# Hyperelastics.jl: A Julia package for hyperelastic material modelling with a large collection of models

**Carson Farmer** [1] and **Hector Medina** [1]

**1** School of Engineering, Liberty University, Lynchburg, VA, United States

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:** Open Journals

**Reviewers:**

- @openjournals

**Submitted:** 01 January 1970
**Published:** unpublished

## Summary

`Hyperelastics.jl` is a Julia (Bezanson et al., 2017) implementation for the largest (70+) collection of hyperelastic material models in existence. The package provides a set of analytical and data-driven strain energy density functions (SEDF) and the tools required to calibrate the models to material tests. The package is designed to leverage multiple-dispatch to define a common set of functions for calculating the SEDF, Second Piola Kirchoff stress tensor, and the Cauchy stress tensor. The package provides: 1) a material model library that is AD compatible and 2) a set of extensible methods for easily defining and testing new material models. The package leverages the `ContinuumMechanicsBase.jl` pacakge for defining the continuum scale quantities and their corresponding relationships.

## Statement of Need

The development of `Hyperelastics.jl` began as a study of the accuracy for a variety of material models for a set of experimental data. Often, researchers rely on custom implementations of material models and the data fitting process to find material parameters that match their experimental data. Hyperelastic models can well represent the nonlinear stress-deformation behavior of many biological tissues (Wex et al., 2015) as well as engineering polymeric materials (Beda, 2014).

The SEDFs included in this package cover most (if not all) of the available analytical models from the literature to date, from constitutive to phenomelogical models. Furthermore, a selection of data-driven models are incldued as a starting point for the development of new methods.

`Hyperelastics.jl` is part of a spinoff Multi-Scale Material Modelling ($M^3$) Suite being developed by Vagus LLC (www.vagusllc.com), as a byproduct result of ongoing multi-functional material research being carried out in the Translational Robotics and Controls Engineering Research (TRACER) Lab at Liberty University. A pure Julia implementation allows for the use of automatic differentiation (AD) packages to calculate the partial derivatives of the SEDF. `Hyperelastics.jl` is designed to leverage multiple-dispatch to define a common set of functions for calculating the SED, Second Piola Kirchoff Stress Tensor, and the Cauchy Stress Tensor. The package provides a set of hyperelastic models and an interface to `Optimization.jl` (Dixit & Rackauckas, 2023) for fitting model parameters.

Currently, most commercial finite element codes only offer a limited number, often less than 10, of hyperelastic models which limits the extent to which researchers are able to accurately model a given material. The closest project to `Hyperelastics.jl` is the `matADi` project by Andreas Dutzler (Dutzler, 2023) which has AD support for 18 material models.

## Comparison with Similar Packages

Similar to the `matADi` project by Andreas Dutzler, `Hyperelastics.jl` provides AD compatible implementations of hyperelastic models. However, `Hyperelastics.jl` provides a significantly larger set of material models and offers the potential for including different compressible material model terms. `matADi` does include models for anisotropic hyperelastic materials which is not support in the current version of `Hyperelastics.jl`. `matADi` is focused on python based implementations while `Hyperelastics.jl` is a pure Julia implementation.

For using material models in a simulation, the leading commercial finite element analysis (FEA) programs, such as Abaqus, Ansys, and LS-DYNA, provide a significantly smaller set of hyperelastic models (often less than 10). Implementing a new model in the commercial programs is often a complex and time-consuming process requiring explicit definition of different partial derivative terms. Furthermore, the model implementations are not compatible between commercial programs. To fill the commercial gap, PolymerFEM link provides a slightly larger set of models for implementation with a focus on visco-elastic and visco-plastic models. From the inital work done with `Hyperelastics.jl`, it is expected that the AD compatibility will allow for a significant improvement in terms of model construction for visco-hyperelastic models. More recently, Wan et al. (Wan et al., 2024) implemented over 70 hyperelastic material models as Abaqus subroutines. The implementation is not AD compatible and is limited to the Abaqus FEA software.

The `Hyperelastics.jl` package aims to fill the gap of providing an open-source and AD compatible implementation of the largest set of Hyperelatic material models available. AD-compatibility reduces the time from model selection to implementation when compared to implementing a new model in a commercial program. Furthermore, the open-source nature allows for the models to be accesible to researchers for further study.

## Use Case: Fitting a Hyperelastic Model to Experimental Data

The common workflow for modelling a new hyperelastic material is shown in figure Figure 1. The process commonly begins with the user proving a set of stress-stretch data for a set of uniaxial and/or biaxial tension/compression experiments. The data is loaded into `HyperelasticUniaxialTest` or `HyperelasticBiaxialTest` depending on the nature of the experiment. If the material is assumed to be incompressible, the data for the other principal stretches is calculated. The selection of hyperelastic models is based on multiple dispatch. The primary functions for computations are: `StrainEnergyDensity`, `SecondPiolaKirchoffStressTensor`, and `CauchyStressTensor`. The Second Piola-Kirchoff and Cauchy Stress Tensors are computed by using the partial derivatives of the strain energy density function. By using AD, the implementation of partial derivatives for the majority of the models can be skipped. Furthmore, extensions are provided to load compatible AD functions based on the AD-backend selected. Material models are considered as `structs` and multiple dispatch is used to select the correct equation for the given model. The model parameters are similarly stored in either `structs`, `NamedTuples`, or other field-based data-types such as those in `ComponentArrays.jl` and `LabelledArrays.jl`.

Once a user has their experimental data, model, and model parameters, an extension is loaded when `Optimization.jl` is used to load a function for calibrating the material model to the experimental data. The `HyperelasticProblem` function take the experimental data, model, model parameters, and AD-backend and creates an `OptimizationProblem` for use with the solvers from `Optimization.jl`. Once the model is calibrated, the `predict` function is used to predict the response of the model to the experimental data. The results are can then be plotted to compare the experimental data to the model prediction.

Additionally, by leveraging multiple dispatch, a selection of data-driven hyperelastic material models have been implemented for use with the same interface as the analytical models. This allows for rapid development of new models and the inclusion of new data-driven models as they become available.

Lastly, with the optimized model parameters, the material model is able to be implemented into a larger simulation or analysis. Some examples would include performing a finite element simulation with `Gridap.jl` or `Ferrite.jl` or interpreting the model parameters in the context of the material microstructure. The package is designed to be extensible and to allow for the rapid development of new models and the inclusion of new data-driven models as they become available.

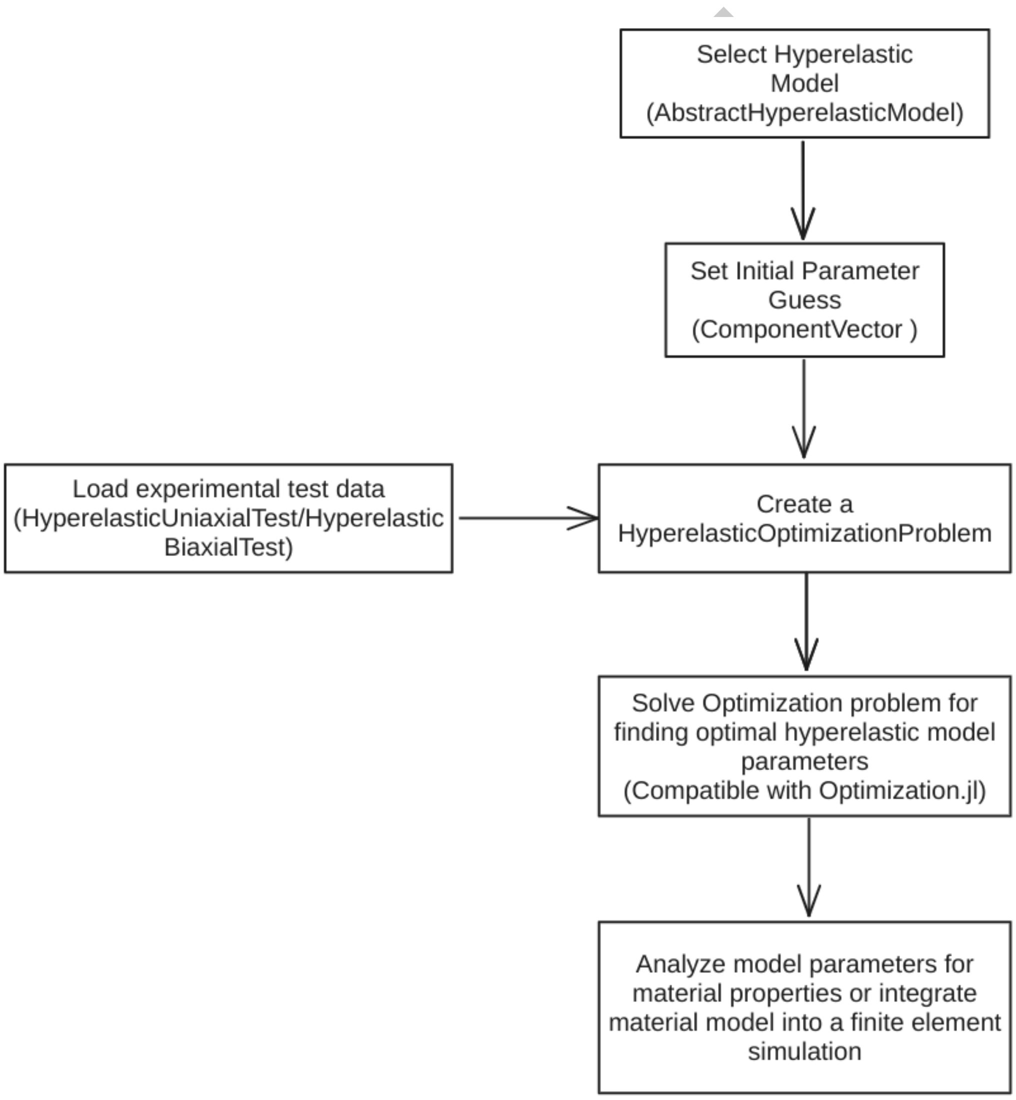
Select Hyperelastic Model (AbstractHyperelasticModel)
Set Initial Parameter Guess (ComponentVector )
Load experimental test data (HyperelasticUniaxialTest/Hyperelastic BiaxialTest)
Create a HyperelasticOptimizationProblem
Solve Optimization problem for finding optimal hyperelastic model parameters (Compatible with Optimization.jl)
Analyze model parameters for material properties or integrate material model into a finite element simulation

**Figure 1:** The most common use case for `Hyperelastics.jl` is to go from experimental or *ab initio* stress-stretch data for uniaxial or biaxial test(s) and proceed to fit and implement the model into a larger simulation or analysis. The respective data types used throughout the process are shown in the figure.

## Example Usage

For an example of going from experimental data to fitting a model, refer to the package documentation.

## Availability

Hyperelastics.jl can be found on github.

## Acknowledgements

The TRACER Lab is supported by the School of Engineering and the Center for Engineering Research and Education (CERE) at Liberty University.

## References

Beda, T. (2014). An approach for hyperelastic model-building and parameters estimation a review of constitutive models. *European Polymer Journal*, *50*, 97–108.

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Dixit, V. K., & Rackauckas, C. (2023). *Optimization.jl: A unified optimization package* (Version v3.12.1). Zenodo. https://doi.org/10.5281/zenodo.7738525

Dutzler, A. (2023). *matADi: Material definition with automatic differentiation*. https://github.com/adtzlr/matadi

Wan, X., Zhang, Y., Zhang, Q., Zhang, L., & Li, F. (2024). User subroutines platform development for rubber hyperelastic constitutive models and its application in finite element analysis. *Computational Materials Science*, *237*, 112885.

Wex, C., Arndt, S., Stoll, A., Bruns, C., & Kupriyanova, Y. (2015). Isotropic incompressible hyperelastic models for modelling the mechanical behaviour of biological tissues: A review. *Biomedical Engineering/Biomedizinische Technik*, *60*(6), 577–592.