

VEMLab

A MATLAB library
for the virtual element method

VEMLab Primer

Version 2.0.1

Rev. 0
April, 2018

Copyright and License

VEMLab, Copyright © 2018
by Alejandro Ortiz-Bernardin
<http://camlab.cl/research/software/vemlab/>

CEMCEN - Center for Modern Computational Engineering
Department of Computer Science
Department of Mechanical Engineering
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile
Av. Beauchef 851, Santiago 8370456, Chile



Your use or distribution of VEMLab or any derivative code implies that you agree to this License.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

TABLE OF CONTENTS

1	New and updated feature summary	3
2	Features of VEMLab	3
3	Source code	3
4	Up and running with VEMLab	4
5	Examples	5
5.1	Displacement patch test	5
5.2	Cantilever beam subjected to a parabolic end load	5
6	VEMLab's website	7

1 New and updated feature summary

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 [1], distmesh2d [2], 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; 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, GiD files and VTK files.

3 Source code

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

<http://camlab.cl/research/software/vemlab/>

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

4 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 (with comments) that are provided inside the folder “test” to write your own .m files or modify the ones provided. More details are given below.

In order 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. Some examples of main functions are provided in the folder “test.”

The folder “test/mesh_files” has various text files that contain the information of the predefined meshes that are read from the example main files that are available in the folder “test.” In order 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.

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

The following must be taken into account 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 set 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 stored in the folder “test/output files.” Inside this folder there are three subfolders that contain specific output files, as follows: the folder “GiD” contains output files that are readable in the postprocessor of GiD (<https://www.gidhome.com/>), the folder “VTK” contains output files that are readable in the Visualization Toolkit VTK (<https://www.vtk.org/>) or in the visualization application ParaView (<https://www.paraview.org/>), and the folder “txt” contains the output files in text format.

5 Examples

5.1 Displacement patch test

This test consists in the solution of the linear elastostatic problem with $\mathbf{b} = \mathbf{0}$ and essential (Dirichlet) boundary conditions $\mathbf{g} = [x_1 \ x_1 + x_2]^T$ imposed along the entire boundary of a unit square domain. Plane strain condition is assumed with the following material parameters: $E_Y = 1 \times 10^7$ psi and $\nu = 0.3$. The main file for this problem is provided as the file `linear_patch_test_linelast2d.m` that is located in the folder “test.” The polygonal mesh and the VEM results are shown in Fig. 5.1. The relative L2-norm of the error and the relative H1-seminorm of the error obtained for the mesh shown in Fig. 5.1(a) are 2.5493×10^{-16} and 1.3766×10^{-15} , respectively. Therefore, as predicted by the theory, the VEM solution coincides with the exact solution given by \mathbf{g} within machine precision.

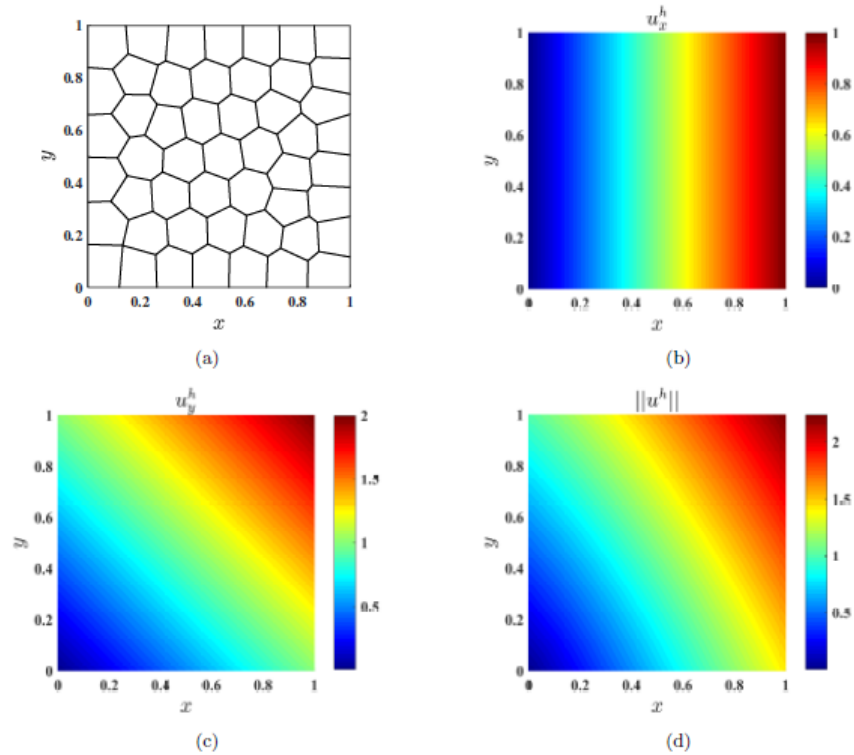


Fig. 5.1: Solution for the displacement patch test using VEMLab. (a) Polygonal mesh, (b) VEM horizontal displacement, (c) VEM vertical displacement, and (d) norm of the VEM displacement. The relative L2-norm of the error is 2.5493×10^{-16} and the relative H1-seminorm of the error is 1.3766×10^{-15} .

5.2 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. 5.2 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:

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

$$u_y = \frac{P}{6E_Y I} (3\nu y^2(L - x) + (3L - x)x^2),$$

where $\bar{E}_Y = E_Y / (1 - \nu^2)$ with the Young's modulus set to $E_Y = 1 \times 10^7$ psi, and $\bar{\nu} = \nu / (1 - \nu)$ with the Poisson's ratio set to $\nu = 0.3$; $L = 8$ 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.

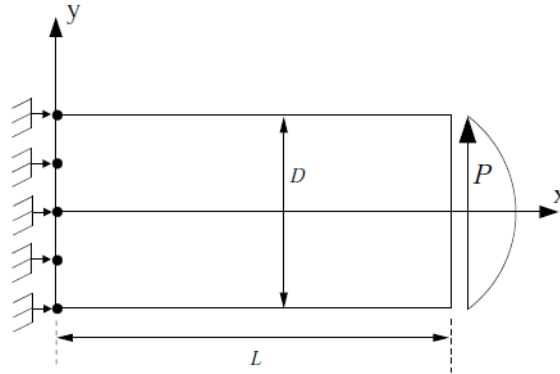


Fig. 5.2: Model geometry and boundary conditions for the cantilever beam problem.

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. 5.3.

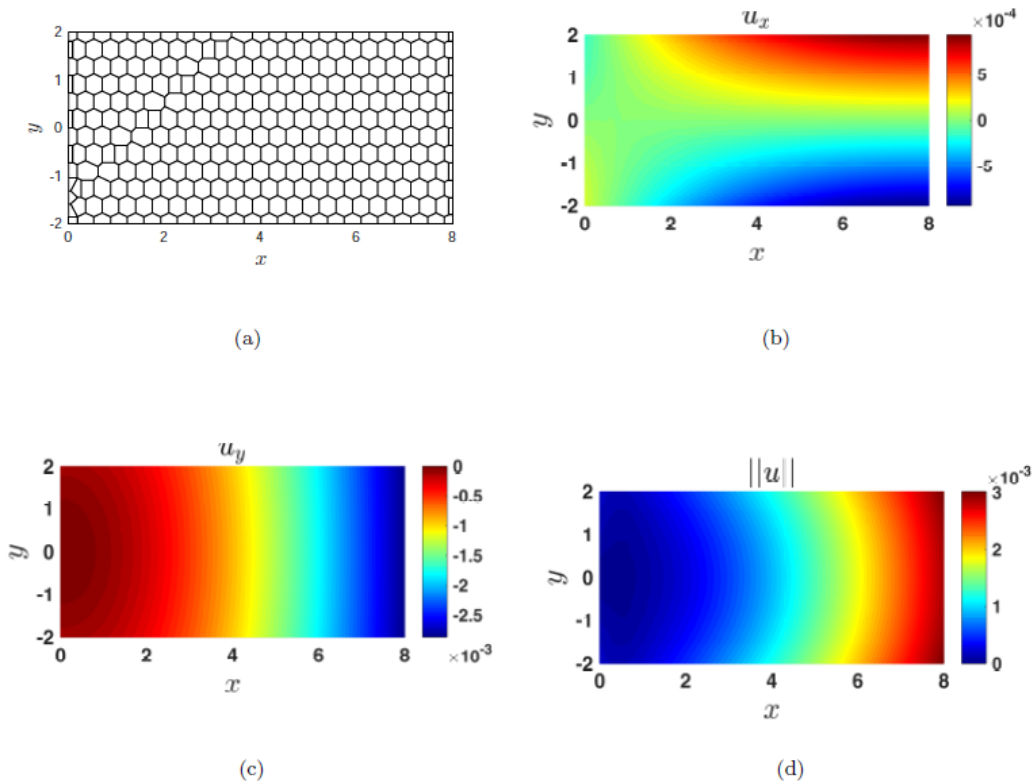


Fig. 5.3: Solution for the cantilever beam subjected to a parabolic end load using VEMLab. (a) Polygonal mesh, (b) VEM horizontal displacement, (c) VEM vertical displacement, (d) norm of the VEM displacement.

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. 5.4, where the relative H^1 -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. 5.4 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.

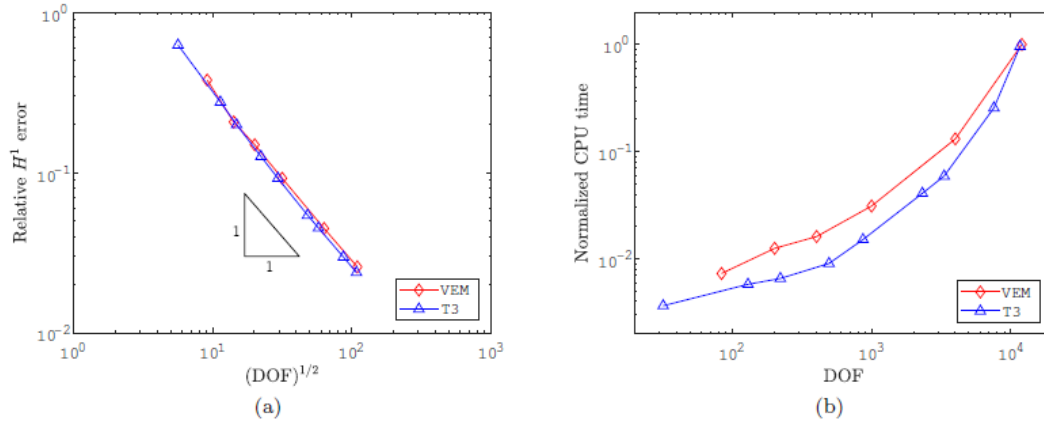


Fig. 5.4: Cantilever beam subjected to a parabolic end load. Performance comparison between the VEM and the FEM (three-node triangles (T3)). (a) Relative H^1 -seminorm of the error as a function of the number of degrees of freedom and (b) normalized CPU time as a function of the number of degrees of freedom.

6 VEMLab's website

Check VEMLab's website for newer versions:

<http://camlab.cl/research/software/vemlab/>

--- THE END ---