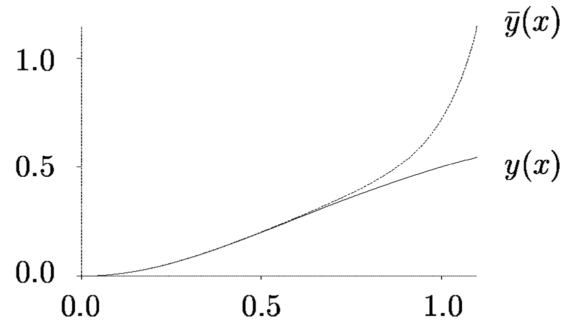

$\bar{y}(x)$
$y(x)$
1.0
0.5
0.0
0.0
0.5
1.0

EULERsches Polygonzugverfahren

S0: *Setze* $\boldsymbol{\eta}_0 = \boldsymbol{y}_0$ *und* $k = 0$.
S1: *Berechne*

$$\boldsymbol{\eta}_{\boldsymbol{k}+1} = \boldsymbol{\eta}_{\boldsymbol{k}} + h\boldsymbol{f}(x_k, \boldsymbol{\eta}_{\boldsymbol{k}}),$$

$$x_{k+1} = x_k + h.$$

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

Allgemeines Einschrittverfahren

S0: *Setze* $\boldsymbol{\eta}_0 = \boldsymbol{y}_0$ *und* $k = 0$.
S1: *Berechne*

$$\boldsymbol{\eta}_{\boldsymbol{k}+1} = \boldsymbol{\eta}_{\boldsymbol{k}} + h\Phi(x_k, \boldsymbol{\eta}_{\boldsymbol{k}}; h)$$

$$x_{k+1} = x_k + h.$$

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

Schrittweitensteuerung bei ESV

Es sei ein ESV mit der Konsistenzordnung p zum Lösen des AWP

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad \boldsymbol{y}(x_0) = \boldsymbol{y}_0$$

gegeben.

S0: *Wähle Grundschrittweite H und eine zu erreichende Genauigkeit ε.*

S1: *Berechne mit dem gegebenen ESV die beiden Näherungen*

$$\boldsymbol{\eta}(x_0 + H; H), \quad \boldsymbol{\eta}(x_0 + H; H/2).$$

S2: *Berechne*

$$\frac{H}{h} = \sqrt[p+1]{\frac{2^p}{2^p - 1} \frac{\left\| \boldsymbol{\eta}(x_0 + H; H) - \boldsymbol{\eta}\left(x_0 + H; \frac{H}{2}\right) \right\|}{\varepsilon}}.$$

S3: *Ist $H/h \gg 2$, so setze $H = 2h$ und gehe zu Schritt* **S1**.

S4: *Setze*

$$
\begin{aligned}
x_0 &= x_0 + H, \\
\boldsymbol{y}_0 &= \boldsymbol{\eta}\left(x_0 + h; \frac{H}{2}\right), \\
H &= 2h
\end{aligned}
$$

und gehe zu Schritt **S1**.

Runge-Kutta-Fehlberg-Verfahren

Es seien zwei ESV mit den Konsistenzordnungen p und $p+1$ zum Lösen des AWP

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad \boldsymbol{y}(x_0) = \boldsymbol{y}_0$$

gegeben.

S0: *Wähle Grundschrittweite H und eine zu erreichende Genauigkeit ε.*

S1: *Berechne mit einem ESV die Näherungen*

$$\boldsymbol{\eta}_1(x_0 + H; H), \quad \boldsymbol{\eta}_2(x_0 + H; H).$$

S2: *Berechne*

$$\frac{H}{h} = \sqrt[p+1]{\frac{\|\boldsymbol{\eta}_1(x_0 + H; H) - \boldsymbol{\eta}_2(x_0 + H; H)\|}{\varepsilon}}.$$

S3: *Ist $H/h \gg 2$, so setze $H = 2h$ und gehe zu Schritt* **S1**.

S4: *Setze*

$$\begin{aligned} x_0 &= x_0 + H, \\ \boldsymbol{y}_0 &= \boldsymbol{\eta}_2(x_0 + H; H), \\ H &= 2h \end{aligned}$$

und gehe zu Schritt **S1**.

$$
\begin{array}{r|rrrr}
\frac{1}{4} & \frac{1}{4} & & & \\
\frac{27}{40} & -\frac{189}{800} & \frac{729}{800} & & \\
1 & \frac{214}{891} & \frac{27}{891} & \frac{650}{891} & \\
\hline
p=2 & \frac{214}{891} & \frac{27}{891} & \frac{650}{891} & \\
p=3 & \frac{533}{2106} & 0 & \frac{1600}{2106} & -\frac{27}{2106}
\end{array}
$$

Schrittweitensteuerung für Quadraturverfahren

Es sei durch

$$Q_n(f) = (b-a)\sum_{i=0}^{n} w_i f(a+\vartheta_i(b-a))$$

ein Quadraturverfahren mit dem Exaktheitsgrad q *zum näherungsweisen Berechnen von*

$$I(f) = \int_a^b f(x)\,dx$$

gegeben.

S0: *Wähle Grundschrittweite* $H \leqq b-a$ *und eine zu erreichende Genauigkeit* ε.
Setze $x_0 = a$, $\eta_0 = 0$ *und* $k = 0$.

S1: *Falls* $x_k = b$ *so setze* $I(f) = \eta_k$. *STOPP*

S2: *Berechne die Näherungen*

$$\bar{\eta}_{k+1} = \eta_k + H \sum_{i=0}^{n} w_i f(a + \vartheta_i H)$$

und

$$\hat{\eta}_{k+1} = \eta_k + \frac{H}{2} \sum_{i=0}^{n} w_i f\left(a + \vartheta_i \frac{H}{2}\right) + \frac{H}{2} \sum_{i=0}^{n} w_i f\left(a + \frac{H}{2} + \vartheta_i \frac{H}{2}\right).$$

S3: *Berechne*

$$\frac{H}{h} = \sqrt[q+2]{\frac{2^{q+1}}{2^{q+1}-1} \frac{\|\bar{\eta}_{k+1} - \hat{\eta}_{k+1}\|}{\varepsilon}}.$$

S4: *Ist* $H/h \gg 2$, *so setze* $H = 2h$ *und gehe zu Schritt* **S2**.

S5: *Setze*

$$\begin{aligned} x_{k+1} &= x_k + H, \\ \eta_{k+1} &= \hat{\eta}_{k+1}, \\ H &= \min\{2h, b - x_{k+1}\}, \\ k &= k+1 \end{aligned}$$

und gehe zu Schritt **S1**.

ADAMS-BASHFORTH-Verfahren

q	$s\beta_{qi}$					s	Ordnung
0	1					1	1
1	3	-1				2	2
2	23	-16	5			12	3
3	55	-59	37	-9		24	4
4	1901	-2774	2616	-1274	251	720	5

NYSTRÖM-Verfahren

q	$s\beta_{qi}$					s	Ordnung
0	2					1	2
1	2	0				1	2
2	7	-2	1			3	3
3	8	-5	4	-1		3	4
4	269	-266	294	-146	29	90	5

MILNE-Verfahren

q	$s\beta_{qi}$					s	Ordnung
0	2					1	1
1	0	2				1	2
2	1	4	1			3	4
3	1	4	1	0		3	4
4	29	124	24	4	-1	90	5

Fixpunktiteration für Korrektor-Verfahren

S0: *Wähle Startwert* $\boldsymbol{\eta}_{p+1}^{(0)}$ *und setze* $\nu = 0$. *Berechne für das*

- ADAMS-MOULTON-*Verfahren*

$$\varrho = \boldsymbol{\eta}_p + h\left[\beta_{q1}\boldsymbol{f}_p + \beta_{q2}\boldsymbol{f}_{p-1} + \cdots + \beta_{qq}\boldsymbol{f}_{p-q}\right]$$

- MILNE-*Verfahren*

$$\varrho = \boldsymbol{\eta}_{p-1} + h\left[\beta_{q1}\boldsymbol{f}_p + \beta_{q2}\boldsymbol{f}_{p-1} + \cdots + \beta_{qq}\boldsymbol{f}_{p-q}\right]$$

S1: *Berechne*

$$\boldsymbol{\eta}_{p+1}^{(\nu+1)} = \varrho + h\beta_{q0}\boldsymbol{f}(x_{p+1}, \boldsymbol{\eta}_{p+1}^{(\nu)}).$$

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

Gragg sche Näherung

Es ist für die Stelle x eine Näherungslösung des AWP

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad \boldsymbol{y}(x_0) = \boldsymbol{y}_0$$

zu berechnen.

S0: *Wähle ein n, setze $h = (x - x_0)/n$ und*

$$\boldsymbol{\eta}_0 = \boldsymbol{\eta}(x_0; h) = \boldsymbol{y}_0,$$
$$\boldsymbol{\eta}_1 = \boldsymbol{\eta}(x_1; h) = \boldsymbol{y}_0 + h\boldsymbol{f}(x_0, \boldsymbol{y}_0).$$

S1: *Für $k = 1, 2, \ldots, n-1$ berechne*

$$\boldsymbol{\eta}_{k+1} = \boldsymbol{\eta}_{k-1} + 2h\boldsymbol{f}(x_k, \boldsymbol{\eta}_k),\ x_{k+1} = x_k + h.$$

S2: *Berechne*

$$\tilde{\boldsymbol{\eta}}(x; h) = \frac{1}{2}\left[\boldsymbol{\eta}_n + \boldsymbol{\eta}_{n-1} + h\boldsymbol{f}(x_n, \boldsymbol{\eta}_n)\right].$$

Extrapolationsverfahren für AWP

S0: *Wähle Grundschrittweite H und eine zu erreichende Genauigkeit ε.*
Setze $x = x_0 + H$, $h_0 = H/2$ und $k = 0$.

S1: *Berechne mit der Schrittweite h_k eine* GRAGG*'sche Näherung*

$$\boldsymbol{T}_{k0} = \tilde{\boldsymbol{\eta}}(x; h_k).$$

S2: *Für $l = 1, \ldots, k$ berechne*

$$\boldsymbol{T}_{k,l} = \boldsymbol{T}_{k,l-1} + \frac{\boldsymbol{T}_{k,l-1} - \boldsymbol{T}_{k-1,l-1}}{\left(\frac{h_{k-l}}{h_k}\right)^2 - 1}.$$

S3: *Gilt*

$$\|\boldsymbol{T}_{k,k} - \boldsymbol{T}_{k,k-1}\| \leqq \varepsilon,$$

so setze $\boldsymbol{y}_0 = \boldsymbol{\eta}(x) = \boldsymbol{T}_{k,k}$, $x_0 = x$ und gehe zu Schritt **S0**.

S4: *Setze $h_{k+1} = h_k/2$ und $k = k+1$. Gehe zu Schritt* **S1**.

Einfaches Schießverfahren für RWP

Gegeben sei das RWP

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad \boldsymbol{r}(\boldsymbol{y}(a), \boldsymbol{y}(b)) = \boldsymbol{o}.$$

S0: *Bestimme den Parameter* $\boldsymbol{s}$ *so, dass die Lösung* $\boldsymbol{y}(x; \boldsymbol{s})$ *des AWP*

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad \boldsymbol{y}(a) = \boldsymbol{s}$$

die Randbedingungen

$$\boldsymbol{r}(\boldsymbol{y}(a; \boldsymbol{s}), \boldsymbol{y}(b; \boldsymbol{s})) = \boldsymbol{r}(\boldsymbol{s}, \boldsymbol{y}(b; \boldsymbol{s})) = \boldsymbol{o}.$$

erfüllt.

Bisektionsverfahren für eindimensionale RWP

Gegeben sei das RWP

$$y' = f(x, y), \quad r(y(a), y(b)) = 0.$$

Für einen gewissen Parameter s *sei* $y(x; s)$ *die Lösung des AWP*

$$y' = f(x, y), \quad y(a) = s.$$

Weiterhin sei eine Funktion F *gemäß*

$$F(s) = r(s, y(b; s))$$

definiert. Gegeben seien Werte s_0 *und* t_0 *mit*

$$F(s_0)F(t_0) < 0.$$

S0: *Setze* $k = 0$.

S1: *Setze* $c = (s_k + t_k)/2$ *und berechne die Lösung des AWP*

$$y' = f(x, y), \quad y(a) = c$$

an der Stelle $x = b$.

S2: *Berechne* $d = F(c)$.

S3: *Für*

$$d \cdot F(s_k) \begin{cases} > & 0 \quad s_{k+1} = c, \quad t_{k+1} = t_k, \\ = & 0 \quad s^* = c, \quad STOPP, \\ < & 0 \quad s_{k+1} = s_k, \quad t_{k+1} = c. \end{cases}$$

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

Newton-Verfahren für eindimensionale RWP

S0: *Wähle* s_0 *und setze* $k=0$.

S1: *Berechne simultan die Lösungen* $y(b;s_k)$ *und* $v(b;s_k)$ *der AWP*

$$y' = f(x,y), \quad y(a) = s_k$$

und $v' = f_y(x,y)v, \quad v(a) = 1$ *an der Stelle* $x=b$.

S2: *Berechne* $F(s_k) = r(s_k, y(b;s_k))$.

S3: *Berechne*

$$F'(s_k) = r_u(s_k, y(b;s_k)) + r_v(s_k, y(b;s_k))v(b;s_k).$$

S4: *Berechne* $s_{k+1} = s_k - \frac{F(s_k)}{F'(s_k)}$.
Setze $k=k+1$ *und gehe zu Schritt* **S1**.

Quasi-Newton-Verfahren für eindimensionale RWP

S0: *Wähle* s_0 *und setze* $k = 0$.

S1: *Berechne die Lösung* $y(b; s_k)$ *des AWP*

$$y' = f(x, y), \quad y(a) = s_k$$

an der Stelle $x = b$.

S2: *Wähle ein* Δs_k *und berechne die Lösung* $y(b; s_k + \Delta s_k)$ *des AWP*

$$y' = f(x, y), \quad y(a) = s_k + \Delta s_k$$

an der Stelle $x = b$.

S3: *Berechne* $F(s_k) = r(s_k, y(b; s_k))$ *und*

$$F(s_k + \Delta s_k) = r(s_k + \Delta s_k, y(b; s_k + \Delta s_k)).$$

S4: *Berechne*

$$D_{\Delta s_k} F(s_k) = \frac{F(s_k + \Delta s_k) - F(s_k)}{\Delta s_k}.$$

S5: *Berechne*

$$s_{k+1} = s_k - \frac{F(s_k)}{D_{\Delta s_k} F(s_k)}$$

S6: *Setze* $k = k + 1$ *und gehe zu Schritt* **S1**.

Quasi-Newton-Verfahren für allgemeine RWP

S0: *Wähle* $\boldsymbol{s}^{(0)}$ *und setze* $k = 0$.

S1: *Berechne die Lösung* $\boldsymbol{y}(b; \boldsymbol{s}^{(k)})$ *des AWP*

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad \boldsymbol{y}(a) = \boldsymbol{s}^{(k)}$$

an der Stelle $x = b$.

S2: *Wähle ein*

$$\Delta \boldsymbol{s}^{(k)} = (\Delta s_1^{(k)}, \ldots, \Delta s_n^{(k)})^T$$

und berechne für $i = 1, \ldots, n$ *die Lösungen*

$$\boldsymbol{y}^{(i)} = \boldsymbol{y}(b; \boldsymbol{s}^{(k)} + \Delta s_i^{(k)} \boldsymbol{e}_i)$$

der AWP

$$\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y}), \quad \boldsymbol{y}(a) = \boldsymbol{s}^{(k)} + \Delta s_i^{(k)} \boldsymbol{e}_i$$

an der Stelle $x = b$.

S3: *Berechne*

$$\boldsymbol{F}\left(\boldsymbol{s}^{(k)}\right) = \boldsymbol{r}\left(\boldsymbol{s}^{(k)}, \boldsymbol{y}\left(b; \boldsymbol{s}^{(k)}\right)\right)$$

und für $i = 1, \ldots, n$

$$\boldsymbol{F}\left(\boldsymbol{s}^{(k)} + \Delta s_i^{(k)} \boldsymbol{e}_i\right) = \boldsymbol{r}\left(\boldsymbol{s}^{(k)} + \Delta s_i^{(k)} \boldsymbol{e}_i, \boldsymbol{y}^{(i)}\right).$$

S4: *Berechne für* $i = 1, \ldots, n$

$$\boldsymbol{D}_{\Delta s_i^{(k)}} \boldsymbol{F}(\boldsymbol{s}^{(k)}) = \frac{\boldsymbol{F}(\boldsymbol{s}^{(k)} + \Delta s_i^{(k)} \boldsymbol{e}_i) - \boldsymbol{F}(\boldsymbol{s}^{(k)})}{\Delta s_i^{(k)}}.$$

S5: *Setze*

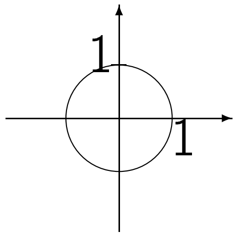
$$\boldsymbol{D}_{\Delta \boldsymbol{s}^{(k)}} \boldsymbol{F}(\boldsymbol{s}^{(k)}) = \left(\boldsymbol{D}_{\Delta s_1^{(k)}} \boldsymbol{F}(\boldsymbol{s}^{(k)}), \ldots, \boldsymbol{D}_{\Delta s_n^{(k)}} \boldsymbol{F}(\boldsymbol{s}^{(k)})\right).$$

S6: *Löse das lineare Gleichungssystem*

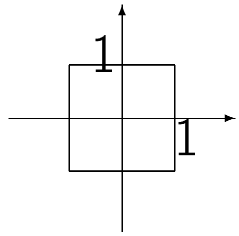
$$\boldsymbol{D}_{\Delta \boldsymbol{s}^{(k)}} \boldsymbol{F}(\boldsymbol{s}^{(k)}) \boldsymbol{d}^{(k)} = \boldsymbol{F}(s^{(k)}).$$

S7: *Setze* $\boldsymbol{s}^{(k+1)} = \boldsymbol{s}^{(k)} - \boldsymbol{d}^{(k)}$.

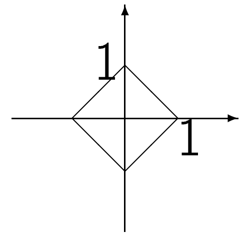
S8: *Setze* $k = k + 1$ *und gehe zu Schritt* **S1**.



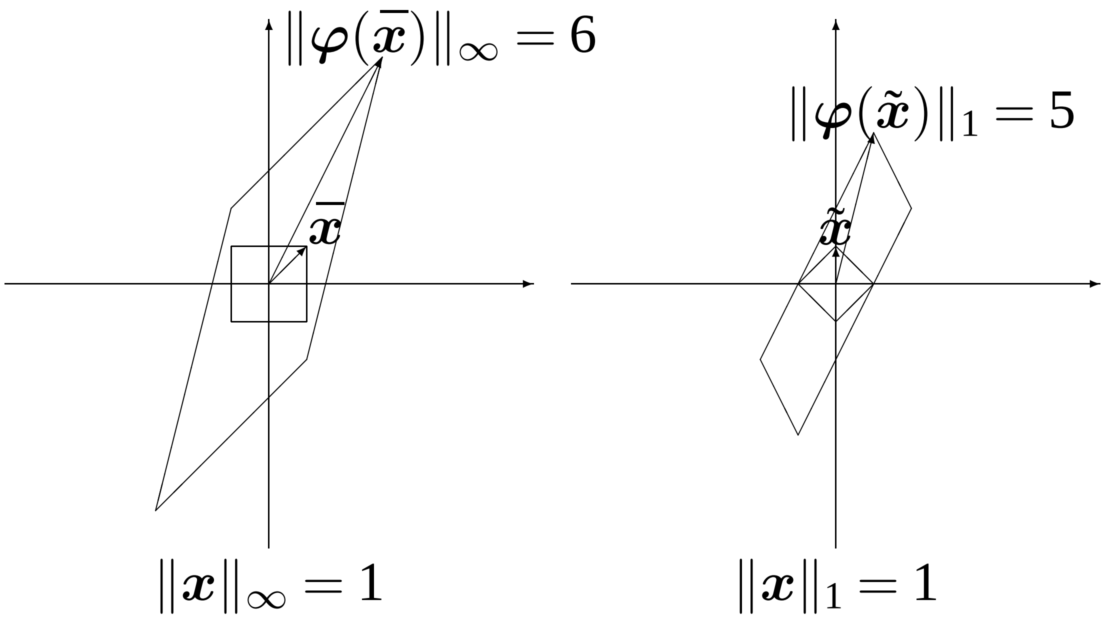
$\|\boldsymbol{x}\|_2 = 1$



$\|\boldsymbol{x}\|_\infty = 1$



$\|\boldsymbol{x}\|_1 = 1$



$\|\boldsymbol{\varphi}(\bar{\boldsymbol{x}})\|_\infty = 6$
$\bar{\boldsymbol{x}}$
$\|\boldsymbol{x}\|_\infty = 1$
$\|\boldsymbol{\varphi}(\tilde{\boldsymbol{x}})\|_1 = 5$
$\tilde{\boldsymbol{x}}$
$\|\boldsymbol{x}\|_1 = 1$

$\boldsymbol{b}$	$\boldsymbol{\delta b}$	$\frac{\lVert\boldsymbol{\delta b}\rVert}{\lVert\boldsymbol{b}\rVert}$	$\boldsymbol{x}$	$\boldsymbol{\delta x}$	$\frac{\lVert\boldsymbol{\delta x}\rVert}{\lVert\boldsymbol{x}\rVert}$	$\frac{\lVert\boldsymbol{\delta x}\rVert\,\lVert\boldsymbol{b}\rVert}{\lVert\boldsymbol{x}\rVert\,\lVert\boldsymbol{\delta b}\rVert}$
$\begin{pmatrix} 1.00 \\ 0.99 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 10^{-5} \end{pmatrix}$	$\approx 5 \cdot 10^{-6}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0.099 \\ 0.100 \end{pmatrix}$	≈ 0.2	39601
$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 10^{-5} \\ 0 \end{pmatrix}$	$\approx 5 \cdot 10^{-6}$	$\begin{pmatrix} 19700 \\ -19900 \end{pmatrix}$	$\begin{pmatrix} -0.0998 \\ 0.0990 \end{pmatrix}$	$\approx 5 \cdot 10^{-6}$	0.995

$\boldsymbol{b}$	$\boldsymbol{x}$	$\boldsymbol{\delta x}$	$\frac{\lVert\boldsymbol{\delta x}\rVert}{\lVert\boldsymbol{x}\rVert}$	$\frac{\lVert\boldsymbol{\delta x}\rVert\,\lVert\boldsymbol{b}\rVert}{\lVert\boldsymbol{x}\rVert\,\lVert\boldsymbol{\delta b}\rVert}$
$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 100 \\ -100 \end{pmatrix}$	$\begin{pmatrix} 24.75 \\ -25.00 \end{pmatrix}$	≈ 0.24875	≈ 25000
$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 19700 \\ -19900 \end{pmatrix}$	$\begin{pmatrix} 4925.25 \\ -4975.00 \end{pmatrix}$	≈ 0.2500	≈ 25000

$\boldsymbol{b}$	$\boldsymbol{x}$	$\boldsymbol{\delta x}$	$\frac{\lVert\boldsymbol{\delta x}\rVert}{\lVert\boldsymbol{x}\rVert}$	$\frac{\lVert\boldsymbol{\delta x}\rVert\,\lVert\boldsymbol{b}\rVert}{\lVert\boldsymbol{x}\rVert\,\lVert\boldsymbol{\delta b}\rVert}$
$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 100 \\ -100 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	0	0
$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 19700 \\ -19900 \end{pmatrix}$	$\begin{pmatrix} -0.2 \\ 0.2 \end{pmatrix}$	$\approx 10^{-5}$	≈ 0.5

LU-Zerlegung mit explizitem Zeilentausch

Es ist die reguläre (n,n)-Matrix $\boldsymbol{A}$ in ein Produkt $\boldsymbol{PA} = \boldsymbol{LU}$ mit einer Permutationsmatrix $\boldsymbol{P}$, einer unteren Einsdreiecksmatrix $\boldsymbol{L}$ und einer oberen Dreiecksmatrix $\boldsymbol{U}$ zu zerlegen. Die Permutationsmatrix $\boldsymbol{P}$ sei dabei auf einem Feld p der Länge n gespeichert.

{Initialisierung}
for $i = 1$ ***to*** n **do**
  $p(i) = i$
endfor
{LU-Zerlegung}
for $j = 1$ ***to*** $n-1$ ***do***
  {Pivotsuche}
  Wähle einen Index

$$i^* \in \{\, i \mid a_{ij} \neq 0, \quad i = j, \ldots, n \,\}.$$

  {Zeilentausch}
  if $i^* \neq j$ **then**
    $pp = p(i^*);\ p(i^*) = p(j);\ p(j) = pp$

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

$aa = a_{i^*k};\ a_{i^*k} = a_{jk};\ a_{jk} = aa$

endfor

endif

{Transformation der Restmatrix}

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

$a_{ij} = a_{ij}/a_{jj}$

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

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

endfor

endfor

endfor

Aufwand: $\sim n^3/3$ *Additionen/Multiplikationen*

Vorwärtssubstitution bei explizitem Zeilentausch

Es ist das lineare Gleichungssystem $\boldsymbol{Ly} = \boldsymbol{Pb}$ *mit der regulären unteren Einsdreiecksmatrix* $\boldsymbol{L}$, *der Permutationsmatrix* $\boldsymbol{P}$ *und einem beliebigen Vektor* $\boldsymbol{b} \in \mathbb{R}^n$ *zu lösen. Die Matrix* $\boldsymbol{L}$ *ist auf dem unteren Dreieck der Matrix* $\boldsymbol{A}$, *die Permutationsmatrix ist auf einem Feld* p *der Länge* n *gespeichert.*

for $j = 1$ ***to*** n ***do***
  $y_j = b_{p(j)}$
endfor
for $j = 1$ ***to*** n ***do***
  for $i = 1$ ***to*** $j - 1$ ***do***
    $y_i = y_i - a_{ij} \cdot y_j$
  endfor
endfor

Aufwand: $\sim n^2/2$ *Additionen/Multiplikationen*

Rücksubstitution bei explizitem Zeilentausch

Es ist das lineare Gleichungssystem $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$ mit der regulären oberen Dreiecksmatrix $\boldsymbol{U}$ und einem beliebigen Vektor $\boldsymbol{y} \in \mathbb{R}^n$ zu lösen. Die Matrix $\boldsymbol{U}$ ist auf dem oberen Dreieck der Matrix $\boldsymbol{A}$ gespeichert.

for $j = n$ **to** 1 **step** -1 **do**
  $x_j = y_j / a_{jj}$
  for $i = 1$ **to** $j - 1$ **do**
    $y_i = y_i - a_{ij} \cdot x_j$
  endfor
endfor

Aufwand: $\sim n^2/2$ *Additionen/Multiplikationen*

LU-Zerlegung mit fiktivem Zeilentausch

Es ist die reguläre Matrix $\boldsymbol{A}$ in ein Produkt $\boldsymbol{PA} = \boldsymbol{LU}$ mit einer Permutationsmatrix $\boldsymbol{P}$, einer unteren Einsdreiecksmatrix $\boldsymbol{L}$ und einer oberen Dreiecksmatrix $\boldsymbol{U}$ zu zerlegen. Die Permutationsmatrix $\boldsymbol{P}$ sei dabei auf einem Feld p der Länge n gespeichert.

{*Initialisierung*}
for $j = 1$ ***to*** n ***do***
  $p(i) = i$
endfor
{*LU-Zerlegung*}
for $j = 1$ ***to*** $n - 1$ ***do***
  {*Pivotsuche*}
  Wähle einen Index

$$i^* \in \left\{ i \;\middle|\; a_{p(i),j} \neq 0, \quad i = j, \ldots, n \right\}.$$

  {*Fiktiver Zeilentausch*}
  if $i^* \neq j$ ***then***
    $pp = p(i^*);\ p(i^*) = p(j);\ p(j) = pp$

```
    endif
    {Transformation der Restmatrix}
    for i = j+1 to n do
      a_{p(i),j} = a_{p(i),j} / a_{p(j),j}
      for k = j+1 to n do
        a_{p(i),k} = a_{p(i),k} - a_{p(i),j} · a_{p(j),k}
      endfor
    endfor
  endfor
```

Aufwand: $\sim n^3/3$ *Additionen/Multiplikationen*

Vorwärtssubstitution bei fiktivem Zeilentausch

Es ist das lineare Gleichungssystem $\boldsymbol{Ly} = \boldsymbol{Pb}$ *mit der regulären unteren Einsdreiecksmatrix* $\boldsymbol{L}$, *der Permutationsmatrix* $\boldsymbol{P}$ *und einem beliebigen Vektor* $\boldsymbol{b} \in \mathbb{R}^n$ *zu lösen. Die Matrix* $\boldsymbol{L}$ *ist auf dem unteren Dreieck der Matrix* $\boldsymbol{PA}$, *die Permutationsmatrix ist auf einem Feld* p *der Länge* n *gespeichert.*

for $i = 1$ ***to*** n ***do***
  $y_i = b_{p(i)}$
  for $j = 1$ ***to*** $i - 1$ ***do***
    $y_i = y_i - a_{p(i)j} \cdot y_j$
  endfor
endfor

Aufwand: $\sim n^2/2$ *Additionen/Multiplikationen*

Rücksubstitution bei fiktivem Zeilentausch

Es ist das lineare Gleichungssystem $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$ mit der regulären oberen Dreiecksmatrix $\boldsymbol{U}$ und einem beliebigen Vektor $\boldsymbol{y} \in \mathbb{R}^n$ zu lösen. Die Matrix $\boldsymbol{U}$ ist auf dem oberen Dreieck der Matrix $\boldsymbol{A}$ gespeichert.

for $j = n$ ***to*** 1 ***step*** -1 ***do***
  $x_j = y_j / a_{p(j)j}$
  for $i = 1$ ***to*** $j - 1$ ***do***
    $y_i = y_i - a_{p(i),j} \cdot x_j$
  endfor
endfor

Aufwand: $\sim n^2/2$ *Additionen/Multiplikationen*

LU-Zerlegung mit Spaltenpivotisierung, fiktivem Zeilentausch und fiktiver Skalierung

Es ist die reguläre Matrix $\boldsymbol{A}$ in ein Produkt $\boldsymbol{PA} = \boldsymbol{LU}$ mit einer Permutationsmatrix $\boldsymbol{P}$, einer unteren Einsdreiecksmatrix $\boldsymbol{L}$ und einer oberen Dreiecksmatrix $\boldsymbol{U}$ zu zerlegen. Die Permutationsmatrix $\boldsymbol{P}$ sei dabei auf einem Feld p der Länge n gespeichert.

```
{Initialisierung}
Wähle eine Genauigkeitsschranke ε > 0.
for i = 1 to n do
   p(i) = i; d_i = 0
   for j = 1 to n do
      d_i = d_i + |a_ij|
   endfor
endfor
{LU-Zerlegung}
for j = 1 to n−1 do
   {Pivotsuche}
   piv = 0
   for k = j to n do
```

if $d_j|a_{kj}| > piv$ **then**
  $piv = d_j|a_{kj}|$
  $i^* = k$
endif
endfor
if $piv \leqq \varepsilon$ **then**
  STOPP
endif
{Fiktiver Zeilentausch}
if $i^* \neq j$ **then**
  $pp = p(i^*);\ p(i^*) = p(j);\ p(j) = pp$
endif
{Transformation der Restmatrix}
for $i = j+1$ **to** n **do**
  $a_{p(i),j} = a_{p(i),j}/a_{p(j),j}$
  for $k = j+1$ **to** n **do**
    $a_{p(i),k} = a_{p(i),k} - a_{p(i),j} \cdot a_{p(j),k}$
  endfor
endfor
endfor

Aufwand: $\sim n^3/3$ *Additionen/Multiplikationen*

LDL^T-Zerlegung

Eine regläre, symmetrische Matrix $\boldsymbol{A}$ *wird in ein Produkt* $\boldsymbol{A} = \boldsymbol{LDL}^T$ *mit einer unteren Einsdreiecksmatrix* $\boldsymbol{L}$ *und einer Diagonalmatrix* $\boldsymbol{D}$ *zerlegt. Von der Matrix* $\boldsymbol{A}$ *ist nur das obere Dreieck einschließlich der Diagonalen gespeichert.*

```
{Initialisierung}
Wähle Genauigkeitsschranke ε > 0.
{(LDL^T-Zerlegung)}
for k = 1 to n−1 do
   if |a_kk| ≦ ε then
      STOPP
   endif
   {Transformation der Restmatrix}
   for i = k+1 to n do
      l = a_ki/a_kk
      for j = i to n do
         a_ij = a_ij − l · a_kj
      endfor
      a_ki = l
   endfor
```

endfor

Aufwand: $\sim n^3/6$ *Additionen/Multiplikationen*

Lösen eines linearen Gleichungssystems bei bekannter LDL^T-Zerlegung

Es ist das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ *mit der regulären symmetrischen Matrix* $\boldsymbol{A}$ *zu lösen. Von der Matrix* $\boldsymbol{A}$ *sei eine Zerlegung der Form* $\boldsymbol{A} = \boldsymbol{LDL}^T$ *mit einer unteren Einsdreiecksmatrix* $\boldsymbol{L}$ *und einer Diagonalmatrix* $\boldsymbol{D}$ *bekannt. Dabei ist die Matrix* $\boldsymbol{L}^T$ *auf dem oberen Dreieck der Matrix* $\boldsymbol{A}$ *und die Diagonalmatrix* $\boldsymbol{D}$ *auf der Diagonale der Matrix* $\boldsymbol{A}$ *gespeichert.*

$\{\boldsymbol{Ly} = \boldsymbol{b}\}$
$\{\boldsymbol{b}$ *wird mit* $\boldsymbol{y}$ *überschrieben.*$\}$
for $k = 1$ ***to*** $n-1$ ***do***
  for $i = k+1$ ***to*** n ***do***
    $b_i = b_i - a_{ki} \cdot b_k$
  endfor
endfor
$\{\boldsymbol{Dz} = \boldsymbol{y}\}$
$\{\boldsymbol{y}$ *steht auf dem Speicherplatz von* $\boldsymbol{b}$ *und wird mit* $\boldsymbol{z}$ *überschrieben.*$\}$
for $i = 1$ ***to*** n ***do***

$b_i = b_i / a_{ii}$

endfor

$\{\boldsymbol{L}^T \boldsymbol{x} = \boldsymbol{z}\}$

{$\boldsymbol{z}$ steht auf dem Speicherplatz von $\boldsymbol{b}$ und wird mit $\boldsymbol{x}$ überschrieben.}

for $k = n$ ***to*** 2 ***step*** -1 ***do***

for $i = 1$ ***to*** $k-1$ ***do***

$b_i = b_i - a_{ik} \cdot b_k$

endfor

endfor

Aufwand: $\sim n^2$ *Additionen/Multiplikationen*

CHOLESKY-Zerlegung

Es ist die symmetrische, positiv definite Matrix $\boldsymbol{A}$ in ein Produkt $\boldsymbol{A} = \hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T$ mit einer unteren Dreiecksmatrix $\hat{\boldsymbol{L}}$ zu zerlegen. Von der Matrix $\boldsymbol{A}$ sind nur das untere Dreieck und die Diagonale gespeichert.

```
{Initialisierung}
Wähle Genauigkeitsschranke ε > 0.
{Zerlegung}
for k = 1 to n do
   if a_kk ≦ ε then
      STOPP
   endif
   a_kk = √(a_kk)
   {Transformation der Restmatrix}
   for i = k+1 to n do
      a_ik = a_ik / a_kk
   endfor
   for i = k+1 to n do
      for j = i to n do
         a_ij = a_ij − l · a_jk
      endfor
```

endfor
endfor

Aufwand: $\sim n^3/6$ *Additionen/Multiplikationen* $+\ n$ *Wurzeln*

HOUSEHOLDER-Matrix zur Transformation eines Vektors auf ein Vielfaches des ersten Einheitsvektors

Es sei ein Vektor $\boldsymbol{a} \in \mathbb{R}^k$ *gegeben.*

S0: *Berechne* $\varrho = \sqrt{\boldsymbol{a}^T \boldsymbol{a}} = \|\boldsymbol{a}\|_2$.

S1: if $\varrho = 0$ **then** $\boldsymbol{H} = \boldsymbol{I}$ **STOPP.**

S2: if $a_1 > 0$ **then** $\varrho = -\varrho$.
Berechne $\boldsymbol{v} = \boldsymbol{a} - \varrho \boldsymbol{e}_1$ *und* $\gamma = \varrho(\varrho - a_1)$.
Die Matrix $\boldsymbol{H} = \boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^T/\gamma$ *ist dann orthogonal und es gilt*

$$\boldsymbol{H}\boldsymbol{a} = \varrho \boldsymbol{e}_1.$$

Aufwand: $\sim n$ *Additionen/Multiplikationen, eine Quadratwurzel.*

Householder-Transformation

Gegeben sei eine Householder-*Matrix*

$$\boldsymbol{H} = \boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^T/\gamma \in \mathbb{R}^k.$$

Zu berechnen ist für einen Vektor $\boldsymbol{x} \in \mathbb{R}^k$ *der Vektor*

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}.$$

S0: *Berechne* $\beta = \boldsymbol{v}^T\boldsymbol{x}/\gamma$.
S1: *Berechne* $\boldsymbol{y} = \boldsymbol{x} - \beta\boldsymbol{v}$
Aufwand: $\sim 2n$ *Additionen/Multiplikationen.*

QR-Zerlegung einer Matrix nach Householder

Es ist die reguläre (n,n)-Matrix $\boldsymbol{A}$ in ein Produkt $\boldsymbol{A} = \boldsymbol{QR}$ mit einer orthogonalen Matrix $\boldsymbol{Q}$ und einer oberen Dreiecksmatrix $\boldsymbol{R}$ zu zerlegen.

{Initialisierung}
Wähle eine Genauigkeitsschranke $\varepsilon > 0$.
{QR-Zerlegung}
for $k = 1$ *to* $n-1$ ***do***
  {Berechnen der Transformationsmatrix $\boldsymbol{H}^{(k)}$)}
  $\varrho_k = \sqrt{a_{kk}^2 + a_{k+1,k}^2 + \cdots + a_{nk}^2}$
  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* n **do**
$\quad \beta = \beta + a_{ik} \cdot a_{ij}$
endfor
$\beta = \beta / \gamma_k$
for $i = k$ *to* n **do**
$\quad a_{ij} = a_{ij} - \beta \cdot a_{ik}$
endfor
endfor
endfor
$\varrho_n = a_{nn}$

Aufwand: $\sim 2n^3/3$ *Additionen/Multiplikationen,* n *Quadratwurzeln.*

Lösen eines Gleichungssystems bei bekannter QR-Zerlegung

Es ist das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ zu lösen.
Für die Matrix $\boldsymbol{A}$ wurde mit obigem Algorithmus eine QR-Zerlegung berechnet.

{*Berechnen von $\boldsymbol{c} = \boldsymbol{Q}^T\boldsymbol{b}$*}
for $k = 1$ ***to*** $n-1$ ***do***
  $\beta = 0$
  for $i = k$ ***to*** n ***do***
    $\beta = \beta + b_i \cdot a_{ik}$
  endfor
  $\beta = \beta/\gamma_k$
  for $i = k$ ***to*** n ***do***
    $b_i = b_i - \beta \cdot a_{ik}$
  endfor
endfor
{*Lösen von $\boldsymbol{Rx} = \boldsymbol{c}$*}
for $k = n$ ***to*** 1 ***step*** -1 ***do***
  for $i = k+1$ ***to*** n ***do***
    $b_k = b_k - b_i \cdot a_{ki}$
  endfor

$b_k = b_k/\varrho_k$

endfor

Aufwand: $\sim n^2$ *Additionen/Multiplikationen.*

JACOBI-Verfahren

Es ist das Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *zu lösen. Für die Matrix* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *gelte*

$$a_{ii} \neq 0, \quad i = 1, \ldots, n.$$

S0: *Wähle einen Startvektor* $\boldsymbol{x}^{(0)} \in \mathbb{R}^{n\times n}$ *und setze* $i = 0$.

S1: *Berechne*

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

$$x_k^{(i+1)} = \frac{1}{a_{kk}} \left(b_k - \sum_{\substack{j=1 \\ j\neq k}}^{n} a_{kj} x_j^{(i)} \right)$$

end

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

GAUSS-SEIDEL-Verfahren

Es ist das Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ *zu lösen. Für die Matrix* $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ *gelte*

$$a_{ii} \neq 0, \quad i = 1, \ldots, n.$$

S0: *Wähle einen Startvektor* $\boldsymbol{x}^{(0)} \in \mathbb{R}^{n \times n}$ *und setze* $i = 0$.

S1: *Berechne*

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

$$x_k^{(i+1)} = \frac{1}{a_{kk}} \left(b_k - \sum_{j=1}^{k-1} a_{kj} x_j^{(i+1)} - \sum_{j=k+1}^{n} a_{kj} x_j^{(i)} \right)$$

end

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

Nachiteration bei LU-Zerlegung

Zu lösen sei das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ *mit* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *und* $\boldsymbol{b} \in \mathbb{R}^n$. *Durch* $\boldsymbol{\mathcal{L}}$ *und* $\boldsymbol{\mathcal{U}}$ *seien die berechneten Dreiecksfaktoren einer* LU*-Zerlegung der Matrix* $\boldsymbol{A}$ *gegeben. Es gelte*

$$\boldsymbol{A} + \boldsymbol{\delta A} = \boldsymbol{P}^T \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{U}}.$$

S0: *Wähle einen Startvektor* $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ *und setze* $i = 0$.

S1: *Berechne das Residuum*

$$\boldsymbol{r}^{(i)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(i)}.$$

S2: *Löse das Gleichungssystem* $\boldsymbol{A}\boldsymbol{\delta x}^{(i)} = \boldsymbol{r}^{(i)}$ *näherungsweise gemäß*

$$\boldsymbol{\mathcal{L}}\boldsymbol{y} = \boldsymbol{P}\boldsymbol{r}^{(i)} \quad \textit{und} \quad \boldsymbol{\mathcal{U}}\boldsymbol{\delta x}^{(i)} = \boldsymbol{y}.$$

S3: *Setze* $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \boldsymbol{\delta x}^{(i)}$, $i = i+1$ *und gehe zu Schritt* **S1**.

Nachiteration bei LDL^T-Zerlegung

Zu lösen ist das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ *mit der symmetrischen Matrix* $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ *und* $\boldsymbol{b} \in \mathbb{R}^n$. *Durch* $\boldsymbol{\mathcal{L}}$ *und* $\boldsymbol{\mathcal{D}}$ *seien die berechneten Faktoren einer* LDL^T*-Zerlegung der Matrix* $\boldsymbol{A}$ *gegeben. Es gelte*

$$\boldsymbol{A} + \boldsymbol{\delta A} = \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{L}}^T.$$

S0: *Wähle einen Startvektor* $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ *und setze* $i = 0$. *Berechne das Residuum*

$$\boldsymbol{r}^{(i)} = \boldsymbol{b} - \boldsymbol{Ax}^{(i)}.$$

S1: *Löse das Gleichungssystem* $\boldsymbol{A\delta x}^{(i)} = \boldsymbol{r}^{(i)}$ *näherungsweise gemäß*

$$\boldsymbol{\mathcal{L}}\boldsymbol{y} = \boldsymbol{r}^{(i)} \quad \textit{und} \quad \boldsymbol{\mathcal{L}}^T\boldsymbol{\delta x}^{(i)} = \boldsymbol{\mathcal{D}}^{-1}\boldsymbol{y}.$$

S2: *Setze* $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \boldsymbol{\delta x}^{(i)}$, $i = i + 1$ *und gehe zu Schritt* **S1**.

Nachiteration bei QR-Zerlegung

Zu lösen ist das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ mit $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ und $\boldsymbol{b} \in \mathbb{R}^n$. Durch $\mathcal{Q}$ und $\mathcal{R}$ seien die berechneten Faktoren einer QR-Zerlegung der Matrix $\boldsymbol{A}$ gegeben. Es gelte

$$\boldsymbol{A} + \boldsymbol{\delta A} = \mathcal{Q}\mathcal{R}.$$

S0: *Wähle einen Startvektor $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ und setze $i = 0$.*
S1: *Berechne das Residuum*

$$\boldsymbol{r}^{(i)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(i)}.$$

S2: *Löse das Gleichungssystem $\boldsymbol{A}\boldsymbol{\delta x}^{(i)} = \boldsymbol{r}^{(i)}$ näherungsweise gemäß*

$$\boldsymbol{y} = \boldsymbol{Q}^T\boldsymbol{r}^{(i)} \quad \textit{und} \quad \mathcal{R}\boldsymbol{\delta x}^{(i)} = \boldsymbol{y}.$$

S3: *Setze $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \boldsymbol{\delta x}^{(i)}$, $i = i + 1$ und gehe zu Schritt* **S1**.

cg-Verfahren von HESTENES und STIEFEL

Zu lösen ist das lineare Gleichungssystem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *mit einer symmetrischen, positiv definiten Matrix* $\boldsymbol{A} \in \mathbb{R}^{n \times n}$.

S0: *Wähle einen Startvektor* $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$.

S1: *Berechne* $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)}$.
Setze $\boldsymbol{h}^{(0)} = \boldsymbol{r}^{(0)}$ *und* $i = 0$.

S2: *Falls* $\boldsymbol{h}^{(i)} = \boldsymbol{o}$ *STOPP.* $\boldsymbol{x}^{(i)}$ *ist die gesuchte Lösung.*

S3: *Berechne*

$$
\begin{aligned}
\alpha_i &= \frac{\boldsymbol{r}^{(i)^T}\boldsymbol{r}^{(i)}}{\boldsymbol{h}^{(i)^T}\boldsymbol{A}\boldsymbol{h}^{(i)}}, \\
\boldsymbol{x}^{(i+1)} &= \boldsymbol{x}^{(i)} + \alpha_i \boldsymbol{h}^{(i)}, \\
\boldsymbol{r}^{(i+1)} &= \boldsymbol{r}^{(i)} - \alpha_i \boldsymbol{A}\boldsymbol{h}^{(i)}, \\
\beta_i &= \frac{\boldsymbol{r}^{(i+1)^T}\boldsymbol{r}^{(i+1)}}{\boldsymbol{r}^{(i)^T}\boldsymbol{r}^{(i)}}, \\
\boldsymbol{h}^{(i+1)} &= \boldsymbol{r}^{(i+1)} + \beta_i \boldsymbol{h}^{(i)}.
\end{aligned}
$$

S4: *Setze* $i = i + 1$ *und gehe zu Schritt* **S2**.

Modifiziertes cg-Verfahren

Zu lösen ist das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ *mit einer regulären Matrix* $\boldsymbol{A} \in \mathbb{R}^{n \times n}$.

S0: *Wähle einen Startvektor* $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$.

S1: *Berechne* $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{Ax}^{(0)}$ *und* $\boldsymbol{h}^{(0)} = \boldsymbol{A}^T \boldsymbol{r}^{(0)}$
Setze $i = 0$.

S2: *Falls* $\boldsymbol{h}^{(i)} = \boldsymbol{o}$ *STOPP.* $\boldsymbol{x}^{(i)}$ *ist die gesuchte Lösung.*

S3: *Berechne*

$$\begin{aligned}
\alpha_i &= \frac{{\boldsymbol{r}^{(i)}}^T \boldsymbol{r}^{(i)}}{{\boldsymbol{h}^{(i)}}^T \boldsymbol{h}^{(i)}}, \\
\boldsymbol{x}^{(i+1)} &= \boldsymbol{x}^{(i)} + \alpha_i \boldsymbol{h}^{(i)}, \\
\boldsymbol{r}^{(i+1)} &= \boldsymbol{r}^{(i)} - \alpha_i \boldsymbol{A} \boldsymbol{h}^{(i)}, \\
\beta_i &= \frac{{\boldsymbol{r}^{(i+1)}}^T \boldsymbol{r}^{(i+1)}}{{\boldsymbol{r}^{(i)}}^T \boldsymbol{r}^{(i)}}, \\
\boldsymbol{h}^{(i+1)} &= \boldsymbol{A}^T \boldsymbol{r}^{(i+1)} + \beta_i \boldsymbol{h}^{(i)}.
\end{aligned}$$

S4: *Setze* $i = i + 1$ *und gehe zu Schritt* **S2**.

Vorkonditioniertes cg-Verfahren

Zu lösen ist das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ mit einer symmetrischen, positiv definiten Matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, von der eine näherungsweise CHOLESKY*-Zerlegung $\boldsymbol{LL}^T = \boldsymbol{A} + \boldsymbol{\delta A}$ bekannt ist.*

S0: *Wähle einen Startvektor $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$.*

S1: *Berechne $\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{Ax}^{(0)}$ und $\boldsymbol{h}^{(0)} = \left(\boldsymbol{LL}^T\right)^{-1} \boldsymbol{r}^{(0)}$.*
Setze $i = 0$.

S2: *Falls $\boldsymbol{h}^{(i)} = \boldsymbol{o}$ STOPP. $\boldsymbol{x}^{(i)}$ ist die gesuchte Lösung.*

S3: *Berechne*

$$\alpha_i = \frac{\boldsymbol{r}^{(i)^T} \left(\boldsymbol{LL}^T\right)^{-1} \boldsymbol{r}^{(i)}}{\boldsymbol{h}^{(i)^T} \boldsymbol{A}\boldsymbol{h}^{(i)}},$$

$$\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \alpha_i \boldsymbol{h}^{(i)},$$
$$\boldsymbol{r}^{(i+1)} = \boldsymbol{r}^{(i)} - \alpha_i \boldsymbol{A}\boldsymbol{h}^{(i)},$$
$$\beta_i = \frac{\boldsymbol{r}^{(i+1)^T} \left(\boldsymbol{LL}^T\right)^{-1} \boldsymbol{r}^{(i+1)}}{\boldsymbol{r}^{(i)^T} \left(\boldsymbol{LL}^T\right)^{-1} \boldsymbol{r}^{(i)}},$$

$$\boldsymbol{h}^{(i+1)} = \left(\boldsymbol{L}\boldsymbol{L}^T\right)^{-1} \boldsymbol{r}^{(i+1)} + \beta_i \boldsymbol{h}^{(i)}.$$

S4: *Setze $i = i + 1$ und gehe zu Schritt* **S2**.

Vorkonditioniertes, modifiziertes cg-Verfahren

Zu lösen ist das lineare Gleichungssystem $\boldsymbol{Ax} = \boldsymbol{b}$ *mit einer regulären Matrix* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$*, von der eine näherungsweise LU-Zerlegung* $\boldsymbol{P}^T\boldsymbol{LU} = \boldsymbol{A} + \boldsymbol{\delta A}$ *bekannt ist.*

S0: *Wähle einen Startvektor* $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$*.*

S1: *Berechne*

$$\boldsymbol{r}^{(0)} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(0)},\ \boldsymbol{h}^{(0)} = \left(\boldsymbol{U}^T\boldsymbol{U}\right)^{-1}\boldsymbol{A}^T\boldsymbol{P}^T\left(\boldsymbol{L}\boldsymbol{L}^T\right)^{-1}\boldsymbol{P}\boldsymbol{r}^{(0)}.$$

Setze $i = 0$.

S2: *Falls* $\boldsymbol{h}^{(i)} = \boldsymbol{o}$ *STOPP.* $\boldsymbol{x}^{(i)}$ *ist die gesuchte Lösung.*

S3: *Berechne*

$$\alpha_i = \frac{\left(\boldsymbol{P}\boldsymbol{r}^{(i)}\right)^T\left(\boldsymbol{L}\boldsymbol{L}^T\right)^{-1}\left(\boldsymbol{P}\boldsymbol{r}^{(i)}\right)}{\boldsymbol{h}^{(i)^T}\left(\boldsymbol{U}^T\boldsymbol{U}\right)\boldsymbol{h}^{(i)}},$$

$$\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \alpha_i\boldsymbol{h}^{(i)},$$
$$\boldsymbol{r}^{(i+1)} = \boldsymbol{r}^{(i)} - \alpha_i\boldsymbol{A}\boldsymbol{h}^{(i)},$$

$$\beta_i = \frac{\left(\boldsymbol{P}\boldsymbol{r}^{(i+1)}\right)^T \left(\boldsymbol{L}\boldsymbol{L}^T\right)^{-1} \left(\boldsymbol{P}\boldsymbol{r}^{(i+1)}\right)}{\left(\boldsymbol{P}\boldsymbol{r}^{(i)}\right)^T \left(\boldsymbol{L}\boldsymbol{L}^T\right)^{-1} \left(\boldsymbol{P}\boldsymbol{r}^{(i)}\right)},$$

$$\boldsymbol{h}^{(i+1)} = \left(\boldsymbol{U}^T\boldsymbol{U}\right)^{-1} \boldsymbol{A}^T\boldsymbol{P}^T \left(\boldsymbol{L}\boldsymbol{L}^T\right)^{-1} \boldsymbol{P}\boldsymbol{r}^{(i+1)} + \beta_i \boldsymbol{h}^{(i)}.$$

S4: *Setze* $i = i + 1$ *und gehe zu Schritt* **S2**.