

TD/TP 5 de Calcul numérique

Méthodes directes et itératives pour résoudre l'équation de la chaleur 1D

HUANG Yupan

Exercice 1 — Problème modèle et discrétisation

Dans ce projet, on considère l'équation de Poisson unidimensionnelle définie sur un intervalle avec des conditions aux limites de Dirichlet. En introduisant un maillage uniforme de l'intervalle et en utilisant un schéma de différences finies centré d'ordre deux pour approximer la dérivée seconde, le problème continu est discrétisé sous la forme d'un système linéaire.

La matrice obtenue après discrétisation est une matrice tridiagonale, symétrique et strictement à diagonale dominante. Ces propriétés assurent de bonnes caractéristiques numériques et constituent une base théorique solide pour l'application des méthodes directes et itératives.

Exercice 2 — Environnement numérique

Les implémentations numériques sont réalisées en langage C, en s'appuyant sur les bibliothèques BLAS et LAPACK pour les opérations d'algèbre linéaire. La compilation est gérée à l'aide d'un Makefile, et les calculs sont exécutés dans un environnement Docker afin de garantir la cohérence de l'environnement de calcul et la reproductibilité des résultats.

Exercice 3 — Questions théoriques

Question 1 : Quelles sont les propriétés de la matrice obtenue après discrétisation ?

Réponse :

La matrice issue de la discrétisation est une matrice tridiagonale, symétrique, définie positive et strictement à diagonale dominante. Ces propriétés garantissent l'unicité de la solution du système linéaire et assurent la convergence des méthodes itératives classiques.

Question 2 : Quel est le principe de base de la méthode de Jacobi ?

Réponse :

La méthode de Jacobi consiste à décomposer la matrice en une partie diagonale et une partie hors diagonale. À chaque itération, toutes les composantes de la solution sont mises à jour uniquement à partir des valeurs de l'itération précédente. Cette méthode est simple à mettre en œuvre et facilement parallélisable, mais sa vitesse de convergence est généralement faible.

Question 3 : Quelle est la principale différence entre les méthodes de Jacobi et de Gauss–Seidel ?

Réponse :

La méthode de Gauss–Seidel utilise immédiatement les composantes nouvellement mises à jour au cours de l'itération, contrairement à la méthode de Jacobi qui n'utilise que les valeurs de l'itération précédente. Cette utilisation des informations les plus récentes permet en général à la méthode de Gauss–Seidel de converger plus rapidement.

Question 4 : Quelle est la forme itérative de la méthode de Richardson ?

Réponse :

La méthode de Richardson est définie par la relation itérative suivante :

$$u^{(k+1)} = u^{(k)} + \alpha(f - Au^{(k)})$$

où α est un paramètre de relaxation dont le choix influence fortement la vitesse de convergence.

Question 5 : Les méthodes de Jacobi et de Gauss–Seidel peuvent-elles être vues comme des cas particuliers de la méthode de Richardson ?

Réponse :

Oui. Les méthodes de Jacobi et de Gauss–Seidel peuvent être interprétées comme des méthodes de Richardson préconditionnées, où le préconditionneur correspond respectivement à la matrice diagonale et à la matrice triangulaire inférieure.

Question 6 : De quelles propriétés de la matrice dépend la convergence des méthodes itératives ?

Réponse :

La convergence des méthodes itératives dépend principalement du rayon spectral de la matrice d'itération. Pour des matrices symétriques, définies positives et strictement à diagonale dominante, comme celles issues du problème de Poisson unidimensionnel, les méthodes de Jacobi et de Gauss–Seidel sont garanties convergentes.

Question 7 : Pourquoi la méthode de Gauss–Seidel converge-t-elle généralement plus vite que la méthode de Jacobi ?

Réponse :

La méthode de Gauss–Seidel exploite les valeurs les plus récentes calculées au cours de l'itération, ce qui permet de réduire plus efficacement les composantes de l'erreur et d'améliorer les propriétés spectrales de la matrice d'itération.

Question 8 : Pourquoi introduire un préconditionnement en pratique ?

Réponse :

Le préconditionnement permet d'améliorer les propriétés spectrales du système linéaire, ce qui accélère significativement la convergence des méthodes itératives et réduit le coût de calcul, en particulier pour les problèmes de grande taille.

Exercice 4 — Discrétisation par différences finies

On considère le problème de Poisson unidimensionnel sur l'intervalle $[0,1]$, soumis à des conditions aux limites de Dirichlet :

$$-u''(x) = 0, x \in (0,1)$$

$$\{ \quad u(0) = T_0$$

$$u(1) = T_1$$

L'intervalle $[0,1]$ est discrétisé uniformément en $n+2$ points, dont n points intérieurs inconnus. Le pas de discrétisation est donné par :

$$h = \frac{1}{(n+1)}$$

La dérivée seconde est approchée par un schéma aux différences finies centrées d'ordre deux :

$$-u''(x_i) \approx \frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2}$$

On obtient ainsi un système linéaire :

$$A u = b$$

La matrice A est tridiagonale symétrique :

diagonale principale égale à 2, sous- et sur-diagonales égales à -1.

La matrice est stockée au format bande (GB) colonne-major compatible LAPACK.

Exercice 5 — Solution analytique et estimation de l'erreur

La solution analytique est :

$$u(x) = T_0 + x (T_1 - T_0)$$

L'erreur relative avant est définie par :

$$\text{relres} = \frac{\|u_{\text{num}} - u_{\text{exact}}\|_2}{\|u_{\text{exact}}\|_2}$$

Résultats — Exercice 5

```
root@f8b93f9d9019:/app# bin/tpPoisson1D_direct
```

```
----- Poisson 1D -----
```

```
Solution with LAPACK
```

```
The relative forward error is relres = 2.004340e-16
```

```
----- End -----
```

```
root@f8b93f9d9019:/app# bin/tpPoisson1D_direct 2
```

```
----- Poisson 1D -----
```

```
Solution with LAPACK
```

```
The relative forward error is relres = 2.004340e-16
```

```
----- End -----
```

Exercice 6 — Résolution directe par LAPACK

Le système linéaire issu de la discrétisation est résolu à l'aide de la routine dgbmv de la bibliothèque LAPACK, spécifiquement conçue pour les matrices bandes.

Cette méthode effectue :

une factorisation LU de la matrice bande,

suivie de substitutions avant et arrière pour obtenir la solution.

La méthode directe est numériquement stable et fournit une solution de haute précision.

Elle constitue une référence fiable pour l'évaluation des méthodes itératives introduites dans les exercices suivants.

Différentes configurations de stockage ont été testées afin de vérifier la cohérence de l'implémentation.

Résultats — Exercice 6

```
root@f8b93f9d9019:/app# bin/tpPoisson1D_direct 1
```

```
----- Poisson 1D -----
```

```
Solution with LAPACK
```

```
The relative forward error is relres = 2.634629e-16
```

----- End -----

Conclusion 4 à 6

Les exercices 4 à 6 ont permis de valider l'ensemble du processus de résolution numérique du problème de Poisson 1D :

la discrétisation par différences finies est correctement formulée ;

la structure tridiagonale du système est efficacement exploitée via un stockage bande ;

la résolution directe par LAPACK fournit une solution d'une précision proche de la précision machine ;

les solutions numériques sont en excellent accord avec la solution analytique.

Ces résultats constituent une base solide pour l'étude des méthodes itératives (Exercices 7 à 9) ainsi que pour l'analyse des formats de stockage creux (Exercice 10).

Exercice 7–9 — Comparaison des méthodes itératives

(Richardson / Jacobi / Gauss-Seidel)

Objectif

Cette section compare le comportement de convergence de trois méthodes itératives appliquées au problème de Poisson unidimensionnel : la méthode de Richardson simple, la méthode de Richardson préconditionnée par Jacobi et la méthode de Richardson préconditionnée par Gauss-Seidel.

Méthode

Les trois méthodes utilisent un cadre itératif unifié :

$$u^{(k+1)} = u^{(k)} + M^{-1} (f - Au^{(k)})$$

où :

M=I: correspond à la méthode de Richardson simple ;

M=D: correspond au préconditionnement de Jacobi ;

M=D–E: correspond au préconditionnement de Gauss-Seidel.

La norme du résidu est enregistrée à chaque itération dans le fichier RESVEC.dat.

F-RESVEC.dat	≡ SOL.dat	≡ EX_SOL.dat
1 1.000000	1 6.332542	1 6.363636
2 0.498332	2 7.670020	2 7.727273
3 0.275644	3 9.014118	3 9.090909
4 0.187561	4 10.365863	4 10.454545
5 0.145744	5 11.725590	5 11.818182
6 0.122117	6 13.092977	6 13.181818
7 0.106689	7 14.467118	7 14.545455
8 0.095487	8 15.846643	8 15.909091
9 0.086732	9 17.229863	9 17.272727
10 0.079543	10 18.614932	10 18.636364
11 0.073411		
12 0.068063		
13 0.063889		
14 0.059536		
15 0.056429		
16 0.053275		
17 0.050421		
18 0.047886		
19 0.045724		
20 0.043867		
21 0.042306		
22 0.041191		
23 0.039751		
24 0.038695		
25 0.037418		
26 0.036408		
27 0.035718		
28 0.035073		
29 0.034564		
30 0.034173		
31 0.033892		
32 0.033711		
33 0.032624		

Résultats numériques et analyse

Les expériences numériques montrent que les trois méthodes convergent. Pour la configuration actuelle du problème, le comportement de convergence de la méthode de Richardson simple et de la méthode de Richardson préconditionnée par Jacobi est presque identique, tandis que la méthode préconditionnée par Gauss-Seidel présente une vitesse de convergence nettement plus élevée : le nombre d'itérations nécessaires pour atteindre le même critère de convergence est environ la moitié de celui requis par les deux premières méthodes (124 vs 64).

Ce phénomène indique que, pour la taille du problème considérée, le préconditionnement de Jacobi n'améliore que faiblement les propriétés spectrales du système, tandis que le préconditionnement de Gauss-Seidel, en exploitant les informations les plus récentes de l'itération, améliore significativement l'efficacité de la convergence.

Exercice 10 — Formats de stockage des matrices creuses (CSR / CSC)

Dans l'Exercice 10, les formats de stockage Band, CSR et CSC sont utilisés pour implémenter le produit matrice-vecteur, et les résultats numériques sont comparés composante par composante. Les sorties affichées dans le terminal montrent que les résultats obtenus avec les trois formats de stockage sont strictement identiques. Cette expérience valide la correction des formats de stockage CSR et CSC ainsi que de leurs implémentations associées pour le produit matrice-vecteur, et montre que différents formats de stockage creux peuvent être choisis de manière flexible selon les besoins du problème, tout en garantissant la cohérence des résultats numériques.

```
root@ec478b9b87f83:/app# bin/tpPoisson1D_sparse
i | band | csr | csc
-----
0 | 1.0 | 1.0 | 1.0
1 | 0.0 | 0.0 | 0.0
2 | 0.0 | 0.0 | 0.0
3 | 0.0 | 0.0 | 0.0
4 | 0.0 | 0.0 | 0.0
5 | 0.0 | 0.0 | 0.0
6 | 0.0 | 0.0 | 0.0
7 | 0.0 | 0.0 | 0.0
8 | 0.0 | 0.0 | 0.0
9 | 1.0 | 1.0 | 1.0
root@ec478b9b87f83:/app#
```

Conclusion

Dans ce projet, nous avons étudié de manière systématique la résolution numérique de l'équation de Poisson unidimensionnelle. De la discrétisation du problème à la mise en œuvre et à la comparaison des méthodes directes et itératives, en passant par les différents formats de stockage matriciel, ce travail illustre de façon complète l'application des méthodes de l'algèbre linéaire numérique à des problèmes concrets.

La comparaison entre les méthodes directes et les algorithmes de décomposition LU spécialisés a permis de mettre en évidence les différences entre diverses implémentations en termes de précision numérique et de complexité de calcul. La comparaison des méthodes itératives de Richardson, Jacobi et Gauss-Seidel a clairement montré l'influence des techniques de préconditionnement sur la vitesse de convergence. Enfin,

les expériences portant sur les formats de stockage des matrices creuses ont confirmé la cohérence des résultats numériques obtenus avec différentes structures de données. Dans l'ensemble, ce projet a non seulement validé la correction des algorithmes numériques étudiés, mais a également approfondi la compréhension des relations entre efficacité algorithmique, stabilité numérique et détails d'implémentation, constituant ainsi une base solide pour l'étude de problèmes numériques de plus grande taille ou de dimension supérieure.