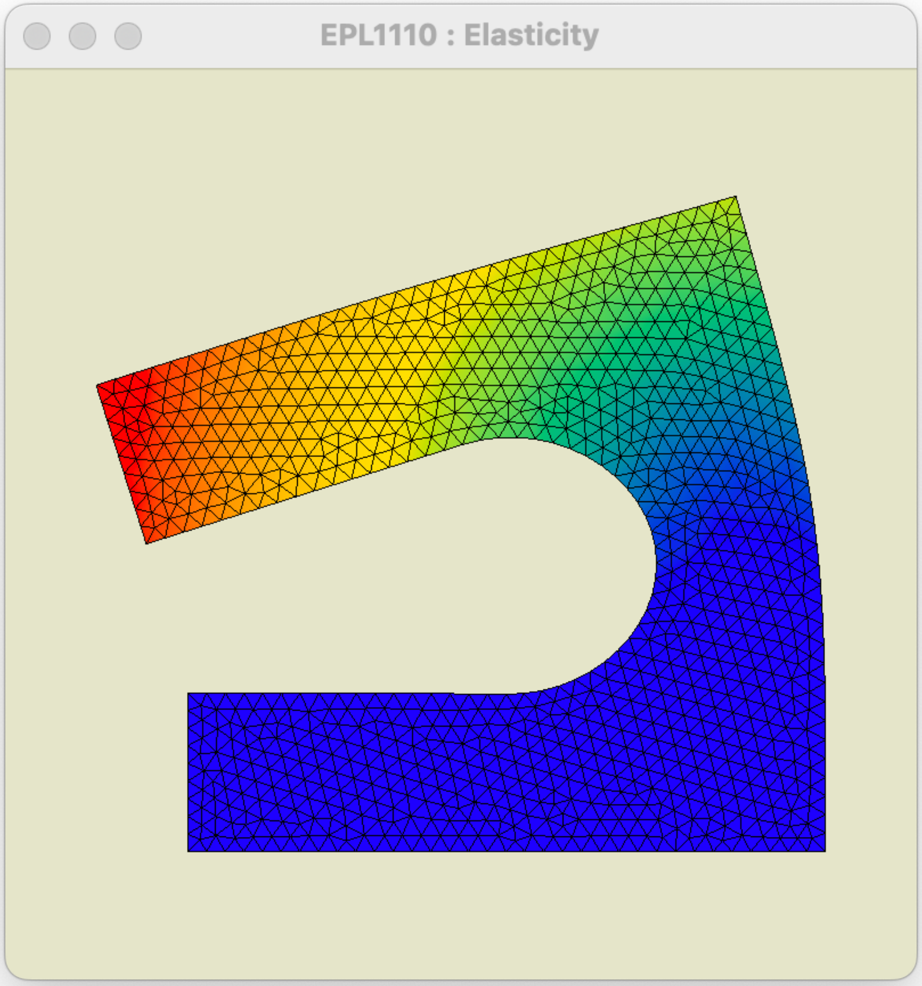
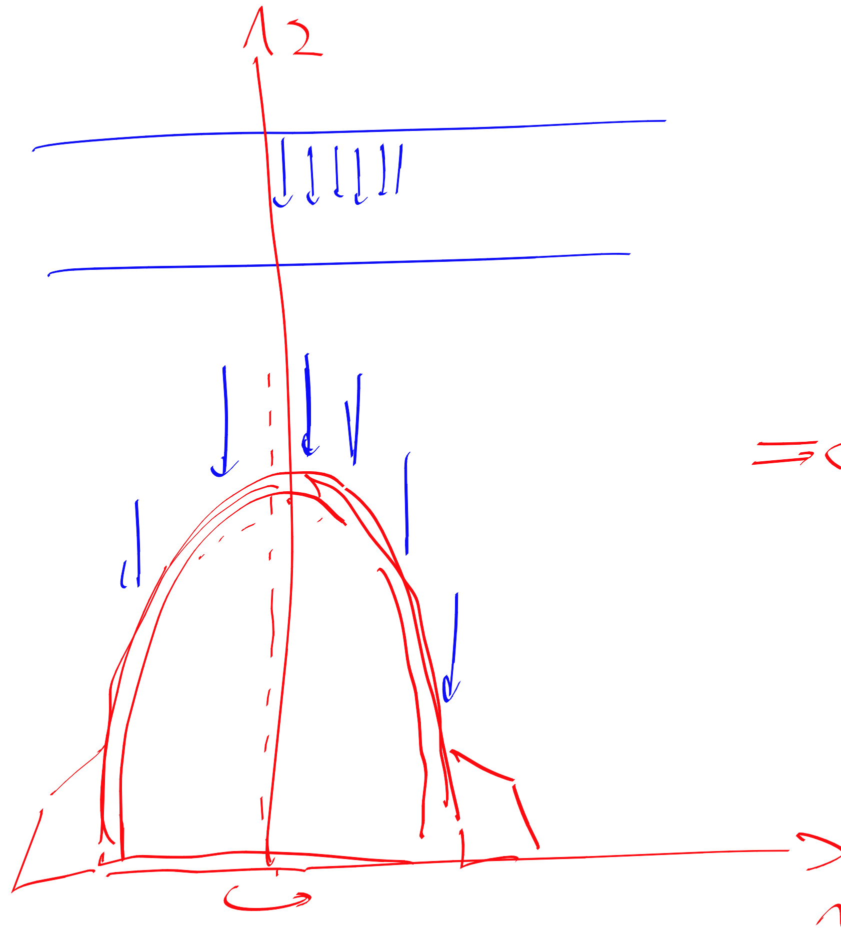
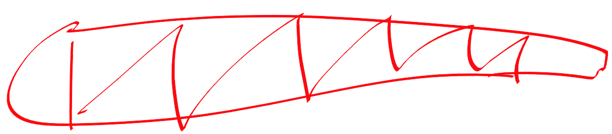
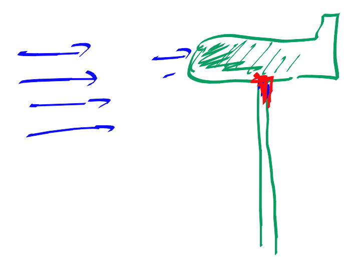
Projet 2023-24



Questions ?







Ecrire un code informatique efficace pour l'élasticité linéaire plane

Tensions planes et déformations planes
Triangles linéaires ou quadratiques
Quads bilinéaires ou biquadratiques
Problèmes axisymétriques
Conditions essentielles en xy et en normale/tangentielle
Conditions naturelles en xy et en normale/tangentielle
Obtenir les tensions dans le domaine

2 parties
dans le projet !

Définir un problème original !
Le résoudre avec votre code !
Analyser le résultat !

Projet 2023-24

mesh.txt

```
Number of nodes 335
     0 :  0.0000000e+00  1.0000000e+00
     1 :  0.0000000e+00  0.0000000e+00
     2 :  1.0000000e+00  1.0000000e+00
     3 :  1.0000000e+00  7.5000000e-01
     4 :  5.0000000e-01  7.5000000e-01
     5 :  5.0000000e-01  2.5000000e-01
     6 :  1.0000000e+00  2.5000000e-01
     7 :  1.0000000e+00  0.0000000e+00
     8 :  0.0000000e+00  9.5000000e-01
     9 :  0.0000000e+00  9.0000000e-01
```

problem.txt

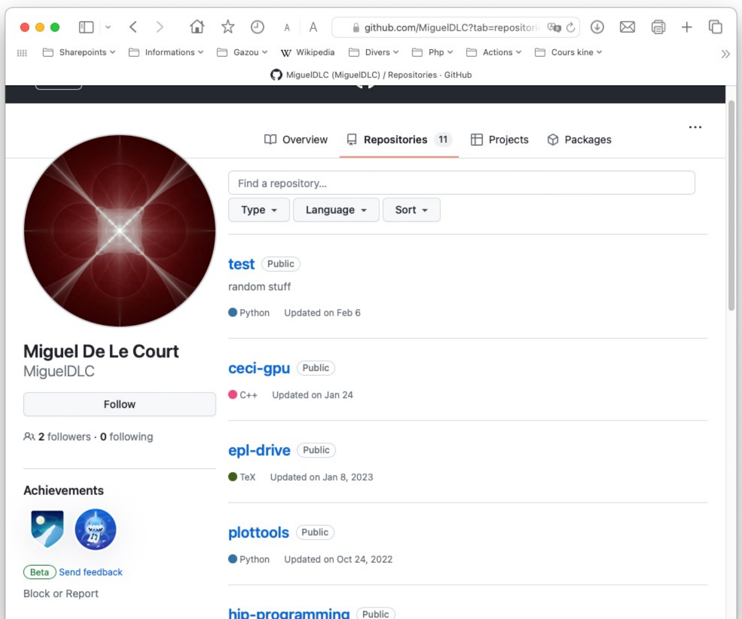
```
Type of problem : planar strains
Young modulus   :  2.1100000e+11
Poisson ratio   :  3.0000000e-01
Mass density    :  7.8500000e+03
Gravity         :  9.8100000e+00
```

myFem

result.txt

```
Elastic deformation 335
     0 :  0.0002330e-03  1.0000000e+00
     1 :  0.0000000e+00  0.0000000e+00
```

Tous les trucs utiles !
La semaine prochaine avec Miguel

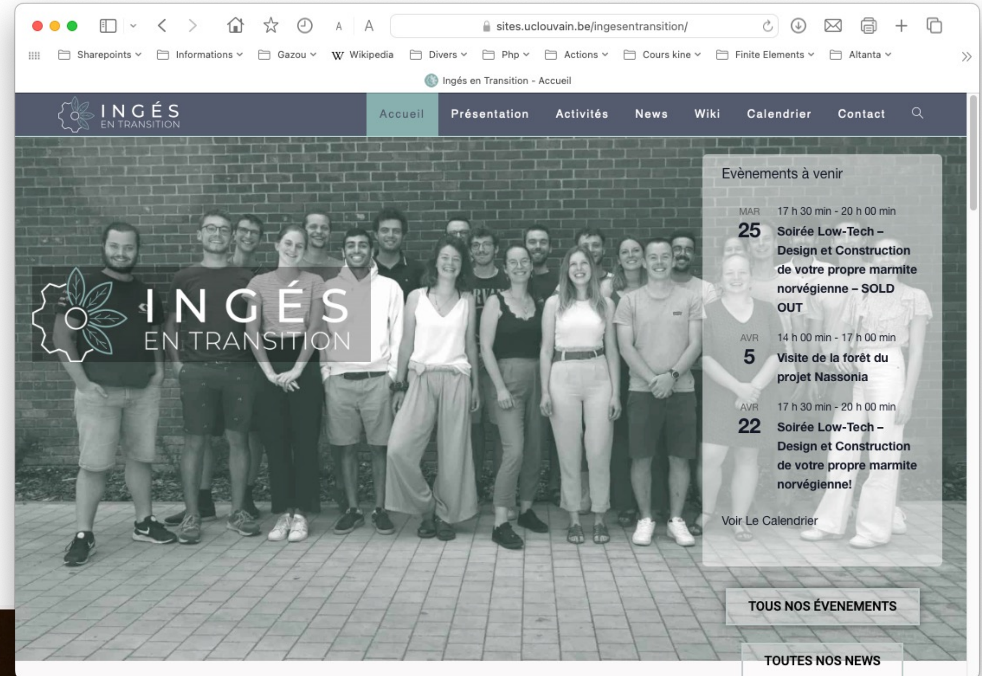


Si !

```
for (iElem = 0; iElem < theMesh->nElem; iElem++) {
    for (j=0; j < nLocal; j++) {
        map[j]  = theMesh->elem[iElem*nLocal+j];
        mapX[j] = 2*map[j];
        mapY[j] = 2*map[j] + 1;
        x[j]    = theNodes->X[map[j]];
        y[j]    = theNodes->Y[map[j]];}

    for (iInteg=0; iInteg < theRule->n; iInteg++) {
        double xsi    = theRule->xsi[iInteg];
        double eta    = theRule->eta[iInteg];
        double weight = theRule->weight[iInteg];
        femDiscretePhi2(theSpace,xsi,eta,phi);
        femDiscreteDphi2(theSpace,xsi,eta,dphidxsi,dphideta);

        double dxdxsi = 0.0;
        double dxdeta = 0.0;
        double dydxsi = 0.0;
        double dydeta = 0.0;
        for (i = 0; i < theSpace->n; i++) {
            dxdxsi += x[i]*dphidxsi[i];
            dxdeta += x[i]*dphideta[i];
            dydxsi += y[i]*dphidxsi[i];
            dydeta += y[i]*dphideta[i]; }
        double jac = fabs(dxdxsi * dydeta - dxdeta * dydxsi);

        for (i = 0; i < theSpace->n; i++) {
            dphidx[i] = (dphidxsi[i] * dydeta - dphideta[i] * dydxsi) / jac;
            dphidy[i] = (dphideta[i] * dxdxsi - dphidxsi[i] * dxdeta) / jac; }
        for (i = 0; i < theSpace->n; i++) {
            for(j = 0; j < theSpace->n; j++) {
                A[mapX[i]][mapX[j]] += (dphidx[i] * a * dphidx[j] +
                                        dphidy[i] * c * dphidy[j]) * jac * weight;
                A[mapX[i]][mapY[j]] += (dphidx[i] * b * dphidy[j] +
                                        dphidy[i] * c * dphidx[j]) * jac * weight;
                A[mapY[i]][mapX[j]] += (dphidy[i] * b * dphidx[j] +
                                        dphidx[i] * c * dphidy[j]) * jac * weight;
                A[mapY[i]][mapY[j]] += (dphidy[i] * a * dphidy[j] +
                                        dphidx[i] * c * dphidx[j]) * jac * weight; }}
         for (i = 0; i < theSpace->n; i++) {
            B[mapY[i]] -= phi[i] * g * rho * jac * weight; }}}

int *theConstrainedNodes = theProblem->constrainedNodes;
for (int i=0; i < theSystem->size; i++) {
    if (theConstrainedNodes[i] != -1) {
        double value = theProblem->conditions[theConstrainedNodes[i]]->value;
        femFullSystemConstrain(theSystem,i,value); }}

return femFullSystemEliminate(theSystem);
```

Les sponsors du projet !
Ingés en transition

24 avril 2024
Salle Nyquist



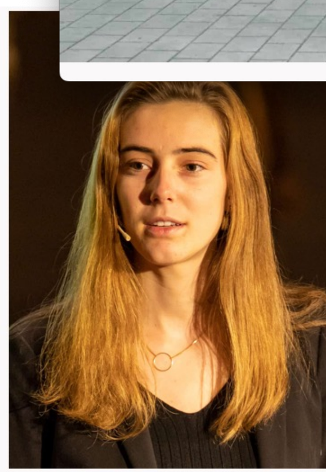
MER
24

24 avril, 18 h 30 min - 20 h 30 min

Table Ronde: Marcher sur du béton pour faire pousser de l'herbe ?

Salle Nyquist (A164) du bâtiment Maxwell Pl. du Levant 3,, Ottignies-Louvain-la-Neuve, Belgique

L'activisme comme levier de la transition écologique ? Des marches pour le climat à code rouge, beaucoup sont ceux qui participent de près ou de loin, de façon radicale ou modérée, à l'activisme écologique. Mais est-ce que tous ces efforts valent-ils la peine ? Quelle est la place de l'activisme dans la transition ? Nous [...]



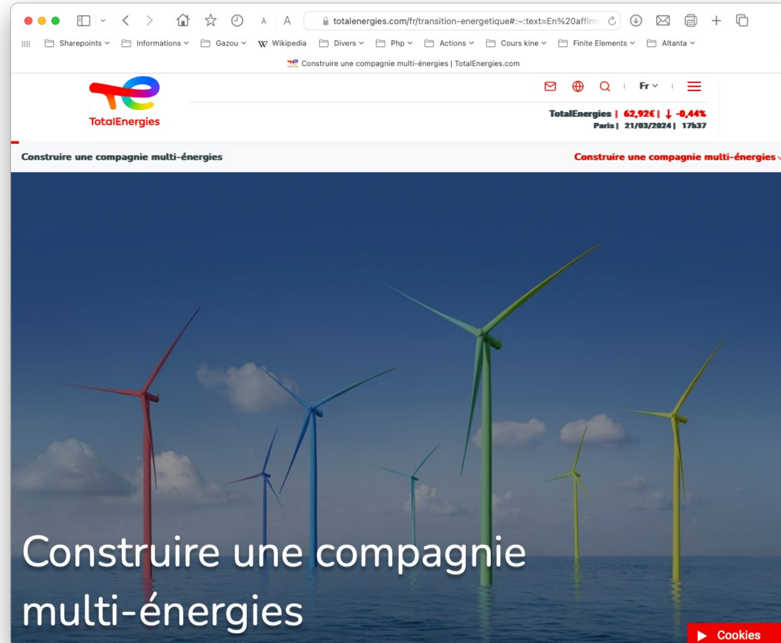
Non, non !
Assister à la Table Ronde ne donne aucun bonus pour le projet !
Bien essayer, Charles. Mais c'est loupé !
Oui : l'activisme dans la transition, c'est un choix politique respectable !

Les sponsors du projet !
La revue des ingénieurs

27 mars 2024
Aula Magna

Non, non !
Assister à la Revue ne donne aucun bonus pour le projet !
Bien essayer, Charles. Mais c'est loupé !
Oui : c'est un joli spectacle mais pas de Youpidou cette année encore :-(

Les sponsors du projet !
Total Energies

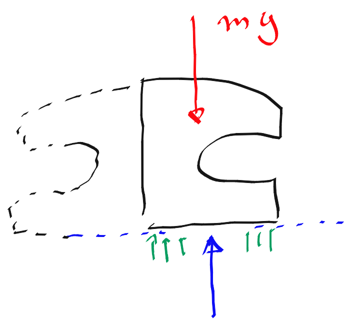


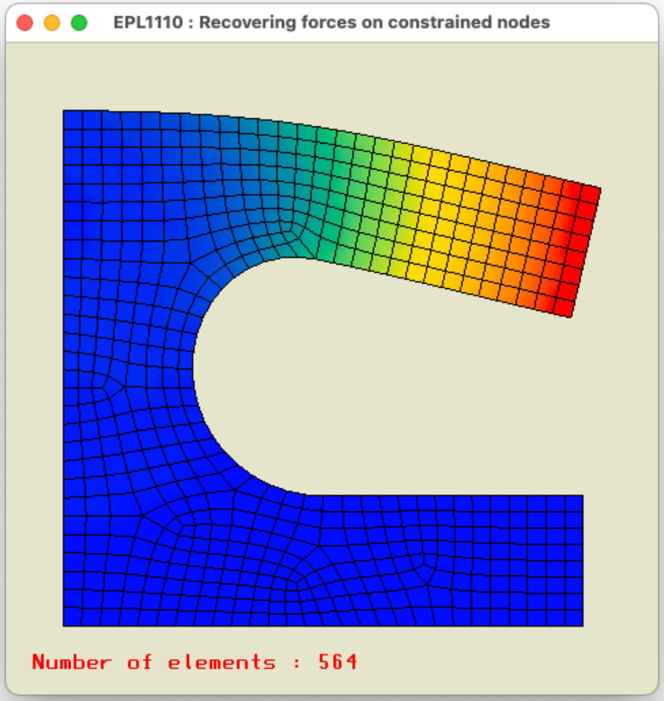
Victoire pour les organisations environnementales : les 20km de Bruxelles mettent fin à leur partenariat avec TotalEnergies

En affirmant son ambition d'être un acteur majeur de la transition énergétique et d'atteindre la neutralité carbone à horizon 2050, ensemble avec la société, TotalEnergies s'engage à faire évoluer en profondeur ses productions et ses ventes, tout en continuant à répondre aux besoins en énergie des populations en croissance.

Le prochain devoir !

Appliquer des conditions de Neumann !
Retrouver les forces de réactions !
Intégrer la densité de force de gravité !
Vérifier la seconde et la troisième loi de Newton !

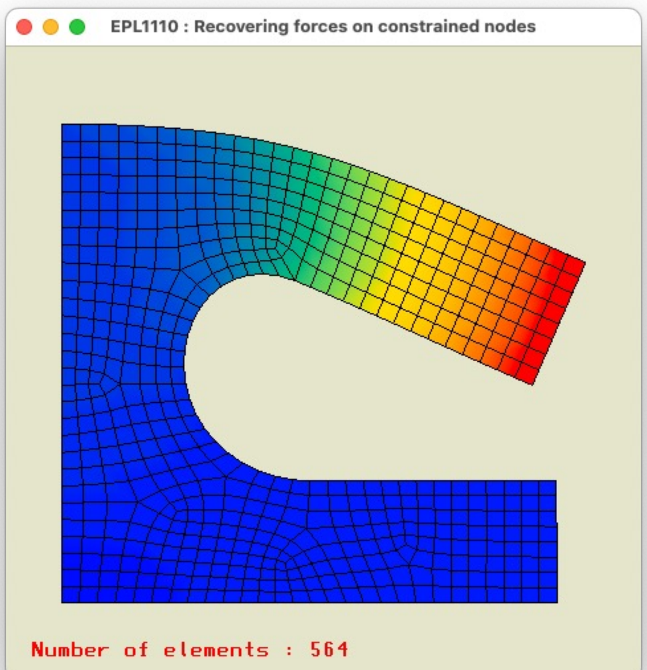


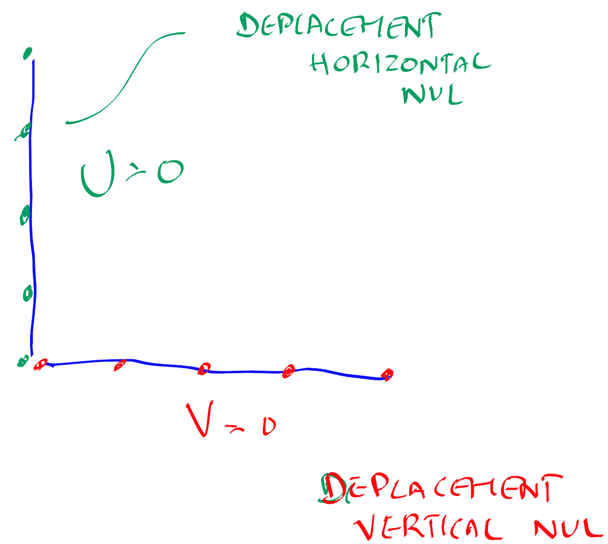



```
void femElasticityAssembleElements(femProblem *theProblem)
void femElasticityAssembleEdges(femProblem *theProblem)
void femElasticitySolve(femProblem *theProblem)
void femElasticityForces(femProblem *theProblem)
void femElasticityIntegrate(femProblem *theProblem, double (*f)(double,double))
```

Conditions aux limites

```
Linear elasticity problem
   Young modulus   E   =  2.1100000e+11 [N/m2]
   Poisson's ratio nu  =  3.0000000e-01 [-]
   Density         rho =  7.8500000e+03 [kg/m3]
   Gravity         g   =  9.8100000e+00 [m/s2]
   Planar strains formulation
   Boundary conditions :
              Symmetry : imposing  0.00e+00 as the horizontal displacement
                Bottom : imposing  0.00e+00 as the vertical displacement
```



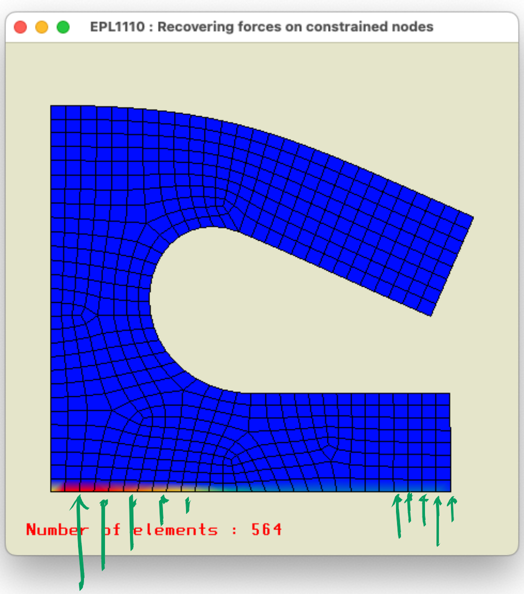


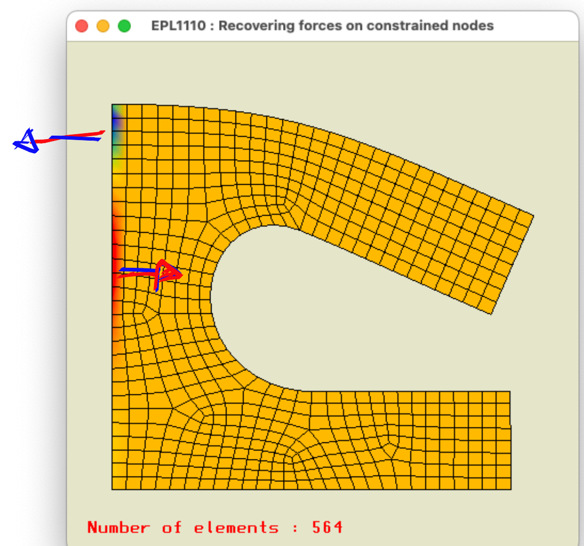
Forces appliquées sur les nœuds contraints

```
Linear elasticity problem
Boundary conditions :
              Symmetry : imposing  0.00e+00 as the horizontal displacement
                Bottom : imposing  0.00e+00 as the vertical displacement
                   Top : imposing -1.00e+04 as the vertical force density
```

```
 ==== Minimum displacement          :  0.0000000e+00 [m]
 ==== Maximum displacement          :  1.6227532e-06 [m]
 ==== Global horizontal force       :  2.0317257e-10 [N]
 ==== Global vertical force         :  1.0000000e+04 [N]
 ==== Weight                        :  0.0000000e+00 [N]
```

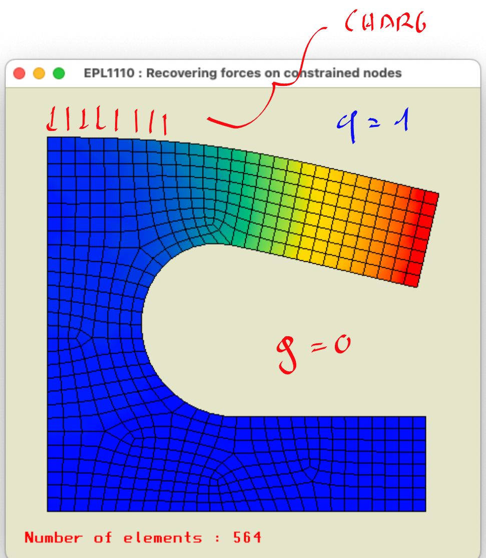
RESIDUS
EQUILIBRE EN Y

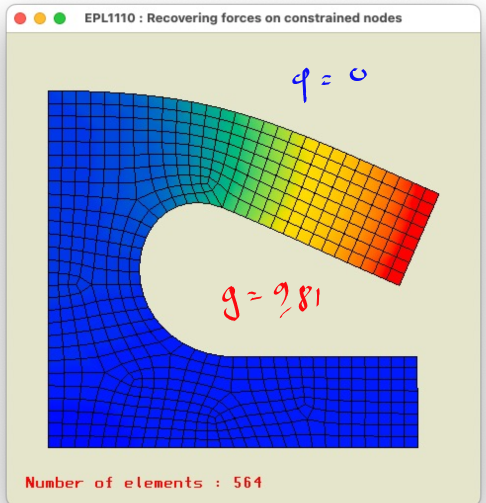


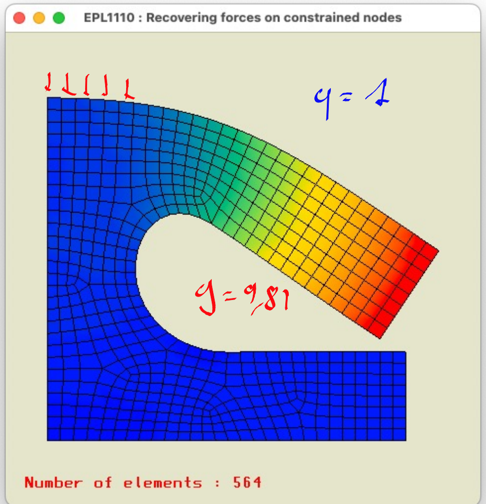


RESIDUS
EQUILIBRE EN X

Modifions le problème







Déformation sous la charge

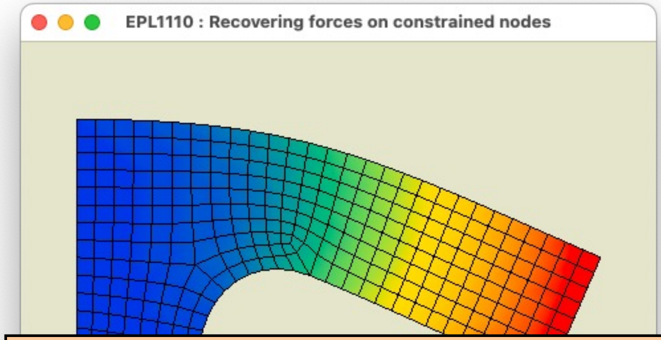
Déformation sous son poids propre

Déformation sous la charge et son poids propre

```
double E   = 211.e9;
    double nu  = 0.3;
    double rho = 7.85e3;
    double g   = 0.00;
    femProblem* theProblem = femElasticityCreate(theGeometry,E,nu,rho,g,PLANAR_STRAIN);
    femElasticityAddBoundaryCondition(theProblem,"Symmetry",DIRICHLET_X,0.0);
    femElasticityAddBoundaryCondition(theProblem,"Bottom",DIRICHLET_Y,0.0);
    femElasticityAddBoundaryCondition(theProblem,"Top",NEUMANN_Y,-1e4);
    femElasticityPrint(theProblem);
```

Modifions le problème

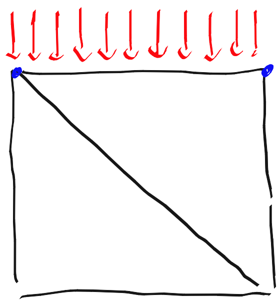
```
Linear elasticity problem
   Young modulus   E   =  2.1100000e+11 [N/m2]
   Poisson's ratio nu  =  3.0000000e-01 [-]
   Density         rho =  7.8500000e+03 [kg/m3]
   Gravity         g   =  9.8100000e+00 [m/s2]
   Planar strains formulation
   Boundary conditions :
              Symmetry : imposing  0.00e+00 as the horizontal displacement
                Bottom : imposing  0.00e+00 as the vertical displacement
```




```
double E   = 211.e9;
    double nu  = 0.3;
    double rho = 7.85e3;
    double g   = 0.00;
    femProblem* theProblem = femElasticityCreate(theGeometry,E,nu,rho,g,PLANAR_STRAIN);
    femElasticityAddBoundaryCondition(theProblem,"Symmetry",DIRICHLET_X,0.0);
    femElasticityAddBoundaryCondition(theProblem,"Bottom",DIRICHLET_Y,0.0);
    femElasticityAddBoundaryCondition(theProblem,"Top",NEUMANN_Y,-1e4);
    femElasticityPrint(theProblem);
```

Conditions de Neumann !

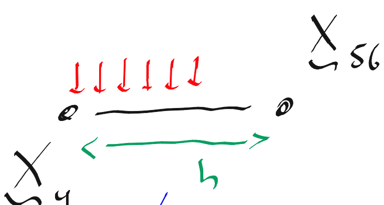
$$\underset{\sim}{g} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} -10^4 \\ 0 \end{bmatrix}$$

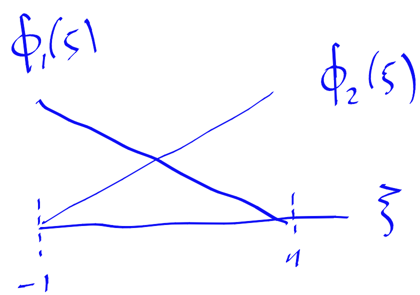


$$\sum_j \langle \underset{\sim}{\nabla} \tau_i \cdot \underset{\sim}{\nabla} \tau_j \rangle U_j = \langle\langle g \tau_i \rangle\rangle_N$$

$$\left\{ \begin{array}{l} \langle\langle g_x \tau_i \rangle\rangle_N \\ \langle\langle g_y \tau_i \rangle\rangle_N \end{array} \right.$$

ASSOCIÉES AUX ÉQUATIONS DES DÉPLACEMENTS EN x y





$$\phi_1 = (1-\xi)/2$$
$$\phi_2 = (1+\xi)/2$$

$$J = \frac{h}{2}$$

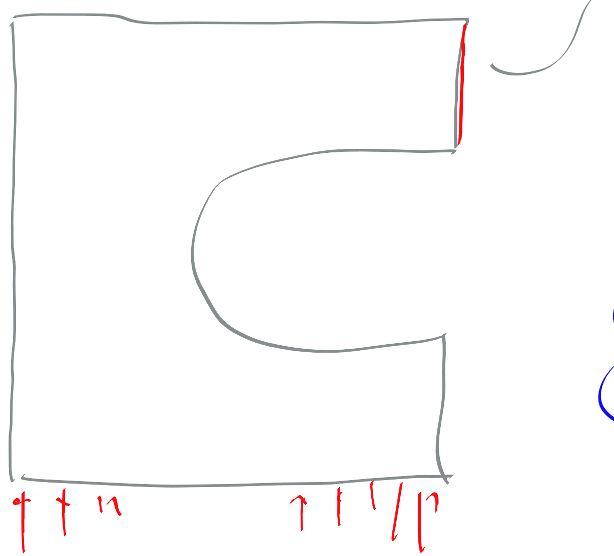
$$h = \sqrt{(X_4 - X_{56})^2 + (Y_4 - Y_{56})^2}$$

Les conditions aux limites d'un problème d'élasticité linéaire.

$$\nabla^2 u + f = 0$$

$$u = \bar{u} \quad \text{sur } \Gamma_D$$

$$n \cdot \nabla u = \bar{g} \quad \text{sur } \Gamma_N$$



ESSENTIELLE
DIRICHLET

IL FAUT

SOIT IMPOSER LE DEPLACEMENT

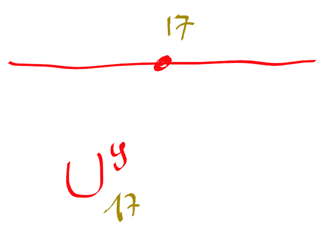
SOIT IMPOSER UNE DENSITE DE FORCE

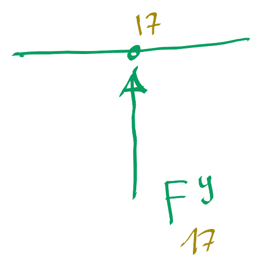
NATURELLE
NEUMANN

On peut appliquer une condition de Dirichlet ou…
une condition de Neumann équivalente.

$$U^y_{17} = V_{17} = 0$$

CONDITION
ESSENTIELLE





« $g_y \tau_{17}$ »

CONDITION
NATURELLE
EQUIVALENTE

Retrouver les forces de contact !

$$U^y_{17} = V_{17} = 0$$

CONDITION ESSENTIELLE

U^y_{17}

$38 = \underbrace{17}\times 2 + \underbrace{1}\ldots$

$$\begin{bmatrix} \\ \ldots\ldots\langle\ \rangle\ldots \\ \\ \end{bmatrix} \begin{bmatrix} \\ V_{17} \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \langle\langle\ \rangle\rangle_N \\ \\ \end{bmatrix}$$

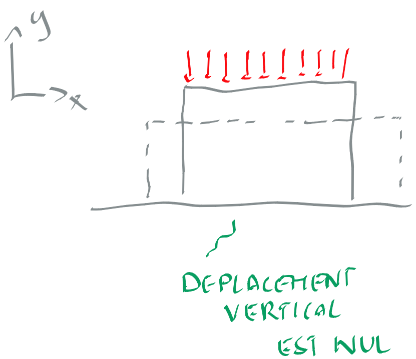
$$\underbrace{\langle\langle\ g_y\ \tau_{17}\ \rangle\rangle}_{F^y_{17}}$$

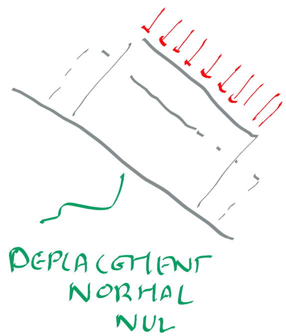
Exprimer les conditions aux limites en termes de normale et tangente

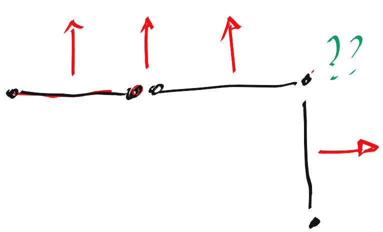
$$\underset{\sim}{n} \cdot \underset{\sim}{t} = 0$$

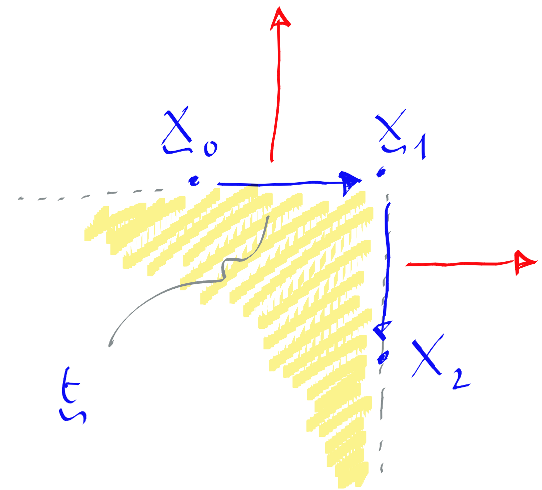
$$\underset{\sim}{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} -t_y \\ t_x \end{bmatrix}$$

$$\underset{\sim}{u} = \begin{bmatrix} u \\ v \end{bmatrix}$$



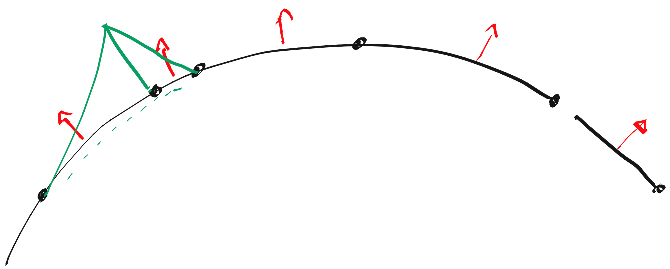




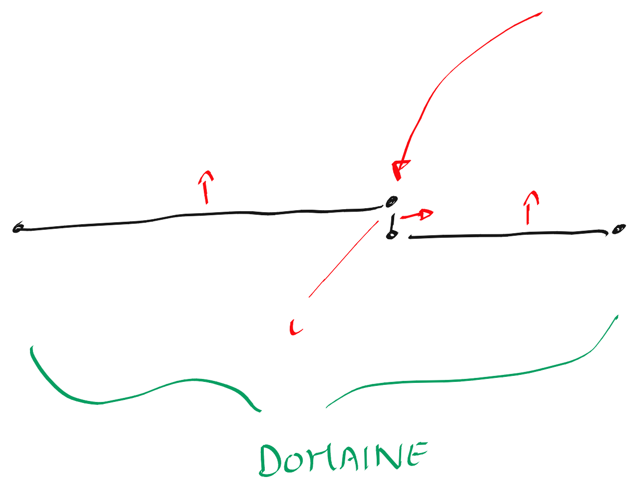


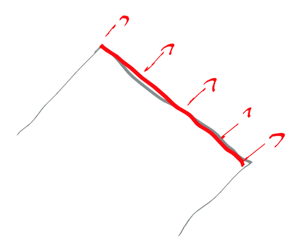
$$\underset{\sim}{t} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \underset{\sim}{X}_1 - \underset{\sim}{X}_0$$

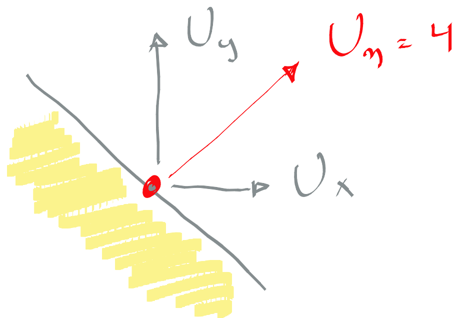
$$\underset{\sim}{n} = \sqrt{n_x^2 + n_y^2}$$

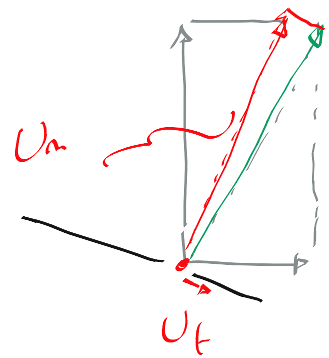


$$\underset{\sim}{N}_i = \langle\!\langle \phi_i \, \underset{\sim}{n} \rangle\!\rangle$$









$$U_n = \alpha U_x + \beta U_y$$
$$U_t = \gamma U_x + \delta U_y$$

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} U_x \\ U_y \end{bmatrix} = \begin{bmatrix} \times & \times \\ \cdots & \cdots \\ \times & \times \end{bmatrix}$$

$$\begin{bmatrix} U_n \\ U_t \end{bmatrix}$$

$$\boxed{n_x U_x + n_y U_y = 4}$$

CECI EST LA CONDITION ESSENTIELLE QUE JE VEUX IMPOSER

$$J(v) = \frac{1}{2} \langle \nabla v \cdot \nabla v \rangle - \langle f v \rangle$$

$$v \simeq v^h = \sum_{i=1}^{n} U_i \tau_i$$

$$J(v^h) = \frac{1}{2} \left\langle \left(\sum_i \nabla \tau_i U_i\right) \left(\sum_j \nabla \tau_j U_j\right) \right\rangle - \left\langle f \left(\sum U_i \tau_i\right) \right\rangle$$

$$A_{ij} = \langle \nabla \tau_i \cdot \nabla \tau_j \rangle$$

$$B_i = \langle f \tau_i \rangle$$

$$= \frac{1}{2} \sum_i \sum_j A_{ij} U_j - \sum_i B_i U_i$$

$$\boxed{\sum_j A_{ij} U_j = B_i}$$

$$\boxed{U_n = a + b U_1}$$

$$\boxed{U_n = a + b\,U_1}$$

$$u^h = \sum_{i=1}^{n-1} \tau_i U_i + \tau_n (a + b\,U_1)$$

$$J(u^h) = \frac{1}{2} \left\langle \left(\sum_i^n \nabla \tau_i U_i\right)\left(\sum_j^n \nabla \tau_j U_j\right) \right\rangle - \left\langle f \left(\sum^n U_i \tau_i\right) \right\rangle$$

$$\boxed{\begin{aligned} J(u^h) = {} & \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} A_{ij} U_i U_j - \sum_{i=1}^{n-1} B_i U_i \\ & + \sum_{i=1}^{n-1} A_{in} U_i (b + a\,U_1) \\ & + \tfrac{1}{2} A_{nn} (b + a\,U_1)(b + a\,U_1) - B_n (b - a\,U_1) \end{aligned}}$$

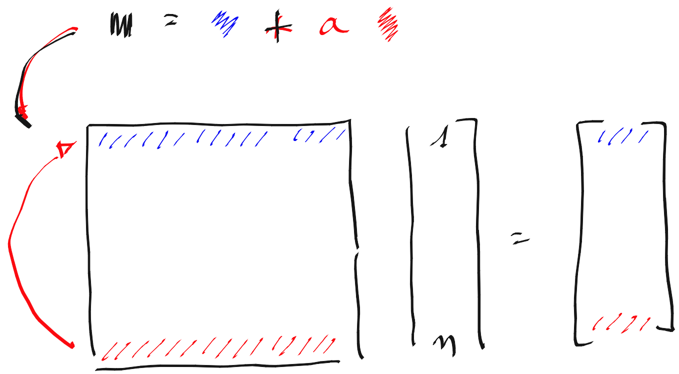
$$0 = \frac{\partial J}{\partial U_i} \qquad i = 1, \dots, n+1$$

$$0 = a\left(\sum A_{nj} U_j + A_{nn}(a + b\,U_1) - B_n\right) + \sum A_{1j} U_j - B_1$$

EQUATION POUR 1

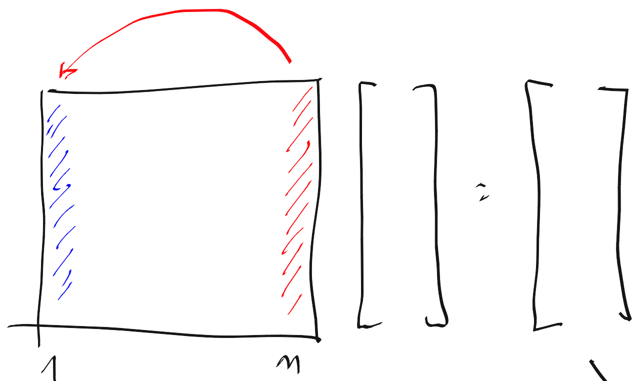
$$\sum_{i=1}^{n-1} A_{ij} U_j - B_i + A_{in} (b + a\,U_1) = 0$$

EQUATIONS 2 ... n-1

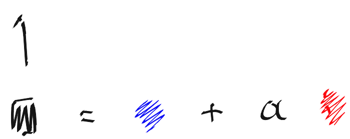


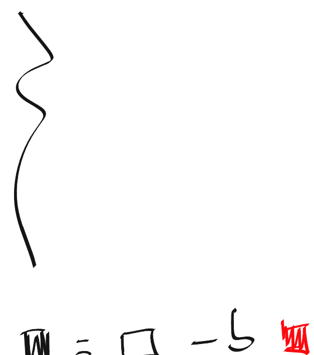
$U_n = b + a\,U_1$

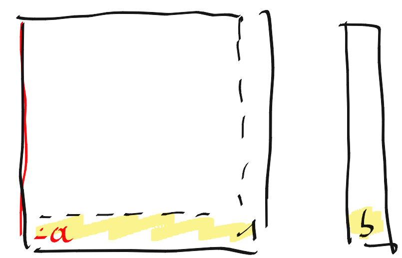
$L1 \leftarrow L_n\, a + L1$



$C1 \leftarrow C_n\, a + C1$



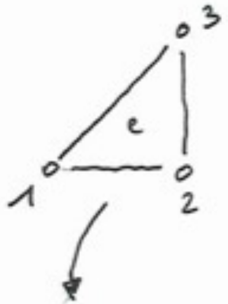




Et l’axisymétrique ?

COMPUTING A LOCAL ELASTICITY MATRIX

$$\left[\begin{array}{c|c} A\langle \tau_{i,x}\,\tau_{j,x}\rangle + C\langle \tau_{i,y}\,\tau_{j,y}\rangle & B\langle \tau_{i,x}\,\tau_{j,y}\rangle + C\langle \tau_{i,y}\,\tau_{j,x}\rangle \\ \hline B\langle \tau_{i,y}\,\tau_{j,x}\rangle + C\langle \tau_{i,x}\,\tau_{j,y}\rangle & A\langle \tau_{i,y}\,\tau_{j,y}\rangle + C\langle \tau_{i,x}\,\tau_{j,x}\rangle \end{array}\right]$$



	$\tau_{i,x}$	$\tau_{i,y}$
1	-1	0
2	1	-1
3	0	1

$$\boxed{\underline{\underline{A}}^{e}_{11} \quad \begin{array}{l} A_{xx11} = A \\ A_{yy11} = C \\ A_{xy11} = 0 \\ A_{yx11} = 0 \end{array}}$$

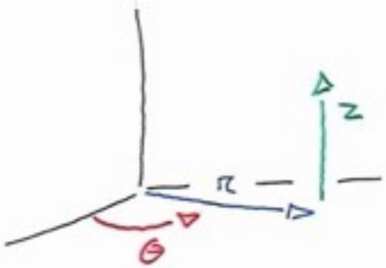
$$\underline{\underline{A}}^{e}_{y} = \left[\begin{array}{cc|cc|cc} \boxed{\begin{array}{cc} A & 0 \\ 0 & C \end{array}} & & -A & B & 0 & -B \\ & & C & -C & -C & 0 \\ \hline & & A+C & -B-C & -C & B \\ & & -B-C & A+C & C & -A \\ \hline & & & & C & 0 \\ & & & & 0 & A \end{array}\right] \tfrac{1}{2}$$

AXISYMMETRIC PROBLEMS

AXISYMMETRY!

$u_\theta = 0$

$\frac{\partial}{\partial \theta} = 0$



$$\underset{\sim}{u} = \begin{bmatrix} u_r \\ u_\theta \\ u_z \end{bmatrix} = \begin{bmatrix} u(r,z) \\ 0 \\ v(r,z) \end{bmatrix}$$

$$\underset{\approx}{\varepsilon} = \begin{bmatrix} u_{r,r} & 0 & (u_{r,z} + u_{z,r})/2 \\ & u_r/r & 0 \\ & & u_{z,z} \end{bmatrix}$$

A, B, C OF PLANAR DEFORMATIONS !

$$\underset{\approx}{\sigma} = \begin{bmatrix} A\,\varepsilon_{rr} + B(\varepsilon_{\theta\theta} + \varepsilon_{zz}) & 0 & 2C\,\varepsilon_{rz} \\ & A\,\varepsilon_{\theta\theta} + B(\varepsilon_{rr} + \varepsilon_{zz}) & 0 \\ & & A\,\varepsilon_{zz} + B(\varepsilon_{rr} + \varepsilon_{\theta\theta}) \end{bmatrix}$$

$$\underline{\underline{\varepsilon}}\left(\begin{bmatrix}\tau_i & 0 & 0\end{bmatrix}\right) = \begin{bmatrix} \tau_{i,r} & 0 & \tau_{i,z}/2 \\ & \tau_i/r & 0 \\ & & 0 \end{bmatrix}$$

$$\underline{\underline{\varepsilon}}\left(\begin{bmatrix}0 & 0 & \tau_i\end{bmatrix}\right) = \begin{bmatrix} 0 & 0 & \tau_{i,r}/2 \\ & 0 & 0 \\ & & \tau_{i,z} \end{bmatrix}$$

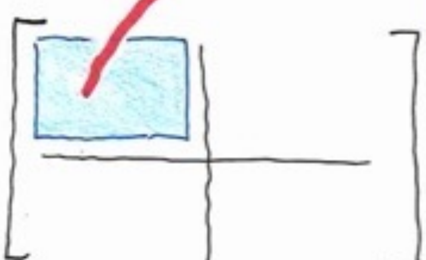
$$\underbrace{2\pi \left[\begin{array}{c|c} \left\langle r\, \underline{\underline{\varepsilon}}\begin{pmatrix}\tau_i\\0\\0\end{pmatrix} : \underline{\underline{\sigma}}\begin{pmatrix}\tau_j\\0\\0\end{pmatrix}\right\rangle & \left\langle r\, \underline{\underline{\varepsilon}}\begin{pmatrix}\tau_i\\0\\0\end{pmatrix} : \underline{\underline{\sigma}}\begin{pmatrix}0\\0\\\tau_j\end{pmatrix}\right\rangle \\ \hline \left\langle r\, \underline{\underline{\varepsilon}}\begin{pmatrix}0\\0\\\tau_i\end{pmatrix} : \underline{\underline{\sigma}}\begin{pmatrix}\tau_j\\0\\0\end{pmatrix}\right\rangle & \left\langle r\, \underline{\underline{\varepsilon}}\begin{pmatrix}0\\0\\\tau_i\end{pmatrix} : \underline{\underline{\sigma}}\begin{pmatrix}0\\0\\\tau_j\end{pmatrix}\right\rangle \end{array}\right]}_{\underline{\underline{A}}_{ij}}$$

$\langle\rangle$

$\langle 2\pi r \quad \rangle = \langle \quad \rangle_{Ax_i}$ ∇

$$\underline{\underline{\varepsilon}}\begin{pmatrix}\tau_i\\0\\0\end{pmatrix} = \begin{bmatrix} \tau_{i,r} & 0 & \tau_{i,z}/2 \\ & \tau_i/r & 0 \\ & & 0\end{bmatrix}$$

$$\underline{\underline{\sigma}}\begin{pmatrix}\tau_j\\0\\0\end{pmatrix} = \begin{bmatrix} A\,\tau_{j,r} + B\,\tau_j/r & 0 & C\,\tau_{j,z} \\ & A\,\tau_j/r + B\,\tau_{j,r} & 0 \\ & & B\,\tau_j/r + B\,\tau_{j,r}\end{bmatrix}$$



$$\boxed{\underline{\underline{\varepsilon}}\begin{pmatrix}\tau_i\\0\\0\end{pmatrix} : \underline{\underline{\sigma}}\begin{pmatrix}\tau_j\\0\\0\end{pmatrix}} = \left\langle \tau_{i,r}\left(A\,\tau_{j,r} + B\,\tau_j/r\right) r \right\rangle + \left\langle \tau_{i,z}\, C\,\tau_{j,z}\, r \right\rangle + \left\langle \frac{\tau_i}{r}\left(A\,\frac{\tau_j}{r} + B\,\tau_{j,r}\right) r \right\rangle$$

$$\underline{\underline{A}}_{ij}$$

IT IS ALMOST LIKE A 2D PROBLEM !

$$
\left[
\begin{array}{c|c}
\begin{aligned}
& A \langle \tau_{i,r}\, \tau_{j,r}\, r \rangle \\
& + C \langle \tau_{i,z}\, \tau_{j,z}\, r \rangle \\
& + B \langle \tau_{i,r}\, \tau_j \rangle \\
& + \langle \underbrace{\tau_i}_{\varepsilon_{\theta\theta}/r} \underbrace{\left(B\, \tau_{j,r} + A \frac{\tau_j}{r} \right)}_{\sigma_{\theta\theta}} \rangle
\end{aligned}
&
\begin{aligned}
& B \langle \tau_{i,r}\, \tau_{j,z}\, r \rangle \\
& + C \langle \tau_{i,z}\, \tau_{j,r}\, r \rangle \\
& + B \langle \tau_i\, \tau_{j,z} \rangle
\end{aligned}
\\ \hline
\begin{aligned}
& B \langle \tau_{i,z}\, \tau_{j,r}\, r \rangle \\
& + C \langle \tau_{i,r}\, \tau_{j,z}\, r \rangle \\
& + B \langle \tau_{i,z}\, \tau_j \rangle
\end{aligned}
&
\begin{aligned}
& A \langle \tau_{i,z}\, \tau_{j,z}\, r \rangle \\
& + C \langle \tau_{i,r}\, \tau_{j,r}\, r \rangle
\end{aligned}
\end{array}
\right]
$$