Projet 2

Notation. On rappelle que votre note de TP correspond à 30% de votre note de UM4MA306 et que cette note de TP est incompressible, c'est à dire sans rattrapage.

Vous pouvez travailler sur le projet individuellement ou par binôme pendant les séances de TP restantes ou bien depuis chez vous. Un rapport et un code, qui doivent être clairs, lisibles et bien organisés, seront à soumettre. La note finale sera décidé lors d'une soutenance orale *individuelle*. En particulier, la note sera déterminée en fonction de votre aisance à utiliser le code, la pertinence de vos explications et réponses aux questions, ainsi que votre recul sur le lien entre résultats théoriques et observations numériques.

Vous êtes bien sûr autorisé à discuter du projet entre vous et avec l'équipe enseignante de l'UE mais toutes similitudes trop importantes pour être considérées comme le fruit d'un heureux hasard, et ceci que ce soit au niveau du code et/ou du rapport, seront considérées comme de la triche et donc sanctionnées sévèrement au niveau de la note.

Le code. Le code doit être envoyé sous la forme d'un ou plusieurs fichiers et doit être développé en Python (extention `.py` ou `.ipynb`). Votre code doit être agréable à lire (noms de variables explicites, indentation, un minimum de commentaires,...). Pour éviter tout problème d'encodage, ne pas utiliser d'accent dans vos commentaires.

Le rapport. Le rapport doit être retourné en format `.pdf`. Dans le cas d'un rapport pdf, de préférence utiliser LaTeX mais vous pouvez également envoyer des notes manuscrites scannées (soigneuses).

Date limite. Le nom de tous vos fichiers doit débuter par `NOM prenom`. Vos fichiers, code et rapport, seront à déposer sur la page Moodle de UM4MA306 **avant le 30/12/2025.**

Soutenance. La soutenance orale aura lieu la semaine des examens. Un planning sera disponible sur Moodle en temps voulu. Cette soutenance consistera à présenter et commenter votre travail. Vous disposerez de 10 minutes pour ce faire, puis s'en suivra une séance de questions. Il vous est fortement conseillé de vous entraîner à cet exercice avant votre passage.

Tout retard, ou non respect des consignes décrites ici, sera sanctionné dans votre note de TP.

Le sujet comporte des parties indépendantes. L'important n'est pas de le finir, l'évaluation portera plus sur la qualité du travail rendu que sa quantité. Pour toute question ou commentaire sur ce projet, il ne faut pas hésiter à prendre contact via Moodle ou par mail :

ani.miraci@sorbonne-universite.fr

Objectif

L'objectif de ce projet est d'implémenter une équation d'onde en 1D en utilisant des différences finies et de considérer des conditions CFL appropriées pour garantir la stabilité. Puis, des taux de convergence expérimentaux seront calculés.

Problème posé

Nous considérons l'équation d'onde :

$$\frac{\partial^2 u}{\partial t^2}(x,t) = c^2 \frac{\partial^2 u}{\partial x^2}(x,t), \quad x \in]-1,1[, \; t \in]0,1[,$$

et des conditions initiales

$$u(x,0) = u_0(x),$$
$$\frac{\partial u}{\partial t}(x,0) = u_1(x)$$

et des conditions aux limites de Dirichlet homogènes

$$u(-1,t) = 0, \quad u(1,t) = 0.$$

Dorénavant, nous considérons la vitesse de l'onde comme $c = 1/2$. De plus, en fixant $a = 0.2$, nous allons considérer le profil d'onde initial suivant :

$$u_0(x) = \begin{cases} (1-(x/a)^2)^2\,(1-2(x/a)^2), & \text{si } |x| \leq a, \\ 0, & \text{sinon,} \end{cases} \quad \text{et} \quad u_1(x) = 0.$$

Tâches à réaliser

i. Solution analytique

(1) Justifier que la solution analytique de l'EDP coïncide avec la solution donnée par la formule de d'Alembert pour $x \in [-1,1], \; t \in [0,1]$ et décrite par

$$u(x,t) = \frac{1}{2}\left[u_0(x-ct) + u_0(x+ct)\right] + \frac{1}{2c}\int_{x-ct}^{x+ct} u_1(s)ds.$$

Définir une fonction Python pour l'évaluer la solution exacte en fonction de x et t.

(2) Tracer la solution à différents moments, par exemple, $t = 0, 0.25, 0.5, 1$ pour visualiser la propagation de l'onde.

(Optionnel) Pour visualiser la propagation de l'onde sous forme d'animation, vous pouvez utiliser le code modèle `animation-template.ipynb` et le modifier pour inclure vos résultats.

ii. Discrétisation

(3) Écrire une fonction `wave1d` prenant en entrée N_x et N_t. Utiliser N_x pour définir la grille spatiale en divisant $[-1,1]$ en N_x points avec un espacement de Δx, donc $x_i = -1+i\Delta x$, $i = 0,\ldots,N_x-1$. De même, définir la grille temporelle avec N_t points et un espacement de Δt, donc $t^n = n\Delta t$, $n = 0,\ldots,N_t-1$. Définir U qui stockera la solution numérique à tous les points de la grille (temps et espace) et sera la sortie de la fonction.

(4) Dans cette fonction, calculer $U_i^n \approx u(x_i,t^n)$ via le schéma explicite

$$\frac{U_i^{n+1} - 2U_i^n + U_i^{n-1}}{\Delta t^2} = c^2\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2}.$$

Après réécriture, cela donne

$$U_i^{n+1} = 2U_i^n - U_i^{n-1} + c^2\frac{\Delta t^2}{\Delta x^2}(U_{i+1}^n - 2U_i^n + U_{i-1}^n).$$

Afin d'implémenter la méthode, suivez ces étapes :

- **Initialiser la solution :**

$$U_i^0 = u_0(x_i), \qquad i = 0,1,\ldots,N_x-1$$

$$U_i^1 = U_i^0 \quad (\text{car } \frac{\partial u}{\partial t}(x,0) = u_1(x) = 0)$$

- **Appliquer les conditions aux limites à tous les instants :**

$$U_0^n = 0, \qquad U_{N_x-1}^n = 0, \quad n = 0,1,\ldots,N_t-1$$

- **Marche en temps (récursion) pour les points intérieurs :**
  Pour $n = 1$ à $N_t - 2$:

$$\begin{cases} U_1^{n+1} = 2U_1^n - U_1^{n-1} + c^2\frac{\Delta t^2}{\Delta x^2}(U_0^n - 2U_1^n + U_2^n) \\ U_2^{n+1} = 2U_2^n - U_2^{n-1} + c^2\frac{\Delta t^2}{\Delta x^2}(U_1^n - 2U_2^n + U_3^n) \\ \vdots \\ U_{N_x-2}^{n+1} = 2U_{N_x-2}^n - U_{N_x-2}^{n-1} + c^2\frac{\Delta t^2}{\Delta x^2}(U_{N_x-3}^n - 2U_{N_x-2}^n + U_{N_x-1}^n) \end{cases}$$

(5) La fonction doit renvoyer la solution calculée U après avoir terminé tous les pas de temps.

(Optionnel) Tout comme pour la solution analytique, vous pouvez réutiliser votre code d'animation pour visualiser la propagation de l'onde dans le temps.

iii. Stabilité et CFL dans la pratique

(6) Analyser le schéma donné et trouver la condition CFL sous laquelle la méthode est stable.

iv. Convergence et analyse d'erreur

Pour étudier la convergence du schéma de différences finies dans l'espace et le temps, nous utilisons la norme L^2 discrète de l'erreur au temps t^n

$$\|e^n\|_2 = \Big(\sum_{i=0}^{N_x-1} \big(U_i^n - u(x_i, t^n)\big)^2 \, \Delta x\Big)^{1/2}.$$

(7) Exécuter des simulations pour différentes valeurs de N_x (par exemple, $N_x = 51, 101, 201, 401$) tout en maintenant le rapport $\Delta t/\Delta x$ constant pour satisfaire la condition CFL. Pour chaque simulation, calculer l'erreur au temps final T et tracer l'erreur en fonction de Δx sur une échelle log-log. Déterminer l'ordre expérimental de convergence dans l'espace.

(8) De même, pour étudier la convergence dans le temps, fixer N_x et faire varier N_t (par exemple, $N_t = 101, 201, 401, 801$). Calculer l'erreur au temps final T pour chaque simulation et tracer l'erreur en fonction de Δt sur une échelle log-log. Déterminer l'ordre expérimental de convergence dans le temps.