CONCOURS BLANC : INFORMATIQUE

AUTOUR DE DONNÉES MÉTÉOROLOGIQUES

1 Mise en situation

La structure peut être modélisée par n éléments de masse m_i (i variant de 1 à n) reliés par des liaisons visco-élastiques eux-mêmes modélisés par des un ressort de raideur k en parallèle d'un élément d'amortissement c. La structure est supposée unidimensionnelle de longueur L. Le nombre d'éléments peut être de l'ordre de plusieurs milliers.

Le déplacement au cours du temps de l'élément i autour de sa position d'équlibre est noté $u_i(t)$. Une force $f_n(t)$ est appliquée sur l'élément n uniquement. L'extrémité gauche de la structure est bloquée. Les effets de la pesanteur sont négligés.

Le théorème de la résultante dynamique appliqué à un élément i (i variant de 2 à $n-1$ inclus) s'écrit sous la forme :

$$m\frac{\mathrm{d}^2 u_i(t)}{\mathrm{d}t^2} = -2k u_i(t) + k u_{i-1}(t) + k u_{i+1}(t) - 2c\frac{\mathrm{d}u_i(t)}{\mathrm{d}t} + c\frac{\mathrm{d}u_{i-1}(t)}{\mathrm{d}t} + c\frac{\mathrm{d}u_{i+1}(t)}{\mathrm{d}t} \quad (1)$$

Le théorème de la résultante dynamique appliqué aux éléments 1 et n donne les équations suivantes :

$$m\frac{\mathrm{d}^2 u_1(t)}{\mathrm{d}t^2} = -2k u_1(t) + k u_2(t) - 2c\frac{\mathrm{d}u_1(t)}{\mathrm{d}t} - c\frac{\mathrm{d}u_2(t)}{\mathrm{d}t} \quad (2)$$

$$m\frac{\mathrm{d}^2 u_n(t)}{\mathrm{d}t^2} = -k\left(u_n(t) - u_{n-1}(t)\right) - c\left(\frac{\mathrm{d}u_n(t)}{\mathrm{d}t} - \frac{\mathrm{d}u_{n-1}(t)}{\mathrm{d}t}\right) + f_n(t) \quad (3)$$

Dans toute la suite, on imposera $f_n(t) = f_{max}\sin(\omega t)$.

2 Résolution d'une équation différentielle

On s'intéresse tout d'abord à un seul système masse ressort. L'application du théorème de la résultante dynamique en projection sur l'axe de déplacement s'écrit donc de la manière suivante :

$$m\cdot\ddot{u}(t) + c\cdot\dot{u}(t) + k\cdot u(t) = f(t)$$

On précise que pour $t \leq 0$, $u(t)=0$, $\dot{u}(t)=0$ et $\ddot{u}(t)=0$.

On pose $\left\{\begin{array}{l} x(t)=u(t) \\ v(t)=\dot{x}(t) \end{array}\right.$.

Question 1 *Réécrire l'équation différentielle sous forme d'un système d'équation en fonction de $x(t)$, $v(t)$, $\dot{x}(t)$ et $\dot{v}(t)$*

Corrigé

On a donc :

$$\left\{\begin{array}{l} v(t)=\dot{x}(t) \\ m\cdot\dot{v}(t) + c\cdot v(t) + k\cdot x(t) = f(t) \end{array}\right.$$

Question 2 *En utilisant un schéma d'Euler explicite et l'équation $v(t) = \dot{x}(t)$ exprimer la suite x_{k+1} en fonction de x_k, v_k et le pas de calcul noté noté h.*

Corrigé

On a $\dfrac{dx(t)}{dt} \simeq \dfrac{x_{k+1} - x_k}{h}$. On a donc $v_k = \dfrac{x_{k+1} - x_k}{h} \Longleftrightarrow x_{k+1} = h \cdot v_k + x_k$.

Question 3 *En utilisant un schéma d'Euler implicite et l'équation $v(t) = \dot{x}(t)$ exprimer la suite x_{k+1} en fonction de x_k, v_k et le pas de calcul noté noté h.*

Corrigé

On a $\dfrac{dx(t)}{dt} \simeq \dfrac{x_k - x_{k-1}}{h}$. On a donc $v_k = \dfrac{x_k - x_{k-1}}{h} \Longleftrightarrow x_k = h \cdot v_k + x_{k-1}$.

Question 4 *En utilisant un schéma d'Euler explicite exprimer alors la suite v_{k+1}.*

Corrigé

On a $\dfrac{dv(t)}{dt} \simeq \dfrac{v_{k+1} - v_k}{h}$. En conséquences, $m \cdot \dot{v}(t) + c \cdot v(t) + k \cdot x(t) = f(t)$ peut donc se mettre sous la forme suivante :

$$m \cdot \frac{v_{k+1} - v_k}{h} + c \cdot v_k + k \cdot x_k = f_k \Longleftrightarrow m \cdot v_{k+1} = f_k h - kh \cdot x_k - ch \cdot v_k + v_k \Longleftrightarrow v_{k+1} = \frac{h}{m} f_k - \frac{kh}{m} x_k + \frac{1 - ch}{m} v_k$$

Question 5 *En déduire que la solution de l'équation différentielle peut se mettre sous la forme :*

$$\begin{cases} x_{k+1} = \alpha \cdot v_k + \beta \cdot x_k \\ v_{k+1} = \gamma \cdot f_k + \delta \cdot x_k + \lambda \cdot v_k \end{cases} \quad \text{où on explicitera } \alpha, \beta, \gamma, \delta \text{ et } \lambda.$$

Corrigé

Dans ses conditions, on a :

$$\alpha = h \quad \beta = 1 \quad \gamma = \frac{h}{m} \quad \delta = -\frac{kh}{m} \quad \lambda = \frac{1 - ch}{m}$$

Question 6 *On note Tsimu le temps de simulation et h le pas de temps. Implémenter en Python le fonction f_omega permettant de créer une liste contenant l'ensemble des valeurs prises par la fonction f_n. On utilisera une boucle while. Les spécifications de la fonction sont les suivantes :*

```python
def f_omega(Tsimu,h,fmax,fsign):
    """
    Entrées :
        * Tsimu (flt) : temps de la simulation en seconde
        * h (flt) : pas de temps de a simulation
        * fmax (flt) : amplitude du signal (en Newton)
        * fsign (flt) : fréquence du signal (en Hertz)
    Sortie :
```

```python
    * F ( list ) :  liste  des  valeurs  de  la  fonction
        f_n (t)= fmax sin (omega *t)
"""
```

Corrigé

```python
def f_omega(Tsimu,h,fmax,fsign):
    """
    Entrées  :
        * Tsimu ( flt )  :  temps de la  simulation  en seconde
        * h ( flt )  :  pas de temps de a  simulation
        * fmax ( flt )  :  amplitude du  signal  (en  Newton)
        *  fsign  ( flt )  :  fr équence du  signal  (en  Hertz)
    Sortie  :
        * F ( list )  :  liste  des  valeurs  de  la  fonction
            f_n (t)= fmax sin (omega *t)
    """
    omega  = 2*math.pi*fsign
    t=0
    F = []
    while  t<Tsimu :
        F.append(fmax*math.sin(omega*t))
        t=t+h
    return  F
```

Question 7 *En utilisant une boucle while, générer, en Python, les listes X et V contenant les termes des suites x_k et v_k ainsi que la liste T contenant le temps. On exprimera les lignes de codes en fonction de α, β, γ, δ et λ.*

Corrigé

```python
t  = 0
T,X,V = [t],[0],[0]
F = f_omega(Tsimu,h,fmax,fsign)
i=0
while  t<Tsimu:
    t  = t+h
    T.append(t)
    X.append(alpha*V[i]+beta*X[i])
    V.append(gamma*F[i]+delta*X[i]+lambd*V[i])
    i=i+1
```

3 Résolution du problème général

Le système d'équations différentielles défini dans la première partie peut s'écrire sous forme matricielle :

$$M\ddot{X}(t)+C\dot{X}(t)+KX(t)=F(t) \quad \text{avec} \quad X(t)=\begin{bmatrix} u_1(t) \\ \dots \\ u_n(t) \end{bmatrix}$$

M, C et K des matrices carrées de taille $n \times n$.

Question 8 *En reprenant les équations (1), (2) et (3) déterminer les matrices M, C, K et F.*

Corrigé

$$M = \begin{pmatrix} m & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & m \end{pmatrix} \quad C = \begin{pmatrix} 2c & -c & 0 & \cdots & \cdots & \cdots & 0 \\ -c & 2c & -c & \ddots & & 0 & \vdots \\ 0 & -c & 2c & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 2c & -c & 0 \\ \vdots & 0 & & \ddots & -c & 2c & -c \\ 0 & \cdots & \cdots & \cdots & 0 & -c & 2c \end{pmatrix} \quad K = \begin{pmatrix} 2k & -k & 0 & \cdots & \cdots & \cdots & 0 \\ -k & 2k & -k & \ddots & & 0 & \vdots \\ 0 & -k & 2k & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 2k & -k & 0 \\ \vdots & 0 & & \ddots & -k & 2k & -k \\ 0 & \cdots & \cdots & \cdots & 0 & -k & 2k \end{pmatrix} \quad F = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ f_n(t) \end{pmatrix}$$

En appliquant le schéma d'Euler explicite à l'équation différentielle matricielle, on montre qu'à l'instant k, le système peut se mettre sous la forme $H \cdot X_k = G_k$, la matrice H étant de la forme suivante (dite tridiagonale) :

$$H = \begin{pmatrix} H_{00} & H_{01} & 0 & 0 & 0 \\ H_{10} & H_{11} & \ddots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & H_{n-1,n-1} & H_{n-1,n} \\ 0 & 0 & 0 & H_{n-1,n} & H_{n,n} \end{pmatrix}$$

Question 9 *L'équation $H \cdot X_k = G_k$ peut être résolue grâce à la méthode du pivot de Gauss. Donner les étapes de cette méthode ainsi que l'objectif ce chacune d'entre elles.*

Corrigé