**FROM ABDU**

1. Deconstruct the lattice using 'deconstructedLattice.m'.
   1. The input Inputs required are:
      1. 3D volumetric binary image
      2. Standard lattice input file, [numNodes,numStruts; nodeXYZ nodeDiam;strutConn strutDiam]
      3. CT image size
   2. Save the output data as ‘fcc\_Deconstructe.mat or fccz\_Deconstructe.mat
2. Extract the struts using the code ‘getLatticeStruts\_Latest’.

A close-up of a statue

Description automatically generated with medium confidence

1. Once the individual struts are extracted, the centroid of the middle strut is located and then cut the connected struts.
2. The next step is to import the extracted struts to Abaqus for simulation
3. Then the results of the simulation is converted into colour map and standard deviation

**FROM BILL**

The sub-directory "1.Extracting Geom" contains:

* "fcc\_deconstructed.mat" and "fccz\_deconstructed.mat" these are instances of the class "deconstructedLattice" which is defined in the "deconstructedLattice.m" file within the folder. If you put these 3 three files in the same dir and open in matlab you should be able to double click or "load( "fcc\_deconstructed.mat")". Once each of these variables are loaded you can browse whats inside the instance. It contains alot of data (from memory) including CT data, registered node centres etc. All the stuff that is required to locate the centre of the nodes within a ct scanned dataset.
* "getLatticeStruts\_latest.m" shows how an instance of the "deconstructedLattice" class is created
  + fccDeconstructed=deconstructedLattice(idealLattice,lattice,ctRes,additionalStrutPoints,maxCTPointCloudPoints,idealBoundaryPoints);
  + In the first couple lines of the code it shows the inputs. Its basically PLG custom files, lattice CT image stacks, etc. Can explain more in another email.
* "extractNodes.m" is basically an example of how nodes including their attached struts can be individually extracted from the instances of the deconstructedLattice saved in the folder ( "fcc\_deconstructed.mat" and "fccz\_deconstructed.mat" )

In the "V2\_ProcessStruts" directory:

* I think each sub folder is a step in the process  "GetIndividualStrutsRaw" :
  + Has a file "getIndividualStrutsV2.m" which uses the data in the "deconstructedLattice" instances  ( "fcc\_deconstructed.mat" and "fccz\_deconstructed.mat" )  and basically loops through each of the struts that have been located in the CT data set and extracts them by masking sections/creating ROIs in the lattice CT scan.
  + "FCC\_0.3\_5x5x5\_2x2x2\_strutsRaw.mat" and  "FCCZ\_0.3\_5x5x5\_2x2x2\_strutsRaw.mat"  are created by the above "getIndividualStrutsV2.m" script, depending on which "deconstructedLattice" instances is loaded
* The second step in the process is probably the subfolder "Process Struts":
  + Contains the file "processStruts.m" I believe this does all the trimming/ creating the "dumbbell" shape of the strut and saves the "FCC\_0.3\_5x5x5\_2x2x2\_strutsProcessed.mat" or "FCCZ\_0.3\_5x5x5\_2x2x2\_strutsProcessed.mat" depending on inputs.
* The final step is "Voxel2Inp" directory:
  + Contains "Voxel2Inp.m" that loads the "\_strutsProcessed.mat" and converts the binary image stacks in to input files that apply a concentrated unit load to the top and encastre bc at bottom of face of individual strut. I think that is an example code that I had sent to abdu showing how it could be structed. I believe it also runs the inp in abaqus.