Mat-1.174 Computational Partial Differential Equations
Mikko Kokkonen, 33015p

Second order base elements for FEM

In this project we study solving the equation

$$-\Delta u(x,y) + k^2 u(x,y) = f(x,y), \quad (x,y) \in \Omega$$

and its one dimensional counterpart

$$-u''(x) + k^2 u(x) = f(x), \quad x \in [0,1]$$

using finite element method with second order elements. For this equation the stiffness and mass matrices can be computed in without numerical integration and it can be used as a model in e.g. various problems in electromagnetics.

First we present details of the procedure in the one-dimensional case and then provide summary of results (from [**?**, Ch. 4]) for two dimensional case.

1 Interpolation polynomials

Interpolation polynomials are defined using the auxiliary polynomials $R_m(n,\zeta)$ that has m zeros at locations $0, 1/n, \ldots, (m-1)/n$ and has value 1 at location m/n. In one-dimensional case these polynomials are:

$$\begin{aligned}
R_0(2,\zeta) &= 1 \\
R_1(2,\zeta) &= \prod_{k=0}^{0} \frac{\zeta - k/2}{1/2 - k/2} = 2\zeta \\
R_2(2,\zeta) &= \prod_{k=0}^{1} \frac{\zeta - k/2}{1 - k/2} = \zeta \cdot 2(\zeta - 1/2) = \zeta(2\zeta - 1)
\end{aligned}$$

The basis functions are constructed from the second order Lagrange interpolation polynomials which can be defined with the auxiliary polynomials (here we have used the fact the simplex (local) coordinates are linearly dependent and have eliminated the other coordinate):

$$\begin{aligned}
\alpha_1(\zeta) &= \alpha_{02}(\zeta) = R_0(2,\zeta)\, R_2(2, 1-\zeta) = (1-\zeta)(2(1-\zeta)-1) = (1-\zeta)(1-2\zeta) \\
\alpha_2(\zeta) &= \alpha_{11}(\zeta) = R_1(2,\zeta)\, R_1(2, 1-\zeta) = 2\zeta \cdot 2(1-\zeta) = 4\zeta(1-\zeta) \\
\alpha_3(\zeta) &= \alpha_{20}(\zeta) = R_2(2,\zeta)\, R_0(2, 1-\zeta) = \zeta(2\zeta - 1)
\end{aligned}$$

2 Local stiffness and mass matrices

For the local stiffness and mass matrices the inner products between two Lagrange polynomials and the inner products between two derivatives of Lagrange polynomial must be computed. The elements of the mass matrix are:

$$
\begin{aligned}
T_{mn} &= \int \alpha_m \alpha_n \mathrm{d}\zeta \\
T_{11} &= \int_0^1 (1-\zeta)^2(1-2\zeta)^2\mathrm{d}\zeta = \frac{2}{15} \\
T_{22} &= \frac{8}{15} \\
T_{33} &= \frac{2}{15} \\
T_{12} &= T_{21} = \frac{1}{15} \\
T_{13} &= T_{31} = -\frac{1}{30} \\
T_{23} &= T_{32} = \frac{1}{15}
\end{aligned}
$$

For the stiffness matrix we need to compute the derivatives of the interpolation polynomials:

$$
\begin{aligned}
\alpha_1'(\zeta) &= -(1-2\zeta) - 2(1-\zeta) = 4\zeta - 3 \\
\alpha_2'(\zeta) &= 4(1-\zeta) - 4\zeta = 4 - 8\zeta \\
\alpha_3'(\zeta) &= (2\zeta - 1) + 2\zeta = 4\zeta - 1
\end{aligned}
$$

The elements of the stiffness matrix are:

$$
\begin{aligned}
S_{mn} &= \int a_m' \cdot a_n' \mathrm{d}\zeta \\
S_{11} &= \int_0^1 (4\zeta-3)^2\mathrm{d}\zeta = \frac{7}{3} \\
S_{22} &= \int_0^1 (4-8\zeta)^2\mathrm{d}\zeta = \frac{16}{3} \\
S_{33} &= \int_0^1 (4\zeta-1)^2\mathrm{d}\zeta = \frac{7}{3} \\
S_{12} &= S_{21} = \int_0^1 (4\zeta-3)(4-8\zeta)\mathrm{d}\zeta = -\frac{8}{3} \\
S_{13} &= S_{31} = \int_0^1 (4\zeta-3)(4\zeta-1)\mathrm{d}\zeta = \frac{1}{3} \\
S_{23} &= S_{32} = \int_0^1 (4-8\zeta)(4\zeta-1)\mathrm{d}\zeta = -\frac{8}{3}
\end{aligned}
$$

3 Global stiffness and mass matrices

Interval $[0,1]$ is assumed to be divided into sub-intervals with points $x_0 = 0 < x_1 < x_2 < \ldots < x_{n+1} = 1$. Define h_j as

$$h_j = x_j - x_{j-1}, \quad j = 1, 2, \ldots, n+1$$

In the one-dimensional 2nd order case there are two types of basis functions, $\phi_{1,j}$ that spans two adjacent intervals and $\phi_{2,j}$ that spans only one interval:

$$
\phi_{1,j}(x) = \begin{cases} 0, & x < x_{j-1} \vee x > x_{j+1} \\ \alpha_1\left(\frac{x - x_{j-1}}{h_j}\right), & x_{j-1} \le x < x_j \\ \alpha_3\left(\frac{x - x_j}{h_{j+1}}\right), & x_j \le x < x_{j+1} \end{cases}
$$

$$\phi_{2,j}(x) = \begin{cases} 0, & x < x_{j-1} \vee x > x_j \\ \alpha_2\left(\frac{x-x_{j-1}}{h_j}\right), & x_{j-1} \le x < x_j \end{cases}$$

The elements of the global stiffness and mass matrices are obtained by integrating products of two basis functions and products of two derivatives of basis functions. The integrals are easily calculated using a change of variables: $y = \frac{x-x_{j-1}}{h_j}$.

We order the basis functions so that elements of the coefficient vector with odd index $2j-1$ correspond to $\phi_{2,j}$ and elements with even index $2j$ correspond to $\phi_{1,j}$. The mass matrix $\mathcal{T}$ is then:

$$\begin{aligned}
\mathcal{T}_{2j,2j} = \int_0^1 \phi_{1,j}(x)\phi_{1,j}(x)\mathrm{d}x &= h_j T_{11} + h_{j+1}T_{33} = \frac{2}{15}(h_j + h_{j+1}) \\
\mathcal{T}_{2j,2j+2} = \mathcal{T}_{2j+2,2j} = \int_0^1 \phi_{1,j}(x)\phi_{i,j+1}(x)\mathrm{d}x &= h_{j+1}T_{13} = -\frac{1}{30}h_{j+1} \\
\mathcal{T}_{2j-1,2j-1} = \int_0^1 \phi_{2,j}(x)\phi_{2,j}(x)\mathrm{d}x &= h_j T_{22} = \frac{8}{15}h_j \\
\mathcal{T}_{2j,2j-1} = \mathcal{T}_{2j-1,2j} = \int_0^1 \phi_{1,j}(x)\phi_{2,j}(x)\mathrm{d}x &= h_j T_{12} = \frac{1}{15}h_j \\
\mathcal{T}_{2j,2j+1} = \mathcal{T}_{2j+1,2j} = \int_0^1 \phi_{1,j}(x)\phi_{2,j+1}(x)\mathrm{d}x &= h_j T_{32} = \frac{1}{15}h_j
\end{aligned}$$

All other entries of $\mathcal{T}$ are zero.

The stiffness matrix $\mathcal{S}$:

$$\begin{aligned}
\mathcal{S}_{2j,2j} = \int_0^1 \phi'_{1,j}(x)\phi'_{1,j}(x)\mathrm{d}x &= \frac{1}{h_j^2}\int_{x_{j-1}}^{x_j}\left[\alpha'_1\left(\frac{x-x_{j-1}}{h_j}\right)\right]^2\mathrm{d}x + \frac{1}{h_{j+1}^2}\int_{x_j}^{x_{j+1}}\left[\alpha'_3\left(\frac{x-x_j}{h_{j+1}}\right)\right]^2\mathrm{d}x \\
&= \frac{1}{h_j}\int_0^1\left[\alpha'_1(x)\right]^2\mathrm{d}x + \frac{1}{h_{j+1}}\int_0^1\left[\alpha'_3(x)\right]^2\mathrm{d}x \\
&= \frac{1}{h_j}S_{11} + \frac{1}{h_{j+1}}S_{33} \\
&= \frac{7}{3}\left(\frac{1}{h_j} + \frac{1}{h_{j+1}}\right)
\end{aligned}$$

The other integrals are computed similarly:

$$\begin{aligned}
\mathcal{S}_{2j,2j} = \int_0^1 \phi'_{1,j}(x)\phi'_{2,j}(x)\mathrm{d}x &= \frac{1}{h_j^2}\int_{x_{j-1}}^{x_j}\alpha'_1\left(\frac{x-x_{j-1}}{h_j}\right)\alpha'_2\left(\frac{x-x_{j-1}}{h_j}\right)\mathrm{d}x \\
&= \frac{1}{h_j}S_{12} \\
&= -\frac{8}{3}\cdot\frac{1}{h_j}
\end{aligned}$$

$$\begin{aligned}
\mathcal{S}_{2j,2j+1} = \mathcal{S}_{2j+1,2j} = \int_0^1 \phi'_{1,j}(x)\phi'_{2,j+1}(x)\mathrm{d}x &= \frac{1}{h_{j+1}^2}\int_{x_j}^{x_{j+1}}\alpha'_3\left(\frac{x-x_j}{h_{j+1}}\right)\alpha'_2\left(\frac{x-x_j}{h_{j+1}}\right)\mathrm{d}x \\
&= \frac{1}{h_{j+1}}S_{32} \\
&= -\frac{8}{3}\cdot\frac{1}{h_{j+1}}
\end{aligned}$$

$$
\begin{aligned}
\mathcal{S}_{2j,2j+2} = \mathcal{S}_{2j+2,2j} = \int_0^1 \phi'_{1,j}(x)\phi'_{1,j+1}(x)\mathrm{d}x &= \frac{1}{h_{j+1}^2}\int_{x_j}^{x_{j+1}} \alpha'_3\left(\frac{x-x_j}{h_{j+1}}\right)\alpha'_1\left(\frac{x-x_j}{h_{j+1}}\right)\mathrm{d}x \\
&= \frac{1}{h_{j+1}} S_{31} \\
&= \frac{1}{3}\cdot\frac{1}{h_{j+1}}
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{S}_{2j+1,2j+1} = \int_0^1 \phi'_{2,j}(x)\phi'_{2,j}(x)\mathrm{d}x &= \frac{1}{h_j^2}\int_{x_{j-1}}^{x_j}\left[\alpha'_2\left(\frac{x-x_{j-1}}{h_j}\right)\right]^2\mathrm{d}x \\
&= \frac{1}{h_j} S_{22} \\
&= \frac{16}{3}\cdot\frac{1}{h_j}
\end{aligned}
$$

4 A case study

This section compares the solutions obtained the first and second order elements to the problem (Problem 2 from Exercise session 5)

$$-u''(x) = f(x), \quad u(0) = u(1) = 0$$

where $f(x) = (3x + x^2)e^x$. The exact solution is

$$u(x) = x(1-x)e^x.$$

A graphical plot of the FEM solution with second order elements is shown in Figure 1.

Figure 2 shows the approximation error of the FEM solutions with first and second order elements. We note that in this case the solution with a few second order elements provide a solution that is much more accurate than the solution with the same number of first order elements.

5 Second order elements in two dimensions

In two dimensional problems the interpolation polynomials are defined using the same auxiliary polynomials as in one dimensional case:

$$\alpha_{ijk}(\zeta_1,\zeta_2,\zeta_3) = R_i(2,\zeta_1)R_j(2,\zeta_2), R_k(2,\zeta_3)$$

There 6 different second order interpolation functions:

$$
\begin{aligned}
\alpha_{200} &= R_2(2,\zeta_1)R_0(2,\zeta_2)R_0(2,\zeta_3) = \zeta_1(2\zeta_1 - 1) \\
\alpha_{020} &= R_0(2,\zeta_1)R_2(2,\zeta_2)R_0(2,\zeta_3) = \zeta_2(2\zeta_2 - 1) \\
\alpha_{002} &= R_0(2,\zeta_1)R_0(2,\zeta_2)R_2(2,\zeta_3) = \zeta_3(2\zeta_3 - 1) \\
\alpha_{110} &= R_1(2,\zeta_1)R_1(2,\zeta_2)R_0(2,\zeta_3) = 4\zeta_1\zeta_2 \\
\alpha_{110} &= R_1(2,\zeta_1)R_0(2,\zeta_2)R_1(2,\zeta_3) = 4\zeta_1\zeta_3 \\
\alpha_{110} &= R_0(2,\zeta_1)R_1(2,\zeta_2)R_1(2,\zeta_3) = 4\zeta_2\zeta_3
\end{aligned}
$$

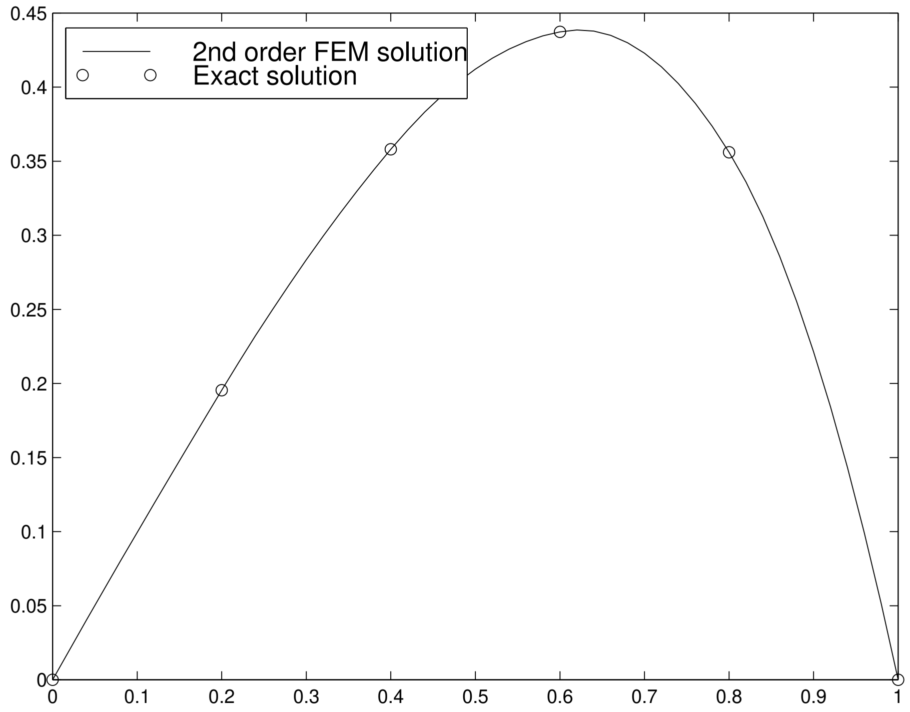


Figure 1: *The FEM solution with second order elements and the exact solution. The FEM solution is obtained with discretization into 5 sub-intervals and 9 basis functions. For the graph the FEM solution has been evaluated at 9 points between every two adjacent grid points.*

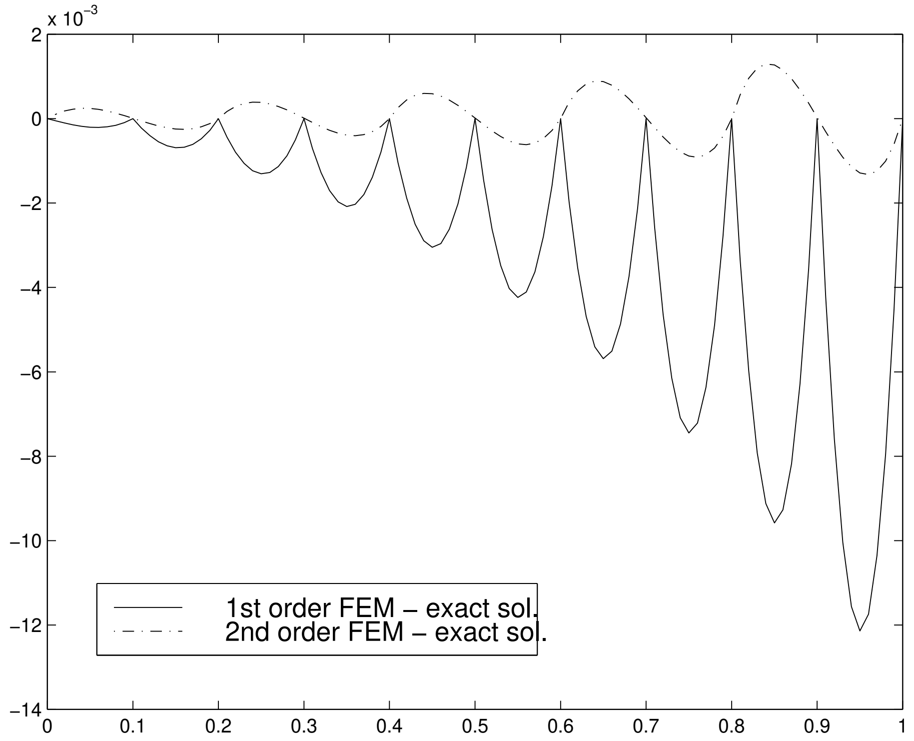


Figure 2: *The error of the FEM solutions with respect to exact solution. The first order FEM solution is obtained with discretization into 10 sub-intervals and 9 basis functions.*

where $(\zeta_1, \zeta_2, \zeta_3)$ are the local co-ordinates that are obtained from the global co-ordinates (x, y) with the equation

$$\zeta_i = \frac{\begin{vmatrix} 1 & x & y \\ 1 & x_{i-1} & y_{i-1} \\ 1 & x_{i+1} & y_{i+1} \end{vmatrix}}{\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}}$$

In the equation above the indices are computed modulo 3 (i.e. $0 \equiv 3$ and $4 \equiv 1$). The first three polynomials have value 1 in one of the vertices of the triangle and the last three have value 1 in one of the midpoints of the edges.

There are also two different types of global basis functions in two dimesions. These are illustrated in Figure 3. We note that the number of triangles in the support of a basis function of the second type depends on the triangulation.

As in the one dimensional case the stiffness and mass matrices can be computed in terms of local co-ordinates and then transformed into global co-ordinates. The mass matrix is given by the equation

$$\begin{aligned} \mathcal{T}_{mn} &= \int \alpha_m \alpha_n \mathrm{d}\Omega \\ &= 2A \int \alpha_m \alpha_n \frac{\mathrm{d}\Omega}{2A} \\ &= 2A\, T_{mn} \end{aligned}$$

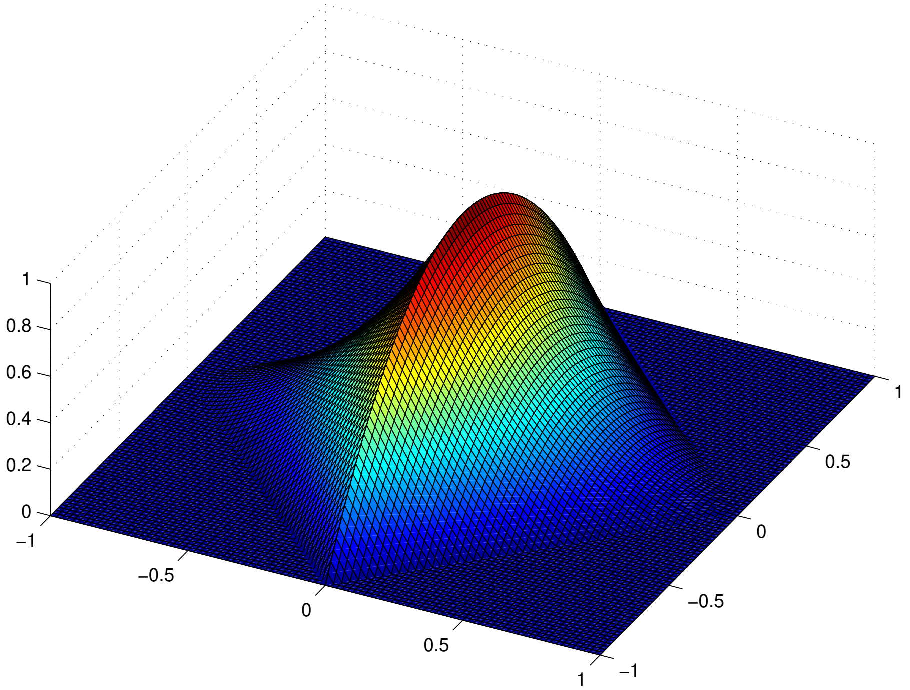
where A is the area of the triangle and the last integral is independent of the shape and size of the triangle and needs therefore to be evaluated only once (in local co-ordinates). The expression for stiffness matrix is more complicated:

$$\begin{aligned} \mathcal{S}_{mn} &= \int \nabla\alpha_m \cdot \nabla\alpha_n \mathrm{d}\Omega \\ &= \sum_{k=1}^{3} \cot(\theta_k) \int \left(\frac{\partial\alpha_m}{\partial\zeta_{k+1}} - \frac{\partial\alpha_m}{\partial\zeta_{k-1}} \right) \left(\frac{\partial\alpha_n}{\partial\zeta_{k+1}} - \frac{\partial\alpha_m}{\partial\zeta_{k-1}} \right) \frac{\mathrm{d}\Omega}{2A} \\ &= \sum_{k=1}^{3} Q_{mn}^{(k)} \cot(\theta_k) \end{aligned}$$

Here $\cot(\theta_k)$ is the angle at vertex k and again the last integral needs to be computed only once. The three matrices $Q^{(k)}$ are dependent because the three first (and three last) interpolation polynomials are symmetric and other two Q matrices can be easily computed once one is known:

$$\begin{aligned} Q^{(2)} &= P^\top Q^{(1)} P \\ Q^{(3)} &= P^\top Q^{(2)} P \end{aligned}$$

P is the permutation array describing the symmetries. The left multiplication by $P^\top$ and right multiplication by P has the effect of permuting the rows and columns. The matrices T and $Q^{(1)}$ have been tabulated in [**?**, Ch. 4] for two dimensional first, second and third order interpolation polynomials. These matrices $\mathcal{S}$ and $\mathcal{T}$ defined in [**?**, Ch. 4] are actually local matrices computed with global co-ordinates. It is not possible to compute "generic" global matrices of a two dimensional problem in a similar way as in one-dimensional problems because the form of these matrices depend on the details of a specific triangulation scheme (the extent of a support of a basis function of second kind depends on the triangulation).



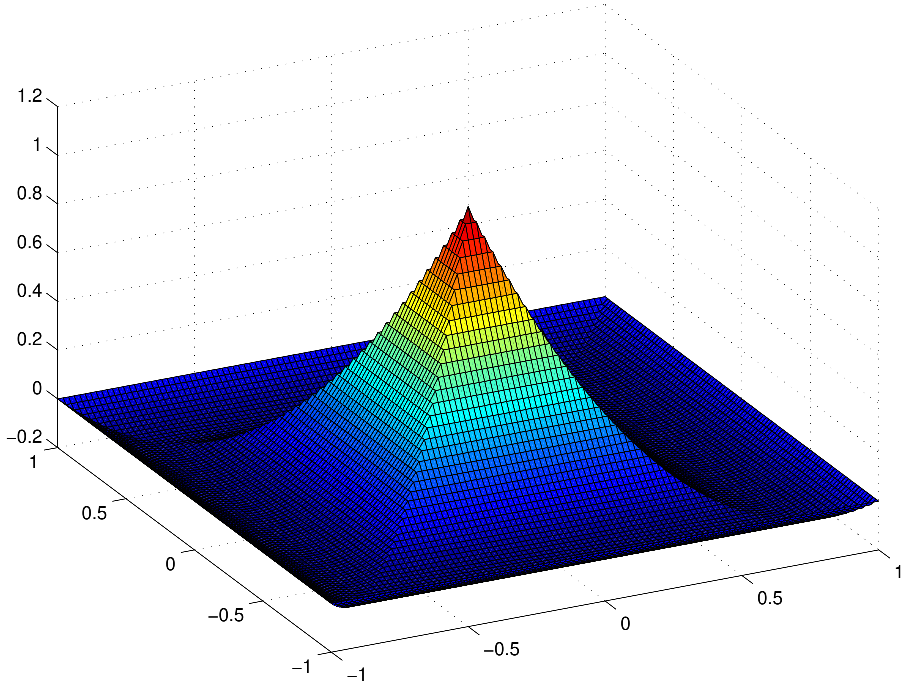


Figure 3: *Basis functions in two dimensions. Upper figure shows basis function of the first type and the lower depicts a basis function of second type whose support consists of four triangles.*

6 MATLAB codes

6.1 One dimensional FEM-solver

```
% This function solves the PDE (-u'' = f, u(0) = u(1) = 0) using FEM
% with 2nd order elements.
function [x, v] = fem2o(ff, h)
nh = length(h);
x = 0;
for k=1:length(h)
  x(k+1) = x(k) + h(k);
end

d = reshape([7 ./ h(1:(nh-1)) + 7 ./ h(2:nh); 16 ./ h(2:nh)], 1, 2*(nh-1))/3;
d = [16/(3*h(1)) d];
ud1 = -[8/h(1) reshape([8 ./ h(2:(nh-1)); 8 ./ h(2:(nh-1))], 1, 2*(nh-2)) 8/h(nh)] / 3;
ud2 = [0 reshape([1 ./ h(2:(nh-1)); zeros(1, nh-2)], 1, 2*nh-4)] / 3;

A = spdiags([[ud2 0 0]' [ud1 0]' d' [0 ud1]' [0 0 ud2]'], [-2 -1 0 1 2], ...
    2*nh-1, 2*nh-1);

b = quad8('fxphi2', 0, h(1), [], [], ff, 0, h(1));
for k=1:(nh-1)
  b(2*k) = quad8('fxphi1', x(k), x(k+2), [], [], ff, x(k), x(k+1), h(k), h(k+1));
  b(2*k+1) = quad8('fxphi2', x(k+1), x(k+2), [], [], ff, x(k+1), h(k+1));
end

v = A \ b';


function y = fxphi1(x, ff, x1, x2, h1, h2)
phi = [a1((x(x<x2) - x1) ./ h1), a3((x(x>=x2) - x2) ./ h2)];
y = feval(ff, x) .* phi;

function y = a1(x)
y = x .* (2*x - 1);

function y = a3(x)
y = (1-x) .* (1-2*x);


function y = fxphi2(x, ff, xs, h)
y = feval(ff, x) .* a2((x - xs) ./ h);

function y = a2(x)
y = 4*x .* (1-x);
```

6.2 Tools for plotting 2D basis functions

```
% This function computes the values of an interpolation polynomial
% trx defines the x co-ordinates of the vertices of the triangle
% try defines the y co-ordinates of the vertices of the triangle
% xrng defines the x range where the basis function is evaluated
% yrng defines the y range where the basis function is evaluated
% fno selects the basis function
function [sfdd, xx, yy] = basefgen(trx, try, xrng, yrng, fno)
```

```
M = trmat(trx, try);

[xx, yy] = meshgrid(xrng, yrng);
x = reshape(xx, 1, prod(size(xx)));
y = reshape(yy, 1, prod(size(yy)));
lc = M * [ones(1, length(x)); x; y];

sz = size(xx);
sfd = trii(lc(1, :), lc(2, :), lc(3, :), fno);
sfdd = reshape(sfd, sz(1), sz(2));


% This function defines the transformation matrix from global to local
% co-ordinates. x (resp. y) contains the x (resp. y) values of co-ordinates
% of the edges of triangles.
function M = trmat(x, y)

A = det([1 1 1; x; y]);
f = [2 3 1];
b = [3 1 2];
M = [x(f)' .* y(b)' - x(b)' .* y(f)', y(f)' - y(b)', x(b)' - x(f)'] / A;


% This function computes the values of an interpolation polynomial.
% x, y, and z are the local co-ordinates where the polynomial is evaluated
% fn selects the basis function
function h = tri_ip(x, y, z, fn)

switch fn
  case 1,
    h = x.*(2.*x-1);
  case 2,
    h = y.*(2.*y-1);
  case 3,
    h = z.*(2.*z-1);
  case 4,
    h = 4*x.*y;
  case 5,
    h = 4*x.*z;
  case 6
    h = 4*y.*z;
end
```

6.2.1 First basis function of Figure 3

```
[sfdd, xx, yy] = basefgen([0 0 1], [1 -1 0], 0:0.02:1, -1:0.02:1, 4);
% Set the values of points outsize triangle to zero.
sfdd = sfdd .* ((yy <= 1 - xx) & (yy >= xx - 1));

[sfdd1, xx1, yy1] = basefgen([0 0 -1], [1 -1 0], -1:0.02:0, -1:0.02:1, 4);
sfdd1 = sfdd1 .* ((yy1 <= 1 + xx1) & (yy1 >= -xx1 - 1));

surf([xx1 xx], [yy1 yy], [sfdd1 sfdd]);

view([.5 -1 1.5])
```

6.2.2 Second basis function of Figure 3

```
[s1, x, y] = basefgen([-1 -1 0], [1 -1 0], -1:0.02:1, -1:0.02:1, 3);
% Next line defines a logical variable that indicates which points are inside
% triangle
t1 = ((y <= -x) & (y >= x));

s2 = basefgen([-1 1 0], [-1 -1 0], -1:0.02:1, -1:0.02:1, 3);
t2 = ((y <= -x) & (y <= x));

s3 = basefgen([1 1 0], [-1 1 0],  -1:0.02:1, -1:0.02:1, 3);
t3 = ((y >= -x) & (y <= x));

s4 = basefgen([-1 1 0], [1 1 0],  -1:0.02:1, -1:0.02:1, 3);
t4 = ((y >= -x) & (y >= x));

s = zeros(size(x));
s(t1) = s1(t1);
s(t2) = s2(t2);
s(t3) = s3(t3);
s(t4) = s4(t4);

surf(x, y, s);

view([-0.5 -1 0.75]);
```