

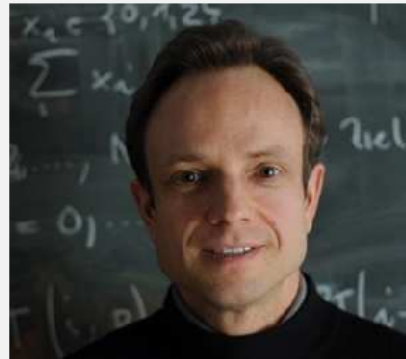
Course Video

Section 3.3.4: QR-Based Solver for Linear Least Squares Problems

Prof. R. Hiptmair, SAM, ETH Zurich

Date: September 23, 2020

(C) Seminar für Angewandte Mathematik, ETH Zürich



Dependencies. Section 3.1.1, Section 3.3.3.2, Section 3.3.3.4

Duration: 16 minutes

III. Direct Methods for Linear Least Squares Problems

3.3. Orthogonal Transformation Methods

3.3.4. QR-Based Solver for Linear Least Squares Problems

Given: $A \in \mathbb{R}^{m,n}$, $\text{rank}(A) = n$, $b \in \mathbb{R}^m$

$$x \in \mathbb{R}^n: \|Ax - b\|_2 \longrightarrow \min$$

QR-dec.: $A = QR$, $Q \in \mathbb{R}^{m,m}$ orthogonal, $Q^T Q = I$

$$\begin{aligned} \|Ax - b\|_2 &= \|QRx - b\|_2 \\ &= \|Q(Rx - Q^T b)\|_2 \\ &= \|Rx - \tilde{b}\|_2 \triangleq \text{triangular LSQ} \\ \tilde{b} &:= Q^T b \end{aligned}$$

Note: $\text{rank}(A) = n \iff \text{rank}(R) = n$

2

$$\|Ax - b\|_2 = \|Q(Rx - Q^T b)\|_2 = \|Rx - \tilde{b}\|_2, \quad \tilde{b} := Q^T b.$$

R_0 regular

$$\|Ax - b\|_2 \rightarrow \min \Leftrightarrow$$

$$\left\| \begin{bmatrix} R_0 \\ 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} \tilde{b}_1 \\ \vdots \\ \tilde{b}_m \end{bmatrix} \right\|_2 \rightarrow \min.$$

$$x = \begin{bmatrix} R_0 \\ 0 \end{bmatrix}^{-1} \begin{bmatrix} \tilde{b}_1 \\ \vdots \\ \tilde{b}_n \end{bmatrix}, \quad \text{with residual } r = b - Ax = Q \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \tilde{b}_{n+1} \\ \vdots \\ \tilde{b}_m \end{bmatrix}.$$

LSQ sol!

We need only $Q^T b := \tilde{b}$
In Eigen: $(A \in \mathbb{R}^{m,n}, m \geq n)$

C++ code 3.3.4.2: EIGEN's built-in QR-based linear least squares solver

```
// Solving a full-rank least squares problem  $\|Ax - b\|_2 \rightarrow \min$  in EIGEN
double lsqsolve_eigen(const MatrixXd& A, const VectorXd& b,
                      VectorXd& x) {
    x = A.householderQr().solve(b);
    return ((A*x - b).norm());
}
```

cost = $O(n^2 m)$



• $Q^T b$: Application of $n-1$ HHMs to b
 $\Rightarrow$ cost = $O(mn)$

• Solve LSE $R_0 x = (\tilde{b})_{1:n}$

$\Rightarrow$ cost = $O(n^2)$
[backward elimination]

Rem.: QR-reuse option,
same as for LU

Comparisons:

Normal equations vs. orthogonal transformations method

Superior numerical stability ($\rightarrow$ Def. 1.5.5.19) of orthogonal transformations methods:

- Use orthogonal transformations methods for least squares problems (3.1.3.7), whenever $A \in \mathbb{R}^{m,n}$ dense and n small.

SVD/QR-factorization cannot exploit sparsity:

- Use normal equations in the expanded form (3.2.0.8)/(3.2.0.9), when $A \in \mathbb{R}^{m,n}$ sparse ($\rightarrow$ Notion 2.7.0.1) and m, n big.

③

Review Questions :

Try to answer review questions 3.3.4.8
without aids.

4

5

7