# STRESS: Surface Topography Reconstruction for Evaluation of Spatiotemporal Stresses

STRESS is an Open Source software package developed in MATLAB and Python for the quantification of the geometry of deformable particles of spherical topology, such as microdroplets or gel microbeads, and enables the automatic quantification of mechanical stresses within living tissues or any 3D multicellular environment from the analysis of oil droplets 3D deformations. The software is being developed in the Campàs Lab at the University of California, Santa Barbara. The software was designed by Ben Gross, Elijah Shelton and Otger Campàs and developed by Ben Gross and Elijah Shelton with the supervision of Otger Campàs. We thank the National Science Foundation (NSF CAREER award, CMMI-1562910) and the National Institute of General Medical Sciences (award number R01GM135380) and the National Institute of Dental and Craniofacial Research (award number R01DE027620) of the National Institutes of Health for financial support.

The STRESS source code is distributed under an Open Source license for non-commercial use (see License.md). The file lebedev_write_SPB.py contains source code owned by others and covered under other licences where indicated. Supplied c++ files are covered by the BSD license.

### Purpose

Surface Topography Reconstruction for Evaluation of Spatiotemporal Stresses (STRESS) is able to analyze tif file z-stacks collected from experimental 3D images of oil microdroplets, reconstruct the geometry of the droplet surface, and use this information to determine the stresses acting upon the droplet. Temporal sequences of droplets can be analyzed together to measure the temporal evolution and characteristics of stresses in the system.

### Installation

The STRESS package itself requires no installation, simply for the files to be downloaded and run together in the same folder. MATLAB (2018+) is required, including the Optimization Toolbox, Parallel Computing Toolbox, and Statistics and Machine Learning Toolbox, as is Python 3.7, including the following python packages: numpy, scipy, tvb-gdist, mpmath, vtk, matplotlib, seaborn. These can be installed using the command 'pip install [package name]'.

The current version of the code has been run on Linux, Windows, and Mac machines, so operating system shouldn't be an issue.

### Input File

The main script, AnalyzeDeluxe3D.m, requires a tif stack (single file) of the 3D fluorescence image of the droplet to be input. The tif file may include multiple time steps, in the case of a time series acquisition, but must only include one channel. The input tif can be located anywhere. If the image field of view is considerably larger than the droplet diameter, we recommend using a tool like FIJI to crop the stack to a height and width approximately equal to twice the droplet diameter. While cropping is not strictly necessary, it can be done easily as described below and will considerably speed up the analysis.

**Output Files**

The analysis code sends all output files in each run to a uniquely named folder in the \Outputs folder sub-directory of the folder where these files are installed. This name is print to the command window in MATLAB at the end of the run.

These outputs include 3 types of files:

1) *pdf plots*: These files are plots of various quantities of interest. They are typically used for quickly combing over the results for debugging purposes or sharing results.

2) *Output_For_MATLAB.mat:* This .mat file can be loaded into MATLAB to view various quantities computed for the analysis in a speadsheet format. These are detailed in the output.txt file.

3) *vtp plots*: For each time frame the droplet is analyzed, there are 5 .vtp files output. The timesteps are found at the end of the label: '...0001.vtp', '...0002.vtp', etc. These files can be visualized in 3D (and over the time frames) in ParaView software:

   **a)** 'Plot_Inputs_justCoordinatesAndCurvatures…vtp' shows the points and mean curvatures calculated from the MATLAB analysis.
   **b)** 'Least_Sq_Ellpsoid_justCoordinatesAndCurvatures…vtp' shows the least-squares ellipsoid fitted to the input segmented droplet points.
   **c)** 'SPH_fit_UV_justCoordinatesAndCurvatures…vtp' shows the fit of the droplet surface using the spherical harmonic basis resolution chosen, using the segmented points from the MATLAB code.
   **d)** 'SPH_fit_ELLPS_LBDV_justCoordinatesAndCurvatures...vtp' shows the least-squares ellipsoid at the lebedev quadrature points used.
   **e)** 'SPH_fit_LBDV_justCoordinatesAndCurvatures...vtp' shows the fit of the droplet surface using the spherical harmonic basis resolution chosen, at the lebedev quadrature points used.

**Instructions**

Cropping & Converting Input to a Tif File Stack:

1. Open the microscope image file (*.tif, *.lsm, etc) in FIJI
2. Draw a rectangle around the droplet, approximately two diameters in height and width. *Note: Adjust the z-slider to view a section near the equator before drawing the rectangle.*
3. Right-click and select 'Duplicate' option
4. In the 'Duplicate' pop-up window:
   a. Enter a name for the cropped image (i.e. 'Crop3D_myDrop')
   b. Check the box 'Duplicate hyperstack'
   c. Enter the channel number corresponding to the fluorescent droplet
   d. Make sure the full range of z slices is included
   e. If more than one timeframe is represented in this stack, specify the range you would like to analyze. *Note: Both single and multiple timeframe sequences are supported in the analysis.*
   f. Click 'OK'

5. Click File -> Save as… -> Tiff… to save cropped tif stack to a folder of your choosing. This is the tif file you will be prompted to locate when you run ‘AnalyzeDeluxe3D.m’ in MATLAB.

Running the Analysis:

1. Locate a properly formatted tif stack of an imaged droplet, or create one using the above instructions.
2. Open ‘AnalyzeDeluxe3D.m’ in MATLAB and hit ‘Run’ to start this script.
3. This code will first check whether or not python 3.7 and the required packages are installed. If not, the code will stop and an error message will appear.
4. In the ‘Select File to Open’ popup, locate the input tif stack file.
5. In the MATLAB Command Window prompt enter the number of timeframes of the droplet that are contained in the input tif stack file. This should be a natural number.
6. In the MATLAB Command Window prompt enter the voxel size in the xy-directions in the tif file, measured in microns. This can be found in the image metadata from ImageJ.
7. In the MATLAB Command Window prompt enter the voxel size in the z-direction in the tif file, measured in microns. This can be found in the image metadata from ImageJ.
8. In the ‘Label Options’ popup, indicate whether the droplet is labeled with a ‘Fluorescent Interior’ (fluorescence marker inside the droplet) or a ‘Florescent Surface’ (fluorescence marker only at the droplet surface).
9. In the MATLAB Command Window prompt enter the droplet’s interfacial tension in mN/m. There are multiple methods to measure the droplet interfacial tension (please see main text of the article describing this STRESS software for more information). If you do not know the interfacial tension value and are only interested in seeing the stress inhomogeneities on the droplet surface or want to obtain values of stress for relative measurements (ratios of stress measurements), please enter 1.
10. In the ‘Segmentation Options’ popup, select ‘Local Edge Fit’.
11. In the next ‘Segmentation Options’ popup, select ‘Parallelize’ to speed up the MATLAB segmentation analysis, or ‘Serialize’ if your computer is busy with other programs.
12. In the MATLAB Command Window prompt enter the desired filter size. For good imaging data this should be 1, and for data where the droplet is not imaged as well this could be as high as 5.
13. In the ‘Set FIDELITY’ popup, the degree of the spherical harmonic fit to the segmented surface points is chosen (this and following inputs are for the python analysis). A lower degree of basis runs faster, but may not capture all of the variation in surface deformations (specially, fine deformations). Also, the code may automatically lower the user-input basis degree chosen here, if the segmentation lacks the resolution needed for such a fit.
14. In the ‘Max LBDV Points’ popup, you can choose ‘Yes’ to use all 5810 possible lebedev quadrature points to integrate and represent the droplet surface, or you can choose ‘No’ to use the minimum number of points needed to integrate the spherical harmonics used in the basis chosen in the previous step. Choosing ‘Yes’ will significantly slow down the code, and most of the improvement in spatial resolution will be related to quantities that rely on geodesic distance calculations. We recommend using this setting only on the droplet time frames you wish to visualize in 3D (i.e. present) using ParaView, or if you want/need high resultion in your visualization of the data.
15. In the ‘ALPHA SETTING’ popup, you can choose alpha (see main text of paper describing the software), namely the fraction stress measurements to exclude as outliers. The default value of 0.05 excludes the largest and smallest 5% of values from such calculations. This can be set anywhere from 0 (no outliers excluded) up to 0.50 (all values excluded as outliers).

**16.** In the ‘Reference Coordinate’ popup, you can select ‘Yes’ to add a reference coordinate to compute orientations of principal stresses in a radial coordinate system (note that orientations and projections of principal stresses using the cartesian coordinate system given by the xyz droplet imaging frame are done automatically and included in the analysis). This input is used to define a radial coordinate system (this reference coordinate being its center), and compute the angle (in degrees) of the principal axes of the ellipsoidal mode of droplet deformation with respect to the radial direction, at each time point. Also, the distance (in microns) between the reference coordinate and droplet center is computed at each time frame. These outputs are described in output.txt; if ‘Yes’ is selected the following input prompts will appear in the MATLAB Command Window:

a) If the voxel sizes are the same in the image you are using to locate the reference coordinate, then enter ‘y’. If not, enter ‘n’, and you will be asked to enter the voxel sizes, as in steps 6-7.

b) You will be asked to enter the row, column, and page (in that order) of the reference coordinate in the image you are using to locate it. These can be measured from ImageJ, but note that the row is the y-value (from the top to bottom of the image) and the column is the x-value (from the left to the right of the image).

c) Note: this will assume the same reference coordinates for each time frame in the tif stack.

**17.** In the MATLAB Command Window you will be asked to specify a unique ID string for this run of the code. This will not affect how the code runs, but is used to label the meta-data.

**18.** In the MATLAB Command Window you will be asked to specify a time stamp string for each droplet time frame. This will not affect how the code runs, but is used to label the meta-data.

**19.** The code should now run! Pay attention to any error messages. The code should output the directory of the outputs at the end.

## Measurement Descriptions

The measured quantities are described in the paper describing STRESS, as well as the output.txt file included in the installation.