Lecture 8: Lagrange Interpolation

Solving via linear system.

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

Solving via basis polynomials. Let $L_k(x) = \frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$ for $k = 0, 1, \cdots, n$. Then $P_n(x) = y_0 L_0(x) + y_1 L_1(x) + \cdots + y_n L_n(x)$.

Error analysis. $\forall x \in [a,b], \exists \xi \in (\min\{x, x_0, x_1, \cdots, x_n\}, \max\{x, x_0, x_1, \cdots, x_n\})$ such that

$$f(x) = P_n(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

Tutorial 3: Lagrange Interpolation

Example 1. Construct the LIP for $f(x) = \log_2 x$ using $x_0 = 1/2, x_1 = 1, x_2 = 2, x_3 = 4$. Find a bound of the absolute error for any $x \in [1/2, +\infty)$.
Solution. $f(0.5) = -1, \quad f(1) = 0, \quad f(2) = 1, \quad f(4) = 2$. Solve using linear system to get $a_0 = -\frac{52}{21}$, $a_1 = \frac{7}{2}$, $a_2 = -\frac{7}{6}$, $a_3 = \frac{1}{7}$. So $P(x) = \frac{1}{7}x^3 - \frac{7}{6}x^2 + \frac{7}{2}x - \frac{52}{21}$. To get an u.b for error, first find $f^{(4)}(x) = -\frac{6}{x^4 \ln 2}$. Note monotonicity. Hence $\left|f^{(4)}(x)\right| \leqslant \frac{6}{(1/2)^4 \ln 2} = \frac{96}{\ln 2}$. Thus an u.b for absolute error $|P(x) - f(x)|$ is $\frac{1}{4!} \times \frac{96}{\ln 2}|(x-1/2)(x-1)(x-2)(x-4)| = \frac{4}{\ln 2}|(x-1/2)(x-1)(x-2)(x-4)|$. $\square$.

Result from Example 2. If nodes are equidistributed, the maximum value of $g(x) = |(x-x_0)(x-x_1)\cdots(x-x_N)|$ must be attained in (x_0, x_1) and (x_{N-1}, x_N) (due to the symmetry). $|g(x^*)| \leqslant \frac{1}{4}N!h^{N+1}$.

Error estimation for equidistributed nodes. $|P_N(x) - f(x)| \leqslant \frac{h^{N+1}}{4(N+1)} \max_{\xi \in [a,b]} \left|f^{(N+1)}(\xi)\right|$, for all $x \in [a,b]$

Exercise 2. Let $P_n(x)$ be the LIP for $f(x) = \cos x$ with $x_k = kh, \quad k = 0, 1, \cdots, n$ where $h = \pi/(2n)$. **1.** Find a positive integer N such that $|P_N(x) - f(x)| < 0.005, \quad$ for all $x \in [0, \pi/2]$.
Solution. For $f(x) = \cos(x)$, $\max_{\xi \in [0,\pi/2]} \left|f^{(N+1)}(\xi)\right| = 1$. Hence it suffices to find $\frac{h^{N+1}}{4(N+1)} < 0.005 \implies N \geq 3$.

Lecture 9: Divided Differences

How to find the Lagrange polynomial. $P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + \cdots + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$ where $a_k = f[x_0, x_1, \cdots, x_k]$ and $f[x_0, x_1, \cdots, x_n] = \frac{f[x_1, x_2, \cdots, x_n] - f[x_0, x_1, \cdots, x_{n-1}]}{x_n - x_0}$. $a_0 = f(x_0)$.

Lecture 10: Cubic Spline Interpolation (CSI)

How to find μ_k and λ_k. $\mu_k = \frac{x_k - x_{k-1}}{x_{k+1} - x_{k-1}}, \quad \lambda_k = \frac{x_{k+1} - x_k}{x_{k+1} - x_{k-1}}, \quad k = 1, 2, \cdots, n-1$

Natural Boundary Conditions. $M_0 = M_n = 0$.

$$\begin{bmatrix} 2 & \lambda_1 & & & & \\ \mu_2 & 2 & \lambda_2 & & & \\ & \mu_3 & 2 & \ddots & & \\ & & \ddots & \ddots & \lambda_{n-2} \\ & & & \mu_{n-1} & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{bmatrix} = \begin{bmatrix} 6f[x_0, x_1, x_2] \\ 6f[x_1, x_2, x_3] \\ 6f[x_2, x_3, x_4] \\ \vdots \\ 6f[x_{n-3}, x_{n-2}, x_{n-1}] \\ 6f[x_{n-2}, x_{n-1}, x_n] \end{bmatrix}$$

Clamped Boundary Conditions. $2M_0 + M_1 = 6f[x_0, x_0, x_1], \quad M_{n-1} + 2M_n = 6f[x_{n-1}, x_n, x_n]$.

$$\begin{bmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & \mu_n & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} 6f[x_0, x_0, x_1] \\ 6f[x_0, x_1, x_2] \\ \vdots \\ 6f[x_{n-2}, x_{n-1}, x_n] \\ 6f[x_{n-1}, x_n, x_n] \end{bmatrix}$$

How to find S_k.
$S_k(x) = M_{k-1}\frac{(x-x_k)^3}{6(x_{k-1}-x_k)} + M_k\frac{(x-x_{k-1})^3}{6(x_k-x_{k-1})} + A_k x + B_k$.
$A_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} - \frac{1}{6}(M_k - M_{k-1})(x_k - x_{k-1})$.
$B_k = \frac{x_k f(x_{k-1}) - x_{k-1} f(x_k)}{x_k - x_{k-1}} + \frac{1}{6}(M_k x_{k-1} - M_{k-1} x_k)(x_k - x_{k-1})$.

Piecewise Linear Interpolation (PLI). If $S(x) = S_k(x), \quad$ for $x \in [x_{k-1}, x_k], k = 1, 2, \cdots, n$ then $S_k(x) = f(x_{k-1})\frac{x - x_k}{x_{k-1} - x_k} + f(x_k)\frac{x - x_{k-1}}{x_k - x_{k-1}}$

Error analysis for PLI on equidistributed nodes. If $x_k = x_0 + kh$, then for $x \in [x_0, x_n]$, $|f(x) - S(x)| \leq \frac{1}{8}h^2 \max_{\xi \in [x_0, x_n]} |f''(\xi)|$

Tutorial 4: Divided Diff and CSI

Example 2: Quadratic spline interpolation. Given $n+1$ nodes $x_0 < x_1 < \cdots < x_{n-1} < x_n$ and a continuous function $f(x)$, find a function $S(x)$ such that **1.** $S(x)$ is first-order differentiable on (x_0, x_n) **2.** $S(x)$ is a quadratic polynomial on (x_{k-1}, x_k) for any $k = 2, 3, \cdots, n$; **3.** $S(x)$ is a linear function on (x_0, x_1). **4.** $S(x_k) = f(x_k)$ for all $k = 0, 1, \cdots, n$.
Solution. $S_1(x) = f(x_0)\frac{x - x_1}{x_0 - x_1} + f(x_1)\frac{x - x_0}{x_1 - x_0}$.
$S_k(x) = \frac{1}{2}M_k\frac{(x-x_{k-1})^2}{x_k - x_{k-1}} + \frac{1}{2}M_{k-1}\frac{(x-x_k)^2}{x_{k-1} - x_k} + C_k, \quad k = 2, 3, \cdots, n$.
$C_k = f(x_k) - \frac{1}{2}M_k(x_k - x_{k-1})$. $M_k = 2f[x_{k-1}, x_k] - M_{k-1}$. Since $M_1 = S_1'(x_1) = f[x_0, x_1]$, M_k for $k = 2, 3, \cdots, n$ can be obtained iteratively.

Lecture 11: Least Squares Approximation

Finding the coefficients. Let $X = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{pmatrix}$ and $\mathbf{a} = (a_0, a_1, \cdots, a_n)^T, \quad \mathbf{y} = (y_0, y_1, \cdots, y_m)^T$. Solve $X^T X a = X^T y$.

Proof for Exercise 1. TODO

Weighted LSA. $W = \operatorname{diag}\{w_0, w_1, \cdots, w_n\}$. Solve $X^T W X \mathbf{a} = X^T W \mathbf{y}$.

Lecture 12: Newton-Cotes Formulae

Ascertaining number of points for error $< \epsilon$. TODO

Trapezoidal Rule. $\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}[f(a) + f(b)]$.

Error for Trapezoidal Rule. $\int_a^b f(x)\mathrm{d}x = \frac{b-a}{2}[f(a) + f(b)] - \frac{1}{12}(b-a)^3 f''(\xi)$

Simpson's Rule. Given $f(a), f\left(\frac{a+b}{2}\right)$ and $f(b)$, $P(x) = f(a) + \frac{f(b)-f(a)}{b-a}(x-a) + \left[2f\left(\frac{a+b}{2}\right) - f(b) - f(a)\right]\frac{2(x-a)(x-b)}{(b-a)^2}$ whose integral is $\left[\frac{2}{3}f\left(\frac{a+b}{2}\right) + \frac{1}{6}f(b) + \frac{1}{6}f(a)\right](b-a)$.

Error for Simpson's Rule. $\int_a^b f(x)\mathrm{d}x = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] - \frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\xi)$

Theorem 3 on Page 4. TODO

Gaussian Elimination to find weights. TODO

Lecture 13: Composite Numerical Integration

Composite Trapezoidal Rule (CTR). Confusing use of h. Read carefully first

Error analysis for CTR. TODO

Composite Simpson's Rule (CSR). TODO

Error analysis for CSR. TODO

Tutorial 5: LSA and Integration

TODO

Miscellaneous

The Gamma function. $\Gamma(x) = \int_0^{+\infty} t^{x-1}\mathrm{e}^{-t}\mathrm{d}t$ and $\Gamma(n) = (n-1)!$ for any positive integer n.

Linear Algebra and Calculus. TODO

General formulae for higher derivatives. Higher order derivatives

Error proofs from lectures/tutorials. LSA proof of optimality, the proof of errors, tutorial identities

Finding the set of x and y values for the lowest error LIP. First find the lowest-error polynomial (LEP), which is $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m$. From Homework Q3,

$$M = \begin{pmatrix} b-a & \frac{b^2-a^2}{2} & \cdots & \frac{b^{m+1}-a^{m+1}}{m+1} \\ \frac{b^2-a^2}{2} & \frac{b^3-a^3}{3} & \cdots & \frac{b^{m+2}-a^{m+2}}{m+2} \\ \vdots & \vdots & & \vdots \\ \frac{b^{m+1}-a^{m+1}}{m+1} & \frac{b^{m+2}-a^{m+2}}{m+2} & \cdots & \frac{b^{2m+1}-a^{2m+1}}{2m+1} \end{pmatrix}$$ and

b $= \left(\int_a^b x^0 f(x)dx, \cdots, \int_a^b x^m f(x)dx\right)^T$. Solve $Ma = b$ to get the coefficients of the LEP. Then reverse-engineer the Gaussian Elimination process for finding a when constructing the LIP to get x and y (possibly non-unique).