CONCOURS CENTRALE•SUPÉLEC

Oral

Python

MP, PC, PSI, TSI

Polynômes

La classe `Polynomial` du module `numpy.polynomial.polynomial` permet de travailler avec des polynômes.

```
from numpy.polynomial import Polynomial
```

Pur créer un polynôme, il faut lister ses coefficients par ordre de degré croissant. Par exemple, pour le polynôme $X^3 + 2X - 3$,

```
p = Polynomial([-3, 2, 0, 1])
```

On peut alors utiliser cette variable comme une fonction pour calculer, en un point quelconque, la valeur de la fonction polynôme associée. Cette fonction peut agir également sur un tableau de valeurs, elle calcule alors la valeur de la fonction polynôme en chacun des points indiqués.

```
p(0)
-3.0
p([1, 2, 3])
array([  0.,   9.,  30.])
```

L'attribut `coef` donne accès aux coefficients ordonnés par degré croissant ; ainsi `p.coef[i]` correspond au coefficient du terme de degré `i`. La méthode `degree` renvoie le degré du polynôme alors que `roots` calcule ses racines.

```
p.coef
array([-3.,  2.,  0.,  1.])
p.coef[1]
2.0
p.degree()
3
p.roots()
array([-0.5-1.6583124j, -0.5+1.6583124j,  1.0+0.j       ])
```

La méthode `deriv` renvoie un nouveau polynôme, dérivé du polynôme initial. Cette méthode prend en argument facultatif un entier positif indiquant le nombre de dérivations à effectuer. De la même manière la méthode `integ` intègre le polynôme, elle prend un paramètre optionnel supplémentaire donnant la constante d'intégration à utiliser, ce paramètres peut être une liste en cas d'intégration multiple ; les constantes d'intégration non précisées sont prises égales à zéro.

```
p.deriv().coef
array([ 2.,  0.,  3.])
p.deriv(2).coef
array([ 0.,  6.])
p.deriv(5).coef
array([-0.])
p.integ().coef
array([ 0.  , -3.  ,  1.  ,  0.  ,  0.25])
p.integ(1, 2).coef  # intégrer une fois avec la constante 2
array([ 2.  , -3.  ,  1.  ,  0.  ,  0.25])
p.integ(2, [1, 2]).coef  # intégrer deux fois
array([ 2.        ,  1.        , -1.5       ,  0.33333333,  0.        ,
        0.05      ])
```

Les opérateurs `+`, `-`, `*` permettent d'additionner, soustraire et multiplier des polynômes. Ils fonctionnent également entre un polynôme et un scalaire. L'opérateur `**` permet d'élever un polynôme à une puissance entière positive.

```
a = Polynomial([1, 2, 1])
b = Polynomial([5, 3])
p = 2*a * b + Polynomial([-7, 2])
p.coef
array([  3.,  28.,  22.,   6.])
(p**2).coef
array([   9.,  168.,  916., 1268.,  820.,  264.,   36.])
```

L'opérateur `/` permet de diviser un polynôme par un scalaire. Pour diviser deux polynômes il faut utiliser l'opérateur `//` qui renvoie le quotient ; l'opérateur `%` calcule le reste.

```
(p / 2).coef
array([  1.5,  14. ,  11. ,   3. ])
q = p // a
r = p % a
q.coef
array([ 10.,   6.])
r.coef
array([-7.,  2.])
(q * a + r).coef
array([  3.,  28.,  22.,   6.])
```