Running Analysis-Synthetic

1. **Open DREAM3D: Pipeline1-Analysis**
   1. Input: EBSD
      1. Enable Filter #1 .ctf (default)
      2. Enable Filter #2 for H5EBSD,
      3. Enable Filter #3 for .ang

\* Disable Filters accordingly

* 1. Change Input File name and location (Filter #1)
  2. Change Output File name and location (Filter #23,24, and 25)
  3. Start Pipeline
  4. Output:
     1. Stats Generator ODF Angle File (Into Pipeline2) as TXT
     2. Pole Figures for Original EBSD as PDF
     3. Feature Data for Original EBSD as CSV (Into Step 2)
     4. DREAM3D Data file with XDMF for Paraview

1. **Open Feature Data file (from 1-b-iii)** and copy EquivalentDiameter (Column J) to a separate csv file (Into LognormFit)
2. **Open MATLAB: LognormFit**
   1. Input: Line2: readmatrix(‘###InputFileFromStep2###.csv’)
   2. Run MATLAB
   3. Output:
      1. Histogram plot of EquivalentDiameter.
      2. Parmhat 1st value: Mu (Into Step 4-a-ii)
      3. Parmhat 2nd value: Sigma (Into Step 4-a-ii)
3. **Open DREAM3D: Pipeline2-StatsGenerator**
   1. Input for StatsGenerator (Filter #1)
      1. Input Stats Generator ODF Angle File TXT file under “ODF Tab”-Bulk Load From File.
      2. Input Mu and Sigma (From Step 3-c-ii & iii)
      3. Adjust Bin Step Size according to how many features there are.
      4. Adjust Min and Max Cut Off according to Min and Max Feature size.
      5. Preset Statistic Models:

**HAVE TO CHANGE EVERYTIME YOU OPEN (quirk of Dream3D)**

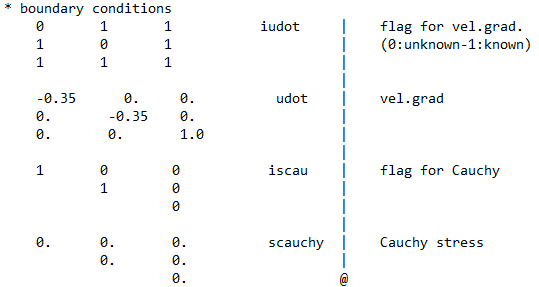
\* B/A and C/A is 3D ellipsoid dimensions ratio of major axis/minor axis from 2 perpendicular planes.

* + - 1. **Equiaxed**: B/A =1 and C/A = 1
      2. **Rolled:** I used ImageJ to determine average dimensions of ellipsoids to determine B/A and C/A.
    1. Stats Generator “Axis ODF” Tab:
       1. + button,
       2. Euler 1, 2, and 3 dependents on how you want to orient the final synthetic volume,
       3. Weight 100000, Sigma 1, Sample Points of 5000.
    2. Run “Create Data”
  1. Input for Initialize Synthetic Volume (Filter #2)

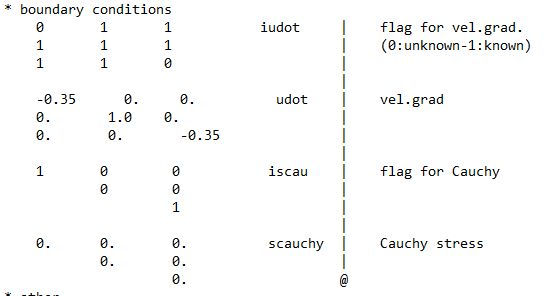
\* Whatever size and resolution make sure the Estimated Primary Features is numerous, i.e. more grains. I usually go with Resolution of 10.

* 1. Change Output File name and location (Filter #8, 9, and 10)
  2. Start Pipeline
  3. Output:
     1. Pole Figures for Synthetic EBSD as PDF
     2. Feature Data for Synthetic EBSD as CSV
     3. DREAM3D Data file with XDMF for Paraview

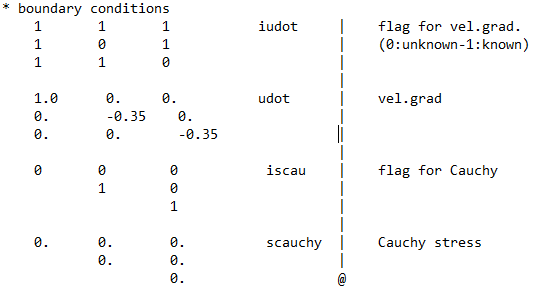
1. **Open DREAM3D: Pipeline3-AbaqusGenerator**
   1. Input: DREAM3D file from Step 1-e-iv (Original) or from Step 4-e-iii (Synthetic)
   2. Change Input File name and location (Filter #1)
   3. Change Output File name and location (Filter #4 and 5)
   4. Output:
      1. Abaqus Surface Mesh as INP
      2. DREAM3D Data file with XDMF for Paraview
2. **Open DREAM3D: Pipeline3-FFTGenerator**
   1. Input: DREAM3D file from Step 1-e-iv (Original) or from Step 4-e-iii (Synthetic)
   2. Change Input File name and location (Filter #1)
   3. Change Output File name and location (Filter #7, 8 and 9)
   4. Output:
      1. Feature Data for Original or Synthetic EBSD as CSV
      2. Los Alamos FFT for EVPFFT as TXT (Into Step 7-b)
      3. DREAM3D Data file with XDMF for Paraview
3. **Open Python: CutTXTFFT**
   1. Line 1-6: Change dimension/location of subvolume within Original or Synthetic volume.
   2. Change Input File name (Line 8, from Step 7-b)
   3. Change Output File name (Line 7)
   4. Run Python
   5. Copy File from Step 7-c to EVPFFT folder
4. **EVPFFT:**
   1. Input: TXFFT from Step 7
   2. Open fft.dim with text editor, change line 5 to the appropriate txfft file dimension
   3. Open fft.in with text editor (X, Y, Z according to load directions)
      1. change line 2 to the appropriate number of points (i.e., 262144 for 64x64x64)
      2. change line 5 to the appropriate name of input fft file
      3. boundary conditions:
         1. Z-load:



* + - 1. Y-load:



* + - 1. X-load:



* + 1. eqincr \* nsteps = max strain. (Default 0.05 or 5% strain)
       1. eqincr = 0.0005
       2. nsteps = 100
  1. Run EVPFFT: (gfortran evpnew10\_X.for or \_Y or \_Z according to load directions) and (./a.out)
  2. Output: dfield.out, efield.out, elfield.out, sfield.out, and str\_str.out

1. **EVPFFT Post processing**
   1. Copy dfield.out, efield.out, elfield.out, sfield.out to “post” folder
   2. Open vtkgbnew2.for with text editor
      1. Change line 2 to the appropriate dimensions
      2. Use vtkgbnew2\_X or \_Y or \_Z according to load directions
   3. Run vtkgbnew2 (gfortran vtkgbnew2\_X.for or \_Y or \_Z and ./a.out)
   4. Output: dfield.vtk, efield.vtk, elfield.vtk, sfield.vtk, gb.vtk, and grains.vtk
      1. Use dfield.vtk/efield.vtk/elfield.vtk/sfield.vtk/grains.vtk AND gb.vtk to plot in Paraview.
2. Stress\_Strain\_Plot.py (Plot Single Direction Load)
3. XYZ\_Load\_Plot.py (Plot XYZ Direction Load)