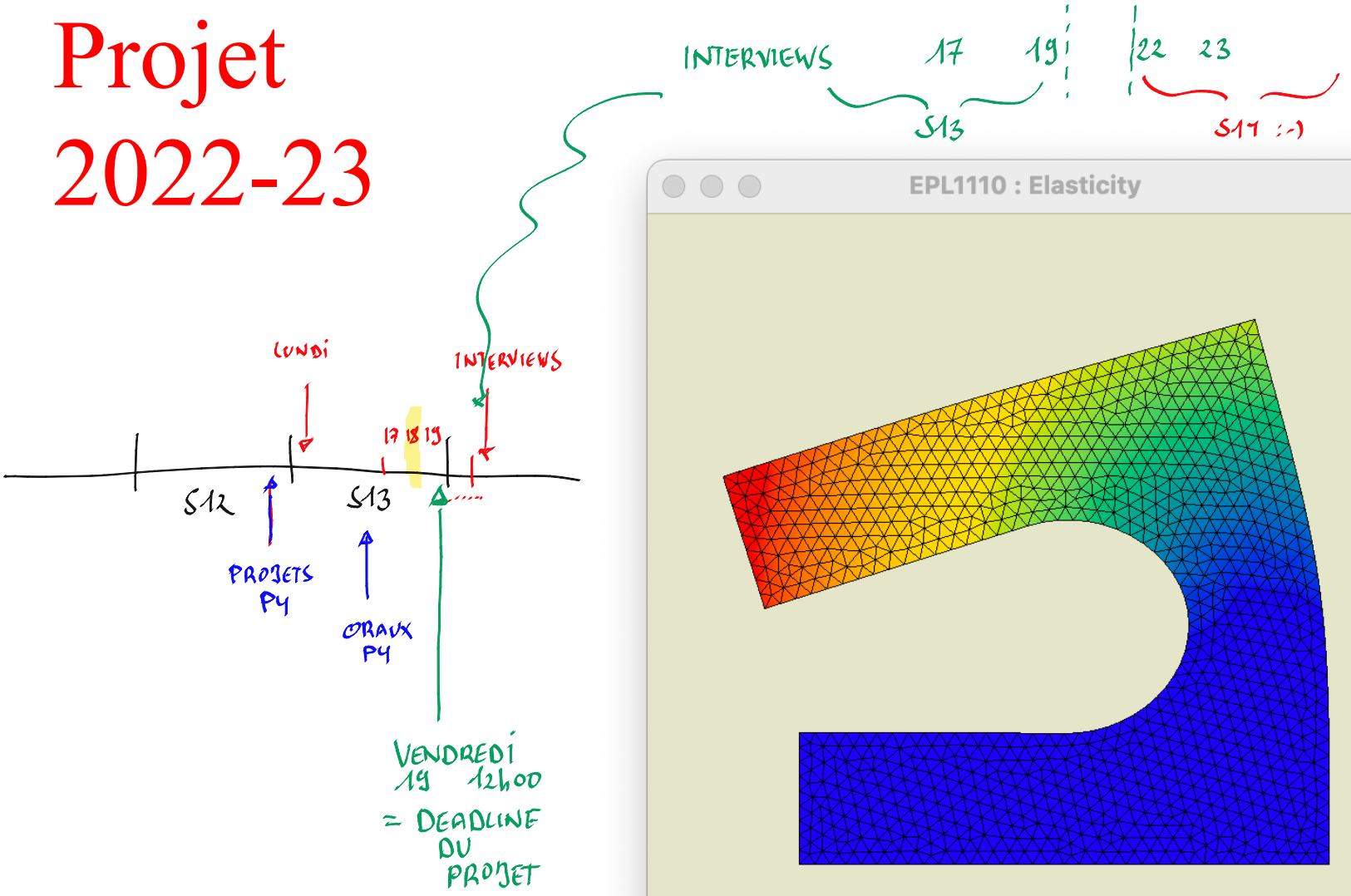
Projet 2022-23



Ecrire un code informatique efficace pour l'élasticité linéaire plane

Tensions planes et déformations planes
Triangles linéaires
Quads bilinéaires
Problèmes axisymétriques
Conditions essentielles en xy et en normale/tangentielle
Conditions naturelles en xy et en normale/tangentielle

Quelques mots sur les IO !

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

Données
Sorties

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
```

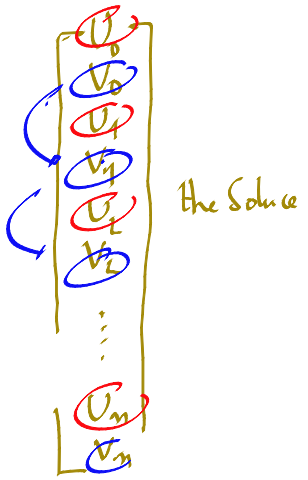
myFem

U.txt

V.txt

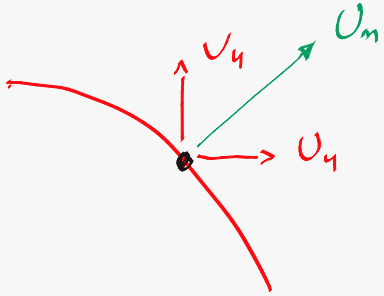
```
Size 335
 0.0000000e+00 0.0000000e+00 6.7147131e-07
-3.9548498e-07-2.3803038e-07 7.0231387e-08
 8.5720310e-08 9.1514097e-08 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00
 0.0000000e+00 0.0000000e+00 0.0000000e+00
```

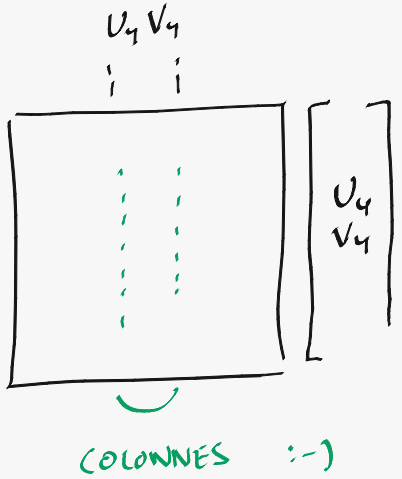
Le programme de calcul !




```
#include "fem.h"

int main(void)
{
    femGeo* theGeometry = geoGetGeometry();
    geoMeshRead("../data/mesh.txt");
    femProblem* theProblem =
              femElasticityRead(theGeometry,"../data/problem.txt");
    femElasticityPrint(theProblem);
    double *theSoluce = femElasticitySolve(theProblem);
    femNodes *theNodes = theGeometry->theNodes;
    femFieldWrite(theNodes->nNodes,2,&theSoluce[0],"../data/U.txt");
    femFieldWrite(theNodes->nNodes,2,&theSoluce[1],"../data/V.txt");
     femElasticityFree(theProblem);
    geoFree();
    return 0;
}
```





..... EQUATION POUR U_N → $U_N = \alpha$

EQUATION POUR U_T

COLONNES :-)

Ecrire le fichier

```
void femElasticityWrite(femProblem *theProblem, const char *filename)
{
   FILE* file = fopen(filename,"w");

   switch (theProblem->planarStrainStress) {
      case PLANAR_STRESS : fprintf(file,"Type of problem    :  Planar stresses  \n"); break;
      case PLANAR_STRAIN : fprintf(file,"Type of problem    :  Planar strains \n"); break;
      case AXISYM        : fprintf(file,"Type of problem    :  Axi-symetric problem \n"); break;
      default :            fprintf(file,"Type of problem    :  Undefined  \n"); break; }
   fprintf(file,"Young modulus      : %14.7e  \n",theProblem->E);
   fprintf(file,"Poisson ratio      : %14.7e  \n",theProblem->nu);
   fprintf(file,"Mass density       : %14.7e  \n",theProblem->rho);
   fprintf(file,"Gravity            : %14.7e  \n",theProblem->g);

   for(int i=0; i < theProblem->nBoundaryConditions; i++) {
        femBoundaryCondition *theCondition = theProblem->conditions[i];
        double value = theCondition->value;
        fprintf(file,"Boundary condition : ");
        switch (theCondition->type) {
            case DIRICHLET_X : fprintf(file," Dirichlet-X       = %14.7e ",value); break;
            case DIRICHLET_Y : fprintf(file," Dirichlet-Y       = %14.7e ",value); break;
            default :          fprintf(file," Undefined        = %14.7e ",value); break; }

        fprintf(file,": %s\n",theCondition->domain->name); }
   fclose(fil
}
```

```
Type of problem    :  Planar strains
Young modulus      :  2.1100000e+11
Poisson ratio      :  3.0000000e-01
Mass density       :  7.8500000e+03
Gravity            :  9.8100000e+00
Boundary condition :  Dirichlet-X       =  0.0000000e+00 : Symetry
Boundary condition :  Dirichlet-Y       =  0.0000000e+00 : Bottom
```

Lire le fichier

Dirichlet - X
Neumann - Y N T

```
femProblem* femElasticityRead(femGeo* theGeometry, const char *filename)
{
    FILE* file = fopen(filename,"r");
    femProblem *theProblem = malloc(sizeof(femProblem));
    theProblem->nBoundaryConditions = 0;
    theProblem->conditions = NULL;


    while (feof(file) != TRUE) {
        ErrorScan(fscanf(file,"%19[^\n]s \n",(char *)&theLine));
        if (strncasecmp(theLine,"Type of problem     ",19) == 0) {
            ErrorScan(fscanf(file,":  %[^\n]s \n",(char *)&theArgument));
        if (strncasecmp(theArgument,"Axi-symetric problem",13) == 0)
               theProblem->planarStrainStress = AXISYM; }
        if (strncasecmp(theLine,"Young modulus       ",19) == 0) {
            ErrorScan(fscanf(file,":  %le\n",&theProblem->E)); }
        if (strncasecmp(theLine,"Boundary condition  ",19) == 0) {
            ErrorScan(fscanf(file,":  %19s = %le : %[^\n]s\n",(char *)&theArgument,
                                    &value,(char *)&theDomain));
            if (strncasecmp(theArgument,"Dirichlet-X",19) == 0)
                typeCondition = DIRICHLET_X;
            femElasticityAddBoundaryCondition(theProblem,theDomain,typeCondition,value); }
        ErrorScan(fscanf(file,"\n")); }
    fclose(file);
    return th
}
```

```
Type of problem     :  Planar strains
Young modulus       :  2.1100000e+11
Poisson ratio       :  3.0000000e-01
Mass density        :  7.8500000e+03
Gravity             :  9.8100000e+00
Boundary condition  :  Dirichlet-X        =  0.0000000e+00 : Symetry
Boundary condition  :  Dirichlet-Y        =  0.0000000e+00 : Bottom
```

Idem pour la géométrie !

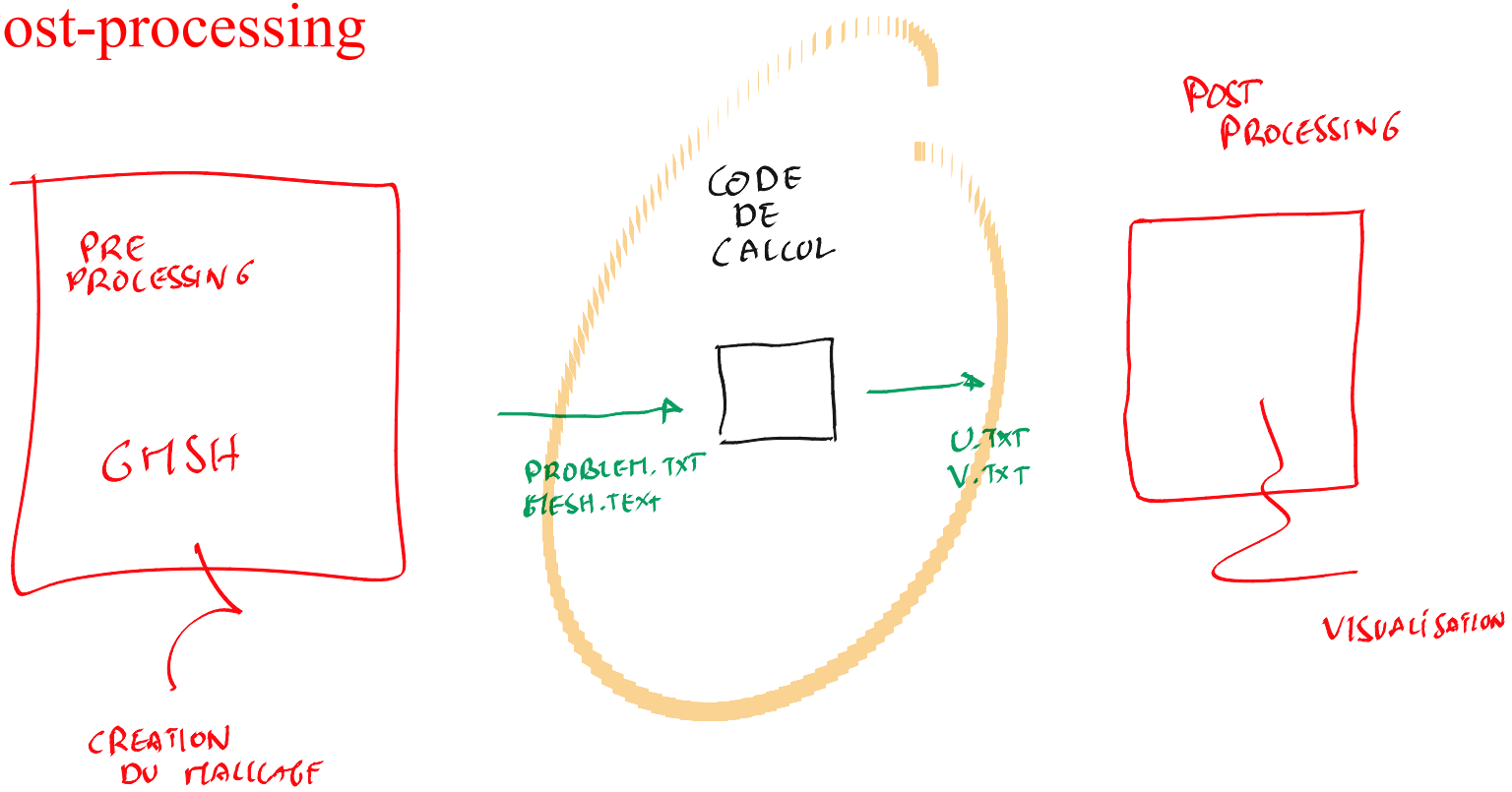
```
Number of nodes 335
     0 :  0.0000000e+00  1.0000000e+00

   333 :  2.3290442e-01  2.1703147e-01
   334 :  2.5379193e-01  1.9828855e-01
Number of edges 108
     0 :       0       8

   106 :     106     107
   107 :     107       7
Number of quads 280
     0 :     194     234     264     189

   278 :     333     321     305     334
   279 :     320     330     332     331
Number of domains 8
  Domain :       0
  Name : Symetry
  Number of elements :      20
     0      1      2      3      4      5      6      7      8      9
    10     11     12     13     14     15     16     17     18     19
```

Pre-processing
Finite Element computation
Post-processing



Questions
Réponses