# Lecture 4: Householder reflector, Givens rotation, Least squares problem

School of Mathematical Sciences, Xiamen University

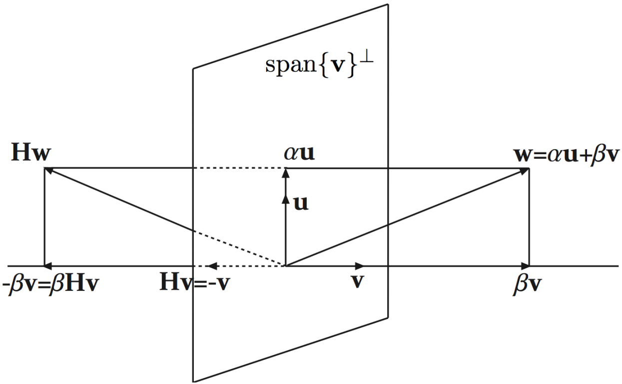
## 1. Householder reflector

- Let $\mathbf{v} \in \mathbb{C}^m$ and $\mathbf{v} \neq \mathbf{0}$. Then the matrix

$$\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^*}{\mathbf{v}^*\mathbf{v}},$$

  is called a *Householder reflector*.

- Geometric interpretation



Exercise: Householder reflector $\mathbf{H}$ satisfies the following properties:

(1) It is *Hermitian*: $\mathbf{H} = \mathbf{H}^*$

(2) It is *unitary*: $\mathbf{H}^* = \mathbf{H}^{-1}$

(3) It is *involutary*: $\mathbf{H}^2 = \mathbf{I}$

Exercise: What are the eigenvalues, the determinant, and the singular values of a Householder reflector $\mathbf{H}$?

Hint: eigenvalues 1 with multiplicity $m-1$ and $-1$ with multiplicity 1.

Exercise: Prove that $\mathbf{I} - \dfrac{\mathbf{v}\mathbf{v}^*}{\mathbf{v}^*\mathbf{v}}$ is the orthogonal projector which projects $\mathbb{C}^m$ onto the *hyperplane* $\mathrm{span}\{\mathbf{v}\}^\perp$ along $\mathrm{span}\{\mathbf{v}\}$.

## Theorem 1

*For all* $\mathbf{x}, \mathbf{y} \in \mathbb{C}^m$ *with* $\mathbf{x} \neq \mathbf{y}$*, there exists a Householder reflector* $\mathbf{H}$ *such that* $\mathbf{Hx} = \mathbf{y}$ *if and only if* $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2$ *and* $\mathbf{x}^*\mathbf{y} \in \mathbb{R}$*.*

## Proof.

"$\Rightarrow$" is easy. "$\Leftarrow$": let $\mathbf{v} = \mathbf{y} - \mathbf{x}$, verify $\mathbf{Hx} = \mathbf{y}$. □

## Corollary 2

*For all nonzero* $\mathbf{x}, \mathbf{y} \in \mathbb{C}^m$ *with* $\mathbf{x} \neq \mathbf{y}$*, there exists a Householder reflector* $\mathbf{H}$ *and* $z \in \mathbb{C}$ *such that* $\mathbf{Hx} = z\mathbf{y}$*.*

## Proof.

Let

$$z = \frac{\|\mathbf{x}\|_2}{\|\mathbf{y}\|_2} \cdot c, \qquad c = \begin{cases} \pm\mathbf{y}^*\mathbf{x}/|\mathbf{x}^*\mathbf{y}|, & \text{if } \mathbf{x}^*\mathbf{y} \neq 0, \\ \mathrm{e}^{\mathrm{i}\theta},\ \theta \in [0, 2\pi), & \text{if } \mathbf{x}^*\mathbf{y} = 0, \end{cases}$$

and $\mathbf{v} = z\mathbf{y} - \mathbf{x}$. Verify $\mathbf{Hx} = z\mathbf{y}$. □

## 2. QR factorization via Householder reflectors

- Householder method: $\mathbf{Q}_n \cdots \mathbf{Q}_2 \mathbf{Q}_1 \mathbf{A} = \mathbf{R}$ is upper-triangular.

$$\underset{\mathbf{A}}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}} \xrightarrow{\mathbf{Q}_1} \underset{\mathbf{Q}_1\mathbf{A}}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{\mathbf{Q}_2} \underset{\mathbf{Q}_2\mathbf{Q}_1\mathbf{A}}{\begin{bmatrix} \times & \times & \times \\ & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{\mathbf{Q}_3} \underset{\mathbf{Q}_3\mathbf{Q}_2\mathbf{Q}_1\mathbf{A}}{\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \boldsymbol{\times} \\ & & \mathbf{0} \\ & & \mathbf{0} \end{bmatrix}}$$

$\times$ denotes an entry not necessarily zero; "*blank*" are zeros

- At the $k$th step, the unitary matrix $\mathbf{Q}_k$ has the form

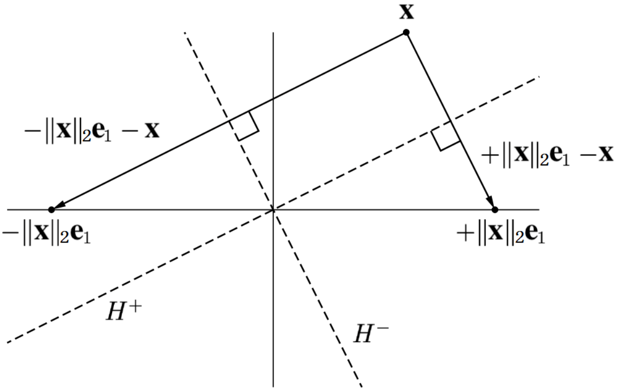
$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{I}_{k-1} & \\ & \mathbf{H}_k \end{bmatrix}.$$

Here $\mathbf{H}_k$ is an $(m-k+1) \times (m-k+1)$ Householder reflector, which maps an $m-k+1$-vector to a scalar multiple of $\mathbf{e}_1$.

- The full QR factorization: $\mathbf{A} = \mathbf{Q}_1^* \mathbf{Q}_2^* \cdots \mathbf{Q}_n^* \mathbf{R} = \mathbf{Q}\mathbf{R}$

- QR factorization with column pivoting: $\mathbf{AP} = \mathbf{QR}$. Consider "`qr`"

## 2.1. Two possible Householder reflections in real case




- Choose the one that moves $\mathbf{x}$ the larger distance, i.e., $\mathbf{v} = -\text{sign}(x_1)\|\mathbf{x}\|_2\mathbf{e}_1 - \mathbf{x}$, or $\mathbf{v} = \text{sign}(x_1)\|\mathbf{x}\|_2\mathbf{e}_1 + \mathbf{x}$
- Convention: $\text{sign}(x_1) = 1$ if $x_1 = 0$

## 2.2. Algorithms

**Algorithm**: Householder QR factorization

**for** $k = 1$ **to** $n$
$\quad \mathbf{x} = \mathbf{A}_{k:m,k}$
$\quad \mathbf{v}_k = \mathrm{sign}(x_1)\|\mathbf{x}\|_2\mathbf{e}_1 + \mathbf{x}$
$\quad \mathbf{v}_k = \mathbf{v}_k / \|\mathbf{v}_k\|_2$
$\quad \mathbf{A}_{k:m,k:n} = \mathbf{A}_{k:m,k:n} - 2\mathbf{v}_k(\mathbf{v}_k^*\mathbf{A}_{k:m,k:n})$
**end**

**Algorithm**: Implicit calculations of $\mathbf{Q}^*\mathbf{b}$ or $\mathbf{Qx}$

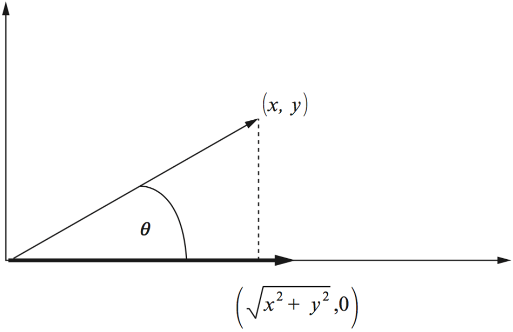
**for** $k = 1$ **to** $n$
$\quad \mathbf{b}_{k:m} = \mathbf{b}_{k:m} - 2\mathbf{v}_k(\mathbf{v}_k^*\mathbf{b}_{k:m})$
**end**
**for** $k = n$ **downto** $1$
$\quad \mathbf{x}_{k:m} = \mathbf{x}_{k:m} - 2\mathbf{v}_k(\mathbf{v}_k^*\mathbf{x}_{k:m})$
**end**

**3. Givens rotation** (We mainly consider the real case).

- The $2 \times 2$ Givens rotation

$$\mathbf{G} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \quad \cos\theta = \frac{x}{\sqrt{x^2+y^2}}, \quad \sin\theta = \frac{y}{\sqrt{x^2+y^2}}$$

rotates vector $\begin{bmatrix} x \\ y \end{bmatrix}$ in $\mathbb{R}^2$ onto the $x$-axis.



- Givens rotation for $\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sqrt{x^2+y^2} \\ 0 \end{bmatrix}$

---

**Algorithm**: Givens rotation zeroing the 2nd entry

---

**function** $[c, s] = \text{givens}(x, y)$

  **if** $y = 0$

    $c = 1, \quad s = 0$

  **else**

    **if** $|y| > |x|$

      $\tau = x/y, \quad s = 1/\sqrt{1+\tau^2}, \quad c = s\tau$

    **else**

      $\tau = y/x, \quad c = 1/\sqrt{1+\tau^2}, \quad s = c\tau$

    **end**

  **end**

---

Exercise: Design a similar algorithm for a Givens rotation zeroing the 1st entry.

- Zeroing a particular entry in a vector using a Givens rotation. Define the $m \times m$ Givens rotation $\mathbf{G}(i, j; \theta)$,

$$
\begin{aligned}
\mathbf{G}(i,j;\theta) &= \mathbf{I} + \begin{bmatrix} \mathbf{e}_i & \mathbf{e}_j \end{bmatrix} \begin{bmatrix} \cos\theta - 1 & \sin\theta \\ -\sin\theta & \cos\theta - 1 \end{bmatrix} \begin{bmatrix} \mathbf{e}_i^\top \\ \mathbf{e}_j^\top \end{bmatrix} \\
&= \begin{array}{c} \begin{bmatrix} \mathbf{I} & & & & \\ & \cos\theta & & \sin\theta & \\ & & \mathbf{I} & & \\ & -\sin\theta & & \cos\theta & \\ & & & & \mathbf{I} \end{bmatrix} \end{array} \begin{array}{l} \\ \text{row i} \\ \\ \text{row j} \\ \\ \end{array} .
\end{aligned}
$$

Exercise: Prove that the matrix $\mathbf{G}(i, j; \theta)$ is orthogonal.

- Creating a sequence of zeros in a vector using Givens rotations

$$\mathbf{G}_n \mathbf{G}_{n-1} \cdots \mathbf{G}_1 \mathbf{x}$$

- QR factorization via Givens rotations?

  Exercise: Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be given as

$$\mathbf{A} = \begin{bmatrix} \alpha_1 & \beta_2 & \beta_3 & \cdots & \beta_n \\ \gamma_2 & \alpha_2 & 0 & \cdots & 0 \\ \gamma_3 & 0 & \alpha_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \gamma_n & 0 & \cdots & 0 & \alpha_n \end{bmatrix}, \qquad \begin{aligned} \alpha_i &\neq 0, \quad i = 1\colon n, \\ \beta_i &\neq 0, \quad i = 2\colon n, \\ \gamma_i &\neq 0, \quad i = 2\colon n. \end{aligned}$$

  Describe an algorithm for QR factorization of $\mathbf{A}$ based on as few Givens rotations as possible.

- Complex case:

$$\mathbf{G} = \begin{bmatrix} c & \bar{s} \\ -s & c \end{bmatrix}, \quad c \in \mathbb{R}, \quad c^2 + |s|^2 = 1.$$

## 4. The least squares problem (LSP)

- LSP: Given $\mathbf{A} \in \mathbb{C}^{m\times n}, \mathbf{b} \in \mathbb{C}^m$; find $\mathbf{x}_{\text{ls}} \in \mathbb{C}^n$ such that

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}_{\text{ls}}\|_2 = \min_{\mathbf{x}\in\mathbb{C}^n} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2.$$

  The *least squares solution*, $\mathbf{x}_{\text{ls}}$, maybe *not* unique. Why?

- Note that the 2-norm corresponds to Euclidean distance.

  LSP means we seek a vector $\mathbf{x}_{\text{ls}} \in \mathbb{C}^n$ such that the vector $\mathbf{A}\mathbf{x}_{\text{ls}}$ is the closest point in range($\mathbf{A}$) to $\mathbf{b}$.

  The *residual*, $\mathbf{r}_{\text{ls}} = \mathbf{b} - \mathbf{A}\mathbf{x}_{\text{ls}}$, is unique. Why?

- Assume that $\mathbf{A}$ and $\mathbf{b}$ are real. Define

$$f(\mathbf{x}) := \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \mathbf{b}^\top\mathbf{b} - \mathbf{x}^\top\mathbf{A}^\top\mathbf{b} - \mathbf{b}^\top\mathbf{A}\mathbf{x} + \mathbf{x}^\top\mathbf{A}^\top\mathbf{A}\mathbf{x}.$$

  Then the gradient of $f(\mathbf{x})$ is

$$\nabla f(\mathbf{x}) = 2\mathbf{A}^\top\mathbf{A}\mathbf{x} - 2\mathbf{A}^\top\mathbf{b}.$$

## 4.1. Example: Polynomial least squares fitting

- Given $m$ distinct $x_1, \ldots, x_m \in \mathbb{C}$ and data $y_1, \ldots, y_m \in \mathbb{C}$ at these points. Consider a polynomial of degree $n-1$,

$$p(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1},$$

s.t., $p(x)$ minimizes

$$\sum_{i=1}^{m} |p(x_i) - y_i|^2 = \|\mathbf{y} - \mathbf{A}\mathbf{c}\|_2^2,$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^{n-1} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix}.$$

## 4.2. Solving the least squares problem

**Theorem 3**

*Let $\mathbf{P}$ be the orthogonal projector onto* $\operatorname{range}(\mathbf{A})$. *A vector* $\mathbf{x}$ *is a least squares solution if and only if* $\mathbf{x}$ *satisfies* $\mathbf{A}\mathbf{x} = \mathbf{P}\mathbf{b}$.

**Proof.**

$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \|\mathbf{P}\mathbf{b} - \mathbf{A}\mathbf{x} + \mathbf{b} - \mathbf{P}\mathbf{b}\|_2^2 = \|\mathbf{P}\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \|\mathbf{b} - \mathbf{P}\mathbf{b}\|_2^2.$ $\square$

**Corollary 4**

*A vector* $\mathbf{x}$ *is a least squares solution if and only if* $\mathbf{x}$ *satisfies* $\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$, *i.e.*, $\mathbf{A}^*\mathbf{r} = \mathbf{0}$, *or* $\mathbf{r} \perp \operatorname{range}(\mathbf{A})$, *where* $\mathbf{r} := \mathbf{b} - \mathbf{A}\mathbf{x}$.

**Proof.**

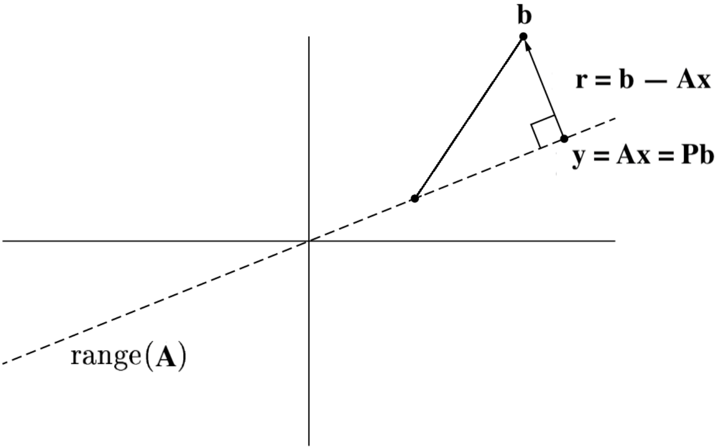
$\because \mathbf{A}^* = \mathbf{A}^*\mathbf{P}$, $\therefore \mathbf{A}^*\mathbf{r} = \mathbf{0} \Leftrightarrow \mathbf{A}^*(\mathbf{P}\mathbf{b} - \mathbf{A}\mathbf{x}) = \mathbf{0} \Leftrightarrow \mathbf{A}\mathbf{x} = \mathbf{P}\mathbf{b}.$ $\square$

- The system $\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$ is called the *normal equations*.

## Corollary 5

*The least squares solution* $\mathbf{x}$ *is unique if and only if* $\mathbf{A}^*\mathbf{A}$ *has full rank.*

## 4.3. Geometric interpretation



## 4.4. Moore–Penrose pseudoinverse solution $\mathbf{A}^{\dagger}\mathbf{b}$

- Let $\mathbf{A} \in \mathbb{C}^{m\times n}$ have an SVD $\mathbf{A} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^*$. The matrix

$$\mathbf{A}^{\dagger} = \mathbf{V}_r \mathbf{\Sigma}_r^{-1} \mathbf{U}_r^* = \sum_{j=1}^{r} \frac{1}{\sigma_j} \mathbf{v}_j \mathbf{u}_j^* \in \mathbb{C}^{n\times m},$$

  is called the *Moore–Penrose pseudoinverse* of $\mathbf{A}$. If $\mathbf{A}$ has full column rank, then $\mathbf{A}^{\dagger} = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*$. (Full row rank case?)

**Theorem 6**

*Let $\mathbf{A} \in \mathbb{C}^{m\times n}$ have rank $r < n$ and $\mathbf{b} \in \mathbb{C}^m$. Then the vector $\mathbf{A}^{\dagger}\mathbf{b}$ is the unique least squares solution with minimal 2-norm.*

**Proof.**

By SVD of $\mathbf{A}$, the least squares solutions can be expressed as

$$\mathbf{x}_{\mathrm{ls}} = \mathbf{A}^{\dagger}\mathbf{b} + \mathbf{V}_{\mathrm{c}}\mathbf{z}, \quad \mathbf{z} \in \mathbb{C}^{n-r}.$$

Then the statement follows from $\mathbf{A}^{\dagger}\mathbf{b} \perp \mathbf{V}_{\mathrm{c}}\mathbf{z}$. $\square$

## 4.5. Full column rank LSP solvers: $\text{rank}(\mathbf{A}) = n$

- Normal equations: classical way to solve LSP, best for speed
- QR factorization: "modern classical" method to solve LSP, numerically stable. By

$$\mathbf{A} = \mathbf{QR} = \begin{bmatrix} \mathbf{Q}_n & \mathbf{Q}_{\text{c}} \end{bmatrix} \begin{bmatrix} \mathbf{R}_n \\ \mathbf{0} \end{bmatrix},$$

  we have

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{C}^n} \|\mathbf{b} - \mathbf{Ax}\|_2 &= \min_{\mathbf{x}\in\mathbb{C}^n} \|\mathbf{b} - \mathbf{QRx}\|_2 = \min_{\mathbf{x}\in\mathbb{C}^n} \|\mathbf{Q}^*\mathbf{b} - \mathbf{Rx}\|_2 \\ &= \min_{\mathbf{x}\in\mathbb{C}^n} \left\| \begin{bmatrix} \mathbf{Q}_n^*\mathbf{b} - \mathbf{R}_n\mathbf{x} \\ \mathbf{Q}_{\text{c}}^*\mathbf{b} \end{bmatrix} \right\|_2 \end{aligned}$$

- SVD, numerically stable, for problems close to rank-deficient. By

$$\mathbf{A} = \mathbf{U\Sigma V}^* = \mathbf{U}_n\mathbf{\Sigma}_n\mathbf{V}^* = \begin{bmatrix} \mathbf{U}_n & \mathbf{U}_{\text{c}} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_n \\ \mathbf{0} \end{bmatrix} \mathbf{V}^*,$$

we have

$$\begin{aligned}\min_{\mathbf{x}\in\mathbb{C}^n}\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2 &= \min_{\mathbf{x}\in\mathbb{C}^n}\|\mathbf{b}-\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*\mathbf{x}\|_2\\ &= \min_{\mathbf{x}\in\mathbb{C}^n}\|\mathbf{U}^*\mathbf{b}-\boldsymbol{\Sigma}\mathbf{V}^*\mathbf{x}\|_2\\ &= \min_{\mathbf{x}\in\mathbb{C}^n}\left\|\begin{bmatrix}\mathbf{U}_n^*\mathbf{b}-\boldsymbol{\Sigma}_n\mathbf{V}^*\mathbf{x}\\ \mathbf{U}_c^*\mathbf{b}\end{bmatrix}\right\|_2.\end{aligned}$$

Exercise: Given $\mathbf{A}\in\mathbb{C}^{m\times n}$ of full column rank, $m>n$, $\mathbf{b}\in\mathbb{C}^m$, $\mathbf{b}\notin\text{range}(\mathbf{A})$ and $\mathbf{QR}=\begin{bmatrix}\mathbf{A} & \mathbf{b}\end{bmatrix}$ (i.e., full QR factorization of $\begin{bmatrix}\mathbf{A} & \mathbf{b}\end{bmatrix}$). Show that

$$\min_{\mathbf{x}\in\mathbb{C}^n}\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2 = |\mathbf{R}(n+1,n+1)|,$$

and the least squares solution is given by

$$\mathbf{x}=\mathbf{R}(1\colon n,1\colon n)\backslash\mathbf{R}(1\colon n,n+1).$$

## 4.6. Rank-deficient LSP solvers: $\operatorname{rank}(\mathbf{A}) = r < n$

- QR factorization with column pivoting:

$$\mathbf{AP} = \mathbf{QR} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $\mathbf{P}$ is a permutation matrix, $\mathbf{Q} \in \mathbb{C}^{m \times m}$ is unitary, and $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$ is nonsingular upper triangular. Introduce the auxiliary vectors

$$\mathbf{Q}^* \mathbf{b} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{P}^* \mathbf{x} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}.$$

The general least squares solution is

$$\mathbf{x}_{\text{ls}} = \mathbf{P} \begin{bmatrix} \mathbf{R}_{11}^{-1}(\mathbf{d}_1 - \mathbf{R}_{12}\mathbf{y}_2) \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{y}_2 = \text{arbitrary}.$$

The case $\mathbf{y}_2 = \mathbf{0}$ yields the least squares solution with at least $n - r$ zero components. Consider "\" in MATLAB.

- Complete orthogonal factorization (also called UTV factorization)

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}^*,$$

where $\mathbf{U} \in \mathbb{C}^{m\times m}$ is unitary, $\mathbf{V} \in \mathbb{C}^{n\times n}$ is unitary, and $\mathbf{R}_{11} \in \mathbb{R}^{r\times r}$ is nonsingular upper triangular. Introduce the auxiliary vectors

$$\mathbf{U}^*\mathbf{b} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{V}^*\mathbf{x} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}.$$

The general least squares solution is

$$\mathbf{x}_{\text{ls}} = \mathbf{V} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \mathbf{V} \begin{bmatrix} \mathbf{R}_{11}^{-1}\mathbf{g}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{y}_2 = \text{arbitrary}.$$

The case $\mathbf{y}_2 = \mathbf{0}$ yields the minimum norm least squares solution.
`http://www.netlib.org/numeralgo/`

Consider `lsqminnorm` in MATLAB.

## 5. Least squares solution flowchart