

Numerical Analysis Homework 1

Chen Wanqi 3220102895 *

Information and Computer Science 2201, Zhejiang University

September 26, 2024

I. Consider the bisection method starting with the initial interval $[1.5, 3.5]$.

I-a. Width of the interval at the n th step

Solution:

The bisection method halves the interval at each step.

For the interval $[1.5, 3.5]$, the initial width is $3.5 - 1.5 = 2$. Therefore, the width at the n -th step is:

$$W_n = \frac{2}{2^n} = \frac{1}{2^{n-1}}$$

I-b. Supremum of the distance between the root r and the midpoint of the interval

Solution:

The distance between the root r and the midpoint of the interval after n steps, denoted as D_n , is always less than or equal to half the width of the interval. In the case of the interval $[1.5, 3.5]$, this becomes:

$$D_n = \frac{2}{2^{n+1}} = \frac{1}{2^n}$$

II. Proof of accuracy with relative error ϵ

We want to determine the number of steps n such that the relative error of the approximation to the root is no greater than ϵ . Specifically, we need to show that this goal is achieved if:

$$n \geq \frac{\log(b_0 - a_0) - \log(\epsilon) - \log(a_0)}{\log(2)} - 1$$

Proof: In the bisection method, the width of the interval after n steps is given by:

$$W_n = \frac{b_0 - a_0}{2^n}$$

This width bounds the absolute error of the root approximation. The midpoint of the interval is the best approximation of the root, and the error in this approximation is at most half the interval width:

$$E_n = \frac{W_n}{2} = \frac{b_0 - a_0}{2^{n+1}}$$

To achieve a relative error no greater than ϵ , the following condition must hold:

$$\frac{E_n}{r} \leq \epsilon$$

Since the root r lies in the interval $[a_0, b_0]$, and $r \geq a_0$, we have:

$$\frac{\frac{b_0 - a_0}{2^{n+1}}}{a_0} \leq \epsilon$$

*Electronic address: 3220102895@zju.edu.cn

Simplifying this:

$$\frac{b_0 - a_0}{2^{n+1}a_0} \leq \epsilon$$

Multiplying both sides by $2^{n+1}a_0$:

$$b_0 - a_0 \leq 2^{n+1}\epsilon a_0$$

Taking the logarithm of both sides:

$$\log(b_0 - a_0) \leq \log(2^{n+1}\epsilon a_0)$$

Using logarithm properties:

$$\log(b_0 - a_0) \leq (n+1)\log(2) + \log(\epsilon) + \log(a_0)$$

Rearranging to solve for n :

$$n+1 \geq \frac{\log(b_0 - a_0) - \log(\epsilon) - \log(a_0)}{\log(2)}$$

Subtracting 1 from both sides gives:

$$n \geq \frac{\log(b_0 - a_0) - \log(\epsilon) - \log(a_0)}{\log(2)} - 1$$

Thus, the number of steps $n \geq \frac{\log(b_0 - a_0) - \log(\epsilon) - \log(a_0)}{\log(2)} - 1$ guarantees that the relative error in the root approximation is no greater than ϵ . \square

III. Perform four iterations of Newton's method for the polynomial equation

The polynomial equation is $p(x) = 4x^3 - 2x^2 + 3 = 0$ with the starting point $x_0 = -1$.

Solution:

Newton's method is defined by the iterative formula:

$$x_{n+1} = x_n - \frac{p(x_n)}{p'(x_n)}$$

Given the polynomial $p(x) = 4x^3 - 2x^2 + 3$, its derivative is:

$$p'(x) = 12x^2 - 4x$$

Starting with $x_0 = -1$, we apply the Newton's method formula for four iterations:

$$x_{n+1} = x_n - \frac{4x_n^3 - 2x_n^2 + 3}{12x_n^2 - 4x_n}$$

The iterations are organized in the following table:

n	x_n	$p(x_n)$	$p'(x_n)$
0	-1.00000	-3.00000	16.00000
1	-0.81250	-0.46582	11.17188
2	-0.77080	-0.02014	10.21289
3	-0.76883	-0.00004	10.16857
4	-0.76883	-0.00000	10.16847

Here I use a Python script (`./src/T4.py`) to help me get the answer!

IV. Consider a variation of Newton's method

In this variation of Newton's method, only the derivative at x_0 is used for all iterations. The update rule is given by:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}$$

Let α be the true root of the function $f(x)$. The error at step n is defined as:

$$e_n = x_n - \alpha$$

To analyze the convergence behavior of this method, we aim to find constants C and s such that:

$$e_{n+1} = C e_n^s$$

where e_{n+1} is the error at step $n+1$, s is a constant, and C may depend on x_n , the true root α , and the derivative of $f(x)$.

Solve:

Assume $f(x)$ is sufficiently smooth and can be expanded in a Taylor series around α . Thus, for x_n near α , we can write:

$$f(x_n) = f(\alpha) + f'(\alpha)(x_n - \alpha) + \frac{f''(\alpha)}{2}(x_n - \alpha)^2 + O((x_n - \alpha)^3)$$

Since α is the true root, we know that $f(\alpha) = 0$, so this simplifies to:

$$f(x_n) = f'(\alpha)(x_n - \alpha) + \frac{f''(\alpha)}{2}(x_n - \alpha)^2 + O((x_n - \alpha)^3)$$

Using the modified Newton's method iteration rule:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}$$

Substitute the Taylor expansion of $f(x_n)$:

$$x_{n+1} = x_n - \frac{f'(\alpha)(x_n - \alpha) + \frac{f''(\alpha)}{2}(x_n - \alpha)^2 + O((x_n - \alpha)^3)}{f'(x_0)}$$

Let $e_n = x_n - \alpha$. Then the update equation becomes:

$$e_{n+1} = e_n - \frac{f'(\alpha)e_n + \frac{f''(\alpha)}{2}e_n^2 + O(e_n^3)}{f'(x_0)}$$

For small e_n , the quadratic term e_n^2 dominates. Therefore, the error at step $n+1$ can be approximated by:

$$e_{n+1} \approx -\frac{f''(\alpha)}{2f'(x_0)}e_n^2$$

This shows that the error follows a quadratic convergence pattern, with:

$$e_{n+1} = C e_n^2$$

where $C = -\frac{f''(\alpha)}{2f'(x_0)}$.

V. Convergence of the Iteration $x_{n+1} = \tan^{-1} x_n$

We are asked to determine whether the iteration $x_{n+1} = \tan^{-1} x_n$ will converge within the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$.

Solution:

Let $f(x) = \tan^{-1} x$ be the iteration function. Thus, we have the recurrence relation:

$$x_{n+1} = f(x_n) = \tan^{-1} x_n.$$

We need to determine whether the iteration converges within the interval

$$\left(-\frac{\pi}{2}, \frac{\pi}{2}\right).$$

To find the fixed point α , we solve the equation:

$$f(\alpha) = \alpha,$$

which leads to:

$$\tan^{-1} \alpha = \alpha.$$

This equation holds when $\alpha = 0$, since $\tan^{-1}(0) = 0$. Therefore, the fixed point of the iteration is $\alpha = 0$.

Next, to prove that $f(x)$ is a contraction mapping, we need to show that there exists a constant k with $0 \leq k < 1$, such that for all x, y in the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$, the following inequality holds:

$$|f(x) - f(y)| \leq k|x - y|.$$

We calculate the derivative of $f(x) = \tan^{-1} x$:

$$f'(x) = \frac{1}{1+x^2}.$$

Since $f'(x)$ is continuous and $0 \leq f'(x) < 1$ for all $x \in (-\frac{\pi}{2}, \frac{\pi}{2})$, this shows that $f(x)$ is a contraction mapping. Specifically, the contraction constant k is $\frac{1}{1+x^2}$, which is less than 1 for all $x \neq 0$.

By the Contraction Mapping Theorem:

If $g(x)$ is a continuous contraction on $[a, b]$, then it has a unique fixed point in $[a, b]$. Furthermore, the fixed-point iteration converges to α for any initial $x_0 \in [a, b]$.

Since $f(x) = \tan^{-1} x$ is a contraction and $x = 0$ is the unique fixed point, the iteration $x_{n+1} = \tan^{-1} x_n$ will converge to $x = 0$ for any initial point $x_0 \in (-\frac{\pi}{2}, \frac{\pi}{2})$. □

VI. Continued Fraction with $p > 1$

Consider the continued fraction:

$$x = \frac{1}{p + \frac{1}{p + \frac{1}{p + \dots}}}$$

We want to find the value of x and prove that the sequence of values converges.

Solution:

Let $x_1 = \frac{1}{p}$, $x_2 = \frac{1}{p + \frac{1}{p}}$, and so forth. We need to show that $\lim_{n \rightarrow \infty} x_n = x$.

The recursive relation is:

$$x_{n+1} = \frac{1}{p + x_n}.$$

This sequence can be interpreted as the fixed point iteration for the function:

$$f(x) = \frac{1}{p + x}.$$

Since $f'(x) = -\frac{1}{(p+x)^2}$ and $|f'(x)| < 1$ for $p > 1$, $f(x)$ is a contraction mapping.

By the Contraction Mapping Theorem, the sequence x_n converges to the unique fixed point of $f(x)$, which is the solution of the equation $x = \frac{1}{p+x}$.

Therefore, the sequence of values converges to:

$$x = \frac{\sqrt{p^2 + 4} - p}{2}.$$

□

VII. Modified Bisection Method for $a_0 < 0 < b_0$

In this case, we start with an initial interval $[a_0, b_0]$, where $a_0 < 0$ and $b_0 > 0$, and wish to derive a similar inequality for the number of steps n needed to achieve a certain accuracy.

VII-a. Number of Steps Inequality

Using the same reasoning as in problem II, the width of the interval after n steps of bisection is given by:

$$W_n = \frac{b_0 - a_0}{2^n}.$$

To achieve an absolute error smaller than ϵ , we require:

$$\frac{b_0 - a_0}{2^n} \leq \epsilon.$$

Solving for n , we get:

$$n \geq \frac{\log(b_0 - a_0) - \log(\epsilon)}{\log 2}.$$

VII-b. Relative Error as a Measure

When $a_0 < 0 < b_0$, the root r lies between negative and positive values. In this case, the relative error, defined as $\frac{|x_n - r|}{|r|}$, can become problematic if r is close to zero, as it would lead to division by a very small number or even undefined behavior when $r = 0$.

Thus, in this situation, the relative error is not an appropriate measure of accuracy. Instead, absolute error should be used, particularly when the root is near zero. □

VIII. Multiple Zeros and Modified Newton's Method

Consider solving $f(x) = 0$ by Newton's method with a starting point x_0 close to a root of multiplicity k . Recall that α is a zero of multiplicity k for f if:

$$f^{(k)}(\alpha) \neq 0, \quad \forall i < k, \quad f^{(i)}(\alpha) = 0.$$

VIII-a. Detecting Multiple Zeros

Multiple zeros can be detected by examining the behavior of the sequence $(x_n, f(x_n))$. Specifically, if α is a multiple zero of f , the convergence rate of the Newton sequence slows down. The typical quadratic convergence for simple roots is reduced to linear convergence in the case of multiple roots. This can be observed by noting that the differences $x_{n+1} - \alpha$ decrease much slower than expected in the vicinity of the root.

Another approach is to look at the ratios:

$$\frac{f(x_n)}{f'(x_n)}.$$

If this ratio remains relatively large near α , it is an indication of a multiple zero, as this value tends to decrease faster for simple roots.

VIII-b. Modified Newton's Method for Multiple Roots

If r is a zero of multiplicity k of the function f , then quadratic convergence can be restored by modifying Newton's iteration as follows:

$$x_{n+1} = x_n - k \frac{f(x_n)}{f'(x_n)}.$$

Proof:

Given that α is a root of multiplicity k , we can express the function $f(x)$ near α as:

$$f(x) = (x - \alpha)^k g(x),$$

where $g(x)$ is a smooth function and $g(\alpha) \neq 0$. Then, the derivative is given by:

$$f'(x) = k(x - \alpha)^{k-1} g(x) + (x - \alpha)^k g'(x).$$

At $x = \alpha$, this simplifies to:

$$f'(x) \approx k(x - \alpha)^{k-1} g(\alpha).$$

The standard Newton's method update step is:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Since $f(x)$ vanishes at a higher rate for multiple roots, this iteration converges linearly instead of quadratically. By modifying the update rule to:

$$x_{n+1} = x_n - k \frac{f(x_n)}{f'(x_n)},$$

we effectively restore the quadratic convergence because the multiplicity k compensates for the slower convergence rate induced by the multiple root. □

References

- handoutsNumPDEs
- ChatGPT, *AI Language Model*, OpenAI Platform, 2024.