

Cahier des charges

Interpolaspline

Stage Applicatif L3 MIN

Année 2019-2020

Table des matières

1	Contexte et définition du problème	2
1.1	Motivations	2
1.2	Problème	2
2	Objectifs	3
3	Limites	5
4	Description fonctionnelle	5
4.1	Fonctionnalités	5
4.2	Entrées et sorties	7
4.2.1	Entrées	7
4.2.2	Sorties	8
5	Organisation temporelle	8

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 continue (fonction continue de \mathbb{R} dans \mathbb{R}^n suivie par ces données, n étant la dimension dans laquelle l'interpolation est faite). C'est le principe de l'interpolation.

Cependant, dans la majorité des expériences, une fonction passant par toutes les données oscillerait trop. La minimisation d'une certaine quantité liée à la distance entre les données et la fonction établie permet de créer une spline la plus proche possible des données. 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 :

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

1.2 Problème

Le problème au coeur de notre projet est la création automatique d'une spline de lissage interpolant des données, en minimisant l'impact des données aberrantes sur la spline. En effet nous savons créer une spline de lissage, mais celle-ci tient compte de toutes les données de la même manière et nécessite un paramètre de lissage explicite. Notre projet se découpe donc en deux sous-problèmes : l'estimation automatique d'un paramètre de lissage en fonction des données, et la gestion des données aberrantes.

2 Objectifs

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

Ensuite, on introduira la notion de spline de lissage pour aborder le cas des données imprécises en fonction d'un paramètre qui permet de contrôler le rapport entre les points et le lissage de la fonction. On cherchera à trouver automatiquement ce paramètre de lissage en fonction de nos entrées.

Pour commencer, on interpolera des points une dimension en utilisant les splines naturelles C^2 qui consistent à faire passer entre chaque point un polynôme de degré inférieur ou égal à 3 et qui soit de classe C^2 dans le cas uniforme et non uniforme.

Pour tenir compte des fluctuations des mesures, on implémentera une méthode de splines naturelles de lissages, qui permettent de minimiser une certaine quantité représentant la différence entre la courbe et les points mesurés.

Les données peuvent contenir des valeurs aberrantes, il faudra identifier ces valeurs, les supprimer ou leur attribuer un poids faible.

Enfin, on adaptera ces méthodes au cas paramétrique (2D).

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 une dimension" (nommée "*Splinesunedimension.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 consignnant les dépendances entre les programmes, et les instructions d'utilisation.
- Un rapport expliquant les diverses méthodes, avec des exemples d'utilisations illustrés.

3 Limites

- Pour les splines de lissages on minimise une quantité composé de deux parties, une qui assure le passage de la spline par la proximités de nos données et l'autre assure le lissage de la spline, d'où il faut faire un choix selon le paramètre de lissage , donc on prend soit une spline qui suit bien le nuage des points mais qui n'est pas très lisse, soit une spline qui est bien lisse mais qui suit pas tous le nuage généré par l'ensemble de nos données .

- Dans le cas 2D il y a plusieurs façons d'interpoler les données donc il faut trouver la méthode la plus optimale .

- Les points aberrants : la détection de ces points dépend de la méthode choisie, donc une même donnée peut considérer comme donnée aberrante dans un modèle cependant qui n'est pas dans un autre modèle, donc il faut trouver une méthode qui gère bien les données aberrantes

Restriction de la solution proposée (cadrage) :

- Les points donnés en entrée seront défini dans un ordre précis (pour le paramétrique).
- Pour les splines de lissages on minimise une quantité composé de deux parties, une qui assure le passage de la spline par la proximités de nos données et l'autre qui assure le lissage de la spline. Il faut faire un choix selon le paramètre de lissage , donc on prend soit une spline qui suit bien le nuage des points mais qui n'est pas très lisse, soit une spline qui est bien lisse mais qui suit pas tous le nuage généré par l'ensemble de nos données.
- Dans le cas 2D il y a plusieurs façons d'interpoler les données : il faut trouver la méthode la plus optimale.
- Les points aberrants : la détection de ces points dépend de la méthode choisie, donc une même donnée peut considérer comme donnée aberrante dans un modèle cependant qui n'est pas dans un autre modèle. Il faut trouver une méthode qui gère bien les données aberrantes.

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 de fonctions

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 données à interpoler. Ce fichier doit contenir 2 colonnes (séparées par un espace) et un nombre inconnu de lignes. La première colonne représente les abscisses, 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 abscisses $\{0, 1, 2, 3\}$, et en ordonnées $\{0, 1, 6, 9\}$.

Puis il renseigne la dimension de ses données (une dimension ou paramétriques) : En une dimension, l'ordre des points donnés n'a pas d'importance. En paramétrique, l'ordre des points forme un paramètre de plus : pour des points $(x_i, y_i), i \in \llbracket 0; n \rrbracket$, la spline sera définie par sa valeur en un paramètre $t \in [0, 1]$ tel que si $t = f(i, n)$, alors $S(t) = (x_i, y_i)$, avec $f(i, n)$ dépendant de la répartition utilisée (par exemple, en répartition uniforme donne $f(i, n) = \frac{i}{n}$).

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

- Mercredi 18 décembre : présentation du cahier des charges et de l'organisation du projet.
- Jeudi 30 avril : soutenance du projet et rendu des codes.