");
  final GridPoint pt = grid.getLocation(this);
  final double effectiveChemokine = (chemokine.get(pt.getX(),
     pt.getY()) - concMin)/(concMax - concMin);

  // Define chemokine dependent mitosis to ensure cells are always
     coming from the outside of wound
  this.mitosisTime = effectiveChemokine * (gMitosisTime -
     apoptosisTime) + apoptosisTime;

  // Define chemokine dependent migration to speed up wound closure
  this.cellSpeed = effectiveChemokine * (cellSpeedMax - cellSpeedMin)
     + cellSpeedMin;

  // Get location
  final int x = pt.getX();
```

```
final int y = pt.getY();

// Check for mitosis
if (mitosisAge >= mitosisTime) {
  mitosisAge = randomClock(); // Quiescent cell sets mitosis age to
     random number
  mitose(woundABMSpace, grid, x, y);

} else {

  // Update cell orientation
  guidanceCue(woundABMSpace, x, y);

  // Check for migration
  if (migrationDistance >= gridUnitSize) {
    final int grids = (int)
       Math.floor(migrationDistance/gridUnitSize);
    this.migrationDistance = migrationDistance - gridUnitSize *
       grids;
    migrate(grid, x, y, grids);
  }

  // Get grids covered by the cell
  List<GridPoint> coveredSites = cellCoverage(x, y);

  // Deposit, degrade, and rotate collagen in the grids beneath the
     cell
  for (GridPoint site : coveredSites) {
```

```
        // Get site x and y coordinates
        final int siteX = site.getX();
        final int siteY = site.getY();

        // Retrieve structural value layers
        final ArrayList<GridValueLayer> collagenLayers =
           (ArrayList<GridValueLayer>)woundABMSpace.
           GridValueLayerList();
        final ArrayList<GridValueLayer> nonColLayers =
           (ArrayList<GridValueLayer>)woundABMSpace.
           nonColGridValueLayerList();

        // Check for rotation, degradation, deposition
        collagenRotation(siteX, siteY, chemokine, collagenLayers);
        collagenDeposition(siteX, siteY, chemokine, collagenLayers,
           depositionTime);
        matrixDegradation(siteX, siteY, chemokine, collagenLayers,
           nonColLayers, degradationTime);
     }
   }

   // Check for apoptosis
   if (apoptosisAge >= apoptosisTime) {
     woundABMSpace.remove(this);
   }

   // Increment the counters
   apoptosisAge = apoptosisAge + timeStep;
   mitosisAge = mitosisAge + timeStep;
```

```
  migrationDistance = migrationDistance + cellSpeed * timeStep;
}
```

APPENDIX B

DIFFUSION IN A 2D PLANE, AN IMPLEMENTATION OF THE IRF METHOD IN MATLAB

Listing B.1: caption

```
% Units: s, um, nM


% Author: Arlynn C Baker
% Created: 2020/01/06
% Modified: 2020/11/06


close all; clear; clc;


% Descretization specifications
ds = 20;   % um
dt = [0.5,5,30,120,400,900];
   %[0.1,0.5,1,2.5,5,10,20,30,50,80,120,150,180,300,400,600,900}
tEnd = 3600;% s


% Diffusion, produciton, and consumption parameters
D = [10,1]; % um^2/s % Note: 0.16 in J Phys after geometric adjustment
   [1000,100,10,1]
kGen = 0.01; % nM/s
kDeg = 0.001; % 1/s


% Define space and create generation and degradation maps
x = -5000:ds:5000;
```

```
y = x;

% Plot styles
plot_irf = ['rs';'r*';'ro';'rd'];
plot_fdm = ['bs';'b*';'bo';'bd'];

% Preallocate
md_irf_min = zeros(1,length(D));
tElapsed_irf_min = zeros(1,length(D));
dt_irf_min = zeros(1,length(D));
md_fdm_min = zeros(1,length(D));
tElapsed_fdm_min = zeros(1,length(D));
dt_fdm_min = zeros(1,length(D));

for i = 1:length(D)
   input = zeros(length(x),length(y));
   [genLocs,solution] = getInput(x,y,i);

   % Calculate diffusion profiles with various time steps and
      calculate the mean differnce from the FE solution.
   C_irf = zeros(length(x),length(y),length(dt)); C_fdm =
      zeros(length(x),length(y),length(dt));
   tElapsed_irf = zeros(1,length(dt)); tElapsed_fdm =
      zeros(1,length(dt));
   md_irf = zeros(1,length(dt)); md_fdm = zeros(1,length(dt));
   irf = cell(1,length(dt));
   b_irf = 0; b_fdm = 0;
   md_irf_min(i) = 100;
   for j = length(dt):-1:1
```

```
   if b_irf == 0
   [C_irf(:, :, j), irf{j}, tElapsed_irf(j)] =
      IterativeIRFDiffusion(input, dt(j), ds, D(i), kGen, genLocs,
      kDeg, tEnd);
   md_irf(j) = mean(mean(abs(C_irf(:, :, j) - solution)));
   end
   if isnan(C_irf(:, :, j))
      b_irf = j;
   end
   if md_irf(j)<md_irf_min(i)
      md_irf_min(i) = md_irf(j);
      tElapsed_irf_min(i) = tElapsed_irf(j);
      dt_irf_min(i) = dt(j);
   end
end
for j = 1:length(dt)
   if b_fdm == 0
      [C_fdm(:, :, j), tElapsed_fdm(j)] =
         FiniteDifferenceDiffusion(input, dt(j), ds, D(i), kGen,
         genLocs, kDeg, tEnd);
      md_fdm(j) = mean(mean(abs(C_fdm(:, :, j) - solution)));
   end
   if min(min(C_fdm(:, :, j)))<= 0
      b_fdm = j;
   end
end

% Correct ill conditioned solutions
```

```
    [md_irf, tElapsed_irf, dt_irf] = trimIllConditioned(md_irf,
       tElapsed_irf, dt, C_irf, 10);
    [md_fdm, tElapsed_fdm, dt_fdm] = trimIllConditioned(md_fdm,
       tElapsed_fdm, dt, C_fdm, 10);
    md_fdm_min(i) = md_fdm(end);
    tElapsed_fdm_min(i) = tElapsed_fdm(end);
    dt_fdm_min(i) = dt_fdm(end);

    % Plot model acuuracy by dt
    figure(1); hold on;
    plot(dt_irf, md_irf, 'r-', dt_fdm, md_fdm, 'b-');
%    plot(dt_irf_min, md_irf_min, plot_irf(i, :), dt_fdm(end),
    md_fdm(end), plot_fdm(i, :));

    % Plot model efficency by dt
    figure(2); hold on;
    plot(dt_irf, tElapsed_irf, 'r-', dt_fdm, tElapsed_fdm, 'b-');
%    plot(dt_irf_min, tElapsed_irf_min, plot_irf(i, :), dt_fdm(end),
    tElapsed_fdm(end), plot_fdm(i, :));
end

% Format plots
figure(1); set(gca, 'XScale', 'log');
legend('IRF 1000', 'FDM 1000', 'IRF 100', 'FDM 100', 'IRF 10', 'FDM
    10', 'IRF 1', 'FDM 1');
ylabel('MAD (nM)'); xlabel('\Deltat (s)');
figure(2); set(gca, 'YScale', 'log', 'XScale', 'log');
legend('IRF 1000', 'FDM 1000', 'IRF 100', 'FDM 100', 'IRF 10', 'FDM
    10', 'IRF 1', 'FDM 1');
```

```
ylabel('Computation Time (s)'); xlabel('\Deltat (s)');

function [md, tElapsed, dt] = trimIllConditioned(md, tElapsed, dt, C,
   maxC)
nans = find(isnan(md));
negs = find(min(min(C))<0);
maxs = find(max(max(C)) >= maxC);
zs = find(max(max(C)) == 0);
remove = flip(unique([nans';negs;maxs;zs]));
for k = 1:length(remove)
   md(remove(k)) = [];
   tElapsed(remove(k)) = [];
   dt(remove(k)) = [];
end
end

function [input, solution] = getInput(x,y,inputNum)
% Get map of production and consumption from PDE toolkit geometry files
if inputNum == 1
   load('ConcPDE_Rand1X1cm_D1000.mat','p','t','u','gd');
elseif inputNum == 2
   load('ConcPDE_Rand1X1cm_D100.mat','p','t','u','gd');
elseif inputNum == 3
   load('ConcPDE_Rand1X1cm_D10.mat','p','t','u','gd');
elseif inputNum == 4
   load('ConcPDE_Rand1X1cm_D1.mat','p','t','u','gd');
else
   load('ChemoConcPDE_Random.mat','p','t','u','gd');
end
```

```
solution = tri2grid(p,t,u,x,y);

[~,h] = size(gd);
edge = 0;
gList = [];
for q = 1:h
   if gd(1,q) == 1
      edge = edge + 1;
      gList(1:3,edge) = gd(2:4,q);
   end
end
input = zeros(length(x),length(y));
for r = 1:length(y)
   for n = 1:length(x)
      for w = 1:edge
         if sqrt((x(n) - gList(1,w))^2 + (y(r) - gList(2,w))^2)<=
            gList(3,w)
            input(r,n) = 1;
         end
      end
   end
end
end
```

APPENDIX C

INITIALIZE STRUCTURAL LAYER BINS METHOD

Listing C.1: initializeStructuralLayerBins: initialize the collagen and non collagen angle bin grid value layers

```
@ScheduledMethod(start = 0, priority = 1)
public void initializeStructuralLayerBins() {

  // Set distribution across all bins based on fiber distribution type
  double[] colDistrib;
  if (initialColDist.equals("Circumferential")) { // (rho = 0.75;
     Theta = 0)
    colDistrib = new double[] {.000043, .000079, .000174, .000377,
       .000782, .001542, .002885, .005124, .008638, .013821, .020988,
       .03025, .041382, .053729, .06621, .07744, .085965, .090574,
       .090574, .085965, .07744, .06621, .053729, .041382, .03025,
       .020988, .013821, .008638, .005124, .002885, .001542, .000782,
       .000377, .000174, .000079, .000043};
  } else if (initialColDist.equals("Longitudinal")) { // (rho = 0.75;
     Theta = 90)
    colDistrib = new double[] {.0912, .0865, .0778, .0664, .0537,
       .0412, .0300, .0207, .0136, .0084, .0050, .0028, .0015, .0007,
       .0004, .0002, .0001, .0000, .0000, .0001, .0002, .0004, .0007,
       .0015, .0028, .0050, .0084, .0136, .0207, .0300, .0412, .0537,
       .0664, .0778, .0865, .0912};
  } else { // Uniform
```

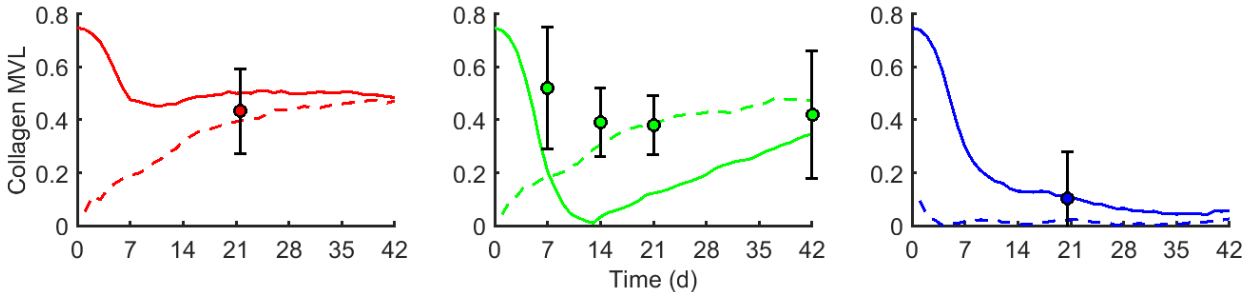
```
  colDistrib = new double [] {.027778, .027778, .027778, .027778,
     .027778, .027778, .027778, .027778, .027778, .027778, .027778,
     .027778, .027778, .027778, .027778, .027778, .027778, .027778,
     .027778, .027778, .027778, .027778, .027778, .027778, .027778,
     .027778, .027778, .027778, .027778, .027778, .027778, .027778,
     .027778, .027778, .027778, .027778};
}

// Set distribution across all bins based on fiber distribution type
double[] nonColDistrib;
if (initialNonColDist.equals("Circumferential")) { // (rho = 0.75;
   Theta = 0)
  nonColDistrib = new double[] {.000043, .000079, .000174, .000377,
     .000782, .001542, .002885, .005124, .008638, .013821, .020988,
     .03025, .041382, .053729, .06621, .07744, .085965, .090574,
     .090574, .085965, .07744, .06621, .053729, .041382, .03025,
     .020988, .013821, .008638, .005124, .002885, .001542, .000782,
     .000377, .000174, .000079, .000043};
} else if (initialNonColDist.equals("Longitudinal")) { // (rho =
   0.75; Theta = 90)
  nonColDistrib = new double[] {.0912, .0865, .0778, .0664, .0537,
     .0412, .0300, .0207, .0136, .0084, .0050, .0028, .0015, .0007,
     .0004, .0002, .0001, .0000, .0000, .0001, .0002, .0004, .0007,
     .0015, .0028, .0050, .0084, .0136, .0207, .0300, .0412, .0537,
     .0664, .0778, .0865, .0912};
} else { // Uniform
  nonColDistrib = new double [] {.027778, .027778, .027778, .027778,
     .027778, .027778, .027778, .027778, .027778, .027778, .027778,
     .027778, .027778, .027778, .027778, .027778, .027778, .027778,
```

```
        .027778, .027778, .027778, .027778, .027778, .027778, .027778,
        .027778, .027778, .027778, .027778, .027778, .027778, .027778,
        .027778, .027778, .027778, .027778};
}
// Scale fiber content (from 2.5) to grid size
double scalingFactor = Math.pow(gridUnitSize/2.5,2);

// Set each grid value layer
ArrayList<GridValueLayer> collagenLayers =
    (ArrayList<GridValueLayer>) GridValueLayerList();
ArrayList<GridValueLayer> nonColLayers = (ArrayList<GridValueLayer>)
    nonColGridValueLayerList();
int binNum = collagenLayers.size();

double woundGridNum = (double) (woundRadius/gridUnitSize);
double wCenter = gridWidth/2;
double hCenter = gridHeight/2;
for (int y = 0; y < gridHeight; y++) {
  for (int x = 0; x < gridWidth; x++) {
    if (includeNonCol == true) {
      if (Math.sqrt(Math.pow((x - wCenter), 2) + Math.pow((y -
          hCenter), 2)) <= woundGridNum) {
        for (int i = 0; i < binNum; i++) {
          collagenLayers.get(i).set(0, x, y);
        }
      } else {
        for (int i = 0; i < binNum; i++) {
          nonColLayers.get(i).set(25 * 36 * scalingFactor *
              nonColDistrib[i], x, y);
```

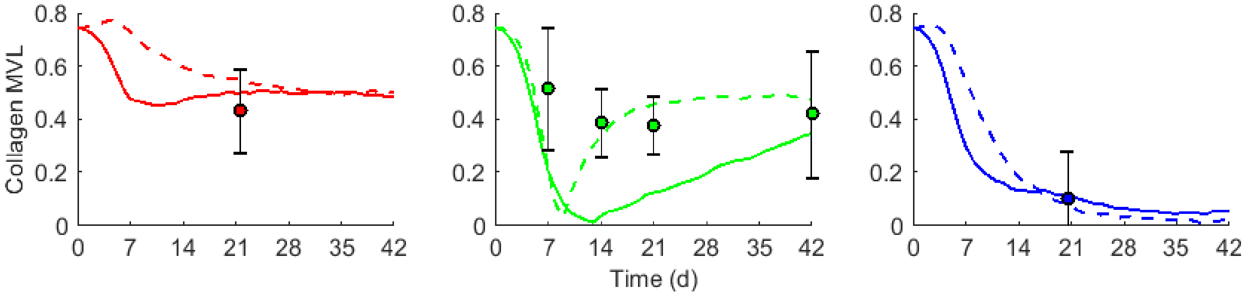
```
          }
        }
      } else {
        for (int i = 0; i < binNum; i++) {
          nonColLayers.get(i).set(0, x, y);
        }
      }
      if (includeWoundCollagen == true) {
        for (int i = 0; i < binNum; i++) {
          collagenLayers.get(i).set(scalePercent * 36 * scalingFactor
             * colDistrib[i], x, y);
        }
      } else {
        if (Math.sqrt(Math.pow((x - wCenter), 2) + Math.pow((y -
           hCenter), 2)) <= woundGridNum) {
          for (int i = 0; i < binNum; i++) {
            collagenLayers.get(i).set(0, x, y);
          }
        } else {
          for (int i = 0; i < binNum; i++) {
            collagenLayers.get(i).set(scalePercent * 36 *
               scalingFactor * colDistrib[i], x, y);
          }
        }
      }
    }
  }
}
```

APPENDIX D

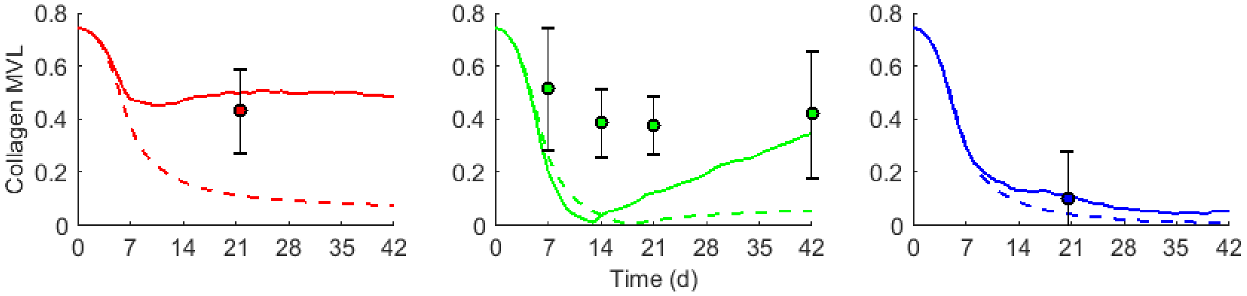
ADDITIONAL COMBINATIONS OF MODEL CHANGES



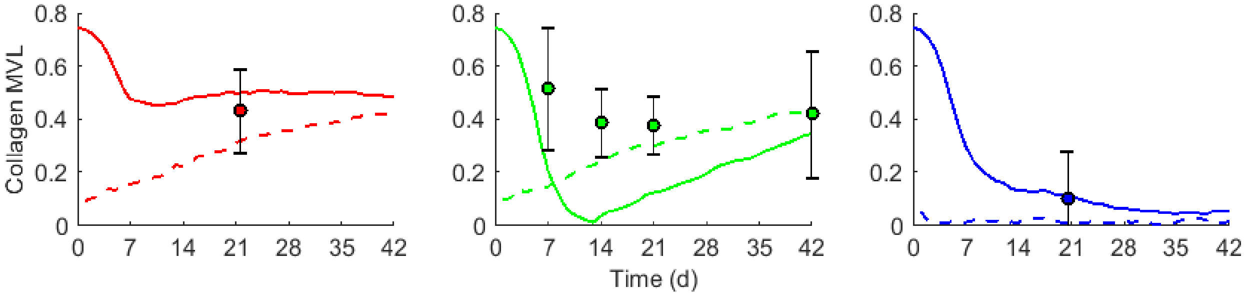
(a) Remove Non Collagen + No Initial Collagen



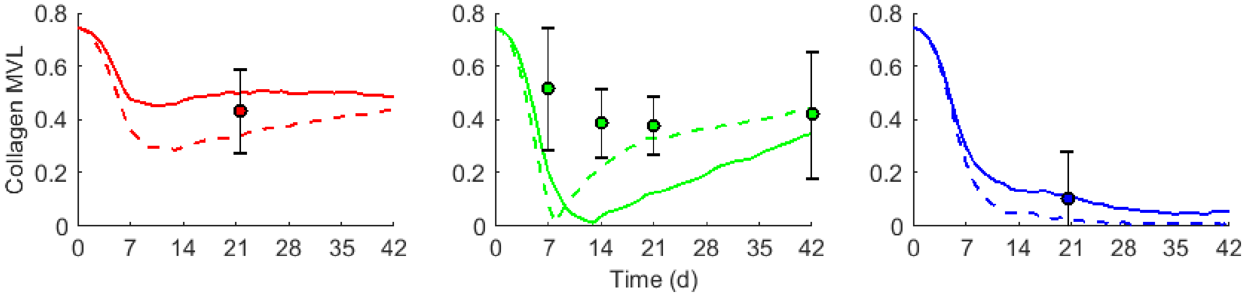
(b) Remove Non Collagen + Scale Structural Cue



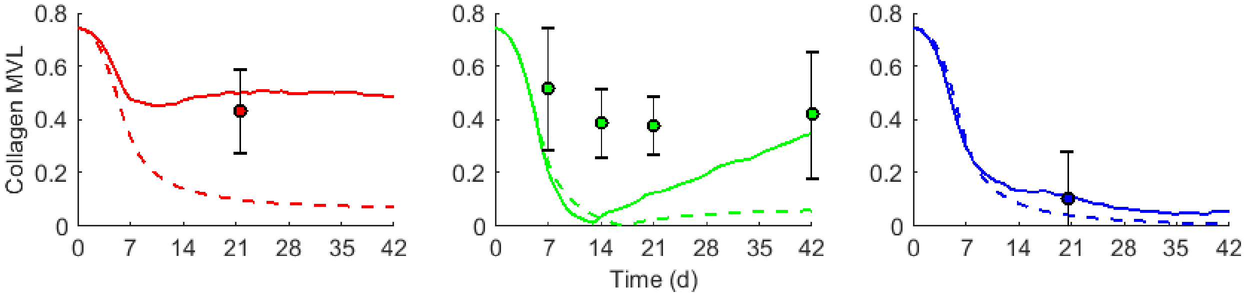
(c) Remove Non Collagen + No Collagen Rotation



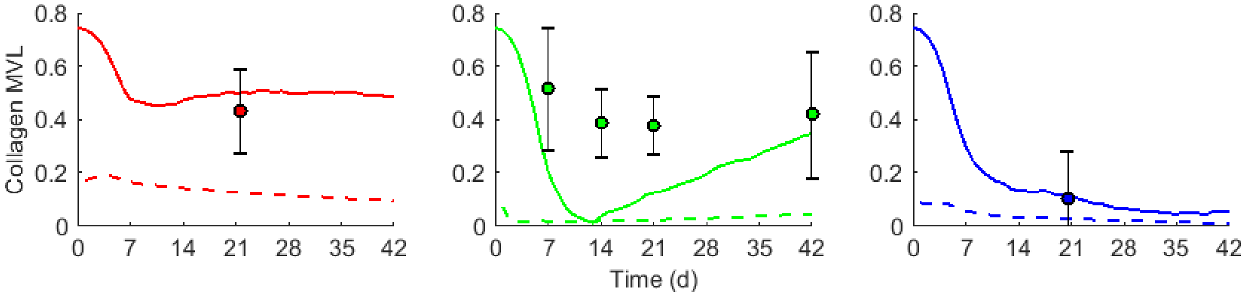
(d) Uniform Non Collagen + No Initial Collagen



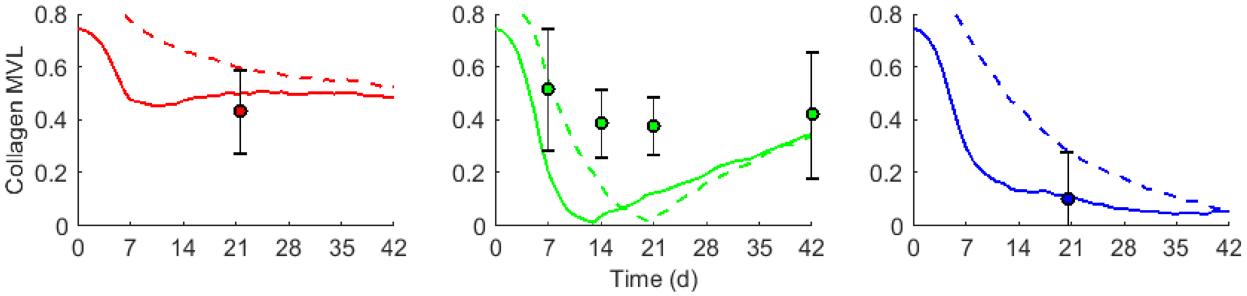
(e) Uniform Non Collagen + Scale Structural Cue



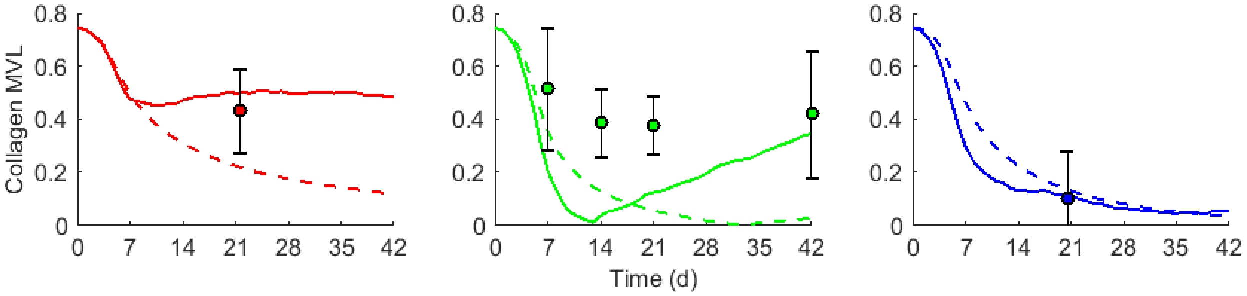
(f) Uniform Non Collagen + No Collagen Rotation



(g) No Initial Collagen + No Collagen Rotation

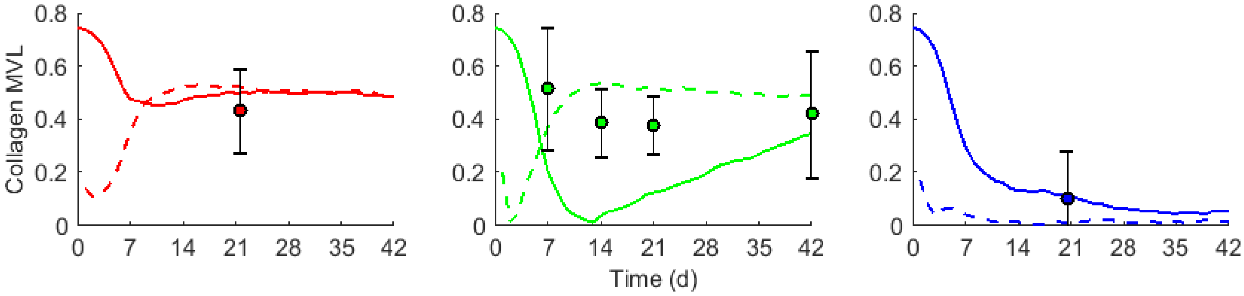


(h) No Initial Collagen + Scale Structural Cue

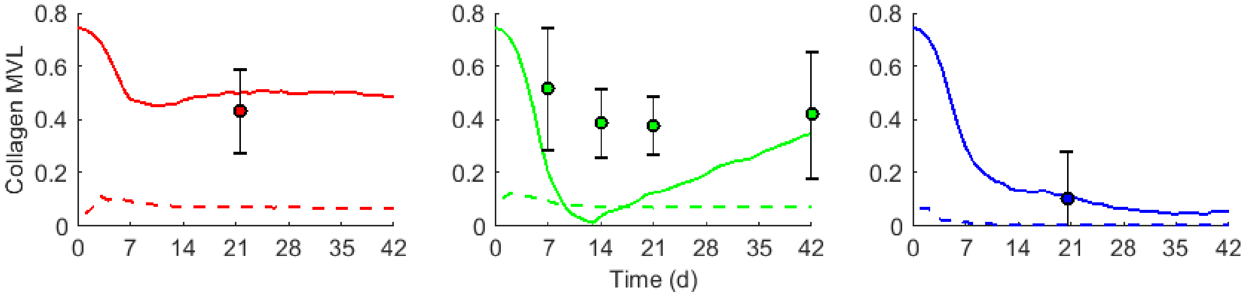


(i) Scale Structural Cue + No Collagen Rotation

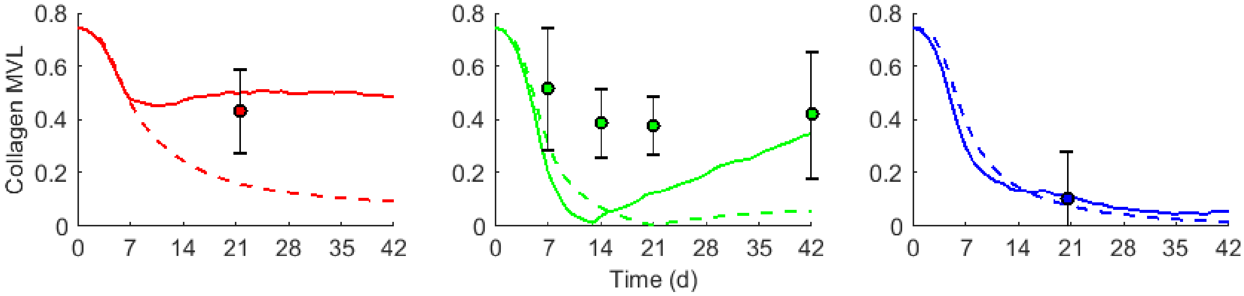
Figure D.1: Second order model change combinations: Collagen mean vector length of the standard (solid line) and modified (dashed line) ABM with experimental data (error bars). Circumferential (red), longitudinal (green), and biaxial (blue) results shown.



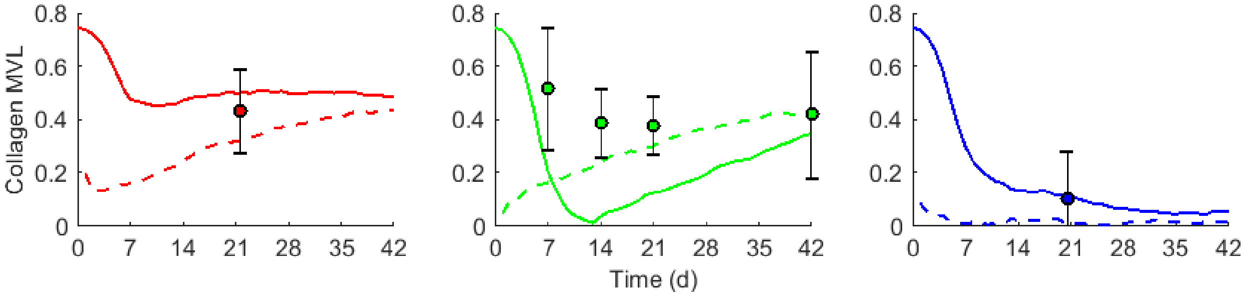
(a) Remove Non Collagen + No Initial Collagen + Scale Structural Cue



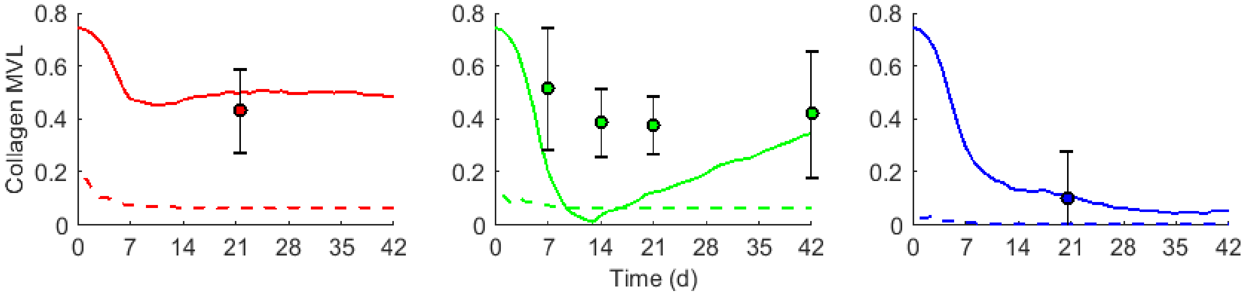
(b) Remove Non Collagen + No Initial Collagen + No Collagen Rotation



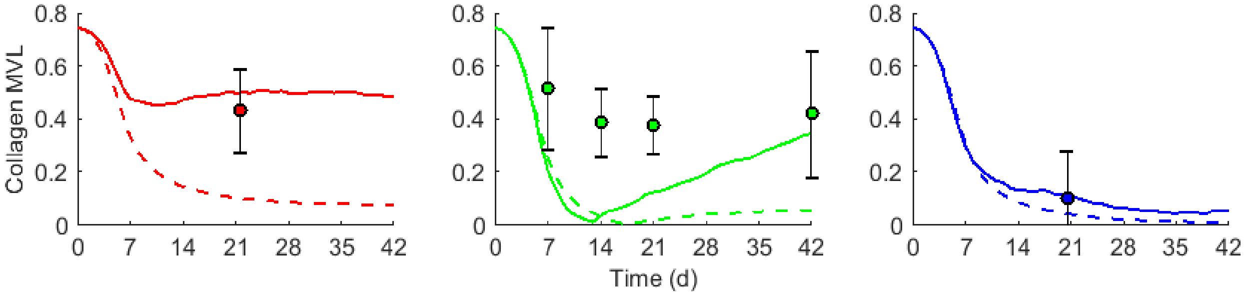
(c) Remove Non Collagen + Scale Structural Cue + No Collagen Rotation



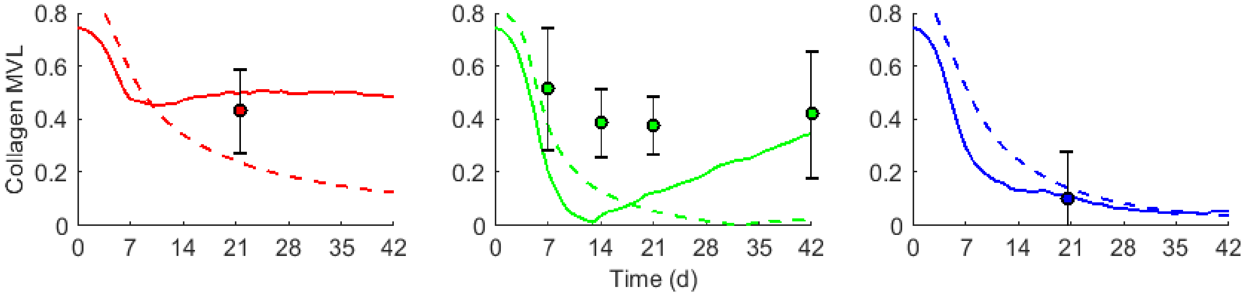
(d) Uniform Non Collagen + No Initial Collagen + Scale Structural Cue



(e) Uniform Non Collagen + No Initial Collagen + No Collagen Rotation

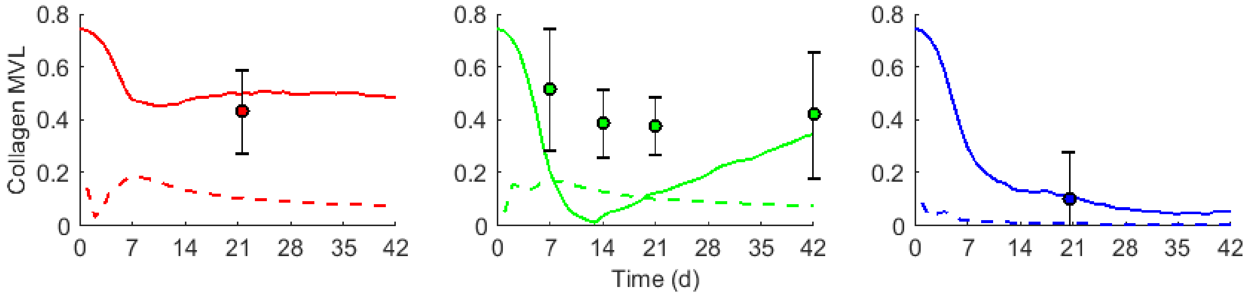


(f) Uniform Non Collagen + Scale Structural Cue + No Collagen Rotation

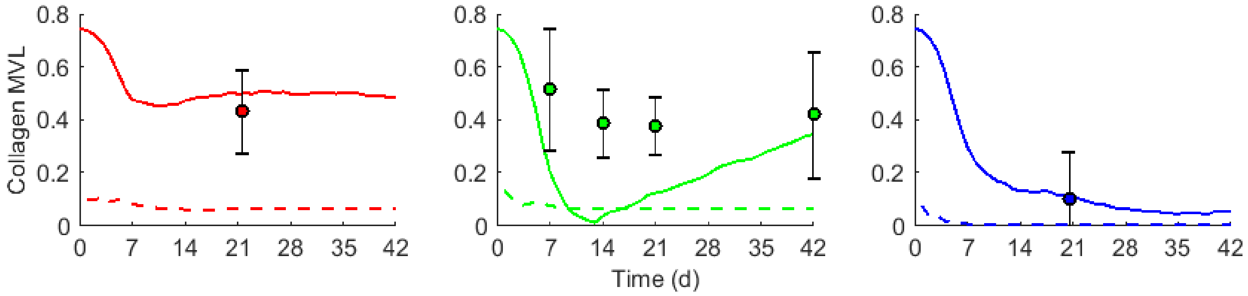


(g) No Initial Collagen + Scale Structural Cue + No Collagen Rotation

Figure D.2: Third order model change combinations: Collagen mean vector length of the standard (solid line) and modified (dashed line) ABM with experimental data (error bars). Circumferential (red), longitudinal (green), and biaxial (blue) results shown.



(a) Remove Non Collagen + No Initial Collagen + Scale Structural Cue + No Collagen Rotation



(b) Uniform Non Collagen + No Initial Collagen + Scale Structural Cue + No Collagen Rotation

Figure D.3: Fourth order model change combinations: Collagen mean vector length of the standard (solid line) and modified (dashed line) ABM with experimental data (error bars). Circumferential (red), longitudinal (green), and biaxial (blue) results shown.

APPENDIX E

COLLISION GUIDANCE IMPLEMENTED WITHING THE MIGRATION METHOD

Listing E.1: Migration Method

```
// Migration method
@SuppressWarnings({ "rawtypes", "unchecked" })
private void migrate(Grid grid, int x, int y, int extent) {

  // Retrieve and correct cell angle
  double angle = this.angleSelection;
  if (angle < 0) {
    angle = angle + 360.0;
  }

  // Search for a migration site starting from farthest to nearest
  final int[] angleWindow = new int[] {45,30,18,15,13,11,10,9,8,7};
  while (extent > 0) {

    // Get list of sites within extent and the angle window that are
       available for migration
    // Note: final List<GridPoint> emptySites = getMigrationSite(grid,
       x, y, extent, angle);
    double smallestAngle = 360;
    int siteX = -1;
    int siteY = -1;
```

```
// Iterate through potential sites
List<GridPoint> emptySites = new ArrayList<GridPoint>();
for (int i = -extent; i <= extent; i++) {
  int xCoor = x + i;
  for (int j = -extent; j <= extent; j++) {
    int yCoor = y + j;

    // Check if the site is within the radial distance
    final double dist = Math.sqrt(Math.pow(xCoor - x,2) +
       Math.pow(yCoor - y,2));
    if ((dist <= extent && dist > extent - 1) || extent == 1) {

      // Correct out of range site angles
      double siteAngle = Math.toDegrees(Math.atan2(yCoor - y,
         xCoor - x));
      if (siteAngle < 0) {
        siteAngle = siteAngle + 360.0;
      }
      if (angle > 270 && siteAngle == 0) {
        siteAngle = 360;
      }

      // Check if the site is the closest site within the angle
         window
      final double siteAngleDiff = Math.abs(siteAngle - angle);
      if (siteAngleDiff <=angleWindow[extent - 1] &&
         siteAngleDiff <= smallestAngle) {

        // Wrapped space
```

```
        if (xCoor < 0) {
          xCoor = xCoor + gridWidth;
        } else if (xCoor >= gridWidth) {
          xCoor = xCoor - gridWidth;
        }
        if (yCoor < 0) {
          yCoor = yCoor + gridHeight;
        } else if (yCoor >= gridHeight) {
          yCoor = yCoor - gridHeight;
        }

        // Record site
        smallestAngle = siteAngleDiff;
        siteX = xCoor;
        siteY = yCoor;
      }
    }
  }
}

// Check if an appropriate site has been identified
if (siteX == -1 || siteY == -1) {

// Check if the appropriate site’s surrounding area is free
} else if (!grid.getObjectsAt(siteX, siteY).iterator().hasNext()) {

  if (extent < gridDiameter) {
    if (findNumNeighbor(grid, siteX, siteY) <= 1) {
      emptySites.add(new GridPoint(siteX, siteY));
```

```
    }
  } else {
    if (findNumNeighbor(grid, siteX, siteY) < 1) {
      emptySites.add(new GridPoint(siteX, siteY));
    }
  }
}
if (!emptySites.isEmpty()) {
  // Select a random migration site from the list of those
     available to move to
  final int migrationSiteIdx =
     RandomHelper.nextIntFromTo(0,(emptySites.size() - 1));
  final GridPoint migrationSite =
     emptySites.get(migrationSiteIdx);
  final int desiredX = migrationSite.getX();
  final int desiredY = migrationSite.getY();
  // Move to new location
  grid.moveTo(this, desiredX, desiredY);
  this.pastMigration = 1;
  extent = 0;

// Abort move if no sites are free
} else if (extent == 1) {
  this.pastMigration = 0;
  this.migrationDistance = 0;

  // Implement collision guidance
  double angleRad = Math.toRadians(angle);
  List<Object> neighbors = findNeighbors(grid, siteX, siteY);
```

```
  if (neighbors.size() >= 1) { // Note: count collisions
    woundABMContextSim.addCollision();
  }
  if (collisionGuidance > 0 && collisionGuidance <= 1) {
    double Xci = 0;
    double Yci = 0;
    for (Object obj:neighbors) {
      double thetaj =
         Math.toRadians(((CellAgentSim)obj).angleSelection);
      if (Math.abs(thetaj - angleRad)%Math.PI < Math.PI/2) {

      } else {
        thetaj = thetaj + Math.PI/2;
      }
      Xci = Xci + Math.cos(thetaj);
      Yci = Yci + Math.sin(thetaj);
    }
    double Xpi = Math.cos(angleRad);
    double Ypi = Math.sin(angleRad);
    double w1 = (1 - collisionGuidance);
    double w2 = collisionGuidance;
    double Xi = (w1 * Xpi + w2 * Xci)/(w1 + w2);
    double Yi = (w1 * Ypi + w2 * Yci)/(w1 + w2);
    double thetai = Math.atan2(Yi,Xi);
    this.angleSelection = Math.toDegrees(thetai);
  }
}

// Reduce search distance
```

```
    extent = extent - 1;
  }
}
```

REFERENCES

[1] E. J. Benjamin, P. Muntner, A. Alonso, M. S. Bittencourt, C. W. Callaway, A. P. Carson, A. M. Chamberlain, A. R. Chang, S. Cheng, S. R. Das, *et al.*, "Heart disease and stroke statistics-2019 update a report from the american heart association," *Circulation*, 2019.

[2] S. Ud-Din and A. Bayat, "Non-animal models of wound healing in cutaneous repair: In silico, in vitro, ex vivo, and in vivo models of wounds and scars in human skin," *Wound Repair and Regeneration*, vol. 25, no. 2, pp. 164–176, 2017.

[3] O. Dewald, G. Ren, G. D. Duerr, M. Zoerlein, C. Klemm, C. Gersch, S. Tincey, L. H. Michael, M. L. Entman, and N. G. Frangogiannis, "Of mice and dogs: Species-specific differences in the inflammatory response following myocardial infarction," *The American journal of pathology*, vol. 164, no. 2, pp. 665–677, 2004.

[4] N. G. Frangogiannis, "The inflammatory response in myocardial injury, repair, and remodelling," *Nature Reviews Cardiology*, vol. 11, no. 5, pp. 255–265, 2014.

[5] E. P. Daskalopoulos, B. J. Janssen, and W. M. Blankesteijn, "Myofibroblasts in the infarct area: Concepts and challenges," *Microscopy and Microanalysis*, vol. 18, no. 1, pp. 35–49, 2012.

[6] M. Doi, S. Kusachi, T. Murakami, Y. Ninomiya, M. Murakami, M. Nakahama, K. Takeda, I. Komatsubara, I. Naito, and T. Tsuji, "Time-dependent changes of decorin in the infarct zone after experimentally induced myocardial infarction in rats: Comparison with biglycan," *Pathology-Research and Practice*, vol. 196, no. 1, pp. 23–33, 2000.

[7] J. W. Holmes, T. K. Borg, and J. W. Covell, "Structure and mechanics of healing myocardial infarcts," *Annu. Rev. Biomed. Eng.*, vol. 7, pp. 223–253, 2005.

[8] J. Hartupee and D. L. Mann, "Neurohormonal activation in heart failure with reduced ejection fraction," *Nature Reviews Cardiology*, vol. 14, no. 1, pp. 30–38, 2017.

[9] C. W. Yancy, M. Jessup, B. Bozkurt, J. Butler, D. E. Casey, M. M. Colvin, M. H. Drazner, G. Filippatos, G. C. Fonarow, M. M. Givertz, *et al.*, "2016 acc/aha/hfsa focused update on new pharmacological therapy for heart failure: An update of the 2013 accf/aha guideline for the management of heart failure: A report of the american college of cardiology/american heart association task force on clinical practice guidelines and the heart failure society of america," *Journal of the American College of Cardiology*, vol. 68, no. 13, pp. 1476–1488, 2016.

[10] E. G. Canty, Y. Lu, R. S. Meadows, M. K. Shaw, D. F. Holmes, and K. E. Kadler, "Coalignment of plasma membrane channels and protrusions (fibripositors) specifies the parallelism of tendon," *The Journal of cell biology*, vol. 165, no. 4, pp. 553–563, 2004.

[11] W. M. Petroll, L. Ma, and J. V. Jester, “Direct correlation of collagen matrix deformation with focal adhesion dynamics in living corneal fibroblasts,” *Journal of cell science*, vol. 116, no. 8, pp. 1481–1491, 2003.

[12] G. M. Fomovsky, A. D. Rouillard, and J. W. Holmes, “Regional mechanics determine collagen fiber structure in healing myocardial infarcts,” *Journal of molecular and cellular cardiology*, vol. 52, no. 5, pp. 1083–1090, 2012.

[13] L. R. Caggiano, J.-J. Lee, and J. W. Holmes, “Surgical reinforcement alters collagen alignment and turnover in healing myocardial infarcts,” *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 315, no. 4, H1041–H1050, 2018.

[14] I. González-Valverde and J. M. Garcıa-Aznar, “Mechanical modeling of collective cell migration: An agent-based and continuum material approach,” *Computer Methods in Applied Mechanics and Engineering*, vol. 337, pp. 246–262, 2018.

[15] J. A. Engelberg, G. E. Ropella, and C. A. Hunt, “Essential operating principles for tumor spheroid growth,” *BMC systems biology*, vol. 2, no. 1, p. 110, 2008.

[16] S. M. Peirce, E. J. Van Gieson, and T. C. Skalak, “Multicellular simulation predicts microvascular patterning and in silico tissue assembly,” *The FASEB journal*, vol. 18, no. 6, pp. 731–733, 2004.

[17] A. D. Rouillard and J. W. Holmes, “Mechanical regulation of fibroblast migration and collagen remodelling in healing myocardial infarcts,” *The Journal of physiology*, vol. 590, no. 18, pp. 4585–4602, 2012.

[18] W. J. Richardson and J. W. Holmes, “Emergence of collagen orientation heterogeneity in healing infarcts and an agent-based model,” *Biophysical journal*, vol. 110, no. 10, pp. 2266–2277, 2016.

[19] M. J. Plank and M. J. Simpson, “Models of collective cell behaviour with crowding effects: Comparing lattice-based and lattice-free approaches,” *Journal of the Royal Society Interface*, vol. 9, no. 76, pp. 2983–2996, 2012.

[20] J. Crank, *The mathematics of diffusion*. Oxford university press, 1979.

[21] G. M. Fomovsky and J. W. Holmes, “Evolution of scar structure, mechanics, and ventricular function after myocardial infarction in the rat,” *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 298, no. 1, H221–H228, 2010.

[22] K. L. Sack, N. H. Davies, J. M. Guccione, and T. Franz, “Personalised computational cardiology: Patient-specific modelling in cardiac mechanics and biomaterial injection therapies for myocardial infarction,” *Heart failure reviews*, vol. 21, no. 6, pp. 815–826, 2016.

[23] S. Thomopoulos, G. M. Fomovsky, and J. W. Holmes, “The development of structural and mechanical anisotropy in fibroblast populated collagen gels,” 2005.

[24] M. Bujak, M. Dobaczewski, K. Chatila, L. H. Mendoza, N. Li, A. Reddy, and N. G. Frangogiannis, “Interleukin-1 receptor type i signaling critically regulates infarct healing and cardiac remodeling,” *The American journal of pathology*, vol. 173, no. 1, pp. 57–67, 2008.

[25] D. A. Siwik, D. L.-F. Chang, and W. S. Colucci, “Interleukin-1β and tumor necrosis factor-α decrease collagen synthesis and increase matrix metalloproteinase activity in cardiac fibroblasts in vitro,” *Circulation research*, vol. 86, no. 12, pp. 1259–1265, 2000.

[26] A. M. Manso, S.-M. Kang, and R. S. Ross, “Integrins, focal adhesions, and cardiac fibroblasts,” *Journal of Investigative Medicine*, vol. 57, no. 8, pp. 856–860, 2009.

[27] B. A. French and J. W. Holmes, “Implications of scar structure and mechanics for post-infarction cardiac repair and regeneration,” *Experimental Cell Research*, vol. 376, no. 1, pp. 98–103, 2019.

[28] D. Park, E. Wershof, S. Boeing, A. Labernadie, R. P. Jenkins, S. George, X. Trepat, P. A. Bates, and E. Sahai, “Extracellular matrix anisotropy is determined by tfap2c-dependent regulation of cell collisions,” *Nature materials*, vol. 19, no. 2, pp. 227–238, 2020.

[29] J. F. Wenk, K. Sun, Z. Zhang, M. Soleimani, L. Ge, D. Saloner, A. W. Wallace, M. B. Ratcliffe, and J. M. Guccione, “Regional left ventricular myocardial contractility and stress in a finite element model of posterobasal myocardial infarction,” *Journal of biomechanical engineering*, vol. 133, no. 4, 2011.

[30] A. C. Estrada, K. Yoshida, S. A. Clarke, and J. W. Holmes, “Longitudinal reinforcement of acute myocardial infarcts improves function by transmurally redistributing stretch and stress,” *Journal of Biomechanical Engineering*, vol. 142, no. 2, 2020.

[31] R. C. Kerckhoffs, J. H. Omens, and A. D. McCulloch, “A single strain-based growth law predicts concentric and eccentric cardiac growth during pressure and volume overload,” *Mechanics research communications*, vol. 42, pp. 40–50, 2012.

[32] J.-J. Lee, L. Talman, S. M. Peirce, and J. W. Holmes, “Spatial scaling in multiscale models: Methods for coupling agent-based and finite-element models of wound healing,” *Biomechanics and modeling in mechanobiology*, vol. 18, no. 5, pp. 1297–1309, 2019.

[33] S. Guido and R. T. Tranquillo, “A methodology for the systematic and quantitative study of cell contact guidance in oriented collagen gels. correlation of fibroblast orientation and gel birefringence,” *Journal of cell science*, vol. 105, no. 2, pp. 317–331, 1993.

[34] J. Liu, X. Guo, X. Ren, H. Tian, Y. Liang, Z. Luo, W. Wang, Y. Wang, D. Zhang, Y. Huang, *et al.*, “A novel fpcl model producing directional contraction through induction of fibrob-

last alignment by biphasic pulse direct current electric field," *Experimental cell research*, vol. 371, no. 2, pp. 426–434, 2018.

[35] M. C. Genau, P. E. Perreault, E. Romito, H. Doviak, C. B. Logdon, S. Ruble, and F. G. Spinale, "Institution of localized high-frequency electrical stimulation targeting early myocardial infarction: Effects on left ventricle function and geometry," *The Journal of thoracic and cardiovascular surgery*, vol. 156, no. 2, pp. 568–575, 2018.

[36] A. Ramirez, J. A. Schwane, C. McFARLAND, and B. Starcher, "The effect of ultrasound on collagen synthesis and fibroblast proliferationin vitro," *Medicine & Science in Sports & Exercise*, vol. 29, no. 3, pp. 326–332, 1997.

[37] S.-C. Fu, W.-T. Shum, L.-K. Hung, M. W.-N. Wong, L. Qin, and K.-M. Chan, "Low-intensity pulsed ultrasound on tendon healing: A study of the effect of treatment duration and treatment initiation," *The American journal of sports medicine*, vol. 36, no. 9, pp. 1742–1749, 2008.