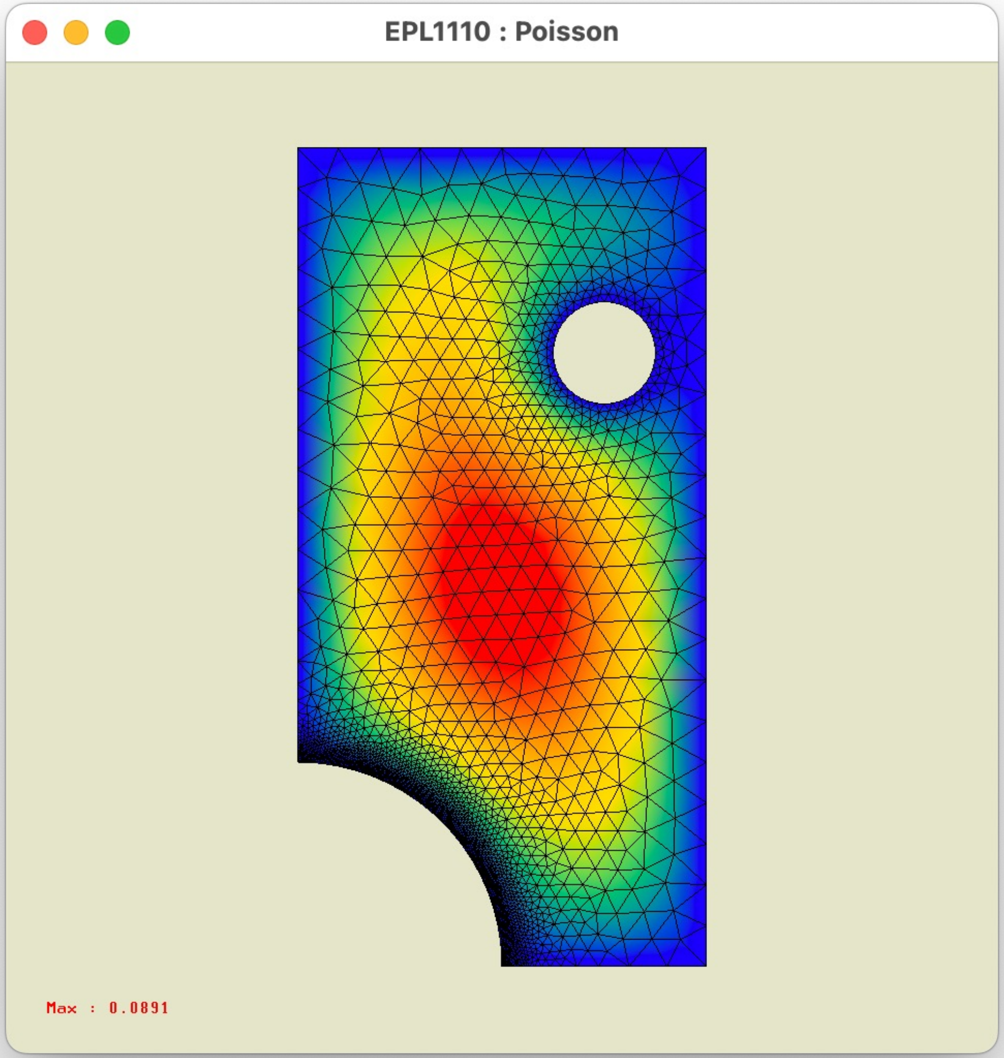
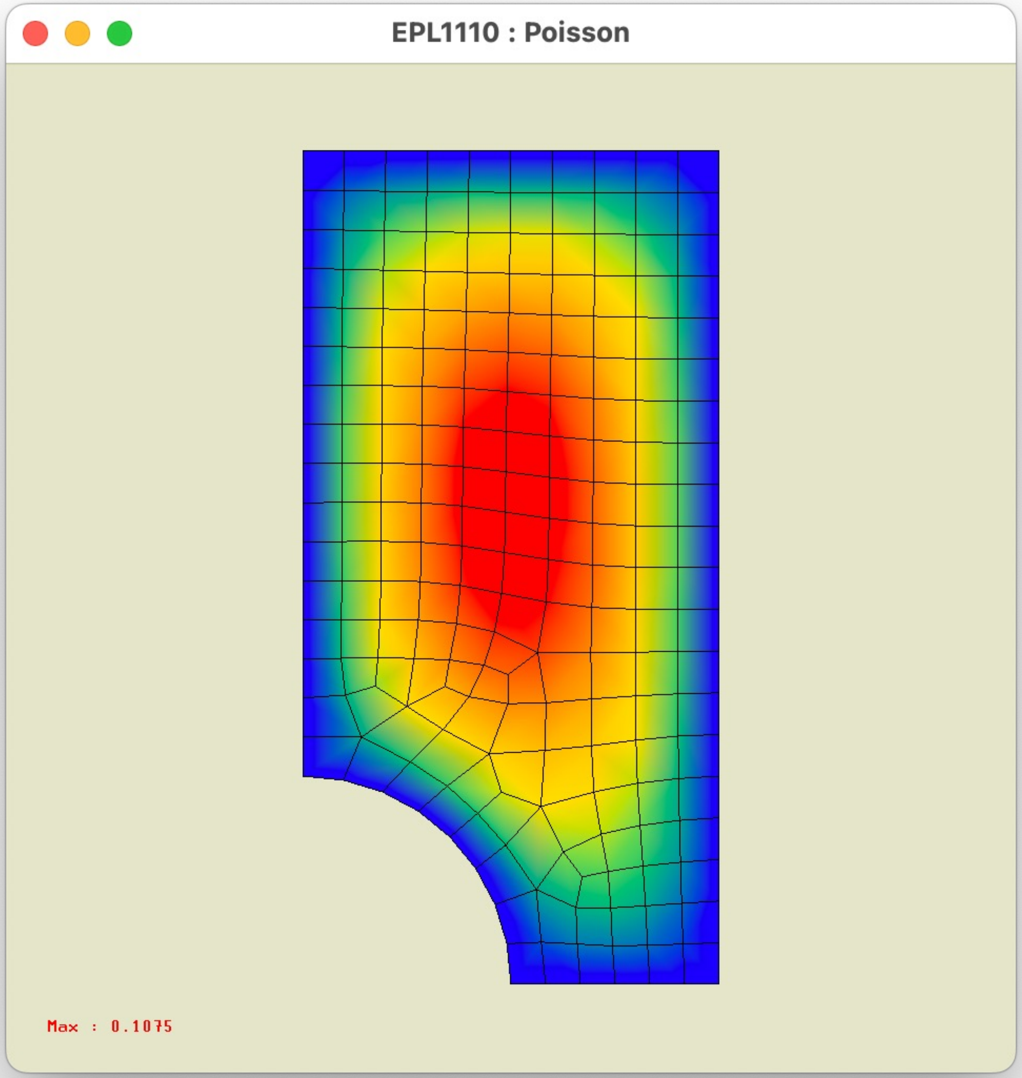


Nous allons faire notre premier
vrai programme d'éléments finisa
avec des triangles !

… et avec des quadrilatères.

Même si gmsh a un peu de difficultés à me créer de jolis maillages composés uniquement de quads !



Typical Elliptic BoundaryValue Problem

Trouver $u(\mathbf{x}) \in \mathcal{U}_s$ tel que

$$
\begin{aligned}
\boldsymbol{\nabla} \cdot (a \boldsymbol{\nabla} u) + f &= 0, & \forall \mathbf{x} \in \Omega, \\
\mathbf{n} \cdot (a \boldsymbol{\nabla} u) &= g, & \forall \mathbf{x} \in \Gamma_N, \\
u &= t, & \forall \mathbf{x} \in \Gamma_D,
\end{aligned}
$$

En général, la fonction n'est pas connue…
Mais, c'est la solution d'une équation aux dérivées partielles !

Commençons par une équation de Poisson

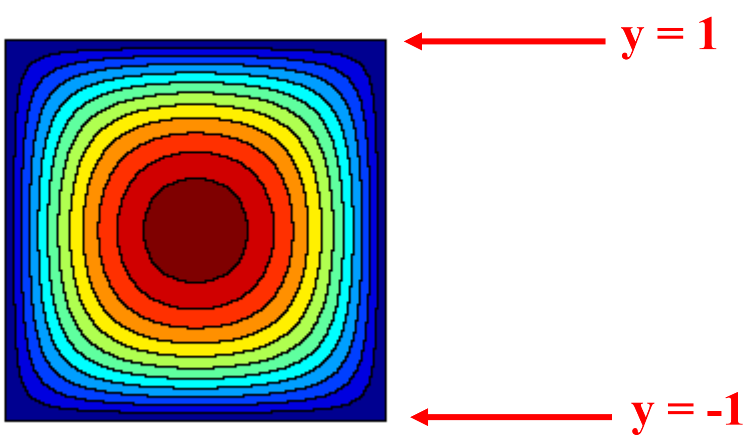
Conditions essentielles
Conditions de Dirichlet

Conditions naturelles
Conditions de Neumann

Un exemple tout simple et bien connu :-)

$$\nabla^2 u(x,y) + 1 = 0, \quad (x,y) \in \Omega,$$

$$u(x,y) = 0, \quad (x,y) \in \partial\Omega,$$



$$u(x,y) = \sum_{i,j\ impairs} C_{ij} \sin\left(\frac{i\pi(x+1)}{2}\right) \sin\left(\frac{j\pi(y+1)}{2}\right)$$

Comment trouver C_{ij} ?

$$\nabla^2 u(x,y) + 1 = 0,$$

$$\sum_{i,j \text{ impairs}} \frac{-\pi^2(i^2+j^2)}{4} C_{ij} \sin\left(\frac{i\pi(x+1)}{2}\right) \sin\left(\frac{j\pi(y+1)}{2}\right) + 1 = 0,$$

En vertu de l'orthogonalité des sinus,

$$\frac{\pi^2(i^2+j^2)}{4} C_{ij} - \underbrace{\int_\Omega \sin\left(\frac{i\pi(x+1)}{2}\right) \sin\left(\frac{j\pi(y+1)}{2}\right) d\Omega}_{16/(ij\pi^2)} = 0,$$

$$\boxed{u(x,y) = \sum_{i,j \ impairs} \frac{64}{\pi^4(i^2+j^2)ij} \sin\left(\frac{i\pi(x+1)}{2}\right) \sin\left(\frac{j\pi(y+1)}{2}\right)}$$

On a ici une solution analytique !
Mais, ce n'est pas le cas dans la plupart des vrais problèmes.
Il s'agit juste d'un exemple pour présenter notre méthode !

Construction du système linéaire discret

$u^h = \sum U_i \tau_i$

$$J(u^h) = \tfrac{1}{2}\int_\Omega (\nabla u^h)\cdot(\nabla u^h)\, d\Omega - \int_\Omega f\, u^h\, d\Omega,$$

$$J(u^h) = \tfrac{1}{2}\int_\Omega (\sum_{i=1}^{n} U_i \nabla\tau_i)\cdot(\sum_{j=1}^{n} U_j \nabla\tau_j)\, d\Omega - \int_\Omega f\, (\sum_{i=1}^{n} U_i\tau_i)\, d\Omega,$$

$$J(u^h) = \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} U_i\, U_j \boxed{\int_\Omega (\nabla\tau_i)\cdot(\nabla\tau_j)\, d\Omega} - \sum_{i=1}^{n} U_i \boxed{\int_\Omega f\, \tau_i\, d\Omega},$$

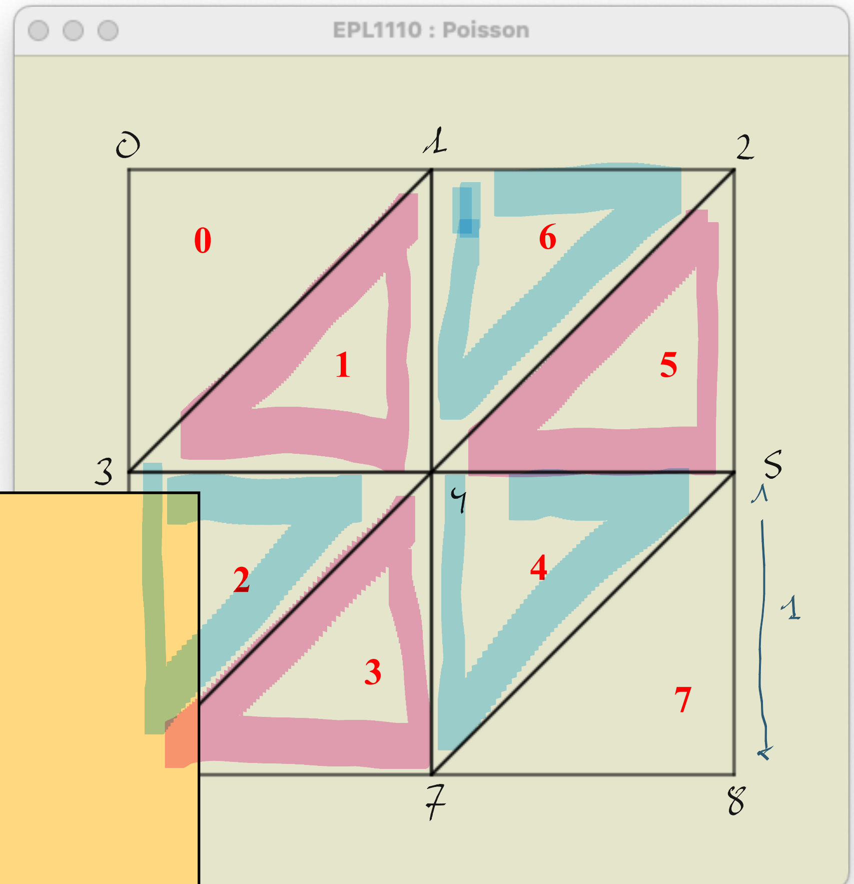
$$J(u^h) = \tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} U_i\, U_j \boxed{A_{ij}} - \sum_{i=1}^{n} U_i \boxed{B_i},$$

$$A_{ij} = \left\langle \frac{\partial\tau_i}{\partial x}\cdot\frac{\partial\tau_j}{\partial x} + \frac{\partial\tau_i}{\partial y}\cdot\frac{\partial\tau_j}{\partial y}\right\rangle$$

$$\oplus \left\langle \frac{\partial\phi_i^e}{\partial x}\frac{\partial\phi_j^e}{\partial x} + \frac{\partial\phi_i^e}{\partial y}\frac{\partial\phi_j^e}{\partial y}\right\rangle_e$$

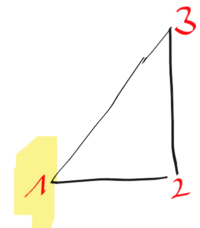
$$0 = \frac{\partial J(u^h)}{\partial U_i} = \sum_{j=1}^{n} \boxed{A_{ij}}\, U_j - \boxed{B_i}, \quad i = 1, n.$$

Faisons un petit exemple !

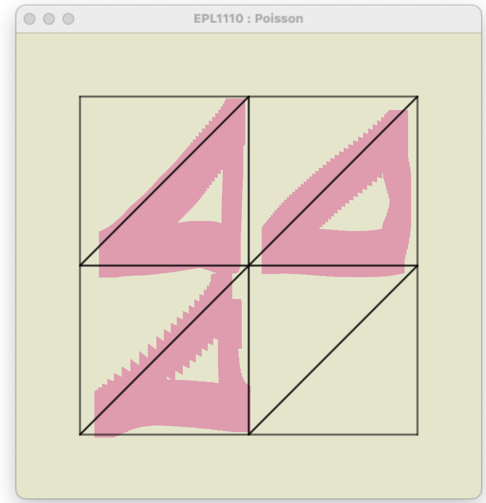



```
Number of nodes 9
      0 :  0.0000000e+00  2.0000000e+00
      1 :  1.0000000e+00  2.0000000e+00
      2 :  2.0000000e+00  2.0000000e+00
      3 :  0.0000000e+00  1.0000000e+00
      4 :  1.0000000e+00  1.0000000e+00
      5 :  2.0000000e+00  1.0000000e+00
      6 :  0.0000000e+00  0.0000000e+00
      7 :  1.0000000e+00  0.0000000e+00
      8 :  2.0000000e+00  0.0000000e+00
Number of triangles 8
      0 :        3        1        0
      1 :        3        4        1
      2 :        6        4        3
      3 :        6        7        4
      4 :        7        5        4
      5 :        4        5        2
      6 :        4        2        1
      7 :        7        8        5
```

Deux matrices locales



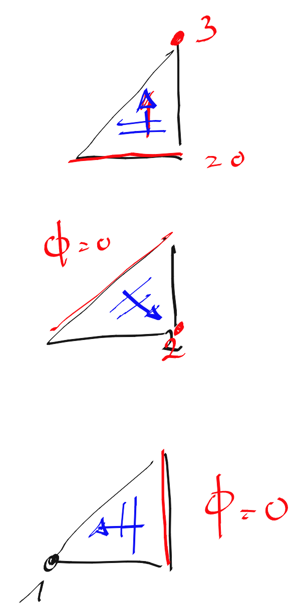
$$\phi_1 = cst - x$$
$$\phi_{1,x} = -1$$
$$\phi_{1,y} = 0$$



$$A^e_{ij} = \int_{\Omega_e} \frac{\partial \phi_i^e}{\partial x}\frac{\partial \phi_j^e}{\partial x} + \frac{\partial \phi_i^e}{\partial y}\frac{\partial \phi_j^e}{\partial y} \, dx\, dy$$

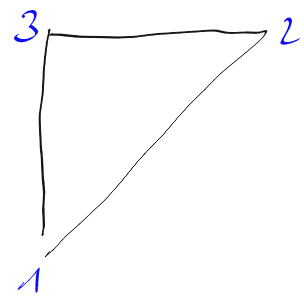
	$\phi_{i,x}$	$\phi_{i,y}$
$i = 1$	-1	0
2	1	-1
3	0	1

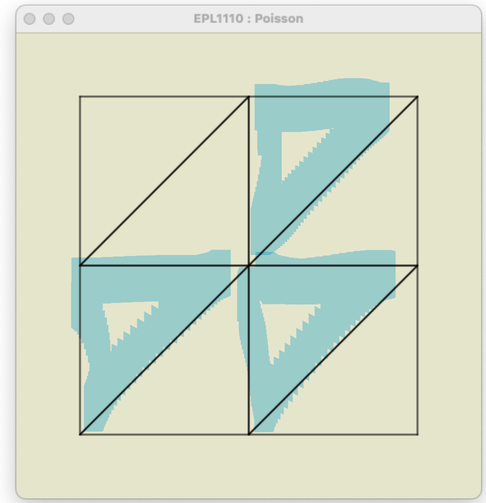
$$\underbrace{A^e_{ij} = \frac{1}{2}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}}_{A^e_{ij}}$$



$$\boxed{\sum \phi_i = 1 \qquad \sum \phi_{i,x} = 0 \qquad \sum \phi_{i,y} = 0}$$

Deux matrices locales



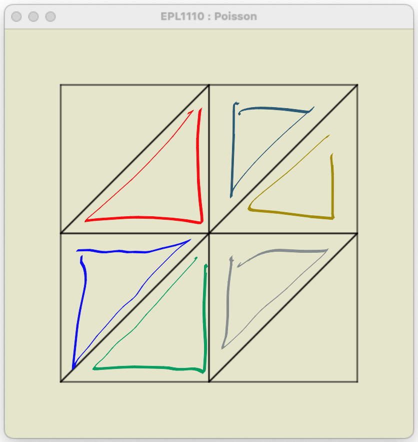


$$A^e_{ij} = \int_{\Omega_e} \frac{\partial \phi_i^e}{\partial x}\frac{\partial \phi_j^e}{\partial x} + \frac{\partial \phi_i^e}{\partial y}\frac{\partial \phi_j^e}{\partial y} \, dx\,dy$$

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

$$A^e_{ij} = \underbrace{\frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}}_{A^e_{ij}}$$

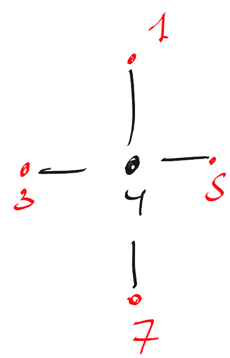
Assemblons les matrices



	1	2	3	4	5	6	7	
1	1 +2	-1		-1 -1				
2	-1	1 +1			-1			
3			1+2	-1 -1		-1		
$\frac{1}{2}$ 4	-1 -1		-1 -1	2+1+1 2+1+1	-1 -1		-1 -1	
5		-1		-1 -1	1+2			
6			-1			1+1	-1	
7				-1 -1		-1	2 1	

$$\langle \underbrace{\nabla \tau_i \cdot \nabla \tau_j}_{A_{ij}} \rangle = \bigoplus A^e_{ij}$$

$$\frac{1}{2}\left[8U_4 - 2U_1 - 2U_3 - 2U_5 - 2U_7 \right]$$

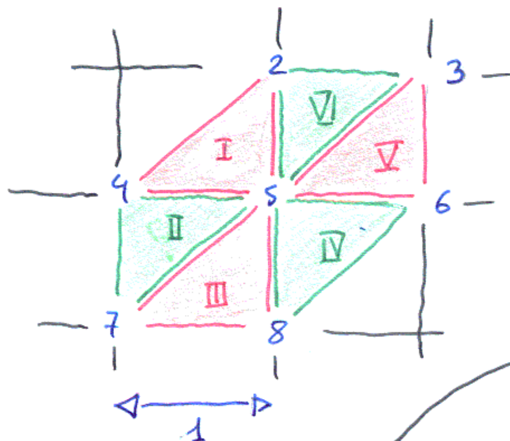


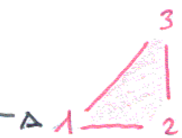
0	:	3	1	0
1	:	3	4	1
2	:	6	4	3
3	:	6	7	4
4	:	7	5	4
5	:	4	5	2
6	:	4	2	1
7	:	7	8	5

STIFFNESS MATRIX CALCULATION

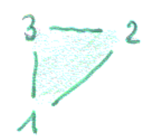
$$\underbrace{\left\langle \frac{\partial \phi_i^e}{\partial x}\frac{\partial \phi_j^e}{\partial x} + \frac{\partial \phi_i^e}{\partial y}\frac{\partial \phi_j^e}{\partial y} \right\rangle_{\Omega_e}}_{A_{ij}^e}$$

$$A_{ij} = \bigoplus A_{ij}^e$$





$$\begin{array}{c|cc} & \phi_{i,x} & \phi_{i,y} \\ \hline 1 & -1 & 0 \\ 2 & 1 & -1 \\ 3 & 0 & 1 \end{array} \longrightarrow \frac{1}{2}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$



$$\begin{array}{c|cc} & \phi_{i,x} & \phi_{i,y} \\ \hline 1 & 0 & -1 \\ 2 & 1 & 0 \\ 3 & -1 & 1 \end{array} \longrightarrow \overbrace{\frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}}^{A_{ij}^e}$$

CHECK !

$\sum \phi_i = 1$

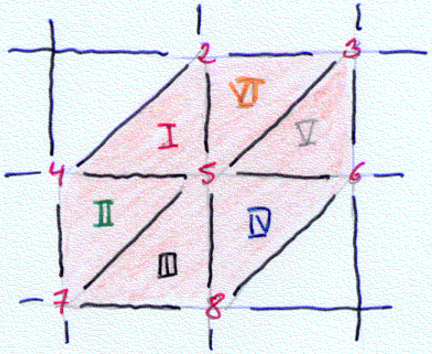
$\sum \phi_{i,x} = 0$

$\sum \phi_{i,y} = 0$

$\phi_1 = (1-x) + \text{cst}$

$\phi_{1,x} = -1$

$\phi_{1,y} = 0$



ASSEMBLING LOCAL MATRICES

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \tfrac{1}{2}$$

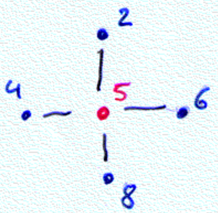
$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tfrac{1}{2}$$

LOCATION TABLE

I	4	5	2
II	7	5	4
III	7	8	5
IV	8	6	5
V	5	6	3
VI	5	3	2

	2	3	4	5	6	7	8
2	1 2	-1		-1 -1			
3	-1	1 1			-1		
4			1 2	-1 -1		-1	
5	$\boxed{-1\ -1}$		$\boxed{-1\ -1}$	$\boxed{2\ 1\ 1 \quad 2\ 1\ 1}$	$\boxed{-1\ -1}$		$\boxed{-1\ -1}$
6		-1		-1 -1	1 2		
7			-1			1 1	-1
8				-1 -1		-1	2 1

$$\frac{1}{2}\left(8U_5 - 2U_2 - 2U_4 - 2U_8 - 2U_6\right)$$



REGULAR LATTICE OF TURNER TRIANGLES = CENTRAL FINITE DIFFERENCES

$$\boxed{U_2 = 4}$$

$$\begin{bmatrix} 4 & 5 & 3 \\ 5 & 7 & 8 \\ 3 & 8 & 18 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad \begin{bmatrix} 1 - 5\,U_2 \\ 2 - 7\,U_2 \\ 3 - 8\,U_2 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 0 & 3 \\ 0 & 1 & 0 \\ 3 & 0 & 18 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_4 \end{bmatrix} = \begin{bmatrix} -19 \\ 4 \\ -29 \end{bmatrix}$$

Comment imposer les conditions essentielles ?

Une structure pour un système linéaire $Ax = b$

```
typedef struct {
    double **A;
    double  *B;
    int size;
} femFullSystem;
```

```
theSystem = femFullSystemCreate(2);
double **A = theSystem->A;
double  *B = theSystem->B;
A[0][0] = 1.0; A[0][1] = 1.0; B[0] = 4;
A[1][0] = 1.0; A[1][1] = 2.0; B[1] = 7;
femFullSystemEliminate(theSystem);
femFullSystemPrint(theSystem);
```

Nous allons faire les tâches suivantes :

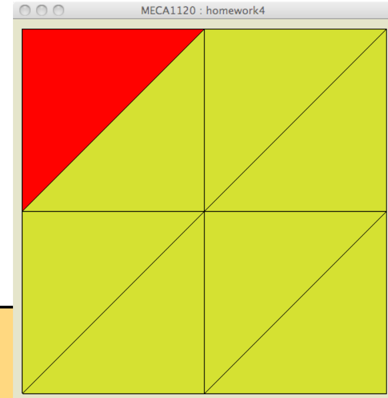
- Allouer la matrice et le vecteur
- Initialiser le système
- Imprimer le système
- Contraindre une valeur
- Résoudre le système

```
+1.0e+00 +1.0e+00 :  +1.0e+00
+1.0e+00 +1.0e+00 :  +3.0e+00
```

Résolution « en place »
par élimination de Gauss
de matrices symétriques
définies positives

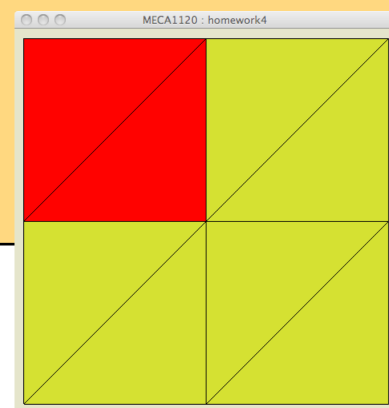
```
femFullSystem* femFullSystemCreate(int size);
void femFullSystemPrint(femFullSystem* mySystem);
void femFullSystemConstrain(femFullSystem* mySystem, int myNode, double myValue);
void femFullSystemEliminate(femFullSystem* mySystem);
```

Assemblage élement…



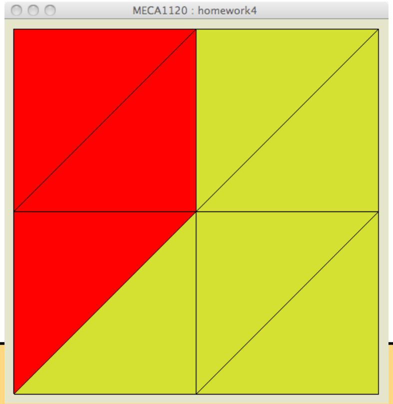

```
+1.0e+00 -5.0e-01          -5.0e-01                                        :  +1.7e-01
-5.0e-01 +5.0e-01                                                          :  +1.7e-01
                                                                           :  +0.0e+00
-5.0e-01                   +5.0e-01                                        :  +1.7e-01
                                                                           :  +0.0e+00
                                                                           :  +0.0e+00
                                                                           :  +0.0e+00
                                                                           :  +0.0e+00
                                                                           :  +0.0e+00
```

```
+1.0e+00 -5.0e-01          -5.0e-01                                        :  +1.7e-01
-5.0e-01 +1.0e+00                   -5.0e-01                               :  +3.3e-01
                                                                           :  +0.0e+00
-5.0e-01                   +1.0e+00 -5.0e-01                               :  +3.3e-01
         -5.0e-01          -5.0e-01 +1.0e+00                               :  +1.7e-01
                                                                           :  +0.0e+00
                                                                           :  +0.0e+00
                                                                           :  +0.0e+00
                                                                           :  +0.0e+00
```

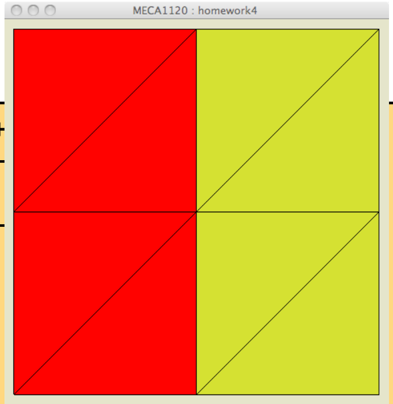


…par élément

Assemblage élément…



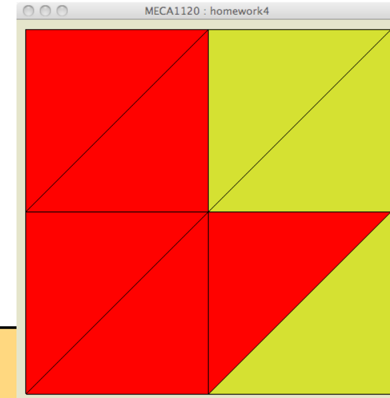
+1.0e+00	-5.0e-01	-5.0e-01				:	+1.7e-01
-5.0e-01	+1.0e+00		-5.0e-01			:	+3.3e-01
						:	+0.0e+00
-5.0e-01		+2.0e+00	-1.0e+00	-5.0e-01		:	+5.0e-01
	-5.0e-01	-1.0e+00	+1.5e+00			:	+3.3e-01
						:	+0.0e+00
		-5.0e-01		+5.0e-01		:	+1.7e-01
						:	+0.0e+00
						:	+0.0e+00



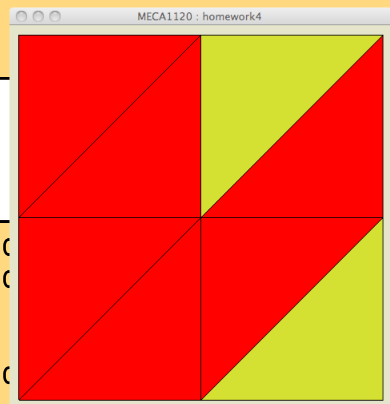
+		-5.0e-01				:	+1.7e-01
-			-5.0e-01			:	+3.3e-01
						:	+0.0e+00
-		+2.0e+00	-1.0e+00	-5.0e-01		:	+5.0e-01
		-1.0e+00	+2.0e+00		-5.0e-01	:	+5.0e-01
						:	+0.0e+00
		-5.0e-01		+1.0e+00	-5.0e-01	:	+3.3e-01
			-5.0e-01	-5.0e-01	+1.0e+00	:	+1.7e-01
						:	+0.0e+00

…par élément

Assemblage élément…



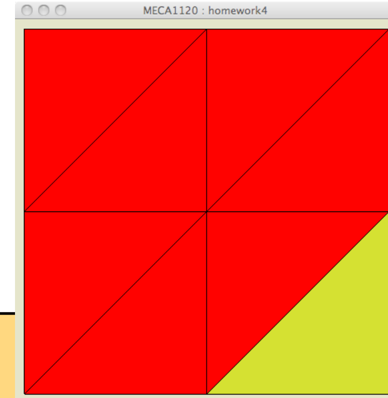

```
+1.0e+00 -5.0e-01           -5.0e-01                                                       +1.7e-01
-5.0e-01 +1.0e+00                    -5.0e-01                                              +3.3e-01
                                                                                       :   +0.0e+00
-5.0e-01                    +2.0e+00 -1.0e+00          -5.0e-01                        :   +5.0e-01
         -5.0e-01           -1.0e+00 +3.0e+00 -5.0e-01          -1.0e+00               :   +6.7e-01
                                     -5.0e-01 +5.0e-01                                 :   +1.7e-01
                            -5.0e-01                   +1.0e+00 -5.0e-01               :   +3.3e-01
                                    )e+00              -5.0e-01 +1.5e+00               :   +3.3e-01
                                                                                       :   +0.0e+00
```




```
+1.0e+00 -5.0e-(                                                                       :   +1.7e-01
-5.0e-01 +1.0e+(                    )e-01                                              :   +3.3e-01
                                              -5.0e-01                                 :   +1.7e-01
-5.0e-01                            )e+00              -5.0e-01                        :   +5.0e-01
         -5.0e-(                    5e+00 -1.0e+00              -1.0e+00               :   +8.3e-01
                                    1.0e+00 +1.5e+00                                   :   +3.3e-01
                            -5.0e-01                   +1.0e+00 -5.0e-01               :   +3.3e-01
                                     -1.0e+00          -5.0e-01 +1.5e+00               :   +3.3e-01
                                                                                       :   +0.0e+00
```

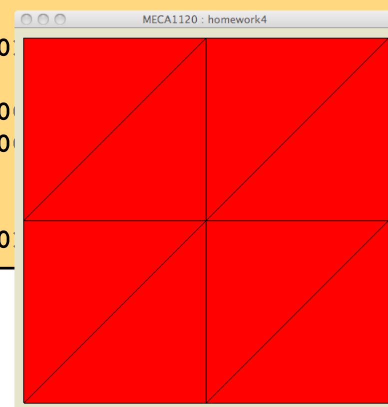
…par élément

Assemblage élément…



+1.0e+00	-5.0e-01		-5.0e-01						:	+1.7e-01
-5.0e-01	+2.0e+00	-5.0e-01		-1.0e+00					:	+5.0e-01
	-5.0e-01	+1.0e+00			-5.0e-01				:	+3.3e-01
-5.0e-01			+2.0e+00	-1.0e+00		-5.0e-01			:	+5.0e-01
	-1.0e+00		-1.0e+00	+4.0e+00	-1.0e+00		-1.0e+00		:	+1.0e+00
		-5.0e-01		-1.0e+00	+1.5e+00				:	+3.3e-01
			-5.0e-01			+1.0e+00	-5.0e-01		:	+3.3e-01
				-1.0e+00		-5.0e-01	+1.5e+00		:	+3.3e-01
									:	+0.0e+00

+1.0e+00	-5.0e-01		-5.0e-01						:	+1.7e-01
-5.0e-01	+2.0e+00	-5.0e-01		-1.0e+00					:	+5.0e-01
	-5.0e-01	+1.0e+00			-5.0e-0				:	+3.3e-01
-5.0e-01			+2.0e+00	-1.0e+00					:	+5.0e-01
	-1.0e+00		-1.0e+00	+4.0e+00	-1.0e+0				:	+1.0e+00
		-5.0e-01		-1.0e+00	+2.0e+0			e-01	:	+5.0e-01
			-5.0e-01						:	+3.3e-01
				-1.0e+00				e-01	:	+5.0e-01
					-5.0e-0			e+00	:	+1.7e-01



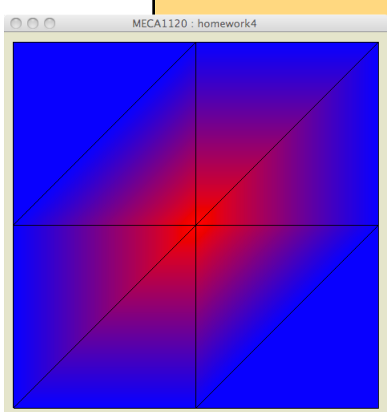
…par élément

Comment imposer les conditions frontières ?

```
def constrain(self,myNode,myValue):
    A = self.A; B = self.B; n = self.n
    for i in range(n):
      B[i] -= myValue * A[i,myNode]
      A[i,myNode] = 0
    for i in range(n):
      A[myNode,i] = 0
    A[myNode,myNode] = 1;
    B[myNode] = myValue;
```

```
for myEdge in theEdges.edges:
    if (myEdge[3] == -1) :
      theSystem.constrain(myEdge[0],0.0)
      theSystem.constrain(myEdge[1],0.0)
```

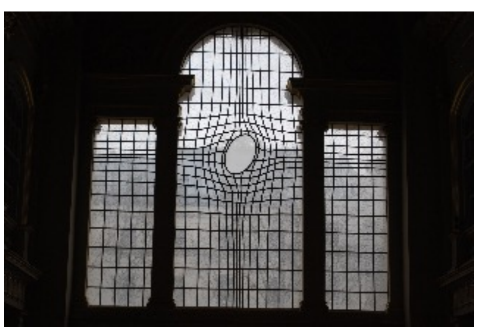
```
+1.0e+00                                                                                          :  +0.0e+00
            +1.0e+00                                                                              :  +0.0e+00
                        +1.0e+00                                                                  :  +0.0e+00
                                    +1.0e+00                                                      :  +0.0e+00
                                                +4.0e+00                                          :  +1.0e+00
                                                            +1.0e+00                              :  +0.0e+00
                                                                        +1.0e+00                  :  +0.0e+00
                                                                                    +1.0e+00      :  +0.0e+00
                                                                                            +1.0e+00 :  +0.0e+00
```

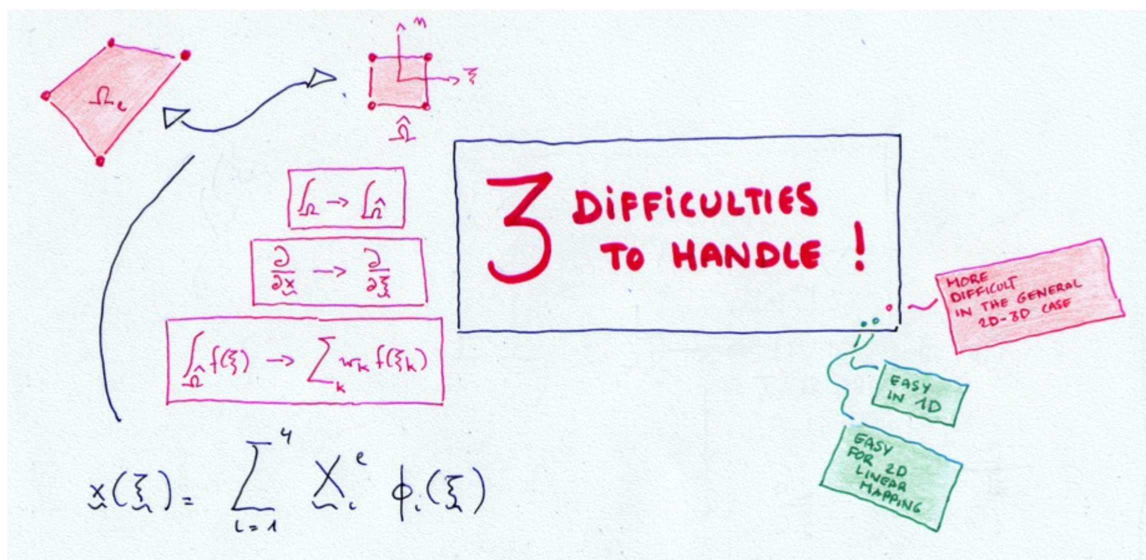


Les éléments triangulaires, c’est bien :-)
Mais, ici des éléments quadrilatères, ce serait mieux !

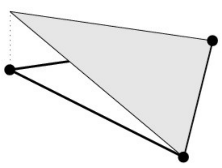
Et on fait comment pour des quadrilatères ?

$$
\begin{aligned}
A_{ij} &= \int_\Omega (\nabla \tau_i) \cdot (\nabla \tau_j) \, d\Omega, \\
B_i &= \int_\Omega f \tau_i \, d\Omega.
\end{aligned}
$$





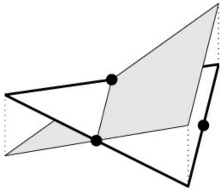
Fonctions de forme usuelles sur le triangle parent



$$
\begin{aligned}
\phi_1(\xi,\eta) &= (1-\xi-\eta),\\
\phi_2(\xi,\eta) &= \xi,\\
\phi_3(\xi,\eta) &= \eta.
\end{aligned}
$$

Fonctions linéaires continues : P_1-C_0

$$
\begin{aligned}
\phi_1(\xi,\eta) &= 1-2\eta,\\
\phi_2(\xi,\eta) &= -1+2(\xi+\eta),\\
\phi_3(\xi,\eta) &= 1-2\xi.
\end{aligned}
$$



Fonctions linéaires non-conformes : P_1-C_{-1}

$$
\begin{aligned}
\phi_1(\xi,\eta) &= 1-3(\xi+\eta)+2(\xi+\eta)^2,\\
\phi_2(\xi,\eta) &= \xi(2\xi-1),\\
\phi_3(\xi,\eta) &= \eta(2\eta-1),\\
\phi_4(\xi,\eta) &= 4\xi(1-\xi-\eta),\\
\phi_5(\xi,\eta) &= 4\xi\eta,\\
\phi_6(\xi,\eta) &= 4\eta(1-\xi-\eta).
\end{aligned}
$$

Fonctions quadratiques continues : P_2-C_0

Il s'agit de fonctions polynomiales définies sur chaque élement

En fonction du choix des fonctions de base, le degré de continuité est différent…
En général, on essaie d'avoir le degré de continuité le plus élevé…
Quoique… on aura quelques surprises !

Le triangle semble optimal pour obtenir des approximations polynomiales complètes…

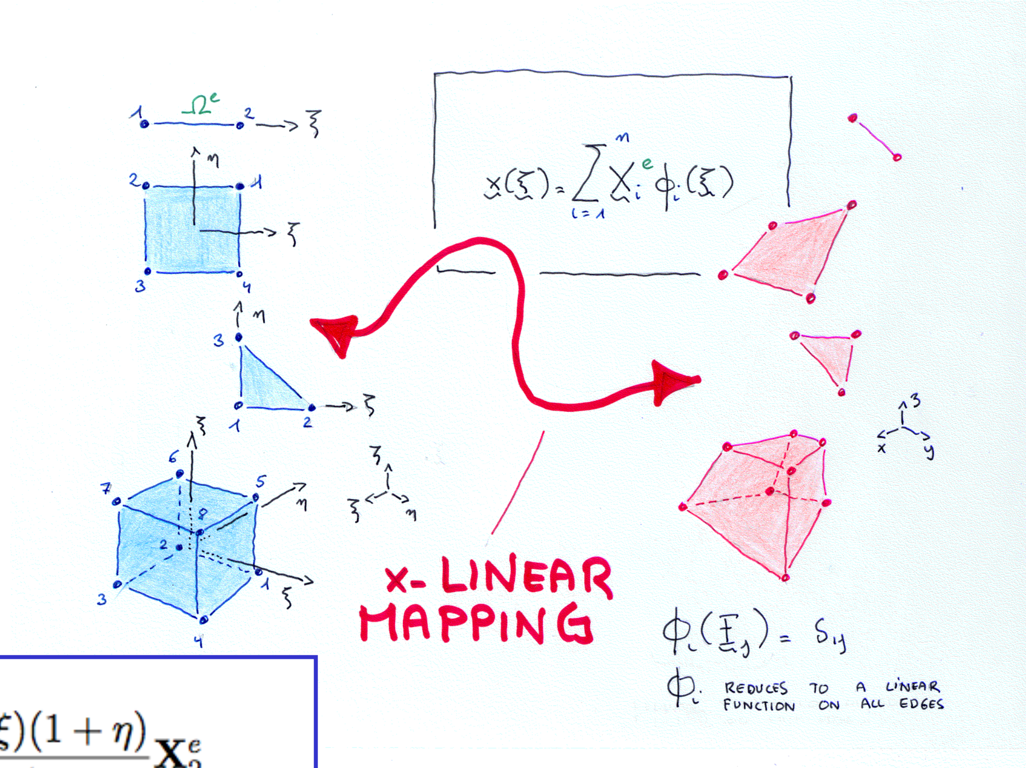
Triangle de Pascal

$$
\begin{array}{ccccccccc}
 & & & 1 & & & & & 0 \\
 & & \xi & & \eta & & & & 1 \\
 & \xi^2 & & \xi\eta & & \eta^2 & & & 2 \\
\xi^3 & & \xi^2\eta & & \xi\eta^2 & & \eta^3 & & 3
\end{array}
$$

Une des caractéristiques des éléments triangulaires est la possibilité de construire un ensemble de fonctions de forme complètes sur le triangle parent avec un nombre optimal de degrés de liberté.

… et pourtant !

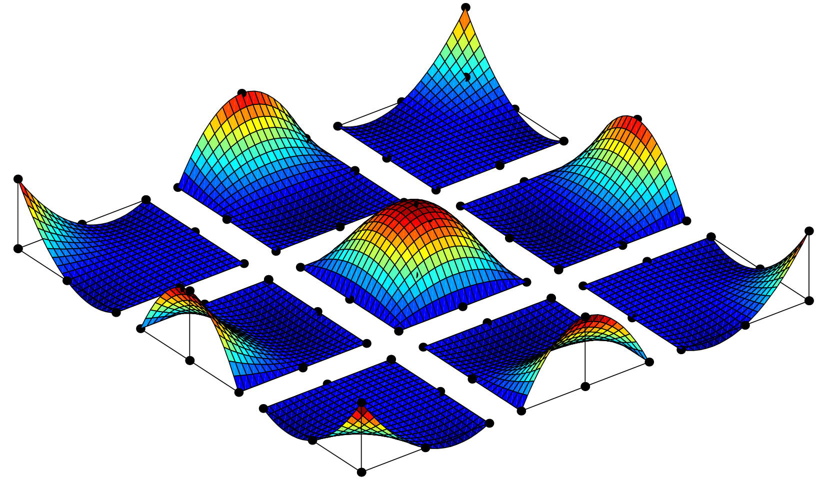
Et pourtant, on a aussi des maillages de quadrilatères, d’hexaèdres…

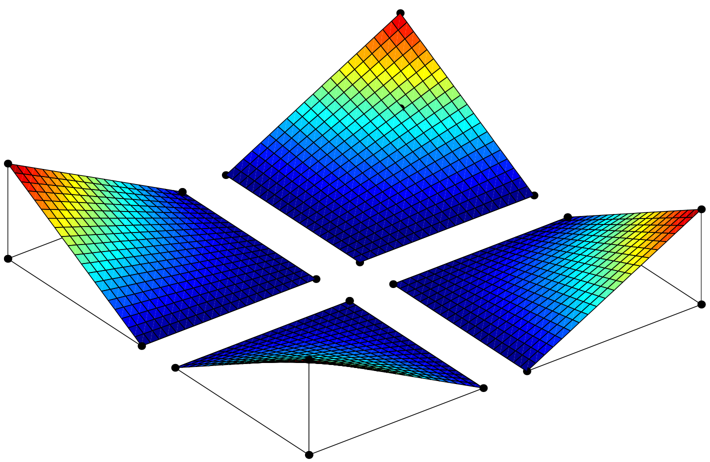


$$
\begin{aligned}
\mathbf{x}(\xi,\eta) \;=\; & \frac{(1+\xi)(1+\eta)}{4}\mathbf{X}_1^e + \frac{(1-\xi)(1+\eta)}{4}\mathbf{X}_2^e \\
& + \frac{(1-\xi)(1-\eta)}{4}\mathbf{X}_3^e + \frac{(1+\xi)(1-\eta)}{4}\mathbf{X}_4^e.
\end{aligned}
$$

Isomorphisme bilinéaire (non linéaire donc !) entre le carré parent et tous les autres quadrilatères

Eléments bilinéaires et biquadratiques





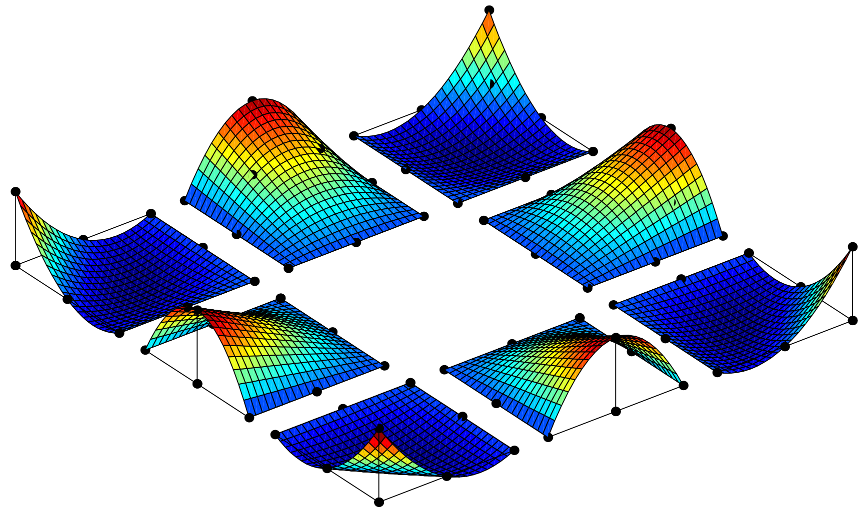
$$
\begin{aligned}
\phi_1(\xi,\eta) &= (1+\xi)(1+\eta)/4,\\
\phi_2(\xi,\eta) &= (1-\xi)(1+\eta)/4,\\
\phi_3(\xi,\eta) &= (1-\xi)(1-\eta)/4,\\
\phi_4(\xi,\eta) &= (1+\xi)(1-\eta)/4.
\end{aligned}
$$

Elément Q_1-C_0

$$
\begin{aligned}
\phi_1(\xi,\eta) &= \xi(1+\xi)\eta(1+\eta)/4,\\
\phi_2(\xi,\eta) &= -\xi(1-\xi)\eta(1+\eta)/4,\\
\phi_3(\xi,\eta) &= \xi(1-\xi)\eta(1-\eta)/4,\\
\phi_4(\xi,\eta) &= -\xi(1+\xi)\eta(1-\eta)/4,\\
\phi_5(\xi,\eta) &= (1+\xi)(1-\xi)\eta(1+\eta)/2,\\
\phi_6(\xi,\eta) &= -\xi(1-\xi)(1-\eta)(1+\eta)/2,\\
\phi_7(\xi,\eta) &= -(1-\xi)(1+\xi)\eta(1-\eta)/2,\\
\phi_8(\xi,\eta) &= \xi(1+\xi)(1-\eta)(1+\eta)/2,\\
\phi_9(\xi,\eta) &= (1-\xi)(1+\xi)(1-\eta)(1+\eta).
\end{aligned}
$$

Elément Q_2-C_0

Elle sert à rien la neuvième fonction ?

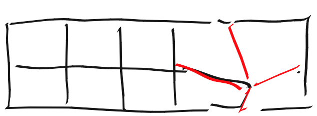


		1			0
	ξ		η		1
ξ^2		$\xi\eta$		η^2	2
	$\xi^2\eta$		$\xi\eta^2$		3
		$\boxed{\xi^2\eta^2}$	*Lagrange*		4

La neuvième fonction de Q_2-C_0 génère un terme d'ordre quatre alors qu'on n'est même pas complet à l'ordre 3 :-)

$$
\begin{aligned}
\phi_1(\xi,\eta) &= (1+\xi)(1+\eta)(\eta+\xi-1))/4,\\
\phi_2(\xi,\eta) &= (1-\xi)(1+\eta)(\eta-\xi-1))/4,\\
\phi_3(\xi,\eta) &= (1-\xi)(1-\eta)(-\eta-\xi-1))/4,\\
\phi_4(\xi,\eta) &= (1+\xi)(1-\eta)(-\eta+\xi-1))/4,\\
\phi_5(\xi,\eta) &= (1-\xi)(1+\xi)(1+\eta)/2,\\
\phi_6(\xi,\eta) &= (1-\xi)(1+\eta)(1-\eta)/2,\\
\phi_7(\xi,\eta) &= (1+\xi)(1-\xi)(1-\eta)/2,\\
\phi_8(\xi,\eta) &= (1+\xi)(1+\eta)(1-\eta)/2.
\end{aligned}
$$

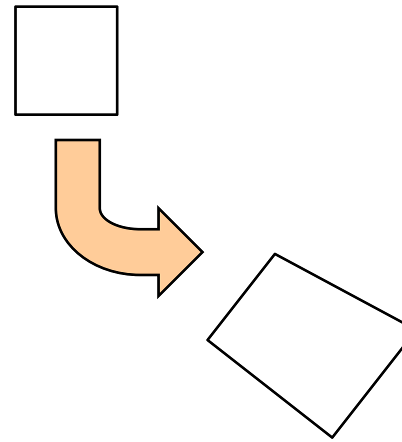
Elément de Serendip : S_2-C_0



L'erreur de Serendip !
La démonstration du recteur :-)

L'élément de Serendip (8 nœuds) et celui de Lagrange (9 nœuds) sont complets à l'ordre deux sur le carré parent

$$a_1 + a_2\xi + a_3\eta + a_4\xi\eta + a_5\xi^2 + a_6\eta^2,$$

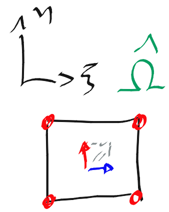


$$\begin{aligned} \mathbf{x}(\xi,\eta) \quad = \quad & \frac{(1+\xi)(1+\eta)}{4}\mathbf{X}_1^e + \frac{(1-\xi)(1+\eta)}{4}\mathbf{X}_2^e \\ & + \frac{(1-\xi)(1-\eta)}{4}\mathbf{X}_3^e + \frac{(1+\xi)(1-\eta)}{4}\mathbf{X}_4^e. \end{aligned}$$

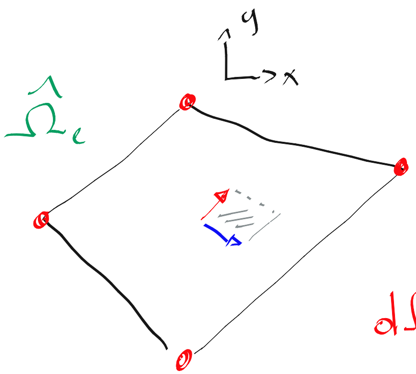
$$b_1 + b_2x(\xi,\eta) + b_3y(\xi,\eta) + b_4x(\xi,\eta)y(\xi,\eta) + b_5x^2(\xi,\eta) + b_6y^2(\xi,\eta),$$

$$c_1 + c_2\xi + c_3\eta + c_4\xi\eta + c_5\xi^2 + c_6\eta^2 + c_7\xi\eta^2 + c_8\eta\xi^2 + c_9\eta^2\xi^2.$$

L'élément de Serendip (8 nœuds) n'est pas complet à l'ordre deux sur un quadrilatère quelconque.
L'élément de Lagrange (9 nœuds) est complet à l'ordre deux sur un quadrilatère quelconque.



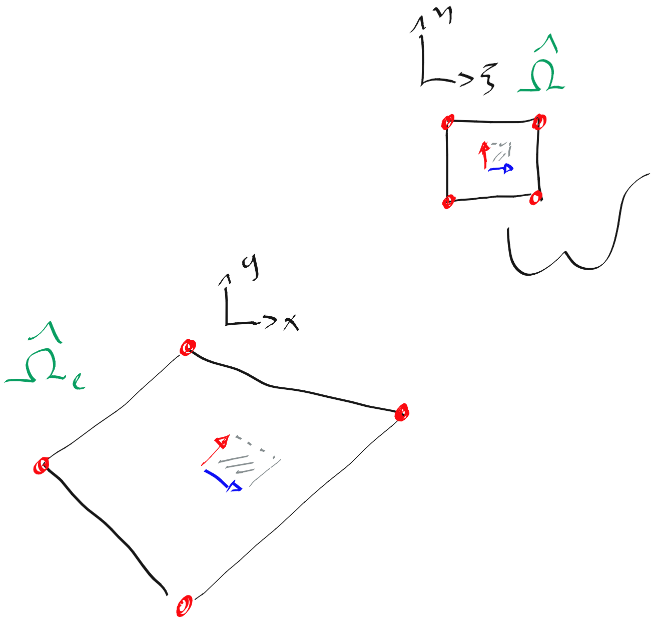
$d\hat{\Omega} = d\xi \, d\eta$



$$d\Omega_e = \left| \underbrace{\frac{\partial \mathbf{x}}{\partial \xi}}_{\sum \mathbf{X}_i^e \frac{\partial \phi_i}{\partial \xi}} d\xi \times \underbrace{\frac{\partial \mathbf{x}}{\partial \eta}}_{\sum \mathbf{X}_i^e \frac{\partial \phi_i}{\partial \eta}} d\eta \right| = \overbrace{\left| \frac{\partial \mathbf{x}}{\partial \eta} \times \frac{\partial \mathbf{x}}{\partial \xi} \right|}^{J_e(\xi,\eta)} \underbrace{d\xi \, d\eta}_{d\hat{\Omega}}$$

$$J_e = \left| \det \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \right|$$

Intégrer sur l'élément parent !



$$d\hat{\Omega} = d\xi \, d\eta$$

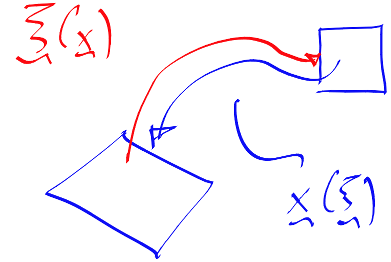
$$\underline{x}(\underline{\xi}) = \sum \underline{X}_i^e \, \phi_i(\underline{\xi})$$

$$d\underline{x}(\underline{\xi}) = \underbrace{\sum \underline{X}_i^e \, \frac{\partial \phi_i(\underline{\xi})}{\partial \underline{\xi}}}_{\frac{\partial \underline{x}}{\partial \underline{\xi}}} \cdot d\underline{\xi}$$

$$J_e = \left| \det\left(\frac{\partial \underline{x}}{\partial \underline{\xi}} \right) \right| \qquad \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

Calculer
le jacobien !

$$\begin{cases} \dfrac{\partial \phi_i}{\partial x} = \dfrac{\partial \phi_i}{\partial \xi}\dfrac{\partial \xi}{\partial x} + \dfrac{\partial \phi_i}{\partial \eta}\dfrac{\partial \eta}{\partial x} \\[2ex] \dfrac{\partial \phi_i}{\partial y} = \dfrac{\partial \phi_i}{\partial \xi}\dfrac{\partial \xi}{\partial y} + \dfrac{\partial \phi_i}{\partial \eta}\dfrac{\partial \eta}{\partial y} \end{cases}$$



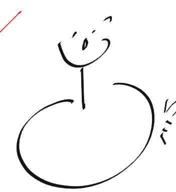
$$\frac{\partial \phi_i}{\partial x} = \frac{\partial \phi_i}{\partial \xi} \cdot \frac{\partial \xi}{\partial x}$$

BEURCK

$$\frac{\partial \phi_i}{\partial \xi} = \frac{\partial \phi_i}{\partial x} \cdot \frac{\partial x}{\partial \xi}$$

$$\sum X_i \frac{\partial \phi_i}{\partial \xi}$$

Calculer
les dérivées en x !

$\Big($  $\Big)^{-1}$ = 

$$\begin{bmatrix} \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \xi}{\partial y} \\[2ex] \dfrac{\partial \eta}{\partial x} & \dfrac{\partial \eta}{\partial y} \end{bmatrix} = \frac{1}{J_e} \begin{bmatrix} \dfrac{\partial y}{\partial \eta} & -\dfrac{\partial x}{\partial \eta} \\[2ex] -\dfrac{\partial y}{\partial \xi} & \dfrac{\partial x}{\partial \xi} \end{bmatrix}$$

$$\begin{bmatrix} \dfrac{\partial \phi_i}{\partial x} & \dfrac{\partial \phi_i}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \phi_i}{\partial \xi} & \dfrac{\partial \phi_i}{\partial \eta} \end{bmatrix} \cdot \begin{bmatrix} \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \xi}{\partial y} \\[2ex] \dfrac{\partial \eta}{\partial x} & \dfrac{\partial \eta}{\partial y} \end{bmatrix}$$

$$= \frac{1}{J_e} \begin{bmatrix} \dfrac{\partial y}{\partial \eta} & -\dfrac{\partial x}{\partial \eta} \\[2ex] -\dfrac{\partial y}{\partial \xi} & \dfrac{\partial x}{\partial \xi} \end{bmatrix}$$

Et zou !

Et faire des intégrations numériques !

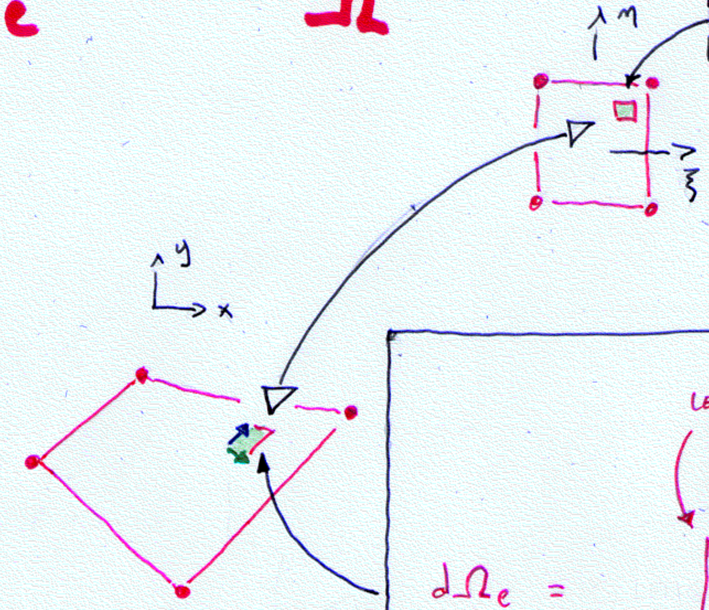
$$\frac{\partial \phi_i}{\partial x} = \underbrace{\frac{1}{J_e}\left[\frac{\partial \phi_i}{\partial \xi}\cdot\frac{\partial y}{\partial \eta} - \frac{\partial \phi_i}{\partial \eta}\cdot\frac{\partial y}{\partial \xi}\right]}_{f(\xi,\eta)}$$

QUOTIENT DE POLYNOMES si $J_e \neq$ CST

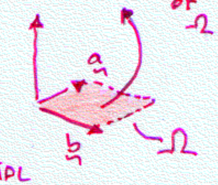
GAUSS LEGENDRE !

$$\int_{\Omega_e} \rightarrow \int_{\hat{\Omega}}$$

$$d\hat{\Omega} = d\xi \, d\eta$$



$|\underline{a} \times \underline{b}| =$ AREA OF Ω

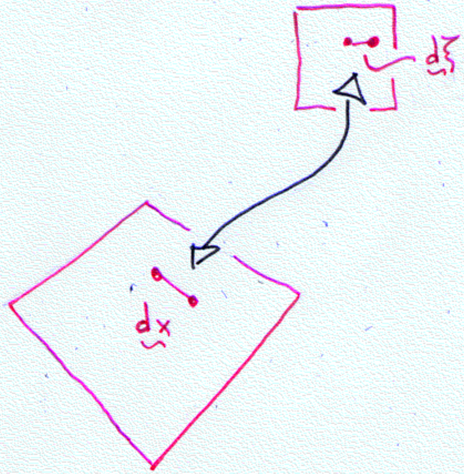


LENGTH

VECTORIAL PRODUCT

$$d\Omega_e = \left| \left(\frac{\partial \underline{x}}{\partial \underline{\xi}} \, d\xi \times \frac{\partial \underline{x}}{\partial \eta} \, d\eta \right) \right|$$

$$= \underbrace{\left| \frac{\partial \underline{x}}{\partial \underline{\xi}} \times \frac{\partial \underline{x}}{\partial \eta} \right|}_{\det\left(\frac{\partial \underline{x}}{\partial \underline{\xi}} \right)} \underbrace{d\xi \, d\eta}_{d\hat{\Omega}}$$



$$\underset{\sim}{x}(\underset{\sim}{\xi}) = \sum_{i=1}^{4} \underset{\sim}{X}_i^e \, \phi_i(\underset{\sim}{\xi})$$

$$d\underset{\sim}{x}(\underset{\sim}{\xi}) = \boxed{\underbrace{\sum_{i=1}^{4} \underset{\sim}{X}_i^e \, \frac{\partial \phi_i}{\partial \underset{\sim}{\xi}}(\underset{\sim}{\xi})}_{\frac{\partial \underset{\sim}{x}}{\partial \underset{\sim}{\xi}} \,=\, \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}}} \cdot d\underset{\sim}{\xi}$$

$$\begin{bmatrix} dx \\ dy \end{bmatrix} \qquad \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}$$

HOW TO CALCULATE J_e ?

$$J^e = \det\left(\frac{\partial \underset{\sim}{x}}{\partial \underset{\sim}{\xi}}\right)$$

$$\frac{\partial}{\partial \underline{x}} \rightarrow \frac{\partial}{\partial \underline{\xi}}$$

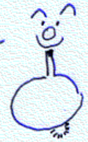
IT IS NOT NECESSARY TO CALCULATE $\underline{\xi}(\underline{x})$!

$\nabla \phi_i$

$$\frac{\partial \phi_i}{\partial \underline{x}} = \frac{\partial \phi_i}{\partial \underline{\xi}} \cdot \frac{\partial \underline{\xi}}{\partial \underline{x}}$$

$$\frac{\partial \phi_i}{\partial \underline{\xi}} = \frac{\partial \phi_i}{\partial \underline{x}} \cdot \frac{\partial \underline{x}}{\partial \underline{\xi}}$$

$$= \sum_{i=1}^{4} \underline{X}_i^e \frac{\partial \phi_i}{\partial \underline{\xi}}$$

$\triangleright$  $= \Big($  $\Big)^{-1}$

$$\begin{bmatrix} \partial \xi / \partial x & \partial \xi / \partial y \\ \partial \eta / \partial x & \partial \eta / \partial y \end{bmatrix} = \frac{1}{J_e} \begin{bmatrix} \partial y / \partial \eta & -\partial x / \partial \eta \\ -\partial y / \partial \xi & \partial x / \partial \xi \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial \phi_i}{\partial x} & \frac{\partial \phi_i}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \phi_i}{\partial \xi} & \frac{\partial \phi_i}{\partial \eta} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix}$$

$$\frac{\partial \phi_i}{\partial x} = \underbrace{\frac{1}{J_e} \left(\overbrace{\frac{\partial y}{\partial \eta}}^{\sum Y_i \frac{\partial \phi_i}{\partial \eta}} \frac{\partial \phi_i}{\partial \xi} - \overbrace{\frac{\partial y}{\partial \xi}}^{\sum Y_i \frac{\partial \phi_i}{\partial \xi}} \frac{\partial \phi_i}{\partial \eta} \right)}_{\text{FCT}(\xi, \eta)}$$

$$\boxed{A_{ij}^e = \int_{\hat{\Omega}} \underbrace{\underbrace{\underline{\nabla} \phi_i \cdot \underline{\nabla} \phi_j}_{\text{FCT}(\xi, \eta)} \; J_e(\xi, \eta)}_{\text{FCT}(\xi, \eta)} \; d\hat{\Omega}}$$

YOU HAVE JUST TO INTEGRATE NOW !

$$\int_{\hat{\Omega}} \underbrace{f(\underline{\xi})}_{\boxed{\underbrace{\underline{\nabla}\phi_i \cdot \underline{\nabla}\phi_j \;\; J_e(\underline{\xi})}_{f(\underline{\xi})}}} \rightarrow \sum_k \underbrace{w_k}_{\boxed{\text{WEIGHTS}}} \; f(\underbrace{\underline{\xi}_k}_{\boxed{\text{INTEGRATION POINTS}}})$$

NUMERICAL INTEGRATION REQUIRED !

ANALYTICAL WAY IS IMPOSSIBLE !

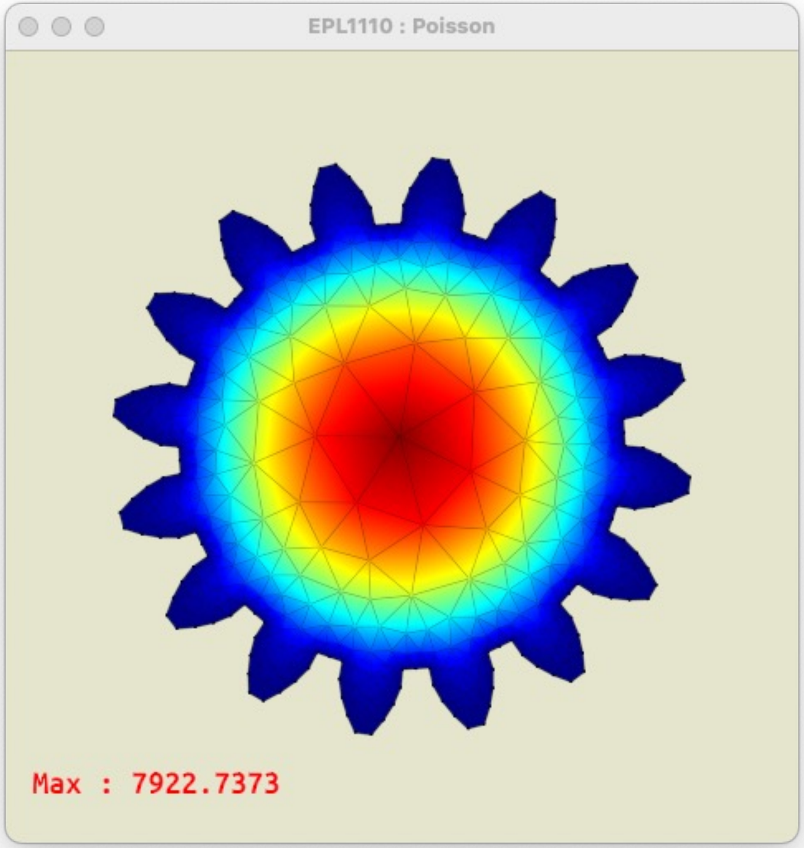
$\sum w_k = 1$!

IF $\int_{\hat{\Omega}} = 1$

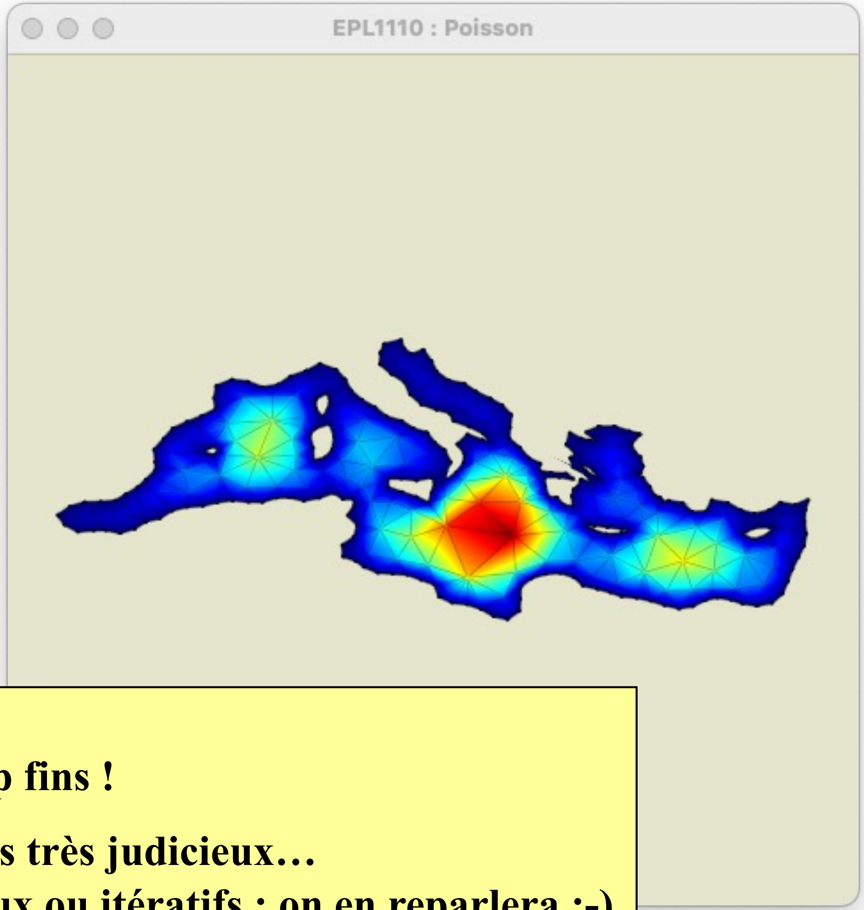
TO BE SELECTED IN A CLEVER WAY

GAUSS LEGENDRE QUADRATURE !

Et zou
On est prêt !



Calculer la solution du problème de Poisson avec des triangles linéaires ou des quads bilinéaires… Le jacobien n'est pas toujours constant !
Une code unique pour les deux cas !



Le code est très très lent…
Ne pas essayer des maillages trop fins !

Utiliser un solveur plein n’est pas très judicieux…
On fera appel à des solveurs creux ou itératifs : on en reparlera :-)

Un petit Poisson perdu dans la Méditerranée…