Manual for microstructure pattern recognition tool

Ondřej Rokoš, Vlastimil Králík

Cech Technical University
Faculty of Civil Engineering
Department of Structural Mechanics
Thákurova 7, 166 29 Prague, Czech Republic

September 21, 2012

Contents

1 Introduction

Application can be used to characterize the structure of highly porous material. Mechanical properties of porous material depends directly on the shape and structure of pores. The obtained geometric parameters (such as distribution of pore walls thicknesses, pore size and shape characteristics) of the highly porous structure are crucial for the selection of the homogenization schemes and subsequent modeling mechanical performance.

2 General notes

Program is divided into three independent parts `CONTOUR`, `MESH` and `SOLVE` to be used sequentially or independently.

`CONTOUR` part serves for pattern recognition in microstructure from image input file and saves recognized data as `*.txt`, `*.xls`, `*.mat` files. Recognized data includes area moments of inertia, centroids, etc.

`MESH` uses previously generated data, decomposes geometry and generates triangular mesh saved in `*.mat` file.

Plane elasticity problem under boundary conditions is solved in `SOLVE` part.

3 Unpacking

Downloaded file `Data.zip` should contain three directories named `gui`, `manual` and `mfiles`. Three ways of running the code are possible: standalone executable gui (graphical user interface), source files gui, source mfiles (no gui—intended for implementation in user's code).

Running the code using source files gui or source mfiles requires full installation of Matlab®. Standalone executable gui requires free MATLAB Compiler Runtime (MCR), version 7.17 R2012a components which can be downloaded from the web pages
http://www.mathworks.com/products/compiler/mcr/index.html.

4 Running standalone executable gui

Note: MATLAB Compiler Runtime (MCR), version 7.17 R2012a components should be installed first.

Following path `Data/gui/executable/win**` three files `GUI_CONTOUR.exe`, `GUI_MESH.exe` and `GUI_SOLVE.exe` are found together with benchmark data example `data.jpg`. Executables are compiled for win32 and win64 OS.

4.1 Gui Contour

Graphical interface of `GUI_CONTOUR.exe` is depicted in fig. 1. Window is divided into four boxes.

Contour data box contains *Name of input* where input file is to be specified (`*.jpg`, `*.bmp`, `*.gif`, `*.tif`) and *Pixel size* which assigns specific length to pixel distances. Program recognizes all contours, but only these with area greater than certain limit are to be taken into account. This limit is specified in *Downlimit area*. Note that *Downlimit area* should consider *Pixel size*.

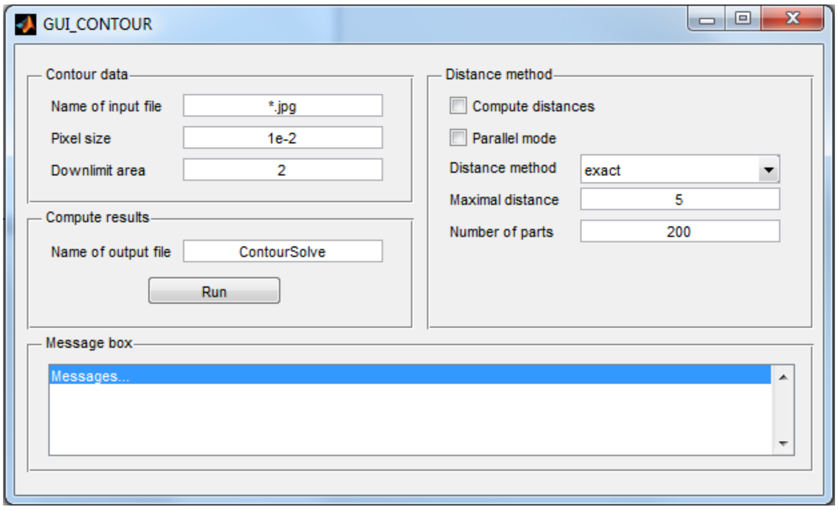


Figure 1: `GUI_CONTOUR.exe` graphical interface

Second box **Distance method** contains checkbox *Compute distances*. Recognized contours have various neighbours with different "wall thicknesses" or "distances". Marking a box enables this functionality. Checkbox *Parallel mode* utilizes multiprocessing during computation. Two methods are available: exact and approximate, which is to be specified in pop-up menu *Distance method*. *Maximal distance* specify what the neighbour is: contours closer to parent are labeled as neighbouring if their distance from parent is less than *Maximal distance*. When approximate method selected, *Number of parts* has to be specified. Program uses halving of interval method along the curve and initial number of intervals has to be specified.

In **Compute results** box name of the output file is entered. As default is set to `ContourSolve`. Push button *Run* starts the evaluation.

Program prints reports to **Message box** part of the window.

After data processing, three output files are generated: `ContourSolve.txt`, `.xls`, `.mat`. These contain variables listed in table 1. `ContourSolve.mat` is a matlab data file. Here are saved `BB` - width of the image, `HH` - height of the image, `PixelSize` - specific length assigned to pixel distance variables and `Database` - structure containing contour data.

4.2 Gui Mesh

Graphical interface is depicted in fig. 2. Window is also divided into four boxes. In **Input parameters** box *Name of input file* is specified. This file is generated in previous step employing `GUI_CONTOUR.exe`, as default is set to `ContourSolve.mat`. Next input is *Jiggle mesh iteration*. Each iteration increases triangle quality, default value is set to 20. *Triangle edge max. size* specifies maximal length of triangle edge. Too large value causes that mesh is not generated.

Output parameters box specifies *Name of output mesh file* and push button *Generate mesh* which launches the generator.

Mesh plotting is available in **Plot parameters** box. *Numbering* specifies which nodes will be numbered in consecutive plot. This is important for boundary conditions and results interpretation. Several possibilities are available such as no numbering, edge points only, all points, triangles only, all points and triangles. Push button *Redraw* updates the plot.

Table 1: Variables in output file generated by `GUI_CONTOUR.exe`

`Fr. with`	ratio of inclusion to matrix in the picture with inhomogeneities
`Fr. without`	ratio of inclusion to matrix in the picture without inhomogeneities
`ID`	identification number of area defined by particular contour
`Tx`	x-coordinate of centroid
`Ty`	y-coordinate of centroid
`A`	area circumscribed by particular contour
`L`	circumference of particular contour
`Iuu`	the second centroidal moment of area with respect to x-axis
`Ivv`	the second centroidal moment of area with respect to y-axis
`Iuv`	the product centroidal moment of area with respect to xy-axis
`I1`	the second moment of area with respect to first principal axis
`alpha1`	angle between the first principal and x-axis
`I2`	the second moment of area with respect to second principal axis
`alpha2`	angle between the second principal and x-axis
`J`	polar area moment of inertia

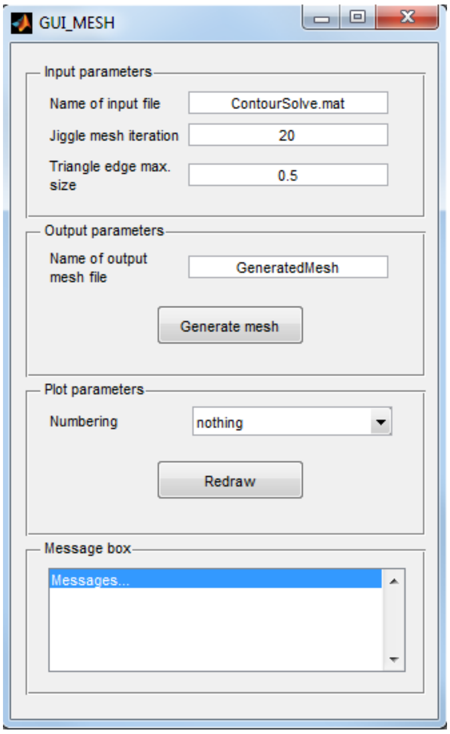


Figure 2: `GUI_MESH.exe` graphical interface

Program prints reports to **Message box** part of the window.

Mesh parameters are saved in `GeneratedMesh.mat` file as default. It contains **p**, **e** and **t** variables which are point, edge and triangle matrices.

4.3 Gui Solve

GUI_MESH.exe interface is depicted in fig. 3. **Input parameters** box specifies *Mesh input file* created by GUI_MESH.exe and boundary conditions in *BC input file*. As default is set to bcinput.txt. Example of the file can be found in the same directory or it's contents is written below:

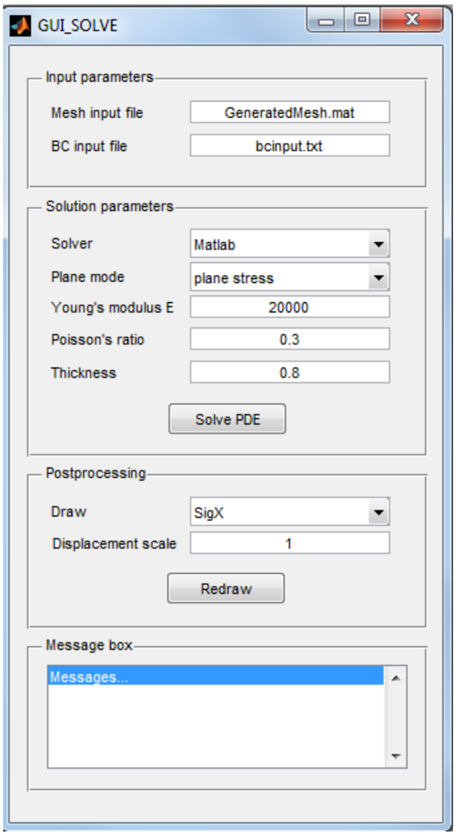


Figure 3: GUI_SOLVE.exe graphical interface

```
NodeSupportX = [1313 1312 701 702 52 716 1309:1311 90 717 1301:1308];
NodeSupportY = [1313 1312 701 702 706 1287 708 1288:1290 710 1291:1294
712 1295 1296 697 1252:1265 698 699 1266:1273];
NodeDisplcContrlX = [153 1274:1282 104 1283:1286];
NodeDisplcContrlY = [];
ux = 5;
uy = 0;
NodeForceX = [];
NodeForceY = [];
Fx = 0;
Fy = 0;
```

Here NodeSupportX specifies nodes supported in X-direction $u_x = 0$, NodeSupportY specifies nodes supported in Y-direction $u_y = 0$, NodeDisplcContrlX specifies nodes with prescribed x-displacements with magnitude ux, NodeDisplcContrlY specifies nodes with prescribed y-displacements with magnitude uy, NodeForceX specifies nodes loaded in x-direction with forces of magnitude Fx, NodeForceY specifies nodes loaded in y-direction with forces of magnitude `Fy`. Colon operator `:` is commonly used in Matlab with meaning `1283:1286 = 1283 1284 1285 1286`.

Solution parameters box specifies as first *Solver* type. Two possibilities are available: Matlab and Oofem. Oofem solver can be downloaded from web pages http://www.oofem.org/en/oofem.html and requires external postprocessor, e.g. ParaView. For this option, `oofem.exe` file has to be situated in the same folder as `GUI_SOLVE.exe`. *Plane mode* specifies whether plane strain or plane stress mode will be used. *Young's modulus E*, *Poisson's ratio* ν are physical parameters of isotropic linear-elastic material, *Thickness* stands for thickness of the plane object.

Postprocessing box offers *Draw* pop-up menu where particular components of stress or strain tensors can be selected. Further choice is displacement with scale factor specified in *Displacement scale* input. Push button *Redraw* updates the plot.

Important messages are printed in **Message box**.

External solver generates several output files, for details see reference web pages (outputs are named `PdeOofemSolve.out` and `PdeOofemSolve.vtu`). When Matlab solver used, output file `PdeMatlabSolve.txt` is generated. Data are here structured as follows:

```
DISPLACEMENTS OUTPUT:
---------------------

Node 1:  ux, uy
Node 2:  ux, uy
...
ELEMENT OUTPUT:
---------------
Element 1:  EpsX, EpsY, EpsZ, GamXY
            SigX, SigY, SigZ, TauXY
...
REACTIONS OUTPUT:
-----------------
Node 52:  Rx, Ry
Node 90:  Rx, Ry
...
```

Node and triangle labels can be viewed in `GUI_MESH` plot.

5 Running source file gui

Using source files gui requires full installation of Matlab®. Code is situated in `Data/gui/source/`, subdirectories `GUI_CONTOUR`, `GUI_MESH` and `GUI_SOLVE`. Graphical interfaces are launched executing `GUI_CONTOUR.m`, `GUI_MESH.m` and `GUI_SOLVE.m` files. Consecutive procedure is the same as in chapters 4.1, 4.2 and 4.3.

6 Running source mfiles

These are intended for implementation in user's code and full installation of Matlab® is required. Source files are situated in `Data/mfiles/` folder. At the beginning of each file necessary input parameters are specified.

In `GUI_CONTOUR` these are `NameInput` string, `PixelSize`, `DownlimitArea`, `ComputeDistances`, `DistanceMethod`, `ParallelMode`, `MaxDist` and `nParts` variables. Meaning of each variable can be found in section 4.1.

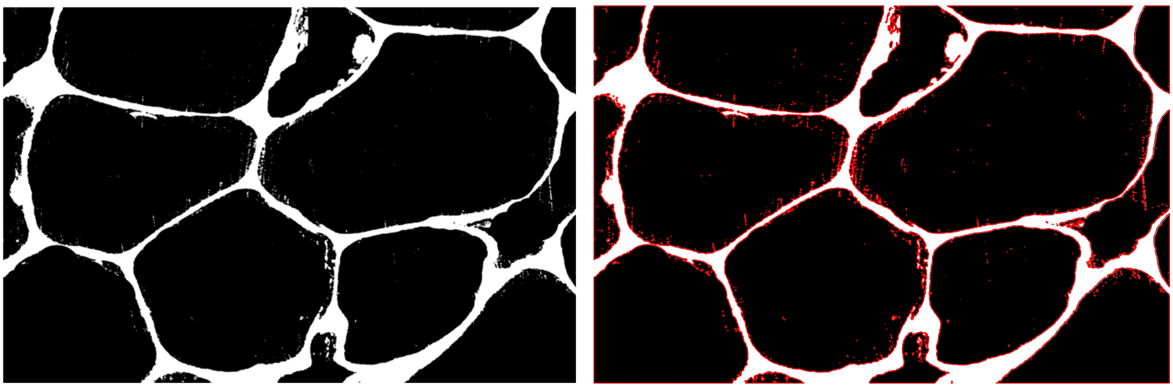
`GUI_MESH` requires `HMax`, `JiggleIter` and `Numbering` variables, cf. section 4.2.

`GUI_SOLVE` then requires `PdeMode`, `PlaneMode` variables, material constants `EModulus`, `Nu`, `Thickness` and boundary conditions `NodeSupportX`, `NodeSupportY`, `NodeDisplcContrlX`, `NodeDisplcContrlY`, `ux`, `uy`, `NodeForceX`, `NodeForceY`, `Fx` and `Fy` variables. Their description can be found in section 4.3.

File `red.m` serves for redrawing output data. Typing `red + ENTER` in Matlab command line after linear elasticity computation, menu with plot options will appear. `matlabSolve.m`, `oofemSolve.m`, `minimalDistance.m` are supporting functions.

7 Example—aluminium foam

Benchmark example `data.jpg` can be found in each subfolder. Initial image is depicted in fig. 4 (a). Fully recognized contours can be found in fig 4 (b) emphasized with red colour. Small inclusions are to be skipped, thus *Downlimit area* is set to 2. Final solution is in figure 5. Consecutively, `GUI_MESH.exe` is launched and mesh generated, cf. fig. 6 (a). Boundary conditions according to tensile test along x-axis are specified, see 6 (b) and section 4.3 `bcinput.txt`.



(a) Input image (b) Fully recognized contours

Figure 4: Benchmark example `data.jpg`

Finally, linear elasticity is solved. All parameters in all three steps are left as default. Deformation with σ_x stress are depicted if fig. 7 (a), (b). This example primarily serves for testing correct functionality of the program. Computed results can be compared with data here introduced.

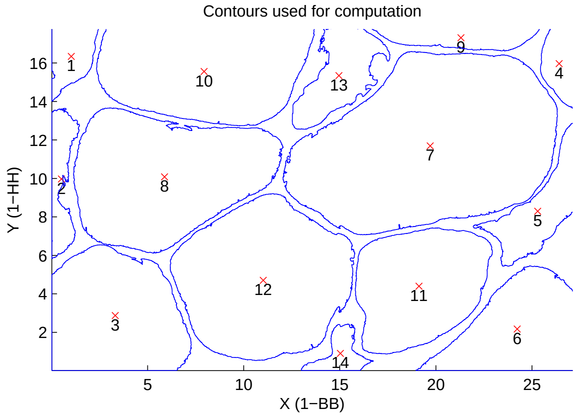
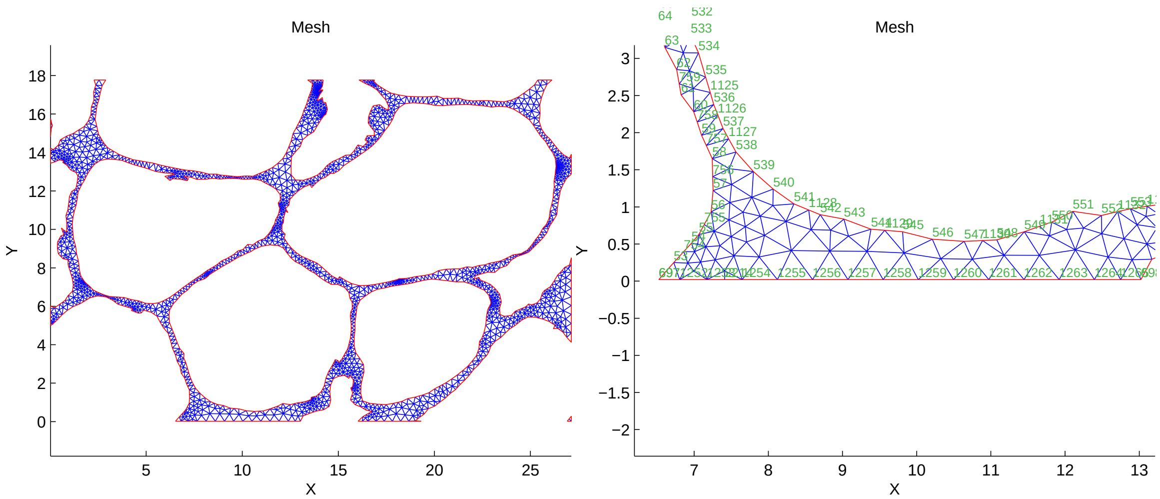


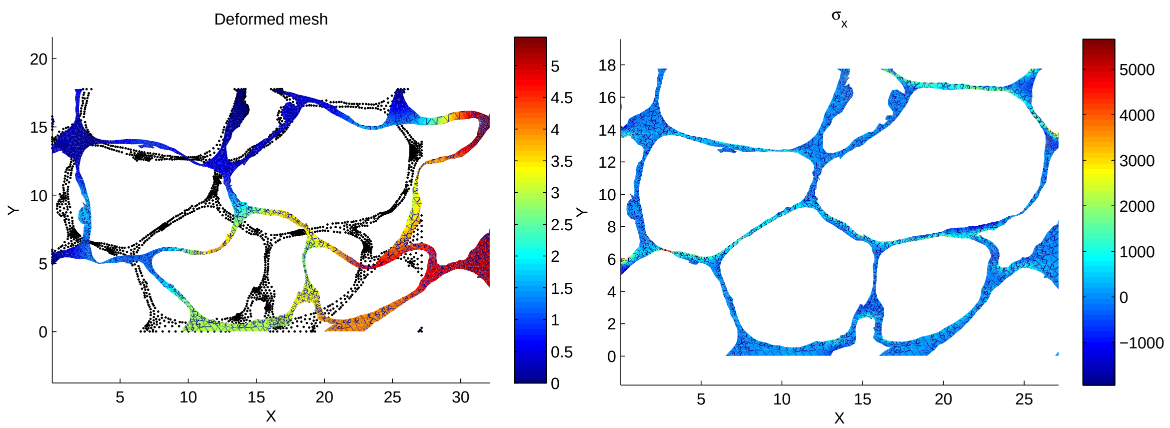
Figure 5: Accepted contours



(a) Mesh

(b) Mesh—detail

Figure 6: Generated mesh



(a) Displacement

(b) σ_x

Figure 7: Linear elasticity