



ACTIVITE MANIPULATION DE POLYNOMES

Documentation du code implémenté



20 DECEMBRE 2019

BATISSE DYLANN
JUNIN THIBAUT

Table des matières

is_poly0	2
degre.....	2
poly2str.....	2
mult_ext	3
make_poly	3
rand_poly	4
sort_monoms	5
merge_monoms	5
add.....	6
sub	7
deriv.....	8
integ.....	8
mult_monoms	9
mult_poly_mono	9
mult	10
polyval	10
creux_to_dense.....	11
polyhorn	11
fpoly.....	12
dessiner	12
demonstration.....	13

is_poly0

Description

Indique si le polynôme est nul.

Usage

```
is_poly0(p)
```

Arguments

p Un polynôme écrit sous forme d'une liste

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
is_poly0(p)
## [1] FALSE
```

degre

Description

Renvoie le degré d'un monôme ou d'un polynôme.

Usage

```
degre(p)
```

Argument

p Un polynôme écrit sous forme d'une liste

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
degre(p)
## [1] 5
```

poly2str

Description

Renvoie une chaîne de caractères représentant le polynôme creux sous la forme $\sum a_i X^i$.

Usage

```
poly2str(p)
```

Argument

p Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
poly2str(p)
## [1] "2*X^5 + -1*X^4 + -2*X^1"
```

mult_ext

Description

Multiplication du polynôme p par un scalaire k.

Usage

```
mult_ext(p, k)
```

Arguments

p Un polynôme écrit sous forme d'une liste.
k Multiplie les coefficients du polynôme p par k.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))  
mult_ext(p, -2)  
poly2str(mult_ext(p, -2))  
## [[1]]  
## [1] -4 5  
## [[2]]  
## [1] 2 4  
## [[3]]  
## [1] 4 1  
## [1] "-4*X^5 + 2*X^4 + 4*X^1"
```

make_poly

Description

Transforme un polynôme plein en un polynôme creux.

Usage

```
make_poly(x)
```

Argument

x Un polynôme plein.

Exemples

```
x <- c(0, -2, 0, 0, -1, 2)  
make_poly(x)  
poly2str(make_poly(x))  
## [[1]]  
## [1] 2 5  
## [[2]]  
## [1] -1 4  
## [[3]]  
## [1] -2 1  
## [1] "2*X^5 + -1*X^4 + -2*X^1"
```

rand_poly

Description

Génère un polynôme aléatoire de degré inférieur à n et dont les coefficients sont tirés dans le vecteur `coeffs`.

Usage

```
rand_poly(n, coeffs)
```

Arguments

`n` Degré maximum.

`coeffs` Vecteur où les coefficients vont être tirés aléatoirement.

Exemples

```
n <- 5
coeffs <- 0:3
rand_poly(n, coeffs)
poly2str(rand_poly(n, coeffs))
## [[1]]
## [1] 1 3
## [[2]]
## [1] 3 2
## [[3]]
## [1] 3 1
## [1] "1*X^3 + 3*X^2 + 3*X^1"
```

sort_monoms

Description

Trie une liste de monômes par degré décroissant, fonction auxiliaire pour add(p, q).

Usage

```
sort_monoms(p)
```

Argument

p Un polynôme écrit sous forme d'une liste.

Exemples

```
x <- c(0, -2, 0, 0, -1, 2)
rand_poly(5, 0:3)
poly2str(rand_poly(5, 0:3))
## [[1]]
## [1] -1 4
## [[2]]
## [1] 2 3
## [[3]]
## [1] -2 1
## [1] "-1*X^4 + 2*X^3 + -2*X^1"
```

merge_monoms

Description

Somme les termes de même degré, fonction auxiliaire pour add(p, q).

Usage

```
merge_monoms(p)
```

Argument

p Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,4), c(-1,4),c(-2,1))
merge_monoms(p)
poly2str(merge_monoms (p))
## [[1]]
## [1] 1 4
## [[2]]
## [1] -2 1
## [1] "1*X^4 + -2*X^1"
```

add

Description

Addition de deux polynômes.

Usage

`add(p, q)`

Arguments

p Un polynôme écrit sous forme d'une liste.
q Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
q <- list(c(1,4), c(7,2),c(-1,0))
add(p, q)
poly2str(add (p, q))
## [[1]]
## [1] 2 5
## [[2]]
## [1] 7 2
## [[3]]
## [1] -2 1
## [[4]]
## [1] -1 0
## [1] "2*X^5 + 7*X^2 + -2*X^1 + -1*X^0"
```

sub

Description

Soustraction de deux polynômes.

Usage

sub(p, q)

Arguments

p Un polynôme écrit sous forme d'une liste.
q Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
q <- list(c(1,4), c(7,2),c(-1,0))
sub(p, q)
poly2str(sub(p, q))
## [[1]]
## [1] 2 5
## [[2]]
## [1] -2 4
## [[3]]
## [1] -7 2
## [[4]]
## [1] -2 1
## [[5]]
## [1] 1 0
## [1] "2*X^5 + -2*X^4 + -7*X^2 + -2*X^1 + 1*X^0"
```


deriv

Description

Retourne le polynôme dérivé du polynôme p.

Usage

deriv(p)

Argument

p Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
deriv(p)
poly2str(deriv(p))
## [[1]]
## [1] 10 4
## [[2]]
## [1] -4 3
## [[3]]
## [1] -2 0
## [1] "10*X^4 + -4*X^3 + -2*X^0"
```

integ

Description

Retourne une primitive du polynôme p.

Usage

integ(p)

Argument

p Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
integ(p)
poly2str(integ(p))
## [[1]]
## [1] 0.3333333 6.0000000
## [[2]]
## [1] -0.2 5.0
## [[3]]
## [1] -1 2
## [1] "0.3333333333333333*X^6 + -0.2*X^5 + -1*X^2"
```

mult_monoms

Description

Multiplie deux monômes, fonction auxiliaire pour mult(p, q).

Usage

```
mult_monoms(m1, m2)
```

Arguments

m1 Monôme 1.
m2 Monôme 2.

Exemples

```
m1 <- c(2,4)
m2 <- c(3, 4)
mult_monoms(m1, m2)
poly2str(mult_monoms(m1, m2))
## [[1]]
## [1] 6 8
## [1] "6*X^8"
```

mult_poly_mono

Description

Multiplie un polynôme par un monôme, fonction auxiliaire pour mult(p, q).

Usage

```
mult_poly_mono(p, m)
```

Arguments

p Un polynôme écrit sous forme d'une liste.
m Monôme par lequel est multiplié p.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
m <- c(2, 4)
mult_poly_mono(p, m)
poly2str(mult_poly_mono(p, m))
## [[1]]
## [1] 4 9
## [[2]]
## [1] -2 8
## [[3]]
## [1] -4 5
## [1] "4*X^9 + -2*X^8 + -4*X^5"
```

mult

Description

Multiplie deux polynômes.

Usage

```
mult(p,q)
```

Arguments

p Un polynôme écrit sous forme d'une liste.
q Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))  
q <- list(c(1,4), c(7,2),c(-1,0))  
mult(p,q)  
poly2str(mult(p,q))  
## [[1]]  
## [1] 1 2  
## [[2]]  
## [1] -1 0  
## [1] "1*X^2 + -1*X^0"
```

polyval

Description

Evaluation d'un polynôme p en un point x par la méthode naïve ($\sum a_i X^i$).

Usage

```
polyval(p, x)
```

Arguments

p Un polynôme écrit sous forme d'une liste.
x Entier qui évalue p.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))  
x <- -1  
polyval(p, x)  
## [1] -1
```

creux_to_dense

Description

Convertit un polynôme creux en un polynôme dense, fonction auxiliaire pour polyhorn(p, x).

Usage

```
creux_to_dense(p)
```

Argument

p Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
creux_to_dense(p)
## [1] 0 -2 0 0 -1 2
```

polyhorn

Description

Evaluation d'un polynôme p en un point x par la méthode de Horner ($\sum a_i X^i$).

Usage

```
polyhorn(p, x)
```

Arguments

p Un polynôme écrit sous forme d'une liste.
x Entier qui évalue p.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))
x <- -1
polyhorn(p, x)
## [1] -1
```

fpoly

Description

Construction de la fonction polynôme associée à p. Evaluation d'un polynôme p en un point x par la méthode naïve ($\sum a_i X^i$).

Usage

```
fpoly(p)
```

Argument

p Un polynôme écrit sous forme d'une liste.

Exemples

```
p <- list(c(2,5), c(-1,4),c(-2,1))  
f <- fpoly(p)  
f(-1)  
## [1] -1
```

dessiner

Description

Dessine la courbe du polynôme p (en bleu), ainsi que celle de sa dérivée (en rouge) et de sa primitive (en vert), sur l'intervalle passé en argument.

Usage

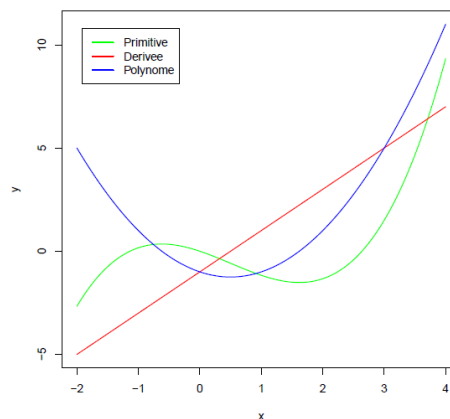
```
dessiner(p, x)
```

Arguments

p Un polynôme écrit sous forme d'une liste.
x Entier qui évalue p.

Exemples

```
p <- make_poly(c(-1,-1,1))  
x <- seq(-2,4,length.out=1000)  
dessiner(p, x)
```



demonstration

Description

Fait des cas de tests pour tous les exercices de l'activité.

Usage

demonstration()

Exemples

```
---TEST CASES DE L'ACTIVITE POLYNOME---
TEST CASE : poly2str(p) == "2*X^5 + -1*X^4 + -2*X^1" TRUE
TEST CASE : poly2str(mult_ext(p,-2)) == "-4*X^5 + 2*X^4 + 4*X^1" TRUE
TEST CASE : poly2str(make_poly(c(0, -2, 0, 0, -1, 2))) == "2*X^5 + -1*X^4 + -2*X^1" TRUE
TEST CASE : poly2str(make_poly(c(-1, 0, -7, 0, 1))) == "1*X^4 + -7*X^2 + -1*X^0" TRUE
TEST CASE : poly2str(rand_poly(5,0:3)) == "3*X^4 + 3*X^3 + 2*X^2 + 3*X^1"
TEST CASE : poly2str(rand_poly(10,0:1)) == "1*X^7 + 1*X^6 + 1*X^5 + 1*X^1"
TEST CASE : poly2str(add(p, list())) == "2*X^5 + -1*X^4 + -2*X^1" TRUE
TEST CASE : poly2str(add(list(), q)) == "1*X^4 + 7*X^2 + -1*X^0" TRUE
TEST CASE : poly2str(add(p,q)) == "2*X^5 + 7*X^2 + -2*X^1 + -1*X^0" TRUE
TEST CASE : poly2str(sub(p,p)) == "" TRUE
TEST CASE : poly2str(sub(p,q)) == "2*X^5 + -2*X^4 + -7*X^2 + -2*X^1 + 1*X^0" TRUE
TEST CASE : poly2str(sub(q,p)) == "-2*X^5 + 2*X^4 + 7*X^2 + 2*X^1 + -1*X^0" TRUE
TEST CASE : poly2str(integ(p)) == "0.3333333333333333*X^6 + -0.2*X^5 + -1*X^2" TRUE
TEST CASE : poly2str(integ(q)) == "0.2*X^5 + 2.3333333333333333*X^3 + -1*X^1" TRUE
TEST CASE : poly2str(deriv(p)) == "10*X^4 + -4*X^3 + -2*X^0" TRUE
TEST CASE : poly2str(deriv(q)) == "4*X^3 + 14*X^1" TRUE
TEST CASE : poly2str(mult(p1,p2)) == "1*X^2 + -1*X^0" TRUE
TEST CASE : poly2str(mult(p3,p3)) == "1*X^4 + -2*X^3 + 3*X^2 + -2*X^1 + 1*X^0" TRUE
---TEST POLYNOMES P EN UN POINT X---
UTILISATION DE POLYVAL
[1] "2*X^5 + -1*X^4 + -2*X^1"
[1] "p(-1) = -1"
[1] "p(0) = 0"
[1] "p(1) = -1"
[1] "1*X^4 + 7*X^2 + -1*X^0"
[1] "q(-1) = 7"
[1] "q(0) = -1"
[1] "q(1) = 7"
UTILISATION DE POLYHORN
[1] "2*X^5 + -1*X^4 + -2*X^1"
[1] "p(-1) = -1"
[1] "p(0) = 0"
[1] "p(1) = -1"
[1] "1*X^4 + 7*X^2 + -1*X^0"
[1] "q(-1) = 7"
[1] "q(0) = -1"
[1] "q(1) = 7"
dessiner(p, x) -> (voir plot)
```