



I NSTITUT

DE R ADIOPROTECTION

ET DE S ÛRETÉ

N UCLÉAIRE

Introduction à la méthode des éléments finis

Y. MONERIE^(a), F. DUBOIS^(b), G. GEYMONAT^(b)

^(a) IRSN/DPAM/SEMCA/LEC, yann.monerie@irsn.fr

^(b) LMGC - CNRS UMR5508, dubois@lmgc.univ-montp2.fr, geymonat@lmgc.univ-montp2.fr

***Direction de la prévention des accidents majeurs
Service d'études et de modélisation du combustible en situations accidentelles
Laboratoire d'études du combustible***

Institut de Radioprotection et de Sûreté Nucléaire

Direction de la prévention des accidents majeurs

Service d'études et de modélisation du combustible en situations accidentelles

Laboratoire d'études du combustible

BP 3 13115 Saint-Paul-lez-Durance Cedex France tél. (33) 04 42 25 77 36 fax (33) 04 42 25 61 43

Introduction à la méthode des éléments finis

Y. MONERIE^(a), F. DUBOIS^(b), G. GEYMONAT^(b)

^(a) IRSN/DPAM/SEMCA/LEC, Cadarache, B.P. 13 13115 SAINT-PAUL-LEZ-DURANCE Cedex FRANCE

^(b) LMGC, Université Montpellier II, CC048, Pl. E. Bataillon, 34095 MONTPELLIER Cedex 5 FRANCE

Decembre 2004 — 106 pages

Résumé:

Ce document constitue une introduction à la mise en œuvre de la méthode des éléments finis par l'utilisation de CAST3M dans le cadre de la mécanique des solides en petites perturbations, et pour l'essentiel pour des comportements élastiques linéaires et stationnaires.

TABLE DES MATIERES

1	Introduction	5
2	Un rapide rappel sur la méthode des éléments finis	6
2.1	Généralités sur les méthodes d'approximation	6
2.2	Principes de la méthode des éléments finis	8
2.2.1	Maillage et mailles	8
2.2.2	Nœuds et éléments	9
2.2.3	Champs discrets et degrés de liberté	9
2.2.4	Fonctions d'interpolation	9
2.2.5	Discrétisation spatiale et interpolation	11
2.3	Problème approché	13
2.3.1	Forme variationnelle des problèmes d'élasticité linéaire	13
2.3.2	Propriétés de $a(u, v)$	14
2.3.3	Théorème de l'énergie potentielle en déplacement	15
2.3.4	Existence et unicité de la solution	15
2.3.5	Formulation variationnelle approchée	17
2.3.6	Stratégie MEF et propriétés de \mathbf{K}	17
2.4	Commentaires et compléments	19
2.4.1	Stockage informatique de \mathbf{K}	19
2.4.2	Les solveurs	19
2.4.3	Conditionnement de \mathbf{K}	21
2.4.4	Application des conditions aux limites	22
2.4.5	Approximation des domaines courbes	23
2.5	Un exemple simple détaillé	25
2.5.1	Solution analytique unidimensionnelle (traction pure)	25
2.5.2	Discrétisation spatiale	26
2.5.3	Formulation variationnelle par sous-domaine	26
2.5.4	Fonctions d'interpolation par sous-domaine	27
2.5.5	Discrétisation de la formulation variationnelle par sous-domaine	27
2.5.6	Matrice de rigidité et second membre élémentaires	28
2.5.7	Assemblage	29
2.5.8	Application des conditions aux limites	30
2.5.9	Post-traitement	31
3	Introduction à CAST3M	33
3.1	Carte de visite de CAST3M	33
3.2	Notions de base	34
3.2.1	Etapes du déroulement d'un calcul avec CAST3M	34
3.2.2	Le GIBIANE et ses règles syntaxiques	36
3.2.3	Exécution du programme	37
3.2.4	Système d'unités	37
3.2.5	Quelques opérateurs élémentaires	37
3.2.6	Principaux objets	38
3.3	Construction du modèle mathématique	40
3.3.1	Options générales de calcul	40
3.3.2	Discrétisation de l'espace	41
3.3.3	Manipulation d'objets géométriques	48

3.3.4	Définition du modèle de calcul	50
3.3.5	Définition des caractéristiques matériaux et structurelles	50
3.3.6	Construction des matrices de rigidité et de masse	51
3.3.7	Prise en compte des conditions aux limites	51
3.3.8	Définition de chargements	52
3.4	Post-traitement des contraintes, opérateur SIGMA	54
4	Quelques exemples d'analyse	56
4.1	Calcul élastique statique	57
4.1.1	Définition du modèle	57
4.1.2	Résolution	57
4.1.3	Post-traitement	58
4.2	Calcul de modes (Hors Programme)	59
4.3	Calcul dynamique (Hors Programme)	61
4.3.1	Réponse à une sollicitation harmonique	63
4.3.2	Superposition modale	64
4.3.3	Dynamique transitoire	68
4.3.4	Calcul non linéaires	71
4.3.5	Calcul fluide acoustique, couplage fluide structure	76
4.3.6	Calcul d'une fonction de transfert	77
4.3.7	Calcul avec contact et frottement	78
5	Travaux Dirigés	81
5.1	Premier contact avec CAST3M	81
5.2	Problème Pont à deux travées	84
5.2.1	Enoncé	84
5.2.2	Corrigé	85
5.3	Problème Poutre	89
5.3.1	Enoncé	89
5.3.2	Corrigé	91
5.4	Problème Membrane	100
5.4.1	Enoncé	100
5.4.2	Corrigé	101
5.5	Problème Etagère	103
5.5.1	Enoncé	103
5.5.2	Corrigé	104

1 Introduction

Ce document propose une introduction à la mise en œuvre de la méthode des éléments finis par l'utilisation de CAST3M. Pour l'essentiel, les applications visées se cantonnent au cadre de la mécanique stationnaire des solides élastiques linéaires sous l'hypothèse des petites perturbations.

Une attention particulière est portée à la limitation à des notions et des outils mathématiques accessibles aux élèves de seconde année d'école d'ingénieurs.

Merci d'adresser vos commentaires et remarques à :
`yann.monerie@irsn.fr`.

2 Un rapide rappel sur la méthode des éléments finis

2.1 Généralités sur les méthodes d'approximation

Les problèmes rencontrés dans le domaine des sciences pour l'ingénieur consistent le plus souvent à simuler le comportement de systèmes physiques complexes, en vue :

- de limiter le nombre de prototypes à fabriquer (contrainte de coût),
- de se substituer à l'expérimentation lorsque celle-ci n'est pas (ou peu) réalisable (domaines spatial, nucléaire).

Ces objectifs se détaillent en de multiples sous-objectifs comme par exemple :

- les problèmes de dimensionnement (résistance d'une structure à un chargement donné, choix de matériaux pour une fonctionnalité donnée, recherche d'un gain de matière, ...),
- les problèmes d'optimisation d'une fonctionnalité ou d'une forme donnée.

Une grande majorité des problèmes physiques concernés consiste à trouver, pour tout point x d'un domaine Ω et pour tout temps t dans un intervalle $[0, T]$, un champ $u(x, t)$ satisfaisant à des *équations aux dérivées partielles* (EDP) dans Ω et à des conditions aux limites sur $\partial\Omega$, la frontière de Ω .

▷ *Equations de la physique*
⇒ EDP

Pour fixer les idées, considérons un problème de mécanique des solides isotropes en élasticité linéaire :

$$\left\{ \begin{array}{ll} \operatorname{div} \sigma + f = \rho \frac{\partial^2 u}{\partial t^2} & \text{dans } [0, T] \times \Omega \quad (\text{équation d'équilibre}) \\ \sigma = \lambda (\operatorname{Tr} \varepsilon) i + 2\mu \varepsilon & \text{dans } [0, T] \times \Omega \quad (\text{loi de comportement}) \\ \varepsilon = \frac{1}{2} (\nabla u + {}^T \nabla u) & \text{dans } [0, T] \times \Omega \quad (\text{petites déformations}) \\ u = \bar{u} & \text{sur } [0, T] \times \partial\Omega_U \quad (\text{déplacements imposés}) \\ \sigma \cdot n = F & \text{sur } [0, T] \times \partial\Omega_F \quad (\text{forces imposées}) \end{array} \right. \quad (1)$$

où $u(x, t)$, $\varepsilon(x, t)$, $\sigma(x, t)$ sont respectivement le champ de déplacements, de déformations et de contraintes, $f(x, t)$ est un champ de forces volumiques, $\rho(x, t)$ la masse volumique, λ et μ les coefficients de Lamé, $\bar{u}(x, t)$ et $F(x, t)$ sont respectivement des champs de déplacement et de force donnés, $n(x, t)$ est la normale unitaire sortante à Ω , $\partial\Omega_U$ et $\partial\Omega_F$ représentent une partition de $\partial\Omega$ (i.e. vérifiant $\partial\Omega_F \cap \partial\Omega_U = \emptyset$ et $\partial\Omega_F \cup \partial\Omega_U = \partial\Omega$) et i est le tenseur unitaire. En notant que la loi de comportement en élasticité linéaire (seconde ligne du problème (1)) s'écrit aussi $\sigma(x, t) = \mathbb{C} : \varepsilon(x, t)$ où \mathbb{C} est le tenseur de rigidité et en regroupant les trois premières équations du système précédent, on obtient l'EDP relative aux problèmes de mécanique des solides en élasticité linéaire :

▷ *Un exemple d'EDP :
élasticité linéaire en
mécanique des solides*

$$\operatorname{div} \left(\frac{1}{2} \mathbb{C} : (\nabla u + {}^T \nabla u) \right) + f = \rho \frac{\partial^2 u}{\partial t^2}. \quad (2)$$

Le membre de gauche de cette équation concerne des dérivées d'espace (divergence, gradient) et le membre de droite des dérivées temporelles (d'où la détermination EDP).

Le plus souvent, les outils mathématiques ne permettent pas une résolution analytique de ces EDP. Même en supposant que l'on s'intéresse à des problèmes stationnaires (pas de dépendance en temps) ou que l'on possède par ailleurs des schémas numériques pour les résolutions en temps, le problème de la résolution en espace demeure. Des méthodes approchées doivent alors être mises en œuvre.

Une méthode d'approximation consiste à transformer les équations aux dérivées partielles en équations algébriques :

▷ *Méthode
d'approximation*

1. en utilisant une représentation discrète du champ inconnu $u(x, t)$,
2. en établissant un problème approché dont la solution est la représentation discrète du champ inconnu $u(x, t)$.

Il existe différentes méthodes d'approximation. Ce document concerne la *méthode des éléments finis (MEF)*. Les principes de cette méthode sont résumés ci-dessous où, pour simplifier, on suppose que les champs considérés ne dépendent que de la variable d'espace x (stationnaire).

2.2 Principes de la méthode des éléments finis

Le champ inconnu $u(x)$ peut avoir une forme compliquée sur le domaine Ω (penser à une fonction à variations rapides de sa courbure, voir FIG. 1). Si l'on cherche à approcher cette fonction à l'aide d'approximations simples telles que les fonctions linéaires *par morceaux*, il apparaît clairement que la précision de cette approximation dépend du nombre de morceaux choisi pour représenter Ω . Cette situation est illustrée sur la FIG. 1 : plus la fonction à approcher $u(x)$ possède d'importantes variations spatiales et plus l'approximation recherchée est simple, plus le domaine Ω de définition de la fonction doit être découpé en petits morceaux Ω_i pour que la précision de l'approximation reste convenable. Plus précisément,

L'approximation éléments finis du champ inconnu $u(x)$ consiste à l'approcher à l'aide d'un champ $u^h(x)$ défini par morceaux sur m sous-domaines Ω_i représentant une partition de Ω :

$$\bigcup_{i=1}^m \Omega_i = \overline{\Omega} \quad \text{et} \quad \mathring{\Omega}_i \cap \mathring{\Omega}_j = \emptyset \quad \forall i \neq j, \quad (3)$$

où $\overline{\Omega}$ est l'adhérence de Ω et $\mathring{\Omega}_i$ désigne l'intérieur d'un sous-domaine Ω_i .

Le champ $u^h(x)$ est alors défini par sa restriction $u_i^h(x)$ sur chaque sous-domaine Ω_i : le champ $u^h(x)$ est dit défini par morceaux.

▷ Notion d'approximation par des champs définis par morceaux

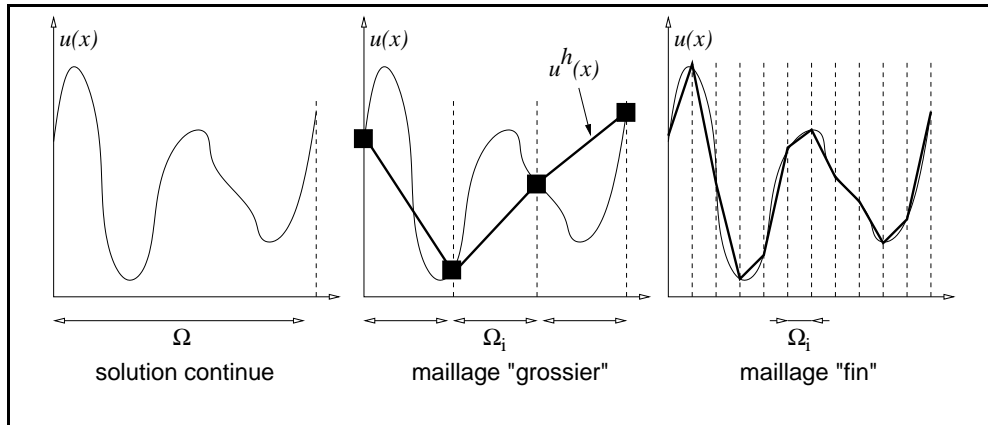


FIG. 1 – Approximation d'une fonction à l'aide de fonctions linéaires par morceaux.

2.2.1 Maillage et mailles

Cette partition de l'adhérence de Ω en un nombre fini de sous-domaines Ω_i est le *maillage* de Ω . Les sous-domaines Ω_i sont les *mailles*. Il est souvent imposé aux mailles de remplir les trois conditions suivantes :

1. chaque maille est un polyèdre,⁽¹⁾

▷ Maillage de $\Omega =$ partition de $\overline{\Omega}$

⁽¹⁾en toute rigueur, si Ω est un polyèdre

2. deux mailles distinctes sont d'intérieurs disjoints (pas de chevauchement des mailles),
3. chaque face d'une mailles est soit la face d'une autre maille soit une partie de la frontière $\partial\Omega$ (maillages conformes).

2.2.2 Nœuds et éléments

Chacun des champs $u_i^h(x)$ est défini à partir d'un ensemble de valeurs (les carrés noirs sur le schéma central de la FIG. 1) données en des points particuliers des sous-domaines Ω_i (sur la FIG. 1, il s'agit des extrémités des sous-domaines Ω_i). Ces points particuliers s'appellent les *nœuds*. Les champs $u_i^h(x)$ sont obtenus par interpolation entre ces valeurs aux nœuds (interpolation linéaire sur la FIG. 1). Cette interpolation et son sous-domaine forment l'*élément*.

▷ *Nœud = point de discrétisation de $u^h(x)$*
 ▷ *Élément = sous-domaine + interpolation*

2.2.3 Champs discrets et degrés de liberté

Les valeurs aux nœuds (les carrés noirs sur FIG. 1), notées u_k et qui permettent de définir les champs $u_i^h(x)$, constituent une représentation discrète du champ inconnu $u(x)$. Les u_k sont appelés *degrés de liberté* (DDL). Ils ne dépendent pas de la variable d'espace x ; ce ne sont pas des champs mais des scalaires. Ils sont en nombre fini N (nombre fini de mailles \Rightarrow nombre fini de nœuds) et constituent le vecteur des inconnues du système éléments finis noté $\{U\}$. L'étape de discrétisation spatiale du champ inconnu $u(x)$ peut s'illustrer ainsi :

▷ *Les DDL*

▷ *Principe de la discrétisation spatiale de $u(x)$*

$$u(x) \in \mathcal{U}_{ad} \xrightarrow{\text{discrétisation}} \{U\} = \left\{ \begin{array}{c} u_1 \\ \vdots \\ u_k \\ \vdots \\ u_N \end{array} \right\} \in \mathbb{R}^N \quad (4)$$

où \mathcal{U}_{ad} est l'ensemble des champs admissibles qui requiert que le champ inconnu $u(x)$ possède certaines propriétés mathématiques adaptées au problème. Dans ce qui suit on suppose le champ inconnu $u(x)$ suffisamment régulier.

2.2.4 Fonctions d'interpolation

Chacun des champs $u_i^h(x)$ appartient à une classe donnée de champs. Typiquement, il s'agit d'une famille de polynômes de bas degrés, comme par exemple :

- les fonctions linéaires (le champ $u^h(x)$ est alors linéaire par morceaux comme dans l'exemple FIG. 1),
- les fonctions quadratiques, etc.

La classe des champs $u_i^h(x)$ constitue l'espace des fonctions d'interpolation de l'élément. On parle d'interpolation linéaire, quadratique, etc.

La juxtaposition de ces fonctions d'interpolation des éléments engendre l'espace des fonctions d'interpolation du domaine Ω auquel appartient le champ $u^h(x)$.

▷ Fonctions d'interpolation locales

▷ Fonctions d'interpolation globales

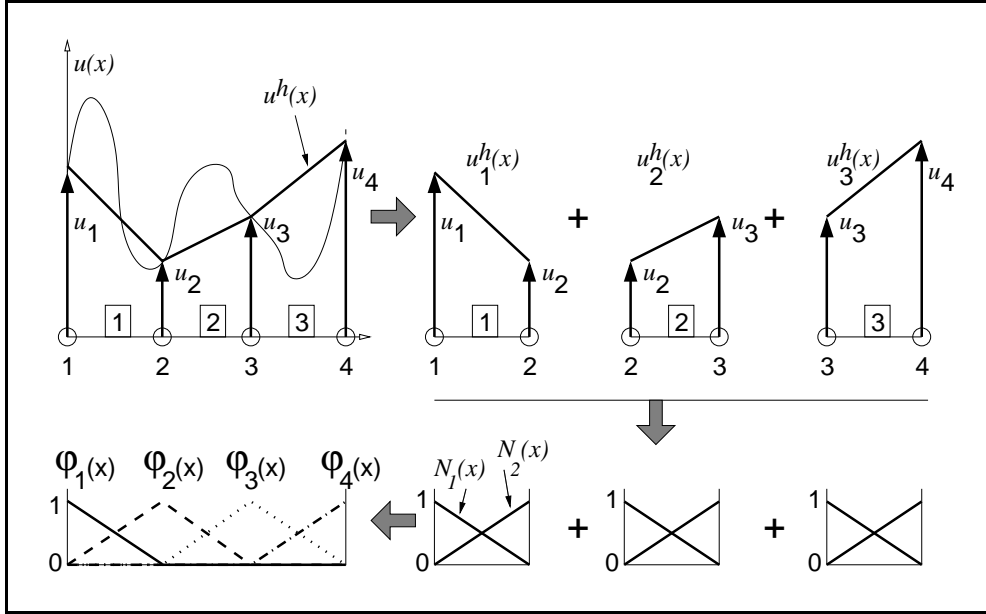


FIG. 2 – Fonctions d'interpolations des éléments locaux et fonctions d'interpolation du domaine global.

La FIG. 2 illustre ces notions. Le champ approché global $u^h(x)$ s'obtient par juxtaposition des champs locaux $u_i^h(x)$ définis sur chaque élément. Cet exemple comporte 3 éléments unidimensionnels à interpolation linéaire. La discrétisation géométrique comporte 4 nœuds, soit 4 DDL pour un champ scalaire : u_1 , u_2 , u_3 et u_4 . Les champs locaux $u_1^h(x)$, $u_2^h(x)$ et $u_3^h(x)$ sont linéaires par morceaux et chacun défini à partir de 2 valeurs u_k : u_1 et u_2 pour $u_1^h(x)$, u_2 et u_3 pour $u_2^h(x)$, u_3 et u_4 pour $u_3^h(x)$. De telles fonctions peuvent se définir à l'aide de 2 polynômes d'ordre 1 prenant chacun la valeur "1" en un nœud et la valeur "0" à l'autre nœud. Les deux fonctions d'interpolation par éléments $N_1(x)$ et $N_2(x)$ conviennent. On a alors :

$$\begin{aligned} u_1^h(x) &= N_1(x)u_1 + N_2(x)u_2, \\ u_2^h(x) &= N_1(x)u_2 + N_2(x)u_3, \\ u_3^h(x) &= N_1(x)u_3 + N_2(x)u_4. \end{aligned}$$

Ce que l'on note de manière générique, pour chaque élément e :

▷ Interpolation locale

$$u_e^h(x) = \langle N_e(x) \rangle \{U_e\}. \quad (5)$$

La juxtaposition de ces fonctions d'interpolation par élément engendre les fonctions d'interpolation du domaine Ω notée $\varphi_1(x)$, \dots , $\varphi_4(x)$. Il s'agit de

fonctions “triangulaires”, chaque $\varphi_i(x)$ ayant la valeur “1” au nœud i et la valeur “0” aux autres nœuds. Le champ global $u^h(x)$ peut alors être construit comme suit :

$$u^h(x) = \varphi_1(x)u_1 + \varphi_2(x)u_2 + \varphi_3(x)u_3 + \varphi_4(x)u_4.$$

Soit, en notation condensée :

▷ Interpolation globale

$$u^h(x) = \langle \varphi(x) \rangle \{U\}. \quad (6)$$

2.2.5 Discrétisation spatiale et interpolation

La qualité de l'approximation du champ inconnu $u(x)$ à l'aide du champ $u^h(x)$ dépend donc du nombre de sous-domaines Ω_i (la discrétisation) et du type d'interpolation utilisée sur chaque élément (le degré d'interpolation dans le cas des polynômes). Deux stratégies peuvent être envisagées pour obtenir la même qualité d'approximation de $u(x)$:

1. une discrétisation grossière (peu de sous-domaines) et une interpolation de degré suffisant pour approcher convenablement le champ inconnu $u(x)$ sur chaque sous-domaine : stratégie notée dl,
2. une discrétisation fine (beaucoup de sous-domaines) et une interpolation de bas degré (interpolation linéaire, par exemple) : stratégie notée iD.

Le plus souvent, la méthode des éléments finis est une stratégie de type iD en invoquant des interpolations de degré suffisant pour garantir une continuité de la solution approchée $u^h(x)$.

Cependant, c'est bien le rôle de l'utilisateur de la méthode des éléments finis d'adapter le compromis discrétisation/interpolation au problème traité. Ce compromis est illustré sur l'exemple donné FIG. 3 d'une poutre de longueur L sollicitée en flexion (encastée à une extrémité et soumise à une force à l'autre extrémité).

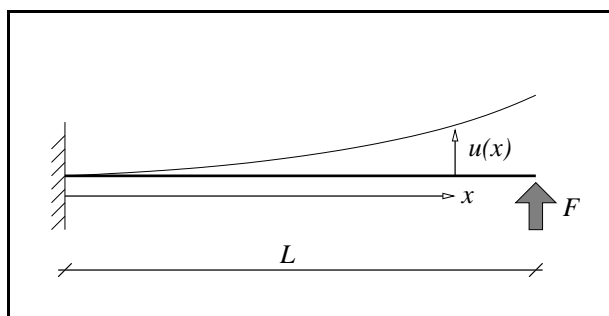


FIG. 3 – Poutre en flexion.

Le champ inconnu $u(x)$ est ici la flèche de la poutre déformée, proportionnelle à la quantité (résultat admis ici) : $Fx^2(3L - x)$, où x désigne l'abscisse le long de la poutre. Cette solution exacte particulière s'exprime comme un polynôme de degré 3.

La stratégie dI pourrait donc se contenter d'un seul sous-domaine $[0, L]$ avec une interpolation polynômiale de degré 3 (les $u_i^h(x)$ sont des polynômes de la forme $a_3x^3 + a_2x^2 + a_1x + a_0$), i.e. un seul élément fini de degré 3. On aurait alors : $m = 1$, $\Omega_1 = \Omega$ et $u_1^h(x) = u^h(x)$. La solution approchée est donc de degré 3 et si elle respecte les conditions aux limites de la poutre cette solution approchée sera donc égale à la solution exacte. Cette situation est illustrée sur la FIG. 4 (à droite).

La stratégie iD pourrait consister en une discrétisation en 15 sous-domaines à interpolation constante, i.e. 15 éléments finis de degré 0. Cette situation est illustrée sur la FIG. 4 (à gauche). Dans ce cas la solution approchée ne coïncide pas avec la solution exacte.

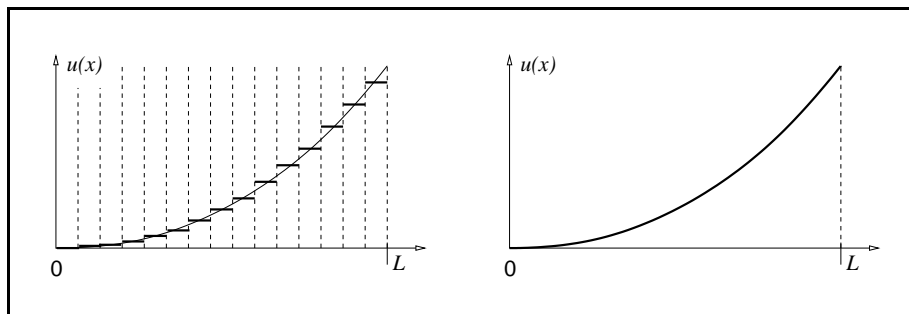


FIG. 4 – Deux stratégies différentes pour traiter un problème de poutre en flexion.

2.3 Problème approché

2.3.1 Forme variationnelle des problèmes d'élasticité linéaire

On se restreint ici, comme précédemment, aux problèmes stationnaires d'élasticité linéaire en mécanique des solides sous l'hypothèse des petites perturbations (HPP) :

▷ *Problème stationnaire d'élasticité linéaire en HPP*

$$\begin{cases} \operatorname{div} \sigma + f = 0 & \text{dans } \Omega \\ \sigma = \mathbb{C} : \varepsilon & \text{dans } \Omega \\ \varepsilon = \nabla_S u & \text{dans } \Omega \\ u = 0 & \text{sur } \partial\Omega_U \\ \sigma \cdot n = F & \text{sur } \partial\Omega_F \end{cases} \quad (7)$$

où $\nabla_S u = 1/2 (\nabla u + {}^T \nabla u)$ et “:” indique le produit doublement contracté.

Pour résoudre ces problèmes (ou plus généralement les systèmes d'EDP), il est judicieux de les écrire sous une *forme variationnelle*. Il s'agit d'une forme intégrale, dite aussi forme faible. Cette forme est obtenue soit à l'aide d'outils mathématiques (formulation variationnelle), soit à l'aide de principes physiques (principe des puissances virtuelles). Elle repose sur l'utilisation du résultat d'analyse fonctionnelle suivant :

$$\int_{\Omega} g(x)h(x)dx = 0, \quad \forall g(x) \Leftrightarrow h(x) = 0, \quad \forall x \in \Omega \quad (8)$$

où Ω , $g(x)$ et $h(x)$ ont certaines régularités non précisées ici.

La forme variationnelle nécessite par ailleurs de supposer que le champ de forces volumiques f et le champ de forces extérieures F sont suffisamment réguliers, et d'introduire l'espace des champs de déplacements cinématiquement admissibles :

$$\mathcal{U}_{ad} = \{v \text{ suffisamment régulier} \mid v = \bar{u} \text{ sur } \partial\Omega_U\}, \quad (9)$$

▷ *Déplacements cinématiquement admissibles*

et l'espace des champs de déplacements compatibles avec les conditions cinématiques :

$$\mathcal{V} = \{v \text{ suffisamment régulier} \mid v = 0 \text{ sur } \partial\Omega_U\}. \quad (10)$$

▷ *Déplacements compatibles avec les conditions cinématiques*

D'une manière générale, ces espaces sont tels que toutes les quantités qui interviennent dans la forme variationnelle donnée ci-dessous (ou dans l'écriture du principe des puissances virtuelles) soient définies. Les champs sont en particulier continus, ainsi qu'un certain nombre de leurs dérivées quand le problème le requiert. L'espace \mathcal{U}_{ad} impose de plus le respect des conditions aux limites du problème (ici une condition sur le déplacement, mais on peut aussi trouver des conditions sur la vitesse, par exemple).

Ces espaces \mathcal{U}_{ad} et \mathcal{V} sont tels que :

- si $w \in \mathcal{U}_{ad}$ et $v \in \mathcal{V}$ alors $w + v \in \mathcal{U}_{ad}$,
- si $w \in \mathcal{U}_{ad}$ et $z \in \mathcal{U}_{ad}$ alors $w - z \in \mathcal{V}$.

L'utilisation de ces espaces permet d'obtenir la formulation variationnelle du problème (7) en faisant le produit scalaire de sa première équation avec un élément v de \mathcal{V} et en intégrant sur Ω :

$$\int_{\Omega} \operatorname{div} \sigma \cdot v \, dx + \int_{\Omega} f \cdot v \, dx = 0.$$

L'utilisation de la formule de Green conduit à (n est la normale unitaire sortante à Ω :

$$\int_{\partial\Omega} \sigma \cdot n \cdot v \, dx - \int_{\Omega} \sigma : \nabla_S v \, dx + \int_{\Omega} f \cdot v \, dx = 0,$$

en décomposant $\partial\Omega$ en $\partial\Omega_U$ (où v est nul car dans \mathcal{V}) et $\partial\Omega_F$ (où $\sigma \cdot n = F$) et en utilisant le fait que $\sigma = \mathbb{C} : \nabla_S u$ et les symétries mineures de \mathbb{C} ($\mathbb{C}_{ijkl} = \mathbb{C}_{jikl}$), on obtient la formulation variationnelle suivante :

$$\left\{ \begin{array}{l} \text{Trouver } u \in \mathcal{U}_{ad} \text{ tel que } \forall v \in \mathcal{V}, \quad \boxed{a(u, v) = l(v)} \\ \text{avec } \left\{ \begin{array}{l} a(u, v) = \int_{\Omega} \nabla_S u : \mathbb{C} : \nabla_S v \, dx \\ l(v) = \int_{\Omega} f \cdot v \, dx + \int_{\partial\Omega_F} F \cdot v \, ds \end{array} \right. \end{array} \right. \quad (11) \quad \triangleright \text{Forme variationnelle}$$

Une autre écriture est possible :

$$\forall z \in \mathcal{U}_{ad}, \quad a(u, u - z) = l(u - z). \quad (12)$$

2.3.2 Propriétés de $a(u, v)$

La forme de $a(u, v)$ indique qu'il s'agit d'une forme bilinéaire. Cette forme bilinéaire possède certaines propriétés liées à celles des coefficients d'élasticité.

Les symétries majeures du tenseur \mathbb{C} ($\mathbb{C}_{ijkl} = \mathbb{C}_{klij}$) induisent la symétrie de $a : a(u, v) = a(v, u)$.

Par ailleurs, les coefficients d'élasticité sont elliptiques :

$\exists \beta > 0$ tel que : $\varepsilon : \mathbb{C} : \varepsilon \geq \beta \varepsilon : \varepsilon$.

Cette dernière propriétés implique :

$$\begin{aligned} a(v, v) &\geq 0, \\ a(v, v) &= 0 \Leftrightarrow v = 0. \end{aligned}$$

La fonctionnelle $a(u, v)$ est :

- une forme bilinéaire
- symétrique,
- définie,
- positive.

▷ *Propriétés utiles de $a(u, v)$*

2.3.3 Théorème de l'énergie potentielle en déplacement

La formulation variationnelle (11) est équivalente au principe de minimum de l'énergie potentielle :

$$\left\{ \begin{array}{l} \text{Trouver } u \in \mathcal{U}_{ad} \text{ tel que } \forall z \in \mathcal{U}_{ad}, \quad \boxed{J(u) \leq J(z)} \\ \text{avec } J(z) = \frac{1}{2}a(z, z) - l(z). \end{array} \right. \quad (13) \quad \text{▷ Minimum de l'énergie potentielle}$$

L'implication (11) \Rightarrow (13) s'obtient comme suit :

$$\begin{aligned} J(z) - J(u) &= \frac{1}{2}a(z, z) - l(z) - \frac{1}{2}a(u, u) - l(u) \\ &= \frac{1}{2} \underbrace{a(z - u, z - u)}_{\geq 0} + \underbrace{a(u, z - u) - l(z - u)}_{=0}. \end{aligned}$$

La réciproque s'obtient en posant $z = u + \alpha(u - w)$ où α est un réel, w est dans \mathcal{U}_{ad} (alors $(u - w) \in \mathcal{V}$, donc $\alpha(u - w) \in \mathcal{V}$ et donc $z \in \mathcal{U}_{ad}$) :

$$\begin{aligned} J(z) &= \frac{1}{2}a(u + \alpha(u - w), u + \alpha(u - w)) - l(u + \alpha(u - w)) \\ &= \frac{1}{2}\alpha^2 a(u - w, u - w) + \alpha a(u, u - w) - \alpha l(u - w) + \underbrace{\frac{1}{2}a(u, u) - l(u)}_{J(u)}, \end{aligned}$$

d'où :

$$\frac{1}{2}\alpha^2 a(u - w, u - w) + \alpha (a(u, u - w) - l(u - w)) \geq 0, \forall \alpha \in \mathbb{R},$$

soit : $a(u, u - w) - l(u - w) = 0, \forall w \in \mathcal{U}_{ad}$ (voir équation (12)). Ce qui achève la preuve.

2.3.4 Existence et unicité de la solution

Ce principe du minimum de l'énergie potentielle permet de montrer l'existence et l'unicité de la solution au problème (7). Du point de vue de la simulation numérique de tels problèmes, cette information est cruciale pour les deux raisons suivantes :

- si ces problèmes n'avaient pas de solution, il serait illusoire d'en chercher une par des méthodes numériques,
- si ces problèmes avaient plusieurs solutions, les méthodes numériques en fourniraient une mais laquelle ?

La preuve de cette existence et unicité repose sur :

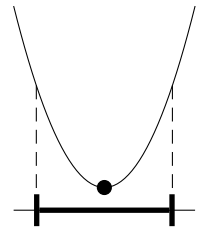
- le cadre fonctionnel,
- les propriétés de $a(u, v)$: bilinéarité, symétrie, définie positivité ; ces propriétés provenant directement de celles des coefficients d'élasticité.

L'existence d'une solution au problème de minimum (13) s'obtient en remarquant que :

1. $J(v)$ est une fonction strictement convexe,
2. \mathcal{U}_{ad} est un espace convexe.

Une fonction strictement convexe possède un minimum et un seul sur un ensemble convexe (penser à une fonction du type : $\frac{1}{2}ax^2 - lx$, avec a et l des réels et a strictement positif, cf $a(u, v)$ définie positive).

▷ Minimum d'une fonction strictement convexe sur un domaine convexe



Le point 2 s'obtient en utilisant directement la définition d'un ensemble convexe :

un ensemble \mathcal{G} est convexe si pour tout x et y dans \mathcal{G} et tout réel α dans $[0, 1]$, $z = \alpha x + (1 - \alpha)y$ est dans \mathcal{G} .

Le point 1 s'appuie sur la définition d'une fonction strictement convexe :

une fonction g est strictement convexe sur un ensemble \mathcal{G} convexe si pour tout x et y distincts dans \mathcal{G} et tout réel α dans $]0, 1[$, on a :

$$g(\alpha x + (1 - \alpha)y) < \alpha g(x) + (1 - \alpha)g(y).$$

Pour $J(v)$, cette propriété de convexité peut se démontrer comme suit (on utilise à plusieurs reprises la bilinéarité de $a(u, v)$, sa symétrie et sa positivité) :

$$\begin{aligned} & J(\alpha x + (1 - \alpha)y) - \alpha J(x) - (1 - \alpha)J(y) \\ &= \frac{1}{2}\alpha^2 a(x, x) + \frac{1}{2}(1 - \alpha)^2 a(y, y) + \alpha(1 - \alpha)a(x, y) - \frac{1}{2}\alpha a(x, x) \\ &\quad - \frac{1}{2}(1 - \alpha)a(y, y) \\ &= -\frac{1}{2}\alpha(1 - \alpha)a(x, x) - \frac{1}{2}\alpha(1 - \alpha)a(y, y) + \alpha(1 - \alpha)a(x, y) \\ &= -\underbrace{\alpha(1 - \alpha)}_{>0} \underbrace{a(x - y, x - y)}_{>0} \\ &< 0. \end{aligned}$$

En complément à ce résultat et sans donner de justifications, il est intéressant de noter que l'*unicité* de la solution au problème de minimum provient du fait qu'il existe une partie de la frontière de Ω sur laquelle des déplacements sont appliqués. En l'absence de telles conditions aux limites, par exemple si l'on n'applique que des forces sur la frontière de Ω , la solution n'est définie qu'à un mouvement de corps rigide près. Mathématiquement, cette situation correspond au cas où $a(u, v)$ n'est plus définie : $a(v, v) = 0$ n'implique plus $v = 0$ mais seulement que v soit de la forme $c + b \wedge x$.

2.3.5 Formulation variationnelle approchée

Il s'agit ici de donner une forme approchée de la formulation variationnelle (11) dont le champ $u^h(x)$ défini par morceaux soit solution.

Sans préciser, on considère un espace \mathcal{U}_{ad}^h de *dimension finie* engendré par la base des fonctions d'interpolation *globales* : $\varphi_1, \dots, \varphi_N$ (i.e. la version discrétisée de l'espace des déplacements admissibles) ; ainsi que \mathcal{V}^h , aussi de dimension finie, le pendant discret de \mathcal{V} .

La formulation variationnelle approchée (ou discrète) est alors :

$$\text{Trouver } u^h \in \mathcal{U}_{ad}^h \text{ tel que } \forall v \in \mathcal{V}^h, \quad \boxed{a(u^h, v) = l(v)} \quad (14) \quad \triangleright \text{Forme variationnelle approchée}$$

Puisque l'on a (voir équation 6) :

$$u^h(x) = \sum_{i=1}^N u_i \varphi_i(x)$$

et que a est bilinéaire, on obtient, pour tout j , en choisissant comme fonctions test (les fonctions tests sont les fonctions v de \mathcal{V} utilisées dans la formulation variationnelle (11)) $v = \varphi_j$:

$$\sum_{i=1}^N a(\varphi_i, \varphi_j) u_i = l(\varphi_j) \quad \forall j = 1, N. \quad (15)$$

Il s'agit donc bien de résoudre un système

$$\boxed{\mathbf{K}\{U\} = \{F\} \quad \text{avec} \quad \begin{cases} \mathbf{K}_{ij} &= a(\varphi_i, \varphi_j) \\ F_j &= l(\varphi_j). \end{cases}} \quad (16) \quad \triangleright \mathbf{K}_{ij} = a(\varphi_i, \varphi_j)$$

On remarque que l'établissement de cette formulation approchée nécessite que les φ_i soient dans \mathcal{U}_{ad}^h , c'est à dire qu'elles possèdent une certaine régularité et notamment que ces fonctions soient *continues* à travers les mailles. Cette condition de continuité impose en particulier les conditions précédemment données sur les mailles au paragraphe 2.2.1.

2.3.6 Stratégie MEF et propriétés de \mathbf{K}

L'égalité (16) entre \mathbf{K}_{ij} et $a(\varphi_i, \varphi_j)$ assure que \mathbf{K} hérite des propriétés de a . Celles qui nous intéressent principalement sont les suivantes :

$\triangleright \mathbf{K}$ est symétrique et inversible

- \mathbf{K} est symétrique,
- \mathbf{K} est (définie donc) inversible.

La propriété d'inversibilité de \mathbf{K} provient du fait que a soit définie. Ceci signifie du point de vue mécanique que \mathbf{K} est inversible lorsque la structure à simuler possède un morceau de sa frontière sur laquelle des conditions aux

limites en déplacement sont appliquées. Le sens de cette condition est clair : si une structure n'était soumise qu'à des forces extérieures, elle n'aurait pas de position d'équilibre (déplacement rigide dans l'espace).

La propriété essentielle de \mathbf{K} dans le cadre de la MEF est la suivante :

■ La matrice \mathbf{K} est creuse.

▷ **\mathbf{K} est creuse**

Cette propriété notable de \mathbf{K} , est induite directement par la stratégie sur laquelle repose la MEF. Parmi les différentes méthodes d'approximation la MEF se caractérise par le choix de fonctions d'interpolation simples associées à une discrétisation fine.

■ La stratégie MEF est :

1. un nombre de sous-domaines de discrétisation grand,
2. des fonctions d'interpolations globales φ_i à support petit.

▷ *Stratégie MEF*

Pour obtenir cette propriété, est associée à chaque nœud i une fonction φ_i non nulle dans toutes les mailles qui possèdent le nœud i et nulle ailleurs (voir FIG. 2). Si le nombre de sous-domaines que l'on utilise pour discrétiser Ω est grand, les φ_i sont donc non nulles sur des domaines petits devant la taille de Ω (support petit). Les termes $a(\varphi_i, \varphi_j)$ sont donc majoritairement nuls et la matrice \mathbf{K} est creuse (pleine de zéro). Cependant, la taille de \mathbf{K} est d'autant plus importante que le nombre de sous-domaines est grand.

Une conséquence notable de ce choix de fonctions φ_i non nulles seulement dans les mailles possédant le nœud i est qu'il est possible de calculer \mathbf{K} par petits "morceaux" : il s'agit des matrices de rigidité élémentaires (voir équation (39)). Ce calcul de \mathbf{K} par "superposition" de matrices élémentaires traduit le fait qu'il est possible de s'intéresser localement à la raideur d'un élément fini sans se soucier du reste de la structure. Dit autrement, il s'agit d'une propriété de *localité* de la MEF : les termes \mathbf{K}_{ij} sont nuls pour tous les couples de nœuds (i, j) qui ne sont pas connectés dans le maillage.

2.4 Commentaires et compléments

Ce paragraphe peut être ignoré en première lecture.

2.4.1 Stockage informatique de \mathbf{K}

Du point de vue informatique, le stockage de la matrice \mathbf{K} est souvent coûteux. Si N est le nombre de DDL, la matrice \mathbf{K} contient N^2 termes. Un calcul de structure industriel nécessite typiquement quelques centaines de milliers de DDL ; on conçoit alors la nécessité de réduire la place prise en mémoire par le stockage de \mathbf{K} . Les propriétés précédentes de \mathbf{K} peuvent être utilisées pour réduire cette place mémoire.

- \mathbf{K} est symétrique : on ne stocke (mais aussi on n'en calcule) que le triangle supérieur ou inférieur ;
- \mathbf{K} est creuse : on peut chercher à s'affranchir du stockage de tous les termes nuls. Trois principales méthodes sont utilisées :
 1. Le *stockage bande* : seules les diagonales du triangle supérieur ou inférieur possédant des termes non nuls sont stockées ; le nombre de diagonale est la *largeur de bande*.
 2. Le *stockage ligne de ciel* : pour chaque ligne du triangle, seule la partie non nulle de la ligne est stockée depuis le terme diagonal jusqu'au dernier terme non nul. Un indicateur de terme diagonal est alors nécessaire.
 3. Le *stockage morse* : seuls les termes non nuls de \mathbf{K} sont stockés. Pour chaque terme un indicateur de place (ligne, colonne) est alors nécessaire.

Dans tous les cas, le stockage sera optimisé dès lors que la largeur de bande de \mathbf{K} est minimisée. Divers algorithmes permettent de diminuer cette largeur de bande. Ils reposent sur une renumérotation la plus intelligente possible des DDL du maillage. Par exemple, pour une structure très élancée, il est judicieux de numéroter les DDL en progressant d'abord selon le côté court de la structure, puis ensuite selon le côté long.

2.4.2 Les solveurs

La résolution d'un problème d'élasticité linéaire par la méthode des éléments finis conduit à la résolution d'un système linéaire. De nombreuses méthodes numériques (les solveurs) permettent la résolution de tels systèmes linéaires. On distingue les méthodes *directes* et les méthodes *itératives*. Les méthodes directes permettent une résolution en un nombre fini d'opérations. Les méthodes itératives permettent d'obtenir la solution comme limite d'une suite ; elles nécessitent en général un *critère d'arrêt*. Sans être exhaustif, quelques unes de ces méthodes sont listées ici. On note ici \mathcal{N} le nombre d'opérations nécessaires pour obtenir la solution.

- **Méthodes directes** Ces méthodes sont recommandées pour les systèmes de petite taille. Soit le système $\mathbf{K}x = b$.

- **Méthode de Gauss** Cette méthode consiste à transformer le système ligne par ligne (à la transformation concernant la ligne r , on note \mathbf{K}^r la matrice transformée et b^r le second membre transformé) pour obtenir une matrice triangulaire, alors immédiate à inverser. On a :

$$\mathbf{K}_{ij}^{r+1} = \mathbf{K}_{ij}^r - \frac{\mathbf{K}_{ir}^r \mathbf{K}_{rj}^r}{\mathbf{K}_{rr}^r} \quad (17)$$

$$b_i^{r+1} = b_i^r - \frac{\mathbf{K}_{ir}^r}{\mathbf{K}_{rr}^r} b_r^r \quad (18)$$

- **Méthode LU** Cette méthode consiste à décomposer \mathbf{K} en un produit LU de deux matrices, l'une triangulaire inférieure L (Lower), l'autre triangulaire supérieure U (Upper). Comme pour la méthode de Gauss, cette décomposition s'effectue ligne par ligne comme suit :

$$\triangleright LU : \mathcal{N} = \frac{N^3}{3}$$

$$L = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ \mathbf{K}_{ir}^r & & 1 \\ \mathbf{K}_{rr}^r & & \end{bmatrix} \text{ et } U = \begin{bmatrix} \mathbf{K}_{11} & \cdots & \mathbf{K}_{1N} \\ & (17) & (17) \\ 0 & & (17) \end{bmatrix}.$$

La résolution du système se fait alors en deux étapes :

$Ly = b \Rightarrow y$ (descente) puis $Ux = y \Rightarrow x$ (remontée).

- **Méthode de Choleski** Si \mathbf{K} est symétrique, définie, positive, cette méthode consiste à décomposer \mathbf{K} en un produit $B^T B$ où B est une matrice triangulaire inférieure puis d'effectuer une descente-remontée comme pour la méthode LU . La matrice B se calcule comme suit :

$$\triangleright \text{Choleski} : \mathcal{N} = \frac{N^3}{6}$$

$$\begin{cases} B_{11} = \sqrt{\mathbf{K}_{11}} \\ B_{j1} = \frac{\mathbf{K}_{j1}}{B_{11}} \end{cases} \text{ puis } \begin{cases} B_{ii} = \sqrt{\mathbf{K}_{ii} - \sum_{k=1}^{i-1} B_{ik}^2} \\ B_{ji} = \frac{1}{B_{ii}} \left(\mathbf{K}_{ij} - \sum_{k=1}^{i-1} B_{ik} B_{jk} \right). \end{cases}$$

- **Méthodes itératives** On cherche \bar{x} solution de $\mathbf{K}\bar{x} = b$ comme la limite d'une suite x^k telle que : $x^{k+1} = Tx^k + h$. Une telle limite converge si le rayon spectral de T est strictement inférieur à 1. En choisissant une décomposition de $\mathbf{K} = M - N$, où M est une matrice facile à inverser, on obtient : $T = M^{-1}N$ et $h = M^{-1}b$. Différentes décomposition " $M - N$ " conduisent à différentes méthodes. Toute matrice \mathbf{K} se décompose en $\mathbf{K} = D - (E + F)$, où D est la diagonale de \mathbf{K} et E et F sont respectivement les opposés de la partie inférieure et supérieure de \mathbf{K} .

- **Méthode de Jacobi** On prend : $M = D$ et $N = E + F$. Une condition permettant d'assurer la convergence de cette méthode est de s'assurer que \mathbf{K} est à diagonale strictement dominante.

- **Méthode de Gauss-Seidel** On choisit : $M = D - E$ et $N = F$. Bien qu'il s'agisse d'une méthode itérative, il est usuel et empirique

$$\triangleright GS : \mathcal{N} \simeq N^2$$

de dire qu'un nombre fini d'itérations suffit pour converger vers la solution.

2.4.3 Conditionnement de \mathbf{K}

Un système linéaire $\mathbf{K}x = b$ obtenu par des méthodes numériques telles que la méthode des éléments finis peut être entaché de certaines erreurs numériques, parfois liées à la précision de la machine sur laquelle sont effectués les calculs. Il convient alors de savoir dans quelle mesure ces erreurs affectent la solution du problème. Considérons l'exemple suivant :

$$\mathbf{K} = \begin{bmatrix} 3 & 4 \\ 3 & 4.001 \end{bmatrix} \text{ et } b = \begin{Bmatrix} 7 \\ 7.001 \end{Bmatrix}$$

La solution à ce système est $\langle 1 \ 1 \rangle$. Si le second membre est perturbé en

$$b = \begin{Bmatrix} 7 \\ 7.004 \end{Bmatrix}$$

alors la solution devient $\langle -3 \ 4 \rangle$. Ce système est donc grandement sensible aux erreurs numériques. Cette sensibilité provient d'un mauvais *conditionnement* de \mathbf{K} qui se définit comme :

$$C(\mathbf{K}) = \|\mathbf{K}\| \|\mathbf{K}^{-1}\|, \quad (19) \quad \triangleright \text{Définition du conditionnement de } \mathbf{K}$$

où $\|\cdot\|$ est une norme matricielle comme, par exemple :

$$\begin{aligned} \|\mathbf{K}\|_1 &= \max_j \sum_{i=1,N} \mathbf{K}_{ij}, \\ \|\mathbf{K}\|_\infty &= \max_i \sum_{j=1,N} \mathbf{K}_{ij}, \\ \|\mathbf{K}\|_2 &= \sqrt{\rho(\mathbf{K}^T \mathbf{K})}, \end{aligned} \quad \triangleright \text{Normes matricielles}$$

avec $\rho(\mathbf{K}^T \mathbf{K})$ le rayon spectral de $\mathbf{K}^T \mathbf{K}$ (i.e. sa plus grande valeur propre en module).

Ce conditionnement permet d'estimer la sensibilité de la solution à une perturbation du second membre δb . On a :

$$\frac{\|\delta x\|}{\|x\|} \leq C(\mathbf{K}) \frac{\|\delta b\|}{\|b\|}. \quad (20)$$

Par conséquent, si $C(\mathbf{K})$ est proche de 1, \mathbf{K} est bien conditionnée, si $C(\mathbf{K})$ est très grand, \mathbf{K} est mal conditionnée. Dans l'exemple précédent, $C(\mathbf{K}) \simeq 1.6 \times 10^4$.

On note, que dans le cas de l'utilisation de la $\|\cdot\|_2$ et si \mathbf{K} est symétrique, définie, positive, on a :

$$C(\mathbf{K}) = \left| \frac{\lambda_{max}}{\lambda_{min}} \right|, \quad (21)$$

où λ_{max} et λ_{min} sont respectivement la plus grande et la plus petite des valeurs propres de \mathbf{K} .

La principale cause de l'accumulation d'erreurs d'arrondis dans les méthodes directes est un mauvais conditionnement de \mathbf{K} .

Du point de vue mécanique, les causes d'un mauvais conditionnement de \mathbf{K} peuvent être :

- une distorsion du maillage,
- une partie de la structure qui se déforme selon un mode proche de celui d'un corps rigide,
- de forts contrastes de raideur dans la structure.

▷ Causes mécaniques d'un mauvais conditionnement de \mathbf{K}

Le conditionnement de \mathbf{K} peut être amélioré en utilisant des *préconditionneurs*.

2.4.4 Application des conditions aux limites

Supposons que l'on cherche à appliquer une condition \bar{u} en déplacement sur le DDL u_i . Trois méthodes principales sont utilisées :

- *Suppression d'équations*. Une technique consiste à venir supprimer du système les lignes et les colonnes correspondant à l'indice i pour lequel on souhaite appliquer le déplacement. Cette méthode est simple et efficace. Elle suppose cependant que l'on n'ait pas besoin de conserver la matrice \mathbf{K} en vue de calculs ultérieurs.
- *La méthode d'Irons*. Cette méthode consiste à ajouter à \mathbf{K}_{ii} une constante γ très grande devant tous les autres termes de \mathbf{K} (méthode de pénalisation) et de remplacer le terme du second membre en place i par $\gamma\bar{u}$. On obtient alors pour la ligne i du système :

$$\underbrace{\sum_{j \neq i} \mathbf{K}_{ij} u_j}_{\text{négligeable}} + \underbrace{(\mathbf{K}_{ii} + \gamma)}_{\simeq \gamma} u_i = \gamma \bar{u}.$$

Cette méthode présente l'inconvénient de rendre la matrice très mal conditionnée (ajout d'un terme grand devant les autres).

- *Le multiplicateur de Lagrange*. Si on introduit une matrice \mathbf{G} telle que :

$$\mathbf{G}\{U\} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \bar{u} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \leftarrow \text{ligne } i = \underbrace{\{\bar{U}\}}_{\text{notation}}$$

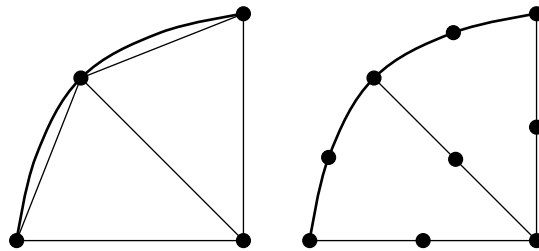
alors la technique des multiplicateurs de Lagrange consiste à réécrire le système élément finis sous la forme des deux systèmes suivants :

$$\begin{aligned} \mathbf{K}\{U\} + {}^T\mathbf{G}\{\lambda\} &= \{F\} \\ \mathbf{G}\{U\} &= \{\bar{U}\} \end{aligned}$$

où le vecteur $\{\lambda\}$ contient les multiplicateurs inconnus. Dans le cas considéré (déplacement \bar{u} imposé sur le DDL u_i), la composante i de $\{\lambda\}$ a une signification physique immédiate : il s'agit de l'effort qu'il faut appliqué au nœud de la structure concerné par le DDL u_i pour assurer que son déplacement soit bien égal à \bar{u} .

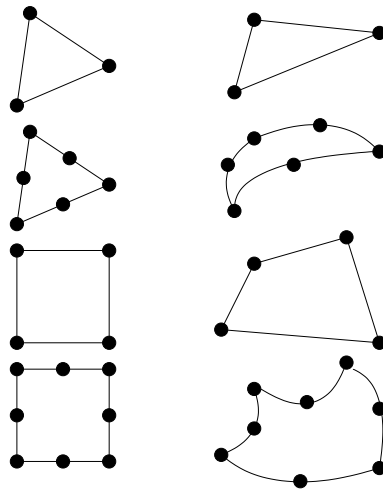
2.4.5 Approximation des domaines courbes

Comme mentionné au paragraphe 2.2.1, chaque maille d'un maillage éléments finis est a priori un polyèdre. Ceci n'est vrai en toute rigueur que si le domaine à mailler est lui-même un polyèdre. Lorsque le domaine Ω à discrétiser possède des frontières courbes, il peut paraître nécessaire d'introduire un grand nombre de nœuds sur les frontières de manière à approcher convenablement le domaine sous peine de se retrouver dans la situation illustrée ci-dessous (figure de gauche : écart important entre le domaine Ω et sa discrétisation). Pour limiter le nombre d'éléments finis à introduire dans la discrétisation, il est possible d'utiliser des éléments de *plus haut degré d'interpolation*. Sur l'illustration ci-dessous (figure de droite), l'utilisation d'éléments triangulaires à interpolation *quadratique* permet d'approcher la frontière à l'aide d'une fonction quadratique par morceaux (des arcs de paraboles).



Un élément fini à interpolation quadratique possède plus de nœuds qu'un élément à interpolation linéaire (il sera donc plus coûteux en termes de temps de calcul) et donc plus de degrés de liberté (il sera donc plus précis). En général, on peut admettre qu'à *même nombre de nœuds* (et donc à temps de calcul équivalent), un maillage constitué d'éléments quadratiques donnera un résultat plus précis qu'un maillage constitué d'éléments linéaires. La figure ci-dessous illustre les modes de déformation des triangles à 3 et 6 nœuds et des quadrilatères à 4 et 8 nœuds. ⁽²⁾

⁽²⁾ Attention, d'un élément linéaire à un élément quadratique, le nombre de nœuds n'est pas systématiquement multiplié par 2 (voir les tétraèdres à 4 nœuds et les tétraèdres à 10 nœuds au paragraphe 3.3.2)



2.5 Un exemple simple détaillé

On considère ici (voir FIG. 5) une barre de longueur L , de section S , constituée d'un matériau élastique homogène isotrope de masse volumique ρ et de module de Young E , orientée selon un axe noté x , encastrée à son extrémité $x = 0$, libre à l'autre extrémité ($x = L$), soumise à son poids propre (accélération de la pesanteur notée g , colinéaire à l'axe x et orientée de $x = 0$ à $x = L$).

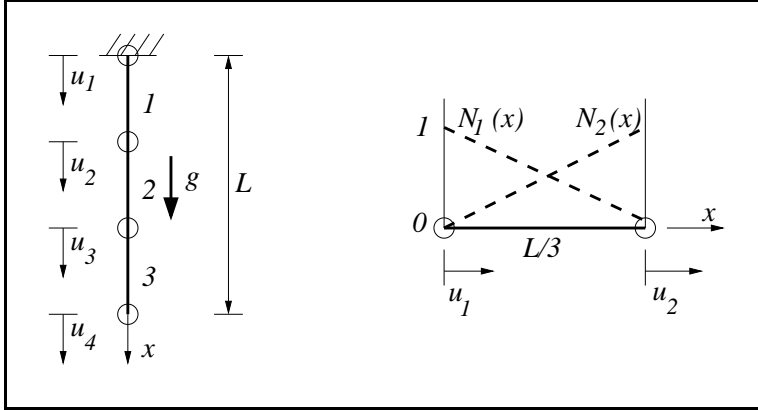


FIG. 5 – Problème d'une barre sous poids propre, encastrée.

2.5.1 Solution analytique unidimensionnelle (traction pure)

Il s'agit ici d'un des rares cas où l'on puisse résoudre analytiquement le problème (1). Le domaine Ω est $S \times [0, L]$, sa frontière $\partial\Omega_U$ se restreint au plan $S \times \{x = 0\}$ (déplacement imposé par l'encastrement) et sa frontière $\partial\Omega_F$ se restreint au plan $S \times \{x = L\}$ (extrémité libre).⁽³⁾ Les trois premières équations du problème (1) s'écrivent ici sur $[0, L]$:

$$\begin{cases} \operatorname{div} \sigma + \rho g = 0 \\ \sigma = E\varepsilon \\ \varepsilon = \frac{du}{dx} \end{cases} \quad (22)$$

Ces équations conduisent à l'EDP simple suivante :

$$E \frac{\partial^2 u}{\partial x^2} + \rho g = 0, \quad (23)$$

dont la solution est de la forme $u(x) = -\frac{\rho g}{E} \frac{x^2}{2} + Cx + D$, où C et D sont des constantes. Leur détermination se fait en utilisant les deux conditions aux limites $u(0) = 0$ et $\sigma(L) = 0$. La première impose $D = 0$ et la seconde

⁽³⁾Attention, le poids n'est pas une force ponctuelle, mais volumique, il ne fait donc pas partie des conditions aux limites mais il intervient dans l'équation d'équilibre.

$Eu'(L) = 0$, soit $C = L\rho g$. Les champs de déplacements et de contraintes solutions du problème sont donc :

$$u(x) = \frac{\rho g}{2E}x(2L - x) \quad \text{et} \quad \sigma(x) = \rho g(L - x). \quad (24)$$

Le champ $u(x)$ se présente sous la forme d'un polynôme de degré 2 et le champ $\sigma(x)$ sous celle d'un polynôme de degré 1. On cherche par la suite à trouver une solution approchée en utilisant une discrétisation éléments finis basée sur des éléments "barre".

2.5.2 Discrétisation spatiale

On considère la discrétisation géométrique proposée sur la FIG. 5 et constituée de 4 nœuds et de 3 sous-domaines de section S et de longueur $L/3$.

2.5.3 Formulation variationnelle par sous-domaine

En notant $f = \rho g$, la formulation variationnelle du problème (23) s'écrit sur Ω :

Trouver $u \in \mathcal{U}_{ad}$, tel que $\forall v \in \mathcal{V}$

$$\int_{\Omega} E \frac{\partial^2 u}{\partial x^2} \cdot v \, dx = - \int_{\Omega} f \cdot v \, dx, \quad (25)$$

où $\mathcal{U}_{ad} = \{u \text{ suffisamment régulier} \mid u(0) = 0\} = \mathcal{V}$.

En intégrant par parties le membre de gauche de l'équation précédente et en ramenant les intégrales volumiques à des intégrales selon les segments $[0, L/3]$, $[L/3, 2L/3]$ et $[2L/3, L]$, on obtient $\forall v \in \mathcal{V}$:

$$\begin{aligned} & \left[ESu' \cdot v \right]_0^{L/3} - \int_0^{L/3} ESu' \cdot v' \, dx \\ & + \left[ESu' \cdot v \right]_{L/3}^{2L/3} - \int_{L/3}^{2L/3} ESu' \cdot v' \, dx \\ & + \left[ESu' \cdot v \right]_{2L/3}^L - \int_{2L/3}^L ESu' \cdot v' \, dx \end{aligned} = \begin{cases} - \int_0^{L/3} Sf \cdot v \, dx \\ - \int_{L/3}^{2L/3} Sf \cdot v \, dx \\ - \int_{2L/3}^L Sf \cdot v \, dx \end{cases}$$

où u' et v' sont respectivement les dérivées de u et v par rapport à x .

Les termes de bords s'annulent deux à deux en $L/3$ et $2L/3$ (continuité de u et de $\sigma \cdot n$ à la frontière des éléments) et ceux relatif à la frontière du domaine Ω ($x = 0$ et $x = L$) se traitent à l'aide des conditions aux limites. En $x = 0$, on a $v = 0$ (car $v \in \mathcal{V}$) et donc $(ESu' \cdot v)|_{x=0} = 0$. En $x = L$, la condition de bord libre impose $\sigma(L) = 0$ et la relation $\sigma = Eu'$ conduit à $(ESu' \cdot v)|_{x=L} = 0$.

Par conséquent, on a la formulation variationnelle suivante :

$$a(u, v) = l(v), \quad \forall v \in \mathcal{U}_{ad}, \quad (26)$$

avec

$$a(u, v) = \int_0^{L/3} ESu' \cdot v' dx + \int_{L/3}^{2L/3} ESu' \cdot v' dx + \int_{2L/3}^L ESu' \cdot v' dx \quad (27)$$

$$l(v) = \int_0^{L/3} Sf \cdot v dx + \int_{L/3}^{2L/3} Sf \cdot v dx + \int_{2L/3}^L Sf \cdot v dx. \quad (28)$$

2.5.4 Fonctions d'interpolation par sous-domaine

La fonctionnelle $a(u, v)$ étant en " $u' \cdot v'$ ", les fonctions d'interpolation (ou fonctions de forme) doivent être des polynômes de degré 1 (au moins). Par ailleurs, la nécessité de raccorder u aux deux extrémités du sous-domaine, impose de posséder deux fonctions de forme (FIG. 5) :

$$N_1(x) = 1 - \frac{x}{L/3} \quad \text{et} \quad N_2(x) = \frac{x}{L/3}. \quad (29)$$

2.5.5 Discrétisation de la formulation variationnelle par sous-domaine

A titre d'illustration on s'intéresse ici, par exemple, au sous-domaine 1, c'est à dire à l'élément numéro 1. Les inconnues de cet élément sont u_1 et u_2 . La relation d'interpolation locale (5) s'écrit ici :

$$u_1^h = \langle N_1(x) \ N_2(x) \rangle \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}. \quad (30)$$

Cette relation vaut pour la discrétisation spatiale v^h du champ v :

$$v_1^h = \langle N_1(x) \ N_2(x) \rangle \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix}. \quad (31)$$

Les dérivées spatiales de ces quantités s'écrivent donc :

$$\left(u_1^h \right)' = \left\langle \frac{\partial N_1(x)}{\partial x} \ \frac{\partial N_2(x)}{\partial x} \right\rangle \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad (32)$$

$$\left(v_1^h \right)' = \left\langle \frac{\partial N_1(x)}{\partial x} \ \frac{\partial N_2(x)}{\partial x} \right\rangle \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix}. \quad (33)$$

En définissant, la matrice \mathbf{B} , dite *matrice gradient élémentaire* (le gradient du vecteur des fonctions de forme de l'élément, qui se réduit ici à un vecteur), comme suit :

▷ *Matrice gradient*
 $\mathbf{B} = \nabla \{N_e(x)\}$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1(x)}{\partial x} & \frac{\partial N_2(x)}{\partial x} \end{bmatrix}, \quad (34)$$

le produit scalaire $u' \cdot v'$ qui intervient dans la fonctionnelle $a(u, v)$ s'écrit sous forme discrète pour l'élément numéro 1 :

$$\begin{aligned} (v_1^h)' \cdot (u_1^h)' &= {}^T \left(\left\langle \frac{\partial N_1(x)}{\partial x} \quad \frac{\partial N_2(x)}{\partial x} \right\rangle \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} \right) \\ &\quad \times \left(\left\langle \frac{\partial N_1(x)}{\partial x} \quad \frac{\partial N_2(x)}{\partial x} \right\rangle \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \right) \\ &= {}^T \left(\mathbf{B} \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} \right) \left(\mathbf{B} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \right) \\ &= \langle v_1 \ v_2 \rangle^T \mathbf{B} \mathbf{B} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}. \end{aligned}$$

En notant \mathbf{D} , la *matrice de Hooke* (il s'agit de la représentation matricielle du tenseur de rigidité \mathbb{C} ; elle dépend donc du problème d'élasticité considéré : tridimensionnelle, contraintes planes, déformations planes, axisymétrie ; et elle se réduit ici à un scalaire) :

▷ *Matrice de Hooke \mathbf{D}*

$$\mathbf{D} = E, \quad (35)$$

les termes (27) et (28) de la formulation variationnelle relatifs à cet élément se discrétisent respectivement de la manière suivante :

$$\langle v_1 \ v_2 \rangle \left(\int_0^{L/3} {}^T \mathbf{B} \mathbf{D} \mathbf{B} S \, dx \right) \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad (36)$$

et

$$\langle v_1 \ v_2 \rangle \left(\int_0^{L/3} S \rho g \begin{Bmatrix} N_1(x) \\ N_2(x) \end{Bmatrix} dx \right). \quad (37)$$

2.5.6 Matrice de rigidité et second membre élémentaires

Les termes précédents devant être vrais pour tout $\langle v_1 \ v_2 \rangle$ admissible, il convient de considérer les deux quantités suivantes :

$$\underbrace{\left(\int_0^{L/3} {}^T \mathbf{B} \mathbf{D} \mathbf{B} S \, dx \right)}_{=\mathbf{K}_e} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \quad \text{et} \quad \underbrace{\left(\int_0^{L/3} S \rho g \begin{Bmatrix} N_1(x) \\ N_2(x) \end{Bmatrix} dx \right)}_{\{F_e\}}. \quad (38)$$

La matrice \mathbf{K}_e constitue la *matrice de rigidité élémentaire* et se définit dans le cas général sur une maille Ω_i :

▷ *Matrice de rigidité élémentaire \mathbf{K}_e*

$$\mathbf{K}_e = \int_{\Omega_i}^T \mathbf{B} \mathbf{D} \mathbf{B}^T d\Omega. \quad (39)$$

Pour l'élément barre de longueur $L/3$ considéré ici, la matrice gradient est :

$$\mathbf{B} = \frac{1}{L/3} [-1 \ 1] \quad (40)$$

et la matrice de rigidité élémentaire \mathbf{K}_e^{barre} vaut par conséquent :

$$\begin{aligned} \mathbf{K}_e^{barre} &= \int_0^{L/3} \left(\frac{1}{L/3} \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \right) ES \left(\frac{1}{L/3} [-1 \ 1] \right) dx \\ &= \frac{ES}{(L/3)^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \int_0^{L/3} dx \\ &= \frac{ES}{(L/3)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \end{aligned}$$

Le second membre élémentaire $\{F_e\}$ vaut :

$$\begin{aligned} \{F_e\} &= \int_0^{L/3} S \rho g \begin{Bmatrix} 1 - \frac{x}{L/3} \\ \frac{x}{L/3} \end{Bmatrix} dx \\ &= S \rho g \begin{Bmatrix} L/3 - \frac{1}{2} \frac{(L/3)^2}{L/3} \\ \frac{1}{2} \frac{(L/3)^2}{L/3} \end{Bmatrix} \\ &= \frac{S \rho g (L/3)}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}. \end{aligned}$$

2.5.7 Assemblage

La matrice de rigidité élémentaire et le second membre élémentaire précédents valent pour les autres éléments barres de la discrétisation. Le système global à résoudre doit alors être composé en sachant à quel élément appartient chaque nœud (*connectivité* topologique). Cette information de connectivité permet d'*assembler* le système :

▷ *Assemblage*

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 + \textcircled{1} & \textcircled{-1} & 0 \\ 0 & \textcircled{-1} & \textcircled{1} + \diamond 1 & \diamond -1 \\ 0 & 0 & \diamond -1 & \diamond 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \frac{\rho g (L/3)^2}{2E} \begin{Bmatrix} 1 \\ 1 + \textcircled{1} \\ + \textcircled{1} + \diamond 1 \\ + \diamond 1 \end{Bmatrix}$$

On obtient :

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \alpha \begin{Bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{Bmatrix} \quad \text{où } \alpha = \frac{\rho g (L/3)^2}{2E} \quad (41)$$

Ce système est le système éléments finis global. Il s'écrit classiquement sous la forme :

▷ *Forme d'un système éléments finis global*

$$\mathbf{K} \{U\} = \{F\}, \quad (42)$$

2.5.8 Application des conditions aux limites

Les conditions aux limites imposées $u_1 = 0$. Cette condition peut être appliquée en supprimant du système linéaire suivant la première ligne et la première colonne. On obtient, le sous-système suivant :

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \alpha \begin{Bmatrix} 2 \\ 2 \\ 1 \end{Bmatrix} \quad (43)$$

qui conduit à :

$$\begin{Bmatrix} u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \alpha \begin{Bmatrix} 5 \\ 8 \\ 9 \end{Bmatrix}. \quad (44)$$

2.5.9 Post-traitement

Cette solution approchée peut être comparée à la solution exacte (24). En prenant arbitrairement $\rho g/E = 1\text{mm}^{-1}$, $L = 1\text{m}$ et $E = 1\text{GPa}$, la solution exacte et la solution approchée peuvent être comparées sur la FIG. 6. Cette comparaison indique l'écart entre une solution exacte qui se présente sous la forme d'un polynôme de degré 2 et une solution approchée qui est linéaire par morceaux.

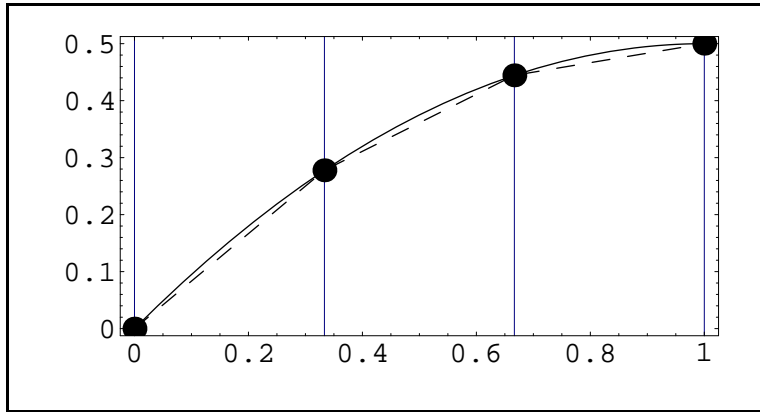


FIG. 6 – Allongement d'une barre sous poids propre en fonction de la position le long de la barre : comparaison entre la solution exacte (trait continu) et la solution approchée pour une discrétisation en trois éléments finis "barre" (points et pointillés).

Il est possible d'estimer sur le champ des contraintes uniaxiales la pertinence de la solution approchée. Si σ_i^h et ε_i^h sont respectivement la contrainte et la déformation dans l'élément numéro i , on a par exemple pour l'élément numéro 1 :

$$\begin{aligned}\sigma_1^h &= \mathbf{D}\varepsilon_1^h = \mathbf{DB} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \\ &= \frac{E}{L/3} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} \\ &= \frac{E\alpha}{L/3} u_2.\end{aligned}$$

En procédant de même pour σ_2^h et σ_3^h , on obtient la FIG. 7. La solution exacte prévoit une contrainte uniaxiale qui évolue linéairement avec la position sur la barre et la solution approchée à l'aide d'éléments finis "barre" prévoit une contrainte uniaxiale constante par morceaux.

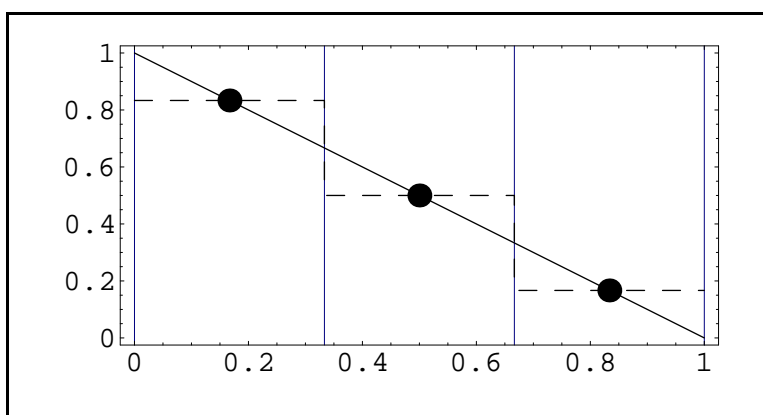


FIG. 7 – Contrainte uniaxiale dans une barre sous poids propre en fonction de la position le long de la barre : comparaison entre la solution exacte (trait continu) et la solution approchée pour une discrétisation en trois éléments finis “barre” (points et pointillés).

3 Introduction à CAST3M

Cette section propose une introduction aux concepts de CAST3M et à son utilisation.

Les paragraphes 4.2 et 4.3, portant respectivement sur les calculs de modes et les calculs dynamiques, sont donnés à titre indicatif. Ils sont en dehors du champ des connaissances requises et à acquérir dans le cadre de cette introduction à la méthode des éléments finis.

3.1 Carte de visite de CAST3M

CAST3M est un code de calcul par éléments finis, initialement développé par le Département des Etudes Mécaniques et Thermiques (DEMT) du CEA.

Cet outil permet de réaliser toutes les étapes d'une simulation numérique :

- pré-traitement : maillage, mise en données, construction du modèle
- calcul : mise en œuvre de la méthode des éléments finis, résolution du problème discrétisé
- post-traitement : traitement des résultats, visualisation.

CAST3M possède un langage de haut niveau (macro-langage), le GIBIANE, constituant une interface entre l'utilisateur et le programme. Ce langage repose sur un ensemble de commandes simples et offre par association de ces commandes un grande souplesse d'utilisation permettant d'adapter le code aux besoins de l'utilisateur.

Enfin, certaines versions de CAST3M sont distribuées gratuitement. De nombreuses documentations sont aussi disponibles.

3.2 Notions de base

3.2.1 Etapes du déroulement d'un calcul avec CAST3M

Les étapes du déroulement d'un calcul avec CAST3M sont :

1. Définition du modèle mathématique (pré-traitement)

▷ *Etapas d'un calcul avec CAST3M*

- Discrétisation géométrique du domaine : définition de points, lignes, surfaces et volumes
- Type d'analyse : déformations planes, contraintes planes, axisymétrie, Fourier, ...
- Type de comportement : mécanique, fluide, thermique, ...
- Type d'éléments finis (cube à interpolation quadratique, triangle à interpolation linéaire, poutre, coque, ...) et caractéristiques géométriques qui ne peuvent pas être déduites du maillage (épaisseur des coques, section des poutres, ...),
- Conditions aux limites et chargement

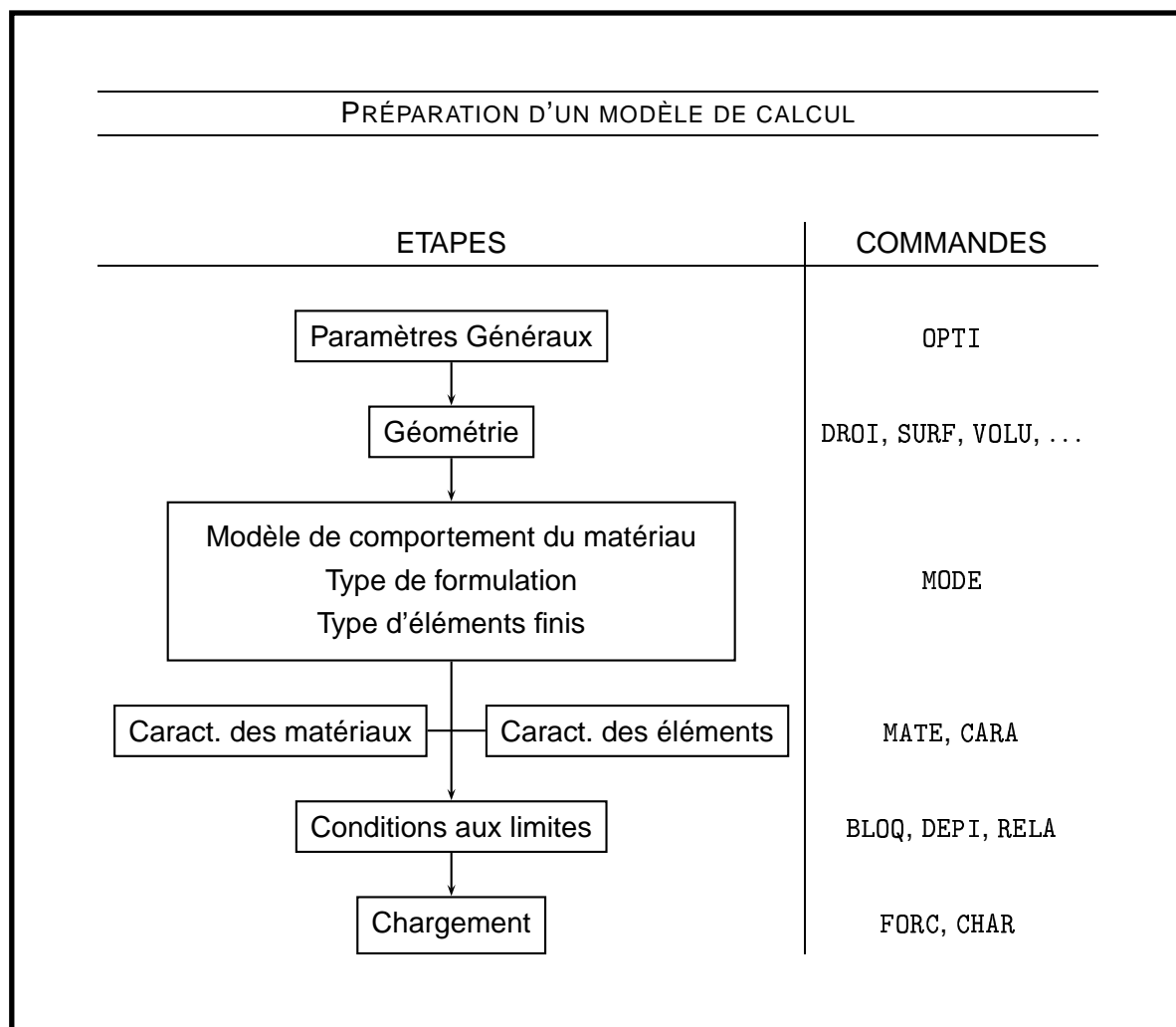
2. Résolution du problème discrétisé (calcul)

- Calcul des matrices élémentaires
- Assemblage
- Applications des chargements externes
- Application des conditions aux limites
- Résolution du système d'équations d'équilibre

3. Analyse des résultats (post-traitement)

- Visualisation de déformées, d'isovaleurs, d'animations,
- Calcul d'énergies, de contraintes équivalentes, ...

L'étape 2 est en général transparente pour l'utilisateur. Son intervention ne concerne que les étapes 1 et 3. La structure de l'étape 1 peut être décrite par le schéma suivant :



3.2.2 Le GIBIANE et ses règles syntaxiques

Le langage GIBIANE permet d'indiquer au programme comment *créer* ou *modifier* un objet. Pour ce faire, deux types d'instructions sont utilisées :

1. **Les directives** permettent de modifier des objets existants ou de les utiliser pour réaliser une action
ex : ELIMINATION 0.0001 GEOM;
ex : TRACER MAILLAGE;
2. **Les opérateurs** permettent de créer de nouveaux objets à partir d'objets existants. Le résultat de la création est affecté à l'aide du signe égal (=)
ex : LIGNE = DROITE POINT1 POINT2;

Ces instructions peuvent être assemblées pour construire des **procédures**. Ces procédures peuvent à leur tour être utilisées comme des instructions, appeler d'autres procédures ou s'appeler elles-mêmes. Des opérations très complexes peuvent ainsi être conçues.

Le GIBIANE possède un certain nombre de **règles syntaxiques** dont les principales sont :

▷ Règles syntaxique du
GIBIANE

- Ecriture en ligne
- Une ligne contient au maximum 72 caractères
- Une ligne peut contenir plusieurs instructions
- Une ligne débutant par * est un commentaire
- Une instruction s'écrit sur 9 lignes maximum
- Une instruction se termine par ;
- Pas de distinction majuscule/minuscule
- Les directives et les opérateurs sont définis par leurs 4 premiers caractères
ex : ELIMINATION, ELIMINA, ELIM
ex : TRACER, TRACE, TRAC
- Les objets et les procédures sont définis par tous leurs caractères et doivent contenir 8 caractères maximum
- Le séparateur est le "blanc" ␣
- Une instruction est interprétée de Gauche à Droite
ex : $1+3*5 \rightarrow 20$
ex : $1+(3*5) \rightarrow 16$
- Les guillemets permettent de prendre un chaîne de caractères in extenso (avec différenciation des majuscules et minuscules)
ex : fin, FIN, 'FIN' arrêtent le calcul
ex : 'fin' n'arrête pas le calcul

3.2.3 Exécution du programme

CAST3M peut être utilisé en batch ou en interactif. Quel que soit le mode d'utilisation :

- la session d'exécution est intégralement mémorisée dans un fichier éditable (fort.98)
- les commandes et les données peuvent être préparées dans un fichier en dehors de CAST3M (*.dgibi), puis exécutées.

A l'usage, il est conseillé de précéder par étapes :

- création de la mise en données (i.e. du pré-traitement) dans un fichier *.**dgibi** et sauvegarde du fichier,
- récupération des informations de la mise en données, lancement de l'analyse et sauvegarde des résultats,
- post-traitement des résultats.

Dans l'exécution du programme, il est nécessaire d'utiliser au moins deux directives :⁽⁴⁾

- OPTI qui permet de déclarer les principaux paramètres du calcul
- FIN qui interrompt l'exécution.

▷ Les 2 directives nécessaires à l'exécution de CAST3M

3.2.4 Système d'unités

CAST3M n'impose aucun système d'unités de mesure. C'est à l'utilisateur de fournir un système d'unités cohérent, vérifiant la loi fondamentale de la dynamique : $F = m\gamma$.

Longueur	Masse	Force	Temps	Masse Volumique	Pression
m	kg	N	s	kg/m^3	$Pa = N/m^2$
mm	$10^3 kg$	N	s	$10^{-12} kg/m^3$	$MPa = N/mm^2$
in	$lb s^2/in$	lb	s	$lb s^2/in^4$	$psi = lb/in^2$

Il existe une exception à cette règle, les angles doivent toujours être exprimés en degrés.

3.2.5 Quelques opérateurs élémentaires

- + - * / . Par exemple somme de deux objets de même type :
I3 = I1 + I2;
- opérateur générant un logique : EGA >EG <EG > < . Par exemple :
BL1 = I1 EGA I2;
- test logique. Par exemple :
SI BL1; ... ; SINON; ... ; FINSI;
- concaténation de deux objets : ET . Par exemple :
P3 = P1 ET P2 ;

⁽⁴⁾A strictement parler, la directive OPTI pourrait être omise car il existe des options par défaut.

3.2.6 Principaux objets

Les objets les plus utiles sont listés ici et leur sens est précisé lorsque cela est nécessaire. Quelques exemples sont donnés.

▷ Objets généraux

ENTIER

FLOTTANT Nombre réel

LISTENTI Liste d'entiers

LISTREEL Liste de réels

LISTEF = 'PROG' 0. 0.5 1. 1. 1. 1. ;

permet de définir une liste de réels. On peut aussi utiliser :

LISTEF = 'PROG' 0. PAS 0.5 NPAS 2 3*1. ;

On définit une seconde liste de réels :

LISTET = 'PROG' 0. PAS 1. 5. ;

EVOLUTION Evolution d'une grandeur en fonction d'un autre à partir de deux listes (i.e. fonction au sens mathématique)

EVOLFT = 'EVOL' 'MANU' 'TEMPS' LISTET 'FORCE' LISTEF ;

définit une fonction réelle à partir de la liste de couples
 $\{(0., 0.), (1., 0.5), (2., 1.), (3., 1.), (4., 1.), (5., 1.)\}$

MOT

LISTMOTS

LOGIQUE Variable logique VRAI/FAUX

TABLE Ensemble d'objets de type quelconque caractérisé par un indice de type quelconque

▷ Objets de maillage

POINT Coordonnées d'un point et densité associée

MAILLAGE Topologie du domaine discret

▷ Objets de calcul

CHPOINT Champ par points⁽⁵⁾ : associe à chaque nœud du maillage des composantes. Ces composantes sont repérées par un nom, soit choisi par l'utilisateur, soit déterminé par l'opérateur qui a créé l'objet

▷ Champ par points

LISTCHPO

MMODEL Association d'un domaine physique, d'un maillage, d'une formulation élément finis et d'un comportement matériau

▷ Modèle

On considère une ligne à 4 nœuds LIG1 (de type MAILLAGE) à laquelle on associe une formulation mécanique, un matériau élastique isotrope et des éléments finis de type coque mince à 2 nœuds :

```
MOD1 = MODEL LIG1 MECANIQUE ELASTIQUE COQ2;
```

MCHAML Champ par éléments⁽⁶⁾ : associe à chaque élément du maillage des composantes.

▷ Champ par éléments

Il est possible de convertir un champ par éléments en un champ par points et réciproquement en utilisant l'opérateur CHANGER.

```
MCHAML  CHAN 'CHPO' → CHPOINT
MCHAML  CHAN 'CHAM' ← CHPOINT
```

On suppose, par exemple, qu'un champ par élément CHAM1 a été défini manuellement sur une ligne LIG1 (à 4 nœuds et donc à 3 éléments), comme suit :

```
CHAM1 = MANU 'CHML' LIG1 1. 2. 3.;
```

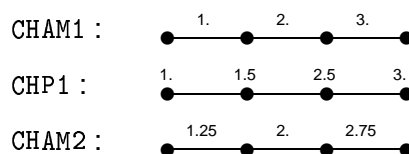
Il est possible de le convertir en champ par points :

```
CHP1 = CHAN 'CHPO' MOD1 CHAM1;
```

puis de nouveau en champ par éléments, en précisant que le nouveau champ prend ses valeurs au centre de 'GRAVITE' des éléments :

```
CHAM2 = CHAN 'CHAM' CHAM1 MOD1 'GRAVITE';
```

On obtient la situation suivante (du fait des moyennes faites par l'opérateur CHANGER :



CHARGEMENT Description spatiale et temporelle d'un chargement.

▷ Objets de post-traitement

VECTEUR Visualisation d'un champ par points à l'aide de vecteurs

DEFORME Caractérisation d'un domaine déformé considéré comme la superposition d'un objet de type MAILLAGE (le domaine initial non déformé) et d'un objet de type CHPOINT (le champ de déplacement).

▷ Déformée

⁽⁵⁾un CHAMPOINT en langage CAST3M

⁽⁶⁾un CHAMELEM en langage CAST3M

3.3 Construction du modèle mathématique

3.3.1 Options générales de calcul

La première étape de la définition d'un modèle consiste à déclarer un certain nombre de paramètres liés au calcul que l'on souhaite effectuer.

Ces paramètres sont déclarés par la directive `OPTI` suivit des paramètres :

- `DIME` la dimension utilisée par les opérateurs de maillage et de calcul :
1, 2, 3 ,
- `MODE` type d'analyse
 - `PLAN CONT` contraintes planes,
 - `PLAN DEFO` déformations planes,
 - `PLAN GENE P1` déformations planes généralisées (P1 point support),
 - `AXIS` axisymétrique,
 - `FOUR NN` analyse en séries de Fourier (NN numéro de l'harmonique),
 - `TRID` tridimensionnel
- `ECHO` niveau de bavardage. 0 pas de données en sortie, 1 toutes les données en sortie,
- `DONN` sert à spécifier un fichier pour lire les données,
- `ELEM` correspond au type d'éléments à fabriquer.

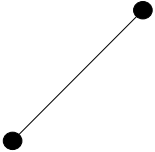
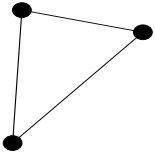
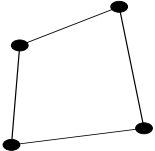
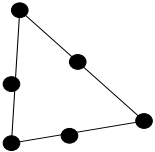
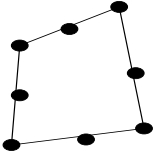
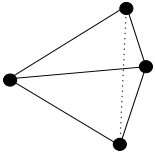
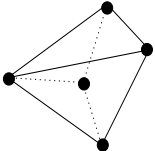
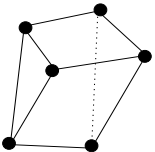
□ Exemple

```
* en 2D
OPTI DIME 2 ELEM QUA4 MODE PLAN CONT;

* en 3D
OPTI DIME 3;
```

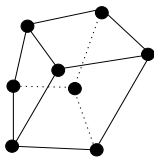

3.3.2 Discrétisation de l'espace

CAST3M dispose d'un certain nombre d'éléments finis, linéaires et quadratiques :

<u>Support géométrique</u>	<u>Type</u>	<u>Eléments finis associés</u>
SEG2		COQ2(axi) POUT,TUYA,BARR ...
TRI3		TRI3,COQ3,DKT ...
QUA4		QUA4,COQ4 ...
TRI6		TRI6,COQ6 ...
QUA8		QUA8,COQ8 ...
TET4		TET4
PYR5		PYR5
PRI6		PRI6

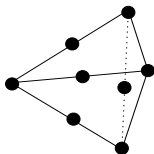
Support géométrique type éléments finis

CUB8



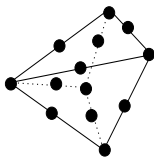
CUB8

TE10



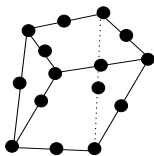
TE10

PY13



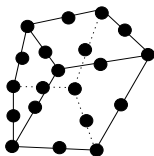
PY13

PR15



PR15

CU20



CU20

Pour créer un maillage sur la base de ces éléments, différents objets géométriques doivent être utilisés.

▷ Création de points

Les points sont créés à partir de leurs coordonnées.

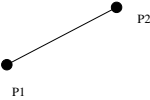
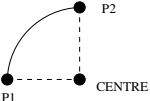
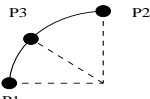
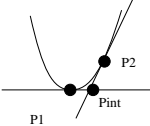
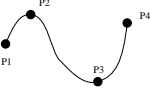
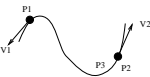

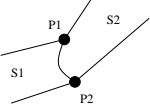
P1 = 0. 0. 0. ; (en 3D)

P2 = 0. 0. ; (en 2D)

CEN1 = ... ;

► Création de lignes et de contours

Les lignes peuvent être créées de différentes manières à partir des points existants.

Type	Définition	Exemples
DROITE (2 points)		LI1= DROI P1 P2 10
CERCLE (3 points)		LI1= CERC P1 CENTRE P2 10
CER3 (3 points)		LI1= CER3 P1 P3 P2 10
PARABOLE (3 points)		LI1= PARA P1 PINT P2 10
CUBP (4 points)		LI1= CUBP P1 P2 P3 P4 10
CUBT (2 pts + 2 vect)		LI1= CUBT P1 V1 P2 V2 10
COURBE (n points)	courbe polynomiale d'ordre n	LI1= COURBE P1 P2 ... PN 10
QUELCONQUE (n points)		LI1= QUEL SEG2 P1 ... PN 10
INTERSECTION (2 pts + 2 surf)		LI1= P1 INTE S1 S2 P2 10

Pour construire un contour, on peut utiliser des lignes déjà existantes :

LI1 = P1 D 10 P2;

LI2 = P2 C 10 P1 P3;

LI3 = P3 D 4 P1;

CONT1 = LI1 ET LI2 ET LI3;

On peut aussi définir un contour directement :

CONT1 = P1 D 10 P2 C 10 P1 P3 D 4;

► Abréviations

DROI(TE) → D

CERC(LE) → C

□ *Exemple*

```

OPTI DIME 2 ELEM SEG2 DENS 1. ;
P0 = 0. 0. ;
P1 = 10. 0. ;
DENS 2. ;
P2 = 10. 10. ;
P3 = 0. 10. ;
CEN1 = 5. 10. ;
L1 = D -10 P0 P1 ;
L2 = D P1 P2 ; N2 = NBEL L2 ;
L3 = D -10 P2 P3 ;
L4 = C ((-1)*N2) P3 CEN1 P0 ;
CONT1 = L1 ET L2 ET L3 ET L4 ;
TRAC CONT1 ;

```

▷ Création de surfaces

Plusieurs opérateurs peuvent être utilisés pour créer des surfaces.

- L'opérateur **SURF** permet de définir des surfaces planes, ou s'appuyant sur une sphère, un cylindre, un cône, un tore

```

SURF1 = SURFACE (CHPO1) | LIG1 | 'PLANE' ;
                        |      | 'SPHERIQUE'  CENTR1 ;
                        |      | 'CYLINDRIQUE' AXEI1 AXEJ1 ;
                        |      | 'CONIQUE'     SOMM1 AXEJ1 ;
                        |      | 'TORIQUE'     CENTR1 AXEJ1 CENTR2
                        |
                        |
                        | 'POLYNOME' N1 N2 P1   P2 (P3 (P4 ... )
                        |              P11 P12 (P13 (P14 ... )
                        |              (P21 (P22 (P23 (P24 ... )
                        |              (
                        |              ...
                        | ('PARAMETRE' U1 U2   V1 V2) ('REGULIER'

```

Par exemple :

```
S1 = SURF (L1 ET L2 ET ... ) PLAN;
```

La ligne définissant le contour peut être constituée de contours extérieurs et intérieurs (délimitant des trous) qui doivent tourner dans des sens opposés.

- Les opérateurs **DALLER** et **REGLER** permettent d'obtenir des maillages réguliers.
- Les opérateurs **TRANSLATION** et **ROTATION** permettent de créer des surfaces par translation et rotation.

```

SURF1 = LIG1 TRANSLATION (N1) ('DINI' DENS1) ('DFIN' DENS2) VEC1 ;
SURF1 = LIG1 ROTA (N1) FLOT1 ('DINI' DENS1) ('DFIN' DENS2) ...
      ... POIN1 (POIN2 si 3D) ;

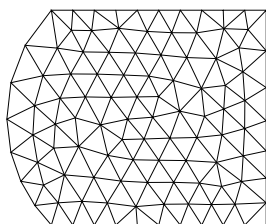
```

□ Exemple

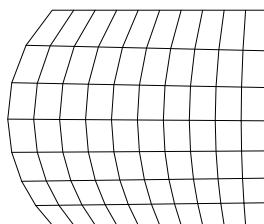
```

OPTI ELEM TRI3 ;
SURF1 = SURF CONT1 'PLAN' ;
TRAC SURF1 ;
OPTI ELEM QUA4 ;
SURF2 = DALL L1 L2 L3 L4 ;
TRAC SURF2 ;
OPTI ELEM TRI3 ;
P4 = 4. 4. ; P5 = 6. 4. ; P6 = 6. 6. ; P7 = 4. 6. ;
L5 = D 5 P4 P7 ;
L6 = D 5 P7 P6 ;
L7 = D 5 P6 P5 ;
L8 = D 5 P5 P4 ;
CONTINT1 = L5 ET L6 ET L7 ET L8 ;
CONTTOT = CONT1 ET CONTINT1 ;
SURF3 = SURF CONTTOT 'PLAN' ;
TRAC SURF3 ;

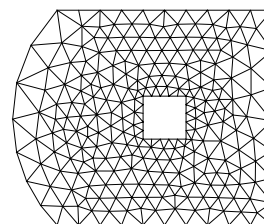
```



SURF1



SURF2



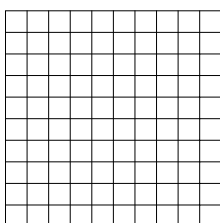
SURF3

□ Exemple

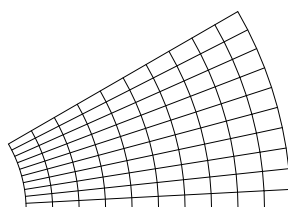
```

S2 = L1 TRANS 10 (0. 10. );
S3 = L1 ROTA 10 30 CEN1;

```



S2



S3

▷ Création de volumes

- L'opérateur VOLU construit le maillage E01 (type MAILLAGE) du volume situé à l'intérieur de l'enveloppe SURF1 (type MAILLAGE).

GE01 = SURF1 VOLU ;

Cet opérateur VOLU permet aussi de construire le volume engendré par translation ou rotation d'une surface.

GE01 = SURF1 VOLU (N1) ('DINI' DENS1) ('DFIN' DENS2) ...

```

...      | 'TRAN'  VEC1      |
      ... | 'ROTA'  FLOT1  AXEI1 AXEJ1 | ;
      ... | 'GENE'  LIG1      |
      ... | SURF2      |

```

- L'opérateur PAVE permet de mailler avec des cubes l'intérieur d'un volume parallélépipédique dont les six faces sont précisées.

VOL1 = PAVE SURF1 SURF2 SURF3 SURF4 SURF5 SURF6 ;

- Un trou dans un volume peut être créé par l'opérateur DIFF qui construit un objet en tant que différence symétrique de deux objets :

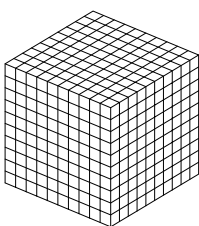
$$a\text{DIFF } b = (a \cup b) - (a \cap b)$$

□ Exemple

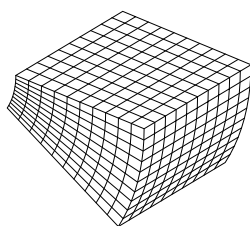
```

VOL1 = SURF6 VOLU 10 'TRANS' VEC3 ;
VOL2 = SURF6 VOLU SURF9 ;

```



VOL1



VOL2

3.3.3 Manipulation d'objets géométriques

▷ Construction d'objets par symétrie

L'opérateur SYME construit l'objet résultant de la symétrie de l'objet GE01 par rapport à un point, une droite ou un plan.

```
GE02 = GE01 SYME | 'POINT' POIN1 ;
                | 'DROIT' POIN1 POIN2 ;
                | 'PLAN' POIN1 POIN2 POIN3 ;
```

▷ Elimination de nœuds

La directive ELIM remplace dans GE01 (type MAILLAGE) tous les nœuds distants de moins de FLOT1 (type FLOTTANT) par un seul point.

```
ELIM (FLOT1) GE01 (GE02) ;
```

▷ Régénérer des éléments

Certaines situations conduisent à créer des éléments ayant des nœuds spatialement confondus. Il peut être intéressant de les remplacer par des éléments où les nœuds doubles n'apparaissent qu'une fois, par exemple triangle pour quadrangle, prisme pour cube.

L'opération se fait en deux temps : il faut d'abord fusionner les nœuds confondus par l'utilisation de l'opérateur ELIMINATION, puis transformer les éléments les contenant par l'utilisation de l'opérateur REGENERER.

```
GE01 = REGE GE02 ;
```

On peut utiliser les passages suivants :

SEG3	→	SEG2		TRI3	→	SEG2
QUA4	→	TRI3		TRI6	→	SEG3
QUA8	→	TRI6		CUB8	→	PR16
CU20	→	PR15				

▷ Trouver des points dans une géométrie

Différentes situations peuvent conduire à la recherche de points dans une géométrie donnée.

- Recherche des points S1 appartenant à l'enveloppe d'un volume VOL1 maillé :

```
S1 = ENVE VOL1;
```

- Recherche du point P1 appartenant à S1 situé aux alentours du point de coordonnées (0. 0.1 0.4) :

```
P1 = S1 POIN PROC (0. 0.1 0.4);
```

- Recherche des points de S1 appartenant à une ligne passant par les points P2 et P3 (au sens de situés à une distance de cette ligne inférieure à une distance donnée DIST (un FLOTTANT))

```
L1 = S1 POIN DROIT P2 P3 DIST;
```

- Recherche des points de S1 appartenant à un plan passant par les points P2, P3 et P4

```
S2 = S1 POIN PLAN P4 P5 P6 DIST;
```


- Récupération d'un nœud donnée de S1

```
N3 = S1 NOEU 3;
```

- Connaître le nombre de nœuds de S1

```
NOVOL1=NBNO S1;
```

► Sauvegarder et restaurer un maillage

Pour sauvegarder un maillage, puis ensuite le relire :

1. En utilisant la commande SORT

- Définir le fichier de sortie :

```
OPTI 'SORT' 'le_nom_de_mon_fichier';
```

- Ecrire le maillage :

```
SORT | GEO1 ('NOOP') | ;
      | 'AVS' (GEO1) (CHPO1) (CHAM1) ('SUIT') ('TEMP' FLO1) |
```

- Définir le fichier d'entrée :

```
OPTI 'LECT' 'le_nom_de_mon_fichier';
```

- Lire le maillage :

```
LIRE (GEO1) ;
```

2. En utilisant la commande SAUV

- Définir le fichier de sortie :

```
OPTI 'SAUV' 'le_nom_de_mon_fichier';
```

- Ecrire les objets OBJET1, ... OBJETi (ici le maillage) :

```
SAUVER ('FORMAT') OBJET1 ... OBJETi ;
```

- Définir le fichier d'entrée :

```
OPTI 'REST' 'le_nom_de_mon_fichier';
```

- Lire le fichier :

```
RESTITUER ( 'FORMAT' ) ;
```

En pratique, les commandes SORT et LIRE ne sont plus utilisées. Elles restent conservées pour des raisons historiques.

► Tracer un maillage

On utilise la commande TRACER.

```
TRAC ((OEIL1) si 3D) | ('QUAL') | OBJET1 ;
                      | ('NOEUD') |
                      | ('COUL' ( COUL1 ) ) |
                      | ('ELEM') |
                      | ('CACH') |
                      | ('FACE') |
                      | ('ECLA' ( RAPP1 ) ) |
                      | ('COUPE' POIN1 POIN2 POIN3 ) |
                      | ('SECT' POIN1 POIN2 POIN3 ) |
                      | ('TITR' 'bla bla...') |
                      | ('NCLK') |
```

3.3.4 Définition du modèle de calcul

Une fois la géométrie définie, il faut définir le type de comportement du matériau dont on a discrétisé la géométrie, ainsi que le type d'élément.

L'opérateur `MODE` (`MODELISER`) permet d'associer à un maillage une formulation, un modèle de comportement du matériau, un type d'élément fini à utiliser et éventuellement un nom de constituant.

```
MODL1 = MODE  GEO1 FOR1  MAT1 ( ...MATn )      ....
          ... ( ELEM1 ... ELEMn ) ('INTEGRE' N1 ) ( 'CONS'  MOT1  ) ;
```

où `FOR1` est la formulation définie par un ou plusieurs mots choisis parmi :

		'MECANIQUE'
		'LIQUIDE'
- formulations simples :		'THERMIQUE'
		'CONVECTION'
		'POREUX'
		'DARCY'
		'FROTTEMENT'
		'RAYONNEMENT'
- formulation couplée :		'LIQUIDE' 'MECANIQUE'

□ Exemple

```
MO = MODE GEO1 MECANIQUE ELASTIQUE COQ3;
où MO est un objet de type MMODEL.
```

▷ Exemple de *MMODEL*

3.3.5 Définition des caractéristiques matériaux et structurelles

L'opérateur `MATE` (`MATERIAU`) crée un champ de propriétés matérielles et/ou géométriques. Pour les éléments qui nécessitent des propriétés matérielles et géométriques, on peut soit les introduire toutes à la fois par `MATE`, soit introduire les propriétés matérielles par `MATE` et les propriétés géométriques par `CARA` puis fusionner les deux champs ainsi obtenus par `ET`.

□ Exemple

```
MA1 = MATE MO YOUN 2.E11 NU 0.3 RHO 7.85E3 EPAI 0.001;
où MA1 est un objet de type MCHAML, sous-type CARACTERISTIQUES
```

▷ Exemple de *matériau*

De manière équivalente on peut écrire :

```
MA = MATE MO YOUN 2.E11 NU 0.3 RHO 7.85E3;
CA = CARA MO EPAI 0.001;
MA1 = MA ET CA;
```

3.3.6 Construction des matrices de rigidité et de masse

Pour la construction de ces matrices on fait appel aux opérateurs RIGI et MASS.

- L'opérateur RIGI calcule la RIGIDITE de différents objets :

```
RIG1 = RIGI  MODL1  MAT1 ( CHAM2 ) ;
RIG1 est de type RIGIDITE de sous-type RIGIDITE.
```

- L'opérateur MASSE calcule les matrices de masse :

```
MASS1 = MASSE  MODL1  MAT1;
MASS1 est de type RIGIDITE de sous-type MASSE.
```

3.3.7 Prise en compte des conditions aux limites

Dans CAST3M, l'application des conditions aux limites se fait par la technique des multiplicateurs de Lagrange. Comme indiqué au paragraphe 2.4.4, cette technique nécessite l'utilisation de la matrice de rigidité **K** et d'une matrice **D** portant sur les multiplicateurs inconnus.

Dans CAST3M, l'utilisateur doit réaliser explicitement l'union de la matrice **K** et de la matrice **D**, tandis que le vecteur des DDL imposés $\{\bar{U}\}$ doit être ajouté au vecteur des chargements équivalents nodaux $\{F\}$.

- L'opérateur BLOQUE construit la rigidité RIG1, associée à des conditions de valeurs imposées sur les inconnues d'un problème discrétisé. Cette rigidité sera ultérieurement à adjoindre à la rigidité de la structure.

```
RIG1=BLOQUE|('MAXI')| |          MOT1 ...          | GEO1 ;
              |('MINI')| |          'RADIAL' POIN1 (POIN2)          |
                  |          'ORTHO'  POIN1 (POIN2)          |
                  |('DEPL')('ROTA') | ('DIRECTION' VEC1)|
```

- L'opérateur DEPI spécifie la valeur de certains blocages.

```
CHPO1 = DEPI  RIG1 | FLOT1 |      ;
              |          |
              | CHPO2 |
```

□ Exemple

```
RIG1 = 'BLOQ' 'UX'  GEO1;
FO1  = 'DEPI' RIG1 0.3 ;
```

▷ Exemple de prise en compte de conditions aux limites

Les noms des degrés de liberté possibles sont :

- degrés de liberté mécaniques
 - pour un calcul en MODE PLAN CONT : UX UY
 - pour un calcul en MODE PLAN DEFO : UX UY
 - pour un calcul en MODE PLAN GENE : UX UY RZ UZ RX RY
 - pour un calcul en MODE AXIS : UR UZ RT
 - pour un calcul en MODE FOUR : UR UZ UT RT
 - pour un calcul en MODE TRID : UX UY UZ RX RY RZ
- degrés de liberté liquides : P PI
- degrés de liberté thermiques : T
- degrés de liberté darcy : TH

3.3.8 Définition de chargements

Hormis les variables imposées (déplacement, rotation, ...), vu précédemment, on peut envisager d'autres types de chargements :

- L'opérateur FORCE construit un champ de forces résultant de l'application d'une force ponctuelle.

```
FORC1 = FORCE | VEC1 | GE01 ;
            | MOTi VALi ... |
```

Les noms de forces possibles sont :

```
pour un calcul en mode PLAN CONT : FX FY
pour un calcul en mode PLAN DEFO : FX FY
pour un calcul en mode PLAN GENE : FX FY FZ(*)
pour un calcul en mode AXIS : FR FZ
pour un calcul en mode FOUR : FR FZ FT
pour un calcul en mode TRID : FX FY FZ
```

La force VEC1 est répartie sur les différents points de GE01. C'est à dire que si GE01 contient 50 points, la force appliquée sur chaque point est 1/50 de VEC1 (ou 1/50 de VALi).

□ Exemple

```
|| F01 = FORC (0. 1.) P2;
```

▷ Exemple d'application
d'une force

- L'opérateur PRES calcule les forces nodales équivalentes d'une pression appliquée sur un objet.

```
CHP01=PRES| 'MASS' MODL1 | FLOT1 GE01 | ;
           |              | CHP02      |
           |
           | 'COQU' MODL1 | FLOT1      | | 'NORM'          | (CAR1);
           |              | CHP02      | | ('DIRE') VEC1    |
           |              |              | | 'POIN' POIN1 |
```

```

|
| 'FISS' MODL1 | FLOT1      | VEC1 POIN1 CAR1 ;
|              | CHP02      |
|
| 'TUYA' MODL1  CAR1 ;

```

- L'opérateur CHAR construit un objet CHAR1 de type CHARGEMENT, de sous-type FORCE, contenant la description spatiale et temporelle du chargement. On peut associer à l'objet CHARGEMENT un nom (donnée facultative).

```

CHAR1 = CHAR (MOT) | MCHAML | EVOL
              | CHPOINT |
              | TABLE1  TABLE2

```

où MOT est une donnée facultative, type CHARACTER*4. Dans le cas de l'utilisation de la procédure PASAPAS ce mot est indispensable.

- la temperature	T
- les déplacements imposes	DIMP
- les temperatures imposees	TIMP
- Les autres chargements (meca)	MECA
- Pression suiveuse	PSUI
- Flux (en consolidation)	FLUX
- Les flux de chaleur (thermique)	Q
- Les temperatures ext (thermique)	TE
- Les variables exterieures	***

3.4 Post-traitement des contraintes, opérateur SIGMA

L'opérateur SIGMA calcule un champ de contraintes à partir d'un champ de déplacements. Pour certains éléments, il s'agit d'efforts (barres, poutres, tuyaux), pour d'autres il s'agit de contraintes généralisées (coques minces). Les contraintes sont calculées dans le repère général pour les éléments massifs et dans le repère local pour les éléments coques, plaques, poutres.

Contraintes calculées				
Éléments finis en formulation MECANIQUE				
Elément fini	Option de calcul	Noms des contraintes	Repère de calcul	Points Supports
CERC	AXIS FOUR	EFFX	local	noeud
BARR	PLAN CONT PLAN DEFO TRID	EFFX	local	centre de gravité
COQ2	PLAN CONT PLAN DEFO AXIS FOUR	N11,NZZ,M11,MZZ N11,NZZ,M11,MZZ N11,N22,M11,M22 N11,N22,N12, M11,M22,M12	local	Points de Gauss
POUT TUYA	TRID	EFFX,EFFY,EFFZ, MOMX,MOMY,MOMZ	local	noeuds
TIMO	TRID	EFFX,EFFY,EFFZ, MOMX,MOMY,MOMZ	local	centre de gravité
TUFI	TRID	EFFX,EFFY,EFFZ, MOMX,MOMY,MOMZ KI ,AIRE	local	centre de gravité
TRI3 QUA4 TRI6 QUA8 ICT3	PLAN CONT PLAN DEFO AXIS FOUR	SMXX,SMYY,SMZZ, SMXY SMRR,SMZZ,SMTT, SMRZ SMRR,SMZZ,SMTT,	global	Points de Gauss

	ICT6				SMRZ, SMRT, SMZT					
	ICQ4									
	ICQ8									

	JOI2		TRID		SMSN, SMN		local		Points de	
	JOI3		PLAN DEFO						Gauss	
			AXIS							

	JOI4		TRID		SMS1, SMS2, SMN		local		Points de	
									Gauss	

	COQ3		TRID		N11, N22, N12,		local		centre de	
					M11, M22, M12				gravité	

	DKT		TRID		N11, N22, N12,		local		Points de	
					M11, M22, M12				Hammer	

	DST		TRID		N11, N22, N12,		local		Points de	
					M11, M22, M12				Hammer	
					V1 , V2					

	COQ4		TRID		N11, N22, N12,		local		Points de	
					M11, M22, M12				Gauss et	
					V1 , V2				centre de	
									gravité	

	COQ6		TRID		SMSS, SMTT, SMST,		local		Points de	
	COQ8				SMSN, SMTN				Gauss	

	CUB8		TRID		SMXX, SMYY, SMZZ,		global		Points de	
	TET4				SMXY, SMXZ, SMYZ				Gauss	
	PRI6									
	PYR6									
	CU20									
	TE10									
	PR15									

	LISP		TRID		NZZ, NXZ, NYZ,		local		Points de	
	LISM				MXZ, MZZ, KI				Gauss	

4 Quelques exemples d'analyse

On présente dans cette section différentes possibilités d'analyse avec CAST3M. Ces possibilités sont illustrées par des exemples.

On différencie les problèmes linéaires (cinématique et comportement) des problèmes non-linéaires. Donc dans un premier temps, sont présentés des exemples linéaires de résolution en statique, de calculs de modes, de dynamique. La résolution de problèmes non-linéaires est ensuite abordée.

Pour chaque cas, des exemples de post-traitement des résultats sont donnés.

4.1 Calcul élastique statique

4.1.1 Définition du modèle

* définition des conditions du calcul

```
OPTI DIME 2;
OPTI MODE PLAN DEFO;
OPTI ELEM TRI3;
```

▷ Calcul 2D, déformation plane, maillage avec des triangles à interpolation linéaire

* définition de la géométrie

```
P1 = ...
P2 = ...
GEO1= ...
```

▷ Définition des points et construction de la géométrie

* définition du modèle

```
MO = MODE GEO1 MECANIQUE ELASTIQUE TRI3;
MA = MATE MO YOUN 2.E11 NU 0.3;
```

▷ Calcul mécanique élastique, sur la géométrie GEO1 maillée avec des TRI3, constituée d'un matériau élastique isotrope

* définition des conditions aux limites

```
LI1 = GEO1 POIN DROIT P1 P2 1.E-6;
CL1 = BLOQ DEPL LI1;
```

▷ Extraction de la ligne P1-P2 sur laquelle on bloque les déplacements

* definition du chargement

```
PF = GEO1 POIN PROC (0. 10.);
FO1 = FORC (0. 20.) PF;
```

▷ Extraction d'un point sur lequel on applique une force selon Y

* construction de la matrice rigidite

```
RI1 = RIGI MO MA;
```

4.1.2 Résolution

* resolution

```
DEP1= RESOU (RI1 ET CL1) FO1;
```

▷ Résolution du système $(\mathbf{K} + \mathbf{CL})\{U\} = \{F\}$, puis $\text{DEPL1} \leftarrow \{U\}$

4.1.3 Post-traitement

* Calcul de la deformee

```
DEF1 = DEFO GEO1 DEP1 1.;
```

▷ Déformée = géométrie
initiale + 1.*(déplacement)

* Calcul des contraintes

```
SI1 = SIGMA MO MA DEP1;
```

▷ $\sigma = DBDEP1$

* Visualisation de sigxx sur la deformee

```
TRAC GEO1 MO (EXCO 'SMXX' SI1 'SMXX') DEF1 ;
```

▷ Utilisation de EXCO pour
extraire la composante xx
de σ , et visualisation

* Calcul de la contrainte de von-mises

```
VM1 = VMIS MO SI1;
```

▷ Utilisation de VMIS pour
calculer la contrainte de
Von Mises de SI1

* Visualisation de VM1 sur la deformee

```
TRAC GEO1 MO VM1 DEF1 ;
```

```
FIN;
```

4.2 Calcul de modes

On suppose que l'on a créé :
une géométrie GE01
une matrice de rigidité RAID1
une matrice de masse MASS1
des conditions aux limites CL1

L'opérateur VIBRATION recherche les valeurs et les modes propres d'un système physique représenté par sa rigidité et sa masse.

```

                                | 'PROCHE'      ... |
SOL1 = VIBRATION | 'INTERVALLE' ... |  RIG1 MASS1  ...
                                | 'SIMULTANE'   ... |

                                ...   ('IMPR') ('TBAS') (LOG1) ;

```

Admettons qu'on souhaite calculer au plus les 10 premières fréquences dans la bande 1-100 Hz :

```

BFREQ = 100.;
AUTO = VIBR INTER 1. BFREQ BASSE 10 (RAID1 et CL1) MASS1;

```

Pour connaître le nombre de modes réellement calculé :

```

NBMODES = DIME AUTO;

```

Pour afficher les déformées des modes au fur et à mesure, on peut faire comme suit :

```

* on crée une table qui va contenir les fréquences
FRE2 = TABLE;
* on crée une table qui va contenir les modes
MOD = TABLE;
* on crée une table qui va contenir les maillages déformés
DEF = TABLE
* on fait une boucle sur le nombre de modes
I = 0;
REPETER BLOC1 (DIME AUTO);
  I = I + 1;
  FRE2.I = TIRE AUTO FREQ RANG I;
  MESS ' MODE ' I ;
  MESS ' -----';
  MESS ' Fréquence calculée : ' FRE2.I 'Hz';
  SAUT 1 LIGN;
* ----- visualisation du mode -----
* on attend une réponse au clavier
MESS 'SOUHAITEZ VOUS VISUALISER CE MODE ? (O/N)';
OBTE REP*MOT ;
SI ( ( EGA REP 'O') OU ( EGA REP 'o') );

```

▷ Opérateur VIBR : AUTO est un objet de type SOLUTION ou de type TABLE si 'TBAS'

▷ Utilisation d'une TABLE

▷ Interrogation de l'utilisateur avec OBTE

```
* on extrait le mode qui est un objet de type CHPOINT
* de AUTO un objet de type SOLUTION :
  MOD.I = TIRE AUTO DEPL RANG I;
* pour visualiser simultanément la structure non déformée
* et déformée. On construit deux objets de type DEFORME ayant
* le même MAILLAGE avec un CHPOINT (déplacement) donné:
  DEFO = DEFO GEO1 MOD.I 0.;
  DEF.I = DEFO GEO1 MOD.I 1. VERT;
* on affiche les objets de type DEFORME
  TITRE 'MODE' I ' barre 3D FREQ. : ' FRE2.I ' Hz';
  TRACE EYE (DEFO ET DEF.I) CACH;
* ----- faire une animation du mode -----
* on attend une réponse au clavier
  MESS 'SOUHAITEZ VOUS VISUALISER UNE ANIMATION DE CE MODE ? (O/N)';
  OBTE REP*MOT ;
  SI ( (EGA REP 'O') OU (EGA REP 'o') );
* on extrait le mode qui est un objet CHPOINT de AUTO un objet
* de type SOLUTION :
  MOD.I = TIRE AUTO DEPL RANG I;
* on fait varier la structure autour de sa position d'équilibre.
* on translate la structure de depart :
  DEPL GEO1 MOINS MOD.I;
* on cree une serie d'images a partir du maillage translate :
  CINE = ANIME 20 GEO1 MOD.I 2;
* on affiche avec TRAC et l'option OSCI
  TRAC OEIL CINE OSCI;

* on translate le MAILLAGE dans sa position initiale
  DEPL GEO1 PLUS MOD.I;
  FINSI;
*
  FINSI;

FIN BLOC1;
```

4.3 Calcul dynamique

Si on souhaite résoudre un problème de dynamique il existe plusieurs possibilités dépendant de la sollicitation. Le problème discret que l'on souhaite résoudre est le suivant :

$$M\ddot{U}(t) + C\dot{U}(t) + KU(t) = F(t) \quad (45)$$

Différents cas se présentent :

- la sollicitation est harmonique, c'est à dire que $F(t) = \bar{F}e^{i\omega t}$. Alors en régime permanent on va avoir une réponse de la forme $U = \bar{U}e^{i\omega t}$. Ce qui permet, en négligeant les termes d'amortissement, d'écrire (45) de façon suivante :

$$(-\omega^2 M + K)\bar{U} = \bar{F} \quad (46)$$

Donc pour une pulsation ω donnée on va déterminer \bar{U} et on pourra obtenir $U = \bar{U}e^{i\omega t}$. Cette approche est aussi utilisable quand lorsqu'on décompose $F(t)$ en série de Fourier on obtient un faible nombre de termes,

- la sollicitation est dans une plage de fréquences réduites, et basses fréquences (domaine modal). Dans ce cas on se sert des propriétés d'orthogonalité des modes, c'est à dire :

$$\psi_i^T M \psi_j = \delta_{ij} = \mu_j \quad (47)$$

$$\psi_i^T K \psi_j = \delta_{ij} \sigma_j \quad (48)$$

$$\lambda_j = \frac{\mu_j}{\sigma_j} \quad (49)$$

où ψ_j est un mode propre de valeur propre λ_j associée, μ_j est la masse généralisée et σ_j la raideur généralisée. Ainsi si on note :

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \quad (50)$$

et :

$$\Psi = \begin{bmatrix} \psi_1 & \cdots & \cdots & \psi_n \end{bmatrix} \quad (51)$$

Alors si on décompose le vecteur solution sur la base des modes :

$$U = \sum_{i=1}^n q_i \psi_i = \Psi q \quad (52)$$

on obtient :

$$\Psi^T M \Psi \ddot{q} + \Psi^T K \Psi q = \Psi^T F$$

$$\begin{aligned} & \Updownarrow \\ & \begin{bmatrix} \mu_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mu_n \end{bmatrix} \begin{pmatrix} \ddot{q}_1 \\ \vdots \\ \ddot{q}_n \end{pmatrix} + \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n \end{bmatrix} \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix} = \begin{pmatrix} \psi_1^T F \\ \vdots \\ \psi_n^T F \end{pmatrix} \end{aligned}$$

qui est un système de n équations différentielles découplées.

Si la sollicitation est dans une gamme de fréquences limitées on tronque le nombre de modes à N on obtient un système $N \times N$ réduit.

Après on peut résoudre avec un grand nombre de méthodes.

- cas général. On effectue une intégration numérique du système (45), par exemple par Newmark :

$$\begin{aligned} M\ddot{U}_{n+1} + KU_{n+1} &= F_{n+1} \\ \dot{U}_{n+1} &= \dot{U}_n + (1 - \gamma)h\ddot{U}_n + \gamma h\ddot{U}_{n+1} \\ U_{n+1} &= U_n + h\dot{U}_n + h^2\left(\frac{1}{2} - \beta\right)\ddot{U}_n + \beta h^2\ddot{U}_{n+1} \end{aligned}$$

4.3.1 Réponse à une sollicitation harmonique

On suppose que l'on a créé GEO1, RAID1, MASS1, CL1 (comme précédemment).

On souhaite résoudre le système linéaire :

$$(\text{RAID1} - \omega^2 \text{MASS1}) U = F$$

On peut procéder comme suit :

```

REPETER BOUCLE;
* on attend la fréquence au clavier
MESS' DONNEZ LA FREQUENCE D EXCITATION' ;
OBTE FR*FLOTTANT ;
SI (FR EGA 0.);
QUITTER BOUCLE;
FINSI;
* on calcule  $\omega^2$  et on affiche car on est parano
W2 = -1.*((2*3.1416*FR)**2) ;
MESS' w2 : ' W2 ;
* calcul de K -  $\omega^2$  M
MMASW2 = W2*MMAS1;
KEQ = (RAID1 ET CL1) ET MMASW2 ;
* on résoud pour FE impose ailleurs
SOL = RESOU KEQ FE;
* on visualise une animation de la réponse
* translation ...
DEPL GEO1 MOINS SOL;
* ... construction des déformées ...
CINE=ANIME 20 GEO1 SOL 2;
SI (NEG GRAPH 'N');
  TRAC EYE CINE OSCI;
FINSI;
* ... translation
DEPL GEO1 PLUS SOL;

```

4.3.2 Superposition modale

Calcul d'une réponse dynamique à l'aide d'algorithmes explicites, par l'algorithme de Fu - de Vogelaere.

Il s'agit de calculer la solution du système d'équations :

$$M\ddot{q} + C\dot{q} = F(q, t) \quad (55)$$

avec :

$$F(q, t) = -Kq + Fl + Fe$$

$$q(0) = Q0 \text{ et } \dot{q}(0) = L0$$

M : matrice diagonale des masses généralisées

C : matrice des amortissements modaux

K : matrice diagonale des raideurs généralisées

Fl : vecteur des forces de liaisons

Fe : vecteur des forces extérieures

Q : vecteur des contributions modales

$Q0$: vecteur des contributions modales initiales

$L0$: vecteur des vitesses modales initiales

Le principe de l'algorithme est le suivant :

- déplacement généralisé premier demi-pas :

$$q_{i,m+1/2} = q_{i,m} + \frac{h}{2}\dot{q}_{i,m} + \frac{h^2}{24}(4G_{i,m} - G_{i,m-1/2} - A_i(4\dot{q}_{i,m} - \dot{q}_{i,m-1/2}))$$

- vitesse généralisée premier demi-pas :

$$\dot{q}_{i,m+1/2} = \frac{4}{4 + A_i h}(\dot{q}_{i,m} + h/4(G_{i,m} + G_{i,m+1/2} - A_i\dot{q}_{i,m}))$$

- déplacement généralisé deuxième demi-pas :

$$q_{i,m+1} = q_{i,m} + h\dot{q}_{i,m} + h^2/6(G_{i,m} + 2G_{i,m+1/2} - A_i(\dot{q}_{i,m} + 2\dot{q}_{i,m+1/2}))$$

- vitesse généralisée premier demi-pas :

$$\dot{q}_{i,m+1} = \frac{6}{6 + A_i h}(\dot{q}_{i,m} + h/6(G_{i,m+1} + 4G_{i,m+1/2} + G_{i,m} - A_i(4\dot{q}_{i,m+1/2} + \dot{q}_{i,m})))$$

L'opérateur qui permet de résoudre ce type de problème est :

TAB1 = DYNE 'DE_VOGELAERE' ...

```
... | TAB2      | (TAB4) (TAB5) | TAB6      | N1 FLOT1 (N2) TAB8 ;
    | TAB3      |                | TAB7      |
    | TAB2 TAB3 |                | TAB6 TAB7 |
```

avec :

- TAB2 une table représentant une base modale ou un ensemble de bases modales (type TABLE). Créée par VIBR option 'TBAS' ou VIBR suivi de TRADUIRE.

- TAB4 une table représentant la matrice des amortissements généralisés (type TABLE). Créé par AMORTISSEMENT.
- TAB6 une table représentant l'évolution des forces libres appliquées (type TABLE). Créé par CHARGEMENT.
- TAB7 une table donnant les conditions initiales (type TABLE).
- TAB8 une table définissant les résultats que l'on veut dans la table de sortie TAB1 (type TABLE).
- N1 le nombre de pas (type ENTIER).
- FLOT1 le pas de temps (type FLOTTANT).
- N2 la sortie tous les N2 pas de calcul (type ENTIER), par défaut $N2 = 1$.
- TAB1 la table contenant les résultats.

```

*****
*   REPONSE TRANSITOIRE D'UNE POUTRE *
*****
OPTION ECHO 1 ;
OPTION DIME 3 ELEM CUB8 ;
opti rest 'poutre.mdl';
restituer;
*
* traduction du calcul des modes en qqch de
* lisible par dyne -> BASE MODALE
*
TT                = TRADUIRE auto;
BA_MODES          = TABLE 'BASE_MODALE' ;
BA_MODES.'MODES'  = TT;
*
* creation de la matrice d'amortissement modal
*
AMOR1              = AMOR BA_MODES ( PROG 10. 10. 10. 10. );
*
TA                = TABLE 'AMORTISSEMENT' ;
TA.'AMORTISSEMENT' = AMOR1 ;
*
PF = poin (enve s) proc ( 0.01 0.005 0.01);
PE2 = poin (enve s) proc ( 0.5 0.005 0.005);
PM2 = poin (enve s) proc ( 0.25 0.005 0.005);
*
* CHARGEMENT
*
* - FORCE PONCTUELLE
F1 = forc (0. 0. 100.) PE2;
*
* - EXITATION A 10 Hz
LI1 = PROG 0. PAS 2.5E-5 1.01;
LI2 = PROG 'SINU' 10. LI1;
EV = EVOL MANU T LI1 F(T) LI2 ;
*
CHA1 = CHAR 'FORC' F1 EV ;
CHA2 = PJBA CHA1 BA_MODES;
TABCHAR = TABLE 'CHARGEMENT' ;
TABCHAR.'BASE_A' = CHA2 ;
*

```

```

* TABLE DE SORTIE
*
TABSOR = TABLE 'SORTIE' ;
TABVAR = TABLE 'VARIABLE' ;
TABSOR.'VARIABLE' = TABVAR ;
TABVAR.'DEPLACEMENT' = VRAI ;
TABVAR.'VITESSE' = FAUX ;
TABVAR.'ACCELERATION' = FAUX ;
*
* APPEL A DYNE DE_VOGELAERE
*
* avec amortissement
TABRES = DYNE DE_VOGELAERE BA_MODES TA TABCHAR TABSOR 40000 2.5E-5 20 ;
* sans amortissement
TABRES2 = DYNE DE_VOGELAERE BA_MODES TABCHAR TABSOR 40000 2.5E-5 20 ;
*
lt      = prog;
luz1    = prog;
luz2    = prog;
nbres   = (dime tabres) - 3 ;
i        = 0;
repetier boucle nbres;
  i      = i+1;
  lt     = lt et (prog (extr TABRES.TEMPS_DE_SORTIE i ));
  dep    = RECO TABRES BA_MODES (extr lt i) DEPL;
  luz1   = luz1 et (prog (EXTR DEP UZ PE2));
  detr   dep;
  dep    = RECO TABRES2 BA_MODES (extr lt i) DEPL;
  luz2   = luz2 et (prog (EXTR DEP UZ PE2));
  detr   dep;
fin boucle;
EV1 = EVOL BLEU  MANU  'TEMPS' LT  'UZ ' LUz1 ;
EV2 = EVOL ROUGE MANU  'TEMPS' LT  'UZ ' LUz2 ;
*
* ---  TRACER LES COURBES
*
ev      = ev * 0.025;
xsup    = 40000*2.5E-5;
DESS (EV1 ET EV2) DATE LOGO TITR 'POUTRE ENCASTREE'
TITX 'TEMPS' TITY 'DEPLACEMENT' XBOR 0. xsup 'LEGE' ;
fin;

```

4.3.3 Dynamique transitoire

La procédure `DYNAMIC` permet d'effectuer un calcul dynamique pas à pas. Elle utilise l'algorithme de Newmark centré ($\beta = 1/4$ $\gamma = 1/2$). Le pas de temps est déterminé à partir d'une fréquence de coupure donnée par l'utilisateur. Les résultats sont calculés à des instants définis par l'utilisateur. Les arguments d'entrée et de sortie de la procédure sont des tables.

TAB1 = DYNAMIC TAB2 ;

```
TAB2.'DEPL' . 'VITE' . 'CHAR'
      . 'CHAR' . 'RIGI' . 'MASS'
      . 'FREQ' . 'INST' . 'DEBU'
      . 'SAUV'
```

avec les champs de TAB2 :

- 'DEPL' : le déplacement initial (type CHPOINT)
- 'VITE' : la vitesse initiale (type CHPOINT)
- 'CHAR' : le chargement (type CHARGEME)
- 'RIGI' : la rigidité (type RIGIDITE)
- 'MASS' : la masse (type RIGIDITE)
- 'AMOR' : l'amortissement (type RIGIDITE) (donnée optionnelle)
- 'FREQ' : la fréquence de coupure (type FLOTTANT)
- 'INST' : les instants de sortie (type LISTREEL)
- 'DEBU' : le temps initial (type FLOTTANT) (donnée optionnelle . 0 par défaut)
- 'SAUV' : mot 'OUI' si on veut (type MOT) sauver les résultats (donnée optionnelle)

TAB1 est un objet de type table contenant autant de tables que d'instants de sortie. Si N1 est le numéro de l'instant. La table TAB1.N1 contient :

- 'TEMP' : l'instant (type FLOTTANT)
- 'DEPL' : le déplacement (type CHPOINT)
- 'VITE' : la vitesse (type CHPOINT)

```

*****
*   REPONSE TRANSITOIRE D'UNE POUTRE *
*****
OPTION ECHO 1 ;
OPTION DIME 3 ELEM CUB8 ;
*
*   MAILLAGE en mètres
*
P1 = 0. 0. 0. ;
P2 = 0. 0.01 0. ;
L1 = D 2 P1 P2 ;
S1 = L1 TRANS 2 (0. 0. .01);
V1 = S1 VOLU TRANS 50 (.5 0. 0.);
PE = POIN (ENVE V1) PROC ( 0.5 0.005 0.01);
PM = poin (enve v1) proc ( 0.25 0.005 0.01);
PF = poin (enve v1) proc ( 0.01 0.005 0.01);
*
*
MOD1 = MODELE V1 MECANIQUE ELASTIQUE CUB8;
MAT1 = MATER MOD1 YOUN 2.E11 NU 0.3  RHO 7800. ;
*
CL1 = BLOQ DEPL S1;
CL2 = BLOQ PE UZ;
*
RIG1 = RIGIDITE MOD1 MAT1;
*
MAS1 = MASSE MOD1 MAT1 ;
*
* -- PREPARATION DES DONNEES POUR L'OPERATEUR DYNAMIC
*
* CHARGEMENT
*
* - FORCE PONCTUELLE
*
F1 = forc (0. 0. 100.) PE;
*
* - excitation a 10 Hz
*
LI1 = PROG 0. PAS 2.5E-5 1.1;
LI2 = PROG 'SINU' 10. LI1;
EV = EVOL MANU T LI1 F(T) LI2 ;

```

```

* module de l'excitation
CHA1 = CHAR 'FORC' F1 EV ;
LIS1 = PROG 0. PAS 0.001 1.01 ;
*
* la frequence de coupure donne => pas_de_temps=0.25/f_cp
*
TAB2 = TABLE;
TAB2.'DEPL' = MANU CHPO V1 3 UX 0. UY 0. UZ 0. ;
TAB2.'VITE' = MANU CHPO V1 3 UX 0. UY 0. UZ 0. ;
TAB2.'CHAR' = CHA1 ;
TAB2.'RIGI' = RIG1 ET CL1;
TAB2.'MASS' = MAS1 ;
TAB2.'FREQ' = 500.;
TAB2.'INST' = LIS1 ;
TAB1 = DYNAMIC TAB2 ;
*
* EXTRACTION DU DEPLACEMENT EN FONCTION DU TEMPS
* ... les temps
LT = PROG ;
* ... les deplacement de PE et PM
LUR1 = PROG ;
LUR2 = PROG ;
I = 0 ;
NBB = (DIME LIS1) - 1 ;
REPETER BB1 NBB ;
  I = I + 1 ;
  LT = LT ET( PROG(TAB1 . I . 'TEMP' ));
  DEP1 = TAB1 . I . 'DEPL' ;
  LUR1 = LUR1 ET( PROG (EXTR DEP1 UZ PE) );
  LUR2 = LUR2 ET( PROG (EXTR DEP1 UZ PM) );
FIN BB1 ;
EV1 = EVOL vert MANU 'TEMPS' LT 'UZ' LUR1 ;
EV2 = EVOL jaune MANU 'TEMPS' LT 'UZ' LUR2 ;
*
* --- TRACER LES COURBES
TABDD = TABLE ;
TABDD.1 = CHAINE 'TITR DEPLACEMENT DE PE EN UZ' ;
TABDD.2 = CHAINE 'TITR DEPLACEMENT DE PM EN UZ' ;
TITR 'VIBRATION D UNE POUTRE' ;
DESS (EV1 ET EV2) TABDD 'LEGE' ;
FIN;

```

4.3.4 Calcul non linéaires

En mécanique, la procédure PASAPAS permet d'effectuer un calcul non linéaire incrémental. La non linéarité peut provenir soit du matériau (plasticité) soit des grands déplacements soit de conditions unilatérales (contact, frottement) soit des trois à la fois. Les résultats sont calculés à des valeurs du paramètre d'évolution (pseudo temps ou temps réel) définies par l'utilisateur.

En thermique, cette procédure permet d'effectuer un calcul linéaire ou non-linéaire en tenant compte de la conduction, de la convection et du rayonnement.

Il est possible d'effectuer un calcul couplé.

```
PASAPAS TAB1 ;
```

TAB1. AMORTISSEMENT	LCAR
AUTOCRIT	LVIAM
AUTORESUS	MAXITERATION
AUTOMATIQUE	MODELE
AUTOPAS	MAXISOUSPAS
BLOCAGES_THERMIQUES	MTOL
BLOCAGES_MECAIQUES	MASSE_CONSTANTE
CONN	MOVA
CONTACT	NON_LOCAL
CELSIUS	PRECISION
CARACTERISTIQUES	PRECDECHARGE
CHARGEMENT	PROCEDURE_PERSO1
CRITERE	PROCEDURE_PERSO2
CTE_STEFAN_BOLTZMANN	PROCEDURE_THERMIQUE
CONSOLIDATION	PROJECTION
CONTRAINTES	PAS_MAX
DYNAMIQUE	RELAXATION_THETA
DEFORMATIONS_INELASTIQUES	RAYONNEMENT
DEPLACEMENTS	RELAXATION_DUPONT
FTOL	SOUS_RELAXATION
FREA1	TEMPS_CALCULES
FROTTEMENT	TEMPS_SAUVES
GRANDS_DEPLACEMENTS	TALPHA_REFERENCE
GRANDES_ROTATIONS	VARIABLES_INTERNES
GRANDES_DEFORMATIONS	VITESSES
K_SIGMA	REACTIONS
K_TANGENT	REPRISE

En sortie, TAB1 permet de retrouver les résultats. Ceux ci sont mis dans des tables dont les indices sont des entiers (0 1 2 ... N) correspondants aux numero de sauvegarde des resultats (0 : temps initial).

indice	objet pointé	commentaires
ERREUR	LOGIQUE	VRAI si erreur au cours de l'exécution de la procédure
CONV	LOGIQUE	VRAI si calcul convergé avant ou pour le nombre maximum d'itérations
TEMPS	TABLE	contient les temps de calcul (correspond à TEMPS_SAUVES)
TEMPERATURES	TABLE	contient les temperatures
DEPLACEMENTS	TABLE	contient les déplacements
CONTRAINTES	TABLE	contient les contraintes
VARIABLES_INTERNES	TABLE	contient les variables internes en nonlinéaire matériau
DEFORMATIONS_INELASTIQUES	TABLE	contient les déformations inélastiques en nonlinéaire matériau
VITESSES	TABLE	contient les vitesses
REACTIONS	TABLE	contient les forces de reactions
REPRISE	ENTIER	Indique que l'on veut reprendre le calcul à partir du nième pas.

□ Exemple

Pour lister le CHPOINT de déplacements calculés pour la valeur du paramètre d'évolution 2.5, il faudra coder :

```
LIST (PECHE TAB1 DEPLACEMENTS 2.5 ) ;
```

ou si on connaît son numero de sauvegarde i :

```
LIST (TAB1.DEPLACEMENTS.i) ;
```

Remarques :

À l'issue d'un calcul effectué avec PASAPAS il est possible de le continuer en utilisant la même table TAB1. Pour pouvoir faire ceci, il faut :

- mettre dans TAB1.'REPRISE' le numéro NUMPAS du pas à partir duquel le calcul doit recommencer. Si rien n'est mis, le calcul reprendra à partir du dernier instant calculé.
- redéfinir les LISTREEL associés aux indices 'TEMPS_CALCULES' et 'TEMPS_SAUVES' de la table TAB1. Si (une de) ces listes contien(nen)t des valeurs inférieures à TAB1.'TEMPS'.NUMPAS, ces valeurs seront ignorées.

La reprise du calcul peut être demandée soit dans la même session de CAST3M (sans quitter le programme), soit dans une nouvelle session. Dans ce dernier cas, la table TAB1 doit être préalablement sauvée avec la directive SAUV, et restituée dans la nouvelle session avec la directive REST.

```

opti echo 1;
opti dime 2;
opti mode plan defo;
P1 = 0. 0.;
P2 = 0. 1.;
opti elem seg2;
l1 = P1 D 4 P2;
opti elem qua4;
s1 = l1 trans 12 (3. 0.);
*trac s1;
* le modele
mo = mode s1 MECANIQUE ELASTIQUE PLASTIQUE qua4;
* construction de la courbe de traction
* sig_eq = 40. + 100. * esp_p
limelas=40.;
pentelas=100.;
myoung=20000.;
plime=limelas/myoung;
HdE=pentelas/myoung;
*
sig00 = limelas/(1.+HdE);
H00 = pentelas/(1+HdE);
*dtrac=plime+0.00001;
list0= prog plime 2.;
list1= prog 0. et list0;
list2= prog 0. et (prog line A H00 B sig00 list0);
brautrac = evol manu list1 list2;
dess brautrac;
* le materiau
ma = mate mo YOUN 20000. NU 0.3 trac brautrac;
* blocage mecanique
cl1 = bloq UX l1;
cl = cl1 et (BLOQ UY p1);
* chargement
pc = poin s1 proc (3. 1.);
clb = bloq ux pc;
clc = bloq uy pc;
fc = depi clc -1. ;
* definition de la table d'entree de pasapas
xtab=table;
* chronologie du chargement

```

```

ltime=prog 0. 1.;
lchar=prog line A 1. B 0. ltime;
evc = evol manu ltime lchar;
cc = char dimp fc evc;
* temps calcules
tcalc= prog 0. pas 0.02 1.;
* incre de sortie
tres = prog 1.;
*
xtab.blocages_mecaniques = (cl et clc et clb);
xtab.caracteristiques = ma;
xtab.modele= mo ;
xtab.chargement = cc;
xtab.temps_calcules=tcalc;
xtab.temps_sauves=tres;
bl1 = VRAI;
xtab.grands_deplacements=bl1;
xtab.grandes_rotations=bl1;
xtab.grandes_deformations=bl1;
xtab.k_sigma=bl1;
xtab.k_tangent=bl1;
*xtab.automatique=bl1;
xtab.precision=1.e-6;
* calcul
pasapas xtab;
* post traitement
nres = dime xtab.contraintes;
i=0;
repeter bou (nres-1);
i=i+1;
sig1 = xtab.contraintes.i;
dep1 = xtab.deplacements.i;
def1 = defo s1 dep1 1.;
epseq1=xtab.variables_internes.i;
*trac s1 mo (exco 'SMXX' sig1 'SMXX') def1;
trac s1 mo epseq1 def1;
fin bou;

```

4.3.5 Calcul fluide acoustique, couplage fluide structure

Il faut définir le modèle comme suit :

```
MODLIQ = MODE GEO LIQUIDE LPR6;
```

et le matériau :

```
MATLIQ = MATE MODLIQ RHO 1.2 RORF 1.2 ...  
... CSON 340. CREF 340. LCAR 1. G 0.;
```

et on construit l'équivalent des matrices de masse et de raideur :

```
MRIG = RIGI MODLIQ MATLIQ;
```

```
MMAS = MASS MODLIQ MATLIQ;
```

En cas de couplage il faut construire la géométrie du couplage (éléments de couplage) :

```
LIALM = LIAI (ENVE LIQ) (ENVE SOL) 1.e-3;
```

Attention à l'ordre des surfaces.

Il faut définir le modèle de la liaison :

```
MODLIA = MODE LIALM MECANIQUE LIQUIDE LIA3;
```

Il faut définir une caractéristique supplémentaire, qui permet de spécifier quel est le maillage du fluide :

```
CARLIA = CARA MODLIA LIQU LIQ;
```

4.3.6 Calcul d'une fonction de transfert

Il faut avoir calculé les modes de la structure (et du fluide ...) : AUTO .

Il faut construire la table des taux d'amortissement modaux :

```
AMOR = PROG (DIME AUTO) * 0.01;
```

Il faut donner les fréquences à calculer :

```
LFREQ = PROG 1. PAS 0.1 BFREQ;
```

Faire le calcul :

```
FTRANS = TRANSFER AUTO AMOR UZ PE UZ PM ...  
...LFREQ DEPL MOPH;
```

Affichage, d'abord on extrait ce qui nous intéresse, ensuite on trace :

```
LMODU = EXTRA FTRANS 'ORDO' 1;  
EVO1 = EVOL MANU 'FREQ(Hz)' LFREQ 'MODULE' LMODU;  
DESS EVOL1 LOGY;
```

4.3.7 Calcul avec contact et frottement

Pour réaliser un calcul avec contact et frottement il faut :

1. Définir un maillage de contact, compose d'elements SEG2 suivant les zones des contours susceptibles d'entrer en contact (objet deformable, objet rigide ...) cf IMP0. L'orientation du contour est tres importante !
2. puis :
 - (a) Définir un maillage de frottement cf IMPF
 - (b) Définir un modele de frottement cf MODE
 - (c) Définir un modele de frottement cf MODE
3. et enfin
 - (a) passer le modele et le materiau de frottement a PASAPAS
 - (b) passer les maillages contact et frottement a PASAPAS

Voici une illustration sur le test GRECO (μ_f est le coeff de frottement).

```

opti dime 2 mode plan defo;
* maillage de la structure
h=40.;
P1 = 0. 0. ;
P2 = h 0.;
P3 = h h;
P4 = 0. h;
opti elem seg2;
l1 = D 32 P1 P2;
l2 = D 32 P2 P3;
l3 = D 32 P3 P4;
l4 = D 32 P4 P1;
opti elem qua4;
s1 = daller plan l1 l2 l3 l4;
* maillage de l'obstacle
p1c=-10. 0.;
p2c= 50. 0.;
opti elem seg2;
l1c = D 1 P2c P1c;
* maillage de contact
mc = impo mail l1 l1c ;
* maillage de frottement
mf = impf mc;
* modele stucture
mo = mode s1 mecanique elastique qua4;
ma = mate mo young 130000 nu 0.2;
* modele frottement

```

```

mof = mode mf frottement coulomb;
muf = 1.3;
maf = mate mof mu muf;
* condition aux limites obstacle
cl1c = bloq depl l1c;
* condition aux limites axe vertical
cl2 = bloq ux l2;
cl2b = bloq uy P2;
* chargement vertical
f3 = PRES MASS MO 50. l3;
* chargement horizontal
f4 = PRES MASS MO 100. l4;
* chronologie du chargement
ltime=prog 0. 1.;
lchar=prog line A 1. B 0. ltime;
evc = evol manu ltime lchar;
cc = char meca (f3 et f4) evc;
* temps calculés
tcalc= prog 0. 1.;
* incre de sortie
tres = prog 1.;
*****
xtab=table;
xtab.blocages_mecaniques = (cl1c et cl2 et cl2b);
xtab.caracteristiques = (ma et maf);
xtab.modele= (mo et mof);
xtab.chargement = cc;
xtab.temps_calculés=tcalc;
xtab.temps_sauves=tres;
bl1 = VRAI;
xtab.grands_deplacements=bl1;
xtab.grandes_rotations=bl1;
xtab.grandes_deformations=bl1;
xtab.k_sigma=bl1;
xtab.contact=mc;
xtab.frottement=mf;
xtab.precision=1.e-6;
* calcul
pasapas xtab;
dep = xtab.deplacements.1;
* trace resultat

```

```

def = defo s1 dep 300. rouge;
def0 = defo s1 dep 0.;
trac (def et def0);
*
reac = xtab.reactions.1;
* postraitement
* on trace les déplacements horizontaux et verticaux
* des noeuds de la zone de contact
* idem, on trace fn,ft,mu*fn (mu*fn pour verifier que ft <= mu*fn
ua = prog;
va = prog;
xa = prog;
fn = prog;
ft = prog;
mufn = prog;
nb = nbno l1;
i=0;
repeter bcl1 nb;
i=i+1;
pa = l1 poin i;
ua = ua et (prog (extr dep ux pa));
va = va et (prog (extr dep uy pa));
xa = xa et (prog (coor 1 pa));
fn = fn et (prog (extr reac fy pa));
ft = ft et (prog (extr reac fx pa));
fin bcl1;
ev1 = evol manu xa ua;
ev2 = evol manu xa va;
ev3 = evol rouge manu 'X' xa 'FN' fn;
ev4 = evol bleu manu 'X' xa 'FT' ft ;
ev5 = evol vert manu 'X' xa 'MU*FN' (-1.*muf*fn);
TAB1=table;
tab1.1 = MOT 'TITR FN' ;
tab1.2 = MOT 'TITR FT' ;
tab1.3 = MOT 'TITR MU*FN' ;
dess ev1;
dess ev2;
dess ( ev3 et ev4 et ev5) tab1 ;

```


5 Travaux Dirigés

5.1 Premier contact avec CAST3M

```

* DEBUT DU FICHIER
* Calcul 2D d'une plaque en flexion 3 points
*
*
*          |
*          | PY
*         \|/
*  -----
*  |               |   | H
*  -----
*  /\               /\
*  <----->
*          2*L
*
* qui donne par symétrie
*
*          |
*          | PY
*         \|/
*  -----<      -
*  |           |<   | H
*  -----<      -
*  /\
*  <----->
*          L
*
*****
* definition des conditions generales de calcul
*****
*
TITR 'FLEXION 3 POINTS EN 2D' ;
OPTI DIME 2;
OPTI MODE PLAN DEFO;
OPTI ELEM TRI3;
*

```

▷ Symétrie géométrique +
symétrie matérielle +
symétrie de chargement

▷ Calcul 2D, Déformation
plane, Éléments
triangulaires à
interpolation linéaire

```
*****
* parametrisation
*****

* dimensions
H      = 5.E-2 ;
L      = 0.5 ;

* maillage
NH     = 10 ;
NL     = 10*NH ;

* materiau elastique lineaire
MODYOUNG = 2.E+11;
COEFPOIS = 0.2;

* chargement
PY     = -6.E2;
*****

* geometrie et maillage
*****

* points
P1  = 0. 0. ;
P2  = 0. H  ;
P3  = L  H  ;
P4  = L  0. ;

* lignes
P1P2 = P1 DROI NH P2 ;
P2P3 = P2 DROI NL P3 ;
P3P4 = P3 DROI NH P4 ;
P4P1 = P4 DROI NL P1 ;

* surfaces
S1   = SURF (P1P2 ET P2P3 ET P3P4 ET P4P1) PLAN ;
* visualisation du maillage avec qualificateurs
TRAC QUAL S1;
*****

* modele et materiau
*****

MOD1 = MODE S1 MECANIQUE ELASTIQUE ISOTROPE TRI3 ;
MAT1 = MATE MOD1 YOUN MODYOUNG NU COEFPOIS;
```

▷ Définition des paramètres utiles

▷ Longueur des variables < 8 caractères

▷ Définition des points (coordonnées X et Y en 2D)

▷ Création de lignes avec l'opérateur DROI

▷ Création d'une surface S1 avec l'opérateur SURF

▷ Tracé du maillage S1 avec les QUALIFICATEURS

▷ Opérateur MODE et MATE

```

*****
* conditions aux limites
*****
CL1 = BLOQ DEPLA P1 ;
CL2 = BLOQ UX P3P4 ;
CL3 = (CL1 ET CL2);
*****
* chargement
*****
* creation
FOR1 = FORCER FY PY P3;
* visualisation
VEC1 = VECT FOR1 2.E-4 FX FY ROUG;
TRAC VEC1 S1 ;
*****
* matrice de rigidite
*****
RIG1 = RIGI MOD1 MAT1;
RIG1 = RIG1 ET CL3;
*****
* resolution
*****
DEPL1 = RESO RIG1 FOR1;
*****
* postraitement
*****
* trace maille initial et deforme
DEFO = DEFO S1 DEPL1 0. BLAN;
DEF1 = DEFO S1 DEPL1 BLEU;
TRAC (DEFO ET DEF1);
* trace idem avec resultante aux appuis
REA1 = REAC DEPL1 RIG1 ;
VEC2 = VECT REA1 2.E-5 FX FY VERT ;
TRAC VEC2 S1 ;
* contrainte de von Mises sur la deformee
SS = SIGM MOD1 MAT1 DEPL1;
SSVV = VMIS MOD1 SS;
TRAC DEF1 MOD1 SSVV;
* fin
FIN;

```

▷ Création des conditions aux limites avec l'opérateur BLOQ

▷ $u_x = u_y = 0$ en P1

▷ $u_x = 0$ sur P3P4

▷ Application en P3 d'une force d'intensité PY orientée selon Y (FY) à l'aide de l'opérateur FORCER

▷ Construction de la matrice \mathbf{K} avec RIGI, et modification de \mathbf{K} par application des conditions aux limites

▷ RESOUD $\mathbf{K}\{U\} = \{F\}$ et $\{U\} \rightarrow \text{DEPL1}$

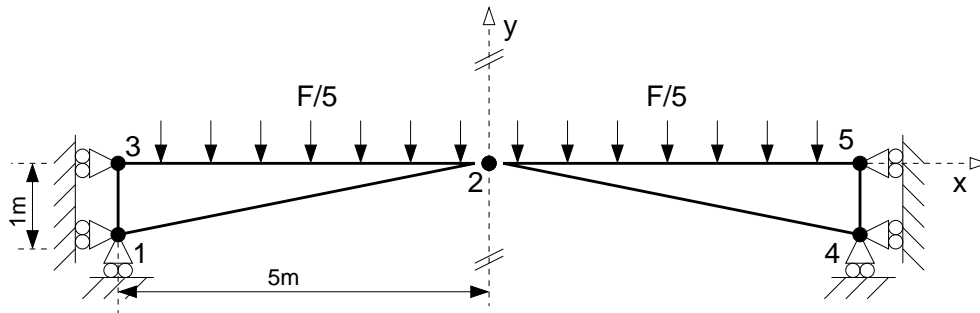
▷ Maillage déformé = maillage initial $+\alpha\{U\}$.
DEFO=initial ($\alpha = 0$)
DEF1=déformé ($\alpha \neq 0$)

▷ Extraction des réactions aux appuis avec REAC

▷ Calcul du champ de contraintes SS avec SIGM et calcul de la contrainte de Von Mises avec VMIS

5.2 Problème Pont à deux travées

5.2.1 Enoncé



Soit un pont à deux travées maillé à l'aide de deux éléments triangulaires à interpolation linéaire (T3, TRI3 dans CASTEM). Ce pont a une largeur de 1 mètre (dans la direction z perpendiculaire au plan de la figure). Il est constitué d'un matériau élastique (module de Young E et coefficient de Poisson $\nu = 0$) et chaque travée supporte une charge F uniformément répartie sur la longueur et l'épaisseur du pont. Les autres conditions aux limites et dimensions sont indiquées sur la figure ci-dessus. On cherche à connaître la déformation du pont sous ce chargement en faisant l'hypothèse des contraintes planes.

► Question 1

Préciser brièvement comment ce problème peut être ramené au traitement d'un problème à deux inconnues (par exemple : les déplacements verticaux des nœuds 2 et 3, soit V_2 et $V_3^{(7)}$) portant sur un seul T3 (par exemple celui défini par les nœuds 1, 2 et 3, noté T3G).

► Question 2

Définir la carte permettant de passer du T3 de référence à T3G et calculer son jacobien.

► Question 3

Calculer la matrice (notée $[K]$) et le second membre du système linéaire 2x2 dont le vecteur des inconnues $\begin{Bmatrix} V_2 \\ V_3 \end{Bmatrix}$ est solution. Quelle est la flèche du pont au nœud 2 par rapport à celle au nœud 3 ?

► Question 4

Donner la valeur maximale de F permettant de respecter le critère de Von-Mises

$$\sqrt{\sigma_{11}^2 + \sigma_{22}^2 + 2\sigma_{11}\sigma_{22} + 3\sigma_{12}^2} \leq \sigma_e/2$$

où σ_e est la limite d'élasticité du matériau constituant le pont.

⁽⁷⁾ Les déplacements selon x seront noté U et ceux selon y seront notés V

5.2.2 Corrigé

▷ Question 1

Compte tenu des symétries (géométrique, matérielle, de chargement), le problème se ramène au traitement d'une seule travée du pont, par exemple TG3. Sur TG3, le vecteur des degrés de liberté est :

$$\langle U^e \rangle = \langle U_1, V_1, U_2, V_2, U_3, V_3 \rangle. \quad (56)$$

Le nœud 1 étant encastré ($U_1 = V_1 = 0$), le plan de symétrie du problème passant par le nœud 2 ($U_2 = 0$) et le nœud 3 ayant son déplacement selon x bloqué ($U_3 = 0$), les seules inconnues du problème sont V_2 et V_3 :

$$\langle U^e \rangle = \langle V_2, V_3 \rangle.$$

▷ Question 2

La carte τ permettant de passer du T3 de référence (noté T3R) à T3G s'écrit :

$$\begin{aligned} \tau : \text{T3R} &\longmapsto \text{T3G} \\ \begin{Bmatrix} \xi \\ \eta \end{Bmatrix} &\longrightarrow \tau(\xi, \eta) = \begin{Bmatrix} x(\xi, \eta) \\ y(\xi, \eta) \end{Bmatrix} = \begin{Bmatrix} N_i(\xi, \eta)x_i \\ N_i(\xi, \eta)y_i \end{Bmatrix}, \\ &= N^e \cdot \begin{Bmatrix} x^e \\ y^e \end{Bmatrix} \quad (\text{notation}) \end{aligned}$$

avec les fonction d'interpolation du T3 et les coordonnées des nœuds :

$$\begin{cases} N_1(\xi, \eta) = 1 - \xi - \eta \\ N_2(\xi, \eta) = \xi \\ N_3(\xi, \eta) = \eta. \end{cases} \quad \text{et} \quad \begin{cases} x_1 = -5 & y_1 = -1 \\ x_2 = 0 & y_2 = 0 \\ x_3 = -5 & y_3 = 0 \end{cases} \quad (57)$$

on obtient la carte et matrice jacobienne suivantes :

$$\begin{cases} x = 5\xi - 5 \\ y = \xi + \eta - 1 \end{cases} \Rightarrow J = \begin{bmatrix} x, \xi & x, \eta \\ y, \xi & y, \eta \end{bmatrix} = \begin{bmatrix} 5 & 0 \\ 1 & 1 \end{bmatrix}; \quad \boxed{\det J = 5} \quad (58)$$

▷ Question 3

Le théorème des puissances virtuelles (formulation variationnelle) s'écrit :

$$\int_{\Omega} \langle \delta \varepsilon \rangle D\{\varepsilon\} d\Omega = \underbrace{\int_{\Omega} f \cdot \delta u d\Omega}_{=0} + \int_{\partial\Omega_F} F \cdot \delta u dS. \quad (59)$$

Sa forme discrétisée est :

$$\sum_e \int_{V^e} \langle \delta \varepsilon \rangle D\{\varepsilon\} dV^e = \sum_e \int_{V^e \cap \partial\Omega_F} F \cdot \delta u dS^e. \quad (60)$$

où \sum_e indique la somme sur les mailles V^e telles que $\Omega = \cup V^e$. D'autre part, on a :

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = N^e \cdot \{U^e\} \quad \text{et} \quad \begin{Bmatrix} \delta u \\ \delta v \end{Bmatrix} = N^e \cdot \{\delta U^e\} \quad (61)$$

$$\{\varepsilon\} = B \cdot \{U^e\} \quad \text{et} \quad \{\delta \varepsilon\} = B \cdot \{\delta U^e\} \quad (62)$$

avec B la matrice opérateur déformation (matrice gradient), qui est en contraintes planes est de dimension $3 \times (2N_{\text{bnoe}})$ (où N_{bnoe} est le nombre de nœuds dans l'élément considéré) et qui s'écrit :

$$B = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \dots \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & \dots \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \dots \end{bmatrix}. \quad (63)$$

Calcul de la matrice de raideur K Le problème considéré ici ne possédant que la maille T3G, le membre de gauche de l'équation (60) devient :

$$e \int_{\text{T3G}} \langle \delta U^e \rangle B^T D B \{U^e\} dx dy \quad (64)$$

où e est l'épaisseur de l'élément réel. Ce qui sur l'élément de référence T3R s'écrit :

$$\langle \delta U^e \rangle \left(e \int_{\text{T3R}} B^T D B \det J d\xi d\eta \right) \{U^e\} \quad (65)$$

$$\Rightarrow \boxed{K = e \int_{\text{T3R}} B^T D B \det J d\xi d\eta}.$$

Le calcul final de K nécessite d'expliciter la matrice de Hooke D et la matrice gradient B .

En contraintes planes, la matrice D , vaut :

$$D = E \begin{bmatrix} \frac{1}{1-\nu^2} & \frac{\nu}{1-\nu^2} & 0 \\ \frac{\nu}{1-\nu^2} & \frac{1}{1-\nu^2} & 0 \\ 0 & 0 & \frac{1}{2(1+\nu)} \end{bmatrix} \underbrace{=}_{\nu=0} E \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}. \quad (66)$$

Par ailleurs, on a (pour $i = 1, 2, 3$) :

$$\left. \begin{aligned} N_{i,x} &= N_{i,\xi}\xi_{,x} + N_{i,\eta}\eta_{,x} \\ N_{i,y} &= N_{i,\xi}\xi_{,y} + N_{i,\eta}\eta_{,y} \end{aligned} \right\} \Rightarrow \begin{cases} N_{1,x} = 0 & N_{2,x} = 1/5 & N_{3,x} = -1/5 \\ N_{1,y} = -1 & N_{2,y} = 0 & N_{3,y} = 1 \end{cases}$$

D'où

$$B = \begin{bmatrix} 0 & 0 & 1/5 & 0 & -1/5 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1/5 & 1 & -1/5 \end{bmatrix}. \quad (67)$$

Des équations (66) et (67), on pourrait tirer l'expression complète de la matrice $B^T D B$. Cependant les seules inconnues du problème étant V_2 et V_3 , on peut se ramener à une matrice B réduite où seul les termes associés aux inconnues V_2 et V_3 sont pris en compte. Si $\{U^e\} = \{V_2, V_3\}$, l'équation (62) indique que les seuls termes de B à considérer dans la forme (67) sont

les colonnes 4 et 6, respectivement associées à V_2 (seconde composante du nœud 2) et à V_3 (seconde composante du nœud 3). D'où :

$$B_{\text{réduite}} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1/5 & -1/5 \end{bmatrix} \Rightarrow B^T DB = \frac{E}{50} \begin{bmatrix} 1 & -1 \\ \text{sym.} & 51 \end{bmatrix}. \quad (68)$$

En prenant $e = 1$, on trouve finalement :

$$K = B^T DB \int_{\text{T3R}} \underbrace{\det J}_{=5} d\xi d\eta = \frac{5}{2} B^T DB = \frac{E}{20} \begin{bmatrix} 1 & -1 \\ \text{sym.} & 51 \end{bmatrix} \quad (69)$$

Calcul du second membre Le second membre de (60) se réduit à une intégration sur [23] :

$$e \int_{[23]} \langle \delta u, \delta v \rangle \left\{ \begin{array}{c} F_x \\ F_y \end{array} \right\} ds \quad (70)$$

Pour réaliser cette intégration sur le segment [23], on considère l'élément ligne à interpolation linéaire (élément L2) qui va du nœud 2 au nœud 3 paramétré par l'abscisse s , dont l'élément de référence est paramétré par $-1 \leq \xi \leq 1$. La nouvelle carte τ' est telle que :

$$s = \tau'(\xi) = N_1(\xi)s_1 + N_2(\xi)s_2 \text{ avec } \begin{cases} N_1(\xi) = (1 - \xi)/2 \\ N_2(\xi) = (1 + \xi)/2 \end{cases} \quad (71)$$

et $s_1 = \sqrt{x_2^2 + y_2^2} = 0$ et $s_2 = \sqrt{x_3^2 + y_3^2} = \sqrt{25 + 0} = 5$. D'où

$$s = \frac{5}{2}(1 + \xi) \quad \text{et} \quad ds = \frac{\partial s}{\partial \xi} d\xi = \frac{5}{2} d\xi \quad (72)$$

De là, en remarquant que $F_x = 0$, $F_y = F/5$, $e = 1m$, en utilisant le changement de variable (72), la transposée de la seconde équation de (61) avec les fonctions d'interpolations de l'élément L2

$$\langle \delta u, \delta v \rangle = \langle \delta U^e \rangle \cdot N^e = \langle \delta V_2, \delta V_3 \rangle \cdot \left\{ \begin{array}{c} N_1 \\ N_2 \end{array} \right\} \quad (73)$$

on trouve que le second membre (70) s'écrit :

$$\langle \delta V_2, \delta V_3 \rangle \left(\frac{F}{5} \int_{-1}^1 \left\{ \begin{array}{c} N_1 \\ N_2 \end{array} \right\} \frac{5}{2} d\xi \right) = \boxed{\langle \delta V_2, \delta V_3 \rangle \cdot \left\{ \begin{array}{c} F/2 \\ F/2 \end{array} \right\}}. \quad (74)$$

Résolution Le TPV discrétisé (60) s'écrit alors (équations (69) et (74)) :

$$\langle \delta V_2, \delta V_3 \rangle \frac{E}{20} \begin{bmatrix} 1 & -1 \\ \text{sym.} & 51 \end{bmatrix} \left\{ \begin{array}{c} V_2 \\ V_3 \end{array} \right\} = \langle \delta V_2, \delta V_3 \rangle \cdot \left\{ \begin{array}{c} F/2 \\ F/2 \end{array} \right\} \quad (75)$$

Le système linéaire à résoudre est donc :

$$\begin{bmatrix} 1 & -1 \\ \text{sym.} & 51 \end{bmatrix} \left\{ \begin{array}{c} V_2 \\ V_3 \end{array} \right\} = \frac{10F}{E} \left\{ \begin{array}{c} 1 \\ 1 \end{array} \right\} \Rightarrow \boxed{V_3 = \frac{2}{5} \frac{F}{E} \quad \text{et} \quad V_2 = 26V_3} \quad (76)$$

▷ **Question 4**

Les contraintes s'expriment par :

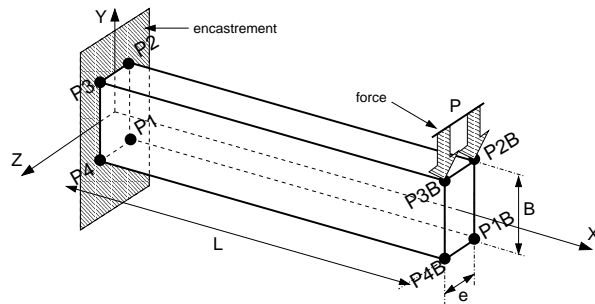
$$\begin{aligned} \{\sigma\} &= D \{\varepsilon\} = DB \{U^e\} \quad \text{où } \langle \sigma \rangle = \langle \sigma_{11}, \sigma_{22}, \sigma_{12} \rangle \\ &= E \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1/5 & 0 & -1/5 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1/5 & 1 & -1/5 \end{bmatrix} \begin{Bmatrix} U_1 \\ V_1 \\ U_2 \\ V_2 \\ U_3 \\ V_3 \end{Bmatrix} \\ &= \begin{Bmatrix} \sigma_{11} = 0 \\ \sigma_{22} = 2F/5 \\ \sigma_{12} = F \end{Bmatrix} \quad (\text{cf } \langle U^e \rangle \text{ réduit à } \langle \delta V_2, \delta V_3 \rangle) \end{aligned}$$

Le respect du critère de Von-Mises, impose ainsi :

$$\boxed{|F| \leq \frac{5\sigma_e}{2\sqrt{79}}} \quad (77)$$

5.3 Problème Poutre

5.3.1 Enoncé



$$\left\{ \begin{array}{l} L = 1m \\ B = 10cm \\ e = 1.2cm \\ E = 200000MPa \\ \nu = 0.3 \\ P = 600N \end{array} \right.$$

On cherche ici à déterminer numériquement à l'aide du logiciel CASTEM la solution du problème de poutre encastree soumise à une sollicitation en flexion présenté sur la figure ci-dessus. Le matériau constitutif de la poutre est homogène, élastique, isotrope. La géométrie, le chargement, les notations et les données numériques sont indiqués sur la figure.

L'objectif est de comparer la solution numérique obtenue à l'aide de différents types d'éléments finis : 1D, puis 3D linéaires et quadratiques.

On rappelle que l'expression analytique de la flèche (ici déplacement selon Y, noté $y(x)$) de la poutre soumise à un tel chargement est (reprendre l'équation de la poutre donnée au TD4) :

$$y(x) = \frac{Px^2}{6EI_z}(3L - x) \quad \text{avec} \quad I_z = \frac{eB^3}{12}. \quad (78)$$

▷ Éléments 1D

Le fichier `poutre_1D_exemple.dgibi` est mis à votre disposition. Il permet de mailler une ligne nommée POUTRE1 à l'aide d'éléments finis de type poutre (POUT, éléments géométriques associés de type SEG2). Compléter la mise en données du problème, la paramétrisation, la description géométrique et le maillage. Vérifier que les conditions aux limites sont correctement appliquées. Résoudre le problème, visualiser la déformée et comparer à la solution analytique. Que peut-on dire de la sensibilité au maillage et de l'influence du coefficient de Poisson sur la solution ?

▷ Éléments 3D linéaires

Le fichier `poutre_3D_exemple.dgibi` est mis à votre disposition. Comme précédemment compléter la mise en données afin de mailler le problème avec des cubes à 8 nœuds.

1. Résoudre le problème et comparer à la solution analytique.
2. Etudier la sensibilité au maillage en respectant la consigne suivante : si n nœuds sont disposés dans l'épaisseur de la poutre, $2n$ nœuds seront mis dans sa hauteur et $16n$ selon sa longueur. Relever pour la flèche en $x = L$, l'évolution du rapport "solution numérique"/"solution analytique" en fonction du nombre de nœuds dans la structure.
3. Visualiser la contrainte σ_{xx} sur le maillage déformé. L'hypothèse "poutre" était-elle justifiée pour ce problème ?

4. Déterminer à partir de quelle longueur l'hypothèse poutre devient vérifiée.

▷ **Eléments 3D quadratiques**

Modifier le fichier `poutre_3D_exemple.dgibi` précédent pour mailler le problème à l'aide de cubes à interpolation quadratique. Répondre aux questions 1 et 2 précédentes. Que dire de la précision des éléments quadratiques ?

5.3.2 Corrigé

Eléments 1D

```
*****
* DEBUT DU FICHIER
*****
*
TITR 'POUTRE MAILLEE 1D' ;
OPTI DIME 3 ELEM SEG2 ;
OPTI ECHO 0;
*
*****
* PARAMETRISATION, DONNEES
*****
* DIMENSIONS
* -----
L      = 1.;
B      = 0.1;
EP     = 0.012 ;
* PARAMETRES DE MAILLAGE
* -----
NL      = 1;
* ELASTICITE
* -----
MODYOUNG = 2.E11;
COEFPOIS = 0.;
* CHARGEMENT
* -----
PY      = -600.;
*****
* GEOMETRIE ET MAILLAGE
*****
* POINTS
* -----
P1      = 0. 0. 0.;
P2      = L 0. 0.;
* LIGNES
* -----
POUTRE1 = P1 DROI NL P2 ;
```

▷ Calcul 3D avec des
éléments poutres (éléments
1D basés sur des segments
à 2 nœuds SEG2)

```
*****
```

```
* MODELE ET MATERIAU
```

```
*****
```

```
MOD1      = MODE POUTRE1 MECANIQUE ELASTIQUE ISOTROPE POUT ;
```

```
*
```

```
MYSECT    = EP*B;
```

```
MYINRY    = (B*(EP**3))/12.;
```

```
MYINRZ    = (EP*(B**3))/12.;
```

```
*
```

```
MAT1      = MATE MOD1 YOUN MODYOUNG NU COEFPOIS ;
```

```
CARA1     = CARA MOD1 SECT MYSECT INRY MYINRY INRZ MYINRZ TORS 1.;
```

```
MAT1      = MAT1 ET CARA1;
```

```
*****
```

```
* CONDITIONS AUX LIMITES ET CHARGEMENT
```

```
*****
```

```
* CL
```

```
* --
```

```
CL1 = BLOQ DEPL ROTA P1 ;
```

```
* CHARGEMENT
```

```
* -----
```

```
FOR1 = FORCER FY PY P2;
```

```
* MATRICE DE RIGIDITE
```

```
* -----
```

```
RIG1 = RIGI MOD1 MAT1;
```

```
RIG1 = RIG1 ET CL1;
```

```
*****
```

```
* RESOLUTION
```

```
*****
```

```
DEPL1 = RESO RIG1 FOR1;
```

```
*****
```

```
* POSTRAITEMENT
```

```
*****
```

```
* COMPARAISON NUMERIQUE / THEORIQUE
```

```
* -----
```

```
DY        = EXTR DEPL1 UY P2 ;
```

```
DYABS     = ABS DY ;
```

```
*
```

```
INERTIE   = (EP*(B**3))/12.;
```

```
DYTHEO    = ABS ((PY*(L**3))/(3*MODYOUNG*INERTIE));
```

```
*
```

▷ *Modèle basé sur des éléments POUT (RE) isotropes*

▷ *Paramétrisation de la section, de I_y et I_z*

▷ *Affectation des coefficients d'élasticité, des CARActéristiques de la poutre*

▷ *Blocage des rotations et déplacements à l'encastrement*

▷ *Application de la force selon Y en P2*

▷ *Extraction de la composante Y du déplacement solution*

▷ *Calcul de la flèche théorique DYTHEO*

```

MESS 'fleche maximale en metre : ' DYABS;
MESS 'fleche theorique en metre : ' DYTHERO;
MESS ' ';
MESS '-----';
MESS '    ->  NUM/THERO =' (DYABS/DYTHERO);
MESS '-----';
* VISUALISATION
* -----
OEIL1=0. 0. 1000.  ;
DEFO = DEFO POUTRE1 DEPL1 0. BLAN ;
DEF1 = DEFO POUTRE1 DEPL1 ROUG ;
TRAC OEIL1 (DEFO ET DEF1) ;
*****
* FIN DU FICHIER
*****
FIN ;

```

▷ Affichage de message
pour comparer la solution
numérique à la solution
théorique

▷ Création d'un OEIL1
permettant de visualiser le
maillage déformé (en
rouge) et le maillage initial
(en bleu) selon l'axe Z

Eléments 3D linéaires

```

*
* DEBUT DU FICHIER POUTRE_3D.DGIBI
*
* Calcul 3D avec des elements
* cubiques a interpolation lineaire : CUB8
TITR 'POUTRE MAILLEE 3D' ;
OPTI DIME 3 ELEM CUB8;
OPTI ECHO 0;
*****
* PARAMETRISATION, DONNEES
*****
* DIMENSIONS
* -----
L = 1.; B = 0.1; EP = 0.012 ;
* PARAMETRES DE MAILLAGE
* -----
NE = 4;
NB = (2*NE);
NL = (8*NB);
* ELASTICITE
* -----
MODYOUNG = 2.E11;
COEFPOIS = 0.;
* CHARGEMENT
* -----
PY      = -600.;
*****
* GEOMETRIE ET MAILLAGE
*****
* POINTS
* -----
P1  = 0. (-1.*B/2.) (-1.*EP/2.);
P2  = 0. (+1.*B/2.) (-1.*EP/2.);
P3  = 0. (+1.*B/2.) (+1.*EP/2.);
P4  = 0. (-1.*B/2.) (+1.*EP/2.);
P1B = L  (-1.*B/2.) (-1.*EP/2.);
P2B = L  (+1.*B/2.) (-1.*EP/2.);
P3B = L  (+1.*B/2.) (+1.*EP/2.);
P4B = L  (-1.*B/2.) (+1.*EP/2.);

```

▷ Calcul 3D avec des éléments cubiques à interpolation linéaire (8 nœuds)

▷ Création des points en 3D pour définir la surface d'encastrement et celle qui lui est opposée

```

* LIGNES
* -----
P1P2 = P1 DROI NE P2 ;
P2P3 = P2 DROI NB P3 ;
P3P4 = P3 DROI NE P4 ;
P4P1 = P4 DROI NB P1 ;
P1BP2B = P1B DROI NE P2B ;
P2BP3B = P2B DROI NB P3B ;
P3BP4B = P3B DROI NE P4B ;
P4BP1B = P4B DROI NB P1B ;
* SURFACE
* -----
SURF1 = DALL P1P2 P2P3 P3P4 P4P1 PLAN ;
SURF2 = DALL P1BP2B P2BP3B P3BP4B P4BP1B PLAN ;
* VOLUME
* -----
VOL1 = SURF1 VOLU NL SURF2 ;
*****
* MODELE ET MATERIAU
*****
MOD1 = MODE VOL1 MECANIQUE ELASTIQUE ISOTROPE;
MAT1 = MATE MOD1 YOUN MODYOUNG NU COEFPOIS;
*****
* CONDITIONS AUX LIMITES ET CHARGEMENT
*****
* CL
* --
CL1 = BLOQ DEPLA ROTA SURF1;
* CHARGEMENT
* -----
GEO1 = VOL1 POIN DROI P2B P3B 1E-3 ;
FOR1 = FORCER FY PY GEO1;
* MATRICE DE RIGIDITE
* -----
RIG1 = RIGI MOD1 MAT1;
RIG1 = RIG1 ET CL1;
*****
* RESOLUTION
*****
DEPL1 = RESO RIG1 FOR1;

```

▷ DALLage des deux
surfaces planes opposées

▷ Création du volume 3D à
partir des deux surfaces
opposées

```

*****
* POSTRAITEMENT
*****

* COMPARAISON NUMERIQUE / THEORIQUE
* -----
DY      = EXTR DEPL1 UY P2B ;
DY2     = EXTR DEPL1 UY P1B ;
DYABS   = ABS ((DY+DY2)/2.);
*
INERTIE = (EP*(B**3))/12.;
DYTHEO  = ABS ((PY*(L**3))/(3*MODYOUNG*INERTIE));
*
MESS 'fleche maximale en metre : ' DYABS;
MESS 'fleche theorique en metre : ' DYTHERO;
*
MESS ' ';
MESS '-----';
MESS '      -> NUM/THERO =' (DYABS/DYTHERO);
MESS ' ';
MESS '      ->      NBNO =' (NBNO VOL1);
MESS '-----';
*
* VISUALISATION
* -----
DEFO = DEFO VOL1 DEPL1 0. BLAN;
DEF1 = DEFO VOL1 DEPL1 BLEU;
TRAC (DEFO ET DEF1);
OEIL1=0. 0. 1000. ;
TRAC OEIL1 (DEFO ET DEF1);
SS   = SIGM MOD1 MAT1 DEPL1;
SSXX = EXCO SMXX SS;
TRAC OEIL1 DEF1 MOD1 SSXX;
*****
*FIN DU FICHIER
*****
FIN ;

```

▷ Calcul de la flèche
numérique comme
moyenne des déplacements
sup et inf selon Y

▷ Extraction puis
visualisation sur le maillage
déformé de la composante
xx de la contrainte SS

Eléments 3D quadratiques

```
* DEBUT DU FICHIER POUTRE_3D.DGIBI
* Calcul 3D avec des elements
* cubiques a interpolation quadratique : CU20
TITR 'POUTRE MAILLEE 3D' ;
OPTI DIME 3 ELEM CU20;
OPTI ECHO 0;
*****
* PARAMETRISATION, DONNEES
*****
* DIMENSIONS
* -----
L = 1.; B = 0.1; EP = 0.012 ;
* PARAMETRES DE MAILLAGE
* -----
NE = 4; NB = (2*NE); NL = (8*NB);
* ELASTICITE
* -----
MODYOUNG = 2.E11; COEFPOIS = 0.;
* CHARGEMENT
* -----
PY    = -600.;
*****
* GEOMETRIE ET MAILLAGE
*****
* POINTS
* -----
P1    = 0. (-1.*B/2.) (-1.*EP/2.);
P2    = 0. (+1.*B/2.) (-1.*EP/2.);
P3    = 0. (+1.*B/2.) (+1.*EP/2.);
P4    = 0. (-1.*B/2.) (+1.*EP/2.);
* LIGNES
* -----
P1P2 = P1 DROI NE P2 ;
P2P3 = P2 DROI NB P3 ;
P3P4 = P3 DROI NE P4 ;
P4P1 = P4 DROI NB P1 ;
* SURFACE
* -----
SURF1 = DALL P1P2 P2P3 P3P4 P4P1 PLAN ;
```

▷ Utilisation d'éléments cubiques à interpolation quadratique (20 nœuds)

▷ Idem que précédemment mais création de 4 points seulement, 4 lignes et maillage d'une seule surface

```

* VOLUME
* -----
VOL1 = SURF1 VOLU NL TRANS (L 0. 0.) ;
*****
* MODELE ET MATERIAU
*****
MOD1 = MODE VOL1 MECANIQUE ELASTIQUE ISOTROPE;
MAT1 = MATE MOD1 YOUN MODYOUNG NU COEFPOIS;
*****
* CONDITIONS AUX LIMITES ET CHARGEMENT
*****
* CL
* --
CL1 = BLOQ DEPLA ROTA SURF1;
* CHARGEMENT
* -----
P2B  = VOL1 POIN PROC (L (B/2.) (-1.*EP/2.));
P3B  = VOL1 POIN PROC (L (B/2.) (EP/2.));
GEO1 = VOL1 POIN DROI P2B P3B 1E-3 ;
FOR1 = FORCER FY PY GEO1;
* visualisation
VEC1 = VECT FOR1 1.E-3 FX FY FZ ROUG;
TRAC VEC1 VOL1 ;
* MATRICE DE RIGIDITE
* -----
RIG1 = RIGI MOD1 MAT1;
RIG1 = RIG1 ET CL1;
*****
* RESOLUTION
*****
DEPL1 = RESO RIG1 FOR1;
*****
* POSTRAITEMENT
*****
* NUMERIQUE / THEORIQUE
* -----
P1B  = VOL1 POIN PROC (L (-1.*B/2.) (-1.*EP/2.));
DY   = EXTR DEPL1 UY P2B ;
DY2  = EXTR DEPL1 UY P1B ;
DYABS = ABS ((DY+DY2)/2.);

```

▷ Création du VOLUME par
TRANSLation de la seule
surface créée

▷ Extraction des points
définissant la droite GEO1
sur laquelle doit
s'appliquer la force

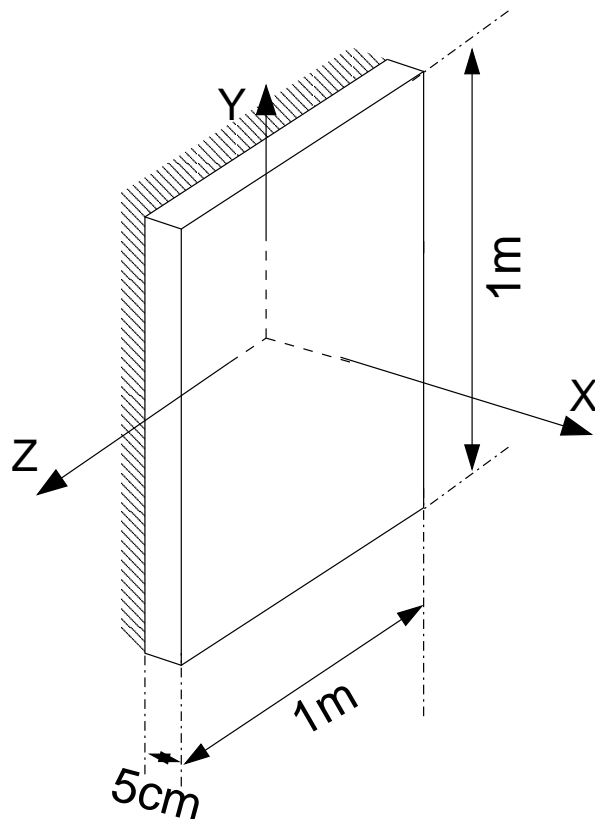
```

*
INERTIE = (EP*(B**3))/12.;
DYTHEO = ABS ((PY*(L**3))/(3*MODYOUNG*INERTIE));
*
MESS 'fleche maximale en metre : ' DYABS;
MESS 'fleche theorique en metre : ' DYTHEO;
*
MESS ' ';
MESS '-----';
MESS '    ->  NUM/THEO =' (DYABS/DYTHEO);
MESS ' ';
MESS '    ->      NBNO =' (NBNO VOL1);
MESS '-----';
*
* VISUALISATION
* -----
*
DEFO = DEFO VOL1 DEPL1 0. BLAN;
DEF1 = DEFO VOL1 DEPL1 BLEU;
TRAC (DEFO ET DEF1);
OEIL1=0. 0. 1000. ;
TRAC OEIL1 (DEFO ET DEF1);
*
SS   = SIGM MOD1 MAT1 DEPL1;
SSXX = EXCO SMXX SS;
TRAC OEIL1 DEF1 MOD1 SSXX;
*****
*FIN DU FICHIER
*****
FIN ;

```

5.4 Problème Membrane

5.4.1 Enoncé



$$\begin{cases} E = 200000 MPa \\ \nu = 0.3 \\ P = 600 N \end{cases}$$

On cherche ici à calculer avec CAST3M une membrane à base carrée encastrée sur les bords de l'une de ses surfaces et soumise à une pression uniformément répartie sur sa surface opposée.

▷ Adaptation d'un fichier

Modifier le fichier POUTRE_3D.DGIBI pour traiter le cas présent à l'aide d'éléments parallélépipédiques à interpolation linéaire pour un maillage qui vous semble convenable.

▷ Visualisation de la déformée

Visualiser la déformée selon un oeil colinéaire à l'axe Z.

▷ Visualisation de la contrainte de Von Mises

Visualiser la contrainte de Von Mises selon un oeil colinéaire à l'axe X+Y+Z et sur le maillage déformé.

▷ Commentaires mécaniques

Que peut-on dire sur les zones en compression et celles en traction ?

5.4.2 Corrigé

```

*
* DEBUT DU FICHIER MEMBRANE.DGIBI
*
* Calcul 3D avec des elements
* cubiques a interpolation quadratique : CUB8
TITR 'POUTRE MAILLEE 3D' ;
OPTI DIME 3 ELEM CUB8;
*OPTI ECHO 0;
*****
* PARAMETRISATION, DONNEES
*****
* DIMENSIONS
L = 0.05; B = 1.; EP = B ;
* PARAMETRES DE MAILLAGE
NE = 30; NB = (NE); NL = 4;
* ELASTICITE
MODYOUNG = 2.E11; COEFPOIS = 0.3;
* CHARGEMENT
PY = -600.;
*****
* GEOMETRIE ET MAILLAGE
*****
* POINTS
P1 = 0. (-1.*B/2.) (-1.*EP/2.); P1B = L (-1.*B/2.) (-1.*EP/2.);
P2 = 0. (+1.*B/2.) (-1.*EP/2.); P2B = L (+1.*B/2.) (-1.*EP/2.);
P3 = 0. (+1.*B/2.) (+1.*EP/2.); P3B = L (+1.*B/2.) (+1.*EP/2.);
P4 = 0. (-1.*B/2.) (+1.*EP/2.); P4B = L (-1.*B/2.) (+1.*EP/2.);
* LIGNES
P1P2 = P1 DROI NE P2 ; P1BP2B = P1B DROI NE P2B ;
P2P3 = P2 DROI NB P3 ; P2BP3B = P2B DROI NB P3B ;
P3P4 = P3 DROI NE P4 ; P3BP4B = P3B DROI NE P4B ;
P4P1 = P4 DROI NB P1 ; P4BP1B = P4B DROI NB P1B ;
* SURFACE
SURF1 = DALL P1P2 P2P3 P3P4 P4P1 PLAN ;
SURF2 = DALL P1BP2B P2BP3B P3BP4B P4BP1B PLAN ;
* VOLUME
VOL1 = SURF1 VOLU NL SURF2 ;
*****
* MODELE ET MATERIAU

```

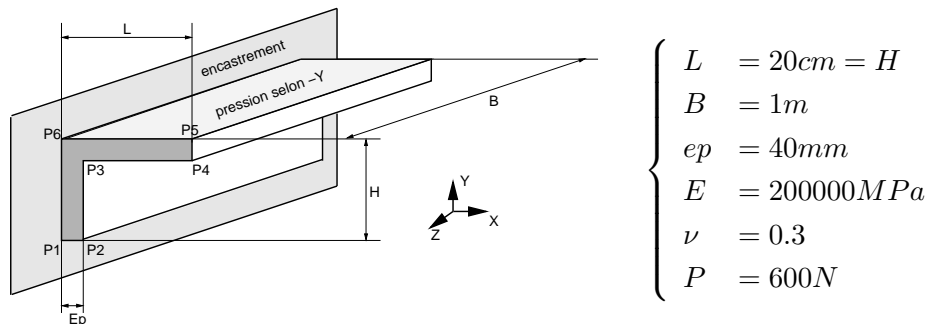
```

*****
MOD1 = MODE VOL1 MECANIQUE ELASTIQUE ISOTROPE;
MAT1 = MATE MOD1 YOUN MODYOUNG NU COEFPOIS;
*****
* CONDITIONS AUX LIMITES ET CHARGEMENT
*****
* CL
CL1 = BLOQ DEPLA ROTA (P1P2 ET P2P3 ET P3P4 ET P4P1);
* CHARGEMENT
FOR1 = FORCER FX PY SURF2;
* MATRICE DE RIGIDITE
RIG1 = RIGI MOD1 MAT1; RIG1 = RIG1 ET CL1;
*****
* RESOLUTION
*****
DEPL1 = RESO RIG1 FOR1;
*****
* POSTRAITEMENT
*****
DEFO = DEFO VOL1 DEPL1 0. BLAN;
DEF1 = DEFO VOL1 DEPL1 BLEU;
OEIL1=0. 0. 1000. ;
TRAC OEIL1 (DEFO ET DEF1);
*
SS = SIGM MOD1 MAT1 DEPL1;
SSV = VMIS MOD1 SS;
OEIL2=-1000. -1000. -1000. ;
TRAC OEIL2 DEF1 MOD1 SSV;
*****
*FIN DU FICHIER
*****
FIN ;

```

5.5 Problème Etagère

5.5.1 Enoncé



On cherche ici à déterminer numériquement à l'aide du logiciel CASTEM la solution du problème présenté sur la figure ci-dessus : étagère murale encastree soumise à une charge uniforme. Le matériau constitutif de la poutre est homogène, élastique, isotrope. La géométrie, le chargement, les notations et les données numériques sont indiqués sur la figure.

▷ **Créer** le fichier de mise en données de ce problème en utilisant les fichiers déjà constitués, afin de procéder à un calcul avec des éléments tétrahédriques à 4 nœuds. On portera attention à la qualité du maillage.

▷ **Visualiser** les contraintes de Von Mises sur le maillage déformé et commenter. Réévaluer si besoin la qualité du maillage.

▷ **Modifier** la mise en données pour traiter le même problème avec une étagère 2 fois plus longue (dans la direction Z) sans modifier le nombre de DDL de la structure. Commenter.

5.5.2 Corrigé

```

* DEBUT DU FICHIER ETAGERE_3D.DGIBI
*
* Calcul 3D avec des elements
* tétrahédriques a interpolation lineaire
TITR 'ETAGERE 3D' ;
OPTI DIME 3 ELEM TET4;
*OPTI ECHO 0;

*****
* PARAMETRISATION, DONNEES
*****
* ----- DIMENSIONS -----
L      = 0.2  ;
H      = 0.2  ;
B      = 1.   ;
EP     = 0.04 ;
* ----- PARAMETRES DE MAILLAGE -----
NE      = 2;
NH      = (18);
NL      = (NH);
NB      = (30);
* ----- ELASTICITE -----
MODYOUNG = 2.E11;
COEFPOIS = 0.3;
* ----- CHARGEMENT -----
PY      = -600.;
*****
* GEOMETRIE ET MAILLAGE
*****
* ----- POINTS -----
P1  = 0. 0. 0.;
P2  = EP 0. 0.;
P3  = EP (H-(1.*EP)) 0.;
P4  = L (H-(1.*EP)) 0.;
P5  = L (H) 0.;
P6  = 0. (H) 0.;

```



```

* ----- LIGNES -----
DENSDEB = (H-(1.*EP))/4.;
DENSFIN = (H-(1.*EP))/70.;
*
P1P2 = P1 DROI NE P2 ;
P2P3 = P2 DROI (-1*(NH)) P3 DINI DENSDEB DFIN DENSFIN ;
P3P4 = P3 DROI (-1*(NL)) P4 DINI DENSFIN DFIN DENSDEB ;
P4P5 = P4 DROI NE P5 ;
P5P6 = P5 DROI (-1*(NL)) P6 DINI DENSDEB DFIN DENSFIN ;
P6P1 = P6 DROI (-1*(NH)) P1 DINI DENSFIN DFIN DENSDEB ;
* ----- SURFACE -----
LTOT1 = P1P2 ET P2P3 ET P3P4 ET P4P5 ET P5P6 ET P6P1;
SURF1 = SURF PLAN LTOT1 ;
* ----- VOLUME -----
VOL1 = SURF1 VOLU NB TRANS (0. 0. B) ;
*****
* MODELE ET MATERIAU
*****
MOD1 = MODE VOL1 MECANIQUE ELASTIQUE ISOTROPE;
MAT1 = MATE MOD1 YOUN MODYOUNG NU COEFPOIS;
*****
* CONDITIONS AUX LIMITES ET CHARGEMENT
*****
* ----- CL -----
*
* EXTRACTION D'UN POINT POUR DEFINIR AVEC
* P6 ET P1 UN PLAN QUI CONTIENT
* LA SURFACE SOUHAITEE NOMMEE GEO1
P6Z = VOL1 POIN PROC (0. H B);;
GEO1 = VOL1 POIN PLAN P6 P1 P6Z 1E-4 COUL VERT ;
CL1 = BLOQ DEPLA ROTA GEO1;
*
* POUR SYMETRISER LA STRUCTURE SELON Z
* DECOMMENTER LES 2 LIGNES CI-DESSOUS !!!!!!!
*CL2= BLOQ UZ SURF1;
*CL1=CL1 ET CL2;
*
* ----- CHARGEMENT -----
*
* CREATION D'UN POINT PAR TRANSLATION DE P6

```

▷ Utilisation de densités plus élevées aux points P3 et P6 pour assurer un maillage raffiné dans la zone la plus sollicitée

▷ Création du volume VOL1 par translation selon Z de la surface SURF1

▷ Pour effectuer un calcul sur une structure 2 fois plus longue sans augmenter le nombre de DDL, on peut appliquer des condition de symétrie (déplacement boqué selon Z) à la surface SURF1

```

* POUR DEFINIR AVEC P6 ET P6Z UN PLAN QUI CONTIENT
* LA SURFACE SOUHAITEE NOMMEE GEO2
P6X = P6 PLUS (((3.)*L) 0. 0.);
GEO2 = VOL1 POIN PLAN P6 P6Z P6X 1E-4 COUL ROUGE ;
FOR1 = FORCER FY PY GEO2;
*
* ----- VISU SURFACE EXTRUDEE ET ZONES CL/CHARGEMENT -----
TRAC (100. -100. 100.) (SURF1 ET GEO1 ET GEO2 );
* ----- MATRICE DE RIGIDITE -----
RIG1 = RIGI MOD1 MAT1;
RIG1 = RIG1 ET CL1;
*****
* RESOLUTION
*****
DEPL1 = RESO RIG1 FOR1;
*****
* POSTRAITEMENT
*****
DEFO = DEFO VOL1 DEPL1 0. BLAN;
DEF1 = DEFO VOL1 DEPL1 10E+4 BLEU;
TRAC (2. -10. 1000.) (DEFO ET DEF1) CACH;
OEIL1=0. 0. 1000. ;
TRAC OEIL1 (DEFO ET DEF1);
*
SS   = SIGM MOD1 MAT1 DEPL1;
SSVV = VMIS MOD1 SS;
TRAC OEIL1 DEF1 MOD1 SSVV;
*
*****
*FIN DU FICHIER
*****
FIN ;

```