

# Iterative Glättungs-Verfahren zur Lösung linearer Gleichungssysteme und deren Parallelisierung

Michael Krätschmer

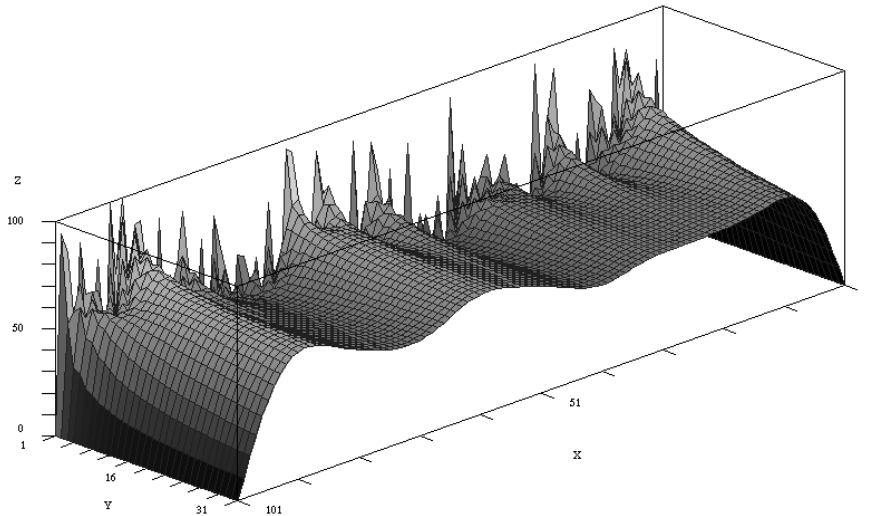


Zurück

Ende

# Glätter

Fehlerentwicklung während der Iteration:



Hochfrequente Fehleranteile werden schnell gedämpft; niederfrequente klingen nur langsam ab.



Zurück

Ende

# Einführung

## Allgemeine Form

$$u^{k+1} = \underbrace{(I - B^{-1}K)}_{=:C} u^k + B^{-1}f$$

bzw.

$$B\Delta u^k = \underbrace{f - Ku^k}_{=:r^k} \quad \text{und} \quad u^{k+1} = u^k + \Delta u^k$$



Zurück

Ende

# Einführung

## Allgemeine Form

$$u^{k+1} = \underbrace{(I - B^{-1}K)}_{=:C} u^k + B^{-1}f$$

bzw.

$$B\Delta u^k = \underbrace{f - Ku^k}_{=:r^k} \quad \text{und} \quad u^{k+1} = u^k + \Delta u^k$$

## Zentraler Konvergenzsatz

$$\rho(C) < 1 \quad \Leftrightarrow \quad u^k \xrightarrow{k \rightarrow \infty} u$$



Zurück

Ende

# Jacobi Verfahren

Approximation der Systemmatrix:  $B := D$

$$D = \begin{pmatrix} K_{11} & & & \\ & K_{22} & & \\ & & \dots & \\ & & & K_{nn} \end{pmatrix}$$



Zurück

Ende

```
Wähle  $u^0$   
 $r := f - Ku^0$   
 $\Delta u := D^{-1}r$   
 $\sigma := \sigma_0 := \langle \Delta u, r \rangle$   
 $k := 0$   
while  $\sigma > TOL \cdot \sigma_0$   
     $k := k + 1$   
     $u^k := u^{k-1} + \Delta u$   
     $r := f - Ku^k$   
     $\Delta u := D^{-1}r$   
     $\sigma := \langle \Delta u, r \rangle$   
end
```

$$u_i^k = u_i^{k-1} + \frac{1}{a_{ii}} \left( f_i - \sum_{j=1}^n a_{ij} u_j^{k-1} \right)$$



Zurück

Ende

# Speicherung der Systemmatrix



Zurück

Ende

$$K = \sum_{i=1}^p A_i^T K_i A_i$$

Im 1D Beispiel:

$$A_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad A_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Führt auf

$$K = \begin{pmatrix} K_1^{11} & K_1^{12} & 0 & 0 \\ K_1^{21} & K_1^{22} + K_2^{11} & K_2^{12} & 0 \\ 0 & K_2^{21} & K_2^{22} + K_3^{11} & K_3^{12} \\ 0 & 0 & K_3^{21} & K_3^{22} \end{pmatrix}$$



Zurück

Ende





Zurück

Ende

$$\begin{aligned}r^k &= f - \left( \sum_{i=1}^p A_i^T K_i A_i \right) u^k \\&= \sum_{i=1}^p A_i^T f_i - \sum_{i=1}^p A_i^T K_i \underbrace{(A_i u^k)}_{:=u_i^k} \\&= \sum_{i=1}^p \underbrace{(A_i^T [f_i - K_i u_i^k])}_{:=r_i^k}\end{aligned}$$

Im 1D Beispiel:

$$\begin{aligned}r^k &= A_1^T \left[ f_1 - K_1 \begin{pmatrix} u_1^k \\ u_2^k \end{pmatrix} \right] + A_2^T \left[ f_2 - K_2 \begin{pmatrix} u_2^k \\ u_3^k \end{pmatrix} \right] \\&\quad + A_3^T \left[ f_3 - K_3 \begin{pmatrix} u_3^k \\ u_4^k \end{pmatrix} \right]\end{aligned}$$



Zurück

Ende

# Matrix zur Lösung der Korrektur-Gleichungssysteme



Zurück

Ende

$$\begin{aligned} D^{-1} &= \text{diag}^{-1} \left( \sum_{i=1}^p A_i^T K_i A_i \right) \\ &= \left( \sum_{i=1}^p A_i^T \text{diag}(K_i) A_i \right)^{-1} \end{aligned}$$

Im 1D Beispiel:

$$D^{-1} = \begin{pmatrix} K_1^{11} & 0 & 0 & 0 \\ 0 & K_1^{22} + K_2^{11} & 0 & 0 \\ 0 & 0 & K_2^{22} + K_3^{11} & 0 \\ 0 & 0 & 0 & K_3^{22} \end{pmatrix}^{-1}$$



Zurück

Ende

$$\Delta u_i^k = D_i^{-1} r_i^k \quad \text{mit} \quad D_i = A_i D A_i^T$$



Zurück

Ende

$$\Delta u_i^k = D_i^{-1} r_i^k \quad \text{mit} \quad D_i = A_i D A_i^T$$

## Fehlerabschätzung

$$\sigma = \langle \Delta u^k, r \rangle = \sum_{i=1}^p \langle \Delta u_i^k, r_i^k \rangle$$



Zurück

Ende

## Korrektur

$$\Delta u_i^k = D_i^{-1} r_i^k \quad \text{mit} \quad D_i = A_i D A_i^T$$

## Fehlerabschätzung

$$\sigma = \langle \Delta u^k, r \rangle = \sum_{i=1}^p \langle \Delta u_i^k, r_i^k \rangle$$

## Update

$$u_i^{k+1} = u_i^k + \Delta u_i^k$$



Zurück

Ende

$$\Delta u_i^k = D_i^{-1} r_i^k \quad \text{mit} \quad D_i = A_i D A_i^T$$

## Fehlerabschätzung

$$\sigma = \langle \Delta u^k, r \rangle = \sum_{i=1}^p \langle \Delta u_i^k, r_i^k \rangle$$

## Update

$$u_i^{k+1} = u_i^k + \Delta u_i^k$$

## Zusammensetzen der neuen Iterierten

$$u^{k+1} = \sum_{i=1}^p A_i^T u_i^{k+1}$$



Zurück

Ende



# Gauß-Seidel Verfahren

Approximation der Systemmatrix:  $B := D + L$

$$D + L = \begin{pmatrix} K_{11} & & & \\ K_{21} & K_{22} & & \\ \vdots & & \ddots & \\ K_{n1} & K_{n2} & \cdots & K_{nn} \end{pmatrix}$$

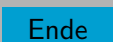
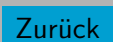


Zurück

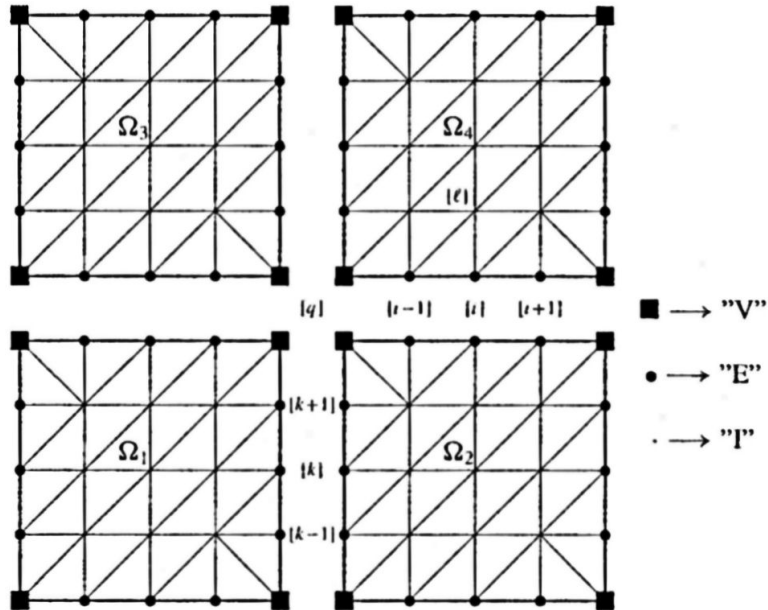
Ende

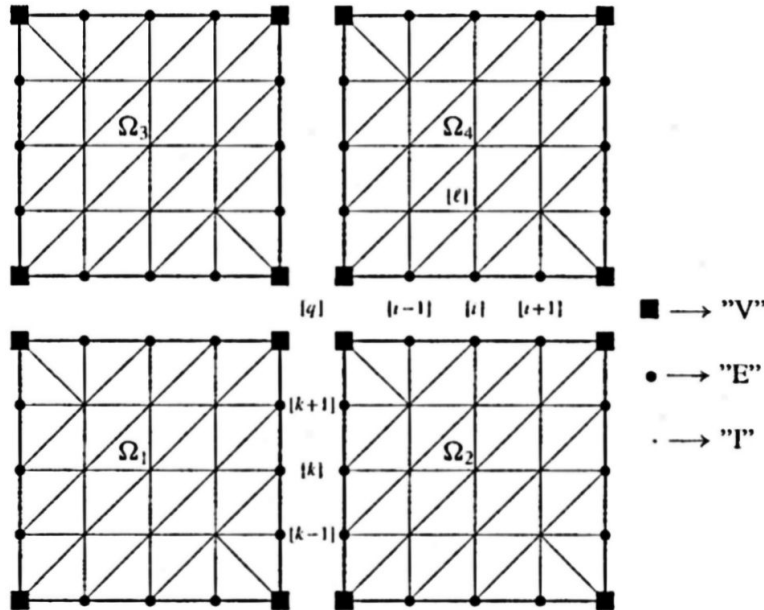
```
Wähle  $u^0$   
 $r := f - Ku^0$   
 $\sigma := \sigma_0 := \langle r, r \rangle$   
 $k := 0$   
while  $\sigma > TOL \cdot \sigma_0$   
     $k := k + 1$   
     $u^k := u^{k-1} + D^{-1} \cdot (f - Lu^k - (D + U) \cdot u^{k-1})$   
     $r := f - Ku^k$   
     $\sigma := \langle r, r \rangle$   
end
```

$$u_i^k = u_i^{k-1} + \frac{1}{a_{ii}} \left( f_i - \sum_{j=1}^{i-1} a_{ij} u_j^k - \sum_{j=i}^n a_{ij} u_j^{k-1} \right).$$



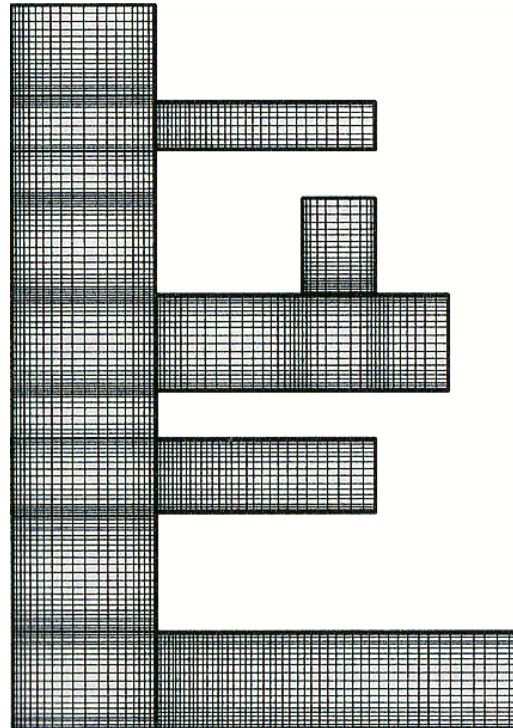
# Parallelisierung über Blockstruktur





$$\begin{pmatrix} K_V & K_{VE} & K_{VI} \\ K_{EV} & K_E & K_{EI} \\ K_{IV} & K_{IE} & K_I \end{pmatrix} \begin{pmatrix} u_V \\ u_E \\ u_I \end{pmatrix} = \begin{pmatrix} f_V \\ f_E \\ f_I \end{pmatrix}$$

# Beispielgitter - blockstrukturiert



Zurück

Ende

# Zusammenfassung

Die klassischen iterativen Methoden lassen sich zur Geschwindigkeitssteigerung parallelisieren. Während das Jacobi-Verfahren universell leicht parallelisierbar ist, hängt beim Gauß-Seidel-Verfahren die Realisierung einer effektiven Parallelisierung an der Struktur des numerischen Gitters.

An sogenannten "Interfaceknoten" kann es sinnvoll sein, trotz schlechterer Konvergenzeigenschaften das Jacobi-Verfahren anzuwenden.



Zurück

Ende