# CS 357 - 03 Errors and big-O notation

Boyang Li (boyangl3)

**Absolute and relative error:**

- Approximate result = True value + error
- **Absolute error:** $e_a = |x - \hat{x}|$
- **Relative error:** $e_r = \dfrac{|x - \hat{x}|}{|x|}$

While $x$ is the true value and $\hat{x}$ is the approximated value.

**Significant digits:** For "true" value ($x$) the number of significant digits begins from the leftmost non-zero digit and ends with the rightmost digit. For example:

- The number 3.14159 has six significant digits.
- The number 0.000350 has three significant digits.

For an approximate result ($\hat{x}$), the number of significant digits can be determined in the following process:

1. Evaluate absolute error $e_a = |x - \hat{x}|$
2. Count first $n$ digits from the first decimal places to the digits following by a number from 0 to 4. Here are 2 examples.
   - $\hat{x} = 3.14159 \to |x - \hat{x}| = 0.000002653 \to 6$ sig-fig
   - $\hat{x} = 3.1415 \to |x - \hat{x}| = 0.000092653 \to 4$ sig-fig

The number of accurate significant digits can be estimated by the relative error:

$$\frac{|x - \hat{x}|}{|x|} \geq 10^{-n+1}$$

then $\hat{x}$ has at most $n$ significant digits.

We can use the rule-of-thumb to calculate the upper bound of relative error, if we have $n$ significant digits, then the relative error

$$\frac{|x - \hat{x}|}{|x|} \leq 10^{-n+1}$$

**Error for vectors:** We are calculating absolute and relative errors for vectors in a similar way, by taking the difference and calculate the norm.

- **Absolute error:** $e_a = \|\mathbf{x} - \hat{\mathbf{x}}\|$
- **Relative error:** $e_r = \dfrac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{\|\mathbf{x}\|}$

**Rounding and truncation error:**

- **Rounding error:** Occurs from rounding values in a computation. For example, store the fraction $1/3 \approx 0.3333$ into the computer.
- **Truncation error:** Occurs when using an approximate algorithm in place of an exact mathematical procedure of function. For example, using finite Taylor Series to approximate a function.

**Big-O Notation:** Used to describe asymptotic behavior. The definition in the cases of approaching 0 or $\infty$ are as follows:

- **"$\infty$" Condition:** $f(x) = \mathcal{O}(g(x))$ as $x \to \infty$ if and only if there exists a value $M$ and some $x_0$ such that $|f(x)| \leq M|g(x)|$ for $\forall x$ where $x \geq x_0$.
- **"0" Condition:** $f(h) = \mathcal{O}(g(h))$ as $h \to 0$ if and only if there exists a value $M$ and some $h_0$ such that $|f(h)| \leq M|g(h)|$ for $\forall h$ where $0 \leq h \leq h_0$.
- **Arbitrary "$a$" Condition:** $f(x) = \mathcal{O}(g(x))$ as $x \to a$ if and only if there exists a value $M$ and some $\delta$ such that $|f(x)| \leq M|g(x)|$ for $\forall x$ where $0 \leq |x - a| \leq \delta$.

Here are some examples of big-O notation:

- **Time complexity:** The time complexity for matrix multiplication (each size is $n \times n$) is $\mathcal{O}(n^3)$.
- **Truncation errors:** A numerical method is called $n$-th order accurate if its truncation error $E(h)$ obeys $\mathcal{O}(h^n)$.