The University of Texas at Austin
Biomedical Engineering
Cockrell School of Engineering
ODEN INSTITUTE
FOR COMPUTATIONAL ENGINEERING & SCIENCES

Modeling Aortic Valve Interstitial Cells with JAX-FEM

[1]James T. Willerson Center for Cardiovascular Modeling & Simulation
Oden Institute for Computational Engineering & Sciences
Department of Biomedical Engineering, UT Austin

Arushi R. Sadam, Gabriel Peery[1], and Michael S. Sacks[1]

FEniCS PROJECT
WILLERSON CENTER FOR CARDIOVASCULAR MODELING & SIMULATION

1. Motivation

- 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. We (WCCMS) simulated these microenvironment modifications to determine AVIC metabolism and contractability, indicators of phenotypic state.
- 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.

2. Data

To isolate and study VICs (activated vs. basal state), a tissue-mimicking synthetic polyethylene glycol (PEG) hydrogel is impregnated with stained AVICs and green fluorescent microbeads.

- AVIC contraction causes the microbeads to shift.
- 3D Traction Force Microscopy then tracks microbead displacement.
- Using Gaussian process regression (GPR) on bead displacement data, AVIC displacement is inferred.[1]

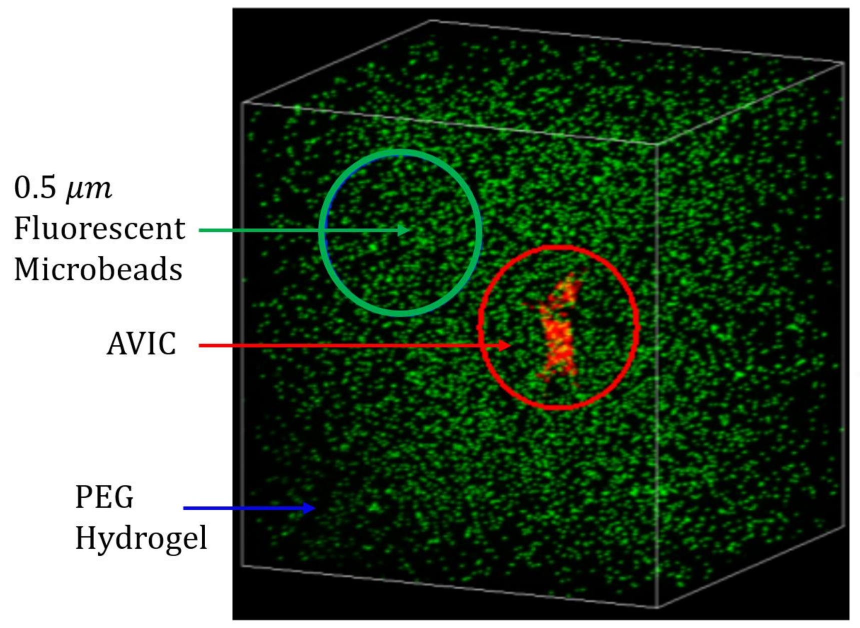

Figure 1: PEG Hydrogel with AVIC and Microbeads [2]

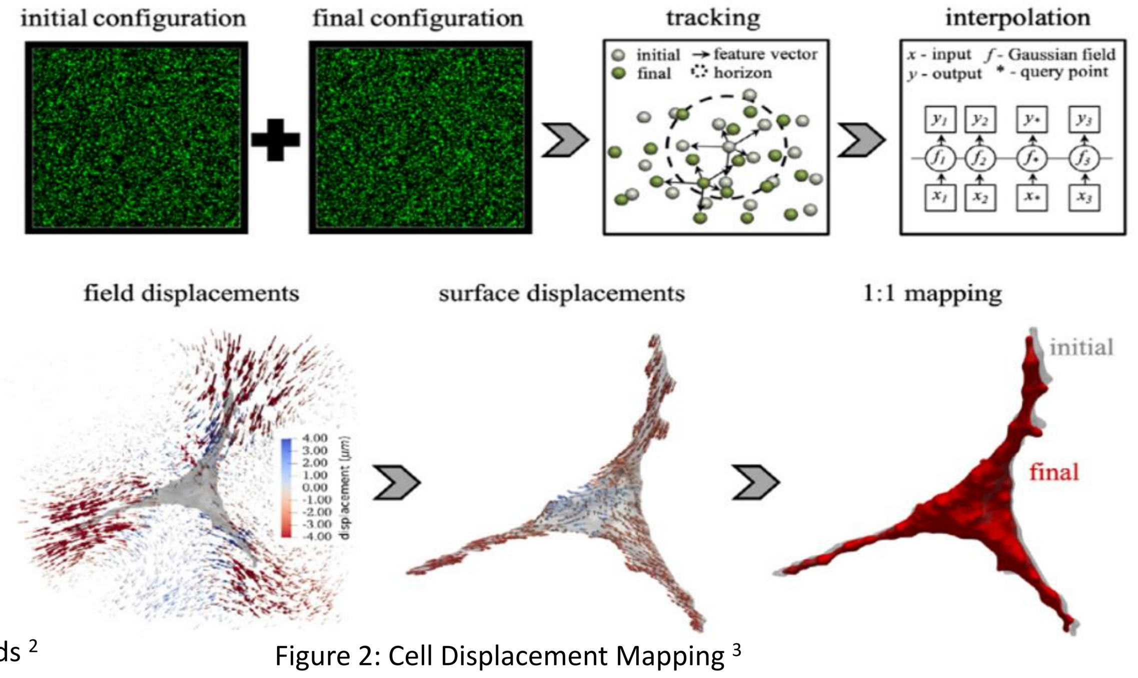

Figure 2: Cell Displacement Mapping [3]

3. The Forward Model

The forward model takes a gel modulus field as input and solves for nodal displacements.
The gel's material stress and forces are determined with an isochoric, hyperelastic neo-Hookean model:

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

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

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

- α is a scalar field that accounts for local stiffness.
- x_0 is the reference gel position.
- ψ is the gel strain energy density
- $\alpha(x_0)C_1(I_1 - 3)$ is the effective material model.
- The other strain energy terms are penalties to enforce near incompressibility.

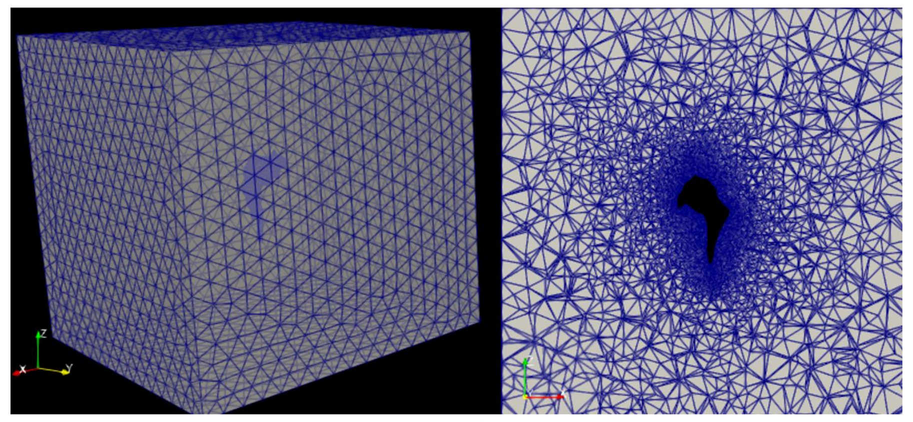

Figure 3: Mesh for $7.93 * 10^5$ Elements (129,758 Nodes).

Solution [JAX-FEM vs FEniCS]

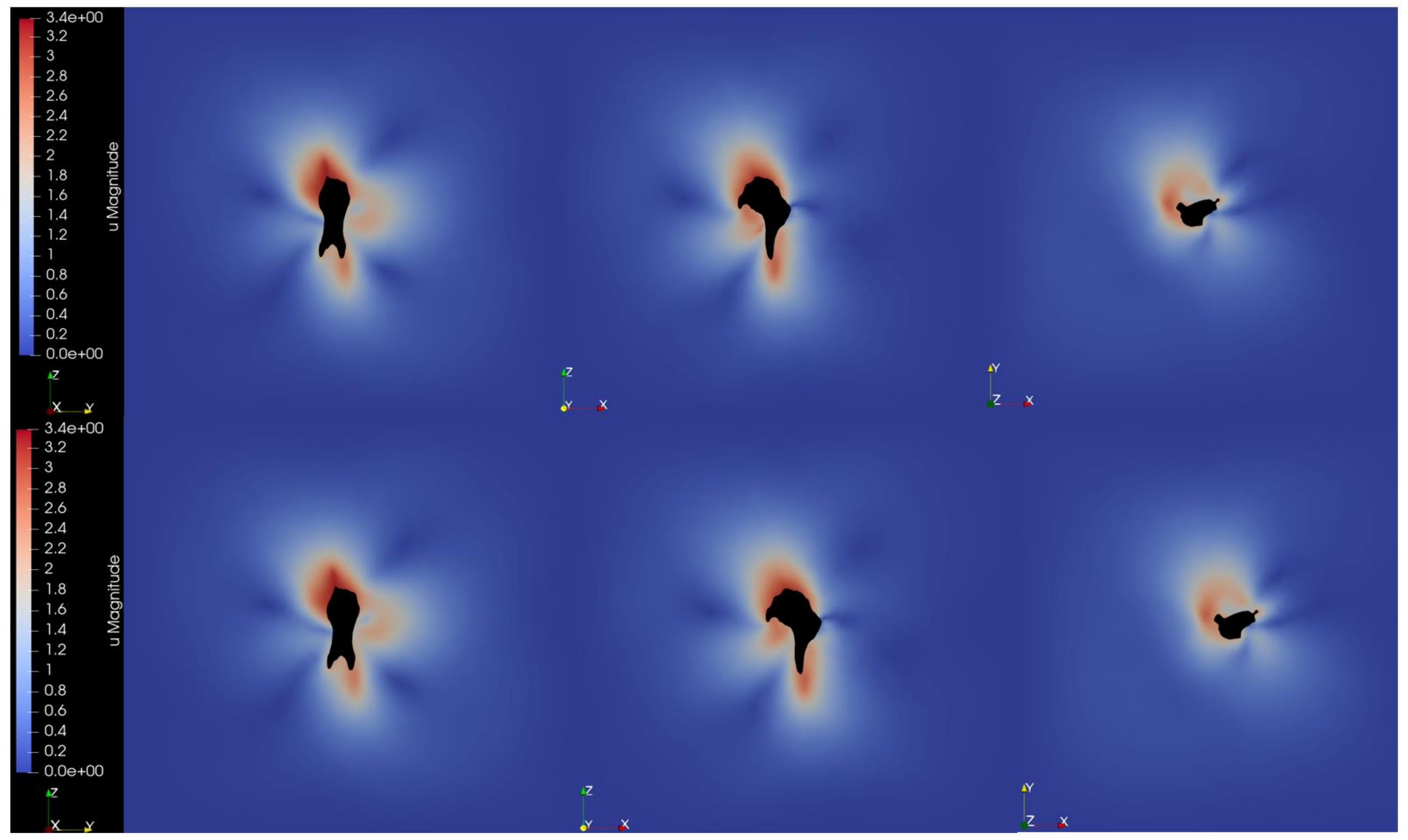

Figure 5: JAX-FEM vs. FEniCS solution cross-sections in current configuration.

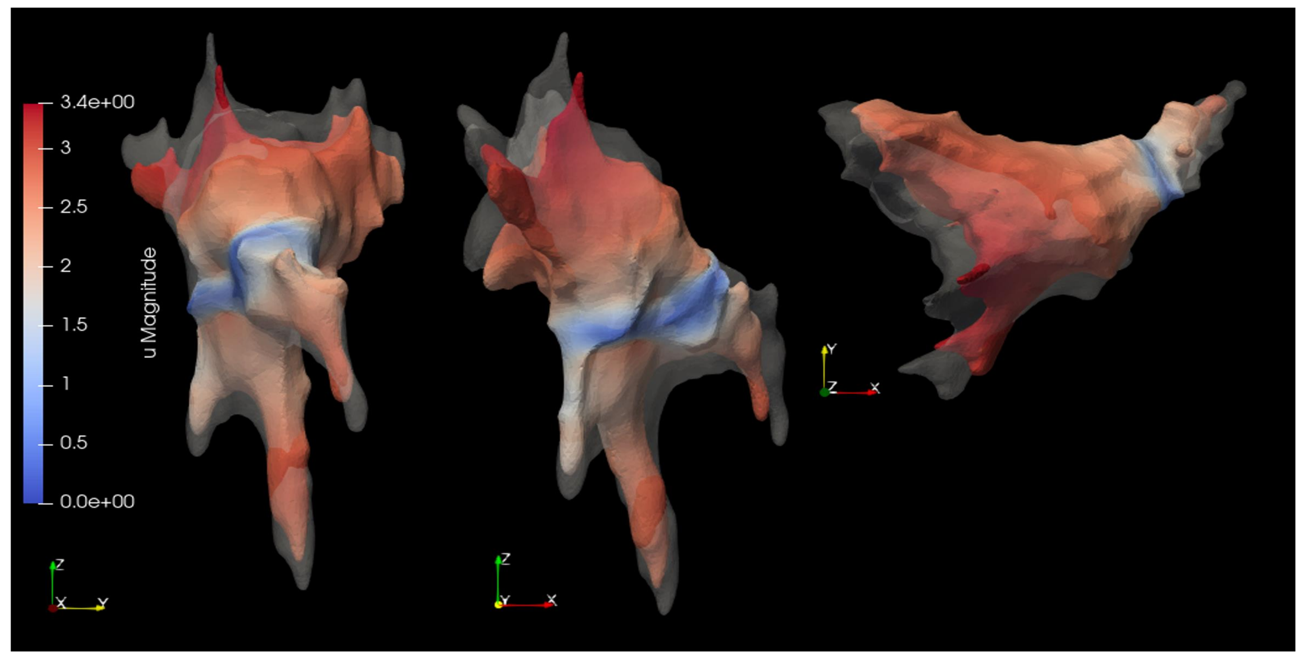

Figure 4: JAX-FEM forward model solution. Gray denotes reference configuration.

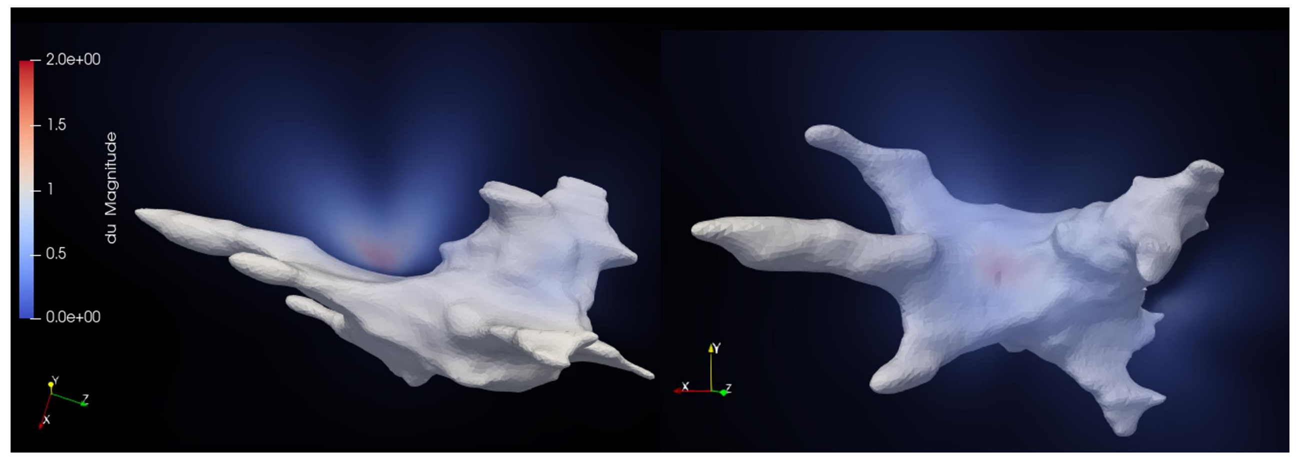

Figure 6: Solution difference between JAX-FEM and FEniCS in reference configuration.

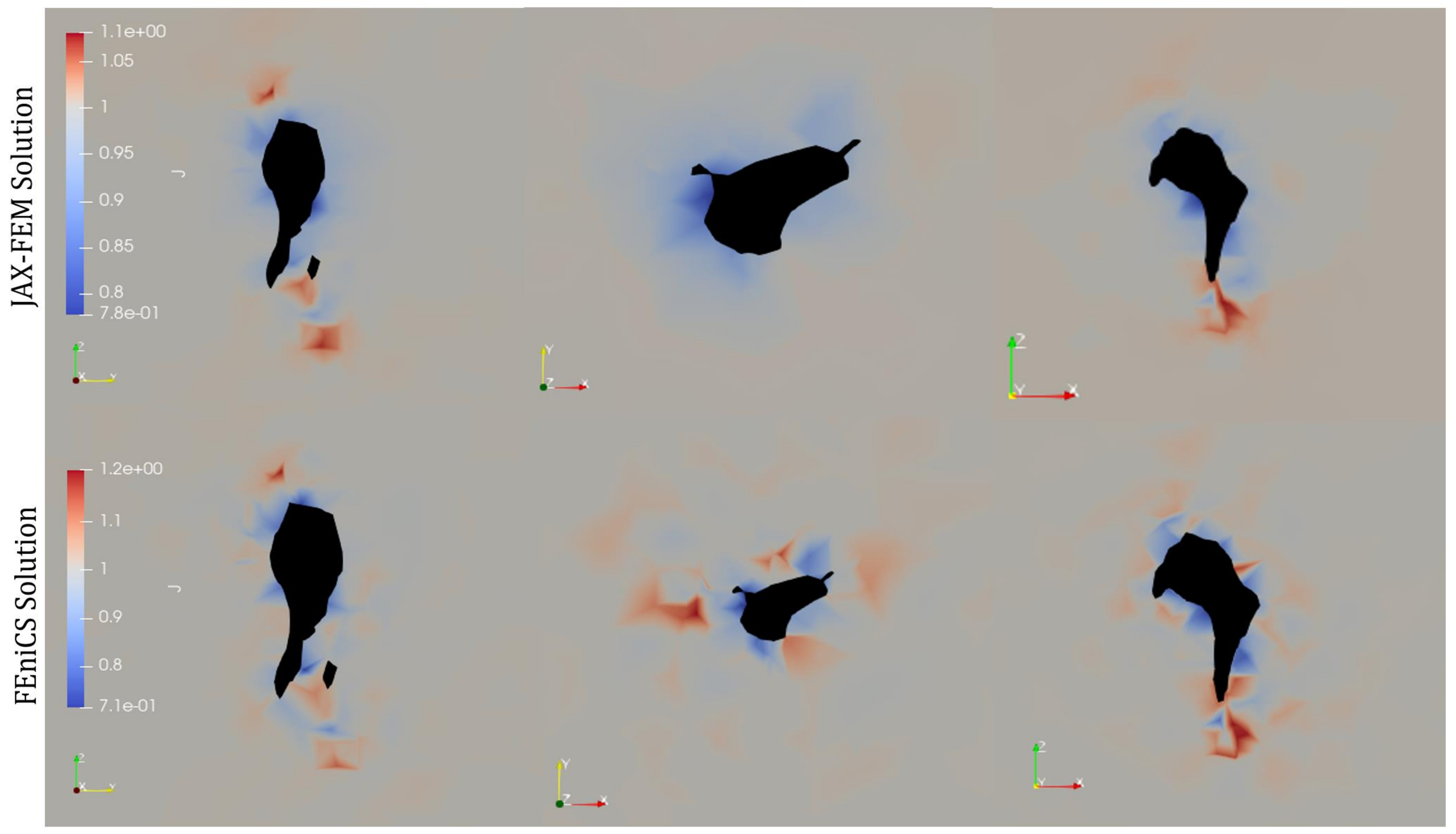

Figure 7: Jacobian field cross-sections of JAX-FEM and FEniCS solutions (should be 1 per incompressibility)

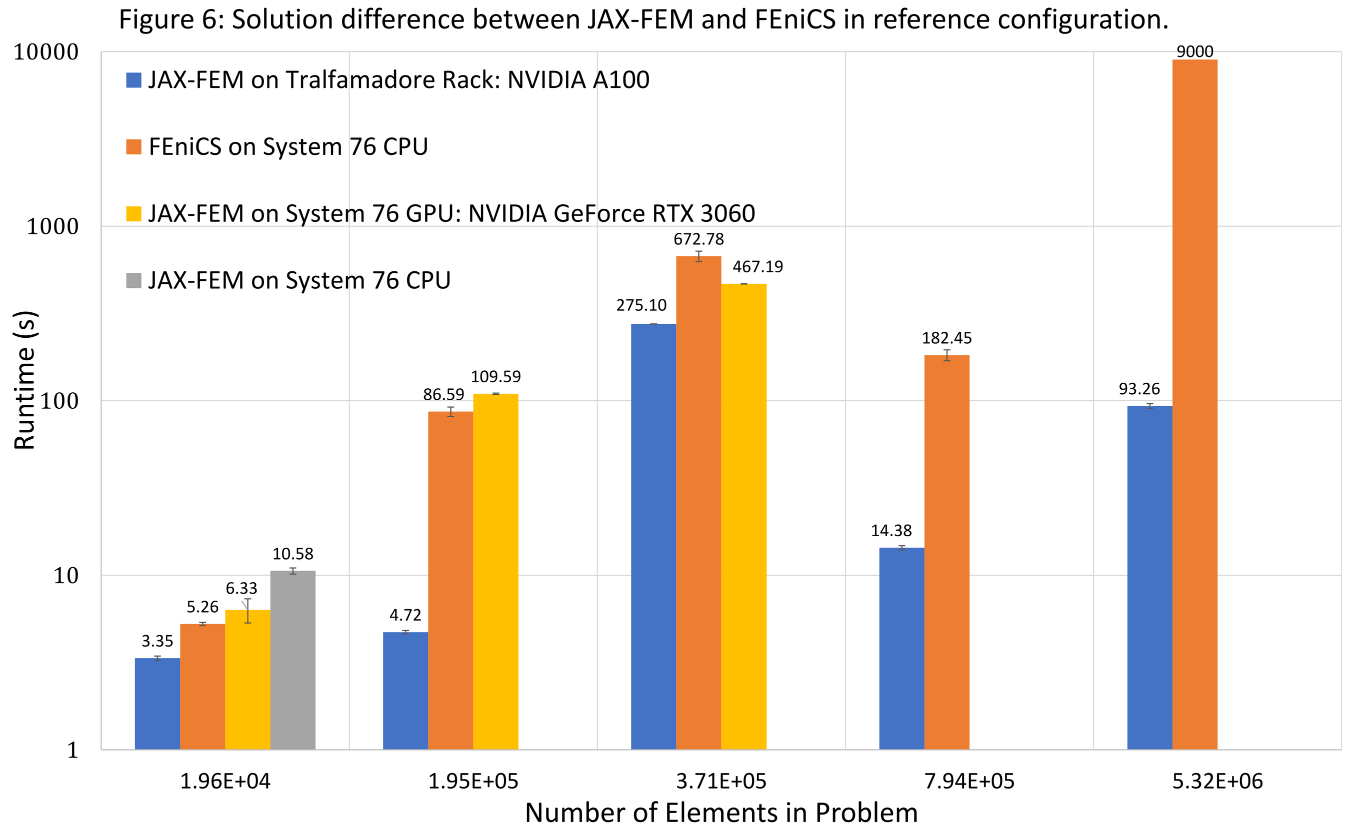

Figure 8: Runtime comparison between JAX-FEM and FEniCS[4]. Note logarithmic runtime scale.

Results

- JAX-FEM's Jacobian field (of the deformation gradient) **is more accurate** than FEniCS.
- JAX-FEM **scales better** than FEniCS: from $7.93 * 10^5$ to $5.32 * 10^6$ elements, FEniCS takes **50x** longer while JAX-FEM is only **6.6x** in runtime.
- JAX-FEM has **higher tolerance** though: JAX-FEM's residual is 10^{-3} while FEniCS is 10^{-9}.
  - 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.

4. The Inverse Model

The inverse model identifies the modulus field with minimal error in simulated nodal displacements, with objective[5,6] as below.

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

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

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

The objective gradient is determined with an adjoint method, allowing for more accuracy and efficiency than with a finite difference gradient approximation.

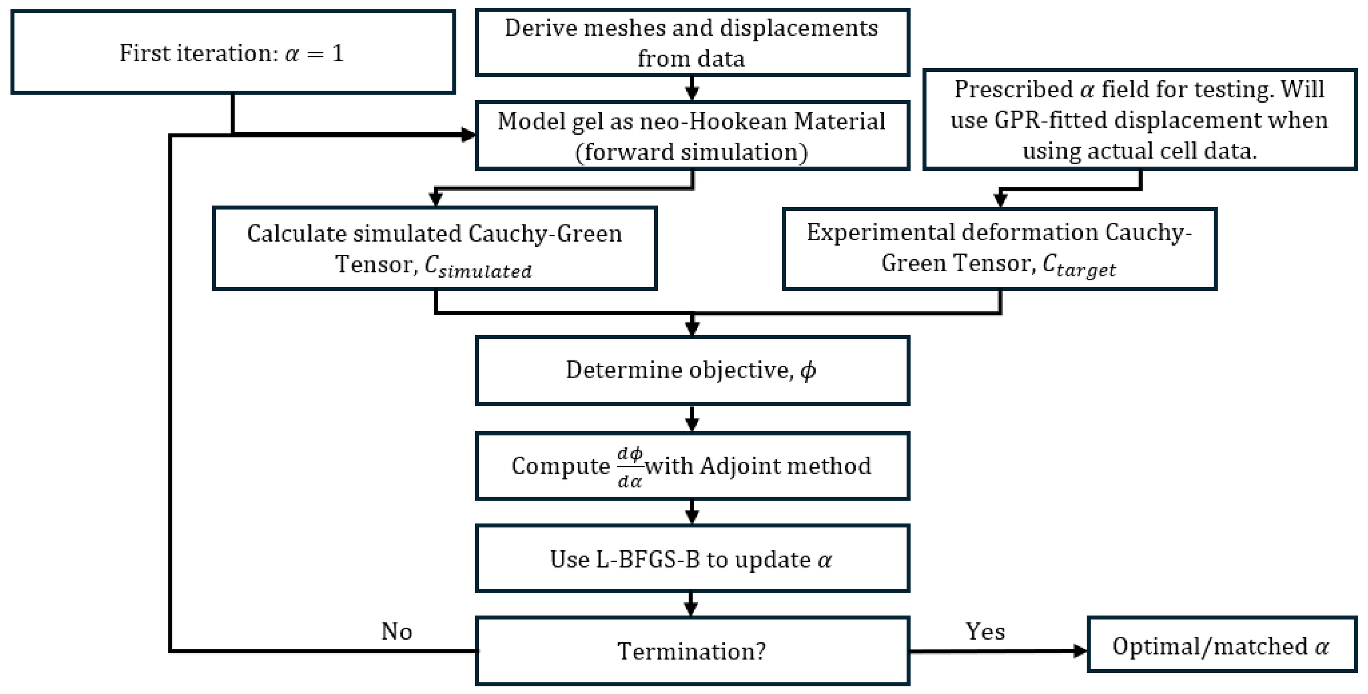

Figure 10: Inverse model pipeline

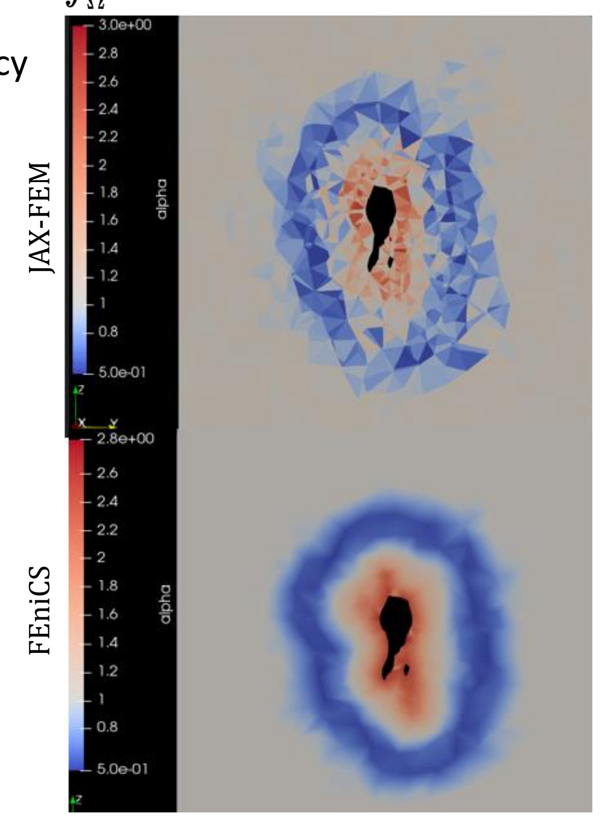

Figure 11: Inverse model recovered α fields[6]

5. The Adjoint Method

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:

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

$$\frac{d\Psi}{d\alpha} = \frac{\partial\Psi}{\partial u}\frac{du}{d\alpha} + \frac{\partial\Psi}{\partial\alpha} = 0$$

$$\Longrightarrow \frac{du}{d\alpha} = -\left[\frac{\partial\Psi}{\partial u}\right]^{-1}\frac{\partial\Psi}{\partial\alpha}$$

The adjoint PDE uses vector-matrix computations, while the tangent linear PDE requires expensive matrix-matrix computations.

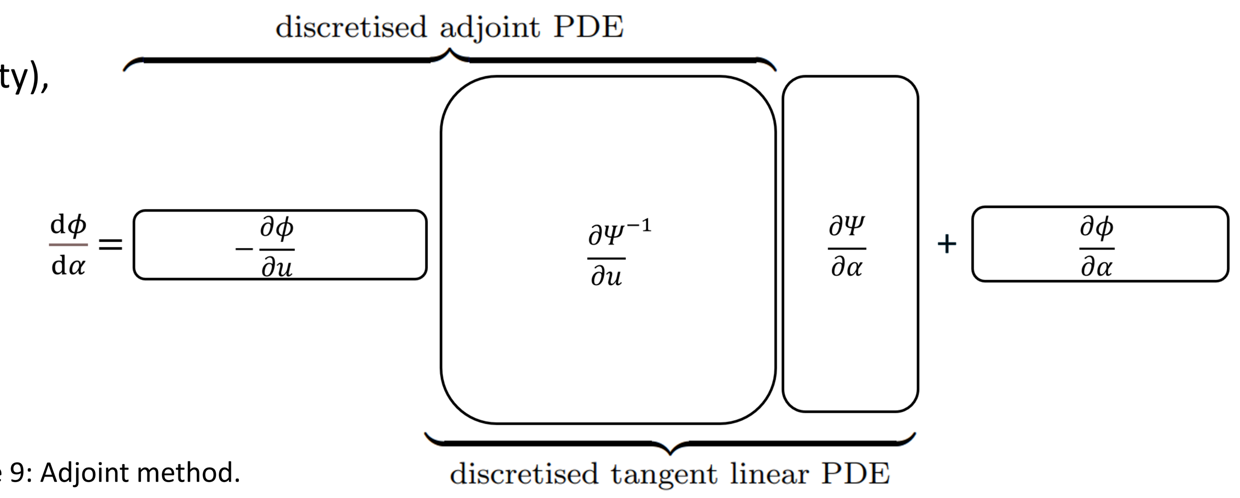

Figure 9: Adjoint method.

6. Conclusion

- JAX-FEM is faster than FEniCS on large problems (over 371K elements or 180K degrees of freedom).
- JAX-FEM's solution better satisfies incompressibility so may be more accurate.

7. Future Steps in JAX FEM

- 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.

[1] GPR infers a distribution over the function of interest directly, rather than optimizing function parameters.
[2] Image courtesy of Dr. Toni West
[3] Khang, Alex, "Estimation of aortic valve interstitial cell-induced 3D remodeling of poly(ethylene glycol) hydrogel environments using an inverse finite element approach".
[4] FEniCS took over 9000 seconds to solve the 5 million element problem; the program was stopped after 2.5 hours. Also, the problem with 371k elements required 7 load steps (while all other problems were solved with 1 load step).
[5] SciPy's optimize-minimize function is used with the L-BFGS-B (Limited-Memory Broyden-Fletcher-Goldfarb-Shanno algorithm) optimization 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 (DoFs).
[6] JAX-FEM natively supports a piece-wise constant α field, while FEniCS supports a piece-wise linear field. Piece-wise constant α fields cause Tikhonov regularizers to diverge, so we use total variance.