TFMLAB: a MATLAB toolbox for 4D traction force microscopy

Jorge Barrasa-Fano[1]: *jorge.barrasafano@kuleuven.be*
Apeksha Shapeti[1]: *apeksha.shapeti@kuleuven.be*
Álvaro Jorge-Peñas[1]: *alvarojorgep@gmail.com*
Mojtaba Barzegari[1]: *mojtaba.barzegari@kuleuven.be*
José Antonio Sanz-Herrera[2]: *jsanz@us.es*
Hans Van Oosterwyck[1,3*]: *hans.vanoosterwyck@kuleuven.be*

[*]Corresponding author

[1]Department of Mechanical Engineering, Biomechanics section, KU Leuven, Belgium

[2]Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, Spain

[3]Prometheus, Division of Skeletal Tissue Engineering, KU Leuven, Leuven, Belgium

Abstract.
We present TFMLAB, a MATLAB software package for 4D (x;y;z;t) Traction Force Microscopy (TFM). While various TFM computational workflows are available in the literature, open-source programs that are easy to use by researchers with limited technical experience and that can analyze 4D in vitro systems do not exist. TFMLAB integrates all the computational steps to compute active cellular forces from confocal microscopy images, including image processing, cell segmentation, image alignment, matrix displacement measurement and force recovery. Moreover, TFMLAB eases usability by means of interactive graphical user interfaces. This work describes the package's functionalities and analyses its performance on a real TFM case.

Keywords:
Traction force microscopy; cell mechanics; Image processing; Finite element method

1. Motivation and Significance

Mechanobiology, the study of interactions between mechanical signals and biological responses, has substantially grown over the last 20 years. The importance of mechanical forces in driving cell behavior is now widely recognized in the literature [1]–[4]. Meanwhile, Traction Force Microscopy (TFM) has become the preferred methodology to quantify forces at the cell-matrix interface. Typically, synthetic or natural hydrogels mixed with cells and fiducial markers are imaged by means of optical microscopy before (stressed state) and after (relaxed state) cell relaxation. Image processing algorithms measure cellular force-induced deformations on the extracellular matrix (ECM) by tracking the movement of fiducial markers between these two states. Finally, cell forces are retrieved by applying mechanical models that relate forces to deformations. The reader is referred to excellent reviews in the field for more details [3], [5], [6].

TFM has traditionally been applied to study cell mechanics in 2D in vitro cultures where cells are seeded on top of a substrate [7]–[9]. Several open-source MATLAB software packages used in 2D (i.e. 2D displacement and force calculation for 2D cell culture) and 2.5D TFM (i.e. 3D displacement and force calculation for 2D cell culture) are available: displacement measurement algorithms such as Particle Image Velocimetry (PIV) [10] and Particle Tracking (PT) [11], or regularization techniques for force recovery [12]. Moreover, "all-in-one" packages such as an ImageJ plugins [13] or the contributions by the groups of Franck [14], Dufrense [15], Sabass [16] or Danuser [17] have brought complex TFM computations closer to experimentalists. Current challenges lie in making TFM available for physiologically more relevant experiments in 3D (cells fully embedded in a hydrogel) and incorporating time dependency (4D). Unlike for 2D TFM, the availability of 3D TFM open-source tools is scarce. Some groups that are currently publishing 3D TFM studies do not provide source codes [18]–[21]. Others, provide source codes but lack force calculation functionalities [22]. An exception are the open source contributions of Fabry's group. They developed *SAENO* [23], the first open source 3D TFM solver for nonlinear, fibrillary materials, like collagen and fibrin. While it remains a unique contribution to the field, we hereby address two limitations regarding its usability by researchers with limited technical experience. First, the software does not provide feedback of intermediate steps involved in the calculations (such as image filtering or shift correction), which requires higher technical expertise from users to select the optimal input parameters for each sample [24]. Second, while the signal to noise ratio of the images heavily depend, among others, on the type of microscope or the hydrogel used, *SAENO* does not allow to adjust image processing operations to new data as they are hidden from the user. More recently, they published *jointforces,* an open source Python package [25]. While they applied it for data from a 3D in vitro model of a tumor spheroid, the software only calculates 2D forces at the equatorial plane of the spheroid.

With TFMLAB, we provide an "all-in-one" open source 4D (3D + time) toolbox that includes all the necessary computational steps for TFM. Experimentalists can input the raw microscopy images and follow every computational step through interactive Graphical User Interfaces (GUIs) for parameter tuning. Users can choose different image filters to process various types of data and use the Free Form Deformation (FFD) algorithm for displacement measurement. In particular, FFD has proven to be superior to PIV and has been used in various experimental studies involving 2D, 3D and 4D (time lapse) measurements in single cell and multicellular systems [26]–[30]. Finally, accurate and computationally efficient force calculation is provided by our recently published physically-based nonlinear inverse method (PBNIM) [31], [32].

TFMLAB combines complex state of the art methods into a closed and user-friendly software that does not require code manipulations and allows for applying 4D TFM in biological studies holding strong promise for mechanobiology.

2. Software Description

2.1. Software Architecture

We developed TFMLAB in the MATLAB framework to provide robust and user-friendly processing of the microscopy data. The software has one main function main.m, which calls other routines at every step of computational TFM (see schematic in Fig 1). However, TFMLAB follows a "black box" fashion and the user only needs to interact with TFMLAB's GUIs. Briefly, TFMLAB has the following steps: (1) microscopy image reading, (2) image processing, (3) displacement measurement and (4) force calculation. There is a GUI for parameter tuning before every step and result folders are stored between steps. While MATLAB orchestrates the workflow, some specific operations are performed by external software (see Table 1).

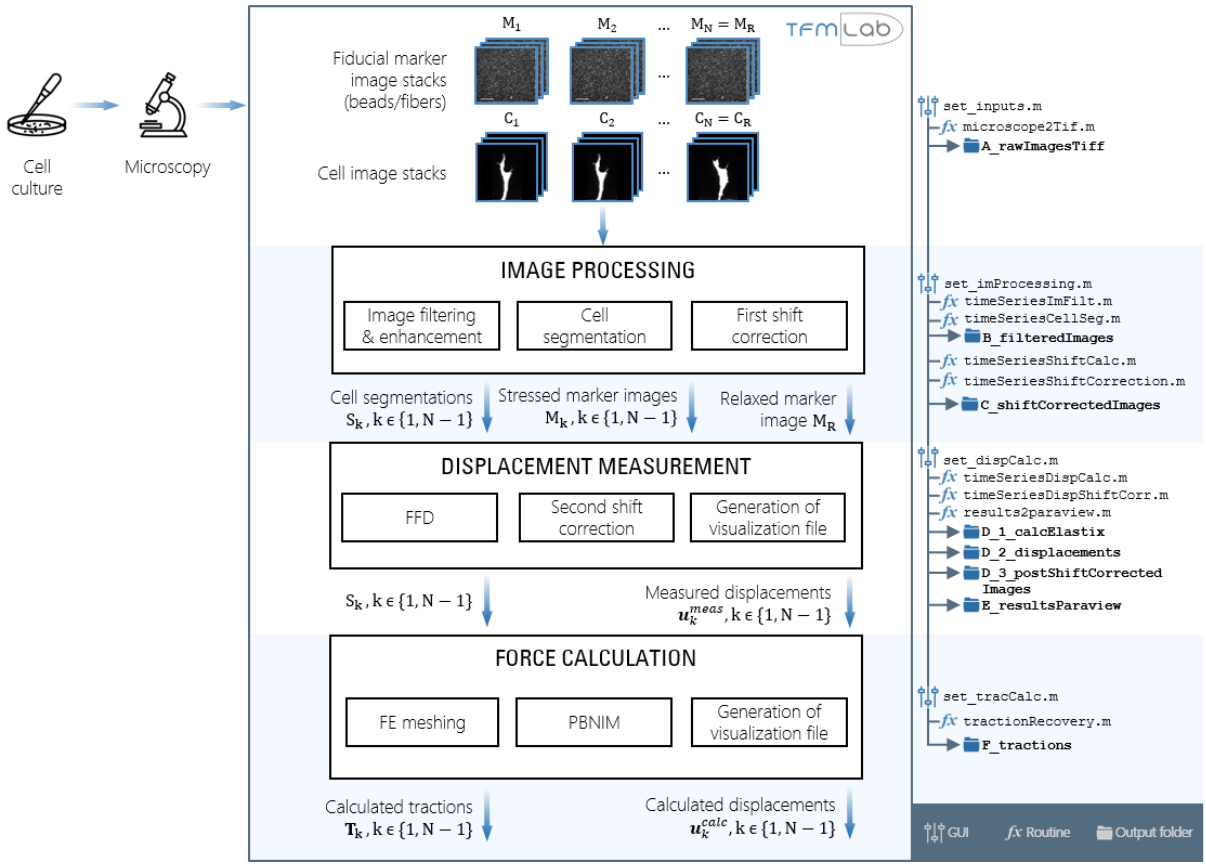

Fig 1. Basic flow chart of TFMLAB. After cell culture and microscopy imaging (left hand side), the user can input marker (M_k) and cell (C_k) image stacks into TFMLAB (top). The main steps of the workflow (middle), the GUI scripts, the name of the key MATLAB functions that perform the steps and the output folders that the software generates (right hand side) are shown. Briefly, after image processing, displacements ($\boldsymbol{u}_k^{meas}$) in the marker images are measured by means of Free Form Deformation (FFD). These displacements and a finite element (FE) mesh of the cell and the hydrogel are used to calculate improved displacements $\boldsymbol{u}_k^{calc}$ and their associated tractions $\boldsymbol{T}_k$, which fulfil the equilibrium of forces in the hydrogel with the Physically-Based Nonlinear Inverse Method (PBNIM). Icons used freely available from [50], [51].

Table 1. External software used in TFMLAB. Please note that Abaqus and FreeFEM are two FE solver alternatives. Only one of them is necessary for force calculation.

Software/toolbox	Freely available	Provided with TFMLAB	Use
Bioformats [33]	Yes	Yes	Microscopy image reading
DIPImage [34]	Yes	No	Image processing: image filtering and enhancement
Elastix [35]	Yes	Yes	Displacement measurement: FFD image registration
Paraview [36]	Yes	No	(Optional) results visualization
iso2mesh [37]	Yes	Yes	Force calculation: Automatic FE meshing
SuiteSparse [38]	Yes	Yes	Force calculation: PBNIM system of equations
Abaqus	No	No	Force calculation: FE solver
FreeFEM [39]	Yes	No	Force calculation: Alternative FE solver

2.2. Software Functionalities

The user must first run the script main.m in MATLAB and the steps depicted in Fig 1 follow, asking for user interaction through different GUIs.

Microscopy image reading

First, the user must provide input data. The GUI *set_inputs* will ask for an input microscopy file. For flexibility, TFMLAB uses the Bio-Formats toolbox which can read multiple microscopy file formats [33]. From the input file, the software reads the channels corresponding to the cells (e.g., imaged after transduction with LifeAct for filamentous actin staining) and the fiduciary markers such as fluorescent beads and/or fibers (e.g., in case of a fibrillar collagen hydrogel imaged by means of second harmonic generation [27]). Any other channels (e.g. bright field, nuclei…) present in the microscopy file are ignored, as they are not necessary for TFM. The user can provide any number of images (time points) for a given sample as long as the field of view of the hydrogel remains constant. All the images are considered to conform different time points of the stressed state while the last image provided is considered to be the relaxed state. Function *microscope2Tif* then converts the input data into .tiff files. This is a standard format that maintains bit depth and is easy to access with any free software such as ImageJ. The data is split by channels (cell, beads and/or fibers) and time points (tp_01, tp_02,…, tp_R) and stored in the A_rawImagesTiff output folder.

Image processing

The GUI *set_imProcessing* allows the user to interactively tune image filtering and enhancement parameters. TFMLAB provides specific filters for cell, bead and fiber images, and tools for cell segmentation. There are multiple filtering options combining the MATLAB image processing toolbox and the DIPImage library [34] for versatility (see examples in section 3). The main function then calls the functions *timeSeriesImFilt* and *timeSeriesCellSeg* to process all the input images and stores them in the B_filteredImages output folder. Rigid image registration is done during the first shift correction step, correcting for any microscope stage related drifts. First, a phase correlation operation is done to calculate the shift between all the images and a reference image with the function *timeSeriesShiftCalc*. If only two

images are provided, the reference is the relaxed image. Otherwise, the reference is the middle time-point. The bead/fiber images are used for this operation. Second, the function *timeSeriesShiftCorrection* aligns the images from all the channels according to the calculated shift, and any non-overlapping image borders are cropped such that all the images have equal size at the end of the operation. The resultant images are stored in the C_shiftCorrectedImages output folder.

Displacement measurement

Displacement measurement is done by means of FFD-based image registration [26]. Briefly, a multivariate B-spline function warps a moving image (a marker image at a certain time point of the stressed state, $\mathrm{M_k}$) to optimize the value of a distance metric that quantifies its similarity to a fixed image (the marker image that represents the relaxed state, $\mathrm{M_R}$). Using the GUI *set_dispCalc* the user can choose between different optimizers, distance metrics and can tune the spacing between the control points of the B-spline curves to adapt it to experimental conditions of each sample. The main function then calls *timeSeriesDispCalc*, which uses the Elastix toolbox for efficient image registration as previously described [29], [30], [40]. The cell segmentations $\mathrm{S_k}$ can be used as masks to ignore beads engulfed by the cells. The parameter and performance files of the image registration are stored in output folder D_1_calcElastix. With a second shift correction, the measured displacement fields $\mathbf{u}_\mathrm{k}^\mathrm{meas}$ are then corrected for any spurious rigid shifts in *timeSeriesDispShiftCorr* and stored in folder D_2_displacements. This shift correction operation is done by subtracting the mean from each displacement field component for the entire image stack. The images (beads/fibers, cell and cell segmentations) are also corrected accordingly and stored in D_3_postShiftCorrectedImages. Finally, function *results2paraview* generates .vtk files for vector field and cell segmentation visualization in the free software Paraview [36] and stores them in folder E_resultsParaview.

Force calculation

The user is asked to tune the force calculation parameters in GUI *set_tracCalc*. The cell segmentations $\mathrm{S_k}$ are used to create a finite element (FE) mesh of the cell surface and the hydrogel domain with the open source mesh generator iso2mesh [37]. The resultant 3D tetrahedral mesh is coarser away from the cell surface. The GUI allows for tuning the element sizes and smoothening. Moreover, the user can define the mechanical parameters of the hydrogel such as the elastic modulus or the Poisson's ratio (assuming linear elastic, isotropic behavior). Since the displacement field $\mathbf{u}_\mathrm{k}^\mathrm{meas}$ is measured only at certain discrete locations (beads/fibers), it is not error free and traction recovery is typically done by applying regularization techniques to avoid overfitting. In TFMLAB, we used our novel PBNIM method which proved to be more accurate than a non-regularized method on in silico and in vitro experiments [31], [32]. Briefly, this method calculates a displacement field, $\mathbf{u}_\mathrm{k}^\mathrm{calc}$, that is minimally different from the measured one, $\mathbf{u}_\mathrm{k}^\mathrm{meas}$, and that is enforced to fulfil equilibrium of forces in the hydrogel domain as a constraint (see details in Appendix B). PBNIM was implemented in the function *tractionRecovery*. Since PBNIM is developed using a FEM framework, a FE solver is required. TFMLAB is compatible with two FE solvers, namely, the commercial Abaqus (Dassault Systemes Simulia Corporation, Providence, RI) and the open source software FreeFEM [39]. Users can also pick between one of these FE solvers to compute tractions. Stiffness matrices are computed through the selected FE solver. If Abaqus is selected, TFMLAB provides it with an

automatically written job file and the results are imported back into the MATLAB workspace in order to preserve the "black box" architecture of TFMLAB. Similarly, TFMLAB automatically interacts with FreeFEM, for which we developed specific FreeFEM routines. The minimization problem (Eq. B.3) is then analytically derived turning into a system of equations solved by function *spqr_solve*, which is part of the open source toolbox SuiteSparse [38]. This gives a displacement field $\mathbf{u}_k^{calc}$, which, unlike $\mathbf{u}_k^{meas}$, fulfils the equilibrium of forces in the hydrogel domain. Following the Finite Element Method (FEM) procedure and assuming a linear elastic behavior, stresses and tractions $\mathbf{T}_k$ are forwardly obtained (Eq. B.4 - Eq. B.6). Finally, .vtk files are generated for result visualization. All the results from this step are stored in the output folder F_tractions.

Output and other software features

After computing the full workflow, the user might be interested in repeating a specific step with different parameters while preserving the previous steps. GUIs allow for skipping steps starting from a specific step of the workflow as opposed to running the entire workflow again. All the parameters selected by the user are stored in .mat files for reproducibility. All the important output data (processed images, displacement fields, force fields, etc) are stored in .mat and text files to allow for further analysis by the user. Together with the .vtk visualization files, TFMLAB also stores a .psvm template for Paraview for automatic 3D vector field rendering.

2.3. Sample code snippets analysis (optional).

3. Illustrative Examples

In this section, we present an example of the performance of TFMLAB using experimentally acquired data from an in vitro model of angiogenesis in which Human Umbilical Vein Endothelial Cells (HUVECs) invade a polyethylene glycol (PEG) hydrogel containing suspended 200nm fluorescent beads. More details on the experimental protocol and microscopy imaging can be found in Appendix A.

For this example, we acquired cell and bead image stacks (290 x 290 x 45 µm) by means of confocal microscopy before and after cell relaxation with Cytochalasin D (see Fig 2a-d and Supplementary Videos 1 and 2), with a voxel size of 0.57x0.57x0.5 μm^3. An overview of the analysis of this data using TFMLAB is provided in Supplementary Video 3. First, the bead images were processed by applying a Difference of Gaussians (DoG) filter to enhance the spherical shape of the beads and noise removal, followed by a contrast stretching operation (see Fig 2e-h). Next, the cell images were processed by applying a penalized least squares-based denoising [41] and the cell body was segmented by applying Otsu thresholding and by removing small binary objects (see Fig 2i-k). Finally, shifts caused by the microscope stage were removed by applying phase-correlation-based rigid image registration (see Fig 2l-o). Before shift correction, a distinct shift that affects all the beads of the image is present (Fig 2l, n). After shift correction, the only bead movements remaining happen around the sprout, indicating cell-matrix interactions (Fig 2m, o).

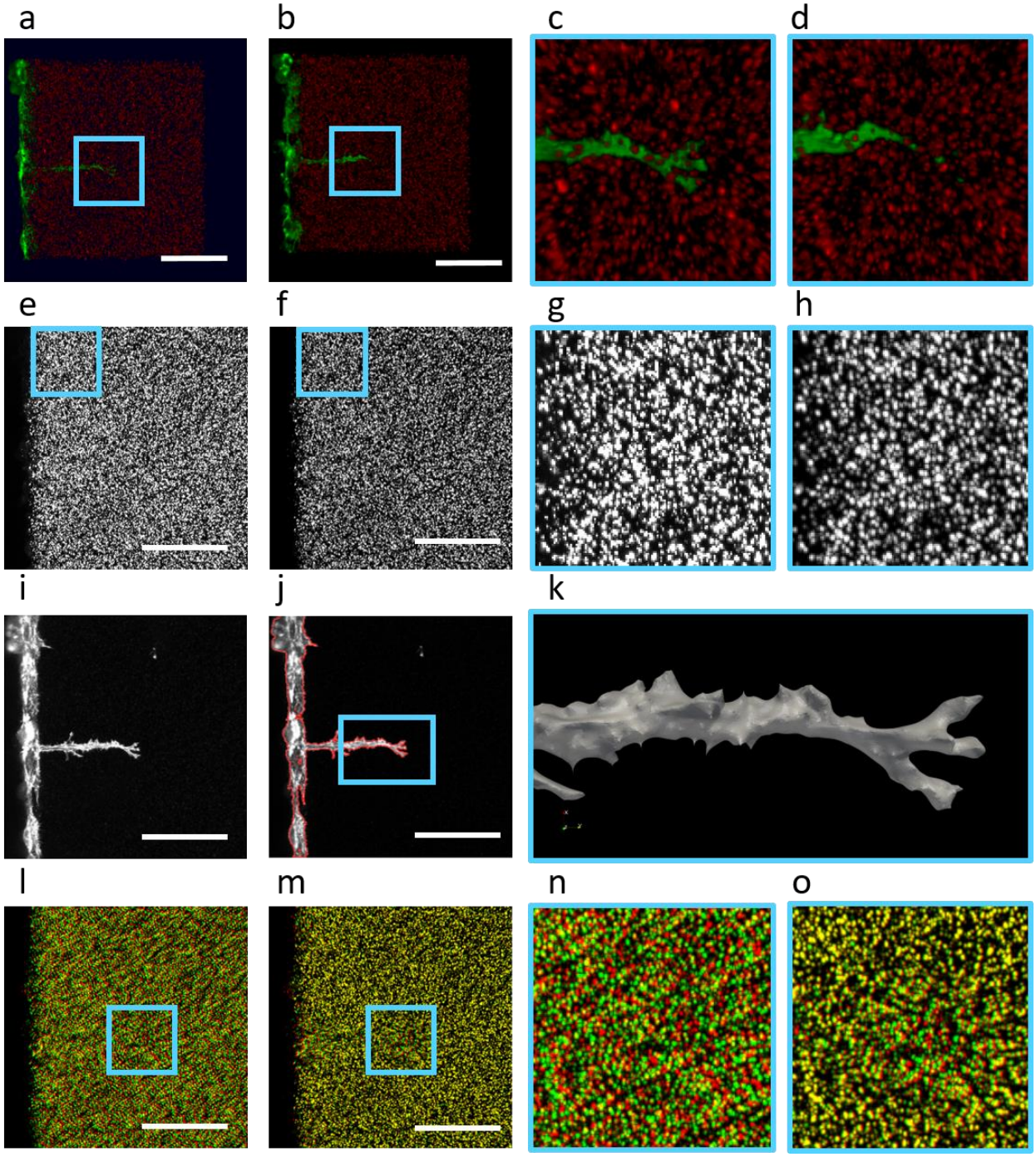

Fig 2. Image processing steps in TFMLAB. (a,b) 3D renders of the stressed and relaxed images as obtained by means of confocal microscopy (cell channel in green, beads channel in red). (c,d) Zoomed in regions from the raw stressed and relaxed images. (e,f) Maximum projections of a bead image before and after image filtering and enhancement. (g,h) Zoomed in regions before and after image filtering and enhancement. (i) Maximum intensity projection of the raw cell image. (j) Maximum intensity projection of the filtered cell image and segmentation boundary (red). (k) 3D render (Paraview) of the resultant cell segmentation. (l,m) Maximum intensity projection overlay of the stressed (green) and relaxed (red) bead images before and after shift correction. (n,o) Zoomed in regions before and after shift correction. Scale bars: 100µm.

185 After image processing, non-rigid displacements are measured by FFD and the PBNIM method is applied
186 to recover cellular forces. Fig 3a shows TFMLAB's force recovery GUI *set_tracCalc*. The cell surface can be

further smoothed prior to FE meshing to ease the meshing process. The user can tune and visualize the resultant FE mesh before computing forces. For this example, the resultant FE mesh had ~90000 4-node 3-D linear tetrahedral elements. The elastic modulus was set to 200Pa based on experimental results and the Poisson's ratio was assumed to be 0.2. Table 2 shows the computational times to complete every step in TFMLAB on an Intel Core i7-7700 CPU 3.60GHz with a Windows operating system and 16 GB of RAM. These numbers exclude the time invested in parameter tuning, which for an experienced user can be around 10 minutes with the majority of this time spent on image filtering, cell segmentation and FE meshing.

Results show a clear pulling pattern near the sprout tip inducing up to ~6µm matrix displacements (see Fig 3b). The recovered tractions present maxima at the tips of the cell's protrusion of around 150Pa (see Fig 3c). While this displacement field pattern around angiogenic sprouts has been reported in previous studies [30], [42]–[44], TFMLAB additionally incorporates quantification of traction magnitude and direction. This can be a crucial tool for the quantification of cell mechanical behavior in 3D, ECM-mimicking

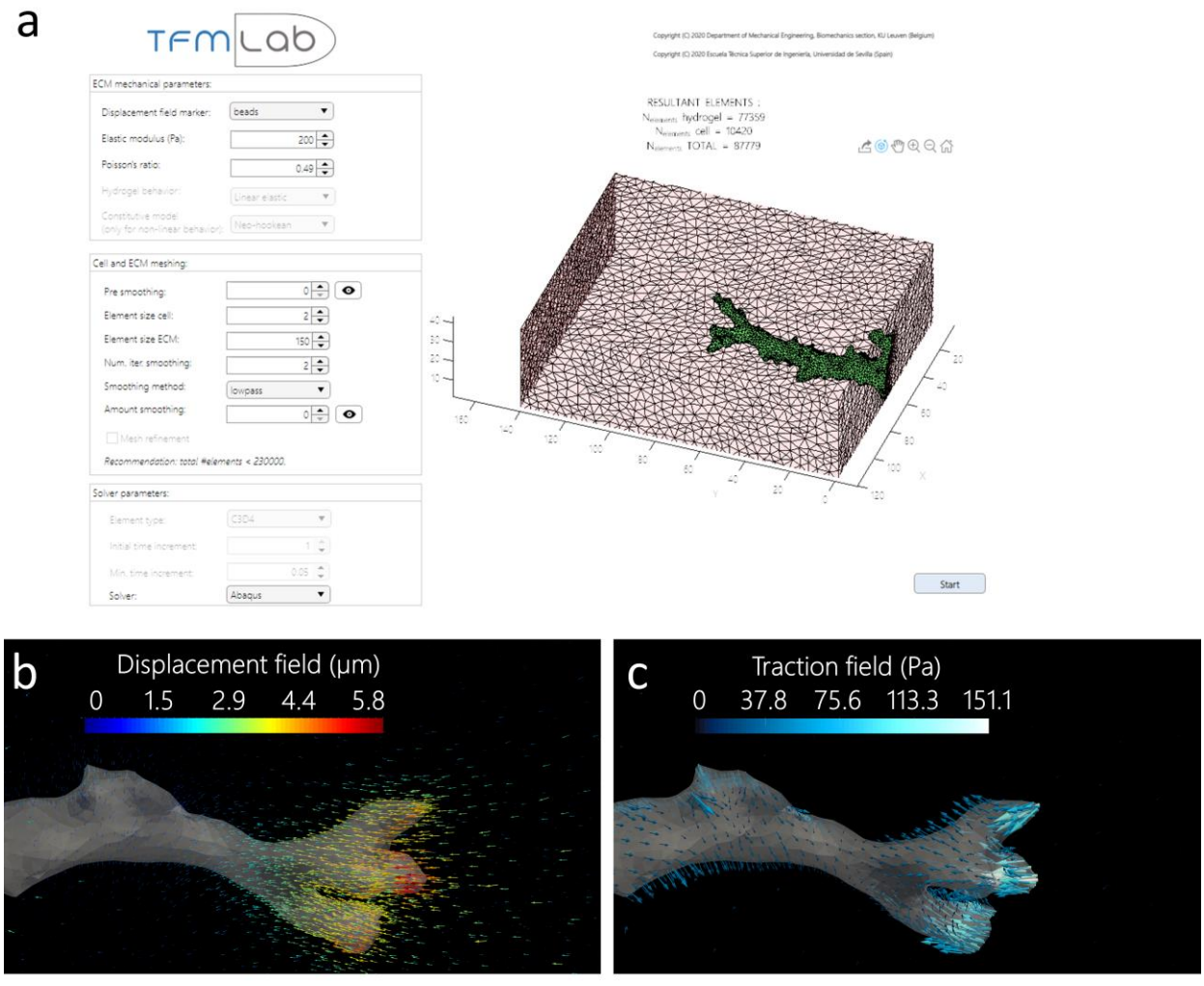

Fig 3. (a) GUI for FE meshing and force calculation. (b) 3D render (Paraview) of the 3D displacement field (arrows) around the sprout tip as calculated by PBNIM. (c) 3D render (Paraview) of the traction field (arrows) around the tip of the sprout as calculated by PBNIM.

hydrogels, both for single cells as well as multicellular structures and for applications such as disease modeling, regenerative medicine and tissue engineering as well as developmental biology. Further advantages of TFMLAB are discussed in section 4.

Table 2. Computational times for every step of TFMLAB for the given example (one time point of a TFM experiment).

TFMLAB step	Computational time (seconds)
Image filtering	25.71
Shift calculation	2.33
Pre-shift correction	19.72
Cell segmentation	9.86
Displacement measurement	63.43
Post-shift correction	31.71
Force calculation	56.15
TOTAL (minutes)	**3.48**

4. Impact

With this work, we provide an easy-to-use and versatile open source code to perform 4D TFM, making advanced cell mechanical data accessible to researchers with various (including limited) technical backgrounds. To our knowledge, the following features make TFMLAB an unprecedented tool in the field of mechanobiology:

- It is an "all in one" software. Currently, a typical researcher running computational 3D TFM uses e.g. ImageJ for image processing and extra software for displacement and force calculation. This leads to time-consuming work adapting input data to ensure compatibility between software. In contrast, researchers have all the tools they need in TFMLAB.
- It includes state of the art methods of proven usefulness in multiple TFM studies [26]–[32].
- While TFMLAB is a "black box" software, its GUIs especially offer ease of usability for researchers with limited programming experience and enable viewing progress of the various steps and parameter tuning.
- TFMLAB does not require having any commercial software other than MATLAB. As other state of the art TFM software [19], [21], [45], TFMLAB is compatible with Abaqus, a sophisticated and technical FE solver. However, it also offers an open source alternative, namely FreeFEM. Moreover, the user needs little knowledge on FE modelling as only basic parameter tuning (via a TFMLAB GUI) is required.
- It is versatile as it can handle different 3D TFM approaches. Current research includes both bead-based TFM [20], [46], [47] and fiber-based TFM (with fiber networks e.g. being imaged by means of label-free techniques such as second harmonic generation or confocal reflection microscopy) [23], [27], [48]. Our software includes specific image filters for both markers and the FFD algorithm handles both of them [27], [30]. Furthermore, a user can run a classical TFM problem (using a stressed image and a relaxed image, as in the example in section 3) or time-lapse TFM (using a sequence of stressed images and one relaxed image). If the researcher's experimental protocol

does not include cell imaging, displacement calculation based on the fiducial markers is still possible.

The compactness, user-friendliness and versatility of our toolbox makes it usable in studying many physiological or pathological processes where cell-matrix mechanics play an important role. Such a tool can become key in unravelling cell mechanical time-dependant interactions with their 3D microenvironment, aspects at the forefront of recent mechanobiological research [25], [30], [48], [49].

5. Conclusions

TFMLAB is an open-source MATLAB toolbox that seeks to bring 4D TFM closer to medical and biological researchers that do not necessarily have a strong technical background, thereby facilitating the transition of TFM technology from (biomedical) engineering – and/or (bio-)physics-dominated research environments towards medicine and biology. All the necessary steps for computing 4D cellular forces are included in one easy-to-use package (microscopy image reading and processing, displacement measurement, force recovery and visualization file generation). We demonstrated the usefulness and efficiency of the software in an example using real experimental data. We expect that researchers will find TFMLAB useful independent of their programming or technical background.

Conflict of Interest

No conflict of interest exists:
We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

The authors thank Dr Marie-Mo Vaeyens, Dr Mar Cóndor and Janne de Jong for their valuable code testing and feedback. The authors also thank the Ranga lab for providing the PEG precursors used in hydrogel preparation. J.B.F. was supported by the Research Foundation Flanders (FWO) (travel grant for a long stay abroad, FWO grant V413019N). J.B.F. and H.V.O. were supported by KU Leuven internal funding C14/17/111. A.S. was supported by the FWO SB grant 1S68818N. M. B. was supported by the FWO project G085018N and the Prosperos project, funded by the Interreg VA Flanders – The Netherlands program, CCI grant no. 2014TC16RFCB046. J.A.S.H. was supported by the José Castillejo fellowship of the Ministerio de Educación, Cultura y Deporte of Spain, grant number CAS17/00096 and Spanish Ministry of Economy and Competitiveness (MINECO) through the project PGC2018-097257-B-C31. H.V.O. was supported by the European Research Council under the European Union's Seventh Framework Program (FP7/2007–2013)/ERC Grant Agreement No. 308223), by an FWO grant (G087018N) and by an FWO / Hercules infrastructure grant (G0H6316N). The financial support is gratefully acknowledged.

References

[1] C. M. Kraning-Rush, J. P. Califano, and C. A. Reinhart-King, "Cellular traction stresses increase with increasing metastatic potential," *PLoS One*, vol. 7, no. 2, Feb. 2012.

[2] K. A. Jansen, R. G. Bacabac, I. K. Piechocka, and G. H. Koenderink, "Cells actively stiffen fibrin networks by generating contractile stress," *Biophys. J.*, vol. 105, no. 10, pp. 2240–2251, Nov.

2013.

[3] U. S. Schwarz and J. R. D. Soiné, "Traction force microscopy on soft elastic substrates: A guide to recent computational advances," *Biochim. Biophys. Acta - Mol. Cell Res.*, vol. 1853, no. 11, pp. 3095–3104, 2015.

[4] M. L. Kutys and C. S. Chen, "Forces and mechanotransduction in 3D vascular biology," *Current Opinion in Cell Biology*, vol. 42. Elsevier Ltd, pp. 73–79, 01-Oct-2016.

[5] J. A. Mulligan, F. Bordeleau, C. A. Reinhart-King, and S. G. Adie, "Measurement of dynamic cell-induced 3D displacement fields in vitro for traction force optical coherence microscopy," *Biomed. Opt. Express*, vol. 8, no. 2, p. 1152, Feb. 2017.

[6] W. J. Polacheck and C. S. Chen, "Measuring cell-generated forces: A guide to the available tools," *Nature Methods*, vol. 13, no. 5. Nature Publishing Group, pp. 415–423, 01-May-2016.

[7] M. Dembo and Y.-L. Wang, "Stresses at the Cell-to-Substrate Interface during Locomotion of Fibroblasts," *Biophys. J.*, vol. 76, no. 4, pp. 2307–2316, 1999.

[8] N. Q. Balaban *et al.*, "Force and focal adhesion assembly: A close relationship studied using elastic micropatterned substrates," *Nat. Cell Biol.*, vol. 3, no. 5, pp. 466–472, Apr. 2001.

[9] A. J. Engler, S. Sen, H. L. Sweeney, and D. E. Discher, "Matrix Elasticity Directs Stem Cell Lineage Specification," *Cell*, vol. 126, no. 4, pp. 677–689, Aug. 2006.

[10] Nobuhito Mori, "FrontPage - mpiv - MATLAB PIV Toolbox." [Online]. Available: http://www.oceanwave.jp/softwares/mpiv/. [Accessed: 07-Apr-2020].

[11] D. Blair and E. Dufresne, "Matlab Particle Tracking." [Online]. Available: http://site.physics.georgetown.edu/matlab/. [Accessed: 07-Apr-2020].

[12] P. C. Hansen, "regtools - File Exchange - MATLAB Central." [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/52-regtools. [Accessed: 07-Apr-2020].

[13] Q. Tseng, "Traction Force Microscopy - ImageJ plugins." [Online]. Available: https://sites.google.com/site/qingzongtseng/tfm. [Accessed: 07-Apr-2020].

[14] The Franck Lab, "DOWNLOADS | francklaboratory." [Online]. Available: https://www.franck.engin.brown.edu/downloads. [Accessed: 07-Apr-2020].

[15] R. W. Style *et al.*, "Traction force microscopy in physics and biology," *Soft Matter*, vol. 10, no. 23. Royal Society of Chemistry, pp. 4047–4055, 21-Jun-2014.

[16] Y. Huang, G. Gompper, and B. Sabass, "A Bayesian traction force microscopy method with automated denoising in a user-friendly software package," *Comput. Phys. Commun.*, p. 107313, Apr. 2020.

[17] S. J. Han, Y. Oak, A. Groisman, and G. Danuser, "Traction microscopy to identify force modulation in subresolution adhesions," *Nat. Methods*, vol. 12, no. 7, pp. 653–656, Jul. 2015.

[18] D. Song, N. Hugenberg, and A. A. Oberai, "Three-dimensional traction microscopy with a fiber-based constitutive model," *Comput. Methods Appl. Mech. Eng.*, vol. 357, p. 112579, Dec. 2019.

[19] W. R. Legant, J. S. Miller, B. L. Blakely, D. M. Cohen, G. M. Genin, and C. S. Chen, "Measurement of mechanical tractions exerted by cells in three-dimensional matrices," *Nat. Methods*, vol. 7, no. 12, pp. 969–971, Dec. 2010.

[20] N. Gjorevski, A. S. Piotrowski, V. D. Varner, and C. M. Nelson, "Dynamic tensile forces drive collective cell migration through three-dimensional extracellular matrices," *Sci. Rep.*, vol. 5, p. 11458, Jul. 2015.

[21] M. Cóndor and J. M. García-Aznar, "An iterative finite element-based method for solving inverse problems in traction force microscopy," *Comput. Methods Programs Biomed.*, vol. 182, p. 105056, Dec. 2019.

[22] E. Lejeune, A. Khang, J. Sansom, and M. S. Sacks, "FM-Track: A fiducial marker tracking software for studying cell mechanics in a three-dimensional environment," *SoftwareX*, vol. 11, 2020.

[23] J. Steinwachs *et al.*, "Three-dimensional force microscopy of cells in biopolymer networks," *Nat. Methods*, vol. 13, no. 2, pp. 171–176, 2016.
[24] M. Cóndor, J. Steinwachs, C. Mark, J. M. García-Aznar, and B. Fabry, "Traction Force Microscopy in 3-Dimensional Extracellular Matrix Networks," in *Current Protocols in Cell Biology*, vol. 75, no. 1, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2017, pp. 10.22.1-10.22.20.
[25] C. Mark *et al.*, "Collective forces of tumor spheroids in three-dimensional biopolymer networks," *Elife*, vol. 9, Apr. 2020.
[26] A. Jorge-Peñas *et al.*, "Free form deformation-based image registration improves accuracy of traction force microscopy," *PLoS One*, vol. 10, no. 12, pp. 1–22, 2015.
[27] A. Jorge-Peñas *et al.*, "3D full-field quantification of cell-induced large deformations in fibrillar biomaterials by combining non-rigid image registration with label-free second harmonic generation," *Biomaterials*, vol. 136, pp. 86–97, Aug. 2017.
[28] A. Izquierdo-Álvarez, D. A. Vargas, Á. Jorge-Peñas, R. Subramani, M.-M. Vaeyens, and H. Van Oosterwyck, "Spatiotemporal Analyses of Cellular Tractions Describe Subcellular Effect of Substrate Stiffness and Coating," *Ann. Biomed. Eng.*, vol. 47, no. 2, pp. 624–637, Feb. 2019.
[29] C. Steuwe *et al.*, "Fast quantitative time lapse displacement imaging of endothelial cell invasion," *PLoS One*, vol. 15, no. 1, p. e0227286, Jan. 2020.
[30] M.-M. Vaeyens *et al.*, "Matrix deformations around angiogenic sprouts correlate to sprout dynamics and suggest pulling activity," *Angiogenesis*, Jan. 2020.
[31] J. A. Sanz-Herrera, J. Barrasa Fano, M. Cóndor, and H. Van Oosterwyck, "Inverse method based on 3D nonlinear physically constrained minimisation in the framework of traction force microscopy.," *Soft Matter*, 2020.
[32] J. Barrasa-Fano, A. Shapeti, J. De Jong, A. Ranga, J. A. Sanz-Herrera, and H. Van Oosterwyck, "Advanced in silico validation framework for three-dimensional Traction Force Microscopy and application to an in vitro model of sprouting angiogenesis," *bioRxiv*, p. 2020.12.08.411603, Dec. 2020.
[33] M. Linkert *et al.*, "Metadata matters: Access to image data in the real world," *Journal of Cell Biology*, vol. 189, no. 5. The Rockefeller University Press, pp. 777–782, 31-May-2010.
[34] "DIPimage - DIPimage & DIPlib." [Online]. Available: http://www.diplib.org/dipimage. [Accessed: 04-May-2020].
[35] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. W. Pluim, "Elastix: A toolbox for intensity-based medical image registration," *IEEE Trans. Med. Imaging*, vol. 29, no. 1, pp. 196–205, Jan. 2010.
[36] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*. Kitware, 2015.
[37] Qianqian Fang and D. A. Boas, "Tetrahedral mesh generation from volumetric binary and grayscale images," in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009, pp. 1142–1145.
[38] T. A. Davis, "Algorithm 915, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization," *ACM Trans. Math. Softw.*, vol. 38, no. 1, pp. 1–22, Nov. 2011.
[39] F. Hecht, "New development in freefem+," *J. Numer. Math.*, vol. 20, no. 3–4, pp. 251–265, Dec. 2012.
[40] A. Jorge-Peñas, A. Muñoz-Barrutia, E. M. De-Juan-Pardo, and C. Ortiz-De-Solorzano, "Validation tool for traction force microscopy," *Comput. Methods Biomech. Biomed. Engin.*, vol. 18, pp. 1377–1385, 2015.
[41] D. Garcia, "Robust smoothing of gridded data in one and higher dimensions with missing values," *Comput. Stat. Data Anal.*, vol. 54, no. 4, pp. 1167–1178, Apr. 2010.
[42] Y. Du, S. C. B. Herath, Q. Wang, D. Wang, H. H. Asada, and P. C. Y. Chen, "Three-Dimensional Characterization of Mechanical Interactions between Endothelial Cells and Extracellular Matrix during Angiogenic Sprouting," *Sci. Rep.*, vol. 6, no. 1, p. 21362, Aug. 2016.
[43] C. Yoon *et al.*, "Myosin IIA–mediated forces regulate multicellular integrity during vascular

sprouting," *Mol. Biol. Cell*, vol. 30, no. 16, pp. 1974–1984, Jul. 2019.

[44] Y. Du, S. C. B. Herath, Q. G. Wang, H. Asada, and P. C. Y. Chen, "Determination of Green's function for three-dimensional traction force reconstruction based on geometry and boundary conditions of cell culture matrices," *Acta Biomater.*, vol. 67, pp. 215–228, Feb. 2018.

[45] J. Toyjanova *et al.*, "3D Viscoelastic traction force microscopy," *Soft Matter*, vol. 10, no. 40, pp. 8095–8106, Aug. 2014.

[46] J. A. Mulligan, X. Feng, and S. G. Adie, "Quantitative reconstruction of time-varying 3D cell forces with traction force optical coherence microscopy," *Sci. Rep.*, vol. 9, no. 1, Dec. 2019.

[47] L. Dong and A. A. Oberai, "Recovery of cellular traction in three-dimensional nonlinear hyperelastic matrices," *Comput. Methods Appl. Mech. Eng.*, vol. 314, pp. 296–313, Feb. 2017.

[48] D. Shakiba *et al.*, "The Balance between Actomyosin Contractility and Microtubule Polymerization Regulates Hierarchical Protrusions That Govern Efficient Fibroblast–Collagen Interactions," *ACS Nano*, Apr. 2020.

[49] M. Cóndor *et al.*, "Breast Cancer Cells Adapt Contractile Forces to Overcome Steric Hindrance," *Biophys. J.*, vol. 116, no. 7, pp. 1305–1312, Apr. 2019.

[50] "agus raharjo on Iconfinder." [Online]. Available: https://www.iconfinder.com/agusraharj. [Accessed: 20-Oct-2020].

[51] "Minimal disease icons by PictureWindow." [Online]. Available: https://www.iconfinder.com/iconsets/minimal-disease. [Accessed: 20-Oct-2020].

Appendices

A. Experimental protocol

Commercially available pooled wild type human umbilical vein endothelial cells (HUVEC, Angio-Proteomie) were cultured in complete endothelial growth medium (EGM-2, Lonza) and used at passage 4. HUVECs were transduced with adenoviral LifeAct-GFP2 (Ibidi) on the day before hydrogel preparation. Enzymatically crosslinked poly-ethylene glycol (PEG) hydrogels comprised of an MMP-sensitive peptide modified PEG precursor (8-arm 40kDa), Lys-RGD peptide (Pepmic), and 0.1 µM Sphingosine-1-Phosphate (Sigma-Aldrich) were suspended with 0.2 µm red fluorescent carboxylated polystyrene microspheres (ThermoFischer). PEG solution was pipetted into a previously described [30] imaging chamber attached to the glass bottom of a petri dish, and further allowed to crosslink for 30 min at room temperature. A confluent cell monolayer was then achieved by seeding 50,000 LifeAct transduced HUVECs per gel and incubating the dishes vertically at 37°C, 5% CO_2 to allow cell adhesion onto the PEG meniscus. Finally, EGM-2 was added and dishes were placed horizontally at standard conditions in the incubator for 16h before experimentation.

Sprouts invading the PEG hydrogel were imaged using a Leica SP8 inverted confocal microscope with a 25x0.95 NA water-immersion objective to obtain image stacks of 290 x 290 x 45 µm at 0.5 µm z-spacing. The green fluorescent cell channel and the red fluorescent beads channel were simultaneously imaged to obtain two sets of images; first when the beads are in a stressed state and the cells are contractile, and the second when the beads are in a relaxed state and the cells are non-contractile. The stressed state was imaged immediately after placing the dish upon the stage and locating the sprout of interest. The relaxed state was then induced by treating the cells with Cytochalasin D (dissolved in DMSO, Sigma Aldrich) at 4 µM for 50 minutes followed by image acquisition in the same location.

B. Traction recovery by means of PBNIM

PBNIM was first developed in the context of nonlinear elasticity. A detailed description of the numerical implementation of this method is provided in [31]. TFMLAB uses an adapted implementation for linear elasticity, as described in our preprint [32]. Briefly, this method calculates a displacement field, $\mathbf{u}^{\mathrm{calc}}$, that is minimally different from the measured one, $\mathbf{u}^{\mathrm{meas}}$, and that is enforced to fulfil equilibrium of forces in the hydrogel domain as a constraint. This can be mathematically written as follows,

$$\min_{\mathbf{u}_{\mathrm{c}}^{\mathrm{calc}},\mathbf{u}_{\mathrm{h}}^{\mathrm{calc}}} \left\| \frac{1}{2}\left(\mathbf{u}_{\mathrm{c}}^{\mathrm{calc}} - \mathbf{u}_{\mathrm{c}}^{\mathrm{meas}}\right)^2 + \frac{1}{2}\left(\mathbf{u}_{\mathrm{h}}^{\mathrm{calc}} - \mathbf{u}_{\mathrm{h}}^{\mathrm{meas}}\right)^2 \right\| \quad \text{s.t.} \quad \mathbf{F}_{\mathrm{h}} = 0 \qquad \textit{Eq. B.1}$$

where $\mathbf{F}$ is the nodal reaction forces vector which accounts for external or internal prescribed forces or displacements and c and h denote the cell boundary domain and the hydrogel domain, respectively. Using a FEM framework and including the equilibrium constraint as a Langrange multiplier, Eq. B.1 can be rewritten as,

$$\min_{\mathbf{u}_c^{calc},\mathbf{u}_h^{calc}} \left\| \frac{1}{2}\left(\mathbf{u}_c^{calc} - \mathbf{u}_c^{meas}\right)^2 + \frac{1}{2}\left(\mathbf{u}_h^{calc} - \mathbf{u}_h^{meas}\right)^2 + \Theta \cdot \eta \right\| \qquad \textit{Eq. B.2}$$

where Θ is the equilibrium constraint equation and η is the Lagrange multiplier. The minimum of Eq. B.2 can be calculated analytically and rewritten in matrix form as (the details or the formulation are given in [32]),

$$\begin{bmatrix} \mathrm{I} & 0 & \mathbf{K}_{ch} \\ 0 & \mathrm{I} & \mathbf{K}_{hh} \\ \mathbf{K}_{hc} & \mathbf{K}_{hh} & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u}_c^{calc} \\ \mathbf{u}_h^{calc} \\ \eta \end{bmatrix} = \begin{bmatrix} \mathbf{u}_c^{meas} \\ \mathbf{u}_h^{meas} \\ 0 \end{bmatrix} \qquad \textit{Eq. B.3}$$

where $\mathbf{K}_{hc}$, $\mathbf{K}_{ch}$ and $\mathbf{K}_{hh}$ are the parts of the stiffness matrix associated with the two domains c and h.

Next, the strain field at each Gauss point of an element (e) is computed as,

$$\varepsilon^{(e)} = \mathbf{B} \cdot \mathbf{u}^{calc\,(e)} \qquad \textit{Eq. B.4}$$

where $\mathbf{B}$ is the gradient matrix of the shape functions**.** The stress tensor is calculated at each Gauss point of an element (e) based on the linear elastic, isotropic and homogeneous constitutive properties as,

$$\boldsymbol{\sigma}^{(e)} = 2\mu \cdot \varepsilon^{(e)} + \lambda \cdot \mathrm{tr}(\varepsilon^{(e)})\mathbf{I} \qquad \textit{Eq. B.5}$$

where μ and λ are the Lamé constants of the hydrogel**.** $\boldsymbol{\sigma}^{(e)}$ is then averaged from Gauss points to the nodes i of the FE mesh. Tractions are computed at the cell boundary domain using the Cauchy relation,

$$\mathbf{T}_i = \boldsymbol{\sigma}_i \cdot \mathbf{n}_i \qquad \textit{Eq. B.6}$$

where $\mathbf{T}_i$ is the traction vector at the nodes i of the FE mesh and $\mathbf{n}_i$ is the outward normal to node i.

Required Metadata

Current code version

Ancillary data table required for subversion of the codebase. Kindly replace examples in right column with the correct information about your current code, and leave the left column as it is.

Table 1 – Code metadata (mandatory)

Nr	Code metadata description	***Please fill in this column***
C1	Current code version	*v1.0*
C2	Permanent link to code/repository used of this code version	*https://gitlab.kuleuven.be/MAtrix/Jorge/tfmlab_public*
C3	Legal Code License	*LGPL v3.0*
C4	Code versioning system used	*Gitlab*
C5	Software code languages, tools, and services used	*MATLAB*
C6	Compilation requirements, operating environments & dependencies	*Image processing toolbox, signal processing toolbox (MATLAB), DIPImage library, Elastix*
C7	If available Link to developer documentation/manual	https://gitlab.kuleuven.be/MAtrix/Jorge/tfmlab_public/-/blob/public_tfmlab/manual.pdf
C8	Support email for questions	jorge.barrasafano@kuleuven.be