

Projet MMSN - Interpolation VS Développement limité

Quentin GAROT

Avril 2024

Table des matières

| | | |
|----------|--|-----------|
| 1 | Cadre général | 3 |
| 1.1 | Interpolation polynomiale d'une fonction | 3 |
| 1.2 | Développement limité et série entière | 3 |
| 2 | Approximation par une méthode d'interpolation polynomiale | 4 |
| 2.1 | Méthode de Neville-Aitken | 5 |
| 2.2 | Pseudo-code et complexité | 5 |
| 2.3 | Choix des points d'interpolation | 6 |
| 2.3.1 | Problème observé - phénomène de Runge | 6 |
| 2.3.2 | Les meilleurs points d'interpolation | 7 |
| 2.4 | Résultats numériques | 9 |
| 3 | Approximation par les développements limités | 10 |
| 3.1 | Condition suffisante pour qu'une fonction soit développable en série entière | 11 |
| 3.2 | Étude sur le disque ouvert de convergence | 12 |
| 3.3 | Calcul du développement limité | 12 |
| 3.3.1 | Approche naïve | 12 |
| 3.3.2 | Algorithme de Horner | 12 |
| 3.4 | Résultats numériques | 13 |
| 3.4.1 | Rayon de convergence $R = +\infty$ | 13 |
| 3.4.2 | Rayon de convergence $R = 1$ | 14 |
| 4 | Comparaison des méthodes | 15 |
| 4.1 | Démarche générale | 15 |
| 4.2 | Résultats numériques | 15 |
| 4.2.1 | Convergence et précision | 15 |
| 4.2.2 | Stabilité | 16 |
| 5 | Conclusion | 19 |

1 Cadre général

Dans ce projet, nous nous intéresserons à l'approximation d'une fonction à valeurs réelles notée f de classe \mathcal{C}^∞ . Nous proposerons alors dans un premier temps une solution à l'aide de l'interpolation lagrangienne qui, découverte par Edward Waring en 1779, permet d'interpoler une série de points par un polynôme qui passe exactement par ces points (appelés aussi nœuds). Nous verrons également une autre approche qui est l'utilisation de développements limités. L'objectif sera de comparer et évaluer ces deux méthodes que nous introduirons dans les deux paragraphes suivants.

1.1 Interpolation polynomiale d'une fonction

Le problème d'interpolation intervient dans plusieurs contextes comme la reconstitution de la topographie d'un fond marin à l'aide de données ponctuelles, le partage de clés secrètes de Shamir en cryptographie, le calcul numérique d'une intégrale ou encore l'approximation polynomiale de fonctions. Commençons par énoncer formellement une présentation générale du problème.

Proposition 1 (Solution du problème d'interpolation) *Soient $(a_i)_{i \in \llbracket 0, n \rrbracket}$ des réels deux à deux distincts et $(b_i)_{i \in \llbracket 0, n \rrbracket}$ des réels quelconques. Alors il existe un unique polynôme P tel que $\deg(P) \leq n$ et $\forall i \in \llbracket 0, n \rrbracket, P(a_i) = b_i$.*

Ce résultat nous permet de définir une première façon d'approximer une fonction f , qui consiste à réaliser une interpolation.

Définition 1 (Polynôme d'interpolation de Lagrange) *Soient $(x_i)_{i \in \llbracket 0, n \rrbracket}$ des réels deux à deux distincts et f une fonction à valeurs réelles telle que $\forall i \in \llbracket 0, n \rrbracket, f(x_i)$ soit bien définie. On appelle polynôme d'interpolation de Lagrange d'ordre n de la fonction f , noté $P_n f$, le polynôme solution du problème d'interpolation aux points $(x_i, f(x_i))_{i \in \llbracket 0, n \rrbracket}$.*

Nous confondrons par la suite le polynôme $P_n f$ et la fonction polynomiale $x \mapsto P_n f(x)$. Soulignons qu'il est plus pratique de manipuler et stocker un polynôme plutôt qu'une fonction. Numériquement, nous trouvons donc intérêts à remplacer une fonction par son polynôme d'interpolation de Lagrange.

1.2 Développement limité et série entière

Un développement limité (noté DL) d'une fonction en un point est une approximation polynomiale de cette fonction au voisinage de ce point. Les développements limités trouvent leur intérêt notamment en physique puisqu'il est fréquent de confondre la fonction avec sa partie régulière, à condition que l'erreur ainsi faite soit suffisamment petite. Dans ce paragraphe, nous justifierons l'assimilation entre développements limités et séries entières. Rappelons une définition plus formelle.

Définition 2 (Développement limité) Soient I un intervalle de \mathbb{R} , $x_0 \in I$ et f une fonction définie sur I (sauf peut-être en x_0) à valeurs réelles. La fonction f admet un développement limité à l'ordre n en x_0 , noté $DL_n(x_0)$, lorsqu'il existe des réels a_0, \dots, a_n et une fonction ϵ définie sur le même domaine que f à valeurs réelles tels que $\lim_{x \rightarrow x_0} \epsilon(x) = 0$ et

$$\forall x \in \mathcal{D}_f, f(x) = \sum_{i=0}^n a_i (x - x_0)^i + (x - x_0)^n \epsilon(x)$$

Pour une fonction régulière, nous pouvons exprimer son développement limité à l'aide de ses dérivées successives. Le théorème qui suit permet de déterminer les développements limités des fonctions usuelles qui seront étudiées par la suite.

Théorème 1 (Formule de Taylor-Young) Soit f une fonction n -fois dérivable sur un intervalle I de \mathbb{R} contenant x_0 . Alors f admet un développement limité à l'ordre n en x_0 qui s'écrit :

$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k + o((x - x_0)^n)$$

Ce résultat ressemble assez au théorème suivant, portant sur l'expression des coefficients d'une série entière, dans le sens où ces derniers peuvent aussi être exprimés à l'aide de dérivées successives.

Théorème 2 (Coefficients d'une série entière) Soit $\sum a_n x^n$ une série entière de la variable réelle de rayon de convergence $R > 0$ et de somme S . Alors $S \in \mathcal{C}^\infty(]-R, R[)$ et $\forall n \in \mathbb{N}$, $a_n = \frac{S^{(n)}(0)}{n!}$.

Ce théorème permet de faire un rapprochement entre série entière et développement limité. En effet, si f est une fonction développable en série entière sur $]-r, r[$, alors elle est de classe \mathcal{C}^∞ sur $]-r, r[$ et la série entière est sa série de Taylor :

$$\forall x \in]-r, r[, f(x) = \sum_{n=0}^{+\infty} \frac{f^{(n)}(0)}{n!} x^n$$

Dans la troisième partie, nous nous intéresserons notamment à des fonctions développables en séries entières puisqu'elles possèdent une propriété intéressante sur leur disque ouvert de convergence.

2 Approximation par une méthode d'interpolation polynomiale

Pour cette partie, fixons $R > 0$ et $f : [-R, R] \rightarrow \mathbb{R}$ une fonction de classe \mathcal{C}^∞ que nous cherchons à interpoler. Soient alors $(x_i)_{i \in [0, n]}$ $(n+1)$ points deux à deux distincts et $P_n f$ le polynôme d'interpolation de Lagrange d'ordre n de f associé à ses points. L'objectif est de tracer d'une part la représentation graphique de f et d'autre part celle de $P_n f$ afin d'avoir une idée de la qualité de l'approximation.

2.1 Méthode de Neville-Aitken

Tracer la représentation graphique de $P_n f$ nécessite la connaissance de $P_n f(x)$ pour tout $x \in [-R, R]$. *A priori*, seules les valeurs $(P_n f(x_i))_{i \in \llbracket 0, n \rrbracket}$ sont connues puisqu'elles correspondent aux nœuds choisis. L'algorithme de Neville-Aitken, dû à Eric Harold Neville, est une méthode récursive du calcul de la valeur du polynôme d'interpolation en un point quelconque. C'est donc grâce à cette méthode numérique que nous serons en mesure de représenter graphiquement le polynôme d'interpolation de Lagrange de f .

Pour expliquer l'algorithme sans perte de généralité, reprenons les notations de la première proposition. Fixons x un réel quelconque et pour $k \in \llbracket 0, n \rrbracket$, $i \in \llbracket 0, n - k \rrbracket$, nous construisons T_k^i le polynôme de degré k tel que $\forall j \in \llbracket i, i + k \rrbracket$, $T_k^i(a_j) = b_j$:

$$\begin{cases} \forall i \in \llbracket 0, n \rrbracket, T_0^i(x) = b_i & (1) \\ \forall k \in \llbracket 1, n \rrbracket, \forall i \in \llbracket 0, n - k \rrbracket, T_k^i(x) = \frac{(x_{i+k} - x)T_{k-1}^i(x) - (x_i - x)T_{k-1}^{i+1}(x)}{x_{i+k} - x_i} & (2) \end{cases}$$

Si nous représentons les valeurs $(T_k^i)_{k, i \in \llbracket 0, n \rrbracket \times \llbracket 0, n - k \rrbracket}$ sous forme de tableau, nous remarquons que l'ordre des calculs peut se faire colonne par colonne. Nous initialisons d'abord la première à l'aide de la relation (1), puis, avec la relation (2), nous calculons chaque terme $T_k^i(x)$ des colonnes suivantes en fonction des termes $T_{k-1}^i(x)$ (colonne précédente, même ligne) et $T_{k-1}^{i+1}(x)$ (colonne précédente, ligne suivante) préalablement calculés.

$$\begin{array}{c|cccc} (a_0, b_0) & T_0^0(x) & T_1^0(x) & \cdots & T_n^0(x) \\ (a_1, b_1) & T_0^1(x) & \cdots & & T_{n-1}^1(x) \\ \vdots & \vdots & \ddots & & \\ (a_n, b_n) & T_0^n(x) & & & \end{array}$$

FIGURE 1 – Organisation des calculs de la méthode de Neville-Aitken

La dernière valeur calculée par l'algorithme est $T_n^0(x)$. Or, rappelons que par définition, T_n^0 est le polynôme de degré n tel que $\forall j \in \llbracket 0, n \rrbracket$, $T_n^0(a_j) = b_j$, donc $T_n^0 = P$ d'où $T_n^0(x) = P(x)$. Nous disposons donc d'un algorithme permettant d'évaluer le polynôme d'interpolation en un point quelconque.

2.2 Pseudo-code et complexité

La méthode de Neville-Aitken nécessite la connaissance d'un jeu de points $((a_i, b_i))_{i \in \llbracket 0, n \rrbracket}$ et d'un point x que nous notons respectivement **set** et **x**. Elle peut être implémentée de la façon suivante :

Algorithm 1 Neville-Aitken

Require: `set` : interpolation data, `x` : real

Ensure: $P(x)$

```
T ← matrix(n + 1, n + 1)           ▷ Initialisation du tableau de calcul
for i ∈ [0, n] do                   ▷ Initialisation de la première colonne - relation (1)
    T[i, 0] ← bi
end for
for k ∈ [1, n] do                   ▷ Itérations - relation (2)
    for i ∈ [0, n - k] do
        T[i, k] ←  $\frac{(a_{i+k} - x)T[i, k - 1] - (a_i - x)T[i + 1, k - 1]}{a_{i+k} - a_i}$ 
    end for
end for
return T[0, n]
```

L'algorithme réalise le calcul de $\frac{(n+1)(n+2)}{2}$ termes (nombre de termes composant la partie supérieure d'une matrice carrée de taille $n + 1$). Nous déduisons alors une complexité en $O(n^2)$.

2.3 Choix des points d'interpolation

Dans cette section, nous mettrons en avant l'importance du choix des points d'interpolation pour une bonne approximation de la fonction f . La signification de « bonne approximation de f » est assez délicate puisque, l'espace des fonctions n'étant pas de dimension finie, les normes ne sont pas équivalentes. Le polynôme de Lagrange pourrait être une très bonne approximation pour une certaine norme tout en étant une très mauvaise pour une autre. Typiquement, si nous considérons l'espace vectoriel des fonctions $(\mathcal{C}^\infty([-R, R]), \|\cdot\|_1)$, où la semi-norme $\|\cdot\|_1$ est définie par

$$\|g\|_1 = \sum_{i=0}^n |g(x_i)|$$

alors le polynôme d'interpolation de Lagrange peut être vu comme le polynôme minimisant l'erreur d'approximation. Cependant, en considérant $(\mathcal{C}^\infty([-R, R]), \|\cdot\|_\infty)$ ce n'est pas forcément le cas. Pour la suite, nous considérerons cet espace vectoriel normé puisqu'il permettra de discuter sur la convergence uniforme.

2.3.1 Problème observé - phénomène de Runge

Une première approche consisterait à prendre des points équi-répartis.

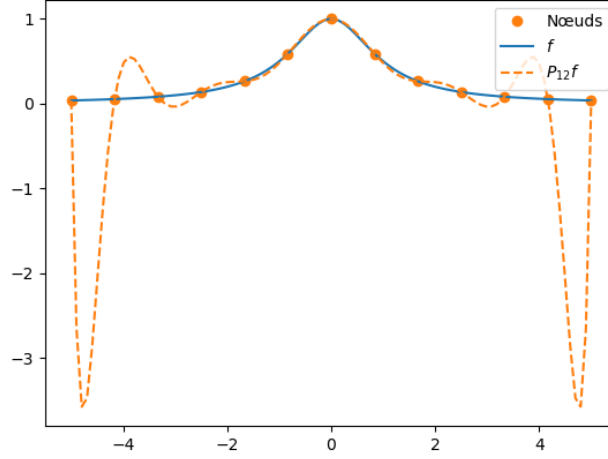


FIGURE 2 – Phénomène de Runge

Pour cet exemple, nous avons représenté la fonction

$$\begin{aligned} f &: [-5, 5] \longrightarrow \mathbb{R} \\ x &\longmapsto \frac{1}{1+x^2} \end{aligned}$$

Runge a montré que l'erreur d'interpolation entre f et $P_n f$ tend vers l'infini lorsque le nombre de nœuds augmente. Plus nous fixons de points où le polynôme a la même valeur que f , moins bien nous approchons la fonction :

$$\lim_{n \rightarrow +\infty} \left(\max_{x \in [-1, 1]} |f(x) - P_n f(x)| \right) = \lim_{n \rightarrow +\infty} \|f - P_n f\|_{\infty} = +\infty$$

Lorsque nous augmentons le nombre de points d'interpolation, nous constatons que le polynôme se met à osciller fortement entre ces derniers avec une amplitude de plus en plus grande.

2.3.2 Les meilleurs points d'interpolation

L'objectif est de minimiser l'erreur d'interpolation $f - P_n f$. Le théorème suivant nous permet déjà de connaître son expression théorique.

Théorème 3 (Erreur d'interpolation) Soient $f \in \mathcal{C}^{n+1}$ et $(x_i)_{i \in \llbracket 0, n \rrbracket}$ des points distincts de $[a, b]$. Alors $\forall x \in [a, b], \exists \xi(x) \in [\min(x_0, \dots, x_n, x), \max(x_0, \dots, x_n, x)]$ tel que

$$f(x) - P_n f(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{j=0}^n (x - x_j)$$

Nous pouvons donc majorer la norme infinie de l'erreur d'interpolation :

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \left\| \prod_{j=0}^n (\cdot - x_j) \right\|_\infty$$

Travaillons ici sur l'intervalle $[-1, 1]$ que nous pouvons dilater si besoin. Considérons l'ensemble des fonctions polynomiales unitaires de degré $n+1$ à $n+1$ racines réelles deux à deux distinctes dans $[-1, 1]$ (scindé à racines simples dans $[-1, 1]$), que nous notons \mathcal{P}_{n+1} . Ainsi, trouver les points $(x_j)_{j \in \llbracket 0, n \rrbracket}$ permettant de minimiser le terme

$$\left\| \prod_{j=0}^n (\cdot - x_j) \right\|_\infty$$

est équivalent à trouver une fonction $\phi \in \mathcal{P}_{n+1}$ telle que $\|\phi\|_\infty$ est minimale. Le théorème qui suit nous permet de trouver une telle fonction ϕ .

Théorème 4 Soit T_{n+1} la fonction polynomiale associée au polynôme de Tchebychev de degré $n+1$. Alors la fonction $\frac{1}{2^n} T_{n+1}$ vérifie

$$\frac{1}{2^n} T_{n+1} \in \mathcal{P}_{n+1} \text{ et } \left\| \frac{1}{2^n} T_{n+1} \right\|_\infty = \min_{\phi \in \mathcal{P}_{n+1}} \|\phi\|_\infty$$

D'après l'équivalence précédente, les racines de $\frac{1}{2^n} T_{n+1}$, que sont $(\cos(\frac{2j+1}{2n+2}\pi))_{j \in \llbracket 0, n \rrbracket}$, fournissent des points d'interpolation permettant de minimiser le majorant de l'erreur trouvé précédemment. En choisissant de tels points d'interpolation, nous obtenons :

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{2^n (n+1)!}$$

Bien que ces points garantissent une minimisation de l'erreur d'approximation, ils ne garantissent pas forcément la convergence uniforme de la suite de fonctions $(P_n f)_{n \in \mathbb{N}}$ vers f . Considérons l'ensemble de fonctions suivant :

$$\mathcal{B} = \{f \in \mathcal{C}^\infty([-1, 1]) \mid \exists \gamma > 0, \forall n \in \mathbb{N}, \forall x \in [-1, 1], |f^{(n)}(x)| < \gamma\}$$

qui contient par exemple les fonctions trigonométriques \sin , \cos et la fonction \exp . Soit $f \in \mathcal{B}$. Par définition, il existe une constante strictement positive γ qui majore toutes les dérivées successives sur l'intervalle $[-1, 1]$. Nous obtenons alors quelque soit $n \in \mathbb{N}$:

$$\|f - P_n f\|_{\infty, [-1, 1]} \leq \frac{\gamma}{2^n (n+1)!}$$

La limite suivante

$$\frac{\gamma}{2^n (n+1)!} \xrightarrow{n \rightarrow +\infty} 0$$

nous permet d'affirmer que $(P_n f)_{n \in \mathbb{N}}$ converge uniformément vers f sur $[-1, 1]$. Cet ensemble de fonctions pour lequel il y a convergence uniforme requière cependant de fortes hypothèses sur toutes les dérivées successives, ce qui, en pratique, n'est pas évident à montrer.

Pour une interpolation lagrangienne d'ordre n , nous choisirons par la suite les points d'abscisses correspondant aux racines de $\frac{1}{2^n} T_{n+1}$ afin de minimiser l'erreur d'approximation. Notons qu'en pratique nous ne sommes pas limités à l'intervalle $[-1, 1]$, nous pouvons effectuer les tests numériques sur des intervalles plus grands en dilatant l'intervalle $[-1, 1]$.

2.4 Résultats numériques

Voici ci-dessous quelques graphiques représentant des approximations de fonctions régulières.

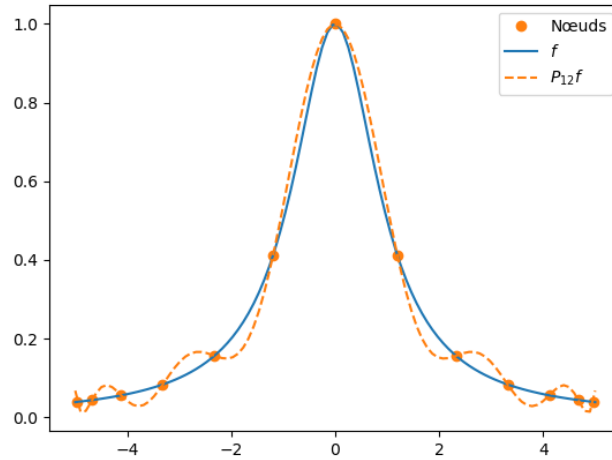


FIGURE 3 – Interpolation polynomiale d'ordre 12 de la fonction de Runge sur le segment $[-5, 5]$

Remarquons que cette approximation est nettement meilleure que celle illustrée sur la figure 2. En effet, l'erreur maximale commise ici est beaucoup plus petite. En augmentant davantage le nombre de points d'interpolation, la courbe d'approximation se confond quasiment avec celle de la fonction f :

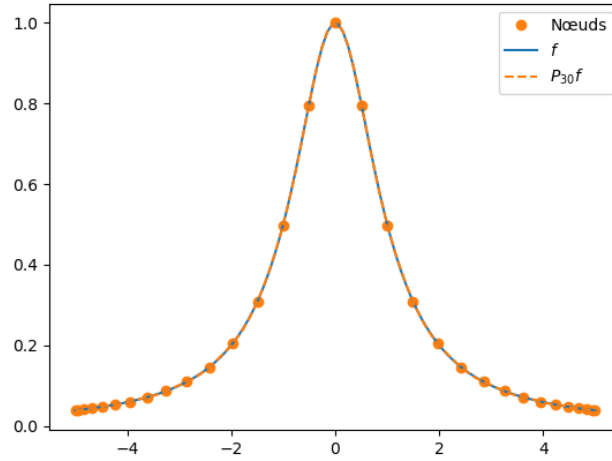


FIGURE 4 – Interpolation polynomiale d'ordre 30 de la fonction de Runge sur le segment $[-5, 5]$

Illustrons le bon fonctionnement de l'algorithme en approximant la fonction cosinus.

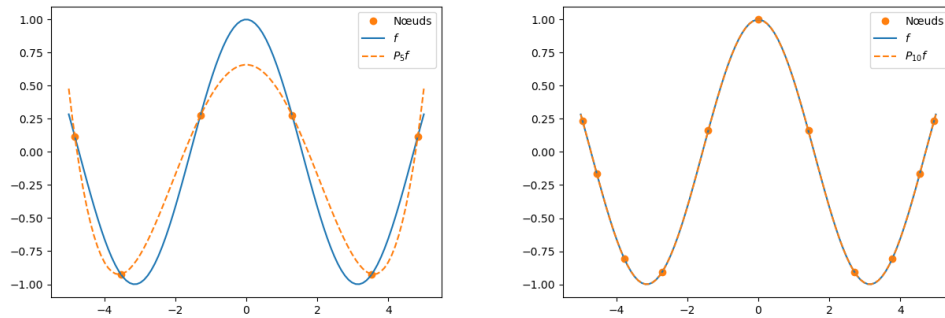


FIGURE 5 – Interpolations lagrangiennes de la fonction cosinus aux ordres 5 et 10 sur l'intervalle $[-5, 5]$

Les résultats ci-dessous donnent alors un aperçu de la convergence de l'erreur commise que nous comparerons plus loin avec celle des développements limités.

3 Approximation par les développements limités

Rappelons que l'objectif de ce projet est de comparer les qualités d'approximation d'une fonction selon les approches d'interpolation lagrangienne et de développement limité. Ce-

pendant, l'interpolation lagrangienne peut se faire à un ordre quelconque (autrement dit nous pouvons choisir autant de points que nous souhaitons pour approximer la fonction étudiée). Nous aimerions qu'il en soit de même pour la méthode d'approximation par développement limité : rappelons que par hypothèse, nous nous intéressons à des fonctions de classe \mathcal{C}^∞ qui, d'après le théorème 1 (formule de Taylor-Young), admettent des développements limités à n'importe quel ordre en tout point. On pourra ainsi comparer l'évolution de l'erreur d'approximation en fonction de l'ordre des deux méthodes.

Un développement limité est une notion locale. Cela signifie que l'approximation de la fonction se fait en un point sur un certain voisinage sans garantie qu'elle soit de bonne qualité sur l'ensemble de définition. Mais comme annoncé en fin de partie 1.2, nous allons principalement nous intéresser aux fonctions développables en séries entières puisqu'elles nous permettront d'établir un résultat de convergence uniforme intéressant sur des intervalles bien définis.

3.1 Condition suffisante pour qu'une fonction soit développable en série entière

Définissons un ensemble presque similaire à l'ensemble \mathcal{B} introduit précédemment :

$$\tilde{\mathcal{B}}_R = \{f \in \mathcal{C}^\infty(]-R, R[) \mid \exists \gamma > 0, \forall n \in \mathbb{N}, \forall x \in [-R, R], |f^{(n)}(x)| < \gamma\}$$

Soient $f \in \tilde{\mathcal{B}}_R$, $n \in \mathbb{N}$ et $x \in]-R, R[$. Alors f est bien à valeurs réelles, $(n+1)$ fois dérivable sur $] - R, R[$ et donc d'après la formule de Taylor-Lagrange, il existe $\xi \in]0, x[$ tel que :

$$f(x) - \sum_{k=0}^n \frac{f^{(k)}(0)}{k!} x^k = \frac{f^{(n+1)}(\xi) x^{n+1}}{(n+1)!}$$

Or, $f \in \tilde{\mathcal{B}}$ donc il existe $\gamma > 0$ qui majore toutes les dérivées successives sur $] - R, R[$. Nous obtenons ainsi la majoration suivante :

$$\left| f(x) - \sum_{k=0}^n \frac{f^{(k)}(0)}{k!} x^k \right| \leq \frac{\gamma |x|^{n+1}}{(n+1)!} \xrightarrow{n \rightarrow +\infty} 0$$

La série $\sum \frac{f^{(k)}(0)}{k!} x^k$ converge donc et sa somme peut s'exprimer ainsi :

$$f(x) = \sum_{k=0}^{+\infty} \frac{f^{(k)}(0)}{k!} x^k$$

Ce résultat étant vrai quelque soit $x \in]-R, R[$, f est donc par définition développable en série entière sur un rayon de convergence au moins supérieur à R . En résumé, nous savons que si $f \in \tilde{\mathcal{B}}_R$, alors nécessairement f est développable en série entière. Pour remarque, ce n'est pas parce qu'une fonction est développable en série entière qu'elle est nécessairement dans cet ensemble.

3.2 Étude sur le disque ouvert de convergence

Théorème 5 (Convergence normale) Soit $\sum a_n x^n$ une série entière de la variable réelle de rayon de convergence R . Alors pour tout $r \in [0, R[$, la série entière converge normalement sur $\mathcal{D}_f(0, r)$ le disque fermé de centre 0 et de rayon r .

Ainsi, comme la convergence normale implique la convergence uniforme, pour une fonction f développable en série entière notée $\sum a_n x^n$, quelque soit $r > 0$ dans $[0, R[$, la suite $(\sum_{k=0}^n a_n(\cdot)^k)_{n \in \mathbb{N}}$ converge uniformément sur $[-r, r]$ vers f .

Il faut cependant faire attention au fait que nous n'avons pas nécessairement convergence normale (ou même uniforme) sur le disque ouvert de convergence $\mathcal{D}_O(0, R)$ comme c'est le cas pour la série entière $\sum (-1)^n x^{2n}$ que nous illustrerons graphiquement plus loin.

3.3 Calcul du développement limité

Pour tracer la représentation graphique de l'approximation par développement limité (ou série entière), nous devons être en mesure de l'évaluer en tout point de l'intervalle sur lequel l'étude est faite. L'approximation étant polynomiale, nous devons établir un algorithme d'évaluation de polynôme dont l'expression de ses coefficients est connue.

3.3.1 Approche naïve

Une première approche simple consisterait à calculer l'évaluation polynomiale comme suit :

Algorithm 2 Polynomial evaluation

Require: \mathbf{a} : coefficients of P , x : real

Ensure: $P(x)$

$s \leftarrow 0$

for $i \in \llbracket 0, \text{length}(\mathbf{a}) - 1 \rrbracket$ **do**

$s \leftarrow s + a_i x^i$

end for

return s

Le nombre de multiplications à réaliser est ici en $O(n^2)$, ce qui représente un coût de calcul assez élevé.

3.3.2 Algorithme de Horner

L'algorithme de Horner est un algorithme d'évaluation polynomiale réduisant le nombre d'opérations à réaliser par rapport à l'approche classique. L'idée est d'écrire le polynôme en question P de degré n de la façon suivante :

$$P = (\cdots ((a_n X + a_{n-1}) X + a_{n-2}) \cdots) X + a_0$$

L'algorithme de Horner peut alors s'implémenter ainsi :

Algorithm 3 Horner

Require: $\mathbf{a} : n + 1$ coefficients of P , x : real

Ensure: $P(x)$

```

 $s \leftarrow a_n$ 
for  $i \leftarrow n - 1, \dots, 0$  do
     $s \leftarrow sx + a_j$ 
end for
return  $s$ 

```

Nous remarquons que le nombre de multiplications à effectuer ici est en $O(n)$ seulement, d'où le choix de cet algorithme pour la suite.

3.4 Résultats numériques

Dans cette section, nous présenterons des graphiques d'approximations de quelques fonctions régulières.

3.4.1 Rayon de convergence $R = +\infty$

Contrairement au développement limité qui est une propriété locale, le développement en série entière est une propriété globale. Cela se remarque plutôt bien pour des fonctions développables en séries entières dont le rayon de convergence est infini. Dans ce cas, le théorème 5 affirme que la série entière converge uniformément sur tout segment de \mathbb{R} , ce qui explique pourquoi, à un ordre assez élevé, l'approximation est très précise en tout point.

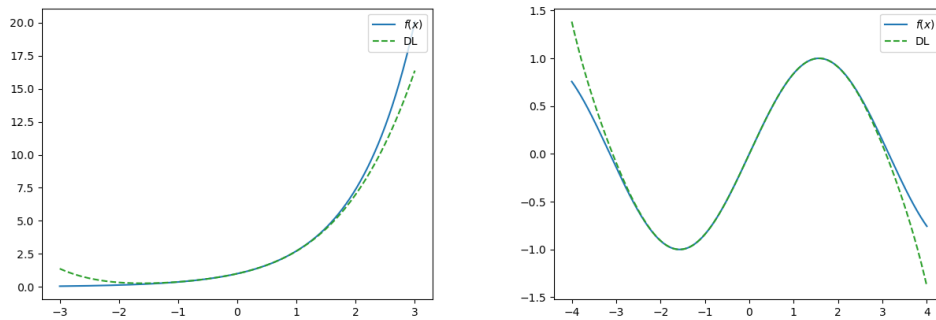


FIGURE 6 – Approximations par séries entières de la fonction \exp à l'ordre 4 sur $[-3, 3]$ à gauche et de la fonction \sin à l'ordre 8 sur $[-4, 4]$ à droite

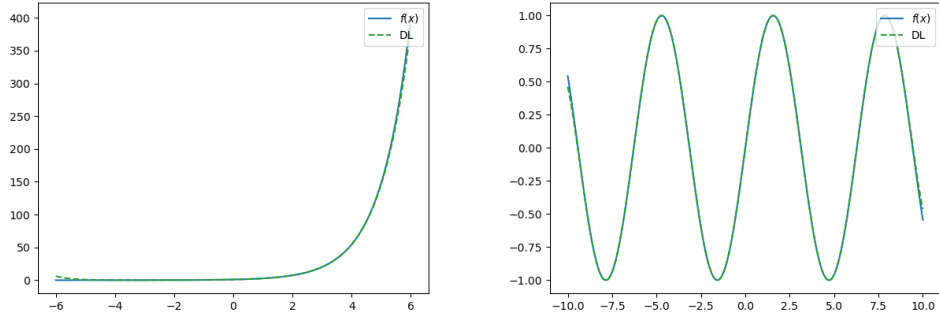


FIGURE 7 – Approximations par séries entières de la fonction \exp à l'ordre 10 sur $[-6, 6]$ à gauche et de la fonction \sin à l'ordre 25 sur $[-10, 10]$ à droite

3.4.2 Rayon de converge $R = 1$

Intéressons nous ici à la fonction

$$f : x \mapsto \frac{1}{1 + x^2}$$

développable en série entière sur $] -1, 1[$:

$$\forall x \in] -1, 1[, f(x) = \sum_{n=0}^{+\infty} (-1)^n x^{2n}$$

Étudions d'une part son approximation sur le disque ouvert de convergence et d'autre part sur un disque fermé quelconque compris dans le disque ouvert de convergence.

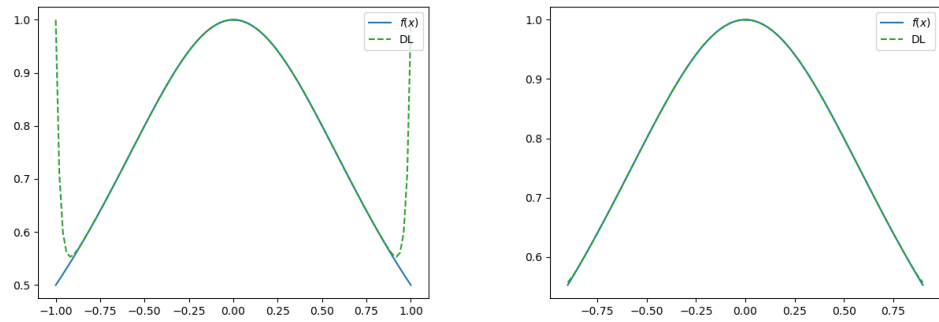


FIGURE 8 – Approximations par série entière de la fonction f à l'ordre 45 sur $] -1, 1[$ à gauche et sur le segment $[-0.9, 0.9]$ à droite

La remarque faite en fin de paragraphe partie 3.2 prend donc tout son sens : pour une approximation contrôlable, il faut faire attention au domaine d'étude et préférer se placer sur un segment inclu dans l'intervalle de convergence.

4 Comparaison des méthodes

4.1 Démarche générale

Pour une fonction $f \in \mathcal{C}^\infty([-r, r])$ et un ordre $N \in \mathbb{N}$ donnés, l'idée est de tracer les courbes représentatives des applications suivantes, définies sur $\llbracket 0, N \rrbracket$:

$$\begin{aligned}\phi_f : n &\mapsto \|f - P_n f\|_{\infty, [-r, r]} \\ \psi_f : n &\mapsto \left\| f - \sum_{k=0}^n \frac{f^{(k)}(0)}{k!} (\cdot)^k \right\|_{\infty, [-r, r]}\end{aligned}$$

Ces dernières nous donneront un aperçu du comportement des erreurs commises (en norme infinie) en fonction de l'ordre d'approximation.

4.2 Résultats numériques

Pour cette section, reprenons des fonctions étudiées dans les parties précédentes.

4.2.1 Convergence et précision

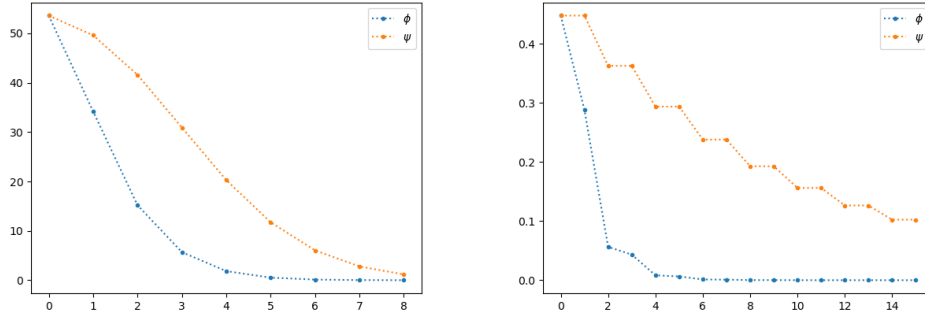


FIGURE 9 – Évolution de l'erreur en fonction de l'ordre pour les fonctions exp à gauche et $x \mapsto 1/(1+x^2)$ à droite

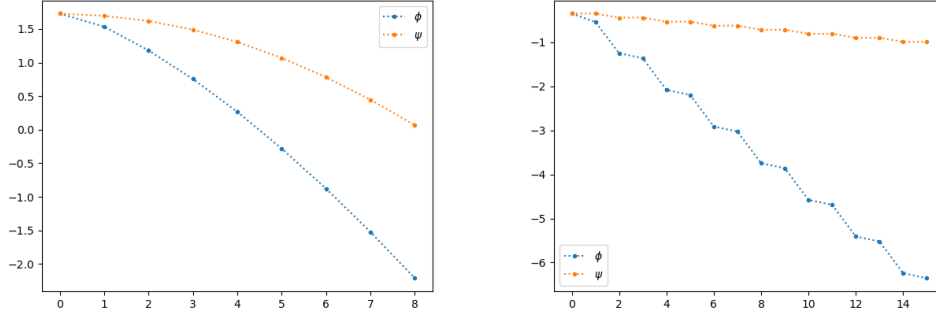


FIGURE 10 – Évolution de l'erreur (en échelle log) en fonction de l'ordre pour les fonctions \exp à gauche et $x \mapsto 1/(1+x^2)$ à droite

De ces premiers résultats, nous remarquons que les deux méthodes convergent bien, ce qui est conforme aux résultats théoriques. Cependant, nous constatons une bien meilleure précision ainsi qu'une convergence uniforme plus rapide pour la méthode d'interpolation lagrangienne.

4.2.2 Stabilité

L'objectif de cette section est de comparer la stabilité des deux méthodes. Pour cela, considérons l'application :

$$\begin{aligned} h &: [-R, R] \longrightarrow \mathbb{R} \\ x &\longmapsto \cos(x)e^x \end{aligned}$$

h est évidemment de classe $\mathcal{C}^\infty([-R, R])$ et nous pouvons montrer que $\forall n \in \mathbb{N}$:

$$\forall x \in [-R, R], h^{(n)}(x) = 2^{\frac{n}{2}} e^x \cos\left(x + \frac{n\pi}{4}\right)$$

Nous ne pouvons pas majorer toutes les dérivées successives par une constante ne dépendant ni de x ni de n à cause du terme $2^{n/2}$. Cependant, pour $x \in [-R, R]$ fixé, la formule de Taylor-Lagrange nous donne l'existence d'un certain ξ tel que :

$$h(x) - \sum_{k=0}^n \frac{h^{(k)}(0)}{k!} x^k = \frac{h^{(n+1)}(\xi) x^{n+1}}{(n+1)!}$$

Nous obtenons donc la majoration suivante :

$$\left| h(x) - \sum_{k=0}^n \frac{h^{(k)}(0)}{k!} x^k \right| \leq \frac{2^{\frac{n}{2}} e^R R^{n+1}}{(n+1)!}$$

Par croissance comparée, la limite

$$\frac{2^{\frac{n}{2}} e^R R^{n+1}}{(n+1)!} \xrightarrow{n \rightarrow +\infty} 0$$

nous permet d'affirmer que h est développable en série entière au moins sur $] -R, R[$. Les graphiques ci-dessous illustrent deux approximations de la fonction h :

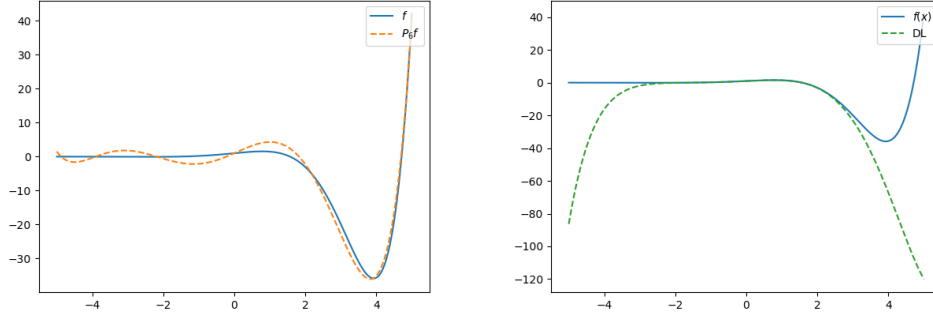


FIGURE 11 – Approximations lagrangienne (à gauche) et par développement en série entière (à droite) de h à l'ordre 7 sur $[-5, 5]$

Pour le même ordre, l'erreur maximale commise pour l'approximation lagrangienne est beaucoup moins importante que celle pour l'approximation par développement en série entière (qui est de l'ordre de 160) : ici, l'approximation par interpolation polynomiale semble globalement meilleure. Pour mieux comprendre, traçons les graphes de ϕ_h et ψ_h :

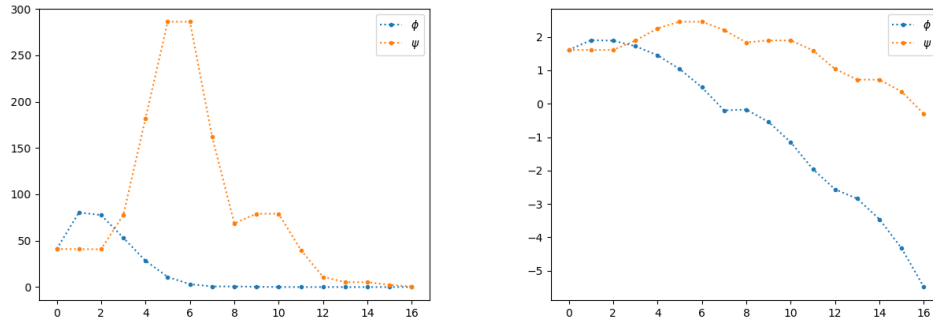


FIGURE 12 – Évolution de l'erreur en fonction de l'ordre pour la fonction h (en échelle log à droite)

Nous remarquons que ψ_h subit une importante variation pour des ordres compris entre 2 et 10 : sur la représentation de gauche, on retrouve l'erreur environ égale à 160 pour une approximation d'ordre 7. De plus, comme h est développable en série entière sur $] -6, 6[$

d'après ce qui a été montré précédemment, alors sa série entière converge uniformément sur le segment $[-5, 5]$ (intervalle sur lequel nous avons représenté les résultats) : malgré une certaine instabilité, les deux graphiques précédents sont conformes à la théorie puisqu'à partir de l'ordre 16, l'approximation semble être de meilleure qualité.

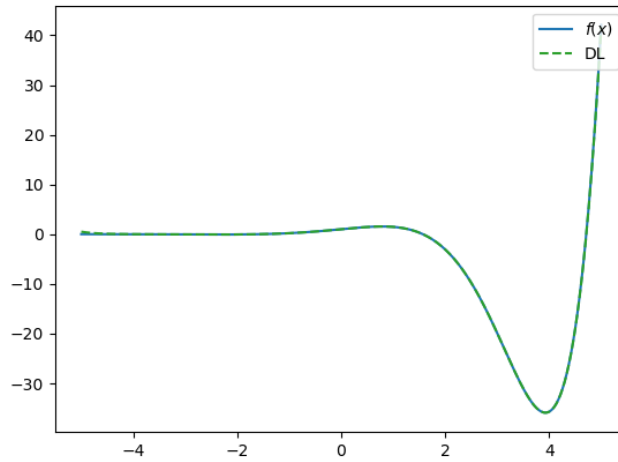


FIGURE 13 – Approximation par série entière de la fonction h à l'ordre 16 sur $[-5, 5]$

Comme le montrent les graphiques qui suivent, nous retrouvons un phénomène similaire pour la fonction \sin :

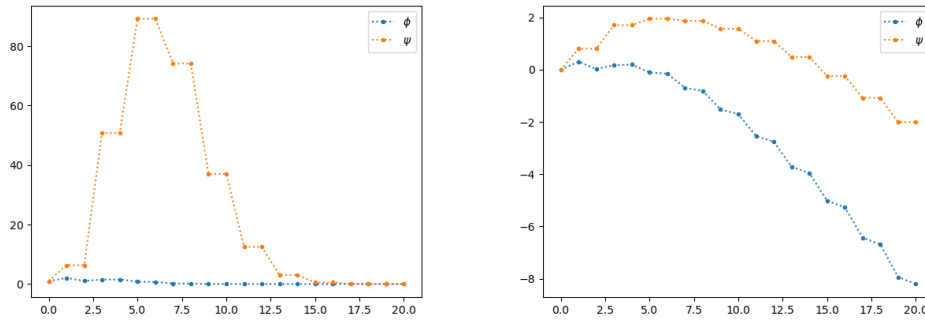


FIGURE 14 – Évolution de l'erreur en fonction de l'ordre pour la fonction \sin (en échelle log à droite)

En effet, pour un certain panel d'ordres, la méthode d'approximation par série entière a du mal à se stabiliser, ce qui n'est pas le cas de la méthode d'approximation lagrangienne qui reste plus précise.

5 Conclusion

En conclusion, ce projet a permis de mettre en évidence l'étude de deux méthodes d'approximation polynomiale de fonctions régulières à valeurs réelles : l'interpolation lagrangienne et les développements limités et/ou en séries entières. Les résultats montrent qu'il existe un ensemble de fonctions pour lequel les deux méthodes convergent uniformément, ce qui signifie qu'il existe toujours un ordre à partir duquel nous pouvons atteindre la précision souhaitée. Ils montrent également qu'en choisissant bien les nœuds et à un ordre donné, la méthode d'interpolation est généralement plus précise et converge plus vite que la méthode de développement en série entière. Cependant, en termes de coût de calculs, la méthode d'interpolation utilise l'algorithme de Neville-Aitken, d'une complexité en $O(n^2)$, tandis que la méthode d'approximation par développements limités utilise l'algorithme de Horner qui est en $O(n)$ seulement. Notons que nous pouvons gagner en complexité puisque nous pouvons remplacer l'algorithme de Horner par l'algorithme de Knuth & Eve, qui est plus rapide, dont l'idée principale consiste à découper le polynôme en une partie paire et une partie impaire, ce qui peut être plus efficace pour certains développements limités. Le choix de ces deux méthodes dépend évidemment de ce qui est souhaité. Pour une approximation d'une fonction au voisinage d'un point, le développement limité est par définition plus adapté et suffisant. Pour une approximation plus globale, le développement en série entière (sous réserve d'existence) et l'interpolation polynomiale sont plus adaptés.