Numerical Analysis Homework (3)

Dimitri Tabatadze · Saturday 16-03-2024

Problem 3.1:

```
fn bisection_method(mut l: f64, mut r: f64, f: fn(f64)->f64, tol: f64) -> f64 {
    while (r - l) / 2.0 > tol {
        let m = (l + r) / 2.0;
        if f(m) == 0.0 {
            return m
        } else if f(l) * f(m) < 0.0 {
            r = m
        } else {
            l = m
        }
    }

    (l + r) / 2.0
}

fn main() {
    let a = |x: f64| 3.0 * x.powi(3) + x.powi(2) - x - 5.0;
    let b = |x: f64| x.cos().powi(2) - x + 6.0;

    let a = bisection_method(-100.0, 100.0, a, 1e-7);
    let b = bisection_method(-100.0, 100.0, b, 1e-7);

    println!("a: {a} \nb: {b}")
}
```

(a) 1.1697261594235897
(b) 6.776092294603586

Problem 3.2:

```
fn secants_method(l: f64, r: f64, f: fn(f64)->f64, tol: f64) -> f64 {
    let mut x = (l, r);

    while (x.0 - x.1).abs() / 2.0 > tol {
        x = (x.1, x.1 - f(x.1) * (x.1 - x.0) / (f(x.1) - f(x.0)));
    }

    return (x.0 + x.1) / 2.0;
}

fn main() {
    let a = |x: f64| 3.0 * x.powi(3) + x.powi(2) - x - 5.0;
    let b = |x: f64| x.cos().powi(2) - x + 6.0;

    let a = secants_method(1.0, 10.0, a, 1e-7);
    let b = secants_method(-10.0, 10.0, b, 1e-7);

    println!("a: {a} \nb: {b}")
}
```

(a) 1.1697262198431786
(b) 6.776092319299465

Problem 3.3:

```rust
fn bisection_method(mut l: f64, mut r: f64, f: fn(f64)->f64, tol: f64) -> f64 {
    while (r - l) / 2.0 > tol {
        let m = (l + r) / 2.0;
        if f(m) == 0.0 {
            return m
        } else if f(l) * f(m) < 0.0 {
            r = m
        } else {
            l = m
        }
    }

    (l + r) / 2.0
}

fn main() {
    let a = |x: f64| (x + 6.0) / (3.0 * x - 2.0);
    let b = |x: f64| x.powi(5);

    let a = bisection_method(-10.0, 10.0, a, 1e-7);
    let b = bisection_method(-1.0, 1.0, b, 1e-7);

    println!("a: {a} \nb: {b}")
}
```

(a) -6.000000014901161
(b) 0

Problem 3.4:

```
fn fixed_point(f: fn(f64)->f64, x_0: f64, tol: f64) -> f64 {
    let mut x = x_0;
    for _ in 0..1000 {
        let new_x = -f(x) / 100.0 + x;
        if (new_x - x).abs() < tol {
            return x
        }
        x = new_x;
    }
    let mut x = x_0;
    for _ in 0..1000 {
        let new_x = f(x) / 100.0 + x;
        if (new_x - x).abs() < tol {
            return x
        }
        x = new_x;
    }
    x_0
}

fn main() {
    let a = |x: f64| x.powf(5.0) + 1.0 * x - 1.0;
    let b = |x: f64| x.sin() - 6.0 * x - 5.0;
    let c = |x: f64| x.ln() + x.powf(2.0) - 3.0;

    println!("Fixed-point iteration method:");
    let a_solution = fixed_point(a, 0.754, 1e-10);
    println!("(a) x = {:.10}", a_solution);

    let b_solution = fixed_point(b, -0.97, 1e-10);
    println!("(b) x = {:.10}", b_solution);

    let c_solution = fixed_point(c, 1.5, 1e-10);
    println!("(c) x = {:.10}", c_solution);
}
```

(a) x = 0.7548776625
(b) x = −0.9708989217
(c) x = 1.5921429345

Problem 3.5:

```
fn newtons_method(f: fn(f64)->f64, der_f: fn(f64)->f64, mut x: f64, tol: f64) -> f64 {
    for _ in 0..20 {
        let new_x = x - f(x) / der_f(x);
        if (new_x - x).abs() < tol {
            x = new_x;
            break
        }
        x = new_x;
    }
    x
}

fn main() {
    let a = |x: f64| x.powf(5.0) + 1.0 * x - 1.0;
    let der_a = |x: f64| 5.0 * x.powf(4.0) + 1.0;
    let b = |x: f64| x.sin() - 6.0 * x - 5.0;
    let der_b = |x: f64| x.cos() - 6.0;
    let c = |x: f64| x.ln() + x.powf(2.0) - 3.0;
    let der_c = |x: f64| 1.0 / x + 2.0 * x;

    let a_solution = newtons_method(a, der_a, 0.754, 1e-8);
    println!("(a) x = {:.10}", a_solution);

    let b_solution = newtons_method(b, der_b, -0.97, 1e-8);
    println!("(b) x = {:.10}", b_solution);

    let c_solution = newtons_method(c, der_c, 1.5, 1e-8);
    println!("(c) x = {:.10}", c_solution);
}
```

(a) x = 0.7548776662
(b) x = −0.9708989235
(c) x = 1.5921429371

Problem 3.6:

Proof: We can use algebra to find the real and the only root of $f(x) = ax + b$ which is $-\frac{b}{a}$. Now, if we start with an initial guess x_0 and apply one iteration of newtons method, we will get

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = x_0 - \frac{ax_0 + b}{a} = x_0 - x_0 - \frac{b}{a} = -\frac{b}{a}$$

which we know is the root of $f(x)$. □

Problem 3.7:

$$\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} = 1 - \frac{1}{-1/1^2} = 2 \\
x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} = 2 - \frac{1/2}{-1/2^2} = 4 \\
x_3 &= x_2 - \frac{f(x_2)}{f'(x_2)} = 4 - \frac{1/4}{-1/4^2} = 8 \\
&\vdots \\
x_{50} &= x_{49} - \frac{f(x_{49})}{f'(x_{49})} = 2^{49} - \frac{2^{-49}}{-2^{-98}} = 2^{50}
\end{aligned}$$