Maaz Saeed - 2021268 - A02

Question # 01

```
bisection (generic function with 1 method)
```

```
function bisection(f, a, b; iters=1000, TOL=1e-3)
    a₁, b₁ = a, b
    if sign(f(a₁)) == sign(f(b₁))
        error("Does not satisfy the IVT theorem!")
    end

    for _=1:iters
        p₁ = 0.5 * (a₁ + b₁) # Midpoint
        fₚ₁ = f(p₁)

        if(fₚ₁) == 0 || (b₁-a₁)/2 < TOL
            return p₁
        end

        if f(a₁)*f(p₁) > 0
            a₁ = p₁
        else
            b₁ = p₁
        end
        print(a₁)
        print("\t\t\t\t")
        print(b₁)
        println()
    end

    return 0.5 * (a₁ + b₁)
end
```

Root of $f(x) = x^3 + 4x^2 - 10$ is 1.36517333984375:

```
let
    root = bisection(x -> x^3 + 4x^2 - 10, 1, 2, TOL=1e-4)
    md"Root of $f(x) = x^3 + 4x^2 - 10$ is $root:

    "
end
```

```
1               1.5
1.25                1.5
1.25                1.375
1.3125              1.375
1.34375             1.375
1.359375                1.375
1.359375                1.3671875
1.36328125              1.3671875
1.36328125              1.365234375
1.3642578125                1.365234375
1.36474609375               1.365234375
1.364990234375              1.365234375
1.3651123046875             1.365234375
```

fixed_point (generic function with 1 method)

```
function fixed_point(g, x₀; iters=1000, TOL=1e-6)
    x = x₀
    for _=1:iters
        xᵢ = g(x)
        println(xᵢ)
        if abs(x - xᵢ) < TOL
            return xᵢ
        end
        x = xᵢ
    end

    x
end
```

Root of $f(x) = \frac{1}{x+1}$ is 0.6180338134001252

```
let
    root = fixed_point(x -> 1 / (x + 1), 0)
    md"Root of $f(x) = \frac{1}{x + 1}$ is $root

    "
end
```

```
1.0
0.5
0.6666666666666666
0.6000000000000001
0.625
0.6153846153846154
0.6190476190476191
0.6176470588235294
0.6181818181818182
0.6179775280898876
0.6180555555555556
0.6180257510729613
0.6180371352785146
0.6180327868852459
0.6180344478216819
0.6180338134001252
```

```
secant (generic function with 1 method)
```

```
function secant(f, x₁, x₂; iters=1000, TOL=1e-6)

    for _=1:iters
        numerator = f(x₂)
        denominator = (f(x₂) - f(x₁)) / (x₂ - x₁)

        xₙ = x₂ - numerator/denominator
        if abs(xₙ - x₂) < TOL
            return xₙ
        end
        x₁ = x₂
        x₂ = xₙ

        println(x₁, "\t\t", x₂)
    end

    x₂
end
```

```
2.682695795244651
```

```
secant(x -> x^7 - 1000, 2, 3, iters=100)
```

```
3		2.4235065565808647
2.4235065565808647		2.596515067663605
2.596515067663605		2.7125518586770143
2.7125518586770143		2.6797406845453278
2.6797406845453278		2.6825984482569294
2.6825984482569294		2.682696117464538
```

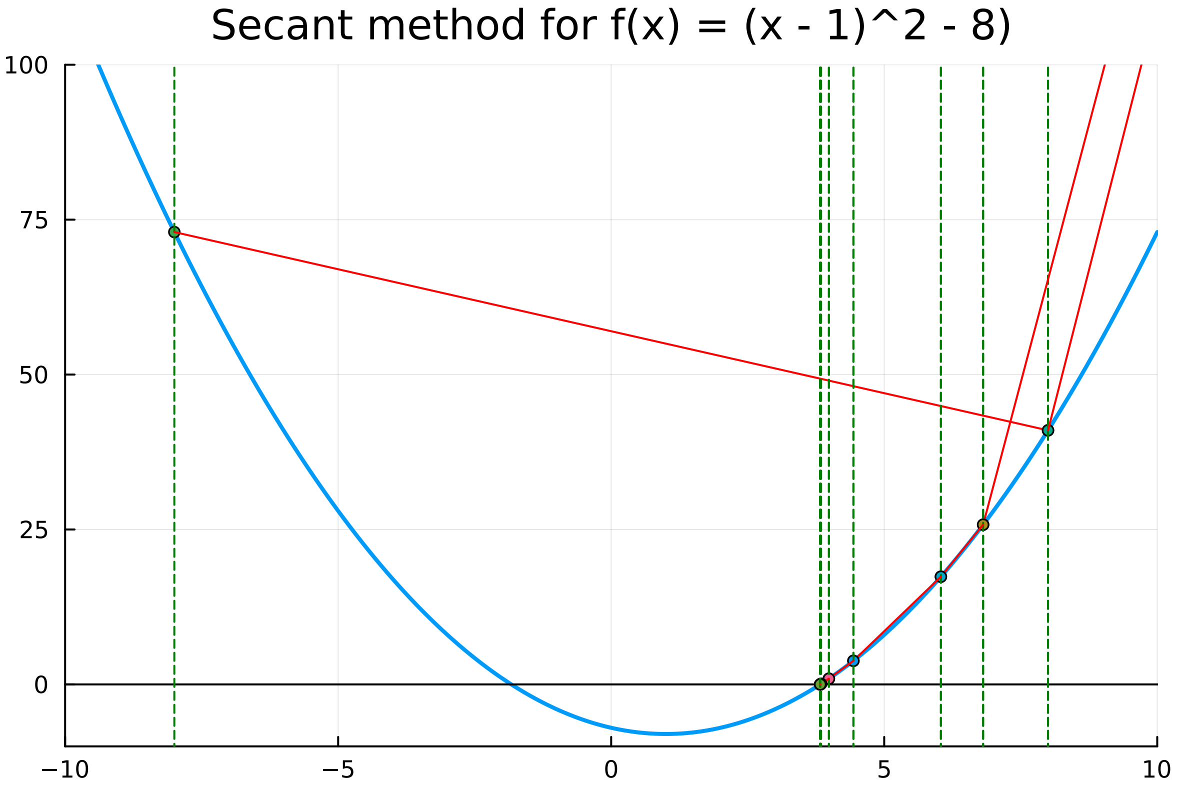
```
using Plots
```

```
next_secant (generic function with 1 method)
```

```julia
function next_secant(f, x₁, x₂)
    numerator = f(x₂)
    denominator = (f(x₂) - f(x₁)) / (x₂ - x₁)
    return x₂, (x₂ - numerator/denominator)
end
```

11

```julia
@bind t Slider(1:20, show_value=true)
```




```julia
begin
    p = plot()
    f(x) = (x - 1)^2 - 8

    to_radians(θ) = θ * (π / 180)

    x_vals = -10:0.01:10
    y_vals = f.(x_vals)

    plot!(p, x_vals, y_vals, label="sin(cos(eˣ))", linewidth=2, legend=false)
    hline!([0], color=:black, linewidth=1, label="y=0")

    xlims!(-10, 10)
    ylims!(-10, 100)

    a, b = -8, 8

    for _ in 1:t
        global a, b
        x, y = next_secant(f, a, b)


        scatter!(p, [a], [f(a)], label="a", markersize=3)
        scatter!(p, [b], [f(b)], label="b", markersize=3)

        vline!([b], color=:green, linewidth=1, linestyle=:dash, linesize=5)
        vline!([a], color=:green, linewidth=1, linestyle=:dash, linesize=5)

        plot!([a, b], [f(a), f(b)], color=:red, linewidth=1)

        a = x
        b = y
        scatter!(p, [], [])
    end

```

```
    title!("Secant method for f(x) = (x - 1)^2 - 8)")
    p
end
```

Question # 02

a. P_3 for $f(x) = \sqrt{x} - \cos(x)$ on [0, 1]

The most accurate is secant followed by bisection and then fixed

```
("[bisection]: 0.6875", "[fixed]: 0.917172409958988", "[secant]: 0.641725945119874")
```

```
let
    f(x) = x^0.5 - cos(x)
    a, b = 0, 1
    f_bisect = bisection(f, a, b, iters=3)
    g(x) = (cos(x))^2
    f_fixed = fixed_point(g, a, iters=3)
    f_secant = secant(f, a, b, iters=3)
    ("[bisection]: $f_bisect", "[fixed]: $f_fixed", "[secant]: $f_secant")
end
```

```
0.5             1
0.5             0.75
0.625               0.75
1.0
0.2919265817264289
0.917172409958988
1       0.6850733573260451
0.6850733573260451      0.643753386653444
0.643753386653444       0.641725945119874
```

b. P_3 for $f(x) = 3(x + 1)(x - 0.5)(x - 1)$ on [-2, 1.5]

```
("[bisection]: -0.90625", "[fixed]: -50.64102564102565", "[secant]: 1.0")
```

```
let
    f(x) = 3*(x+1)*(x-0.5)*(x-1)
    a, b = -2, 1.5
    f_bisect = bisection(f, a, b, iters=3)
    g(x) = (3 / ((x-0.5)*(x-1))) - 1
    f_fixed = fixed_point(g, a, iters=3)
    f_secant = secant(f, a, b, iters=3)
    ("[bisection]: $f_bisect", "[fixed]: $f_fixed", "[secant]: $f_secant")
end
```

```
-2              -0.25
-1.125              -0.25
-1.125              -0.6875
-0.6
0.7045454545454544
-50.64102564102565
1.5     1.0
```

c. Within 0.01 for $f(x) = x^3 - 7x^2 + 14x - 6$ on [3.2, 4]

```
("[bisection]: 3.41484375", "[fixed]: 3.4142131528865782", "[secant]: 3.414213562369419(
```

```
let
    f(x) = x^3-7x^2+14x-6
    a, b = 3.2, 4
    f_bisect = bisection(f, a, b)
    g(x) = 6 / (x^2-7x+14)
    f_fixed = fixed_point(g, a)
    f_secant = secant(f, a, b)
    ("[bisection]: $f_bisect", "[fixed]: $f_fixed", "[secant]: $f_secant")
end
```

```
3.2                3.6
3.4000000000000004                3.6
3.4000000000000004                3.5
3.4000000000000004                3.45
3.4000000000000004                3.4250000000000003
3.4125000000000005                3.4250000000000003
3.4125000000000005                3.41875
3.4125000000000005                3.4156250000000004
3.4140625000000004                3.4156250000000004
3.2608695652173916
3.3200836820083706
3.366304638753767
3.3939061686727188
3.4066599841838165
3.4115868753762815
3.413324827061882
3.413915808710127
3.414114141803185
3.4141804033015712
3.4142025072488837
3.4142098770982274
3.414212333922196
3.4142131528865782
4        3.2424242424242404
3.2424242424242404       3.2821357943309164
3.2821357943309164       3.674795442889349
3.674795442889349        3.343431945668226
3.343431945668226        3.3798342560662618
3.3798342560662618       3.4233113012531717
3.4233113012531717       3.413290807771148
3.413290807771148        3.4141907852727904
3.4141907852727904       3.4142136206807083
```

d. Within 0.00001 for $f(x) = e^x - x^2 + 3x - 2$ on [0, 1]

```
("[bisection]: 0.2568359375", "[fixed](Diverges): NaN", "[secant]: 0.257530285452674")
```

```julia
let
    f(x) = e^x - x^2 + 3x - 2
    a, b = 0, 1
    f_bisect = bisection(f, a, b)
    g(x) = e^x - 3x + 2
    f_fixed = fixed_point(g, a)
    f_secant = secant(f, a, b)
    ("[bisection]: $f_bisect", "[fixed](Diverges): $f_fixed", "[secant]: $f_secant")
end
```

```
0                0.5
0.25                0.5
0.25                0.375
0.25                0.3125
0.25                0.28125
0.25                0.265625
0.25                0.2578125
0.25390625              0.2578125
0.255859375             0.2578125
3.0
13.085536923187668
481884.4392851071
Inf
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
NaN
```

e. Within 0.00001 for f(x) = 2xcos(2x)-(x+1)² on [-3, -2]

```
("[bisection]: -2.1904296875", "[fixed]: 3.4199683913634524e180", "[secant]: -2.1913080
```

```
let
    f(x) = 2x*cos(2*x)-(x+1)^2
    a, b = -3, -2
    f_bisect = bisection(f, a, b)
    g(x) = (x+1)^2/(2*cos(2*x))
    f_fixed = fixed_point(g, a,iters=10)
    f_secant = secant(f, a, b,iters=1000)
    ("[bisection]: $f_bisect", "[fixed]: $f_fixed", "[secant]: $f_secant")
end
```

```
-2.5                    -2
-2.25                   -2
-2.25                   -2.125
-2.25                   -2.1875
-2.21875                    -2.1875
-2.203125                   -2.1875
-2.1953125                  -2.1875
-2.19140625                 -2.1875
-2.19140625                 -2.189453125
2.0829638531902153
-9.144956360931152
39.141387729616966
-833.033591818722
664706.106010415
-2.2366038408818787e11
2.732936960859598e22
2.175185952229039e45
-2.5687962170835077e90
3.4199683913634524e180
-2        -2.1419331747181767
-2.1419331747181767     -2.2012800280775195
-2.2012800280775195     -2.1909137553700675
-2.1909137553700675     -2.1913050605596935
-2.1913050605596935     -2.1913080126822
```

Question # 03

Results of iterations are printed in each function call!

a) **P3 for $f(x) = \sqrt{x} - \cos x = 0$ on [0,1]:** Fixed point iteration converged rapidly, confirming the function's behavior in this interval.

b) **P3 for $f(x) = 3(x+1)(x-1.2)(x-1) = 0$ on [-2,1.5]:** The bisection method effectively located the root, showcasing its reliability for polynomial functions.

c) **Within $10^{-2} for f(x) = x^3 - 7x^2 + 14x - 6 = 0$ on [3.2,4]:** All three methods resulted in a good precision value of 3.414...

d) **Within $10^{-5} for f(x) = e^x - x^2 + 3x - 2 = 0$ on [0,1]:** Both fixed point and secant methods got the roots, where as the choice of g for fixed point resulted in divergence.

e) **Within** $10^{-5} for f(x) = 2x\cos(2x) - (x+1)^2 = 0$ **on [-3,-2]:** The secant and bisection methods provided a reliable solution, where as for fixed point the starting point -3 is not a good choice.