

Last updated: 2025/04/13 – Max Hockenberry

The following instructions will guide you through processing the provided example 3D TFM dataset.

Before getting started, you will need to:

- Download the latest version of the Legant Lab 3D TFM code @ XXX
- Download MATLAB (most modern versions should work fine but haven't been rigorously tested. We use MATLAB 2024b).
- Download the following MATLAB toolboxes:
  - o Statistics and machine learning
  - o Curve Fitting
  - o Optimization toolbox
  - o Parallel computing toolbox
  - o Global optimization toolbox
  - o Image processing toolbox
  - o Partial Differential equations toolbox
  - o Computer Vision toolbox

## Example dataset contents:

- 3D volumetric data set of a cytoplasmic GFP, SiR-DNA stained fibroblast migrating through a 5-micron wide, 40-micron long confinement. Individual channels are separated into separate folders while the bead folder also contains the reference frame where the cell is removed with the detergent SDS (Cell, DNA, and Beads folders).
- 2D masks of the cell shape and nuclei derived from cytoplasmic GFP and nucleus signal (Cell\_mask and DNA\_mask folders).
- Averaged swelling profile derived from imaging several confinements without cells present, measuring the SDS induced geometry specific swelling, and averaging the profiles (averagedSwellingProfile5x40.mat).
- 5x40 confinement FEA geometry in Abaqus format derived from hypermesh (5x40SolverDeck.inp).
- 5x40 confinement FEA nodal displacements solution in two parts (nodalDisplacements1.mat, nodalDisplacements2.mat).
- Example data set outputs including processed displacement field (ExampleDataSetDisplacements.mat), processed traction vectors (2024-11-11\_ComputedTractionAlignedValues.mat), quality control plots (QA Plots folder), aligned STL of the substrate geometry with input images for 3D visualization (alignedSTLs folder), Amira traction and displacement vector and traction magnitude output folders (2024-11-11\_Disps, 2024-11-11\_Traction, 2024-11-11\_TractionContour).

## A note on FEA modeling:

We made use of hypermesh and optistruct to perform our FEA modeling but most FEA software should in principle be capable of reproducing the analysis necessary to compute the geometry specific Green's Function. In general, you will need to compute a solution that contains the nodal displacements of each node (of the elements) in the data set for a unit point force in the X, Y, and Z directions. Thus, for a model with 1000 nodes, you would expect a 3000 x 3000 matrix (1000 Nodes \* X \* Y \* Z displacements by each force in X Y Z). This matrix can then be used as the Green's matrix assuming the input geometry and mechanical properties remain consistent with experimental data. For a more detailed analysis, consult the original 3D TFM manuscript:

<https://doi.org/10.1038/nmeth.1531>.

## Computing Displacement Fields:

1. Launch MATLAB and set the path to include all the folders and subfolders in TFM.
2. Run main\_tracking\_script.m
3. A GUI will open to facilitate input of data. Fill out the fields with paths to data, voxel sizes in microns, and approximate parameters for the MLE bead fitting algorithm. Press continue.

UI Figure

Post Treatment File  C:\Users\Max\Documents\GitHub\TI

Pre Treatment Folder  C:\Users\Max\Documents\GitHub\TI

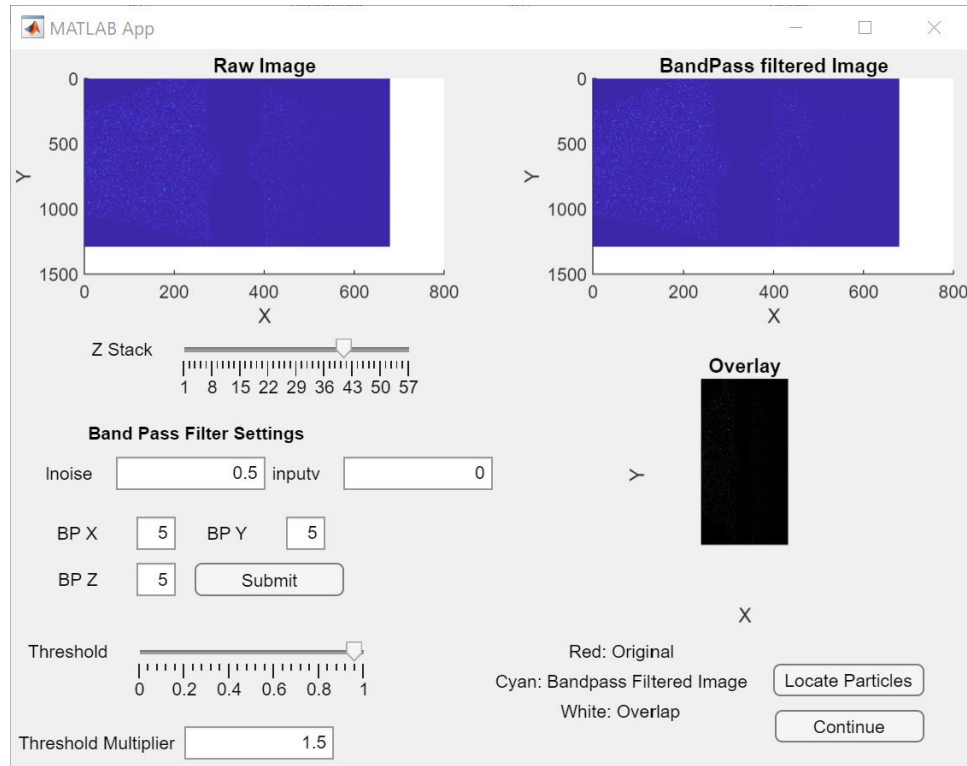
Analysis Output Folder  C:\Users\Max\Documents\GitHub\TI

Analysis Name

Voxel Size (um)		Bead Parameters	
X	<input type="text" value="0.2"/>	Threshold	<input type="text" value="0"/>
Y	<input type="text" value="0.2"/>	Min. Size	<input type="text" value="3"/>
Z	<input type="text" value="0.8"/>	Max Size	<input type="text" value="1000"/>
		Win. Size	<input type="text" value="5"/>
		Intensity	<input type="text" value="4500"/>
		Multiplier	<input type="text" value="1.5"/>

- a.
4. After a moment, the band pass filtering GUI will open. This GUI sets the parameters for the preprocessing bandpass filter. Here you can control the size in pixels of bandpass in each direction, the level of expected noise (lnoise), and the clipping on bandpass filter (inputv). For most applications adjusting the bandpass filter size for the approximate size of the bead

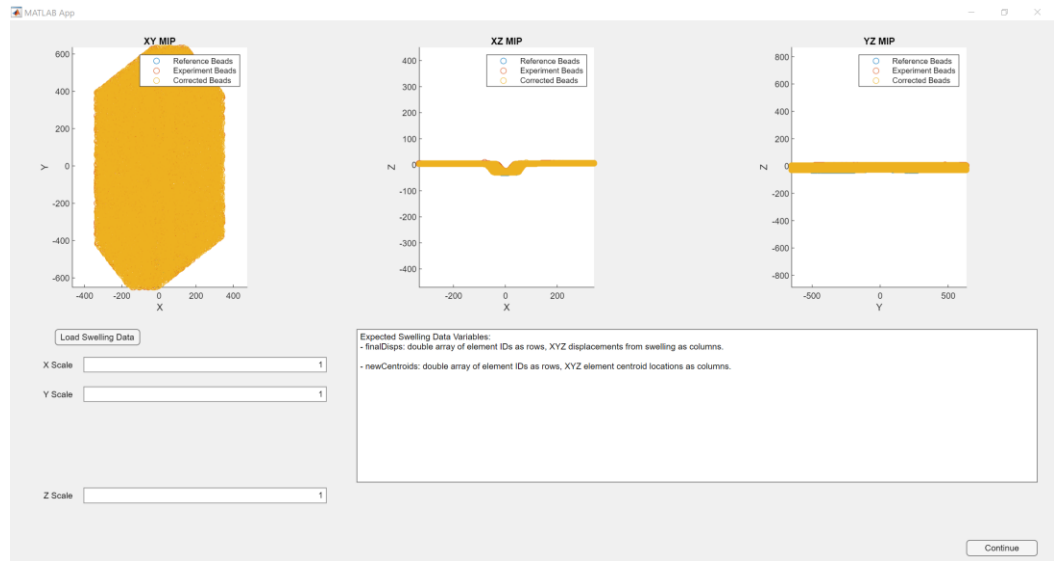
3D PSF and the threshold of the bead grayscale values is sufficient. The objective is to get a set of parameters such that the bandpass filtered image on the top right retains the beads, but has significantly reduced noise which may be erroneously detected as beads. You can press locate particles to run the bead localization script (this is slow) which will show cross marks on the overlay image corresponding to the found bead positions as a way of assessing the performance of the bandpass GUI settings. Press continue.



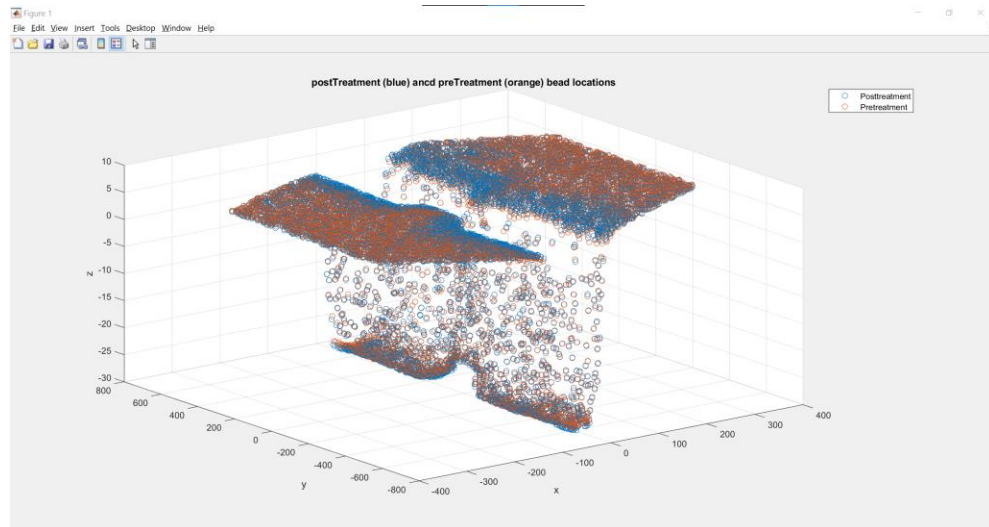
a.

- The next GUI that opens handles the swelling correction. Here you press load swelling data to provide a .mat file containing a swelling correction dataset that is interpolated to the experimental data to correct swelling induced by addition of a detergent to remove cells. This moves the reference bead positions which can be evaluated by the three maximum intensity projections figures. The scaling of the swelling correction in each dimension can be controlled by the x, y, and z scale fields. In practice, this is unnecessary assuming the swelling correction was gathered on the same substrate that the experimental data was taken or on substrates of equivalent stiffness and response to swelling. Press continue.

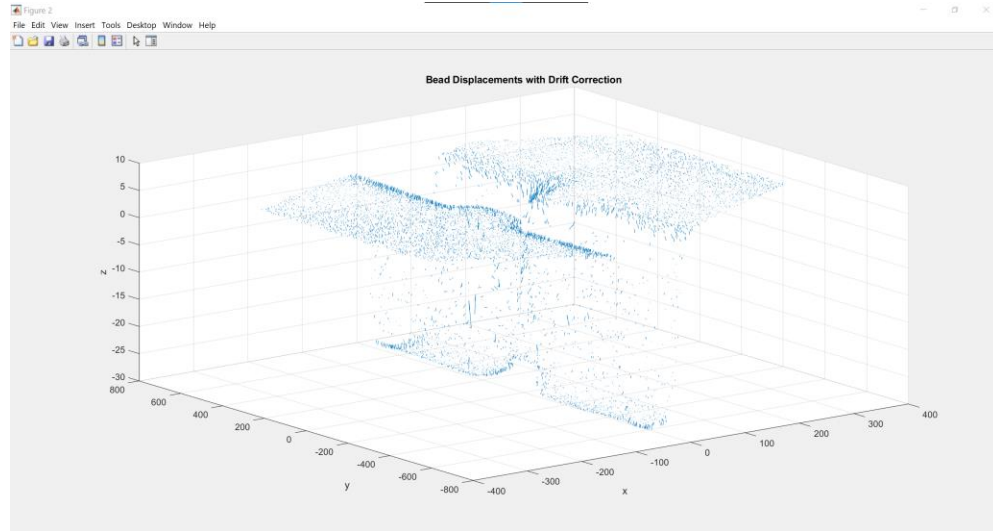
## Legant Lab 3D TFM Matlab Code Example Instructions



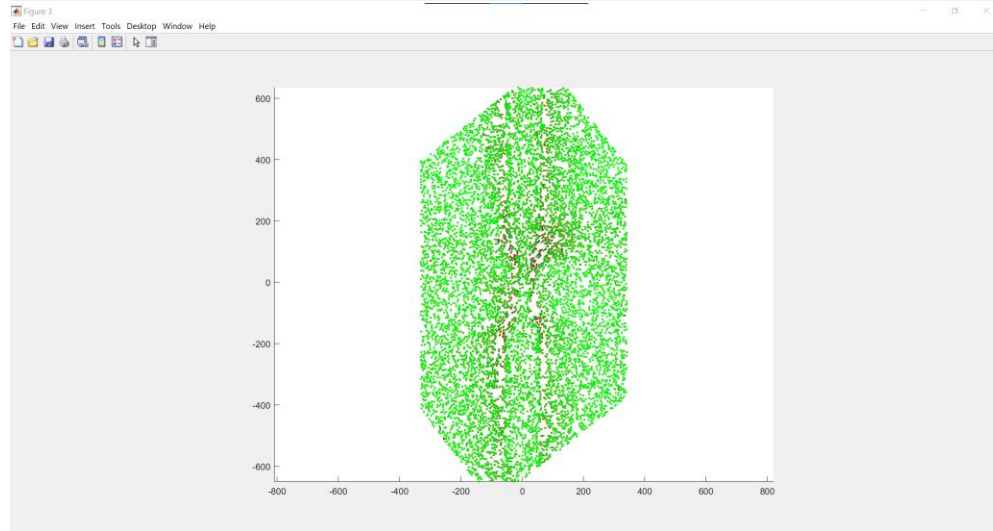
- a.
- From here the particle linking algorithm will link all particle localizations by comparing the experimental images to the reference image. This has been set up assuming a set of timelapse images and has thus been parallelized. The progress of the script can be seen in the MATLAB command window. Once complete, the script will output the following figures and save the output variables into the folder specified in the input GUI.



a.



b.

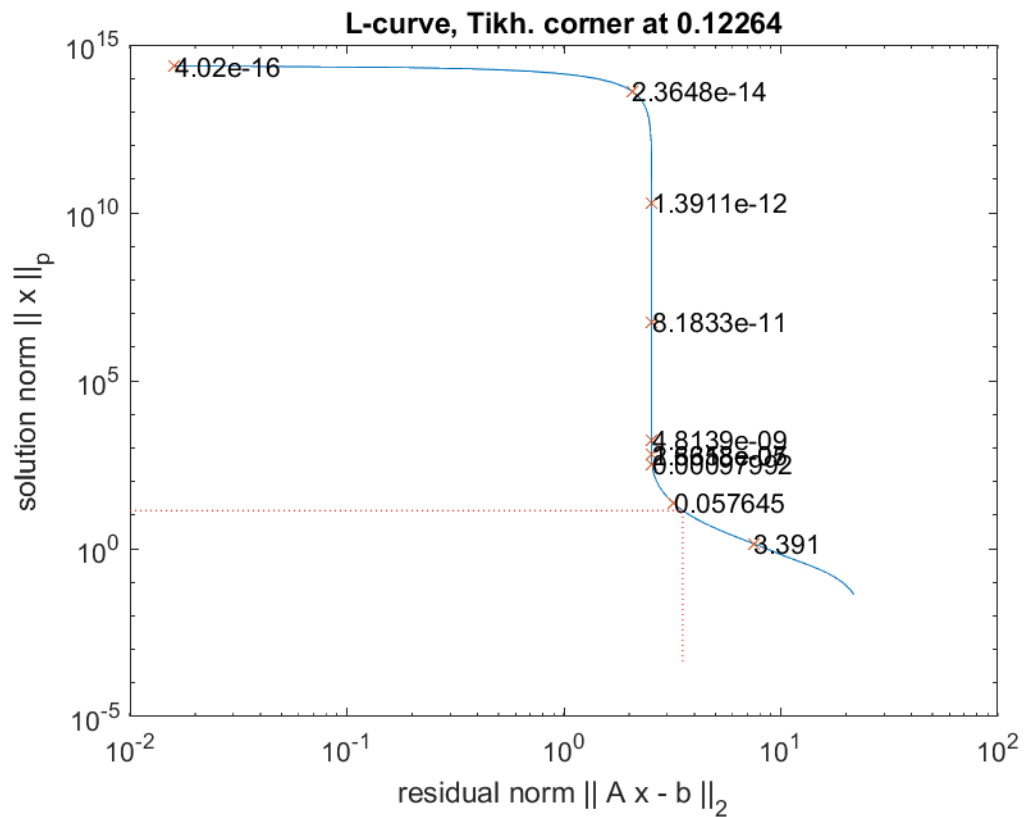


c.

7. A final note regarding the bead linking algorithm. In the current release of the code, the settings for the algorithm are hard coded into the trackBeads.m script and can be adjusted manually there if desired. The parameters found here worked generally for the substrates and geometries tested in this manuscript but may need to be adjusted if the displacements substantially differ in magnitude, density of localizations, or noise. This can be done on lines 97-104. A description of these parameters can be found in displacementCalc\_recursive\_Max\_2024\_02\_19.m and smoothFilt\_2024\_02\_20.m. A troubleshooting figure is left commented out in line 219 of the trackBeads.m script that can be used to evaluate the performance of the parameters if desired (just add a stop prior to line 219, uncomment and adjust parameters and rerun lines 203-221).

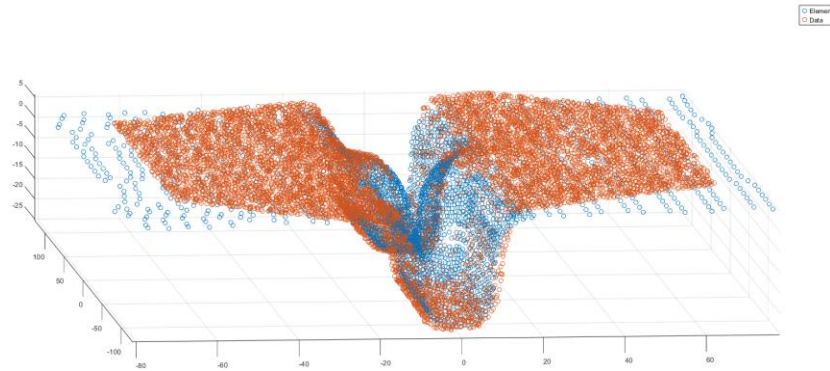
## Computing Traction Fields:

1. Open processTFM\_General\_2024\_11\_11.m and adjust the following variables:
  - a. outputFold (Line 39): The folder path to where you would like the output files and figures saved.
  - b. voxelSize (Line 45): The size of the voxel in X,Y, and Z in microns.
  - c. fileName (Line 47): The path to the geometry file for the given geometry.
  - d. disps (Line 66): The path to the displacements produced by main\_tracking\_script.m
  - e. Masks (Line 91): The path to the folder containing the masks of the cell boundary for masking the displacements.
  - f. dilation (Line 102): The amount of dilation in pixels for the mask of the cell boundary.
  - g. scale (Line 219): The Young's modulus (in Pa) of your substrate assuming your Green's matrix was computed for a rigidity of 1 Pa.
2. Evaluate the L curve based on your data (Lines 187-197). Choose a value for regularization and set as reg\_corner\_timelapse (Line 201).

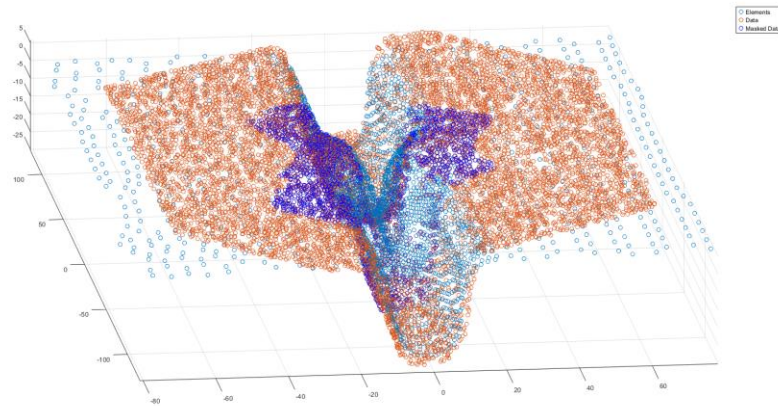


- a.
3. Several quality control figures are outputted to assess varying steps of the traction reconstruction process. Examples are below:

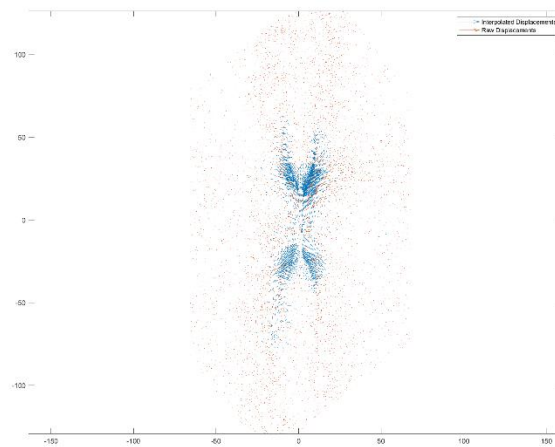
a. Model and data alignment



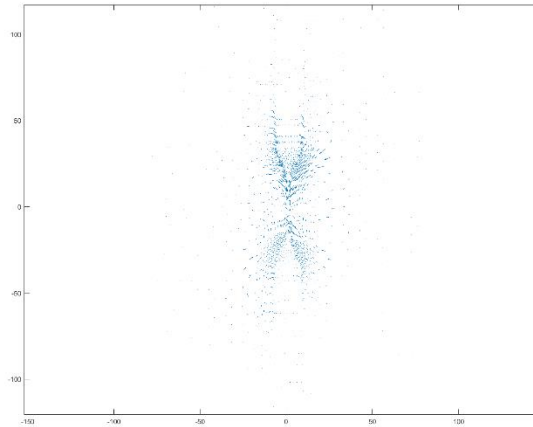
b. Masked Data



c. Interpolated Displacements



d. Traction Field



4. The following variables and outputs are produced by the script:
- a. Amira vector fields containing the displacements, tractions, and traction magnitudes.
  - b. An aligned STL for the substrate geometry model and the experimental data.
  - c. `traction_vector`: A cell array containing a double array for each timepoint with an  $n \times 6$  array where  $n$  is the element number and columns 1:3 are the XYZ coordinate of the centroid of the element and 4:6 are the traction measurement in X, Y, and Z respectively.
  - d. Various other variables corresponding to displacements, element centroids and areas, etc. Consult the documented script for more details and examples of their usage.

## Important Notes:

Don't hesitate to reach out if you run into issues running this code. You should always critically evaluate the results of these functions and scripts to ensure accuracy. This code is provided under an MIT license:

### MIT License

Copyright (c) [2025] [Max Hockenberry]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.



THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.