Numerical Computation Programming Practice Assignment 1

Course Name	Numerical Computation算
Assignment Requirements	**2.3 Use Newton's method to find:** 1. **The positive root** of the equation x3-x2-x-1=0, and 2. **The smallest positive root** of the equation cosx=21+sinx. **Requirement:** The iterative difference must satisfy ｜xk-xk-1｜< 1/2×10-5. Write a program to implement Exercise 2.3. Determine the convergence of the solution to Exercise 2.3, handwrite and photograph the steps, and paste them into this report (only the derivation of the iteration formula is needed). Write a program to verify it; do not call existing library functions, but complete the core algorithm yourself. You do not need to draw a flowchart or write a principle overview. The derivation part should be slightly detailed, and comments should be added to the code to explain parameter settings. The data results should be detailed and aesthetically pleasing; necessary identification and explanatory analysis should be provided for the experimental results.
Purpose of Programming Practice	The process and convergence of Newton's Iteration Method
Programming Practice Environment	Terminal Julia

Content (Algorithm, Program, Steps, and Method)	(code below)

```julia
using Printf    # External formatting library
# Reading the original function:
println("Enter the function:")
user_input_f = readline()
f = eval(Meta.parse("x -> $user_input_f"))

# Reading its derivative:
println("Enter its derivative:")
user_input_df = readline()
df = eval(Meta.parse("x -> $user_input_df"))

# Reading the initial value of the recursion:
println("Enter the initial value:")
x0 = parse(Float64, readline())

# Reading the tolerance:
println("Enter your precision:")
e = parse(Float64, readline())

# Reading the maximum iteration:
println("Enter the max iteration:")
max_iter = parse(Int, readline())

function newton(f, df, x0, e, max_iter)
    flag = false
    count = 0
    x = x0
    x_new = x0

    while true
        fx = f(x)
        dfx = df(x)

        x_new = x - fx / dfx    # The recursion function
        count += 1  # The iteration counter

        if abs(x_new - x) <= e
            flag = true
            break
        elseif count <= max_iter
            x = x_new
        else
            break
        end
    end

    if flag
        d = max(0, -floor(Int, log10(e)))
        format = Printf.Format("The root is: %." * string(d) * "f\n")
        result = Printf.format(format, x_new)
        print(result)
```

	<pre> else
	        println("Failed to converge within $max_iter iterations...")
	    end
	end
	
	newton(f, df, x0, e, max_iter)</pre>

Results (Text or Screenshot, use text as much as possible to reduce storage) and Analysis	Output results: >>> Enter the function: x^3-x^2-x-1 Enter its derivative: 3*x^2-2x-1 Enter the initial value: 2 Enter your precision: 5e-6 Enter the max iteration: 100 The root is: 1.839287 >>> Enter the function: sin(x)-cos(x)+1/2 Enter its derivative: cos(x)+sin(x) Enter the initial value: pi/6 Enter your precision: 5e-6 Enter the max iteration: 100 The root is: 0.424031
Summary	Through this programming practice, I have mastered: The programming method for the Newton iteration formula, and further understood the problems encountered during the Julia programming process: A problem with the format of the error appeared in the input/output processing. I solved it this way: I sought the help of an AI large model and solved the error format problem.

Instructions/Notes

Section Name	Description
Environment	The software and hardware environment (configuration) used for the programming practice.
Content (Algorithm, Program, Steps, and Method)	This is extremely important content for the programming practice report. This section should clearly state the principles, laws, algorithms, or operating methods used for the programming practice, and the steps involved, especially steps that differ from the content taught in class. If necessary, a flowchart (or structural diagram of the programming practice setup) should be drawn, accompanied by corresponding text explanations, which can save a lot of text and make the report concise and clear.
Data Recording and Calculation	Refers to the data measured and the calculation results from the programming practice.
Conclusion (Results)	That is, drawing conclusions based on the phenomena observed and the data measured during the programming practice process.
Summary	The experience, thoughts, and suggestions for this programming practice. You can write about the reasons for the success or failure of the practice, post-practice reflections, suggestions, etc.