



**Universidad del País Vasco**

Facultad de Ciencia y Tecnología

# NSDE: Practice 8

## Group 2

Nerea Andia

Ander Cano

Mikel Lamela

Aitor Larrinoa

May 2021

# Finite element method for elliptic problems.

## 8.1 Galerkin method.

### Statement

We saw that for the model BVP (**??**) with **Dirichlet BC**

$$\begin{cases} -\left(k(x)\,u'(x)\right)' = f(x), & 0 < x < l, \\ u(0) = u(l) = 0 \end{cases} \tag{??}$$

if $u$ satisfies the strong form (**??**), then $u$ also satisfies the weak form (1), for all $v \in V = \{v \in \mathcal{C}^2[0,l] : v(0) = v(l) = 0\}$

$$\int_0^l k(x)\,u'(x)\,v'(x)\,dx = \int_0^l f(x)\,v(x)\,dx \tag{1}$$

If now we recognize that, with $(\cdot,\cdot)$ the $L^2$ inner product,

$$\int_0^l f(x)\,v(x)\,dx = (f,v)$$

then the weak form (1) of the model BVP (**??**) can be written as

$$\text{find} \quad u \in V \quad s.t. \quad a(u,v) = (f,v), \qquad \forall v \in V. \tag{2}$$

where we have defined the symmetric bilinear form $a(\cdot,\cdot)$ by

$$a(u,v) = \int_0^l k(x)\,u'(x)\,v'(x)\,dx.$$

The Galerkin idea is simple. Choose a finite-dimensional subspace $V_n$ of $V$, supposing that $\{\phi_1, \phi_1, \ldots, \phi_n\}$ is a basis for $V_n$, and reduce (2) to the subspace:

$$\text{find} \quad v_n \in V_n \quad s.t. \quad a(v_n, \phi_i) = (f, \phi_i), \qquad {\scriptstyle i=1,2,\ldots,n.} \tag{3}$$

$$v_n \in V_n \quad \Rightarrow \qquad v_n = \sum_{j=1}^{n} u_j\,\phi_j. \tag{4}$$

Finding $v_n \quad \Leftrightarrow \quad$ determining $u_1, u_2, \ldots, u_n$.

$$a\left(\sum_{j=1}^{n} u_j\,\phi_j, \phi_i\right) = \sum_{j=1}^{n} a(\phi_j, \phi_i)\,u_j = (f, \phi_i)\,, \qquad i = 1, 2, \ldots, n.$$

This is a system of $n$ linear equations for the unknowns $u_1, u_2, \ldots, u_n$. Defining a matrix $\mathbf{K} \in^{n\times n}$ and a vector $\mathbf{f} \in^{n}$ by

$$K_{ij} = a(\phi_j, \phi_i), \qquad f_i = (f, \phi_i)\,, \qquad i, j = 1, 2, \ldots, n,$$

finding $v_n$ is equivalent to solving the linear system

$$\mathbf{K}\,\mathbf{u} = \mathbf{f}.$$

Let $V_n \equiv F_N$ be the subspace of $V$ spanned by the orthogonal basis, w.r.t. the ordinary $L^2$ inner product on the interval $[0, 1]$:

$$\{\sin(\pi x), \sin(2\pi x), \ldots, \sin(N\pi x)\}.$$

Consider the BVP

$$\begin{cases} -\left((1+x)\,u'(x)\right)' = x, \quad 0 < x < 1, \\ u(0) = u(1) = 0 \end{cases} \tag{5}$$

((**??**) with $k(x) = (1 + x)$ and $f(x) = x$), exact solution: $u(x) = \frac{x}{2} - \frac{x^2}{4} - \frac{\ln(1+x)}{4\ln 2}$.

(a) Write a program in *Mathematica* that solves (5) using Galerkin method and compares the approximate solution with the exact one.

## Solution

First of all, we will solve the problem (5) using Galerkin method with n = 10. Then, we will compare the approximation with the exact solution.

Implementing the Galerkin method in *Mathematica*, we obtain a really good approximation. To see this clearly let´s plot first the comparison of both plots, where the two graphic shown practically equal.

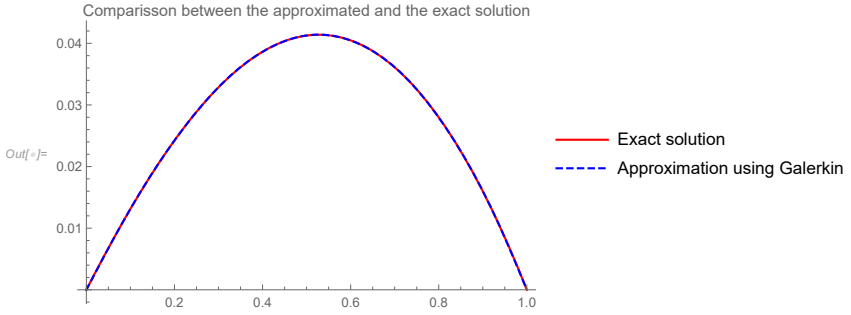


Figure 1: Comparison of the Galerkin Method and the exact solution

Note that the BC are satisfied, since at the boundary points the approximation is 0.

Secondly, we have computed the error to assure that the approximation is good. So, in the following figure are shown the graphic error and the discrete error.

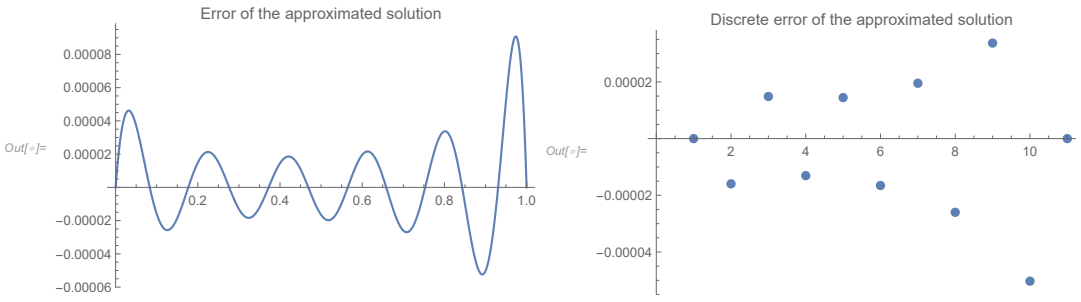


Figure 2: Error of the Galerkin method approximation

To finish, the error is quite small. Moreover, the maximum absolute value of the error is $5.0303 \cdot 10^{-15}$, so the Galerkin method apply to this problem can be consider as acceptable.

## Statement

We have also considered the simple BVP (6), that is (**??**) when $k(x) = 1$ and $l = 1$

$$\begin{cases} -u''(x) = f(x), & 0 < x < 1, \\ u(0) = u(1) = 0 \end{cases} \tag{6}$$

(b) Write a program in *Mathematica* that solves (6), using Galerkin method and with

$$f(x) = (3x + x^2)\,\exp(x) \tag{**??**}$$

and

$$f(x) = \sin(x)\,\exp(x). \tag{**??**}$$

Compare the approximate solutions with the ones obtained using FD method for elliptic problems.

## Solution

We have to solve the same problem for two different f(x). In both cases we will consider n = 10.

### f(x) = (3 x + x$^2$) exp(x)

First, let's see that the Boundary Condition are satisfied, seeing the approximation of the solution using Galerkin Method.

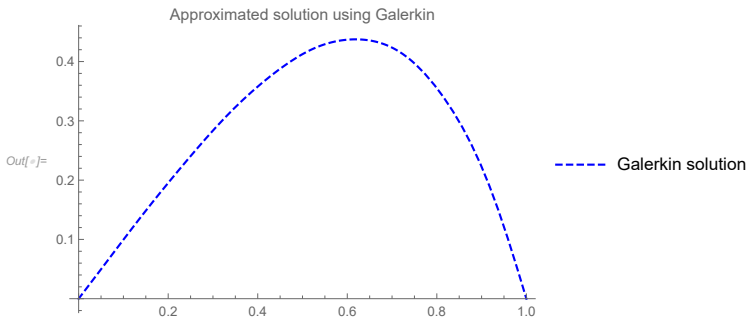


Figure 3: Approximated solution using Galerkin

Note that the boundary condition are clearly satisfied since they are 0. In order to have an idea of how good the approximation is, let's compare the Galerkin approximation with the exact solution as in the previous section.

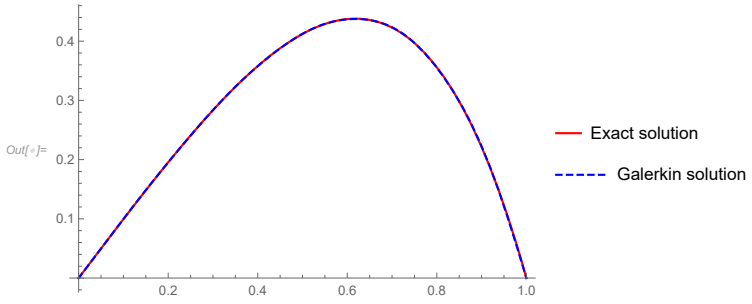


Figure 4: Comparison between the exact solution and the Galerkin approximation

The approximation is really good, now we compute the discrete error of the approximation. The maximum absolute error in this case is $1.2 \cdot 10^{-3}$. So, we expect that the error will be small.

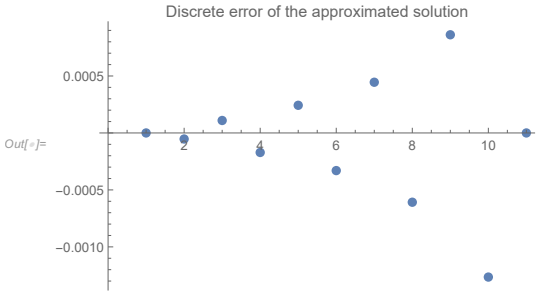


Figure 5: Discrete error of the approximated solution

To continue with, we will compare the Galerkin method with FD method for elliptic problems. We implement both methods in the *Mathematica* and we obtain the following comparison plot.

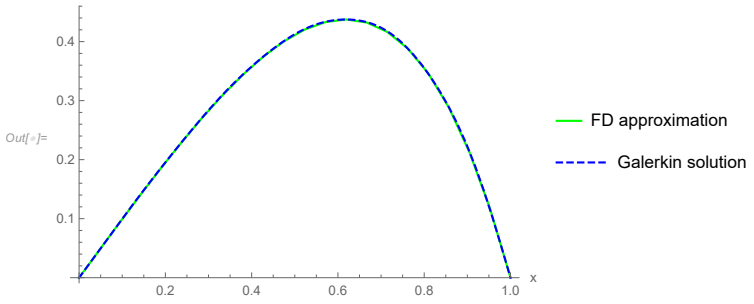


Figure 6: Caption

If we observe the plot in detail, we can appreciate that the FD approximation is lees smooth than the Galerkin one. However, the Galerkin method is much more slower than FD in *Mathematica*. In fact, the time obtained making use of Galerkin is 28.5 whereas the time obtained making use of FD approximation is 0.11.

### f(x) = sin(x) exp(x)

We will follow the same structure as in the previous f(x). First, we will see the Galerkin approximation.

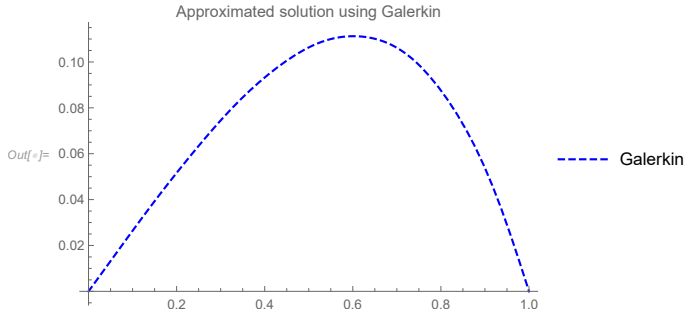


Figure 7: Caption

The boundary conditions are satisfied since they are zero in the boundary points of the approximation. As always, let's plot together the exact solution and the approximation.

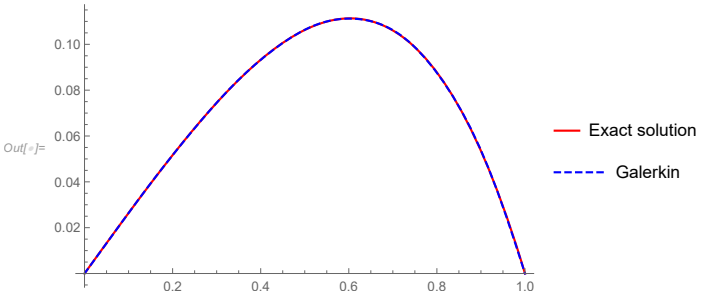


Figure 8: Caption

The solution is well approximated by the Galerkin Method. However, we compute the error to assure our conclusion.

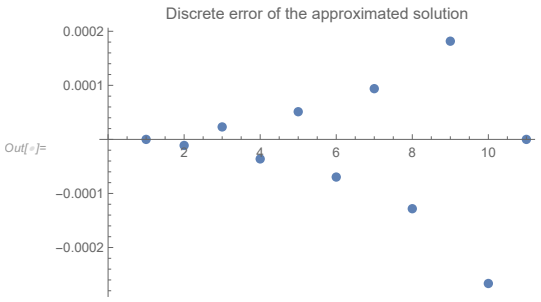


Figure 9: Caption

As the approximation goes to the right, the error gets bigger. Taking the maximum value in the right boundary, which a value of $2.66 \cdot 10^{-4}$. But, as it is small value it can be consider a good approximation.

Now, we will compare with FD method and see which one is better, taking into account the numerical approximation and the computational times. To begin, the comparison plot

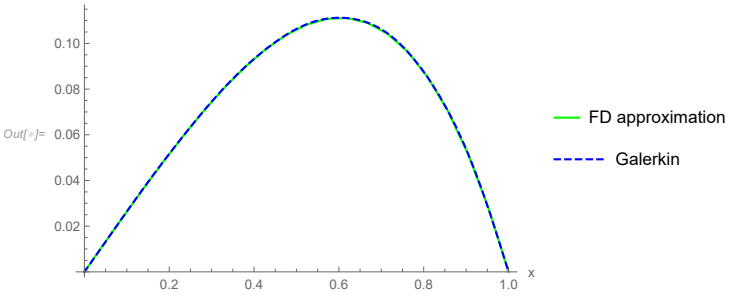


Figure 10: Caption

As we could expect after studying the previous f(x) function, the approximation are practically equal, but FD method is again faster than Galerkin Method. The time obtained with Galerkin is 37.359 and the one obtained with FD is 0.234.

## 8.2 Piecewise polynomials and FEM.

### Statement

**Finite element methods** use subspaces of piecewise polynomials; for simplicity, we will use only piecewise linear functions. A function $p : [0, l] \to$ is **piecewise linear** if, for each $i = 1, 2, \ldots, n$ there exist constants $a_i$, $b_i$ with

$$p(x) = a_i\, x + b_i, \quad \forall x \in (x_{i-1}, x_i).$$

We define, for a fixed mesh on $[0, l]$

$$S_n = \{p : [0, l] \to \ \mid p \text{ is continuous and piecewise lineal}, p(0) = p(l) = 0\}.$$

A piecewise linear function is completely determined by its *nodal values.*

For $n = 0, 1, 2, \ldots, n$, define $\phi_i$, the *tent* function, to be that piecewise linear function satisfying

$$\phi_i(x_j) = \begin{cases} 1, & j = i, \\ 0, & j \neq i. \end{cases} \tag{7}$$

The *support* of $\phi_i$ is the interval $[x_{i-1}, x_{i+1}]$ (for $i = 1, 2, \ldots, n-1$) and $[x_0, x_1]$ and $[x_{n-1}, x_n]$, when $i = 0$ and $i = n$, respectively.

Each $p \in S_n$ satisfies

$$p(x) = \sum_{i=1}^{n-1} p(x_i)\, \phi_i(x),$$

that is, every $p \in S_n$ can be written as a linear combination of $\{\phi_1(x), \phi_2(x), \ldots, \phi_{n-1}(x)\}$.

We have to compute, for $i = 1, 2, \ldots, n-1$

$$K_{ij} = a(\phi_j, \phi_i) = \int_0^1 k(x)\, \phi_j'(x)\, \phi_i'(x)\, dx,$$

and

$$f_i = (f, \phi_i) = \int_0^1 f(x)\, \phi_i(x)\, dx.$$

Again we compute $v_n$, supposing that $\{\phi_1, \phi_1, \ldots, \phi_{n-1}\}$ is a basis for $V_n$. Then we have to

$$\text{find} \quad v_n \in V_n \quad s.t. \quad a(v_n, \phi_i) = (f, \phi_i), \qquad {\scriptstyle i=1,2,\ldots,n-1.}$$

$$v_n \in V_n \quad \Rightarrow \qquad v_n = \sum_{j=1}^{n-1} u_j\, \phi_j.$$

Finding $v_n \quad \Leftrightarrow \quad$ determining $u_1, u_2, \ldots, u_{n-1}$.

As before, this is a system of $n-1$ linear equations for the unknowns $u_1, u_2, \ldots, u_{n-1}$. Defining a matrix $\mathbf{K} \in^{n \times n}$ and a vector $\mathbf{f} \in^n$ by

$$K_{ij} = a(\phi_j, \phi_i), \qquad f_i = (f, \phi_i)\,, \qquad i, j = 1, 2, \ldots, n-1,$$

finding $v_n$ is equivalent to solving the linear system

$$\mathbf{K}\,\mathbf{u} = \mathbf{f}.$$

We use a regular mesh with $n$ subintervals and $h = 1/n$. We then have

$$\phi_i(x) = \begin{cases} \frac{1}{h}\,(x - (i-1)\,h)\,, & x_{i-1} < x < x_i, \\ -\frac{1}{h}\,(x - (i+1)\,h)\,, & x_i < x < x_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

and

$$\phi_i'(x) = \begin{cases} \frac{1}{h}, & x_{i-1} < x < x_i, \\ -\frac{1}{h}, & x_i < x < x_{i+1}, \\ 0, & \text{otherwise}. \end{cases}$$

On the other hand the problem with Inhomogeneous Dirichlet conditions

$$\begin{cases} -\left(k(x)\,u'(x)\right)' = f(x) \\ u(0) = \alpha, \qquad u(l) = \beta \end{cases} \tag{9}$$

which can be solved using the method of **shifting of data**: we find a function $g(x)$ satisfying the BC and define a new BVP for the unknown $w(x) = u(x) - g(x)$.

The only difference with the homogeneous Dirichlet problem is that we modify the right hand side vector $\mathbf{f}$. We choose $g_n$ to be piecewise linear and satisfy $g_n(x_0) = \alpha, \quad g_n(x_n) = \beta$: the simplest is

$$g_n = \alpha\phi_0 + \beta\phi_n$$

with $\phi_0$ and $\phi_n$ defined as the basis functions (7).

As before, we obtain the linear system $\mathbf{K}\,\mathbf{u} = \mathbf{f}$. The matrix $\mathbf{K}$ does not change, but $\mathbf{f}$ becomes

$$\begin{aligned} f_i &= (f, \phi_i) - a(g_n, \phi_i) = \\ &= \begin{cases} (f, \phi_1) - \alpha\, a(\phi_0, \phi_1), & i = 1 \\ (f, \phi_i), & i = 2, 3, \ldots, n-2 \\ (f, \phi_{n-1}) - \beta\, a(\phi_n, \phi_{n-1}), & i = n-1 \end{cases} \end{aligned} \tag{10}$$

and now the solution is

$$v_n = \sum_{j=1}^{n-1} u_j\,\phi_j + g_n.$$

(a) Write a program in *Mathematica* that solves (6), that is (**??**) when $k(x) = 1$ and $l = 1$, using FEM method with piecewise polynomials of first order and with

$$f(x) = (3x + x^2)\,\exp(x) \tag{??}$$

and

$$f(x) = \sin(x)\,\exp(x). \tag{??}$$

Compare the approximate solutions with the ones obtained using FD method for elliptic problems.

## Solution

This time we are solving again the same problem as in a) part but we are using FEM now. We are considering n = 10 points. Since we have to show the solution for two different functions, we will first show the solution for the first function and then we will show the solution for the other function, as in a).

### $\mathbf{f(x) = (3x + x^2)exp(x)}$

We start showing the approximated solution using FEM.

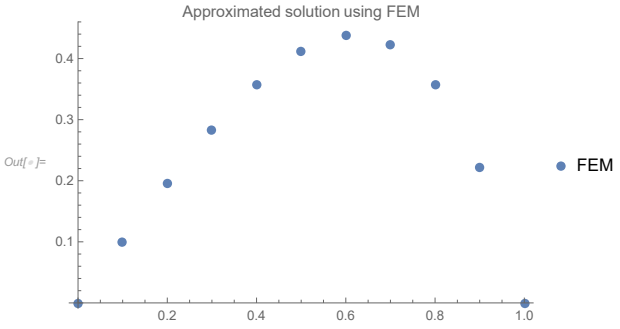


Figure 11: Approximated solution using FEM

Now we will compare this plot we have just shown with the solution obtained with FD method. We have considered n = 10 points in order to compare the results in a correct way.

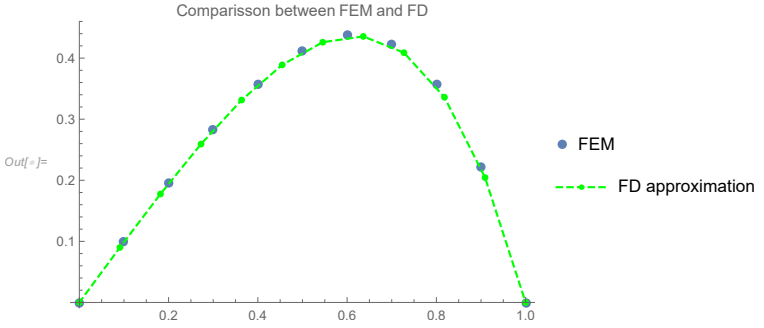


Figure 12: comparison between FD and FEM for f1

As we can observe, the points obtained with FEM and the ones obtained with FD do not coincide. This means that there are differences between both methods and one of them is going to be better than the other. Let's go and see which of these two is better.

In order to see which of these two is better, we are going to attend to the error. The next plot indicates the error of FEM with the exact one.

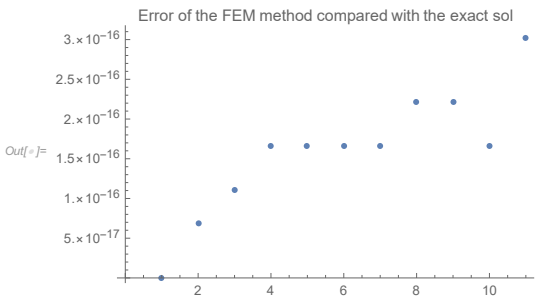


Figure 13: comparison of errors of FD and FEM with the exact solution for f1

As we saw in practice 5, the maximum value of the difference of the errors using FD approximation is 0.0000668168, whereas now we have seen that the error of the FEM method compared to the exact solution is of order of $10^{-16}$. Hence, we have shown that the error using FEM is much smaller than the one using FD.

Then, we can conclude that the FEM solution is a better solution. However, in terms of time, we have to say that the solution obtained with FD is faster, as we can see in the *Mathematica* Notebook.

Let's take now 20 points in the FD solution in order to get a better solution and then we will compare the results.

The next plot represents the comparison between FEM with n = 20 points and FD. The following graphic shows the comparison between both approximation:

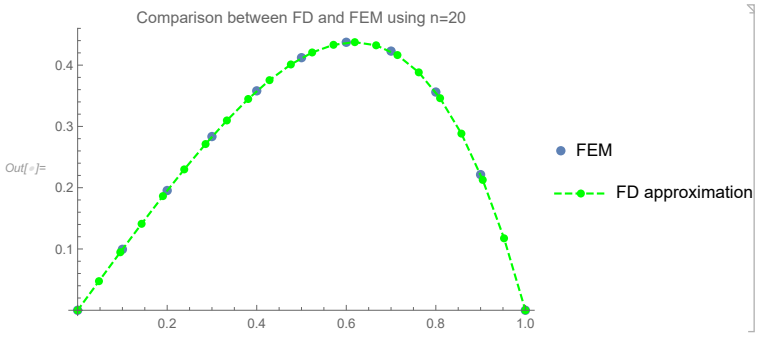


Figure 14: Comparison between Fem with 20 points and FD approximation

As we can see, the solutions are more similar than the ones obtained considering only n = 10 points for FD. Then, we can conclude that, as we increase the number of points, a better comparison we get.

**f(x) = sin(x) exp(x)**

Now, we solve the same problem, but we change the function into f(x) = $sin(x)exp(x)$

Let's start plotting the approximated solution making use of FEM.

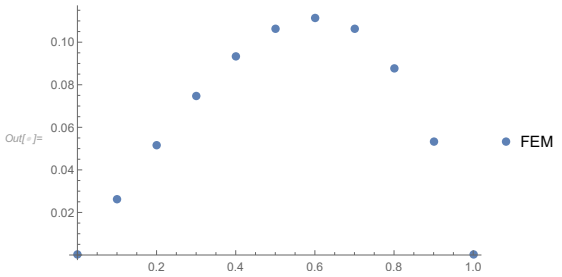


Figure 15: Approximated FEM solution for f2

Now, considering n = 10 points, we are going to compare this solution we have just plotted with the one we obtain making use of FD approximation.

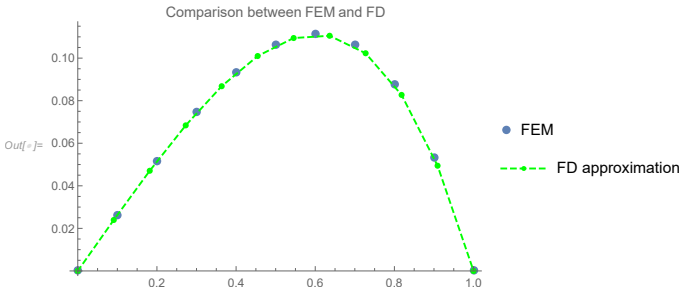


Figure 16: Approximated FEM solution and FD solution with n = 10 for f2

Again, the same way as we have seen before, with the other function, we can appreciate that the approximations do not fit very well together. Indeed, we can expect that the FEM solution is better, due to the result we have obtained with the other function.

The error obtained with FEM can be seen in the next two plots.

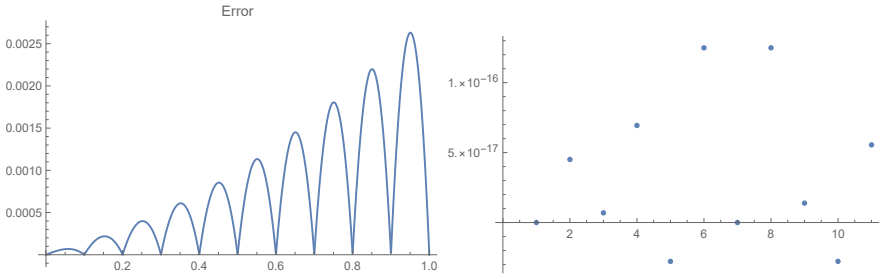


Figure 17: Error and the discrete error of the FEM approximation compared to the exact solution

It can be seen that the order of the error is $10^{-16}$, whereas the maximum value of the error according to FD is $6.68 \cdot 10^{-5}$, which has been obtained in the 5th computer practice. Then, as we expected, the error is smaller in the case of FEM, thus, we can say that the FEM solution is better.

Although, in order to get a better solution, it would be interesting to analyze the case of n = 20 points with FD.

Let's attend to it, and let's see what happens if we increase the number of points.

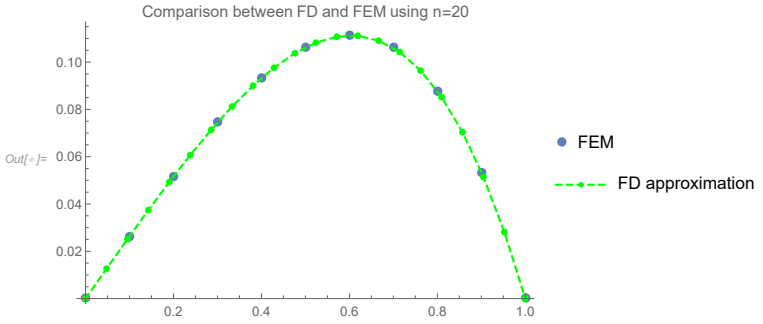


Figure 18: Comparison between FD (n=20 points) and FEM for f2

As we can see, now we can say that the approximations fit in a better way than before. Therefore, we conclude that, as we increase the number of points in FD, we get better approximations. Nevertheless, we keep having less CPU time in FD than in FEM.

## 8.3 Problem with Neumann BC

### Statement

$$\begin{cases} -\left(k(x)\,u'(x)\right)' = f(x) \\ \frac{du}{dx}(0) = \alpha, \qquad \frac{du}{dx}(l) = \beta \end{cases} \tag{11}$$

We have seen that in this case the weak form becomes

$$\text{find} \quad u \in C^2[0,l] \quad \text{such that} \quad a(u,v) = (f,v) + k(l)\,v(l)\,\beta - k(0)\,v(0)\,\alpha \qquad \forall v \in C^2[0,l]. \tag{12}$$

Now we have to extent the $S_n$ space to $\tilde{S}_n$ adding the basis function $\phi_0$ and $\phi_n$:

$$\tilde{S}_n = \{p : [0,l] \to \ \mid p \text{ is continuous and piecewise lineal}\} =$$

$$= \text{span}\{\phi_0, \phi_1, \ldots, \phi_n\}.$$

As in the case of Dirichlet BC, we have to solve $\tilde{\mathbf{K}}\,\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$ where

$$\tilde{K}_{ij} = a(\phi_j, \phi_i) = \int_0^l k(x)\,\phi_j'(x)\,\phi_i'(x)\,dx, \quad \tilde{f}_i = (f, \phi_i) = \int_0^l f(x)\,\phi_i(x)\,dx,$$

but now $i, j = 0, 1, 2, \ldots, n-1, n$, so it is a system of $n+1$ eqs. in the $n+1$ unknowns $\tilde{u}_0, \tilde{u}_1, \ldots, \tilde{u}_n$.

We face the difficulty that $\tilde{\mathbf{K}}$ is now **a singular matrix**.

To solve this singular system correctly, we must add another eq., appropriately chosen (the null space of $\tilde{\mathbf{K}}$ is 1D).

A simple choice is $\tilde{u}_n = 0$ (corresponding to choose the solution of (11) with $u(l) = 0$).

We can impose this additional eq. by simply removing the last row and the last column of $\tilde{\mathbf{K}}$, and the last entry from $\tilde{\mathbf{f}}$.

On the other side, $\tilde{\mathbf{f}}$ becomes

$$\begin{aligned} \tilde{f}_i &= (f, \phi_i) + k(l)\,\phi_i(l)\,\beta - k(0)\,\phi_i(0)\,\alpha = \\ &= \begin{cases} (f, \phi_0) - \alpha\,k(0), & i = 0 \\ (f, \phi_i), & i = 1, 2, 3, \ldots, n-1 \\ (f, \phi_{n-1}) + \beta\,k(l), & i = n \end{cases} \end{aligned} \tag{13}$$

Now the approximate solution is given by

$$v_n = \sum_{i=0}^{n} \tilde{u}_i \phi_i = \sum_{i=0}^{n-1} \tilde{u}_i \phi_i,$$

where the coefficients $\tilde{u}_0, \tilde{u}_1, \ldots, \tilde{u}_n$ satisfy $\tilde{\mathbf{K}}\,\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$.

(a) Write a program in *Mathematica* that solves (11) when $k(x) = 1$ and $l = 1$, using FEM method with piecewise polynomials of first order and with

$$f(x) = (x^2 + x - 1)\,\exp(x), \qquad \alpha = 0 \quad \text{and} \quad \beta = 0. \tag{14}$$

Compare the approximate solutions with the one obtained using the `DSolve` command.

## Solution

This time we are solving the same problem but this time we are considering Neumann boundary conditions.

Since we are asked to compare the exact and the approximated solutions, we will plot both of them and then we will show both together.

First, let's plot the exact solution:

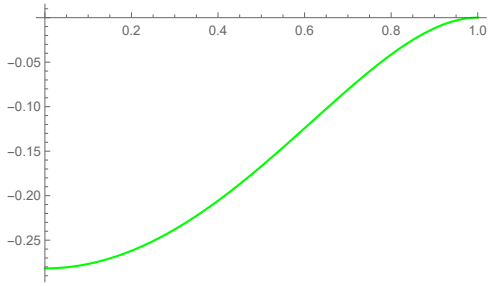


Figure 19: Exact solution

Once we have the exact one done, we plot now the approximated solution using FEM method with piecewise polynomials of first order.

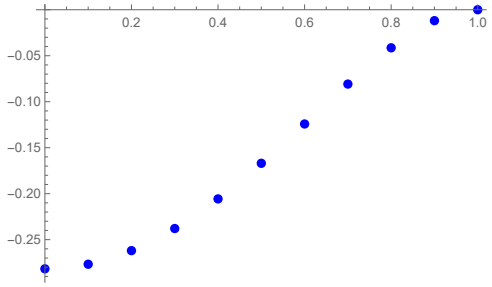


Figure 20: Approximated solution

The first impression is that this approximation is quite good, but we are going to be certain about it after we see this following graphic:

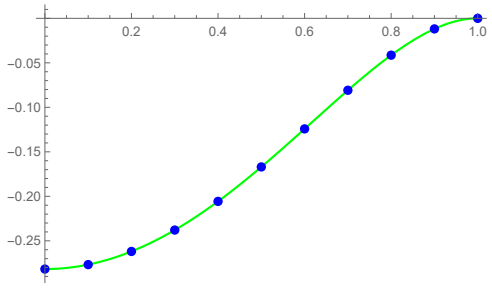


Figure 21: Exact solution vs approximated solution

It is very good, indeed. It seems that both solutions are very similar. In order to see how similar they are, let's see the discrete errors. We can expect that they are going to be very small:

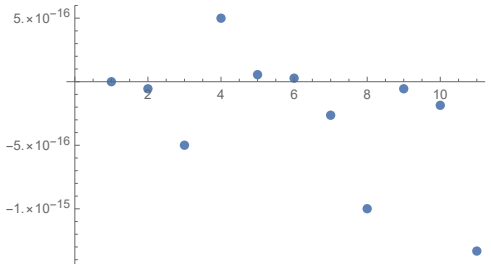


Figure 22: Discrete errors

As we can see, the order of the error is $10^{-15}$. Thus, we can say that the approximation we have obtained is a really good approximation to the solution. However, the highest error is in final part of the interval, the error has not a growing trend.