

Truncation Errors and the Taylor Series

Chapter 4

How does a CPU compute the following functions for a specific x value?

$\cos(x)$ $\sin(x)$ e^x $\log(x)$ etc.

- Non-elementary functions such as, *trigonometric, exponential* and others are expressed in an approximate fashion using **Taylor series** when their values, derivatives, and integrals are computed.
- **Taylor series** provides a means to predict the value of a function at one point in terms of the function value and its derivatives at another point.

Taylor Series (n^{th} order approximation):

$$f(x_{i+1}) \cong f(x_i) + \frac{f'(x_i)}{1!}(x_{i+1} - x_i) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \dots + \frac{f^{(n)}(x_i)}{n!}(x_{i+1} - x_i)^n + R_n$$

The Reminder term, R_n , accounts for all terms from $(n+1)$ to infinity.

$$R_n = \frac{f^{(n+1)}(\varepsilon)}{(n+1)!} h^{(n+1)}$$

Define the *step size* as $h=(x_{i+1}- x_i)$, the *series* becomes:

$$f(x_{i+1}) \cong f(x_i) + \frac{f'(x_i)}{1!} h + \frac{f''(x_i)}{2!} h^2 + \dots + \frac{f^{(n)}(x_i)}{n!} h^n + R_n$$

Any smooth function can be approximated as a polynomial.

Take $x = x_{i+1}$ Then $f(x) \approx f(x_i)$ **zero order** approximation

$$f(x) \cong f(x_i) + f'(x_i)(x - x_i) \quad \text{first order approximation}$$

Second order approximation:

$$f(x) \cong f(x_i) + \frac{f'(x_i)}{1!}(x - x_i) + \frac{f''(x_i)}{2!}(x - x_i)^2$$

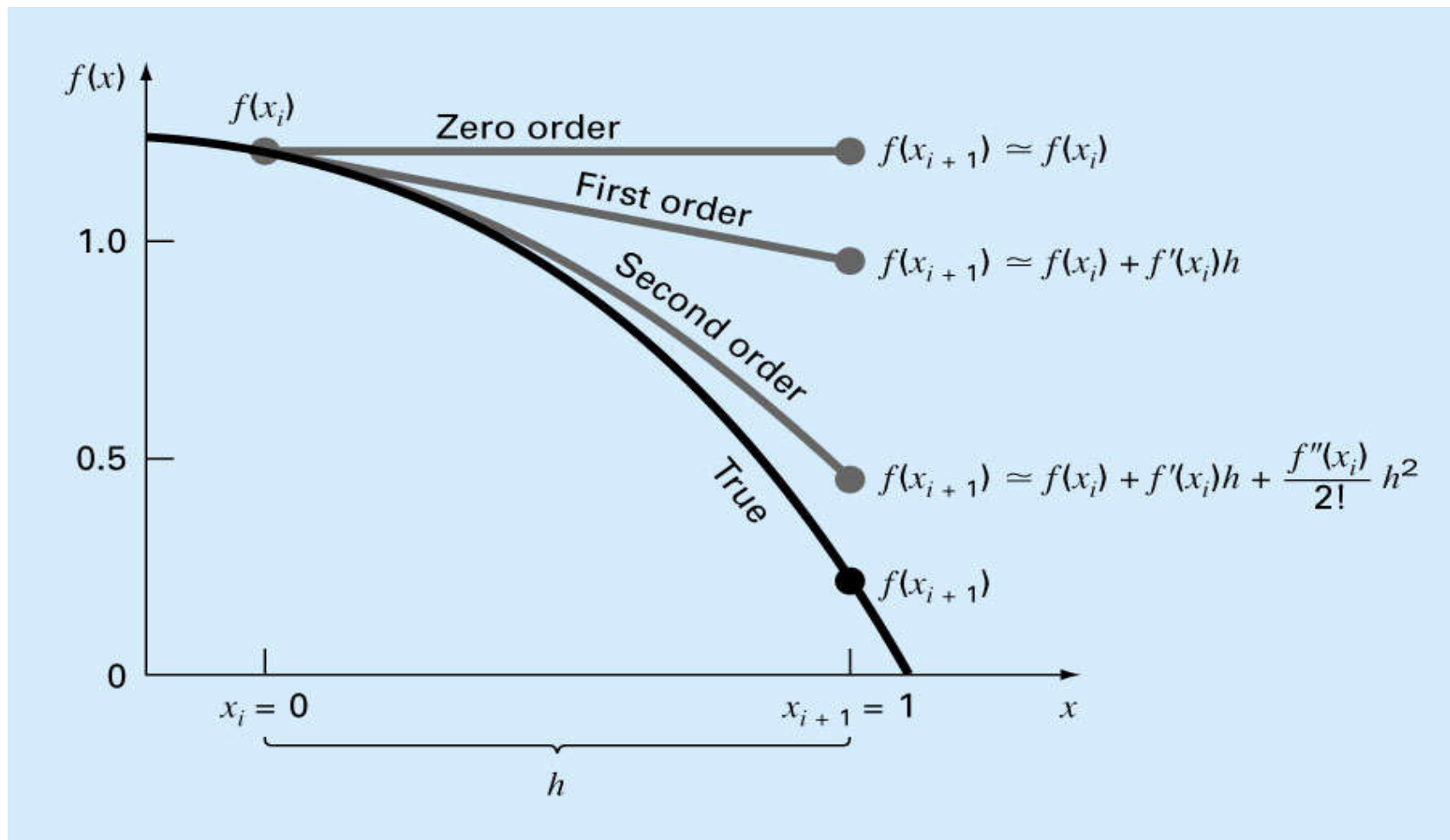
n^{th} order approximation:

$$f(x) \cong f(x_i) + \frac{f'(x_i)}{1!}(x - x_i) + \frac{f''(x_i)}{2!}(x - x_i)^2 + \dots + \frac{f^{(n)}(x_i)}{n!}(x - x_i)^n + R_n$$

- Each additional term will contribute some improvement to the approximation. Only if an infinite number of terms are added will the series yield an exact result.
- In most cases, only a few terms will result in an approximation that is close enough to the true value for practical purposes

Example

Approximate the function $f(x) = 1.2 - 0.25x - 0.5x^2 - 0.15x^3 - 0.1x^4$ from $x_i = 0$ with $h = 1$ and **predict** $f(x)$ at $x_{i+1} = 1$.



Example:

computing $f(x) = e^x$ using Taylor Series expansion

$$f(x_{i+1}) \cong f(x_i) + \frac{f'(x_i)}{1!}(x_{i+1} - x_i) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \dots + \frac{f^{(n)}(x_i)}{n!}(x_{i+1} - x_i)^n + R_n$$

Choose $x = x_{i+1}$ and $x_i = 0$ Then $f(x_{i+1}) = f(x)$ and $(x_{i+1} - x_i) = x$

Since First Derivative of e^x is also e^x :

$$(2.) (e^x)'' = e^x \quad (3.) (e^x)''' = e^x, \quad \dots \quad (n^{\text{th}}.) (e^x)^{(n)} = e^x$$

As a result we get:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Looks familiar?

Maclaurin series for e^x

Yet another example:

computing $f(x) = \cos(x)$ using Taylor Series expansion

Choose $x = x_{i+1}$ and $x_i = 0$ Then $f(x_{i+1}) = f(x)$ and $(x_{i+1} - x_i) = x$

Derivatives of $\cos(x)$:

$$(1.) (\cos(x))' = -\sin(x) \qquad (2.) (\cos(x))'' = -\cos(x),$$

$$(3.) (\cos(x))''' = \sin(x) \qquad (4.) (\cos(x))'''' = \cos(x),$$

.....

As a result we get:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

Notes on Taylor expansion:

- Each additional term will contribute some improvement to the approximation. Only if an infinite number of terms are added will the series yield an exact result.
- In most cases, several terms will result in an approximation that is close enough to the true value for practical purposes
- Reminder value **R** represents the **truncation error**
- The order of truncation error is $h^{n+1} \rightarrow R=O(h^{n+1})$,
If $R=O(h)$, halving the step size will halve the error.
If $R=O(h^2)$, halving the step size will quarter the error.