# Appli Poutre - Notes de développement

Utilisateur GitHub : `lacorrep`

Juin 2017

# Table des matières

# Chapitre 1

# Modélisation physique

## 1.1 Hypothèses

— Poutre isostatique (vérifié avant chaque résolution)
— Petites déformations (la déformation n'induit pas de nouveaux efforts internes)
— Sections droites (indéformables)
— Poutres d'Euler-Bernoulli : les sections restent perpendiculaires à la ligne moyenne

L'hypothèse d'Euler-Bernoulli implique que la rotation de la section est égale à la pente de la déformée (en petites transformations).

## 1.2 Équations du problème

Les équations à résoudre sont :

$$\begin{cases} \dfrac{\mathrm{d}u}{\mathrm{d}x} = \dfrac{1}{ES}N(x) \\[2mm] \dfrac{\mathrm{d}^2 v}{\mathrm{d}x^2} = \dfrac{1}{EI}M(x) \end{cases}$$

On doit avoir 1 CL sur $u$ et 2 CL sur $v$.

Les deux problèmes sont clairement découplés.

### 1.2.1 Calcul des efforts internes $N$ et $M$

Les efforts sont calculés par coupure en isolant la partie droite.

### 1.2.2 Calcul des déplacements horizontaux $u$

On intègre la première équation entre $0$ et $x$ :

$$u(x) = u(0) + \int_0^x \frac{1}{ES} N(x)\,\mathrm{d}x$$

Cependant $u(0)$ n'est pas forcément connu.

On pose

$$U(x) = \int_0^x \frac{1}{ES} N(x)\,\mathrm{d}x$$

ce qui est la solution lorsque $u(0) = 0$.

Via les données saisies par l'utilisateur, on connaît la position $x_0$ de la liaison qui empêche le déplacement horizontal. On impose alors de manière forte $u(x_0) = 0$ :

$$\boxed{u(x) = U(x) - U(x_0)}$$

### 1.2.3 Calcul des déplacements verticaux $v$

à faire

Via les données saisies par l'utilisateur, on connaît les positions $x_i$ de la ou des liaisons qui empêchent le déplacement vertical. On impose alors cette condition de manière forte.

Si on a une seule liaison en $x_1$ : la solution est définie à une constante près

$$\boxed{v(x) = V(x) - V(x_1)}$$

Si on a deux liaisons en $x_1$ et $x_2$ : la solution est définie à un polynôme près. On pose

$$\alpha = \frac{V(x_2) - V(x_1)}{x_2 - x_1}$$

$$\boxed{v(x) = V(x) - \Big(\alpha \times (x - x_1) + V(x_1)\Big)}$$

# Chapitre 2

# Implémentation

## 2.1 Affichage de la poutre

parler de la marge sur les côtés pour afficher la déformée

## 2.2 Discrétisation des équations d'équilibre local

On prend autant de points que la poutre fait de pixels à l'écran.

## 2.3 Détermination des efforts d'appui

On note $\vec{R}$, $M$ les réactions des liaisons et $\vec{F}$, $C$ les chargements. Le PFS donne :

$$\begin{cases} \vec{R}+\vec{F} = & \vec{0} \\ M + C = & 0 \end{cases}$$

$$\begin{cases} R_X = & -F_X \\ R_Y = & -F_Y \\ M = & -C \end{cases}$$

L'unique CL sur $u$ est introduite par un unique effort horizontal $R_X$. Les deux CL sur $v$ et ses dérivées peuvent être appliquées par un effort vertical $R_Y$ et par un autre effort vertical $R'_Y$ ou un moment $M$.

$$\Rightarrow \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} R_X \\ R_Y \\ R'_Y \text{ ou } M \end{bmatrix} = \begin{bmatrix} -\sum F_X \\ -\sum F_Y \\ -\sum C \end{bmatrix}$$

Les efforts horizontaux ne génèrent pas de moments sur la poutre. Ainsi l'équilibre horizontal ne dépend que des efforts horizontaux. Il n'y a qu'une seule force de réaction horizontale.

$$\boxed{R_X = -\sum F_X}$$

Pour déterminer les autres efforts, on résout le système suivant par la méthode de Cramer :

$$\boxed{\begin{bmatrix} \dots & \dots \\ \dots & \dots \end{bmatrix} \begin{bmatrix} R_Y \\ R'_Y \text{ ou } M \end{bmatrix} = \begin{bmatrix} -\sum F_Y \\ -\sum C \end{bmatrix}}$$

Sur la première ligne, $j$-ième colonne : on a $1$ si la $j$-ième inconnue est une force, $0$ si c'est un moment.

Sur la deuxième ligne, $j$-ième colonne : on a $x$ la distance de la force à l'extrémité gauche de la poutre si la $j$-ième inconnue est une force, $1$ si c'est un moment.

**Exemples** pour une poutre bi-appuyée (en $x$ et $x'$) :

$$\begin{bmatrix} 1 & 1 \\ x & x' \end{bmatrix} \begin{bmatrix} R_Y \\ R'_Y \end{bmatrix}$$

pour une poutre encastrée (en $x$) :

$$\begin{bmatrix} 1 & 0 \\ x & 1 \end{bmatrix} \begin{bmatrix} R_Y \\ M \end{bmatrix}$$

**Résolution**

$$\boxed{[A]\{R\} = \{F\} \Leftrightarrow \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} R1 \\ R2 \end{bmatrix} = \begin{bmatrix} -\sum F_Y \\ -\sum C \end{bmatrix}}$$

$$R1 = \frac{a_{01} \times \sum C - a_{11} \times \sum F_Y}{\det A}$$
$$R2 = \frac{a_{10} \times \sum F_Y - a_{00} \times \sum C}{\det A}$$

# Chapitre 3

# Bibliothèques et dépendances

| Nom | Version | Utilité | Source |
|---|---|---|---|
| jQuery | 3.2.1 | Permet de manipuler facilement le DOM. | `jquery.com` |
| jQuery UI | 1.12.1 | Permet de définir rapidement des composants d'interface (*drag and drop*, redimensionnement). | `jqueryui.com` |
| Touch Punch | 0.2.3 | Assure la compatibilité de jQuery UI avec les événements tactiles. | `touchpunch.furf.com` |
| Bootstrap | 3.3.7 | Framework d'interface et responsive design. | `getbootstrap.com` |