Operation of the FLatt Pack program

I. Maskery

August 26, 2021

1 The FLatt Pack GUI

This section provides a guide to the use of FLatt Pack, beginning with program installation and ending with the creation of output files representing a lattice structure. Where relevant, and where this aids the efficient use of FLatt Pack, explanations for the inclusion of certain GUI elements are given.

After installation (described in section 1.1) FLatt Pack presents the user with a sequence of windows corresponding to the lattice design steps. These include:

1. Designating the lattice geometry.
2. Choosing the lattice dimensions and cell repetitions.
3. Choosing the cell type and orientation.
4. Specifying the lattice volume fraction (with options for volume fraction grading).
5. Choosing to add an exterior 'skin' to the lattice.
6. Specifying the output filename and file types.
7. Inspecting the lattice design with a 3D graphical preview.
8. Exporting the lattice geometry in the specified file types.

The FLatt Pack GUI windows relating to the above steps are illustrated and described in the following sections, as are several optional subwindows and two preliminary windows which convey general information on the program's operation. The user is able to move back and forth through these windows, and therefore revise previous design choices, using the '< Back' and 'Next >' buttons found in the lower right of each window. Each window also features a 'Cancel' button which, like the '×' in the upper right corner of the window, closes the program.

1.1 Installation and system requirements

FLatt Pack is written in Matlab and compiled into a standalone application using the Matlab Compiler. As such, neither a Matlab installation, nor a Matlab license, are required to run FLatt Pack on a host PC. Its only dependency is Matlab Runtime, a set of shared libraries which is automatically installed by the FLatt Pack installer or can be obtained freely from the Mathworks website (www.mathworks.com/products/compiler/matlab-runtime.html).

The FLatt Pack installer is freely available via a public Github repository (www.github.com/ian27ax/FLatt_Pack_dist). This provides access to the most recent version of the software, which is frequently updated with new features and bug fixes. Users may report issues and feature requests through the Github tracker. The installer (figure 1) guides the user through several steps (selecting the installation folder, etc.), including the installation of Matlab Runtime if it is not detected on the host PC. An internet connection is therefore required for the installation of FLatt Pack; if this is not available, the user can contact the corresponding author to obtain a copy of the FLatt Pack installer with Runtime included.

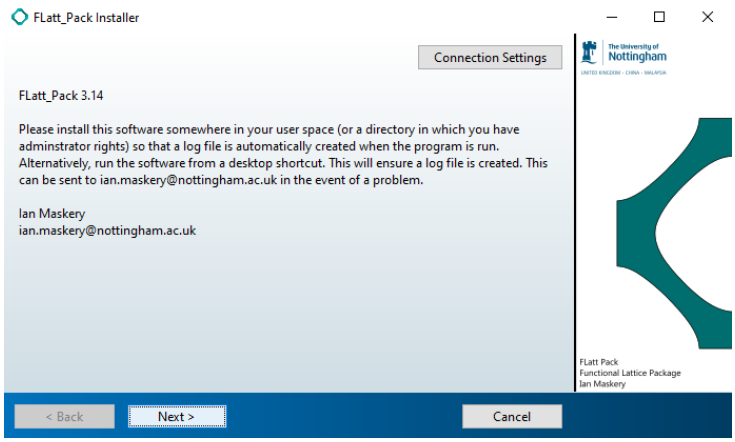

Figure 1: FLatt Pack Installer.

The host PC should have a 64-bit Windows operating system. There are no specific system requirements regard-

ing memory, or CPU type or speed, but the performance of FLatt Pack will be adversely affected if insufficient memory is available to store the numerical arrays generated during its operation. It is recommended that the host PC has at least 16 GB of RAM.

1.2 User agreement and program outline

The first window of the FLatt Pack GUI is a user agreement notification (figure 2), which the user should read before proceeding. The 'User agreement' window also contains buttons to open a new email to the corresponding author, for general queries and bug reporting, and a browser window to the corresponding author's website, where publications related to lattice design are listed. There is also hover-over text which informs the user how to cite the software, should it be used in future work.

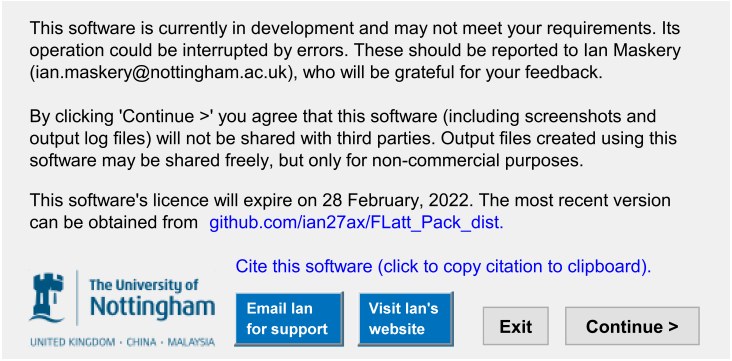

Figure 2: FLatt Pack GUI - User agreement.

FLatt Pack is usually distributed with a time-limited license. The end date of the license is given in the 'User agreement' window, after which FLatt Pack will cease to work and the user will be prompted to contact the corresponding author or obtain a new version of the program from the Github repository. This is principally to ensure that users have access to the most recent version of FLatt Pack, including any bug fixes or performance improvements made since the user's version was compiled.

Following the user agreement, the user is shown the FLatt Pack 'Program outline' (figure 3), which provides a simple description of the lattice design steps. These are broadly equivalent to those listed above, but with wording which emphasises the creation of STL files for AM.

1.3 Geometry selection

The user is next presented with the 'Geometry' window (figure 4). Using either the clickable figures toward the bottom of the window, or the radio buttons in the upper

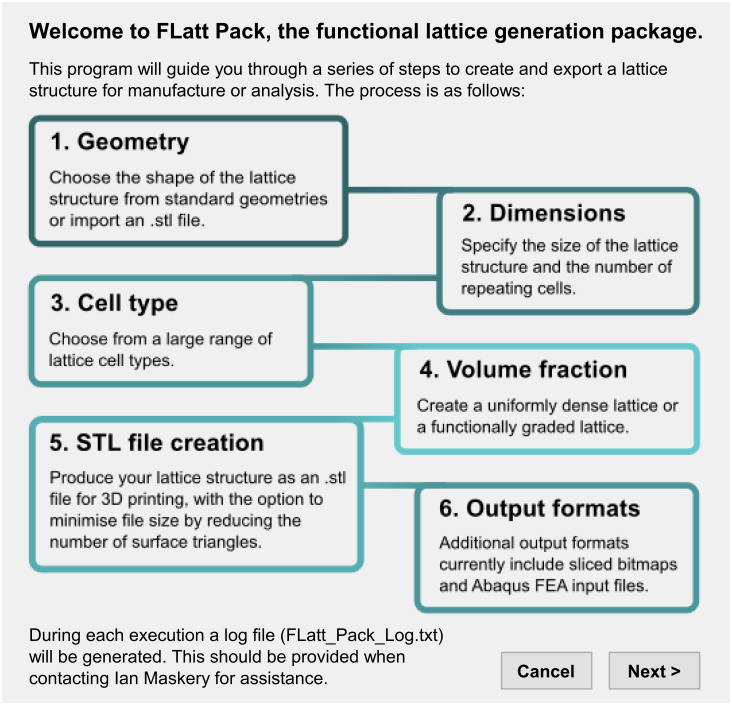

Figure 3: FLatt Pack GUI - Program outline.

left, the user can select between three primitive geometries for the lattice; a cuboid, a cylinder or a spheroid, or can select the 'Custom' option to import a 3D geometry in the form of an STL file. This STL mesh will then define the design space in which the lattice is generated.

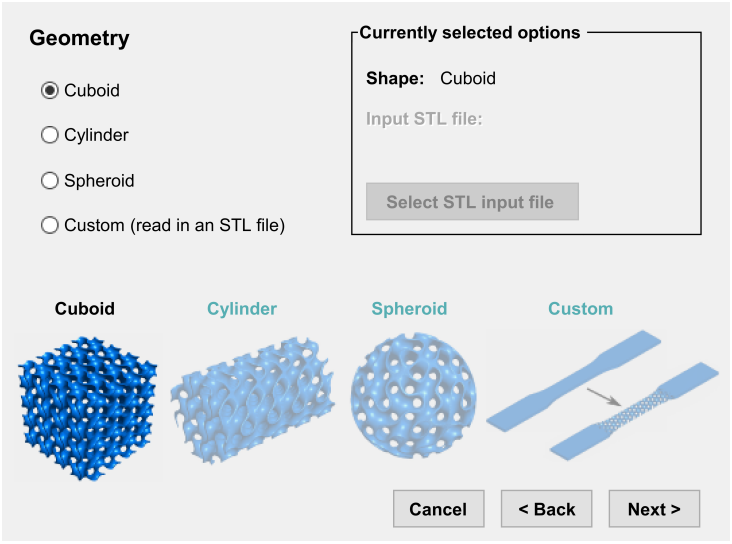

Figure 4: FLatt Pack GUI - Geometry.

As the user selects between the 'Cuboid', 'Cylinder', 'Spheroid' and 'Custom' options, the relevant figure is highlighted and the others are dimmed to illustrate the current choice. This feature is present in several of the

FLatt Pack GUI windows, and is used to make the lattice design process as clear as possible.

1.4 Dimensions and cell repetitions

After choosing the lattice geometry, the user is asked to specify the dimensions and cell repetitions along each direction. The 'Dimensions and cell repetitions' window (figure 5) provides editable text boxes for dimensions (in mm) along the x, y and z directions, as well as numbers of cells along those directions. The user can ensure the generation of cubic lattice cells by maintaining an equivalent ratio between the dimension and cell numbers in each direction; e.g., in figure 5, 30:3 in x, 40:4 in y and 50:5 in z. However, lattice cells of any aspect ratio are permitted; these generally result in cells which appear 'stretched' or 'squashed' compared to their conventional cubic forms; see figure 6 for example.

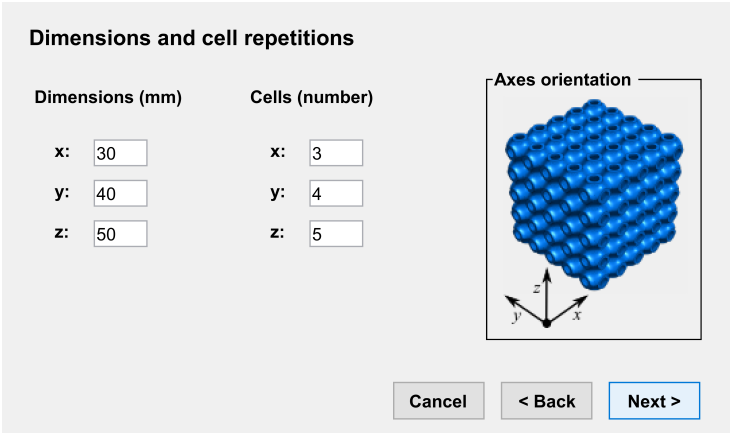

Figure 5: FLatt Pack GUI - Dimensions and cell repetitions.

If the user opted to import an STL mesh as the basis for the lattice geometry, the dimensions in the 'Dimensions and cell repetitions' window will be non-editable; they will be determined directly from the coordinates of the input STL mesh vertices. However, the user still has control over the numbers of lattice cells.

1.5 Cell type selection

In the 'Cell type' window (figure 7), the user is presented in the upper left with a choice of four 'classes' (or 'phases'). These represent distinct types of lattice structure; selecting between them alters the range of cell types shown in the central section of the window. There are eight cell types available for both the 'Network' and 'Matrix' class; these are the TPMS gyroid, lidinoid, O,C-TO, I-WP, diamond, primitive, neovius and split-P. Five cell types are available for the 'Honeycomb' class; these are honeycomb versions of the gyroid, primitive, I-WP, lidinoid and diamond cell types. In the 'Strut' class there are two cell types; body-centred-cubic and simple cubic.

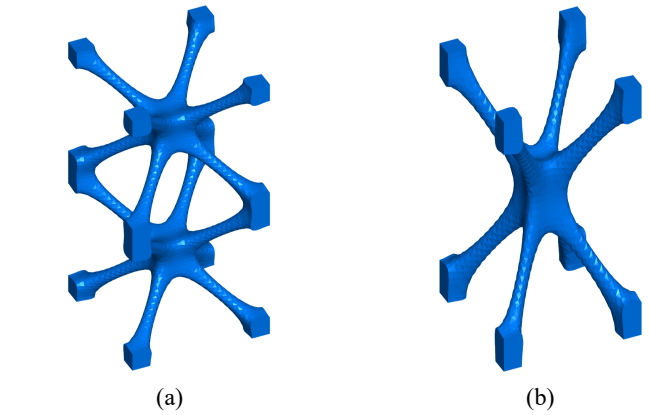

Figure 6: A lattice of $1 \times 1 \times 2$ body-centred-cubic cells is shown in (a), while (b) shows a single unit cell with the same bounding dimensions. The result is a modified cell geometry which is 'stretched' in the z direction compared to its conventional form.

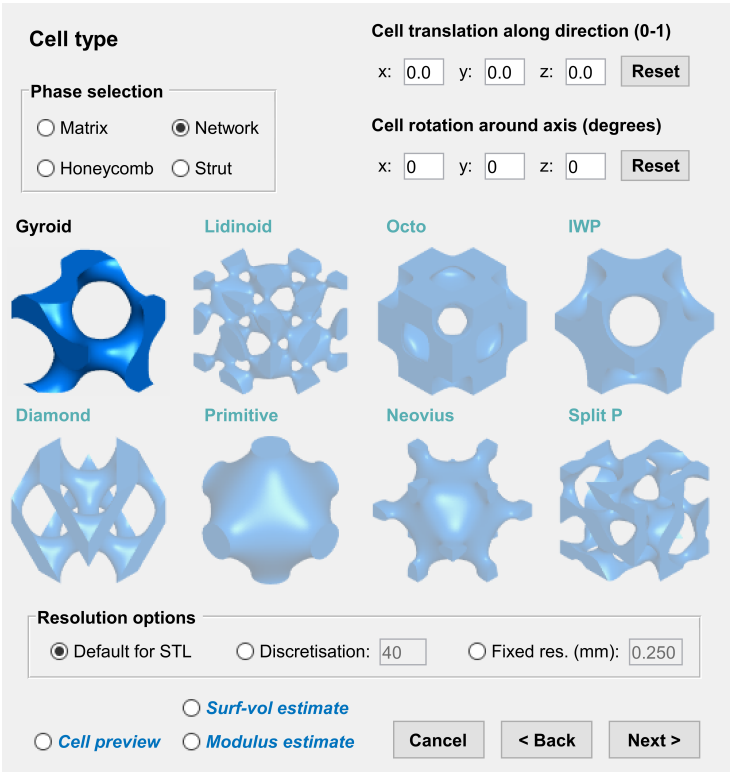

Figure 7: FLatt Pack GUI - Cell type.

Once the class and cell type are chosen, the user may translate the unit cell in the x, y and z directions and ro-

tate it about each axis (applied in the order z, y, x) using the options in the upper right of the 'Cell type' window. The facility to orient the struts or walls of the cell is included so that the designer may align the high-stiffness direction of a lattice with the direction of an applied load, resulting in minimal deformation.

The 'Cell type' window also contains 'Reset' buttons for cell translation and rotation. These set the editable text box values for translation and rotation to '0', thus restoring the *unmodified* version of the chosen cell type.

The effects of translation and rotation are best represented using the 'Cell preview' radio button in the lower left of the the 'Cell type' window. This opens the 'Cell preview' subwindow (figure 8), featuring a 3D preview of the chosen cell type. The preview shows a small lattice of 2 or 4 cell tessellations. Cell translations and rotations are applied in this subwindow in real-time, allowing the user to see the effect of their choices before proceeding. Furthermore, the vertical slider to the right of this subwindow allows the user to see the effect of modifying the volume fraction of their chosen cell type.

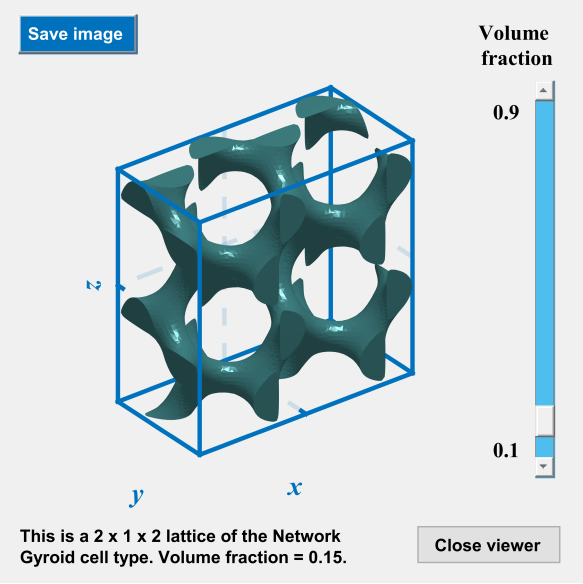

Figure 8: FLatt Pack GUI - Cell preview.

Three further elements of the 'Cell type' window merit attention. The first is the 'Modulus estimate' radio button, which opens a subwindow of the same name (figure 9). This provides a plot of relative elastic modulus against volume fraction for the currently selected cell type. Three curves are given, corresponding to modulus estimates along the crystallographic [001], [111] and [110] directions. The 'Modulus estimate' subwindow therefore provides an estimate of the lattice elastic modulus and anisotropy, through comparison of the [001], [111] and [110] curves. This information allows the user to select the cell type, volume fraction and orientation most suitable for the intended application of the lattice specimen or component.

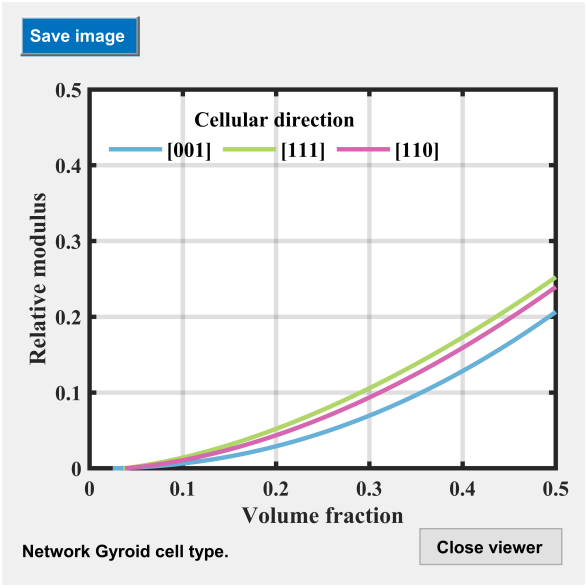

Figure 9: FLatt Pack GUI - Modulus estimate.

Second, the user may select the 'Surf-vol estimate' radio button, which opens a separate subwindow with a contour plot of surface-to-volume ratio for the currently selected cell type, across a range of cell sizes and volume fractions (figure 10). This is designed to provide an easily interpretable guide to how the surface-to-volume ratio is affected by the user's lattice design choices. For example, the user can easily compare the contour plots of several cell types, and proceed to choose the cell type with maximal or minimal surface area, according to the demands of the lattice application.

The last element of note in the 'Cell type' window is the box situated below the cell illustrations which contains three 'Resolution options'. These define the resolution of the design space mesh. Editable text boxes allow the user to specify the resolution, either in distance units using the 'Fixed res. (mm)' option, or by prescribing the number of increments, μ, along each axis in a unit cell using the 'Discretisation' option. For example, a discretisation setting of 40 means a single unit cell is generated in a numerical array of $40 \times 40 \times 40$ elements; a lattice of $n_x \times n_y \times n_z$ cells is generated in an array of size $40n_x \times 40n_y \times 40n_z$.

The 'Fixed res. (mm)' and 'Discretisation' options are included to allow the user to precisely control the resolution of voxel meshes and hexahedral FE meshes made

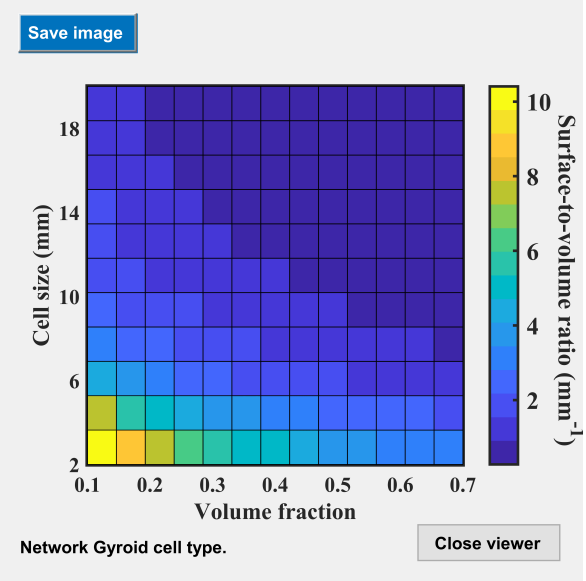

Figure 10: FLatt Pack GUI - Surface-to-volume ratio estimate.

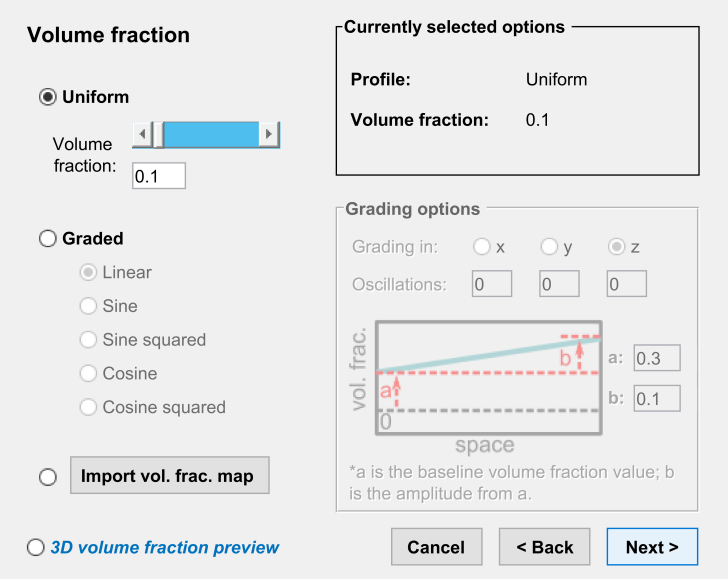

Figure 11: FLatt Pack GUI - Volume fraction.

using FLatt Pack. For the creation of STL files, the user is encouraged to select the 'Default for STL' option, which provides a spatial resolution sufficient to render the geometry of the chosen cell accurately in the output STL file, without the resolution being *overspecified*. The 'Default for STL' option aims to avoid the generation of unnecessarily large numerical arrays representing the lattice design space, therefore reducing FLatt Pack's memory consumption and execution time compared to a 'one resolution for all lattices' approach.

1.6 Volume fraction selection

The 'Volume fraction' window (figure 11) presents options to create a lattice of uniform volume fraction, graded volume fraction (with some inbuilt mathematical functions for grading) or custom volume fraction, using an input file to describe the volume fraction distribution. The choice is made using the 'Uniform' and 'Graded' radio buttons, as well as the 'Import vol. frac. map' button, in the lower left of the window. A box in the upper right of the 'Volume fraction' window shows the user's currently selected options.

Selecting the 'Uniform' option, a slider and editable text box are highlighted, allowing the user to specify a volume fraction for the whole lattice structure. Volume fraction takes values between 0 and 1, representing an entirely void or solid design space, respectively. Internally in FLatt Pack, a volume fraction threshold of 0.9 is used to prevent the creation of almost completely solid structures with separate enclosed voids. These should be avoided in general due to the inability to remove trapped excess material following many AM processes.

A limit is also placed on the lowest accessible volume fraction for each cell type. This is done for two reasons:

1. Low volume fraction cells possess small geometrical features; i.e., thin walls or struts. This necessitates the generation of very large Matlab arrays in order to render the cell geometry accurately. Therefore, limiting the lowest volume fraction available to the user helps minimise FLatt Pack's memory consumption and run time, whilst not overly restricting the user's design choices.

2. Some cell types, for example the network phase primitive cell, cease to have structural connectivity below a certain volume fraction. Limiting the lowest available volume fraction in these cases ensures that the lattice remains structurally intact, can be manufactured and is capable of supporting an applied load.

The 'Graded' volume fraction option activates a range of grading function radio buttons ('Linear', 'Sine', 'Sine squared', 'Cosine' and 'Cosine squared') and the 'Grading options' box on the right of the window. These allow the user to choose between linear volume fraction grading, or grading based on sine or cosine functions. The 'Grading options' box has editable text boxes to specify the minimum and maximum volume fractions. The user must also specify in which direction, or directions, the grading should occur and, for the trigonometric functions, the

number of oscillations (complete sine or cosine waves) in each direction.

Figures 12 and 13 illustrate FLatt Pack's inbuilt volume fraction grading options. In figure 12, the volume fraction is given by a linear gradient in the z direction between values of 0.1 at the base of the lattice design space, and 0.9 at the top. In figure 13, the grading is given by one complete sine squared function in both the x and z directions, again between values of 0.1 and 0.9.

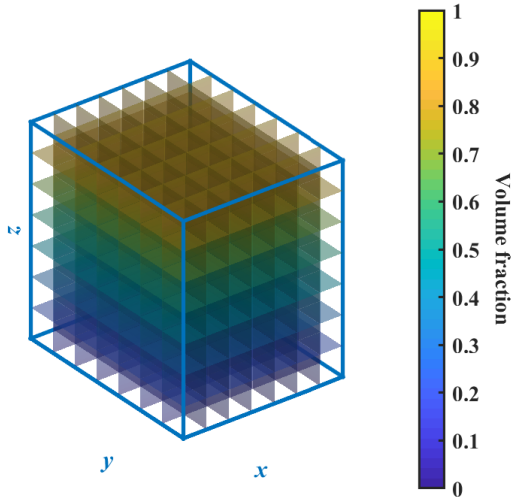

Figure 12: Volume fraction preview (linear grading).

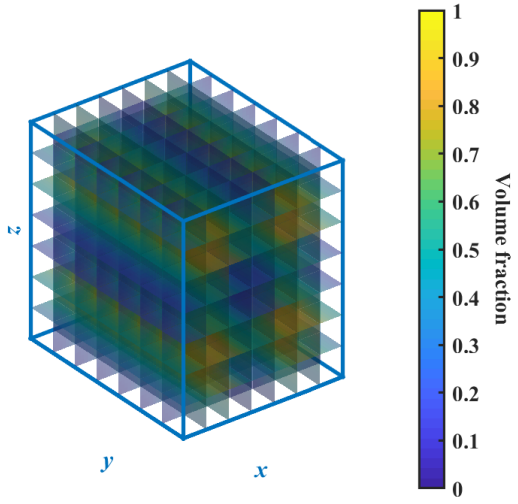

Figure 13: Volume fraction preview (sine squared grading).

The 'Import vol. frac. map' button allows the user to specify the volume fraction in the lattice design space by importing a file. FLatt Pack currently supports two input file types for this purpose:

1. A Matlab array (.mat) representing the 3D lattice design space, where each element is taken to represent a small volume of space and has a value between 0 (void) and 1 (solid).

2. A 24-bit or monochrome bitmap image (.bmp), where each pixel has a colour between black (r g b = 0 0 0), corresponding to void in the lattice design space, and white (r g b = 255 255 255), corresponding to solid. The bitmap is taken to represent the volume fraction distribution in the xy plane, with the full 3D lattice design space then being constructed by replicating that plane throughout the z direction.

Finally, the 'Volume fraction' window contains a radio button in the lower left corner to provide a '3D volume fraction preview'. This opens a subwindow (figure 14) showing a 3D preview of the volume fraction distribution in the lattice design space, and is updated in real-time according to the user's choices.

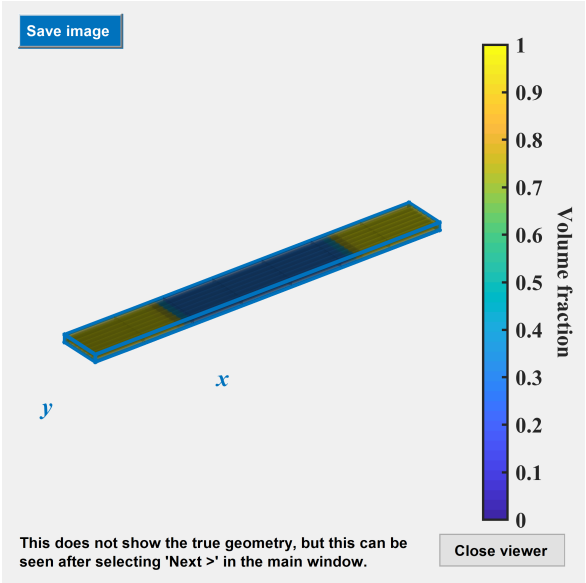

Figure 14: FLatt Pack GUI - Volume fraction preview (custom volume fraction map).

Proceeding from the 'Volume fraction' window the user is given another opportunity to inspect the volume fraction distribution. The 'Volume fraction and geometry preview' window (figure 15) again shows the volume fraction distribution in the lattice design space, but this time it is modified by the true lattice geometry, giving the user a clearer impression of their choices at this stage in the lattice design process. Note that figures 14 and 15 show the same volume fraction distribution, but the latter also reflects the geometry of the component, in this case a tensile test specimen with solid end pieces.

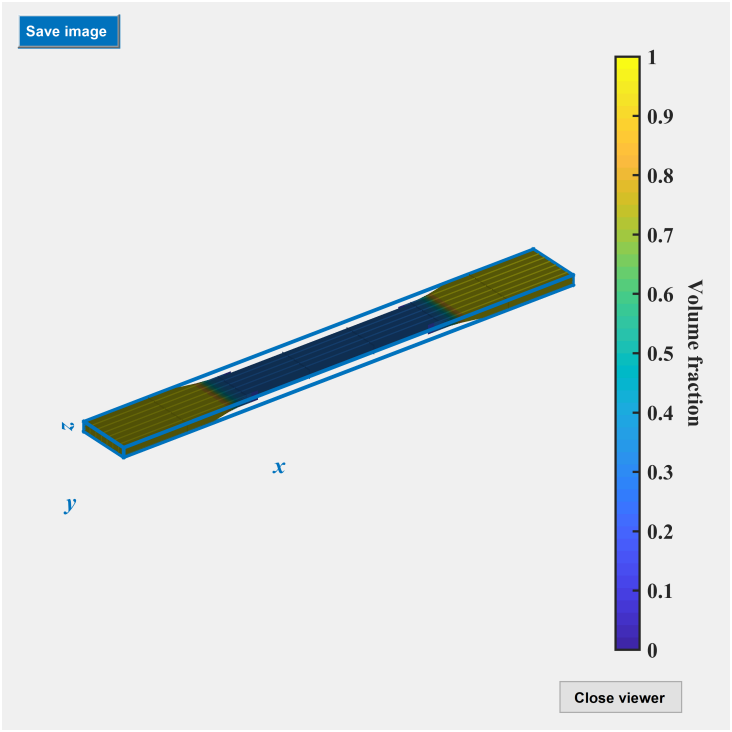

Figure 15: FLatt Pack GUI - Volume fraction and geometry preview.

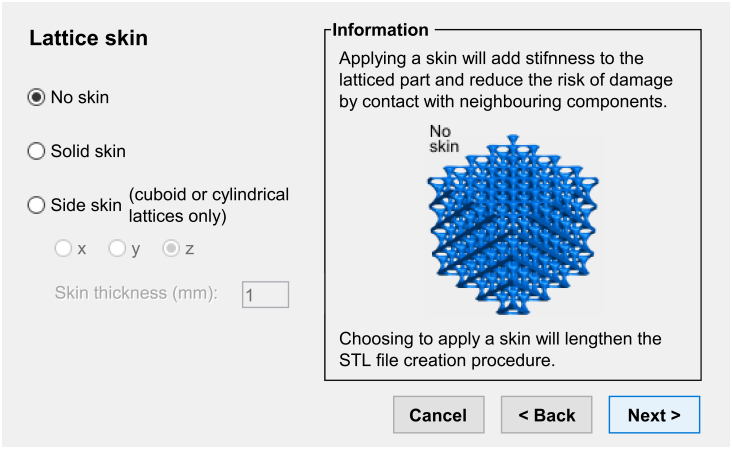

Figure 16: FLatt Pack GUI - Lattice skin.

1.7 Solid skin inclusion

The 'Lattice skin' window (figure 16) provides three skin options. Selecting between them changes the figure on the right of the window, which illustrates their effect on the lattice design. The 'No skin' option will leave the lattice structure unmodified. Depending on the chosen geometry and cell type, this may lead to exposed cell walls or struts which could be vulnerable to damage. If the user's objective is to determine some bulk property of a lattice (e.g., mechanical, thermal or acoustic performance) through manufacture and testing or simulation, it is recommended not to include an exterior skin, because the skin tends to alter the performance through additional mass, material connectivity and stiffness. However, it must be acknowledged in such studies that a lattice without a solid exterior skin will also exhibit edge effects where the periodicity and connectivity of the structure is terminated. For this reason, it is important to use appropriate boundary conditions in simulations and ensure sufficient numbers of repeating cells to approximate a homogeneous material distribution (as we briefly covered previously [1]).

The 'Solid skin' option is used to completely enclose the lattice with a conformal skin, which provides protection for the cell walls or struts as well as additional structural stiffness. The skin thickness is designated in an editable text box on the left of the window. The user must note that the inclusion of a solid skin will lead to completely enclosed voids. Such structures are not recommended for some AM processes, such as powder bed fusion and vat polymerisation, because they do not permit the removal of feedstock (powder, photopolymer resin, etc.) after manufacture. AM processes such as material extrusion and jetting do not have the same manufacturing restriction.

Lastly, the 'Side skin' option allows the user to add a solid skin to one opposing pair of faces of a cuboid lattice (e.g., the top and bottom faces), or to the ends or circumference of a cylindrical lattice. This option exists principally for the design of compressive test specimens, which are commonly used for the determination of structure-property relationships for AM lattices. For these specimens, it is sometimes preferable to apply loading to solid, flat surfaces rather than directly to the cell walls or struts.

1.8 Filename and output file selection

Following the addition or omission of a lattice skin, the user proceeds to the 'Output files' window (figure 17). Using the 'Save as . . . ' button in the lower left of the window, the user can specify the filename and folder for their files. The default filename consists of the user's lattice design choices; the class, cell type, dimensions, volume fraction, etc., for convenient referencing.

On the right of the 'Output files' window are checkboxes corresponding to a range of output file types. These are:

- 'STereoLithography (.stl)' - The format used extensively throughout AM and 3D printing. The binary version of the STL file, rather than the ASCII, is generated.

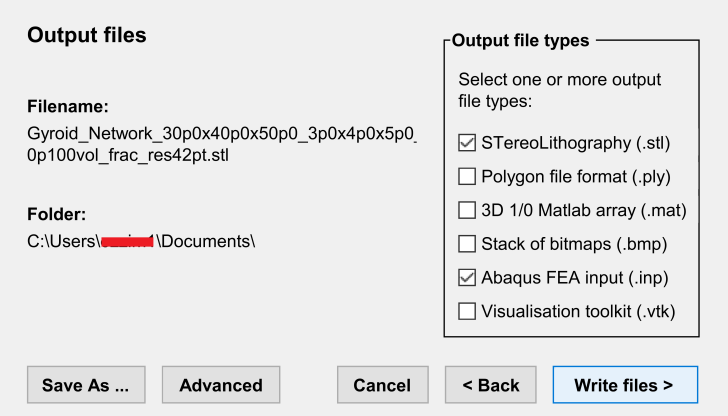

Figure 17: FLatt Pack GUI - Output files.

- 'Polygon file format (.ply)' - Like the STL, this provides a boundary representation of the 3D geometry, but with smaller file sizes. The binary version is generated.
- '3D 1/0 Matlab array (.mat)' - A Matlab logical array. The solid and void domains of the lattice are represented by '1' and '0' elements, respectively.
- 'Stack of bitmaps (.bmp)' - A folder containing a set of bitmap images. These are 'slices' of the lattice structure in the xy plane, with each image filename appended with a number according to its position in the z direction. In each bitmap image, the solid and void domains of the lattice are designated with black and white pixels, respectively.
- 'Abaqus FEA input (.inp)' - A hexahedral finite element mesh composed of hexahedral elements written in the Abaqus input file format. More information on this output option is given in section 1.10.
- 'Visualistion toolkit (.vtk)' - A structured dataset representing the lattice with VTK's XML syntax. The solid and void domains of the lattice are designated with '1' and '0', respectively.

Both the STL and PLY output formats are boundary representations. These are meshes of connected triangles marking the boundary between the solid lattice domain and the surrounding void space. The other output formats (MAT, BMP, INP and VTK) are based on a binarised version of the 3D surface array; elements or pixels in these files directly represent solid or void regions of the discretised lattice desin space. The 'Advanced' button in the lower left of the 'Output files' window allows the user to specify the element or pixel size in each of the MAT, BMP, INP and VTK output formats.

1.9 Triangle mesh reduction

If the user chooses to export an STL or PLY file, they will next be presented with the 'Triangle mesh reduction' window (figure 18). Using the slider or editable text box, the user specifies the degree of triangle reduction applied to the boundary mesh, which can significantly reduce the size of the output file. A value of 100% indicates that no reduction should be applied; the output mesh will comprise a relatively large number of small triangles. Values lower than 100% apply triangle reduction by replacing groups of small triangles with fewer larger ones. An 'Information' box on the right of the window provides notes to help the user make their choice.

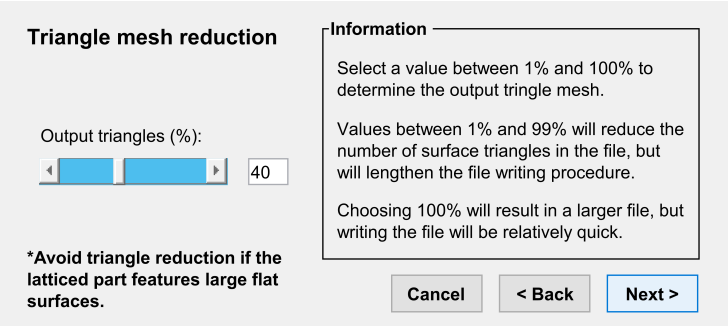

Figure 18: FLatt Pack GUI - Triangle mesh reduction.

1.10 FEA input file selection

If the user chooses to export an Abaqus FEA input (.inp) file, they will next encounter the 'Abaqus FEA job creation' window (figure 19). Radio buttons on the left allow the user to select between exporting the just the hexahedral mesh or an analysis input file, which additionally contains material properties, boundary conditions and a history output request.

The first option exports the lattice geometry as a 'Hexahedral mesh only', where the output file contains just the node coordinates and element information. The second option, 'Displacement load', is used to define a simulation with displacement loading. The elements are assigned a linear elastic material model, with the elastic modulus specified by the user on the right of the window. The elements are of type *C3D8R* (3D eight-node linear isoparametric element with reduced integration). The boundary conditions include a tensile or compressive displacement applied at the nodes of one side of the lattice structure, chosen using the 'Load direction' x, y or z radio buttons, while the nodes of the other side are fixed with either the *XSYMM*, *YSYMM* or *ZSYMM* condition. The magni-

tude of the applied displacement is specified as a percentage of the lattice structure size along the chosen loading direction. For example, if the user specifies a 2% magnitude for the displacement and the lattice has dimensions $40 \times 40 \times 40$ mm, the applied displacement will be 0.8 mm. The 'Displacement load' Abaqus FEA job is suited to simulating a compression or tension test, which can be used to determine the stiffness of the lattice structure and examine the stress distribution in the lattice walls or struts.

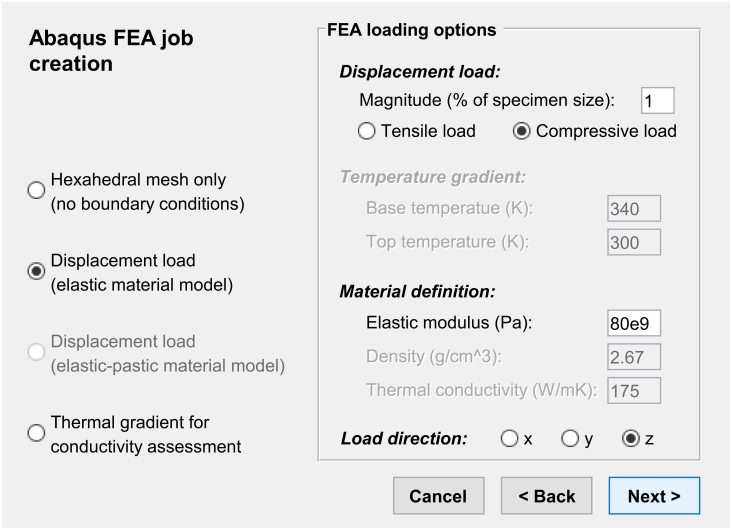

Figure 19: FLatt Pack GUI - Abaqus FEA job creation.

The third option, 'Thermal gradient for conductivity assessment', is used to define a steady-state heat transfer analysis. The temperatures of two opposing sides of the lattice (again, in x, y or z) are set by the user, as are the relevant material properties; mass density (in g/cm^3) and thermal conductivity (in W/mK). The elements are of type *DC3D8* (similar to *C3D8R*, but used for heat transfer and without reduced integration). This Abaqus FEA job allows the user to examine the local heat flux and temperature distributions resulting from the thermal gradient across the lattice.

1.11 Lattice preview and finish screen

If the user chooses to export an STL or PLY file, they are given the option to inspect a 3D preview of the entire lattice structure before the file is exported. This is presented in the 'Lattice preview' subwindow (figure 20), along with some information about the structure, including a note of the lattice design choices (cell type, dimensions, etc.) as well as the volume and surface area. This preview and structural information are intended to give the user a final opportunity to revise their design choices if the lattice does not correspond to their initial concept. This is done by closing the 'Lattice preview' subwindow and returning to the relevant section of the FLatt Pack GUI with the '< Back' buttons in each window.

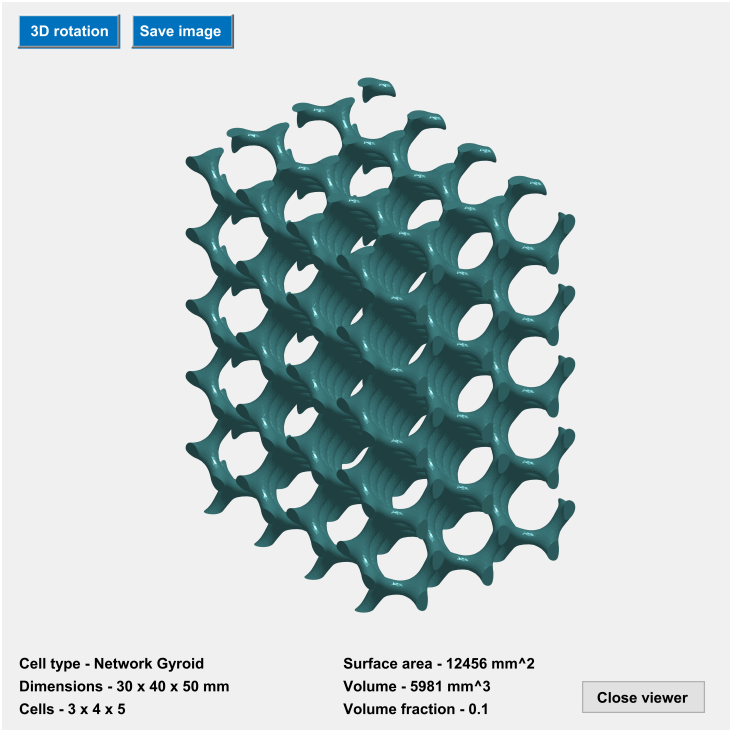

Figure 20: FLatt Pack GUI - Lattice preview.

Once the user's specified files are saved, they are shown the 'File creation complete' window (figure 21), which simply allows them to close FLatt Pack with the 'Finish' button or restart the lattice design process to create another structure.

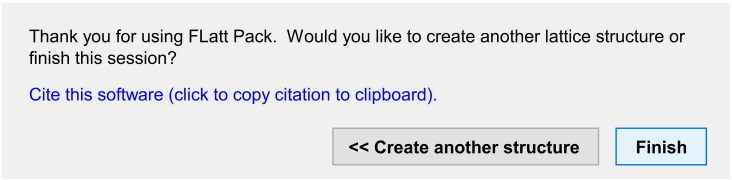

Figure 21: FLatt Pack GUI - File creation complete.

2 Examples

2.1 Lattice specimen for mechanical testing

Here we look at the use of FLatt Pack to investigate the response of a lattice structure under quasi-static compressive loading. For this example, a network phase gyroid

lattice is examined, comprising $5 \times 5 \times 5$ cells with dimensions of $50 \times 50 \times 50$ mm and volume fraction of 0.15. (Arrangements of $5 \times 5 \times 5$ cells have previously been shown to provide stiffness results reflective of homogeneous lattices, unlike structures with fewer cells [1].)

The investigation comprises two parts; (i) AM fabrication and mechanical testing, and (ii) numerical modelling with FE analysis, as illustrated in figure 22. Both routes begin with the creation of appropriate files in FLatt Pack. For manufacture an STL file is required (figure 22(a)), while for FE analysis an Abaqus INP file is required (figure 22(d)).

The user proceeds through the 'Geometry' and 'Dimensions and cell repetitions' windows of FLatt Pack, then arrives at the 'Cell type' window. The 'Modulus estimate' subwindow is consulted (see figure 9), which indicates that aligning the cubic [111] direction of the lattice with the applied load will provide the stiffest solution. This is achieved with rotations of 35.26° and 45° around the x and y axes, respectively. A uniform volume fraction of 0.15 is selected in the 'Volume fraction' window, and the user opts not to include an exterior skin, so that the observed deformation, stiffness, etc., may be ascribed to the lattice type *in general*, and may therefore be useful in making future design choices.

At the 'Output files' window, the user chooses to export STL and Abaqus INP files. A triangle mesh reduction setting of 50% is chosen, leading to an output STL file of 28 MB containing $\sim 600,000$ surface triangles. Following its generation in FLatt Pack, the surface mesh can be reduced further using dedicated AM build preparation software (e.g., Materialise Magics [2]), which is generally required anyway, to perform minor STL repairs, as well as the slicing and hatching operations to create an AM machine input file.

The specimen is fabricated by polymer laser sintering of PA2200 (figure 22(b)) before being subject to quasi-static compression using a universal testing machine with a load transducer. The resulting stress-strain curve (figure 22(c)) exhibits the typical features of low volume fraction cellular structures under compression; an initial elastic region at low strain, a long platic plateau, and a densification region at high strain, in which the stress increases significantly due to the collapse and subsequent contact of the cell struts. The elastic modulus can be extracted from the low strain linear region; in this case it is $E = 38.8 \pm 0.1$ MPa.

Pursuing the numerical modelling route, at the 'Abaqus FEA job creation' window, the user selects the 'Displacement load' option and provides the elastic modulus of the material used for manufacturing. In this example, the modulus was given as 1.695 GPa, based on our previous study of laser sintered PA2200 [3]. The displacement load is set to 1% of the height of the specimen (i.e., 0.5 mm) and the load direction is set as z, mirroring the load applied in physical testing. The output INP file is 97 MB in size, containing ~ 1.2 million nodes and $\sim 900,000$ hexahedral elements. This is run using the Abaqus/Standard solver (Dassault Systemes - Vélizy-Villacoublay, France), providing an output database (ODB) file with node and element information for the FE solution. As an example of such information, the von Mises stress distribution for the network gyroid lattice is shown in figure 22(e), in which it can be seen that the greatest stress is found in those struts which are near-vertically aligned or, more precisely, those which are closely aligned with the direction of the applied load.

2.2 Lattice infill for ease of manufacture

In this example, FLatt Pack is used to generate lattice infill for a solid part, providing a design that can be manufactured quicker and with less material than the original. This is the ubiquitous application of lattice generation in AM, since it enables materially-efficient manufacture and light weight products. The demonstration case here is a piston rod from the popular digital design sharing website Thingiverse (www.thingiverse.com), as shown in figure 23(a).

The original model STL was loaded into FLatt Pack using the 'Custom' option in the 'Geometry' window. The 'Dimensions and cell repetitions' window then provides the dimensions of the part, which in this case are approximately $33 \times 11 \times 68$ mm. The user specifies cell repetitions of $9.3 \times 3 \times 18.9$, maintaining the aspect ratios of the part dimensions, and therefore ensuring cubic lattice cells (see figure 6). The user selects the 'Very fine' input STL resolution option, to preserve as much as possible the quality of the part's outer form. In the following windows the user selects the simple cubic cell type, a uniform volume fraction of 0.2, and a solid skin of thickness 1 mm. At the 'Triangle mesh reduction' window the user opts to export 100% of the surface triangles, providing an output STL file 184 MB in size, containing ~ 3.8 million triangles. The file is then post-processed with AM build preparation software to make minor STL repairs, and for slicing and hatching. The AM build preparation software may also be used to introduce a small hole in the part skin for the removal of excess feedstock if necessitated by the AM process. The final design, with the interior lattice exposed

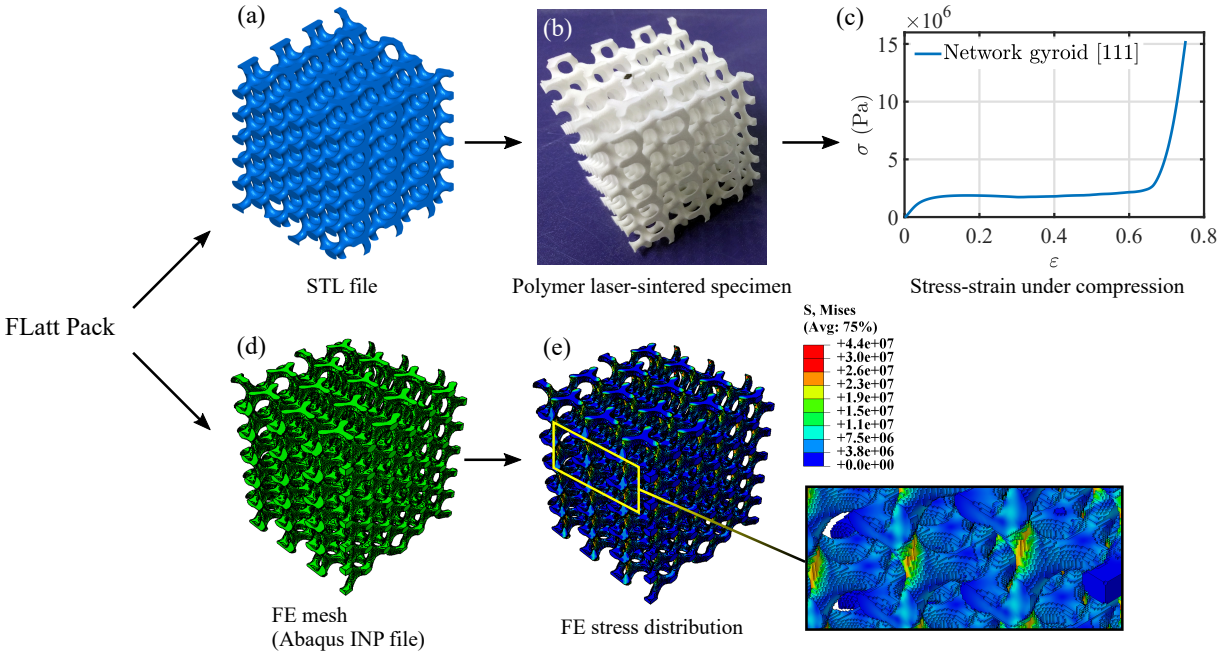

Figure 22: The use of FLatt Pack to investigate the response of a lattice structure under quasi-static loading.

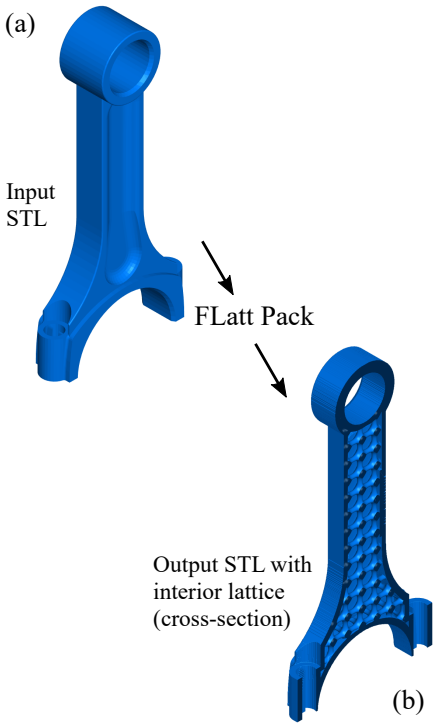

Figure 23: The use of FLatt Pack to provide a supporting lattice infill.

in cross-section, is shown in figure 23(b).

2.3 Lattice generation based on an input volume fraction distribution

This last example demonstrates the use of FLatt Pack to generate a lattice based on an arbitrary volume fraction distribution in an input file. The part in question is a tensile test specimen (or 'dog bone'), which the user wishes to create with solid ends to facilitate better gripping during the tensile test.

The user begins by selecting the 'Custom' option in the 'Geometry' window, then importing the file 'ASTM_type_1_tensile_bar.stl', which is included with the FLatt Pack installation (figure 24(a)). The user selects the 'Fine' input STL resolution option and specifies cell repetitions of 25.8 × 2.95 × 0.5, maintaining the aspect ratio of the part dimensions, as in the previous example.

In the following windows the user selects the honeycomb primitive cell type and uses the 'Import vol. frac. map' button to import the file 'Density_map_for_ASTM_tensile_bars.bmp' (figure 24(b)). As described in section 1.6, this bitmap image contains volume fraction values represented as colours, where black (r g b = 0 0 0) corresponds to void in the lattice design space, and white (r g b = 255 255 255) corresponds to solid. The 'Density_map_for_ASTM_tensile_bars.bmp' image has a central grey region (r g b = 77 77 77) corresponding to $\rho^* = 77/255 = 0.3$, which transitions to the white, or solid, end pieces. The bitmap was created using the free and open-source vector graphics editor, Inkscape

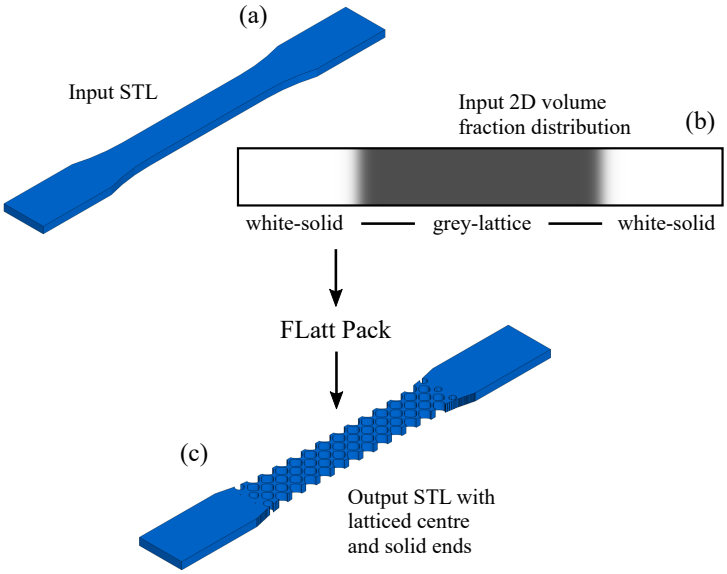

Figure 24: Lattice generation based on an input volume fraction distribution.

(www.inkscape.org), though other image editors could just as easily be used. At the 'Triangle mesh reduction' window the user opts to export 100% of the surface triangles, providing an output STL file 54 MB in size, containing $\sim$ 1.1 million triangles. The final design is shown in figure 24(c).

References

[1] I. Maskery, A. O. Aremu, L. Parry, R. D. Wildman, C. J. Tuck, and I. A. Ashcroft, "Effective design and simulation of surface-based lattice structures featuring volume fraction and cell type grading," *Materials and Design*, vol. 155, pp. 220–232, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S026412751830443X

[2] (2020) Materialise 3-matic. [Online]. Available: https://www.materialise.com/en/software/3-matic/modules/lattice-module

[3] I. Maskery, L. Sturm, A. O. Aremu, A. Panesar, C. B. Williams, C. J. Tuck, R. D. Wildman, I. A. Ashcroft, and R. J. M. Hague, "Insights into the mechanical properties of several triply periodic minimal surface lattice structures made by polymer additive manufacturing," *Polymer*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0032386117311175