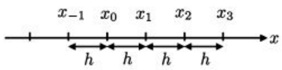
TP2 : Dérivation numérique

F. Saïd

1 Notion de maillage

Un maillage (ou grille numérique) peut être défini comme un ensemble de points régulièrement espacés sur un intervalle donné. L'espacement ou pas du maillage est la distance entre les points adjacents de la grille. Dans la suite, étant donné un maillage, on désignera par x_j le j-ième point de la grille et par h l'espacement entre x_{j-1} et x_j.



2 Dérivées d'ordre 1

— La dérivée d'une fonction f en un point a est définie par

$$f'(a) = \lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

La dérivée en $x = a$ est la pente de la tangente à la courbe de f au point du plan d'abscisse a. Pour estimer cette pente, on utilise les valeurs de la fonction f dans un voisinage du point $x = a$. Il existe différentes formules pour cela ; on les appelle formules par **différences finies**.

— Nous présentons ci-dessous 3 formules de différences finies qui utilisent les valeurs de f en deux points ; on parle de "formules ou schémas à deux points".

1. Le schéma "**avant**" (forward en anglais) estime la pente en x_j en utilisant le segment de droite qui relie $(x_j, f(x_j))$ et $(x_{j+1}, f(x_{j+1}))$:

$$f'(x_j) \approx \frac{f(x_{j+1}) - f(x_j)}{x_{j+1} - x_j} = \frac{f(x_{j+1}) - f(x_j)}{h}$$

2. Le schéma "**arrière**" (backward en anglais) estime la pente en x_j en utilisant le segment de droite qui relie $(x_{j-1}, f(x_{j-1}))$ et $(x_j, f(x_j))$:

$$f'(x_j) \approx \frac{f(x_j) - f(x_{j-1})}{x_j - x_{j-1}} = \frac{f(x_j) - f(x_{j-1})}{h}$$

3. Le schéma "**centré**" (central en anglais) estime la pente en x_j en utilisant le segment de droite qui relie $(x_{j-1}, f(x_{j-1}))$ et $(x_{j+1}, f(x_{j+1}))$:

$$f'(x_j) \approx \frac{f(x_{j+1}) - f(x_{j-1})}{x_{j+1} - x_{j-1}} = \frac{f(x_{j+1}) - f(x_{j-1})}{2h}$$

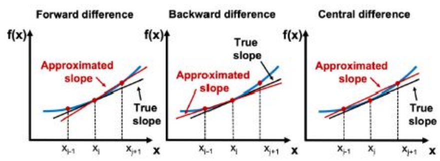


FIGURE 1 – Schémas de différences finies à 2 points

Exercice 1 :

Python possède une commande qui peut être utilisée pour calculer directement les différences finies : pour un vecteur u, la commande $d = np.diff(u)$ produit un tableau d dans lequel les entrées sont les différences des éléments adjacents du tableau initial u . En d'autres termes, $d(i) = u(i+1) - u(i)$.

Utiliser le script qui suit sur l'intervalle $[0, 2\pi]$:

1. pour représenter la dérivée exacte de la fonction $f : x \mapsto \cos(x)$ et son approximation numérique avec un schéma de différences finies avant, pour un pas $h = 0.1$
2. pour calculer l'erreur absolue maximale commise

```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use("seaborn-poster")
%matplotlib inline

# step size
h = 0.1
# define grid
x = np.arange(0, 2*np.pi, h)
# compute function
y = np.cos(x)

# compute vector of forward differences
forward_diff = np.diff(y)/h
# compute corresponding grid
x_diff = x[:-1:]
# compute exact solution
exact_solution = -np.sin(x_diff)

# Plot solution
plt.figure(figsize = (12, 8))
plt.plot(x_diff, forward_diff, "-", \
         label = "Finite difference approximation")
plt.plot(x_diff, exact_solution, label = "Exact solution")
plt.legend()
plt.show()

# Compute max error between
# numerical derivative and exact solution
max_error = max(abs(exact_solution - forward_diff))
print(max_error)
```

FIGURE 2 – Code du schéma avant

Exercice 2 : S'inspirer de l'exercice précédent pour écrire des scripts permettant de mettre en oeuvre le schéma de différences finies arrière et le schéma de différences finies centré.

Exercice 3 :

Les schémas de différences finies produisent une erreur d'approximation numérique qui diminue avec le pas de discrétisation h. Ainsi, les schémas avant et arrière produisent une erreur de l'ordre de $\mathcal{O}(h)$ tandis que que le schéma centré produit une erreur de l'ordre de $\mathcal{O}(h^2)$. Nous proposons d'illustrer ces résultats dans cet exercice.

1. Utiliser le code qui suit pour calculer la dérivée numérique de $f(x) = \cos(x)$ à l'aide de la formule de différences avant pour une taille de pas h décroissante. Ce code permet de tracer l'erreur maximale entre la dérivée approchée et la dérivée réelle en fonction de h. On pourra constater que la pente de la droite générée (échelle log-log) est de 1 ; par conséquent, l'erreur est proportionnelle à h, ce qui signifie que, comme attendu, la formule de différences finies avant est $\mathcal{O}(h)$.
2. Faire de même avec le schéma de différences finies centré et constater que l'erreur est de l'ordre de $\mathcal{O}(h^2)$.

```
# define step size
h = 1
# define number of iterations to perform
iterations = 20
# list to store our step sizes
step_size = []
# list to store max error for each step size
max_error = []

for i in range(iterations):
    # halve the step size
    h /= 2
    # store this step size
    step_size.append(h)
    # compute new grid
    x = np.arange(0, 2 * np.pi, h)
    # compute function value at grid
    y = np.cos(x)
    # compute vector of forward differences
    forward_diff = np.diff(y)/h
    # compute corresponding grid
    x_diff = x[:-1]
    # compute exact solution
    exact_solution = -np.sin(x_diff)

    # Compute max error between
    # numerical derivative and exact solution
    max_error.append(max(abs(exact_solution - forward_diff)))

# produce log-log plot of max error versus step size
plt.figure(figsize = (12, 8))
plt.loglog(step_size, max_error, "v")
plt.show()
```

FIGURE 3 – Le schéma avant est d'ordre 1

Exercice 4 :

Nous nous intéressons maintenant au schéma aux différences finies suivant :

$$\boxed{f'(x_j) \approx \frac{f(x_{j-2}) - 8f(x_{j-1}) + 8f(x_{j+1}) - f(x_{j+2})}{12h}}$$

Cette formule donne une approximation de la dérivée en x_j d'ordre h^4 ; elle offre donc une meilleure précision que la formule des différences centrées mais elle requiert deux fois plus de calculs.

Pour comprendre d'où vient cette formule, considérons les développements limités qui suivent au point x_j :

$$\begin{aligned}
f(x_{j-2}) &= f(x_j) - 2hf(x_j) + \frac{4h^2 f''(x_j)}{2} - \frac{8h^3 f^{(3)}(x_j)}{6} + \frac{16h^4 f^{(4)}(x_j)}{24} - \frac{32h^5 f^{(5)}(x_j)}{120} + \cdots \\
f(x_{j-1}) &= f(x_j) - hf(x_j) + \frac{h^2 f''(x_j)}{2} - \frac{h^3 f^{(3)}(x_j)}{6} + \frac{h^4 f^{(4)}(x_j)}{24} - \frac{h^5 f^{(5)}(x_j)}{120} + \cdots \\
f(x_{j+1}) &= f(x_j) + hf(x_j) + \frac{h^2 f''(x_j)}{2} + \frac{h^3 f^{(3)}(x_j)}{6} + \frac{h^4 f^{(4)}(x_j)}{24} + \frac{h^5 f^{(5)}(x_j)}{120} + \cdots \\
f(x_{j+2}) &= f(x_j) + 2hf(x_j) + \frac{4h^2 f''(x_j)}{2} + \frac{8h^3 f^{(3)}(x_j)}{6} + \frac{16h^4 f^{(4)}(x_j)}{24} + \frac{32h^5 f^{(5)}(x_j)}{120} + \cdots
\end{aligned}$$

On annule les termes en h^2, h^3 et h^4 en effectuant :

$$f(x_{j-2}) - 8f(x_{j-1}) + 8f(x_{j+1}) - f(x_{j+2}) = 12f'(x_j) + \mathcal{O}(h^4)$$

soit

$$\boxed{f'(x_j) = \frac{f(x_{j-2}) - 8f(x_{j-1}) + 8f(x_{j+1}) - f(x_{j+2})}{12h} + \mathcal{O}(h^4)}$$

Écrire un script permettant de comparer graphiquement ce schéma et le schéma centré avec la fonction $f : x \mapsto \cos(x)$ pour un pas $h = 0.1$.

3 Dérivées d'ordre supérieur

Exercice 1 :

1. Utiliser les formules des développements limités pour établir les schémas de différences finies suivants :

$$\boxed{\begin{aligned}
f''(x_j) &= \frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1})}{h^2} + \mathcal{O}(h^2) \\
f''(x_j) &= \frac{-f(x_{j+3}) + 4f(x_{j+2}) - 5f(x_{j+1}) + 2f(x_j)}{h^2} + \mathcal{O}(h^2) \\
f^{(3)}(x_j) &= \frac{f(x_{j+3}) - 3f(x_{j+2}) + 3f(x_{j+1}) - f(x_j)}{\cancel{h^2}\ h^3} + \mathcal{O}(h)
\end{aligned}}$$

2. Écrire des scripts mettant en oeuvre ces schémas pour la fonction $f : x \mapsto \cos x$ sur l'intervalle $[0, 2\pi]$, avec un pas $h = 0.1$.