Recall: If $f(x)$ is sufficiently often differentiable, then
\[ f(x+h) = f(x) + h f'(x) + \frac{1}{2} h^2 f''(x) + \ldots + \frac{1}{n!} h^n f^{(n)}(x) + \text{remainder}, \]

\[ \text{remainder} = \frac{1}{(n+1)!} h^{n+1} f^{(n+1)}(x) = O(h^{n+1}) \]

For differential equation $(t_{k+1} = t_k + h)$

\[ y(t_{k+1}) = y(t_k) + h y'(t_k) + \frac{1}{2} h^2 y''(t_k) + \ldots \]

If you truncate the Taylor series after a few terms, you get a Taylor series method.

Easiest one: Euler's Method

\[ y_{k+1} = y_k + h y'_k \]

\[ = y_k + h f(t_k, y_k) \]

Local error: Assume there is no error at $t_k$: $y(t_k) = y_k$

True solution
\[ y(t_{k+1}) = y(t_k) + h y'(t_k) + \frac{1}{2} h^2 y''(t_k) + \ldots \]

Numerical
\[ y_{k+1} = y_k + h y'_k \]

Local error
\[ e_k = y(t_{k+1}) - y_{k+1} = \frac{1}{2} h^2 y''(t_k) + \ldots = O(h^2) \]

Definition: A numerical method which depends on a step size $h$ is of order $p$ if error $= O(h^p)$.

Fact: For any reasonable method, and for well-behaved $f$, the global error has one order less than the local error.

$\Rightarrow$ Euler's method is of order 1.
Basic idea of Proof:
For an interval \([a, b]\), it takes \(O\left(\frac{1}{h}\right)\) steps to get from \(a\) to \(b\),
and each step has error \(O(h^{p+1})\) \(\Rightarrow\) sum of local errors is \(O(h^p)\).

My opinion: Euler's method is very simple, but has poor accuracy.
Its main application is as an example to illustrate various concepts.

What about the second order Taylor series method?

\[ y_{k+1} = y_k + h y'_k + \frac{1}{2} h^2 y''_k \]

\[ h \cdot f(t_k, y_k) \]

\[ y''(t_k) = \frac{d}{dt} [y'(t_k)] = \frac{d}{dt} [f(t, y(t))] \]

\[ = f_t \cdot \frac{dt}{dt} + f_y \cdot \frac{dy}{dt} = f_t + f_y \cdot \frac{dy}{dt} \]

\[ = f_t + f_y \cdot y' = f \]

This method, and higher order methods, work well,
but the user has to compute \(y'', y'''\ldots\)
Many times that is not even possible (e.g. if \(f(t, y)\) is a subroutine, not a formula).
Polynomial Interpolation

Given \((t_0, y_0), (t_1, y_1), \ldots, (t_n, y_n)\), find polynomial
\[ p(t) = a_0 + a_1 t + \ldots + a_n t^n \]
of degree \(n\), so that \(p(t_i) = y_i, i = 0 \ldots n\)

**Method 1: Linear Equations**

**Example**
\[
\begin{align*}
(t_0, y_0) &= (-1, 1) \\
(t_1, y_1) &= (1, 1) \\
(t_2, y_2) &= (3, 2) \\
(t_3, y_3) &= (5, 3)
\end{align*}
\]

\[ p(-1) = 1 \]
\[ p(1) = 1 \]
\[ p(3) = 2 \]
\[ p(5) = 3 \]

\[ p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \]

\[ a_0 - a_1 + a_2 - a_3 = 1 \]
\[ a_0 + a_1 + a_2 + a_3 = 1 \]
\[ a_0 + 3a_1 + 9a_2 + 27a_3 = 2 \]
\[ a_0 + 5a_1 + 25a_2 + 125a_3 = 3 \]

**Solution:**
\[ p(t) = \frac{39t + 9t^2 - t^3}{48} \]

**Method 2: Lagrange Polynomials**

\[ L_{n,k}(t) = \prod_{\substack{i=0 \atop i \neq k}}^{n} \frac{t-t_i}{t_k-t_i}, \quad \text{so} \quad L_{n,k}(t_i) = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases} \]

\[ p(t) = \sum_{k=0}^{n} y_k L_{n,k}(t) \]

**Example:**
\[ L_{3,1}(t) = \frac{(t-t_0)(t-t_2)(t-t_3)}{(t_1-t_0)(t_1-t_2)(t_1-t_3)} = \frac{(t+1)(t-3)(t-5)}{(1+1)(1-3)(1-5)} \]

\[ L_{3,1}(-1) = L_{3,1}(3) = L_{3,1}(5) = 0, \quad L_{3,1}(1) = 1 \]
Polynomial interpolation is the basis of many algorithms, including numerical integration and differentiation.

**Examples**

1. \[ \int_a^b f(t) \, dt \approx (b-a) \cdot \frac{f(a)+f(b)}{2} \]
   - **Trapezoidal rule**
   - \( p = \) straight line through 2 points

2. \[ \int_a^b f(t) \, dt \approx (b-a) \cdot f\left(\frac{a+b}{2}\right) \]
   - **Midpoint rule**
   - \( p = \) horizontal line through 1 point

3. \[ \int_a^b f(t) \, dt \approx \frac{b-a}{6} \left[ f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \]
   - **Simpson’s rule**
   - \( p = \) parabola through 3 points

**Warning:** Polynomial interpolation with many points is not numerically stable. In practice, never use more than 4 or 5 points.
Generalized Polynomial Interpolation

A point $t_k$ can be repeated. If $t_k$ is repeated $m$ times, data is $y_k, y_k', \ldots, y_k^{(m-1)}$.

Fact: This has a unique solution.

Error is $\frac{w(t)}{(n+1)!} f^{(n+1)}(\xi)$, $w(t) = (t-t_0) \cdots (t-t_n)$

with appropriate repetitions

one extreme: standard interpolation (all points different)

intermediate case: Hermite interpolation (every point twice)

Given $t_0 \ldots t_n, y_0 \ldots y_n, y_0' \ldots y_n'$

$2n+2$ data $\Rightarrow p$ has degree $2n+1$

Error $\frac{(t-t_0)^2 \cdots (t-t_n)^2}{(2n+2)!} f^{(2n+2)}(\xi)$

other extreme: all points the same

Given $t_0, y_0, y_0', \ldots, y_0^{(n)}$

Taylor polynomial

Error $\frac{(t-t_0)^n}{(n+1)!} f^{(n+1)}(\xi)$

Look at size of error

\[ \frac{(t-t_0)(t-t_1) \cdots (t-t_n)}{(n+1)!} f^{(n+1)}(\xi) \]

growth fast with $n$

This is good \\

Grows like $t^n$ outside

\[ Q \]

completely outside our control
**Result 1**

Keep $t$ among $t_0$ to $t_n$ (interpolation).

Do not move $t$ outside very much (extrapolation).

![Interpolation Points](image)

$f(x) = \sin x$

*Good fit*  
*Interpolating polynomial*

*Bad fit*

**Result 2**

For given $f$, do not increase $n$ very much.

Range example

$f(t) = \frac{1}{1+t^2}$

*Interpolating polynomial*

Another consideration: In many situations, we have a choice of using a high accuracy method, with few steps, or a low accuracy method (many steps, but little work per step). Usually, the overall work needed for a given accuracy is smallest around order 4 or 5.