

Analyse numérique matricielle

TP6

Vernet Elliot, Courtois Thibault, Imperatore Valentin

January 2023

Introduction

Dans ce TP, nous allons résoudre le Laplacien 1D par des chaînes de Markov. L'objectif est de s'approcher le plus possible de la solution exacte, notamment en faisant varier le pas de discrétisation ainsi que le nombre de simulations effectuées. On comparera deux méthodes utilisant les chaînes de Markov.

Résolution analytique d'équations différentielles

La solution analytique de l'équation $u''(x)=0$ pour $x \in [0; 1]$ est :
 $u(x) = c_1x + c_2$. Avec les conditions aux limites : $u(0) = a$ et $u(1) = b$, on obtient :
 $u(x) = (b-a)x+a$

Résolution numérique d'équations différentielles

Grâce à un développement de Taylor, on peut écrire :

$$-u''(x_i) \approx \frac{-u_{i-1}+2u_i-u_{i+1}}{h^2} = 0$$

Avec $i=1$, on obtient :

$$-u''(x_1) \approx \frac{-u_0+2u_1-u_2}{h^2} = 0 \text{ d'où } \frac{2u_1-u_2}{h^2} = \frac{a}{h^2} \text{ car } u_0 = a \text{ d'après les conditions limites.}$$

$$\text{Puis avec } i=2, \text{ on obtient : } -u''(x_2) \approx \frac{-u_1+2u_2-u_3}{h^2} = 0$$

Soit par récurrence :

$$A_h = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & \dots & 0 & -1 & -2 \end{pmatrix}$$

Méthode de Monte-Carlo

On considère maintenant la chaîne de Markov X_k à $n+1$ états $(0, 1, \dots, n-1, n)$ telle que $P(X_{k+1} = i+1 / X_k = i) = P(X_{k+1} = i-1 / X_k = i) = \frac{1}{2}$ pour $i \in \llbracket 1; n-1 \rrbracket$.

On rappelle que $u(i) = ap + b(1-p)$ où p est la probabilité de sortir en zéro. On choisit arbitrairement $a=1$ et $b=2$.

On calcule numériquement p puis u_5 par une méthode de Monte-Carlo avec N simulations pour $n = 10$ et $N = 10, 100, 1000, 10000$ puis on les recense dans le tableau suivant :

N	10	100	1000	10000
u_5	1.4	1.45	1.512	1.4971
p	0.6	0.55	0.488	0.5029

La valeur exacte de p est 0.5 et la valeur exacte de u_5 est $u(5) = 1 \times 0.5 + 2 \times 0.5 = 1.5$. Les valeurs obtenues numériquement pour p et u_5 sont très proches des valeurs exactes et plus la valeur de N augmente plus on se rapproche de la valeur exacte.

On va maintenant calculer toutes les valeurs de u_i pour $n = 20$ avec $i \in \llbracket 1; n-1 \rrbracket$ et pour $N = 10000$ pour avoir des mesures plus précises puis on les recense dans le tableau suivant :

i	1	2	3	4	5	6	7	8	9	10
u_i	1.0533	1.1010	1.1477	1.1978	1.2492	1.3026	1.3441	1.4015	1.4507	1.4949
i	11	12	13	14	15	16	17	18	19	
u_i	1.5541	1.6060	1.6415	1.7004	1.7433	1.7993	1.8512	1.8967	1.9477	

Les valeurs exactes de u_i pour $n = 20$ avec $i \in \llbracket 1; n-1 \rrbracket$ sont recensées dans le tableau suivant :

i	1	2	3	4	5	6	7	8	9	10
u_i	1	1.05	1.1	1.15	1.2	1.25	1.3	1.35	1.4	1.45
i	11	12	13	14	15	16	17	18	19	
u_i	1.5	1.55	1.6	1.65	1.7	1.75	1.8	1.85	1.9	

Les valeurs obtenues grâce à l'algorithme de Monte-Carlo sont très proches (identiques à 10^{-2} près) des valeurs exactes.

Etude des états visités

On calcule numériquement les u_i avec $i \in \llbracket 1; 9 \rrbracket$ à l'aide du programme fourni avec N simulations pour $n = 10$ et $N = 10, 100, 1000, 10000$ puis on les recense dans le tableau suivant :

N	10	100	1000	10000
u_1	1.0	1.1429	1.1111	1.1064
u_2	1.2	1.2258	1.1935	1.2036
u_3	1.4286	1.2941	1.3034	1.3018
u_4	1.5556	1.4074	1.4067	1.3994
u_5	1.6	1.52	1.5040	1.4953
u_6	1.8571	1.5977	1.5972	1.5959
u_7	1.8571	1.6667	1.7010	1.6975
u_8	1.8571	1.7324	1.8025	1.7982
u_9	2.0	1.8387	1.9016	1.8994

Plus la valeur de N (c'est-à-dire le nombre de simulations) augmente, plus les valeurs u_i obtenues avec l'algorithme sont proches des valeurs exactes. Cette tendance ne s'observe pas très bien sur les valeurs du tableau mais beaucoup mieux sur les graphes suivants :

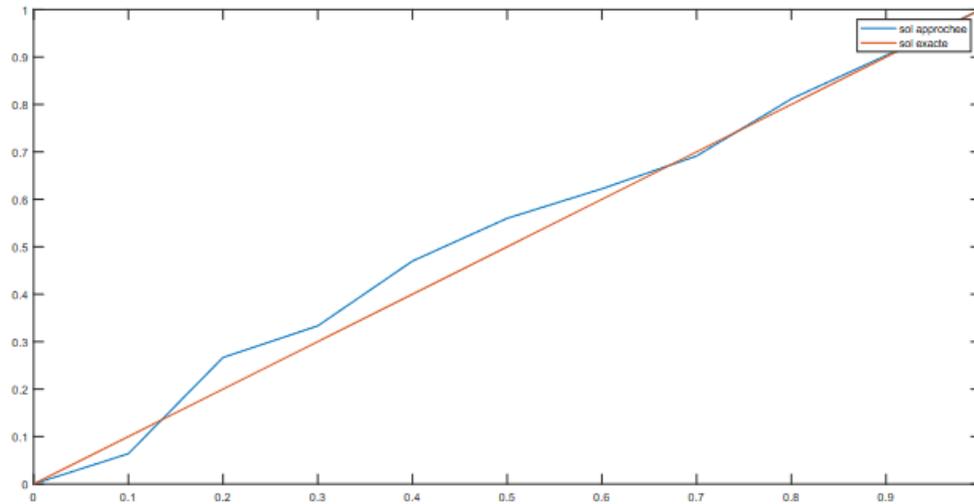


Figure 1: $N = 100$ du programme Markov avec $a = 0$ et $b = 1$

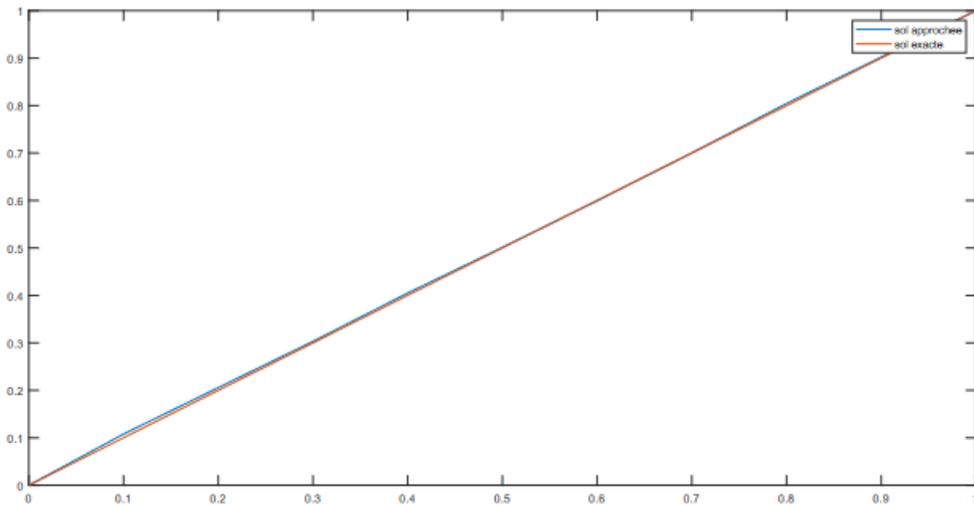


Figure 2: $N = 10000$ du programme Markov avec $a = 0$ et $b = 1$

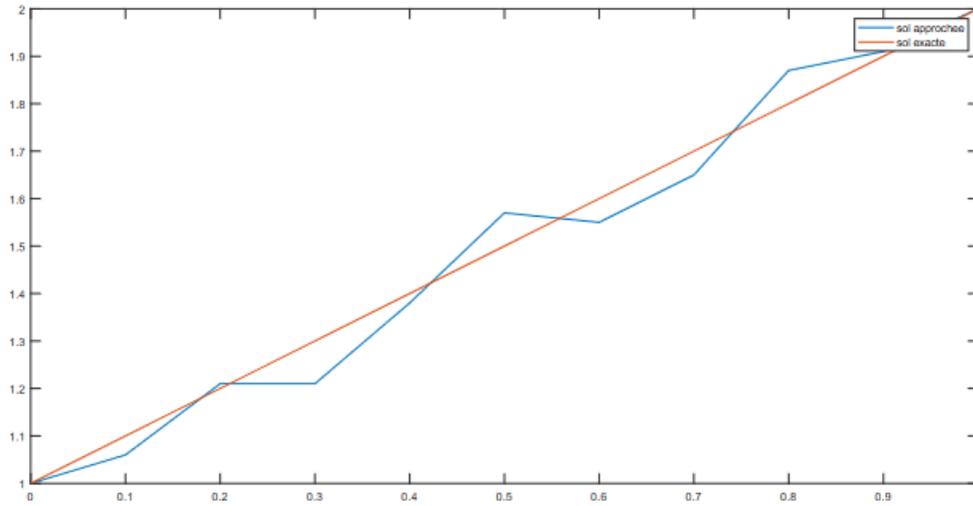


Figure 3: $N = 100$ du programme Markov avec $a = 0$ et $b = 1$

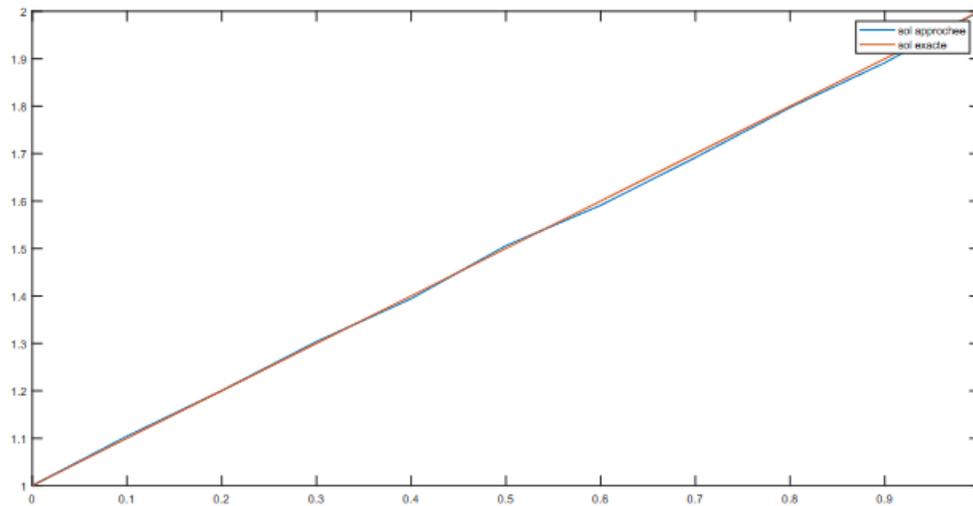


Figure 4: $N = 10000$ du programme Monte-Carlo avec $a = 1$ et $b = 2$

Temps pour programme Monte-Carlo avec $n = 10$ et $N = 10\ 000$: **0.3691 s**

Temps pour programme Markov avec $n = 10$ et $N = 10\ 000$: **0.3774 s**

Précision du programme Markov :

i	1	2	3	4	5	6	7	8	9
$ u_{ex} - u_i $	0.0021	0.0042	0.0131	0.0173	0.0145	0.0151	0.0103	0.0097	0.0074

Précision du programme Monte-Carlo :

i	1	2	3	4	5	6	7	8	9
$ u_{ex} - u_i $	0.0037	0.0007	0.0079	0.0040	0.0022	0.0029	0	0.0005	0.0006

Etonnamment, le programme de Monte-Carlo semble plus précis.

Conclusion

Dans ce TP, on a résolu l'équation du Laplacien 1D analytiquement puis numériquement en la discrétisant par différences finies. Dans la seconde partie, on a résolu cette équation via deux méthodes de chaînes de Markov. On a comparé la précision et le temps de calcul de ces deux méthodes.