Exercice 1 – Construction d'une gamme

En musique, construire une gamme revient à répartir des fréquences entre deux valeurs f_0 et $2f_0$, la première correspondant à une note de référence, et la seconde à la même note une octave au-dessus.

On considère que deux notes sonnent harmonieusement si elles sont en quinte, c'est-à-dire lorsque le rapport de leurs fréquences est 3/2 (ou 2/3). Un principe est donc le suivant : pour obtenir une nouvelle note de la gamme, on multiplie la dernière fréquence par $\frac{3}{2}$. On a donc d'abord f_0 (par exemple *do*), puis $\frac{3}{2}f_0$ (*sol*), etc. Mais la troisième note obtenue ainsi correspond à la fréquence $\frac{9}{4}f_0 > 2f_0$: on a changé d'octave. On ramène donc la note dans l'octave précédente en divisant le résultat par 2 : la note obtenue correspond donc à la fréquence $\frac{9}{8}f_0$ (*ré*). On poursuit la construction sur le même principe.

On note $(u_n)_{n\in\mathbb{N}}$ la suite des rapports à f_0 ainsi obtenue : $u_0 = 1$, $u_1 = \frac{3}{2}$, $u_2 = \frac{9}{8}$, etc.

Les questions **1**, **3.a**, **3.c** *et* **4** *sont des questions de mathématiques.*

1. Vérifier que pour tout $n \in \mathbb{N}$,

$$u_{n+1} = \begin{cases} \dfrac{3}{2}u_n & \text{si } 1 \leqslant u_n \leqslant \dfrac{4}{3} \\ \dfrac{3}{4}u_n & \text{si } \dfrac{4}{3} < u_n \leqslant 2. \end{cases}$$

2. Écrire une fonction `liste_fractions(n)`, qui renvoie la liste des termes $u_0, \ldots, u_n$.

3. Pour tout $n \in \mathbb{N}$, on note $q(n)$ l'unique entier vérifiant $2^{q(n)} \leqslant 3^n < 2^{q(n)+1}$.

a. Montrer que pour tout $n \in \mathbb{N}$,
$$u_n = \frac{3^n}{2^{q(n)}}.$$

b. Écrire une fonction `liste_exposants(n)`, faisant uniquement des calculs sur des entiers, qui renvoie la liste des termes $q(0), \ldots, q(n)$.
On veillera à limiter le nombre de multiplications.

c. Vérifier que pour tout $n \in \mathbb{N}$,
$$q(n) = \lfloor n \log_2(3) \rfloor.$$

d. En déduire une fonction `u(n)` permettant de calculer u_n sans la formule de récurrence.

e. Vérifier sur quelques exemples la cohérence des résultats.

f. Donner un avantage et un inconvénient de chacune des trois façons de calculer u_n : formule de récurrence, formule de la question **3.a** avec uniquement des calculs sur des entiers pour obtenir $q(n)$, formule utilisant la question **3.c**.

4. Expliquer pourquoi les valeurs u_n sont deux à deux distinctes et pourquoi, pour tout $n \in \mathbb{N}^*$, on a $u_n \neq 1$ et $u_n \neq 2$.

Ainsi, après avoir augmenté d'un certain nombre de quintes, on n'obtiendra jamais un multiple exact de f_0, et le procédé ne s'arrêtera pas. Il faut donc choisir de s'arrêter lorsque u_n est suffisamment proche de 1, ou 2.

5. Écrire une fonction `recherche(eps)`, d'argument `eps`, nombre flottant strictement positif (dont la valeur est notée ε), qui renvoie l'indice du premier terme u_n pour lequel on a

$$1 < u_n < 1 + \varepsilon.$$

Tester avec $\varepsilon = 0{,}1$ puis $\varepsilon = 0{,}02$.

La note correspondant à u_n sera ainsi proche de la note située une octave au-dessus de la note de départ, et la gamme est alors construite, à partir des rapports de fréquences donnés par $u_0, \ldots, u_{n-1}$. À partir des deux exemples ci-dessus, on construit les gammes de Pythagore, respectivement diatonique et chromatique.

6. Écrire des instructions permettant d'obtenir les fréquences, arrondies à l'unité par défaut, des douze premières notes ainsi construites, en imposant la présence du *la* de fréquence 440 Hz devant correspondre à u_3 (attention, cela ne signifie pas que *la* est la 4e note de cette gamme). On pourra utiliser la méthode `sort` pour trier les valeurs.

7. On s'intéresse à la répartition des nombres u_n dans l'intervalle $[1, 2[$. Dans les fonctions suivantes, on fera attention à ne pas recalculer de nombreuses fois les mêmes nombres u_n.

a. Écrire une fonction `test(x,eps,N)` d'arguments `x`, nombre flottant de l'intervalle $[1, 2[$, `eps`, nombre flottant strictement positif (dont la valeur est notée ε), et `N` entier naturel, renvoyant la liste des valeurs n pour lesquelles on a $n \leqslant N$ et $|x - u_n| < \varepsilon$.

b. Écrire une fonction `moyenne(a,b,N)` d'arguments `a`, `b`, nombres flottants de l'intervalle $[1, 2[$, et `N` entier naturel, renvoyant le nombre moyen de termes u_n vérifiant $n \leqslant N$ et $a \leqslant u_n \leqslant b$.

c. Écrire une fonction `moitie(N,eps)` d'arguments `eps`, nombre flottant strictement positif (dont la valeur est notée ε) et `N` entier naturel, renvoyant la plus petite valeur de b, à ε près, pour laquelle `moyenne(1,b,N)` $\geqslant \frac{1}{2}$.
Que peut-on conjecturer ?

Exercice 2 – Résolution numérique d'équations différentielles

On considère une équation différentielle d'ordre 2

$$x''(t) = f(t, x(t), x'(t)),$$

que l'on écrit sous la forme d'un système de deux équations

$$\begin{cases} x'(t) = v(t) \\ v'(t) = f(t, x(t), v(t)) \end{cases}$$

pour $t \in [0, T]$, avec $T > 0$ fixé. On fixe un pas de temps $h > 0$ et on calcule, pour tout $n \in \mathbb{N}$ tel que $nh \in [0, T]$, une approximation (x_n, v_n) de $(x(nh), v(nh))$.

1. Rappeler les relations de récurrence liant les valeurs x_n, x_{n+1}, v_n et v_{n+1} lors de l'application de la méthode d'Euler.

2. Programmer la méthode d'Euler (la fonction f sera passée en paramètre).

3. En donner la complexité temporelle dans le pire des cas, en fonction de T et h (on exprimera le résultat avec la notation O).

4. Mettre en œuvre cette méthode et représenter graphiquement les solutions obtenues, avec $T = 10$, $x(0) = 0$ et $v(0) = 1$, et plusieurs valeurs de h, dans les cas suivants :

a. L'équation (adimensionnée) d'un oscillateur amorti

$$x''(t) + \frac{1}{10}x'(t) + x(t) = 0,$$

i.e. avec

$$f : (t, x, v) \mapsto -x - \frac{1}{10}v.$$

Commenter les résultats.

b. L'équation

$$x''(t) - \sin(x(t)) = 0,$$

i.e. avec

$$f : (t, x, v) \mapsto \sin(x).$$

5. On remplace les relations de la question **1** par les relations

$$\begin{cases} x_{n+1} = x_n + hv_n \\ v_{n+1} = v_n + hf(nh, x_{n+1}, v_n) \end{cases}$$

Programmer cette méthode pour l'équation de la question **3.a**. Commenter.

6. On remplace les relations de la question **1** par les relations

$$\begin{cases} x_{n+1} = x_n + hv_{n+1} \\ v_{n+1} = v_n + hf(nh, x_{n+1}, v_n) \end{cases}$$

(méthode d'Euler implicite en x).

a. Écrire une équation vérifiée par x_{n+1}, en considérant connus x_n et v_n.

b. Rappeler le principe de la méthode de Newton pour résoudre une équation de la forme

$$F(x) = 0.$$

c. Dans le cas de l'équation de la question **3.b**, écrire une fonction `Newton(x,v,h,eps)` permettant de calculer une approximation de x_{n+1} à partir de x_n et v_n (valeurs de `x` et `v` respectivement). Préciser le critère d'arrêt utilisé (qui fera intervenir la valeur ε de `eps`).

d. Programmer alors la méthode de résolution décrite dans cette question **5**, pour l'équation de la question **3.b**.

e. Comparer les trois méthodes sur cette équation, avec $T = 100$, $h = 0{,}01$, $x(0) = 0$ et $v(0) = 1$.

7. Si x est une solution de l'équation différentielle $x''(t) = f(t, x(t), x'(t))$ sur $[0, T]$, on s'intéresse à la quantité

$$E(x) = \frac{1}{2} \int_0^T x(t)^2 \, dt,$$

qui peut correspondre à une énergie.

Écrire une fonction `energie(x,h)`, avec `x` une liste de valeurs approchées de x aux points d'une subdivision de pas h (liste qui pourrait par exemple être obtenue avec les algorithmes des questions précédentes), renvoyant une approximation de $E(x)$ par la méthode des trapèzes.

Correction disponible fin juillet à l'adresse : https ://padlet.com/aurelien_monteillet/dl_2018_2019