Numerical Method Python Implementations

August 22, 2024

0.1 Importing Libraries

[ ]:
```python
import numpy as np
from math import sqrt
```

0.2 Define Problem Function

$$f(R) = \exp\left[\frac{-0.05R}{10}\right]\cos\left[0.05\sqrt{\frac{10^4}{5}-\left(\frac{R}{10}\right)^2}\right]-0.01$$

[ ]:
```python
def func(R):
    return np.exp(-R*0.05/10)*np.cos(sqrt(10**4/5-(R/10)**2)*0.05) - 0.01
```

0.3 Define Bisection Method

[ ]:
```python
def bisection(f, a, b, tol):
    if np.sign(f(a)) == np.sign(f(b)):
        raise Exception("The scalars a and b do not bound a root")

    iterations = 1
    m = (a + b) / 2
    error = np.inf

    while error > tol:

        print(f'Iteration-{iterations},\txl = {a:.6f},\txu = {b:.6f},\tx_r = {m:
↪.6f},\tf(x_r) = {f(m):.6f},\tError = {error:.6f}')
        m_old = m
        if np.sign(f(m)) == np.sign(f(a)):
            a = m
        else:
            b = m
        m = (a + b) / 2
        error = abs(m - m_old) / abs(m)

        iterations += 1
    print(f'Iteration-{iterations},\txl = {a:.6f},\txu = {b:.6f},\tx_r = {m:.
↪6f},\tf(x_r) = {f(m):.6f},\tError = {error:.6f}')
```

```
    # Create a DataFrame from the iterations list
    return m
```

0.4 Define False Position Method

[ ]:
```
def false_position_custom(xl, xu, err, f):
    step = 1
    condition = True
    x_r = xu - (xl - xu) * f(xu) / (f(xl) - f(xu))  # Inisialisasi x_r pertama

    while condition:
        # Menyimpan nilai x_r lama
        x_r_old = x_r

        # Menghitung x_r baru
        x_r = xu - (xl - xu) * f(xu) / (f(xl) - f(xu))

        # Menghitung error berdasarkan perubahan x_r
        error = abs(x_r - x_r_old) / abs(x_r) if step > 1 else np.inf

        # Menampilkan hasil iterasi termasuk xl, xu, x_r, f(x_r), dan error
        print(f'Iteration-{step},\txl = {xl:.6f},\txu = {xu:.6f},\tx_r = {x_r:.
 ↪6f},\tf(x_r) = {f(x_r):.6f},\tError = {error:.6f}')

        # Memperbarui batas interval
        if f(xl) * f(x_r) < 0:
            xu = x_r
        else:
            xl = x_r

        # Memperbarui kondisi perulangan
        step += 1
        condition = error > err
    print(f'Iteration-{step},\txl = {xl:.6f},\txu = {xu:.6f},\tx_r = {x_r:.
 ↪6f},\tf(x_r) = {f(x_r):.6f},\tError = {error:.6f}')

    return x_r
```

0.5 Call Bisection Method

[ ]:
```
bisection(func, 0, 400, 0.01)
```

```
Iteration-1,    xl = 0.000000,  xu = 400.000000,        x_r = 200.000000,
f(x_r) = -0.163092,     Error = inf
Iteration-2,    xl = 200.000000,        xu = 400.000000,        x_r =
300.000000,       f(x_r) = -0.029503,     Error = 0.333333
Iteration-3,    xl = 300.000000,        xu = 400.000000,        x_r =
350.000000,       f(x_r) = 0.020915,      Error = 0.142857
Iteration-4,    xl = 300.000000,         xu = 350.000000,         x_r =
325.000000,       f(x_r) = -0.003155,     Error = 0.076923
Iteration-5,    xl = 325.000000,         xu = 350.000000,         x_r =
337.500000,       f(x_r) = 0.009150,      Error = 0.037037
Iteration-6,    xl = 325.000000,         xu = 337.500000,         x_r =
331.250000,       f(x_r) = 0.003067,      Error = 0.018868
Iteration-7,    xl = 325.000000,         xu = 331.250000,         x_r =
328.125000,       f(x_r) = -0.000026,     Error = 0.009524
```

[ ]:
```
328.125
```

0.6 Call False Position Method

[ ]:
```
false_position_custom(0, 400, 0.01, func)
```

```
Iteration-1,    xl = 0.000000,  xu = 400.000000,        x_r = 363.428484,
f(x_r) = 0.032985,      Error = inf
Iteration-2,    xl = 0.000000,  xu = 363.428484,        x_r = 345.272549,
f(x_r) = 0.016527,      Error = 0.052584
Iteration-3,    xl = 0.000000,  xu = 345.272549,        x_r = 336.409001,
f(x_r) = 0.008098,      Error = 0.026348
Iteration-4,    xl = 0.000000,  xu = 336.409001,        x_r = 332.121403,
f(x_r) = 0.003923,      Error = 0.012910
Iteration-5,    xl = 0.000000,  xu = 332.121403,        x_r = 330.057061,
f(x_r) = 0.001890,      Error = 0.006254
Iteration-6,    xl = 0.000000,  xu = 330.057061,        x_r = 330.057061,
f(x_r) = 0.001890,      Error = 0.006254
```

[ ]:
```
330.0570611015365
```