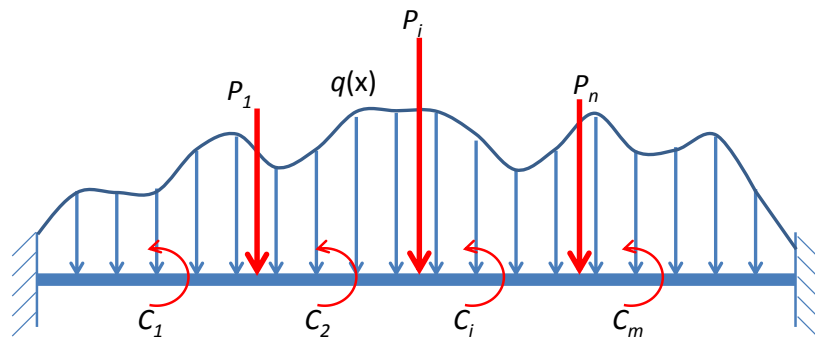


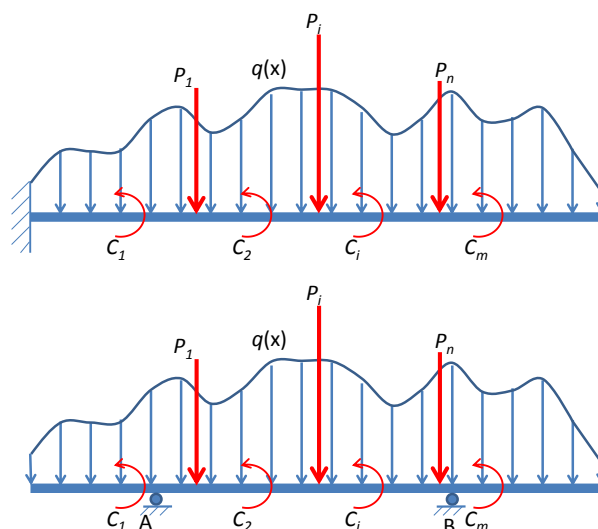
Projet 1 Equation différentiels :

Le programme doit résoudre l'équation différentielle de la déformée en vue d'effectuer une étude d'une poutre. L'utilisateur doit entrer les dimensions de la poutre, son chargement ainsi que ses conditions aux limites. En sortie le programme doit fournir, l'équation de la déformée, les moments fléchissant, les efforts tranchants, ainsi que les contraintes normales.



- Entrez les dimensions de la poutre (longueur et section)
- entrez l'équation du chargement réparti (constant ou trapézoïdal), les charges localisées (forces ou moments) et leurs emplacements.
- discrétisez la poutre en n nœuds.
- transformez l'équation différentielle de la déformée en un système d'équations linéaire.
- résolvez ce système pour avoir la déformée, dérivez deux fois la déformée pour avoir le moment fléchissant, puis une troisième fois pour l'effort tranchant.

Projet 2 : intégrales :



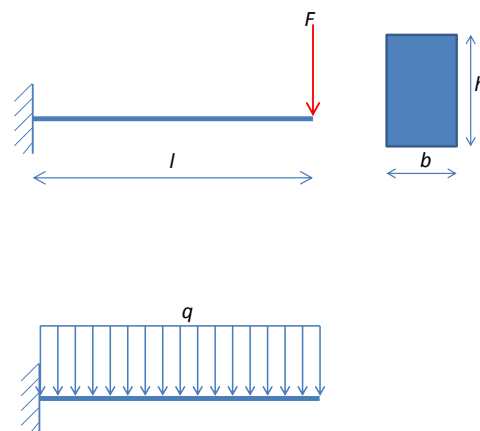
Le programme doit calculer la flèche en un point particulier d'une poutre isostatique en utilisant le théorème des travaux virtuels. . L'utilisateur doit entrer les dimensions de la poutre, son chargement

ainsi que ses conditions aux limites. De plus, il doit préciser l'abscisse du point à laquelle nous désirons calculer la flèche. En sortie le programme doit fournir la flèche.

- Entrez le module d'Young ainsi que le moment d'inertie de la poutre.
- Entrez les abscisses des extrémités de la poutre, et ses conditions aux limites. (s'il y a un encastrement il vaut mieux toujours lui donner l'abscisse 0)
- Entrez l'équation de la charge répartie puis les charges concentrées.
- Calculez le moment fléchissant de la poutre réelle et la poutre virtuelle en chaque point.
- Calculez la flèche au point demandé.

Projet 3 : Optimisation

Le programme doit fournir la section minimale de la poutre isostatique sous contrainte de ne pas dépasser la contrainte limite du matériau ni une flèche limite admissible.



- Entrez le type de la poutre.
- Utilisez les formules de la RDM pour localiser la section critique et la flèche maximale.
- La fonction objectif est formé de la somme de la surface de la section et d'une pénalité due à la violation de la contrainte limite et de la flèche limite.

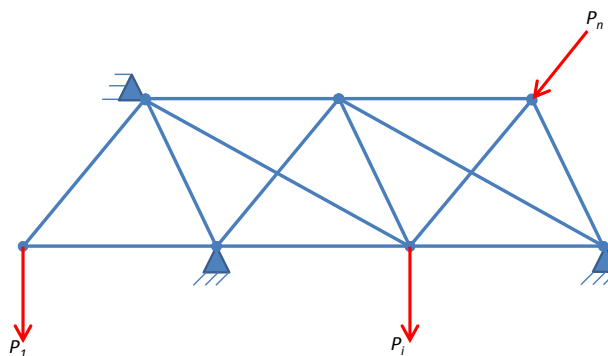
$$f = S + \lambda_1 \cdot \text{abs}(\sigma_{\max}) + \lambda_2 \cdot \text{abs}(v_{\max})$$

$$\lambda_1 = \begin{cases} 0 & \text{pour } \sigma_{\max} < \sigma_{ad} \\ 10 & \text{pour } \sigma_{\max} \geq \sigma_{ad} \end{cases}$$

$$\lambda_2 = \begin{cases} 0 & \text{pour } |v_{\max}| < v_{ad} \\ 10 & \text{pour } |v_{\max}| \geq v_{ad} \end{cases}$$

- Cherchez b et h qui minimisent f .

Projet 4 : optimisation :



Dans ce programme, nous allons utiliser le théorème de l'énergie potentielle totale pour effectuer le calcul des efforts dans un treillis. Le treillis peut être isostatique ou hyperstatique.

- a) Entrez la section des barres utilisées.
- b) Entrez les coordonnées des nœuds.
- c) Précisez les nœuds bloqués et les directions de blocage.
- d) Précisez les charges.
- e) L'énergie potentielle totale est l'énergie élastique stockée dans les barres déformées moins le travail des forces extérieures. Créez cette fonction en fonction des coordonnées des nœuds avant et après déformation.
- f) Minimisez l'énergie potentielle totale pour trouver la déformée du treillis puis extraire les efforts dans les barres.

Méthode du nombre d'or pour la minimisation unidimensionnelle :

```

a0=input('a0= ');          %valeur initiale estimée
G=((5^0.5)+1)/2;          %facteur d'expansion supérieur a 1 nous utilisons le
grand nombre d'or 1.618
P=potentielle(a0);        %énergie potentielle totale en a0
PP=(potentielle(a0+0.01)-potentielle(a0-0.01))/0.02; %dérivée en a0
de=-0.1*abs(P)/PP;        %pas initial
a1=a0+de;                 %deuxième point particulier
a2=a1+G*(a1-a0);          %troisième point particulier
a3=a2+G*(a2-a1);          %quatrième point particulier
while potentielle(a3)<potentielle(a2) %détermination d'un intervalle
initial
    a0=a1;
    a1=a2;
    a2=a3;
    a3=a2+G*(a2-a1);
end
G=G-1;                    %petit nombre d'or pour la réduction de l'intervalle
while (a3-a0)>eps
    if potentielle(a2)>potentielle(a1)
        a3=a2;
    else a0=a1;
    end
    a1=G*a0+(1-G)*a3;
    a2=(1-G)*a0+G*a3;
end

```

Le passage d'un problème poly-dimensionnel à un problème unidimensionnel se fait suivant la méthode du gradient conjugué expliquée en classe.

Bon Courage