CPFEM simulations from EBSD data: using MTEX2Gmsh for PRISMS-Plasticity

Dorian Depriester

January 19, 2021

This paper describes the steps for using the CPFEM software PRISMS-Plasticity from EBSD data, thanks to the MTEX, MTEX2Gmsh and Gmsh softwares.

Contents

1 Introduction 1

2 Requirements 2

3 Step-by-step procedure 2
- 3.1 Process EBSD data 2
- 3.2 Generate mesh 3
- 3.3 Export grain properties 3
- 3.4 Edit input parameters 3

4 Troubleshooting 4
- 4.1 Wrong/missing volume names 4
- 4.2 Wrong element type 5

1 Introduction

PRISMS-Plasticity is an open-source software for performing 3D Crystal Plasticity Finite Element (CPFE) simulations [1]. It is based on the deal.II library [2] for Finite Element Method (FEM) and takes advantage of parallel computing through Message Passing Interface (MPI), allowing to investigate large-scale problems, such as polycrystalline aggregates.

MTEX2Gmsh is a software dedicated to FEM mesh generation from Electron BackScatter Diffraction (EBSD) data [3]. It is based on the MTEX toolbox for MATLAB®, used for grain reconstruction [4], and converts the geometry in a Gmsh-readable file [5] so that it can be meshed. The resulting mesh is characterized by smooth grain boundaries and very realistic grains shapes. In addition, it allows to significantly reduce the number of elements thanks to the conforming part of the mesh within each grain.

This paper briefly describes the procedure for generating input data (e.g. mesh and grain properties file) for PRISMS-Plasticity from EBSD data, using MTEX, MTEX2Gmsh and Gmsh.

2 Requirements

The following softwares/toolboxes must be installed on your computer:

- MATLAB® 2013 (or newer),
- MTEX (available at `https://mtex-toolbox.github.io/`),
- Gmsh (available at `https://gmsh.info/`),
- MTEX2Gmsh (available at `https://github.com/DorianDepriester/MTEX2Gmsh`),
- PRISMS-Plasticity (available at `https://github.com/prisms-center/plasticity`).

Note that PRISMS-Plasticity can be installed on a remote machine (e.g. computing cluster).

3 Step-by-step procedure

3.1 Process EBSD data

In MATLAB®, process the EBSD data as you wish (remove small grains, fill holes etc.) and reconstruct the grains, e.g.:

```
grains=calcGrains(ebsd);
```

and make sure the bottom-left corner of the map is at $(0, 0)$ coordinates:

```
V=grains.V;
ori=min(V);
grains.V=V-repmat(ori,size(grains.V,1),1);
```

Then, construct the gmshGeo object:

```
G=gmshGeo(grains);
```

For the record, check the size of the Region of Interest (RoI):

```
max(grains.V)
ans =
          70.0729    70.0611
```

3.2 Generate mesh

Since 3D orientations are necessary to define the grains, CPFE will be performed in 3D. In 3D, deal.II only handles hexahedron elements. Thus, the mesh has to be generated with the "`HexOnly`" option for element type. In addition, the volumes corresponding to each grain must have no prefix in their name:

```
mesh(G, 'mesh.msh', 'ElementType', 'HexOnly', 'GrainPrefix', '',
    'thickness', 1)
```

Setting the thickness to a particular value (1 here) will help defining the boundary conditions (see section 3.4).

3.3 Export grain properties

In PRISMS-plasticity, the grain properties are read from a tabular file where the i-th lines gives (in that order):

1. the grain ID,
2. the first component of the Rodrigues vector,
3. the second component of the Rodrigues vector,
4. the third component of the Rodrigues vector,
5. and the phase ID of the i-th grain.

Such data format can be obtained with:

```
exportGrainProps(G, 'orientations.txt', 'PRISMS')
```

MATLAB® will print the indexing conventions it has used for the phases (useful for multiphased materials).

3.4 Edit input parameters

Move the mesh file (`mesh.msh`) and the orientation file (`orientations.txt`) in the same folder as the input parameter for PRISMS-Plasticity (say `prm.prm`). Now, edit this file and check for:

```
set Domain size X = 70.0729
set Domain size Y = 70.0611
set Domain size Z = 1.0
```

The X and Y sizes must be consistent with the size of the RoI (see section 3.1) whereas the Z size must be equal to the mesh thickness.

Turn on external mesh and specify the location for the mesh:

```
set Use external mesh                        = true
set Name of file containing external mesh = mesh.msh
```

When using an external mesh, the boundary conditions are applied to the nodes depending on their coordinates. Due to round-off error, a margin is used to check whether a node belongs to a boundary or not. The width of this margin is defined with the following option:

```
set External mesh parameter = 0.001
```

If this value is too small, some nodes may miss boundary conditions; conversely, if this value is too large, the margin may hang over the RoI.

Finally, check that the orientation file is used:

```
set Orientations file name  = orientations.txt
```

4 Troubleshooting

This section describes the typical errors thrown by deal.II or PRISMS-Plasticity when using an external mesh, generated from MTEX2Gmsh. Some hints for fixing those errors are provided.

4.1 Wrong/missing volume names

```
An error occurred in line <1714> of file </opt/dealii-9.2.0/
    source/grid/grid_in.cc> in function
     void dealii::GridIn<dim, spacedim>::read_msh(std::istream&)
          [with int dim = 3; int spacedim = 3; std::istream = std
         ::basic_istream<char>]
The violated condition was:
     line == begin_elements_marker[gmsh_file_format == 10 ? 0 :
         1]
 Additional information:
     The string <$EndNodes> is not recognized at the present
         position of a Gmsh Mesh file.
```

This error is thrown when the mesh volumes associated to the grains are not properly named. Check that no prefix is used when meshing:

```
mesh(..., 'GrainPrefix','')
```

4.2 Wrong element type

```
An error occurred in line <2009> of file </opt/dealii-9.2.0/
    source/grid/grid_in.cc> in function
     void dealii::GridIn<dim, spacedim>::read_msh(std::istream&)
         [with int dim = 3; int spacedim = 3; std::istream = std
         ::basic_istream<char>]
The violated condition was:
     false
Additional information:
     The Element Identifier <6> is not supported in the deal.II
        library when reading meshes in 3 dimensions.
```

This error means that their are some wedges (6-node elements) in the mesh. Check that you have used “`HexOnly`” as element type when meshing, not “`Hex`”.

References

[1] Mohammadreza Yaghoobi, Sriram Ganesan, Srihari Sundar, Aaditya Lakshmanan, Shiva Rudraraju, John E. Allison, and Veera Sundararaghavan. PRISMS-Plasticity: An open-source crystal plasticity finite element software. *Computational Materials Science*, 169:109078, 2019. ISSN 0927-0256. doi: https://doi.org/10.1016/j.commatsci.2019.109078. URL http://www.sciencedirect.com/science/article/pii/S0927025619303696.

[2] W. Bangerth, R. Hartmann, and G. Kanschat. Deal.II–a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software*, 33(4):24–es, August 2007. ISSN 0098-3500. doi: 10.1145/1268776.1268779. URL https://doi.org/10.1145/1268776.1268779.

[3] Dorian Depriester and Régis Kubler. MTEX2Gmsh: a tool for generating 2D meshes from EBSD data. *Journal of Open Source Software*, 5(52):2094, 2020. doi: 10.21105/joss.02094. URL https://doi.org/10.21105/joss.02094.

[4] Florian Bachmann, Ralf Hielscher, and Helmut Schaeben. Grain detection from 2d and 3d EBSD data—specification of the MTEX algorithm. *Ultramicroscopy*, 111(12):1720 – 1733, 2011. ISSN 0304-3991. doi: https://doi.org/10.1016/j.ultramic.2011.08.002. URL http://www.sciencedirect.com/science/article/pii/S0304399111001951.

[5] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi: https://doi.org/10.1002/nme.2579. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579.