

Anordnungstechniken für konvektionsdominante Probleme im \mathbb{R}^3

Inhalt:

- Einführung
- Dimensionsunabhängige Verfahren
- Algorithmen für planare Graphen
- Anordnungen im \mathbb{R}^3
- Numerische Beispiele

Einführung

- betrachtet wird Konvektions-Diffusions-Gleichung mit Konvektionsrichtung b

$$-\varepsilon \Delta u + \langle b, \nabla u \rangle = f, \quad \text{in } \Omega \subset \mathbb{R}^3, \quad \varepsilon > 0.$$

- FEM-Diskretisierung führt zu Gleichungssystem mit Systemmatrix

$$A = A^D + A^C, \quad \text{mit}$$

- A^D entspricht dem Diffusionsanteil und ist symmetrisch,
 - A^C entspricht dem Konvektionsanteil und ist unsymmetrisch.
- Problem: bei dominierender Konvektion verschlechtert sich die Konvergenz (z.B. bei Mehrgitter)
 - Lösung: geeignetes Umsortieren der Unbekannten (z.B. entsprechend der Konvektionsrichtung).

Graph einer Matrix

- Der Graph G einer Matrix $A \in \mathbb{R}^{n \times n}$ ist definiert durch:

$$V(G) := \{1, \dots, n\} \quad \text{und}$$

$$E(G) := \{(i, j) \in V(G) \times V(G) : |a_{ij}| > 0\}.$$

- Kantenmengen für “starke” Matrixeinträge:

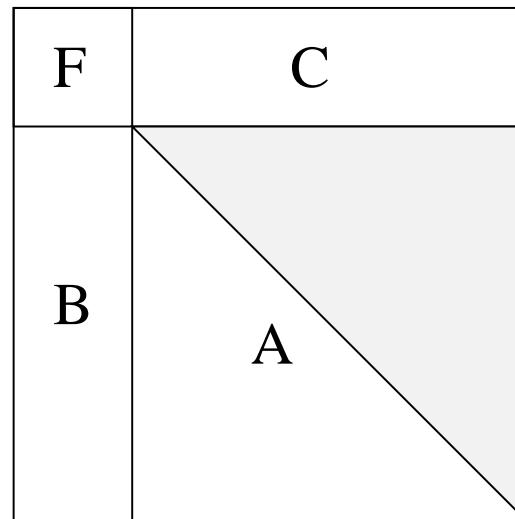
$$E_3(G) := \{(i, j) \in E(G) : |a_{ij}| > \kappa_1 |a_{ii}|\}, \quad 0 \leq \kappa_1 < 1$$

$$E_7(G) := \{(i, j) \in E(G) : |a_{ij}| > \kappa_2 |a_{ji}|\}, \quad \kappa_2 \geq 1.$$

- Bei linearem FEM-Ansatz entsprechen die Punkte bzw. Kanten im Netz den Knoten und Kanten im Graph.

Das Feedback-Vertex-Set Problem

- Feedback-Vertex-Set (FVS) eines Graphen $G = (V, E)$ ist Menge von Knoten, deren Entfernen den Graphen azyklisch macht.
- Bei Anordnung der Matrix mittels FVS ergibt sich Blockstruktur:



mit FVS-Knoten in F und oberer Dreiecksmatrix A .

- gesucht: kleines FVS im Vergleich zu A
- aber: FVS-Problem NP-vollständig

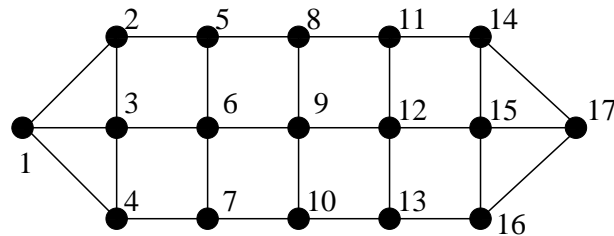
Dimensionsunabhängige Verfahren

Für die folgenden Verfahren wird lediglich die Matrix A bzw. deren Graph $G = (V, E)$ benötigt.

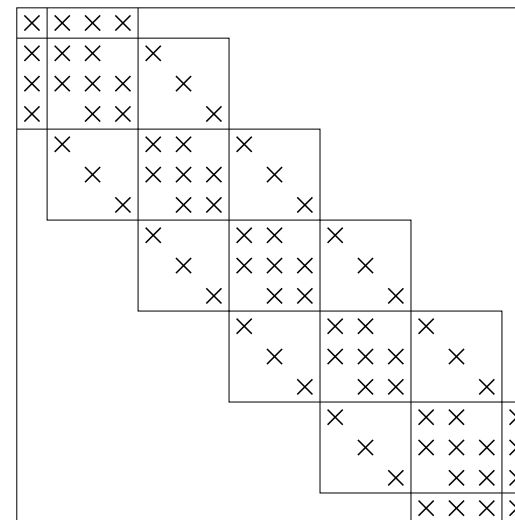
Reverse Cuthill-McKee (RCM)

- anwendbar bei symmetrischer Besetztheitsstruktur der Matrix
- Matrix wird auf Blocktridiagonalgestalt gebracht
- Reduktion des Fill-In's bei Gaußelimination

Beispiel für RCM:



Matrixgraph



angeordnete Matrix

Heuristischer FVS-Algorithmus

- Erweiterung eines Algorithmus zur exakten Lösung des FVS-Problems für spezielle Graphenklasse.
- Der Algorithmus beruht auf Transformationen des Graphen $G = (V, E)$:
 - t_1 - Sei $v \in V$ und $(v, v) \in E$. Setze $G = G - \{v\}$.
 - $t_{2,3}$ - Sei $v \in V$ ohne Nachfolger/Vorgänger. Setze $G = G - \{v\}$.
 - $t_{4,5}$ - Sei $v \in V$ mit genau einem Nachfolger (Vorgänger). Setze $G = G - \{v\}$ und füge Kanten zwischen allen Vorgängern (Nachfolgern) und dem Nachfolger (Vorgänger) zum Graphen hinzu.
 - t_6 - Falls t_1, \dots, t_5 auf keinen Knoten in G anwendbar, wähle einen beliebigen Knoten $v \in V$ und setze $G = G - \{v\}$.

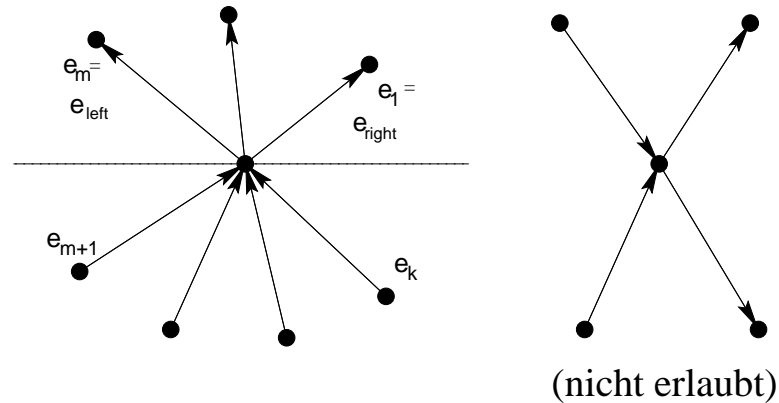
- Algorithmus:

```
FVS =  $\emptyset$ ;  
while  $|V(G)| > 0$  do  
    Wähle  $v \in V(G)$  und  $t \in \{t_1, \dots, t_5\}$  mit  $t$  anwendbar auf  $v$ ;  
    if kein  $t_i$  auf einen Knoten anwendbar then  
        sei  $v \in V$  beliebig und setze  $t = t_6$ ;  
     $G = t(G)$ ;  
    if  $t \in \{t_1, t_6\}$  then  $FVS = FVS \cup \{v\}$ ;  
endwhile;
```

- Falls t_6 nicht benutzt wird, ist das FVS minimal; Reihenfolge der gewählten Knoten dabei nicht relevant.
- Modifikation bei der Wahl des Knotens in t_6 :
 - a) Wahl eines Knotens mit global maximalem Grad: FVS wird kleiner, die Laufzeit aber nichtlinear.
 - b) Wahl eines Knotens mit lokal maximalem Grad: Algorithmus in Linearzeit, aber FVS größer als bei a).

Algorithmen für planare Graphen

- wichtige Voraussetzung ist die *One-Flow-Direction-Bedingung* (OFD-Bedingung): für $v \in V$ liegen die eingehenden und ausgehenden Kanten (sortiert nach dem Winkel) in zwei disjunkten Sektoren:

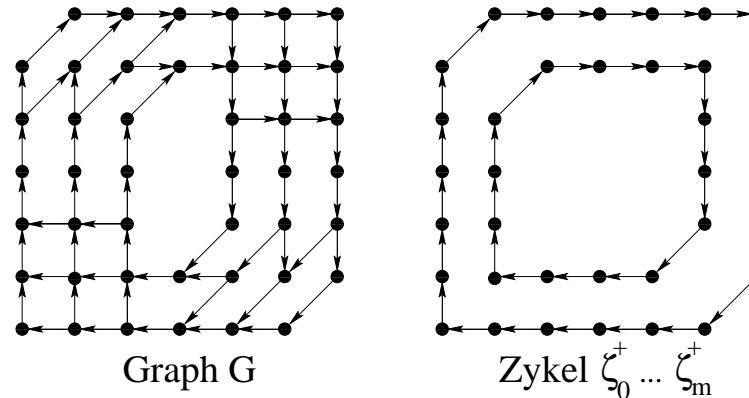


- Durch die OFD-Bedingung lassen sich *linke* und *rechte* Kanten $e_{left}(v)$ und $e_{right}(v)$ zuordnen.
- Konstruktion von Zykeln in G mit Iteration:

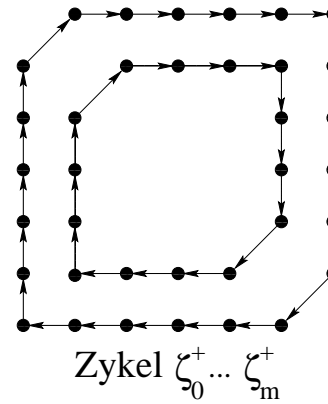
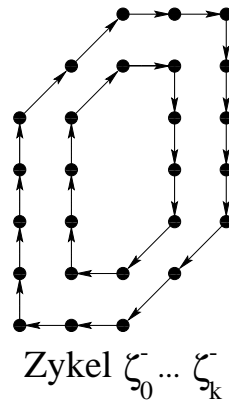
$$v_0 = v, \quad v_{i+1} = \text{linker Nachfolger von } v_i$$

- Sukzessive Anwendung der obigen Iteration und nachfolgendes Entfernen der konstruierten

Zykel führt zu einer Zerlegung des Graphen in Zykel $\zeta_0^+, \dots, \zeta_m^+$:

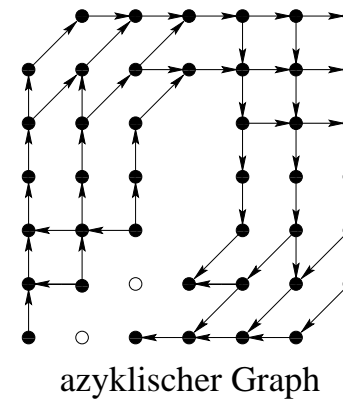
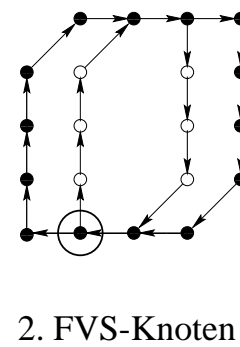
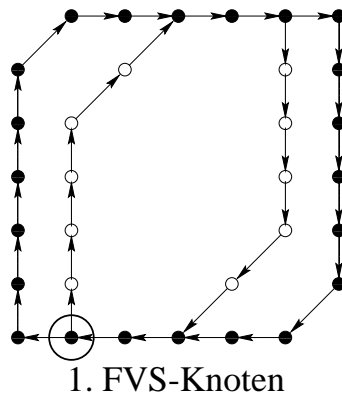


- Ordnet man Matrix entsprechend Reihenfolge der Knoten in den Zykeln an, entspricht jeder Zykel einem periodischen Tridiagonal-Block; diese Anordnung kann z.B. für ein Block-Gauß-Seidel-Verfahren genutzt werden.
- Obige Iteration kann auch mit den jeweils rechten Nachfolgern durchgeführt werden. Man erhält Zykel $\zeta_0^-, \dots, \zeta_k^-$ zusätzlich zu den Zykeln $\zeta_0^+, \dots, \zeta_m^+$:



FVS-Algorithmus

- Für das Bestimmen eines FVS-Knotens nutzt man die Schnittmenge bestimmter Zykel ζ_i^+ und ζ_j^- :



Bemerkung:

- Algorithmus (theoretisch) in Linearzeit ausführbar, da für das Bestimmen der Zykel und ein Update der Daten nur lokale Knotenmengen betrachtet werden müssen.
- Berechnetes FVS hat höchstens die doppelte Kardinalität eines minimalen FVS für den Graphen G .

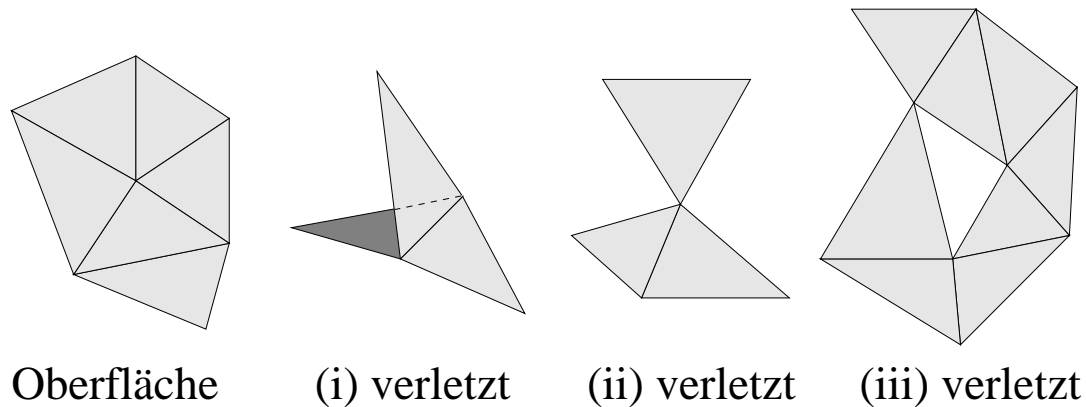
Anordnungen im \mathbb{R}^3

- Sei $\mathcal{T} = (V, E, F)$ Tetraedergitter mit Knoten $V(\mathcal{T})$, Kanten $E(\mathcal{T})$ und Dreiecksflächen $F(\mathcal{T})$.
- Problem: Im \mathbb{R}^3 keine planaren Graphen.
- Lösung: Konstruktion von Ebenen, die zumindest lokal Planarität erfüllen und Hauptinformation über den Fluß enthalten.

Oberflächen in \mathcal{T}

$\mathcal{S} \subseteq \mathcal{T}$ heißt *Oberfläche*, falls:

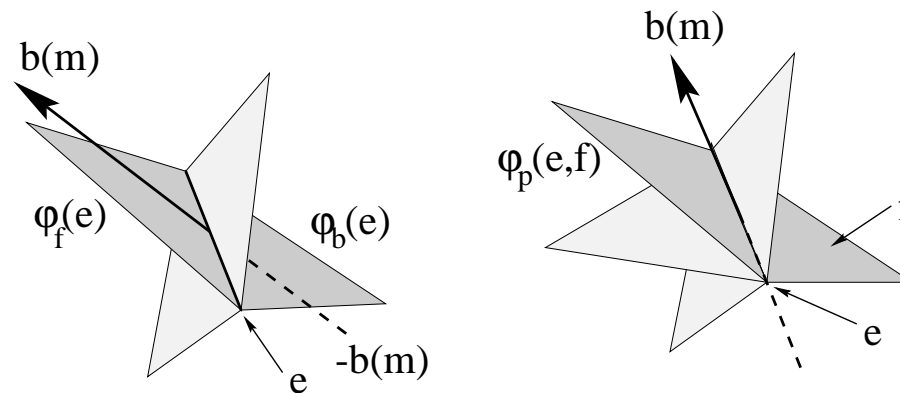
- i) Jede Kante $e \in E(\mathcal{S})$ mit höchstens zwei Flächen verbunden ist,
- ii) Je zwei Flächen $f, g \in F(\mathcal{S})$ stets durch einen Pfad von benachbarten Flächen verbunden sind.
- iii) Für zwei Flächen $f, g \in F(\mathcal{S})$ mit gemeinsamen Knoten $v \in V(\mathcal{S})$ existiert Pfad von benachbarten Flächen, die auch v enthalten.



- Zusatzforderung: Oberflächen sind orientierbar.

Flußoberflächen

- Sei $b : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ die Konvektionsrichtung.
- Betrachtet werden “starke” Kanten mit $\|b(m(e))\| > \sigma$ mit $m(e)$ Kantenmittelpunkt und $\sigma > 0$.
- Definition von Funktionen, die jeder Kante $e \in E(\mathcal{T})$ eine anliegende Fläche entsprechend Konvektionsrichtung im Kantenmittelpunkt zuordnen:
 - $\varphi_f(e)$ ist Fläche mit kleinstem Winkel zu $b(m)$.
 - $\varphi_b(e)$ ist Fläche mit kleinstem Winkel zu $-b(m)$.
 - für $e \parallel b(m)$: $\varphi_p(e, f)$ ist Fläche mit größtem Winkel zu f .



- Flußfunktion $\varphi : E(\mathcal{T}) \times F(\mathcal{T}) \rightarrow F(\mathcal{T})$:

$$\varphi(e, f) = \begin{cases} \varphi_p(e, f) & , & e \parallel b(m(e)) \\ \varphi_f(e) & , & e \not\parallel b(m(e)) \text{ und } \angle(\varphi_f(e), f) > \angle(\varphi_b(e), f) \\ \varphi_b(e) & , & \text{sonst} \end{cases}$$

- Oberfläche $\mathcal{S} \subseteq \mathcal{T}$ heißt **Flußoberfläche mit minimalem Winkel** α falls:
 - $F(\mathcal{S})$ nur Flächen entsprechend φ enthält und
 - für benachbarte $f, g \in F(\mathcal{S})$ gilt: $\angle(f, g) \geq \alpha$.

Konstruktion der Flußoberflächen

- Für jede noch nicht benutzte Kante $e \in E(\mathcal{T})$ mit $f = \varphi_f(e)$ wird eine Flußoberfläche \mathcal{S} konstruiert.

- Algorithmus für eine Flußoberfläche:

```

proc SURFACE(  $e, f, \mathcal{S}$  )
   $g := \varphi(e, f)$ ;
  if  $g \notin \mathcal{S}$  und  $g$  erfüllt (S1), (S2) und (FS) then
     $\mathcal{S} := \mathcal{S} \cup \{g\}$ ;
    for all Kanten  $e' \neq e$  in  $g$  do Surface( $e', g, \mathcal{S}$ );
  endif; end;

```

(S1) jede Kante $e' \neq e$ von g hat höchstens eine anliegende Fläche $g' \neq g$ in \mathcal{S} .

(S2) eine Kante $e' \neq e$ von g hat genau eine anliegende Fläche $g' \neq g$ in \mathcal{S} oder jeder Knoten $v \notin e$ von g ist nicht in \mathcal{S} .

(FS) Alle anliegenden Flächen haben einen Winkel $> \alpha$ zueinander.

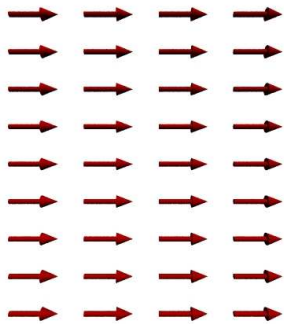
- Bedingungen S1, S2 und FS sind stärker als Definition von Flußoberfläche.
- Varianten des Algorithmus: verschiedene Möglichkeiten für den Durchlauf des Netzes, z.B. Tiefen- oder Breitensuche.

Numerische Beispiele

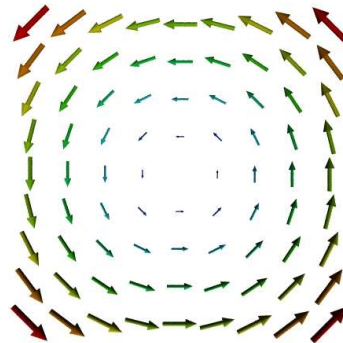
- Im folgenden gilt: $\Omega = (0, 1)^3$ und $\varepsilon = 10^{-5}$, Kantenmenge für Graphen ist E_3 mit $\kappa_2 = 0.2$.
- Verwendete Konvektionsrichtungen:

– 2D-Flüsse:

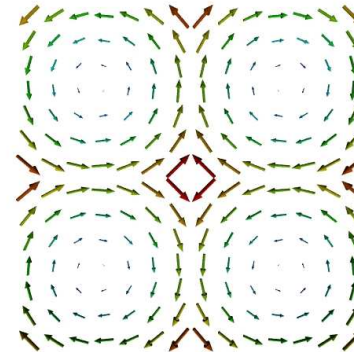
XLine



Circle



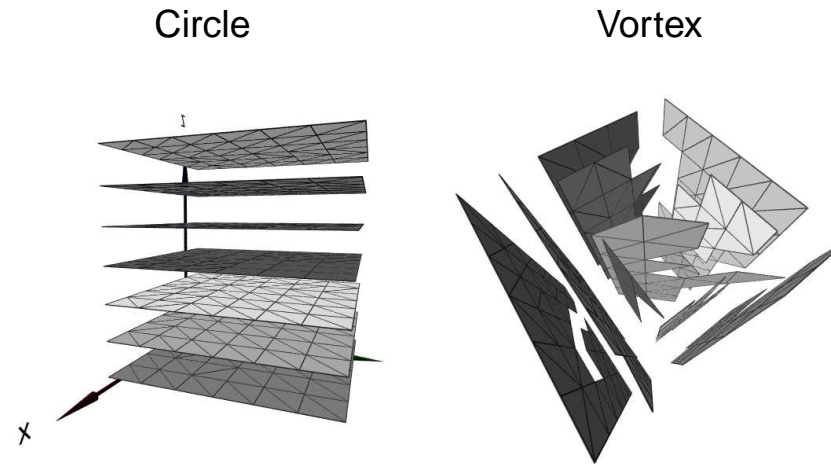
4 Circles



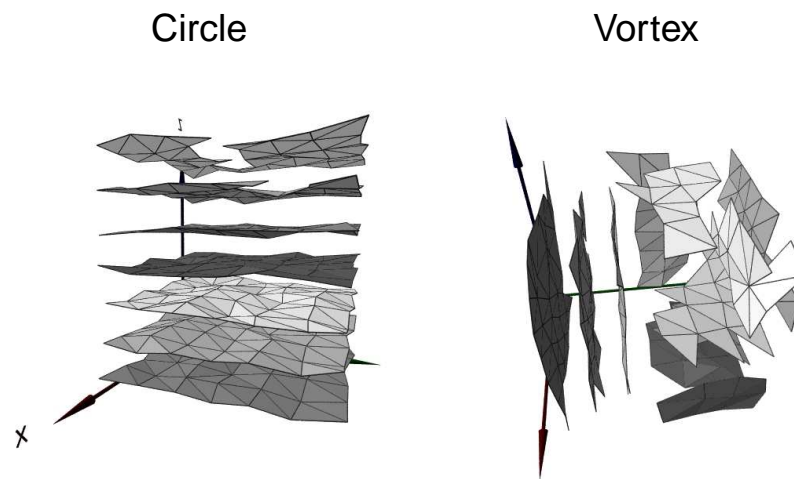
– 3D-Fluß: Vortex

Flußoberflächen

- Regelmäßiges Gitter:

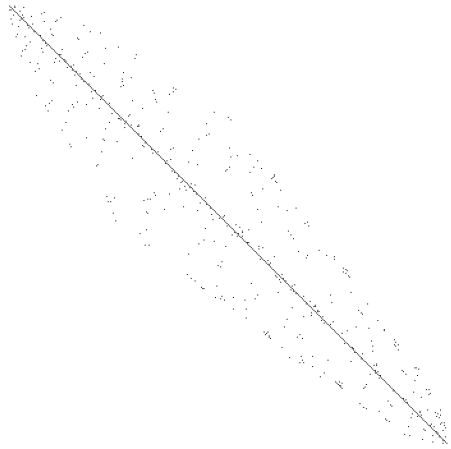


- Unregelmäßiges Gitter:

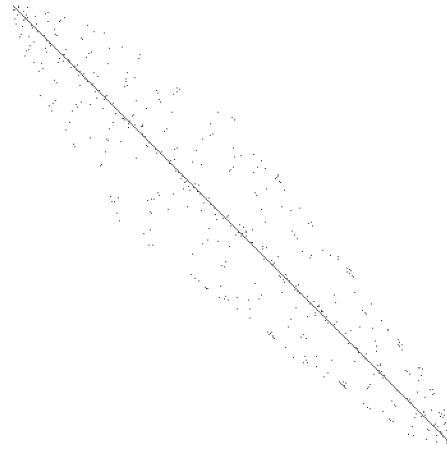


Beispielmatrizen

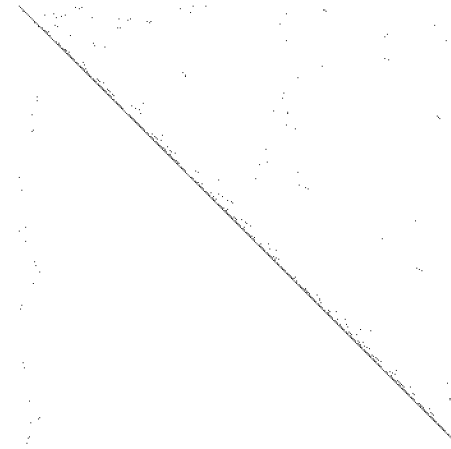
ohne Anordnung



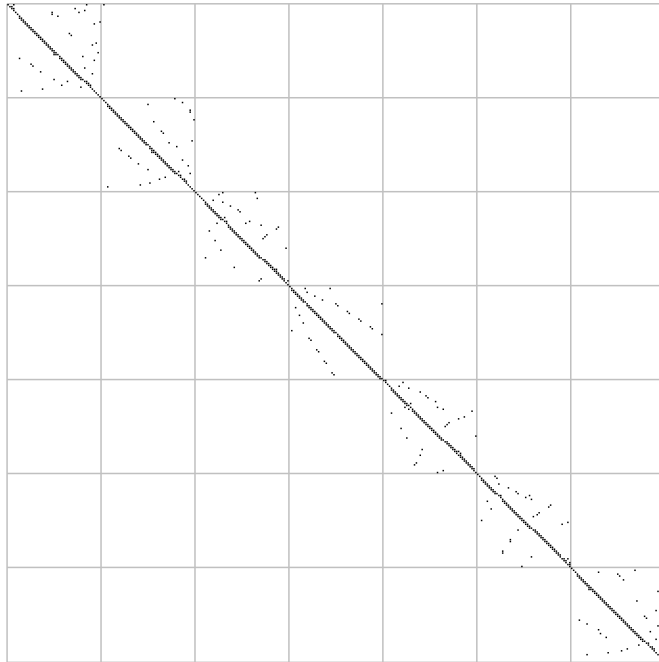
RCM



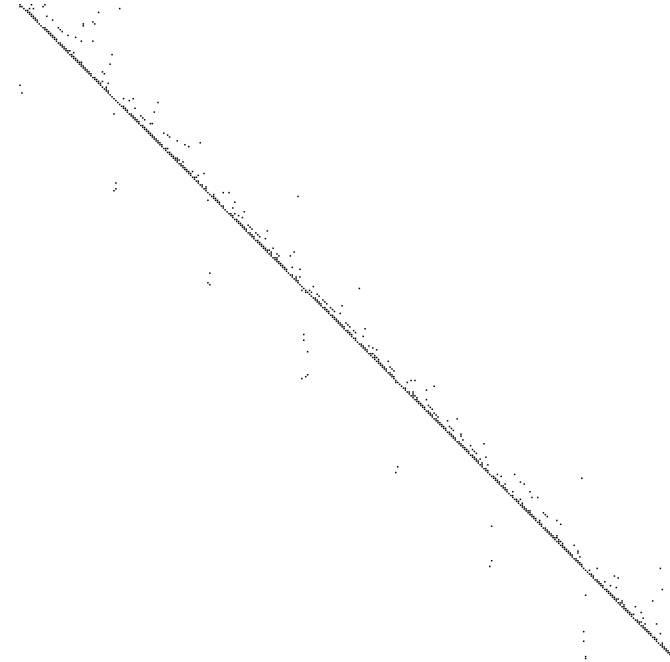
globales FVS



Zykel auf FS



FVS auf FS



Konvergenzraten

- Aufwand für Flußoberflächen:
- Löser:
 - symmetrisch Gauß-Seidel bei allen Verfahren ohne FVS
 - Rückwärts-Gauß-Seidel bei Verfahren mit FVS
- $h = \frac{1}{16}$, 3375 Unbekannte

Anordnung	XLine	Circle	4 Circles	Vortex
ohne	0.08	0.93 (361)	0.85 (177)	0.76 (103)
RCM	0.03	0.89 (228)	0.75 (98)	0.60 (55)
globales FVS	0.002	0.74 (95)	0.61 (57)	0.44 (35)
Zykel auf FS	0.002	0.85 (175)	0.73 (89)	0.56 (49)
FVS auf FS	0.003	0.73 (89)	0.55 (47)	0.62 (61)

$h = \frac{1}{32}$:	Anordnung	XLine	Circle	4 Circles	Vortex
	ohne	0.41 (18)	0.93 (361)	0.85 (177)	0.76 (103)
	RCM	0.30 (12)	0.89 (228)	0.75 (98)	0.60 (55)
	globales FVS	0.01 (3)	0.74 (95)	0.61 (57)	0.44 (35)
	Zykel auf FS	0.01 (3)	0.85 (175)	0.73 (89)	0.56 (49)
	FVS auf FS	0.01 (4)	0.73 (89)	0.55 (47)	0.62 (61)

$h = \frac{1}{16}$:	Anordnung	Circle	Vortex	Circle	Vortex
	ohne	0. ()	0. ()	0. ()	0. ()
	RCM	0. ()	0. ()	0. ()	0. ()
	globales FVS	0. ()	0. ()	0. ()	0. ()
	Zykel auf FS	0. ()	0. ()	0. ()	0. ()
	FVS auf FS	0. ()	0. ()	0. ()	0. ()

Offene Probleme

- Algorithmus für planares FVS auf Flußoberflächen,
- Anordnung der Flußoberflächen untereinander