Round-off and truncation errors

Lecture 5

Textbook:

Chapra & Canale, *Numerical Methods for Engineers*, Chapters 2-4

Numerical errors

Round-off

Truncation

Ranges of variables in Python (and other languages)

- **Integer** (limited by memory)
- **Float** (largest $\sim 10^{308}$, smallest $\sim 10^{-324}$)
- **Complex** (same as **float**)

NumPy variables

Data type	Description
bool_	Boolean (True or False) stored as a byte
int_	Default integer type (same as C `long`; normally either `int64` or `int32`)
intc	Identical to C `int` (normally `int32` or `int64`)
intp	Integer used for indexing (same as C `ssize_t`; normally either `int32` or `int64`)
int8	Byte (-128 to 127)
int16	Integer (-32768 to 32767)
int32	Integer (-2147483648 to 2147483647)
int64	Integer (-9223372036854775808 to 9223372036854775807)
uint8	Unsigned integer (0 to 255)
uint16	Unsigned integer (0 to 65535)
uint32	Unsigned integer (0 to 4294967295)
uint64	Unsigned integer (0 to 18446744073709551615)
float_	Shorthand for `float64`.
float16	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
float32	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
float64	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
complex_	Shorthand for `complex128`.
complex64	Complex number, represented by two 32-bit floats (real and imaginary components)
complex128	Complex number, represented by two 64-bit floats (real and imaginary components)

source: scipy.org

Round-off errors are due to approximate representation of floating-point numbers.

```
from math import pi
print(pi)
```

```
3.141592653589793
```

$$\pi = 3.14159265358979323846264338327950288419716939937 \ldots$$

Subtractive cancellation

$$x = 1$$
$$y = 1 + 10^{-15}\sqrt{2}$$

```
x=1.0
y=1.0+1e-15*sqrt(2)
dt=1e-15*sqrt(2)
dn=y-x
```

```
dt = 1.4142135623730953e-15
dn = 1.3322676295501878e-15
Relative error: 0.06150861208762892
```

Similar accumulation of round-off in addition (large number + small number).

Truncation Errors

Truncation errors are created by truncating the math.

Example: MacLaurin Series for exponential function

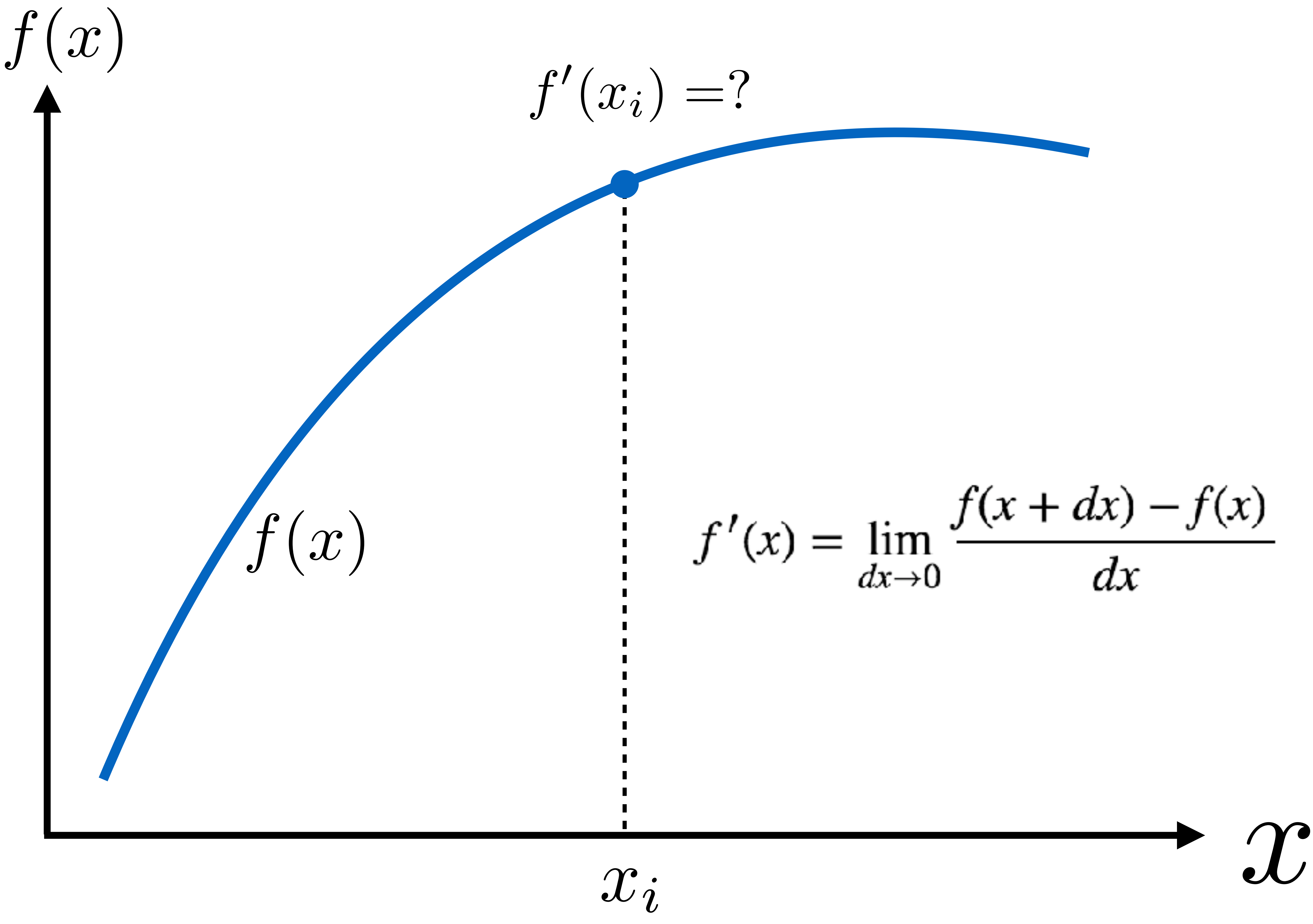
$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Example: MacLaurin Series for exponential function

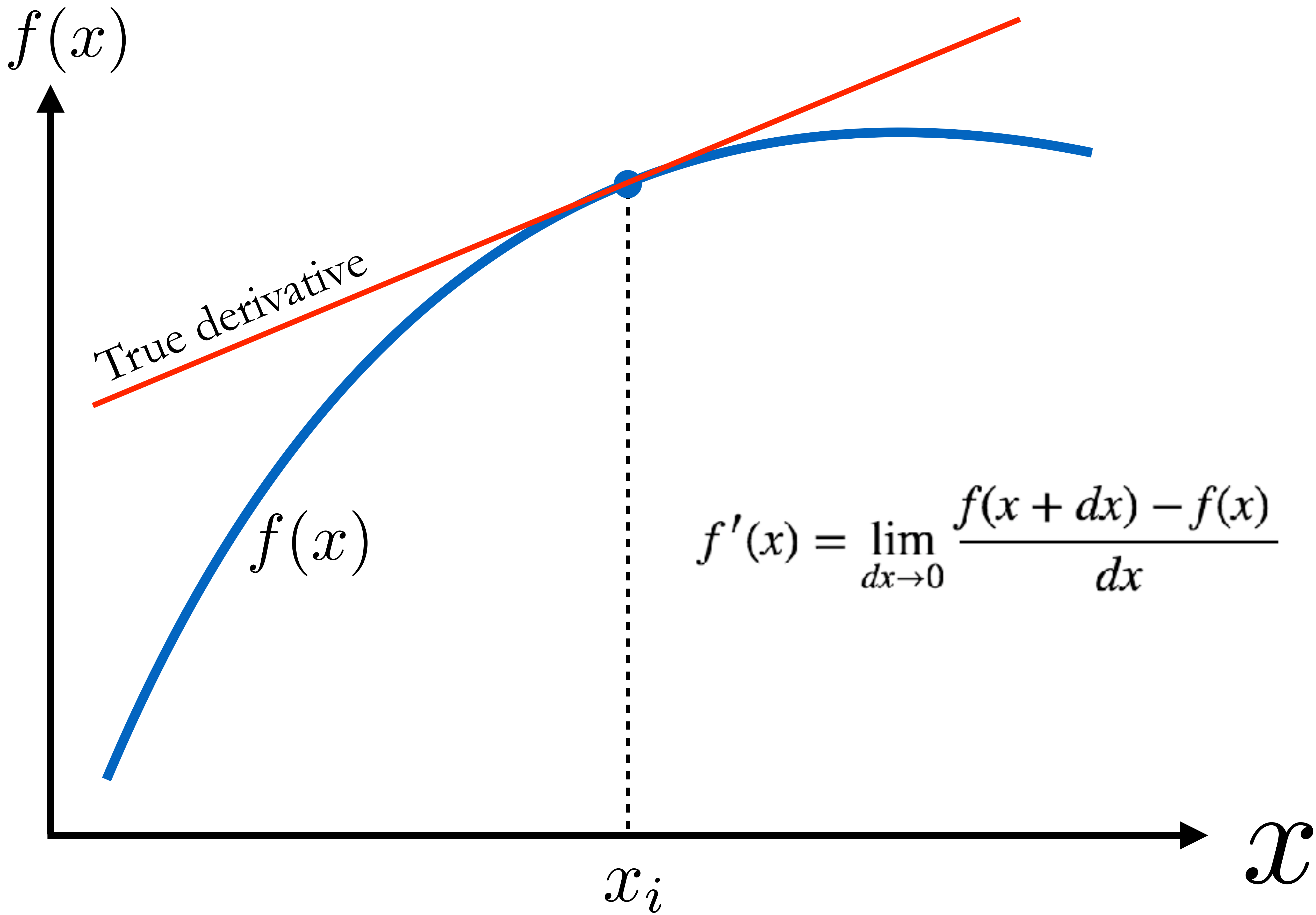
$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \simeq 1 + x + \frac{x^2}{2!}$$

Numerical Differentiation and numerical errors

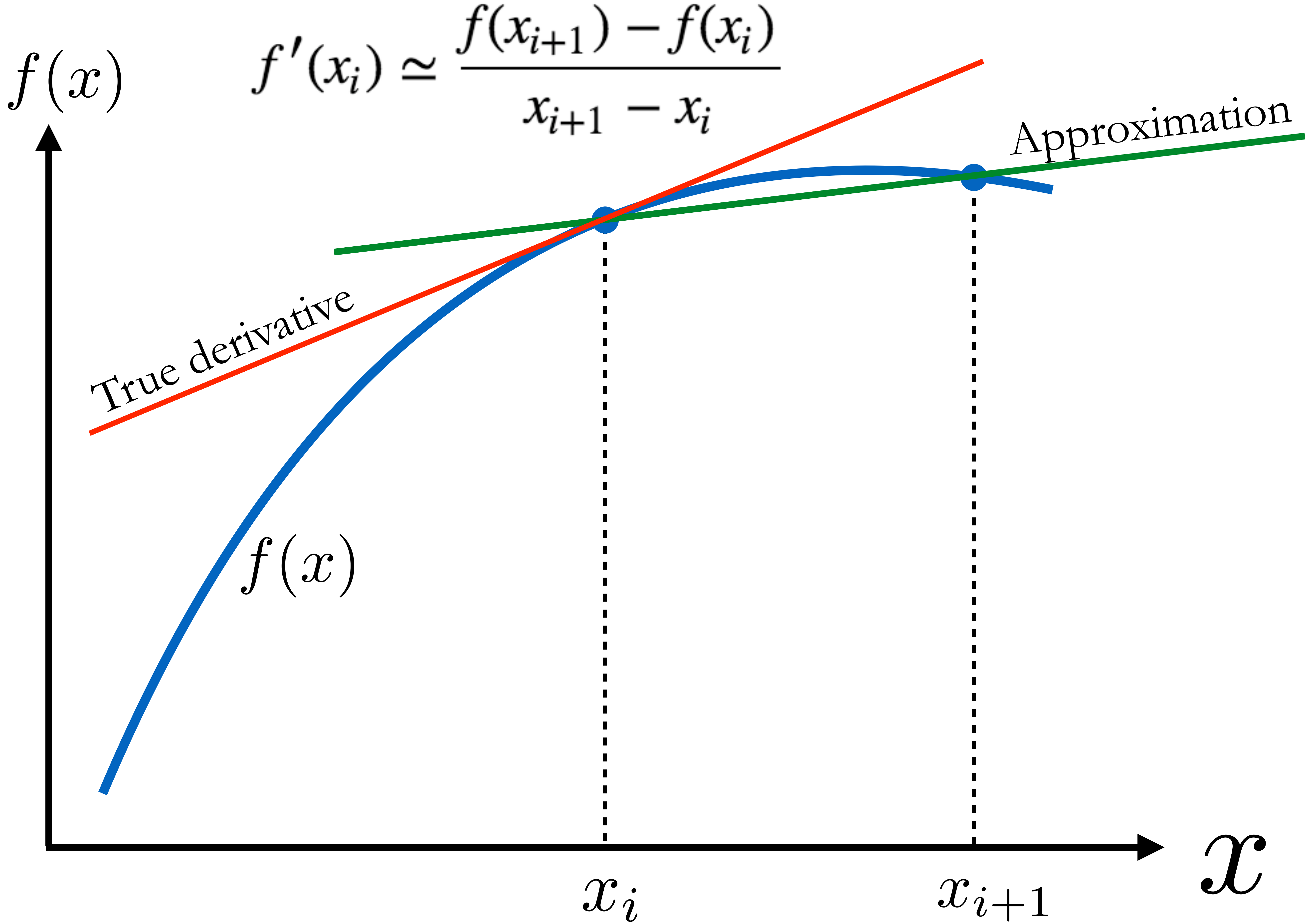
(Chapter 4.3)



$f(x)$
$f'(x_i) = ?$
$f(x)$
$f'(x) = \lim_{dx \to 0} \frac{f(x + dx) - f(x)}{dx}$
x_i
x



$f(x)$
True derivative
$f(x)$
$f'(x) = \lim_{dx \to 0} \frac{f(x+dx) - f(x)}{dx}$
x_i
x



$f'(x_i) \simeq \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$
$f(x)$
Approximation
True derivative
$f(x)$
x_i
x_{i+1}
x

Truncation error estimate using Taylor series

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

$$h = x_{i+1} - x_i$$

$$R_n = \int_{x_i}^{x_{i+1}} \frac{(x_{i+1} - x)^n}{n!} f^{(n+1)}(x)dx = \frac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1}$$

Truncation error estimate using Taylor series

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + R_1$$

Truncation error estimate using Taylor series

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + R_1$$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} - \frac{R_1}{x_{i+1} - x_i}$$

approximation

Truncation error

Truncation error estimate using Taylor series

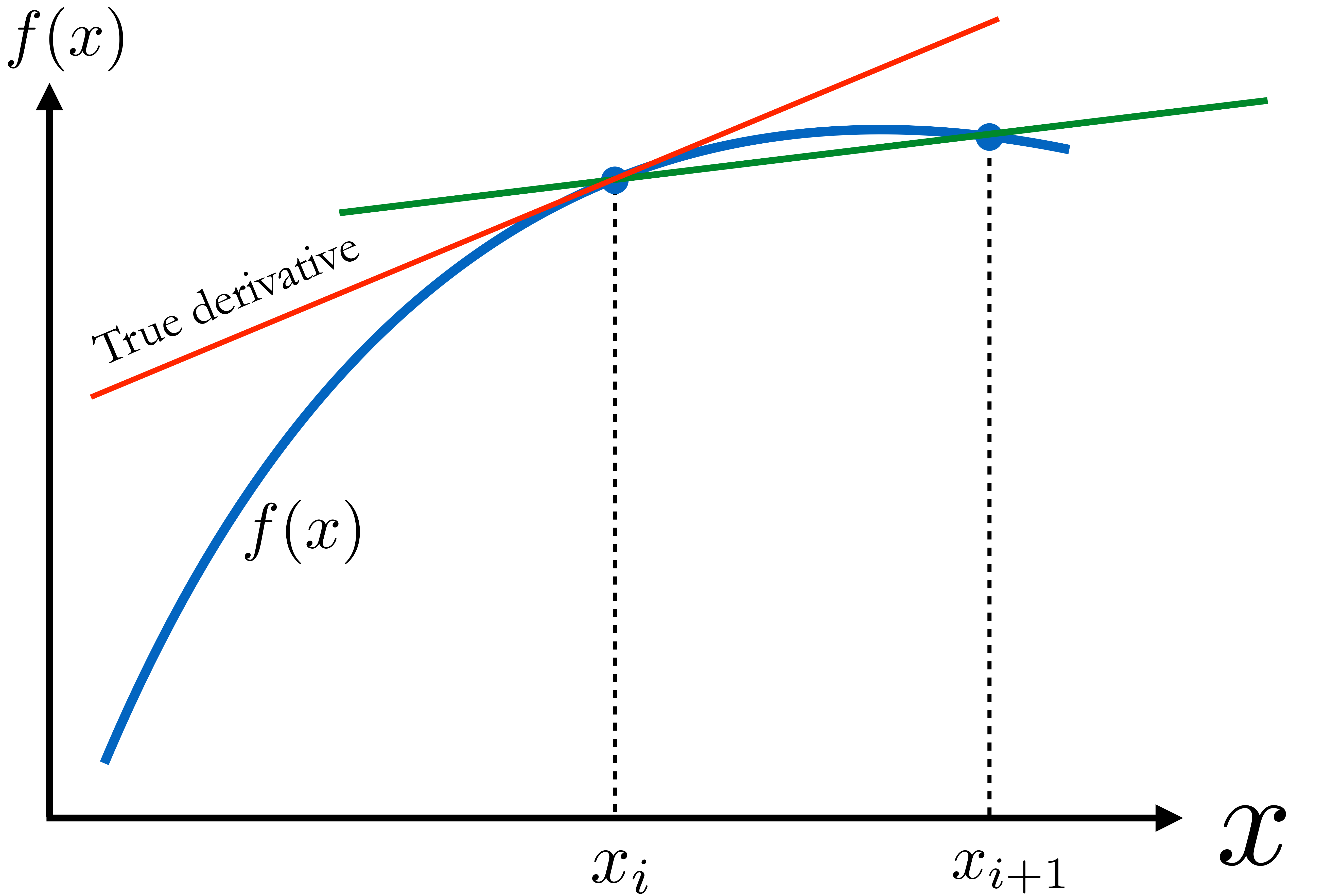
$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!} h^{n+1} = \mathcal{O}(h^{n+1})$$

Truncation error estimate using Taylor series

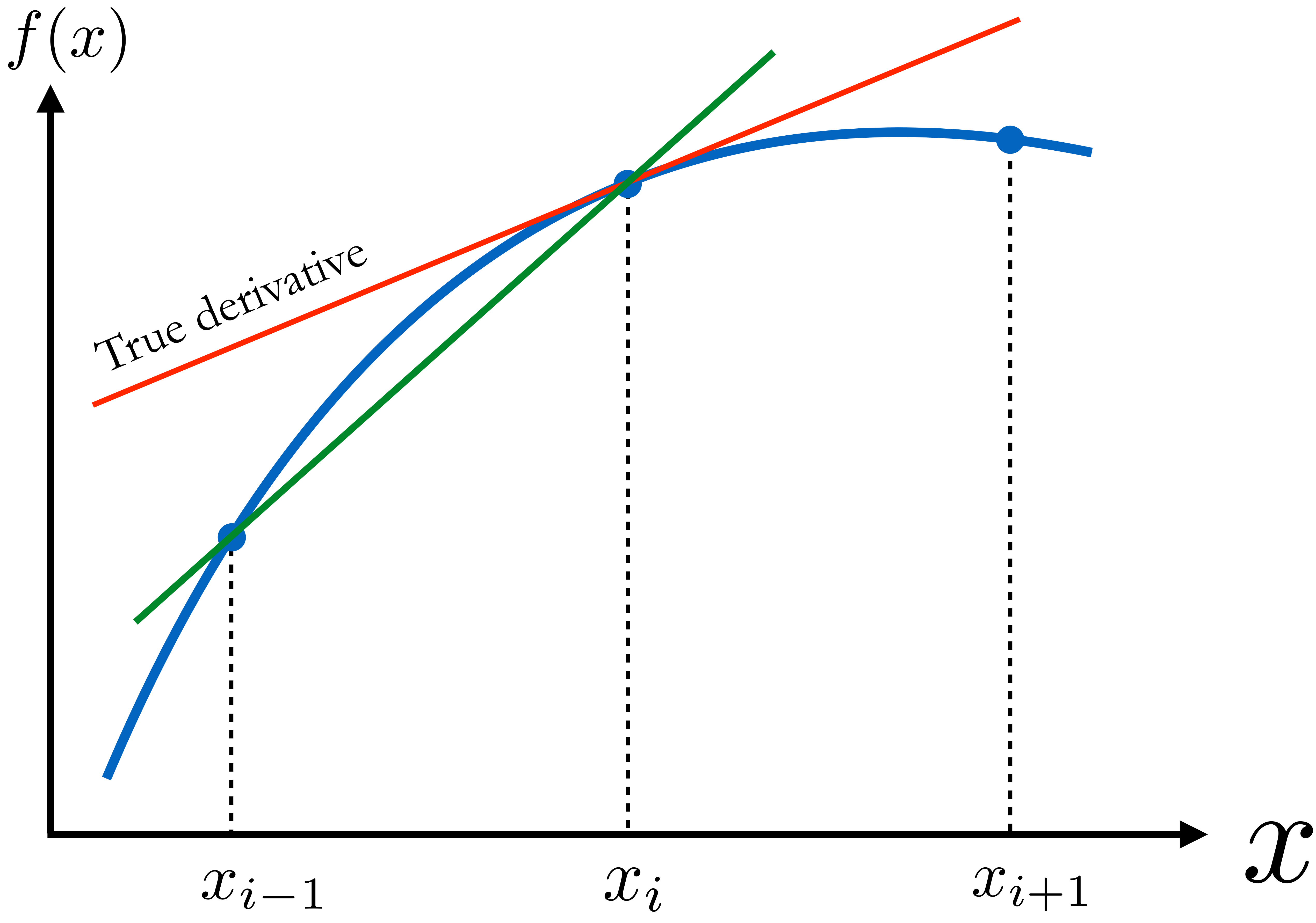
$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} + \mathcal{O}(h)$$

Error $\propto h$

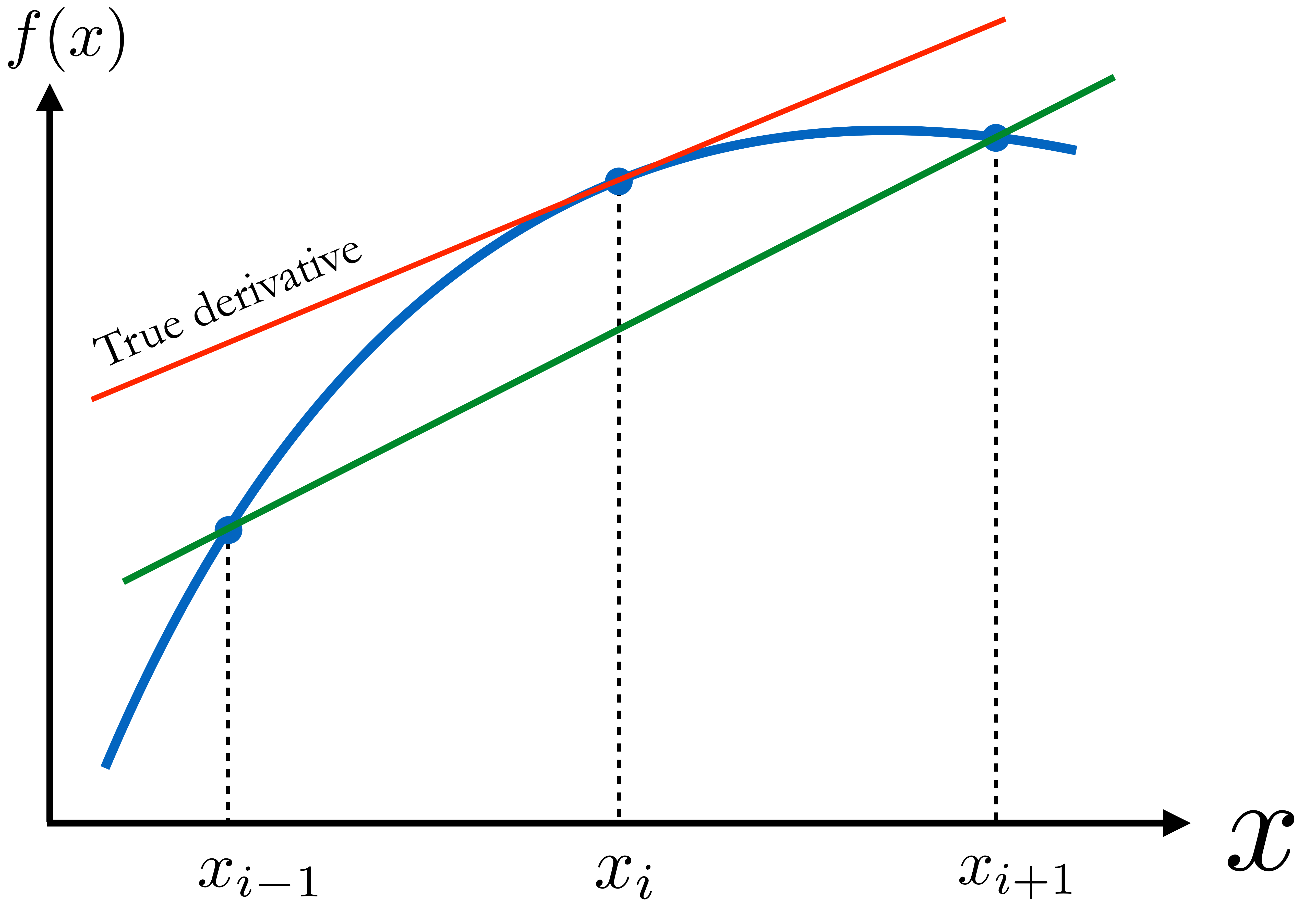
Halving stepsize, halves the truncation error.



$f(x)$
True derivative
$f(x)$
x_i
x_{i+1}
x



$f(x)$
True derivative
x
x_{i-1}
x_i
x_{i+1}


$f(x)$
True derivative
x_{i-1}
x_i
x_{i+1}
x

- Forward formula:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} + \mathcal{O}(h)$$

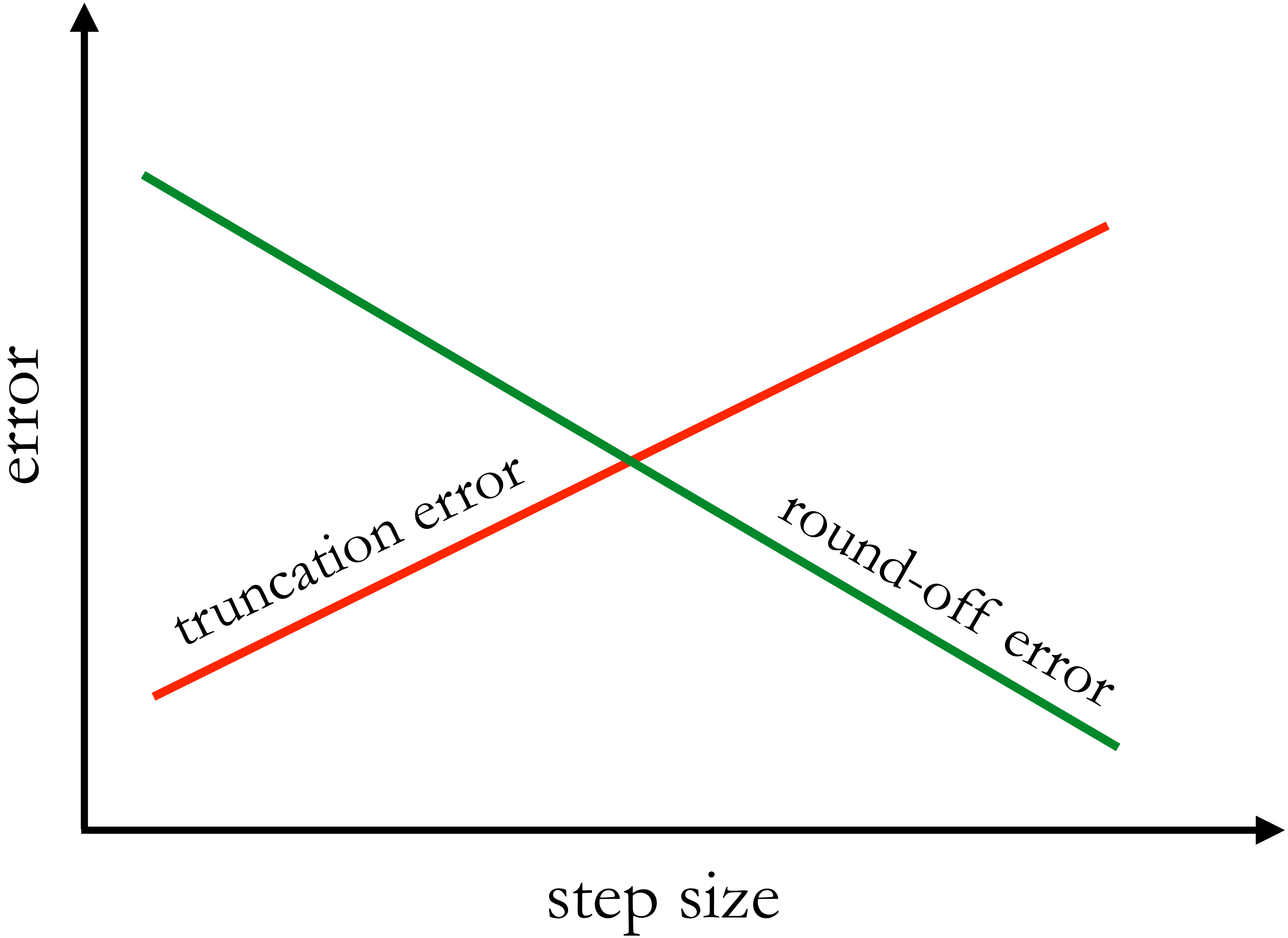
- Backward formula:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} + \mathcal{O}(h)$$

- Centered formula:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} + \mathcal{O}(h^2)$$

See Chapter 4.1 for derivation.


error
truncation error
round-off error
step size

Centered approximation:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} - \frac{f^{(3)}(\xi)}{6}h^2$$

True value

Finite-difference approximation

Truncation error

$$f(x_{i-1}) = \tilde{f}(x_{i-1}) + e_{i-1}$$

True value

Rounded value

Round-off error

$$f(x_{i+1}) = \tilde{f}(x_{i+1}) + e_{i+1}$$

$$f'(x_i) = \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} + \frac{e_{i+1} - e_{i-1}}{2h} - \frac{f^{(3)}(\xi)}{6}h^2$$

True value | Finite-difference approximation | Round-off error | Truncation error

Assume $e_{i\pm1} \leq \epsilon$ (thus $e_{i+1} - e_{i-1} \leq 2\epsilon$) and $f^{(3)}(\xi) \leq M$.

Total error

$$\left| f'(x_i) - \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} \right| \leq \frac{\epsilon}{h} + \frac{h^2 M}{6}$$

$$f'(x_i) = \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} + \frac{e_{i+1} - e_{i-1}}{2h} - \frac{f^{(3)}(\xi)}{6}h^2$$

True value | Finite-difference approximation | Round-off error | Truncation error

Assume:

- $e_{i\pm1} \leq \epsilon$. Thus, $e_{i+1} - e_{i-1} \leq 2\epsilon$.
- $f^{(3)}(\xi) \leq M$.

Total error

$$\left| f'(x_i) - \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} \right| \leq \frac{\epsilon}{h} + \frac{h^2 M}{6}$$

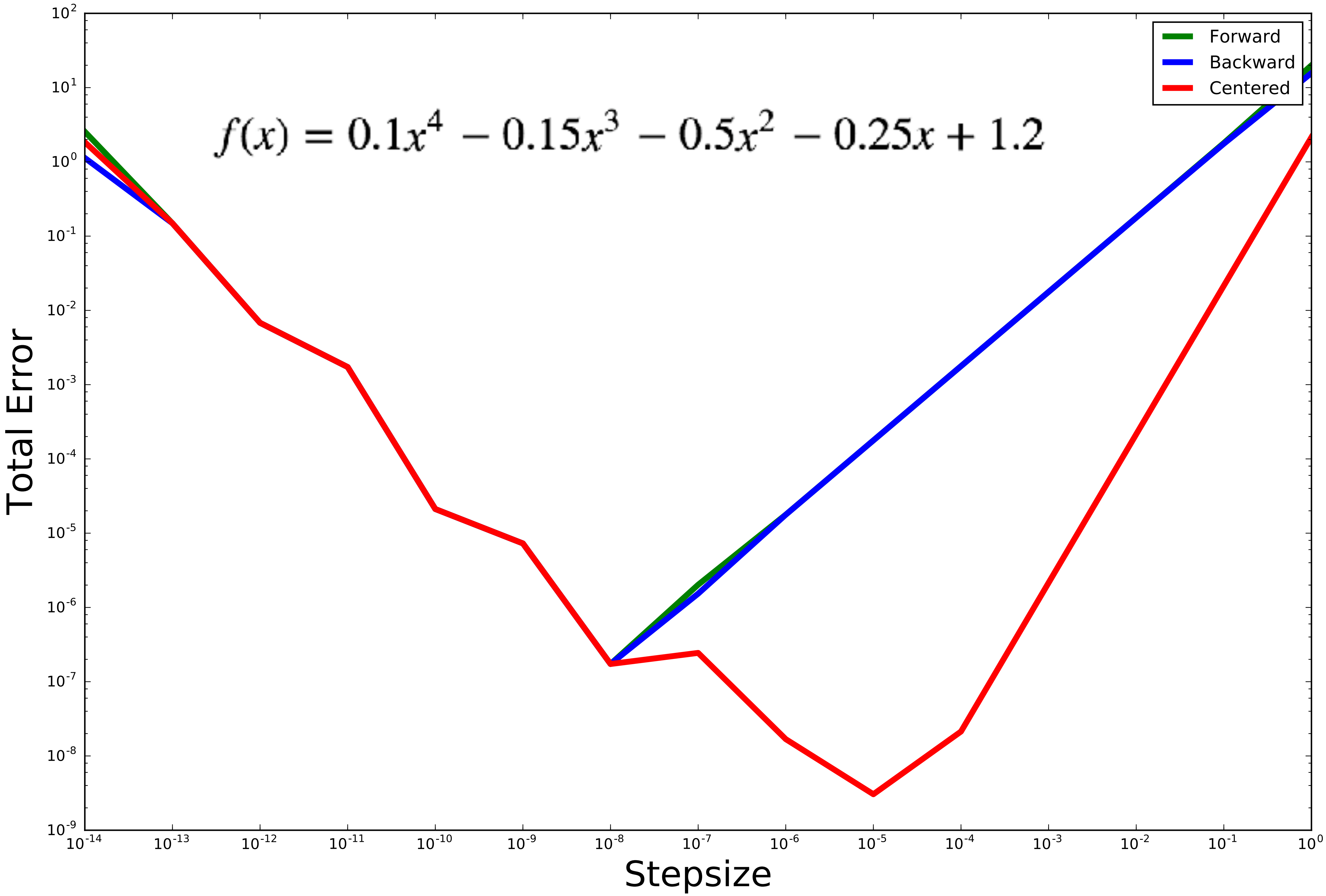
Total error

$$\left| f'(x_i) - \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} \right| \leq \frac{\epsilon}{h} + \frac{h^2 M}{6}$$

Optimal step size

$$h_{\text{opt}} = \left(\frac{3\epsilon}{M} \right)^{1/3}$$

Optimal step size



Error Propagation and Condition Number

(Chapter 4.2)

$$f(x) = f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x})$$

$$\frac{f(x) - f(\tilde{x})}{f(\tilde{x})} = \frac{f'(\tilde{x})(x - \tilde{x})}{f(\tilde{x})}$$

$$\frac{f(x) - f(\tilde{x})}{f(\tilde{x})} = \frac{\tilde{x} f'(\tilde{x})}{f(\tilde{x})} \frac{x - \tilde{x}}{\tilde{x}}$$

Condition
number

$$\frac{f(x) - f(\tilde{x})}{f(\tilde{x})} = \text{Condition number} \times \frac{(x - \tilde{x})}{\tilde{x}}$$

Small condition number = error decreases.

Large condition number = ill-conditioned.

Supplementary Material

```python
x=1.1+2.2
if (x==3.3):
    print("Condition x=3.3 is met")
```

Bad code!

```python
x=1.1+2.2
eps=1e-13
if (abs(x-3.3)<eps):
    print("Condition x=3.3 is met")
```

Good code

```
Condition x=3.3 is met
```

If $\tilde{x}$ is an approximation of x, its effect on $f(x)$ is:

$$\Delta f(\tilde{x}) = |f(x) - f(\tilde{x})|$$

Using Taylor series

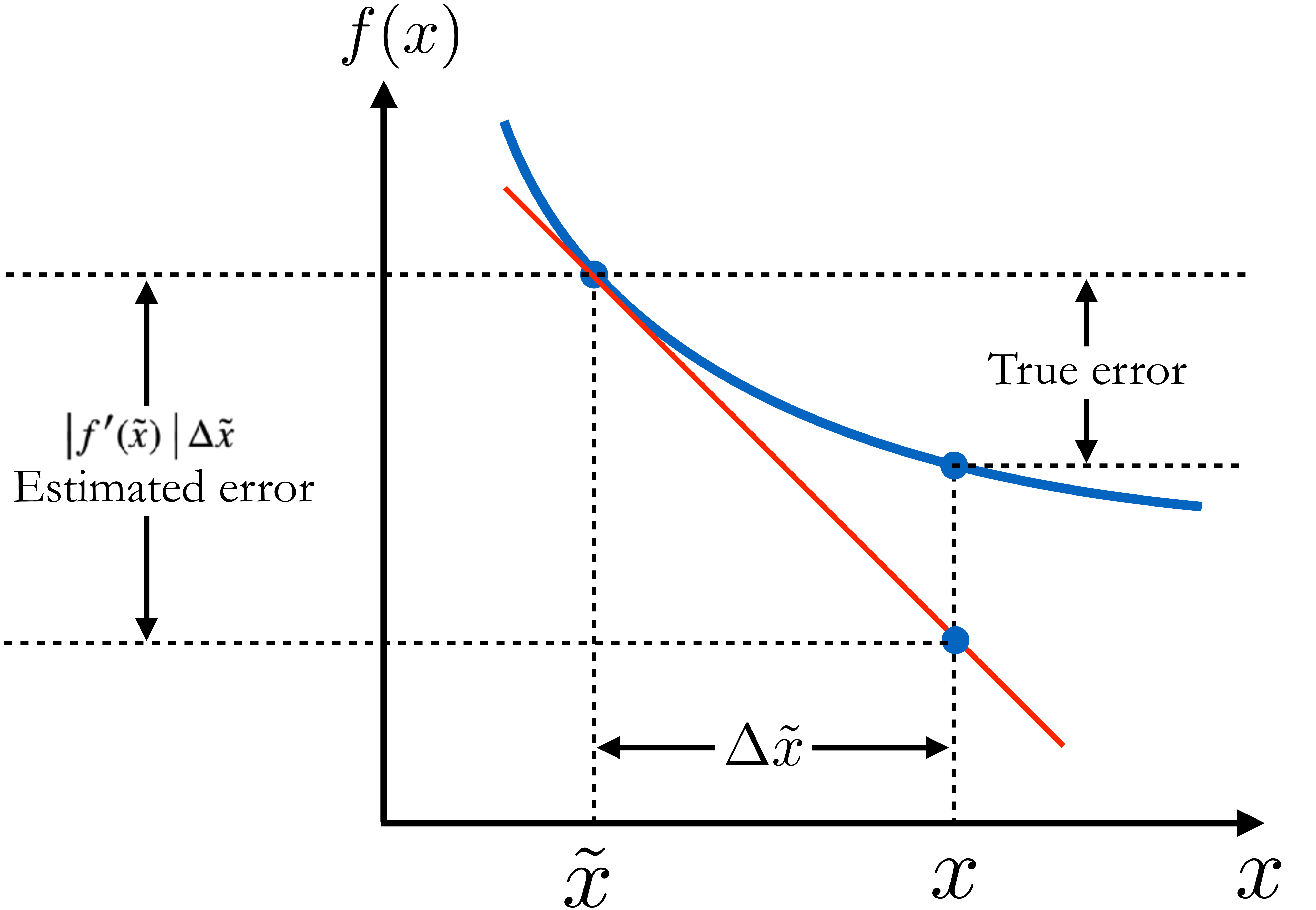
$$f(x) = f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x}) + \frac{f''(\tilde{x})}{2}(x - \tilde{x})^2 + \ldots$$

we get

$$f(x) - f(\tilde{x}) \simeq f'(\tilde{x})(x - \tilde{x})$$

or

$$\Delta f(\tilde{x}) = \left| f'(\tilde{x}) \right| \Delta \tilde{x}$$


$f(x)$
$|f'(\tilde{x})|\Delta\tilde{x}$
Estimated error
True error
$\Delta\tilde{x}$
$\tilde{x}$
x
x