Pregunta 3

Si p(x)=2x^3+x^2-2x+1 entonces

c=[1, -2, 1, 2]

p(2) se calcula como eval_poli(c,2)

Si p'(x)=6x^2+2x-2 entonces

p'(2) se calcula como eval_deriv_poli(c,2)

In [48]:
```
import numpy as np
def eval_poli(c,x):
    n = len(c)
    t = 0
    for k in range(n):
        t = c[n-k-1]+t*x
    return t

def eval_deriv_poli(c,x):
    n = len(c)
    t = 0
    for k in range(n-1):
        t = (n-k-1)*c[n-k-1]+t*x
    return t

def division(c,a):
# divide q(x)=p(x)(x-a)+r(x)
    n = len(c)
# n>2
    t = c[n-1]
    d = np.zeros_like(c[:-1])
    for k in range(n-1):
        s = c[n-k-2]
        d[n-k-2] = t
        t = s + t*a
    return d
def newton(c,x0,tol=1E-5):
    h=1
    while np.abs(h)>tol:
        h = -eval_poli(c,x0)/eval_deriv_poli(c,x0)
        x0 = x0 + h
    return x0
```

Cada vez que calculamos una raiz r con el metodo de newton

dividimos p(x) entre x-r y continuamos con el proceso

In [62]:
```
c = np.array([24+0j,26+0j,9+0j,1+0j])
a = np.zeros_like(c[:-1])
n = len(a)
for k in range(n):
    a[k]=newton(c,x0=1+1j)
    print(f"a{k+1}={-a[k]}")
    c = division(c,a[k])
```

```
a1=(1.999999999995419+7.150127393010309e-12j)
a2=(3.000000000007639-1.4003574270789062e-11j)
a3=(3.9999999999969424+6.853446877778753e-12j)
```

In [63]:
```
c = np.array([90+0j,153+0j,77+0j,5+0j,1+0j])
b = np.zeros_like(c[:-1])
n = len(b)
for k in range(n):
    b[k]=newton(c,x0=1+1j)
    print(f"b{k+1}={-b[k]}")
    c = division(c,b[k])
```

```
b1=(1.0728937989343776-0.3772748535222686j)
b2=(1.0728937989343776+0.3772748535222686j)
b3=(1.427106201061275-8.218600665433565j)
b4=(1.4271062010699693+8.218600665433565j)
```