

Git Project: Image Analysis and FEM

G. Bowker, S. Lister, J. Collins, S. Rehman, F. Livera

1 Introduction

Finite element modelling (FEM) is the process of segmenting a defined geometry by meshing, and solving coupled partial differential physics equations for individual elements within the built mesh, given inputs of boundary conditions surrounding the geometry, a series of timesteps over which the simulation runs, and materials parameters that describe how the geometry will behave under the specified conditions. The defined geometry may be a two dimensional image or a three dimensional object. For the purposes of this project, microstructural images have been used to estimate the physical behavior of a material on the microscopic scale. Image processing software based in Python is used to prepare such images for conversion to a finite element modelling geometry input. There are a variety of different open-source finite element solvers and image processing software available for use in this task, including: sfepy, OOF2, DAMASK, gmsh, sci-kit image and scipy.

2 Aims and Objectives

2.1 Aims

- Pre-process images.
- Perform finite element analysis on pre-processed images.
- Discuss results.

2.2 Objectives

- To pre-process several microstructural images through the use of sci-kit image and scipy in jupyter notebook by binarising the image into two distinct phases.
- To create a finite element mesh for the processed image using an adaptive algorithm.
- To research materials parameters for each of the respective phases present in the geometry.
- To define boundary conditions, field equations, material parameters and outputs of the simulation.
- To post-process outputs of the simulation in order to extract useful results, plot, and produce conclusions based on a comparison of simulated results to that of experimental results found in literature, with a consideration of the assumptions used to create the model.
- To describe possible future work to build upon the developed workflow.

3 Image Analysis

A key requirement for this project was to build a python script allowing JPG images to be read as an array, manipulated and then processed with Python packages such as Sci-Kit Image (skimage) [1]. Several microstructure images have been used in this project, from a variety of sources including previous research data, scientific literature and the provided database, all with full and sufficient permissions. Images were then pre-processed accordingly through various techniques such as segmentation, removal of noise and measurement testing, before being imported into FE simulations to create a mesh. This section will further detail how this process was undertaken and broken down through the use of test cases to ensure that the python script and skimage were performing accurately and as desired.

The first test case that was carried out was processing an image of a ferritic microstructure taken from the Materials Science and Engineering: An Introduction by W. D. Callister, as seen in Figure 1 [2]. Firstly, an understanding of how to navigate and operate the skimage package was developed, as the software was new to the authors, through testing different algorithms and operations independently on the test image. This included manipulating image colour and contrast, converting the image to grayscale and cropping the image to remove the scale bar. Other features that were tested at this stage was the use of filters such as the Sobel filter, which is commonly used in image processing to emphasise the edges in an image, aiding edge detection algorithms. A watershed transform was another image processing technique that was trialled at this stage, where the image is treated as a topographical map, with brightness determining elevation of each point and then identifying the lines which run across the peaks to segment the map. Finally, these operations were combined to process the image shown in Figure 2 as effectively as possible, before the measuring tools were used to estimate average grain size.

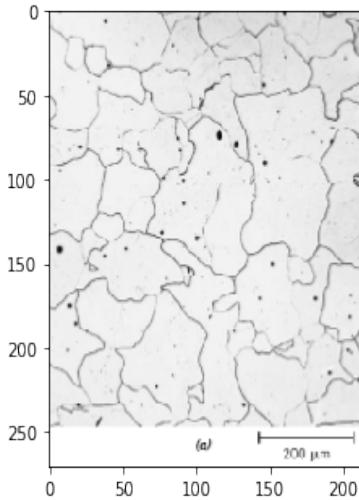


Figure 1: A figure showing the initial microstructure image taken from [2].

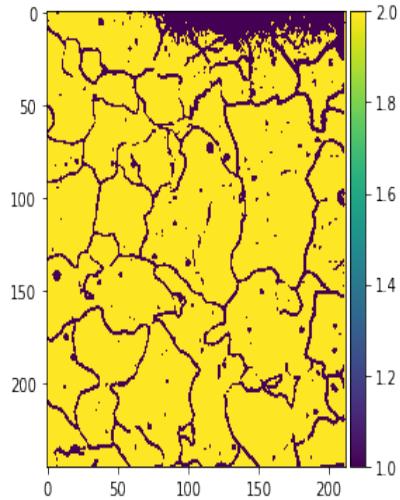


Figure 2: A figure showing the final image after segmentation and processing.

Secondly, in test case 2, the effectiveness of the code developed in the first test case was applied to an image of a thermally etched steel microstructure in Figure 3 showing prior austenite grains. This image was taken from previous research by Joshua Collins and authorised by their academic supervisor, Dr Ed Pickering. This microstructure image, as seen below, had previously been the subject of average grain size analysis via the linear intercept method. The average grain size was calculated using the developed image processing Python script and compared with experimental results to assess the accuracy of the approach. The image was processed using many of the techniques tested out previously, with the image first being read before being cropped to remove unfocused areas of the micrograph. The image was then segmented via the application of the Sobel filter, application of markers and then a watershed transform to fill in regions of the elevation map before the grains could finally be segmented and labelled individually. Finally, the image was converted back to gray scale, to allow each grain to be assigned a phase number associated with material properties. This allowed the "image" data to be exported as a numpy array and saved as a Comma Separated Value (CSV) file for easy importation into the Gmsh meshing software later.

As mentioned previously the successfulness of the second test case was validated by comparing the calculated average grain size from the python algorithm to the previously measured value using the linear intercept method. To do this, the area of each segmented grain in Figure 4 was measured and then the average diameter of each grain was estimated. This was done by assuming that the grains were perfect circles. Although this is a large assumption which is clearly not true, it yielded excellent results as an average grain size of 23.5 microns was estimated, with the previously measured average of 24.9 microns. Such a small difference in average grain size clearly demonstrates the viability of this approach and provided reassurance that the algorithm could accurately be applied to other images.

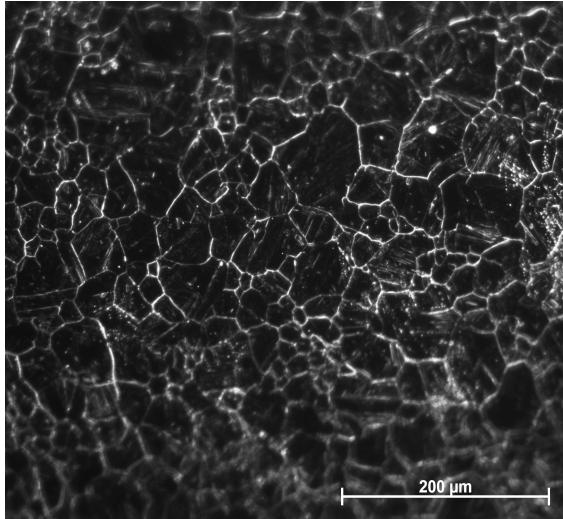


Figure 3: A figure showing the initial SA508 thermally etched steel microstructure image.

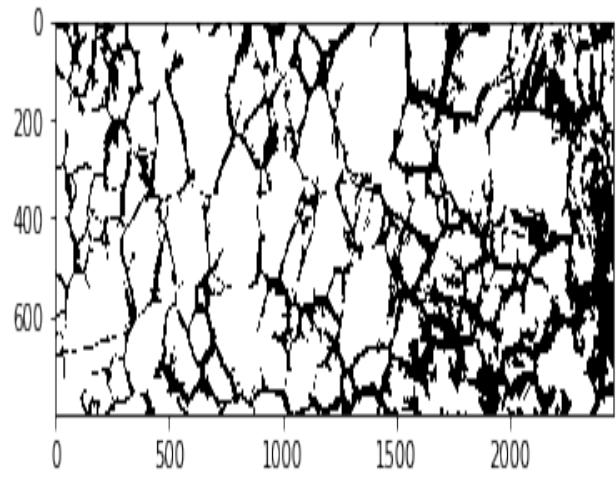


Figure 4: A figure showing the final steel microstructure image after segmentation and processing.

Test case 3 was then commenced, seeking to apply the algorithm to Figure 5, this time a brightfield image of an $\alpha + \beta$ microstructure of the Ti-6Al-4V alloy, provided by Dr Pratheeck Shantraj and shown below. The same procedure as in the previous test case was applied to the image and the image was successfully converted into a segmented, binary image shown in Figure 6 and then exported into the Gmsh software in order to create a square based mesh for subsequent FEM analysis.

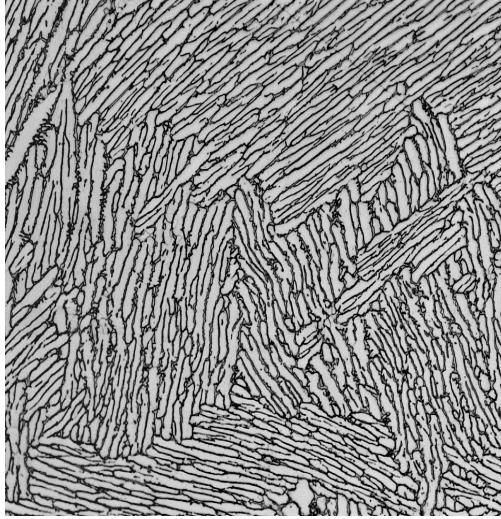


Figure 5: A figure showing the initial Ti-64 microstructure image.

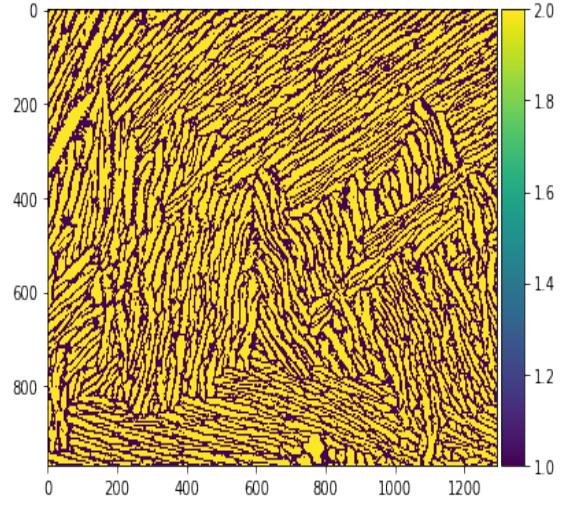


Figure 6: A figure showing the final Ti-64 microstructure image after processing.

Test case 4 then took the same image from the previous test case and processed and segmented a single grain, shown in Figure 7 for meshing and FEM analysis. This analysis was done with the use of the additional skimage sub-packages; threshold filters and clean border segmentation to remove the surrounding grains from the image. The binary, single grain image seen below in Figure 8 demonstrates that this approach is valid for application to, and isolation of, individual grains, as well as a multitude of grains – as demonstrated in test case 3. The successful application of the 4 test cases allowed the authors to gain a high level of confidence in the algorithm and therefore move on to the next stage of the project, meshing and FE.

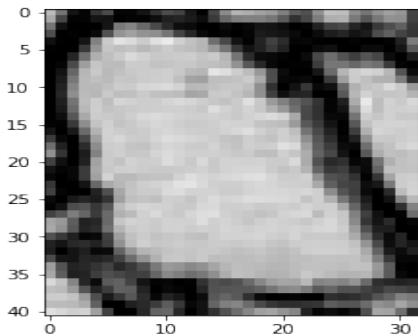


Figure 7: A figure showing the initial image of a single grain of the Ti-64 microstructure.

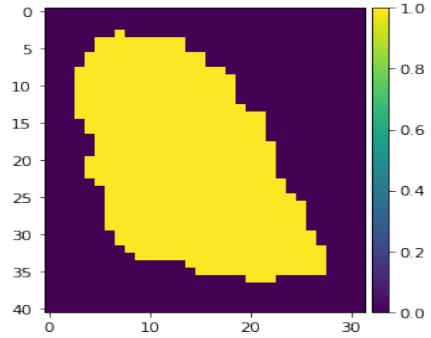


Figure 8: A figure showing the final image of a single grain of Ti-64 after segmentation and processing.

Finally, the rigorously tested image processing code was compiled into a single function, to allow it to be easily applied to any image. The function termed 'process_image' takes 5 input arguments and returns a binary image that can be inputted straight into the OOF2 program for meshing and simulations. OOF2 was used for the final function as opposed to Gmsh as during testing it became clear that it provided a more robust method for this particular application, which will be discussed further in the next section. The input arguments are as follows; file = the input image, e = exposure value, u/v are maximum and minimum threshold parameters respectively, and output = file name for the output file. The function therefore reads the input image, converts it to grayscale (if it is currently a rgb file) and then adjusts the exposure as per the input value of e. The Sobel filter is then applied using the values of u and v as the max/min markers to isolate grains and separate them from grain boundaries. Following this the watershed transform is applied to create a binary image which is then saved in the specified output file. The function permits the user to quickly change the input arguments and see the results easily, allowing comparisons to be made and the best possible output to be produced for the next step, meshing. Figure 9 shows an example of a Ti-64 microstructure image that was processed using the final function. This image would then go on to form the basis of the final simulations in the following section.



Figure 9: A figure showing a processed Ti-64 microstructure image that was used for the final simulation.

4 Finite Element Modelling

The FEM simulations were run using the Object Oriented Finite (OOF) element analysis program called OOF2, created by the National Institute of Standards and Technology (available here: <https://www.ctcms.nist.gov/oof/oof2/>) [3]. Minor changes to the simulation were altered in the Python code, rather than using the OOF2 graphical user interface (GUI) as this saved time. Four test cases were trialed to ensure the simulation was generating results as expected, the simplest case is heat diffusion through a simple shape and the most complex incorporating heat diffusion and displacement in a single grain of Ti-6Al-4V. Once the test cases proved successful, the FEM simulation was run for a larger section of Ti-6Al-4V and the heat diffusion and displacement were determined.

4.1 Model Validation

The first test case is a simple star shape, test cases 2-4 are all Ti-6Al-4V (Figure 10).

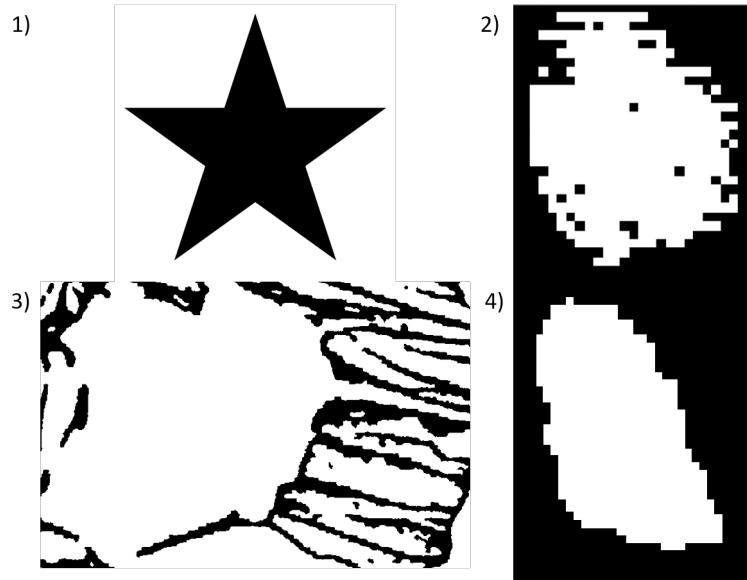


Figure 10: OOF2 input files for the test cases

Each test case input was a binary file, where the black and white regions were specified as separate materials. The thermo-physical properties of the black and white regions were assigned as specified in Table 1: Young's Modulus, E , Poisson's Ratio, ν , thermal conductivity, κ , and thermal expansion coefficient, α . The microstructure skeleton was created using a 40x40 QuadSkeleton grid, which was annealed, the edges swapped and the skeleton smoothed to improve the homogeneity index (shown Table 1). The mesh was then generated from the skeleton, with the mapping and interpolation orders remaining at 1 to save on computational time. The fields were then defined in accordance to Table 1 on the mesh, constrained in-plane as active (such that the FE output contains the specified parameters). The boundary conditions are set as 0 °C on the left edge and 1000 °C on the right edge. When the force field was applied, the left edge displacement in (x,y) was fixed at (0,0). The solver tolerance was set to 1×10^{-13} with 1000 iterations and the results are visualised. For test case 4, the stress contour map was created in post processing within the OOF2 program where the flux was set to stress and the invariant was set to trace.

Table 1: Physical Properties, Homogeneity Index, Model Fields and Boundary Conditions of Test Cases 1-4

Test Case	Black Region Properties	White Region Properties	Homogeneity Index	Fields	Boundary Conditions
1	$E = 300 \text{ GPa}$ $\nu = 0.33$ $\kappa = 1000 \text{ W/mK}$ $\alpha = 5 \times 10^{-6} \text{ K}^{-1}$	$E = 10 \text{ GPa}$ $\nu = 0.27$ $\kappa = 1 \text{ W/mK}$ $\alpha = 1 \text{ K}^{-1}$	0.992	Temperature	0 - 1000 °C
2	$E = 117 \text{ GPa}$ $\nu = 0.33$ $\kappa = 390 \text{ W/mK}$ $\alpha = 51 \times 10^{-6} \text{ K}^{-1}$	$E = 200 \text{ GPa}$ $\nu = 0.27$ $\kappa = 41 \text{ W/mK}$ $\alpha = 36 \text{ K}^{-1}$	0.996	Temperature Force	0 - 1000 °C Left side (x,y) fixed at (0,0)
3	$E = 150 \text{ GPa}$ $\nu = 0.33$ $\kappa = 8.5 \text{ W/mK}$ $\alpha = 10 \times 10^{-6} \text{ K}^{-1}$	$E = 117 \text{ GPa}$ $\nu = 0.34$ $\kappa = 6.7 \text{ W/mK}$ $\alpha = 8.6 \times 10^{-6} \text{ K}^{-1}$	0.958	Temperature Force	0 - 1000 °C Left side (x,y) fixed at (0,0)
4	$E = 40 \text{ GPa}$ $\nu = 0.33$ $\kappa = 15 \text{ W/mK}$ $\alpha = 1 \times 10^{-3} \text{ K}^{-1}$	$E = 117 \text{ GPa}$ $\nu = 0.34$ $\kappa = 6.7 \text{ W/mK}$ $\alpha = 8.6 \times 10^{-6} \text{ K}^{-1}$	0.995	Temperature Force	0 - 1000 °C Left side (x,y) fixed at (0,0)

The results from test case 1 are shown in Figure 11 and clearly show the heat diffusion through the star shape. The properties assigned to the image, where the black region conducts the heat more effectively than the white region, can clearly be seen in the heat profile. The thermal gradient is much more steady when the heat flows through a white region compared to the longest cross-section of the black star shape, which remains at approximately 400 - 500 °C from point to point. These results suggest the simulation is working correctly and so, the simulation can be made more complex with the inclusion of the force field.

The simulation run for test case 2 involved the temperature and force fields in the OOF2 GUI, and the results are shown in Figure 12. The material properties used for this simulation are similar to that of the alpha and beta regions of Ti-6Al-4V. The difference in the coefficient of thermal expansion between the regions has caused a displacement, which is clearly visible towards the right of the simulation result images, with the displacement range being higher in the y-direction by 5.03 μm. The difference between the thermal conductivity values assigned to each region is small, so the thermal gradient is relatively consistent across the grain. A more complex structure can be looked at for the next test case.

Test case 3 has a binary image of Ti-6Al-4V with multiple grains as the input file, to determine the impact of multiple grain boundaries on the simulation quality. The temperature and displacement contours are shown in Figure 13. The material properties are similar between the black and white regions to simplify the simulation and ensure an output is obtained within a reasonable computational time frame. The results are as expected for a simulation with minor material property

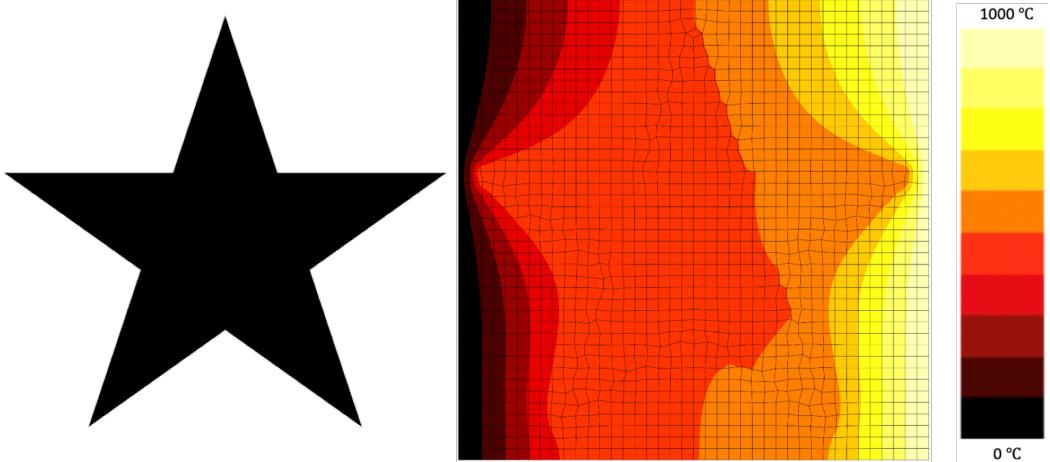


Figure 11: Test case 1 results, showing the original input file and the heat diffusion through the star shape

variations, with the temperature varying linearly in the x-direction and the displacement being consistent in the x- and y-directions. The inclusion of a stress contour can be examined for the next simulation, as it is clear that multiple grains can be processed in the simulation.

Test case 4 used a binary input of a Ti-6Al-4V grain which was image processed using the method described in the Image Analysis section. The material properties in this simulation were exaggerated to determine the impact on the results and ensure the model holds true for such cases. The large difference in coefficient of thermal expansion between the black and white regions is clear in the visualisation of the mesh over the contour plots shown in Figure 14. The y-displacement totals $40.6 \mu\text{m}$ and the x-displacement totals $20.4 \mu\text{m}$, which is significantly larger than the previous tests. The stress map was added as an output for this test case and the result show the most significant stresses were concentrated on the grain boundaries, particularly at higher temperatures, where stresses reached $2.61 \times 10^5 \text{ N/m}^2$.

The test results have shown that the simulation is able to process binary images with multiple grains to produce contour plots of temperature distribution, displacement along x and y, and stress distribution. This means that the FEM is capable of running the final simulation to gain information about the heat diffusion through a multi-grain Ti-6Al-4V image and the results can be trusted.

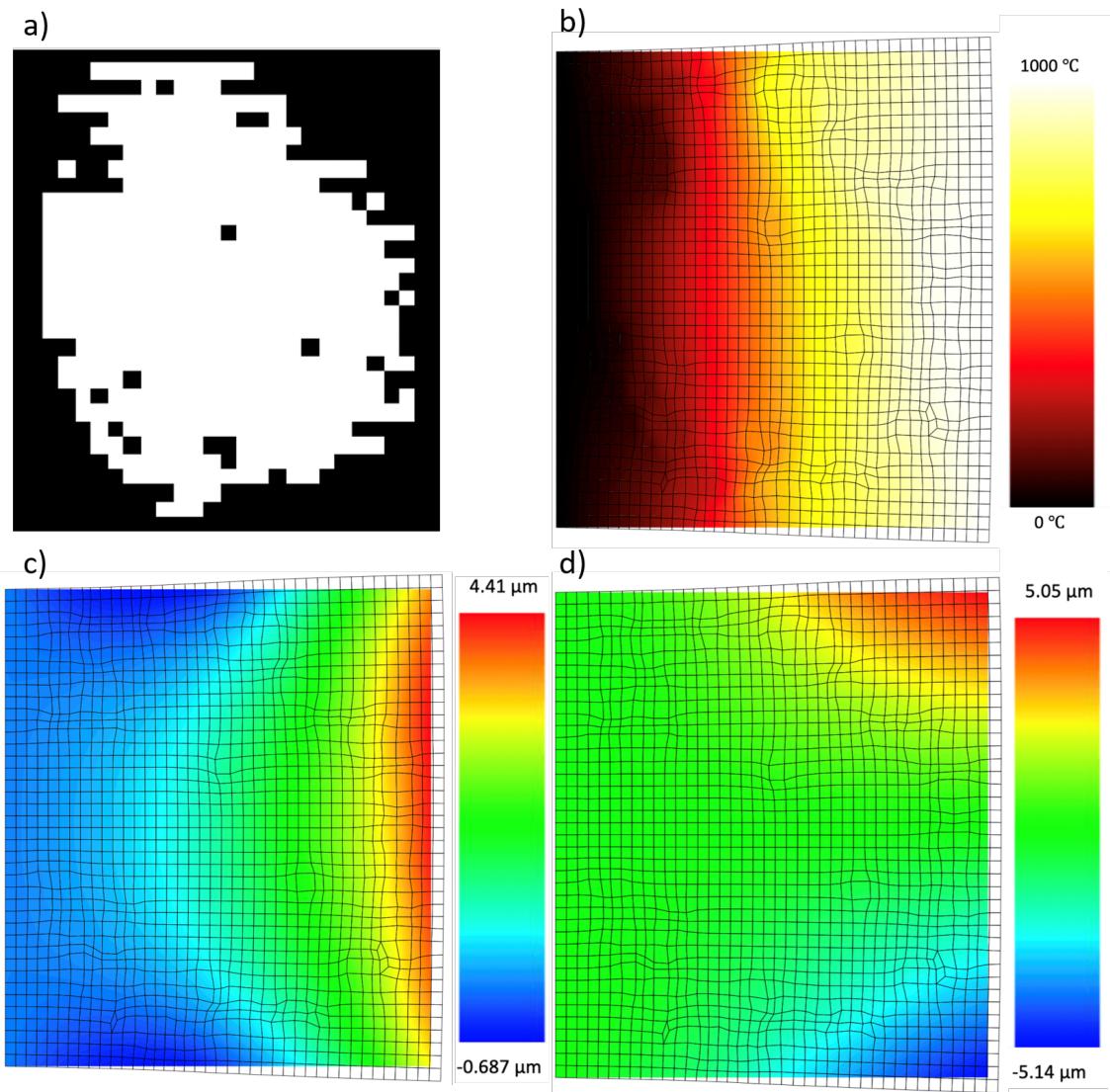


Figure 12: Test case 2 results, showing the a) original microstructure of Ti-6Al-5V b) heat diffusion through the grain c) displacement in the x-direction d) displacement in y-direction

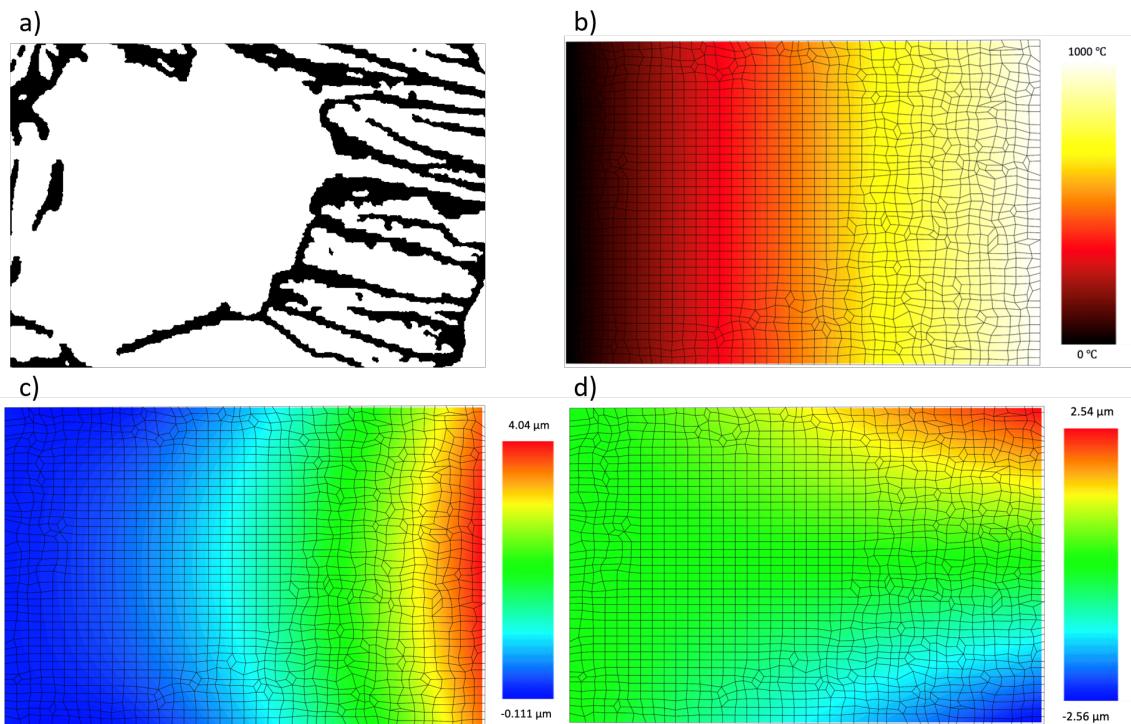


Figure 13: Test case 3 results, showing the a) original microstructure of Ti-6Al-5V b) heat diffusion through the grains c) displacement in the x-direction d) displacement in y-direction

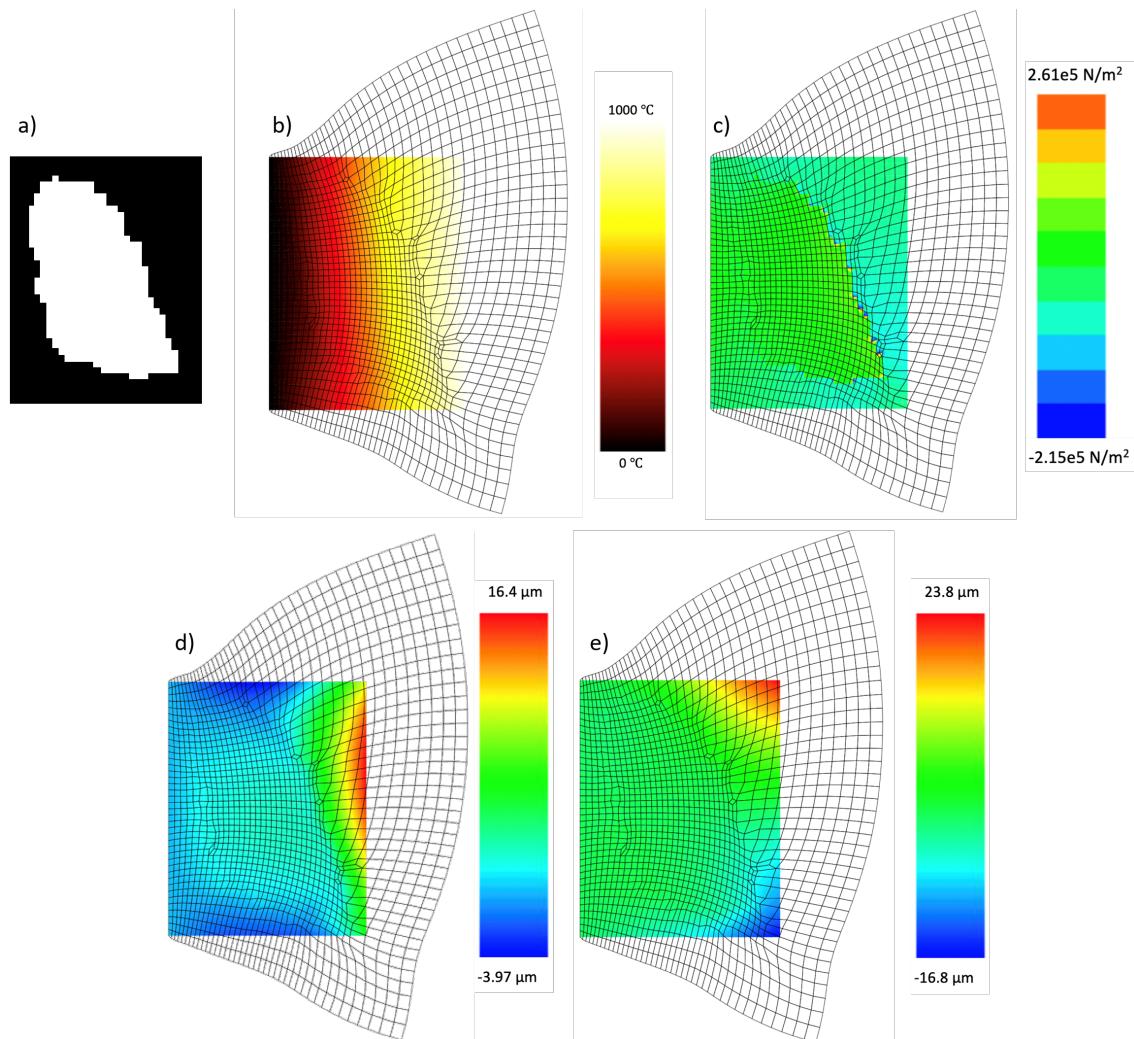


Figure 14: Test case 4 results, showing the a) original microstructure of Ti-6Al-5V b) heat diffusion through the grain c) stress within the grain d) displacement in the x-direction e) displacement in y-direction

4.2 FEM of a Globular Ti-6Al-4V Microstructure

The FEM was used to simulate heat diffusion, displacement and stress distribution in a binary image of a globular Ti-6Al-4V microstructure 15. Only a section of the original image was used to save on computational time.



Figure 15: Image of globular Ti-6Al-4V which has been previously processed, showing the section (highlighted in red) for the FEM simulation

The black (Ti-beta) and white (Ti-alpha) regions of the microstructure are considered to be separate materials within the simulation. The thermo-physical properties of the alpha and beta regions are assigned as specified in Table 2. The microstructure skeleton was created using a 40x40 QuadSkeleton grid, which may include tri elements to gain greater accuracy despite the increased computational times. The skeleton was annealed, the edges swapped and the skeleton smoothed to improve the homogeneity index to 0.940. The mesh was then generated from the skeleton, with the mapping and interpolation orders remaining at 1 to save on computational time. The temperature and force fields were then defined on the mesh, constrained in-plane as active (such that the FE output contains the specified parameters). The boundary conditions for the temperature were set as 0 °C on the left edge and 1000 °C on the right edge. The force field boundary condition was fixed on the left edge, such that there is no (x,y) displacement on the edge. The solver tolerance is set to 1×10^{-13} with 1000 iterations and the results are visualised. The stress contour map was created in post processing within the OOF2 program where the flux was set to stress and the invariant set to trace.

Table 2: Thermo-physical properties of the Ti-6Al-4V used in the heat diffusion simulation

Property	Alpha Phase	Beta Phase
Young's Modulus, E (GPa)	92.8	75.8
Poisson Ratio, ν	0.35	0.31
Thermal conductivity, κ (W/mK)	7.6	11.3
Thermal expansion, α ($\times 10^{-6}$ K $^{-1}$)	8.6	11.8

5 Results and Discussion

The FEM simulation was successful in outputting all parameters (figure 16), showing that the OOF2 program can handle multi-grain images with two sets of material parameters. The heat diffusion looks constant through the alpha and beta grains which is expected by their thermal conductivity values being similar. The x- and y- displacement contours are fairly consistent with previously generated results in the test cases, with the maximum displacement at 2.64 μm and 2.55 μm respectively. These displacement values are more similar than previous results, due to the image containing a more even quantity of alpha and beta and the coefficients of thermal expansion are more similar than the test cases.

The FEM stress distribution for the Ti-6Al-4V image shows similar results to test case 4, where stress increases with decreasing temperature. The alpha regions show higher stress values relative to the beta regions at equivalent temperatures. Alpha Ti-6Al-4V has a lower coefficient of thermal expansion and hence the shrinkage with decreasing temperature is less significant, causing the stress to be higher. For this reason, there are stress concentrations at grain boundaries due to the shrinkage mismatch. This shows that the simulation is working well to give information about the stress distribution in the globular microstructure of Ti-6Al-4V.

The simulation is limited in its usefulness as temperature dependent parameters have not been used. It also assumes that the Ti-6Al-4V microstructure is constant, when cooling from 1000 °C to 0 °C goes through the β -transus at 980 °C and the β to metastable α' at 800 °C [4]. The simulation was fixed on the left edge which may also affect the stress distribution in the microstructure.

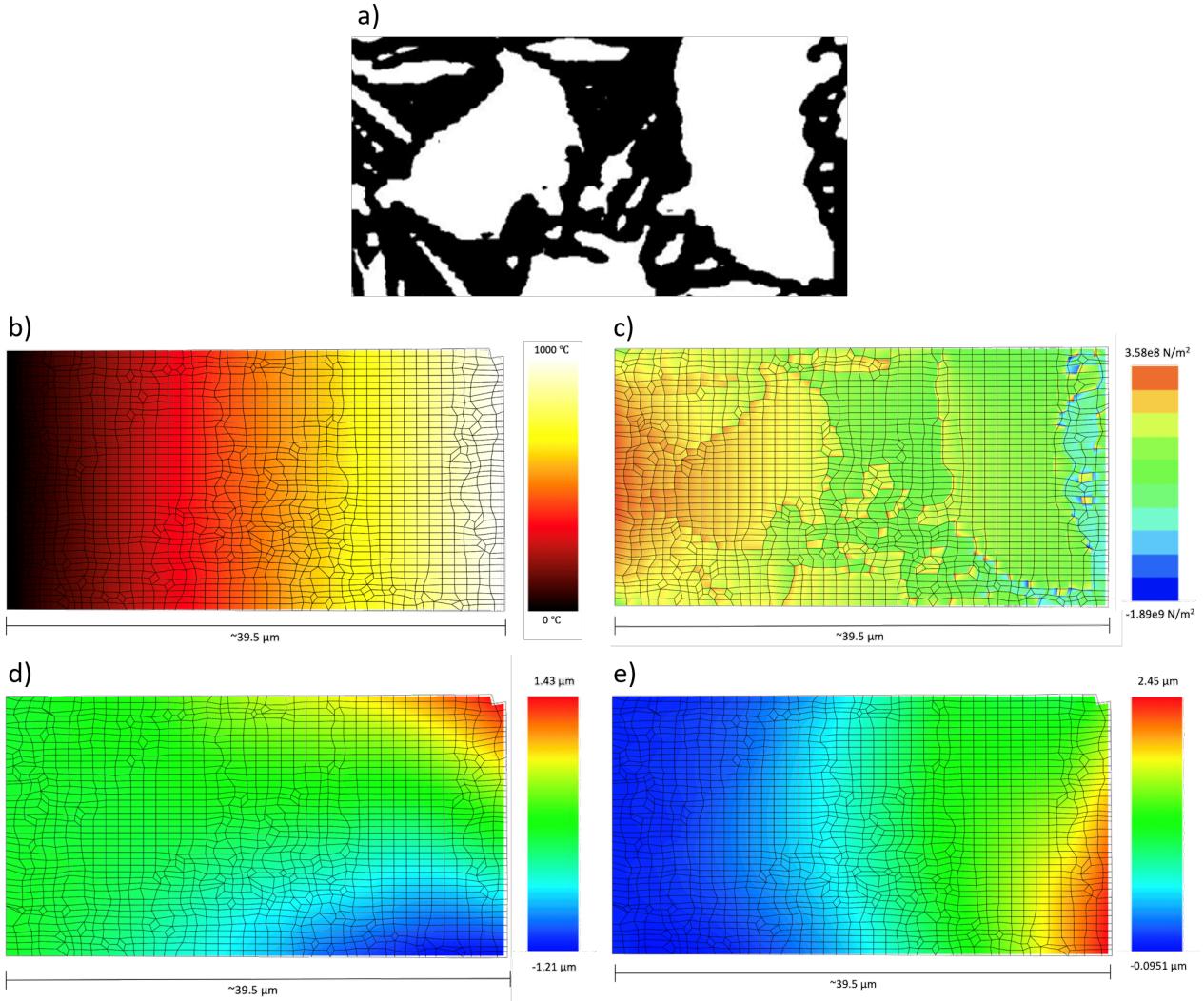


Figure 16: FEM simulation results, showing the a) original microstructure of Ti-6Al-4V b) heat diffusion through the grains c) stress within the grains d) displacement in the x-direction e) displacement in y-direction

6 Conclusion

This project has successfully obtained microstructural images that have been pre-processed and imported into a simulation using available open-source FE and image-analysis packages. In terms of the image preprocessing, multiple test cases were conducted until a function which is capable of segmenting and binarising a image is obtained. The function was subsequently used on sections of a globular Ti64 microstructure to be imported in the FE processing package: OOF2. OOF2 is based on a set of C++ classes for finite elements and material properties, tied together in a Python infrastructure; hence, any simulation was adjusted with a few lines of Python code. To ensure OOF2 was utilised correctly and to validate results, multiple tests cases were also conducted for the FE processing side of the project. Each subsequent test case was increased in model complexity, where the final simulation involved a combination of post-processing techniques from each test case, the correct material parameters for the α and β phases of Ti64, a refined FE mesh and defined boundary conditions and field equations. The results of the final simulation included contour maps

for the temperature distribution, the displacements in the x and y directions and the stress concentrated areas in the microstructure. Although the aim of this project was achieved, the results of the simulation have not been compared to any existing simulation results and the FE software package, OOF2, contains many bugs which make each simulation unreliable and unreplicable.

Further work for this project could be to compare the globular microstructure simulation results to the lamellar microstructure for Ti64 and to analyse any notable differences. The use of temperature dependent parameters and the impact of the microstructural change upon cooling could be added to improve the simulation. An additional test case for the FE processing could be introduced which includes a 3D model, which could be obtained from simply extruding 2D images. This can then be inputted into the FE package: OOF3D, which works in the same manner as OOF2 but in three dimensions. Although the homogeneity index was high, a mixture of element types (quad and tri) could also be used in the mesh to provide a more accurate representation of the microstructure. The image processing function can also be optimised through the use of loops and combining scripts. Finally, a software package with proven reliability and less bugs could be used to provide reliable results e.g. SfePy, FiPy.

7 Appendices

7.1 Appendix 1.0 - SfePy

Simple finite elements in Python (SfePy) uses finite element methods to solve coupled partial differential equation (PDE) in systems up to three dimensions. SfePy is a powerful software that allows complex physical problems to be coded quickly and easily. It has been used successfully in a variety of disciplines, ranging from biomechanical modelling [5] to the computational analysis of acoustic transmission coefficients [6].

In this report, the input file to the SfePy software is a microstructural image which must first be 'cleaned' through segmentation, mesh generation and noise reduction. It can then be imported into the software as a mesh file, where boundary and initial conditions are applied. Fields are then created which can be used to define variables which may be 'unknown field', 'test field' or 'parameter field' [7] and the material properties are defined. Complications arose in attempting to use sfepy for the processed images, therefore the group opted to use the OOF2 FEM software.

7.2 Appendix 2.0 - Gmsh

Gmsh is a tool for creating a finite element mesh from an image or a geometry built from scratch. The software allows for editing of algorithm scripts based in ASCII to finely tune node and element positions or even adaptively mesh. This was originally used to create input files for sfepy, however OOF2 was chosen instead. Shown is an example of a adaptive generated mesh for one of the images.

References

- [1] Stéfan van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and The scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 2014.
- [2] William D Callister Jr. Materials Science and Engineering - An Introduction (5th ed.). *Anti-Corrosion Methods and Materials*, 47(1), 2000.
- [3] Andrew Reid, Rhonald Lua, Edwin García, and Valerie Coffman. Modelling microstructures with oof2. *International Journal of Materials and Product Technology*, 35:361–373, 01 2009.
- [4] Harry Shipley, Darren McDonnell, Mark Culleton, R. Lupoi, G. O'Donnell, and Daniel Trimble. Optimisation of process parameters to address fundamental challenges during selective laser melting of ti-6al-4v: A review. *International Journal of Machine Tools and Manufacture*, 128, 01 2018.
- [5] Robert Cimrman and Eduard Rohan. Two-scale modeling of tissue perfusion problem using homogenization of dual porous media. *International Journal for Multiscale Computational Engineering - INT J MULTISCALE COMPUT ENG*, 8:81–102, 01 2010.
- [6] E. Rohan and V. Lukeš. Homogenization of the acoustic transmission through a perforated layer. *Journal of Computational and Applied Mathematics*, 234(6):1876 – 1885, 2010. Eighth International Conference on Mathematical and Numerical Aspects of Waves (Waves 2007).

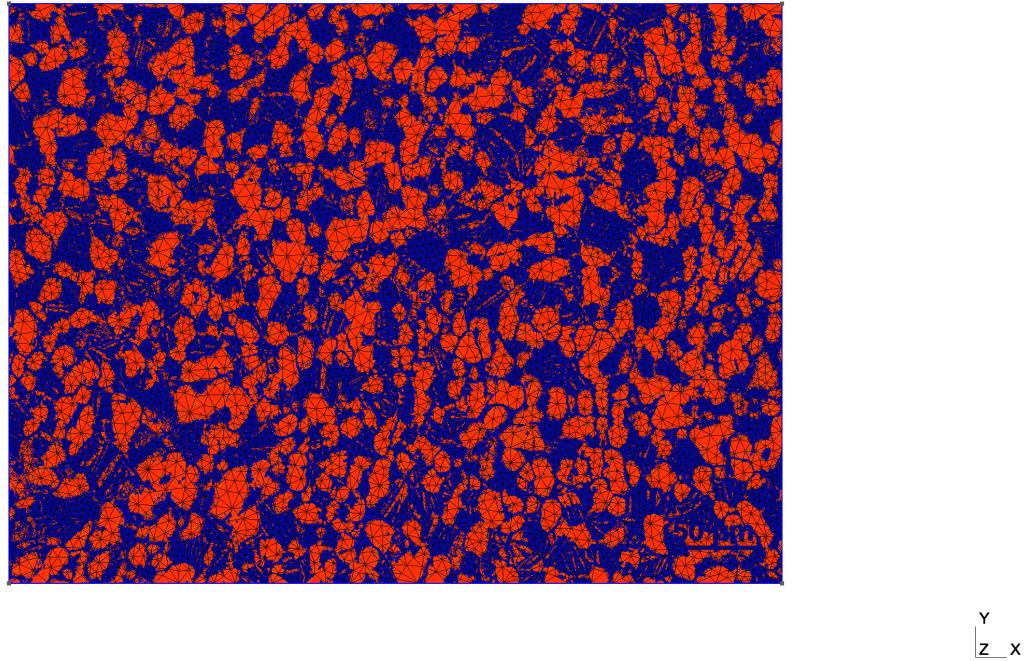


Figure 17: Gmsh adaptive mesh generated for image: Ti64BFi20xbinary.png

- [7] Robert Cimrman, Vladimír Lukeš, and Eduard Rohan. Multiscale finite element calculations in python using sfepy. *Advances in Computational Mathematics*, 45(4):1897–1921, 2019.