Finite Elements (1D)

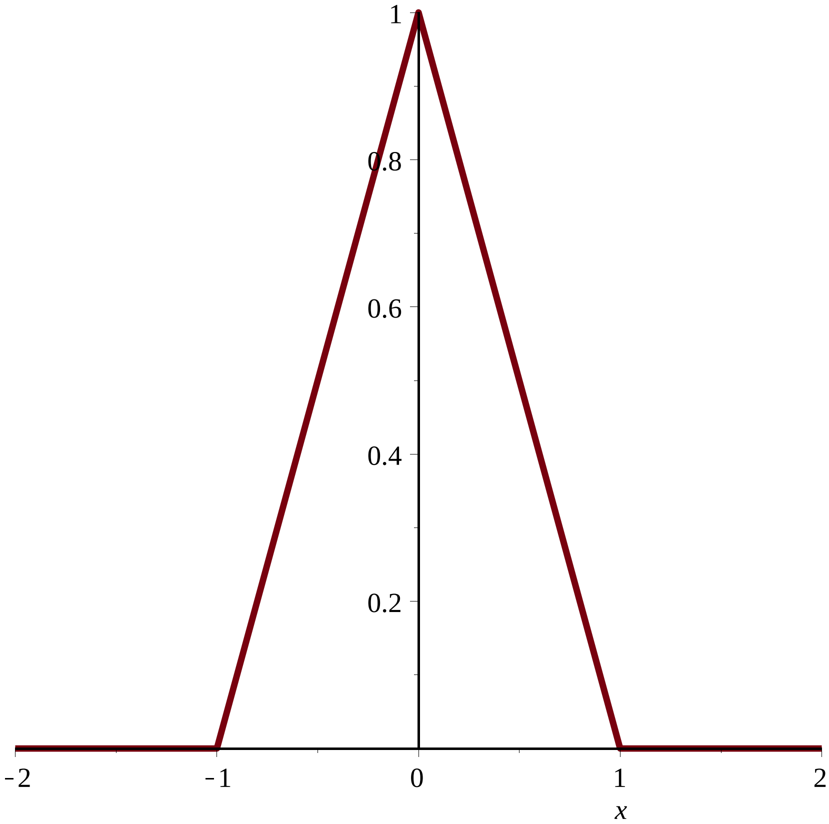
```
> restart;
> with(LinearAlgebra):
```

Building a basis Φ of test and ansatz functions

Template for basis functions f0

```
> f0 := piecewise(x<-1, 0, x<0, x+1, x<=1, 1-x, 0);
> plot(f0, x=-2..2, thickness=3);
```

$$f0 := \begin{cases} 0 & x < -1 \\ x+1 & x < 0 \\ 1-x & x \leq 1 \\ 0 & otherwise \end{cases}$$



Now, specify mesh width h and position x0

```
> n := 3;
> h := [seq(1/(n+1),i=1..n)];
> x0 := [seq(i/(n+1),i=1..n)];
```

$$n := 3$$

$$h := \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right]$$

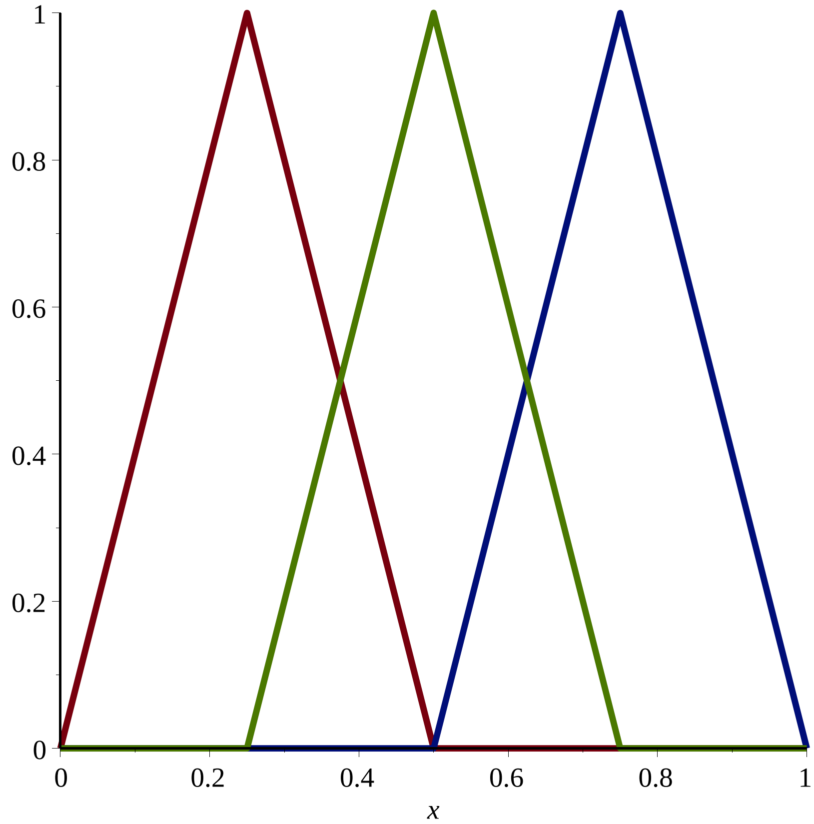
$$x0 := \left[\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\right] \quad \textbf{(1.2.1)}$$

Define the basis functions

```
> for i from 1 to n do
    phi[i] := convert(subs(x=(x-x0[i])/h[i],f0), piecewise,
  x):
  od:
```

Plot the basis functions

```
> plot({seq(phi[i], i=1..n)}, x=0..1,thickness=3);
```



Stiffness matrix and mass matrix

Stiffness matrix A

Stiffness matrix: $a_{i,j} := \int_0^1 \left(\frac{\partial}{\partial x}\phi_i\right)\left(\frac{\partial}{\partial x}\phi_j\right) dx$

```
> A := Matrix(n,n):
  for i from 1 to n do
    for j from 1 to n do
       A[i,j] := int( diff(phi[i],x)*diff(phi[j] ,x), x=0.
  .1);
    od;
  od;
  A;
```

$$\begin{bmatrix} 8 & -4 & 0 \\ -4 & 8 & -4 \\ 0 & -4 & 8 \end{bmatrix} \qquad (2.1.1)$$

Mass matrix B

Mass matrix: $b_{i,j} := \int_0^1 \phi_i \phi_j \, dx$

```
> B := Matrix(n,n):
  for i from 1 to n do
     for j from 1 to n do
       B[i,j] := int(phi[i]*phi[j], x=0..1);
     od;
  od;
  B;
```

$$\begin{bmatrix} \frac{1}{6} & \frac{1}{24} & 0 \\ \frac{1}{24} & \frac{1}{6} & \frac{1}{24} \\ 0 & \frac{1}{24} & \frac{1}{6} \end{bmatrix} \qquad (2.2.1)$$

A very simple example

We solve the (embarrassingly simple) equation $u = x\,(1-x)$ using the bilinear form

$$b(u, v) = \int_0^1 u(x)\, v(x)\, dx.$$

The result u_h in V_h minimises the L2 norm of $u_h - x\,(1-x)$.

```
> bb := x*(1-x);
> b := Vector(n):
  for i from 1 to n do
     b[i] := int(phi[i]*bb, x=0..1):
  od:
  b;
```

$$bb := x\,(1-x)$$

$$\begin{bmatrix} \frac{17}{384} \\ \frac{23}{384} \\ \frac{17}{384} \end{bmatrix} \qquad \textbf{(3.1)}$$

Solution

```
>  u := LinearSolve(B,b); evalf(%);
```

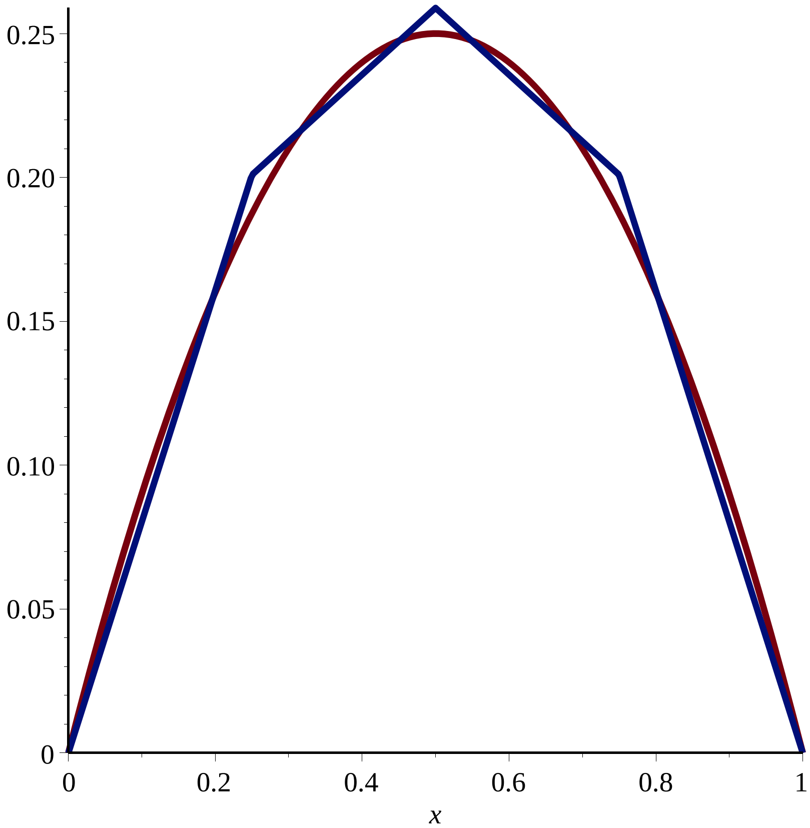
B is the system matrix, b the right hand side.
The respective piecewise linear function *uu* is obtained as a linear combination of the basis functions bs

```
> uu := convert(add(u[i]*phi[i],i=1..n), piecewise, x):
```

$$u := \begin{bmatrix} \frac{45}{224} \\ \frac{29}{112} \\ \frac{45}{224} \end{bmatrix}$$

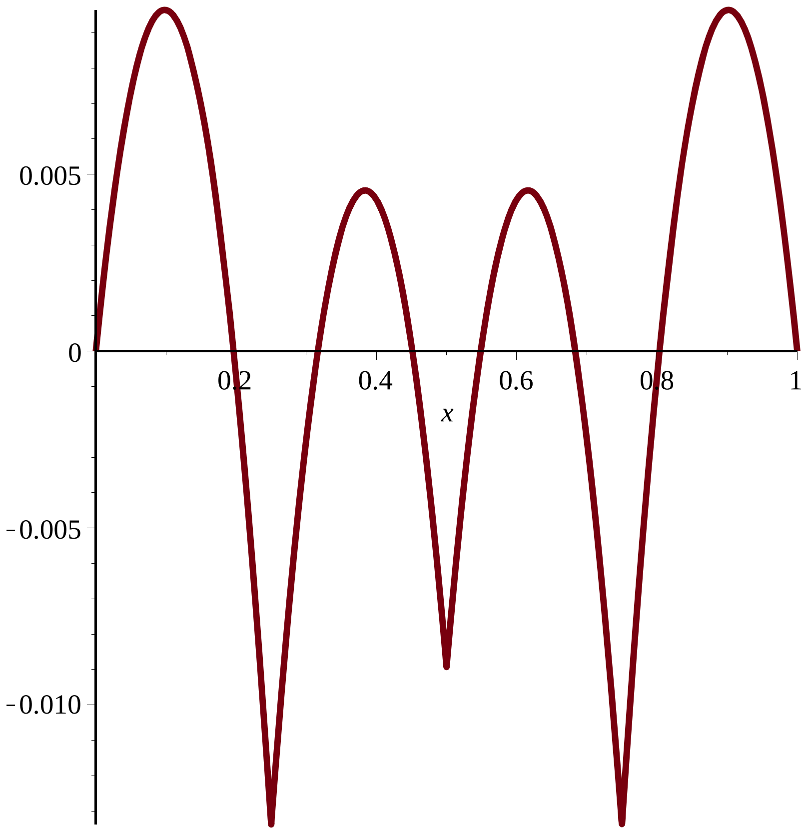
$$\begin{bmatrix} 0.2008928571 \\ 0.2589285714 \\ 0.2008928571 \end{bmatrix} \qquad \textbf{(3.1.1)}$$

```
> plot({uu,bb}, x=0..1,thickness=3);
```




```
> plot(bb-uu, x=0..1,thickness=3);
```

The difference between bb and the approximtion uu



Poisson's equation in 1D

For Poisson's equation, A (the stiffness matrix) is the system matrix.
We can retain x(1-x) as the right hand side (thus solving the equation -u" = x(1-x)), so b is still our right hand side of the Linear System.

The equation can still be solved exactly:

```
> usol := x -> -x^3/6 + x^4/12 + x/12;
  diff( -usol(x), x,x); usol(0); usol(1);
```

$$usol := x \rightarrow -\frac{1}{6}x^3 + \frac{1}{12}x^4 + \frac{1}{12}x$$

$$x - x^2$$

$$0$$

$$0 \qquad \textbf{(4.1)}$$

Use parabolic function as right-hand side:

```
> bb := x*(1-x);
> b := Vector(n):
  for i from 1 to n do
     b[i] := int(phi[i]*bb, x=0..1):
  od:
  b;
```

$$bb := x\,(1-x)$$

$$\begin{bmatrix} \frac{17}{384} \\ \frac{23}{384} \\ \frac{17}{384} \end{bmatrix} \quad (4.2)$$

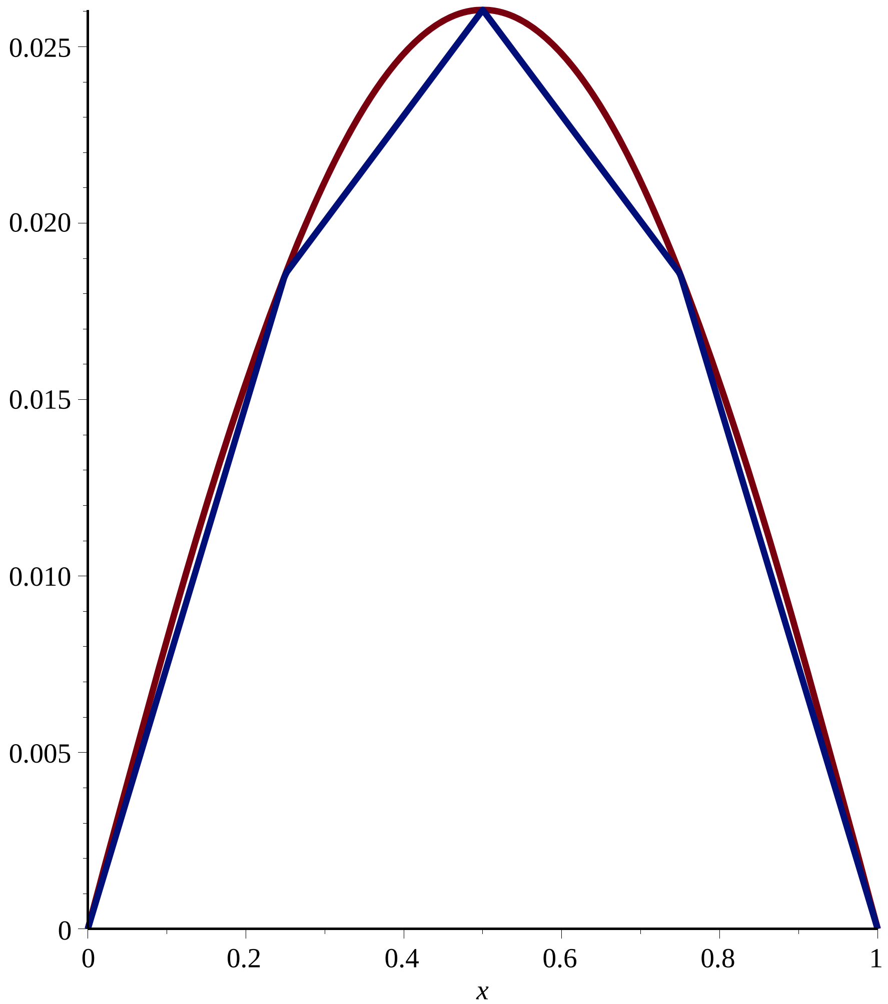
The FE solution, however, is:

```
> u := LinearSolve(A,b); evalf(%);
```

$$u := \begin{bmatrix} \frac{19}{1024} \\ \frac{5}{192} \\ \frac{19}{1024} \end{bmatrix}$$

$$\begin{bmatrix} 0.01855468750 \\ 0.02604166667 \\ 0.01855468750 \end{bmatrix} \quad (4.3)$$

```
> uu := convert(add(u[i]*phi[i],i=1..n), piecewise, x):
> plot({uu,usol(x)}, x=0..1,thickness=3);
```


0.025
0.020
0.015
0.010
0.005
0
0
0.2
0.4
0.6
0.8
1
x