Métodos Númericos
Trabalho 1 - Algoritmos

Carlos Eduardo Cassimiro da SIlva

```
f = função que desejamos a raiz
itr = quantidade de iterações
a, b = intervalo onde está a raiz

Bissecao(f, itr, a, b)
{
    xbsc <- (a+b)/2
    Se itr > 1 faça:
        Se f(a)*f(xbsc) faça:
            xbsc <- Bissecao(f, itr-1, a, xbsc)
        Se f(a)*f(xbsc) > 0:
            xbsc <- Bissecao(f, itr-1, xbsc, b)
    retorne xbsc
}


Falsa_posicao(f, itr, a, b)
{
    xbsc <- (a*f(b)-b*f(a))/(f(b)-f(a))
    Se itr > 1 faça:
        Se f(a)*f(xbsc) faça:
            xbsc <- Falsa_posicao(f, itr-1, a, xbsc)
        Se f(a)*f(xbsc) > 0:
            xbsc <- Falsa_posicao(f, itr-1, xbsc, b)
    retorne xbsc
}


Metodo_secante(f, itr, a, b):
{
    Enquanto itr>0 e f(b)!=0.0 faça:
        c <- b - (f(b)*(a-b))/(f(a)-f(b))
        a <- b
        b <- c
        itr <- itr - 1
    Retorne a
}
```

```
x = ponto inicial na absissa
f`(x) = derivada de f(x)

Metodo_Newton(f, itr, x)
{
    Enquanto itr>0 e f(x)!=0.0 faça:
        x2 <- x - (f(x) / f`(x))
        x <- x2
        itr <- irt - 1
    Retorne x
}


g1(x) = função de iteração 1
g2(x) = função de iteração 2
g3(x) = função de iteração 3

Ponto_fixo(f, itr, x):
{
    cond <- True
    Se cond = True e (|g1`(x)|<1 e |g1`(x+1)|<1) faça:
        g(x) <- g1`(x)
        cond <- False
    Se cond = True e (|g2`(x)|<1 e |g2`(x+1)|<1) faça:
        g(x) <- g2`(x)
        cond <- False
    Se cond = True e (|g3`(x)|<1 e |g3`(x+1)|<1) faça:
        g(x) <- g3`(x)
        cond <- False
    Para i de  0 até itr faça:
        x <- g(x)
        Se f(x) = 0.0 faça:
            Sair
    Retorne x
}
```