

TP8 : Décomposition d'un polynôme en produit de polynômes sans facteur multiple

N'oubliez pas d'exécuter (valider avec la touche Entrée) les commandes Maple (texte en rouge) avant de les utiliser.

– Dans $\mathbf{Q[x]}$

On veut écrire un programme qui prenne en entrée un polynôme P de $\mathbf{Q[x]}$ et rende une liste de polynômes Q_i tels que :

- $P = Q_1 Q_2^2 \dots Q_k^k$,
- chaque Q_i soit sans facteur multiple,
- si $i \neq j$ alors $\text{pgcd}(Q_i, Q_j) = 1$,

Remarque : Q_i n'est pas forcément irréductible, c'est le produit des facteurs irréductibles apparaissant avec une puissance i dans P .

On utilise l'algorithme exposé en cours :

Initialisation

On calcule

$$A_1 = \text{pgcd}(P, P') \text{ et } B_1 = \frac{P}{A_1}$$

(donc $A_1 = Q_2 Q_3^2 \dots Q_k^{(k-1)}$ et $B_1 = Q_1 Q_2 \dots Q_k$)

Boucle

On calcule

$$A_2 = \text{pgcd}(A_1, A_1') \text{ et } B_2 = \frac{A_1}{A_2}$$

(donc $A_2 = Q_3 Q_4^2 \dots Q_k^{(k-2)}$ et $B_2 = Q_2 Q_3 \dots Q_k$)

et $Q_1 = \frac{B_1}{B_2}$, on a extrait le premier des Q_i

et on recommence jusqu'à ...

- Trouvez un test d'arrêt.
- Faites des tests de votre fonction. Vous pouvez commencer par un polynôme test simple du genre :

```
[ > p:=mul((x-i)^i, i=1..5); p:=expand(p);
```

Fonctions utiles : **diff** pour calculer la dérivée, **gcd** pour calculer le pgcd, **quo** pour calculer

le quotient (a besoin du nom de la variable).

– Solution

```
> SansFactMult:=proc(p)
  local a,b,i,q,j;
  a[1]:=gcd(p,diff(p,x));b[1]:=quo(p,a[1],x);i:=1;
  while a[i]<>1 do
    a[i+1]:=gcd(a[i],diff(a[i],x));
    b[i+1]:=quo(a[i],a[i+1],x);
    q[i]:=quo(b[i],b[i+1],x);i:=i+1
  od;
  [seq(q[j],j=1..i-1)]
end;
```

Construisons maintenant un polynôme test simple :

```
> p:=mul((x-i)^i,i=1..5);
      p:=(x-1)(x-2)2(x-3)3(x-4)4(x-5)5
> p:=expand(p);
p:=432000000 x - 864000000 + 1348952000 x3 - 424015067 x6 + 845928448 x5
   - 1258456700 x4 - 981360000 x2 - 1822678 x10 + 10629552 x9 - 47283632 x8
   + 161614309 x7 + 234290 x11 + x15 - 55 x14 + 1400 x13 - 21868 x12
> SansFactMult(p);
      [x-1, x-2, x-3, x-4]
```

Il manque le facteur $x-5$: lorsqu'on s'arrête $a_i=1$ donc a_{i-1} n'a pas de facteur multiple, c'est Q_k et le dernier Q_i calculé est Q_{k-1} . De même $b_i=Q_k$. Il faut donc rajouter b_i à notre liste.

```
> SansFactMult:=proc(p)
  local a,b,i,q,j;
  a[1]:=gcd(p,diff(p,x));b[1]:=quo(p,a[1],x);i:=1;
  while a[i]<>1 do
    a[i+1]:=gcd(a[i],diff(a[i],x));
    b[i+1]:=quo(a[i],a[i+1],x);
    q[i]:=quo(b[i],b[i+1],x);i:=i+1
  od;
  [seq(q[j],j=1..i-1),b[i]]
end;
> SansFactMult(p);
      [x-1, x-2, x-3, x-4, x-5]
```

Ca marche.

Un test plus compliqué :

```
> p:=expand((x-2)*(x^4-1)^2*(x^2-9)^3*(x^2+7)^5);
p:=-12252303 x - 4667544 x3 - 19008080 x6 + 24467562 x5 - 48935124 x4
   + 9335088 x2 + 10014944 x10 - 12165265 x9 + 24330530 x8 + 9504040 x7
   - 5007472 x11 + 173008 x15 - 346016 x14 - 63156 x13 + 126312 x12 + 24504606 x25
   + 8 x23 - 214 x21 - 2040 x19 + 13375 x17 - 2 x24 - 16 x22 + 428 x20 + 4080 x18
```

```

- 26750 x16
> SansFactMult(p);
[x-2, x4-1, x2-9, 1, x2+7]

```

Encore un test :

```

> p:=mul((x^i-i)^i,i=1..5);
p := (x-1)(x2-2)2(x3-3)3(x4-4)4(x5-5)5
> p:=expand(p);
p := 86400000 x - 86400000 - 115200000 x6 - 64800000 x5 - 21600000 x4
+ 86400000 x2 + 81040000 x10 + 93200000 x9 + 82800000 x8 - 36000000 x7
+ 16960000 x11 - 3068000 x15 - 54440000 x14 - 86100000 x13 - 54940000 x12
+ 7147923 x25 - 1634625 x23 - 19830425 x21 + 2918000 x19 + 46335500 x17
+ 5897325 x24 - 12393300 x22 - 17407075 x20 + 30057500 x18 + 33880500 x16 - x54
- 1150291 x30 + 4181452 x26 + 539177 x27 - 1552975 x28 - 1969288 x29 - 185609 x31
+ 47467 x35 + 243613 x34 + 422354 x33 + 366447 x32 - 65852 x36 + 8607 x40
- 6456 x39 - 40594 x38 - 75338 x37 + 23 x50 - 1055 x45 + 117 x48 + 611 x43 - 452 x46
+ 10533 x41 - 1097 x44 + 5274 x42 + 80 x49 - 59 x47 + x55 - 3 x51 - 5 x52 - 4 x53
> SansFactMult(p);
[x-1, 1, x3-3, x2+2, x5-5, x2-2]

```

C'est normal car $(x^4-4)^4$ se décompose en $(x^2+2)^4(x^2-2)^4$, donc le facteur x^2-2 apparaît à la puissance 6.

Dans $\mathbb{Z}/p\mathbb{Z}[x]$

On se demande si on peut utiliser cet algorithme dans $\mathbb{Z}/p\mathbb{Z}[x]$.

Faites un test dans $\mathbb{Z}/3\mathbb{Z}[x]$ sur le polynôme suivant

```

> p :=
x19+2*x18+x17+x16+2*x15+x12+2*x11+x10+x8+2*x7+x6+x5+x3+x2+2;

```

en calculant à la main les polynômes A_i, B_i dans $\mathbb{Z}/3\mathbb{Z}[x]$.

Fonctions utiles : pour calculer modulo p vous devez utiliser les fonctions **Gcd** et **Quo**.

```

> p1:=diff(p,x) mod 3;
p1 := x18 + 2 x16 + x15 + x10 + x9 + 2 x7 + 2 x6 + 2 x4 + 2 x
> a1:=Gcd(p,p1) mod 3;b1:=Quo(p,a1,x) mod 3;
a1 := x14 + 2 x13 + 2 x12 + x11 + 2 x10 + 2 x9 + x5 + 2 x4 + 2 x3 + 2 x2 + x + 1
b1 := x5 + 2 x3 + 2 x2 + x + 2
> a11:=diff(a1,x) mod 3;
a11 := 2 x13 + 2 x12 + 2 x10 + 2 x9 + 2 x4 + 2 x3 + x + 1
> a2:=Gcd(a1,a11) mod 3;b2:=Quo(a1,a2,x) mod 3;

```

$$a2 := x^{12} + x^9 + x^3 + 2$$

$$b2 := x^2 + 2x + 2$$

```
> a21:=diff(a2,x) mod 3;
```

$$a21 := 0$$

- Comprenez-vous d'où vient le problème ?

La dérivée de A_2 est nulle alors que ce polynôme n'est pas une constante, ses monômes ont tous des puissances multiples de 3.

- Pour mieux comprendre ce qui se passe vous pouvez utiliser la fonction **Factor** pour décomposer le polynôme P et les polynômes A_i, B_i dans $\mathbb{Z}/3\mathbb{Z}[x]$.

```
> Factor(p) mod 3;
```

$$(x^2+x+2)^3 (x^2+2x+2)^2 (x+1)(x^2+1)^4$$

```
> Factor(a1) mod 3;Factor(b1) mod 3;
```

$$(x^2+x+2)^3 (x^2+2x+2)(x^2+1)^3$$

$$(x^2+2x+2)(x+1)(x^2+1)$$

On remarque que les facteurs de multiplicité 3 (ou multiple de 3) ne baissent pas de multiplicité dans A_1 et donc disparaissent de B_1 ,

```
> Factor(a2) mod 3;Factor(b2) mod 3;
```

$$(x^2+x+2)^3 (x^2+1)^3$$

$$x^2+2x+2$$

jusqu'au moment où on arrive à un polynôme A_i (ici A_2) qui ne contient plus que des facteurs de multiplicité 3 et donc sa dérivée est nulle.

- Ecrire une fonction moins exigeante qui décompose le polynôme P en produit $Q_1 Q_2 \dots Q_k$ de polynômes sans facteur multiple.

On remarque que le polynôme B_1 est sans facteur multiple donc on le garde et on continue à décomposer le polynôme A_1 jusqu'à avoir un A_i dont la dérivée est nulle :

soit c'est une constante et on a fini,

soit $A_i = Q(x^p)$ donc $A_i = Q(x)^p$ et on décompose $Q(x)$.

```
> SansFactMultMod:=proc(P,p)
```

```
local P1,A1,B1;
```

```
P1:=diff(P,x) mod p;
```

```
if P1<>0 then A1:=Gcd(P,P1) mod p;B1:=Quo(P,A1,x) mod p;SansFactMultMod(A1,p)*B1
```

```
else if degree(P)<>0 then
```

```
SansFactMultMod(subs(x=x^(1/p),P),p)^p else P fi fi
```

```
end:
```

```
> SansFactMultMod(p,3);
```

$$(x^4+x^3+x+2)^3 (x^2+2x+2)(x^5+2x^3+2x^2+x+2)$$

```
> Factor(x^4+x^3+x+2) mod 3;Factor(x^2+2*x+2) mod 3;Factor(x^5+2*x^3+2*x^2+x+2) mod 3;
```

$$(x^2 + 1)(x^2 + x + 2)$$

$$x^2 + 2x + 2$$

$$(x^2 + 2x + 2)(x + 1)(x^2 + 1)$$

On a bien une décomposition en produit de polynômes sans facteur multiple.

Cette décomposition est moins élégante que celle obtenue sur $\mathbf{Q}[x]$: le facteur $x^2 + 1$ de multiplicité 4 se trouve dans le facteur $x^4 + x^3 + x + 2$ à la multiplicité 3 et dans le facteur $x^5 + 2x^3 + 2x^2 + x + 2$ à la multiplicité 1. Les polynômes Q_i ne sont plus premiers entre eux.