Finite Difference Method
Motivation

For a given smooth function $f(x)$, we want to calculate the derivative $f'(x)$ at a given value of $x$.

Suppose we don’t know how to compute the analytical expression for $f'(x)$, or it is computationally very expensive. However you do know how to evaluate the function value:

```python
def f(x):
    # do stuff here
    feval = ...
    return feval
```

We know that:

$$f'(x) = \lim_{h \to 0} \left( \frac{f(x + h) - f(x)}{h} \right)$$

Can we just use $f'(x) \approx \frac{f(x+h) - f(x)}{h}$ as an approximation? How do we choose $h$? Can we get estimate the error of our approximation?
Finite difference method

For a differentiable function $f: \mathbb{R} \rightarrow \mathbb{R}$, the derivative is defined as:

$$f'(x) = \lim_{h \to 0} \left( \frac{f(x + h) - f(x)}{h} \right)$$

Taylor Series centered at $x$, where $\bar{x} = x + h$

$$f(x + h) = f(x) + f'(x) h + f''(x) \frac{h^2}{2} + f'''(x) \frac{h^3}{6} + \cdots$$

$$f(x + h) = f(x) + f'(x) h + O(h^2)$$

$$f'(x) = \frac{f(x + h) - f(x)}{h} + O(h)$$

We define the **Forward Finite Difference** as:

$$df(x) = \frac{f(x + h) - f(x)}{h} \rightarrow f'(x) = df(x) + O(h)$$

Therefore, the **truncation error** of the forward finite difference approximation is bounded by:

$$| f'(x) - df(x) | \leq Mh$$
In a similar way, we can write:

\[ f(x - h) = f(x) - f'(x) h + O(h^2) \rightarrow f'(x) = \frac{f(x) - f(x - h)}{h} + O(h^2) \]

And define the **Backward Finite Difference** as:

\[ df(x) = \frac{f(x) - f(x - h)}{h} \rightarrow f'(x) = df(x) + O(h) \]

And subtracting the two Taylor approximations

\[ f(x + h) = f(x) + f'(x) h + f''(x) \frac{h^2}{2} + f'''(x) \frac{h^3}{6} + \cdots \]
\[ f(x - h) = f(x) - f'(x) h + f''(x) \frac{h^2}{2} - f'''(x) \frac{h^3}{6} + \cdots \]

\[ f(x + h) - f(x - h) = 2f'(x) h + f''''(x) \frac{h^3}{6} + O(h^5) \]

\[ f'(x) = \frac{f(x + h) - f(x - h)}{2h} + O(h^2) \]

And define the **Central Finite Difference** as:

\[ df(x) = \frac{f(x + h) - f(x - h)}{2h} \rightarrow f'(x) = df(x) + O(h^2) \]
How accurate is the finite difference approximation? How many function evaluations (in additional to \( f(x) \))? 

**Forward Finite Difference:**

\[
\frac{df(x)}{dx} = \frac{f(x+h) - f(x)}{h} \rightarrow f'(x) = df(x) + O(h)
\]

Truncation error: \( O(h) \)
Cost: 1 function evaluation

**Backward Finite Difference:**

\[
\frac{df(x)}{dx} = \frac{f(x) - f(x-h)}{h} \rightarrow f'(x) = df(x) + O(h)
\]

Truncation error: \( O(h) \)
Cost: 1 function evaluation

**Central Finite Difference:**

\[
\frac{df(x)}{dx} = \frac{f(x+h) - f(x-h)}{2h} \rightarrow f'(x) = df(x) + O(h^2)
\]

Truncation error: \( O(h^2) \)
Cost: 2 function evaluation

Our typical trade-off issue! We can get **better accuracy** with Central Finite Difference with the (possible) **increased computational** cost.

How small should the value of \( h \)?
Example

\[ f(x) = e^x - 2 \]
\[ f'(x) = e^x \]

We want to obtain an approximation for \( f'(1) \)

\[
\text{df}_{\text{approx}} = \frac{(e^{x+h} - 2) - (e^x - 2)}{h}
\]

<table>
<thead>
<tr>
<th>( h )</th>
<th>( \text{error} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000E+00</td>
<td>1.952492E+00</td>
</tr>
<tr>
<td>5.000000E-01</td>
<td>8.085327E-01</td>
</tr>
<tr>
<td>2.500000E-01</td>
<td>3.699627E-01</td>
</tr>
<tr>
<td>1.250000E-01</td>
<td>1.771983E-01</td>
</tr>
<tr>
<td>6.250000E-02</td>
<td>8.674402E-02</td>
</tr>
<tr>
<td>3.125000E-02</td>
<td>4.291906E-02</td>
</tr>
<tr>
<td>1.562500E-02</td>
<td>2.134762E-02</td>
</tr>
<tr>
<td>7.812500E-03</td>
<td>1.064599E-02</td>
</tr>
<tr>
<td>3.906250E-03</td>
<td>5.316064E-03</td>
</tr>
<tr>
<td>1.953125E-03</td>
<td>2.656301E-03</td>
</tr>
<tr>
<td>9.765625E-04</td>
<td>1.327718E-03</td>
</tr>
<tr>
<td>4.882812E-04</td>
<td>6.637511E-04</td>
</tr>
<tr>
<td>2.441406E-04</td>
<td>3.318485E-04</td>
</tr>
<tr>
<td>1.220703E-04</td>
<td>1.659175E-04</td>
</tr>
<tr>
<td>6.103516E-05</td>
<td>8.295707E-05</td>
</tr>
<tr>
<td>3.051758E-05</td>
<td>4.147811E-05</td>
</tr>
<tr>
<td>1.525879E-05</td>
<td>2.073897E-05</td>
</tr>
<tr>
<td>7.629395E-06</td>
<td>1.036945E-05</td>
</tr>
<tr>
<td>3.814697E-06</td>
<td>5.184779E-06</td>
</tr>
<tr>
<td>1.907349E-06</td>
<td>2.592443E-06</td>
</tr>
</tbody>
</table>

Truncation error

\[
\text{error}(h) = \text{abs}(f'(x) - \text{df}_{\text{approx}})
\]
Example

Should we just keep decreasing the perturbation $h$, in order to approach the limit $h \to 0$ and obtain a better approximation for the derivative?
Uh-Oh!

What happened here?

\[ f(x) = e^x - 2, \quad f'(x) = e^x \rightarrow f'(1) \approx 2.7 \]

Forward Finite Difference

\[ df(1) = \frac{f(1 + h) - f(1)}{h} \]

If \( h \) is very “small”, we will have the issue of cancelation!
When computing the finite difference approximation, we have two competing sources of errors: Truncation errors and **Rounding errors**

\[
df(x) = \frac{f(x + h) - f(x)}{h} \leq \frac{\varepsilon_m |f(x)|}{h}
\]
Loss of accuracy due to rounding

Minimize the total error

Truncation error: \( \text{error} \sim Mh \)

Rounding error: \( \text{error} \sim \frac{\epsilon_m |f(x)|}{h} \)

Optimal “h”

Minimize the total error

\[ \text{error} \sim \frac{\epsilon_m |f(x)|}{h} + Mh \]

Gives

\[ h = \sqrt{\frac{\epsilon_m |f(x)|}{M}} \]