Finite Elements lab assignments

Martijn Papendrecht: 4369971

February 19, 2018

Contents

1 1D Assignment **3**
1.1 Theory 3
1.2 Modelling 3

2 2D Assignments **8**
2.1 Assignment 1 8

1 1D Assignment

1.1 Theory

Starting with the differential equation and boundary conditions

$$\begin{aligned} -D\frac{d^2u}{dx^2} + \lambda u = f(x), \\ -D\frac{du}{dx}(0) = 0, -D\frac{du}{dx}(1) = 0 \end{aligned} \tag{1}$$

we can derive the weak formulation.

$$\begin{aligned} \int_0^1 \left[-D\frac{d^2u}{dx^2} + \lambda u\right] \phi dx &= \int_0^1 f(x)\phi dx \\ \int_0^1 -D\left(\frac{d}{dx}\left(\frac{du}{dx}\phi\right) - \frac{d\phi}{dx}\frac{du}{dx}\right) + \lambda u\phi &= \int_0^1 f(x)\phi dx \\ -D\int_0^1 \frac{d}{dx}\left(\frac{du}{dx}\phi\right) dx + \int_0^1 D\frac{d\phi}{dx}\frac{du}{dx} + \lambda u\phi dx &= \int_0^1 f(x)\phi dx \\ \int_0^1 D\frac{d\phi}{dx}\frac{du}{dx} + \lambda u\phi dx &= \int_0^1 f(x)\phi dx \end{aligned} \tag{2}$$

Here ϕ is the test function satisfying the smoothness requirements. The boundary conditions are already containted in this weak formulation.

We can derive the Galerkin formulation by substitution $\phi = \phi_i$ and $u \approx u^N = \sum_{j=1}^N c_j\phi_j$ in eq. (2).

$$\begin{aligned} \int_0^1 D\frac{d\phi_i}{dx}\frac{d}{dx}\left(\sum_{j=1}^N c_j\phi_j\right) + \lambda\left(\sum_{j=1}^N c_j\phi_j\right)\phi_i dx &= \int_0^1 f(x)\phi_i dx \\ \sum_{j=1}^N c_j \int_0^1 D\frac{d\phi_i}{dx}\frac{d\phi_j}{dx} + \lambda\phi_i\phi_j dx &= \int_0^1 f(x)\phi_i dx \end{aligned} \tag{3}$$

And finally we can write in the form $S\vec{u} = \vec{f}$ where

$$S_{ij} = \int_0^1 D\frac{d\phi_i}{dx}\frac{d\phi_j}{dx} + \lambda\phi_i\phi_j dx \tag{4}$$

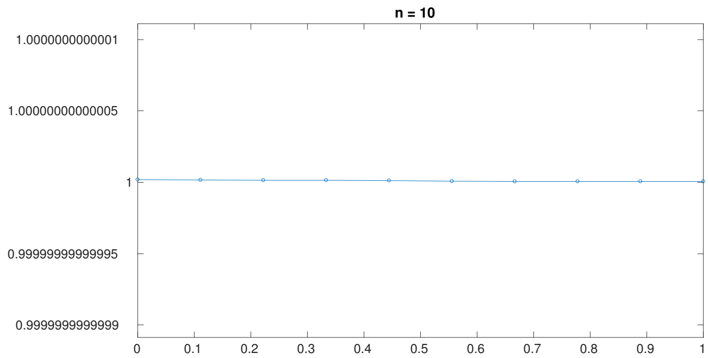
$$f_i = \int_0^1 f(x)\phi_i dx \tag{5}$$

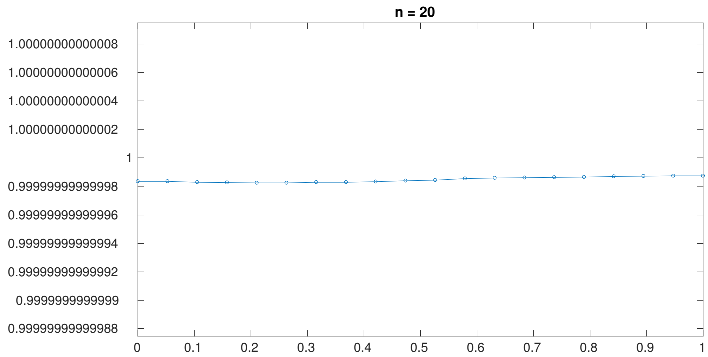
1.2 Modelling

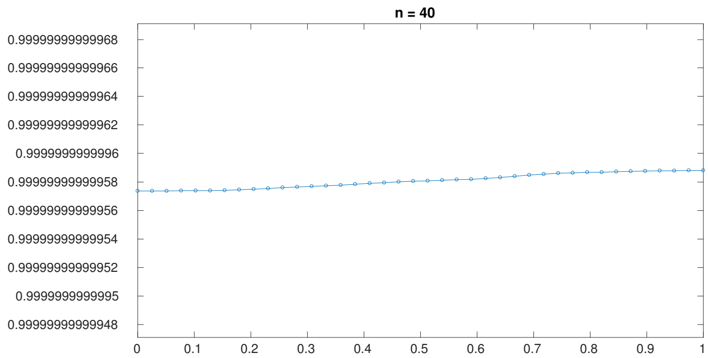
All figures are made using Matlab. The code is available at Github [1]. In these figures the elements aren't equally spaced as prescribed in the assignment. Every element has an additional random offset of $\pm\frac{1}{2n}$. This has been done to test the code without the assumption of equally spaced vertices.

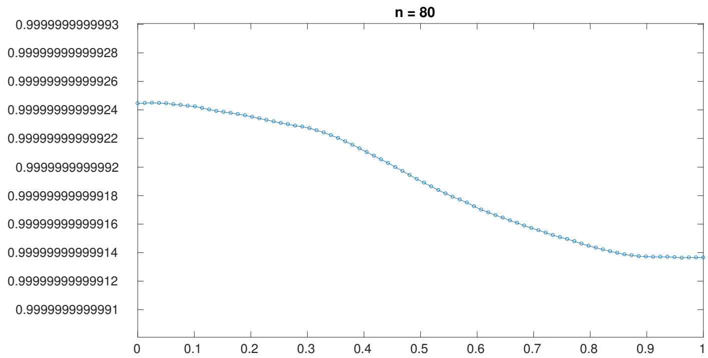
[1] https://github.com/MPapendrecht/FiniteElements.git

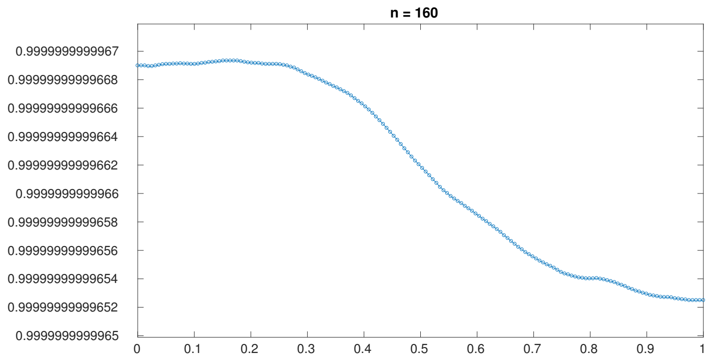
The figures below show the results for different number of vertices n. $\lambda = 1$ and $D = 1$ and the function $f(x) = 1$





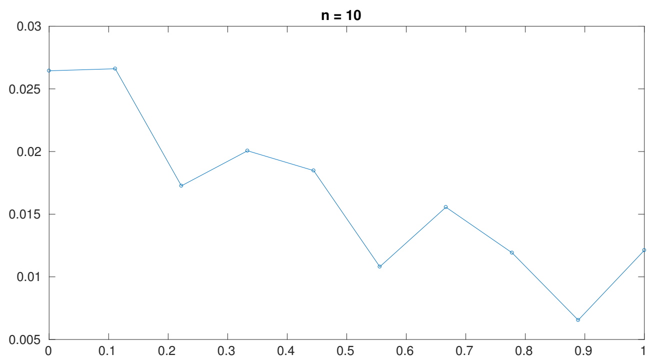


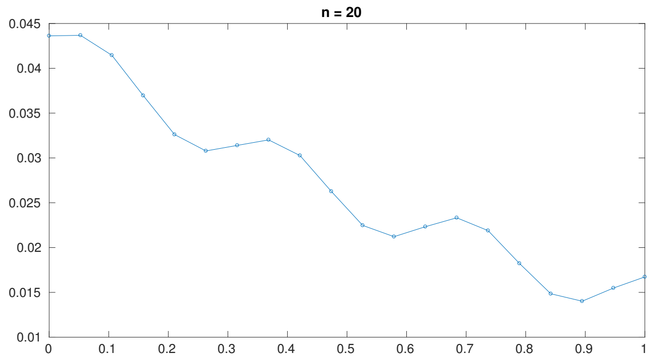


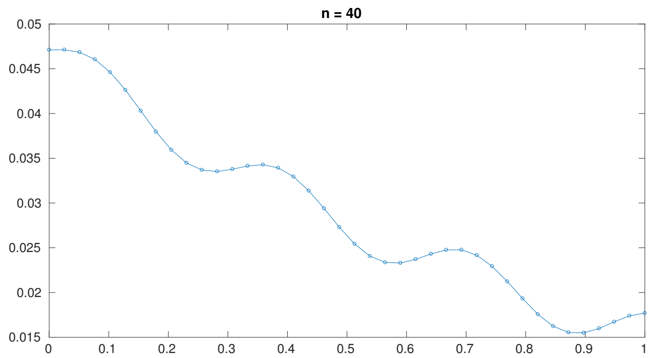


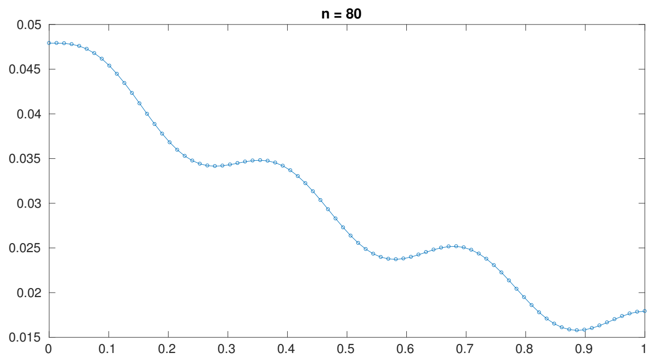
The solution is as expected very close to the analytical solution, and clearly satisfies the boundary conditions.

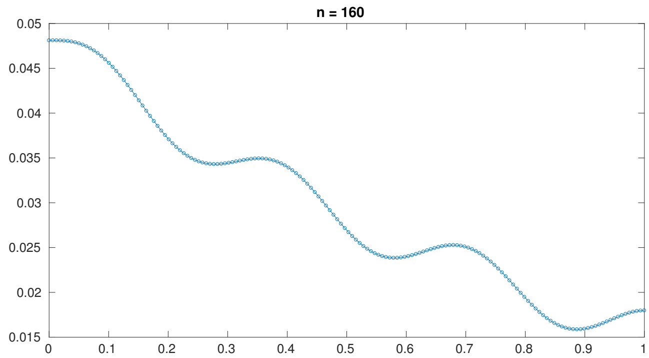
The figures below show the results for different number of vertices n as well. The same parameters are used, however the function $f(x) = sin(20x)$.



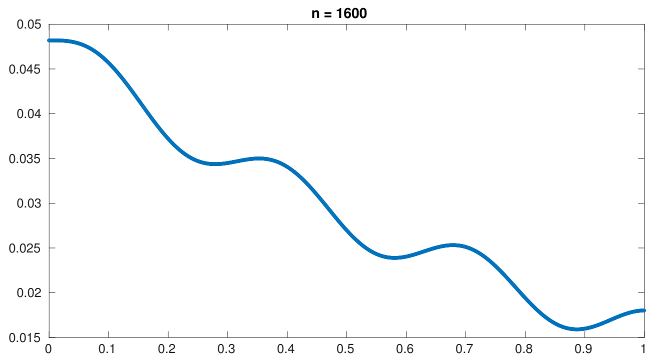

n = 20
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


n = 40
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


n = 80
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



The solution with $n = 10$ shows the boundary conditions aren't satisfied properly because the amount of vertices is too small. The other solutions satisfy the boundary conditions much better. Increasing n further doesn't seem to make the function improve much more, which can be seen in the figure below.



2 2D Assignments

2.1 Assignment 1