

Screenshot of a Discord chat window for the server "EPL Bac & Master". The channel is "#elements-finis". The chat history shows several messages:

- favicc12 (02/02/2025 18:58): ah ok merci
- mriiz (04/02/2025 12:40): Hello, je recherche un binôme pour le projet et les devoirs, si ça interesse quelqu'un vous pouvez me dm
- Ralofurry de Feurbois* (Hier à 13:17): Askip c'est ici que je dois dire des conneries pour passer sur le grand écran ? # agora
- ShrendQuickx (Hier à 13:20): Peut être que tu apparaitras pendant le cours de vendredi !
- le.conquerant (Hier à 19:52): Vive éléments finis ^^
- BombaTomx:-) (Aujourd'hui à 15:27): Vous avez ça aussi?

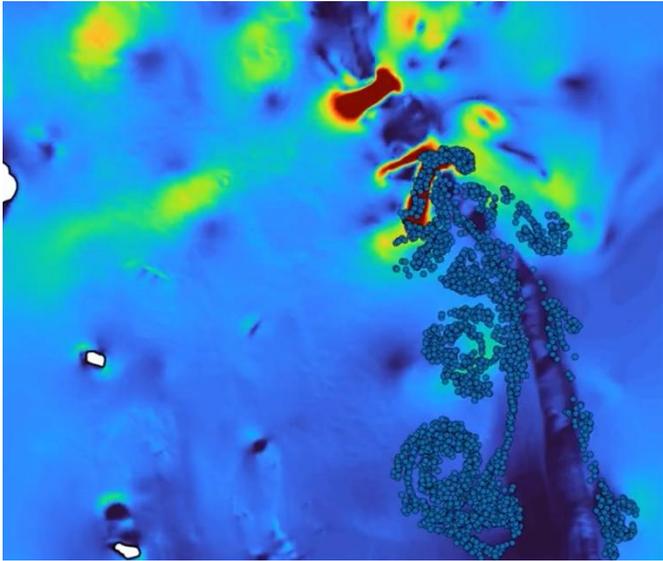
The message from BombaTomx:-) includes an image of a triangular mesh with red dots at the nodes and a text box below it that reads: $\text{Integral} = 5.4303101e-01$.

The interface shows a sidebar with various channels, a search bar at the top right, and a message input field at the bottom.

Introduction aux éléments finis



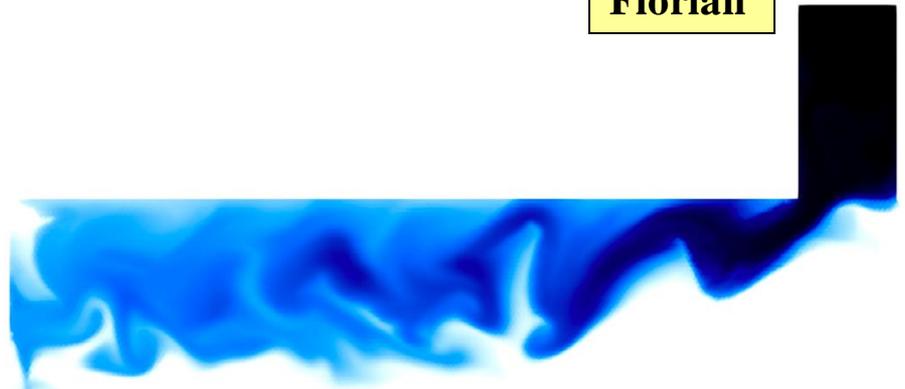
Introduction aux éléments finis

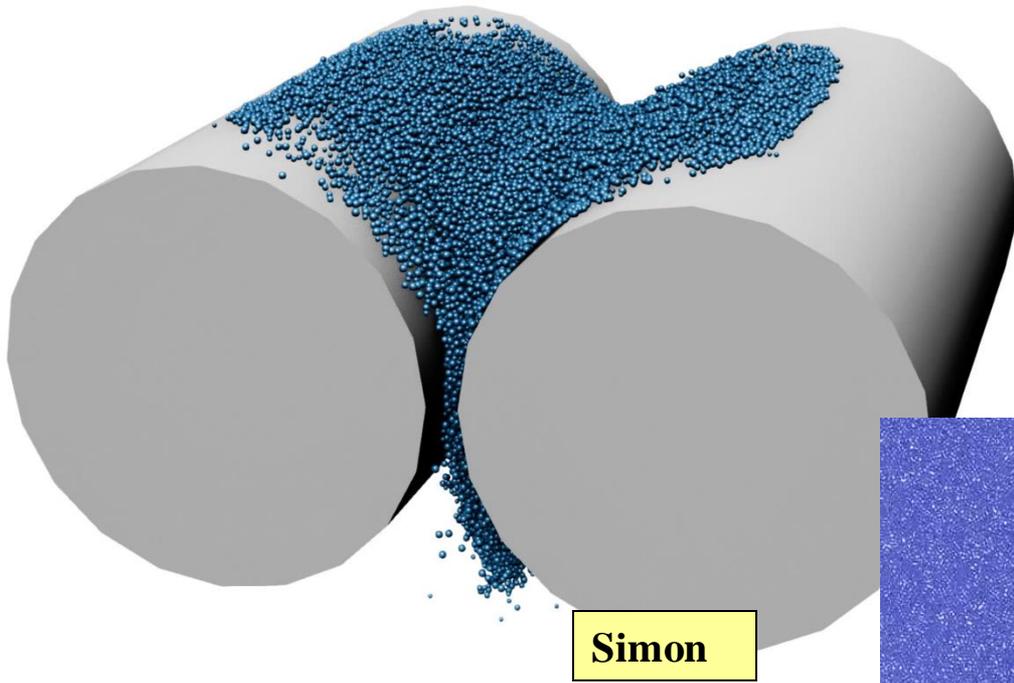


Miguel

Finite Elements

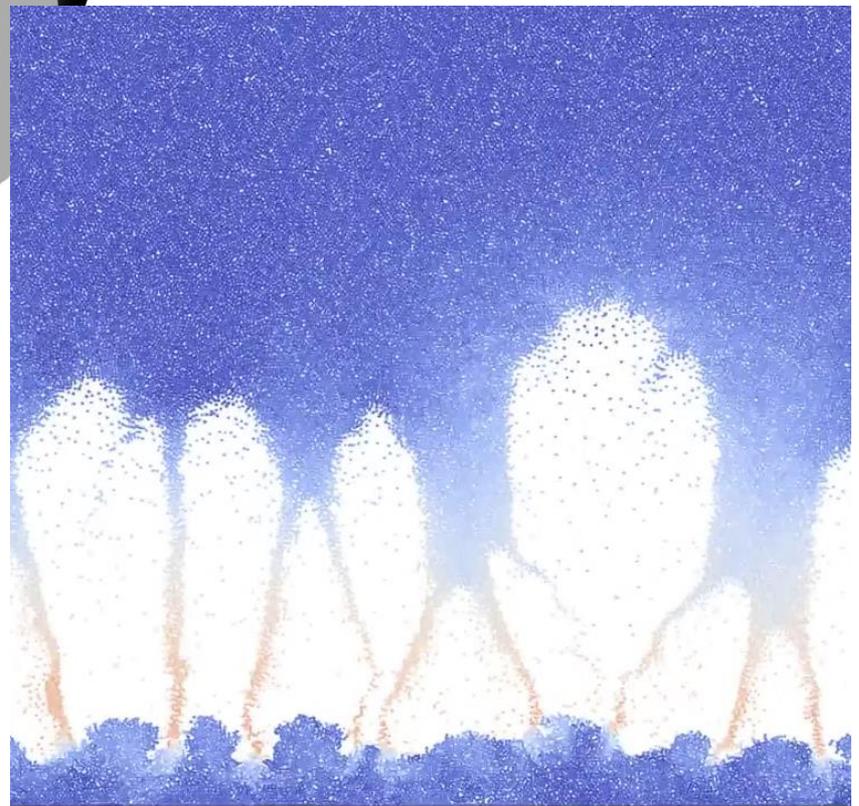
Florian





Simon

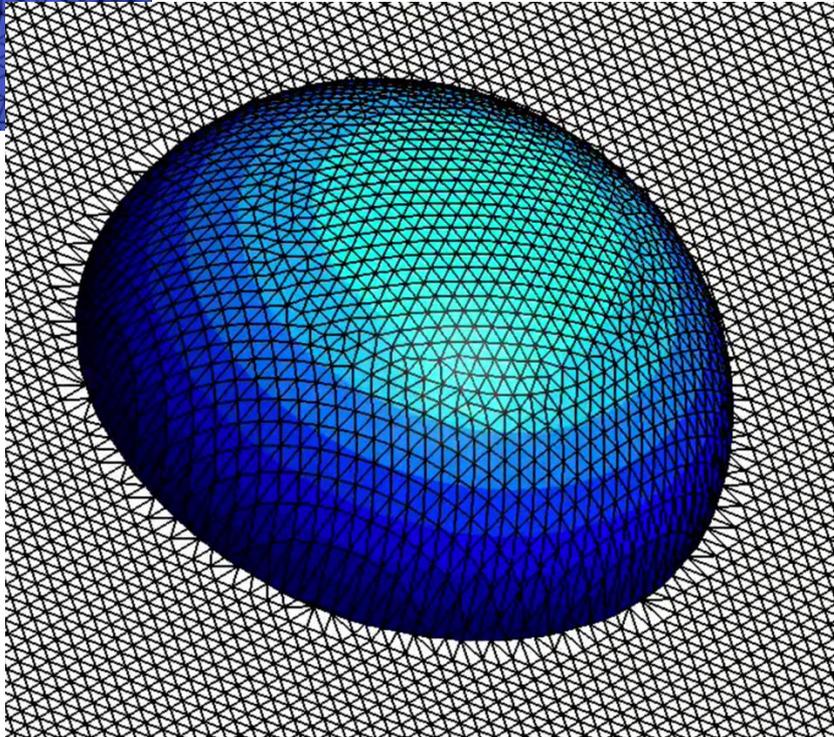
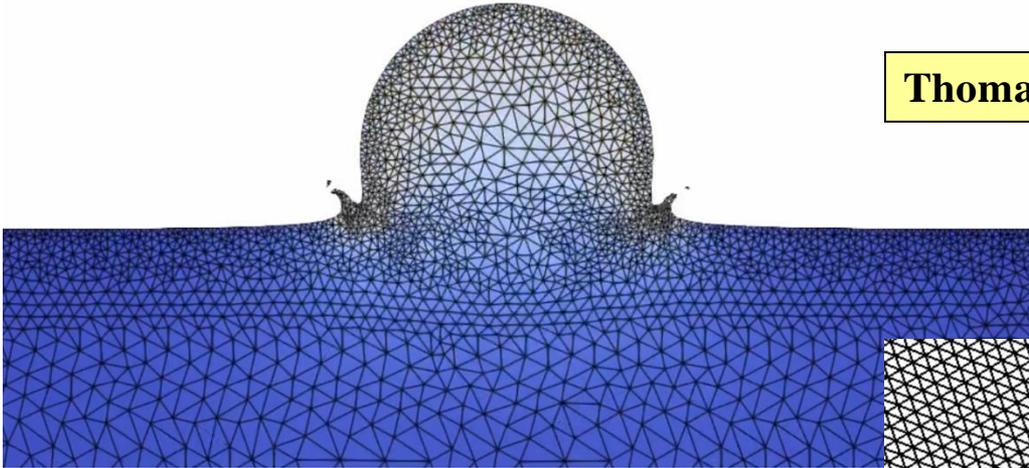
Michel



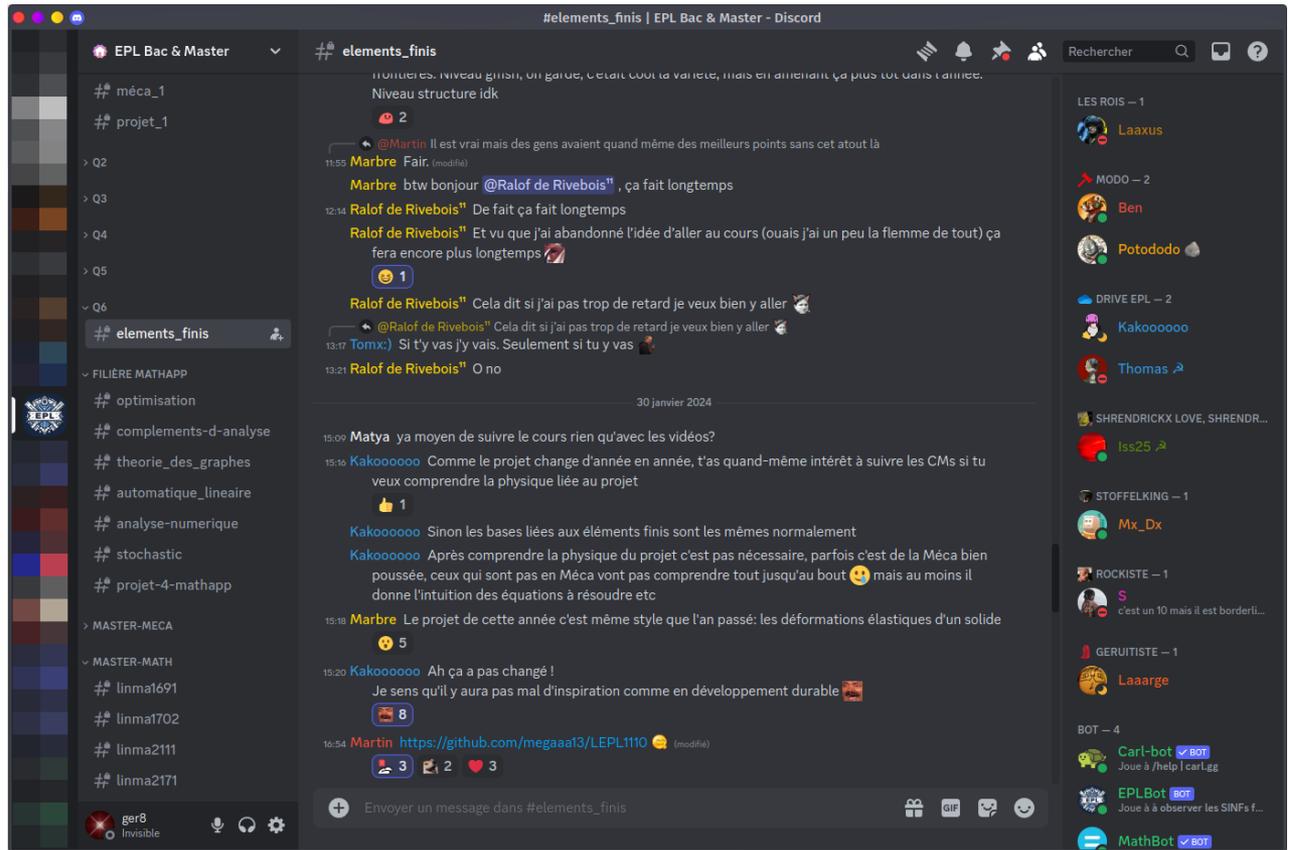
Finite Elements

Thomas

Alexandre



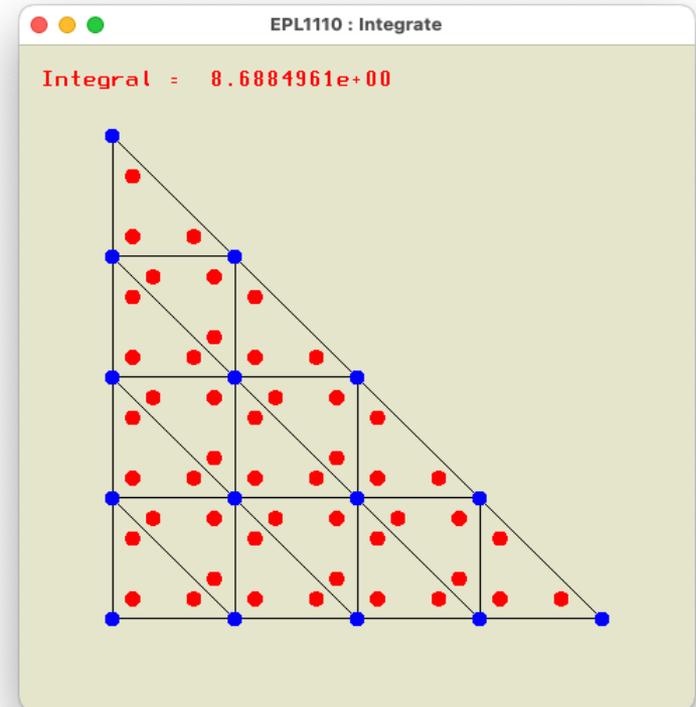
Finite Elements



Faut-il seulement
regarder les videos ?

Homework 1

$$\underbrace{\int_{\hat{\Omega}} f(x, y) \, dx \, dy}_I \approx \underbrace{\sum_{k=1}^3 w_k f(x_k, y_k)}_{I_h}$$



Ecrire la règle de Hammer

hello.c

```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
    return 0;
}
```

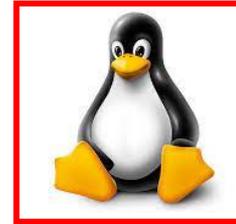
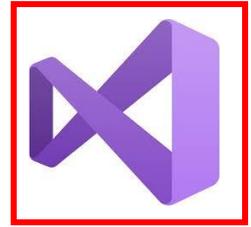
*Vous aimeriez apprendre à programmer,
mais vous ne savez pas par où commencer ?*

*(autrement dit : vous en avez marre des cours trop
compliqués que vous ne comprenez pas ? :-)*



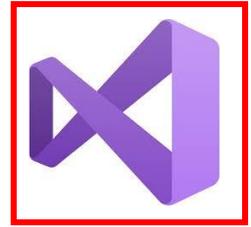
```
Cours1 -- bash -- 71x16
Last login: Fri Feb 4 08:32:27 on ttys001
(base) mac-1U0-352:~ vl$ cd LaTeX/ep111110-2122/Slides/Cours1
(base) mac-1U0-352:Cours1 vl$ ls
a.out  hello.c
(base) mac-1U0-352:Cours1 vl$ touch hello.c
(base) mac-1U0-352:Cours1 vl$ cc hello.c
(base) mac-1U0-352:Cours1 vl$ ./a.out
hello, world
(base) mac-1U0-352:Cours1 vl$
```

Comment compiler hello.c sur votre ordinateur ?



```
Cours1 -- -bash -- 93x27
(base) mac-100-352:Cours1 vl$
(base) mac-100-352:Cours1 vl$ touch hello.c
(base) mac-100-352:Cours1 vl$ cc -c hello.c
(base) mac-100-352:Cours1 vl$ ls
hello.c hello.o
(base) mac-100-352:Cours1 vl$ cc -o hello hello.c
(base) mac-100-352:Cours1 vl$ ./hello
hello, world
(base) mac-100-352:Cours1 vl$ cc -o yep hello.o
(base) mac-100-352:Cours1 vl$ ./yep
hello, world
(base) mac-100-352:Cours1 vl$
```

Avoir un compilateur sous Windows ?



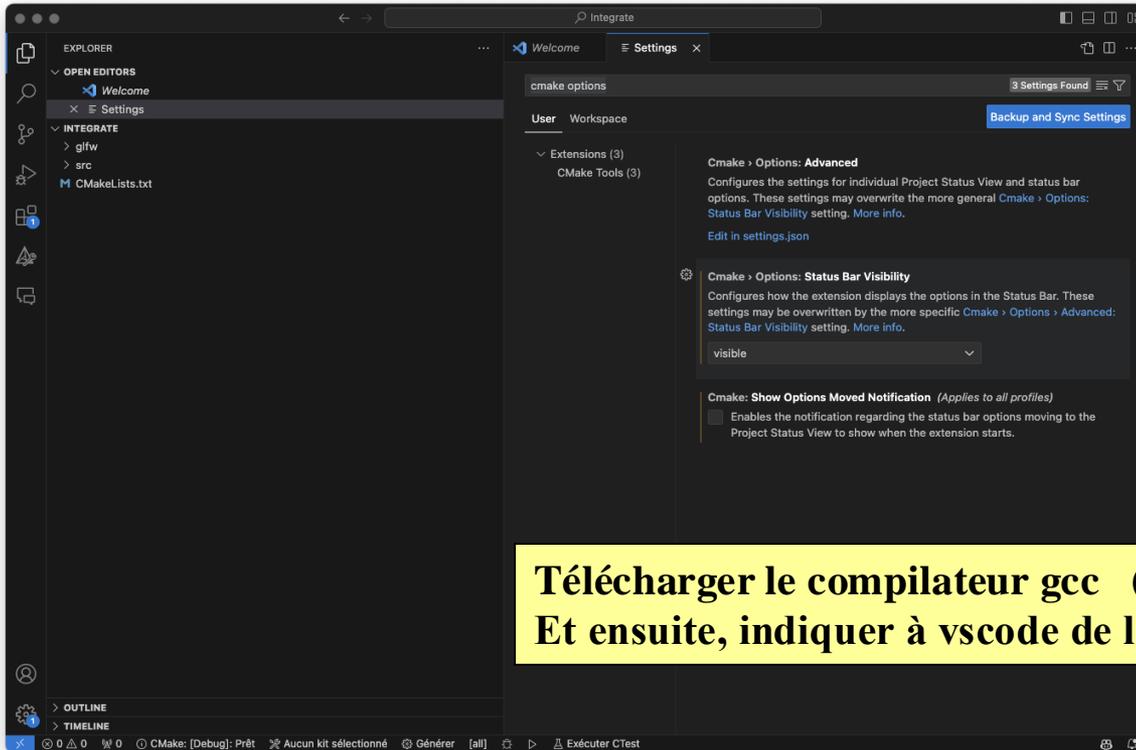
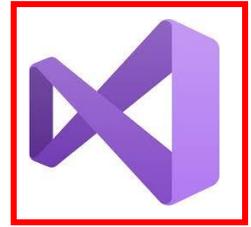
The screenshot shows a web browser window with the URL <https://meubank.github.io/tdm-gcc/download/>. The page title is "Download | tdm-gcc". The main heading is "Download". Below it, the text reads: "The easiest way to get TDM-GCC is via an installer." Underneath, it says "Download a TDM-GCC installer:". There are three download options listed in blue buttons:

- tdm-gcc-webdl.exe**: Minimal online installer. Select the components you want, and it downloads and unpacks them. Either edition, latest release only. (GCC 10.3.0)
- tdm64-gcc-10.3.0-2.exe**: 64+32-bit MinGW-w64 edition. Includes GCC C/C++, GNU binutils, mingw32-make, GDB (64-bit), the MinGW-w64 runtime libraries and tools, and the windows-default-manifest package.
- tdm-gcc-10.3.0.exe**: 32-bit-only MinGW.org edition. Includes GCC C/C++, GNU binutils, mingw32-make, GDB (32-bit), the MinGW.org runtime libraries and tools, and the windows-default-manifest package.

Below the download options, there are links for "The following links are for TDM-GCC releases", "TDM Distributed Package", "GCC, TDM32 or TDM64", and "GCC, MinGW.org".

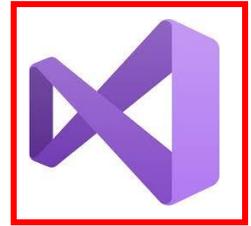
Télécharger cmake
Télécharger le compilateur gcc (pas celui de Microsoft !)
Et ensuite, indiquer à vscode de l'utiliser !
Et tout devrait aller tout seul.... (enfin, on espère !)

Avoir un compilateur sous Windows ?



**Télécharger le compilateur gcc (pas celui de Microsoft !)
Et ensuite, indiquer à vscode de l'utiliser !**

Avoir un compilateur sous Mac ?



Search results for « xcode »

Xcode
Outils de développement
★★★★☆ 304

ARTICLE
Xcode, ça sert à quoi déjà ?

GPT Code Creator for Xcode
Outils de développement

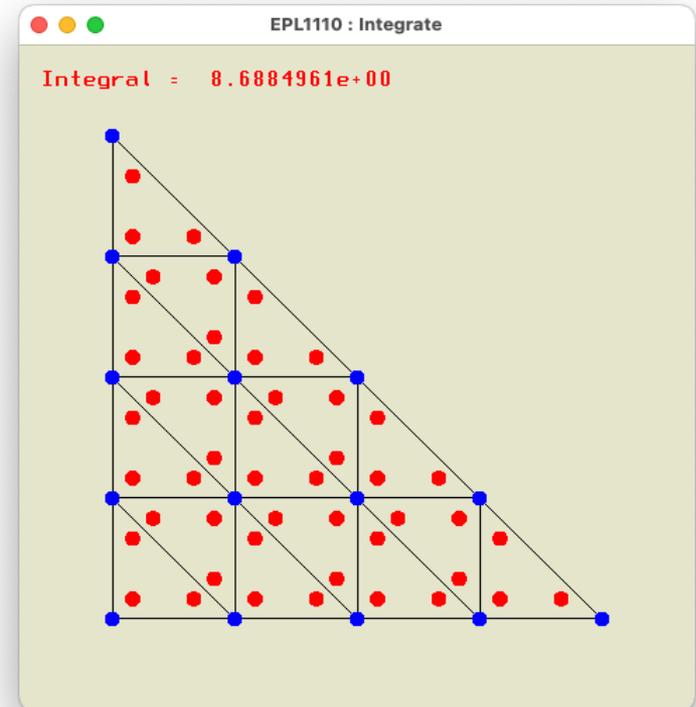
Templates for Swift
Outils de développement

Installer Xcode
Obtenir les commandes dans le terminal
Installer cmake
Et ensuite, vscode trouvera tout cela tout seul !

VL Vincent Legat

Homework 1

$$\underbrace{\int_{\hat{\Omega}} f(x, y) \, dx \, dy}_I \approx \underbrace{\sum_{k=1}^3 w_k f(x_k, y_k)}_{I_h}$$



Ecrire la règle de Hammer

mainBasic.c

```
#include <stdio.h>
#include <math.h>

double integrate(double x[3], double y[3], double(*f)(double,double));
double integrateRecursive(double x[3], double y[3], double(*f)(double,double), int n);

double fun(double x, double y) { return cos(x) + y * y; }
double stupid(double x, double y) { return 1.0; }

int main()
{
    double x[3] = {0, 1, 0};
    double y[3] = {0, 0, 1};
    int n;
    printf("Surface integration      : %14.7e \n", integrate(x,y,stupid));
    printf("More funny integration : %14.7e \n", integrate(x,y,fun));
    for (n=0; n <= 4; n++) {
        double I = integrateRecursive(x,y,fun,n);
        printf("Recursive integration (n = %2d) : %14.7e \n", n,I);
    }
    return 0;
}
```

homework.c et mainBasic.c

```
double integrate(double x[3], double y[3], double (*f) (double, double))
{
    double I = 3.14;
    return I;
}
```

```
double integrateRecursive(double x[3], double y[3], double (*f)(double,double), int n)
{
    double I = 0.0;
    return I;
}
```

```
double fun(double x, double y) { return cos(x) + y * y; }
double stupid(double x, double y) { return 1.0; }
```

```
int main()
{
    double x[3] = {0, 1, 0};
    double y[3] = {0, 0, 1};
    int n;
    printf("Surface integration      : %14.7e \n", integrate(x,y,stupid));
    printf("More funny integration : %14.7e \n", integrate(x,y,fun));
    for (n=0; n <= 1; n++) {
        double I = integrateRecursive(x,y,fun,n);
        printf("Recursive integration (n = %2d) : %14.7e \n", n,I); }
    return 0;
}
```

Comment compiler le devoir sur votre ordinateur ?



homework.c



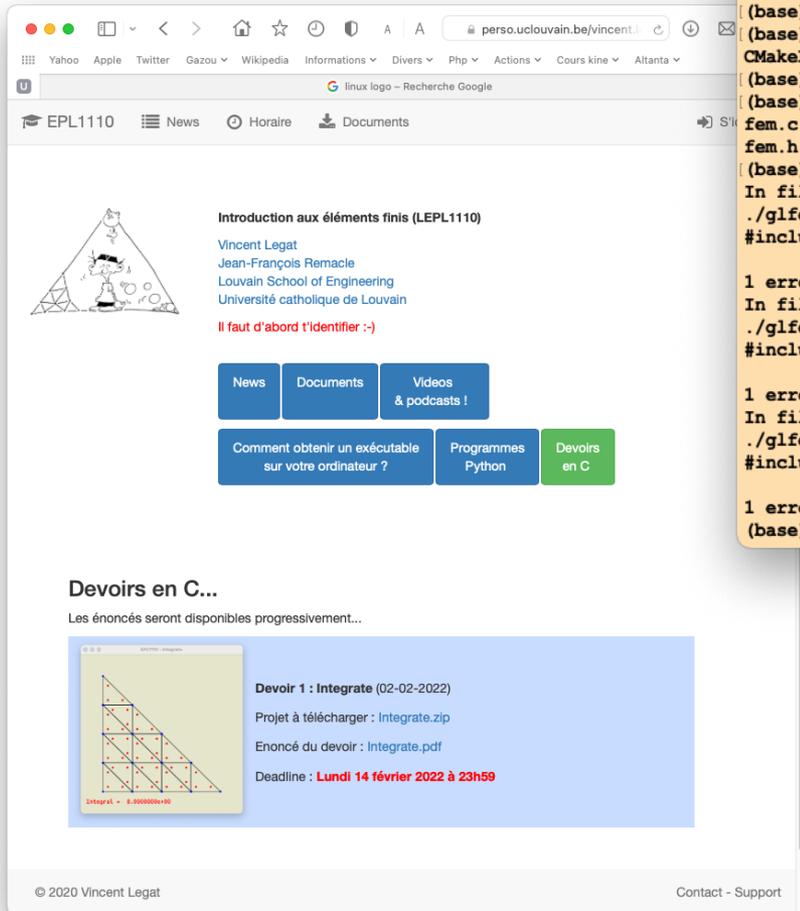
homeworkSoluce.c



mainBasic.c

```
Cours1 — -bash — 79x29
(base) mac-1U0-352:Cours1 v1$
(base) mac-1U0-352:Cours1 v1$ cc -o myFem mainBasic.c homework.c
(base) mac-1U0-352:Cours1 v1$ ./myFem
Surface integration      : 3.1400000e+00
More funny integration  : 3.1400000e+00
Recursive integration (n = 0) : 0.0000000e+00
Recursive integration (n = 1) : 0.0000000e+00
Recursive integration (n = 2) : 0.0000000e+00
Recursive integration (n = 3) : 0.0000000e+00
Recursive integration (n = 4) : 0.0000000e+00
(base) mac-1U0-352:Cours1 v1$ cc -o myFem mainBasic.c homeworkSoluce.c
(base) mac-1U0-352:Cours1 v1$ ./myFem
Surface integration      : 5.0000000e-01
More funny integration  : 5.4302895e-01
Recursive integration (n = 0) : 5.4302895e-01
Recursive integration (n = 1) : 2.1721229e+00
Recursive integration (n = 2) : 8.6884961e+00
Recursive integration (n = 3) : 3.4753986e+01
Recursive integration (n = 4) : 1.3901594e+02
(base) mac-1U0-352:Cours1 v1$
```

Bon : c'est aussi simple ?



perso.uclouvain.be/vincent.l...

Yahoo Apple Twitter Gazou v Wikipedia Informations v Divers v Php v Actions v Cours kine v Altanta v

linux logo - Recherche Google

EPL1110 News Horaire Documents

 Introduction aux éléments finis (LEPL1110)

Vincent Legat
Jean-François Remacle
Louvain School of Engineering
Université catholique de Louvain

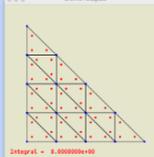
Il faut d'abord t'identifier :-)

News Documents Videos & podcasts !

Comment obtenir un exécutable sur votre ordinateur ? Programmes Python Devoirs en C

Devoirs en C...
Les énoncés seront disponibles progressivement...



 **Devoir 1 : Integrate (02-02-2022)**

Projet à télécharger : [Integrate.zip](#)

Enoncé du devoir : [Integrate.pdf](#)

Deadline : **Lundi 14 février 2022 à 23h59**

© 2020 Vincent Legat Contact - Support

```
src -- -bash -- 90x27
IntegrateWithoutBov      homework.c      myFem
hello                    homeworkSoluce.c  yep
hello.c                  mainBasic.c
(base) mac-1U0-352:Cours1 v1$ cd Integrate
(base) mac-1U0-352:Integrate v1$ ls
CMakeLists.txt  glfw      src
(base) mac-1U0-352:Integrate v1$ cd src
(base) mac-1U0-352:src v1$ ls
fem.c           glfem.c      homework.c
fem.h           glfem.h      main.c
(base) mac-1U0-352:src v1$ gcc -o myFem *.c
In file included from glfem.c:10:
./glfem.h:21:10: fatal error: 'GLFW/glfw3.h' file not found
#include <GLFW/glfw3.h>
               ^~~~~~
1 error generated.
In file included from homework.c:3:
./glfem.h:21:10: fatal error: 'GLFW/glfw3.h' file not found
#include <GLFW/glfw3.h>
               ^~~~~~
1 error generated.
In file included from main.c:5:
./glfem.h:21:10: fatal error: 'GLFW/glfw3.h' file not found
#include <GLFW/glfw3.h>
               ^~~~~~
1 error generated.
(base) mac-1U0-352:src v1$
```

Eh non !

perso.uclouvain.be/vincent.legat/zouLab/ep1110.php?action=

Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta Finite Elements Méthodes numériques

linux logo - Recherche Google

EPL1110 News Horaire Documents Mon profil Mon binome

Exécution et soumission d'un programme sur le serveur...

Groupe : 1 (Megat-jeremacle)
Binôme : Remacle, Jean-François
Deadline : February 14 2022 23:59:59.
Now : February 04 2022 09:04:25.

```
1  
2 #include <math.h>  
3  
4 #include "glfen.h"  
5  
6  
7  
8  
9  
10  
11
```

Position: Ln 11, Ch 1 Total: Ln 65, Ch 1656

Exécuter le programme sur le serveur Voir le diagnostic

Valider son programme

© 2020 Vincent Legat

perso.uclouvain.be/vincent.legat/zouLab/ep1110.php?...

Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta Finite Elements Méthodes numériques

linux logo - Recherche Google

EPL1110 News Horaire Documents Mon profil Mon binome Hello Vincent Deconnexion



Introduction aux éléments finis (LEPL110)

Vincent Legat
Jean-François Remacle
Louvain School of Engineering
Université catholique de Louvain

News Documents Videos & podcasts !

Comment obtenir un exécutable sur votre ordinateur ? Programmes Python Devoirs en C

Liste des étudiants Liste des groupes Equipe didactique Former son groupe

Soumettre le devoir 1 *Integrate*

© 2020 Vincent Legat Contact - Support

Soumettre votre devoir !

**Tous les devoirs seront corrigés de manière automatique !
Bien veiller à ce que la version soumise soit bien compilée !
Aucune soumission tardive ne sera admise !
Les devoirs sont réalisés par groupe !**

perso.uclouvain.be/vincent.legat/zouLab/epi

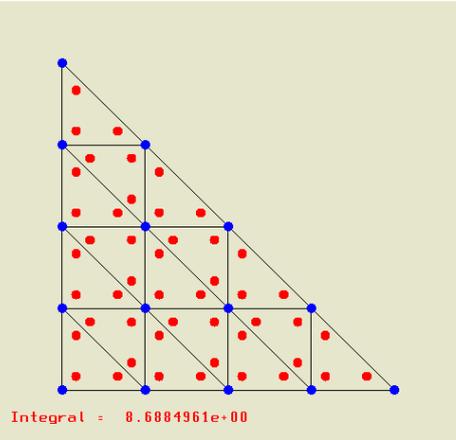
Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta Finite Elements

linux logo - Recherche Google

EPL1110 News Horaire Documents Mon profil Mon binome

Exécution et soumission d'un programme sur le serveur...

Groupe : 1 (legat-jeremacle)
Binôme : Remacle, Jean-François
Deadline : February 14 2022 23:59:59.
Now : February 04 2022 09:06:11.



Integral = 8.6884961e+00

© 2020 Vincent Legat

perso.uclouvain.be/vincent.legat/zouLab/epi

Yahoo Apple Twitter Gazou Wikipedia Informations Divers Php Actions Cours kine Altanta Finite Elements

linux logo - Recherche Google

EPL1110 News Horaire Documents Mon profil Mon binome Hello Vincent Deconnexion

```
int i,j;
const int nodes[4][3] = {{0,3,5},{3,1,4},{5,4,2},{3,4,5}};
const double xsi[6] = {0.0,1.0,0.0,0.5,0.5,0.0};
const double eta[6] = {0.0,0.0,1.0,0.0,0.5,0.5};
double xLoc[3];
double yLoc[3];

if (n <= 0) return integrate(x,y,f);

double I = 0.0;
for (i=0; i<4; i++) {
  for (j=0; j<3; j++) {
    double xsiLoc = xsi[nodes[i][j]];
    double etaLoc = eta[nodes[i][j]];
    xLoc[j] = interpolate(x,xsiLoc,etaLoc);
    yLoc[j] = interpolate(y,xsiLoc,etaLoc);
    I += integrateRecursive(xLoc,yLoc,f,n-1);
  }
}

return I;
}
```

Faire une nouvelle soumission Voir le diagnostic

Valider son programme

© 2020 Vincent Legat Contact - Support

Valider
et verifier son devoir !

homework.c

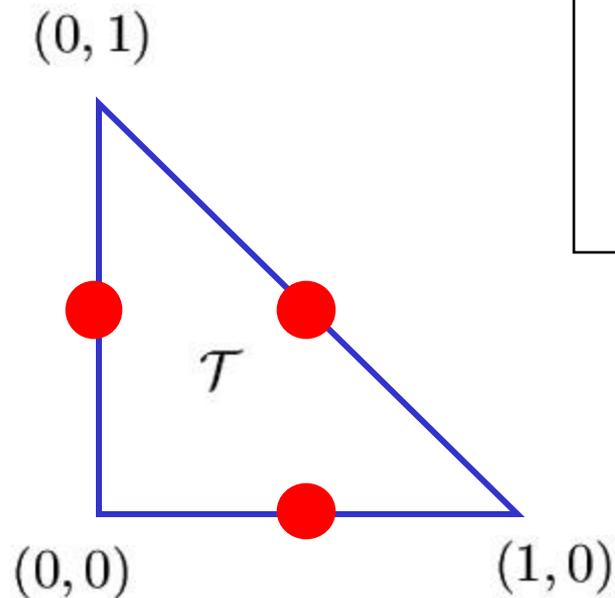
La solution doit se trouver dans le fichier homework.c uniquement.... On ne regarde jamais main.c !

```
#include <stdio.h>
#include <math.h>

double integrate(double x[3], double y[3], double (*f) (double, double))
{
    double I = 3.14;
    int I;
    for (i=0; i<3; i++)
        printf("    === node %d : %14.7e %14.7e \n",i+1,x[i],y[i]);
    return I;
}

double integrateRecursive(double x[3], double y[3], double (*f) (double,double), int n)
{
    double I = 0.0;
    return I;
}
```

Intégration sur un triangle : Règle de Hammer à 3 points



$$\underbrace{\int_{\mathcal{T}} f(x, y) \, dx \, dy}_{I} \approx \underbrace{\sum_{k=1}^3 w_k f(X_k, Y_k)}_{I^h}$$

	X_k	Y_k	w_k
1	0.5	0.0	1/6
2	0.5	0.5	1/6
3	0.0	0.5	1/6

Démontrer que la formule de Hammer à trois points permet d'intégrer exactement n'importe quel polynôme à deux variables de degré deux : $a + bx + cy + dx^2 + ey^2 + fxy$

Question

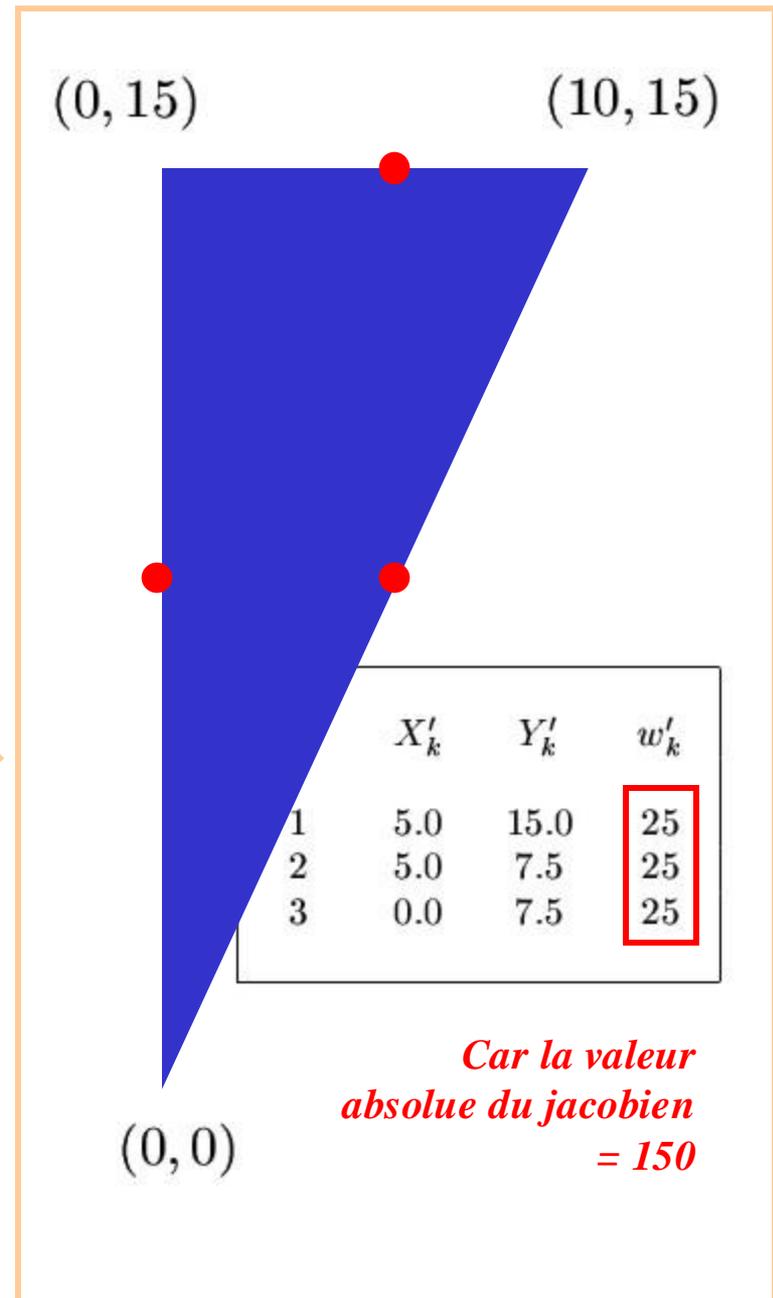
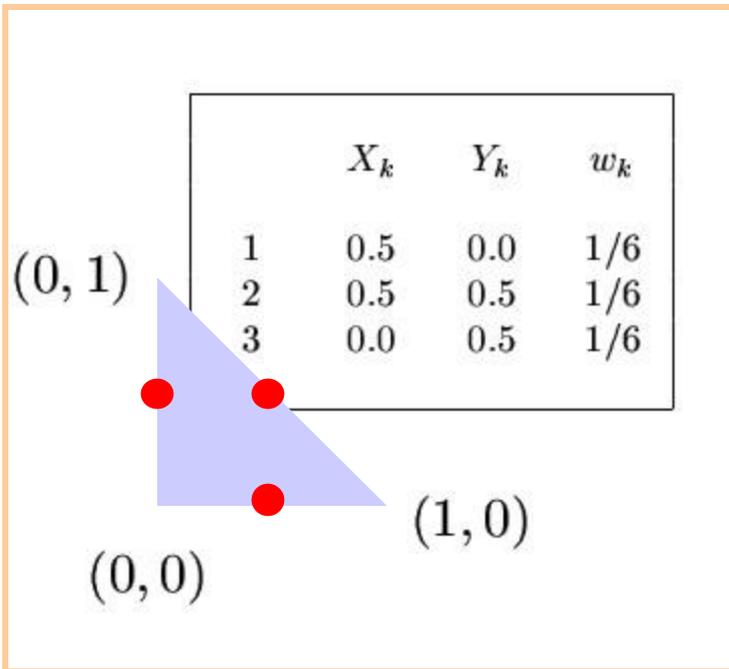
$$\begin{aligned}
 I &= \frac{a}{2} + b \int_0^1 x \int_0^{1-x} dy \, dx + c \int_0^1 y \int_0^{1-y} dx \, dy \\
 &\quad + d \int_0^1 x^2 \int_0^{1-x} dy \, dx + e \int_0^1 y^2 \int_0^{1-y} dx \, dy + f \int_0^1 x \int_0^{1-x} y \, dy \, dx \\
 &= \frac{a}{2} + b \left[\frac{x^2}{2} - \frac{x^3}{3} \right]_0^1 + c \left[\frac{y^2}{2} - \frac{y^3}{3} \right]_0^1 \\
 &\quad + d \left[\frac{x^3}{3} - \frac{x^4}{4} \right]_0^1 + e \left[\frac{y^3}{3} - \frac{y^4}{4} \right]_0^1 + f \left[\frac{x^2}{4} - \frac{x^3}{3} + \frac{x^4}{8} \right]_0^1 \\
 &= \frac{a}{2} + \frac{b}{6} + \frac{c}{6} + \frac{d}{12} + \frac{e}{12} + \frac{f}{24} \\
 &= I^h
 \end{aligned}$$



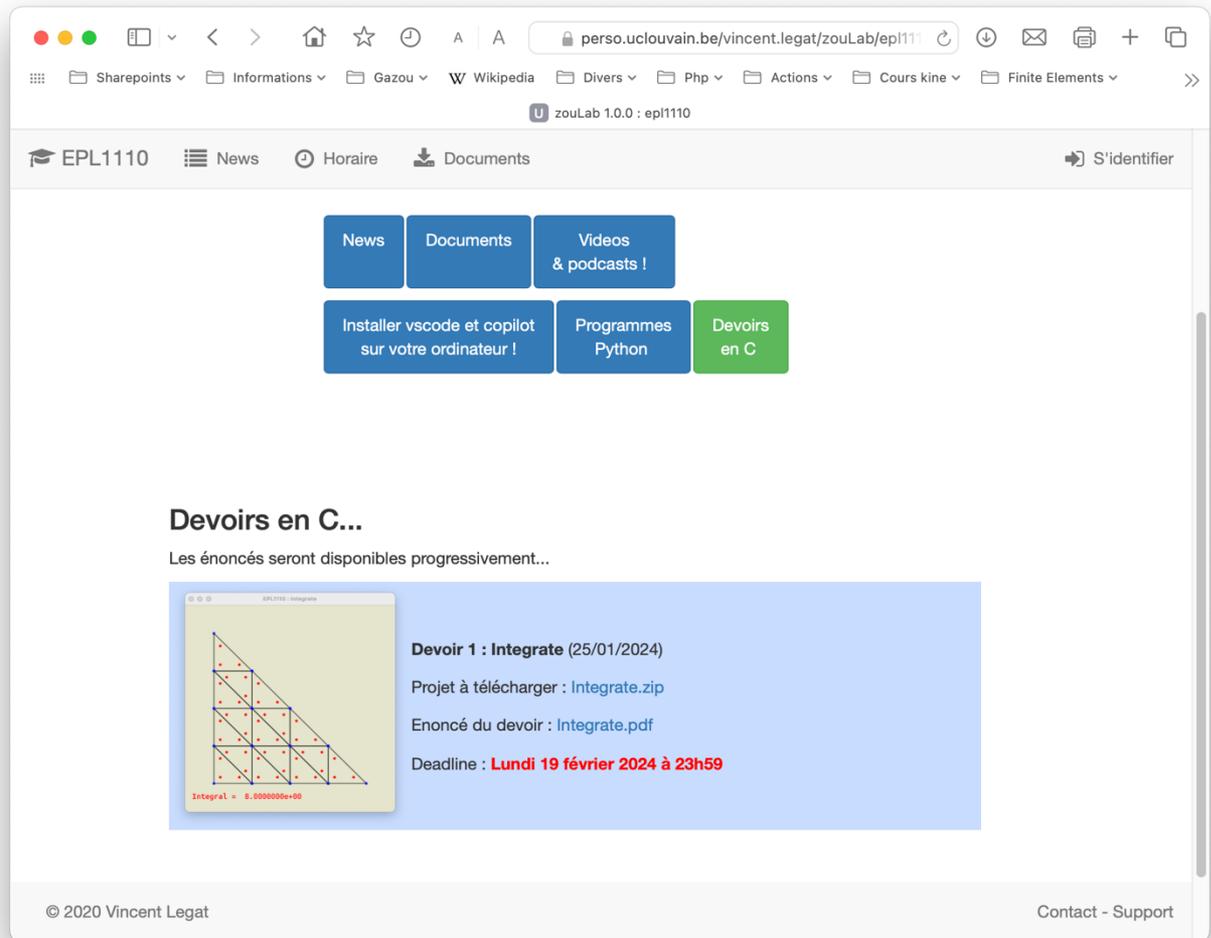
**Degré de
précision**

Et un autre triangle ?

$$\begin{aligned}x' &= 10x \\ y' &= 15 - 15y\end{aligned}$$



Effectuons cela ensemble !



perso.uclouvain.be/vincent.legat/zouLab/epl1110

Sharepoints Informations Gazou Wikipedia Divers Php Actions Cours kine Finite Elements

zouLab 1.0.0 : epl1110

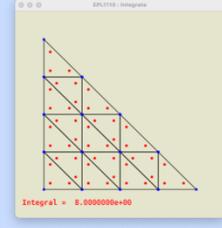
EPL1110 News Horaire Documents S'identifier

News Documents Videos & podcasts !

Installer vscode et copilot sur votre ordinateur ! Programmes Python Devoirs en C

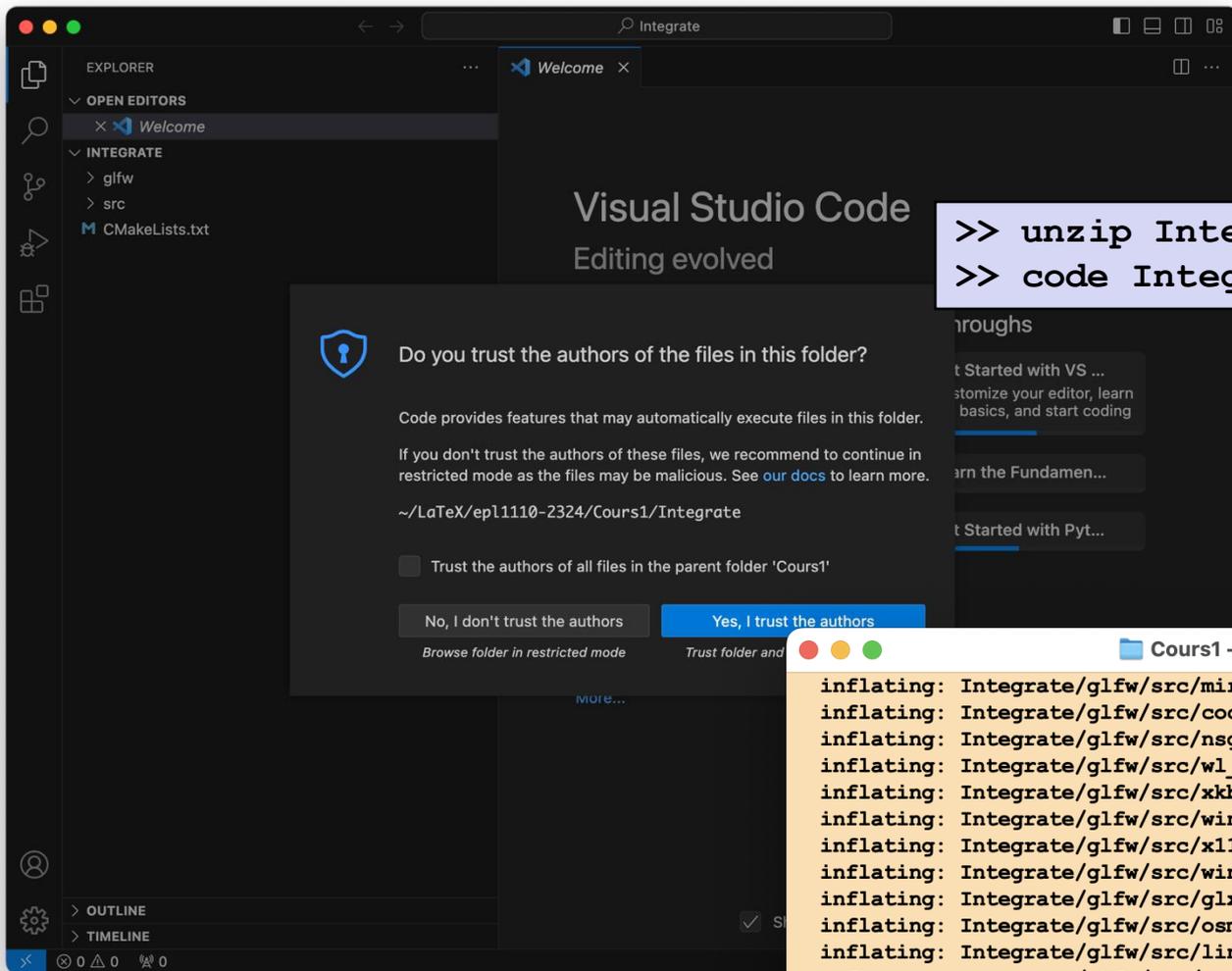
Devoirs en C...

Les énoncés seront disponibles progressivement...



Devoir 1 : Integrate (25/01/2024)
Projet à télécharger : [Integrate.zip](#)
Enoncé du devoir : [Integrate.pdf](#)
Deadline : **Lundi 19 février 2024 à 23h59**

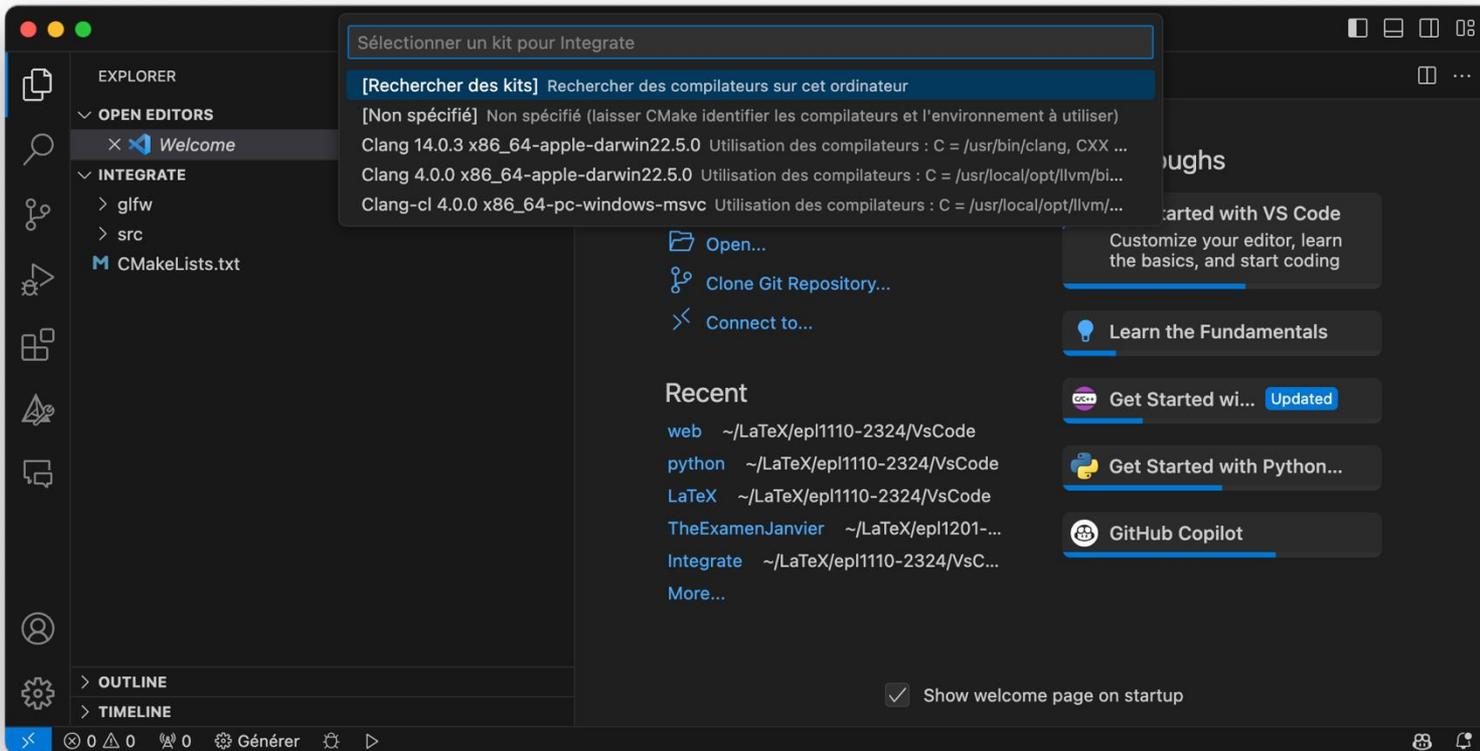
© 2020 Vincent Legat Contact - Support



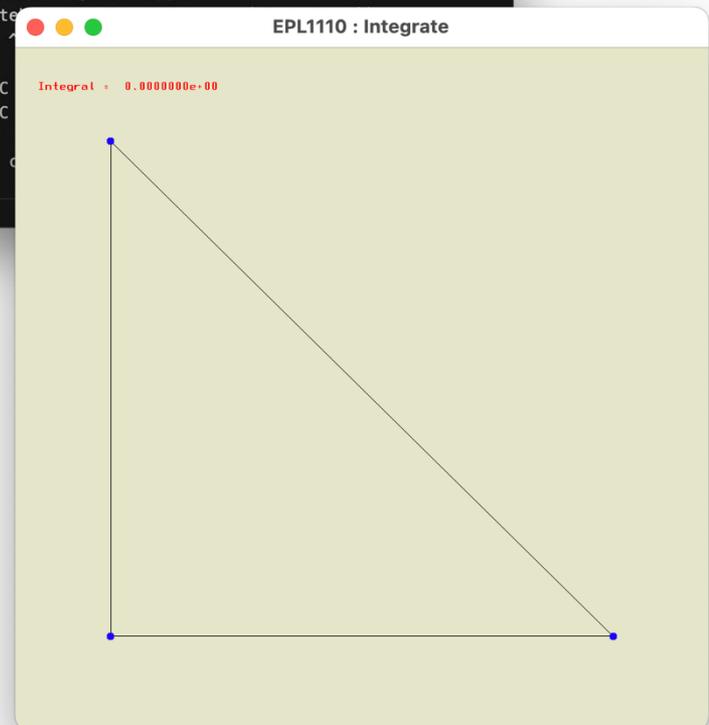
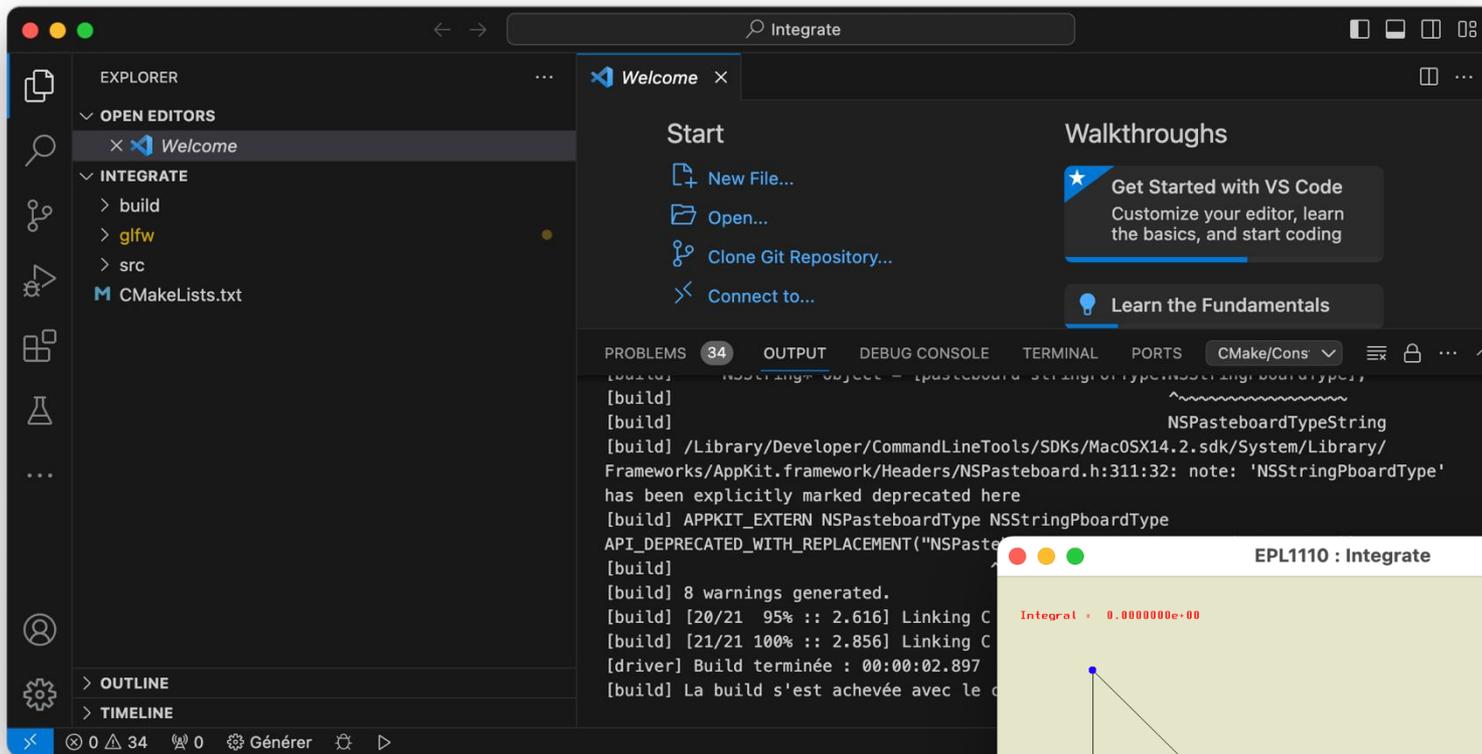
```
>> unzip Integrate.zip  
>> code Integrate
```

```
Cours1 -- -bash -- 72x22  
inflating: Integrate/glfw/src/mir_platform.h  
inflating: Integrate/glfw/src/cocoa_time.c  
inflating: Integrate/glfw/src/nsgl_context.m  
inflating: Integrate/glfw/src/wl_platform.h  
inflating: Integrate/glfw/src/xkb_unicode.c  
inflating: Integrate/glfw/src/win32_window.c  
inflating: Integrate/glfw/src/x11_monitor.c  
inflating: Integrate/glfw/src/win32_platform.h  
inflating: Integrate/glfw/src/glx_context.c  
inflating: Integrate/glfw/src/osmesa_context.h  
inflating: Integrate/glfw/src/linux_joytick.c  
inflating: Integrate/glfw/src/cocoa_monitor.m  
inflating: Integrate/glfw/src/window.c  
inflating: Integrate/src/fem.h  
inflating: Integrate/src/.DS_Store  
inflating: Integrate/src/glfem.h  
inflating: Integrate/src/fem.c  
inflating: Integrate/src/main.c  
inflating: Integrate/src/homework.c  
inflating: Integrate/src/glfem.c  
(base) mac-1U0-352:Cours1 v1$ code Integrate  
(base) mac-1U0-352:Cours1 v1$
```

code integrate



Sélectionner le compilateur !
Lancer la compilation et l'exécution !

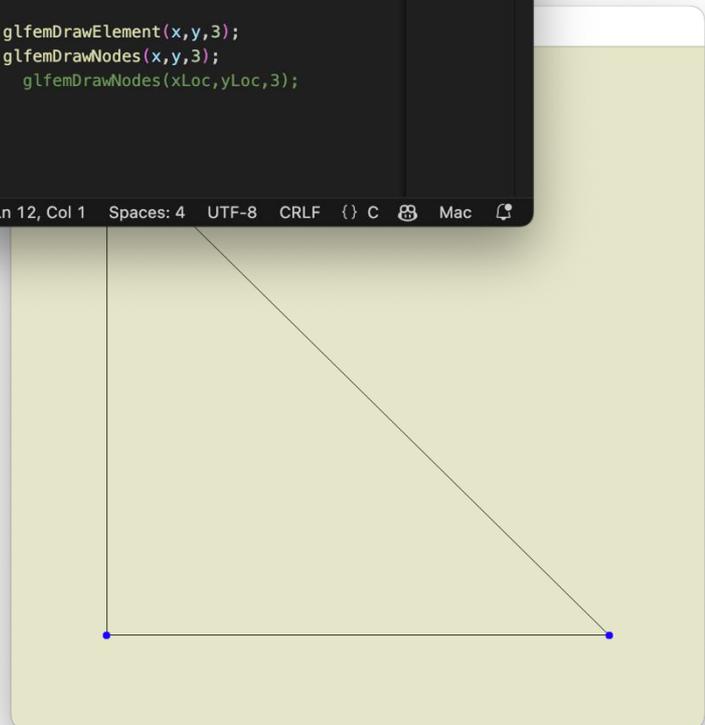


Et zou !

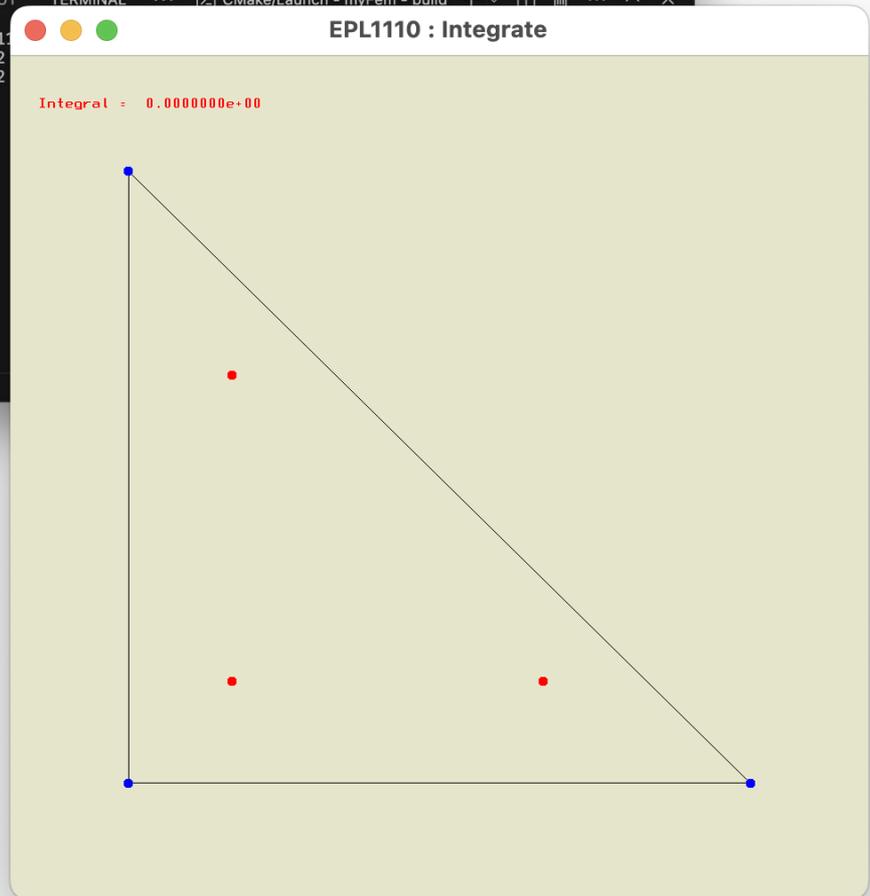
The screenshot shows a code editor window titled "Integrate" with a file named "homework.c". The code defines a function `integrate` that takes three double arrays `x`, `y`, and a pointer to a function `f`. It calculates the integral of `f(x, y)` over a triangular domain. The domain is defined by the vertices `(0,0)`, `(1,0)`, and `(0,1)`. The function `f` is a constant function `w` with value `1.0/6.0`. The code uses `glfw` for drawing, showing a triangle and its nodes.

```
src > C homework.c > integrate(double [3], double [3], double (*)(double, double))
5
6 double integrate(double x[3], double y[3], double (*f) (double, double))
7
8     double I = 0;
9     double xLoc[3];
10    double yLoc[3];
11
12    double w[3] = {1.0/6.0, 1.0/6.0, 1.0/6.0};
13
14    //
15    // ... A modifier :-)
16    //
17    //
18    // Pour dessiner l'element, les sommets du triangle :-)
19    // Decommenter la ligne pour dessiner aussi les points d'integration
20    //
21
22    glfwSetColor(GLFEM_BLACK); glfwDrawElement(x,y,3);
23    glfwSetColor(GLFEM_BLUE);  glfwDrawNodes(x,y,3);
24    // glfwSetColor(GLFEM_RED);  glfwDrawNodes(xLoc,yLoc,3);
25
26
27
28    return I;
```

Et l'intelligence artificielle
va écrire le devoir...

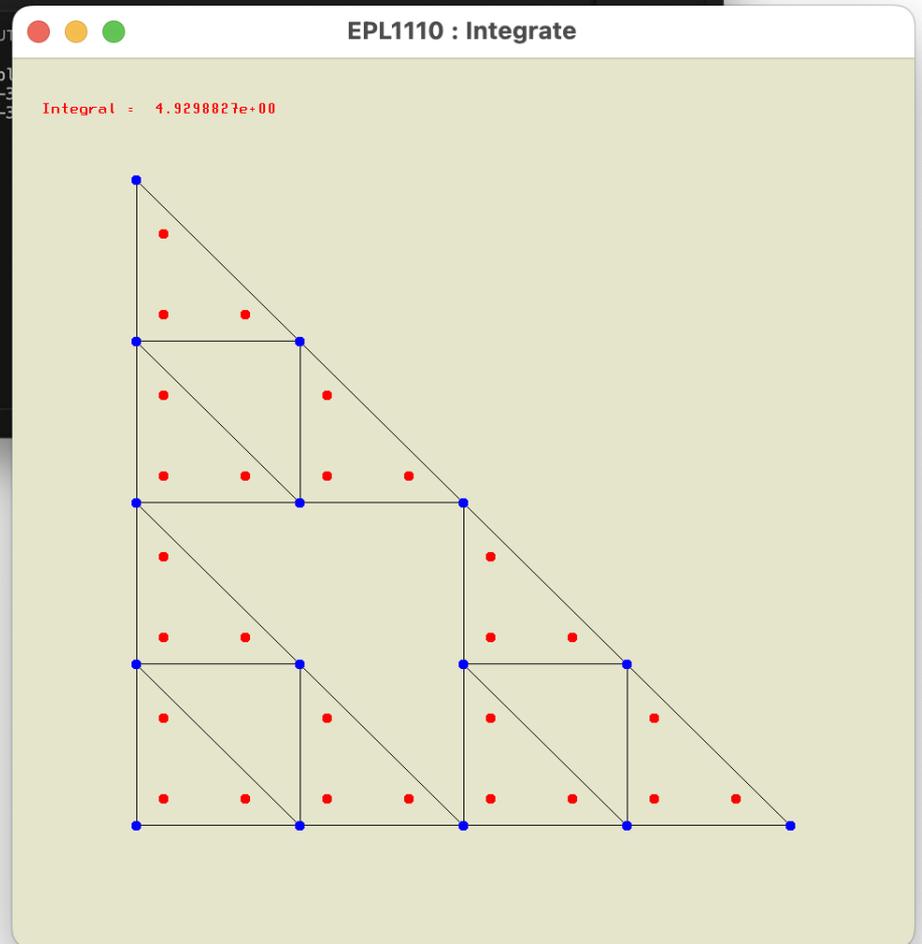


```
src > C homework.c > integrate(double [3], double [3], double*)(double, double)
15 xLoc[0] = x[0] * (1 - xsi[0] - eta[0]) + x[1] * xsi[0] + x[2] * eta
16 xLoc[1] = x[0] * (1 - xsi[1] - eta[1]) + x[1] * xsi[1] + x[2] * eta
17 xLoc[2] = x[0] * (1 - xsi[2] - eta[2]) + x[1] * xsi[2] + x[2] * eta
18 yLoc[0] = y[0] * (1 - xsi[0] - eta[0]) + y[1] * xsi[0] + y[2] * eta
19 yLoc[1] = y[0] * (1 - xsi[1] - eta[1]) + y[1] * xsi[1] + y[2] * eta
20 yLoc[2] = y[0] * (1 - xsi[2] - eta[2]) + y[1] * xsi[2] + y[2] * eta
21
22 //
```



Et copilot
a fait le job !

```
src > C homework.c x integrate(double [3], double [3], double*)(double, double)
15 xLoc[0] = x[0] * (1 - xsi[0] - eta[0]) + x[1] * xsi[0] + x[2] * eta
16 xLoc[1] = x[0] * (1 - xsi[1] - eta[1]) + x[1] * xsi[1] + x[2] * eta
17 xLoc[2] = x[0] * (1 - xsi[2] - eta[2]) + x[1] * xsi[2] + x[2] * eta
18 yLoc[0] = y[0] * (1 - xsi[0] - eta[0]) + y[1] * xsi[0] + y[2] * eta
19 yLoc[1] = y[0] * (1 - xsi[1] - eta[1]) + y[1] * xsi[1] + y[2] * eta
20 yLoc[2] = y[0] * (1 - xsi[2] - eta[2]) + y[1] * xsi[2] + y[2] * eta
21
22 //
```



Enfin
plus ou moins....

CHAT: GITHUB COPILOT

Remember, these are just a few examples. The application of FEM is vast and spans across many other fields as well.

vlegat

Write a small abstract about finite elements

GitHub Copilot

Sure, here is a small abstract about the finite element method:

The Finite Element Method (FEM) is a powerful numerical technique used in a wide range of engineering disciplines. It is known for its flexibility in handling complex geometries and boundary conditions. FEM is extensively used in various fields such as structural engineering, mechanical engineering, electrical engineering, aerospace engineering, civil engineering, biomedical engineering, thermal engineering, and fluid dynamics. It enables engineers to predict the behavior of structures, components, and systems under different loads and conditions. By dividing the domain into smaller elements, FEM approximates the solution to complex problems by solving a system of equations. This method has revolutionized engineering analysis and design, allowing engineers to optimize designs, improve performance, and ensure the safety and reliability of engineering systems.

```

%x toto.tex
0 \begin{document}
7 \begin{tikzpicture}
8 \draw (0,0) circle (1);
9 \draw (-0.5,0.5) circle (0.3);
10 \draw (0.5,0.5) circle (0.3);
11 \draw (0,0) ellipse (1.2 and 0.6);
12 \end{tikzpicture}
13
14
15
16
17 % Your text describing the finite elements method goes
18
19 The Finite Element Method (FEM) is a powerful numerical
20
21
22 \begin{itemize}
23 \item Structural Engineering
24
25 \item Mechanical Engineering
26
27 \item Electrical Engineering

```

PROBLEMS OUTPUT DEBUG CONSOLE

[Loading MPS to PDF converter (version 0.9.60 Copyright Copyright (c) 2005-2012 TeX Systems, Inc.) (/usr/local/texlive/2022/texmf-dist/fonts/enc/dvips/cm-super/local/texlive/2022/texmf-dist/fonts/enc/dvips/cm-super/epstopdf-sys.cfg) [1{/usr/local/texlive/2022/texmf-dist/fonts/enc/dvips/cm-super/epstopdf-sys.cfg}] Output written on toto.pdf (2 pages). SyncTeX written on toto.synctex.gz. Transcript written on toto.log. Latexmk: Getting log file 'toto.log'. Latexmk: Examining 'toto.fls'. Latexmk: Examining 'toto.log'. Latexmk: Log file says output to 'toto.pdf'. Latexmk: All targets (toto.pdf) are up-to-date.

Et même
le rapport en LaTeX !

Présentation Page 1 sur 2

Inspecteur Réduire/agrandir Partager Surbrillance Pivoter Annoter Remplissage de formulaire Recherche

The Finite Element Method (FEM) is a powerful numerical technique used in a wide range of engineering disciplines. It is known for its flexibility in handling complex geometries and boundary conditions. FEM is extensively used in various fields such as structural engineering, mechanical engineering, electrical engineering, aerospace engineering, civil engineering, biomedical engineering, thermal engineering, and fluid dynamics. It enables engineers to predict the behavior of structures, components, and systems under different loads and conditions. By dividing the domain into smaller elements, FEM approximates the solution to complex problems by solving a system of equations. This method has revolutionized engineering analysis and design, allowing engineers to optimize designs, improve performance, and ensure the safety and reliability of engineering systems.

- **Structural Engineering:** FEM is used to predict the behavior of buildings, bridges, towers, and other structures under various loads to ensure they can safely withstand the forces they will be subjected to.
- **Mechanical Engineering:** It's used in the design and analysis of mechanical components and systems, such as predicting the stress and strain in mechanical parts under load.
- **Electrical Engineering:** FEM is used in the design and analysis of electrical and electronic components, such as antennas, waveguides, and microelectronic devices.
- **Aerospace Engineering:** It's used to simulate and analyze the behavior of aircraft and spacecraft structures under various conditions.
- **Civil Engineering:** FEM is used in the analysis and design of civil infrastructure like dams, tunnels, and roads.
- **Biomedical Engineering:** It's used to model biological systems, such as the flow of blood in arteries, the mechanical behavior of bones and tissues, etc.

The image shows a VS Code editor window on the left with a file explorer on the left sidebar. The main editor area displays an HTML file named 'ep1110-vscode.html'. The code includes a DOCTYPE declaration, a meta charset of 'utf-8', a viewport meta tag, and links to Bootstrap CSS and jQuery. A preview window on the right shows the rendered HTML content, which is a page titled 'Compilation et exécution d'un programme'. The browser address bar shows the file path: 'file:///Users/vj/LaTeX/ep1110-2324/VsCode/...'. The browser tabs include 'Sharepoints', 'Informations', 'Gazou', 'Wikipedia', 'Divers', 'Php', 'Actions', 'Cours kine', and 'Finite Elements'. The page content in the browser matches the text in the VS Code editor.

Et même
ZouLab !

Compilation et exécution d'un programme

Pour construire votre projet d'éléments finis, nous allons utiliser un certain nombre d'outils. Cette année, nous ferons appel à l'intelligence artificielle pour vous aider à écrire votre code. Pour cela, nous allons utiliser un éditeur de code, Vscode, et un plugin, Copilot, qui est une intelligence artificielle développée par OpenAI. Nous allons également utiliser un gestionnaire de version, Git, et un dépôt de code, GitHub. Enfin, nous allons utiliser un compilateur, GCC, pour compiler notre code. Nous allons également utiliser un outil de gestion de projet, CMake, pour générer les fichiers de configuration de notre projet. Evidemment, tout cela peut paraître compliqué, mais nous allons vous guider pas à pas pour vous aider à maîtriser ces outils. Et même pour écrire cette page web, l'intelligence artificielle de Copilot m'a aidé à écrire ce texte. C'est assez magique en fait !

Installation de VsCode

Attention : **nous utiliserons VsCode (Visual Studio Code) qui est un logiciel distinct de Visual Studio**. Visual Studio est un environnement de développement intégré (IDE Integrated Development Environment), tandis que VsCode est un éditeur de code léger et polyvalent.

- Sur le site (bien noter le mot `code` dans l'URL) :

<https://code.visualstudio.com/Download>



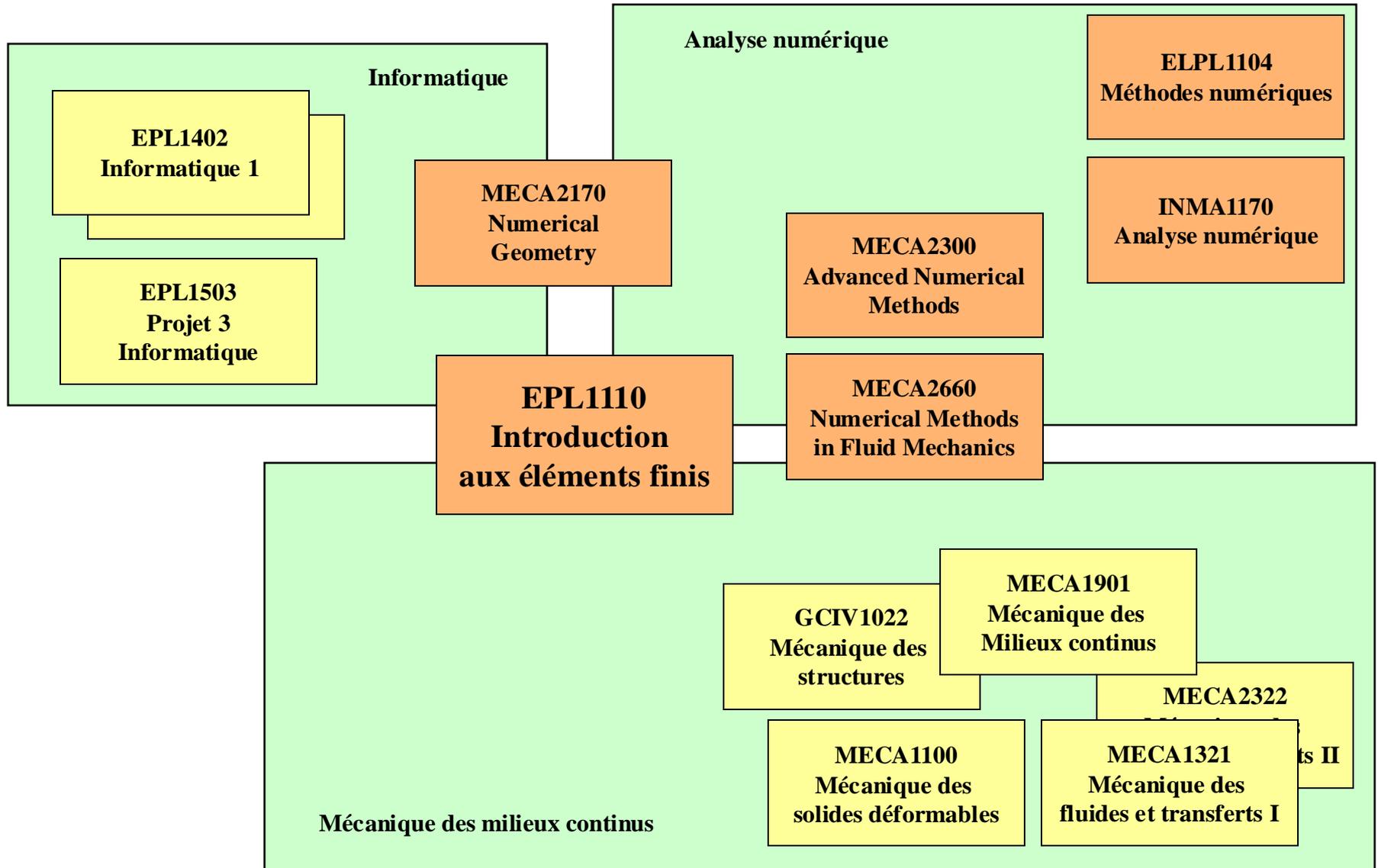
A l'issue de ce cours, vous serez capables de...

- Comprendre la méthode des éléments finis
- Réaliser un petit programme en C
- Certifier et valider une simulation
- Choisir la voie numérique la plus efficace
- Estimer la précision d'un résultat
- Découvrir les joies et les aléas du numérique
- Découvrir les joies de l'intelligence artificielle

Non, non : ceci on ne fera pas !

- Apprendre le génie logiciel de l'orienté-objet
- Utiliser des logiciels commerciaux
- Faire de l'analyse numérique théorique
- Faire du calcul parallèle
- Résoudre les équations de Navier-Stokes
- Créer automatiquement des maillages

Et les autres cours....



Elasticité linéaire pourquoi ?

Centrale nucléaire de Tihange 1

Rapport de synthèse
Long Term Operation 2015-2025
Version finale



A vous de choisir
votre problème précis
à résoudre avec vos coaches !

The image displays a web application interface for the course "Introduction aux éléments finis (LEPL1110)". The browser address bar shows the URL: `perso.uclouvain.be/vincent.legat/zouLab/ep1110.php?action=loginCheck`. The navigation menu includes: EPL1110, News, Horaire, Documents, Mon profil, My nice coach, **My good coach**, Mon binome, Hello Vincent, and Deconnexion. The main content area is titled "Introduction aux éléments finis (LEPL1110)". A modal window titled "My good coach :-)" is open, showing contact information for "Leyskens, Thomas" (thomas.leyskens@uclouvain.be) and a Mickey Mouse avatar. The background shows course materials and navigation buttons like "Videos & podcasts!", "Programmes", and "Devoirs". The footer includes "© 2020 Vincent Legat" and "Contact - Support".

Objectifs du projet

Réaliser	Créer une application pour prédire un tsunami.
Certifier	Tester et valider le travail de votre groupe.
Expliquer	Expliquer de manière efficace et rapide à l'enseignant et aux autres étudiants ce que vous avez réalisé.
Comprendre	Comprendre ce que vous avez réalisé. Comprendre ce que d'autres groupes ont réalisé.

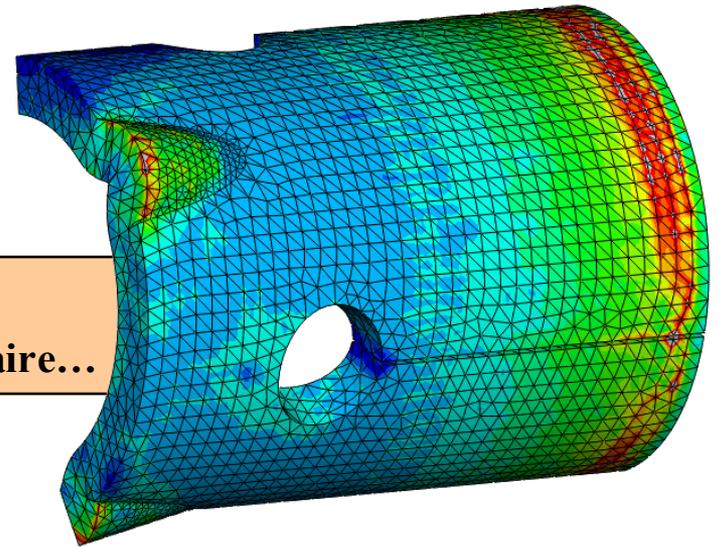
Exercices : 8 petits problèmes

Quelques petits problèmes
élémentaires pour apprivoiser le C

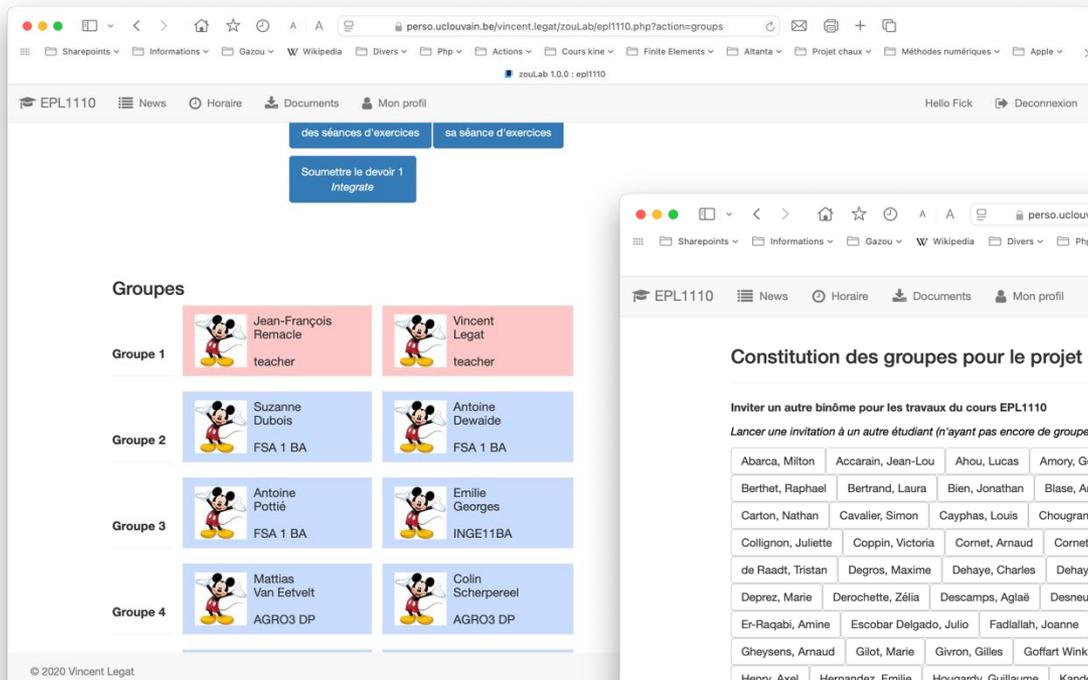
Projet en C :

Une petite application efficace pour l'élasticité linéaire...

The Practice of Programming :
Simplicity, Clarity, Generality. (B.K Kernighan & R. Pike 99)

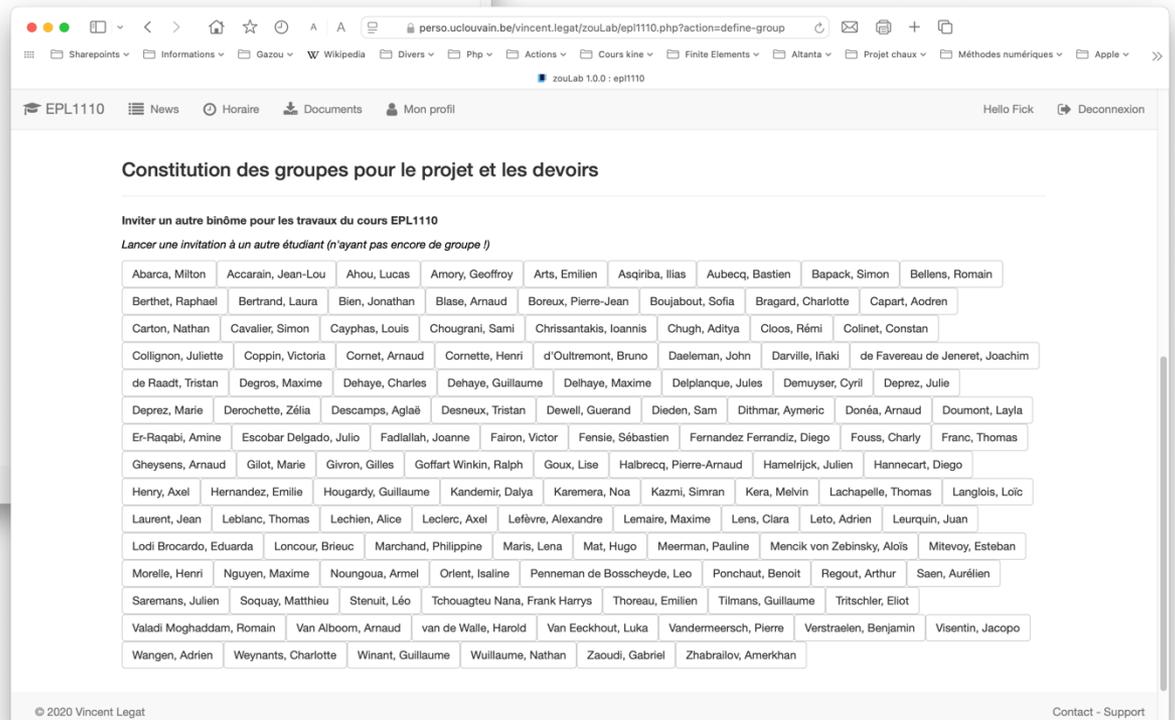


Former un groupe de deux étudiants (pas trois) !



The screenshot shows the course page for EPL1110. At the top, there are navigation links for 'des séances d'exercices' and 'sa séance d'exercices', and a button for 'Soumettre le devoir 1 Integrate'. Below this, a section titled 'Groupes' displays four groups, each with a Mickey Mouse icon and the names of the members and their course level.

Group	Members	Course Level
Groupe 1	Jean-François Remacle, Vincent Legat	teacher
Groupe 2	Suzanne Dubois, Antoine Dewaide	FSA 1 BA
Groupe 3	Antoine Pottié, Emilie Georges	FSA 1 BA, INGE11BA
Groupe 4	Mattias Van Eetvelt, Colin Scherpereel	AGRO3 DP

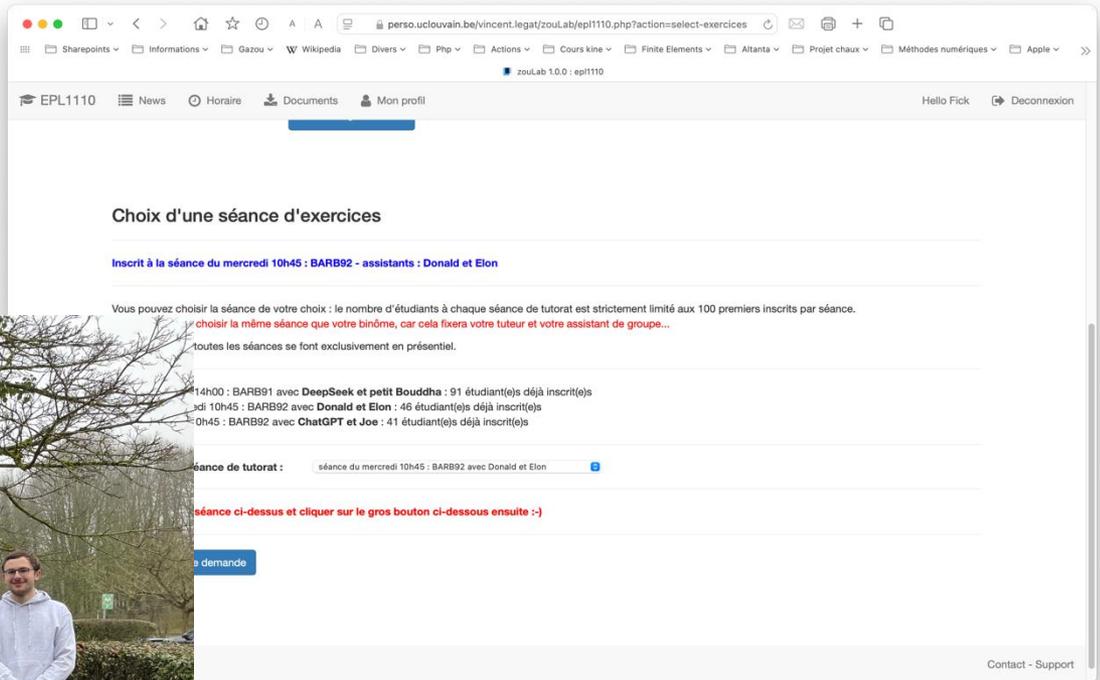


The screenshot shows the 'Constitution des groupes pour le projet et les devoirs' page. It includes a heading 'Inviter un autre binôme pour les travaux du cours EPL1110' and a sub-heading 'Lancer une invitation à un autre étudiant (n'ayant pas encore de groupe !)'. Below this is a large grid of student names for selection.

Inviter un autre binôme pour les travaux du cours EPL1110									
Lancer une invitation à un autre étudiant (n'ayant pas encore de groupe !)									
Abarca, Milton	Accarain, Jean-Lou	Ahou, Lucas	Amory, Geoffroy	Arts, Emilien	Asqirba, Ilias	Aubecq, Bastien	Bapack, Simon	Bellens, Romain	
Berthet, Raphael	Bertrand, Laura	Bien, Jonathan	Blase, Arnaud	Boreux, Pierre-Jean	Boujabout, Sofia	Bragard, Charlotte	Capart, Aodren		
Carton, Nathan	Cavalier, Simon	Casyphas, Louis	Chougrani, Sami	Chrissantakis, Ioannis	Chugh, Aditya	Cloos, Rémi	Colinet, Constan		
Collignon, Juliette	Coppin, Victoria	Cornet, Arnaud	Cornette, Henri	d'Outremont, Bruno	Daeleman, John	Darville, Ifakhi	de Favereau de Jeneret, Joachim		
de Raadt, Tristan	Degros, Maxime	Dehaye, Charles	Dehaye, Guillaume	Dehaye, Maxime	Delplanque, Jules	Demuyser, Cyril	Deprez, Julie		
Deprez, Marie	Derochette, Zélia	Descamps, Aglaé	Desneux, Tristan	Dewell, Guerland	Dieden, Sam	Dithmar, Aymeric	Donéa, Arnaud	Doumont, Layla	
Er-Raqabi, Amine	Escobar Dalgado, Julio	Fadlallah, Joanne	Fairon, Victor	Fensie, Sébastien	Fernandez Ferrandiz, Diego	Fouss, Charly	Franc, Thomas		
Gheysens, Arnaud	Gilot, Marie	Givron, Gilles	Goffart Winkin, Ralph	Goux, Lise	Halbreccq, Pierre-Arnaud	Hamelrijck, Julien	Hannecart, Diego		
Henry, Axel	Hernandez, Emilie	Hougaard, Guillaume	Kandemir, Dalya	Karamera, Noa	Kazmi, Simran	Kera, Melvin	Lachapelle, Thomas	Langlois, Loïc	
Laurent, Jean	Leblanc, Thomas	Lechien, Alice	Leclerc, Axel	Lefèvre, Alexandre	Lemaire, Maxime	Lens, Clara	Leto, Adrien	Leurquin, Juan	
Lodi Brocardo, Eduarda	Loncour, Brieuc	Marchand, Philippine	Maris, Lena	Mat, Hugo	Meerman, Pauline	Mencik von Zebinsky, Alois	Mitevoy, Esteban		
Morelle, Henri	Nguyen, Maxime	Noungoua, Armel	Orient, Isaline	Penneman de Bosscheyde, Leo	Ponchaut, Benoit	Regout, Arthur	Saen, Aurélien		
Saremans, Julien	Soquay, Matthieu	Stenuit, Léo	Tchouagteu Nana, Frank Harrys	Thoreau, Emilien	Tilmans, Guillaume	Tritschler, Eliot			
Valadi Moghaddam, Romain	Van Alboom, Arnaud	van de Walle, Harold	Van Eeckhout, Luka	Vandermeersch, Pierre	Verstraelen, Benjamin	Visentin, Jacopo			
Wangen, Adrien	Weynants, Charlotte	Winant, Guillaume	Wuillaume, Nathan	Zaoudi, Gabriel	Zhabrailov, Amerkhan				

Avoir une séance !

Avoir un tuteur et un assistant !



The screenshot shows a web browser window with the URL `perso.uclouvain.be/vincent.legat/zouLab/epi1110.php?action=select-exercices`. The page title is "Choix d'une séance d'exercices". The user is logged in as "Hello Fick".

Inscrit à la séance du mercredi 10h45 : BARB92 - assistants : Donald et Elon

Vous pouvez choisir la séance de votre choix : le nombre d'étudiants à chaque séance de tutorat est strictement limité aux 100 premiers inscrits par séance.

choisir la même séance que votre binôme, car cela fixera votre tuteur et votre assistant de groupe...

toutes les séances se font exclusivement en présentiel.

- 14h00 : BARB91 avec **DeepSeek et petit Bouddha** : 91 étudiant(e)s déjà inscrit(e)s
- 10h45 : BARB92 avec **Donald et Elon** : 46 étudiant(e)s déjà inscrit(e)s
- 0h45 : BARB92 avec **ChatGPT et Joe** : 41 étudiant(e)s déjà inscrit(e)s

séance de tutorat :

séance ci-dessus et cliquer sur le gros bouton ci-dessous ensuite :-)

Contact - Support



Planning

Il est impossible de faire le projet après la deadline !

Plus d'assistant !

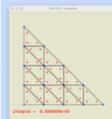
Plus de tuteur !

Pas de projet en août !

Cela commence dès lundi !

Devoirs en C...

Les énoncés seront disponibles progressivement...



Devoir 1 : Integrate (25/01/2024)

Projet à télécharger : [Integrate.zip](#)

Énoncé du devoir : [Integrate.pdf](#)

Deadline : **Lundi 17 février 2025 à 23h59**

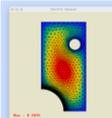


Devoir 2 : GeoMesh (12/02/2024)

Projet à télécharger : [GeoMesh.zip](#)

Énoncé du devoir : [GeoMesh.pdf](#)

Deadline : **Lundi 24 février 2025 à 23h59**

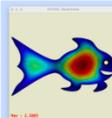


Devoir 3 : Poisson (26/02/2024)

Projet à télécharger : [Poisson.zip](#)

Énoncé du devoir : [Poisson.pdf](#)

Deadline : **Lundi 3 mars 2025 à 23h59**

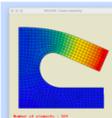


Devoir 4 : BandSolver (04/03/2024)

Projet à télécharger : [BandSolver.zip](#)

Énoncé du devoir : [BandSolver.pdf](#)

Deadline : **Lundi 10 mars 2025 à 23h59**

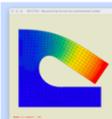


Devoir 5 : LinearElasticity (11/03/2024)

Projet à télécharger : [LinearElasticity.zip](#)

Énoncé du devoir : [LinearElasticity.pdf](#)

Deadline : **Lundi 17 mars 2025 à 23h59**



Devoir 6 : LinearElasticityForces (23/03/2024)

Projet à télécharger : [LinearElasticityForces.zip](#)

Énoncé du devoir : [LinearElasticityForces.pdf](#)

Deadline : **Lundi 24 mars 2025 à 23h59**

S2-S8

Faire le projet !

**Avec l'aide de votre assistant
Avec l'aide de votre tuteur**

Remise du projet : 25 mars

S9-S13

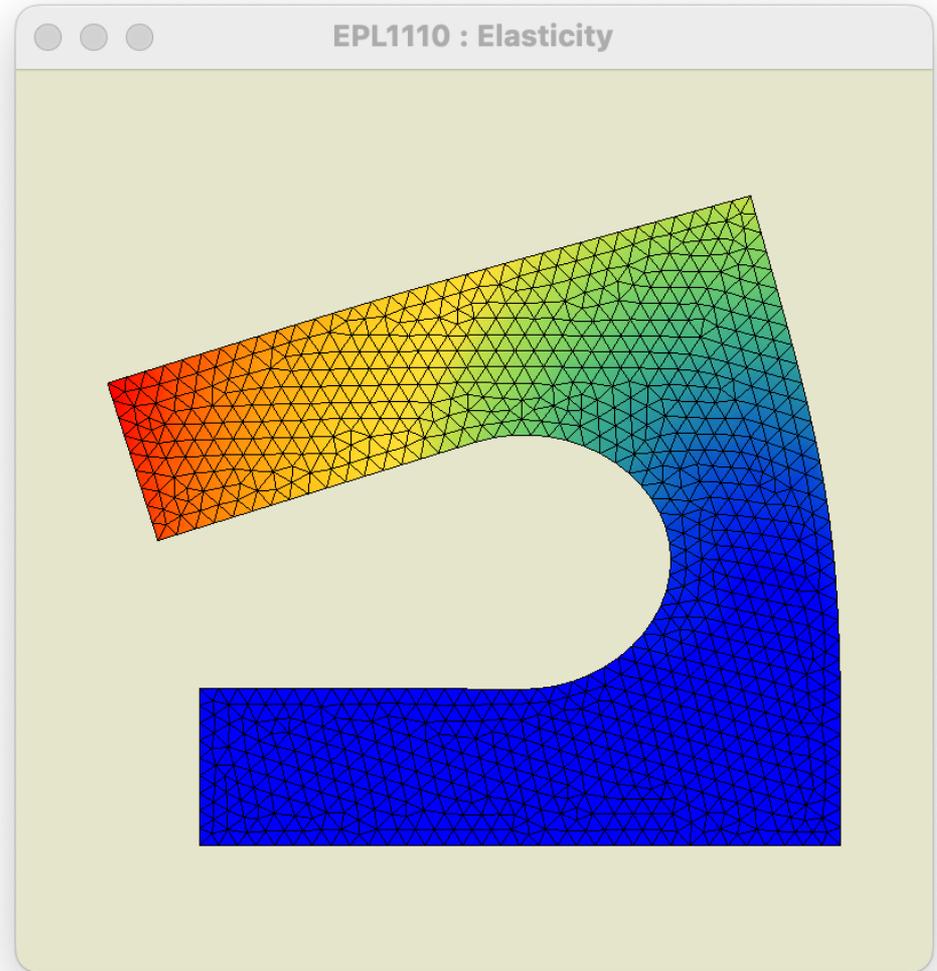
Aspects théoriques !

Cours à l'ancienne :-)

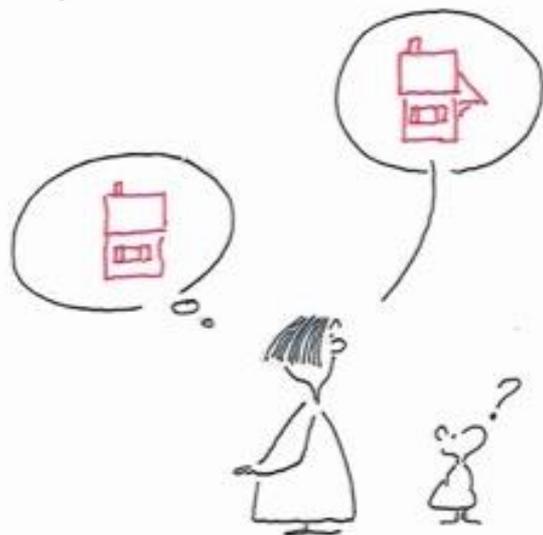
Pas de séances d'exercices !

Examen écrit en session

Elasticité linéaire



Le projet



J-15



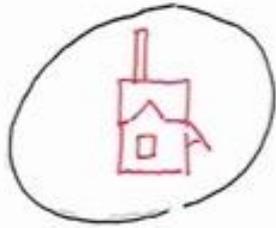
J-14



J-12



J-10



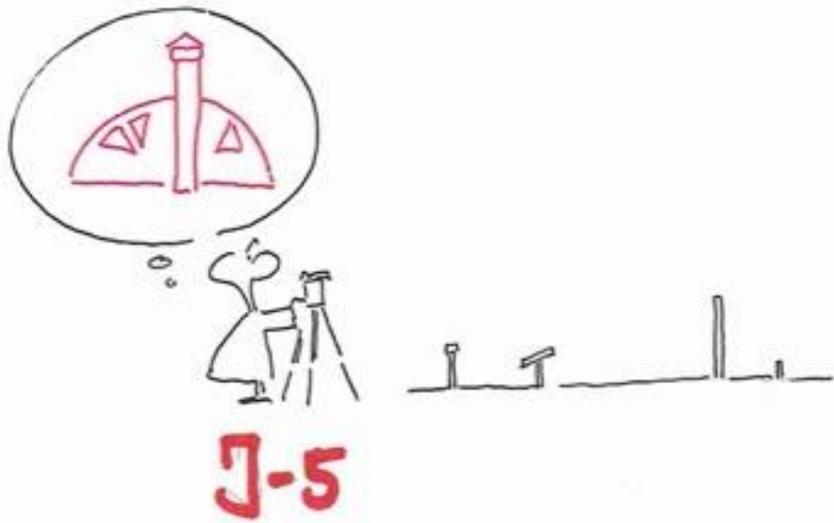
J-14



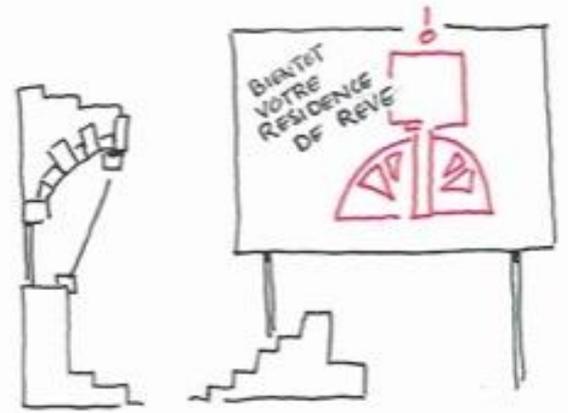
J-12

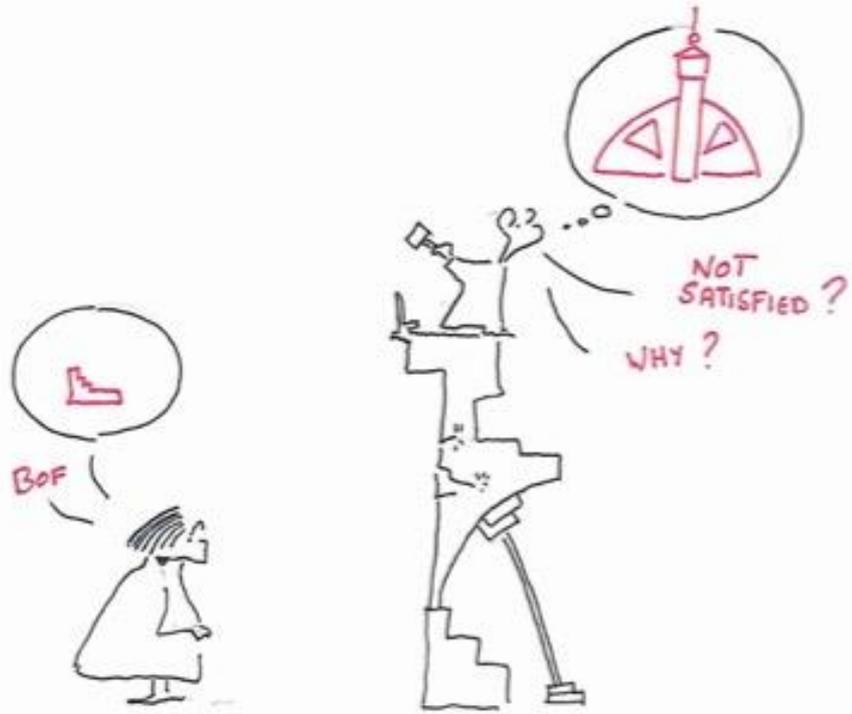


J-10

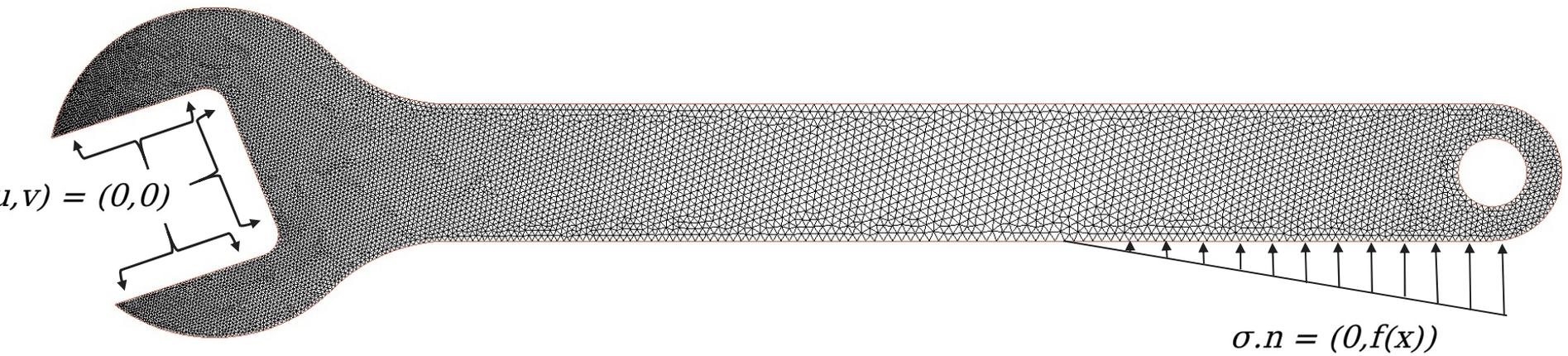


J-4





J



Quelle est la force maximale
avant de casser l'outil ?

Conservation de la quantité de mouvement

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{f} = 0$$

Conservation du moment de la quantité de mouvement

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T$$

Loi de comportement de Hooke

$$\boldsymbol{\sigma} = \underbrace{\frac{E}{(1+\nu)}}_{2\mu} \boldsymbol{\epsilon} + \underbrace{\frac{E\nu}{(1+\nu)(1-2\nu)}}_{\lambda} \text{tr}(\boldsymbol{\epsilon}) \boldsymbol{\delta}$$

Cinématique

$$\boldsymbol{\epsilon} = \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right)$$

Modèle mathématique
de l'élasticité linéaire

Créer une géométrie !

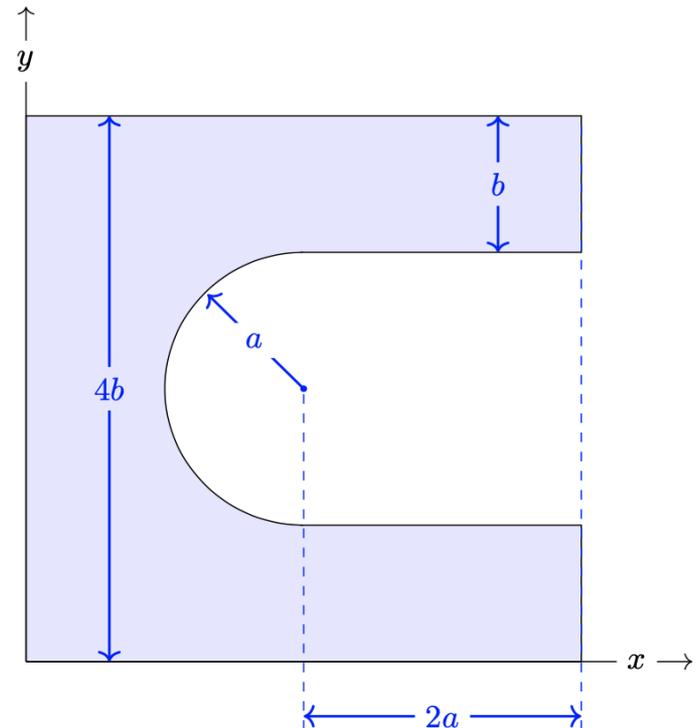
Construire le maillage !

Définir la géométrie et le maillage !

Le maillage et la géométrie sont définis comme suit :

```
double Lx = 1.0;
double Ly = 1.0;

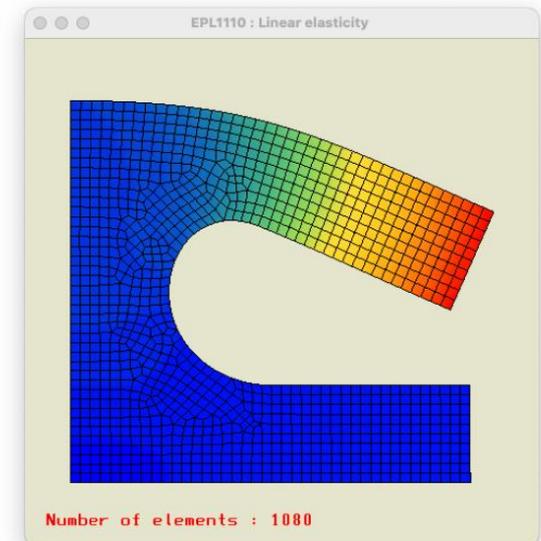
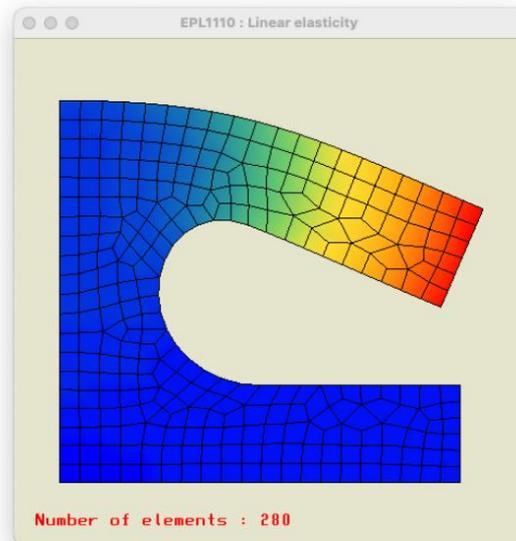
theGeometry->LxPlate = Lx;
theGeometry->LyPlate = Ly;
theGeometry->h = Lx * 0.05;
theGeometry->elementType = FEM_TRIANGLE;
```



Que va-t-on résoudre ?

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

$$\underbrace{\langle \boldsymbol{\epsilon}(\hat{\mathbf{u}}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u}) \rangle}_{a(\hat{\mathbf{u}}, \mathbf{u})} = \underbrace{\langle \hat{\mathbf{u}} f \rangle + \ll \hat{\mathbf{u}} g \gg_N}_{b(\hat{\mathbf{u}})}, \quad \forall \hat{\mathbf{u}} \in \hat{\mathcal{U}},$$



Et on minimise
toujours une fonctionnelle !

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

$$\underbrace{\langle \boldsymbol{\epsilon}(\hat{\mathbf{u}}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{u}). \rangle}_{a(\hat{\mathbf{u}}, \mathbf{u})} = \underbrace{\langle \hat{\mathbf{u}}f \rangle + \ll \hat{\mathbf{u}}g \gg_N}_{b(\hat{\mathbf{u}})}, \quad \forall \hat{\mathbf{u}} \in \hat{\mathcal{U}},$$

$$J(\mathbf{v}) = \underbrace{\frac{1}{2} \langle \boldsymbol{\epsilon}(\mathbf{v}) : \mathbf{C} : \boldsymbol{\epsilon}(\mathbf{v}). \rangle}_{\frac{1}{2}a(\mathbf{v}, \mathbf{v})} - \underbrace{\langle \mathbf{v}f \rangle + \ll \mathbf{v}g \gg_N}_{b(\mathbf{v})},$$

En deux dimensions !

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} = \begin{bmatrix} A\epsilon_{xx} + B\epsilon_{yy} & 2C\epsilon_{xy} \\ 2C\epsilon_{xy} & A\epsilon_{yy} + B\epsilon_{xx} \end{bmatrix}$$

Acier

$$E = 2.11 \cdot 10^{11} \text{ [N/m}^2\text{]}$$

$$\nu = 0.3$$

$$\rho = 7.85 \cdot 10^3 \text{ [kg/m}^3\text{]}$$

Déformations planes

Tensions planes

$$A = \frac{E(1 - \nu)}{(1 + \nu)(1 - 2\nu)}$$

$$A = \frac{E}{(1 - \nu^2)}$$

$$B = \frac{E\nu}{(1 + \nu)(1 - 2\nu)}$$

$$B = \frac{E\nu}{(1 - \nu^2)}$$

$$C = \frac{E}{2(1 + \nu)}$$

Déformations planes...

$$\sigma_{xx} = \frac{E}{(1+\nu)(1-2\nu)} ((1-\nu)\epsilon_{xx} + \nu\epsilon_{yy}),$$

$$\sigma_{yy} = \frac{E}{(1+\nu)(1-2\nu)} (\nu\epsilon_{xx} + (1-\nu)\epsilon_{yy}),$$

$$\sigma_{zz} = \frac{E\nu}{(1+\nu)(1-2\nu)} (\epsilon_{xx} + \epsilon_{yy}),$$

$$\sigma_{xy} = \frac{E}{(1+\nu)} \epsilon_{xy},$$

$$\sigma_{xx} = \frac{E}{(1-\nu^2)} (\epsilon_{xx} + \nu\epsilon_{yy}),$$

$$\sigma_{yy} = \frac{E}{(1-\nu^2)} (\nu\epsilon_{xx} + \epsilon_{yy}),$$

$$\sigma_{xy} = \frac{E}{(1+\nu)} \epsilon_{xy}.$$

...tensions planes

La fonctionnelle à minimiser !

$$\begin{aligned}\frac{1}{2} a(\mathbf{u}, \mathbf{u}) &= \frac{1}{2} \langle \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\sigma}(\mathbf{u}) \rangle, \\ &= \frac{1}{2} \langle \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{xy} & \epsilon_{yy} \end{bmatrix} : \begin{bmatrix} A\epsilon_{xx} + B\epsilon_{yy} & 2C\epsilon_{xy} \\ 2C\epsilon_{xy} & A\epsilon_{yy} + B\epsilon_{xx} \end{bmatrix} \rangle, \\ &= \frac{1}{2} \langle \begin{bmatrix} \epsilon_{xx} & \epsilon_{yy} & 2\epsilon_{xy} \end{bmatrix} \cdot \begin{bmatrix} A & B & 0 \\ B & A & 0 \\ 0 & 0 & C \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \end{bmatrix} \rangle.\end{aligned}$$

Matrice et vecteur locaux à construire !

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

$$\mathbf{B}_i = \left[\begin{array}{c} \langle \tau_i f_x \rangle + \langle\langle \tau_i g_x \rangle\rangle \\ \langle \tau_i f_y \rangle + \langle\langle \tau_i g_y \rangle\rangle \end{array} \right].$$

Et on soumet au serveur...



```
=====  
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  
=====
```

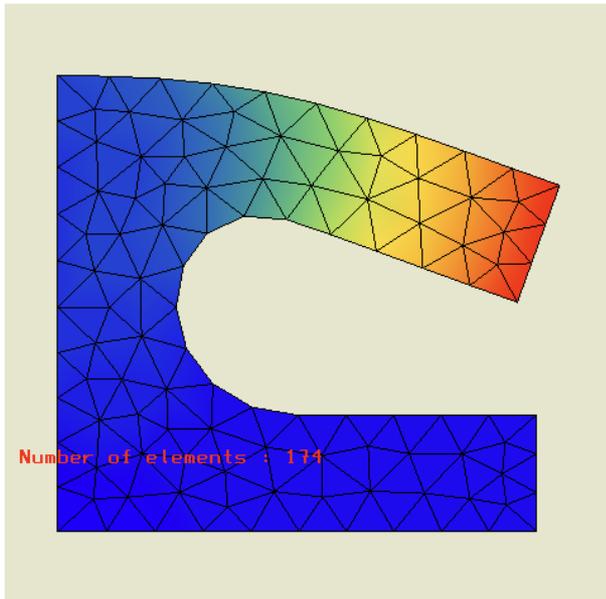
```
Pivot index 1 Pivot value 0.000000e+00
```

```
-----  
Error in src/fem.c at line 555 :  
Cannot eliminate with such a pivot
```

```
----- Yek Yek !!
```

```
La compilation ou l'execution du test 0 a plante :-(
```

Et après avoir fait le devoir...



Diagnostic du serveur

```
***** TEST 0 *****
gcc -o /tmp/work/exe src/qlfem.c src/main.c src/fem.c src/mesaglfem.c homework.c -I src -lm -lrt -Wall -g -Wno-
** COMPILATION SUCCEED **

/usr/bin/time -o /work/outputStat.txt -f %R %Z %e %M sudo -ustudent ./exe
** RUN SUCCEED **
Wall-clock time : 0.24s (limit 40s)
Approximated maximum memory usage : 112.98Mb
Approximated total memory allocation : 28.8203Mb
***** RUN OUTPUT *****
Info : Meshing 1D...
Info : [ 0%] Meshing curve 1 (Line)
Info : [ 20%] Meshing curve 2 (Line)
Info : [ 30%] Meshing curve 3 (Line)
```

PROBLÈME ELLIPTIQUE



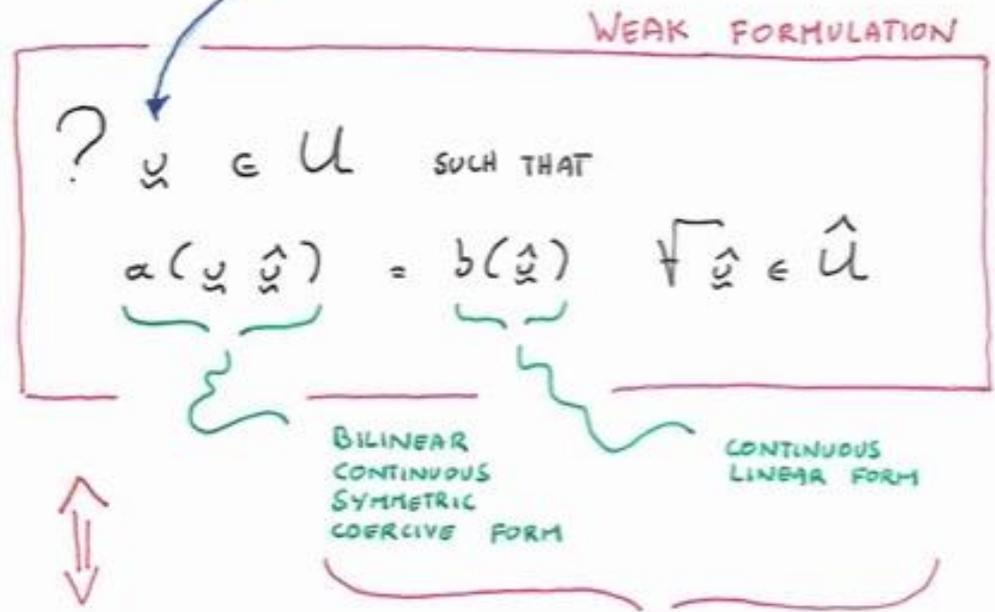
ELEMENTS FINIS *

 * "USUAL FEM"
CLASSICAL
GALERKIN
FORMULATION

ABSTRACT GENERIC ELLIPTIC PROBLEM

HEAT CONDUCTION
 LINEAR ELASTICITY
 SIMPLIFIED MODELS OF LINEAR ELASTICITY
 BEAM / SHELLS
 ROPE / MEMBRANE
 STOKES PROBLEM

Now,
 WE CONSIDER
 A VECTORIAL UNKNOWN FIELD !



ASSUMPTIONS
 REQUIRED
 TO HAVE AN ABSTRACT
 MINIMIZATION
 PROBLEM

? $\underline{u} \in \mathcal{U}$ SUCH THAT

$$\mathcal{J}(\underline{u}) = \min_{\underline{v} \in \mathcal{U}} \underbrace{\frac{1}{2} a(\underline{v}, \underline{v}) - b(\underline{v})}_{\mathcal{J}(\underline{v})}$$

MINIMIZATION
 PROBLEM

HEAT CONDUCTION

CONSERVATION LAW

$$-\nabla \cdot \underline{q} + f = 0$$

HEAT FLOW

CONSTITUTIVE LAW

$$\underline{q} = -k \nabla u$$

FOURIER

TEMPERATURE

? u SUCH THAT

$$\nabla \cdot (\overbrace{-\underline{q}(u)}^{-k \nabla u}) + f = 0 \quad \text{in } \Omega$$

$$-\underline{q} \cdot \underline{n} = g \quad \text{ON } \Gamma_2$$

$$u = 0 \quad \text{ON } \Gamma_0$$

3D LINEAR ISOTROPIC ELASTICITY



HOOKE ELASTIC BODY
SMALL DEF.

CONSTITUTIVE LAW

$$\underline{\underline{\sigma}} = \underbrace{\frac{E}{(1+\nu)}}_{2\mu} \underline{\underline{\epsilon}} + \underbrace{\frac{E\nu}{(1+\nu)(1-2\nu)}}_{\lambda} \text{tr}(\underline{\underline{\epsilon}}) \underline{\underline{1}}$$

CONSERVATION LAWS

$$\nabla \cdot \underline{\underline{\sigma}} + \underline{\underline{f}} = 0$$

$$\underline{\underline{\sigma}} = \underline{\underline{\sigma}}^T$$

DEFORMATION TENSOR

$$\underline{\underline{\epsilon}} \triangleq \frac{1}{2} (\nabla_{\underline{\underline{u}}} + (\nabla_{\underline{\underline{u}}})^T)$$

Boundary Value Problem:

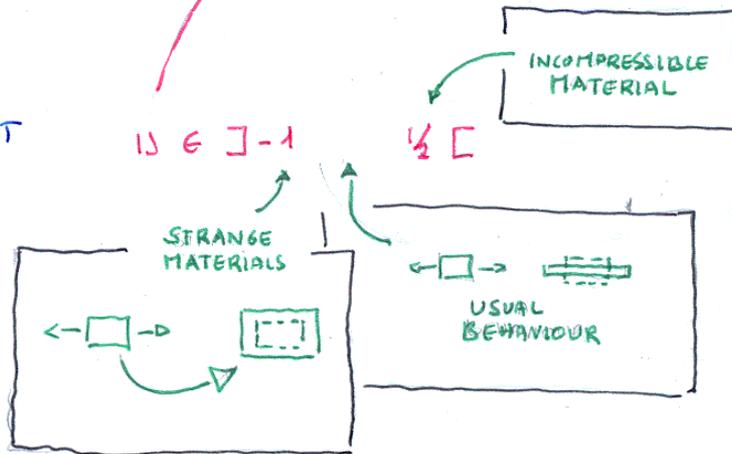
$$\nabla \cdot \underline{\underline{\sigma}}(\underline{\underline{u}}) + \underline{\underline{f}} = 0 \quad \text{in } \Omega$$

$$\underline{\underline{\sigma}} \cdot \underline{\underline{n}} = \underline{\underline{g}} \quad \text{ON } \Gamma_2$$

$$\underline{\underline{u}} = 0 \quad \text{ON } \Gamma_0$$

$$\underline{\underline{\sigma}} = \underbrace{\frac{E}{(1+\nu)}}_{2\mu \text{ SHEAR MODULUS}} \underline{\underline{\epsilon}} + \underbrace{\frac{E\nu}{(1+\nu)(1-2\nu)}}_{\lambda} t_n(\underline{\underline{\epsilon}}) \underline{\underline{S}}$$

POISSON'S COEFFICIENT



$\lambda \rightarrow \infty$
IF $\nu \rightarrow 1/2$

STOKES PROBLEM

CONSERVATION LAWS

$$\cancel{(\underline{v} \cdot \nabla) \underline{v}} = \nabla \cdot \underline{\underline{D}} + \underline{f}$$

$$\nabla \cdot \underline{v} = 0$$

CREEPING FLOW
 $Re \ll 0$

INCOMPRESSIBLE FLOW

CONSTITUTIVE LAW

NEWTONIAN FLUID

$$\underline{\underline{D}} = 2\mu \underline{\underline{d}} \rightarrow \underline{\underline{S}}$$

RATE OF DEFORMATION TENSOR

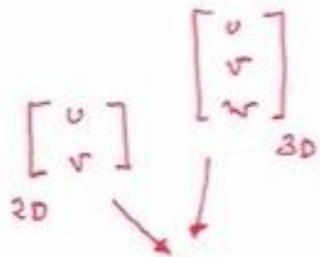
$$\underline{\underline{d}} \triangleq \frac{1}{2} (\nabla \underline{v} + (\nabla \underline{v})^T)$$

? (\underline{v}, p)

$$\begin{cases} \nabla \cdot (\underline{\underline{D}}(\underline{v}, p)) + \underline{f} = 0 \\ \nabla \cdot \underline{v} = 0 \end{cases} \quad \text{in } \Omega$$

$$\begin{aligned} \underline{\underline{D}} \cdot \underline{e}_3 &= \underline{g} & \text{on } \sqrt{2} \\ \underline{v} &= \underline{0} & \text{on } \sqrt{2} \end{aligned}$$

ABSTRACT GENERIC DISCRETE FORMULATION



$$\underbrace{u(x)}_{\in \mathcal{U}}$$

$$\approx \underbrace{u^h(x)}_{\in \mathcal{U}^h \subset \mathcal{U}} = \sum_{i=1}^m \underbrace{u_i}_{\in \mathcal{U}} \tau_i(x)$$

$$\dim(\mathcal{U}^h) = 2m$$

$$\dots = 3m$$

3D PROBLEMS

$$\uparrow$$

2D PROBLEMS

? u^h SUCH THAT

$$a(u^h, \hat{u}^h) = b(\hat{u}^h)$$

$$\forall \hat{u}^h \in \hat{\mathcal{U}}^h$$

DISCRETE
FORMULATION

$$\mathcal{U}^h = \text{SPAN} \left\{ \underbrace{\begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} \begin{bmatrix} \tau_2 \\ 0 \end{bmatrix} \dots \begin{bmatrix} \tau_m \\ 0 \end{bmatrix}}_m \underbrace{\begin{bmatrix} 0 \\ \tau_1 \end{bmatrix} \dots \begin{bmatrix} 0 \\ \tau_m \end{bmatrix}}_m \right\}$$

$$\begin{aligned}
 \langle \hat{u} \cdot (\nabla \cdot \underline{g}(u)) \rangle & \stackrel{\text{TIP \#1}}{=} \langle \nabla \cdot (\hat{u} \cdot \underline{g}(u)) \rangle - \langle \nabla \hat{u} : \underline{g}(u) \rangle \\
 & \stackrel{\text{TIP \#2}}{=} \langle \nabla \cdot \underline{g}(u) \rangle \\
 & \stackrel{\text{TIP \#3}}{=} \langle \nabla \cdot (\underline{g}(u)) \rangle
 \end{aligned}$$

Annotations:

- \hat{u} is labeled with $\epsilon \hat{u}$.
- $\underline{g}(u)$ is labeled with ϵu .
- The term $\langle \nabla \hat{u} : \underline{g}(u) \rangle$ is labeled with $\epsilon \langle \nabla \hat{u} : \underline{g}(u) \rangle$.
- The final expression $\langle \nabla \cdot (\underline{g}(u)) \rangle$ is labeled with $\epsilon \langle \nabla \cdot \underline{g}(u) \rangle$.

**USUAL
CALCULUS !**

IS $\alpha(\hat{u}, u)$
SYMMETRIC?

$$\begin{aligned}
 \langle \nabla_{\hat{u}} : \mathbb{D}(u) \rangle &= \langle \underbrace{\left(\frac{\nabla_{\hat{u}} + (\nabla_{\hat{u}})^T}{2} \right)}_{\text{SYM}} + \underbrace{\left(\frac{\nabla_{\hat{u}} - (\nabla_{\hat{u}})^T}{2} \right)}_{\text{ANTI-SYM}} : \underbrace{\mathbb{D}(u)}_{\text{SYM}} \rangle \\
 &= \langle \underline{\underline{\underline{\varepsilon}}(\hat{u})} : \mathbb{D}(u) \rangle \\
 &= \langle \underline{\underline{\underline{\varepsilon}}(\hat{u})} : \underbrace{\mathbb{C}}_{\text{GENERALIZED HOOKE'S LAW}} : \underline{\underline{\underline{\varepsilon}}}(u) \rangle
 \end{aligned}$$

WEAK FORMULATION

? $u \in U$ SUCH THAT

$$\underbrace{\langle \underline{\underline{\hat{u}}}: \underline{\underline{\hat{u}}}: \underline{\underline{u}} \rangle}_{a(\hat{u}, u)} = \underbrace{\langle \hat{u} \cdot f \rangle + \langle \hat{u} \cdot g \rangle_N}_{b(\hat{u})}$$

$\forall \hat{u} \in \hat{U}$

FOR
SUITABLE
ONLY \subseteq !

? $u \in U$ SUCH THAT

$$J(u) = \min_{v \in U} \underbrace{\frac{1}{2} a(u, v)}_{J(v)} - b(v)$$

MINIMIZATION PROBLEM

$$\frac{1}{2} \alpha(\underline{v}, \underline{v}) = \left\langle \frac{1}{2} \lambda \underline{\underline{\epsilon}}(\underline{v}) : \underline{\underline{\epsilon}}(\underline{v}) + \mu \underline{\underline{\epsilon}}(\underline{v}) : \underline{\underline{\epsilon}}(\underline{v}) \right\rangle$$

ENERGY OF DEFORMATION

MUST BE A QUADRATIC POSITIVE FORM IN ORDER TO OBTAIN A MINIMIZATION PROBLEM



$\mu > 0$
$\frac{3}{2} \lambda + \mu > 0$

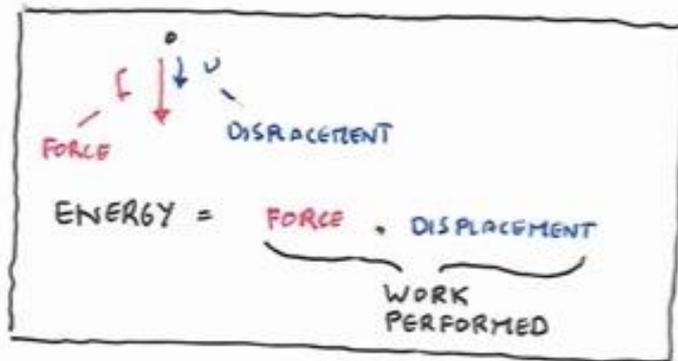
$(\epsilon_{12} \neq 0)$

$(\epsilon_{11} = \epsilon_{22} = \epsilon_{33} \neq 0)$

ADMISSIBLE VALUES FOR LAMÉ COEFFICIENTS

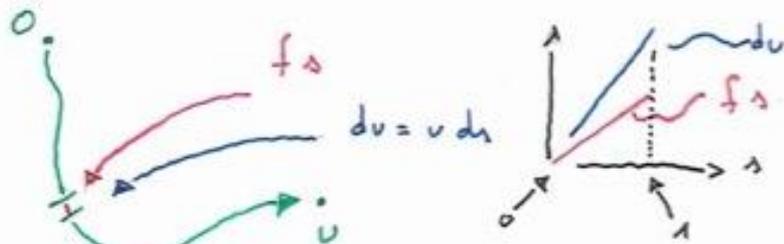
"DEFORMATION ENERGY"

WHAT IS IT?



ENERGY OF DEFORMATION

= WORK PERFORMED TO DEFORM THE STRUCTURE AND IS STORED INSIDE. SUCH AN ENERGY IS RECOVERED WHEN EXTERNAL FORCES ARE REMOVED



VERY VERY SLOW LOADING
 → QUASI-STATIC APPROACH

- IRREVERSIBILITY
 - DISSIPATION
- ARE NEGLECTED!

WORK PERFORMED = $\int_0^u f du$

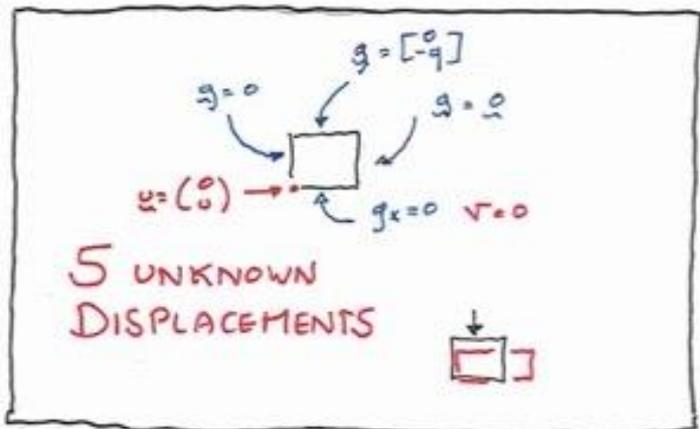
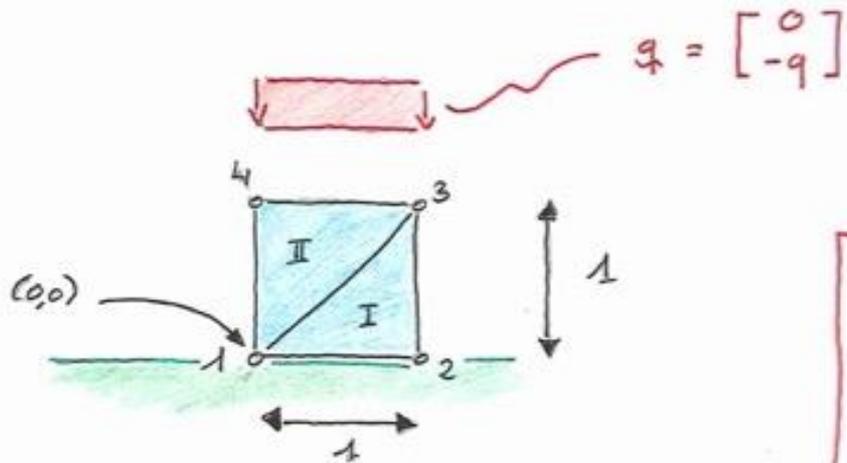
= $\int_0^1 f u ds$

= $f u \int_0^1 ds$

= $\frac{1}{2} f u$!

NUMERICAL EXAMPLE

$$\xi_c = \begin{bmatrix} 0 \\ -q \end{bmatrix}$$



$$? \hat{u}^h = \sum_{i=1}^m \hat{u}_i \tau_i$$

$$\langle \underline{\underline{\epsilon}}(\hat{u}^h) : \underline{\underline{\sigma}}(\hat{u}^h) \rangle = \langle f \cdot \hat{u}^h \rangle + \langle \langle g \cdot \hat{u}^h \rangle \rangle_N$$

$$\begin{bmatrix} \tau_i \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ \tau_i \end{bmatrix}$$

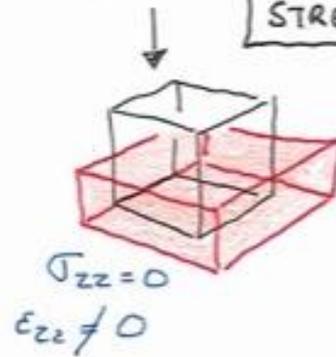
2m TEST FUNCTIONS

ANALYTICAL SOLUTION

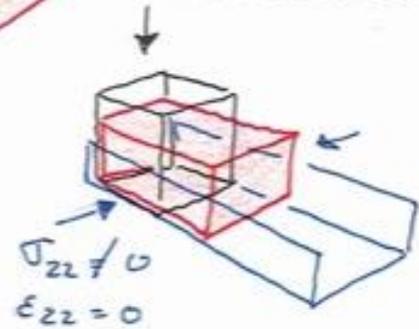
PLANAR STRESSES

$$\begin{aligned} \sigma_c &= \frac{q}{E} \begin{bmatrix} \nu x \\ -y \end{bmatrix} \\ \epsilon_{zz} &= \frac{q}{E} \begin{bmatrix} \nu & 0 \\ 0 & -1 \end{bmatrix} \end{aligned}$$

PLANAR STRESSES



PLANAR DEFORMATIONS



$$\Delta \cdot \epsilon_{zz} = 0$$

= 0!

$$\begin{aligned} \sigma_{xx} &= \frac{E}{(1-\nu^2)} \left(\underbrace{\epsilon_{xx}}_{\frac{q\nu}{E}} + \nu \underbrace{\epsilon_{yy}}_{-\frac{q\nu}{E}} \right) = 0 \\ \sigma_{yy} &= \frac{E}{(1-\nu^2)} \left(\nu \underbrace{\epsilon_{xx}}_{\frac{q}{E}} - \underbrace{\epsilon_{yy}}_{(1-\nu^2)\frac{q}{E}} \right) = -q! \end{aligned}$$

$$\begin{bmatrix} \tau_i \\ 0 \end{bmatrix} \text{ OR } \begin{bmatrix} 0 \\ \tau_i \end{bmatrix}$$

$$\langle \underbrace{\| \mathbb{E}(\hat{u}_i) \|}_{\text{red}} : \underbrace{\| \mathbb{E}(\hat{u}_j) \|}_{\text{green}} \rangle$$

$$\sum_j U_j \mathbb{E} \begin{pmatrix} \tau_j \\ 0 \end{pmatrix} + V_j \mathbb{E} \begin{pmatrix} 0 \\ \tau_j \end{pmatrix}$$

$$\sum_j A_{ij} \cdot U_j$$

$$\sum_j \begin{bmatrix} U_j \\ V_j \end{bmatrix} \tau_j = \sum_j U_j \begin{bmatrix} \tau_j \\ 0 \end{bmatrix} + V_j \begin{bmatrix} 0 \\ \tau_j \end{bmatrix}$$

$$\sum_j \begin{bmatrix} \langle \mathbb{E}(\hat{u}_i) : \mathbb{E}(\hat{u}_j) \rangle & \langle \mathbb{E}(\hat{u}_i) : \mathbb{E}(\hat{u}_j) \rangle \\ \langle \mathbb{E}(\hat{u}_i) : \mathbb{E}(\hat{u}_j) \rangle & \langle \mathbb{E}(\hat{u}_i) : \mathbb{E}(\hat{u}_j) \rangle \end{bmatrix} \cdot \begin{bmatrix} U_j \\ V_j \end{bmatrix}$$

DISCRETE OPERATOR

$$\underline{\underline{\sigma}} \begin{pmatrix} \tau_{iz} \\ 0 \end{pmatrix} = \begin{bmatrix} \tau_{iz,x} & \tau_{iz,y}/2 \\ \tau_{iz,y}/2 & 0 \end{bmatrix}$$

$$\underline{\underline{\sigma}} \begin{pmatrix} 0 \\ \tau_{iz} \end{pmatrix} = \begin{bmatrix} 0 & \tau_{iz,x}/2 \\ \tau_{iz,x}/2 & \tau_{iz,y} \end{bmatrix}$$

$$\underline{\underline{\sigma}} \begin{pmatrix} \underline{\underline{\epsilon}} \end{pmatrix} = \begin{bmatrix} A \epsilon_{xx} + B \epsilon_{yy} & 2C \epsilon_{xy} \\ 2C \epsilon_{xy} & A \epsilon_{yy} + B \epsilon_{xx} \end{bmatrix}$$

HOW
TO
CALCULATE
IT ?

$\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$	$\frac{E}{(1-\nu^2)}$
$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{E\nu}{(1-\nu^2)}$
	$\frac{E}{2(1+\nu)}$

PLANAR
DEFORMATIONS

PLANAR
STRESSES

A

B

C

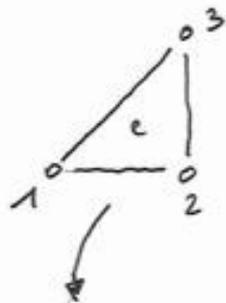
$$\underline{\underline{\sigma}} : \underline{\underline{\epsilon}} = \epsilon_{xx} (A \epsilon_{xx} + B \epsilon_{yy}) + 4C \epsilon_{xy} \epsilon_{xy} + \epsilon_{yy} (A \epsilon_{yy} + B \epsilon_{xx})$$

$$\underline{\underline{A}}_{ij} = \left[\begin{array}{cc} \langle \tau_{i,x} A \tau_{j,x} \rangle + \langle \tau_{i,y} C \tau_{j,y} \rangle & \langle \tau_{i,x} B \tau_{j,y} \rangle + \langle \tau_{i,y} C \tau_{j,x} \rangle \\ \langle \tau_{i,y} B \tau_{j,x} \rangle + \langle \tau_{i,x} C \tau_{j,y} \rangle & \langle \tau_{i,y} A \tau_{j,y} \rangle + \langle \tau_{i,x} C \tau_{j,x} \rangle \end{array} \right]$$

$$\underline{\underline{\epsilon}} \left(\begin{smallmatrix} 0 \\ \tau_i \end{smallmatrix} \right) : \underline{\underline{\sigma}} \left(\underline{\underline{\epsilon}} \left(\begin{smallmatrix} \tau_j \\ 0 \end{smallmatrix} \right) \right)$$

**LOCAL
ELASTICITY
MATRIX**

COMPUTING A LOCAL ELASTICITY MATRIX



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

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

$$A_{ij}^e = \begin{cases} A_{xxij} = A \\ A_{yyij} = C \\ A_{xyij} = 0 \\ A_{yxij} = 0 \end{cases}$$

$$A_{ij}^e =$$

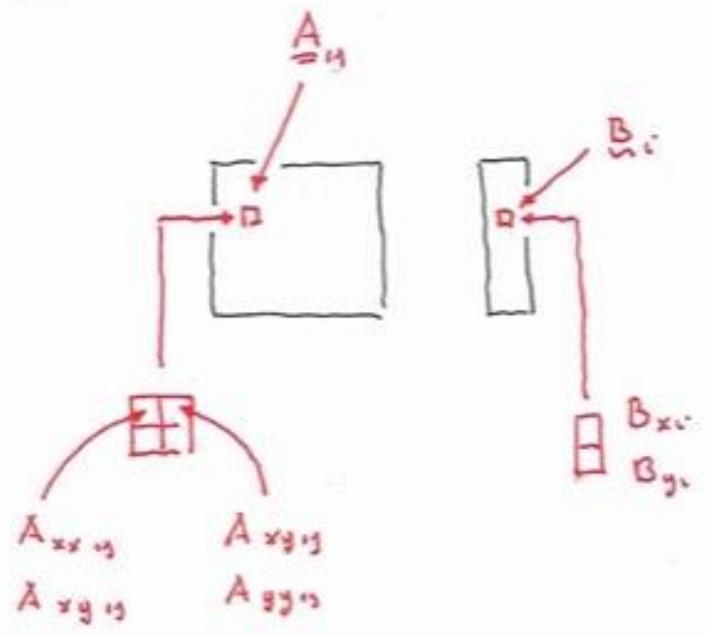
$$\left[\begin{array}{cc|cc} A & 0 & -A & B \\ 0 & C & C & -C \\ \hline & & A+C & -B-C \\ & & -B-C & A+C \\ \hline & & & C \\ & & & 0 \\ & & & 0 & A \end{array} \right] \frac{1}{2}$$

$$\underline{B}_i = \langle\langle g \cdot \hat{u}^h \rangle\rangle_N$$

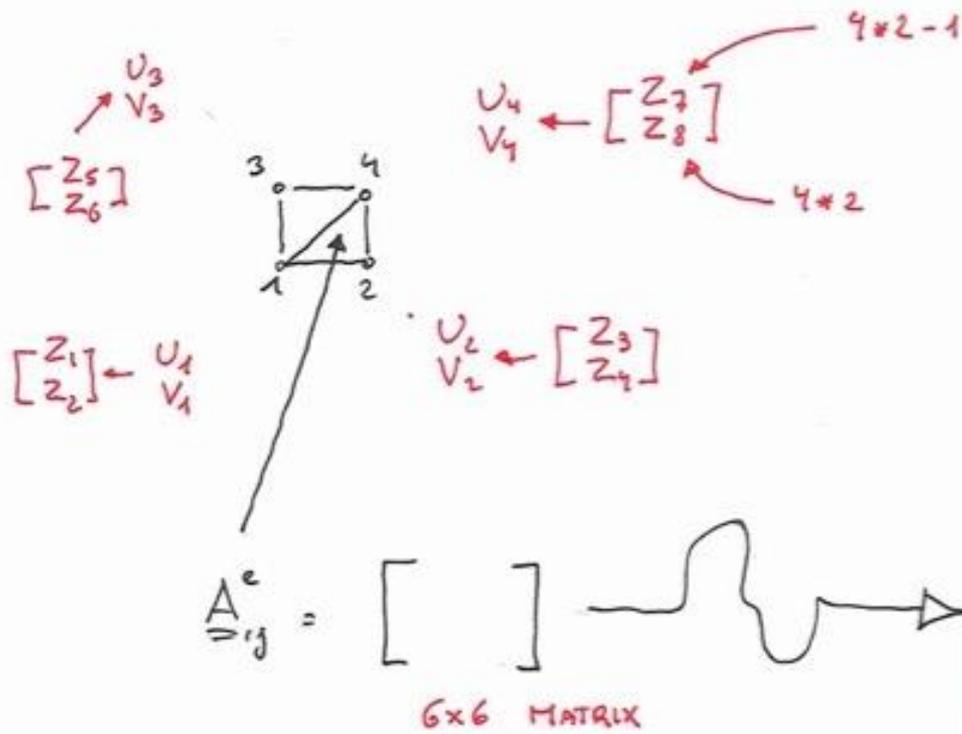
$\left[\begin{matrix} \tau_i \\ 0 \end{matrix} \right]$ OR $\left[\begin{matrix} 0 \\ \tau_i \end{matrix} \right]$

$$\begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

$$\begin{bmatrix} \langle\langle g_x \tau_i \rangle\rangle \\ \langle\langle g_y \tau_i \rangle\rangle \end{bmatrix}$$



AND
HYPER
VECTOR !



ASSEMBLING PROCEDURE

$$\left[\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{array} \right] \left[\begin{array}{c} Z_1 \\ \vdots \\ Z_8 \end{array} \right] = \left[\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{array} \right]$$

GLOBAL HORIZONTAL INDEX

$$= 2 * \underbrace{\text{GLOBAL SCALAR INDEX}}_{\text{GLOBAL}} - 1$$