ME 280A
Homework 4
Ahmad Zareei

Introduction

In this problem set we will solve, the 1-D linear conservation equation as

$$\frac{d}{dx}\left(A_1\frac{du}{dx}\right) = f\left(x\right)$$

with finite element method with adaptive meshing. This equation is a 1-D linear elliptic differential equation. For this equation, so far, we have seen numerical FEM solution with different boudnary conditions (either Neumann or Dirichlet) and also with discontinuous A_1. Moreover, we have used different elements (linear, quadratic or cubic) to solve this equation. In the previous homework, we also worked through preconditioning method for solving the final linear system of equations. In this homework we will discuss the effect of adaptive meshing on number of element and convergence of FEM for a certain accuracy.

Adaptive meshing automatically will refine, coersen or change the mesh in order to achieve a solution with specific accuracy in a very optimal fashion. Roughly speaking, we will end up with smaller elements where solution has higher variations and larger elements where solution is smooth. Adaptive finite element method will start with a rough mesh, and then check if the solution in each element satisfy a criteria (e.g. error less than a certain tolerance). If the element fails to satisfy the accuracy needed, adjustment (in here choping into two elements) will be made to obtain the desired solution. In this assignment, we will work through the method and implementation technique that we used to solve 1-D linear conservation equation.

Objectives

(I) Solving 1-D linear elliptic equation with FEM for two different cases of a continuoous and discontinuous A_1.

(II) Investigating the effects of fixing a node at discontinuity point of A_1 and comparing with the case where elements have a varying A_1.

(III) Understanding the local error and checking the variation of this error over the domain

(IV) Implementing the Adaptive Finite Element Method (AFEM)

(V) Investigating the effects of AFEM on number of the elements

(VI) Investigating the effects of AFEM on local variation error

(VII) Investigating the effect of local error on the global error

Problem, Procedure and Results

1. Consider the two following problems, with their analytical solutions, defined over the domain $\Omega = (0, L)$:

case a)

$$\frac{d}{dx}\left(A_1 \frac{du}{dx}\right) = -\,90\pi^2 \sin(3\pi x)\sin(36\pi x^3) + (10\sin(3\pi x)+5)(216\pi x\cos(36\pi x^3) - 11664\pi^2 x^4 \sin(36\pi x^3)) \tag{1}$$

with $A_1 = 1.0$, $L = 1$ and boundary conditions of $u(0) = 0$ and $u(L) = 0$. The true solution in this case is as

$$u^{True} = (10\sin(3\pi x)+5)\sin(36\pi x^3) \tag{2}$$

case b)

$$\frac{d}{dx}\left(A_1 \frac{du}{dx}\right) = 256\sin(\frac{3}{4}\pi x)\cos(16\pi x) \tag{3}$$

with

$$A_1 = \begin{cases} 0.2 & x < 1/3 \\ 2.0 & x \geq 1/3 \end{cases} \tag{4}$$

and $L = 1$. Boundary conditions in this case are: $u(0) = 0$ and $A_1 \mathrm{d}u/\mathrm{d}x|_{x=L} = 1$. The true solution in this case can be found the assignment. Compute the finite element solution u^N to both problems using linear equal-sized elements. For each problem, determine how many elements are needed in order to achieve

$$e^N = \frac{||u - u^N||_{A_1(\Omega)}}{||u||_{A_1(\Omega)}} \leq \text{TOL } = 0.05, \tag{5}$$

where

$$||u||_{A_1(\Omega)} = \sqrt{\int_\Omega \frac{du}{dx} A_1 \frac{du}{dx} dx}. \tag{6}$$

For the second problem, do the process twice. First, solve it with equally spaced elements. Then, place a mesh node exactly at the discontinuity, and mesh each side with equally sized elements so that all of the elements on the right side are the same size, and all of the elements on the left side are the same size. By uniformly increasing the number of elements on each side (so that $h_{left} \approx h_{right}$), determine how many total elements are required to satisfy the error criterion.

Solution

We did the first steps as previous homeworks, where we devide the interval of $[0, 1]$ to N equally sized intervals with length of $1/N$. Then as before, we computed the stiffness matrix $[K]$ and the loading vector $\{L\}$. As we derived in the previous homeworks, contribution of element e to the ijth element of the stiffness matrix i.e. K^e_{ij} is as

$$K^e_{ij} = \int_{\Omega_i} \frac{\mathrm{d}\phi_i}{\mathrm{d}x} A_1 \frac{\mathrm{d}\phi_j}{\mathrm{d}x} \mathrm{d}x = \int_{-1}^{1} \frac{\mathrm{d}\hat{\phi}_i}{\mathrm{d}\zeta} A_1 \frac{\mathrm{d}\hat{\phi}_j}{\mathrm{d}\zeta} \frac{\mathrm{d}x}{\mathrm{d}\zeta} \mathrm{d}\zeta \tag{7}$$

where ϕ_i is the linear shape function which is 1 on node i and zero on every other nodes and $\hat{\phi}_i$ is the linear shape function on the master element. Master element is an element in the ζ-world where this element is in the interval of $[-1, 1]$ and all the computations are done in this domain. Also contribution of element e to the loading vector is as

$$L^e_i = \int_{\Omega_i} f(x)\phi_i(x)\mathrm{d}x = \int_{-1}^{1} f(x(\zeta))\hat{\phi}_i(\zeta)\frac{\mathrm{d}x}{\mathrm{d}\zeta}\mathrm{d}\zeta \tag{8}$$

Each elements contribution will then be added to gloabal stiffness matrix $[K]$ and loading vector $\{L\}$. For adding these contribution to the global matrix, we need the position of each node of the elements. This iformation is stored in `element`, where it shows the nodes of each element. In our case that it is a simple $1D$ problem, this connectivity matrix is as

$$[e]_{\text{table}} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ ... & \\ N & N+1 \end{bmatrix}. \tag{9}$$

List of nodes is stored in vector `points`, where it shows the position of nodes in the x direction i.e. the position of node i is just `points(i)`. For integration as before, we use Guass quadrature method.

We have discussed the implementation of boundary conditions of Dirichlet or Neumann type boundary conditions in previous assignments (Assignment 1 for Dirichlet boundary condition/ Assignment 2 for Neumann boundary condition).

The only point to mention is that in **case(b)** for the second part, where we want to have a node at discontinuity point ($x = 1/3$), we put `floor(N/3)` elements on the left and the rest will go to the right of discontinuity point.

The results for number of points for the proper error of $e^N = 0.05$ are tabulated in table 1. Figures for this solution are plotted in figures 1, 2 and 3.

2. Plot the position of the first node of each element X_I versus E_I , where

$$E_I = \frac{\frac{1}{h_I}||u - u^N||^2_{A_1(\Omega)}}{\frac{1}{L}||u||^2_{A_1(\Omega_I)}}, \tag{10}$$

case(a)	**case(b)**	**case(b)** with fixed point at discontinuity
1456	369	252

Table 1: Number of elements needed for different cases with linear finite element and error e^N of 0.05

Here I is the element index, hI is the length of element I, Ω_I is the domain of element I, and the error norm over an individual element is defined as

$$||u||_{A_1(\Omega)} = \sqrt{\int_\Omega \frac{du}{dx} A_1 \frac{du}{dx} dx}. \tag{11}$$

Solution

We computed the E_I on each element as suggested in (10) and plotted the results with respect to the left node of each element and the results are shown in figures 4 and 5.

3. Modify the 1D FEM code from the previous assignments so that it can automatically refine the mesh using the following process:

   - Solve the finite element proble for a given mesh.
   - Compute E_I on each element.
   - Subdivide every element in which $E_I \geq \mathrm{TOL}_E$ into two equal parts.
   - Repeat until no elements need to be subdivided

   Refine the mesh (by dividing elements into two) until $E_I \geq \mathrm{TOL}_E$ for all I. Use this criterion to refine your mesh, starting with $N = 16$ equal-sized elements. Solve both of the problems using automatic refinement.

   - By adjusting TOL_E, determine how many elements are needed to achieve $e^N \leq \mathrm{TOL} = 0.05$ using this procedure.
   - Plot the final solution together with the exact solution. Use appropriate markers to denote how the elements are distributed in your final numerical solution.
   - Tabulate the final number of elements that fall into each of the initial 16 elements.
   - Plot X_I versus E_I for the solution.
   - For the second problem, repeat the process, but mesh around the discontinuity in $A1$. Put the sixth of the initial mesh points exactly on the discontinuity and have equal sized elements on either side, so that there are 5 elements to the left and 11 to the right. Perform the refinement procedure and comment on differences from the original initial mesh.

Solution

In this section in order to implement the adaptive finite element method (AFEM), we followed the following procedure. First we devide the interval of $[0, 1]$ into 16 elements and solve for the solution for this mesh and then computed the local error E_I as defined before for each element. Then, we go over all the elements, and fetch the connectivity matrix `e` elements. Now, if the element does not satisfy the accuracy demanded for each element E_I, we split the element in two and add a point to vector of `points` and then update the connectivity matrix `e`, stiffness matrix $[K]$ and loading vector $\{L\}$. After all of updates, we again solve the linear system of equations and again compute the local error of each element. We will repeat this procedure untill the local error in each element satisfy our error condition.

To better understand the method, consider this example. Suppose that one of the elements that does not satisfy the error criteria, is element 5 with node numbers `elements = e(5,:)`. We first fetch the positions from the vector `points` by reading `p=points(e(5,:))`. Now we want to split this element into two parts. We first add a point to the list of nodes as

```
points(end+1)=(p(1)+p(2))/2.
```

Then we update the connectivity matrix as

```
e(5,:)=[elements(1),length(points)]
```

and

```
e(end+1,:)=[length(points),elements(2)].
```

Now we update the stiffness matrix and the loading vector. For this, we first substract the contribution of the previous element that is now splitted. Then we compute the contribution of these new nodes and add them to stiffness matrix and loading vector.

For adjusting the E_I, we did the following computation, as

$$E_I = \frac{\frac{1}{h_I}||u - u^N||^2_{A_1(\Omega)}}{\frac{1}{L}||u||^2_{A_1(\Omega_I)}} \quad \rightarrow \quad h_I E_I = \frac{||u - u^N||^2_{A_1(\Omega)}}{||u||^2_{A_1(\Omega_I)}} \tag{12}$$

So we will have

$$\left(e^N\right)^2 = \sum \frac{||u - u^N||^2_{A_1(\Omega)}}{||u||^2_{A_1(\Omega_I)}} = \sum h_I E_I \tag{13}$$

If we suppose that all of the elements have the local error less than TOL_E i.e $E_I \leq \mathrm{TOL}_E$, we will have

$$\left(e^N\right)^2 = \sum h_I E_I \leq \sum h_I \mathrm{TOL}_E = \mathrm{TOL}_E \sum h_I \tag{14}$$

since $\sum h_I = 1$, we will have

$$\left(e^N\right)^2 \leq \mathrm{TOL}_E \tag{15}$$

Since we want $e^N \leq 0.05$, if we set $\mathrm{TOL}_E \leq 0.05^2$, we will have

$$\left(e^N\right)^2 \leq \mathrm{TOL}_E \leq 0.05^2 \quad \rightarrow \quad e^N \leq 0.05 \tag{16}$$

So by setting the tolerance of local error E_I to $0.05^2 = (0.0025)$, we will be sure that the total error is less than $e^N \leq 0.05$. Of course this value for TOL_E is not an optimum value for each element. However, this value of TOL_E is greedy and for the worst case, we are stil sure that the value of e^N is less than or equal to 0.05.

Results for this case are shown in figures 6, 7 and 8. The local error versus first node of each element is also shown in figures 9 and 10.

Number of elements in each interval is tabulated for different cases in table 2. Note that, in case (b), the interval numbers show different intervals. Since for the case where a node is fixed at the discontinuity point will change the intervals. in case (a) and case (b) in table 2, interval i is the interval of $[(i-1)/16, i/16]$, where in case (b) with fixed node, intervals are

$$[0, 1/15], [1/15, 2/15], [2/15, 3/15], [3/15, 4/15], [4/15, 5/15]$$
$$[11/33, 13/33], [13/33, 15/33], [15/33, 17/33], ...[31/33, 33/33]$$

Conclusion

In this problem set, we used adaptive meshing to solve a 1-D linear elliptic equation. We see that the use of adaptive meshing will decrease the number of nodes needed to solve a problem. Adaptive meshing will automatically refine the mesh where the solution has higher variations and error has higher values. Solving with adaptive finite element method, we can only have a control over the local error, while at the end we need a control over the global error. We saw that for a greedy choice of local error as $\mathrm{TOL}_E = \left(e^N\right)^2$, we can reach the global error wanted. This choice of local error is greedy and there could be other optimal local error used to give a better result. However, still we see that the number of nodes decrease dramatically from 1456 to 424 elements as we see in the first case. This is one order of magnitude decreaes in number of elements needed.

In the second case, we saw that if we could set a node at discontinuity point, number of elements will decrease from 369 to 252. This means that using a mesh where we use our knowledge of the solution, can decrease number of elements needed to reach a certain global error.

Interval	**case (a)**	**case (b)**	**case(b)-fixed node**
1	1	4	4
2	1	7	9
3	1	12	12
4	2	14	17
5	1	18	20
6	2	15	7
7	3	7	9
8	6	10	9
9	8	11	10
10	6	12	10
11	13	12	10
12	37	12	10
13	66	12	10
14	92	10	10
15	99	8	7
16	86	6	6
Total	**424**	**170**	**160**

Table 2: Number of elements needed in each interval for different cases with adaptive finite element method and local tolerance error of $\mathrm{TOL}_E = 0.05^2$. The fourth column is the case where a node is fixed at discontinuity point.

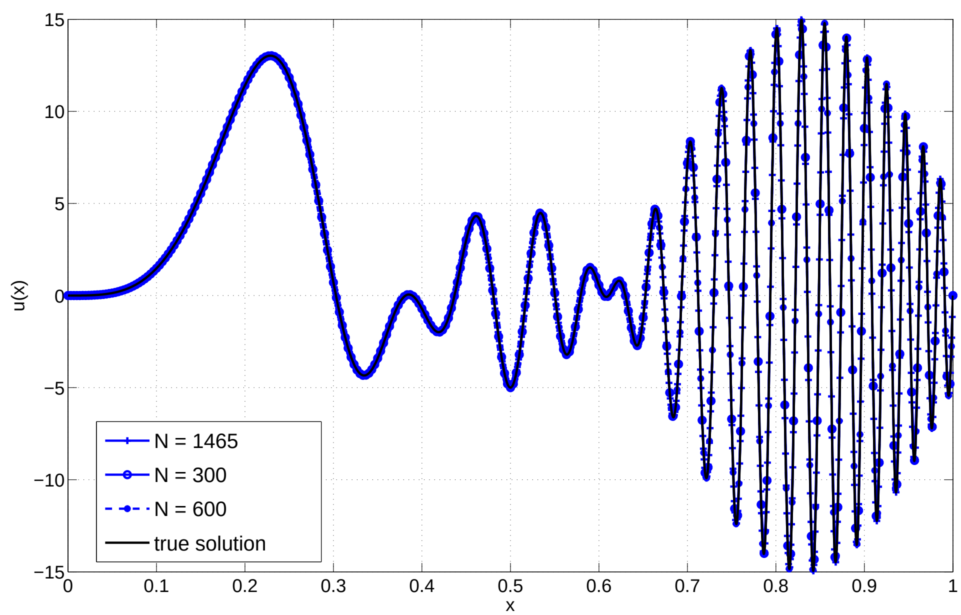

Figure 1: Solution of **case (a)**, with $N = 1456$ for the error of 0.05 with true solution. Some number of nodes in between $N = 300, 700$ are plotted to visually see the convergence

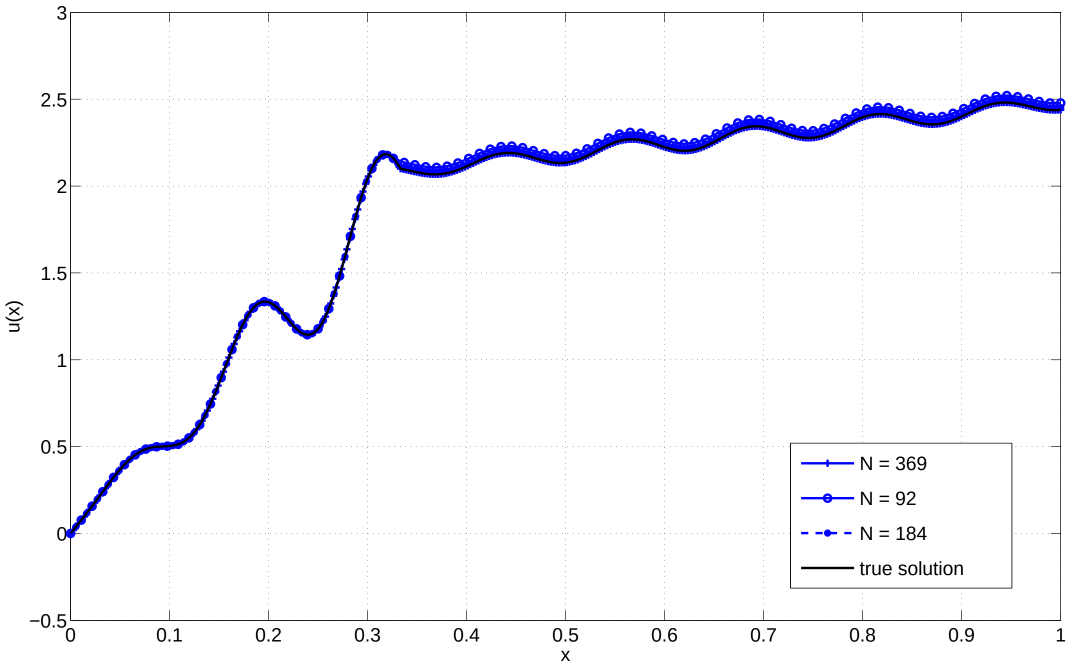

Figure 2: Solution of **case (b)**, with $N_0 = 369$ for the error of 0.05 with true solution. Some number of nodes in between $N = 92(N_0/2)$ and $N = 184(N_0/2)$ are plotted to visually see the convergence

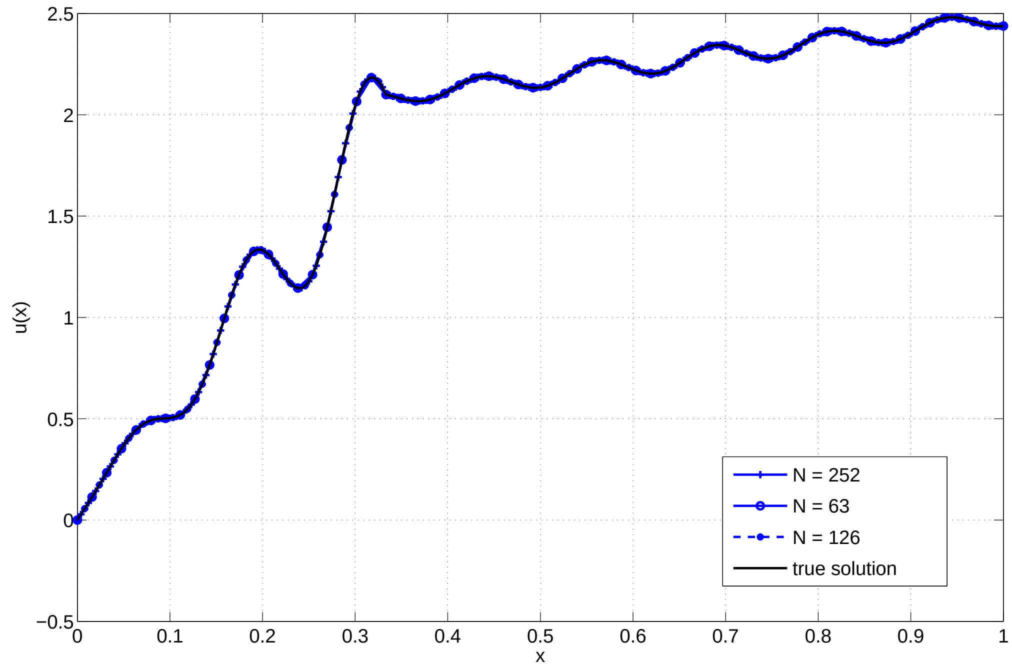

Figure 3: Solution of **case (b)** where we have a fixed node at the discontinuity point, with $N_0 = 252$ for error of 0.05 with true solution. Some number of nodes in between $N = 63(N_0/2)$ and $N = 126(N_0/2)$ are also plotted

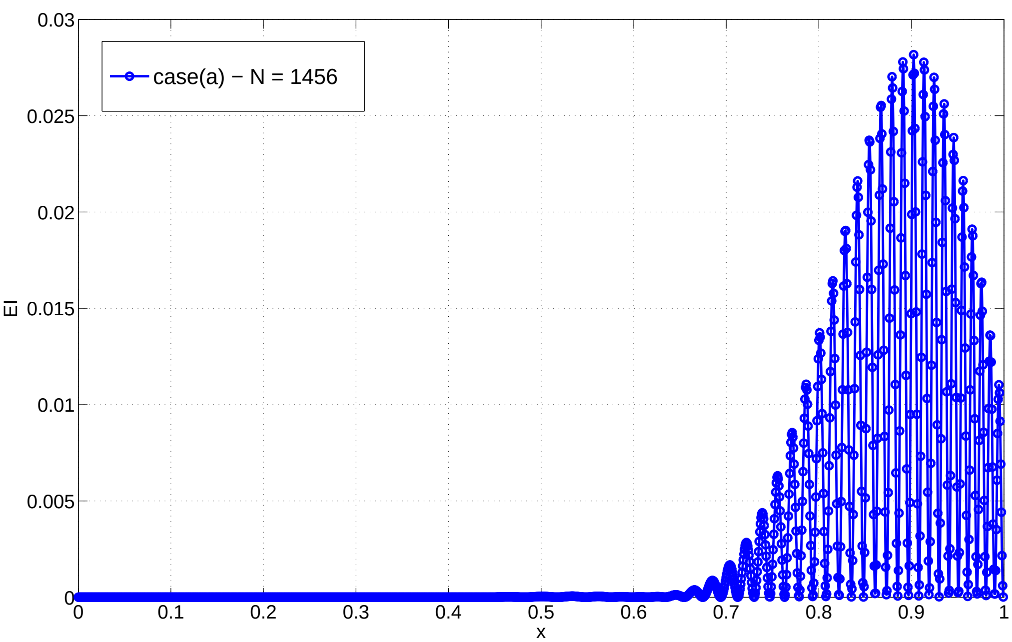

Figure 4: E_I versus X_I for **case (a)** for $N = 1456$ with global error of e^N of 0.05

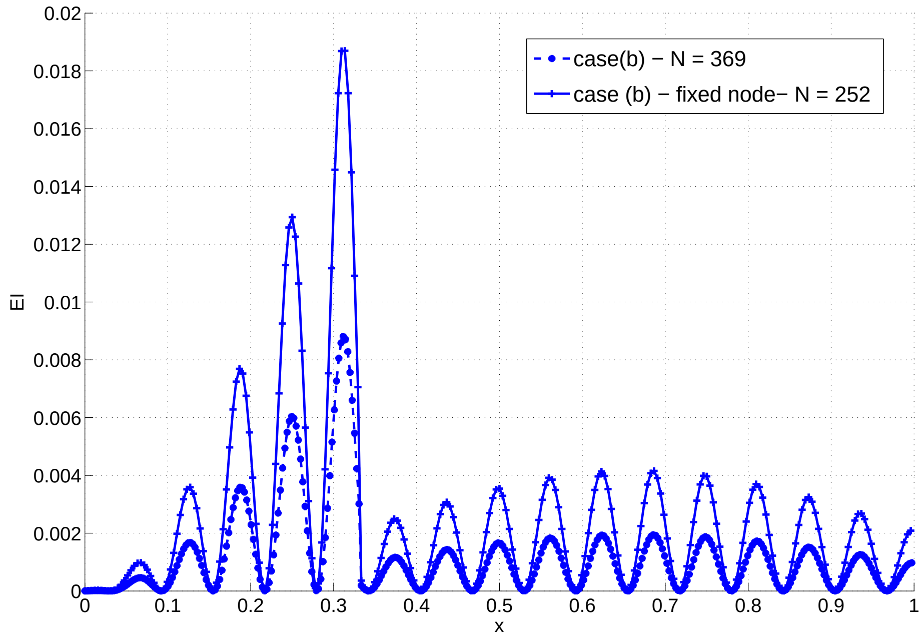

Figure 5: E_I versus X_I for **case (b)** for two cases where nodes are equally spaced and $N = 369$ and where a node is fixed at discontinuity point and $N = 252$. In all of the cases global error e^N is 0.05

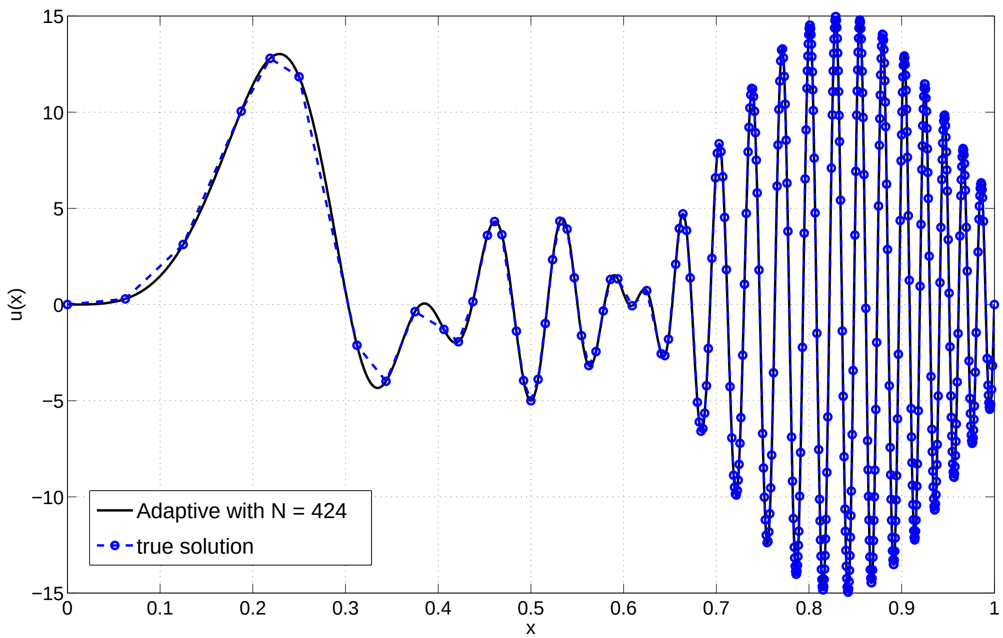

Figure 6: Adaptively solved, numerical solution of **case (a)** for local error of $\mathrm{TOL}_E = 0.05^2$. Number of elemenents for this case is $N = 424$.

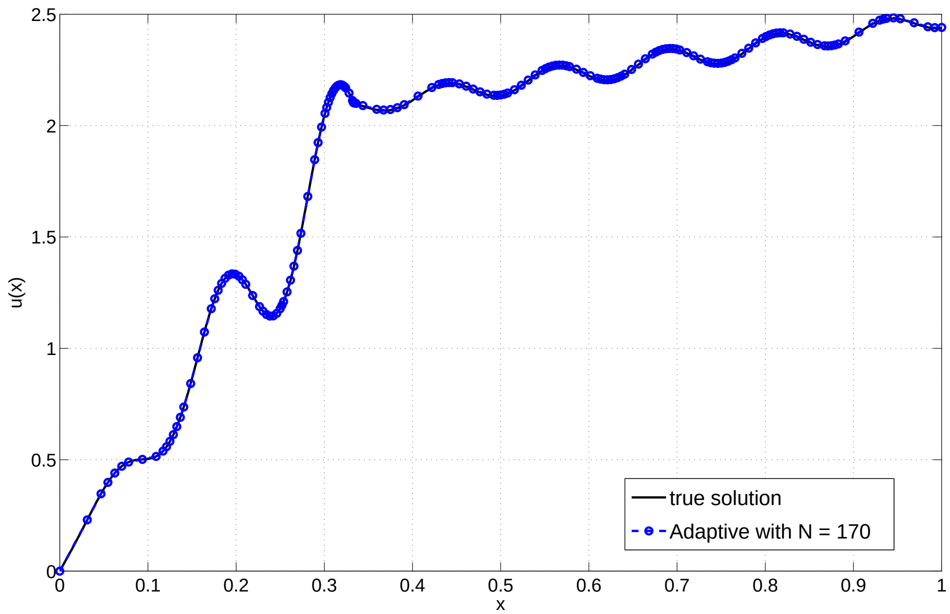

Figure 7: Adaptively solved, numerical solution of **case (b)** for local error of $\mathrm{TOL}_E = 0.05^2$. Number of elemenents for this case is $N = 170$.

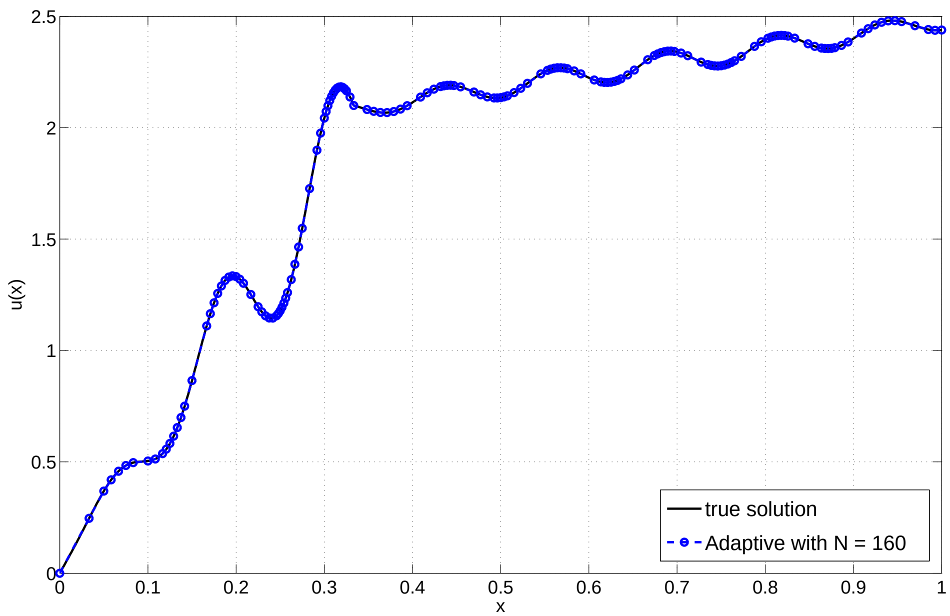

Figure 8: Adaptively solved, numerical solution of **case (b)** with a node fixed at discontinuity point and local error of $\mathrm{TOL}_E = 0.05^2$. Number of elemenents for this case is $N = 424$.

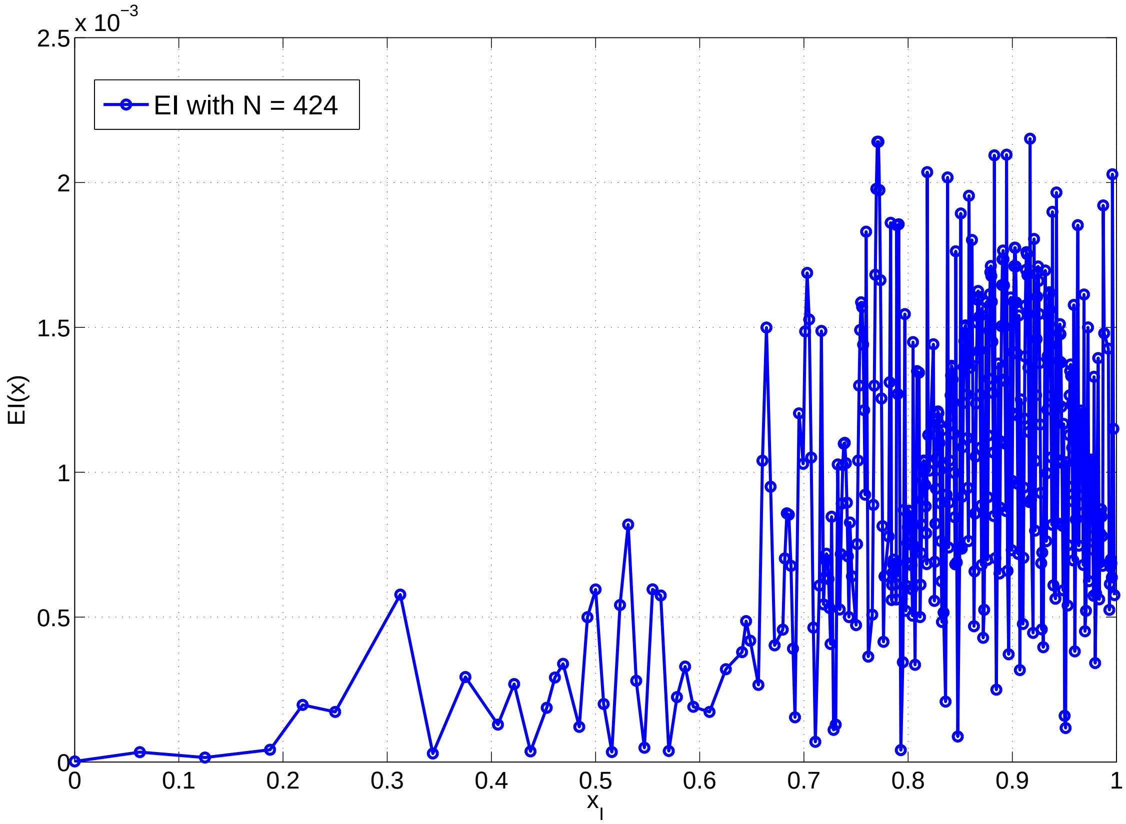

Figure 9: Local error of EI versus first node of elements for **case (a)**. The solution in this case is adaptively solved and number of elements is $N = 424$

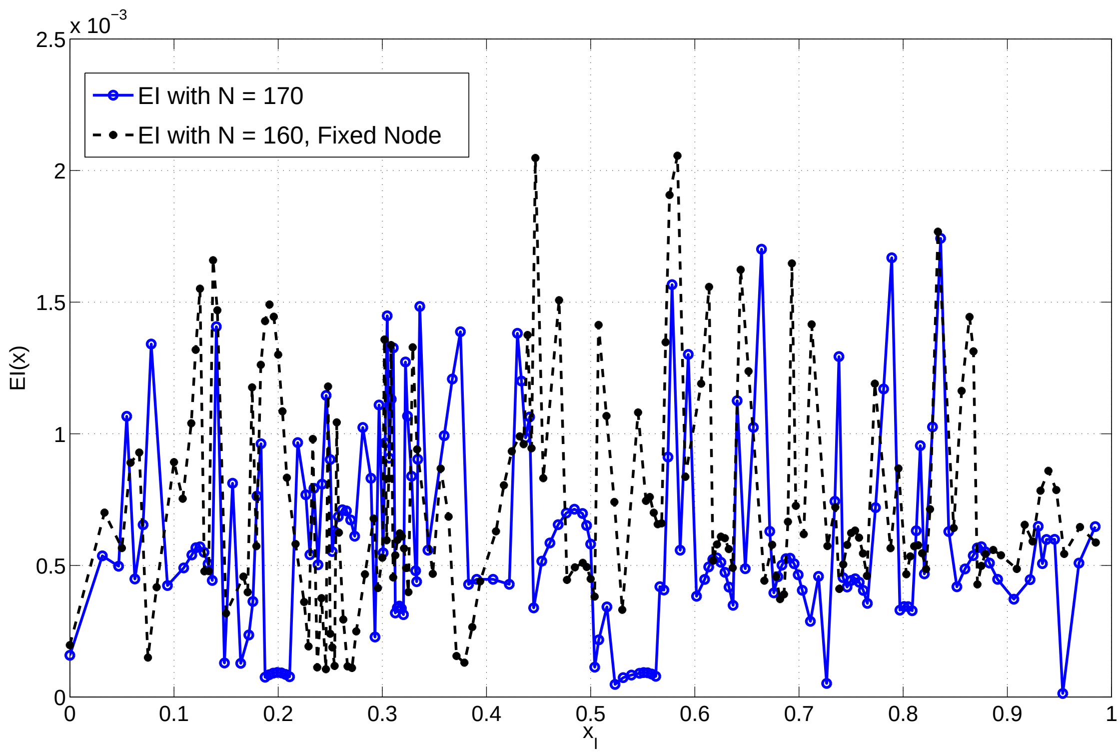

Figure 10: Local error of EI versus first node of elements for **case (b)**. The solution in this case is adaptively solved and number of elements is $N = 170$ for equally spaced elements and $N = 160$ for the case where a node is fixed at the discontinuity point