Hovedemnet her er bestemmelsen af egenløsninger for en matrix **A**(nxn). Fokus er på de to problemer og tilhørende metoder a) og b):

a) ***Problem***: Givet **A**(nxn), bestem den numerisk største egenværdi λ_1 og dens egenvektor $\mathbf{v_1}$.
   Metode: Potensmetoden.
b) ***Problem***: Givet en symmetrisk **A**(nxn), bestem egenløsningerne $\{\lambda_k , \mathbf{v_k}\}$ hvor k = 1, 2,.., n.
   Metode: Potensmetoden udvidet. Potensmetoden anvendes på de n matricer: $\mathbf{P_1}$, $\mathbf{P_2}$,..., $\mathbf{P_n}$.
   Her er $\mathbf{P_k}$ dannet, så λ_k er den numerisk største egenværdi for $\mathbf{P_k}$ hvor k = 1, 2, … , n.

Potensmetoden, forudsætninger for konvergens af potensmetoden og algoritme

Givet matricen **A**(nxn) og vektoren $\mathbf{z_0}$(nx1), som er en brugervalgt startvektor for algoritmen. Potensmetoden vil finde egenløsningen $\{\lambda_1 , \mathbf{v_1}\}$ hvis følgende forudsætninger er opfyldt:

1. **A**(nxn) har egenløsningerne $\{\lambda_k, \mathbf{v_k}\}$ hvor k = 1, 2, 3, … , n.
2. $\{\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \dots, \mathbf{v_{n-1}}, \mathbf{v_n}\}$ er lineært uafhængige og er dermed en basis for $\mathbb{R}^n$
3. $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq |\lambda_4| \geq \dots \geq |\lambda_{n-1}| \geq |\lambda_n|$. Bemærk den skarpe ulighed her: $|\lambda_1| > |\lambda_2|$.
   Dette skal forstås således:
   Egenløsningerne er navngivet, så de numeriske værdier af egenværdierne er aftagende hen gennem sekvensen $\{\lambda_1 , \mathbf{v_1}\}$, $\{\lambda_2 , \mathbf{v_2}\}$,..., $\{\lambda_n , \mathbf{v_n}\}$. Og: Den numeriske værdi $|\lambda_1|$ af den numerisk største egenværdi λ_1 er ***større end*** den numeriske værdi $|\lambda_2|$ af den numerisk næststørste egenværdi λ_2.
4. Opskrives $\mathbf{z_0}$ i basis $\{\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \dots, \mathbf{v_{n-1}}, \mathbf{v_n}\}$:
   $\mathbf{z_0} = \alpha_1 \cdot \mathbf{v_1} + \alpha_2 \cdot \mathbf{v_2} + \alpha_3 \cdot \mathbf{v_3} + \dots + \alpha_{n-1} \cdot \mathbf{v_{n-1}} + \alpha_n \cdot \mathbf{v_n}$ skal følgende være opfyldt: $\alpha_1 \neq 0$.

Algoritme for potensmetoden

Initialisering

Vælg: $\mathbf{z_0}$ startvektoren for iterationen.
ε tolerancen for iterationens stopkriterium.
N den øvre grænse for antallet af tilladte iterationer. (Nødbremse!)

Sæt: i = 0 iterationstælleren 0-stilles.

Find: L_0 den numerisk største koordinat i $\mathbf{z_0}$

Beregn: $\mathbf{y_0} = \frac{1}{L_0} \mathbf{z_0}$

Iteration i = 1, 2, 3, ...

Givet: $\mathbf{y_{i-1}}$

Beregn: $\mathbf{z_i} = \mathbf{A} \cdot \mathbf{y_{i-1}}$

Find: L_i den numerisk største koordinat i $\mathbf{z_i}$

Beregn: $\mathbf{y_i} = \frac{1}{L_i} \mathbf{z_i}$

Stopkriterium

Stop hvis: $|L_i - L_{i-1}| < \varepsilon \quad \vee \quad |\mathbf{y_i} - \mathbf{y_{i-1}}| < \varepsilon \quad \vee \quad i = N$

Løsningskriterium

Ved terminering: $i < N \Rightarrow \mathbf{y_i} \approx \mathbf{v_1} \quad \wedge \quad L_i \approx \lambda_1$

Overvejelse af stopkriteriet for potensmetoden. Formålet med overvejelsen er at bekræfte eller afkræfte formodningen om, at I er bedre end II:

- I: Stop hvis $|\mathbf{y_i} - \mathbf{y_{i-1}}| < \varepsilon \quad \vee \quad i = N$
- II: Stop hvis $|L_i - L_{i-1}| < \varepsilon \quad \vee \quad |\mathbf{y_i} - \mathbf{y_{i-1}}| < \varepsilon \quad \vee \quad i = N$

Note 1. Potensmetoden i matematisk fremstilling.
Hvis de ovenstående betingelser 1, 2, 3 og 4 er opfyldt, anvend følgende
Algoritme:

Ud fra startvektoren

$\mathbf{z_0}$

som normeres således:

Find: L_0 den numerisk største koordinat i $\mathbf{z_0}$

Beregn: $\mathbf{y_0} = \frac{1}{L_0} \cdot \mathbf{z_0}$

Udfør iterationerne i = 1, 2, 3, ….

Givet: $\mathbf{y_{i-1}}$

Beregn: $\mathbf{z_i} = \mathbf{A} \cdot \mathbf{y_{i-1}}$

Find: L_i den numerisk største koordinat i $\mathbf{z_i}$

Beregn: $\mathbf{y_i} = \frac{1}{L_i}\mathbf{z_i}$

Da fås konvergenserne:

$L_i \rightarrow \lambda_1$ for $i \rightarrow \infty$

$\mathbf{y_i} \rightarrow \mathbf{v_1}$ for $i \rightarrow \infty$

Note 2. Vektornormering i potensmetoden udføres i ∞-normen og ***ikke*** i 2-normen.

Med $\mathbf{z_i} = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$ fås $L_i = 5$ og $\mathbf{y_i} = \frac{1}{L_i}\mathbf{z_i} = \begin{bmatrix} 1 \\ 0.2 \\ 0.4 \end{bmatrix}$. Bemærk: $\mathbf{y_i}^{\mathbf{T}}\mathbf{y_i} = |\mathbf{y_i}|^2 \neq 1$

Med $\mathbf{z_i} = \begin{bmatrix} 4 \\ 0 \\ -5 \end{bmatrix}$ fås $L_i = -5$ og $\mathbf{y_i} = \frac{1}{L_i}\mathbf{z_i} = \begin{bmatrix} -0.8 \\ 0 \\ 1 \end{bmatrix}$. Bemærk: $\mathbf{y_i}^{\mathbf{T}}\mathbf{y_i} = |\mathbf{y_i}|^2 \neq 1$

Note 3. Eksempel

$$\mathbf{A}_1 = \begin{bmatrix} 3 & 0.5 & 0 & 0.5 \\ 0.5 & 2 & 0.5 & 1 \\ 0 & 0.5 & 3 & 0.5 \\ 0.5 & 1 & 0.5 & 2 \end{bmatrix}$$ har egenløsningerne

$$\{\lambda_1, \mathbf{v_1}\} = \{4, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}, \quad \{\lambda_2, \mathbf{v_2}\} = \{3, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\}, \quad \{\lambda_3, \mathbf{v_3}\} = \{2, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\} \text{ og } \{\lambda_4, \mathbf{v_4}\} = \{1, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}$$

Med $\mathbf{z_0} = \begin{bmatrix} 3 \\ 1 \\ 1 \\ -1 \end{bmatrix}$ fås konvergenserne: $L_i \rightarrow 4$ for $i \rightarrow \infty$ og $\mathbf{y_i} \rightarrow \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ for $i \rightarrow \infty$

Bemærk at betingelserne 1, 2, 3 og 4 for konvergens er opfyldt.

Note 4.
Vedr. konvergensforudsætning 4, her i to eksempler a) og b) begge med n = 4.

Med opskrivning af $\mathbf{z_0}$ i basis $\{\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4}\}$, dvs $\mathbf{z_0} = \alpha_1 \cdot \mathbf{v_1} + \alpha_2 \cdot \mathbf{v_2} + \alpha_3 \cdot \mathbf{v_3} + \alpha_4 \cdot \mathbf{v_4}$ gælder:
Potensmetodens konvergens mod egenløsningen $\{\lambda_1, \mathbf{v_1}\}$ forudsætter/kræver $\alpha_1 \neq 0$.

Eksempel a)

Med $\mathbf{z_0} = \mathbf{v_1} + \mathbf{v_2} + \mathbf{v_3} + \mathbf{v_4} = \begin{bmatrix} 3 \\ 1 \\ 1 \\ -1 \end{bmatrix}$ hvor $\alpha_1 = 1$ er konvergensbetingelse 4 opfyldt.

Konvergens: $L_i \rightarrow \lambda_1 = 4$ for $i \rightarrow \infty$ og $\mathbf{y_i} \rightarrow \mathbf{v_1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ for $i \rightarrow \infty$

Eksempel b)

Med $\mathbf{z_0} = \mathbf{v_2} + \mathbf{v_3} + \mathbf{v_4} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ -2 \end{bmatrix}$ hvor $\alpha_1 = 0$ er konvergensbetingelse 4 ikke opfyldt.

Konvergens: $L_i \rightarrow \lambda_2 = 3$ for $i \rightarrow \infty$ og $\mathbf{y_i} \rightarrow \mathbf{v_2} = \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$ for $i \rightarrow \infty$

Fordi: $\mathbf{z_0} \in \text{Span}(\mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4})$ = > $\boldsymbol{A^i z_0} \in \text{Span}(\mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4})$, og i ***underrummet*** $\text{Span}(\mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4})$ er $\lambda_2 = 3$ den numerisk største egenværdi, som potensmetoden derfor vil konvergere imod.

Note 5. Eksempel

$$A_2 = \begin{bmatrix} 2.25 & 0.75 & 0.25 & 0.75 \\ 0.75 & 3.25 & 0.75 & -0.75 \\ 0.25 & 0.75 & 2.25 & 0.75 \\ 0.75 & -0.75 & 0.75 & 3.25 \end{bmatrix}$$ har egenløsningerne

$\{\lambda_1, \mathbf{v_1}\} = \{4, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}$, $\{\lambda_2, \mathbf{v_2}\} = \{4, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}$, $\{\lambda_3, \mathbf{v_3}\} = \{2, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\}$ og $\{\lambda_4, \mathbf{v_4}\} = \{1, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\}$

Bemærk: $\lambda_1 = \lambda_2 = 4$ = > Betingelse 3 for konvergens er ikke opfyldt.

<u>Transformationer af egenværdispektret</u>

Givet matricen **A(nxn)** med egenløsningerne $\{\lambda_k, \mathbf{v_k}\}$ hvor k = 1, 2, … , n.
Lad c være en skalar m være et positivt heltal.

Dermed har vi følgende transformationer:

Matrix $\mathbf{B_1} = c \cdot \mathbf{A}$ har egenløsningerne: $\{c \cdot \lambda_k, \mathbf{v_k}\}$ hvor k = 1, 2, … , n.

Matrix $\mathbf{B_2} = \mathbf{A} - c\cdot\mathbf{I}$ har egenløsningerne: $\{\lambda_k - c, \mathbf{v_k}\}$ hvor k = 1, 2, … , n.

Matrix $\mathbf{B_3} = \mathbf{A}^{-1}$ har egenløsningerne: $\{\frac{1}{\lambda_k}, \mathbf{v_k}\}$ hvor k = 1, 2, … , n.

Matrix $\mathbf{B_4} = \mathbf{A^m}$ har egenløsningerne: $\{\lambda_k^m, \mathbf{v_k}\}$ hvor k = 1, 2, … , n.

Matrix $\mathbf{B_5} = (\mathbf{A} - c\cdot\mathbf{I})^{-1}$ har egenløsningerne: $\{\frac{1}{\lambda_k - c}, \mathbf{v_k}\}$ hvor k = 1, 2, … , n.

Eksempel

Ovenfor er givet matricen $\mathbf{A_1}$(4x4) har egenløsningerne $\{\lambda_k, \mathbf{v_k}\}$ hvor k = 1, 2, 3, 4.
Dan matrix $\mathbf{A_3} = \mathbf{A_1} - c\cdot\mathbf{I}$ med $c = 6$ og find således:
$\mathbf{A_3} = \mathbf{A_1} - 6\cdot\mathbf{I}$ har egenløsningerne: $\{\lambda_k - 6, \mathbf{v_k}\}$ hvor k = 1, 2, 3, 4.

$$\mathbf{A_1} = \begin{bmatrix} 3 & 0.5 & 0 & 0.5 \\ 0.5 & 2 & 0.5 & 1 \\ 0 & 0.5 & 3 & 0.5 \\ 0.5 & 1 & 0.5 & 2 \end{bmatrix} \Rightarrow \mathbf{A_3} = \mathbf{A_1} - 6\cdot\mathbf{I} = \begin{bmatrix} -3 & 0.5 & 0 & 0.5 \\ 0.5 & -4 & 0.5 & 1 \\ 0 & 0.5 & -3 & 0.5 \\ 0.5 & 1 & 0.5 & -4 \end{bmatrix}$$ som har egenløsningerne:

$$\{\lambda_4 - 6, \mathbf{v_4}\} = \{-5, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}, \{\lambda_3 - 6, \mathbf{v_3}\} = \{-4, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\}, \{\lambda_2 - 6, \mathbf{v_2}\} = \{-3, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\} \text{ og } \{\lambda_1 - 6, \mathbf{v_1}\} = \{-2, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}$$

Anvendelse af potensmetoden:

Med $\mathbf{z_0} = \begin{bmatrix} 3 \\ 1 \\ 1 \\ -1 \end{bmatrix}$ fås konvergenserne: $L_i \rightarrow -5$ for $i \rightarrow \infty$ og $\mathbf{y_i} \rightarrow \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$ for $i \rightarrow \infty$

Bemærk at betingelserne 1, 2, 3 og 4 for konvergens er opfyldt.

Potensmetoden anvendt på en symmetrisk A(nxn**) med n distinkte egenværdier.**

Bestemmelse af egenløsningerne $\{\lambda_k, \mathbf{v_k}\}$ hvor k = 1, 2,.., n ***til en symmetrisk matrix* A(**nxn**).**
A(nxn**)** har dermed n reelle egenværdier λ_k og n ortogonale egenvektorer $\mathbf{v_k}$ hvor k = 1, 2,..., n.
Det forudsættes her, at: $|\lambda_1| > |\lambda_2| > |\lambda_3| > |\lambda_4| > \ldots > |\lambda_{n-1}| > |\lambda_n|$.
Dermed er egenværdierne distinkte, dvs den algebraiske multiplicitet = 1 for samtlige λ_k k = 1, 2,.., n.

Metoden har udgangspunkt i spektralfremstillingen $\mathbf{A} = \mathbf{S}\cdot\mathbf{\Lambda}\cdot\mathbf{S}^{-1}$ hvor:
Søjlerne i **S(**nxn**)** er de indbyrdes ***ortogonale*** egenvektorer $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \ldots, \mathbf{v_{n-1}}, \mathbf{v_n}$.
Diagonalelementerne i diagonalmatricen **Λ(**nxn**)** er egenværdierne $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{n-1}, \lambda_n$.
Hvis egenvektorerne $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \ldots, \mathbf{v_{n-1}}, \mathbf{v_n}$ vælges som ***enhedsvektorer*** fås: $\mathbf{S^{-1}} = \mathbf{S^T}$ og dermed
$\mathbf{A} = \mathbf{S}\cdot\mathbf{\Lambda}\cdot\mathbf{S}^{-1} = \mathbf{S}\cdot\mathbf{\Lambda}\cdot\mathbf{S^T} = \lambda_1\mathbf{v_1}\mathbf{v_1^T} + \lambda_2\mathbf{v_2}\mathbf{v_2^T} + \lambda_3\mathbf{v_3}\mathbf{v_3^T} + \ldots + \lambda_k\mathbf{v_k}\mathbf{v_k^T} + \ldots + \lambda_{n-1}\mathbf{v_{n-1}}\mathbf{v_{n-1}^T} + \lambda_n\mathbf{v_n}\mathbf{v_n^T}$

Potensmetoden bestemmer kun én egenløsning, nemlig matricens numerisk største egenværdi og dens tilhørende egenvektor.
Den følgende metode bestemmer samtlige egenløsninger $\{\lambda_k, \mathbf{v_k}\}$ k = 1, 2,.., n for **A(**nxn**)** ved at anvende potensmetoden på hver af de n matricer $\mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}, \mathbf{P_4}, \ldots, \mathbf{P_n}$ som præsenteres nedenfor.

Metoden har udgangspunkt i spektralfremstillingen, hvor $\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \ldots, \mathbf{v_{n-1}}, \mathbf{v_n}$ er ***ortho-normale***:
$\mathbf{A} = \lambda_1\mathbf{v_1}\mathbf{v_1^T} + \lambda_2\mathbf{v_2}\mathbf{v_2^T} + \lambda_3\mathbf{v_3}\mathbf{v_3^T} + \ldots + \lambda_k\mathbf{v_k}\mathbf{v_k^T} + \ldots + \lambda_{n-1}\mathbf{v_{n-1}}\mathbf{v_{n-1}^T} + \lambda_n\mathbf{v_n}\mathbf{v_n^T}$

Potensmetoden anvendes nu på de n matricer $\mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}, \mathbf{P_4}, \dots, \mathbf{P_n}$ hvor

$\mathbf{P_1} = \mathbf{A}$,

$\mathbf{P_2} = \mathbf{A} - \lambda_1\mathbf{v_1}\mathbf{v_1}^\mathbf{T}$,

$\mathbf{P_3} = \mathbf{A} - (\lambda_1\mathbf{v_1}\mathbf{v_1}^\mathbf{T} + \lambda_2\mathbf{v_2}\mathbf{v_2}^\mathbf{T})$,

$\mathbf{P_4} = \mathbf{A} - (\lambda_1\mathbf{v_1}\mathbf{v_1}^\mathbf{T} + \lambda_2\mathbf{v_2}\mathbf{v_2}^\mathbf{T} + \lambda_3\mathbf{v_3}\mathbf{v_3}^\mathbf{T})$,

$\vdots$

$\mathbf{P_n} = \mathbf{A} - (\lambda_1\mathbf{v_1}\mathbf{v_1}^\mathbf{T} + \lambda_2\mathbf{v_2}\mathbf{v_2}^\mathbf{T} + \lambda_3\mathbf{v_3}\mathbf{v_3}^\mathbf{T} + \dots + \lambda_{n-1}\mathbf{v_{n-1}}\mathbf{v_{n-1}^T})$

Konklusion:

Anvendt på $\mathbf{P_1}$ finder potensmetoden egenløsningen $\{\lambda_1, \mathbf{w_1}\}$. Her er $\mathbf{w_1} = |\mathbf{w_1}|\cdot\mathbf{v_1}$ og $|\mathbf{w_1}| \neq 1$

Anvendt på $\mathbf{P_2}$ finder potensmetoden egenløsningen $\{\lambda_2, \mathbf{w_2}\}$. Her er $\mathbf{w_2} = |\mathbf{w_2}|\cdot\mathbf{v_2}$ og $|\mathbf{w_2}| \neq 1$

Anvendt på $\mathbf{P_3}$ finder potensmetoden egenløsningen $\{\lambda_3, \mathbf{w_3}\}$. Her er $\mathbf{w_3} = |\mathbf{w_3}|\cdot\mathbf{v_3}$ og $|\mathbf{w_3}| \neq 1$

Anvendt på $\mathbf{P_4}$ finder potensmetoden egenløsningen $\{\lambda_4, \mathbf{w_4}\}$. Her er $\mathbf{w_4} = |\mathbf{w_4}|\cdot\mathbf{v_4}$ og $|\mathbf{w_4}| \neq 1$

:

Anvendt på $\mathbf{P_n}$ finder potensmetoden egenløsningen $\{\lambda_n, \mathbf{w_n}\}$. Her er $\mathbf{w_n} = |\mathbf{w_n}|\cdot\mathbf{v_n}$ og $|\mathbf{w_n}| \neq 1$

Konsekvens: (Bemærk, $\mathbf{w_k}$ er frembragt ved potensmetodenormering, dvs i $\infty -$ normen!)

Hver gang potensmetoden finder en egenløsning $\{\lambda_k, \mathbf{w_k}\}$ skal $\mathbf{v_k}$ beregnes ved at normere $\mathbf{w_k}$:

Principielt således: $\mathbf{v_k} = \dfrac{\mathbf{w_k}}{|\mathbf{w_k}|}$ => det ***implementeres*** således: $\lambda_k\mathbf{v_k}\mathbf{v_k}^\mathbf{T} = \lambda_k \dfrac{1}{\mathbf{w_k^T w_k}} \mathbf{w_k}\mathbf{w_k}^\mathbf{T}$

Eksempel:

$$A = \begin{bmatrix} 3 & 0.5 & 0 & 0.5 \\ 0.5 & 2 & 0.5 & 1 \\ 0 & 0.5 & 3 & 0.5 \\ 0.5 & 1 & 0.5 & 2 \end{bmatrix} \quad \text{har egenløsningerne}$$

$$\{\lambda_1, \mathbf{w_1}\} = \{4, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}, \quad \{\lambda_2, \mathbf{w_2}\} = \{3, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\}, \quad \{\lambda_3, \mathbf{w_3}\} = \{2, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\} \quad \text{og} \quad \{\lambda_4, \mathbf{w_4}\} = \{1, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}$$

Altså:

$$\mathbf{A} = \mathbf{P_1} = \begin{bmatrix} 3 & 0.5 & 0 & 0.5 \\ 0.5 & 2 & 0.5 & 1 \\ 0 & 0.5 & 3 & 0.5 \\ 0.5 & 1 & 0.5 & 2 \end{bmatrix} \quad \text{og potensmetoden finder:} \quad \{\lambda_1, \mathbf{w_1}\} = \{4, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}$$

$$=> \quad |\mathbf{w_1}| = 2 \quad => \quad \mathbf{v_1} = \frac{1}{2}\cdot\mathbf{w_1} \quad => \quad \lambda_1\mathbf{v_1}\mathbf{v_1}^\mathbf{T} = \lambda_1 \frac{1}{\mathbf{w_1^T w_1}} \mathbf{w_1}\mathbf{w_1}^\mathbf{T} = 4\cdot\frac{1}{4}\cdot\mathbf{w_1}\mathbf{w_1}^\mathbf{T} \quad =>$$

$$\mathbf{P_2} = \mathbf{A} - \lambda_1\mathbf{v_1}\mathbf{v_1}^\mathbf{T} = \mathbf{P_1} - \mathbf{w_1}\mathbf{w_1}^\mathbf{T} = \begin{bmatrix} 3 & 0.5 & 0 & 0.5 \\ 0.5 & 2 & 0.5 & 1 \\ 0 & 0.5 & 3 & 0.5 \\ 0.5 & 1 & 0.5 & 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -0.5 & -1 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ -1 & -0.5 & 2 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix}$$

$$\mathbf{P_2} = \begin{bmatrix} 2 & -0.5 & -1 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ -1 & -0.5 & 2 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix} \quad \text{og potensmetoden finder:} \quad \{\lambda_2, \mathbf{w_2}\} = \{3, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\}$$

=> $|\mathbf{w_2}| = \sqrt{2}$ => $\mathbf{v_2} = \frac{1}{\sqrt{2}} \cdot \mathbf{w_2}$ => $\lambda_2 \mathbf{v_2 v_2^T} = \lambda_2 \frac{1}{\mathbf{w_2^T w_2}} \mathbf{w_2 w_2^T} = 3 \cdot \frac{1}{2} \cdot \mathbf{w_2 w_2^T}$ =>

$\mathbf{P_3} = \mathbf{A} - (\lambda_1 \mathbf{v_1 v_1^T} + \lambda_2 \mathbf{v_2 v_2^T}) = \mathbf{P_2} - \frac{3}{2} \cdot \mathbf{w_2 w_2^T}$

$$= \begin{bmatrix} 2 & -0.5 & -1 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ -1 & -0.5 & 2 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix} - \frac{3}{2} \cdot \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix}$$

$\mathbf{P_3} = \begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix}$ og potensmetoden finder $\{\lambda_3, \mathbf{w_3}\} = \{2, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\}$

=> $|\mathbf{w_3}| = 2$ => $\mathbf{v_3} = \frac{1}{2} \cdot \mathbf{w_3}$ => $\lambda_3 \mathbf{v_3 v_3^T} = \lambda_3 \frac{1}{\mathbf{w_3^T w_3}} \mathbf{w_3 w_3^T} = 2 \cdot \frac{1}{4} \cdot \mathbf{w_3 w_3^T}$ =>

$\mathbf{P_4} = \mathbf{A} - (\lambda_1 \mathbf{v_1 v_1^T} + \lambda_2 \mathbf{v_2 v_2^T} + \lambda_3 \mathbf{v_3 v_3^T}) = \mathbf{P_3} - \frac{1}{2} \cdot \mathbf{w_3 w_3^T}$

$$= \begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix} - \frac{1}{2} \cdot \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & -0.5 \\ 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 \end{bmatrix}$$

$\mathbf{P_4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & -0.5 \\ 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 \end{bmatrix}$ og potensmetoden finder $\{\lambda_4, \mathbf{w_4}\} = \{1, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}$

Samlet oversigt over matricerne $\mathbf{P_1}$, $\mathbf{P_2}$, $\mathbf{P_3}$, $\mathbf{P_4}$ og deres respektive egenløsninger:

$\mathbf{P_1} = \begin{bmatrix} 3 & 0.5 & 0 & 0.5 \\ 0.5 & 2 & 0.5 & 1 \\ 0 & 0.5 & 3 & 0.5 \\ 0.5 & 1 & 0.5 & 2 \end{bmatrix}$ $\{4, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}$, $\{3, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\}$, $\{2, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\}$ og $\{1, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}$

$\mathbf{P_2} = \begin{bmatrix} 2 & -0.5 & -1 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ -1 & -0.5 & 2 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix}$ $\{0, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}$, $\{3, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\}$, $\{2, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\}$ og $\{1, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}$

$\mathbf{P_3} = \begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix}$ $\{0, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}$, $\{0, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\}$, $\{2, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\}$ og $\{1, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}$

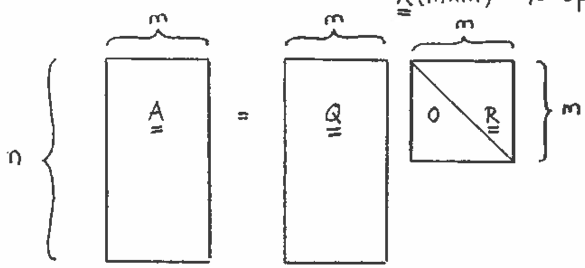
$\mathbf{P_4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & -0.5 \\ 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 \end{bmatrix}$ $\{0, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}\}$, $\{0, \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}\}$, $\{0, \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\}$ og $\{1, \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}\}$

QR factorizations

Recall that the Gram-Schmidt orthogonalization process gave us a QR factorization of A $(n \times m)$ where $n \geq m$:

- $A = QR$ where Q $(n \times m)$ has columns $q_1, \dots, q_m$ which is an ortho-normal basis of the column space of A

  R $(m \times m)$ is upper triangular



Here :
$Q = Q_1$
$R = R_1$
(Q_1 and R_1 : see below)

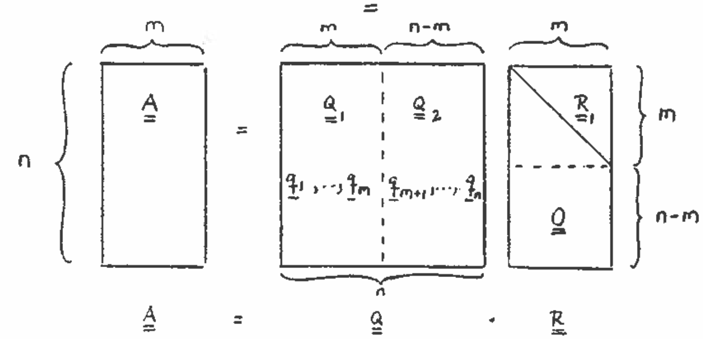
This QR-factorization is a so-called "thin" QR-factorization as opposed to a "full" QR factorization :

- $A = QR$ where Q $(n \times n)$ is orthogonal : $Q^T = Q^{-1}$ and has columns $q_1, \dots, q_n$ which is an ortho-normal basis of the (full) n-dimensional space.

  $q_1, \dots, q_m$ is a basis of the column space of A

  $q_{m+1}, \dots, q_n$ is a basis of the null space of A^T

  R $(n \times m)$ is generalized upper triangular



Here :
$Q = [Q_1 \; Q_2]$
and
$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$
R_1 $(m \times m)$ is upper triangular

In the full factorization :

$$A = Q \cdot R = Q_1 \cdot R_1 + Q_2 \cdot 0 = Q_1 \cdot R_1$$

$Q = [Q_1 \; Q_2]$ $\qquad$ Q_1 $(n \times m)$
$\qquad\qquad\qquad\quad$ Q_2 $(n \times (n-m))$
$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ $\qquad$ R_1 $(m \times m)$ upper triangular
$\qquad\qquad\qquad\quad$ 0 $((n-m) \times m)$ contains all zeros

- Methods computing the full factorization : $A = QR$: $\begin{cases} Q = [Q_1 \; Q_2] \\ R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \end{cases}$
  - Householder reflections
  - Givens rotations

  Methods computing the thin factorization : $A = QR$: $\begin{cases} Q = Q_1 \\ R = R_1 \end{cases}$
  - Gram-Schmidt
  - modified Gram Schmidt

- A few important properties of orthogonal matrices :

1) Q $(n \times n)$ is orthogonal $\iff$ columns $q_1, \dots, q_n$ of Q are mutually orthogonal unit vectors

2) Q is orthogonal $\iff$ Q^T is orthogonal

3) Q is orthogonal $\iff$ $Q^T = Q^{-1}$ $\iff$ $Q^T Q = Q Q^T = I$

4) Q_1 and Q_2 are orthogonal $\Rightarrow$ $Q = Q_1 \cdot Q_2$ is orthogonal

5) $y = Q^T x$ $\Rightarrow$ $\| y \| = \| x \|$ multiplying by Q^T (or Q) preserves the length

Using QR factorization to find the LS-solution to overdetermined systems of linear equations.

- Summary

Overdetermined system of linear equations $\quad A x = b \quad \begin{cases} A\,(n\times m) \\ x\,(m\times 1) \\ b\,(n\times 1) \end{cases} \quad n > m$

Normal equations $\quad A^T A x = A^T b \quad \begin{cases} A^T A\,(m\times m) \\ A^T b\,(m\times 1) \end{cases}$

LS-solution x^* (solution to norm. eqs.) $\quad x^* = (A^T A)^{-1} A^T b$

Projection vector b_p (b projected on column space of A)

$$b_p = A x^* = A(A^T A)^{-1} A^T b$$

Projection matrix P

$$b_p = P b$$
$$P = A(A^T A)^{-1} A^T$$

Error vector e

$$e = b - A x^*$$
$$\|e\|^2 = (b - A x^*)^T (b - A x^*) = \min_x (b - A x)^T (b - A x)$$

- Using "thin" QR-factorization : $A = Q R$ where $Q = Q_1$ $\quad R = R_1$

We find : $\quad A x = b \iff Q_1 R_1 x = b$
$\iff Q_1^T Q_1 R_1 x = Q_1^T b$
$\iff R_1 x = Q_1^T b$
$\iff x^* = R_1^{-1} Q_1^T b$

So : $\quad x^*$ is the solution to $R_1 x = Q_1^T b$ where $R_1\,(m\times m)$ is upper triangular so that the solution (only) requires backwards substitution.

- Using full QR-factorization : $A = Q R$ where $Q(n\times n)$ is orthogonal. $\quad$ where $Q = [Q_1 \; Q_2]$, $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$

With $\quad b_p = Q_1 Q_1^T b \quad$ the projection of b on the column space of A

$b_{po} = Q_2 Q_2^T b \quad$ the projection of b on the null space of A^T

we will show that

$$b = Q_1 Q_1^T b + Q_2 Q_2^T b = b_p + b_{po}$$

Proof :

$$Q^T b = \begin{array}{c} m\{ \\ n-m\{ \end{array}\underbrace{\begin{bmatrix} Q_1^T \\ \hdashline Q_2^T \end{bmatrix}}_{n} \underbrace{\begin{bmatrix} b \end{bmatrix}}_{1} \}n = \underbrace{\begin{bmatrix} Q_1^T b \\ \hdashline Q_2^T b \end{bmatrix}}_{1} \begin{array}{l} \}m \\ \}n-m \end{array}$$

$$Q Q^T b = n\{\begin{bmatrix} \underbrace{Q_1}_{m} & \underbrace{Q_2}_{n-m} \end{bmatrix} \underbrace{\begin{bmatrix} Q_1^T b \\ \hdashline Q_2^T b \end{bmatrix}}_{1} \begin{array}{l} \}m \\ \}n-m \end{array} = Q_1 Q_1^T b + Q_2 Q_2^T b$$

$$\iff \quad b = Q_1 Q_1^T b + Q_2 Q_2^T b = b_p + b_{po} \qquad (Q Q^T = I(n\times n))$$

End of proof

This result is useful in the context of LS solution of an overdetermined system of linear equations :

$A x = b \implies$ 1) $\quad A x = b_p$
2) $\quad e = b - b_p = b - A x = b_{po}$

1) $\quad A x = b_p \iff Q_1 R_1 x = Q_1 Q_1^T b$
$\iff R_1 x = Q_1^T b$
$\iff x^* = R_1^{-1} Q_1^T b \qquad$ (LS solution)

2) $\quad e = b_{po} \iff e = Q_2 Q_2^T b \qquad$ (LS error)
$\iff \|e\|^2 = (Q_2^T b)^T (Q_2^T b)$

So :

Given the full QR factorization : $A = QR$ $\quad Q = [Q_1 \; Q_2]$, $\quad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$

Find the LS solution to $Ax = b$ as follows :

1) Compute : $Q_1^T b$ and $Q_2^T b$

2) Solve : $R_1 x = Q_1^T b$ by backwards substitution

Solution is : $x^* = R_1^{-1} Q_1^T b$ $\quad$ LS solution

3) Compute : $e = Q_2 Q_2^T b$ $\quad$ LS error

$$\|e\|^2 = (Q_2^T b)^T (Q_2^T b)$$

Notice : Computation of e does not require availability of x^*.

Example

Given: $A = Q \cdot R$

$$\begin{bmatrix} 2 & 4 \\ 2 & 2 \\ 2 & 4 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} \frac12 & \frac12 & \frac{1}{\sqrt2} & 0 \\ \frac12 & -\frac12 & 0 & \frac{1}{\sqrt2} \\ \frac12 & \frac12 & -\frac{1}{\sqrt2} & 0 \\ \frac12 & -\frac12 & 0 & -\frac{1}{\sqrt2} \end{bmatrix} \begin{bmatrix} 4 & 6 \\ 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where

$$Q_1 = \begin{bmatrix} \frac12 & \frac12 \\ \frac12 & -\frac12 \\ \frac12 & \frac12 \\ \frac12 & -\frac12 \end{bmatrix} \text{ and } \{q_1, q_2\} = \left\{ \begin{bmatrix} \frac12 \\ \frac12 \\ \frac12 \\ \frac12 \end{bmatrix}, \begin{bmatrix} \frac12 \\ -\frac12 \\ \frac12 \\ -\frac12 \end{bmatrix} \right\}$$ is an orthonormal basis of the column space of A

while

$$Q_2 = \begin{bmatrix} \frac{1}{\sqrt2} & 0 \\ 0 & \frac{1}{\sqrt2} \\ -\frac{1}{\sqrt2} & 0 \\ 0 & -\frac{1}{\sqrt2} \end{bmatrix} \text{ and } \{q_3, q_4\} = \left\{ \begin{bmatrix} \frac{1}{\sqrt2} \\ 0 \\ -\frac{1}{\sqrt2} \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{1}{\sqrt2} \\ 0 \\ -\frac{1}{\sqrt2} \end{bmatrix} \right\}$$ is an orthonormal basis of the null space of A^T

$$R_1 = \begin{bmatrix} 4 & 6 \\ 0 & 2 \end{bmatrix}$$

Let $b = 2q_1 - q_2 + 2\sqrt2\, q_3 - \sqrt2\, q_4 = Q_1 \begin{bmatrix} 2 \\ -1 \end{bmatrix} + Q_2 \begin{bmatrix} 2\sqrt2 \\ -\sqrt2 \end{bmatrix}$

$$= \begin{bmatrix} 1 - \frac12 + 2 - 0 \\ 1 + \frac12 + 0 - 1 \\ 1 - \frac12 - 2 - 0 \\ 1 + \frac12 + 0 + 1 \end{bmatrix} = \begin{bmatrix} 2.5 \\ 0.5 \\ -1.5 \\ 2.5 \end{bmatrix}$$

Now, find the LS-solution to $Ax = b$

- Solving the normal equations $A^T A x = A^T b$:

$$A^T A = \begin{bmatrix} 16 & 24 \\ 24 & 40 \end{bmatrix} \qquad A^T b = \begin{bmatrix} 8 \\ 10 \end{bmatrix}$$

$$A^T A x = A^T b \iff \left[\begin{array}{cc|c} 16 & 24 & 8 \\ 24 & 40 & 10 \end{array}\right] \iff \left[\begin{array}{cc|c} 16 & 24 & 8 \\ 0 & 4 & -2 \end{array}\right]$$

$$\iff x^* = \begin{bmatrix} 1.25 \\ -0.5 \end{bmatrix}$$

$$\Rightarrow b_p = A x^* = \begin{bmatrix} 0.5 \\ 1.5 \\ 0.5 \\ 1.5 \end{bmatrix} \left(= 2q_1 - q_2 \right)$$

$$\Rightarrow e = b - b_p = \begin{bmatrix} 2 \\ -1 \\ -2 \\ 1 \end{bmatrix} \left(= 2\sqrt2\, q_3 - \sqrt2\, q_4 \right)$$

- Using the factorization $A = Q \cdot R$:

$$Q_1^T b = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \qquad Q_2^T b = \begin{bmatrix} \frac{4}{\sqrt2} \\ -\frac{2}{\sqrt2} \end{bmatrix} = \begin{bmatrix} 2\sqrt2 \\ -\sqrt2 \end{bmatrix}$$

(as expected since $b = Q_1 Q_1^T b + Q_2 Q_2^T b$)

$$R_1 x = Q_1^T b \iff \left[\begin{array}{cc|c} 4 & 6 & 2 \\ 0 & 2 & -1 \end{array}\right] \iff x^* = \begin{bmatrix} 1.25 \\ -0.5 \end{bmatrix}$$

$$e = Q_2 Q_2^T b = Q_2 \begin{bmatrix} 2\sqrt2 \\ -\sqrt2 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ -2 \\ 1 \end{bmatrix} \left(= 2\sqrt2\, q_3 - \sqrt2\, q_4 \right)$$

The thin QR-factorization by modified Gram-Schmidt orthogonalization

• Purpose

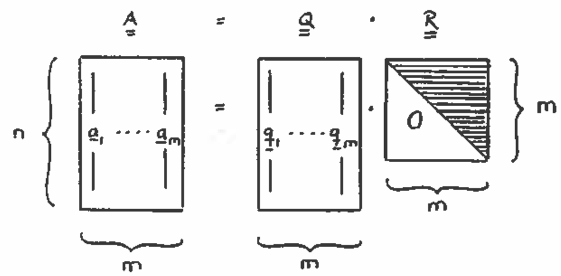
Given: $A\,(n \times m)$ where $n \geq m$ and $\operatorname{rank}(A) = m$ is full.
The columns of A are $a_1, a_2, \dots, a_m$

Compute: $Q\,(n \times m)$ which is ortho-normal.
The columns of Q are $q_1, q_2, \dots, q_m$
which is a set of mutually orthogonal unit vectors forming a basis of $\operatorname{Col}\{A\}$

$R\,(m \times m)$ which is upper triangular.

So that: $A = Q \cdot R$ is a thin QR-factorization of A

$A = Q \cdot R$



• Method

The method is an algorithm which will construct Q and R in m iterations.
With the iteration index $k = 1, 2, 3, \dots, m-1, m$ results are –

On iteration k : Compute q_k column number k of Q

$r_{kk}, \dots, r_{km}$ row number k of R

Proceed as follows :
Since each iteration will transform A we initialize prior to the iterations,

For $k = 0$, let $A_0 = A$ with columns $a_1^{(0)}, a_2^{(0)}, a_3^{(0)}, \dots, a_m^{(0)}$

For $k = 1$, compute $r_{11} = \| a_1^{(0)} \|$

$q_1 = \frac{1}{r_{11}} \cdot a_1^{(0)}$

compute A_1 by transformation of A_0 (columns $2, \dots, m$) ($a_1^{(0)}$ is copied)

column-updates $a_j^{(1)} = a_j^{(0)} - (q_1^T \cdot a_j^{(0)}) \cdot q_1$ $\quad j = 2, \dots, m$

save $r_{1j} = q_1^T \cdot a_j^{(0)}$ $\quad j = 2, \dots, m$

column-update (copy) $a_1^{(1)} = a_1^{(0)}$ (column 1, copy)

Status
q_1 is computed for Q
$r_{11}, r_{12}, \dots, r_{1m}$ are computed for R
$a_1^{(1)} = r_{11} \cdot q_1$ is set in A_1
$a_2^{(1)}, \dots, a_m^{(1)}$ are set in A_1 and : are all orthogonal to q_1

For $k = 2$, compute $r_{22} = \| a_2^{(1)} \|$

$q_2 = \frac{1}{r_{22}} \cdot a_2^{(1)}$

Compute A_2 by transformation of A_1 (columns $3, \dots, m$) ($a_2^{(1)}$ is copied)

column-updates $a_j^{(2)} = a_j^{(1)} - (q_2^T \cdot a_j^{(1)}) \cdot q_2$ $\quad j = 3, \dots, m$

save $r_{2j} = q_2^T \cdot a_j^{(1)}$ $\quad j = 3, \dots, m$

column-update (copy) $a_2^{(2)} = a_2^{(1)}$ (column 2, copy)

Status
q_2 is computed for Q
$r_{22}, r_{23}, \dots, r_{2m}$ are computed for R
$a_2^{(2)} = r_{22} \cdot q_2$ is set in A_2
$a_3^{(2)}, \dots, a_m^{(2)}$ are set in A_2 and : are all orthogonal to q_1 and q_2

For any k, $k = 1, \dots, m-1$ compute $r_{kk} = \| a_k^{(k-1)} \|$

$q_k = \frac{1}{r_{kk}} \cdot a_k^{(k-1)}$

compute A_k by transformation of A_{k-1} (columns $k+1, \dots, m$) ($a_k^{(k-1)}$ is copied)

column-updates $a_j^{(k)} = a_j^{(k-1)} - (q_k^T \cdot a_j^{(k-1)}) \cdot q_k$

save $r_{kj} = q_k^T \cdot a_j^{(k-1)}$ $\Big\}\ j = k+1, \dots, m$

column-update (copy) $a_k^{(k)} = a_k^{(k-1)}$ (column k, copy)

Status
q_k is computed for Q
$r_{kk}, \dots, r_{km}$ are computed for R
$a_k^{(k)} = r_{kk} \cdot q_k$ is set in A_k
$a_{k+1}^{(k)}, \dots, a_m^{(k)}$ are set in A_k and : are all orthogonal to $q_1, \dots, q_k$

For $k = m$, (last iteration)

Compute $r_{mm} = \| a_m^{(m-1)} \|$

$$q_m = \frac{1}{r_{mm}} \cdot a_m^{(m-1)}$$

compute A_m by transformation of A_{m-1} implies only the copying :

column-update (copy) $a_m^{(m)} = a_m^{(m-1)}$

Status
- q_m is computed for Q (done!)
- r_{mm} is computed for R (done!)
- $a_m^{(m)} = r_{mm} \cdot q_m$ is set in A_m (done!)

- Structural notes

The columns of A : $a_1, a_2, a_3, \ldots, a_m$

The columns of A_m : $a_1^{(1)}, a_2^{(2)}, a_3^{(3)}, \ldots, a_m^{(m)}$

which is : $r_{11} \cdot q_1, r_{22} \cdot q_2, r_{33} \cdot q_3, \ldots, r_{mm} \cdot q_m$

The columns of Q : $q_1, q_2, q_3, \ldots, q_m$

where

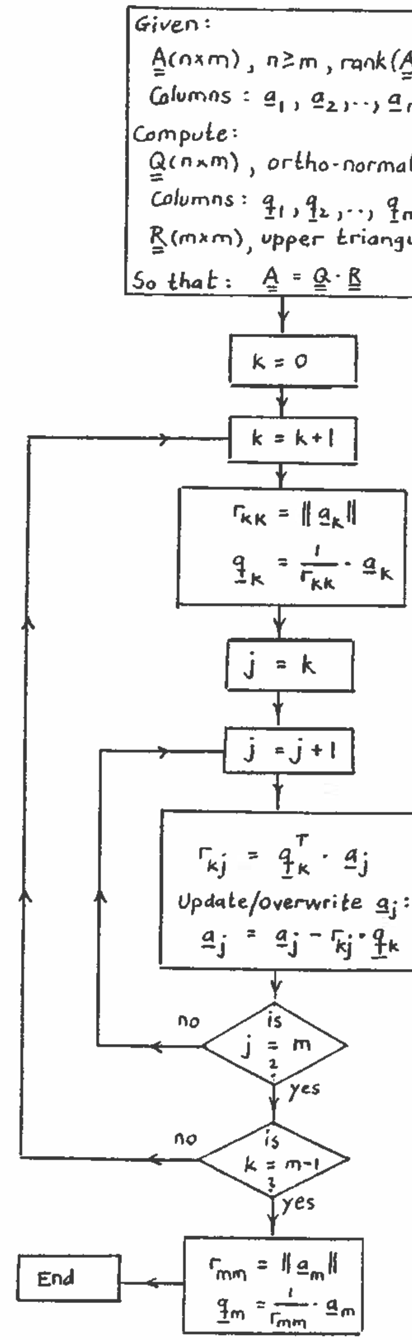
$$a_1^{(1)} = a_1$$
$$a_2^{(2)} = a_2 - r_{12} \cdot q_1$$
$$a_3^{(3)} = a_3 - r_{13} \cdot q_1 - r_{23} \cdot q_2$$
$$a_4^{(4)} = a_4 - r_{14} \cdot q_1 - r_{24} \cdot q_2 - r_{34} \cdot q_3$$
$$\vdots$$
$$a_m^{(m)} = a_m - r_{1m} \cdot q_1 - r_{2m} \cdot q_2 - r_{3m} \cdot q_3 - r_{4m} \cdot q_4 - \ldots - r_{mm} \cdot q_m$$

<=>

$$a_1 = r_{11} \cdot q_1$$
$$a_2 = r_{12} \cdot q_1 + r_{22} \cdot q_2$$
$$a_3 = r_{13} \cdot q_1 + r_{23} \cdot q_2 + r_{33} \cdot q_3$$
$$a_4 = r_{14} \cdot q_1 + r_{24} \cdot q_2 + r_{34} \cdot q_3 + r_{44} \cdot q_4$$
$$\vdots$$
$$a_m = r_{1m} \cdot q_1 + r_{2m} \cdot q_2 + r_{3m} \cdot q_3 + r_{4m} \cdot q_4 + \ldots + r_{mm} \cdot q_m$$

<=> $A = Q \cdot R$

Implementation plan for QR-factorization by modified Gram-Schmidt



Notes :
- The algorithm is iterative and performs iterations $k = 1, 2, 3, \ldots, m$
- On iteration $k = 1, .., m-1$
  - In R : Sets row k $r_{kk}, \ldots, r_{km}$
  - In Q : Sets column k q_k
  - In A : Sets column k $r_{kk} \cdot q_k$. Overwrites/updates columns $k+1, .., m$. No change in columns $1, .., k-1$
- On iteration $k = m$
  - In R : Sets r_{mm}
  - In Q : Sets q_m
  - In A : No changes

← • Here:
From A, select col. k and
In R: set r_{kk} in row k
In Q: set q_k in col. k =>
In A: col. k is now $r_{kk} \cdot q_k$

← • Here:
In R : Remaining elements r_{kj} with $j = k+1, .., m$ are set
In A: columns a_j with $j = k+1, .., m$ are updated
The projection of a_j on q_k is :

$$a_{jp} = \frac{q_k^T \cdot a_j}{q_k^T \cdot q_k} \cdot q_k \Rightarrow$$

$$a_{jp} = r_{kj} \cdot q_k \Rightarrow$$

The updated a_j is orthogonal to q_k and thus to $q_1, \ldots, q_{k-1}$ as well

← • Here:
On exit from iteration $k = m-1$ the current a_m is orthogonal to $q_1, .., q_{m-1}$ allowing :
In R set r_{mm}, in Q set q_m

Computing the QR-factorization by Gram-Schmidt orthogonalization

The objective is

Given: $\underline{\underline{A}}(n \times m)$ with $n \geq m$, $\operatorname{rank}(\underline{\underline{A}}) = m$, and columns $\underline{a}_1, \dots, \underline{a}_m$

Compute: $\underline{\underline{Q}}(n \times m)$ which is ortho-normal: $\underline{\underline{Q}}^T \underline{\underline{Q}} = \underline{\underline{I}}(m \times m)$. Columns: $\underline{q}_1, \dots, \underline{q}_m$

$\underline{\underline{R}}(m \times m)$ which is upper triangular

So that: $\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}}$

$k = 1$: $\quad \underline{v}_1 = \underline{a}_1$

$$\underline{q}_1 = \frac{1}{r_{11}} \cdot \underline{v}_1 \quad \text{with } r_{11} = \|\underline{v}_1\|$$

$k = 2$: $\quad \underline{v}_2 = \underline{a}_2 - \dfrac{\underline{q}_1^T \cdot \underline{a}_2}{\underline{q}_1^T \cdot \underline{q}_1} \cdot \underline{q}_1$

$$\underline{q}_2 = \frac{1}{r_{22}} \cdot \underline{v}_2 \quad \text{with } r_{22} = \|\underline{v}_2\|$$

$k = 3$: $\quad \underline{v}_3 = \underline{a}_3 - \dfrac{\underline{q}_1^T \cdot \underline{a}_3}{\underline{q}_1^T \cdot \underline{q}_1} \cdot \underline{q}_1 - \dfrac{\underline{q}_2^T \cdot \underline{a}_3}{\underline{q}_2^T \cdot \underline{q}_2} \cdot \underline{q}_2$

$$\underline{q}_3 = \frac{1}{r_{33}} \cdot \underline{v}_3 \quad \text{with } r_{33} = \|\underline{v}_3\|$$

$\vdots$

$k = m$: $\quad \underline{v}_m = \underline{a}_m - \dfrac{\underline{q}_1^T \cdot \underline{a}_m}{\underline{q}_1^T \cdot \underline{q}_1} \cdot \underline{q}_1 - \dfrac{\underline{q}_2^T \cdot \underline{a}_m}{\underline{q}_2^T \cdot \underline{q}_2} \cdot \underline{q}_2 - \dfrac{\underline{q}_3^T \cdot \underline{a}_m}{\underline{q}_3^T \cdot \underline{q}_3} \cdot \underline{q}_3 - \dots - \dfrac{\underline{q}_{m-1}^T \cdot \underline{a}_m}{\underline{q}_{m-1}^T \cdot \underline{q}_{m-1}} \cdot \underline{q}_{m-1}$

$$\underline{q}_m = \frac{1}{r_{mm}} \cdot \underline{v}_m \quad \text{with } r_{mm} = \|\underline{v}_m\|$$

Now, with $r_{jk} = \underline{q}_j^T \cdot \underline{a}_k$ and using $\underline{q}_k^T \underline{q}_k = 1$:

$$r_{11} \cdot \underline{q}_1 = \underline{a}_1$$
$$r_{22} \cdot \underline{q}_2 = \underline{a}_2 - r_{12} \cdot \underline{q}_1$$
$$r_{33} \cdot \underline{q}_3 = \underline{a}_3 - r_{13} \cdot \underline{q}_1 - r_{23} \cdot \underline{q}_2$$
$$\vdots$$
$$r_{mm} \cdot \underline{q}_m = \underline{a}_m - r_{1m} \cdot \underline{q}_1 - r_{2m} \cdot \underline{q}_2 - r_{3m} \cdot \underline{q}_3 - \dots - r_{m-1,m} \cdot \underline{q}_{m-1}$$

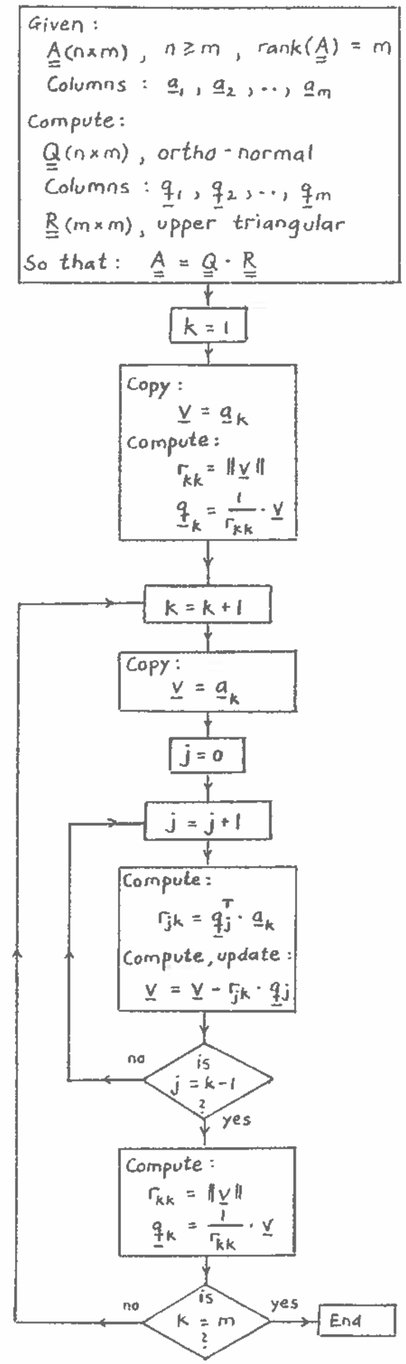
$\Leftrightarrow$

$$\underline{a}_1 = r_{11} \cdot \underline{q}_1$$
$$\underline{a}_2 = r_{12} \cdot \underline{q}_1 + r_{22} \cdot \underline{q}_2$$
$$\underline{a}_3 = r_{13} \cdot \underline{q}_1 + r_{23} \cdot \underline{q}_2 + r_{33} \cdot \underline{q}_3$$
$$\vdots$$
$$\underline{a}_m = r_{1m} \cdot \underline{q}_1 + r_{2m} \cdot \underline{q}_2 + r_{3m} \cdot \underline{q}_3 + \dots + r_{m-1,m} \cdot \underline{q}_{m-1} + r_{mm} \cdot \underline{q}_m$$

$\Leftrightarrow$

$$\underline{\underline{A}} = \underline{\underline{Q}} \cdot \underline{\underline{R}}$$

Implementation plan for QR-factorization by Gram-Schmidt orthogonalization



Notes

- The algorithm is iterative and performs iterations $k = 1, 2, 3, \dots, m$
- On iteration $k = 1, 2, \dots, m$

  In $\underline{\underline{R}}$: Sets column k: $r_{1k}, r_{2k}, \dots, r_{kk}$

  In $\underline{\underline{Q}}$: Sets column k: $\underline{q}_k$

  In $\underline{\underline{A}}$: No changes

- On iteration $k = 1$

  Compute:

  $r_{11} = \|\underline{a}_1\|$

  $\underline{q}_1 = \frac{1}{r_{11}} \cdot \underline{a}_1$

  Using the temporary $\underline{v}$ is really not necessary with $k = 1$ as no projection is performed on iteration $k = 1$

- On iteration $k = 2, \dots, m$

  Compute:

  $$\underline{v} = \underline{a}_k - \sum_{j=1}^{k-1} \frac{\underline{q}_j^T \cdot \underline{a}_k}{\underline{q}_j^T \cdot \underline{q}_j} \cdot \underline{q}_j$$

  $$\underline{q}_k = \frac{1}{\|\underline{v}\|} \cdot \underline{v} \quad (\Rightarrow \underline{q}_k^T \underline{q}_k = 1)$$

  $r_{jk} = \underline{q}_j^T \cdot \underline{a}_k \quad j = 1, 2, \dots, k-1$

  $r_{kk} = \|\underline{v}\| \quad j = k$

- Implementation, since:

  $\underline{v} = \underline{a}_k - \sum_{j=1}^{k-1} r_{jk} \cdot \underline{q}_j$ and

  $\underline{q}_k = \frac{1}{r_{kk}} \cdot \underline{v}$ then:

- On iteration k, with $j = 0$

  Copy: $\underline{v} = \underline{a}_k$

  $j = 1, 2, \dots, k-1$

  Compute: $r_{jk} = \underline{q}_j^T \underline{a}_k$

  Compute, update: $\underline{v} = \underline{v} - r_{jk} \cdot \underline{q}_j$

  $j = k$

  Compute:

  $r_{kk} = \|\underline{v}\|$

  $\underline{q}_k = \frac{1}{r_{kk}} \cdot \underline{v}$

Using the QR-factorization $\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}}$ for design of LS-test examples

In general $\underline{\underline{A}}(n \times m)$ with $n > m$ is given and $\underline{\underline{Q}}$ and $\underline{\underline{R}}$ and found:

$\underline{\underline{Q}}(n \times m)$ has the ortho-normal-columns $\underline{q}_1, \dots, \underline{q}_m$

$\{\underline{q}_1, \dots, \underline{q}_m\}$ is an ortho-normal basis of $Col\{\underline{\underline{A}}\}$

$\underline{\underline{R}}(m \times m)$ is upper triangular

Here, we will

1) Create $\underline{\underline{Q}}$ as follows:
   Enter $n = 4, 8$, or 16. Enter m with $m < n$
   Columns of $\underline{\underline{Q}}$ are selected: $\underline{q}_k = \underline{s}_k \quad k = 1, \dots, m$ as m columns of $\underline{\underline{S}}_n$
2) Enter / choose the elements of $\underline{\underline{R}}$
3) Compute $\underline{\underline{A}} = \underline{\underline{Q}}\,\underline{\underline{R}}$
4) Create $\underline{b}$ as follows:
   Choose/enter the desired $\underline{x}^*(m \times 1)$ and compute $\underline{b}_p = \underline{\underline{A}}\,\underline{x}^*$
   Choose/enter $\underline{t}^*((n-m) \times 1)$ where $\underline{t}^{*T} = [t_1 \ t_2 \cdots t_{n-m}]$
   Compute $\underline{e} = t_1 \cdot \underline{c}_{m+1} + t_2 \cdot \underline{c}_{m+2} + \dots + t_{n-m} \cdot \underline{c}_n \quad \in Null\{\underline{\underline{A}}^T\}$
   Here: $\underline{c}_1, \underline{c}_2, \dots, \underline{c}_n$ are the n columns of $\underline{\underline{A}}_n$

   Compute: $\underline{b} = \underline{b}_p + \underline{e}$

Now: $\underline{\underline{A}}\,\underline{x} = \underline{b}$ has the L.S. solution $\underline{x}^*$ with $\underline{e} = \underline{b} - \underline{\underline{A}}\,\underline{x}^*$

Example

1) Enter: $n = 4$ and $m = 2 \quad \Rightarrow \quad \underline{\underline{Q}} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \\ 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$

2) Enter: $\underline{\underline{R}} = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$

3) Compute: $\underline{\underline{A}} = \underline{\underline{Q}} \cdot \underline{\underline{R}} \quad \Rightarrow \quad \underline{\underline{A}} = \begin{bmatrix} 1 & 2 \\ 1 & -1 \\ 1 & 2 \\ 1 & -1 \end{bmatrix}$

4) Enter: $\underline{x}^* = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \underline{t}^* = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \Rightarrow \quad \underline{b}_p = \underline{\underline{A}}\,\underline{x}^* = \begin{bmatrix} 4 \\ 1 \\ 4 \\ 1 \end{bmatrix}$

$$\underline{e} = 3 \cdot \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} + 2 \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ -3 \\ -2 \end{bmatrix} \quad \Rightarrow \quad \underline{b} = \underline{b}_p + \underline{e} = \begin{bmatrix} 7 \\ 3 \\ 1 \\ -1 \end{bmatrix}$$

Find the Least Squares solution $\underline{x}^*$ to $\underline{\underline{A}}\,\underline{x} = \underline{b}$

$$\underline{\underline{A}}\,\underline{x} = \underline{b} \quad \Rightarrow \quad \underline{\underline{Q}}\,\underline{\underline{R}}\,\underline{x} = \underline{b} \quad \Rightarrow \quad \underline{\underline{R}}\,\underline{x} = \underline{\underline{Q}}^T\underline{b} \quad \Rightarrow \quad \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

As expected: $\Rightarrow \quad \underline{x}^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \quad \Rightarrow \quad \underline{e} = \underline{b} - \underline{\underline{A}}\,\underline{x}^* = \begin{bmatrix} 7 \\ 3 \\ 1 \\ -1 \end{bmatrix} - \begin{bmatrix} 4 \\ 1 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ -3 \\ -2 \end{bmatrix}$

- The following 3 matrices are orthogonal:

$$\underline{\underline{A}}_4 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & -1 \end{bmatrix}$$

columns: $\underline{a}_k$ where $k = 1, \dots, 4$
$|\underline{a}_k| = \sqrt{4} \quad k = 1, 2$
$|\underline{a}_k| = \sqrt{2} \quad k = 3, 4$

$$\underline{\underline{A}}_8 = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

columns: $\underline{a}_k$ where $k = 1, \dots, 8$
$|\underline{a}_k| = \sqrt{8} \quad k = 1, 2$
$|\underline{a}_k| = \sqrt{4} \quad k = 3, 4$
$|\underline{a}_k| = \sqrt{2} \quad k = 5, \dots, 8$

$$\underline{\underline{A}}_{16} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

columns: $\underline{a}_k$ $\quad k = 1, \dots, 16$
$|\underline{a}_k| = \sqrt{16} \quad k = 1, 2$
$|\underline{a}_k| = \sqrt{8} \quad k = 3, 4$
$|\underline{a}_k| = \sqrt{4} \quad k = 5, \dots, 8$
$|\underline{a}_k| = \sqrt{2} \quad k = 9, \dots, 16$

- The following 3 matrices are diagonal:

$\underline{\underline{D}}_4 \ (4 \times 4)$ with elements of the diagonal: $d_k = \dfrac{1}{|\underline{a}_k|} \quad k = 1, \dots, 4$
where $|\underline{a}_k| \quad k = 1, \dots, 4$ are the lengths of columns of $\underline{\underline{A}}_4$

$\underline{\underline{D}}_8 \ (8 \times 8)$ with elements of the diagonal: $d_k = \dfrac{1}{|\underline{a}_k|} \quad k = 1, \dots, 8$
where $|\underline{a}_k| \quad k = 1, \dots, 8$ are the lengths of columns of $\underline{\underline{A}}_8$

$\underline{\underline{D}}_{16} \ (16 \times 16)$ with elements of the diagonal: $d_k = \dfrac{1}{|\underline{a}_k|} \quad k = 1, \dots, 16$
where $|\underline{a}_k| \quad k = 1, \dots, 16$ are the lengths of columns of $\underline{\underline{A}}_{16}$

- The following 3 matrices are ortho-normal:

$\underline{\underline{S}}_4 = \underline{\underline{A}}_4 \cdot \underline{\underline{D}}_4 \qquad \underline{\underline{S}}_8 = \underline{\underline{A}}_8 \cdot \underline{\underline{D}}_8 \qquad \underline{\underline{S}}_{16} = \underline{\underline{A}}_{16} \cdot \underline{\underline{D}}_{16}$

Minimization of a function $f(\underline{x}) = f(x_1, x_2, \ldots, x_n)$

- Here $f(\underline{x})$ is a function of the multivariable input $\underline{x}$ where $\underline{x}^T = [x_1\ x_2 \cdots x_n]$ and $f(\underline{x})$ is twice differentiable.

  The purpose of the class of iterative algorithms presented below:
  - From a chosen initial: $\underline{x}_0$
  - Generate a sequence: $\underline{x}_0, \underline{x}_1, \underline{x}_2, \ldots, \underline{x}_k, \underline{x}_{k+1}, \ldots, \underline{x}_M$
    where the final $\underline{x}_M$: $\underline{x}_M \cong \underline{x}^*$
  - Here: $\underline{x}^*$ is a local minimum of $f(\underline{x})$

- The basic (shared) logic of the class of algorithms is:

  Initialization, choose
  $\underline{x}_0$ the initial $\underline{x}_k$
  ε a tolerance
  N the maximum number of allowed iterations

  On iteration $k = 0, 1, 2, \ldots$
  given $\underline{x}_k$
  compute $\underline{s}_k$ the search direction vector where $|\underline{s}_k| = 1$
  compute α_k where
  $$f(\underline{x}_k + \alpha_k \cdot \underline{s}_k) = \min_{\alpha > 0} f(\underline{x}_k + \alpha \cdot \underline{s}_k)$$
  compute $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \cdot \underline{s}_k$

  Termination criterion
  terminate if: $|\underline{x}_{k+1} - \underline{x}_k| < \varepsilon \ \vee \ |\underline{\nabla} f(\underline{x}_{k+1})| < \varepsilon \ \vee \ k+1 = N$

- The individual algorithms of the class are characterized by
  1) The strategy for computation of $\underline{s}_k$
  2) The strategy for computation of α_k

- Strategies for computation of $\underline{s}_k$
  - The steepest descent direction : $\underline{v} = -\underline{\nabla} f(\underline{x}_k) \ \Rightarrow \ \underline{s}_k = \frac{1}{|\underline{v}|} \cdot \underline{v}$
  - The Newton direction : $\underline{v} = -\left[\underline{\nabla}^2 f(\underline{x}_k)\right]^{-1} \cdot \underline{\nabla} f(\underline{x}_k) \ \Rightarrow \ \underline{s}_k = \frac{1}{|\underline{v}|} \cdot \underline{v}$
  - The BFGS direction : $\underline{v} = -\underline{\underline{H}}_k \cdot \underline{\nabla} f(x_k) \ \Rightarrow \ \underline{s}_k = \frac{1}{|\underline{v}|} \cdot \underline{v}$

    where, Initially: $\underline{\underline{H}}_0 = \underline{\underline{I}}(n \times n)$

$$\left.\begin{aligned} \underline{t}_k &= \underline{x}_{k+1} - \underline{x}_k \\ \underline{g}_k &= \underline{\nabla} f(\underline{x}_{k+1}) - \underline{\nabla} f(\underline{x}_k) \\ \underline{q}_k &= \underline{t}_k - \underline{\underline{H}}_k \cdot \underline{g}_k \end{aligned}\right\} \quad \left\{\begin{aligned} &\underline{\underline{H}}_k \text{ update:} \\ &\underset{0,1,2,\ldots}{k=} \quad \underline{\underline{H}}_{k+1} = \underline{\underline{H}}_k + \frac{\underline{q}_k \cdot \underline{t}_k^T + \underline{t}_k \cdot \underline{q}_k^T}{\underline{g}_k^T \cdot \underline{t}_k} - \frac{\underline{q}_k^T \cdot \underline{g}_k}{(\underline{g}_k^T \cdot \underline{t}_k)^2} \cdot \underline{t}_k \cdot \underline{t}_k^T \end{aligned}\right.$$

- Strategi til beregning af α_k

Golden Section **(det gyldne snits metode) er en iterativt intervalreducerende metode til bestemmelse af en tilnærmelse til x^* hvor $f(x^*)$ er et lokalt minimum for funktionen $f(x)$.**
Metoden forudsætter, at man kender intervallet [a,b] hvorom det gælder:
1. **$f(x)$ er unimodal over [a,b]**
2. **x^* er et indre punkt i [a,b] – således at: $a < x^* < b$**

Det gyldne snits konstant er: $c = \frac{\sqrt{5}-1}{2} \approx 0.618 \ \Rightarrow \ 1 - c = \frac{3-\sqrt{5}}{2} \approx 0.382$

Når de 4 værdier x_1, x_2, x_3 og x_4 optræder i det følgende gælder det overalt at
1. $x_1 < x_2 < x_3 < x_4$ og
2. $x_2 = x_1 + (1-c)(x_4 - x_1)$ *samt* $x_3 = x_4 - (1-c)(x_4 - x_1)$

Golden Section algoritmen

Initialisering	**Sæt** $x_1 = a$ $x_4 = b$ $x_2 = x_1 + (1-c)(x_4 - x_1)$ $x_3 = x_4 - (1-c)(x_4 - x_1)$ **Vælg** ε **som er tolerancen i stopkriteriet**
Iteration k = 1, 2, 3,...	**Givet** x_1, x_2, x_3 **og** x_4 **Test** $f(x_2) \geq f(x_3)$ **Hvis ja, sæt** $x_1 = x_2$ $x_2 = x_3$ $x_3 = x_4 - (1-c)(x_4 - x_1)$ **Hvis nej, sæt** $x_4 = x_3$ $x_3 = x_2$ $x_2 = x_1 + (1-c)(x_4 - x_1)$
Stopkriterium	**Stop hvis** $\lvert x_4 - x_1 \rvert < \varepsilon$
Ved terminering	**Sæt** $x^* = ½(x_4 + x_1)$

Indkredsningsalgoritme
Formål: Frembringelse af et startinterval [a,b] for Golden Section algoritmen.
Givet: x_1 **hvor** $f'(x_1) < 0$ **og**
d **hvor** $d > 0$ **er en brugervalgt start-steplængde**

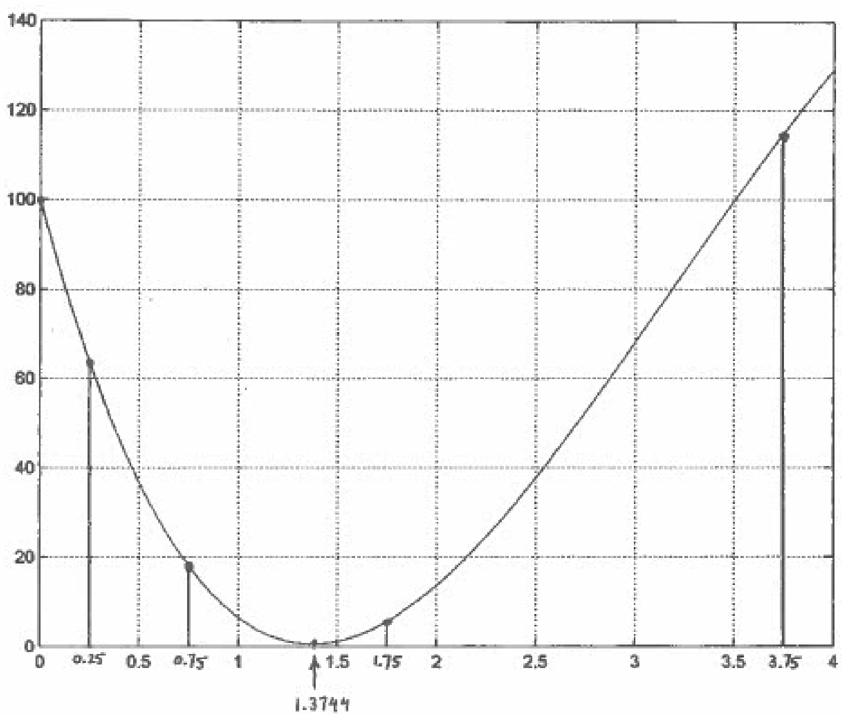
Algoritme

$x_2 = x_1 + d$
while $f(x_2) \geq f(x_1)$
  $d = 0.1 \cdot d$
  $x_2 = x_1 + d$
end
$d = 2 \cdot d$
$x_3 = x_2 + d$
while $f(x_2) > f(x_3)$
  $x_1 = x_2$
  $x_2 = x_3$
  $d = 2 \cdot d$
  $x_3 = x_2 + d$
end
$a = x_1$
$b = x_3$

```
>> xk = [4 3];
>> sk = [-0.8 -0.6];
>> tmax = 4;
>> d = 0.25;
>> PlotCrossSectionP4(xk, sk, tmax)
>> tInt = GSIntervalP4(xk, sk, d)

tInt =

    0.7500    3.7500

>> eps = 0.0000000001;
>> tk = GoldenSectionP4(tInt, xk, sk, eps)

tk =

    1.3744
```

Indkredsning af startinterval til Golden Section

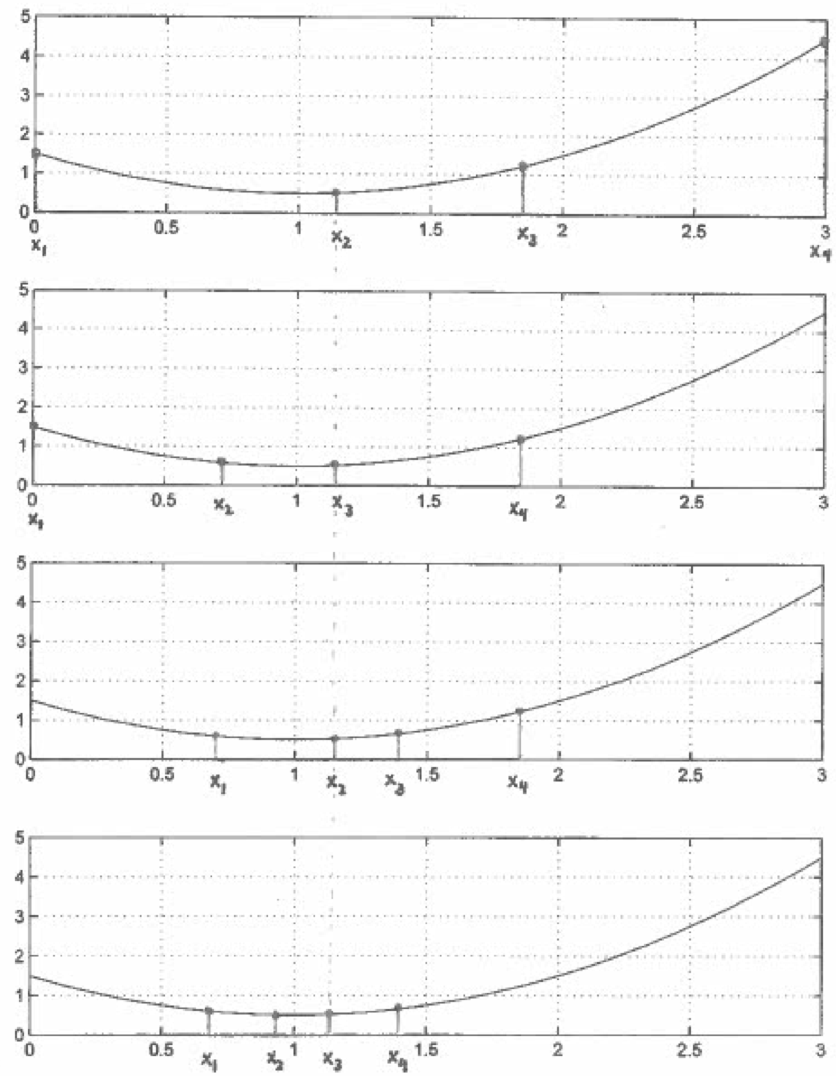
23



Golden Section

$$P_2(x) = (x-1)^2 + \frac{1}{2} = x^2 - 2x + \frac{3}{2}$$

24



$$\tau = \frac{\sqrt{5}-1}{2} = \frac{x_3 - x_1}{x_4 - x_1} = \frac{x_4 - x_2}{x_4 - x_1} \cong 0.6180$$

Functions of n variables $x_1, x_2, \dots, x_n$

Example, with $n = 2$

- $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is the vector of variables : the location vector

- The functional expression

$$f(\underline{x}) = f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

- The partial derivatives of order 1

$$\frac{\partial f}{\partial x_1} = 2\cdot(x_1^2 + x_2 - 11)\cdot 2x_1 + 2\cdot(x_1 + x_2^2 - 7)\cdot 1$$
$$= 4x_1^3 + 4x_1x_2 - 44x_1 + 2x_1 + 2x_2^2 - 14$$
$$= 4x_1^3 + 4x_1x_2 + 2x_2^2 - 42x_1 - 14$$

$$\frac{\partial f}{\partial x_2} = 2\cdot(x_1^2 + x_2 - 11)\cdot 1 + 2\cdot(x_1 + x_2^2 - 7)\cdot 2x_2$$
$$= 2x_1^2 + 2x_2 - 22 + 4x_1x_2 + 4x_2^3 - 28x_2$$
$$= 4x_2^3 + 4x_1x_2 + 2x_1^2 - 26x_2 - 22$$

- The gradient vector

$$\underline{\nabla} f(\underline{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 4x_1^3 + 4x_1x_2 + 2x_2^2 - 42x_1 - 14 \\ 4x_2^3 + 4x_1x_2 + 2x_1^2 - 26x_2 - 22 \end{bmatrix}$$

- The partial derivatives of order 2

$$\frac{\partial^2 f}{\partial x_1^2} = 12x_1^2 + 4x_2 - 42 \qquad \frac{\partial^2 f}{\partial x_1 \partial x_2} = 4x_1 + 4x_2$$

$$\frac{\partial^2 f}{\partial x_2 \partial x_1} = 4x_2 + 4x_1 \qquad \frac{\partial^2 f}{\partial x_2^2} = 12x_2^2 + 4x_1 - 26$$

- The Hessian matrix

$$\underline{\underline{\nabla}}^2 f(\underline{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 12x_1^2 + 4x_2 - 42 & 4x_1 + 4x_2 \\ 4x_1 + 4x_2 & 12x_2^2 + 4x_1 - 26 \end{bmatrix}$$

- Analysis of $f(\underline{x})$ in the neighbourhood of the location $P_0(x_1, x_2) = (4, 3)$

The location vector: $\underline{x}_0 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$

The value of $f(\underline{x})$: $f(\underline{x}_0) = f(4,3) = 100$

The gradient : $\underline{\nabla} f(\underline{x}_0) = \begin{bmatrix} 140 \\ 88 \end{bmatrix}$

The Hessian : $\underline{\underline{\nabla}}^2 f(\underline{x}_0) = \begin{bmatrix} 162 & 28 \\ 28 & 98 \end{bmatrix}$

The tangent plane : $z = 140\cdot(x_1 - 4) + 88\cdot(x_2 - 3) + 100$

The contour : $(x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 = 100$

Contour plot : $3 \le x_1 \le 5 \ \wedge \ 2 \le x_2 \le 4$ (see figures)

Cross-section : plot of $(t\,,\ f(\underline{x}_0 + t\cdot\underline{u}))$ (see figures)
with $\underline{u} = \begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix}$ and $-1 \le t \le 1$

The directional derivative : With $\underline{u} = \begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix}$

$$D_{\underline{u}} f(\underline{x}) = \underline{\nabla} f(\underline{x}_0)^T \cdot \underline{u}$$
$$= [140 \ \ 88] \begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix} = 154.4$$

- Taylor approximations of $f(\underline{x})$ with expansion point $\underline{x} = \underline{x}_0$

· Order $n = 0$ (horizontal plane) :

$T_0(\underline{x}) = f(\underline{x}_0) \quad \Rightarrow \quad z = 100$

· Order $n = 1$ (tangent plane) :

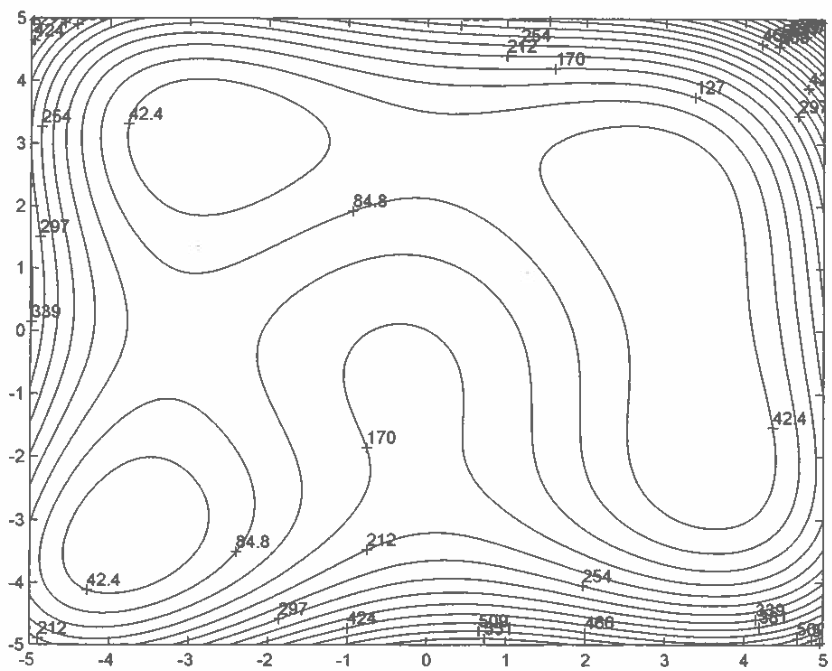
$$T_1(\underline{x}) = f(\underline{x}_0) + \underline{\nabla} f(\underline{x}_0)^T \cdot (\underline{x} - \underline{x}_0) \quad \Rightarrow$$
$$z = 100 + [140 \ \ 88] \cdot \begin{bmatrix} x_1 - 4 \\ x_2 - 3 \end{bmatrix}$$

· Order $n = 2$ (quadratic form) :

$$T_2(\underline{x}) = f(\underline{x}_0) + \underline{\nabla} f(\underline{x}_0)^T \cdot (\underline{x} - \underline{x}_0) + \tfrac{1}{2}(\underline{x} - \underline{x}_0)^T \underline{\underline{\nabla}}^2 f(\underline{x}_0) \cdot (\underline{x} - \underline{x}_0) \Rightarrow$$
$$z = 100 + [140 \ \ 88] \cdot \begin{bmatrix} x_1 - 4 \\ x_2 - 3 \end{bmatrix} + \tfrac{1}{2} \cdot [(x_1 - 4) \ \ (x_2 - 3)] \cdot \begin{bmatrix} 162 & 28 \\ 28 & 98 \end{bmatrix} \cdot \begin{bmatrix} x_1 - 4 \\ x_2 - 3 \end{bmatrix}$$

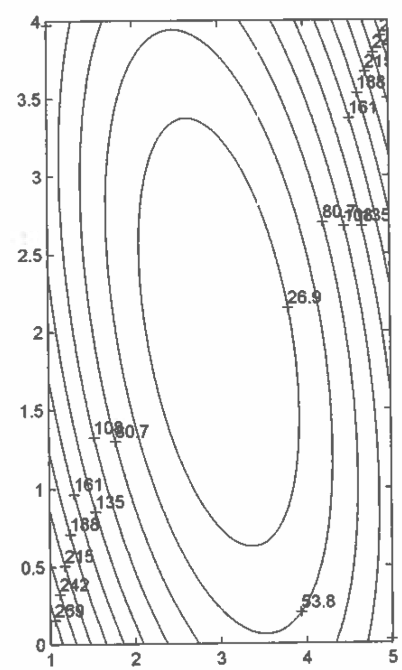
27

$$f(\underline{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

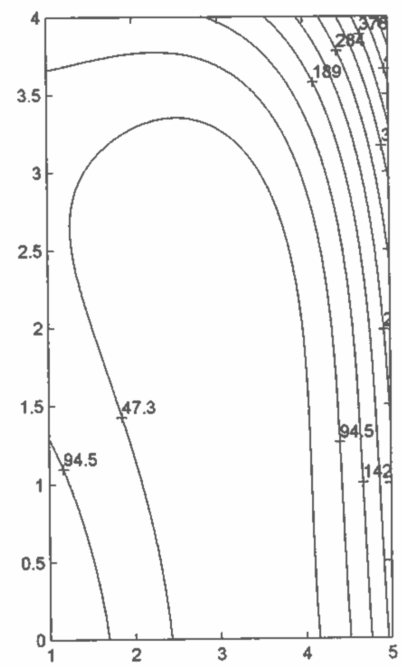


28

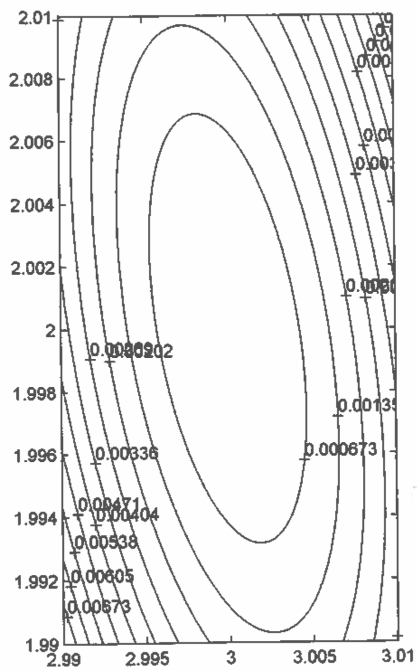
$T_2(\underline{x}) \quad \underline{x}_0 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$



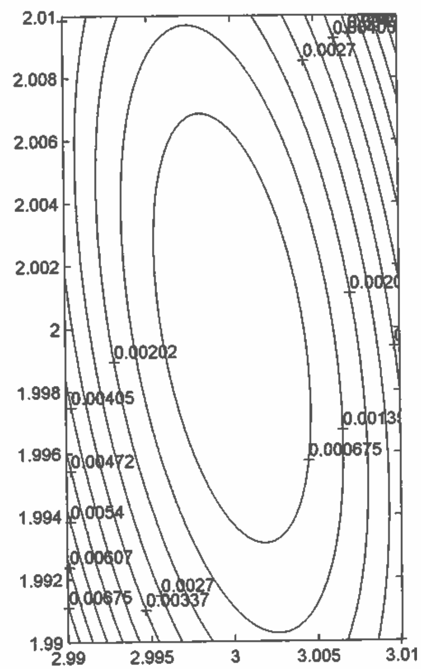
$f(\underline{x})$



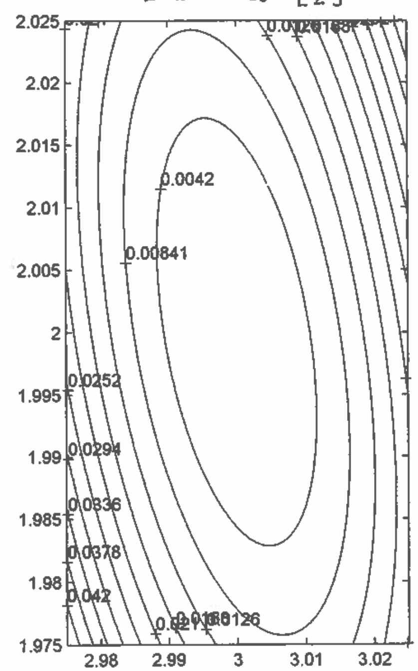
$T_2(\underline{x}) \quad \underline{x}_o = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$



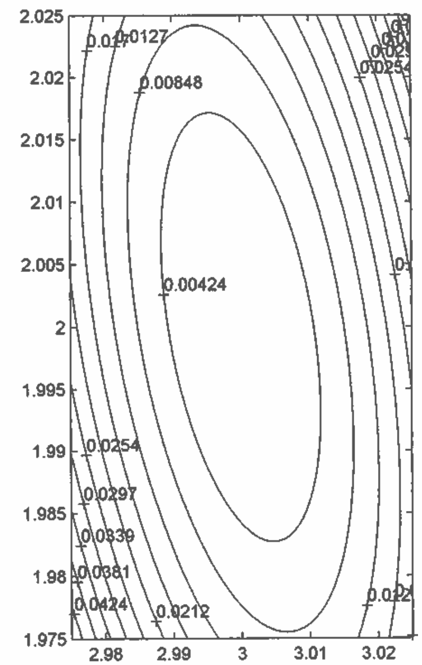
$f(\underline{x})$



$T_2(\underline{x}) \quad \underline{x}_o = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$



$f(\underline{x})$

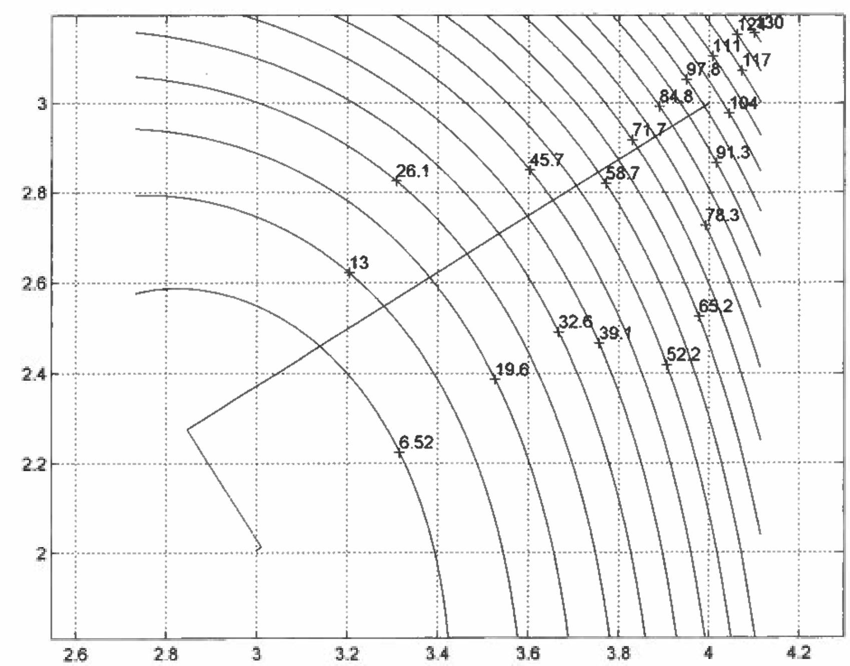


```
>> [xmin,k,xkMat] = MinimizeP4()
Enter the starting-vector x0 as follows: Enter [2 -5]
to initialize the iteration with x0 = [2 -5]
Enter your starting-vector for the iteration here: x0 = [4 3]
Enter your termination criterion tolerance here: eps = 0.0000000001
Enter your initial displacement for a G-S interval here: d = 0.25
Enter your displacement for partial derivatives here: h = 0.00001
Enter your the max. number of allowed iterations here: N = 1000

xmin =

   2.99999999999963
   2.00000000000077


k =

    17


xkMat =

                  0   4.00000000000000   3.00000000000000
   1.00000000000000   2.84633135547752   2.27483685201444
   2.00000000000000   3.01076582373729   2.01323635785849
   3.00000000000000   2.99761761575176   2.00497177454968
   4.00000000000000   3.00041783975958   2.00051687855828
   5.00000000000000   2.99990656137432   2.00019550305169
   6.00000000000000   3.00001656555678   2.00002049729668
   7.00000000000000   2.99999629392125   2.00000775502318
   8.00000000000000   3.00000065732195   2.00000081341200
   9.00000000000000   2.99999985291986   2.00000030776500
  10.00000000000000   3.00000002608929   2.00000003228320
  11.00000000000000   2.99999999416249   2.00000001221438
  12.00000000000000   3.00000000103515   2.00000000128266
  13.00000000000000   2.99999999976787   2.00000000048559
  14.00000000000000   3.00000000004117   2.00000000005116
  15.00000000000000   2.99999999999073   2.00000000001940
  16.00000000000000   3.00000000000164   2.00000000000203
  17.00000000000000   2.99999999999963   2.00000000000077

>> PlotIterationsP4(xkMat,1,4)
>> PlotIterationsP4(xkMat,4,7)
>> PlotIterationsP4(xkMat,7,10)
>> PlotIterationsP4(xkMat,10,13)
>> PlotIterationsP4(xkMat,13,17)
```

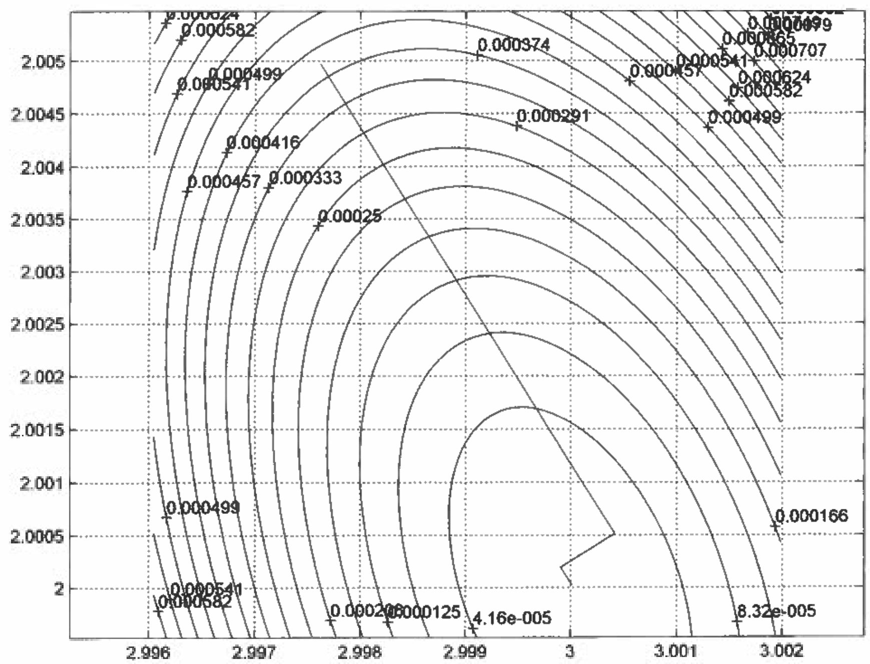
SD
Partial derivatives
By expressions
(analytical)

31

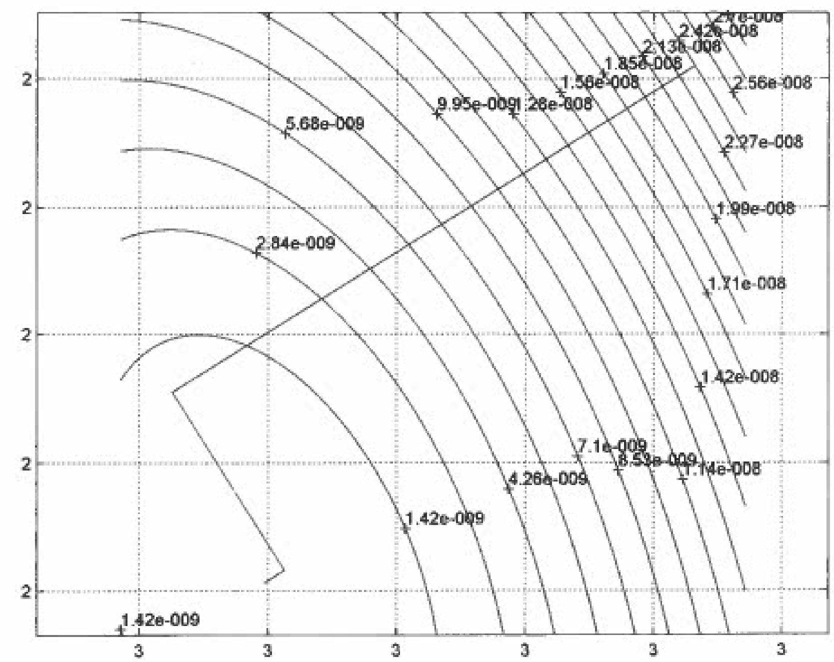
32



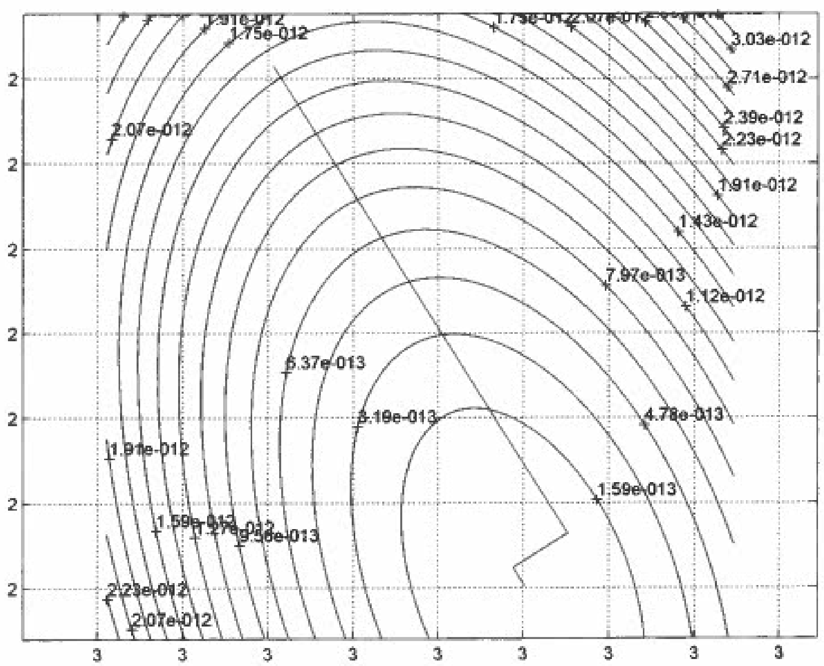
33


2.996
2.997
2.998
2.999
3
3.001
3.002
2
2.0005
2.001
2.0015
2.002
2.0025
2.003
2.0035
2.004
2.0045
2.005
0.000582
0.000374
0.000707
0.000291
0.000416
0.000333
0.000457
0.00025
0.000499
0.000166
4.16e-005
8.32e-005

34

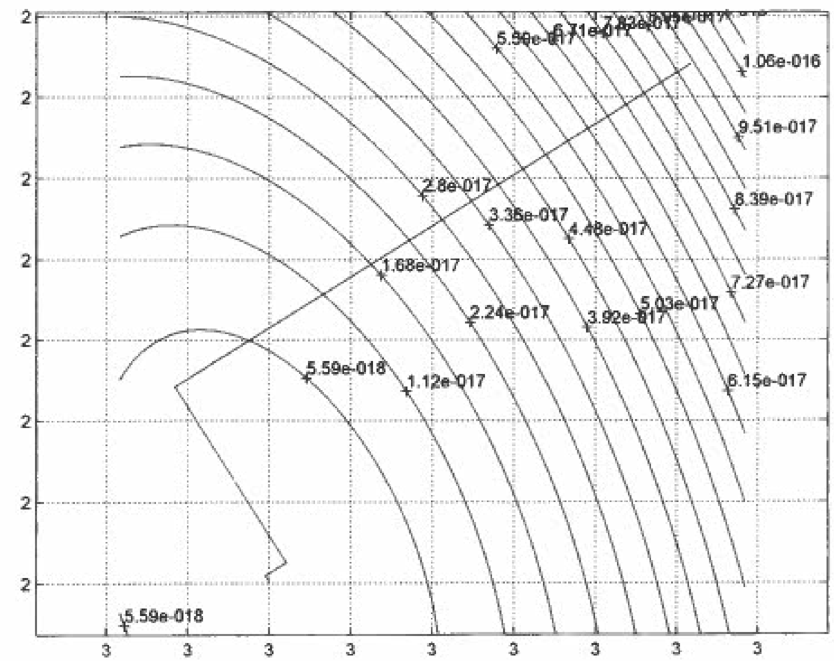

3
2
5.68e-009
2.84e-009
1.42e-009
4.26e-009
7.1e-009
1.14e-008
1.42e-008
1.71e-008
1.99e-008
2.27e-008
2.56e-008
1.56e-008

35



1.91e-012
1.75e-012
3.03e-012
2.71e-012
2.07e-012
2.39e-012
2.23e-012
1.91e-012
1.43e-012
7.97e-013
1.12e-012
6.37e-013
3.19e-013
4.78e-013
1.91e-012
1.59e-013
1.59e-012
9.56e-013
2.23e-012
2.07e-012

36



5.59e-017
1.06e-016
9.51e-017
2.8e-017
3.35e-017
4.48e-017
8.39e-017
1.68e-017
7.27e-017
2.24e-017
5.03e-017
3.92e-017
5.59e-018
1.12e-017
6.15e-017
5.59e-018