In [1]:

```
from sympy import *
```

In [27]:

```
def halleys_method(function, initial, iterations, precision):
    x = Symbol('x')
    iters = iterations

    lfunction = lambdify(x, function, 'numpy')

    dfunction = function.diff(x)
    ldfunction = lambdify(x, dfunction, 'numpy')

    d2function = dfunction.diff(x)
    ld2function = lambdify(x, d2function, 'numpy')

    while (lfunction(initial) > precision) and (iters > 0):
        t = - lfunction(initial)/ldfunction(initial)
        r = ld2function(initial)*t**2/ldfunction(initial)
        initial = initial + t**2/(t + 0.5*r)
        iters -= 1

    return {'root': initial, 'iterations': iterations - iters}
```

In [39]:

```
x = Symbol('x')
func = x**3 + 6*x**2 + 9*x - 4
halleys_method(func, 3.6, 10, 0.001)
```

Out[39]:

```
{'root': 0.3553270038206912, 'iterations': 3}
```

In [42]:

```
# невязка
root = halleys_method(func, 3.6, 10, 0.001)['root']
residual = lambdify(x, func, 'numpy')(root)
residual
```

Out[42]:

```
0.00034933313787899323
```