# Mathematical methods for engineering. The finite element method

Ana Alonso

University of Trento

Doctoral School in Environmental Engineering - January 25, 2012

# FEM: Remarks on programming

$$\begin{cases} -\Delta u = f & \text{on } \Omega = (0,1)\times(0,1) \\ u = 0 & \text{in } \partial\Omega \end{cases}$$

$$\begin{cases} \text{Find } u \in V \text{ such that} \\ \displaystyle\int_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega f\, v\, dx \quad \forall v \in V \end{cases}$$

We consider a triangulation of the domain $\mathcal{T}_h = \{K_k\}_{k=1}^{nel}$ and the finite dimensional space $V_h$ of piecewise linear functions on $\mathcal{T}_h$.

$$\begin{cases} \text{Find } u_h \in V_h \text{ such that} \\ \displaystyle\int_\Omega \nabla u_h \cdot \nabla v_h \, dx = \int_\Omega f\, v_h\, dx \quad \forall v_h \in V_h \end{cases}$$

# The linear system

Let $\{\phi_i\}_{i=1}^{\mathtt{ndof}}$ be a base of $V_h$. Then $u_h(x) = \sum_{j=1}^{\mathtt{ndof}} U_j \phi_j(x)$ and in particular

$$\int_\Omega \nabla \Big( \sum_{j=1}^{\mathtt{ndof}} U_j \phi_j(x) \Big) \cdot \nabla \phi_i(x)\, dx = \int_\Omega f(x)\, \phi_i(x)\, dx \quad \forall i = 1, \ldots, \mathtt{ndof}$$

$$\sum_{j=1}^{\mathtt{ndof}} U_j \int_\Omega \nabla \phi_j(x) \cdot \nabla \phi_i(x)\, dx = \int_\Omega f(x)\, \phi_i(x)\, dx \quad \forall i = 1, \ldots, \mathtt{ndof}$$

$$a_{i,j} = \int_\Omega \nabla \phi_j(x) \cdot \nabla \phi_i(x)\, dx\,, \quad f_i = \int_\Omega f(x)\, \phi_i(x)\, dx$$

$$A\,\mathbf{U} = \mathbf{f}$$

# The element matrix $A^K$ and the element load $\mathbf{f}^K$

$$a_{i,j} = \int_\Omega \nabla\phi_j(x)\cdot\nabla\phi_i(x)\,dx = \sum_{l=1}^{nel}\int_{K_l}\nabla\phi_j(x)\cdot\nabla\phi_i(x)\,dx$$

Each element stiffness matrices is a $3\times 3$ matrix.

$$S^K = \begin{bmatrix} s_{1,1} & s_{1,2} & s_{1,3}\\ s_{2,1} & s_{2,2} & s_{2,3}\\ s_{3,1} & s_{3,2} & s_{3,3}\end{bmatrix}$$

In order to calculate $s_{i,j} = \int_K \nabla\phi_i\cdot\nabla\phi_j$ (local numeration) it is easier to work on the reference element.

Analogously

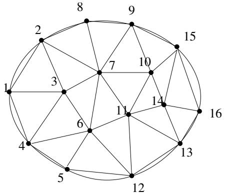
$$f_i = \int_\Omega f(x)\,\phi_i(x)\,dx = \sum_{l=1}^{nel}\int_{K_l} f(x)\,\phi_i(x)\,dx$$

and we can consider an element load $\mathbf{f}^K$ that is a vector with three components.

# Main steps

- Construction and representation of the triangulation.
- Computation of the element stiffness matrices $A^K$ and element loads $\mathbf{f}^K$.
- Assembly of the global stiffness matrix $A$ and load vector $\mathbf{f}$.
- Solution of the system of equations $A\,\mathbf{U} = \mathbf{f}$.
- Postprocesing.

# Representation of the triangulation

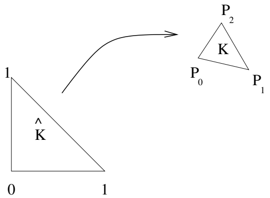



$$elem = \begin{bmatrix} 1 & 1 & 3 & 2 & \dots & 14 \\ 3 & 4 & 7 & 7 & \dots & 16 \\ 2 & 3 & 2 & 8 & \dots & 15 \end{bmatrix}$$

To represent a triangulation we will use:

- A matrix `coor` containing for each node its coordinates.
  - `coor` has two rows and `nnod` columns.
  - In this way we have numerated the nodes.
- Another matrix `elem` containg for each element its nodes.
  - `elem` has three rows and `nel` columns.
  - In this way we have a local numeration for the nodes of each triangle.

# The reference element




$$\hat{K} \rightsquigarrow K$$

This trasformation makes

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightsquigarrow \begin{bmatrix} p_0 \\ q_0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightsquigarrow \begin{bmatrix} p_1 \\ q_1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} p_2 \\ q_2 \end{bmatrix}$$

$$\mathbf{x} = B\hat{\mathbf{x}} + \mathbf{d} \quad \hat{\mathbf{x}} = B^{-1}(\mathbf{x} - \mathbf{d})$$

with

$$B = \begin{bmatrix} p_1 - p_0 & p_2 - p_0 \\ q_1 - q_0 & q_2 - q_0 \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} p_0 \\ q_0 \end{bmatrix}$$

# The element matrix

Concerning the gradients

$$\frac{\partial \phi_i}{\partial x_1} = \frac{\partial \hat{\phi}_i}{\partial \hat{x}_1}\frac{\partial \hat{x}_1}{\partial x_1} + \frac{\partial \hat{\phi}_i}{\partial \hat{x}_2}\frac{\partial \hat{x}_2}{\partial x_1} = \frac{\partial \hat{\phi}_i}{\partial \hat{x}_1} m_{1,1} + \frac{\partial \hat{\phi}_i}{\partial \hat{x}_2} m_{2,1}$$

where $M = B^{-1}$. Analogously

$$\frac{\partial \phi_i}{\partial x_2} = \frac{\partial \hat{\phi}_i}{\partial \hat{x}_1} m_{1,2} + \frac{\partial \hat{\phi}_i}{\partial \hat{x}_2} m_{2,2}$$

Hence

$$\int_K \nabla \phi_i \cdot \nabla \phi_j \, d\mathbf{x} = |\det(B)| \int_{\hat{K}} B^{-T} \nabla \hat{\phi}_i \cdot B^{-T} \nabla \hat{\phi}_j \, d\hat{\mathbf{x}}$$

# The element matrix

Concerning the gradients

$$\frac{\partial \phi_i}{\partial x_1} = \frac{\partial \hat{\phi}_i}{\partial \hat{x}_1}\frac{\partial \hat{x}_1}{\partial x_1} + \frac{\partial \hat{\phi}_i}{\partial \hat{x}_2}\frac{\partial \hat{x}_2}{\partial x_1} = \frac{\partial \hat{\phi}_i}{\partial \hat{x}_1} m_{1,1} + \frac{\partial \hat{\phi}_i}{\partial \hat{x}_2} m_{2,1}$$

where $M = B^{-1}$. Analogously

$$\frac{\partial \phi_i}{\partial x_2} = \frac{\partial \hat{\phi}_i}{\partial \hat{x}_1} m_{1,2} + \frac{\partial \hat{\phi}_i}{\partial \hat{x}_2} m_{2,2}$$

Hence

$$\int_K \nabla \phi_i \cdot \nabla \phi_j \, d\mathbf{x} = |\det(B)| \int_{\hat{K}} B^{-T} \nabla \hat{\phi}_i \cdot B^{-T} \nabla \hat{\phi}_j \, d\hat{\mathbf{x}}$$

Concerning the functions

$$\int_K \phi_i \, \phi_j \, d\mathbf{x} = |\det(B)| \int_{\hat{K}} \hat{\phi}_i \, \hat{\phi}_j \, d\hat{\mathbf{x}}$$

# selem

```
function s=selem(K)
for i=1:2
   B(:,i)=K(:,i+1)-K(:,1);
end
GFref=[-1 1 0; -1 0 1]; % each column is the gradient
                        % of the corresponding base function
                        % in the reference element.
GF=inv(B)'*GFref;
for i=1:3
   for j=1:3
      s(i,j)=GF(:,i)'*GF(:,j);
   end
end
s=s*abs(det(B))/2;      % 1/2 is the area of the reference
                        % element
```

# Assembly of the global stiffness matrix $A$

$$elem = \begin{bmatrix} 1 & 2 & 2 & \ldots & 30 \\ 2 & 8 & 3 & \ldots & 36 \\ 7 & 7 & 8 & \ldots & 35 \end{bmatrix}$$

```
for l=1:number of elements
calculate the stiffness matrices of the element
   for i=1:3
     for j=1:3
       A(elem(i,l),elem(j,l))=A(elem(i,l),elem(j,l))+s(i,j)
     end
   end
end
```

# The load vector $\mathbf{f}$

- For each element there is a $3 \times 1$ load vector

$$\mathbf{f}_i^K = \int_K f(\mathbf{x})\phi_i(\mathbf{x})\,d\mathbf{x} \quad i = 1, 2, 3.$$

- They are calculated using numerical integration.

Then the vector $\mathbf{f}$ is assembly

```
for l=1:number of elements
calculate the load vector of the element
   for i=1:3
     f(elem(i,l))=f(elem(i,l))+fel(i)
   end
end
```

# felem - 1

] Three points quadrature rule:

$$\int_K g(x)\,dx \approx \frac{|K|}{3}\sum_{k=1}^{3} g(P_k)$$

For $i = 1, 2, 3$

$$f_i = \int_K f(x)\phi_i(x)\,dx \approx \frac{|K|}{3}\sum_{k=1}^{3} f(P_k)\phi_i(P_k) = \frac{|K|}{3} f(P_i)$$

because $\phi_i(P_k) = \delta_{i,k}$.

# felem - 2

```
function f=felem(K,fun)
for i=1:3
  x=K(1,i);
  y=K(2,i);
  f(i)=eval(fun);
end
for i=1:2
   B(:,i)=K(:,i+1)-K(:,1);
end
f=f*abs(det(B))/6;        % abs(det(B))/2 is the area of K
```

# Input data

- The matrix `coor` and the matrix `elem` giving the triangulation.
- A vector `bc` giving the nodes in the boundary.
- The load $f$ (as a string).

# The program

```
function sol=femDir(coor,elem,bc,fun)
ndof=length(coor);
nelem=length(elem);
A=zeros(ndof);
F=zeros(ndof,1);
for l=1:nelem
  K=coor(:,elem(:,l));
  s=selem(K);
  f=felem(K,fun);
  for i=1:3
    for j=1:3
      A(elem(i,l),elem(j,l))=A(elem(i,l),elem(j,l))+s(i,j);
    end
    F(elem(i,l))= F(elem(i,l))+f(i);
  end
end
bound=find(bc(1,:));
intern=find(bc(1,:)-1);
A=A(intern,intern);
F=F(intern);
x=A\F;
sol=zeros(ndof,1);
sol(intern)=x;
sol(bound)=0;
```

# creadata

```
function [coor,elem,bc]=creadata(N)
ndof=(N+1)^2;
for j=1:N+1
  for i=1:N+1
    coor(1,(j-1)*(N+1)+i)=(i-1)/N;
    coor(2,(j-1)*(N+1)+i)=(j-1)/N;
  end
end
elem=[];
for j=1:N
  for i=1:N
    k=(j-1)*(N+1)+i;
    elemA=[k;k+1;k+N+1];
    elemB=[k+1;k+1+N+1;k+N+1];
    elem=[elem elemA elemB];
  end
end
```

# Example

Solve

$$\begin{cases} -\Delta u = 8\pi^2 \sin(2\pi x)\sin(2\pi y) & \text{in } \Omega = (0,1)\times(0,1) \\ u = 0 & \text{on } \partial\Omega \end{cases}$$

Solution: $u = \sin(2\pi x)\sin(2\pi y)$.