A MATLAB library
for the virtual element method

VEMLAB Primer

Version 2.4

January 2022

Copyright and License

CAMLAB
Computational and Applied Mechanics Laboratory
Department of Mechanical Engineering
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile
Av. Beauchef 851, Santiago 8370456, Chile

TABLE OF CONTENTS

1 Summary of updates

From VEMLAB v2.3 to VEMLAB v2.4:

- Add several options for in plot_and_output_options.m. New options: delete existing output files, plot front/isometric view of the figure, print and save pdf figures, save .fig figures, set figure resolution, color system for the colorbar, colorbar ticks and lines, figure ticks, figure grid, set min/max limit of the colorbar.
- Several enhancements to the output figures.
- Improvements for plotting of FEA results on FEM2DQ4 elements.
- Add several example test problems in "test" folder.
- Create a polygonal mesh from a 3-node triangular mesh (see folder "mesher").
- Three stabilization options are now available in LinearElastostatics module.
- Mesh size parameter is now computed as the average area of the polygons in the mesh.
- Update VEMLAB Primer.

From VEMLAB v2.2.2 to VEMLAB v2.3:

- Add possibility of using multiple materials in the Poisson module.
- Update triangulate_polygon function so that it uses the built-in triangulation matlab function.
- Bug fix in writing to txt file.
- Add possibility to write output files suitable to be read in Convex Polygon Packing (CPP) program.
- Add some options for controlling the plotting of the mesh (plot_mesh_linewidth, plot_mesh_nodes, plot_mesh_nodesize, plot_mesh_axis).
- Add sparse solver to VEM2D/FEM2D in LinearElastostatics and Poisson modules.
- Remove L-shape example from the test folder due to bad behavior in plotting stresses and strains.

From VEMLAB v2.2.1 to VEMLAB v2.2.2:

- Fix function max_edge_size.m.
- Add an L-shape example in the test folder.
- Update calculation of the polygon's area: MATLAB's "polyarea.m" is now used.

From VEMLAB v2.2 to VEMLAB v2.2.1:

- Add option to explicitly switch off all MATLAB figures in function "plot_and_output_options.m".
- Facilitate compatibility to run VEMLAB in Octave.
- Update manual with a guide to running VEMLAB in Octave.

From VEMLAB v2.1 to VEMLAB v2.2:

- Fix disp() in plot_and_ouput_options.m: disp("Hello") seems to work only in newer versions of MATLAB. So, it is changed to the standard MATLAB format disp('Hello').
- Results that are postprocessed in the graphical user interface of GiD are now ordered in folders.
- Add option to plot deformed domain in MATLAB figures when using the LinearElastostatics module (see function "plot_and_ouput_options.m" located in the folder "config".)
- Add a function to read a meshfile having the domain type declared as "Custom", which is useful for defining the meshfile manually or using an external mesh generator or using a customized version of the mesh generators available in

VEMLAB. (See example "Creating and using a custom meshfile" in the VEMLAB Primer for details.)
- Add more details to the VEMLAB Primer.

From VEMLAB v2.0.2 to VEMLAB v2.1:
- Add customized wrench domain (for PolyMesher mesh generator only).
- Add customized plate with a hole domain (for PolyMesher mesh generator only).
- Add the following test: "square_plate_with_source2_poisson2d.m" in test folder.
- Add the following test: "plate_with_hole_linelast2d.m" in test folder.
- Add the following test: "wrench_linelast2d.m" in test folder.
- Fix iteration counter in PolyMesher function.

From VEMLAB v2.0.1 to VEMLAB v2.0.2:
- Fix several bugs when using vemlab_method='FEM2DQ4' and vemlab_method = 'FEM2DT3'.
- Add a control variable in config.m that permits to explicitly set the number of Gauss points to integrate the FEM2DQ4 stiffness matrix and body force vector.

From VEMLAB v2.0 to VEMLAB v2.0.1: the following features have been added
- More detailed manual in folder “doc.”
- Improvement to the plotting of axis and fonts in MATLAB figures.

From VEMLAB v1.0 to VEMLAB v2.0: the following features have been added
- Two-dimensional Poisson problem
- Setup of plot and output options in function “plot_and_output_options” located in folder “config.”
- Additional plotting options (stresses, strains, fluxes and gradients) to MATLAB figures, text files and GiD files.
- Option to plot solutions to VTK files.

VEMLAB 1.0: (Initial release of code)
- Two-dimensional linear elastostatics (plane strain and plane stress)
- Solution methods: VEM (polygonal elements), FEM (3-node triangles, 4-node quadrilateral)
- Boundary conditions: Dirichlet, Neumann on boundary edges; can be a constant or a function.
- Meshers: PolyMesher, distmesh2d, quad4mesh; customized for rectangular domains only (requires adjustments for other domain types)
- Meshes need to be generated separately and stored inside folder “mesh_files” located in the folder “test.”
- Meshes must be generated with the functions “create_” located in the folder “mesher.”
- Solutions can be plotted to MATLAB figures, text files and GiD files.

2 Features of VEMLAB

VEMLAB is a free and open source MATLAB library for the virtual element method.

Features:

- Two-dimensional linear elastostatics (plane strain and plane stress) and two-dimensional Poisson problem.

- Solution methods: linear VEM (polygonal elements), FEM (3-node triangles, 4-node quadrilaterals).
- Boundary conditions: Dirichlet, Neumann on boundary edges; can be a constant or a function.
- Meshers: PolyMesher, distmesh2d, quad4mesh; PolyMesher is customized for rectangular domain, wrench domain and plate with a hole domain; distmesh2d and quad4mesh are customized for rectangular domain only. Domains can be extended for any of the meshers, but it requires adjustments to some interface functions (see the instructions that are available in functions create_polygonal_mesh.m, create_quadrilateral_mesh.m and create_triangular_mesh.m in folder "mesher").
- Meshes need to be generated separately and saved to folder "test/mesh_files."
- Meshes must be generated with the functions "create_" located in the folder "mesher." Then, the files containing the generated meshes will be automatically saved to folder "test/mesh_files" for their use.
- Solutions can be saved to MATLAB figures; plotted and saved to PDF figures; saved to text files, GiD files and VTK files.

3 Source code

All the information related to VEMLAB and its source code is available on the web:

https://camlab.cl/software/vemlab/

Download the code before proceeding with the rest of this primer.

4 First steps

Once the software is downloaded there are two fonts that need to be installed so that the output figures are plotted correctly. If these fonts are not present, MATLAB will use default fonts. These fonts are Segoe UI Semibold and Good Times RG. For Windows systems, these fonts are included in folder "vemlab/utilities/fonts". The instructions to install these fonts are also provided in the same folder. For completeness, the instructions are also provided here.

Segoe UI Semibold: This font must be installed directly in your operating system. In Windows system this is done in Settings-->Fonts or just type "Font settings" in Window's search utility.

Good Times RG: This font must be added to the Java Runtime Environment (JRE) that ships with MATLAB (if that is the Java version being used by the program). Below are steps to install the fonts:

1. Copy the TTF fonts to "<matlabroot>\sys\java\jre\win64\jre\lib\fonts", e.g., "C:\Program Files\MATLAB\R2019a\sys\java\jre\win64\jre\lib\fonts". Note: You might need to be an administrator to copy files to this folder.

2. Restart MATLAB.

3. You should now be able to see the fonts in MATLAB's font preference panel. Note: If using the default JVM shipped with MATLAB, you must do this for each MATLAB version installed.

5 Up and running with VEMLAB

VEMLAB is a library. You need to create a main .m file and place it inside the folder "test." The main file has the typical structure of a FEM simulation. Simply follow the test problems (they are given with detailed comments) that are provided inside the folder "test" to write your own .m files or modify the ones provided. More details are given below.

To run a simulation a main .m function must be prepared. This main function, when executed, will start the simulation, and drive it until its end. Several examples of main functions are provided in the folder "test."

The folder "test/mesh_files" has various text files that contain the information of predefined meshes that are ready to be used and read from the example main files that are available in the folder "test." To generate new meshes, the user must use the following functions that are available with instructions in the folder "mesher":

- create_polygonal_mesh.m for polygonal elements.
- create_quadrilateral_mesh.m for four-node quadrilateral elements.
- create_triangular_mesh.m for three-node triangular elements.
- create_polygonal_mesh_from_T3_mesh.m for polygonal elements starting from a 3-node triangular mesh.

The previous functions create text files that contain the information of the generated meshes and save them to the folder "test/mesh_files" so that they are available to be read from the main functions.

Meshes can also be generated from other sources (or manually) provided that the format of the VEMLAB meshes is followed (see example mesh files in folder "test/mesh_files"). A manual creation of a mesh file is detailed later in this primer.

Important note about mesh generation: create_polygonal_mesh.m is a wrapper for PolyMesher, create_triangular_mesh.m is a wrapper for distmesh2d, and create_quadrilateral_mesh.m is a wrapper for quad4mesh; PolyMesher is customized for rectangular domain, wrench domain and plate with a hole domain; distmesh2d and quad4mesh are customized for rectangular domain only. Domains can be extended for any of the meshers, but it requires some adjustments to some interface functions (see the instructions that are available in functions create_polygonal_mesh.m, create_quadrilateral_mesh.m and create_triangular_mesh.m in folder "mesher").

The following must be considered when preparing the main file:

The method to be used in the simulation is specified by the variable vemlab_method. For instance, to perform the simulation with

- VEM, declare this variable as vemlab_method='VEM2D'.
- FEM, declare this variable as vemlab_method='FEM2DQ4' for four-node quadrilateral elements, and vemlab_method='FEM2DT3' for three-node triangular elements.

The four-node quadrilateral and three-node triangular elements are particular instance of polygonal elements, and as such they can be used to simulate with the VEM specifying vemlab_method='VEM2D'. However, polygonal meshes with elements of more than three edges cannot be used when specifying vemlab_method='FEM2DT3', and polygonal meshes with elements of three or more than four edges cannot be used when specifying vemlab_method='FEM2DQ4'.

Before starting the simulation, it is important to setup the options for the plots and output files. These options can be activated or deactivated in the function plot_and_output_options.m that is located in the folder "config."

The output files created with the simulation are saved to folder "test/output_files." Inside this folder there are five subfolders that contain specific output files, as follows:

- Folder "GiD" contains output files that are readable in the postprocessor of GiD (https://www.gidhome.com/).
- Folder "VTK" contains output files that are readable in the Visualization Toolkit VTK (https://www.vtk.org/) or in the visualization application Para-View (https://www.paraview.org/).
- Folder "txt" contains the output files in text format.
- Folder "CPP" contains output files that are readable in the Convex Polygon Packing program.
- Folder "matlab_figures" contains the PDF and .fig output figures.

6 Examples

6.1 Displacement patch test

This test consists in the solution of the linear elastostatic problem with $\boldsymbol{b} = \boldsymbol{0}$ and essential (Dirichlet) boundary conditions $\boldsymbol{g}$ = [x1 x1 +x2]^T imposed along the entire boundary of a unit square domain. Plane strain condition is assumed with the following material parameters: $E_Y = 1\times10^7$ psi and $\nu = 0.3$. The main file for this problem is provided as the file linear_patch_test_linelast2d.m that is in the folder "test." The polygonal mesh and the VEM results are shown in Fig. 6.1. The relative L2-norm of the error and the relative H1-seminorm of the error obtained for the mesh shown in Fig. 6.1(a) are 2.5493x10-16 and 1.3766x10-15, respectively. Therefore, as predicted by the theory, the VEM solution coincides with the exact solution given by $\boldsymbol{g}$ within machine precision.

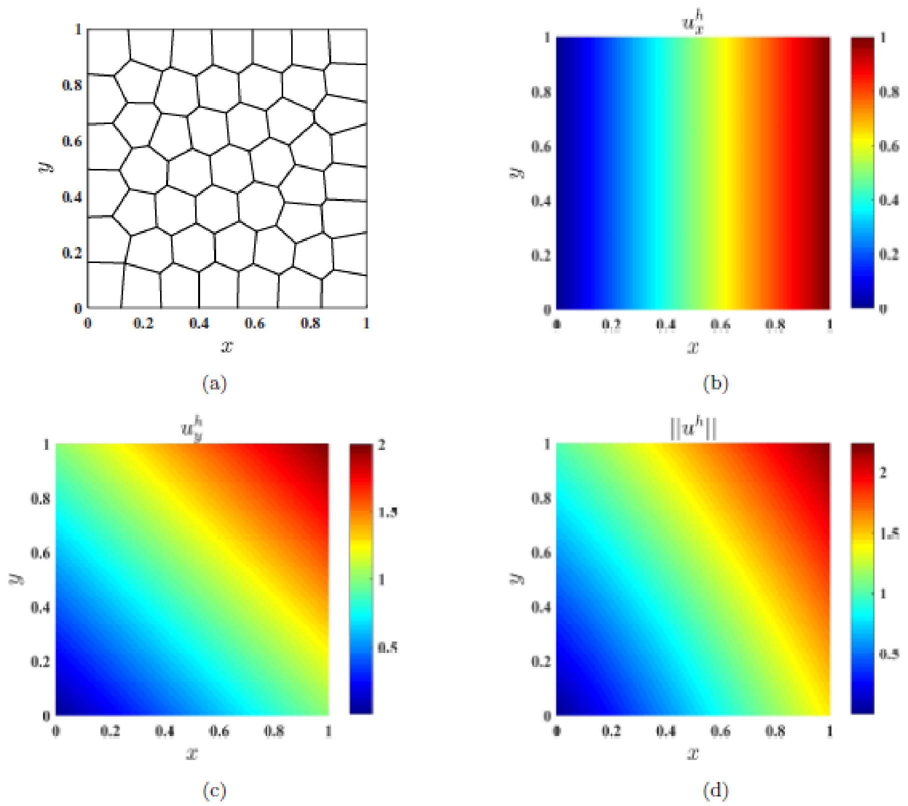


Fig. 6.1: Solution for the displacement patch test using VEMLAB. (a) Polygonal mesh, (b) VEM horizontal displacement, (c) VEM vertical displacement, and (d) norm of the VEM displacement. The relative L2-norm of the error is 2.5493x10-16 and the relative H1-seminorm of the error is 1.3766x10-15.

6.2 Cantilever beam subjected to a parabolic end load

The VEM solution for the displacement field on a cantilever beam of unit thickness subjected to a parabolic end load P is computed using VEMLAB. Fig. 6.2 illustrates the geometry and boundary conditions. Plane strain state is assumed. The essential boundary conditions on the clamped edge are applied according to the analytical solution given by Timoshenko and Goodier:

$$u_x = -\frac{Py}{6\overline{E}_Y I}\left((6L-3x)x + (2+\overline{\nu})y^2 - \frac{3D^2}{2}(1+\overline{\nu})\right),$$
$$u_y = \frac{P}{6\overline{E}_Y I}\left(3\overline{\nu}y^2(L-x) + (3L-x)x^2\right),$$

where $\overline{E}_Y = E_Y/(1-\nu^2)$ with the Young's modulus set to E_Y = 1×10^7 psi, and $\overline{\nu} = \nu/(1-\nu)$ with the Poisson's ratio set to ν = 0.3; L = 8 in. is the length of the beam, D = 4 in. is the height of the beam, and I is the second-area moment of the beam section. The total load on the traction boundary is P = -1000 lbf.

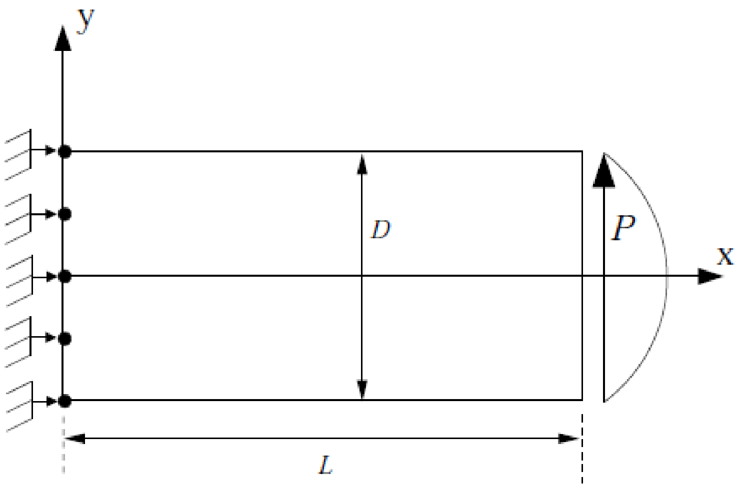
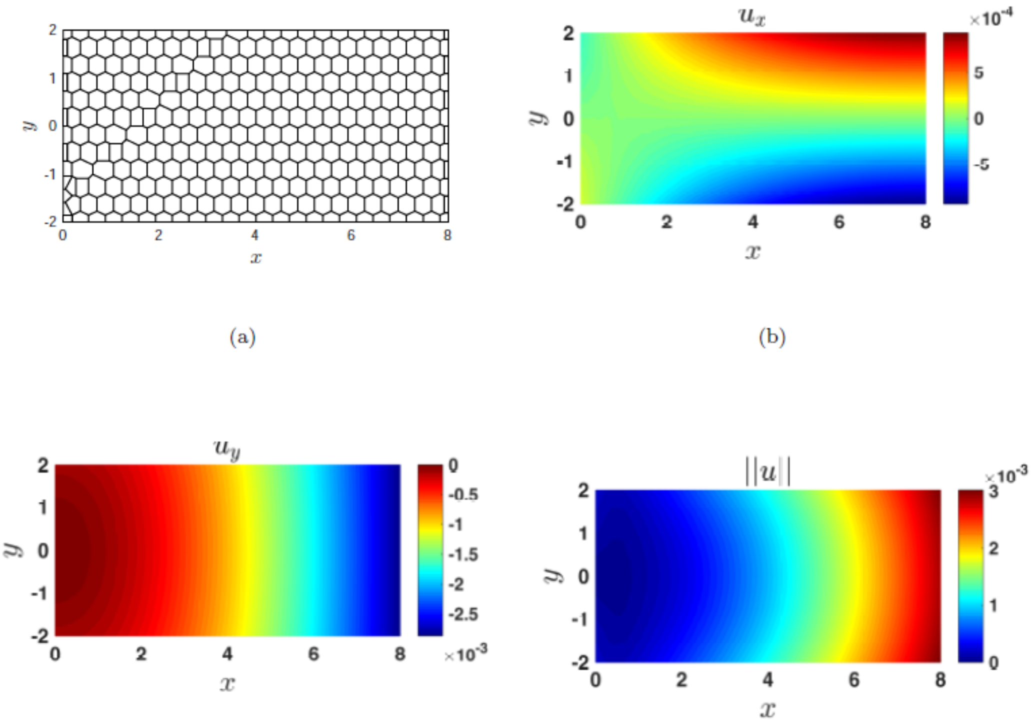


Fig. 6.2: Model geometry and boundary conditions for the cantilever beam problem.

In order to solve this problem in VEMLAB, the function cantilever_beam_linelast2d.m is used. This function is located in the folder "test." The polygonal mesh and the VEM displacements results are shown in Fig. 6.3.

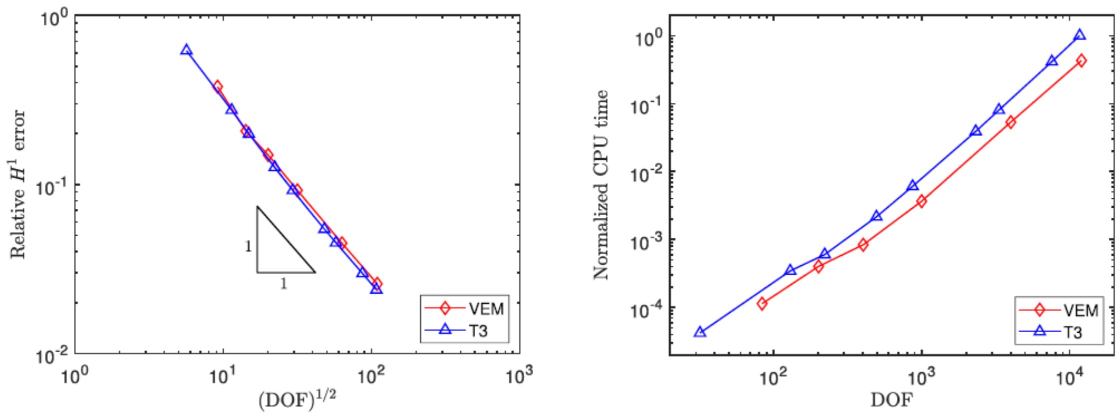


(a) (b)

(c) (d)

Fig. 6.3: Solution for the cantilever beam subjected to a parabolic end load using VEMLAB. (a) Polygonal mesh, (b) VEM horizontal displacement, (c) VEM vertical displacement, (d) norm of the VEM displacement.

A performance comparison between VEM and FEM is conducted. For the FEM simulations, three-node triangles (T3) are used. The performance of the two methods are compared in Fig. 6.4, where the relative H1-seminorm of the error and the normalized CPU time are each plotted as a function of the number of degrees of freedom (DOF). The normalized CPU time is defined as the ratio of the CPU time of a particular model analyzed to the maximum CPU time found for any of the models analyzed. From Fig. 6.4 it is observed that for equal number of degrees of freedom both methods deliver similar accuracy and the computational costs are about the same as the mesh is refined.



(a) (b)

Fig. 6.4: Cantilever beam subjected to a parabolic end load. Performance comparison between the VEM and the FEM (three-node triangles (T3)). (a) Relative H1-seminorn of the error as a function of the number of degrees of freedom and (b) normalized CPU time as a function of the number of degrees of freedom.

6.3 Creating and using a custom meshfile

This example illustrates how to define a customized mesh when the mesh is not generated with the mesh generation functions provided in the source code of VEMLAB.

Consider the cantilever beam subjected to an end load that is shown in Fig. 6.5. The domain is a rectangle of dimensions 27.2 x 17. The material parameters are *E*y = 1e7 and ν = 0.3, and plane strain condition is assumed. The end load is set to -10000. The mesh has been defined manually. The clamped side is defined as the Dirichlet boundary and the loaded end is defined as the Neumann boundary. The custom mesh file starts with the keyword “Custom” and is followed, as usual, by the quantity of nodes (in this case, 44) and their coordinates; then, the number of polygonal elements (in this case 7) and their connectivity. The three final lines of the mesh file correspond, respectively, to the list of the nodes located on the Dirichlet boundary (in this case 2 nodes), the list of nodes located on the Neumman boundary (in this case 2 nodes) and the bounding box xmin, xmax, ymin, ymax, where the domain is contained. The bounding box area can be larger than the domain area. The bounding box is only used for plotting purposes using MATLAB figures --- it is not used for any other type of plot (e.g., GiD and VTK). The complete meshfile is provided in the folder “test/mesh_files” as the file “vem_letters_7poly_elems.txt” and is reproduced below. The main function used to run this problem is the function vem_letters_linelast2d.m that is in the folder “test”.

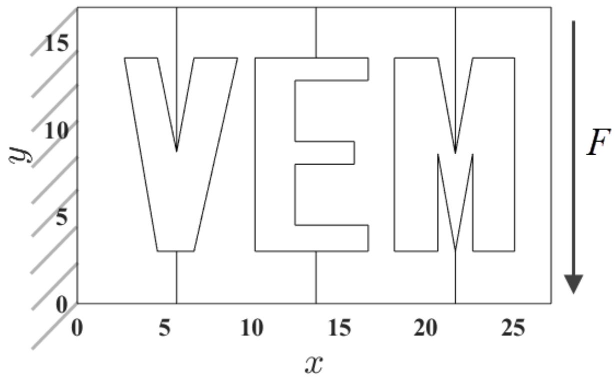


Fig. 6.5: A cantilever beam subjected to an end load and discretized with a customized mesh formed by 7 polygonal elements.

```
# domain type
Custom
# nodal coordinates: number of nodes followed by the coordinates
44
0.00   0.00
5.70   0.00
5.70   3.00
4.60   3.00
2.70   14.10
4.60   14.10
5.70   8.70
5.70   17.00
0.00   17.00
13.70  0.00
13.70  3.00
10.20  3.00
10.20  14.10
13.70  14.10
13.70  17.00
16.70  3.00
16.70  4.50
12.50  4.50
12.50  8.00
15.90  8.00
15.90  9.30
12.50  9.30
```

```
12.50  12.80
16.70  12.80
16.70  14.10
21.70  0.00
21.70  3.00
20.70  8.60
20.70  3.00
18.20  3.00
18.20  14.10
20.70  14.10
21.70  8.60
21.70  17.00
22.70  8.60
22.70  3.00
25.10  3.00
25.10  14.10
22.70  14.10
27.20  0.00
27.20  17.00
6.70   3.00
9.20   14.10
6.70   14.10
# element connectivity: number of elements followed by the connectivity (each line:
nodes_per_element(nel) node1 node 2 ... node_nel)
7
9 1 2 3 4 5 6 7 8 9
8 4 3 42 43 44 7 6 5
13 2 10 11 12 13 14 15 8 7 44 43 42 3
14 12 11 16 17 18 19 20 21 22 23 24 25 14 13
23 10 26 27 28 29 30 31 32 33 34 15 14 25 24 23 22 21 20 19 18 17 16 11
12 30 29 28 27 35 36 37 38 39 33 32 31
11 26 40 41 34 33 39 38 37 36 35 27
# indices of nodes located on the Dirichlet boundary
1 9
# indices of nodes located on the Neumann boundary
40 41
# xmin, xmax, ymin, ymax of the bounding box
0 27.2 0 17
```

7 Plotting of results

Plots can be directly obtained through MATLAB's figures by setting the following parameters in function plot_and_output_options.m that is located in folder "config":

```
create_matlab_contour_plots='yes'; % (for numerical solution)
create_matlab_exact_contour_plots='yes'; % (for exact solution)
```

The figures can also be printed and saved to PDF files (the output PDFs are saved to folder "test/output_files/matlab_figures/") by setting

```
print_figures='yes'; % (for numerical solution)
print_exact_figures='yes'; % (for exact solution)
```

The figures can also be saved to .fig files (the output .fig files are saved to folder "test/output_files/matlab_figures/") by setting

```
save_matlab_figures='yes'; % (for numerical solution)
save_exact_matlab_figures='yes'; % (for exact solution)
```

The resolution of the MATLAB's figures when plotting to PDF files or saving to .fig files is controlled by

```
print_figures_resolution=600;
```

where it has been set to 600 dpi. This value can be changed as required by the user.

Many other customizations of the output figures can be made by setting the appropriate parameters in plot_and_output_options.m (see Section 8 of this primer for details). Several example output MATLAB's figures are provided in Section 10 of this primer.

As an alternative, plots can be visualized externally in the GiD postprocessor. This is an independent process and can be performed even if plots were set to be displayed on MATLAB's figures. When the following is set in plot_and_output_options.m:

```
write_solutions_to_GiD_file='yes';
```

results are written to GiD files and saved to folder "test/output_files/GiD". GiD can be downloaded from its webpage: https://www.gidhome.com/. The procedure to visualize the GiD output files is described next.

The following figures summarize the procedure to visualize the stresses in the post-process module of GiD. The same procedure applies for any of the other variables computed by VEMLAB (e.g., displacements, strains, gradient, fluxes, etc.).

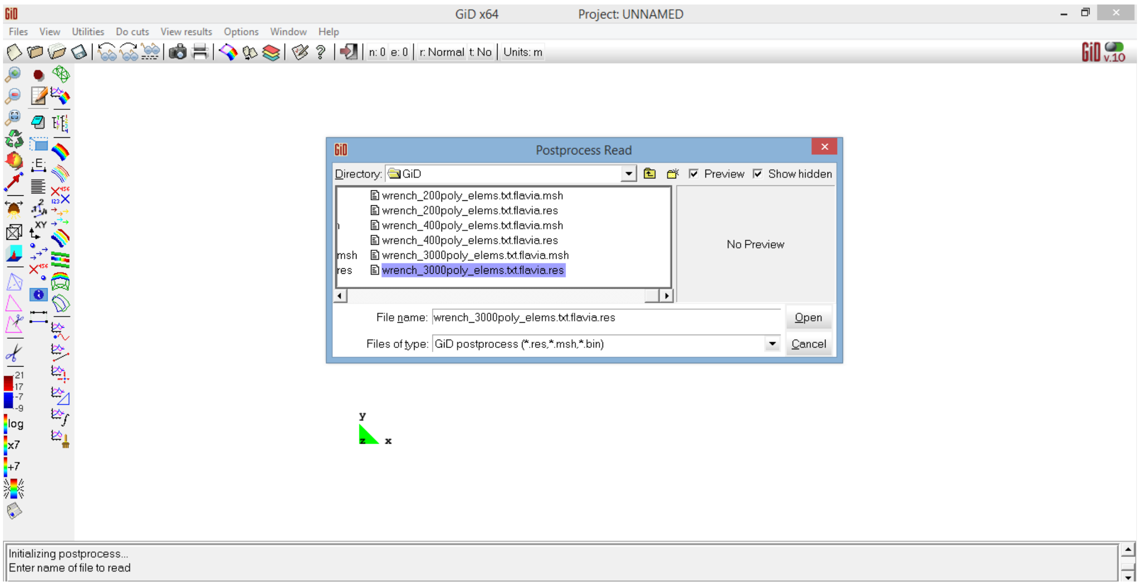


Fig. 7.1: In the postprocess module, open the .res file that was saved to folder "test/output_files/GiD."

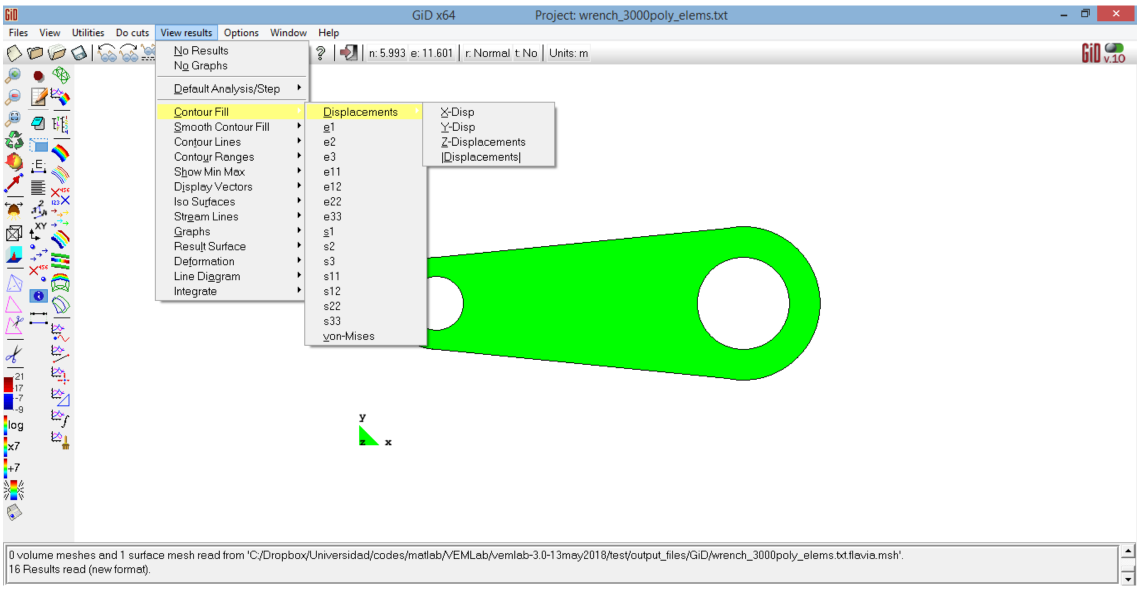


Fig. 7.2: In the postprocess module, click on "View results" to display a menu with various options for plotting results. See for instance, "Contour Fill."

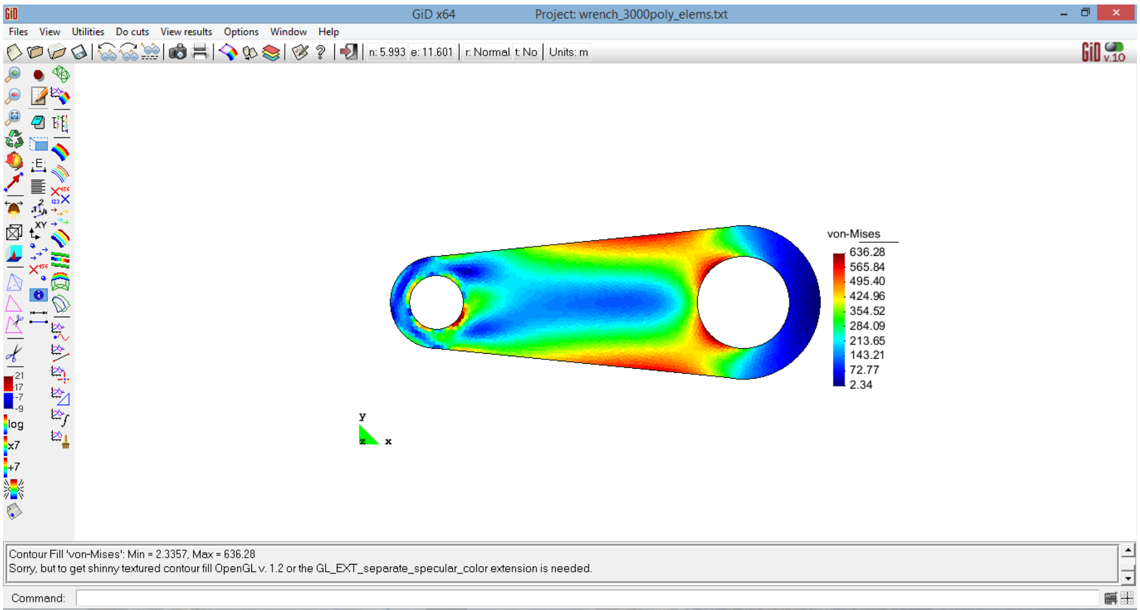


Fig. 7.3: In View results->Contour Fill, click on von-Mises to plot the von Mises stress.

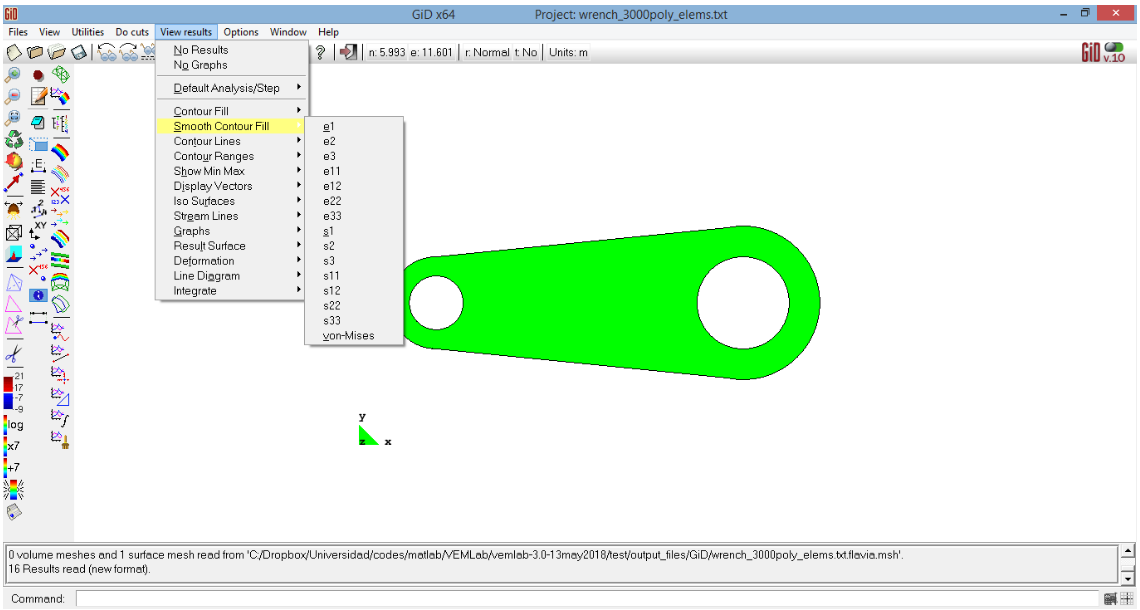


Fig. 7.4: If desired, stresses can be smoothed using the menu View results->Smooth Contour Fill.

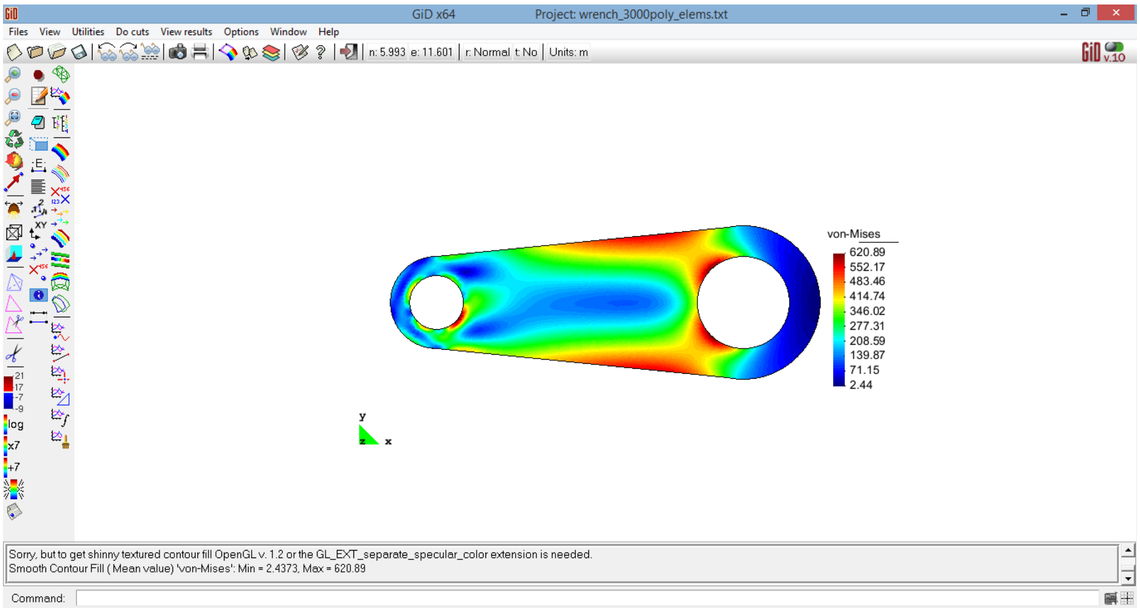


Fig. 7.5: In View results->Smooth Contour Fill, click on von-Mises to plot the smoothed von Mises stress.

Also, GiD can be used to plot deformed shapes as summarized in the following figures.

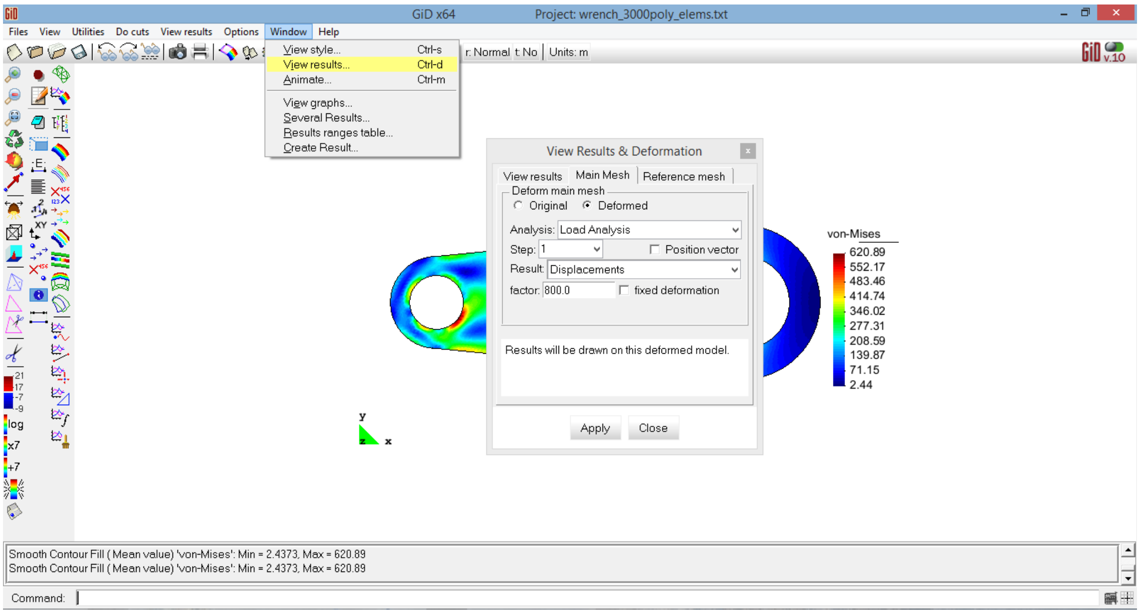


Fig. 7.6: In the postprocess module, click on Window->View results and on the displayed box click on Main Mesh, select Deformed and enter the desired magnification factor in the factor text box.

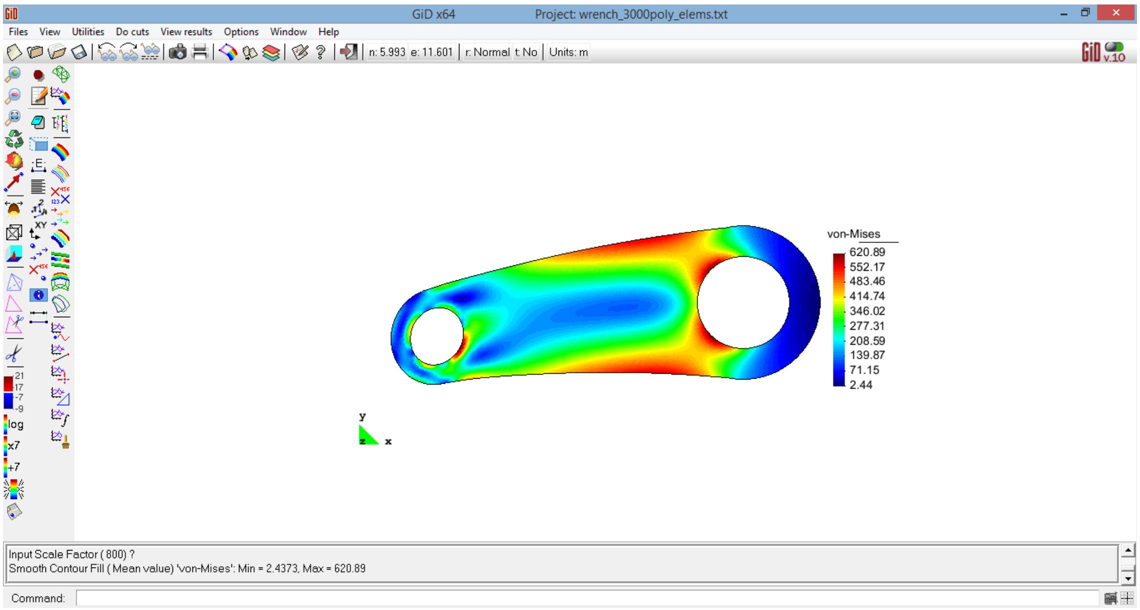


Fig. 7.7: Once the magnification factor is set, click on Apply to display the magnified deformed shape.

8 Setting plot and output options

The plot and output options must be setup by the user. If these are not setup, the program will use the default options. To setup these options go to the file plot_and_output_options.m that is in the folder “config” and activate by setting “yes” or deactivate by setting “no” the available options. Some options include the following (see function plot_and_output_options.m for the complete set of option parameters that are available):

```
%% GENERAL

delete_existing_output_figures='yes';
delete_existing_GiD_output_files='yes';
delete_existing_txt_output_files='yes';
delete_existing_VTK_output_files='yes';

plot_mesh='yes';
plot_mesh_linewidth = 1.0; % size of the lines in the mesh
plot_mesh_nodes = 'yes';   % draw a circle at the nodes
plot_mesh_nodesize = 2.0;  % the size of the circle that represents the node
plot_mesh_axis = 'yes';    % plot the global coordinate system

create_matlab_contour_plots='yes'; % (for numerical solution)
create_matlab_exact_contour_plots='yes'; % (for exact solution)
front_matlab_plot_view_orientation='yes'; % otherwise view is isometric
poo.plot_mesh_over_results='no';
poo.plot_figure_axis='yes';
poo.print_figures='yes'; % (for numerical solution)
poo.print_exact_figures='yes'; % (for exact solution)
poo.save_matlab_figures='no'; % (for numerical solution)
poo.save_exact_matlab_figures='no'; % (for exact solution)
poo.print_figures_resolution=600;

matlab_colormap='RdYlBu';  % 'RdYlBu', 'spectral', 'parula', 'jet', 'hsv', 'hot'
colorbar_tick_label_notation='%.2e';  % '%.2e', '%.2f', '%.4e', '%.4f', etc

write_solutions_to_text_file='yes';
write_solutions_to_GiD_file='yes';
write_solutions_to_VTK_file='yes';
write_solutions_to_CPP_file='yes';
```

```
%% POISSON MODULE

% plotting of main variables to MATLAB/GiD/VTK figures
poisson2d_plot_scalar_field.u='yes';

% plotting of fluxes to MATLAB/GiD figures
poisson2d_plot_flux.qx='no';
poisson2d_plot_flux.qy='no';
poisson2d_plot_flux.qnorm='yes';   % norm of the flux

% plotting of gradients to MATLAB/GiD figures
poisson2d_plot_grad.dx='no';
poisson2d_plot_grad.dy='no';
poisson2d_plot_grad.dnorm='yes';   % norm of the gradient

%% LINEARELASTICITY MODULE

% options for plotting deformed domain to MATLAB figures
linelast2d_plot_deformed_domain='yes';
linelast2d_scale_for_plotting_deformed_domain=1; % a number > 1 will scale the
                                                 % deformed domain when plotting to
                                                 % MATLAB figures

% plotting of main variables to MATLAB/GiD/VTK figures
linelast2d_plot_displacement.ux='yes';
linelast2d_plot_displacement.uy='yes';
linelast2d_plot_displacement.unorm='yes';  % norm of the displacement

% plotting of stresses to MATLAB/GiD figures
linelast2d_plot_stress.s11='no';
linelast2d_plot_stress.s12='no';
linelast2d_plot_stress.s22='no';
linelast2d_plot_stress.s33='no';
linelast2d_plot_stress.s1='no';
linelast2d_plot_stress.s2='no';
linelast2d_plot_stress.s3='no';
linelast2d_plot_stress.vm='no';
linelast2d_plot_stress.p='yes';

% plotting of strains to MATLAB/GiD figures
linelast2d_plot_strain.e11='no';
linelast2d_plot_strain.e12='no';
linelast2d_plot_strain.e22='no';
linelast2d_plot_strain.e33='no';
linelast2d_plot_strain.e1='no';
linelast2d_plot_strain.e2='no';
linelast2d_plot_strain.e3='no';
```

9 Running VEMLAB in Octave

We are grateful to Dr. Stefan Holst, EMAG Application Manager, Technology Group, CD-adapco/Siemens PLM Software, for his kind advice on making VEMLAB to run in Octave.

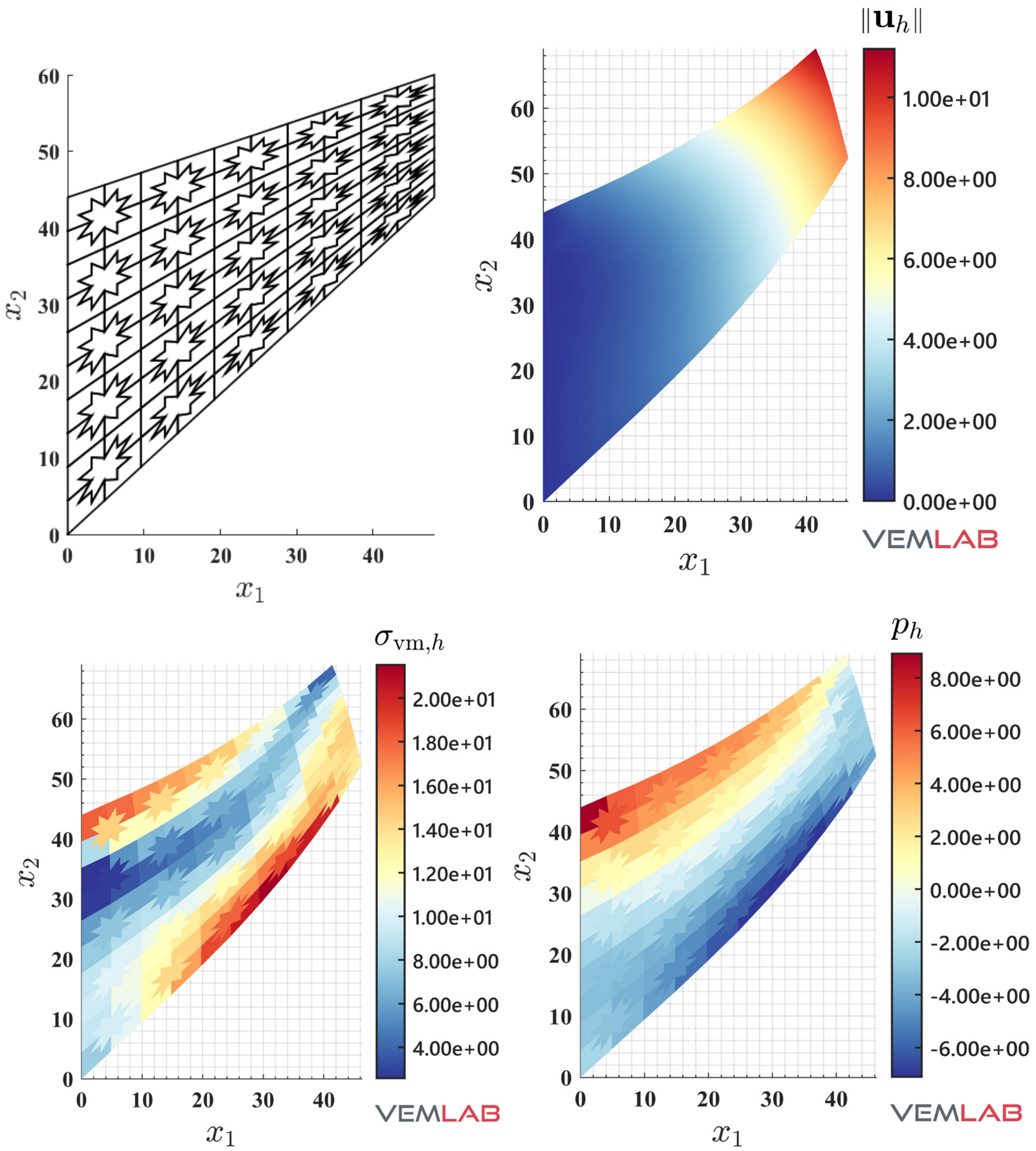
In Octave, the “computer” function that is at the beginning of each test file, returns a different name than one of those expected by VEMLAB when running in MATLAB, i.e., 'PCWIN', 'PCWIN64', 'GLNX86' or 'GLNXA64'. To fix this, at the beginning of each test file simply redefine the variable “opsystem” as follows: opsystem='PCWIN' or opsystem=' GLNX86' if the machine where Octave is installed is a Windows machine or a Linux machine, respectively.

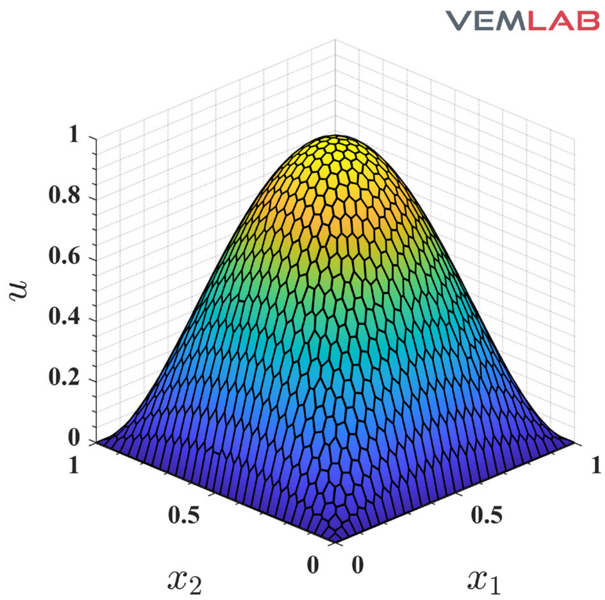
In addition, Octave presents some issues when plotting VEMLAB results to MATLAB figures (on small meshes it will do the work, but on larger meshes it will crash). To fix this, switch off all the MATLAB figures by setting the following parameters in the function plot_and_output_options.m that is in the folder “config”:

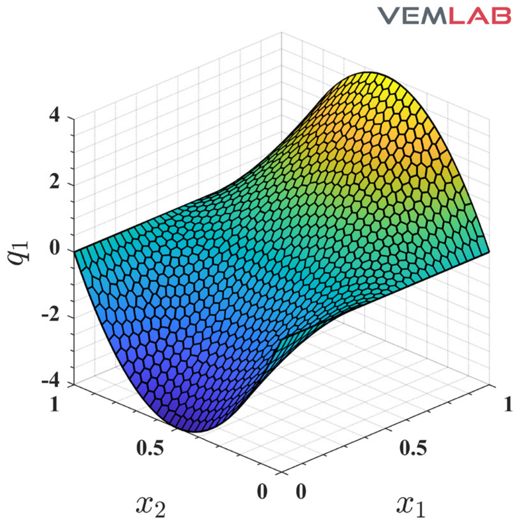
```
create_matlab_contour_plots='no';
plot_mesh='no';
plot_mesh_over_results='no';
write_solutions_to_text_file='yes';
write_solutions_to_GiD_file='yes';
write_solutions_to_VTK_file='yes';
```

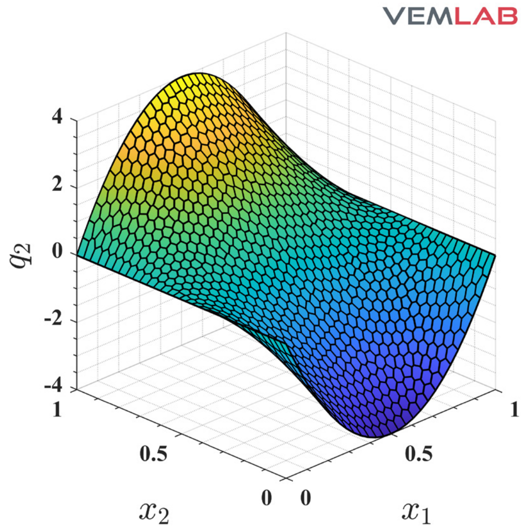
Make sure the last three parameters are set to 'yes' so that one can have access to VEMLAB results through text files or can postprocess results in GiD and VTK/Paraview.

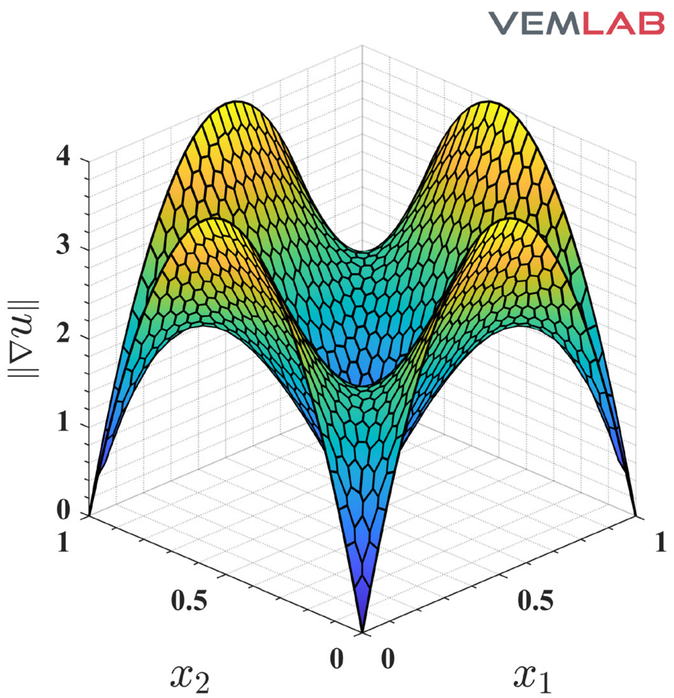
10 Sample MATLAB's output figures

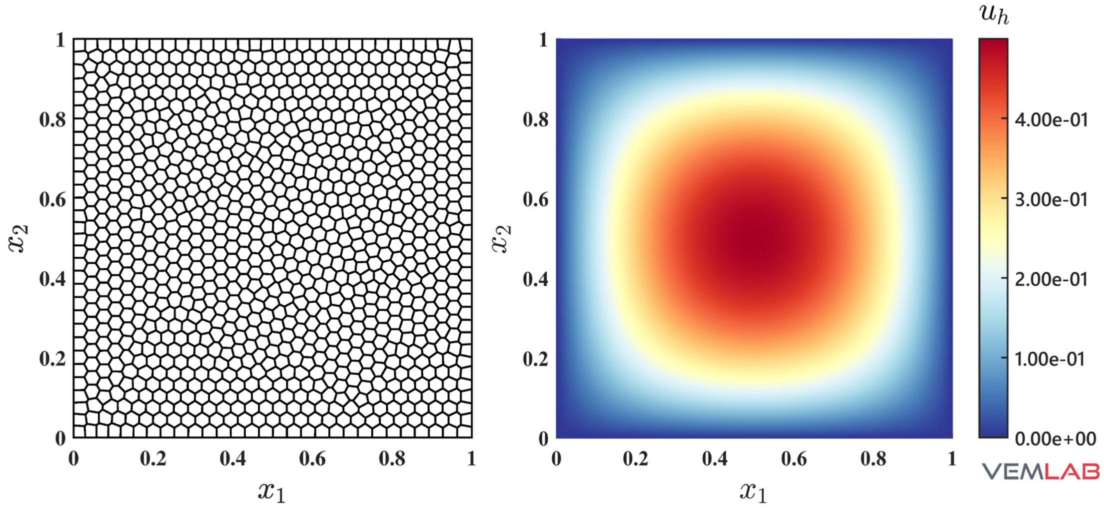



u
x_2
x_1

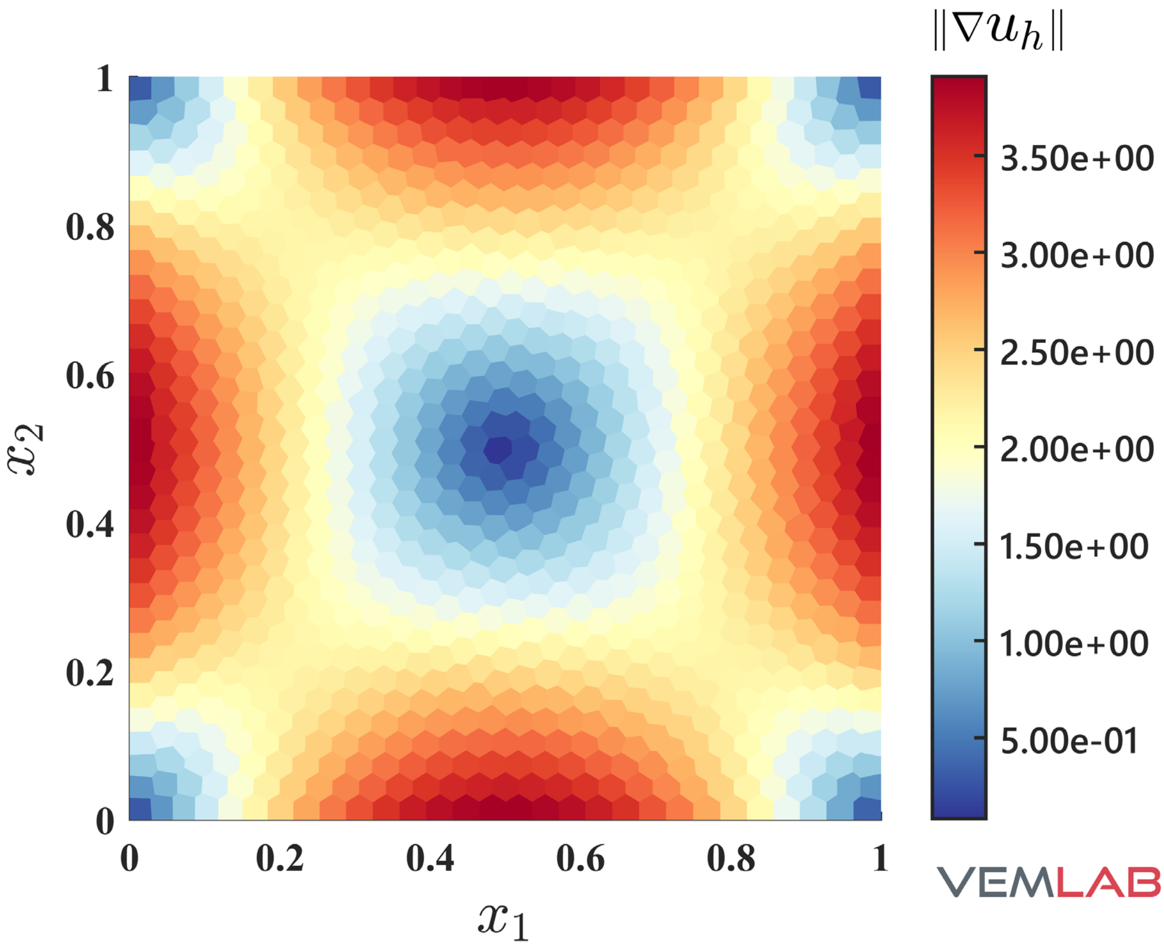

q_1
x_2
x_1


q_2
x_2
x_1

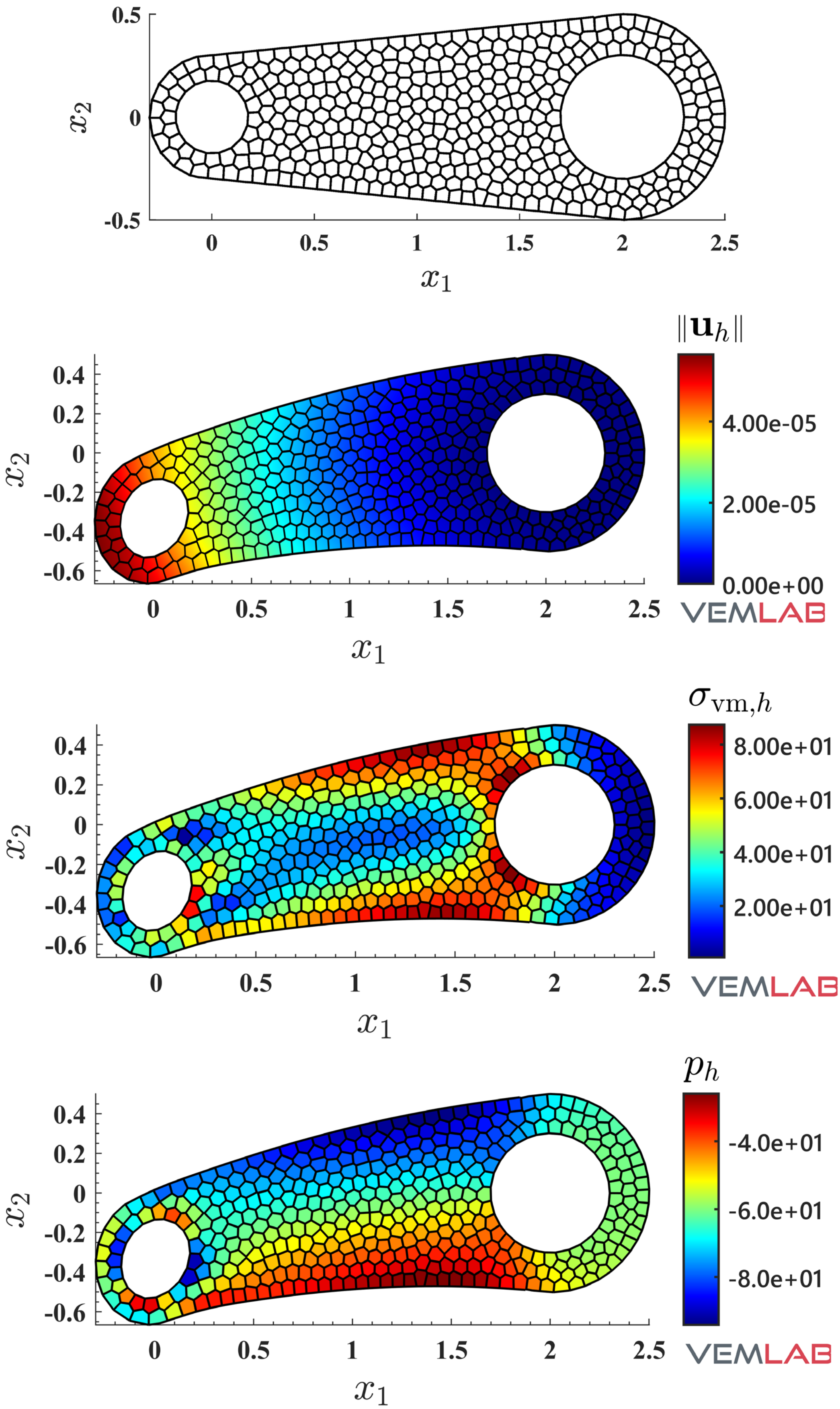

‖∇u‖
x_2
x_1

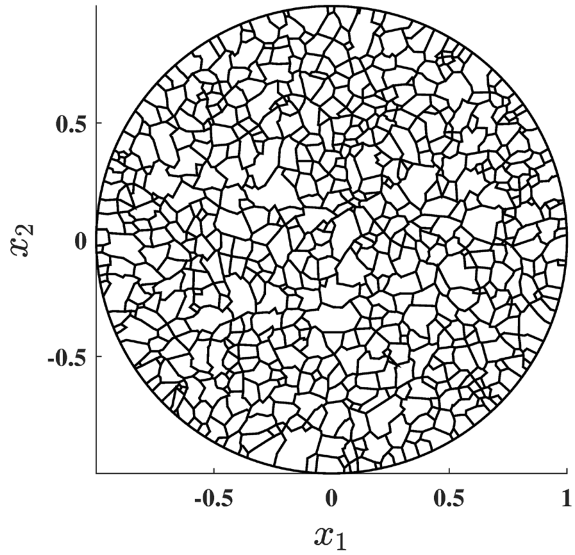


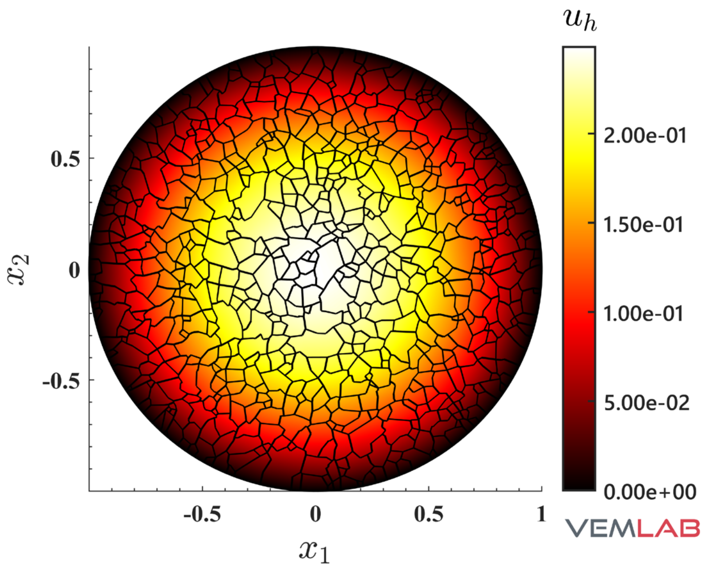
u_h
4.00e-01
3.00e-01
2.00e-01
1.00e-01
0.00e+00
VEMLAB
x_1
x_2

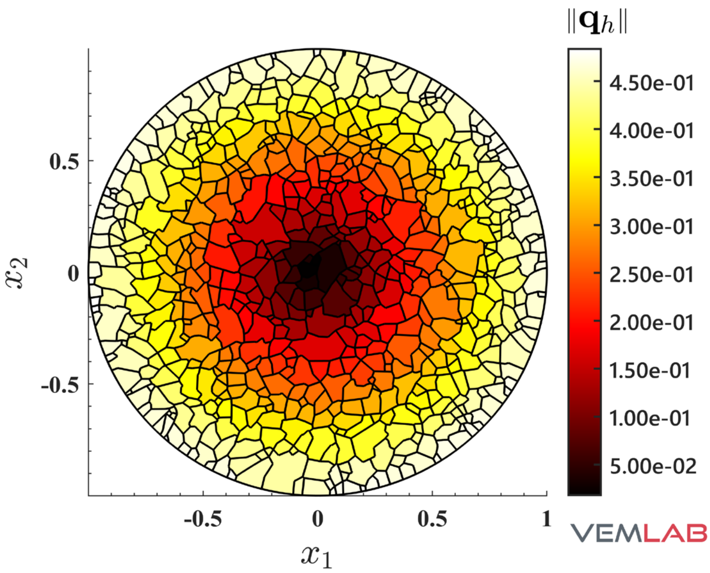


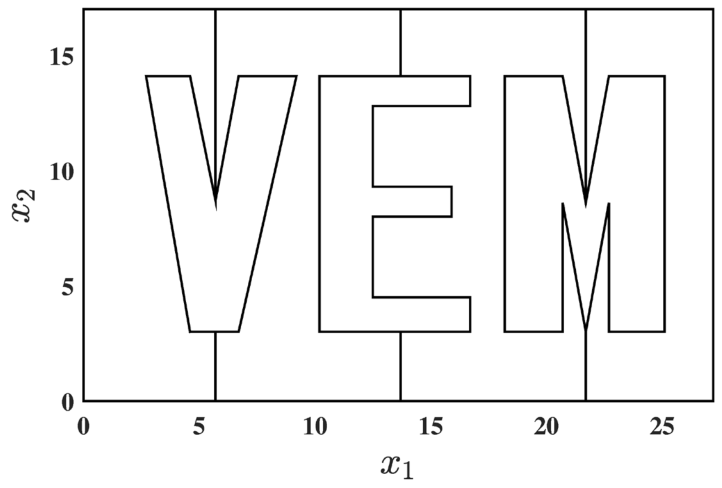
$\|\nabla u_h\|$
3.50e+00
3.00e+00
2.50e+00
2.00e+00
1.50e+00
1.00e+00
5.00e-01
VEMLAB
x_1
x_2

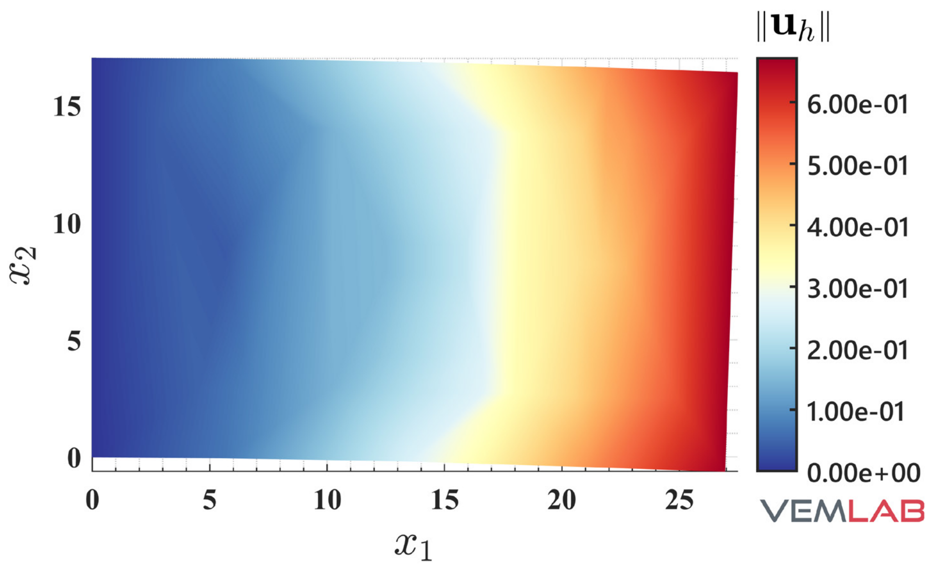

0.5
0
-0.5
0
0.5
1
1.5
2
2.5
x_1
x_2
$\|\mathbf{u}_h\|$
4.00e-05
2.00e-05
0.00e+00
VEMLAB
$\sigma_{\mathrm{vm},h}$
8.00e+01
6.00e+01
4.00e+01
2.00e+01
VEMLAB
p_h
-4.0e+01
-6.0e+01
-8.0e+01
VEMLAB
0.4
0.2
-0.2
-0.4
-0.6

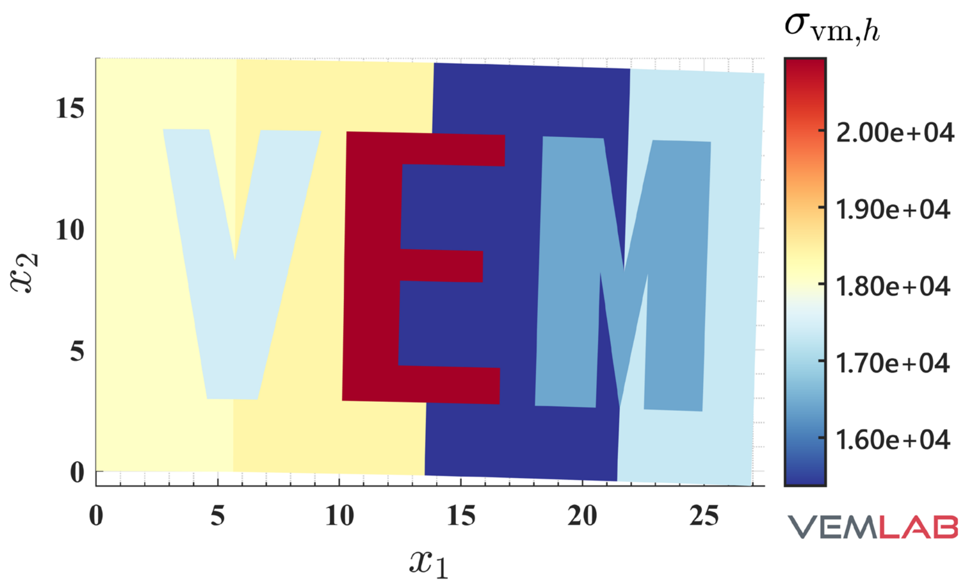

0.5
0
-0.5
x_2
-0.5
0
0.5
1
x_1

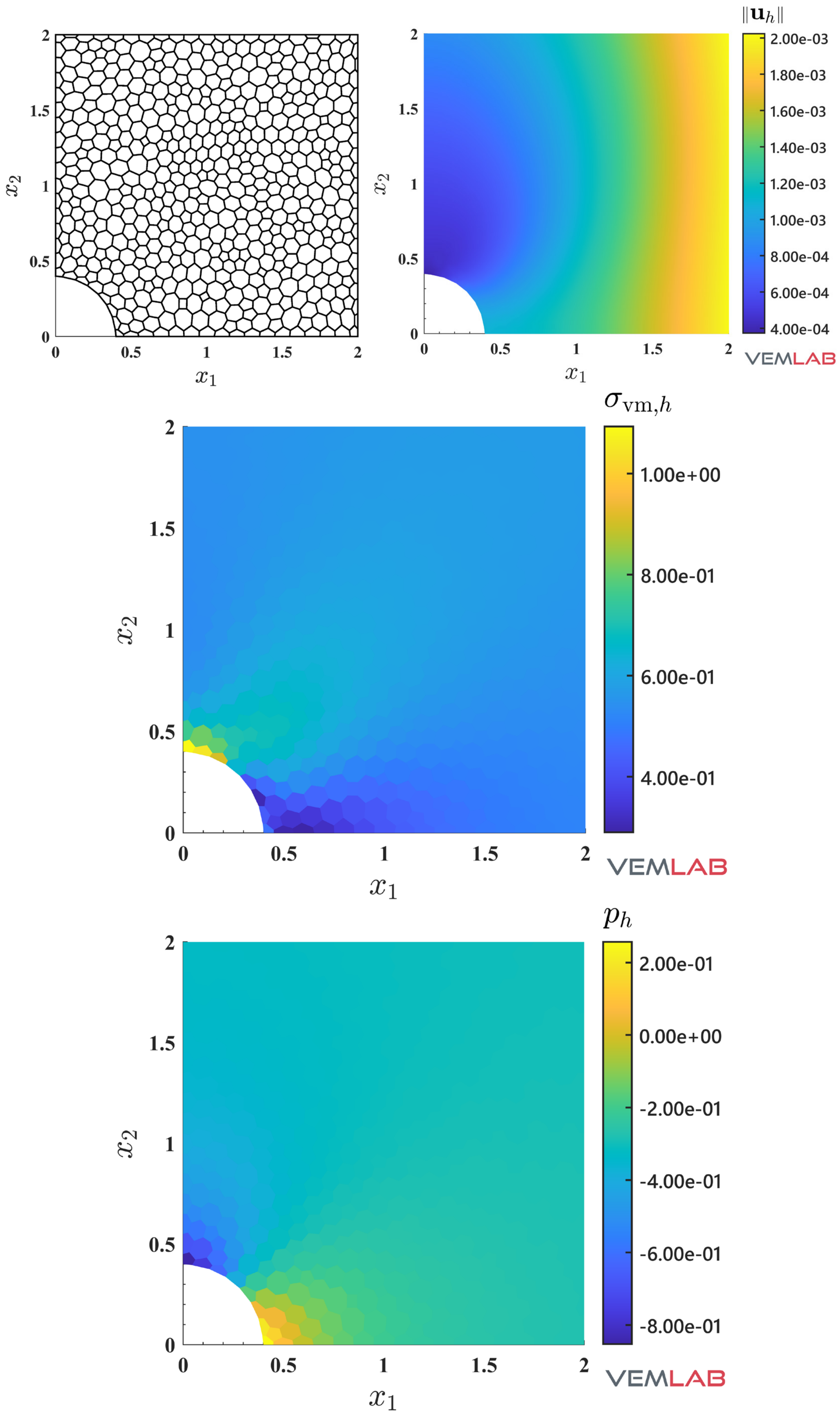

u_h
2.00e-01
1.50e-01
1.00e-01
5.00e-02
0.00e+00
0.5
0
-0.5
x_2
-0.5
0
0.5
1
x_1
VEMLAB

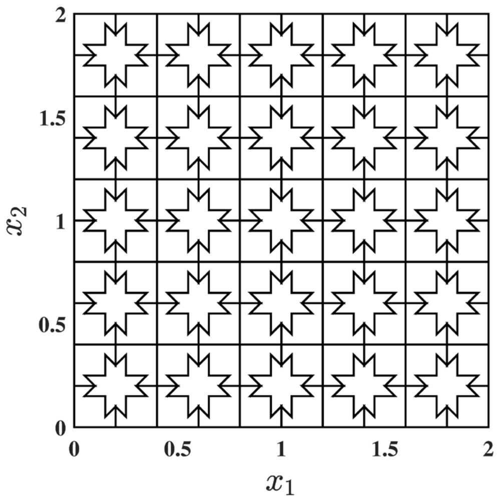

$\|\mathbf{q}_h\|$
4.50e-01
4.00e-01
3.50e-01
3.00e-01
2.50e-01
2.00e-01
1.50e-01
1.00e-01
5.00e-02
0.5
0
-0.5
x_2
-0.5
0
0.5
1
x_1
VEMLAB

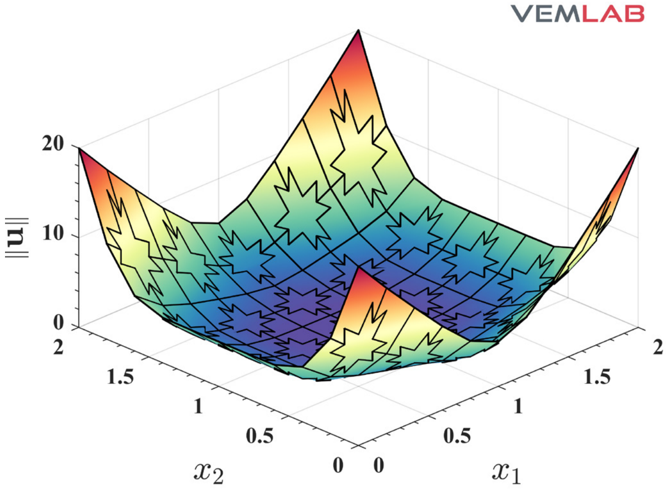

15
10
5
0
0
5
10
15
20
25
x_2
x_1

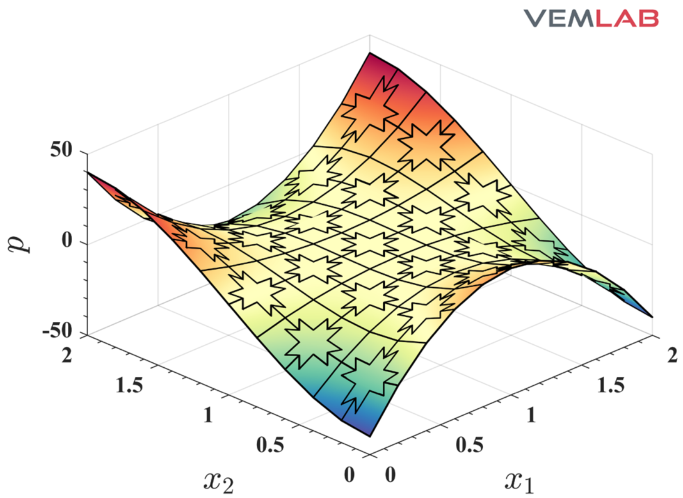

$\|\mathbf{u}_h\|$
6.00e-01
5.00e-01
4.00e-01
3.00e-01
2.00e-01
1.00e-01
0.00e+00
VEMLAB
15
10
5
0
0
5
10
15
20
25
x_2
x_1


$\sigma_{\mathrm{vm},h}$
2.00e+04
1.90e+04
1.80e+04
1.70e+04
1.60e+04
VEMLAB
15
10
5
0
0
5
10
15
20
25
x_2
x_1


$\|\mathbf{u}_h\|$
2.00e-03
1.80e-03
1.60e-03
1.40e-03
1.20e-03
1.00e-03
8.00e-04
6.00e-04
4.00e-04
VEMLAB
$\sigma_{\mathrm{vm},h}$
1.00e+00
8.00e-01
6.00e-01
4.00e-01
VEMLAB
p_h
2.00e-01
0.00e+00
-2.00e-01
-4.00e-01
-6.00e-01
-8.00e-01
VEMLAB
x_1
x_2







11 VEMLAB's website

Check VEMLAB's website for newer versions:

https://camlab.cl/software/vemlab/

--- THE END ---