Cahier des charges

Interpolaspline

Stage Applicatif L3 MIN

Année 2019-2020

Table des matières

1 Contexte et définition du problème

1.1 Motivations

Lors d'expériences scientifiques, des données sont acquises. Cependant, ces données sont très souvent discrètes ce qui ne permet pas d'établir facilement une loi (fonction de $\mathbb{R}$ dans $\mathbb{R}^n$ suivie par ces données, n étant la dimension dans laquelle l'interpolation est faite) continue. C'est le principe de l'interpolation.

Cependant, dans la majorité des expériences, une fonction passant par toutes les données oscillerait trop. On trouve donc une fonction minimisant une certaine quantité, liée à la distance entre les données et la fonction établie. Dans ce cas, on parle d'approximation.

Lors des expériences, si une donnée est très éloignée des autres, on peut souhaiter l'ignorer. Cela peut-être une erreur de mesure, un changement de conditions, une grosse imprécision, etc. Ces données sont dites aberrantes, et c'est ce cas qui nous intéresse dans ce projet.

Voici quelques exemples d'utilisation de l'interpolation et de l'approximation :

— Étalonnage d'instruments de mesure, en métrologie
— Découverte de lois à partir d'expériences, dans tous les domaines de la recherche
— Identifier des données incohérentes : une faute de frappe, un produit défectueux

1.2 Problème

L'idée consiste à interpoler des données en tenant compte des points aberrants, trouver une ou plusieurs méthodes pour les détecter et les traiter suivant qu'elles soient uniformes ou non.

Ensuite, introduire la notion de spline de lissage pour aborder le cas des données imprécises en fonction d'un paramètre rho qui permet de contrôler le rapport entre les points et le lissage de la fonction. Trouver une ou plusieurs façons de trouver automatiquement ce paramètre de lissage rho en fonction de nos entrées. Pour commencer, nous allons interpoler des points en utilisant les splines naturelles cubiques C2 qui consiste à faire passer entre chaque point un polynôme de degré inférieur ou égal à 3 et qui soit de classe C2 dans le cas uniforme et non uniforme. Ces données peuvent contenir des valeurs aberrantes, il faut dans un premier temps trouver un moyen d'identifier ces valeurs, les supprimer ou leur attribuer un poids faible, utiliser des méthodes de minimisation pour trouver de nouvelles données et les inclure dans notre spline de lissage en 1D et le cas paramétrique 2D.

2 Objectifs

Une spline naturelle est une fonction définie par morceau par des polynômes cubiques, dont la dérivée seconde est continue, et dont les dérivées secondes aux extrémités de l'intervalle de définition sont nulles. Le but du projet est de pouvoir créer automatiquement une spline de lissage approximant un ensemble de données contenant des points aberrants, à une dimension puis à deux dimensions (splines naturelles paramétriques).

Contenu

Le projet sera livré sous la forme d'une archive contenant :

- Plusieurs fichiers python regroupant les différentes fonctionnalités décrites à la section 4.1 :
  - Une librairie "Splines 1D" (nommée "$Splines1D.py$"), qui regroupe les méthodes de création des splines à une dimension
  - Une librairie "Splines Paramétriques" (nommée "$SplinesParametriques.py$"), qui regroupe les méthodes de création des splines paramétriques
  - Une librairie "Données Aberrantes" (nommée "$DonneesAberrantes.py$"), qui regroupe les méthodes de gestion des points aberrants
  - Une librairie "Automatisation" (nommée "$Automatisation.py$"), qui regroupe les méthodes de configuration automatique des paramètres des splines
  - Un fichier "Interface" (nommé "$Interface.py$"), qui contient le programme principal, qui est le seul que l'utilisateur doit lancer
- Un lisez-moi consignant les dépendances entre les programmes, et les instructions d'utilisation.
- Un rapport expliquant les diverses méthodes, avec des exemples d'utilisations illustrés.

3 Limites

Restriction de la solution proposée (cadrage) :

— Les points donnés en entrée seront défini dans un ordre précis.

4 Description fonctionnelle

4.1 Fonctionnalités

Le programme permettra de créer une approximation de données stockées dans un fichier. L'utilisateur pourra :

- Créer la spline naturelle qui interpole exactement tous les points, en une dimension ou en paramétrique
- Créer la spline de lissage associée aux données, en une dimension ou en paramétrique
- Créer la spline de lissage, en tenant compte des points aberrants
- Choisir la quantité à minimiser dans une liste

4.2 Entrées et sorties

4.2.1 Entrées

Lors du lancement du programme, il sera demandé à l'utilisateur d'entrer le chemin vers le fichier contenant les donnnées à interpoler. Ce fichier doit contenir 2 colonnes (séparées par un espace) et un nombre inconnu de lignes. La première colonne représente les abcisses, et la deuxième les ordonnées. Le fichier peut contenir des commentaires, précédés par un dièse et sans caractères spéciaux.

Exemple :

```
# nom : fichier_de_depart.txt
0 0
1 1
2 6 # Cette valeur est etrange
3 9
4 16
```

Ce fichier donnera en abcisses $\{0, 1, 2, 3\}$, et en ordonnées $\{0, 1, 6, 9\}$.

Puis il reseigne la dimension de ses données (1D ou paramétriques) : EXPLIQUER BLABLABLA

Il choisit ensuite le type de sortie qu'il souhaite, entre une spline exacte et une spline de lissage.

Enfin, l'utilisateur peut demander au programme de gérer les points aberrants, avec l'une des méthodes proposées.

4.2.2 Sorties

À la fin de l'exécution, le programme affichera un graphe représentant les données fournies par l'utilisateur, ainsi que la spline interpolant ces données (spline dépendant des choix de l'utilisateur). La discrétisation de cette spline sera également fournie en sortie, dans le fichier "$nom\_du\_fichier\_de\_depart.res$".

Ce fichier sera au même format que le fichier passé en entrée, avec en première ligne la dimension de la spline ("$1D$" ou "$2D$").

Exemple :

```
# 1D
0 0
1 1
2 4
3 9
4 16
```

5 Organisation temporelle

Dates importantes et livraison :

— Mercredi 18 décembre : présentation du cahier des charges et de l'organisation du projet.

— Jeudi 30 avril : soutenance du projet et rendu des codes.