

# Interpolaspline

Rapport du stage applicatif

L3 MIN

*Décembre 2019*

CORBILLE Clément, DOUMBOUYA Mohamed, EL BOUCHOUARI  
Zakaria, HEDDIA Bilel, PIASENTIN Béryl, RODET Amélys

## Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Organisation de la semaine</b>	<b>2</b>
<b>2 Contexte et objectifs</b>	<b>3</b>
<b>3 Définition des tâches</b>	<b>4</b>
3.1 Création des tâches . . . . .	4
3.2 Contenu des tâches . . . . .	4
<b>4 Cahier des charges</b>	<b>6</b>
<b>5 Planification des tâches</b>	<b>7</b>
5.1 Organisation des tâches . . . . .	7
5.1.1 Construction du diagramme de Gantt . . . . .	7
5.1.2 Graphe des dépendances . . . . .	8
5.2 Répartition des tâches . . . . .	10
<b>6 Risques</b>	<b>10</b>
<b>Conclusion</b>	<b>10</b>
<b>Annexes</b>	<b>10</b>

# Introduction

Dans le cadre de notre troisième année de licence mathématiques-informatique à l'Université Grenoble Alpes, nous devons effectuer un stage applicatif à l'UFR IM<sup>2</sup>AG. Ce stage, qui consiste en un projet, est une étape indispensable pour mettre en pratique les connaissances acquises durant cette formation et pour acquérir de nouvelles compétences de recherche. Les projets ont lieu durant quatre semaines, réparties en deux périodes de respectivement une puis trois semaines. Des groupes de six étudiants ont été construits en fonction des affinités de chacun pour les différents sujets proposés.

Le sujet de notre projet est l'interpolation avec données aberrantes. Il nous donne l'opportunité d'approfondir nos connaissances en mathématiques appliquées, plus précisément d'approfondir l'enseignement d'algèbre linéaire pour le graphique et la CAO (Conception Assistée par Ordinateur) que nous avons reçu, ainsi que la programmation en langage python. De plus, ce stage va nous permettre d'acquérir des compétences en travail de groupe.

## 1 Organisation de la semaine

L'objectif de cette semaine de stage était de mettre en place l'organisation nécessaire à la réalisation du projet qui aura lieu en avril.

Nous avons commencé par désigner un chef de projet : Béryl a été choisie à cet effet. La chef de projet veille sur le bon déroulement de chaque tâche en plus de sa participation aux différentes tâches. Nous avons ensuite défini un nom pour le projet, en rapport avec le contenu mais également percutant. "Interpolaspline" a été le nom retenu, en tant que fusion du mot "interpolation" et du mot "spline", deux notions principales du sujet.

Durant cette semaine d'organisation, chaque membre du groupe a été responsable d'une ou de plusieurs tâches. Être responsable d'une tâche signifie s'assurer que celle-ci est correctement réalisée et terminée dans les délais fixés, mais pas obligatoirement qu'elle doit être réalisée seul.

Les premiers jours ont été mis à profit pour comprendre le sujet et trouver ce sur quoi nous allions travailler pendant trois semaines. Il a fallu ensuite établir un plan de travail. Cette réflexion nous a permis de construire un diagramme de Gantt et un graphe de dépendances des tâches, mais aussi de répartir les tâches entre les différents membres du groupe. Chacun sait donc de quelle tâche il est responsable et peut se projeter dans la période de réalisation.

Une fois tous ces points mis en place, nous avons établi le cahier de charges. Nous y avons décrit le problème, nos objectifs, le cadre du projet. La description des fonctionnalités livrées à la fin du projet y est également détaillée, ainsi que l'organisation temporelle du projet.

Tous les membres du groupes ont participé à la rédaction du rapport basé sur cette semaine de travail, incluant le cahier des charges, mais aussi à la préparation de la soutenance. Pour terminer cette semaine d'organisation, chacun a pris le temps de bien comprendre ce qu'il devrait faire durant les trois semaines de réalisation.

## 2 Contexte et objectifs

Ce stage applicatif est un court projet permettant de mettre en pratique des connaissances acquises durant la troisième année de licence en parcours mathématiques-informatique, et par ailleurs d'en acquérir sur des notions annexes au programme pédagogique de la licence, au cours de sa réalisation. Le sujet principal du projet sera l'interpolation (par spline de lissage) de données contenant des points aberrants. Il consiste en particulier en l'apprentissage de la gestion de ces derniers, en une et deux dimensions.

Notre projet ne possède pas de client, mais le travail effectué sera évalué par une équipe d'enseignants en charge de superviser notre stage applicatif. Ces enseignants feront partie de notre jury lors de la soutenance finale.

Il n'y a pas de hiérarchie au sein de notre groupe. Le seul rôle particulier sera notre chef de projet, Beryl PIASENTIN. Ci-dessous le reste de l'équipe :

- Clément CORBILLE
- Mohamed DOUMBOUYA
- Zakaria EL BOUCHOUARI
- Bilel HEDDIA
- Amelys RODET

Ce stage applicatif sera effectué dans son intégralité dans le bâtiment IM<sup>2</sup>AG à l'UFR. L'essentiel du travail sera réalisé sur les postes informatiques du bâtiment, aucune expérience particulière n'est prévue. Le sujet du projet ne nécessite pas de circonstances temporelles autres que les périodes qui sont prévues : nous avons une semaine consacrée à la gestion de notre projet du 10 au 18 décembre puis 3 semaines en avril consacrées à sa réalisation (du 6 au 10, du 14 au 17 puis du 27 au 30).

Il n'est pas question de budget dans ce projet. La seule ressource (autre qu'humaine) sera matérielle. Le matériel nécessaire comprend les postes informatiques (un par personne) et les logiciels intégrés dans ceux-ci qui nous permettront de réaliser le travail demandé. Nous avons besoin de Python, dont les bibliothèques usuelles de mathématiques (numpy), d'affichage (matplotlib), et d'interface graphique (tkinter).

Un objectif à court terme de ce stage applicatif est de le terminer dans les temps imposés par le corps enseignant. Un objectif à long terme est d'acquérir de la pratique concernant la gestion d'un projet similaire à celui d'un stage.

## 3 Définition des tâches

### 3.1 Création des tâches

Dans le but d'interpoler des données en utilisant des splines de lissage tenant compte des valeurs aberrantes, nous nous devons de savoir comment aborder le sujet. En discutant entre membres du groupe, nous avons convenu qu'il fallait dans un premier temps approfondir les notions générales sur les splines naturelles et de lissage en fonction de la répartition des points, ainsi que trouver un moyen d'automatiser l'estimation du paramètre de lissage. Puis nous avons prévu d'étudier des méthodes d'identification des valeurs aberrantes que l'on souhaite à terme implémenter en deux dimensions, mais aussi des méthodes permettant d'interpoler des splines sans commencer par détecter les valeurs aberrantes.

### 3.2 Contenu des tâches

Les tâches sont les suivantes :

- Approfondissement des splines naturelles
- Approfondissement des splines de lissage uniformes
- Approfondissement des splines de lissages non-uniformes
- Estimation automatique du paramètre de lissage
- Identification des points aberrants
- Etude des splines de lissage avec points aberrants
- Etude de l'algorithme RANSAC
- Redéfinition d'autres fonctions d'erreur à minimiser
- Finalisation du projet

Voici quelques explications du contenu scientifique constituant la base des tâches listées précédemment.

#### Splines naturelles

L'interpolation est une opération mathématique consistant à déterminer une fonction passant par un nombre fini de points donnés.

Soient  $(x_i, y_i)$  les  $n + 1$  données,  $i \in \llbracket 0; n \rrbracket$ . Une spline est une fonction définie par morceaux sur chaque intervalle  $[x_i, x_{i+1}]$ ,  $i \in \llbracket 0; n - 1 \rrbracket$ . Le principe d'une spline est de combiner plusieurs morceaux de fonctions (en général on utilise des polynômes de petit degré) pour former une fonction d'une classe définie.

Les splines cubiques  $C^2$  sont les splines définies par des polynômes cubiques dont la dérivée seconde est continue. On appelle spline naturelle la spline cubique  $C^2$  dont les dérivées secondes aux extrémités de l'intervalle de définition sont nulles. La spline naturelle existe toujours pour des données distinctes, et est unique. Le principe de la spline cubique naturelle permet d'obtenir un système tridiagonal simple qui peut être résolu facilement, afin d'obtenir les coefficients des polynômes.

### **Splines de lissage**

Les splines de lissage sont des splines cubiques naturelles dont chaque polynôme cubique est une approximation des données se trouvant sur son intervalle de définition. Cette spline ne passera dans la plupart des cas pas par tous les points, mais est la spline qui minimise une quantité liée à la distance entre les données et la spline. En général, la spline approximant les données cherche à minimiser l'erreur au carré : c'est l'approximation aux moindres carrés.

Les splines de lissage permettent d'éviter les oscillations qui seraient présentes avec une spline naturelle passant par tous les points, provoquées avec un nombre de données très grand.

### **Paramètre de lissage**

La quantité à minimiser pour construire la spline approximant des données est composée de deux parties dont la seconde est pondérée par un paramètre de lissage : la première assure le passage de la spline par nos données tandis que la seconde assure le lissage de la courbe. Cela signifie que plus on augmente la valeur du paramètre de lissage, plus le lissage sera fort ce qui donne par suite une spline passant de moins en moins proche des données.

### **Données aberrantes**

En statistiques, une donnée aberrante est une valeur ou une observation qui est « distante » des autres observations effectuées sur le même phénomène, c'est-à-dire que c'est une valeur qui s'écarte fortement des valeurs des autres observations : elle est anormalement faible ou élevée par rapport à celles qui lui sont proches. Sur un graphe on peut les visualiser comme des points qui se situent loin du nuage de points principal.

Les méthodes pour créer une spline de lissage malgré les données aberrantes sont variées : certaines déterminent les points aberrants pour ensuite les supprimer ou leur associer un poids faible tandis que d'autres créent la spline tout en gérant les points aberrants, sans les détecter auparavant.

## RANSAC

RANSAC est l'abréviation de RANdom SAmple Consensus, c'est à dire consensus d'échantillons aléatoires. C'est le nom d'un algorithme qui gère les points aberrants sans prendre le temps de les détecter dans un premier temps. Cet algorithme est itératif, et non-déterministe. Un algorithme non déterministe est un algorithme qui ne produit pas toujours la même sortie pour une même entrée.

## Interpolation paramétrique

L'objectif de l'interpolation paramétrique est de trouver un polynôme paramétrique qui interpole les données. Pour ce faire, deux méthodes sont possibles :

- L'interpolation séparée des  $n + 1$  données  $(t_i, x_i)$  et des  $n + 1$  données  $(t_i, y_i)$ ,  $i \in \llbracket 0; n \rrbracket$ , avec  $t$  un paramètre bien choisi. On obtient deux splines  $P$  et  $Q$  interpolant respectivement les données  $(t_i, x_i)$  et  $(t_i, y_i)$ ,  $i \in \llbracket 0; n \rrbracket$ . En associant les deux splines obtenues grâce au paramètre  $t$ , on obtient une spline paramétrique  $S$  interpolant les données de départ :  $S(t) = (P(t), Q(t))$ . En particulier,  $S(t_i) = (x_i, y_i)$ ,  $i \in \llbracket 0; n \rrbracket$ .
- L'interpolation en une fois des données, sans appliquer la méthode à une dimension sur chaque dimension.

## 4 Cahier des charges

Le cahier des charges est l'annexe 4. Nous y avons détaillé le projet, du sujet à l'archive rendue, en passant par sa réalisation. Ce cahier des charges représente notre compréhension globale du sujet ainsi que les principales idées d'organisation et de réalisation.

Ce document est censé être le lien entre notre équipe et le client, mais il est ici le lien entre notre équipe et l'équipe d'enseignants responsable du stage applicatif. Nous allons l'utiliser durant la seconde fin du projet. Notre objectif est de réaliser tous les objectifs qui y sont définis, dans les délais impartis.

## 5 Planification des tâches

### 5.1 Organisation des tâches

#### 5.1.1 Construction du diagramme de Gantt

Il nous faut donc planifier de façon la plus optimale possible nos tâches à accomplir sur les dates qui nous sont allouées pour le projet. Le diagramme de Gantt sera notre principal outil.

La construction du diagramme fut basée sur un développement par prototypes successifs, c'est-à-dire de façon itérative, nous sommes “revenus dessus” plusieurs fois pour l'affiner tout en déterminant et analysant les dépendances des tâches.

Ci-joint en annexe 1 notre diagramme de Gantt terminé.

Nous l'avons représenté sous la forme d'un tableau à double entrée comprenant le détail des tâches (et leur numéro de tâche respectif) en ordonnée et la chronologie représentée par les jours de projet en abscisse.

La hiérarchie des tâches est montrée. Ainsi la tâche 1 regroupe 1a et 1b et 5a contenue dans la 5 regroupe 5aa, 5ab, 5ac et 5ad. La durée de ces tâches de haut niveau sont représentées par des bandelettes jaunes sur le diagramme. Les tâches de niveau inférieur sont planifiées par des cases grises contenant un chiffre désignant le nombre de personne nous paraissant indispensable pour sa réalisation.

Pour gérer les ressources humaines, nous avons ajouté une ligne à la fin du diagramme pour contrôler les effectifs affectés et aider à la prise de décision pour la répartition du travail.

Un diagramme de Gantt “réduit” comprenant uniquement les tâches de plus haut niveau est également fourni en annexe 2.

### 5.1.2 Graphe des dépendances

Le graphe des dépendances a pour objectif de représenter ces dépendances de la façon la plus claire possible. Malgré les nombreuses dépendances, nous avons trouvé un moyen pour rendre ce graphe lisible et facilement compréhensible.

#### Tâches "parents" :

Pour trouver les tâches nécessaires au début d'une, il suffit de remonter dans l'arborescence. Par exemple dans l'exemple ci-dessous, la tâche 6d a besoin de 6bb, 6c et 6aa.

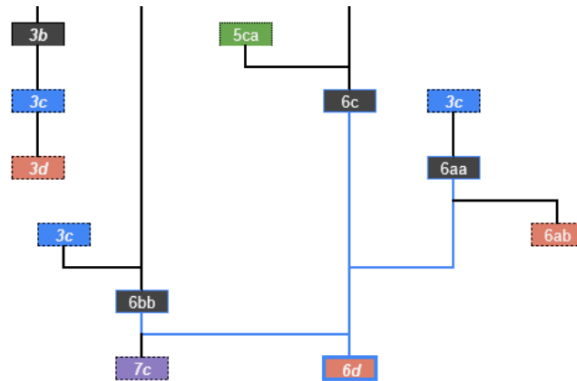


FIGURE 1 – Exemple de tâches "parents"

#### Tâches "fils" :

De même, pour trouver les tâches découlant de la réalisation d'une précédente, il suffit de descendre dans le graphe.

Par exemple dans l'exemple ci-après, la réalisation de 4b permet de commencer 4c et 5bb (même si 5bb aura également besoin de la réalisation de 5ba).

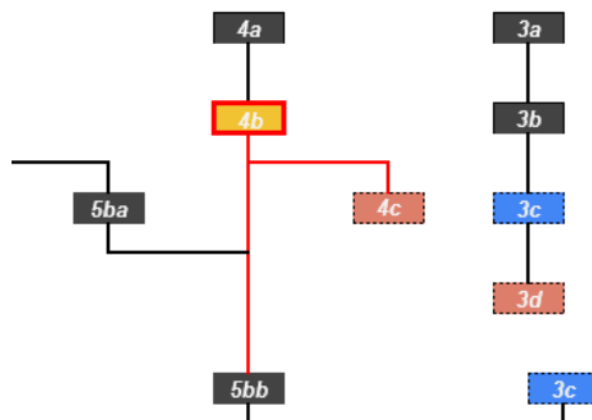


FIGURE 2 – Exemple de tâches "fils"



### Tâches “à-multiples-apparitions” :

Le meilleur moyen que nous avons trouvé pour éviter le surplus de connexions dans le graphe dû aux multiples dépendances, est de faire apparaître une tâche plusieurs fois. Voilà pourquoi un jeu de couleur à été employé pour permettre une meilleure visibilité de ces tâches “complexes”.

Par exemple dans le cas ci-dessous, la tâche 3c dépend de la fin des tâches 0a, 5ac, 5bc et 3b et sa réalisation est nécessaire pour commencer 6bb et 6aa.

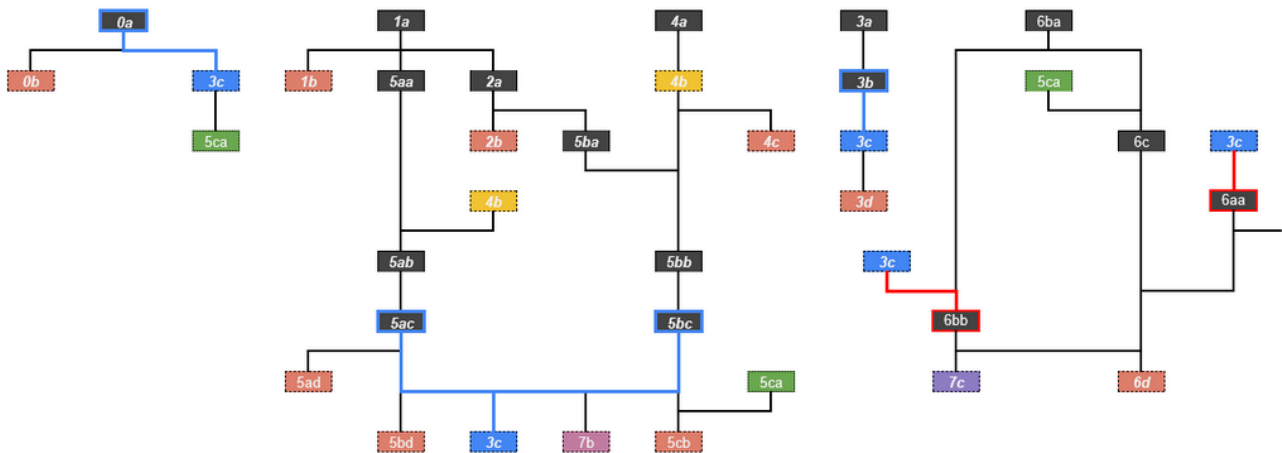


FIGURE 3 – Exemple de tâches "à-multiples-apparitions"

Le graphe des dépendances complet est visible en annexe 4.

Moyens de communication :

Même si nous avons sérieusement planifié notre projet, nous avons conscience qu'il est rare que tout se passe comme prévu. Pour éviter l'effet tunnel, autrement dit anticiper les retards et réagir au plus vite sur les ressources nécessaires sur une même tâche, nous mettrons en place des bilans sur le travail de chacun lors de chaque demi-journée et ainsi leur permettre à chacun de partager au groupe son avancée. Des réunions plus complètes seront également mises en place, cette fois-ci tous les deux jours.

Nous utiliserons Discord comme moyen de communication intra-groupe. Pour ce qui est des fichiers du projet, nous utiliserons le service web d'hébergement Github (et également les applis de Google drive pour la mise en commun et la modification en temps réel sur les fichiers destinés au rapport final).

## 5.2 Répartition des tâches

Pour éviter les risques, la répartition du travail s'est effectuée dans l'optique d'avoir un minimum de demi-journées où la totalité des membres de groupe est responsable d'une tâche, de sorte à pouvoir répondre aux risques de retard sur l'avancée d'une tâche.

Le groupe entier a pu participer à la réunion visant à répartir le travail, pour que les membres puissent choisir de travailler sur les tâches où ils se sentiront les plus efficaces, pour également limiter les risques de retard.

Le tableau légendé fourni en annexe 3 est le résultat de cette réunion.

## 6 Risques

Le risque majeur du projet est de ne pas avoir estimé correctement le temps nécessaire pour réaliser certaines tâches. En effet, si une fonctionnalité met trop de temps à être implémentée, certaines tâches qui en dépendent risquent de prendre du retard, cela est l'effet boule de neige. Pour éviter cela, le diagramme de Gantt prévoit des membres "flottants", qui se greffent sur les tâches posant des difficultés.

## Conclusion

Nous avons confiance en l'aboutissement du projet. En effet, nous avons construit une bonne cohésion de groupe, malgré le fait que l'on n'ait pas l'habitude de travailler ensemble. De plus, tous les membres du groupes ont bien compris la notion de base du sujet (les splines), développée durant l'enseignement d'algèbre linéaire pour le graphique et la CAO. Certains d'entre nous ont également déjà reçu des enseignements de statistiques approfondis. Cela nous permettra de mieux appréhender l'identification des données aberrantes, la répartition des tâches étant faite en fonction des capacités de chacun.

Dans l'idéal, nous souhaiterions réussir à développer le projet d'une manière qui permette de pouvoir ensuite facilement traiter des données dans n'importe quelle dimension.

## Annexes

1. Diagramme de Gantt (en 2 parties)
2. Diagramme de Gantt simplifié et répartition des tâches
3. Graphe de dépendance des tâches
4. Cahier des charges

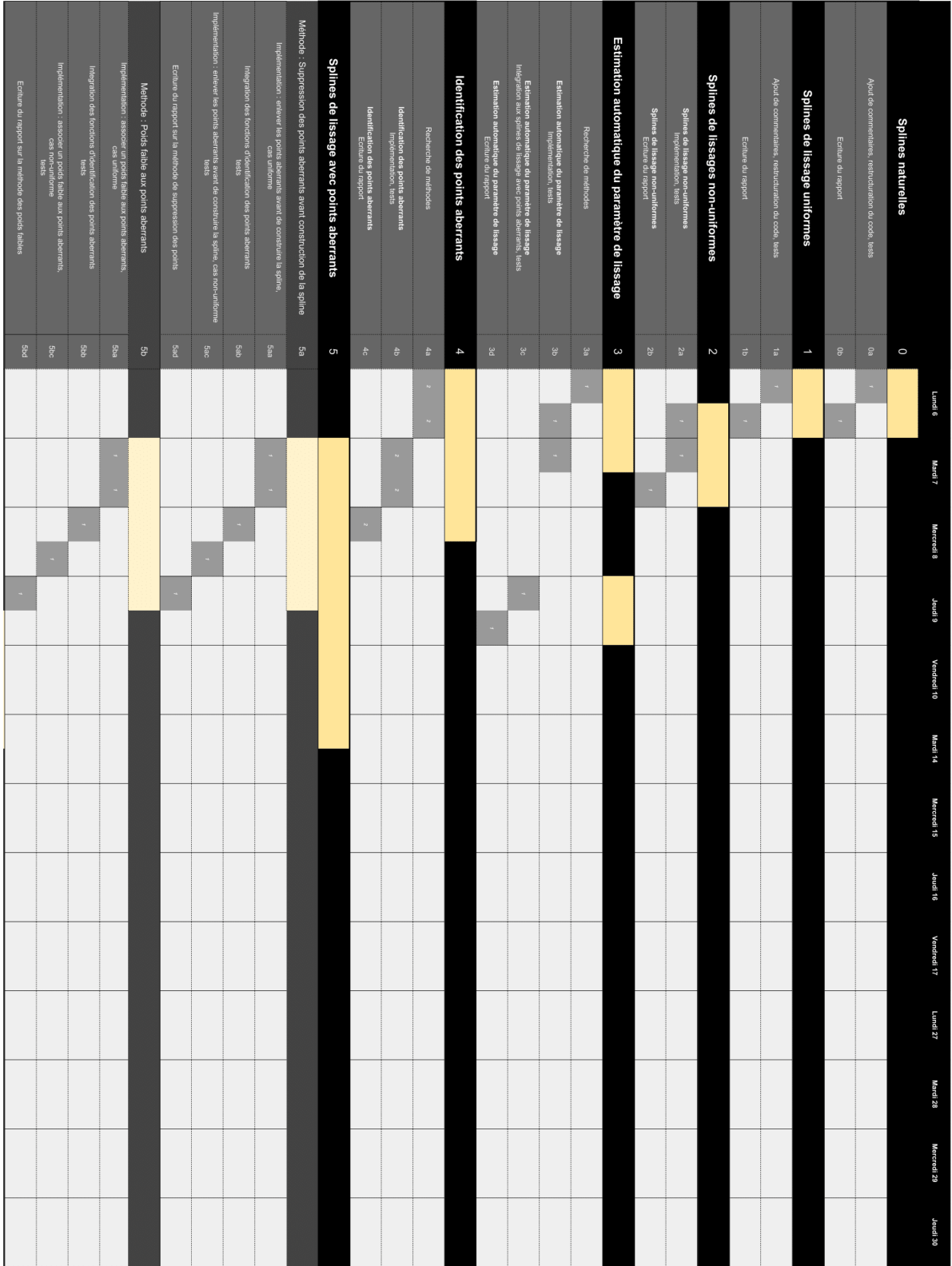


FIGURE 4 – Diagramme de Gantt (partie 1)



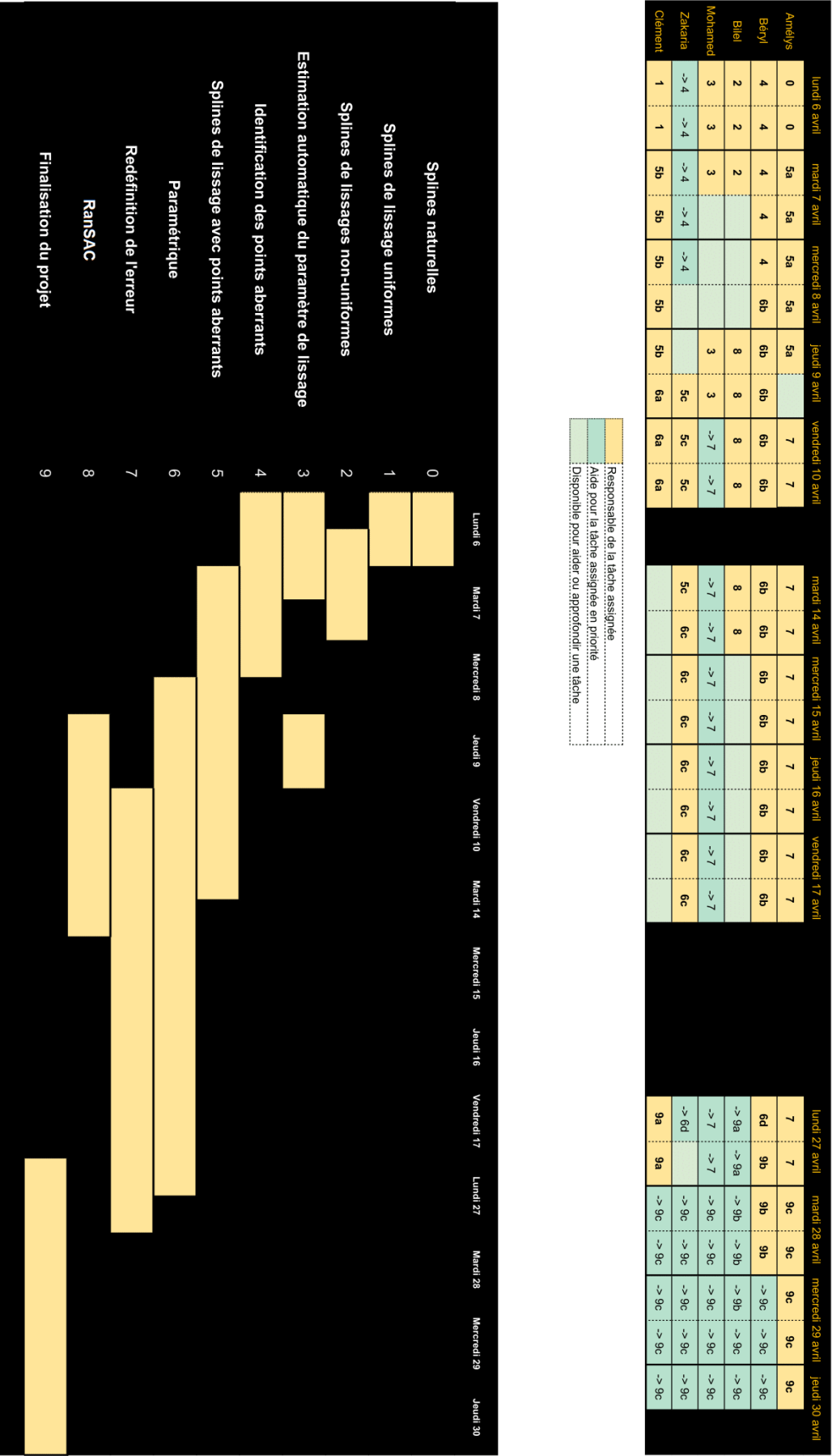


FIGURE 6 – Diagramme de Gantt simplifié et répartition des tâches

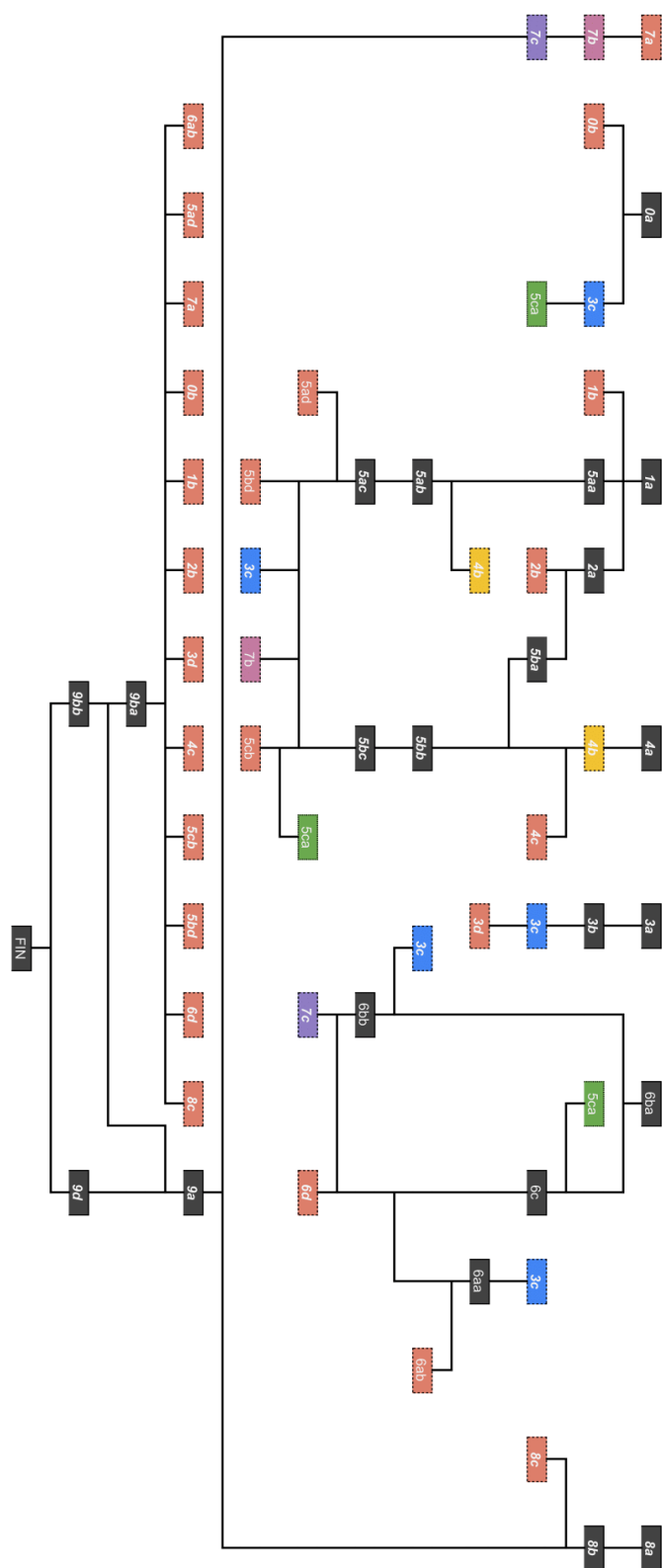


FIGURE 7 – Graphe de dépendances des tâches

# Cahier des charges

## Interpolaspline

Stage Applicatif L3 MIN

*Année 2019-2020*

## Table des matières

<b>1</b>	<b>Contexte et définition du problème</b>	<b>2</b>
1.1	Motivations . . . . .	2
1.2	Problème . . . . .	2
<b>2</b>	<b>Objectifs</b>	<b>3</b>
<b>3</b>	<b>Limites</b>	<b>5</b>
<b>4</b>	<b>Description fonctionnelle</b>	<b>5</b>
4.1	Fonctionnalités . . . . .	5
4.2	Entrées et sorties . . . . .	6
4.2.1	Entrées . . . . .	6
4.2.2	Sorties . . . . .	7
<b>5</b>	<b>Organisation temporelle</b>	<b>7</b>

# 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

Notre objectif est de réussir à résoudre le problème évoqué précédemment, en une mais aussi en deux dimensions. Pour cela, l'idée est d'interpoler des données en tenant compte des points aberrants, c'est à dire de trouver une ou plusieurs méthodes pour les détecter et les traiter.

Pour commencer, nous interpolerons des données à une dimension en utilisant les splines naturelles cubiques  $C^2$ . Une spline naturelle cubique  $C^2$  est une fonction définie par morceaux par des polynômes cubiques, dont la dérivée seconde est continue et dont les dérivées secondes aux extrémités de l'intervalle de définition sont nulles.

Puis, nous implémenterons une ou plusieurs méthodes permettant de créer des splines de lissage en tenant compte des données aberrantes. Les splines de lissage sont des fonctions qui minimisent une certaine quantité liée à la différence entre la courbe et les données mesurées. L'identification des valeurs aberrantes sera implémentée pour ensuite les supprimer avant de créer la spline, ou pour leur attribuer un poids faible lors de la minimisation de la quantité.

Enfin, nous adapterons ces méthodes au cas paramétrique (données à 2 dimensions).

## 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 "*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 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

Voici les limites du projet :

- Le choix automatique du paramètre de lissage ne pourra pas être idéal pour tous les cas. En effet, la quantité minimisée pour les splines de lissages est composée de deux parties : une qui assure le passage de la spline à proximité de nos données, et l'autre qui assure le lissage de la spline. Nous ne pouvons pas, dans la plupart des cas, avoir une spline qui suit bien le nuage de points et qui est bien lisse. Il faut donc faire un choix de pondération, qui pourrait ne pas être le même dans deux contextes différents avec les mêmes données.
- La robustesse de la gestion des points aberrants dépendra des données. En effet, chaque méthode possède ses points faibles, et possèdera donc au moins un jeu de données pour lequel elle ne sera pas efficace : une même donnée peut être considérée comme donnée aberrante par une méthode mais pas par une autre.

## 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 deux dimensions) : en une dimension, l'ordre des points donnés n'a pas d'importance. En paramétrique (deux dimensions), l'ordre des points forme un paramètre de plus : pour  $n + 1$  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, la répartition uniforme donne  $f(i, n) = \frac{i}{n}$ ). Il faut donc que les points soient rentrés dans l'ordre dans le cas paramétrique.

L'utilisateur 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

- Du mardi 10 au mercredi 18 décembre : organisation du projet, écriture du cahier des charges
- Mercredi 18 décembre : présentation du cahier des charges et de l'organisation du projet
- Du lundi 6 au jeudi 30 avril : réalisation du projet
- Jeudi 30 avril : soutenance du projet et rendu des codes