# TP3 : Interpolation polynomiale et Base de Newton

XIA Qiaoqiao et AIT HAMID Adam

## 1 Introduction

Le TP3 a pour objectif d'explorer une variante des méthodes d'interpolation de fonction par un polynôme vues précédemment dans les TP1 et TP2. L'approche ici consiste à travailler dans la base de Newton pour réaliser l'interpolation polynomiale. L'idée est de mettre à jour le polynôme d'interpolation plus efficacement lors de l'ajout de nouveaux points d'interpolation.

## 2 Base de Newton

La base de Newton est une base de polynômes utilisée pour l'interpolation polynomiale. Dans cette base, les polynômes sont définis comme suit :

$$N_k(x) = (x - x_0) \times (x - x_1) \times \ldots \times (x - x_{k-1})$$

pour $k = 0, 1, 2, \ldots, n$, où $x_i$ sont les points d'interpolation.

## 3 Mise en place du système triangulaire

D'après le théorème de la polynôme d'interpolation sur la base de Newton, nous avons :

$$P(x) = a0 + a1N1(x) + a2N2(x) + ... + an \cdot Nn(x)$$

Donc, en substituant xi à la place de x, nous obtenez: $yi = a0 + a1 \cdot N1(xi) + a2 \cdot N2(xi) + ... + an \cdot Ni(xi)$, car pour les N qui se situe supérieur à i, ils devient nulle.
Ainsi le système triangulaire:
$y0 = a0 + a1N1(x0)$
$y1 = a0 + a1N1(x1) + a2N2(x1)$
$y2 = a0 + a1N1(x2) + a2N2(x2) + a3N3(x2)$
$yn = a0 + a1N1(xn) + a2N2(xn) + ... + an \cdot Nn(xn)$

## 4 Différences divisées

Voici un exemple qui nous permet d'utiliser la méthode différences divisées pour trouver les coefficients a:

| x | y | | | | |
|---|---|---|---|---|---|
| -1 | -12 | | | | |
| 1 | 0 | 6 | | | |
| 3 | -20 | -2 | -1 | | |
| 7 | 2724 | 342 | 56 | 14.25 | |
| 10 | 10053 | 915 | 101 | 14.57 | 0.11 |

Pour i = 0 : $y0 = -12$
Pour i = 1 : $\frac{y_1-y_0}{x_1-x_0} = \frac{0-(-12)}{1-(-1)} = 6$
Pour i = 2 : $\frac{y_2-y_0}{x_2-x_0} = \frac{-20-(-12)}{3-(-1)} = -2$ $\frac{y_2-y_1}{x_2-x_1} = \frac{-2-6}{3-1} = -1$
Pour i = 3 : $\frac{y_3-y_0}{x_3-x_0} = \frac{2724-(-12)}{7-(-1)} = 342$ $\frac{y_3-y_1}{x_3-x_1} = \frac{342-6}{7-1} = 56$ $\frac{y_3-y_2}{x_3-x_2} = \frac{56-(-1)}{7-3} = 14,25$
Pour i = 4 : $\frac{y_4-y_0}{x_4-x_0} = \frac{10053-(-12)}{10-(-1)} = 915$ $\frac{y_4-y_1}{x_4-x_1} = \frac{915-6}{10-1} = 101$ $\frac{y_4-y_2}{x_4-x_2} = \frac{101-(-1)}{10-3} = 14.57$ $\frac{y_4-y_3}{x_4-x_3} = \frac{14.57-14,25}{10-7} = 0.11$

Donc, les valeurs soulignés sont des valeurs de coefficients de $a_0, a_1, a_2, a_3, a_4$.

# 5 Implémentation tp

Le code fourni implémente les différentes étapes du TP en Python. Il commence par définir une fonction $f(x)$ qui représente la fonction sinus sur un intervalle. Ensuite, il calcule les différences divisées et utilise la base de Newton pour obtenir le polynôme d'interpolation. Le code permet également d'ajouter un point d'interpolation et de mettre à jour le polynôme d'interpolation en conséquence. Enfin, il affiche graphiquement la fonction originale, le polynôme d'interpolation initial et le polynôme après l'ajout du point d'interpolation.

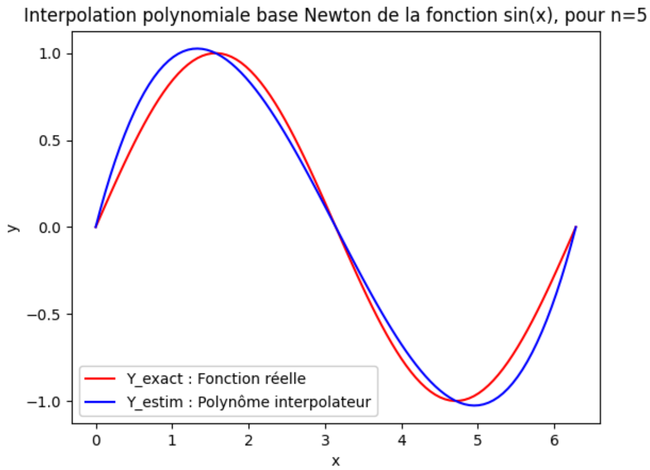


Figure 1: Interpolation de la base de Newton de la fonction $f(x) = \sin(x)$, pour n=5

Nous remarquons également quand on ajoute un point d'interpolation, le polynôme d''interpolation est plus précise et proche de la fonction cherchée.

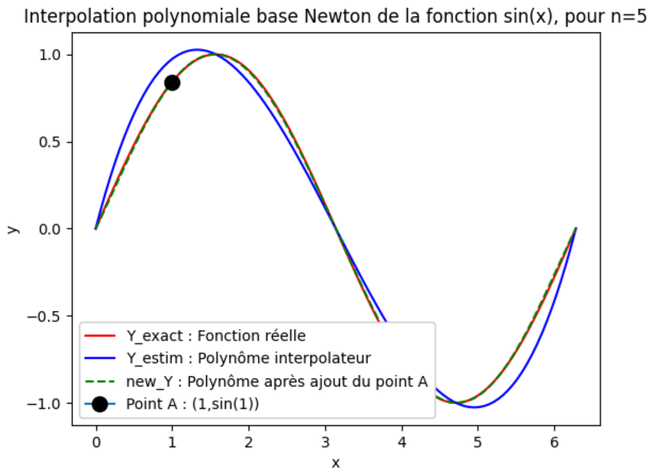


Figure 2: Interpolation de la base de Newton de la fonction après l'ajout du point $(1, sin(1))$ $f(x) = \sin(x)$, pour n=5

# 6 Conclusion

Ce TP a permis d'explorer l'interpolation polynomiale en utilisant la base de Newton et les différences divisées. Il met en évidence l'avantage de travailler dans la base de Newton en termes d'efficacité lors de la mise à jour du polynôme d'interpolation suite à l'ajout de nouveaux points. L'interpolation polynomiale reste un outil puissant pour approximer une fonction sur un intervalle donné en utilisant un polynôme de degré limité.