TD1

How to get the files ?

Download: http://dms.mat.mines-paristech.fr/Programme/Module-B2/ **codes_matlab.tar**

- Open a terminal and go to your homedirectory « perso » === > **cd**
- Create a new repertory ============================= > **mkdir TD_EF**
- Go in TD_EF ============================= > **cd perso/TD_EF**

- Move « codes_matlab.tar » in TD_EF
  & extract (uncompress) files ============= > **tar xvf codes_matlab.tar**

How to run the programs ?

Matlab code repertory ==== in ==== > perso/TD_EF/codes_matlab

Run matlab code: **cd perso/TD_EF/code_matab**
./matlab ./

GMSH repertory ==== in ==== > perso/TD_EF/gmsh-1.60.1

Run GMSH: **cd /usr/local/gmsh-1.60.1/**
./gmsh

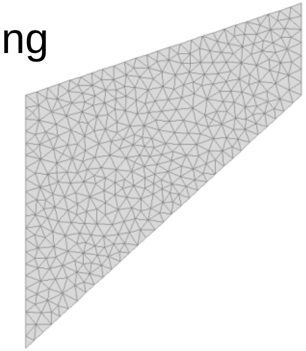
Introduction

Go to matlab codes directory : **cd perso/TD_EF/codes_matlab**
Run matlab: **./matlab ./**

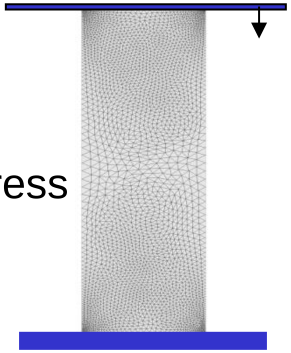
- Some examples: Input file (perso/TD_EF/codes_matlab**/**input) :

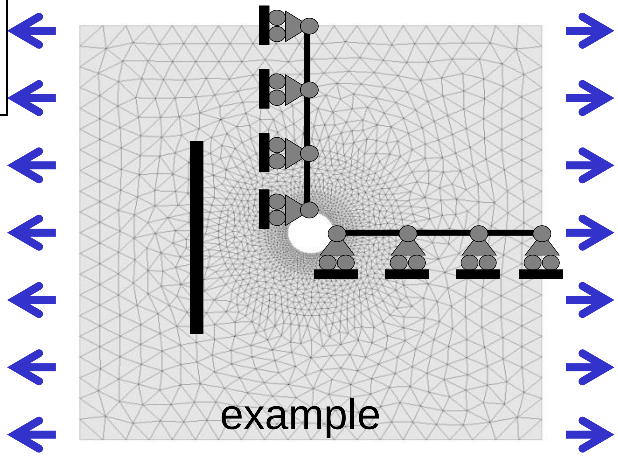
- 3 types of files:
  - *.geo: input file gmsh (geometric description)
  - *.msh: output file gmsh (Mesh description)
  - *.inp: input file matlab_code

wing



Compress





- Tutorial code

linel_T.m (« pedagogical » version)

Main programm.
Run the code: linel_T (on the matlab command window)

linel_T_fast.m (« faster » version of linel_T)

Main steps ...

1- Geometric description of the structure: **== > intput/file.geo**

Users: *- geometric definition*
- physical line, surface
- mesh refinement parameter

2- Mesh generation: **== > input/file.msh**

Gmsh: - run gmsh and open file.geo
- Options: mesh, 2D, *first order (T3) ou second order (T6)*
- save (== > file.msh)

3- Problem definition **== > input/file.inp**

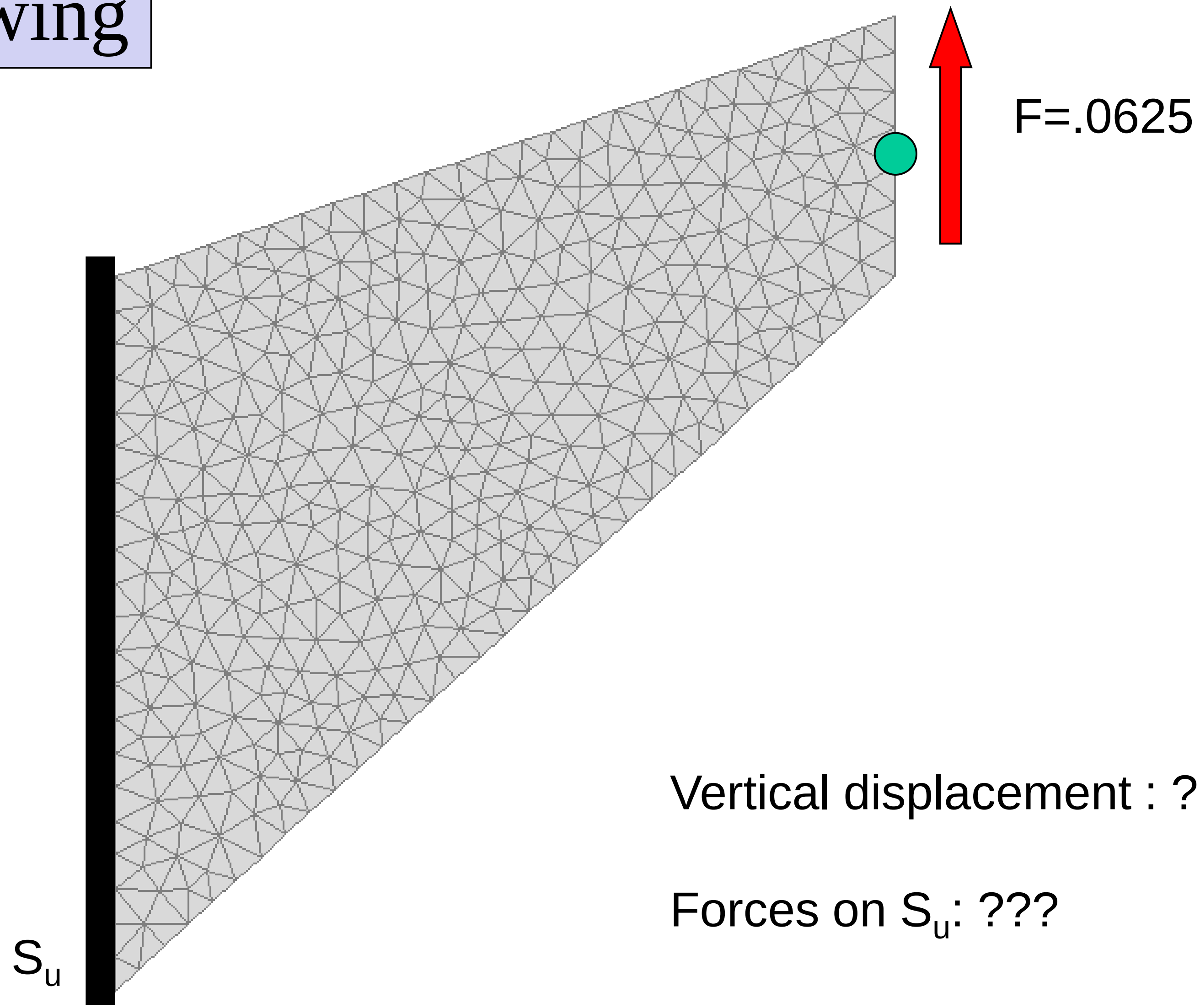
Users : *- define the problem to be solved*
(static, dynamic, ..., material properties, boundary conditions ….)

4- Compute the Finite Element approximation

Matlab: - **run linel_T** or **linel_T_fast** ….

Airplane wing: a simplify model !

Airplane wing



Vertical displacement : ???

Forces on S_u: ???

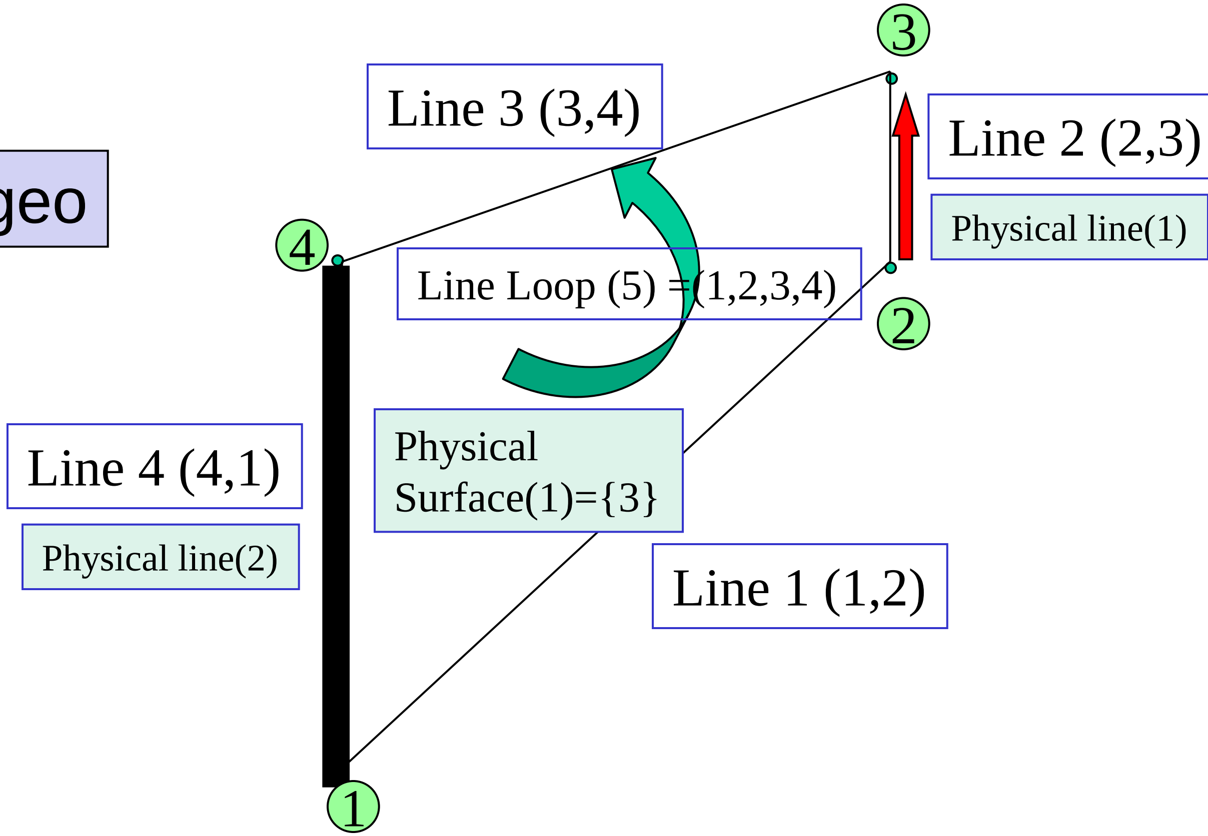
Mesh generation: wing.geo

Input file : input/wing.geo

Mesh refinement parameter

wing.geo

```
lc1 = 20;

Point(1) = {0.0,0.0,0.0,lc1};
Point(2) = {48.0,44.0,0.0,lc1};
Point(3) = {48.0,60.0,0.0,lc1};
Point(4) = {0.0,44.0,0.0,lc1};

Line(1) = {1,2};
Line(2) = {2,3};
Line(3) = {3,4};
Line(4) = {4,1};
Line Loop(5) = {1,2,3,4};

Plane Surface(1) = {5};

Physical Line(1) = {2};
Physical Line(2) = {4};
Physical Surface(3) = {1};
```



Physical Line (Boundary conditions)

- Physical Line(1) : (traction force)
  Line 2: given traction force
- Physical Line(2) : (displacement)
  Line 4: embedding (imposed displacement)

Mesh generation: wing.geo

wing.geo

Mesh refinement parameter

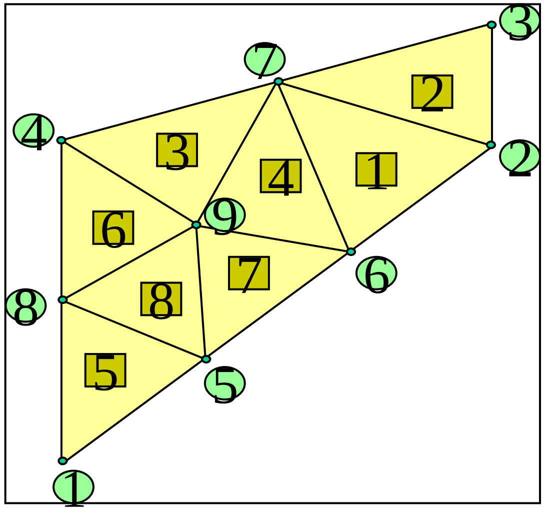
```
lc1 = 20;

Point(1) = {0.0,0.0,0.0,lc1};
Point(2) = {48.0,44.0,0.0,lc1};
Point(3) = {48.0,60.0,0.0,lc1};
Point(4) = {0.0,44.0,0.0,lc1};

Line(1) = {1,2};
Line(2) = {2,3};
Line(3) = {3,4};
Line(4) = {4,1};
Line Loop(5) = {1,2,3,4};

Plane Surface(1) = {5};

Physical Line(1) = {2};
Physical Line(2) = {4};
Physical Surface(3) = {1};
```

- Run gmsh : **cd perso/TD_EF/gmsh-1.60.1/**
  ./gmsh

- open « wing.geo »
- mesh, 2D (Shortcut: esc F3)
- *option: first order (T3) or second order (T6)*
- save (== > generation of wing.msh)

- To modify « input file » == > Edit

TO DO:
- change mesh refinement parameter
- generate the new meshes



Input data: wing.inp

Input file : input/wing.inp

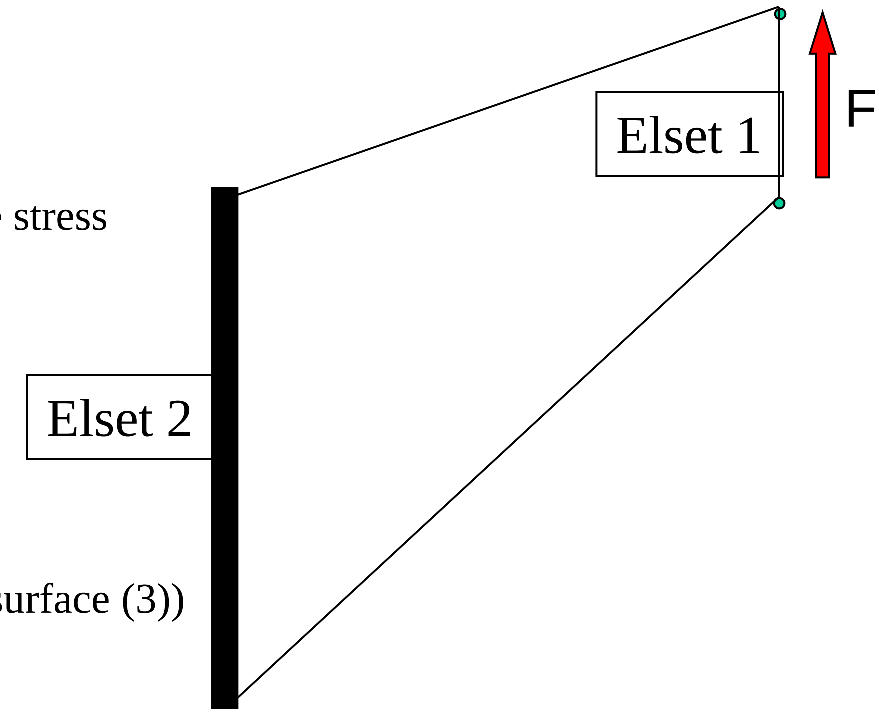
wing.inp

```
wing.msh

*ANALYSIS
STATIC,TYPE=PLANESTRESS
**

*MATERIAL,TYPE=ISOTROPIC
YOUNG=1
POISSON=.3333333
**

*SOLID
ELSET=3
**

*DBC
ELSET=2,DIR=1,VAL=0.
ELSET=2,DIR=2,VAL=0.
**

*TBC
ELSET=1,DIR=2,VAL=.0625
**

*ENDFILE
```

Analyse : Static, plane stress

Material caracteristic

Domain:
ELSET=3 (Physical surface (3))



Boundary conditions :

Displacement: DBC
ELSET = 2 (Physical Line(2))
Direction 1 ; Value = 0
Direction 2 ; Value = 0

Traction : TBC
ELSET = 1 (Physical Line(1))
Direction 2 :
Value F=0.0625

Input data: wing.inp

wing.geo

```
wing.msh

*ANALYSIS
STATIC,TYPE=PLANESTRESS
**

*MATERIAL,TYPE=ISOTROPIC
YOUNG=1
POISSON=.3333333
**

*SOLID
ELSET=3
**

*DBC
ELSET=2,DIR=1,VAL=0.
ELSET=2,DIR=2,VAL=0.
**

*TBC
ELSET=1,DIR=2,VAL=.0625
**

*ENDFILE
```

Run matlab :

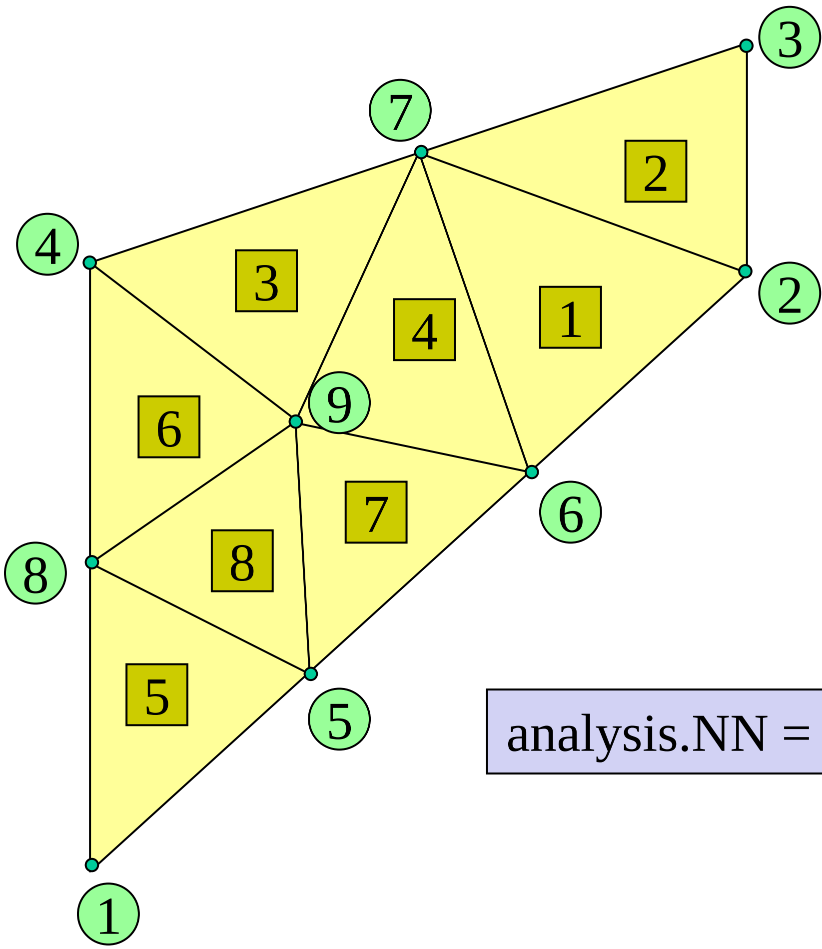
Linel_T.m

But before ….

… let’s have a look on the main program

Tables: Analysis, materials

```
[analysis materials connec coor...                    % reads input file
   dof displ TD]=read_input(pname,fname,t2);
```




```
analysis =

       NN:  9
       NE:  8
     Etag:  '3'
       ne:  3
       ns:  2
       ng:  1
     type:  'PLANESTRESS'
      nTD:  1
      neq:  12
```

nodes
surface elt
'3' ==>T3 ; '6' ==> T6
#
#
Gauss points
segment (imposed traction)
unknown

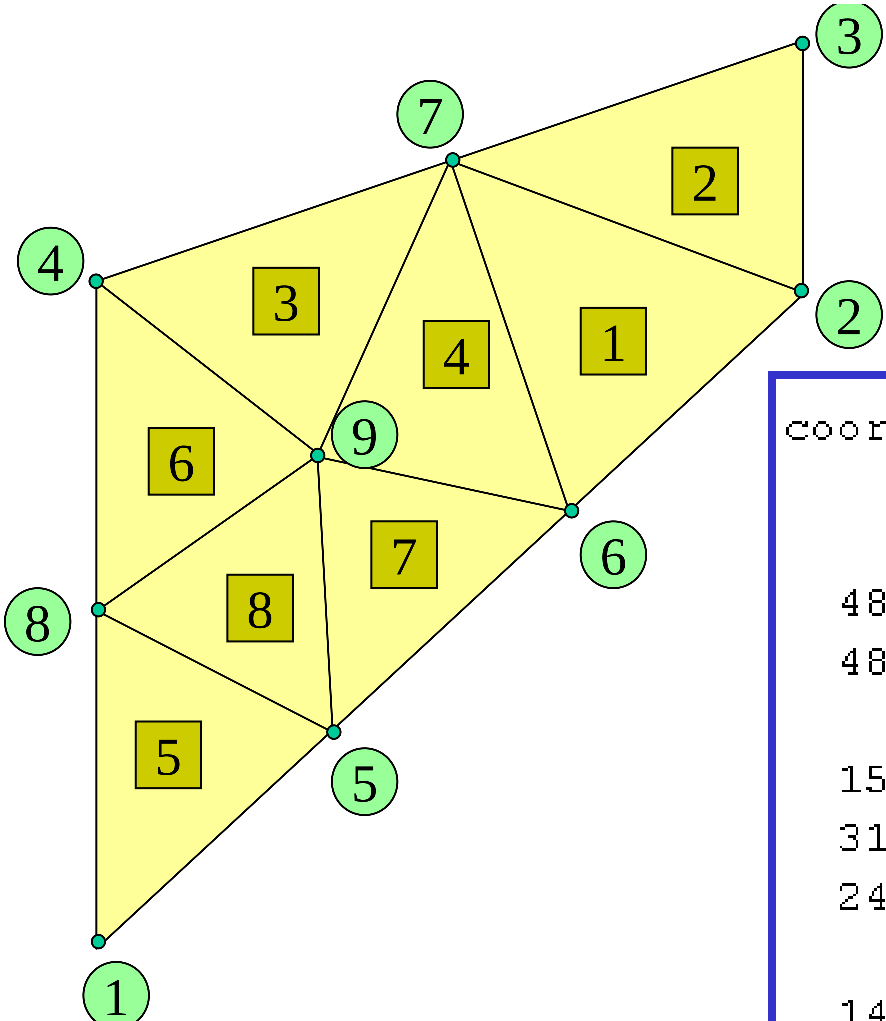
analysis.NN = 9

```
materials =

       type:  [1x11 char]
      young:  1
    poisson:  0.33333330000000
          A:  [3x3 double]
```

File : utility/read_input.m

Tables: connec, coor

```
[analysis materials connec coor...                          %
    dof displ TD]=read_input(pname,fname,t2);
```

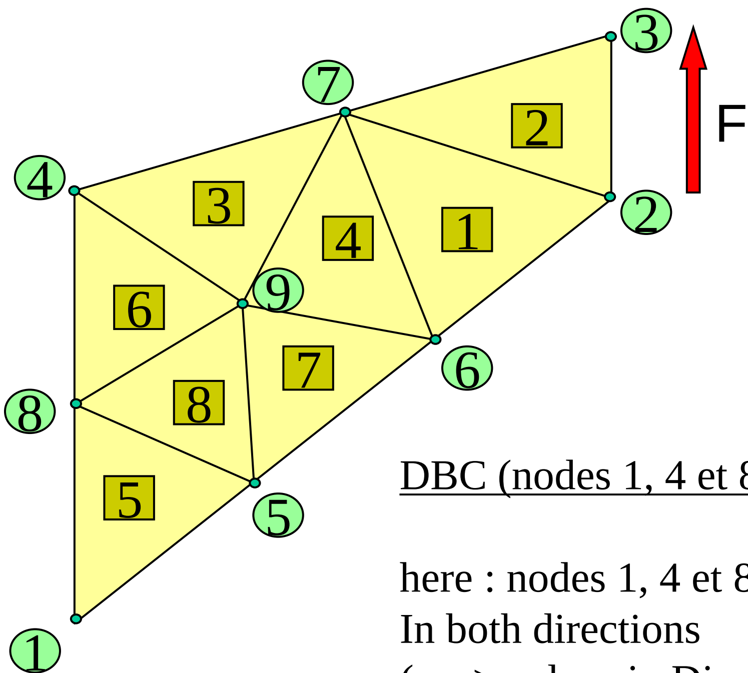



```
connec =

     6     2     7
     2     3     7
     4     9     7
     7     9     6
     1     5     8
     8     9     4
     6     9     5
     5     9     8
```

```
coor =

                  0                   0
  48.00000000000000   44.00000000000000
  48.00000000000000   60.00000000000000
                  0   44.00000000000000
  15.99999999999980   14.66666666666648
  31.99999999999951   29.33333333333289
  24.00000000000040   52.00000000000014
                  0   22.00000000000000
  14.36198494227833   32.70159631834299
```

Tables: dof & TD & displ

```
[analysis materials connec coor...                    % reads input file
    dof displ TD]=read_input(pname,fname,t2);
```



DBC (nodes 1, 4 et 8) :

here : nodes 1, 4 et 8
In both directions
(== > values in Displ)

```
dof =

    -1    -10
     1      2
     3      4
    -4    -13
     5      6
     7      8
     9     10
    -8    -17
    11     12
```

Displ =

0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0

TD (nodes 2 et 3) :

Direction : 2
Value : 0.0625

```
TD =

      dir: 2
      val: 0.06250000000000
    nodes: [2 3]
```

Initialisation of the Displacement vector:

- Zero (for unknown dof)
- Value of the imposed displacement for other nodes

Problem formulation

trouver $\underline{u} \in \mathcal{C}(\underline{u}^{\rm D})$ tel que

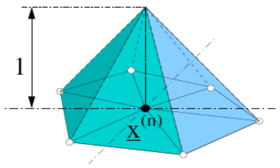
$$\int_\Omega \underline{\underline{\varepsilon}}[\underline{u}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, {\rm d}V = \int_\Omega \rho \underline{f} . \underline{w} \, {\rm d}V + \int_{S_T} \underline{T}^{\rm D} . \underline{w} \, {\rm d}S \qquad \forall \underline{w} \in \mathcal{C}(\underline{0}).$$

$$\underline{u}_h(\underline{x}) = \underline{u}_h^{(\rm D)}(\underline{x}) + \underline{u}_h^{(0)}(\underline{x}) \quad \in \mathcal{C}_h(\underline{u}^{\rm D}) \qquad (\underline{x} \in \Omega_h),$$

$$\text{avec} \quad \underline{u}_h^{(\rm D)}(\underline{x}) = \sum_{(n,j)\,|\,\texttt{dof}(n,j)=0} \tilde{N}_n(\underline{x}) u_j^{\rm D}(\underline{x}^{(n)}) \underline{e}_j \qquad \in \mathcal{C}_h(\underline{u}^{\rm D})$$

$$\text{et} \quad \underline{u}^{(0)} = \sum_{(n,j)\,|\,\texttt{dof}(n,j)>0} \tilde{N}_n(\underline{x}) u_j^{(n)} \underline{e}_j \qquad \in \mathcal{C}_h(\underline{0});$$

$$\underline{w}(\underline{x}) = \sum_{(n,j)\,|\,\texttt{dof}(n,j)>0} \tilde{N}_n(\underline{x}) w_j^{(n)} \underline{e}_j \quad \in \mathcal{C}_h(\underline{0}) \qquad (\underline{x} \in \Omega_h).$$



Trouver $\underline{u}_h^{(0)} \in \mathcal{C}_h(\underline{0})$ tel que : $\forall \underline{w} \in \mathcal{C}_h(\underline{0})$,

$$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, {\rm d}V = - \int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\rm D)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, {\rm d}V + \int_{\Omega_h} \underline{f} . \underline{w} \, {\rm d}V + \int_{S_{T,h}} \underline{T}^{\rm D} . \underline{w} \, {\rm d}S.$$

Problem formlation

Trouver $\underline{u}_h^{(0)} \in \mathcal{C}_h(\underline{0})$ tel que : $\forall \underline{w} \in \mathcal{C}_h(\underline{0})$,

$$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}]\, \mathrm{d}V = -\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}]\, \mathrm{d}V + \int_{\Omega_h} \underline{f} . \underline{w}\, \mathrm{d}V + \int_{S_{T,h}} \underline{T}^{\mathrm{D}} . \underline{w}\, \mathrm{d}S.$$

1- $$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}]\, \mathrm{d}V = \sum_{e=1}^{\mathrm{N_E}} \int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}]\, \mathrm{d}V$$

$$\int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}]\, \mathrm{d}V = \{\mathbb{W}_e\}^{\mathrm{T}} [\mathbb{K}_e] \{\mathbb{U}_e\}$$

avec $$[\mathbb{K}_e] = \int_{\Delta_e} [B_e(\underline{a})]^{\mathrm{T}} [A] [B_e(\underline{a})] J_e(\underline{a})\, \mathrm{d}V(\underline{a})$$

2- $$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}]\, \mathrm{d}V = \sum_{e=1}^{\mathrm{N_E}} \int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}]\, \mathrm{d}V$$

Local stiffness matrix: Ke
Directory: ElementName
Program : Stiff_linel_**ElementName**.m

3- $$\int_{\Omega_h} \underline{f} . \underline{w}\, \mathrm{d}V + \int_{S_{T,h}} \underline{T}^{\mathrm{D}} . \underline{w}\, \mathrm{d}S.$$

Local extrenal forces: Fe
Directory: ElementName
Program : nf_traction_**ElementName**.m

$$\int_{E_e} \underline{f} . \underline{w}\, \mathrm{d}V + \int_{\Gamma_{\mathrm{T}}^e} \underline{T}^{\mathrm{D}} . \underline{w}\, \mathrm{d}S = \{\mathbb{W}_e\}^{\mathrm{T}} \{\mathbb{F}_e^{\mathrm{vol}} + \mathbb{F}_e^{\mathrm{surf}}\} = \{\mathbb{W}_e\}^{\mathrm{T}} \{\mathbb{F}_e^{\mathrm{ext}}\}$$

Local stifness matrix

Local Stiffness Matrix: Ke

Directory: ElementName
Program : Stiff_linel_ElementName.m

$$\int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h] : \mathcal{A} : \underline{\underline{\varepsilon}}[\underline{w}] \, \mathrm{d}V = \{\mathbb{W}_e\}^{\mathrm{T}} [\mathbb{K}_e] \{\mathbb{U}_e\}$$

$$\text{avec} \quad [\mathbb{K}_e] = \int_{\Delta_e} [B_e(\underline{a})]^{\mathrm{T}} [A] [B_e(\underline{a})] J_e(\underline{a}) \, \mathrm{d}V(\underline{a}) \approx \sum_{g=1}^{G} w_g [B_e(\underline{a}_g)]^{\mathrm{T}} [A] [B_e(\underline{a}_g)] J_e(\underline{a}_g)$$

```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Linear elastic stiffness matrix: T3
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function Ke=stiff_linel_T3(T,A)

x11=T(1,1); x21=T(2,1); x31=T(3,1);
x12=T(1,2); x22=T(2,2); x32=T(3,2);
S=.5*((x21-x11)*(x32-x12)-...
      (x31-x11)*(x22-x12));
Be=[x22-x32,0,x32-x12,0,x12-x22,0;
    0,x31-x21,0,x11-x31,0,x21-x11;
    x31-x21,x22-x32,x11-x31, ...
    x32-x12,x21-x11,x12-x22]/(2*S);
Ke=S*Be'*A*Be;
```

```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Linear elastic stiffness matrix: T6
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function Ke=stiff_linel_T6(T,A)

a_gauss=1/6*[4 1 1; 1 4 1; 1 1 4];
w_gauss=[1/6 1/6 1/6];
Ke=zeros(12,12);
for g=1:3,
  a=a_gauss(g,:);
  DN=[4*a(1)-1 0 -4*a(3)+1 4*a(2) ...
      -4*a(2) 4*(a(3)-a(1));
      0 4*a(2)-1 -4*a(3)+1 4*a(1) ...
      4*(a(3)-a(2)) -4*a(1)]';
  J=T'*DN;
  detJ=J(1,1)*J(2,2)-J(1,2)*J(2,1);
  invJ=1/detJ*[ J(2,2) -J(1,2); ...
               -J(2,1)  J(1,1)];
  GN=DN*invJ;
  Be=[GN(1,1) 0 GN(2,1) 0 GN(3,1) 0 ...
      GN(4,1) 0 GN(5,1) 0 GN(6,1) 0;
      0 GN(1,2) 0 GN(2,2) 0 GN(3,2)...
      0 GN(4,2) 0 GN(5,2) 0 GN(6,2);
      GN(1,2) GN(1,1) GN(2,2) GN(2,1)...
      GN(3,2) GN(3,1) GN(4,2) GN(4,1)...
      GN(5,2) GN(5,1) GN(6,2) GN(6,1)];
  Ke=Ke+Be'*A*Be*detJ*w_gauss(g);
end
```

Local external forces

Local external forces : Fe

Directory: ElementName
Program : nf_traction_ElementName.m

$$\int_{\Gamma_T^e} \underline{T}^D . \underline{w} \, dS = \{\mathbb{W}_e\}^T \left\{ \int_\Sigma [N(\underline{b})]^T \{\underline{T}^D(\underline{x}(\underline{b}))\} \hat{J}(\underline{b}) \, db_1 db_2 \right\} = \{\mathbb{W}_e\}\{\mathbb{F}_e^{surf}\}$$

```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Nodal forces due to surface tractions:
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function Fe=nf_tractions_T3(L,val,idir)

x11=L(1,1); x12=L(1,2);
x21=L(2,1); x22=L(2,2);
H=sqrt((x21-x11)^2+(x22-x12)^2);
NL=H/2*[1 0 1 0;
        0 1 0 1];
TD=zeros(2,1);
if idir>0,
  TD(idir)=val;
else
  n=[(x22-x12); -(x21-x11)]/H;
  TD=val*n;
end
Fe=NL'*TD;
```

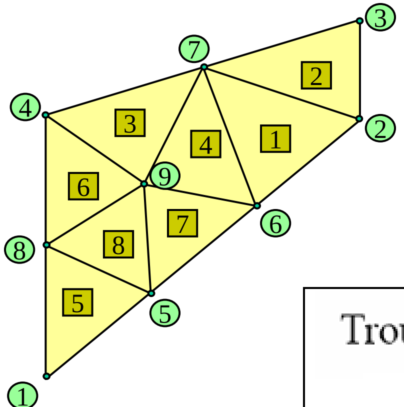
Pressure loading : idir=0

```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Nodal forces due to surface tractions: T6
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function Fe=nf_tractions_T6(L,val,idir)

a_gauss=[-1/sqrt(3) 1/sqrt(3)];
w_gauss=[1 1];
Fe=zeros(6,1);
for g=1:2
 a=a_gauss(g);
 DN=[a-.5  a+.5 -2*a];
 J=DN*L;
 detJ=sqrt(J(1)^2+J(2)^2);
 TD=zeros(2,1);
 if idir>0,
  TD(idir)=val;
 else
  TD=val/detJ*[J(2) -J(1)]';
 end
 NL=[.5*a*(a-1) .5*a*(1+a) 1-a^2];
 N=[NL(1) 0 NL(2) 0 NL(3) 0;
    0 NL(1) 0 NL(2) 0 NL(3)];
 Fe=Fe+N'*TD*detJ*w_gauss(g);
end
```

Assembly

Trouver $\underline{u}_h^{(0)} \in \mathcal{C}_h(\underline{0})$ tel que : $\forall \underline{w} \in \mathcal{C}_h(\underline{0})$,

$$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, \mathrm{d}V = - \int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, \mathrm{d}V + \int_{\Omega_h} \underline{f} . \underline{w} \, \mathrm{d}V + \int_{S_{T,h}} \underline{T}^{\mathrm{D}} . \underline{w} \, \mathrm{d}S.$$



$$\{\mathbb{W}\} = \{w_1^{(2)}, w_2^{(2)}, w_1^{(3)}, w_2^{(3)}, w_1^{(5)}, w_2^{(5)}, w_1^{(6)}, w_2^{(6)}, w_1^{(7)}, w_2^{(7)}, w_1^{(9)}, w_2^{(9)}\}^T$$

$$\{\mathbb{U}\} = \{u_1^{(2)}, u_2^{(2)}, u_1^{(3)}, u_2^{(3)}, u_1^{(5)}, u_2^{(5)}, u_1^{(6)}, u_2^{(6)}, u_1^{(7)}, u_2^{(7)}, u_1^{(9)}, u_2^{(9)}\}^T$$

Trouver $\{\mathbb{U}\} \in \mathbb{R}^{\mathrm{N}}$ tel que : $\forall \{\mathbb{W}\} \in \mathbb{R}^{\mathrm{N}}$, $\quad \{\mathbb{W}\}^{\mathrm{T}}[\mathbb{K}]\{\mathbb{U}\} = \{\mathbb{W}\}^{\mathrm{T}}\{\mathbb{F}\}$,

$$\{\mathbb{W}\}^{\mathrm{T}}[\mathbb{K}]\{\mathbb{U}\} = \sum_{e=1}^{\mathrm{N_E}} \int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, \mathrm{d}V,$$

$$\{\mathbb{W}\}^{\mathrm{T}}\{\mathbb{F}\} = \sum_{e=1}^{\mathrm{N_E}} \left\{ - \int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, \mathrm{d}V + \int_{E_e} \underline{f} . \underline{w} \, \mathrm{d}V + \int_{\Gamma_{\mathrm{T}}^e} \underline{T}^{\mathrm{D}} . \underline{w} \, \mathrm{d}S, \right\}$$

$$==> \quad [K]\{U\} = \{F\} \quad \text{with} \; \{F\} = \{F^{vol}\} + \{F^{surf}\} + \{F^{U}\}$$

Assembly

Trouver $\underline{u}_h^{(0)} \in \mathcal{C}_h(\underline{0})$ tel que : $\forall \underline{w} \in \mathcal{C}_h(\underline{0})$,

$$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \,\mathrm{d}V = -\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \,\mathrm{d}V + \int_{\Omega_h} \underline{f}.\underline{w} \,\mathrm{d}V + \int_{S_{T,h}} \underline{T}^{\mathrm{D}}.\underline{w} \,\mathrm{d}S.$$

$$[K]\{U\} = \{F\}$$

with

$$\{F\} = \{F^{vol}\} + \{F^{surf}\} + \{F^{U}\}$$

1- $$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \,\mathrm{d}V = \sum_{e=1}^{\mathrm{N_E}} \int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \,\mathrm{d}V$$

Contribution to the global stiffness matrix: [K]

$$\int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \,\mathrm{d}V = \{\mathbb{W}_e\}^{\mathrm{T}}[\mathbb{K}_e]\{\mathbb{U}_e\}$$

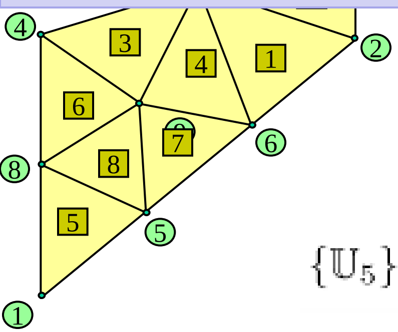
avec $$[\mathbb{K}_e] = \int_{\Delta_e} [B_e(\underline{a})]^{\mathrm{T}}[A][B_e(\underline{a})] J_e(\underline{a}) \,\mathrm{d}V(\underline{a})$$

Contribution to the global right hand side (imposed displacement): {F^u}

2- $$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \,\mathrm{d}V = \sum_{e=1}^{\mathrm{N_E}} \int_{E_e} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \,\mathrm{d}V$$

Boundary condition: displacement

Example : Element 5



Trouver $\underline{u}_h^{(0)} \in \mathcal{C}_h(\underline{0})$ tel que : $\forall \underline{w} \in \mathcal{C}_h(\underline{0})$,

$$\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(0)}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, \mathrm{d}V = - \int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h^{(\mathrm{D})}] : \boldsymbol{\mathcal{A}} : \underline{\underline{\varepsilon}}[\underline{w}] \, \mathrm{d}V + \int_{\Omega_h} \underline{f} . \underline{w} \, \mathrm{d}V + \int_{S_{T,h}} \underline{T}^{\mathrm{D}} . \underline{w} \, \mathrm{d}S.$$

$$\{\mathbb{U}_5\} = \{u_1^{(D1)}, u_2^{(D1)}, u_1^{(5)}, u_2^{(5)}, u_1^{(D8)}, u_2^{(D8)}\}^T \qquad \{\mathbb{W}_5\} = \{0, 0, w_1^{(5)}, w_2^{(5)}, 0, 0\}^T$$

$$\{\mathbb{U}_5^{(D)}\} = \{u_1^{(D1)}, u_2^{(D1)}, u_1^{(D8)}, u_2^{(D8)}\}^T \qquad \{\mathbb{U}_5^{(0)}\} = \{u_1^{(5)}, u_2^{(5)}\}^T \qquad \{\mathbb{W}_5^{(0)}\} = \{w_1^{(5)}, w_2^{(5)}\}^T$$

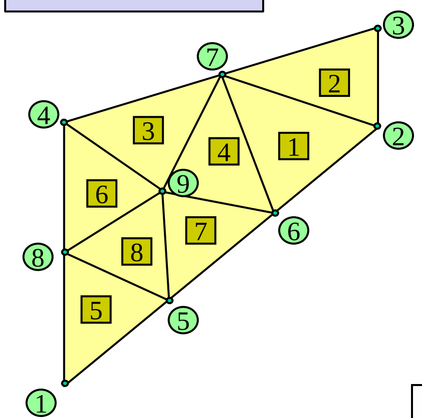
$$\{W_{e=5}\}^T [K_{e=5}] \{U_{e=5}\} = \{0 \quad 0 \quad w_1^{(5)} \quad w_2^{(5)} \quad 0 \quad 0\} \begin{bmatrix} K_{e=5}^{(11)} & K_{e=5}^{(12)} & K_{e=5}^{(13)} & K_{e=5}^{(14)} & K_{e=5}^{(15)} & K_{e=5}^{(16)} \\ K_{e=5}^{(21)} & K_{e=5}^{(22)} & K_{e=5}^{(23)} & K_{e=5}^{(24)} & K_{e=5}^{(25)} & K_{e=5}^{(26)} \\ K_{e=5}^{(31)} & K_{e=5}^{(32)} & K_{e=5}^{(33)} & K_{e=5}^{(34)} & K_{e=5}^{(35)} & K_{e=5}^{(36)} \\ K_{e=5}^{(41)} & K_{e=5}^{(42)} & K_{e=5}^{(43)} & K_{e=5}^{(44)} & K_{e=5}^{(45)} & K_{e=5}^{(46)} \\ K_{e=5}^{(51)} & K_{e=5}^{(52)} & K_{e=5}^{(53)} & K_{e=5}^{(54)} & K_{e=5}^{(55)} & K_{e=5}^{(56)} \\ K_{e=5}^{(61)} & K_{e=5}^{(62)} & K_{e=5}^{(63)} & K_{e=5}^{(64)} & K_{e=5}^{(65)} & K_{e=5}^{(66)} \end{bmatrix} \begin{Bmatrix} U_1^{(1)} \\ U_2^{(1)} \\ U_1^{(5)} \\ U_2^{(5)} \\ U_1^{(8)} \\ U_1^{(8)} \end{Bmatrix}$$

$$\{W_{e=5}\}^T [K_{e=5}] \{U_{e=5}\} = \{W_{e=5}^0\}^T \begin{bmatrix} K_{e=5}^{(33)} & K_{e=5}^{(34)} \\ K_{e=5}^{(43)} & K_{e=5}^{(44)} \end{bmatrix} \{U_{e=5}^0\} + \{W_{e=5}^0\}^T \begin{bmatrix} K_{e=5}^{(31)} & K_{e=5}^{(32)} & K_{e=5}^{(35)} & K_{e=5}^{(36)} \\ K_{e=5}^{(41)} & K_{e=5}^{(42)} & K_{e=5}^{(45)} & K_{e=5}^{(46)} \end{bmatrix} \{U_{e=5}^D\}$$

$$\{\mathbb{W}_5\}^T [\mathbb{K}_5] \{\mathbb{U}_5\} = \{\mathbb{W}_5^{(0)}\}^T [\mathbb{K}_5^{(00)}] \{\mathbb{U}_5^{(0)}\} + \{\mathbb{W}_5^{(0)}\}^T [\mathbb{K}_5^{(0D)}] \{\mathbb{U}_5^{(D)}\}$$

Boundary condition: displacement

Element 5



$$\{W_{e=5}\}^T[K_{e=5}]\{U_{e=5}\} = \begin{Bmatrix} 0 \\ 0 \\ w_1^{(5)} \\ w_2^{(5)} \\ 0 \\ 0 \end{Bmatrix}^T \begin{bmatrix} K_{e=5}^{(11)} & K_{e=5}^{(12)} & K_{e=5}^{(13)} & K_{e=5}^{(14)} & K_{e=5}^{(15)} & K_{e=5}^{(16)} \\ K_{e=5}^{(21)} & K_{e=5}^{(22)} & K_{e=5}^{(23)} & K_{e=5}^{(24)} & K_{e=5}^{(25)} & K_{e=5}^{(26)} \\ K_{e=5}^{(31)} & K_{e=5}^{(32)} & K_{e=5}^{(33)} & K_{e=5}^{(34)} & K_{e=5}^{(35)} & K_{e=5}^{(36)} \\ K_{e=5}^{(41)} & K_{e=5}^{(42)} & K_{e=5}^{(43)} & K_{e=5}^{(44)} & K_{e=5}^{(45)} & K_{e=5}^{(46)} \\ K_{e=5}^{(51)} & K_{e=5}^{(52)} & K_{e=5}^{(53)} & K_{e=5}^{(54)} & K_{e=5}^{(55)} & K_{e=5}^{(56)} \\ K_{e=5}^{(61)} & K_{e=5}^{(62)} & K_{e=5}^{(63)} & K_{e=5}^{(64)} & K_{e=5}^{(65)} & K_{e=5}^{(66)} \end{bmatrix} \begin{Bmatrix} U_1^{(1)} \\ U_2^{(1)} \\ U_1^{(5)} \\ U_2^{(5)} \\ U_1^{(8)} \\ U_1^{(8)} \end{Bmatrix}$$

$$\{\mathbb{W}_5\}^T[\mathbb{K}_5]\{\mathbb{U}_5\} = \{\mathbb{W}_5^{(0)}\}^T[\mathbb{K}_5^{(00)}]\{\mathbb{U}_5^{(0)}\} + \{\mathbb{W}_5^{(0)}\}^T[\mathbb{K}_5^{(0D)}]\{\mathbb{U}_5^{(D)}\}$$

```
nodes=connec(e,:)                 →   nodes=[1 5 8]
dofe=dof(nodes,:)                 →   dofe=[-1  -10;
                                              5    6;
                                             -8  -17]
dofe=reshape(dofe',[1,6])         →   dofe=[-1 -10 5  6 -8 -17]
pe=find(dofe>0);                  →   pe=[3 4];'Local' numbering
Ie=dofe(pe);                      →   Ie=[5 6];'Global'numbering
K(Ie,Ie)=K(Ie,Ie)+Ke(pe,pe);
pe_UDe=find(dofe<0);              →   pe_UDe=[1 2 5 6];
Ie_UDe=-dofe(pe_UDe);             →   Ie_UDe=[1 10 8 17];
UDe=displ(Ie_UDe)';               →   UDe=[0 0 0 0];
F(Ie)=F(Ie)-Ke(pe,pe_UDe)*UDe
```

Contribution to [K]

```
dof =

    -1   -10
     1     2
     3     4
    -4   -13
     5     6
     7     8
     9    10
    -8   -17
    11    12
```

Contribution to {F}

Matlab translation ...

```
D=2;
Dne=D*analysis.ne;
Dns=D*analysis.ns;
for e=1:analysis.NE,
  nodes=connec(e,:);
  T=coor(nodes,:);
  Ke=eval(['stiff_linel_T',analysis.Etag,...
                '(T,materials.A)']);
  dofe=reshape(dof(nodes,:)',[1,Dne]);
  pe=find(dofe>0);
  Ie=dofe(pe);
  K(Ie,Ie)=K(Ie,Ie)+Ke(pe,pe);
  pe_UDe=find(dofe<0);
  Ie_UDe=-dofe(pe_UDe);
  UDe=displ(Ie_UDe)';
  F(Ie)=F(Ie)-Ke(pe,pe_UDe)*UDe;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% assemblage phase: nodal loads due to surfac
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for s=1:analysis.nTD,
  nodes=TD(s).nodes;
  idir=TD(s).dir;
  val=TD(s).val;
  L=coor(nodes,:);
  Fe=eval(['nf_tractions_T',analysis.Etag,...
                     '(L,val,idir)']);
  dofs=reshape(dof(nodes,:)',[Dns,1]);
  ps=find(dofs>0);
  Is=dofs(ps);
  F(Is)=F(Is)+Fe(ps);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solution phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
U=K\F;
clear K F
```

```
Ke=stiff_linel_T3(T,materials.A);
```

Displacement boundary condition

```
Fe=nf_tractions_T3(L,val,idir);
```

```
analysis =

       NN: 9
       NE: 8
     Etag: '3'
       ne: 3
       ns: 2
       ng: 1
     type: 'PLANESTRESS'
      nTD: 1
      neq: 12
```

```
connec =

     6     2     7
     2     3     7
     4     9     7
     7     9     6
     1     5     8
     8     9     4
     6     9     5
     5     9     8
```

```
dof =

    -1   -10
     1     2
     3     4
    -4   -13
     5     6
     7     8
     9    10
    -8   -17
    11    12
```

Post traitement: Cauchy Stress components

Directory : ElementName
Program : StressG_linel_ElementName.m

$$\{\underline{\underline{\varepsilon}}[\underline{v}_h](\underline{x})\} = [B_e(\underline{a})]\{\mathbb{V}_e\}, \qquad \{\sigma\} = [A]\{\varepsilon\}$$

```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Stresses in one T3 element due to displa
% elastic constitutive law
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function stressG=stressG_linel_T3(T,A,Ve)

x11=T(1,1); x21=T(2,1); x31=T(3,1);
x12=T(1,2); x22=T(2,2); x32=T(3,2);
S=.5*((x21-x11)*(x32-x12)-...
      (x31-x11)*(x22-x12));
Be=[x22-x32,0,x32-x12,0,x12-x22,0;
    0,x31-x21,0,x11-x31,0,x21-x11;
    x31-x21,x22-x32,x11-x31, ...
    x32-x12,x21-x11,x12-x22]/(2*S);
stressG=(A*Be*Ve)';
```

To be compare with stiff_linel_T3 …

```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Linear elastic stiffness matrix: T3
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function Ke=stiff_linel_T3(T,A)

x11=T(1,1); x21=T(2,1); x31=T(3,1);
x12=T(1,2); x22=T(2,2); x32=T(3,2);
S=.5*((x21-x11)*(x32-x12)-...
      (x31-x11)*(x22-x12));
Be=[x22-x32,0,x32-x12,0,x12-x22,0;
    0,x31-x21,0,x11-x31,0,x21-x11;
    x31-x21,x22-x32,x11-x31, ...
    x32-x12,x21-x11,x12-x22]/(2*S);
Ke=S*Be'*A*Be;
```

```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Stresses at gauss points: T6
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function stressG=stressG_linel_T6(T,A,Ve)

a_gauss=1/6*[4 1 1; 1 4 1; 1 1 4];
stressG=zeros(3,3);
for g=1:3,
  a=a_gauss(g,:);
  DN=[4*a(1)-1  0  -4*a(3)+1  4*a(2) ...
      -4*a(2)  4*(a(3)-a(1));
      0  4*a(2)-1  -4*a(3)+1  4*a(1)  ...
      4*(a(3)-a(2))  -4*a(1)]';
  J=T'*DN;
  detJ=J(1,1)*J(2,2)-J(1,2)*J(2,1);
  invJ=1/detJ*[ J(2,2) -J(1,2); ...
               -J(2,1)  J(1,1)];
  GN=DN*invJ;
  Be=[GN(1,1)  0 GN(2,1)  0 GN(3,1)  0  ...
      GN(4,1)  0 GN(5,1)  0 GN(6,1)  0;
      0  GN(1,2)  0  GN(2,2)  0  GN(3,2)...
      0  GN(4,2)  0  GN(5,2)  0  GN(6,2);
      GN(1,2)  GN(1,1)  GN(2,2)  GN(2,1)...
      GN(3,2)  GN(3,1)  GN(4,2)  GN(4,1)...
      GN(5,2)  GN(5,1)  GN(6,2)  GN(6,1)];
  stressG(g,:)=(A*Be*Ve)';
end
```

Post traitement: Cauchy Stress components

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Post-processing phase: computation of nodal stresses vi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ic=find(dof>0);                              % finds act
displ(ic)=U(dof(ic));                        % and copie
stress=zeros(analysis.NN,3);                 % initializes "stress"
counter=zeros(analysis.NN,1);                % initializes "counter"
for e=1:analysis.NE,                         % computes nodal stresses
  nodes=connec(e,:);
  T=coor(nodes,:);                           % creates element
  Ve=reshape(displ(nodes,:)',[Dne,1]);
  stressG=eval(['stressG_line1_T',...        % computes stresses in one element ...
      analysis.Etag,'(T,materials.A,Ve)']);  % at gauss points
  stressN=eval(['G2N_T',...                  % and extrapolates to vertex nodes
      analysis.Etag,'(T,stressG)']);
  stress(nodes,:)=stress(nodes,:)+stressN;   % adds stresses to triangle nodes
  counter(nodes)=counter(nodes)+1;
end
for icomp=1:3
  stress(:,icomp)=stress(:,icomp)./counter;  % naive average of stresses
end
```

Matlab function: ic = find(dof>0);
= > returns the indices of « dof » that satisfy the condition

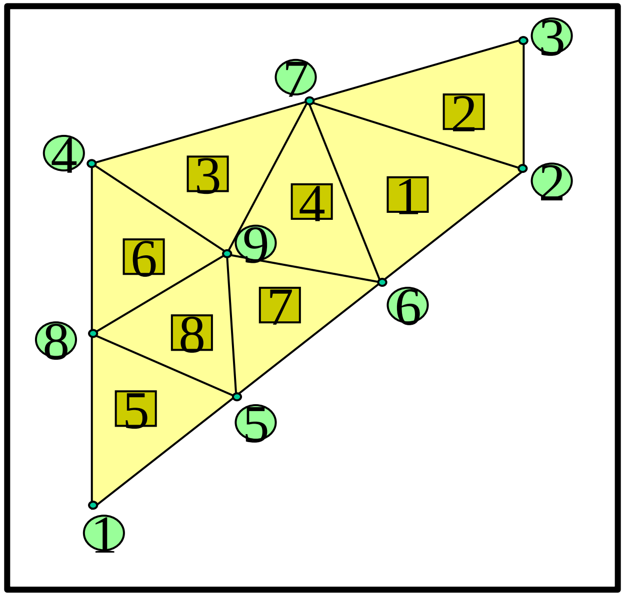
```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Extrapolation from Gauss points to nodes: T3
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function qN=G2N_T3(T,qG)

qN=[qG; qG; qG];
```

```
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% Extrapolation from Gauss points to nodes: T6
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function qN=G2N_T6(T,qG)

N=1/9*[2 -1 -1 4 1 4;
       -1 2 -1 4 4 1;
       -1 -1 2 1 4 4];
xg=[ones(3,1) N*T];                          % physical coordinates of gauss points
M=[ones(6,1) T];
qN=M*(xg\qG);                                % linear extrapolation to nodes
```

EX 1: Extract some information

Write matlab code lines:

- To display total number of element
- To display total number of nodes
- To find all nodes that belong to the line x=0



Ex 2: Assembly

- 1 - How is compute the coefficients K(5,7),K(5,10) and K(12,9) ?
  - where does it come from ?
  - which elements contribute to its computation ?
  - which coefficients of K_e contribute to its computation ?

- 2 – What is the contribution of :

  K_2(1,2), K_4(4,6) and K_5(1,6)

  - Verify with the code

```
dof =

    -1   -10
     1     2
     3     4
    -4   -13
     5     6
     7     8
     9    10
    -8   -17
    11    12
```

```
connec =

     6     2     7
     2     3     7
     4     9     7
     7     9     6
     1     5     8
     8     9     4
     6     9     5
     5     9     8
```

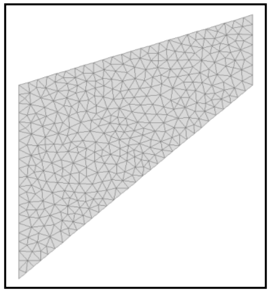
EX 3: Potential energy (DM)

- **1- Matlab code lines to evaluate the potential energy for the FE solution**

$$\mathcal{P}(\underline{u}_h) = \frac{1}{2}\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h]:\mathcal{A}:\underline{\underline{\varepsilon}}[\underline{u}_h]\,\mathrm{d}V - \int_{\Omega_h} \underline{f}.\underline{u}_h\,\mathrm{d}V - \int_{S_{T,h}} \underline{T}^{\mathrm{D}}.\underline{u}_h\,\mathrm{d}S$$

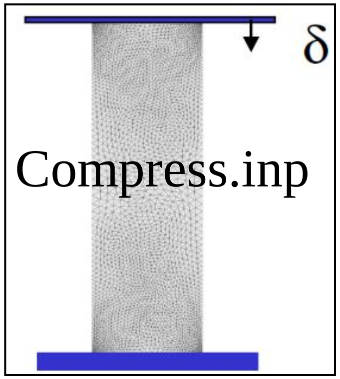
- **2- Convergence analysis (Wing case)**

  Plot the potential energy in function of the number of elements (mesh refinement) for T3 and T6 elements.



wing.inp

- **3- Compute the potential energy for the example « compress.inp »**



Work in pairs

Send by email: (clement.olivier@safran.fr)

- *Your names*
- *Matlab code lines*
- *1 pages (max) of comments and analysis*

Hand in next session

Correction EX 1: extract some information

Write matlab code lines:

- **To display total number of element:**

  == > NbElement = analysis.NE

  or == > NbElement = size(connec,1)

- **To display total number of nodes:**

  == > NbNodes = analysis.NN

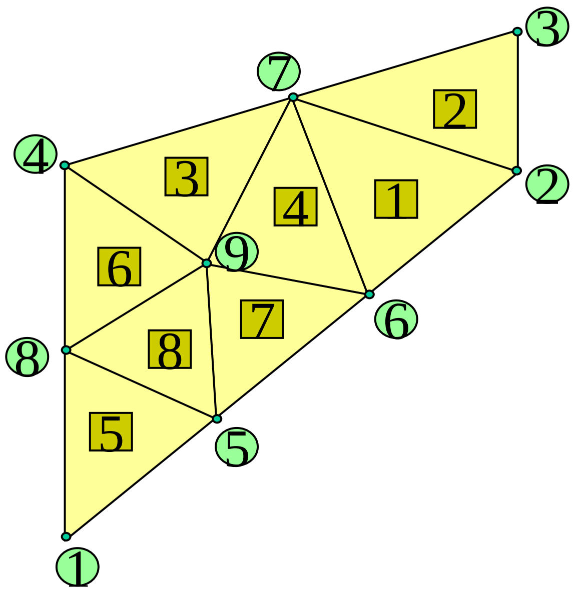
  or == > NbNodes = size(coor,1)

- **To find all nodes that belong to the line x=0**

```
SX0=find(coor(:,1)<(1.d-4));
```

```
SX0 =

   1
   4
   8
```

== > SX0 contains the list (column) of the nodes that belong to the line x=0



Correction EX 2 – Assembly (1/2)

- 1 - How is compute the coefficient K(5,7), K(5,10) and K(12,9) ?
  - where does it come from ?
  - which elements contribute to its computation ?
  - which coefficients of K_e contribute to its computation ?

K(5,7)

- Node 5 (direction 1) & Node 6 (direction 1)
- Element 7
- K(1,4) = K_7(5,1)

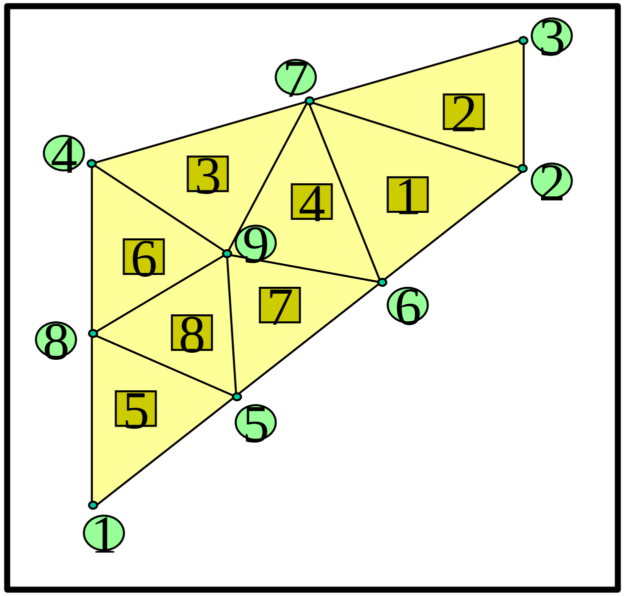
- 2 – Same question for K(5,10)

K(5,10)

- Node 5 (direction 1) & Node 7 (direction 2)
- No common support of shape functions
- K(5,11) = 0

- 3 – Same question for K(12,9)

- Node 9 (direction 2) & Node 7 (direction 1)
- Elements 3 & 4
- K(12,9) = K_3(4,5) + K_4(4,1)



dof =

-1	-10
1	2
3	4
-4	-13
5	6
7	8
9	10
-8	-17
11	12

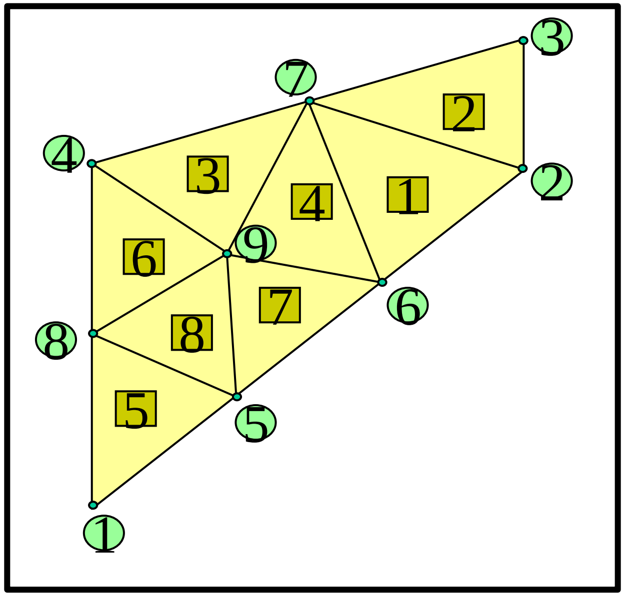
connec =

6	2	7
2	3	7
4	9	7
7	9	6
1	5	8
8	9	4
6	9	5
5	9	8

Correction EX 2 – Assembly (2/2)

- 2 – What is the contribution of :

  K_2(1,2), K_4(4,6) and [K_5(2,3) and K_5(3,2)]



K_2(1,2)

- dof_2 = 1 ↔ Node : 2 Dir 1 ↔ dof = 1
- dof_2 = 2 ↔ Node : 2 Dir 2 ↔ dof = 2
- K(1,2) ← K_2(1,2) **(but ≠)**

K_4(4,6)

- dof_4 = 4 ↔ Node : 9 Dir 2 ↔ dof = 12
- dof_4= 6 ↔ Node : 6 Dir 2 ↔ dof = 8
- K(12,8) ← K_4(4,6) **(but ≠)**

K_5(3,2) and K_5(2,3)

- dof_5 = 3 ↔ Node : 5 Dir 1 ↔ dof = 5
- dof_5 = 2 ↔ Node : 1 Dir 2 ↔ dof = -10
- F^u(5) ← K_5(3,2) *U^(1)^2 **(but ≠)**
- K_5(2,3) does not contribute

dof =

-1	-10
1	2
3	4
-4	-13
5	6
7	8
9	10
-8	-17
11	12

connec =

6	2	7
2	3	7
4	9	7
7	9	6
1	5	8
8	9	4
6	9	5
5	9	8

Hints EX 3: Potential energy

- **1- Matlab code lines to evaluate the potential energy for the FE solution**

$$\mathcal{P}(\underline{u}_h) = \frac{1}{2}\int_{\Omega_h} \underline{\underline{\varepsilon}}[\underline{u}_h] : \mathcal{A} : \underline{\underline{\varepsilon}}[\underline{u}_h]\,\mathrm{d}V - \int_{\Omega_h} \underline{f}.\underline{u}_h\,\mathrm{d}V - \int_{S_{T,h}} \underline{T}^{\mathrm{D}}.\underline{u}_h\,\mathrm{d}S$$

Prove: $P(\underline{u}_h) = \sum_e \left(\frac{1}{2}\{\mathbf{U}_e\}^T[\mathbf{K}_e]\{\mathbf{U}_e\} - \{\mathbf{U}_e\}^T\{\mathbf{F}_e^{ext}\} \right) = \frac{1}{2}\{\mathbf{U}\}^T[K]\{\mathbf{U}\} - \{\mathbf{U}\}^T\{\mathbf{F}\} + P(\underline{u}_h^{(D)})$