Finite Element Methods

Project report

Presented to: **Dr. Wassim Habchi**

Student:
Youssef Jammal

Apply an energy balance to an infinitely small element of the wall and derive the PDE governing this problem and its corresponding boundary conditions

The governing equation to the problem is

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)+Q=\rho C_p\frac{\partial T}{\partial t}$$

Simplifications

1) 1D problem along the x-axis
2) No heat generation
3) Steady state

The equation becomes

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right)=0$$

The boundary conditions

1) At x=0, the temperature is 15 (T(0)=15)
2) At x=0.4, the heat flux is at -10 (q(0.4)=-10)

Deriving the weak Galerkin formulation

$$\int_{xa}^{xb}\left[-k\frac{dT}{dx}\frac{dw}{dx}\right]d\mathrm{x}=\mathrm{w}(x_b)q_2-\mathrm{w}(x_a)q_1$$

Elementary equations

Derive the solution approximation

By using:

- $T_h^e=\sum_{j=1}^{n}T_j^e\,\Psi_j^e$
- $x_b-x_a=he$
- $\bar{x}=x-x_a$

$$k_{ij}^e=\int_0^{he}\left[-k(\bar{x}+x_a)\frac{d\Psi_j^e}{d\bar{x}}\frac{d\Psi_i^e}{d\bar{x}}\right]d\bar{x}$$

$$Q_i=q_2-q_1$$

Derive the shape functions and their derivatives with respect to $\bar{x}$

Shape functions	Derivatives of shape functions
$\Psi_1 = -\frac{9\,\bar{x}^3}{2\,he^3} + \frac{9\,\bar{x}^2}{he^2} - \frac{11\,\bar{x}}{2\,he} + 1$	$\frac{\mathrm{d}}{\mathrm{dx}}\Psi_1 = -\frac{27\,\bar{x}^2}{2\,he^3} + \frac{18\,\bar{x}}{he^2} - \frac{11}{2\,he}$
$\Psi_2 = \frac{27\,\bar{x}^3}{2\,he^3} - \frac{45\,\bar{x}^2}{2\,he^2} + \frac{9\,\bar{x}}{he}$	$\frac{\mathrm{d}}{\mathrm{dx}}\Psi_2 = \frac{81\,\bar{x}^2}{2\,he^3} - \frac{45\,\bar{x}}{he^2} + \frac{9}{he}$
$\Psi_3 = -\frac{27\,\bar{x}^3}{2\,he^3} + \frac{18\,\bar{x}^2}{he^2} - \frac{9\,\bar{x}}{2\,he}$	$\frac{\mathrm{d}}{\mathrm{dx}}\Psi_3 = -\frac{81\,\bar{x}^2}{2\,he^3} + \frac{45\,\bar{x}}{he^2} - \frac{9}{2\,he}$
$\Psi_4 = \frac{9\,\bar{x}^3}{2\,he^3} - \frac{9\,\bar{x}^2}{2\,he^2} + \frac{\bar{x}}{he}$	$\frac{\mathrm{d}}{\mathrm{dx}}\Psi_4 = \frac{27\,\bar{x}^2}{2\,he^3} - \frac{9\,\bar{x}}{he^2} + \frac{1}{he}$

Gauss Quadrature

- By replacing $\bar{x}$ by $\left(\frac{\xi+l}{2}\right)he$ in $G(\bar{x}) = -k(\bar{x}+x_a)\frac{d\Psi_j^e}{d\bar{x}}\frac{d\Psi_i^e}{d\bar{x}}$ and multiplying the result by $\frac{he}{2}$ we get $\hat{G}(\xi)$
- This is the equation used the in pre-processing $\sum_{i=1}^{N_{GP}}\hat{G}(\xi_i)W_i\,d\xi$
- The amount of Gauss points needed are 3, as the derivatives of the shape functions are of order 2, and the k is of a order 1, then the function G(*x)* is of order 5, $N_{GP} = (order\ of\ G(x)+1) \div 2 = 3$

N_{GP}	**Locations, ξ_i**	**Weights, W_i**
3	0.0000000000	0.8888888889
	±0.7745966692	0.5555555555
	±0.7745966692	0.5555555555

MATLAB and COMSOL comparison

- For MATLAB I used 1000 cubic elements as my input with 39001 post processing nodes.
- For COMSOL I used 39 000 elements in order to put it as a benchmark and compare.

Results

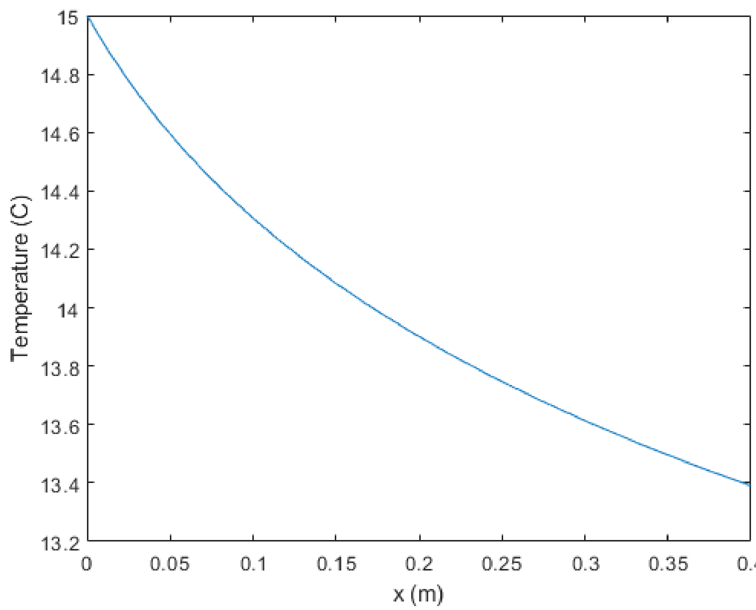

Figure 1: temperature variation in the wall (my MATLAB code)

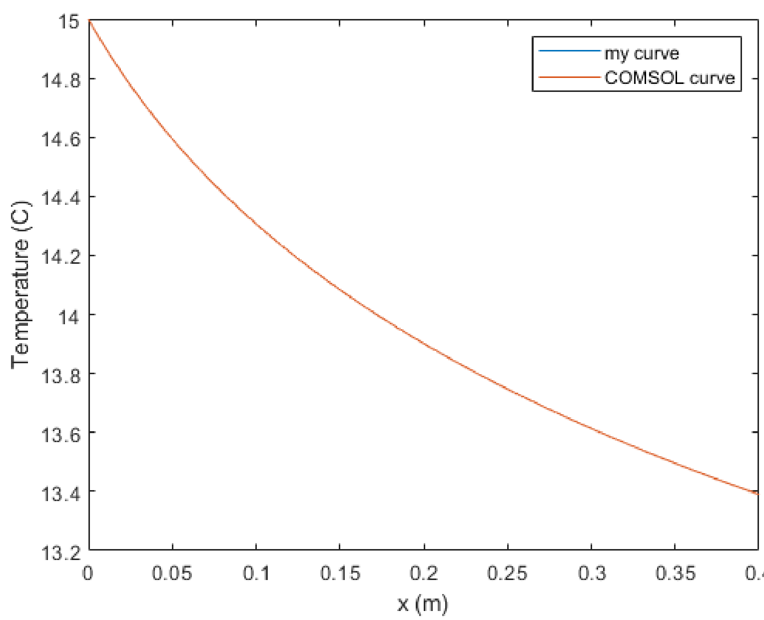

Figure 2: temperature variation in the wall (my MATLAB code versus COMSOL)

Convergence analysis:

- For MATLAB I used varying elements from 1 to 1000, All post processes are set at 39001 nodes to compare the solution and not the processing accuracy.
- For COMSOL I used 39 000 elements in order to put it as a benchmark and compare.

Results

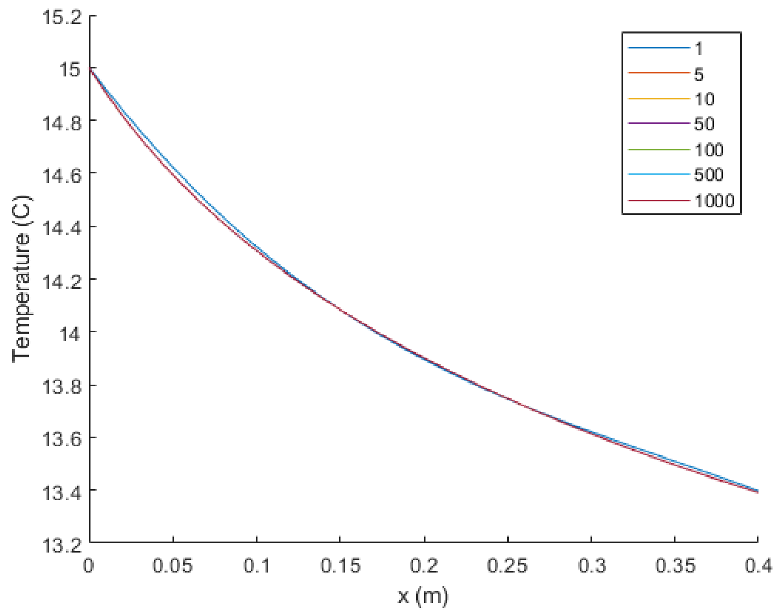

Figure 3: the convergence of the MATLAB solutions

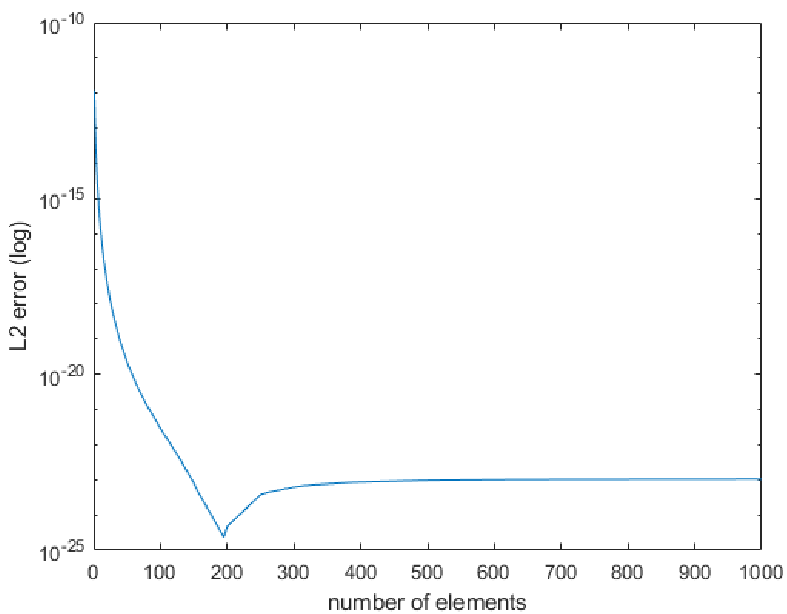

Figure 4: semi log graph of the L2 error between the COMSOL solution and the MATLAB derived solutions for various elements

Discussion

1. In the second figure it can be seen that the graph derived in MATLAB completely overlaps with the one derived in COMSOL. As the difference cannot be seen with the naked eye, I computed the errors L_1 = 2.7957e-08, L_2 = 1.0463e-23, and $L_{infinity}$ = 3.2346e-12. Looking at the L_2 error, we can see it is minimal at the order -23, an acceptable error would be of the order -6, now looking at the $L_{infinity}$ we can see it is of the order -12, which means even when analyzing in a conservative manner, the error is still much lower than the acceptable error.
2. 1000 elements can be considered extreme, and so I repeated the same process using MATLAB to analyze the convergence of the solution as the number of elements is added, the 3rd figure represents the convergence but with the same problem as the second, the convergence was not visible to the naked eye. Therefore, the 4th figure represents the error using 1 to 1000 with respect to the benchmarked COMSOL solution, and we can see the error goes down to the order of more or less -23 at which it starts increasing and decreasing, this random increase and decrease of the error is due to the machine's limiting precision being reached.