

## Part II: Implementation (reattempted)

The purpose of this exercise was to use the Finite Element Method (FEM) to approximate, and study, the solution to the problem presented in Eq. 1.

$$-(p(x)u'(x))' + q(x)u(x) = f(x) \quad (1)$$

$$u(0) = \alpha, u(1) = \beta \quad (2)$$

This equation was solved on the domain  $x \in [0, 1]$ . The analytical solution was assumed to exist uniquely. The variables in Eq. 1 are as denoted below in Eq. 3.

$$u(x) \in C^2[0, 1] \quad (3)$$

$$f, q \in C^0[0, 1] \quad q \geq 0 \quad (4)$$

$$p \in C^1[0, 1] \quad p > 0 \quad (5)$$

The problem domain of one unit in one dimension was divided into a nonuniform but structured mesh. The rule for determining element size is shown in the equation below.

$$h_j = \begin{cases} 0.9\Delta x & \text{for odd } j \\ 1.1\Delta x & \text{for even } j \end{cases}$$

The  $\Delta x$  referred to is the average size of any given element. Specifically, it is calculated by use of Eq. 6.

$$\Delta x = \frac{1}{N+1} \quad (6)$$

Here,  $N$  is the number of degrees of freedom;  $N+1$  is the number of elements. Tests were conducted for  $N = 10, 20, 40, 80, 160, 320$ . These six mesh size tests were conducted for six different cases which are described in Table 1.

A computer code was written to evaluate the posed problems. The results follow this section and the code itself is ~~appended to this report~~ can be viewed online at [www.github.com/JustinClough/Learning\\_Codes/tree/master/FEA](https://www.github.com/JustinClough/Learning_Codes/tree/master/FEA); it was not printed to conserve paper. Plots of the error for  $N = 10, 20, 40$  with respect to location are presented in Figures 1 and Figure 2 for the first case. Plots of the error for  $N = 10, 20, 40$  with respect to location are presented in Figures 3 and Figure 4 for the second case. Tabulated data, in the form of error norms and convergence order are presented in Tables 2 through 6.

Table 1: Case numbers and descriptions.

Case Number	$\alpha$	$\beta$	$p$	$q$	$u$
1	0	0	3	2	$x(x-1)(\sin(5x) + 3e^x)$
2	0	0	$1+x$	0	$x(x-1)(\sin(5x) + 3e^x)$
3	4	4	3	2	4
4	-2	-1	3	2	$x-2$
5	-3	-2	3	2	$x^2-3$

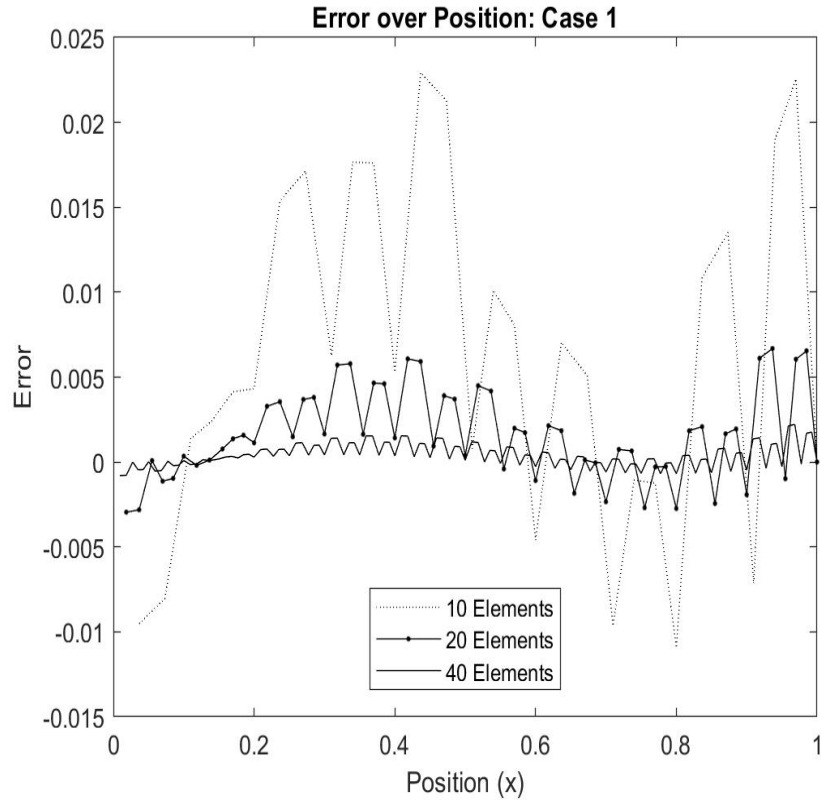


Figure 1: Error in solution over domain of problem.

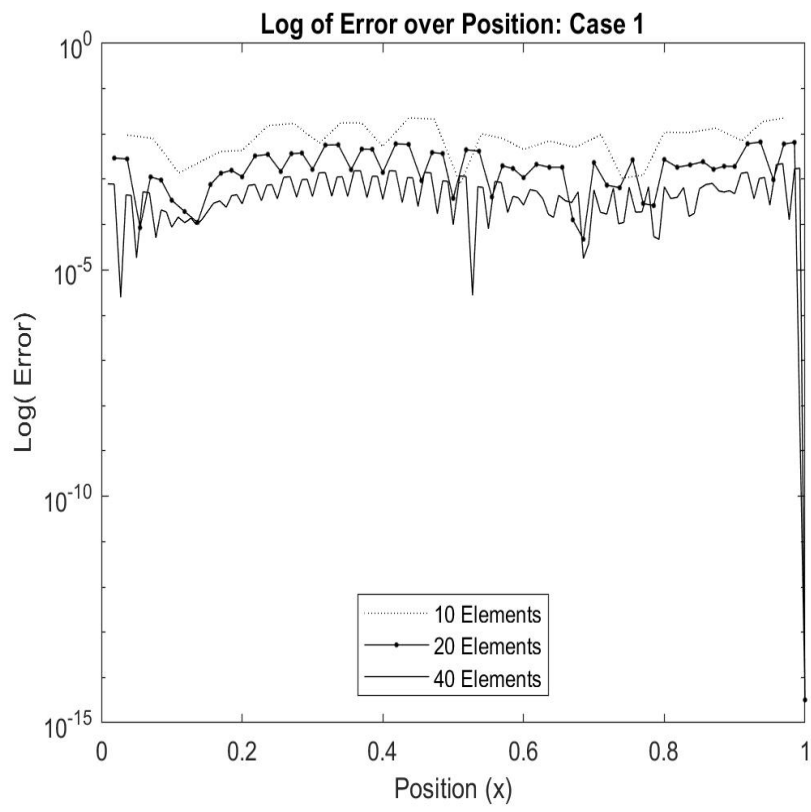


Figure 2: Log of error in solution over domain of problem.

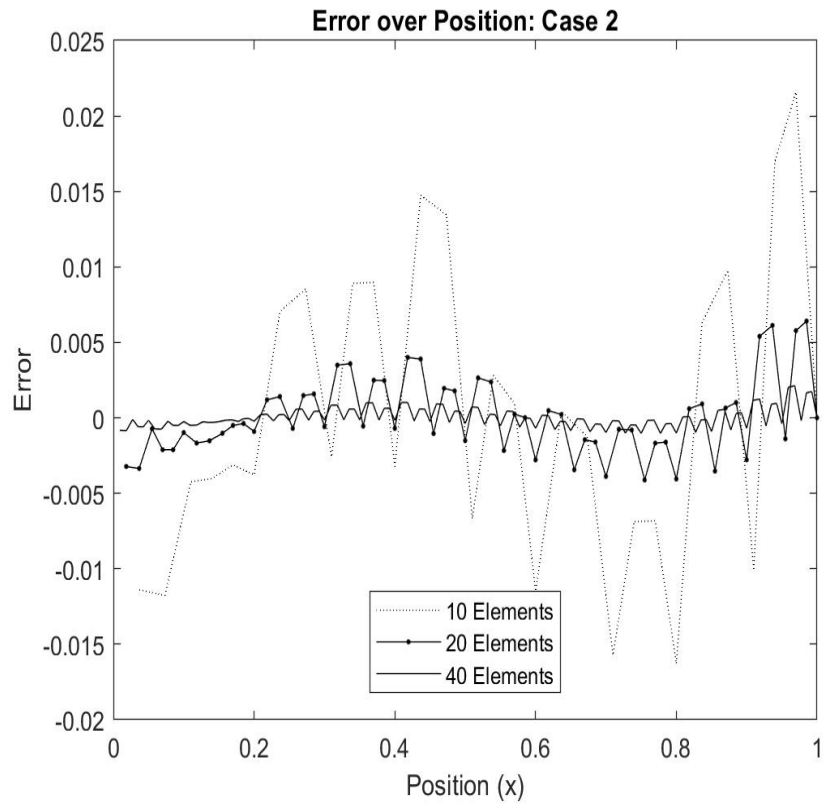


Figure 3: Error in solution over domain of problem.

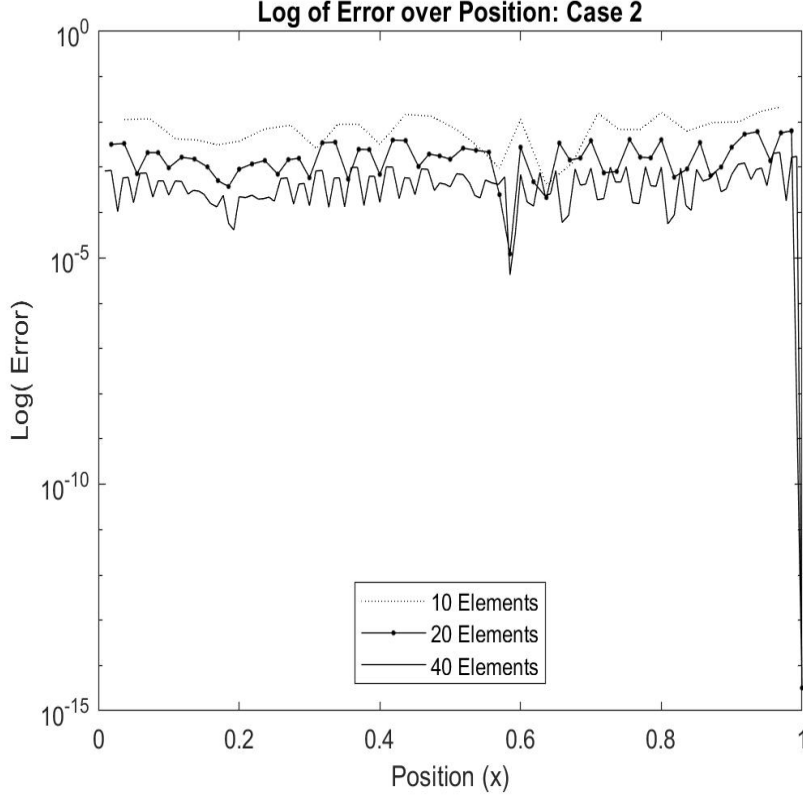


Figure 4: Log of error in solution over domain of problem.

Table 2: Errors and convergence orders for Case 1.

$N + 1$	$\ e_h(\cdot)\ _{L^2([0,1])}$	Order	$\ e_h(\cdot)\ _{L^\infty([0,1])}$	Order	$\ e_h(\cdot)\ _h$	Order
10	3.528607e-03	0.00	2.291486e-02	0.00	3.317002e-01	0.00
20	8.963598e-04	1.97	6.683362e-03	1.77	1.792580e-01	0.88
40	2.250262e-04	1.99	2.211256e-03	1.59	9.377064e-02	0.93
80	5.631579e-05	1.99	6.268937e-04	1.81	4.803148e-02	0.96
160	1.408267e-05	1.99	1.663275e-04	1.91	2.431665e-02	0.98
320	3.520901e-06	1.99	4.280166e-05	1.95	1.223537e-02	0.99

The following was done to incorporate non-homogeneous Dirichlet boundary conditions. First, the only change to the weak form of the problem would be separating the trial and test spaces; the solution must have a non-zero value on the specified boundary whereas the trial function must be zero on the boundaries. For the FEM formulation, the contributions of the non-zero boundary conditions are taken account for in the forcing vector. No changes need to be made to represent  $U_h$  on

Table 3: Errors and convergence orders for Case 2.

$N + 1$	$\ e_h(\cdot)\ _{L^2([0,1])}$	Order	$\ e_h(\cdot)\ _{L^\infty([0,1])}$	Order	$\ e_h(\cdot)\ _h$	Order
10	5.291277e-03	0.00	2.157976e-02	0.00	3.260128e-01	0.00
20	1.310432e-03	2.01	6.406604e-03	1.75	1.774688e-01	0.87
40	3.273072e-04	2.00	2.134625e-03	1.58	9.327537e-02	0.92
80	8.184745e-05	1.99	6.168818e-04	1.79	4.790149e-02	0.96
160	2.046769e-05	1.99	1.650486e-04	1.90	2.428336e-02	0.98
320	5.117846e-06	1.99	4.264007e-05	1.95	1.222695e-02	0.98

Table 4: Errors and convergence orders for Case 3.

$N + 1$	$\ e_h(\cdot)\ _{L^2([0,1])}$	Order	$\ e_h(\cdot)\ _{L^\infty([0,1])}$	Order	$\ e_h(\cdot)\ _h$	Order
10	5.058005e-15	0.00	1.243450e-14	0.00	1.539017e-14	0.00
20	2.019833e-15	1.32	7.105427e-15	0.80	2.019843e-14	-0.39
40	3.587761e-14	-4.15	1.021405e-13	-3.84	1.571914e-13	-2.96
80	7.490578e-14	-1.06	2.904343e-13	-1.50	5.474745e-13	-1.80
160	9.489067e-14	-0.34	2.859935e-13	0.02	8.401636e-13	-0.61
320	4.464905e-12	-5.55	1.042100e-11	-5.18	1.436984e-11	-4.09

Table 5: Errors and convergence orders for Case 4.

$N + 1$	$\ e_h(\cdot)\ _{L^2([0,1])}$	Order	$\ e_h(\cdot)\ _{L^\infty([0,1])}$	Order	$\ e_h(\cdot)\ _h$	Order
10	2.170933e-15	0.00	5.329071e-15	0.00	6.018740e-15	0.00
20	6.371945e-16	1.76	2.442491e-15	1.12	7.750267e-15	-0.36
40	1.067982e-14	-4.06	3.175238e-14	-3.70	5.341707e-14	-2.78
80	2.419360e-14	-1.17	7.016610e-14	-1.14	1.886608e-13	-1.82
160	3.201968e-14	-0.40	1.316725e-13	-0.90	3.214517e-13	-0.76
320	1.626372e-12	-5.66	3.860690e-12	-4.87	5.246423e-12	-4.02

interior elements. However, on the boundary, additional hat functions are used to recover the assigned boundary values. The matrix  $\mathbf{A}$  is not changed as the original PDE does not change. The forcing vector  $\mathbf{F}$  includes the addition of the boundary terms on the first and last entry.

The error measures behave as expected for Cases 1 and 2; the order of the  $L^2$  and  $L^\infty$  norms tend to 2.0 and the order of the energy norm tends to 1.0. The solution to Cases 3 and 4 was captured up to machine precision. This is shown in the first row of Tables 4 and 5. The reason for the increasing norm values with increasing mesh density was due to round off error in calculating the norms. For

Table 6: Errors and convergence orders for Case 5.

$N + 1$	$\ e_h(\cdot)\ _{L^2([0,1])}$	Order	$\ e_h(\cdot)\ _{L^\infty([0,1])}$	Order	$\ e_h(\cdot)\ _h$	Order
10	3.411114e-15	0.00	2.688889e-03	0.00	5.648304e-02	0.00
20	1.557481e-15	1.13	6.722222e-04	2.00	2.877427e-02	0.97
40	2.048400e-14	-3.71	1.680556e-04	2.00	1.451849e-02	0.98
80	4.361838e-14	-1.09	4.201389e-05	2.00	7.291860e-03	0.99
160	6.317536e-14	-0.53	1.050347e-05	2.00	3.654057e-03	0.99
320	2.883586e-12	-5.51	2.625868e-06	2.00	1.829057e-03	0.99

example, assuming machine precision is  $\epsilon = 10^{-16}$  then  $3 * 320 * \epsilon = 9.6 \times 10^{-14}$ ; this represents sum taken to compute the  $L^2$  norm on the mesh with  $N + 1 = 320$  with floating point arithmetic in the best case senario. The  $L^2$  norm for Case 5 behaves similar to that of Cases 3 and 4. The  $L^\infty$  and energy norms behave similar to that of Cases 1 and 2.

The **Eigen** software package was used as the linear solver for this project. The matrix inversion method used was Householder-QR with pivoting.