CMIS18 - HW4

Isabela Blucher

May 18, 2018

Introduction

Real-world problems that use descriptions of the laws of physics are usually expressed in terms of partial differential equations. For the vast majority of problems, these PDEs usually can't be solved in an analytical form. Instead, approximation methods are constructed to make the discretization of the problem possible, and thus, the solving of it with numerical methods. The solution of these numerical methods are approximations of the solution of the original PDE. The Finite Element Method (FEM) is used to compute these approximations.

In class we derived the equations for a 1-dimensional problem using the FEM. In the next section, we apply the same process to derive formulas for the 2-dimensional case, which will later be implemented in MATLAB.

Formula derivation

For the 2D problem we want to find the unknown function $u(\mathbf{p}) : \mathbb{R}^2 \to \mathbb{R}$ on some domain $\mathbf{p} = [x \quad y]^T \in \Omega \in \mathbb{R}^2$. We are given the PDE

$$\frac{\partial^2 u(\mathbf{p})}{\partial x^2} + \frac{\partial^2 u(\mathbf{p})}{\partial y^2} = c(\mathbf{p})$$

where $c(\mathbf{p}) : \mathbb{R}^2 \to \mathbb{R}$ is some known function and $u(\mathbf{p}) = a(\mathbf{p})$ for all $\mathbf{p} \in \Gamma$, which represents our boundary for this problem. We will do this using the FEM.

Considering that we have elements that are triangles (consist of nodes i, j, k in counter clock-wise order), we want to find an approximation of $u(\mathbf{p}) \approx \hat{u}(\mathbf{p}) = \mathbf{N}^e \hat{u}^e$, and we choose trial functions $v^e(\mathbf{p}) = w^e(\mathbf{p}) = \mathbf{N}^e \delta u^e$. For a triangular element we use barycentric coordinates (area weighted) as shape functions for the elements of $\mathbf{N}^e$. These are linear functions in x and y. We can differentiate them and find $\frac{\partial \mathbf{N}^e}{\partial x}$ and $\frac{\partial \mathbf{N}^e}{\partial y}$, which are constants independent of the coordinates x and y.

Given all of that, we have the integral

$$\int_{\Omega^e} v^e \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - c\right) d\Omega + \int_{\Gamma^e} w^e (u - a)\, d\Gamma = 0$$

Since v and w are complementary by design, we will focus on the domain integral for solving the approximation and for deriving the formulas for K^e and f^e, which can later be easily implemented. The following steps shows the derivation of both formulas.

$$\int_{\Omega^e} v^e \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - c\right) d\Omega = 0$$

$$\int_{\Omega^e} v^e \frac{\partial^2 u}{\partial x^2}\, d\Omega + \int_{\Omega^e} v^e \frac{\partial^2 u}{\partial y^2}\, d\Omega - \int_{\Omega^e} v^e c\, d\Omega = 0$$

Integrating by parts the first two terms of the equation above

$$\left[v^e \frac{\partial u}{\partial x}\right]_{\Omega^e} - \int_{\Omega^e} \frac{\partial v^e}{\partial x} \frac{\partial u}{\partial x}\, d\Omega + \left[v^e \frac{\partial u}{\partial y}\right]_{\Omega^e} - \int_{\Omega^e} \frac{\partial v^e}{\partial y} \frac{\partial u}{\partial y}\, d\Omega - \int_{\Omega^e} v^e c\, d\Omega = 0$$

Cleaning up yields and setting v to zero on Ω^e we get

$$\int_{\Omega^e} \frac{\partial v^e}{\partial x} \frac{\partial u}{\partial x}\, d\Omega + \int_{\Omega^e} \frac{\partial v^e}{\partial y} \frac{\partial u}{\partial y}\, d\Omega + \int_{\Omega^e} v^e c\, d\Omega = 0$$

Applying substitution to u with the approximations we defined previously we get

$$\int_{\Omega^e} \frac{\partial v^e}{\partial x}\frac{\partial N^e}{\partial x}\hat{u}^e\, d\Omega + \int_{\Omega^e} \frac{\partial v^e}{\partial y}\frac{\partial N^e}{\partial y}\hat{u}^e\, d\Omega + \int_{\Omega^e} v^e c\, d\Omega = 0$$

Applying substitution to v^e, we get

$$\int_{\Omega^e} \frac{\partial(N^e\delta u^e)}{\partial x}\frac{\partial N^e}{\partial x}\hat{u}^e\, d\Omega + \int_{\Omega^e} \frac{\partial(N^e\delta u^e)}{\partial y}\frac{\partial N^e}{\partial y}\hat{u}^e\, d\Omega + \int_{\Omega^e} (N^e\delta u^e) c\, d\Omega = 0$$

Since $N^e\delta u^e$ is a scalar, it holds that $N^e\delta u^e = (N^e\delta u^e)^T$

$$\int_{\Omega^e} \frac{\partial(N^e\delta u^e)^T}{\partial x}\frac{\partial N^e}{\partial x}\hat{u}^e\, d\Omega + \int_{\Omega^e} \frac{\partial(N^e\delta u^e)^T}{\partial y}\frac{\partial N^e}{\partial y}\hat{u}^e\, d\Omega + \int_{\Omega^e} (N^e\delta u^e)^T c\, d\Omega = 0$$

Now, if we separate $(\delta u^e)^T$ from the integrals and see that the equation must hold for arbitrary values of δu, we get the terms for K (multiplied by $\hat{u}^e$) and $-f$

$$\left(\int_{\Omega^e} \frac{\partial(N^e)^T}{\partial x}\frac{\partial N^e}{\partial x}\, d\Omega + \int_{\Omega^e} \frac{\partial(N^e)^T}{\partial y}\frac{\partial N^e}{\partial y}\, d\Omega\right)\hat{u}^e + \int_{\Omega^e} (N^e)^T c\, d\Omega = 0$$

With that, we have derived a formula for $f^e = \int_{\Omega^e} (N^e)^T c\, d\Omega$. To find K^e, we need to see that since the partial derivatives of N^e are not dependent on variables they can be taken out of the integral, leaving us with

$$K^e = \left(\frac{\partial(N^e)^T}{\partial x}\frac{\partial N^e}{\partial x}\int_{\Omega^e} d\Omega + \frac{\partial(N^e)^T}{\partial y}\frac{\partial N^e}{\partial y}\int_{\Omega^e} d\Omega\right)$$

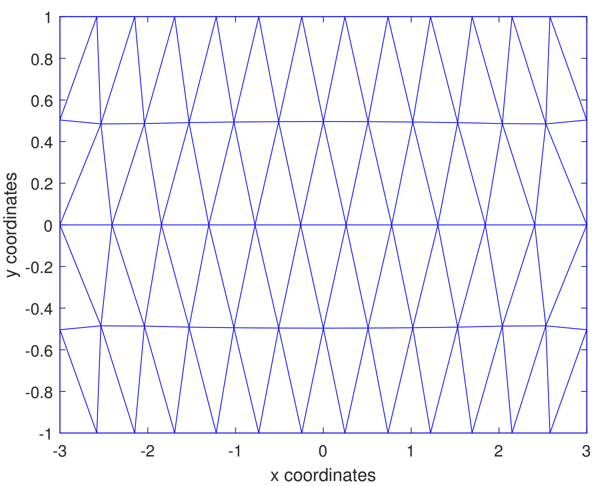
We can see that the integral $\int_{\Omega^e} d\Omega$ represents the area of element e in the domain Ω, which we will represent with the variable A^e. So the final derived formula for K^e is

$$K^e = \left(\frac{\partial(N^e)^T}{\partial x}\frac{\partial N^e}{\partial x} + \frac{\partial(N^e)^T}{\partial y}\frac{\partial N^e}{\partial y}\right) A^e$$

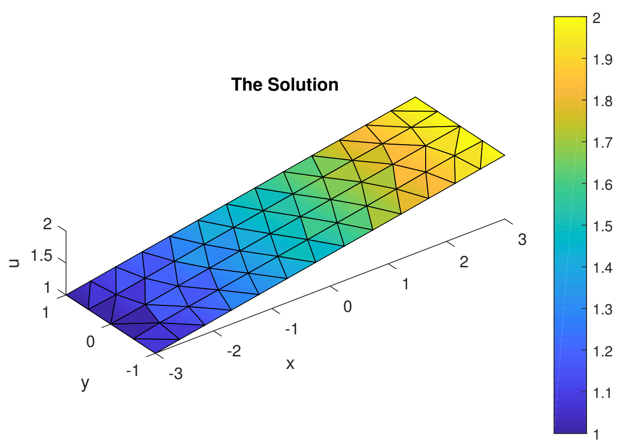
Experiments

After having derived the formulas for K^e and f^e, it becomes very easy to implement a script to assemble the matrices and find an approximate solution to our original PDE. The example being used for the experiments below is the one given the slides. This means that we are using a rectangular domain that is 6×2 in size, with $a = 1, b = 2, c = 0$, and boundaries treated as point-wise conditions.

The triangular mesh we have been given, and the mesh on top of the solution obtained from computing $K\hat{u} = f$, are illustrated in the following plots in Figure 1.



(a) Triangular mesh

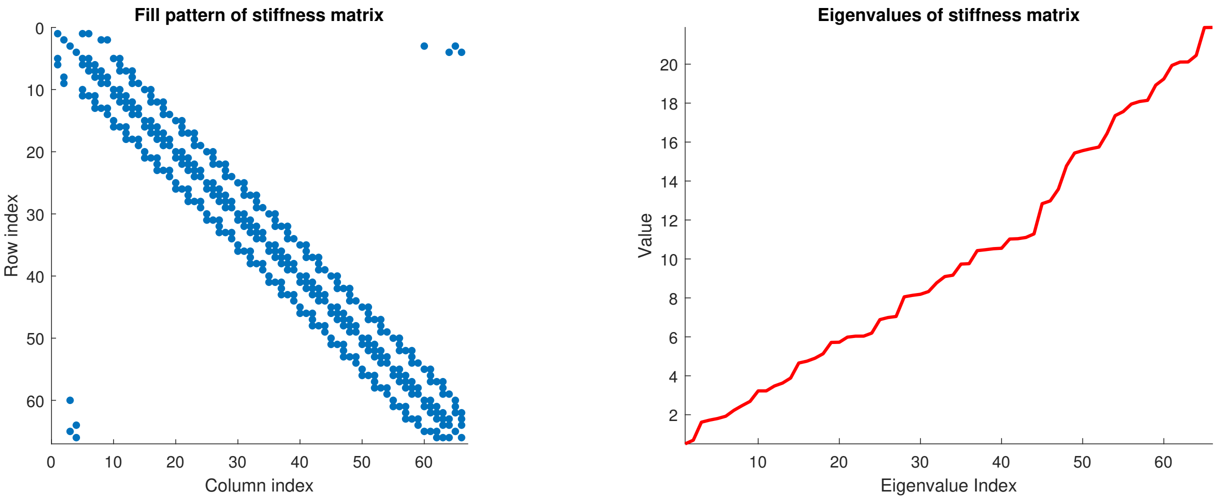


(b) Solution to of approximation

Figure 1: Mesh and solution for the original problem given on the slides using FEM

As expected, since our know function c is set to zero, we have no changes in our slope, and thus have a straight line from one boundary to the other.

We can also inspect the fill pattern of the K matrix and see how its eigenvalues behave.



(a) Fill pattern of stiffness matrix

(b) Eigenvalues of stiffness matrix

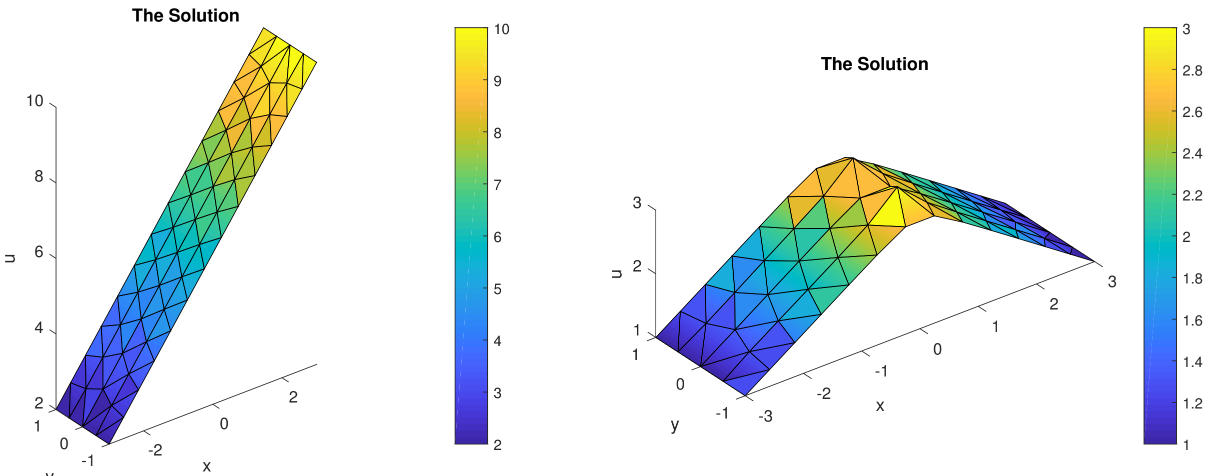
Figure 2: Behavior of stiffness matrix for the mesh and problem given on the slides

From the plots in Figure 2, we can visually inspect the fill pattern of the matrix and observe that it is symmetric, which means we expect it to have all real eigenvalues. This can be confirmed by looking at the eigenvalues plot and how they are increasing in value the more the index increases.

Changing parameter values

One thing that is expected to change is the solution if we introduce different values for the boundaries or intermediary points between them. The following discussion will follow changes to the values of a, b and the introduction of more points in the boundaries and then the subsequent plotting these changes.

Figure 3 shows the solutions with the same mesh and different values for a and b. The values set were $a = 2$ and $b = 10$.



(a) Solution with a = 2 and b = 10

(b) Solution with middle edge introduced

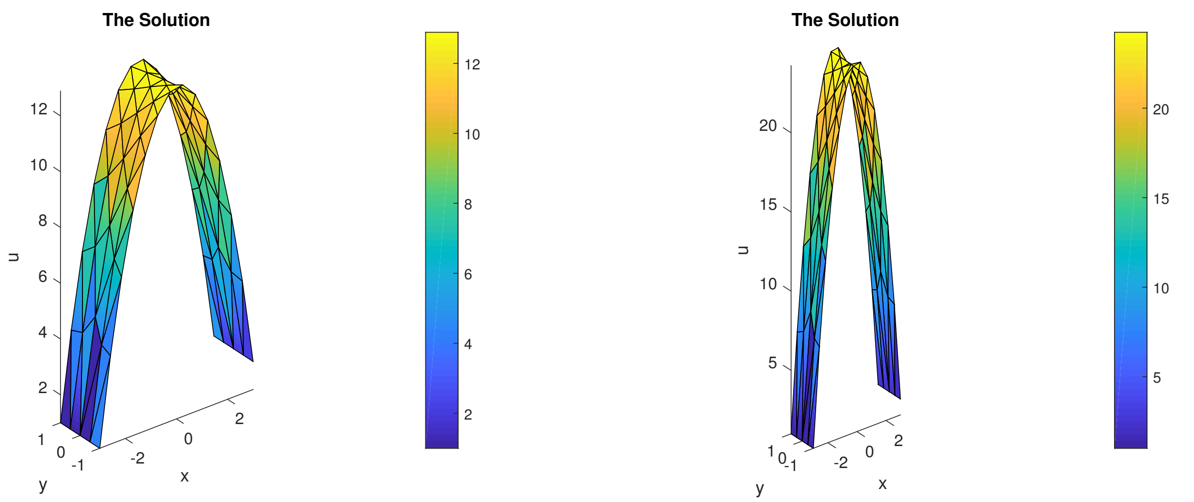
Figure 3: Solution behavior for different parameter values

As we can see on the first solution plot, there still are no changes to the slope, but the line is a lot steeper given the new boundary values. As for the second picture, a middle edge was introduced so that now we have three parameter values. We set the outer edges to 1 and the middle edge to 3. With that we can see that our solution has the shape of a ”tent”, with a peak in the middle.

Changing source term

The source term in a PDE describing some sort of wave equation describes how every coordinate of our mesh will change over time. By changing its initialization, we should be able to see differences in the solution's behavior.

For the plots in Figure 4, instead of initializing f with all zeros, we set it to all 0.5s or all ones. The values of parameters a, b are still respectively 1 and 2. We can see that the shape changes, even with constant source terms.



(a) Solution with f initialized as 0.5

(b) Solution with f initialized as 1

Figure 4: Solution behavior for different source term initializations

From visual inspection we can see that we get a curved shape for both solutions, with the second reaching bigger values of u.

Comparison between 1D and 2D cases

Both the 1-dimensional and 2-dimensional cases for the FEM follow very similar steps of derivation for the formulas and implementation. The differences between both are in the details.

For example, while the 1-dimensional element was the space between two nodes, the 2-dimensional element is a triangle, which changes how elements are implemented, since know we have to keep track of not only node positions in the mesh, but also the index of its vertices.

Both cases make use of trial and shape functions. For the 1-dimensional shape function we use a hat-function and for the 2-dimensional case we use the triangle's barycentric coordinates (area weighted coordinates).

Both cases need integrating to achieve the solution, for the 1-dimensional case it is possible to integrate on the whole domain from x_1 to x_n. For the 2-dimensional case, element-wise integrals are used, which means that we are integrating for each element of the domain.

Besides that, the assembly of elements into the matrix K and dealing with the boundary conditions are very similar to one another, and almost no changes were necessary to implement.

Conclusion

In conclusion, the FEM is important so we can deal with the physical world's governing equations that are PDEs with no analytical solution. It is useful to break down very big and untreatable domains into smaller elements that can be dealt with separately and then be put together again. This method can handle a wide variety of complex geometries and restraints and can be applied to a wide range of real-world engineering problems.