Nous vous proposons aujourd'hui, après une séance de remise en forme sur la manipulation des polynômes sous Maple, de programmer le très fameux algorithme d'Euclide pour le calcul du pgcd de deux polynômes. Nous étendrons ensuite cet algorithme afin d'obtenir un couple de solutions de la relation de Bézout.

1 Rappels sur les polynômes

Nous avons repris ici quelques lignes proposées au TD1 qui permettent de se familiariser avec les polynômes. Assurez-vous de bien comprendre la gestion faite par Maple des objets que vous manipulez.

```
[> P:=X^2+X+1;
[> P(Y);
[> subs(X=1,P);X;P;
[> X:=1;P;
[> Q:=X^3+1;
[> X:=2;P;Q;
[> X:='X';P;Q;
[> diff(P,X);diff(P,X$2);
```

Il est plus simple de ne **jamais** affecter de valeur à `X`, et d'utiliser `subs` pour évaluer un polynôme en un point.

Testez les outils suivant : `degree, expand, simplify, diff, factor, coeff, lcoeff`, et assurez-vous que vous savez les utiliser en faisant les exercices suivants (attention : certaines questions sont une utilisation directe des outils précédents, d'autres demandent de creuser un peu plus...).

Exercice 1 *Quels sont les degrés et les coefficients dominants des polynômes suivants :*
$(1-(2+X)^6)^3$
$(X+1)^{100}-(X-1)^{100}$

Exercice 2 *Donner l'ordre de multiplicité de la racine 1 dans les polynômes suivants :*
$X^{2n+1}-(2n+1)X^{n+1}+(2n+1)X^n-1$
$X^{2n}-n^2X^{n+1}+2(n^2-1)X^n-n^2X^{n-1}+1$

Exercice 3 *Factorisez les polynômes suivants :*
$3X^3-2X^2-3X+2$
$16X^5-20X^3+5X-1$

Exercice 4 (facultatif) *Soient $P=a_0+\ldots+a_nX^n$ et $b\in\mathbb{C}$. On définit par récurrence les complexes $(q_k)_{0\leq k\leq n}$ par $q_0=a_n$, et $q_{k+1}=q_kb+a_{n-k-1}$. Montrer que l'on a : $q_n=P(b)$.*
Il s'agit de la méthode de Hörner, utilisée en informatique pour évaluer les polynômes avec un minimum de calculs. Écrivez une procédure qui calcule la valeur d'un polynôme P en un point b. (Indication : il peut être plus simple de supposer que P est donné sous la forme d'une liste de ses coefficients).

2 Algorithme d'Euclide

Soient A et B deux polynômes tels que $degre(A)\geq degre(B)$.
On définit les deux suites de polynômes $(A_n)_{n\geq 0}$ et $(B_n)_{n\geq 0}$ ainsi :

$$\begin{cases} A_0=A \quad A_{n+1}=B_n \\ B_0=B \quad B_{n+1}\equiv A_n[B_n], \text{ pour } n \text{ t.q. } B_n\neq 0\end{cases}$$

où la notation $B_{n+1}\equiv A_n[B_n]$ siginifie que B_{n+1} est égal au reste de la division euclidienne de A_n par B_n.

Montrer qu'il existe un entier N tel que $B_N=0$, et que l'on a :

$$\text{pgcd}(A_{n+1},B_{n+1})=\text{pgcd}(A_n,B_n) \quad \forall n<N$$

En déduire que $\text{pgcd}(A,B)=B_{N-1}$.

2.1 avec la division euclidienne de Maple

Maple possède bien évidemment une fonction pour effectuer la division euclidienne de deux polynômes. Pour savoir comment elle fontionne, allez voir l'aide sur les fonctions `quo` et `rem` et testez-les sur quelques exemples.

En utilisant la fonction `rem` de Maple, définissez une procédure `Euclide` prenant en agrument deux polynômes non nuls A et B (et éventuellement le nom de leur variable X), et retournant le pgcd normalisé de A et B.

2.2 en reprogrammant la division euclidienne

En réfléchissant à la façon dont vous faites la division euclidienne de deux polynômes "à la main", définissez une procédure `division` prenant en argument deux polynômes non nuls A et B et retournant l'unique couple (Q,R) tel que :

$$\begin{cases} A=BQ+R \\ degre(R)<degre(B) \text{ ou } R=0\end{cases}$$

Adaptez ensuite votre procédure `Euclide` en une nouvelle procédure `Euclide2` utilisant votre procédure `division`.

2.3 algorithme d'Euclide étendu

L'agorithme d'Euclide peut facilement être étendu de sorte à permettre le calcul de "témoins" de Bézout. En effet, le théorème de Bézout nous donne l'existence de deux polynômes U et V tels que :

$$A.U+B.V=\text{pgcd}(A,B)$$

Reprenons la relation de récurrence donnée plus haut :

$$\begin{cases} A_0=A \quad A_{n+1}=B_n \\ B_0=B \quad B_{n+1}\equiv A_n[B_n], \text{ pour } n \text{ t.q. } B_n\neq 0\end{cases}$$

On a en fait l'égalité $A_n=B_n.Q+R$ (division euclidienne de A_n par B_n) et on pose simplement $B_{n+1}=R$. On peut en particulier écrire $B_{n+1}=A_n-B_n.Q$, et en "remontant", on pourrait ainsi obtenir B_{N-1} sous la forme $A.U+B.V$.

Donnez les relations de récurrence permettant d'obtenir U et V. Écrivez une nouvelle procédure `Euclide3` prenant en argument deux polynômes non nuls A et B et retournant un triplet (P,U,V) tel que $P=\text{pgcd}(A,B)$ et $A.U+B.V=P$.