UNIVERSITY OF COLORADO BOULDER

MCEN 5228: COMPUTATIONAL FLUID DYNAMICS

One-Dimensional Finite Element Coding Project

Author:

Lara BURI

March 23, 2021

I. Template Boundary Value Problem

For this project the following boundary value problem was used:

$$\frac{\partial^2 u}{\partial x^2} = f(x)\ \forall\ x \in [a, b]$$

$$u = u_a\ at\ x = a$$

$$u = u_b\ at\ x = b$$

$$f(x) = k^2 \cos\left(\frac{\pi k x}{L}\right) + \alpha(1 - k^2) \sin\left(\frac{2\pi k x}{L}\right)$$

Where $a = 0.$ $b = 1$, $L = 1.0$, $u_a = 0.0$, $u_b = 1.0$, and $\alpha = 5.0$.

A. Exact Solution

The pages below present the analytical solution for $u(x)$ through the integration of the $f(x)$ equation from the template boundary value problem. This solution will be used to compare the FEM simulation results.

Part 3a

Sunday, February 28, 2021 12:35 PM

Obtain exact solution for this problem.

$$\frac{\partial^2 u}{\partial x^2} = k^2 \cos\left(\frac{\pi k x}{L}\right) + \alpha(1-k^2)\sin\left(\frac{2\pi k x}{L}\right)$$

B.C.s: ① $x(0) = 0$
② $x(1) = 1$

Integrate Twice to get $u(x)$:

$$k^2 \int \cos\left(\frac{\pi k x}{L}\right) dx + \alpha(1-k^2) \int \sin\left(\frac{2\pi k x}{L}\right) dx$$

$$\frac{k^2 L}{\pi k} \sin\left(\frac{\pi k x}{L}\right) - \frac{\alpha(1-k^2)L}{2\pi k} \cos\left(\frac{2\pi k x}{L}\right) + C_1$$

$$\frac{k^2 L}{\pi k} \int \sin\left(\frac{\pi k x}{L}\right) dx - \frac{\alpha(1-k^2)L}{2\pi k} \int \cos\left(\frac{2\pi k x}{L}\right) dx + C_1 \int dx$$

$$-\frac{k^2 L}{\pi k} \frac{L}{\pi k} \cos\left(\frac{\pi k x}{L}\right) - \frac{\alpha(1-k^2)L}{2\pi k} \frac{L}{2\pi k} \sin\left(\frac{2\pi k x}{L}\right) + C_1 x + C_2$$

Part 3a

Sunday, March 14, 2021 9:41 AM

Apply Boundary Conditions:

① $x(0) = 0$

$$0 = -\frac{L^2}{\pi^2}\cos(0) - \cancelto{0}{\frac{\alpha(1-k^2)L^2}{4\pi^2 k^2}\sin(0)} + \cancelto{0}{C_1(0)} + C_2$$

$$0 = \frac{-L^2}{\pi^2} + C_2$$

$$\boxed{C_2 = \frac{L^2}{\pi^2}}$$

② $x(1) = 1$

$$1 = \frac{-L^2}{\pi^2}\cos\left(\frac{\pi k}{L}\right) - \frac{\alpha(1-k^2)L^2}{4\pi^2 k^2}\sin\left(\frac{2\pi k}{L}\right) + C_1 + \frac{L^2}{\pi^2}$$

$$C_1 = 1 - \frac{L^2}{\pi^2} + \frac{\alpha(1-k^2)L^2}{4\pi^2 k^2}\sin\left(\frac{2\pi k}{L}\right) + \frac{L^2}{\pi^2}\cos\left(\frac{\pi k}{L}\right)$$

Final Equation:

$$u(x) = -\frac{k^2 L}{\pi k}\,\frac{L}{\pi k}\cos\left(\frac{\pi k x}{L}\right) - \frac{\alpha(1-k^2)L}{2\pi k}\,\frac{L}{2\pi k}\sin\left(\frac{2\pi k x}{L}\right)$$

$$+ \left[1 - \frac{L^2}{\pi^2} + \frac{\alpha(1-k^2)L^2}{4\pi^2 k^2}\sin\left(\frac{2\pi k}{L}\right) + \frac{L^2}{\pi^2}\cos\left(\frac{\pi k}{L}\right)\right]x + \frac{L^2}{\pi^2}$$

Part 3a

Sunday, March 14, 2021 9:49 AM

$$u(x) = \frac{-L^2}{\pi^2}\left[\cos\left(\frac{\pi k x}{L}\right) - \frac{\alpha(1-k^2)}{4k^2}\sin\left(\frac{2\pi k x}{L}\right)\right]$$

$$+\frac{L^2}{\pi^2}x\left[\frac{\pi^2}{L^2} - 1 + \frac{\alpha(1-k^2)}{4k^2}\sin\left(\frac{2\pi k}{L}\right) + \cos\left(\frac{\pi k}{L}\right)\right] + \frac{L^2}{\pi^2}$$

$* \; L = 1, \quad \alpha = 5$

$$\boxed{\begin{aligned} u(x) &= -\frac{1}{\pi^2}\left[\cos(\pi k x) - \frac{5(1-k^2)}{4k^2}\sin(2\pi k x)\right] \\ &+ \frac{1}{\pi^2}x\left[\pi^2 - 1 + \frac{5(1-k^2)}{4k^2}\sin(2\pi k) + \cos(\pi k)\right] + \frac{1}{\pi^2} \end{aligned}}$$

B. Pseudocode

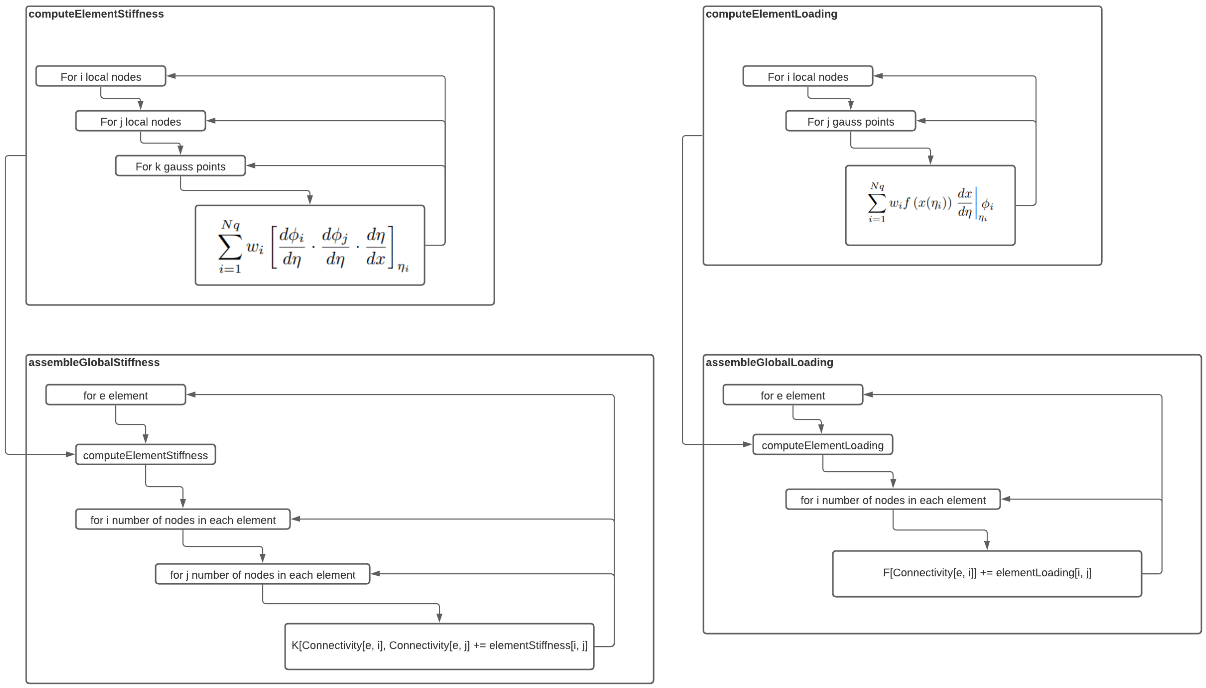
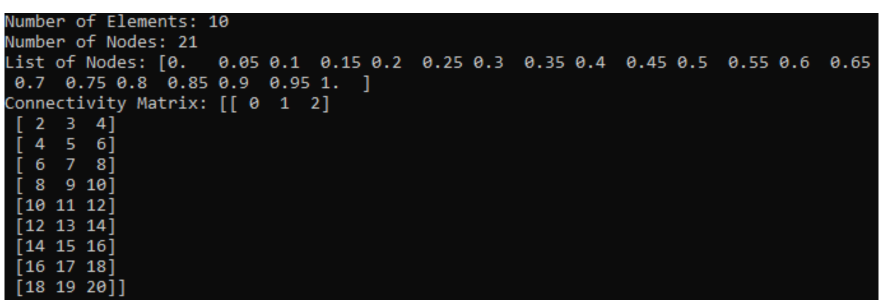


Fig. 1 Algorithm for Computing Element and Global Stiffness Matrix and Loading Vector

II. Programming 1D Meshing and Mesh Data-Structures

Using the functions *generateMeshNodes* and *generateMeshConnectivity*, mesh information can be obtained with the inputs of the domain interval, the polynomial order, and the mesh size. For benchmarking these functions, the following test cases were used with results shown below:

1) Domain Interval: [0.0, 1.0]; Polynomial Order: 2; Mesh Size: 0.1



```
Number of Elements: 10
Number of Nodes: 21
List of Nodes: [0.   0.05 0.1  0.15 0.2  0.25 0.3  0.35 0.4  0.45 0.5  0.55 0.6  0.65
 0.7  0.75 0.8  0.85 0.9  0.95 1.  ]
Connectivity Matrix: [[ 0  1  2]
 [ 2  3  4]
 [ 4  5  6]
 [ 6  7  8]
 [ 8  9 10]
 [10 11 12]
 [12 13 14]
 [14 15 16]
 [16 17 18]
 [18 19 20]]
```

2) Domain Interval: [0.0, 1.0]; Polynomial Order: 1; Mesh Size: 0.1

```
Number of Elements: 10
Number of Nodes: 11
List of Nodes: [0.  0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1. ]
Connectivity Matrix: [[ 0  1]
 [ 1  2]
 [ 2  3]
 [ 3  4]
 [ 4  5]
 [ 5  6]
 [ 6  7]
 [ 7  8]
 [ 8  9]
 [ 9 10]]
```

3) Domain Interval: [0.0, 1.0]; Polynomial Order: 3; Mesh Size: 0.1

```
Number of Elements: 10
Number of Nodes: 31
List of Nodes: [0.         0.03333333 0.06666667 0.1        0.13333333 0.16666667
 0.2        0.23333333 0.26666667 0.3        0.33333333 0.36666667
 0.4        0.43333333 0.46666667 0.5        0.53333333 0.56666667
 0.6        0.63333333 0.66666667 0.7        0.73333333 0.76666667
 0.8        0.83333333 0.86666667 0.9        0.93333333 0.96666667
 1.        ]
Connectivity Matrix: [[ 0  1  2  3]
 [ 3  4  5  6]
 [ 6  7  8  9]
 [ 9 10 11 12]
 [12 13 14 15]
 [15 16 17 18]
 [18 19 20 21]
 [21 22 23 24]
 [24 25 26 27]
 [27 28 29 30]]
```

III. Programming Isoparametric 1D Elements

Using the functions *lineElementShapeFunction* and *lineElementShapeDerivatives*, plots of the shape functions and the shape function derivatives can be generated for an element. These plots can be used for benchmarking to ensure the calculations are correct for this section. Below are the results for four test cases.

1) 2-Node Linear Element Shape Functions and Derivatives

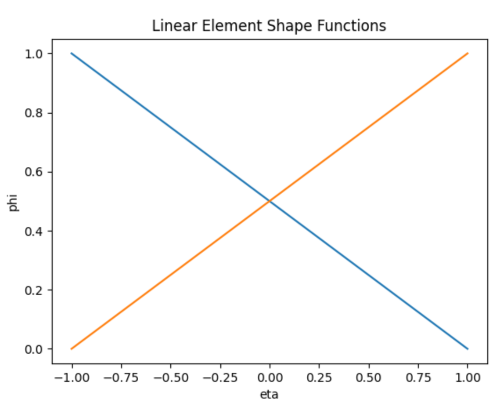


Fig. 2 Linear Element Shape Functions

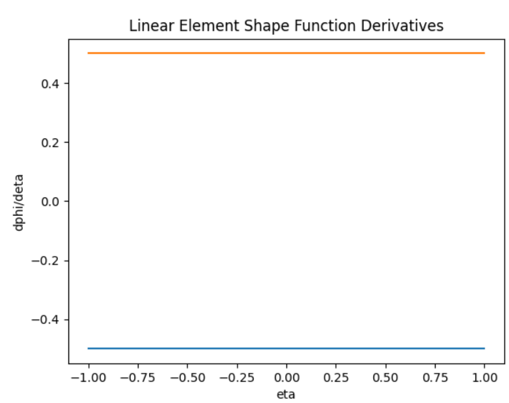
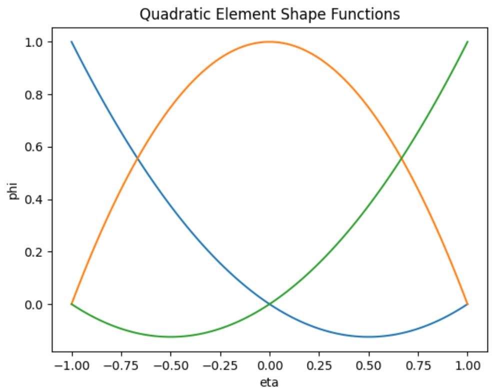


Fig. 3 Linear Element Shape Function Derivatives

2) 3-Node Quadratic Element Shape Functions and Derivatives



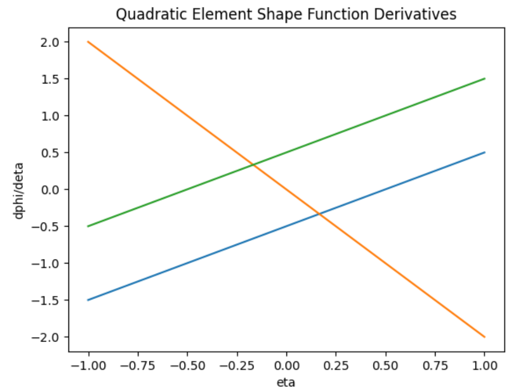
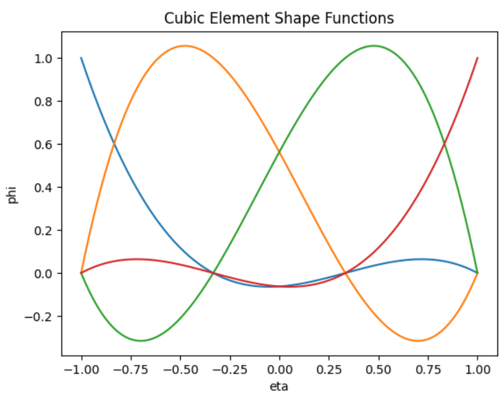


Fig. 4 Quadratic Element Shape Functions

Fig. 5 Quadratic Element Shape Function Derivatives

3) 4-Node Cubic Element Shape Functions and Derivatives



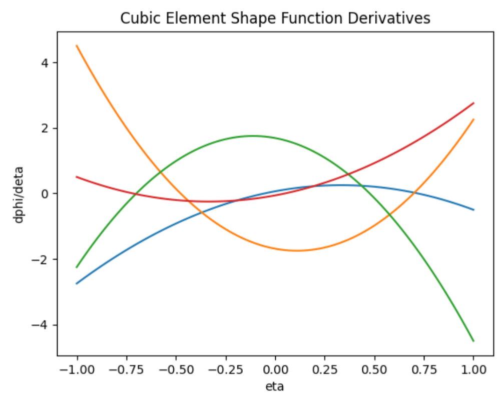
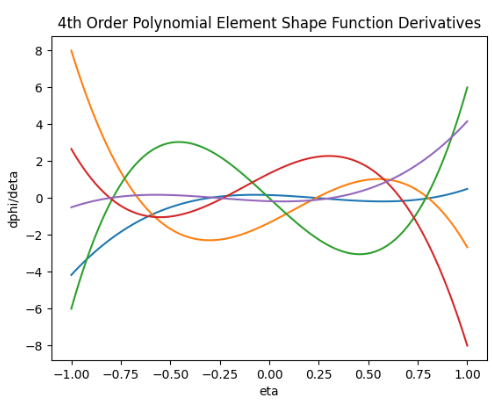


Fig. 6 Cubic Element Shape Functions

Fig. 7 Cubic Element Shape Function Derivatives

4) 5-Node 4th Order Polynomial Element Shape Functions and Derivatives



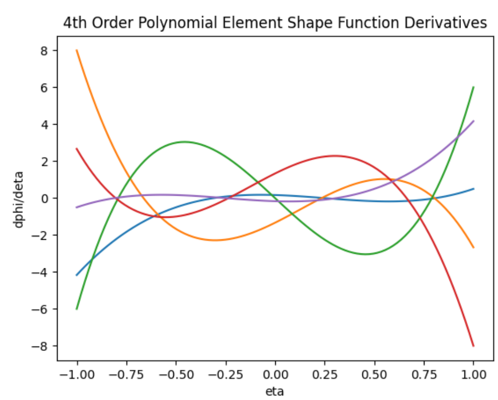


Fig. 8 4th Order Element Shape Functions

Fig. 9 4th Order Shape Function Derivatives

IV. Programming the Local Evaluation of Element Matrices

Using the functions *computeElementStiffness* and *computeElementLoading*, element stiffness and loading vectors are generated using a test function $f(x) = x$. The following benchmarking cases were used to test these functions:

1) Domain Interval: [0.0, 1.0]; Polynomial Order: 1; Mesh Size: 0.1

```
kij= [[ 10. -10.]
 [-10.  10.]]
fi= [0.0025 0.0025]
```

2) Domain Interval: [0.0, 1.0]; Polynomial Order: 2; Mesh Size: 0.1

```
kij= [[ 23.3333209 -26.6666418   3.3333209]
 [-26.6666418  53.3332836 -26.6666418]
 [  3.3333209 -26.6666418  23.3333209]]
fi= [0.         0.00333333 0.00166667]
```

V. Programming the Finite Element Assembly for the Given Boundary Value Problem

Using the functions *assembleFlobalStiffness* and *assembleGlobalLoading*, the global stiffness matrix and global loading vector can be computed for a given polynomial order and mesh size. The following test cases were used with the results shown below:

1) Domain Interval: [0.0, 1.0]; Polynomial Order: 1; Mesh Size: 0.5

```
K= [[ 2. -2.  0.]
 [-2.  4. -2.]
 [ 0. -2.  2.]]
F= [0.0625 0.25   0.1875]
```

2) Domain Interval: [0.0, 1.0]; Polynomial Order: 2; Mesh Size: 0.3

```
K= [[ 6.99999627 -7.99999254  0.99999627  0.          0.          0.
   0.        ]
 [-7.99999254 15.99998508 -7.99999254  0.          0.          0.
   0.        ]
 [ 0.99999627 -7.99999254 13.99999254 -7.99999254  0.99999627  0.
   0.        ]
 [ 0.          0.         -7.99999254 15.99998508 -7.99999254  0.
   0.        ]
 [ 0.          0.          0.99999627 -7.99999254 13.99999254 -7.99999254
   0.99999627]
 [ 0.          0.          0.          0.         -7.99999254 15.99998508
  -7.99999254]
 [ 0.          0.          0.          0.          0.99999627 -7.99999254
   6.99999627]]
F= [2.60208521e-18 3.70370543e-02 3.70370025e-02 1.11111163e-01
 7.40740050e-02 1.85185272e-01 5.55555038e-02]
```

3) Domain Interval: [0.0, 1.0]; Polynomial Order: 3; Mesh Size: 0.1

```
K= [[ 10. -10.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [-10.  20. -10.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  0. -10.  20. -10.   0.   0.   0.   0.   0.   0.   0.]
 [  0.   0. -10.  20. -10.   0.   0.   0.   0.   0.   0.]
 [  0.   0.   0. -10.  20. -10.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0. -10.  20. -10.   0.   0.   0.   0.]
 [  0.   0.   0.   0.   0. -10.  20. -10.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0. -10.  20. -10.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0. -10.  20. -10.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0. -10.  20. -10.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0. -10.  10.]]
F= [0.0025 0.01   0.02   0.03   0.04   0.05   0.06   0.07   0.08   0.09
 0.0475]
```

VI. Solution and Analysis Using the Template Boundary Value Problem

Table 1 shows the error from comparing the numerical and analytical solutions across the entire domain. The error was calculated for varying k values at different mesh sizes to see how the k value and mesh size affects the solution. It is evident that the error generally decreases as the mesh size decreases. It can also be seen that increasing the k value will cause the error to be overall higher, even though it decreases with each decreasing mesh size.

Table 1 Error Values for Varying k Values and Mesh Sizes

k Value	Mesh Size h	Error
2	0.1	0.04762807
	0.05	0.0215928
	0.01	0.00419912
	0.005	0.00209773
4	0.1	0.33656387
	0.05	0.11734492
	0.01	0.03068371
	0.005	0.01030220
8	0.1	12.1181614
	0.05	0.70678412
	0.01	0.08758067
	0.005	0.04330249
16	0.1	15.9997129
	0.05	24.5248506
	0.01	0.37719286
	0.005	0.17709698
32	0.1	185.093484
	0.05	31.9994257
	0.01	1.95187213
	0.005	0.75645417

The following groups of figures present a graphical analysis of the error analysis shown in Table 1. Each group is a set of solutions for one k value for each mesh size as presented in the table.

Figures 10, 11, 12, and 13 show a comparison of the exact solution to the FEM solution for a k value of 2.0. It is evident that even with larger mesh sizes, the exact solution and the FEM solution agree well, though the solution is not as well resolved at the highest mesh size.

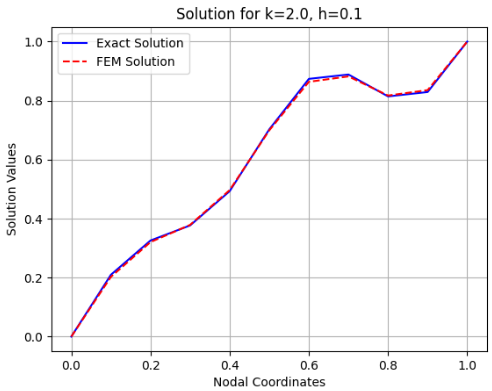


Fig. 10 **k = 2.0, h = 0.1**

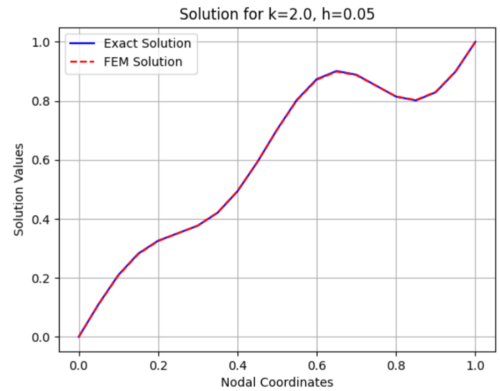


Fig. 11 **k = 2.0, h = 0.05**

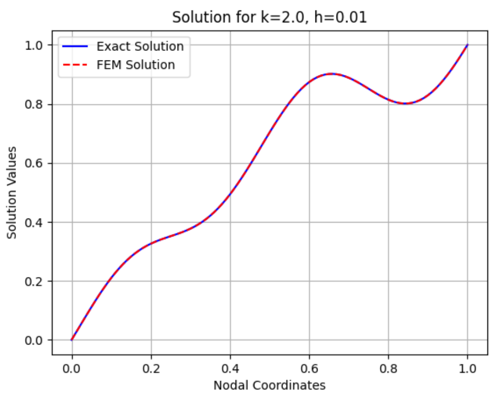


Fig. 12 **k = 2.0, h = 0.01**

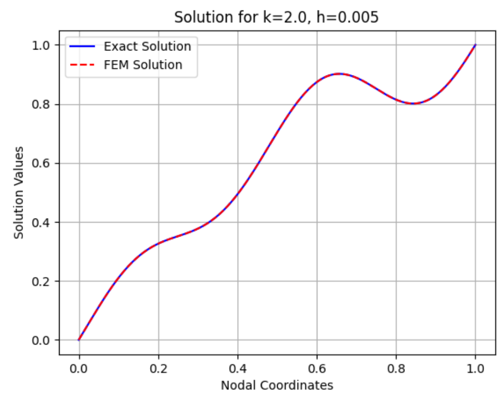


Fig. 13 **k = 2.0, h = 0.005**

Figures 14, 15, 16, and 17 show a comparison of the exact solution to the FEM solution for a k value of 4.0. At this k value, there is less agreement between solutions for larger mesh sizes. This agrees with the error analysis in Table 1, for a k value of four, the error is much higher at a mesh size of 0.1.

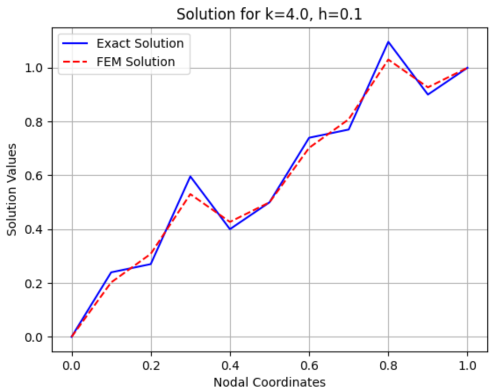


Fig. 14 k = 4.0, h = 0.1

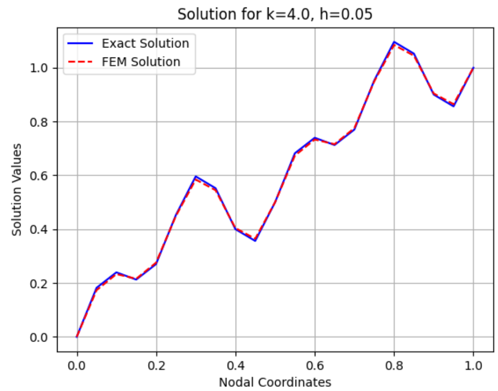


Fig. 15 k = 4.0, h = 0.05

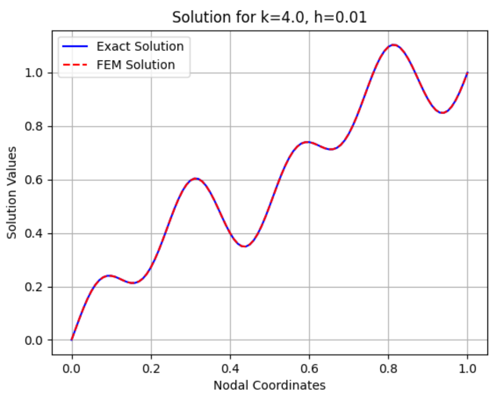


Fig. 16 k = 4.0, h = 0.01

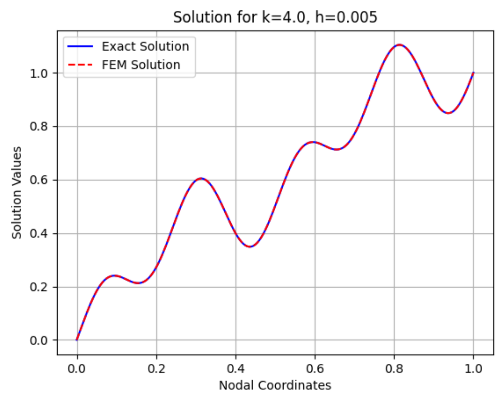


Fig. 17 k = 4.0, h = 0.005

Figures 18, 19, 20, and 21 show a comparison of the exact solution to the FEM solution for a k value of 8.0. For this k value, it is evident that the solution breaks down at a mesh value of 0.1. The error value for this mesh size is very high and there is no similarity of the FEM solution to the exact solution. For a mesh size of 0.05, the solutions agree better but are still not well resolved over the mesh.

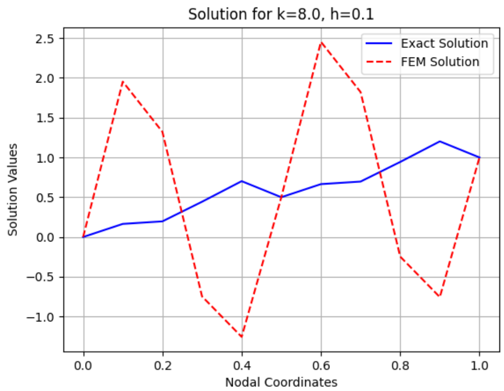


Fig. 18   k = 8.0, h = 0.1

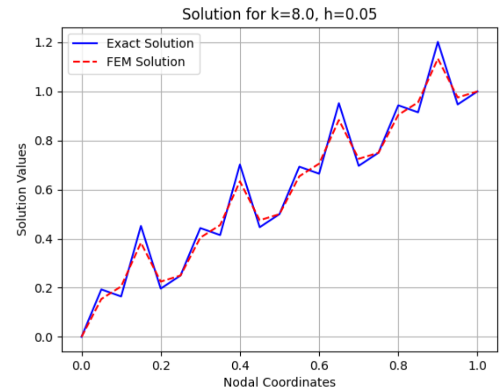


Fig. 19   k = 8.0, h = 0.05

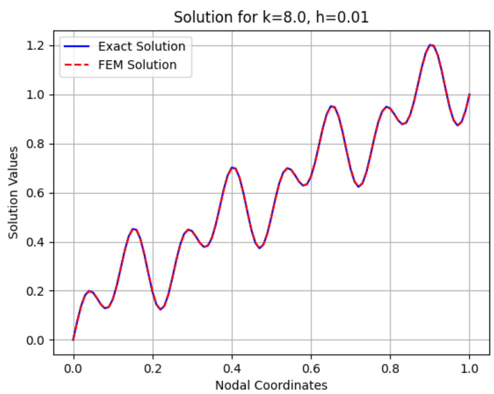


Fig. 20   k = 8.0, h = 0.01

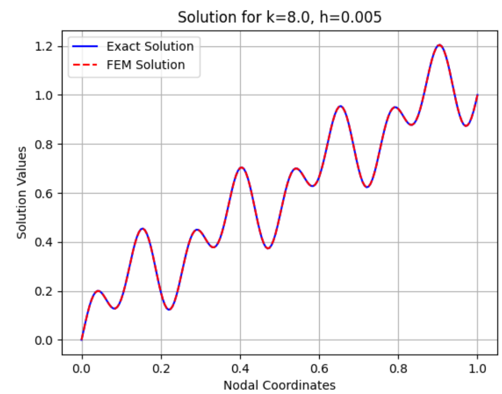


Fig. 21   k = 8.0, h = 0.005

Figures 22, 23, 24, and 25 show a comparison of the exact solution to the FEM solution for a k value of 16.0. At this k value, the solutions for both mesh sizes 0.1 and 0.05 break down. The error values are very high and the FEM solutions have no resemblance to the exact solutions.

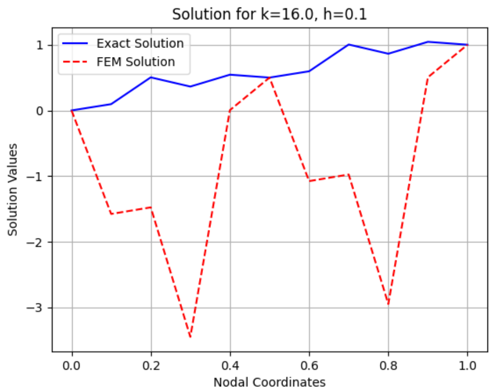


Fig. 22 k = 16.0, h = 0.1

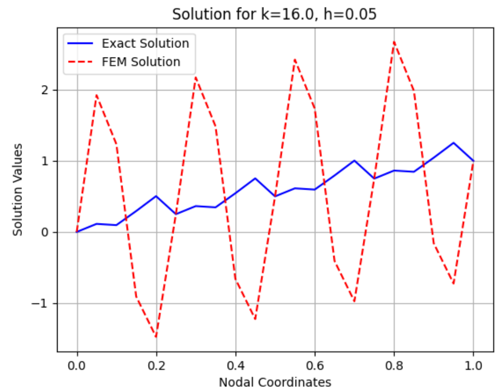


Fig. 23 k = 16.0, h = 0.05

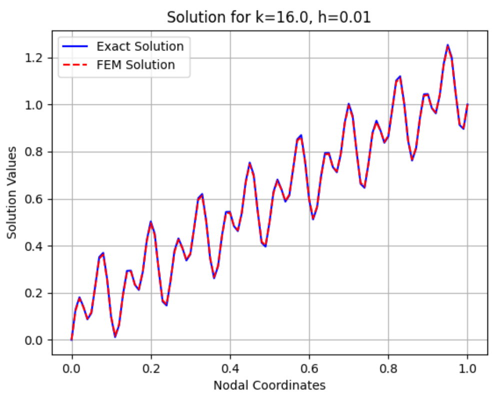


Fig. 24 k = 16.0, h = 0.01

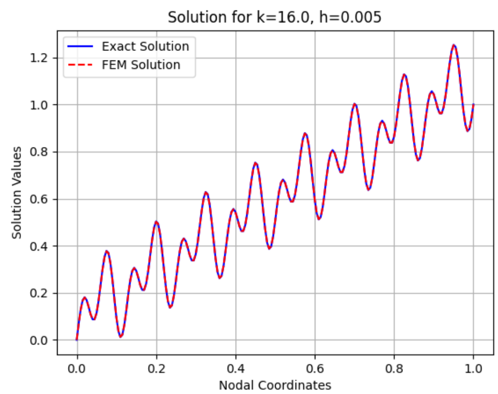


Fig. 25 k = 16.0, h = 0.005

Finally, Figures 26, 27, 28, and 29 show a comparison of the exact solution to the FEM solution for a k value of 32.0. Again, for mesh sizes of 0.1 and 0.05, the FEM solution does not resemble the exact solution and there are very high error values. The error values overall for this k value are higher than all the other error values for the other cases presented in Table 1.

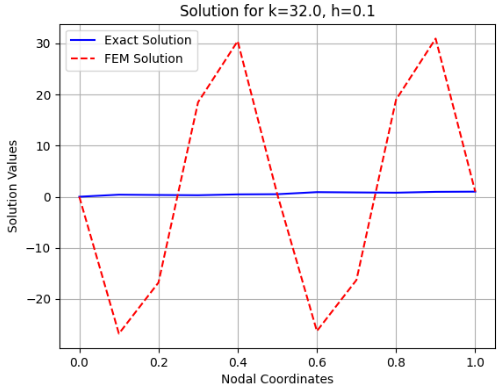


Fig. 26 k = 32.0, h = 0.1

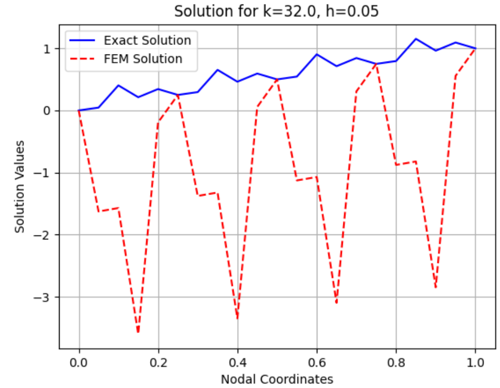


Fig. 27 k = 32.0, h = 0.05

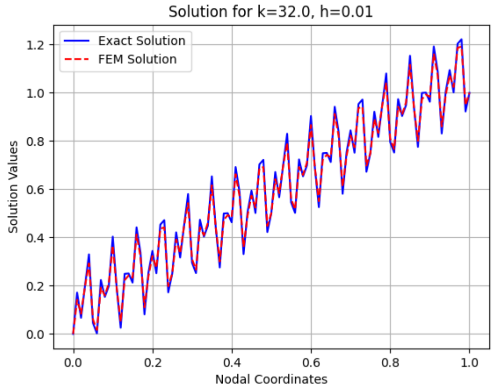


Fig. 28 k = 32.0, h = 0.01

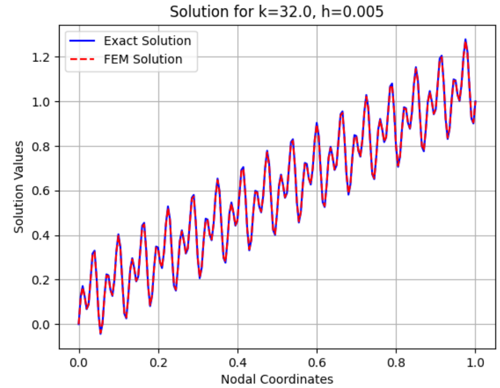


Fig. 29 k = 32.0, h = 0.005