Portique à 3 niveaux,
libre, sans amortissement

On étudie un bâtiment qui est assimilé ici comme un portique à 3 niveau. En partant de la base vers le haut on donne les caractéristiques suivantes : M1= 2000kg, M2=1500kg et M3=1000kg. La rigidité de chaque niveau est respectivement K1=4.5x10^6 N/m K2=3.0 x 10^6 N/m et K3=1.5 x10^6 N/m. La hauteur de chaque niveau et L1=3m
1. Trouver les équation de mouvement du portique, équivalent à un système à 3DDL si au moment initial le système est sorti de l'équilibre (déplacements initiaux   uinit={0.01, 0.03, 0.01} et ensuite lâché sans vitesse vinit={0,0, 0}

Pour cela
a - Faire une analyse modale
b - reformuler et résoudre le problème dans l'espace modale
c - revenir dans l'espace réelle et tracer la solution

```
(*conditions initiales  : on deplace de leur position de l'équilibre les DDL 1,
2 et 3 par u1ini, u2init et u3init chacun avec une vitesse  v1in, v2in , v3in *)
uinit = {0.01, 0.03, 0.01};
vinit = {0, 0, 0};
```

```
ValNumeriques = {M1 → 1000, M2 → 1500, L1 → 3, b1 → 0.20, h1 → 0.2, L2 → 3.5,
  b2 → 0.2, h2 → 0.2, K1 → 2 * 12 * (b1 * h1^3 / 12) * (360 * 10^6) / (L1^3),
  K2 → 2 * 12 * (b2 * h2^3 / 12) * (360 * 10^6) / (L2^3)}
```

```
In[•]:= matK = K3 * {{5, -2, 0}, {-2, 3, -1}, {0, -1, 1}}
Out[•]= {{5 K3, -2 K3, 0}, {-2 K3, 3 K3, -K3}, {0, -K3, K3}}
```

```
In[•]:= MatrixForm[FullSimplify[Inverse[matK]]]
```
apparence m··· simplifie complè··· matrice inverse

Out[•]//MatrixForm=

$$\begin{pmatrix} \frac{1}{3\,K3} & \frac{1}{3\,K3} & \frac{1}{3\,K3} \\ \frac{1}{3\,K3} & \frac{5}{6\,K3} & \frac{5}{6\,K3} \\ \frac{1}{3\,K3} & \frac{5}{6\,K3} & \frac{11}{6\,K3} \end{pmatrix}$$

```
In[•]:= matM = M3 * {{2, 0, 0}, {0, 1.5, 0}, {0, 0, 1}}
Out[•]= {{2 M3, 0, 0}, {0, 1.5 M3, 0}, {0, 0, M3}}
```

In[ ]:= (*Matrice dynamique D *)MatrixForm[MatD = FullSimplify[Inverse[matK].matM]]
dér·· apparence matricielle simplifie complè·· matrice inverse

Out[ ]//MatrixForm=

$$\begin{pmatrix} \frac{2\,M3}{3\,K3} & \frac{0.5\,M3}{K3} & \frac{M3}{3\,K3} \\ \frac{2\,M3}{3\,K3} & \frac{1.25\,M3}{K3} & \frac{5\,M3}{6\,K3} \\ \frac{2\,M3}{3\,K3} & \frac{1.25\,M3}{K3} & \frac{11\,M3}{6\,K3} \end{pmatrix}$$

In[ ]:= (* Les valeurs propres de la matrice dynamiques D λ sont égale à 1/(ω^2) *)
dérivée d

Eigenvalues[MatD]
valeurs propres

Out[ ]= $\left\{\frac{2.84524\,M3}{K3}, \frac{0.622433\,M3}{K3}, \frac{0.282331\,M3}{K3}\right\}$

In[ ]:= {ω1, ω2, ω3} = 1 / Sqrt[Eigenvalues[MatD]] //. {M3 → 1000, K3 → 1.5 * 10^6}
raci··· valeurs propres

Out[ ]= {22.9608, 49.0907, 72.8897}

In[ ]:= (* les fréquences sont donc *)
{f1, f2, f3} = (1 / (2 * π)) * {ω1, ω2, ω3}

Out[ ]= {3.65432, 7.81303, 11.6008}

(* La fréquence fondamentale est la plus petite, ici fn=f1=3.65432 Hz *)

In[ ]:= (* Traitement des valeurs propres *)
MatrixForm[matv = -1 * Eigenvectors[MatD]]
apparence matricielle vecteurs propres

Out[ ]//MatrixForm=

$$\begin{pmatrix} 0.245503 & 0.527472 & 0.813328 \\ 0.502056 & 0.448537 & -0.739429 \\ 0.666127 & -0.694062 & 0.273045 \end{pmatrix}$$

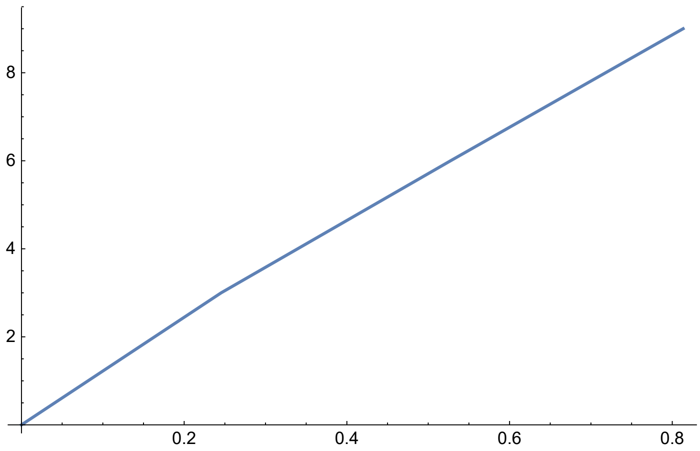
(* Nota : le logiciel de calcul utilisé ici
donne les vecteurs propres normalisés et rangés en lignes,
De ce fait la matrice "matv" ici, renvoyé par le logiciel,
ne coincide pas avec la matrice de passage [P],
mais à sa transposée [P]'. AfPour obtenir la matrice de passage
(vecteurs rangés en colonnes) on va definir la matrice de passage comme
le transpose de "matv". On profitera en passage pour maultiplier par -
active les messages
1 pour obtenir un vecteur 1 plutôt positif : ça ne change rien tout
vecteur propre multiplié par un scalair est aussi un vecteur propre *)
matP = -1 * Transpose[Eigenvectors[MatD]]
transposée vecteurs propres

Out[ ]= {{0.245503, 0.502056, 0.666127},
{0.527472, 0.448537, -0.694062}, {0.813328, -0.739429, 0.273045}}

In[ ]:=
```
(*Traitement des modes  : représentation graphique des modes*)
haut = {0, L1, 2 * L1, 3 * L1} //. {L1 → 3};
vectTrace1 = {0, matv[[1, 1]], matv[[1, 2]], matv[[1, 3]]};
vectTrace2 = {0, matv[[2, 1]], matv[[2, 2]], matv[[2, 3]]};
vectTrace3 = {0, matv[[3, 1]], matv[[3, 2]], matv[[3, 3]]};
```

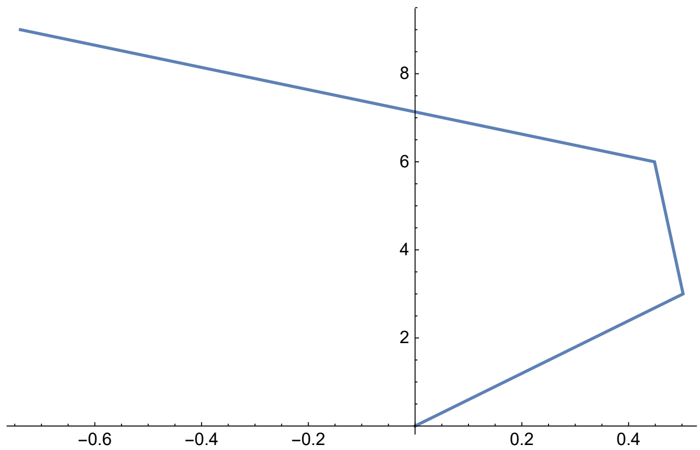
In[ ]:=
```
ListPlot[Transpose[{vectTrace1, haut}], Joined → True]
```
tracé de li··· transposée joint vrai

Out[ ]=

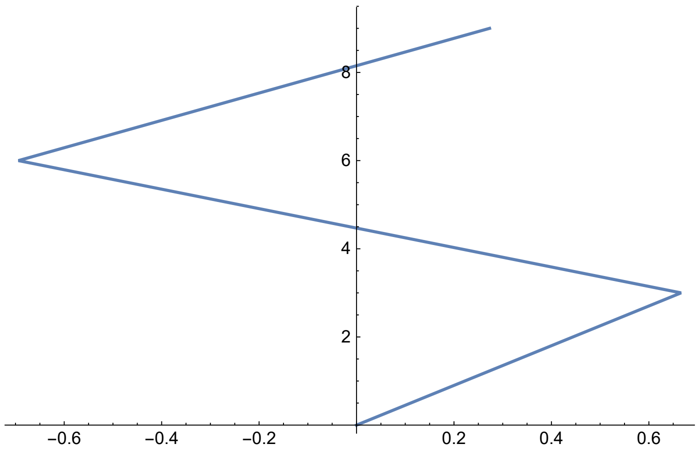


In[ ]:=
```
ListPlot[Transpose[{vectTrace2, haut}], Joined → True]
```
tracé de li··· transposée joint vrai

Out[ ]=



In[ ]:=
```
ListPlot[Transpose[{vectTrace3, haut}], Joined → True]
```
tracé de li··· transposée joint vrai

Out[ ]=



```
(* LE vecteur propre sont orthogonales par rapport
 à [K] et [M] Constatons la diagonalisation de [K] et de [M] *)
```

In[ ]:=
```
MatrixForm[
 MatKEtoile = Chop[(Transpose[matP].matK.matP //. {M3 → 1000, K3 → 1.5 * 10^6}), 10^-8]]
```
apparence matricielle rempl··· transposée

Out[ ]//MatrixForm=

$$\begin{pmatrix} 632312. & 0 & 0 \\ 0 & 3.25976\times10^{6} & 0 \\ 0 & 0 & 8.95004\times10^{6} \end{pmatrix}$$

In[ ]:=
```
MatrixForm[
 MatMEtoile = Chop[(Transpose[matP].matM.matP //. {M3 → 1000, K3 → 1.5 * 10^6}), 10^-8]]
```
apparence matricielle rempl··· transposée

Out[ ]//MatrixForm=

$$\begin{pmatrix} 1199.38 & 0 & 0 \\ 0 & 1352.65 & 0 \\ 0 & 0 & 1684.59 \end{pmatrix}$$

In[ ]:=
```
(* On peut vérifier que  chaque ωi correspond
    active les messages
 bien à celui d'un SDOF avec mase Mi_etoile et Ki_etoile *)
Sqrt[MatKEtoile[[1, 1]] / MatMEtoile[[1, 1]]]
racine carrée
Sqrt[MatKEtoile[[2, 2]] / MatMEtoile[[2, 2]]]
racine carrée
Sqrt[MatKEtoile[[3, 3]] / MatMEtoile[[3, 3]]]
racine carrée
```

Out[ ]= 22.9608

Out[ ]= 49.0907

Out[ ]= 72.8897

```
(* Le problème sera maintenant resolu dans l'espace
 transformée : on a trois equations séparées  chacune à un degrée
   de liberté. Pour qu'on puisse résoudre ces équations on doit
   exprimer aussi les conditions initiales dans la nouvelle base *)
```

In[ ]:=
```
(* Conditions initiales *)
uinit = {0.01, 0.03, 0.0};
vinit = {0, 0, 0};
```

In[ ]:=
```
vinitEtoile = Transpose[matP].vinit
              transposée
```

Out[ ]= {0., 0., 0.}

In[ ]:=
```
uinitEtoile = Transpose[matP].uinit
              transposée
```

Out[ ]= {0.0182792, 0.0184767, -0.0141606}

```
(* les equations s'écrivent dans l'espace des variables étoilées
   M11*ü1_etoile +K1*u1_etoile =
 0 avec u1initetoile = 0.026412456912326486` et v1initEtoile =0
    Ainsi de suite pour les deux autres équations. Or ces équation,
                                                  ou
dans l'espace transformé sont chacune l'équation d'un
 osciallteur à 1 DDL. La forme gérale de la solution de ces
 équations est (voir les slides correspondants #29 du cours )
```

$$\boxed{u(t) = \frac{V_0}{\omega_n}\sin\omega_n t + u_0\cos\omega_n t}$$

```
*)
```

In[•]:=
```
u1etoile = uinitEtoile[[1]] * Cos[ω1 * t]
                               cosinus
u2etoile = uinitEtoile[[2]] * Cos[ω2 * t]
                               cosinus
u3etoile = uinitEtoile[[3]] * Cos[ω3 * t]
                               cosinus
```

Out[•]= 0.0182792 Cos[22.9608 t]

Out[•]= 0.0184767 Cos[49.0907 t]

Out[•]= -0.0141606 Cos[72.8897 t]

In[•]:=
```
(* La soluton dans l'espace des varaibles transformées est donc  *)
usoletoile = {u1etoile, u2etoile, u3etoile}
```

Out[•]= {0.0182792 Cos[22.9608 t], 0.0184767 Cos[49.0907 t], -0.0141606 Cos[72.8897 t]}

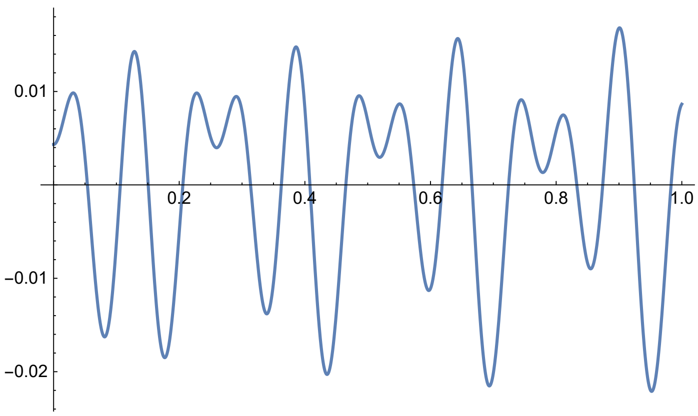
```
(*Pour revenir dans l'espace réelle il faut
 faire la transformation inverse des cordonnées *)
```

In[•]:=
```
usol = matP.usoletoile
```

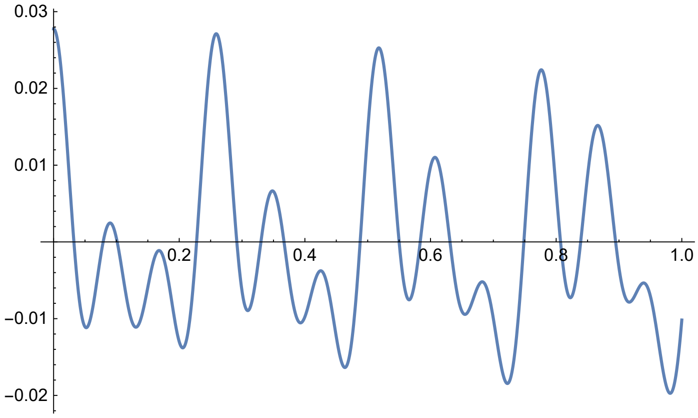
Out[•]= {0.00448759 Cos[22.9608 t] + 0.00927631 Cos[49.0907 t] - 0.00943274 Cos[72.8897 t],
0.00964175 Cos[22.9608 t] + 0.00828746 Cos[49.0907 t] + 0.00982832 Cos[72.8897 t],
0.014867 Cos[22.9608 t] - 0.0136622 Cos[49.0907 t] - 0.00386647 Cos[72.8897 t]}

In[•]:=
```
Plot[usol[[1]], {t, 0, 1}]
tracé de courbes
```

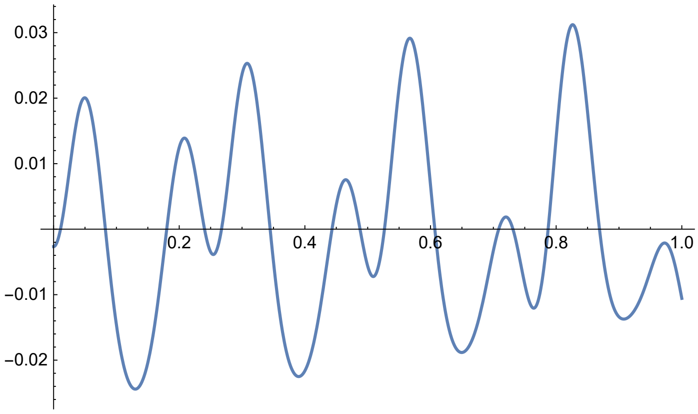
Out[•]=



In[ ]:=
```
Plot[usol[[2]], {t, 0, 1}]
```
tracé de courbes

Out[ ]=


In[ ]:=
```
Plot[usol[[3]], {t, 0, 1}]
```
tracé de courbes

Out[ ]=


```
(* Nota : Chacun des déplacements peut être écrit
  comme une somme des vecteurs propres miltiplies par un facteur β
  Par exemple:  *)
```