Symbolic Regression of Hyperelastic Laws with a Stress-Based Custom Loss: Invariant Features, Robust Implementation, and a Mooney–Rivlin Case Study

(Your Name)

September 6, 2025

Abstract

We present a robust workflow to discover interpretable hyperelastic strain-energy functions by symbolic regression (SR). The method regresses a candidate energy $\Psi(I_1^b, I_2^b, J)$ from data while enforcing physics through a *stress-based custom loss*. Given features $X = [I_1^b, I_2^b, J]^\top$ and first Piola–Kirchhoff targets, the loss differentiates Ψ w.r.t. invariants, assembles stress, and minimizes the MSE on selected stress components (e.g. P_{11}). We detail implementation aspects in Julia (`SymbolicRegression.jl` + `DynamicDiff.jl`), common pitfalls (elementwise finiteness checks, function signatures, scalar vs. vector indexing), and demonstrate the pipeline on a Mooney–Rivlin benchmark with synthetic data. The resulting framework is stable, extensible, and preserves model interpretability.

1 Motivation and Contributions

Classical hyperelastic models (Neo–Hookean, Mooney–Rivlin, Ogden) postulate $\Psi(\cdot)$ and fit parameters. Symbolic regression (SR) instead *discovers* a functional form from data with algebraic building blocks, yielding concise, interpretable constitutive laws.

Contributions.

- A stress-driven custom loss that ties SR directly to continuum mechanics: we differentiate $\Psi(I_1^b, I_2^b, J)$, assemble $\boldsymbol{\sigma}$ and $\mathbf{P}$, and compare against measured/ground-truth stresses.
- A pragmatic, numerically robust Julia implementation with invariant features, volumetric stabilization, and strict NaN/Inf guards.
- A Mooney–Rivlin case study with synthetic uniaxial data, showing that SR rediscovers a compact and physically meaningful energy.
- Documentation of subtle implementation pitfalls and their fixes: (i) finiteness checks on arrays; (ii) argument order for the stress assembler; (iii) scalar indexing for batched derivatives.

2 Background

2.1 Kinematics and invariants

Let $\mathbf{F}$ be the deformation gradient with $J = \det \mathbf{F} > 0$. Define $\mathbf{B} = \mathbf{F}\mathbf{F}^\top$ and $\mathbf{C} = \mathbf{F}^\top\mathbf{F}$. Isochoric (bar) invariants remove volumetric stretch via

$$\bar{\mathbf{B}} = J^{-2/3}\mathbf{B}, \qquad I_1^b = \operatorname{tr}\left(\bar{\mathbf{B}}\right), \qquad I_2^b = \tfrac{1}{2}\left[(\operatorname{tr}\bar{\mathbf{B}})^2 - \operatorname{tr}\left(\bar{\mathbf{B}}^2\right)\right]. \tag{1}$$

We consider $\Psi = \Psi(I_1^b, I_2^b, J)$, with a volumetric penalty $U(J)$ embedded in Ψ.

2.2 Stresses from Ψ

A standard deviatoric/volumetric split for the Cauchy stress is

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\text{dev}} + \boldsymbol{\sigma}_{\text{vol}}, \tag{2}$$

$$\boldsymbol{\sigma}_{\text{dev}} = \frac{2}{J}\left(g_1\,\bar{\mathbf{B}} + g_2\,(I_1^b\mathbf{I} - \bar{\mathbf{B}})\right)_{\text{dev}}, \tag{3}$$

$$\boldsymbol{\sigma}_{\text{vol}} = J\,g_J\,\mathbf{I}, \tag{4}$$

where $g_1 = \partial\Psi/\partial I_1^b$, $g_2 = \partial\Psi/\partial I_2^b$, $g_J = \partial\Psi/\partial J$, and $\operatorname{dev}(\cdot)$ is the deviatoric projection. The first Piola–Kirchhoff stress follows as

$$\mathbf{P} = J\,\boldsymbol{\sigma}\,\mathbf{F}^{-\top}. \tag{5}$$

3 SR pipeline and custom loss

Data layout

We train on samples $n = 1, \dots, N$ with

$$X = \begin{bmatrix} I_1^b \\ I_2^b \\ J \end{bmatrix} \in \mathbb{R}^{3\times N}, \qquad \{\mathbf{F}_n\}_{n=1}^N, \qquad y \in \mathbb{R}^N \text{ where } y_n = (P_{11})_n.$$

SR proposes a candidate $\Psi(\cdot)$; we use automatic differentiation to obtain (g_1, g_2, g_J) per sample, assemble $\mathbf{P}$, and minimize MSE on P_{11}.

Loss definition (per-sample and batched)

For each n:

$$g_{1,n} = \frac{\partial\Psi}{\partial I_1^b}\Big|_{X_n}, \qquad g_{2,n} = \frac{\partial\Psi}{\partial I_2^b}\Big|_{X_n}, \qquad g_{J,n} = \frac{\partial\Psi}{\partial J}\Big|_{X_n}, \tag{6}$$

$$\mathbf{P}_n = \texttt{P\_from\_Psi\_full}\big(g_{1,n}, g_{2,n}, g_{J,n}, \mathbf{F}_n\big), \qquad \ell_n = \big(\,(\mathbf{P}_n)_{11} - y_n\,\big)^2. \tag{7}$$

The loss is $\mathcal{L} = \frac{1}{N}\sum_n \ell_n$.

Numerical guards & common pitfalls

Finiteness checks. `isfinite(x)` is scalar-only. For arrays use `all(isfinite, x)`. Accidentally forming tuples in `all` calls can trigger cryptic *_all_tuple* frames; use the pattern:

```
if !(all(isfinite, v1) && all(isfinite, v2)) return Inf end.
```

Function signatures. Our assembler is

```
P_from_Psi_full(::Real, ::Real, ::Real, ::AbstractMatrix).
```

Always pass scalars *per sample*: `P_from_Psi_full(g1[n], g2[n], gJ[n], F_list[n])`.

Indexing. Mixing per-batch vectors with per-sample operations causes signature mismatches; index before calling the assembler.

4 Mooney–Rivlin case study

4.1 Model and synthetic data

A compressible Mooney–Rivlin energy reads

$$\Psi_{\mathrm{MR}}(I_1^b, I_2^b, J) = C_1\,(I_1^b - 3) + C_2\,(I_2^b - 3) + U(J), \tag{8}$$

with $U(J)$ a volumetric penalty (e.g. $U = \frac{\kappa}{2}(J-1)^2$ or a $\kappa \log J$ form). We generate synthetic uniaxial data by prescribing stretches λ and forming

$$\mathbf{F} = \mathrm{diag}(\lambda, \lambda_t, \lambda_t), \qquad \lambda_t = \lambda^{-1/2} \text{ for } J = 1 \text{ (incompressible case)},$$

then compute $\mathbf{P}$ from Ψ_{MR} and collect $X = [I_1^b, I_2^b, J]$ and $y = P_{11}$.

4.2 SR configuration (sketch)

We allow a compact operator set to bias towards interpretable formulas:

$$\mathcal{O}_{\mathrm{binary}} = \{+, -, \times\}, \quad \mathcal{O}_{\mathrm{unary}} = \{\mathrm{safe\_log}, \mathrm{safe\_sqrt}, \exp\}.$$

The search runs for a fixed number of iterations with our custom loss. A validation split helps avoid overfitting.

4.3 Results and discussion

On synthetic Mooney–Rivlin data, SR typically recovers an energy of the form

$$\Psi^\star \approx \tilde{C}_1\,(I_1^b - 3) + \tilde{C}_2\,(I_2^b - 3) + \tilde{U}(J),$$

up to algebraic transforms. Stress–stretch curves (P_{11} vs. λ) match the reference closely, confirming that a stress-driven loss guides SR to physically meaningful minima.

5 Reproducibility recipe

1. **Data.** Generate synthetic uniaxial data from a known Mooney–Rivlin law:
   1.1. Choose (C_1, C_2, κ) and a stretch grid $\lambda \in [\lambda_{\min}, \lambda_{\max}]$.
   1.2. Form $\mathbf{F}(\lambda)$, compute J, $\bar{\mathbf{B}}$, and (I_1^b, I_2^b).
   1.3. Evaluate $\mathbf{P}$ from the analytic model; set $y = P_{11}$.
2. **Features.** Build $X = \begin{bmatrix} I_1^b \\ I_2^b \\ J \end{bmatrix}$ column-wise.
3. **SR config.** Choose compact operator sets; cap complexity (optional).
4. **Loss.** Use the custom stress MSE on P_{11} with derivative-based assembly.
5. **Validation.** Hold out samples or stretches to monitor generalization.

6 Common issues & fixes

- **`MethodError: isfinite(::Vector{Float64}).`** Use `all(isfinite, v)` for arrays; keep `isfinite(x)` for scalars.
- **`no method matching P_from_Psi_full(::Vector,...)`** Ensure you pass *scalars*: `g1[n], g2[n], gJ[n]` and `F_list[n]`.
- **Silent NaNs/Infs.** Early-return `Inf` from the loss if any sample becomes non-finite; assert at assembly points.

7 Outlook

The framework extends to richer operator sets (e.g. rational or piecewise terms), multi-component stress fitting, and constraints (e.g. convexity surrogates). Incorporating weak polyconvexity checks during search is promising.

A Flow diagram (SR pipeline)

B Loading Scenarios

We summarize three standard loading paths used for identification and validation of incompressible and nearly-incompressible hyperelastic models. For each case we give a representative deformation gradient $\mathbf{F}$, principal stretches, and a schematic.

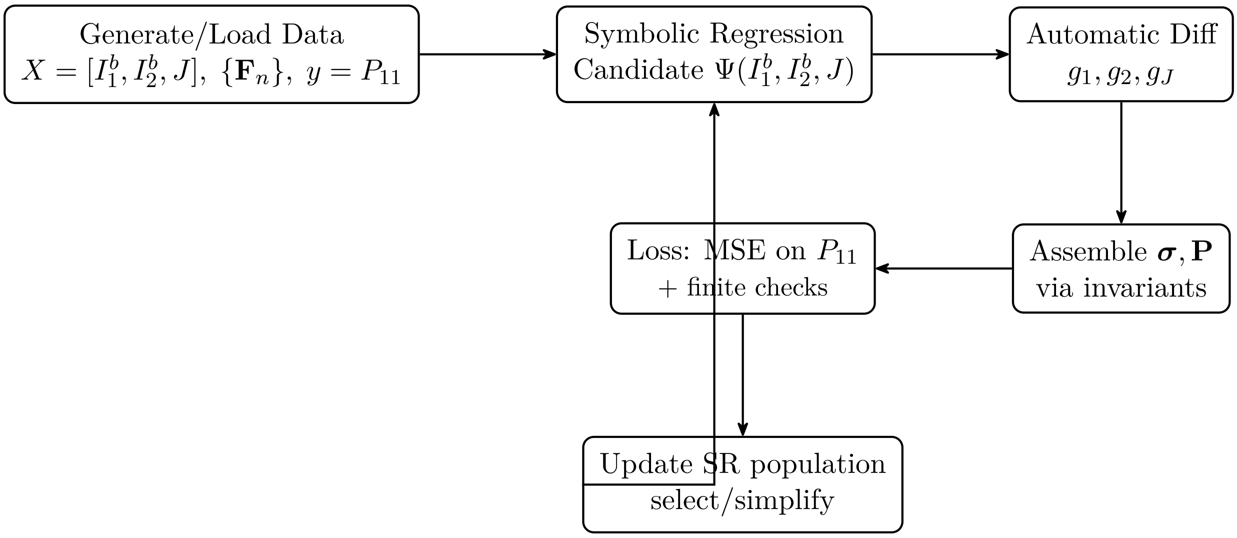

Figure 1: End-to-end pipeline tying SR to continuum stress assembly.

B.1 Uniaxial Tension/Compression

For a stretch λ in the x-direction and incompressibility (or nearly so), a common choice is

$$\mathbf{F}_{\text{uni}} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda^{-1/2} & 0 \\ 0 & 0 & \lambda^{-1/2} \end{bmatrix}, \qquad J = \det \mathbf{F}_{\text{uni}} = 1 \text{ (incompressible)}.$$

The associated invariants of the left Cauchy–Green tensor $\mathbf{b} = \mathbf{F}\mathbf{F}^\top$ are

$$I_1 = \operatorname{tr}(\mathbf{b}), \qquad I_2 = \tfrac{1}{2}\Big[(\operatorname{tr}\mathbf{b})^2 - \operatorname{tr}\Big(\mathbf{b}^2\Big)\Big].$$

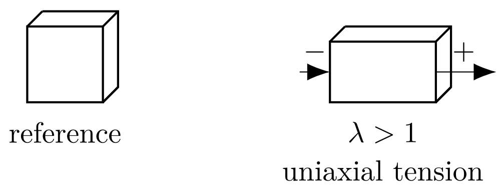

Figure 2: Uniaxial loading: reference and deformed sketches.

B.2 Equibiaxial Tension

Two equal in-plane stretches λ with incompressibility give

$$\mathbf{F}_{\text{equi}} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda^{-2} \end{bmatrix}, \qquad J = 1.$$

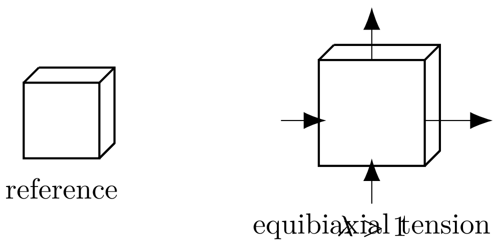

Figure 3: Equibiaxial loading: reference and deformed sketches.

B.3 Plane Strain (Pure Shear Test)

A typical plane-strain (pure-shear) choice is

$$\mathbf{F}_{\rm ps} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad J = 1.$$

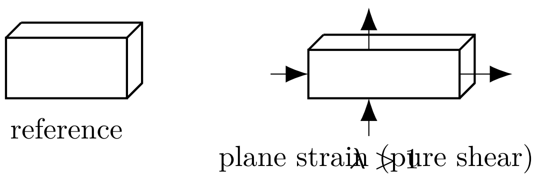

Figure 4: Plane-strain loading: reference and deformed sketches.

Remarks on Stress Extraction. In our pipeline, nominal stresses (first Piola–Kirchhoff) P_{11} are assembled from the energy density $\Psi(I_1, I_2, J)$ via

$$P = \frac{\partial \Psi}{\partial \mathbf{F}} = 2\,\frac{\partial \Psi}{\partial I_1}\,\mathbf{F} - 2\,\frac{\partial \Psi}{\partial I_2}\,\mathbf{F}^{-\top}\mathbf{b}\mathbf{F}^{-\top} + \frac{\partial \Psi}{\partial J}\,J\,\mathbf{F}^{-\top},$$

specialized to each $\mathbf{F}$ above. This provides consistent targets for symbolic regression when fitting Ψ from data.

C Gradient of Strain Energy with Respect to Invariants

We consider isotropic hyperelastic materials whose strain-energy density is written as $\Psi = \Psi(I_1, I_2, J)$, where $I_1 = \operatorname{tr}\mathbf{C}$, $I_2 = \frac{1}{2}\big[(\operatorname{tr}\mathbf{C})^2 - \operatorname{tr}(\mathbf{C}^2)\big]$ are the first two invariants of the right Cauchy–Green tensor $\mathbf{C} = \mathbf{F}^\top\mathbf{F}$, and $J = \det\mathbf{F}$ is the Jacobian of the deformation. Invoking the chain rule and the standard identities $\partial I_1/\partial\mathbf{F} = 2\mathbf{F}$, $\partial I_2/\partial\mathbf{F} = 2(I_1\mathbf{F} - \mathbf{F}\mathbf{C})$, and $\partial J/\partial\mathbf{F} = J\,\mathbf{F}^{-\top}$, the first Piola–Kirchhoff stress reads (see, e.g., [1, 2, 3])

$$\mathbf{P} \;=\; 2\,\Psi_{I_1}\,\mathbf{F} \;+\; 2\,\Psi_{I_2}\,\big(I_1\,\mathbf{F} - \mathbf{F}\mathbf{C}\big) \;+\; \Psi_J\,J\,\mathbf{F}^{-\top}, \tag{9}$$

where $\Psi_{(\cdot)}$ denotes partial derivatives w.r.t. the listed arguments. This form is objective and isotropic by construction since Ψ depends only on invariants.

C.1 Isochoric–volumetric split

For nearly incompressible media, it is common to split $\Psi = \Psi_{\text{iso}}(I_1^b, I_2^b) + U(J)$, where $I_1^b = J^{-2/3} I_1$ and $I_2^b = J^{-4/3} I_2$ are the modified (isochoric) invariants of $\mathbf{B} = \mathbf{F}\mathbf{F}^\top$ and $U(J)$ penalizes volume changes [2, 3]. Denoting $g_1 = \partial\Psi/\partial I_1^b$, $g_2 = \partial\Psi/\partial I_2^b$ and $g_J = \partial U/\partial J$, the stress becomes

$$\mathbf{P} \;=\; 2\Big[g_1\, J^{-2/3}\mathbf{F} \;+\; g_2\, J^{-2/3}\big(I_1^b\,\mathbf{F} - J^{-2/3}\mathbf{F}\mathbf{C}\big)\Big] \;+\; g_J\, J\, \mathbf{F}^{-\top}. \tag{10}$$

The first bracket is deviatoric (trace-free in Cauchy form), while the second term produces the hydrostatic response. If $U(J) = \frac{1}{2}\,\kappa(J-1)^2$ or $U(J) = \frac{1}{2}\,\kappa(\ln J)^2$, then $g_J = \kappa\,(J-1)$ or $g_J = \kappa\, \ln J/J$, respectively, and the volumetric nominal stress is always $g_J\, J\, \mathbf{F}^{-\top}$.

C.2 Automatic & symbolic differentiation in learning Ψ

When Ψ is *learned* from data (e.g., via symbolic regression or neural networks), one differentiates Ψ with respect to its inputs to obtain g_1, g_2, and g_J and then assembles stresses using (10). This strategy, sometimes called *Sobolev training* in machine learning, ties the model to mechanics by fitting not only function values but also their derivatives [4]. Recent works show that learning $\Psi(I_1, I_2, J)$ (or $\Psi(I_1^b, I_2^b, J)$) with automatic/symbolic differentiation yields accurate and interpretable hyperelastic models [5, 6, 7].

C.3 Check against the implementation (Julia)

Your Julia routine `P_from_Psi_full` computes the isochoric part exactly as in (10), using $g_1 = \partial\Psi/\partial I_1^b$ and $g_2 = \partial\Psi/\partial I_2^b$ from automatic differentiation. This is correct and consistent with the literature. For the volumetric term, however, the code applies

$$\mathbf{P}_{\text{vol}}^{(\text{code})} \;=\; g_J\, \frac{J}{2}\, \mathbf{F}^{-\top},$$

whereas the continuum result is $\mathbf{P}_{\text{vol}} = g_J\, J\, \mathbf{F}^{-\top}$, cf. (9). The extra factor 1/2 underestimates volumetric stiffness by a factor of two. We recommend replacing `(dPsi_dJ * (J/2.0))` by `(dPsi_dJ * J)` in both the ground-truth generator and the loss evaluation.

Diagnostics and best practices. (i) Ensure Ψ is normalized so that $g_1, g_2, g_J = 0$ at the reference state $(I_1^b, I_2^b, J) = (3, 3, 1)$, or include a small penalty for residual stresses at $\mathbf{F} = \mathbf{I}$. (ii) When using $U(J) = \frac{1}{2}\kappa(\ln J)^2$, guard $\ln J$ by clamping J away from 0 in the loss. (iii) Always report the deformation modes (uniaxial, equibiaxial, pure shear, volumetric) used for fitting and validation since they excite different combinations of (I_1^b, I_2^b, J) [1, 2].

Complexity	Loss	Equation
1	5.344	I_{2b}
3	0.237	$I_{2b} + I_{1b}$
5	0.026	$I_{2b} + (I_{1b} \cdot 0.834)$
7	0.002	$J \cdot (I_{2b} + (I_{1b} \cdot 0.834))$
9	1.259×10^{-5}	$(J - 0.129) \cdot (I_{1b} + (I_{2b} \cdot 1.125))$
11	4.611×10^{-6}	$(((J \cdot 1.597) - 0.617) \cdot I_{2b}) + (I_{1b} \cdot 0.870)$
13	2.288×10^{-8}	$J \cdot ((I_{1b} \cdot 0.870) + (((J \cdot 0.181) - -0.799) \cdot I_{2b}))$
15	3.333×10^{-10}	$((((J \cdot 0.169) + 0.820) \cdot I_{2b}) + (0.878 \cdot I_{1b})) \cdot (J - 0.009)$

Table 1: Top discovered expressions (rounded to 3 decimals in equations).

D Regression results

References

References

[1] R. W. Ogden, *Non-Linear Elastic Deformations*, Dover, 1997.

[2] G. A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*, Wiley, 2000.

[3] J. Bonet and R. D. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*, 2nd ed., Cambridge University Press, 2008.

[4] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Świrszcz, and R. Pascanu, “Sobolev Training for Neural Networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] M. Flaschel, S. Kumar, and L. De Lorenzis, “Unsupervised discovery of interpretable hyperelastic constitutive laws,” *Computer Methods in Applied Mechanics and Engineering*, 381:113852, 2021.

[6] R. Abdusalamov, M. Hillgärtner, and M. Itskov, “Automatic generation of interpretable hyperelastic material models by symbolic regression,” *International Journal for Numerical Methods in Engineering*, 124(15):3373–3398, 2023.

[7] G. Kissas, C. Wang, and G. E. Karniadakis, “The language of hyperelastic materials,” *Computer Methods in Applied Mechanics and Engineering*, 428:170530, 2024.

E Extending Symbolic Regression to Viscoelasticity

Notation. We use $\mathbf{F}$ for the deformation gradient, $\mathbf{C} = \mathbf{F}^\top\mathbf{F}$, $\mathbf{B} = \mathbf{F}\mathbf{F}^\top$, $J = \det \mathbf{F}$, the rate-of-deformation tensor $\mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^\top)$ with $\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1}$, and the Cauchy stress $\boldsymbol{\sigma}$.

E.1 Two-stage learning strategy

We propose a staged workflow:

1. **Elastic identification.** Learn $\Psi_{\text{iso}}(I_1^b, I_2^b)$ and $U(J)$ using low-rate or equilibrium data as in Sec. C (fit $g_1 = \partial\Psi/\partial I_1^b$, $g_2 = \partial\Psi/\partial I_2^b$, $g_J = \partial\Psi/\partial J$).
2. **Viscous augmentation.** With Ψ fixed, learn a viscous contribution $\boldsymbol{\sigma}_\text{v}$ from rate-dependent histories (relaxation, creep, ramps, cyclic), enforcing dissipation $\mathcal{D} = \boldsymbol{\sigma}_\text{v} : \mathbf{D} \geq 0$.

E.2 Kelvin–Voigt baseline (finite strain)

A robust baseline is a deviatoric Kelvin–Voigt model combined with the learned elastic stress:

$$\boldsymbol{\sigma} \;=\; \boldsymbol{\sigma}_\text{e}(\mathbf{B}; I_1^b, I_2^b, J) \;+\; 2\,\eta_\text{dev}\,\mathbf{D}_\text{dev} \;+\; \kappa_\text{v}\;\text{tr}(\mathbf{D})\,\mathbf{I}, \tag{11}$$

where $\boldsymbol{\sigma}_\text{e}$ is the elastic Cauchy stress derived from Ψ, and $\mathbf{D}_\text{dev} = \mathbf{D} - \frac{1}{3}\,\text{tr}(\mathbf{D})\,\mathbf{I}$. Often $\kappa_\text{v} = 0$ for nearly incompressible materials. In nominal form, $\mathbf{P} = J\,\boldsymbol{\sigma}\,\mathbf{F}^{-\top}$. This model corresponds to a quadratic pseudo-potential of dissipation $\Phi = \eta_\text{dev}\,\mathbf{D}_\text{dev} : \mathbf{D}_\text{dev} + \frac{1}{2}\kappa_\text{v}(\text{tr}\,\mathbf{D})^2$, and guarantees $\mathcal{D} = 2\,\eta_\text{dev}\|\mathbf{D}_\text{dev}\|^2 + \kappa_\text{v}(\text{tr}\,\mathbf{D})^2 \geq 0$.

E.3 Generalized Maxwell (Prony series) with internal variables

To capture fading memory, augment the deviatoric stress by N Maxwell branches with Prony series parameters:

$$\dot{\mathbf{s}}_i + \frac{1}{\tau_i}\,\mathbf{s}_i \;=\; 2\,G_i\,\mathbf{D}_\text{dev}, \qquad \boldsymbol{\sigma}'_\text{v} = \sum_{i=1}^{N}\mathbf{s}_i, \qquad G(t) = G_\infty + \sum_{i=1}^{N} G_i\,e^{-t/\tau_i}, \tag{12}$$

where $(\cdot)'$ denotes deviatoric part. The unconditionally stable exact update over a time step Δt is

$$\mathbf{s}_i^{n+1} \;=\; \alpha_i\,\mathbf{s}_i^n \;+\; 2\,G_i\,(1-\alpha_i)\,\frac{\mathbf{E}_\text{dev}^{n+1} - \mathbf{E}_\text{dev}^n}{\Delta t}, \quad \alpha_i = e^{-\Delta t/\tau_i}, \tag{13}$$

with $\mathbf{E}$ a chosen strain measure (for small steps, $\mathbf{E}_\text{dev} \approx \int \mathbf{D}_\text{dev}\,dt$). In practice, one may also use the midpoint rule $\mathbf{s}_i^{n+1} = \alpha_i\,\mathbf{s}_i^n + 2\,G_i\,\alpha_i\,\Delta t\,\mathbf{D}_\text{dev}^{n+1}$, which differentiates cleanly in AD frameworks. The full nominal stress is $\mathbf{P} = \mathbf{P}_\text{e} + J\,\boldsymbol{\sigma}_\text{v}\,\mathbf{F}^{-\top}$.

E.4 SR design choices for viscosity

We outline three complementary SR routes:

1. **Parametric Prony fit.** Choose $\{\tau_i\}$ on a log-spaced grid and learn nonnegative $\{G_i\}$ (and optionally G_∞), or learn both using positive reparameterizations ($G_i = \mathrm{softplus}(\cdot)$, $\tau_i = \mathrm{softplus}(\cdot)$). This preserves linearity of the branch update and guarantees $\mathcal{D} \geq 0$.

2. **SR for viscosity law.** Postulate $\boldsymbol{\sigma}'_{\mathrm{v}} = 2\,\mu(I_1^b, I_2^b, J, \mathcal{I}_{\mathbf{D}^b})\,\mathbf{D}_{\mathrm{dev}}$, where $\mathcal{I}_{\mathbf{D}^b}$ are invariants of the isochoric rate $\mathbf{D}^b$. Use SR to discover a sparse, interpretable $\mu(\cdot)$ with a nonnegativity constraint (e.g., softplus envelope).

3. **SR for evolution laws.** Keep the Maxwell structure but let the driving term be a sparse function discovered by SR: $\dot{\mathbf{s}}_i + \mathbf{s}_i/\tau_i = \sum_k c_{ik}\,\boldsymbol{\phi}_k(\mathbf{C}, \mathbf{D})$, enforcing $\sum_i \mathbf{s}_i : \mathbf{D} \geq 0$.

E.5 Losses, constraints, and differentiation through time

Given sequences $\{\mathbf{F}(t_m)\}_{m=0}^{M}$ and measured components of $\mathbf{P}$ or $\boldsymbol{\sigma}$, define

$$\mathcal{L} \;=\; \sum_{\mathrm{seq}} \sum_{m} w_m \left\| \mathcal{O}[\mathbf{P}(\mathbf{F}(t_m); \theta)] - y_m \right\|_2^2 \;+\; \lambda_{\mathrm{diss}} \sum_{m} \max\big(0,\, -\boldsymbol{\sigma}_{\mathrm{v}}(t_m) : \mathbf{D}(t_m)\big) \;+\; \lambda_{\mathrm{reg}} \|\theta\|_1, \tag{14}$$

where θ collects parameters (elastic and viscous). Time integration of (12) is differentiable: use exact updates (13) or an AD-friendly one-step scheme (implicit Euler or midpoint). For identifiability, include diverse histories (step-relaxation, creep, ramp at multiple rates, cyclic loading) and loading modes (uniaxial, equibiaxial, pure shear).

E.6 Implementation notes (Julia)

- Reuse the learned Ψ and its gradients (g_1, g_2, g_J) for the elastic part (Sec. C).
- Compute $\mathbf{D}$ per step via $\mathbf{D}^{n+1} = \frac{1}{2\Delta t}(\mathbf{F}^{n+1}\mathbf{F}^{n+1\top} - \mathbf{F}^n\mathbf{F}^{n\top})\mathbf{B}^{-1/2}$ or use the simple finite difference $\frac{1}{2\Delta t}(\mathbf{L} + \mathbf{L}^\top)$ with $\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1}$ if $\dot{\mathbf{F}}$ is available.
- Update $\{\mathbf{s}_i\}$ with $\alpha_i = \exp(-\Delta t/\tau_i)$; form $\boldsymbol{\sigma}'_{\mathrm{v}} = \sum_i \mathbf{s}_i$ and push-forward to $\mathbf{P}_{\mathrm{v}} = J\,\boldsymbol{\sigma}_{\mathrm{v}}\,\mathbf{F}^{-\top}$.
- Enforce positivity via reparameterization: $G_i = \mathrm{softplus}(\hat{G}_i)$, $\tau_i = \mathrm{softplus}(\hat{\tau}_i)$, $\eta_{\mathrm{dev}} = \mathrm{softplus}(\hat{\eta})$.

References

References

[1] G. A. Holzapfel, *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*, Wiley, 2000.

[2] J. C. Simo and T. J. R. Hughes, *Computational Inelasticity*, Springer, 1998.

[3] S. Reese and S. Govindjee, "A theory of finite viscoelasticity and numerical aspects," *International Journal of Solids and Structures*, 35(26–27):3455–3482, 1998.

[4] R. M. Christensen, *Theory of Viscoelasticity*, 2nd ed., Dover, 2003.

[5] Y. C. Fung, *Biomechanics: Mechanical Properties of Living Tissues*, 2nd ed., Springer, 1993. (QLV)

[6] J. D. Ferry, *Viscoelastic Properties of Polymers*, 3rd ed., Wiley, 1980.