PRISMS-Plasticity

Crystal Plasticity

Simple Tension example -Dual Phase microstructure

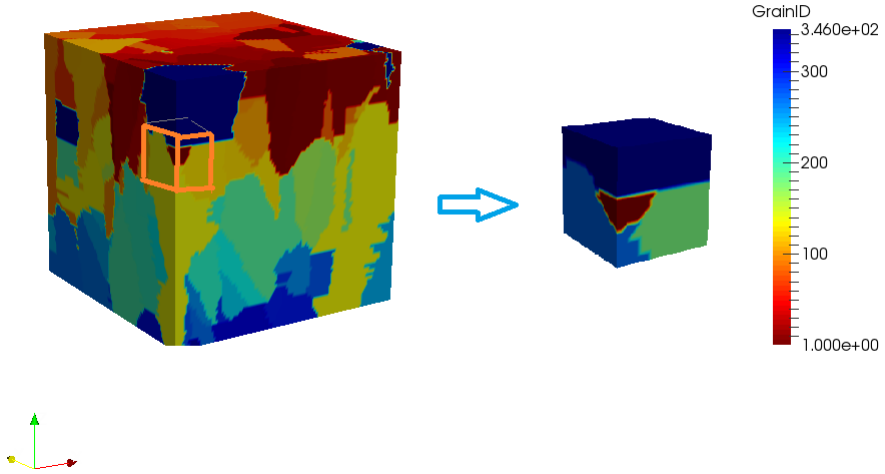

Figure 1: Input microstructure (3D Materials Atlas [2])

This is an illustrative example of a simple tension deformation problem for a dual-phase microstructure. A real microstructure was tested with the material parameters of fcc Copper which were obtained from [1] and AZ31 Mg alloy which were obtained from [3].

Input Parameters

```
/*FE parameters*/
#define feOrder 1 // Basis function interpolation order (1-linear)
#define quadOrder 2 // Quadrature point order n^3 (2->8 quadrature points)


/*Mesh parameters*/
//Set the length of the domain in all three dimensions
//Each axes spans from zero to the specified length
#define spanX 1.0
#define spanY 1.0
#define spanZ 1.0
// The number of elements in each direction is 2^(refineFactor) * subdivisions
// For optimal performance, use meshRefineFactor primarily to determine the
    element size
#define subdivisionsX 1
```

```
#define subdivisionsY 1
#define subdivisionsZ 1
#define meshRefineFactor 3 // 2^n*2^n*2^n elements(3->8*8*8 =512 elements)
#define writeMeshToEPS true //Only written for serial runs and if number of
    elements < 10000

/*Solution output parameters*/
#define writeOutput true // flag to write output vtu and pvtu files
#define outputDirectory "."
#define skipOutputSteps 0
#define output_Eqv_strain true
#define output_Eqv_stress true
#define output_Grain_ID true
#define output_Phase_ID true

/*Solver parameters*/
#define linearSolverType PETScWrappers::SolverCG // Type of linear solver
#define totalNumIncrements 100 // No. of increments
#define maxLinearSolverIterations 50000 // Maximum iterations for linear solver
#define relLinearSolverTolerance 1.0e-10 // Relative linear solver tolerance
#define maxNonLinearIterations 4 // Maximum no. of non-linear iterations
#define absNonLinearTolerance 1.0e-18 // Non-linear solver tolerance
#define relNonLinearTolerance 1.0e-3 // Relative non-linear solver tolerance
#define stopOnConvergenceFailure false // Flag to stop problem if convergence
    fails

/*Adaptive time-stepping parameters*/
#define enableAdaptiveTimeStepping false //Flag to enable adaptive time steps
#define adaptiveLoadStepFactor 0.5 // Load step factor
#define adaptiveLoadIncreaseFactor 1.25
#define succesiveIncForIncreasingTimeStep 10


#define multiplePhase true

//Phase1


//Elastic Parameters
double elasticStiffness1[6][6]={{170.0e3, 124.0e3, 124.0e3, 0, 0, 0},
    {124.0e3, 170.0e3, 124.0e3, 0, 0, 0},
    {124.0e3, 124.0e3, 170.0e3, 0, 0, 0},
    {0, 0, 0, 75.0e3, 0, 0},
    {0, 0, 0, 0, 75.0e3, 0},
    {0, 0, 0, 0, 0, 75.0e3}}; //    Elastic Stiffness Matrix -Voigt Notation (MPa)



//Crystal Plasticity parameters
#define numSlipSystems1 12 // generally 12 for FCC
#define latentHardeningRatio1 1.4 //q1
```

```
double initialSlipResistance1[numSlipSystems1]= {16.0, 16.0, 16.0, 16.0, 16.0,
    16.0, 16.0, 16.0, 16.0, 16.0, 16.0, 16.0}; //CRSS of the slip sytems
double initialHardeningModulus1[numSlipSystems1]= {180.0, 180.0, 180.0, 180.0,
    180.0, 180.0, 180.0, 180.0, 180.0, 180.0, 180.0, 180.0}; //Hardening moduli
    of slip systems
double powerLawExponent1[numSlipSystems1]= {2.25, 2.25, 2.25, 2.25, 2.25, 2.25,
    2.25, 2.25, 2.25, 2.25, 2.25, 2.25}; // Power law coefficient
double saturationStress1[numSlipSystems1]= {148.0, 148.0, 148.0, 148.0, 148.0,
    148.0, 148.0, 148.0, 148.0, 148.0, 148.0, 148.0}; // Saturation stress


//Slip systems files
#define slipDirectionsFile1 "slipDirections1.txt" // Slip Directions File
#define slipNormalsFile1 "slipNormals1.txt" // Slip Normals File



//Phase2

//Elastic Parameters
double elasticStiffness2[6][6]={{59.3e3, 25.7e3, 21.4e3, 0, 0, 0},
    {25.7e3, 59.3e3, 21.4e3, 0, 0, 0},
    {21.4e3, 21.4e3, 61.5e3, 0, 0, 0},
    {0, 0, 0, 16.4e3, 0, 0},
    {0, 0, 0, 0, 16.4e3, 0},
    {0, 0, 0, 0, 0, 16.8e3}}; //     Elastic Stiffness Matrix -Voigt Notation (MPa)


//Crystal Plasticity
//slip parameters
#define numSlipSystems2 18 // Total No. of slip systems (slip)
#define latentHardeningRatio2 1.4 //q1

double initialSlipResistance2[numSlipSystems2]= {25.0, 25.0, 25.0, 68.0, 68.0,
    68.0, 68.0, 68.0, 68.0, 68.0, 68.0, 68.0, 68.0, 68.0, 68.0, 68.0, 68.0,
    68.0}; //CRSS of slip sytems
double initialHardeningModulus2[numSlipSystems2]= {100.0, 100.0, 100.0, 130.0,
    130.0, 130.0, 130.0, 130.0, 130.0, 130.0, 130.0, 130.0, 130.0, 130.0,
    130.0, 130.0, 130.0}; //Hardening moduli of slip systems
double powerLawExponent2[numSlipSystems2]= {1.1, 1.1, 1.1, 0.8, 0.8, 0.8, 0.8,
    0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8}; // Power law
    coefficient
double saturationStress2[numSlipSystems2]= {70.0, 70.0, 70.0, 210.0, 210.0,
    210.0, 210.0, 210.0, 210.0, 210.0, 210.0, 210.0, 210.0, 210.0, 210.0,
    210.0, 210.0}; // Saturation stress

//Twin parameters

#define numTwinSystems 6 // No. of twin systems

double initialSlipResistanceTwin[numTwinSystems]= {40.0, 40.0, 40.0, 40.0, 40.0,
    40.0}; //CRSS of twin sytems
```

```
double initialHardeningModulusTwin[numTwinSystems]= {50.0, 50.0, 50.0, 50.0,
    50.0, 50.0}; //Hardening moduli of twin systems
double powerLawExponentTwin[numTwinSystems]= {1.1, 1.1, 1.1, 1.1, 1.1, 1.1};//
    Power law coefficient
double saturationStressTwin[numTwinSystems]= {50.0, 50.0, 50.0, 50.0, 50.0,
    50.0}; // Saturation stress

#define twinThresholdFraction 0.25 // threshold fraction of characteristic twin
    shear (<1)
#define twinSaturationFactor 0.25 // twin growth saturation factor
    (<(1-twinThresholdFraction))
#define twinShear 0.129 // characteristic twin shear

//Slip systems files
#define slipDirectionsFile2 "slipDirections2.txt" // Slip Directions File
#define slipNormalsFile2 "slipNormals2.txt" // Slip Normals File

//Twin systems files
#define twinDirectionsFile "twinDirections.txt" // Slip Directions File
#define twinNormalsFile "twinNormals.txt" // Slip Normals File




// Crystal Plasticity Constitutive model tolerances (for advanced users)
#define modelStressTolerance 1.0e-6 // Stress tolerance for the yield surface
    (MPa)
#define modelMaxSlipSearchIterations 1 // Maximum no. of active slip search
    iterations
#define modelMaxSolverIterations 3 // Maximum no. of iterations to achieve
    non-linear convergence
#define modelMaxPlasticSlipL2Norm 0.8 // L2-Norm of plastic slip strain-used for
    load-step adaptivity


//Read Input Microstructure
unsigned int numPts[3]={20, 20, 22}; // No. of voxels in x,y and z directions
#define grainIDFile "grainID.txt" // Grain ID File
#define headerLinesGrainIDFile 5 // No. of header Lines
#define grainOrientationsFile "orientations.txt" // Slip Normals File
```

References

[1] Anand, L., and M. Kothari. "A computational procedure for rate-independent crystal plasticity." Journal of the Mechanics and Physics of Solids 44.4 (1996): 525-558.

[2] 3D Materials Atlas AL6XN+Reconstruction

[3] Choi, S-H., et al. "Simulation of stress concentration in Mg alloys using the crystal plasticity finite element method." Acta Materialia 58.1 (2010): 320-329.

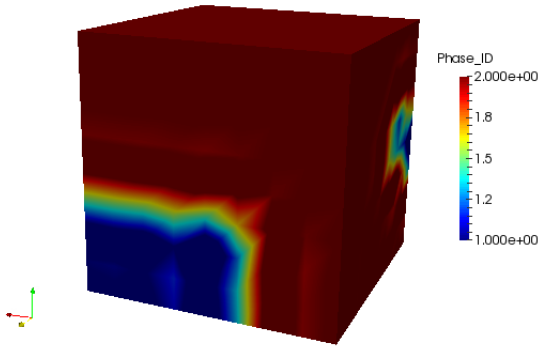

Figure 2: Dual-phase microstructure

Table 1: FCC Copper Slip Systems

System Number	Slip Direction	Slip Plane
1	[1 1 0]	(1 1 1)
2	[-1 0 1]	(1 1 1)
3	[0 1 -1]	(1 1 1)
4	[1 0 1]	(-1 1 1)
5	[-1 -1 0]	(-1 1 1)
6	[0 1 -1]	(-1 1 1)
7	[-1 0 1]	(1 -1 1)
8	[0 -1 -1]	(1 -1 1)
9	[1 1 0]	(1 -1 1)
10	[-1 1 0]	(-1 -1 1)
11	[1 0 1]	(-1 -1 1)
12	[0 -1 -1]	(-1 -1 1)

Table 2: HCP Magnesium Slip Systems

System Number	Slip Direction	Slip Plane
1	[1 1 -2 0]	(0 0 0 1)
2	[-2 1 1 0]	(0 0 0 1)
3	[1 -2 1 0]	(0 0 0 1)
4	[1 -2 1 0]	(1 0 -1 0)
5	[2 -1 -1 0]	(0 1 -1 0)
6	[1 1 -2 0]	(-1 1 0 0)
7	[1 -2 1 0]	(1 0 -1 1)
8	[-2 1 1 0]	(0 1 -1 1)
9	[-1 -1 2 0]	(-1 1 0 1)
10	[-1 2 -1 0]	(-1 0 1 1)
11	[2 -1 -1 0]	(0 -1 1 1)
12	[1 1 -2 0]	(1 -1 0 1)
13	[-1 -1 2 3]	(1 1 -2 2)
14	[1 -2 1 3]	(-1 2 -1 2)
15	[2 -1 -1 3]	(-2 1 1 2)
16	[1 1 -2 3]	(-1 -1 2 2)
17	[-1 2 -1 3]	(1 -2 1 2)
18	[-2 1 1 3]	(2 -1 -1 2)
19	[-1 0 1 1]	(1 0 -1 2)
20	[1 0 -1 1]	(-1 0 1 2)
21	[-1 1 0 1]	(1 -1 0 2)
22	[1 -1 0 1]	(-1 1 0 2)
23	[0 -1 1 1 1]	(0 1 -1 2)
24	[0 1 -1 1]	(0 -1 1 2)