

Analyse numérique et mathématiques appliquées

Christophe Antonini¹, Olivier Teytaud², Pierre Borgnat³, Annie Chateau⁴, and
Edouard Lebeau⁵

¹Enseignant en CPGE, Institut Stanislas, Cannes

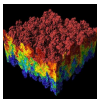
²Chargé de recherche INRIA, Université d'Orsay, Orsay

³Chargé de recherche CNRS, ENS Lyon, Lyon

⁴Maitre de conférence, Université Montpellier-2, Montpellier

⁵Enseignant en CPGE, Lycée Henri Poincaré, Nancy

26 novembre 2022



Un peu d'analyse numérique.

1 Analyse numérique et mathématiques appliquées

Ce modeste chapitre ne sera qu'une très brève introduction, plus culturelle que technique, aux nombreux domaines d'analyse numérique dont nous ne parlons pas ailleurs dans cet ouvrage. Des références utiles pour aller (beaucoup) plus loin sont [1] (analyse numérique matricielle, optimisation), [2] (calcul approché d'intégrales, équations différentielles), [3] (calcul approché d'intégrales par quasi-Monte-Carlo), [4] (Monte-Carlo), et l'incontournable wikipédia ([5]). On parlera ici de recherche des zéros d'une fonction, de recherche des minima d'une fonction, de calcul approché d'intégrales de fonctions. On ne parlera pas de nombreux champs passionnants des mathématiques appliquées : calculs en éléments finis (pour la résolution numérique d'équations aux dérivées partielles), systèmes dynamiques, contrôle optimal ou contrôle approché. On trouvera ailleurs dans cette ouvrage l'analyse de Fourier, les équations différentielles, l'approximation de fonctions, les probabilités et statistiques.

1.1 Extrema

Les extrema sont l'exemple naturel de mathématiques directement utiles. Si vous avez un processus industriel dont les paramètres θ entraînent un coût $f(\theta)$, alors vous pouvez naturellement être intéressés par minimiser $f(\theta)$; vous cherchez donc des extrema de f . Après quelques généralités (1.1.1), nous nous pencherons sur la compacité (1.1.2), utile pour garantir l'existence d'extrema,

puis sur le calcul différentiel (1.1.3) qui nous permettra de caractériser les extremas, et enfin sur la notion de convexité (1.1.4), qui permet de montrer qu'un optimum local est global, sans oublier une section pour ceux qui veulent aller plus loin (1.1.5). Quelques éléments plus pratiques sur la recherche d'extremas sont donnés en section 1.

1.1.1 Cadre et définitions

Pour ce chapitre, on travaillera sur une application f continue d'un ouvert U d'un espace de Banach E dans \mathbb{R} .

DÉFINITION 0.1 minimum relatif

On dit que f admet un **minimum relatif** ou **minimum local** en $x \in U$ si il existe un voisinage V de x tel que pour tout v dans V $f(x) \leq f(v)$.

On dit que f admet un **minimum relatif strict** ou **minimum local strict** en $x \in U$ si il existe un voisinage V de x tel que pour tout $v \neq x$ dans V $f(x) < f(v)$.

On dit que f admet un **minimum global** en $x \in U$ si pour tout v dans U $f(x) \leq f(v)$.

On dit que f admet un **minimum global strict** en $x \in U$ si pour tout $v \neq x$ dans U $f(x) < f(v)$.

On définit de même les notions de **maximum relatif**, **maximum relatif strict**, **maximum global**, **maximum global strict**, en remplaçant les \leq par des \geq et les $<$ par des $>$.

1.1.2 Résultats liés à la compacité

PROPOSITION 0.1

Soit E un espace vectoriel normé de dimension finie et $f \in C^0$ de E dans \mathbb{R} telle que

$$\lim_{\|x\| \rightarrow \infty} f(x) = \infty$$

Alors f est minorée et atteint son minimum.

Démonstration On se donne $A > 0$ tel que $\|x\| > A$ implique $f(x) > f(0)$. On considère alors $K = \overline{B}(0, A)$. K est compact car on est en dimension finie (les compacts d'un espace vectoriel normé de dimension finie sont les fermés bornés). f atteint donc sa borne inf (voir corollaire ??).

La condition 0.1 et d'autres conditions s'y rapportant sont parfois qualifiés de coercivité.

COROLLAIRE 0.2

Quelques applications :

- La distance d'un point à un fermé non vide est minorée et le minimum est atteint.
- Aussi les deux applications suivantes, empruntées à [6] :
 - étant donnée une application $f C^0$ de $[0,1]$ dans \mathbb{R} , il existe un polynôme P minimisant $\|f - P\|_\infty$ parmi les polynômes de degré $\leq n$.
 - le **théorème de D'Alembert-Gauss**, stipulant que tout polynôme à coefficients complexes et de degré ≥ 1 admet une racine, en considérant $z \mapsto |P(z)|$ (corollaire : tout polynôme à coefficients dans \mathbb{C} est scindé dans $\mathbb{C}[X]$).

Application 0.1 Voir le corollaire ?? sur la trigonalisation de matrices complexes, ou la partie ?? sur les suites récurrentes linéaires.

1.1.3 Résultats de calcul différentiel

Quand une fonction est dérivable, chercher un minimum local revient à chercher un point en lequel les dérivées (1ère et/ou seconde) vérifient certaines conditions ; d'où l'importance de caractérisations locales (par calcul différentiel) des minima ou maxima d'une fonction.

Il est à noter que l'optimisation sous contraintes (cf section 1 pour quelques mots sur ce sujet) donne lieu à des conditions bien différentes.

THÉORÈME 0.3 Condition nécessaire du premier ordre

Si x est un minimum relatif de f et si f est différentiable en x , alors la différentielle de f en x est nulle.

Résultats au premier ordre

Démonstration

$$df(x)(h) \geq 0 \text{ car } \frac{f(x+th)-f(x)}{t} \geq 0 \text{ si } t \geq 0$$
$$df(x)(h) \leq 0 \text{ car } \frac{f(x+th)-f(x)}{t} \leq 0 \text{ si } t \leq 0$$

⚠ Attention 0.2 Pas de réciproque ; par exemple $x \mapsto x^3$ de \mathbb{R} dans \mathbb{R} a une différentielle nulle en 0 et n'a ni maximum ni minimum en zéro.

DÉFINITION 0.2

Si $df(x) = 0$, on dit que x est un point critique.

Pour aller plus loin on peut s'intéresser aux extréma liés ; voir pour cela [1].

THÉORÈME 0.4 Condition nécessaire du second ordre

Soit f deux fois différentiable en x .
Alors si f admet un minimum local en x , $f''(x)$ est positive.

Résultats du second ordre

Démonstration

Supposons $f''(x)$ non positive; alors il existe v tel que $f''(x)(v, v)$ soit < 0 . On se donne un voisinage V de x sur lequel $f \leq f(x)$ et sur lequel la formule de Taylor-Young ?? donne

$$f(x + t.v) \leq f(x) + \frac{1}{4}f''(x)(tv, tv)$$

(rappelons que par le résultat précédent $f'(x)$ est nul)
on peut toujours choisir un tel voisinage V car

$$f(x + t.v) = f(x) + \frac{1}{2}f''(x)(tv, tv) + o(t^2)$$

pour t assez petit

et donc $f(x + tv) - f(x)$ est alors négatif pour ces valeurs de t (à part pour $t = 0$).

THÉORÈME 0.5 Condition suffisante du second ordre

Supposons E de dimension finie. Soit f deux fois différentiables en x , $f'(x) = 0$, et $f''(x)$ définie positive. Alors f admet un minimum relatif strict en x .

Démonstration On considère simplement un minimum de $f''(x)(u, u)$ pour $\|u\| = 1$ et la conclusion vient rapidement.

Intuition On peut se passer d'hypothèse de dimension finie à condition d'imposer que $f''(x)$ vérifie $\exists \alpha > 0; f''(x)(u, u) > \alpha \|u\|^2$.

1.1.4 La convexité

Pour une introduction à la convexité, voir la partie ??.

Les résultats liés à la convexité sont très intuitifs, et se justifient rigoureusement sans trop de difficulté : on pourra consulter [1] pour moult développements. Le théorème suivant est très naturel :

THÉORÈME 0.6

Un minimum local d'une fonction convexe définie sur une partie convexe est en fait un minimum global.

Une fonction strictement convexe définie sur une partie convexe admet au plus un minimum et si un tel minimum existe il est strict.

Bien sûr, cela ne nous dit pas comment traiter de problèmes non-convexes. Divers algorithmes permettent de trouver des minima de fonctions en très grande dimension et en temps raisonnable, lorsque le problème est très convexe; le cas général, non convexe, est bien sûr beaucoup plus délicat. Il est facile de montrer que l'algorithme consistant à simplement tirer au sort n points dans \mathbb{R}^d suivant une distribution gaussienne et à sélectionner le meilleur de ces points trouve l'essentiel minimum de toute fonction mesurable à mesure que $n \rightarrow \infty$ (au sens où $\inf_{i \in [1, n]} f(X_i)$ où les X_i sont aléatoires simples tend presque sûrement vers le minimum essentiel de f , quelle que soit la fonction f). L'inconvénient de cet algorithme est bien sûr sa lenteur. Une façon de concilier cet algorithme et des méthodes plus usuelles de descente de gradient ou de Newton est d'effectuer

plusieurs algorithmes de Newton (ou quasi-Newton) à partir de points initiaux différents. On peut aussi se référer à l'important champ des algorithmes évolutionnaires qui essaient de faire mieux (plus rapide) que la recherche aléatoire ci-dessus, tout en en préservant la robustesse.

1.1.5 Pour aller plus loin

La partie 1 contient quelques développements, dont l'indispensable méthode de Newton.

La notion de quasi-convexité, plus générale que la convexité, est parfois utile en recherche d'extrema ; une fonction f est quasi-convexe si $\{x; f(x) < t\}$ est convexe quel que soit t . En particulier, certaines méthodes d'optimisation, dites méthodes de recherche directe (on trouvera plus facilement "direct search methods" sur internet), parviennent à trouver les minima de fonctions quasi-convexes sous des hypothèses très générales.

On peut aussi s'intéresser aux extrêmes d'une fonction, sous contraintes. On pourra pour cela consulter [1]. On n'a pas ici formulé les caractérisations par calcul différentiel de l'optimalité locale d'un point en termes de conditions sur le jacobien ou la hessienne. Le lecteur est encouragé à se référer à la section ?? pour y découvrir la hessienne, à la section ?? pour y découvrir jacobienne et jacobien, et à [1] pour en apprendre plus sur le sujet de l'énorme lien entre algèbre linéaire (et en particulier matricielle) et optimisation. En résumé :

- le gradient est nul si la jacobienne est nulle.
- la dérivée seconde $f''(x)$ est définie positive si la matrice hessienne de f en x est définie positive (ce qui, rappelons-le, est beaucoup plus fort que le fait que le hessien soit strictement positif).

1.2 Recherche du zéro d'une fonction

Soit f une application de \mathbb{R}^d dans \mathbb{R}^d dont on cherche un point où elle s'annule. La méthode ci-dessous est appelée **méthode de Newton-Raphson** ou simplement **méthode de Newton** quand $d = 1$.

- Choisir un point initial $x \in \mathbb{R}^d$.
- Tant que $f(x)$ n'est pas assez proche de 0, $x \leftarrow x - (f'(x))^{-1}f(x)$.

Ceci implique que $f'(x)$ soit toujours inversible. Notons bien que f peut être à valeurs dans un espace multidimensionnel. On pourra se référer à [1] pour plus d'informations.

Exemple 0.3 (pour $d = 1$) : recherche d'une valeur approchée de $\sqrt{2}$: zéro de la fonction $x \mapsto x^2 - 2$ (cet exemple est aisément généralisable).

La méthode explicitée plus haut consiste à étudier une suite récurrente suivant la relation

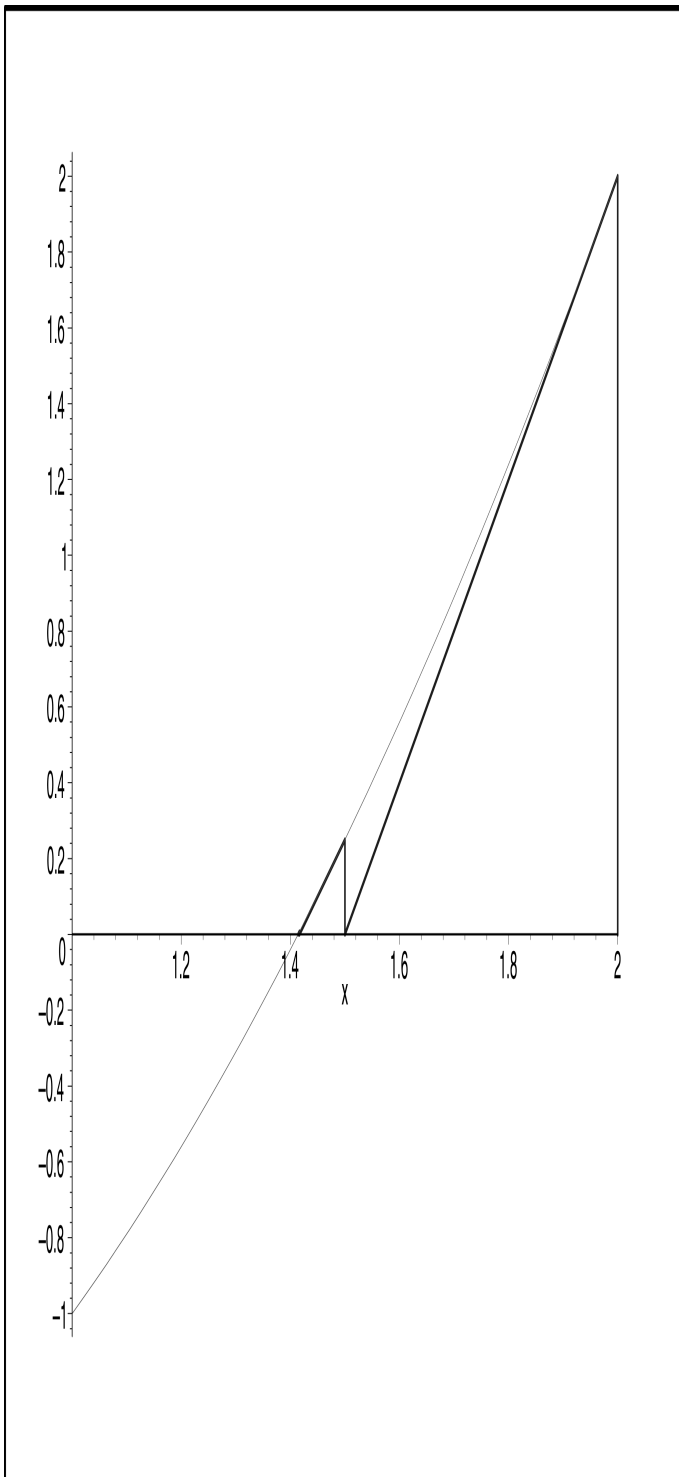
$$\forall n \in \mathbb{N}, u_{n+1} = u_n - \frac{u_n^2 - 2}{2u_n} = \frac{u_n^2 + 2}{2u_n},$$

d'où

$$\forall n \in \mathbb{N}, u_{n+1} - \sqrt{2} = \frac{(u_n - \sqrt{2})^2}{2u_n}.$$

Une étude rapide de la suite (u_n) permet de montrer que si $u_0 > \sqrt{2}$, alors on a $\forall n, u_n > \sqrt{2}$, et (u_n) tend en décroissant vers 0 avec

$$v_{n+1} = \frac{v_n^2}{2(v_n + \sqrt{2})} \quad \text{avec} \quad v_n = u_n - \sqrt{2}.$$



1.3 Recherche du minimum d'une fonction

La recherche du minimum (quand il existe) d'une fonction est un très vaste problème. Différents cas intéressants se gèrent de manières très différentes. En général, on cherche à trouver x tel que $f(x)$ soit minimal et $g(x) \geq 0$; f est appelée **fonction objectif** et g est appelé **contrainte**; g peut-être à valeurs dans \mathbb{R}^k , et on peut avoir deux coordonnées opposées pour inclure des cas d'égalité.

Voyons tout d'abord le cas du continu, i.e. des fonctions de domaine inclus dans \mathbb{R}^d (non dégénéré).

- f et g sont des applications affines : il s'agit de la **programmation linéaire**. La traditionnelle méthode du simplexe est désormais dépassée par des algorithmes dont on prouve qu'ils sont polynomiaux, et notamment les **méthodes de points intérieurs**, qui sont en fait utilisables aussi pour des applications non-linéaires (mais avec beaucoup moins de garanties que dans le cas linéaire).
- f et g sont quadratiques : on parle alors de **programmation quadratique**. Différentes méthodes existent ; pour pouvoir travailler en très grande dimension, on résout en général le problème sur un sous-ensemble de variables (les autres variables étant fixées), et on change de sous-ensemble de variables à chaque fois. Si les sous-ensembles sont choisis suffisamment petits, alors à chaque étape une résolution exacte est possible (par exemple, certains programmes utilisent des sous-ensembles à deux variables seulement).
- Cas général dans des espaces continus : les méthodes de points intérieurs sont fréquemment utilisées, ainsi que les **descentes de gradient** (méthode dite d'ordre 1) ou les méthodes d'ordre 2 (utilisant la hessienne).

Sous sa forme la plus élémentaire, une descente de gradient se formule comme suit :

1. Choisir x un point initial.
2. Initialiser $n = 1$.

3. Ajouter 1 à la variable n . Corriger x en $x - (\eta/n)\nabla f(x)$, où ∇f est le gradient de f .
4. Si $g(x)$ est < 0 , alors on « rectifie » x . Cela peut se faire par projection si la forme de g le permet. Sinon, on peut remplacer $x - (\eta/n)\nabla f(x)$ par $x - (\eta/n)\nabla(f(x) - \mu_n \ln(g(x)))$, où μ est une fonction décroissante de n .

Les méthodes d'ordre 2 utilisent la hessienne, parfois explicitement (en calculant la hessienne) et plus souvent implicitement (en l'approchant à partir des valeurs du gradient rencontrées aux différentes étapes, comme dans l'algorithme BFGS). Par exemple, on peut utiliser les conditions d'optimalité (cf le chapitre 1.1) pour transformer la recherche de $\arg \min f$ en la recherche de x tel que $f'(x) = 0$ (sous certaines conditions bien sûr), ou d'autres caractérisations des optima. On voit ainsi qu'on peut se ramener à la recherche des zéros d'une fonction (possiblement multidimensionnelle). Il est important de noter l'importance de la convexité; minimiser une fonction convexe (ou quasi-convexe, ou strictement convexe) est plus facile que minimiser une fonction non-convexe car, grosso-modo, lorsque le gradient s'annule, on a plus de chances d'être au minimum (il y aurait beaucoup à dire ici, selon que l'on a des contraintes ou non, selon la régularité de la fonction; nous en restons ici à un tableau d'ensemble).

Il faut noter que le traitement des contraintes est un problème très délicat, surtout lorsque le nombre de contraintes est grand. De nombreuses techniques d'optimisation peuvent se modifier par un passage dans le dual (en utilisant les coefficients de Lagrange); le problème contraint devient ainsi un problème non-contraint dans lequel on cherche un point-selle. La méthode de point intérieur consiste (typiquement) à minimiser à chaque étape $f(x) - \mu_n \ln(g(x))$, avec μ tendant vers 0 et convenablement choisi (notez que pour cela, il faut disposer d'un point où x où $g(x)$ soit positif, sans quoi $\ln(g(x))$ a peu de sens!).

Mais l'optimisation ne se restreint pas, loin de là, à l'optimisation dans des espaces continus. Le cas d'optimisation sur \mathbb{Z}^d (ou même $\{0,1\}^d$) au lieu de \mathbb{R}^d est un problème important aux nombreuses applications concrètes.

Une technique usuelle pour le cadre discret consiste à minimiser un prolongement régulier de f sur \mathbb{R}^d (pour autant qu'on puisse en trouver un). Une fois le résultat obtenu, on choisit une variable non-entière (par exemple $x_3 = 4.5$) et on remplace le problème par deux problèmes similaires, l'un avec la contrainte $x_3 \leq 4$, l'autre avec la contrainte $x_3 \geq 5$. On voit ainsi qu'on se ramène à deux problèmes plus petits; chacun des problèmes va se ramener à deux problèmes, etc; l'astuce est alors d'arriver à « couper » des branches, i.e. à établir rigoureusement que certaines branches ne conduiront jamais à une solution plus intéressante que celles que l'on a déjà. Ce principe est connu sous le nom de "branch&bound", ou "branch& cut".

Des méthodes d'exploration aléatoire ont aussi été définies. Elles sont compatibles avec des domaines continus ou discrets, ou mixtes (certaines variables dans \mathbb{R} et d'autres dans \mathbb{Z} ou $\{0,1\}$). Les méthodes ainsi définies comportent :

- le recuit simulé (on pourra par exemple se référer à [7]);
- les algorithmes évolutionnaires ([8]);
- les algorithmes à estimation de distribution.

Ces méthodes ont en particulier bonne réputation dans les cas désespérés où aucune méthode ne marche.

Par ailleurs, on a parlé ici d'optimisation non-linéaire, mais un cadre particulier est celui de problèmes où l'on cherche à déterminer un contrôleur, c'est-à-dire le cas où l'on cherche une fonction apte à prendre des décisions (pardon de ce raccourci de langage). Le lecteur intéressé est renvoyé à

[9] pour une introduction à ces problèmes passionnants mettant en jeu une forme bien particulière de décomposition¹ : le principe de Bellman (parfois aussi dit principe de Massé ou principe de Pontryagin sous des formes variées).

Enfin, on a ici seulement parlé de l'optimisation d'une fonction sous contraintes ; dans beaucoup de cas, on a plusieurs fonctions à optimiser (par exemple, maximiser l'efficacité et l'esthétique, tout en minimisant le coût). Il s'agit là du champ de l'optimisation multi-objectifs, où l'on cherche non pas un extremum, mais (à peu près) tous les compromis optimaux entre les différents critères.

1.4 Calcul approché d'une intégrale

On se donne une fonction f , dont on cherche à calculer l'intégrale sur D (D pouvant être une partie quelconque de \mathbb{R}^n).

1.4.1 Méthode déterministe mono-dimensionnelle

Une référence claire à ce propos est [2] (comme d'ailleurs pour beaucoup de choses en analyse numérique). On supposera ici que D est un intervalle $[a, b]$ ² borné de \mathbb{R} . Pour des intégrales multi-dimensionnelles, quoique théoriquement les méthodes ci-dessous puissent être envisagées, on utilise plutôt les méthodes proposées dans les sections qui suivent.

Une solution générale est la **méthode de Newton-Cotes** d'indice l : $\int_{[a,b]} f$ est alors approximé par

$$\sum_{i=0}^{k-1} (\alpha_{i+1} - \alpha_i) I(f, \alpha_i, \alpha_{i+1})$$

où α_i est défini par $\alpha_i = a + i(b - a)/k$, et où $I(f, x, y)$ est défini par

$$I(f, x, y) = \sum_{j=0}^l w_j f(x + j(b - a)/l).$$

w_j est défini comme suit :

- **méthode des trapèzes** ($l = 1$) : $w_0 = w_1 = \frac{1}{2}$ (converge en $1/k^2$ si la dérivée seconde existe et est bornée).
- **méthode de Simpson** ($l = 2$) : $w_0 = \frac{1}{6}$, $w_1 = \frac{2}{3}$, $w_2 = \frac{1}{6}$ (converge en $1/k^4$ si la dérivée quatrième existe et est bornée).
- **méthode de Boole-Villarceau** ($l = 4$) : $w_0 = \frac{7}{90}$, $w_1 = \frac{16}{45}$, $w_2 = \frac{2}{15}$, $w_3 = \frac{16}{45}$, $w_4 = \frac{7}{90}$ (converge en $1/k^6$ si la dérivée sixième existe et est bornée).

Noter que l'on peut améliorer ces méthodes : pour calculer $I(f, x, y)$, on ne considère plus des points équirépartis entre x et y , mais les racines de polynômes orthogonaux... Par exemple, prenons (ce qui correspondrait à $l = 3$)

$$I(f, x, y) = \sum_{j=1}^3 c_j f(t_j x + (1 - t_j)y),$$

1. On parle de décomposition lorsqu'on ramène un « gros » problème d'optimisation à plusieurs problèmes plus petits.

2. Cette notation pouvant désigner $]a, b[$, $[a, b]$, $[a, b]$ ou $[a, b[$

avec $c_1 = c_3 = \frac{5}{18}$, $c_2 = \frac{4}{9}$ et $t_1 = 1 - t_3 = \frac{5 - \sqrt{15}}{10}$, $t_2 = \frac{1}{2}$. On arrive alors à une convergence en $1/k^6$ (si la dérivée sixième de f existe et est bornée).

Ces valeurs des t_i sont les racines du polynôme $20X^3 - 30X^2 + 12X - 1$, qui est orthogonal à $\mathbb{R}_2[X]$ pour le produit scalaire $(f, g) = \int_0^1 f(x)g(x)dx...$

Ce choix des t_i est le **seul** permettant cette vitesse de convergence. n

1.4.2 Méthode probabiliste ou quasi-aléatoire : méthode de Monte-Carlo et quasi-Monte-Carlo

Les méthodes précédemment décrites requièrent certaines régularités de la fonction pour assurer une convergence rapide. D'autres méthodes, plus lentes, mais exportables à toute dimension et moins exigeantes en matière de régularité, existent ; par exemple, la méthode de Monte-Carlo (dont les prolongements biaisés sont très nombreux), n'a que très peu d'hypothèses pour garantir la convergence. La méthode de quasi-Monte-Carlo est un peu plus gourmande d'hypothèses (de régularité, et de dimension modérée), mais via les techniques récentes de "scrambling" (introduisant un peu d'aléatoire dans les très déterministes suites quasi-aléatoires) combinent les avantages de robustesse (grande dimension, et peu d'hypothèses de régularité) de la méthode de Monte-Carlo et les avantages de la méthode quasi-aléatoire (vitesse accrue lorsque les fonctions sont de bon aloi).

Voir la partie « probabilités » pour Monte-Carlo ; pour quasi-Monte-Carlo, on pourra se référer à [3]. Il est aussi à noter que calculer des intégrales peut aussi se faire par des méthodes de billard ou par des Monte-Carlo par chaînes de Markov.

Références

- [1] P.G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Dunod, 1998.
- [2] J.-P. Demailly, *Analyse numérique et équations différentielles*, Presses Universitaires de Grenoble, 1996.
- [3] H. Niederreiter, *Random Number Generation and Quasi-Monte-Carlo Methods*, Society of Industrial and Applied Mathematics, 1992.
- [4] P.S. Toulouse, *Thèmes de probabilité et statistiques*, Dunod, 1999.
- [5] Wikipédia, *L'encyclopédie Libre*, Wikipédia, Wikipédia Fondation.
- [6] A. Pommellet, *Cours d'analyse*, Ellipses 1994.
- [7] H. Brézis, *Analyse fonctionnelle*, Masson, 1983.
- [8] T. Baeck, *Evolutionary Algorithms in theory and practice*, Oxford Univ. Press, 1995.
- [9] G. Demange, J.-C. Rochet, *Méthodes mathématiques de la finance*, Economica, 2ème édition, 1997.