Introduction

The Moncrief Summer Internship allows students to work with faculty from UT Austin's Oden Institute of Computational Engineering and Sciences. I worked in the Willerson Center for Cardiovascular Modelling and Simulation (WCCMS), a research center developing computational biomechanical models to understand heart valves and heart disease progression; ultimately, these models are used to develop/improve clinical interventions (transcatheter aortic valve replacement procedures) and prosthetic devices (MitraClip repair, hydrogel-polyurethane fiber composites for valve replacement). With patient-specific valve function models, cardiologists/cardiovascular surgeons can optimize cardiovascular function for patients, thus decreasing risk-adverse side effects of invasive treatments.

Work Environment

My advisor is Dr. Michael Sacks, the director of WCCMS and a professor in the Biomedical Engineering department at UT Austin. I was mentored primarily by Gabriel Peery, a 2nd year CSEM (Computational Science, Engineering, and Mathematics) PhD student at UT Austin who works on modelling aortic valve interstitial cells (AVICs). Occasionally, if I/Gabriel wanted additional verification on some math or general concepts, we talked with Kenneth Meyer and Benjamin Thomas, 3rd year CSEM PhD students using a Neural Network Finite Element approach to model mitral valves, specifically, leaflet contact.

I participated in weekly meetings between Dr. Sacks, Dr. Toni West, a biologist who works on activating and imaging valve interstitial cells, and Gabriel.

Technical Summary

Aortic valve interstitial cells (AVICs) play key roles in valvular heart disease. When mechanically stimulated, AVICs secrete enzymes and collagen that degrade or stiffen their local microenvironments. To isolate and study AVICs (activated vs. basal state), a tissue-mimicking synthetic polyethylene glycol (PEG) hydrogel (300 μm cube) is impregnated with a stained AVIC. With 3D Traction Force Microscopy and Gaussian Process Regression, AVIC displacement upon activation is inferred.

We (WCCMS) simulated these microenvironment modifications to determine AVIC metabolism and contractability, indicators of phenotypic state. However, current AVIC models are slow, especially for highly detailed problems with finer meshes and realistic cell contraction models. We evaluate JAX-FEM, a GPU-accelerated differentiable finite element (FE) package against the standard FEniCS finite element toolkit, across different AVIC problem sizes to seek improvements in runtime.

We develop a forward model that takes a gel modulus field as input and solves for gel nodal displacements. Then, with an inverse model that optimizes an objective function iteratively (via the forward model), the modulus field with minimal error in simulated nodal displacements can be identified.

For the forward model, the gel's material stress and forces are determined with an isochoric, hyperelastic neo-Hookean model:

$$\Psi = \alpha(\mathbf{x_0})C_1(I_1 - 3) \; - \; 2C_1 \ln J \; + \; D_1(\ln J)^2$$

$$\mathbf{F} = \frac{d\mathbf{x}}{d\mathbf{x_0}} \qquad \mathbf{C} = \mathbf{F}^T\mathbf{F} \qquad J = \det(\mathbf{F}) \qquad I_1 = \mathrm{tr}(\mathbf{C})$$

$$C_1 = 50Pa, \quad \frac{D_1}{C_1} = 10,000$$

where ψ is the gel strain energy density, α is a scalar field that accounts for local stiffness, and $\boldsymbol{x_0}$ is the reference gel position. $\alpha(\boldsymbol{x_0})C_1(I_1 - 3)$ is the effective material

model, while the other strain energy terms are penalties to enforce near incompressibility. $D_1\,(\ln J)^2$ is quadratic to penalize both material compression and expansion. $2C_1 \ln J$ is added to prevent volume locking.

To account for nonlinear cell deformations, we implement load steps: displacement is incrementally applied with each intermediate solution acting as the next original configuration.

The JAX-FEM displacement solution, $\boldsymbol{u}$ (μm), is similar to the FEniCS solution (Figure 1):

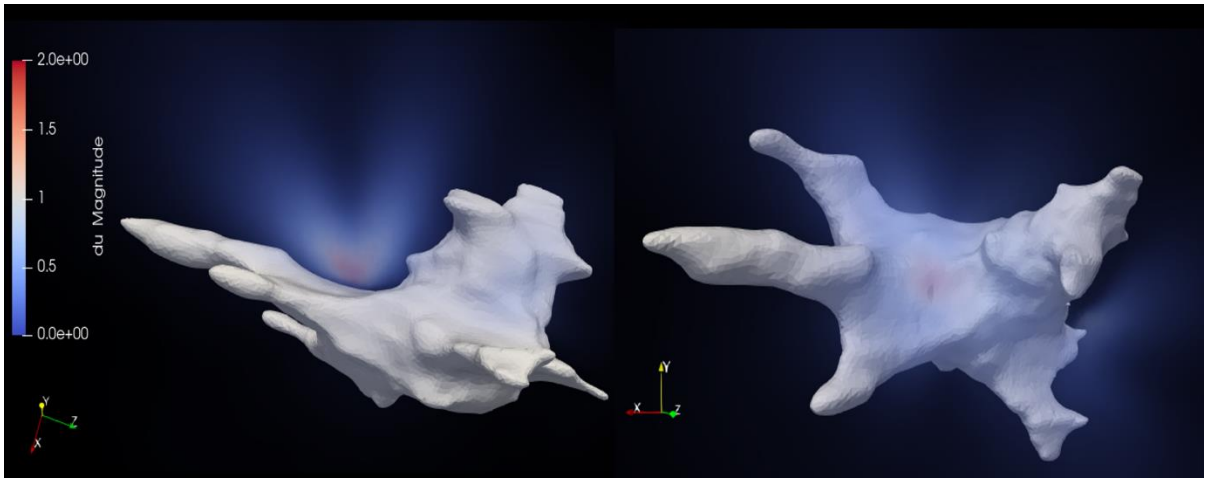

Figure 1: Solution difference between JAX-FEM and FEniCS. Reference configuration.

Note that JAX-FEM has a higher tolerance though: JAX-FEM's residual is 10^{-3} while FEniCS is 10^{-9}. This is likely due to preconditioner difference: Incomplete LU factorization (ILU) vs. Algebraic Multigrid (AMG). FEniCS fails to solve the problem with an ILU preconditioner while AMG has slower convergence than ILU for JAX-FEM. Meanwhile, the Jacobian of the deformation gradient is more accurate (closer to 1, per incompressibility) in JAX-FEM than in FEniCS (Figure 2).

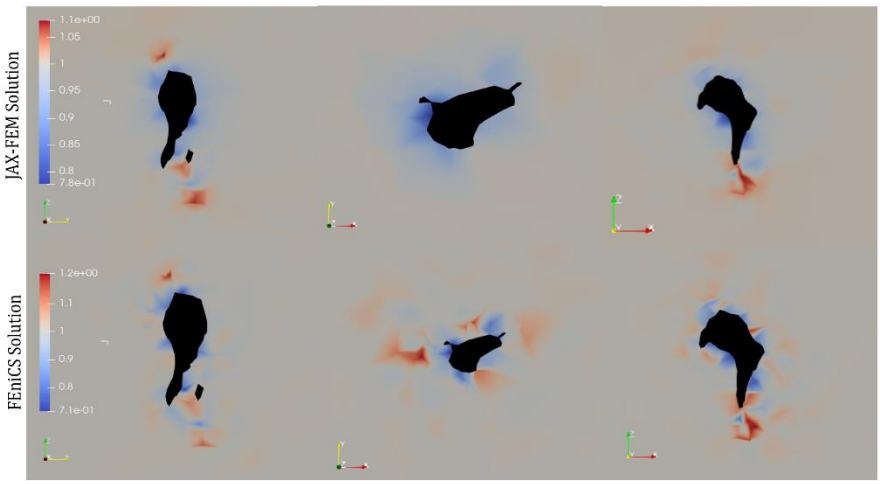

Figure 2: Jacobian field cross-sections.

Due to the lack of convergence in JAX-FEM, however, JAX-FEM's solution is not ruled out as more accurate than FEniCS.

When run on GPU, JAX-FEM scales better than FEniCS (Figure 3), especially for finer mesh sizes.

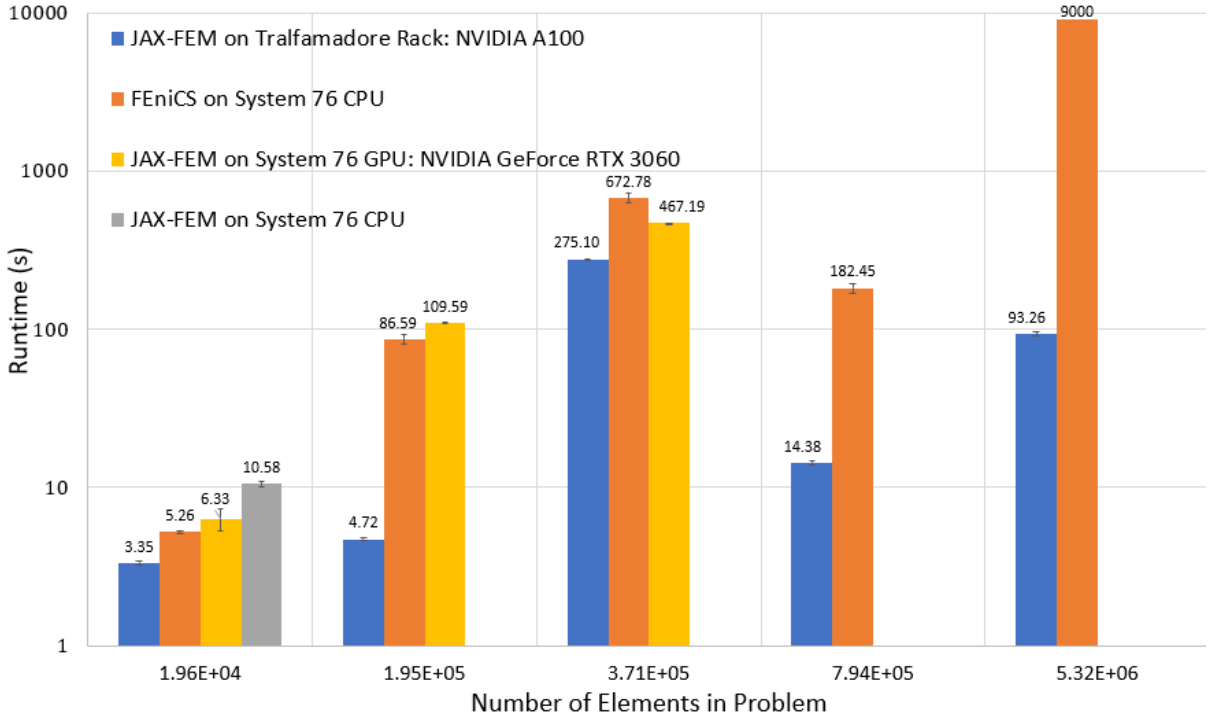

Figure 3: Runtime comparison between JAX-FEM and FEniCS (logarithmic scale).

From $7.93 * 10^5$ to $5.32 * 10^6$ elements, FEniCS takes 50x longer while JAX-FEM is only 6.6x in runtime.

The inverse model objective function measures solution accuracy:

$$\phi = \int_\Omega \xi : \xi \, d\Omega + \gamma \int_\Omega (\nabla\alpha \cdot \nabla\alpha)^{\frac{1}{2}} \, d\Omega \qquad \xi = \mathbf{C}_{target} - \mathbf{C}_{simulated} \qquad \Omega = \text{reference gel domain}$$

$$\mathbf{C} := \mathbf{F}^T\mathbf{F} \text{ (the Cauchy-Green tensor)} \qquad \gamma = \text{regularization}$$

$$\hat{\alpha} = \min_{\alpha(\text{DoFs})} \phi \qquad \text{Ideally, Tikhonov regularization would be used instead:} \quad \int_\Omega (\nabla\alpha \cdot \nabla\alpha) \, d\Omega$$

JAX-FEM natively supports a piece-wise constant α field, while FEniCS supports a piece-wise linear field. We attempt to use Tikhonov regularization on a test case (Figure 4), two adjacent tetrahedral elements (denoted by 1 and 0 subscripts):

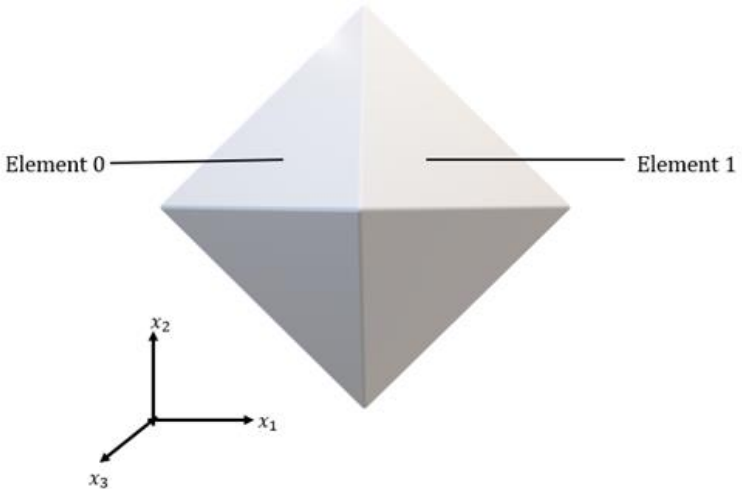

Figure 4: Tikhonov regularization test case.

Given the following properties of the Dirac delta,

$$f(x) = \begin{Bmatrix} y_1, & x < a \\ y_2, & x > a \end{Bmatrix}$$

$$f'(x) = (y_2 - y_1)\,\delta(x - a)$$

$$\int_{R^3} f(x)\,\delta(x)dx = f(0)$$

the Tikhonov regularization term can be defined as:

$$\nabla\alpha = \begin{bmatrix} \frac{\partial\alpha}{\partial x_1} = \delta(x_1)(\alpha_1 - \alpha_0) \\ \frac{\partial\alpha}{\partial x_2} = 0 \\ \frac{\partial\alpha}{\partial x_3} = 0 \end{bmatrix}$$

$$\Rightarrow \nabla\alpha \cdot \nabla\alpha = \int_0^1 \int_0^1 \int_{-\infty}^{\infty} (\alpha_1 - \alpha_0)\,\delta(x_1)^2 \, dx_1\, dx_2\, dx_3 = (\alpha_1 - \alpha_0) \int_0^1 \int_0^1 \int_{-\infty}^{\infty} \delta(x_1)^2 \, dx_1\, dx_2\, dx_3$$

assuming the tetrahedrons span from 0 to 1 in the x_2 and x_3 directions. Since there are no other elements (with different α values) in the x_2 and x_3 directions, $\frac{\partial\alpha}{\partial x_2}$ and $\frac{\partial\alpha}{\partial x_3}$ are 0.

The regularization term is further simplified:

$$\int_{R^3} f(x)\,\delta(x)dx = f(0) \Rightarrow \int_{-\infty}^{\infty} \delta(x_1)^2\,dx_1 = \int_{-\infty}^{\infty} \delta(x_1)\,\delta(x_1)\,dx_1 = \delta(0)$$

Thus, since piece-wise constant α fields cause Tikhonov regularizers to diverge, we use total variance.

To update the modulus field, we use SciPy's optimize-minimize function with the L-BFGS-B (Limited-Memory Broyden-Fletcher-Goldfarb-Shanno algorithm) algorithm. While BFGS stores a dense n x n approximation of the inverse Hessian (2nd derivative) matrix, L-BFGS-B stores only a few vectors that represent the approximation implicitly, making it well suited for inverse models with many degrees of freedom.

To test the inverse model, we test the model's ability to recover a prescribed modulus field (taken as the true stiffness field). The process is detailed in Figure 5:

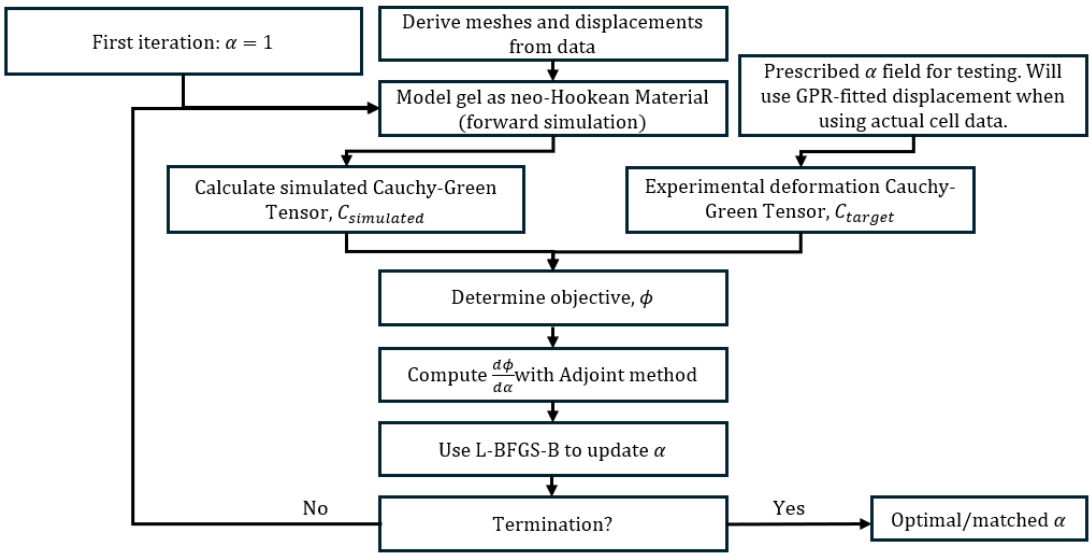

Figure 5: Inverse model pipeline

The objective gradient is determined with an adjoint method, allowing for more accuracy and efficiency than with a finite difference gradient approximation. The strain energy density, Ψ, provides a PDE relating the solution (u, the displacement field) and

the optimization parameter (α, the stiffness field), such that $\Psi(u, \alpha) = 0$. Setting derivatives to zero (first-order optimality), we have:

$$\frac{d\phi}{d\alpha} = \frac{\partial \phi}{\partial u}\frac{du}{d\alpha} + \frac{\partial \phi}{\partial \alpha}$$

$$\frac{d\Psi}{d\alpha} = \frac{\partial \Psi}{\partial u}\frac{du}{d\alpha} + \frac{\partial \Psi}{\partial \alpha} = 0$$
$$\implies \frac{du}{d\alpha} = -[\frac{\partial \Psi}{\partial u}]^{-1}\frac{\partial \Psi}{\partial \alpha}$$

A visual representation of objective gradient is shown in Figure 6:

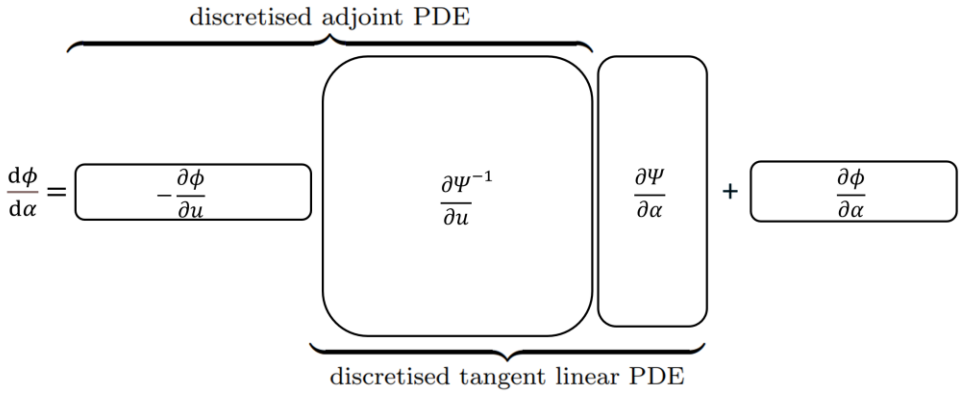

Figure 6: Adjoint method

The adjoint PDE uses vector-matrix computations, while the tangent linear PDE requires expensive matrix-matrix computations; we compute the adjoint PDE first for efficiency.

The recovered modulus field difference is shown below (Figure 7):

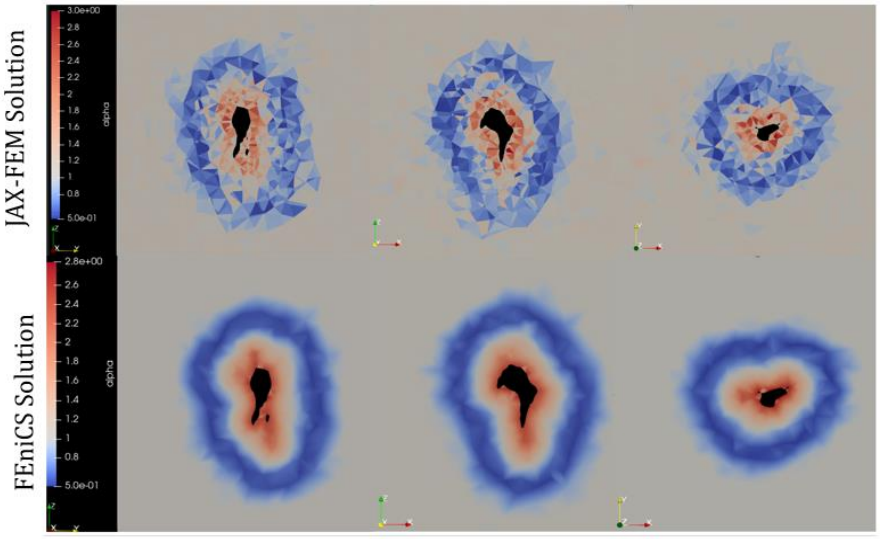

Figure 7: Inverse model recovered modulus fields.

Since the inverse model has not been fine tuned (use a Tikhonov regularize r, adjust γ value, etc.), timing data has not been acquired.

Conclusion

Future steps for the JAX-FEM model are to:

- Implement a stronger preconditioner to decrease residual below 10^{-9}.
- Implement a piecewise linear α field with Tikhonov regularization for the inverse model.
- Run the inverse model on a real cell to recover the α field.
- Develop advanced AVIC stress fiber models.

I plan to continue work in Dr. Sack's lab either in improving the JAX-FEM model or in simulating the mitral valve using a physics-based neural network model.