

# Polynômes : boucle for, fonction mathématique

```
<sxh python; title : poly05-math-sinus.py> #!/usr/bin/python # -*- coding: UTF-8 -*- """ écriture d'un programme pour évaluer des polynomes """ from math import *
```

```
def polyeval(x,a):
```

```
    """ Fonction s'occupant uniquement de l'évaluation du polynome fonction de
    x
    avec les coefficients dans la liste a"""
    n = len(a)-1
    p = 0. # initialisation
    for i in range(n+1):
        p = p + a[i]*x**i #calcul et addition de chacun des termes
    return p
```

```
# utilisation dans une structure de répétition (boucle for) # en combinaison avec une autre fonction
mathématique (sinus) varcoef = [1.,2.,3.,4.,5.,6.,7.,8.,9.,10.] for j in range(0,11,1):
```

```
    vax = float(j) * 0.1
    rep = sin(polyeval(vax,varcoef))
    print vax,rep
```

```
</sxh>
```

S'il y a beaucoup d'abscisses pour lesquelles il faut évaluer le polynôme, ces calculs sont nombreux !

**Mais au fait, combien d'opérations élémentaires (addition, multiplication) faut-il effectuer pour chaque évaluation d'un polynôme de degré  $n$**

Cela augmente "beaucoup plus vite" que  $n$ . En fait le nombre de multiplications de la fonction `polyeval` (les opérations les plus fréquentes et les plus lentes à réaliser) se comporte comme  $n^2$  !

Il y a sûrement moyen de faire mieux, d'économiser des opérations. Essayez sur un exemple, et trouvez une méthode systématique. On doit pouvoir arriver à un nombre de multiplications proportionnel à  $n$  !

[Solution à la page suivante !](#)

From: <https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link: <https://dvillers.umons.ac.be/wiki/teaching:progappchim:polynomes-5?rev=1456217772>

Last update: **2016/02/23 09:56**

