White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/144255/

Version: Published Version

Article:
Smyl, D., Bossuyt, S. and Liu, D. (2019) OpenQSEI: A MATLAB package for quasi static elasticity imaging. SoftwareX, 9. pp. 73-76. ISSN: 2352-7110

https://doi.org/10.1016/j.softx.2019.01.004

Reuse
This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

SoftwareX 9 (2019) 73–76

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

OpenQSEI: A MATLAB package for quasi static elasticity imaging

Danny Smyl [a], Sven Bossuyt [a], Dong Liu [b,c,d,*]

[a] Department of Civil and Structural Engineering, University of Sheffield, Sheffield, UK
[b] CAS Key Laboratory of Microscale Magnetic Resonance and Department of Modern Physics, University of Science and Technology of China (USTC), Hefei 230026, China
[c] Hefei National Laboratory for Physical Sciences at the Microscale, USTC, China
[d] Synergetic Innovation Center of Quantum Information and Quantum Physics, USTC, China

ARTICLE INFO

Article history:
Received 28 February 2018
Received in revised form 26 October 2018
Accepted 7 January 2019

Keywords:
Elasticity imaging
Inverse problems
Medical imaging
Structural health monitoring

ABSTRACT

Quasi Static Elasticity Imaging (QSEI) aims to computationally reconstruct the inhomogeneous distribution of the elastic modulus using a measured displacement field. QSEI is a well-established imaging modality used in medical imaging for localizing tissue abnormalities. More recently, QSEI has shown promise in applications of structural health monitoring and materials characterization. Despite the growing usage of QSEI in multiple fields, fully-executable open source packages are not readily available. To address this, OpenQSEI is developed using a MATLAB platform and shared in an online community setting for continual algorithmic improvement. In this article, we describe the mathematical background of QSEI and demonstrate the basic functionalities of OpenQSEI with examples.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Code metadata

Current code version	v3
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_14
Legal Code License	MIT
Code versioning system used	git
Software code languages, tools, and services used	MATLAB
Compilation requirements, operating environments & dependencies	MATLAB 2017b or Octave 4.2.1 and either Mac OS X, Linux, or Windows
If available Link to developer documentation/manual	
Support email for questions	danny.smyl@aalto.fi

1. Motivation, significance, and background

Quasi Static Elasticity Imaging (QSEI) is a well-established imaging modality used in medical imaging [1–6] and an increasingly popular modality in applications of structural health monitoring and materials characterization [7–14]. Potential applications of QSEI include medical imaging of arteriosclerosis [15] or tissue abnormalities manifested by, for example, breast cancer [3]. Some engineering applications of QSEI include detection of distributed or localized damage [9] and characterization of unknown elastic parameters [14]. Owing to the multiplicity of QSEI applications, researchers have made significant progress in developing QSEI algorithms. Recent improvements have, for example, included speedups using gradient-based approaches [4], the inclusion of non-linear material models [1,2], and implementation of machine learning [10]. Despite these developments, fully-executable open source QSEI software is currently unavailable. To address this, we have developed an open source, MATLAB executable QSEI platform: OpenQSEI. In the following, we will provide the mathematical background for QSEI/OpenQSEI and briefly discuss preliminaries of the mathematical functionalities and their software integration.

The aim of QSEI is to reconstruct the distribution of the elasticity modulus E using a measured displacement field u_m with a computational inverse approach. The observation model for such an inverse problem has the form

$$u_m = U(E) + e \tag{1}$$

where e is the assumed Gaussian-distributed noise and $U(E)$ is the forward model solved using a discretized numerical method (the

* Corresponding author.
E-mail addresses: danny.smyl@gmail.com (D. Smyl), dong2016@ustc.edu.cn (D. Liu).

https://doi.org/10.1016/j.softx.2019.01.004
2352-7110/© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

FEM is adopted herein). In a medical setting, u_m is often provided by either (i) tracking the displacements of speckles using ultrasound or (ii) using phase information from the measured magnetic field obtained through Nuclear Magnetic Resonance Imaging. In structural applications, u_m may be obtained using Digital Image Correlation or discrete measurements, such as strain gauges or extensometers. For any field of application, having the ability to simulate u_m with added noise is an excellent way of testing algorithmic performance, most often this is done using the forward model.

Based on the observation model for Eq. (1), we may write the constrained and regularized Least-Squares (LS) solution as

$$\ell = \arg \min_{E_L < E < E_U} \{\|L_e(u_m - U(E))\|^2 + p_E(E)\} \tag{2}$$

where $p_E(E)$ is the regularization functional of choice, $L_e^T L_e = C_e^{-1}$ where C_e is the observation noise covariance matrix, E_L and E_U are the user-defined constraints, and $\|\cdot\|$ denotes the Euclidean norm. The regularization term is included due to the ill-posed nature of the inverse problem, meaning that standard LS approaches may yield non-unique solutions. The choice of the appropriate regularization functional (prior model) is important in obtaining accurate reconstructions. For example, in cases where sharp features are present, Total Variation (TV) regularization may be most appropriate; however, in cases where E is relatively smooth, smoothness promoting regularization (L_2 and weighted) may provide the best results [16].

To solve the inverse problem, an iterative constrained Gauss–Newton (GN) optimization regime is used. The regime is equipped with a line-search algorithm to determine the step size s_k in the solution $\theta_k = \theta_{k-1} + s_k\bar{\theta}$ where θ_k is the current estimate and $\bar{\theta}$ is the LS update. Such an approach requires the Jacobian $J = \frac{\partial U(E)}{\partial E}$ at each iteration k, which is computed using the perturbation method with the choice of first- or second-order central differencing. In future versions of the code, additional optimization methods will be incorporated, such as full Newton, quasi-Gauss, gradient descent, primal dual interior point methods, etc. For the first published version, the GN method was selected due to its fast minimization of the cost function and its tolerance to errors resulting from inaccuracies in computing J using perturbation [4].

The overall goals of this work are to (a) develop a user-friendly software package where the mathematics are condensed into easily executable and readable MATLAB functions and (b) share the software in an online community setting for continual algorithmic improvement. Based on the mathematics outlined herein and goal (a), we designed OpenQSEI to have the following basic MATLAB functions for each *mathematical functionality*. *Forward model/data simulation:* Meshing, inclusion generation, and FEM solution to the elasticity problem (plane-stress). *Prior models:* L_2 smoothness, weighted smoothness, and TV. *Constrained optimization:* Cost function (cf. Eq. (2)), barrier functions for constraints, generation of gradients/Hessians, and line search. In addition, functionalities for plotting and visualizing are included.

2. Software description

2.1. Software overview

OpenQSEI is designed to allow the user to easily access different prior models and optimization functionalities for imaging the elastic modulus using experimental or simulated displacement fields. To do this, the OpenQSEI environment includes the essential functions for (i) simulation of data and a FEM implementation of the forward model (ii) three commonly used prior models (TV, L_2, and weighted smoothness),[1] (iii) estimation of the optimization starting point (best homogeneous estimate), constraints using quadratic polynomial barrier functions, computation of the Jacobian, and line search used during optimization and (iv) determination of gradients and Hessians related to the prior models and constraints. A flow chart outlining the basic software architecture is highlighted in Fig. 1.

2.2. Software functionalities

The primary software functionalities of OpenQSEI aim to solve the forward problem, provide appropriate prior models, and optimize the QSEI inverse problem. For the forward in-plane elasticity problem, the package is equipped with a FEM regime, `FMDL`. `FMDL` uses piece-wise linear triangular elements, which are discretized utilizing user input geometry by employing MATLAB Delaunay Triangulation via `meshgen`.

Prior models using weighted smoothness and TV are provided through the implementation of functions `WeightedSmoothnessPrior` or `getTVMat`. For `WeightedSmoothnessPrior`, the user may input spatial correlation length and the predicted range of E. When employing `getTVMat`, the user may tailor the TV stabilization parameter β and weighting parameter α for the physics of his/her particular problem. The L_2 prior, on the other hand, is simply generated as a multiple of the identity matrix $\mathbf{I}$, i.e. $\lambda\mathbf{I}$. Therefore only user input of the scalar λ and generation of $\mathbf{I}$ are required during the optimization regime.

To solve the inverse problem, the main MATLAB file `OpenQSEI` contains the shell LS-GN iterative optimization regime calling the necessary optimization functions. The optimization regime is executed after initialization of all required parameters, such as meshing, initial guess via `BestHomogeneousE`, prior matricies, etc. The optimization, which runs until the residual $\frac{\ell_k - \ell_{k-1}}{\ell_{k-1}}$ is below some user-defined tolerance (tol), includes iterative updating of the cost function, step size, and appropriate gradients/Hessians via `costfun`, `linesearch_uniform`, and `gradshessians`. Throughout the optimization process, the user is updated with useful figures including the minimization of the cost function, step-size calculation, and data mismatch. A representative snippet of the optimization plots is shown in Fig. 2 after four iterations using the L_2 prior.

3. Illustrative example — reconstruction of a unicorn

In the following, we provide an example of a unicorn-shaped elastic inclusion in a stretched plate. The user interface is entirely included within the main MATLAB file `OpenQSEI.m`. For this exercise, pre-loaded boundary conditions, external forces, and the image of a unicorn superimposed onto the meshing (by interpolating gray-scale pixel intensities of a unicorn image onto the triangulation and normalizing with respect to the maximum elastic modulus value) are contained in the files `cond.mat`, `force.mat`, and `Eunicorn.mat`, respectively. The meshing and boundary condition data are provided in Table 1. Once the user has opened the file, he/she must change the directory to the OpenQSEI folder on their computing system. After this preliminary, OpenQSEI is fully operable and the user may select the choice of prior model and constraints and execute the code.

In this example, we reconstruct the true image of the unicorn, with 1.0% noise standard deviation added to the simulated displacement field, using all three prior models (TV, weighted

[1] For a comprehensive description of TV, L_2, and weighted smoothness regularization techniques, we refer the reader to [16–18].

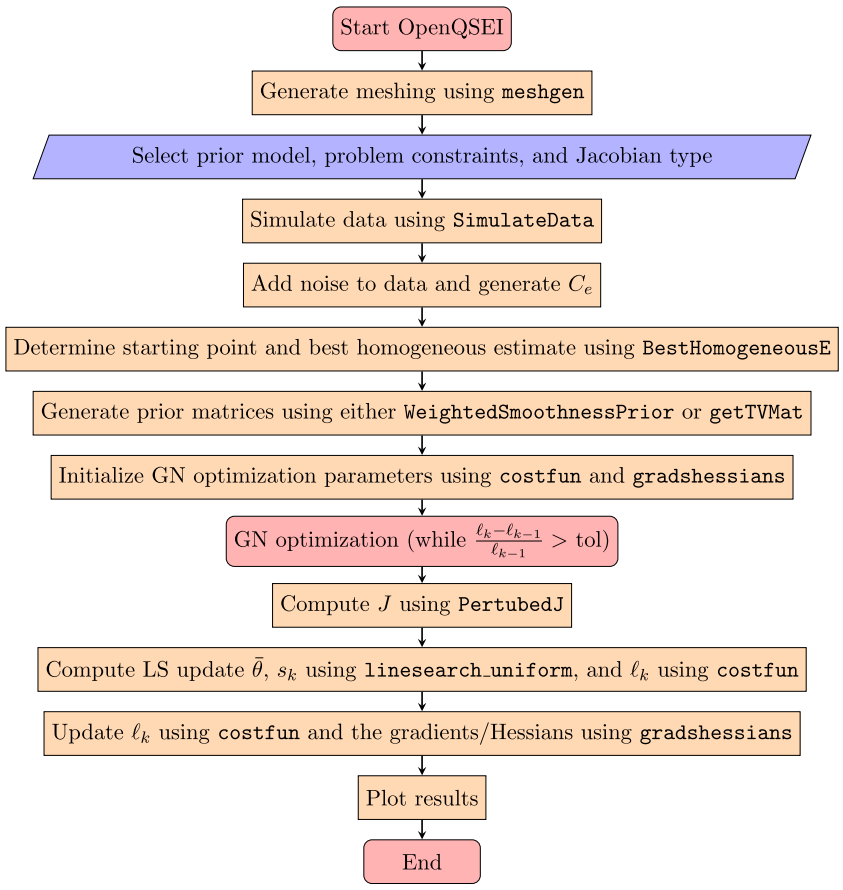

Fig. 1. Outline of the OpenQSEI architecture and primary functions. Note that in the functions `costfun` and `PertubedJ` require evaluation of the forward model `FMDL`.

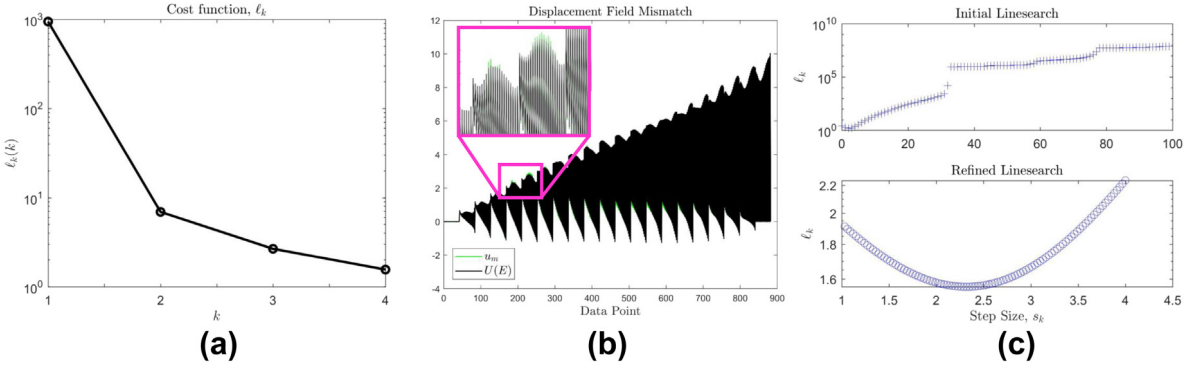

Fig. 2. Figures generated during the optimization portion of OpenQSEI, (a) minimization of the cost function $\ell_k(k)$, (b) data mismatch of the measured u_m and simulated displacement field $U(E)$, and (c) visualization of the line search using `linesearch_uniform`.

Table 1
Summary of the FEM mesh used in the illustrative examples. Note that (i) x and y refer to the horizontal and vertical directions, respectively, and (ii) the node numbers are given by the meshing output automatically provided by the `SimulateData` function.

Nodes (#)	Elements (#)	Fixity, x (node #s)	Fixity, y (node #s)	Applied force, x (node #s)	Applied force, y (unitless force/node)
441	800	1–21	1-21, 421–441	421–441	2.75

smoothness, and L_2 regularization). We note that no attempt was made to optimize the regularization parameters; however, the user may do so at his/her own discretion. The constraints selected were $E_{\text{L}} = 0 < E < E_{\text{U}} = 300$ and the range of E in the true unicorn image was $80 \leq E \leq 260$, where the units of E are arbitrary. Moreover, the stopping criteria during optimization were selected as $\frac{\ell_k-\ell_{k-1}}{\ell_{k-1}} \leq 10^{-6}$. Reconstructions for this example are shown in Fig. 3.

Fig. 3 demonstrates that all three priors resulted in a satisfactory estimation of E and captured the overall shape of the unicorn. It should be remarked that the true image has sharp edges, which is best suited for TV prior and is visible in the overall improved

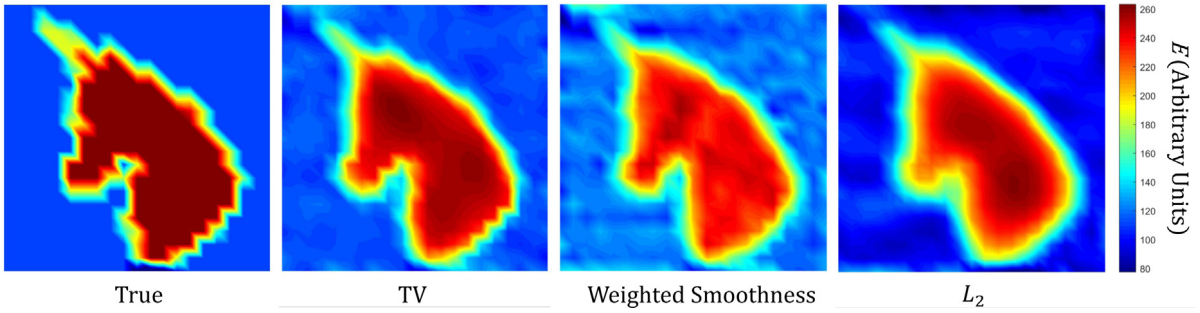

Fig. 3. Reconstructions using OpenQSEI with three different prior models (labeled below each figure).

reconstruction quality relative to the smoothness priors. In cases where the unicorn or other distributions of E are more smoothly dispersed, L_2 and weighted smoothness priors may be more appropriate prior models. We would like to note that, since the noise is generated randomly for each run of the code, some differences in reconstructions are expected due to the ill-posed nature of the QSEI problem.

4. Impact

Despite the clear relevance of QSEI in multiple research facets, to the best of our knowledge, there are no open-source QSEI packages available. To that end, OpenQSEI is developed using a MATLAB-based platform to solve the non-linear ill-posed problem in a least-squares framework. OpenQSEI users have the flexibility to select, experiment, and solve QSEI problems with different prior models, constraints, optimization parameters, geometries, boundary conditions, force fields, and more.

Due to the rapid advances with the field of QSEI over the recent years by numerous researchers (for example, [1,2,10]), we have decided to make the software package OpenQSEI a digital, communal project, which will allow users to continually improve OpenQSEI via the online platform (see references for additional details). Potential near-term advances improving the applicability and performance of OpenQSEI over the coming years may include (i) a faster semi-analytical regime for determining the Jacobian, (ii) higher-order h, p, k forward models [19], (iii) parallelization of the forward model for large-scale problems, (iv) additional prior models, for example, L_1, Lasso, and Ridge Regression methods, (v) estimation of unknown boundaries and/or forces, (vi) non-Gaussian noise models, and (vi) different optimization approaches, such as full Newton, quasi-Gauss, gradient descent, primal dual interior point methods, and more. Moreover, we look forward to extending the 2D QSEI to a 3D version utilizing data obtained from, e.g. Digital Volume Correlation.

5. Conclusions

In this article, we presented the MATLAB-based Quasi Static Elasticity Imaging (QSEI) package: OpenQSEI. The software promotes the open use of QSEI for users in a variety of fields, including medical imaging a structural engineering. The present version of the software allows users the ability to solve QSEI problems with different prior models, constraints, optimization parameters, geometries, boundary conditions, force fields, and more. In the future, via the OpenQSEI online platform, we hope to engage other researchers to further develop and improve the capabilities of OpenQSEI for a large suite of potential applications.

Acknowledgments

Authors DS and SB would like to acknowledge the support of the Department of Mechanical Engineering at Aalto University, Finland throughout this project. DL was supported by Anhui Provincial Natural Science Foundation, China (1708085MA25). We would like to thank Chris Wolfe for his assistance in establishing the github repository.

References

[1] Goenezen S, Barbone P, Oberai AA. Solution of the nonlinear elasticity imaging inverse problem: The incompressible case. Computer Methods in Applied Mechanics and Engineering 2011;200(13):1406–20.
[2] Gokhale NH, Barbone PE, Oberai AA. Solution of the nonlinear elasticity imaging inverse problem: the compressible case. Inverse Problems 2008;24(4):045010.
[3] Konofagou E. Quo vadis elasticity imaging? Ultrasonics 2004;42(1-9):331–6.
[4] Oberai AA, Gokhale NH, Feijóo GR. Solution of inverse problems in elasticity imaging using the adjoint method. Inverse Problems 2003;19(2):297.
[5] Parker K, Gao L, Lerner R, Levinson S. Techniques for elastic imaging: a review. IEEE Engineering in Medicine and Biology Magazine 1996;15(6):52–9.
[6] Ophir J, Cespedes I, Ponnekanti H, Yazdi Y, Li X. Elastography: a quantitative method for imaging the elasticity of biological tissues. Ultrasonic Imaging 1991;13(2):111–34.
[7] Smyl DJ, Antin K-N, Liu D, Bossuyt S. Coupled Digital Image Correlation and Quasi-Static Elasticity Imaging of inhomogeneous orthotropic composite structures. Inverse Problems 2018.
[8] Mei Y, Stover B, Kazerooni NA, Srinivasa A, Hajhashemkhani M, Hematiyan M, Goenezen S. A comparative study of two constitutive models within an inverse approach to determine the spatial stiffness distribution in soft materials. International journal of mechanical sciences 2018;140:446–54.
[9] Smyl D, Bossuyt S, Liu D. Stacked elasticity imaging approach for visualizing defects in the presence of background inhomogeneity. Journal of Engineering Mechanics 2018. (in press).
[10] Hoerig C, Ghaboussi J, Insana MF. An information-based machine learning approach to elasticity imaging. Biomechanics and modeling in mechanobiology 2017;16(3):805–22.
[11] Karageorghis A, Lesnic D, Marin L. The method of fundamental solutions for three-dimensional inverse geometric elasticity problems. Computers & Structures 2016;166:51–9.
[12] Mei Y, Kuznetsov S, Goenezen S. Reduced boundary sensitivity and improved contrast of the regularized inverse problem solution in elasticity. Journal of Applied Mechanics 2016;83(3):031001.
[13] Moussawi A, Lubineau G, Florentin E, Blaysat B. The constitutive compatibility method for identification of material parameters based on full-field measurements. Computer methods in applied mechanics and engineering 2013;265:1–14.
[14] Bonnet M, Constantinescu A. Inverse problems in elasticity. Inverse Problems 2005;21(2):R1.
[15] Barbone PE, Gokhale NH. Elastic modulus imaging: on the uniqueness and nonuniqueness of the elastography inverse problem in two dimensions. Inverse problems 2004;20(1):283.
[16] Mueller JL, Siltanen S. Linear and nonlinear inverse problems with practical applications. SIAM; 2012.
[17] Strong D, Chan T. Edge-preserving and scale-dependent properties of total variation regularization. Inverse problems 2003;19(6):S165.
[18] Kaipio JP, Kolehmainen V, Vauhkonen M, Somersalo E. Inverse problems with structural prior information. Inverse problems 1999;15(3):713.
[19] Surana KS, Reddy J. The Finite Element Method for Boundary Value Problems: Mathematics and Computations. CRC Press; 2016.