PRISMS-Plasticity user manual-V1.3.0

Mohammadreza Yaghoobi, Aaditya Lakshmanan, Srihari Sundar

User input parameters file

FE parameters

Polynomial order of interpolation function (1 $\implies$ linear basis functions)

```
set Order of finite elements = 1
```

Quadrature point order ($2 \implies 2^n$ quadrature points where n is the physical dimension)

```
set Order of quadrature = 2
```

Domain parameters

Assuming that the simulation domain is a cuboid of arbitrary dimensions.

Number of physical dimensions for the simulation

```
set Number of dimensions = 3
```

The size of the domain in the x direction.

```
set Domain size X = 1.0
```

The size of the domain in the y direction.

```
set Domain size Y = 1.0
```

The size of the domain in the z direction.

```
set Domain size Z = 1.0
```

Mesh parameters

Meshing can be performed through deal.II or by reading an external mesh through a file generated using Gmsh.

Deal.II mesh generator

The mesh generation starts with a single unit cell, and slices it in x, y and z directions as many times as the refinement factor indicates. The mesh is then written out if the corresponding flag is set to true. The number of mesh subdivisions in the x direction.

```
set Subdivisions X = 1
```

The number of mesh subdivisions in the y direction.

```
set Subdivisions Y = 1
```

The number of mesh subdivisions in the z direction.

```
set Subdivisions Z = 1
```

The number of initial refinements of the coarse mesh (3 $\implies$ each coarse element is subdivided into $2^3 = 8$ smaller elements)

```
set Refine factor = 3
```

Only written for serial runs and if number of elements $<$ 10000

```
set Write Mesh To EPS = true
```

Important Note:
It is always more efficient to use the *set Refine factor* to refine the mesh rather than using the *subdivisions in the x,y, and z directions*.

External Mesh Generation

The external mesh with the format of Gmsh can be read.

Flag to indicate whether to use external mesh.

```
set Use external mesh = true
```

Name of external mesh file.

```
set Name of file containing external mesh = n200-id4_hex.msh
```

The external mesh parameter: The ratio of defined region size to the Domain size.

```
set External mesh parameter = 0.05
```

Note: When an external mesh is read, the x,y,z poistions are not precisely defined due to roundoff errors. This matters in the case of BCs assignment. Accordingly, here, a margin is defined in which if a point is located in this margin, it will be considered as BCs. The size of this region in each direction = (External mesh parameter)*(Domain size in that direction).

Solver output parameters

Flag to write output vtu and pvtu files

```
set Write Output = true
```

Output Directory name. For the current version, this directory has to be already present before running the simulation. A directory can be created using the `mkdir` Unix command.

```
set Output Directory = results
```

Number of Output Steps to skip

```
set Skip Output Steps = 0
```

Output Equivalent strain

```
set Output Equivalent strain = true
```

Output Equivalent stress

```
set Output Equivalent stress = true
```

Output Grain ID

```
set Output Grain ID = true
```

Output Twin fractions (`true` only when crystal structure is HCP)

```
set Output Twin fractions = true
```

Flag to write Quadrature Output

```
set Write Quadrature Output = true
```

Number of Quadrature Output Steps to skip

```
set Skip Quadrature Output Steps = 0
```

Tabular Time output:
One can use this feature to output the results (Both visualization and Quadrature outputs) at specific time table they define here.

Flag for Tabular Time output

```
set Tabular Output = true
```

Simulation times at which outputs are generated.

```
set Tabular Time Output Table  = 0,0.25,0.48
```

Output Userdefined Average Variable:
One can use this feature to output user defined variables as average values in the stressstrain.txt file.

Flag for Userdefined Average Variable

```
set Output Userdefined Average Variable = true
```

Number of Output Userdefined Average Variable which is intended to be added to the stressstrain.txt.

```
set Number of Output Userdefined Average Variable  = 1
```

The user needs to define the variables they want to output the average values in the file plasticity/src/-materialModels/crystalPlasticity/updateAfterIncrement.cc in the following lines:

```
local_userDefinedAverageOutput(0)=
  local_userDefinedAverageOutput(0)+0;
```

This is when we have only one added variable. The 0 on the right side side should be replaced by the variables one intend to print out in the average in the stressstrain.txt. If one intend to add more variables to stressstrain.txt file, they can copy and paste the line and change the index accordingly. For example, for two variables, it will become as follows:

```
local_userDefinedAverageOutput(0)=
  local_userDefinedAverageOutput(0)+0;
local_userDefinedAverageOutput(1)=
  local_userDefinedAverageOutput(1)+0;
```

Again, 0 on the right side should be replaced for each of the target variables.

Output Buffer Layer Removal Feature:
One can use this feature to exclude certain region of the sample as buffer layers from the average value calculation in the stressstrain.txt file. This feature is specifically useful for the cases user wants to add a buffer layer to exclude the effects of boundary conditions. The results are then averaged in the sample excluding the region defined as the buffer layers.

Flag for using buffer layer removal from the average values reported in stressstrain.txt:

```
set Output Buffer Layer Removal Feature = true
```

The axis normal to the plane the buffer layers are defined. It is a integer and for x-axis normal =0, y-axis normal=1, and z-axis normal=2.

```
set Output Buffer Layer Removal dimension x0 y1 z2  = 2
```

For example, in the above case, the normal axis is z-axis and all the element with the z-center larger than the upper bound and z-center smaller than the lower bound will be excluded from the calculation of average values in the stressstrain.txt.

The upper and lower bounds are defined as follws:

```
set Output Buffer Layer Removal Lower Bound  = 0.2
set Output Buffer Layer Removal Upper Bound = 0.8
```

Output Userdefined visualization Variable:
One can use this feature to output user defined variables in the visualization file projectedFields. This file can be opened using the Paraview for visualization and postprocessing. Up to 24 variables can be defined and processed as an additional user-defined variables.

For each of extra variables for visualization, one need to add a flag. For example, in the case of having two extra visualization parameters, on should have the following lines:

```
set Output Variable 1 = true
set Output Variable 2 = true
```

The user needs to define the variables they want to output in their visualize in the file plasticity/src/materialModels/crystalPlasticity/updateAfterIncrement.cc. For example, for adding two additional variables for post-processing, they should modify these two lines:

```
this->postprocessValues(cellID, q, 3, 0) = 0;
this->postprocessValues(cellID, q, 4, 0) = 0;
```

The 0 on the right side side should be replaced by the variables one intend to visualize in the using the file projectedFields. One should note that all these variables are calculated at the quadrature points and bring back to the nodal points for the purpose of visualization.

Note: Even if all output generation fields are disabled, a `stressstrain.txt` file is generated for each step which includes the average Green-Lagrange strain tensor, Cauchy stress tensor, twin activity volume (TwinRealVF), twin volume fraction (TwinMade), and slip activity (SlipTotal). If one wants to compare the average slip and twin activities, i.e., TwinRealVF vs SlipTotal, they should consider that average twin activity volume is the twin pseudo-slip systems shear activity divided by the characteristic twin shear constant.
In the case of Quadrature Output file, it outputs the following variables for all quadrature points: GrainID, Phase ID (If Advanced Twinning model is not used), det(J), twin (1 if it is reoriented due to twinning and 0 if not), x,y,z (coordinates of quadrature point),rotnew(1), rotnew(2), rotnew(3) (updated Rodrigues vector of orientation), 9 components of Fe (elastic deformation gradient tensor), 9 components of Fp, 9 components of Cauchy stress tensor, slip activity for 84 slip systems (if there are less systems, it will be appeared as 0), and 6 twin activity volume. Again, the twin activity volume is the twin pseudo-slip systems shear activity divided by Characteristic twin shear constant.

Boundary conditions information

Boundary conditions (BCs) can be applied by defining the applied displacements or velocity gradient tensor. Different boundary conditions can be applied to the sample including:

Simple BCs
Velocity gradient tensor
Cyclic BCs
Tabular BCs
SEM-DIC (Digital Image Correlation) BCs
Periodic BCs
Nodal BCs
Tabular Torsion BCs
Tabular Neumann BCs

The corresponding input lines for each of these methods is presented here.

Displacement method

This flag defines if one use *Simple BCs*. It will be considered *true* if it is not defined. In the case of other BCs types, it should be defined as *false*.

```
#Flag to indicate whether to use Simple (Basic) BCs
set Use Simple BCs                                  = true
```

File name containing BC information

```
set Boundary condition filename = BCinfo.txt
```

A sample boundary condition specification could look like:

```
# Header lines = 2
# FaceID DoF FinalDisplacement
5 1 0
5 2 0
5 3 0
6 3 0.01
```

FaceID refers to the face identifier and follows the numbering as indicated in Fig. 1. **DoF** refers to the direction of application of displacement, where 1, 2 and 3 refer to the x, y and z directions respectively. In the case of cyclic loading the displacement applied refers to the displacement amplitude.

Number of header lines in BC file

```
set BC file number of header lines = 2
```

Number of boundary conditions

```
set Number of boundary conditions = 4
```

Velocity gradient tensor

This flag defines if one use *Simple BCs*. In this case, it should be defined as *false*.

```
#Flag to indicate whether to use Simple (Basic) BCs
set Use Simple BCs                                  = false
```

Flag to indicate whether to use velocity gradient tensor to apply BCs

```
set Use velocity gradient BC = true
```

Velocity gradient tensor including the multiplication factor

Row 1

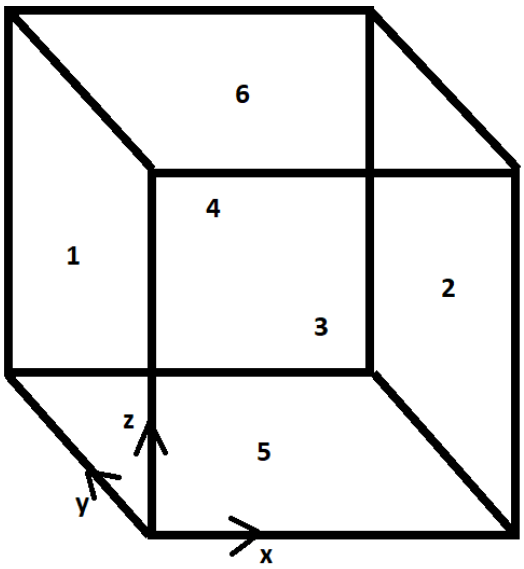

Figure 1: Description of faces required for defining BCs

```
set Velocity gradient row 1 = -0.005, 0, 0
```

Row 2

```
set Velocity gradient row 2 = 0, -0.005, 0
```

Row 3

```
set Velocity gradient row 3 = 0, 0, 0.01
```

Cyclic loading information

This flag defines if one use *Simple BCs*. In this case, it should be defined as *false*.

```
#Flag to indicate whether to use Simple (Basic) BCs
set Use Simple BCs                                  = false
```

Flag to indicate if cyclic loading is enabled

```
set Enable cyclic loading = true
```

Face that is cyclically deformed

```
set Cyclic loading face = 6
```

Direction along which the face is cyclically deformed

```
set Cyclic loading direction = 3
```

Time for finishing quarter of a cyclic loading cycle. One cycle is time taken for starting from 0 displacement and ending at 0 displacement, with a positive loading rate.

```
set Quarter cycle time = 0.05
```

Tabular BCs

This flag defines if one use *Simple BCs*. In this case, it should be defined as *false*.

```
#Flag to indicate whether to use Simple (Basic) BCs
set Use Simple BCs                                  = false
```

Flag to indicate whether to use Tabular BCs

```
set Use Tabular BCs                                 = true
```

File name containing Tabular BC information

```
set Tabular Boundary condition filename          = BCinfoTable
  .txt
```

Number of time data for Tabular BCs (it includes the initial BCs)

```
set Number of time data for Tabular BCs          = 5
```

Number of tabular boundary conditions

```
set Number of tabular boundary conditions        = 4
```

Table for Time intervals of Tabular BCs

```
set Tabular Time Table                  = 0,5,15,25,30
```

The format for the BCinfoTable.txt (input file for Tabular BCs) is as follows:

```
1 1 0 0 0 0 0
3 2 0 0 0 0 0
5 3 0 0 0 0 0
6 3 0 -0.05 0.05 -0.05 0
```

The format is very similar to the Simple BCs input file. However, there is no header line for Tabular BCs input file. The first column is the *FaceID*. The second column is *DoF*. The following columns are the displacement at the times defined in the *Tabular Time Table*. For example, the Tabular BCs defined above is fixing faces 1,3 and 5 throughout the simulations. Face 6 is loaded in the 3 direction, i.e., z direction in Fig. 2.

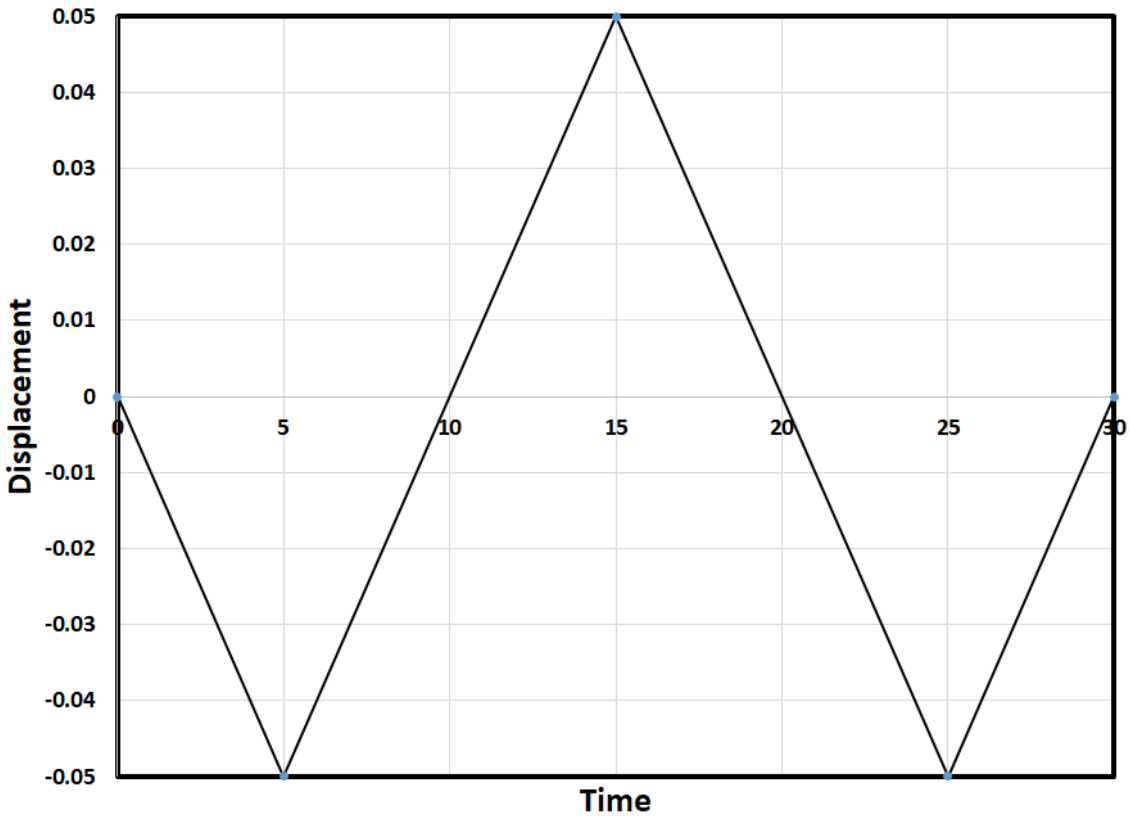

Figure 2: Tabular BCs (displacement versus time)

SEM-DIC (Digital Image Correlation) BCs

This flag defines if one use *Simple BCs*. In this case, it should be defined as *false*.

```
set Use Simple BCs                                    = false
```

Flag to indicate whether to use DIC BCs

```
set Use DIC pipeline                                   = true
```

Number of time data for DIC BCs (it includes the initial BCs)

```
set Number of Input data for DIC experiment= 3
```

Table for Time intervals of DIC BCs

```
set DIC Time Table =0,1,2
```

Number of DIC BCs input point in the x-direction

```
set Number of Points in DIC input in X direction =3801
```

Number of DIC BCs input point in the y-direction

```
set Number of Points in DIC input in y direction =3801
```

Number of DIC BCs input point in the z-direction

```
set Number of Points in DIC input in z direction =2
```

DIC Boundary condition filename 1 (x == 0.0)

```
set DIC Boundary condition filename 1 = bc1.txt
```

The file *bc1.txt* includes the BCs input for the side with (x==0). The format for the bc1.txt (input file for DIC BCs) is as follows:

```
0 0 0 0.45654 0.85664 0.45654 0.85664
0.019531  0 0 0.45717 0.85666 0.45717 0.85666
0.039063  0 0 0.45789 0.85675 0.45789 0.85675
.
.
.
74.219  0 0 0.20748 -0.37916  0.20748 -0.37916
```

This file is for a side of *(x==0)* from a square sample with the length of *74.219*. The first column is the position of that point in *y-direction*. For each line, the columns after the first column defines the displacement history in the *x* and *y* directions. For example for the first line, it is related to the point with (y=0). The second and third columns, i.e., 0 0, are the *u* and *v* (displacement in the *x* and *y* directions) at time 0 (The first data in the *DIC Time Table*). The fourth and fifth columns, i.e., 0.45654 0.85664, are the *u* and *v* (displacement in the *x* and *y* directions) at time 1 (The second data in the *DIC Time Table*). The sixth and seventh columns, i.e., 0.45654 0.85664, are the *u* and *v* (displacement in the *x* and *y* directions) at time 2 (The third data in the *DIC Time Table*).

DIC Boundary condition filename 2 (x == spanX)

```
set DIC Boundary condition filename 2 = bc2.txt
```

The file *bc2.txt* includes the BCs input for the side with (x==spanX). The format is similar to the bc1.txt file.

DIC Boundary condition filename 3 (y == 0)

```
set DIC Boundary condition filename 3 = bc3.txt
```

The file *bc3.txt* includes the BCs input for the side with (y==0). The format is similar to the bc1.txt file, except that the first column is the the position of that point in *x-direction*.

DIC Boundary condition filename 4 (y == spanY)

```
set DIC Boundary condition filename 4 = bc4.txt
```

The file *bc4.txt* includes the BCs input for the side with (y==spanY). The format is similar to the bc3.txt file.

Note: In the case of DIC BCs, the point with *x=y=z=0* is fixed in the *z-direction* (normal to the DIC measurements axes).

Periodic BCs

Included in PRISMS-Plasticity is the capability to perform CPFE simulations with fully periodic boundary conditions (PBCs). This is crucial in calculations of fatigue resistance when smaller microstructure instantiations are employed that represent bulk, i.e., subsurface, material response. Przybyla (2010) (See the references) presented the details of the PBCs required to mimic the subsurface microstructure during the fatigue simulation. In PRISMS-Fatigue framework, PBCs are implemented using a set of linear constraints. These constraints are defined on three components of opposing Faces, parallel Edges, and Vertices. Rigid body motion is also excluded by assigning additional constraints. Three sets of constraints are applied on nodes on opposing Faces (in the direction perpendicular to those Faces), nodes on each set of four parallel Edges (in the two directions perpendicular to the direction in which the Edges are parallel), and Vertices (in all directions). The contraction or expansion of the dimensions of the actual microstructure volume are considered. For further detail, one can read the **section 5.5** of Przybyla (2010) (See the references).

This flag defines if one use *Simple BCs*. In this case, it should be defined as *false*.

```
set Use Simple BCs                                    = false
```

Flag to indicate whether to use Periodic BCs

```
set Use Periodic BCs                                   = true
```

The name of the file which include the constraints between the Vertices, Edges, and Faces. For a specific type of BCs, this should not be changed. For example, if one models the fully periodic BCs with different loading conditions, this file should not be changed.

```
set Periodic Boundary condition Constraint filename =
  PeriodicBCsConstraints.txt
```

This file includes the linear constraints for Vertices, Edges, and Faces. For each constraint, the file has two lines, first one for DOFs and second one for their corresponding coefficients. The convention for the DOFs are defined in **section 5.5** of Przybyla PhD Thesis (2010). For the Vertices constraints, each row has 5 columns. The first column defines the number of DOFs involves in this constraint. All remaining 4 columns correspond to the DOFs of Vertices. In the case of Edges and Faces, we only have the latter 4 columns. In the cases of constraints for Edges, the first two columns correspond to the DOFs of the Edges, while the third and fourth columns correspond to the Vertices. In the case of Faces, the first two columns correspond to the DOFs of the Faces, while the third and fourth columns correspond to the Vertices.
The DOFs convention for Vertices, Edges, and Faces used here are described as follws:

Vertices:
0=V000$_x$, 1 = $V000_y$, 2 = $V000_z$, 3 = $V100_x$, 4 = $V100_y$, 5 = $V100_z$, 6 = $V010_x$, 7 = $V010_y$, 8 =

$V010_z, 9 = V001_x, 10 = V001_y, 11 = V001_z$
$12 = V110_x, 13 = V110_y, 14 = V110_z, 15 = V101_x, 16 = V101_y, 17 = V101_z, 18 = V011_x, 19 = V011_y, 20 = V011_z, 21 = V111_x, 22 = V111_y, 23 = V111_z$

Edges:
0=EX000$_x, 1 = EX000_y 2 = EX000_z 3 = EX001_x 4 = EX001_y 5 = EX001_z 6 = EX010_x 7 = EX010_y, 8 = EX010_z 9 = EX011_x 10 = EX011_y 11 = EX011_z 12 = EY000_x, 13 = EY000_y, 14 = EY000_z, 15 = EY001_x, 16 = EY001_y, 17 = EY001_z, 18 = EY100_x, 19 = EY100_y, 20 = EY100_z, 21 = EY101_x, 22 = EY101_y, 23 = EY101_z, 24 = EZ000_x, 25 = EZ000_y, 26 = EZ000_z, 27 = EZ010_x, 28 = EZ010_y, 29 = EZ010_z, 30 = EZ100_x, 31 = EZ100_y, 32 = EZ100_z, 33 = EZ110_x, 34 = EZ110_y, 35 = EZ110_z$

Faces:
0=FXN$_x, 1 = FXN_y, 2 = FXN_z, 3 = FXP_x, 4 = FXP_y, 5 = FXP_z, 6 = FYN_x, 7 = FYN_y, 8 = FYN_z, 9 = FYP_x, 10 = FYP_y, 11 = FYP_z, 12 = FZN_x, 13 = FZN_y, 14 = FZN_z, 15 = FZP_x, 16 = FZP_y, 17 = FZP_z$

Important note: The first column of all **Faces** constraint lines must always start with the positive Faces, i.e., FXP,FYP, and FZP.

For example, these are three constrains for Faces, Edges, and Vertices:

Vertices:
V110$_x - V010_x - V100_x + V000_x = 0$
In the periodic BCs file, this becomes:
4 12 6 3 0
4 1 -1 -1 1

Edges:
-EX001$_x + EX011_x - V011_x + V001_x = 0$
In the periodic BCs file, this becomes:
3 9 18 9
-1 1 -1 1

One should note that we named this section Edges to show that it includes the DOFs of the Edges. However, the last two columns belong to the Vertices DOFs.

Faces:
-FXN$_x + FXP_x - V100_x + V000_x = 0$
As mentioned earlier, the first column of all face constraint lines must always start with the positive Faces, i.e., FXP,FYP, and FZP.
So we first rearrange the equation as follows:
FXP$_x - FXN_x - V100_x + V000_x = 0$
In the periodic BCs file, this becomes:
3 0 3 0
1 -1 -1 1

One should note that we named this section Faces to show that it includes the DOFs of the Faces. However, the last two columns belong to the Vertices DOFs.

The number of linear constrains for Vertices, Edges, and Faces are defined next.

```
set Number of Vertices Constraints  = 12
set Number of Edges Constraints  = 14
set Number of Faces Constraints = 6
```

The above numbers describes the sample with periodic BCs in 2 directions and free surface in the third one. In the case of fully periodic BCs, the numbers become as follows:

```
set Number of Vertices Constraints = 12
set Number of Edges Constraints  = 27
set Number of Faces Constraints  = 9
```

The next part defines the BCs which are applied to the Vertices. The notation used here is in line with the **section 5.5** of Przybyla PhD Thesis (2010). The first line control which Vertices DOF the BCs are applied to by assigning the value of 1. Otherwise, the value is 0 and that DOF has no predefined BCs. The second line define the values for BCs at the end of the total time.

```
# 0=V000_1;1=V000_2;2=V000_3; 3=V100_1;4=V100_2;5=V100_3; 6=
  V010_1;7=V010_2;8=V010_3; 9=V001_1;10=V001_2;11=V001_3;
# 12=V110_1;13=V110_2;14=V110_3; 15=V101_1;16=V101_2;17=V101_3
  ; 18=V011_1;19=V011_2;20=V011_3; 21=V111_1;22=V111_2;23=
  V111_3;
set Vertices Periodic BCs row 1 = 1,1,1, 1,1,1, 1,0,1, 1,1,0,
  1,0,0, 1,0,0, 0,0,0, 1,0,0
set Vertices Periodic BCs row 2 = 0,0,0, 0.05,0,0, 0,0,0,
  0,0,0, 0.05,0,0, 0.05,0,0, 0,0,0, 0.05,0,0
```

One can also define the periodic BCs along Tabular loading. First, this flag should be set to true:

```
set Use Tabular Periodic BCs = true
```

Next, the second line of the applying BCs in the input file lines, i.e., **set Vertices Periodic BCs row 2**, will be the applied BCs at the time of the periodic Tabular time, which is defined as follows:

```
set periodic Tabular time = 10
```

This is to define the **base rate** at which the BCs are applied. In other words, the **base rate** of BCs is equal to **(Vertices Periodic BCs row 2)/(periodic Tabular time)** for each DOF of Vertices.

For example, here, for the DOF of **3=V100**$_1$, we defined the value of 0.05 in **set Vertices Periodic BCs row 2**. The **base rate** can then be calculated as:

base rate=0.05/(set periodic Tabular time)=0.05/10=0.005.

Finally, the Tabular BCs is applied using Tabular Periodic Time Table and their coefficients as follows:

```
set Tabular Periodic Time Table = 0,10,30,50,70,90,110
set Tabular Periodic Time Table Coefficient = 1,-1,1,-1,1,-1,1
```

The first line defines the time table, and second line defines the corresponding coefficients for each of these time spans. For each time span (i.e., between each two times inside the **Time Table**), the rate of deformation is applied as **(Corresponding Coefficient*base rate)**.

For example, the above values for these two lines state that:
Time$_1 = 0; Time_2 = 10; Time_3 = 30; Time_4 = 50; Time_5 = 70; Time_6 = 90; Time_7 = 110;$

The corresponding coefficients are:
Coefficient$_1 = 1; Coefficient_2 = -1; Coefficient_3 = 1; Coefficient_4 = -1; Coefficient_5 = 1;$
$Coefficient_6 = -1; Coefficient_7 = 1;$

For the first time span, the applied BCs between *(Time$_1$=0)* to *(Time$_2$=10)* has the rate of:

*(Coefficient$_1$=1)**__(base rate)__

From *(Time$_2$=10)* and *(Time$_3$=30)*, the rate is the *(Coefficient$_2$=-1)**__(base rate)__.

This tabular loading defines a cyclic loading with equal max compression and tension strains.

The sample input files for different loading conditions and fully periodic BCs is provided in the following folder:
applications/crystalPlasticity/fcc/periodicBCs

Nodal BCs

One can define specific nodal dof Dirichlet BCs using this feature. To define this BCs set, one should define the nodal coordinate system of the considered node along with the final displacement and its corresponding dof as a separate file and provide the name of that file in the prm.prm input file. For an application, please check the work of Stopka et al. (2021), in which a type of BCs is defined as minimal non-periodic boundary conditions. Here are the lines in prm.prm related to Nodal BCs:

Flag to indicate whether to use Nodal BCs

```
set Enable Nodal Displacement BCs                    = true
```

Number of nodal dofs one defines for the simulation. One should note that the number of rows in the nodal BCs input file should be equal to the number defined here:

```
set Number of Nodal Displacement BCs= 4
```

The tolerance to define how close the coordinate system of a node should be to the one defined in the Nodal BCs input file to be considered as a part of this preset nodal BCs. The final threshold distance in each direction is defined as the value of this threshold multiplied by the sample size in that direction.

```
set Tolerance for Nodal Displacement BCs = 1e-8
```

The name of the Nodal BCs input file.

```
set Nodal Displacement BCs filename = NodalBCs.txt
```

The format for the NodalBCs.txt (input file for Tabular BCs) is as follows:

```
0 0 0 0 0
0 0 0 1 0
0 0 0 2 0
1 0 0 1 0
```

In the NodalBCs.txt, each line define BCs for one dof. The first three columns are (x,y,z) coordinate of the node. The 4th column defines the dof for the defined BCs for that node, in which 0 is x, 1 is y, and 2 is z. The last column is the final displacement for that dof. For example, the presented file has applied 4 predefined nodal dofs. The first three line fix the node (x=0,y=0,z=0) along all three directions throughout the simulation. The 4th line fix the node (x=1, y=0,z=0) along y-direction.

Tabular Torsion BCs

One can define the tabular torsion BCs using the following lines.

Flag to indicate whether to use Tabular Torsion BCs

```
set Use Torsion BCs = true
```

The torque axis. Here, 0 is for torque axis parallel to x-axis, 1 is for torque axis parallel to y-axis, and 2 is for torque axis parallel to z-axis.

```
set Torsion Axis  x 0  y 1  z 2  = 2
```

The next line defines the tabular time data set for torsion.

```
set Tabular Time Table for Torsion = 0, 0.1, 0.2
set Tabular Torsion BCs =0, 0.001, -0.001
```

The first data in "set Tabular Torsion BCs" does not matter. From the time 0 to 0.1, the angular velocity of 0.001 is applied to the sample in which positive sign means counterclockwise. Here, the angular torque applied at each time step of dt is 0.001*dt. From 0.1 to 0.2, the angular velocity becomes -0.001.

While we defined the torque axis, we still need to define where axis pass through the sample. The coordinates of that point within the plane is defined here.

```
set Center point for Torsion= 0.5, 0.5
```

For the presented data, the torque axis is parallel to z-axis, i.e., Torsion Axis = 2, and it passes through the point (x=0.5,y=0.5).

Tabular Neumann BCs

One can define the Tabular Neumann BCs using the following lines. This feature provide the Neumann BCs in the form of uniform traction for each of the sample face IDs.

Flag to indicate whether to use Tabular Neumann BCs

```
set Use Neumann BCs = true
```

The name of the input file for Neumann BCs.

```
set Tabular Neumann Boundary condition filename=NeumannBCs.txt
```

The next line defines the number of time data for Neumann BCs that defines the tabular Neumann BCs.

```
set Number of time data for Tabular Neumann BCs = 3
```

The next line defines the number of Neumann BCs we define within the input file for Neumann BCs. One should note that the number of rows in the input file for Neumann BCs should be equal to the number defined here:

```
set Number of tabular Neumann boundary conditions = 1
```

The next line defines the Boundary IDs that the uniform traction is applied for each of the tabular Neumann boundary conditions. In this case, since we have only one tabular Neumann boundary conditions, we need to define one Boundary IDs. Boundary IDs following the displacement IDs convention presented in Figure 1.

```
set Boundary IDs of Neumann BCs = 6
```

The next line defines the dof that the uniform traction is applied to the sample for each of the tabular Neumann boundary conditions. In this case, since we have only one tabular Neumann boundary conditions, we need to define one dof. Here, 0 means traction along x-axis, 1 means traction along y-axis, and 2 means traction along z-axis.

```
set dof of Neumann BCs = 2
```

The next line defines the tabular time table for the Neumann BCs in the form of uniform traction.

```
set Tabular Time Neumann BCs = 0,0.1,1
```

The value of the uniform traction is linearly interpolated using these time table from the values defined in the input file for Neumann BCs, for example NeumannBCs.txt. In this case, since we have ”Number of tabular Neumann boundary conditions = 1”, the NeumannBCs.txt contains only one line, and since we have ”Number of time data for Tabular Neumann BCs = 3”, within that line, it should have 3 columns each definening the traction magnitude for the corresponding time data. Here is an example for the described NeumannBCs.txt:

0 10 -50

Let’s explain the applied tabular Neumann BCs defined in the previous lines. Here, we have 1 applied uniform traction on Boundary ID of 6 (see Figure 1) along z-axis. The magnitude of traction from time1 = 0 to time2 = 0.1 linearly increases from 0 to 10, and after that it linearly varies until it becomes -50 at time3 = 1.

Solver parameters

ΔT for every increment

```
set Time increments = 0.1
```

Total simulation time

```
set Total time = 10
```

Maximum iterations for linear solver

```
set Maximum linear solver iterations = 50000
```

Relative linear solver tolerance

```
set Relative linear solver tolerance = 1.0e-10
```

Maximum non-linear iterations in the Newton-Raphson method

```
set Maximum non linear iterations = 4
```

Constitutive model solver parameters

Stress tolerance for the yield surface (MPa)

```
set Stress Tolerance = 1.0e-9
```

Maximum number of active slip search iterations

```
set Max Slip Search Iterations = 1
```

Input microstructure

Number of voxels in x direction

```
set Voxels in X direction = 20
```

Number of voxels in y direction

```
set Voxels in Y direction = 20
```

Number of voxels in z direction

```
set Voxels in Z direction = 20
```

Grain IDs file name

```
set Grain ID file name = grainID.txt
```

Number of header Lines in grain ID file (these are to be skipped)

```
set Header Lines GrainID File = 20
```

Grain orientations file name

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

Element deletion feature: This feature delete elements with the specific grain ID. Be careful, if this feature is used, the user cannot use ”set Refine factor” for mesh refinement. In other words, set Refine factor should be 0 in the case of element deletion feature.

```
set Use Element Deletion =true
```

The elements with this grain ID will be deleted from the mesh.

```
set GrainID of Deleted Elements = 0
```

Multiphase Parameters

Flag to indicate if Multiphase is enabled

```
set Enable Multiphase = true
```

Number of phases in the sample

```
set Number of Phases = 2
```

The phase id is inserted in the **Orientations file** after grain id, rot 1, rot 2, and rot 3 in 5th column.

Additional Voxel info

Number of Additional Voxel info in addition to the three orientation components and Phase if multiphase is enabled

```
set Additional Voxel info = 2
```

The aditional voxel info is inserted in the **Orientations file** after the columns grain ID, rot 1, rot 2, rot 3, and phase id (in the case of multiphase).

User Defined Material Model Parameters

This section is related to the define **user defined material models** including the number of material constants and state variables, material constants, and initial value of state variables. Here, it is assumed we have two phases with user defined material models.

Flag to indicate if User Material Model is enabled

```
set Enable User Material Model = true
```

Flag to indicate if User Material Model is enabled Phase 1

```
set Enable User Material Model 1 = true
```

Number of User Material Constants in a Material model Phase 1

```
set Number of User Material Constants 1 = 5
```

Number of User Material State Variables in a Material model Phase 1

```
set Number of User Material State Variables 1 = 2
```

Material Constants in a Material model Phase 1

```
set User Material Constants 1 = 0.25, 0.35, 20000,-1,-0.256
```

Material State Variables in a Material model Phase 1

```
set User Material State Variables Initial Values 1 = -0.1,
  10000
```

Flag to indicate if User Material Model is enabled Phase 2

```
set Enable User Material Model 2 = true
```

Number of User Material Constants in a Material model Phase 2

```
set Number of User Material Constants 2 = 10
```

Number of User Material State Variables in a Material model Phase 2

```
set Number of User Material State Variables 2 = 1
```

Material Constants in a Material model Phase 2

```
set User Material Constants 2 =
  1,-0.1,0.1,0,20,1,-0.1,0.1,0,20
```

Material State Variables in a Material model Phase 2

```
set User Material State Variables Initial Values 2 = -10000
```

Elasticity parameters

Input elastic stiffness matrix in Voigt notation as separate rows. The units are MPa.

Row 1

```
set Elastic Stiffness row 1 = 170.0e3, 124.0e3, 124.0e3, 0, 0,
   0
```

Row 2

```
set Elastic Stiffness row 2 = 124.0e3, 170.0e3, 124.0e3, 0, 0,
   0
```

Row 3

```
set Elastic Stiffness row 3 = 124.0e3, 124.0e3, 170.0e3, 0, 0,
   0
```

Row 4

```
set Elastic Stiffness row 4 = 0, 0, 0, 75.0e3, 0, 0
```

Row 5

```
set Elastic Stiffness row 5 = 0, 0, 0, 0, 75.0e3, 0
```

Row 6

```
set Elastic Stiffness row 6 = 0, 0, 0, 0, 0, 75.0e3
```

Slip parameters

Number of Slip Systems

```
set Number of Slip Systems = 18
```

Latent Hardening Ratio

```
set Latent Hardening Ratio = 1.4
```

Initial slip resistances

```
set Initial Slip Resistance =
  0.25,0.25,0.25,10,10,10,10,10,10,10,10,10,15,15,15,15,15,15
```

Hardening moduli of slip systems

```
set Initial Hardening Modulus = 5.0, 5.0, 5.0, 100, 100.0,
  100.0, 100.0, 100.0, 100.0, 100.0, 100.0, 100.0, 200.0,
  200.0, 200.0, 200.0, 200.0, 200.0
```

Power law coefficient in slip rate equation

```
set Power Law Exponent = 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1
```

Saturation stress

```
set Saturation Stress = 185.0, 185.0, 185.0, 160.0, 160.0,
  160.0, 160.0, 160.0, 160.0, 160.0, 160.0, 160.0, 200.0,
  200.0, 200.0, 200.0, 200.0, 200.0
```

Slip Directions File

```
set Slip Directions File = slipDirections.txt
```

Slip Normals File

```
set Slip Normals File = slipNormals.txt
```

Kinematic hardening parameters

This section controls the parameters used in the **rate-independent constitutive model with kinematic hardening**. The model description and implementation is according tho two works of Staroselsky and Cassenti (2010,2011) (See the references). C1 and C2 are two material constants which govern the evolution of backstress term inside the kinematic hardening formulation.

Flag to indicate if kinematic hardening is enabled

```
set Enable Kinematic Hardening = true
```

C1 Slip Kinematic Hardening parameters

```
set C1 Slip Kinematic Hardening = 10.0, 10.0, 10.0, 10.0,
  10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0,
  10.0, 10.0, 10.0, 10.0
```

C2 Slip Kinematic Hardening parameters

```
set C2 Slip Kinematic Hardening = 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1
```

Twin parameters

Flag to indicate if system twins

```
set Twinning enabled = true
```

Note: If one is using the *advanced twin model* (please read the manual **PRISMS-Plasticity: Constitutive Models Library**), they need to put another flag here as follows:

Flag to indicate if Advanced Twinning Model enabled

```
set Advanced Twinning Model enabled                    = true
```

Please check the input file in the application regarding to use of advanced model, which is in the following folder:
plasticity/applications/crystalPlasticity/hcp/advancedTwinModel-Cyclic/

Also, one should replace the original *calculatePlasticity.cc* inside *plasticity/src/materialModels/crystalPlasticity/* with the one related to advanced twin model inside the material library: *plasticity/src/-materialModels/crystalPlasticity/MaterialModels/RateIndependentAdvancedTwinModel/* and recompile the *PRISMS-Plasticity software*.

Number of Twin Systems

```
set Number of Twin Systems = 6
```

Note: If one is using the *advanced twin model* (please read the manual **PRISMS-Plasticity: Constitutive Models Library**), they need to have twice the number of twin systems as an input for initial Initial CRSS, Hardening moduli, power law exponent, and saturation stress. For example, if one defines 6 twin system, they should have 12 inputs. Please check the following folder for sample prm.in file:
plasticity/applications/crystalPlasticity/hcp/advancedTwinModel-Cyclic/

Initial CRSS of the twin systems

```
set Initial Slip Resistance Twin = 2.0, 2.0, 2.0, 2.0, 2.0,
  2.0
```

Hardening moduli of twin systems

```
set Initial Hardening Modulus Twin = 10.00, 10.00, 10.00,
  10.00, 10.00, 10.00
```

Power law exponents of twin systems

```
set Power Law Exponent Twin = 0.25, 0.25, 0.25, 0.25, 0.25,
  0.25
```

Saturation stress of twin systems

```
set Saturation Stress Twin = 185.00, 185.00, 185.00, 185.00,
  185.00, 185.00
```

Characteristic twin shear

```
set Characteristic Twin Shear = 0.129
```

Twin growth saturation factor

```
set Twin Saturation Factor = 0.25
```

Threshold fraction of characteristic twin shear (< 1)

```
set Twin Threshold Fraction = 0.25
```

Twin Directions File

```
set Twin Directions File = twinDirections.txt
```

Twin Normals File

```
set Twin Normals File = twinNormals.txt
```

If the kinematic hardening is enabled, one should define the kinematic hardening parameters C1 and C2 for the twinning systems.

C1 Twin Kinematic Hardening parameters

```
set C1 Twin Kinematic Hardening = 10.0, 10.0, 10.0, 10.0,
  10.0, 10.0
```

C2 Twin Kinematic Hardening parameters

```
set C2 Twin Kinematic Hardening = 1, 1, 1, 1, 1, 1
```

Defining the second phase in Multiphase problem

In the case of Multiphase problems, another set of material parameters should be defined after the first one, which is the primary phase (Phase 1). In this case, the phase number will be added to the input lines. For example, here, a very simple second phase is also defined. If user defined material model is used for this phase 2, it should be defined in the section User Defined Material Model Parameters as described above.

Elasticity parameters for phase 2

Input elastic stiffness matrix in Voigt notation as separate rows. The units are MPa.

Row 1

```
set Elastic Stiffness 2 row 1 = 170.0e3, 124.0e3, 124.0e3, 0,
  0, 0
```

Row 2

```
set Elastic Stiffness 2 row 2 = 124.0e3, 170.0e3, 124.0e3, 0,
  0, 0
```

Row 3

```
set Elastic Stiffness 2 row 3 = 124.0e3, 124.0e3, 170.0e3, 0,
  0, 0
```

Row 4

```
set Elastic Stiffness 2 row 4 = 0, 0, 0, 75.0e3, 0, 0
```

Row 5

```
set Elastic Stiffness 2 row 5 = 0, 0, 0, 0, 75.0e3, 0
```

Row 6

```
set Elastic Stiffness 2 row 6 = 0, 0, 0, 0, 0, 75.0e3
```

Slip parameters for phase 2

Number of Slip Systems for Phase 2

```
set Number of Slip Systems 2 = 12
```

Latent Hardening Ratio for Phase 2

```
set Latent Hardening Ratio 2 = 1
```

Initial slip resistances for Phase 2

```
set Initial Slip Resistance 2 =
  0.25,0.25,0.25,10,10,10,10,10,10,10,10,10
```

Hardening moduli of slip systems for Phase 2

```
set Initial Hardening Modulus 2 = 5.0, 5.0, 5.0, 100, 100.0,
  100.0, 100.0, 100.0, 100.0, 100.0, 100.0, 100.0
```

Power law coefficient in slip rate equation for Phase 2

```
set Power Law Exponent 2 = 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1
```

Saturation stress for Phase 2

```
set Saturation Stress 2 = 185.0, 185.0, 185.0, 160.0, 160.0,
  160.0, 160.0, 160.0, 160.0, 160.0, 160.0, 160.0
```

Slip Directions File for Phase 2

```
set Slip Directions File 2 = slipDirections.txt
```

Slip Normals File for Phase 2

```
set Slip Normals File 2 = slipNormals.txt
```

Building the library

Obtain the code from github repository:

```
git clone https://github.com/prisms-center/plasticity.git
```

Change directory into the newly cloned folder:

```
cd plasticity
```

Building the library:

```
cmake .
make -j 8
```

Build the crystal plasticity application:

```
cd applications/crystalPlasticity
cmake .
make release
```

Running an example simulation

```
cd fcc/compression
```

Running the compression example case:

```
mpirun -n 8 ../../main prm.in
```

Reference

M. Yaghoobi, S. Ganesan, S. Sundar, A. Lakshmanan, S. Rudraraju, J.E. Allison, V. Sundararaghavan, “PRISMS-Plasticity: An open-source crystal plasticity finite element software” Computational Materials Science 169 (2019) 109078.

M. Yaghoobi, J.E. Allison, V. Sundararaghavan, ”Multiscale modeling of twinning and detwinning behavior of HCP polycrystals” International Journal of Plasticity (2020), 102653.

C.P. Przybyla, ”Microstructure-sensitive extreme value probabilities of fatigue in advanced engineering alloys.” PhD Thesis, Georgia Institute of Technology (2010).

K.S. Stopka, M. Yaghoobi, J.E. Allison, and D.L. McDowell, ”Effects of boundary conditions on microstructure-sensitive fatigue crystal plasticity analysis,” Integr. Mater. Manuf. Innov., 10 (2021) 393.