Rapport d'activité 2

Groupe Trioux – Veal Phan

Cette semaine, nous nous sommes concentré sur la problématisation de notre sujet : nous avons donc principalement discuté de la manière dont nous prévoyons d'implémenter les méthodes d'interpolation du sujet.

1. Choix d'une fonction à interpoler

Afin de pouvoir calculer de façon simple la qualité de nos interpolations, nous avons décidé de choisir dans un premier temps une fonction connue, et de procéder à une interpolation sur quelques points de son graphe. Compte tenu du fait que les techniques d'interpolation que nous allons étudier se basent sur des polynômes ou des fonctions trigonométriques, il nous a paru judicieux de choisir une fonction non polynomiale et non trigonométrique. Nous avons donc choisi arbitrairement d'étudier la fonction de masse de la loi exponentielle :

$$f(x) = \lambda e^{-\lambda x}$$

Les abscisses des points sur lesquels porteront les interpolations seront générées par les fonctions suivantes sous forme de listes de réels :

```
a,b : des réels,  les bornes entre lesquelles nous allons prendre nos points
n : un entier, le nombre de points à prendre
x : une liste de réels, les abscisses des points que nous allons utiliser pour
l'interpolation. Contient n+1 éléments, [x0, x1, ...,xn]
k : un entier, une variable d'incrément

fonction distribReg(réel a, réel b, entier n) : liste de réels
DEBUT
   pour(k de 0 à n par +1) faire
      x[k] <- a + k*(b-a)/n
   fait
   distribReg <- x
FIN

fonction distribTchebychev(réel a, réel b, entier n) : liste de réels
DEBUT
   pour(k de 0 à n par +1) faire
   x[k] <- (a+b)/2 - ((b-a)/2) * cos((k+0.5) * pi * (1/(n+1)))
   fait
   distribTchebychev <- x
FIN
```

Nous réfléchissons encore au langage à utiliser pour implémenter notre projet : nous hésitons entre utiliser Maxima ou Python. Maxima est en effet fourni avec des packages et des fonctions basiques permettant d'implémenter les différentes interpolations très facilement, mais représente ainsi en quelque sorte la solution de facilité.

2. Les méthodes d'interpolation

2.1. Interpolation de Lagrange

L'interpolation selon Lagrange, consiste à obtenir l'unique polynôme de degré au plus n-1 passant par les n points d'un échantillon donné. On associe à chaque point un polynôme de degré n-1 passant par ce même point tel que tous les autres polynômes considérés s'annulent en ce même point. La somme pondérée de ces polynômes permet d'obtenir le polynôme de Lagrange.

Maxima est fourni avec la fonction « interp » du package « interpolation » qui permet d'obtenir le polynôme correspondant à une série de points.

2.2. Interpolation de l'Hermite

Le principe de l'interpolation de l'Hermite est très proche de celui de l'interpolation de Lagrange, la différence principale étant que nous nous intéressons en plus aux valeurs des dérivées aux points choisis. Nous supposons que nous pourrons réutiliser les fonctions écrites pour l'interpolation de Lagrange construire notre interpolation de de l'Hermite.

2.3. Interpolation des moindres carrés

L'interpolation des moindres carrés consiste à interpoler notre série de points par un polynôme de degrés choisi. Ce polynôme est de telle sorte qu'il minimise la somme du carré des erreurs, c'est-à-dire la différence entre les points de la série et les points du polynôme à ces mêmes abscisses.

Les coefficients sont obtenus par des opérations matricielles et ne posent à première vue pas de difficulté majeure.

2.4. Approximation par un polynôme trigonométrique

Nous ne nous sommes pas encore penché sur cette méthode-là.

3. Programme envisagé de la semaine prochaine

La semaine prochaine, nous comptons finir de problématiser les différentes techniques d'interpolation, c'est-à-dire écrire les algorithmes des méthodes Lagrangiennes et des moindres carrés, formaliser la méthode de l'Hermite et l'approximation par un polynôme trigonométrique.