Summary : QR algorithm

A tridiagonal [through Householder or Givens]

$A^{(0)} = A$, for $k = 0, 1, 2, \ldots$

$A^{(k)} - \mu^k I = Q^{(k)} R^{(k)}$ QR factorization

$A^{(k+1)} = R^{(k)} Q^{(k)} + \mu^k I$ multiply in reverse order

end

[shifted version, $\mu^k \in \mathbb{R}$, approximation to eigenvalue]

$$A^{(k+1)} = R^{(k)} Q^{(k)} = Q^{(k)T} A^{(k)} Q^{(k)}$$

so, $A^{(k)}$ and $A^{(k+1)}$ have same eigenvalues since $Q^{(k)}$ orthogonal.

$$= \underbrace{Q^{(k)T} Q^{(k-1)T} \cdots Q^{(1)T}}_{\bar{Q}^{(k)T}} A \underbrace{Q^{(1)} Q^{(2)} \cdots Q^{(k)}}_{\bar{Q}^{(k)}} \xrightarrow{k \to \infty} \text{diagonal matrix}$$

$\longrightarrow$ we find the eigenvalues. How about the eigenvectors?

Start with $B \in \mathbb{R}^{n \times n}$, $B = B^T$

$$\xrightarrow[\text{Householder}]{\text{step 1}} P_n^T B P_n = A$$

orthogonal tridiagonal

$$\xrightarrow[\text{QR-algo}]{\text{step 2}} \bar{Q}^{(k)T} A \bar{Q}^{(k)} = D$$

diagonal

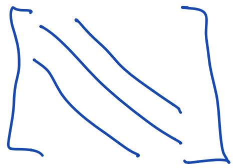
$$\xrightarrow[\text{together}]{\text{putting}} \bar{Q}^{(k)T} P_n^T B P_n \bar{Q}^{(k)} = D$$

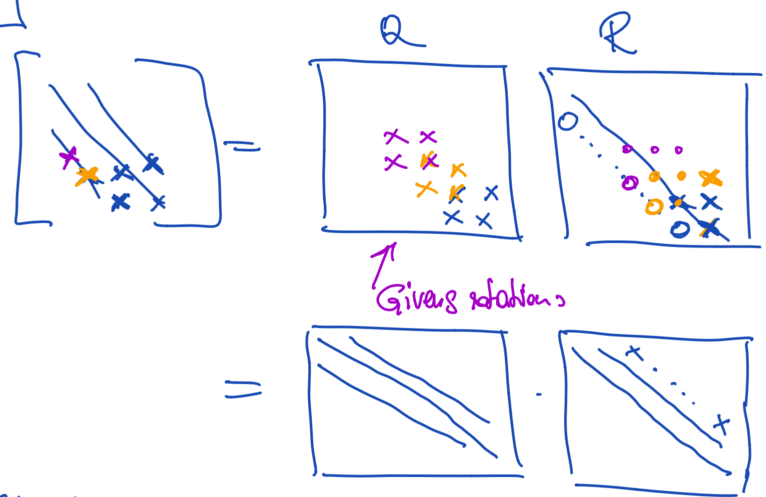
$$\Longrightarrow B\left(P_n\bar{Q}^{(k)}\right) = \left(P_n\bar{Q}^{(k)}\right)D$$

$$B \cdot \begin{bmatrix} | & | & \cdots & | \end{bmatrix} = \begin{bmatrix} | & | & \cdots & | \end{bmatrix} \begin{bmatrix} \times & & \\ & \ddots & \\ & & \times \end{bmatrix}$$

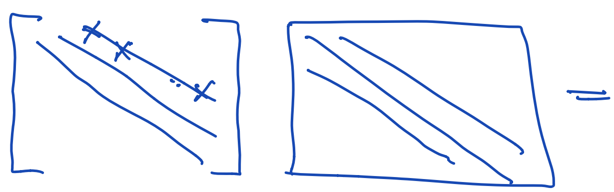
eigenvectors are columns of $P_n\bar{Q}^{(k)}$!

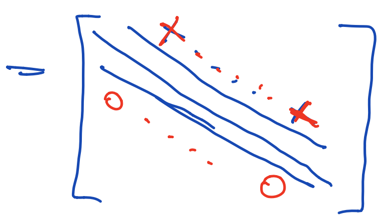
Remark: QR algorithm always only works with tridiagonal matrices:

$A =$  tridiagonal

QR algo: $A^{(k)} =$ 

$$A^{(k+1)} = Q^{(k)T} A^{(k)} Q^{(k)} = R^{(k)} Q^{(k)} =$$

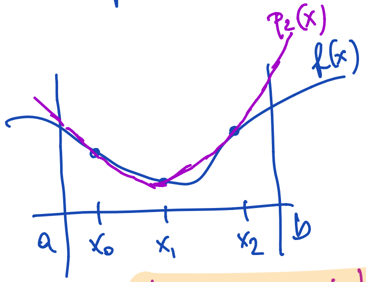




Since $A^{(k+1)}$ is symmetric, 'x' must be zero, and thus $A^{(k+1)}$ is tridiagonal.

§ 6.2 Polynomial Interpolation

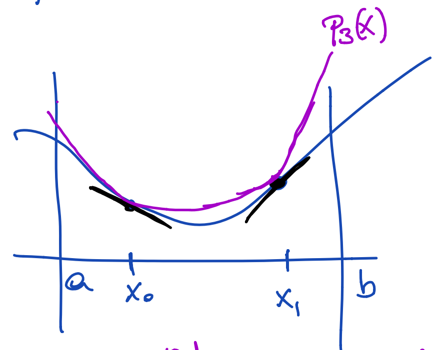
Approximate functions using polynomials that coincide with function (and derivatives) at node points.



$P_2(x)$ ... quadratic

$P_2(x_i) = f(x_i) \quad i = 0,1,2$

Lagrange interpolation
(only uses function values)



$P_3(x)$ ... 3rd order polynomial

$P_3(x_i) = f(x_i) \quad i = 0,1$

$P_3'(x_i) = f'(x_i) \quad i = 0,1$

Hermite Interpolation
(function values + derivatives)

<u>Interpolation problem:</u>

$n \geq 1$, $x_0, \dots, x_n$ distinct (i.e. $x_i \neq x_j$ for $i \neq j$) $y_0, \dots, y_n \in \mathbb{R}$

Find polynomial $p_n \in P_n$ such that $p_n(x_i) = y_i \quad i = 0, \dots, n$

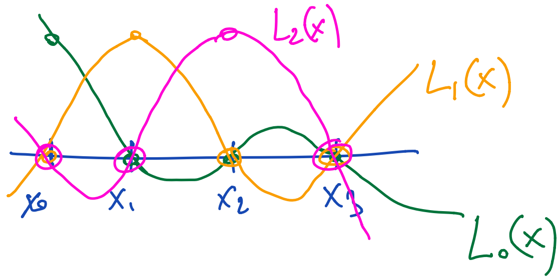
P_n ... space of polynomials of degree $\leq n$, i.e.

$$P_n = \left\{ a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \,\middle|\, a_n, \dots, a_0 \in \mathbb{R} \right\}$$

<u>Lemma:</u> $n \geq 1$. There exists polynomials $L_k \in P_n$ $k = 0, 1, \dots, n$ such that

$$L_k(x_i) = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases} \quad \text{for all } i, k$$

Moreover, $p_n(x) = \sum_{k=0}^{n} L_k(x) y_k$ is the interpolating polynomial.



Proof:

$$L_k(x) = C_k \prod_{\substack{i=0 \\ i \neq k}}^{n} (x - x_i)$$

$C_k \in \mathbb{R}$

$$L_k(x_k) = C_k \prod_{\substack{i=0 \\ i \neq k}}^{n} (x_k - x_i) \overset{!}{=} 1$$

$$\Longrightarrow C_k = \frac{1}{\prod_{i \neq k}^{n} (x_k - x_i)} \Longrightarrow L_k(x) = \frac{\prod_{i \neq k} (x - x_i)}{\prod_{i \neq k} (x_k - x_i)}$$

satisfies the condition.

$$p_n(x_i) = \sum_{k=0}^{n} L_k(x_i)\, y_k = y_i \qquad i = 0, 1, \ldots, n.$$

interpolating polynomial $\square$