

Simulation numérique par éléments finis d'un essai de traction avec le logiciel Cast3M

13 juin 2016



Etudiants :

Jules Clamon

Claire Driguez

Tom Masson

Sinwen Mohamed Stambouli

Enseignant-responsable du projet :

Clement Keller

Référence du projet : STPI/P6/2016 – 16

Date de remise du rapport : 13/06/2016

Référence du projet : STPI/P6/2016 – 16

Intitulé du projet : Simulation numérique par éléments finis d'un essai de traction avec le logiciel Cast3M

Type de projet : Simulation numérique

Objectifs du projet : Ce projet a pour objectif de réaliser une simulation numérique par éléments finis d'un essai de traction avec le logiciel Cast3M sur une éprouvette . Cela requiert à la fois des connaissances en programmation et en mathématique . Cet apprentissage doit se faire en autonomie et avec l'aide du professeur et permet de développer l'esprit d'équipe. Cet objectif se caractérise par la prise en main du du logiciel et la compréhension des outils mathématiques utiliser.

Mots-clefs du projet : éléments finis, programmation, traction, numérique.

INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE ROUEN

Département Sciences et Techniques Pour l'Ingénieur

avenue de l'Université - 76801 Saint Etienne du Rouvray

- tél : +33 (0)2 32 95 97 00 - fax : +33 (0)2 32 95 98 60

Table des matières

I	Explication de la méthode des éléments finis pour le cas d'une élasticité à une dimension	5
1	Obtention d'une équation différentielle précise	5
1.1	Définition de l'objet d'étude et des paramètres influençant les caractéristiques de l'effort produisant la déformation	5
1.2	Application du principe fondamental de la dynamique (PFD) . . .	6
1.3	Amélioration de l'expression différentielle par la méthode des résidus pondérés	7
2	Démarche de résolution de l'équation différentielle trouvée par discrétisation	8
3	Résolution de l'équation par expression matricielle	9
4	Exemple	9
4.1	PFD	9
4.2	Loi de comportement en élasticité	9
4.3	Condition limite	9
4.4	Forme integrale	10
4.5	Ecriture du résidu du premier élément	11
II	Notice d'utilisation simplifiée de Cast3m	13
5	Déclaration des points	13
6	Choix du maillage et traçage des droites	13
7	Surfaces	14
8	Conditions limites	14
9	Définition du modèle	14
10	Définition du mouvement	15
11	Calcul pas à pas	15
12	Post traitement	16
13	Critique des résultats	17

Introduction

Le semestre 4 de notre formation d'ingénieur est constitué de matières variées et les élèves suivent des thématiques différentes. Après le projet informatique et le projet mathématique du semestre 3, les élèves de l'INSA réalisent un projet de physique dans le cadre de la matière P6. Ce projet se réalise par équipes de quelques élèves. Notre projet consiste à réaliser une simulation numérique par éléments finis d'un essai de traction avec le logiciel Cast3M. Pour ce faire, nous disposons de 13 séances de cours, en plus de nos recherches personnelles. Le cahier des charges est simple : il faut réaliser une simulation numérique de traction sur une éprouvette avec le logiciel Cast3M et comparer nos résultats avec l'essai de traction sur une vraie éprouvette.

La méthode des éléments finis consiste à estimer une solution d'un problème aux limites statiques ou dynamiques telle que la mécanique des milieux continus ce qui est notre cas. Cette méthode est indispensable dans différents domaines comme la mécanique, l'énergétique et le génie civil et autres. En ce qui concerne la partie codage, il faut apprendre dans un premier temps à maîtriser le logiciel afin de pouvoir tracer la pièce et définir le maillage puis coder les calculs étudiés dans la première partie afin de pouvoir exploiter les résultats.

Première partie

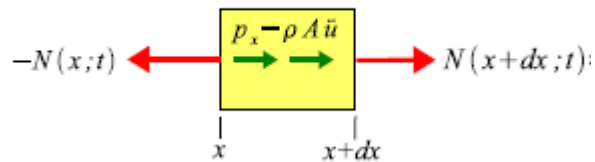
Explication de la méthode des éléments finis pour le cas d'une élasticité à une dimension

Tout d'abord, le bilan des forces, et l'application du principe fondamental de la dynamique permet d'obtenir une équation représentative du mouvement sous forme différentielle. On effectue la méthode des résidus ponctuels détaillée ultérieurement pour ensuite obtenir une forme intégrale. Les solutions de cette équation sont difficiles à obtenir. On effectue alors la méthode des travaux virtuels, on obtient une deuxième forme intégrale, et après une méthode d'approximation par discrétisation, consistant à « remplacer des relations portant sur des fonctions continues et dérivables par un nombre fini de relations algébriques portant sur des valeurs précises prises par la fonction en un nombre fini de points sur leur ensemble de définition. ». On met ensuite la solution sous forme matricielle. Le but est d'obtenir l'approximation la plus précise de la solution représentant le mouvement.

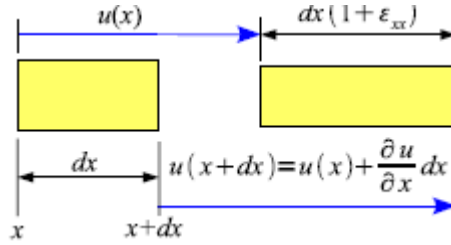
1 Obtention d'une équation différentielle précise

1.1 Définition de l'objet d'étude et des paramètres influençant les caractéristiques de l'effort produisant la déformation

On définit l'objet subissant la déformation. Considérons une poutre d'axe x , de section de droite A , de longueur dx . Un effort normal $N(x; t)$ est exercé sur la poutre.



Cet effort a pour conséquence le déplacement et l'allongement de la poutre. Tout d'abord, on définit $u(x, t)$ le champ de déplacement relatif au temps et à la position x , pour tout x appartenant à $[x_0, x_E]$ sur l'axe x .



Ensuite, on sait que l'allongement est lié au champ de déplacement puisque l'écart entre la position finale engendrée par $u(x)$ et $u(x + dx)$ est non nul si la pièce s'est allongée.

On note alors l'allongement unitaire ε tel que :

$$\varepsilon = \frac{u(x + dx) - u(x)}{dx} = \frac{\partial u}{\partial x} \quad (1)$$

De plus, l'allongement unitaire ε est la conséquence de la force normale exercée sur la surface. D'après la loi de Hooke, soit σ la contrainte normale et E le module de young

On a $\sigma = \varepsilon E$ avec $\sigma = \frac{N(x,t)}{A}$

L'allongement unitaire ε est en plus dû à la capacité du matériau à se dilater ainsi qu'à la variation de la température due à l'effort.

On pose alors $\varepsilon_{th} = \alpha \Delta T$ avec α le coefficient de dilation

Finalement, on a

$$\varepsilon = \frac{\sigma}{E} + \alpha \Delta T \quad (2)$$

De l'expression de l'allongement unitaire ε , on obtient l'expression de la contrainte σ telle que :

$$\sigma = \varepsilon E - \varepsilon_{th} E = (\varepsilon - \varepsilon_{th}) E \quad (3)$$

Ce qui nous donne finalement une expression de $N(x, t)$ nécessaire à la résolution de l'équation différentielle ultérieure, telle que

$$N(x, t) = A(\varepsilon - \varepsilon_{th}) E \quad (4)$$

1.2 Application du principe fondamental de la dynamique (PFD)

On effectue ensuite un PFD, on obtient l'équation suivante, avec p_x la force linéique portée par la poutre

$$-N(x, t) + N(x + dx, t) + p_x dx = \rho A \frac{\partial^2 u}{\partial t^2} dx \quad (5)$$

avec ρ la masse volumique

Sachant que $N(x + dx, t) = N(x, t) + \frac{\partial N(x,t)}{\partial x} dx$

On obtient finalement :

$$\frac{\partial N(x, t)}{\partial x} + p_x = \rho A \frac{\partial^2 u}{\partial t^2} \quad (6)$$

1.3 Amélioration de l'expression différentielle par la méthode des résidus pondérés

On introduit la variable $r(u)$ appelée résidu et vérifiant l'équation suivante :

$$r(u) = -\frac{\partial N(x, t)}{\partial x} - p_x + \rho A \frac{\partial^2 u}{\partial t^2} \quad (7)$$

l'équation est vérifiée lorsque $r(u) = 0$

En remplaçant $N(x, t)$ par σA , on obtient :

$$r(u) = -\frac{\partial \sigma A}{\partial x} - p_x + \rho A \frac{\partial^2 u}{\partial t^2} \quad (8)$$

On multiplie ensuite chaque membre de l'inégalité par une fonction dit « arbitraire » $u^*(x, t)$ et l'on intègre telle que :

$$W(u; u^*) = \int_{x_0}^{x_E} u^* r dx = \int_{x_0}^{x_E} u^* \left(-\frac{\partial \sigma A}{\partial x} - p_x + \rho A \frac{\partial^2 u}{\partial t^2} \right) dx \quad (9)$$

On effectue une intégration par partie telle que :

$$\int_{x_0}^{x_E} u^* \frac{\partial \sigma A}{\partial x} dx = [u^* \sigma A]_{x_0}^{x_E} - \int_{x_0}^{x_E} \varepsilon^* \sigma A dx \quad (10)$$

avec $\varepsilon^* = \frac{\partial u^*}{\partial x}$

On obtient :

$$W(u; u^*) = -[u^* \sigma A]_{x_0}^{x_E} + \int_{x_0}^{x_E} \varepsilon^* \sigma A dx - \int_{x_0}^{x_E} u^* p_x dx + \int_{x_0}^{x_E} u^* \rho A \frac{\partial^2 u}{\partial t^2} dx \quad (11)$$

Plus précisément :

$$W(u; u^*) = -[u^* \sigma A]_{x_0}^{x_E} + \int_{x_0}^{x_E} \varepsilon^* (\varepsilon - \alpha \Delta T) E A dx - \int_{x_0}^{x_E} u^* p_x dx + \int_{x_0}^{x_E} u^* \rho A \frac{\partial^2 u}{\partial t^2} dx = 0 \quad (12)$$

On appelle cette expression « la forme intégrale faible d'un problème d'élasticité ».

2 Démarche de résolution de l'équation différentielle trouvée par discrétisation

La méthode de discrétisation consiste à diviser la pièce en éléments finis. Ces éléments sont reliés entre eux par des noeuds. La pièce devient « un maillage ».

Chaque élément fini e à son propre champ de déplacement $u^e(x, t)$ où chaque noeud i à son propre vecteur $u_i^e(x, t)$ inclu dans le champ $u^e(x, t)$.

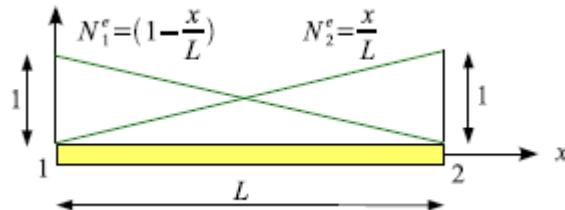
Ce qui permet, à toute position x , d'y associer un déplacement $u(x, t)$ par interpolation en connaissant les valeurs des vecteurs de déplacement sur les noeuds à proximité.

Le champ de déplacement $u^e(x, t)$ dans l'élément e a pour expression :

$$u^e(x; t) = [N_1^e(x) \quad \dots \quad N_i^e(x) \quad \dots \quad N_{n^e}^e(x)] \begin{Bmatrix} u_1^e(t) \\ \vdots \\ u_i^e(t) \\ \vdots \\ u_{n^e}^e(t) \end{Bmatrix} = [N^e(x)] \{u^e(t)\}$$

Où $N_i^e(x)$ sont les fonctions d'interpolation élémentaire.

Par exemple, si on prends un élément à deux noeuds avec :



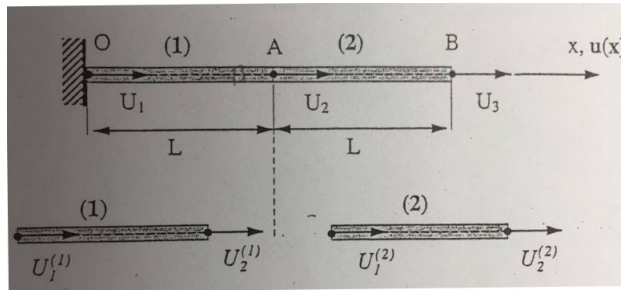
On sait que pour $x=1$, $u^e(x, t) = u_1^e(t)$ et pour $x=2$ $u^e(x, t) = u_2^e(t)$, on choisit alors deux fonctions d'interpolation $N_1^e(x) = 1 - \frac{x}{L}$ et $N_2^e(x, t) = \frac{x}{L}$ de sorte que, pour tout x dans l'intervalle L , $u(x, t)$ peut être calculé en fonction de sa proximité avec $u_1^e(x)$ et $u_2^e(x)$. Par exemple si x est proche du premier noeud subissant le déplacement $u_1^e(x)$, $N_1^e(x)$ sera grand et $N_2^e(x, t)$ faible, donc $u(x, t)$ s'exprimera proportionnellement à $u_1^e(t)$. En effet, on a $u^e(x, t) = N_1^e(x)u_1^e(t) + N_2^e(x, t)u_2^e(t)$. Le déplacement est donc défini en tout point de l'élément fini.

On peut alors effectuer une « représentation globale » de toute la pièce en définissant $u(x, t)$ en fonction des vecteurs déplacement de chaque noeud définissant toute la pièce et de leur fonction précise d'interpolation. La somme des fonctions d'interpolation doit toujours être égale à 1.

3 Résolution de l'équation par expression matricielle

Grâce à la méthode par discrétisation, on connaît une expression de $u(x, t)$ en fonction de valeur de déplacement connu. On remplace donc dans l'expression les variables par des matrices définissant ces variables. L'intégration revient à intégrer les fonctions d'interpolation. On obtient un système linéaire à résoudre, et une expression de $u(x, t)$ en fonction de termes connus. On a alors fait une approximation la plus précise possible de $u(x, t)$ numériquement.

4 Exemple



4.1 PFD

$$-N(x) + N(x + dx) + p(x) = 0$$

$$\frac{dN}{dx} + p(x) = 0$$

4.2 Loi de comportement en élasticité

$$\sigma = E * \xi$$

ξ c'est la déformation E le module de Young

$$\xi = \frac{\Delta l}{l_0} = \frac{l - l_0}{l_0} = \frac{u(x + dx) - u(x)}{dx} = \frac{du}{dx}$$

$$N = ES\xi = ES \frac{du}{dx}$$

4.3 Condition limite

$$u(x = 0) = 0$$

$$N(x = 0) = -2LgS$$

4.4 Forme integrale

Expression du résidu :

$$r(u) = \frac{dN}{dx} + p(x) = 0$$

$$w(u, u^*) = \int_0^{2L} u^* \left(\frac{dN}{dx} + p(x) \right) dx$$

$$w(u, u^*) = \int_0^{2L} u^* \left(S \frac{d\sigma}{dx} + p(x) \right) dx = \int_0^{2L} \left(u^* S \frac{d\sigma}{dx} \right) dx + \int_0^{2L} u^* p(x) dx$$

$$w(u, u^*) = [u^* S \sigma_x]_0^{2L} - \int_0^{2L} \frac{du^*}{dx} S \sigma dx + \int_0^{2L} u^* p(x) dx$$

$$w(u, u^*) = [u^* S \sigma_x]_0^{2L} - \int_0^{2L} \xi^* E S \xi dx + \int_0^{2L} u^* p(x) dx$$

$$u^{(1)}(x) = N_1^{(1)}(x) u_1^{(1)} + N_2^{(1)}(x) u_2^{(1)}$$

avec :

$$-N_1(x) = 1 - \frac{x}{L}$$

$$-N_2(x) = \frac{x}{L}$$

$$\text{si } x = 0 \quad u^{(1)}(0) = 1 u_1^{(1)}$$

$$\text{si } x = 1 \quad u^{(1)}(1) = u_2^{(1)}$$

$$u'(x) = \begin{bmatrix} N_1^{(1)} & N_2^{(2)} \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \end{bmatrix} = [N]^{(1)} [U]$$

$$u^{*(1)} = [N]^{(1)} [U^*]$$

$$\xi = \frac{du}{dx} = \begin{bmatrix} \frac{dN_1^{(1)}}{dx} & \frac{dN_2^{(1)}}{dx} \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \end{bmatrix} = [B]^{(1)} [U]^{(1)}$$

$$\xi^* = [B]^{(1)} [u^*]^{(1)}$$

4.5 Ecriture du résidu du premier élément

$$I_1 = \int_0^L \xi^* SE \xi dx = SE \int_0^L \xi^* \xi dx$$

$$I_1 = SE \int_0^L [B]^{T(1)} [U^*]^{(1)T} [B]^{(1)} [U]^{(1)} dx$$

$$I_1 = SE [U^*]^{(1)T} [U]^{(1)} \int_0^L [B]^{T(1)} [B]^{(1)} dx$$

$$N_1(x) = 1 - \frac{x}{L} \frac{dN_1}{dx} = -\frac{1}{L}$$

$$N_2(x) = \frac{x}{L} \frac{dN_2}{dx} = \frac{1}{L}$$

$$B^T \quad B = \begin{bmatrix} \frac{1}{L^2} & -\frac{1}{L^2} \\ -\frac{1}{L^2} & \frac{1}{L^2} \end{bmatrix}$$

$$I_1 = SE [U^*]^{(1)T} \{ U \}^{(1)} \int_0^L \begin{bmatrix} \frac{1}{L^2} & -\frac{1}{L^2} \\ -\frac{1}{L^2} & \frac{1}{L^2} \end{bmatrix} dx$$

$$I_1 = \frac{SE}{L} [U^*]^{(1)T} [U]^{(1)} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$\int_0^L [U^*]^{(1)T} [N]^{(1)T} p(x) dx = [u^*]^{(1)T} \int_0^L [N]^{T(1)} p(x) dx$$

element 1

$$K^{(1)} = \frac{ES}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$F^* = \int_0^L [N]^{(1)T} p(x) dx = \int_0^L \begin{bmatrix} 1 - \frac{x}{L} \\ \frac{x}{L} \end{bmatrix} gSL dx = gS \int_0^L \begin{bmatrix} \frac{L}{2} \\ \frac{L}{2} \end{bmatrix} = \frac{gSL\sigma}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

element 2

$$K^{(2)} = \frac{ES}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$F^{(2)} = \frac{gSL\sigma}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$K = \begin{bmatrix} [K^{(1)}] & 0 \\ 0 & [K^{(2)}] \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$[u^* S\sigma_x]_0^2$ condition aux limites sur la structure
pour l'element 1

$$[u^* S\sigma_x]_0^1 = [U^*]^{(1)T} [B]^{(1)T} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

$$R = 2LSg\sigma$$

$$B = \begin{bmatrix} -\frac{1}{L} \\ \frac{1}{L} \end{bmatrix}$$

$$[u^* S\sigma_x]_L^{2L} = [U^*]^{(1)T} [B]^{(2)T} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

pour la structure $[U^*]^T [B]^{(2)T} \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix}$

$$w(u, u^*) = [U^*]^T (-K [U] + [F] + \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix}) = 0$$

;

$$w(u, u^*) = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \frac{SLg\sigma}{2} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 2SLg\sigma \\ 0 \\ 0 \end{bmatrix}$$

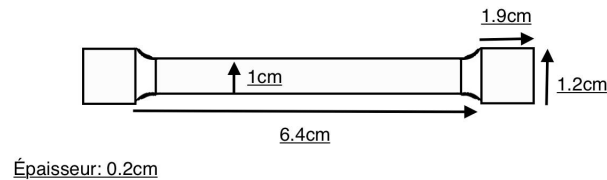
avec $u_1 = 0$ car la barre est bloquée vers le haut.

Deuxième partie

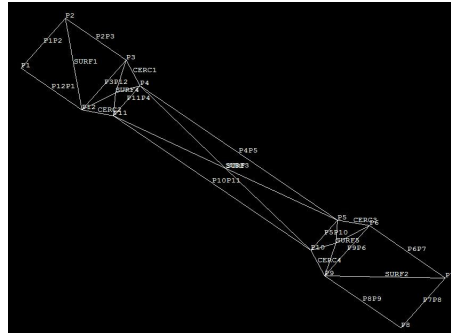
Notice d'utilisation simplifiée de Cast3m

5 Déclaration des points

Il faut premièrement déclarer les divers points importants de la pièce avec des coordonnées prises dans un repère choisi au préalable. Pour cela, nous avons du prendre des mesures précises de la pièce afin de réduire l'erreur commise :



6 Choix du maillage et traçage des droites



Les variables NELEM sont utilisées ici pour définir le nombre d'éléments de maillage dans les diverses parties de la pièce. Plus la déformation est importante ou intéressante, plus le nombre d'élément de maillage est élevé. Afin de tracer la pièce sur le logiciel, nous l'avons découpé en cinq parties géométriques faciles à déclarer. Nous avons donc deux carrés, un rectangle et deux trapèzes. L'instruction de traçage est construite de la manière suivante :

$$P1P2 = P1 \text{ DROITE NELEM1 } P2 ;$$

P1P2 est le nom du segment, choisi explicitement afin d'avoir un code maintenable et simple à comprendre. P1 DROITE indique le fait que le logiciel doit

se placer aux coordonnées de P1 déclarées précédemment et qu'il doit tracer une droite ou un segment de droite. Enfin NELEM1 P2 signifie qu'il doit mettre NELEM1 éléments de maillage sur cette section et que le segment s'arrête au point P2. Nous avons donc fait ceci pour tous les autres points afin de tracer la pièce entière. Cependant, il faut coder le traçage des pièces dans le même sens sinon le logiciel n'interprétera pas correctement les instructions.

7 Surfaces

Une fois les instructions des droites écrites, il faut déclarer des surfaces toujours en gardant le même sens de codage :

```
SURF1 = SURF PLAN (P1P2 ET P2P3 ET P3P12 ET P12P1)
```

SURF1 est le nom de la variable utilisé ici, SURF PLAN sont deux opérateurs qui indiquent au logiciel qu'il faut tracer une surface plane. Ceci est suivi des points qui constituent la surface.

8 Conditions limites

Suite au traçage complet de la pièce, il faut déclarer les conditions limites de la pièce :

```
CL1 = BLOQ SURF1 DEPL ROTA
```

CL1 est le nom de la variable qui stocke la première condition limite sur la surface SURF1. Cette condition bloque BLOQ la rotation ROTA et le déplacement DEPL de la surface concernée. Nous avons également une CL2 qui bloque la SURF2 en rotation et sur l'axe Ux et Uz afin d'avoir une déformation logique. Finalement, une astuce que nous avons utilisé ici, c'est de déclarer une variable NT qui réunit toutes les conditions limites afin de clarifier le code par la suite.

9 Définition du modèle

Dans cette partie, il faut définir les caractéristiques de la pièce et du matériau utilisé. C'est-à-dire qu'il faut indiquer le type de mécanique voulu, le module d'Young, le coefficient de Poisson et l'épaisseur :

```
M1 = MODE SURF MECANIQUE ELASTIQUE DKT;
```

```
MAT1 = MATE M1 YOUN 2.1e11 NU 0.3;
```

```
CAR1 = CARA M1 EPAI 0.2;
```

Avec l'élément DKT qui correspond au *Discrete Kirchhoff Triangle*.

10 Définition du mouvement

Dans cette partie, l'utilisateur a le choix entre utiliser un déplacement imposé et appliquer une force. Dans cet exemple nous utiliserons un déplacement imposé au système.

```
DEPL1 = DEPI CL4 1. ;
```

On indique donc après l'opérateur DEPI, l'endroit où l'on veut que déplacement soit imposé, puis sa valeur. Suite à cela, nous avons paramétré différentes variables afin de tracer la courbe de traction sous forme d'un objet évolution :

```
LIST1 = PROG 0. 1. ;
```

```
LISF1 = PROG 0. 1. ;
```

```
LIS1=PROG 0. pas 0.1 1. ;
```

```
EVF1 = EVOL MANU 'TEMPS' LIST1 'DEPL' LISF1 ;
```

```
CHA1 = CHARG 'MECA' DEPL1 EVF1 ;
```

```
*TRAC EVF1* ;
```

11 Calcul pas à pas

L'étape du calcul pas à pas équivaut à la résolution du calcul non linéaire. Pour cela, il faut commencer par déclarer un tableau TAB1 :

```
TAB1 = TABLE ;
```

Puis nous donnons les options et les données nécessaires au calcul. Ici nous aurons, par exemple, besoin des blocages mécaniques, du modèle, du chargement, des caractéristiques. . .

```
TAB1.'BLOCAGES_MECANIQUES' = NT ;
```

```
TAB1.'MODELE' = M1 ;
```

```
TAB1.'CHARGEMENT' = CHA1 ;
```

```
TAB1.'CARACTERISTIQUES' = CM1 ;
```

```
TAB1.'TEMPS_CALCULES' = LIS1 ;
```

```
TAB1.'TEMPS_SAUVES' = LIS1 ;
```

```
TAB1.'GRANDS_DEPLACEMENTS' = FAUX ;
```

```
PASAPAS TAB1 ;
```

12 Post traitement

C'est l'étape qui est la plus délicate car c'est celle qui permettra de récupérer et d'exploiter les données de l'essai de traction. En effet nous avons eu quelque difficultés à réaliser le post traitement, nous avons eu besoin de l'aide du professeur chargé du projet notamment à cause du manque d'informations que l'on pouvait trouver sur internet concernant cette étape du code.

Tout d'abord voici une explication détaillée du post traitement sur notre essai en déformation élastique :

Il faut définir le déplacement imposé à notre géométrie totale, c'est à dire orienter et définir la déformation de notre éprouvette en fonction de notre essai et de nos besoins. Après cela on définit les déformations (opérateur EPSS = calcul des déformations pour chaque modèle élémentaire) : -la déformation totale -la déformation transversale -la déformation en cisaillement, celle ci ne fonctionne pas dans notre cas.

Longitudinal

EPSS1 = EXCO EPSS EPS1 ;

Transversal

EPTT1 = EXCO EPTT EPS1 ;

L'opérateur EXCO crée a partir d'un champ, un champ de même type en extrayant une ou plusieurs composantes données.

Après, SIGMA correspond à la contrainte et il faut donc la définir sur toute la pièce, en longitudinal et en transversal. On va ensuite tracer tout ce qu'on a défini pour obtenir un résultat exploitable.

Le but est de récupérer toutes les données, c'est à dire pas à pas la valeur des contraintes et des déformations pour chaque déplacement, on a pour cela défini une boucle dans notre code. Lorsque le programme termine l'exécution, toutes les valeurs sont transmises dans un tableau qui va nous permettre l'exploitation.

Le logiciel affiche aussi sous forme d'un code couleur les contraintes et les déformations avec leurs valeurs en coloriant la pièce, avec une échelle de couleurs correspondant aux valeurs. En traçant les courbes on peut alors récupérer les informations dont on a besoin. C'est le principe de la simulation sous logiciel.

Enfin nous avons eu besoin de l'aide d'un professeur, Mr Benoit Vielle, directement en interaction avec Mr Keller, pour la dernière partie du code car nous ne comprenions pas à quel point elle est indispensable, et somme difficilement capable de l'expliquer.

Pour la déformation plastique nous n'avions que peu de choses à modifier dans notre code, notamment :

- cette fois ci au lieu d'imposer un déplacement nous avons été obligés d'imposer une force
- dans la définition du modèle nous avons modifié la variable N1 afin d'indiquer qu'on travaillait dans une mécanique élastique plastique.

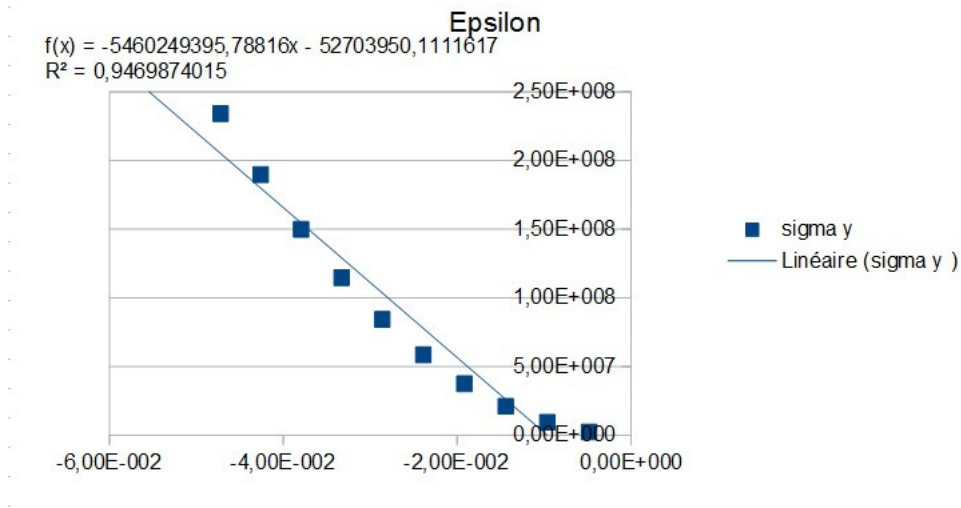
- un nouveau coefficient rentre en compte dans la variable MAT1, noté H qui correspond au *paramètre d'écroissage*, c'est le durcissement d'un métal sous l'effet de sa déformation plastique.

13 Critique des résultats

Les résultats que nous avons obtenus ne reflètent pas la réalité et ne correspondent malheureusement pas à nos attentes. Effectivement, nous obtenons les graphiques suivant :

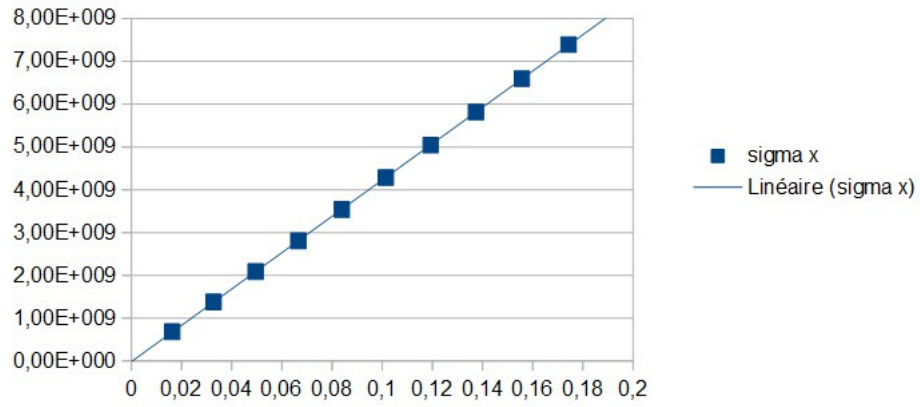
Pour l'élasticité :

On peut normalement de manière graphique le module d'Young à la rupture, la courbe obtenue étant fautive nous ne pouvons pas fournir une bonne valeur. Par ailleurs le programme mettait énormément de temps à s'exécuter, il nous aurait donc fallu un ordinateur plus puissant pour observer une rupture, malgré le fait que des optimisations auraient pu être faites dans le code du programme, qui pourtant à été réalisé à l'aide d'un professeur. Nous ne cernons donc pas bien pourquoi cela n'a pas fonctionné.



$$f(x) = 42440439420,2917x - 14310278,3468315 \text{ Ep}$$

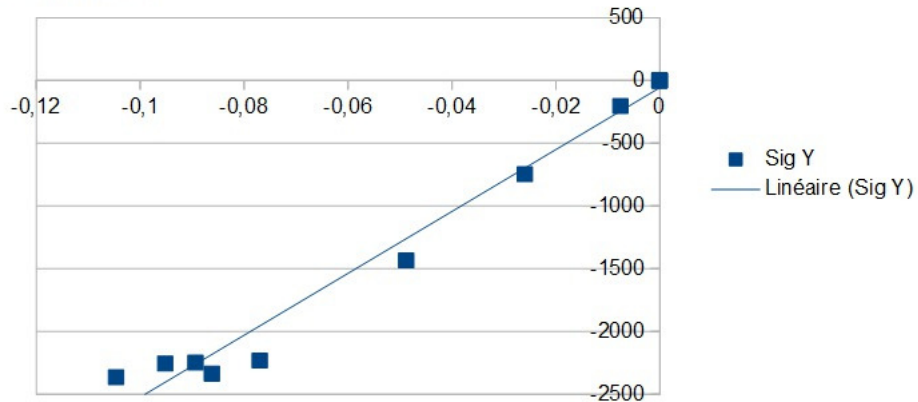
$$R^2 = 0,9999952003$$

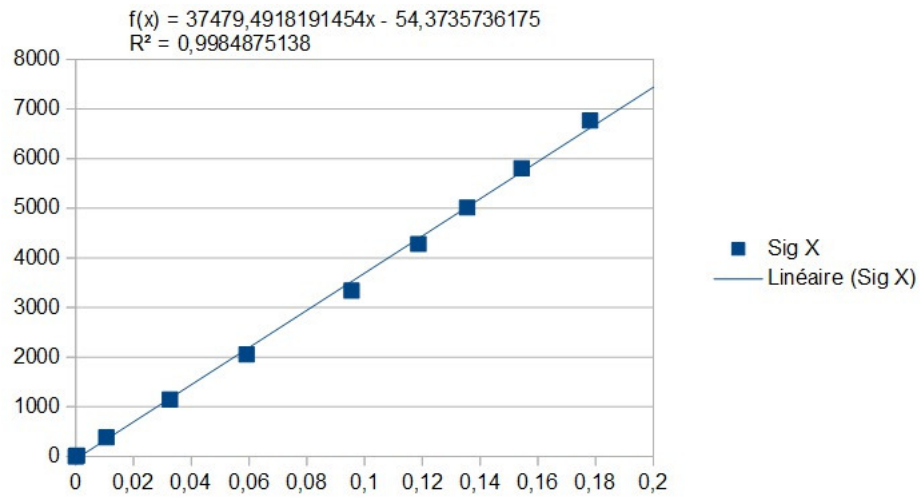


Pour la plasticité : Le coefficient directeur de la droite est censé correspondre au module d'Young, malheureusement on voit bien que sa valeur n'est pas du tout conforme à la valeur théorique.

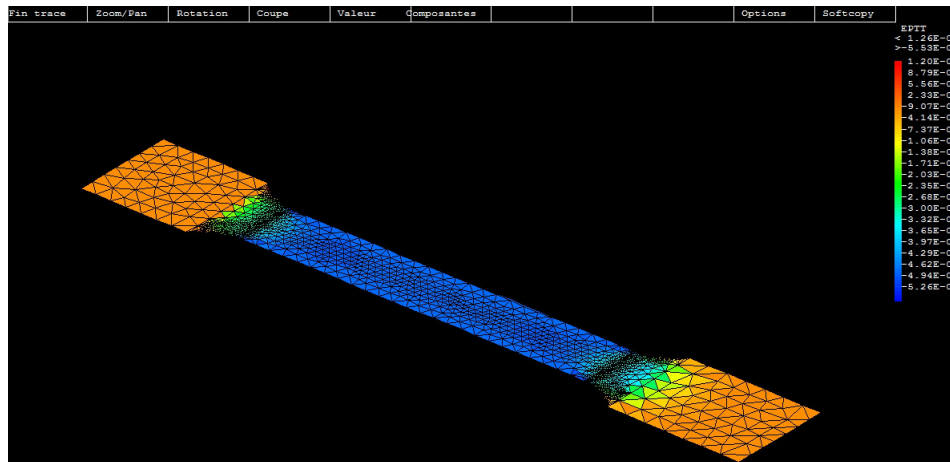
$$f(x) = 24661,2750934932x - 57,2079090486 \text{ E}$$

$$R^2 = 0,9789818442$$

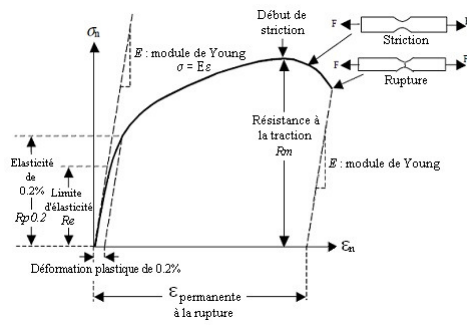




Voici un exemple de graphique couleur que nous obtenons à la fin de l'exécution du programme :



Ainsi nous pouvons donc voir que ces résultats ne sont pas cohérents avec la théorie. En appliquant les méthodes indiquées par notre professeur nous ne retrouvons pas la bonne valeur du module d'Young. Nous avons donc trouvé un graphique présentant une courbe que nous aurions du obtenir suite à l'exécution du programme :



Conclusion :

Dès le début du projet, notre professeur Monsieur Keller et nous mêmes avons choisi de répartir le travail de la manière suivante : Claire et Sinwen allaient s'occuper essentiellement de la méthode mathématique des éléments finis, la comprendre en profondeur, ce choix correspondait au fait qu'ils désiraient faire Génie Mathématique l'année prochaine. Tandis que Tom et Jules allaient s'occuper de la partie simulation sur le logiciel Cast3m puisqu'ils désiraient faire ASI et Perf-II. Nous avons mis un peu de temps à comprendre le fonctionnement du logiciel qui utilise un langage de codage spécifique, et malheureusement à part une notice d'utilisation disponible sur internet, qui n'était pas assez complète et pas spécialement bien expliquée, nous n'avons pas beaucoup d'informations car à l'INSA un seul professeur sait se servir de Cast3m. Ce fut la principale difficulté de ce projet, nous avons souvent dû faire appel à l'aide de Mr Vielle par l'intermédiaire de Mr Keller, car ils sont collègues. Malheureusement les résultats obtenus sont très éloignés de ce qu'on espérait, c'est un peu décevant certes, mais cela n'enlève rien au fait que nous avons beaucoup appris. En effet cela s'apparentait presque à de la recherche pour nous puisque nous n'avons aucune connaissances concernant le thème abordé.

Jules Calmon :

Pour ma part, de nombreux points positifs sont à souligner après coup sur ce projet. Tout d'abord il est important et formateur pour un élève ingénieur de travailler en équipe, puisqu'il sera amené à le faire dans son métier. Ensuite malgré des résultats décevant concernant notre partie, nous ne sommes pas découragés pour autant et ce pourrait être un point d'amélioration que de réussir la bonne simulation d'un essai de traction. Nous avons découvert un domaine que nous ne connaissions pas et avons ainsi enrichi notre culture scientifique. Par ailleurs se heurter à l'échec avant d'entamer sa vie professionnelle est préparateur. J'ai apprécié l'essai de traction réalisé en salle de TP MECA car on a pu voir une application directe et concrète du projet.

Tom Masson :

Finalement, ce projet m'a permis d'apprendre différentes choses telle que le travail de recherche en équipe, la communication ainsi qu'une méthode afin de simuler un essai de traction numériquement. Effectivement, jusqu'à maintenant, nous avons déjà réalisé des projets en groupe mais il n'y avait presque pas de recherche dans ces derniers. Pendant ce projet, nous avons du en quelques sortes, être autodidactes avec l'aide du professeur bien entendu. Ce projet a été très formateur pour moi puisqu'il m'a montré qu'il faut bien prendre en compte toutes les variables avant de commencer le projet notamment en ce qui concerne le logiciel utilisé.

Claire Driguez :

Ce projet m'a permis de découvrir la méthode des éléments finis et une de ses applications. J'ai trouvé très intéressant de découvrir une méthode mathématique utilisée par de nombreux logiciels de simulation numérique. Comprendre la méthode des éléments finis a été très enrichissant, car cela m'a permis de mobi-

liser mes connaissances mathématiques théoriques, les appliquer à un problème concret par l'intermédiaire de la résolution d'équations différentielles.

Sinwen Mohamed Stambouli :

Le choix de ce projet m'as permis de développer mon esprit de cohésion et ma capacité à travailler en groupe, en effet ce projet m'a permis de concrétiser un cursus scolaire qui était jusque là souvent abstrait. La découverte de la méthode des éléments finis et des mathématiques qui se cachaient derrière est l'un des points fort de ce projet. Finalement, même si on a pas réussi à obtenir les résultat souhaiter il n'en demeure pas moins vrai que cette expérience fût enrichissante.

Références

- [1] http://iut.univ-lemans.fr/ydlogi/cours/mef_elas_1d.pdf
- [2] <http://www.larousse.fr/dictionnaires/francais/discr%C3%A9tisation/25871>
- [3] [http://www.mecastyle.com/image/images/courbe%20traction\(1\).jpg](http://www.mecastyle.com/image/images/courbe%20traction(1).jpg)

Annexe : Code du projet pour Cast3m (Plasticité)

```
* fichier : plastique_final.dgibi
*
*-----
* Simulation numerique par elements finis
* d'un essai de traction - Cast3m
*-----
*
OPTI ECHO 0;
TITRE 'P6 - Maillage non uniforme - Courbure';
OPTI DIME 3 ELEM SEG2;
```

```
*****
* Points *
*****
P1 = 0. 0. 0.;
P2 = 0. 1.5 0.;
P3 = 1.6 1.5 0.;
P4 = 2.2 1.2 0.;
P5 = 7.4 1.2 0.;
P6 = 8. 1.5 0.;
P7 = 10. 1.5 0.;
P8 = 10. 0. 0.;
P9 = 8. 0. 0.;
P10 = 7.4 0.3 0.;
P11 = 2.2 0.3 0.;
P12 = 1.6 0. 0.;
P13 = 2.2 2.4 0.;
P14 = 2.2 -0.9 0.;
P15 = 7.4 2.4 0.;
P16 = 7.4 -0.9 0.;
```

```
OEIL1 = 0. 0. 100.;
```

```
*****
* Maillage *
*****
NELEM1 = 7;
NELEM2 = 25;
```



```

NELEM3 = 16;
NELEM4 = 40;
*variables déclarées pour modifier simplement le maillage

```

```

*****
*           Carré 1           *
*****
P1P2 = P1 DROITE NELEM1 P2;
P2P3 = P2 DROITE NELEM1 P3;
P3P12 = P3 DROITE NELEM1 P12;
P12P1 = P12 DROITE NELEM1 P1;

```

```

*****
*           Carré 2           *
*****
P6P7 = P6 DROITE NELEM1 P7;
P7P8 = P7 DROITE NELEM1 P8;
P8P9 = P8 DROITE NELEM1 P9;
P9P6 = P9 DROITE NELEM1 P6;

```

```

*****
*           Réctangle 3       *
*****
P4P5 = P4 DROITE NELEM2 P5;
P5P10 = P5 DROITE NELEM2 P10;
P10P11 = P10 DROITE NELEM2 P11;
P11P4 = P11 DROITE NELEM2 P4;

```

```

*****
*           Trapèze 4         *
*****
CERC1 = CERC NELEM4 P4 P13 P3;
CERC2 = CERC NELEM4 P12 P14 P11;

```

```

*****
*           Trapèze 5         *
*****
CERC3 = CERC NELEM4 P6 P15 P5;
CERC4 = CERC NELEM4 P9 P16 P10;

```

```

*****
*           Surfaces          *
*****
SURF1 = SURF PLAN (P1P2 ET P2P3 ET P3P12 ET P12P1);
SURF2 = SURF PLAN (P6P7 ET P7P8 ET P8P9 ET P9P6);
SURF3 = SURF PLAN (P4P5 ET P5P10 ET P10P11 ET P11P4);

```

```

SURF4 = SURF PLAN (CERC1 ET P11P4 ET CERC2 ET P3P12);
SURF5 = SURF PLAN (CERC4 ET P9P6 ET CERC3 ET P5P10);

SURF = SURF1 ET SURF2 ET SURF3 ET SURF4 ET SURF5;

TRACE SURF;

*****
*      Conditions limites      *
*****
CL1 = BLOQ SURF1 DEPL ROTA;
CL2 = BLOQ SURF2 ROTA;
CL3 = BLOQ SURF2 UZ;
NT = CL1 ET CL2 ET CL3;

*****
*      Définition Modele      *
*****
M1  = MODE SURF MECANIQUE ELASTIQUE PLASTIQUE CINEMATIQUE DKT;
MAT1 = MATE M1 YOUN 210E3 NU 0.3 SIGY 100 H 2000;
CAR1 = CARA M1 EPAI 0.2;

CM1 = MAT1 ET CAR1;

* DEPLACEMENT IMPOSE
*DEPL1 = DEPI CL4 0.5;
* contrainte imposée = 300 MPa
DEPL1 = force (72. 0. 0.) P9P6;

LIST1 = PROG 0. 1.;
LISF1 = PROG 0. 1.;

LIS1=PROG 0. pas 0.1 1.;

EVF1 = EVOL MANU 'TEMPS' LIST1 'DEPL' LISF1;
CHA1 = CHARG 'MECA' DEPL1 EVF1;
*TRAC EVF1*;

*****
*      CALCUL PAS A PAS      *
*****
TAB1 = TABLE;
TAB1. 'BLOCAGES_MECANIQUES' = NT;
TAB1. 'MODELE' = M1;
TAB1. 'CHARGEMENT' = CHA1;
TAB1. 'CARACTERISTIQUES' = CM1;

```

```

TAB1. 'TEMPS_CALCULES' = LIS1;
TAB1. 'TEMPS_SAUVES' = LIS1;
TAB1. 'GRANDS_DEPLACEMENTS' = FAUX;

PASAPAS TAB1;

*****
*      POST TRAITEMENT      *
*****

PAS=10;

*
GEOTOT = EXTR TAB1. 'MODELE' 'MAIL';

*-----
*  DEPLACEMENTS
*-----
*
DEP2 = TAB1. 'DEPLACEMENTS'.PAS;
*
DEF0 = DEFO GEOTOT DEP2 0. BLAN;
DEF2 = DEFO GEOTOT DEP2 1. VERT;

* definition orientation longitudinale et transversale
VT1=1. 0. 0.;
VT2=0. 1. 0.;

EPS1TOT = EPSI M1 DEP2 CM1;
EPS1     = RTENS EPS1TOT M1 VT1 VT2;
*longitudinal
EPSS1    = EXCO EPSS EPS1;
*transversal
EPTT1    = EXCO EPTT EPS1;
*cisaillement
*EPST1 = EXCO EPST EPS1;
TRAC EPSS1 M1 SURF;
TRAC EPTT1 M1 SURF;
*TRAC EPST1 M1 SURF*;

SIGTOT1 = SIGMA M1 DEP2 CM1;
SIG1     = RTENS SIGTOT1 M1 VT1 VT2;

SIGSS1 = EXCO N11 SIG1;
SIGTT1 = EXCO N22 SIG1;
SIGST1 = EXCO N12 SIG1;

```

```

TRAC SIGSS1 M1 SURF;
TRAC SIGTT1 M1 SURF;
TRAC SIGST1 M1 SURF;

i1 = 0 ;
ind1 = index (tab1 . déplacements) ;
n1 = dime ind1 ;

OPTI IMPR 'exit';

REPETER BLOC1 N1;

*
* EXTRACTION VALEURS CENTRE ZONE UTILE (ELT 685)
*

      DEPX = EXTR (TAB1.DEPLACEMENTS.I1) 'UX' P7;
      DEPY = EXTR (TAB1.DEPLACEMENTS.I1) 'UY' P7;

* extraction des valeurs dans le repère global
DEPO = TAB1.'DEPLACEMENTS'.I1;
EPS0 = EPSI M1 DEPO CM1;
EPS0 = RTENS EPS0 M1 VT1 VT2;
ep_x1 = EXTR EPS0 EPSS 1 685 1;
ep_y1 = EXTR EPS0 EPTT 1 685 1;

SIG0 = SIGMA M1 DEPO CM1;
SIG0 = RTENS SIG0 M1 VT1 VT2;
sig_x1 = EXTR SIG0 N11 1 685 1;
sig_y1 = EXTR SIG0 N22 1 685 1;

mess DEPX DEPY ep_x1 ep_y1 sig_x1 sig_y1;

i1 = i1 + 1;

fin bloc1 ;

FIN;

```