The Linear Least Squares Problem

- In general, $Ax = b$ with $m > n$ has no solution
- Instead, try to minimize the residual $r = b - Ax$
- With the 2-norm we obtain the linear least squares problem (LSP):

  \[
  \text{Given } A \in \mathbb{C}^{m \times n}, m \geq n, b \in \mathbb{C}^m, \\
  \text{find } x \in \mathbb{C}^n \text{ such that } \|b - Ax\|_2 \text{ is minimized}
  \]

- The minimizer $x$ is the solution to the normal equations

  \[
  A^* A x = A^* b
  \]

  or, in terms of the pseudoinverse $A^+$:

  \[
  x = A^+ b, \quad \text{where } A^+ = (A^* A)^{-1} A^* \in \mathbb{C}^{n, m}
  \]
Geometric Interpretation

- Find the point $Ax$ in $\text{range}(A)$ closest to $b$
- This $x$ will minimize the 2-norm of $r = b - Ax$
- $Ax = Pb$ where $P$ is an orthogonal projector onto $\text{range}(A)$, so the residual must be orthogonal to $\text{range}(A)$
Solving the LSP – 1. Normal Equations

- If $A$ has full rank, $A^* A$ is square, hermitian positive definite system
- Solve by *Cholesky factorization* (Gaussian elimination)

**Algorithm: Least Squares via Normal Equations**

1. Form the matrix $A^* A$ and the vector $A^* b$
2. Compute the Cholesky factorization $A^* A = R^* R$
3. Solve the lower-triangular system $R^* w = A^* b$ for $w$
4. Solve the upper-triangular system $Rx = w$ for $x$

- Work $\sim$ Forming $A^* A$ + Cholesky $\sim mn^2 + n^3/3$ flops
- Fast, but sensitive to rounding errors
Solving the LSP – 2. QR Factorization

- Using $A = \hat{Q}\hat{R}$, $b$ can be projected onto $\text{range}(A)$ by $P = \hat{Q}\hat{Q}^*$
- Insert into $Ax = b$ to get $\hat{Q}\hat{R}x = \hat{Q}\hat{Q}^*b$, or $\hat{R}x = \hat{Q}^*b$

Algorithm: Least Squares via QR Factorization

1. Compute the reduced QR factorization $A = \hat{Q}\hat{R}$
2. Compute the vector $\hat{Q}^*b$
3. Solve the upper-triangular system $\hat{R}x = \hat{Q}^*b$ for $x$

- Work $\sim$ QR Factorization $\sim 2mn^2 - 2n^3/3$ flops
- Good stability, relatively fast, used in MATLAB’s “backslash” \ 

5
Solving the LSP – 3. SVD

- Using \( A = \hat{U}\hat{\Sigma}\hat{V}^* \), \( b \) can be projected onto \( \text{range}(A) \) by \( P = \hat{U}\hat{U}^* \)
- Insert into \( Ax = b \) to get \( \hat{U}\hat{\Sigma}\hat{V}^*x = \hat{U}\hat{U}^*b \), or \( \hat{\Sigma}\hat{V}^*x = \hat{U}^*b \)

**Algorithm: Least Squares via SVD**

1. Compute the reduced SVD \( A = \hat{U}\hat{\Sigma}\hat{V}^* \)
2. Compute the vector \( \hat{U}^*b \)
3. Solve the diagonal system \( \hat{\Sigma}w = \hat{U}^*b \) for \( w \)
4. Set \( x = Vw \)

- Work \( \sim \text{SVD} \sim 2mn^2 + 11n^3 \) flops
- Very good stability properties, use if \( A \) is close to rank-deficient