$1.000\ 000\ 000$	$10.095\ 266\ 145 \pm 0.643\ 500\ 904i$
$2.000\ 000\ 000$	$11.793\ 633\ 881 \pm 1.652\ 329\ 728i$
$3.000\ 000\ 000$	$13.992\ 358\ 137 \pm 2.518\ 830\ 070i$
$4.000\ 000\ 000$	$16.730\ 737\ 466 \pm 2.812\ 624\ 894i$
$4.999\ 999\ 928$	$19.502\ 439\ 400 \pm 1.940\ 330\ 347i$
$6.000\ 006\ 944$	
$6.999\ 697\ 234$	
$8.007\ 267\ 603$	
$8.917\ 250\ 249$	
$20.846\ 908\ 101$	

JACOBI-Basisverfahren

S0: *(Initialisierung) Setze* $\boldsymbol{A}^{(0)} = \boldsymbol{A}$, $k = 0$ *und berechne* $\omega_0 = \omega(\boldsymbol{A})$. *Wähle ein* $\varepsilon > 0$.

S1: *(Abbruchtest) Falls* $2\omega_k \leqq \varepsilon^2$ *STOPP.*

S2: *(Pivotwahl) Wähle Indizes* $p = p(k)$ *und* $q = q(k)$ *mit* $a_{pq}^{(k)} \neq 0$.

S3: *(Iterationsschritt)*

- **S4**: *Berechne* $t = t_k, c = c_k, s = s_k$ *und* $\tau = \tau_k$ *gemäß (9.5)-(9.7).*
- **S5**: *Berechne* $\boldsymbol{A}^{(k+1)}$ *aus* $\boldsymbol{A}^{(k)}$ *gemäß (9.8)-(9.12).*
- **S6**: *Berechne* $\omega_{k+1} = \omega_k - \left(a_{pq}^{(k)}\right)^2$.

Setze $k = k + 1$ *und gehe zu Schritt* **S1**.

Aufwand pro Schritt: $\sim 4n$ *Add./Sub.* + $\sim 3n$ *Mult./Div.* + *2 Quadratwurzeln.*

Basisverfahren der Vektoriteration

S0: *(Initialisierung) Wähle Vektor* $\boldsymbol{v}^{(0)} \in \mathbb{R}^n$ *mit* $\|\boldsymbol{v}^{(0)}\|_2 = 1$ *und setze* $k = 0$.

S1: *(Iteration) Berechne* $\boldsymbol{w}^{(k+1)} = \boldsymbol{A}\boldsymbol{v}^{(k)}$.

S2: *(Normierung) Setze* $\boldsymbol{v}^{(k+1)} = \boldsymbol{w}^{(k+1)}/\|\boldsymbol{w}^{(k+1)}\|_2$.

S3: *Setze* $k = k+1$ *und gehe zu Schritt* **S1**.

Aufwand pro Schritt:

- *Eine Auswertung* $Matrix \times Vektor$ *(***S1***)*
- $\sim n$ *Add./Sub. +* $\sim 2n$ *Mult./Div. + 1 Quadratwurzel (***S2***)*

Basisverfahren der Teilraumiteration

S0: *(Initialisierung) Wähle Teilraum* $\boldsymbol{\mathcal{Y}}^{(0)} \subset \mathbb{R}^n$ *mit*

$$\dim\left(\boldsymbol{\mathcal{Y}}^{(0)}\right) = p$$

und setze $k = 0$.

S1: *(Iteration) Definiere*

$$\boldsymbol{\mathcal{Y}}^{(k+1)} = \boldsymbol{A}\boldsymbol{\mathcal{Y}}^{(k)} = \left\{ \boldsymbol{A}\boldsymbol{y} \;\middle|\; \boldsymbol{y} \in \boldsymbol{\mathcal{Y}}^{(k)} \right\}.$$

S2: *Setze* $k = k+1$ *und gehe zu Schritt* **S1**.

Teilraumiteration mit orthonormalen Basen

S0: *(Initialisierung) Wähle ein m mit $p \leqq m \leqq n$ und eine spaltenorthonormale Startmatrix $\boldsymbol{V}^{(0)} \in \mathbb{R}^{n\times m}$. Setze $k = 0$.*

S1: *(Iteration) Berechne $\boldsymbol{W}^{(k+1)} = \boldsymbol{A}\boldsymbol{V}^{(k)}$.*

S2: *(Orthogonalisierung) Berechne spaltenorthonormale Matrix*

$$\boldsymbol{V}^{(k+1)} \in \mathbb{R}^{n\times m}$$

und eine obere Dreiecksmatrix $\boldsymbol{R}^{(k+1)} \in \mathbb{R}^{m\times m}$ mit

$$\boldsymbol{W}^{(k+1)} = \boldsymbol{V}^{(k+1)}\boldsymbol{R}^{(k+1)}.$$

S3: *Setze $k = k + 1$ und gehe zu Schritt* **S1**.

Aufwand pro Schritt:

- *m Auswertungen $Matrix \times Vektor$ (**S1**)*
- *$\sim K_1 n m^2$ Add./Mult. (**S2**)*
  ($K_1 = 1$ für SCHMIDT*sches Orthogonalisierungsverfahren)*

Rayleigh-Ritz-Algorithmus

Es seien die symmetrische Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *und die spaltenorthonormale Matrix* $\boldsymbol{V} \in \mathbb{R}^{n\times p}$ *gegeben.*

S0: • *Berechne* $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{V} \in \mathbb{R}^{n\times p}$.
• *Berechne* $\boldsymbol{P} = \boldsymbol{V}^T\boldsymbol{B} \in \mathbb{R}^{p\times p}$.

S1: *Berechne die Eigenwertzerlegung* $\boldsymbol{P} = \boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^T$.

S2: *Berechne* $\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{X}$.

Aufwand:

- m *Auswertungen* $Matrix \times Vektor$ $+ \sim np^2/2$ *Add./Mult.* (**S0**)
- $\sim K_2 p^3$ *Add./Mult.* (**S1**) *(*K_2 *hängt vom verwendeten Verfahren ab)*
- $\sim np^3$ *Add./Mult.* (**S2**)

Basisverfahren der Inversen Iteration

S0: *(Initialisierung) Wähle ein* $\mu \neq \lambda_j, j = 1, \ldots, n$ *und einen Startvektor* $\boldsymbol{v}^{(0)} \in \mathbb{R}^n$ *mit* $\|\boldsymbol{v}^{(0)}\| = 1$*. Setze* $k = 0$*.*

S1: *(Iteration) Berechne* $\boldsymbol{w}^{(k+1)}$ *aus*

$$\boldsymbol{A}_\mu \boldsymbol{w}^{(k+1)} = (\boldsymbol{A} - \mu \boldsymbol{I})\boldsymbol{w}^{(k+1)} = \boldsymbol{v}^{(k)}.$$

S2: *(Normierung) Setze* $\boldsymbol{v}^{(k+1)} = \boldsymbol{w}^{(k+1)} / \|\boldsymbol{w}^{(k+1)}\|$*.*

S3: *Setze* $k = k + 1$ *und gehe zu Schritt* **S1***.*

Inverse Iteration bei guter Eigenwertnäherung

Es sei μ *ein im Sinne von*

$$|\mu - \lambda| \leqq \mathrm{eps} F_0 \|\boldsymbol{A}\| = \varepsilon_0$$

gute Näherung für den Eigenwert λ *von* $\boldsymbol{A}$*.*

S0: *(Initialisierung)*

- *Berechne Zerlegung* $\boldsymbol{A}_\mu = \boldsymbol{P}_\mu^T \boldsymbol{L}_\mu \boldsymbol{U}_\mu$.
- *Berechne* $\boldsymbol{w}^{(0)}$ *aus* $\boldsymbol{U}_\mu \boldsymbol{w}^{(0)} = \boldsymbol{e} = (1, \ldots, 1)^T$.
- *Wähle ein* $\varepsilon > 0$ *und setze* $k = 0$.

S1: *(Iteration) Berechne* $\boldsymbol{w}^{(k+1)}$ *aus*

$$\boldsymbol{A}_\mu \boldsymbol{w}^{(k+1)} = \boldsymbol{P}_\mu^T \boldsymbol{L}_\mu \boldsymbol{U}_\mu \boldsymbol{w}^{(k+1)} = \boldsymbol{v}^{(k)}.$$

S2: *(Normierung) Setze*

$$\omega_{k+1} = \|\boldsymbol{w}^{(k+1)}\|, \quad \boldsymbol{v}^{(k+1)} = \boldsymbol{w}^{(k+1)} / \omega_{k+1}.$$

S3: *(Abbruchtest) Falls* $\varepsilon \cdot \omega_{k+1} < 1$ *so setze* $k = k + 1$ *und gehe zu Schritt* **S1**.

Aufwand:

- $\sim n^3/3$ *Add./Mult.* (**S0**)
- $\sim n^2$ *Add./Mult. pro Schritt für vollbesetztes* $\boldsymbol{A}$ (**S1** *und* **S2**)

RAYLEIGH-Quotienten-Iteration

S0: *(Initialisierung) Wähle* $\boldsymbol{v}^{(0)}$ *mit* $\|\boldsymbol{v}^{(0)}\|_2 = 1$ *und ein* $\varepsilon > 0$. *Setze* $k = 0$.

S1: *(Berechnen der Verschiebung) Berechne*

$$\varrho_k = RQ(\boldsymbol{v}^{(k)}) = {\boldsymbol{v}^{(k)}}^T \boldsymbol{A}\boldsymbol{v}^{(k)}.$$

S2: *(Inverse Iteration) Berechne* $\boldsymbol{w}^{(k+1)}$ *aus*

$$(\boldsymbol{A} - \varrho_k \boldsymbol{I})\boldsymbol{w}^{(k+1)} = \boldsymbol{v}^{(k)}.$$

S3: *(Normierung) Setze* $\omega_{k+1} = \|\boldsymbol{w}^{(k+1)}\|_2$ *und* $\boldsymbol{v}^{(k+1)} = \boldsymbol{w}^{(k+1)}/\omega_{k+1}$.

S4: *(Abbruchtest) Falls* $\varepsilon \cdot \omega_{k+1} < 1$ *so setze* $k = k + 1$ *und gehe zu Schritt* **S1**.

Aufwand pro Schritt:

- $\sim n^2$ *Add./Mult. (***S1***)*
- $\sim K_1 n^3$ *Add./Mult. für vollbesetztes* $\boldsymbol{A}$ *(***S2***)*
- $\sim K n$ *Add./Mult. für tridiagonales* $\boldsymbol{A}$ *(***S2***)*
- $\sim n$ *Add. +* $\sim 2n$ *Mult. + 1 Quadratwurzel (***S3***)*

Tridiagonalisierung einer symmetrischen Matrix mit HOUSEHOLDER-Spiegelungen

S0: *(Initialisierung) Setze* $k = 1$ *und* $\boldsymbol{A}^{(1)} = \boldsymbol{A}$.

S1: *(Berechnen der* HOUSEHOLDER*-Spiegelung) Berechne* $\boldsymbol{H}^{(k)}$ *nach (9.19)-(9.26).*
(Iteration) Berechne

$$\boldsymbol{A}^{(k+1)} = \boldsymbol{H}^{(k)}\boldsymbol{A}^{(k)}\boldsymbol{H}^{(k)}$$

nach (9.28)-(9.37).

S2: *(Abbruch) Falls* $k < n-2$ *so setze* $k = k+1$ *und gehe zu Schritt* **S1**.

Aufwand: $\sim 2n^3/3$ *Add./Mult. und* $\sim n$ *Quadratwurzeln falls* $\boldsymbol{A}$ *voll besetzt ist.*

LANCZOS-Algorithmus

S0: *(Initialisierung) Wähle Vektor* $\boldsymbol{q} \in \mathbb{R}^n$ *mit* $\|\boldsymbol{q}\|_2 = 1$.
Setze $\boldsymbol{q}_1 = \boldsymbol{q}$ *und* $k = 1$, $b_1 \boldsymbol{q}_0 = \boldsymbol{o}$.

S1 *(Iteration) Berechne*

$$\begin{aligned} a_k &= \boldsymbol{q}_k^T \boldsymbol{A} \boldsymbol{q}_k, \\ \boldsymbol{r}_k &= \boldsymbol{A}\boldsymbol{q}_k - a_k \boldsymbol{q}_k - b_k \boldsymbol{q}_{k-1}, \\ b_{k+1} &= \|\boldsymbol{r}_k\|_2. \end{aligned}$$

S2: *(Abbruchtest)*

- *Falls* $b_{k+1} \neq 0$, *so berechne* $\boldsymbol{q}_{k+1} = \boldsymbol{r}_k / b_{k+1}$, *setze* $k = k+1$ *und gehe zu Schritt* **S1**.
- *Falls* $b_{k+1} = 0$, *so setze* $m = k$. *STOPP*

Aufwand pro Schritt:

- *Eine Auswertung Matrix* $\times$ *Vektor*
- $\sim 4n$ *Add./Sub.* + $\sim 5n$ *Mult./Div.* + *1 Quadratwurzel.*

QR-Algorithmus ohne Verschiebungen

S0: *(Initialisierung) Setze*

$$\boldsymbol{A}^{(0)} = \boldsymbol{A}, \quad \boldsymbol{V}^{(0)} = \boldsymbol{I}, \quad k = 0.$$

S1: *(QR-Zerlegung) Berechne eine orthogonale Matrix* $\boldsymbol{Q}^{(k)}$ *und eine obere Dreiecksmatrix* $\boldsymbol{R}^{(k)}$ *mit*

$$\boldsymbol{A}^{(k)} = \boldsymbol{Q}^{(k)}\boldsymbol{R}^{(k)}.$$

S2: *(Iteration) Berechne*

$$\boldsymbol{A}^{(k+1)} = \boldsymbol{R}^{(k)}\boldsymbol{Q}^{(k)}, \quad \boldsymbol{V}^{(k+1)} = \boldsymbol{V}^{(k)}\boldsymbol{Q}^{(k)}.$$

S3: *Setze* $k = k+1$ *und gehe zu Schritt* **S1**.

QR-Algorithmus mit Verschiebungen

S0: *(Initialisierung) Setze* $\boldsymbol{A}^{(0)} = \boldsymbol{A}, \boldsymbol{V}^{(0)} = \boldsymbol{I}$ *und* $k = 0$.

S1: *(Spektralverschiebung) Wähle Verschiebungsparameter* μ_k.

S2: *(QR-Zerlegung) Berechne eine orthogonale Matrix* $\boldsymbol{Q}^{(k)}$ *und eine obere Dreiecksmatrix* $\boldsymbol{R}^{(k)}$ *mit*

$$\boldsymbol{A}^{(k)} - \mu_k \boldsymbol{I} = \boldsymbol{Q}^{(k)} \boldsymbol{R}^{(k)}.$$

S3: *(Iteration) Berechne*

$$\boldsymbol{A}^{(k+1)} = \boldsymbol{R}^{(k)} \boldsymbol{Q}^{(k)} + \mu_k \boldsymbol{I}, \quad \boldsymbol{V}^{(k+1)} = \boldsymbol{V}^{(k)} \boldsymbol{Q}^{(k)}.$$

S3 *Setze* $k = k + 1$ *und gehe zu Schritt* **S1**.

QR-Algorithmus mit Rayleigh-Quotienten-Verschiebungen

S0: *(Initialisierung) Setze*

$$A^{(0)} = A, \quad V^{(0)} = I, \quad k = 0.$$

S1: *(Spektralverschiebung) Wähle Verschiebungsparameter* $\mu_k = a_{nn}^{(k)}$.
(QR-Zerlegung) Berechne eine orthogonale Matrix $Q^{(k)}$ *und eine obere Dreiecksmatrix* $R^{(k)}$ *mit*

$$A^{(k)} - \mu_k I = Q^{(k)} R^{(k)}.$$

S2: *(Iteration) Berechne*

$$A^{(k+1)} = R^{(k)} Q^{(k)} + \mu_k I, \quad V^{(k+1)} = V^{(k)} Q^{(k)}.$$

S3: *Setze* $k = k + 1$ *und gehe zu Schritt* **S1**.

QR-Algorithmus mit WILKINSON-Verschiebungen

S0: *(Initialisierung) Setze*

$$A^{(0)} = A, \quad V^{(0)} = I, \quad k = 0.$$

S1: *(Spektralverschiebung)*

- *Berechne die Eigenwerte μ' und μ'' der symmetrischen Matrix*

$$P^{(k)} = \begin{pmatrix} a^{(k)}_{n-1,n-1} & a^{(k)}_{n-1,n} \\ a^{(k)}_{n,n-1} & a^{(k)}_{n,n} \end{pmatrix} \in \mathbb{R}^{2\times 2}.$$

- *Ordne μ' und μ'' so, dass*

$$|\mu' - a^{(k)}_{n,n}| < |\mu'' - a^{(k)}_{n,n}|$$

bzw.

$$|\mu'| < |\mu''|$$

im Falle

$$|\mu' - a_{n,n}^{(k)}| = |\mu'' - a_{n,n}^{(k)}|$$

gilt.

- *Setze* $\mu_k = \mu'$.

S2: *(QR-Zerlegung) Berechne eine orthogonale Matrix* $\boldsymbol{Q}^{(k)}$ *und eine obere Dreiecksmatrix* $\boldsymbol{R}^{(k)}$ *mit*

$$\boldsymbol{A}^{(k)} - \mu_k \boldsymbol{I} = \boldsymbol{Q}^{(k)} \boldsymbol{R}^{(k)}.$$

S3: *(Iteration) Berechne*

$$\boldsymbol{A}^{(k+1)} = \boldsymbol{R}^{(k)} \boldsymbol{Q}^{(k)} + \mu_k \boldsymbol{I}, \quad \boldsymbol{V}^{(k+1)} = \boldsymbol{V}^{(k)} \boldsymbol{Q}^{(k)}.$$

S4: *Setze* $k = k+1$ *und gehe zu Schritt* **S1**.

Expliziter QR-Schritt

S0: *(Initialisierung) Wähle Verschiebungsparameter* μ_k *und bilde*

$$\bar{\boldsymbol{A}} = \boldsymbol{A}^{(k)} - \mu_k \boldsymbol{I}.$$

Setze $\boldsymbol{G}_{01} = \boldsymbol{I}$ *und* $\bar{\boldsymbol{V}} = \boldsymbol{V}^{(k)}$.

S1: *(Transformation) Für* $j = 1, \ldots, n-1$ *führe aus:*

- *Lege* $\boldsymbol{G}_{j,j+1}$ *so fest, dass* $(\boldsymbol{G}_{j,j+1}\bar{\boldsymbol{A}})_{j+1,j} = 0$ *gilt.*
- *Bilde*

$$\bar{\boldsymbol{A}} = \boldsymbol{G}_{j,j+1}\bar{\boldsymbol{A}},\ \bar{\boldsymbol{A}} = \bar{\boldsymbol{A}}\boldsymbol{G}_{j-1,j} \quad \bar{\boldsymbol{V}} = \bar{\boldsymbol{V}}\boldsymbol{G}^T_{j,j+1}.$$

S2: *Berechne*

$$\bar{\boldsymbol{A}} = \bar{\boldsymbol{A}}\boldsymbol{G}^T_{n-1,n}, \quad \boldsymbol{A}^{(k+1)} = \bar{\boldsymbol{A}} + \mu_k \boldsymbol{I}$$

und setze

$$\boldsymbol{V}^{(k+1)} = \bar{\boldsymbol{V}}.$$

Aufwand:

- $\sim 4n$ *Add./Sub.*, $\sim 13n$ *Mult./Div. und* n *Quadratwurzeln zum Berechnen der Matrix* $\boldsymbol{A}^{(k+1)}$.
- $\sim 3n^2$ *Add./Sub. und* $\sim 3n^2$ *Mult./Div. zum Berechnen von* $\boldsymbol{V}^{(k+1)}$.

Impliziter QR-Schritt

S0: *(Initialisierung) Wähle Verschiebungsparameter* μ_k. *Setze* $\bar{\boldsymbol{A}} = \boldsymbol{A}^{(k)}$ *und* $\bar{\boldsymbol{V}} = \boldsymbol{V}^{(k)}$.

S1: *(Starttransformation)*

- *Lege* $\boldsymbol{G}_{12}$ *so fest, dass* $[\boldsymbol{G}_{12}(\bar{\boldsymbol{A}} - \mu_k \boldsymbol{I})]_{21} = 0$ *gilt.*
- *Bilde* $\bar{\boldsymbol{A}} = \boldsymbol{G}_{12}\bar{\boldsymbol{A}}\boldsymbol{G}_{12}^T$ *und* $\bar{\boldsymbol{V}} = \bar{\boldsymbol{V}}\boldsymbol{G}_{12}^T$.

S2: *(Resttransformation) Für* $j = 2, \ldots, n-1$ *führe aus:*

- *Lege* $\boldsymbol{G}_{j,j+1}$ *so fest, dass* $(\boldsymbol{G}_{j,j+1}\bar{\boldsymbol{A}})_{j+1,j-1} = 0$ *gilt.*
- *Bilde* $\bar{\boldsymbol{A}} = \boldsymbol{G}_{j,j+1}\bar{\boldsymbol{A}}\boldsymbol{G}_{j,j+1}^T$ *und* $\bar{\boldsymbol{V}} = \bar{\boldsymbol{V}}\boldsymbol{G}_{j,j+1}^T$.

S3: *Setze* $\boldsymbol{A}^{(k+1)} = \bar{\boldsymbol{A}}$ *und* $\boldsymbol{V}^{(k+1)} = \bar{\boldsymbol{V}}$.

Aufwand:

- $\sim 6n$ *Add./Sub.* + $\sim 11n$ *Mult./Div.* + n *Quadratwurzeln für das Berechnen von* $\boldsymbol{A}^{(k+1)}$.
- $\sim 3n^2$ *Add./Sub.* + $\sim 3n^2$ *Mult./Div. für das Berechnen von* $\boldsymbol{V}^{(k+1)}$.

QR-Algorithmus für Tridiagonalmatrizen

Gegeben seien die (n,n)-Tridiagonalmatrix

$$\boldsymbol{T} = \mathrm{trid}(a_1, \ldots, a_n, b_2, \ldots, b_n)$$

und die orthogonale (n,n)-Matrix $\boldsymbol{V}$.

S0: *(Initialisierung) Setze k=1, q=n.*

S1: *(Abbruchtest) Falls $q = 1$: STOPP.*

S2: *(Nullsetzen kleiner Nichtdiagonalelemente und Festlegen des zu aktuellen Diagonalblocks*

$$\boldsymbol{T}^{(p,q)} = \mathrm{trid}(a_p, \ldots, a_q, b_{p+1}, \ldots, b_q)$$

der Dimension $m = q - p + 1$ von $\boldsymbol{T}$)

Für p=q, q-1, ...,2 führe aus:

- *Ist $|b_p|$ hinreichend klein, so*
  - *setze $b_p = 0$.*
  - *Ist $p = q$, so setze $q = q - 1$ und gehe zu Schritt* **S1**.

- *Ist $p<q$, so gehe zu Schritt* **S3**.

- *Setze $p=1$.*

S3: *(k-ter QR-Schritt) Führe QR-Schritt mit* WILKINSON-*Verschiebung für*

$$\boldsymbol{A}^{(k)}=\boldsymbol{T}^{(p,q)}\in\mathbb{R}^{m\times m}\quad,\quad\boldsymbol{V}^{(k)}=\boldsymbol{V}^{(p,q)}\in\mathbb{R}^{n\times m}$$

auf dem Platz von $\boldsymbol{T}$ bzw. $\boldsymbol{V}$ durch. Dabei bezeichnet $\boldsymbol{V}^{(p,q)}$ die aus den Spalten p bis q von $\boldsymbol{V}$ gebildete spaltenorthonormale Matrix.

S4: *Setze $k=k+1$ und gehe zu Schritt* **S1**.

Lösen der Normalgleichungen mit CHOLESKY-Zerlegung

Es ist das lineare Ausgleichsproblem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2$$

mit der spaltenregulären Matrix (m,n)-$\boldsymbol{A}$ *zu lösen.*

S0 *Berechne die* (n,n)-*Matrix* $\boldsymbol{M} = \boldsymbol{A}^T\boldsymbol{A}$.

S1 *Bestimme eine* CHOLESKY-*Zerlegung* $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$ *mit einer unteren* (n,n)-*Dreiecksmatrix* $\boldsymbol{L}$ *mit positiven Diagonalelementen.*

S2 *Berechne* $\boldsymbol{z} = \boldsymbol{A}^T\boldsymbol{y}$.

S3 *Löse die Dreieckssysteme* $\boldsymbol{L}\boldsymbol{w} = \boldsymbol{z}$ *und* $\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{w}$.

Aufwand:

S0 $\sim mn^2/2$ *Additionen/Multiplikationen,*

S1 $\sim n^3/6$ *Additionen/Multiplikationen*$+$ n *Quadratwurzeln,*

S2 $\sim mn$ *Additionen/Multiplikationen,*

S3 $\sim n^2$ *Additionen/Multiplikationen*

Normalgleichungsverfahren mit Nachiteration

Es ist das lineare Ausgleichsproblem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2$$

mit der spaltenregulären (m,n)*-Matrix* $\boldsymbol{A}$ *zu lösen.*

S0 *Berechne die* (n,n)*-Matrix* $\boldsymbol{M} = \boldsymbol{A}^T\boldsymbol{A}$.
Bestimme eine CHOLESKY*-Zerlegung* $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$ *mit einer unteren* (n,n)*-Dreiecksmatrix* $\boldsymbol{L}$ *mit positiven Diagonalelementen.*
Setze $\boldsymbol{x}^{(0)} = \boldsymbol{o}$ *und* $k = 0$.

S1 *Berechne* $\boldsymbol{r}^{(k)} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^{(k)}$ *und* $\boldsymbol{g}^{(k)} = \boldsymbol{A}^T\boldsymbol{r}^{(k)}$.

S2 *Löse die Dreieckssysteme*

$$\boldsymbol{L}\boldsymbol{w} = \boldsymbol{g}^{(k)}, \quad \boldsymbol{L}^T\boldsymbol{h}^{(k)} = \boldsymbol{w}.$$

S3 *Setze* $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{h}^{(k)}$, $k = k+1$ *und gehe zu Schritt* **S1**

Lösen eines linearen Ausgleichsproblems mit HOUSEHOLDER-Transformationen

Es ist das lineare Ausgleichsproblem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2$$

mit der spaltenregulären (m,n)*-Matrix* $\boldsymbol{A}$ *zu lösen.*

{Initialisierung}
Wähle eine Genauigkeitsschranke $\varepsilon > 0$.
{QR-Zerlegung}
for $k = 1$ ***to*** n ***do***
  {Berechnen der Transformationsmatrix $\boldsymbol{H}^{(k)}$*}*
  $\varrho_k = \sqrt{a_{kk}^2 + a_{k+1,k}^2 + \cdots + a_{mk}^2}$
  if $\varrho_k \leqq \varepsilon$ ***then***
    STOPP
  endif
  if $a_{kk} > 0$ ***then***
    $\varrho_k = -\varrho_k$

endif

$a_{kk} = a_{kk} - \varrho_k$

$\gamma_k = -\varrho_k \cdot a_{kk}$

$\{Transformation\ der\ Restmatrix\}$

for $j = k+1$ ***to*** n ***do***

   $\beta = 0$

   for $i = k$ ***to*** m ***do***

      $\beta = \beta + a_{ik} \cdot a_{ij}$

   endfor

   $\beta = \beta / \gamma_k$

   for $i = k$ ***to*** m ***do***

      $a_{ij} = a_{ij} - \beta \cdot a_{ik}$

   endfor

   $\{Transformation\ des\ Vektors\ \boldsymbol{y}\}$

   $\beta = 0$

   for $i = k$ ***to*** m ***do***

      $\beta = \beta + a_{ik} \cdot y_i j$

   endfor

   $\beta = \beta / \gamma_k$

   for $i = k$ ***to*** m ***do***

      $y_i = y_i - \beta \cdot a_{ik}$

   endfor

endfor

endfor
$\varrho_n = a_{nn}$
$\{$*Lösen von* $\boldsymbol{Rx} = \boldsymbol{c}\}$
for $k = n$ **to** 1 **step** -1 **do**
  for $i = k + 1$ **to** n **do**
    $y_k = y_k - y_i \cdot a_{ki}$
  endfor
  $y_k = y_k / \varrho_k$
endfor

Aufwand:
Transf.: $\sim(m - n/3)n^2$ *Additionen/Multiplikationen,* n *Quadratwurzeln,*
Lösen: $\sim n^2/2$ *Additionen/Multiplikationen.*

Nachiteration für HOUSEHOLDER-Verfahren bei Ausgleichsproblemen

Es ist das lineare Ausgleichsproblem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2$$

mit der spaltenregulären (m, n)*-Matrix* $\boldsymbol{A}$ *zu lösen.*

S0 *Berechne die QR-Zerlegung* $\boldsymbol{A} = \boldsymbol{Q}\begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{pmatrix}$ *mit einer orthogonalen* (m, m)*-Matrix* $\boldsymbol{Q}$ *und einer oberen* (n, n)*-Dreiecksmatrix* $\boldsymbol{R}$.
Setze $\boldsymbol{x}^{(0)} = \boldsymbol{o}$, $\boldsymbol{r}^{(0)} = \boldsymbol{y}$ *und* $k = 0$.

S1 *Berechne* $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{r}^{(k)} - \boldsymbol{A}\boldsymbol{x}^{(k)}$ *und* $\boldsymbol{f} = -\boldsymbol{A}^T\boldsymbol{r}^{(k)}$.
Berechne

$$\boldsymbol{Q}^T\boldsymbol{e} = \begin{pmatrix} \mu \\ \boldsymbol{v} \end{pmatrix}, \quad \mu \in \mathbb{R}^n, \quad \boldsymbol{v} \in \mathbb{R}^{m-n}.$$

S2 *Löse das Dreieckssystem* $\boldsymbol{R}^T\boldsymbol{w} = \boldsymbol{f}$.

Löse das Dreieckssystem $\boldsymbol{Rh} = \mu - \boldsymbol{w}$.

S3 *Berechne*

$$\boldsymbol{s} = \boldsymbol{Q} \begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{v} \end{pmatrix}.$$

Setze $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{h}$, $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} + \boldsymbol{s}$, $k = k+1$ *und gehe zu Schritt* **S1**

Modellalgorithmus für unrestringierte Minimierung

(Initialisierung) Wähle einen Startpunkt $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ *und setze* $k = 0$.

S0 *(Abbruchkriterium) Ist* $f'(\boldsymbol{x}^{(k)};\boldsymbol{p}) \geqq 0$ *für alle Richtungen* $\boldsymbol{p} \in \mathbb{R}^n$*, dann STOPP;* $\boldsymbol{x}^{(k)}$ *ist stationäre Lösung.*

S1 *(Richtungswahl) Wähle Abstiegsrichtung* $\boldsymbol{p}^{(k)}$ *mit*

$$f'(\boldsymbol{x}^{(k)};\boldsymbol{p}^{(k)}) < 0.$$

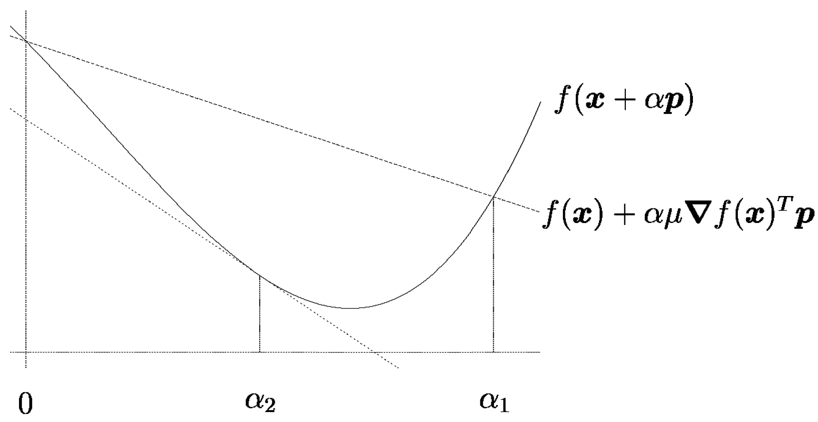
S2 *(Schrittweitenbestimmung) Bestimme Schrittweite* α_k*, so dass*

$$f(\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}) < f(\boldsymbol{x}^{(k)}).$$

S3 *(Iterationsschritt) Setze*

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}, \quad k = k+1$$

und gehe zu Schritt **S1**.



$f(\boldsymbol{x} + \alpha \boldsymbol{p})$
$f(\boldsymbol{x}) + \alpha\mu \boldsymbol{\nabla} f(\boldsymbol{x})^T \boldsymbol{p}$
0
α_2
α_1

Verfahren der Oren-Luenberger-Klasse

S0 *(Initialisierung) Wähle einen Startvektor* $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ *und eine positiv definite Startmatrix* $\boldsymbol{H}_0 \in \mathbb{R}^{n\times n}$ *(z.B.* $\boldsymbol{H}_0 = \boldsymbol{I}$*). Setze* $k = 0$*.*

S1 *(Abbruchbedingung) Falls* $\nabla f(\boldsymbol{x}^{(k)}) = \boldsymbol{o}$*, so STOPP;* $\boldsymbol{x}^{(k)}$ *ist stationäre Lösung.*

S2 *(Richtungswahl) Berechne eine Abstiegsrichtung*

$$\boldsymbol{p}^{(k)} = -\boldsymbol{H}_k \nabla f(\boldsymbol{x}^{(k)}).$$

S3 *(Schrittweitenbestimmung) Bestimme eine Schrittweite* α_k
als exakte, Powell- *oder* Armijo-*Schrittweite.*

S4 *(Iterationsschritt) Setze*

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)},\\
\boldsymbol{s}^{(k)} &= \alpha_k \boldsymbol{p}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)},\\
\boldsymbol{q}^{(k)} &= \nabla f(\boldsymbol{x}^{(k+1)}) - \nabla f(\boldsymbol{x}^{(k)}).
\end{aligned}$$

Wähle Konstanten $\gamma_k > 0$ *und* $\Theta_k \geqq 0$ *und berechne*

$$\boldsymbol{H}_{k+1} = \Psi(\boldsymbol{H}_k, \boldsymbol{s}^{(k)}, \boldsymbol{q}^{(k)}; \gamma_k, \Theta_k)$$

$wobei$

$$
\begin{aligned}
\Psi(\boldsymbol{H},\boldsymbol{s},\boldsymbol{q};\gamma,\Theta)=\gamma\boldsymbol{H} \quad &+ \quad \left(1+\gamma\Theta\frac{\boldsymbol{q}^T\boldsymbol{H}\boldsymbol{q}}{\boldsymbol{s}^T\boldsymbol{q}}\right)\frac{\boldsymbol{s}\boldsymbol{s}^T}{\boldsymbol{s}^T\boldsymbol{q}} \\
&- \quad \gamma\frac{1-\Theta}{\boldsymbol{q}^T\boldsymbol{H}\boldsymbol{q}}\boldsymbol{H}\boldsymbol{q}\cdot\boldsymbol{q}^T\boldsymbol{H} \\
&- \quad \frac{\gamma\Theta}{\boldsymbol{s}^T\boldsymbol{q}}(\boldsymbol{s}\cdot\boldsymbol{q}^T\boldsymbol{H}+\boldsymbol{H}\boldsymbol{q}\cdot\boldsymbol{s}^T).
\end{aligned}
$$

Aufdatierung einer LDL^T-Zerlegung $\bar{\boldsymbol{B}} = \boldsymbol{B} + \boldsymbol{v}\boldsymbol{v}^T$

Von der positiv definiten (n,n)-Matrix $\boldsymbol{B}$ sei eine Zerlegung

$$\boldsymbol{B} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$$

mit einer unteren (n,n)-Einsdreiecksmatrix $\boldsymbol{L}$ und einer (n,n)-Diagonalmatrix $\boldsymbol{D}$ bekannt.
Es sind eine untere Einsdreiecksmatrix $\bar{\boldsymbol{L}}$ und eine Diagonalmatrix $\bar{\boldsymbol{D}}$ derart zu berechnen, dass

$$\bar{\boldsymbol{B}} = \boldsymbol{B} + \boldsymbol{v}\boldsymbol{v}^T = \bar{\boldsymbol{L}}\bar{\boldsymbol{D}}\bar{\boldsymbol{L}}^T$$

gilt.

$t_0 = 1$; $\boldsymbol{v}^{(1)} = \boldsymbol{v}$
for $j = 1$ ***to*** n ***do***
  $u_j = v_j^{(j)}$
  $t_j = t_{j-1} + u_j^2/d_j$
  $d_j = d_j t_j / t_{j-1}$
  $\beta_j = u_j/(d_j t_j)$
  for $k = j+1$ ***to*** n ***do***

$$v_k^{(j+1)} = v_k^{(j)} - u_j l_{kj}$$
$$\bar{l}_{kj} = l_{kj} + \beta_j v_k^{(j+1)}$$

endfor

endfor

Aufdatierung einer LDL^T-Zerlegung $\bar{\boldsymbol{B}} = \boldsymbol{B} - \boldsymbol{v}\boldsymbol{v}^T$

Von der positiv definiten (n,n)-Matrix $\boldsymbol{B}$ sei eine Zerlegung $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$ mit einer unteren Einsdreiecksmatrix $\boldsymbol{L}$ und einer Diagonalmatrix $\boldsymbol{D}$ bekannt.
Es sind eine untere Einsdreiecksmatrix $\bar{\boldsymbol{L}}$ und eine Diagonalmatrix $\bar{\boldsymbol{D}}$ derart zu berechnen, dass

$$\bar{\boldsymbol{B}} = \boldsymbol{B} - \boldsymbol{v}\boldsymbol{v}^T = \bar{\boldsymbol{L}}\bar{\boldsymbol{D}}\bar{\boldsymbol{L}}^T$$

gilt.

Wähle ein $\delta > 0$
Löse das Gleichungssystem $\boldsymbol{L}\boldsymbol{u} = \boldsymbol{v}$
$t_{n+1} = 1 - \boldsymbol{u}^T\boldsymbol{D}^{-1}\boldsymbol{u}$
if $t_{n+1} < \delta$ ***then***
  $t_{n+1} = \delta$
endif
for $j = n$ ***to*** 1 ***step*** -1 ***do***
  $t_j = t_{j+1} + u_j^2/d_j$
  $d_j = d_j t_{j+1}/t_j$

$\beta_j = -u_j/(d_j t_{j+1})$
$v_j^{(j)} = u_j$
for $k = j+1$ ***to*** n ***do***
  $v_k^{(j+1)} = v_k^{(j)} - u_j l_{kj}$
  $l_{kj} = l_{kj} + \beta_j v_k^{(j+1)}$
endfor
endfor

Trust-region-Verfahren

S0 *Wähle Konstanten* $0 < \varrho_1 < \varrho_2 < 1$, $0 < \sigma_1 < 1 < \sigma_2$ *und* $\Delta_0 > 0$.
Wähle einen Startpunkt $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ *und setze* $k = 0$.

S1 *Berechne die Lösung* $\boldsymbol{p}^*$ *der Aufgabe*

$$\min_{\|\boldsymbol{p}\| \leqq \Delta_k} f_{\boldsymbol{x}^{(k)}}(\boldsymbol{p}).$$

S2 *Falls* $f(\boldsymbol{x}^{(k)}) = f_{\boldsymbol{x}^{(k)}}(\boldsymbol{p}^*)$ *STOPP*
($\boldsymbol{x}^{(k)}$ *ist stationäre Lösung.)*

S3 *Berechne*

$$r = \frac{f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^{(k)} + \boldsymbol{p}^*)}{f(\boldsymbol{x}^{(k)}) - f_{\boldsymbol{x}^{(k)}}(\boldsymbol{p}^*)}.$$

Falls $r \geqq \varrho_1$, *so* $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{p}^*$.
Falls $r < \varrho_1$, *wähle ein* $\Delta_{k+1} \in (0, \sigma_1 \Delta_k]$.
Falls $\varrho_1 \leqq r < \varrho_2$, *wähle ein* $\Delta_{k+1} \in [\sigma_1 \Delta_k, \Delta_k]$
Falls $\varrho_2 \leqq r$, *wähle ein* $\Delta_{k+1} \in [\Delta_k, \sigma_2 \Delta_k]$

S4 *Setze* $k = k + 1$ *und gehe zu Schritt* **S1**.