If $f^{(n)}(x), \forall n$ are available, Taylor polynomial is an approximation:

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2!}f''(x_0)(x-x_0)^2 + \cdots$$

Example:

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

Need only information of one point
If finite terms are selected, the approximation is accurate only near $x_0$

Primal use of Taylor polynomials for numerical methods: derivation of numerical techniques but not approximation

Subsequently we will discuss some interpolation and approximation techniques.

Roughly speaking, we have:

- Interpolation: find a function passing all the given points
Approximation: find a function so error to the given points is minimized
Lagrange Polynomials I

- Given \((x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\), find a polynomial function passing all of them.
- Why consider polynomials: the simplest form of functions.
- Degree 1 polynomial passing two points: a straight line.
Lagrange Polynomials II

- Given \((x_0, f(x_0)), (x_1, f(x_1))\). Define

\[
L_0(x) \equiv \frac{x - x_1}{x_0 - x_1}, \quad L_1(x) \equiv \frac{x - x_0}{x_1 - x_0},
\]

\[
P(x) \equiv L_0(x)f(x_0) + L_1(x)f(x_1)
\]

Then

\[
P(x_0) = 1f(x_0) + 0f(x_1) = f(x_0)
\]

\[
P(x_1) = 0f(x_1) + 1f(x_1) = f(x_1)
\]

- Generalization: higher-degree polynomials
• $(n + 1)$ points $\Rightarrow$ consider a polynomial with degree at most $n$

$$(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$$

• Construct $L_{n,k}(x)$. We hope that

$$L_{n,k}(x_i) = \begin{cases} 
0 & i \neq k \\
1 & i = k 
\end{cases}$$
Lagrange Polynomials IV

\( n \): degree, \( k \): index. Then

\[
P(x) = \sum_{k=0}^{n} L_{n,k}(x)f(x_k)
\]

Thus we have

\[
P(x_i) = f(x_i)
\]

Define

\[
L_{n,k}(x) \equiv \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}
\]
Note that here we assume $x_i \neq x_j$
Spline Interpolation I

- Disadvantages of using Lagrange polynomials
  - Very high degrees
  - Large fluctuation
- We will instead try piecewise polynomial approximation
- Different polynomials on each interval
- Linear: a series of straight lines joining

\[(x_0, f(x_0)), \ldots, (x_n, f(x_n))\]
Example

\[ x = [0 \ 1 \ 2 \ 3], \ y = [0 \ 1 \ 4 \ 3] \]
Function forms: \( s_0(x), \ldots, s_{n-1}(x) \)

\[
s_j(x) = \frac{x - x_{j+1}}{x_j - x_{j+1}} y_j + \frac{x - x_j}{x_{j+1} - x_j} y_{j+1}, \quad x_j \leq x \leq x_{j+1}
\]

Disadvantage: not differentiable at end points
A possibility is to use higher-degree $s_j(x)$ and hope that

$$s_0'(x), \ldots, s_{n-1}'(x)$$

are continuous

Thus we require that these functions satisfy

$$s_j(x_j) = f(x_j), j = 0, \ldots, n - 1,$$

$$s_{n-1}(x_n) = f(x_n)$$

$$s_{j+1}(x_{j+1}) = s_j(x_{j+1}), j = 0, \ldots, n - 2$$

$$s_{j+1}'(x_{j+1}) = s_j'(x_{j+1}), j = 0, \ldots, n - 2$$
How many conditions:

\[ n + 1 + 2(n - 1) = 3n - 1 \]

Quadratic piecewise interpolation:

3\(n\) variables

Each interval: a quadratic polynomial, 3 variables

Most common piecewise approximation: cubic polynomials (Spline)

4\(n\) variables