Elle / FFT: Information on input and output files
Florian Steinbach

For any collaboration using Elle / FFT, please also contact Albert Griera at Universitat Autònoma de Barcelona

I cannot guarantee for a correct description of all files and have never used some of them, so please handle this documentation with care. I will only highlight the input numbers I used so far and that need changing for basic simulations. For beginners it is probably easier to leave the other values at default values.

1. Introduction

When starting an Elle/FFT simulation, we will need at least some files within the directory:

- The Elle file, storing crystal orientations in flynn attribute EULER_3 or in unodes, only if NO orientations are found in unodes, the code uses flynns and transfers the flynn orientations to the superimposed unodes. Also, the Elle file should contain the flynn attribute VISCOSITY, that acts as phase indicator. Counting of phases starts with 1, use consequtive numbers. There is a post-processing script that can transfer flynn to unode attribute for further use. But, do not store phases in unodes initially, the elle2fft and fft2elle codes need changing to make this working
- The ppc.in file that gives information about the simulation run
- The crystal files (.sx) that indicate crystal properties for each phase, we need as many .sx files as we have phases
- Actually optional, but almost impossible to do simulations without: A bash script (.sh) controlling the simulation

2. FFT input files

2.1 The phase or crystal files (.sx), available for calcite, halite, ice, olivine and quartz

```
SLIP SYSTEMS FOR ICE
HEX                icryst
   1.  1.  1.629   crystal axes (cdim(i))
   3               nmodesx (total # of modes listed in the file)
   3               nmodes  (# of modes to be used in the calculation)
   1  2  3         mode(i) (label of the modes to be used)
   BASAL SLIP
  1   3   3   1.0   0.0  0            modex,nsmx,nrsx,gamd0x,twshx,isectwx
  1.0  1.0   0.0   0.0  0.0           tau0xf,tau0xb,tau1x,thet0,thet1
  1.0   1.0   0.0   0.0               hselfx,hlatex
   0  0  0  1    1  1 -2  0
   0  0  0  1    1 -2  1  0
   0  0  0  1   -2  1  1  0
   PRISMATIC SLIP
  2    3   3  1.0   0.0  0           modex,nsmx,nrsx,gamd0x,twshx,isectwx
  20.0  20.0   0.0    0.0   0.0          tau0xf,tau0xb,tau1x,thet0,thet1
  1.0   1.0   0.0   0.0                hselfx,hlatex
   0  1 -1  0    2 -1 -1  0
   1  0 -1  0   -1  2 -1  0
   1 -1  0  0    1  1 -2  0
   PYRAMIDAL SLIP
  3   6   3   1.0   0.0   0           modex,nsmx,nrsx,gamd0x,twshx,isectwx
  20.0  20.0   0.0   0.0  0.0         tau0xf,tau0xb,tau1x,thet0,thet1
  1.0   1.0   0.0   0.0               hselfx,hlatex
   1  1 -2  2    1  1 -2 -3
  -1 -1  2  2   -1 -1  2 -3
   2 -1 -1  2    2 -1 -1 -3
  -2  1  1  2   -2  1  1 -3
   1 -2  1  2    1 -2  1 -3
  -1  2 -1  2   -1  2 -1 -3
```

Red circles: Critical resolved shear stresses (resistence to slip) on each slip system. For each system, keep the two values constant.

Green circles: Stress exponents

2.2 The ppc.in file

Change the 1st and 2nd line according to the number of phases. The “relative volume fraction” is actually not important, only make sure to type as many numbers as you have phases and make them add up to 1

The first blocks (“INFORMATION ABOUT PHASE”) are necessary to read information from .sx files, add blocks for more phases. Indicate the correct name of corresponding .sx file and the TOTAL number of unodes in Elle file
ATTENTION: The phases in Elle file stored in flynn attribute “VISCOSITY” will be assigned to these blocks, i.e. VISCOSITY=1, means use information in block 1, VISCOSITY=2, block 2 etc.

Below only change the “vel.grad-matrix” to indicate a velocity gradient (red circle) and the time (in secs) for one step(green cricle), which together define your incremental strain matrix during the simulation run. The values in this example will result in dextral simple shear with a shear strain increment of 0.02 per step. If using other Elle-processes make sure the time for FFT and Elle-processes is the same or scale the model accordingly

Change the number of iterations per step (“itmax”, blue circle) to decrease errors. To find the best solution between correctness and computation time, a value of 400 seems to be a good compromise, but probably needs to be increased or can be decreased in different kinds of setups

```
2                       number of phases (nph)
0.8  0.2                relative vol. fract. of phases (wph(i)) -- NOT USED, but make sure to have
as many numbers as phases and their sum is 1
*INFORMATION ABOUT PHASE #1
16384                   total number of unodes to read in filetext (not only this phase)
* name and path of texture file (filetext, as created by elle2fft)
make.out
* name and path of single crystal file (filecrys, .sx)
phase1.sx
1.0       FACT_BOUND factor of CRSS for special fourier points, leave at 1
*INFORMATION ABOUT PHASE #2
16384                   total number of unodes to read in filetext (not only this phase)
* name and path of texture file (filetext, as created by elle2fft)
make.out
* name and path of single crystal file (filecrys, .sx)
phase2.sx
1.0       FACT_BOUND factor of CRSS for special fourier points, leave at 1
*INFORMATION ABOUT TEST CONDITIONS
1.  1.  1.                      RVE dimensions (delt)
* boundary conditions
    1       1       1           iudot     |    flag for vel.grad.
    1       1       1                     |    (0:unknown-1:known)
    1       1       1                     |
                                  |
    0.     2e-12    0.          udot      |    vel.grad
    0.      0.      0.                    |
    0.      0.      0.                    |
                                  |
    0       0       0           iscau     |    flag for Cauchy
            0       0                     |
                    0                     |
                                  |
    0.      0.      0.          scauchy   |    Cauchy stress
            0.      0.                    |
                    0.                   @
* other
1e10      eqincr (if ictrl>=0) or tdot (if ictrl=-1)
-1              ictrl (1-6: strain comp, 0: VM eq, -1: tdot)
*INFORMATION ABOUT RUN CONDITIONS
1             nsteps
0.000000001    err
100            itmax
0              IRECOVER read grain states from STRESS.IN  (1) or not (0)?
1.0  99.0      xlfac0 (dum if irecover=1),xlfac1 (dum if irecover=0)
0              ISAVE write grain states in  STRESS.OUT (1) or not (0)?
0              IWRITEG write G*.OUT files (1) or not (0)
1              IUPDATE update tex & RVE dim and write TEX.OUT (1) or not
*additional parameters to estimate dislocation
0.01            length scale, keep the same than in Elle file
4.5e-10         burgers vector length
```

The last two lines need changing when using the dislocation density model. Make sure to use the same “length scale” than in the Elle file (in meter, scaling Elle units to real length units) and type the Burgers vector length in meter (assume it would be constant throughout the model...not really polyphase...)

2.3 The make.out file

See example below: Rows are for unodes, columns mean:
First row: Number of grains taken into account, here always zero, FFT only uses unodes
Column 1-3: Euler angles
Column 4-6: Position of Fourier point in i,j,k in computational grid
Column 7: If we track grains (flynns), the flynn ID, otherwise 0
Column 8: Phase ID, start counting with 1

→ Created by the elle2fft process, stores initial microstructure information BEFORE a FFT run
→ The first line is zero since flynns are not taken into account by FFT (but are still important for elle2fft and fft2elle!!). You can also add flynns (grains) here, which will cause FFT to output average strain rate and stress tensors per grain. However, the basecode would need chaning to do that and computation time is longer!

0

-39.2293	70.7095	92.7172	1	1	1	0	1
-35.1868	71.5467	93.8618	2	1	1	0	1
-39.1516	72.8912	91.8911	3	1	1	0	1
-32.9241	71.8924	92.8912	4	1	1	0	1
-40.7491	68.8923	90.1891	5	1	1	0	1

3. FFT output files

3.1 The all.out file

→ Storing information of bulk model box
→ First four lines:
1. Errors...Not really sure how to treat this. Actually the equations are solved iteratively upon convergence. This is a measure of how good the convergence is
2. SVM: Bulk von Mises stress, DVM: bulk von Mises strain rate, Wdot/DVM: rotation/bulk von Mises strain rate
3. Components of bulk stress tensor
4. Compotents of bulk strain rate tensor (actually dictated by what was the input in ppc.in)

→ "Activ per mode" indicates the slip system activities for the three slip systems. At the moment the code is optimized to work with three systems and would need updates to work with a different number

```
*****************************************************
 ERR s,d,eq, IT     = 0.272E-03 0.176E-03 0.200E-08 100
 SVM, DVM, Wdot/DVM =  0.221E-02  0.115E-11  0.000E+00
 S11,22,33,23,13,12 =  0.135E-04 -0.312E-03  0.299E-03  0.270E-04 -0.286E-03  0.120E-02
 D11,22,33,23,13,12 =  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.100E-11
 TAUs, TAUh         =  1.00000  1.00000
 EfVisc 1,2         =   0.0000000    0.0000000
 MacroOmega S,D     =  0.00000  0.00000
 Activ per mode     =  0.64795  0.16735  0.18469
*****************************************************
       XOR       MRK
 SEQ_1         SEQ_2         DEQ_1         DEQ_2
 S23_1         S23_2         S13_1         S13_2
 TAs_1         TAs_2         TAh_1         TAh_2
 GDs_1         GDs_2         GDh_1         GDh_2
```

3.2 The tex.out file

→ Containing information for every unode (or Fourier point)

→ Not displayed here, but contains in total 12 columns and as many rows as you have unodes. Each row represents information for one unode, the columns mean.

Column 1: Euler alpha
Column 2: Euler beta
Column 3: Euler gamma
Column 4: Phase ID, but does not work: For some reason it is always 1
Column 5: von Mises strain rate normalized to bulk value
Column 6: von Mises stress
Column 7: Basal activity
Column 8: Prismatic activity (pyramidal = 1-(basal+prismatic))
Column 9: GND[1] (fft2elle adds this to any pre-existing dislocation density)
Column 10: Not quite sure about this one, aparently it is rotation work...
Column 11: Fourier point ID (=unode ID + 1)
Column 12: If flynns or grains are tracked: Grain ID where the unode sits in

[1] GND means geometrically necessary dislocation density

3.3 The file unodeang.out:

→ Contains 4 columns and as many rows as you have unodes:
Column 1: Unode ID
Column 2: Euler alpha
Column 3: Euler beta
Column 4: Euler gamma

3.4 The file unodexyz.out:

Contains 4 columns and as many rows as you have unodes:

→ Stores the x,y,z (for 2D models do not worry about z too much) of each unode AFTER this step of deformation. ATTENTION: This position is only for this step of deformation and assumes that we started with a square grid: To track the real increments of deformation we need to track these values, which happens in U_FINITE_STRAIN attribute in the Elle file, when user switched this option on by indicating the number of unodes per dimension in the 3rd input of fft2elle:

FS_fft2elle -i file.elle -u bla bla DIMENSION (128,256 etc.)

→ Before you do the model you need to decide if you wish to track that, otherwise the information will be lost.

Column 1: Unode ID
Column 2: x position
Column 3: y position
Column 4: z position

3.5 The file temp-FFT.out

→ There is also one called temp.out, that is necessary as well for some reason beyond my knowledge, but this one is read by fft2elle. It is simple and looks s.th. like this:

```
 1.000000  1.000000  1.000000
 0.000000  0.000000  0.000000
 0.020000  0.000000  0.000000
```

First row stores model box sizes in x,y,z (changes only if there is a pure shear component in the strain tensor). I do not know what the 2nd line does, but the 1st value

3.6 The file err.out

→ I have never really used this file. It contains information for every iteration solving for the stress and strain tensors for each point in the computational grid during FFT. If you look in the file you will see that the columns are named according to what they are. Here is what I think what they mean (but I am not 100% sure):

IT: Iteration step
ERRD: I guess some error in strain rate field convergence
ERRS: The same like ERRD but for stress
DVM: Bulk von Mises strain rate
SVM: Bulk von Mises stress
WD/DVM: Work rate or so
EFFMU13: I am unsure...
SINH DINH: No idea
ACT ... : Slip system activities

4. GBM phase database (phase_db.txt)

The file “phase_db.txt” is an input to the polyphase GBM code, it was created by Jens Roessiger. It needs to be stored in the same folder from where the GBM code is called. Some blocks in it are outdated and not used by the code any more. Here, I highlight the most important ones. Remember that there is still a property indicating the number of phases (line 14), which is not displayed in the image below

```
29 #    turning it off means essentially dislocations are not used in the energy calcu
30 # G) Dislocation Density scaling along phase boundaries (0 ..... 1)
31 ############################################################
32 not used:
33 1 0 0 2e-9 3.6e-10 1 0
34 2 1 0 2e-9 0       0 0
35
36 ############################################################
37 ######          PHASE BOUNDARY PROPERTIES           ######
38 # Boundaries are defined by A and B
39 # A) Phase Number one
40 # B) Phase Number two
41 # C) Mobility of these boundary segments
42 #    below -10°C 7.5e-5 (Duval Book - Creep and Fracture of Ice)
43 #    above -10°C 1.0e-4 (P. Duval and O. Castelnau, Dynamic Recrystallization of Ic
44 #    (2.3 m4/s·J or m2/Kg·s Nasello et al. 2005 Intrinsic Mobility Mo; 2nd stage of
45 # D) Surface Energy of these segments (Jm-2)
46 # E) GB Activation Energy (Q) (J mol-1)
47 # mobil = mobility * exp( -(Q ) / ( R * T ) );
48 ############################################################
49
50 1 1 0.023  0.065  51.1e3   ice-ice
51 1 2 0.0023 0.52   51.1e3   ice-air
52 2 2 2.3    0.0032 51.1e3   air-air
53
54 ############################################################
55 ######          OUTDATED: MELT TRACKING             ######
56 # A) Use the Unode layer to track the given phase.
57 #    ( -1 --> do Diffusion shifting )
58 #    ( -2 --> do nothing )
59 #
60 ############################################################
61
62 -2
63
64 ############################################################
65 ######                VERBOSE STUFF                 ######
66 # A) print energies for given node (-1 = don't)
67 ############################################################
68 -1
69
70 ############################################################
71 ######          OUTDATED: CLUSTER_TRACKING          ######
72 ############################################################
73 # The cluster tracking multiplier energy function is
74 # defined by these values...
75 # area_percentage = (area_new) / (area_old)
76 # area_multiplier = A * (area_percentage -1) ^ D + B * (area_percentage - 1) + C
77 #
78 # A B C D
79 ############################################################
80 0.1 0 0 2
81
82 ############################################################
```

Important: Boundary properties !

The first red squares indicate energy per dislocation line (non-basal for ice Ih in this case). For the air-air phase (line 34), this is of course zero as there are no dislocations in air. The dotted red squares show scaling of dislocation energy within one phase (first value) and at a phase boundary (second value). The value is multiplied with the strain energy of the corresponding unode. As you can see: No strain energies at ice-air boundaries are used, although the dislocation density at the boundary may be high.