Master Calcul Haute Performance et Simulation

Rapport

Projet Validation numérique

Réalisé par :

Dekkal Dyhia

Année universitaire : 2022/2023

1 Tp1

1.1 Différences centrées d'ordre 2

— Rappeler l'approximation d'une dérivée seconde par le schéma dit "à 3 points" (ou aussi appelé différences centrées d'ordre 2).

1.2 Résolution d'une EDO par différences centrées

L'EDO $(E) : y''(x) - y(x) = 0$ et $y(0) = 1, y(1) = e$ admet $y(x) = \exp(x)$ comme solution.

1. Préciser l'expression d'une discrétisation avec n points intérieurs à l'intervalle $I = [0,1]$. On définit un maillage uniforme de pas $h = 1/(1+N)$

   par les points

   $$h = 1/(1+N)$$

   par les points

   $$x_i = \{x_0 + ih$$

   ,

   La méthode des différences finies est une approche classique pour approximer une solution continue sur une partition uniforme de l'intervalle : $0I = [0,1]$ Soit N un entier et $h = 1/(1+N)$ le pas d'espace uniforme , la partition est définie par

   $$\{x_i = x_0 + ih, i = 1......N+1\}$$

   Une discrétisation par la méthode des différences finies consiste alors à approcher une dérivée continue par une approximation discrète.

   Nous allons exprimer l'erreur de l'approximation liée au schéma de discrétisation (2)sur un noeud x_i du maillage .

   Nous avons le développement en série de Taylor :

$$\begin{cases} U(x_{i+1}) = U(x_i) + hU'(x_i) + \frac{h^2}{2}\mathrm{U"(x_i)} + \frac{h^3}{6}\mathrm{U}^{(3)}(x_i) + \frac{h^4}{24}\mathrm{U}^{(4)}(x_i) + O(h^5) \\ U(x_{i-1}) = U(x_i) - hU'(x_i) + \frac{h^2}{2}\mathrm{U"(x_i)} - \frac{h^3}{6}\mathrm{U}^{(3)}(x_i) + \frac{h^4}{24}\mathrm{U}^{(4)}(x_i) + O(h^5) \end{cases}$$

En additionnant les développements précédents, on obtient Après simplification, on obtient :

$-U(x_{i+1}) + 2ux_i - Ux_{i-1} = -h^2u"x_i - \frac{h^4}{24}\mathrm{U}^{(4)} + O(h^4)$

$$\underbrace{U''(x_i) = \frac{U(x_{i-1}) - 2U(x_i) + U(x_{i+1})}{h^2}}_{\textit{Shéma DF centré en } x_i \textit{ à 3 points}(x_{i-1},x_i,x_{i+1})} \quad - \quad \underbrace{(\frac{h^2}{12}\mathrm{U}^{(4)+O(h^4)})}_{\textit{erreur de descritisation}}$$

2. Expliciter le schémas aux différences centrées appliqué à l'EDO (E) comme un système linéaire $Ay = b$.

   Le problème nous conduit au système approché suivant :

$$\begin{cases} \frac{y(i-1)-2y(i)+y(i+1)}{h^2} + y_i = 0, 1 \leq i \leq N \\ U_0 = \alpha \\ U(N+1) = \beta \end{cases}$$

   En utilisant les conditions aux limites $y_0 = \alpha, y(N+1) = \beta$

$$\begin{cases} \frac{1}{h^2} - \frac{2y(1)-y(2)}{h^2} - y_1 = 0, i = 0 \\ \frac{y(i-1)-2y(i)+y(i+1)}{h^2} - y_i = 0, 2 \leq i \leq N-1 \\ \frac{y(N-1)+2y(N)}{h^2} + \frac{e}{h^2} - y_N = 0, i = N \end{cases}$$

3. Préciser la forme de la matrice A

   Ces N équations on les écrit sous forme matricielle : $A_h y_h = b_h$

$$A_h = \frac{1}{h^2}\begin{pmatrix} -(2+h^2) & 1 & 0 & \dots & \dots & 0 \\ 1 & -(2+h^2) & 1 & 0 & \dots & \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & & & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 1 \\ 0 & 0 & \dots & 0 & 1 & -(2+h^2) \end{pmatrix}$$

4. Préciser le second membre b en prenant en compte les conditions aux limites de (E)

$$b_h = \begin{pmatrix} -\frac{\alpha}{h^2} \\ 0 \\ : \\ : \\ : \\ -\frac{\beta}{h^2} \end{pmatrix}$$

comme $\alpha = 1 et \beta = e$

5. Préciser le sens du vecteur solution y.

$$y_h = \begin{pmatrix} y_1 \\ y_2 \\ : \\ : \\ : \\ y_n \end{pmatrix}$$

6. Coder cette résolution dans le langage et avec les bibliothèques de votre choix :

Il convient d'écrire le système sous forme :

$$\frac{1}{h^2}[-2I - \begin{pmatrix} h^2 & -1 & 0 & \dots & \dots & 0 \\ -1 & h^2 & -1 & 0 & \dots & \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & & & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -1 \\ 0 & 0 & \dots & 0 & -1 & h^2 \end{pmatrix}] \begin{pmatrix} y_1 \\ y_2 \\ : \\ : \\ : \\ y_n \end{pmatrix} = \begin{pmatrix} -\frac{\alpha}{h^2} \\ 0 \\ : \\ : \\ : \\ -\frac{\beta}{h^2} \end{pmatrix}$$

1.3 Un premier essai pour 20 points de discrétisation

Pour l'EDO (E) et le schéma précédent :

effectuer une première résolution de validation pour 20 points intérieurs

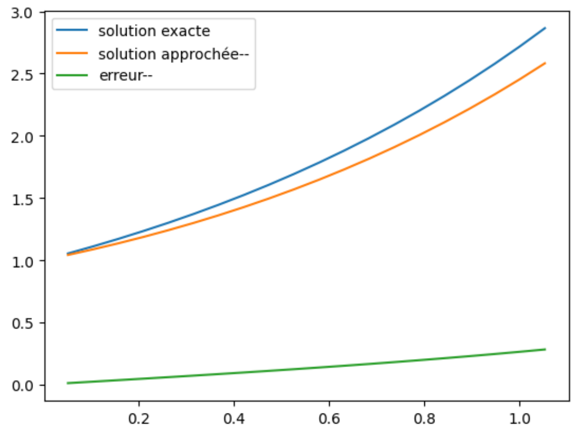
— Définir et tracer l'erreur de cette résolution sur I

Pour calculer l'erreur on calcule la solution approchée ainsi que la solution exacte et ceci sur les points de descritisations définit sur l'intervalle $[0, 1]$

en suite on fera la soustraction entre la solution approchée et la solution exacte en valeur absolue

$$err = |solutionexacte - solutionapprochée|$$

On trace le graphe pour voir l'évolution de l'erreur avec les 20 points :



Sur le graphe on voit que l'erreur augmente lorsqu'on avance avec le pas et la solution approchée diverge de la solution exacte

1.4 Variation du pas de discrétisation et observation de l'erreur

On va maintenant faire varier le pas de discrétisation, par exemple en le diminuant d'un facteur 10 àpartir de 10−1, ie. $10-1, 10-2, 10-3, \ldots$.

— Rappeler ce que l'analyse numérique prédit sur les différentes solutions ainsi obtenues :

Pour améliorer l'approximation (c'est à dire dimininuer l'erreur), il faut faire décroitre le pas h, Mais par contre cela engendre une augmentation de la taille du système à résoudre .

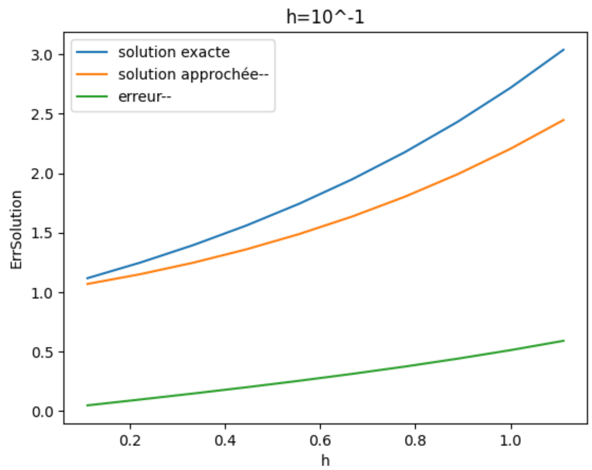
— Observer l'évolution de l'erreur de cette résolution quand le pas de discrétisation varie.

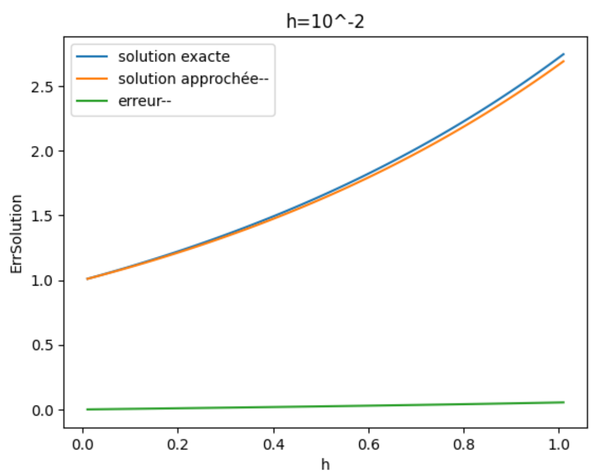
On voit que quand le N est grand la solution approchée converge vers la solution exacte

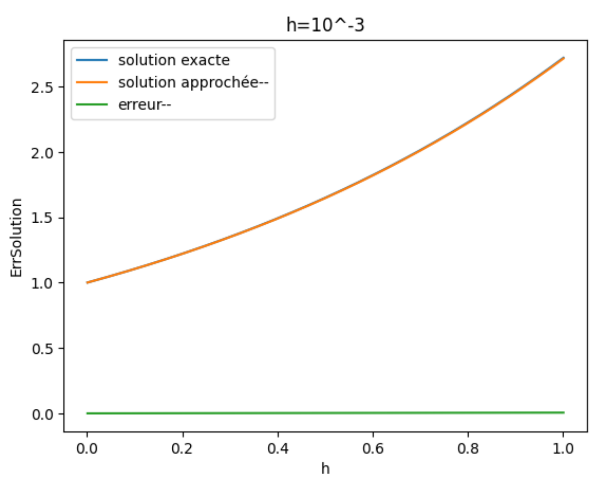
L'erreur maximale dans le cas 10^{-1} est de : 0.5908875228082286

L'erreur maximale dans ce cas 10^{-2}est de : 0.054748213813891056

L'erreur maximale dans ce cas 10^{-3} est de : 0.0054403074238087434







— Vous adapterez la plage de variation du pas aux capacités de calcul de votre machine.

on variant le pas arrivant à $10^{-}5$ ma machine n'est plus capable d'effectuer les calculs

```
MemoryError: Unable to allocate 74.5 GiB for an array with shape (100000, 100000) and data type float64
```

```
Entrée [61]: print("Memory utilised (bytes): ", sys.getsizeof(A))

             Memory utilised (bytes):  800000128
```

Comme nous voyons sur l'image le de nombre bytes pour le stockage de la matrice A pour une taille=10000 au delà de cette taille on ne peut plus stocker.

— Que déduire de ces premières expériences

On déduit que la valeur maximale de l'erreur tends vers zero quand la taille de la matrice est grande et cela engendre un temps de calcul important ce qui augmente le stockage au niveau de la machine par contre ce dernier est limié , d'où la necessité de trouver une autre manière de stocker(stockage optimal) pour pallier au problème de l'incapacité de calcul.

1.5 Traiter des problèmes de taille importante :

— Vous avez du constater qu'il est très vite hors de portée de résoudre cette EDO pour des discrétisations assez fines. Expliquer pourquoi ?

car le stockage des valeurs nulles de la matrice tridiagonale au niveau de la mémoire prend beaucoup d'espace (un stockage important)

— Quelle sont ces caractéristiques ? Quelles sont les conséquences théoriques attendues ? Comment les exploiter en pratique ?

parmi les caractéristiques : Après la discrétisations on se retouve avec un système linéaire où la matrice tridiagonanle (Nombre de zeros est important)(Matrice creuse) les conséquences théoriques attendue la complexité pour la résolution du système pour une telle matrice est de $O(N)$

mais par contre avec une grande taille posera problème pour les caculs Pour expoloiter ça en pratique on s'appuit sur le stockage format CSR ou bien CSC pour un stockage optimal

— Le traitement optimisé :

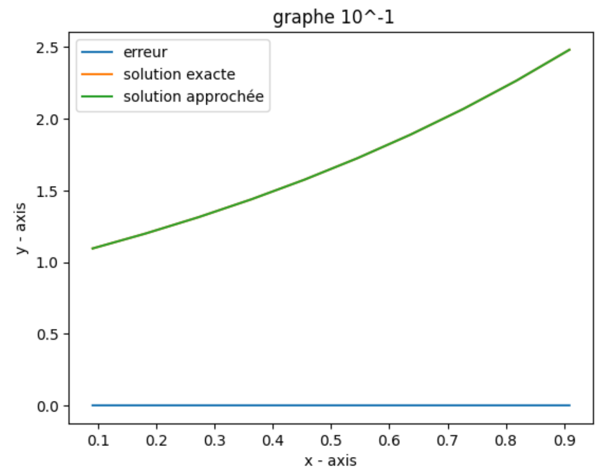
Coder l'algorithme jaccobi d'une manière optimisée :

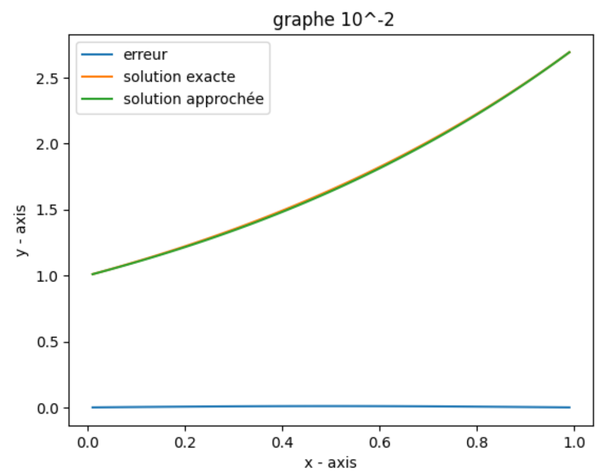
pour le calcul du résidu il suffit de le calculer en exploitant seulement les éléments non nuls et la manière de faire c'est de les stocker dans des variables en suite on itère pour tout les éléments x_i

ainsi le résultat final sera le vecteur x qui est la solution approchée (voir le code)

Pour la partie application :

— Reprendre la résolution de l'EDO (E) pour des pas de discrétisation variant d'un facteur 10 entre 10^{-1} et 10^{-8}.





Pour que la solution apprchée converge vers la solution exacte il faut prendre le nombre d'itérations $it = N * N$

Mais celà prendra aussi du temps mais c'est faisable en terme de stockage mais en terme d'optimistion c'est pas optimisé donc il faut passer au parallélisme pour réduire le nombre d'itérations

Conclusion :

Durant ce tp on constate que :

Les erreurs de descritisations dimininuent et convergent vers 0 lorsque le le N est grand (un nombre de calcul important) ce qui veut dire dans ce cas que la solution approchée converge vers la solution exacte .