Finite Element Method in 1D BVP

Class Project for fulfilling the requirements of MATH 549: Scientific Computing

Fahim Alam

January 31, 2020

Abstract

Finite Element Method (FEM) is a powerful numerical approach,used to approximate the solutions of many intricate problems using an array of mathematical techniques (opposed to Finite difference method which approximates the actual differential equation directly).The name comes from the fact that the method subdivides a larger problem into simpler parts which are called Finite Elements. The equations that model these finite elements are solved and assembled back into the larger system of equations which model the entire problem. In this project, we will try to solve boundary vale problems with single variable using this method.

Introduction

We start by assuming a differential equation which has the form,

$$A(u(x)) = f(x)$$

where, $A(u(x))$ is some differential equation and $f(x)$ is some given function. We also can form an approximation to the solution of the DE by

$$u_N = \sum_{j=1}^{N} \phi_j(x) b_j$$

where,b_j's are needed to be found out and ϕ_j's are some approximation functions. Substituting this to the given DE, we can get the residual,

$$r_N = A(u_N(x)) - f(x) \neq 0$$

Now our aim is to determine the coefficients b_j which will make r_N the minimum over the domain of the solution. That's why we want,

$$\int_\Omega r_N w_i(x) dx = 0$$

where, $w_i(x), i = 1, \cdots, N$ is a test (=Weight) function and also a set of linearly independent functions. This method is called Weighted residual method. Choosing $w_i(x) = \phi_i(x)$, we get Galerkin Approximation method and choosing $w_i(x) = \delta(x - x_i)$, we get collocation method. Here, x_i is i-th collocation point of the domain and δ is dirac-delta function which is defined such that its value is zero for all nonzero values of its arguments.

$$\delta(x-x_0)=0, \text{ when } x\neq x_0$$
$$\int_\Omega \delta(x-x_0)f(x)dx=f(x_0)$$

Between these two methods, one might choose to use Galerkin's method while solving finite element method problems. The reason behind is that Finite element method is based on Galerkin's method.

Finite Element method

Now we will try to solve the following problem:

$$-p(x)u''(x)+q(x)u(x)=f(x), x\in(0,1)$$
$$u(0)=0, u(1)=0$$

Given that $f\in L^2(0,1), p(x), q(x)\in L^\infty(0,1)$ such that $p(x), q(x)\geq p_0, q_0\geq 0$ almost everywhere in $(0,1)$.

Does this problem have a unique solution? To answer that, we assume two solutions $u_1, u_2\in H_0^1$ that satisfy our current BVP, that means we can write,

$$a(v,u_1)=(v,f)$$
$$a(v,u_2)=(v,f),\ \forall v\in H_0^1$$
$$\text{Subtraction provides }\ a(v,u_1-u_2)=0\ \forall v\in H_0^1$$

Clearly we need to pick v in such a manner that $a(u_1-u_2,u_1-u_2)=0$. If $q(x)$ is positive (on the given interval) then $a(u_1-u_2,u_1-u_2)>0$ except at $u_1=u_2$. Now what if $q(x)\equiv 0$,then $a(u_1-u_2,u_1-u_2)=0$ when $u_1'-u_2'=0$. That makes u_1-u_2 equal to some constant, which is impossible since both of them satisfy boundary conditions so u_1 must be equal to u_2.Hence we have an unique solution for the problem.
Basic steps of FEM are illustrated (from a general perspective) in the following subsections using the Galerkin version of the weighted residual method.

Strong Solution

$$-p(x)u''(x)+q(x)u(x)=f(x), x\in(0,1)$$
$$u(0)=0, u(1)=0$$

Weak form

We obtain the weak form we integrate by parts the product of the strong form of the equation multiplied by a test function, v which is infinitely smooth with compact support i.e. $v \in C_c^\infty(0,1)$ with an additional condition that $v(0) = v(1) = 0$,

$$\int_0^1 \Big(- p(x)u''(x) + q(x)u(x)\Big) v(x)dx = \int_0^1 f(x)v(x)dx$$

Then integrating by parts, we have,

$$\int_0^1 p(x)u'(x)v'(x)dx + \int_0^1 q(x)u(x)v(x)dx = \int_0^1 f(x)v(x)dx \tag{1}$$

where the boundary term has vanished because $v(0) = v(1) = 0$. This equation has the general form

$$a(u,v) = (f,v), \ \forall u \in V$$

where, $a(\cdot,\cdot)$ is the bilinear form i.e. $a(u,v) = \int_0^1 pu'v'dx + \int_0^1 quvdx$ and $(\cdot,\cdot)$ is the linear form i.e.$(f,v) = \int_0^1 fvdx$. We choose a space $H(0,1)$ that satisfy the boundary conditions. We say, $u \in H$ is a weak solution in V if (1) is satisfied, for all $v \in H$ and $V = \{u, w \in H(0,1) : v(0) = v(1) = 0\}$.

Galerkin method

The problem is now to find $u_N \in V_N \subset V$ such that

$$\int_0^1 pu'v'dx + \int_0^1 quvdx = \int_0^1 fvdx \tag{2}$$

where, V_N is the discrete subspace of V.

Solution Algorithm

Now our aim is to solve (2) and to do so, we introduce a basis for V and V_N and make an anzats

$$u \approx u_N = \sum_{j=1}^{N} b_j v_j(x) \tag{3}$$

where, $v_j : (a,b) \to \mathbb{R}, j = 1, \cdots, N$ is basis for V_N which has the form,

$$v_j(x) = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}}, & \text{if } x_{j-1} < x < x_j. \\ \frac{x_{j+1}-x}{x_{j+1}-x_j}, & \text{if } x_j < x < x_{j+1} \\ 0, & \text{Otherwise} \end{cases}$$

Inserting (3) into equation (2) and letting $v = v_k, k = 1, \cdots, N$,we obtain,

$$\int_0^1 pv_k'(x)\Big(\sum_{j=1}^{N} b_j v_j'(x)\Big)dx + \int_0^1 qv_k(x)\Big(\sum_{j=1}^{N} b_j v_j(x)\Big)dx = \int_0^1 v_k(x)fdx \tag{4}$$

which has the equivalent form,

$$\sum_{j=1}^{N} b_j a(v_j, v_k) = (f, v_k), k = 1, \cdots, N \tag{5}$$

where, $a(\cdot,\cdot)$ is the bilinear form on $H(0,1)\times H(0,1)$ and $(\cdot,\cdot)$ is the linear form on $H(0,1)$.
We recognize (5) as a system of linear equations:

$$\mathcal{A}\mathcal{B}=\mathcal{F} \tag{6}$$

where,$\mathcal{A}_{j,k}=a(v_j,v_k), j=1,\cdots,N; k=1,\cdots,N; \mathcal{B}_j=b_j$ and $\mathcal{F}=(f,v_k)$.
Now we solve for the vector $[b_1,\cdots,b_N]^T$ and this completes our solution.

Calculation of the co-efficients of the matrices

We call $\mathcal{A}$ and $\mathcal{F}$ as stiffness matrix and load vector respectively. The entries $\mathcal{A}_{jk}$ form the matrix $\mathcal{A}$ and we need to compute it. To do so, letting $x_j-x_{j-1}=h_j$ and $x_{j+1}-x_j=h_{j+1}$ we derive, $v_j'(x)$ by,

$$v_j(x)=\begin{cases}\frac{x-x_{j-1}}{h_j},\, x\in[x_{j-1},x_j]\\ \frac{x_{j+1}-x}{h_{j+1}},\, x\in[x_j,x_{j+1}]\\ 0,\, x\notin[x_{j-1},x_{j+1}]\end{cases} \implies v_j'(x)=\begin{cases}\frac{1}{h_j},\, x\in[x_{j-1},x_j]\\ -\frac{1}{h_{j+1}},\, x\in[x_j,x_{j+1}]\\ 0,\, x\notin[x_{j-1},x_{j+1}]\end{cases}$$

We name $(v_j',v_k'),(v_j,v_k)$ as S,T respectively. Now we compute,

$$\begin{aligned}S_{j,j}&=\int_a^b p(x)\big(w_j'(x)\big)^2dx=\frac{p_j}{h_j}+\frac{p_{j+1}}{h_{j+1}}, j=1,\cdots,N\\ S_{j,j+1}&=\int_a^b p(x)v_j'(x)v_{j+1}'(x)dx=-\frac{p_{j+1}}{h_{j+1}}, j=1,\cdots,N\\ S_{j,j-1}&=-\frac{p_j}{h_j}, j=1,\cdots,N\end{aligned}$$

For T, we observe that v_jv_k is quadratic (i.e the product of two linear function). We can use Simpson's integration and that will give us, $T_{j,j}=\frac{1}{3}[q_jh_j+q_{j+1}h_{j+1}], T_{j,j+1}=q_{j+1}h_{j+1}\left(\frac{1}{6}\right), T_{j,j-1}=q_jh_j\left(\frac{1}{6}\right), j=1,\cdots,N$. Combining all these matrices lead us to the actual matrix,

$$\mathcal{A}=S_{j,k}+T_{j,k},\ j=1,\cdots,N, k=1,\cdots,N$$

which is a tridiagonal matrix and the entries of the matrix are,

$$\begin{aligned}\mathcal{A}_{j,j}&=\frac{p_j}{h_j}+\frac{p_{j+1}}{h_{j+1}}+\frac{1}{3}\Bigg[q_jh_j+q_{j+1}h_{j+1}\Bigg], j=1,\cdots,N,\\ \mathcal{A}_{j,j+1}&=-\frac{p_{j+1}}{h_{j+1}}+q_{j+1}h_{j+1}\left(\frac{1}{6}\right), j=1,\cdots,N,\\ \mathcal{A}_{j,j-1}&=-\frac{p_j}{h_j}+q_jh_j\left(\frac{1}{6}\right), j=1,\cdots,N\end{aligned}$$

Now it's time for load vector formulation. Using Trapezoidal rule,

$$\begin{aligned}\mathcal{F}=\int_0^1 fv_kdx&=\int_{x_{j-1}}^{x_{j+1}} fv_kdx=\int_{x_{j-1}}^{x_j} fv_kdx+\int_{x_j}^{x_{j+1}} fv_kdx\\ &\approx\big[f(x_{j-1})v_k(x_{j-1})+f(x_j)v_k(x_j)\big]\frac{h_j}{2}+\big[f(x_j)v_k(x_j)+f(x_{i+1})v_k(x_{j+1})\big]\frac{h_{j+1}}{2}.\end{aligned}$$

where, $j = 1, \cdots, N, k = 1, \cdots, N$. So far,we have some ideas about the structure of the stiffness matrix, let's talk about it's features. Is the matrix is positive definite[1]? The answer is yes.
We can show it by setting $\mathbf{v} = \{\phi_1, \phi_2, \cdots, \phi_N\}^T$ from $u_N = \sum_{j=1}^{N} b_j \phi_j(x)$.Then,

$$\begin{aligned}\mathbf{v}^T \mathcal{A} \mathbf{v} &= \sum_{j,k}^{N} (b_j \phi_j', b_k \phi_k') + (b_j \phi_j, b_k \phi_k) \\ &= (\phi', \phi') + (\phi, \phi) \\ &= \|\phi'\|_{L_2}^2 + \|\phi\|_{L_2}^2 > 0\end{aligned}$$

So, the proof is completed. But if $\|\phi'\| = 0$ then $\phi = C$ and as a result of boundary conditions $C = 0$. Therefore $u_N = 0$. Since $\{\phi_1, \phi_2, \cdots, \phi_N\}$ is linearly independent, which means b_j must be zero but $j = 1, \cdots, N$ so $\|\phi'\| > 0$.
Now the burning question aries how to solve $\mathcal{AB} = \mathcal{F}$? Generally, there are two categories of method which can handle these type of equations. One of them is direct methods which includes Gaussian elimination, LU, Cholesky, and QR decomposition. When the matrix size is too large, these methods are costly and we do not get direct access to the entire matrix at once. On the other hand, the next category is called iterative methods. The idea is to assume an appropriate guess for the solution. Then keep iterating towards the solution ($\mathcal{AB} - \mathcal{F} = 0$). In the case of symmetric problem, one might choose conjugate gradient method to solve this kind of linear system. Other iterative methods are Jacobi iteration,Gauss-Seidel method and Successive Overrelaxation (SOR). Although iterative methods are very effective in handling large matrices but it's productivity completely depends on the preconditioner/solver.
According to our matrix (for the current problem), we observe that $\mathcal{A}$ is strictly diagonally dominant i.e. $|\mathcal{A}_{j,j}| > \sum_{j \neq k} |\mathcal{A}_{j,k}|$ which implies it must be invertible. We choose direct methods to solve for the matrix.

MATLAB implementation

Case 1:
We will use Galerkin FEM to solve the following BVP, where we have chosen, $p(x), q(x)$ as constant and $f(x) = x^3$.

$$u''(x) + u(x) = x^3, \; u(0) = u(1) = 0$$

N	L2 Error	Order of convergence	Max error	Order of convergence
5	7.37E-04		3.86E-02	
10	1.91E-04	1.95125	9.50E-03	2.02225
20	4.86E-05	1.97005	2 .37E-03	2.00218
40	1.23E-05	1.983527	5 .92E-04	2.00005

Table 1: L2 error, Max error and their order of convergence for the problem

[1]A matrix (A) is positive definite if $\mathbf{v}^T A \mathbf{v} > 0$ whenever $\mathbf{v} \neq 0$.

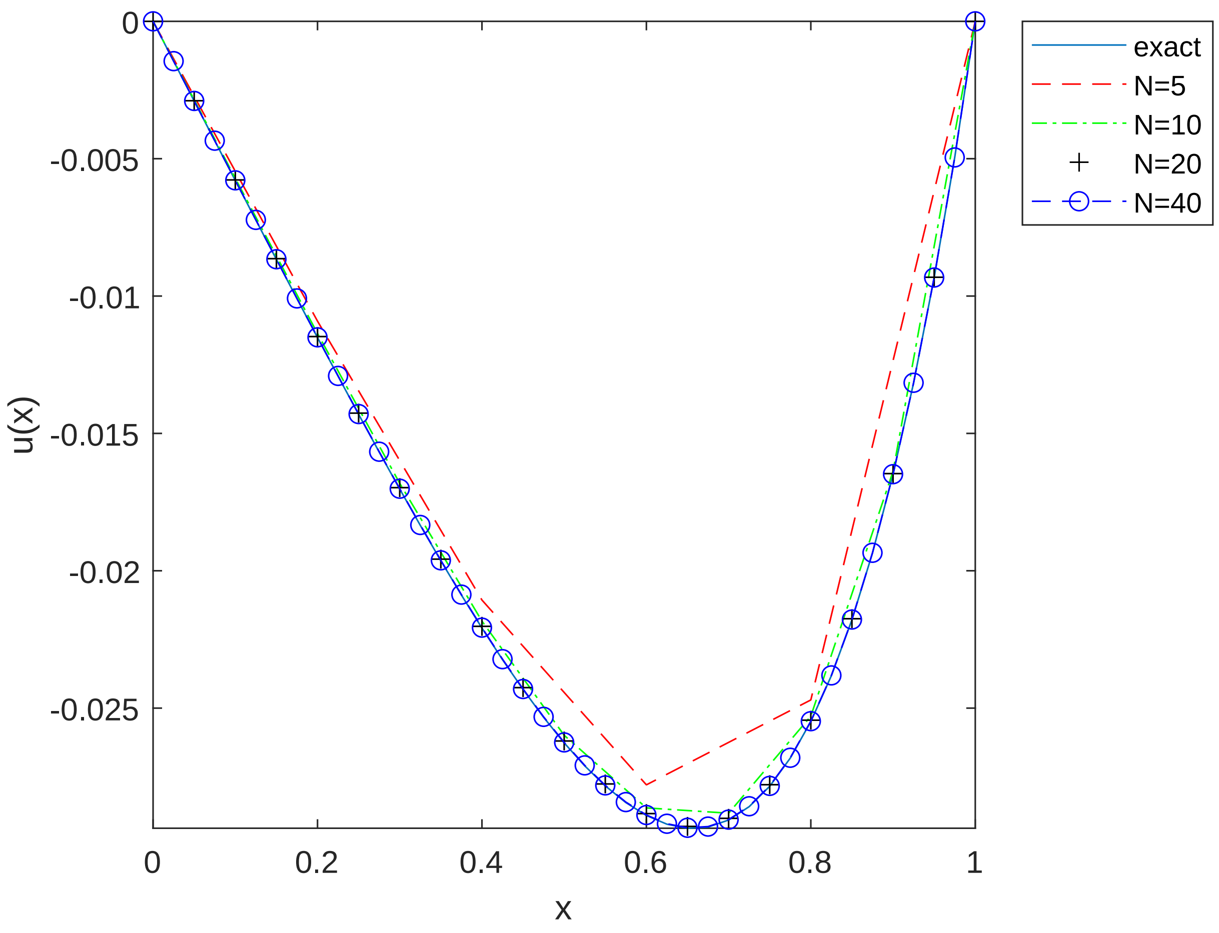

Figure 1: Output for $u'' + u = x^3, u(0) = u(1) = 0$

Using $cond(A)$ in MATLAB, we have found out the condition number[2] of the stiffness matrix (about 204).It turns out the matrix is well-posed. Changing $N = 5, 10, 20, 40$, we get better approximation. Another possible solution to get better approximation by choosing quadratic basis function that also might reduce running time for the program. From the table 1 we can see it's order of convergence is 2.

Case 2:
Now lets try to solve another problem. This time we choose $p(x) = (x+1), q(x) =$some constant.

$$-\frac{d}{dx}\left((x+1)\frac{d}{dx}u\right) + 6u = -12x^4 + 44x^3 - 2x + 1, u(0) = u(1) = 0$$

which has the exact solution, $u(x) = -2x^4 + 2x^3 - x^2 + x$.

Just like before we get the order of convergence is 2 in both cases as expected.

[2]Condition number of a matrix indicates if the solution of the linear system is sensitive to small changes

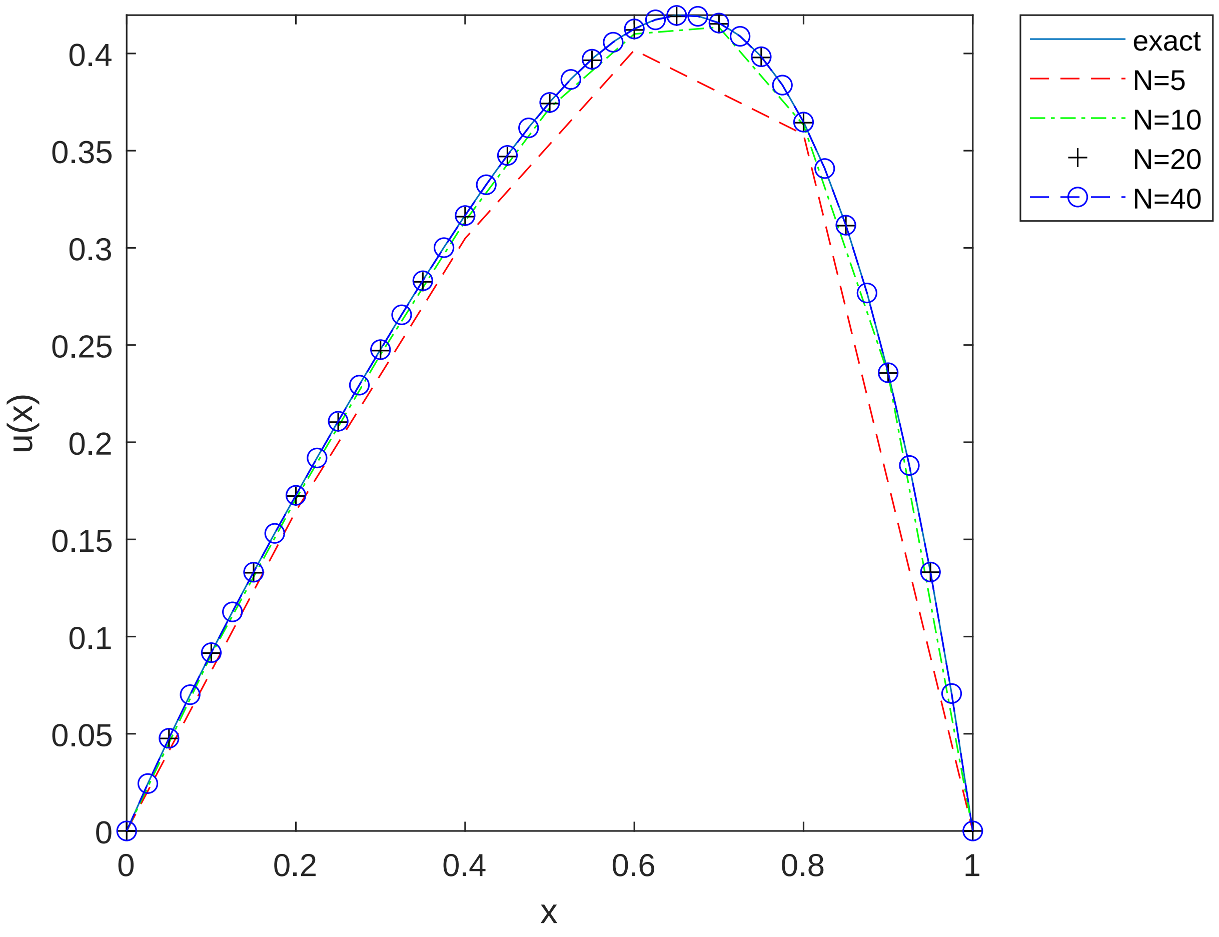

Figure 2: Output for $-\frac{d}{dx}\left((x+1)\frac{d}{dx}u\right)+6u=-12x^4+44x^3-2x+1, u(0)=u(1)=0$

N	L2 Error	Order of Convergence	Max Error	Order of Convergence
5	0.008		0.029	
10	0.002	1.946	0.0072	2.0082
20	5.23E-04	1.9687	0.0018	2.0046
40	1.32E-04	1.9832	4.49E-04	2.0006

Table 2: L2 error, Max error and their order of convergence for the problem

Conclusion and Future work

The project has demonstrated how to solve Dirichlet boundary value problem using Galerkin FEM and it's implementation in MATLAB. We have solved the problems when we can solve the problems analytically. Currently, the author is working on the problems when the differential equations can not be solved analytically.

Bibliography

[1] R. Hughes. *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[2] Young W. Kwon, Hyochoong Bang. *The Finite Element Method Using MATLAB*,2nd Edition,

[3] Class notes of the lecture of B. Khouider on Scientific Computing, Fall-2019.

[4] O. Axelsson, V. A. Barker *Finite Element Solution of Boundary Value Problems: Theory and Computation*,1984.