KUTAISI INTERNATIONAL UNIVERSITY

Numerical Linear Algebra

Ramaz Botchorishvili

Kutaisi International University

December 7, 2022

KUTAISI INTERNATIONAL UNIVERSITY

Numerical Linear Algebra

Ramaz Botchorishvili

Kutaisi International University

December 7, 2022

Non stationary iterative methods, QR factorization, Gram-Schmidt orthogonalization

- Method of minimal residuals
- Steepest Descent
- Gram-Schmidt orthogonalization
- QR and reduced QR factorization
- Q & A

Recap of Previous Lecture

- Recap of Previous Lecture
- Termination criteria
- Quadratic functional and linear systems
- CP2

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k (b - Ax^{(k)})$

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, ...$$

- $x^{(k+1)} = x^{(k)} + \alpha_k (b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $A \in \mathbb{R}^{n\times n}, A = A^T > 0,\ Ae^{(k)} = A(x - x^{(k)}) = b - Ax^{(k)} = r_k$

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $A \in \mathbb{R}^{n \times n}, A = A^T > 0,\ Ae^{(k)} = A(x - x^{(k)}) = b - Ax^{(k)} = r_k$
- "Best" $\equiv$ minimal residual $\|r^{(k+1)}\|_2$

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $A \in \mathbb{R}^{n\times n}, A = A^T > 0,\ Ae^{(k)} = A(x - x^{(k)}) = b - Ax^{(k)} = r_k$
- "Best" $\equiv$ minimal residual $\|r^{(k+1)}\|_2$
- $r^{(k+1)} = b - Ax^{(k+1)} = b - A(x^{(k)} + \alpha_k r_k) = b - Ax^{(k)} - \alpha_k Ar_k$

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $A \in \mathbb{R}^{n\times n}, A = A^T > 0,\ Ae^{(k)} = A(x - x^{(k)}) = b - Ax^{(k)} = r_k$
- "Best" $\equiv$ minimal residual $\|r^{(k+1)}\|_2$
- $r^{(k+1)} = b - Ax^{(k+1)} = b - A(x^{(k)} + \alpha_k r_k) = b - Ax^{(k)} - \alpha_k Ar_k$
- $r^{(k+1)} = r_k - \alpha_k Ar_k$

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \dots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $A \in \mathbb{R}^{n\times n}, A = A^T > 0,\ Ae^{(k)} = A(x - x^{(k)}) = b - Ax^{(k)} = r_k$
- "Best" $\equiv$ minimal residual $\|r^{(k+1)}\|_2$
- $r^{(k+1)} = b - Ax^{(k+1)} = b - A(x^{(k)} + \alpha_k r_k) = b - Ax^{(k)} - \alpha_k Ar_k$
- $r^{(k+1)} = r_k - \alpha_k Ar_k$
- $\varphi(\alpha) = \|r_k - \alpha Ar_k\|_2 \Rightarrow$

Method of minimal residuals, Derivation, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k (b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $A \in \mathbb{R}^{n\times n}, A = A^T > 0,\ Ae^{(k)} = A(x - x^{(k)}) = b - Ax^{(k)} = r_k$
- "Best" $\equiv$ minimal residual $\|r^{(k+1)}\|_2$
- $r^{(k+1)} = b - Ax^{(k+1)} = b - A(x^{(k)} + \alpha_k r_k) = b - Ax^{(k)} - \alpha_k A r_k$
- $r^{(k+1)} = r_k - \alpha_k A r_k$
- $\varphi(\alpha) = \|r_k - \alpha A r_k\|_2 \Rightarrow$ "Best" $\alpha_{opt} = arg\min \varphi(\alpha)$

Method of minimal residuals, Derivation, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n \times n}, A = A^T > 0$

Method of minimal residuals, Derivation, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n\times n}, A = A^T > 0$
-

$$\begin{aligned}\varphi^2(\alpha) = \|r_k - \alpha A r_k\|_2^2 = (r_k - \alpha A r_k, r_k - \alpha A r_k) = \\ = (r_k, r_k) - \alpha(r_k, A r_k) - \alpha(A r_k, r_k) + \alpha^2 (A r_k, A r_k) \\ = \|r_k\|_2^2 - 2\alpha(r_k, A r_k) + \alpha^2 \|A r_k\|_2^2\end{aligned}$$

Method of minimal residuals, Derivation, 2

- $f(x) = 0.5(Ax, x) - (b, x),\ A \in \mathbb{R}^{n\times n}, A = A^T > 0$
- $$\begin{aligned}\varphi^2(\alpha) = \|r_k - \alpha A r_k\|_2^2 = (r_k - \alpha A r_k, r_k - \alpha A r_k) = \\ = (r_k, r_k) - \alpha(r_k, Ar_k) - \alpha(Ar_k, r_k) + \alpha^2(Ar_k, Ar_k) \\ = \|r_k\|_2^2 - 2\alpha(r_k, Ar_k) + \alpha^2\|Ar_k\|_2^2\end{aligned}$$
- "Best" $\alpha_{opt} = arg \min \varphi(\alpha)$

Method of minimal residuals, Derivation, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n\times n}, A = A^T > 0$
- $$\begin{aligned} \varphi^2(\alpha) = \|r_k - \alpha A r_k\|_2^2 = (r_k - \alpha A r_k, r_k - \alpha A r_k) = \\ = (r_k, r_k) - \alpha(r_k, Ar_k) - \alpha(Ar_k, r_k) + \alpha^2(Ar_k, Ar_k) \\ = \|r_k\|_2^2 - 2\alpha(r_k, Ar_k) + \alpha^2\|Ar_k\|_2^2 \end{aligned}$$
- "Best" $\alpha_{opt} = arg\min \varphi(\alpha)$
- $\varphi'_\alpha = 2\alpha(Ar_k, Ar_k) - 2(Ar_k, r_k)$

Method of minimal residuals, Derivation, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n\times n}, A = A^T > 0$
- $$\begin{aligned}\varphi^2(\alpha) = \|r_k - \alpha A r_k\|_2^2 = (r_k - \alpha A r_k, r_k - \alpha A r_k) = \\ = (r_k, r_k) - \alpha(r_k, Ar_k) - \alpha(Ar_k, r_k) + \alpha^2(Ar_k, Ar_k) \\ = \|r_k\|_2^2 - 2\alpha(r_k, Ar_k) + \alpha^2\|Ar_k\|_2^2\end{aligned}$$
- "Best" $\alpha_{opt} = arg\min \varphi(\alpha)$
- $\varphi'_\alpha = 2\alpha(Ar_k, Ar_k) - 2(Ar_k, r_k)$
- $\varphi'_\alpha = 0 \Rightarrow \alpha_{opt} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$

Method of minimal residuals, derivation, 3

Algorithm of the method of minimal residuals for solving
$Ax = b, A = A^T > 0$

Method of minimal residuals, derivation, 3

Algorithm of the method of minimal residuals for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$

Method of minimal residuals, derivation, 3

Algorithm of the method of minimal residuals for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:

Method of minimal residuals, derivation, 3

Algorithm of the method of minimal residuals for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:
  1. $r_k = b - Ax^{(k)}$

Method of minimal residuals, derivation, 3

Algorithm of the method of minimal residuals for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:
  1. $r_k = b - Ax^{(k)}$
  2. $\alpha_k = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$

Method of minimal residuals, derivation, 3

Algorithm of the method of minimal residuals for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:
  1. $r_k = b - Ax^{(k)}$
  2. $\alpha_k = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
  3. $x^{(k+1)} = x^{(k)} + \alpha_k r_k$

Method of minimal residuals, convergence, 1

Method of minimal residuals for solving $Ax = b, A = A^T > 0$ is **non-stationary Richardson**

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:
  1. $r_k = b - Ax^{(k)}$
  2. $\alpha_k = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
  3. $x^{(k+1)} = x^{(k)} + \alpha_k r_k$

Method of minimal residuals, convergence, 1

Method of minimal residuals for solving $Ax = b, A = A^T > 0$ is **non-stationary Richardson**

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:
  1. $r_k = b - Ax^{(k)}$
  2. $\alpha_k = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
  3. $x^{(k+1)} = x^{(k)} + \alpha_k r_k$

$$\Downarrow$$

$$\begin{aligned} x^{(k+1)} = x^{(k)} + \alpha_k r_k = x^{(k)} + \alpha_k(b - Ax^{(k)}) \\ \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k(b - Ax^{(k)}) \\ \frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b \end{aligned}$$

$$B_{minres} = B_{R,\alpha_k}, \quad x^{(k+1)} = (I - \alpha_k A)x^{(k)} + b$$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$
- $\lambda_1(A) > \lambda_2(A) > ... > \lambda_n(A) > 0$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$
- $\lambda_1(A) > \lambda_2(A) > \ldots > \lambda_n(A) > 0$
- $\Downarrow$
- $\lim_{k\to\infty} x_{minres}^{(k)} = A^{-1}b$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$
- $\lambda_1(A) > \lambda_2(A) > ... > \lambda_n(A) > 0$
- $\Downarrow$
- $\lim_{k\to\infty} x_{minres}^{(k)} = A^{-1}b$
- $\|r_{minres,k}\|_2 \leq \rho_{opt}^k \|r_{minres,0}\|_2$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$
- $\lambda_1(A) > \lambda_2(A) > ... > \lambda_n(A) > 0$
- $\Downarrow$
- $\lim_{k\to\infty} x_{minres}^{(k)} = A^{-1}b$
- $\|r_{minres,k}\|_2 \leq \rho_{opt}^k \|r_{minres,0}\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)}$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$
- $\lambda_1(A) > \lambda_2(A) > ... > \lambda_n(A) > 0$
- $\Downarrow$
- $\lim_{k\to\infty} x_{minres}^{(k)} = A^{-1}b$
- $\|r_{minres,k}\|_2 \le \rho_{opt}^k \|r_{minres,0}\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)}$

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$
- $\lambda_1(A) > \lambda_2(A) > ... > \lambda_n(A) > 0$
- $\Downarrow$
- $\lim_{k\to\infty} x_{minres}^{(k)} = A^{-1}b$
- $\|r_{minres,k}\|_2 \le \rho_{opt}^k \|r_{minres,0}\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)}$

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$
- $\lambda_1(A) > \lambda_2(A) > ... > \lambda_n(A) > 0$
- $\Downarrow$
- $\lim_{k\to\infty} x_{minres}^{(k)} = A^{-1}b$
- $\|r_{minres,k}\|_2 \leq \rho_{opt}^k \|r_{minres,0}\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)}$

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$

Method of minimal residuals, convergence, 2

Theorem 12.1

- $Ax = b, A = A^T > 0$
- $\lambda_1(A) > \lambda_2(A) > ... > \lambda_n(A) > 0$
- $\Downarrow$
- $\lim_{k\to\infty} x_{minres}^{(k)} = A^{-1}b$
- $\|r_{minres,k}\|_2 \leq \rho_{opt}^k \|r_{minres,0}\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)}$

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt}

□

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k}+Ax^{(k)}=b$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A) + \lambda_n(A)}$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k,r_k)}{(Ar_k,Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k,r_k)}{(Ar_k,Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)}$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k \Rightarrow$
  $Ax^{(k+1)} - Ax^{(k)} = \alpha_k A r_k$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k \Rightarrow$
  $Ax^{(k+1)} - Ax^{(k)} = \alpha_k Ar_k \Rightarrow -(b - Ax^{(k+1)}) + (b - Ax^{(k)}) = \alpha_k Ar_k$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k \Rightarrow$ $Ax^{(k+1)} - Ax^{(k)} = \alpha_k Ar_k \Rightarrow -(b - Ax^{(k+1)}) + (b - Ax^{(k)}) = \alpha_k Ar_k$
- $r_{minres,k+1} = (I - \alpha_{minres,k}A)r_{minres,k} = B_{R_{minres,k}} r_{minres,k}$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k \Rightarrow$
  $Ax^{(k+1)} - Ax^{(k)} = \alpha_k Ar_k \Rightarrow -(b - Ax^{(k+1)}) + (b - Ax^{(k)}) = \alpha_k Ar_k$
- $r_{minres,k+1} = (I - \alpha_{minres,k}A)r_{minres,k} = B_{R_{minres,k}} r_{minres,k}$
- $r_{opt,k+1} = (I - \alpha_{opt}A)r_{opt,k} = B_{R_{opt,k}} r_{opt,k} \equiv B_{R_{opt,k}} r_k$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k \Rightarrow$
  $Ax^{(k+1)} - Ax^{(k)} = \alpha_k Ar_k \Rightarrow -(b - Ax^{(k+1)}) + (b - Ax^{(k)}) = \alpha_k Ar_k$
- $r_{minres,k+1} = (I - \alpha_{minres,k}A)r_{minres,k} = B_{R_{minres,k}} r_{minres,k}$
- $r_{opt,k+1} = (I - \alpha_{opt}A)r_{opt,k} = B_{R_{opt,k}} r_{opt,k} \equiv B_{R_{opt,k}} r_k$
- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k \Rightarrow$
  $Ax^{(k+1)} - Ax^{(k)} = \alpha_k Ar_k \Rightarrow -(b - Ax^{(k+1)}) + (b - Ax^{(k)}) = \alpha_k Ar_k$
- $r_{minres,k+1} = (I - \alpha_{minres,k}A)r_{minres,k} = B_{R_{minres,k}} r_{minres,k}$
- $r_{opt,k+1} = (I - \alpha_{opt}A)r_{opt,k} = B_{R_{opt,k}} r_{opt,k} \equiv B_{R_{opt,k}} r_k$
- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k \Rightarrow$
  $Ax^{(k+1)} - Ax^{(k)} = \alpha_k Ar_k \Rightarrow -(b - Ax^{(k+1)}) + (b - Ax^{(k)}) = \alpha_k Ar_k$
- $r_{minres,k+1} = (I - \alpha_{minres,k}A)r_{minres,k} = B_{R_{minres,k}} r_{minres,k}$
- $r_{opt,k+1} = (I - \alpha_{opt}A)r_{opt,k} = B_{R_{opt,k}} r_{opt,k} \equiv B_{R_{opt,k}} r_k$
- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt}\|r_k\|_2$

Method of minimal residuals, convergence, 3

Proof.

- Richardson: $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} + Ax^{(k)} = b$
- **Minimal residuals** $\alpha_k = \alpha_{minres,k} = \frac{(Ar_k, r_k)}{(Ar_k, Ar_k)}$
- **Richardson with optimal step** $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_1(A)+\lambda_n(A)}$
- Approach: $\alpha_{minres,k}$ vs α_{opt} for the same r_k, x_k
- $\frac{x^{(k+1)}-x^{(k)}}{\alpha_k} = b - Ax^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha_k r_k \Rightarrow$
  $Ax^{(k+1)} - Ax^{(k)} = \alpha_k Ar_k \Rightarrow -(b - Ax^{(k+1)}) + (b - Ax^{(k)}) = \alpha_k Ar_k$
- $r_{minres,k+1} = (I - \alpha_{minres,k}A)r_{minres,k} = B_{R_{minres,k}} r_{minres,k}$
- $r_{opt,k+1} = (I - \alpha_{opt}A)r_{opt,k} = B_{R_{opt,k}} r_{opt,k} \equiv B_{R_{opt,k}} r_k$
- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt}\|r_k\|_2$
- $\|r_{minres,k+1}\| \leq \rho_{opt}\|r_{minres,k}\|_2$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt} \|r_k\|_2$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt}\|r_k\|_2$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt}\|r_k\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}\|r_{minres,k}\|_2$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \le \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \le \|r_{opt,k+1}\|_2 \le \rho_{opt}\|r_k\|_2$
- $\|r_{minres,k+1}\|_2 \le \rho_{opt}\|r_{minres,k}\|_2$
- $\|r_{minres,k+1}\|_2 \le \rho_{opt}^2\|r_{minres,k-1}\|_2$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt} \|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A) - \lambda_n(A)}{\lambda_1(A) + \lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt} \|r_k\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt} \|r_{minres,k}\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}^2 \|r_{minres,k-1}\|_2$
- ...
- $\|r_{minres,k+1}\| \leq \rho_{opt}^{k+1} \|r_{minres,0}\|_2$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt}\|r_k\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}\|r_{minres,k}\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}^2\|r_{minres,k-1}\|_2$
- ...
- $\|r_{minres,k+1}\| \leq \rho_{opt}^{k+1}\|r_{minres,0}\|_2$
- $\rho_{opt} < 1 \Rightarrow lim_{k\to\infty}\|r_{minres,k}\|_2 = 0$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt}\|r_k\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}\|r_{minres,k}\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}^2\|r_{minres,k-1}\|_2$
- $\ldots$
- $\|r_{minres,k+1}\| \leq \rho_{opt}^{k+1}\|r_{minres,0}\|_2$
- $\rho_{opt} < 1 \Rightarrow lim_{k\to\infty}\|r_{minres,k}\|_2 = 0$
- $e^{(k)} = x - x^{(k)} = A^{-1}b - x^{(k)} = A^{-1}(b - Ax^{(k)}) = A^{-1}r_k$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A)-\lambda_n(A)}{\lambda_1(A)+\lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt}\|r_k\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}\|r_{minres,k}\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}^2\|r_{minres,k-1}\|_2$
- ...
- $\|r_{minres,k+1}\| \leq \rho_{opt}^{k+1}\|r_{minres,0}\|_2$
- $\rho_{opt} < 1 \Rightarrow lim_{k\to\infty}\|r_{minres,k}\|_2 = 0$
- $e^{(k)} = x - x^{(k)} = A^{-1}b - x^{(k)} = A^{-1}(b - Ax^{(k)}) = A^{-1}r_k$
- $\|e_{minres}^{(k)}\|_2 \leq \|A^{-1}\|_2\|r_{minres,k}\|_2 \to^{k\to\infty} 0$

Method of minimal residuals, convergence, 4

Proof.

- $\|r_{opt,k+1}\|_2 = \|B_{R_{opt,k}} r_{opt,k}\|_2 \leq \|B_{R_{opt,k}}\|_2 \|r_{opt,k}\|_2 = \rho_{opt}\|r_k\|_2$
- $\rho_{opt} = \frac{\lambda_1(A) - \lambda_n(A)}{\lambda_1(A) + \lambda_n(A)} < 1$
- $\|r_{minres,k+1}\|_2 \leq \|r_{opt,k+1}\|_2 \leq \rho_{opt}\|r_k\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}\|r_{minres,k}\|_2$
- $\|r_{minres,k+1}\|_2 \leq \rho_{opt}^2\|r_{minres,k-1}\|_2$
- ...
- $\|r_{minres,k+1}\| \leq \rho_{opt}^{k+1}\|r_{minres,0}\|_2$
- $\rho_{opt} < 1 \Rightarrow lim_{k\to\infty}\|r_{minres,k}\|_2 = 0$
- $e^{(k)} = x - x^{(k)} = A^{-1}b - x^{(k)} = A^{-1}(b - Ax^{(k)}) = A^{-1}r_k$
- $\|e_{minres}^{(k)}\|_2 \leq \|A^{-1}\|_2\|r_{minres,k}\|_2 \to^{k\to\infty} 0$

□

Gradient method, steepest descent, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

Gradient method, steepest descent, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, ...$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$

Gradient method, steepest descent, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k (b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$

Gradient method, steepest descent, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, ...$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?

Gradient method, steepest descent, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n \times n}, A = A^T > 0$

Gradient method, steepest descent, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k (b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n \times n}, A = A^T > 0$
- $Ax^{(k+1)} = b \;\equiv\; x^{(k+1)} = arg\min f(x)$

Gradient method, steepest descent, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, ...$$

- $x^{(k+1)} = x^{(k)} + \alpha_k(b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n\times n}, A = A^T > 0$
- $Ax^{(k+1)} = b \; \equiv \; x^{(k+1)} = arg\min f(x)$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k)$

Gradient method, steepest descent, 1

- Non-stationary Richardson method

$$\frac{x^{(k+1)} - x^{(k)}}{\alpha_k} + Ax^{(k)} = b, \quad k = 0, 1, 2, \ldots$$

- $x^{(k+1)} = x^{(k)} + \alpha_k (b - Ax^{(k)})$
- $x^{(k+1)} = x^{(k)} + \alpha_k r_k$
- What is "best" α_k?
- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n \times n}, A = A^T > 0$
- $Ax^{(k+1)} = b \ \equiv \ x^{(k+1)} = arg \min f(x)$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k)$
- $\Rightarrow$ "Best" $\alpha_{opt} = arg \min \varphi(\alpha)$

Gradient method, steepest descent, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n \times n}, A = A^T > 0$

Gradient method, steepest descent, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n\times n}, A = A^T > 0$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k) = 0.5(A(x^{(k)} + \alpha r_k), x^{(k)} + \alpha r_k) - (b, x^{(k)} + \alpha r_k)$

Gradient method, steepest descent, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n \times n}, A = A^T > 0$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k) = 0.5(A(x^{(k)} + \alpha r_k), x^{(k)} + \alpha r_k) - (b, x^{(k)} + \alpha r_k)$
- $$\begin{aligned}\varphi(\alpha) = 0.5[(Ax^{(k)}, x^{(k)}) + \alpha(Ax^{(k)}, r_k) + \alpha(Ar_k, x^{(k)}) + \alpha^2(Ar_k, r_k)] \\ -(b, x^{(k)}) - \alpha(b, r_k)\end{aligned}$$

Gradient method, steepest descent, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n\times n}, A = A^T > 0$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k) = 0.5(A(x^{(k)} + \alpha r_k), x^{(k)} + \alpha r_k) - (b, x^{(k)} + \alpha r_k)$
- $$\begin{aligned}\varphi(\alpha) = 0.5[(Ax^{(k)}, x^{(k)}) + \alpha(Ax^{(k)}, r_k) + \alpha(Ar_k, x^{(k)}) + \alpha^2(Ar_k, r_k)]\\ -(b, x^{(k)}) - \alpha(b, r_k)\end{aligned}$$
- $\varphi(\alpha) = f(x^{(k)}) + \alpha(Ax^{(k)} - b, r_k) + 0.5\alpha^2(Ar_k, r_k)$

Gradient method, steepest descent, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n \times n}, A = A^T > 0$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k) = 0.5(A(x^{(k)} + \alpha r_k), x^{(k)} + \alpha r_k) - (b, x^{(k)} + \alpha r_k)$
- $$\varphi(\alpha) = 0.5[(Ax^{(k)}, x^{(k)}) + \alpha(Ax^{(k)}, r_k) + \alpha(Ar_k, x^{(k)}) + \alpha^2(Ar_k, r_k)] \\ -(b, x^{(k)}) - \alpha(b, r_k)$$
- $\varphi(\alpha) = f(x^{(k)}) + \alpha(Ax^{(k)} - b, r_k) + 0.5\alpha^2(Ar_k, r_k)$
- $\varphi(\alpha) = f(x^{(k)}) - \alpha(r_k, r_k) + 0.5\alpha^2(Ar_k, r_k)$

Gradient method, steepest descent, 2

- $f(x) = 0.5(Ax, x) - (b, x)$, $A \in \mathbb{R}^{n \times n}, A = A^T > 0$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k) = 0.5(A(x^{(k)} + \alpha r_k), x^{(k)} + \alpha r_k) - (b, x^{(k)} + \alpha r_k)$
- $$\begin{aligned}\varphi(\alpha) = 0.5[(Ax^{(k)}, x^{(k)}) + \alpha(Ax^{(k)}, r_k) + \alpha(Ar_k, x^{(k)}) + \alpha^2(Ar_k, r_k)] \\ -(b, x^{(k)}) - \alpha(b, r_k)\end{aligned}$$
- $\varphi(\alpha) = f(x^{(k)}) + \alpha(Ax^{(k)} - b, r_k) + 0.5\alpha^2(Ar_k, r_k)$
- $\varphi(\alpha) = f(x^{(k)}) - \alpha(r_k, r_k) + 0.5\alpha^2(Ar_k, r_k)$
- "Best" $\alpha_{opt} = arg \min \varphi(\alpha)$

Gradient method, steepest descent, 2

- $f(x) = 0.5(Ax, x) - (b, x),\ A \in \mathbb{R}^{n\times n}, A = A^T > 0$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k) = 0.5(A(x^{(k)} + \alpha r_k), x^{(k)} + \alpha r_k) - (b, x^{(k)} + \alpha r_k)$
- $$\varphi(\alpha) = 0.5[(Ax^{(k)}, x^{(k)}) + \alpha(Ax^{(k)}, r_k) + \alpha(Ar_k, x^{(k)}) + \alpha^2(Ar_k, r_k)] \\ -(b, x^{(k)}) - \alpha(b, r_k)$$
- $\varphi(\alpha) = f(x^{(k)}) + \alpha(Ax^{(k)} - b, r_k) + 0.5\alpha^2(Ar_k, r_k)$
- $\varphi(\alpha) = f(x^{(k)}) - \alpha(r_k, r_k) + 0.5\alpha^2(Ar_k, r_k)$
- "Best" $\alpha_{opt} = arg\min \varphi(\alpha)$
- $$\varphi'_\alpha = \alpha(Ar_k, r_k) - (r_k, r_k)$$

Gradient method, steepest descent, 2

- $f(x) = 0.5(Ax, x) - (b, x),\ A \in \mathbb{R}^{n\times n}, A = A^T > 0$
- $\varphi(\alpha) = f(x^{(k)} + \alpha r_k) = 0.5(A(x^{(k)} + \alpha r_k), x^{(k)} + \alpha r_k) - (b, x^{(k)} + \alpha r_k)$
- $$\varphi(\alpha) = 0.5[(Ax^{(k)}, x^{(k)}) + \alpha(Ax^{(k)}, r_k) + \alpha(Ar_k, x^{(k)}) + \alpha^2(Ar_k, r_k)] \\ -(b, x^{(k)}) - \alpha(b, r_k)$$
- $\varphi(\alpha) = f(x^{(k)}) + \alpha(Ax^{(k)} - b, r_k) + 0.5\alpha^2(Ar_k, r_k)$
- $\varphi(\alpha) = f(x^{(k)}) - \alpha(r_k, r_k) + 0.5\alpha^2(Ar_k, r_k)$
- "Best" $\alpha_{opt} = arg \min \varphi(\alpha)$
- $$\varphi'_\alpha = \alpha(Ar_k, r_k) - (r_k, r_k)$$
- $\varphi'_\alpha = 0 \Rightarrow \alpha_{opt} = \frac{(r_k, r_k)}{(Ar_k, r_k)}$

Gradient method, steepest descent, 3

Algorithm of gradient method(steepest descent) for solving
$Ax = b, A = A^T > 0$

Gradient method, steepest descent, 3

Algorithm of gradient method(steepest descent) for solving
$Ax = b, A = A^T > 0$

- Choose $x^{(0)}$, set $k = 0$

Gradient method, steepest descent, 3

Algorithm of gradient method(steepest descent) for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:

Gradient method, steepest descent, 3

Algorithm of gradient method(steepest descent) for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:
  1. $r_k = b - Ax^{(k)}$

Gradient method, steepest descent, 3

Algorithm of gradient method(steepest descent) for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:
  1. $r_k = b - Ax^{(k)}$
  2. $\alpha_k = \frac{(r_k, r_k)}{(Ar_k, r_k)}$

Gradient method, steepest descent, 3

Algorithm of gradient method(steepest descent) for solving
$$Ax = b, A = A^T > 0$$

- Choose $x^{(0)}$, set $k = 0$
- Do until termination criterion is satisfied:
  1. $r_k = b - Ax^{(k)}$
  2. $\alpha_k = \frac{(r_k, r_k)}{(Ar_k, r_k)}$
  3. $x^{(k+1)} = x^{(k)} + \alpha_k r_k$

Orthogonal vectors

Problem 12.2

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2$.

Orthogonalization of two vectors 1

- $y_1 = x_1$

Orthogonal vectors

Problem 12.2

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2$.

Orthogonalization of two vectors 1

- $y_1 = x_1$
- $y_2 = x_2 + \alpha y_1$

Orthogonal vectors

Problem 12.2

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2$.

Orthogonalization of two vectors 1

- $y_1 = x_1$
- $y_2 = x_2 + \alpha y_1$
- $y_1 \perp y_2$

Orthogonal vectors

Problem 12.2

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2$.

Orthogonalization of two vectors 1

- $y_1 = x_1$
- $y_2 = x_2 + \alpha y_1$
- $y_1 \perp y_2 \Rightarrow (y_1, y_2) = 0$

Orthogonal vectors

Problem 12.2

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2$.

Orthogonalization of two vectors 1

- $y_1 = x_1$
- $y_2 = x_2 + \alpha y_1$
- $y_1 \perp y_2 \Rightarrow (y_1, y_2) = 0$

$$\Downarrow$$

- $(y_1, x_2) + \alpha(y_1, y_1) = 0$

Orthogonal vectors

Problem 12.2

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2$.

Orthogonalization of two vectors 1

- $y_1 = x_1$
- $y_2 = x_2 + \alpha y_1$
- $y_1 \perp y_2 \Rightarrow (y_1, y_2) = 0$

$$\Downarrow$$

- $(y_1, x_2) + \alpha(y_1, y_1) = 0$
- $\alpha = -\frac{(y_1, x_2)}{(y_1, y_1)}$

Orthogonal vectors

Problem 12.2

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2$.

Orthogonalization of two vectors 1

- $y_1 = x_1$
- $y_2 = x_2 + \alpha y_1$
- $y_1 \perp y_2 \Rightarrow (y_1, y_2) = 0$

$$\Downarrow$$

- $(y_1, x_2) + \alpha(y_1, y_1) = 0$
- $\alpha = -\frac{(y_1,x_2)}{(y_1,y_1)}$
- $y_1 = x_1, y_2 = x_2 - \frac{(y_1,x_2)}{(y_1,y_1)} y_1$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1 / \|x_1\|$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1/\|x_1\| \to \|y_1\| = 1$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1/\|x_1\| \rightarrow \|y_1\| = 1$
- $\tilde{y}_2 = x_2 + \alpha y_1$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1/\|x_1\| \rightarrow \|y_1\| = 1$
- $\tilde{y}_2 = x_2 + \alpha y_1$
- $y_1 \perp \tilde{y}_2$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1/\|x_1\| \to \|y_1\| = 1$
- $\tilde{y}_2 = x_2 + \alpha y_1$
- $y_1 \perp \tilde{y}_2 \Rightarrow (y_1, \tilde{y}_2) = 0$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1/\|x_1\| \rightarrow \|y_1\| = 1$
- $\tilde{y}_2 = x_2 + \alpha y_1$
- $y_1 \perp \tilde{y}_2 \Rightarrow (y_1, \tilde{y}_2) = 0$

$$\Downarrow$$

- $(y_1, x_2) + \alpha(y_1, y_1) = 0$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1 / \|x_1\| \to \|y_1\| = 1$
- $\tilde{y}_2 = x_2 + \alpha y_1$
- $y_1 \perp \tilde{y}_2 \Rightarrow (y_1, \tilde{y}_2) = 0$

$$\Downarrow$$

- $(y_1, x_2) + \alpha(y_1, y_1) = 0$
- $\alpha = -\frac{(y_1, x_2)}{(y_1, y_1)}$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1/\|x_1\| \rightarrow \|y_1\| = 1$
- $\tilde{y}_2 = x_2 + \alpha y_1$
- $y_1 \perp \tilde{y}_2 \Rightarrow (y_1, \tilde{y}_2) = 0$

$$\Downarrow$$

- $(y_1, x_2) + \alpha(y_1, y_1) = 0$
- $\alpha = -\frac{(y_1, x_2)}{(y_1, y_1)}$
- $\tilde{y}_2 = x_2 - \frac{(y_1, x_2)}{(y_1, y_1)} y_1$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1/\|x_1\| \rightarrow \|y_1\| = 1$
- $\tilde{y}_2 = x_2 + \alpha y_1$
- $y_1 \perp \tilde{y}_2 \Rightarrow (y_1, \tilde{y}_2) = 0$

$$\Downarrow$$

- $(y_1, x_2) + \alpha(y_1, y_1) = 0$
- $\alpha = -\frac{(y_1, x_2)}{(y_1, y_1)}$
- $\tilde{y}_2 = x_2 - \frac{(y_1, x_2)}{(y_1, y_1)} y_1$
- $y_2 = \tilde{y}_2/\|\tilde{y}_2\|$

Orthogonal vectors 2

Problem 12.3

$x_1 \in \mathcal{R}^n, x_2 \in \mathcal{R}^n$, linearly independent. Obtain y_1, y_2 such that $y_1 \perp y_2, \|y_1\| = 1, \|y_2\| = 1$.

Orthonormalization of two vectors

- $y_1 = x_1/\|x_1\| \rightarrow \|y_1\| = 1$
- $\tilde{y}_2 = x_2 + \alpha y_1$
- $y_1 \perp \tilde{y}_2 \Rightarrow (y_1, \tilde{y}_2) = 0$

$$\Downarrow$$

- $(y_1, x_2) + \alpha(y_1, y_1) = 0$
- $\alpha = -\frac{(y_1,x_2)}{(y_1,y_1)}$
- $\tilde{y}_2 = x_2 - \frac{(y_1,x_2)}{(y_1,y_1)} y_1$
- $y_2 = \tilde{y}_2/\|\tilde{y}_2\| \rightarrow \|y_2\| = 1$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1 \Rightarrow (y_i, \tilde{y}_k) = 0, i = 1, 2, ..., k-1$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1 \Rightarrow (y_i, \tilde{y}_k) = 0, i = 1, 2, ..., k-1 \Downarrow$
- $(y_1, x_k) + \alpha_1 (y_1, y_1) = 0$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1 \Rightarrow (y_i, \tilde{y}_k) = 0, i = 1, 2, ..., k-1 \Downarrow$
- $(y_1, x_k) + \alpha_1 (y_1, y_1) = 0 \Rightarrow \alpha_1 = -\frac{(y_1, x_k)}{(y_1, y_1)} = (y_1, x_k)$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1 \Rightarrow (y_i, \tilde{y}_k) = 0, i = 1, 2, ..., k-1 \Downarrow$
- $(y_1, x_k) + \alpha_1 (y_1, y_1) = 0 \Rightarrow \alpha_1 = -\frac{(y_1, x_k)}{(y_1, y_1)} = (y_1, x_k)$
- ...
- $(y_{k-1}, x_k) + \alpha_{k-1}(y_{k-1}, y_{k-1}) = 0$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1 \Rightarrow (y_i, \tilde{y}_k) = 0, i = 1, 2, ..., k-1 \Downarrow$
- $(y_1, x_k) + \alpha_1 (y_1, y_1) = 0 \Rightarrow \alpha_1 = -\frac{(y_1, x_k)}{(y_1, y_1)} = (y_1, x_k)$
- ...
- $(y_{k-1}, x_k) + \alpha_{k-1}(y_{k-1}, y_{k-1}) = 0 \Rightarrow \alpha_{k-1} = \frac{-(y_{k-1}, x_k)}{(y_{k-1}, y_{k-1})} = (y_{k-1}, x_k)$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1 \Rightarrow (y_i, \tilde{y}_k) = 0, i = 1, 2, ..., k-1 \Downarrow$
- $(y_1, x_k) + \alpha_1 (y_1, y_1) = 0 \Rightarrow \alpha_1 = -\frac{(y_1, x_k)}{(y_1, y_1)} = (y_1, x_k)$
- ...
- $(y_{k-1}, x_k) + \alpha_{k-1}(y_{k-1}, y_{k-1}) = 0 \Rightarrow \alpha_{k-1} = \frac{-(y_{k-1}, x_k)}{(y_{k-1}, y_{k-1})} = (y_{k-1}, x_k)$
- $\tilde{y}_k = x_k - (y_1, x_k) y_1 - ... - (y_{k-1}, x_k) x_{k-1}$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1 \Rightarrow (y_i, \tilde{y}_k) = 0, i = 1, 2, ..., k-1 \Downarrow$
- $(y_1, x_k) + \alpha_1 (y_1, y_1) = 0 \Rightarrow \alpha_1 = -\frac{(y_1, x_k)}{(y_1, y_1)} = (y_1, x_k)$
- ...
- $(y_{k-1}, x_k) + \alpha_{k-1}(y_{k-1}, y_{k-1}) = 0 \Rightarrow \alpha_{k-1} = \frac{-(y_{k-1}, x_k)}{(y_{k-1}, y_{k-1})} = (y_{k-1}, x_k)$
- $\tilde{y}_k = x_k - (y_1, x_k) y_1 - ... - (y_{k-1}, x_k) x_{k-1}$
- $y_k = \tilde{y}_k / \|\tilde{y}_k\|$

Orthogonal vectors 3

Problem 12.4

$$y_1, y_2, ..., y_{k-1} \in \mathcal{R}^n, \|y_i\| = 1, y_i \perp y_j, i \neq j, 1 \leq i, j, \leq k-1, x_k \in \mathcal{R}^n.$$

$y_1, y_2, ..., y_{k-1}, x_k$ - linearly independent. Obtain y_k such that

$$y_k \perp y_i, i = 1, 2, ..., k-1, \|y_k\| = 1$$

Orthonormal vector to a set of vectors

- $\tilde{y}_k = x_k + \alpha_1 y_1 + ... + \alpha_{k-1} x_{k-1}$
- $\tilde{y}_k \perp y_i, i = 1, 2, ..., k-1 \Rightarrow (y_i, \tilde{y}_k) = 0, i = 1, 2, ..., k-1 \Downarrow$
- $(y_1, x_k) + \alpha_1 (y_1, y_1) = 0 \Rightarrow \alpha_1 = -\frac{(y_1, x_k)}{(y_1, y_1)} = (y_1, x_k)$
- ...
- $(y_{k-1}, x_k) + \alpha_{k-1}(y_{k-1}, y_{k-1}) = 0 \Rightarrow \alpha_{k-1} = \frac{-(y_{k-1}, x_k)}{(y_{k-1}, y_{k-1})} = (y_{k-1}, x_k)$
- $\tilde{y}_k = x_k - (y_1, x_k) y_1 - ... - (y_{k-1}, x_k) x_{k-1}$
- $y_k = \tilde{y}_k / \|\tilde{y}_k\| \rightarrow \|y_k\| = 1$

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output: n-vectors $y_1, y_2, ..., y_m$

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1} (y_j, x_i) y_j$

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output: n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{||\tilde{y}_i||}$

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1} (y_j, x_i) y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{||\tilde{y}_i||}$
- $2nm^2$ flops for m different n-vectors

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1} (y_j, x_i) y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$
- $2nm^2$ flops for m different n-vectors

Some properties of CGS

- CGS approach: compute vector which is orthogonal to all previously constructed vectors

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, \ldots, x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, \ldots, y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, \ldots, m$, inner product $(.,.)$
  - for $i = 1, 2, \ldots, m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1} (y_j, x_i) y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$
- $2nm^2$ flops for m different n-vectors

Some properties of CGS

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- Sensitive to small perturbations

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$
- $2nm^2$ flops for m different n-vectors

Some properties of CGS

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- Sensitive to small perturbations
- $y_i, i = 1, 2, ..., n$ may not be orthogonal due to numerical errors

Classic Gram-Schmidt (CGS)

- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1} (y_j, x_i) y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$
- $2nm^2$ flops for m different n-vectors

Some properties of CGS

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- Sensitive to small perturbations
- $y_i, i = 1, 2, ..., n$ may not be orthogonal due to numerical errors
- Modification having better stability needed (Modified Gram-Schmidt)

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output: n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - do:
    1. $x_i^{(1)} = x_i, i = 1, 2, ..., m$

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - do:
    1. $x_i^{(1)} = x_i, i = 1, 2, ..., m$
    2. $y_1 = x_1^{(1)} / \|x_1^{(1)}\|$

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j, \; i \neq j, \; i, j = 1, 2, ..., m$, inner product $(.,.)$
  - do:
    1. $x_i^{(1)} = x_i, i = 1, 2, ..., m$
    2. $y_1 = x_1^{(1)} / \|x_1^{(1)}\|$
    3. $x_i^{(2)} = x_i^{(1)} - (x_i^{(1)}, y_1) y_1, i = 2, 3, ..., m$

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - do:
    1. $x_i^{(1)} = x_i, i = 1, 2, ..., m$
    2. $y_1 = x_1^{(1)} / \|x_1^{(1)}\|$
    3. $x_i^{(2)} = x_i^{(1)} - (x_i^{(1)}, y_1) y_1, i = 2, 3, ..., m$
    4. $y_2 = x_2^{(2)} / \|x_2^{(2)}\|$

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j, \; i \neq j, \; i, j = 1, 2, ..., m$, inner product $(.,.)$
  - do:
    1. $x_i^{(1)} = x_i, i = 1, 2, ..., m$
    2. $y_1 = x_1^{(1)} / \|x_1^{(1)}\|$
    3. $x_i^{(2)} = x_i^{(1)} - (x_i^{(1)}, y_1) y_1, i = 2, 3, ..., m$
    4. $y_2 = x_2^{(2)} / \|x_2^{(2)}\|$
    5. $x_i^{(3)} = x_i^{(2)} - (x_i^{(2)}, y_2) y_2, i = 3, 4, ..., m$

Modified Gram-Schmidt (MGS), 1

Example 12.5

CGS approach vs MGS approach

- CGS approach: compute vector which is orthogonal to all previously constructed vectors
- MGS approach: compute vector and make all remaining vectors orthogonal to this vector

- Modified Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$, linearly independent
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j, \; i \neq j, \; i, j = 1, 2, ..., m$, inner product $(.,.)$
  - do:
    1. $x_i^{(1)} = x_i, i = 1, 2, ..., m$
    2. $y_1 = x_1^{(1)} / \|x_1^{(1)}\|$
    3. $x_i^{(2)} = x_i^{(1)} - (x_i^{(1)}, y_1) y_1, i = 2, 3, ..., m$
    4. $y_2 = x_2^{(2)} / \|x_2^{(2)}\|$
    5. $x_i^{(3)} = x_i^{(2)} - (x_i^{(2)}, y_2) y_2, i = 3, 4, ..., m$
    6. $y_3 = x_3^{(3)} / \|x_3^{(3)}\|$
    7. ...

Modified Gram-Schmidt (MGS), 2

- Reduced QR factorization using MGS algorithm, see p.204 in the textbook
- $A \in \mathbb{R}^{n \times m}, n > m, rank(A) = m, A = QR, Q \in \mathbb{R}^{n \times m}, R \in \mathbb{R}^{m \times m},$
- Orthogonality with MGS

$$Q^T Q = I + E, \|E\| \approx \mu cond(A)$$

- Orthogonality of Q is improved by applying MGS to columns of Q
- For better results apply MGS twice

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n \times n}, Q^T Q = I$

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n \times n}, Q^T Q = I$
- $\exists$ upper trapezoidal R :

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n\times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n\times n}, Q^T Q = I$
- $\exists$ upper trapezoidal R :
    - $R \in \mathcal{R}^{m\times m}$

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n \times n}, Q^T Q = I$
- $\exists$ upper trapezoidal R :
  - $R \in \mathcal{R}^{m \times m}$
  - $r_{ij} = 0, \text{ for } i \geq n + 1$

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n\times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n\times n}, Q^TQ = I$
- $\exists$ upper trapezoidal R :
  - $R \in \mathcal{R}^{m\times m}$
  - $r_{ij} = 0, \text{ for } i \geq n+1$
- $A = QR$

Definition 12.7

Reduced QR fatorization of rectangular matrix:

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n \times n}, Q^T Q = I$
- $\exists$ upper trapezoidal R :
  - $R \in \mathcal{R}^{m \times m}$
  - $r_{ij} = 0, \text{ for } i \geq n + 1$
- $A = QR$

Definition 12.7

Reduced QR fatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n\times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n\times n}, Q^TQ = I$
- $\exists$ upper trapezoidal R :
  - $R \in \mathcal{R}^{m\times m}$
  - $r_{ij} = 0,$ for $i \geq n+1$
- $A = QR$

Definition 12.7

Reduced QR fatorization of rectangular matrix:

- $A \in \mathcal{R}^{n\times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n\times m}, Q^TQ = I$

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n \times n}, Q^T Q = I$
- $\exists$ upper trapezoidal R :
  - $R \in \mathcal{R}^{m \times m}$
  - $r_{ij} = 0, \;$ for $i \geq n + 1$
- $A = QR$

Definition 12.7

Reduced QR fatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n \times m}, Q^T Q = I$
- $\exists$ triangular $R \in \mathcal{R}^{m \times m}$

QR factorization

Definition 12.6

QR Fcatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n \times n}, Q^T Q = I$
- $\exists$ upper trapezoidal R :
  - $R \in \mathcal{R}^{m \times m}$
  - $r_{ij} = 0, \text{ for } i \geq n + 1$
- $A = QR$

Definition 12.7

Reduced QR fatorization of rectangular matrix:

- $A \in \mathcal{R}^{n \times m}$
- $\exists$ orthogonal $Q \in \mathcal{R}^{n \times m}, Q^T Q = I$
- $\exists$ triangular $R \in \mathcal{R}^{m \times m}$
- $A = QR$

QR factorization

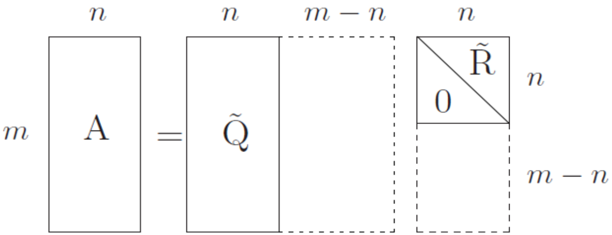

Figure: $A = QR$, $A = \tilde{Q}\tilde{R}$ - reduced QR factorization. From Quarteroni et al.

QR factorization

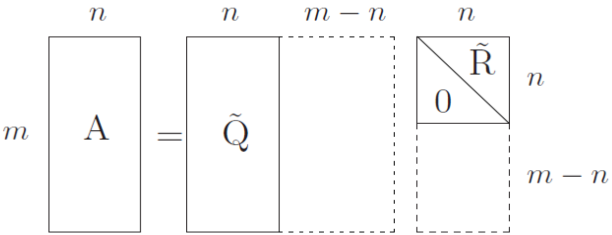

Figure: $A = QR$, $A = \tilde{Q}\tilde{R}$ - reduced QR factorization. From Quarteroni et al.

Theorem 12.8

- $A \in \mathcal{R}^{n\times m}$

QR factorization

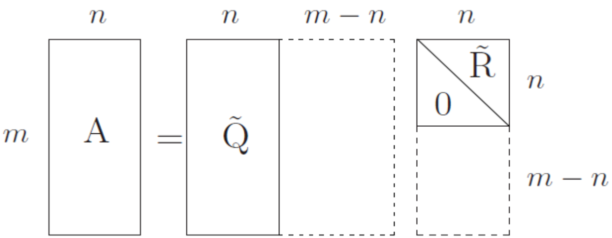

Figure: $A = QR$, $A = \tilde{Q}\tilde{R}$ - reduced QR factorization. From Quarteroni et al.

Theorem 12.8

- $A \in \mathcal{R}^{n \times m}$
- $\boldsymbol{rank}(A) = m$

QR factorization

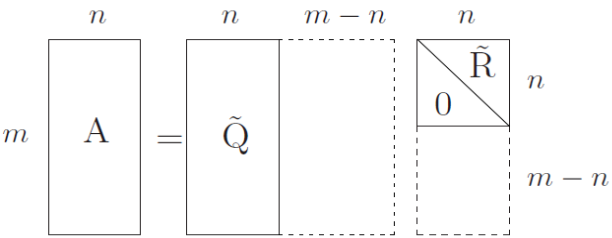

Figure: $A = QR$, $A = \tilde{Q}\tilde{R}$ - reduced QR factorization. From Quarteroni et al.

Theorem 12.8

- $A \in \mathcal{R}^{n \times m}$
- $\boldsymbol{rank}(A) = m$
- $\exists QR$ *factorization*

QR factorization

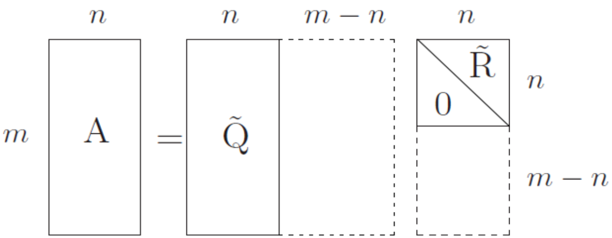

Figure: $A = QR$, $A = \tilde{Q}\tilde{R}$ - reduced QR factorization. From Quarteroni et al.

Theorem 12.8

- $A \in \mathcal{R}^{n\times m}$
- $\textbf{\textit{rank}}(A) = m$
- $\exists QR$ *factorization*
- $\Downarrow$

QR factorization

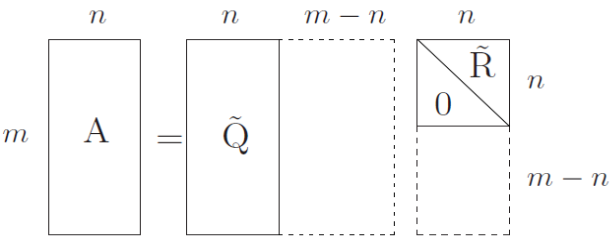

Figure: $A = QR$, $A = \tilde{Q}\tilde{R}$ - reduced QR factorization. From Quarteroni et al.

Theorem 12.8

- $A \in \mathcal{R}^{n \times m}$
- $\textbf{rank}(A) = m$
- $\exists QR$ *factorization*
- $\Downarrow$
- $\exists$ *reduced* QR *factorization* $A = \tilde{Q}\tilde{R}$

QR factorization

Example 12.9

```
A= [[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
Q= [[-0.07761505 -0.83305216  0.53358462]
 [-0.31046021 -0.45123659 -0.8036038 ]
 [-0.54330537 -0.06942101  0.00645373]
 [-0.77615053  0.31239456  0.26356544]]
R= [[-1.28840987e+01 -1.45916299e+01 -1.62991610e+01]
 [ 0.00000000e+00 -1.04131520e+00 -2.08263040e+00]
 [ 0.00000000e+00  0.00000000e+00 -3.39618744e-15]]
```

Figure: Reduced QR factorization

QR factorization

Example 12.9

```
A= [[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
Q= [[-0.07761505 -0.83305216  0.53358462]
 [-0.31046021 -0.45123659 -0.8036038 ]
 [-0.54330537 -0.06942101  0.00645373]
 [-0.77615053  0.31239456  0.26356544]]
R= [[-1.28840987e+01 -1.45916299e+01 -1.62991610e+01]
 [ 0.00000000e+00 -1.04131520e+00 -2.08263040e+00]
 [ 0.00000000e+00  0.00000000e+00 -3.39618744e-15]]
```

Figure: Reduced QR factorization

- A - tall matrix

QR factorization

Example 12.9

```
A= [[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
Q= [[-0.07761505 -0.83305216  0.53358462]
 [-0.31046021 -0.45123659 -0.8036038 ]
 [-0.54330537 -0.06942101  0.00645373]
 [-0.77615053  0.31239456  0.26356544]]
R= [[-1.28840987e+01 -1.45916299e+01 -1.62991610e+01]
 [ 0.00000000e+00 -1.04131520e+00 -2.08263040e+00]
 [ 0.00000000e+00  0.00000000e+00 -3.39618744e-15]]
```

Figure: Reduced QR factorization

- A - tall matrix
- Q - tall matrix, orthogonal

QR factorization

Example 12.9

```
A= [[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
Q= [[-0.07761505 -0.83305216  0.53358462]
 [-0.31046021 -0.45123659 -0.8036038 ]
 [-0.54330537 -0.06942101  0.00645373]
 [-0.77615053  0.31239456  0.26356544]]
R= [[-1.28840987e+01 -1.45916299e+01 -1.62991610e+01]
 [ 0.00000000e+00 -1.04131520e+00 -2.08263040e+00]
 [ 0.00000000e+00  0.00000000e+00 -3.39618744e-15]]
```

Figure: Reduced QR factorization

- A - tall matrix
- Q - tall matrix, orthogonal
- R - square matrix, upper triangular

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n\times k}$
- A is tall matrix, $n > k$

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n \times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \ a_2 \ ...a_k)$, a_i - column of A

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n \times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \ a_2 \ ...a_k)$, a_i - column of A
- a_i is n-vector

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n \times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \; a_2 \; ...a_k)$, a_i - column of A
- a_i is n-vector
- Output: $Q = (q_{ij})_{n \times k}$, $R = (r_{ij})_{k \times k}$

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n\times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \; a_2 \; ...a_k)$, a_i - column of A
- a_i is n-vector
- Output: $Q = (q_{ij})_{n\times k}$, $R = (r_{ij})_{k\times k}$
- Q is tall matrix, $n > k$

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n \times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \; a_2 \; ... a_k)$, a_i - column of A
- a_i is n-vector
- Output: $Q = (q_{ij})_{n \times k}$, $R = (r_{ij})_{k \times k}$
- Q is tall matrix, $n > k$
- $Q = (q_1 \; q_2 \; ... q_k)$, q_i - column of Q

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n \times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \; a_2 \; ... a_k)$, a_i - column of A
- a_i is n-vector
- Output: $Q = (q_{ij})_{n \times k}$, $R = (r_{ij})_{k \times k}$
- Q is tall matrix, $n > k$
- $Q = (q_1 \; q_2 \; ... q_k)$, q_i - column of Q
- q_i is n-vector

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n \times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \ a_2 \ ...a_k)$, a_i - column of A
- a_i is n-vector
- Output: $Q = (q_{ij})_{n \times k}$, $R = (r_{ij})_{k \times k}$
- Q is tall matrix, $n > k$
- $Q = (q_1 \ q_2 \ ...q_k)$, q_i - column of Q
- q_i is n-vector
- $q_i \perp q_j, i \neq j$, $\|q_i\| = 1, i, j = 1, 2, ..., k \Rightarrow Q^T Q = I$,

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n \times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \ a_2 \ ...a_k)$, a_i - column of A
- a_i is n-vector
- Output: $Q = (q_{ij})_{n \times k}$, $R = (r_{ij})_{k \times k}$
- Q is tall matrix, $n > k$
- $Q = (q_1 \ q_2 \ ...q_k)$, q_i - column of Q
- q_i is n-vector
- $q_i \perp q_j, i \neq j$, $\|q_i\| = 1, i, j = 1, 2, ..., k \Rightarrow Q^T Q = I$,
- R is upper triangular matrix

Reduced QR factorization, 1

- Input: $A = (a_{ij})_{n \times k}$
- A is tall matrix, $n > k$
- $A = (a_1 \ a_2 \ ...a_k)$, a_i - column of A
- a_i is n-vector
- Output: $Q = (q_{ij})_{n \times k}$, $R = (r_{ij})_{k \times k}$
- Q is tall matrix, $n > k$
- $Q = (q_1 \ q_2 \ ...q_k)$, q_i - column of Q
- q_i is n-vector
- $q_i \perp q_j, i \neq j$, $\|q_i\| = 1, i, j = 1, 2, ..., k \Rightarrow Q^T Q = I$,
- R is upper triangular matrix
- $$A = QR$$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1\ a_2\ ...a_k$ yields $q_1\ q_2\ ...q_k$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to a_1 a_2 ...a_k yields q_1 q_2 ...q_k
- Gramm-Schmidt algorithm

Reduced QR factorization, 2

- Applying Gramm-Schmidt to a_1 a_2 ...a_k yields q_1 q_2 ...q_k
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1 \ a_2 \ ...a_k$ yields $q_1 \ q_2 \ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output:n-vectors $y_1, y_2, ..., y_m$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1\ a_2\ ...a_k$ yields $q_1\ q_2\ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1\ a_2\ ...a_k$ yields $q_1\ q_2\ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(., .)$
  - for $i = 1, 2, ..., m$ do:

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1 \ a_2 \ ...a_k$ yields $q_1 \ q_2 \ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j, \ i \neq j, \ i, j = 1, 2, ..., m$, inner product $(., .)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1 \ a_2 \ ...a_k$ yields $q_1 \ q_2 \ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output: n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j, \ i \neq j, \ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1} (y_j, x_i) y_j$
    2. if $\tilde{y}_i = 0$, terminate

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1 \ a_2 \ ...a_k$ yields $q_1 \ q_2 \ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j, \ i \neq j, \ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1\ a_2 \ldots a_k$ yields $q_1\ q_2 \ldots q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i,j = 1,2,...,m$, inner product $(.,.)$
  - for $i = 1,2,...,m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1\ a_2\ ...a_k$ yields $q_1\ q_2\ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output: n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$
- $\tilde{q}_i = a_i - \sum_{j=1}^{i-1}(q_j, a_i)q_j$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1\ a_2\ ...a_k$ yields $q_1\ q_2\ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output: n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$
- $\tilde{q}_i = a_i - \sum_{j=1}^{i-1}(q_j, a_i)q_j$
- $q_i = \frac{\tilde{q}_i}{\|\tilde{q}_i\|}$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1\ a_2\ ...a_k$ yields $q_1\ q_2\ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output:n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i,j = 1,2,...,m$, inner product $(.,.)$
  - for $i = 1,2,...,m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$
- $\tilde{q}_i = a_i - \sum_{j=1}^{i-1}(q_j, a_i)q_j$
- $q_i = \frac{\tilde{q}_i}{\|\tilde{q}_i\|}$
- $a_i = \sum_{j=1}^{i-1}(q_j, a_i)q_j + \|\tilde{q}_i\|q_i$

Reduced QR factorization, 2

- Applying Gramm-Schmidt to $a_1\ a_2\ ...a_k$ yields $q_1\ q_2\ ...q_k$
- Gramm-Schmidt algorithm
  - Input: n-vectors $x_1, x_2, ..., x_m$
  - Output: n-vectors $y_1, y_2, ..., y_m$
  - $y_i \perp y_j,\ i \neq j,\ i, j = 1, 2, ..., m$, inner product $(.,.)$
  - for $i = 1, 2, ..., m$ do:
    1. $\tilde{y}_i = x_i - \sum_{j=1}^{i-1}(y_j, x_i)y_j$
    2. if $\tilde{y}_i = 0$, terminate
    3. $y_i = \frac{\tilde{y}_i}{\|\tilde{y}_i\|}$
- $\tilde{q}_i = a_i - \sum_{j=1}^{i-1}(q_j, a_i)q_j$
- $q_i = \frac{\tilde{q}_i}{\|\tilde{q}_i\|}$
- $a_i = \sum_{j=1}^{i-1}(q_j, a_i)q_j + \|\tilde{q}_i\|q_i$
- $r_{ij} = \begin{cases} i < j : r_{ij} = (a_j, q_i) \\ i = j : r_{ii} = \|\tilde{q}_i\| \\ i > j : r_{ij} = 0 \end{cases}$

Reduced QR factorization, 3

- $a_i = \sum_{j=1}^{i-1}(q_j, a_i)q_j + \|\tilde{q}_i\| q_i$
  - $r_{ij} = \begin{cases} i < j : r_{ij} = (a_j, q_i) \\ i = j : r_{ii} = \|\tilde{q}_i\| \\ i > j : r_{ij} = 0 \end{cases}$,

Reduced QR factorization, 3

- $a_i = \sum_{j=1}^{i-1}(q_j, a_i)q_j + \|\tilde{q}_i\|q_i$
  - $r_{ij} = \begin{cases} i < j : r_{ij} = (a_j, q_i) \\ i = j : r_{ii} = \|\tilde{q}_i\| \\ i > j : r_{ij} = 0 \end{cases}, \; A = QR$

Reduced QR factorization, 3

- $a_i = \sum_{j=1}^{i-1}(q_j, a_i)q_j + \|\tilde{q}_i\| q_i$
- $r_{ij} = \begin{cases} i < j : r_{ij} = (a_j, q_i) \\ i = j : r_{ii} = \|\tilde{q}_i\| \\ i > j : r_{ij} = 0 \end{cases}, \quad A = QR$
-

$$a_1 = r_{11} q_1 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{11} \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

Reduced QR factorization, 3

- $a_i = \sum_{j=1}^{i-1} (q_j, a_i) q_j + \|\tilde{q}_i\| q_i$
- $r_{ij} = \begin{cases} i < j : r_{ij} = (a_j, q_i) \\ i = j : r_{ii} = \|\tilde{q}_i\| \\ i > j : r_{ij} = 0 \end{cases}, \; A = QR$
-
$$a_1 = r_{11} q_1 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{11} \\ 0 \\ \dots \\ 0 \end{pmatrix}$$
-
$$a_2 = r_{12} q_1 + r_{22} q_2 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{12} \\ r_{22} \\ \dots \\ 0 \end{pmatrix}$$

Reduced QR factorization, 3

- $a_i = \sum_{j=1}^{i-1}(q_j, a_i)q_j + \|\tilde{q}_i\|q_i$
- $r_{ij} = \begin{cases} i < j : r_{ij} = (a_j, q_i) \\ i = j : r_{ii} = \|\tilde{q}_i\| \\ i > j : r_{ij} = 0 \end{cases}, \ A = QR$
- $$a_1 = r_{11}q_1 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{11} \\ 0 \\ \dots \\ 0 \end{pmatrix}$$
- $$a_2 = r_{12}q_1 + r_{22}q_2 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{12} \\ r_{22} \\ \dots \\ 0 \end{pmatrix}$$
- $$a_k = r_{1k}q_1 + r_{2k}q_2 + \dots + r_{kk}q_k = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{1k} \\ r_{2k} \\ \dots \\ r_{kk} \end{pmatrix}$$

Reduced QR factorization, 4

- $$a_1 = r_{11}q_1 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{11} \\ 0 \\ \dots \\ 0 \end{pmatrix}$$
- $$a_2 = r_{12}q_1 + r_{22}q_2 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{12} \\ r_{22} \\ \dots \\ 0 \end{pmatrix}$$
- $$a_k = r_{1k}q_1 + r_{2k}q_2 + \dots + r_{kk}q_k = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{1k} \\ r_{2k} \\ \dots \\ r_{kk} \end{pmatrix}$$

Reduced QR factorization, 4

- $$a_1 = r_{11} q_1 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{11} \\ 0 \\ \dots \\ 0 \end{pmatrix}$$
- $$a_2 = r_{12} q_1 + r_{22} q_2 = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{12} \\ r_{22} \\ \dots \\ 0 \end{pmatrix}$$
- $$a_k = r_{1k} q_1 + r_{2k} q_2 + \dots + r_{kk} q_k = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{1k} \\ r_{2k} \\ \dots \\ r_{kk} \end{pmatrix}$$
- $$\begin{pmatrix} a_1 & a_2 & \dots & a_k \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1k} \\ 0 & r_{22} & \dots & r_{1k} \\ . & . & \dots & . \\ 0 & 0 & \dots & r_{kk} \end{pmatrix}$$

Reduced QR factorization, 5

- $$\begin{pmatrix} a_1 & a_2 & \dots & a_k \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1k} \\ 0 & r_{22} & \dots & r_{1k} \\ . & . & \dots & . \\ 0 & 0 & \dots & r_{kk} \end{pmatrix}$$
- $$A = QR$$
- Q: is QR factorization always possible?

Reduced QR factorization, 5

- $$\begin{pmatrix} a_1 & a_2 & \dots & a_k \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & \dots & q_k \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1k} \\ 0 & r_{22} & \dots & r_{1k} \\ . & . & \dots & . \\ 0 & 0 & \dots & r_{kk} \end{pmatrix}$$
- $$A = QR$$
- Q: is QR factorization always possible?
- A:

Theorem 12.10

For $A \in \mathcal{R}^{n\times m}$ always exist $Q \in \mathcal{R}^{n\times m}$ with orthonormal columns and upper triangular $R \in \mathcal{R}^{m\times m}$ such that $A = QR$

Reduced QR factorization

```
A= [[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
Q= [[-0.07761505 -0.83305216  0.53358462]
 [-0.31046021 -0.45123659 -0.8036038 ]
 [-0.54330537 -0.06942101  0.00645373]
 [-0.77615053  0.31239456  0.26356544]]
R= [[-1.28840987e+01 -1.45916299e+01 -1.62991610e+01]
 [ 0.00000000e+00 -1.04131520e+00 -2.08263040e+00]
 [ 0.00000000e+00  0.00000000e+00 -3.39618744e-15]]
A-QR= [[-1.33226763e-15 -7.10542736e-15 -5.32907052e-15]
 [-8.88178420e-16 -1.77635684e-15 -8.88178420e-16]
 [ 0.00000000e+00 -3.55271368e-15 -3.55271368e-15]
 [-1.77635684e-15 -5.32907052e-15 -3.55271368e-15]]
Pseudo inverse =  [[-0.48333333 -0.24444444 -0.00555556  0.23333333]
 [-0.03333333 -0.01111111  0.01111111  0.03333333]
 [ 0.41666667  0.22222222  0.02777778 -0.16666667]]
```

Figure: Pseudo inverse, $A^{\dagger} = R^{-1}Q^{T}$

Q & A