

Differentiable plasticity FEM in JAX

what a mechanical test sees, and how its derivatives flow

Raj Mahat
Georgia Institute of Technology

Learning notes, October 2026

These notes build a small, fully differentiable finite-element model of a punch test on an elastic–plastic plate, and use it to follow how a change in the hardening law travels through the computation to the measured load. The model was written on core JAX; the figures come from it or are drawn in TikZ. The material, geometry and parameter values are generic and chosen for teaching.

Contents

1	What a mechanical test sees	1
2	J₂ plasticity with combined hardening	2
3	The return map and its Jacobian	3
4	The incremental finite-element problem	4
5	Linearizing the forward model	8
6	A local Bayesian view with three parameters	12
7	Where this leads	12

1 What a mechanical test sees

A test imposes boundary data (a displacement, a traction, contact with a fixture) and a loading path. The body responds with fields $u(x, t)$, $\varepsilon(x, t)$, $\sigma(x, t)$ and internal variables $q(x, t)$. We observe only a compressed record: here the punch load at each depth, $y = \{L(d_k)\}_{k=1}^{N_d}$. The constitutive law is never measured directly. It reaches the data only after passing through the local plastic response, equilibrium with contact, the loading history and the readout (Fig. 1). Calibration is therefore an inverse problem,

$$y_{\text{obs}} = \mathcal{G}(\theta) + \epsilon, \tag{1}$$

and the derivative $\partial\mathcal{G}/\partial\theta$ is the object that tells us what the test can say about θ .

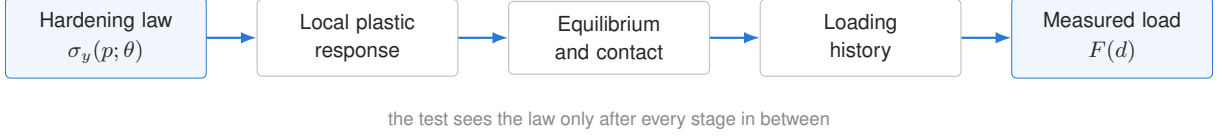


Figure 1: The chain from the hardening law to the measured load.

2 J₂ plasticity with combined hardening

With $s = \text{dev } \sigma$, back stress X and accumulated plastic strain p ,

$$f = \sigma_{eq}(s - X) - \sigma_y(p) \leq 0, \quad \sigma_{eq}(\xi) = \sqrt{\frac{3}{2} \xi : \xi}, \quad (2)$$

$$\dot{\varepsilon}^p = \dot{\gamma} n, \quad n = \frac{3}{2} \frac{\xi}{\sigma_{eq}(\xi)}, \quad \dot{p} = \dot{\gamma}, \quad (3)$$

$$\sigma_y(p) = \sigma_0 + Q(1 - e^{-\beta p}) \quad (\text{Voce, isotropic}), \quad (4)$$

$$\dot{X} = \frac{2}{3} C \dot{\varepsilon}^p - \gamma_k \dot{p} X \quad (\text{Armstrong–Frederick, kinematic}). \quad (5)$$

Isotropic hardening grows the elastic domain; kinematic hardening moves it (Fig. 2). Under a load reversal the second gives early reverse yielding, the Bauschinger effect (Fig. 3). Positivity of $\sigma_y(0)$ and $\sigma'_y \geq 0$ keep the incremental problem well posed. In tabulated form (as in commercial codes) σ_y is simply a function of p ; here we use the three-parameter Voce chart $\theta = (\sigma_0, Q, \beta)$.

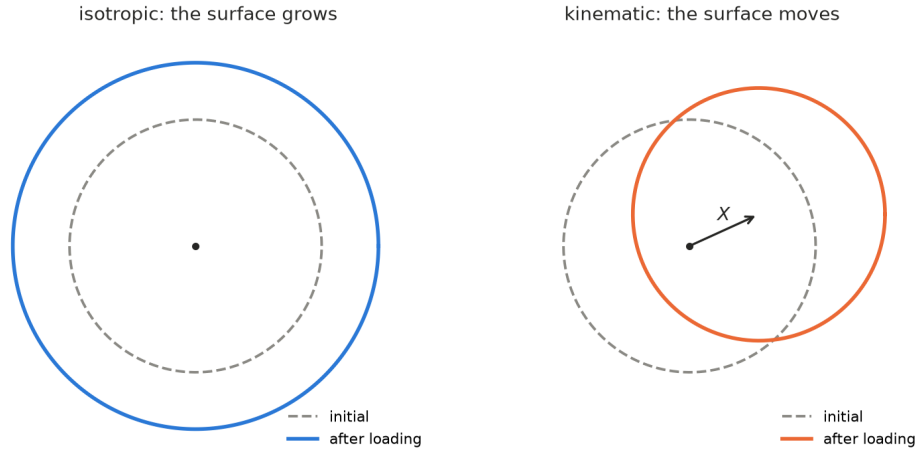


Figure 2: Deviatoric-plane view: isotropic hardening dilates the surface, kinematic hardening translates it by X .

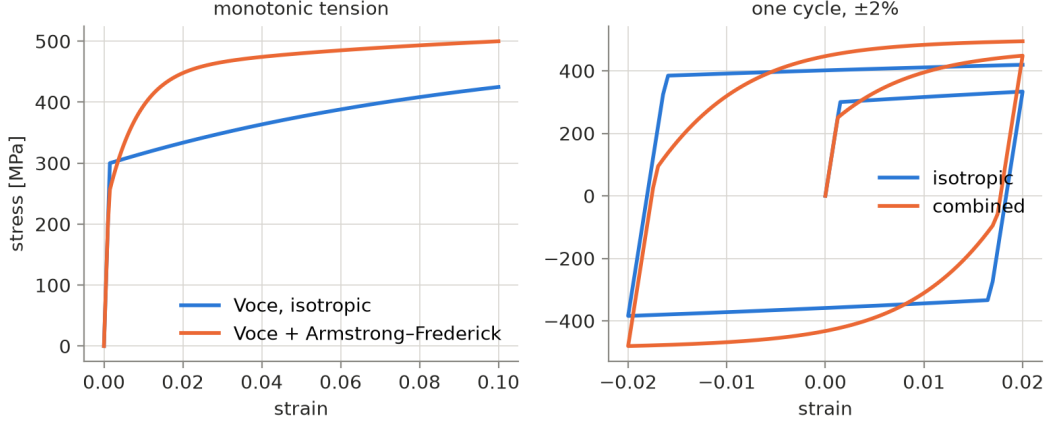


Figure 3: Monotonic tension and one strain cycle for pure Voce and for Voce + Armstrong-Frederick.

3 The return map and its Jacobian

Backward Euler over an increment, with local unknowns $v = (\sigma, \varepsilon^p, X, p)$, gives the local residual $h(v) = 0$ (Fig. 4): elastic update, flow rule, back-stress evolution, accumulation and yield consistency. Starting from the trial state $s^{\text{tr}} = 2G \text{dev}(\varepsilon - \varepsilon_n^p)$, everything collapses to one scalar equation for $\Delta\gamma$:

$$g(\Delta\gamma) = \left\| s^{\text{tr}} - \frac{X_n}{1 + \gamma_k \Delta\gamma} \right\| - \left(2G\sqrt{\frac{3}{2}} + \sqrt{\frac{2}{3}} \frac{C}{1 + \gamma_k \Delta\gamma} \right) \Delta\gamma - \sqrt{\frac{2}{3}} \sigma_y(p_n + \Delta\gamma) = 0. \quad (6)$$

In code this root is found with Newton inside `jax.lax.custom_root`. The implicit function theorem then gives every derivative of the result without differentiating through the iterations:

$$\frac{d\Delta\gamma}{d(\cdot)} = - \left(\frac{\partial g}{\partial \Delta\gamma} \right)^{-1} \frac{\partial g}{\partial (\cdot)}. \quad (7)$$

The algorithmic (consistent) tangent $\mathbb{D} = \partial\sigma_{n+1}/\partial\varepsilon_{n+1}$ is then a single `jax.jacfwd`. For pure isotropic hardening it agrees with the textbook closed form

$$\mathbb{D} = K \mathbf{1} \otimes \mathbf{1} + 2G\theta_1 \mathbb{I}_{\text{dev}} - 2G\theta_2 \hat{n} \otimes \hat{n}, \quad \theta_1 = 1 - \frac{3G\Delta\gamma}{\sigma_{eq}^{\text{tr}}}, \quad \theta_2 = \frac{1}{1 + H/3G} - (1 - \theta_1), \quad (8)$$

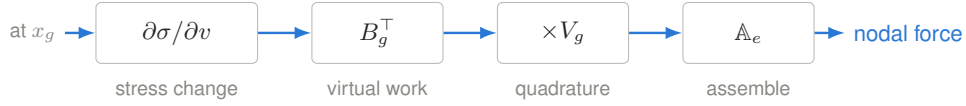
to round-off, and it needs no change for the combined model.

h_v — the return-map Jacobian

$h^\sigma = \sigma_k - \mathbb{C} : (\varepsilon_k - \varepsilon_k^p)$	elastic stress update
$h^{\varepsilon^p} = \varepsilon_k^p - \varepsilon_{k-1}^p - \Delta\gamma_k n_k$	flow rule
$h^X = X_k - X_{k-1} - \Delta\gamma_k \left[\frac{2}{3} C n_k - \gamma_k X_k \right]$	back-stress evolution
$h^p = p_k - p_{k-1} - \Delta\gamma_k$	accumulation
$h^f = \sigma_{eq}(\xi_k) - \sigma_y(p_k; \theta)$	yield consistency

$\xi_k = \sigma_k - X_k$, $n_k = \frac{3}{2} \text{dev } \xi_k / \sigma_{eq}(\xi_k)$. h_θ is non-zero only in the yield row h^f (isotropic parameters). $h_\varepsilon = -\mathbb{C}$ in row h^σ . $h_{q_{k-1}} = -I$ in rows h^{ε^p} , h^X , h^p .

r_v — local stress change $\rightarrow$ nodal force



$$r_v = \mathbb{A}_e \sum_g B_g^T (\partial\sigma/\partial v)_g V_g$$

Figure 4: The local residual and the map from a local stress change to nodal force.

4 The incremental finite-element problem

The punch is pushed in small depth steps $d_1, \dots, d_N$. Each step (increment k) is a full nonlinear solve; once it converges the plastic history is stored (committed) and becomes the start of the next step, and the punch load L_k is read off.

k	increment (load step)
U, λ	nodal displacements; contact pressure on the contact surface Γ_C
x_g	a Gauss point, where the material law is evaluated
$v = (\sigma, \varepsilon^p, X, p)$	local unknowns: stress, plastic strain, back-stress, accumulated plastic strain
$q = (\varepsilon^p, X, p)$	history: the part of v stored and carried to the next increment
$\theta = (\sigma_0, Q, \beta)$	hardening parameters
h, r	local residual (material law) and global residual (force balance)
subscripts	partial derivatives, e.g. $h_v = \partial h / \partial v$, $r_U = \partial r / \partial U$
B, K_k	strain-displacement matrix; tangent stiffness of increment k
L_k	punch load at increment k

Reading Fig. 5. Down the left: impose d_k from the stored history; update which nodes touch the ball; compute strain at every Gauss point; try an elastic step, and if the trial stress lies outside the yield surface pull it back with the return map; differentiate that converged update to get the consistent tangent; assemble r_k and K_k , take a Newton step, and repeat until the residual is small; then commit and read the load. The right side states the same in equations.

Per increment three residuals are coupled (Fig. 5): the local constitutive residual h_k at every Gauss point, global equilibrium r_k , and the contact conditions. Newton solves the linearized saddle

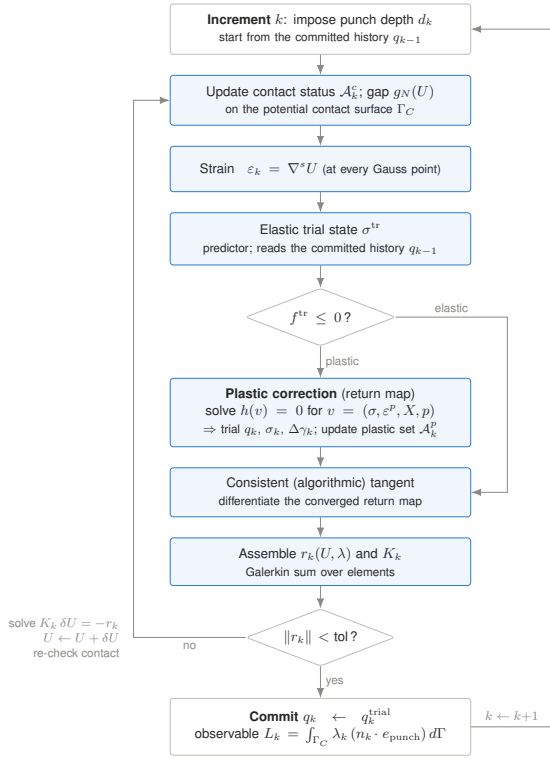
problem; condensing the local solve into the global one gives the consistent tangent

$$K_k = r_U - r_v h_v^{-1} h_U, \quad (9)$$

the material’s response to nodal motion. Because K_k is exact, Newton converges quadratically once the contact set settles.

The teaching model is an axisymmetric clamped plate (radius 2 mm, thickness 0.5 mm) pressed by a rigid ball (radius 1.25 mm). It uses 4-node quads with 2×2 Gauss points, small strain, and frictionless penalty contact, so the saddle reduces to $K_k \delta U = -r_k$ with the penalty stiffness inside K_k . The pieces map one-to-one onto JAX (Fig. 7): the internal force is the transpose of the strain map (`jax.vjp`), the tangent is `jax.jacfwd` of the residual, and the global Newton also sits in `custom_root`, so $\partial U / \partial(\cdot) = -K^{-1} \partial R / \partial(\cdot)$ is applied directly.

Six building blocks (Figs. 4 and 6). Everything in the next section is assembled from six small maps, each drawn as a pipeline from input to output. h_v is the local block: the five equations a Gauss point satisfies in one step. The parameters appear only in its yield row, the strain only in its stress row, and the stored history only in the three evolution rows. r_v turns a local stress change into nodal force ($B^\top$, Gauss weight, assembly), and ∇^s is its partner in the other direction, nodal displacement to Gauss strain. E_k answers “given extra forces, how do the nodes move?”: solve with K_k , then project with Π_k so contact is respected. C_k turns a strain change into a change of stored history through the same h_v^{-1} (S_q only selects the history entries). The readout keeps only the normal contact pressure on the patch and sums it to the single number δL_k .



Three coupled residuals per increment

$$\begin{aligned} h_k(v; q_{k-1}, \varepsilon(U)) &= 0 && \text{local constitutive (each Gauss point)} \\ r_k(U; \lambda; \sigma(v)) &= 0 && \text{global equilibrium} \\ g_N \geq 0, \quad \lambda_N \geq 0, \quad \lambda_N g_N &= 0 && \text{contact (KKT)} \end{aligned}$$

Newton solves the linearized saddle problem

$$\begin{bmatrix} K_k & -B_{1,k}^\top \\ B_{2,k} & 0 \end{bmatrix} \begin{bmatrix} \delta U \\ \delta \lambda \end{bmatrix} = - \begin{bmatrix} r_k \\ c_k \end{bmatrix}$$

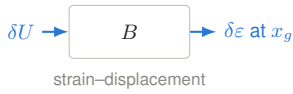
The local solve is condensed into the tangent

$$K_k = r_U - r_v h_v^{-1} h_U \quad \text{the return map moves into the operator: the material's response to nodal motion}$$

K_k is the elastoplastic consistent tangent, not a frozen elastic one; its factors are kept at convergence and reused for every sensitivity. In the code, contact is a smooth penalty, so the saddle reduces to $K_k \delta U = -r_k$ with the penalty stiffness inside K_k .

Figure 5: The forward solve for one increment, with the residuals it balances and the condensed tangent.

∇^s — nodal displacement $\rightarrow$ Gauss strain



the transpose partner of $B_g^\top$ inside r_v : force out, strain back in. Together they are the only route between a Gauss point and the mesh.

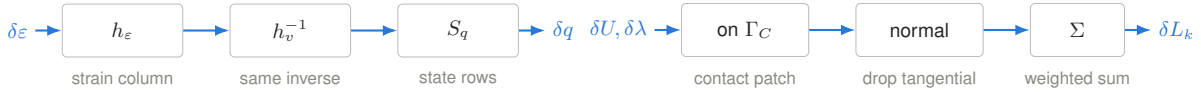
E_k — the constrained solve



$$\begin{aligned} K_k &= r_U - r_v h_v^{-1} h_U \\ \Pi_k &= B_{1,k}^\top (B_{2,k} K_k^{-1} B_{1,k}^\top)^{-1} B_{2,k} K_k^{-1} \end{aligned}$$

$-r_v h_v^{-1} h_U$ is the material's response to nodal motion. Π_k enforces non-penetration; K_k^{-1} alone would let the punch pass through the plate.

C_k, S_q — back into the state



$$C_k = -S_q h_v^{-1} h_\varepsilon$$

the same local inverse, fed by the strain column; S_q is a 0/1 selector picking (ε^p, X, p) out of v .

R_{obs} — field $\rightarrow$ one number

$$\text{frictionless: } \delta L_k = e_\Gamma^\top \delta \lambda_k$$

the whole contact field is summed to the punch load: one number per increment.

Figure 6: Kinematics, the constrained solve, the state update and the readout.

step	JAX tool
Every Gauss point	<code>jax.vmap</code>
Plastic multiplier $\Delta\gamma$	<code>jax.lax.custom_root</code>
Consistent tangent K	<code>jax.jacfd(residual)</code>
Internal force $B^T\sigma$	<code>jax.vjp(strain)</code>
Global Newton	<code>jax.lax.custom_root</code>
Load increments	<code>jax.lax.scan</code>
$dF/d\theta$, one pass per parameter	<code>jax.jacfd</code>
$dF/d\theta$, one pass per output	<code>jax.jacrev</code>

Figure 7: Which JAX tool does which job.

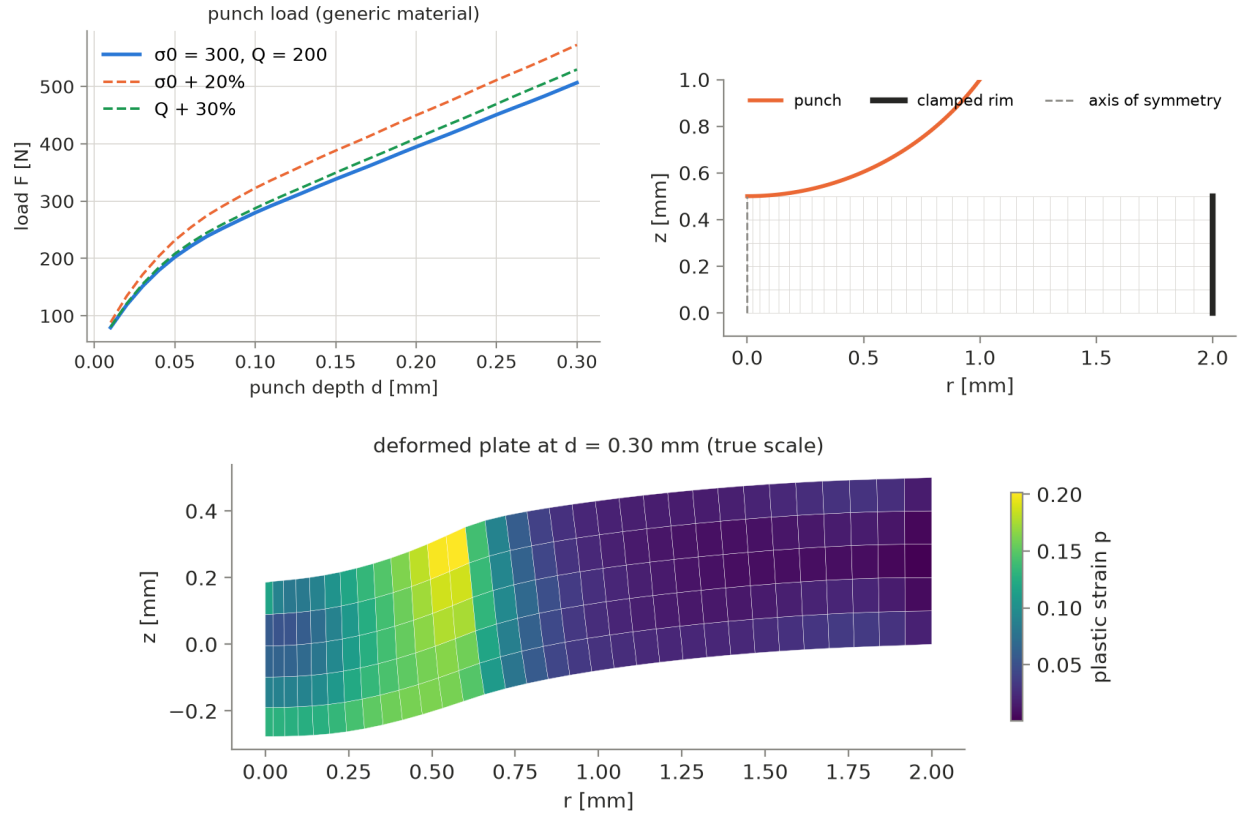


Figure 8: Load–depth curves (with two parameter changes), the mesh, and accumulated plastic strain at the last increment.

5 Linearizing the forward model

The question: if σ_0 , Q or β is nudged by $\delta\theta$, how much does the load at each depth change? The answer is $J = \partial L / \partial \theta$, one row per depth and one column per parameter. No new solver is needed; the building blocks above are reused in a fixed order.

A parameter change enters in one place: the yield row, evaluated at the plastic strain reached at each plastic Gauss point,

$$\left. \frac{\partial h^f}{\partial \theta} \right|_g = -\frac{\partial \sigma_y}{\partial \theta}(p_k(x_g); \theta). \quad (10)$$

From there it is *resisted* by the local block h_v^{-1} , *redistributed* by equilibrium under contact (the same factorized K_k as the forward Newton, one extra right-hand side), *carried* to later increments through the committed state, and *read out* as a change in load (Fig. 9). With $s_k = \partial q_k / \partial \theta$, the pseudoload and the state recursion are

$$f_{H,k} = r_v h_v^{-1} (h_{q_{k-1}} s_{k-1} + h_\theta) + f_{\Gamma,k}, \quad (11)$$

$$s_k = -S_q h_v^{-1} (h_\varepsilon B U_{\theta,k} + h_{q_{k-1}} s_{k-1} + h_\theta). \quad (12)$$

Writing $s_k = \Phi_k s_{k-1} + \Upsilon_k \delta\theta$ and unrolling (Fig. 10) shows that every increment injects, and the number of carry steps on a path is how old that injection is. Assembled, the Jacobian factors as

$$J = \mathcal{D} + \mathcal{H} \circ (\text{Id} - \mathcal{T})^{-1} \circ \mathcal{G}, \quad (\text{Id} - \mathcal{T})^{-1} = \sum_{n=0}^{N-1} \mathcal{T}^n, \quad (13)$$

a finite Neumann series because the history transfer $\mathcal{T}$ is strictly causal (Fig. 11). Inside each operator the same two paths appear, one ending in the state and one in the load (Fig. 12).

Reading the charts. In Fig. 9 follow the blue boxes: the parameter shifts the yield stress at each plastic point (elastic points feel nothing); together with the history change carried from earlier steps, $s_{k-1} = \partial q_{k-1} / \partial \theta$, this becomes an equivalent nodal force $f_{H,k}$ ($f_{\Gamma,k}$ is the matching contribution of the contact rows); that force is solved with the already factorized system; the result is split into the new history sensitivity s_k , which feeds the next step, and the load sensitivity. Fig. 10 is the same bookkeeping as a signal-flow diagram: Υ injects, Φ carries, $\mathcal{H}$ reads through the history and $\mathcal{D}$ reads directly. Unrolled, an injection made at step j reaches the load at step k only after $k-1-j$ carries, so each depth's load sensitivity sums the contributions of all earlier steps. Fig. 11 stacks the increments into operators, and since a step can only affect later ones, $(\text{Id} - \mathcal{T})^{-1}$ is an exact finite sum. Fig. 12 opens the four boxes: all are the same chain of building blocks fed either by the parameter source h_θ (blue: Υ_k , $\mathcal{D}_k$) or the history source $h_{q_{k-1}}$ (orange: Φ_k , $\mathcal{H}_k$). The commit path ends in the stored state, minus what re-equilibration of the plate gives back; the exit path ends in the load through the contact pressure. The dashed branch matters only with friction and is zero here.

Direct or adjoint. Direct evaluation fills J column by column (one extra solve per parameter per increment); adjoint fills it row by row (one backward sweep per output). With three parameters and thirty depths, direct is cheaper; with many parameters and a scalar objective, adjoint is. In JAX these are `jacfwd` and `jacrev`. Both were checked, together with finite differences and a hand-written direct recursion, against each other (Fig. 13).

Where the parameters enter: the yield row, at the strain reached

$$\left. \frac{\partial h^f}{\partial \theta} \right|_g = - \frac{\partial \sigma_y}{\partial \theta} (p_k(x_g); \theta)$$

The pseudoload, before condensation

$$f_{H,k} = r_v h_v^{-1} (h_{q_{k-1}} s_{k-1} + h_\theta) + f_{\Gamma,k}$$

$s_{k-1} = \partial q_{k-1} / \partial \theta$ is the history sensitivity carried from the previous increment; h_v^{-1} is the local constitutive block, visible before the Schur step.

Same factorized saddle, one extra right-hand side per parameter

$$\begin{bmatrix} K_k & -B_{1,k}^\top \\ B_{2,k} & 0 \end{bmatrix} \begin{bmatrix} U_{\theta,k} \\ \lambda_{\theta,k} \end{bmatrix} = \begin{bmatrix} f_{H,k} \\ 0 \end{bmatrix}$$

Carry the history, read out the load

$$s_k = -S_q h_v^{-1} (h_\varepsilon B U_{\theta,k} + h_{q_{k-1}} s_{k-1} + h_\theta), \quad \frac{\partial L_k}{\partial \theta} = e_\Gamma^\top \lambda_{\theta,k}$$

No new solver machinery: *resist* (h_v^{-1}), *redistribute* (equilibrium under contact), *carry* (state recursion), *read out*.

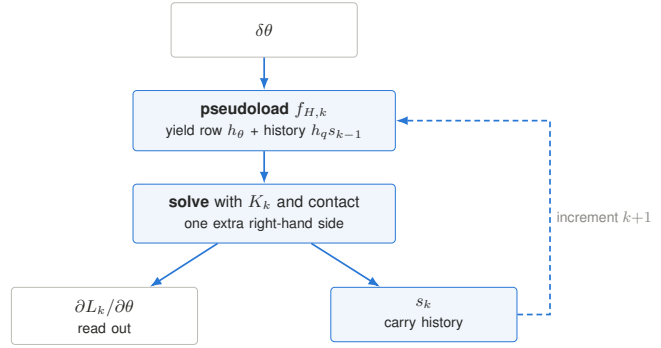
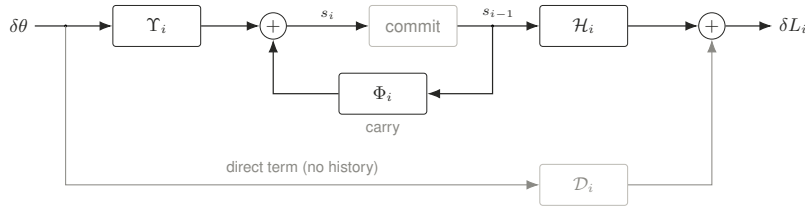


Figure 9: Sensitivity rides the same structure as the forward solve.

(a) one increment, the recursion $s_i = \Phi_i s_{i-1} + \Upsilon_i \delta \theta$



(b) unrolled $\delta L_k = \mathcal{D}_k \delta \theta + \sum_{j \leq k-1} \mathcal{H}_k \Phi(k-1, j) \Upsilon_j \delta \theta$ one term highlighted

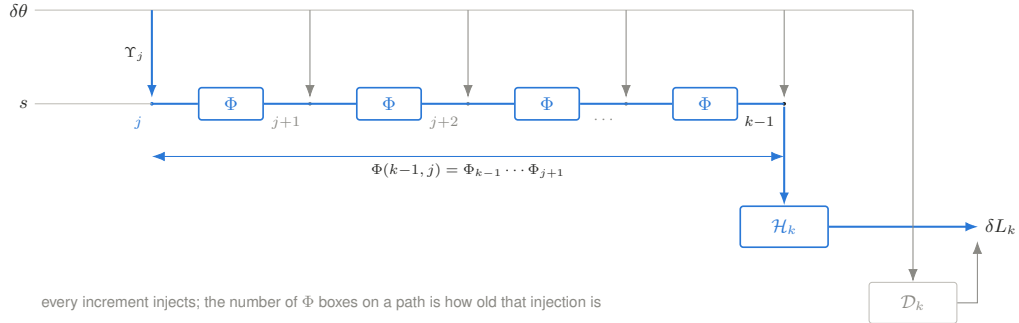


Figure 10: (a) The recursion for one increment. (b) Unrolled over increments, one injection highlighted.

Per increment (from the recursion chart)

$$s_k = \Phi_k s_{k-1} + \Upsilon_k \delta\theta, \quad \delta L_k = \mathcal{H}_k s_{k-1} + \mathcal{D}_k \delta\theta$$

Stack all increments: the assembled Jacobian, factor by factor

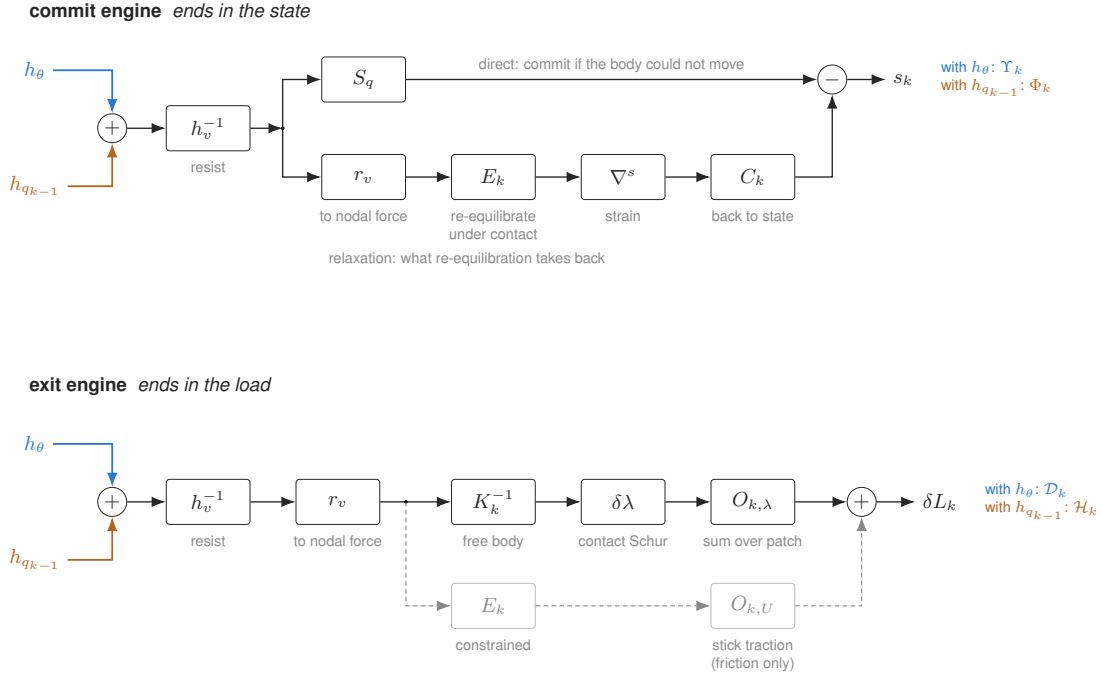
$$J = \underbrace{\mathcal{D}}_{\text{direct}} + \underbrace{\mathcal{H}}_{\text{observation}} \circ \underbrace{(\text{Id} - \mathcal{T})^{-1}}_{\text{history propagation}} \circ \underbrace{\mathcal{G}}_{\text{parameter injection}}$$

- $\mathcal{D}$ direct injection: θ enters the current increment's pseudoload
- $\mathcal{G}$ parameter injection: the yield-row source, one row per Gauss point
- $\mathcal{T}$ one-step history transfer: committed state carried to the next increment
- $\mathcal{H}$ observation: the contact readout compressing fields to the load

$\mathcal{T}$ is strictly causal (nilpotent), so the inverse is a finite sum: $(\text{Id} - \mathcal{T})^{-1} = \text{Id} + \mathcal{T} + \mathcal{T}^2 + \dots + \mathcal{T}^{N-1}$

increment j reaches every later readout only through the stored state: causality as a Neumann series

Figure 11: The assembled Jacobian, factor by factor.



The two sources add: with only h_θ the composite is Υ_k , with only h_{qk-1} it is Φ_k ; a real increment has both. The commit engine carries the minus and ends in the state; the exit engine carries the plus and ends in the load. Both are the same factorized saddle solve, read two ways.

Figure 12: Inside the operators: the state path and the load path, each fed by the parameter source and the history source.

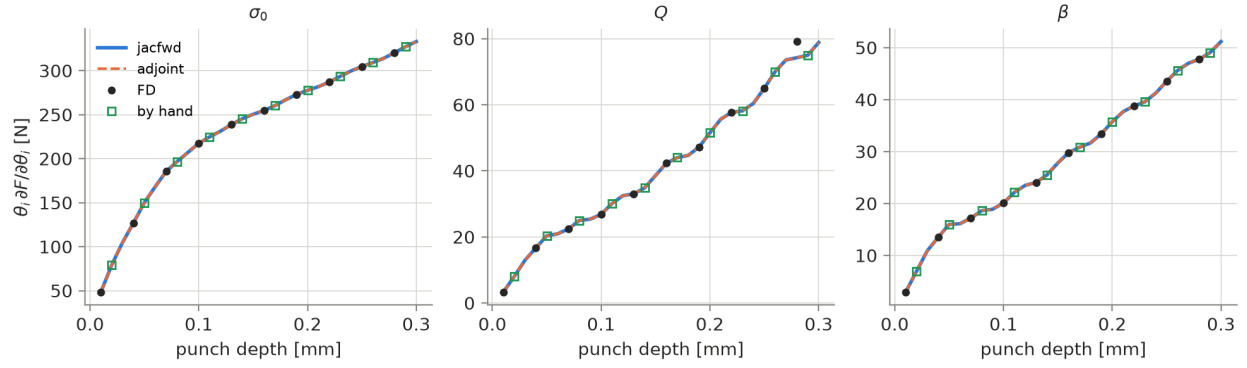


Figure 13: $\theta_i \partial L / \partial \theta_i$ for the punch: forward mode, adjoint, finite differences and the hand-written recursion agree.

6 A local Bayesian view with three parameters

Linearize at θ_0 : $\mathcal{G}(\theta_0 + \delta\theta) \approx \mathcal{G}(\theta_0) + J\delta\theta$, with noise $\epsilon \sim \mathcal{N}(0, \Sigma)$. Then

$$\mathcal{J}(\theta) = \frac{1}{2}\|y - \mathcal{G}(\theta)\|_{\Sigma^{-1}}^2 + \frac{1}{2}\|\theta - \theta_{\text{pr}}\|_{Q_{\text{pr}}}^2, \quad Q_{\text{post}} \approx Q_{\text{pr}} + \Gamma, \quad \Gamma = J^\top \Sigma^{-1} J. \quad (14)$$

Γ (the Fisher information, or Gauss–Newton Hessian) carries everything the experiment contributes locally. Its generalized eigenpairs $\Gamma\psi_i = \lambda_i Q_{\text{pr}}\psi_i$ give the directions the data inform: along mode i the posterior variance is the prior variance times $1/(1+\lambda_i)$, so modes with $\lambda_i > 1$ have their variance at least halved. Comparing Γ for tension and for the punch (Fig. 14) shows which combinations of (σ_0, Q, β) each test pins down. All of this is local: one anchor, one noise model, one three-parameter chart.

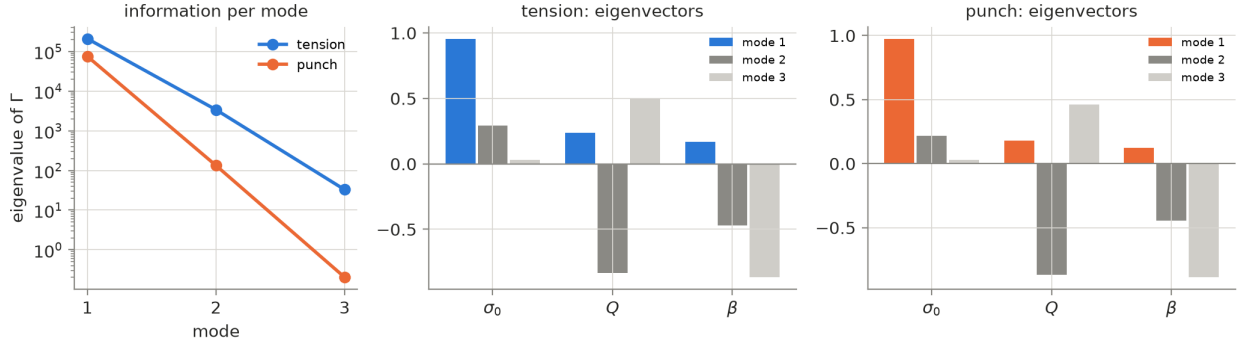


Figure 14: Eigenvalues and eigenvectors of Γ in log-parameters, 1% noise relative to each test's peak.

7 Where this leads

The same machinery can be asked a sharper question: not “how does the load change with three constants” but “where in plastic-strain space, at each depth, is the load sensitive to the hardening law at all”. Figure 15 is a first look (work in progress).

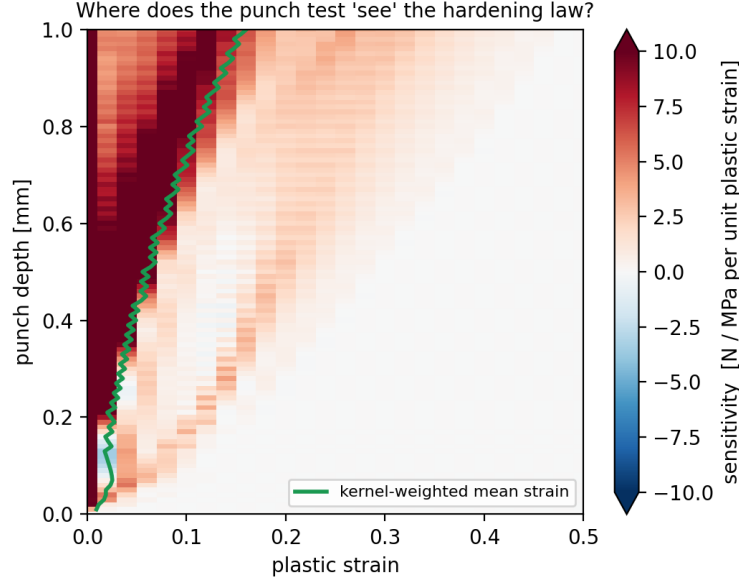


Figure 15: Work in progress: sensitivity of the punch load to the hardening law, resolved in plastic strain, with the kernel-weighted mean strain.

References

- J. C. Simo and T. J. R. Hughes, *Computational Inelasticity*, Springer (1998).
- E. A. de Souza Neto, D. Perić and D. R. J. Owen, *Computational Methods for Plasticity*, Wiley (2008).
- J.-L. Chaboche, A review of some plasticity and viscoplasticity constitutive theories, *Int. J. Plasticity* 24 (2008) 1642–1693.
- P. J. Armstrong and C. O. Frederick, A mathematical representation of the multiaxial Bauschinger effect (1966); reprinted in *Materials at High Temperatures* 24 (2007) 1–26.
- P. Wriggers, *Computational Contact Mechanics*, 2nd ed., Springer (2006).
- M. Kleiber, H. Antúnez, T. D. Hien and P. Kowalczyk, *Parameter Sensitivity in Nonlinear Mechanics*, Wiley (1997).
- P. Michaleris, D. A. Tortorelli and C. A. Vidal, Tangent operators and design sensitivity formulations for transient non-linear coupled problems, *Int. J. Numer. Meth. Eng.* 37 (1994) 2471–2499.
- M. Blondel et al., Efficient and modular implicit differentiation, *NeurIPS* (2022).
- J. Bradbury et al., JAX: composable transformations of Python+NumPy programs (2018), <https://github.com/google/jax>.
- A. Spantini et al., Optimal low-rank approximations of Bayesian linear inverse problems, *SIAM J. Sci. Comput.* 37 (2015) A2451–A2487.