EPREUVE COMMUNE DE TIPE 2005 - Partie D

Sources et contrôle des erreurs dans les calculs numériques

Temps de préparation:.....2h15 minutes
Temps de présentation devant le jury:.....10 minutes
Entretien avec le jury:.....10 minutes

GUIDE POUR LE CANDIDAT :

Le dossier ci-joint comporte au total: 13 pages
Document principal : 13 pages

Travail **suggéré** au candidat :
Faire une étude et une présentation structurée du document, en ne suivant pas nécessairement le même ordre. L'utilisation de schémas est recommandée pour expliquer les différentes méthodes numériques présentées dans ce document. Utiliser autant que possible les transparents mis à disposition.

CONSEILS GENERAUX POUR LA PREPARATION DE L'EPREUVE:

⋆ Lisez le dossier en entier dans un temps raisonnable.

⋆ Réservez du temps pour préparer l'exposé devant le jury.

- Vous pouvez écrire sur le présent dossier, le surligner, le découper,...mais tout sera à remettre au jury en fin d'oral.
- En fin de préparation, rassemblez et ordonnez soigneusement TOUS les documents (transparents, etc.) dont vous comptez vous servir pendant l'oral, ainsi que le dossier, les transparents et les brouillons utilisés pendant la préparation. En entrant dans la salle d'oral, vous devez être prêts à débuter votre exposé.
- A la fin de l'oral, vous devez remettre au jury le présent dossier, les transparents et les brouillons utilisés pour cette partie de l'oral, ainsi que TOUS les transparents et autres documents présentés pendnat votre prestation.

Sources et contrôle des erreurs dans les calculs numériques

Introduction

Le calcul numérique sur ordinateur a pris au cours des dernières décennies une place extrèmement importante dans le domaine scientifique et a permis des avancées technologiques spectaculaires. Les prévisions météorologiques, le développement aéronautique, l'imagerie médicale sont autant de domaines où le calcul sur ordinateur est devenu indispensable.
Malgré tous les avantages qu'il procure, un inconvénient majeur du calcul numérique est que les calculs sont faits sur un nombre fini de bits (32 ou 64 généralement), impliquant par là-même une erreur qualifiée d'erreur d'arrondi. Même si chaque erreur est faible ($\sim 10^{-7}$ pour 32 bits par exemple), du fait des milliards d'opérations réalisées l'erreur finale peut être importante. Il est par conséquent absolument nécessaire de comprendre et de contrôler le devenir de ces erreurs pour pouvoir justifier la validité et la précision des résultats obtenus.
Nous allons voir dans ce document, au travers de trois exemples d'algorithmes numériques qui interviennent dans d'innombrables applications, quelles sont les sources d'erreurs et comment tirer profit d'une analyse rigoureuse de leur évolution pour obtenir des résultats numériques corrects.

1 Formules de quadrature

Les formules de quadrature permettent de calculer les intégrales de façon approchée. Elles sont utilisées dans le cas où la primitive de la fonction à intégrer n'est pas connue (ou trop complexe à calculer explicitement), ou bien dans le cas où cette fonction n'est connue que de façon discrète, c'est-à-dire par un nombre fini de valeurs. C'est le cas par exemple pour un signal numérique échantillonné à une certaine fréquence, dont on veut calculer l'énergie ou le spectre.
On se propose donc d'approcher la valeur de l'intégrale suivante:

$$I = \int_a^b f(x)\ dx, \tag{1}$$

où a et b sont des réels [1] et f une fonction de classe $C^\infty([a,b])$.
Nous ne nous intéresserons ici qu'aux formules de quadrature dites *composées*. On

[1] on supposera ici que a et b sont finis, c'est-à-dire que l'on ne considère pas dans ce texte d'intégrales généralisées.

introduit tout d'abord une *segmentation* de l'intervalle $[a, b]$, c'est-à-dire $N+1$ réels distincts, $\{\alpha_i,\ i = 0, ..., N\}$ vérifiant

1) $\alpha_0 = a$ et $\alpha_N = b$

2) $\forall i = 0, ..., N-1 \quad \alpha_i < \alpha_{i+1}$.

En utilisant la relation de Chasles, on obtient immédiatement:

$$I = \sum_{i=0}^{N-1} \int_{\alpha_i}^{\alpha_{i+1}} f(x)\ dx, \tag{2}$$

ce qui nous amène à calculer une approximation de l'intégrale sur chaque intervalle $[\alpha_i, \alpha_{i+1}]$. Pour ce faire, la méthode consiste à approcher f sur cet intervalle par un polynôme, puis d'intégrer celui-ci exactement. Plus précisément, on cherche dans le cas général, une formule de la forme:

$$\int_{\alpha_i}^{\alpha_{i+1}} f(x)\ dx \simeq (\alpha_{i+1} - \alpha_i)\ \sum_{j=0}^{l} \omega_{ij}\ f(\xi_{ij}), \tag{3}$$

où les $\{\xi_{ij},\ j = 0, ..., l\}$ sont des points distincts de l'intervalle $[\alpha_i, \alpha_{i+1}]$ appelés *points de quadrature* et où les $\{\omega_{ij},\ j = 0, ..., l\}$ sont des réels appelés *poids* de la formule de quadrature.
Pour simplifier la présentation, on supposera que les points $\{\xi_{ij},\ j = 0, ..., l\}$ forment une segmentation régulière de l'intervalle $[\alpha_i, \alpha_{i+1}]$, *i.e*,

$$\xi_{ij} = \alpha_i + j \times \frac{(\alpha_{i+1} - \alpha_i)}{l}, \quad j = 0, .., l. \tag{4}$$

On obtient dans ce cas les formules de quadrature dites de Newton-Côtes.
Les points de quadrature étant connus, il reste à calculer les poids. On se ramène alors par changement de variable à l'intervalle $[\alpha_i, \alpha_{i+1}] = [-1, 1]$ subdivisé alors par les points $\{\tau_j = -1 + j \times \frac{2}{l}, \quad j = 0, .., l\}$.
Le polynôme P_l de degré l interpolant une fonction f définie sur $[-1, 1]$ aux points $(\tau_j)_{j=0,..,l}$ est défini par:

$$P_l(\tau_j) = f(\tau_j), \quad j = 0, .., l. \tag{5}$$

Ce polynôme est unique et s'obtient immédiatement à partir des polynômes générateurs de Lagrange:

$$P_l(x) = \sum_{j=0}^{l} f(\tau_j) L_j(x), \tag{6}$$

où

$$L_j(x) = \prod_{k=0, k\neq j}^{l} \frac{(x - \tau_k)}{(\tau_j - \tau_k)}. \tag{7}$$

On approche alors l'intégrale sur $[-1,1]$ par:

$$\int_{-1}^{1} f(x)\ dx \simeq \int_{-1}^{1} P_l(x)\ dx = 2\sum_{j=0}^{l} \omega_j f(\tau_j), \tag{8}$$

avec

$$\omega_j = \frac{1}{2}\int_{-1}^{1} L_j(x)\ dx. \tag{9}$$

Après changement de variable inverse, les coefficients ω_j restent inchangés et l'on obtient la formule de quadrature finale:

$$\underline{\int_{\alpha_i}^{\alpha_{i+1}} f(x)\ dx \simeq (\alpha_{i+1}-\alpha_i)\ \sum_{j=0}^{l} \omega_j\ f(\xi_{ij})}, \tag{10}$$

où les poids sont donnés par la formule (9).

1.1 Exemples

Dans le cas $l=0$ ou $l=1$ il est très simple d'obtenir les formules suivantes:

a) l=0 formule des rectangles:

$$\int_{\alpha_i}^{\alpha_{i+1}} f(x)\ dx \simeq (\alpha_{i+1}-\alpha_i) \times f(\alpha_i) \tag{11}$$

b) l=1 formule des trapèzes:

$$\int_{\alpha_i}^{\alpha_{i+1}} f(x)\ dx \simeq (\alpha_{i+1}-\alpha_i) \times \frac{f(\alpha_i)+f(\alpha_{i+1})}{2}. \tag{12}$$

Pour $l=2$ on obtient la formule de Simpson:

$$\int_{\alpha_i}^{\alpha_{i+1}} f(x)\ dx \simeq (\alpha_{i+1}-\alpha_i) \times \left(\frac{f(\alpha_i)}{6} + \frac{2}{3} f(\frac{\alpha_{i+1}+\alpha_i}{2}) + \frac{f(\alpha_{i+1})}{6}\right) \tag{13}$$

1.2 Ordre et convergence des formules de quadrature

Lorsque l'on utilise en pratique une formule de quadrature, il faut pouvoir estimer l'erreur que l'on commet par rapport à la valeur exacte de l'intégrale (que l'on ne connait pas). Une étude théorique, que nous ne détaillerons pas ici, permet d'obtenir une estimation de cette erreur. Les résultats principaux sont les suivants:

Définition 1: On dit que la formule de quadrature (10) est d'*ordre* r si pour tout polynôme Q de degré inférieur où égal à r :

$$\int_{\alpha_i}^{\alpha_{i+1}} Q(x)\ dx = (\alpha_{i+1} - \alpha_i)\ \sum_{j=0}^{l} \omega_j\ Q(\xi_{ij}), \tag{14}$$

et qu'il existe au moins un polynôme d'ordre $r+1$ pour lequel la formule est inexacte.

On montre alors très facilement que la formule des rectangles (11) est d'ordre 0, que celle des trapèzes (12) est d'ordre 1 et celle de Simpson (13) d'ordre 3.

En définissant l'erreur sur chaque intervalle $[\alpha_i, \alpha_{i+1}]$ par:

$$E_i = \int_{\alpha_i}^{\alpha_{i+1}} f(x)\ dx - (\alpha_{i+1} - \alpha_i)\ \sum_{j=0}^{l} \omega_j\ f(\xi_{ij}), \tag{15}$$

l'erreur totale sur l'intégrale I est donnée par:

$$E = \sum_{i=0}^{N-1} E_i. \tag{16}$$

Le théorème suivant fournit une estimation de l'erreur locale.

Théorème 1:
Si f est une fonction au moins de classe $C^{r+1}([a,b])$ et que la formule de quadrature est d'ordre r, alors il existe une constante $C_i > 0$, indépendante de $h_i = (\alpha_{i+1} - \alpha_i)$, telle que

$$|E_i| \leq C_i\ h_i^{r+2} \max_{x \in [\alpha_i, \alpha_{i+1}]} |f^{(r+1)}(x)|. \tag{17}$$

On déduit, sous les hypothèses de ce théorème, le résultat d'estimation globale de l'erreur:

$$\exists\ C > 0, \quad \text{telle que} \quad \underline{|E| \leq C\ (b-a)\ h^{r+1} \max_{x \in [a,b]} |f^{(r+1)}(x)|}, \tag{18}$$

où $h = \max_{i=0,..,N-1}(h_i)$ et C est une constante indépendante de h.

Ce résultat assure qu'en théorie, pour une fonction donnée, l'erreur de la formule de quadrature tend vers 0 lorsque le pas de la segmentation, h, diminue et que diviser ce pas par 10 divisera l'erreur par 10 pour la méthode des rectangles, par 100 pour les trapèzes et par 10000 pour Simpson.

Sur la figure 1, nous avons représenté l'erreur, obtenue numériquement, entre la valeur exacte de I et sa valeur approchée par la méthode des rectangles ainsi que des

trapèzes. On remarque que la décroissance de cette erreur suit la décroissance théorique jusqu'à un seuil de saturation pour la méthode des trapèzes.
Cette saturation s'explique par la présence dans les calculs d'erreurs d'arrondi qui n'ont pas été prises en compte dans les estimations (17) et (18).

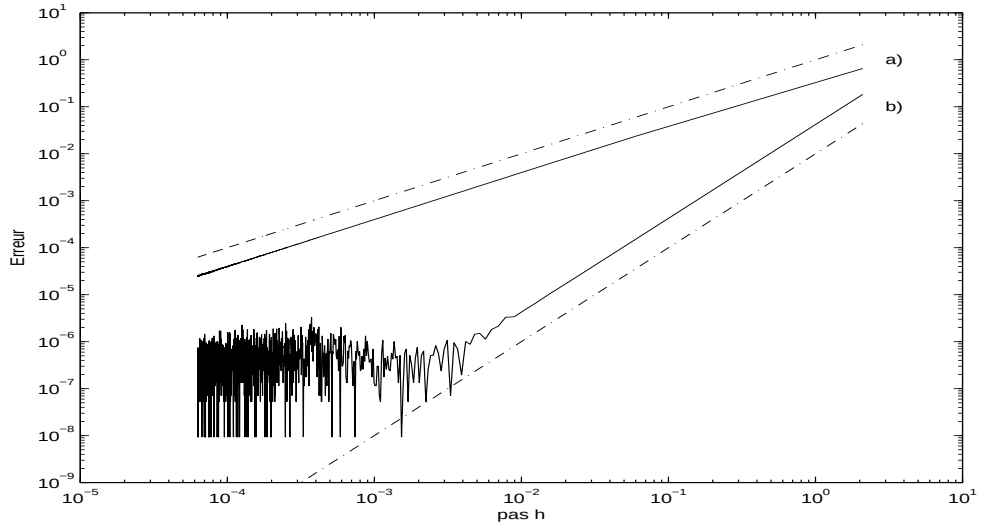

Figure 1: *Erreur entre l'intégrale I et son approximation par une formule de quadrature pour la fonction $f(x) = x^4$ sur l'intervalle $[0, 2\pi]$, en fonction du pas h. Le pas local h_i est constant et égal à $h = \frac{2\pi}{N}$. Les deux axes sont en échelle logarithmique. a): en trait plein l'erreur mesurée pour la formule des rectangles et en pointillé la courbe $y = h$. b): en trait plein l'erreur mesurée pour la formule des trapèzes et en pointillé la courbe $y = h^2/100$.*

1.3 Influence des erreurs d'arrondi

Nous supposons que ces erreurs, dues au calcul de l'ordinateur sur un nombre fini de bits, sont principalement faites lors de l'évaluation de la fonction f:

$$\tilde{f}(\xi_{ij}) = f(\xi_{ij}) + \varepsilon_{ij}. \tag{19}$$

On suppose également que ces erreurs ont un majorant: $\forall i, j \ |\varepsilon_{ij}| \leq \varepsilon$.
En pratique, pour un calcul sur ordinateur en simple précision, (c'est le cas du calcul de la figure 1), on a $\varepsilon \sim 10^{-7}$.
L'erreur locale s'écrit alors:

$$\tilde{E}_i = E_i - h_i \sum_{j=0}^{l} \omega_j \varepsilon_{ij}, \tag{20}$$

qui se majore facilement par:

$$|\tilde{E}_i| \leq |E_i| + \varepsilon h_i \sum_{j=0}^{l} |\omega_j|. \tag{21}$$

L'erreur globale peut donc être majorée par:

$$|\tilde{E}| \leq \sum_{i=0}^{N-1} |E_i| + \varepsilon \sum_{i=0}^{N-1} h_i \sum_{j=0}^{l} |\omega_j|. \tag{22}$$

Si les poids sont tous positifs,(c'est le cas des méthodes des rectangles, des trapèzes et de Simpson), alors

$$\sum_{j=0}^{l} |\omega_j| = \sum_{j=0}^{l} \omega_j = 1, \tag{23}$$

car ces méthodes sont au moins d'ordre 0.
En utilisant cette remarque et le résultat (17), on obtient finalement

$$|\tilde{E}| \leq C\ (b-a)\ h^{r+1} \max_{x\in[a,b]} |f^{(r+1)}(x)| + (b-a)\varepsilon. \tag{24}$$

L'erreur $|\tilde{E}|$ décroit donc avec le pas h comme prévu par l'estimation théorique (18), puis sature autour de la valeur $(b-a)\varepsilon$, ce qui explique le comportement observé sur la figure 1.

La compréhension et l'analyse des sources d'erreurs dans les formules de quadrature permet en pratique de choisir au mieux les paramètres (ordre de la méthode et valeur du pas) en fonction de la précision que l'on cherche sur la valeur de I et du nombre de calculs que l'on est prêt à effectuer.

Dans la partie suivante, nous nous intéressons à la résolution numérique de systèmes linéaires et aux sources d'erreurs qui en résultent.

2 Résolution de systèmes linéaires

Le développement et l'analyse d'algorithmes de résolution de systèmes linéaires est un des domaines de base du calcul scientifique.
En effet, la plupart des méthodes permettant de résoudre numériquement des équations aux dérivées partielles ou d'optimiser des formes ou encore de chercher la solution d'une équation non-linéaire par exemple, passent par la solution d'un ou plusieurs systèmes linéaires. Ces systèmes peuvent comporter des millions d'inconnues ce qui nécessite l'utilisation d'algorithmes numériques extrèmement performants. De par la quantité

phénoménale de calculs à effectuer, les sources d'erreurs sont multiples et se doivent d'être contrôlées.
Nous considérons donc le problème suivant:

$$\text{Trouver } X \in I\!R^n \text{ tel que } AX = b, \tag{25}$$

avec A une matrice réelle inversible de taille $n \times n$ et $b \in I\!R^n$ un vecteur donné.
Le premier algorithme qui vient à l'esprit est celui de Cramer. C'est une méthode théoriquement exacte qui donne les composantes du vecteur X à partir de calculs de déterminants et se programme donc facilement sur un ordinateur. Malgré tout, cette méthode est totalement inutilisable en pratique pour des matrices de taille supérieure à 13×13. En effet, un simple calcul du nombre d'opérations à réaliser pour obtenir la solution montre que celui-ci est proportionnel à $(n+1)!$. En supposant que l'on dispose d'un ordinateur réalisant 1 milliard d'opérations à la seconde, (ce qui est déjà extrèmement performant), on obtient les temps de calcul suivants:

- matrice 20×20: 1620 années
- matrice 50×50: 4.9×10^{49} années!!

De très nombreuses méthodes ont donc été développées, souvent à partir de l'algorithme de base du pivot de Gauss. Celui-ci consiste à trianguler la matrice A à partir de combinaisons linéaires de lignes, puis à résoudre le système triangulaire.
On montre facilement que le nombre d'opérations de cette méthode est proportionnel à n^3, ce qui donne un temps de 10^{-4} secondes sur l'ordinateur précédent pour la matrice 50×50.
Beaucoup de méthodes plus rapides existent (pour des matrices symétriques, symétriques définies positives, des matrices creuses,...),mais il est tout à fait justifié d'utiliser le pivot de Gauss lorsque A est quelconque et de taille raisonnable.
Bien que cette méthode soit exacte, les erreurs d'arrondi par exemple peuvent perturber les coefficients de la matrice ou du second membre. Il arrive aussi très souvent que ces coefficients proviennent de calculs approchés (formules de quadrature par exemple) et ne soient donc qu'une approximation des coefficients du système exact que l'on veut résoudre. Numériquement on résoud donc un système de la forme

$$(A + \delta A)\tilde{X} = b + \delta b, \tag{26}$$

où δA matrice $n \times n$ et δb vecteur de $I\!R^n$ représentent les perturbations.
Avant d'énoncer les résultats théoriques, l'exemple du système suivant est assez parlant. La solution du système $AX = b$ où

$$A = \begin{pmatrix} 8 & 6 & 4 & 1 \\ 1 & 4 & 5 & 1 \\ 8 & 4 & 1 & 1 \\ 1 & 4 & 3 & 6 \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} 19 \\ 11 \\ 14 \\ 14 \end{pmatrix} \quad \text{est} \quad X = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}. \tag{27}$$

Si l'on perturbe aléatoirement le vecteur b:

$$b + \delta b = \begin{pmatrix} 19.01 \\ 11.05 \\ 14.07 \\ 14.05 \end{pmatrix} \quad \text{on obtient} \quad \tilde{X} = \begin{pmatrix} \text{-}2.34 \\ 9.745 \\ \text{-}4.85 \\ \text{-}1.34 \end{pmatrix}. \tag{28}$$

soit une erreur relative sur la solution (mesurée en norme euclidienne) de

$$\frac{\|\tilde{X} - X\|}{\|X\|} = 5.6, \tag{29}$$

alors que l'erreur relative de la perturbation était seulement de l'ordre de 3×10^{-3}.
Le même phénomène se produit lorsque l'on perturbe la matrice:

$$A + \delta A = \begin{pmatrix} 8 & 6.01 & 4 & 1 \\ 1 & 4.05 & 5 & 1 \\ 8.03 & 4 & 1 & 1 \\ 1 & 4 & 3 & 6.07 \end{pmatrix} \quad \text{donne comme solution} \quad \tilde{X} = \begin{pmatrix} 1.69 \\ \text{-}0.81 \\ 2.23 \\ 1.46 \end{pmatrix}. \tag{30}$$

Il est donc indispensable de connaitre une estimation de l'erreur que l'on commet par rapport à X solution du système $AX = b$, c'est-à-dire l'amplitude de l'erreur δX définie par $\tilde{X} = X + \delta X$.
Pour cela, nous introduisons la norme matricielle subordonnée à la norme euclidienne, définie par:

$$\|A\| = \sup_{X \in \mathbb{R}^n, X \neq 0} \frac{\|AX\|}{\|X\|}. \tag{31}$$

Le théorème suivant répond en partie au problème posé.

Théorème 2

Soir A une matrice inversible et b un vecteur non nul.

a) Si X et $\tilde{X} = X + \delta X$ sont les solutions respectives des systèmes

$$AX = b \quad \text{et} \quad A\tilde{X} = b + \delta b, \tag{32}$$

alors

$$\frac{\|\delta X\|}{\|X\|} \leq \|A\|\|A^{-1}\| \frac{\|\delta b\|}{\|b\|}. \tag{33}$$

b) Si X et $\tilde{X} = X + \delta X$ sont les solutions respectives des systèmes

$$AX = b \quad \text{et} \quad (A + \delta A)\tilde{X} = b, \tag{34}$$

alors

$$\frac{\|\delta X\|}{\|\tilde{X}\|} \leq \|A\|\|A^{-1}\| \frac{\|\delta A\|}{\|A\|}. \tag{35}$$

Le réel positif $\|A\|\|A^{-1}\|$ est appelé *conditionnement* de la matrice A et noté cond(A).

Ce théorème se démontre très facilement et nous apprend que les perturbations sur la solution sont majorées par les erreurs sur les données, multipliées par le conditionnement. En particulier, même si la perturbation est faible, l'erreur sur la solution peut être importante si cond(A) est grand. C'est le cas de l'exemple précédent pour lequel cond$(A)\simeq 3198$.
Il est donc indispensable en pratique de travailler avec des matrices dont le conditionnement est faible ce qui garantit une faible amplification des erreurs.
En utilisant le fait que la norme définie par (31) est une norme matricielle ($\|AB\| \leq ||A||||B||$), on obtient immédiatement que cond$(A)\geq 1$.
On dira alors qu'une matrice est bien conditionnée si cond$(A)\simeq 1$ et mal conditionnée si cond$(A)>> 1$.
Le calcul exact du conditionnement revient très cher en pratique (plus que la résolution du système) et l'on se contente généralement d'une estimation donnée par un algorithme rapide puisque seul son ordre de grandeur importe.
Lorsque l'on a un système linéaire mal conditionné à résoudre, on a tout intérêt à utiliser un préconditionneur, c'est à dire une matrice C inversible telle que

$$\text{cond}(CA) << \text{cond}(A), \tag{36}$$

avec idéalement cond$(CA)\simeq 1$ et de résoudre le système $CAX = Cb$.
Le choix optimal de la matrice C est évidemment $C = A^{-1}$, ce qui est absurde en pratique!
Le problème du choix d'un bon préconditionneur est un problème très difficile et fait l'objet de nombreuses recherches à l'heure actuelle.

Nous allons voir dans la partie suivante l'influence d'un mauvais conditionnement de matrice dans la résolution d'une équation différentielle.

2.1 Résolution numérique d'une équation différentielle

On s'intéresse à la résolution numérique par la méthode des *différences finies* d'un problème différentiel très simple: Trouver $U(x)$ telle que

$$-U''(x) = f(x) \qquad x \in]0,1[\tag{37}$$

avec f une fonction donnée et vérifiant les conditions aux limites

$$U(0) = 1 \quad \text{et} \quad U(1) = 0. \tag{38}$$

La méthode consiste dans un premier temps à segmenter l'intervalle $[0,1]$ en $N+1$ points avec un pas h constant, c'est à dire à introduire les points:

$$x_i = i \times h, \quad i = 0, ..., N \quad \text{avec} \quad h = \frac{1}{N}, \tag{39}$$

puis à calculer une approximation de U aux points x_i, que l'on notera U_i.
On approche alors la dérivée U'' aux point x_i par la formule suivante:

$$U''(x_i) = \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} + o(h^2) \quad i = 1, ..., N-1. \tag{40}$$

En écrivant alors le système (37) aux points x_i, on obtient un système linéaire:

$$\begin{aligned}
U_0 &= 1 \\
-\frac{U_2 - 2U_1 + U_0}{h^2} &= f(x_1) \\
&... \qquad (41)\\
-\frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} &= f(x_i) \qquad (42)\\
&... \qquad (43)\\
U_N &= 0 \qquad (44)
\end{aligned}$$

qui s'écrit sous forme matricielle $A_N \mathcal{U}_N = b_N$ avec $\mathcal{U}_N = (U_1, U_2, ..., U_{N-1})^T$

$$A_N = \frac{1}{h^2}\begin{pmatrix}
2 & -1 & & & & & \\
-1 & 2 & -1 & & & & \\
 & \ddots & \ddots & \ddots & & 0 & \\
 & & \ddots & \ddots & \ddots & & \\
 & 0 & & \ddots & \ddots & \ddots & \\
 & & & & -1 & 2 & -1 \\
 & & & & & -1 & 2
\end{pmatrix} \quad \text{et} \quad b_N = \begin{pmatrix}
f(x_1) + \frac{1}{h^2} \\
f(x_2) \\
\vdots \\
\vdots \\
\vdots \\
f(x_{N-2}) \\
f(x_{N-1})
\end{pmatrix}. \tag{45}$$

Les valeurs de la solution exacte U aux points x_i sont donc approchées en résolvant un système linéaire tri-diagonal, ce qui du point de vue algorithmique est relativement simple.
Il se pose alors la question de la convergence de la méthode, c'est-à-dire de savoir si $\|\mathcal{U}_N - \mathcal{U}\|$ tend vers 0 lorsque N tend vers l'infini, avec $\mathcal{U} = (U(x_1), U(x_2), ..., U(x_{N-1}))^T$.
On peut effectivement démontrer théoriquement que:

$$\|\mathcal{U}_N - \mathcal{U}\| \sim \frac{1}{N^2}, \tag{46}$$

ce qui garantit la convergence de la méthode et renseigne également sur la vitesse de cette convergence.

Malheureusement, ce résultat ne tient pas compte de l'influence des erreurs d'arrondi dans la résolution du système linéaire. D'après la partie précédente, on sait que celle-ci dépend essentiellement du conditionnement de la matrice A_N.
De par sa forme simple, il est possible ici de calculer directement ce conditionnement et d'obtenir:

$$\mathrm{cond}(A_N) \sim \frac{4}{\pi^2} N^2. \tag{47}$$

Ce résultat est extrèmement défavorable et conduit à un dilemme: lorsque N est grand le vecteur $\mathcal{U}_N$ est théoriquement très proche de la solution exacte, mais plus N est grand, plus le calcul de $\mathcal{U}_N$ est délicat à cause d'éventuelles amplifications des erreurs d'arrondi ce qui risque d'éloigner irrémediablement $\mathcal{U}_N$ de la solution exacte.
Ce phénomène s'observe sur la figure 2 où l'on remarque qu'à partir d'un certain pas h, l'erreur entre la solution exacte et la solution approchée 'explose' et ne tend plus vers 0.
Il est donc nécessaire, même pour ce problème très simple d'utiliser un préconditionneur pour garantir la convergence vers la solution exacte.

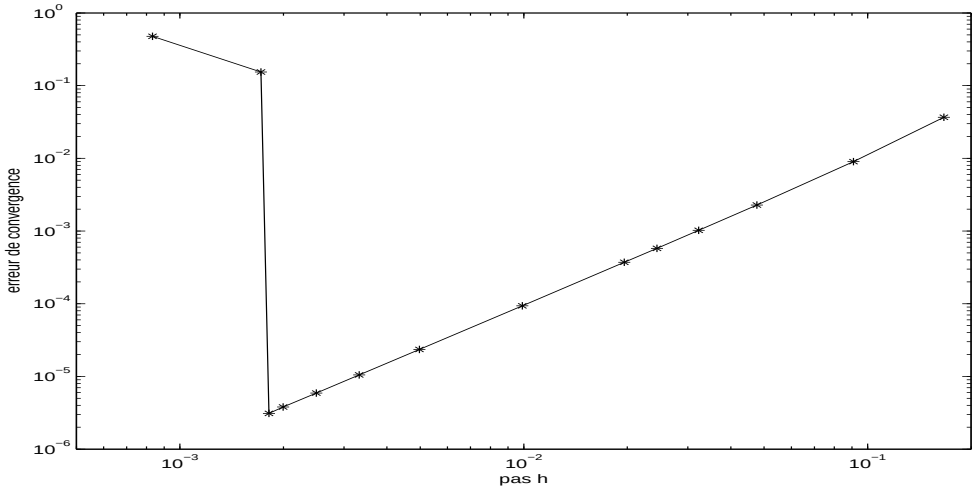

Figure 2: *Erreur en norme Euclidienne entre la solution exacte U et son approximation* $\mathcal{U}_N$ *par la méthode de différences finies, en fonction du pas* $h = 1/N$. *Les deux axes sont en échelle logarithmique. On a résolu l'équation (37) avec* $f(x) = -2\pi \sin(\pi x) + \pi^2(1-x)\cos(\pi x)$ *ce qui donne comme solution exacte* $U(x) = (1-x)\cos(\pi x)$.

3 Conclusion

Nous venons de voir à partir de quelques exemples l'importance de maitriser les sources d'erreurs et leur propagation dans les calculs numériques. A l'heure où les simulations numériques de phénomènes physiques prennent une place grandissante dans le monde scientifique et industriel, il est capital de savoir en contrôler toutes les étapes et de garantir ainsi une bonne adéquation entre les résultats et la réalité physique. Ceci passe entre autres par une analyse mathématique précise des algorithmes utilisés et de leur stabilité vis-à-vis des erreurs inhérentes au calcul sur ordinateur.