Orriginal author: Matthew McMillan, DATE: 07/05/18

description

This program takes a variety of inputs 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 is present in the complex example 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, 1 input loads a custom file.
- set: used to set parameters in the PLG using name value pairs
- defineUnit: used to define the individual unit cell
- latticeGenerate: generates a lattice unit cell based on inputs
- cellReplication: replicates a unit cell based on inputs
- cleanLattice: removes duplicate vertices or faces and orders the vertices in Z
- scale: resize the existing structure
- translate: translates the lattice structure.
- rotate: rotates the lattice structure
- plus: combines an existing PLG object into another plg object usefull for generating complex lattice shapes
- plot: displays a rendering of the beam model that represents the lattice.
- 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

See subfolder unitCell in the PLG code

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 git repository.

Examples

A 3x4x5 BCC lattice with x struts, 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 3mf file with a resoultion of 30. See complex example for the use of translation, roration and plus.

```
obj = PLG();
obj = set(obj,'resolution',12);
obj = set(obj,'strutDiameter',0.3);
obj = set(obj,'unitSize',[4,4,4]);
obj = set('sphereAddition',true);
obj = set('sphereDiameter',0.5);
```

```
obj = set(obj,'sphereDiameter',0.5);
obj = set(obj,'origin',[6,7,8]);
obj = set(obj,'replications',[3,4,5]);

obj = defineUnit(obj,{'bcc','xRods'});
obj = cellReplication(obj);
obj = cleanLattice(obj);

saveStl(obj,'exampleOut.stl');

obj = set(obj,'resolution',30);
obj = defineUnit(obj,{'bcc','xRods'});
obj = cellReplication(obj);
obj = cleanLattice(obj);
save3mf(obj,'exampleOut.3mf');
```

SubClass addSupport

This class is a subclass for the PLG method that adds support structures to a existing custom/xlsx file the initial call determines which vertices to add as supports based on incline and distance (relative to total height) from the lowest vertex. the padSupport creates vertical rods with a given diameter essentially rising the structure up on pin supports.

SubClass splitStrut

Enables splitting of a bad custom file where beam do interesect but this is not present in the file. splitStruts will identify these and split the beams in two.