

Algorithmique et Programmation

Examen sur machine

G1: monasse(at)imagine.enpc.fr G2: boulc-ha(at)imagine.enpc.fr
G3: yoann(at)le-bars.net G4: arnaud.carayol(at)univ-mlv.fr
G5: liuz(at)imagine.enpc.fr G6: vialette(at)univ-mlv.fr

14/12/12

1 Enoncé

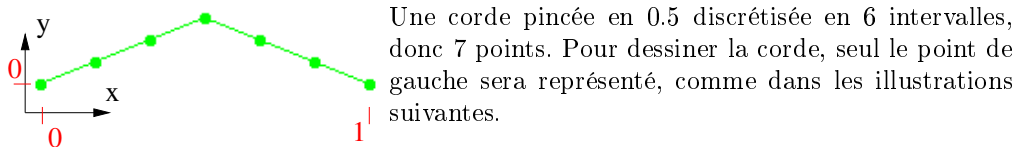
1.1 Corde vibrante

On se propose de simuler numériquement l'équation des ondes linéaire homogène, à laquelle obéit une corde vibrante fixée à une extrémité et avec une impulsion imposée à l'autre :

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad (1)$$

Nous allons utiliser un schéma numérique aux différences finies explicite pour simuler la forme de la corde en vibration.

La corde sera supposée de longueur 1 et discrétisée en n intervalles, c'est à dire qu'on ne traitera qu'un nombre fini de points (les noeuds) de coordonnées (x_i, y_i) pour $i \in [0, n]$. Les $x_i = i\Delta x/n$ sont répartis de façon homogène et les y_i varient en fonction des vibrations. Au repos et sans excitation, $\forall i \in [0, n], y_i = 0$.



1.2 Travail demandé

Il est plus important de livrer un code clair et qui compile sans warning, même s'il ne répond pas à toutes les questions. Pour cela, vérifiez à chaque étape que votre programme compile et se lance correctement. Dans ce devoir, prenez garde aux fuites mémoire : un principe à appliquer est qu'il faut autant de désallocations que d'allocations.

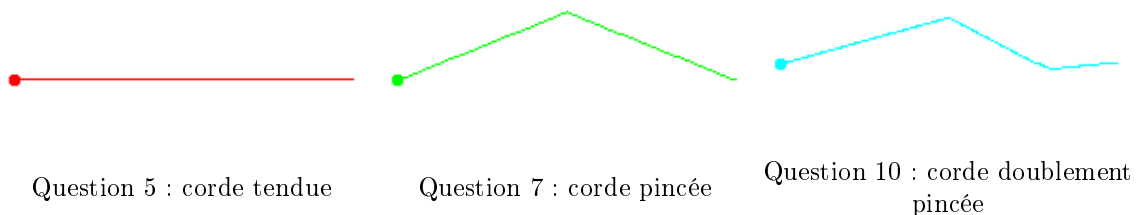


FIGURE 1 – Cordes au repos et pincées

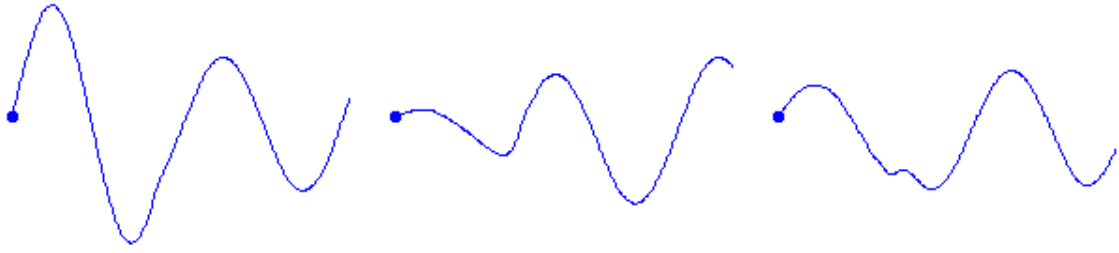


FIGURE 2 – Quelques étapes de la corde animée (question 12)

1. Créez un projet Imagine++ avec un fichier contenant le `main`, qui ouvre une fenêtre de taille fixe. Un fichier `CMakeLists.txt` à adapter à vos noms de fichier vous a été fourni.
2. On discrétise une corde en n intervalles, donc $n + 1$ points. Dans un fichier `corde.h`, écrivez une structure `Corde`, comprenant un tableau de `float` (les valeurs y des points) et le nombre d'intervalles n . Note : n sera variable au cours de l'énoncé (cf question 8).
3. Dans `corde.cpp`, implémentez des fonctions :
 - `cree_corde` prenant un nombre d'intervalles, allouant le tableau et renvoyant la structure corde ainsi créée ;
 - `detruis_corde` libérant le tableau ;
 - `copie_corde` qui renvoie une nouvelle corde, copie de celle passée en argument.
 N'oubliez pas de *déclarer* ces fonctions dans `corde.h`.
4. Ecrire la fonction `dessine_corde` prenant en argument une corde et une couleur. Elle réserve une marge de 10 pixels à gauche et à droite et calcule une échelle de manière à ce que la corde occupe toute la largeur restante. Cette valeur sert aussi de facteur d'échelle en y . On relie les points par des segments de droite.
5. Dessiner une corde sans excitation, tous les y_i à 0. Attendre un clic avant de l'effacer (tracer en blanc pour cela).
6. Ecrire une fonction `pince_corde` qui modifie une corde passée en paramètre en mettant le y_i à la valeur v (i correspondant à un x de la corde entre 0 et 1, x et v étant des paramètres de la fonction). De plus $y_0 = y_n = 0$ et les autres y_j sont répartis suivant une fonction affine, de telle sorte que l'on ait deux segments, l'un de $(0, 0)$ à (x_i, v) , l'autre de (x_i, v) à $(1, 0)$.
7. Tracer une corde pincée en $x = 0.5$ (milieu) de valeur $v = 0.2$.
8. Ecrire une fonction `double_corde` qui double le nombre d'intervalles de la corde passée en paramètre. Tracer la corde précédente après doublement dans un autre couleur (l'aspect ne devrait guère changer).
9. Ecrire un opérateur d'addition `+` prenant deux cordes, vérifiant qu'elles ont même longueur, et renvoyant une nouvelle corde dont les y_i sont la somme des y_i des cordes.
10. Créez une corde pincée en 0.8 de valeur -0.1 , ajoutez-la à la corde précédente et affichez la corde résultante. Attendre le clic avant de l'effacer.
11. Créez une fonction `vibre_corde` prenant les cordes aux instants $t - 1$ et $t - 2$ en paramètres et appliquant une itération de la vibration :
 - Les nouveaux y_i sont obtenus par la formule

$$y_i(t) = 2y_i - y_i(t - 2) + 0.5(y_{i+1} - 2y_i + y_{i-1})$$

- (les y_i sans temps sont à $t - 1$).
 - Les y_0 et y_n restent à leur valeur à $t - 1$.
 - Avant la fin de la fonction, le résultat est écrit dans $t - 1$ et l'ancien $t - 1$ en $t - 2$ de façon à préparer la prochaine itération. Soyez particulièrement attentifs aux fuites mémoire potentielles dans cette fonction.
12. On applique à l'extrémité droite de la corde précédente une impulsion sinusoïdale $0.2 \sin \frac{i\pi}{100}$ (i étant un indice de boucle représentant le temps). Pour cela, on fait vibrer la corde à chaque itération, on impose la valeur à droite et on dessine la corde. Faire l'animation de cette vibration.

Important : Quand vous avez terminé, créez une archive du projet à votre nom en ZIP, RAR ou TGZ (pas 7z). N'incluez que les fichiers source, pas les fichiers binaires créés par CMake. Ne vous déconnectez pas avant que le surveillant ne soit passé vous voir pour copier cette archive sur clé USB.