Finite Element Methods for the Poisson Equation with Dirichlet Boundary Conditions

Andrew Tomassone

Abstract

Finite Element Methods have a long history of being applied to approximate solutions to differential equations. These methods have solid mathematical theory behind them & yield fast-converging, accurate results.

1. Background

We attempt to solve the Poisson equation for homogeneous Dirichlet boundary conditions:

Given a function F, we want to find u satisfying,

$$-\Delta u = F \text{ in } \Omega$$

$$u = g \text{ on } \partial\Omega$$

Computationally, we cannot solve this problem as it is an infinite-dimensional one. To resolve this, we transform the problem into a finite-dimensional matrix system using the *Finite Element Method.* First, we take our domain, Ω, & break it up into finitely many *elements.* In our case, these elements are equilateral triangles. This new domain is a finite dimensional space we will call V_h, consisting only of piecewise linear functions. V_h can be defined as follows:

$$V_h = \{v_h(x,y) : v_h(x,y) \text{ piecewise plane above each element, continuous on } \Omega,\ \&\ v_h = 0 \text{ on } \partial\Omega\}.$$

The dimension of V_h is denoted by n and corresponds to the number of interior nodes (nodes not on the boundary) in the domain (a node is simply a vertex of a triangle in this case). Thus, the problem we are trying to solve can now be written as

$$\iint_\Omega \nabla u_h \cdot \nabla v_h \, dA = \iint_\Omega f \cdot v_h \, dA \ \ \forall\, v_h \in V_h$$

We then construct basis functions $\{\phi_1, \phi_2, ..., \phi_n\}$ which will be tent functions & define our approximate solution $u_h \in V_h$ to be $\sum_{i=1}^{n} c_i \phi_i$ & solve the matrix system $Ac = b$ where the vector c contains the function values (heights) of our approximation u_h. This matrix system can be expressed as follows:

$$\sum_{i=1}^{n} c_i \iint_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, dA = \iint_{\Omega} F \cdot \phi_j \, dA \;\; \forall \, j = 1, 2, ..., n.$$

2. Some Examples

We will start by looking at a simple polynomial interpreted over the unit square.

$$-\Delta u = f = -2x(x-1) - 2y(y-1)$$

$$u = 0 \text{ on } \partial\Omega$$

The actual solution to this differential equation is,

$$u = x(x-1)y(y-1)$$

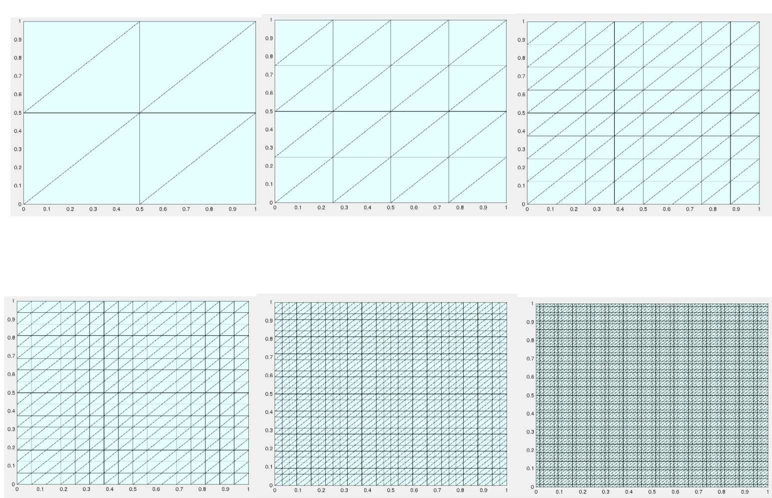

Figure 1: Top level view of the Unit Square meshes

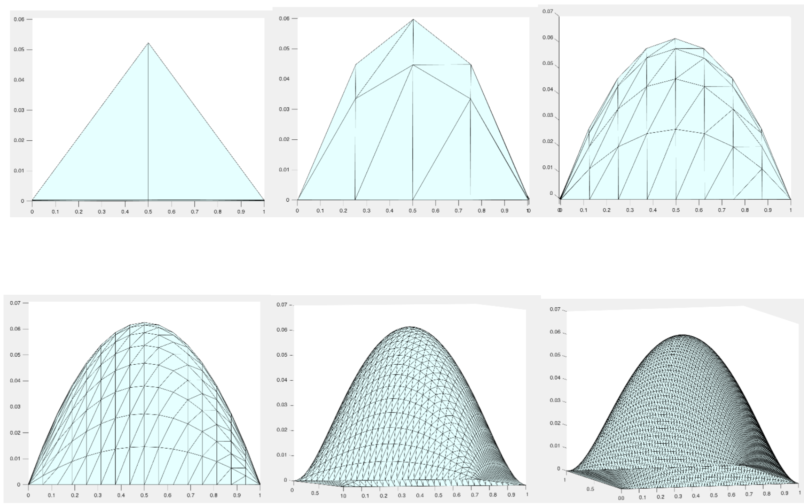

Figure 2: FEM Approximations

Mesh	$\|u - u_h\|$	RoC	$\|\nabla u - \nabla u_h\|$	RoC	$\|v - v_h\|$	RoC	$\|\nabla u - \nabla v_h\|$	RoC
1	1.865792e-02	—	8.051888e-02	—	1.865792e-02	—	8.051888e-02	—
2	5.714893e-03	0.31	4.164236e-02	0.52	5.714893e-03	0.31	4.221450e-02	0.52
3	1.508724e-03	0.26	2.133758e-02	0.51	1.508724e-03	0.26	2.141410e-02	0.51
4	3.824646e-04	0.25	1.073574e-02	0.50	3.824646e-04	0.25	1.074513e-02	0.50
5	9.595118e-05	0.25	5.376319e-03	0.50	9.595118e-05	0.25	5.377509e-03	0.50
6	2.400881e-05	0.25	2.689219e-03	0.50	2.400881e-05	0.25	2.689366e-03	0.50
7	6.003518e-06	0.25	1.344742e-03	0.50	6.003518e-06	0.25	1.344760e-03	0.50

RoC = Rate of Convergence

Now, we will look at a more complicated function over the same domain, still with

$$u = 0 \text{ on } \partial\Omega.$$

The actual solution to this differential equation is,

$$u = x(x-1)y(y-1)(x-0.25)(y-0.25)(y-0.75)$$

Mesh	$\|u - u_h\|$	RoC	$\|\nabla u - \nabla u_h\|$	RoC	$\|v - v_h\|$	RoC	$\|\nabla u - \nabla v_h\|$	RoC
1	1.865792e-02	—	8.051888e-02	—	1.865792e-02	—	8.051888e-02	—
2	3.962325e-04	0.31	2.049413e-03	0.52	3.962325e-04	0.31	2.146028e-03	0.52
3	1.113920e-04	0.26	1.115025e-03	0.51	1.113920e-04	0.26	1.118033e-03	0.51
4	3.002002e-05	0.25	5.832604e-04	0.50	3.002002e-05	0.25	5.836894e-04	0.50
5	7.652551e-06	0.25	2.951166e-04	0.50	7.652551e-06	0.25	2.951669e-04	0.50
6	1.922565e-06	0.25	1.480045e-04	0.50	1.922565e-06	0.25	1.480110e-04	0.50
7	4.812335e-07	0.25	7.405833e-05	0.50	4.812335e-07	0.25	7.405913e-05	0.50

RoC = Rate of Convergence

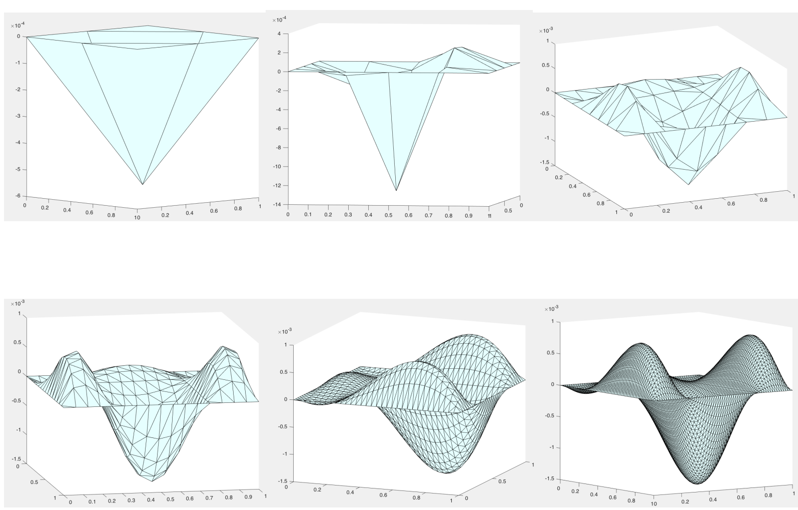

Figure 3: FEM Approximations

Now, we will look at an L-shaped domain with

$$u = g \text{ on } \partial\Omega,$$

where $g(x, y)$ is a function that agrees with u on the boundary. The actual solution to this differential equation is,

$$u = x(x-1)y(y-1)$$

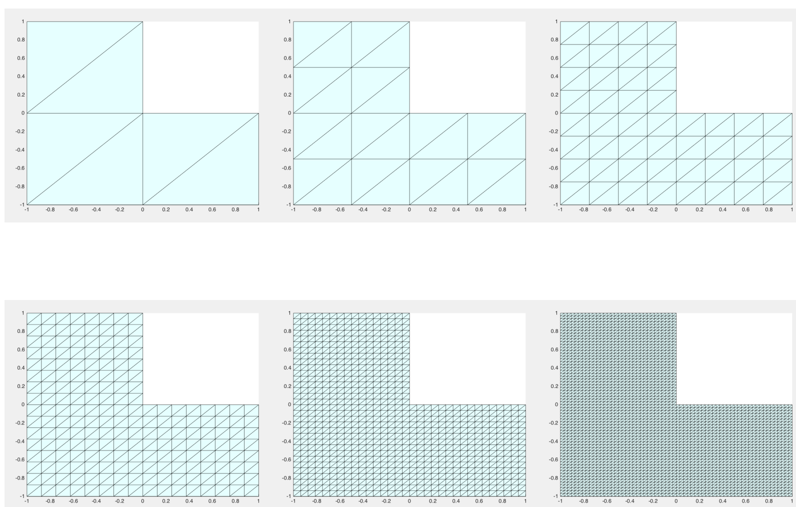

Figure 4: Top level view of the L-meshes

Mesh	$\|u - u_h\|$	RoC	$\|\nabla u - \nabla u_h\|$	RoC	$\|v - v_h\|$	RoC	$\|\nabla u - \nabla v_h\|$	RoC
1	8.369789e-01	—	2.228394	—	8.369789e-01	—	2.287817	—
2	2.123404e-01	0.31	1.168803	0.52	2.123404e-01	0.31	1.171495	0.52
3	5.326746e-02	0.26	5.915275e-01	0.51	5.326746e-02	0.26	5.919690e-01	0.51
4	1.332560e-02	0.25	2.966711e-01	0.50	1.332560e-02	0.25	2.967145e-01	0.50
5	3.331613e-03	0.25	1.484496e-01	0.50	3.331613e-03	0.25	1.484556e-01	0.50
6	8.328834e-04	0.25	7.423911e-02	0.50	8.328834e-04	0.25	7.423982e-02	0.50
7	2.082164e-04	0.25	3.712134e-02	0.50	2.082164e-04	0.25	3.712143e-02	0.50

RoC = Rate of Convergence

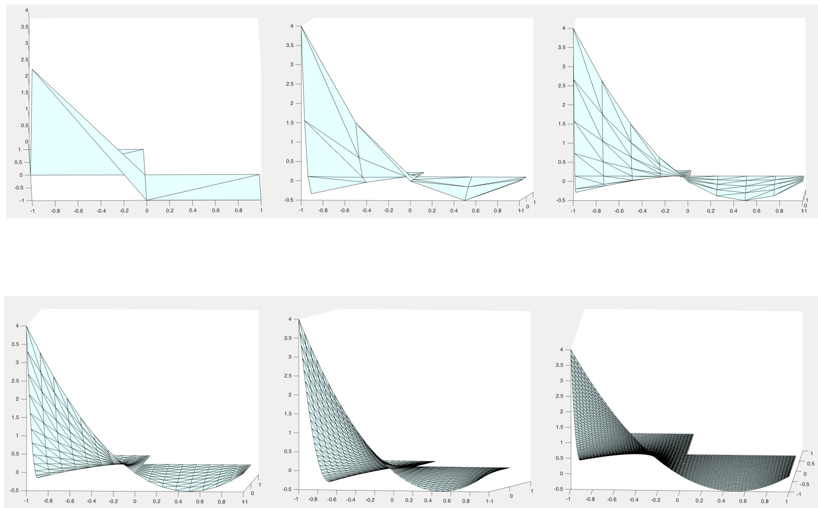

Figure 5: FEM Approximations

Here is another function interpreted over the same L-shaped domain with

$$u = g \text{ on } \partial\Omega,$$

where $g(x, y)$ is a function that agrees with u on the boundary. The actual solution to this differential equation is,

$$u = x(x-1)y(y-1)(x-0.25)(y-0.25)(y-0.75)$$

Mesh	$\|u - u_h\|$	RoC	$\|\nabla u - \nabla u_h\|$	RoC	$\|v - v_h\|$	RoC	$\|\nabla u - \nabla v_h\|$	RoC
1	3.148107	—	7.720276	—	1.865792e-02	—	8.051888e-02	—
2	8.968554e-01	0.31	2.049413e-03	0.52	3.148107	0.31	4.309236	0.52
3	2.350134e-01	0.26	2.234230	0.51	2.350134e-01	0.26	2.234481	0.51
4	5.942484e-02	0.25	1.128674	0.50	5.942484e-02	0.25	1.128706	0.50
5	1.489476e-02	0.25	5.658415e-01	0.50	1.489476e-02	0.25	5.658455e-01	0.50
6	3.726046e-03	0.25	2.831109e-01	0.50	3.726046e-03	0.25	2.831114e-01	0.50
7	9.316660e-04	0.25	1.415793e-01	0.50	9.316660e-04	0.25	1.415794e-01	0.50

RoC = Rate of Convergence

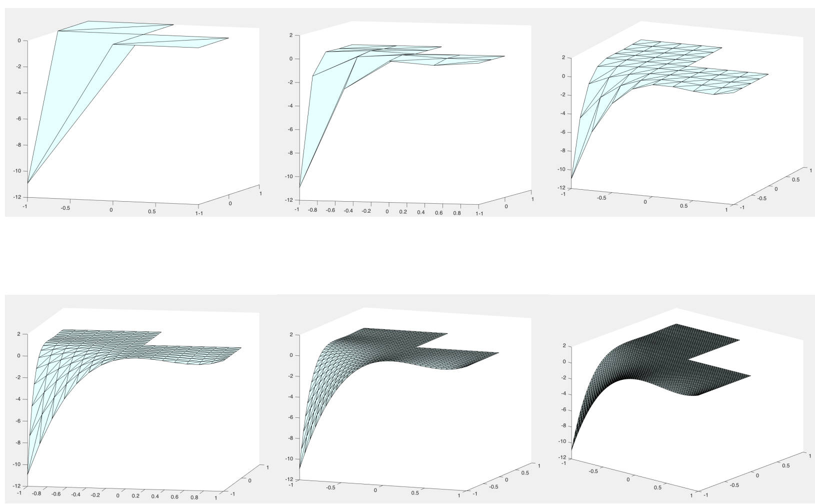

Figure 6: FEM Approximations