1 ME 531 - WARP3D Presentation

https://answers.uillinois.edu/illinois.engineering/page.php?id=79993 – (Windows users) can download MobaXterm (recommended by Engineering IT) from the link provided in this webpage

Login to Engineering Workstation (EWS) from terminal – MobaXterm (windows) / Terminal (Mac)

Enter in the command line: ssh *<yournetid>*@linux.ews.illinois.edu

Enter your password.

Download and unzip Warp3D

Copy-paste following command (should take 5-10 seconds to download)

```
wget https://www.dropbox.com/s/9t94fqj1v2rdw0e/warp3d_distribution_18.3.2_4230.zip?dl=0 --output-document Warp3D.zip
```

Unzip downloaded zip file (should take a few minutes)

```
unzip Warp3D.zip
```

Rename directory to a more convenient filename:

```
mv warp3d_distribution_18.3.2_4230 Warp3D_ME531
```

Add the directories which contain relevant Warp3D executables to your PATH

```
export PATH=”$HOME/Warp3D_ME531/run_linux/:$PATH”
```

```
export LD_LIBRARY_PATH=”$HOME/Warp3D_ME531/linux_packages/lib/:$LD_LIBRARY_PATH”
```

```
export PATH=”$HOME/Warp3D_ME531/warp3d2exii/:$PATH”
```

Make a directory to store all example simulations:

```
mkdir Warp3D_Examples
```

Go to the directory:

```
cd Warp3D_Examples
```

Copy the following folders that contain the simulation input scripts directly from an online repository:

```
git clone https://github.com/Sameer531/CP_SingleElem
```

This should download the folder – CP_SingleElem that contains two input scripts – cp-tension.input and output.txt. Go into the folder:

cd CP_SingleElem

Copy a required library file from the installation to the current folder

cp ~/Warp3D_ME531/linux_packages/lib/libiomp5.so ./

Run the simulation:

warp3d_Intel.omp < cp-tension.input

It will generate output text files corresponding to displacements (wnd*_text), strain (wee*_text), stress (wes*_text) and crystal-plasticity state variables (wem*_text_crystal_plasticity)

To analyze the output, the data files must be converted to a file suitable for visualization using Paraview. Warp3D directory contains a python script that can be used for the conversion.

Load python into your environment (Mac users may not need the line below since Python may be pre-installed, in which case skip to the line after)

module load python3/3.10.0

python3 ~/Warp3D_ME531/warp3d2exii/warp3d2exii

```
=> Filename for Exodus II output (e.g., results): PViewOut.exo
=> Path to flat file for model description (e.g. model.text): single_e_model_flat.text
=> Path to directory with results files (e.g., ./): ./
=> Input time file with physical times for each step (y/n)? n
=> Write a large model Exodus II file [recommend no] (y/n)? y
```

This generates the file PViewOut.exo, which along with all the text outputs will help analyze the output data.

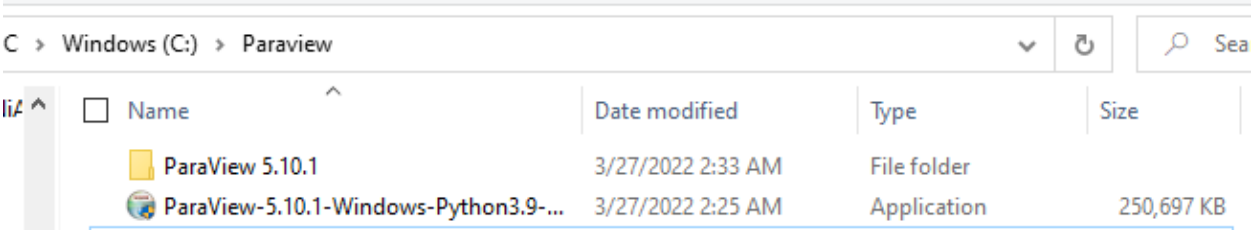
These files need to be downloaded to your system, through MobaXterm (Windows) or using FileZilla (Mac), and then visualized using Paraview.

In MobaXterm look for the  icon to the top-left of the terminal. Select the CP_SingleElem folder, and then click on this icon to download the folder with all the input files and the output data files.

Download and Install Paraview:

- Download the windows installer from https://www.paraview.org/download/ - ParaView-5.10.1-Windows-Python3.9-msvc2017-AMD64.exe
- Move the installer to any preferred folder and double-click – Follow the instructions to install Paraview on your system

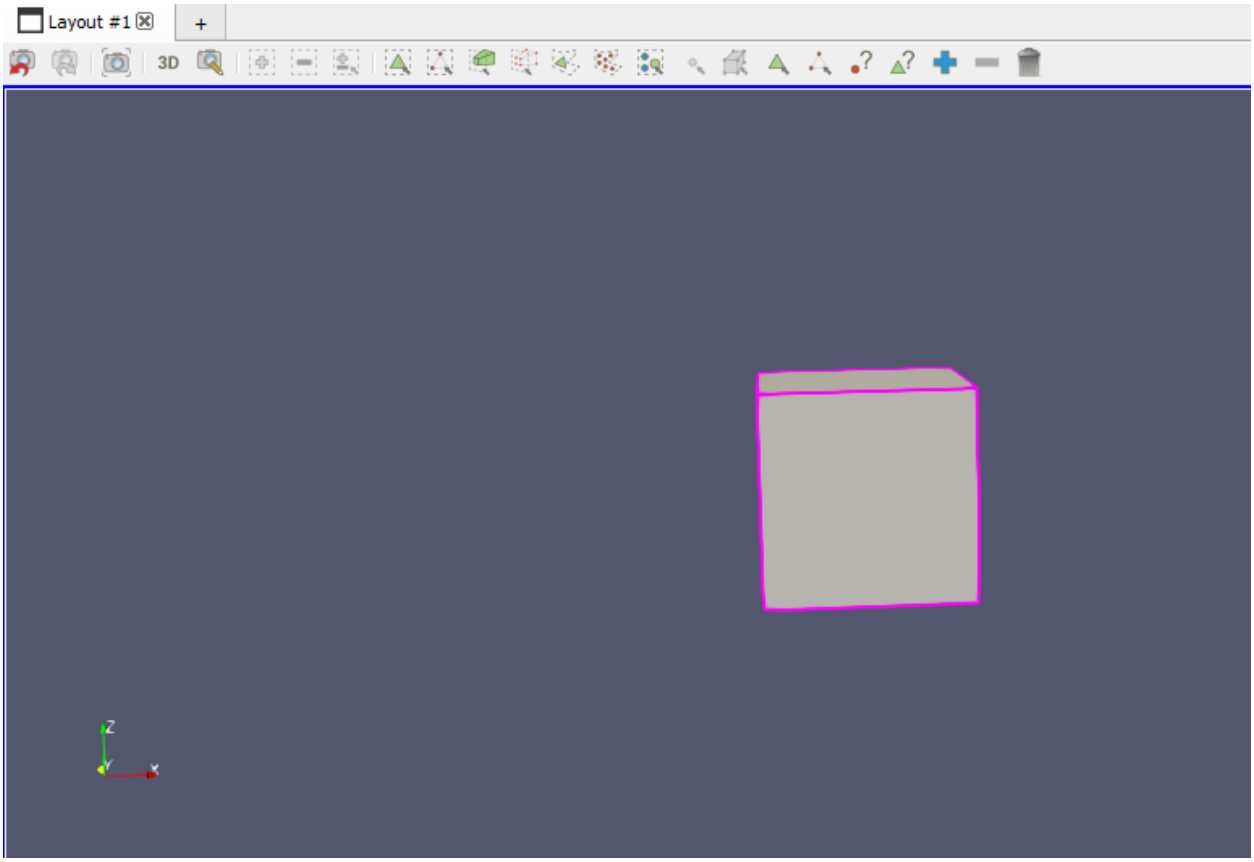
It looks like this for me:



- Paraview was installed in C:\Paraview\ParaView 5.10.1\ in my system
- To open Paraview, go to the folder \ParaView 5.10.1\bin\
- Scroll down to find the executable – paraview.exe
- Double-click to open

Open the exodus file in Paraview

- Once you open the file, you won't see anything yet
- In the Properties tab below, select the variables of interest (stresses, strains, displacements, mises strain, mises stress, etc.) – And then hit "Apply"



- Now you should be able to see the model
- You can play the entire deformation - 

To create a contour plot with a certain variable, in the Properties tab, scroll down to the “Coloring” option and select the appropriate variable. For instance, I select Strain_xz (which is the direction in which displacement/stress is applied).

Then I want to select an appropriate scale – For which you can select  on the top left

That should scale over the minimum and maximum values throughout all steps – Now you can play the deformation and see the increase in strain.

To extract the data, we need to use Filters in Paraview.

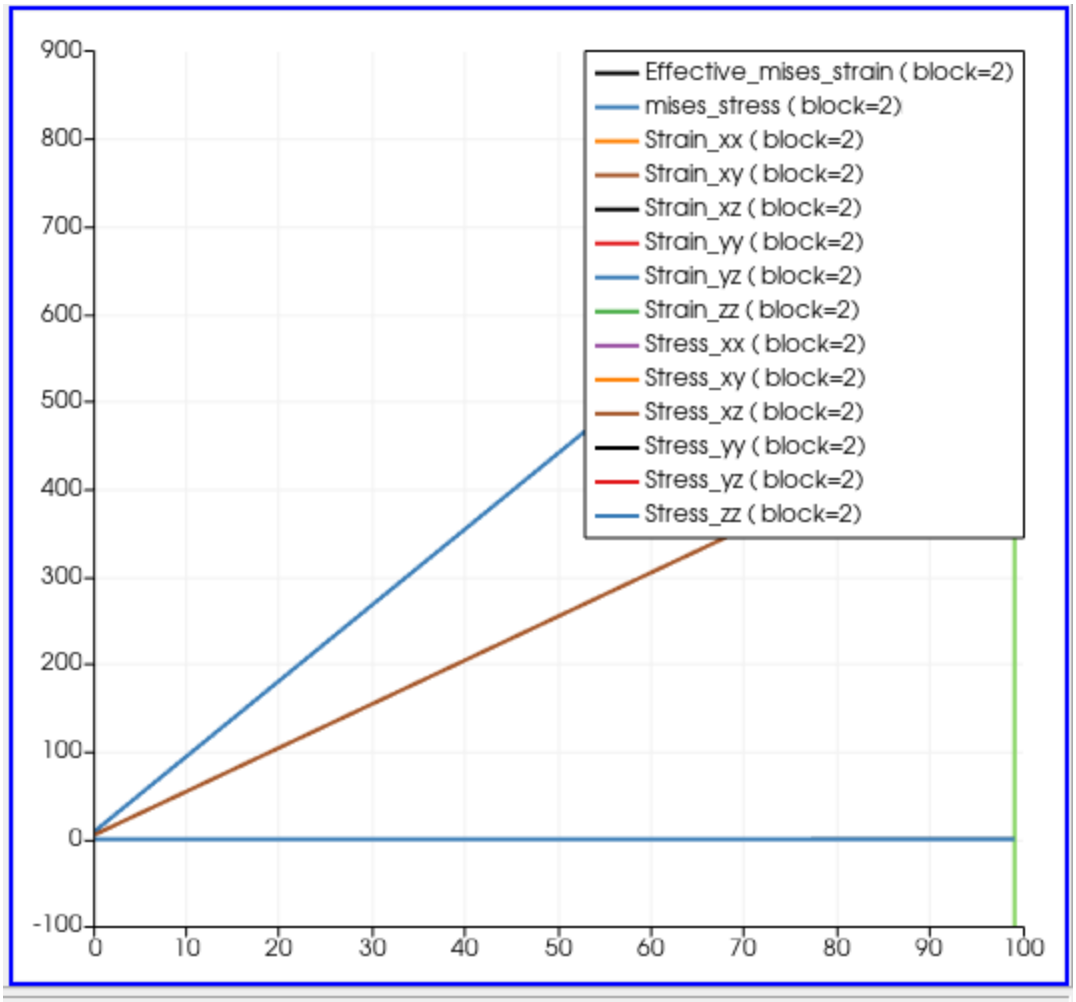
First you need to select the part of your model from which you would like to extract data. In this example, we only have **one element**, and we select the entire cube.

Click on  which is located on the tool bar directly above the viewing area (the viewing area is where you see the model). Then click+Drag to place a rectangular selection area on your model and select it.

Now on the Menu bar, go to Filters, and then click Search. Type “Plot selection over time” and you should see the option. Hit enter to select the filter

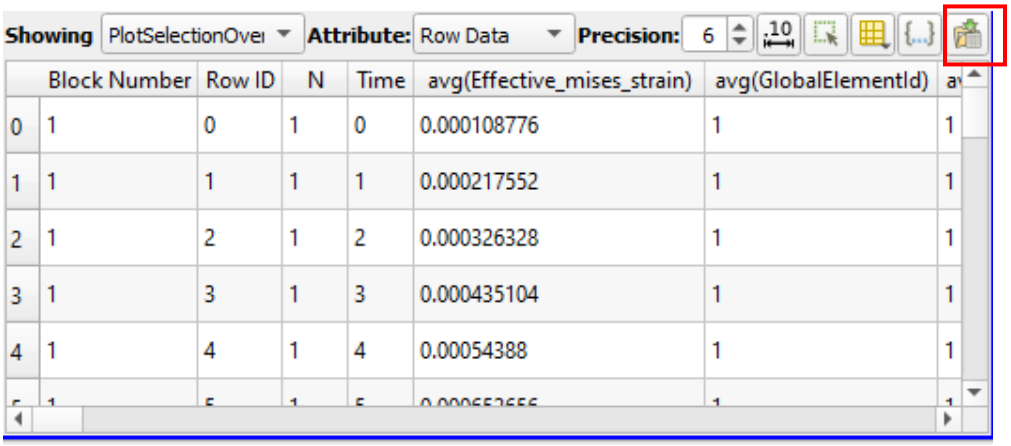
To make sure that this Filter is going to act on the model – Check the “Copied Selection” in the Properties bar. It must not be empty and must have node coordinates of the element.

Now hit Apply – All the data is plotted over time



You can see all this in a spreadsheet – Click on  above the plot and select Spreadsheet view.

Then you can select the **top rightmost icon** in this tile



	Block Number	Row ID	N	Time	avg(Effective_mises_strain)	avg(GlobalElementId)	a
0	1	0	1	0	0.000108776	1	1
1	1	1	1	1	0.000217552	1	1
2	1	2	1	2	0.000326328	1	1
3	1	3	1	3	0.000435104	1	1
4	1	4	1	4	0.00054388	1	1

That will extract all the data (plotted over time step) into a csv file which you can open with Excel.

Basics

- Crystal plasticity in WARP3D is just another material model (*cp*)
- Slightly more things to define:
  - One or more crystals
  - A material model combining:
    - Crystal definitions
    - Orientations
    - Various other material properties (density, thermal expansion coefficients)

Crystals

- Defining a crystal is basically the same as defining a regular set of material properties
  - Different keyword (*crystal*)
  - Need to use consecutive numbers to ID
- Then define a *cp* material
  - Can have multiple crystals/angle combinations per element – Taylor homogenization
  - Crystal definitions:
    - Single – just give a crystal number ($n_{crystals} = 1$)
    - File – different crystals in each element, next slide
  - Angle definitions:
    - Single – just give a set of Euler angles ($n_{crystals} = 1$)
    - File – different angles in each element

Crystal angle/file format

<el #> <cry#> <ang1> <ang2> <ang3>

repeat ncrystals times

<el #> <cry#> <ang1> <ang2> <ang3>

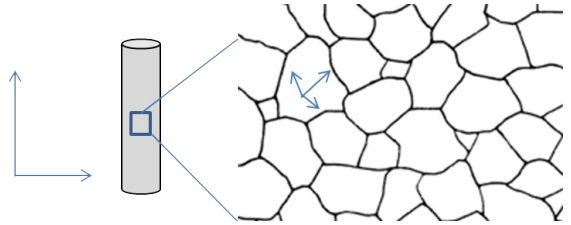
repeat ncrystals times

…

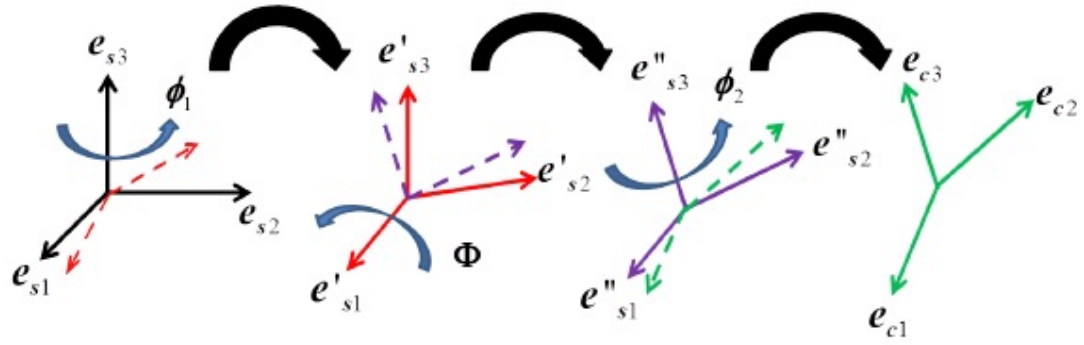
- Can omit <cry#> if single crystal or angles if single angle

Crystal orientation definition (in terms of grain angles)

warp3d



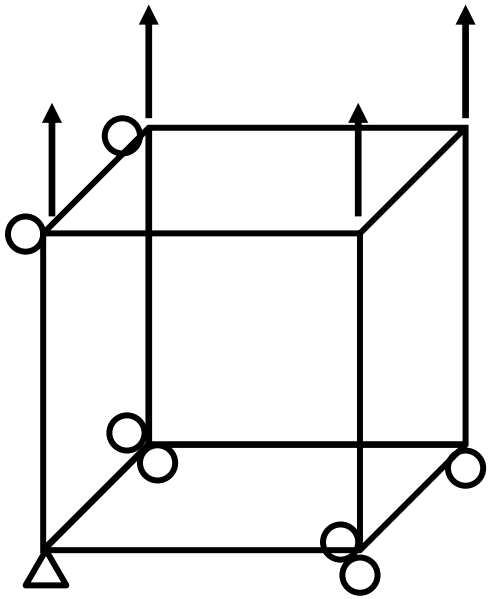
Kocks (symmetric)	Ψ	Θ	ϕ
Bunge	ϕ_1-π/2	Φ	π/2–ϕ_2



Courtesy: K Chatterjee

Actual example

- Single element
- Tension (displacement) loading



Properties

- Al-ish:

crystal 1

properties slip_type fcc elastic_type isotropic e 78811.2 nu 0.33,
mu_0 29628.3 D_0 0.0 t_0 204.0 harden_n 20 theta_0 180.0,
b 3.5E-7 boltz 1.3806E-20 tau_a 0.0 tau_hat_y 155.0 g_0_y 0.007808,
q_y 2.0 p_y 0.5 eps_dot_0_y 1.0E13 tau_hat_v 25.0 g_0_v 0.00488,
q_v 2.0 p_v 0.5 eps_dot_0_v 1.0E7 hardening geometric k_0 0.0

- Things I didn't mention:
  - mu_0, D_0, t_0 – temperature dependent shear modulus
  - t_0 doubles as "default temperature"

Properties

material al_cp

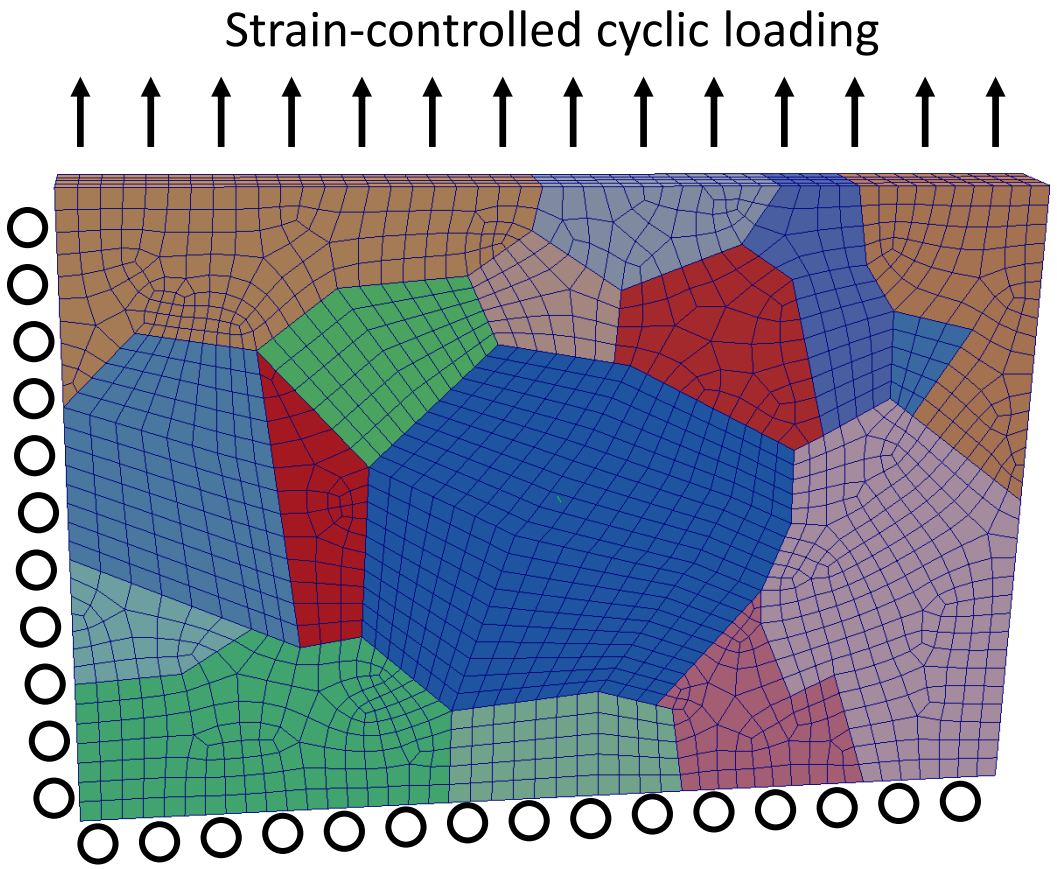
properties cp alpha 1.08E-5 rho 0.0 angle_convention kocks,

n_crystals 1 angle_type degrees crystal_input single crystal_type 1,
orientation_input single angles 55.0 45.0 0.0 debug off tolerance 1.0E-10

- n_crystals – number of crystals per element
- crystal_input single crystal_type 1 – all use crystal 1
- orientation_input single angles 55.0 45.0 0.0 – manually specify single orientation

More complicated/realistic example

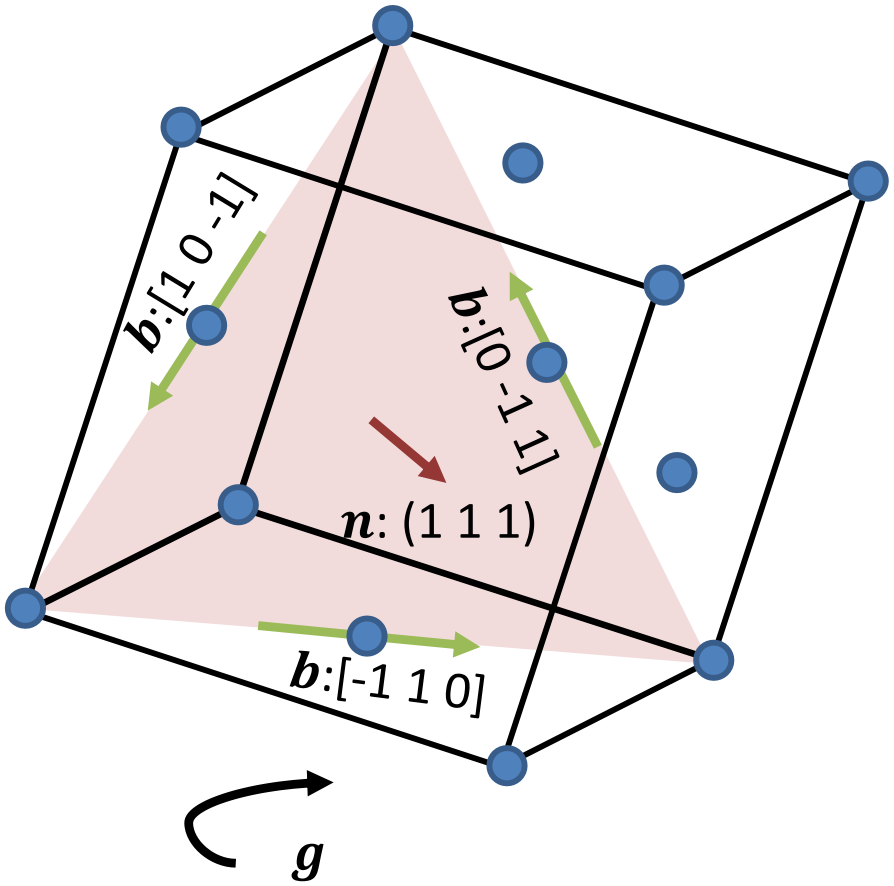
- Small grain cluster
  - 16 grains
  - ~6000 finite elements
- Same material properties
- Each element has 1 crystal, uses the same crystal definition
- But angle input from file
- Cyclic (tension/compression) loading



Overview

- WARP3D crystal plasticity
  - Based on Asaro/Needleman flow rule
  - MTS hardening
  - (Geometric hardening)
  - Large strains via the Green-Naghdi objective rate
  - Large plastic rotations with an exponential integrator
- You do not have the current version!
  - Symmetrized tangent (poor convergence)
  - Only MTS hardening available
  - In general, less stable than current branch

Slip system orientation



- For small strains -- slip system directions combination of:
  - Crystal system type (FCC, BCC, HCP, …)
  - Grain orientation
- Ex: $\boldsymbol{n}^{(s)} = \boldsymbol{g}^T \boldsymbol{n}_{FCC}^{(i)}$
  - $\boldsymbol{n}_{FCC}^{(i)}$ one of the 12 FCC system normals
  - $\boldsymbol{g}$ is the rotation from the lab coordinates to the crystal lattice orientation

Power law resolved shear/slip relation

- The usual assumption is the *viscoplastic* equivalent of the *Schmid law*:

$$\dot{\gamma}^{(s)} = \frac{\dot{\gamma}_0}{\tilde{\tau}} \left| \frac{\tau^{(s)}}{\tilde{\tau}} \right|^{n-1} \tau^{(s)}$$

or:

$$\Delta\gamma^{(s)} = \frac{\Delta\gamma_0}{\tilde{\tau}_{n+1}} \left| \frac{\tau_{n+1}^{(s)}}{\tilde{\tau}_{n+1}} \right|^{n-1} \tau_{n+1}^{(s)}$$

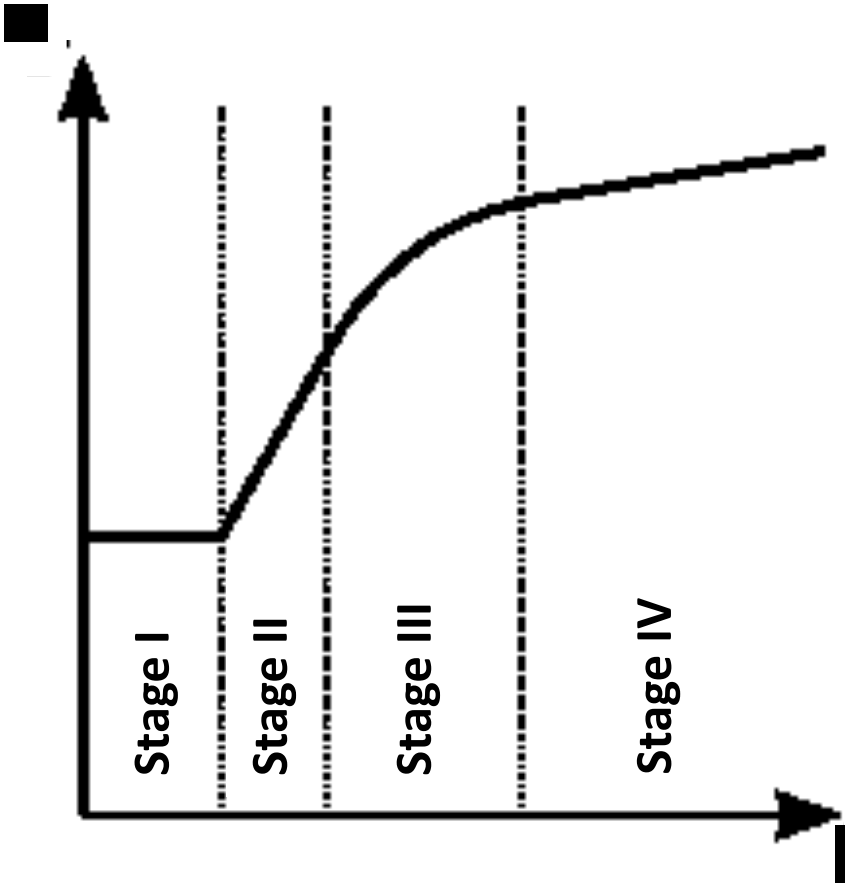
- $\Delta\gamma_0$ is a reference strain rate (we use $\sqrt{(3/2)\Delta\,\boldsymbol{\varepsilon}_{n+1} : \Delta\,\boldsymbol{\varepsilon}_{n+1}}$)
- $\tau^{(s)}$ is the resolved shear stress on system s: $\tau_{n+1}^{(s)} = \boldsymbol{\sigma}_{n+1} : \mathrm{sym}\left(\boldsymbol{b}^{(s)} \otimes\right.$

Crystal plasticity hardening models

- In WARP3D we use isotropic hardening models – all slip systems at a material point have the same strength
- The value of the slip system strength is (only) state variable of the model
- Lots of different choices for hardening model

Mechanical Threshold Stress

- MTS – the slip system strength at 0K (and zero strain rate)
- Scale to rate/temperature with Arrhenius-type scale factors
- Work hardening: Voce Law (Stages II and III)



Kok, Beaudoin, and Tortorelli. *Int. J. Plast.* 18 (2002).
Kocks, Argon, and Ashby. *Prog. Mater. Sci.* 19 (1975).

The equations

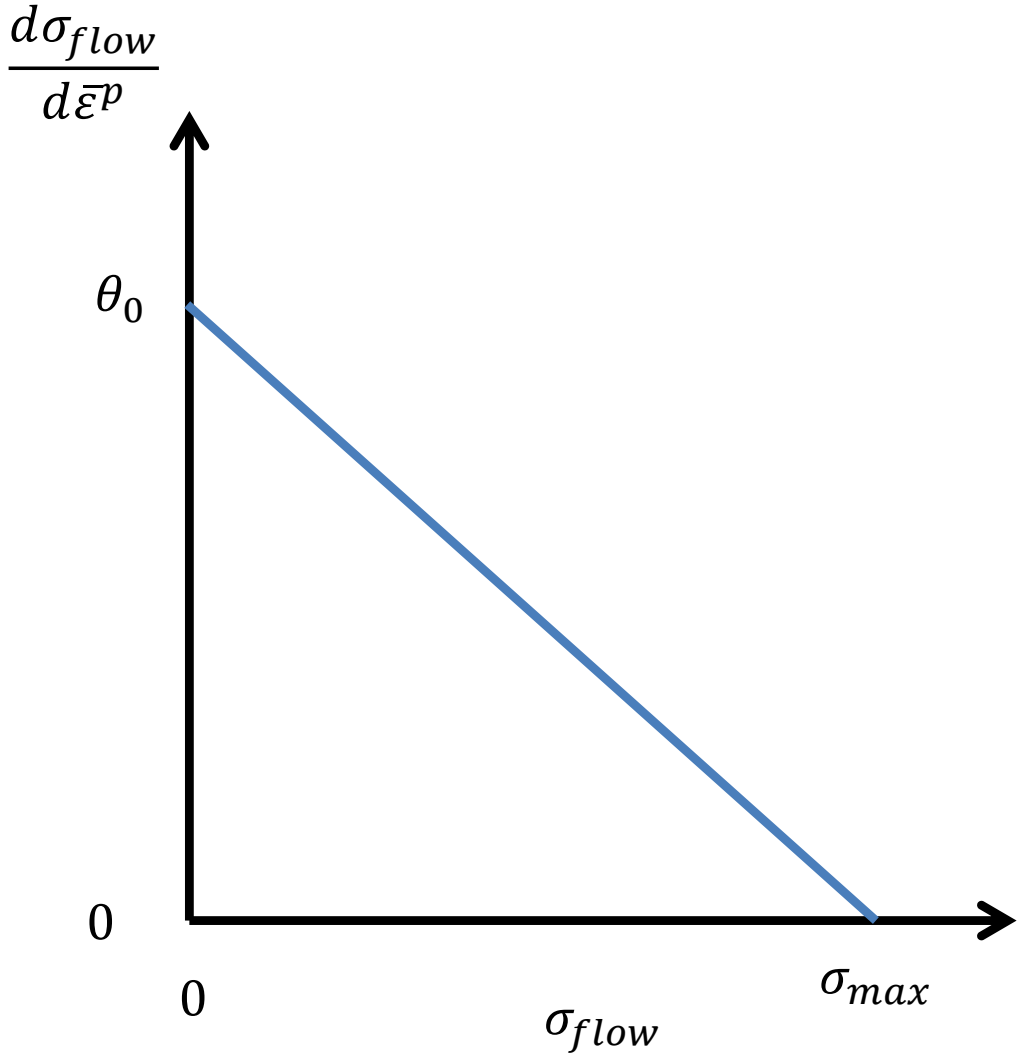
- $\tilde{\tau} = \tau_a + \hat{\tau}_y S_y(T, \dot{\varepsilon}) + \tau_w$
  - τ_a -- any athermal contribution
  - $\hat{\tau}_y$ -- MTS for intrinsic barriers to dislocation motion – inclusions, 2nd phases, etc.
  - $S_y(T, \dot{\varepsilon})$ -- MTS scale factor:

$$S_y = \left\{1 - \left[\frac{kT}{\mu b^3 g_0} \log\left(\frac{\dot{\varepsilon}_0}{\dot{\varepsilon}}\right)\right]^{1/q}\right\}^{1/p}$$

  - τ_w -- Backstress due to other (static) dislocations
    - Could scale by $S_v(T, \dot{\varepsilon})$, but we don't
    - Get from a Voce law

Voce law

- Model of stage III hardening
- $\dot{\tau}_w = \theta_0 \left(1 - \frac{\tau_w}{\tau_v}\right) \sum_{s=1}^{n_{slip}} \left|\dot{\gamma}^{(s)}\right|$
  - Again, use an implicit integration
  - Think of $\sum_{s=1}^{n_{slip}} \left|\dot{\gamma}^{(s)}\right|$ as the rate of plastic strain
  - $\tau_v = \widehat{\tau}_v S_v \left(T, \dot{\varepsilon}\right)$ -- MTS
- $\frac{d\sigma_{flow}}{d\bar{\varepsilon}^p} = \theta_0 \left(1 - \frac{\sigma_{flow}}{\sigma_{max}}\right)$
- θ_0 is the stage II slope



Summary: CP material properties

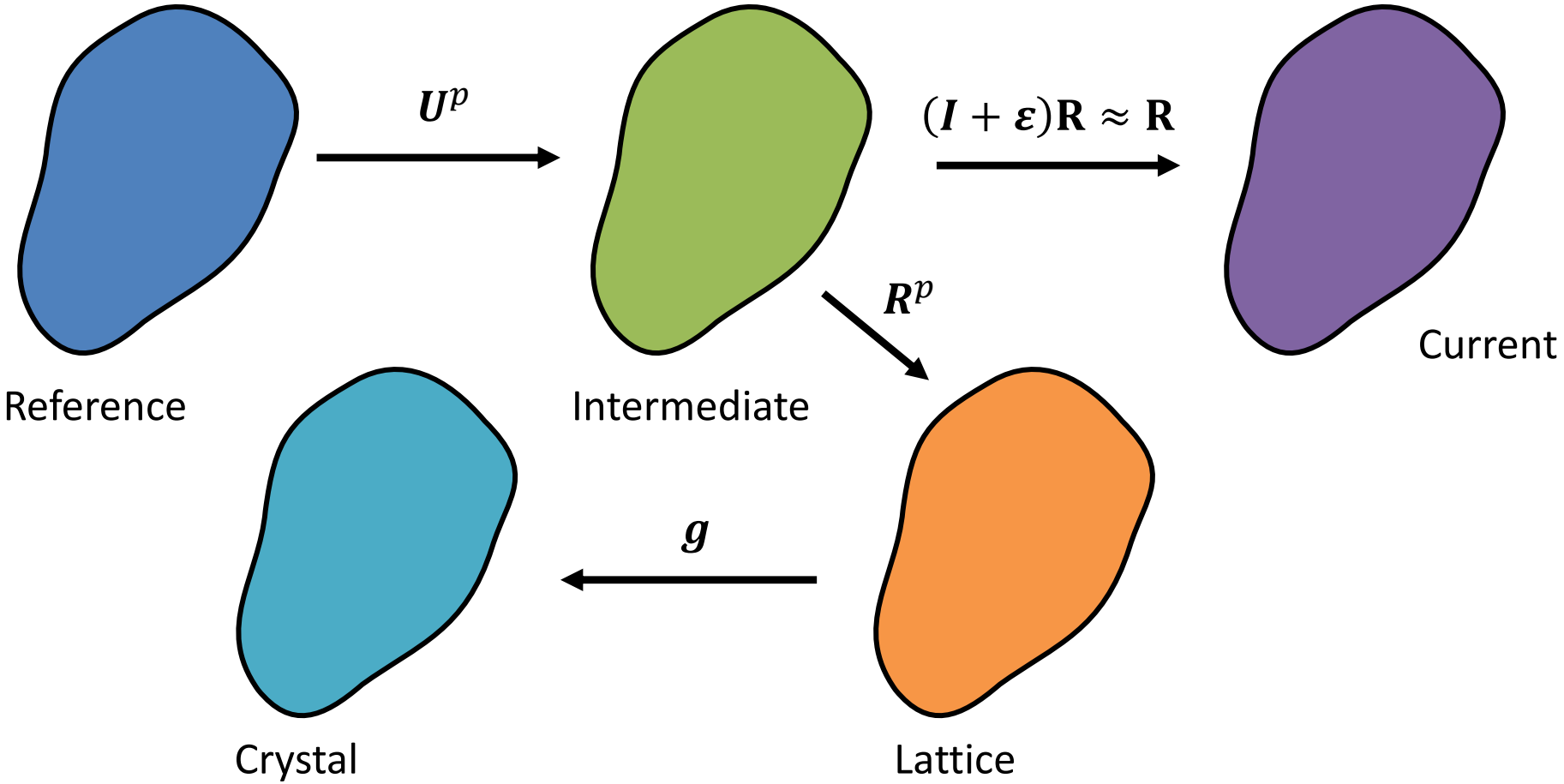
- Elastic properties:
  - WARP3D has isotropic (E, ν) and cubic (E, ν, μ)
- Slip systems:
  - FCC or BCC or custom (you can define your own)
  - Orientation of each element
- Rate sensitivity exponent n

Summary: CP material properties

- Constants
  - k – Boltzmann constant
  - b – Burgers vector
- Hardening properties
  - τ_a
  - Intrinsic (yield): $\hat{\tau}_y$, $\dot{\varepsilon}_0^y$, g_0^y , q_y, p_y
  - Work hardening
    - Voce: θ_0
    - Target strength: $\hat{\tau}_v$, $\dot{\varepsilon}_0^v$, g_0^v , q_v, p_v

Making things more complicated: Lattice rotation

- Lattice orientation does not stay the same with deformation
- Lattice deforms under *elastic* rotations
- $\boldsymbol{R} = \boldsymbol{R}^e \boldsymbol{R}^p$ or $\boldsymbol{R}^e = \boldsymbol{R}\boldsymbol{R}^{pT}$



Integrating lattice rotations

- Lattice rotations affect:
  - Slip system directions/normals
  - Elasticity tensors
- Plus they indicate the evolution of texture
- With some fun kinematics, can find:

$$\dot{\boldsymbol{R}}^{p} = \left\{ \sum_{s=1}^{n_{slip}} \dot{\gamma}^{(s)} \boldsymbol{R}^{pT} \operatorname{skew}\left(\boldsymbol{b}^{(s)} \otimes \boldsymbol{n}^{(s)} \right) \boldsymbol{R}^{p} \right\} \boldsymbol{R}^{p}$$

- Integrate with an exponential operator to maintain orthogonally

Making things more complicated: Geometric hardening

- Just a quick overview
- MTS hardening accounts for statically stored, forest dislocations, not excess, necessary dislocations
  - In 1D: $\rho_{excess} = \rho_+ - \rho_-$
  - In 3D: $\alpha_{excess} = -\nabla \times \boldsymbol{F}^{e-1} \approx -\nabla \times \boldsymbol{R}^{eT}$
- Since we can calculate the necessary density, we should apply it to hardening
  - One parameter k_0
  - Strain gradient plasticity