3 Das Lanczos–Verfahren

3.1 Idee

Ausgehend von einem (normierten) Startvektor v_0 soll durch wiederholtes Anwenden der (hermiteschen) Matrix A eine Ortho–Normalbasis $\{v_0, v_1, ...\}$ iterativ konstruiert werden.
Erster Schritt:

$$\begin{aligned} Av_0 &= \beta_0 v_1 + \alpha_0 v_0 \\ \text{mit} \quad \alpha_0 &= v_0^\dagger A v_0 \\ \beta_0 &= \|Av_0 - \alpha_0 v_0\| \end{aligned}$$

$\alpha_0 v_0$ ist also die Projektion von $Av_0 \parallel v_0$ und β_0 sorgt als Normierungsfaktor für $\|v_1\| = 1$. Die Phase von β_0 ist willkürlich gewählt: $\beta_0 \geq 0$.
Zweiter Schritt:

$$\begin{aligned} Av_1 &= \beta_1 v_2 + \alpha_1 v_1 + \gamma_0 v_0 \\ \text{mit} \quad \alpha_1 &= v_1^\dagger A v_1 \\ \gamma_0 &= v_0^\dagger A v_1 \\ &= (v_1^\dagger A v_0)^* \\ &= \beta_0^* \\ \beta_1 &= \|Av_1 - \alpha_1 v_1 - \beta_0^* v_0\| \end{aligned}$$

Dritter Schritt:

$$\begin{aligned} Av_2 &= \beta_2 v_3 + \alpha_2 v_2 + \gamma_1 v_1 + \delta_0 v_0 \\ \text{mit} \quad \alpha_2 &= v_2^\dagger A v_2 \\ \gamma_1 &= \beta_1^* \\ \delta_0 &= v_0^\dagger A v_2 \\ &= (v_2^\dagger A v_0) \\ &= 0 \\ \beta_2 &= \|Av_2 - \alpha_2 v_2 - \beta_1^* v_1\| \end{aligned}$$

usw.

3.2 Rekursion

$$Av_n = \beta_n v_{n+1} + \alpha_n v_n + \beta_{n-1} v_{n-1} \tag{3.1}$$

$$\alpha_n = v_n^\dagger A v_n \tag{3.2}$$

$$\beta_n = \|Av_n - \alpha_n v_n - \beta_{n-1} v_{n-1}\| \tag{3.3}$$

Die Phasenwahl des Normierungsfaktors impliziert $\beta_n \geq 0$. Die Rekursion bricht nicht ab, solange $\beta_n > 0$.
Mathematisch ist die Orthogonalität *aller* v_i garantiert, obwohl zur Konstruktion von v_{n+1} explizit nur auf v_n und v_{n-1} zurückgegriffen wird. In der *numerischen* Implementierung ist die Orthogonalität von v_{n+1} zu $v_0, ..., v_{n-1}$ dagegen nicht einmal im Rahmen der Rechengenauigkeit garantiert – sie kann durch akkumulierte Rundungsfehler Schaden nehmen.
In der Praxis findet man, dass der Lanczos–Algorithmus aus dem Rauschen der Rundungsfehler gelegentlich neue Startvektoren generiert und damit Teile des Vektorraums mehrfach überdeckt.
Eine leichte Verbesserung erreicht man mit einer alternativen Formel für α_n, die wenigstes die Orthogonalität von v_{n+1} zu v_n im Rahmen der Rechengenauigkeit garantiert: aus der Rekursion (3.1) entnimmt man

$$\beta_n v_n^\dagger v_{n+1} = v_n^\dagger (A v_n - \beta_{n-1} v_{n-1}) - \alpha_n v_n^\dagger v_n$$

Da die Normierung $v_n^\dagger v_n = 1$ jeweils explizit vorgenommen wird, wird $v_n^\dagger v_{n+1} = 0$ erreicht mit

$$\alpha_n = v_n^\dagger (A v_n - \beta_{n-1} v_{n-1}) \tag{3.4}$$

Diese zu (3.2) mathematische äquivalente Form ist also numerisch vorzuziehen. Man definiert dann noch den Hilfsvektor

$$q_n \equiv A v_n - \beta_{n-1} v_{n-1} \tag{3.5}$$

und bekommt so den folgenden...

3.3 Algorithmus

Startvektor: s
Vorspann:

$$\begin{aligned} \|s\| &= 0 \quad \Rightarrow \quad \text{exit} \\ v_0 &= s/\|s\| \\ q_0 &= A v_0 \end{aligned}$$

Iterationen $n = 0, 1, 2, ...$:

$$\begin{aligned} \alpha_n &= v_n^\dagger q_n \\ q'_n &= q_n - \alpha_n v_n \\ \beta_n &= \|q'_n\| \\ \beta_n &= 0 \quad \Rightarrow \quad \text{exit} \\ v_{n+1} &= q'_n/\beta_n \\ q_{n+1} &= A v_{n+1} - \beta_n v_n \end{aligned}$$

Die beiden Arbeitsvektoren q und q' sind im Programm identisch, und man braucht insgesamt vier Vektoren (s, v_n, v_{n+1}, q_n). Wenn man den Startvektor überschreibt, kommt man sogar mit drei Vektoren aus.

3.4 Tridiagonal–Matrix

Nach Konstruktion:

$$T_{ij} = v_i^\dagger A v_j \tag{3.6}$$

$$T = \begin{pmatrix} \alpha_0 & \beta_0 & & \\ \beta_0 & \alpha_1 & \beta_1 & \\ & \beta_1 & \alpha_2 & \ddots \\ & & \ddots & \ddots \end{pmatrix} \tag{3.7}$$

d.h. A ist in der Lanczos–Basis *tridiagonal*.
Während der Iterationen erscheinen nacheinander die Untermatrizen

$$T^{(0)} = \begin{pmatrix} \alpha_0 \end{pmatrix} \tag{3.8}$$

$$T^{(1)} = \begin{pmatrix} \alpha_0 & \beta_0 \\ \beta_0 & \alpha_1 \end{pmatrix} \tag{3.9}$$

$$T^{(2)} = \begin{pmatrix} \alpha_0 & \beta_0 & \\ \beta_0 & \alpha_1 & \beta_1 \\ & \beta_1 & \alpha_2 \end{pmatrix} \tag{3.10}$$

$$\cdots$$

Man kann beweisen: solange $\beta_i \neq 0$ gilt, sind die Spektren der $T^{(n)}$ nicht entartet und nach folgendem Schema ineinander verschachtelt:

$$\begin{array}{ccccc} & & \lambda_0^{(0)} & & \\ & \lambda_0^{(1)} & & \lambda_1^{(1)} & \\ \lambda_0^{(2)} & & \lambda_1^{(2)} & & \lambda_2^{(2)} \\ & & \cdots & & \end{array} \tag{3.11}$$

Demnach ist jedes $\lambda_0^{(n)}$ eine obere Schranke für $\lambda_{min}(A)$ und jedes $\lambda_n^{(n)}$ eine untere Schranke für $\lambda_{max}(A)$. Die Erfahrung zeigt, dass gerade die extremalen Eigenwerte von $T^{(n)}$ schnell gegen die von A konvergieren.

3.5 Fehlerabschätzung

Das folgende **Theorem** ist eine der vielen Varianten des Kreistheorems von **Gershgorin** (1931):
Sei A eine normale (z.B. hermitesche, symmetrische) $(N \times N)$–Matrix, $\sigma \in C$, $v \in C^N$ und $\|v\| = 1$, ferner

$$\delta = \|(A - \sigma)v\|. \tag{3.12}$$

Dann liegt (mindestens) ein Eigenwert λ_i von A in der Kreisscheibe um σ mit Radius δ:

$$|\lambda_i - \sigma| \leq \delta \tag{3.13}$$

Beweis:
A hat eine Spektraldarstellung

$$
\begin{aligned}
A &= \sum_i u_i \lambda_i u_i^\dagger \\
\Rightarrow \quad A - \sigma &= \sum_i u_i (\lambda_i - \sigma) u_i^\dagger \\
\Rightarrow \quad (A - \sigma) v &= \sum_i u_i (\lambda_i - \sigma)(u_i^\dagger v) \\
\Rightarrow \quad \delta^2 &= \sum_i |\lambda_i - \sigma|^2 \, |u_i^\dagger v|^2 \qquad (3.14)
\end{aligned}
$$

Wäre nun $|\lambda_i - \sigma| > \delta$ für alle i, dann folgte

$$\sum_i |\lambda_i - \sigma|^2 \, |u_i^\dagger v|^2 \quad > \quad \delta^2 \sum_i |u_i^\dagger v|^2 \; = \; \delta^2$$

Das widerspricht Glg.(3.14), und so folgt die Behauptung.
Anwendung auf das Lanczos–Verfahren:
Sei $\lambda_i^{(n)}$ ein Eigenwert von $T^{(n)}$ und $h_i^{(n)}$ der zugehörige (normierte) Eigenvektor:

$$\left(T^{(n)} - \lambda_i^{(n)}\right) h_i^{(n)} \quad = \quad 0$$

Dann nehmen wir $\sigma = \lambda_i^{(n)}$ und (in der Lanczos–Basis)

$$
\begin{aligned}
v &= \begin{pmatrix} h_i^{(n)} \\ 0 \\ \vdots \end{pmatrix} \\
\Rightarrow \quad (A - \sigma) v &= \left(\begin{array}{c|c} T^{(n)} - \lambda_i^{(n)} & \cdots \\ \hline \beta_n & \cdots \\ & \cdots \end{array} \right) \left(\begin{array}{c} h_i^{(n)} \\ \hline 0 \\ .. \end{array} \right) \\
&= \left(\begin{array}{c} 0 \\ \hline \beta_n (h_i^{(n)})_n \\ 0 \\ .. \end{array} \right) \\
&= \beta_n (h_i^{(n)})_n v_{n+1} \\
\Rightarrow \quad \delta_i^{(n)} &= |\beta_n| \, |(h_i^{(n)})_n| \qquad (3.15)
\end{aligned}
$$

und nach dem Theorem hat A (mindestens) einen Eigenwert im Intervall $\lambda_i^{(n)} \pm \delta_i^{(n)}$. Diese Fehlerabschätzung ist *berechenbar*, dann die Diagonalisierung von $T^{(n)}$ lässt sich zwischen den Lanczos–Iterationen mit geringem Aufwand durchführen. Sie ist allerdings nicht besonders nützlich, wenn es darum geht, ein Abbruch–Kriterium für den Algorithmus zu formulieren, denn die Fehlerschranke *überschätzt* den tatsächlichen Fehler beträchtlich.

3.6 Lanczos und CG im Krylov–Raum

3.6.1 Details zur CG–Iteration

Matrix: $A = A^\dagger > 0$
Iteration:

$$\begin{aligned} p_0 &= r_0 \\ r_{n+1} &= r_n - \tilde{\alpha}_n A p_n \qquad (3.16) \\ p_{n+1} &= r_{n+1} + \tilde{\beta}_n p_n \qquad (3.17) \end{aligned}$$

Eigenschaften:

$$\begin{aligned} r_n^\dagger p_k &= 0 \qquad n > k \qquad (3.18) \\ p_n^\dagger A p_k &= 0 \qquad n \neq k \qquad (3.19) \\ r_n^\dagger r_k &= 0 \qquad n \neq k \qquad (3.20) \\ p_n^\dagger r_n &= r_n^\dagger r_n \qquad (3.21) \end{aligned}$$

Als erstes beweisen wir (3.18),(3.19) durch vollständige Induktion über n, d.h. wir schließen auf

$$\begin{aligned} r_{n+1}^\dagger p_k &= 0 \qquad k \leq n \\ p_{n+1}^\dagger A p_k &= 0 \qquad k \leq n \end{aligned}$$

Die Relationen mit $k = n$ dienen zur Festlegung von $\tilde{\alpha}_n$ und $\tilde{\beta}_n$:

$$\begin{aligned} r_{n+1}^\dagger p_n &= 0 \\ \Rightarrow \quad \tilde{\alpha}_n &= \frac{p_n^\dagger r_n}{p_n^\dagger A p_n} \qquad (3.22) \\ p_{n+1}^\dagger A p_n &= 0 \\ \Rightarrow \quad \tilde{\beta}_n &= -\frac{p_n^\dagger A r_{n+1}}{p_n^\dagger A p_n} \qquad (3.23) \end{aligned}$$

und für $k < n$ rechnen wir nach:

$$\begin{aligned} r_{n+1}^\dagger p_k &= (r_n^\dagger - \tilde{\alpha}_n^* p_n^\dagger A) p_k \;=\; 0 \\ p_{n+1}^\dagger A p_k &= (r_{n+1}^\dagger + \tilde{\beta}_n^* p_n^\dagger) A p_k \\ &= r_{n+1}^\dagger A p_k \;=\; 0 \\ \text{wegen} \quad \tilde{\alpha}_k A p_k &= r_k - r_{k+1} \\ &= (p_k - \tilde{\beta}_{k-1} p_{k-1}) - (p_{k+1} - \tilde{\beta}_k p_k) \\ &\in \text{span}\{p_{k-1}, p_k, p_{k+1}\} \end{aligned}$$

Die beiden anderen Relation (3.20),(3.21) folgen nun direkt:

$$\begin{aligned} r_n^\dagger r_k &= r_n^\dagger (p_k - \tilde{\beta}_{k-1} p_{k-1}) \\ &= 0 \qquad n > k \\ p_n^\dagger r_n &= (r_n^\dagger + \tilde{\beta}_{n-1}^* p_{n-1}^\dagger) r_n \\ &= r_n^\dagger r_n \end{aligned}$$

Alternative Ausdrücke für die Koeffizienten:

$$\tilde{\alpha}_n = \frac{r_n^\dagger r_n}{p_n^\dagger A p_n} \geq 0 \tag{3.24}$$

$$\begin{aligned} \tilde{\beta}_n &= -\frac{\tilde{\alpha}_n p_n^\dagger A r_{n+1}}{r_n^\dagger r_n} \\ &= \frac{(r_{n+1}^\dagger - r_n^\dagger) r_{n+1}}{r_n^\dagger r_n} \\ &= \frac{r_{n+1}^\dagger r_{n+1}}{r_n^\dagger r_n} \geq 0 \end{aligned} \tag{3.25}$$

3.6.2 Krylov–Raum

Aufgrund der CG–Iteration ist klar, dass sowohl die $\{p_k\}$ als auch die $\{r_k\}$ jeweils einen Krylov–Raum aufspannen (ausgehend von $p_0 = r_0$). Tatsächlich stellen sie zwei verschiedene Basissysteme (A–konjugiert bzw. orthogonal) für *dieselbe* Folge von Unterräumen dar:

$$\begin{aligned} \mathcal{K}_n &= \mathrm{span}\{p_0, .., p_{n-1}\} \\ &= \mathrm{span}\{r_0, .., r_{n-1}\} \end{aligned}$$

Das sieht man bei ihrer iterativen Konstruktion: $\mathcal{K}_n$ wird nach Glg.(3.16) und (3.18) durch den Restvektor $r_n \perp \mathcal{K}_n$ zu $\mathcal{K}_{n+1}$ erweitert. Anschließend wird der nächste Shiftvektor $p_n \in \mathcal{K}_{n+1}$ nach Glg.(3.17) gebildet, der seinerseits A-orthogonal zu $\mathcal{K}_n$ ist.

3.6.3 Lanczos–Basis

Die Restvektoren bilden gerade die Lanczos–Basis bzgl. A, ausgehend von r_0:

$$\begin{aligned} A r_n &= A p_n - \tilde{\beta}_{n-1} A p_{n-1} \\ &= \tilde{\alpha}_n^{-1} (r_n - r_{n+1}) - \frac{\tilde{\beta}_{n-1}}{\tilde{\alpha}_{n-1}} (r_{n-1} - r_n) \\ \Rightarrow \quad A \hat{r}_n &= \beta_n \hat{r}_{n+1} + \alpha_n \hat{r}_n + \beta_{n-1} \hat{r}_{n-1} \qquad (3.26) \\ \text{mit} \quad \hat{r}_n &\equiv r_n / \|r_n\| \end{aligned}$$

$$\alpha_n = \tilde{\alpha}_n^{-1} + \tilde{\alpha}_{n-1}^{-1}\tilde{\beta}_{n-1} \tag{3.27}$$

$$\beta_n = -\tilde{\alpha}_n^{-1}\frac{\|r_{n+1}\|}{\|r_n\|} = -\tilde{\alpha}_n^{-1}\tilde{\beta}_n^{1/2} \tag{3.28}$$

Wie man sieht, lassen sich die Elemente der Lanczos–Tridiagonalisierung in einfacher Weise aus den CG–Koeffizienten gewinnen.

3.6.4 Minimum–Eigenschaft

Nach n CG–Iterationen sind wir angelangt bei

$$\begin{aligned} x_n &= x_0 + \sum_{k=0}^{n-1} \tilde{\alpha}_k p_k \\ \text{mit} \quad x_n - x_0 &\in \mathcal{K}_n \\ r_n &\perp \mathcal{K}_n \end{aligned}$$

und das ist in gewisser Weise optimal: jede andere Wahl der Koeffizienten $\tilde{\alpha}_k$ hätte zu einer anderen Näherungslösung geführt:

$$\begin{aligned} x &= x_n + y \\ r &= r_n - Ay \\ \text{mit} \quad y &\in \mathcal{K}_n \\ r_n^\dagger y &= 0 \end{aligned}$$

und eine kurze Rechnung zeigt:

$$\begin{aligned} r^\dagger A^{-1} r &= r_n^\dagger A^{-1} r_n - r_n^\dagger y - y^\dagger r_n + y^\dagger A y \\ &= r_n^\dagger A^{-1} r_n + y^\dagger A y \end{aligned}$$

d.h. x_n minimiert $r^\dagger A^{-1} r$ im Rahmen der ihm zur Verfügung stehenden Shifts $x_n - x_0 \in \mathcal{K}_n$, denn $y^\dagger A y > 0$ für alle $y \neq 0$.

3.6.5 Schranke für den Restvektor

Da die Shift–Vektoren des CG aus dem Krylov–Raum stammen, der durch wiederholte Anwendung von A auf den Startvektor r_0 erzeugt wird, stellt sich der akkumulierte Shift $x_0 \to x_n$ letzten Endes dar als ein Polynom in A (Grad $\leq n-1$), angewandt auf r_0:

$$\begin{aligned} x_n &= x_0 + P_{n-1}(A) r_0 \\ \Rightarrow \quad r_n &= [1 - A P_{n-1}(A)]\, r_0 \\ &\equiv R_n(A)\, r_0 \end{aligned} \tag{3.29}$$

Hier ist $R_n(x)$ ein Polynom vom Grad $\leq n$ mit der Eigenschaft $R_n(0) = 1$. Im folgenden brauchen wir die Spektraldarstellung von A:

$$A = \sum_i u_i \lambda_i u_i^\dagger$$

und gewichtete Normen der Art

$$\begin{aligned} \|r\|_{A^{-1}}^2 &\equiv r^\dagger A^{-1} r \\ &= \sum_i |u_i^\dagger r|^2 \lambda_i^{-1} \end{aligned}$$

Weil der CG den Restvektor in der Norm $\|r_n\|_{A^{-1}}$ minimiert, kann man eine Konvergenzabschätzung bekommen, indem man R_n durch ein geeignetes, explizit bekanntes Polynom $\check{R}_n$ ersetzt:

$$\begin{aligned} \|r_n\|_{A^{-1}} &= \|R_n(A) r_0\|_{A^{-1}} \\ &\leq \|\check{R}_n(A) r_0\|_{A^{-1}} \\ &\leq \max_i |\check{R}_n(\lambda_i)| \; \|r_0\|_{A^{-1}} \end{aligned} \tag{3.30}$$

Wenn über das Spektrum von A nur bekannt ist, dass

$$\lambda_i \in [\lambda_{min}, \lambda_{max}]$$

dann liefert die Theorie der **Chebyshev**–Approximation optimale Polynome $\check{R}_n$ in der Variablen

$$z = \frac{\lambda_{max} + \lambda_{min} - 2\lambda}{\lambda_{max} - \lambda_{min}} \tag{3.31}$$

und zwar

$$\check{R}_n(\lambda) = \frac{T_n(z)}{T_n(a)}. \tag{3.32}$$

Hier treten die Chebyshev–Polynome $T_n(x)$ auf, definiert durch

$$\begin{aligned} x \in [-1, 1]: &\quad x = \cos\varphi \quad T_n(x) = \cos n\varphi \\ x \geq 1: &\quad x = \cosh\alpha \quad T_n(x) = \cosh n\alpha \end{aligned}$$

Die Eigenwerte λ_i werden von Glg.(3.31) nach $z_i \in [-1, 1]$ abgebildet, wo $|T_n(z_i)| \leq 1$. Der Punkt $\lambda = 0$ geht nach $z = a$:

$$\begin{aligned} a &\equiv \frac{\kappa + 1}{\kappa - 1} > 1 \\ \text{mit} \quad \kappa &= \frac{\lambda_{max}}{\lambda_{min}} \end{aligned} \tag{3.33}$$

Wir führen einen Parameter γ ein:

$$\begin{aligned} \cosh\gamma &\equiv a = \frac{\kappa + 1}{\kappa - 1} \\ \Rightarrow \quad T_n(a) &= \cosh n\gamma \end{aligned}$$

Damit schätzt man ab:

$$|\check{R}_n(\lambda_i)| \leq \frac{1}{\cosh \gamma n} \tag{3.34}$$

$$\Rightarrow \quad \frac{\|r_n\|_{A^{-1}}}{\|r_0\|_{A^{-1}}} \leq \frac{1}{\cosh \gamma n} \tag{3.35}$$

Für die Konvergenz von $\|r_n\|$ selber folgt:

$$\begin{aligned} \lambda_{min}^{1/2} \|r\|_{A^{-1}} &\leq \|r\| \leq \lambda_{max}^{1/2} \|r\|_{A^{-1}} \\ \Rightarrow \quad \frac{\|r_n\|}{\|r_0\|} &\leq \frac{\sqrt{\kappa}}{\cosh \gamma n} \end{aligned} \tag{3.36}$$

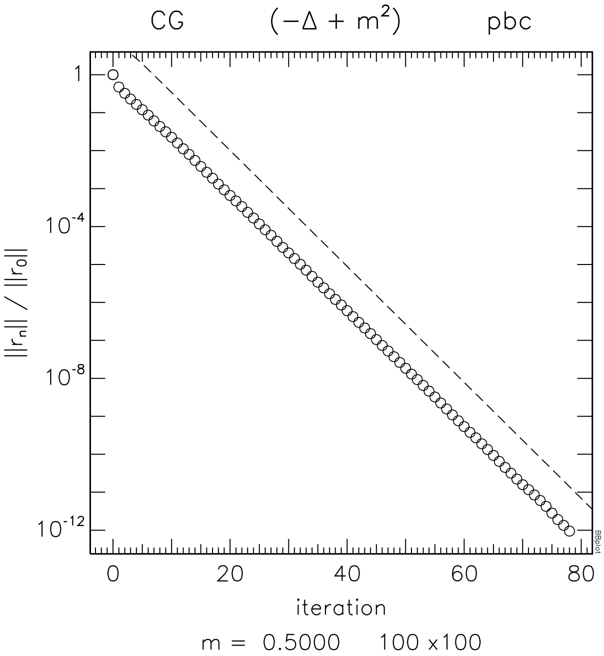
und für die Differenz zur exakten Lösung $x_\infty = A^{-1} b$:

$$\begin{aligned} x_\infty - x &= A^{-1} r \\ \Rightarrow \quad \|x_\infty - x\| &= \|r\|_{A^{-2}} \\ \Rightarrow \quad \lambda_{max}^{-1/2} \|r\|_{A^{-1}} &\leq \|x_\infty - x\| \leq \lambda_{min}^{-1/2} \|r\|_{A^{-1}} \\ \Rightarrow \quad \frac{\|x_\infty - x_n\|}{\|x_\infty - x_0\|} &\leq \frac{\sqrt{\kappa}}{\cosh \gamma n} \end{aligned} \tag{3.37}$$

In allen Fällen konvergiert die Schranke asymptotisch $\sim e^{-\gamma n}$. Den Zusammenhang zwischen **Konditionszahl** κ und **Konvergenzrate** γ kann man auch schreiben als

$$\tanh \frac{\gamma}{2} = \kappa^{-1/2} = \sqrt{\frac{\lambda_{min}}{\lambda_{max}}} \tag{3.38}$$

und sieht so besonders gut, wie sich mit wachsender Konditionszahl die Konvergenz verlangsamt.
Beispiele für die Konvergenz des CG (Laplace–Operator mit verschiedenen Randbedingungen) finden sich in Abb. 1.



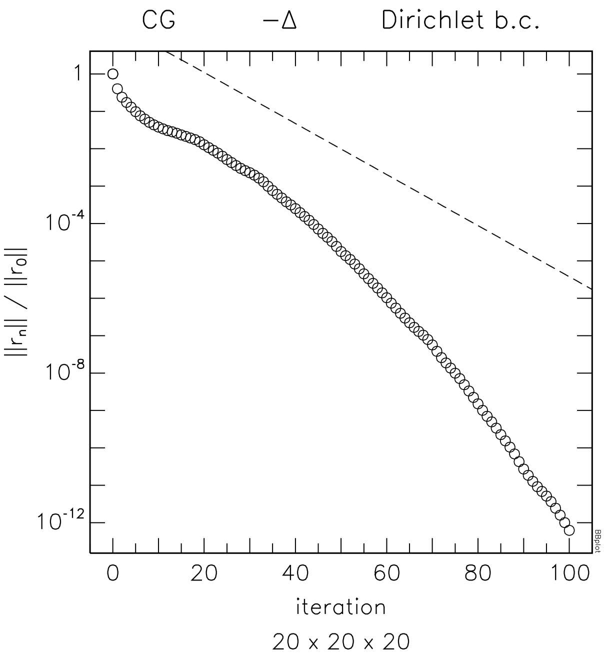


Abbildung 1: Konvergenz des CG. Die gestrichelte Linie stellt die Schranke (3.36) dar.