

Equation hyperbolique scalaire (non)linéaire et raffinement de maillage

On considère l'équation hyperbolique scalaire

$$\begin{cases} \partial_t u(t, x) + \partial_x f(u(t, x)) = 0, \\ u(0, x) = u_0(x), \end{cases} \quad (1)$$

$f \in C^2(\mathbb{R})$, pour $x \in [-1, 1]$ avec des conditions aux bords périodiques. Même si la donnée initiale est régulière, le modèle génère des discontinuités en temps fini. Le but du projet est de proposer une technique de raffinement de maillage pour observer plus précisément les zones de discontinuités.

Tests de validation

Pour tester le code, on considère :

- un flux linéaire $f(u) = au$, $a \in \mathbb{R}^*$ et une donnée initiale régulière (par exemple sinusoïdale),
- un flux de Burgers $f(u) = u^2/2$ avec une donnée initiale de type *créneau*.

Approximation de la solution

On utilise une méthode volumes finis et un schéma numérique de Lax-Friedrichs et/ou Godunov. Les résultats numériques obtenus doivent être comparés à la solution exacte du problème (au moins en certains temps).

Gestion du maillage

On utilise une liste chaînée pour stocker les mailles du maillage initial. Une maille du maillage est définie **au moins** par ses bornes, la valeur de u_i^n dans la maille, et un pointeur sur la maille suivante. On impose d'utiliser la syntaxe suivante (vous pouvez ajouter des méthodes et des membres mais pas en supprimer) :

```
// On crée une classe cellule.
class cellule
{
    friend class maillage; //les deux classes sont amies

public :
    // Membres
    double a,b;           //bornes de la cellule
    double uin;           // valeur de  $u_i^n$  dans la cellule
    cellule * next;       // L'adresse de la suivante.

    // CONSTRUCTEUR
    cellule(); // par défaut
    cellule(const cellule &c); //par copie
};
```

```
// La classe qui g rera le maillage.
class maillage
{
private :
    cellule * tete;    // Pointeur sur la premiere cellule.
    int nb_cellules;   // Nombre de cellules du maillage.

public :
    // CONSTRUCTEURS
    maillage(); // par default
    maillage(const maillage &m); //par recopie

    // M THODES DE MODIFICATION
    // Ajout a une position
    int add_position(double valeur, int position);

    // Retrait a une position
    void remove_position(int position);

    // METHDOES DE CONSULTATION
    int consult_position(int position, int & valeur) const;

    // SURCHARGE D'OPERATEUR
    // Donner la valeur de u_i^n dans maille d'indice=position
    double operator[] (int position);
    friend ostream & operator<<(ostream &, const maillage &);
};
```

Raffinement de maillage

Le projet consiste   expliquer et mettre en  uvre une technique de raffinement de maillage pour d tecter plus pr cis ment les discontinuit s du probl me hyperbolique consid r . Il n'est pas obligatoire de d tailler un processus de d raffinement.

Contraintes et attendus du projet

- Le projet est   rendre le **vendredi 29 novembre 18h**. Il faudra d poser sur MADOC, dans le d p t pr vu   cet effet :
 - le code stock  dans un r pertoire zipp , contenant **uniquement** les fichiers sources (.cpp, .h ou .hpp et le makefile) ;
 - un rapport (r dig  en LaTeX, au format pdf) d'environ 5 pages, illustrations num riques comprises.
- Le code doit  tre impl ment  en C++. Tout code non comment  ou ne compilant entra nera une note comprise entre 0 et 5 (sur 20).
- Le rapport doit illustrer vos comp tences. Il est demand  d'expliquer rigoureusement votre algorithme de raffinement de maillage, d'expliquer votre m thode num rique volumes finis, de donner et commenter vos r sultats num riques. Un soin particulier doit  tre donn    votre rapport.
- Vous  tes libre dans l'impl mentation des classes, fonctions et algorithmes, hormis dans la gestion des listes cha n es qui doivent respecter la structure pr sent e plus haut. En particulier il faudra inclure la surcharge des op rateurs [] et <<.