# TP2. Polynômes.

**NOM :**
**PRÉNOM :**

*Lancer* `xmaple`.
*Si vous êtes perdu, utiliser l'aide de maple : ? suivi du nom de la commande.*
*Écrivez vos réponses dans les encadrés de cette feuille. Vous pouvez aussi m'envoyer votre fichier maple par mail (*`francois.chapon@math.univ-toulouse.fr`*), au format L3TP2-NOM-PRENOM.mw si vous mettez des commentaires pour numéroter les questions et les exercices à l'aide de* `#`.

Histoire de bien commencer, taper `print(‘Bonjour‘);` *(attention, c'est un accent grave).*

**Exercice 1 :** On commence en douceur.

a) Écrire une procédure, à l'aide de conditions `if...then...else`, donnant la ou les solutions, quand elles existent, de l'équation du second degré

$$ax^2 + bx + c = 0,$$

avec $a, b, c \in \mathbb{R}$, $a \neq 0$.

```
soltrinome :=proc(a,b,c)
```

*N'oubliez pas la condition $a \neq 0$.*

b) Donner les valeurs exactes et approchées des solutions de l'équation $73x^2 + 18x - 17 = 0$.

**Exercice 2 :** Soit $P$ un polynôme de la forme $P(x) = \sum_{k=0}^{n} a_k x^k$.

a) Quel est le nombre totale d'opérations à effectuer pour évaluer $P(x)$ pour un certain $x$ donné ?

b) On définit les listes en maple à l'aide de crochets `[...,...]`. Par exemple, tester

```
l:=[1,2,-12,3];
```

Le nombre d'éléments de la liste `l` est alors donné par la commande `nops(l)` (ici 4), pour $i$ allant de 1 à $n$ le $i^{\text{e}}$ élément de la liste est obtenu par la commande `op(i,l)`, et tous les éléments de $l$ sont obtenus par `op(l)`. Pour notre exemple, on obtient donc

```
> nops(l);  4
> op(3,l); -12
> op(l);  1,2,-12,3
```

Plutôt que de définir un polynôme comme une fonction, on définira un polynôme en une indéterminée $x$ en tant que expression formelle. Par exemple, pour définir le polynôme $P(x) = 1-2x^2+3x^3$, on fera

```
P:=1-2*x^2+3*x^3;
```

La valeur de $P$ en par exemple 2 est alors donnée par `subs(x=2,P)`. Le degré de $P$ s'obtient par `degree(P)` et le coefficient de $x^k$ par `coeff(p,x,k)`.
Écrire une fonction `listecoeffs` qui a un polynôme $p$ associe une liste dont les éléments sont les coefficients de $p$.

c) En remarquant que

$$\begin{aligned} P(x) &= a_0 + x(a_1 + a_2 x + \cdots + a_n x^{n-1}) \\ &= a_0 + x(a_1 + x(a_2 + \cdots + a_n x^{n-2})) \\ &\vdots \\ &= a_0 + x(a_1 + x(a_2 + x(\cdots(a_{n-1} + a_n x)))), \end{aligned}$$

écrire une procédure qui pour une liste $[a_0, a_1, \dots, a_n]$, évalue le polynôme $P$ au point $x$ (algorithme de Horner).

```
HornerPoly :=proc(y : :list)
```

d) Vérifier la procédure à l'aide de la fonction `expand` pour le polynôme associé à la liste $[1, 0, -4, 7, 8, 2, 1]$.

e) Quel est le nombre totale d'opérations à effectuer dans l'algorithme de Horner pour évaluer $P(x)$ ? Comparer avec la question a).

**Exercice 3 :** La règle de Descartes permet de comparer le nombre de racines strictement positives d'un polynôme avec le nombre de changement de signes de la suite des coefficients du polynôme, c'est-à-dire

**Définition :** Soit $a = (a_0, \ldots, a_n)$ une suite finie d'éléments de $\mathbb{R}^*$. Le nombre de changement de signes de $a$, noté $N(a)$, est défini récursivement par $N(a_0) = 0$, et pour tout $n \geq 1$

$$N(a_0, \ldots, a_n) = \begin{cases} N(a_1, \ldots, a_{n-1}) + 1, & \text{si } a_{n-1}a_n < 0, \\ N(a_1, \ldots, a_{n-1}), & \text{sinon.} \end{cases}$$

On étend cette définition à une suite finie $a$ d'élément de $\mathbb{R}$ en considérant la suite $b$ obtenue en supprimant les zéros de $a$, et en définissant alors $N(a) = N(b)$.
On a alors le théorème suivant.

**Théorème :** Le nombre $\mu(P)$ de racines strictement positives (comptées avec leur ordre de multiplicité) d'un polynôme $P(x) = \sum_{i=0}^{n} a_i x^i$ est majoré par $N(a_0, \ldots, a_n)$. De plus, la différence $N(a_0, \ldots, a_n) - \mu(P)$ est paire. Si de plus, le polynôme $P$ a toutes ses racines réelles, alors on a l'égalité $\mu(P) = N(a_0, \ldots, a_n)$.

Exemple : Le polynôme $P(x) = 1 - 2x - x^2 + x^4$ admet deux changements de signes, et donc 0 ou 2 racines strictement positives.

a) Écrire une procédure, qui a une liste $a$ d'éléments de $\mathbb{R}$ associe une liste obtenue en supprimant les zéros de $a$. On pourra initialiser la liste par la liste nulle `[]`, et utiliser le symbole `<>` pour $\neq$.

```
listezero :=proc(a : :list)
```

b) Écrire une procédure qui à une liste $[a_0, \dots, a_n]$ définissant les coefficients du polynôme $P(x) = \sum_{i=0}^{n} a_i x^i$, associe le nombre de changement de signes $N(a_0, \dots, a_n)$.

```
Descartesrules :=proc(a : :list)
```

On pourra par exemple tester la procédure sur le polynôme $P(x) = 2 + 3x - 6x^2 + 8x^4 - 3x^6 - 5x^7 + 2x^{10} - x^{11} - 2x^{13} - 7x^{14}$.

c) Définir le polynôme $P(x) = \sum_{i=1}^{30} \sin(\frac{200}{i}) x^{i-1}$. À l'aide des procédures précédentes, donner le nombre de changements de signes de la suite de ses coefficients. On pourra utiliser la commande `evalf` pour évaluer une liste.

**Exercice 4 :** Les polynômes de Tchebychev (de première espèce) $(T_n)_{n\geq 1}$ peuvent être définis de la façon suivante : $T_0(x) = 1$, $T_1(x) = x$, et pour $n \geq 2$,

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x).$$

a) Écrire une procédure construisant les polynômes de Tchebychev.

```
Tcheby :=proc(n)
```

b) Tester l'expression des polynômes $T_n$ en $\cos\theta$, pour différentes valeurs de $n$. On pourra utiliser les commandes `expand` et `combine` pour simplifier les expressions. Conjecturer alors l'expression de $T_n(\cos\theta)$, et donner une preuve.

c) En déduire les racines de $T_n$.