NENG 685, Class 3, Fall 2017
Differentiation & Integration
October 11th, 2017

Differentiation

We're building the skills required to solve

$$\frac{1}{v}\frac{\partial}{\partial t}\phi(\vec{r},t) - \nabla \cdot D\nabla\phi(\vec{r},t) + \Sigma_a\phi(\vec{r},t) = \nu\Sigma_f\phi(\vec{r},t) + S(\vec{r},t)\ .$$

In practice, we can rarely get an analytical solution for ϕ so we have to solve it some other way. If we don't know what ϕ looks like, how do we take its derivative? We need **numerical differentiation** so that we can come up with a way to express $\nabla^2\phi$ or $\frac{\partial}{\partial t}\phi$ (or anything similar in any other equation).

Problem: Given a function $f(x)$ at $x_0, x_1, \ldots, x_n$, approximate $f'(x)$ or $f''(x)$, etc.

Numerical differentiation is useful when a function is defined by

- data
- differential equations (ODEs, PDEs)

Example Given f is $C^2 \in [a,b]$ and $x_0 \in [a,b]$, find an approximation to $f'(x_0)$.

We're going to come at this from **Taylor's theorem**, which gives an approximation of a k-times differentiable function around a given point by a k-th order Taylor polynomial.

We'll start with a second order expansion for the point x_0:

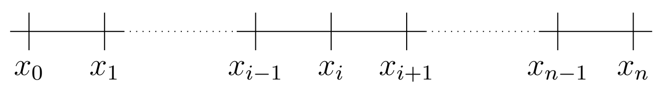
$$f(x) = \sum_{k=0}^{n}\frac{f^k(x_0)}{k!}(x-x_0)^k f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(c)}{2}(x-x_0)^2$$

$$\therefore f'(x_0) = \frac{f(x) - f(x_0)}{(x - x_0)} - \frac{f''(c)}{2}(x - x_0)$$

$$\text{Let } h = x - x_0 \to x = x_0 + h$$

$$\therefore f'(x_0) = \underbrace{\frac{f(x_0 + h) - f(x_0)}{h}}_{\text{computable approx.}} - \underbrace{\frac{f''(c)}{2}h}_{\text{error term}}$$

We can extend this principle to get higher levels of accuracy. To do that we do a Taylor expansion for each point in our collection and choose how many points to combine and in what ways.



$$\begin{aligned}
f(x_i) &= f(x_i) \\
f(x_i \pm h) &= f(x_i) \pm hf'(x_i) + h^2\frac{f''(x_i)}{2} \pm h^3\frac{f'''(x_i)}{6} + f^{(4)}(c_1)\frac{h^4}{24} \\
f(x_i \pm 2h) &= f(x_i) \pm 2hf'(x_i) + 2h^2 f''(x_i) \pm \frac{4}{3}h^3 f'''(x_i) + \frac{2}{3}h^4 f^{(4)}(c_2)
\end{aligned}$$

Note: Here, this expansion follows the same formula as the original Taylor expansion with the substitutions of $x_0 = x$ and $x = x \pm h$.

Forward difference:

$\underline{O(h)}$: combine the point and the next point forward

$$f'(x_i) = \frac{f(x_i + h) - f(x_i)}{h} - \frac{1}{2}hf''(c)$$

$\underline{O(h^2)}$: combine the point and the next two points forward; we're going to need to figure out the coefficients though:

$$
\begin{aligned}
&af(x_i) + bf(x_i + h) + cf(x_i + 2h) = f'(x_i) \\
&af(x_i) + b[f(x_i) + hf'(x_i) + h^2\frac{f''(x_i)}{2} + h^3\frac{f'''(c_1)}{6}] \\
&\qquad + c[f(x_i) + 2hf'(x_i) + 2h^2 f''(x_i) + \frac{4}{3}h^3 f'''(c_2)] = f'(x_i) \\
&(a + b + c)f(x_i) + h(b + 2c)f'(x_i) + h^2(\frac{b}{2} + 2c)f''(x_i) \\
&\qquad + h^3(\frac{b}{6} + \frac{4c}{3})f'''(\mu) = f'(x_i)
\end{aligned}
$$

Use mean value theorem to get $\mu \in [c_1, c_2]$ to be able to combine the error terms to get only one term.

KEY set the coefficients to get what we want:

$$
\begin{aligned}
(a + b + c) &= 0 \\
h(b + 2c) &= 1 \\
h^2(\frac{b}{2} + 2c) &= 0
\end{aligned}
$$

We don't have more degrees of freedom, so we're left with f''' for the error term.

Solving all of that gives

$$
\begin{aligned}
a &= -\frac{3}{2h} \\
b &= \frac{2}{h} \\
c &= -\frac{1}{2h} \\
\text{error term } &= -\frac{1}{3}h^2 f'''(\mu) \\
\therefore \quad &\boxed{f'(x_i) = \frac{-3f(x_i) + 4f(x_i + h) - f(x_i + 2h)}{2h} - \frac{1}{3}h^2 f'''(\mu)}
\end{aligned}
$$

Backward difference:

Essentially, the signs all flip because we use points in the other direction.

$\underline{O(h)}$: combine the point and the next point backward

$$f'(x_i) = \frac{f(x_i) - f(x_i - h)}{h} + \frac{1}{2}hf''(\mu)$$

$\underline{O(h^2)}$: combine the point and the next two points backward

$$f'(x_i) = \frac{3f(x_i) - 4f(x_i - h) + f(x_i - 2h)}{2h} + \frac{1}{3}h^2 f'''(\mu)$$

Central difference:

Use points on either side (the average of forward and backward difference) $\rightarrow$ more accuracy for the same number of evaluation points.

$\underline{O(h^2)}$: combine one point on either side

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h} - \frac{1}{6}h^2 f'''(\mu)$$

$\underline{O(h^4)}$: combine two points on either side

$$f'(x_i) = \frac{f(x_i - 2h) - 8f(x_i - h) + 8f(x_i + h) - f(x_i + 2h)}{2h} - \frac{1}{30}h^4 f^{(5)}(\mu)$$

Higher Order Derivatives

By solving for different terms in the Taylor expansion, combining points, and solving for coefficients we can get other derivative terms as well.

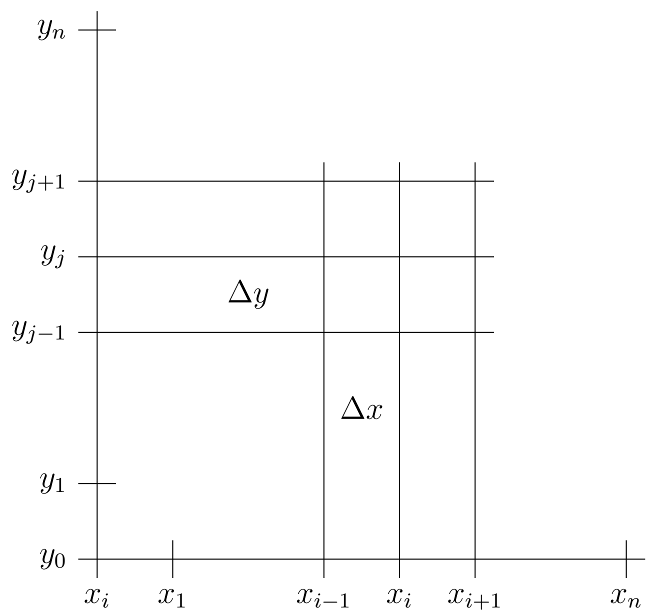
E.g. $af(x_i + h) + bf(x_i) + cf(x_i - h) = f''(x_i)$

These are all $O(h^2)$:

$$f''(x_i) = \frac{f(x_i - h) - 2f(x_i) + f(x_i + h)}{h^2} + \frac{h^2}{12} f^{(4)}(\mu)$$
$$f^{(3)}(x_i) = \frac{-f(x_i - 2h) + 2f(x_i - h) - 2f(x_i + h) + f(x_i + 2h)}{2h^3}$$
$$f^{(4)}(x_i) = \frac{f(x_i - 2h) - 4f(x_i - h) + 6f(x_i) - 4f(x_i + h) + f(x_i + 2h)}{h^4}$$

Two Variables

$$\frac{\partial^2 f_{i,j}}{\partial x^2} = \frac{\partial}{\partial x}(\frac{\partial f_{i,j}}{\partial x}) = \frac{f_{i-1,j} - 2f_{i,j} + f_{i+1,j}}{\Delta x^2}$$
$$\frac{\partial^2 f_{i,j}}{\partial y^2} = \frac{\partial}{\partial y}(\frac{\partial f_{i,j}}{\partial y}) = \frac{f_{i,j-1} - 2f_{i,j} + f_{i,j+1}}{\Delta y^2}$$



Integration

Why might we need to numerically integrate?

- The integrand, $f(x)$, is complicated enough that its indefinite or even definite integral over a particular range cannot be found.

- The indefinite or definite integral can be found, but may take long time to calculate analytically.
- The integrand is given in terms of a table. That is, we are provided with discrete values of $f(x)$ at $n+1$ data points.

For us, we use this particularly for the scattering and fission in the Transport Equation:

$$\int_{4\pi} d\hat{\Omega}' \int_0^\infty dE' \Sigma_s(E', \hat{\Omega}' \to E, \hat{\Omega}) \psi(\vec{r}, E', \hat{\Omega}', t) + \frac{\chi(E)}{4\pi} \int_0^\infty dE' \nu(E') \Sigma_f(E') \int_{4\pi} d\hat{\Omega}' \psi(\vec{r}, E', \hat{\Omega}', t)$$

The general strategy is:

- Approximate the integrand $f(x)$ with either a global interpolating polynomial defined over the whole domain of integration, or a collection of local interpolating polynomials defined over intervals that are subtended by a small group of data points.
- Integrate the interpolating polynomial(s) over their individual domains of definition.

Thus, given $f \in [a, b]$, compute $I(f) = \int_a^b f(x)dx$.

The way we do this is called **numerical integration** or **numerical quadrature**:

$$I(f) \approx I_n(f) = \sum_{i=0}^{n} w_i f(x_i)$$

$w_i \equiv$ quadrature weights
$x_i \equiv$ quadrature points

Approach #1
Fix quadrature points (x_i), then choose quadrature weights (w_i). Usually fit a polynomial to $f(x_i)$ and integrate.

1. <u>Newton-Cotes</u>: use a single poly over the entire interval
2. <u>composite Newton-Cotes</u>: split the interval
3. <u>Romberg integration</u>: extrapolation

Approach #2
For a given n, choose the "best" quadrature points (x_i) and quadrature weights (w_i):
Gaussian Quadrature

the **degree of precision** of a quadrature formula, $I_n(f)$, is the positive integer m s.t.:

1. $I(p) = I_n(p)$ for every poly of degree $\leq m$.
2. $I(p) \neq I_n(p)$ for some poly of degree $m + 1$.

Newton-Cotes formula

idea: fix $x_0, x_1, \dots, x_n$ then interpolate $f(x)$ by a poly of degree $\leq n$.

$$\begin{array}{ccccc} I(f) & \approx & I_n(f) & = & I(p_n) \\ \text{true integral} & \approx & \text{N-C formula} & = & \text{true integral of interpolating poly} \end{array}$$

The only error is in the polynomial interpolation.

Lagrange form of NC

Choosing Lagrange polynomials prevents us from needing to recalculate weights for every $f(x)$ (since we've fixed the points). Note, we're assuming equally spaced points: $h = (b - a)/n$. This is not a necessary assumption, but indexing of h would be required for unequally spaced points.

$$\begin{aligned}
P_n(x) &= \sum_{i=0}^{n} f(x_i)L_i(x) \\
I(P_n(x)) &= \sum_{i=0}^{n} \big[\int_a^b L_i(x)dx\big] f(x_i) \\
&= \sum_{i=0}^{n} w_i f(x_i) \\
\text{where } w_i &= \int_a^b L_i(x)dx = \int_a^b \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)} \\
&\text{sub in } x = a + sh \\
&= (b-a)\frac{1}{n}\int_0^n \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(s-i)}{(k-i)} ds
\end{aligned}$$

We can see w_i are not dependent on $f(x)$.

Trapezoid Rule

This is the Lagrange form of Newton-Cotes integration with $n = 1$.

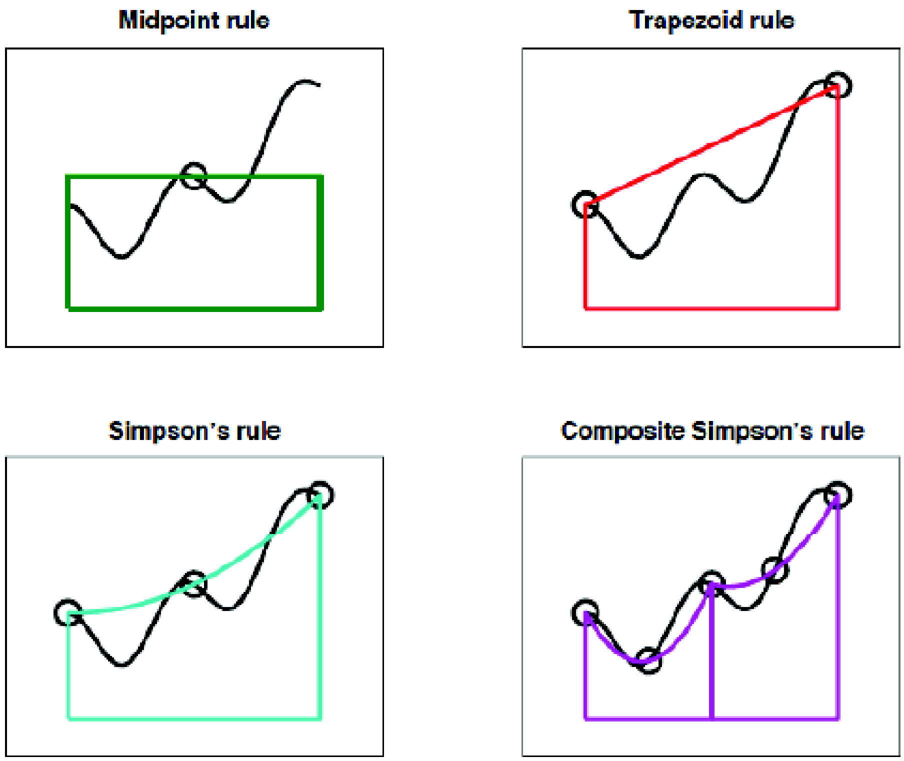


Figure 1: examples of how different integration rules capture a function

$x_0 = a$, $x_1 = b$, $h = x_1 - x_0$

$$L_0(x) = \frac{x - x_1}{x_0 - x_1}\,, \qquad L_1(x) = \frac{x - x_0}{x_1 - x_0}$$

$$w_0 = \int_{x_0}^{x_1} (\frac{x - x_1}{x_0 - x_1}) dx\,;$$

let $x = x_0 + th\,,\quad dx = h\,dt\,,\quad t = \dfrac{x - x_0}{h}\,,$

change bounds and note that the denominator is $-h$

$$w_0 = h \int_0^1 (\frac{x_1 - x_0 - ht}{h}) dt = \int_0^1 h(1 - t) dt$$

$$w_0 = h\big(t - \frac{1}{2}t^2\big)|_0^1 = \frac{1}{2}h$$

$$w_1 = \int_{x_0}^{x_1} (\frac{x - x_0}{x_1 - x_0}) dx = \frac{1}{2}h$$

$$\therefore I_1(f) = w_0 f(x_0) + w_1 f(x_1) = \boxed{\frac{h}{2}\big(f(x_0) + f(x_1)\big)} \rightarrow \text{area of a trapezoid}$$

The trapezoid rule has degree of precision 1.

Error Analysis

1. interpolate:
$$f(x) = P_n(x) + R_n(x)$$

2. integrate:
$$\underbrace{\int_a^b f(x)dx}_{I(f)} = \underbrace{\int_a^b P_n(x)dx}_{I_n(f)} + \int_a^b R_n(x)dx$$

Example: Trapezoid Rule

Recall from Lagrange polynomials that we know how to express R_n. We can then use that to compute the error of the integral.

$$\begin{aligned} R_1(x) &= \frac{f''(c)}{2!}(x-x_0)(x-x_1) \\ I(f) - I_1(f) &= \int_{x_0}^{x_1} \frac{f''(c)}{2!} \underbrace{(x-x_0)(x-x_1)}_{\text{doesn't change sign over interval}} \, dx \end{aligned}$$

Recall that c depends on x. To get an expression, we're going to use the mean value theorem (which I briefly referenced last time).

<u>Mean Value Theorem for Integrals</u>

$$\int_a^b f(x)g(x)dx = f(\xi)\int_a^b g(x)dx \, , \qquad a \leq \xi \leq b$$

Where $g(x)$ must always be positive or always negative (cannot change sign over the interval).

Thus, we can use MVT for our integral to see that

$$\begin{aligned} I(f) - I_1(f) &= \frac{1}{2}f''(\eta)\int_{x_0}^{x_1}(x-x_0)(x-x_1)dx \\ &= \frac{1}{2}f''(\eta)\left(\frac{-h^3}{6}\right) \rightarrow \boxed{\frac{-h^3}{12}f''(\eta)} \end{aligned}$$

is the error term for trapezoid rule. You can see that this has <u>degree of precision 1 b/c the second derivative drives the error</u> and *the error changes as $O(h^3)$ with mesh spacing*.

Simpson's Rule

This is the Lagrange form of Newton-Cotes integration with $n = 2$.

$$
\begin{aligned}
&x_0 = a \qquad x_1 = \frac{a+b}{2} \qquad x_2 = b \qquad h = \frac{b-a}{2}\\
&L_0 = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\\
&L_1 = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\\
&L_2 = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\\
&w_0 = \int_{x_0}^{x_2} L_0(x)dx = \frac{h}{3}\\
&w_1 = \int_{x_0}^{x_2} L_1(x)dx = \frac{4h}{3}\\
&w_2 = \int_{x_0}^{x_2} L_2(x)dx = \frac{h}{3}\\
&\quad \text{Note: weights sum to } 2h\text{, the size of the interval}\\
&\quad \therefore I_2(f) = \frac{h}{3}\big(f(x_0) + 4f(x_1) + f(x_2)\big)
\end{aligned}
$$

The error term comes from integrating $R_2 = \frac{1}{3!}f'''(c)(x-x_0)(x-x_1)(x-x_2)$. However, the middle term now changes sign over the interval and we can't just use MVT.

We're going to skip the details, but what we get out is that the error for Simpson's Rule looks like:

$$
\frac{-f^{(4)}(\eta)}{4!}\frac{h^5}{\frac{4}{15}} \to \boxed{\frac{-f^{(4)}(\eta)}{90}h^5}
$$

Thus, Simpson's rule has degree of precision 3 and is $O(h^5)$.

In General:

n	error	precision
odd	$O(h^{n+2})$	n
even	$O(h^{n+3})$	$n+1$

Simpson's 3/8 rule:

If $n = 3$:
$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}\big(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\big) - \frac{3h^5}{80}f^{(4)}(\xi)$$
where $x_0 < \xi < x_3$ and $h = (x_3 - x_0)/3$.

Composite Newton-Cotes

Some difficulties:

1. To add points we must recompute the weights, w_i, which can be a lot of work.
2. high order polynomial interpolation on equally spaced points is BAD. (using optimally-spaced points is Gaussian quadrature; we'll get to that later).

Idea: subdivide $[a, b]$ into subintervals. On each of these apply a low-order Newton-Cotes formula.

Composite Trapezoid:

Just apply the trapezoid rule on each subinterval. Given $x_0, x_1, \ldots, x_n$ and $f(x_0), f(x_1), \ldots f(x_n)$, the grid space is $h = \frac{x_n - x_0}{n}$.

DRAW PICTURE
$$\begin{aligned}\int_a^b f(x)dx &= \sum_{i=1}^{n}\int_{x_{i-1}}^{x_i} f(x)dx \\ &= \sum_{i=1}^{n}\Big[\frac{h}{2}\big(f(x_{i-1}) + f(x_i)\big) - \frac{h^3}{12}f''(c_i)\Big] \\ &= \frac{h}{2}\Big(f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\Big) - \frac{h^3}{12}\underbrace{\sum_{i=1}^{n} f''(c_i)}_{\substack{nf''(\mu),\\ n=(b-a)/h}}\end{aligned}$$
$$\int_a^b f(x)dx = \boxed{\frac{h}{2}\Big(f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\Big) - \frac{h^2}{12}(b-a)f''(\mu)}$$

Composite Simpson:

Similarly, apply the Simpson rule on each subinterval. Note that for the functionally-equivalent sense of n intervals we really need to use $2n$ points since each Simpson is applied using 3 points, and the end points are re-used. Therefore $\int_a^b f(x)dx =$

$$\sum_{i=1}^{n/2}\int_{x_{2i-1}}^{x_{2i}} f(x)dx = \sum_{i=1}^{n/2}\Big[\frac{h}{3}\big(f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})\big) - \frac{f^{(4)}(c_i)}{90}h^5\Big]$$

$$\text{MVT: } \sum_{i=1}^{n/2} f^{(4)}(c_i) = \frac{n}{2}f^{(4)}(\mu) = \frac{b-a}{2h}f^{(4)}(\mu)$$

$$= \frac{h}{3}\big(f(a) + \underbrace{4\sum_{i=1}^{n/2} f(x_{2i-1})}_{\text{odd points}} + \underbrace{2\sum_{i=1}^{n/2-1} f(x_{2i})}_{\text{even points}} + f(b)\big) - \frac{h^4}{180}(b-a)f^{(4)}(\mu)$$

Open Newton-Cotes

The previous examples were called **closed** Newton-Cotes because $f(x)$ is evaluated at the first and last points.

Open Newton-Cotes does not evaluate $f(x)$ at the endpoints. The node points are still defined as $x_j = x_0 + jh, j = 0, \ldots, n$, but now

Item	Open	Closed
h	$\frac{b-a}{n+2}$	$\frac{b-a}{n}$
x_0	$a+h$	a
x_n	$b-h$	b

Now our real endpoints are x_{-1} and x_{n+1} (so we're doing $\int_{x_{-1}}^{x_{n+1}} f(x)dx$), which gives

$$w_i = \int_a^b L_i(x)dx = \int_a^b \prod_{\substack{i=0\\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k - x_i)} = (b-a)\frac{1}{n+2}\int_0^n \prod_{\substack{i=0\\ i\neq k}}^{n} \frac{(s-i)}{(k-i)}ds$$

The midpoint rule is what we get with $n=0$; it only uses one point:

$$\int_a^b f(x)dx = \int_{x_{-1}}^{x_1} f(x)dx = 2hf(x_0) + \frac{h^3}{3}f''(\xi)$$

where $x_{-1} < \xi < x_1$ and $h = (b-a)/2$.

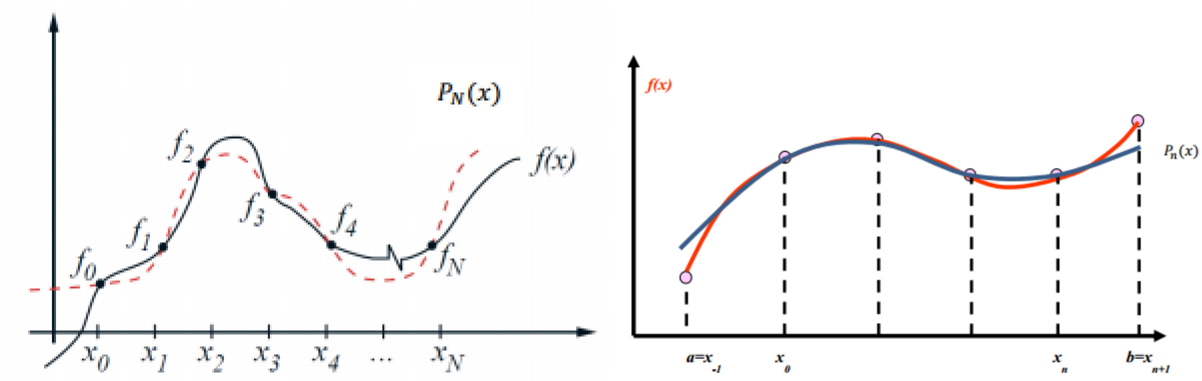


Figure 2: [left] example of **closed** Newton Cotes (where the interpolation *matches* at the end points); [right] example of **open** Newton Cotes (where the interpolation *does not match* at the end points)

$\underline{n=1}$:

$$\int_{x_{-1}}^{x_2} f(x)dx = \frac{3h}{2}[f(x_0) + f(x_1)] + \frac{3h^3}{4} f''(\xi)$$

where $x_{-1} < \xi < x_2$ and $h = (b-a)/3$.

$\underline{n=2}$:

$$\int_{x_{-1}}^{x_3} f(x)dx = \frac{4h}{3}[2f(x_0) - f(x_1) + 2f(x_2)] + \frac{14h^5}{45} f^{(4)}(\xi)$$

where $x_{-1} < \xi < x_3$ and $h = (b-a)/4$.