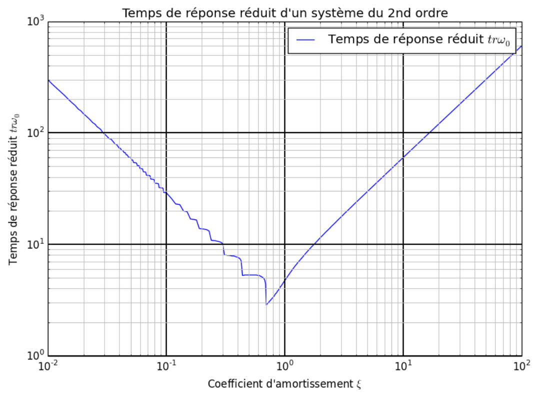
1 Mise en situation

L'équation différentielle d'un système du second ordre peut se mettre sous la forme :

$$s(t)+\frac{2\xi}{\omega_0}\cdot\frac{\mathrm{d}s(t)}{\mathrm{d}t}+\frac{1}{\omega_0^2}\cdot\frac{\mathrm{d}^2 s(t)}{\mathrm{d}t^2}=K\cdot e(t)$$

en notant :
- K : le gain statique ;
- ξ : le coefficient d'amortissement ;
- $e(t)$ et $s(t)$: l'entrée et la sortie du système.

Objectif L'objectif de ces travaux est de construire le programme permettant de tracer l'abaque du temps de réponse réduit utilisé en asservissement pour connaître le temps de réponse à 5% des systèmes d'ordre 2.



2 Résolution de l'équation différentielle

On cherche à résoudre $s(t)+\frac{2\xi}{\omega_0}\cdot\frac{\mathrm{d}s(t)}{\mathrm{d}t}+\frac{1}{\omega_0^2}\cdot\frac{\mathrm{d}^2 s(t)}{\mathrm{d}t^2}=K\cdot e(t)$. On pose : $\left\{\begin{array}{l} y_1(t)=s(t)\\ y_2(t)=\frac{\mathrm{d}s(t)}{\mathrm{d}t}\end{array}\right.$.

Question 1 *Donner la solution **numérique** de l'équation différentielle du second ordre en utilisant le changement de variable proposé et le schéma d'Euler explicite. On notera h le pas du schéma.*

Question 2 *Donner la solution **numérique** de l'équation différentielle $y_2(t)=\frac{ds(t)}{dt}$ en utilisant un schéma d'Euler explicite. Vous mettrez la solution sous la forme $\left\{\begin{array}{l} y_{1,k+1}=y_{1,k}+h\Phi_{1,k}\\ y_{2,k+1}=y_{2,k}+h\Phi_{2,k}\end{array}\right.$ avec $\Phi_{1,k}$ et $\Phi_{2,k}$ deux fonctions à expliciter.*

Les variables K, ω_0, ξ, h sont données. On considère que $s(0)=0$ et $s'(0)=0$. On considère que les fonctions $\Phi_1(k)$ et $\Phi_2(k)$ ont été implémentées et peuvent être évaluées en utilisant les fonctions Python `phi_1` et `phi_2` . On note T le temps de simulation.

Question 3 *Donner en Python la suite d'instructions permettant de disposer des listes* `les_y1` *et* `les_y2`.

Question 4 *Donner en Python la suite d'instructions permettant de tracer la réponse du système en fonction du temps.*

3 Recherche du temps de réponse à 5%

Question 5 *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 courbes deux courbes, afficher le temps de réponse à 5%.*

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

Question 6 *En utilisant des schémas (un schéma par méthode), expliquer la méthode de résolution de l'équation en utilisant la méthode de Newton et en utilisant une méthode par dichotomie.*

Question 7 *Donner l'algorithme de la méthode de Newton.*

Question 8 *Dans notre cas, quelle méthode semble la plus appropriée ? Quel(s) problème(s) peut-on rencontrer ? Comment l'(es) éviter ?*

Question 9 *Proposer une méthode pour déterminer le temps de réponse à 5%.*

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

On reprend l'équation différentielle présentée précédemment. On suppose que toutes les conditions initiales sont nulles. Pour une entrée unitaire de type échelon unitaire $e(t) = 1$, $K = 1$ et $t \geq 0$ on montre que :

- si $\xi < 1$, le régime est pseudo périodique et : $s(t) = 1 - \dfrac{e^{-\xi\omega_0 t}}{\sqrt{1-\xi^2}} \sin\left(\omega_0 t\sqrt{1-\xi^2} + \arcsin\sqrt{1-\xi^2}\right)$;
- si $\xi = 1$, le régime est critique et : $s(t) = 1 - (1+\omega_0 t)e^{-\omega_0 t}$;
- si $\xi > 1$, le régime est apériodique et : $s(t) = 1 + \dfrac{e^{-\omega_0 t\left(\xi+\sqrt{\xi^2-1}\right)}}{2\left(\xi\sqrt{\xi^2-1}+\xi^2-1\right)} - \dfrac{e^{-\omega_0 t\left(\xi-\sqrt{\xi^2-1}\right)}}{2\left(\xi\sqrt{\xi^2-1}-\xi^2+1\right)}$.

Dans la suite, on considèrera que s est une fonction du temps réduit $t \cdot \omega_0$ que l'on notera $t\omega_0$ ou `tom0`. On notera indifféremment le coefficient d'amortissement z ou ξ.

On dispose des fonctions Python `f_pseudo`, `f_critique` et `f_aperiodique` permettant d'évaluer la fonction pour s pour un couple $(t\omega_0, \xi)$.

Question 10 *Donner, en Python, le contenu de la fonction f_s permettant de définir la fonction $(t\omega_0, \xi) \longrightarrow s(t\omega_0\xi)$ 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, flt : temps de réponse réduit
        * z, flt : coefficient d'amortissement
    Sortie :
        * s(tom0,z), flt
    """
```

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

Question 12 *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 â+ ou - 5%
        False si la valeur n'est pas dans la bande â+ ou - 5%
    """
```

On donne la fonction suivante permettant de connaître le temps de réponse réduit à partir duquel la réponse indicielle d'un système est dans la bande à plus ou moins 5% pour un coefficient d'amortissement particulier.

■ Python
```
def calcul_tom0(z,tom0=500):
    """
    Recherche du temps de réponse à5%
    Entrées :
       * z, flt : coefficient d'amortissement
       * tom0 (flt, optionnel) : si non précisé, on calcule le temps de réponse en partant de tom0 = 500
    Sortie :
       * tom0 (flt) : temps de réponse à5%
    """
    pas_tom0=0.05
    x = f_s(tom0,z)
    if z<0.7:
        while is_in_strip(x) :
            tom0  = tom0 - pas_tom0
            x = f_s(tom0,z)
        tom0=tom0+pas_tom0
    else :
        while not is_in_strip(x) :
            tom0  = tom0 + pas_tom0
            x = f_s(tom0,z)
        tom0=tom0-pas_tom0
    return tom0
```

Question 13 *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 ? On pourra s'aider de l'abaque donné sur la première page.*

L'algorithme suivant permet de créer les listes xx et yy permettant de tracer l'abaque du temps de réponse réduit en fonction du coefficient d'amortissement.

■ Python
```
xx,yy = [],[]
n = 1000
z = 0.01
tom0 = 500
pasz = 0.01
while z<=100:
    if z<0.7:
        tom0=calcul_tom0(z,tom0)
    else :
        tom0=calcul_tom0(z,0)
    if z<0.1:
        pasz = 0.001
    elif z<1:
        pasz = 0.01
    elif z<10:
        pasz = 0.1
    else :
        pasz = 1
    xx.append(z)
    yy.append(tom0)
    z=z+pasz
```

Question 14

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 arguments de la fonction calcul_tom0 dans chacun des cas.*

Question 15 *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.*

Question supplémentaire

Question 16 *En utilisant la méthode des trapèzes, calculer l'intégrale de $s(t)$ pour un intervalle de T secondes, avec un pas de temps h.*