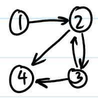
CS 370 Final Definition, Theorem and Examples

FPNS (t, β, L, U) $\pm 0.d_1 d_2 \ldots d_t \times \beta^p$ where $d_1 \neq 0$, $d_i \in \{0, \ldots, \beta - 1\}$. IEEE Single Precision: $1, 8, 23, -126, 127$. IEEE Double Precision: $1, 11, 52, -1022, 1023$.

AbsErr $= |x - \overline{x}|$. **RelErr** $= \frac{|x - \overline{x}|}{x}$. MaxRelErr: **Machine Epsilon**. The smallest number that when added to 1 becomes the next number $\beta/2 \cdot \beta^{-t}$. $\frac{\mu(x) - x}{x} = \delta$ where $|\delta| \leq E$. $\mu(x) = (1 + \delta)x$. If not in range (overflow | underflow)

$P' = P + \frac{1}{R} e d^T$. **Markov Transition Matrix** $M = \alpha P' + (1 - \alpha) \frac{1}{R} e e^T$. One click: $p^{n+1} = M p^n$. $p = Mp$. $(I - M)p = 0$.



$P' = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} \\ 1 & 0 & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{2} & 0 & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$

Eigenvector x: $Qx = \lambda x$. Since $p = Mp$, hence has eigenvector 1.

Solving upper triangular matrix: back substitution. Solving lower triangular matrix: forward substitution. Gaussian Elimination.

LU factorization: any square matrix A can be factored into a product of an upper triangular and lower triangular matrices such that $LU = PA$.

LU factorization Applications

1. Solving $Ax = b$. $Ax = b \Rightarrow PAx = Pb \Rightarrow LUx = Pb$. Solve $Lz = Pb$. Solve $Ux = z$. LU factorization $O(N^3)$ + forward sub + back sub $O(N^2)$.
2. Solving $AX = B$. LU factorization $O(N^3)$. Solve $LUx_i = Pb_i$ which is $O(N^2)$ each.

LU factorization Example: **note** put the one that has greatest absolute value on top.

$$\begin{bmatrix} 1 & 1 & 3 & 0 \\ 2 & 1 & -1 & 1 \\ -1 & 2 & 3 & -1 \\ 3 & -1 & -1 & 2 \end{bmatrix} \xrightarrow{\text{Swap } 1,4} \begin{bmatrix} 3 & -1 & -1 & 2 \\ 2 & 1 & -1 & 1 \\ -1 & 2 & 3 & -1 \\ 1 & 1 & 3 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & -1 & -1 & 2 \\ 0 & \frac{5}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{5}{3} & \frac{8}{3} & -\frac{1}{3} \\ 0 & \frac{4}{3} & \frac{10}{3} & -\frac{2}{3} \end{bmatrix} \rightarrow \begin{bmatrix} 3 & -1 & -1 & 2 \\ 0 & \frac{5}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 3 & 0 \\ 0 & 0 & \frac{18}{5} & -\frac{2}{5} \end{bmatrix} \xrightarrow{\text{Swap } 3,4}$$

$$\begin{bmatrix} 3 & -1 & -1 & 2 \\ 0 & \frac{5}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & \frac{18}{5} & -\frac{2}{5} \\ 0 & 0 & 3 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & -1 & -1 & 2 \\ 0 & \frac{5}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & \frac{18}{5} & -\frac{2}{5} \\ 0 & 0 & 0 & \frac{1}{3} \end{bmatrix}, L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{2}{3} & 1 & 0 & 0 \\ \frac{1}{3} & \frac{4}{5} & 1 & 0 \\ -\frac{1}{3} & 1 & \frac{5}{6} & 1 \end{bmatrix}, U = \begin{bmatrix} 3 & -1 & -1 & 2 \\ 0 & \frac{5}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & \frac{18}{5} & -\frac{2}{5} \\ 0 & 0 & 0 & \frac{1}{3} \end{bmatrix}, P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Singular Value Decomposition: $A = U \Sigma V$. U, V are unitary matrices. Unitary matrix: $M^T = M^{-1}$.

SVD Applications

1. The rank of A is the number of nonzero singular values
2. Lower-Rank Approximation using the SVD. If you want a rank-k approximation to A, simply set all singular values to zero except for the largest k. (used for image compression)
3. query matching

Numerical Solution of ODEs. $\frac{dy(t)}{dt} = f(t, y)$, $y(t_0) = y^{(i)}$.

Euler's Method. Starting with the initial state, we can approximate the solution by taking small steps in time. $y_{k+1} = y_k + h_k f(t_k, y_k)$. Local error $l_{n+1} = |\hat{y_n}(x_{n+1}) - y_{n+1}|$. **Global error** $\Sigma_{n+1} = |\hat{y_0}(x_{n+1}) - y_{n+1}|$. The **local error** is $O(h^2)$, the global error is $O(h)$.

Euler's Method Golf Example: $(x(0), y(0)) = (0, 0)$. Golfer hit the ball with initial velocity vector (v_x, v_y). Dynamics Model: $\frac{dx(t)}{dt} = v_x$, $\frac{d^2 y(t)}{dt} = -g$. Let $z_1 = x$, $z_2 = y$, $z_3 = \frac{dy}{dt} = \frac{dz_2}{dt}$. Then

$$\frac{d}{dt} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = f(t, z) = \begin{bmatrix} v_x \\ z_3 \\ -g \end{bmatrix} \quad \& \quad z_0 = \begin{bmatrix} 0 \\ 0 \\ v_y \end{bmatrix}$$

Let $v_x = 30, v_y = 12, g = 9.81$, $h = 0.1$. Then

$$z^{(1)} = z^{(0)} + hf(t_0, z^{(0)}) = \begin{bmatrix} 0 \\ 0 \\ 12 \end{bmatrix} + 0.1 \begin{bmatrix} 30 \\ 12 \\ -9.81 \end{bmatrix} = \begin{bmatrix} 3 \\ 1.2 \\ 11.019 \end{bmatrix}$$

$$z^{(2)} = z^{(1)} + hf(t_1, z^{(1)}) = \begin{bmatrix} 3 \\ 1.2 \\ 11.019 \end{bmatrix} + 0.1 \begin{bmatrix} 30 \\ 11.019 \\ -9.81 \end{bmatrix} = \begin{bmatrix} 6 \\ 2.3019 \\ 10.038 \end{bmatrix}$$

Modified Euler Method (improved euler OR 2nd-order Runge-kutta)

1. Start with an euler step. $y^E_{n+1} = y_n + h_n f(t_n, y_n)$. Let $\overline{f_1} = f(t_n, y_n)$.
2. $\overline{f_2} = f(t_{n+1}, y^E_{n+1})$.
3. $y^M_{n+1} = y_n + h_n \frac{\overline{f_1 + f_2}}{2}$.

The local error is $O(h^3)$. Global error is $O(h^2)$. Use taylor expansion to **calculate local error**.

Given n data points (x_i, y_i), $i = 1, \ldots, n$ with $x_i \neq x_j$ if $i \neq j$, $\exists!$ polynomial of degree at most $n-1$, $p(x) = c_1 + c_2 x + \ldots + c_n x^{n-1}$ such that $p(x_i) = y_i$.

Monomial Form: Vandermonde System: n equations, n unknowns. **Lagrange Form**: $L_i(x) = \dfrac{\ldots(x - x_{i-1})(x - x_{i+1})\ldots}{\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots}$. $p(x) = \sum_{i=1}^{n} L_i(x) y_i$.

Monomial drawback: solve a linear system, matrix entries gets larger as n gets bigger, difficult to invert as you include more points, espicially if two points have similar x-values. Difficult to solve accurately.

Piecewise (polynomial, linear) interpolation; **Cubic spline interpolation**: $S(x)$ is called a cubic spline if 1). $S(x)$ is an interpolant. 2). $S(x)$ is piecewise cubic. 3). $S(x)$ is twice differentiable.

There are $4n-4$ unknowns for all cubic pieces. There are $2n-2$ equations for interpolant, $n-2$ for continuous of $S'(x)$, $n-2$ for continuous of $S''(x)$. The last 2 equations cam be 1). Clamped spline $S'(x_1)$ and $S'(x_n)$ are specified. 2). Natural spline $S''(x_1) = S''(x_n) = 0$. 3). Periodic spline $S'(x_1) = S'(x_n)$, $S''(x_1) = S''(x_n)$ (assumes $y_1 = y_n$). 4). Combinations

Use $p_i(x) = a_{i-1}\dfrac{(x_{i+1} - x)^3}{6h_i} + a_i \dfrac{(x - x_i)^3}{6h_i} + b_i(x_{i+1} - x) + c_i(x - x_{i-1})$. Interpolant constant gives $b_i = \frac{y_i}{h_i} - a_{i-1}\frac{h_i}{6}$, $c_i = \frac{y_{i+1}}{h_i} - a_i \frac{h_i}{6}$. $S''(x)$ is continuous by design.

$$p_i'(x_{i+1}) = p_{i+1}'(x_{i+1}) \Rightarrow a_{i-1}\frac{h_i}{6} + a_i \frac{h_i + h_{i+1}}{3} + a_{i+1}\frac{h_{i+1}}{6} = \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i}$$

$$\begin{bmatrix} \frac{h_1}{3} & \frac{h_1}{6} & \cdots & & & \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{h_i}{6} & \frac{h_i+h_{i+1}}{3} & \frac{h_{i+1}}{6} & \\ & & & \ddots & \ddots & \ddots \\ & & & & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_i \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{y_2 - y_1}{h_1} - v_1 \\ \frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1} \\ \vdots \\ \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \\ \vdots \\ v_2 - \frac{y_n - y_{n-1}}{h_{n-1}} \end{bmatrix}$$ where v_1, v_2 are slope at end points

Bezier curve 1). passes through $(x_0, y_0), (x_N, y_N)$, 2). tangent to $point_1 \to point_2$, 3). tangent to $point_{N-1} \to point_N$

Fourier series $f(x) = a_0 + \sum_{k=1}^{\infty}\left[a_k \cos(\frac{2\pi k x}{N}) + b_k \sin(\frac{2\pi k x}{N})\right] = \sum_{k=-\infty}^{\infty} C_k \left(\cos(\frac{2\pi k x}{N}) + i \sin(\frac{2\pi k x}{N})\right)$, with $C_k = \overline{C_{-k}}$.

Complex Inner Product: $\langle a\,,\, b\,\rangle = \sum_{n=1}^{N} a_n \overline{b_n} = \overline{\langle b\,,\, a\,\rangle}$.

$W_N^k = (e^{2\pi i/N})^k = e^{2\pi i k/N}$. $f_n = \frac{1}{N}\sum_{k=0}^{N-1} F_k W_N^{nk} = \frac{1}{N}\langle \vec{F}\,,\, \overline{W_N(n)}\,\rangle$, $F_k = \sum_{n=0}^{N-1} f_n \overline{W_N^{nk}} = \langle \vec{F}\,,\, W_N(k)\,\rangle$.

Discrete Fourier transform $M = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \overline{W}^1 & \overline{W}^2 & \cdots & \overline{W}^{N-1} \\ \vdots & & & & \\ 1 & \overline{W}^{N-1} & \overline{W}^{(N-1)2} & \cdots & \overline{W}^{(N-1)(N-1)} \end{bmatrix}$, $\vec{F} = M\vec{f}$, $\vec{f} = \frac{1}{N}\overline{M}\vec{F}$. $M^{-1} = \frac{1}{N}\overline{M}$, where $\overline{M}$ is the matrix formed by conjugating everything in M.

$$\begin{bmatrix} -1 \\ 2 \\ 4 \\ 3 \\ 1 \\ 3 \\ 4 \\ 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 \\ 5 \\ 8 \\ 5 \\ \hline -2(\overline{W_8^0}) \\ -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}i(\overline{W_8^1}) \\ 0(\overline{W_8^2}) \\ -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}i(\overline{W_8^3}) \end{bmatrix} \Rightarrow \begin{bmatrix} 8 \\ 10 \\ \hline -8 \\ 0 \\ \hline -2 \\ -\sqrt{2} \\ \hline -2 \\ \sqrt{2} \end{bmatrix} \Rightarrow \begin{bmatrix} 18 \\ \hline -2 \\ \hline -8 \\ \hline -8 \\ \hline -2-\sqrt{2} \\ \hline -2+\sqrt{2} \\ \hline -2+\sqrt{2} \\ \hline -2-\sqrt{2} \end{bmatrix} \Rightarrow \begin{bmatrix} 18 \\ -2 \\ \hline -8 \\ -8 \\ \hline -2-\sqrt{2} \\ -2+\sqrt{2} \\ \hline -2+\sqrt{2} \\ -2-\sqrt{2} \end{bmatrix} \Rightarrow \begin{bmatrix} 18 \\ -8 \\ -2 \\ -8 \\ \hline -2-\sqrt{2} \\ -2+\sqrt{2} \\ -2+\sqrt{2} \\ -2-\sqrt{2} \end{bmatrix} \Rightarrow \begin{bmatrix} 18 \\ -2-\sqrt{2} \\ -8 \\ -2+\sqrt{2} \\ -2 \\ -2+\sqrt{2} \\ -8 \\ -2-\sqrt{2} \end{bmatrix}$$

Least square $y = A\beta + r$, $E(\beta) = r^T r = ||\, r\,||^2$, differentiate and set to zero gives $A^T A \beta = A^T y$, to solve, multiply by $(A^T A)^{-1}$, $\beta = (A^T A)^{-1} A^T$.