~~the interpolation interval.~~

Range phenomenon: polynomial - wiggle near the ends of interpolation.

Multivariate Newton's method: Algorithm

$\underline{x}_0$ = initial vector

solve $D_F(\underline{x}_k)\underline{s} = -F(\underline{x}_k)$

$\underline{x}_{k+1} = \underline{x}_k + \underline{s}$ for $k = 0, 1, 2, \ldots$

Jacobian matrix

- 3 variables: $\underline{x} = (u, v, w)$
- 3 functions: f_1, f_2, f_3

$$D_F(\underline{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial w} \\ \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial w} \\ \frac{\partial f_3}{\partial u} & \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial w} \end{bmatrix}$$

Interpolation, extrapolation, approximation

- Interpolation
  - The new value $x \neq x_i$ is inside the range of the interpolation points $x_0, x_1, \ldots, x_n$.
- Extrapolation
  - The new value z is outside this range.
- Approximation
  - Some norm $\|v - y\|$ of the difference of the vectors $\underline{v} = [v(x_0), \ldots, v(x_n)]$ and $\underline{y} = [y_0, \ldots, y_n]$ is minimized.

$(\sin x)' = \cos x$

$(\cos x)' = -\sin x$

$(\tan x)' = \sec^2 x$

$(\ln x)' = \frac{1}{x}$

$(a^x)' = a^x \ln a$

$\left(\frac{f(x)}{g(x)}\right)' = \frac{g(x)f'(x) - f(x)g'(x)}{g^2(x)}$

Lagrange interpolation (n = 3)

Given n data points $(x_1, y_1), \ldots, (x_n, y_n)$, the polynomial of degree $d = n-1$ that interpolates the points is:

$$P_2(x) = y_1\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

Theorem. Let $(x_1, y_1), \ldots, (x_n, y_n)$ be n points in a plane with distinct x_i. Then there exists **one and only one** polynomial P of degree $n-1$ or less that satisfies $P(x_i) = y_i$ for $i = 1, \ldots, n$.

Interpolation error

Assume we have $y_i = f(x_i)$, $i = 0, 1, \ldots, n$ and an interpolating polynomial $P(x)$. The interpolation error at x is

$$f(x) - P(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{n!} f^{(n)}(c)$$

c lies between the smallest and largest of x_i.

Normal equations for least squares

Given an inconsistent system

$$\underset{m*n}{A}\underline{x} = \underset{m*1}{\underline{b}},$$

solve

$$\underset{n*n}{(A^T A)}\underline{\tilde{x}} = \underset{n*1}{A^T \underline{b}}$$

for the least squares solution $\underline{\tilde{x}}$ that minimizes the Euclidean length of the residual $\underline{r} = \underline{b} - A\underline{\tilde{x}}$.

Newton's divided differences

0-th divided difference

$f[x_i] = P(x_i) = y_i$ $\qquad c_0 = y_1$

1-st divided difference

$f[x_i\ x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$ $\qquad c_1 = \frac{y_2 - y_1}{x_2 - x_1}$

k-th divided difference

$f[x_i \cdots x_{i+k}] = \frac{f[x_{i+1} \cdots x_{i+k}] - f[x_i \cdots x_{i+k-1}]}{x_{i+k} - x_i}$

The divided differences are best arranged in a triangular array:

x_1	$f[x_1]$ y_1	$\frac{f[x_2]-f[x_1]}{x_2-x_1}$	$\frac{f[x_2\ x_3]-f[x_1\ x_2]}{x_3-x_1}$	$\frac{f[x_2\ x_3\ x_4]-f[x_1\ x_2\ x_3]}{x_4-x_1}$
x_2	$f[x_2]$ y_2	$\frac{f[x_3]-f[x_2]}{x_3-x_2}$	$\frac{f[x_3\ x_4]-f[x_2\ x_3]}{x_4-x_2}$	
x_3	$f[x_3]$ y_3	$\frac{f[x_4]-f[x_3]}{x_4-x_3}$		
x_4	$f[x_4]$ y_4			

$P(x) = f[x_1] + f[x_1\ x_2](x - x_1)$
$+ f[x_1\ x_2\ x_3](x-x_1)(x-x_2)$
$+ f[x_1\ x_2\ x_3\ x_4](x-x_1)(x-x_2)(x-x_3)$
$+ f[x_1 \cdots x_n](x-x_1)(x-x_2)\cdots(x-x_n)$

Euclidean length (2-norm) $\rightarrow \|\underline{r}\|_2 = \sqrt{r_1^2 + \cdots + r_m^2}$

Squared error

$$\mathrm{SE} = r_1^2 + \cdots + r_m^2$$

Root mean squared error

$$\mathrm{RMSE} = \sqrt{\mathrm{SE}/m} = \sqrt{(r_1^2 + \cdots + r_m^2)/m}$$

$$\text{error magnification factor} = \frac{\text{relative forward error}}{\text{relative backward error}} = \mathrm{cond}(A) = \|A\| \times \|A^{-1}\|$$

Remedy: avoid forming $A^T A$

models.

Linear: $y = c_1 + c_2 x$

Parabola: $y = c_1 + c_2 x + c_3 x^2$

$y = c_1 + c_2 \cos 2\pi t + c_3 \sin 2\pi t$

- Exponential model: $y = c_1 e^{c_2 t}$
- Applying the natural logarithm!

$\ln y = \ln(c_1 e^{c_2 t})$

$\ln y = \ln c_1 + \ln e^{c_2 t} = \ln c_1 + c_2 t$

Let $k = \ln c_1$

$\ln y = k + c_2 t$

QR-fac

Given an m x n inconsistent system $A\underline{x} = \underline{b}$

- $A\underline{x} = \underline{b}$
- $QR\underline{x} = \underline{b}$
- $Q^{-1}QR\underline{x} = Q^{-1}\underline{b} = Q^T\underline{b}$
- $R\underline{x} = Q^T\underline{b}$ directly using back substitution!

Gram-Schmidt

1st unit vector: $y_1 = A_1$ and $q_1 = \frac{y_1}{\|y_1\|_2}$.

2nd unit vector:

$$y_2 = A_2 - q_1(q_1^T A_2), \text{ and } q_2 = \frac{y_2}{\|y_2\|_2}.$$

j-th unit vector:

$$y_j = A_j - q_1(q_1^T A_j) - q_2(q_2^T A_j) - \ldots - q_{j-1}(q_{j-1}^T A_j) \text{ and } q_j = \frac{y_j}{\|y_j\|_2}.$$

$$\underbrace{(D_r(\underline{x}_k))^T}_{n \times m} \underbrace{D_r(\underline{x}_k)}_{m \times n} \underbrace{\underline{v}_k}_{n \times 1} = \underbrace{-D_r(\underline{x}_k)^T}_{n \times m} \underbrace{r(\underline{x}_k)}_{m \times 1}$$

Gauss Newton

$\underline{x}_0$ = initial vector $[x_0, y_0]^T$ $\qquad \underline{x}_k = [x, y]^T$

solve $(D_r(\underline{x}_k)^T D_r(\underline{x}_k))\underline{v} = -D_r(\underline{x}_k)^T r(\underline{x}_k)$

$\underline{x}_{k+1} = \underline{x}_k + \underline{v}$ for $k = 0, 1, 2, \ldots$

$r_1(x,y)$
$r_2(x,y)$
$r_3(x,y)$
$r(x,y) = [r_1\ r_2\ r_3]^T$

$$D_r(x,y) = \begin{bmatrix} \partial r_1/x & \partial r_1/y \\ \partial r_2/x & \partial r_2/y \\ \partial r_3/x & \partial r_3/y \end{bmatrix} = \begin{bmatrix} \frac{x-x_1}{S_1} & \frac{y-y_1}{S_1} \\ \frac{x-x_2}{S_2} & \frac{y-y_2}{S_2} \\ \frac{x-x_3}{S_3} & \frac{y-y_3}{S_3} \end{bmatrix}$$

$r_1(x,y) = \sqrt{(x-x_1)^2 + (y-y_1)^2} - R_1$

$\frac{\partial r_1}{x} = \frac{1}{2}((x-x_1)^2 + (y-y_1)^2)^{-\frac{1}{2}}(2x - 2x_1)$

$= \frac{x - x_1}{S_1} \qquad S_1 = \sqrt{(x-x_1)^2 + (y-y_1)^2}$

Gauss-Newton

$\underline{x}_0$ = initial vector

$$\underline{x}_{k+1} = \underline{x}_k \underbrace{- (D_r(\underline{x}_k)^T D_r(\underline{x}_k))^{-1} D_r(\underline{x}_k)^T r(\underline{x}_k)}_{\underline{v}_k} \text{ for } k = 0, 1, 2, \ldots$$

$\underline{x}_{k+1} = \underline{x}_k + \underline{v}_k$