Preliminaries

Definitions

• [Rate of Convergence] Let $(\alpha_n)_n, (\beta_n)_n$ be sequences that converges to α and 0 respectively. If $\exists K$ s.t. $\lvert\alpha_n - \alpha\rvert \leq K\lvert\beta_n\rvert$ for large n, then $(\alpha_n)_n$ converges to α with <u>rate of convergence</u> $O(\beta_n)$. Write $\alpha_n = \alpha + O(\beta_n)$
• [Big O] Let F, G be functions s.t. $\lim_{h\to 0} F(h) = L$ and $\lim_{h\to 0} G(h) = 0$. If $\exists K$ s.t. $\lvert F(h) - L\rvert \leq K\lvert G(h)\rvert$ for small enough h, then write $F(h) = L + O\big(G(h)\big)$
• [Order of Convergence] Let $(p_n)_n$ be a sequence that converges to p, with $p_n \neq p\ \forall n$. If $\lim_{n\to\infty} \frac{\lvert p_{n+1} - p\rvert}{\lvert p_n - p\rvert^\alpha} = \lambda$, then $(p_n)_n$ <u>converges to p of order α</u>, with asymptotic error constant λ
    ○ [Linearly Convergent] $\alpha = 1$ and $\lambda < 1$ implies linearly convergent
    ○ [Quadratically Convergent] $\alpha = 2$ implies quadratically convergent
• [Superlinearly Convergent] Say sequence $(p_n)_n$ is <u>superlinearly convergent</u> to p if $\lim_{n\to\infty} \frac{\lvert p_{n+1} - p\rvert}{\lvert p_n - p\rvert} = 0$

Root Finding

Definitions

- [Horner Method] A method for evaluating polynomials via nesting
  - $P(x_0) = a_0 + x_0\big(a_1 + x_0(a_2 + \cdots)\big)$
  - $P'(x_0)$ and $P(x_0)$ can be evaluated in a single pass by considering $P(x) = (x - x_0)Q(x) + b_0$ with $Q(x) = b_1 + b_2 x + \cdots$ and calculating $(b_i)_i$
- [Bisection Method] Finds solution to $f(x) = 0$ for continuous function f on interval $[a, b]$; requires that $f(a)f(b) < 0$
- [Fixed Point] Given function g, p is a fixed point for g if $g(p) = p$
- [Fixed Point Iteration] Start with p_0 and define $p_{n+1} = g(p_n)$ to generate $(p_n)_n$
- [Newton's Method] Root finding to $f(x) = 0$; $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
  - View as a special fixed point iteration with $g(p) = p - \frac{f(p)}{f'(p)}$
  - Fails if $f'(x_n) = 0$ for some x_n
  - If $f(p) = 0$ and $f'(p) \neq 0$, then for starting values sufficiently close to p, Newton's method will converge at least quadratically
- [Secant's Method] Root finding; $x_{n+1} = x_n - \frac{f(x_n)}{\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}}$
  - Doesn't need derivative evaluation (unlike Newton)
- [Simple Zero] Let $f \in C^1[a, b]$. Then f has a simple zero at $p \in (a, b)$ if $f(p) = 0$ but $f'(p) \neq 0$.
- [Zero of Multiplicity m] A solution p of $f(x) = 0$ is a zero of multiplicity m of f if for $x \neq p$, we can write $f(x) = (x - p)^m q(x)$ where $\lim_{x \to p} q(x) \neq 0$
- [Forward Difference] $\Delta p_n = p_{n+1} - p_n$, $\Delta^k p_n = \Delta(\Delta^{k-1} p_n) = \Delta^{k-1} p_{n+1} - \Delta^{k-1} p_n$
- [Aitken's Δ^2 Method] Define $\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n} = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n} = \{\Delta^2\}(p_n)$. The assumption is that $(\hat{p}_n)_n$ converges more rapidly to p than $(p_n)_n$
- [Forward Difference] $\Delta p_n = p_{n+1} - p_n$
- [Steffensen's Method]
  - $p_0^{(0)}$; $p_1^{(0)} = g\left(p_0^{(0)}\right)$; $p_2^{(0)} = g\left(p_1^{(0)}\right)$
  - $p_0^{(1)} = \{\Delta^2\}\left(p_0^{(0)}\right)$; $p_1^{(1)} = g\left(p_0^{(1)}\right)$; $p_2^{(1)} = g\left(p_1^{(1)}\right)$
  - $p_0^{(2)} = \{\Delta^2\}\left(p_0^{(1)}\right)$
- [Müller's Method] Finds a solution to $f(x) = 0$ given three initial points p_0, p_1, p_2; uses parabolas to interpolate and find closer root to p_2; can find real and complex roots.
  - $p_3 = p_2 - \frac{2c}{b + \operatorname{sign}(b)\sqrt{b^2 - 4ac}}$

Theorems

- [2.1] Let $f \in C[a, b]$ with $f(a)f(b) < 0$. Then, bisection method generates $(p_n)_n$ with $|p_n - p| \leq \frac{b-a}{2^n}$ i.e. $p_n = p + O\left(\frac{1}{2^n}\right)$
- [2.3] Let $g \in C[a, b]$ and $g(x) \in [a, b]$. Then g has at least one fixed point in $[a, b]$. If in addition, $g'(x)$ exists on (a, b) and $\exists K < 1$ s.t. $|g'(x)| \leq K\ \forall x \in (a, b)$, then there is exactly one fixed point in $[a, b]$.
- [2.4] Let $g \in C[a, b]$ be s.t. $g(x) \in [a, b]\ \forall x \in [a, b]$. Suppose g' exists on (a, b) and $\exists K \in (0,1)$ s.t. $|g'(x)| \leq K$ for $x \in (a, b)$. Then, for any number $p_0 \in [a, b]$, $\lim_{n \to \infty} p_n = p$.
  - [2.5] $|p_n - p| \leq K^n \max(p_0 - a, b - p_0)$
  - [2.8] If $g'(p) \neq 0$, then for any $p_0 \neq p$ in $[a, b]$, the sequence $p_n = g(p_{n-1})$ converges linearly with asymptotic constant $|g'(p)|$

- [2.6] Let $f \in C^2[a,b]$. If $p \in (a,b)$ s.t. $f(p) = 0$ and $f'(p) \neq 0$, then $\exists \delta > 0$ s.t. Newton's method generates sequence $(p_n)_n$ converging to p for any initial $p_0 \in [p-\delta, p+\delta]$
- [2.9] Let p be a solution of $g(x) = x$. Suppose $g'(p) = 0$ and g'' continuous with $|g''(x)| < M$ on an open interval I containing p, then $\exists \delta > 0$ s.t. for $p_0 \in [p-\delta, p+\delta]$, the sequence $(p_n)_n$ converges at least quadratically to p. Moreover, for sufficiently large values of n, $|p_{n+1} - p| < \frac{M}{2}|p_n - p|^2$
- [2.11] Let $f \in C^1[a,b]$. Then f has a simple zero at $p \in (a,b)$ if and only if $f(p) = 0$ and $f'(p) \neq 0$.
- [2.14] Let $(p_n)_n$ converge linearly to limit p with $\lim_{n\to\infty} \frac{p_{n+1}-p}{p_n-p} < 1$. Then Aitken's Δ^2 sequence $(\hat{p}_n)_n$ converges to p faster in the sense that $\lim_{n\to\infty} \frac{\hat{p}_n-p}{p_n-p} = 0$
- [2.15] Suppose $x = g(x)$ has solution p with $g'(p) \neq 1$. If there exists $\delta > 0$ s.t. $g \in C^3[p-\delta, p+\delta]$, then Steffensen's method gives quadratic convergence for any $p_0 \in [p-\delta, p+\delta]$

Interpolation

Definitions (General Interpolation)

- [Lagrange Interpolating Polynomial] $P(x) = \sum_{i=0}^{n} f(x_i) \prod_{j \neq i} \frac{(x-x_i)}{(x_i - x_j)} = \sum_{i=0}^{n} f(x_i) L_i(x)$
  - Then $f(x) = \sum_{k=0}^{n} f(x_k) L_k(x) + \frac{(x-x_0)\dots(x-x_n)}{(n+1)!} f^{(n+1)}\big(\xi(x)\big)$ for some $\xi(x) \in I$
  -
- [Neville's Method] Let $Q(x)$ interpolate $f(x)$ at $x_0, \dots, x_k$ and $\hat{Q}(x)$ interpolate $f(x)$ at $x_1, \dots, x_{k+1}$, then $P(x) = \frac{(x-x_{k+1})Q(x) - (x-x_0)\hat{Q}(x)}{x_0 - x_{k+1}}$ interpolates all points.
  - $Q_i(x) = f(x_i)$
  - $Q_{i,i+1,\dots,j,j+1}(x) = \frac{(x-x_{j+1})Q_{i,\dots,j}(x) - (x-x_i)Q_{i+1,\dots,j+1}(x)}{x_i - x_{j+1}}$
  - Can do on-the-fly memoisation

Neville	Divided Difference
$\begin{array}{ccccccccc} f(x_0) & \searrow \\ f(x_1) & \overset{\longrightarrow}{\searrow} & Q_{0,1}(x) & \searrow \\ f(x_2) & \overset{\longrightarrow}{\searrow} & Q_{1,2}(x) & \overset{\longrightarrow}{\searrow} & Q_{0,1,2}(x) \\ \vdots & & & & & & \ddots \\ f(x_{n-2}) & \overset{\longrightarrow}{\searrow} & Q_{n-3,n-2}(x) & \overset{\longrightarrow}{\searrow} & Q_{n-4,n-3,n-2}(x) & \overset{\longrightarrow}{\searrow} & \cdots \\ f(x_{n-1}) & \overset{\longrightarrow}{\searrow} & Q_{n-2,n-1}(x) & \overset{\longrightarrow}{\searrow} & Q_{n-3,n-2,n-1}(x) & \overset{\longrightarrow}{\searrow} & \cdots & \searrow \\ f(x_n) & \longrightarrow & Q_{n-1,n}(x) & \longrightarrow & Q_{n-2,n-1,n}(x) & \longrightarrow & \cdots & \longrightarrow & Q_{0,1,\cdots,n}(x) \end{array}$	$\begin{array}{ccccccccc} f[x_0] & \searrow \\ f[x_1] & \overset{\longrightarrow}{\searrow} & f[x_0, x_1] & \searrow \\ f[x_2] & \overset{\longrightarrow}{\searrow} & f[x_1, x_2] & \overset{\longrightarrow}{\searrow} & f[x_0, x_1, x_2] \\ \vdots & & & & & & \ddots \\ f[x_{n-2}] & \overset{\longrightarrow}{\searrow} & f[x_{n-3}, x_{n-2}] & \overset{\longrightarrow}{\searrow} & f[x_{n-4}, x_{n-3}, x_{n-2}] & \overset{\longrightarrow}{\searrow} & \cdots \\ f[x_{n-1}] & \overset{\longrightarrow}{\searrow} & f[x_{n-2}, x_{n-1}] & \overset{\longrightarrow}{\searrow} & f[x_{n-3}, x_{n-2}, x_{n-1}] & \overset{\longrightarrow}{\searrow} & \cdots & \searrow \\ f[x_n] & \longrightarrow & f[x_{n-1}, x_n] & \longrightarrow & f[x_{n-2}, x_{n-1}, x_n] & \longrightarrow & \cdots & \longrightarrow & f[x_0, \cdots, x_n] \end{array}$

- [Divided Difference] Given $\big(x_0, f(x_0)\big), \dots, \big(x_n, f(x_n)\big)$, produces the coefficients of the interpolating polynomial of the specific form $P_n(x) = a_0 + a_1(x - x_0) + \dots + a_n(x - x_0)\dots(x - x_{n-1})$
  - [Zeroth Divided Difference] $f[x_i] = f(x_i)$
  - $f[x_i, x_i] = f'(x_i)$
  - $f\big[x_i, \dots, x_j\big] = \frac{f[x_{i+1}, \dots, x_j] - f[x_i, \dots, x_{j-1}]}{x_j - x_i}$
  - $a_n = f[x_0, \dots, x_{n-1}]$
  - $P_n(x) = f[x_0] + \sum_{i=1}^{n} f[x_0, \dots, x_i](x - x_0)\dots(x - x_{i-1})$
  - Can also do on-the-fly memoisation
- [Osculating Polynomial] Let $x_0, x_1, \dots, x_n$ be $n+1$ distinct numbers in $[a, b]$ and m_i be a nonnegative integer. Suppose that $f \in C^m[a, b]$ where $m = \max m_i$. Then, the osculating polynomial approximating f is $P(x)$ of the least degree s.t. $\frac{\mathrm{d}^k}{\mathrm{d}x^k} P(x_i) = \frac{\mathrm{d}^k f(x_i)}{\mathrm{d}x^k}$ for $i \in \{0, \dots, n\}$ and $k \in \{0, \dots, m\}$
- [Hermite Polynomial] A Hermite polynomial is an osculating polynomial for $m_i = 1$.
  - Let $f \in C^1[a, b]$ with $x_0, \dots, x_n \in [a, b]$ distinct. Then the unique polynomial $P(x)$ of least degree agreeing with f and f' at $x_0, \dots, x_n$ is the Hermite polynomial of degree at most $2n + 1$ satisfying $H(x_i) = f(x_i)$, $H'(x_i) = f'(x_i)$
  - If $f \in C^{2n+2}[a, b]$, then $f(x) = H_{2n+1}(x) + \frac{(x-x_0)^2 \dots (x-x_n)^2}{(2n+2)!} f^{(2n+2)}\big(\xi(x)\big)$ for $\xi(x) \in (a, b)$

Theorems

- [Weierstrass Approximation Theorem] Let $f \in C([a, b])$. For every $\epsilon > 0$, $\exists P(x)$ s.t. $|f(x) - P(x)| < \epsilon \ \forall x \in [a, b]$
- [Lagrange Interpolation] Let $(x_i, y_i)_{i=1}^{n}$ be points. Then $P(x) = \sum_{i=1}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$ interpolates the points.
- [3.3] Let $x_0, \dots, x_n$ be distinct points in $[a, b]$ and $f \in C^{n+1}[a, b]$. Then $\forall x \in [a, b]$, $\exists \xi_x$ s.t. $f(x) = P(x) + \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x - x_0)\dots(x - x_n)$

- $g(t) = f(t) - P(t) - \big(f(x) - P(x)\big)\prod_{i=0}^{n}\frac{t-x_i}{x-x_i}$

- [3.6] Suppose $f \in C^n([a,b])$ and $x_0,\dots,x_n$ distinct in $[a,b]$. Then $\exists\xi \in (a,b)$ with $f[x_0,\dots,x_n] = \frac{f^{(n)}(\xi)}{n!}$
- [Forward Difference] $P_n(x) = P_n(x_0+sh) = f(x_0+sh) + \sum_{k=1}^{n}\binom{s}{k}\Delta^k f(x_0)$
  - $h = x_{i+1} - x_i$ (equal difference)
  - $x = x_0 + sh$
  - $\Delta f(x_0) = f(x_1) - f(x_0)$
  - $\Delta^2 f(x_0) = \Delta f(x_1) - \Delta f(x_0)$

Definitions (Splines)

- [Cubic Spline Interpolant] Let f be defined on $[a,b]$
  - $S_j(x)$: cubic polynomial on $[x_j, x_{j+1}]$ for $j \in \{0,\dots,n-1\}$
  - $S_j(x_j) = f(x_j)$
  - $S_j(x_{j+1}) = f(x_{j+1})$
  - $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$
  - $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$
  - [Natural] $S''(x_0) = S''(x_n) = 0$
  - [Clamped] $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$
- [Guidepoint] Helps to specify the derivative at an endpoint
  - For (x_0,y_0), the guidepoint is $(x_0+\alpha_0, y_0+\beta_0)$
  - For (x_1,y_1), the guidepoint is $(x_1-\alpha_1, y_1-\beta_1)$
  - Cubic Hermite polynomial satisfies $x'(0) = \alpha_0$, $x'(1) = \alpha_1$, $y'(0) = \beta_0$, $y'(1) = \beta_1$
  - $x(t) = [2(x_0-x_1) + (\alpha_0+\alpha_1)]t^3 + [3(x_1-x_0) - (\alpha_1+2\alpha_0)]t^2 + \alpha_0 t + x_0$
  - $y(t) = [2(y_0-y_1) + (\beta_0+\beta_1)]t^3 + [3(y_1-y_0) - (\beta_1+2\beta_0)]t^2 + \beta_0 t + y_0$
- [Cubic Bezier Polynomial] Given $n+1$ data points $\big(x(t_0), y(t_0)\big),\dots,\big(x(t_n), y(t_n)\big)$ and $\frac{\mathrm{d}y}{\mathrm{d}x}|_{t_i}$, find $2n$ cubic Hermite polynomials satisfying: $x_i(t) = x(t), y_i(t) = y(t)$.
  - $x(t) = [2(x_0-x_1) + 3(\alpha_0+\alpha_1)]t^3 + [3(x_1-x_0) - 3(\alpha_1+2\alpha_0)]t^2 + 3\alpha_0 t + x_0$
  - $y(t) = [2(y_0-y_1) + 3(\beta_0+\beta_1)]t^3 + [3(y_1-y_0) - 3(\beta_1+2\beta_0)]t^2 + 3\beta_0 t + y_0$
  - $t \in [0,1]$
- [Bezier Curve Algorithm]
  - Input: $n, (x_0,y_0),\dots,(x_n,y_n), (x_0^+,y_0^+),\dots,(x_{n-1}^+,y_{n-1}^+), (x_1^-,y_1^-),\dots,(x_n^-,y_n^-)$
  - Output: $\left\{a_0^{(i)}, a_1^{(i)}, a_2^{(i)}, a_3^{(i)}, b_0^{(i)}, b_1^{(i)}, b_2^{(i)}, b_3^{(i)}\right\}$

Differentiation

Forward Difference Formula

- [Forward Difference Formula] $f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h}$ where $h > 0$
- [Backward Difference Formula] $f'(x_0) = \frac{f(x_0+h)-f(x_0)}{h}$ where $h < 0$
- [Error Formula] $\left|\left(f(x_0+h) - \frac{f(x_0)}{h}\right) - f'(x_0)\right| = \left|\frac{h}{2}f''(\xi)\right|$ for $\xi \in (x_0, x_0+h)$
- [Three Point Formula] $O(h^2)$
  - $L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}$, $L_0'(x) = \frac{2x-x_1-x_2}{(x_0-x_1)(x_0-x_2)}$
  - $f'(x_k) = f(x_0)\frac{2x_k-x_1-x_2}{(x_0-x_1)(x_0-x_2)} + f(x_1)\frac{2x_k-x_0-x_2}{(x_1-x_0)(x_1-x_2)} + f(x_2)\frac{2x_k-x_0-x_1}{(x_2-x_0)(x_2-x_1)} + \frac{1}{6}f^{(3)}(\xi_k)\prod_{j\neq k}(x_k-x_j)$
  - [Three-Point Endpoint Formula]
    - $f'(x_0) = \frac{1}{2h}[-3f(x_0)+4f(x_0+h)-f(x_0+2h)] + \frac{h^2}{3}f^{(3)}(\xi_0)$
    - $\xi_0 \in (x_0, x_0+2h)$
  - [Three-Point Midpoint Formula]
    - $f'(x_0) = \frac{1}{2h}[-f(x_0-h)+f(x_0+h)] - \frac{h^2}{6}f^{(3)}(\xi_1)$
    - $\xi_1 \in (x_0-h, x_0+h)$
- [Five Point Formula] $O(h^4)$
  - [Five-Point Midpoint Formula]
    - $f'(x_0) = \frac{1}{12h}\big(f(x_0-2h)-8f(x_0-h)+8f(x_0+h)-f(x_0+2h)\big) + \frac{h^4}{30}f^{(5)}(\xi)$
    - $\xi \in (x_0-2h, x_0+2h)$
  - [Five-Point Endpoint Formula]
    - $f'(x_0) = \frac{1}{12h}\big(-25f(x_0)+48f(x_0+h)-36f(x_0+2h)+16f(x_0+3h)-3f(x_0+4h)\big) + \frac{h^4}{5}f^{(5)}(\xi)$
    - $\xi \in (x_0, x_0+4h)$
- [Second Derivative Midpoint Formula] $O(h^2)$ if $f^{(4)}$ bounded
  - $f''(x_0) = \frac{1}{h^2}\big(f(x_0-h)-2f(x_0)+f(x_0+h)\big) - \frac{h^2}{12}f^{(4)}(\xi)$
  - $\xi \in (x_0-h, x_0+h)$
- [Round off Error] $\frac{\xi}{h} + \frac{h^2}{6M}$ where M is a bound for $|f|$

Extrapolation

- Combine inaccurate $O(h)$ approximations to get formulas with higher order error
- M: estimand, $N_1(h)$: estimator
  - $M = N_1(h) + K_1h + K_2h^2 + K_3h^3 + \cdots$ where K_i are constants
  - $M = N_1\left(\frac{h}{2}\right) + K_1\frac{h}{2} + K_2\left(\frac{h}{2}\right)^2 + K_3\left(\frac{h}{2}\right)^3 + \cdots$
- [Normal Extrapolation]
  - $N_2(h) = 2N_1\left(\frac{h}{2}\right) - N_1(h)$
  - $N_{j+1}(h) = \frac{2^j}{2^j-1}N_j\left(\frac{h}{2}\right) - \frac{1}{2^j-1}N_j(h)$, $M - N_j(h) = O(h^j)$
- [Even Power Extrapolation] $N_{j+1}(h) = N_j\left(\frac{h}{2}\right) + \frac{N_j\left(\frac{h}{2}\right)-N_j(h)}{4^j-1}$ where $M - N_{j+1}(h) = O\left(h^{2(j+1)}\right)$

- [Richardson Extrapolation]

$$\begin{array}{llll}
O(h) & O(h^2) & O(h^3) & O(h^4) \\
\hline
N_1(h) \searrow & & & \\
\quad\ \to & & & \\
N_1(\frac{h}{2}) \searrow & N_2(h) \searrow & & \\
\quad\ \to & \quad\ \to & & \\
N_1(\frac{h}{4}) \searrow & N_2(\frac{h}{2}) \searrow & N_3(h) \searrow & \\
N_1(\frac{h}{8}) \to & N_2(\frac{h}{4}) \to & N_3(\frac{h}{2}) \to & N_4(h)
\end{array}$$

Integration

Definition

- [Degree of Precision] The degree of precision of a quadrature formula is the largest positive integer n s.t. the formula is exact for x^k for $k \in \{0,1,\dots,n\}$

Method of Quadrature

- [Quadrature] Select set of distinct nodes $x_0 < \cdots < x_n$ from interval $[a,b]$ and integrate the Lagrange interpolating polynomial $P_n(x) = \sum_{i=0}^{n} f(x_i)L_i(x)$
    - $\int_a^b f(x)\,\mathrm{d}x = \sum_{i=0}^{n} a_i f(x_i) + \frac{1}{(n+1)!}\int_a^b \prod_{i=0}^{n}(x-x_i)\, f^{(n+1)}\big(\xi(x)\big)\,\mathrm{d}x$ where $a_i = \int_a^b L_i(x)\,\mathrm{d}x$
- [Trapezoidal Rule] $x_0 = a, x_1 = b, h = b - a$
    - $\int_a^b f(x)\,\mathrm{d}x = \frac{h}{2}\big(f(x_0) + f(x_1)\big) - \frac{h^3}{12} f''(\xi)$
    - Degree of precision: 1
- [Simpson Rule] $x_0 = a,\ x_1 = \frac{a+b}{2},\ x_2 = b,\ h = \frac{b-a}{2}$
    - $\int_a^b f(x)\,\mathrm{d}x = \frac{h}{3}\big(f(x_0) + 4f(x_1) + f(x_2)\big) - \frac{h^5}{90} f^{(4)}(\xi)$
    - Degree of precision: 3
- [Closed Newton-Cotes Formula] $x_0 = a,\ x_n = b,\ h = \frac{b-a}{n}$
    - n even: $\int_a^b f(x)\,\mathrm{d}x = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!}\int_0^n t^2(t-1)\dots(t-n)\,\mathrm{d}t$
    - n odd: $\int_a^b f(x)\,\mathrm{d}x = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!}\int_0^n t(t-1)\dots(t-n)\,\mathrm{d}t$
- [Open Newton-Cotes Formula]

Composite Methods

- [Composite Simpson's Rule] Let $f \in C^4[a,b]$, n even, $h = \frac{b-a}{n}$ and $x_j = a + jh$. Then $\exists \xi \in (a,b)$ s.t.
    - $\int_a^b f(x)\,\mathrm{d}x = \frac{h}{3}\left(f(a) + 2\sum_{i=1}^{\frac{n}{2}-1} f(x_{2i}) + 4\sum_{i=1}^{\frac{n}{2}} f(x_{2i-1}) + f(b)\right) - \frac{b-a}{180} h^4 f^{(4)}(\xi)$
- [Composite Trapezoidal Rule]
    - $\int_a^b f(x)\,\mathrm{d}x = \frac{h}{2}\left(f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right) - \frac{b-a}{12} h^2 f''(\xi)$
- [Composite Midpoint Rule]
    - $\int_a^b f(x)\,\mathrm{d}x = 2h\sum_{i=0}^{\frac{n}{2}} f(x_{2i}) + \frac{b-a}{6} h^2 f''(\xi)$
- [Round Off Error Stability] Round off error does not depend on number of calculations performed i.e. independent of composite integration techniques and n

Romberg Integration

- $R_{n,k}$: $n+1$ number of points subdividing the interval, k
- $R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1}-1}\big(R_{k,j-1} - R_{k-1,j-1}\big)$ for $k = j, j+1, \dots$
- [Algorithm]
    - $R_{1,1} = \frac{h}{2}\big(f(a) + f(b)\big)$
    - $R_{n,1} = \frac{1}{2}\big(R_{n-1,1} + 2h_n \sum_{j=1}^{2^{n-2}} f(a + (k-0.5)h)\big)$ where $h_n = \frac{b-a}{2^{n-1}}$
        - $R_{n,1}$ is just dividing the interval into 2^{n-1} pieces and use composite method
    - $R_{n,i} = R_{n,i-1} + \frac{R_{n,i-1} - R_{1,i-1}}{4^{i-1}-1}$
        - Other Romberg terms come for free
        - For Simpson's, $R_{n,i} = I + O\big(h_i^{2i+2}\big)$

$$\begin{array}{cccc}
O(h_k^4) & O(h_k^6) & O(h_k^8) & O(h_k^{10}) \\ \hline
R_{1,1} \; \substack{\searrow \\ \rightarrow} & & & \\
R_{2,1} \; \substack{\searrow \\ \rightarrow} & R_{2,2} \; \substack{\searrow \\ \rightarrow} & & \\
R_{3,1} \searrow & R_{3,2} \searrow & R_{3,3} \searrow & \\
R_{4,1} \rightarrow & R_{4,2} \rightarrow & R_{4,3} \rightarrow & R_{4,4}
\end{array}$$