

Devoir maison n°4 : Méthode de Newton

Jules Charlier, Thomas Diot, Pierre Gallois, Jim Garnier
TE1

Partie A -

Partie B -

Partie C - Algorithmes

1)

2)

3)

Tentons maintenant de simplifier et d'optimiser ce code :

```
f = lambda x: x**3 - 2

def newton(f, x, h=1e-4, epsilon=1e-6):
    while abs(y := f(x)) > epsilon:
        derivee = (f(x + h) - f(x - h)) / (2 * h)
        x -= (y / derivee)
    return x
```

Pour aller encore plus loin dans la simplification, changeons de langage pour Haskell :

```
f :: (Num r) => r -> r
f x = x^3 - 2

derivee f x h = (f (x + h) - f (x - h)) / 2*h

newton f h e x =
    if (abs . f) x > e
    then newton f h e (x - (f(x) / (derivee f x h)))
    else x
```

```
main = do
  let result = newton f 1e-4 1e-6 1
  (putStrLn . show) result
```

