

Introduction rapide à Python

Un cours *très* accéléré sur les bases de Python, couvrant largement

Bibliothèques (NumPy et Matplotlib, en particulier)

Variables

Espace

Découpage et affectation de tableaux

Step 1: Convection linéaire 1-D

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad (1)$$

Math

Cette section présente au lecteur les “parties” de l’EDP et comment les discrétiser.

Introduire l’idée d’une grille

Développer l’équation en utilisant la définition d’une dérivée

Discrétiser en petits morceaux

Réorganiser pour résoudre en u_i^{n+1}

Conditions initiales et aux limites

Python

Importation de bibliothèques

Affectation de variables

Tracé 2D basique

Boucles for (simples)

Vidéos YouTube sur ordre de convergence, erreur de troncature, etc...

Step 2: 1-D Nonlinear Convection

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0 \quad (2)$$

Math

Introduire l’EDP non-linéaire

Développer l’équation en utilisant la définition de la dérivée

Discrétiser

Résoudre en u_i^{n+1}

CFL Condition

Un court détour sur la condition CFL, l’ordre de convergence et comment faire exploser les choses.

Math

Le nombre de Courant

Explication du comportement d'explosion lorsque l'onde parcourt une distance $> dx$ pendant un temps dt

Python

Introduction rapide à la définition d'une fonction pour utiliser du code à plusieurs reprises

Step 3: 1-D Diffusion

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \quad (3)$$

Math

Introduire l'équation de diffusion

Discretiser la dérivée du 2e ordre en utilisant le développement en série de Taylor

Discretiser la dérivée temporelle en utilisant la définition de dérivé

Python

Rien de nouveau, toujours pas (encore) de fonctions utilisées

Étape 4 : Équation de Burgers en 1-D

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \quad (4)$$

Math

Introduire l'équation de Burgers

Notez qu'il s'agit d'une combinaison de diffusion et de convection non linéaire

Introduire différents C.I. et C.L. pour un comportement périodique

Introduce Burgers' Equation

Note that it is combination of diffusion and non-linear convection

Introduce different I.C. and B.C. for periodic behavior

par exemple. Qu'est-ce que u_{i+1}^n signifie à la fin du transparent ?

Python

Présentation de Sympy

Jolie impression

Résolution symbolique des dérivées

Utilisation de Lambdify pour rendre les solutions "accessibles" à Numpy

Matplotlib

Tracer plusieurs lignes par figure

Définition des styles de ligne

Légendes

Opérations sur les tableaux

Une autre brève introduction à la gestion des calculs avec des opérations (spécifiques) de tableau au lieu d'itérer sur l'ensemble du tableau.

Python

Opérations sur les tableaux, découpage et copie

Remarque sur l'utilisation de `%timeit` fonction magique pour comparer les performances

Étape 5 : Convection linéaire en 2D

Math

Introduction to 2D grid

Extension of current discretization rules into i, j flatland

Discretize 2D equation and solve for unknown

Introduction à la grille 2D

Extension des règles de discrétisation actuelles en terrain plat 2D

Discrétiser l'équation 2D et résoudre

Python

meshgrid (grille de maillage)

Axes3D

tracés surf et wireframe

Démonstration que les résultats de boucles for imbriquées et les résultats des opérations (spécifiques) sur les tableaux sont les mêmes

Étape 6 : Convection non linéaire en 2D

Math

Introduction des EDP couplées

Discrétisation de deux équations

Résolution pour trouver $u_{i,j}^{n+1}$ et $v_{i,j}^{n+1}$

0.1 Étape 7 : Diffusion en 2D

Math

- Introduction à l'équation de diffusion en 2D
- Discrétisation d'équation, etc

0.1.1 Python

Rien de nouveau, bien que les fonctions soient utilisées pour afficher les résultats (devrait probablement passer à `jsanim`)

Étape 8 : Equation de Burgers en 2D