Polynômes : graphes de fonctions polynomiales

<sxh python; title : poly08-plot.py> #!/usr/bin/python # -*- coding: UTF-8 -*- “”“ écriture d'un programme pour évaluer des polynomes ”“” from math import * from pylab import * # librairies de graphiques (matplotlib)

def polyeval(x,a):

```
    """application de l'agorithme de Horner
    cf. http://fr.wikipedia.org/wiki/M%C3%A9thode_de_Ruffini-Horner
    """
    n=len(a)-1 # n = ordre du polynome
    p=a[n]
    for i in range(n-1,-1,-1):
        p=p*x+a[i]
    return p
```

def polyscal(a,s):

```
    """polynome multiplié par un scalaire s """
    b=[]
    for coef in a:
        b.append(coef*s)
    return b    # on retourne les coefficients multipliés par s
```

def polyadd(a,b):

```
    """ Addition de deux polynomes de coefficients a et b
    """
    r=a[:] # on travaille sur une copie de a pour ne pas le modifier
    t=b[:] # idem pour b
    g=[]   # polynome somme
    n1=len(r) # ordre du premier polynome
    n2=len(t) # ordre du second polynome
    if n1>n2: # premier polynome de plus haut degré que le second
        for i in range (n1-n2):
            t.append(0)
    elif n1<n2: # second polynome de plus haut degré que le premier
         for i in range (n2-n1):
             r.append(0)
    # r et t ont à présent la même longueur
    for i in range (len(r)):
        g.append(r[i]+t[i])
    return g  # on retourne les coefficients additionnés dans la liste g
```

différents tests :

absc=[] ordo=[] coef=[0,9,0,-120,0,432,0,-576,0,256] # un polynome un peu particulier xa=-1.04 while xa < 1.04:

```
  absc.append(xa)
  ordo.append(polyeval(xa,coef))
  xa=xa+0.01
```

plot(absc,ordo) show() </sxh>

Il s'agit à présent de créer d'autres fonctionnalités sur les polynômes !

Des propositions 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-8?rev=1352459575

Last update: **2012/11/09 12:12**