General

solve $Au = b$ w. $U_k := \begin{pmatrix} | & & | \\ u(f_1) & \dots & u(f_k) \\ | & & | \end{pmatrix}$

Decompose $U_k = Q_k R_k$ w. $Q_k^T Q_k = 1$

Now: solve $Q^T A Q v = Q^T b$

$Q_k = \begin{pmatrix} | & & | \\ q_1 & \dots & q_k \\ | & & | \end{pmatrix}$ $R_k \begin{pmatrix} r_{1,1} & & r_{1,k} \\ & \ddots & \\ & & r_{k,k} \end{pmatrix}$

$\Rightarrow u(f_k) = \sum_{i=1}^{k} q_i r_{k,i}$

$$Q_k^T A = \begin{pmatrix} — & q_1^T & — \\ & \vdots & \\ — & q_k^T & — \end{pmatrix} \begin{pmatrix} | & & | \\ a_1 & \dots & a_N \\ | & & | \end{pmatrix}$$

$$= \begin{pmatrix} q_1^T a_1 & q_1^T a_2 & \dots & q_1^T a_N \\ \vdots & & \ddots & \vdots \\ q_k^T a_1 & & \dots & q_k^T a_N \end{pmatrix}$$

$$= \begin{pmatrix} — & q_1^T A & — \\ & \vdots & \\ — & q_k^T A & — \end{pmatrix}$$

$$Q_k^T A Q_k = \begin{pmatrix} — & q_1^T A & — \\ & \vdots & \\ — & q_k^T A & — \end{pmatrix} \begin{pmatrix} | & & | \\ q_1 & \cdots & q_k \\ | & & | \end{pmatrix}$$

$$= \begin{pmatrix} q_1^T A q_1 & \cdots & q_1^T A q_k \\ & & \\ q_k^T A q_1 & \cdots & q_k^T A q_k \end{pmatrix}$$

$$= \left(\begin{array}{c|c} Q_{k-1}^T A Q_{k-1} & Q_{k-1}^T A q_k \\ \hline q_k^T A Q_{k-1} & q_k^T A q_k \end{array}\right)$$

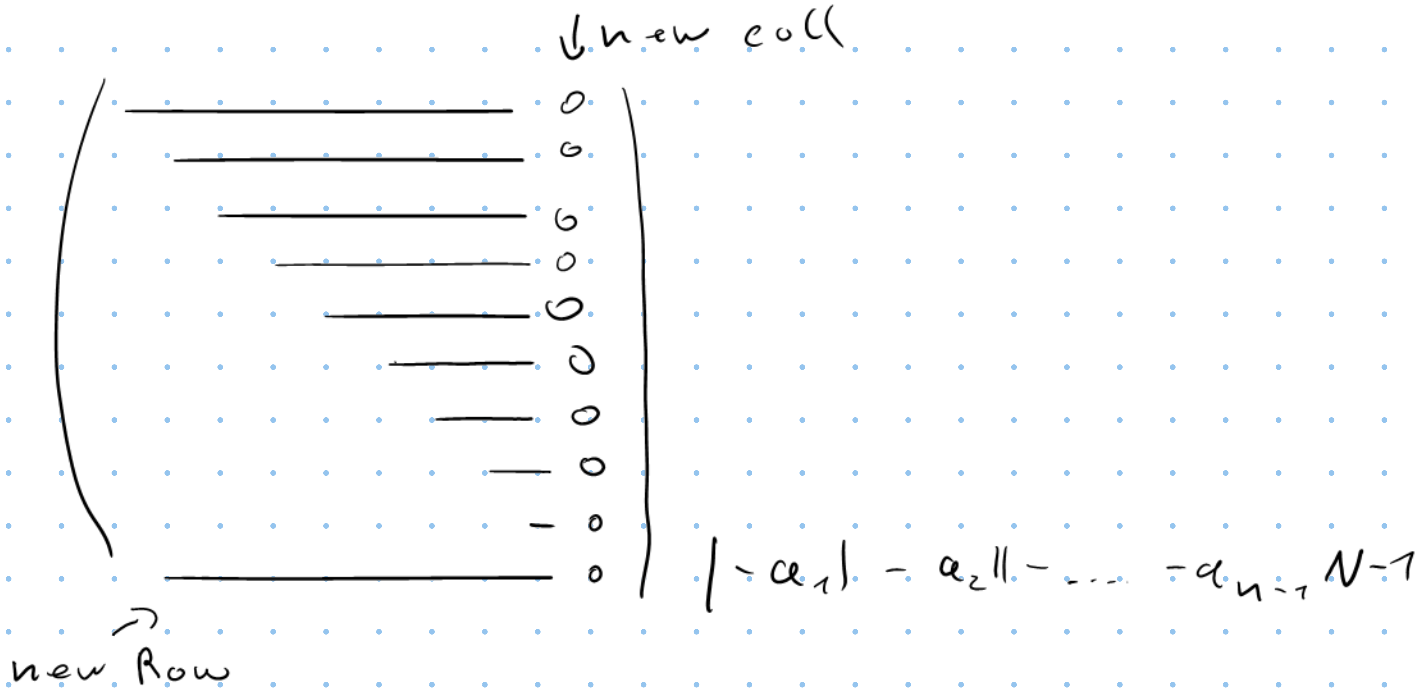
$$\Big|\ A^T = A$$

$$= \left(\begin{array}{c|c} Q_{k-1}^T A Q_{k-1} & (q_k A Q_{k-1})^T \\ \hline q_k^T A Q_{k-1} & q_k^T A q_k \end{array}\right)$$

=> Using a Gram-Schmidt based QR-decomposition

=> Projection is basically free

=> Reduces time of assembling the reduced mats



$\Rightarrow$ Nested Gauss exists

$\Rightarrow$ reduces time to solve in the online Phase

$\Rightarrow$ Cheaper evaluation of the error estimator

$\Rightarrow$ slightly higher Memory cost, since the LR-decompositions need to be stored

e.g.: 1000 100×100 complex matrices

$\Rightarrow$ 20e6 floats $\Rightarrow$ 160 MByte ≈ 0

Assembling new portmat row

$$P_i = p_i p_i^H$$

new row: $q_k^H P_i Q_k = q_k^H p_i p_i^H Q_k$

exploit sparsity:

elems$_{\neq 0}$ of p_i : $\hat{p}_i \Rightarrow \hat{p}_i = p_i[\text{inds}]$

$$\hat{q}_k = q_k[\text{inds}]$$

$$\hat{Q}_k = Q_k[\text{inds}, :]$$

$$q_k^H P_i Q_k = \hat{q}_k^H \hat{p}_i \hat{p}_i^H \hat{Q}_k$$

Portmat residual exploiting sparsity

$$r_p = P x = p p^H x = p \hat{p}^H \hat{x}$$

$$\gamma U^H P_i U = \gamma U^H p_i p_i^H U = (\gamma U^H p_i)(U^H p_i)^H$$