Kapitel 4

Iterationsverfahren für lineare Gleichungssysteme

4.1 Gesamt- und Einzelschrittverfahren

Das Standardverfahren zur Lösung linearer Gleichungssysteme ist das Eliminationsverfahren. Bei sehr großen Systemen mit spezieller Struktur können iterative Verfahren aber günstiger sein. Dies trifft insbesondere zu auf lineare Systeme, deren Matrizen nur sehr wenige von Null verschiedene Elemente haben (dünnbesetzte Matrizen).

Sei $Ax = b$ zu lösen. Sei $A = D + L + R$ mit

$$D = \begin{pmatrix} a_{1,1} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & a_{n,m} \end{pmatrix}, \; L = \begin{pmatrix} 0 & & & \mathbf{O} \\ a_{1,1} & \ddots & \ddots & \\ \vdots & & & \\ a_{n,1} & \dots & a_{n,n-1} & 0 \end{pmatrix}, \; R = \begin{pmatrix} 0 & a_{1,2} & \dots & a_{1,2} \\ & \ddots & & \vdots \\ & & \ddots & a_{n-1,n} \\ \mathbf{O} & & & 0 \end{pmatrix}.$$

Das Gesamtschritt- oder Jacobi-Verfahren zur Lösung von $(D+L+R)x = b$ lautet

$$Dx^{k+1} + Lx^k + Rx^k = b\,,$$

während das Einzelschritt- oder Gauß-Seidel-Verfahren gemäß

$$(D+L)x^{k+1} + Rx^k = b$$

iteriert. Elementweise lauten Gesamt- bzw. Einzelschrittverfahren

$$
\begin{aligned}
x_i^{k+1} &= \left(b_i - \sum_{j \neq i} a_{i,j} x_j^k \right) / a_{i,i} \,, \\
x_i^{k+1} &= \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{k+1} - \sum_{j=1+1}^{n} a_{i,j} x_j^k \right) / a_{i,i} \,.
\end{aligned}
$$

Der rechentechnische Unterschied zwischen den beiden Verfahren zeigt sich am deutlichsten bei der Programmierung

$GS(X,n)$ /$\star$ Führt einen Gesamtschritt durch $\star$/
{ for $i = 1, \ldots, n$
    $x_i^1 = (b_i - \sum_{j \neq i} a_{i,j} x_j)/a_{i,i};$
    $x = x^1;$
}
$ES(X,n)$ /$\star$ Führt einen Einzelschritt durch $\star$/
{ for $i = 1, \ldots, n$
    $x_i = (b_i - \sum_{j \neq i} a_{i,j} x_j)/a_{i,i};$
}

Die Iterationsvorschrift des Einzelschrittverfahrens wird also durch sukzessives Überschreiben ausgeführt. Man braucht also nicht zwei Vektoren x, x^1 zu halten, sondern kommt mit einem aus. Da x häufig sehr groß ist, ist dies von Vorteil. Darüber hinaus werden wir sehen, daß das Einzelschrittverfahren häufig schneller konvergiert als das Gesamtschrittverfahren.

Beispiel: Wir wollen das Dirichlet-Problem

$$-\Delta u = f \quad \text{in } \Omega \,, \quad u = 0 \quad \text{auf } \partial\Omega$$

für das Quadrat $\Omega = (0,1)^2$ in $\mathbb{R}^2$ lösen. Hier ist $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$ der Laplace-Operator. Man überdeckt Ω durch ein Gitter mit Schrittweite $h = \frac{1}{n}$ und ersetzt die Differentialgleichung im Gitterpunkt $\begin{pmatrix} i \\ j \end{pmatrix} h$ durch

$$4u_{i,j} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = h^2 f_{i,j} \,, \quad i,j = 1, \ldots, n-1 \,.$$

Dies ist ein lineares System von $(n-1)^2$ Gleichungen in ebensovielen Unbekannten $u_{i,j}$; die $u_{i,j}$ mit $i,j \in \{0,n\}$ werden Null gesetzt. Ist h hinreichend klein, so hofft man, daß $u_{i,j}$ eine gute Näherung für $u\left(\begin{pmatrix} i \\ j \end{pmatrix} h\right)$ ist.

Ein Einzelschritt lautet nun einfach

$$\begin{aligned} \text{for } i &= 1,\ldots,n-1 \quad \text{for } j = 1,\ldots,n-1 \\ u_{i,j} &= (h^2 f_{i,j} + u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})/4 \; ; \end{aligned}$$

Beide Verfahren kann man beschleunigen durch Einführung eines "Relaxationsparameters" ω. Für das Gesamtschrittverfahren setzt man

$$x_i^{k+1} = (1-\omega)x_i^k + \omega(b_i - \sum_{j\neq i} a_{i,j}x_j^k)/a_{i,i} \; ,$$

und für das Einzelschrittverfahren

$$x_i^{k+1} = (1-\omega)x_i^k + \omega\left(b_i - \sum_{j=1}^{i-1} a_{i,j}x_j^{k+1} - \sum_{j=i+1}^{n} a_{i,j}x_j^k\right)/a_{i,i} \; .$$

Man bildet also gewichtete Summen der k-ten und der $k+1$-ten Näherung. Das letzte Verfahren heißt SOR (successive overrelaxation) und spielt eine wichtige Rolle. Für die Programmierung von SOR gilt das über das Einzelschrittverfahren Gesagte.

In Matrizenschreibweise haben alle diese Verfahren die Form

$$x^{k+1} = Bx^k + c \; .$$

Dabei ist für das Gesamtschrittverfahren

$$B = -D^{-1}(L+R) \; , \quad c = D^{-1}b \; ,$$

für das Einzelschrittverfahren

$$B = -(D+L)^{-1}R \; , \quad c = (D+L)^{-1}b$$

und für das SOR-Verfahren

$$B = (D+\omega L)^{-1}((1-\omega)D - \omega R) \; , \quad c = \omega(D+\omega L)^{-1}b \; .$$

4.2 Konvergenz

Alle unsere Konvergenzaussagen werden auf folgendem Satz beruhen:

Satz 4.2.1 *Das Iterationsverfahren*

$$x^{k+1} = Bx^k + c$$

konvergiert genau dann für jede Wahl von x^0 *und* c*, wenn* $\rho(B) < 1$*. In diesem Fall konvergiert es gegen die eindeutig bestimmte Lösung* $\overline{x}$ *von* $\overline{x} = B\overline{x} + c$*.*

Beweis: Ist $\rho(B) < 1$, so gibt es nach Satz I.3.2 eine Norm $\|\cdot\|$, so daß $\|B\| < 1$. Damit ist $g(x) = Bx + c$ kontrahierend im ganzen Raum. Konvergenz gegen $\overline{x}$ folgt dann aus Satz III.2.2.

Sei umgekehrt das Verfahren für alle x^0, c konvergent. Wir wählen für x^0, c den Eigenvektor y von B mit Eigenwert λ. Dann ist

$$x^k = (1 + \lambda + \cdots + \lambda^k)y \; .$$

Dies ist nur konvergent für $|\lambda| < 1$. Also ist $\rho(B) < 1$.

$\square$

Definition 4.2.1 *Eine* (n,n)*-Matrix* A *über* K *erfüllt das starke Zeilensummenkriterium, wenn* $|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|, \quad i = 1, \ldots, n$*. Sie erfüllt das schwache Zeilensummenkriterium, wenn diese Ungleichungen mit mindestens einer Ausnahme im schwachen Sinne (d.h. mit* $\geq$ *an Stelle von* $>$*) gelten.*

A *heißt reduzibel, wenn man* $\{1, \ldots, n\}$ *so in zwei disjunkte nichtleere Mengen* I*,* J *aufteilen kann, daß* $a_{i,j} = 0$ *für* $(i,j) \in I \times J$*. Andernfalls heißt* A *irreduzibel.*

Ist A reduzibel, so kann man A durch Zeilen- und Spaltenvertauschungen in die Form

$$A = \begin{pmatrix} A_1 & 0 \\ A_3 & A_2 \end{pmatrix}$$

bringen. Das Gleichungssystem $Ax = b$ zerfällt dann in zwei kleinere Systeme mit den Matrizen A_1, A_2.

Satz 4.2.2 *Für die (n, n)-Matrix A sei eine der folgenden Bedingungen erfüllt:*

(a) A erfülle das starke Zeilensummenkriterium.

(b) A erfülle das schwache Zeilensummenkriterium und sei irreduzibel.

Dann konvergiert das Gesamtschrittverfahren zu Lösung von $Ax = b$ für jedes b und x^0 gegen die eindeutig bestimmte Lösung $\overline{x}$ von $A\overline{x} = b$.

Beweis: Wir zeigen, daß $\rho(D^{-1}(L + R)) < 1$, wo $A = D + L + R$. Unter der Voraussetzung (a) ist

$$\|D^{-1}(L+R)\|_\infty = \max_{i=1}^{n} \frac{1}{|a_{i,i}|} \sum_{j \neq i} |a_{i,j}| < 1 ,$$

also in der Tat $\rho(D^{-1}(L + R)) < 1$. Unter der Voraussetzung (b) folgt nur $\rho(D^{-1}(L + R)) \leq 1$. Es genügt daher zu zeigen, daß $D^{-1}(L + R)$ keinen Eigenwert λ mit $|\lambda| = 1$ hat. Wäre λ ein solcher und x ein dazugehöriger Eigenvektor mit $\|x\|_\infty = 1$, so hätten wir für alle $i \in I = \{i : |x_i| = 1\}$

$$1 = |x_i| = |\lambda||x_i| = (D^{-1}(L+R)x)_i| = \frac{1}{|a_{i,i}|} \left| \sum_{j \neq i} a_{i,j} x_j \right| \leq 1 ,$$

also

$$\left| \sum_{j \neq i} a_{i,j} x_j \right| = |a_{i,i}| , \quad i \in I .$$

Wegen des schwachen Zeilensummenkriteriums kann $J = \{1, \ldots, n\} \setminus I$ nicht leer sein. Da A irreduzibel ist, gibt es $(i_0, j_0) \in I \times J$ mit $a_{i_0,j_0} \neq 0$. Dann ist

$$
\begin{aligned}
1 = |\lambda||x_{i_0}| &= |(D^{-1}(L+R)x)_{i_0}| = \frac{1}{|a_{i_0,i_0}|}\left|\sum_{j\neq i_0} a_{i_0,j}x_j\right| \\
&< \frac{1}{|a_{i_0,i_0}|}\sum_{j\neq i_0}|a_{i_0,j}| \le 1 \; .
\end{aligned}
$$

Dieser Widerspruch zeigt, daß es keinen solchen Eigenwert λ geben kann.

□

Satz 4.2.3 *Die (n,n)-Matrix A sei positiv definit. Dann konvergiert das SOR-Verfahren zur Lösung von $Ax = b$ genau dann für jede Wahl von x^0 und b, wenn $0 < \omega < 2$.*

Beweis: Wir haben zu zeigen, daß für die SOR-Matrix $B_\omega = (D+\omega L)^{-1}((1-\omega)D - \omega R)$ der Spektralradius $\rho(B_\omega) < 1$ ist für $0 < \omega < 2$. Sei x ein Eigenvektor von B_ω zum Eigenwert λ mit $|\lambda| = \rho(B_\omega)$, also

$$((1-\omega)D - \omega R)x = \lambda(D+\omega L)x \; .$$

Inneres Produkt mit x ergibt

$$(1-\omega)(Dx,x) - \omega(Rx,x) = \lambda((Dx,x) + \omega(Lx,x)) \; .$$

Wir setzen $d = (Dx,x)$, $\ell = (Lx,x)$. Weil A hermitesch ist, gilt $(Rx,x) = (L^*x,x) = (x,Lx) = \overline{\ell}$, also

$$(1-\omega)d - \omega\overline{\ell} = \lambda(d+\omega\ell)$$

oder

$$\lambda = \frac{(1-\omega)d - \omega\overline{\ell}}{d+\omega\ell} \; .$$

Mit $\ell = \alpha + i\beta$, α, $\beta \in \mathbb{R}$ bekommen wir, weil $d > 0$,

$$|\lambda|^2 = \frac{((1-\omega)d - \omega\alpha)^2 + \omega^2\beta^2}{(d+\omega\alpha)^2 + \omega^2\beta^2} \; .$$

Es ist also genau dann $|\lambda| < 1$, wenn

$$|(1-\omega)d - \omega\alpha| < |d + \omega\alpha| \ .$$

Mit $\alpha' = \alpha/d$ lautet dies

$$|1 - \omega - \omega\alpha'| < |1 + \omega\alpha'| \ . \tag{2.1}$$

Da A positiv definit ist, gilt

$$0 < (Ax, x) = d + \ell + \overline{\ell} = d + 2\alpha = d(1 + 2\alpha') \ ,$$

also $\alpha' > -1/2$. Für $0 < \omega < 2$ können wir die Betragsstriche bei $|1 + \omega\alpha'|$ in (2.1) weglassen und erhalten

$$|1 - \omega - \omega\alpha'| < 1 + \omega\alpha'$$

als Bedingung für $|\lambda| < 1$. Dies ist aber für $0 < \omega < 2$ richtig.

□

Die Bedingung an ω kann man nicht abschwächen. Es gilt nämlich

Satz 4.2.4 *Sei B_ω die SOR-Matrix für eine beliebige Matrix mit nichtverschwindenden Diagonalelementen. Dann gilt*

$$\rho(B_\omega) \geq |\omega - 1| \ .$$

Beweis: Es ist

$$B_\omega = (I + \omega D^{-1} L)^{-1}((1-\omega)I - \omega D^{-1} R) \ .$$

Die Diagonale von B_ω ist also $(1-\omega)I$. Sind $\lambda_1, \ldots, \lambda_n$ die Eigenwerte von A (nicht notwendig verschieden), so gilt

$$\rho(B_\omega)^n \geq |\lambda_1 \cdots \lambda_n| = |1 - \omega|^n \ ,$$

und daraus folgt die Behauptung.

□

4.3 Das Verfahren der konjugierten Gradienten (CG)

Ein Problem des SOR-Verfahrens liegt in der Bestimmung des optimalen Wertes von ω. Das CG-Verfahren kommt ohne einen solchen Parameter aus und hat trotzdem gute Konvergenzeigenschaften.

Sei A eine reelle positiv definite (n,n)-Matrix. Zu lösen sei $Ax = b$. Wir setzen

$$f(x) = \frac{1}{2}(x, Ax) - (x, b) \ .$$

Satz 4.3.1 *$\overline{x}$ ist genau dann Lösung von $A\overline{x} = b$, wenn $f(\overline{x}) \leq f(x)$ für alle $x \in \mathbb{R}^n$.*

Beweis: Ist $\overline{x}$ die gesuchte Lösung, so ist

$$\frac{1}{2}(A(x-\overline{x}), (x-\overline{x})) = \frac{1}{2}(Ax, x) - (x, b) + \frac{1}{2}(A\overline{x}, \overline{x}) \ .$$

$\overline{x}$ ist also der eindeutig bestimmte Vektor, welcher $f(x)$ minimiert.

□

Anstelle einer Lösung von $Ax = b$ können wir also genausogut das Minimum von $f(x)$ suchen. Sei x^0 eine Ausgangsnäherung für $\overline{x}$. Wir suchen eine bessere Näherung für $\overline{x}$ in der Form $x^0 + \alpha d^0$, wo $d^0 \neq 0$ eine “Suchrichtung” ist. Wir bestimmen α so, daß $f(x^0 + \alpha d^0)$ minimal ist. Wegen $f'(x) = Ax - b$ ist

$$\begin{aligned}
\frac{d}{d\alpha} f(x^0 + \alpha d^0) &= (f'(x^0 + \alpha d^0), d^0) \\
&= (A(x^0 + \alpha d^0) - b, d^0) \\
&= (r^0, d^0) + \alpha(Ad^0, d^0) \ , \\
\frac{d^2}{d\alpha^2} f(x^0 + \alpha d^0) &= (Ad^0, d^0) > 0 \ ,
\end{aligned}$$

wobei wir hier und unten $r^k = Ax^k - b$ setzen. Entlang $x^0 + \alpha d^0$ ist also f genau dann minimal, wenn die erste Ableitung von f verschwindet. Dies ist für

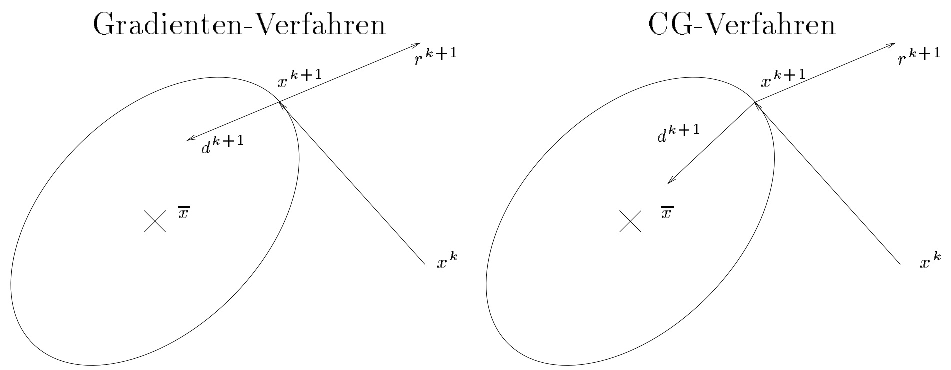
$$\alpha_0 = -\frac{(r^0, d^0)}{(Ad^0, d^0)}$$

der Fall. Geometrisch bedeutet diese Wahl, daß $r^1 \perp d^0$, und d^0 liegt im Tangentialraum des Ellipsoids $f(x) = f(x^1)$ im Punkte x^1. Die neue Approximation für $\overline{x}$ ist dann $x^1 = x^0 + \alpha d^0$.

Wie wählt man nun d^0? Eine naheliegende Wahl ist $d^0 = -r^0$, d.h. man läuft in Richtung des steilsten Abstiegs von f. Dies führt zu den Verfahren des steilsten Abstiegs (steepest descent) oder Gradientenverfahren:

Initialisierung	:	Wähle x^0, $r^0 = Ax^0 - b$, $d^0 = -r^0$.
Für $k = 0, 1, 2, \ldots$	:	$z = Ad^k$
		$\alpha_k = -\dfrac{(d^k, d^k)}{(z, d^k)}$
		$x^{k+1} = x^k + \alpha_k d^k \quad , \quad d^{k+1} = d^k + \alpha_k z$.

Die Suchrichtung beim k-ten Schritt ist $d^k = -r^k$.



Wir haben gesehen, daß für das Gradientenverfahren $r^{k+1} \perp d^k$ ist, also $d^{k+1} \perp d^k$. Dies bedeutet, daß x^k sich $\overline{x}$ auf einem Zick-Zack-Kurs annähert. Dies läßt für die Konvergenzgeschwindigkeit nichts Gutes hoffen.

Beim CG-Verfahren ersetzt man die eindimensionale Minimierung durch eine zweidimensionale. Sind x^0, x^1 nach den Gradientenverfahren berechnet, so setzt man

$$x^2 = x^1 - \alpha r^1 + \beta d^0$$

und bestimmt α, β so, daß $f(x^2)$ minimal ist. Die Bedingungen für ein Minimum sind

$$\begin{aligned} \frac{\partial f}{\partial \alpha}(x^2) &= (Ax^2 - b, r^1) = 0 \\ \frac{\partial f}{\partial \beta}(x^2) &= (Ax^2 - b, d^0) = 0 \ . \end{aligned}$$

Auflösen dieser Gleichungen nach α, β liefert x^2. Wir benutzen nur die zweite Gleichung, welche wir in der Form

$$(r^1 - \alpha A r^1 + \beta A d^0, d^0) = 0$$

schreiben können. Wegen $(r^1, d^0) = 0$ folgt

$$\beta = \alpha \beta_0 \quad , \quad \beta_0 = \frac{(Ar^1, d^0)}{(Ad^0, d^0)}$$

und damit

$$x^2 = x^1 + \alpha d^1 \quad , \quad d^1 = -r^1 + \beta_0 d^0 \ .$$

Nun können wir α durch eine eindimensionale Minimierung bestimmen. Das Resultat α_1 kennen wir schon vom Gradientenverfahren, nämlich

$$\alpha_1 = -\frac{(r^1, d^1)}{(d^1, Ad^1)} \ .$$

Damit ist x^2 bestimmt, allerdings in einer sehr unpraktischen Form: Die Berechnung von x^2 erfordert drei Anwendungen der Matrix A auf einen Vektor. Wir wollen mit einer Anwendung auskommen. Dazu schreiben wir β_0 in der Form

$$\beta_0 = \frac{(r^1, Ad^0)}{(d^0, Ad^0)} = \frac{(r^1, (r^1 - r^0)/\alpha_0)}{(d^0, Ad^0)} = \frac{(r^1, r^1)}{(r^0, r^0)} \ ,$$

wobei, wie beim Gradientenverfahren, $\alpha_0 = (r^0, r^0)/(d^0, Ad^0)$. Für α_1 haben wir die α_0 entsprechende Form

$$\alpha_1 = -\frac{(r^1, -r^1 + \beta_0 d^0)}{(d^1, Ad^1)} = \frac{(r^1, r^1)}{(d^1, Ad^1)} ,$$

denn $(r^1, d^0) = 0$. Damit lautet das CG-Verfahren:

$$\begin{array}{lll}
\text{Initialisierung} & : & \text{Wähle } x^0,\ r^0 = Ax^0 - b,\ \ d^0 = -r^0 \\
\text{Für } k = 0, 1, \ldots & : & z = Ad^k \\
 & & \alpha_k = \dfrac{(r^k, r^k)}{(d^k, z)} \\
 & & x^{k+1} = x^k + \alpha_k d^k \ ,\ \ r^{k+1} = r^k + \alpha_k z \\
 & & \beta_k = \dfrac{(r^{k+1}, r^{k+1})}{(r^k, r^k)} \\
 & & d^{k+1} = -r^{k+1} + \beta_k d^k \quad .
\end{array}$$

Für $k = 0$ macht also das CG-Verfahren einen Gradientenschritt. Die weiteren Schritte führen wie das Gradientverfahren eine eindimensionale Minimierung durch, aber nicht entlang des steilsten Abstiegs $-r^k$, sondern entlang der Richtung d^k. Das Verfahren bricht ab, wenn $r^k = 0$. In diesem Fall ist $x^k = \overline{x}$.

Satz 4.3.2 *Das CG-Verfahren bricht nach höchstens n Schritten mit $r^k = 0$ ab. Solange es nicht abbricht, gilt*

(a) $(d^i, r^k) = 0$ *für* $i < k$,

(b) $(r^k, d^k) = -\|r^k\|^2$,

(c) $(r^i, r^k) = 0$ *für* $i < k$,

(d) $(d^i, Ad^k) = 0$ *für* $i < k$,

(e) $sp\ \{d^0, \ldots, d^k\} = sp\ \{r^0, \ldots, r^k\}$.

Beweis: Wir beweisen (a) - (e) durch Induktion nach k. Die Aussage über den Abbruch folgt dann aus (c), weil es nicht mehr als n linear unabhängige Vektoren r^k gibt. Solange $r^k \neq 0$ ist, ist nach (b) auch $d^k \neq 0$. Das Verfahren bricht also nur mit $r^k = 0$ ab. Für $k = 0$ sind (a) - (e) trivial. Sei (a) - (e) richtig für ein $k \geq 0$. Wir zeigen, daß (a) - (e) dann auch für $k+1$ richtig ist, wenn nur $r^{k+1} \neq 0$.

(a) Es ist für $i < k$

$$(d^i, r^{k+1}) = (d^i, r^k + \alpha_k A d^k) = 0$$

wegen (a) und (d) für k. Für $i = k$ ist

$$(d^k, r^{k+1}) = 0 \ ,$$

weil x^{k+1} durch Minimierung entlang d^k entsteht.

(b) Es ist

$$(r^{k+1}, d^{k+1}) = (r^{k+1}, -r^{k+1} + \beta_k d^k) = -\|r^{k+1}\|^2 \ ,$$

weil (a) auch für $k+1$ gilt.

(c) Weil (a) auch für $k+1$ gilt, ist für $i < k+1$ mit $\beta_{-1} = 0$

$$(r^i, r^{k+1}) = (-d^i + \beta_{i-1} d^{i-1}, r^{k+1}) = 0 \ .$$

(d) Aus (b) für $k+1$ folgt $d^{k+1} \neq 0$. Für $i < k$ ist

$$\begin{aligned} (d^i, Ad^{k+1}) &= (d^i, A(-r^{k+1} + \beta_k d^k)) \\ &= -(Ad^i, r^{k+1}) + \beta_k (d^i, Ad^k) \\ &= \tfrac{1}{\alpha_i}(r^i - r^{i+1}, r^{k+1}) + \beta_k (d^i, Ad^k) \ . \end{aligned}$$

Dies verschwindet wegen (c) für $k+1$ und (d) für k. Für $i = k$ ist

$$\begin{aligned} (d^k, Ad^{k+1}) &= (Ad^k, d^{k+1}) = \tfrac{1}{\alpha_k}(r^{k+1} - r^k, -r^{k+1} + \beta_k d^k) \\ &= \tfrac{1}{\alpha_k}((r^{k+1}, r^{k+1}) - \beta_k (r^k, d^k)) \end{aligned}$$

wegen (a), (c) für $k+1$, und dies verschwindet nach Definition von β_k.

(e) Es ist sp $\{d^0, \ldots, d^k\} =$ sp $\{r^0, \ldots, r^k\}$ und $d^{k+1} = -r^{k+1} + \beta_k d^k$. Also ist auch sp $\{d^0, \ldots, d^{k+1}\} =$ sp $\{r^0, \ldots, r^{k+1}\}$.

$\square$

Bemerkung: Man nennt Vektoren x, y mit $(x, Ay) = 0$ konjugiert (bezüglich A). Nach Teil (d), (e) des Satzes entstehen die Richtungen d^k aus den Gradienten r^k durch Orthogonalisierung bezüglich des inneren Produktes (x, Ay). Dies erklärt den Namen "konjugierte Gradienten".

Der folgende Satz gibt eine Erklärung für die Effizienz des CG-Verfahrens.

Satz 4.3.3 *Solange das CG-Verfahren nicht abbricht, minimiert x^k die Funktion $f(x)$ in $x^0 +$ sp $\{d^0, \ldots, d^{k-1}\}$.*

Beweis: Sei $f(x)$ minimal für $x = x^0 + \sum_{i=0}^{k-1} c_i d^i$, also

$$\frac{\partial}{\partial c_i} f\left(x^0 + \sum_{i=0}^{k-1} c_i d^i\right) = 0 \ , \qquad i = 0, \ldots, k-1 \ .$$

Wegen

$$f\left(x^0 + \sum_{i=0}^{k-1} c_i d^i\right) = \frac{1}{2}\left(x^0 + \sum_{i=0}^{k-1} c_i d^i, A\left(x^0 + \sum_{i=0}^{k-1} c_i d^i\right)\right) - \left(x^0 + \sum_{i=0}^{k-1} c_i d^i, b\right)$$

$$= \ f(x^0) + \frac{1}{2}\sum_{i,j=0}^{k-1} c_i c_j (d^i, Ad^j) + \sum_{i=0}^{k-1} c_i (d^i, Ax^0 - b)$$

$$= \ f(x^0) + \frac{1}{2}\sum_{i=0}^{k-1} c_i^2 (d^i, Ad^i) + \sum_{i=0}^{k-1} c_i (d^i, r^0) \ ,$$

wo wir Satz 3.1 (d) benutzt haben, gilt

$$c_i = -\frac{(d^i, r^0)}{(d^i, Ad^i)} \ , \qquad i = 0, \ldots, k-1 \ .$$

Nach Konstruktion des CG-Verfahrens ist

$$\begin{aligned} d_i &= -r^i + \beta_{i-1} d^{i-1} \\ &= -r^i - \beta_{i-1}(r^{i-1} + \beta_{i-2} d^{i-2}) \\ &= \cdots \\ &= -r^i - \beta_{i-1} r^{i-1} - \beta_{i-1}\beta_{i-2} r^{i-2} - \cdots + \beta_{i-1} \cdots \beta_0 r^0 \ . \end{aligned}$$

Nach Satz 3.1 (c) gilt also

$$(d^i, r^0) = -\beta_{i-1} \cdots \beta_0 (r^0, r^0) = (r^i, r^i)$$

aufgrund der Definition der β_k. Also haben wir

$$c_i = \frac{(r^i, r^i)}{(d^i, Ad^i)} = \alpha_i \ , \qquad i = 0, \ldots, k-1 \ .$$

Also gilt für den minimalen Vektor x in der Tat

$$x = x^0 + \sum_{i=0}^{k-1} c_i d^i = x^0 + \sum_{i=0}^{k-1} \alpha_i d^i = x^k \ .$$

□

Satz 4.3.4 *Für das CG-Verfahren gilt die Fehlerabschätzung*

$$\|x^k - \overline{x}\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|x^0 - \overline{x}\|_A \ ,$$

wo κ die Kondition von A in der euklidischen Norm und $\| \cdot \|_A$ die Energie-Norm $\|x\|_A = (x, Ax)^{1/2}$ bedeutet.

Beweis: Nach Satz 3.1 ist $f(x) = \frac{1}{2}\|x - \overline{x}\|_A^2 - \frac{1}{2}\|\overline{x}\|_A^2$. Nach Satz 3.3 minimiert x^k die Funktion $f(x)$ in x^0+ sp $\{d^0, \ldots, d^{k-1}\}$, also minimiert x^k auch $\|x - \overline{x}\|_A$ in diesen Raum, der nach Satz 3.2 identisch ist mit x^0+ sp $\{r^0, \ldots, r^{k-1}\}$. Wir zeigen

$$\mathrm{sp}\{r^0, \ldots, r^{k-1}\} = \mathrm{sp}\{r^0, Ar^0, \ldots, A^{k-1} r^0\} \ . \tag{3.1}$$

Für $k = 1$ ist dies richtig. Sei es richtig für ein $k \geq 1$. Es ist $r^k = r^{k-1} + \alpha_{k-1} A d^{k-1}$, und nach Satz IV.3.2 (e) ist $d^{k-1} \in$ sp $\{r^0, \ldots, r^{k-1}\} =$ sp $\{r^0, \ldots, A^{k-1} r^0\}$, also $Ad^{k-1} \in$ sp $\{r^0, \ldots, A^k r^0\}$ und damit $r^k \in$ sp $\{r^0, \ldots, A^k r^0\}$. Umgekehrt ist dann auch $A^k r^0 \in$ sp $\{r^0, \ldots, r^k\}$. Damit ist (3.1) gezeigt.

x^k hat also die Gestalt

$$x^k = x^0 + c_0 r^0 + \cdots c_{k-1} A^{k-1} r^0$$

oder

$$\begin{aligned} x^k - \overline{x} &= x^0 - \overline{x} + (c_0 I + \cdots + c_{k-1} A^k)(x^0 - \overline{x}) \\ &= Q(A)(x^0 - \overline{x}) \end{aligned}$$

mit einem Polynom Q der Ordnung k und $Q(0) = 1$. Da A positiv definiti ist, existiert $A^{1/2}$, und $A^{1/2}A = AA^{1/2}$. Also ist

$$\begin{aligned} \|x^k - \overline{x}\|_A &= (x^k - \overline{x}, A(x^k - \overline{x}))^{1/2} \\ &= \|A^{1/2}(x^k - \overline{x})\|_2 \\ &= \|A^{1/2} Q(A)(x^0 - \overline{x})\|_2 \\ &= \|Q(A) A^{1/2}(x^0 - \overline{x})\|_2 \\ &\leq \|Q(A)\|_2 \|A^{1/2}(x^0 - \overline{x})\|_2 \\ &= \|Q(A)\|_2 \|x^0 - \overline{x}\|_A \ . \end{aligned}$$

Seien $0 < \lambda_1 \leq \cdots \leq \lambda_n$ die Eigenwerte von A. Dann ist

$$\|Q(A)\|_2 = \max_{\lambda_1 \leq \lambda \leq \lambda_n} |Q(\lambda)| \ .$$

Wir suchen nun Q so zu bestimmen, daß $Q(1) = 1$ und $|Q|$ möglichst klein in $[\lambda_1, \lambda_n]$. Dazu führen wir die Tschebitscheff-Polynome T_k erster Art ein. Es ist

$$T_0(x) = 1 \ , \ T_1(x) = x \ , \ T_{k+1}(x) + T_{k-1}(x) = 2x T_k(x) \ .$$

Man bestätigt leicht, daß

$$T_k(x) = \cos kt \quad , \quad x = \cos t \quad , \quad 0 \leq t \leq \pi \ .$$

Wir setzen

$$Q(x) = \frac{T_k\left(\dfrac{\lambda - \lambda_1 - \lambda_n}{\lambda_1 - \lambda_n}\right)}{T_k\left(-\dfrac{\lambda_1 + \lambda_n}{\lambda_1 - \lambda_n}\right)}$$

Offenbar ist Q ein Polynom vom Grade k mit $Q(0) = 1$, und es ist

$$\max_{\lambda_1 \le \lambda \le \lambda_n} |Q(\lambda)| = \frac{1}{|T_k(-\frac{\lambda_1 - \lambda_n}{\lambda_1 - \lambda_n})|}$$

Nun gilt aber für $x \ge 1$

$$T_k(x) \ge \frac{1}{2}(x + \sqrt{x^2 - 1})^k \,. \tag{3.2}$$

Dazu setzen wir $x = \cosh t$, also $t = \operatorname{arccosh} x = \ell n(x + \sqrt{x^2 - 1})$. Aus der Rekursion für T_k folgt

$$\begin{aligned} T_k(x) &= \cosh kt \\ &\ge \frac{1}{2} e^{kt} \\ &= \frac{1}{2}(x + \sqrt{x^2 - 1})^k \end{aligned}$$

und dies ist (3.2). Mit $x = \frac{\lambda_1 + \lambda_n}{\lambda_1 - \lambda_n} = \frac{1+\kappa}{1-\kappa}$, $\kappa = \kappa(A) = \frac{\lambda_n}{\lambda_1}$ haben wir

$$x + \sqrt{x^2 - 1} = \frac{\sqrt{\kappa} + 1}{\sqrt{\kappa} - 1} \,.$$

Damit folgt die Behauptung.

$\square$

4.4 Vorkonditionierung

Die großen linearen Gleichungssysteme mit dünnbesetzter Matrix, für welche iterative Verfahren besonders geeignet sind, haben häufig eine sehr große Kondition. Wir wollen uns dies an dem Beispiel

$$-\Delta u = f \quad \text{in} \quad \Omega = (0,1)^2 \ , \quad u = 0 \quad \text{auf } \partial\Omega$$

klarmachen. Die diskrete Version aus §1 ist

$$\begin{aligned} 4u_{k,\ell} - u_{k+1,\ell} - u_{k-1,\ell} - u_{k,\ell+1} - u_{k,\ell-1} &= h^2 f_{k,\ell} \\ k,\ell = 1,\dots,n-1 \ , \quad h &= 1/n \ , \\ u_{k,\ell} = 0 \quad \text{falls} \quad k \in \{0,n\} \ \text{ oder } \ \ell &\in \{0,n\} \ . \end{aligned} \tag{4.1}$$

Dabei bedeutet $u_{k,\ell}$ eine Näherung für u in dem Gitterpunkt $(hk, h\ell)$. Wir numerieren die Gitterpunkte und die Gleichungen (4.1) etwa zeilenweise von 1 bis $N = (n-1)^2$. Man sieht dann leicht, daß die Matrix A von (4.1) eine dünnbesetzte symmetrische (N,N)-Matrix ist.

Wir berechnen die Eigenwerte von A. Mit $v_k = \sin \alpha\pi kh$ ist wegen der Additionstheoreme

$$2v_k - v_{k+1} - v_{k-1} = 2(1 - \cos \alpha\pi h) v_k \ .$$

Mit $v_{k,\ell} = \sin \alpha\pi kh \sin \beta\pi \ell h$ ist also

$$4v_{k,\ell} - v_{k+1,\ell} - v_{k-1,\ell} - v_{k,\ell+1} - v_{k,\ell-1} = 2(2 - \cos \alpha\pi h - \cos \beta\pi h) v_{k,\ell} \ .$$

Wählen wir nun α, β ganz und $1 \le \alpha$, $\beta < n$, so ist $v_{k,\ell} = 0$ für $k \in \{0,n\}$ oder $\ell \in \{0,n\}$ und damit der Vektor $(v_{k,\ell})_{k,\ell=1,\dots,n-1}$ Eigenvektor von A zum Eigenwert

$$\lambda_{\alpha,\beta} = 2(2 - \cos \alpha\pi h - \cos \beta\pi h) \ .$$

Damit haben wir alle N Eigenwerte von A gefunden. Die Kondition $\kappa(A)$ von A bezüglich der euklidischen Norm berechnet sich nun als Quotient von größtem und kleinstem Eigenwert von A, also

$$\kappa(A) = \frac{\lambda_{n-1,n-1}}{\lambda_{1,1}} = \frac{1 + \cos h\pi}{1 - \cos h\pi} \ .$$

Für kleine h verhält sich dies wie $4/(h\pi)^2$, ist also sehr groß.

Berechnen wir die Konvergenzgeschwindigkeit des Gesamtschrittverfahrens! Die Iterationsmatrix ist $B = -D^{-1}(L+R)$ mit $A = D+L+R$, $D = 4I$. Sei μ ein Eigenwert von B. Dann ist $(1-\mu)/4$ Eigenwert von A, also $\mu = 1-\lambda/4$. Die Eigenwerte von B sind also

$$\mu_{\alpha,\beta} = 1 - \lambda_{\alpha,\beta}/4 = \frac{1}{2}(\cos\alpha\pi h + \cos\beta\pi h) \; .$$

Damit erhalten wir

$$\rho(B) = \max_{1\le\alpha,\beta<n} |\mu_{\alpha,\beta}| = \cos\pi h \; .$$

Für kleine h verhält sich dies wie $1 - \pi^2 h^2/2$. Die Konvergenz des Gesamtschrittverfahrens ist also sehr langsam.

Gesamtschritt- und CG-Verfahren konvergieren also beide langsam, doch mit einem bedeutenden Unterschied: Während sich der Fehler beim Gesamtschrittverfahren wie $(1-\pi^2h^2/2)^k$ verhält, verhält er sich beim CG-Verfahren (allerdings in einer anderen Norm) wie $(1-h\pi)^k$. Das CG-Verfahren konvergiert also weniger langsam als das Gesamtschrittverfahren. Man kann zeigen, daß das SOR-Verfahren bei optimaler Wahl von ω ungefähr so schnell wie das CG-Verfahren konvergiert. Insgesamt muß man also bei allen diesen Verfahren mit langsamer Konvergenz rechnen, und zwar wird die Konvergenz umso langsamer, je kleiner h wird.

Abhilfe schafft hier die Vorkonditionierung. Dazu schreiben wir $Ax = b$ in der Form

$$C^{-1}A(C^*)^{-1}C^*x = C^{-1}b \tag{4.2}$$

mit einer nichtsingulären Matrix C. C wird so gewählt, daß

$$\kappa(C^{-1}A(C^*)^{-1}) \ll \kappa(A) \; .$$

(4.2) hat dann die Gestalt

$$\tilde{A}\tilde{x} = \tilde{b} \; ,$$

$$\tilde{A} = C^{-1}A(C^*)^{-1} \; , \quad \tilde{x} = C^*x \; , \quad \tilde{b} = C^{-1}b \; ,$$

wobei $\tilde{A}$ eine viel bessere Kondition hat als A. Überdies ist $\tilde{A}$ hermitesch (positiv definit), falls dies für A der Fall ist.

Wegen

$$
\begin{aligned}
(C^{-1})^* \tilde{A} C^* &= (C^{-1})^* C^{-1} A (C^*)^{-1} C^* = M^{-1} A \, , \\
M &= CC^*
\end{aligned}
$$

ist $\kappa(\tilde{A}) = \kappa(M^{-1}A)$. Um für $\tilde{A}$ eine gute Kondition zu bekommen, wird man also $M \sim A$ wählen wollen. Gleichzeitig muß M aber eine Cholesky-Faktorisierung $M = CC^*$ besitzen, so daß C^{-1}, $(C^*)^{-1}$ leicht auf Vektoren anwendbar ist. Für $A = I + L + L^*$ ist

$$
M = (I + L)(I + L^*)
$$

eine Möglichkeit.

Andere Möglichkeiten sind Dekompositionsmethoden und Mehrgitterverfahren. Diese führen zu von h unabhängigen Konvergenzraten. Sie sind Gegenstand weiterführender Vorlesungen, wie etwa "Numerik partieller Differentialgleichungen".