Handout
"Preparations to use DAMASK"

Philip Eisenlohr

June 16, 2015

1 Data Visualization

1. Install the open-source visualization tool called "ParaView" that can be found at `http://www.paraview.org`.

2 Server Connection

The machine located at `10.194.30.25` features a working installation of DAMASK. You can connect to this machine using either the account name "sandbox" or talk to Shalini to set up a personal account for you. We will connect as user "sandbox" in the following.

1. Install a client for Secure Shell (SSH) connections on your computer. This could be, for instance
   - PuTTy (`http://www.putty.org`) for Windows.
   - OpenSSH (`http://openssh.sourceforge.net`) for Windows.
   - preinstalled command `ssh` for Linux and Mac OS.
2. Install a client for Secure Copy (SCP) connections on your computer. This could be, for instance
   - WinSCP (`http://winscp.net`) for Windows.
   - preinstalled command `scp` for Linux and Mac OS.
3. With your SSH client of choice, connect to

- server IP: `10.194.30.25`
- user: `sandbox`
- pwd: `FullOfSand`

You should be greeted by

```
      Using environment with ...
      DAMASK /opt/DAMASK
      Spectral Solver /usr/local/bin/DAMASK_spectral
      Post Processing /usr/local/bin/postResults
      Multithreading DAMASK_NUM_THREADS=1
      PETSc location /opt/petsc/3.6.0
      MSC.Marc/Mentat /msc
```

4. For later use, create yourself a subdirectory in the home folder of user `sandbox`

```
mkdir YourName
```

3 Further Reading

1. Skim through the brief explanation of DAMASK at http://damask.mpie.de/
2. Read more in-depth starting at http://damask.mpie.de/Usage/GeneralUsage.

Handout
"2D crystal plasticity with DAMASK"

Philip Eisenlohr

June 16, 2015

1 Purpose

This lab should demonstrate how to use pre-processing scripts to generate geometry and material configuration input for two-dimensional simulations of isotropic plasticity and crystal plasticity using the spectral method solver shipped as part of DAMASK. We assume that you are working on a machine that is properly configured and start within a directory that should hold this labs content and outputs.

2 Geometry

1. Generate a "Seeds File" containing 40 points within a unit cube that will be later used to tessellate a volume discretized into $64 \times 64 \times 1$ voxels.

```
seeds_fromRandom -N 40 --grid 64 64 1 > 40grains.seeds
```

2. Generate an artificial grain structure using Voronoi tessellation around the seed points created above:

```
geom_fromVoronoiTessellation --grid 64 64 1 40grains.seeds
```

3. Visualize the outcome using

```
seeds_check 40grains.seeds
geom_check 40grains.mesh
```

   and transferring the resulting `seeds_40grains.vtu` and `mesh_40grains.vtr` file to your computer to be opened in ParaView.

3 Material configuration

1. Produce part of the necessary material configuration

```
geom_fromVoronoiTessellation --config 40grains.seeds
```

2. Copy the resulting `40grains_material.config` file

```
cp 40grains_material.config material.config
```

and add parts for homogenization and crystallite to `material.config`.

```
<homogenization>
[SX]
type none

<crystallite>
[essential]
(output) texture
(output) f
(output) p
(output) orientation
(output) grainrotation
```

3. For the phase definition, two alternatives shall be tried. To have purely isotropic behavior, specify

```
<phase>
{/opt/DAMASK/code/config/Phase_J2_AluminumIsotropic.config}
```

Truly crystalline (i.e. anisotropic) behavior will result with

```
<phase>
{/opt/DAMASK/code/config/Phase_Phenopowerlaw_Aluminum.config}
```

4 Load case

1. To pull the polycrystalline patch along y (upwards), we specify

```
Fdot * 0 0 0 1e-3 0 0 0 0 stress 0 * * * * * * * * time 100 incs 200 freq
    ↪ 5
```

as the only line within a file called "`tensionY.load`".

5 Simulation

1. Start the simulation with

```
DAMASK_spectral --load tensionY.load --geom 40grains.geom >
    ↪ 40grains_tensionY.out &
```

2. You can observe the progress of your simulation with

```
tail -f 40grains_tensionY.out
```

6 Post Processing

6.1 Spatial average

1. To extract the data of all increments in a spatially averaged way, issue

```
postResults 40grains_tensionY.spectralOut --cr f,p
```

   This will produce a subdirectory `postProc` with a single file in it. The file format of such files is termed "ASCIItable" in the context of DAMASK.

2. Check what data has been written to the ASCIItable

```
cd postProc
showTable --label 40grains_tensionY.txt
```

3. Based on the tensors of deformation gradient and first Piola–Kirchhoff stress ("f" and "p"), we can add derived quantities to the file

```
addStrainTensors 40grains_tensionY.txt --left --logarithmic
addCauchy 40grains_tensionY.txt
addMises 40grains_tensionY.txt --strain 'ln(V)' --stress Cauchy
```

4. The resulting data can either be graphed with appropriate tools, or displayed directly for a quick glance

```
filterTable < 40grains_tensionY.txt --white
    ↪ inc,'Mises(ln(V))','Mises(Cauchy)'
```

   Remember that `filterTable` operates *in place* when specifying an input file; which is why input redirection (`command < inputfile`) is used above.

6.2 Spatially resolved

1. To generate a spatially resolved dataset from the binary result file, one needs to use the "separation" (grouping) capability of `postResults` and should split the output into one ASCIItable per increment

```
postResults 40grains_tensionY.spectralOut --increments --range 200 200 1
    ↪ --split --separation x,y,z --cr
    ↪ texture,f,p,orientation,grainrotation
```

   Here, we were only interested in the last increment (200, see `tensionY.load`), as specified by the option `--range start end step`.

2. Again, adding derived quantities works similarly as for the spatially averaged case

```
addStrainTensors 40grains_tensionY_inc200.txt --left --logarithmic
addCauchy 40grains_tensionY_inc200.txt
addMises 40grains_tensionY_inc200.txt --strain 'ln(V)' --stress Cauchy
```

3. To mimic an orientation map as resulting, for instance, from an EBSD measurement, the orientation data in the ASCIItable can be transformed into an inverse pole figure (IPF)

```
addIPFcolor 40grains_tensionY_inc200.txt --pole 0 0 1 --symmetry cubic
    ↪ --quaternion orientation
```

   Hint: check with `showTable --label` what happens to the ASCIItable after each command...

4. As we presently deal with two-dimensional datasets, is it convenient to use image-generating scripts to visualize our data fields

```
imageData 40grains_tensionY_inc200.txt --label 'Mises(ln(V))' --dimension
    ↪ 64 64 --color bluered
imageDataRGB 40grains_tensionY_inc200.txt --label IPF_001_cubic
    ↪ --dimension 64 64
```

   These commands result in Portable Network Graphics (PNG) images that show the equivalent left logarithmic strain and "ND" inverse pole figure map, respectively.