Les Polynômes

Résumé

Ce nouveau TP Maths–Info (TPMI, prononcez [tpmi]) est l'occasion de pratiquer vos deux matières préférées : l'algèbre et la programmation ! En effet, le programme du premier semestre étant particulièrement chargé en algèbre, il manquait une notion essentielle : les polynômes. Vous savez bien évidemment qui ils sont, vous les connaissez (presque) tous, mais nous allons vous faire découvrir leur secrets au travers de cet atelier.

1 Introduction

L'arithmétique du premier semestre vous a permis de (re)découvrir des notions fondamentales telles que la divisibilité, les nombres premiers entre eux, les nombres premiers tout court... La fin du cours vous a amené à de nombreuses propositions mais surtout au théorème fondamental de l'arithmétique. Il s'avère en fait que l'arithmétique ne se limite pas à l'étude des entiers : les polynômes y ont également toute leur place.

1.1 Définition d'un polynôme

Commençons tout d'abord par rappeler quelques définitions relatives à ces objets.

Définition 1 (Polynôme). Un *polynôme* P à coefficients dans $\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$ est une expression de la forme $P(X) = a_nX^n + \cdots + a_1X + a_0$ avec $a_i \in \mathbb{K}$ avec
— a_i *coefficient de degré* i du polynôme P, $0 \leq i \leq n$,
— et X l'*indéterminée*.

Notons que la lettre $\mathbb{K}$ représente la plupart du temps l'ensemble $\mathbb{R}$ ou $\mathbb{C}$. Ce sera le cas pour toute la suite de ce cours. Notons également que l'ensemble des polynômes à une indéterminée X et à coefficients dans $\mathbb{K}$ est noté $\mathbb{K}[X]$.

Exemple 2. Ainsi, le polynôme $P(X) = 3X^2 - 2X + 1 \in \mathbb{R}[X]$ défini sur $\mathbb{R}$ signifie que les coefficients 3, -2 et 1 sont des éléments de $\mathbb{R}$. Le coefficient de degré 2 vaut 3, celui de degré 1 vaut -2 et celui de degré 0 vaut 1.

Nous pouvons remarquer que si la définition d'un polynôme telle que nous l'avons présentée colle parfaitement à l'usage que l'on en fait, elle manque quelque peu de formalisme. En effet, qu'est-ce qu'une expression en mathématiques ? Pour cela, il est peut-être plus simple de se demander comment représenter un polynôme en algorithmique.

Exercice 1.
L'objectif de cet exercice est de trouver une représentation simple de polynôme, qui soit compatible au niveau mathématique et au niveau programmation. Supposons que l'on demande à un utilisateur un polynôme P à coefficients dans $\mathbb{R}$.

1) Quelle est la première question à lui poser afin de préparer un espace de stockage suffisamment grand pour P ?
2) Que va ensuite saisir l'utilisateur ?
3) Dans quoi va-t-on stocker sa réponse ?
4) Définissez une structure de données nommée `Polynome` qui permettra de stocker l'ensemble des informations nécessaires à la manipulation d'un polynôme. Si besoin, aidez-vous de la section 3.1.

On tire donc de l'exercice précédent la définition suivante pour un polynôme.

Définition 3 (Polynôme). Un *polynôme* de $\mathbb{K}[X]$ est une suite presque nulle[1] $(a_i)_{i\in\mathbb{N}}$ d'éléments de $\mathbb{K}$

Ainsi, le polynôme $P(X) = 3X^2 - 2X + 1$ se note aussi $P(X) = 1, -2, 3, 0, 0, \ldots$ Il manque maintenant quelques définitions importantes pour avoir le même langage.

1.2 Définitions annexes

Comment nommer la « taille » d'un polynôme ? Peut-il être segmenté ? Existe-t-il des éléments plus importants que d'autres ? Tant de questions qui taraudent votre esprit la nuit ! Heureusement, les définitions suivantes vont vous apporter une réponse qui devrait vous satisfaire.

Définition 4.

— Le *degré* d'un polynôme P est le plus petit entier n tel que $a_i = 0$ pour tout $i > n$. Il est noté $\deg(P)$.
— Par convention, le degré du polynôme nul est $-\infty$.
— Le coefficient de degré $\deg(P)$ d'un polynôme P est appelé *coefficient dominant*.
— Un polynôme dont le coefficient dominant est égal à 1 est dit *unitaire*.
— Un *monôme* est un polynôme dont seul un terme de la suite est non nul. Il est donc de la forme $P = a_k X^k$.
— Le *terme dominant* d'un polynôme est le monôme de plus haut degré.

Exemple 5. Étudions le polynôme $P(X) = 2X^7 - \frac{9}{4}X^3$. Son *degré* est 7 puisque s'il est rédigé sous forme de suite, tous les termes d'indice supérieurs à 7 sont nuls. Son *coefficient dominant* est 2 et il n'est donc *pas unitaire*. Il est composé de la somme des monômes $2X^7$ et $-\frac{9}{4}X^3$ et son terme dominant est $2X^7$. ◢

Exercice 2.
Considérons le polynôme $P = 2X^6 + 3$. Complétez les phrases suivantes.

1) P a pour coefficient dominant …
2) P a pour terme dominant …
3) P est de degré …
4) P … un monôme.
5) P … unitaire.

Exercice 3.
Pour chacune des questions suivantes, donnez un exemple de polynôme P dans $\mathbb{R}[X]$ vérifiant la propriété souhaitée.

1. suite dont tous les termes sont nuls à partir d'un certain rang

1) P est un monôme de degré 3.
2) P est un polynôme unitaire.
3) $\deg(P) = 0$.
4) P est un monôme unitaire de terme dominant X.

Exercice 4.
L'objectif de cet exercice est de rédiger un ensemble de fonctions permettant de manipuler les polynômes. Pour cela, la définition d'un polynôme sera réalisée au travers d'un fichier de configuration. La première ligne donne le degré d du polynôme. La ligne suivante donne les coefficients par ordre décroissant de degré. Par exemple, le fichier de configuration suivant permet de générer le polynôme $-3X^4 + 7X^3 - 2X$.

```
4
-3 7 0 -2 0
```

Pour cela, vous devez rédiger la fonction `ChargerPolynome()` prenant les paramètres suivants :
- une chaine de caractères `path` donnant le chemin vers le fichier de configuration,
- le polynôme à compléter, passé par adresse.

Cette fonction doit renvoyer la valeur 0 si tout s'est bien passé ou un code d'erreur sinon. Pour cela, suivez les instructions suivantes, et aidez-vous si besoin de la section 3.1.
1) Ouvrez le fichier `path` et vérifiez que l'ouverture s'est correctement réalisée.
2) Lisez la première valeur du fichier pour stocker la taille de votre tableau de coefficients.
3) Allouez le tableau de coefficients.
4) Poursuivez la lecture de la ligne afin de remplir chaque case du tableau.
5) Fermez le fichier puis retournez 0 si tout s'est bien passé.

Rédigez également les fonctions `LibererPolynome()` et `AfficherPolynome()` permettant respectivement de libérer la mémoire allouée pour un polynôme et afficher les coefficients du polynôme passé en argument. Testez votre code sur plusieurs exemples avant de continuer ce TP.

2 Opérations sur les polynômes

Maintenant que la définition d'un polynôme est parfaitement maitrisée, il est temps de voir quelles opérations sont accessibles. Tout comme les réels, les polynômes ont des droits. Ils peuvent s'additionner, se multiplier (même avec des réels) et se comparer. Mais avant tout, il est important de définir l'égalité de deux polynômes.

Définition 6 (Égalité). Deux polynômes de $\mathbb{K}[X]$ sont égaux si et seulement si leurs coefficients sont égaux.

Au contraire de l'égalité, la comparaison de deux polynômes n'est pas particulièrement intuitive. Pourtant, elle est nécessaire, particulièrement dans la division euclidienne de deux polynômes (car oui, deux polynômes ont le droit de se diviser).

Exercice 5.
Nous cherchons à définir la relation « plus grand que » entre deux polynômes. Pour cela, considérons deux polynômes P et Q à coefficients dans $\mathbb{K}[X]$.
1) D'après vous, quels sont les éléments comparables ?
2) Qui est le plus grand ? X ou $2X$?

3) Qui est le plus grand ? $2X + 1$ ou $X + 2$?
4) Qui est le plus grand ? $(i + 1)X$ ou $\pi X + \sqrt{2}$?
5) Quels sont les éléments *vraiment* comparables ?
6) Non mais, en vrai, quels sont les éléments comparables ? En plus c'est pas dur...

Exercice 6.
Implémentez la fonction `EgalitePolynome` permettant tester l'égalité des deux polynômes passés en argument. Elle renvoie 1 en cas d'égalité, 0 sinon.

2.1 Addition

Il existe trois opérations standard de manipulation des polynômes qui sont l'addition (facile), la multiplication par un scalaire (facile aussi) et le produit de deux polynômes (moins facile). Voyons maintenant ces trois opérations ainsi que quelques exemples.

Définition 7 (Addition). Soient $P = \sum_i a_i X^i$ et $Q = \sum_i b_i X^i$ deux polynômes de $\mathbb{K}[X]$. Leur somme est définie par :

$$P + Q = \sum_i (a_i + b_i) X^i.$$

Exemple 8. Calculons la somme des polynômes $P = 2X^4 + X^3 + 4X + 2$ et $Q = -2X^4 + 3X^2 - X + 5$. On a alors

$$P + Q = (2 - 2)X^4 + X^3 + 3X^2 + (4 - 1)X + 2 + 5 = X^3 + 3X^2 + 3X + 7.$$

Exercice 7.
Nous cherchons maintenant à trouver la propriété reliant une somme de polynôme et le degré des polynômes y intervenant. Pour cela,considérons les polynômes suivants.

$$\begin{aligned} P &= 4X^3 + X - 2, \\ Q &= -X^5 + 3X^3 + X^2 - 3X - 1, \\ R &= -X^4 - 2X^3 + 5X^2 + 4. \end{aligned}$$

1) Calculez les sommes $P + Q$, $Q + R$ et $P + Q + R$.
2) Quel est le degré des sommes précédemment calculée ?
3) Déduisez-en la propriété comparant $\deg(P + Q)$ avec $\deg(P)$ et $\deg(Q)$.

Exercice 8.
Complétez votre librairie sur les polynômes en rédigeant la fonction suivante.

```
int AddPolynome (Polynome P, Polynome Q, Polynome *Res);
```

Votre fonction permet de réaliser le calcul `Res = P+Q`. Elle renvoie 0 en cas de succès, un code d'erreur sinon. Notez que `Res` est passé par adresse. Testez votre code en vérifiant les calculs de l'exercice 7 et vérifiez vos calculs en lançant votre code.

2.2 Multiplication

Sur les entiers, la multiplication d'un entier n par un facteur k est définie comme la somme du terme n, k fois. On écrit alors $kn = n + n + \cdots + n$. Cette multiplication est appelée pour les polynômes *multiplication par un scalaire*.

Définition 9 (Multiplication par un scalaire). Soient $P = \sum_i a_i X^i$ dans $\mathbb{K}[X]$ et $\lambda \in \mathbb{K}$. La multiplication de P par λ est définie par :

$$\lambda P = \sum_i \lambda a_i X^i$$

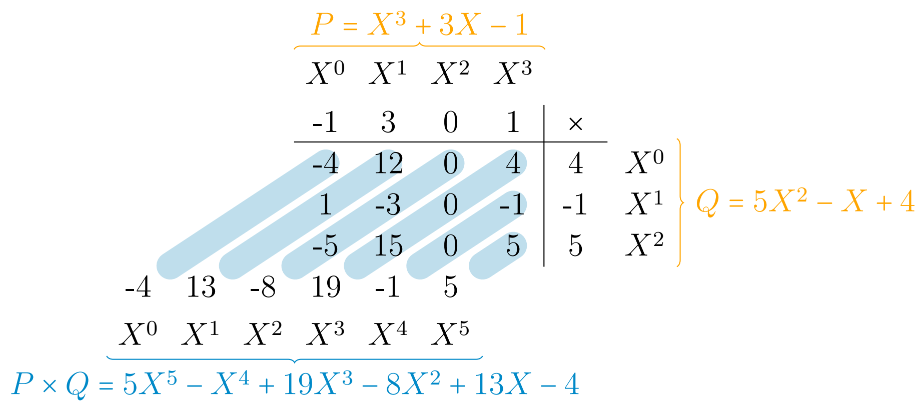
Le produit de deux polynômes est quant à lui plus complexe, mais vous l'avez déjà manipulé au lycée. En effet, il suffit d'appliquer la distributivité de la multiplication sur la somme des monômes et le tour est joué ! Sauf que ce petit jeu peut s'avérer long suivant la taille du polynôme.

Exercice 9.

Nous cherchons à définir la multiplication de deux polynômes. Pour cela, considérons les polynômes $P = X^3 + 3X - 1$ et $Q = 5X^2 - X + 4$.

1) A l'aide des règles de distributivité, calculez le produit $R = PQ$.
2) Quel est le degré du polynôme R ?
3) Déduisez-en la propriété donnant le degré d'un produit de deux polynômes.

En rappelant qu'un polynôme s'écrit aussi sous la forme d'une suite presque nulle de coefficients, il est possible de trouver une astuce pour calculer plus rapidement la multiplication de deux polynômes P et Q. On définit pour cela un tableau T de $\deg(Q)+1$ lignes et $\deg(P)+1$ colonnes. La figure suivante vous donne un exemple en utilisant les polynômes de l'exercice 9.



Rappelons que le tableau `T` est indexé de 0 à $\deg(Q)$ suivant les lignes et de 0 à $\deg(P)$ suivant les colonnes. Notons les polynômes P et Q de la façon suivante.

$$P = p_0 + p_1 X + p_2 X^2 + \cdots + p_m X^m \text{ et } P = q_0 + q_1 X + q_2 X^2 + \cdots + q_n X^n.$$

Dans notre exemple, p_0 vaut donc -1 et $q_2 = 5$. La tableau `T` permet de calculer facilement le polynôme $P \times Q$. En effet, la case en ligne `i`, colonne `j` donne la valeur du coefficient $q_i p_j$ associé au monôme de degré `i + j`. Il est donc aisé de calculer tous les produits des coefficients.

La suite de l'algorithme consiste à additionner les coefficients associés à un même monôme. Pour cela, le tableau se parcourt par diagonale. En effet, si la case `T[i][j]` donne un coefficient associé à un monôme de degré `i + j`, la case `T[i-1][j+1]` aussi, de même que `T[i+1][j-1]` ou `T[i-2][j+2]`. Sommer les termes d'une diagonale revient donc à calculer un coefficient du polynôme produit.

Exercice 10.

1) À l'aide de l'exemple précédent, calculez le produit $(X^2 + 3X - 1) \times (-X + 2)$.
2) À l'aide de l'exemple précédent, calculez le produit $(X^4 - 2X^3 + X) \times (-X^2 + X + 1)$.

L'exercice précédent nous donne donc la proposition suivante permettant de calculer le produit de deux polynômes.

Proposition 10 (Multiplication). Soient $P = \sum_i a_i X^i$ et $Q = \sum_i b_i X^i$ deux polynômes de $\mathbb{K}[X]$. Leur produit est défini par :

$$PQ = \sum_k \left(\sum_{i=0}^{k} a_i b_{k-i} \right) X^k.$$

Preuve. Considérons les polynômes P et Q suivants, de degré respectifs m et n.

$$P[X] = a_0 + a_1 X + \cdots + a_m X^m + \ldots$$
$$Q[X] = b_0 + b_1 X + \cdots + b_n X^n + \ldots$$

On a alors

$$PQ = \sum_{i,j} a_i b_j X^{i+j} = \sum_k \sum_{i+j=k} a_i b_j X^k = \sum_k \left(\sum_{i=0}^{k} a_i b_{k-i} \right) X^k,$$

ce qui conclut la démonstration. ■

Exercice 11.

Complétez votre librairie sur les polynômes en rédigeant la fonction suivante.

```c
int MultPolynome (Polynome P, Polynome Q, Polynome *Res);
```

Votre fonction permet de réaliser le calcul `Res = PQ`. Elle renvoie 0 en cas de succès, un code d'erreur sinon. Notez que `Res` est passé par adresse. Pour la coder, n'utilisez pas la matrice mais plutôt le résultat de la proposition 10. Testez votre code en vérifiant les calculs de l'exercice 10.

3 Annexes

3.1 Allocation de polynômes

Lors de ce TP, vous devez manipuler des polynômes au travers de pointeurs. Pour cela, leur allocation est cruciale. Rappelons que le but de l'allocation est de réserver l'espace mémoire suffisant pour stocker une donnée. Ici, nous avons besoin de stocker :

— un entier correspondant au degré du polynôme

— un tableau de coefficients réels.

Le polynôme étant entièrement défini par l'utilisateur, la taille du tableau de coefficients n'est pas constante d'une exécution à l'autre. Il faut donc allouer le tableau de coefficients de façon dynamique.

Ces deux données (le degré et le pointeur sur le tableau de coefficients) sont particulièrement liées. Il est donc logique de les regrouper sous une structure. Le code suivant permet donc de définir un polynôme au travers d'une structure nommée `Polynome`.

```c
typedef struct Polynome_s
{
  int deg;        // degre du polynome
  float *coeffs;  // tableau des coefficients
}Polynome;
```

Lors de la création d'une fonction modifiant un polynôme, il est obligatoire de réaliser un passage par adresse de la variable, sans quoi le degré ne peut pas être modifié. Notez qu'il est toujours possible de modifier les valeurs stockées dans le tableau car son adresse est stockée dans le pointeur `coeffs`. Un exemple d'une telle fonction est `ChargerPolynome` qui lit un fichier de configuration et remplit le polynôme passé en argument. Le prototype de cette fonction étant `ChargerPolynome (char *path, Polynome *p)`, l'accès au degré s'écrit donc `p->deg`.