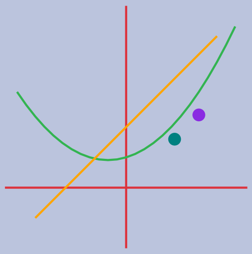
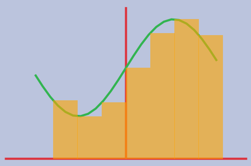
NUMERICAL METHODS

August 2025

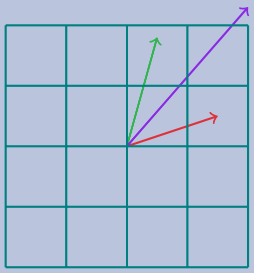
DS 288 and UMC 202



Newton’s Method



Integration



Linear Algebra

Ratikanta Behera

Department of Computational and Data Sciences
Indian Institute of Science, Bangalore, IN

November 14, 2025

Contents

Introduction to Numerical Methods

1.1. Numerical Methods

Numerical methods are mathematical techniques used to solve problems that cannot be solved analytically or where analytical solutions are impractical. These methods provide approximate solutions using computational algorithms. They are essential because many real-world problems lack closed-form solutions, and complex systems require computational approaches. Also, it is important to note that engineering and scientific computing rely heavily on numerical methods.

1.2. Scientific Computing

Scientific computing brings together mathematical modelling, numerical analysis, and computer science to solve problems arising from science and engineering. The mathematical backbone of scientific computing is *numerical mathematics*. Numerical methods are finite-step procedures aimed at computing approximate solutions to a prescribed precision.

1.3. Numerical Solutions

Numerical solutions are not analytical solutions; these are approximate solutions that have limited precision of computation and, hence, are less accurate than analytical solutions. There may be multiple numerical methods to solve the same problem.

A **numerical calculation** is the direct manipulation of numbers to find a result. Example: $\frac{(17.36)^2-1}{17.36+1} = 16.36$. A **symbolic calculation** is the manipulation of symbols according to algebraic rules. Example: $\frac{x^2-1}{x+1} = x - 1$

An **analytical solution** is an exact representation, possibly in terms of fractions, π, e, etc., while a **numerical solution** is a decimal approximation: $0.25,\ 3.14159\ldots$

1.4. Numerical Approximations

Floating Point Representation

A nonzero real x in normalized k-digit decimal floating point form is,

$$x = \pm(0.a_1a_2\ldots a_k) \times 10^n, \quad a_1 \neq 0$$

where n is the *exponent*, and $(0.a_1a_2\ldots a_k)$ is the *mantissa*.

Example 1.4.1. $12345.67 \to 0.1234567 \times 10^5, \quad 0.00123 \to 0.123 \times 10^{-2}$.

Significant Figures

Significant figures indicate the precision of a number. There are some of the rules for the calculation of the number of significant figures,

1. All nonzero digits are significant.
2. Zeros between nonzero digits are significant.

3. Leading zeros are not significant.

Example 1.4.2. 549 *has* 3 *significant figures;* 0.000034 *has* 2 *significant figures.*

1.4.1. Approximation and Rounding

Most real numbers in decimal form cannot be exactly represented in floating-point. Given,

$$x = \pm(0.a_1 a_2 \ldots a_k a_{k+1} \ldots) \times 10^n,$$

round to k digits by,

$$a_k^\star = \begin{cases} a_k, & a_{k+1} < 5, \\ a_k + 1, & a_{k+1} \geq 5. \end{cases}$$

1.5. Basics and Pre-requisites

Continuity of a Function

A function f is *continuous* at a point c if its graph has no break, jump, or hole at c. More precisely,

Definition 1.5.1 (Continuity at a Point)**.** A function $f : X \to Y$ is **continuous at** $c \in X$ if,

$$\lim_{x \to c} f(x) = f(c).$$

That is, the left-hand limit, right-hand limit, and the function value at c exist and are equal,

$$\lim_{x \to c^-} f(x) = \lim_{x \to c^+} f(x) = f(c).$$

Definition 1.5.2 (Continuity on an Interval)**.** f is **continuous on an open interval** (a, b) if it is continuous at every point in (a, b). And, f is **continuous on a closed interval** $[a, b]$ if,

$$f \text{ is continuous on } (a, b), \quad \lim_{x \to a^+} f(x) = f(a), \quad \text{and} \quad \lim_{x \to b^-} f(x) = f(b).$$

Differentiability of a Function

Definition 1.5.3 (Differentiability)**.** A function f is **differentiable at** c if the derivative,

$$f'(c) = \lim_{h \to 0} \frac{f(c+h) - f(c)}{h} \quad \text{exists.}$$

Note: Differentiability implies continuity, but continuity does not necessarily imply differentiability. For example, functions with cusps or corners are continuous but not differentiable.

Taylor Series

Theorem 1.5.4 (Taylor's Theorem)**.** *Suppose f is $n+1$ times differentiable at $x = a$. Then around point a, $f(x)$ can be expressed as,*

$$f(x) = \mathcal{P}_n(x) + \mathcal{R}_n(x) = \sum_{i=0}^{n} \frac{f^{(i)}(a)}{i!}(x-a)^i + \mathcal{R}_n(x).$$

Here, $f^{(j)}(a)$ is the j^{th} derivative of f at $x = a$, and $j!$ denotes factorial. Also, $\mathcal{P}_n(x)$ is polynomial of degree n and, $\mathcal{R}_n(x)$ is remainder of degree $n+1$ given as,

$$\mathcal{R}_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1} \qquad \textit{where, } \xi \in [a, x]$$

NOTE: limiting cases are as follows,

$$\lim_{n\to\infty} \mathcal{P}_n(x) = f(x) \qquad \text{and, } \lim_{n\to\infty} \mathcal{R}_n(x) = 0$$

Mean Value Theorem

Theorem 1.5.5 (Mean Value Theorem). *If f is continuous on the closed interval $[a, b]$ and differentiable on the open interval (a, b), then there exists some $c \in (a, b)$ such that,*

$$f'(c) = \frac{f(b) - f(a)}{b - a}.$$

Interpretation. There exists a point c where the instantaneous rate of change (the tangent slope) equals the average rate of change over $[a, b]$ (the secant slope).

Intermediate Value Theorem

Theorem 1.5.6 (Intermediate Value Theorem). *If f is continuous on a closed interval $[a, b]$ and k is any number between $f(a)$ and $f(b)$, then there exists some $c \in (a, b)$ such that,*

$$f(c) = k.$$

NOTE: If $f(a)f(b) < 0$, then f has at least one root in (a, b).

1.6. Errors in Numerical Computation

1. **Inherent error:** Present in the problem data itself. It can occur due to modelling considerations as well.
2. **Rounding error:** Due to rounding in intermediate steps.
3. **Truncation error:** Approximation of infinite processes by finite ones. The remainder term $\mathcal{R}_n(x)$ in Theorem 1.5.4 is the incorporated truncation error in a function approximation.

Rounding errors are one of the most important errors when we deal with machines (eg, computers) because in machines,

$$dx + dx + dx + \cdots + \text{n times} \neq n \times dx$$

Because if we assume for a single operation, the machine makes a round-off error of 0.1, then $dx + dx + dx + \ldots$ will result in $0.1 \times n$ error while $n \times dx$ results in 0.1 error. Hence, it is very crucial to decide the operations and algorithms while working with machines.

1.6.1. Absolute, Relative, and Percentage Errors

Let x be the true value and x' be the approximation,

$$\varepsilon_{\text{absolute}} = |x - x'|, \quad \varepsilon_{\text{relative}} = \frac{|x - x'|}{|x|}, \quad \varepsilon_{\text{percentage}} = 100 \cdot \varepsilon_{\text{relative}}$$

Relative and percentage errors are unit-independent, while absolute error takes the unit of the true value.

NOTE I: When the true value x is unknown and one is working with an iterative method to solve some problem, then absolute error can be considered as $|x_n - x_{n-1}|$.

NOTE II: Relative errors are more meaningful, as there are cases when absolute error analysis fails. Assume $x = 0.01$ and $x' = 0.012$, here, $\varepsilon_{\text{absolute}} = 0.002$ while $\varepsilon_{\text{relative}} = 0.2$ which is very high and makes more sense.

1.7. Convergence

A method is **convergent** if $x_n \to \alpha$ (the true root) as $n \to \infty$. Or, simply, we can say that,

$$\lim_{n\to\infty} x_n = \alpha$$

Order of convergence: If we have the relation of type,

$$|x_{n+1} - \alpha| \leq \lambda |x_n - \alpha|^p,$$

then p is the order of convergence (or, asymptotic convergence rate (if $n \to \infty$)), and λ is the asymptotic error constant. It is important to note that p has a primary effect on convergence while λ has a secondary effect on convergence (assume $p = 1$, then the method will be convergent iff $|\lambda| < 1$).

Some of the general orders of convergence are as follows,

1. $p = 1$: Linear convergence.
2. $1 < p < 2$: Superlinear convergence.
3. $p = 2$: Quadratic convergence.
4. $p = 3$: Cubic convergence.

Root Finding Methods

Solving Nonlinear Equations

We often want to find roots of $f(x) = 0$, where f may be algebraic or transcendental. Polynomials of degree ≤ 4 can be solved exactly, but for higher degrees or more complicated functions, numerical methods are necessary. Typical methods include Bracketing methods (Bisection, Regula Falsi) and Open methods (Newton-Raphson, Secant, Fixed-Point Iteration).

2.1. Bisection Method

We want to find the root of $f(x)$ in the interval $[a, b]$ where we are given only that $f(a)f(b) < 0$. As the Intermediate Value Theorem 1.5.6 proves that there exists at least 1 root in $[a, b]$. Hence, the bisection method follows the following steps,

1. Compute midpoint $m = (a + b)/2$.
2. If $f(m) = 0$, stop.
3. Else replace interval with half that contains the root (using sign check).
4. Repeat until desired accuracy.

Error bound: After n steps, interval length is $\frac{b-a}{2^n}$. It is easy to prove that, for tolerance ϵ,

$$n \geq \frac{\log(b-a) - \log \epsilon}{\log 2}.$$

Example 2.1.1. *For $x^3 + x - 4 = 0$ in $(1, 4)$ with $\epsilon = 10^{-3}$,*

$$n \geq \frac{\log(3) - \log(10^{-3})}{\log 2} \approx 12 \textit{ iterations.}$$

2.1.1. Error Analysis for the Bisection Method

Let $f \in C[a, b]$ i.e., continuous in $[a, b]$ and suppose $f(a)f(b) < 0$. Denote the initial interval by $[a_0, b_0] = [a, b]$. Then, the bisection method generates a sequence of midpoints,

$$x_n = \frac{a_{n-1} + b_{n-1}}{2}$$

that converges to a root α of function f.

Error Bound and Convergence: At each iteration, the interval length is halved,

$$b_n - a_n = \frac{1}{2}(b_{n-1} - a_{n-1}) = \frac{1}{2^n}(b_0 - a_0) = \frac{1}{2^n}(b - a).$$

Since $\alpha \in [a_n, b_n]$ and x_n is the midpoint,

$$|x_n - \alpha| \leq \frac{1}{2}(b_{n-1} - a_{n-1}) = \frac{b-a}{2^n}, \quad n \geq 1.$$

Thus, the method is guaranteed to converge and the error decreases at least by a factor of 2 each iteration.

Order of Convergence: The error after n steps is,

$$|e_n| = |x_n - \alpha| \leq \frac{b-a}{2^n}.$$

Therefore,

$$\frac{|e_{n+1}|}{|e_n|} \leq \frac{1}{2}, \qquad n = 1, 2, 3, \dots$$

This shows that the bisection method converges **linearly** (order $p = 1$) with asymptotic error constant $k \leq 0.5$.

Remarks and Stopping Criteria:

1. The condition $f(a)f(b) < 0$ ensures the existence of at least one root, but not its uniqueness; the sequence will converge to one of them.
2. Stopping criteria when the exact root is unknown (using threshold ϵ),
   (a) Successive-iterate test: $|x_n - x_{n-1}| < \epsilon$.
   (b) Relative error test: $\frac{|x_n - x_{n-1}|}{|x_n|} < \epsilon$.
   (c) Small function value: $|f(x_n)| < \epsilon$.

Advantages:

1. Guaranteed convergence for continuous f with $f(a)f(b) < 0$.
2. Error reduces by at least half each step.

Disadvantages:

1. Converges only linearly (slow).
2. Requires two initial guesses with opposite signs.
3. Cannot detect complex roots; fails if f has same sign at both ends.
4. No benefit from choosing initial guesses close to the root.
5. May fail on some functions (e.g., $f(x) = x^2$ has no bracketing interval with $f(a)f(b) < 0$).

2.2. Regula-Falsi (False-Position) Method

The Regula Falsi method is a bracketing method for finding roots of a function $f(x)$. It combines features of the bisection method and linear interpolation, and is applicable when f is continuous on $[a, b]$ and $f(a)f(b) < 0$. It still requires $f(a)f(b) < 0$ and produces a new approximation that (in many cases) converges faster.

Idea of Regula-Falsi Method. Given (a_0, b_0) with $f(a_0)f(b_0) < 0$, join the points $(a_0, f(a_0))$ and $(b_0, f(b_0))$ by a straight line (secant). The x-intercept of this line is taken as the next approximation to the root.

Derivation of Regula-Falsi Method

Given interval $[a, b]$ with $f(a)f(b) < 0$, the straight line through $(a, f(a))$ and $(b, f(b))$ has equation,

$$y - f(a) = \frac{f(b) - f(a)}{b - a}(x - a).$$

Setting $y = 0$ (where the chord crosses the x-axis) gives the next approximation,

$$x_1 = a - f(a)\frac{b - a}{f(b) - f(a)}.$$

We then check the sign of $f(x_1)$ to decide the new bracketing interval,

$$\begin{cases} \text{If } f(a)f(x_1) < 0 & \Rightarrow [a, x_1], \\ \text{If } f(x_1)f(b) < 0 & \Rightarrow [x_1, b]. \end{cases}$$

General iteration formula (with x_{n-1}, x_n as endpoints),

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}, \quad n = 1, 2, 3, \ldots$$

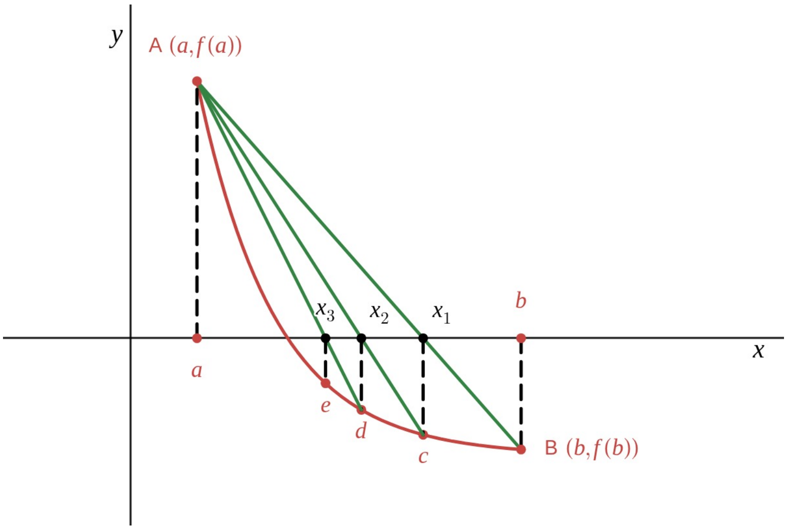


Figure 2.1: Graphical representation for the Regula-Falsi (False-Position) Method

NOTE: For functions such as $f(x) = x^{10} - 1$ on $(0, 1.3)$, the method can stagnate on one endpoint, slowing convergence. The method can also suffer from serious round-off problems when $f(b) - f(a)$ is very small (the difference between two nearly equal numbers).

Example 2.2.1. *Approximate $\sqrt{3}$ to two decimal places: $f(x) = x^2 - 3$, $[a, b] = [1, 2]$.*

- $x_1 = 1 - (-2)\frac{2-1}{1} = 1.6667$, $f(x_1) < 0 \ \Rightarrow [1.6667, 2]$.
- $x_2 \approx 1.7273$, $f(x_2) < 0 \ \Rightarrow [1.7273, 2]$.
- $x_3 \approx 1.7317073$, $x_4 \approx 1.7320262$, $x_5 \approx 1.7320491$.
- $x_8 \approx 1.7320508$ *(correct to two decimals).*

Comparison with Bisection method:

1. Both guarantee the convergence, and both require two initial guesses with opposite signs.
2. Both have order $p = 1$ (linear) **[Prove Yourself !]**, but Regula Falsi often converges faster in practice.
3. However, for some functions (e.g., $x^{10} - 1$), it can stagnate on one endpoint, leading to slow convergence.
4. It is also more prone to very serious round-off errors since the denominator $f(b) - f(a)$ can be a very small difference of nearly equal values.

2.3. Newton's Method (Newton–Raphson Method)

Newton's method is one of the most widely used techniques for approximating roots of differentiable functions. It can also be generalised to systems of nonlinear equations, nonlinear integral equations, and differential equations. It often converges very rapidly if the initial guess is sufficiently close to the root.

Derivation of Newton's Method

Given a differentiable function $f(x)$ and an initial approximation x_0 near the root α, the tangent to $y = f(x)$ at the point $(x_0, f(x_0))$ is given by,

$$y - f(x_0) = f'(x_0)(x - x_0).$$

The x-intercept of this tangent line satisfies,

$$0 - f(x_0) = f'(x_0)(x - x_0) \quad \Rightarrow \quad x = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Thus, the iteration formula is,

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \quad n = 1, 2, \ldots$$

This process is repeated until the sequence $\{x_n\}$ converges.

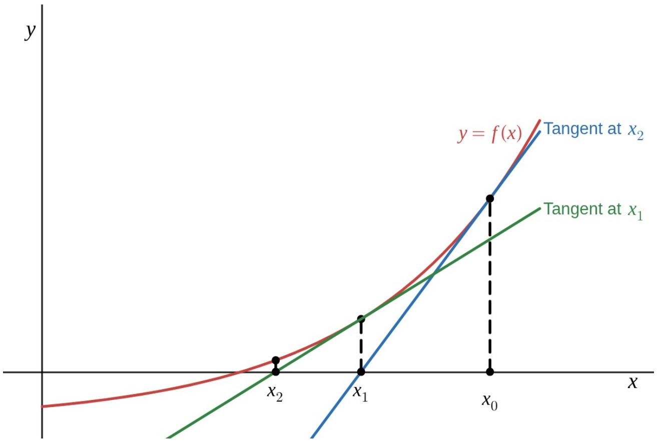


Figure 2.2: Graphical representation for Newton's Method (Newton–Raphson Method)

2.3.1. Convergence Analysis

Let α be a simple root of $f(x)$, with $f'(\alpha) \neq 0$, and $e_n = \alpha - x_n$. From Taylor's Theorem 1.5.4 we get,

$$f(\alpha) = f(x_n) + e_n f'(x_n) + \frac{e_n^2}{2} f''(\zeta_n) = 0, \quad \text{for some } \zeta_n \text{ between } x_n \text{ and } \alpha.$$

Rearranging and substituting into the iteration error, we get,

$$e_{n+1} = -\frac{e_n^2}{2} \frac{f''(\zeta_n)}{f'(x_n)},$$

so that

$$|e_{n+1}| \leq \frac{1}{2} \frac{|f''(\zeta_n)|}{|f'(x_n)|} |e_n|^2,$$

which shows **quadratic convergence** ($p = 2$) when α is a simple root.

Theorem 2.3.1 (Convergence of Newton Method)**.** *If $f \in C^2[a, b]$, $\alpha \in (a, b)$ is a root with $f'(\alpha) \neq 0$, then there exists $\delta > 0$ such that for any initial guess $x_0 \in [\alpha - \delta, \alpha + \delta]$, Newton's method converges to α quadratically.*

NOTE: If $M = \frac{1}{2} \frac{\max |f''(x)|}{\min |f'(x)|}$ on a neighbourhood of α and $M|x_0 - \alpha| < 1$, then Newton's method converges.

Example 2.3.2. *Solve $x^2 - 3 = 0$. To proceed, assume,*

$$f(x) = x^2 - 3, \quad f'(x) = 2x.$$

Hence, the iteration will be of the form,

$$x_n = x_{n-1} - \frac{x_{n-1}^2 - 3}{2x_{n-1}}.$$

*Hence, for $x_0 = 1.5$, $x_1 = 1.75$, $x_2 \approx 1.73205$ (**correct to 8 decimal places in just two steps**). This shows the power of the Newton-Raphson method.*

Advantages:

1. Very fast convergence when it works (quadratic order).
2. Requires only one initial guess.
3. Simple geometric interpretation (tangent line).

Disadvantages:

1. Convergence not guaranteed — requires a good starting point.
2. May converge to an unintended root or diverge.
3. Requires computation of $f'(x) \neq 0$.
4. Slower for multiple roots; near extrema, iteration may oscillate.

2.4. Multiple Roots and Modified Newton's Method

Definition 2.4.1 (Multiplicity of a root)**.** A root α of $f(x)$ has multiplicity m if,

$$f(x) = (x-\alpha)^m q(x), \quad q(\alpha) \neq 0.$$

For a simple root, $m = 1$ and $f'(\alpha) \neq 0$.

For $m > 1$, Newton's method loses quadratic convergence and becomes linear. **[Try to prove yourself !]**

Derivation of Modified Newton Method

Let us define an auxiliary/supporting function,

$$\mu(x) = \frac{f(x)}{f'(x)}.$$

If α is a root of multiplicity m of f, it is a *simple* root of $\mu(x)$. Applying Newton's method to $\mu(x)$ gives,

$$x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{[f'(x_n)]^2 - f(x_n)f''(x_n)}.$$

This is the **Modified Newton's Method** which restores quadratic convergence for multiple roots, at the cost of calculating $f''(x)$ (if it exists, otherwise the method fails).

Example 2.4.2. *Let $f(x) = (x-1.1)^3(x-2.1)$, with $x_0 = 0.5$, the standard Newton method converges very slowly, while the modified method approaches 1.1 in just two steps.*

NOTE: Modified Newton method has quadratic convergence regardless of multiplicity, but it needs $f''(x)$, and subtraction in the denominator can cause round-off issues.

2.5. Secant Method

The secant method is a derivative-free variant of Newton's method, using a finite difference instead of $f'(x)$. It is also known as the discrete version of Newton's method.

Iteration/Update Formula

Starting with two approximations x_{n-2} and x_{n-1},

$$f'(x_{n-1}) \approx \frac{f(x_{n-1}) - f(x_{n-2})}{x_{n-1} - x_{n-2}},$$

Newton's update becomes,

$$x_n = x_{n-1} - f(x_{n-1})\frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}.$$

Geometrically, $(x_{n-1}, f(x_{n-1}))$ and $(x_{n-2}, f(x_{n-2}))$ determine a secant line whose x-intercept is the next iterate. Unlike Regula-Falsi, the interval is not forced to bracket the root, so convergence is not guaranteed.

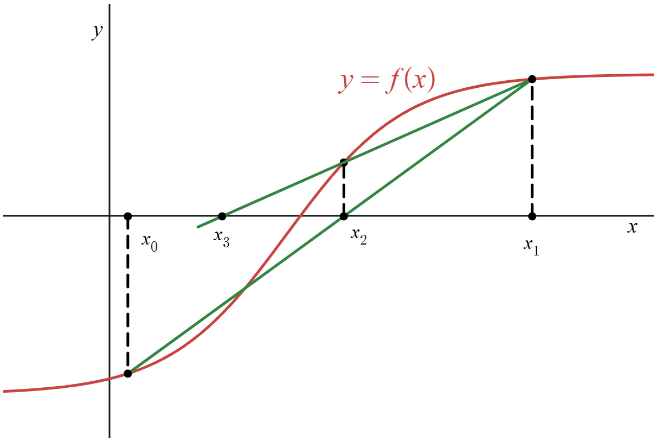


Figure 2.3: Graphical representation for Secant Method

Example 2.5.1. *Solve* $x^3 - 2x - 5 = 0$*, with initial guesses* $x_{-1} = 2, x_0 = 3$*,*

$$x_1 \approx 2.05883, \quad x_2 \approx 2.08126, \quad x_3 \approx 2.09482.$$

2.5.1. Convergence Analysis

Let α be the true root and $e_n = x_n - \alpha$, hence, by update equation we get,

$$e_{n+1} = e_n - f(x_n)\frac{e_n - e_{n-1}}{f(x_n) - f(x_{n-1})} \quad \implies e_{n+1} = \frac{e_{n-1}f(x_n) - e_n f(x_{n-1})}{f(x_n) - f(x_{n-1})},$$

(expanding Taylor Series for $f(x_n)$ and $f(x_{n-1})$ at $x = \alpha$)

$$\implies e_{n+1} = \frac{\frac{1}{2}e_n e_{n-1}\left[e_n f''(\alpha) - e_{n-1}f''(\alpha)\right]}{f'(\alpha)(e_n - e_{n-1}) + \frac{1}{2}f''(\alpha)(e_n^2 - e_{n-1}^2)}$$

Taking the limiting case, we get,

$$\lim_{n\to\infty} e_{n+1} = \frac{1}{2}e_n e_{n-1}\frac{f''(\alpha)}{f'(\alpha)} \quad \implies \quad e_{n+1} \propto e_n e_{n-1}$$

Now assume the order of convergence be p i.e., $e_{n+1} \propto e_n^p$ for some p, and hence we have the following relations, $e_{n+1} \propto e_n^p$, $e_n \propto e_{n-1}^p$, and $e_{n+1} \propto e_n e_{n-1}$.

Substituting which we get a relation of the form,

$$e_{n-1}^p \propto e_{n-1}^{\frac{p+1}{p}}$$

Analyzing which gives the quadratic equation of the form, $p^2 - p - 1 = 0$, which results in a positive value of $p = 1.618$. Hence, the secant method shows **superlinear convergence** ($p = 1.618$)

Advantages:

1. Faster than bisection and regula-falsi (order ≈ 1.618).
2. No need to evaluate derivatives.
3. Only one function evaluation per step after starting.

Disadvantages:

1. No guaranteed convergence; may diverge.
2. Possible division by a small difference $\Rightarrow$ round-off errors.
3. No error bounds on iterates.

2.6. Fixed-Point Method (Fixed-Point Iteration)

Definition 2.6.1 (Fixed Point)**.** A point α is called a *fixed point* of a function $g(x)$ if,

$$g(\alpha) = \alpha.$$

Example 2.6.2. *1.* $g(x) = \sqrt{3x+4} \implies \alpha = 4$ *is a fixed point.*

2. $g(x) = x^2 - x$ *has fixed points at* 0 *and* 2*.*

3. $g(x) = x^2 - 2$ *has fixed points at* -1 *and* 2*.*

Given an equation $f(x) = 0$, it can often be rearranged into the form $x = g(x)$, such that finding a root of f is equivalent to finding a fixed point of g.

Iteration/Update Scheme

Define a sequence via iteration,

$$x_n = g(x_{n-1}), \quad n = 1, 2, 3, \ldots,$$

starting from an initial approximation x_0.

Key questions when using fixed-point iteration are as follows,

1. Does $\{x_n\}$ always converge to a root α ?
2. If it converges, is α a root of $g(x) = x$?
3. How should $g(x)$ be chosen to guarantee convergence ?

2.6.1. Existence, Uniqueness, and Convergence

Theorem 2.6.3. *Suppose g is continuous on $[a, b]$ and maps the interval into itself,*

$$a \le g(x) \le b \quad \forall x \in [a, b].$$

Assume further that $g'(x)$ exists on $[a, b]$ and satisfies,

$$|g'(x)| \le k < 1, \quad \forall x \in [a, b].$$

Then,

1. *g has at least one fixed point $\alpha \in [a, b]$.*
2. *This fixed point is unique.*
3. *For any initial guess $x_0 \in [a, b]$, the sequence defined by $x_n = g(x_{n-1})$ converges to α.*

Example 2.6.4. *Find a suitable $g(x)$ and an interval for the equation $x^3 - x - 1 = 0$, to compute its smallest positive root by fixed-point iteration.*

- *By the Intermediate Value Theorem 1.5.6, the root lies in $[1, 2]$ since $f(1) = -1$ and $f(2) = 5$.*
- *One possible rearrangement is $x = (1 + x)^{1/3}$, we get, $g(x) = (1 + x)^{1/3}$.*
- *Computing the derivative, we get, $g'(x) = \frac{1}{3}(1 + x)^{-2/3}$.*
- *Over $[1, 2]$, $|g'(x)| < 1$ and $g(x)$ maps $[1, 2]$ into itself.*
- *Thus, iteration $x_n = g(x_{n-1})$ converges to the root $\alpha \approx 1.3247$.*

n	0	1	2	3	4	5	6	7
x_n	1.5	1.3572	1.3309	1.3259	1.3249	1.3248	1.3247	1.3247

Proof. (a) **Existence:** Define $h(x) = g(x) - x$. Since $a \le g(x) \le b$, we have,

$$h(a) = g(a) - a \ge 0, \quad h(b) = g(b) - b \le 0.$$

If $h(a) = 0$ or $h(b) = 0$, then a or b is a fixed point. Otherwise, by the Intermediate Value Theorem 1.5.6, there exists $\alpha \in (a, b)$ such that,

$$h(\alpha) = 0 \implies g(\alpha) = \alpha.$$

(b) **Uniqueness:** Assume g has two fixed points $\alpha_1 \neq \alpha_2$. By the Mean Value Theorem 1.5.5,

$$|g(\alpha_1) - g(\alpha_2)| = |g'(\zeta)||\alpha_1 - \alpha_2| \le k|\alpha_1 - \alpha_2|, \quad \text{for some } \zeta \text{ between } \alpha_1 \text{ and } \alpha_2 \text{ with } k < 1.$$

Hence,

$$|\alpha_1 - \alpha_2| = |g(\alpha_1) - g(\alpha_2)| \le k|\alpha_1 - \alpha_2| < |\alpha_1 - \alpha_2|,$$

a contradiction unless $\alpha_1 = \alpha_2$. Therefore, the fixed point is unique.

(c) **Convergence:** If α is a fixed point,

$$g(\alpha) = \alpha,$$

and $x_n = g(x_{n-1})$, we apply the Mean Value Theorem 1.5.5 to g between x_{n-1} and α,

$$\begin{aligned} |x_n - \alpha| &= |g(x_{n-1}) - g(\alpha)| \\ &= |g'(\zeta)|\,|x_{n-1} - \alpha| \quad (\zeta \text{ between } x_{n-1} \text{ and } \alpha) \\ &\le k\,|x_{n-1} - \alpha|, \end{aligned}$$

So by induction, we get,

$$|x_n - \alpha| \le k^n |x_0 - \alpha|.$$

Since $0 < k < 1$, $k^n \to 0$, hence $x_n \to \alpha$.

□

Error Estimates

Corollary 2.6.5. *If g satisfies the fixed-point theorem assumptions, then,*

1. *Absolute error bound:*
$$|x_n - \alpha| \le k^n \max\{x_0 - a,\ b - x_0\}.$$
2. *Alternate bound (for $n \ge 1$):*
$$|x_n - \alpha| \le \frac{k^n}{1-k}|x_1 - x_0|.$$

Proof: Alternate Bound. For $n = 1$:

$$|x_0 - \alpha| \le |x_1 - \alpha| + |x_0 - x_1| \le k|x_0 - \alpha| + |x_0 - x_1|,$$

which rearranges to

$$|x_0 - \alpha| \le \frac{1}{1-k}|x_0 - x_1|.$$

Combining with the induction result from the proof of convergence, we get the desired result. □

Number of Iterations Required

If a number is rounded to r decimal places, the absolute error satisfies,

$$\varepsilon_a \le \frac{1}{2}\,10^{-r}.$$

We want $|x_n - \alpha| < \frac{1}{2}\,10^{-r}$. From the convergence value of fixed-point iteration, we get,

$$k^n|x_0 - \alpha| \le \frac{1}{2}\,10^{-r}.$$

Equivalently,

$$n \ge \frac{\ln\left[\frac{(1/2)10^{-r}}{|x_0-\alpha|}\right]}{\ln(k)}.$$

Since $\ln(k) < 0$ (for $0 < k < 1$), take the ceiling for the minimal integer n.

Example 2.6.6. *Revisit $x^3 - x - 1 = 0$. Choose $g(x)$ that satisfies the fixed-point theorem, determine k, and estimate n for $r = 4$.* ***[Do Yourself !]***

Graphical Interpretation of $g'(x)$: Convergence behaviour is tied to $m = g'(\alpha)$ (can be seen in Fig. 2.4),

1. Case I: $0 < g'(\alpha) < 1$ — monotone, convergent.
2. Case II: $-1 < g'(\alpha) < 0$ — oscillating, convergent.
3. Case III: $g'(\alpha) > 1$ — divergent, monotone.
4. Case IV: $g'(\alpha) < -1$ — divergent, oscillatory.

Order of Convergence

Theorem 2.6.7. *Let α be a fixed point of g.*

1. *If $g'(\alpha) \ne 0$ and $|g'(\alpha)| < 1$, convergence is* ***linear*** *($p = 1$).*

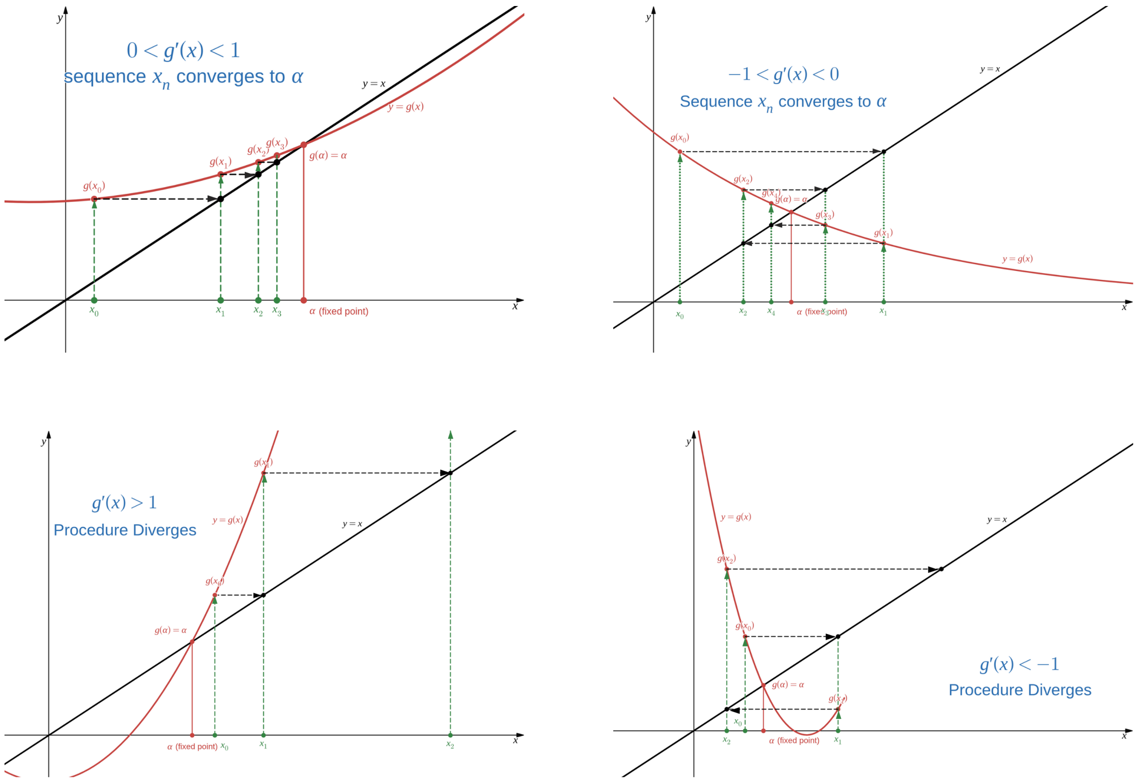


Figure 2.4: Graphical Interpretation of Fixed-Point Iterations

2. *If $g'(\alpha) = 0$ but $g''(\alpha) \neq 0$, convergence is* ***quadratic*** *($p = 2$); specifically,*

$$|x_{n+1} - \alpha| \approx \frac{|g''(\alpha)|}{2}|x_n - \alpha|^2.$$

3. *If $g'(\alpha) = g''(\alpha) = 0$ but $g'''(\alpha) \neq 0$, convergence is* ***cubic*** *($p = 3$).*

Idea for $p = 2$. Expand Taylor Series for function $g(x)$ about $x = \alpha$,

$$g(x) = g(\alpha) + g'(\alpha)(x - \alpha) + \frac{1}{2}g''(\xi)(x - \alpha)^2.$$

If $g'(\alpha) = 0$, $g(\alpha) = \alpha$, substituting x_n gives the quadratic error recursion. □

Some Observations on Fixed-Point Iterations:

1. Can converge very quickly if $|g'(\alpha)| \ll 1$.
2. For $g'(\alpha) = 0$, $g''(\alpha) \neq 0$, order $p = 2$.
3. No guaranteed convergence if g fails the conditions $a \leq g(x) \leq b$, $|g'(x)| < 1$.
4. Poor choice of g can lead to divergence.

General Order Result: If $g^{(m)}(\alpha) \neq 0$ but $g'(\alpha) = \cdots = g^{(m-1)}(\alpha) = 0$, and $g^{(m)}$ is continuous and bounded near α, then the order of convergence is m. **[Try to prove yourself !]**

Understanding and Implementation of Fixed-Point Method

Given an equation $f(x) = 0$, it can often be rearranged into the form $x = g(x)$, such that finding a root of f is equivalent to finding a fixed point of g.

We want to find a fixed point α such that $g(\alpha) = \alpha$, then the fixed point iterations as defined follow, $x_{n+1} = g(x_n)$,

Conditions for Fixed-Point iterations

There exists a fixed point $p \in [a, b]$ such that $g(\alpha) = \alpha$ for,

1. Existence Condition I: $g(x)$ must continuous on $[a, b]$.
2. Existence Condition II: $g(x)$ must be bounded on $[a, b]$ by $[a, b]$.
3. Uniqueness Condition: $|g'(x)| < 1 \quad \forall x \in [a, b]$

Visualization. Existence Condition II shows that the fixed-point iteration converges in the box formed by $[a, b] \times [a, b]$, and hence, we mainly deal with the domain $[a, b]$ for the fixed-point iteration method.

Guarantee of Convergence

Let us assume a starting point $x_0 \in [a, b]$, then the iteration proceeds as, $x_1 = g(x_0), x_2 = g(x_1), \ldots, x_n = g(x_{n-1})$. Now this will take a form,

$$|x_n - \alpha| = |g(x_n) - g(\alpha)|$$

Applying the Mean Value Theorem 1.5.5, we get,

$$|\varepsilon_n| = |g'(\gamma)|\,|\varepsilon_{n-1}| \qquad \text{where, } \gamma \in [x_{n-1}, \alpha]$$

Now, from the uniqueness condition for fixed-point iterations, we get $|\varepsilon_n| < |\varepsilon_{n-1}|$. Hence, convergence is guaranteed.

NOTE: Using the uniqueness condition and convergence condition, we get $\lim_{n\to\infty} \varepsilon_n = 0$.

Generalization of Order of Convergence

On similar lines of the above convergence proof, we get, $\varepsilon_{n+1} = |g(x_n) - g(\alpha)|$. Now expanding $g(x_n)$ in Taylor series about $x = \alpha$ we get,

$$g(x_n) = g(\alpha) + g'(\alpha)(x_n - \alpha) + g''(\alpha)\frac{(x_n - \alpha)^2}{2!} + \ldots$$
$$\varepsilon_{n+1} = g'(\alpha)\varepsilon_n + g''(\alpha)\frac{\varepsilon_n^2}{2!} + g'''(\alpha)\frac{\varepsilon_n^3}{3!} + \ldots$$

Hence, using this equation we can say that, if $g^{(m)}(\alpha) \neq 0$ but $g'(\alpha) = \cdots = g^{(m-1)}(\alpha) = 0$, and $g^{(m)}$ is continuous and bounded near α, then the order of convergence is m.

2.6.2. Convergence of Newton's Method using Fixed-Point Iterates

Given that Newton's method is of the form,

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

Now we can see this as the fixed-point iteration of the form,

$$x = g(x) \qquad \text{where, } g(x) = x - \frac{f(x)}{f'(x)}$$

Let us assume that this satisfies the existence conditions and the uniqueness condition. Now, we will find out $g'(\alpha)$, $g''(\alpha)$, $g'''(\alpha)$, $\ldots$ until when the terms becomes non-zero.

$$g'(\alpha) = \frac{f(\alpha)f''(\alpha)}{(f'(\alpha))^2} = 0, \qquad g''(\alpha) = \frac{-f'''(\alpha)}{6f''(\alpha)} \neq 0$$

Hence, from the generalized order of convergence for the fixed-point method, the order of convergence of Newton's method is **quadratic** ($p = 2$).

NOTE I: In case of multiple roots, $f(x) = (x - \alpha)^m q(x)$ where, $\lim_{x \to \alpha} q(x) \neq 0$ and $m \geq 2$ is the multiplicity of root α. Applying L'Hopital's rule to $g'(\alpha)$ we get,

$$g'(\alpha) = \frac{m-1}{m}$$

Hence, the Newton method becomes **linearly convergent** ($p = 1$) in the case of multiple roots.

NOTE II: It is clear that when $m \to \infty$, the convergence of the Newton method is not guaranteed.

$$\lim_{m \to \infty} g'(\alpha) = 1$$

2.7. Muller's Method

Muller's method is an extension of the secant method that uses a quadratic polynomial (parabola) interpolation to approximate roots. Unlike the secant method, which requires two initial guesses, Muller's method starts with three initial approximations x_0, x_1, x_2.

The method finds the next approximation x_3 by determining the intersection of the parabola passing through the points $(x_0, f(x_0)), (x_1, f(x_1))$, and $(x_2, f(x_2))$ with the x-axis. Muller's method can find both real and complex roots.

Derivation of the method

Define the quadratic polynomial $P(x) = a(x - x_2)^2 + b(x - x_2) + c$ passing through the three given points. From interpolation conditions,

$$\begin{cases} f(x_0) = a(x_0 - x_2)^2 + b(x_0 - x_2) + c, \\ f(x_1) = a(x_1 - x_2)^2 + b(x_1 - x_2) + c, \\ f(x_2) = c. \end{cases}$$

Solving the above equations, the coefficients we get,

$$a = \frac{(x_1 - x_2)[f(x_0) - f(x_2)] - (x_0 - x_2)[f(x_1) - f(x_2)]}{(x_0 - x_2)(x_1 - x_2)(x_0 - x_1)},$$

$$b = \frac{(x_0 - x_2)^2[f(x_1) - f(x_2)] - (x_1 - x_2)^2[f(x_0) - f(x_2)]}{(x_0 - x_2)(x_1 - x_2)(x_0 - x_1)},$$

$$c = f(x_2).$$

We find roots of $P(x) = 0$, i.e.,

$$x - x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Rationalizing to avoid subtractive cancellation,

$$x_3 = x_2 - \frac{2c}{b \pm \text{sgn}(b)\sqrt{b^2 - 4ac}}, \qquad \text{where, } \text{sgn}(b) = \begin{cases} +1 & b > 0, \\ -1 & b < 0. \end{cases}$$

The sign is chosen to maximize the denominator magnitude for numerical stability. Muller's method converges with order of 1.84 **[Try to prove this yourself !]**, which is better than the Secant method due to taking 3 points into consideration than 2 points in the Secant method.

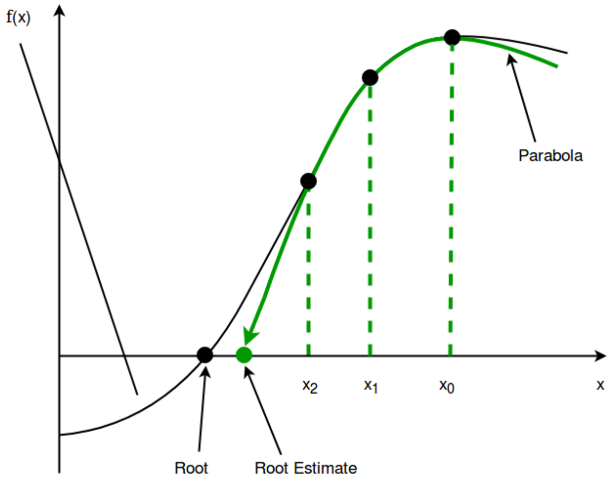


Figure 2.5: Graphical representation for Muller's Method

Complex Root Approximation

Muller's method naturally extends to complex roots since the square root can become complex even if initial approximations are real. For $b \in \mathbb{C}$, define $\operatorname{sgn}(b) = \operatorname{sgn}(\Re(b))$, noting this may not cover purely imaginary b. Nonetheless, the method effectively approximates complex roots.

Example 2.7.1. *Find a root of $f(x) = 3x + \sin x - e^x$ between 0 and 1, correct to 3 decimal places.*

Take $x_0 = 0, x_1 = 1, x_2 = 0.5$, and evaluate $f(x)$ at these points, $f(x_0) = -1, f(x_1) = 1.123189, f(x_2) = 0.330704$. Now, compute coefficients a, b, c and then calculate subsequent approximations $x_3 = 0.354914, x_4 = 0.360465, x_5 = 0.3604217$. Then, root is approximated as 0.36042 to 3 decimal places.

Advantages:

1. Faster convergence than Bisection, Secant, and Regula-Falsi methods (order approximately 1.84).
2. Initial approximations need not bracket the root.
3. No need to evaluate derivatives.
4. Can approximate complex roots from real starting points.

Disadvantages:

1. Requires three good initial approximations for best performance.
2. May diverge for poor initial guesses.
3. Close or nearly collinear points cause accuracy and stability problems.
4. Implementation is more complex compared to other root-finding methods.

2.8. Summary for Root-finding Methods and Practice Questions

Practice Questions

1. Find the minimum number of iterations required for the Bisection method to converge to an error of $O(0.125)$ and the initial range is $[0, 2]$. Also, write the condition when the root is guaranteed to be found.

Method	Root Type	Order of Convergence	Guarantee of Convergence
Bisection	simple	$\alpha = 1$	YES
	multiple (odd)	$\alpha = 1$	YES
	multiple (even)	Not Available	Not Available
False Position	simple	$\alpha = 1$	YES
	multiple (odd)	$\alpha = 1$	YES
	multiple (even)	Not Available	Not Available
Secant	simple	$\alpha = 1.62$	NO
	multiple	$\alpha = 1$	NO
Newton's	simple	$\alpha = 2$	NO
	multiple	$\alpha = 1$	NO
Modified Newton's	simple	$\alpha = 2$	NO
	multiple	$\alpha = 2$	NO

Table 2.1: Summary of Root Finding Methods and Their Asymptotic Convergence Rates

2. Prove that the Bisection method always converges to a root in $[a, b]$ provided that $f(x)$ is continuous on $[a, b]$ and $f(a)f(b) < 0$. What would happen if $f(x)$ is not continuous on $[a, b]$? Give an example.

3. Both the Bisection and Regula-Falsi methods have order of convergence $p = 1$ (linear). Explain conceptually, with reference to their algorithms, why the Regula-Falsi method will often converge to a required accuracy in fewer steps than the Bisection method for most functions. Show their respective convergence graphs as well.

4. If the multiplicity of the root of the function $f(x)$ is m, then find out the order of convergence of the following 2 methods,

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

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

5. Consider the function $f(x) = \tan x - x$ near $x = 0$. Discuss why the Newton-Raphson method fails for an initial guess $x_0 = 0$. What alternative would you suggest and why?

6. Show that the order of convergence of the Newton-Raphson method for a simple root is quadratic, but for a multiple root of multiplicity m, it is linear. Derive the relevant error recurrence relation for the multiple root case.

7. Compare the stopping criteria for Newton-Raphson and the Secant methods. Discuss the advantages and disadvantages of using a fixed number of iterations versus a tolerance on $|f(x_n)|$ or $|x_n - x_{n-1}|$.

8. Show graphically the difference between the Secant method and the Regula-Falsi method.

9. Let us assume $f(x) = x^2 + 2x + 1$, then evaluate a fixed-point iteration method for finding the roots of the function. If possible, then find the rate of convergence for the method.

10. Consider the fixed-point iteration $x_{n+1} = g(x_n)$ where $g(\alpha) = \alpha$ and $g'(\alpha) = 0$. Derive the expression for the quadratic order of convergence near the fixed point using Taylor's theorem.

11. Explain why the Newton-Raphson method can fail to converge if the initial guess x_0 is not sufficiently close to the root α. Include an example function or scenario demonstrating this.

12. Using the function $f(x) = (x - 1)^3$, apply the Newton-Raphson iteration formula and analyze the convergence behaviour. What modifications would you suggest for improving convergence?

13. What conditions are necessary for the convergence of a fixed-point iteration involving multiple variables (and equations)? What conditions are needed for such a process to be quadratically convergent?

14. Prove the order of convergence of Muller's method is approximately 1.84. Explain why it can converge faster than the Secant method.

15. Given the function $f(x) = x^3 - 7x^2 + 14x - 6$, apply two iterations of Muller's method starting from initial points $x_0 = 1$, $x_1 = 2$, and $x_2 = 2.5$. Analyze the error reduction and discuss how the choice of initial points affects convergence.

Programming Questions

1. Find the root of $f(x) = x^3 - x - 1$ in with tolerance of $O(0.001)$ using Bisection method.
2. Find the root of $f(x) = x^2 - 4$ in with tolerance of $O(0.001)$ using Regula-Falsi method.
3. Find the root of $f(x) = x^2 - 10x + 25$ using Newton-Raphson method and modified Newton-Raphson method starting with $x_0 = 4.5$.
4. Find root of $f(x) = x^3 - 2x - 5$ using Secant method with initial guesses $x_0 = 2, x_1 = 3$.
5. Find root of $f(x) = x - \cos x$ using fixed point method starting with $x_0 = 1$.
6. Find root of $f(x) = x^3 - 1$ using Muller's method with initial guesses $x_0 = 0, x_1 = 0.5, x_2 = 1$.

Interpolation and Polynomial Approximation

3.1. Interpolation

Example: Consider the list of students taking admission in different years:

Year	2006	2009	2010	2012	2015	2016	2018
Students	600	800	950	1000	1050	1100	1200

One may ask whether these data could be used to provide a reasonable estimate of students, say in 2008 or 2011 or 2017. Predictions of these types can be obtained using a function fitting the given data. This process is called **interpolation**.

If data (x_i, y_i), $i = 0, 1, 2, \ldots, n$ are available from an experiment or otherwise, such that y_i depends on x_i, then we want to find the nature of the relationship of y on x.

The goal is to approximate the value of y at some value of x not listed among the x_i, or determine a function that in some sense approximates the data.

The points where the values of polynomial and the function coincide are called **interpolating points** or **nodes** or **tabular points**. The polynomial is known as the **interpolating polynomial**.

The function $f(x)$ is generally replaced by a polynomial $p_n(x)$:

$$p_n(x) = a_0 + a_1 x + \cdots + a_n x^n$$

3.2. Weierstrass Approximation Theorem

The following theorem is the basis for polynomial approximation:

Theorem 3.2.1 (Weierstrass Approximation Theorem)**.** *Suppose that $f \in C[a, b]$. For each $\epsilon > 0$ there exists a polynomial $P(x)$, such that*

$$\|f(x) - P(x)\| < \epsilon, \text{ for all } x \in [a, b]$$

The theorem says nothing about finding the polynomial or its order.

The Weierstrass Approximation Theorem guarantees that we (maybe with substantial work) can find a polynomial that fits into the tube around the function f, no matter how thin we make the tube. Another important reason for considering the class of polynomials in the approximation of functions is that the derivative and indefinite integral of a polynomial are easy to determine and are also polynomials.

3.3. Existence and Uniqueness of Interpolating Polynomial

Definition 3.3.1. Let $x_0, x_1, \ldots, x_n$ be $n+1$ distinct points in the interval $[a, b]$. Then $p_n(x)$ is an interpolating polynomial to $f(x)$ if

$$p_n(x_i) = f(x_i) \text{ for } i = 0, 1, 2, \ldots, n$$

or

$$p_n(x_i) = f(x_i), \quad p_n'(x_i) = f'(x_i) \text{ for } i = 0, 1, 2, \ldots, n$$

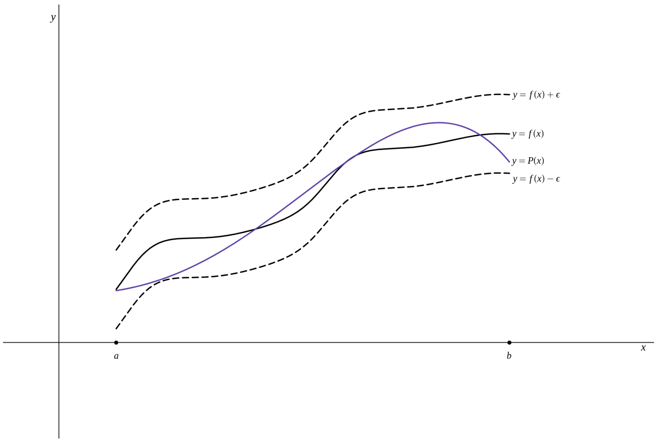


Figure 3.1: Graphical representation for Weierstrass Theorem

The derivative conditions may be replaced by more general conditions involving higher-order derivatives.

Note: The Taylor expansion works very hard to be accurate in the neighborhood of one point. But we want to fit data at many points (in an extended interval).

3.3.1. Existence and Uniqueness

Suppose we have $n+1$ distinct points $x_0 < x_1 < \cdots < x_n$. If $p_n(x)$ is a polynomial interpolating $f(x)$ at a set of $n+1$ points, then:

$$p_n(x_i) = f(x_i) \text{ for } i = 0, 1, 2, \ldots, n$$

We can write:

$$\begin{aligned} a_0 + a_1 x_0 + \cdots + a_n x_0^n &= f(x_0) = f_0 \\ a_0 + a_1 x_1 + \cdots + a_n x_1^n &= f(x_1) = f_1 \\ &\vdots \\ a_0 + a_1 x_n + \cdots + a_n x_n^n &= f(x_n) = f_n \end{aligned}$$

This is a system of $n+1$ linear equations in $n+1$ unknowns: $a_0, a_1, \ldots, a_n$.

This system will have a unique solution if the determinant

$$\Delta = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix} \neq 0$$

Indeed, the value of the determinant is not zero:

$$\Delta = \prod_{0 \leq j < i \leq n} (x_i - x_j)$$

Therefore, a unique interpolating polynomial exists.

Example 3.3.2. *Find the interpolating polynomial for the following data:* $f(-1) = 0,\ f(0) = 1,\ f(1) = 2$
Solution*:Consider the interpolating polynomial*

$$p(x) = a_0 + a_1 x + a_2 x^2$$

$$a_0 = 1, a_1 = 1, a_2 = 0$$

Thus

$$p(x) = 1 + x$$

3.4. Lagrange Interpolating Polynomial

Introduction: Polynomial interpolation is a method of estimating values between known data points. The Lagrange interpolation provides a direct formula for constructing a polynomial that passes exactly through given points.

We will define a function that passes through the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$. First, let's define:

$$l_0(x) = \frac{x - x_1}{x_0 - x_1} \quad \text{and} \quad l_1(x) = \frac{x - x_0}{x_1 - x_0}$$

Key Properties: Notice that $l_0(x_0) = 1$, $l_0(x_1) = 0$, $l_1(x_0) = 0$, and $l_1(x_1) = 1$. These are called basis functions because they form a foundation for our interpolation.

Then define the interpolating polynomial:

$$p(x) = l_0(x)f(x_0) + l_1(x)f(x_1)$$

We can verify that $p(x_0) = f(x_0)$ and $p(x_1) = f(x_1)$.

$p(x)$ is the unique linear polynomial passing through the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$.

Example 3.4.1. *(Linear Interpolation): Find the linear interpolating polynomial for points* $(2, 5)$ *and* $(6, 13)$.

Solution: *Here* $x_0 = 2$, $f(x_0) = 5$, $x_1 = 6$, $f(x_1) = 13$.

$$\begin{aligned} l_0(x) &= \frac{x-6}{2-6} = \frac{x-6}{-4} = \frac{6-x}{4} \\ l_1(x) &= \frac{x-2}{6-2} = \frac{x-2}{4} \end{aligned}$$

$$\begin{aligned} p(x) &= \frac{6-x}{4} \cdot 5 + \frac{x-2}{4} \cdot 13 \\ &= \frac{5(6-x) + 13(x-2)}{4} \\ &= \frac{30 - 5x + 13x - 26}{4} \\ &= \frac{4 + 8x}{4} = 1 + 2x \end{aligned}$$

Verification: $p(2) = 1 + 2(2) = 5$, $p(6) = 1 + 2(6) = 13$

3.4.1. General Lagrange Interpolating Polynomial

Let $x_0, x_1, \ldots, x_n$ be $n + 1$ distinct points in $[a, b]$.

Consider an interpolating polynomial $p(x)$ of degree $\leq n$:

$$p(x) = l_0(x)f(x_0) + l_1(x)f(x_1) + \cdots + l_n(x)f(x_n)$$

where $l_i(x)$, $0 \leq i \leq n$ are polynomials of degree n called **Lagrange basis polynomials**.

Interpolation Condition: The polynomial $p(x)$ will satisfy the interpolating conditions $p(x_i) = f(x_i)$ if and only if:

$$l_i(x_j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

This condition is called the **Kronecker delta property**.

The polynomial $l_i(x)$ that satisfies this condition can be written as:

$$l_i(x) = \frac{(x - x_0)}{(x_i - x_0)} \cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \cdots \frac{(x - x_n)}{(x_i - x_n)} \quad \text{where} 0 \leq i \leq n.$$

Compact Notation: We can write this more compactly as:

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

The functions $l_i(x)$ are called **Lagrange coefficients** and the polynomial $p(x)$ is called the **Lagrange interpolating polynomial**.

Theorem 3.4.2. *If $x_0, x_1, \ldots, x_n$ are $n + 1$ distinct numbers and f is a function whose values are given at these numbers, then a unique polynomial $p(x)$ of degree at most n exists with $f(x_k) = p(x_k)$, for each $k = 0, 1, \ldots, n$:*

$$p(x) = \sum_{k=0}^{n} l_k(x) f(x_k)$$

where

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j}$$

Particular Cases of Lagrange Interpolation

For $n = 1$ (linear interpolation):

$$p_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

For $n = 2$ (quadratic interpolation):

$$p_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2)$$

Example 3.4.3. *Find the interpolating polynomial for the data: $f(0) = 1$, $f(-1) = 2$, $f(1) = 3$.*

Solution: *The degree of the interpolating polynomial is at most two.*
$x_0 = 0$, $x_1 = -1$, $x_2 = 1$
Computing the Lagrange basis polynomials:

$$\begin{aligned}
l_0(x) &= \frac{(x - (-1))(x - 1)}{(0 - (-1))(0 - 1)} = \frac{(x + 1)(x - 1)}{(1)(-1)} = \frac{-(x^2 - 1)}{1} = 1 - x^2 \\
l_1(x) &= \frac{(x - 0)(x - 1)}{(-1 - 0)(-1 - 1)} = \frac{x(x - 1)}{(-1)(-2)} = \frac{x(x - 1)}{2} \\
l_2(x) &= \frac{(x - 0)(x - (-1))}{(1 - 0)(1 - (-1))} = \frac{x(x + 1)}{(1)(2)} = \frac{x(x + 1)}{2}
\end{aligned}$$

Thus:

$$\begin{aligned}
p_2(x) &= (1-x^2)\cdot 1 + \frac{x(x-1)}{2}\cdot 2 + \frac{x(x+1)}{2}\cdot 3 \\
&= 1 - x^2 + x(x-1) + \frac{3x(x+1)}{2} \\
&= 1 - x^2 + x^2 - x + \frac{3x^2+3x}{2} \\
&= 1 - x + \frac{3x^2+3x}{2} \\
&= 1 - x + \frac{3x}{2} + \frac{3x^2}{2} \\
&= 1 + \frac{x}{2} + \frac{3x^2}{2} = \frac{2+x+3x^2}{2}
\end{aligned}$$

Verification: $p_2(0) = 1$, $p_2(-1) = \frac{2-1+3}{2} = 2$, $p_2(1) = \frac{2+1+3}{2} = 3$

Example 3.4.4. *Find the quadratic interpolating polynomial for the points* $(1,2)$, $(3,8)$, *and* $(4,15)$.

Solution: *Here* $x_0 = 1$, $f(x_0) = 2$; $x_1 = 3$, $f(x_1) = 8$; $x_2 = 4$, $f(x_2) = 15$.
Computing the Lagrange basis polynomials:

$$\begin{aligned}
l_0(x) &= \frac{(x-3)(x-4)}{(1-3)(1-4)} = \frac{(x-3)(x-4)}{(-2)(-3)} = \frac{(x-3)(x-4)}{6} \\
l_1(x) &= \frac{(x-1)(x-4)}{(3-1)(3-4)} = \frac{(x-1)(x-4)}{(2)(-1)} = -\frac{(x-1)(x-4)}{2} \\
l_2(x) &= \frac{(x-1)(x-3)}{(4-1)(4-3)} = \frac{(x-1)(x-3)}{(3)(1)} = \frac{(x-1)(x-3)}{3}
\end{aligned}$$

Thus:

$$\begin{aligned}
p_2(x) &= 2\cdot\frac{(x-3)(x-4)}{6} + 8\cdot\left(-\frac{(x-1)(x-4)}{2}\right) + 15\cdot\frac{(x-1)(x-3)}{3} \\
&= \frac{(x-3)(x-4)}{3} - 4(x-1)(x-4) + 5(x-1)(x-3)
\end{aligned}$$

Expanding each term:

$$\begin{aligned}
(x-3)(x-4) &= x^2 - 7x + 12 \\
(x-1)(x-4) &= x^2 - 5x + 4 \\
(x-1)(x-3) &= x^2 - 4x + 3
\end{aligned}$$

Substituting:

$$\begin{aligned}
p_2(x) &= \frac{x^2-7x+12}{3} - 4(x^2-5x+4) + 5(x^2-4x+3) \\
&= \frac{x^2-7x+12}{3} - 4x^2 + 20x - 16 + 5x^2 - 20x + 15 \\
&= \frac{x^2-7x+12}{3} + x^2 - 1 \\
&= \frac{x^2-7x+12+3x^2-3}{3} \\
&= \frac{4x^2-7x+9}{3}
\end{aligned}$$

Verification: $p_2(1) = \frac{4-7+9}{3} = 2$, $p_2(3) = \frac{36-21+9}{3} = 8$, $p_2(4) = \frac{64-28+9}{3} = 15$

Example 3.4.5. *Find the cubic interpolating polynomial for the data points:* $(0,1)$, $(1,4)$, $(2,9)$, $(3,16)$.

Solution: *We have $x_0 = 0,\ f(x_0) = 1$; $x_1 = 1,\ f(x_1) = 4$; $x_2 = 2,\ f(x_2) = 9$; $x_3 = 3,\ f(x_3) = 16$.*
The Lagrange basis polynomials are:

$$l_0(x) = \frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)} = \frac{(x-1)(x-2)(x-3)}{-6}$$
$$l_1(x) = \frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)} = \frac{x(x-2)(x-3)}{(1)(-1)(-2)} = \frac{x(x-2)(x-3)}{2}$$
$$l_2(x) = \frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)} = \frac{x(x-1)(x-3)}{(2)(1)(-1)} = -\frac{x(x-1)(x-3)}{2}$$
$$l_3(x) = \frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)} = \frac{x(x-1)(x-2)}{6}$$

The cubic interpolating polynomial is:

$$p_3(x) = 1 \cdot l_0(x) + 4 \cdot l_1(x) + 9 \cdot l_2(x) + 16 \cdot l_3(x)$$

Expanding and simplifying (calculations omitted for brevity):

$$p_3(x) = x^2 + 2x + 1 = (x+1)^2$$

Note: This simplifies to a quadratic because the data points $(0,1)$, $(1,4)$, $(2,9)$, $(3,16)$ correspond to the function $f(x) = (x+1)^2$, which is indeed quadratic.
Verification: $p_3(0) = 1$, $p_3(1) = 4$, $p_3(2) = 9$, $p_3(3) = 16$

3.4.2. Interpolation Error

Motivation: When we use polynomial interpolation to approximate a function, we need to understand how accurate our approximation is. The error analysis tells us how the true function $f(x)$ differs from our interpolating polynomial $p(x)$.

Theorem 3.4.6 (Rolle's Theorem)**.** *If $f(x)$ is a real valued continuous function on $[a, b]$ and $f'(x)$ exists on (a, b). Also if $f(a) = f(b)$, then there exists at least one number $c \in (a, b)$ such that $f'(c) = 0$.*

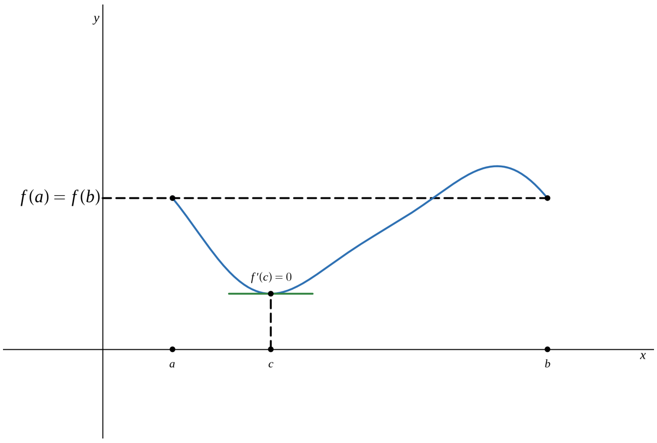


Figure 3.2: Graphical representation for Rolle's Theorem

Theorem 3.4.7 (Generalized Rolle's Theorem)**.** *Suppose $f \in C[a, b]$ is n-times differentiable on (a, b). If $f(x)$ is zero at the $n + 1$ distinct points $x_0, x_1, \ldots, x_n$ in $[a, b]$, then there exists a number $c \in (a, b)$ such that $f^{(n)}(c) = 0$.*

Theorem 3.4.8 (Interpolation Error)**.** *Let $f(x) \in C^{n+1}[a, b]$ and $x_i \in [a, b]$ for $i = 0, 1, \ldots, n$. Consider $p_n(x)$ as a polynomial interpolating $f(x)$ at x_i, $i = 0, 1, \ldots, n$. Then:*

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1) \cdots (x - x_n)$$

where ξ is an unknown point in the interval (a, b).

Consider $M = \max_{a \le x \le b} |f^{(n+1)}(x)|$, *then:*

$$|E(x)| = |f(x) - p(x)| \le \frac{M}{(n+1)!}|(x - x_0)(x - x_1)\cdots(x - x_n)|$$

Further, if $N = \max_{a \le x \le b} |(x - x_0)(x - x_1)\cdots(x - x_n)|$ *then:*

$$|E(x)| \le \frac{MN}{(n+1)!}$$

Interpretation: The error depends on three factors:

- The $(n+1)$-th derivative of $f(x)$ (how "curved" the function is)
- The distance from the interpolation points
- The factorial $(n+1)!$ (which grows rapidly, suggesting diminishing returns for high-degree polynomials)

Example 3.4.9. *For linear interpolation with nodes* x_0, x_1*, the error bound is*

$$|E(x)| \le \frac{M}{8}(x_1 - x_0)^2, \qquad M = \max_{x_0 \le x \le x_1} |f''(x)|.$$

Proof:
The interpolation error formula is

$$|E(x)| \le \frac{M}{2!}\,|(x - x_0)(x - x_1)|.$$

Define

$$g(x) = (x - x_0)(x_1 - x),$$

so that $|(x - x_0)(x - x_1)| = g(x)$ *on* $[x_0, x_1]$. *Differentiate:*

$$g'(x) = (x_1 - x) - (x - x_0) = x_1 + x_0 - 2x.$$

Setting $g'(x) = 0$ *gives*

$$x = \frac{x_0 + x_1}{2}.$$

At this midpoint,

$$g\left(\tfrac{x_0+x_1}{2}\right) = \left(\tfrac{x_1-x_0}{2}\right)^2 = \frac{(x_1 - x_0)^2}{4}.$$

Hence,

$$|E(x)| \le \frac{M}{2} \cdot \frac{(x_1 - x_0)^2}{4} = \frac{M}{8}(x_1 - x_0)^2.$$

Example 3.4.10. *Determine the step size* h *that can be used in the tabulation of* $f(x) = \sin x$ *in the interval* $[1, 3]$ *so that the linear interpolation will be correct to four decimal places after rounding. Given* $f(x) = \sin x$*, we compute:*

$$f'(x) = \cos x$$

$$f''(x) = -\sin x$$

Find $\max |f''(x)|$ *on* $[1, 3]$. *We need to find:*

$$\max_{x\in[1,3]} |f''(x)| = \max_{x\in[1,3]} |-\sin x| = \max_{x\in[1,3]} |\sin x|$$

Since $\sin x$ *achieves its maximum value of 1 at* $x = \frac{\pi}{2} \approx 1.5708$*, and* $\frac{\pi}{2} \in [1, 3]$*, we have:*

$$\max_{x\in[1,3]} |f''(x)| = 1$$

For 4-decimal-place accuracy after rounding, the error must be less than:

$$\varepsilon = 0.5 \times 10^{-4} = 0.00005$$

Apply linear interpolation error bound. That is the error bound for linear interpolation is:

$$|E(x)| \leq \frac{h^2}{8} \max_{x\in[a,b]} |f''(x)|$$

Substituting our values:

$$|E(x)| \leq \frac{h^2}{8} \cdot 1 = \frac{h^2}{8}$$

Solve for h*, That us setting up the inequality for the required accuracy:*

$$\frac{h^2}{8} \leq 0.00005$$

Solving for h*:*

$$h^2 \leq 8 \times 0.00005 = 0.0004$$
$$h \leq \sqrt{0.0004} = 0.02$$

The step size must satisfy:

$$\boxed{h \leq 0.02}$$

This means we can use any step size up to and including $h = 0.02$ *to guarantee that linear interpolation of* $\sin x$ *on* $[1, 3]$ *will be accurate to four decimal places after rounding.*

Example 3.4.11. *Estimate the error when approximating* $f(x) = e^x$ *using linear interpolation between* $x_0 = 0$ *and* $x_1 = 0.5$.
Solution: *For* $f(x) = e^x$*, the second derivative is*

$$f''(x) = e^x.$$

On the interval $[0, 0.5]$*, we have*

$$M = \max_{0\leq x\leq 0.5} |f''(x)| = e^{0.5} \approx 1.649.$$

The error bound is

$$|E(x)| \leq \frac{M}{8}(x_1 - x_0)^2.$$

Substituting values:

$$|E(x)| \leq \frac{1.649}{8}(0.5 - 0)^2 = \frac{1.649}{8} \cdot 0.25 = \frac{1.649 \times 0.25}{8}.$$

Thus,

$$|E(x)| \approx 0.0515.$$

The maximum error occurs at the midpoint $x = \frac{x_0+x_1}{2} = 0.25$.

Practical Considerations:

- For equally spaced points, the error is minimized when evaluating near the center of the interval
- High-degree interpolation can lead to oscillations (Runge phenomenon)
- For better accuracy with many points, consider piecewise interpolation

3.5. Newton's Divided Difference

Motivation: In Lagrangian polynomial formulation, if a tabular point is added to the data then all Lagrangian polynomials are to be constructed fresh. Therefore, another form of interpolating polynomial is needed to meet this requirement. Since we already proved that the interpolating polynomial is unique, only the form will be different.

The Newton's divided difference method has several advantages:

- Easy to add new data points without recalculating everything
- Provides insight into the smoothness of the data
- Allows for systematic error analysis
- Computationally efficient for incremental calculations

3.5.1. Newton's Divided Difference Interpolating Formula

Nodes: $x_0, x_1, \dots, x_n$

Functional values: $f(x_0), f(x_1), \dots, f(x_n)$

Definition of Divided Differences:

The divided difference of **zeroth order** for argument x_0 is denoted by $f[x_0]$, defined by:

$$f[x_0] = f_0$$

The divided difference of **first order** for arguments x_0, x_1 is denoted by $f[x_0, x_1]$, defined by:

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$$

Geometric Interpretation: $f[x_0, x_1]$ represents the slope of the line connecting points (x_0, f_0) and (x_1, f_1).

The divided difference of **second order** for arguments x_0, x_1, x_2 is denoted by $f[x_0, x_1, x_2]$, defined by:

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

General Definition: The nth order divided difference of the function f at $n + 1$ nodes $x_0, x_1, \dots, x_n$ is:

$$f[x_0, x_1, \dots, x_n] = \frac{f[x_1, x_2, \dots, x_n] - f[x_0, x_1, \dots, x_{n-1}]}{x_n - x_0}$$

Important Properties of Divided Differences:

- Divided differences are symmetric: $f[x_0, x_1, \dots, x_n] = f[x_{\pi(0)}, x_{\pi(1)}, \dots, x_{\pi(n)}]$ for any permutation π
- $f[x_0, x_1, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}$ for some ξ in the smallest interval containing all x_i

Interpolating polynomial in Newton form:

$$\begin{aligned} p_n(x) = f(x_0) &+ (x - x_0)f[x_0, x_1] \\ &+ (x - x_0)(x - x_1)f[x_0, x_1, x_2] \\ &+ (x - x_0)(x - x_1)(x - x_2)f[x_0, x_1, x_2, x_3] \\ &+ \cdots + (x - x_0)(x - x_1)\cdots(x - x_{n-1})f[x_0, x_1, \dots, x_n] \end{aligned}$$

This is called **Newton's divided difference formula** for interpolating polynomial.

Compact Form:

$$p_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k] \prod_{j=0}^{k-1} (x - x_j)$$

Example 3.5.1. *Construct the divided difference table for the data:*

x	$f(x)$
1	2
3	8
4	15
6	35

Solution: *We compute the divided differences step by step: First order divided differences:*

$$f[1,3] = \frac{8-2}{3-1} = \frac{6}{2} = 3$$
$$f[3,4] = \frac{15-8}{4-3} = \frac{7}{1} = 7$$
$$f[4,6] = \frac{35-15}{6-4} = \frac{20}{2} = 10$$

Second order divided differences:

$$f[1,3,4] = \frac{f[3,4]-f[1,3]}{4-1} = \frac{7-3}{3} = \frac{4}{3}$$
$$f[3,4,6] = \frac{f[4,6]-f[3,4]}{6-3} = \frac{10-7}{3} = 1$$

Third order divided difference:

$$f[1,3,4,6] = \frac{f[3,4,6]-f[1,3,4]}{6-1} = \frac{1-\frac{4}{3}}{5} = \frac{-\frac{1}{3}}{5} = -\frac{1}{15}$$

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
1	2			
		3		
3	8		$\frac{4}{3}$	
		7		$-\frac{1}{15}$
4	15		1	
		10		
6	35			

The Newton's interpolating polynomial is:

$$p_3(x) = 2 + 3(x-1) + \frac{4}{3}(x-1)(x-3) - \frac{1}{15}(x-1)(x-3)(x-4)$$

Example 3.5.2. *Using the data from Example 3.5.1, find the value of $f(2.5)$.*

Solution: *Using the polynomial from Example 3.5.1:*

$$\begin{aligned}
p_3(2.5) &= 2 + 3(2.5-1) + \frac{4}{3}(2.5-1)(2.5-3) - \frac{1}{15}(2.5-1)(2.5-3)(2.5-4) \\
&= 2 + 3(1.5) + \frac{4}{3}(1.5)(-0.5) - \frac{1}{15}(1.5)(-0.5)(-1.5) \\
&= 2 + 4.5 - 1 - \frac{1.125}{15} \\
&= 5.5 - 0.075 = 5.425
\end{aligned}$$

3.5.2. Newton's Forward Form for Interpolation

Newton divided difference formula when $x_0, x_1, x_2, \dots, x_n$ are equispaced, i.e., $h = x_{i+1} - x_i$ for $i = 0, 1, 2, \dots, n$.

Forward Difference Operator: The forward difference operator is denoted as Δ and defined as:

$$\Delta f(x) = f(x+h) - f(x)$$

Higher Order Forward Differences:

- Zero order: $\Delta^0 f_i = f_i$
- First order: $\Delta^1 f_i = f_{i+1} - f_i$
- Second order: $\Delta^2 f_i = \Delta f_{i+1} - \Delta f_i = f_{i+2} - 2f_{i+1} + f_i$
- Third order: $\Delta^3 f_i = f_{i+3} - 3f_{i+2} + 3f_{i+1} - f_i$
- nth order: $\Delta^n f_i = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i$

General Formula for $\Delta^n f_i$:

$$\Delta^n f_i = \sum_{k=0}^{n} (-1)^{n-k} \binom{n}{k} f_{i+k}$$

Relationship with Divided Differences: For equally spaced points with spacing h:

$$f[x_0, x_1, \dots, x_k] = \frac{\Delta^k f_0}{k! h^k}$$

For equally spaced points, we can write $x = x_0 + sh$ where $s = \frac{x - x_0}{h}$, so $x - x_k = (s-k)h$. Therefore:

$$P_n(x_0 + sh) = f[x_0] + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f(x_0)$$

This is **Newton's Forward Difference Formula**.

Binomial Coefficient: $\binom{s}{k} = \frac{s(s-1)(s-2)\cdots(s-k+1)}{k!}$

Example 3.5.3. *Construct forward difference table for equally spaced data with $h = 1$:*

x	$f(x)$
0	1
1	4
2	9
3	16
4	25

Solution: *We compute the forward differences:*

$$\begin{aligned}
\Delta f_0 &= f_1 - f_0 = 4 - 1 = 3 \\
\Delta f_1 &= f_2 - f_1 = 9 - 4 = 5 \\
\Delta f_2 &= f_3 - f_2 = 16 - 9 = 7 \\
\Delta f_3 &= f_4 - f_3 = 25 - 16 = 9
\end{aligned}$$

$$\begin{aligned}
\Delta^2 f_0 &= \Delta f_1 - \Delta f_0 = 5 - 3 = 2 \\
\Delta^2 f_1 &= \Delta f_2 - \Delta f_1 = 7 - 5 = 2 \\
\Delta^2 f_2 &= \Delta f_3 - \Delta f_2 = 9 - 7 = 2
\end{aligned}$$

$$\Delta^3 f_0 = \Delta^2 f_1 - \Delta^2 f_0 = 2 - 2 = 0$$
$$\Delta^3 f_1 = \Delta^2 f_2 - \Delta^2 f_1 = 2 - 2 = 0$$

$$\Delta^4 f_0 = \Delta^3 f_1 - \Delta^3 f_0 = 0 - 0 = 0$$

x	$f(x)$	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
0	*1*				
		3			
1	*4*		*2*		
		5		*0*	
2	*9*		*2*		*0*
		7		*0*	
3	*16*		*2*		
		9			
4	*25*				

Since $\Delta^2 f$ is constant (= 2), this indicates the data follows a quadratic pattern. Using Newton's forward formula with $s = x$ (since $h = 1$ and $x_0 = 0$):

$$\begin{aligned} p_2(x) &= f_0 + \binom{s}{1}\Delta f_0 + \binom{s}{2}\Delta^2 f_0 \\ &= 1 + s \cdot 3 + \frac{s(s-1)}{2} \cdot 2 \\ &= 1 + 3s + s(s-1) \\ &= 1 + 3s + s^2 - s \\ &= 1 + 2s + s^2 = (s+1)^2 \end{aligned}$$

Therefore: $p_2(x) = (x+1)^2$

Example 3.5.4. *Using the forward difference table from Example 3.5.3, find $f(1.5)$.*

Solution: *Here $x = 1.5$, $x_0 = 0$, $h = 1$, so $s = \frac{1.5-0}{1} = 1.5$. Using Newton's forward formula:*

$$\begin{aligned} f(1.5) &= f_0 + \binom{s}{1}\Delta f_0 + \binom{s}{2}\Delta^2 f_0 \\ &= 1 + 1.5 \cdot 3 + \frac{1.5 \cdot 0.5}{2} \cdot 2 \\ &= 1 + 4.5 + 0.75 = 6.25 \end{aligned}$$

Verification: $(1.5+1)^2 = (2.5)^2 = 6.25$

3.5.3. Newton's Backward Difference

Motivation: When we want to interpolate near the end of the data table, backward differences provide better numerical stability.

If we reorder $\{x_0, x_1, \ldots, x_n\} \to \{x_n, \ldots, x_1, x_0\}$, and define the **backward difference operator** ∇:

$$\nabla f(x_n) = f(x_n) - f(x_{n-1})$$

Higher Order Backward Differences:

- Zero order: $\nabla^0 f_i = f_i$
- First order: $\nabla^1 f_i = f_i - f_{i-1}$

- Second order: $\nabla^2 f_i = \nabla f_i - \nabla f_{i-1} = f_i - 2f_{i-1} + f_{i-2}$
- nth order: $\nabla^n f_i = \nabla^{n-1} f_i - \nabla^{n-1} f_{i-1}$

General Formula for $\nabla^n f_i$:

$$\nabla^n f_i = \sum_{k=0}^{n} (-1)^k \binom{n}{k} f_{i-k}$$

We can write down **Newton's Backward Difference Formula**:

$$P_n(x) = f_n + \binom{s}{1}\nabla f_n + \binom{s+1}{2}\nabla^2 f_n + \binom{s+2}{3}\nabla^3 f_n + \cdots$$

where $s = \frac{x-x_n}{h}$

Example 3.5.5. *Using the same data from Example 3.5.3, construct the backward difference table:*

Solution: *We compute the backward differences:*

$$\begin{aligned}
\nabla f_1 &= f_1 - f_0 = 4 - 1 = 3 \\
\nabla f_2 &= f_2 - f_1 = 9 - 4 = 5 \\
\nabla f_3 &= f_3 - f_2 = 16 - 9 = 7 \\
\nabla f_4 &= f_4 - f_3 = 25 - 16 = 9
\end{aligned}$$

$$\begin{aligned}
\nabla^2 f_2 &= \nabla f_2 - \nabla f_1 = 5 - 3 = 2 \\
\nabla^2 f_3 &= \nabla f_3 - \nabla f_2 = 7 - 5 = 2 \\
\nabla^2 f_4 &= \nabla f_4 - \nabla f_3 = 9 - 7 = 2
\end{aligned}$$

$$\begin{aligned}
\nabla^3 f_3 &= \nabla^2 f_3 - \nabla^2 f_2 = 2 - 2 = 0 \\
\nabla^3 f_4 &= \nabla^2 f_4 - \nabla^2 f_3 = 2 - 2 = 0
\end{aligned}$$

$$\nabla^4 f_4 = \nabla^3 f_4 - \nabla^3 f_3 = 0 - 0 = 0$$

x	$f(x)$	∇f	$\nabla^2 f$	$\nabla^3 f$	$\nabla^4 f$
0	*1*				
1	*4*	*3*			
2	*9*	*5*	*2*		
3	*16*	*7*	*2*	*0*	
4	*25*	*9*	*2*	*0*	*0*

Example 3.5.6. *Using the backward difference table, find $f(3.5)$.*

Solution: *Here we use $x_n = 4$ as the reference point. $x = 3.5$, $x_n = 4$, $h = 1$, so $s = \frac{3.5-4}{1} = -0.5$.*
Using Newton's backward formula:

$$\begin{aligned}
f(3.5) &= f_4 + \binom{s}{1}\nabla f_4 + \binom{s+1}{2}\nabla^2 f_4 \\
&= 25 + (-0.5)(9) + \frac{(-0.5)(0.5)}{2}(2) \\
&= 25 - 4.5 - 0.25 = 20.25
\end{aligned}$$

Verification: $(3.5+1)^2 = (4.5)^2 = 20.25$

Choice Between Forward and Backward Differences:

- Use **forward differences** when interpolating near the beginning of the data table
- Use **backward differences** when interpolating near the end of the data table
- For central interpolation, consider using central differences or averaging both methods

Relationship Between Forward and Backward Differences:

$$\Delta f_{i-1} = \nabla f_i$$

$$\Delta^n f_i = \nabla^n f_{i+n}$$

Error Analysis: The truncation error for Newton's interpolation formulas is:

$$E(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x - x_k)$$

where ξ lies in the interval containing x and all interpolation points.

Example 3.5.7. *Suppose we want to add the point $(5, 36)$ to our data from Example 3.5.1. Show how Newton's method makes this easy.*

Solution: *The beauty of Newton's method is that when we add a new data point, we don't need to recalculate the entire table - we just extend it by adding one more column.*

Our original data was: $(1, 2)$, $(3, 8)$, $(4, 15)$, $(6, 35)$

Now we're adding $(5, 36)$ *to get:* $(1, 2)$, $(3, 8)$, $(4, 15)$, $(5, 36)$, $(6, 35)$

Step 1: *Calculate the new first-order divided differences involving* $(5, 36)$*:*

$$f[4, 5] = \frac{36 - 15}{5 - 4} = \frac{21}{1} = 21$$

$$f[5, 6] = \frac{35 - 36}{6 - 5} = \frac{-1}{1} = -1$$

Step 2: *Calculate the new second-order divided differences:*

$$f[3, 4, 5] = \frac{f[4, 5] - f[3, 4]}{5 - 3} = \frac{21 - 7}{2} = \frac{14}{2} = 7$$

$$f[4, 5, 6] = \frac{f[5, 6] - f[4, 5]}{6 - 4} = \frac{-1 - 21}{2} = \frac{-22}{2} = -11$$

Step 3: *Calculate the new third-order divided differences:*

$$f[1, 3, 4, 5] = \frac{f[3, 4, 5] - f[1, 3, 4]}{5 - 1} = \frac{7 - \frac{4}{3}}{4} = \frac{\frac{21-4}{3}}{4} = \frac{17/3}{4} = \frac{17}{12}$$

$$f[3, 4, 5, 6] = \frac{f[4, 5, 6] - f[3, 4, 5]}{6 - 3} = \frac{-11 - 7}{3} = \frac{-18}{3} = -6$$

Step 4: *Calculate the new fourth-order divided difference:*

$$f[1, 3, 4, 5, 6] = \frac{f[3, 4, 5, 6] - f[1, 3, 4, 5]}{6 - 1} = \frac{-6 - \frac{17}{12}}{5}$$

$$= \frac{\frac{-72-17}{12}}{5} = \frac{-89/12}{5} = -\frac{89}{60}$$

Extended Divided Difference Table:

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}]$
1	2				
		3			
3	8		$\frac{4}{3}$		
		7		$\frac{17}{12}$	
4	15		7		$-\frac{89}{60}$
		21		-6	
5	36		-11		
		-1			
6	35				

The new fourth-degree Newton interpolating polynomial is:

$$\begin{aligned} p_4(x) &= p_3(x) + f[1,3,4,5,6](x-1)(x-3)(x-4)(x-5) \\ &= 2 + 3(x-1) + \frac{4}{3}(x-1)(x-3) - \frac{1}{15}(x-1)(x-3)(x-4) \\ &\quad - \frac{89}{60}(x-1)(x-3)(x-4)(x-5) \end{aligned}$$

This demonstrates Newton's method's efficiency: we only needed to compute one new column of divided differences rather than reconstructing the entire polynomial from scratch.

3.6. Piecewise Polynomial Interpolation

Large numbers of data points result in higher degree interpolating polynomials. The oscillating nature of high degree polynomials induces more error in interpolation, a phenomenon known as **Runge's phenomenon**.

Problems with High-Degree Polynomials

- As the degree increases, polynomials tend to oscillate wildly between data points
- Small changes in data can cause large changes in the interpolant
- Computational instability increases with polynomial degree
- The error may actually increase with more data points

Solution Strategy

To overcome this inaccuracy:

- Total interval is divided into small sub-intervals
- On each interval a lower degree interpolating polynomial is constructed
- Different pieces are joined together with certain continuity conditions
- Approximation of functions by this type is called **piecewise polynomial interpolation**

3.6.1. Piecewise Linear Interpolation

The simplest piecewise interpolation consists of joining data points $(x_0, f_0), (x_1, f_1), \ldots, (x_n, f_n)$ by straight lines.

On interval $[x_{i-1}, x_i]$, we define the linear polynomial:

$$p_{1i}(x) = f_{i-1} + (x - x_{i-1})f[x_{i-1}, x_i], \quad i = 1, 2, \ldots, n$$

Alternative Form: Using the two-point form:

$$p_{1i}(x) = f_{i-1} \cdot \frac{x_i - x}{x_i - x_{i-1}} + f_i \cdot \frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Properties:

- $p_{1i}(x_{i-1}) = f_{i-1}$ and $p_{1i}(x_i) = f_i$

- The function is continuous everywhere
- The derivative has jump discontinuities at the nodes

Example 3.6.1. *Construct a piecewise linear interpolant for the data:*

x	0	1	2	3
$f(x)$	1	2	5	4

Solution: *We need three linear pieces. For each interval* $[x_{i-1}, x_i]$*:*
For $[0,1]$***:***

$$\begin{aligned} p_1(x) &= 1 + (x-0)\cdot\frac{2-1}{1-0} \\ &= 1 + x \end{aligned}$$

For $[1,2]$***:***

$$\begin{aligned} p_2(x) &= 2 + (x-1)\cdot\frac{5-2}{2-1} \\ &= 2 + 3(x-1) \\ &= -1 + 3x \end{aligned}$$

For $[2,3]$***:***

$$\begin{aligned} p_3(x) &= 5 + (x-2)\cdot\frac{4-5}{3-2} \\ &= 5 - (x-2) \\ &= 7 - x \end{aligned}$$

Therefore:

$$P(x) = \begin{cases} 1+x, & \text{if } x \in [0,1] \\ -1+3x, & \text{if } x \in [1,2] \\ 7-x, & \text{if } x \in [2,3] \end{cases}$$

To find $P(1.5)$*: Since* $1.5 \in [1,2]$*, we use:*

$$P(1.5) = -1 + 3(1.5) = 3.5$$

Error Analysis

For equispaced points with spacing $h = \frac{x_n - x_0}{n}$:

$$|f(x) - P(x)| \leq \frac{Mh^2}{8}$$

where $M = \max_{x\in[x_0,x_n]} |f''(x)|$.

Example 3.6.2. *Estimate the maximum error when approximating* $f(x) = x^3$ *on* $[0,2]$ *using piecewise linear interpolation with 4 equal subintervals.*

Solution:
Given: $h = \frac{2-0}{4} = 0.5$ *and* $f''(x) = 6x$
On $[0,2]$*:* $M = \max_{0\leq x\leq 2} |6x| = 12$
Maximum error:

$$\begin{aligned} E &\leq \frac{12 \times (0.5)^2}{8} \\ &= \frac{12 \times 0.25}{8} \\ &= 0.375 \end{aligned}$$

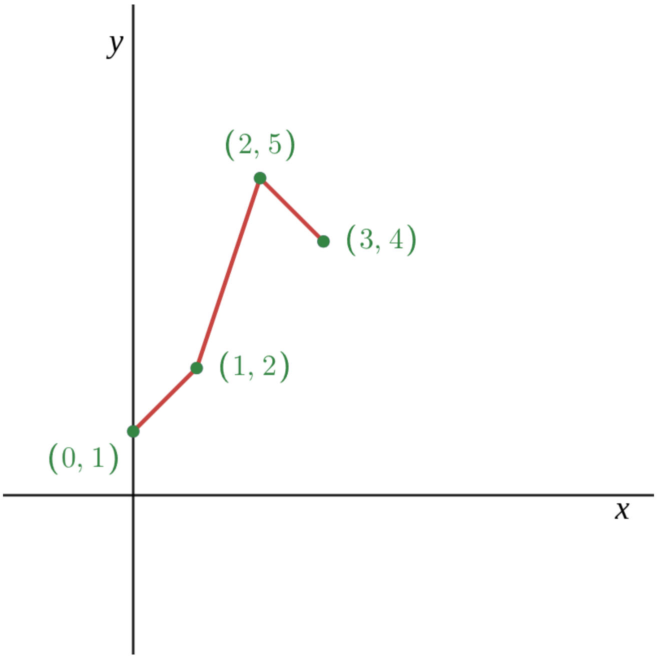


Figure 3.3: Example 3.6.1

3.6.2. Piecewise Quadratic Interpolation

Divide the interval $[a, b]$ into $2n$ equal intervals with:

$$h = \frac{b-a}{2n}$$

The quadratic interpolating polynomial for interval $[x_{2i-2}, x_{2i}]$ is:

$$\begin{aligned} P_{2i}(x) = f_{2i-2} &+ (x - x_{2i-2})f[x_{2i-2}, x_{2i-1}] \\ &+ (x - x_{2i-2})(x - x_{2i-1})f[x_{2i-2}, x_{2i-1}, x_{2i}] \end{aligned}$$

Example 3.6.3. *Construct piecewise quadratic interpolation for the data:*

x	*0*	*1*	*2*	*3*	*4*
$f(x)$	*1*	*2*	*5*	*10*	*17*

Solution:

We'll use two quadratic pieces:
First piece $[0, 2]$ *using points* $(0, 1)$, $(1, 2)$, $(2, 5)$*:*
Divided differences:

$$\begin{aligned} f[0,1] &= \frac{2-1}{1-0} = 1 \\ f[1,2] &= \frac{5-2}{2-1} = 3 \\ f[0,1,2] &= \frac{f[1,2]-f[0,1]}{2-0} = \frac{3-1}{2} = 1 \end{aligned}$$

Therefore:

$$\begin{aligned} P_1(x) &= 1 + (x-0)\cdot 1 + (x-0)(x-1)\cdot 1 \\ &= 1 + x + x(x-1) \\ &= 1 + x^2 \end{aligned}$$

Second piece *$[2,4]$ using points $(2,5)$, $(3,10)$, $(4,17)$:*
Divided differences:

$$\begin{aligned} f[2,3] &= \frac{10-5}{3-2} = 5 \\ f[3,4] &= \frac{17-10}{4-3} = 7 \\ f[2,3,4] &= \frac{f[3,4]-f[2,3]}{4-2} = \frac{7-5}{2} = 1 \end{aligned}$$

Therefore:

$$\begin{aligned} P_2(x) &= 5 + (x-2)\cdot 5 + (x-2)(x-3)\cdot 1 \\ &= 5 + 5(x-2) + (x-2)(x-3) \\ &= 5 + 5x - 10 + x^2 - 5x + 6 \\ &= 1 + x^2 \end{aligned}$$

Notice both pieces give the same formula: $P(x) = 1 + x^2$ for $x \in [0,4]$.
This occurs because the original data follows exactly the function $f(x) = 1 + x^2$.

Error Analysis

For equally spaced points:

$$|f(x) - P(x)| \leq \frac{Mh^3}{9\sqrt{3}}$$

where $M = \max_{x\in[a,b]} |f'''(x)|$.

3.6.3. Cubic Spline Interpolation

Definition 3.6.4 (Cubic Spline)**.** Given a function f defined on $[a,b]$ and nodes $a = x_0 < x_1 < \cdots < x_n = b$, a cubic spline interpolant S for f satisfies:

(a) $S(x)$ is a cubic polynomial $S_j(x)$ on each subinterval $[x_j, x_{j+1}]$;

(b) $S(x_j) = f(x_j)$ for all j (Interpolation condition);

(c) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ (First derivative continuity);

(d) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ (Second derivative continuity);

(e) Boundary conditions:

  (i) $S''(x_0) = S''(x_n) = 0$ (Natural boundary)

  (ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (Clamped boundary)

The cubic spline on interval $[x_j, x_{j+1}]$ has the form:

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

Algorithm for Natural Cubic Splines

Let $h_j = x_{j+1} - x_j$ and $\alpha_j = \frac{3}{h_j}(f_{j+1} - f_j) - \frac{3}{h_{j-1}}(f_j - f_{j-1})$.

The system of equations is:

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \alpha_j$$

for $j = 1, 2, \ldots, n-1$, with $c_0 = c_n = 0$.

Once c_j values are found:

$$\begin{aligned} a_j &= f_j \\ b_j &= \frac{f_{j+1} - f_j}{h_j} - \frac{h_j(2c_j + c_{j+1})}{3} \\ d_j &= \frac{c_{j+1} - c_j}{3h_j} \end{aligned}$$

Example 3.6.5. *Construct a natural cubic spline that passes through points* $(1,2)$, $(2,3)$, *and* $(3,5)$.

Solution:

Given: $(x_0, f_0) = (1,2)$, $(x_1, f_1) = (2,3)$, $(x_2, f_2) = (3,5)$
Step 1: *Calculate intervals*

$$h_0 = 2 - 1 = 1, \quad h_1 = 3 - 2 = 1$$

Step 2: *Calculate α values*

$$\begin{aligned} \alpha_1 &= \frac{3}{1}(5-3) - \frac{3}{1}(3-2) \\ &= 6 - 3 = 3 \end{aligned}$$

Step 3: *Solve tridiagonal system For natural spline, $c_0 = c_2 = 0$. The system reduces to:*

$$\begin{aligned} 2(1+1)c_1 &= 3 \\ 4c_1 &= 3 \\ c_1 &= \frac{3}{4} \end{aligned}$$

Step 4: *Calculate coefficients*
For piece 0: $a_0 = 2$, $c_0 = 0$

$$\begin{aligned} b_0 &= \frac{f_1 - f_0}{h_0} - \frac{h_0(2c_0 + c_1)}{3} \\ &= \frac{3-2}{1} - \frac{1(2 \cdot 0 + \frac{3}{4})}{3} \\ &= 1 - \frac{3/4}{3} = 1 - \frac{1}{4} = \frac{3}{4} \\ d_0 &= \frac{c_1 - c_0}{3h_0} = \frac{\frac{3}{4} - 0}{3 \cdot 1} = \frac{1}{4} \end{aligned}$$

For piece 1: $a_1 = 3$, $c_1 = \frac{3}{4}$, $c_2 = 0$

$$\begin{aligned} b_1 &= \frac{f_2 - f_1}{h_1} - \frac{h_1(2c_1 + c_2)}{3} \\ &= \frac{5-3}{1} - \frac{1(2 \cdot \frac{3}{4} + 0)}{3} \\ &= 2 - \frac{3/2}{3} = 2 - \frac{1}{2} = \frac{3}{2} \\ d_1 &= \frac{c_2 - c_1}{3h_1} = \frac{0 - \frac{3}{4}}{3 \cdot 1} = -\frac{1}{4} \end{aligned}$$

Therefore:

$$S(x) = \begin{cases} 2 + \frac{3}{4}(x-1) + \frac{1}{4}(x-1)^3, & x \in [1,2] \\ 3 + \frac{3}{2}(x-2) + \frac{3}{4}(x-2)^2 - \frac{1}{4}(x-2)^3, & x \in [2,3] \end{cases}$$

Example 3.6.6. *Construct a natural cubic spline for the data:*

x	*0*	*1*	*2*	*3*
$f(x)$	*1*	*4*	*2*	*3*

Solution: Step 1: $h_0 = h_1 = h_2 = 1$
Step 2: *Calculate α values*

$$\alpha_1 = \frac{3}{h_1}(f_2 - f_1) - \frac{3}{h_0}(f_1 - f_0) = \frac{3}{1}(2-4) - \frac{3}{1}(4-1) = -6 - 9 = -15$$
$$\alpha_2 = \frac{3}{h_2}(f_3 - f_2) - \frac{3}{h_1}(f_2 - f_1) = \frac{3}{1}(3-2) - \frac{3}{1}(2-4) = 3 + 6 = 9$$

Step 3: *Set up tridiagonal system with $c_0 = c_3 = 0$:*
The general form is: $h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = \alpha_j$

$$2(h_0 + h_1)c_1 + h_1 c_2 = \alpha_1$$
$$h_1 c_1 + 2(h_1 + h_2)c_2 = \alpha_2$$

Substituting values:

$$2(1+1)c_1 + 1 \cdot c_2 = -15$$
$$1 \cdot c_1 + 2(1+1)c_2 = 9$$

This gives us:

$$4c_1 + c_2 = -15$$
$$c_1 + 4c_2 = 9$$

From the second equation: $c_1 = 9 - 4c_2$
Substituting into the first:

$$4(9 - 4c_2) + c_2 = -15$$
$$36 - 16c_2 + c_2 = -15$$
$$36 - 15c_2 = -15$$
$$15c_2 = 51$$
$$c_2 = \frac{51}{15} = \frac{17}{5}$$

Therefore: $c_1 = 9 - 4 \cdot \frac{17}{5} = 9 - \frac{68}{5} = \frac{45-68}{5} = -\frac{23}{5}$

Example 3.6.7. *Construct a clamped cubic spline for points* $(0,1)$, $(1,2)$, $(2,0)$ *with boundary conditions* $S'(0) = 0$ *and* $S'(2) = -1$.
Solution: *Given:* $(x_0, f_0) = (0,1)$, $(x_1, f_1) = (1,2)$, $(x_2, f_2) = (2,0)$ $h_0 = h_1 = 1$, $S'(0) = 0$, $S'(2) = -1$
For clamped splines, the system is:

$$2h_0c_0 + h_0c_1 = 3\left[\frac{f_1 - f_0}{h_0} - S'(0)\right] = 3[1-0] = 3$$
$$h_0c_0 + 2(h_0 + h_1)c_1 + h_1c_2 = 3\left[\frac{f_2 - f_1}{h_1} - \frac{f_1 - f_0}{h_0}\right] = 3[-2-1] = -9$$
$$h_1c_1 + 2h_1c_2 = 3\left[S'(2) - \frac{f_2 - f_1}{h_1}\right] = 3[-1+2] = 3$$

The system becomes:

$$
\begin{aligned}
2c_0 + c_1 &= 3 \\
c_0 + 4c_1 + c_2 &= -9 \\
c_1 + 2c_2 &= 3
\end{aligned}
$$

From equations (1) and (3): $c_0 = \frac{3-c_1}{2}$ *and* $c_2 = \frac{3-c_1}{2}$ *Substituting into equation (2):*

$$
\begin{aligned}
\frac{3-c_1}{2} + 4c_1 + \frac{3-c_1}{2} &= -9 \\
3 - c_1 + 4c_1 &= -18 \\
3 + 3c_1 &= -18 \\
c_1 &= -7
\end{aligned}
$$

Therefore: $c_0 = c_2 = 5$ *Computing other coefficients:*

$$
\begin{aligned}
a_0 = 1, \quad a_1 &= 2 \\
b_0 &= \frac{2-1}{1} - \frac{1(10-7)}{3} = 1 - 1 = 0 \\
b_1 &= \frac{0-2}{1} - \frac{1(-14+5)}{3} = -2 + 3 = 1 \\
d_0 &= \frac{-7-5}{3} = -4 \\
d_1 &= \frac{5+7}{3} = 4
\end{aligned}
$$

Therefore:

$$
S(x) = \begin{cases} 1 + 5x^2 - 4x^3, & x \in [0,1] \\ 2 + (x-1) - 7(x-1)^2 + 4(x-1)^3, & x \in [1,2] \end{cases}
$$

Verification: $S'(2) = 1 - 14 + 12 = -1$

Error Bounds

If $f \in C^4[a,b]$ and $h = \max_j(x_{j+1} - x_j)$, then:

$$
\max_{a \le x \le b} |f(x) - S(x)| \le \frac{5h^4}{384} \max_{a \le x \le b} |f^{(4)}(x)|
$$

Comparison of Methods

Method	**Continuity**	**Error Order**	**Smoothness**
Piecewise Linear	C^0	$O(h^2)$	Not smooth
Piecewise Quadratic	C^0	$O(h^3)$	Not smooth at nodes
Cubic Splines	C^2	$O(h^4)$	Very smooth

3.7. Hermite's Interpolation

In many practical applications, we not only know the function values at certain points but also have information about the derivatives. For example, in physics, we might know both position and velocity at specific times. Hermite interpolation allows us to utilize this additional derivative information to construct a more accurate polynomial approximation.

Key Advantage: Unlike Lagrange and Newton interpolation which only match function values, Hermite interpolation matches both function values and derivative values, providing better local approximation properties.

Definition 3.7.1 (Hermite Interpolating Polynomial)**.** Let $x_0, x_1, \ldots, x_n$ be $n+1$ distinct points in interval $[a, b]$. The polynomial $P(x)$ is a **Hermite interpolating polynomial** for $f(x)$ if:

$$P(x_i) = f(x_i) \quad \text{and} \quad P'(x_i) = f'(x_i), \quad i = 0, 1, \ldots, n$$

Geometric Interpretation: The polynomial $P(x)$ and function $f(x)$ agree in both function values and first derivative values at each x_i. This means they have the same tangent lines at the interpolation points.

Degree Analysis: With $n+1$ points, we have:

- $n+1$ function value conditions: $P(x_i) = f(x_i)$
- $n+1$ derivative conditions: $P'(x_i) = f'(x_i)$
- Total: $2(n+1) = 2n+2$ conditions
- Therefore, we need a polynomial of degree at most $2n+1$

Construction of Hermite Polynomials

Definition 3.7.2 (Hermite Polynomial Formula)**.** If $f \in C^1[a, b]$ and $x_0, x_1, \ldots, x_n$ are distinct numbers in $[a, b]$, the unique polynomial of least degree agreeing with f and f' at $x_0, \ldots, x_n$ is:

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) H_{n,j}(x) + \sum_{j=0}^{n} f'(x_j) \tilde{H}_{n,j}(x)$$

where $L_{n,j}(x)$ is the jth Lagrange coefficient polynomial of degree n, and:

$$\begin{aligned} H_{n,j}(x) &= [1 - 2(x - x_j) L'_{n,j}(x_j)] L^2_{n,j}(x) \\ \tilde{H}_{n,j}(x) &= (x - x_j) L^2_{n,j}(x) \end{aligned}$$

Basis Function Properties:

- $H_{n,j}(x_i) = \delta_{ij}$ and $H'_{n,j}(x_i) = 0$ for all i, j
- $\tilde{H}_{n,j}(x_i) = 0$ and $\tilde{H}'_{n,j}(x_i) = \delta_{ij}$ for all i, j

Alternative Construction: Newton's Divided Difference Approach

For a point x_i where we know both $f(x_i)$ and $f'(x_i)$, we use:

$$f[x_i, x_i] = f'(x_i)$$

Example 3.7.3. *Find the Hermite polynomial that interpolates $f(x)$ at $x_0 = 0$ and $x_1 = 1$ given: $f(0) = 1$, $f'(0) = 2$, $f(1) = 4$, $f'(1) = -1$.*

Solution:

We need a cubic polynomial (degree $\leq 2 \cdot 2 - 1 = 3$).
Step 1: *Construct the Lagrange polynomials*

$$\begin{aligned} L_{1,0}(x) &= \frac{x-1}{0-1} = 1 - x \\ L_{1,1}(x) &= \frac{x-0}{1-0} = x \end{aligned}$$

Their derivatives:

$$\begin{aligned} L'_{1,0}(x) &= -1 \\ L'_{1,1}(x) &= 1 \end{aligned}$$

Step 2: *Construct the Hermite basis functions*

$$\begin{aligned} H_{1,0}(x) &= [1 - 2(x-0)L'_{1,0}(0)](1-x)^2 \\ &= [1 - 2x(-1)](1-x)^2 \\ &= (1+2x)(1-x)^2 \end{aligned}$$

$$\begin{aligned} H_{1,1}(x) &= [1 - 2(x-1)L'_{1,1}(1)]x^2 \\ &= [1 - 2(x-1)(1)]x^2 \\ &= (3-2x)x^2 \end{aligned}$$

$$\begin{aligned} \tilde{H}_{1,0}(x) &= (x-0)(1-x)^2 = x(1-x)^2 \\ \tilde{H}_{1,1}(x) &= (x-1)x^2 \end{aligned}$$

Step 3: *Construct the Hermite polynomial*

$$\begin{aligned} H_3(x) &= f(0) \cdot H_{1,0}(x) + f(1) \cdot H_{1,1}(x) + f'(0) \cdot \tilde{H}_{1,0}(x) + f'(1) \cdot \tilde{H}_{1,1}(x) \\ &= 1 \cdot (1+2x)(1-x)^2 + 4 \cdot (3-2x)x^2 + 2 \cdot x(1-x)^2 + (-1) \cdot (x-1)x^2 \end{aligned}$$

Expanding each term:

$$\begin{aligned} (1+2x)(1-x)^2 &= (1+2x)(1-2x+x^2) = 1 - 3x^2 + 2x^3 \\ 4(3-2x)x^2 &= 12x^2 - 8x^3 \\ 2x(1-x)^2 &= 2x(1-2x+x^2) = 2x - 4x^2 + 2x^3 \\ -(x-1)x^2 &= -x^3 + x^2 \end{aligned}$$

Therefore:

$$\begin{aligned} H_3(x) &= (1 - 3x^2 + 2x^3) + (12x^2 - 8x^3) + (2x - 4x^2 + 2x^3) + (-x^3 + x^2) \\ &= 1 + 2x + 6x^2 - 5x^3 \end{aligned}$$

Verification:

$$\begin{aligned} H_3(0) &= 1 \\ H_3(1) &= 1 + 2 + 6 - 5 = 4 \\ H'_3(x) &= 2 + 12x - 15x^2 \\ H'_3(0) &= 2 \\ H'_3(1) &= 2 + 12 - 15 = -1 \end{aligned}$$

Example 3.7.4. *Solve the previous example using the divided difference approach.*
Solution: *Create a divided difference table with repeated nodes:*

x	$f[x]$	$f[x,x]$	$f[x,x,x]$	$f[x,x,x,x]$
0	*1*			
		2		
0	*1*		*1*	
		3		-2
1	*4*		-4	
		-1		
1	*4*			

Computing the divided differences:

$$
\begin{aligned}
f[0,0] &= f'(0) = 2 \\
f[1,1] &= f'(1) = -1 \\
f[0,1] &= \frac{f(1)-f(0)}{1-0} = \frac{4-1}{1} = 3 \\
f[0,0,1] &= \frac{f[0,1]-f[0,0]}{1-0} = \frac{3-2}{1} = 1 \\
f[0,1,1] &= \frac{f[1,1]-f[0,1]}{1-0} = \frac{-1-3}{1} = -4 \\
f[0,0,1,1] &= \frac{f[0,1,1]-f[0,0,1]}{1-0} = \frac{-4-1}{1} = -5
\end{aligned}
$$

The Newton form of the Hermite polynomial is:

$$
\begin{aligned}
H_3(x) &= f[0] + f[0,0]\cdot x + f[0,0,1]\cdot x(x-0) + f[0,0,1,1]\cdot x(x-0)(x-1) \\
&= 1 + 2x + 1\cdot x^2 + (-5)\cdot x^2(x-1) \\
&= 1 + 2x + x^2 - 5x^3 + 5x^2 \\
&= 1 + 2x + 6x^2 - 5x^3
\end{aligned}
$$

Verification:

$$
\begin{aligned}
&H_3(0) = 1 \\
&H_3(1) = 1 + 2 + 6 - 5 = 4 \\
&H_3'(x) = 2 + 12x - 15x^2 \\
&H_3'(0) = 2 \\
&H_3'(1) = 2 + 12 - 15 = -1
\end{aligned}
$$

Example 3.7.5. *A projectile's height $h(t)$ and velocity $v(t) = h'(t)$ are measured at three times:*

Time t	*Height $h(t)$*	*Velocity $h'(t)$*
0	0	20
1	15	10
2	20	0

Find the height at $t = 0.5$ using Hermite interpolation.

Solution:

Using the divided difference approach:

t	$h[t]$	$h[t,t]$	$h[\cdot,\cdot,\cdot]$	$h[\cdot,\cdot,\cdot,\cdot]$	$h[\cdot,\cdot,\cdot,\cdot,\cdot]$	$h[\cdot,\cdot,\cdot,\cdot,\cdot,\cdot]$
0	0					
		20				
0	0		-5			
		15		-2.5		
1	15		-10		0.833	
		10		-2.5		-0.417
1	15		-5		0	
		5		-5		
2	20		0			
		0				
2	20					

Computing the divided differences:

$$\begin{aligned}
h[0,0] &= h'(0) = 20 \\
h[1,1] &= h'(1) = 10 \\
h[2,2] &= h'(2) = 0 \\
h[0,1] &= \frac{h(1)-h(0)}{1-0} = \frac{15-0}{1} = 15 \\
h[1,2] &= \frac{h(2)-h(1)}{2-1} = \frac{20-15}{1} = 5 \\
h[0,0,1] &= \frac{h[0,1]-h[0,0]}{1-0} = \frac{15-20}{1} = -5 \\
h[0,1,1] &= \frac{h[1,1]-h[0,1]}{1-0} = \frac{10-15}{1} = -5 \\
h[1,1,2] &= \frac{h[1,2]-h[1,1]}{2-1} = \frac{5-10}{1} = -5 \\
h[0,0,1,1] &= \frac{h[0,1,1]-h[0,0,1]}{1-0} = \frac{-5-(-5)}{1} = 0 \\
h[0,1,1,2] &= \frac{h[1,1,2]-h[0,1,1]}{2-0} = \frac{-5-(-5)}{2} = 0
\end{aligned}$$

At $t = 0.5$:

$$\begin{aligned}
H_3(0.5) &= 0 + 20(0.5) + (-5)(0.5)^2 + 0(0.5)^2(0.5-1) \\
&= 0 + 10 - 1.25 + 0 \\
&= 8.75 \textit{ meters}
\end{aligned}$$

Error Analysis

Definition 3.7.6 (Error Bound for Hermite Interpolation)**.** If $f \in C^{2n+2}[a,b]$, then for any $x \in [a,b]$:

$$f(x) = H_{2n+1}(x) + \frac{(x-x_0)^2(x-x_1)^2\cdots(x-x_n)^2}{(2n+2)!} f^{(2n+2)}(\zeta(x))$$

for some $\zeta(x) \in (a,b)$.

Therefore:

$$|f(x) - H_{2n+1}(x)| \le \frac{M}{(2n+2)!}\prod_{j=0}^{n}(x-x_j)^2$$

where $M = \max_{x\in[a,b]} |f^{(2n+2)}(x)|$.

Key Properties:

- **Uniqueness:** The Hermite polynomial of degree at most $2n+1$ is unique
- **Continuity:** Better continuity properties than standard interpolation
- **Local Behavior:** Matches both function and derivative, giving superior local approximation
- **Convergence:** Faster convergence than Lagrange interpolation for smooth functions

Computational Considerations:

- The divided difference approach is often more numerically stable

- Lagrange-based approach gives explicit formulas but can be computationally intensive
- For large datasets, consider piecewise Hermite interpolation
- Software implementations typically use the divided difference approach

3.8. Comparison with Other Methods

Method	**Information Used**	**Degree**	**Continuity**	**Error Order**
Lagrange	Function values	n	C^0	$O(h^{n+1})$
Newton	Function values	n	C^0	$O(h^{n+1})$
Hermite	Function + derivatives	$2n+1$	C^1	$O(h^{2n+2})$
Cubic Splines	Function values	Piecewise 3	C^2	$O(h^4)$

When to Use Hermite Interpolation:

- When derivative information is available
- For problems requiring smooth interpolation
- When high accuracy is needed with few data points
- In applications where physical continuity conditions matter (e.g., trajectory planning)

Practice Questions

1. For the given functions $f(x)$, let $x_0 = 0$, $x_1 = 0.6$, and $x_2 = 0.9$. Construct interpolation polynomials of degree at most one and at most two to approximate $f(0.45)$, and find the actual error.

   (a) $f(x) = \cos(x)$

   (b) $f(x) = \sqrt{1+x}$

   (c) $f(x) = \ln(x+1)$

   (d) $f(x) = \tan(x)$

2. Show that the polynomial interpolating the following data has degree 3.

x	-2	-1	0	1	2	3
$f(x)$	1	4	11	16	13	-4

3. (a) Show that the Newton forward divided-difference polynomials

$$\begin{aligned} P(x) &= 3 - 2(x+1) + 0(x+1)(x) + (x+1)(x)(x-1) \\ \text{and}\quad Q(x) &= -1 + 4(x+2) - 3(x+2)(x+1) + (x+2)(x+1)(x) \end{aligned}$$

   both interpolate the data

x	-2	-1	0	1	2
$f(x)$	-1	3	1	-1	3

   (b) Why does part (a) not violate the uniqueness property of interpolating polynomials?

4. The following data are given for a polynomial $P(x)$ of unknown degree.

x	0	1	2	3
$P(x)$	4	9	15	18

   Determine the coefficient of x^3 in $P(x)$ if all fourth-order forward differences are 1.

5. The Newton forward divided-difference formula is used to approximate $f(0.3)$ given the following data.

x	0.0	0.2	0.4	0.6
$f(x)$	15.0	21.0	30.0	51.0

   Suppose it is discovered that $f(0.4)$ was understated by 10 and $f(0.6)$ was overstated by 5. By what amount should the approximation to $f(0.3)$ be changed?

6. Let $f(x) = 3xe^x - e^{2x}$.

   (a) Approximate $f(1.03)$ by the Hermite interpolating polynomial of degree at most three using $x_0 = 1$ and $x_1 = 1.05$. Compare the actual error to the error bound.

   (b) Repeat (a) with the Hermite interpolating polynomial of degree at most five, using $x_0 = 1$, $x_1 = 1.05$, and $x_2 = 1.07$.

7. A car traveling along a straight road is clocked at a number of points. The data from the observations are given in the following table, where the time is in seconds, the distance is in feet, and the speed is in feet per second.

Time	0	3	5	8	13
Distance	0	225	383	623	993
Speed	75	77	80	74	72

   (a) Use a Hermite polynomial to predict the position of the car and its speed when $t = 10$ s.

   (b) Use the derivative of the Hermite polynomial to determine whether the car ever exceeds a 55 mi/h speed limit on the road. If so, what is the first time the car exceeds this speed?

(c) What is the predicted maximum speed for the car?

8. (a) Show that $H_{2n+1}(x)$ is the unique polynomial of least degree agreeing with f and f' at $x_0, \dots, x_n$. [Hint: Assume that $P(x)$ is another such polynomial and consider $D = H_{2n+1} - P$ and D' at $x_0, x_1, \dots, x_n$.]

   (b) Derive the error term in Theorem 3.9. [Hint: Use the same method as in the Lagrange error derivation, Theorem 3.3, defining

$$g(t) = f(t) - H_{2n+1}(t) - \frac{(t-x_0)^2 \cdots (t-x_n)^2}{(x-x_0)^2 \cdots (x-x_n)^2}[f(x) - H_{2n+1}(x)]$$

   and using the fact that $g'(t)$ has $(2n+2)$ distinct zeros in $[a, b]$.]

9. Let $z_0 = x_0$, $z_1 = x_0$, $z_2 = x_1$, and $z_3 = x_1$. Form the following divided-difference table.

$z_0 = x_0$	$f[z_0] = f(x_0)$			
		$f[z_0, z_1] = f'(x_0)$		
$z_1 = x_0$	$f[z_1] = f(x_0)$		$f[z_0, z_1, z_2]$	
		$f[z_1, z_2]$		$f[z_0, z_1, z_2, z_3]$
$z_2 = x_1$	$f[z_2] = f(x_1)$		$f[z_1, z_2, z_3]$	
		$f[z_2, z_3] = f'(x_1)$		
$z_3 = x_1$	$f[z_3] = f(x_1)$			

Show that the cubic Hermite polynomial $H_3(x)$ can also be written as $f[z_0] + f[z_0, z_1](x - x_0) + f[z_0, z_1, z_2](x - x_0)^2 + f[z_0, z_1, z_2, z_3](x - x_0)^2(x - x_1)$.

10. Determine the free cubic spline S that interpolates the data $f(0) = 0$, $f(1) = 1$, and $f(2) = 2$.

11. Determine the clamped cubic spline s that interpolates the data $f(0) = 0$, $f(1) = 1$, $f(2) = 2$ and satisfies $s'(0) = s'(2) = 1$.

12. Given the partition $x_0 = 0$, $x_1 = 0.05$, and $x_2 = 0.1$ of $[0, 0.1]$, find the piecewise linear interpolating function F for $f(x) = e^{2x}$. Approximate $\int_0^{0.1} e^{2x}\,dx$ with $\int_0^{0.1} F(x)\,dx$, and compare the results to the actual value.

13. A clamped cubic spline s for a function f is defined on $[1, 3]$ by

$$s(x) = \begin{cases} s_0(x) = 3(x-1) + 2(x-1)^2 - (x-1)^3, & \text{if } 1 \le x < 2, \\ s_1(x) = a + b(x-2) + c(x-2)^2 + d(x-2)^3, & \text{if } 2 \le x \le 3. \end{cases}$$

Given $f'(1) = f'(3)$, find a, b, c, and d.

14. Construct a free cubic spline to approximate $f(x) = \cos \pi x$ by using the values given by $f(x)$ at $x = 0, 0.25, 0.5, 0.75,$ and 1.0. Integrate the spline over $[0, 1]$, and compare the result to $\int_0^1 \cos \pi x\,dx = 0$. Use the derivatives of the spline to approximate $f'(0.5)$ and $f''(0.5)$. Compare these approximations to the actual values.

15. The Vandermonde matrix X is define as

$$V_n(x) = \det \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^n \\ 1 & x & x^2 & \cdots & x^n \end{bmatrix}$$

   (a) Show that $V_n(x)$ is a polynomial of degree n, and that its roots are $x_0, \dots, x_{n-1}$. Obtain the formula

$$V_n(x) = (x - x_0) \cdots (x - x_{n-1}) V_{n-1}(x_{n-1})$$

   Hint: Expand the last row of $V_n(x)$ by minors to show that $V_n(x)$ is a polynomial of degree n and to find the coefficient of the term x^n.

   (b) Show

$$\det(X) \equiv V_n(x_n) = \prod_{0 \le j < i \le n} (x_i - x_j)$$

Programming Questions

1. **Lagrange Interpolation**

   Write a Python function `lagrange_interpolation(x_points, y_points, x)` that computes the value of the Lagrange interpolating polynomial at a given point x, using lists of x-coordinates (`x_points`) and y-coordinates (`y_points`). Test your function with the following data:

   ```
   x_points = [0, 1, 2, 3]
   y_points = [1, 2, 4, 8]
   x = 1.5
   ```

2. **Newton Divided Differences**

   Write a Python function `newton_divided_differences(x_points, y_points)` that returns the coefficients of the Newton interpolating polynomial using divided differences.
   Then, write a function `newton_interpolate(coefficients, x_points, x)` to evaluate the polynomial at x. Use the same data as above and test at $x = 1.5$.

3. **Piecewise Linear Interpolation**

   Write a Python function `piecewise_linear(x_points, y_points, x)` that performs piecewise linear interpolation. Test with:

   ```
   x_points = [0, 1, 2, 3]
   y_points = [1, 2, 4, 8]
   x = 1.5
   ```

4. **Natural Cubic Spline**

   Implement a function `natural_cubic_spline(x_points, y_points)` that computes the coefficients for a natural cubic spline. Then, write a function `eval_natural_spline(x_points, coefficients, x)` to evaluate the spline at x. Use the same data and test at $x = 1.5$.

5. **Clamped Cubic Spline**

   Extend your cubic spline implementation to handle clamped boundary conditions with
   `clamped_cubic_spline(x_points, y_points, yprime0, yprimen)`, where `yprime0` and `yprimen` are derivatives at the endpoints. Test with the same data and `yprime0 = 1`, `yprimen = 4` at $x = 1.5$.

6. **Hermite Interpolation**

   Write a Python function `hermite_interpolation(x_points, y_points, y_prime_points, x)` that computes the Hermite interpolating polynomial value at x. Test with:

   ```
   x_points = [0, 1]
   y_points = [1, 2]
   y_prime_points = [0, 1]
   x = 0.5
   ```

Numerical Solutions of Nonlinear System of Equations

4.1. System of Nonlinear Equations

A system of nonlinear equations has the form:

$$f_1(x_1, x_2, \ldots, x_n) = 0 \tag{4.1}$$

$$f_2(x_1, x_2, \ldots, x_n) = 0 \tag{4.2}$$

$$\vdots \tag{4.3}$$

$$f_n(x_1, x_2, \ldots, x_n) = 0 \tag{4.4}$$

where each function f_i can be thought of as mapping a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ of the n-dimensional space $\mathbb{R}^n$ into the real line $\mathbb{R}$.

This system of n nonlinear equations in n unknowns can also be represented by defining a function $\mathbf{F}$ mapping $\mathbb{R}^n$ into $\mathbb{R}^n$ as:

$$\mathbf{F}(x_1, \ldots, x_n) = [f_1(x_1, \ldots, x_n), f_2(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)]^T$$

If vector notation is used to represent the variables $x_1, x_2, \ldots, x_n$, then the system assumes the form:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

The functions $f_1, f_2, \ldots, f_n$ are called the **coordinate functions** of $\mathbf{F}$.

4.2. Continuity and Limits for Vector-Valued Functions

Definition 4.2.1. Let f be a function defined on a set $D \subset \mathbb{R}^n$ and mapping into $\mathbb{R}$. The function f is said to have the limit L at $\mathbf{x}_0$, written

$$\lim_{\mathbf{x} \to \mathbf{x}_0} f(\mathbf{x}) = L$$

if, given any number $\varepsilon > 0$, a number $\delta > 0$ exists with

$$|f(\mathbf{x}) - L| < \varepsilon, \text{ whenever } \mathbf{x} \in D \text{ and } 0 < \|\mathbf{x} - \mathbf{x}_0\| < \delta$$

Definition 4.2.2. Let $\mathbf{F}$ be a function from $D \subset \mathbb{R}^n$ into $\mathbb{R}^n$ of the form

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}))^T$$

where f_i is a mapping from $\mathbb{R}^n$ into $\mathbb{R}$ for each i. We define

$$\lim_{\mathbf{x} \to \mathbf{x}_0} \mathbf{F}(\mathbf{x}) = \mathbf{L} = (L_1, L_2, \ldots, L_n)^T$$

if and only if $\lim_{\mathbf{x} \to \mathbf{x}_0} f_i(\mathbf{x}) = L_i$ for each $i = 1, 2, \ldots, n$.

The function $\mathbf{F}$ is continuous at $\mathbf{x}_0 \in D$ provided $\lim_{\mathbf{x} \to \mathbf{x}_0} \mathbf{F}(\mathbf{x})$ exists and

$$\lim_{\mathbf{x} \to \mathbf{x}_0} \mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x}_0)$$

4.3. Fixed-Point Theory

Definition 4.3.1 (Fixed Point)**.** A function $\mathbf{G}$ from $D \subset \mathbb{R}^n$ into $\mathbb{R}^n$ has a fixed point at $\mathbf{p} \in D$ if $\mathbf{G}(\mathbf{p}) = \mathbf{p}$.

Theorem 4.3.2 (Fixed-Point Theorem)**.** *Let $D = \{(x_1, x_2, \ldots, x_n)^T \mid a_i \leq x_i \leq b_i$, for each $i = 1, 2, \ldots, n\}$ for some collection of constants $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$.*

Suppose $\mathbf{G}$ is a continuous function from $D \subset \mathbb{R}^n$ into $\mathbb{R}^n$ with the property that $\mathbf{G}(\mathbf{x}) \in D$ whenever $\mathbf{x} \in D$. Then $\mathbf{G}$ has a fixed point in D.

Suppose that all the component functions of $\mathbf{G}$ have continuous partial derivatives and a constant $K < 1$ exists with

$$\left| \frac{\partial g_i(\mathbf{x})}{\partial x_j} \right| \leq \frac{K}{n}$$

whenever $\mathbf{x} \in D$, for each $j = 1, 2, \ldots, n$ and each component function g_i. Then the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by an arbitrarily selected $\mathbf{x}^{(0)}$ in D and generated by

$$\mathbf{x}^{(k)} = \mathbf{G}(\mathbf{x}^{(k-1)})$$

for each $k \geq 1$ converges to the unique fixed point $\boldsymbol{\alpha} \in D$.

The convergence satisfies:

$$\|\mathbf{x}^{(k)} - \boldsymbol{\alpha}\|_\infty \leq \frac{K^k}{1-K} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|_\infty$$

Example 4.3.3. *Place the nonlinear system*

$$3x_1 - \cos(x_2 x_3) - \frac{1}{2} = 0 \tag{4.5}$$

$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0 \tag{4.6}$$

$$e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0 \tag{4.7}$$

in a fixed-point form $\mathbf{x} = \mathbf{G}(\mathbf{x})$ by solving the ith equation for x_i.

Solution: *Solving the ith equation for x_i gives the fixed-point problem:*

$$x_1 = \frac{1}{3}\cos(x_2 x_3) + \frac{1}{6} \tag{4.8}$$

$$x_2 = \frac{1}{9}\sqrt{x_1^2 + \sin x_3 + 1.06} - 0.1 \tag{4.9}$$

$$x_3 = -\frac{1}{20}e^{-x_1 x_2} - \frac{10\pi - 3}{60} \tag{4.10}$$

Let $\mathbf{G} : \mathbb{R}^3 \to \mathbb{R}^3$ be defined by $\mathbf{G}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), g_3(\mathbf{x}))^T$ where the component functions are as defined above. For $\mathbf{x} = (x_1, x_2, x_3)^T$ in $D = \{(x_1, x_2, x_3)^T \mid -1 \leq x_i \leq 1$, for each $i = 1, 2, 3\}$:

$$|g_1(x_1, x_2, x_3)| \leq \frac{1}{3}|\cos(x_2 x_3)| + \frac{1}{6} \leq 0.50$$

$$|g_2(x_1, x_2, x_3)| = \frac{1}{9}\sqrt{x_1^2 + \sin x_3 + 1.06} - 0.1 < 0.09$$

$$|g_3(x_1, x_2, x_3)| \leq \frac{1}{20}e + \frac{10\pi - 3}{60} < 0.61$$

Thus $\mathbf{G}(\mathbf{x}) \in D$ whenever $\mathbf{x} \in D$. The partial derivatives are bounded:

$$\left| \frac{\partial g_i(\mathbf{x})}{\partial x_j} \right| \leq 0.281$$

for each $i = 1, 2, 3$ and $j = 1, 2, 3$, so the condition holds with $K = 3(0.281) = 0.843$.

4.4. Newton-Raphson Method for Nonlinear Systems

4.4.1. Motivation and Development

An appropriate fixed-point method in the one-dimensional case uses a function ϕ with the property that $g(x) = x - \phi(x)f(x)$ gives quadratic convergence to the fixed point α of the function g.

From this condition, Newton's method evolved by choosing $\phi(x) = \frac{1}{f'(x)}$, assuming that $f'(x) \neq 0$.

A similar approach in the n-dimensional case involves a matrix $\mathbf{A}(\mathbf{x})$ where each of the entries $a_{ij}(\mathbf{x})$ is a function from $\mathbb{R}^n$ into $\mathbb{R}$.

This requires that $\mathbf{A}(\mathbf{x})$ be found so that

$$\mathbf{G}(\mathbf{x}) = \mathbf{x} - \mathbf{A}(\mathbf{x})^{-1}\mathbf{F}(\mathbf{x})$$

gives quadratic convergence to the solution of $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

4.4.2. The Jacobian Matrix

Define the **Jacobian matrix** $\mathbf{J}(\mathbf{x})$ by:

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{pmatrix}$$

An appropriate choice for $\mathbf{A}(\mathbf{x})$ is $\mathbf{A}(\mathbf{x}) = \mathbf{J}(\mathbf{x})$. The function $\mathbf{G}$ is defined by:

$$\mathbf{G}(\mathbf{x}) = \mathbf{x} - \mathbf{J}(\mathbf{x})^{-1}\mathbf{F}(\mathbf{x})$$

The functional iteration procedure evolves from selecting $\mathbf{x}^{(0)}$ and generating, for $k \geq 1$:

$$\mathbf{x}^{(k)} = \mathbf{G}(\mathbf{x}^{(k-1)}) = \mathbf{x}^{(k-1)} - \mathbf{J}(\mathbf{x}^{(k-1)})^{-1}\mathbf{F}(\mathbf{x}^{(k-1)})$$

This is called **Newton's method for nonlinear systems**.

4.4.3. Algorithm for Newton's Method

Step 1: Choose initial approximation $\mathbf{x}^{(0)}$

Step 2: For $k = 0, 1, 2, \ldots$ until convergence:

- Compute $\mathbf{F}(\mathbf{x}^{(k)})$ and $\mathbf{J}(\mathbf{x}^{(k)})$
- Solve the linear system: $\mathbf{J}(\mathbf{x}^{(k)})\mathbf{y}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)})$
- Update: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{y}^{(k)}$

Step 3: Stop when $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \varepsilon$

Example 4.4.1. *Apply Newton's method to the system:*

$$3x_1 - \cos(x_2x_3) - \frac{1}{2} = 0 \tag{4.11}$$

$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0 \tag{4.12}$$

$$e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0 \tag{4.13}$$

with $\mathbf{x}^{(0)} = (0.1, 0.1, -0.1)^T$.

Solution: *Define*

$$\mathbf{F}(x_1, x_2, x_3) = (f_1(x_1, x_2, x_3), f_2(x_1, x_2, x_3), f_3(x_1, x_2, x_3))^T$$

where the component functions are as given above. The Jacobian matrix is:

$$\mathbf{J}(x_1, x_2, x_3) = \begin{pmatrix} 3 & x_3 \sin(x_2 x_3) & x_2 \sin(x_2 x_3) \\ 2x_1 & -162(x_2 + 0.1) & \cos x_3 \\ -x_2 e^{-x_1 x_2} & -x_1 e^{-x_1 x_2} & 20 \end{pmatrix}$$

For $\mathbf{x}^{(0)} = (0.1, 0.1, -0.1)^T$*:*

$$\mathbf{F}(\mathbf{x}^{(0)}) = (-0.199995, -2.269833417, 8.462025346)^T$$

$$\mathbf{J}(\mathbf{x}^{(0)}) = \begin{pmatrix} 3 & 9.999833334 \times 10^{-4} & 9.999833334 \times 10^{-4} \\ 0.2 & -32.4 & 0.995004165 \\ -0.0990049833 & -0.0990049833 & 20 \end{pmatrix}$$

Solving $\mathbf{J}(\mathbf{x}^{(0)})\mathbf{y}^{(0)} = -\mathbf{F}(\mathbf{x}^{(0)})$ *gives:*

$$\mathbf{y}^{(0)} = (0.3998696728, -0.08053315147, -0.4215204718)^T$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{y}^{(0)} = (0.4998696782, 0.01946684853, -0.5215204718)^T$$

Continuing this process:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\lVert\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\rVert_\infty$
0	*0.1000000000*	*0.1000000000*	*-0.1000000000*	
1	*0.4998696728*	*0.0194608485*	*-0.5215204718*	*0.4215204718*
2	*0.5000142403*	*0.0015885914*	*-0.5235569638*	*1.788* $\times 10^{-2}$
3	*0.5000000113*	*0.0000124448*	*-0.5235984500*	*1.576* $\times 10^{-3}$
4	*0.5000000000*	*8.516* $\times 10^{-10}$	*-0.5235987755*	*1.244* $\times 10^{-5}$
5	*0.5000000000*	*-1.375* $\times 10^{-11}$	*-0.5235987756*	*8.654* $\times 10^{-10}$

The method converges to the solution $(0.5, 0, -\pi/6)^T$ *with quadratic convergence.*

4.5. Two-Variable Newton's Method

For the system:

$$f_1(x_1, x_2) = 0 \tag{4.14}$$
$$f_2(x_1, x_2) = 0 \tag{4.15}$$

Suppose (x_1, x_2) is an approximate solution. We compute corrections h_1 and h_2 so that $(x_1 + h_1, x_2 + h_2)$ will be a better approximate solution.

Using linear terms in the Taylor expansion:

$$0 \approx f_1(x_1, x_2) + h_1 \frac{\partial f_1}{\partial x_1} + h_2 \frac{\partial f_1}{\partial x_2} \tag{4.16}$$
$$0 \approx f_2(x_1, x_2) + h_1 \frac{\partial f_2}{\partial x_1} + h_2 \frac{\partial f_2}{\partial x_2} \tag{4.17}$$

The coefficient matrix is the Jacobian matrix:

$$\mathbf{J} = \begin{pmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 \end{pmatrix}$$

The solution is:

$$\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = -\mathbf{J}^{-1} \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix}$$

Hence, Newton's method becomes:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{J}(\mathbf{x}^{(k)})^{-1}\mathbf{F}(\mathbf{x}^{(k)})$$

4.5.1. Convergence Conditions

The convergence depends on the initial approximation $\mathbf{x}^{(0)}$. A sufficient condition for convergence is:

$$\|\mathbf{J}_k^{-1}\| < 1$$

However, a necessary and sufficient condition for convergence is:

$$\rho(\mathbf{J}_k^{-1}) < 1$$

where $\rho(\mathbf{J}_k^{-1})$ is the spectral radius of the matrix $\mathbf{J}_k^{-1}$.

If the method converges, then its rate of convergence is quadratic. The iterations are stopped when:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| \leq \varepsilon$$

Example 4.5.1. *Solve the system with initial approximation* $(1.5, 0.5)$*:*

$$x^2 + xy + y^2 = 7 \tag{4.18}$$

$$x^3 + y^3 = 9 \tag{4.19}$$

Solution: *Define:*

$$f(x, y) = x^2 + xy + y^2 - 7$$

$$g(x, y) = x^3 + y^3 - 9$$

The Jacobian matrix is:

$$\mathbf{J}_k = \begin{pmatrix} 2x_k + y_k & x_k + 2y_k \\ 3x_k^2 & 3y_k^2 \end{pmatrix}$$

The inverse is:

$$\mathbf{J}_k^{-1} = \frac{1}{D_k} \begin{pmatrix} 3y_k^2 & -(x_k + 2y_k) \\ -3x_k^2 & 2x_k + y_k \end{pmatrix}$$

where $D_k = 3y_k^2(2x_k + y_k) - 3x_k^2(x_k + 2y_k)$*. Starting with* $(x_0, y_0) = (1.5, 0.5)$*:* ***Iteration 1:***

$$\mathbf{J}_0^{-1} = \frac{1}{-14.25} \begin{pmatrix} 0.75 & -2.5 \\ -6.75 & 3.5 \end{pmatrix}$$

$$\mathbf{F}(1.5, 0.5) = (-3.75, -5.50)^T$$

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} 1.5 \\ 0.5 \end{pmatrix} + \begin{pmatrix} 0.7675 \\ 0.4254 \end{pmatrix} = \begin{pmatrix} 2.2675 \\ 0.9254 \end{pmatrix}$$

Continuing similarly:

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} 2.0373 \\ 0.9645 \end{pmatrix}$$

$$\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} 2.0013 \\ 0.9987 \end{pmatrix}$$

The method converges to the exact solution $(2, 1)$*.*

4.6. Iteration Methods for Nonlinear Systems

Consider the system:

$$f(x,y) = 0 \tag{4.20}$$
$$g(x,y) = 0 \tag{4.21}$$

We can write this in the equivalent form:

$$x = F(x,y) \tag{4.22}$$
$$y = G(x,y) \tag{4.23}$$

If (ζ, η) is the solution, then:

$$\zeta = F(\zeta, \eta) \tag{4.24}$$
$$\eta = G(\zeta, \eta) \tag{4.25}$$

The general iteration method is:

$$x_{k+1} = F(x_k, y_k) \tag{4.26}$$
$$y_{k+1} = G(x_k, y_k) \tag{4.27}$$

4.6.1. Error Analysis

Let $\epsilon_k = \zeta - x_k$ and $\delta_k = \eta - y_k$ be the errors in the kth iteration. Then:

$$\epsilon_{k+1} = \epsilon_k F_x(x_k, y_k) + \delta_k F_y(x_k, y_k) \tag{4.28}$$
$$\delta_{k+1} = \epsilon_k G_x(x_k, y_k) + \delta_k G_y(x_k, y_k) \tag{4.29}$$

This can be written as:

$$\boldsymbol{\epsilon}_{k+1} = \mathbf{A}_k \boldsymbol{\epsilon}_k$$

where $\boldsymbol{\epsilon}^{(k)} = [\epsilon_k, \delta_k]^T$ and

$$\mathbf{A}_k = \begin{pmatrix} F_x & F_y \\ G_x & G_y \end{pmatrix}$$

Convergence Conditions:

- Sufficient condition: $\|\mathbf{A}_k\| < 1$ for each k
- Necessary and sufficient condition: $\rho(\mathbf{A}_k) < 1$ for each k

For the maximum absolute row sum norm:

$$|F_x(x_k, y_k)| + |F_y(x_k, y_k)| < 1$$

$$|G_x(x_k, y_k)| + |G_y(x_k, y_k)| < 1$$

Example 4.6.1. *Solve the system:*

$$f(x,y) = x^2 + 3x + y - 5 = 0 \tag{4.30}$$
$$g(x,y) = x^2 + 3y^2 - 4 = 0 \tag{4.31}$$

Solution: *We write the system in equivalent form:*

$$x = x + \alpha(x^2 + 3x + y - 5) = F(x, y) \tag{4.32}$$

$$y = y + \beta(x^2 + 3y^2 - 4) = G(x, y) \tag{4.33}$$

where α and β are parameters to be determined. Computing partial derivatives:

$$F_x(x, y) = 1 + (2x + 3)\alpha \tag{4.34}$$

$$F_y(x, y) = \alpha \tag{4.35}$$

$$G_x(x, y) = 2\beta x \tag{4.36}$$

$$G_y(x, y) = 1 + 6\beta y \tag{4.37}$$

At $(x_0, y_0) = (0.5, 0.5)$:

$$F_x(0.5, 0.5) = 1 + 4\alpha \tag{4.38}$$

$$F_y(0.5, 0.5) = \alpha \tag{4.39}$$

$$G_x(0.5, 0.5) = \beta \tag{4.40}$$

$$G_y(0.5, 0.5) = 1 + 3\beta \tag{4.41}$$

For convergence:

$$|1 + 4\alpha| + |\alpha| < 1$$

$$|\beta| + |1 + 3\beta| < 1$$

Taking $\alpha = -1/4$ and $\beta = -1/6$:

$$x_{k+1} = x_k - \frac{1}{4}(x_k^2 + 3x_k + y_k - 5)$$

$$y_{k+1} = y_k - \frac{1}{6}(x_k^2 + 3y_k^2 - 4)$$

Starting with $(x_0, y_0) = (0.5, 0.5)$:

$$(x_1, y_1) = (1.1875, 1.0)$$

$$(x_2, y_2) = (0.944336, 0.931641)$$

$$(x_3, y_3) = (1.030231, 1.015702)$$

$$(x_4, y_4) = (0.988288, 0.989647)$$

$$(x_5, y_5) = (1.005482, 1.003828)$$

The method converges to the solution $(1, 1)$.

Least Square solutions

5.1. Introduction to Least Squares

Motivation: On a small data set, a single polynomial may not be sufficient. When the data has substantial error, least-squares approximation provides a better approach than exact interpolation.

Least-squares approximation involves finding a straight line in the form:

$$y = a_1 x + a_0$$

that best fits a set of n data points (x_i, y_i) for $i = 1, 2, \ldots, n$.

At each data point (x_i, y_i), the error e_i is defined as: $e_i = y_i - (a_1 x_i + a_0)$

5.2. Criteria for Best Fit

Several approaches can be considered for defining the "best" fit:

5.2.1. Sum of Errors

Minimize the sum of all individual errors: $E \equiv \sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - (a_1 x_i + a_0))$

Problem: This can result in zero-sum where positive and negative individual errors (even very large errors) cancel out.

5.2.2. Sum of Absolute Errors

Minimize the sum of the absolute values of individual errors: $E \equiv \sum_{i=1}^{n} |e_i| = \sum_{i=1}^{n} |y_i - (a_1 x_i + a_0)|$

Advantage: Individual errors cannot cancel out, and the total error is always positive.

5.2.3. Sum of Squared Errors (Least Squares)

Minimize the sum of the squares of individual errors: $E \equiv E(a_0, a_1) = \sum_{i=1}^{m} e_i^2 = \sum_{i=1}^{m} [y_i - (a_1 x_i + a_0)]^2$

This is the **least squares criterion** and is most commonly used because:

- It's mathematically tractable (differentiable)
- It gives more weight to larger errors
- It has desirable statistical properties

5.3. Linear Least Squares

Let there be a collection of data $\{(x_i, y_i)\}_{i=1}^{m}$. To fit the best least squares line requires minimizing the sum of squares of individual errors.

The parameters a_0 and a_1 are determined by setting the partial derivatives equal to zero:

$$\frac{\partial E}{\partial a_0} = \frac{\partial}{\partial a_0} \sum_{i=1}^{m} [y_i - (a_1 x_i + a_0)]^2 = -2 \sum_{i=1}^{m} (y_i - a_1 x_i - a_0) = 0 \tag{5.1}$$

$$\frac{\partial E}{\partial a_1} = \frac{\partial}{\partial a_1} \sum_{i=1}^{m} [y_i - (a_1 x_i + a_0)]^2 = -2 \sum_{i=1}^{m} (y_i - a_1 x_i - a_0) x_i = 0 \tag{5.2}$$

Simplifying these equations gives us the **normal equations**:

$$a_0 \cdot m + a_1 \sum_{i=1}^{m} x_i = \sum_{i=1}^{m} y_i \tag{5.3}$$

$$a_0 \sum_{i=1}^{m} x_i + a_1 \sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} x_i y_i \tag{5.4}$$

Solving the two equations, we get:

$$a_0 = \frac{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i - \sum_{i=1}^{m} x_i y_i \sum_{i=1}^{m} x_i}{m \left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)^2} \tag{5.5}$$

$$a_1 = \frac{m \sum_{i=1}^{m} x_i y_i - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i}{m \left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)^2} \tag{5.6}$$

Example 5.3.1. *Find the least square straight line fit to the following data:*

x	0	2	5	7
$f(x)$	1	5	12	20

Solution: *First, we compute the necessary summations from the data, where the number of data points is $m = 4$:*

$$\sum_{i=1}^{4} x_i = 0 + 2 + 5 + 7 = 14$$

$$\sum_{i=1}^{4} y_i = 1 + 5 + 12 + 20 = 38$$

$$\sum_{i=1}^{4} x_i^2 = 0^2 + 2^2 + 5^2 + 7^2 = 0 + 4 + 25 + 49 = 78$$

$$\sum_{i=1}^{4} x_i y_i = (0)(1) + (2)(5) + (5)(12) + (7)(20) = 0 + 10 + 60 + 140 = 210$$

We solve for a_0 and a_1 using the explicit formulas. First, we calculate the common denominator:

$$\begin{aligned} D &= m \left(\sum_{i=1}^{m} x_i^2\right) - \left(\sum_{i=1}^{m} x_i\right)^2 \\ &= 4(78) - (14)^2 \\ &= 312 - 196 \\ &= 116 \end{aligned}$$

Now we solve for the slope, a_1:

$$\begin{aligned} a_1 &= \frac{m\sum_{i=1}^{m} x_i y_i - \sum_{i=1}^{m} x_i \sum_{i=1}^{m} y_i}{D} \\ &= \frac{4(210) - (14)(38)}{116} \\ &= \frac{840 - 532}{116} \\ &= \frac{308}{116} = \frac{77}{29} \end{aligned}$$

And finally, we solve for the y-intercept, a_0:

$$\begin{aligned} a_0 &= \frac{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i - \sum_{i=1}^{m} x_i y_i \sum_{i=1}^{m} x_i}{D} \\ &= \frac{(78)(38) - (210)(14)}{116} \\ &= \frac{2964 - 2940}{116} \\ &= \frac{24}{116} = \frac{6}{29} \end{aligned}$$

Thus, the least squares straight line fit is given by the equation:

$$y = \frac{77}{29}x + \frac{6}{29}$$

5.4. General Least Squares Approximation

The linear least squares method can be extended to fit polynomials of higher degree or other functional forms.

For a polynomial of degree n: $P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$

The error function becomes: $E(a_0, a_1, \ldots, a_n) = \sum_{i=1}^{m}[y_i - P_n(x_i)]^2$

Setting $\frac{\partial E}{\partial a_j} = 0$ for $j = 0, 1, \ldots, n$ gives us the normal equations:

$$\sum_{k=0}^{n} a_k \sum_{i=1}^{m} x_i^{j+k} = \sum_{i=1}^{m} x_i^j y_i, \quad j = 0, 1, \ldots, n \tag{5.7}$$

This system can be written in matrix form as: $\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y}$

where $\mathbf{A}$ is the Vandermonde-like matrix: $\mathbf{A} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{pmatrix}$

Example 5.4.1. *Find the least squares polynomial of degree 3 for the data in the following table. Compute the error E.*

x_i	*1.0*	*1.1*	*1.3*	*1.5*	*1.9*	*2.1*
y_i	*1.84*	*1.96*	*2.21*	*2.45*	*2.94*	*3.18*

Solution: *We want to find the least squares polynomial of degree 3, which has the form:*

$$P_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

For $n = 3$ and $m = 6$ data points, the normal equations form a system of 4 linear equations. The system can be written in matrix form $\mathbf{Ma} = \mathbf{b}$, where the entries of matrix $\mathbf{M}$ and vector $\mathbf{b}$ are sums of powers of x_i and products of x_i and y_i.

$$\begin{pmatrix} m & \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 \\ \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \sum x_i^6 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \\ \sum x_i^3 y_i \end{pmatrix}$$

First, we compute the required sums from the data:

$$m = 6$$
$$\sum_{i=1}^{6} x_i = 8.9$$
$$\sum_{i=1}^{6} x_i^2 = 14.17$$
$$\sum_{i=1}^{6} x_i^3 = 24.023$$
$$\sum_{i=1}^{6} x_i^4 = 42.8629$$
$$\sum_{i=1}^{6} x_i^5 = 79.51919$$
$$\sum_{i=1}^{6} x_i^6 = 151.801$$

And the sums involving y_i:

$$\sum_{i=1}^{6} y_i = 14.58$$
$$\sum_{i=1}^{6} x_i y_i = 22.808$$
$$\sum_{i=1}^{6} x_i^2 y_i = 38.0602$$
$$\sum_{i=1}^{6} x_i^3 y_i = 67.20082$$

Substituting these values gives the system:

$$\begin{pmatrix} 6 & 8.9 & 14.17 & 24.023 \\ 8.9 & 14.17 & 24.023 & 42.8629 \\ 14.17 & 24.023 & 42.8629 & 79.51919 \\ 24.023 & 42.8629 & 79.51919 & 151.801 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 14.58 \\ 22.808 \\ 38.0602 \\ 67.20082 \end{pmatrix}$$

Solving this system of linear equations yields the coefficients:

$$a_0 \approx 1.8643$$
$$a_1 \approx -1.0211$$
$$a_2 \approx 1.2949$$
$$a_3 \approx -0.2238$$

Thus, the least squares polynomial of degree 3 is:

$$P_3(x) = -0.2238x^3 + 1.2949x^2 - 1.0211x + 1.8643$$

To compute the error E, we find the sum of the squared differences between the actual y_i values and the values predicted by our polynomial, $P_3(x_i)$.

$$E = \sum_{i=1}^{6} (y_i - P_3(x_i))^2$$

The values are calculated in the following table:

x_i	y_i	$P_3(x_i)$	$y_i - P_3(x_i)$	$(y_i - P_3(x_i))^2$
1.0	*1.84*	*1.9143*	*-0.0743*	*0.00552*
1.1	*1.96*	*2.0100*	*-0.0500*	*0.00250*
1.3	*2.21*	*2.2336*	*-0.0236*	*0.00056*
1.5	*2.45*	*2.4908*	*-0.0408*	*0.00166*
1.9	*2.94*	*3.0638*	*-0.1238*	*0.01533*
2.1	*3.18*	*3.3579*	*-0.1779*	*0.03165*

Summing the final column gives the error E:

$$E = 0.00552 + 0.00250 + 0.00056 + 0.00166 + 0.01533 + 0.03165$$

$$E \approx 0.05722$$

5.5. Properties of Least Squares Approximation

The **design matrix**, often denoted by $\mathbf{A}$ or $\mathbf{X}$, is a matrix that organizes the independent variables of a model. Each row corresponds to a single observation (e.g., a data point (x_i, y_i)), and each column corresponds to a basis function or predictor variable (e.g., $1, x, x^2, \ldots$). This structure allows the entire system of equations for the data to be expressed concisely in the matrix form $\mathbf{Aa} \approx \mathbf{y}$.

Orthogonality Property: The least squares solution is characterized by the property that the residual vector, $\mathbf{e} = \mathbf{y} - \mathbf{A}\hat{\mathbf{a}}$, is orthogonal to the column space of the design matrix $\mathbf{A}$. This geometric condition implies that the error vector is perpendicular to each of the basis functions used for the approximation. The orthogonality condition, expressed as $\mathbf{A}^T\mathbf{e} = \mathbf{0}$, is mathematically equivalent to the normal equations.

Uniqueness: If the columns of the design matrix $\mathbf{A}$ are linearly independent, the matrix has full column rank. This ensures that the matrix $\mathbf{A}^T\mathbf{A}$ is positive definite and therefore invertible. Consequently, the least squares solution $\hat{\mathbf{a}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$ is unique.

Statistical Properties: According to the Gauss-Markov theorem, if the random errors have a mean of zero, are uncorrelated, and possess constant variance (homoscedasticity), then the ordinary least squares estimator is the Best Linear Unbiased Estimator (BLUE). This signifies that among all estimators that are linear combinations of the observations and are unbiased, the least squares estimator has the minimum variance.

5.6. Applications and Extensions

- **Weighted Least Squares:** This method extends ordinary least squares to cases where observations have non-constant variance (heteroscedasticity). By minimizing a weighted sum of squared residuals, $\sum w_i(y_i - f(\mathbf{x}_i))^2$, the method assigns greater importance to observations with higher precision (smaller variance). The weights are typically chosen to be inversely proportional to the variance of the errors.
- **Nonlinear Least Squares:** This is an iterative approach used for fitting a model that is a nonlinear function of its parameters. Since no closed-form solution for the parameters exists, numerical optimization algorithms like the Gauss-Newton or Levenberg-Marquardt algorithm are employed. These algorithms progressively refine parameter estimates to find the values that minimize the sum of the squared residuals.

- **Regularized Least Squares:** Regularization is a technique used to address ill-posed problems, multicollinearity, and to prevent overfitting by adding a penalty term to the objective function. **Ridge regression** adds an L_2 penalty ($\lambda \sum a_k^2$) which shrinks coefficients towards zero. **Lasso** regression adds an L_1 penalty ($\lambda \sum |a_k|$) which can force some coefficients to become exactly zero, thereby performing feature selection.
- **Robust Regression:** This class of methods is designed to be less sensitive to outliers than ordinary least squares. Because the sum of squared residuals gives substantial weight to large deviations, outliers can heavily influence the solution. Robust methods, such as M-estimation or Least Absolute Deviations, modify the objective function to reduce the influence of such extreme data points.

The least squares method is fundamental in statistics, data analysis, and numerical approximation, providing a principled way to fit models to data in the presence of noise and measurement errors.

5.7. Discrete Least Squares for Polynomials

Motivation: Given a set of data points $\{x_i, y_i\}_{i=1}^m$, we seek to find an algebraic polynomial of degree $n < m - 1$ that best fits the data. This is particularly useful when the data contains experimental error, where an exact interpolation would be misleading.

The approximating polynomial has the form:

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

The goal is to choose the constants $a_0, a_1, \ldots, a_n$ to minimize the sum of the squared errors, E.

$$\begin{aligned} E(a_0, \ldots, a_n) &= \sum_{i=1}^m (y_i - P_n(x_i))^2 \\ &= \sum_{i=1}^m \left(y_i - \sum_{j=0}^n a_j x_i^j \right)^2 \end{aligned}$$

5.7.1. Derivation of the Normal Equations

For the error E to be minimized, the partial derivative with respect to each coefficient a_j must be zero.

$$\frac{\partial E}{\partial a_j} = 0, \quad \text{for each } j = 0, 1, \ldots, n$$

Taking the derivative yields:

$$\frac{\partial E}{\partial a_j} = \sum_{i=1}^m 2 \left(y_i - \sum_{k=0}^n a_k x_i^k \right) (-x_i^j) = 0$$

Rearranging the terms, we arrive at the **normal equations**:

$$\sum_{k=0}^n a_k \left(\sum_{i=1}^m x_i^{j+k} \right) = \sum_{i=1}^m y_i x_i^j, \quad \text{for } j = 0, 1, \ldots, n$$

This is a system of $n + 1$ linear equations for the $n + 1$ unknown coefficients $a_0, \ldots, a_n$. Expanded, the system looks like this:

$$\begin{aligned} a_0 \sum x_i^0 + a_1 \sum x_i^1 + \cdots + a_n \sum x_i^n &= \sum y_i x_i^0 \\ a_0 \sum x_i^1 + a_1 \sum x_i^2 + \cdots + a_n \sum x_i^{n+1} &= \sum y_i x_i^1 \\ &\vdots \\ a_0 \sum x_i^n + a_1 \sum x_i^{n+1} + \cdots + a_n \sum x_i^{2n} &= \sum y_i x_i^n \end{aligned}$$

This system has a unique solution, provided that the x_i values are distinct.

Example 5.7.1. *Fit the following data with a discrete least squares polynomial of degree at most 2.*

x_i	0	0.25	0.50	0.75	1.00
y_i	1.00	1.2840	1.6487	2.1170	2.7183

Solution: *The resulting polynomial is* $P_2(x) = 1.0051 + 0.8647x + 0.8432x^2$.

5.7.2. Linearization for Non-Polynomial Models

Sometimes data is better modeled by a non-polynomial function, such as an exponential relationship $y = be^{ax}$. A direct application of least squares leads to nonlinear normal equations for a and b, which are difficult to solve.

$$\frac{\partial E}{\partial b} = -2\sum(y_i - be^{ax_i})(e^{ax_i}) = 0$$
$$\frac{\partial E}{\partial a} = -2\sum(y_i - be^{ax_i})(bx_ie^{ax_i}) = 0$$

A common strategy is to linearize the model by taking the natural logarithm:

$$\ln y = \ln b + ax$$

Letting $Y = \ln y$, $a_0 = \ln b$, and $a_1 = a$, the problem is transformed into finding the best-fit line $Y = a_1x + a_0$, which can be solved using the standard linear least squares method.

5.8. Continuous Least Squares for Functions

The principle of least squares can be extended to approximate a continuous function $f \in C[a, b]$ with a polynomial $P_n(x)$. Instead of summing over discrete data points, we integrate over the interval $[a, b]$.

The error E is defined as:

$$E(a_0, \ldots, a_n) = \int_a^b [f(x) - P_n(x)]^2 \, dx = \int_a^b \left(f(x) - \sum_{k=0}^{n} a_kx^k\right)^2 dx$$

Minimizing E by setting $\frac{\partial E}{\partial a_j} = 0$ gives the continuous normal equations:

$$\sum_{k=0}^{n} a_k \int_a^b x^{j+k} dx = \int_a^b x^j f(x) dx, \quad \text{for each } j = 0, 1, \ldots n$$

Example 5.8.1. *Find the least squares approximating polynomial of degree 2 for the function* $f(x) = \sin(\pi x)$ *on the interval* $[0, 1]$. ***Solution:*** *The normal equations for* $P_2(x) = a_2x^2 + a_1x + a_0$ *are:*

$$a_0 \int_0^1 1dx + a_1 \int_0^1 xdx + a_2 \int_0^1 x^2dx = \int_0^1 \sin(\pi x)dx$$
$$a_0 \int_0^1 xdx + a_1 \int_0^1 x^2dx + a_2 \int_0^1 x^3dx = \int_0^1 x\sin(\pi x)dx$$
$$a_0 \int_0^1 x^2dx + a_1 \int_0^1 x^3dx + a_2 \int_0^1 x^4dx = \int_0^1 x^2\sin(\pi x)dx$$

Solving the integrals gives a 3×3 *system of linear equations, which yields the solution:* $a_0 \approx -0.050465$, $a_1 \approx 4.12251$, *and* $a_2 \approx -4.12251$. *Thus,*

$$P_2(x) = -4.12251x^2 + 4.12251x - 0.050465$$

5.9. Orthogonal Polynomials and Least Squares

Solving the system of normal equations can be computationally expensive and prone to ill-conditioning. Using a basis of orthogonal functions simplifies the problem immensely.

5.9.1. Linearly Independent and Orthogonal Functions

Definition (Linear Independence): A set of functions $\{\phi_0, \ldots, \phi_n\}$ is linearly independent on $[a, b]$ if the condition $c_0\phi_0(x) + \cdots + c_n\phi_n(x) = 0$ for all $x \in [a, b]$ implies that $c_0 = c_1 = \cdots = c_n = 0$.

Theorem 5.9.1. *Suppose that for each $j = 0, 1, \ldots, n$, $\phi_j(x)$ is a polynomial of degree exactly j. Then the set of polynomials $\{\phi_0, \phi_1, \ldots, \phi_n\}$ is linearly independent on any interval $[a, b]$.*

Proof. We will prove this by contradiction. Assume that the set $\{\phi_0, \phi_1, \ldots, \phi_n\}$ is linearly dependent. By definition of linear dependence, there must exist a set of constants $c_0, c_1, \ldots, c_n$, with at least one constant being non-zero, such that the following equation holds for all $x \in [a, b]$:

$$c_0\phi_0(x) + c_1\phi_1(x) + \cdots + c_{n-1}\phi_{n-1}(x) + c_n\phi_n(x) = 0 \tag{5.8}$$

Let's differentiate this equation n **times** with respect to x.

Since each $\phi_j(x)$ is a polynomial of degree j, its n-th derivative will be zero if $j < n$. After differentiating Equation (5.13) n times, all terms for $j = 0, 1, \ldots, n-1$ will vanish:

$$\frac{d^n}{dx^n}(c_j\phi_j(x)) = 0, \quad \text{for } j < n$$

The only term that may not vanish is the one involving $\phi_n(x)$. Let the polynomial $\phi_n(x)$ be written as $\phi_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$, where $a_n \neq 0$ since it is of degree n.

The n-th derivative of $\phi_n(x)$ is:

$$\frac{d^n}{dx^n}\phi_n(x) = a_n \cdot n!$$

This is a non-zero constant. After differentiating Equation (5.13) n times, we are left with:

$$c_n \frac{d^n}{dx^n}\phi_n(x) = c_n(a_n \cdot n!) = 0$$

Since $a_n \neq 0$ and $n! \neq 0$, the only way for this equation to be true is if $c_n = 0$.

Now, substituting $c_n = 0$ back into Equation (5.13), we get:

$$c_0\phi_0(x) + c_1\phi_1(x) + \cdots + c_{n-1}\phi_{n-1}(x) = 0$$

We can repeat the process. If we differentiate this new equation $n - 1$ times, we will similarly find that c_{n-1} must be zero.

Continuing this process iteratively, we conclude that all coefficients must be zero:

$$c_n = c_{n-1} = \cdots = c_1 = c_0 = 0$$

This contradicts our initial assumption that at least one of the constants was non-zero. Therefore, the assumption of linear dependence must be false.

Thus, the set $\{\phi_0, \phi_1, \ldots, \phi_n\}$ is **linearly independent**. □

Example 5.9.2. *Let $\phi_0(x) = 2$, $\phi_1(x) = x - 3$, and $\phi_2(x) = x^2 + 2x + 7$, and $Q(x) = a_0 + a_1x + a_2x^2$. Show that there exist constants c_0, c_1, and c_2 such that $Q(x) = c_0\phi_0(x) + c_1\phi_1(x) + c_2\phi_2(x)$.*

Solution : *Here $\{\phi_0, \phi_1, \phi_2\}$ is linearly independent on any interval $[a, b]$. First note that*

$$1 = \frac{1}{2}\phi_0(x), \quad x = \phi_1(x) + 3 = \phi_1(x) + \frac{3}{2}\phi_0(x), \textit{and}$$

$$x^2 = \phi_2(x) - 2x - 7 = \phi_2(x) - 2\left[\phi_1(x) + \frac{3}{2}\phi_0(x)\right] - 7\left[\frac{1}{2}\phi_0(x)\right] \tag{5.9}$$

$$= \phi_2(x) - 2\phi_1(x) - \frac{13}{2}\phi_0(x). \tag{5.10}$$

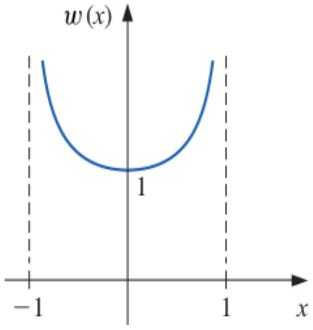
$$Q(x) = a_0\left[\frac{1}{2}\phi_0(x)\right] + a_1\left[\phi_1(x) + \frac{3}{2}\phi_0(x)\right] + a_2\left[\phi_2(x) - 2\phi_1(x) - \frac{13}{2}\phi_0(x)\right] \tag{5.11}$$

$$= \left(\frac{1}{2}a_0 + \frac{3}{2}a_1 - \frac{13}{2}a_2\right)\phi_0(x) + [a_1 - 2a_2]\phi_1(x) + a_2\phi_2(x). \tag{5.12}$$

Definition (Weight Function): An integrable function $w(x)$ is a weight function on $[a,b]$ if $w(x) \geq 0$ for all $x \in [a,b]$ and is not zero on any subinterval.

The purpose of an weight function is to assign varying degrees of importance to approximations on certain portions of the interval.

Example 5.9.3. $w(x) = \frac{1}{\sqrt{1-x^2}}$ *places less emphasis near the center of the interval* $(-1,1)$*, and more when* $|x|$ *is near 1.*



Definition (Orthogonality): A set of functions $\{\phi_0, \ldots, \phi_n\}$ is an **orthogonal set** on $[a,b]$ with respect to the weight function $w(x)$ if

$$\int_a^b w(x)\phi_k(x)\phi_j(x)dx = \begin{cases} 0, & \text{when } j \neq k, \\ \alpha_j > 0, & \text{when } j = k \end{cases}$$

If $\alpha_j = 1$ for all j, the set is called **orthonormal**.

5.9.2. Approximation with Orthogonal Functions

If we approximate $f(x)$ with a linear combination of orthogonal functions, $P(x) = \sum_{k=0}^{n} a_k\phi_k(x)$, the error to minimize is:

$$E = \int_a^b w(x)\left[f(x) - \sum_{k=0}^{n} a_k\phi_k(x)\right]^2 dx$$

The normal equations become:

$$\sum_{k=0}^{n} a_k \int_a^b w(x)\phi_k(x)\phi_j(x)dx = \int_a^b w(x)f(x)\phi_j(x)dx$$

Due to orthogonality, the sum on the left simplifies because the integral is zero for all $k \neq j$. This leaves:

$$a_j \int_a^b w(x)[\phi_j(x)]^2 dx = \int_a^b w(x)f(x)\phi_j(x)dx$$

This diagonalizes the system, and each coefficient can be solved for independently:

$$a_j = \frac{\int_a^b w(x)f(x)\phi_j(x)dx}{\int_a^b w(x)[\phi_j(x)]^2 dx} = \frac{1}{\alpha_j}\int_a^b w(x)f(x)\phi_j(x)dx$$

Example 5.9.4. *Find a linear polynomial approximation to $f(x) = x^3$ on the interval $[0,1]$ using the least squares approximation with $w(x) = 1$.*

Solution : *We are looking for a linear polynomial of the form $P_1(x) = a_1x + a_0$ that minimizes the least squares error integral:*

$$E(a_0, a_1) = \int_0^1 [f(x) - P_1(x)]^2 dx = \int_0^1 [x^3 - (a_1x + a_0)]^2 dx$$

The normal equations for the continuous least squares problem are given by:

$$\begin{aligned} a_0 \int_0^1 1 \cdot 1 \, dx + a_1 \int_0^1 x \cdot 1 \, dx &= \int_0^1 1 \cdot f(x) \, dx \\ a_0 \int_0^1 1 \cdot x \, dx + a_1 \int_0^1 x \cdot x \, dx &= \int_0^1 x \cdot f(x) \, dx \end{aligned}$$

First, we compute the required integrals.

$$\begin{aligned} \int_0^1 1 \, dx &= [x]_0^1 = 1 \\ \int_0^1 x \, dx &= \left[\frac{x^2}{2}\right]_0^1 = \frac{1}{2} \\ \int_0^1 x^2 \, dx &= \left[\frac{x^3}{3}\right]_0^1 = \frac{1}{3} \end{aligned}$$

Next, we compute the integrals involving $f(x) = x^3$.

$$\begin{aligned} \int_0^1 f(x) \, dx &= \int_0^1 x^3 \, dx = \left[\frac{x^4}{4}\right]_0^1 = \frac{1}{4} \\ \int_0^1 x f(x) \, dx &= \int_0^1 x \cdot x^3 \, dx = \int_0^1 x^4 \, dx = \left[\frac{x^5}{5}\right]_0^1 = \frac{1}{5} \end{aligned}$$

Now, we substitute these values back into the normal equations:

$$a_0(1) + a_1\left(\frac{1}{2}\right) = \frac{1}{4} \tag{5.13}$$

$$a_0\left(\frac{1}{2}\right) + a_1\left(\frac{1}{3}\right) = \frac{1}{5} \tag{5.14}$$

This is a system of two linear equations. From equation (5.13)*, we can express a_0 in terms of a_1:*

$$a_0 = \frac{1}{4} - \frac{1}{2}a_1$$

Substitute this into equation (5.14)*:*

$$\begin{aligned} \frac{1}{2}\left(\frac{1}{4} - \frac{1}{2}a_1\right) + \frac{1}{3}a_1 &= \frac{1}{5} \\ \frac{1}{8} - \frac{1}{4}a_1 + \frac{1}{3}a_1 &= \frac{1}{5} \\ \left(\frac{-3+4}{12}\right)a_1 &= \frac{1}{5} - \frac{1}{8} \\ \frac{1}{12}a_1 &= \frac{8-5}{40} = \frac{3}{40} \\ a_1 &= \frac{12 \cdot 3}{40} = \frac{36}{40} = \frac{9}{10} \end{aligned}$$

Now, we find a_0:

$$a_0 = \frac{1}{4} - \frac{1}{2}\left(\frac{9}{10}\right) = \frac{1}{4} - \frac{9}{20} = \frac{5}{20} - \frac{9}{20} = -\frac{4}{20} = -\frac{1}{5}$$

Thus, the linear least squares polynomial approximation is:

$$P_1(x) = \frac{9}{10}x - \frac{1}{5}$$

Practice Questions

1. The following list contains homework grades and the final-examination grades for 30 numerical analysis students. Find the equation of the least squares line for this data, and use this line to determine the homework grade required to predict minimal A (90%) and D (60%) grades on the final.

Homework	**Final**	**Homework**	**Final**
302	45	323	83
325	72	337	99
285	54	337	70
339	54	304	62
334	79	319	66
322	65	234	51
331	99	337	53
279	63	351	100
316	65	339	67
347	99	343	83
343	83	314	42
290	74	344	79
326	76	185	59
233	57	340	75
254	45	316	45

2. Show that the normal equations resulting from discrete least squares approximation yield a symmetric and nonsingular matrix and hence have a unique solution.

   [*Hint:* Let $A = (a_{ij})$, where
$$a_{ij} = \sum_{k=1}^{m} x_k^{i+j-2}$$
   and $x_1, x_2, \ldots, x_m$ are distinct with $n < m - 1$. Suppose A is singular and that $\mathbf{c} \neq \mathbf{0}$ is such that $\mathbf{c}^T A \mathbf{c} = 0$. Show that the nth-degree polynomial whose coefficients are the coordinates of $\mathbf{c}$ has more than n roots, and use this to establish a contradiction.]

3. Find the linear least squares polynomial approximation on the interval $[-1, 1]$ for the following functions.

   a. $f(x) = x^2 - 2x + 3$

   b. $f(x) = x^3$

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

   d. $f(x) = e^x$

   e. $f(x) = \frac{1}{2}\cos(x) + \frac{1}{3}\sin(2x)$

   f. $f(x) = \ln(x + 2)$

4. Suppose $\{\phi_0, \phi_1, \ldots, \phi_n\}$ is any linearly independent set in Π_n. Show that for any element $Q \in \Pi_n$, there exist unique constants $c_0, c_1, \ldots, c_n$ such that
$$Q(x) = \sum_{k=0}^{n} c_k \phi_k(x).$$

5. Given the data:

x_i	4.0	4.2	4.5	4.7	5.1	5.5	5.9	6.3	6.8	7.1
y_i	102.56	113.18	130.11	142.05	167.53	195.14	224.87	256.73	299.50	326.72

Table 5.2: Data values of x_i and y_i.

a. Construct the least squares polynomial of degree 1, and compute the error.

b. Construct the least squares polynomial of degree 2, and compute the error.

c. Construct the least squares polynomial of degree 3, and compute the error.

d. Construct the least squares approximation of the form be^{ax}, and compute the error.

e. Construct the least squares approximation of the form bx^a, and compute the error.

Numerical Differentiation

6.1. Introduction to Numerical Differentiation

Numerical differentiation is the process of finding numerical approximations for the derivative of a function. The derivative represents the instantaneous rate of change, but in practice, we often only have function values at a set of discrete points. The goal is to use these discrete values to estimate the derivative.

The foundation for these methods is the Taylor series expansion of a function $f(x)$ about a point x_j:

$$f(x) = f(x_j) + f'(x_j)(x - x_j) + \frac{f''(x_j)}{2!}(x - x_j)^2 + \frac{f'''(x_j)}{3!}(x - x_j)^3 + \cdots \tag{6.1}$$

6.2. First-Derivative Formulas from Taylor Series

We can derive several formulas by truncating the Taylor series at different orders. We will use the notation $f_j = f(x_j)$ and assume a constant step size h between points.

6.2.1. Forward Difference Formula

Let $x = x_{j+1}$ and define the step size as $h = x_{j+1} - x_j$. Substituting this into the Taylor series gives:

$$f(x_{j+1}) = f(x_j) + f'(x_j)h + \frac{f''(x_j)}{2!}h^2 + \frac{f'''(x_j)}{3!}h^3 + \cdots$$

Solving for $f'(x_j)$, which we denote as f'_j:

$$f'_j = \frac{f_{j+1} - f_j}{h} - \frac{h}{2}f''(x_j) - \frac{h^2}{6}f'''(x_j) - \cdots$$

By truncating the series after the first term, we obtain the **forward difference formula**, which has an error of order h, denoted $O(h)$.

$$f'_j = \frac{f_{j+1} - f_j}{h} + O(h)$$

The term $f_{j+1} - f_j$ is known as the first forward difference, Δf_j.

6.2.2. Backward Difference Formula

Similarly, we can evaluate the Taylor series at $x = x_{j-1}$, with $h = x_j - x_{j-1}$. This leads to the **backward difference formula**:

$$f'_j = \frac{f_j - f_{j-1}}{h} + O(h)$$

The term $f_j - f_{j-1}$ is the first backward difference, ∇f_j.

6.2.3. Central Difference Formula

A more accurate formula can be derived by subtracting the Taylor expansion for f_{j-1} from the expansion for f_{j+1}:

$$f_{j+1} = f_j + hf_j' + \frac{h^2}{2}f_j'' + \frac{h^3}{6}f_j''' + \cdots$$
$$f_{j-1} = f_j - hf_j' + \frac{h^2}{2}f_j'' - \frac{h^3}{6}f_j''' + \cdots$$

Subtracting the second equation from the first cancels the even-order terms:

$$f_{j+1} - f_{j-1} = 2hf_j' + \frac{h^3}{3}f_j''' + \cdots$$

Solving for f_j' gives the **central difference formula**. Note that the leading error term is of order h^2, making this formula significantly more accurate for small h.

$$f_j' = \frac{f_{j+1} - f_{j-1}}{2h} + O(h^2)$$

Example 6.2.1. *Let $f(x) = x^3$. Find an approximation for $f'(2)$ using a step size of $h = 0.1$. Compare the forward, backward, and central difference formulas.*

Solution: *The true value is $f'(x) = 3x^2$, so $f'(2) = 3(2^2) = 12$. We need the function values at $x = 1.9, 2.0,$ and 2.1.*

- $f(1.9) = 1.9^3 = 6.859$
- $f(2.0) = 2.0^3 = 8.000$
- $f(2.1) = 2.1^3 = 9.261$

Now we apply the formulas:

- ***Forward Difference:*** $f'(2) \approx \frac{f(2.1)-f(2.0)}{0.1} = \frac{9.261-8.000}{0.1} = 12.61$. *(Error = 0.61)*
- ***Backward Difference:*** $f'(2) \approx \frac{f(2.0)-f(1.9)}{0.1} = \frac{8.000-6.859}{0.1} = 11.41$. *(Error = -0.59)*
- ***Central Difference:*** $f'(2) \approx \frac{f(2.1)-f(1.9)}{2(0.1)} = \frac{9.261-6.859}{0.2} = 12.01$. *(Error = 0.01)*

This clearly shows the superior accuracy of the central difference formula.

6.3. Higher-Order Derivatives

Formulas for higher derivatives can be derived by combining more Taylor series expansions.

6.3.1. Second Derivatives

By combining the Taylor expansions for f_{j+1} and f_{j+2} around x_j, one can derive a forward difference formula for the second derivative:

$$f_j'' = \frac{f_{j+2} - 2f_{j+1} + f_j}{h^2} + O(h)$$

This is related to the second forward difference operator, $\Delta^2 f_j = f_{j+2} - 2f_{j+1} + f_j$. Similarly, a backward approximation is $f_j'' = \frac{\nabla^2 f_j}{h^2}$. A more accurate central difference formula for the second derivative is $f_j'' = \frac{f_{j+1}-2f_j+f_{j-1}}{h^2} + O(h^2)$.

Example 6.3.1. *Let $f(x) = x^3$. Find an approximation for $f''(2)$ using the central difference formula with $h = 0.1$.*

Solution: *The true value is $f''(x) = 6x$, so $f''(2) = 6(2) = 12$. We use the same function values as the previous example. The central difference formula for the second derivative is:*

$$f''_j \approx \frac{f_{j+1} - 2f_j + f_{j-1}}{h^2}$$

Substituting our values:

$$f''(2) \approx \frac{f(2.1) - 2f(2.0) + f(1.9)}{(0.1)^2} = \frac{9.261 - 2(8.000) + 6.859}{0.01} = \frac{0.12}{0.01} = 12.00$$

The approximation is exact in this case because the higher-order error terms for this specific polynomial are zero.

6.3.2. Higher-Order Accuracy Formulas

By using more data points, one can construct formulas with a higher order of accuracy. For example, the **three-point forward difference formula** for the first derivative uses values at x_j, x_{j+1}, and x_{j+2} to achieve an error of $O(h^2)$:

$$f'_j = \frac{-3f_j + 4f_{j+1} - f_{j+2}}{2h} + O(h^2)$$

Example 6.3.2. *Consider the function $f(x) = e^{-x}\sin(x/2)$. We wish to calculate an approximation for $f'(1.4)$ using data points at $x = 1.2, 1.4, 1.6, 1.8$ (so $h = 0.2$). The true value is approximately $f'(1.4) \approx -0.06456$. The table below compares the results from different formulas.*

Difference Formula	***Approximate*** $f'(1.4)$	***% Relative Error***
Two-point backward	$\frac{f(1.4)-f(1.2)}{0.2} = -0.0560$	*13.22%*
Two-point forward	$\frac{f(1.6)-f(1.4)}{0.2} = -0.0702$	*8.66%*
Two-point central	$\frac{f(1.6)-f(1.2)}{2(0.2)} = -0.0631$	*2.28%*
Three-point forward	$\frac{-3f(1.4)+4f(1.6)-f(1.8)}{2(0.2)} = -0.0669$	*3.56%*

As expected, the central difference formula, with its $O(h^2)$ accuracy, provides the best estimate among the two-point formulas.

Numerical Integration

7.1. Introduction to Numerical Integration (Quadrature)

Numerical integration, also known as **quadrature**, provides methods for approximating definite integrals. This is essential in two main scenarios:

- When the function to be integrated, $f(x)$, does not have an explicit or easily found antiderivative (e.g., $f(x) = e^{-x^2}$).
- When the function is only known as a set of discrete data points.

The core strategy is to replace the function $f(x)$ with a simpler function that is easy to integrate, typically an interpolating polynomial $P_n(x)$. We select a set of distinct nodes $\{x_0, x_1, \ldots, x_n\}$ within the interval $[a, b]$ and construct the Lagrange interpolating polynomial:

$$P_n(x) = \sum_{i=0}^{n} f(x_i) L_i(x)$$

The integral of $f(x)$ is then approximated by the integral of $P_n(x)$.

The integral of $f(x)$ can be expressed as the sum of the integral of the polynomial and an error term:

$$\begin{aligned}\int_a^b f(x)dx &= \int_a^b \left(\sum_{i=0}^{n} f(x_i) L_i(x) \right) dx + \text{Error} \\ &= \sum_{i=0}^{n} \left(\int_a^b L_i(x)dx \right) f(x_i) + \text{Error}\end{aligned}$$

This leads to the general form of a quadrature formula:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} a_i f(x_i)$$

where the weights are defined as $a_i = \int_a^b L_i(x)dx$. The error term is given by:

$$E(f) = \frac{1}{(n+1)!} \int_a^b \prod_{i=0}^{n} (x - x_i) f^{(n+1)}(\zeta(x))dx$$

7.2. Newton-Cotes Formulas

Newton-Cotes formulas are a family of quadrature rules that use equally spaced nodes. They are categorized as either "closed" (including the endpoints) or "open" (using only interior points).

7.2.1. Closed Newton-Cotes Formulas

These formulas use nodes $x_i = a + ih$ for $i = 0, 1, \ldots, n$, where $h = (b - a)/n$.It is called closed as the endpoints are included.

The Trapezoidal Rule (n=1)

The Trapezoidal rule approximates the integral $\int_a^b f(x)dx$ by integrating the first-degree Lagrange polynomial $P_1(x)$ through the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$, where $x_0 = a$, $x_1 = b$, and $h = b - a$.

$$\begin{aligned}\int_a^b f(x)dx \approx \int_{x_0}^{x_1} P_1(x)dx &= \int_{x_0}^{x_1} \left[\frac{x-x_1}{x_0-x_1}f(x_0) + \frac{x-x_0}{x_1-x_0}f(x_1)\right] dx \\ &= f(x_0)\int_{x_0}^{x_1} \frac{x-x_1}{-h}dx + f(x_1)\int_{x_0}^{x_1} \frac{x-x_0}{h}dx \\ &= f(x_0)\left[-\frac{(x-x_1)^2}{2h}\right]_{x_0}^{x_1} + f(x_1)\left[\frac{(x-x_0)^2}{2h}\right]_{x_0}^{x_1} \\ &= f(x_0)\left(\frac{h}{2}\right) + f(x_1)\left(\frac{h}{2}\right) = \frac{h}{2}[f(x_0) + f(x_1)]\end{aligned}$$

Error Term Derivation:

The error in Lagrange interpolation is $f(x) - P_1(x) = \frac{f''(\zeta_x)}{2!}(x - x_0)(x - x_1)$. Integrating this gives the error of the quadrature rule.

$$E(f) = \int_{x_0}^{x_1} \frac{f''(\zeta_x)}{2}(x - x_0)(x - x_1)dx$$

Since $(x - x_0)(x - x_1)$ does not change sign on $[x_0, x_1]$, we can apply the Weighted Mean Value Theorem for Integrals.

$$\begin{aligned}E(f) &= \frac{f''(\zeta)}{2}\int_{x_0}^{x_1}(x - x_0)(x - x_1)dx \quad \text{for some } \zeta \in (x_0, x_1) \\ &= \frac{f''(\zeta)}{2}\left[\frac{x^3}{3} - \frac{(x_0+x_1)x^2}{2} + x_0x_1x\right]_{x_0}^{x_1} \\ &= \frac{f''(\zeta)}{2}\left(-\frac{h^3}{6}\right) = -\frac{h^3}{12}f''(\zeta)\end{aligned}$$

Example 7.2.1. *Use the Trapezoidal rule to approximate $\int_1^2 x^2 dx$.*

Solution: *The exact value is $\int_1^2 x^2dx = [\frac{x^3}{3}]_1^2 = \frac{8}{3} - \frac{1}{3} = \frac{7}{3} \approx 2.3333$.*
Here, $a = 1, b = 2$, so $h = 1$.
The function is $f(x) = x^2$.

$$\int_1^2 x^2dx \approx \frac{1}{2}[f(1) + f(2)] = \frac{1}{2}[1^2 + 2^2] = \frac{1}{2}[1 + 4] = 2.5$$

The error is significant because we are approximating a parabola with a single straight line.

Simpson's 1/3 Rule (n=2)

Simpson's rule approximates $\int_a^b f(x)dx$ by integrating the second-degree Lagrange polynomial $P_2(x)$ through three equally spaced nodes: $x_0 = a$, $x_1 = a + h$, and $x_2 = b = a + 2h$.

$$\int_a^b f(x)dx \approx \int_{x_0}^{x_2} [f(x_0)L_0(x) + f(x_1)L_1(x) + f(x_2)L_2(x)]\, dx$$

By integrating the Lagrange basis polynomials (a detailed but straightforward process), we find the weights:

$$\int_{x_0}^{x_2} L_0(x)dx = \frac{h}{3}, \quad \int_{x_0}^{x_2} L_1(x)dx = \frac{4h}{3}, \quad \int_{x_0}^{x_2} L_2(x)dx = \frac{h}{3}$$

Substituting these weights gives the final formula:

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)]$$

Error Term Derivation

The error derivation is more complex and typically uses a cubic Hermite interpolating polynomial. The error term for this higher-order polynomial is $f(x) - H_3(x) = \frac{f^{(4)}(\zeta_x)}{4!}(x - x_0)(x - x_1)^2(x - x_2)$.

$$E(f) = \int_{x_0}^{x_2} \frac{f^{(4)}(\zeta_x)}{24}(x - x_0)(x - x_1)^2(x - x_2)dx$$

The polynomial term $(x - x_0)(x - x_1)^2(x - x_2)$ is non-positive on $[x_0, x_2]$, so we can again apply the Weighted Mean Value Theorem.

$$E(f) = \frac{f^{(4)}(\zeta)}{24} \int_{x_0}^{x_2} (x - x_0)(x - x_1)^2(x - x_2)dx \quad \text{for some } \zeta \in (x_0, x_2)$$

With a change of variable $u = x - x_1$, the integral becomes:

$$\int_{-h}^{h} (u + h)u^2(u - h)du = \int_{-h}^{h} (u^4 - h^2u^2)du = \left[\frac{u^5}{5} - \frac{h^2u^3}{3}\right]_{-h}^{h} = -\frac{4h^5}{15}$$

Substituting this result gives the error term:

$$E(f) = \frac{f^{(4)}(\zeta)}{24}\left(-\frac{4h^5}{15}\right) = -\frac{h^5}{90}f^{(4)}(\zeta)$$

Because the error depends on the fourth derivative, the rule has a degree of accuracy of 3.

Example 7.2.2. *Compare the Trapezoidal and Simpson's rule approximations for $\int_0^2 f(x)dx$ for $f(x) = x^2$ and $f(x) = x^4$.*

Solution: *For both rules, $h = 1$ for Simpson's ($x_0 = 0, x_1 = 1, x_2 = 2$) and $h = 2$ for Trapezoidal ($x_0 = 0, x_1 = 2$).*

Method	***Approximation for*** $f(x) = x^2$	***Approximation for*** $f(x) = x^4$
Trapezoidal	$\frac{2}{2}[f(0) + f(2)] = 0^2 + 2^2 = 4$	$\frac{2}{2}[0^4 + 2^4] = 16$
Simpson's	$\frac{1}{3}[f(0) + 4f(1) + f(2)] = \frac{1}{3}[0 + 4(1) + 4] = \frac{8}{3}$	$\frac{1}{3}[0 + 4(1) + 16] = \frac{20}{3}$
Exact Value	$\frac{8}{3} \approx 2.667$	$\frac{32}{5} = 6.4$

As expected, Simpson's rule is exact for $f(x) = x^2$ and provides a much better approximation for $f(x) = x^4$.

Other Closed Formulas :

- **Simpson's 3/8 Rule (n=3):** Uses four points and has a degree of accuracy of 3.

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] - \frac{3h^5}{80}f^{(4)}(\zeta)$$

- **Boole's Rule (n=4):** Uses five points and has a degree of accuracy of 5.

$$\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] - \frac{8h^7}{945}f^{(6)}(\zeta)$$

7.2.2. Degree of Accuracy

The **degree of accuracy**, also known as the degree of precision, of a quadrature formula is a measure of its sophistication and indicates the class of polynomials for which the rule is perfectly accurate.

Definition: The degree of accuracy of a quadrature formula is the largest positive integer n such that the formula is exact for all polynomials of degree $k \leq n$.

In other words, a rule has a degree of accuracy n if its error is zero for any polynomial of degree up to n, but is not zero for some polynomial of degree $n + 1$. This is directly related to the derivative in the error term of the formula.

- **Trapezoidal Rule:** The error term for the Trapezoidal rule is $E(f) = -\frac{h^3}{12} f''(\zeta)$. Since the second derivative of any linear polynomial ($P_1(x) = c_1 x + c_0$) is zero, the error term vanishes for all polynomials of degree 1 or less. Therefore, the Trapezoidal rule has a **degree of accuracy of 1**.
- **Simpson's 1/3 Rule:** The error term for Simpson's rule is $E(f) = -\frac{h^5}{90} f^{(4)}(\zeta)$. The fourth derivative of any cubic polynomial ($P_3(x) = c_3 x^3 + c_2 x^2 + c_1 x + c_0$) is zero, making the rule exact for all polynomials of degree 3 or less. Therefore, Simpson's rule has a **degree of accuracy of 3**.

7.2.3. Open Newton-Cotes Formulas

These formulas use nodes $x_i = a + (i + 1)h$ for $i = 0, 1, \ldots, n$, where $h = (b - a)/(n + 2)$. They do not use the endpoints a and b, which is useful for integrals where the function is singular at the endpoints.

- **Midpoint Rule (n=0):** This is the simplest open formula.

$$\int_{x_{-1}}^{x_1} f(x)dx = 2hf(x_0) + \frac{h^3}{3} f''(\zeta)$$

- **Two-Point Open Rule (n=1):**

$$\int_{x_{-1}}^{x_2} f(x)dx = \frac{3h}{2}[f(x_0) + f(x_1)] + \frac{3h^3}{4} f''(\zeta)$$

- **Three-Point Open Rule (n=2):**

$$\int_{x_{-1}}^{x_3} f(x)dx = \frac{4h}{3}[2f(x_0) - f(x_1) + 2f(x_2)] + \frac{14h^5}{45} f^{(4)}(\zeta)$$

- **Four-Point Open Rule (n=3):**

$$\int_{x_{-1}}^{x_4} f(x)dx = \frac{5h}{24}[11f(x_0) + f(x_1) + f(x_2) + 11f(x_3)] + \frac{95h^5}{144} f^{(4)}(\zeta)$$

Example 7.2.3. *Compare the closed Trapezoidal rule (n=1) and open Midpoint rule (n=0) for* $\int_0^{\pi/4} \sin x dx \approx 0.29289322$.

Solution:

- ***Closed (Trapezoidal):*** $h = \pi/4$.

$$\int_0^{\pi/4} \sin x dx \approx \frac{\pi/4}{2}[\sin(0) + \sin(\pi/4)] \approx 0.27768$$

(Error ≈ 0.0152*)*

- ***Open (Midpoint):*** *We need to integrate from $x_{-1} = 0$ to $x_1 = \pi/4$. This implies $h = (b-a)/2 = \pi/8$, and the midpoint is $x_0 = \pi/8$.*
$$\int_0^{\pi/4} \sin x dx \approx 2hf(x_0) = 2(\pi/8)\sin(\pi/8) \approx 0.30056$$
(Error ≈ 0.0077)

In this case, the open Midpoint rule provides a more accurate result.

7.3. Composite Numerical Integration

To improve accuracy, we can divide the integration interval $[a, b]$ into n smaller subintervals and apply a simple quadrature rule to each one. This is the basis of composite rules.

7.3.1. Composite Trapezoidal Rule

For n subintervals, with $h = (b-a)/n$ and $x_j = a + jh$:

$$\int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right] - \frac{b-a}{12}h^2 f''(\mu)$$

7.3.2. Composite Simpson's Rule

For an even number of subintervals n, with $h = (b-a)/n$:

$$\int_a^b f(x)dx = \frac{h}{3}\left[f(a) + 2\sum_{j=1}^{(n/2)-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(b)\right] - \frac{b-a}{180}h^4 f^{(4)}(\mu)$$

The global error for the composite rules decreases as $O(h^2)$ for the Trapezoidal rule and $O(h^4)$ for Simpson's rule, making Simpson's rule much more efficient.

Example 7.3.1. *Determine the number of subintervals n required to approximate $\int_0^\pi \sin(x)dx$ to within 0.00002 using composite rules.*

Solution: *The exact value is 2. For $f(x) = \sin(x)$, $|f''(\mu)| \leq 1$ and $|f^{(4)}(\mu)| \leq 1$ for $\mu \in [0, \pi]$.*

- ***Composite Trapezoidal:*** *The error must satisfy $|\frac{\pi-0}{12}h^2 f''(\mu)| \leq \frac{\pi}{12}h^2 < 0.00002$. Since $h = \pi/n$, we have $\frac{\pi^3}{12n^2} < 0.00002$, which implies $n^2 > \frac{\pi^3}{12(0.00002)} \approx 129192.5$. Thus, $n > \sqrt{129192.5} \approx 359.4$. We must take $n = 360$.*
- ***Composite Simpson's:*** *The error must satisfy $|\frac{\pi-0}{180}h^4 f^{(4)}(\mu)| \leq \frac{\pi}{180}h^4 < 0.00002$. Since $h = \pi/n$, we have $\frac{\pi^5}{180n^4} < 0.00002$, which implies $n^4 > \frac{\pi^5}{180(0.00002)} \approx 85017$. Thus, $n > \sqrt[4]{85017} \approx 17.07$. Since n must be even, we must take $n = 18$.*

This demonstrates the superior efficiency of Simpson's rule.

7.4. Error Analysis and Optimal Step Size (h)

In numerical differentiation, a fundamental challenge is selecting an appropriate step size, h. While the approximation error, known as **truncation error**, is typically of the form Ch^p and thus decreases as $h \to 0$, another type of error, **round-off error**, behaves in the opposite manner.

7.4.1. The Trade-off Between Truncation and Round-off Error

Numerical differentiation formulas contain h in the denominator. As h becomes very small, we are forced to divide by a small number. This division amplifies any existing errors in the function values themselves. These initial errors are called round-off errors and are unavoidable in computation.

Specifically, if the function values are given in a table or computed with finite precision, they are not exact. We can model this by saying the computed value f_i is related to the true value $f(x_i)$ by:

$$f(x_i) = f_i + \epsilon_i$$

where ϵ_i is the round-off error associated with the i-th function value.

As h decreases, the truncation error gets smaller, but the round-off error gets larger. This means there is an optimal value of h that minimizes the total error, below which the results will actually worsen due to the dominance of round-off error.

7.4.2. Deriving the Optimal Step Size

Let's analyze this trade-off for the forward-difference formula:

$$f'(x_0) = \frac{f(x_1) - f(x_0)}{h} - \frac{h}{2} f''(\zeta)$$

We can express the computed derivative using the inexact function values f_0 and f_1:

$$\begin{aligned} f'(x_0) &= \frac{(f_1 + \epsilon_1) - (f_0 + \epsilon_0)}{h} - \frac{h}{2} f''(\zeta) \\ &= \frac{f_1 - f_0}{h} + \underbrace{\frac{\epsilon_1 - \epsilon_0}{h}}_{\text{Round-off Error (RE)}} - \underbrace{\frac{h}{2} f''(\zeta)}_{\text{Truncation Error (TE)}} \end{aligned}$$

The total error in our approximation is the sum of the Round-off Error and the Truncation Error. Let's find the magnitude of these errors. Let $\epsilon = \max(|\epsilon_0|, |\epsilon_1|)$ be the maximum round-off error in the function values, and let $M_2 = \max_{x \in [x_0, x_1]} |f''(x)|$.

- The bound on the round-off error is: $|RE| \leq \frac{|\epsilon_1| + |\epsilon_0|}{h} \leq \frac{2\epsilon}{h}$
- The bound on the truncation error is: $|TE| \leq \frac{h}{2} M_2$

The total error $E(h)$ is bounded by $E(h) \leq \frac{2\epsilon}{h} + \frac{h}{2} M_2$. To find the optimal h that minimizes this total error, we can use two equivalent criteria:

1. Balance the errors: $|RE| = |TE|$.
2. Minimize the sum: $|RE| + |TE|$ is a minimum.

Using the first criterion, we set the error bounds equal:

$$\begin{aligned} \frac{2\epsilon}{h} &= \frac{h}{2} M_2 \\ 4\epsilon &= h^2 M_2 \\ h^2 &= \frac{4\epsilon}{M_2} \end{aligned}$$

This gives the optimal step size, h_{optimal}:

$$h_{\text{adaptive}} = 2\sqrt{\frac{\epsilon}{M_2}}$$

The second criterion (minimizing the sum) yields the same result by taking the derivative of the total error with respect to h and setting it to zero.

We have

$$\frac{2\epsilon}{h} + \frac{h}{2}M_2 = \text{minimum}, \quad \Rightarrow -\frac{2\epsilon}{h^2} + \frac{1}{2}M_2 = 0 \tag{7.1}$$

Thus

$$h_{adaptive} = 2\sqrt{\epsilon/M_2}$$

Example 7.4.1. *For the formula* $f'(x_0) = \frac{-3f(x_0)+4f(x_1)-f(x_2)}{2h} + \frac{h^2}{3}f'''(\zeta)$*, determine the optimal step size* h*. Use* $f(x) = \ln(x)$ *near* $x = 2.0$*, with a maximum round-off error of* $\epsilon = 5 \times 10^{-6}$.

Solution: *The optimal step size* h *is found by balancing the round-off error (RE) and the truncation error (TE).*

- ***Establish Error Bounds:***
  - ***Round-off Error****: The magnitude is bounded by:*

$$|RE| \leq \frac{|-3\epsilon_0| + |4\epsilon_1| + |-\epsilon_2|}{2h} \leq \frac{8\epsilon}{2h} = \frac{4\epsilon}{h}$$

  - ***Truncation Error****: The magnitude is bounded by:*

$$|TE| \leq \frac{h^2}{3}M_3, \quad \textit{where } M_3 = \max|f'''(x)|$$

- ***Find*** M_3 ***for*** $f(x) = \ln(x)$***:*** *The third derivative is* $f'''(x) = 2/x^3$*. Near* $x = 2$*, we can approximate* $M_3 \approx |f'''(2)| = 2/2^3 = 0.25$.

- ***Calculate Optimal*** h***:*** *Set the error bounds equal to find the minimum total error:*

$$\frac{4\epsilon}{h} = \frac{h^2}{3}M_3$$
$$h^3 = \frac{12\epsilon}{M_3}$$
$$h_{optimal} = \sqrt[3]{\frac{12\epsilon}{M_3}} = \sqrt[3]{\frac{12 \times (5 \times 10^{-6})}{0.25}} = \sqrt[3]{0.00024}$$
$$h_{optimal} \approx 0.06214$$

  *The optimal step size is approximately **0.062**.*

7.5. Introduction to Richardson's Extrapolation

Richardson's extrapolation is a powerful technique used to generate high-accuracy results by combining low-order approximations. This method is applicable whenever the error of an approximation formula has a predictable structure, typically as a power series in the step size h.

The core idea is that if we know the form of the error, we can use calculations with two different step sizes to cancel out the leading error term, resulting in a more accurate approximation. For example, the central difference formula has an error expansion:

$$f'(x) = \underbrace{\frac{f(x+h) - f(x-h)}{2h}}_{N_1(h)} - \frac{h^2}{6}f'''(x) - \frac{h^4}{120}f^{(5)}(x) - \cdots$$

This predictable error structure is what allows for extrapolation.

7.5.1. The Extrapolation Process

Let's assume we have an approximation formula $N_1(h)$ for an unknown value M, and its error can be expressed as a series in h:

$$M = N_1(h) + K_1 h + K_2 h^2 + K_3 h^3 + \cdots$$

Here, the truncation error is $O(h)$. Now, we compute the approximation again, but with half the step size, $h/2$:

$$M = N_1(h/2) + K_1(h/2) + K_2(h/2)^2 + K_3(h/2)^3 + \cdots$$

To eliminate the leading error term ($K_1 h$), we can multiply the second equation by 2 and subtract the first equation:

$$\begin{aligned} 2M - M &= (2N_1(h/2) - N_1(h)) + (K_1 h - K_1 h) + (K_2 h^2/2 - K_2 h^2) + \cdots \\ M &= \underbrace{N_1(h/2) + [N_1(h/2) - N_1(h)]}_{\text{New approximation, } N_2(h)} - \frac{K_2}{2} h^2 - \cdots \end{aligned}$$

We have created a new formula, $N_2(h) = N_1(h/2) + [N_1(h/2) - N_1(h)]$, which has a more accurate error of order $O(h^2)$.

Example 7.5.1. *Use the forward-difference formula ($O(h)$) to approximate the derivative of $f(x) = \ln(x)$ at $x_0 = 1.8$ using $h = 0.1$ and $h = 0.05$. Then, use extrapolation to find a better approximation.*

Solution: *The true value is $f'(1.8) = 1/1.8 = 0.\overline{5}$.*

- *With $h = 0.1$:* $N_1(0.1) = \frac{\ln(1.9)-\ln(1.8)}{0.1} \approx 0.54067$
- *With $h = 0.05$:* $N_1(0.05) = \frac{\ln(1.85)-\ln(1.8)}{0.05} \approx 0.54798$

Now, we apply the extrapolation formula for an $O(h)$ approximation:

$$\begin{aligned} N_2(0.1) &= N_1(0.05) + [N_1(0.05) - N_1(0.1)] \\ &= 0.54798 + [0.54798 - 0.54067] = 0.55529 \end{aligned}$$

The error of the extrapolated value ($|0.55555 - 0.55529| \approx 2.6 \times 10^{-4}$) is significantly smaller than the error of the initial approximations.

7.5.2. Extrapolation for $O(h^2)$ Formulas

Many formulas, like the central difference formula, have truncation errors that contain only even powers of h:

$$M = N_1(h) + K_1 h^2 + K_2 h^4 + K_3 h^6 + \cdots$$

To eliminate the $K_1 h^2$ term, we again evaluate at h and $h/2$. Multiplying the equation for $h/2$ by 4 and subtracting the equation for h gives:

$$3M = 4N_1(h/2) - N_1(h) - \frac{3K_2}{4} h^4 - \cdots$$

This yields a new formula, $N_2(h)$, with an error of $O(h^4)$:

$$N_2(h) = \frac{4N_1(h/2) - N_1(h)}{3} = N_1(h/2) + \frac{N_1(h/2) - N_1(h)}{3}$$

7.6. Romberg Integration

Romberg integration is a specific and powerful application of Richardson's extrapolation to the Composite Trapezoidal Rule. The error for the Composite Trapezoidal rule can be expressed as a series containing only even powers of h, which makes it a perfect candidate for extrapolation.

$$\int_a^b f(x)dx = \text{Trap}(h) + K_1 h^2 + K_2 h^4 + K_3 h^6 + \cdots$$

The process begins by calculating approximations using the Composite Trapezoidal rule with a sequence of decreasing step sizes, typically for $n = 1, 2, 4, 8, \ldots$ subintervals. These initial approximations are denoted $R_{k,1}$, where k is the level of refinement.

We then build a triangular table of approximations, where each subsequent column is more accurate. The general extrapolation formula is:

$$R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}, \quad \text{for } k \geq j \geq 2$$

The values $R_{k,j}$ are approximations of order $O(h^{2j})$.

Example 7.6.1. *Use Romberg integration to find a high-accuracy approximation for $\int_0^\pi \sin(x)dx$. The exact value is 2.*

Solution: *First, we compute the initial Trapezoidal approximations (the first column of the Romberg table):*

- $n = 1, h = \pi$: $R_{1,1} = \frac{\pi}{2}[\sin(0) + \sin(\pi)] = 0$
- $n = 2, h = \pi/2$: $R_{2,1} = \frac{\pi/4}{[} \sin(0) + 2\sin(\pi/2) + \sin(\pi)] = 1.570796$
- $n = 4, h = \pi/4$: $R_{3,1} = \frac{\pi/8}{[} \ldots] = 1.896119$

Now, we apply the extrapolation formula to generate the second column ($O(h^4)$):

- $R_{2,2} = R_{2,1} + \frac{R_{2,1} - R_{1,1}}{3} = 1.570796 + \frac{1.570796 - 0}{3} \approx 2.094395$
- $R_{3,2} = R_{3,1} + \frac{R_{3,1} - R_{2,1}}{3} = 1.896119 + \frac{1.896119 - 1.570796}{3} \approx 2.004560$

We can go one step further to generate the first entry in the third column ($O(h^6)$):

- $R_{3,3} = R_{3,2} + \frac{R_{3,2} - R_{2,2}}{15} = 2.004560 + \frac{2.004560 - 2.094395}{15} \approx 1.998571$

The Romberg table begins as follows, rapidly converging to the exact answer of 2.

$R_{k,1}$ ($O(h^2)$)	$R_{k,2}$ ($O(h^4)$)	$R_{k,3}$ ($O(h^6)$)
0		
1.570796	*2.094395*	
1.896119	*2.004560*	*1.998571*

7.7. Adaptive Quadrature Methods

Composite rules use a uniform step size across the entire interval, which is inefficient for functions that have regions of high variation and regions of low variation. An adaptive method adjusts the step size to use smaller steps over "bumpy" regions and larger steps over "flat" regions.

For example, a function like $y(x) = e^{-3x}\sin(4x)$ changes rapidly near $x = 0$ but becomes very smooth as x increases. An adaptive method would automatically concentrate the computational effort where it is most needed.

7.7.1. The Adaptive Strategy

The core idea is to estimate the error locally and subdivide the interval if the error is too large.

1. Start with an interval $[a, b]$ and a desired tolerance ε.

2. Calculate a "coarse" approximation, for example, using Simpson's rule once on the whole interval: $S_1 = S(a, b)$.
3. Calculate a "finer" approximation by applying Simpson's rule to the two half-intervals: $S_2 = S(a, \frac{a+b}{2}) + S(\frac{a+b}{2}, b)$.
4. Estimate the error of the finer approximation S_2. For Simpson's rule, this estimate is:

$$\text{Error}(S_2) \approx \frac{1}{15}|S_2 - S_1|$$

5. If the estimated error is less than the tolerance ε, accept S_2 as the result for the interval.
6. If the error is greater than ε, apply the entire procedure recursively to the two subintervals $[a, \frac{a+b}{2}]$ and $[\frac{a+b}{2}, b]$ with a halved tolerance $\varepsilon/2$.

Example 7.7.1. *Use adaptive quadrature based on Simpson's rule to approximate* $\int_0^{\pi/2} \sin(x)dx = 1$ *with a tolerance* $\varepsilon = 0.001$.

Solution:

- ***Step 1 (Coarse Approx):*** $S(0, \pi/2) = \frac{\pi/4}{3}[\sin(0) + 4\sin(\pi/4) + \sin(\pi/2)] \approx 1.00228$
- ***Step 2 (Finer Approx):***
  - $S(0, \pi/4) = \frac{\pi/8}{3}[\sin(0) + 4\sin(\pi/8) + \sin(\pi/4)] \approx 0.29293$
  - $S(\pi/4, \pi/2) = \frac{\pi/8}{3}[\sin(\pi/4) + 4\sin(3\pi/8) + \sin(\pi/2)] \approx 0.70720$
  - *Total finer approx:* $S_2 = 0.29293 + 0.70720 = 1.00013$
- ***Step 3 (Error Estimate):***

$$\textit{Error} \approx \frac{1}{15}|S_2 - S_1| = \frac{1}{15}|1.00013 - 1.00228| \approx 0.000143$$

- ***Step 4 (Conclusion):*** *Since the estimated error* (0.000143) *is less than the tolerance* (0.001)*, we accept the finer approximation. The result is* 1.00013*. No further subdivision is needed.*

7.8. Method of Undetermined Coefficients

This is an alternative way to derive quadrature formulas. The idea is to propose a general form for the integral and its error, and then solve for the unknown coefficients by forcing the formula to be exact for successive powers of x (i.e., for $f(x) = 1, x, x^2, \ldots$).

7.8.1. Derivation of the Trapezoidal Rule

We assume the integral can be written in the form:

$$\int_a^b f(x)\,dx = A_0 f(a) + A_1 f(b) + \alpha f''(\beta)$$

Our goal is to find the unknown coefficients A_0, A_1, and α. We do this by plugging in simple polynomials, for which we know the exact integral.

- **Case 1:** $f(x) = 1$. Here $f'(x) = 0$, $f''(x) = 0$. The formula becomes:

$$\int_a^b 1\,dx = A_0(1) + A_1(1) + 0 \implies b - a = A_0 + A_1$$

- **Case 2:** $f(x) = x$**.** Here $f'(x) = 1$, $f''(x) = 0$. The formula becomes:

$$\int_a^b x\,dx = A_0(a) + A_1(b) + 0 \implies \frac{b^2 - a^2}{2} = aA_0 + bA_1$$

Solving this system of two linear equations for A_0 and A_1 gives:

$$A_0 = \frac{b-a}{2} \quad \text{and} \quad A_1 = \frac{b-a}{2}$$

The integration rule thus far is $\int_a^b f(x)\,dx = \frac{b-a}{2}[f(a) + f(b)] + \alpha f''(\beta)$.

- **Case 3:** $f(x) = x^2$**.** Now we use $f(x) = x^2$ to find the error coefficient α. Here $f''(x) = 2$.

$$\int_a^b x^2\,dx = \frac{b-a}{2}[a^2 + b^2] + \alpha(2)$$
$$\frac{b^3 - a^3}{3} = \frac{(b-a)(a^2+b^2)}{2} + 2\alpha$$

Solving for α (after some algebra) gives:

$$2\alpha = \frac{b^3 - a^3}{3} - \frac{b^3 - a^3 + a^2b - ab^2}{2} = \frac{-(b-a)^3}{6}$$
$$\alpha = -\frac{(b-a)^3}{12}$$

Substituting this back, we get the **Trapezoidal Rule** with its error term:

$$\int_a^b f(x)\,dx = \frac{b-a}{2}[f(a) + f(b)] - \frac{(b-a)^3}{12} f''(\beta)$$

7.8.2. Derivation of Simpson's 3/8 Rule

We can use the same method for higher-order rules.

Example 7.8.1. *Find the coefficients for the Simpson's 3/8 rule.* ***Problem:*** *Assume a formula of the form:*

$$\int_{x_0}^{x_3} f(x)\,dx = A_0 f_0 + A_1 f_1 + A_2 f_2 + A_3 f_3 + \alpha f^{(4)}(\beta)$$

where the nodes are $x_0 = 0, x_1 = h, x_2 = 2h, x_3 = 3h$. ***Solution Outline:*** *To solve for the four coefficients A_0, A_1, A_2, A_3, we would enforce the rule to be exact for $f(x) = 1, x, x^2,$ and x^3. This creates a 4×4 system of linear equations.*

- $f(x) = 1 : \int_0^{3h} 1\,dx = 3h = A_0 + A_1 + A_2 + A_3$
- $f(x) = x : \int_0^{3h} x\,dx = \frac{9h^2}{2} = A_1(h) + A_2(2h) + A_3(3h)$
- $f(x) = x^2 : \int_0^{3h} x^2\,dx = 9h^3 = A_1(h^2) + A_2(4h^2) + A_3(9h^2)$
- $f(x) = x^3 : \int_0^{3h} x^3\,dx = \frac{81h^4}{4} = A_1(h^3) + A_2(8h^3) + A_3(27h^3)$

Solving this system yields: $A_0 = \frac{3h}{8}$, $A_1 = \frac{9h}{8}$, $A_2 = \frac{9h}{8}$, $A_3 = \frac{3h}{8}$. We would then use $f(x) = x^4$ to solve for the error coefficient α, which gives $\alpha = -\frac{3h^5}{80}$. The final formula is ***Simpson's 3/8 Rule****:*

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}[f_0 + 3f_1 + 3f_2 + f_3] - \frac{3h^5}{80} f^{(4)}(\beta)$$

Gaussian Quadrature

8.1. Introduction to Gaussian Quadrature

All Newton-Cotes formulas (Trapezoidal, Simpson's, etc.) use function values at nodes that are **equally spaced**. The core idea of Gaussian Quadrature is that by **optimally choosing the nodes** (locations x_i) as well as the weights (c_i), we can achieve a much higher degree of accuracy.

The general form of a Gaussian quadrature rule is:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} c_i f(x_i)$$

In a Newton-Cotes formula with n nodes, the nodes are fixed, and we only solve for n weights, giving us n parameters. This typically results in a degree of accuracy of $n-1$. In Gaussian Quadrature, both the n nodes x_i and the n weights c_i are unknown parameters. This gives us $2n$ parameters to choose. With $2n$ free parameters, we can construct a formula that is exact for all polynomials of degree $2n-1$ or less.

8.2. Derivation of the Two-Point Gaussian Rule

Let's derive the formula for $n=2$ on the standard interval $[-1,1]$. We want to find the $2n=4$ parameters c_1, c_2, x_1, x_2 such that the formula

$$\int_{-1}^{1} f(x)\,dx \approx c_1 f(x_1) + c_2 f(x_2)$$

is exact for all polynomials of degree $2n-1=3$ or less. We achieve this by enforcing exactness for $f(x)=1, x, x^2, x^3$.

- **For $f(x)=1$:**
$$\int_{-1}^{1} 1\,dx = 2 \quad \Longrightarrow \quad c_1 + c_2 = 2$$
- **For $f(x)=x$:**
$$\int_{-1}^{1} x\,dx = 0 \quad \Longrightarrow \quad c_1x_1 + c_2x_2 = 0$$
- **For $f(x)=x^2$:**
$$\int_{-1}^{1} x^2\,dx = \frac{2}{3} \quad \Longrightarrow \quad c_1x_1^2 + c_2x_2^2 = \frac{2}{3}$$
- **For $f(x)=x^3$:**
$$\int_{-1}^{1} x^3\,dx = 0 \quad \Longrightarrow \quad c_1x_1^3 + c_2x_2^3 = 0$$

Solving this system of four non-linear equations (with some algebraic manipulation) yields the unique solution:

$$c_1 = 1, \quad c_2 = 1, \quad x_1 = -\frac{1}{\sqrt{3}}, \quad x_2 = \frac{1}{\sqrt{3}}$$

This gives the **two-point Gaussian quadrature formula**:

$$\int_{-1}^{1} f(x)\,dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

This simple formula has a degree of accuracy of 3.

Example 8.2.1. *Use the two-point Gaussian quadrature formula to approximate $\int_{-1}^{1} x^3 dx$ and $\int_{-1}^{1} x^4 dx$.* ***Solution:***

- *For $f(x) = x^3$:*

$$\int_{-1}^{1} x^3 dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) = \left(-\frac{1}{\sqrt{3}}\right)^3 + \left(\frac{1}{\sqrt{3}}\right)^3 = -\frac{1}{3\sqrt{3}} + \frac{1}{3\sqrt{3}} = 0$$

The exact answer is 0. The formula is exact, as expected.

- *For $f(x) = x^4$:*

$$\int_{-1}^{1} x^4 dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) = \left(-\frac{1}{\sqrt{3}}\right)^4 + \left(\frac{1}{\sqrt{3}}\right)^4 = \frac{1}{9} + \frac{1}{9} = \frac{2}{9}$$

The exact answer is $\int_{-1}^{1} x^4 dx = [\frac{x^5}{5}]_{-1}^{1} = \frac{2}{5}$. The formula is not exact, because the polynomial is degree 4, which is greater than the degree of accuracy (3).

8.3. Gaussian Quadrature: The General Case

To find the n weights c_i and n nodes x_i, we require the formula to be exact for $f(x) = x^k$ for $k = 0, 1, \dots, 2n-1$. This creates a system of $2n$ non-linear equations for the $2n$ unknowns:

$$\sum_{i=1}^{n} c_i (x_i)^k = \int_{-1}^{1} x^k \, dx \quad \text{for } k = 0, 1, \dots, 2n-1$$

The right-hand side values are known:

$$\int_{-1}^{1} x^k \, dx = \begin{cases} \frac{2}{k+1} & \text{if } k \text{ is even} \\ 0 & \text{if } k \text{ is odd} \end{cases}$$

Solving this system is complex. In practice, the nodes x_i are found to be the roots of the n-th degree **Legendre polynomial**, and the weights c_i are then solved for using the first n equations.

Practice Questions

1. **Basic First Derivative Approximations**
   The following data were generated from the function $f(x) = x^2 \cos(x)$.

x	$f(x)$
0.0	0.00000
0.2	0.03920
0.4	0.14730

   a. Use the two-point **forward-difference** formula to approximate $f'(0.2)$.
   b. Use the two-point **backward-difference** formula to approximate $f'(0.2)$.
   c. Use the three-point **central-difference** formula to approximate $f'(0.2)$.
   d. The true value is $f'(0.2) \approx 0.38415$. Calculate the absolute error for each approximation.

2. **Forward and Backward Difference Formulas**
   Use forward-difference formulas and backward-difference formulas to determine each missing entry in the following table:

x	$f(x)$	$f'(x)$
0.5	0.4794	
0.6	0.5646	
0.7	0.6442	

   The data were taken from $f(x) = \sin x$. Compute the actual errors and find error bounds using the error formulas.

3. **Three-Point Formula for First Derivative**
   Use the most accurate three-point formula to determine each missing entry in the following table:

x	$f(x)$	$f'(x)$
1.1	9.025013	
1.2	11.02318	
1.3	13.46374	
1.4	16.44465	

   The data were taken from $f(x) = xe^x$. Compute the actual errors and find error bounds.

4. **Second Derivative Approximation**
   Let $f(x) = e^{2x}$.

   a. Use the central-difference formula for the second derivative,

$$f''(x_0) \approx \frac{f(x_0 - h) - 2f(x_0) + f(x_0 + h)}{h^2}$$

   to find an approximation for $f''(1.0)$ using a step size of $h = 0.1$.
   b. Calculate the true value of $f''(1.0)$ and determine the absolute error of your approximation.

5. **Approximating Derivatives from Tabular Data I**
   Use the numerical differentiation formulas from this section to determine, as accurately as possible, approximations for each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
2.1	-1.709847	
2.2	-1.373823	
2.3	-1.119214	
2.4	-0.9160143	
2.5	-0.7470223	
2.6	-0.6015966	

b.

x	$f(x)$	$f'(x)$
-3.0	9.367879	
-2.8	8.233241	
-2.6	7.180350	
-2.4	6.209329	
-2.2	5.320305	
-2.0	4.513417	

6. **Richardson Extrapolation**
   Let $f(x) = \sqrt{1 + x^2}$.

   a. Use the three-point centered formula with $h = 0.1$ to approximate $f'(0)$.

   b. Use Richardson extrapolation with $h = 0.1$ and $h = 0.05$ to improve the approximation.

   c. Compare with the exact value of $f'(0)$.

7. **Optimal Step Size I**
   For the central-difference formula, $f'(x_0) \approx \frac{f(x_0+h)-f(x_0-h)}{2h}$, the total error is bounded by:

$$E(h) \leq \frac{\epsilon}{h} + \frac{h^2}{6}M_3$$

   where ϵ is the maximum round-off error in function evaluation and $M_3 = \max|f'''(x)|$. Suppose you are calculating the derivative of $f(x) = \sin(x)$ near $x = \pi/4$, and the maximum round-off error in computing $\sin(x)$ is $\epsilon = 5\times10^{-8}$. Find the **optimal step size**, h, that minimizes the total error bound.

8. **Optimal Step Size II**
   Consider the function $f(x) = \cos(\pi x)$.

   a. Find the optimal step size h for approximating $f'(0.3)$ using the central difference formula, given that round-off errors are bounded by $\epsilon = 10^{-8}$.

   b. Use this optimal h to compute $f'(0.3)$ and compare with the exact value.

   c. What is the theoretical minimum total error?

9. **High-Accuracy Approximation and Error Bounding**
   Use the following data and the knowledge that the first five derivatives of f are bounded on the interval $[1, 5]$ as follows:

$$|f'(x)| \leq 2, \quad |f''(x)| \leq 3, \quad |f'''(x)| \leq 6, \quad |f^{(4)}(x)| \leq 12, \quad |f^{(5)}(x)| \leq 23$$

x	1	2	3	4	5
$f(x)$	2.4142	2.6734	2.8974	3.0976	3.2804

   a. Approximate $f'(3)$ as accurately as possible using the given data.

   b. Find a bound for the error in your approximation.

10. **Position, Velocity, and Acceleration**
    The following data represent the position $s(t)$ of a moving object:

t	0	0.2	0.4	0.6	0.8	1.0
$s(t)$	0	0.12	0.48	1.08	1.92	3.00

    a. Estimate the velocity $v(t) = s'(t)$ at each time point.

    b. Estimate the acceleration $a(t) = s''(t)$ at $t = 0.4$ and $t = 0.6$.

    c. If $s(t) = 3t^2$, compare your numerical results with exact values.

11. **Deriving Third Derivative Formula**
    Derive a method for approximating $f'''(x_0)$ using the values $f(x_0-2h)$, $f(x_0-h)$, $f(x_0)$, $f(x_0+h)$, and $f(x_0+2h)$ that has the optimal order of accuracy. What is the error term? Test your formula on $f(x) = x^4$ at $x_0 = 1$ with $h = 0.1$.

12. **Comparing Trapezoidal and Simpson's Rules**
Use the basic (non-composite) **Trapezoidal rule** and **Simpson's 1/3 rule** to approximate the following integral:

$$\int_0^1 \frac{2}{1+x^2}\,dx$$

The exact value of the integral is $\pi/2 \approx 1.57080$. Compare the accuracy of the two methods.

13. **Basic Integration Rules**
Approximate the following integrals using both the Trapezoidal Rule and Simpson's Rule:

    a. $\displaystyle\int_0^2 x^2 \ln(x^2+1)\,dx$

    b. $\displaystyle\int_{-1}^1 x^2 e^{-x}\,dx$

    c. $\displaystyle\int_0^1 \frac{x}{x^2+4}\,dx$

    d. $\displaystyle\int_1^3 \frac{1}{x^2}\,dx$

14. **Composite Trapezoidal Rule**
Use the Composite Trapezoidal Rule with $n = 4$ and $n = 8$ subintervals to approximate:

$$\int_1^3 \frac{1}{x}\,dx$$

Compare with the exact value $\ln 3 = 1.0986123$.

15. **Composite Simpson's Rule I**
Use the Composite Simpson's Rule with $n = 4, 6, 8$ to approximate:

$$\int_0^\pi x \sin x\,dx$$

The exact value is π. Compute the errors and verify that they decrease as $O(h^4)$.

16. **Composite Simpson's Rule and Error**
You are asked to approximate the integral

$$\int_0^2 xe^x\,dx$$

using the **Composite Simpson's Rule**. The error term for this rule is given by $-\frac{(b-a)}{180}h^4 f^{(4)}(\mu)$.

    a. Find the fourth derivative of $f(x) = xe^x$.

    b. Determine the minimum number of subintervals, n, required to ensure the absolute error is less than 10^{-4}.

17. **Determining Step Size for a Given Accuracy**
Determine the values of n and h required to approximate the definite integral

$$\int_0^2 e^{2x}\sin(3x)\,dx$$

to within a tolerance of 10^{-4}.

    a. Use the **Composite Trapezoidal rule**.

    b. Use the **Composite Simpson's rule**.

    c. Use the **Composite Midpoint rule**.

18. **Romberg Integration I**
Approximate the integral $\int_1^2 \frac{1}{x}\,dx$ using **Romberg integration**.

a. Calculate the first column of the Romberg table, $R_{1,1}$ and $R_{2,1}$, which correspond to the Composite Trapezoidal rule with $n = 1$ and $n = 2$ subintervals.

b. Use these values to compute the first extrapolated value, $R_{2,2}$.

c. Compare the absolute error of $R_{1,1}$, $R_{2,1}$, and $R_{2,2}$. The exact value is $\ln(2) \approx 0.69315$.

19. **Romberg Integration II**
Use Romberg integration to compute $R_{3,3}$ for the following integrals:

a. $\displaystyle\int_0^2 x^3 e^x \, dx$

b. $\displaystyle\int_1^2 \frac{1}{x} \, dx$

c. $\displaystyle\int_0^{\pi/2} \sin^2 x \, dx$

20. **Integration of a Piecewise Function**
Let the function f be defined by the piecewise expression:

$$f(x) = \begin{cases} x^3 + 1, & 0 \le x \le 0.1, \\ 1.001 + 0.03(x - 0.1) + 0.3(x - 0.1)^2 + 2(x - 0.1)^3, & 0.1 < x \le 0.2, \\ 1.009 + 0.15(x - 0.2) + 0.9(x - 0.2)^2 + 2(x - 0.2)^3, & 0.2 < x \le 0.3. \end{cases}$$

a. Investigate the continuity of the derivatives of f at the points where the pieces meet.

b. Use the **Composite Trapezoidal rule** with $n = 6$ to approximate $\int_0^{0.3} f(x) \, dx$, and estimate the error using the error bound formula.

c. Use the **Composite Simpson's rule** with $n = 6$ to approximate $\int_0^{0.3} f(x) \, dx$. Are the results more accurate than in part (b)? Discuss why or why not, considering your findings from part (a).

21. **Gaussian Quadrature**
The two-point Gaussian quadrature formula is given by:

$$\int_{-1}^{1} f(x) \, dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

a. Use a change of variables to approximate:

$$\int_0^1 e^{-x^2} \, dx$$

b. Use the three-point Gaussian quadrature formula (with nodes at $x = 0, \pm\sqrt{3/5}$ and weights $8/9, 5/9, 5/9$) to approximate the same integral.

c. Compare the accuracy of both methods.

Initial-Value Problems for Ordinary Differential Equations

9.1. Introduction to Differential Equations

A differential equation is an equation that involves one dependent variable and its derivatives with respect to one or more independent variables.

- **Ordinary Differential Equations (ODE):** A differential equation involving only ordinary derivatives with respect to **only one** independent variable.
- **Partial Differential Equations (PDE):** A differential equation involving partial derivatives with respect to **more than one** independent variable.

Example 9.1.1.

- ***ODE:*** $\frac{d^2y}{dx^2} + 3\frac{dy}{dx} - 4y = \sin(x)$*. Here, y is the dependent variable and x is the single independent variable.*
- ***PDE:*** $\frac{\partial u}{\partial t} = c^2\frac{\partial^2 u}{\partial x^2}$*. Here, u is the dependent variable, and it depends on two independent variables, t and x.*

The **order** of a differential equation is the order of the highest derivative present. The general ordinary differential equation of the **nth order** can be written as:

$$F(x, y, y', y'', \cdots, y^{(n)}) = 0$$

In this chapter, we will focus on first-order initial-value problems (IVPs), which have the form $\frac{dy}{dx} = f(x, y)$ with an initial condition $y(x_0) = y_0$.

9.2. Well-Posed Problems

For a numerical method to be effective, the problem it is solving should be "well-posed." This means the problem is a stable and predictable model of a physical phenomenon.

The initial value problem

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0, \quad a \leq x \leq b \tag{9.1}$$

is said to be a **well-posed problem** if it satisfies three conditions:

1. A **unique solution**, $y(x)$, exists.
2. The solution exhibits **continuous dependence on initial data**.

   This means that a small change in the starting condition (y_0) or a small, continuous disturbance in the equation itself ($\delta(x)$) should only lead to a small change in the final solution $y(x)$.

   More formally, there exist constants $\epsilon_0 > 0$ and $k > 0$ such that for any ϵ (with $\epsilon_0 > \epsilon > 0$), whenever $|\delta_0| < \epsilon$ and $\delta(x)$ is a continuous function with $|\delta(x)| < \epsilon$ for all $x \in [a, b]$, the *perturbed problem*:

   $$\frac{dz}{dx} = f(x, z) + \delta(x), \quad z(x_0) = y_0 + \delta_0 \tag{9.2}$$

   has a unique solution $z(x)$ that satisfies:

   $$|y(x) - z(x)| < k\epsilon \quad \forall x \in [a, b]$$

3. The problem specified by (9.2) is called a perturbed problem.

9.3. Existence and Uniqueness of Solutions

Before attempting to solve an IVP, we must ask:

- Does a solution even exist?
- If a solution exists, is it the only one?

The following example shows why these questions are necessary.

Example 9.3.1. *Consider the IVP:*

$$\frac{dy}{dx} = \frac{3y}{x}, \quad y(0) = 1$$

- *The general solution to the ODE is $y(x) = cx^3$.*
- *Applying the initial condition $y(0) = 1$, we get $1 = c \cdot (0)^3$, or $1 = 0$, which is impossible.*
- *Therefore, this IVP has **no solution**.*
- *If we modify the initial condition to $y(0) = 0$, we get $0 = c \cdot (0)^3$, which is $0 = 0$. This is true for any constant c. The new IVP has an **infinite number of solutions** (e.g., $y = x^3, y = 2x^3, y = -5x^3, \ldots$).*

9.3.1. Existence of a Solution

Theorem 9.3.2 (Peano Existence Theorem)**.** *Let $f(x, y)$ be a **continuous** function in the closed rectangular domain*

$$\mathbf{R = \{(x, y) : |x - x_0| \leq a, |y - y_0| \leq b\}}.$$

*Then, the IVP $\frac{dy}{dx} = f(x, y)$ with $y(x_0) = y_0$ has **at least one solution** in the interval $I = \{x : |x - x_0| < h\}$, where $h = \min\left\{a, \frac{b}{M}\right\}$ and $M = \max\limits_{(x,y)\in R} |f(x, y)|$.*

This theorem provides a *sufficient* condition (if f is continuous, a solution exists), but it is not a *necessary* condition.

Example 9.3.3. *Consider the IVP $\frac{dy}{dx} = \frac{y}{x}$ with $y(0) = 0$.*

- *Here $f(x, y) = y/x$, $x_0 = 0$, $y_0 = 0$.*
- *In any rectangular domain R around $(0, 0)$, $f(x, y)$ is continuous everywhere **except** at $x = 0$.*
- *Because the condition of the theorem (continuity in R) is violated, the theorem **cannot be applied**. It tells us nothing about the existence of a solution.*
- *However, we can see by inspection that $y(x) = cx$ is a solution for any c, so the IVP does have solutions.*

9.3.2. Uniqueness of the Solution

Theorem 9.3.4 (Picard's Uniqueness Theorem)**.** *Let $f(x, y)$ and $\frac{\partial f}{\partial y}$ be **continuous** functions in the closed rectangular domain*

$$\mathbf{R = \{(x, y) : |x - x_0| \leq a, |y - y_0| \leq b\}}.$$

*Then, the IVP $\frac{dy}{dx} = f(x, y)$ with $y(x_0) = y_0$ has a **unique solution** in the interval $I = \{x : |x - x_0| < h\}$, where $h = \min\left\{a, \frac{b}{M}\right\}$ and $M = \max\limits_{(x,y)\in R} |f(x, y)|$.*

This theorem is also a sufficient condition, but not a necessary one. A problem may have a unique solution even if $\frac{\partial f}{\partial y}$ is not continuous. A weaker (but more general) condition than continuity of $\frac{\partial f}{\partial y}$ is the **Lipschitz condition**.

9.4. Picard's Iteration Method

Picard's method is a theoretical tool that provides a way to find the solution of an IVP by constructing a sequence of functions that converge to the actual solution.

The IVP

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0 \tag{9.3}$$

is equivalent to the integral equation:

$$y(x) = y_0 + \int_{x_0}^{x} f(t, y(t))dt \tag{9.4}$$

This conversion is the foundation of the method, as it turns a differential problem into an integral one, which can be solved by successive approximations.

We start with an initial guess, $y_0(x) = y_0$, and iteratively generate new approximations:

$$y_1(x) = y_0 + \int_{x_0}^{x} f(t, y_0(t))dt = y_0 + \int_{x_0}^{x} f(t, y_0)dt \tag{9.5}$$

$$y_2(x) = y_0 + \int_{x_0}^{x} f(t, y_1(t))dt \tag{9.6}$$

$$\vdots$$

$$y_n(x) = y_0 + \int_{x_0}^{x} f(t, y_{n-1}(t))dt \tag{9.7}$$

Theorem 9.4.1. *If the function $f(x, y)$ satisfies the conditions of the existence and uniqueness theorem, then the successive approximations $y_n(x)$ generated by Picard's iteration converge to the unique solution $y(x)$ of the IVP.*

Example 9.4.2. *Apply Picard iteration for the IVP $\frac{dy}{dx} = 2x(1 - y)$ with $y(0) = 2$.*

Here $f(x, y) = 2x(1 - y)$, $x_0 = 0$, and our initial guess is $y_0(x) = y_0 = 2$. ***Iteration 1:***

$$\begin{aligned} y_1(x) &= y_0 + \int_{x_0}^{x} f(t, y_0(t))dt \\ &= 2 + \int_{0}^{x} 2t(1-2)dt = 2 + \int_{0}^{x} (-2t)dt = 2 - [t^2]_0^x = 2 - x^2 \end{aligned}$$

Iteration 2:

$$\begin{aligned} y_2(x) = y_0 + \int_{x_0}^{x} f(t, y_1(t))dt &= 2 + \int_{0}^{x} 2t(1 - (2 - t^2))dt \\ &= 2 + \int_{0}^{x} 2t(t^2 - 1)dt = 2 + \int_{0}^{x} (2t^3 - 2t)dt \\ &= 2 + \left[\frac{t^4}{2} - t^2\right]_0^x = 2 - x^2 + \frac{x^4}{2} \end{aligned}$$

Iteration 3:

$$\begin{aligned} y_3(x) = y_0 + \int_{x_0}^{x} f(t, y_2(t))dt &= 2 + \int_{0}^{x} 2t\left(1 - \left(2 - t^2 + \frac{t^4}{2}\right)\right)dt \\ &= 2 + \int_{0}^{x} 2t\left(t^2 - \frac{t^4}{2} - 1\right)dt = 2 + \int_{0}^{x} \left(2t^3 - t^5 - 2t\right)dt \\ &= 2 + \left[\frac{t^4}{2} - \frac{t^6}{6} - t^2\right]_0^x = 2 - x^2 + \frac{x^4}{2} - \frac{x^6}{6} \end{aligned}$$

The pattern emerges: $y_n(x) = 2 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \cdots + (-1)^n \frac{x^{2n}}{n!}$. As $n \to \infty$, $y_n(x)$ converges to:

$$y(x) = 1 + \left(1 - x^2 + \frac{(x^2)^2}{2!} - \frac{(x^2)^3}{3!} + \cdots\right) = 1 + e^{-x^2}$$

Check with exact solution: *The ODE is separable:* $\frac{dy}{1-y} = 2x\,dx$*. Integrating gives* $\int \frac{dy}{y-1} = \int -2x\,dx$*, so* $\ln|y-1| = -x^2 + C$*. This gives* $y - 1 = Ke^{-x^2}$*, or* $y(x) = 1 + Ke^{-x^2}$*. Using* $y(0) = 2$*:* $2 = 1 + Ke^0 \implies K = 1$*. The exact solution is* $y(x) = 1 + e^{-x^2}$*. Thus, the Picard iterates converge to the unique solution.*

9.5. Numerical Methods: Introduction

Picard's method is a powerful theoretical tool, but the integrations are often too complex to perform analytically. For most real-world problems, we must use numerical methods to find an approximate solution at discrete points.

9.5.1. The Role of Taylor's Series

Numerical methods are based on Taylor's series expansion. Recall that for a function $y(x)$ that is infinitely differentiable at x_i, we can write its value at a nearby point $x_{i+1} = x_i + h$:

$$\begin{aligned} y(x_{i+1}) &= y(x_i + h) \\ &= y(x_i) + \frac{y'(x_i)}{1!}h + \frac{y''(x_i)}{2!}h^2 + \frac{y'''(x_i)}{3!}h^3 + \cdots + \frac{y^{(n+1)}(\zeta_n)}{(n+1)!}h^{n+1} \end{aligned}$$

for some $\zeta_n \in (x_i, x_{i+1})$.

9.5.2. Discretization and Grid Setup

We first approximate the continuous interval $[a, b]$ with a finite set of grid points:

$$a = x_0 < x_1 < x_2 < \cdots < x_n = b$$

- **Step length:** The spacing between points is $h_i = x_i - x_{i-1}$.
- **Uniform grid:** We often use a constant step length h, so $x_i = x_0 + ih$.

We use y_i to denote the **numerical approximation** of the true solution $y(x_i)$. The goal of a numerical method is to generate the set of numbers $\{y_0, y_1, \ldots, y_n\}$, which is the numerical solution to the IVP.

9.6. Euler's Method

Euler's method is the simplest numerical method, derived by truncating the Taylor series after the first-order term. Starting with the Taylor expansion:

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(\zeta_n)$$

Since $y'(x) = f(x, y(x))$, we have $y'(x_n) = f(x_n, y(x_n))$.

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + \frac{h^2}{2}y''(\zeta_n)$$

By dropping the error term and replacing the true solution $y(x_i)$ with its numerical approximation y_i, we get the **Euler's Method** formula:

$$y_{n+1} = y_n + hf(x_n, y_n)$$

The term $T_n = \frac{h^2}{2}y''(\zeta_n)$ that we dropped is called the **local truncation error**. It is "local" because it's the error introduced in a single step, assuming y_n was perfectly accurate.

9.6.1. Geometrical Interpretation

Euler's method is a simple "point-slope" method.

- Start at the known point (x_0, y_0).
- Calculate the slope at this point using the ODE: $m_0 = f(x_0, y_0)$.
- Follow this slope m_0 for a horizontal distance h to find the next point:

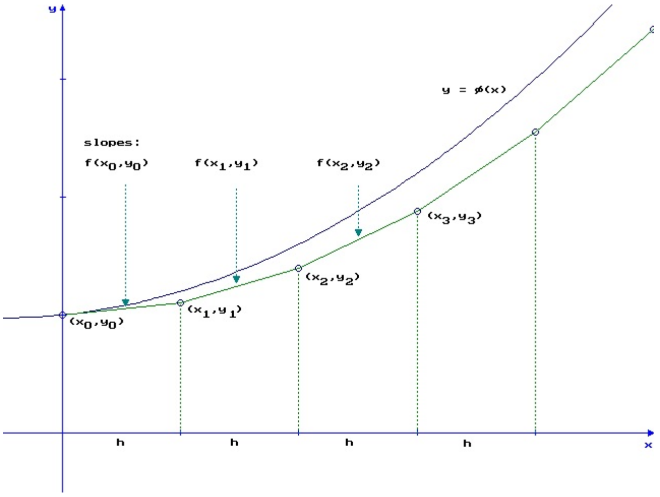
$$y_1 = y_0 + h \cdot m_0 = y_0 + hf(x_0, y_0)$$

- Repeat the process from the new point (x_1, y_1):

$$y_2 = y_1 + hf(x_1, y_1)$$

- In general: $y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$.

Geometrically, the numerical solution is a sequence of connected line segments, where the slope of each segment is determined at its starting point.



Example 9.6.1. *Find an approximate value of y corresponding to $x = 0.5$ for the IVP $\frac{dy}{dx} = x + y$, $y(0) = 1$, using $h = 0.1$.*

Here $x_0 = 0, y_0 = 1, h = 0.1$, and $f(x, y) = x + y$. The exact solution is $y(x) = 2e^x - x - 1$.

n	x_n	y_n *(Approx)*	$y(x_n)$ *(Exact)*
0	*0.0*	*1.0000*	*1.0000*
1	*0.1*	$y_1 = y_0 + hf(x_0, y_0) = 1.0 + 0.1(0.0 + 1.0) = 1.1000$	*1.1103*
2	*0.2*	$y_2 = y_1 + hf(x_1, y_1) = 1.1 + 0.1(0.1 + 1.1) = 1.2200$	*1.2428*
3	*0.3*	$y_3 = y_2 + hf(x_2, y_2) = 1.22 + 0.1(0.2 + 1.22) = 1.3620$	*1.3997*
4	*0.4*	$y_4 = y_3 + hf(x_3, y_3) = 1.362 + 0.1(0.3 + 1.362) = 1.5282$	*1.5836*
5	*0.5*	$y_5 = y_4 + hf(x_4, y_4) = 1.5282 + 0.1(0.4 + 1.5282) = 1.7210$	*1.7974*

The required approximation is $y(0.5) \approx 1.7210$. The error is $|1.7974 - 1.7210| \approx 0.0764$.

Example 9.6.2. *For the IVP $\frac{dy}{dx} = \frac{y-x}{y+x}$, $y(0) = 1$, find the value of y for $x = 0.08$ with $h = 0.02$.*

Here $x_0 = 0, y_0 = 1, h = 0.02$, and $f(x, y) = (y - x)/(y + x)$.

n	x_n	y_n	$f(x_n, y_n)$
0	*0.00*	*1.0000*	$(1.0000 - 0.00)/(1.0000 + 0.00) = 1.0000$
1	*0.02*	$1.0000 + 0.02(1.0000) = 1.0200$	$(1.0200 - 0.02)/(1.0200 + 0.02) \approx 0.9615$
2	*0.04*	$1.0200 + 0.02(0.9615) = 1.0392$	$(1.0392 - 0.04)/(1.0392 + 0.04) \approx 0.9260$
3	*0.06*	$1.0392 + 0.02(0.9260) = 1.0577$	$(1.0577 - 0.06)/(1.0577 + 0.06) \approx 0.8926$
4	*0.08*	$1.0577 + 0.02(0.8926) = 1.0756$	—

The required approximation is $y(0.08) \approx 1.0756$*.*

9.7. Modified Euler's (Predictor-Corrector) Method

9.7.1. Motivation for Improvement

Euler's method is easy to implement, but it is not very accurate. The solution is only correct if the function is linear, as it uses the slope from the *beginning* of the interval to represent the slope across the *entire* interval. A very small step size h is required for a meaningful result.

A clear improvement would be to use a better slope, such as the **average of the slopes** at the beginning and end of the interval:

$$\text{Average Slope} = \frac{f(x_n, y_n) + f(x_{n+1}, y_{n+1})}{2}$$

This leads to the (implicit) formula:

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

This is an **implicit method** because the unknown y_{n+1} appears on both sides.

9.7.2. The Predictor-Corrector Algorithm

We solve the implicitness by using a two-step process.

1. **Predictor Step:** First, we "predict" a value for y_{n+1} using the simple (explicit) Euler's method.
$$y_{n+1}^{(p)} = y_n + hf(x_n, y_n)$$
2. **Corrector Step:** We use this predicted value to evaluate the slope at x_{n+1}. Then we use the average slope to find a "corrected" value for y_{n+1}.
$$y_{n+1}^{(c)} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(p)})]$$

This is the **Modified Euler's Method**, also known as the Heun's method or a second-order Runge-Kutta method.

9.7.3. Geometrical Interpretation

- At (x_n, y_n), calculate the initial slope $m_1 = f(x_n, y_n)$.
- Use m_1 to predict a temporary point $(x_{n+1}, y_{n+1}^{(p)})$.
- At this temporary point, calculate a second slope $m_2 = f(x_{n+1}, y_{n+1}^{(p)})$.
- Go back to (x_n, y_n) and advance using the *average* of these two slopes, $m_{\text{avg}} = (m_1 + m_2)/2$.
- The final corrected point is $y_{n+1} = y_n + h \cdot m_{\text{avg}}$.

Example 9.7.1. *Find an approximate value of* y *at* $x = 0.5$ *for the IVP* $\frac{dy}{dx} = x + y$, $y(0) = 1$, *using* $h = 0.1$.

We compare the results from Standard Euler and Modified Euler.

Solution with Standard Euler's Method:

From the previous example, $y(0.5) \approx 1.7210$.

n	x_n	y_n	$y(x_n)$ *(Exact)*
0	0.0	1.0000	1.0000
1	0.1	1.1000	1.1103
2	0.2	1.2200	1.2428
3	0.3	1.3620	1.3997
4	0.4	1.5282	1.5836
5	0.5	1.7210	1.7974

Solution with Modified Euler's Method:

Here $f(x, y) = x + y$ *and* $h = 0.1$. *The formula is* $y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(p)})]$.

- ***n=0:*** $x_0 = 0, y_0 = 1$.
  - $f(x_0, y_0) = 0 + 1 = 1.0$
  - *Predictor:* $y_1^{(p)} = 1.0 + 0.1(1.0) = 1.1$
  - $f(x_1, y_1^{(p)}) = 0.1 + 1.1 = 1.2$
  - *Corrector:* $y_1 = 1.0 + \frac{0.1}{2}[1.0 + 1.2] = 1.0 + 0.05(2.2) = 1.1100$
- ***n=1:*** $x_1 = 0.1, y_1 = 1.1100$.
  - $f(x_1, y_1) = 0.1 + 1.1100 = 1.2100$
  - *Predictor:* $y_2^{(p)} = 1.1100 + 0.1(1.2100) = 1.2310$
  - $f(x_2, y_2^{(p)}) = 0.2 + 1.2310 = 1.4310$
  - *Corrector:* $y_2 = 1.1100 + \frac{0.1}{2}[1.2100 + 1.4310] = 1.1100 + 0.05(2.641) = 1.24205$

Continuing this process gives the following table:

n	x_n	y_n *(Mod-Euler)*	$y(x_n)$ *(Exact)*
0	0.0	1.0000	1.0000
1	0.1	1.1100	1.1103
2	0.2	1.24205	1.2428
3	0.3	1.39846	1.3997
4	0.4	1.58180	1.5836
5	0.5	1.79489	1.7974

The required approximation is $y(0.5) \approx 1.79489$. ***Comparison:*** *The error for standard Euler was* ≈ 0.0764. *The error for Modified Euler is* $|1.7974 - 1.79489| \approx 0.0025$, *which is significantly more accurate.*

Example 9.7.2. *Find an approximate value of* y *corresponding to* $x = 0.5$ *with* $h = 0.1$ *for the IVP:*

$$\frac{dy}{dx} = \log(x + y), \quad y(0) = 2$$

Here $x_0 = 0, y_0 = 2, h = 0.1,\ f(x, y) = \ln(x + y)$

x_n	y_n	$y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_n + hf(x_n, y_n))\right]$
0.0	*2.00000*	$2.0 + 0.05[\ln(0 + 2) + \ln(0.1 + 2.0 + 0.1\ln(2))] = 2.07338$
0.1	*2.07338*	$2.07338 + 0.05[\ln(0.1 + 2.07338) + \ln(0.2 + 2.07338 + 0.1\ln(2.17338))] = 2.15494$
0.2	*2.15494*	$2.15494 + 0.05[\ln(0.2 + 2.15494) + \ln(0.3 + 2.15494 + 0.1\ln(2.35494))] = 2.24439$
0.3	*2.24439*	$2.24439 + 0.05[\ln(0.3 + 2.24439) + \ln(0.4 + 2.24439 + 0.1\ln(2.54439))] = 2.34144$
0.4	*2.34144*	$2.34144 + 0.05[\ln(0.4 + 2.34144) + \ln(0.5 + 2.34144 + 0.1\ln(2.74144))] = 2.44582$
0.5	*2.44582*	

Thus the required approximation value is $y(0.5) \approx 2.44582$

9.8. Error Analysis of Euler's Method

Here we develop the theoretical bounds for the error in Euler's method. This requires two supporting lemmas.

9.8.1. Supporting Lemmas

Lemma 1:

For all $x \geq -1$ (not just $x \geq 1$) and any positive m, then $0 \leq (1+x)^m \leq e^{mx}$.

Proof. Apply Taylor's theorem to $f(x) = e^x$ at $x_0 = 0$. For any $x > 0$:

$$e^x = 1 + x + \frac{x^2}{2}e^{\zeta}, \quad \text{for some } 0 < \zeta < x$$

Since $\frac{x^2}{2}e^{\zeta} \geq 0$, we have $1 + x \leq e^x$. Since $1 + x \geq 0$, we can raise both sides to the power m:

$$0 \leq (1+x)^m \leq (e^x)^m = e^{mx}$$

□

Lemma 2:

If s and t are positive real numbers, and $\{a_i\}_{i=0}^{k}$ is a sequence satisfying $a_0 \geq -t/s$ and

$$a_{i+1} \leq (1+s)a_i + t \quad \text{for each } i = 0, 1, \dots, k-1$$

then

$$a_{i+1} \leq e^{(i+1)s}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}$$

Proof. We expand the recursive inequality:

$$\begin{aligned}
a_{i+1} &\leq (1+s)a_i + t \\
&\leq (1+s)[(1+s)a_{i-1} + t] + t = (1+s)^2 a_{i-1} + (1+s)t + t \\
&\leq (1+s)^2[(1+s)a_{i-2} + t] + (1+s)t + t = (1+s)^3 a_{i-2} + [1 + (1+s) + (1+s)^2]t \\
&\vdots \\
&\leq (1+s)^{i+1}a_0 + [1 + (1+s) + (1+s)^2 + \cdots + (1+s)^i]t
\end{aligned}$$

The term in brackets is a finite geometric series:

$$\sum_{j=0}^{i}(1+s)^j = \frac{(1+s)^{i+1}-1}{(1+s)-1} = \frac{1}{s}[(1+s)^{i+1}-1]$$

Substituting this back in:

$$\begin{aligned} a_{i+1} &\leq (1+s)^{i+1}a_0 + \frac{t}{s}[(1+s)^{i+1}-1] \\ &= (1+s)^{i+1}\left(a_0+\frac{t}{s}\right)-\frac{t}{s} \end{aligned}$$

Applying Lemma 1 (with $1+s$ in place of x), we know $(1+s)^{i+1} \leq e^{(i+1)s}$.

$$a_{i+1} \leq e^{(i+1)s}\left(a_0+\frac{t}{s}\right)-\frac{t}{s}$$

□

9.8.2. Lipschitz Condition

To use these lemmas, the function $f(x,y)$ must satisfy a condition that bounds how quickly it can change with respect to y. This is the Lipschitz condition.

Definition 9.8.1 (Lipschitz Condition)**.** A function $f(x,y)$ is said to satisfy a **Lipschitz condition** in the variable y on a set $R = \{(x,y) : a \leq x \leq b, -\infty < y < \infty\}$ if a constant $L \geq 0$ exists such that for any two points (x,y_1) and (x,y_2) in R:

$$|f(x,y_1)-f(x,y_2)| \leq L|y_1-y_2|$$

The constant L is called a **Lipschitz constant** for f.

This condition essentially states that the slope of f in the y direction is bounded by L. If $\frac{\partial f}{\partial y}$ is continuous and bounded on R, then $L = \max|\frac{\partial f}{\partial y}|$.

Example 9.8.2. *Show that $f(x,y) = 2x+3y$ is a Lipschitz function on the domain $R = \{(x,y) : 0 \leq x \leq 1, -\infty < y < \infty\}$.*

Solution: *We need to find a constant $L \geq 0$ such that $|f(x,y_1)-f(x,y_2)| \leq L|y_1-y_2|$ for any (x,y_1) and (x,y_2) in R.*

We take the absolute difference:

$$\begin{aligned} |f(x,y_1)-f(x,y_2)| &= |(2x+3y_1)-(2x+3y_2)| \\ &= |2x+3y_1-2x-3y_2| \\ &= |3y_1-3y_2| \\ &= 3|y_1-y_2| \end{aligned}$$

In this case, the inequality $|f(x,y_1)-f(x,y_2)| \leq L|y_1-y_2|$ is satisfied with the constant $L=3$. Therefore, $f(x,y) = 2x+3y$ satisfies the Lipschitz condition with Lipschitz constant $L=3$. ***Note:*** *A common way to find L for differentiable functions is to find the maximum bound of the partial derivative with respect to y. Here:*

$$\left|\frac{\partial f}{\partial y}\right| = \left|\frac{\partial}{\partial y}(2x+3y)\right| = |3| = 3$$

Since this partial derivative is continuous and bounded by $L=3$, the function is Lipschitz.

9.8.3. Global Error Bound Theorem

Theorem 9.8.3. *Let $y(x)$ be the unique solution to the IVP $y' = f(x, y), y(x_0) = y_0$. Assume that:*

1. *$f(x, y)$ is continuous and satisfies a Lipschitz condition with constant L on the domain.*
2. *There exists a constant M such that $|y''(x)| \leq M$ for all $x \in [x_0, b]$.*

If $\omega_0, \omega_1, \ldots, \omega_N$ are the approximations generated by Euler's method with step size h, then the global error $e_i = y(x_i) - \omega_i$ at each step is bounded by:

$$|e_i| = |y(x_i) - \omega_i| \leq \frac{Mh}{2L}[e^{(x_i - x_0)L} - 1] \quad \textit{for } i = 0, 1, \ldots, N$$

Proof. The error at step $i = 0$ is $|e_0| = |y(x_0) - \omega_0| = 0$, so the formula holds. From Taylor's theorem, the exact solution $y(x_i) = y_i$ satisfies:

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(\zeta_i) = y_i + hf(x_i, y_i) + \frac{h^2}{2}y''(\zeta_i) \tag{9.8}$$

The Euler's method approximation is:

$$\omega_{i+1} = \omega_i + hf(x_i, \omega_i)$$

Subtracting these two equations gives the error at the next step, e_{i+1}:

$$e_{i+1} = y_{i+1} - \omega_{i+1} = (y_i - \omega_i) + h[f(x_i, y_i) - f(x_i, \omega_i)] + \frac{h^2}{2}y''(\zeta_i)$$
$$|e_{i+1}| \leq |y_i - \omega_i| + h|f(x_i, y_i) - f(x_i, \omega_i)| + \frac{h^2}{2}|y''(\zeta_i)|$$

Applying the Lipschitz condition ($|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|$) and the bound $|y''(x)| \leq M$:

$$|e_{i+1}| \leq |e_i| + hL|y_i - \omega_i| + \frac{h^2 M}{2}$$

$$|e_{i+1}| \leq (1 + hL)|e_i| + \frac{h^2 M}{2} \tag{9.9}$$

This is a recursive inequality. We apply Lemma 2, with $a_i = |e_i|$, $s = hL$, and $t = \frac{h^2 M}{2}$.

$$|e_i| \leq e^{i \cdot s}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}$$

Substituting back s, t, a_i, and $a_0 = |e_0| = 0$:

$$|e_i| \leq e^{i(hL)}\left(0 + \frac{h^2 M/2}{hL}\right) - \frac{h^2 M/2}{hL}$$
$$|e_i| \leq e^{ihL}\left(\frac{hM}{2L}\right) - \frac{hM}{2L}$$

Since $x_i = x_0 + ih$, we have $ih = x_i - x_0$. This gives the final bound:

$$|e_i| \leq \frac{Mh}{2L}[e^{(x_i - x_0)L} - 1]$$

□

This theorem shows that the error at any point x_i is $O(h)$. This means that if we halve the step size h, we can expect to halve the global error. The method is **convergent** because as $h \to 0$, the error bound goes to 0.

A weakness of the theorem is that M (the bound on y'') is often unknown. However, we can find y'' by differentiating the original ODE:

$$y'' = \frac{d}{dx}(f(x, y)) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}f$$

Example 9.8.4. *Consider the IVP $y' = y - x^2 + 1$ with $y(0) = 0.5$ on $0 \le x \le 2$. Find the approximation error bound for Euler's method with $h = 0.2$.*

Solution:

- ***Find L:*** *$f(x, y) = y - x^2 + 1$. We have $\frac{\partial f}{\partial y} = 1$. This is bounded by $L = 1$.*
- ***Find M:*** *We need to find y''. The exact solution is $y(x) = (x + 1)^2 - 0.5e^x$.*

$$y'(x) = 2(x + 1) - 0.5e^x$$

$$y''(x) = 2 - 0.5e^x$$

  We need to bound $|y''(x)|$ on $[0, 2]$. The function $y''(x)$ is decreasing. $y''(0) = 1.5$. $y''(2) = 2 - 0.5e^2 \approx 2 - 3.69 = -1.69$. The maximum absolute value is at $x = 2$. So, $M = |2 - 0.5e^2| \approx 1.694$. (The slide uses $0.5e^2 - 2$, which is the same value).

- ***Apply Error Bound:*** *Using $h = 0.2, L = 1, M = 0.5e^2 - 2$:*

$$|y_i - \omega_i| \le \frac{(0.2)(0.5e^2 - 2)}{2(1)}[e^{(x_i - 0)\cdot 1} - 1]$$

$$|y_i - \omega_i| \le 0.1(0.5e^2 - 2)(e^{x_i} - 1)$$

  For example, at the final step $x_i = 2$:

$$|y(2) - \omega_{10}| \le 0.1(0.5e^2 - 2)(e^2 - 1) \approx 0.1(1.694)(6.389) \approx 1.082$$

9.9. Taylor's Series Method

Euler's method is a Taylor method of order 1. We can achieve higher accuracy by including more terms from the Taylor series.

Given the IVP $\frac{dy}{dx} = f(x, y)$ with $y(x_0) = y_0$: We can find the higher derivatives of y by successive differentiation using the chain rule:

$$\begin{aligned} y'(x) &= f(x, y) \\ y''(x) &= f_x + f_y \frac{dy}{dx} = f_x + f_y f \\ y'''(x) &= f_{xx} + f_{xy} f + f_y f_x + (f_{yx} + f_{yy} f) f + f_y (f_x + f_y f) \\ &\vdots \end{aligned}$$

The **Taylor's Series Method of Order** n is given by:

$$y_{i+1} = y_i + h y'(x_i) + \frac{h^2}{2!} y''(x_i) + \cdots + \frac{h^n}{n!} y^{(n)}(x_i)$$

The local truncation error for this method is $O(h^{n+1})$.

Example 9.9.1. *Using the 4th-order Taylor series method, find $y(0.1)$ for the IVP:*

$$y' = x^2 y - 1, \quad y(0) = 1$$

Solution: *Here $x_0 = 0, y_0 = 1, h = 0.1$. We need derivatives up to order 4.*

- *$y' = x^2 y - 1$*
- *$y'' = 2xy + x^2 y'$*

- $y''' = 2y + 2xy' + 2xy' + x^2y'' = 2y + 4xy' + x^2y''$
- $y^{(4)} = 2y' + 4y' + 4xy'' + 2xy'' + x^2y''' = 6y' + 6xy'' + x^2y'''$

Now, evaluate these at the starting point $(x_0, y_0) = (0, 1)$*:*

- $y'(0) = (0)^2(1) - 1 = -1$
- $y''(0) = 2(0)(1) + (0)^2(-1) = 0$
- $y'''(0) = 2(1) + 4(0)(-1) + (0)^2(0) = 2$
- $y^{(4)}(0) = 6(-1) + 6(0)(0) + (0)^2(2) = -6$

Plug these into the 4th-order Taylor formula:

$$\begin{aligned}
y_1 &= y_0 + hy'(0) + \frac{h^2}{2}y''(0) + \frac{h^3}{6}y'''(0) + \frac{h^4}{24}y^{(4)}(0) \\
y(0.1) \approx y_1 &= 1 + (0.1)(-1) + \frac{(0.1)^2}{2}(0) + \frac{(0.1)^3}{6}(2) + \frac{(0.1)^4}{24}(-6) \\
&= 1 - 0.1 + 0 + \frac{0.001}{3} - \frac{0.0001}{4} \\
&= 1 - 0.1 + 0.00033333 - 0.000025 = 0.90030833
\end{aligned}$$

So, $y(0.1) \approx 0.90031$*. To find* $y(0.2)$*, we would repeat this entire process starting from* $(x_1, y_1) = (0.1, 0.90031)$.

9.10. Runge-Kutta (RK) Methods

9.10.1. Motivation

The Taylor series method is very accurate but has a major practical disadvantage: it requires finding, programming, and evaluating several higher-order partial derivatives of $f(x, y)$. This can be extremely difficult or impossible.

Runge-Kutta methods are designed to achieve the same accuracy as high-order Taylor methods, but require *only* evaluations of the first-order derivative function $f(x, y)$ at different points within the step.

9.10.2. Background: Taylor's Series in Two Variables

The derivation of RK methods relies on the Taylor series expansion for a function of two variables, $f(t, y)$, around (t_0, y_0):

$$\begin{aligned}
P_n(t, y) = f(t_0, y_0) &+ \left[(t - t_0)\frac{\partial f}{\partial t} + (y - y_0)\frac{\partial f}{\partial y}\right]_{(t_0, y_0)} \\
&+ \left[\frac{(t - t_0)^2}{2}\frac{\partial^2 f}{\partial t^2} + (t - t_0)(y - y_0)\frac{\partial^2 f}{\partial t \partial y} + \frac{(y - y_0)^2}{2}\frac{\partial^2 f}{\partial y^2}\right]_{(t_0, y_0)} + \cdots
\end{aligned}$$

This is used to expand $f(x + h, y + k)$ in the derivation and match terms with the Taylor series for $y(x + h)$.

9.10.3. Second-Order Runge-Kutta (RK2)

The goal is to approximate $y(x_{n+1})$ by $y_n + h \cdot \text{(average slope)}$. A general form for a second-order method is:

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$

where k_1 and k_2 represent slope estimates. The most common choice is the **Modified Euler's (Heun's) Method**:

$$\begin{aligned} k_1 &= hf(x_n, y_n) \quad \text{(Slope at the beginning)} \\ k_2 &= hf(x_n + h, y_n + k_1) \quad \text{(Slope at the predicted end)} \end{aligned}$$

This method has a local truncation error of $O(h^3)$, just like the 2nd-order Taylor method.

9.10.4. Fourth-Order Runge-Kutta (RK4)

The most widely used numerical method for ODEs is the **Classic 4th-Order Runge-Kutta Method**. It has a local truncation error of $O(h^5)$ and a global error of $O(h^4)$. It requires four evaluations of $f(x, y)$ per step:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where the k_i values are:

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \\ k_3 &= hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \\ k_4 &= hf(x_n + h, y_n + k_3) \end{aligned}$$

This formula is a weighted average: k_1 is the slope at the beginning, k_2 and k_3 are two estimates of the slope at the midpoint, and k_4 is the slope at the end.

Example 9.10.1. *Use the RK4 method with* $h = 0.2$ *to find* $y(0.2)$ *for the IVP:*

$$\frac{dy}{dx} = x + y, \quad y(0) = 1$$

Solution: *Here* $f(x, y) = x + y$, $x_0 = 0$, $y_0 = 1$, *and* $h = 0.2$.

- $k_1 = hf(x_0, y_0) = 0.2(0 + 1) = 0.2$
- $k_2 = hf(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}) = 0.2f(0.1, 1 + \frac{0.2}{2}) = 0.2f(0.1, 1.1) = 0.2(0.1 + 1.1) = 0.24$
- $k_3 = hf(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}) = 0.2f(0.1, 1 + \frac{0.24}{2}) = 0.2f(0.1, 1.12) = 0.2(0.1 + 1.12) = 0.244$
- $k_4 = hf(x_0 + h, y_0 + k_3) = 0.2f(0.2, 1 + 0.244) = 0.2f(0.2, 1.244) = 0.2(0.2 + 1.244) = 0.2888$

Now, we find y_1*:*

$$\begin{aligned} y_1 &= y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ &= 1 + \frac{1}{6}(0.2 + 2(0.24) + 2(0.244) + 0.2888) \\ &= 1 + \frac{1}{6}(0.2 + 0.48 + 0.488 + 0.2888) \\ &= 1 + \frac{1}{6}(1.4568) = 1 + 0.2428 = 1.2428 \end{aligned}$$

So, $y(0.2) \approx 1.2428$. *(Note: The exact solution is* $y(0.2) = 2e^{0.2} - 0.2 - 1 \approx 1.2428055$, *so this is a very accurate result in just one step).*

9.11. Higher-Order Equations and Systems of DEs

Numerical methods for first-order equations can be extended to solve systems of first-order equations. This is powerful because any m-th order ODE can be converted into a system of m first-order ODEs.

9.11.1. Converting Higher-Order ODEs to Systems

A general m-th order IVP has the form:

$$y^{(m)}(x) = f(x, y, y', \dots, y^{(m-1)})$$

with initial conditions $y(a) = \alpha_1, y'(a) = \alpha_2, \dots, y^{(m-1)}(a) = \alpha_m$.

We perform a substitution by defining a new set of variables:

$$\begin{aligned}
y_1(x) &= y(x) \\
y_2(x) &= y'(x) \\
y_3(x) &= y''(x) \\
&\vdots \\
y_m(x) &= y^{(m-1)}(x)
\end{aligned}$$

Differentiating these, we get a system of m first-order ODEs:

$$\begin{aligned}
\frac{dy_1}{dx} &= y'(x) = y_2(x) \\
\frac{dy_2}{dx} &= y''(x) = y_3(x) \\
&\vdots \\
\frac{dy_{m-1}}{dx} &= y^{(m-1)}(x) = y_m(x) \\
\frac{dy_m}{dx} &= y^{(m)}(x) = f(x, y_1, y_2, \dots, y_m)
\end{aligned}$$

The initial conditions become:

$$y_1(a) = \alpha_1, \quad y_2(a) = \alpha_2, \quad \dots, \quad y_m(a) = \alpha_m$$

9.11.2. RK4 for Systems of Equations

The RK4 method is generalized for a system of m equations. We use vector notation: $\mathbf{w}_j \approx \mathbf{y}(x_j)$, where $\mathbf{w}_j = [w_{1,j}, w_{2,j}, \dots, w_{m,j}]^T$. For each step j, we must calculate m components for each of the four k-stages. For $i = 1, 2, \dots, m$:

$$\begin{aligned}
k_{1,i} &= hf_i(x_j, w_{1,j}, w_{2,j}, \dots, w_{m,j}) \\
k_{2,i} &= hf_i\left(x_j + \frac{h}{2}, w_{1,j} + \frac{1}{2}k_{1,1}, w_{2,j} + \frac{1}{2}k_{1,2}, \dots, w_{m,j} + \frac{1}{2}k_{1,m}\right) \\
k_{3,i} &= hf_i\left(x_j + \frac{h}{2}, w_{1,j} + \frac{1}{2}k_{2,1}, w_{2,j} + \frac{1}{2}k_{2,2}, \dots, w_{m,j} + \frac{1}{2}k_{2,m}\right) \\
k_{4,i} &= hf_i\left(x_j + h, w_{1,j} + k_{3,1}, w_{2,j} + k_{3,2}, \dots, w_{m,j} + k_{3,m}\right)
\end{aligned}$$

After all $4m$ k-values are found, we update each component of $\mathbf{w}$:

$$w_{i,j+1} = w_{i,j} + \frac{1}{6}(k_{1,i} + 2k_{2,i} + 2k_{3,i} + k_{4,i}) \quad \text{for } i = 1, \dots, m$$

Example 9.11.1. *Convert the 2nd-order IVP*

$$y'' - 2y' + 2y = e^{2x}\sin(x), \quad y(0) = -0.4, \quad y'(0) = -0.6$$

into a system of first-order IVPs and use RK4 with $h = 0.1$ to approximate $y(0.1)$ and $y'(0.1)$.

Solution: 1. Convert to a System: *Let $y_1(x) = y(x)$ and $y_2(x) = y'(x)$. The system becomes:*

$$\begin{aligned}
y_1' &= y_2 \\
y_2' &= e^{2x}\sin(x) - 2y + 2y' = e^{2x}\sin(x) - 2y_1 + 2y_2
\end{aligned}$$

So, $f_1(x, y_1, y_2) = y_2$ and $f_2(x, y_1, y_2) = e^{2x}\sin(x) - 2y_1 + 2y_2$. The initial conditions are $y_1(0) = -0.4$ and $y_2(0) = -0.6$. We use $w_{1,0} = -0.4$ and $w_{2,0} = -0.6$. $h = 0.1$.
2. Calculate k-values for $j = 0$:

- ***Stage 1:***
- $k_{1,1} = hf_1(x_0, w_{1,0}, w_{2,0}) = 0.1(w_{2,0}) = 0.1(-0.6) = -0.06$
- $k_{1,2} = hf_2(x_0, w_{1,0}, w_{2,0}) = 0.1[e^0\sin(0) - 2(-0.4) + 2(-0.6)] = 0.1[0 + 0.8 - 1.2] = -0.04$
- ***Stage 2:*** *(Use $x = 0.05, w_1 = -0.4 + k_{1,1}/2 = -0.43, w_2 = -0.6 + k_{1,2}/2 = -0.62$)*
- $k_{2,1} = hf_1(\dots) = 0.1(w_2) = 0.1(-0.62) = -0.062$
- $k_{2,2} = hf_2(\dots) = 0.1[e^{0.1}\sin(0.05) - 2(-0.43) + 2(-0.62)] \approx -0.032476$
- ***Stage 3:*** *(Use $x = 0.05, w_1 = -0.4 + k_{2,1}/2 = -0.431, w_2 = -0.6 + k_{2,2}/2 = -0.616238$)*
- $k_{3,1} = hf_1(\dots) = 0.1(w_2) = 0.1(-0.616238) \approx -0.061628$
- $k_{3,2} = hf_2(\dots) = 0.1[e^{0.1}\sin(0.05) - 2(-0.431) + 2(-0.616238)] \approx -0.031524$
- ***Stage 4:*** *(Use $x = 0.1, w_1 = -0.4 + k_{3,1} = -0.461628, w_2 = -0.6 + k_{3,2} = -0.631524$)*
- $k_{4,1} = hf_1(\dots) = 0.1(w_2) = 0.1(-0.631524) \approx -0.063152$
- $k_{4,2} = hf_2(\dots) = 0.1[e^{0.2}\sin(0.1) - 2(-0.461628) + 2(-0.631524)] \approx -0.021786$

3. Update Approximations:

$$\begin{aligned} w_{1,1} &= w_{1,0} + \frac{1}{6}[k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}] \\ &= -0.4 + \frac{1}{6}[-0.06 + 2(-0.062) + 2(-0.061628) + (-0.063152)] \approx -0.461733 \\ w_{2,1} &= w_{2,0} + \frac{1}{6}[k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}] \\ &= -0.6 + \frac{1}{6}[-0.04 + 2(-0.032476) + 2(-0.031524) + (-0.021786)] \approx -0.631631 \end{aligned}$$

The approximations at $x = 0.1$ are $y(0.1) \approx w_{1,1} = -0.461733$ and $y'(0.1) \approx w_{2,1} = -0.631631$.

9.12. Introduction to Multistep Methods

Unlike one-step methods (like Euler or Runge-Kutta) which only use information from the previous point x_i to find the value at x_{i+1}, **m-step methods** use information from the previous m points $(x_i, x_{i-1}, \dots, x_{i+1-m})$ to find the next value.

This use of more historical information can lead to more efficient and accurate methods, as they can build a better model of the function's behavior.

9.12.1. General Form of an m-Step Method

Given the initial-value problem

$$y' = f(x, y), \quad a \le x \le b, \quad y(a) = \alpha$$

An m-step method to find the approximation w_{i+1} at x_{i+1} is given by the general formula:

$$\begin{aligned} w_{i+1} = &a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ &+ h\big[b_m f(x_{i+1}, w_{i+1}) + b_{m-1}f(x_i, w_i) + \cdots + b_0 f(x_{i+1-m}, w_{i+1-m})\big] \end{aligned}$$

- This formula is used for $i = m-1, m, \ldots, N-1$, where $h = (b-a)/N$.
- The coefficients $a_0, \ldots, a_{m-1}$ and $b_0, \ldots, b_m$ are constants that define the specific method.
- Because the method requires m previous points, it cannot be used at the beginning of the interval. We need to use a one-step method (like RK4) or known exact values to generate the necessary starting values:

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \ldots, \quad w_{m-1} = \alpha_{m-1}$$

- **Explicit (Open) Method:** If $b_m = 0$, the formula only uses known values from the past. w_{i+1} can be calculated directly.
- **Implicit (Closed) Method:** If $b_m \neq 0$, the unknown w_{i+1} appears on both sides of the equation (inside the $f(\ldots)$ term). This requires solving an equation to find w_{i+1}, often using algebraic iteration or a predictor-corrector approach.

9.12.2. Adams-Bashforth and Adams-Moulton Methods

The two most common families of multistep methods are the explicit **Adams-Bashforth** methods and the implicit **Adams-Moulton** methods. Here are two high-order examples:

Example 9.12.1 (Fourth-Order Adams-Bashforth (Explicit))**.** *Given the starting values w_0, w_1, w_2, w_3:*

$$\begin{aligned} w_{i+1} = w_i + \frac{h}{24}\big[&55f(x_i, w_i) - 59f(x_{i-1}, w_{i-1}) \\ &+ 37f(x_{i-2}, w_{i-2}) - 9f(x_{i-3}, w_{i-3})\big] \end{aligned}$$

This is a 4-step explicit method ($m = 4, b_4 = 0$).

Example 9.12.2 (Fourth-Order Adams-Moulton (Implicit))**.** *Given the starting values w_0, w_1, w_2:*

$$\begin{aligned} w_{i+1} = w_i + \frac{h}{24}\big[&9f(x_{i+1}, w_{i+1}) + 19f(x_i, w_i) \\ &- 5f(x_{i-1}, w_{i-1}) + f(x_{i-2}, w_{i-2})\big] \end{aligned}$$

This is a 3-step implicit method ($m = 3$, but it's called "four-step" in some contexts due to its $O(h^4)$ error. Here, $b_3 = 9/24 \neq 0$).

9.13. Derivation of Multistep Methods

The foundation for all multistep methods is the exact integral form of the ODE:

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t, y(t))\, dt$$

The core idea is to approximate the (usually non-integrable) function $f(t, y(t))$ with a polynomial $P(t)$.

$$y(t_{i+1}) \approx w_i + \int_{t_i}^{t_{i+1}} P(t)\, dt$$

9.13.1. Derivation of Adams-Bashforth (Explicit) Methods

To derive an explicit m-step method, we form a backward-difference interpolating polynomial $P_{m-1}(t)$ through the m previously calculated points:

$$(t_i, f_i), (t_{i-1}, f_{i-1}), \ldots, (t_{i+1-m}, f_{i+1-m})$$

where $f_k \approx f(t_k, y(t_k))$. The error from this interpolation is

$$f(t, y(t)) - P_{m-1}(t) = \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(t - t_i)(t - t_{i-1}) \cdots (t - t_{i+1-m})$$

Integrating this full expression from t_i to t_{i+1} gives the approximation and the error. We use the variable substitution $t = t_i + sh$, which means $dt = h\,ds$, and the integration limits change from $[t_i, t_{i+1}]$ to $[0, 1]$.

$$\begin{aligned}\int_{t_i}^{t_{i+1}} f(t, y(t))\,dt &= \int_{t_i}^{t_{i+1}} P_{m-1}(t)\,dt + \int_{t_i}^{t_{i+1}} (\text{Error Term})\,dt \\ &= \int_0^1 P_{m-1}(t_i + sh)h\,ds + \int_0^1 (\text{Error Term})h\,ds\end{aligned}$$

The backward-difference polynomial $P_{m-1}(t)$ can be written in terms of s and the backward-difference operator ∇:

$$P_{m-1}(t_i + sh) = \sum_{k=0}^{m-1} (-1)^k \binom{-s}{k} \nabla^k f(t_i, y(t_i))$$

The integral becomes:

$$\begin{aligned}\int_{t_i}^{t_{i+1}} f(t, y(t))\,dt = &\sum_{k=0}^{m-1} \nabla^k f_i \left(h(-1)^k \int_0^1 \binom{-s}{k}\,ds \right) + \text{Error} \\ =&h\left[f_i + \frac{1}{2}\nabla f_i + \frac{5}{12}\nabla^2 f_i + \frac{3}{8}\nabla^3 f_i + \cdots \right] + \text{Error}\end{aligned}$$

The error term is also integrated:

$$\text{Error} = \frac{h^{m+1}}{m!} \int_0^1 s(s+1)\cdots(s+m-1) f^{(m)}(\xi_s, y(\xi_s))\,ds$$

Since $s(s+1)\cdots(s+m-1)$ does not change sign on $[0, 1]$, we can use the Weighted Mean Value Theorem for Integrals to simplify this error term to:

$$\text{Error} = h^{m+1} f^{(m)}(\mu_i, y(\mu_i))(-1)^m \int_0^1 \binom{-s}{m}\,ds$$

for some $\mu_i \in (t_{i+1-m}, t_{i+1})$.

Example 9.13.1 (Derivation of the Three-Step Adams-Bashforth Method)**.** *Use the general derivation with $m = 3$ to find the 3-step Adams-Bashforth method.*

Solution: *We take the first three terms ($k = 0, 1, 2$) from the integrated polynomial:*

$$y(t_{i+1}) \approx y(t_i) + h\left[f_i + \frac{1}{2}\nabla f_i + \frac{5}{12}\nabla^2 f_i \right]$$

Now, we expand the backward-difference operators ∇:

- $f_i = f(t_i, y(t_i))$
- $\nabla f_i = f(t_i, y(t_i)) - f(t_{i-1}, y(t_{i-1}))$
- $\nabla^2 f_i = f(t_i, y(t_i)) - 2f(t_{i-1}, y(t_{i-1})) + f(t_{i-2}, y(t_{i-2}))$

Substitute these into the equation:

$$\begin{aligned}y(t_{i+1}) &\approx y(t_i) + h\left\{ f_i + \frac{1}{2}[f_i - f_{i-1}] + \frac{5}{12}[f_i - 2f_{i-1} + f_{i-2}] \right\} \\ &= y(t_i) + h\left[\left(1 + \frac{1}{2} + \frac{5}{12}\right) f_i + \left(-\frac{1}{2} - \frac{10}{12}\right) f_{i-1} + \left(\frac{5}{12}\right) f_{i-2} \right] \\ &= y(t_i) + h\left[\left(\frac{12 + 6 + 5}{12}\right) f_i + \left(\frac{-6 - 10}{12}\right) f_{i-1} + \left(\frac{5}{12}\right) f_{i-2} \right] \\ &= y(t_i) + \frac{h}{12}[23f(t_i, y(t_i)) - 16f(t_{i-1}, y(t_{i-1})) + 5f(t_{i-2}, y(t_{i-2}))]\end{aligned}$$

*This gives the **three-step Adams-Bashforth method**:*

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2,$$
$$w_{i+1} = w_i + \frac{h}{12}[23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})]$$

9.13.2. Local Truncation Error (LTE)

The LTE is the error a method makes in a single step, assuming all previous values ($w_k = y(t_k)$) are perfectly accurate.

Definition 9.13.2. For an m-step method

$$\begin{aligned} w_{i+1} =& a_{m-1}w_i + \cdots + a_0 w_{i+1-m} \\ &+ h[b_m f(t_{i+1}, w_{i+1}) + \cdots + b_0 f(t_{i+1-m}, w_{i+1-m})] \end{aligned}$$

the **local truncation error** $\tau_{i+1}(h)$ is defined by substituting the true solution $y(t)$ into the formula:

$$\begin{aligned} \tau_{i+1}(h) =& \frac{y(t_{i+1}) - a_{m-1}y(t_i) - \cdots - a_0 y(t_{i+1-m})}{h} \\ &- \left[b_m f(t_{i+1}, y(t_{i+1})) + \cdots + b_0 f(t_{i+1-m}, y(t_{i+1-m}))\right] \end{aligned}$$

Note: Since $y'(t) = f(t, y(t))$, the second term is just $\sum b_k y'(t_k)$. The LTE is essentially how well the difference equation for y matches the derivative y'.

9.14. Summary of Adams-Bashforth (Explicit) Methods

- **Two-step explicit method:** ($m = 2$)

$$w_0 = \alpha, \quad w_1 = \alpha_1$$
$$w_{i+1} = w_i + \frac{h}{2}\left[3f(t_i, w_i) - f(t_{i-1}, w_{i-1})\right]$$
$$\text{Local Truncation Error: } \tau_{i+1}(h) = \frac{5}{12}y'''(\mu_i)h^2$$

- **Three-step explicit method:** ($m = 3$)

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2$$
$$w_{i+1} = w_i + \frac{h}{12}\left[23f(t_i, w_i) - 16f(t_{i-1}, w_{i-1}) + 5f(t_{i-2}, w_{i-2})\right]$$
$$\text{Local Truncation Error: } \tau_{i+1}(h) = \frac{3}{8}y^{(4)}(\mu_i)h^3$$

- **Four-step explicit method:** ($m = 4$)

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3$$
$$w_{i+1} = w_i + \frac{h}{24}\left[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})\right]$$
$$\text{Local Truncation Error: } \tau_{i+1}(h) = \frac{251}{720}y^{(5)}(\mu_i)h^4$$

- **Five-step explicit method:** ($m = 5$)

$$w_0 = \alpha, \ldots, w_4 = \alpha_4$$
$$w_{i+1} = w_i + \frac{h}{720}[1901f(t_i, w_i) - 2774f(t_{i-1}, w_{i-1}) + 2616f(t_{i-2}, w_{i-2}) - \cdots]$$
$$\text{Local Truncation Error: } \tau_{i+1}(h) = \frac{95}{288}y^{(6)}(\mu_i)h^5$$

9.15. Summary of Adams-Moulton (Implicit) Methods

Implicit methods are derived in a similar way, but the interpolating polynomial includes the point (t_{i+1}, f_{i+1}) as well. This generally leads to smaller error constants and better stability.

- **Two-step implicit method:** $(m = 2)$

$$w_0 = \alpha, \quad w_1 = \alpha_1$$
$$w_{i+1} = w_i + \frac{h}{12}\big[5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})\big]$$
$$\text{Local Truncation Error: } \tau_{i+1}(h) = -\frac{1}{24}y^{(4)}(\mu_i)h^3$$

- **Three-step implicit method:** $(m = 3)$

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2$$
$$w_{i+1} = w_i + \frac{h}{24}\big[9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})\big]$$
$$\text{Local Truncation Error: } \tau_{i+1}(h) = -\frac{19}{720}y^{(5)}(\mu_i)h^4$$

- **Four-step implicit method:** $(m = 4)$

$$w_0 = \alpha, \ldots, w_3 = \alpha_3$$
$$w_{i+1} = w_i + \frac{h}{720}[251f(t_{i+1}, w_{i+1}) + 646f(t_i, w_i) - 264f(t_{i-1}, w_{i-1}) + \cdots]$$
$$\text{Local Truncation Error: } \tau_{i+1}(h) = -\frac{3}{160}y^{(6)}(\mu_i)h^5$$

9.16. Predictor-Corrector Methods

Implicit methods are more accurate but have a major practical problem: w_{i+1} is on both sides of the equation. If $f(t, y)$ is nonlinear, we cannot solve for w_{i+1} algebraically.

The solution is to use a **Predictor-Corrector** pair:

1. **Predict (P):** Use an *explicit* method (like Adams-Bashforth) to get a first estimate, $w_{i+1}^{(p)}$.
2. **Correct (C):** Use this estimate on the right-hand side of an *implicit* method (like Adams-Moulton) to get a final, more refined value, $w_{i+1}^{(c)}$.

A very common pair is the 4th-order Adams-Bashforth (Predictor) and 4th-order Adams-Moulton (Corrector).

Example 9.16.1 (Comparing AB and AM Methods)**.** *Consider the IVP $y' = y - t^2 + 1$, $0 \le t \le 2$, $y(0) = 0.5$. Use $h = 0.2$ and exact values for starting points to compare the 4-step Adams-Bashforth (AB) and 3-step Adams-Moulton (AM) methods.*

Solution: *The exact solution is $y(t) = (t+1)^2 - 0.5e^t$. We use this to get the starting values:*

- $w_0 = y(0.0) = 0.5000000$
- $w_1 = y(0.2) = 0.8292986$
- $w_2 = y(0.4) = 1.2140877$
- $w_3 = y(0.6) = 1.6489406$

(a) Adams-Bashforth 4-Step (Explicit):

$$w_{i+1} = w_i + \frac{h}{24}\left[55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}\right]$$

For $i = 3$ (to find w_4 at $t = 0.8$):

$$\begin{aligned}
f_3 &= f(0.6, w_3) = 1.6489406 - 0.6^2 + 1 = 2.2889406\\
f_2 &= f(0.4, w_2) = 1.2140877 - 0.4^2 + 1 = 2.0540877\\
f_1 &= f(0.2, w_1) = 0.8292986 - 0.2^2 + 1 = 1.7892986\\
f_0 &= f(0.0, w_0) = 0.5000000 - 0.0^2 + 1 = 1.5000000
\end{aligned}$$

$$w_4 = 1.6489406 + \frac{0.2}{24}\left[55(2.2889) - 59(2.0541) + 37(1.7893) - 9(1.5000)\right] \approx 2.1273124$$

(b) Adams-Moulton 3-Step (Implicit):

$$w_{i+1} = w_i + \frac{h}{24}\left[9f(t_{i+1}, w_{i+1}) + 19f_i - 5f_{i-1} + f_{i-2}\right]$$

For $i = 2$ (to find w_3 at $t = 0.6$):

$$w_3 = w_2 + \frac{0.2}{24}\left[9(w_3 - 0.6^2 + 1) + 19f_2 - 5f_1 + f_0\right]$$

Since $f(t, y)$ is linear in y, we can solve for w_3 directly.

$$w_3\left(1 - \frac{0.2 \cdot 9}{24}\right) = w_2 + \frac{0.2}{24}\left[9(-0.36 + 1) + 19f_2 - 5f_1 + f_0\right]$$

Solving this gives $w_3 \approx 1.6489341$. ***Results:*** *The table shows the AM method is significantly more accurate.*

		Adams-Bashforth (4-step)		*Adams-Moulton (3-step)*	
t_i	*Exact*	w_i	*Error*	w_i	*Error*
0.0	*0.5000000*				
0.2	*0.8292986*				
0.4	*1.2140877*				
0.6	*1.6489406*	*(start)*		*1.6489341*	*0.0000065*
0.8	*2.1272295*	*2.1273124*	*0.0000828*	*2.1272136*	*0.0000160*
1.0	*2.6408591*	*2.6410810*	*0.0002219*	*2.6408298*	*0.0000293*
...	*...*	*...*	*...*	*...*	*...*
2.0	*5.3054720*	*5.3075838*	*0.0021119*	*5.3052587*	*0.0002132*

9.17. Stability, Consistency, and Convergence

For a numerical method to be reliable, it must have three key properties.

Definition 9.17.1. Consistency: A method is **consistent** if its local truncation error $\tau_i(h)$ approaches zero as the step size h approaches zero.

$$\lim_{h\to 0} \max_{1\le i\le N} |\tau_i(h)| = 0$$

This means the formula correctly represents the differential equation as h gets small.

Convergence: A method is **convergent** if the numerical solution w_i approaches the true solution $y(t_i)$ at every point as the step size h approaches zero.

$$\lim_{h\to 0} \max_{1\le i\le N} |w_i - y(t_i)| = 0$$

This is the ultimate goal: a smaller step size should give a better answer.

Stability: A method is **stable** if small errors introduced at one step (like round-off error) do not grow uncontrollably in subsequent steps.

Theorem 9.17.2 (Stability Theorem - Simplified)**.** *Suppose an IVP is approximated by a one-step method $w_{i+1} = w_i + h\phi(t_i, w_i, h)$. If ϕ is continuous and satisfies a Lipschitz condition in w, then:*

- *The method is **stable**.*
- *The method is **convergent** if and only if it is **consistent**.*

Consistency is shown by checking if $\phi(t, y, 0) = f(t, y)$.

Example 9.17.3. *Show that Euler's method is convergent.*

Solution: *From the global error bound theorem, we have:*

$$\max_{1\le i\le N} |w_i - y(t_i)| \le \frac{Mh}{2L}|e^{L(b-a)} - 1|$$

Taking the limit as $h \to 0$:

$$\lim_{h\to 0} \max_{1\le i\le N} |w_i - y(t_i)| \le \lim_{h\to 0} \left[\frac{M|e^{L(b-a)} - 1|}{2L} \cdot h \right] = 0$$

Since the limit is 0, Euler's method is convergent. The rate of convergence is $O(h)$.

Example 9.17.4. *Verify that the Modified Euler method is stable and convergent.*

Solution: *The method is $w_{i+1} = w_i + h \cdot \phi(t_i, w_i, h)$, where*

$$\phi(t, w, h) = \frac{1}{2}f(t, w) + \frac{1}{2}f(t + h, w + hf(t, w))$$

1. Consistency: *We check the consistency condition $\phi(t, w, 0) = f(t, w)$:*

$$\phi(t, w, 0) = \frac{1}{2}f(t, w) + \frac{1}{2}f(t + 0, w + 0 \cdot f(t, w)) = \frac{1}{2}f(t, w) + \frac{1}{2}f(t, w) = f(t, w)$$

The method is consistent.
2. Stability: *We check if ϕ is Lipschitz in w. Assume f is Lipschitz with constant L.*

$$\begin{aligned}
|\phi(t, w, h) - \phi(t, \bar{w}, h)| &= \left| \frac{1}{2}[f(t, w) - f(t, \bar{w})] + \frac{1}{2}[f(t + h, w + hf(t, w)) - f(t + h, \bar{w} + hf(t, \bar{w}))] \right| \\
&\le \frac{1}{2}|f(t, w) - f(t, \bar{w})| + \frac{1}{2}|f(t + h, \ldots) - f(t + h, \ldots)| \\
&\le \frac{1}{2}L|w - \bar{w}| + \frac{1}{2}L|(w + hf(t, w)) - (\bar{w} + hf(t, \bar{w}))| \\
&\le \frac{1}{2}L|w - \bar{w}| + \frac{1}{2}L|(w - \bar{w}) + h(f(t, w) - f(t, \bar{w}))| \\
&\le \frac{1}{2}L|w - \bar{w}| + \frac{1}{2}L|w - \bar{w}| + \frac{1}{2}Lh|f(t, w) - f(t, \bar{w})| \\
&\le L|w - \bar{w}| + \frac{1}{2}Lh(L|w - \bar{w}|) \\
&= \left(L + \frac{hL^2}{2} \right) |w - \bar{w}|
\end{aligned}$$

*For a given $h_0 > 0$, ϕ is Lipschitz with constant $L' = L + \frac{h_0 L^2}{2}$. Since ϕ is continuous and Lipschitz, the Stability Theorem implies the method is **stable**.*
3. Convergence: *Since the method is both consistent and stable, it is **convergent**.*

Practice Questions

1. **Existence and Uniqueness of Solutions I**
   Show that each of the following initial-value problems has a unique solution, and find the solution.

   a. $y' = y\cos t, \quad 0 \le t \le 1, \quad y(0) = 1.$

   b. $y' = \frac{2}{t}y + t^2e^t, \quad 1 \le t \le 2, \quad y(1) = 0.$

   c. $y' = -\frac{2}{t}y + t^2e^t, \quad 1 \le t \le 2, \quad y(1) = \sqrt{2}e.$

   d. $y' = \frac{4t^3y}{1+t^4}, \quad 0 \le t \le 1, \quad y(0) = 1.$

2. **Existence and Uniqueness of Solutions II**
   Show that the following initial-value problems have unique solutions in a given interval, and find the solution:

   a. $y' = \frac{2-2ty}{t^2+1}, \quad 1 \le t \le 2, \quad y(1) = 2$

   b. $y' = \frac{y^2+y}{t}, \quad 1 \le t \le 3, \quad y(1) = -2$

   c. $y' = -y + t^2, \quad 0 \le t \le 2, \quad y(0) = 1$

3. **Euler's Method I**
   Use Euler's method to approximate the solutions for each of the following initial-value problems.

   a. $y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0$, with $h = 0.5$

   b. $y' = 1 + (t-y)^2, \quad 2 \le t \le 3, \quad y(2) = 1$, with $h = 0.5$

   c. $y' = 1 + y/t, \quad 1 \le t \le 2, \quad y(1) = 2$, with $h = 0.25$

   d. $y' = \cos 2t + \sin 3t, \quad 0 \le t \le 1, \quad y(0) = 1$, with $h = 0.25$

4. **Euler's Method II**
   Use Euler's method to approximate the solutions for each of the following initial-value problems:

   a. $y' = (y^2 + y)/t, \quad 1 \le t \le 3, \quad y(1) = -2$, with $h = 0.5$

   b. $y' = \sin t + e^{-t}, \quad 0 \le t \le 1, \quad y(0) = 0$, with $h = 0.25$

   c. $y' = (t + 2t^3)y^3 - ty, \quad 0 \le t \le 2, \quad y(0) = 1$, with $h = 0.2$

5. **Euler's Method with Interpolation**
   Given the initial-value problem

$$y' = \frac{1}{t^2} - \frac{y}{t} - y^2, \quad 1 \le t \le 2, \quad y(1) = -1,$$

   with exact solution $y(t) = -1/t$:

   a. Use Euler's method with $h = 0.05$ to approximate the solution, and compare it with the actual values of y.

   b. Use the answers generated in part (a) and linear interpolation to approximate the following values of y, and compare them to the actual values.

      i. $y(1.052)$ **ii.** $y(1.555)$ **iii.** $y(1.978)$

6. **Modified Euler Method**
   Use the Modified Euler method to approximate the solutions to each of the following initial-value problems.

   a. $y' = y/t - (y/t)^2, \quad 1 \le t \le 2, \quad y(1) = 1$, with $h = 0.1$; actual solution $y(t) = \frac{t}{1+\ln t}$.

   b. $y' = 1 + y/t + (y/t)^2, \quad 1 \le t \le 3, \quad y(1) = 0$, with $h = 0.2$; actual solution $y(t) = t\tan(\ln t)$.

   c. $y' = -(y+1)(y+3), \quad 0 \le t \le 2, \quad y(0) = -2$, with $h = 0.2$; actual solution $y(t) = -3 + 2(1 + e^{-2t})^{-1}$.

   d. $y' = -5y + 5t^2 + 2t, \quad 0 \le t \le 1, \quad y(0) = \frac{1}{3}$, with $h = 0.1$; actual solution $y(t) = t^2 + \frac{1}{3}e^{-5t}$.

7. **Taylor Method of Order Two I**
   Use the Taylor method of order two with $h = 0.1$ to approximate the solution to

$$y' = 1 + t\sin(ty), \quad 0 \le t \le 2, \quad y(0) = 0.$$

8. **Taylor Method of Order Two II**
Given the initial-value problem

$$y' = \frac{2}{t}y + t^2 e^t, \quad 1 \leq t \leq 2, \quad y(1) = 0,$$

with exact solution $y(t) = t^2(e^t - e)$:

   a. Use Taylor's method of order two with $h = 0.1$ to approximate the solution, and compare it with the actual values of y.

   b. Use the answers generated in part (a) and linear interpolation to approximate y at the following values, and compare them to the actual values of y.

      i. $y(1.04)$ **ii.** $y(1.55)$ **iii.** $y(1.97)$

   c. Use Taylor's method of order four with $h = 0.1$ to approximate the solution, and compare it with the actual values of y.

   d. Use the answers generated in part (c) and piecewise cubic Hermite interpolation to approximate y at the following values, and compare them to the actual values of y.

      i. $y(1.04)$ **ii.** $y(1.55)$ **iii.** $y(1.97)$

9. **Runge-Kutta Method of Order Four**
Use the Runge-Kutta method of order four to approximate the solutions to the following initial-value problems. Compare with the actual solutions:

   a. $y' = 2t - 3y + 1, \quad 1 \leq t \leq 2, \quad y(1) = 5$, with $h = 0.2$
   Actual solution: $y(t) = \frac{1}{3} + \frac{2t}{3} + \frac{44}{9}e^{-3(t-1)}$

   b. $y' = te^{-2t} - 2y, \quad 0 \leq t \leq 1, \quad y(0) = 0$, with $h = 0.1$
   Actual solution: $y(t) = \frac{1}{4}te^{-2t} - \frac{1}{8}e^{-2t} + \frac{1}{8}$

   c. $y' = 1 + t\sin(ty), \quad 0 \leq t \leq 2, \quad y(0) = 0$, with $h = 0.2$

10. **Adams-Bashforth Methods I**
Use each of the Adams-Bashforth methods to approximate the solutions to the following initial-value problems. In each case use starting values obtained from the Runge-Kutta method of order four. Compare the results to the actual values.

   a. $y' = y/t - (y/t)^2, \quad 1 \leq t \leq 2, \quad y(1) = 1$, with $h = 0.1$; actual solution $y(t) = \frac{t}{1+\ln t}$.

   b. $y' = 1 + y/t + (y/t)^2, \quad 1 \leq t \leq 3, \quad y(1) = 0$, with $h = 0.2$; actual solution $y(t) = t\tan(\ln t)$.

   c. $y' = -(y + 1)(y + 3), \quad 0 \leq t \leq 2, \quad y(0) = -2$, with $h = 0.1$; actual solution $y(t) = -3 + 2(1 + e^{-2t})^{-1}$.

   d. $y' = -5y + 5t^2 + 2t, \quad 0 \leq t \leq 1, \quad y(0) = \frac{1}{3}$, with $h = 0.1$; actual solution $y(t) = t^2 + \frac{1}{3}e^{-5t}$.

11. **Adams-Bashforth Methods II**
Use all the Adams-Bashforth methods to approximate the solutions to the following initial-value problems. In each case use exact starting values, and compare the results to the actual values.

   a. $y' = te^{3t} - 2y, \quad 0 \leq t \leq 1, \quad y(0) = 0$, with $h = 0.2$; actual solution $y(t) = \frac{1}{5}te^{3t} - \frac{1}{25}e^{3t} + \frac{1}{25}e^{-2t}$.

   b. $y' = 1 + (t - y)^2, \quad 2 \leq t \leq 3, \quad y(2) = 1$, with $h = 0.2$; actual solution $y(t) = t + \frac{1}{1-t}$.

   c. $y' = 1 + y/t, \quad 1 \leq t \leq 2, \quad y(1) = 2$, with $h = 0.2$; actual solution $y(t) = t\ln t + 2t$.

   d. $y' = \cos 2t + \sin 3t, \quad 0 \leq t \leq 1, \quad y(0) = 1$, with $h = 0.2$; actual solution $y(t) = \frac{1}{2}\sin 2t - \frac{1}{3}\cos 3t + \frac{4}{3}$.

12. **Adams-Bashforth Three-Step Method**
Use the Adams-Bashforth three-step method to approximate the solutions to the following initial-value problems. Use starting values obtained from the Runge-Kutta method of order four:

   a. $y' = t - y^2, \quad 0 \leq t \leq 2, \quad y(0) = 1$, with $h = 0.2$

   b. $y' = (y/t)^2 + y/t, \quad 1 \leq t \leq 4, \quad y(1) = 1$, with $h = 0.2$

13. **Adams Fourth-Order Predictor-Corrector Algorithm**
Use the Adams Fourth-Order Predictor-Corrector Algorithm with $h = 0.2$ to approximate the solution to:

$$y' = 1 - y + \tan^{-1}(t), \quad 0 \leq t \leq 3, \quad y(0) = 0$$

Use the Runge-Kutta method of order four to compute starting values.

14. **Picard's Method**
Picard's method for solving the initial-value problem

$$y' = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

is described as follows: Let $y_0(t) = \alpha$ for each t in $[a, b]$. Define a sequence $\{y_k(t)\}$ of functions by

$$y_k(t) = \alpha + \int_a^t f(\tau, y_{k-1}(\tau))\, d\tau, \quad k = 1, 2, \ldots.$$

   a. Integrate $y' = f(t, y(t))$, and use the initial condition to derive Picard's method.
   b. Generate $y_0(t)$, $y_1(t)$, $y_2(t)$, and $y_3(t)$ for the initial-value problem

$$y' = -y + t + 1, \quad 0 \leq t \leq 1, \quad y(0) = 1.$$

   c. Compare the result in part (b) to the Maclaurin series of the actual solution $y(t) = t + e^{-t}$.

15. **Projectile Motion with Air Resistance**
A projectile of mass $m = 0.11$ kg shot vertically upward with initial velocity $v(0) = 8$ m/s is slowed due to the force of gravity, $F_g = -mg$, and due to air resistance, $F_r = -kv|v|$, where $g = 9.8$ m/s^2 and $k = 0.002$ kg/m. The differential equation for the velocity v is given by

$$mv' = -mg - kv|v|.$$

   a. Find the velocity after $0.1, 0.2, \ldots, 1.0$ s.
   b. To the nearest tenth of a second, determine when the projectile reaches its maximum height and begins falling.

16. **Systems of ODEs: Second-Order to First-Order Conversion**
The differential equation

$$y'' - y = 0, \quad y(0) = 1, \quad y'(0) = 0$$

has the exact solution $y(t) = \cosh(t)$.

   a. Convert this second-order equation into a system of first-order equations.
   b. Use the Runge-Kutta method of order four with $h = 0.1$ to approximate $y(1)$ and $y'(1)$.
   c. Compare your results with the exact values $\cosh(1) \approx 1.5430806$ and $\sinh(1) \approx 1.1752012$.

17. **Numerical Differentiation from ODE Solutions**
Consider the initial-value problem:

$$y' = f(t, y), \quad 0 \leq t \leq 1, \quad y(0) = 1$$

Suppose you have computed an approximate solution using the RK4 method with $h = 0.1$, giving values $y_0, y_1, \ldots, y_{10}$.

   a. Use numerical differentiation to estimate $y'(0.5)$ from your computed values.
   b. Compare this with $f(0.5, y_5)$ from the differential equation itself.
   c. What does any discrepancy tell you about the accuracy of your solution?

18. **Arc Length via Numerical Methods**
The arc length of a curve $y = f(x)$ from $x = a$ to $x = b$ is given by:

$$L = \int_a^b \sqrt{1 + [f'(x)]^2}\, dx$$

For $f(x) = \sin x$ on $[0, \pi]$:

   a. Use numerical differentiation to compute $f'(x)$ at several points.
   b. Use numerical integration (Simpson's Rule with $n = 8$) to approximate the arc length.
   c. The exact arc length is approximately 3.8202. Compute your error.