











pyMechT: A Python package for mechanics of soft tissues

Ankush Aggarwal ^{1,2}, Ross Williams ^{1,2}, Claire Rosnel ^{1,2}, Silvia Renon ^{1,2}, Jude M. Hussain^{1,2}, André F. Schmidt^{1,2}, Shiting Huang ^{1,2}, Sean McGinty ^{1,2}, and Andrew McBride ^{1,2}

1 Glasgow Computational Engineering Centre (GCEC), University of Glasgow, G12 8LT, United Kingdom
2 James Watt School of Engineering, University of Glasgow, G12 8LT, United Kingdom

DOI: [10.21105/joss.07490](https://doi.org/10.21105/joss.07490)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Kevin M. Moerman](#) 

Reviewers:

- [@siboles](#)
- [@jpsferreira](#)
- [@mpeirlinck](#)

Submitted: 27 September 2024

Published: 12 February 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

pyMechT fills an important gap for simulating simplified models in soft tissue mechanics, such as for ex-vivo testing protocols. Instead of detailed finite element simulations, which can be time-consuming and excessive for certain scenarios, pyMechT allows one to configure and run simulations extremely quickly. Moreover, the Python package makes it straightforward to perform parameter estimation and Bayesian inference. Its unique capabilities include incorporating layered structure of tissues and residual stresses.

Statement of need

Mechanics of soft tissues plays an important role in several physiological problems, including cardio-vascular and musculoskeletal systems. Common ex-vivo biomechanical testing protocols used to characterize tissues include uniaxial extension for one-dimensional structures, such as tendons and ligaments, biaxial extension for planar tissues, such as heart valves and skin, and inflation-extension for tubular tissue structures, such as blood vessels (Figure 1). These experiments aim to induce a uniform deformation that can be easily related to the generated stresses.

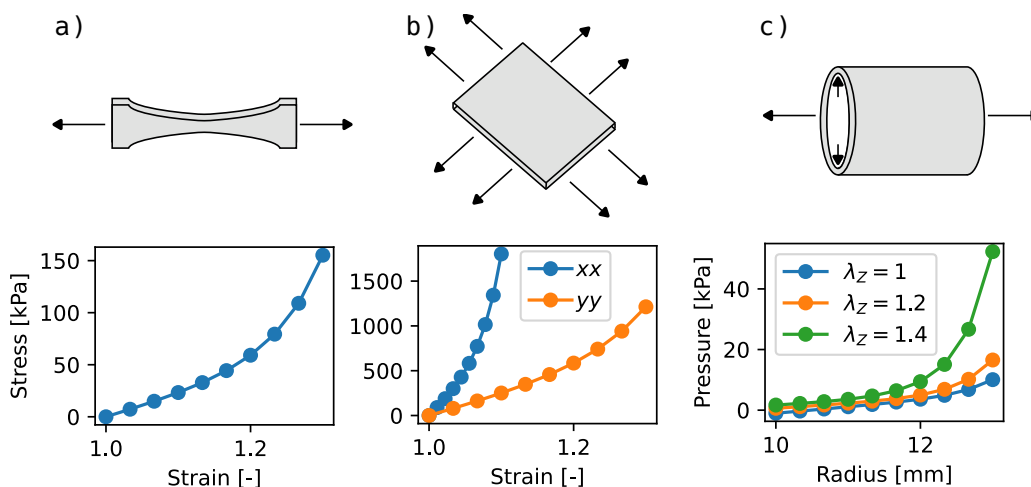


Figure 1: Three common ex-vivo experimental protocols and corresponding load-deformation plots: a) uniaxial extension, b) planar biaxial extension, and c) extension-inflation at different longitudinal stretch λ_z

While several finite element analysis packages are available for performing biomechanical simulation, they are generally intended for more complex scenarios involving non-uniform/non-affine deformations. For simulating the ex-vivo experiments, which induce close-to-uniform deformations, in-house codes are commonly developed. However, the absence of a common framework can lead to lack of consistency and reproducibility. Moreover, advanced analyses require statistical approaches, such as Monte Carlo simulations and Bayesian inference. To fill this gap, we have developed the open-source Python package pyMechT.

Structure

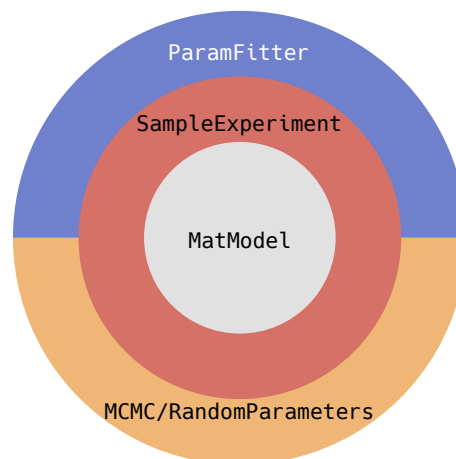


Figure 2: Structure of pyMechT

The package is implemented in Python using an object-oriented structure. The package builds upon widely-used Python libraries: NumPy, SciPy, Pandas, Matplotlib, and PyTorch. pyMechT consists of four main modules (see Figure 2): 1) MatModel for defining constitutive models for materials, 2) SampleExperiment for simulating ex-vivo uniaxial/biaxial/inflation-extension experiments, 3) ParamFitter for performing parameter estimation based on experimental data, and 4) MCMC/RandomParameters for performing Bayesian inference using Monte Carlo (MC) or Markov Chain Monte Carlo (MCMC) simulations. Currently, there are eighteen material models implemented in MatModel, including fourteen analytical hyperelastic models, two data-based hyperelastic models, and one structural model. In addition, an arbitrary hyperelastic model is also implemented, where a user-defined form of the free energy functional is automatically implemented based on symbolic differentiation. Below is the list of the material models available to-date:

- 'NH': Neo-Hookean model
- 'MR': Mooney-Rivlin model
- 'YEOH': Yeoh model
- 'LS': Lee-Sacks model
- 'MN': May-Newman model
- 'GOH': Gasser-Ogden-Holzapfel model
- 'HGO': Holzapfel-Gasser-Ogden model
- 'expl1': A model with an exponential of I1
- 'polyI4': A model with a polynomial of I4
- 'HY': Humphrey-Yin model
- 'Holzapfel': Holzapfel model
- 'volPenalty': A penalty model for volumetric change
- 'ArrudaBoyce': Arruda-Boyce model

- 'Gent': Gent model
- 'spline11': A spline model of I1
- 'spline114': A spline model of I1 and I4
- 'StructModel': A structural model with fiber distribution
- 'ARB': Arbitrary model with user-defined strain energy density function

A particular focus is on parameters, for which a custom dictionary has been implemented named `ParamDict`. This dictionary facilitates handling large numbers of parameters via string-based identifiers (“Keys”), and stores lower/upper bounds, fixed/variable flags, in addition to the current parameter values. The dictionary can also be saved/read as csv files. An example set of parameters is shown in Table 1 below.

Table 1: Example set of parameters saved as `ParamDict` object where “Key” acts as string-based identifier

| Keys | Value | Fixed? | Lower bound | Upper bound |
|-------|-------|--------|-------------|-------------|
| mu_0 | 100 | No | 0.01 | 500 |
| L10 | 0.3 | No | 0.1 | 0.5 |
| L20 | 1.0 | No | 0.1 | 2.0 |
| thick | 0.05 | Yes | - | - |
| phi | 50 | No | 0 | 90 |

Documentation and examples

Detailed documentation is hosted on [readthedocs](#). The documentation starts with an overview of the package, and leads to a basic tutorial that helps one getting started and briefly demonstrates all of the essential features. Additionally, eleven examples have been provided to illustrate all the features and options available in `pyMechT`. These include the unique features of modeling layered structures with different reference dimensions, which is commonly encountered in biological soft tissues. Simulating such a model with any finite element software would be non-trivial. Then, the theoretical background of the implemented models is provided, before concluding with a package reference automatically generated using `Sphinx`.

Advantages over finite element simulation

In principle, the problems that can be solved using `pyMechT` can also be solved using any finite element simulation software. However, `pyMechT` offers the following advantages:

- Geometry and mesh creation would be required for a finite element simulation, which usually takes some time. However, the pre-defined geometrical features in `pyMechT` means that one only needs to choose the right class and parameters. In addition, no meshing is required. This means that setting up the problem (i.e., defining the geometry/mesh and loads) is much faster in `pyMechT`. Once the model has been setup, the computational time required to solve it is comparable, depending on the finite element mesh density.
- Enforcing incompressibility in a finite element simulation can be numerically challenging, necessitating approaches such as Lagrange multiplier with a three-field formulation. Instead, in `pyMechT`, the incompressibility is analytically enforced *exactly*, thus making the results more robust.
- The fast nature of simulations in `pyMechT` makes it feasible to run $\mathcal{O}(10^5)$ simulations in several minutes, thus facilitating Monte Carlo and Bayesian inference. This adds the capability of calculating, not only the mean response, but also the confidence intervals of model fits and predictions.

- The reference zero-stress state of biological tissues can be unknown or ambiguous. Moreover, the biological tissues are heterogeneous, with multiple layers each of varying properties. These aspects are non-trivial to incorporate in a finite element simulation, due to the need for recreating the geometry and/or incompatibility of the initial state. However, it is straightforward to simulate these in pyMechT.

Overall, there are many other tools that can perform constitutive model fitting. Commercial finite element software [Abaqus](#) and [Ansys](#) have in-built constitutive model fitting tools, such as, [PolyUMod](#) and [MCalibration](#). [Hyperfit](#) is a commercial software specifically for constitutive model fitting, with the advantage of having a graphical user interface. However, these are commercial and are not free/open-source. There are alternative open-source tools for constitutive model fitting, such as [matmodelfit](#) and [hyperelastic](#). However, these are not specifically focused on tissues and lack the capability of simulating layered samples or inflation-extension experiment on tubular structures, common in tissue mechanics. Lastly, most of the existing tools do not incorporate Bayesian inference, which is important for providing a confidence interval on fitted parameters and model predictions.

Uses in literature

pyMechT has been used for Bayesian model selection based on extensive planar biaxial extension data ([Aggarwal, Hudson, et al., 2023](#)). This work required rapid simulation of varied constitutive models, which was facilitated by pyMechT. Similarly, the Bayesian inference via Markov Chain Monte Carlo in pyMechT was used to infer the distribution of aortic biomechanical and geometrical properties based on in-vivo measurements (as likelihood) and ex-vivo biaxial extension data (as prior distribution) ([Aggarwal et al., 2025](#)). Moreover, data-driven model developed in [Aggarwal, Jensen, et al. \(2023\)](#) has been used in pyMechT via the `splineI1` and `splineI1I4` material models.

Conclusion and future plans

pyMechT fills an important gap and allows soft tissue biomechanics researchers to model ex-vivo testing setups in a fast, robust, and flexible manner. The package is numerically efficient and extensively documented. It has facilitated several publications, and we believe that it can benefit the wider community. In the future, we plan to extend the capabilities of the package to include more material models, such as inelastic (viscoelastic, plastic, damage, growth & remodeling) pre-defined formulations, and other ex-vivo setups (such as microindentation using Hertz contact model). Lastly, the package could be coupled with others to allow multi-physics simulations, such as for hemodynamics ([Coccarelli et al., 2021a, 2021b](#)) and biochemical regulation ([Coccarelli et al., 2024](#)).

Author contributions

Ankush Aggarwal: Conceptualization, Methodology, Software, Writing - Original Draft, Supervision. **Ross Williams:** Methodology, Software, Writing - Review and Editing. **Claire Rosnel:** Software, Testing. **Silvia Renon:** Testing. **Jude M. Hussain:** Testing. **André F. Schmidt:** Testing. **Shiting Huang:** Software. **Sean McGinty:** Supervision, Writing - Review and Editing. **Andrew McBride:** Supervision, Writing - Review and Editing.

References

Aggarwal, A., Coccarelli, A., & McGinty, S. (2025). *Heterogeneity and multi-scale modeling in vascular biomechanics* (J. Liao & J. Y. Wong, Eds.). Springer Nature Switzerland. ISBN: 978-3-031-81742-7

- Aggarwal, A., Hudson, L. T., Laurence, D. W., Lee, C.-H., & Pant, S. (2023). A bayesian constitutive model selection framework for biaxial mechanical testing of planar soft tissues: Application to porcine aortic valves. *Journal of the Mechanical Behavior of Biomedical Materials*, 138, 105657. <https://doi.org/10.1016/j.jmbbm.2023.105657>
- Aggarwal, A., Jensen, B. S., Pant, S., & Lee, C.-H. (2023). Strain energy density as a gaussian process and its utilization in stochastic finite element analysis: Application to planar soft tissues. *Computer Methods in Applied Mechanics and Engineering*, 404, 115812. <https://doi.org/10.1016/j.cma.2022.115812>
- Coccarelli, A., Carson, J. M., Aggarwal, A., & Pant, S. (2021a). A framework for incorporating 3D hyperelastic vascular wall models in 1D blood flow simulations. *Biomechanics and Modeling in Mechanobiology*, 20(4), 1231–1249. <https://doi.org/10.1007/s10237-021-01437-5>
- Coccarelli, A., Carson, J. M., Aggarwal, A., & Pant, S. (2021b). *1D-hyperelastic-haemodynamics: Version1* (Version V1). Zenodo. <https://doi.org/10.5281/zenodo.4522152>
- Coccarelli, A., Pant, S., Polydoros, I., & Harraz, O. F. (2024). A new model for evaluating pressure-induced vascular tone in small cerebral arteries. *Biomechanics and Modeling in Mechanobiology*, 23(1), 271–286. <https://doi.org/10.1007/s10237-023-01774-7>