1 Mise en situation

2 Résolution de l'équation différentielle

Question 1 *En utilisant le changement de variable proposé et en utilisant un schéma d'Euler explicite, mettre l'équation différentielle sous forme d'un système de deux équations différentielles d'ordre 1.*

Correction On a : $\begin{cases} y_2(t) = \dfrac{\mathrm{d}y_1(t)}{\mathrm{d}t} \\ y_1(t) + \dfrac{2\xi}{\omega_0}\cdot y_2(t) + \dfrac{1}{\omega_0^2}\cdot\dfrac{\mathrm{d}y_2(t)}{\mathrm{d}t} = K\cdot e(t) \end{cases}$.

En mettant le système sous forme numérique, on a : $\begin{cases} y_{2,k} = \dfrac{y_{1,k+1}-y_{1,k}}{\mathrm{h}} \\ y_{1,k} + \dfrac{2\xi}{\omega_0}\cdot y_{2,k} + \dfrac{1}{\omega_0^2}\cdot\dfrac{y_{2,k+1}-y_{2,k}}{\mathrm{h}} = K\cdot e_k \end{cases}$.

Question 2 *Donner la solution **numérique** de l'équation différentielle $y_2(t) = \dfrac{ds(t)}{dt}$ en utilisant un schéma d'Euler explicite.*

Correction D'après le schéma d'Euler explicite, on a $y_{2,k} = \dfrac{y_{1,k+1}-y_{1,k}}{\mathrm{h}}$

Question 3 *Mettre la solution **numérique** sous la forme :* $\begin{cases} y_{1,k+1} = y_{1,k} + h\Phi_1(t_k, y_{1,k}, y_{2,k}) \\ y_{2,k+1} = y_{2,k} + h\Phi_2(t_k, y_{1,k}, y_{2,k}) \end{cases}$ *avec* $\Phi_1(t_k, y_{1,k}, y_{2,k})$ *et* $\Phi_2(t_k, y_{1,k}, y_{2,k})$ *deux fonctions à expliciter.*

Correction En appliquant le changement de variable à l'équation différentielle, on a :

$$\begin{cases} y_2(t) = \dfrac{\mathrm{d}y_1(t)}{\mathrm{d}t} \\ y_1(t) + \dfrac{2\xi}{\omega_0}\cdot y_2(t) + \dfrac{1}{\omega_0^2}\cdot\dfrac{\mathrm{d}y_2(t)}{\mathrm{d}t} = K\cdot e(t) \end{cases} \Leftrightarrow \begin{cases} y_2(t) = \dfrac{\mathrm{d}y_1(t)}{\mathrm{d}t} \\ \dfrac{\mathrm{d}y_2(t)}{\mathrm{d}t} = K\omega_0^2\cdot e(t) - 2\xi\omega_0\cdot y_2(t) - \omega_0^2\cdot y_1(t) \end{cases}$$

En utilisant la formulation de Cauchy, on a :

$$\begin{cases} \Phi_1(t, y_1(t), y_2(t)) = y_2(t) \\ \Phi_2(t, y_1(t), y_2(t)) = K\omega_0^2\cdot e(t) - 2\xi\omega_0 y_2(t) - \omega_0^2 y_1(t) \end{cases}.$$

En utilisant le schéma d'Euler explicite, on a donc :

$$\begin{cases} y_{1,k+1} = y_{1,k} + h\Phi_1(t_k, y_{1,k}, y_{2,k}) \\ y_{2,k+1} = y_{2,k} + h\Phi_2(t_k, y_{1,k}, y_{2,k}) \end{cases} \Rightarrow \begin{cases} y_{1,k+1} = y_{1,k} + h\,y_{2,k} \\ y_{2,k+1} = y_{2,k} + h\left(K\omega_0^2\cdot e_k - 2\xi\omega_0 y_{2,k} - \omega_0^2 y_{1,k}\right) \end{cases}.$$

Les variables K, ω_0, ξ, h sont données. On considère que les fonctions $\Phi_1(t_k, y_{1,k}, y_{2,k})$ et $\Phi_2(t_k, y_{1,k}, y_{2,k})$ ont été implémentées et peuvent être évaluées en utilisant les fonctions Python `phi_1` et `phi_2` d'arguments t_k, $y_{1,k}$ et $y_{2,k}$. On note T le temps de simulation.

Question 4 *Donner en Python la suite d'instructions permettant de disposer des listes* `les_y1` *et* `les_y2`.

Correction

```
les_y1=[0] #initialisation avec conditions initiales
les_y2=[0]
n=int(T/h)
for i in range(1,n):
    les_y1.append(les_y1[-1]+h*phi_1(i*h,les_y1[-1],les_y2[-1])
    les_y2.append(les_y2[-1]+h*phi_2(i*h,les_y1[-1],les_y2[-1])
```

Question 5 *Donner en Python la suite d'instructions permettant de tracer la réponse du système en fonction du temps.*

Correction

```
les_t=[i*h for i in range(n)]
plt.plot(les_t,les_y1)
plt.show()
```

3 Recherche du temps de réponse à 5%

Question 6 *Pour $K=1$ et pour une entrée de type échelon, tracer l'allure de la réponse temporelle pour $\xi<0,7$ et pour $\xi>1$. Pour chacune des deux courbes, tracer la bande à $\pm 5\%$ et repérer le temps de réponse à 5%.*

Correction Voir le cours

Pour un système du second ordre de gain $K=1$ sollicité par un échelon unitaire, il est nécessaire de résoudre l'équation $s(t)=0,95$ et/ou $s(t)=1,05$ pour déterminer le temps de réponse à 5%.

Question 7 *En illustrant votre réponse par des schémas (un schéma par méthode), expliquer la méthode de résolution de l'équation $s(t)=0,95$ en utilisant la méthode de Newton puis en utilisant la méthode par dichotomie.*

Correction Voir le cours

Question 8 *Donner l'algorithme de la méthode de Newton.*

Correction Voir le cours

Question 9 *Pour la détermination du temps de réponse à 5%, analyser les deux méthodes de résolution. Une méthode vous semble-t-elle plus appropriée ? Justifier. Quel(s) problème(s) peut-on rencontrer ? Comment l'(es) éviter ?*

Correction Pour la solution avec $\xi>1$, la courbe coupe une seule fois la bande à $\pm 5\%$. Les deux méthodes pourraient être utilisées puisque sur l'intervalle de temps étudié un seul zéro est obtenu. La difficulté rencontrée avec la méthode de Newton est le choix de la valeur de départ x_0 qui ne peut pas être égale à 0 où la courbe présente une tangente horizontale. La difficulté rencontrée avec la méthode de Dichotomie est le choix des deux bornes a et b. Pour la solution avec $\xi<0,7$, la courbe coupe plusieurs fois la bande à $\pm 5\%$. Nous n'avons besoin que de la dernière valeur. Là aussi, les deux méthodes sont difficilement exploitables : choix des bornes pour Dichotomie et choix du point de départ pour Newton avec une courbe qui change régulièrement de pente.

Question 10 *Proposer une méthode pour déterminer le temps de réponse à 5%.*

Correction Méthode "lente" d'approche du zéro par pas constant en partant de zéro quand $\xi>0.7$ et en partant d'une "grande" valeur de t quand $\xi<0,7$.

4 Tracé de l'abaque de temps de réponse à 5 %

Question 11 *Donner, en Python, le contenu de la fonction f_s permettant de définir la fonction $(t\omega_0,\xi)\rightarrow s(t\omega_0,\xi)$ dans le cas où $\xi\in\mathbb{R}_+^*$. On donne ci-dessous les spécifications de la fonction.*

```python
def f_s(tom0,z) :
    """
    Fonction permettant de calculer la réponse indicielle d'un système du second ordre.
    Entrées :
        * tom0, float : temps de réponse réduit
        * z, float : coefficient d'amortissement
    Sortie :
        * s(tom0,z), float
    """
    if z<0 :
        return None
    elif z<1 :
        return f_pseudo(tom0,z)
    elif z==1:
        return f_critique(tom0)
    else :
        return f_aperiodique(tom0,z)
```

On note t_r le temps de réponse à 5%. L'abaque du temps de réponse permet de tracer le produit $t_r\omega_0$ en fonction du coefficient d'amortissement ξ.

Question 12 *Quelle est la valeur finale prise par $s(t)$?*

Correction La valeur finale est 1.

Question 13 *Écrire en Python la fonction* `is_in_strip` *ayant les spécifications suivantes :*

```python
def is_in_strip(x):
    """
    Fonction permettant de savoir si une valeur est dans la bande des + ou - 5% de la valeur finale.
    Entrée :
        x, flt : réel
    Sortie :
        True si la valeur est dans la bande à+ ou - 5%
        False si la valeur n'est pas dans la bande à+ ou - 5%
    """
    if x>.95 and x<1.05:
        return True
    else:
        return False
```

Question 14 *Expliquer le mode de recherche du temps de réponse à 5% dans le cas où $z < 0,7$ puis dans le cas où $z \geq 0,7$. Pourquoi distingue-t-on ces 2 cas ?*

Correction Pour déterminer le temps de réponse à 5%, on cherche le dernier temps pour lequel, le signal est dans la bande à plus ou moins 5%. En régime permanent, le signal est dans la bande. En « remontant le temps » la première valeur hors de la bande correspond donc au temps de réponse recherché.

Lorsque $\xi < 0,7$, le système est oscillant, et le temps de réponse est mesuré lorsque les oscillations deviennent «petites». Il est donc préférable de partir de la fin.

Lorsque $\xi > 0,7$, on sait que dès lors que le signal entre dans la bande, il n'en sortira plus. Il est donc plus rapide de commencer par le début.

Question 15

Correction

1. *Donner l'intervalle de variation de* `z` *pour le tracé demandé.*
2. *Donner le pas de* `z` *sur chacun des intervalles.*
3. *Pourquoi ne pas conserver le même pas sur chacun de ces intervalles ?*
4. *En vous aidant du tracé de l'abaque, expliquer pourquoi* `tom0` *est calculé différemment suivant la valeur de* `z` *? Expliquer le choix des valeurs des arguments de la fonction* `calcul_tom0` *dans chacun des cas. Comment pourrait-on réduire le temps de calcul pour construire la liste* `yy` *?*

Correction
1. z variera de 0,01 à 100.
2. On a :
   - si $0,01 < z < 0,1$: $pas z = 0,001$;
   - si $0,1 < z < 1$: $pas z = 0,01$;
   - si $1 < z < 10$: $pas z = 0,1$;
   - si $10 < z < 100$: $pas z = 1$.
3. La courbe est tracée en diagramme log-log. On adopte donc un pas différent par décade.
4. Lorsque $\xi < 0.7$, $tom0$ décroit lorsque z croît. Pour trouver plus rapidement $tom0$ on peut donc utiliser la valeur prise précédemment (dans ce cas, la recherche de tom0 dans la fonction calcul_tom0 se fait à rebours). A l'inverse, lorsque $\xi > 0.7$, $tom0$ croit lorsque z croît. On choisit dans ce cas, de partir à chaque fois de 0 (en modifiant le code, il serait possible de partir du $tom0$ calculé précédemment, mais il faudrait réfléchir à la gestion de la bascule de $\xi < 0,7$ à $\xi > 0,7$).

Question 16 *On souhaite stocker les données des listes* `xx` *et* `yy` *dans un fichier texte encodé en ASCII. Chacun des nombres doit être stocké sur 17 caractères. Indiquer la taille du fichier ainsi créé.*

Correction 90 valeurs sont calculées pour chaque intervalles de z. xx et yy ont donc une taille de 360 éléments. En prenant en compte les 34 caractères, l'espace et le retour à la ligne (2 caractères), une ligne a donc une taille de 37 octets. Le fichier sera donc d'approximativement de 13 320 octets.