**Finite Element Analysis of Bone-Anchored Limbs**

**Standard Operating Procedure**

This document provides details of how to create and analyze finite element models of the residual limb of individuals with bone-anchored limbs (BALs). Instructions for designing a finite-element (FE) model of the residual limb prosthetic anatomy, finite element analysis (FEA) of the FE model, and determination of stresses are given. This method requires running a preliminary musculoskeletal (MSK) model in OpenSim to determine patient-specific muscle and joint force vectors throughout the task of interest. These forces are applied to the designed FE model to determine a temporal view of stresses over time. The overall process has eight steps:

Step Zero: Pre-start checklist

Step One: Determination of joint and muscle vectors (p. 2)

Step Two: Image Registration (p. 3)

Step Three: Image Segmentation (p. 6)

Step Four: Positioning the Femur and Adding PMMA: Stance and Fall (p. 8)

Step Five: Building the FE Meshes (p. 13)

Step Six: Preliminary Finite Element Analysis (p. 17)

Step Seven: Add elastic-perfectly plastic material properties to bone elements (p. 19)

Step Eight: Final Finite Element Analysis (p. 20)

A collage of several images of human body parts

Description automatically generated

**Step 0: Pre-analysis checklist**

Before starting this SOP, several files (listed below) are necessary for analysis. You can get these from your associated gait analyst.

* DICOMImages/CT: A computed tomography (CT) scan of the entire residual anatomy is necessary for FE model generation. Post-operative or pre-operative scans are both acceptable, but note that small errors in bone material property estimation may become apparent due to metal artifact from implant noise in post-operative CT scans will be introduced.
* DICOMImages/Radiograph: A post-operative radiograph of the patient is required for the reconstruction of implant geometry.
* MSK/.osim file: A OpenSim model file, which holds musculoskeletal modeling data for the individual participant, is required for calculating muscle and joint force vectors.
  + MSK/Model\_Geometries: several files that hold bone geometries for OpenSim help visualize the main .osim file.
  + While not necessary for analysis, they help visualize patient anatomy.
* MSK/Model\_Geometries/residualLimbGeometry.STL: The STL file holding the residual limb anatomy
* MSK/[…]\_StaticOpimization\_force.sto: This .sto file holds muscle force magnitudes, calculated using MSK modeling, throughout the activity of interest.
* MSK/[…]\_Kinematics\_q.sto: This .sto file holds joint force magnitudes, broken into individual axis components (i.e., inferior-superior, medial-lateral, and posterior-anterior).
* MSK/.STL

**Step 1: Gather Muscle and Joint Force Data**

The purpose of the following step is to extract 2 files of importance for the proceeding finite element analysis: 1) muscle direction vectors (an Excel sheet of x, y, and z force components for each muscle force), and 2) joint direction vectors (an Excel sheet of x, y, and z force components for each joint)

1. Open OpenSim and load the associated OpenSim model file, by navigating to “file” (in the top left), then selecting “Open Model” in the drop-down, and navigating to the correct .osim file in the patient directory (normally found in the “ModelGeometries” folder)
   1. Please refer to <https://simtk.org/projects/opensim> for information on download procedures of OpenSim.
2. Associate a partnering motion file (several options are acceptable, including the static optimization and RRA kinematic data files) by navigating to “file” (in the top left), then selecting “load motion” in the drop-down, and navigating to the correct .sto file in the patient directory (normally found in the “RRA\_Kinematics” folder)
   1. Select the “play” button in the top middle of the OpenSim software to verify that the motion was appropriately synchronized with the OpenSim model (if the model starts moving erratically, either the model or motion is incorrect)
3. Go to the tools button, and scroll down to user plug-ins, and select the MuscleForceDirection.dll plug-in, load it into the model
   1. If the user plug-ins option is available, or MuscleForceDirection.dll isn’t available, it probably hasn’t been downloaded into your OpenSim software: please use the instructions in appendix A.1 (“Installing the MuscleForceDirection.dll plug-in”)
4. Load the MuscleForceDirection.dll plug-in, by going to the Tools tab and selecting the ”MuscleForceDirection.dll” option in the bottom.
5. Create a new analysis by going to the Tools tab, and selecting analyze.
6. Go to the analyze tab, select add, and load the MuscleForceDirection analysis.
7. Highlight the MuscleForceDirection analysis, go to options, and set up the analysis per figure 1

A screenshot of a computer

Description automatically generated

*Figure 1: MuscleForceDirection analysis*

1. Run the analysis, and move the 2 resulting files (titled “[OpenSim model name]\_MuscleForceDirection\_attachments.sto” and “[OpenSim model name]\_MuscleForceDirection\_vectors.sto”) to the model file location you have chosen.
2. Open each file in Excel by:
   1. Opening a black Excel sheet
   2. Navigate to “file”, then “open…”
   3. Change “all excel files” to “all files” in the bottom right search box
   4. Navigate to the location of the .sto file
   5. Select, and follow the prompts with the default options
   6. After importing the file, remove all header rows, leaving only the column titles row and values row (see figure 2)
3. Save the file per the below criteria
   1. If the original .sto file ends in “\_attachments.sto”, save the file as “muscleAttachments.xlsx”
   2. If the original .sto file ends in “\_vectors.sto”, save the file as “muscleVectors.xslx”

A screenshot of a computer

Description automatically generated

*Figure 2: Finished Excel sheet of OpenSim data*

1. Open the calculated muscle magnitude file .sto file (normally in the “SO\_forces” folder) and open the file in Excel using the directions listed in step 9 above, and save the file as “muscleMags.xlsx”
2. Open all (normally two files; a “hip” and “knee” file) the calculated joint magnitude .sto files (normally in the “JRFs” folder), and concatenate the two folders together by copying all the rows and columns from one file, and appending them to the end of the other file (make sure that the time points are synchronized!): save the resulting file as “jointMags.xlsx”

You should now have four files, all saved to the same location

1. “muscleAttachments.xlsx”
2. “muscleVectors.xlsx”
3. “muscleMags.xlsx”
4. “jointMags.xlsx”

**Step 2: Calibrating the Muscle and Joint Force Data**

Once all the necessary joint and muscle data has been successfully pulled from OpenSim, it needs to be organized. As muscle forces and joint forces may have different timepoints attributed to them, the arrays need to be interpolated and organized to have the same timepoints for FE analysis. After these muscle and joint forces have been calibrated, they will be imported into the final stress analysis script for Abaqus to reference during it’s FE analysis.

1. Open the “PythonCodev2.ipynb” Jupyter Notebook file in your IDE of choice.
   1. PythonCodev2.ipynb will import the muscle magnitudes, muscle directions, and joint magnitudes from their respective Excel sheets and orient them.
   2. NOTE: this script requires the Numpy, Matplotlib, and Pandas libraries to be installed on your local computer to run correctly. If you do not have them installed, please refer to Appendix 2 for directions on how to install a python library.
2. In cell 2 (the node with the comment “#User input data”, change the directory paths of the files to the locations of the muscleMags, muscleVecs, and jointMags files you have saved locally on your computer. (In figure below, the section to be changed will be the lines in orange) (NOTE: color scheme may be different for your computer, i.e., not orange).
3. In cell 2, change the residualLimbName variable to the name of the residual limb bone *exactly as it appears in it’s corresponding OpenSim file*. For example, for a patient with a transfemoral amputation of the right limb, the residual bone might be named “femur\_r”.
4. In cell 2, change the distalProstheticName variable to the name of the prosthetic device attached to the residual limb bone in OpenSim *exactly as it appears in it’s corresponding OpenSim file*. For example, for a patient with a transfemoral amputation of the right limb, the distal prosthetic might be named “walker\_knee\_r”.

A computer screen with a black background

Description automatically generated

1. Select the “run all” option: this is done to populate the results arrays.
2. Scroll down to the bottom of the file. A very long string of characters has been printed. Copy all of the lines, and copy them into the final stress analysis Python script between the two long double-asterisk (#\*\*\*\*\*) lines.
3. Save the stress analysis script, and close the analysis Jupiter file.

**Step 3: Calibrating the Computed Tomography Scan to Calculate Material Properties**

The pre-operative computed tomography scan must be calibrated and converted to material properties in order to design a accurate finite element model of the residual anatomy.

1. Open the “CalibrateCTv12.ipynb” file in your IDE of choice.
   1. NOTE: this script requires the Numpy, Matplotlib, Pydicom, OS, and ipywidgets libraries to be installed on your local computer to run correctly. If you do not have them installed, please refer to Appendix 2 for directions on how to install a python library.
2. In cell 2, under the “User Input” section, change the parentDirectory variable to the directory path of the CT scan (the folder with the large number of individual DICOM scans). NOTE: please leave a backward slash “/” at the end of the directory path, otherwise the code will not correctly run.
3. In cell 2, under the “User Input” section, change the saveDirectory variable to desired save location of the calibrated CT scan (the folder with the large number of individual DICOM scans). NOTE: please leave a backward slash “/” at the end of the directory path, otherwise the code will not correctly run.
4. In cell 2, under the “User Input” section, change the patient variable to the current patient name being analyzed.
5. In cell 2, under the “User Input” section, change the identifier variable to the constant beginning character string of the DICOM images. For example, if CT image titles are IM\_001, IM\_002, IM\_003, and IM\_004, the identifier will be “IM\_”.
6. Run the “Import needed libraries”, “User input”, and “Scan import” sections.
7. Run the first two cells of the “Plot uncalibrated scan” section. NOTE: there is a small bug that periodically makes this section run indefinitely. If this occurs, please restart the file and re-run the sections.
8. An interactive plot like the figure below will be visualized. The blue square is a region that is sampled for an average value. Use this region to select a homogenous region of air, fat, and muscle (left, middle, and right figures below), and replace the “air”, “fat”, and “muscle” variables with the sampled values from the scan. To do this, use the sliders above the image to change the sampling square position, and copy the value just above the scan.
   1. The “j” slider changes the image slice.
   2. The “a” slider changes the vertical position of the sampling square.
   3. The “b” slider changes the horizontal position of the sampling square.
   4. The “L” slider changes the sampling square size. You should increase the square as much as possible while keeping the material being sampled homogenous.

A close-up of a x-ray

Description automatically generated A close-up of a ct scan

Description automatically generated A close-up of a ct scan

Description automatically generated

1. Once the “air”, “fat”, and “muscle” variables have been updated on the scan, run the “Calibrate scan and export” section. Two .raw files will be exported: one holding the calculated bone density values (ending in “\_AFM\_p.raw”), and one holding the calculated material strength values (ending in “\_AFM\_E.raw”).

**Step 4: Import the Residual Limb Model**

A finite element model of the residual limb anatomy, including the bone-anchored limb, must be designed so use in Abaqus for stress analysis.

1. Open ScanIP, select the “Stack or raw images” option under the “New project from” ribbon.
2. Select the “select file(s)…” option in the top left of the pop-up that appeared, and navigate to the RAW file holding calculated material strength values that was created using the medical image calibration script of the last step.
3. Fill in the “Spacing” data of the original CT scan you used. If you do not know what these values are, refer to appendix 3 for instructions on how to get this information.
   1. The first section on the right, titled “Spacing” should be filled with the appropriate scan voxel dimensions (voxel x size, voxel y size, and voxel z size, if using variables from appendix 3).
   2. Please make sure that the “Treat as raw data” option is checked!
   3. Under the “Raw image settings” section, change the pixel type to “signed short (16 bit)”
   4. Fill in the “Size – X (pixels)”, “Size – Y (pixels)”, and “Size – Z (pixels)” boxes in with the width, length, and depth of the scan dimensions (in pixels). If using data from appendix 3, these values will correspond to variables “image x voxels”, “image y voxels”, and “image z voxels”, respectively).
4. NOTE: once all the blank boxes have been filled in correctly, a small “ok” should appear below the “Check:” line (seen in the “File size check” section).
5. After importing the scan successfully, save it locally to your computer.

A screenshot of a computer

Description automatically generated

**Step 5: Segment and Orient the Residual Limb Model**

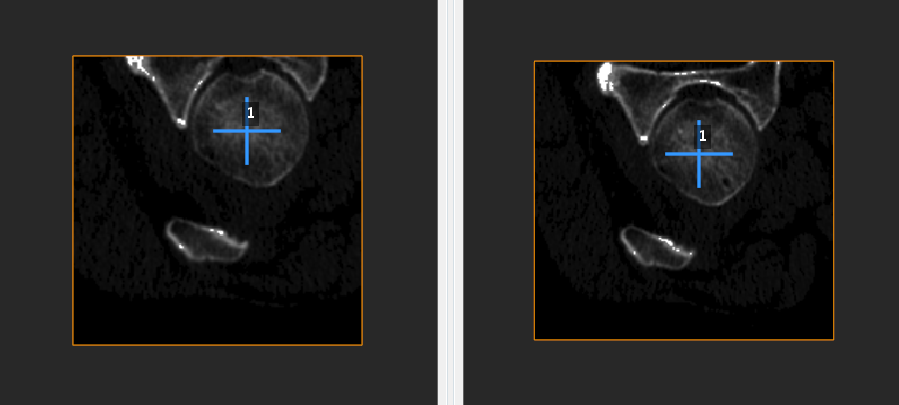
After importing the calibrated image, the pre-segmented residual limb model used in the OpenSim analysis needs to be oriented with the residual limb anatomy seen in the scan. Next, a model of the bone-anchored limb needs to be oriented along the medullary canal of the residual limb. Lastly, a finite element model of the entire mask series needs to be created and exported for analysis in Abaqus.

1. Open a blank OpenSim image, with a size of 1000x1000x1000 pixels, with each pixel having a 1mmx1mmx1mm size.
2. Navigate to file -> import -> STL/CAD file
3. Navigate to the location of the STL model of the residual limb anatomy used in OpenSim.
4. In the pop-up window, change the “Factor (to mm)” option to 1000. There is no need to change any of the other variables on the importation pop-up window that appears: just click the import button at the bottom right of the pop-up.
   1. NOTE: if there are any geometrical issues with the model, OpenSim will ask if it can correct the errors: you can say YES if that is an option.
5. A new surface should appear in the “surfaces” dropdown of the Dataset browser. At the top of the screen, navigate to the “Surface Tools” ribbon, and select the “Surface to mask” tool. (the icon looks like an arrow pointing from an orange set of triangles to a green staircase on a checkered background). On the toolbox that appears at the left of the screen, select “Generate mask”. OpenSim will probably ask if “padding” is acceptable: it is ok to select the YES option.
6. Save the model file somewhere locally on your computer.
7. Next, open the ScanIP file that holds the calibrated medical image data.
8. Perform a *rough* segmentation of the residual limb anatomy. This segmentation does not need to be precise: it will simply be used to help orient the OpenSim residual limb model. You may attempt to orient the OpenSim residual limb model directly to the medical image scan, but this rarely succeeds. You may use whatever ScanIP tools best help you to quickly draft a segmentation.
9. Save the model, and close the file.
10. Open the OpenSim residual limb geometry model.
11. Under the “Image Segmentation” ribbon, select the “register datasets” window (above the “transforms sub-ribbon) (the icon looks like a google maps thumbtack)
12. In the “register dataset from” option, select “foreign object”. A navigation window should appear: navigate to the ScanIP model that holds the calibrated CT scan, and select open: lastly, select the mask option, and click next at the bottom right of the pop-up window.
13. With “Registration method” set to “Landmark and automatic”, apply four landmarks (listed below) on both models. NOTE: changing the orientation frames (seen just to the left of each slice slider), may help visualize the images better. Also note that the model orientation and image orientation are most likely NOT closely aligned yet, so be careful in placing markers on the two models.
14. To add landmarks for registration, click “Add landmark.” Click on a point in the baseline image (left window), and then click on the same point in the follow-up image. On transfemoral models, use the following four landmarks (**Fig. 2**):
    1. Center of the femoral head
    2. Tip of the lesser trochanter
    3. Tip of the greater trochanter
    4. Proximal origin of the calcar

On transtibial models, use the following landmarks:

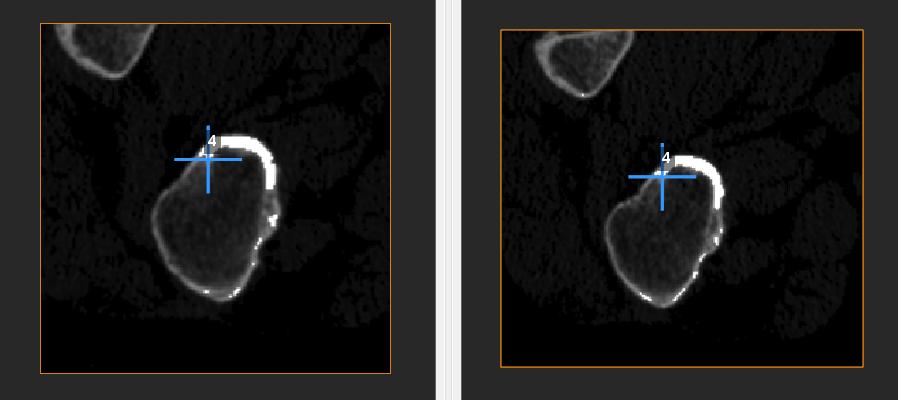
* 1. Left condyle
  2. Right condyle
  3. Intercondylar eminence
  4. Tip of deltoid tuberosity

A close-up of a person's body

Description automatically generated

A comparison of a pair of images

Description automatically generated with medium confidence



1. Once the landmarks have been placed, select the next button. The orientation *should* be successful. If it is not, attempt other techniques (including only landmark segmentation or only automatic segmentation) to get the OpenSim model to closely orient with the residual limb scan.
2. Verify that the OpenSim model has been appropriately oriented with the medical image by scrolling through the image stack to verify that there is no drift in overlaying of the model.
3. Next, repeat steps 2 through 4 to import the designed bone-anchored implant into the model file.
4. Go to the surface tools ribbon, and select the “position and rotation” tool: rotate and position the bone-anchored implant to sit in the middle of the medullary canal, with the abutment sitting just distal to the end of the residual limb.
5. Next, use manual segmentation tools to split the residual limb anatomy into individual sections. Transfemoral models should be segmented into 5 sections:
   1. Femoral head (green in figure below) (title the mask “femoral head”)
   2. Femoral neck (brown in figure below) (title the mask “femoral neck”)
   3. Trochanteric (yellow in figure below) (title the mask “trochanteric”)
   4. Subtrochanteric (red in figure below) (title the mask “subtrochanteric”)
   5. Diaphysis (pink in figure below) (title the mask “shaft”)

Transtibial models should be segmented into 3 sections:

* 1. Proximal epiphysis (pink in figure below) (title the mask “epiphysis”)
  2. Distal epiphysis (yellow in figure below) (title the mask “metaphysis”)
  3. Diaphysis (green in figure below) (title the mask “diaphsysis”)

A close-up of several parts of a bone

Description automatically generated

1. Next, a cortical shell around the bone-anchored implant needs to be created.
   1. Select the bone-anchored limb, create a copy of the mask. Name this mask “interface” (if transfemoral) or “shell” (if transtibial). Use the dilate tool in the morphological sub-ribbon of the image processing ribbon to dilate the copy of the bone-anchored limb mask axially (i.e., not along the longitudinal axis of the limb). The settings for the dilation will most likely be those below.

A screenshot of a computer

Description automatically generated

* 1. Use Boolean operations to subtract the original implant mask from the cortical shell mask.
  2. Subtract both the implant and cortical shell masks from any bone masks that are in contact with the cortical shell and implant masks.

1. Lastly, apply a PMMA mask to the proximal head of the residual limb for uniform application of forces in the FE stress analysis.
2. Next, highlight all of the masks generated (bone, implant, and PMMA masks) and create an FE model that includes all of these masks.
3. Right click on the generated FE model, and select “model configuration”.
   1. Under the “general” tab, in “export options” section, change the coordinate system to “Global”.
   2. Navigate to the materials tab. For each mask, assign the appropriate material property profiles from below. (top = PMMA, middle = bone, bottom = implant metal).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Generate and export the FE model to your local machine.

**Step 6: Set up final analysis script**

Once the FE model has been generated, we need to finalize the setup of the FE stress analysis script.

1. Open the .inp file generated from the previous step in notepad.
2. Save this notepad file as a separate text file (I suggest “modelNodes.txt”, or something similar).
3. Delete all data from the script that is not the node coordinate rows. (see figure below for highlighted region that should stay).

A screenshot of a computer

Description automatically generated

1. Open the Jupyter notebook script “getMuscleNodes.ipynb”.
2. In the “Step 1” section, change the “muscleCoordsFile” and “nodesCoordFile” variables to the directory locations of the Excel sheet holding muscle insertion coordinates, and the .txt file holding the node coordinates (generated from the steps just above).
3. Scroll down to the cell holding the “muscle\_names” variable, and replace the list with the muscle\_names list generated in pythonCodev2 (from step 2).
4. Scroll to the bottom of the file: three long lists will have been generated. Copy the three lists, and paste them into the corresponding muscle coordinate variable lists in the “loadApplication” python file.

**Step 7: Run the final stress analysis**

1. Open Abaqus CAE
2. Import the FE model generated from step 5 into the ABAQUS session using File > Import > Model.
3. Create a new analysis step (Static, General).
4. Apply a fixed (U1 = U2 = U3 = 0) boundary condition to the top surface of the PMMA cap on the model.
5. Generate a displacement load condition to the bottom surface of the bone-anchored limb (NOTE: any displacement value is acceptable: these values will be updated automatically before each analysis).
6. Open the “loadApplication” python file: and change the modelName variable to the title of the model that was imported into Abaqus CAE.
7. Go to run -> script -> and navigate to the loadApplication python script.
8. Open the final analysis script, and change the following variable values:
   1. modelName: to the name of the model imported into Abaqus
   2. distalLimbName: to the name of the load application applied to the bottom surface of the bone-anchored limb.
   3. baseDirectory: to the directory address of the desired folder for stress reports to be saved to.
9. Go to run -> script -> and navigate to the final stress analysis python script.
10. The final analysis may take some time to run (several hours is not uncommon).

**Appendices**

Appendix 1: Installing the MuscleForceDirection.dll plug-in

1. Make sure the plugin has been built and tested on your platform. If you're unsure of your platform, go to Help>About OpenSim>Details. The plugin must have been created using the same version of OpenSim (e.g. 4.0) and build system (e.g. Visual Studio 2017). Plugin distributors should supply this information to users. The remaining steps appply to Windows only, for Mac please go to step 7 directly.
2. Close OpenSim.
3. Download the MuscleForceDirection.dll plug-in from <https://simtk.org/projects/force_direction>
4. Place the plugin under the plugins folder within your OpenSim installation folder (if you installed this in the default location on Windows, this will be C:\OpenSim <version number>\plugins).
5. Launch OpenSim. You should then see the plugin as a menu option under the User Plugins menu.
6. From the User Plugins menu, click the name of the plugin to load it into OpenSim. You will be given the option to always preload the plugin each time OpenSim is launched. It is advised that you not do this until the plugin has been tested.

Appendix 2: Installing a Python library

1. Open a Command Prompt or Terminal by searching for "Command Prompt" in the Start menu and open it.
2. Verify pip Installation (pip is the standard package installer for Python). To check if it's installed, run the following command:

*pip --version*

If it's not installed, refer to the Python documentation for installation instructions.

1. Install the Library by using the pip install command followed by the library name:

*pip install <library\_name>*

Replace <library\_name> with the actual name of the library (e.g., numpy, pandas).

If no errors occur, the library is installed correctly.

Appendix 3: Get spacing and pixel data from a CT scan

To determine the image spacing of the following variables (voxel x size, voxel y size, voxel z size, image x voxels, image y voxels, and image z voxels), and pixel settings of the original CT scan that was used for analysis:

1. Open ImageJ (a simple, free medical image viewing software)
   1. NOTE: if you do not have ImageJ downloaded locally on your device, you can get a version from [here](https://imagej.net/ij/)
2. Go to file -> Import -> Image Sequence, and navigate to your folder location of the uncalibrated CT image for the patient being analyzed.
3. Press Ctrl + I (capital i), or navigate to Image -> Show Info…
4. Navigate to the rows that start with “Width: “, “Height: “, and “Depth: “. The values to the right of these row headers (*in parentheses:* this is important to distinguish, as the values not in parentheses are length measurements, which is not what we are looking for) are the image x voxels, image y voxels, and image z voxels, respectively. Please record these values, as they are necessary for image importation in scanIP. (in the example figure below, these values will be 512, 512, and 413 voxels, respectively).
5. Navigate to the row that starts with “Voxel size:”. A 3 number string is provided, in the format “voxel x size”mm x “voxel y size”mm x “voxel z size”mm. Please record these values, as they are necessary for image importation in ScanIP. (in the example figure below, these values will be 0.5, 0.5, and 0.9 mm, respectively).

NOTE: the rows to be searched for can be easily found by copying the text from the ImageJ popup, copying it into a notepad file, and using the search (Ctrl + F) key. Alternatively, the values are found at near the bottom of the file.

A screenshot of a computer program

Description automatically generated