

1 Introduction

L'idée de ce projet est née de la présentation par Emmanuel Ostenne, lors d'une réunion du groupe AMECMI de l'IREM de Lille, d'une activité, en l'occurrence « *Température piscine* » que l'on trouve [ici](#), destinée à des élèves de Troisième .

Emmanuel et moi-même avons pensé qu'il pourrait être intéressant de proposer le thème de l'interpolation et, de fait, des polynômes de Lagrange à des élèves de Terminale S, plus particulièrement à ceux qui suivent la spécialité Mathématique, ce qui permettrait alors de faire le lien entre les travaux de recherche des élèves de Troisième et ceux réalisés par les lycéens.

Le présent document a pour but de décrire la réalisation du projet au travers des séquences successives de travail.

Il montre la chronologie du questionnement des élèves et les outils, théoriques et informatiques, mis en place pour résoudre les problèmes qui sont apparus alors.

L'approche de la notion d'interpolation avec des élèves de lycée ne peut être la même que celle déployée dans le supérieur. C'est ce qui explique mon choix d'un « départ » en douceur.

D'autre part, les méthodes exposées sont parfois celles proposées par les élèves, et ne sont pas nécessairement optimisées.

Il est donc attendu de la part des éventuels lecteurs, experts ou non, une certaine indulgence 😊 .

2 Objectifs

Les objectifs de ce projet sont multiples :

- découvrir une notion mathématique hors programme, mais qui peut être traitée, en acceptant néanmoins d'admettre quelques notions et résultats théoriques, avec des outils du programme de TS,
- développer les compétences de démarche de recherche et d'investigation,
- développer le travail de groupe,
- développer l'utilisation, quand cela s'avère nécessaire, des outils informatiques,
- développer l'expression orale.

3 Mise en œuvre

N'ayant pas cette année le groupe *Spé Math* en responsabilité, la mise en œuvre a posé quelques difficultés pratiques d'organisation.

Il m'a été très difficile de placer quelques heures à l'emploi du temps des élèves, et ce de façon pérenne.

J'ai donc décidé, avec l'accord des élèves concernés, de leur proposer de réaliser le travail à la maison, puis, de faire une synthèse lors de séances placées sur des créneaux horaires disponibles ponctuellement.

Cela a constitué un frein à une progression sereine du projet, le temps séparant deux réunions étant parfois trop important pour mobiliser les élèves efficacement.

4 Séances

Première étape : Présentation du projet et premier travail

Ce qui suit a été présenté aux élèves lors d'une première séance.

- **Deux problèmes classiques :**

1. Interpolation : on considère une aile d'avion, qu'on soumet à des vents de 10, 50, 100, 200 km/h, et dont on calcule les déformations pour ces valeurs. On veut savoir comment elle résistera à un vent de 150km/h
2. Extrapolation : on connaît la population française de 1800 à 2017 et on veut en déduire une estimation de la population française dans les 10 prochaines années.

- **Un problème de calculs :**

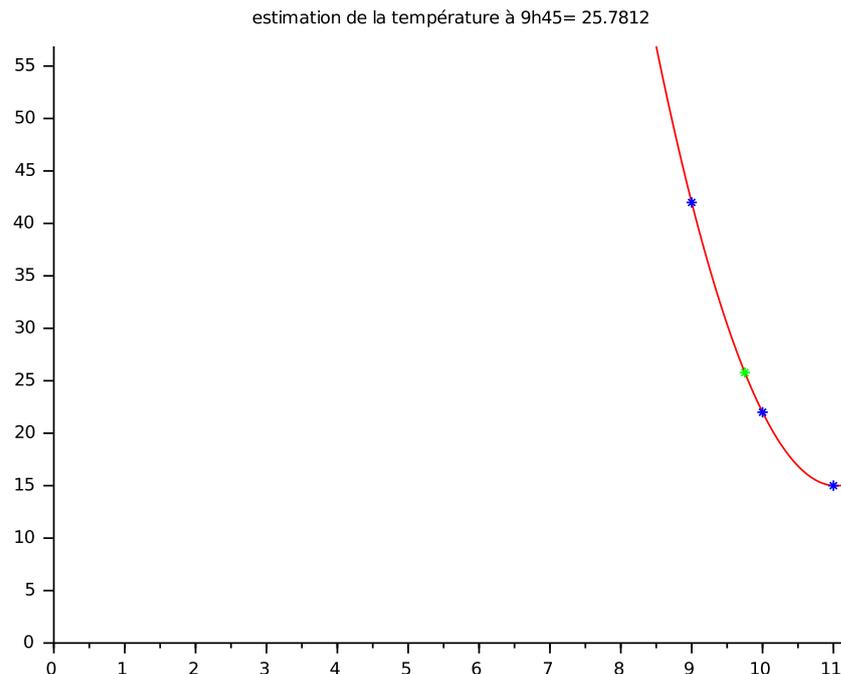
On connaît la fonction f mais, pour des raisons pratiques (dérivée, intégrale, ...), on l'approche par une fonction polynomiale.

LE PROBLÈME MATHÉMATIQUE EST LE SUIVANT : ON SE DONNE $n + 1$ MESURES f_0, \dots, f_n EN $n + 1$ POINTS DISTINCTS x_0, \dots, x_n ET ON CHERCHE À DÉTERMINER UN POLYNÔME P DE DEGRÉ INFÉRIEUR OU ÉGAL À n , AVEC $m \leq n$, QUI « APPROCHE » LES MESURES f_0, \dots, f_n .

L'approche que l'on choisit est le cas où $m = n$: c'est le polynôme d'interpolation.

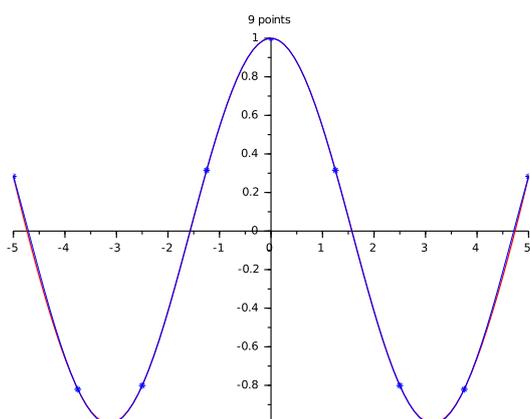
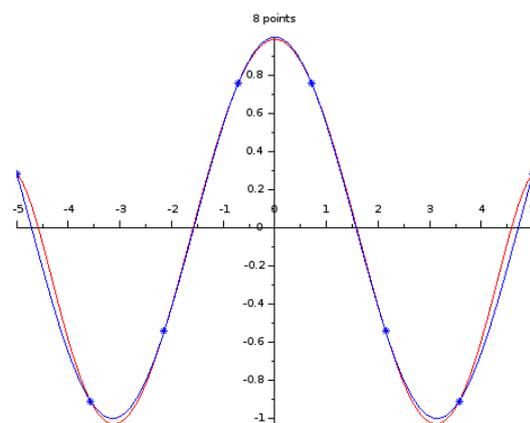
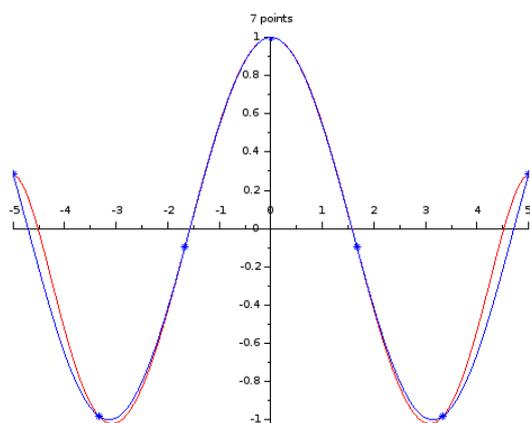
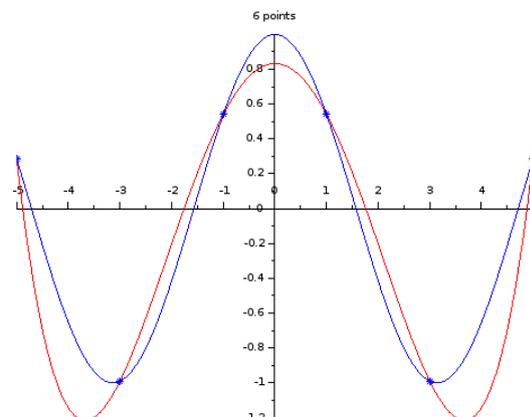
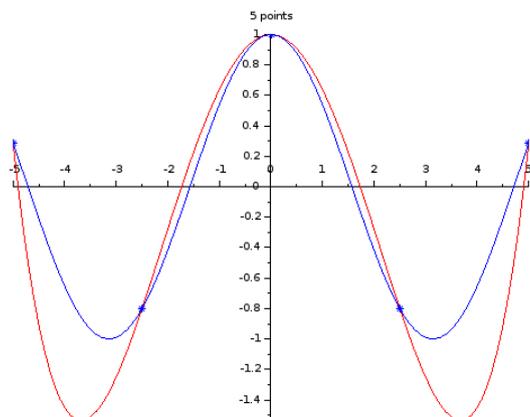
Voici deux exemples qui mettent en œuvre la méthode d'interpolation de Lagrange et que les élèves pourront appréhender en fin de projet :

1. On met dans un réfrigérateur un récipient contenant de l'eau tiède et on mesure la température de l'eau. Les résultats sont 42°C à 8h, 22°C à 9h et 15°C à 10h.
Comment estimer la température attendue à 9h45 ?



La courbe tracée est obtenue à l'aide d'un polynôme de Lagrange de degré 2. Le point vert donne l'estimation.

2. La fonction à interpolier est la fonction **Sinus**, que vous connaissez. On trace sur l'intervalle $[-5;5]$ les polynômes d'interpolation, en rouge, ainsi que la fonction **Sinus**, en bleu, lorsque l'on connaît respectivement 5, 6, 7, 8 et 9 points :



1. **On connaît deux points**

Dans ce cas, $n = 1$, et on a donc les coordonnées de deux points $(x_0; f_0)$ et $(x_1; f_1)$.

? comment faire ?

On peut penser que la solution est triviale 

A vous de jouer 

2. **On connaît trois points**

Dans ce cas, $n = 2$, et on a donc les coordonnées de trois points $(x_0; f_0)$, $(x_1; f_1)$ et $(x_2; f_2)$.

? comment faire ?

Quel sera le degré du polynôme P ?

On peut là encore penser que la solution est relativement simple à trouver 

En vous appuyant sur vos connaissances des classes antérieures, à vous de jouer 

Mise en œuvre de la recherche :

Dans un premier temps, je vous laisse libres d'élaborer toutes les stratégies qui vous viendront à l'esprit, puis dans un deuxième temps, nous ferons un premier bilan de vos recherches et expérimentations, pour, dans un troisième temps, passer à la recherche d'une formalisation du problème dans le cas de $n + 1$ points, appelés aussi « **nœuds** » et la mise en place d'un processus de résolution.

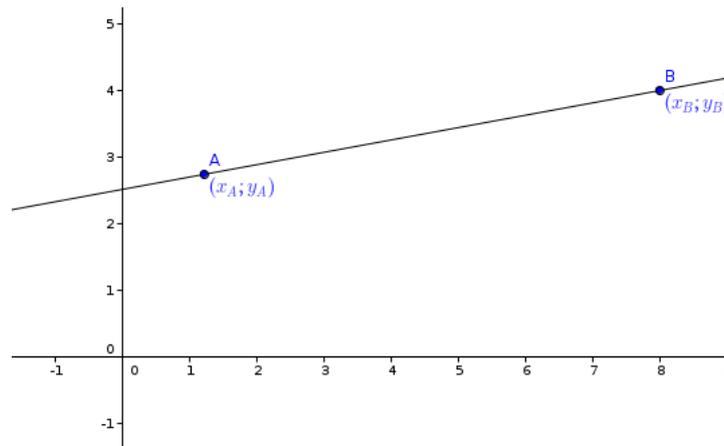
Deuxième étape : solutions pour deux, puis trois points

Voici quelques éléments de solutions et leurs mises en œuvre proposées par les élèves

1. On connaît deux points

Dans ce cas, $n = 1$, et on a donc les coordonnées de deux points $(x_A ; y_A)$ et $(x_B ; y_B)$.

La solution est bien évidemment naturelle. Les élèves se demandent si le professeur n'a pas sous-évalué leurs capacités!!!!



La fonction sera ici la fonction affine

$$f : x \mapsto f(x) = \frac{y_A - y_B}{x_A - x_B} x + y_A - \frac{y_A - y_B}{x_A - x_B} x_A \quad (1)$$

2. On connaît trois points

Dans ce cas, $n = 2$, et on a donc les coordonnées de trois points $(x_A ; y_A)$, $(x_B ; y_B)$ et $(x_C ; y_C)$.

Il est évident pour les élèves que le degré de la fonction polynomiale f sera égale à 2, donc définie par $f(x) = ax^2 + bx + c$, où a, b et c sont les inconnues.

Ils suggèrent de résoudre le système

$$f(x_A) = y_A$$

$$f(x_B) = y_B$$

$$f(x_C) = y_C$$

Ils ne pensent pas à résoudre le problème « à la main » sur feuille, mais ont envie d'utiliser la « machine ». Cela montre un changement d'attitude des élèves face à un problème donné : « vite la machine!!!! ».

Remarques :

- On pourra constater un peu plus tard qu'ils auront besoin de compétences solides pour dompter la « machine ». 😊
- Pour résoudre le système, les élèves empruntent deux voies différentes, certains utilisent le logiciel de calcul numérique *Scilab* (Il y a eu un temps dédié à l'introduction du calcul matriciel qui est l'un des attendus du programme de la spécialité Mathématique), et d'autres, le logiciel de calcul formel *wx-Maxima*, les deux logiciels, *gratuits*, ayant été découverts en début d'année.
- La manipulation des deux logiciels devant se faire à la maison, les élèves sont invités à montrer ponctuellement l'avancement de leur travail et des questions qui peuvent se poser quant à une bonne interprétation et utilisation des deux langages.

Troisième étape : Utilisation du calcul formel et de *wxMaxima* pour trois points

On donne aux élèves les trois points $A(1;6)$, $B(-3;-8)$ et $C(8;-4)$.

Les élèves ont vite formulé que, si $f(x) = ax^2 + bx + c$, alors ils doivent résoudre le système

$$\begin{cases} a+b+c = 6 \\ 9a-3b+c = -8 \\ 64a+8b+c = -4 \end{cases}$$

Voici ce que donne la feuille de calcul d'un élève dans une console *wxMaxima*

Déclarons la fonction f :

```
(%i1) f(x):=a*x^2+b*x+c;
```

```
(%o1) f(x) := c + b · x + a · x2
```

Demandons de résoudre le système suivant :

```
(%i2) linsolve([f(1)=6, f(-3)=-8, f(8)=-4], [a,b,c]);
```

```
(%o2) [a = - $\frac{69}{154}$ , b =  $\frac{401}{154}$ , c =  $\frac{296}{77}$ ]
```

Demandons d'afficher des valeurs approchées de a , b et c :

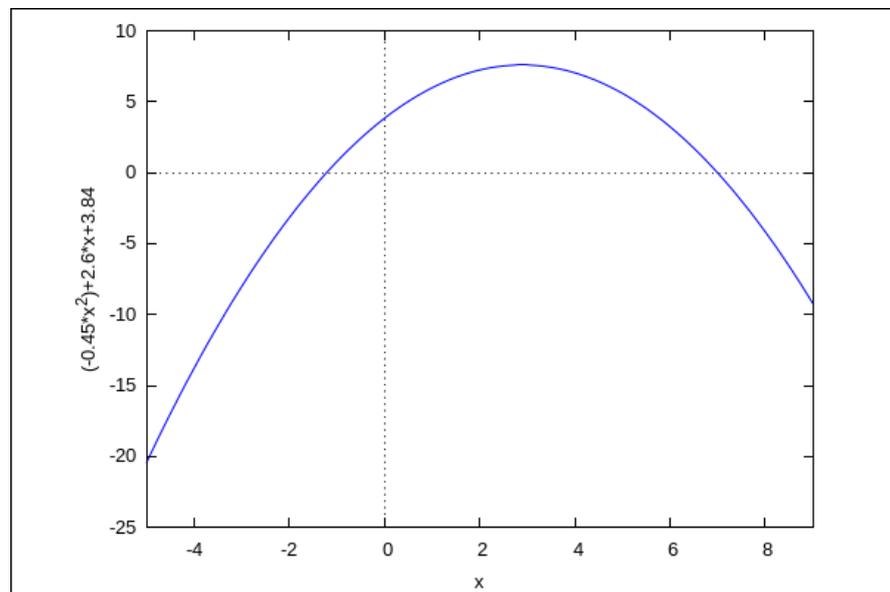
```
(%i3) float(%);
```

```
(%o3) [a = -0.448051948051948, b = 2.603896103896104, c = 3.844155844155844]
```

Traçons la fonction polynomiale associée sur l'intervalle $[-5;9]$:

```
(%i4) wxplot2d([-0.45*x^2+2.6*x+3.84], [x,-5,9])
```

```
(%t4)
```



Si on veut reprendre le problème dans le cas général de trois points, voilà ce que les élèves affichent dans la console :

(%i5) linsolve([f(xa)=ya, f(xb)=yb, f(xc)=yc], [a,b,c]);

$$(%o5) \left[a = \frac{(xb-xc)ya + xc yb - xb yc + xa (yc-yb)}{xa (xc^2-xb^2) - xb xc^2 + xb^2 xc + xa^2 (xb-xc)}, b = -\frac{(xb^2-xc^2) ya + xc^2 yb - xb^2 yc + xa^2 (yc-yb)}{xa (xc^2-xb^2) - xb xc^2 + xb^2 xc + xa^2 (xb-xc)}, c = \frac{(xb^2 xc - xb xc^2) ya + xa^2 (xb yc - xc yb) + xa (xc^2 yb - xb^2 yc)}{xa (xc^2-xb^2) - xb xc^2 + xb^2 xc + xa^2 (xb-xc)} \right]$$

-->

(%i6) factor(%);

$$(%o6) \left[a = \frac{-xb ya + xc ya + xa yb - xc yb - xa yc + xb yc}{(xb-xa)(xc-xa)(xc-xb)}, b = -\frac{-xb^2 ya + xc^2 ya + xa^2 yb - xc^2 yb - xa^2 yc + xb^2 yc}{(xb-xa)(xc-xa)(xc-xb)}, c = \frac{-xb^2 xc ya + xb xc^2 ya + xa^2 xc yb - xa xc^2 yb - xa^2 xb yc + xa xb^2 yc}{(xb-xa)(xc-xa)(xc-xb)} \right]$$

Il n'est pas évident d'exhiber une formule générale en essayant d'exploiter les résultats affichés précédemment.

L'aide du professeur sera la bienvenue dans une prochaine séance 😊 !!!!!

Quatrième étape : Utilisation du calcul matriciel et de *Scilab* pour trois points

Après toute une partie consacrée à l'introduction du calcul matriciel, qui est l'un des attendus du programme de la spécialité Mathématique, vient le temps, pour les élèves, d'exploiter leurs nouvelles connaissances.

Sur l'exemple $A(1;6)$, $B(-3;-8)$ et $C(8;-4)$, les élèves ont à résoudre l'équation matricielle $AX = B$, d'inconnue le vecteur colonne X .

$$\text{Où } A = \begin{pmatrix} 1 & 1 & 1 \\ 9 & -3 & 1 \\ 64 & 8 & 1 \end{pmatrix}, X = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \text{ et } B = \begin{pmatrix} 6 \\ -8 \\ -4 \end{pmatrix}$$

Les élèves doivent se familiariser avec la syntaxe du langage *Scilab*. Cela prend un certain temps non négligeable, d'autant plus que le langage *Python* est utilisé en enseignement obligatoire.

Ci-après la console d'un élève :

```
Console Scilab 5.5.2
-->A=[1 1 1;9 -3 1;64 8 1]
A =

    1.    1.    1.
    9.   -3.    1.
   64.    8.    1.

-->B=[6 ; -8 ; -4]
B =

    6.
   -8.
   -4.

-->X=inv(A)*B
X =

   -0.4480519
    2.6038961
    3.8441558

-->|
```

Ils peuvent constater que la fonction associée est la même, aux arrondis près, bien évidemment.

Mais l'inconvénient du calcul numérique est identifié ici : il donne des valeurs approchées des coefficients, et non les valeurs exactes 😞, comme *wxMaxima* a pu le faire précédemment .

Cinquième étape : Polynôme de Lagrange et *wxMaxima* pour trois points

Il n'est pas possible, bien évidemment, pour les élèves de trouver naturellement une formule du polynôme de Lagrange.

Je vais donc « donner » une formule qu'ils vont comparer à ce qu'ils avaient obtenu dans une séance précédente. On considère la fonction P définie par

$$P(x) = y_A \frac{(x - x_B)(x - x_C)}{(x_A - x_B)(x_A - x_C)} + y_B \frac{(x - x_A)(x - x_C)}{(x_B - x_A)(x_B - x_C)} + y_C \frac{(x - x_A)(x - x_B)}{(x_C - x_A)(x_C - x_B)} \quad (2)$$

Il est demandé aux élèves, sur l'exemple $A(1;6)$, $B(-3;-8)$ et $C(8;-4)$, de comparer la fonction P avec celle obtenue par résolution du système linéaire associé.

Cette fois-ci, les élèves renoncent, dans un premier temps à utiliser un outil informatique, sans doute car n'ayant pas trouvé comment formuler le problème en langage machine, et se lancent, de fait, dans les calculs, montrant parfois un grand désarroi.

Bien leur en a pris quand même, car ils retrouvent les coefficients affichés par *wxMaxima*, à savoir

$$a = -\frac{69}{154}, \quad b = \frac{401}{154}, \quad c = \frac{296}{77}$$

Voici une copie d'une console :

```
(%i16) P(x) := ya*(x-xb)*(x-xc)/((xa-xb)*(xa-xc))+yb*(x-xa)*(x-xc)/((xb-xa)*(xb-xc))+yc*(x-xa)*(x-xb)/((xc-xa)*(xc-xb)
(%o16) P(x) := \frac{yc(x-xa)(x-xb)}{(xc-xa)(xc-xb)} + \frac{yb(x-xa)(x-xc)}{(xb-xa)(xb-xc)} + \frac{ya(x-xb)(x-xc)}{(xa-xb)(xa-xc)}
```

```
(%i22) xa:1;ya:6;xb:-3;yb:-8;xc:8;yc:-4;
(%o17) 1
(%o18) 6
(%o19) -3
(%o20) -8
(%o21) 8
(%o22) -4
```

```
(%i23) P(x);
(%o23) -\frac{4(x-1)(3+x)}{77} - \frac{3(x-8)(3+x)}{14} - \frac{2(x-8)(x-1)}{11}
```

```
(%i24) expand(%);
(%o24) -\frac{69x^2}{154} + \frac{401x}{154} + \frac{296}{77}
```

Ils se demandent alors si la formule (2), dans le cas de trois points, peut s'appliquer à deux points. Après quelques hésitations et discussions avec le groupe, ils proposent la formule :

$$P(x) = y_A \frac{(x - x_B)}{(x_A - x_B)} + y_B \frac{(x - x_A)}{(x_B - x_A)}$$

puis la comparent à la formule (1) établie dans le premier paragraphe :

```
(%i23) Q(x) := ya*(x-xb)/(xa-xb)+yb*(x-xa)/(xb-xa);
(%o23) Q(x) := \frac{yb(x-xa)}{xb-xa} + \frac{ya(x-xb)}{xa-xb}

(%i24) f(x) := (ya-yb)*x/(xa-xb)+ya-(ya-yb)*xa/(xa-xb);
(%o24) f(x) := \frac{-(ya-yb)xa}{xa-xb} + ya + \frac{(ya-yb)x}{xa-xb}

(%i25) Q(x) - f(x);
(%o25) -\frac{xa(yb-ya)}{xa-xb} + \frac{(x-xa)yb}{xb-xa} - \frac{x(ya-yb)}{xa-xb} + \frac{(x-xb)ya}{xa-xb} - ya

(%i26) ratsimp(%);
(%o26) 0
```

Ça marche 😊 !!!!

Conclusion : Ne reste plus aux élèves qu'à généraliser et à programmer la formule établie.

Sixième étape : Écrire un programme en langage *Scilab*

Après discussion avec les élèves, la formule suivante se dégage :

Soient n un entier naturel, et $n + 1$ points M_i , i entier compris entre 0 et n , de coordonnées respectives $(x_i; y_i)$.

Il existe^a un polynôme P de degré au plus n vérifiant

Pour tout j entier, compris entre 0 et n , $P(x_j) = y_j$

à savoir

$$P(X) = \sum_{i=0}^n y_i L_i(X) = \sum_{i=0}^n y_i \prod_{i \neq j} \frac{(X - x_j)}{(x_i - x_j)}$$

a. L'unicité du polynôme P est mise de côté ici.

Programmation de la formule

Le langage choisi par le professeur est celui du logiciel de calcul numérique *Scilab*.

Trois raisons à ce choix :

- plusieurs élèves l'ont découvert et manipulé en classe de seconde, et certains utilisent en cours de SI son homologue payant, à savoir *Matlab*,
- la syntaxe est très proche de celle de *Matlab*,
- il enrichit la panoplie des outils informatiques susceptibles d'être utilisés par un futur étudiant en sciences.

Voici un code mettant en scène la détermination des coefficients du polynôme P et une application à l'exemple $A(1;6)$, $B(-3;-8)$ et $C(8;-4)$:

Le code source :

```
1 function P=Flagrange(X,Y)//X est le vecteur qui contient le x_i et Y c
  elui qui contient les y_i;
2 //P est le polynôme d'interpolation de Lagrange
3 n=length(X);// n+1 est le nombre de points. n sera donc le degré du po
  lynôme P.
4 x=poly(0,"x");P=0;//x est l'indéterminée du polynôme P
5 for i=1:n, L=1;
6 for j=[1:i-1,i+1:n] L=L*(x-X(j))/(X(i)-X(j));end //on a effectué les d
  ifférences divisées
7 P=P+L*Y(i);
8 end
9 disp(P)// on affiche P
10 endfunction
11 //on prend 3 points (1;6) , (-3;-8) et (8;-4) [voir l'exemple traité en
  début de document]
12 X=[1,-3,8]
13 Y=[6,-8,-4]
14 P=Flagrange(X,Y)
```

Le résultat affiché dans la console :

```
-->exec('/media/jm/CSDC-39E4/ANNEE-2017-2018/Approfondisse
Attention : Redéfinition de la fonction : Flagrange
```

```
3.8441558 + 2.6038961x - 0.4480519x2
```

```
-->
```

Ça marche 😊 !!!

Remarques :

- Sans l'aide et la participation du professeur, l'écriture du code aurait constitué un obstacle important, voir insurmontable, pour les élèves (voir les commentaires qui suivent sur les difficultés rencontrées par les élèves pour coder un algorithme répondant au problème). Cette considération n'est absolument pas à négliger dans la réalisation d'un tel travail.
- Les élèves ont pu enfin lire la fiche *Température piscine* et répondre au problème posé.

Quelques difficultés rencontrées par les élèves pour coder l'algorithme

- `function P=Flagrange(X,Y)`
Il n'est pas naturel pour les élèves de définir une fonction, encore moins de choisir les paramètres X et Y en tant que vecteurs.
- `n=length(X)`
Ils oublient souvent, dans un premier temps, de déclarer la variable n, ce qui leur posera pour les deux boucles **for** un problème d'arrêt.
- `x=poly(0,"x");P=0`
La syntaxe est donnée par le professeur, sinon, le problème devient insurmontable.
- `for i=1 :n, L=1`
La première boucle apparaît assez naturellement.
- `for j=[1 :i-1,i+1 :n]`
Le problème est de ne pas oublier que $j \neq i$, mais aussi de trouver la bonne syntaxe pour traduire la contrainte. Le professeur donne la réponse.
- `L=L*(x-X(j))/(X(i)-X(j))`
Les élèves sont parfois bloqués dans le calcul de la nouvelle valeur de la variable L en fonction de l'ancienne, donc dans l'écriture de l'affectation.
- `end`
- `P=P+L*Y(i)`
Même constat que précédemment.
- `end`
- `disp(P)`
- `endfunction`
- `X=[1,-3,8]`
- `Y=[6,-8,-4]`
- `P=Flagrange(X,Y)`
Les élèves ont déclaré la fonction **Flagrange**, mais ensuite, que font-ils ? Il ne leur est pas toujours évident, ni naturel, de l'appeler en ayant pris soin de définir au préalable les vecteurs X et Y.

5 Un peu de théorie

Il n'est pas question ici de décrire exhaustivement la théorie algébrique permettant de définir et de déterminer les polynômes de Lagrange, mais de sensibiliser les élèves de TS, peut-être futurs étudiants en CPGE ou à l'université, à des notions et méthodes qu'ils parcourront plus en détail et rigoureusement l'an prochain. Ce qui suit ne donne que des pistes à explorer bien plus rigoureusement que je ne l'ai fait.

Cette partie a été exposée aux élèves trop rapidement sans doute, et nécessiterait un retour pour consolider les notions nouvelles.

5.1 Symboles de Kronecker

Définition : Symbole de Kronecker

Soit I un ensemble d'indices.

On appelle **symbole de Kronecker** la fonction $\delta : I \times I \rightarrow \{0, 1\}$ définie par :

$$\begin{cases} \delta(i, j) = 1 & \text{si } i = j \\ \delta(i, j) = 0 & \text{si } i \neq j \end{cases}$$

Il est d'usage de remplacer la notation $\delta(i, j)$ par δ_{ij} .

5.2 Polynômes de Lagrange

Définition : Polynômes de Lagrange

Soit $I = \llbracket 1; n \rrbracket$ et x_1, x_2, \dots, x_n , n réels distincts.

Les polynômes L_1, L_2, \dots, L_n , définis par $L_i(X) = \prod_{i \neq j} \frac{(X - x_j)}{(x_i - x_j)}$,

sont appelés **Polynômes de Lagrange** des x_1, \dots, x_n .

Remarque

On vérifie facilement que, pour tout $i \in I$ et $j \in I$, $L_i(x_j) = \delta_{ij}$

5.3 Polynômes d'interpolation de Lagrange

Propriété : Polynômes d'interpolation de Lagrange

Soit $I = \llbracket 1; n \rrbracket$. Soient x_1, x_2, \dots, x_n , n réels distincts et y_1, \dots, y_n , n réels.

Le polynôme $P(X) = \sum_{i=1}^n y_i L_i(X)$ est l'unique polynôme de degré inférieur ou égal à $n - 1$ vérifiant, $\forall i \in I$, $P(x_i) = y_i$.

Démonstration : $I = \llbracket 1; n \rrbracket$.

- **Existence** Soient $P = \sum_{i=1}^n y_i L_i$

Les polynômes $(L_i)_{i \in I}$ sont de degré $n - 1$ et donc, par somme, P est de degré inférieur ou égal à $n - 1$.

De plus, $\forall j \in I$, $P(x_j) = \sum_{i=1}^n y_i L_i(x_j) = \sum_{i=1}^n y_i \delta_{ij} = y_j$

- **Unicité** Soient P_1 et P_2 deux polynômes de degrés inférieurs ou égaux à $n - 1$, vérifiant , $\forall i \in I$, $P_1(x_i) = P_2(x_i) = y_i$.

Le polynôme $P_1 - P_2$ admet x_1, \dots, x_n pour racines distinctes, donc le polynôme $P_1 - P_2$ admet au moins n racines.

Or, le degré de $P_1 - P_2$ est inférieur ou égal à $n - 1$, donc, nécessairement $P_1 - P_2$ est le polynôme nul, et donc $P_1 = P_2$.

Remarque : Il est nécessaire de définir très rigoureusement la notion de polynôme pour une formalisation théorique des théorèmes utilisés ici, permettant de faire le lien, entre autre, entre α racine d'un polynôme, ordre de multiplicité de α et divisibilité de P par $X - \alpha$.

6 Conclusions

Voici quelques conclusions tirées de cette expérience :

- Le projet est réalisable avec des élèves « motivés » de TS, plus particulièrement ceux qui suivent la spécialité Mathématique.
- Il faut, en amont du projet, préparer les élèves à l'utilisation du calcul formel et d'un langage de programmation, ce qui, dans le nouveau programme de la classe de seconde appliqué depuis la rentrée 2017, puis du cycle Terminale qui suit, en attendant le contenu des nouveaux programmes liés à la réforme du lycée à venir, est un préalable incontournable.
- La démarche d'investigation est au cœur du projet, mais le prix à payer pour le professeur est d'accepter l'incertitude du déroulement de celui-ci. Comment les élèves réagissent devant un problème donné ? Il est demandé au professeur beaucoup de réactivité.....
- Ce projet peut être réalisé dans le cadre de séances d'AP en approfondissement.
- Et, de fait, peut aider un élève de TS à consolider son projet d'orientation post-bac.
- L'expérience montre que les élèves ont appréhendé le problème de l'interpolation en empruntant parfois des chemins différents, n'utilisant pas les mêmes outils, mais ils se sont mutuellement enrichis du travail et des remarques de chacun.
- En ce qui concerne la question de savoir comment lever les blocages des élèves pour coder, il faut en amont que les élèves aient été familiarisés à l'algorithmie, et à la syntaxe d'un langage choisi, si possible, par l'équipe pédagogique du lycée. Mais les nouveaux programmes iront, je l'espère, dans le sens de ma proposition.
- Ce qui m'incite à vous passer ceci :

*« Il faut prendre conscience de l'apport d'autrui, d'autant plus riche que la différence avec
soi-même »*

Albert Jacquard

7 Annexe

Les différents algorithmes qui précèdent ont été codés en langage *Scilab* .

Il m'a paru indispensable de proposer ces algorithmes codés en langage *Python*, pour répondre aux exigences du programme officiel, qui préconise l'utilisation de ce langage à partir de la classe de seconde.

Il faut remarquer que les deux codes qui suivent utilisent la librairie *sympy* que l'on peut découvrir en suivant ce lien <http://live.sympy.org/>.

7.1 Code qui donne les valeurs exactes des coefficients

On exécute le code suivant :

```
from math import*
from sympy import*
X=[1,-3,8]
Y=[6,-8,-4]
# X est le vecteur qui contient le x_i et Y celui qui contient les y_i;
# P sera le polynôme d'interpolation de Lagrange
def Flagrange(X,Y):
    n=len(X)
# n+1 est le nombre de points. n sera donc le degré du polynôme P
    x=symbols("x")
    P=0.
# x est l'indéterminée du polynôme P
    for i in range(0,n):
        L=1
        for j in range(0,n):
            if i==j: continue
            L=L*(x-X[j])/(X[i]-X[j])
# on a effectué les différences divisées
        P=P+L*Y[i];
# l'affectation qui suit permet, grâce à la fonction simplify, de renvoyer les coefficients
# sous forme de fractions irréductibles
    return simplify(P)
```

La commande `print(Flagrange(X,Y))` dans la console affiche alors en sortie

$$-69*x**2/154 + 401*x/154 + 296/77$$
 None

Ce qui donne
$$P(x) = -\frac{69x^2}{154} + \frac{401x}{154} + \frac{296}{77}$$

On pourra constater que les affichages donnent les mêmes valeurs exactes des coefficients, que l'on utilise le calcul formel sous *wxMaxima*, ou en langage *Python* en important la librairie *sympy*.

7.2 Code qui affiche les coefficients approchés sous forme décimale

On exécute le code suivant :

```
from math import*
from sympy import*
X=[1,-3,8]
Y=[6,-8,-4]
# X est le vecteur qui contient le x_i et Y celui qui contient les y_i;
# P sera le polynôme d'interpolation de Lagrange
def Flagrange(X,Y):
    n=len(X)
# n+1 est le nombre de points. n sera donc le degré du polynôme P
    x=symbols("x")
    P=0.
# x est l'indéterminée du polynôme P
    for i in range(0,n):
# On affecte à L le flottant 1. , ce qui aura pour conséquence de renvoyer des flottants pour la suite
# du programme
        L=1.
        for j in range(0,n):
            if i==j: continue
            L=L*(x-X[j])/(X[i]-X[j])
# on a effectué les différences divisées
        P=P+L*Y[i]
    return simplify(P)
```

La commande `print(Flagrange(X,Y))` dans la console affiche alors en sortie

```
-0.448051948051948*x**2 + 2.6038961038961*x + 3.84415584415584 None
```

Ce qui donne $P(x) = -0.448051948051948x^2 + 2.6038961038961x + 3.84415584415584$

On pourra constater que les affichages donnent des arrondis des coefficients identiques, que l'on travaille en langage *Scilab* ou en langage *Python*.