

# Obliczenia naukowe

## Lista 3

Szymon Janiak

November 18, 2023

### Zadanie 1

#### Opis problemu

Napisać funkcję rozwiązującą równanie  $f(x) = 0$  metodą bisekcji.

#### Dane wejściowe

- **f** — funkcja  $f$  w postaci anonimowej funkcji
- **a, b** — liczby typu `Float64` określające końce przedziału początkowego
- **delta, epsilon** — liczby typu `Float64` określające dokładności obliczeń

Czwórka wartości (**r**, **val**, **it**, **err**).

- **r** — przybliżenie pierwiastka równania  $f(x) = 0$
- **v** — wartość funkcji w  $r$
- **it** — liczba wykonanych iteracji
- **err** — sygnalizacja błędu, możliwe wartości:
  - 0 — brak błędu
  - 1 — funkcja nie zmienia znaku w przedziale  $[a; b]$

#### Rozwiązanie

---

**Algorithm 1** bisection method

---

```
val ← 0
it ← 0
e ← b − a
u ← f(a)
v ← f(b)
r ←  $\frac{1}{2} * (a + b)$ 
if sign(u) = sign(v) then
    err ← 1
    return r, val, it, err
end if
while abs(e) >  $\epsilon$  and abs(f(r)) >  $\delta$  do
    e ← frace2
    r ← a + e
    val ← f(r)
    it ← it + 1
    if abs(e) <  $\delta$  or abs(val) <  $\epsilon$  then
        return r, val, it, err
    end if
    if sign(val) ≠ sign(u) then
        g ← r
        v ← val
    else
        a ← r
        u ← val
    end if
end while
return r, val, it, err
```

---

## Zadanie 2

### Opis problemu

Napisać funkcję rozwiązującą równanie  $f(x) = 0$  metodą Newtona.

### Dane wejściowe

- **f** — funkcja  $f$  w postaci anonimowej funkcji
- **pf** — pochodna funkcji  $f$  w postaci anonimowej funkcji
- **x0** — przybliżenie początkowe
- **delta**, **epsilon** — liczby typu `Float64` określające dokładności obliczeń
- **maxit** — liczba całkowita określająca dopuszczalną liczbę iteracji

### Dane wyjściowe

Czwórka wartości (**r**, **v**, **it**, **err**).

- **r** — przybliżenie pierwiastka równania  $f(x) = 0$
- **v** — wartość funkcji w  $r$
- **it** — liczba wykonanych iteracji
- **err** — sygnalizacja błędu, możliwe wartości:
  - 0 — metoda zbieżna
  - 1 — nie osiągnięto wymaganej dokładności w **maxit** iteracji
  - 2 — pochodna bliska zeru

### Rozwiązanie

---

**Algorithm 2** Newton method

---

```
val ← f(x0)
valprime ← 0
x1 ← 0
it ← 1
if abs(v) < ε then
    err ← 0
    return x0, val, it, err
end if
for it to maxit do
    valprime ← pf(x0)
    x1 = x0 - fracvalvalprime
    val ← f(x1)
    if abs(valprime) ≤ NEARZERO or isInf(abs(valprime)) then
        err ← 2
        return x0, f(x0), it, err
    end if
    if abs(x1 - x0) < δ or abs(val) < ε then
        return return x1, val, it, err
    end if
    x0 ← x1
end for
err ← 1
return x0, val, it, err
```

---