Orriginal author: Matthew McMillan,

DATE: 06/11/17

description

This program takes a variety ofinputs and writes a lattice file of various output formats for use in other programs. The use of the PLGBatch file simplifies the creation of lattice(s). Example implementations are presented at the end of this file

Methods

The following is a list of methods and a quick overview of their function, it should obvious from the name what it is they do.

- PLG: Creates a PLG object based on inputs below. if running with 15 inputs the next four functions are called as sub functions.
- latticeGenerate: generates a lattice unit cell based on inputs
- cellReplication: replicates a unit cell based on inputs
- addDiams: add unique diameters for each strut and sphere
- cleanLattice: removes duplicate vertices or faces and orders the vertices in Z
- load: loads an existing custom input file.
- translate: translates the lattice structure.
- rotate: rotates the lattice structure
- plot: displays a rendering of the beam model that represents the lattice.
- splitStruts: New method that splits strut intersections that are not already defined and then creates a new lattice to replace the old. This method is not thoroughly tested and should be considered buggy. This method calls many subfunctions stored in their own method group.
- save: this method is in its own method group and each sub save method will be named according to its save out type. Eg saveStl saves out stl format.

generating a unit cell

To generate a unit cell you must define a new subfunction under the "cell type methods group". The nodes are the coordinates and the faces are the connections between defined coordinates. This new function must then be added to the switch case under latticeGenerate.

Note that not all unit cells are thoroughly tested.

Inputs

- **1 input -** File location of a .custom or csv file containing a custom set of vertices and connections (faces).
- **15 inputs -** See code for details however 15 inputs are required when generating a regular lattice from scratch.

other

- most up to date non class based version in archive next to git origin location.

- pull PLGgit to get updates with commits.

- a current version of the class should be present if not then pull the get repository.

Examples

A 3x4x5 BCC lattice with a 0.3mm strut diameter, 4mm unit cell and 0.5mm ball diameter with its origin moved by 6,7,8 and then saved as a stl (12 facet resolution) and abaqus input file

```
obj = PLG('bcc',12,0.3,...
1,0.5,12,4,4,4,...
3,4,5,6,7,8);
saveStl(obj,fileName,folderName);
saveAbaqus(obj,fileName,folderName);
```