

Rapport TP 1 - Estimation de paramètres

- [Rapport TP 1 - Estimation de paramètres](#)
 - [Introduction](#)
 - [Structure du code](#)
 - [Méthodes implémentées](#)
 - [Implémentation de moindres_carres](#)
 - [Implémentation de erreur_apprentissage](#)
 - [Implémentation de erreur_generalisation](#)

Introduction

Ce TP se donne pour but d'étudier le processus d'estimation des paramètres d'une courbe de Bézier de degré variable, ainsi que des méthodes permettant le choix de son degré.

Structure du code

Pour des raisons de clarté du code, des choix d'organisation ont été faits afin d'améliorer la structure du projet. Ainsi, la majorité des fonctions fournies avec les sources ont été rassemblées dans la classe [Lib](#).

Voici donc l'API de LIB utilisée dans les fonctions détaillées ci-dessous:

```
Sealed class Lib:
  Static methods:
    Lib : bernstein(x, d, i) # Polynome de Bernstein (cf documentation)
    Lib : bezier(beta_0, beta, beta_d, x) # Courbe de Bézier (f dans le
sujet)
    Lib : gNoise(input, sigma) # Générateur de bruit gaussien additif
    Lib : mean_square_error(input, output) # Erreur quadratique moyenne
```

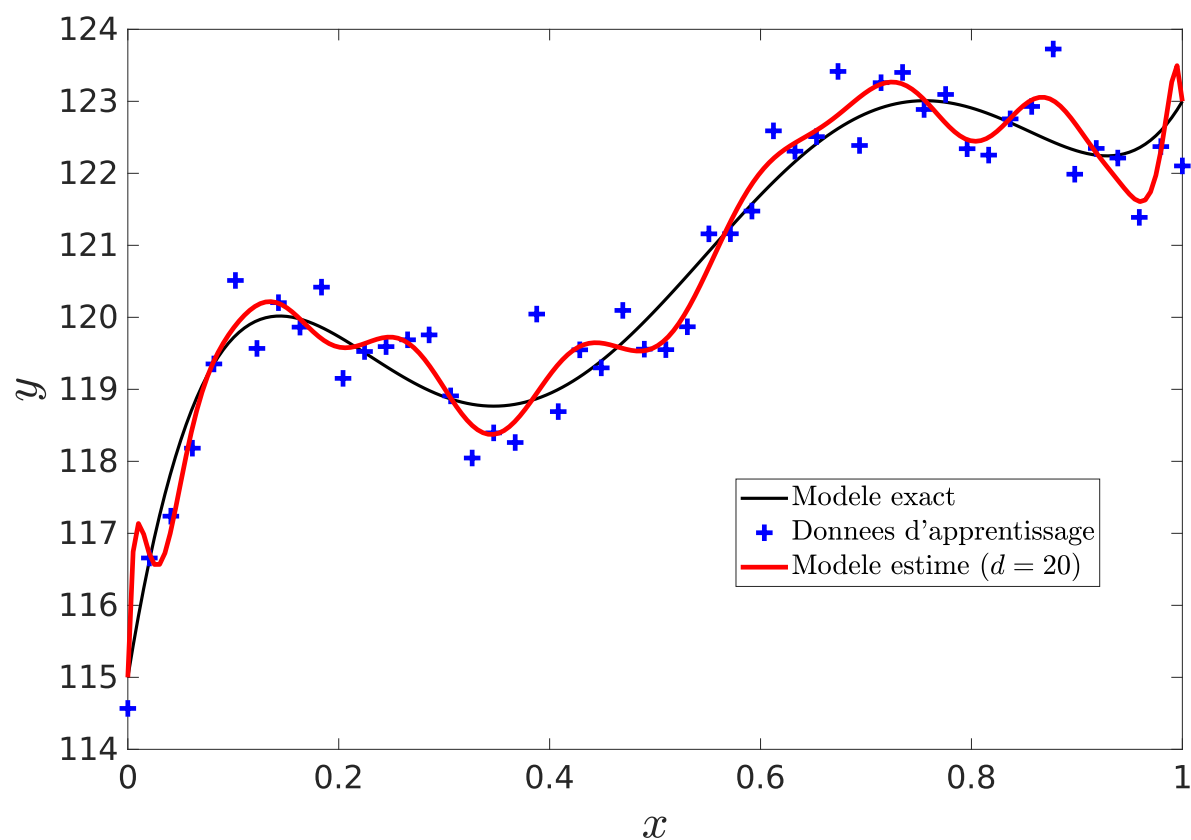
Ainsi qu'un exemple d'utilisation :

```
y = Lib.gNoise(Lib.bezier(beta_0, beta, beta_d, x), sigma);
```

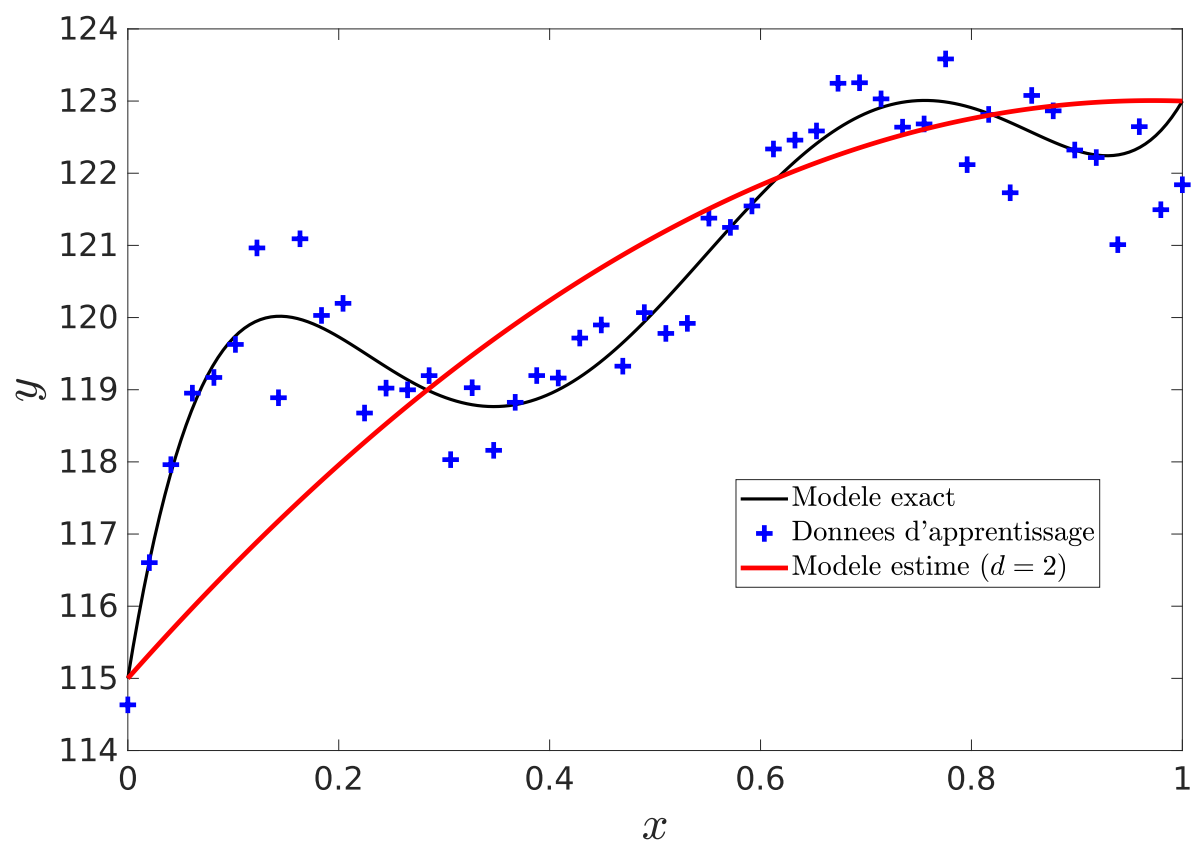
Méthodes implémentées

Implémentation de [moindres_carres](#)

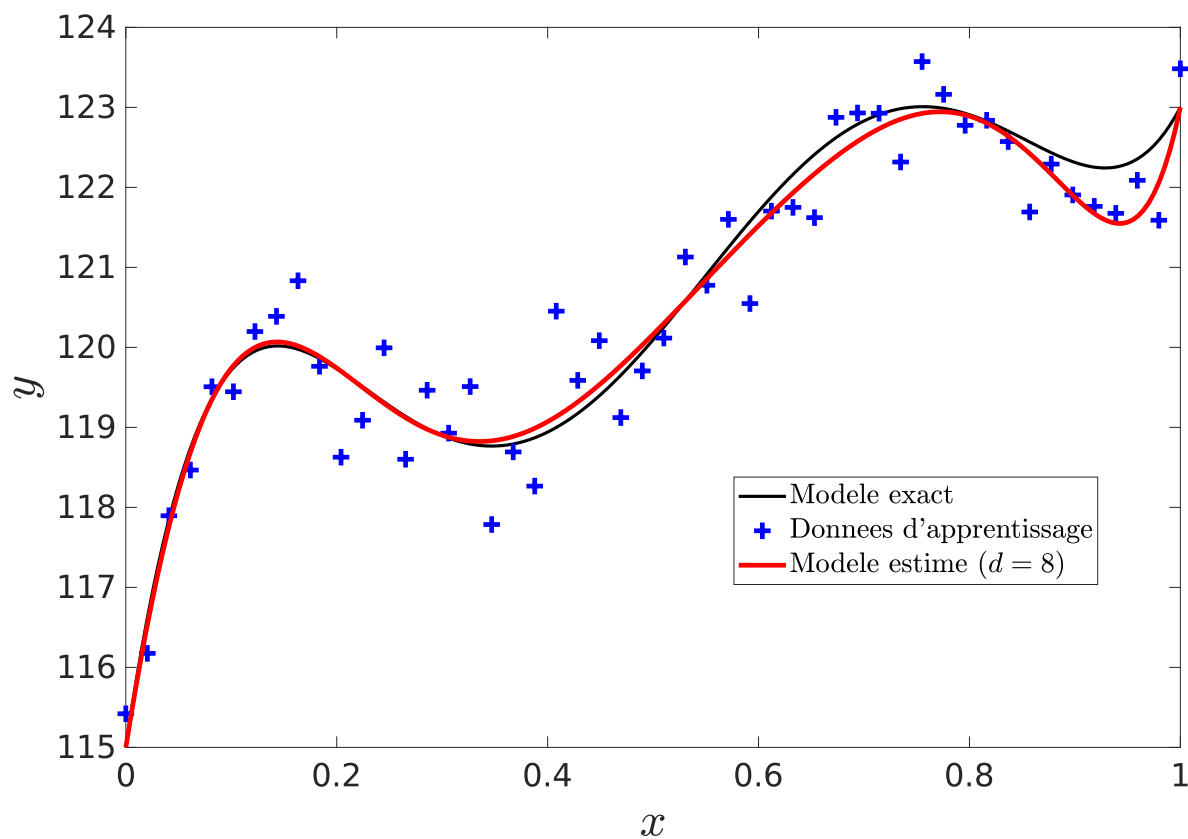
Après exécution du script [exercice_1](#), on obtient une solution approchée de $\hat{\beta}$ qui produit la courbe de Bézier suivante, de degré $d = 20$ choisi au préalable :



La courbe précédente est très clairement biaisée par un surentrainement. Selon la même logique, pour des valeurs de d faibles, on obtient un modèle sous-entraîné, comme le montre le résultat suivant, obtenu pour $d = 2$:



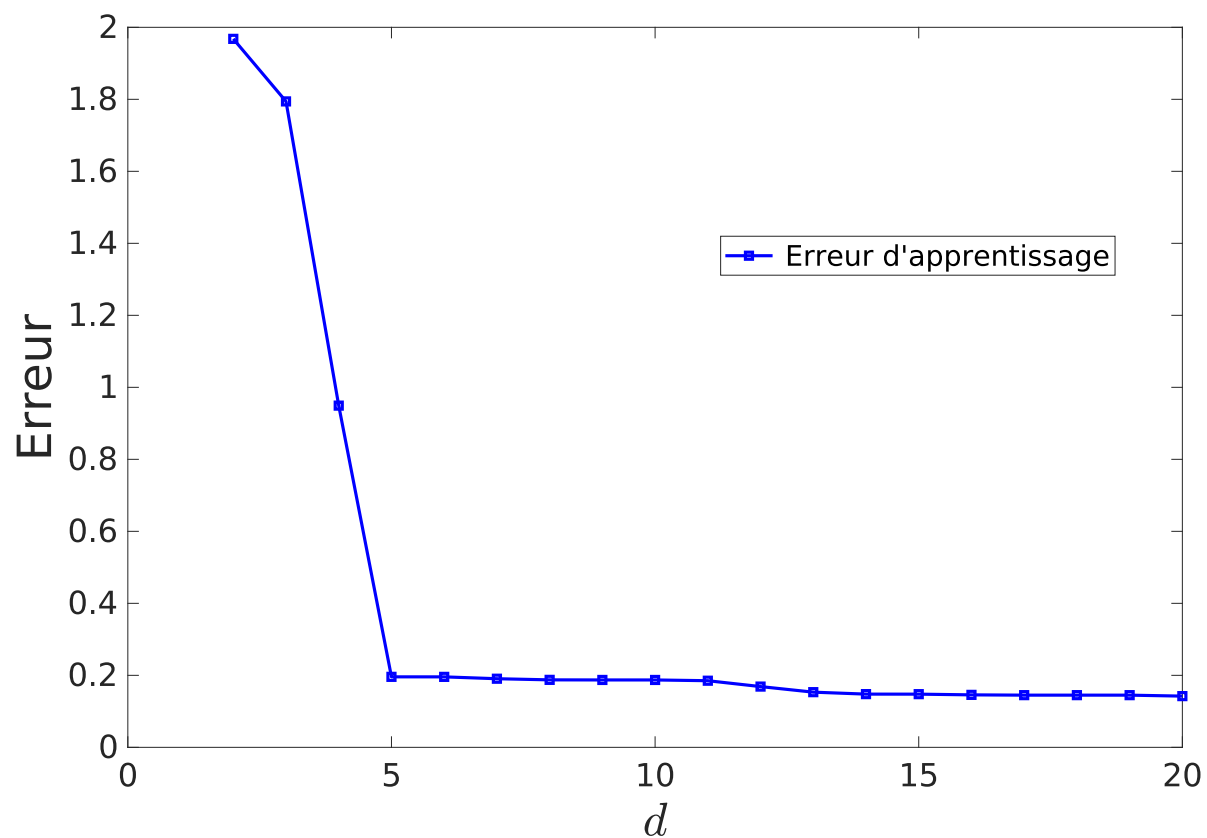
On se rend cependant très rapidement compte de la nécessité de choisir une valeur du degré plus adaptée, comme par exemple $d = 8$, dont la courbe associée est visible ci-dessous :



Ce résultat est bien plus satisfaisant, et semble coller bien mieux à la courbe du modèle exact.

Implémentation de `erreur_apprentissage`

L'exécution du script `exercice_2` permet de visualiser l'évolution de l'erreur quadratique moyenne d'apprentissage en fonction du degré de la courbe de Bézier.



C'est un résultat cohérent, puisque l'erreur décroît en fonction du degré de la courbe de Bézier. Il est cependant pertinent de noter qu'elle varie considérablement moins à partir de $d = 5$.

Implémentation de `erreur_generalisation`