

**Erstes Beispielproblem:** die *diskretisierte Poisson-Gleichung*

$$\mathbf{A}_1 \mathbf{x} = \mathbf{b}, \quad \mathbf{A}_1 \in \mathbb{R}^{m \times m}, \quad m := (n-1)^2, \quad nh = 1,$$

wobei

$$\mathbf{A}_1 = \frac{1}{h^2} \begin{pmatrix} \mathbf{T} & -\mathbf{I} & & & \\ -\mathbf{I} & \mathbf{T} & -\mathbf{I} & & \emptyset \\ & \cdots & \cdots & \cdots & \\ \emptyset & & -\mathbf{I} & \mathbf{T} & -\mathbf{I} \\ & & & -\mathbf{I} & \mathbf{T} \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \emptyset \\ & \cdots & \cdots & \cdots & \\ \emptyset & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}.$$

$\mathbf{T} \in \mathbb{R}^{(n-1) \times (n-1)}$  und  $\mathbf{I}$  ist die  $(n-1) \times (n-1)$ -Identitätsmatrix.

## Zweites Beispielproblem:

die *diskretisierte Konvektions-Diffusionsgleichung*

$$\mathbf{A}_2 \mathbf{x} = \mathbf{b}, \quad \mathbf{A}_2 \in \mathbb{R}^{m \times m}, \quad m := (n-1)^2,$$

wobei

$$\mathbf{A}_2 = \frac{1}{h^2} \begin{pmatrix} \mathbf{T} & -\tilde{n}\mathbf{I} & & & \\ -\tilde{s}\mathbf{I} & \mathbf{T} & -\tilde{n}\mathbf{I} & & \emptyset \\ & \cdots & \cdots & \cdots & \\ \emptyset & & -\tilde{s}\mathbf{I} & \mathbf{T} & -\tilde{n}\mathbf{I} \\ & & & -\tilde{s}\mathbf{I} & \mathbf{T} \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} \tilde{z} & -\tilde{o} & & & \\ -\tilde{w} & \tilde{z} & -\tilde{o} & & \emptyset \\ & \cdots & \cdots & \cdots & \\ \emptyset & & -\tilde{w} & \tilde{z} & -\tilde{o} \\ & & & -\tilde{w} & \tilde{z} \end{pmatrix},$$

mit

$$\begin{aligned} \tilde{n} &= \varepsilon - h\hat{s}^-, & \tilde{s} &= \varepsilon + h\hat{s}^+, \\ \tilde{o} &= \varepsilon - h\hat{c}^-, & \tilde{w} &= \varepsilon + h\hat{c}^+, \\ \tilde{z} &= \tilde{n} + \tilde{s} + \tilde{o} + \tilde{w} = 4\varepsilon + h(|\hat{c}| + |\hat{s}|), \end{aligned}$$

und

$$\begin{aligned} \hat{c} &:= \cos(\beta), & \hat{c}^+ &:= \max(\hat{c}, 0), & \hat{c}^- &:= \min(\hat{c}, 0), \\ \hat{s} &:= \sin(\beta), & \hat{s}^+ &:= \max(\hat{s}, 0), & \hat{s}^- &:= \min(\hat{s}, 0). \end{aligned}$$

## 13.2 Eigenschaften von Steifigkeitsmatrizen

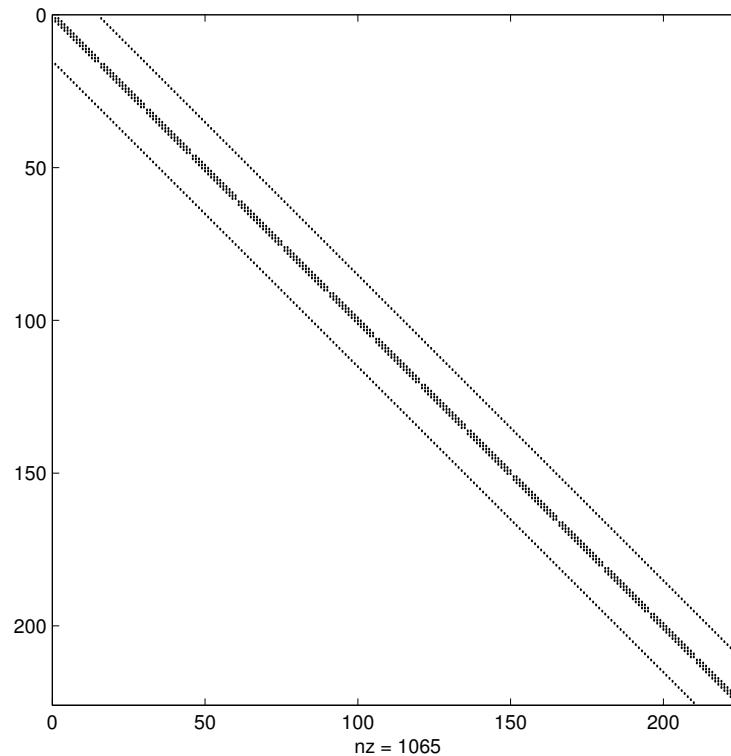
---

### Hochdimensionalität:

Anzahl der Unbekannten ist bei  $d$  Ortsvariablen proportional zu  $h^{-d}$ .

### Dünnbesetztheit:

Steifigkeitsmatrizen sind immer dünnbesetzt, d. h. nur eine gleichmäßig beschränkte Anzahl von Einträgen pro Zeile ist von Null verschieden.



### **Blockstruktur:**

Bei Diskretisierungen auf regelmäßigen Rechteckgittern haben die sich ergebenden Steifigkeitsmatrizen oft eine regelmäßige Blockstruktur.

### **Schlechte Kondition:**

Die Konditionszahl von Steifigkeitsmatrizen nimmt oft rasch zu, wenn die Schrittweite des Gitters kleiner wird.

### **Symmetrie, Positiv-Definitheit:**

Für ein symmetrisches elliptisches Randwertproblem ist die entsprechende Steifigkeitsmatrix oft symmetrisch positiv definit.

### **Diagonaldominanz:**

Die Matrizen  $A_1$  und  $A_2$  sind beide irreduzibel diagonaldominant. Die durch Finite-Elemente-Methoden und Finite-Volumen-Verfahren erzeugten Steifigkeitsmatrizen sind im allgemeinen *nicht* irreduzibel diagonaldominant.

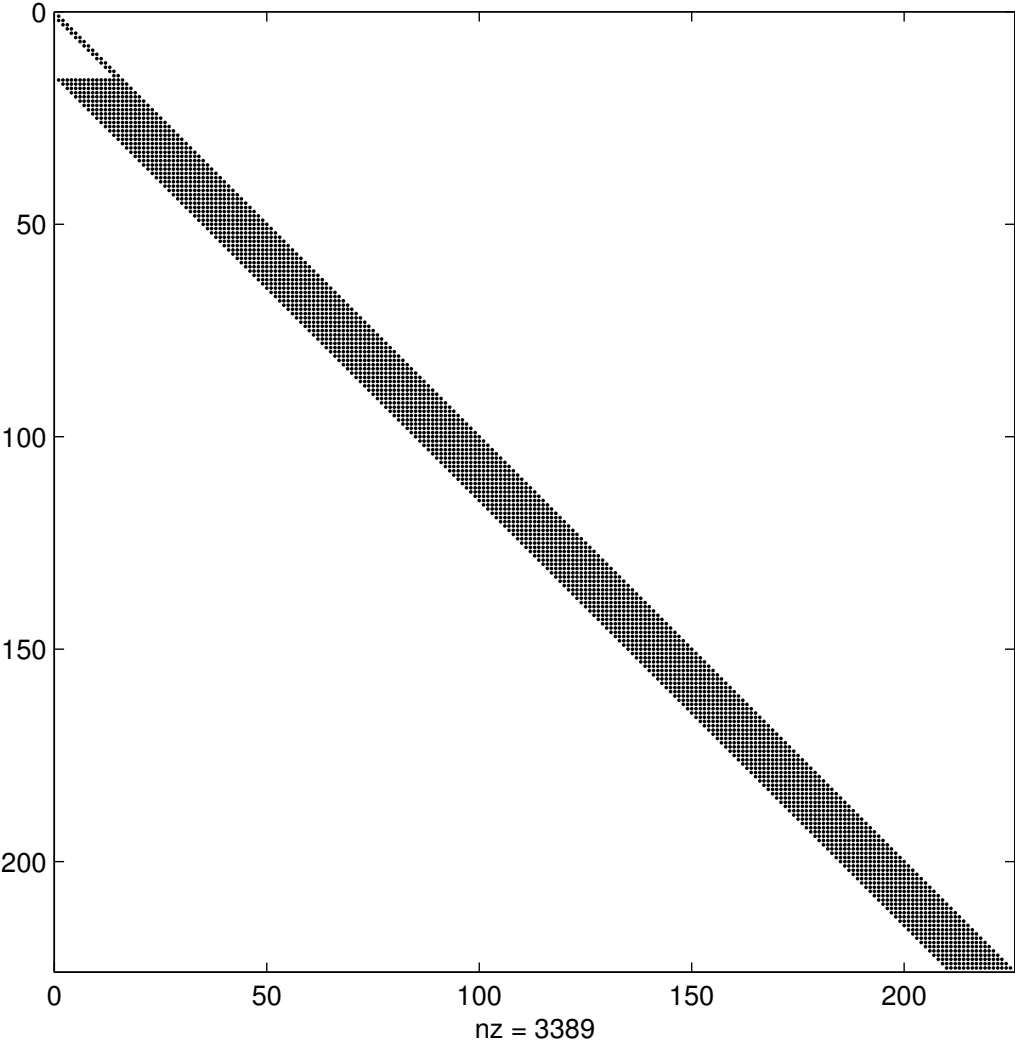
**Bemerkung 13.2.** Als direkter Löser für ein Gleichungssystem mit der Matrix  $A_1$  würde sich die Cholesky-Zerlegung  $A_1 = LDL^T$  anbieten.

Nichtnulleinträge der Matrizen  $A_1$  und  $L$ :

$h$	1/16	1/32	1/64	1/128	
$\text{nz}(A_1)$	1065	4681	19593	80137	$\approx 5h^{-2}$
$\text{nz}(L)$	3389	29821	250109	2048509	$\approx h^{-3}$

Das Verhalten  $\text{nz}(L) \approx h^{-3}$  gegenüber  $\text{nz}(A_1) \approx 5h^{-2}$  zeigt den ungünstigen Effekt des „fill-in“.

Nichtnullelemente von  $\mathbf{L}$  bei der Cholesky-Zerlegung der Matrix  $\mathbf{A}_1$ :



Aufgabe:

Für  $\mathbf{A} \in \mathbb{R}^{n \times n}$  (nichtsingulär) und  $\mathbf{b} \in \mathbb{R}^n$  ist das Gleichungssystem

$$\mathbf{Ax} = \mathbf{b}$$

zu lösen. Wir nehmen an, daß  $n$  groß ( $n > 10000$ ) und  $\mathbf{A}$  dünnbesetzt ist.

Typische Beispiele sind die Matrizen  $\mathbf{A}_1$  und  $\mathbf{A}_2$ .

Einfacher Ansatz: das Gleichungssystem als *Fixpunktgleichung*

$$\mathbf{x} = \mathbf{x} + \mathbf{C}(\mathbf{b} - \mathbf{A}\mathbf{x}) =: \Phi(\mathbf{x}),$$

wobei  $\mathbf{C} \in \mathbb{R}^{n \times n}$  eine geeignet zu wählende nichtsinguläre Matrix ist.

Dies führt auf die Fixpunktiteration

$$\begin{aligned} \mathbf{x}^{k+1} &= \Phi(\mathbf{x}^k) = \mathbf{x}^k + \mathbf{C}(\mathbf{b} - \mathbf{A}\mathbf{x}^k) \\ &= (\mathbf{I} - \mathbf{C}\mathbf{A})\mathbf{x}^k + \mathbf{C}\mathbf{b}, \quad k = 0, 1, 2, \dots \end{aligned}$$

Für den Fehler  $\mathbf{e}^k := \mathbf{x}^k - \mathbf{x}^*$  ergibt sich

$$\mathbf{e}^{k+1} = \mathbf{x}^{k+1} - \mathbf{x}^* = \Phi(\mathbf{x}^k) - \Phi(\mathbf{x}^*) = (\mathbf{I} - \mathbf{C}\mathbf{A})\mathbf{e}^k,$$

also

$$\mathbf{e}^k = (\mathbf{I} - \mathbf{C}\mathbf{A})^k \mathbf{e}^0, \quad k = 0, 1, 2, \dots$$



**Satz 13.3.** Dieses Verfahren konvergiert für jeden Startwert  $\mathbf{x}^0$  gegen die Lösung  $\mathbf{x}^*$  des Gleichungssystems genau dann, wenn

$$\rho(\mathbf{I} - \mathbf{CA}) < 1$$

gilt, wobei  $\rho(\mathbf{I} - \mathbf{CA})$  der Spektralradius, also der betragsmäßig größte Eigenwert, von  $\mathbf{I} - \mathbf{CA}$  ist.

**Folgerung 13.5.** Für eine beliebige Vektornorm  $\|\cdot\|$  mit zugehöriger Operatornorm gilt:

$$\|\mathbf{x}^k - \mathbf{x}^*\| \leq \|\mathbf{I} - \mathbf{CA}\|^k \|\mathbf{x}^0 - \mathbf{x}^*\|, \quad k = 0, 1, 2, \dots$$

Die Konvergenz ist folglich für jeden Startwert  $\mathbf{x}^0$  gesichert, falls für *irgendeine* Norm die Bedingung

$$\|\mathbf{I} - \mathbf{CA}\| < 1$$

erfüllt ist.

Die Größe  $\|\mathbf{I} - \mathbf{CA}\|$  heißt die *Kontraktionszahl* der Fixpunktiteration.

$\rho(\mathbf{I} - \mathbf{CA})$  ist ein sinnvolles *Maß für die Konvergenzgeschwindigkeit*. Je kleiner  $\rho(\mathbf{I} - \mathbf{CA})$  ist, desto schneller wird der Fehler reduziert. Es gilt:

$$\sigma_k = \left( \frac{\|\mathbf{e}^k\|}{\|\mathbf{e}^0\|} \right)^{\frac{1}{k}} \rightarrow |\lambda_1| = \rho(\mathbf{I} - \mathbf{CA}) \quad \text{für } k \rightarrow \infty.$$

Für eine hinreichend große Anzahl  $k$  von ausgeführten Iterationsschritten ist  $\rho(\mathbf{I} - \mathbf{CA})$  etwa die mittlere Fehlerreduktion  $\sigma_k$ .

Um den Anfangsfehler  $\|\mathbf{e}^0\|$  um den Faktor  $\frac{1}{e}$  zu reduzieren, sind  $k = (-\ln \sigma_k)^{-1}$  Iterationsschritte nötig.

Die sogenannte *asymptotische Konvergenzrate*

$$-\ln(\rho(\mathbf{I} - \mathbf{CA}))$$

Ist ein sinnvolles Maß für die asymptotische Konvergenzgeschwindigkeit.

Es geht um folgenden Balanceakt:

Zu  $A \in \mathbb{R}^{n \times n}$  wähle  $C \in \mathbb{R}^{n \times n}$  so, daß

- $A^{-1}$  durch  $C$  genügend gut in dem Sinne approximiert wird, daß  $\rho(I - CA)$  möglichst klein ist,
- die Operation

$$y \mapsto Cy$$

mit möglichst geringem Aufwand durchführbar ist.

*Beachte:*

Bei den meisten iterativen Verfahren wird die Operation  $y \mapsto Cy$  durchgeführt, ohne daß die Matrix  $C$  explizit berechnet wird.

## Komplexität eines iterativen Verfahrens

---

Um unterschiedliche Verfahren miteinander vergleichen zu können, gehen wir davon aus, daß

- eine Klasse von Gleichungssystemen  $\mathbf{Ax} = \mathbf{b}$  vorliegt (z.B.  $\mathbf{A}_1\mathbf{x} = \mathbf{b}$  wie oben mit  $n = \frac{1}{h} \in \mathbb{N}$  beliebig),
- vorgegeben ist, mit welchem Faktor  $R$  ein (beliebiger) Startfehler reduziert werden soll ( $\|\mathbf{e}^k\| \leq \frac{\|\mathbf{e}^0\|}{R}$ ).

*Komplexität eines iterativen Verfahrens:* die Größenordnung der Anzahl arithmetischer Operationen, die benötigt werden, um für das vorliegende Problem (aus der Problemklasse) eine Fehlerreduktion um den Faktor  $R$  zu bewirken.

Liegt dem Gleichungssystem eine Diskretisierung mit Gitterweite  $h$  und einem Diskretisierungsfehler von der Ordnung  $\mathcal{O}(h^\ell)$  zugrunde, wäre  $R^{-1} \approx h^\ell$  sinnvoll.

## 13.3.2 Das Jacobi-Verfahren

---

Beim *Jacobi-* oder *Gesamtschrittverfahren*:

$$C = (\text{diag}(A))^{-1}.$$

Die Iterationsvorschrift lautet

$$\mathbf{x}^{k+1} = (\mathbf{I} - \mathbf{D}^{-1}\mathbf{A})\mathbf{x}^k + \mathbf{D}^{-1}\mathbf{b}.$$

In Komponentenschreibweise ergibt sich:

### **Algorithmus 13.7 (Jacobi-Verfahren).**

Gegeben: Startvektor  $\mathbf{x}^0 \in \mathbb{R}^n$ . Für  $k = 0, 1, \dots$  berechne:

$$a_{i,i}x_i^{k+1} = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{i,j}x_j^k + b_i, \quad i = 1, 2, \dots, n.$$

Beim Jacobi-Verfahren wird die *i-te Gleichung nach der i-ten Unbekannten  $x_i$  aufgelöst*, wobei für die übrigen Unbekannten ( $x_j, j \neq i$ ) Werte aus dem vorherigen Iterationsschritt verwendet werden.

**Rechenaufwand.** Der Rechenaufwand *pro Iterationsschritt* ist beim Jacobi-Verfahren angewandt auf eine dünnbesetzte Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  vergleichbar mit einer Matrix-Vektor-Multiplikation  $\mathbf{A}\mathbf{x}$ , beansprucht also  $\mathcal{O}(n)$  Operationen.

## Konvergenz

**Satz 13.9.** Für das Jacobi-Verfahren gelten folgende Konvergenzkriterien:

- Falls sowohl  $\mathbf{A}$  als auch  $2\mathbf{D} - \mathbf{A}$  symmetrisch positiv definit sind, folgt

$$\rho(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}) < 1.$$

- Falls  $\mathbf{A}$  irreduzibel diagonaldominant ist, gilt

$$\rho(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}) \leq \|\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}\|_{\infty} < 1.$$

## Beispiel 13.10.

---

Das Diffusionsproblem mit der Matrix  $\mathbf{A}_1$ . Es gilt

$$\begin{aligned}\rho(\mathbf{I} - \mathbf{CA}) &= \rho(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}_1) \\ &= \max\left\{ \left| 1 - \frac{1}{4}h^2\lambda \right| \mid \lambda \text{ Eigenwert von } \mathbf{A}_1 \right\} \\ &= 1 - 2 \sin^2\left(\frac{1}{2}\pi h\right) = \cos(\pi h) \approx 1 - \frac{1}{2}\pi^2 h^2.\end{aligned}$$

Für die asymptotische Konvergenzrate gilt somit

$$-\ln(\rho(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}_1)) \approx -\ln\left(1 - \frac{1}{2}\pi^2 h^2\right) \approx \frac{1}{2}\pi^2 h^2.$$

Um einen Startfehler um einen Faktor  $R$  zu reduzieren, sind etwa

$$K = \frac{-\ln R}{\ln \rho(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}_1)} \approx \frac{2}{\pi^2 h^2} \ln R$$

Iterationsschritte erforderlich.

**Beachte:** Die geschätzte Anzahl der notwendigen Iterationsschritte wächst in Abhängigkeit der Schrittweite  $h$  wie die Konditionszahlen der betreffenden Matrizen.

Sei  $\#$  die Anzahl von Iterationsschritten, die zur Reduktion des Startfehlers um einen Faktor  $R = 10^3$  benötigt werden, und  $K$  die theoretische Schätzung von  $\#$  aus:

$$K = \frac{-\ln 10^3}{\ln(\cos \pi h)} \approx \frac{2}{\pi^2 h^2} \ln 10^3.$$

### Ergebnisse:

$h$	1/40	1/80	1/160	1/320
$\#$	2092	8345	33332	133227
$K$	2237	8956	35833	143338

**Komplexität.** Da  $m \sim h^{-2}$ , folgt, daß  $K \sim m$  Iterationsschritte benötigt werden. Da jeder Schritt einen zu  $m$  proportionalen Aufwand erfordert, ergibt sich, daß die Komplexität des Jacobi-Verfahrens (für diese Problemstellung) etwa  $cm^2$  ist.



## Beispiel 13.11.

---

Das Konvektions-Diffusionsproblem mit der Matrix  $\mathbf{A}_2$ .

Wir führen für  $\beta = \frac{\pi}{6}$  ein Experiment wie in Beispiel 13.10 durch.

**Ergebnisse (#):**

$h$	1/40	1/80	1/160	1/320
$\varepsilon = 1$	2155	8611	34415	137597
$\varepsilon = 10^{-2}$	194	587	1967	7099
$\varepsilon = 10^{-4}$	75	148	293	597

Für den Fall  $\varepsilon = 1$  (dominante Diffusion) gilt  $\kappa_2(\mathbf{A}_2) \sim \frac{1}{h^2}$  und für  $\varepsilon = 10^{-4}$  (dominante Konvektion) gilt  $\kappa_2(\mathbf{A}_2) \sim \frac{1}{h}$ .

Diese Beziehungen entsprechen den Wachstumsfaktoren 4 und 2 in den Zeilen für  $\varepsilon = 1$  bzw für  $\varepsilon = 10^{-4}$ .

### 13.3.3 Das Gauß-Seidel-Verfahren

Zerlegung:

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}.$$

Mit  $\mathbf{C} := (\mathbf{D} - \mathbf{L})^{-1}$  ergibt sich:

$$\mathbf{x}^{k+1} = (\mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A})\mathbf{x}^k + (\mathbf{D} - \mathbf{L})^{-1}\mathbf{b}.$$

Äquivalent:

$$(\mathbf{D} - \mathbf{L})\mathbf{x}^{k+1} = \mathbf{U}\mathbf{x}^k + \mathbf{b}.$$

In Komponentenschreibweise heißt dies

#### **Algorithmus 13.12 (Gauß-Seidel-Verfahren).**

Gegeben: Startvektor  $\mathbf{x}^0 \in \mathbb{R}^n$ . Für  $k = 0, 1, \dots$  berechne:

$$x_i^{k+1} = a_{i,i}^{-1} \left( b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{k+1} - \sum_{j=i+1}^n a_{i,j} x_j^k \right), \quad i = 1, 2, \dots, n.$$

Bei der Berechnung der  $i$ -ten Komponente von  $\mathbf{x}^{k+1}$  verwendet man *die bereits vorher berechneten Komponenten der neuen Annäherung.*

**Rechenaufwand.** Für eine dünnbesetzte Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  ist der Rechenaufwand pro Iterationsschritt bei der Gauß-Seidel-Methode vergleichbar mit dem Aufwand beim Jacobi-Verfahren, beträgt also  $\mathcal{O}(n)$  Operationen.

## Konvergenz

**Satz 13.13.** Für das Gauß-Seidel-Verfahren gelten folgende Konvergenzkriterien:

- Falls  $\mathbf{A}$  symmetrisch positiv definit ist, folgt

$$\rho(\mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A}) < 1.$$

- Falls  $\mathbf{A}$  irreduzibel diagonaldominant ist, gilt

$$\rho(\mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A}) \leq \|\mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A}\|_{\infty} < 1.$$

## Beispiel 13.15.

---

Das Diffusionsproblem mit der Matrix  $\mathbf{A}_1$ . Dafür gilt:

$$\rho(\mathbf{I} - \mathbf{CA}_1) = \rho(\mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A}_1) = (\rho(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}_1))^2$$

Hieraus ergibt sich:

$$\rho(\mathbf{I} - \mathbf{CA}_1) = \cos^2(\pi h) \approx 1 - \pi^2 h^2 .$$

$$-\ln(\rho(\mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A}_1)) \approx -\ln(1 - \pi^2 h^2) \approx \pi^2 h^2 .$$

Um einen Startfehler um einen Faktor  $R$  zu reduzieren sind (asymptotisch) etwa  $K$  Iterationsschritte erforderlich, wobei

$$K = \frac{-\ln R}{\ln(\rho(\mathbf{I} - (\mathbf{D} - \mathbf{L})^{-1}\mathbf{A}_1))} \approx \frac{1}{\pi^2 h^2} \ln R .$$

**Ergebnisse:**

$h$	1/40	1/80	1/160	1/320
$\#$	1056	4193	16706	66694
$K$	1119	4478	17916	71669

## Komplexität

Für das Gauß-Seidel-Verfahren angewandt auf die diskrete Poisson-Gleichung

$$\mathbf{A}_1 \mathbf{x} = \mathbf{b}, \quad \mathbf{A}_1 \in \mathbb{R}^{m \times m}, \quad m = (n - 1)^2, \quad n = \frac{1}{h}$$

ist die Komplexität  $cm^2$  Operationen.

## Beispiel 13.16.

---

Die Resultate des Gauß-Seidel-Verfahrens hängen von der Anordnung der Unbekannten ab.

**Beispiel:** Die diskrete Konvektions-Diffusionsgleichung mit  $\beta = \frac{\pi}{6}$ .  
 $h = \frac{1}{160}$ ,  $R = 10^3$  und  $\mathbf{x}^0$ ,  $\mathbf{b}$  wie in Beispiel 13.11.

Gauß-Seidel-Verfahren mit Standardanordnung:

$\varepsilon$	1	$10^{-2}$	$10^{-4}$
#	17197	856	14

Gitterpunkte in *umgekehrter Reihenfolge*:

$\varepsilon$	1	$10^{-2}$	$10^{-4}$
#	17220	1115	285

Für ein Problem, bei dem die Konvektion dominant ist, ist es für das Gauß-Seidel-Verfahren vorteilhaft, bei der Anordnung der Unbekannten (Gitterpunkte) die Strömungsrichtung zu berücksichtigen.

### 13.3.4. SOR-Verfahren

Der Iterationsschritt beim Gauß-Seidel Verfahren läßt sich auch als

$$x_i^{k+1} = x_i^k - a_{i,i}^{-1} \left( \sum_{j=1}^{i-1} a_{i,j} x_j^{k+1} + \sum_{j=i}^n a_{i,j} x_j^k - b_i \right), \quad i = 1, 2, \dots, n,$$

darstellen.

#### **Algorithmus 13.17 (SOR-Verfahren).**

Gegeben: Startvektor  $\mathbf{x}^0 \in \mathbb{R}^n$ , Parameter  $\omega \in (0, 2)$ .

Für  $k = 0, 1, \dots$  berechne:

$$x_i^{k+1} = x_i^k - \omega a_{i,i}^{-1} \left( \sum_{j=1}^{i-1} a_{i,j} x_j^{k+1} + \sum_{j=i}^n a_{i,j} x_j^k - b_i \right), \quad i = 1, 2, \dots, n.$$

Das SOR-Verfahren hat folgende Matrix-Darstellung:

$$(\mathbf{D} - \omega \mathbf{L}) \mathbf{x}^{k+1} = \mathbf{D} \mathbf{x}^k - \omega (\mathbf{D} - \mathbf{U}) \mathbf{x}^k + \omega \mathbf{b}$$

**Rechenaufwand.** Beim SOR-Verfahren ist der Rechenaufwand pro Iterationsschritt vergleichbar mit dem Aufwand beim Gauß-Seidel-Verfahren, also  $\mathcal{O}(n)$  Operationen für ein dünnbesetztes  $\mathbf{A} \in \mathbb{R}^{n \times n}$ .

## Konvergenz

**Satz 13.18.** Sei  $\mathbf{M}_\omega := \mathbf{I} - \left(\frac{1}{\omega}\mathbf{D} - \mathbf{L}\right)^{-1}\mathbf{A}$  die Iterationsmatrix des SOR-Verfahrens. Es gilt:

- $\rho(\mathbf{M}_\omega) \geq |\omega - 1|$ .
- Für  $\mathbf{A}$  s.p.d. gilt:

$$\rho(\mathbf{M}_\omega) < 1 \quad \text{für alle } \omega \in (0, 2).$$

- Sei  $\mathbf{A}$  irreduzibel diagonaldominant mit  $a_{i,j} \leq 0$  für alle  $i \neq j$  und  $a_{i,i} > 0$  für alle  $i$ . Dann gilt:

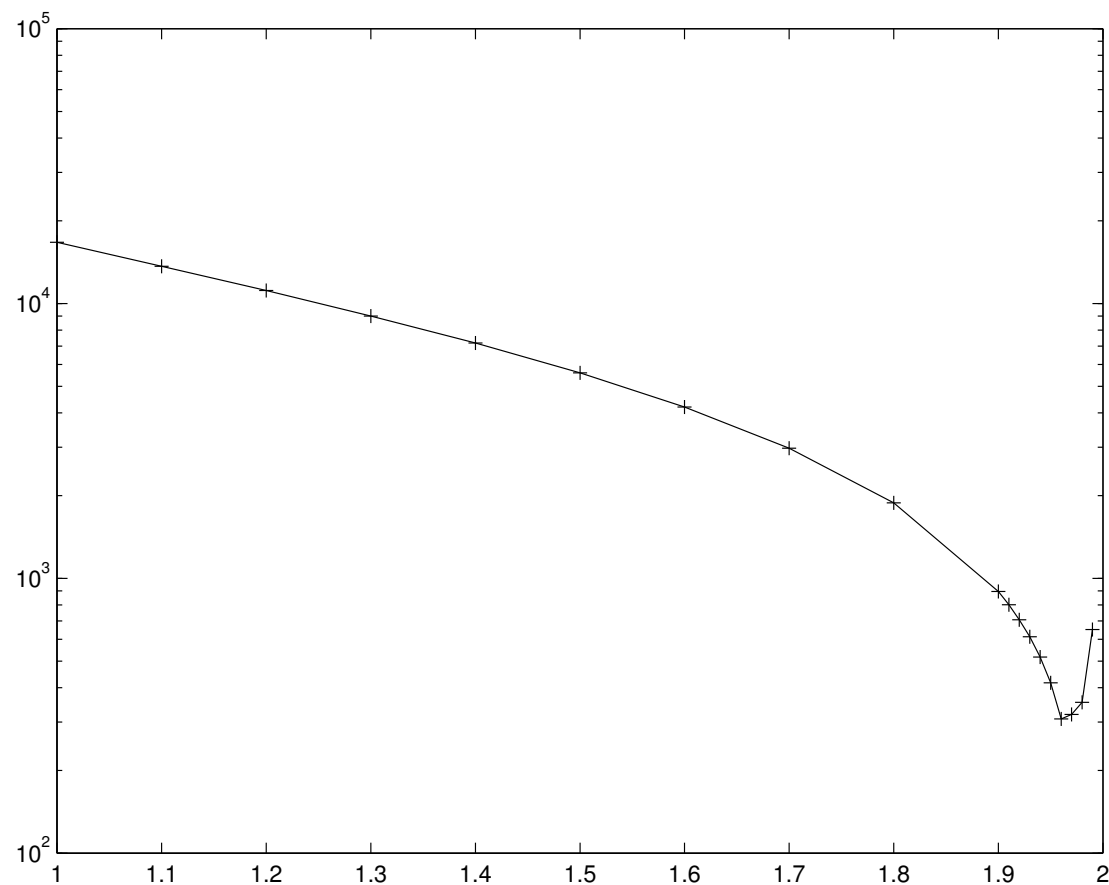
$$\rho(\mathbf{M}_\omega) < 1 \quad \text{für alle } \omega \in (0, 1].$$



## Beispiel 13.19.

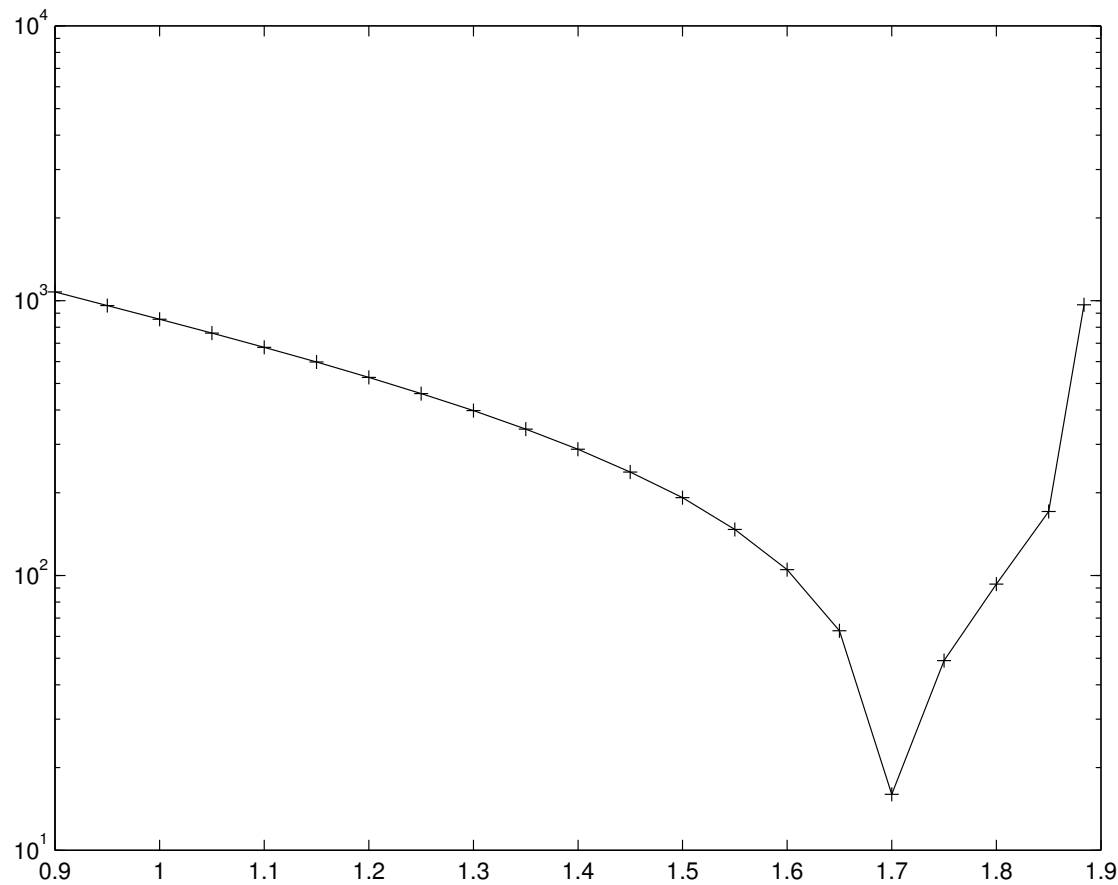
Die Diffusionsgleichung wie in den Beispielen 13.10 und 13.15 für den Fall  $h = \frac{1}{160}$ .

Anzahl der Iterationen in Abhängigkeit von  $\omega$ :



Die Konvektions-Diffusionsgleichung wie in den Beispielen 13.11 und 13.16 für den Fall  $\varepsilon = 10^{-2}$ ,  $h = \frac{1}{160}$ .

Anzahl der Iterationen in Abhängigkeit von  $\omega$ :



Der optimale Wert von  $\omega$  ist stark problemabhängig.  
Für die meisten Probleme ist dieser optimale Wert nicht bekannt.  
Eine Ausnahme bildet die diskretisierte Poisson-Gleichung (Matrix  $\mathbf{A}_1$ ):

### **Satz 13.20.**

Wir betrachten die diskretisierte Poisson-Gleichung  $\mathbf{A}_1 \mathbf{x} = \mathbf{b}$ .

Sei  $\mu := \rho(\mathbf{I} - \mathbf{D}^{-1} \mathbf{A}_1) < 1$  der Spektralradius der Iterationsmatrix des Jacobi-Verfahrens.

Sei  $\mathbf{M}_\omega$  die Iterationsmatrix des SOR-Verfahrens.

Dann ist  $\rho(\mathbf{M}_\omega)$  für den Relaxationsparameter

$$\omega_{\text{opt}} := \frac{2}{1 + \sqrt{1 - \mu^2}} = 1 + \left( \frac{\mu}{1 + \sqrt{1 - \mu^2}} \right)^2$$

optimal und

$$\rho(\mathbf{M}_{\omega_{\text{opt}}}) = \omega_{\text{opt}} - 1.$$

## Komplexität.

Für die diskretisierte Poisson-Gleichung  $\mathbf{A}_1 \mathbf{x} = \mathbf{b}$  ergibt sich beim SOR-Verfahren mit dem *optimalen*  $\omega$ -Wert *eine sehr große Komplexitätsverbesserung im Vergleich zum Jacobi- und zum Gauß-Seidel-Verfahren.*

$$\rho(\mathbf{M}_{\omega_{\text{opt}}}) = \omega_{\text{opt}} - 1 = \left( \frac{\cos(\pi h)}{1 + \sin(\pi h)} \right)^2 = \frac{1 - \sin(\pi h)}{1 + \sin(\pi h)} \approx 1 - 2\pi h.$$

$$K = -\frac{\ln R}{\ln \rho(\mathbf{M}_{\omega_{\text{opt}}})} \approx \frac{1}{2\pi h} \ln R \approx \frac{\ln R}{2\pi} \sqrt{m}.$$

Da der Aufwand pro Iteration proportional zu  $m$  ist, hat dieses Verfahren eine Komplexität (für die vorliegende Problemstellung) von

$$c m^{1.5} \quad \text{Operationen.}$$

## Beispiel 13.21.

---

Poisson-Gleichung  $A_1 \mathbf{x} = \mathbf{b}$  mit  $\mathbf{x}^0$ ,  $\mathbf{b}$  und  $R$  wie in Beispiel 13.10.

Wir verwenden das SOR-Verfahren mit  $\omega = \omega_{\text{opt}}$ .

**Ergebnisse:**

$h$	1/40	1/80	1/160	1/320
#	73	146	292	585
$K$	44	88	176	352

Eine SSOR-Iteration sieht also wie folgt aus:

$$x_i^{k+\frac{1}{2}} = x_i^k - \omega a_{i,i}^{-1} \left( \sum_{j=1}^{i-1} a_{i,j} x_j^{k+\frac{1}{2}} + \sum_{j=i}^n a_{i,j} x_j^k - b_i \right), \quad i = 1, 2, \dots, n$$

$$x_i^{k+1} = x_i^{k+\frac{1}{2}} - \omega a_{i,i}^{-1} \left( \sum_{j=i+1}^n a_{i,j} x_j^{k+1} + \sum_{j=1}^i a_{i,j} x_j^{k+\frac{1}{2}} - b_i \right), \quad n \geq i \geq 1$$

Die Iterationsmatrix dieser Methode ist

$$\mathbf{M}_\omega = \mathbf{I} - \mathbf{C}_\omega \mathbf{A}, \quad \mathbf{C}_\omega := \omega(2 - \omega)(\mathbf{D} - \omega \mathbf{U})^{-1} \mathbf{D} (\mathbf{D} - \omega \mathbf{L})^{-1}.$$

Wenn die Matrix  $\mathbf{A}$  symmetrisch positiv definit ist, ist für  $\omega \in (0, 2)$  die Matrix  $\mathbf{C}_\omega$  in der Iterationsmatrix der SSOR-Methode auch symmetrisch positiv definit.

## 13.4 Die Methode der konjugierten Gradienten

Wenn  $\mathbf{A} \in \mathbb{R}^{n \times n}$  s.p.d. ist, gelten die folgenden zwei grundlegenden Eigenschaften:

- Für  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  definiert  $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{A}} := \mathbf{x}^T \mathbf{A} \mathbf{y}$  ein *Skalarprodukt*.
- Sei

$$f(\mathbf{x}) := \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x}, \quad \mathbf{b}, \mathbf{x} \in \mathbb{R}^n.$$

Dann gilt, daß  $f$  ein eindeutiges Minimum hat und

$$\mathbf{A} \mathbf{x}^* = \mathbf{b} \Leftrightarrow f(\mathbf{x}^*) = \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}).$$

**Lemma 13.22.** Sei  $f$  wie oben. Die Richtung des steilsten Abstiegs von  $f$  an der Stelle  $\mathbf{x}$ , d. h.  $\mathbf{s} \in \mathbb{R}^n$  so, daß die Richtungsableitung

$$\left. \frac{d}{dt} f\left(\mathbf{x} + t \frac{\mathbf{s}}{\|\mathbf{s}\|_2}\right) \right|_{t=0} = (\nabla f(\mathbf{x}))^T \left(\frac{\mathbf{s}}{\|\mathbf{s}\|_2}\right)$$

minimal ist, wird durch  $\mathbf{s} = -\nabla f(\mathbf{x}) = \mathbf{b} - \mathbf{A} \mathbf{x}$  gegeben.

Sei  $U$  ein Unterraum von  $\mathbb{R}^n$ . Es gilt:

$$f(\hat{\mathbf{x}}) = \min_{\mathbf{x} \in U} f(\mathbf{x}) \iff \|\hat{\mathbf{x}} - \mathbf{x}^*\|_{\mathbf{A}} = \min_{\mathbf{x} \in U} \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}}$$

Das folgende Lemma zeigt, wie man eine solche Aufgabe löst.

**Lemma 13.24.** Sei  $U_k$  ein  $k$ -dimensionaler Teilraum von  $\mathbb{R}^n$ , und  $\mathbf{p}^0, \mathbf{p}^1, \dots, \mathbf{p}^{k-1}$  eine  $\mathbf{A}$ -orthogonale Basis dieses Teilraumes:  $\langle \mathbf{p}^i, \mathbf{p}^j \rangle_{\mathbf{A}} = 0$  für  $i \neq j$ . Sei  $\mathbf{v} \in \mathbb{R}^n$ , dann gilt für  $\mathbf{u}^k \in U_k$ :

$$\|\mathbf{u}^k - \mathbf{v}\|_{\mathbf{A}} = \min_{\mathbf{u} \in U_k} \|\mathbf{u} - \mathbf{v}\|_{\mathbf{A}}$$

genau dann, wenn  $\mathbf{u}^k$  die  $\mathbf{A}$ -orthogonale Projektion von  $\mathbf{v}$  auf  $U_k$  ist. Außerdem hat  $\mathbf{u}^k$  die Darstellung

$$\mathbf{u}^k = \sum_{j=0}^{k-1} \frac{\langle \mathbf{v}, \mathbf{p}^j \rangle_{\mathbf{A}}}{\langle \mathbf{p}^j, \mathbf{p}^j \rangle_{\mathbf{A}}} \mathbf{p}^j.$$



Der Einfachheit halber:  $\mathbf{x}^0 = 0$ .

Die folgenden Teilschritte definieren die beim CG-Verfahren erzeugten Näherungen  $\mathbf{x}^1, \mathbf{x}^2, \dots$  der Lösung  $\mathbf{x}^*$ :

$$U_1 := \text{span}\{\mathbf{r}^0\},$$

Für  $k = 1, 2, 3, \dots$ , falls  $\mathbf{r}^{k-1} = \mathbf{b} - \mathbf{A}\mathbf{x}^{k-1} \neq 0$ :

CG<sub>a</sub>: Bestimme  $\mathbf{A}$ -orthogonale Basis  $\mathbf{p}^0, \dots, \mathbf{p}^{k-1}$  von  $U_k$ .

CG<sub>b</sub>: Bestimme  $\mathbf{x}^k \in U_k$ , so daß

$$\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{A}} = \min_{\mathbf{x} \in U_k} \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}}.$$

CG<sub>c</sub>: Erweiterung des Teilraumes:

$$U_{k+1} := \text{span}\{\mathbf{p}^0, \dots, \mathbf{p}^{k-1}, \mathbf{r}^k\}, \text{ wobei } \mathbf{r}^k := \mathbf{b} - \mathbf{A}\mathbf{x}^k.$$

## Erläuterung dieser Teilschritte:

- CG<sub>a</sub>:** Die  $\mathbf{A}$ -orthogonale Basis von  $U_k$  wird benötigt, um die Annäherung  $\mathbf{x}^k$  in Teilschritt CG<sub>b</sub> effizient und stabil zu berechnen.
- CG<sub>b</sub>:**  $\mathbf{x}^k$  ist die (bezüglich  $\|\cdot\|_{\mathbf{A}}$ ) *optimale* Approximation von  $\mathbf{x}^*$  in  $U_k$ .
- CG<sub>c</sub>:**  $U_{k+1} = U_k \oplus \text{span}\{\mathbf{r}^k\}$ :  
Die Erweiterung des Raumes  $U_k$  ist *optimal* in dem Sinne, daß  $\mathbf{r}^k$  an der Stelle  $\mathbf{x}^k$  die Richtung des steilsten Abstiegs von  $f$  ist

## Verfahren der konjugierten Gradienten (CG)

Es ergibt sich der berühmte

### Algorithmus 13.27 (Verfahren der konjugierten Gradienten).

Gegeben:  $\mathbf{A} \in \mathbb{R}^{n \times n}$  s.p.d.,  $\mathbf{b} \in \mathbb{R}^n$ , Startvektor  $\mathbf{x}^0 \in \mathbb{R}^n$ ,  
 $\beta_{-1} := 0$ . Berechne  $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$ . Für  $k = 1, 2, \dots$ , falls  $\mathbf{r}^{k-1} \neq \mathbf{0}$ :

$$\mathbf{p}^{k-1} = \mathbf{r}^{k-1} + \beta_{k-2}\mathbf{p}^{k-2}, \quad \beta_{k-2} = \frac{\langle \mathbf{r}^{k-1}, \mathbf{r}^{k-1} \rangle}{\langle \mathbf{r}^{k-2}, \mathbf{r}^{k-2} \rangle} \quad (k \geq 2),$$

$$\mathbf{x}^k = \mathbf{x}^{k-1} + \alpha_{k-1}\mathbf{p}^{k-1}, \quad \alpha_{k-1} = \frac{\langle \mathbf{r}^{k-1}, \mathbf{r}^{k-1} \rangle}{\langle \mathbf{p}^{k-1}, \mathbf{A}\mathbf{p}^{k-1} \rangle},$$

$$\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha_{k-1}\mathbf{A}\mathbf{p}^{k-1}.$$

Die CG-Methode ist ein *nichtlineares* Verfahren:

$$\mathbf{e}^{k+1} = \psi(\mathbf{e}^k),$$

wobei  $\psi$  eine *nichtlineare* Funktion ist.

**Rechenaufwand.** Im Algorithmus 13.27 werden pro Iterationsschritt nur *eine* Matrix-Vektor-Multiplikation, zwei Skalarprodukte und drei Vektor-Operationen der Form  $\mathbf{x} + \alpha\mathbf{y}$  benötigt. Der Aufwand pro Iterationsschritt ist vergleichbar mit dem des Jacobi- oder Gauß-Seidel-Verfahrens, also  $\mathcal{O}(n)$  Operationen für ein dünnbesetztes  $\mathbf{A} \in \mathbb{R}^{n \times n}$ .

## Konvergenz

**Satz 13.29.** Gegeben sei  $\mathbf{Ax} = \mathbf{b}$  mit  $\mathbf{A} \in \mathbb{R}^{n \times n}$  s.p.d. Sei  $\mathbf{x}^*$  die Lösung von  $\mathbf{Ax} = \mathbf{b}$ . Dann gilt für die durch Algorithmus 13.27 gelieferten Näherungen  $\mathbf{x}^k$

$$\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{A}} \leq 2 \left( \frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^k \|\mathbf{x}^0 - \mathbf{x}^*\|_{\mathbf{A}}, \quad k = 0, 1, 2, \dots$$

## Beispiel 13.30.

---

Das Diffusionsproblem  $\mathbf{A}_1 \mathbf{x} = \mathbf{b}$ .

Für die Konditionszahl der Matrix  $\mathbf{A}_1$  gilt

$$\kappa_2(\mathbf{A}_1) = \frac{\cos^2(\frac{1}{2}\pi h)}{\sin^2(\frac{1}{2}\pi h)} \approx \left(\frac{2}{\pi h}\right)^2,$$

also

$$\hat{\rho} := \frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \approx \frac{\frac{2}{\pi h} - 1}{\frac{2}{\pi h} + 1} = \frac{1 - \frac{1}{2}\pi h}{1 + \frac{1}{2}\pi h} \approx 1 - \pi h.$$

$$K := -\frac{\ln(2R)}{\ln(1 - \pi h)} \approx \frac{1}{\pi h} \ln(2R)$$

**Ergebnisse:**

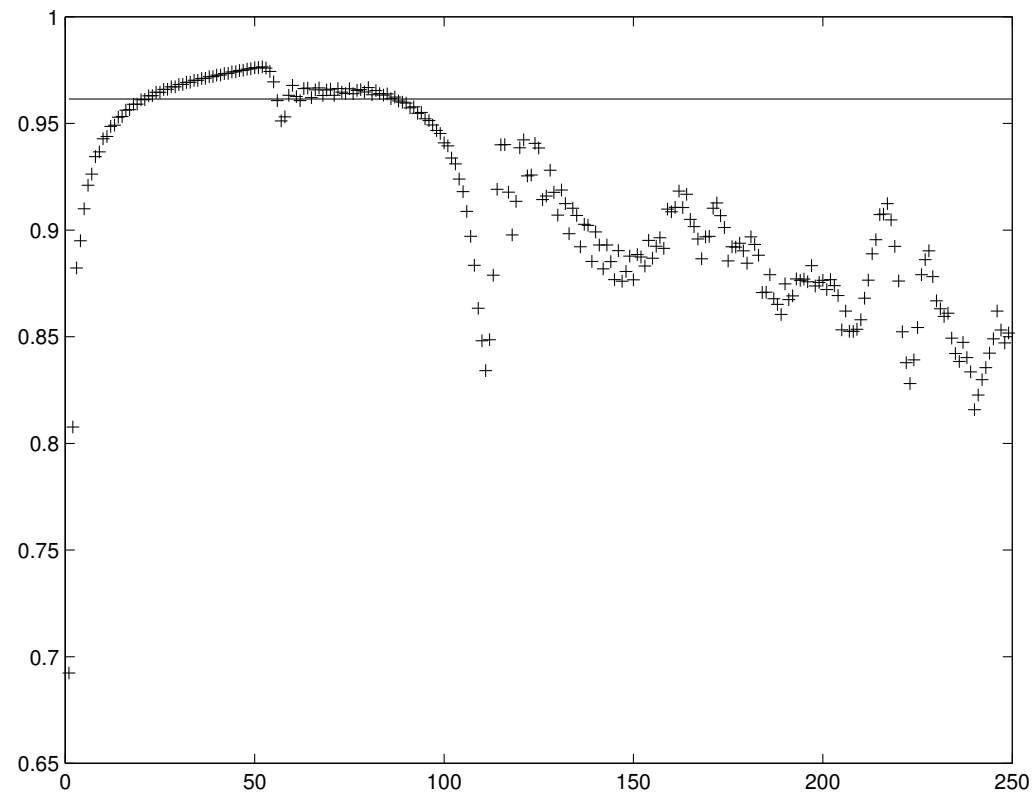
$h$	1/40	1/80	1/160	1/320
$\#$	65	130	262	525

# Superlineare Konvergenz

Für den Fall  $h = \frac{1}{80}$  wird die Fehlerreduktion in der Energie-Norm

$$\rho_k := \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{A}}}{\|\mathbf{x}^{k-1} - \mathbf{x}^*\|_{\mathbf{A}}},$$

in den ersten 250 Iterationsschritten dargestellt.



**Komplexität.** Das CG-Verfahren für die diskrete Poissongleichung:

$$\mathbf{A}_1 \mathbf{x} = \mathbf{b}, \quad \mathbf{A}_1 \in \mathbb{R}^{m \times m}, \quad m = (n - 1)^2, \quad n = \frac{1}{h}.$$

Als Fehlerreduktionsfaktor wird  $R = 10^3$  gewählt.

Die Komplexität des CG-Verfahrens (für diese Problemstellung) ist

$$c m^{1.5} \text{ Operationen,}$$

also von der gleichen Größenordnung wie beim SOR-Verfahren mit optimalem  $\omega$ -Wert.

Ein großer Vorteil der CG-Methode im Vergleich zum SOR-Verfahren ist, daß sie *keinen vom Benutzer zu wählenden Parameter enthält*.

**Idee:** Die Matrix  $\mathbf{A}$  wird mit einer geeigneten zu wählenden symmetrisch positiv definiten Matrix  $\mathbf{W}$  vorkonditioniert.

Da die Matrix  $\mathbf{W}$  symmetrisch positiv definit ist, existiert die Cholesky-Zerlegung

$$\mathbf{W} = \mathbf{L}\mathbf{D}\mathbf{L}^T =: \mathbf{L}_1\mathbf{L}_1^T,$$

Wir betrachten das transformierte System

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}} \quad \text{mit} \quad \tilde{\mathbf{A}} := \mathbf{L}_1^{-1}\mathbf{A}\mathbf{L}_1^{-T}, \quad \tilde{\mathbf{x}} := \mathbf{L}_1^T\mathbf{x}, \quad \tilde{\mathbf{b}} = \mathbf{L}_1^{-1}\mathbf{b} .$$

Die Matrix  $\tilde{\mathbf{A}}$  ist symmetrisch positiv definit.

Der CG-Algorithmus ist also auf dieses Problem anwendbar.



Wenn man den CG-Algorithmus für das transformierte System in die ursprünglichen Variablen  $\mathbf{x} = \mathbf{L}_1^{-T} \tilde{\mathbf{x}}$  umschreibt, ergibt sich folgender

**Algorithmus 13.32 (PCG-Verfahren).**

Gegeben:  $\mathbf{A}, \mathbf{W} \in \mathbb{R}^{n \times n}$  s.p.d.,  $\mathbf{b} \in \mathbb{R}^n$ , Startvektor  $\mathbf{x}^0 \in \mathbb{R}^n$ ,  $\beta_{-1} := 0$ . Berechne  $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0$ ,  $\mathbf{z}^0 = \mathbf{W}^{-1}\mathbf{r}^0$  (löse  $\mathbf{W}\mathbf{z}^0 = \mathbf{r}^0$ ).  
Für  $k = 1, 2, 3, \dots$ , falls  $\mathbf{r}^{k-1} \neq 0$ :

$$\mathbf{p}^{k-1} = \mathbf{z}^{k-1} + \beta_{k-2}\mathbf{p}^{k-2}, \quad \beta_{k-2} = \frac{\langle \mathbf{z}^{k-1}, \mathbf{r}^{k-1} \rangle}{\langle \mathbf{z}^{k-2}, \mathbf{r}^{k-2} \rangle} \quad (k \geq 2),$$

$$\mathbf{x}^k = \mathbf{x}^{k-1} + \alpha_{k-1}\mathbf{p}^{k-1}, \quad \alpha_{k-1} = \frac{\langle \mathbf{z}^{k-1}, \mathbf{r}^{k-1} \rangle}{\langle \mathbf{p}^{k-1}, \mathbf{A}\mathbf{p}^{k-1} \rangle},$$

$$\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha_{k-1}\mathbf{A}\mathbf{p}^{k-1},$$

$$\mathbf{z}^k = \mathbf{W}^{-1}\mathbf{r}^k \quad (\text{löse } \mathbf{W}\mathbf{z}^k = \mathbf{r}^k).$$

**Beachte:** *nur* die Matrizen  $\mathbf{A}$  und  $\mathbf{W}$  treten auf und nicht  $\tilde{\mathbf{A}}$ ,  $\tilde{\mathbf{L}}_1$ .

Die Konvergenz des PCG-Verfahrens wird durch die Konditionszahl

$$\kappa_2(\tilde{\mathbf{A}}) = \kappa_2(\mathbf{L}_1^{-1} \mathbf{A} \mathbf{L}_1^{-T}) = \kappa_2(\mathbf{W}^{-1} \mathbf{A})$$

bedingt.

Um eine gute Effizienz des PCG-Verfahrens zu gewährleisten, ist  $\mathbf{W}$  so zu wählen, daß

- $\kappa_2(\mathbf{W}^{-1} \mathbf{A})$  möglichst klein ist,
- die Lösung von

$$\mathbf{W} \mathbf{z}^k = \mathbf{r}^k$$

mit möglichst geringem Aufwand ( $\mathcal{O}(n)$  Operationen) berechnet werden kann.

Manchmal ist es schon hilfreich,  $\mathbf{W} = \text{diag}(\mathbf{A})$  zu nehmen.

## Unvollständige Cholesky-Zerlegung als Vorkonditionierung

$A$  läßt sich in der Form

$$A = L_1 L_1^T$$

mit einer unteren Dreiecksmatrix  $L_1$  zerlegen (Cholesky).

Um *Dünnbesetztheit* zu erhalten greift man auf eine *unvollständige* Cholesky-Zerlegung zurück. Sei

$$E := \{(i, j) \mid a_{i,j} \neq 0\}$$

Es wird eine *näherungsweise* Faktorisierung konstruiert:

$$A = \tilde{L}_1 \tilde{L}_1^T + R,$$

mit:

- $\tilde{L}_1$  eine untere Dreiecksmatrix,
- $(\tilde{L}_1)_{i,j} = 0$  für  $(i, j) \notin E$ .

Die Konstruktion einer näherungsweise Faktorisierung  $A \approx \tilde{L}_1 \tilde{L}_1^T$  beruht auf der *unvollständigen Durchführung* einer Methode zur Berechnung der vollständigen Cholesky-Zerlegung.

### Algorithmus 13.34 (Unvollständige Cholesky-Methode).

Gegeben:  $\mathbf{A} \in \mathbb{R}^{n \times n}$  s.p.d., Muster  $E$ .

Berechne für  $k = 1, 2, \dots, n$ :

$$\tilde{l}_{k,k} = \left( a_{k,k} - \sum_j' \tilde{l}_{k,j}^2 \right)^{\frac{1}{2}} ;$$

für  $i = k + 1, \dots, n$ : falls  $(i, k) \in E$ ,

$$\tilde{l}_{i,k} = \left( a_{i,k} - \sum_j'' \tilde{l}_{i,j} \tilde{l}_{k,j} \right) / \tilde{l}_{k,k} ;$$

Hierbei sind die Summen nur über Indizes aus dem Muster zu führen:

$$\sum_j' = \sum_{j=1, (k,j) \in E}^{k-1}, \quad \sum_j'' = \sum_{j=1, (i,j) \in E, (k,j) \in E}^{k-1} .$$

Im PCG-Algorithmus:  $\mathbf{W} = \tilde{\mathbf{L}}_1 \tilde{\mathbf{L}}_1^T$ .

Für viele Probleme:  $\kappa_2(\mathbf{W}^{-1} \mathbf{A}) = \kappa(\tilde{\mathbf{L}}_1^{-1} \mathbf{A} \tilde{\mathbf{L}}_1^{-T}) \ll \kappa(\mathbf{A})$ .

Die Systeme  $\tilde{\mathbf{L}}_1 \tilde{\mathbf{L}}_1^T \mathbf{z}^k = \mathbf{r}^k$  sind einfach lösbar.

### Algorithmus 13.36 (Modifiziertes unvollständiges Cholesky)

Gegeben:  $A \in \mathbb{R}^{n \times n}$  s.p.d., Muster  $E$ . Berechne für  $k = 1, 2, \dots, n$ :

$$\hat{l}_{k,k} = \left( a_{k,k} - \sum_j' \hat{l}_{k,j} \right)^{\frac{1}{2}} ;$$

für  $i = k + 1, \dots, n$ :

falls  $(i, k) \in E$ :

$$\hat{l}_{i,k} = \left( a_{i,k} - \sum_j'' \hat{l}_{i,j} \hat{l}_{k,j} \right) / \hat{l}_{k,k} ;$$

sonst:

$$a_{i,i} = a_{i,i} - \sum_j'' \hat{l}_{i,j} \hat{l}_{k,j} ;$$

Man kann zeigen, daß für  $\hat{L}_1 := (\hat{l}_{i,j})_{1 \leq i,j \leq n}$ :

$$(\hat{L}_1 \hat{L}_1^T)_{i,j} = a_{i,j} \quad \text{für } (i,j) \in E, \quad i \neq j,$$

$$\sum_{j=1}^n (\hat{L}_1 \hat{L}_1^T)_{i,j} = \sum_{j=1}^n a_{i,j} \quad \text{für alle } i.$$

## Beispiel 13.37.

---

Die diskretisierte Poisson-Gleichung.

PCG-Verfahren:

ICCG: CG + unvollständige Cholesky Vorkonditionierung.

MICCG: CG + mod. unvollständige Cholesky Vorkonditionierung.

#: Anzahl der Iterationsschritte, die zur Reduktion des Startfehlers  $\|\mathbf{x}^0 - \mathbf{x}^*\|_2$  um einen Faktor  $R = 10^3$  benötigt werden ( $\mathbf{x}^0 := 0$ ).

**Ergebnisse:**

	$h$	1/40	1/80	1/160	1/320
$\mathbf{W} = \tilde{\mathbf{L}}_1 \tilde{\mathbf{L}}_1^T$ , #		20	40	79	157
$\mathbf{W} = \hat{\mathbf{L}}_1 \hat{\mathbf{L}}_1^T$ , #		8	11	14	20

**Komplexität.** PCG-Verfahren für die diskrete Poissongleichung.  
Als Fehlerreduktionsfaktor wird  $R = 10^3$  gewählt.

Die Konvergenz der ICCG-Methode ist schneller als die der Methode ohne Vorkonditionierung. Die Komplexität ist aber für beide Methoden

$$c m^{1.5} \text{ Operationen,}$$

wobei die Konstante  $c$  beim PCG-Verfahren kleiner ist.

Man kann zeigen, daß die MICCG-Methode die Komplexität

$$c m^{1.25} \text{ Operationen}$$

hat.

Sei

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{C}(\mathbf{b} - \mathbf{A}\mathbf{x}^k), \quad \mathbf{A} \text{ symmetrisch positiv definit.}$$

Man kann die Matrix  $\mathbf{C}$  als eine Näherung für  $\mathbf{A}^{-1}$  interpretieren. Deshalb liegt der Ansatz

$$\mathbf{W} = \mathbf{C}^{-1}$$

als Vorkonditionierung für das CG-Verfahren nahe.

Allerdings muß die Matrix  $\mathbf{W}$  symmetrisch positiv definit sein.

Beispiele:

- die Jacobi-Methode, mit  $\mathbf{C}^{-1} = \text{diag}(\mathbf{A})$ .
- die SSOR-Methode.

In PCG muß  $\mathbf{z}^k = \mathbf{W}^{-1}\mathbf{r}^k$  also  $\mathbf{z}^k = \mathbf{C}\mathbf{r}^k$  berechnet werden.

$\mathbf{z}^k$  ist das Resultat einer Iteration des linearen iterativen Verfahrens angewandt auf  $\mathbf{A}\mathbf{z} = \mathbf{r}^k$  mit Startvektor 0:

$$\mathbf{b} = \mathbf{r}^k, \quad \mathbf{x}^0 = 0 \quad \Rightarrow \quad \mathbf{x}^1 = \mathbf{C}\mathbf{r}^k.$$



## Beispiel 13.39.

---

Die diskretisierte Poisson-Gleichung.

Das PCG-Verfahren mit der SSOR-Methode als Vorkonditionierung.  
Als Fehlerreduktionsfaktor wird  $R = 10^3$  gewählt;  $x^0 = 0$ .

**Ergebnisse:**

$h$	1/40	1/80	1/160	1/320
$\omega = 2/(1 + \sin(\pi h))$	1.854	1.924	1.961	1.981
#	11	16	22	32