

## Zwei Modellprobleme

Eindimensionales elliptisches Randwertproblem

$$-u''(x) = f(x), \quad x \in [0, 1]$$

$$u(0) = g_0, \quad u(1) = g_1$$

Zweidimensionales elliptisches Randwertproblem

$$\Omega = [0, 1] \times [0, 1]$$

$$\overbrace{-u_{xx}(x, y) - u_{yy}(x, y)}^{-\Delta u(x, y)} = f(x, y), \quad (x, y) \in \Omega$$

$$u(x, y) = g(x, y), \quad (x, y) \in \partial\Omega$$

# Diskretisierung mittels Finiter Differenzen

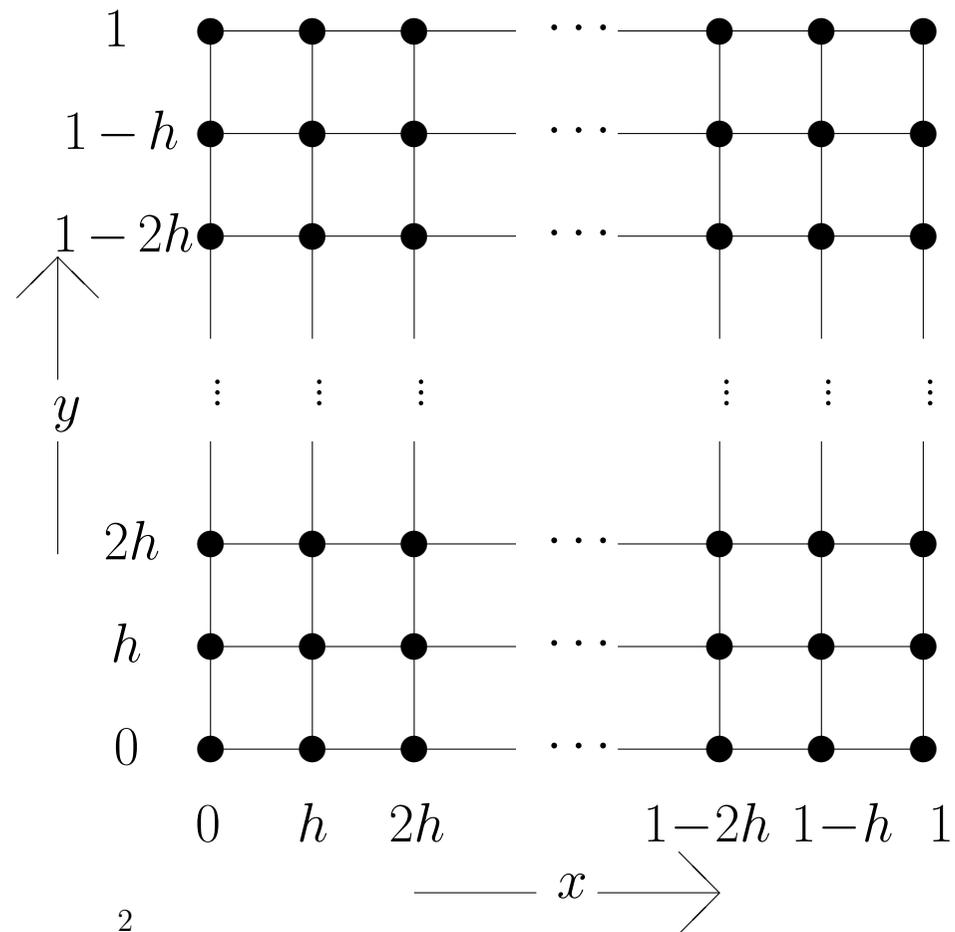
1D-Problem:  $h = \frac{1}{N+1}$ .  $[0, 1] \rightarrow \Omega_h = \{0, h, 2h, \dots, 1-h, 1\}$



2D-Problem:

$\Omega = [0, 1]^2 \rightarrow \Omega_h =$

$\{(k, l)h : k, l = 0, \dots, N+1\}$



# Diskretisierung mittels Finiter Differenzen

1D Randwertproblem

Differentialgleichung

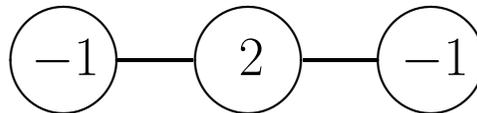
$$-y''(x) = f(x), \quad y(0) = g_0, \quad y(1) = g_1$$

↓

Differenzengleichung

$$\frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} = f_i, \quad u_0 = g_0, \quad u_{N+1} = g_1, \quad \text{wobei } u_i \equiv u(ih) \quad \forall i$$

Differenzenstern



Lineares Gleichungssystem

$$T_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & \\ -1 & \cdots & \cdots & \\ & \cdots & \cdots & -1 \\ & & -1 & 2 \end{pmatrix} \longrightarrow T_h u = f.$$

# Diskretisierung mittels Finiter Differenzen

2D Randwertproblem

Differentialgleichung

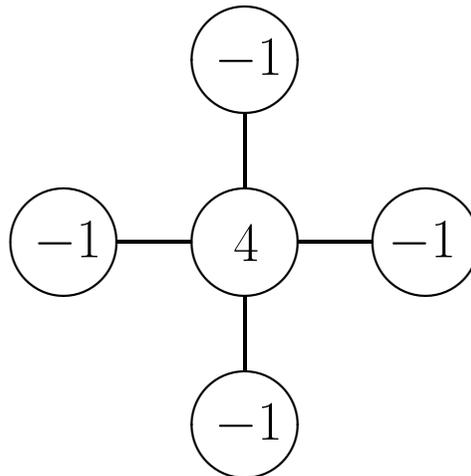
$$-\Delta y(x, y) = f(x, y), \text{ in } \Omega = [0, 1]^2, \quad y = g \text{ auf } \partial\Omega$$

↓

Differenzengleichung

$$\frac{-u_{i-1,j} - u_{i,j-1} + 4u_{ij} - u_{i+1,j} - u_{i,j+1}}{h^2} = f_{ij}$$

Differenzenstern



Lineares Gleichungssystem

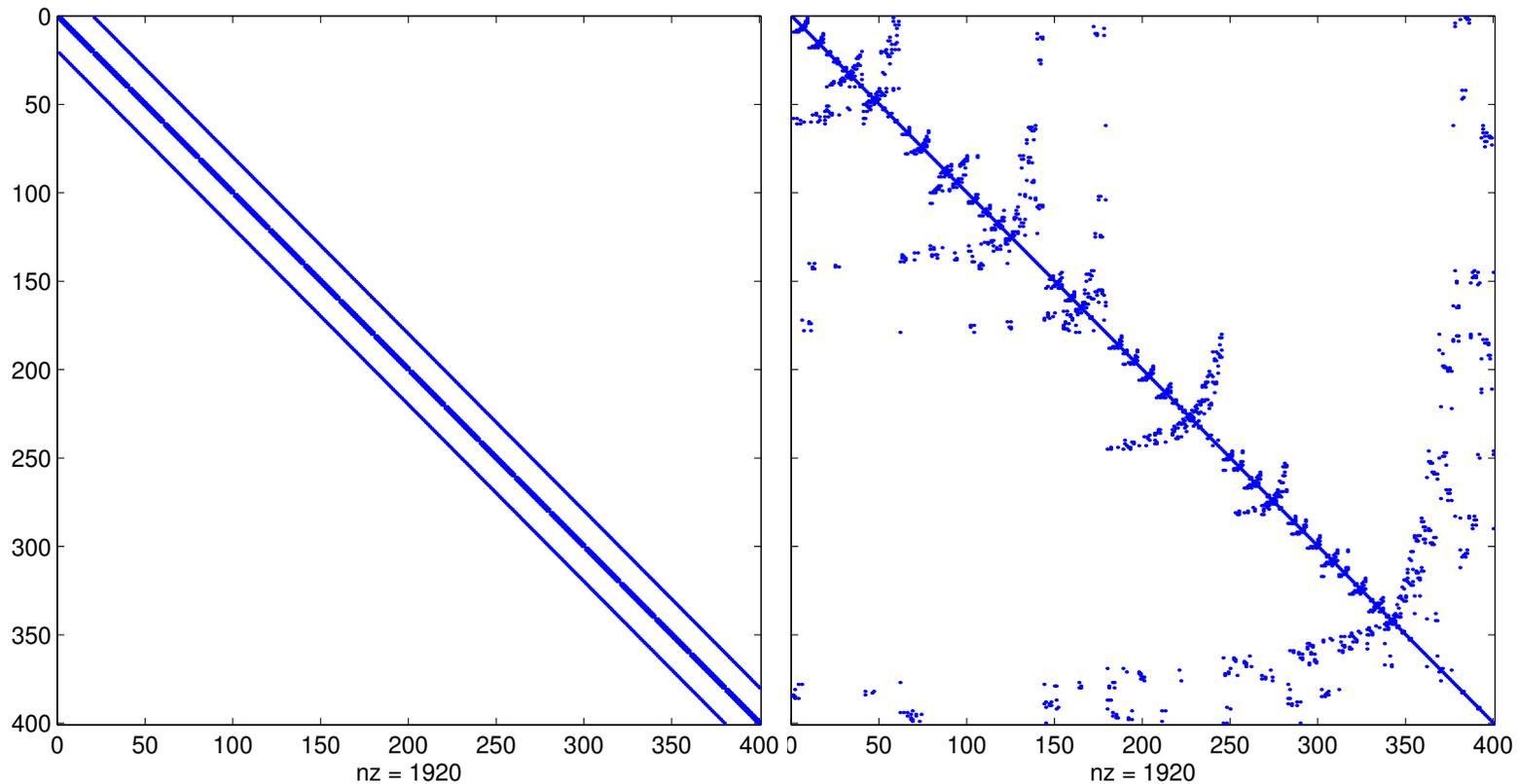
$$A_h u = f$$

# Elementare Lösungstechniken — Direkter Löser

1D/2D RWP: Matrix schwach besetzt, symmetrisch positiv definit

Angepasster direkter Löser  $A = LL^T$ , Cholesky-Zerlegung

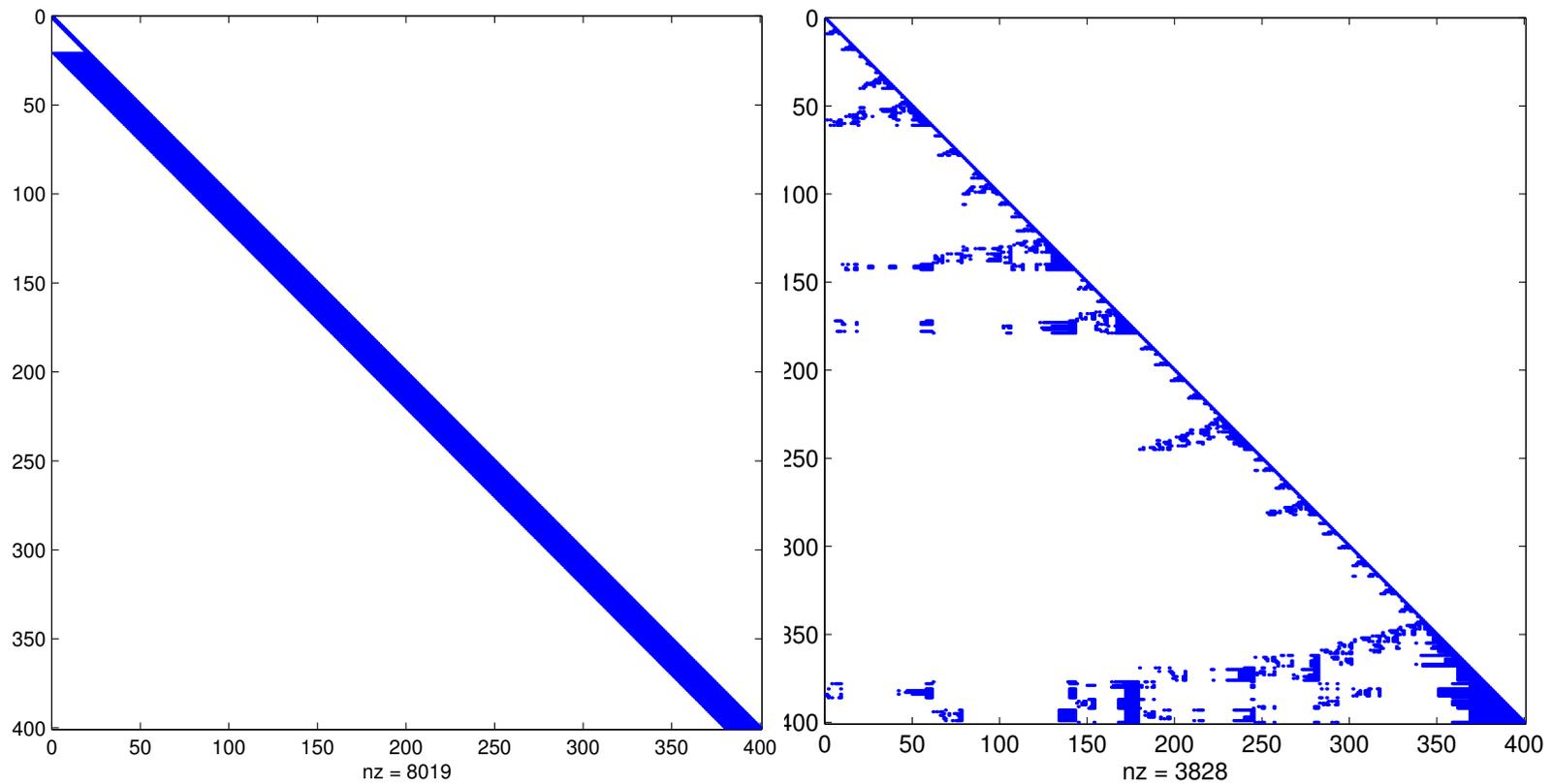
Dabei Matrixeinträge umordnen (“symmetric minimum degree”)



# Direkter Löser

## 2D Randwertproblem

Dreiecksfaktoren  $L$  aus Choleskyzerlegung



## Direkter Löser

### 2D Randwertproblem

$h$	Problemgröße	Rechenzeit ( flops )	Speicherbedarf (Nichtnullen.)
$\frac{1}{31}$	961	$2.68 \cdot 10^5$	$1.11 \cdot 10^4$
$\frac{1}{63}$	3969	$2.95 \cdot 10^6$	$6.55 \cdot 10^4$
$\frac{1}{127}$	16129	$2.89 \cdot 10^7$	$3.56 \cdot 10^5$
$\frac{1}{255}$	65025	$2.92 \cdot 10^8$	$1.89 \cdot 10^6$

- Rechenzeit wächst mehr als linear  
(Faktor 10 bei Vervierfachung der Problemgröße)
- Speicherbedarf wächst mehr als linear  
(Faktor 6 bei Vervierfachung der Problemgröße)

# Elementare Lösungstechniken — Iterativer Löser

Grunditerationsverfahren.

$$x^{(k+1)} = x^{(k)} + B^{-1}(b - Ax^{(k)}), \forall k, \quad x^{(0)} = 0.$$

Zerlegung von  $A$

$$A = D - E - E^T = \left( \begin{array}{c} \diagdown \end{array} \right) - \left( \begin{array}{c} \triangleleft \end{array} \right) - \left( \begin{array}{c} \triangleright \end{array} \right)$$

Mögliche Approximationen  $B$  an  $A$ .

$B$	$D$	$D - E$	$(D - E)D^{-1}(D - E^T)$
Bezeichnung	Jacobi	Gauß–Seidel	symmetrisches Gauß–Seidel

Iteration stoppen, falls  $\|b - Ax^{(k)}\|_\infty \leq 10^{-8} \|b\|_\infty$ .

# Einfache Iterativer Löser

## 2D Randwertproblem

$h$	Problemgröße	Jacobi		Gauß-Seidel	
		Rechenzeit ( flops )	Anzahl Iterationen	Rechenzeit ( flops )	Anzahl Iterationen
$\frac{1}{31}$	961	$5.43 \cdot 10^7$	3834	$3.43 \cdot 10^7$	1918
$\frac{1}{63}$	3969	$9.06 \cdot 10^8$	15354	$5.73 \cdot 10^8$	7678
$\frac{1}{127}$	16129	$1.48 \cdot 10^{10}$	61432	$9.37 \cdot 10^9$	30717

- Ergebnis niederschmetternd, schlimmer als direkter Löser
- Rechenzeit wächst mehr als linear  
(Faktor 16 bei Vervierfachung der Problemgröße)
- Anzahl Iterationen wächst!  
(Faktor 4 bei Vervierfachung der Problemgröße)

## Iterativer Löser — Polynomielle Verfahren

$A$  symmetrisch positiv definit  $\Rightarrow$  verwende vorkond.  $cg$ -Verfahren.

### 2D Randwertproblem

$h$	Problemgröße	Jacobi Vork.		sym. Gauß-Seidel Vork.	
		Rechenzeit ( flops )	Anzahl Iterationen	Rechenzeit ( flops )	Anzahl Iterationen
1/31	961	$1.29 \cdot 10^6$	59	$1.16 \cdot 10^6$	34
1/63	3969	$1.09 \cdot 10^7$	120	$9.08 \cdot 10^6$	64
1/127	16129	$9.06 \cdot 10^7$	245	$7.00 \cdot 10^7$	121
1/255	65025	$7.45 \cdot 10^8$	499	$5.09 \cdot 10^8$	218

- Ergebnis deutlich besser, aber immer noch unbefriedigend
- Rechenzeit wächst weiterhin mehr als linear, Iterationszahl steigt!
- Approximation an die Ausgangsmatrix anscheinend zu grob

## Iterativer Löser — Unvollst. Dreieckszerl.

1 Schritt Cholesky:

$$A = \begin{pmatrix} \alpha & v^\top \\ v & B \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ v & I \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha} & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} \alpha & v^\top \\ 0 & I \end{pmatrix},$$

wobei  $S = B - \frac{vv^\top}{\alpha}$  Schur-Komplement. Wiederhole Vorgang für  $S$ .

$$S = \tilde{L}\tilde{D}^{-1}\tilde{L}^\top \Rightarrow A = \begin{pmatrix} \alpha & 0 \\ v & \tilde{L} \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & \tilde{D} \end{pmatrix}^{-1} \begin{pmatrix} \alpha & v^\top \\ 0 & \tilde{L}^\top \end{pmatrix},$$

Unvollständige Zerlegung. Ersetze  $S$ :

$$S \longrightarrow \tilde{S}$$

wobei  $\tilde{S}$  identisch mit  $S$ , dort wo  $B$  von Null verschieden ist.

Liefert approximative Zerlegung von  $A$ .

$$A \approx (\hat{D} - \hat{E})\hat{D}^{-1}(\hat{D} - \hat{E})^\top.$$

“ICHOL(0)” (bzw. ILU(0) im unsymmetrischen Fall).

## Iterativer Löser — Unvollst. Dreieckszerl.

ICHOL(0) als Vorkond. für das cg-Verfahren

### 2D Randwertproblem

$h$	Problem- größe	ICHOL(0) Vork.	
		Rechenzeit ( flops )	Anzahl Iterationen
1/31	961	$9.40 \cdot 10^5$	29
1/63	3969	$7.40 \cdot 10^6$	55
1/127	16129	$5.69 \cdot 10^7$	104
1/255	65025	$4.51 \cdot 10^8$	204

- Ergebnis etwas besser als sym. Gauß-Seidel, trotzdem unbefriedigend
- Verfahren skaliert immer noch nicht

# Elementare Lösungstechniken

## Zusammenfassung

- Direkte Löser
  - ⊕ Keine Konvergenzprobleme
  - ⊖ hohen Speicherbedarf
  - ⊖ Rechenzeit
- Einfache Iterationsverfahren (Jacobi, Gauß–Seidel)
  - ⊕ geringer Speicherbedarf
  - ⊖ große Konvergenzprobleme
  - ⊖ Enorme Rechenzeit
- cg mit Vorkond. (ICHOL(0), symm. Gauß–Seidel)
  - ⊕ geringer Speicherbedarf
  - ⊖ Konvergenzprobleme für große  $n$
  - ⊖ Rechenzeit für große  $n$

# Glättungsanalyse

Warum sind elementare Verfahren so schlecht?

Der Einfachheit halber: 1D-Problem, Jacobi  $B = \frac{2}{h^2}I$ , Grunditeration.

Fehler  $e^{(k+1)} = x - x^{(k+1)}$  der Gleichung  $T_h x = b$ .

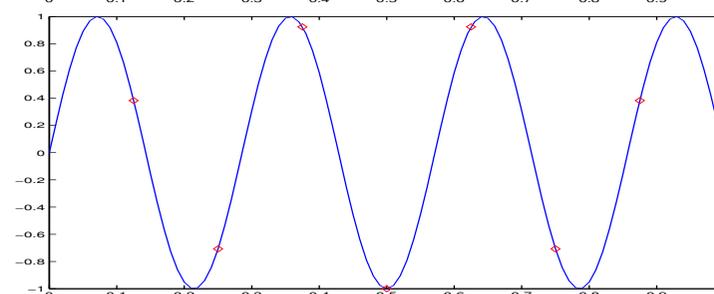
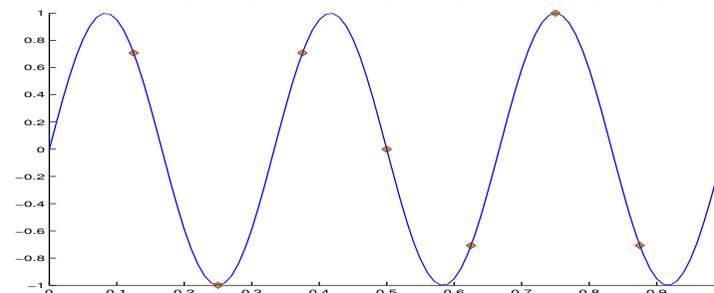
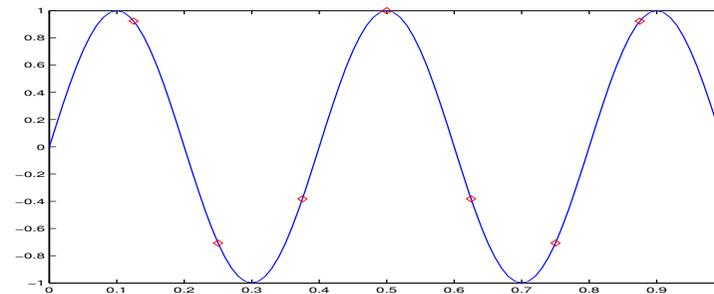
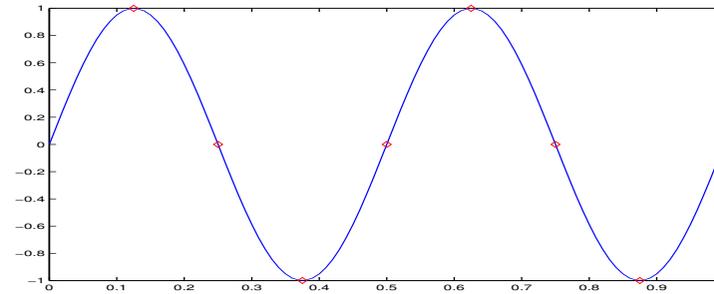
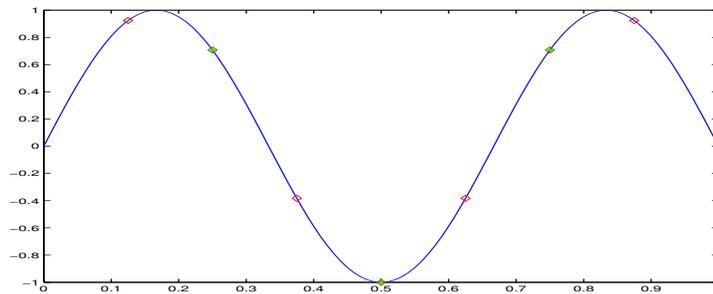
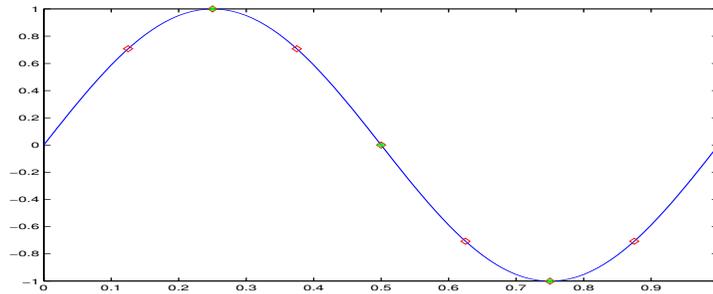
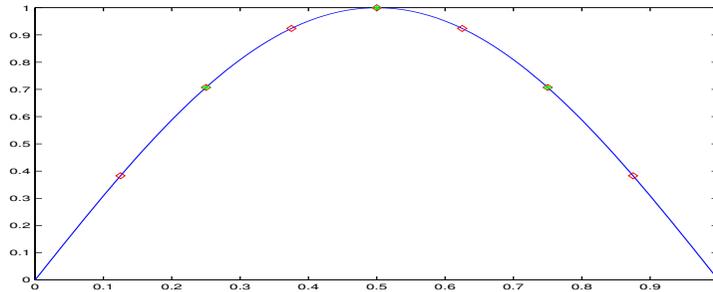
$$e^{(k+1)} = x - x^{(k+1)} = x - \left[ x^{(k)} + \frac{h^2}{2}(T_h x - T_h x^{(k)}) \right] = \left( I - \frac{h^2}{2}T_h \right) e^{(k)}$$

Zur Analyse verwende für  $e^{(k)}$  die Eigenvektoren von  $T_h$ .

Eigenvektoren  $s^{(k)}$  von  $T_h$ : Restriktion von  $s^{(k)}(x) = \sin(k\pi x)$  auf  $\Omega_h$ ,  $\forall k$ .

Zugehörige Eigenwerte:  $\lambda_k = \frac{4}{h^2} \sin^2 \frac{k\pi h}{2}$ ,  $\forall k$

# Glättungsanalyse — Eigenvektoren

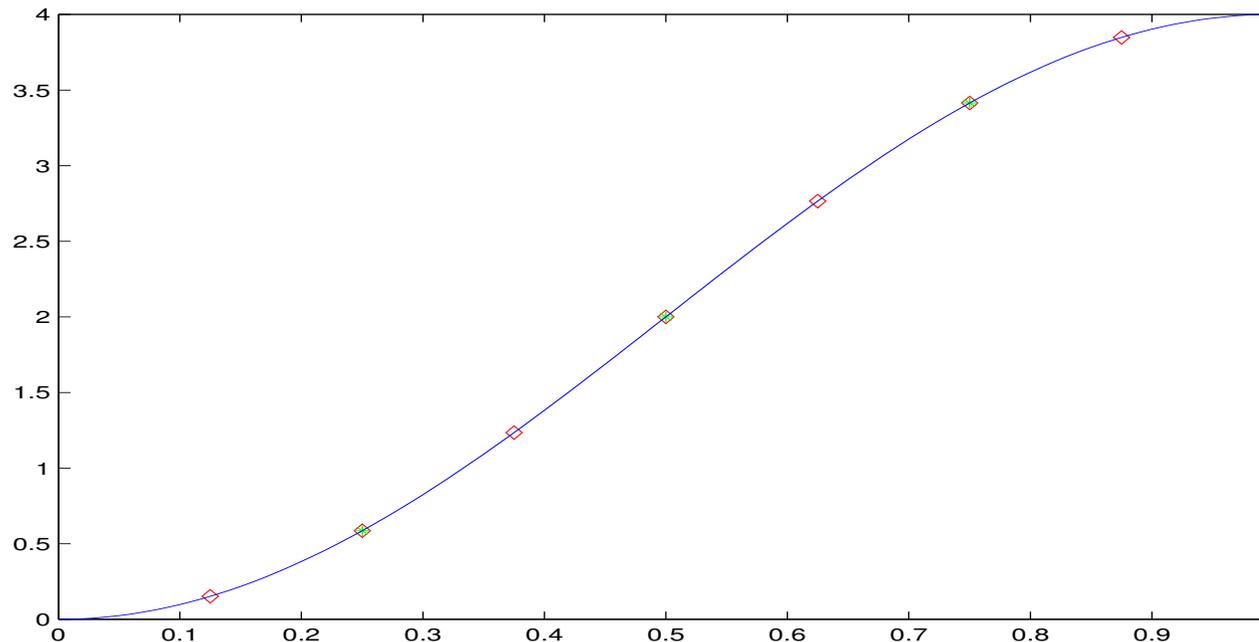


# Glättungsanalyse — Eigenvektoren

Beobachtung. Bei höherer Diskretisierung  $N \rightarrow 2N$ :

- niedrige Frequenzen bleiben erhalten
- höhere Frequenzen kommen hinzu

Eigenwerte (bis auf den Faktor  $\frac{1}{h^2}$ )

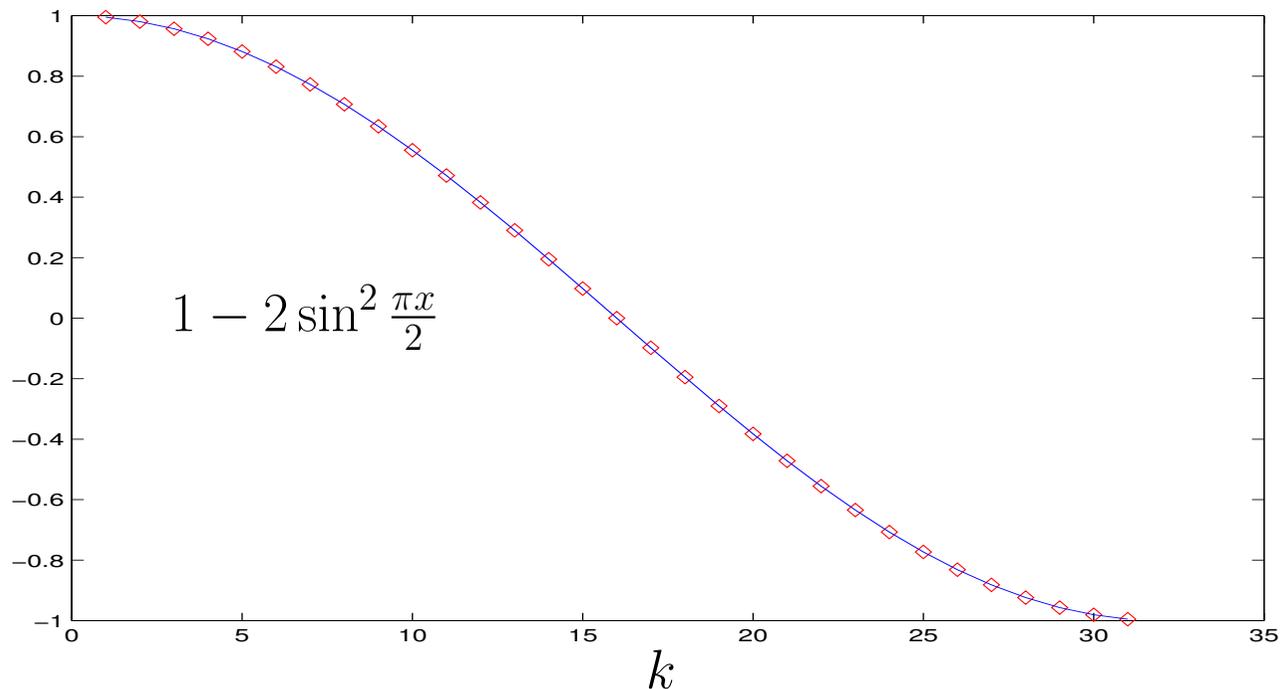


# Glättungsanalyse — Jacobi–Verfahren

Schreibe Fehler  $e$  als  $e = \sum_{k=1}^N \alpha_k s^{(k)}$

$$(I - \frac{h^2}{2} T_h) e = \sum_{k=1}^N \alpha_k (s^{(k)} - \frac{h^2}{2} T_h s^{(k)}) = \sum_{k=1}^N \alpha_k \left( 1 - 2 \sin^2 \frac{k\pi h}{2} \right) s^{(k)}.$$

Dämpfungsfaktoren  $|1 - 2 \sin^2 \frac{k\pi h}{2}|$  für die Eigenvektoren



# Glättungsanalyse — Jacobi–Verfahren

Beobachtung.

- Frequenzen im mittleren Bereich (um  $\frac{N}{2}$ ) werden stark gedämpft,
- keine Dämpfung für niedrige Frequenzen oder hohe Frequenzen

Steuere Dämpfungsverhalten durch Dämpfungsparameter  $\omega$ ,  $B \rightarrow \frac{1}{\omega}B$ .

Im Falle des Jacobi–Verfahrens

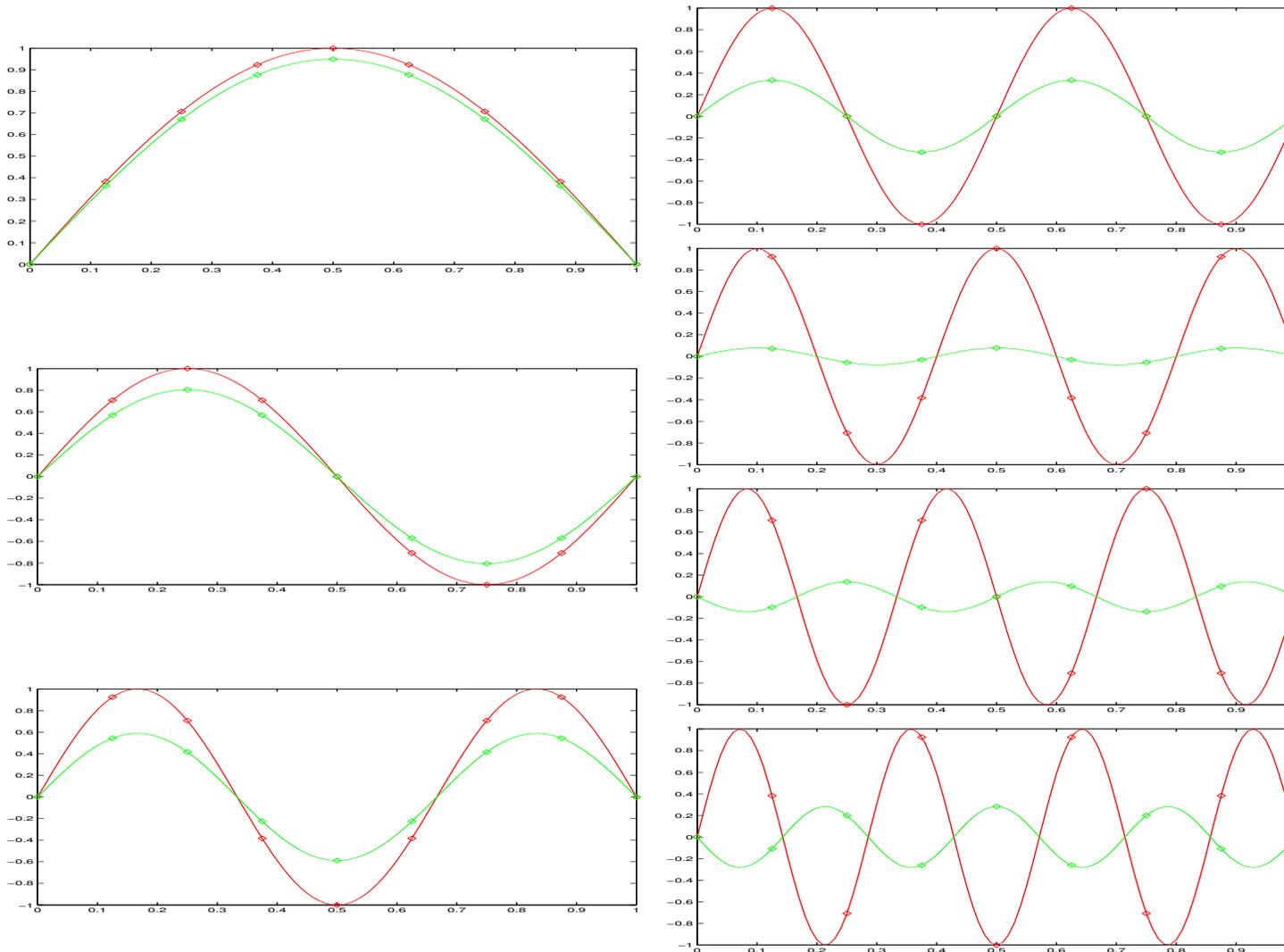
$$\left(I - \frac{\omega h^2}{2} T_h\right) e = \sum_{k=1}^N \alpha_k \left(s^{(k)} - \frac{\omega h^2}{2} T_h s^{(k)}\right) = \sum_{k=1}^N \alpha_k \left(1 - 2\omega \sin^2 \frac{k\pi h}{2}\right) s^{(k)}.$$

Optimiere  $\omega$  für hohe Frequenzen ( $k \geq \frac{N}{2}$ )

$$\begin{aligned} \max_{k > \frac{N}{2}} \left| 1 - 2\omega \sin^2 \frac{k\pi h}{2} \right| &= \min \\ \implies \omega &= \frac{2}{3}, \quad \max_{k > \frac{N}{2}} \left| 1 - \frac{4}{3} \sin^2 \frac{k\pi h}{2} \right| \leq \frac{1}{3} \end{aligned}$$

# Glättungsanalyse — Jacobi-Verfahren

Geglättete Eigenvektoren **vorher** und **nachher**



# Glättungsanalyse — Zusammenfassung

Beobachtungen.

- ungedämpftes Jacobi–Verfahren glättet nur Frequenzen im mittleren Bereich
- gedämpftes ( $\omega = \frac{2}{3}$ ) Jacobi–Verfahren dämpft hohe Frequenzen
- Praktisch keine Dämpfung bei niedrigen Frequenzen
- **ABER:** Niedrigen Frequenzen treten bei Problem  $T_{2h}$  der Dimension  $2h$  wieder auf
- Konsequenz. Brauchen zusätzlichen Korrekturschritt, der die glatten Anteile reduziert.  
Idee: Verwende hierzu gröberes Gitter  $\Omega_{2h}$  und  $T_{2h}$ .

# Grobitterkorrektur

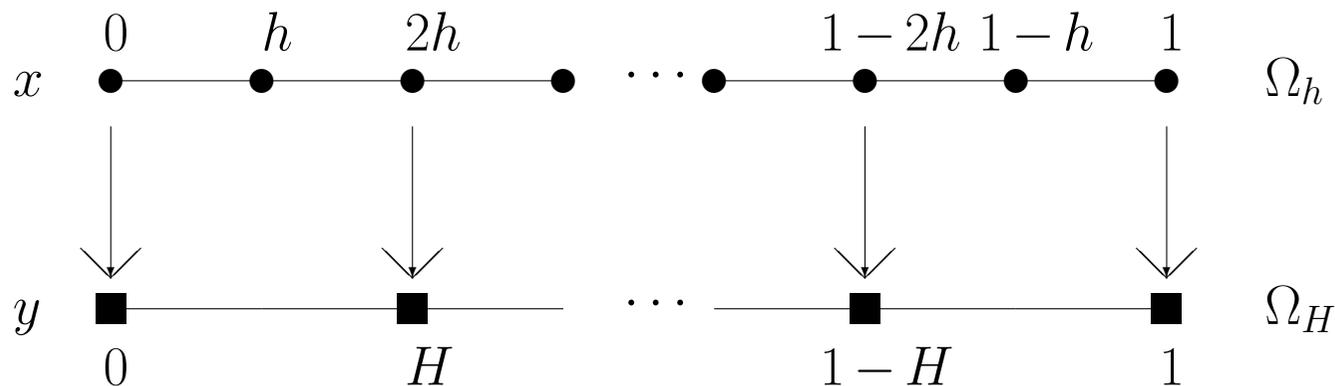
Glättungsschritt benötigt zusätzlichen Grobitterkorrekturschritt.

$H = 2h$ , Übergang von  $\Omega_h \longrightarrow \Omega_H$  Restriktion  
 $\Omega_H \longrightarrow \Omega_h$  Prolongation

$x \in \Omega_h$ , Werte auf dem Gitter  $x(h), \dots, x((N-1)h)$

$y \in \Omega_H$ , Werte auf dem Gitter  $y(H), \dots, y(\frac{N-1}{2}H)$

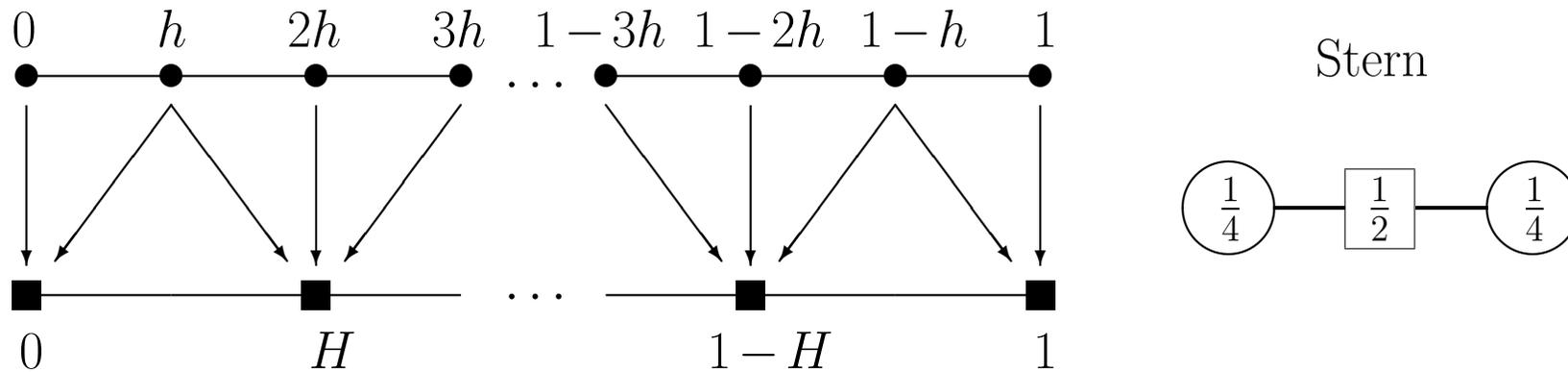
Triviale Restriktion schlecht:  $y(iH) = x(iH) = x(2ih), \forall i$



# Grobitterkorrektur — Gitterübergang

Restriktion  $R$ . Mittlung über die Nachbarn

$$y(iH) = \frac{1}{4}x(iH - h) + \frac{1}{2}x(iH) + \frac{1}{4}x(iH + h), \quad \forall i$$



Prolongation  $P$ . Lineare Interpolation.

$$x(ih) = \begin{cases} y(jH) & \text{falls } i = 2j \\ \frac{1}{2}y(jH) + \frac{1}{2}y(jH + H) & \text{falls } i = 2j + 1 \end{cases}, \quad \forall i$$

Prinzip der Restriktion wird umgekehrt.  $P = 2R^\top$

## Zweigitterverfahren

Kombination Jacobi (gedämpft) / Grobgitterkorrektur  $T_h^{-1} \approx PT_H^{-1}R$ .

$$x_h^{(k+\frac{1}{2})} = x_h^{(k)} + \frac{2}{3} \cdot \frac{h^2}{2} (b_h - T_h x_h^{(k)})$$

$$x_h^{(k+1)} = x_h^{(k+\frac{1}{2})} + P_h T_H^{-1} R_h (b_h - T_h x_h^{(k+\frac{1}{2})})$$

Bisher: für  $T_H$  Gleichungssystem lösen (z.B. mit Cholesky)

Ausweg: wende die Zweigitter-Verfahren wiederum (rekursiv) auf  $T_H$  an

Für  $b_H = R_h (b_h - T_h x_h^{(k+\frac{1}{2})})$ , ersetze  $T_H x_H = b_H$  durch

$$\left. \begin{aligned} x_H^{(\frac{1}{2})} &= x_H^{(0)} + \frac{2}{3} \cdot \frac{H^2}{2} (b_H - T_H x_H^{(0)}) \\ x_H^{(1)} &= x_H^{(\frac{1}{2})} + P_H T_{2H}^{-1} R_H (b_H - T_H x_H^{(\frac{1}{2})}) \end{aligned} \right\} \implies x_h^{(k+1)} \approx x_h^{(k+\frac{1}{2})} + P_h x_H^{(1)}$$

Problemreduktion  $h \longrightarrow 2h = H \longrightarrow 2H = 4h \longrightarrow \dots$

# Mehrgitterverfahren

Hierarchische Gitter  $\Omega_{h_1} \subseteq \Omega_{h_2} \subseteq \dots \subseteq \Omega_{h_l}$ , z.B.  $h_{s+1} = h_s/2, \forall s$

Zu jedem Gitter  $\Omega_{h_s}$  diskretisierte Gleichung  $A_{h_s} x_{h_s} = b_{h_s}$

Zwischen den Gittern Übergänge  $\Omega_{h_s} \longrightarrow \Omega_{h_{s-1}}$  Restriktion  $R_s$   
 $\Omega_{h_{s-1}} \longrightarrow \Omega_{h_s}$  Prolongation  $P_s$

Auf jedem Gitter Glätter  $S_{h_s}$ , z.B. mit Hilfe Zerlegung von  $A_{h_s}$

$$A_{h_s} = D_{h_s} - E_{h_s} - E_{h_s}^\top = \left( \begin{array}{c} \diagdown \end{array} \right) - \left( \begin{array}{c} \triangleleft \end{array} \right) - \left( \begin{array}{c} \triangleright \end{array} \right)$$

Wähle z.B.  $S_{h_s} = \frac{1}{3} \cdot D_{h_s}$  (gedämpftes Jacobi-Verf.)

oder  $S_{h_s} = D_{h_s} - E_{h_s}$  (Gauß-Seidel)

oder  $S_{h_s} = (D_{h_s} - E_{h_s}) D_{h_s}^{-1} (D_{h_s} - E_{h_s}^\top)$  (symm. Gauß-Seidel)

# Mehrgitterverfahren — Algorithmus

Gegeben:  $h_s$  für ein  $s \in \{1, 2, \dots, l\}$ , rechte Seite  $b_{h_s}$ , Näherung  $x_{h_s}$   
Mehrgitterschritt berechnet rekursiv neue Näherung  $y_{h_s}$

**Falls**  $s = 1$ , löse  $A_{h_1}y_{h_1} = b_{h_1}$  direkt.

**Sonst**  $y_{h_s} := x_{h_s}$

**Vorglättung**  $\left\{ \begin{array}{l} \text{Für } k = 1, 2, \dots, \nu_1 \\ y_{h_s} := y_{h_s} + S_{h_s}^{-1}(b_{h_s} - A_{h_s}y_{h_s}) \end{array} \right.$

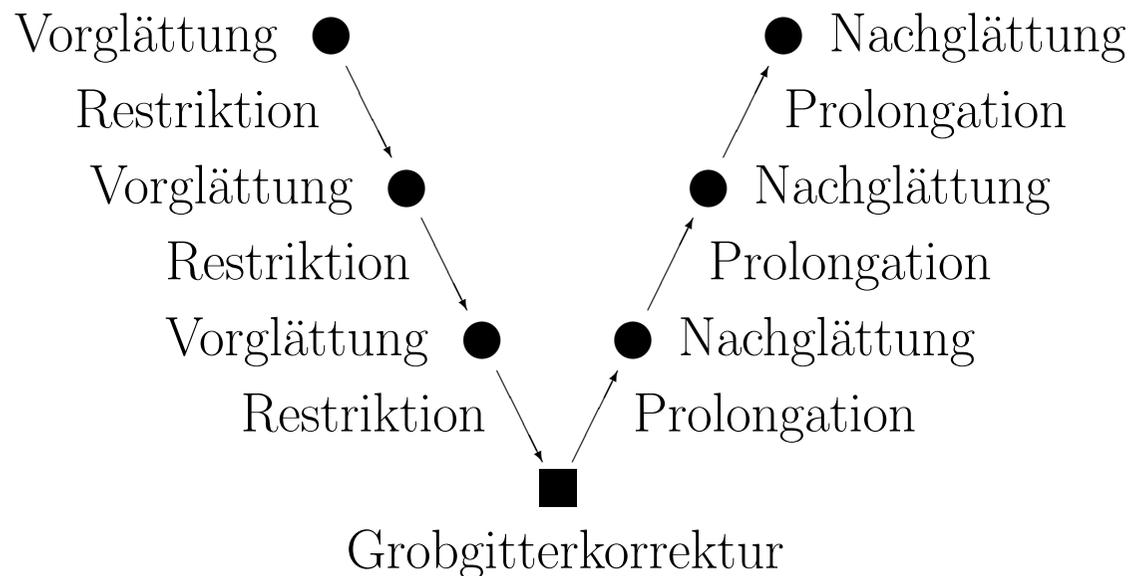
**Grobitterkorrektur**  $\left\{ \begin{array}{l} \text{Für } k = 1, 2, \dots, \mu \\ x_{h_{s-1}} := 0, b_{h_{s-1}} := R_s(b_{h_s} - A_{h_s}y_{h_s}) \\ \text{Rufe } \underline{\text{Mehrgitterschritt}} \text{ auf mit } h_{s-1}, b_{h_{s-1}}, x_{h_{s-1}} \\ \rightarrow \text{Ergebnis } y_{h_s}. \\ y_{h_s} := y_{h_s} + P_s y_{h_{s-1}} \end{array} \right.$

**Nachglättung**  $\left\{ \begin{array}{l} \text{Für } k = 1, 2, \dots, \nu_2 \\ y_{h_s} := y_{h_s} + S_{h_s}^{-1}(b_{h_s} - A_{h_s}y_{h_s}) \end{array} \right.$

## Mehrgitterverfahren — Anmerkungen

- Bei mehrmaliger Anwendung nur Summe  $\nu = \nu_1 + \nu_2$  benötigt.
- Mehrgitterschritt für  $\mu = 1$  rekursiven Aufruf für  $\Omega_{h_s} \rightarrow$  “V-Zyklus”
- Mehrgitterschritt für  $\mu = 2$  rekursive Aufrufe für  $\Omega_{h_s} \rightarrow$  “W-Zyklus”

### Schema eines Mehrgitterschrittes “V-Zyklus”



# Mehrgitterverfahren — 1D Fall

## 1D Randwertproblem, Mehrgitterverfahren

$h$	V-Zykl., $\nu = 1$		V-Zykl., $\nu = 2$		W-Zykl., $\nu = 1$		W-Zykl., $\nu = 2$	
	Zeit (flops)	Anz. Iter.	Zeit (flops)	Anz. Iter.	Zeit (flops)	Anz. Iter.	Zeit (flops)	Anz. Iter.
$\frac{1}{511}$	$5.8 \cdot 10^5$	21	$5.7 \cdot 10^5$	15	$2.4 \cdot 10^6$	17	$1.6 \cdot 10^6$	9
$\frac{1}{1023}$	$1.2 \cdot 10^6$	21	$1.2 \cdot 10^6$	16	$5.5 \cdot 10^6$	17	$3.7 \cdot 10^6$	9
$\frac{1}{2047}$	$2.5 \cdot 10^6$	22	$2.4 \cdot 10^6$	16	$1.2 \cdot 10^7$	16	$9.2 \cdot 10^6$	10
$\frac{1}{4095}$	$4.9 \cdot 10^6$	22	$4.9 \cdot 10^6$	16	$2.6 \cdot 10^7$	16	$1.8 \cdot 10^7$	9

- Verfahren skaliert linear!
- Mehr Glättung  $\rightarrow$  weniger Iterationen, aber teurer
- W-Zyklus hier aufwendiger, dafür weniger Iterationen

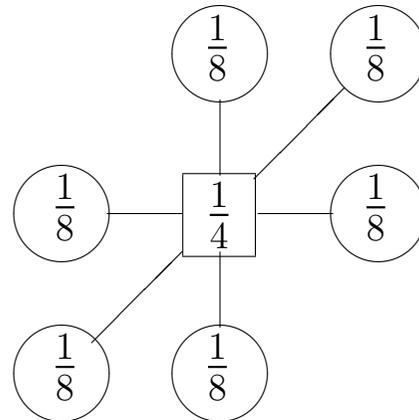
## Mehrgitterverfahren im 2D–Fall

$x \in \Omega_h, y \in \Omega_H$ . Werte auf dem Gitter  $x(ih, jh), y(kH, lH) \forall i, j, k, l$ .

Übertragung Restriktion:  $\forall i, j$

$$\begin{aligned} y(iH, jH) = & \frac{1}{4}x(iH, jH) + \frac{1}{8} [x(iH - h, jH) + x(iH + h, jH) \\ & + x(iH, jH - h) + x(iH, jH + h) \\ & + x(iH - h, jH - h) + x(iH + h, jH + h)] \end{aligned}$$

In Form eines Sternes



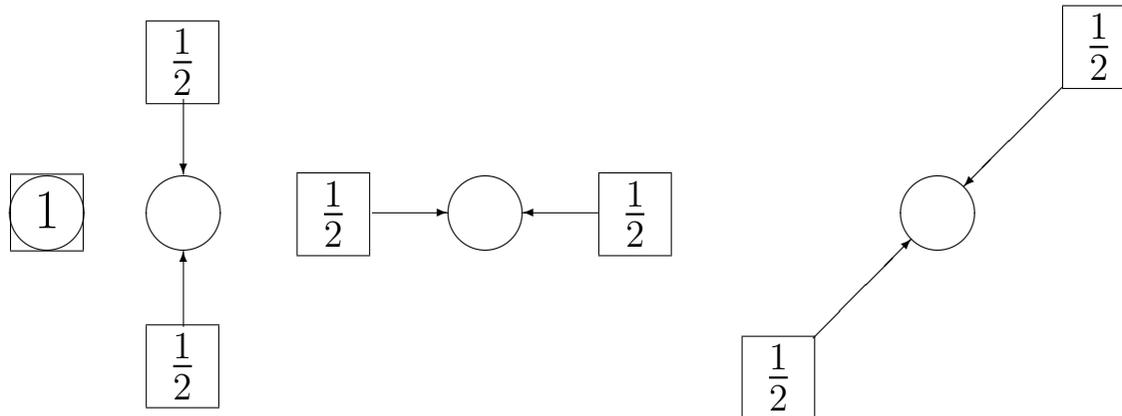
Ein weiteres Beispiel wäre die fehlenden Nachbarn auch mit einzubeziehen.

# Mehrgitterverfahren im 2D-Fall

Hierzu passende Prolongation.

$$x(ih, jh) = \begin{cases} y(kH, lH) & i = 2k, j = 2l \\ \frac{1}{2} [y(kH, lH) + y(kH + H, lH)] & i = 2k + 1, j = 2l \\ \frac{1}{2} [y(kH, lH) + y(kH, lH + H)] & i = 2k, j = 2l + 1 \\ \frac{1}{2} [y(kH, lH) + y(kH + H, lH + H)] & i = 2k + 1, j = 2l + 1 \end{cases}$$

Zugehörige Sterne



## Mehrgitterverfahren im 2D–Fall

2D Randwertproblem, Mehrgitterverfahren

$h$	Cholesky	V–Zykl., $\nu = 2$		W–Zykl., $\nu = 2$	
	Zeit (flops)	Zeit (flops)	Anz. Iter.	Zeit (flops)	Anz. Iter.
$\frac{1}{31}$	$2.7 \cdot 10^5$	$1.7 \cdot 10^6$	26	$2.3 \cdot 10^6$	20
$\frac{1}{63}$	$3.0 \cdot 10^6$	$7.7 \cdot 10^6$	27	$1.0 \cdot 10^7$	20
$\frac{1}{127}$	$2.9 \cdot 10^7$	$3.2 \cdot 10^7$	27	$4.5 \cdot 10^7$	21
$\frac{1}{255}$	$2.9 \cdot 10^8$	$1.3 \cdot 10^8$	28	$1.9 \cdot 10^8$	21

- direktes Verfahren schneidet noch ganz gut ab
- Mehrgitterverfahren hier mit Jacobi. Verbesserung mit Gauß–Seidel–Glätter

## Mehrgitterverfahren als Vorkonditionierer

Im Prinzip kann jedes zusammengesetzte iterative Verfahren gelesen werden als ein Schritt eines anderen Verfahren. Z.B.

$$\begin{aligned}x^{(k+1)} &= x^{(k)} + B_1(b - Ax^{(k)}) \\x^{(k+2)} &= x^{(k+1)} + B_2(b - Ax^{(k+1)})\end{aligned}$$

1. Schritt hat Iterationsmatrix  $I - B_1A$ , 2. Schritt  $I - B_2A$ . Für das zusammengesetzte Verfahren bekommt man

$$\begin{aligned}I - BA &\equiv (I - B_2A)(I - B_1A) = I - [B_2 + B_1 - B_2AB_1]A \\&\Rightarrow x^{(k+2)} = x^{(k)} + [B_1 + B_2 - B_2AB_1](b - Ax^{(k)})\end{aligned}$$

Prinzip auf mehr als 2 Schritte übertragbar. Beispiele

- Symm. Gauß–Seidel.  $A = D - E - E^\top$ .  $B_1 = (D - E^\top)^{-1}$ ,  $B_2 = (D - E)^{-1}$ .  $\Rightarrow B = [B_1 + B_2 - B_2AB_1] = (D - E)^{-1}D(D - E^\top)^{-1}$
- ein Mehrgitterschritt auf feinstem Gitter inklusive rekursivem Aufruf

## Mehrgitterverfahren als Vorkonditionierer

Mehrgitterapproximation als Vorkonditionierer für  $cg$ . Hierzu nötig:

- $\nu_1 = \nu_2$  (Genauso viel Nachglättung wie Vorglättung)
- bei Nachglättung  $S_{h_s}^\top$  anstelle von  $S_{h_s}$ .

### 2D Randwertproblem $cg$ mit Mehrgittervork.

$h$	V-Zykl., $\nu_1 = \nu_2 = 1$				W-Zykl., $\nu_1 = \nu_2 = 1$			
	ged. Jacobi		Gauß-Seidel		ged. Jacobi		Gauß-Seidel	
	Zeit (flops)	Anz. Iter.	Zeit (flops)	Anz. Iter.	Zeit (flops)	Anz. Iter.	Zeit (flops)	Anz. Iter.
$\frac{1}{31}$	$6.8 \cdot 10^5$	11	$6.6 \cdot 10^5$	9	$1.1 \cdot 10^6$	10	$9.7 \cdot 10^5$	8
$\frac{1}{63}$	$2.9 \cdot 10^6$	11	$2.8 \cdot 10^6$	9	$4.7 \cdot 10^6$	10	$4.3 \cdot 10^6$	8
$\frac{1}{127}$	$1.3 \cdot 10^7$	12	$1.2 \cdot 10^7$	9	$2.0 \cdot 10^7$	10	$1.8 \cdot 10^7$	8
$\frac{1}{255}$	$5.3 \cdot 10^7$	12	$4.7 \cdot 10^7$	9	$8.2 \cdot 10^7$	10	$7.5 \cdot 10^7$	8

# Mehrgitterverfahren — Zusammenfassung

- ⊕ überlegene auf das Problem angepasstes Verfahren
- ⊕ Verfahren skaliert linear
- ⊕ Speicherbedarf moderat
- ⊕ Auf ähnliche Probleme übertragbar (Finite Elemente, Sattelpunktprobleme, Parabolische Aufgaben, ...)
- ⊖ Benötigt problembedingt eine vorher generierte Gitterhierarchie
- ⊖ Nicht mehr einsetzbar falls nur das feinste Gitter bekannt ist.
- ⊖ Für viele Problemklassen muss es individuell angepasst werden (robustere Glätter, andere Restriktion/Prolongation)

Versuch eines Ausweges in den letzten drei Fällen:

## Algebraische Mehrgitterverfahren