Understanding Crystal Plasticity

A Physics-Based Approach to Computational Mechanics

Santiago Garcia Botero

University of Texas at San Antonio

February 10, 2026

What is Crystal Plasticity?

When you bend a paperclip, it stays bent. That permanent deformation is called **plasticity**.

- **Elasticity:** Stretching bonds (reversible, like a spring).
- **Plasticity:** Moving dislocations (irreversible, like sliding cards).

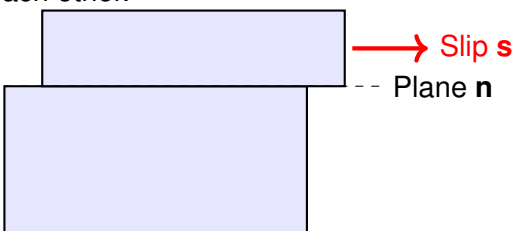
Why "Crystal" Plasticity? Metals are crystals (ordered atoms). Usually, deformation doesn't happen everywhere at once. It happens on specific planes in specific directions.

The Mechanism: Slip Systems

Imagine a deck of cards. It's easy to slide cards past each other, but hard to pull them apart. In metals, blocks of atoms "slip" past each other.

Slip System (α):

- **Slip Plane ($\mathbf{n}^{\alpha}$):** The plane with the highest atomic density (smoothest ride).
- **Slip Direction ($\mathbf{s}^{\alpha}$):** The direction with the closest packed atoms.



For FCC Metals (Aluminum, Copper), there are **12 Slip Systems**. This code simulates all 12 of them simultaneously.

Driving Force: Resolved Shear Stress

We apply a stress tensor $\boldsymbol{\sigma}$ (force per area) to the whole crystal. But only the shear component *on the slip plane* causes slip.

Schmid's Law:

$$\tau^{\alpha} = \boldsymbol{\sigma} : (\mathbf{s}^{\alpha} \otimes \mathbf{n}^{\alpha})$$

Where:

- $\mathbf{s}^{\alpha}$: Slip direction vector (direction of movement).
- $\mathbf{n}^{\alpha}$: Slip plane normal vector (perpendicular to the plane).
- $\mathbf{s}^{\alpha} \otimes \mathbf{n}^{\alpha}$: Schmid Tensor (Geometric projection factor).
- τ^{α}: Resolved Shear Stress (Scalar driving force).

Implications of Schmid's Law

Because it is a geometric projection:

- If you pull the crystal perpendicular to the slip direction, $\tau^{\alpha} = 0$ (No slip!).
- The stress state must align with the crystal lattice to cause deformation.

In this project, τ^{α} is calculated in the driver and passed as an **input** to our flow rule subroutine.

What Stops Dislocations?

Dislocations don't move freely. They hit obstacles. To move, they need energy to overcome these barriers.

1. **Long Radius Barriers (Athermal):** Large precipitates, grain boundaries. Thermal energy (vibration) isn't enough to help. You just need pure force (stress).
2. **Short Radius Barriers (Thermal):** Other dislocations, atomic lattice friction. Thermal vibrations **help** you jump over these.

Effective Stress Definition

Effective Stress (τ_{eff}): The stress actually available to do work against thermal barriers.

$$\tau_{drive} = \tau^{\alpha} - \tau_{back} \quad \text{(Backstress resists motion)}$$

Mathematical Decomposition:

$$\tau_{eff} = \langle |\tau_{drive}| - S_{athermal} \rangle$$

Where:

- $S_{athermal}$: Long-range resistance (cannot be jumped by heat).
- $\langle x \rangle$: Macaulay brackets (returns 0 if $x < 0$).

If Driving Stress ¡ Athermal Resistance, nothing happens ($\dot{\gamma} = 0$).

The Kocks-Argon-Ashby Flow Rule

We model plastic flow as a **Thermally Activated Process**. Dislocations must wait for a random thermal fluctuation to help them jump over an obstacle.
The probability of a jump is given by Boltzmann statistics:

$$P \propto \exp\left(-\frac{\Delta G}{k_B T}\right)$$

Thus, the slip rate $\dot{\gamma}$ is proportional to this probability.

The Flow Rate Equation

When we expand the energy term ΔG, we get the full flow rule:

$$\dot{\gamma}^{\alpha} = \dot{\gamma}_{ref} \exp\left[-\frac{\Delta G(\tau_{eff})}{k_B T}\right] \operatorname{sgn}(\tau)$$

Where:

- $\dot{\gamma}_{ref}$: Reference slip rate (attempt frequency).
- k_B: Boltzmann constant.
- T: Temperature (Kelvin).
- τ_{eff}: Effective stress driving the motion.

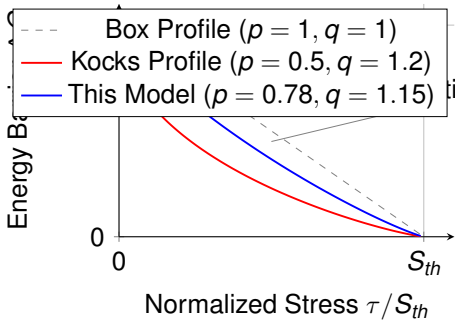
This equation bridges the gap between atomic vibrations (nanoseconds) and macroscopic deformation (seconds).

Energy Barrier Profile

The activation energy ΔG is not constant. It is reduced by the applied stress. We use a phenomenological shape function for the obstacle profile:

$$\Delta G(\tau_{eff}) = \Delta G_0 \left[1 - \left(\frac{\tau_{eff}}{S_{thermal}}\right)^p\right]^q$$

- ΔG_0: Total energy required at zero stress (Helmholtz free energy).
- p, q: Shape parameters defining the "width" and "tail" of the obstacle.
- As $\tau_{eff} \to S_{thermal}$, the barrier vanishes ($\Delta G \to 0$).



Physics of Hardening: Why Metal Gets Stronger

As metal deforms, dislocation density ρ increases (from 10^{10} to $10^{15} m^{-2}$). Dislocations get tangled and impede each other.

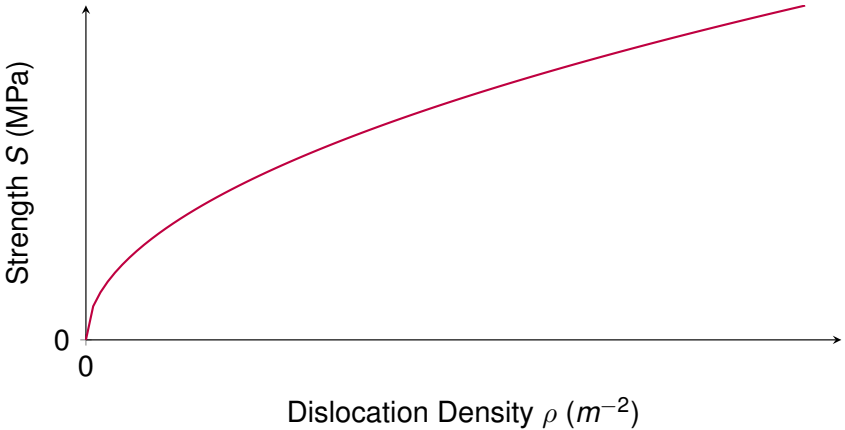
Taylor Hardening Law:

$$S \propto \mu b \sqrt{\rho}$$

Where:

- S: Slip Resistance (Current strength of the material).
- μ: Shear modulus (Spring constant of the lattice).
- b: Burgers vector (Magnitude of the lattice distortion).
- ρ: Dislocation density (Line length per unit volume).

Visualizing Hardening



In our model, we separate ρ into ρ_{SSD} (Statistically Stored) and ρ_{GND} (Geometrically Necessary).

Mathematical Decomposition of Resistance

We assume different obstacles act in specific ways.

1. Athermal Resistance ($S_{athermal}$): Long-range interactions. Add as squares (assume random distribution).

$$S_{athermal} = \sqrt{(S_{SSD,a})^2 + (S_{GND,a})^2} + S_{Ind,a}$$

2. Thermal Resistance ($S_{thermal}$): Short-range obstacles (e.g., forest dislocations). Thermal E can bypass these.

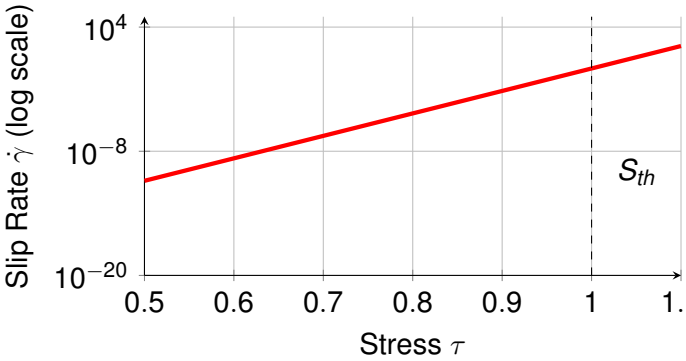
$$S_{thermal} = \sqrt{(S_{SSD,th})^2 + (S_{GND,th})^2 + (S_{cross,th})^2} + S_{Ind,th}$$

Key Insight: We calculate the effective stress $\tau_{eff} = \langle |\tau| - S_a \rangle$ first, removing the long-range part, before checking if we can jump the thermal part.

The Mathematical Challenge: Stiffness

The Arrhenius equation is extremely non-linear ("Stiff").

$$\text{Rate } \dot{\gamma} = \dot{\gamma}_{ref} \cdot \exp\left(-\frac{\Delta G_0}{kT}\left(1 - \frac{\tau}{S}\right)\right)$$



A small change in stress τ causes a **massive** change in rate $\dot{\gamma}$.
Explicit integration ($\Delta t \approx 10^{-9}$s) is too slow. We need Implicit Integration ($\Delta t \approx 1$s).

Deriving the Analytic Jacobian

We need to linearize the non-linear flow rule. We calculate $\frac{\partial \dot{\gamma}}{\partial \tau}$ using the Chain Rule. Let $R = \frac{\tau_{eff}}{S_{th}}$. Then $\dot{\gamma} \propto \exp(-A(1 - R^p)^q)$.

$$\begin{aligned}\frac{\partial \dot{\gamma}}{\partial \tau} &= \frac{\partial \dot{\gamma}}{\partial \Delta G} \cdot \frac{\partial \Delta G}{\partial R} \cdot \frac{\partial R}{\partial \tau} \\ &= \left(-\frac{\dot{\gamma}}{k_B T}\right) \cdot \left(\Delta G_0 \cdot q(1 - R^p)^{q-1} \cdot (-pR^{p-1})\right) \cdot \left(\frac{1}{S_{th}}\right)\end{aligned}$$

This looks messy, but it is **exact**. Implementing this analytic derivative prevents numerical errors and makes the simulation converge quadratically fast.

Code Structure

The simulation is implemented in modern Fortran (F90+).

- `driver.f90`: Sets up the crystal state, defines the stress tensor, and calls the constitutive updates.
- `computeFlowRule.f90`: The ”Physics Kernel”. It takes the stress and internal state inputs and returns the plastic slip rate $\dot{\gamma}$ and its derivatives.

Vectorized Design: The arrays are dimensioned `(12)`, corresponding to the 12 FCC slip systems. The code solves for all systems simultaneously in a single pass.

The Physics Kernel

Inside `computeFlowRule.f90`:

1. **Effective Stress Calculation:**

$$\tau_{eff} = |\tau_{resolved} - \tau_{back}| - S_{athermal}$$

2. **Conditional Activation:** The code checks `if (stress > resistance)`. Systems with insufficient stress are skipped immediately, saving CPU cycles.
3. **Arrhenius Update:** Active systems update their slip based on the thermal activation energy barrier we derived earlier.

Numerical Optimization: The Jacobian

Many codes use ”Finite Differences” to find the slope (perturbing the system twice). This is slow and error-prone.

Our Approach: Analytic Derivatives We programmed the exact derivative equations directly into the Fortran code:

$$\texttt{dIncSlip\_dStress} = \texttt{rate} \times \frac{\Delta G}{k_B T} \times \dots$$

This allows the global solver (Newton-Raphson) to converge in **3-4 iterations** instead of 20+, making the simulation extremely fast.

Summary

Physics-Based Constitutive Modeling:

- **Thermodynamics:** Deformation is treated as a thermally activated rate process.
- **Microstructure:** Hardening is linked to dislocation density evolution ($\sqrt{\rho}$).
- **Mathematics:** Separating resistances allows us to model temperature dependence correctly.
- **Numerics:** The analytic Jacobian ensures that our highly non-linear stiff equations can be solved efficiently.

This framework allows engineers to predict the strength of jet engine turbine blades and car chassis components from basic atomic principles.