Lecture September 11

Eigenvalue problems

$A \in \mathbb{R}^{(n\times n)}$ $\quad (\mathbb{C}^{(n\times n)})$

$x \in \mathbb{R}^n$

$$Ax = \lambda x$$

standard approach:

orthogonal matrix $S \in \mathbb{R}^{n\times n}$

$$S S^T = S^T S = \mathbb{1}$$

$S = [s_1 \, s_2 \, \dots \, s_n]$ orthogonal column vectors

$$\langle s_i, s_j \rangle = \delta_{ij}$$

$$s_i = [s_{1i}, s_{2i}, s_{3i}, \dots s_{ni}]^T$$

$$SAS^T = D = [\lambda_1 \, \lambda_2, \dots \lambda_n]$$

$$S \cdot \quad Ax = \lambda x$$

$$SAx = \lambda Sx$$

$$I = S^T S$$

$$S = S^T$$
$$SS^T = I$$

$$SAS^T(Sx) = \lambda(Sx)$$

Similarity/orthogonal (unitary) transformations preserve the eigenvalues but change the eigenvectors.

$$x \Rightarrow Sx = y$$

$$\langle x_i, x_j \rangle = \delta_{ij} = \underline{x_i^T x_j}$$

$$x_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \cdots \quad x_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$x_1^T x_2 = 0 \qquad x_1^T x_1 = \underline{1}$$

These transformations (project 2a)

$$y_i = S x_i$$

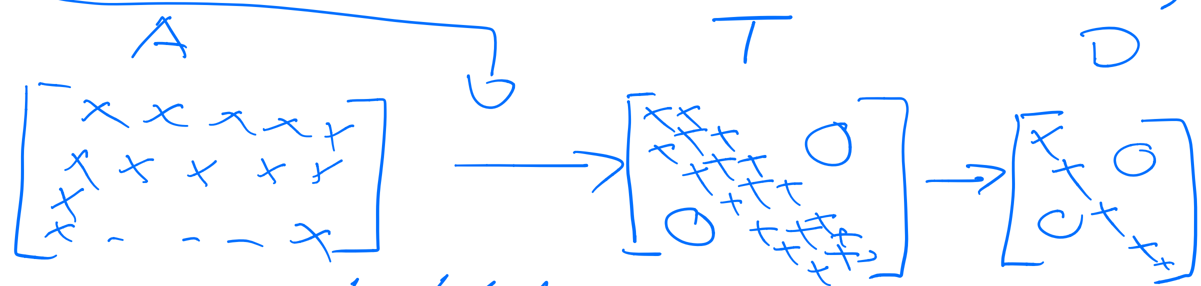
$$\langle y_i, y_j \rangle = \langle S x_i, S x_j \rangle$$
$$= x_i^T \underbrace{S^T S}_{I} x_j = x_i^T x_j = \delta_{ij}$$

orthogonality and norm are preserved.

When we want eigenvalues we need repeated operations

$$S_M \cdots S_3 S_2 S_1 A S_1^T S_2^T S_3^T \cdots S_M^T = D$$

Two-major steps (A is symmetric)



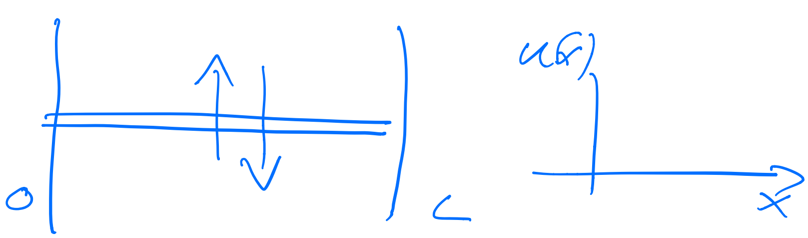
Householder's Transformations S_H

Project 2

$$\gamma \frac{d^2 u(x)}{dx^2} = -\underline{F}\, u(x)$$

Two-point boundary value problem

$u(0) = u(L) = 0 \quad x \in [0, L]$



Discretized and scaled equations $\xi = \alpha \cdot x$

$\uparrow$ Dimensionless

$\alpha = \frac{1}{L}$

natural length scale.

$$\frac{d^2 u(\xi)}{d\xi^2} = -\lambda\, u(\xi) \quad \Big| \quad \lambda = \frac{F L^2}{\gamma}$$

$$\Rightarrow \quad T \cdot u = \lambda u$$

$$u \in [u_1, u_2 \ldots u_{m-1}]$$

$$\xi_i = \xi_0 + i \cdot h \qquad i = 0, 1, 2, \ldots, n$$

$$T = (a, d, a)$$

$$d = 2/h^2 \qquad a = -1/h^2$$

$$T = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & & & 0 \\ -1 & 2 & \ddots & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ 0 & & & -1 & 2 \end{bmatrix}$$

Jacobi's algorithm:

$A \in \mathbb{R}^{2\times 2}$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \rightarrow \begin{bmatrix} d_{11} & 0 \\ 0 & d_{22} \end{bmatrix}$$

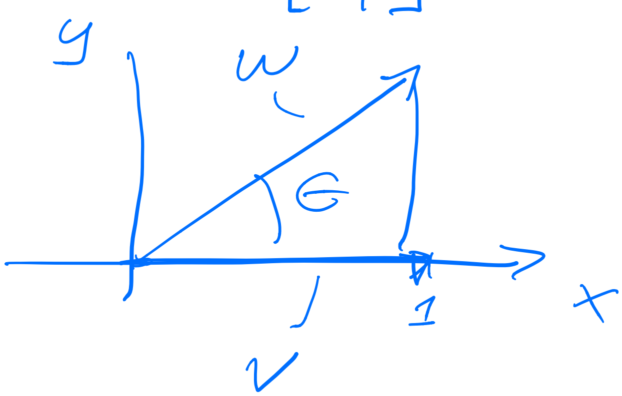
a_{21}

in 2-dims Jacobi's method is a Rotation along an axis

$$S = \begin{array}{c} \begin{matrix} C & S \end{matrix} \\ \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \\ \begin{matrix} S & C \end{matrix} \end{array}$$

$$S \cdot S^T = I$$

$$V = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \qquad w = Sv = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$



$x_1 = 1$

$y_1 = 0$

$$w = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\Rightarrow \quad x_2 = c \cdot x_1 - s y_1$$

$$y_2 = s x_1 + c y_1$$

$$\begin{bmatrix} d_{11} & 0 \\ 0 & d_{22} \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} a_{11} & \underline{a_{12}} \\ \underline{a_{12}} & a_{22} \end{bmatrix}$$

$$S A S^T$$

$$\times \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

$$\underline{d_{11}} = \underline{a_{11}\boxed{c}^2} - 2 a_{12} \boxed{cs} + a_{22} \boxed{s}^2$$

$$\underline{d_{22}} = a_{22}\boxed{c}^2 + a_{12} \boxed{cs} \cdot 2 + a_{11} \boxed{s}^2$$

$$\boxed{d_{12} = d_{21}} = 0 = (a_{11} - a_{22}) c s$$

$$+ a_{12} (c^2 - s^2)$$

$$\left(\frac{a_{11} - a_{22}}{\underline{a_{12}}} \right) \frac{cs}{c^2} + \frac{c^2}{c^2} - \frac{s^2}{s^2} = 0$$

$$\tan\theta = t = \frac{\sin\theta}{\cos\theta} = \frac{s}{c}$$

$$\tau = \frac{\underline{a_{22}} - \underline{a_{11}}}{2 \underline{a_{12}}}$$

$$t^2 + 2\tau t - 1 = 0 \Rightarrow$$

$$\boxed{\underline{t} = -\tau \pm \sqrt{1 + \tau^2}}$$

$$c = \frac{1}{\sqrt{1+t^2}}$$

$$s = t \cdot c$$

Need to implement

For an $n \times n$ matrix A

$$S = \begin{array}{c} \\ \\ \\ j \\ l \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccc} & & i & k & & \end{array} \\ \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & & & & \\ 0 & \cdots & \boxed{\cos\theta} & \boxed{\sin\theta} & & \\ & & \boxed{-\sin\theta} & \boxed{\cos\theta} & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} \end{array}$$

How do we find $\underline{a_{jk}}$ and a_{li}?

The most efficient way is to rotate along the largest non-diagonal matrix element for each rotation.

$\Rightarrow$ need an algorithm which finds the largest

a_{jk}

$\mathbb{R}^{n \times n}$

$S_M S_{M-1} \ldots S_1 A S_1^T \ldots S_{M-1}^T S_M^T$

only one set of non-diagonal elements that are transformed.