TP13 – Équation des ondes et FFT

Le problème

Étant données deux fonctions v et w définies sur $\mathbb{R}$, régulières et à support compact, et une fonction f définie sur $\mathbb{R} \times \mathbb{R}$ régulière et à support compact, on recherche $u(t, x)$ solution régulière de

$$\begin{cases} \partial_{tt} u(t,x) - c^2 \partial_{xx} u(t,x) &= f(t,x) \quad \text{pour } x \in \mathbb{R},\ t > 0, \\ u(0,w) = w(x) \quad \text{et} \quad \partial_t u(0,x) &= v(x) \quad \text{pour } x \in \mathbb{R}. \end{cases} \tag{P}$$

Le paramètre réel $c > 0$ est la vitesse des ondes dans le milieu considéré, supposée dans ce travail constante (en temps) et uniforme (en espace), égale à $c = 1$.

Stratégie

On introduit un paramètre $L > 0$ suffisamment grand de sorte que le support en espace des données w, v et f soit inclus dans l'intervalle $[-L, L]$. Pendant un certain intervalle de temps $[0, T]$, la solution du problème (P) coïncide alors avec celle du même problème posé sur $[-L, L]$ avec des conditions de bord périodiques. Nous allons traiter ce problème par une méthode spectrale et l'utilisation de la FFT, en particulier à l'aide des fonctions intégrées de `scipy` que nous allons dans un premier temps expérimenter.

Fourier discret et FFT en Python

◦ Parcourir la documentation pour les fonctions suivantes :
`from scipy.fft import fft, ifft, fftshift, ifftshift`

◦ Implémenter le code suivant :

```
L = 10
N = 100
X = -L+2*L/N*np.arange(0,N)
U = 1+np.cos(10*pi*X/L)
Uhat = fft(U)
Freq = -N/2+np.arange(N)
UhatC = fftshift(Uhat)
Unew = ifft(Uhat)
```

◦ Comparer graphiquement et numériquement les valeurs des variables `U` et `Unew`.

◦ Représenter sur un premier graphique la partie réelle des composantes de `Uhat` et sur un second graphique la partie réelle des composantes de `UhatC` en fonction de celles du vecteur `Freq`.

Retour à l'équation des ondes

◦ On suppose que $f \equiv 0$ et que la solution du problème (P) posé sur $[-L, L]$-périodique se décompose à chaque instant sous la forme d'une somme trigonométrique finie

$$u(t,x) = \sum_{k=-N/2}^{N/2-1} c_k(t) e^{ik\pi x/L},$$

pour un certain entier pair non-nul N. Déterminer puis résoudre à la main une EDO vérifiée par chacune des fonctions c_k.

○ Toujours dans le cas un second membre nul $f \equiv 0$, utiliser la FFT pour résoudre le problème (P) à travers les EDO précédemment identifiées. On considérera par exemple les données suivantes : $L = 10$, $c = 1$, $N = 50$, $T = 1$ et

$$w(x) = \exp(-50(\pi(x-5)/10)^2), \quad v(x) = \exp(-50(\pi(x+5)/10)^2)$$

L'approximation de la solution $u(t,x)$ pour $t \in [0,T]$ et $x \in [-L, L[$ (sur une grille discrète appropriée) sera stockée dans un `numpy.array 2D UXT` et pourra être représentée en espace/temps par la commande suivante :

```
P = plt.imshow(UXT.T,extent=[-L,L,0,T],aspect='auto',origin='lower')
plt.colorbar(P)
plt.show()
```

○ Généraliser la méthode au cas d'un terme source non-homogène.