

D.S. 1

POINT MILIEU

1 Préliminaires

Question 1

Écrire une fonction `subdivise(a, b, p)` qui reçoit deux réels a et b et un entier strictement positif p et qui renvoie la subdivision de $[a; b]$ en p intervalles de même largeur sous la forme de la liste des $p + 1$ points de séparation : $x_0 = a < x_1 < \dots < x_{p-1} < x_p = b$ avec $x_{i+1} - x_i$ constant.

La fonction ne doit pas utiliser `linspace`.

Elle doit renvoyer une réponse censée même quand on a $a \geq b$.

Les paramètres d'une équation différentielle sont

1. une fonction `phi(y, t)` qui définit l'équation $y' = \varphi(y, t)$.

Par exemple pour $y' = t.(y + \sin(t))$ on peut écrire, en supposant que `sin` a été importée,

```
def phi(y, t):  
    return t*(y + sin(t))
```

2. La liste `T` des valeurs en lesquelles on souhaite calculer une valeur approchée de la solution.
3. La valeur de la condition initiale en `T[0]` : y_0 .

Question 2

Compléter la fonction ci-dessous pour qu'elle donne une solution d'une équation différentielle en utilisant la méthode d'Euler (explicite)

```
def Euler(phi, y0, T):  
    n = .....  
    # initialisation du résultat  
    # valeur initiale  
    for i in range(...):  
        dt = T[i+1] - T[i]  
        # calcul d'un nouveau point  
        # plusieurs lignes sont possibles  
    return Y
```

2 Justification mathématique

Dans cette partie, toutes les fonctions considérées sont définies et de classe \mathcal{C}^∞ sur le segment $[a; b]$; on note $M_n(f)$ la norme infinie de $f^{(n)}$ sur $[a; b]$.

Pour x_0 et x appartenant à $[a; b]$, on utilisera le développement de Taylor à l'ordre $n \leq N$ de la fonction f en x par rapport à x_0 avec la majoration du reste :

$$f(x) = f(x_0) + \sum_{k=1}^n \frac{(x-x_0)^k}{k!} f^{(k)}(x_0) + R_n(f, x_0, x) \text{ avec } |R_n(f, x_0, x)| \leq \frac{M_{n+1}}{(n+1)!} (x-x_0)^{n+1}$$

Le but de cette partie est de déterminer des approximations de $p(a, b) = \frac{f(b) - f(a)}{b - a}$ en fonction de valeurs de la dérivée de f : si on peut écrire

$$\left| \frac{f(b) - f(a)}{b - a} - \alpha \right| \leq \beta \cdot (b - a)^m$$

on dira que α est une **valeur approchée à l'ordre m** de $p(a, b)$.

On cherchera à obtenir une valeur maximale pour m et une valeur minimale pour β .

Par exemple la formule de Taylor à l'ordre 1 de f en b par rapport à a donne $f(b) = f(a) + (b - a)f'(a) + R_1(f, a, b)$ donc $\frac{f(b) - f(a)}{b - a} - f'(a) = \frac{R_1(f, a, b)}{b - a}$ puis

$$\left| \frac{f(b) - f(a)}{b - a} - f'(a) \right| \leq \frac{M_2(f)}{2} (b - a) : f'(a) \text{ est une valeur approchée à l'ordre 1 de } p(a, b).$$

Question 3

En utilisant les développements de Taylor à l'ordre 1 de f en a et en b par rapport à c avec $c \in [a; b]$, montrer que $f'(c)$ est une valeur approchée à l'ordre 1 de $p(a, b)$ pour tout c .

On ne cherchera pas la meilleure valeur possible pour la constante β .

Parmi les points de $[a; b]$, $m = \frac{a+b}{2}$ joue un rôle particulier.

On a, par exemple, $b - m = \frac{b-a}{2} = -(a - m)$ et $(b - m)^2 = (a - m)^2$.

Question 4

En utilisant les développements de Taylor à l'ordre 2 de f en a puis en b par rapport à m , montrer que $f'(m)$ est une valeur approchée à l'ordre 2 de $p(a, b)$. On devra trouver une valeur $\beta = \frac{M_3(f)}{24}$.

Il existe une autre valeur approchée d'ordre 2 qui est à l'origine de la méthode des trapèzes pour les intégrales et de la méthode de Heun pour les équations différentielles.

On pose $g(x) = f(x) - \frac{1}{2}(x - a)(f'(x) + f'(a))$.

Question 5

1. Calculer g' et g'' .
2. Prouver que $M_2(g) \leq \frac{b-a}{2} M_3(f)$
3. En utilisant le développement de Taylor à l'ordre 1 de g en b par rapport à a , montrer que $\frac{1}{2}(f'(a) + f'(b))$ est une valeur approchée à l'ordre 2 de $p(a, b)$.

La démonstration proposée donne un majorant avec $\beta = \frac{M_3(g)}{4}$; on peut prouver un majorant avec $\beta = \frac{M_3(g)}{12}$, ce n'est pas demandé ici. Ce majorant est optimum, il est atteint pour $f : x \mapsto x^3$.

3 Applications

On déduit de la partie précédente que la dérivée au point milieu est un bon choix pour approcher un taux d'accroissement. Nous allons appliquer cette idée dans trois algorithmes numériques.

3.1 Calcul d'une intégrale

On a $\int_a^b f'(t)dt = f(b) - f(a)$ proche de $(b-a)f'(m)$.

Pour calculer une valeur approchée d'une intégrale on subdivise l'intervalle en n intervalles de même largeur sur lesquels on calcule l'intégrale à l'aide de l'approximation ci-dessus : si $x_k = a + k\frac{b-a}{n}$

on a $\int_a^b f(t)dt = \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(t)dt$ dont on calcule la valeur approchée

$$\int_a^b f(t)dt \sim \sum_{k=0}^{n-1} (x_{k+1} - x_k)f(m_k) \text{ avec } m_k = \frac{x_k + x_{k+1}}{2}$$

Question 6

Avec les notations ci-dessus, on pose $h = \frac{b-a}{n}$. Prouver que $m_k = a + (k + \frac{1}{2})h$.

Question 7

En déduire une fonction Python `integralePM(f, a, b, n)` (pour intégrale avec Point Milieu) qui calcule une valeur approchée de $\int_a^b f(t)dt$ en utilisant la méthode décrite ci-dessus.

On privilégiera une fonction qui ne calcule qu'une fois le rapport $\frac{b-a}{n}$.

Par exemple, l'intégrale entre 0 et 1 de f vaut π

```
def f(x):
    return 4/(1+x**2)
```

`integralePM(f, 0, 1, 100)` donne 3.1416009869231245, l'erreur est inférieure à 10^{-5} .

3.2 Résolution d'équations différentielles

- f est la solution de l'équation différentielle $y' = \phi(y, t)$ avec $f(t_0) = y_0$.
- On approche f aux points de $T = [t_0, t_1, \dots, t_{n-1}]$ en des valeurs placées dans $Y = [y_0, y_1, \dots, y_{n-1}] : y_i \sim f(t_i)$.
- On veut utiliser la formule du point milieu :

$$y_{i+1} = y_i + (t_{i+1} - t_i)f'(m_i) \text{ avec } m_i = \frac{1}{2}(t_i + t_{i+1})$$

- On a $f'(m_i) = \varphi(f(m_i), m_i)$ mais on ne connaît pas $f(m_i)$, on l'approche par la méthode d'Euler explicite :

$$f(m_i) \sim z_i = y_i + \frac{1}{2}(t_{i+1} - t_i)\varphi(y_i, t_i)$$

Question 8

Écrire une fonction `point_milieu(phi, y0, T)` qui calcule et renvoie la liste des valeurs approchées de la solution avec la méthode exposée ci-dessus.

3.3 Dérivation discrète

La méthode du point milieu peut être lue de la droite vers la gauche pour calculer une valeur approchée de la dérivée.

Question 9

Pour a et b réels, on pose $h = \frac{b-a}{2}$ et $m = \frac{a+b}{2}$.

Déterminer a et b en fonction de m et h .

En déduire que l'approximation du point milieu peut s'écrire $f'(m) \sim \frac{f(m+h) - f(m-h)}{2h}$.

On suppose qu'une fonction f est définie par la liste de ses valeurs en des points de $T = [t_0, t_1, \dots, t_{n-1}]$ **régulièrement espacés** : $t_{i+1} - t_i = h$ constant.

On a donc une liste $Y = [y_0, y_1, \dots, y_{n-1}]$ avec $y_i = f(t_i)$ ¹.

Question 10

Montrer que la méthode du point milieu donne l'approximation

$$f'(t_i) \sim \frac{y_{i+1} - y_{i-1}}{2h} \text{ pour } 1 \leq i \leq n-2$$

Que peut-on choisir pour approcher $f'(t_0)$ et $f'(t_{n-1})$?

Question 11

En utilisant ces formules, écrire une fonction de docstring :

```
def derivee_symetrique(Y, T):
    """ Entrées : une liste T régulièrement espacée
        : une liste Y de même longueur avec
            Y[i] = f(T[i])
        Sortie : une liste DY de même longueur
            DY[i] est une approximation de f'(T[i]) """
```

La fonction f n'est pas sensée connue.

On ne calculera h qu'une seule fois.

1. Plus généralement y_i est une valeur approchée de $f(t_i)$.

4 Modélisation

Question 12

En utilisant les équations de Maxwell simplifiées par l'A.R.Q.S. et la loi d'Ohm locale montrer que le vecteur \vec{j} vérifie $\vec{\Delta}\vec{j} - \mu_0\gamma\frac{\partial}{\partial t}\vec{j} = 0$ dans un conducteur.

On utilisera la formule $\text{rot}(\text{rot}\vec{A}) = \text{grad}(\text{div}\vec{A}) - \vec{\Delta}\vec{A}$.

γ est la conductivité électrique du milieu.

On se place dans le cas d'un conducteur cylindrique d'axe $0x$ en régime sinusoïdal de pulsation ω . On fait les hypothèses simplificatrices suivantes, en plus de celles utilisées ci-dessus.

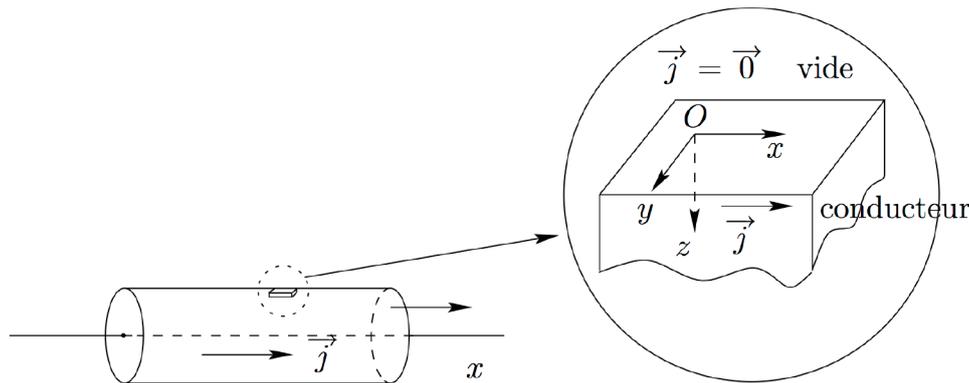


FIGURE I.1 – Dessin de M Reix

- On assimile le cylindre à un demi-espace, on néglige la courbure.
- Le vecteur j est dirigé selon $0x$
- Il ne dépend que du temps et de la distance à l'axe, donc de la hauteur z
- Il dépend sinusoïdalement du temps, avec une pulsation ω .

On peut donc écrire $\vec{j} = j_x \cdot \vec{e}_x = R(z) \cdot \cos(\omega \cdot t + \theta(z)) \vec{e}_x$.

Pour simplifier les calculs on introduit la fonction complexe

$$\underline{j} = \mathbb{R}(z) \cdot e^{i\theta(z)} \cdot e^{i\omega t} \cdot \vec{e}_x = J(z) \cdot e^{i\omega t} \cdot \vec{e}_x$$

On a employé i pour $\sqrt{-1}$ afin de ne pas confondre avec la fonction \vec{j} .

\vec{j} est la partie réelle de \underline{j} dont on suppose qu'elle vérifie l'équation donnée dans l'exercice 12.

Question 13

Montrer que la fonction J est solution de

$$J''(z) - i\omega\mu_0\gamma J(z) = 0$$

On se ramène à une équation différentielle vectorielle du premier ordre en posant $u = \begin{pmatrix} J \\ J' \end{pmatrix}$:

$$u' = \begin{pmatrix} J' \\ J'' \end{pmatrix} = \begin{pmatrix} J' \\ i\omega\mu_0\gamma J \end{pmatrix} = \Psi(u, z)$$

Les tableaux **numpy** permettent de faire les calculs vectoriels et acceptent les nombres complexes.

- Le complexe $a + ib$ est défini par `complex(a, b)`.
- Les opérations sont les mêmes
- La partie réelle d'un complexe z est `z.real`, sa partie imaginaire est `z.imag`

Par exemple une condition initiale $u_0 = (1, (1 + 2i) \cdot \sqrt{2})$ peut s'écrire

```
u0 = np.array([1, complex(1, 2) * 2**0.5])
```

On définit les constantes globale utilisées

```
mu0 = 4*np.pi*1e-7
gamma = 6*1e7      # conductivité du cuivre
omega = 2*np.pi*50 # pour 50 Hz
z0 = 2e-3         # profondeur étudiée, 2mm
```

Question 14

Écrire une fonction `psi(u, z)` qui reçoit un vecteur `u` à deux composantes et une valeur de `z` et qui renvoie le vecteur à deux composantes défini ci-dessus.

On peut alors définir une liste de hauteurs `Z = subdivise(0, z0, 1000)`

et calculer une solution `U = point_milieu(psi, u0, Z)`. `U` est alors une liste de vecteurs à deux composantes qui correspondent respectivement à $J(r)$ et $J'(r)$

Question 15

Écrire les instructions qui permettent de calculer la liste des $J(r)$, notée `J` : ses composantes sont les `U[i][0]` pour les indices `i` possibles.

Il n'est pas nécessaire d'écrire une fonction.

`J` est une liste de complexes : les modules de ces complexes correspondent aux amplitudes (notées `R` ci-dessus) et leurs argument à la phase (θ).

Question 16

Écrire les instructions qui permettent de calculer la liste `R` des amplitudes et la liste `th` des phases. L'argument d'un nombre complexe est calculé par la fonction `np.angle(z)` du module `numpy`, son module est obtenu par la fonction `abs`.

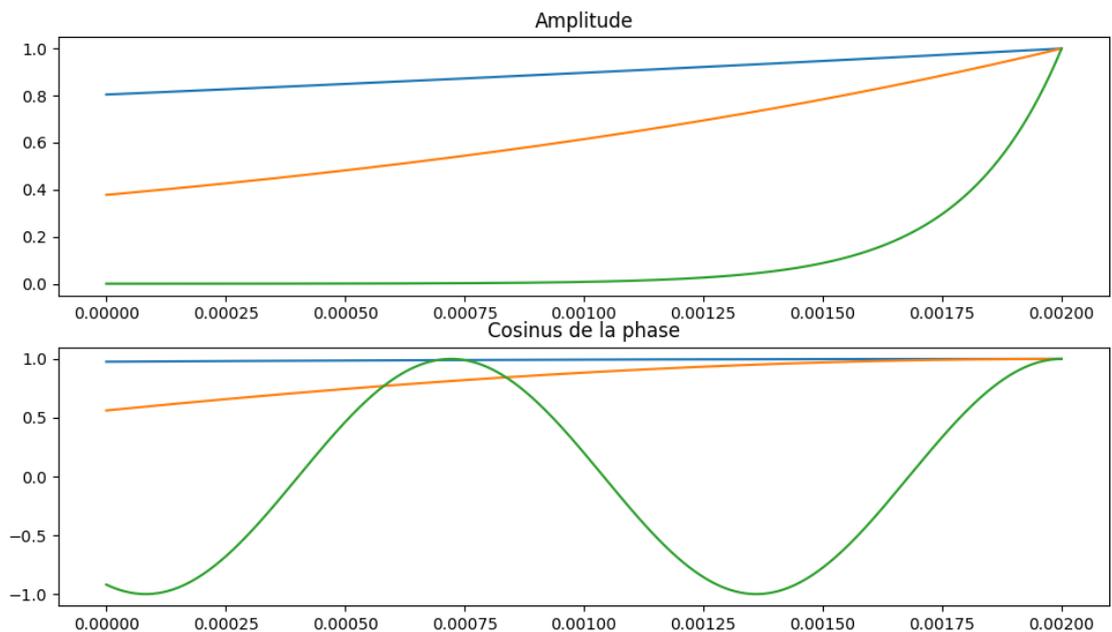
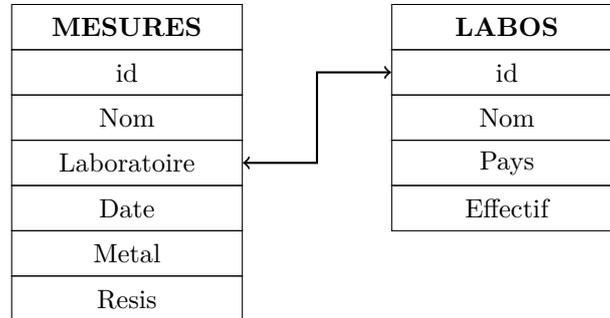


FIGURE I.2 – Résultats pour 0, 1000 et 100000 Hertz

5 Exploitation de données

Une base de données contient des informations sur les mesures faites par différents laboratoires pour mesurer la résistivité de différents métaux. Elle comporte deux tables.



La table **MESURES** contient les attributs suivants :

- id** : identifiant de la manipulation (entier), clé primaire,
- Nom** : nom du chercheur (chaîne de caractères),
- Laboratoire** : identifiant du laboratoire (entier),
- Date** : année de la mesure (entier),
- Metal** : nom du métal mesuré (chaîne de caractères),
- Resis** : résistivité mesurée (réel).

La table **LABOS** contient les attributs suivants :

- id** : identifiant du laboratoire (entier), clé primaire,
- Nom** : nom du laboratoire (chaîne de caractères),
- Pays** : pays du laboratoire (chaîne de caractères),
- Effectif** : nombre du chercheurs du laboratoire (entier).

Question 17

Quelle est l'utilité d'une clé primaire ?

Pouvait-on choisir une autre clé primaire dans la table MESURES ?

Question 18

Écrire une requête SQL permettant d'extraire les résistivités mesurées pour le métal "Cuivre" en donnant l'année de la mesure.

Question 19

Écrire une requête SQL permettant d'extraire la moyenne des mesures de résistivité du métal "Titane".

Question 20

Écrire une requête SQL permettant d'extraire la moyenne des résistivité mesurées par metal

Question 21

Écrire une requête SQL permettant de déterminer, pour chaque pays, le nombre total de mesures effectuées en 2015.

Question 22

Écrire une requête SQL permettant de déterminer les laboratoires pour lesquels le nombre moyen de mesures effectué par membre du laboratoire est inférieur à 2.

Question 23

Déterminer les mesures (identifiant, année, nom du chercheur, métal) qui s'écartent de plus de 2 pour cent de la moyenne des mesure pour le métal considéré.

On pourra utiliser le résultat de la question 20

6 Algorithmique

Pour déterminer la résistivité, on doit chercher la valeur en laquelle une fonction admet un minimum. Pour que le minimum soit déterminé sans ambiguïté on admet que la fonction f est continue et **unimodale** avec minimum sur un intervalle $[\alpha; \beta]$, c'est-à-dire qu'il existe $\mu \in]\alpha; \beta[$ tel que

- f est strictement décroissante sur $[\alpha; \mu]$ et
- f est strictement croissante sur $[\mu; \beta]$.

Le but est de déterminer μ . Pour cela on va l'encadrer par deux suite adjacentes.

On suppose que l'on connaît deux réels a_n et b_n tels que $\alpha \leq a_n \leq \mu \leq b_n \leq \beta$.

On définit $c_n = \frac{2a_n + b_n}{3}$ et $d_n = \frac{a_n + 2b_n}{3}$: on a $a_n < c_n < d_n < b_n$.

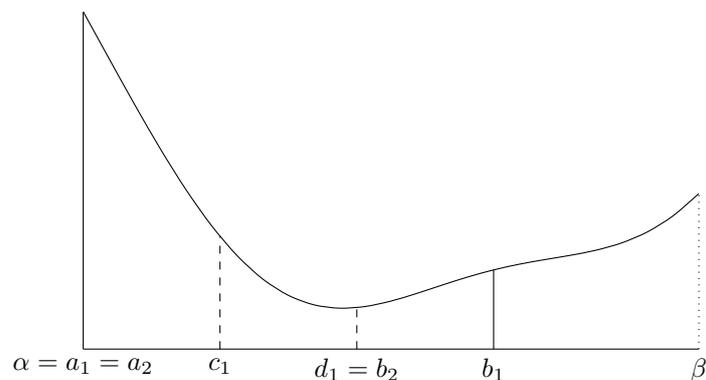
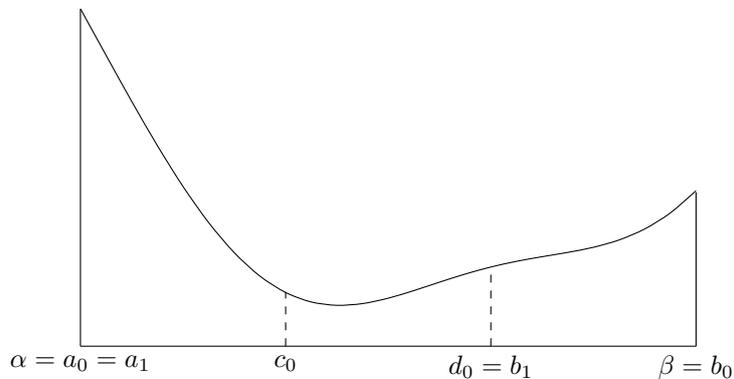
Question 24

Prouver que si on a $f(c_n) \leq f(d_n)$ alors on a $\mu \leq d_n$.

Prouver que si on a $f(c_n) \geq f(d_n)$ alors on a $c_n \leq \mu$.

On définit donc $a_{n+1} = a_n$ et $b_{n+1} = d_n$ si $f(c_n) \leq f(d_n)$ et $a_{n+1} = c_n$, $b_{n+1} = b_n$ sinon.

En commençant par $a_0 = \alpha$ et $b_0 = \beta$ on définit ainsi les deux suites par récurrence.



Question 25

Écrire `minimumT(f, alpha, beta, epsilon = 1e-8)` qui détermine une valeur approchée de μ en appliquant l'algorithme en renvoyant a_n dès qu'on a $|b_n - a_n| < \epsilon$.

Question 26

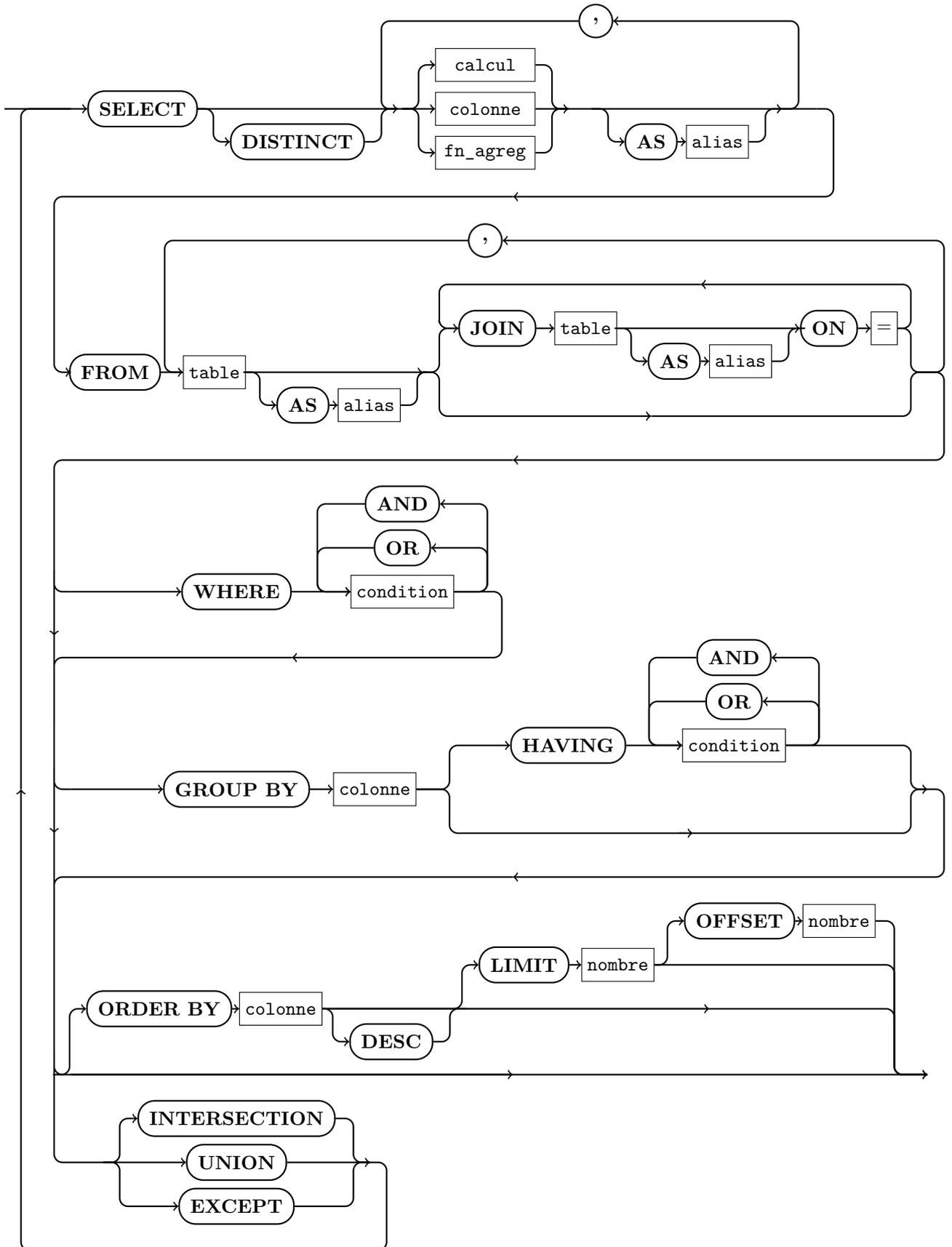
Évaluer le nombre de calcul de $f(x)$ effectués par l'appel de `minimumT(f, alpha, beta, epsilon)` en fonction de $\beta - \alpha$ et ϵ .

7 Aide-mémoire SQL

La page suivante montre la structure d'une requête SQL : on suit le chemin fléché, les bifurcation expriment un choix entre plusieurs possibilités. Un de ces choix est parfois de revenir en arrière pour répéter une portion de requête.

Quelques remarques

1. Dans la clause SELECT, `fn_agreg` désigne une fonction d'agrégation : c'est l'application d'une fonction statistique à une colonne. Les fonction statistiques usuelles sont COUNT, SUM, AVG, MAX, MIN.
2. Lorsqu'il y a une fonction d'agrégation sans clause GROUP BY, le résultat n'a qu'une ligne, il n'y a pas de sens à faire afficher une colonne.
3. Lorsqu'il y a une clause GROUP BY, la clause SELECT doit contenir la colonne qui a servi au regroupement ; les seuls autres résultats qui peuvent être affichés sont des fonctions d'agrégation ou des calculs avec ces mêmes fonctions.
4. Les conditions utilisées après un HAVING sont celles qui utilisent le résultat d'une fonction d'agrégation. Toutes les autres conditions sont à écrire dans la clause WHERE.
5. Dans une jointure "=" désigne une condition d'égalité structurelle entre des colonnes de deux tables, l'une étant une des table que l'on vient de joindre.
6. Lorsque l'on utilise plusieurs tables, dans le cas d'une jointure, il faut préfixer les noms de colonnes par le nom de la table partout où on utilise une colonne. On doit remplacer le nom de la table par son alias lorsqu'il existe ; il sera donc souvent utile de donner un alias court.
7. Le résultat d'une requête est une table. On peut donc utiliser cette table dans une autre requête en écrivant la requête première entre parenthèses.
8. De même, lorsqu'une requête renvoie un résultat unique, on peut utiliser la valeur de ce résultat dans une condition ; on écrit la requête entre parenthèses.
9. Dans la clause ORDER BY on peut remplacer le nom d'une colonne par son numéro d'ordre dans la liste de la clause SELECT. On peut aussi utiliser l'alias défini dans la clause SELECT.



8 Solutions

Solution de la question 1 -

```
def subdivide(a, b, p):
    h = (b-a)/p
    X = [0]*(p+1)
    for i in range(p+1):
        X[i] = a + i*h
    return X
```

Solution de la question 2 -

```
def Euler(phi, y0, T):
    n = len(T)
    Y = [0]*n
    Y[0] = y0
    for i in range(...):
        dt = T[i+1] - T[i]
        Y[i+1] = Y[i] + dt*phi(Y[i], T[i])
    return Y
```

Solution de la question 3 -

$f(b) = f(c) + (b - c)f'(c) + R_1(f, c, b)$ et $f(a) = f(c) + (a - c)f'(c) + R_1(f, c, a)$.

En soustrayant, on obtient

$$f(b) - f(a) = ((b - c) - (a - c))f'(c) + R_1(f, c, b) - R_1(f, c, a)$$

donc $f(b) - f(a) = (b - a)f'(c) + R_1(f, c, b) - R_1(f, c, a)$ puis

$$\left| \frac{f(b) - f(a)}{b - a} - f'(c) \right| = \frac{|R_1(f, c, b) - R_1(f, c, a)|}{b - a} \leq \frac{|R_1(f, c, b)| + |R_1(f, c, a)|}{b - a}.$$

Or on a $|R_1(f, c, b)| \leq \frac{M_2(f)}{2}(b - c)^2 \leq \frac{M_2(f)}{2}(b - a)^2$ et aussi $|R_1(f, c, a)| \leq \frac{M_2(f)}{2}(b - a)^2$ d'où

$$\left| \frac{f(b) - f(a)}{b - a} - f'(c) \right| \leq \frac{M_2(f)}{2}(b - a) + \frac{M_2(f)}{2}(b - a) = M_2(f)(b - a).$$

Solution de la question 4 -

En soustrayant $f(b) = f(m) + (b - m)f'(m) + \frac{1}{2}(b - m)^2 f''(c) + R_3(f, m, b)$

et $f(a) = f(m) + (a - m)f'(m) + \frac{1}{2}(a - m)^2 f''(c) + R_3(f, m, a)$, on obtient

$f(b) - f(a) = ((b - c) - (a - c))f'(m) + R_3(f, m, b) - R_3(f, m, a)$ donc

$$\left| \frac{f(b) - f(a)}{b - a} - f'(m) \right| = \frac{|R_3(f, m, b) - R_3(f, m, a)|}{b - a} \leq \frac{|R_3(f, m, b)| + |R_3(f, m, a)|}{b - a}.$$

Or on a $|R_3(f, m, b)| \leq \frac{M_3(f)}{6}(b - c)^3 \leq \frac{M_3(f)}{48}(b - a)^3$

et aussi $|R_3(f, m, a)| \leq \frac{M_3(f)}{48}(b - a)^3$ d'où $\left| \frac{f(b) - f(a)}{b - a} - f'(m) \right| \leq \frac{M_3(f)}{24}(b - a)^2$.

Solution de la question 5 -

- $g'(x) = f'(x) - \frac{1}{2}(f'(x) + f'(a)) - \frac{x-a}{2}.f''(x) = \frac{1}{2}(f'(x) - f'(a)) - \frac{x-a}{2}.f''(x)$; $g'(a) = 0$.

$$g''(x) = \frac{1}{2}f''(x) - \frac{x-a}{2}.f^{(3)}(x) - \frac{1}{2}f''(x) = -\frac{x-a}{2}.f^{(3)}(x).$$

- $|g''(x)| \leq \frac{|x-a|}{2}|f^{(3)}(x)| \leq \frac{b-a}{2}M_3(f)$, ce qui majore $M_2(g)$.

- $|g(b) - g(a) - (b - a)g'(a)| \leq \frac{(b-a)^2}{2}M_2(g)$ donne, en divisant par $b - a$ le résultat.

Solution de la question 6 -

$$m_k = \frac{a + k\frac{b-a}{n} + a + (k+1)\frac{b-a}{n}}{2} = \frac{2a + (2k+1)\frac{b-a}{n}}{2} = a + \left(k + \frac{1}{2}\right)h.$$

Solution de la question 7 -

```
def integralePM(f, a, b, n):
    h = (b-a)/n
    m = a + h/2
    somme = 0
    for i in range(n):
        somme = somme + f(m)
        m = m + h
    return somme*h
```

Solution de la question 8 -

```
def point_milieu(phi, y0, T):
    n = len(T)
    Y = [0]*n
    Y[0] = y0
    for i in range(n-1):
        dt = T[i+1] - T[i]
        m = (T[i] + T[i+1])/2
        z = Y[i] + dt/2*phi(Y[i], T[i])
        Y[i+1] = Y[i] + dt*phi(z, m)
    return Y
```

Solution de la question 9 - $a = m - h$, $b = m + h$ et $b - a = 2h$ donnent

$$f(m) = f\left(\frac{a+b}{2}\right) \sim \frac{f(b) - f(a)}{b - a} = \frac{f(m+h) - f(m-h)}{2h}$$

Solution de la question 10 - $y_{i+1} = f(t_{i+1}) = f(t_i + h)$ et $y_{i-1} = f(t_{i-1}) = f(t_i - h)$.
On peut choisir les expressions moins précises $f'(t_0) \sim \frac{y_1 - y_0}{h}$ et $f'(t_{n-1}) \sim \frac{y_{n-1} - y_{n-2}}{h}$.

Solution de la question 11 -

```
def derivee_symetrique(Y, T):
    n = len(T)
    DY = [0]*n
    h = T[1] - T[0]
    DY[0] = (Y[1] - Y[0])/h
    DY[n-1] = (Y[n-1] - Y[n-2])/h
    for i in range(1, n-1):
        DY[i] = (Y[i+1] - Y[i-1])/(2*h)
    return DY
```

Solution de la question 12 - Les équations simplifiées donnent

$$\operatorname{div} \vec{B} = \vec{0}, \operatorname{div} \vec{E} = \vec{0}, \operatorname{rot} \vec{E} = -\frac{\partial}{\partial t} \vec{B} \text{ et } \operatorname{rot} \vec{B} = \mu_0 \vec{j}.$$

Alors $\operatorname{rot}(\operatorname{rot} \vec{E}) = \operatorname{rot}\left(-\frac{\partial}{\partial t} \vec{B}\right) = -\frac{\partial}{\partial t}(\operatorname{rot} \vec{B}) = -\mu_0 \frac{\partial}{\partial t} \vec{j}$ d'où

$$-\mu_0 \frac{\partial}{\partial t} \vec{j} = \operatorname{grad}(\operatorname{div} \vec{E}) - \vec{\Delta} \vec{E} = -\vec{\Delta} \vec{E} = -\vec{\Delta} \left(\frac{1}{\gamma} \vec{E}\right) \text{ avec la loi locale.}$$

Solution de la question 13 -

On a $\frac{\partial^2}{\partial x^2} \vec{j} = 0$, $\frac{\partial^2}{\partial y^2} \vec{j} = 0$, $\frac{\partial^2}{\partial z^2} \vec{j} = J''(z).e^{i\omega t}.e_x$ et $\frac{\partial}{\partial t} \vec{j} = i\omega J(z).e^{i\omega t}.e_x$

Solution de la question 14 -

```
def psi(u, z):
    y, dy = u
    ddy = 1j*mu0*gamma*omega*y
    return np.array([dy, ddy])
```

Solution de la question 15 -

```
n = len(U)
J = [0]*n
for i in range(n):
    J[i] = U[i][0]
```

Ou J = [u[0] for u in U]

Solution de la question 16 -

```
n = len(J)
R = [0]*n
th = [0]*n
for i in range(n):
    R[i] = abs(J[i])
    th[i] = np.angle(J[i])
```

Ou R = [abs(j) for j in J], th = [np.angle(j) for j in J]

Solution de la question 17 - Une clé primaire permet de différencier les différentes mesures. Non, il n'y a pas d'ensembles d'attributs ne contenant pas l'identifiant dont on puisse assurer qu'il permette de différencier les lignes de la table : un même chercheur peut avoir mesuré deux fois dans l'année la résistivité d'un métal et avoir trouvé la même valeur.

Solution de la question 18 -

```
select Date, Resis
from MESURES
where Metal = "Cuivre"
```

Solution de la question 19 -

```
select avg(resis)
from MESURES
where Metal = "Titane"
```

Solution de la question 20 -

```
select Metal, avg(Resis)
from MESURES
group by Metal
```

Solution de la question 21 -

```
select l.Pays, count()
from MESURES as m join LABOS as l on m.laboratoire = l.id
where m.Date = 2015
group by l.Pays
```

Solution de la question 22 -

```
select l.nom, count()/l.effectif as ratio
from MESURES as m join LABOS as l on m.laboratoire = l.id
group by l.nom
having ratio < 2
```

Solution de la question 23 -

```
select me.id, me.date, me.nom, me.metal
from MESURES as me
      join (select Metal, avg(Resis) as moy
           from MESURES
           group by Metal) as mo
      on me.metal = mo.metal
where me.resis < 0.98*mo.moy or me.resis > 1.02*mo.moy
```

Solution de la question 24 -

On prouve la contraposée, c'est-à-dire $\mu > d_n$ implique $f(c_n) > f(d_n)$.

Si on a $d_n < \mu$ alors, comme f est strictement décroissante sur $[\alpha; \mu]$ et $\alpha \leq a_n < c_n < d_n \leq \mu$, on peut conclure que $f(c_n) > f(d_n)$.

De même pour l'autre propriété.

Solution de la question 25 -

```
def minimumT(f, alpha, beta, epsilon = 1e-8):
    a, b = alpha, beta
    while abs(b-a) >= epsilon:
        c = (2*a+b)/3
        d = (a+2*b)/3
        if f(c) <= f(d):
            b = d
        else:
            a = c
    return a
```

Solution de la question 26 - On a $b_{n+1} - a_{n+1} = \frac{2}{3}(b_n - a_n)$ donc $b_n - a_n = (\beta - \alpha)\left(\frac{2}{3}\right)^n$.

Lorsque le programme renvoie le résultat on a $b_n - a_n = (\beta - \alpha)\left(\frac{2}{3}\right)^n < \varepsilon$ donc

$n > \frac{\ln\left(\frac{\varepsilon}{\beta - \alpha}\right)}{\ln\left(\frac{2}{3}\right)}$. Si on pose $K = \frac{\ln\left(\frac{\varepsilon}{\beta - \alpha}\right)}{\ln\left(\frac{2}{3}\right)}$, on stoppe au plus tard pour $n = \lfloor K \rfloor + 1$. Comme il y

a 2 calculs de f à chaque itération, on effectue donc au plus $2(\lfloor K \rfloor + 1)$ calculs de f .