Vorlesungsskript
Sommersemester 2001

Parallele FEM

Vorlesender:	**Prof. Arnd Meyer**
Erstellung des Skripts:	**René Schneider**
Zweitkorrektur:	**Peter Nestler**

Technische Universität Chemnitz
Fakultät für Mathematik

Inhaltsverzeichnis

1 Grundlagen

1.1 Problem: Parallel

Basis: Näherungsweise Lösung von Partiellen Differentialgleichungen mittels Diskretisierung (FEM)
Übliche Schritte:

1. definiere ein Netz
2. Aufstellen des zugehörigen linearen Gleichnungs Systems (LGS): $K\underline{u} = \underline{b}$
3. Lösen des LGS mit sinnvoller Genauigkeit
4. Auswertung

Dabei entstehen sehr große LGS $\Rightarrow$ man versucht das Problem durch Zerlegung des Rechengebietes in parallel bearbeitbare kleinere Probleme zu lösen.

1.2 FEM und iterative Löser

Beispiel: *Laplace Gleichung*

$$\begin{array}{lrcll} & -\Delta u & = & f \;|_{\Omega} & \Omega \subset \mathbb{R}^2 \\ \textit{Randbedingungen:} & u & = & g \;|_{\Gamma_D} & \textit{Dirichlet-Rand} \\ & \frac{\partial u}{\partial n} & = & 0 \;|_{\Gamma_N} & \textit{Neumann-Rand} \end{array}$$

Wobei: $\Gamma_D \cup \Gamma_N = \partial\Omega$
$\Rightarrow$ *Übliche Umwandlung in Bilinearform (Variationsformulierung)*
ges:

$$u \in H^1(\Omega) \textit{ mit } u = g \;|_{\Gamma_D} \textit{ und}$$

$$a(u,v) = \langle f, v\rangle \qquad \forall v \in H_0^1(\Omega) := \left\{ v \in H^1(\Omega) : \, v = 0 \;|_{\Gamma_D} \right\}$$

hier:

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, d\Omega,$$

$$\langle f, v\rangle = \int_\Omega f \cdot v \, d\Omega$$

Um zu einer Diskretisierung zu gelangen schränkt man sich auf einen Raum $\mathbb{V} \subset H^1(\Omega)$ ein. Um eine kürzere Notation zu erhalten sei die Bedingung $u = g$ auf dem Rand Γ_D in $\mathbb{V}$ erfüllbar und im weiteren für die Näherungen immer gefordert. Also:
ges:

$$u \in \mathbb{V}$$

$$\text{mit} \quad a(u,v) = \langle f, v\rangle \qquad \forall v \in \mathbb{V}_0 = \mathbb{V} \cap H_0^1(\Omega)$$

Man wählt dann eine Basis Φ in $\mathbb{V}$, z.B. stetige, stückweise lineare Funktionen φ_i mit Träger um einen Knoten einer Dreiecksvernetzung des Gebietes Ω, so das $\varphi_i(x_j) = \delta_{i,j}$.

Bezeichnungen:

$\Phi := [\varphi_1, \varphi_2, \ldots, \varphi_N]$ (Zeilenvektor der Basisfunktionen)

$\mathbb{V} := span(\Phi)$

$\Rightarrow u = \sum\limits_{i=1}^{N} u_i \varphi_i = \Phi \underline{u} \in \mathbb{V}$

$\underline{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^N$

$a(\Phi\underline{u}, \varphi_i) = \langle f, \varphi_i \rangle \qquad \forall i : \, x_i \notin \Gamma_D$

$\Leftrightarrow \sum\limits_{j=1}^{N} u_j a(\varphi_j, \varphi_i) = \langle f, \varphi_i \rangle \qquad \forall i : \, x_i \notin \Gamma_D$

$u_j = g(x_j) \qquad \forall j : \, x_j \in \Gamma_D$

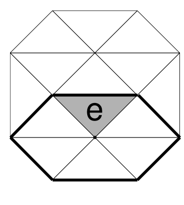
$\Longrightarrow$ def:

$$\begin{aligned} b_i &:= \langle f, \varphi_i \rangle \qquad \forall i \\ K &:= [k_{i,j}]_{i,j=1}^{N} \\ k_{i,j} &:= a(\varphi_j, \varphi_i) \end{aligned}$$

$\Longrightarrow K\underline{u} = \underline{b}$

Bemerkung 1. *Einige Zeilen i ($x_i \in \Gamma_D$) dieses Gleichungsystems müßten eigentlich weggelassen werden, und die u_j, die zu den auf dem Dirichletrand liegenden Knoten gehören, durch die Entsprechung der Dirichlet Randbedingung ersetzt werden. Dies wird hier im Folgenden nicht berücksichtigt, da zum Lösen des LGS Iterationsverfahren genutzt werden, und man somit nur den Startvektor $\underline{u}^{(0)}$ richtig setzten muss und dafür gesorgt werden muss, daß die Werte, die zu den Dirichlet Knoten gehören, nicht mehr verändert werden.*

Bemerkung 2. *Übliche Realisierung der Berechnung von K und $\underline{b}$*
K und $\underline{b}$ werden elementweise berechnet. Sei $N_i^e(x)$ Einschränkung von φ_i auf das Element e.



$$\Longrightarrow \quad a(\varphi_j, \varphi_i) = \sum_e \int_e \left(\nabla N_j^e \cdot \nabla N_i^e \right) \, d\Omega$$

Also wird für jedes Element (mit $K := 0$ beginnend) folgendes getan:

1. *Berechne Elementmatrix $K_e := \left[a(N_j^e, N_i^e)\right]_{i,j=1}^{N_e}$, z.B. durch Quadraturformel. („Generieren“)*
2. $K_{i,j} = K_{i,j} + K_{e;i,j} \qquad \forall i,j$, *(„Assemblieren“)*

Bemerkung 3. *Beziehung FEM-System zu Repräsentation im PCG*

- *Üblich zur Lösung von $K\underline{u} = \underline{b}$ sind Iterationsverfahren, z.B. PCG (Preconditioned Conjugate Gradient method)*
  Dieses Verfahren wird üblicherweise im $\mathbb{R}^N$ betrachtet, man kann es jedoch auch im Raum $\mathbb{V}_0$ beschreiben.
  Algorithmus 1. *PCG*
  <u>Start:</u>
  $$\begin{array}{lcll} \underline{u} & & & \textit{Startvektor} \\ \underline{r} & = & K\underline{u} - \underline{b} & \textit{Residuum} \\ \underline{w} & = & C^{-1}\underline{r} & \textit{Vorkonditioniertes Residuum} \\ \underline{q} & = & \underline{w} & \textit{Suchrichtung} \\ \gamma & = & \underline{w}^T\underline{r} & \end{array}$$
  <u>Iteration:</u>
  $$\begin{array}{lcl} \underline{v} & := & K\underline{q} \\ \delta & := & \underline{v}^T\underline{q} \\ \alpha & := & -\gamma/\delta \\ \underline{u} & := & \underline{u} + \alpha\underline{q} \\ \underline{r} & := & \underline{r} + \alpha\underline{v} \\ \underline{w} & := & C^{-1}\underline{r} \\ \hat{\gamma} & := & \underline{w}^T\underline{r} \\ \beta & := & \hat{\gamma}/\gamma \\ \underline{q} & := & \underline{w} + \beta\underline{q} \\ \gamma & := & \hat{\gamma} \end{array}$$
  $\Longrightarrow$ *Beziehungen $\mathbb{R}^N$ zu $\mathbb{V}$*
- $\underline{u} \in \mathbb{R}^N \longleftrightarrow u = \Phi\underline{u}$

- *für das Residuum gilt*

$$\begin{aligned}
\underline{r} &= K\underline{u} - \underline{b} \\
&= \left[\sum_j a(\varphi_j, \varphi_i) u_j - \langle f, \varphi_i \rangle \right]_{i=1}^N \\
&= \left[a(\Phi\underline{u}, \varphi_i) - \langle f, \varphi_i \rangle \right]_{i=1}^N \\
&= \left[a(u, \varphi_i) - \langle f, \varphi_i \rangle \right]_{i=1}^N \\
&= \left[\langle \boldsymbol{r}, \varphi_i \rangle \right]_{i=1}^N
\end{aligned}$$

d.h. für alle $u \in \mathbb{V}$ *existiert das Residuenfunktional* $\boldsymbol{r} \in \mathbb{V}^*$ *(oder* $H^{-1}(\Omega)$*), es ist definiert durch:*

$$\langle \boldsymbol{r}, v \rangle = a(u, v) - \langle f, v \rangle \qquad \forall v \in \mathbb{V}_0 \quad (\textit{oder sogar}\ \forall v \in H_0^1(\Omega))$$

wir erhalten $(<)N$ *Funktionalwerte:* $\langle \boldsymbol{r}, \varphi_i \rangle$ *im Vektor* $\underline{r}$

im Gegensatz dazu:

- $\underline{w}$ *repräsentiert eine Funktion* $w := \Phi\underline{w} \in \mathbb{V}_0$
- $\underline{q}$ *repräsentiert eine Funktion* $q := \Phi\underline{q} \in \mathbb{V}_0$
(ist sinnvoll da: $\underline{u} = \underline{u} + \alpha\underline{q} \quad \Longleftrightarrow \quad u = u + \alpha q$*)*

- *interessant ist auch noch:* $\underline{w} = C^{-1}\underline{r}$
  a. *sei* $C = I$*:*
  $w = \Phi\underline{w} = \Phi\underline{r} = \sum\limits_{i=1}^N \varphi_i \langle \boldsymbol{r}, \varphi_i \rangle$
  b. *C beliebig:*
  Sei $\Psi = [\psi_1, \ldots, \psi_N]$ *eine* *<u>andere</u>* *Basis in* $\mathbb{V}_0 \subset \mathbb{V} = span(\Phi)$
  $\Rightarrow w = \sum\limits_{i=1}^N \psi_i \langle \boldsymbol{r}, \psi_i \rangle$
  also ist Ψ *anbhängig von* C*.*
  Basiswechsel $\Rightarrow \quad \exists Q \in \mathbb{R}^{N \times N}$ *regulär:* $\Psi = \Phi Q$*.*
  betrachten:

$$\begin{bmatrix} \langle \boldsymbol{r}, \psi_1 \rangle \\ \vdots \\ \langle \boldsymbol{r}, \psi_N \rangle \end{bmatrix} = \begin{bmatrix} \langle \boldsymbol{r}, \Phi q_1 \rangle \\ \vdots \\ \langle \boldsymbol{r}, \Phi q_N \rangle \end{bmatrix} = \begin{bmatrix} \sum_i \langle \boldsymbol{r}, \varphi_i \rangle q_{i,1} \\ \vdots \\ \sum_i \langle \boldsymbol{r}, \varphi_i \rangle q_{i,N} \end{bmatrix} = Q^T \underline{r}$$

$$\begin{aligned}
\Longrightarrow \quad w &= \sum \psi_i \langle \boldsymbol{r}, \psi_i \rangle &&= \Psi \begin{bmatrix} \langle \boldsymbol{r}, \psi_1 \rangle \\ \vdots \\ \langle \boldsymbol{r}, \psi_N \rangle \end{bmatrix} \\
&= \Psi Q^T \underline{r} &&= \Phi Q Q^T \underline{r} \\
&= \Phi \underline{w} \\
\Longleftrightarrow \quad \underline{w} &= Q Q^T \underline{r} &&= C^{-1} \underline{r}
\end{aligned}$$

1.3 Gundlegende Gebietszerlegung (DD) und ihre Wirkung

DD - Domain Decomposition - Gebietszerlegung

Problemstellung:

- haben Gebiet $\Omega \subset \mathbb{R}^d$
- gesucht ist $u \in \mathbb{V}$ mit $a(u, v) = \langle f, v \rangle \qquad \forall v \in \mathbb{V}_0$

Ansatz:

- ○ definieren Teilgebiete Ω_s mit:

  $\overline{\Omega} = \bigcup\limits_{s=1}^{p} \overline{\Omega_s}$

  $\Omega_{s_1} \cap \Omega_{s_2} = \emptyset \qquad \forall s_1 \neq s_2$ (d.h. nichtüberlappende Teilgebiete)
- ○ andere Bezeichnungen für die Ω_s sind z.B.: Subdomain, „Superelement“, „Substruktur“
- ○ $\Gamma_c := \bigcup\limits_{s} \partial\Omega_s$ wird Koppelrand genannt
- ○ fordern konforme Vernetzung [1] in jedem Teilgebiet, und daß die Vernetzung über den Koppelrand Γ_c hinweg konform ist
- ○ fordern weiter, daß jedes Element e genau einem Teilgebiet Ω_s zugeordnet ist

  $\Longrightarrow$

$$\begin{aligned}
a(u, v) &= \int\limits_{\Omega} \nabla u \cdot \nabla v \, d\Omega \\
&= \sum_s \int\limits_{\Omega_s} \nabla u \cdot \nabla v \, d\Omega \\
&= \sum_s a_s(u, v)
\end{aligned}$$

[1] konforme Vernetzung $\Rightarrow$ keine „hängenden“ Knoten

- führen besondere Knotennummerierung ein:

$$\Big(\underbrace{x_1,\dots,x_{N_c}}_{\in\,\Gamma_c} \;\Big|\; \underbrace{x_{N_c+1},\dots,x_{N_c+N_{I,1}}}_{\in\,\mathrm{int}(\Omega_1)} \;\Big|\; \dots \;\Big|\; \underbrace{\dots,x_N}_{\in\,\mathrm{int}(\Omega_p)} \Big)$$

wobei: $N_c \;\hat{=}\;$ Anzahl Knoten auf dem Koppelrand Γ_c
$N_{I,s} \;\hat{=}\;$ Anzahl Knoten im Inneren von Ω_s

$\Longrightarrow \quad N = N_c + \sum\limits_{s=1}^{p} N_{I,s}$

- die Knotenbasis wird analog nummeriert

$$\mathbb{V} = \mathrm{span}(\Phi) \qquad \text{„zerfällt“ in Teilräume, z.B.:}$$

$$\begin{aligned} \mathbb{V}_{I,1} &= \mathrm{span}(\varphi_{N_c+1},\dots,\varphi_{N_c+N_{I,1}}) \\ \mathbb{V}_I &= \mathrm{span}(\varphi_{N_c+1},\dots,\varphi_N) \\ &= \mathbb{V}_{I,1} \dot{+} \mathbb{V}_{I,2} \dot{+} \dots \dot{+} \mathbb{V}_{I,p} \quad {}^{2} \\ \mathbb{V}_c &= \mathrm{span}(\varphi_1,\dots,\varphi_{N_c}) \end{aligned}$$

$\mathbb{V}_{I,s}$ sind somit Funktionen mit Träger $\subseteq \mathrm{int}(\Omega_s)$, d.h.

$$v(x) \equiv 0 \qquad \forall x \notin \Omega_s \qquad \forall v \in \mathbb{V}_{I,s}$$

- Schreibweise:

$$I_{N\times N} \;=:\; \Big(\underbrace{V_c}_{N_c} \;\Big|\; \underbrace{V_I}_{N_I} \Big) = \left[\begin{array}{c|c} I_{N_c\times N_c} & 0 \\ \hline 0 & I_{N_I\times N_I} \end{array}\right]$$

Spalten

Damit und mit analoger Schreibweise ergibt sich:

$$\begin{aligned} \mathbb{V}_c &= \mathrm{span}(\Phi V_c) \\ \mathbb{V}_I &= \mathrm{span}(\Phi V_I) \\ \mathbb{V}_{I,s} &= \mathrm{span}(\Phi V_{I,s}) \end{aligned}$$

- somit kommt man auf die <u>Idee</u>:
Der Prozessor P_s hält sämtliche Informationen über $\overline{\Omega_s}$ (Geometrie, Knoten, finite Elemente, Funktionswerte ...) und bearbeitet alles was sich auf $\overline{\Omega_s}$ bezieht.

$u \in \mathbb{V}: \quad u = \Phi\underline{u} \qquad \underline{u} \in \mathbb{R}^N$

definieren:

N_s	Anzahl der Knoten/Freiheitsgrade in $\overline{\Omega_s}$
$N_{c,s}$	Anzahl der Knoten auf $\partial\Omega_s$
$\underline{u}_s \in \mathbb{R}^{N_s}$	entspricht der Funktion eingeschränkt auf $\overline{\Omega_s}$
$H_s \in \mathbb{R}^{N_s\times N}$	als Boolesche Matrix[3], so daß $\underline{u}_s = H_s\underline{u}$

[2] „$\dot{+}$“ - bezeichnet hier die direkte Summe
[3] Boolsche Matrix - exakt eine 1 pro Zeile, Rest 0

Folgerungen: $\begin{aligned} N_c &\leq \sum N_{c,s} \\ N_s &= N_{I,s} + N_{c,s} \end{aligned}$

- betrachten nun Darstellung von K und $\underline{b}$ unter DD - Gegebenheiten, d.h. im Prozessor P_s (wo nur Daten zu $\overline{\Omega_s}$ gegeben sind):

  berechenbar sind somit:

$K_s \in \mathbb{R}^{N_s \times N_s}$	Assemblierung der Elementmatrizen $\forall e \in \Omega_s$
$\underline{b}_s \in \mathbb{R}^{N_s}$	Assemblierung der Element-rechte Seiten $\forall e \in \Omega_s$

  Wir suchen also die Beziehungen zwischen $K, \underline{b}$ und $K_s, \underline{b}_s$:

  $K_s \in \mathbb{R}^{N_s \times N_S} \longleftarrow a_s(\varphi_j, \varphi_i)$

  $K \in \mathbb{R}^{N \times N} \longleftarrow a(\varphi_j, \varphi_i) = \sum\limits_s a_s(\varphi_j, \varphi_i)$

  $\Longrightarrow$

$$\begin{aligned} K &= \sum_{s=1}^{p} H_s^T K_s H_s \\ \underline{b} &= \sum_{s=1}^{p} H_s^T \underline{b}_s \end{aligned}$$

- **<u>Fazit</u>**:
  Es gibt zwei Typen von Vektoren in der DD - Darstellung:

  1. $\underline{u}$ ist Vektor von <u>Funktions</u>werten (zu $u \in \mathbb{V}$)
     $\Rightarrow \quad \underline{u}_s = H_s \underline{u}$
  2. $\underline{b}, \underline{r}$ sind Vektoren mit <u>Funtional</u>werten
     $\Rightarrow \quad \underline{b} = \sum H_s^T \underline{b}_s$

- Folgerung:

$$\begin{aligned} \underline{v} &= K\underline{u} \\ &= \sum H_s^T K_s (H_s \underline{u}) \\ &= \sum H_s^T (K_s \underline{u}_s) \\ &= \sum H_s^T \underline{v}_s \end{aligned}$$

  mit $\underline{v}_s = K_s \underline{u}_s$. Das Ergebnis kann also lokal (vom Prozessor P_s) berechnet werden, und ist ein Vektor des 2. Typs.
  Analog:

$$\begin{aligned} \underline{r} &= K\underline{u} - \underline{b} \\ &= \sum H_s^T (K_s \underline{u}_s - \underline{b}_s) \\ &= \sum H_s^T \underline{r}_s \\ \underline{r}_s &= K_s \underline{u}_s - \underline{b}_s \end{aligned}$$

- Parallele Realisierung des PCG:
  - Vektor $\underline{w}$ aus Algorithmus 1 ist vom Typ 1.
  - $\underline{r}$ ist vom 2. Typ
  - $\underline{q}$ ist vom 1. Typ, und somit
  - $K\underline{q}$ ist vom 2. Typ.

  d.h.:

$$\begin{aligned}\underline{w}^T\underline{r} &= \underline{w}^T(\sum H_s^T\underline{r}_s)\\ &= \sum(H_s\underline{w})^T\underline{r}_s\\ &= \sum_{s=1}^{p}(\underline{w}_s^T\underline{r}_s)\end{aligned}$$

  das Skalarprodukt kann also lokal mit anschliesender globaler Summation berechnet werden. Somit ist der Kommunikationsaufwand gering.
  ähnlich verhält es sich bei:

$$\underline{q}^T K\underline{q} = \underline{q}^T\left(\sum H_s^T(K_s\underline{q}_s)\right) = \sum \underline{q}_s^T K_s\underline{q}_s$$

  wodurch man δ berechnen kann

Algorithmus 2. *parallel PCG (als Programm im Prozessor P_s)*
Start:

$\underline{u}$			*Startvektor*
$\underline{u}_s$	$=$	$H_s\underline{u}$	*lokaler Startvektor*
$\underline{r}_s$	$=$	$K_s\underline{u}_s - \underline{b}_s$	*lokales Residuum*
$\underline{r}$	$=$	$\sum r_s$	*Residuum*
$\underline{w}_s$	$=$	$H_s\underline{w}$	*lokaler Teil des Vorkonditionierten Residuums ($C\underline{w} = \underline{r}$)*
$\underline{q}_s$	$=$	$\underline{w}_s$	*Suchrichtung*
γ_s	$:=$	$\underline{w}_s^T\underline{r}_s$	
γ	$=$	$\sum\gamma_s$	

Iteration:

$$
\begin{array}{lrcl}
\textit{1.} & \underline{v}_s & := & K_s \underline{q}_s \\
\textit{2.} & \delta_s & := & \underline{v}_s^T \underline{q}_s \\
\textit{3.} & \delta & := & \sum \delta_s \\
\textit{4.} & \alpha & := & -\gamma/\delta \\
\textit{5.} & \underline{u}_s & := & \underline{u}_s + \alpha \underline{q}_s \\
\textit{6.} & \underline{r}_s & := & \underline{r}_s + \alpha \underline{v}_s \\
\textbf{7.} & \underline{w}_s & & \textit{aus } C\underline{w} = \underline{r} \textit{ berechnen} \\
\textit{8.} & \hat{\gamma}_s & := & \underline{w}_s^T \underline{r}_s \\
\textit{9.} & \hat{\gamma} & := & \sum \hat{\gamma}_s \\
\textit{10.} & \beta & := & \hat{\gamma}/\gamma \\
\textit{11.} & \underline{q}_s & := & \underline{w}_s + \beta \underline{q}_s \\
\textit{12.} & \gamma & := & \hat{\gamma}
\end{array}
$$

Bemerkung 4. *Bei den Schritten 3. und 9. tritt ein geringer Kommunikationsaufwand auf (globale Summierung eines Skalars). Aber bei Schritt 7. kann der Kommunikationsaufwand sehr groß werden, z.B. wenn $C = I$:*

$$
\underline{w}_s = H_s \underline{w} = H_s \underline{r} = H_s \left(\sum_{j=1}^{p} H_j^T \underline{r}_j \right) = \underline{r}_s + \sum_{j \neq s} H_s H_j^T \underline{r}_j
$$

Hierbei definiert $H_s H_j^T \underline{r}_j$ den Teil der Daten von Prozessor P_j der zu Prozessor P_s gesendet werden muß, um $\underline{w}_s$ zu erzeugen.

Die Übrigen Schritte des Algorithmus arbeiten alle nur mit lokalen Daten (d.h. schnell).

2 FEM - Substrukturtechnik und ihre Parallelisierung

(seit Anfang der '60er Jahre von Ingenieuren benutzt)

2.1 FE - Substrukturtechnik

Wir betrachten zunächst DD-Datensplitting:

$\overline{\Omega} = \bigcup_s \overline{\Omega}_s$	-	Gebietszerlegung
Γ_c	-	Koppelrand
$\mathbb{V}_{I,s}$	-	Funktionenräume zum jeweiligen Inneren der Teilgebiete

mit der Nummerierung:

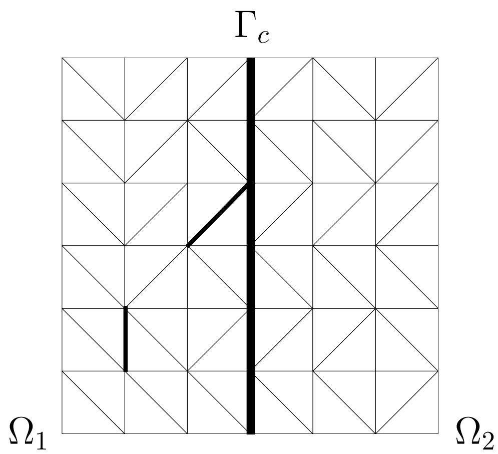
$$\Phi = (\underbrace{\varphi_1, \dots, \varphi_{N_c},}_{\Gamma_c} \underbrace{\varphi_{N_c+1}, \dots, \varphi_{N_c+N_{I,1}},}_{\Omega_1} \dots)$$

Dadurch läst sich dann die Matrix K darstellen als:

$$K = [a(\varphi_j, \varphi_i)]_{i,j=1}^N = \begin{bmatrix} K_c & K_{cI} \\ K_{Ic} & K_I \end{bmatrix}$$

mit den Bezeichnungen:

$$\begin{aligned} K_c = [a(\varphi_j, \varphi_i)]_{i,j=1}^{N_c} &\in \mathbb{R}^{N_c \times N_c} \\ K_I &\in \mathbb{R}^{N_I \times N_I} \\ K_{Ic} &\in \mathbb{R}^{N_I \times N_c} \\ K_{cI} &\in \mathbb{R}^{N_c \times N_I} \end{aligned}$$



Bemerkung 5. *Falls die Bilinearform $a(.,.)$ symmetrisch ist, so gilt:*

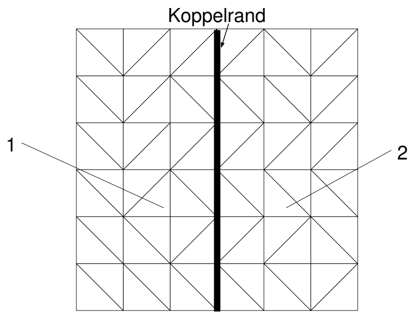
$$K_{Ic} = (K_{cI})^T$$

Bemerkung 6. *Die Matrix K_I ist blockdiagonal mit:*

$$K_I = diag(K_{I,1}, K_{I,2}, \ldots, K_{I,p})$$

Beispiel:

a. *$p = 2$ und auf dem gesammten Rand Dirichlet-Randbedingungen.*



$$K = \begin{bmatrix} K_c & K_{cI,1} & K_{cI,2} \\ K_{Ic,1} & K_{I,1} & 0 \\ K_{Ic,2} & 0 & K_{I,2} \end{bmatrix}$$

Mit $K = \sum H_s^T K_s H_s$ folgt weiter:

$$K_c = K_{c,1} + K_{c,2}$$

b. *sonst*

$$H_s \quad = \quad \begin{array}{cc} \begin{bmatrix} H_{c,s} & 0 & \ldots & 0 & 0 & 0\ldots \\ 0 & 0 & \ldots & 0 & I & 0\ldots \end{bmatrix} & \begin{array}{c} N_{c,s} \\ N_{I,s} \end{array} \\ \begin{array}{cccccc} N_c & N_{I,1}\ldots & & & N_{I,s} & \ldots \end{array} & \end{array}$$

Dabei ergibt sich die Matrix $H_{c,s}$ analog zu dem obigen Fall $p = 2$. Daraus folgt:

$$K_c \quad = \quad \sum_{s=1}^{p} H_{c,s}^T K_{c,s} H_{c,s}$$

$$K_{Ic} \quad = \quad \begin{bmatrix} K_{Ic,1} \cdot H_{c,1} \\ \vdots \\ K_{Ic,p} \cdot H_{c,p} \end{bmatrix}$$

Nun „zerfällt“ die Gleichung

$$K\underline{u} = \underline{b}$$

in:

$$\begin{bmatrix} K_c & K_{cI} \\ K_{Ic} & K_I \end{bmatrix} \begin{bmatrix} \underline{u}_c \\ \underline{u}_I \end{bmatrix} = \begin{bmatrix} \underline{b}_c \\ \underline{b}_I \end{bmatrix} \tag{1}$$

mit den Bezeichnungen:

$$\begin{aligned}
\underline{b}_c &= \sum H_{c,s}\underline{b}_{c,s} \\
\underline{b}_I &= \begin{bmatrix} \underline{b}_{I,1} \\ \vdots \\ \underline{b}_{I,p} \end{bmatrix}
\end{aligned}$$

Aus der Gleichung (1) erhält man durch Umformen:

$$\begin{aligned}
\underline{u}_I &= K_I^{-1}\left(\underline{b}_I - K_{Ic}\underline{u}_c\right) \\
K_c\underline{u}_c &= \underline{b}_c - K_{cI}K_I^{-1}\left(\underline{b}_I - K_{Ic}\underline{u}_c\right) \\
K_c\underline{u}_c &= \underline{b}_c - K_{cI}K_I^{-1}\underline{b}_I + K_{cI}K_I^{-1}K_{Ic}\underline{u}_c \\
\left(K_c - K_{cI}K_I^{-1}K_{Ic}\right)\underline{u}_c &= \underline{b}_c - K_{cI}K_I^{-1}\underline{b}_I \\
S_c\underline{u}_c &= \widetilde{\underline{b}}_c
\end{aligned}$$

mit den neuen Bezeichnungen:

$$\begin{aligned}
S_c &= K_c - K_{cI}K_I^{-1}K_{Ic} \\
\widetilde{\underline{b}}_c &= \underline{b}_c - K_{cI}K_I^{-1}\underline{b}_I
\end{aligned}$$

S_c wird Schur-Komplement genannt.
Aus diesen Gleichungen entstehen drei Techniken zur Lösung...

2.2 Drei Techniken zur Lösung

2.2.1 Technik I: mit Assemblieren von S_c

Motiviert durch die Gleichung

$$S_c = \sum_s H_{c,s}^T \left(K_{c,s} - K_{cI,s}K_{I,s}^{-1}K_{Ic,s}\right) H_{c,s}$$

wird folgendes Schema abgearbeitet:

1. *parallel* auf jedem Prozessor s

   - berechne K_s (d.h. $= K_{c,s}, K_{cI,s}, K_{Ic,s}, K_{I,s}$), $\underline{b}_{c,s}$ und $\underline{b}_{I,s}$
   - löse $K_{I,s}W_{Ic,s} = K_{Ic,s}$
   - löse $K_{I,s}v_{I,s} = \underline{b}_{I,s}$
   - $S_{c,s} = K_{c,s} - K_{cI,s}W_{Ic,s}$
   - $\widetilde{\underline{b}}_{c,s} = \underline{b}_{c,s} - K_{cI,s}v_{I,s}$

2. *Kommunikation*

- assembliere $S_c = \sum_s H_{c,s}^T S_{c,s} H_{c,s}$
- assembliere $\underline{\widetilde{b}}_c = \sum_s H_{c,s}^T \underline{\widetilde{b}}_{c,s}$

3. *1 Prozessor*

   - löse $S_c \underline{u}_c = \underline{\widetilde{b}}_c$
   - sende $\underline{u}_c$ an alle Prozessoren $\Rightarrow \quad \underline{u}_{c,s}$

4. *parallel* auf jedem Prozessor s

   - löse $K_{I,s} \underline{u}_{I,s} = \underline{b}_{I,s} - K_{Ic,s} \underline{u}_{c,s}$

Bemerkung 7. *Zum Aufwand:*

Schritt		*2D*	*3D*
	Anzahl Unbekannter (N)	$\mathcal{O}(h^{-2})$	$\mathcal{O}(h^{-3})$
1.	*Cholesky (als direkter Löser) für $K_{I,s}$*	$\mathcal{O}(h^{-4})$	$\mathcal{O}(h^{-7})$
1.	*$N_{c,s}$ Stück LGS lösen*	$\mathcal{O}(h^{-1}h^{-3})$	$\mathcal{O}(h^{-2}h^{-5})$
2.	*Datenaufkommen (Kommunikation)*	$\mathcal{O}(h^{-2})$	$\mathcal{O}(h^{-4})$
3.	*Direkter Löser (da S_c fast voll besetzt)*	$\mathcal{O}(h^{-3})$	$\mathcal{O}(h^{-6})$

Also ist das wohl nicht der optimale Weg.

2.2.2 Technik II: ohne Assemblierung von S_c

Grundlegende Idee hier ist, einen iterativen Löser für $S_c \underline{u}_c = \underline{\widetilde{b}}_c$ zu verwenden, der mit (nur) Multiplikationen auskommt.

$$
\begin{aligned}
S_c q_c &= \sum_s H_{c,s}^T \left(S_{c,s} q_{c,s}\right) \\
S_{c,s} q_{c,s} &= K_{c,s} q_{c,s} - K_{cI,s} \left(K_{I,s}^{-1} K_{Ic,s} q_{c,s}\right)
\end{aligned}
$$

<u>Vorteile:</u>

1. geringeres Datenaufkommen, nur noch $\mathcal{O}(h^{-1})$ bei 2D und $\mathcal{O}(h^{-2})$ bei 3D
2. nur ein LGS $\quad K_{I,s} v_{I,s} = K_{Ic,s} q_{c,s} \quad$ statt $N_{c,s}$ Stück
3. PCG wird für das LGS $\quad S_c \underline{u}_c = \underline{\widetilde{b}}_c \quad$ benutzt und $\kappa(S_c) \ll \kappa(K)$
   genauer gilt für Partielle Differentialgleichungen zweiter Ordnung (bei üblicher FEM) $\kappa(K) = \mathcal{O}(h^{-2})$ und $\kappa(S_c) = \mathcal{O}(h^{-1})$.

Somit ergibt sich also folgendes Schema:

1. *parallel* auf jedem Prozessor s

   - berechne $K_{c,s}$, $K_{cI,s}$, $K_{Ic,s}$, $K_{I,s}$, $\underline{b}_{c,s}$ und $\underline{b}_{I,s}$

- faktorisiere $K_{I,s} = R_{I,s} R_{I,s}^T$ z.B. mit Cholesky
- berechne $\underline{\widetilde{b}}_{c,s} = \underline{b}_{c,s} - K_{cI,s} \left(K_{I,s}^{-1} \underline{b}_{I,s} \right)$
- setze Startvektor $\underline{u}_{c,s}^{(0)}$, dabei muß gelten $\underline{u}_{c,s}^{(0)} = H_{c,s} \underline{u}_c^{(0)}$

2. *paralleles PCG* (Alg. 2) für $\quad S_c \underline{u}_c = \underline{\widetilde{b}}_c$

   - möglich durch

$$S_c = \sum_s H_{c,s}^T S_{c,s} H_{c,s} \qquad \underline{\widetilde{b}}_c = \sum_s H_{c,s}^T \underline{\widetilde{b}}_{c,s}$$

   - ist weitestgehend parallel
   - Vorkonditionierer $C_c \sim S_c \quad$ möglich?

3. *parallel* auf jedem Prozessor s

   - löse $K_{I,s} \underline{u}_{I,s} = \underline{b}_{I,s} - K_{Ic,s} \underline{u}_{c,s}$

Wenn man also noch einen guten Vorkonditionierer für S_c hätte wäre dieses Schema zufriedenstellend. Also machen wir uns ein paar Gedanken darüber.

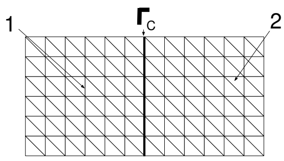
2.2.3 Schur-Komplement Vorkonditionierung

Zur Analyse von $\sigma(S_c)$ [4] und $\sigma(C_c^{-1} S_c)$ bedienen wir uns eines Modellproblems, der Lapalace Gleichung (Bsp. 1.2):

$$\begin{array}{lrcll} & -\Delta u & = & f & |_{\Omega} \quad \Omega \subset \mathbb{R}^2 \text{ ein rechteckiges Gebiet} \\ \text{Randbedingung:} & u & = & 0 & |_{\partial\Omega} \end{array}$$

DD:	in zwei gleichgroße wiederum rechteckige Teilgebiete
FE:	jeweils ein äquidistantes Rechteckgitter mit $m \times N_c$ Gitterpunkten, die Gitter - Rechtecke ergeben je zwei rechtwinklige Dreiecke gleicher Größe

z.B. $N_c = 5$ Zeilen, $m = 5$ Spalten



Numerierung:

1. auf Γ_c

[4] $\sigma(.)$ bezeichnet hier das Spektrum einer Matrix

2. in Ω_s parallel zu Γ_c, dann weg von Γ_c

Damit ergibt sich:

$$
\begin{aligned}
K_1 &= K_2 \\
K_s &= \begin{bmatrix} K_{c,s} & K_{cI,s} \\ K_{Ic,s} & K_{I,s} \end{bmatrix} \qquad\qquad s = 1,2 \\
S_c &= K_c - 2K_{cI,1}K_{I,1}^{-1}K_{Ic,1} \\
K_c &= 2K_{c,1} \\
&= tridiag(-1,4,-1) \\
&= 2I_{N_c} + T_{N_c} \\
T_{N_c} &:= tridiag(-1,2,-1) \\
K_{Ic,s} &= \begin{bmatrix} -I_{N_c} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&= -e_m \otimes I_{N_c} \\
K_{I,s} &= \begin{bmatrix} \boxed{\text{-1,4,-1}} & -I & & \\ -I & \boxed{\text{-1,4,-1}} & -I & \\ & -I & \boxed{\text{-1,4,-1}} & \ddots \\ & & \ddots & \ddots \end{bmatrix} \\
&= I_m \otimes T_{N_c} + T_m \otimes I_{N_c}
\end{aligned}
$$

Die Eigenwerte $\lambda_i^{(N_c)}$ und Eigenvektoren $u_i^{(N_c)}$ der Matrix

$$
T_{N_c} = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}
$$

sind bekanntlich

$$\begin{aligned}
\lambda_i^{(N_c)} &= 2 - 2\cos\frac{\pi i}{N_c+1} \\
&= 4\sin^2\frac{\pi i}{2\,(N_c+1)} \\
u_i^{(N_c)} &= \sqrt{\frac{2}{N_c+1}}\left[\sin\frac{j i \pi}{N_c+1}\right]_{j=1}^{N_c} \\
\Lambda_{N_c} &:= diag(\lambda_1^{(N_c)},\ldots,\lambda_{N_c}^{(N_c)}) \\
U_{N_c} &:= \left[u_1^{(N_c)},\ldots,u_{N_c}^{(N_c)}\right] \\
\Longrightarrow \qquad T_{N_c} &= U_{N_c}\Lambda_{N_c}U_{N_c}^T
\end{aligned}$$

$$\begin{aligned}
U &:= U_m \otimes U_{N_c} \\
\Lambda &:= I_m \otimes \Lambda_{N_c} + \Lambda_m \otimes I_{N_c}
\end{aligned}$$

Aus den Eigenschaften des Kronecker Produktes folgt damit:

$$K_{I,s} = U\Lambda U^T$$

noch ein paar Vorüberlegungen:

$$\begin{aligned}
\alpha_i &:= \sqrt{\frac{2}{m+1}}\sin\frac{\pi i}{m+1} \\
\Longrightarrow \qquad e_1^T U_m &= [\alpha_1,\ldots,\alpha_m]
\end{aligned}$$

$$\begin{aligned}
\left(e_1^T \otimes I_{N_c}\right) U\Lambda^{-1}U^T\left(e_1 \otimes I_{N_c}\right) &= \left(\left(e_1^T U_m\right)\otimes U_{N_c}\right)\Lambda^{-1}\left(\left(U_m^T e_1\right)\otimes U_{N_c}^T\right) \\
&= U_{N_c}\left[\alpha_1 I, \alpha_2 I, \ldots, \alpha_m I\right]\Lambda^{-1}\begin{bmatrix}\alpha_1 I \\ \alpha_2 I \\ \vdots \\ \alpha_m I\end{bmatrix} U_{N_c}^T
\end{aligned}$$

$$\begin{aligned}
\Lambda &= \begin{bmatrix}\lambda_1^{(m)} I + \Lambda_{N_c} & & & \\ & \lambda_2^{(m)} I + \Lambda_{N_c} & & \\ & & \ddots & \\ & & & \lambda_m^{(m)} I + \Lambda_{N_c}\end{bmatrix} \\
\Longrightarrow \qquad \Lambda^{-1} &= \begin{bmatrix}(\lambda_1^{(m)} I + \Lambda_{N_c})^{-1} & & & \\ & (\lambda_2^{(m)} I + \Lambda_{N_c})^{-1} & & \\ & & \ddots & \\ & & & (\lambda_m^{(m)} I + \Lambda_{N_c})^{-1}\end{bmatrix}
\end{aligned}$$

Somit können wir für das Schur-Komplement feststellen:

$$\begin{aligned}
S_c &= K_c - 2K_{cI,1}K_{I,1}^{-1}K_{Ic,1} \\
&= U_{N_c}\left(2I_{N_c} + \Lambda_{N_c}\right)U_{N_c}^T - 2\left(e_1^T \otimes I_{N_c}\right)U\Lambda^{-1}U^T\left(e_1 \otimes I_{N_c}\right) \\
&= U_{N_c}\left(2I_{N_c} + \Lambda_{N_c} - 2\sum_{i=1}^{m}\alpha_i^2(\lambda_i^{(m)}I_{N_c} + \Lambda_{N_c})^{-1}\right)U_{N_c}^T \\
\Longrightarrow \quad S_c &= f(T_{N_c}) \\
f(\tau) &= 2 + \tau - 2\sum_{i=1}^{m}\frac{\alpha_1^2}{\lambda_i^{(m)} + \tau}
\end{aligned}$$

Dieses $f(\tau)$ ist somit monoton wachsend in τ. Da die Eigenwerte $\lambda_i^{(N_c)}$ von T_{N_c} alle im Interval $(0, 4)$ liegen kann man also die Eigenwerte von S_c abschätzen durch:

$$\begin{aligned}
f(\lambda_1^{(N_c)}) &= \lambda_{min}(S_c) \geq f(0) \\
f(\lambda_{N_c}^{(N_c)}) &= \lambda_{max}(S_c) \leq 2 + 4 = 6
\end{aligned}$$

Betrachten wir nun $f(0)$:

$$\begin{aligned}
f(0) &= 2 - 2\sum_{i=1}^{m}\frac{\sin^2\frac{i\pi}{m+1}}{4\sin^2\frac{i\pi}{2(m+1)}} \cdot \frac{2}{m+1} \\
&= 2 - 2\sum_{i=1}^{m}\frac{4\sin^2\frac{i\pi}{2(m+1)} \cdot \cos^2\frac{i\pi}{2(m+1)}}{4\sin^2\frac{i\pi}{2(m+1)}} \cdot \frac{2}{m+1} \\
&= 2 - \frac{4}{m+1}\sum_{i=1}^{m}\cos^2\frac{i\pi}{2(m+1)} \\
&= 2 - \frac{2}{m+1}\left(\sum_{i=1}^{m}\left(1 + \cos\frac{i\pi}{m+1}\right)\right) \\
&= 2 - \frac{2}{m+1}\left(m + \sum_{i=1}^{m}\cos\frac{i\pi}{m+1}\right) \\
&= 2 - \frac{2}{m+1}m \\
&= 2\left(1 - \frac{m}{m+1}\right) \\
&= \frac{2}{m+1} \\
&= \mathcal{O}(h)
\end{aligned}$$

Somit ist $\kappa(S_c) = \mathcal{O}(h^{-1})$ was bereits wesentlich besser ist als $\kappa(K) = \mathcal{O}(h^{-2})$. Damit stellt sich jedoch auch schon die Frage ob sich nicht sogar ein Vorkonditionierer C_c finden läßt, so das $\kappa(C_c^{-1}S_c) = \mathcal{O}(1)$. Dies wäre z.B. der Fall wenn

$C_c = g(T_{N_c})$ und

$$\underline{\gamma} \leq \frac{f(\tau)}{g(\tau)} \leq \overline{\gamma} \qquad \forall \tau \in (0,4), \quad \overline{\gamma}/\underline{\gamma} = \mathcal{O}(1)$$

Dryja fand Ende der '80er Jahre eine Lösung hierfür:

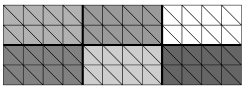
$$g(\tau) = \sqrt{\tau}$$

d.h.:

$$C_c^{-1} = g(T_{N_c}) = U_{N_c} \Lambda_{N_c}^{-\frac{1}{2}} U_{N_c}^T$$

Dies läßt sich auch noch sehr effizient implementieren, zumindestens wenn $N_c = 2^L - 1$ ist, denn dann kann man U_{N_c} durch die Fast Fourier Transformation realisieren.
$\Longrightarrow$ der „Dryja Precoditioner" für eindimensionale Koppelränder

Man kann dies sogar noch verallgemeinern. Bei vielen Teilgebieten enthält Γ_c Kreuzungspunkte.



Hier läßt sich ein Vorkonditionierer der Form $C_c = blockdiag(C_{c,1}, C_{c,2}, \ldots, C_{c,k})$ in Verbindung mit einem Grobgitterlöser anwenden. (nach Bramble/Pasciak/Schatz: BPS-Peceonditioner)

Fazit zu Technik II:

- geht gut parallel
- wenn man den obigen Vorkonditionierer anwendet ist die Anzahl der Iterationen beim PCG $\mathcal{O}(1)$
- aber die Multiplikationen

$$S_{c,s}q_{c,s} = K_{c,s}q_{c,s} - K_{cI,s}v_{I,s}^{-1} \qquad (2)$$
$$K_{I,s}v_{I,s} = K_{Ic,s}q_{c,s} \qquad (3)$$

sind teuer da das LGS (3) sehr genau gelöst werden muß

2.2.4 Technik III: PCG für $K\underline{u} = \underline{b}$

(ohne Assemblierung der Gesammtsteifigkeitsmatrix)
Hier wird ein Vorkonditionierer verwendet, der auf einer Block-Cholesky-Idee beruht:

Das Schur-Komplement Gleichungssystem ist äquivalent zu einer Block-Cholesky-Zerlegung

$$K = \begin{bmatrix} K_c & K_{cI} \\ K_{Ic} & K_I \end{bmatrix} = \begin{bmatrix} I_c & K_{cI}K_I^{-1} \\ 0 & I_I \end{bmatrix} \begin{bmatrix} S_c & 0 \\ 0 & K_I \end{bmatrix} \begin{bmatrix} I_c & 0 \\ K_I^{-1}K_{Ic} & I_I \end{bmatrix}$$

$\Longrightarrow$ verwenden PCG für $K\underline{u} = \underline{b}$ und als Vorkonditionierer:

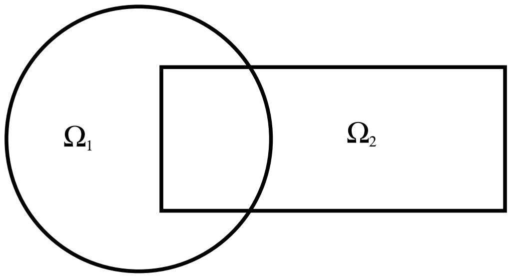
$$C^{-1} := \begin{bmatrix} I & 0 \\ -C_I^{-1}K_{Ic} & I \end{bmatrix} \begin{bmatrix} C_c^{-1} & 0 \\ 0 & C_I^{-1} \end{bmatrix} \begin{bmatrix} I & -K_{cI}C_I^{-1} \\ 0 & I \end{bmatrix}$$

und wählen $C_I = blockdiag(C_{I,1}, C_{I,2}, \ldots, C_{I,p})$. Wenn nun C_c ein Vorkonditionierer für S_c ist (analog zu 2.2.3), und $C_{I,s}$ Vorkonditionierer für $K_{I,s}$ ist, so ist C Vorkonditionierer für K. Dieser Vorkonditionierer läßt sich mit Hilfe der ASM leicht und effizient untersuchen.

3 Die Additive Schwarzsche Methode (ASM)

3.1 Historie: Multiplikative Schwarzsche Methode (MSM)

Die ASM ist eine Methode zur Konstruktion von „aparallelen“ Vorkonditionierern. Bevor wir uns jedoch mit der Methode selbst beschäftigen, wollen wir einen Blick in die Geschichte werfen um zu verstehen wo die Idee herkommt.
Die MSM wurde 1869 von dem Mathematiker H. Schwarz entwickelt. Es ist eine Methode um Laplace Probleme auf (überlappend) zusammengesetzten Gebieten zu lösen.
Beispiel:



ges: *Funktion* $u : \mathbb{R}^2 \mapsto \mathbb{R}$ *mit:*

$$\begin{aligned} -\Delta u &= f \quad |_{\Omega} \qquad \Omega = \Omega_1 \cup \Omega_2 \\ u &= 0 \quad |_{\partial\Omega} \end{aligned}$$

Die Lösung mittels MSM geschieht nun durch eine Iteration:
Start: $k = 1$, $u^{1,(0)}$ *als Lösung von* $-\Delta u^{1,(0)} = f \quad |_{\Omega_1}\,,\ u^{1,(0)} = 0 \quad |_{\partial\Omega_1}$

1. *löse:*
$$-\Delta u^{2,(k)} = f \quad |_{\Omega_2}\,,\ u^{2,(k)} = \begin{cases} u^{1,(k-1)} & |_{(\partial\Omega_2)\cap\Omega_1} \\ 0 & sonst \end{cases}$$

2. *löse:*
$$-\Delta u^{1,(k)} = f \quad |_{\Omega_1}\,,\ u^{1,(k)} = \begin{cases} u^{2,(k)} & |_{(\partial\Omega_1)\cap\Omega_2} \\ 0 & sonst \end{cases}$$

3. $k = k + 1$, *gehe zu 1.*

Analyse dieser Iteration

definieren
$$u^{(k)} := \begin{cases} u^{1,(k)} & |_{\Omega_1} \\ u^{2,(k)} & sonst \end{cases}$$

dann gilt für die Fehlerfunktion $z^{(k)} := u - u^{(k)}$*:*

$$z^{(k+1)} = (\boldsymbol{I} - \boldsymbol{P}_1)(\boldsymbol{I} - \boldsymbol{P}_2)z^{(k)}$$

wobei $\boldsymbol{P}_i$ *die Orthoprojektion bezüglich des Energiefunktionals* $a(.,.)$ *ist und* $\boldsymbol{I}$ *die Identität.*

Wenn man sich den Term

$$(\boldsymbol{I} - \boldsymbol{P}_1)(\boldsymbol{I} - \boldsymbol{P}_2) = \boldsymbol{I} - \boldsymbol{P}_1 - \boldsymbol{P}_2 + \boldsymbol{P}_1\boldsymbol{P}_2$$

näher betrachtet fällt auf das es der Term $\boldsymbol{P}_1\boldsymbol{P}_2$ ist, der nicht parallel für die unterschiedlichen Teilgebiete berechnet werden kann. Bei diesem Nachteil setzt die ASM ein. Bei der ASM ist die entsprechende Entwicklung der Fehlerfunktion:

$$\begin{aligned} z^{(k+1)} &= (\boldsymbol{I} - \omega(\boldsymbol{P}_1 + \boldsymbol{P}_2))\, z^{(k)} \\ &= z^{(k)} - \omega w^1 - \omega w^2 \\ w^i &:= \boldsymbol{P}_i z^{(k)} \end{aligned}$$

Somit sind die w^i parallel berechenbar.

3.2 Der Algorithmus ASM und seine Matrixrepräsentation

Wir betrachten hier den FE-Raum $\mathbb{V}(\subset H_0^1(\Omega))$ zu einem Variationsproblem mit der Bilinearform $a(u,v): \quad \mathbb{V} \times \mathbb{V} \longmapsto \mathbb{R}$, und suchen die FE-Lösung $u^* \in \mathbb{V}\,(u^* = \Phi\underline{u}^*$, mit Φ ist Basis in $\mathbb{V}$).
Wenn wir ein Iterationsverfahren zur Lösung verwenden erhalten wir in der k-ten Iteration die Näherung u für die Lösung u^*. Somit können wir den Fehler z definieren durch $z := u - u^*$.
In dieser Notation stellt sich ein Vorkonditionierer C^{-1} dar als Abbildung:

$$\mathbf{r} \longmapsto w$$

wobei w eine Näherung für z sein soll. Das oben bereits verwendete Residuum $\mathbf{r}$ ist gegeben durch

$$\langle \mathbf{r}, v\rangle = a(u,v) - \langle f, v\rangle \qquad \forall v \in \mathbb{V}$$

oder in Matrixschreibweise:

$$\begin{aligned} \underline{r} &= K\underline{u} - \underline{b} \\ \underline{w} &= C^{-1}\underline{r}, \\ \underline{r} &= [\langle \mathbf{r}, \varphi_i\rangle]_{i=1}^N \end{aligned}$$

Wir gehen hier davon aus, daß die Bilinearform $a(.,.)$ coerciv ist, somit repräsentiert sie ein Skalarprodukt, welches eine Norm induziert, die sogenannte Energienorm

$$|||u||| = \sqrt{a(u,u)}.$$

Sei nun $\mathbb{V} = \mathbb{V}_0 + \mathbb{V}_1 + \ldots + \mathbb{V}_p$, $\mathbb{V}_i$ Teilräume von $\mathbb{V}$, Φ eine Basis von $\mathbb{V}$. Dann gibt es Matrizen $V_i \in \mathbb{R}^{N\times N_i}$ die vollen Rang haben, so daß $\mathbb{V}_i = span(\Phi V_i) \quad \forall i = 0, 1, \ldots, p$. D.h. insbesondere, daß $dim(\mathbb{V}_i) = N_i$ ist.

Definition 1. *Eine Abbildung* $\boldsymbol{P}_i : \quad \mathbb{V} \longmapsto \mathbb{V}_i$ *heißt* *Orthoprojektor* *bezüglich* $a(.,.)$*, wenn für alle* $u \in \mathbb{V}$ *gilt:*

$$a(\boldsymbol{P}_i u, v) = a(u, v) \qquad \forall v \in \mathbb{V}_i \tag{4}$$

Was heißt das nun in Matrixschreibweise?
Seien $u_i := \mathbf{P}_i u =: \Phi V_i \underline{u}_i, \quad u = \Phi \underline{u} \in \mathbb{V}, \quad v \in \mathbb{V}_i, \quad v = \Phi V_i \underline{v}$ und $\underline{v} \in \mathbb{R}^{N_i}$.
Die Forderung (4) ist äquivalent zu:

$$\begin{array}{rrcll} & a(\Phi V_i \underline{u}_i, \Phi V_i \underline{v}) & = & a(\Phi \underline{u}, \Phi V_i \underline{v}) & \forall \underline{v} \in \mathbb{R}^{N_i} \\ = & \underline{v}^T V_i^T K V_i \underline{u}_i & = & \underline{v}^T V_i^T K \underline{u} & \forall \underline{v} \in \mathbb{R}^{N_i} \\ \\ \text{also:} & \left(V_i^T K V_i\right) \underline{u}_i & = & V_i^T K \underline{u} & \end{array}$$

Damit erhalten wir eine alternative Repräsentation:

$$u_i = \mathbf{P}_i u = \Phi V_i \left(V_i^T K V_i\right)^{-1} V_i^T K \underline{u} \tag{5}$$

Definition 2. *ASM* *bedeutet einen Vorkonditionierer zu benutzen, mit:*

$$\begin{array}{rcl} w & = & \boldsymbol{P} z = \boldsymbol{P}(u - u^*) \\ \boldsymbol{P} & = & \sum\limits_{i=0}^{p} \boldsymbol{P}_i \end{array}$$

d.h.:

a. in $\mathbb{V}$: $w = \sum w_i$, die w_i werden definiert durch

$$a(w_i, v) = a(z, v) \qquad \forall v \in \mathbb{V}_i$$

übrigens ist $a(z, v) = a(u, v) - \langle f, v \rangle = \langle \mathbf{r}, v \rangle$

b. im $\mathbb{R}^N$: $w_i = \Phi V_i \underline{w}_i$,
mit der alternativen Repräsentation (5) erhält man also:

$$\begin{array}{rcl} \underline{w}_i & = & \left(V_i^T K V_i\right)^{-1} V_i^T K \underline{z} \\ & = & \left(V_i^T K V_i\right)^{-1} V_i^T K \left(\underline{u} - \underline{u}^*\right) \\ & = & \left(V_i^T K V_i\right)^{-1} V_i^T \underline{r} \end{array}$$

Bemerkung 8. *Das heißt, im Algorithmus wird man die Vorkonditionierer-Anwendung* $\underline{w} = C^{-1} \underline{r}$ *ausdrücken durch:*

1. *löse:* $\left(V_i^T K V_i\right) \underline{w}_i = V_i^T \underline{r}$

2. $\underline{w} = \sum\limits_{i=0}^{p} V_i \underline{w}_i$

Bemerkung 9. *Man kann den Vorkonditionierer auch auffassen als:*

$$C^{-1} = \sum_{i=0}^{p} \underbrace{V_i}_{\textit{Prolongation}} \underbrace{\left(V_i^T K V_i\right)^{-1}}_{\textit{Teilraumlösung}} \underbrace{V_i^T}_{\textit{Restriktion}}$$

Folgerung 1. *Für beliebige $u = \Phi\underline{u} \in \mathbb{V}$ und $v = \Phi\underline{v} \in \mathbb{V}$ gilt:*

$$a(\boldsymbol{P}u, v) = \underline{v}^T K C^{-1} K \underline{u}$$

3.3 Spektraläquivalenzabschätzung

Ziel ist eine Abschätzung der Form:

$$\underline{\gamma} C \leq K \leq \overline{\gamma} C \tag{6}$$

Die Konstanten $\underline{\gamma}$ und $\overline{\gamma}$ sind für die Konvergenzgeschwindigkeit des PCG (Alg. (1)) wichtig, da für den Fehler im k-ten Schritt die folgende Abschätzung gilt:

$$||z^{(k)}||_K \sim \eta^k; \qquad \eta = \frac{1-\sqrt{\xi}}{1+\sqrt{\xi}}; \qquad \xi = \underline{\gamma}/\overline{\gamma}$$

3.3.1 Der allgemeine Fall: $\mathbb{V} = \sum_{i=0}^{p} \mathbb{V}_i$, $p \geq 1$

Lemma 1 (Lemma von Lions (1989)).
Voraussetzungen:

- $\forall v \in \mathbb{V} \quad \exists v_i \in \mathbb{V}_i$, *mit* $v = \sum v_i$
- $\sum_{i=0} a(v_i, v_i) \leq c_L^2 a(v, v)$, *für eine Konstante* $c_L > 0$

Dann gilt:

$$a(v, v) \leq c_L^2 a(\boldsymbol{P}v, v) \qquad \forall v \in \mathbb{V}$$

Wobei wie schon früher $\boldsymbol{P} = \sum \boldsymbol{P}_i$ Summe orthogonaler Projektionen
$\boldsymbol{P}_i : \quad \mathbb{V} \longmapsto \mathbb{V}_i$ *ist.*

Beweis.

$$\begin{aligned}
|||v|||^2 &= a(v,v) && (7)\\
&= a(v,\sum v_i) && (8)\\
&= \sum a(v,v_i) && (9)\\
&= \sum a(v,\mathbf{P}_i v_i) && (10)\\
&= \sum a(\mathbf{P}_i v,v_i) && (11)\\
&\le \sum |||\mathbf{P}_i v|||\cdot|||v_i||| && (12)\\
&\le \left(\sum |||\mathbf{P}_i v|||^2\right)^{1/2}\cdot\left(\sum |||v_i|||^2\right)^{1/2} && (13)\\
&\le \left(\sum |||\mathbf{P}_i v|||^2\right)^{1/2} c_L|||v||| && (14)\\
\Longrightarrow |||v||| &\le c_L\left(\sum a(\mathbf{P}_i v,\mathbf{P}_i v)\right)^{1/2} = c_L a(\mathbf{P}v,v)^{1/2} && (15)
\end{aligned}$$

(12) → (13) mit Cauchy Schwarzscher Ungleichung

(13) → (14) laut Voraussetzung.

□

Bemerkung 10. *Das Ergebnis ist äquivalent zu:*

$$\lambda_{min}(\boldsymbol{P}) = \min_{v\in\mathbb{V}} \frac{a(\boldsymbol{P}v,v)}{a(v,v)} \ge \frac{1}{c_L^2}$$

Bemerkung 11. *Offenbar gilt $\lambda_{max}(\boldsymbol{P}) \le p+1$, da $\boldsymbol{P}$ Summe von $p+1$ Orthoprojektoren ist. Somit gilt die Abschätzung (6) mit den Konstanten:*

$$\underline{\gamma} = \frac{1}{c_L^2} \qquad \overline{\gamma} = p+1$$

Beweis.

$$\begin{array}{rcccll}
& \frac{1}{c_L^2} & \le & \frac{a(\mathbf{P}v,v)}{a(v,v)} & \le & p+1 \\
\Longleftrightarrow & \frac{1}{c_L^2} & \le & \frac{\underline{v}^T K C^{-1} K \underline{v}}{\underline{v}^T K \underline{v}} & \le & p+1 \quad \forall \underline{v}\in\mathbb{R}^N \\
\Longleftrightarrow & \frac{1}{c_L^2} & \le & \frac{\underline{u}^T C^{-1} \underline{u}}{\underline{u}^T K^{-1} \underline{u}} & \le & p+1 \quad |\, \underline{u} := K\underline{v} = K^T\underline{v} \\
\Longleftrightarrow & \frac{1}{c_L^2} & \le & \lambda_i(KC^{-1}) & \le & p+1 \quad \forall i = 1,\dots,N \\
\Longleftrightarrow & \frac{1}{c_L^2} & \le & \frac{\underline{v}^T K \underline{v}}{\underline{v}^T C \underline{v}} & \le & p+1
\end{array}$$

□

Dabei sind die $\lambda_i(KC^{-1})$ die Eigenwerte von KC^{-1}, also die nichttrivialen Lösungen von $C^{-1}x = \lambda K^{-1}x$, gleich den Eigenwerten von $C^{-1}K$.

3.3.2 Der Fall p=1 mit $\mathbb{V} = \mathbb{V}_0 \dot{+} \mathbb{V}_1$ (direkte Summe)

Hier ist der Winkel zwischen den Räumen $\mathbb{V}_0$ und $\mathbb{V}_1$ von zentraler Bedeutung. Deshalb wollen wir ihn zunächst definieren.

Definition 3. *Der Winkel $\angle(\mathbb{V}_0, \mathbb{V}_1)$ ist durch die folgende Gleichung definiert:*

$$\gamma := \sup_{v_0 \in \mathbb{V}_0, v_1 \in \mathbb{V}_1} \frac{a(v_0, v_1)}{|||v_0||| \cdot |||v_1|||} =: \cos\angle(\mathbb{V}_0, \mathbb{V}_1)$$

Bemerkung 12. *Da $\mathbb{V} = \mathbb{V}_0 \dot{+} \mathbb{V}_1$ direkte Summe der Räume $\mathbb{V}_0$ und $\mathbb{V}_1$ ist gilt $\gamma < 1$.*

Bemerkung 13. *Man gewinnt durch diese Definition die „<u>verschärfte Cauchy-Ungleichung</u>“*

$$a(v_0, v_1) \leq \gamma |||v_0||| \cdot |||v_1||| \qquad \forall v_0 \in \mathbb{V}_0, v_1 \in \mathbb{V}_1.$$

Lemma 2. *Für den Fall p=1 mit $\mathbb{V} = \mathbb{V}_0 \dot{+} \mathbb{V}_1$ gilt die Abschätzung (6) mit den Konstanten:*

$$\underline{\gamma} = 1 - \gamma \qquad \overline{\gamma} = 2$$

Beweis. Wir zeigen das die Voraussetzungen des Lemma von Lions (Lemma 1) mit $c_L^{-2} = 1 - \gamma$ erfüllt sind:

$$\begin{aligned}
a(v, v) &= a(v_0 + v_1, v_0 + v_1) \\
&= |||v_0|||^2 + |||v_1|||^2 + 2a(v_0, v_1) \\
&\geq |||v_0|||^2 + |||v_1|||^2 - 2\gamma |||v_0||| \cdot |||v_1||| \\
&\geq (1 - \gamma)\left(|||v_0|||^2 + |||v_1|||^2\right) \\
a(v, v) &\geq (1 - \gamma) \sum_{i=0}^{1} a(v_i, v_i)
\end{aligned}$$

Damit folgt die Behauptung aus dem Lemma von Lions.

□

Bemerkung 14. *$\overline{\gamma} = 2$ ist noch nicht optimal. Wir werden noch zeigen $\overline{\gamma} = 1 + \gamma$.*

Wie kann man nun in diesem Fall den Vorkonditionierer C bzw. C^{-1} darstellen?

$$\begin{aligned}
\mathbb{V} &= span(\Phi) \\
&= span(\Phi\,[V_0, V_1]) \\
\mathbb{V}_i &= span(\Phi V_i) \\
C^{-1} &= V_0 \left(V_0^T K V_0\right)^{-1} V_0^T + V_1 \left(V_1^T K V_1\right)^{-1} V_1^T
\end{aligned}$$

Die zusammengesetzte Matrix $V = [V_0, V_1] \in \mathbb{R}^{N\times N}$ ist bei dieser Notation eine reguläre Matrix. Man kann also auch die folgende Darstellung verwenden:

$$C^{-1} = V \begin{bmatrix} \left(V_0^T K V_0\right)^{-1} & \\ & \left(V_1^T K V_1\right)^{-1} \end{bmatrix} V^T$$

invertieren ergibt:

$$C = V^{-T} \begin{bmatrix} V_0^T K V_0 & \\ & V_1^T K V_1 \end{bmatrix} V^{-1}$$

Damit wollen wir nun den Beweis von Bemerkung 14 angehen.

Beweis.

$$\begin{aligned}
v &= \Phi \underline{v} \\
&= v_0 + v_1 \\
\underline{v}^T K \underline{v} &= a(v, v) \\
&= a(v_0 + v_1, v_0 + v_1) \\
&= |||v_0|||^2 + |||v_1|||^2 + 2a(v_0, v_1) \\
&\leq |||v_0|||^2 + |||v_1|||^2 + 2\gamma |||v_0||| \cdot |||v_1||| \\
&\leq (1+\gamma)\left(|||v_0|||^2 + |||v_1|||^2\right)
\end{aligned}$$

D.h., die Behauptung ist bewiesen wenn $|||v_0|||^2 + |||v_1|||^2 = \underline{v}^T C \underline{v}$.
Hierzu nun:

$$\begin{aligned}
v &= \Phi \underline{v} \\
&= \Phi V (V^{-1} \underline{v}) \\
\underline{w} &= V^{-1} \underline{v} \\
&= \begin{bmatrix} \underline{w}_0 \\ \underline{w}_1 \end{bmatrix} \\
V \underline{w} &= V_0 \underline{w}_0 + V_1 \underline{w}_1
\end{aligned}$$

$$\begin{aligned}
|||v_0|||^2 + |||v_1|||^2 &= a(\Phi V_0 \underline{w}_0, \Phi V_0 \underline{w}_0) + a(\Phi V_1 \underline{w}_1, \Phi V_1 \underline{w}_1) \\
&= \underline{w}_0^T V_0^T K V_0 \underline{w}_0 + \underline{w}_1^T V_1^T K V_1 \underline{w}_1 \\
&= \underline{w}^T \begin{bmatrix} V_0^T K V_0 & \\ & V_1^T K V_1 \end{bmatrix} \underline{w} \\
&= \underline{v}^T \left(V^{-T} \begin{bmatrix} V_0^T K V_0 & \\ & V_1^T K V_1 \end{bmatrix} V^{-1} \right) \underline{v} \\
&= \underline{v}^T C \underline{v}
\end{aligned}$$

□

Fazit:
Der Fall $\mathbb{V} = \mathbb{V}_0 \dot{+} \mathbb{V}_1$ läßt sich auch im Kontext einer Basistransformation interpretieren:

- K ist Steifigkeitsmatrix zur Basis Φ
- definieren Basis $\Psi := \Phi V$, dann ist $V^T K V$ Steifigkeitsmatrix zur Basis Ψ
- $V = [V_0, V_1]$ definiert die Unterräume $\mathbb{V}_0$ und $\mathbb{V}_1$
- wir haben die Konstante $\gamma < 1$

$$\gamma = \max_{u \in \mathbb{V}_0, v \in \mathbb{V}_1} \frac{a(u,v)}{(a(u,u) \cdot a(v,v))^{1/2}} = \max_{\underline{w}_0, \underline{w}_1} \frac{\underline{w}_1^T V_1^T K V_0 \underline{w}_0}{\left(\underline{w}_0^T V_0^T K V_0 \underline{w}_0 \cdot \underline{w}_1^T V_1^T K V_1 \underline{w}_1\right)^{1/2}}$$

- damit gilt:

$$(1-\gamma) \begin{bmatrix} V_0^T K V_0 & \\ & V_1^T K V_1 \end{bmatrix} \leq V^T K V \leq (1+\gamma) \begin{bmatrix} V_0^T K V_0 & \\ & V_1^T K V_1 \end{bmatrix}$$

- und somit auch:

$$(1-\gamma) V^{-T} \begin{bmatrix} V_0^T K V_0 & \\ & V_1^T K V_1 \end{bmatrix} V^{-1} \leq K \leq (1+\gamma) V^{-T} \begin{bmatrix} V_0^T K V_0 & \\ & V_1^T K V_1 \end{bmatrix} V^{-1}$$

- für den Vorkonditionierer $C := V^{-T} \begin{bmatrix} V_0^T K V_0 & \\ & V_1^T K V_1 \end{bmatrix} V^{-1}$ gilt die Spektralabschätzung

$$(1-\gamma) C \leq K \leq (1+\gamma) C.$$

3.3.3 Inexakte Lösung für Subprobleme

Bei der ASM werden Subprobleme der Form

$$\left(V_i^T K V_i\right) \underline{w}_i = V_i^T \underline{r}$$

gelöst um dann das vorkonditionierte Residuum zu akkumulieren:

$$\underline{w} = \sum_{i=0}^{p} V_i \underline{w}_i$$

Um das Lösen der Subprobleme effektiv zu gestalten benötigen wir noch einen Subproblem-Vorkonditionierer C_i mit:

$$\underline{\gamma}_i C_i \leq V_i^T K V_i \leq \overline{\gamma}_i C_i \qquad \forall i = 0, \ldots, p$$

Wir definieren damit einen neuen Vorkonditionierer $\widetilde{C}$:

$$\widetilde{C}^{-1} = \sum_{i=0}^{p} V_i C_i^{-1} V_i^T$$

dann gilt:

$$\gamma_1 \widetilde{C} \leq K \leq \gamma_2 \widetilde{C}$$

mit den Konstanten: $p > 1:\quad \gamma_1 = c_L^{-2} \cdot \min_i \underline{\gamma}_i \qquad \gamma_2 = (p+1) \max_i \overline{\gamma}_i$

$p = 1:\quad \gamma_1 = (1-\gamma) \min_i \underline{\gamma}_i \qquad \gamma_2 = (1+\gamma) \max_i \overline{\gamma}_i$

Beweis. (für $p = 1$)
rechte Seite der Abschätzung:

$$\begin{aligned}
\underline{v}^T \widetilde{C}^{-1} \underline{v} &= \sum_i (V_i^T \underline{v})^T C_i^{-1} (V_i^T \underline{v}) \\
&\leq \sum_i (V_i^T \overline{v})^T \underline{\gamma}_i \left(V_i^T K V_i\right)^{-1} (V_i^T \underline{v}) \\
&\leq \left(\max_i \overline{\gamma}_i\right) \underline{v}^T C^{-1} \underline{v} \\
&\leq (1+\gamma) \left(\max_i \overline{\gamma}_i\right) \underline{v}^T K^{-1} \underline{v}
\end{aligned}$$

linke Seite der Abschätzung:

$$\begin{aligned}
\underline{v}^T \widetilde{C}^{-1} \underline{v} &= \sum_i (V_i^T \underline{v})^T C_i^{-1} (V_i^T \underline{v}) \\
&\geq \sum_i (V_i^T \underline{v})^T \underline{\gamma}_i \left(V_i^T K V_i\right)^{-1} (V_i^T \underline{v}) \\
&\geq \left(\min_i \underline{\gamma}_i\right) \underline{v}^T C^{-1} \underline{v} \\
&\geq (1-\gamma) \left(\min_i \underline{\gamma}_i\right) \underline{v}^T K^{-1} \underline{v}
\end{aligned}$$

□

Wir unterscheiden im weiteren die zwei Verfahren:

$\underline{w}$	$:=$	$C^{-1}\underline{r}$	„exakte“ ASM (i.A. schwer zu realisieren)
$\underline{w}$	$:=$	$\widetilde{C}^{-1}\underline{r}$	„vorkonditionierte“ ASM

3.4 Anwendungen: DD-Vorkonditionierer

3.4.1 Die näherungsweise diskret harmonische Basis und der DD-Vorkonditionierer

Wir betrachten hier eine Basis Φ

$$\Phi = \left[\varphi_1, \ldots, \varphi_{N_c}, \varphi_{N_c+1}, \ldots, \varphi_{N_c+N_{I,1}}, \ldots \varphi_N\right]$$

mit der besonderen Knotennummerierung entsprechend der DD.

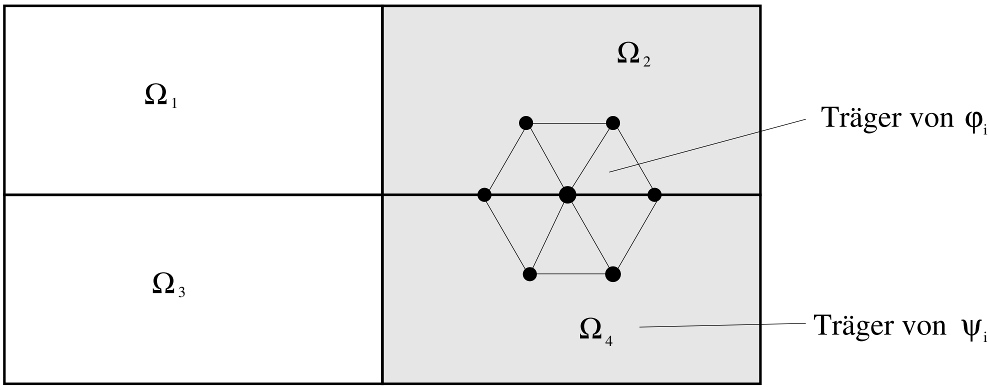
Definition 4. *Die (exakt) diskret harmonische Basis* Ψ *zu einer Basis* Φ *(wie oben) besteht aus*

$$\Psi = \left[\psi_1, \ldots, \psi_{N_c}, \varphi_{N_c+1}, \ldots, \varphi_{N_c+N_{I,1}}, \ldots, \varphi_N\right],$$

wobei die $\psi_i \in \mathbb{V}$ *definiert sind durch*

$$\begin{aligned} a(\psi_i, v) &= 0 \qquad \forall v \in \mathbb{V} \\ \psi_i\,|_{\Gamma_c} &= \varphi_i. \end{aligned}$$

(Sie unterscheidet sich also von der Basis Φ *nur in den Basisfunktionen die zu den Koppelknoten gehören.)*



Bemerkung 15. *Die exakt harmonische Basis würde durch*

$$\begin{aligned} \Delta\psi_i &= 0 \\ \psi_i\,|_{\Gamma_c} &= \varphi_i \end{aligned}$$

definiert sein, allerdings ist dann nicht $\psi_i \in \mathbb{V}$.

Wir definieren im Folgenden Ψ als näherungsweise diskret harmonische Basis. Dann gibt es eine reguläre Matrix $V \in \mathbb{R}^{N\times N}$, so daß

$$\Psi = \Phi V$$

gilt. Diese Matrix V kann man dann wie folgt zerlegen:

$$\begin{aligned} V &= [\ \underbrace{V_c}_{N_c},\ \underbrace{V_I}_{N_I}\ ] \text{ -Spalten} \\ V_I &= \begin{bmatrix} 0 \\ I_{N_I} \end{bmatrix} \\ V_C &= \begin{bmatrix} I_{N_c} \\ -B_I^{-1} K_{Ic} \end{bmatrix} \end{aligned}$$

(Zur Definition von B_I siehe Bemerkung 16)
Durch diese Zerlegung kann man nun wieder die Funktionenräume definieren:

$$\begin{aligned} \mathbb{V}_c &:= span(\Phi V_c) \\ \mathbb{V}_I &:= span(\Phi V_I) \\ &= \mathbb{V}_{I,1} + \mathbb{V}_{I,2} + \cdots + \mathbb{V}_{I,p} \end{aligned}$$

Wie im Kapitel 3.3.2 kann man nun auch hier wieder die Konstante γ bestimmen. (Cosinus des Winkels zwischen den Räumen)

$$\gamma^2 = \max_{\underline{v}_c \in \mathbb{R}^{N_c}, \underline{v}_I \in \mathbb{R}^{N_I}} \frac{\left(\underline{v}_c^T V_c^T K V_I \underline{v}_I\right)^2}{\left(\underline{v}_c^T V_c^T K V_c \underline{v}_c\right) \left(\underline{v}_I^T V_I^T K V_I \underline{v}_I\right)}$$

Wie schon früher zerlegen wir nun auch wieder K in

$$K = \begin{bmatrix} K_c & K_{cI} \\ K_{Ic} & K_I \end{bmatrix}$$

wodurch sich die Folgenden Äquivalenzen ergeben:

$$\begin{aligned} V_I^T K V_I &= K_I \\ V_c^T K V_I &= V_c^T \begin{bmatrix} K_{cI} \\ K_I \end{bmatrix} \\ &= K_{cI} \left(I - B_I^{-T} K_I\right) \\ V_c^T K V_c &= V_c^T \begin{bmatrix} K_c - K_{cI} B_I^{-1} K_{Ic} \\ (I - K_I B_I^{-1}) K_{Ic} \end{bmatrix} \\ &= K_c - K_{cI} B_I^{-1} K_{Ic} - K_{cI} B_I^{-T} \left(I - K_I B_I^{-1}\right) K_{Ic} \end{aligned}$$

Bemerkung 16. *Wenn $B_I = K_I$ gilt, so ist $\gamma = 0$, also $\mathbb{V} = \mathbb{V}_c \oplus \mathbb{V}_I$.*[5] *In diesem Fall ist Ψ die exakt diskret harmonische Basis, deshalb heißt für jedes beliebige andere B_I unsere benutzte Basis „näherungsweise diskret harmonisch“.*

Bemerkung 17. *Man kann $V_c^T K V_c$ auch im Kontext des Schur Komplements S_c (siehe Kapitel 2.1) betrachten:*

$$\begin{aligned} V_c^T K V_c &= K_c - K_{cI} B_I^{-1} K_{Ic} - K_{cI} B_I^{-T} \left(I - K_I B_I^{-1}\right) K_{Ic} \\ &= \left(K_c - K_{cI} K_I^{-1} K_{Ic}\right) + K_{cI} K_I^{-1} K_{Ic} \\ &\quad - K_{cI} B_I^{-1} K_{Ic} - K_{cI} B_I^{-T} \left(I - K_I B_I^{-1}\right) K_{Ic} \\ V_c^T K V_c &= S_c + T_c \end{aligned}$$

[5] $\oplus$ bezeichnet die Orthogonale Summe

$$
\begin{aligned}
S_c &= K_c - K_{cI} K_I^{-1} K_{Ic} \\
T_c &:= K_{cI}\left(K_I^{-1} - B_I^{-1}\right) K_{Ic} - K_{cI} B_I^{-T}\left(I - K_I B_I^{-1}\right) K_{Ic} \\
&= K_{cI}\left(K_I^{-1} - B_I^{-1} - B_I^{-T}\left(I - K_I B_I^{-1}\right)\right) K_{Ic} \\
T_c &= K_{cI}\left(B_I^{-T} - K_I^{-1}\right) K_I \left(B_I^{-1} - K_I^{-1}\right) K_{Ic}
\end{aligned}
$$

Lemma 3.

$$\gamma = \cos\angle(\mathbb{V}_c, \mathbb{V}_I) = \sqrt{\frac{\mu}{1+\mu}}$$

mit

$$\mu = \rho(S_c^{-1} T_c) = \max_{\underline{x} \in \mathbb{R}^{N_c}} \frac{\underline{x}^T T_c \underline{x}}{\underline{x}^T S_c \underline{x}}$$

Beweis.

$$\gamma^2 = \max_{\underline{v}_c \in \mathbb{R}^{N_c}, \underline{v}_I \in \mathbb{R}^{N_I}} \frac{\left(\underline{v}_c^T \left(K_{cI}\left(I - B_I^{-T} K_I\right)\right) \underline{v}_I\right)^2}{\left(\underline{v}_c^T \left(S_c + T_c\right) \underline{v}_c\right)\left(\underline{v}_I^T K_I \underline{v}_I\right)} \tag{16}$$

zur Auswertung erst eine kurze Nebenrechnung:

$$
\begin{aligned}
B &:= K_{cI}\left(I - B_I^{-T} K_I\right) \\
A &:= S_c + T_c \\
C &:= K_I
\end{aligned}
$$

$$
\begin{aligned}
\max_{\underline{u},\underline{v}} \frac{\left(\underline{u}^T B \underline{v}\right)^2}{\underline{u}^T A \underline{u} \cdot \underline{v}^T C \underline{v}} &= \max \frac{\left(\underline{u}^T B C^{-1/2} \underline{w}\right)^2}{\underline{u}^T A \underline{u} \cdot \underline{w}^T \underline{w}} \qquad |\underline{w} := C^{1/2} \underline{v} \\
&= \max_{\underline{u}} \frac{\underline{u}^T B C^{-1} B^T \underline{u}}{\underline{u}^T A \underline{u}} \\
&= \rho(A^{-1} B C^{-1} B^T)
\end{aligned}
$$

Durch Anwenden dieser Überlegungen auf Gleichung (16) erhält man:

$$
\begin{aligned}
\gamma^2 &= \rho\left(\left(S_c + T_c\right)^{-1}\left(K_{cI}\left(I - B_I^{-T} K_I\right)\right) K_I^{-1}\left(K_{cI}\left(I - B_I^{-T} K_I\right)\right)^T\right) \\
&= \rho\left(\left(S_c + T_c\right)^{-1} T_c\right) \\
&= \rho\left(\left(I + S_c^{-1} T_c\right)^{-1} S_c^{-1} T_c\right)
\end{aligned}
$$

Da $f(\tau) = \frac{\tau}{1+\tau}$ eine monoton wachsende Funktion ist, folgt

$$\gamma^2 = \frac{\mu}{1+\mu}$$

mit dem in der Behauptung definierten μ.

□

<u>Fazit:</u>
ASM mit $\mathbb{V}_c, \mathbb{V}_I$ ist möglich. Jedoch muß μ möglichst klein sein, um die Methode effektiv zu machen. „Exakte ASM“ heißt also Gleichungssysteme mit $V_c^T K V_c = S_c + T_c$ und $V_I^T K V_I = K_I$ zu lösen. Das ist i.A. sehr aufwendig.
Aus diesem Grund verwendet man die „vorkonditionierte ASM“.

Definition 5. *Sei C_I Vorkonditionierer für K_I und C_c Vorkonditionierer für $S_c(+T_c)$ (z.B. Dryja), mit den Spektralabschätzungen:*

$$\begin{array}{lcl} \underline{\gamma}_I C_I & \leq K_I \leq & \overline{\gamma}_I C_I \\ \underline{\gamma}_c C_c & \leq S_c + T_c \leq & \overline{\gamma}_c C_c \end{array}$$

Dann heißt die Matrix

$$C = \begin{bmatrix} I & K_{cI} B_I^{-T} \\ 0 & I \end{bmatrix} \begin{bmatrix} C_c & 0 \\ 0 & C_I \end{bmatrix} \begin{bmatrix} I & 0 \\ B_I^{-1} K_{Ic} & I \end{bmatrix}$$

<u>DD-Vorkonditionierer</u> (aus der ASM).

3.4.2 Die Dirichlet - Dirichlet - DD und die Neumann - Dirichlet - DD

Hier betrachten wir den Fall $p = 2$. Also

$$\begin{aligned} K &= \begin{bmatrix} K_c & K_{cI,1} & K_{cI,2} \\ K_{Ic,1} & K_{I,1} & 0 \\ K_{Ic,2} & 0 & K_{I,2} \end{bmatrix} \\ V &= \begin{bmatrix} I & 0 & 0 \\ -B_{I,1}^{-1} K_{Ic,1} & I & 0 \\ -B_{I,2}^{-1} K_{Ic,2} & 0 & I \end{bmatrix} \\ &= \begin{bmatrix} V_c & V_I \end{bmatrix} \end{aligned}$$

$$\begin{aligned} V_c &= \begin{bmatrix} I \\ -B_{I,1}^{-1} K_{Ic,1} \\ -B_{I,2}^{-1} K_{Ic,2} \end{bmatrix} \\ V_I &= \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix} \end{aligned}$$

Das entspricht den bisherigen Definitionen von V_c und V_I. Um nun den DD-Vorkonditionierer anwenden zu können, fehlen uns noch ein paar Zutaten:

$$\begin{aligned} B_{I,s} &\approx K_{I,s} \\ C_{I,s} &\approx K_{I,s} \\ C_c &\approx S_c \end{aligned}$$

Durch die Wahl dieser Zutaten ergibt sich nun eine Vielzahl möglicher Verfahren. Hier zwei Beispiele:

1. klassisch:
   setzen $C_{I,s} = B_{I,s} = K_{I,s}$, d.h. man löst auf beiden Ω_s ein FE - Gleichungssystem mit Dirichlet Randbedingungen.
   Deshalb wird diese Methode „Dirichlet - Dirichlet - DD" genannt.

2. alternativ:

   $$\text{Man setzt}\quad V_0 = \underbrace{\begin{bmatrix} I & 0 \\ 0 & I \\ -B_{I,2}^{-1}K_{Ic,2} & 0 \end{bmatrix}}_{N_c + N_{I,1}} \quad \text{und} \quad V_1 = \underbrace{\begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}}_{N_{I,2}} \quad \text{- Spalten}$$

   und definiert die beiden zugehörigen Teilräume $\mathbb{V}_0$ und $\mathbb{V}_1$. Jetzt kann man vorläufig noch $B_{I,2} = K_{I,2}$ setzen. Dies ergibt die Neumann-Dirichlet-DD. Hierzu folgende drei Übungsaufgaben:

   (a) Die exakte ASM aufschreiben. (Es ist nicht möglich diese Methode wirklich zu implementieren.)

   (b) Eine einfachste vorkonditionierte ASM aufschreiben. Dabei soll mit $K_{I,2}$ (exakt) und mit $K_1 = \begin{bmatrix} K_{c,1} & K_{cI,1} \\ K_{Ic,1} & K_{I,1} \end{bmatrix}$ gelöst werden.
   Desweiteren sollen die Konstanten $\underline{\gamma}$ und $\overline{\gamma}$ der Spektraläquivalenzabschätzung für diesen Vorkonditionierer berechnet werden.

   $$\underline{\gamma} C \leq K \leq \overline{\gamma} C$$

   (Hinweis: In dem bereits betrachteten Modellfall ($\Omega_1 = \Omega_2$, siehe Abschnitt 2.2.3) ergibt sich $\underline{\gamma} = 1, \overline{\gamma} = 2$.)

   Analog zu obiger Überlegung heißt diese Methode Neumann-Dirichlet-DD.

   (c) Eine vorkonditionierte ASM aufschreiben, mit näherungsweisen Lösern für K_1 (Matrix zugehörig zum Raum $\mathbb{V}_0$) und $K_{I,2}$. Dabei sollen insbesondere $C_{I,2}$, $B_{I,2}$ und C_1 berücksichtigt werden.

3.4.3 Parallele Implementierung des DD-Vorkonditionierers aus 3.4.1

Der DD-Vorkonditionierer besitzt die Matrixdarstellung

$$C^{-1} = \sum_{i=0}^{1} V_i C_i^{-1} V_i^T \tag{17}$$

$$= \begin{bmatrix} I & 0 \\ -B_I^{-1}K_{Ic} & I \end{bmatrix} \begin{bmatrix} C_c^{-1} & 0 \\ 0 & C_I^{-1} \end{bmatrix} \begin{bmatrix} I & -K_{cI}B_I^{-T} \\ 0 & I \end{bmatrix} \tag{18}$$

Im allgemeinen Fall $p > 2$ haben die Matrizen K_{Ic} und K_I bekanntlich die folgende Struktur:

$$\begin{aligned} K_{Ic} &= \begin{bmatrix} K_{Ic,1}H_{c,1} \\ K_{Ic,2}H_{c,2} \\ \vdots \\ K_{Ic,p}H_{c,p} \end{bmatrix} \\ K_I &= \begin{bmatrix} K_{I,1} & & & \\ & K_{I,2} & & \\ & & \ddots & \\ & & & K_{I,p} \end{bmatrix} \end{aligned}$$

Die Blockdiagonalstruktur von K_I vererbt sich automatisch auch an B_I und C_I. Diese Matrizen bzw. Operationen sind also lokal auf jedem Prozessor definiert. Betrachten wir nun die Operation $\underline{w} := C^{-1}\underline{r}$ hinsichtlich ihrer parallelen Durchführbarkeit:
zunächst die beteiligten Vektoren und ihre DD-Darstellung

$$\begin{aligned} \underline{w} &= \begin{bmatrix} \underline{w}_c \\ \underline{w}_{I,1} \\ \vdots \\ \underline{w}_{I,p} \end{bmatrix} \\ \underline{w}_s &= H_s\underline{w} \\ &= \begin{bmatrix} \underline{w}_{c,s} \\ \underline{w}_{I,s} \end{bmatrix} \\ &= \begin{bmatrix} H_{c,s}\underline{w}_c \\ \underline{w}_{I,s} \end{bmatrix} \\ \underline{r} &= \begin{bmatrix} \underline{r}_c \\ \underline{r}_I \end{bmatrix} \\ \underline{r}_c &= \sum_s H_{c,s}^T \underline{r}_{c,s} \\ \underline{r}_I &= \begin{bmatrix} \underline{r}_{I,1} \\ \vdots \\ \underline{r}_{I,p} \end{bmatrix} \end{aligned}$$

Betrachtet man nun die Multiplikation eines Vektors mit der Matrix C^{-1} in ihrer Darstellung (18), so läßt sich diese Multiplikation zerlegen in drei Operationen. (entsprechend den drei Matrizen aus Gleichung (18) von rechts nach links)

1. $$\begin{bmatrix} \underline{r}_c - K_{cI}B_I^{-1}\underline{r}_I \\ \underline{r}_I \end{bmatrix} = \begin{bmatrix} \underline{y}_c \\ \underline{r}_I \end{bmatrix}$$

$$\Longrightarrow \qquad \underline{y}_c = \sum_s H_{c,s}^T \left(\underline{r}_{c,s} - K_{cI,s} B_{I,s}^{-1} \underline{r}_{I,s} \right)$$

d.h. für diese Operation braucht man nur auf den lokalen Vektor $\underline{r}_{c,s}$ den lokal berechenbaren Vektor $-K_{cI,s}B_{I,s}^{-1}\underline{r}_{I,s}$ addieren, und man hat den Vektor $\underline{r}_{c,s}$ durch das Ergebnis ersetzt. Also ist für diesen Schritt keine Kommunikation notwendig.

2. im unteren Teil: $\underline{v}_{I,s} = C_{I,s}^{-1}\underline{r}_{I,s}$ kann lokal berechnet werden.
   im oberen Teil: $\underline{w}_c = C_c^{-1}\underline{y}_c$ hier wird ein Datenaustausch über Γ_c notwendig.

   Es ist zu bemerken, daß $\underline{w}_c$ berechnet wird, jedoch für den Prozessor P_s ist nur $\underline{w}_{c,s}$ relevant. Für den Vorkonditionierer C_c^{-1} können z.B. Dryja oder BPS verwendet werden.

3. $$\underline{w}_{I,s} = \underline{v}_{I,s} - B_{I,s}^{-1}\left(K_{Ic,s}\underline{w}_{c,s}\right)$$

   kann ebenfalls rein lokal in jedem Prozessor berechnet werden.

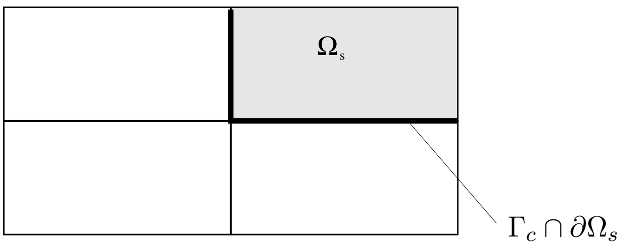
Was braucht man also hierfür:

- Einen Schur-Komplement Vorkonditionierer C_c
- $B_{I,s}$ zum definieren der näherungsweise diskret harmonischen Basis
- $C_{I,s}$ einen Vorkonditionierer für $K_{I,s}$

Bemerkung 18 (Zur Bedeutung der Matrix $B_{I,s}$).

$$\underline{w}_{c,s} \longmapsto B_{I,s}^{-1}K_{Ic,s}\underline{w}_{c,s} =: \underline{v}_{I,s}$$

Im Falle $B_{I,s} = K_{I,s}$ repräsentiert der Vektor $\begin{bmatrix} \underline{w}_{c,s} \\ -B_{I,s}^{-1}K_{Ic,s}\underline{w}_{c,s} \end{bmatrix}$ eine FE-Funktion mit vorgegebenen Werten auf dem Teilgebietsrand $\Gamma_c \cap \partial\Omega_s$ die mit minimaler Energie in das Innere Ω_s fortgesetzt ist.



Beweis. (zur Minimalität der Fortsetzung)

Sei $\underline{w}_c$ beliebig. Wir suchen dasjenige $\underline{w}_I$ welches für $w := \Phi \cdot \begin{bmatrix} \underline{w}_c \\ \underline{w}_I \end{bmatrix}$ das folgende Minimierungsproblem löst.

$$a(w,w) \longrightarrow \min_{\underline{w}_I}$$

$$\begin{aligned} a(w,w) &= \underline{w}^T K \underline{w} \\ &= \begin{bmatrix} \underline{w}_c \\ \underline{w}_I \end{bmatrix}^T \begin{bmatrix} K_c & K_{cI} \\ K_{Ic} & KI \end{bmatrix} \begin{bmatrix} \underline{w}_c \\ \underline{w}_I \end{bmatrix} \end{aligned}$$

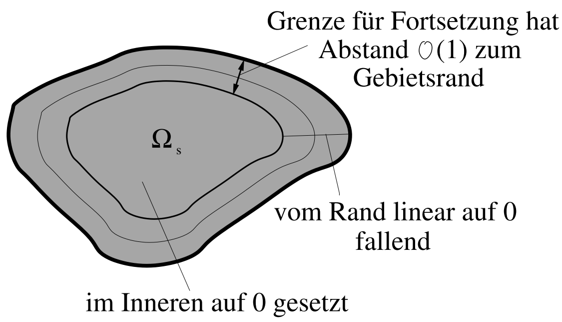
hier benutzen wir den Ansatz $\underline{w}_I = -K_I^{-1} K_{Ic} \underline{w}_c + \underline{v}_I$. Damit erhalten wir:

$$\begin{aligned} a(w,w) = {} & \begin{bmatrix} \underline{w}_c \\ -K_I^{-1} K_{Ic} \underline{w}_c \end{bmatrix}^T K \begin{bmatrix} \underline{w}_c \\ -K_I^{-1} K_{Ic} \underline{w}_c \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \underline{v}_I \end{bmatrix}^T K \begin{bmatrix} \underline{w}_c \\ -K_I^{-1} K_{Ic} \underline{w}_c \end{bmatrix}}_{= 0 \text{ da orthogonal}} \\ & + \underbrace{\begin{bmatrix} \underline{w}_c \\ -K_I^{-1} K_{Ic} \underline{w}_c \end{bmatrix}^T K \begin{bmatrix} 0 \\ \underline{v}_I \end{bmatrix}}_{= 0 \text{ da orthogonal}} + \underline{v}_I^T K_I \underline{v}_I \end{aligned}$$

□

Fazit:

- $X_{Ic} := B_I^{-1} K_{Ic}$ ist Fortsetzungsoperator von Funktionen auf dem Koppelrand ins Innere
- bestes X_{Ic} ist eines, das mit minimaler Energie fortsetzt
- B_I muß also billig zu implementieren sein, und trotzdem die obige Fortsetzung mit möglichst kleiner Energie realisieren
  z.B.: Nepomnjashchikh



3.5 Weitere parallele Vorkonditionierer mit ASM

3.5.1 Jacobi und Block-Jacobi Vorkonditionierer

Dazu betrachtet man die folgende Zerlegung:

$$\begin{aligned} \mathbb{V} &= \sum_{i=1}^{N} \mathbb{V}_i \\ \mathbb{V}_i &:= span(\varphi_i) \\ &= span(\Phi V_i) \end{aligned}$$

$$\Longrightarrow \qquad V_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i$$

$$V_i^T K V_i = k_{i,i}$$

Anwenden der ASM führt auf $C = diag(K)$ als Vorkonditionierer. ($\kappa(C^{-1}K) = \mathcal{O}(h^{-2})$)

Dieses Verfahren kann man auch Blockweise durchführen:
Seien $I_k \subset \{1, \ldots, N\}$ Indexmengen mit den Eigenschaften:

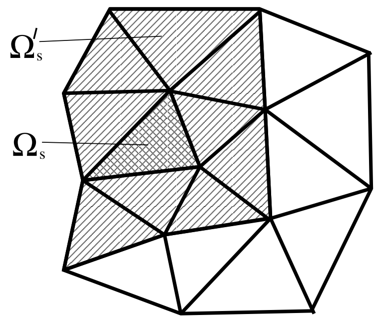
$$\begin{aligned} \bigcup_{k=1}^{p} I_k &= \{1, \ldots, N\} \\ I_k \cap I_{k'} &= \emptyset \qquad \forall k \neq k' \end{aligned}$$

Wir definieren die Teilräume $\mathbb{V}_k := span(\{\varphi_i, i \in I_k\})$. Die dazu gehörigen Transformationsmatrizen V_k sind jeweils einige Spalten der Einheitsmatrix im $\mathbb{R}^{N \times N}$. Damit sind dann die Matrizen $V_k^T K V_k$ so etwas wie Hauptdiagonalblöcke (konkret „Hauptuntermatrizen") von K. Der mit der ASM daraus entstehende Vorkonditionierer heißt Block-Jacobi-Vorkonditionierer.

3.5.2 Überlappende DD

Wir betrachten zunächst ein Grobnetz.
z.B. die bisherige DD: $\{\Omega_s\}$ ein Grobnetz. Wir definieren eine überlappende Zerlegung $\{\Omega'_s\}$ die die folgenden beiden Eigenschaften erfüllt:

$$\Omega_s \subset \Omega'_s\,, \qquad \Omega = \bigcup \Omega'_s$$



Wir definieren weiterhin den Grobgitterraum $\mathbb{V}_0 = \mathbb{V}_H$, z.B. stückweise lineare Funktionen auf dem Grobnetz. Die FE-Räume $\mathbb{V}_s$ zum Feinnetz $\mathbb{V}$ werden nun definiert durch $\mathbb{V}_s = \mathbb{V} \cap H^1(\Omega'_s)$.
Damit hat dann eine Funktion $v \in \mathbb{V}$ eine (nicht eindeutige) Zerlegung $v = \sum\limits_{s=0}^{p} v_s, \quad v_s \in \mathbb{V}_s$.
Resultat: Der durch die ASM entstehende Vorkonditionierer hat die Eigenschaft:

$$\kappa(P) = \kappa(C^{-1}K) \leq const$$

D.h. die Konditionszahl wird unabhängig von h (auch unabhängig von H.)
Die Implementierung dieses Verfahrens ist jedoch schwierig, da i.A. unsere FE-Datenzerlegung nicht anwendbar ist.

3.5.3 Hierarchische Vorkonditionierer nach YSERENTANT

(1988 veröffentlicht)
Idee:
Die bisher behandelten Vorkonditionierer waren immer faktorisiert gegeben: (siehe z.B. DD-Vorkonditionierer)

$$C^{-1} \;=\; UB^{-1}U^T$$

$$\begin{aligned} \underline{w} &:= C^{-1}\underline{r} \\ &= U\left(B^{-1}\left(U^T\underline{r}\right)\right) \end{aligned}$$

Das kann man so interpretieren, das immer erst ein Basiswechsel erfolgt, in der anderen Basis gerechnet wird, und dann zurücktransformiert wird.

Mit dieser Interpretation stellt sich die Frage, ob in $\mathbb{V} = span(\Phi)$ eine besonders günstige Basis Ψ existiert, in der die Konditionszahl der Steifigkeitsmatrix $\kappa(\hat{K})$ klein ist. ($\hat{K} := [a(\psi_j, \psi_i)]_{i,j=1}^N$)

Eine solche Basis ist z.B. die sogenannte Hierarchische Basis:

$$\Psi \quad = \quad [\underbrace{\psi_1, \ldots, \psi_{N_0}}_{\substack{\text{Knotenbasis}\\ \text{zum Grobnetz}}}, \quad \underbrace{\psi_{N_0+1}, \ldots, \psi_{N_1}}_{\substack{\underline{\text{neue}}\text{ Knoten}\\ \text{zu Level 1}}}, \quad \ldots, \quad \underbrace{\psi_{N_{L-1}+1}, \ldots, \psi_N}_{\text{Level } L}]$$

$$span(\Psi) = span(\Phi) = \mathbb{V}$$

$$\exists Q \in \mathbb{R}^{N \times N}: \quad \Psi = \Phi Q$$

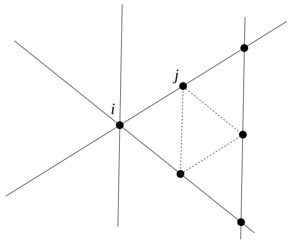
Zum Vergleich der beiden Basen: (für 2D Laplace-Problem)

Basis	Φ	Ψ
Steifigkeitsmatrix	$K := [a(\varphi_j, \varphi_i)]_{i,j=1}^N$	$\hat{K} := [a(\psi_j, \psi_i)]_{i,j=1}^N$
Anzahl NNE	$\sim N$	$\approx N^2$
Konditionszahl	$\mathcal{O}(h^{-2})$	$\mathcal{O}(L^2) = \mathcal{O}(\lvert \ln(h)\rvert^2)$

Es stellt sich die Frage ob man die Vorteile kombinieren kann.

$$\begin{aligned} \hat{K} &= Q^T K Q \\ \lambda_i(\hat{K}) &= \lambda_i(Q^T K Q) \\ &= \lambda_i(Q Q^T K) \\ &= \lambda_i(C^{-1} K) \\ \\ C^{-1} &= Q Q^T \end{aligned}$$

Wir nutzen also PCG für $K\underline{u} = \underline{b}$ mit dem oben definierten Vorkonditionierer C^{-1}. Die Anzahl der benötigten Iterationen ist dabei $\mathcal{O}(L)$. Dabei ist besonders zu bemerken das eine Multiplikation mit $C^{-1} = QQ^T$ nur $\sim N$ Flops kostet, wenn man hierfür einen "Vater-Sohn" Algorithmus verwendet.



Dazu betrachtet man die Verfeinerungsgleichung

$$\varphi_i^{(l-1)}(x) = \varphi_i^{(l)}(x) + \sum_{(i,j)} \alpha_{i,j}\varphi_j^{(l)}(x),$$

wobei die Summation über (i,j) bedeutet das über diejenigen Knotennummern j summiert wird die die im Level l "Sohn"-Knoten von Knoten i in Level $(l-1)$ sind. D.h. die Knoten j entstehen durch Verfeinerung eines Elementes welches Knoten i beinhaltet (siehe obige Skizze). Die Koeffizienten $\alpha_{i,j}$ hängen vom Typ der Elemente ab, für lineare Dreieckselemente $\alpha_{i,j} = 1/2$. In Blockmatrix-Form lautet obige Verfeinerungsgleichung

$$\Phi^{l-1} = \Phi^l \begin{bmatrix} I \\ P_{l-1} \end{bmatrix},$$

was für die Basiswechselmatrix

$$Q = \begin{bmatrix} I & 0 \\ P_{l-1} & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ P_{l-2} & I \end{bmatrix} \cdot \ldots \cdot \begin{bmatrix} I & 0 \\ P_0 & I \end{bmatrix}$$

ergibt.
Für die linearen Elemente kann man die Operationen $y = Q^T r$ bzw. $w = Qy$ danach durch folgende Algorithmen realisieren.

Algorithmus 3. *(Berechnen von $w = QQ^T r$)*

$y := r$
für alle Knoten in der Sohn→Väter Liste, feinster Level → gröbster

$$\begin{aligned} y(\text{Vater1}) &:= y(\text{Vater1}) + \tfrac{1}{2}y(\text{Sohn}) \\ y(\text{Vater2}) &:= y(\text{Vater2}) + \tfrac{1}{2}y(\text{Sohn}) \end{aligned}$$

end

$w := y$
für alle Knoten in der Sohn→Väter Liste, gröbster Level → feinster

$$w(\text{Sohn}) := w(\text{Sohn}) + \tfrac{1}{2}\left(w(\text{Vater1}) + w(\text{Vater2})\right)$$

end

Was ist dabei bei der Parallelisierung zu beachten?

geg:

- $\overline{\Omega} = \bigcup\limits_s \overline{\Omega}_s$ die DD-Datenverteilung
- $\underline{r} = \sum\limits_{s=1}^{p} H_s^T \underline{r}_s$
- im Prozessor P_s: $\underline{w}_s = H_s \underline{w}$
- eine lokal hierarchische Vernetzung, Liste der Vater-Sohn Beziehungen lokal gespeichert

Die drei wesentlichen Schritte hierbei:

1. Schritt: $\underline{y} := Q^T \underline{r}$, das heißt $\langle r, \varphi_i \rangle$ wird umgewandelt in $\langle r, \psi_i \rangle$
   $\underline{y} = \sum H_s^T \underline{y}_s$, man kann also die $\underline{y}_s$ rein lokal berechnen
2. Schritt: Umwandeln des verteilten Datentyps in Knotenwerte
   seien z.B. $\underline{\widetilde{y}} = \underline{y}$, die beiden Vektoren aber in unterschiedlicher Darstellung:

$$\begin{aligned} \underline{y} &= \sum H_s^T \underline{y}_s \\ \underline{\widetilde{y}}_s &= H_s \underline{\widetilde{y}} \end{aligned}$$

$$\Longrightarrow \quad \begin{aligned} \underline{\widetilde{y}}_s &= H_s \sum H_j^T \underline{y}_j \\ &= \underline{y}_s + \underbrace{\sum_{j \neq s} (H_s H_j^T) \underline{y}_j}_{\textit{von anderen Prozessoren}} \end{aligned}$$

3. Schritt: $\underset{\text{dargestellt bzgl. } \Phi}{\underline{w}} = \underset{\text{bezüglich } \Psi}{Q \, \underline{\widetilde{y}}}$

   Diese Berechnung kann rein lokal durchgeführt werden, auf dem Koppelrand berechnen dann mehrere Prozessoren das gleiche.

Fazit:

Vorteile:

1. Das Verfahren ist billig, nur $\sim N$ Operationen für C^{-1}
2. Es ist effizient. Die Anzahl der Iterationen ist lediglich proportional zum Verfeinerungslevel.
3. Es ist leicht parallelisierbar.
4. Es funktioniert für adaptive Vernetzung analog.

Aber: Die Eigenschaft, daß die Anzahl der Iterationen proportional zur Anzahl der Verfeinerungslevel ist, gilt nur in 2D. In 3D gilt (siehe 3.5.4)

$$\kappa(C^{-1}K) = \mathcal{O}(h^{-1})$$

mögliche Verbesserungen:

- bisher ist der Vorkonditionierer QQ^T nur vom Netz abhängig, also unabhängig von der PDE. Dies kann, z.B. bei $div(p(x)grad(u))$ mit springendem Koeffizient $p(x)$, von Nachteil sein.
  Als Ausweg kann man hier zusätzlich noch Jacobi einbauen:

$$\begin{array}{rrcl} & C^{-1} & = & J^{-1/2}QQ^TJ^{-1/2} \\ \text{oder} & C^{-1} & = & QJ^{-1}Q^T \\ \text{mit} & J & = & \text{diag}(K) \end{array}$$

- zusätzlicher Grobgitterlöser

$$C^{-1} = J^{-1/2}Q \begin{bmatrix} C_0^{-1} & \\ & I \end{bmatrix} Q^TJ^{-1/2}$$

  C_0 kann z.B. die Steifigkeitsmatrix zum Grobnetz oder die BPS-Näherung sein.

3.5.4 BPX - Vorkonditionierung

Die sehr gute Effizienz der hierarchischen Vorkonditionierung gilt leider nur in 2D:

$$\kappa(C^{-1}K) = \begin{cases} \mathcal{O}(|\ln h|^2) & in\,2D \\ \mathcal{O}(h^{-1}) & in\,3D \end{cases}$$

Deshalb entwickelte man Anfang der ’90er Jahre eine Verbesserung:

BPX (Bramble, Pasciak, Xu)

Definition 6.

$$\begin{array}{lll} \Phi^{(0)} & \textit{Knotenbasis in Level } 0 & \textit{Dimension } N_0 \\ \Phi^{(1)} & \textit{Knotenbasis in Level } 1 & \textit{Dimension } N_1 \\ \vdots & \vdots & \vdots \\ \Phi^{(L)} = \Phi & \textit{Knotenbasis in Level } L & \textit{Dimension } N_L = N \end{array}$$

Dabei sei $\Phi^{(l+1)}$ jeweils die Verfeinerung von $\Phi^{(l)}$. Weiterhin ist die Matrix $Q_l \in \mathbb{R}^{N\times N_l}$ jeweils definiert durch $\Phi^{(l)} = \Phi Q_l$.

Dadurch ergibt sich eine Struktur in den Q_l:

$$
\begin{array}{lcccl}
l = L & \Longrightarrow & Q_L & = & I \\[1em]
l = L-1 & \Longrightarrow & Q_l & = & \begin{bmatrix} I \\ P_{L-1} \end{bmatrix} \\[1em]
l = L-2 & \Longrightarrow & Q_l & = & \begin{bmatrix} I \\ P_{L-1} \end{bmatrix} \begin{bmatrix} I \\ P_{L-2} \end{bmatrix} \\
\vdots & & & &
\end{array}
$$

Betrachten wir nun auf Basis der ASM das Raumsplitting:

$$\mathbb{V} = \mathbb{V}_0 + \mathbb{V}_1 + \ldots + \mathbb{V}_L; \qquad \mathbb{V}_l = span(\Phi^{(l)})$$

Bemerkung 19. *Die dadurch entstehende Zerlegung einer Funktion $u \in \mathbb{V}$*

$$u = \sum_{l=0}^{L} u_l$$

in Teilraumfunktionen $u_l \in \mathbb{V}_l$ ist i.A. nicht eindeutig.

Definition 7 („stable splitting“ von $\{\mathbb{V}, a(.,.)\}$).
Sei $b_j(u,v)$ eine Bilinearform über $\mathbb{V}_j \times \mathbb{V}_j$ (Näherung für $a(.,.)$)

$$|||u|||^2 := \inf \sum_{j=0}^{L} b_j(u_j, u_j)$$

Dabei wird das Infimum über alle $[u_0, \ldots, u_L]$ mit $u = \sum_{j=0}^{L} u_j$ gebildet.
Das splitting $\{\mathbb{V}_j, b_j(.,.)\}$ heißt stabil (stable), wenn

$$\underline{\gamma} \leq \frac{a(u,u)}{|||u|||^2} \leq \overline{\gamma}$$

gilt, mit Konstanten $\underline{\gamma}, \overline{\gamma}$, unabhängig von h.

Satz 1. *Sei $\{\mathbb{V}_j, b_j(.,.)\}$ stable splitting von $\{\mathbb{V}, a(.,.)\}$, mit Basen $\Phi^{(l)}$. Dann ist durch*

$$w = \sum_{l=0}^{L} \sum_{j=1}^{N_l} \frac{\langle r, \varphi_j^{(l)} \rangle}{b_j(\varphi_j^{(l)}, \varphi_j^{(l)})} \varphi_j^{(l)}$$

ein Vorkonditionierer gegeben, mit $\kappa(C^{-1}K) = \mathcal{O}(1)$.

An dieser Stelle gleich noch ein paar Bemerkungen zum Begriff „stable splitting“:

Bemerkung 20. *Sei* $b_j(u,v) = a(u,v) \quad \forall j$*.* $\mathbb{V} = \sum_{j=0}^{L} \mathbb{V}_j$ *ist stable splitting genau dann, wenn* $|||u|||^2 = \inf_{alle\, splittings} \sum_{j=0}^{L} a(u_j,u_j) \sim a(u,u)$ *gilt.*

Beweis.
„$\Longleftarrow$": folgt direkt aus der Definition

„$\Longrightarrow$": betrachten $\underline{\gamma}$ und $\overline{\gamma}$
für beliebiges $u = \sum u_j$ gilt:

$$
\begin{aligned}
a(u,u) &\leq \overline{\gamma} \sum_j a(u_j,u_j) \qquad (19)\\
\Longleftrightarrow \quad \sum_i \sum_j a(u_i,u_j) &\leq \overline{\gamma} \sum_j a(u_j,u_j) \qquad (20)
\end{aligned}
$$

Aus der Chauchy-Schwarz'schen Ungleichung folgt,

$$\exists\, \Theta = [\theta_{ij}]_{i,j=0}^{L},\, \theta_{ij} \leq 1 : \quad a(u_i,u_j) \leq \theta_{ij} \sqrt{a(u_i,u_i) \cdot a(u_j,u_j)}.$$

Damit gilt

$$
\begin{aligned}
a(u,u) &= \sum_i a(u_i,u_i) + \sum_{i \neq j} a(u_i,u_j)\\
&\leq \sum_i a(u_i,u_i) + \sum_{i \neq j} \theta_{ij} \sqrt{a(u_i,u_i) \cdot a(u_j,u_j)}\\
&\leq (1+\rho(\Theta)) \sum_j a(u_j,u_j).
\end{aligned}
$$

Der letzte Schritt folgt aus der folgenden kleinen Nebenrechnung:

$$
\begin{aligned}
x &= [x_0,\ldots,x_L]^T \in \mathbb{R}^{L+1}\\
x_i &:= \sqrt{a(u_i,u_i)}\\
&\Longrightarrow\\
\sum_i a(u_i,u_i) + \sum_{i \neq j} \theta_{ij} \sqrt{a(u_i,u_i) \cdot a(u_j,u_j)} &= ((I+\Theta)x,x)\\
&\leq (1+\rho(\Theta))\,(x,x)
\end{aligned}
$$

somit erfüllt

$$\overline{\gamma} = 1 + \rho(\Theta) \leq L+1$$

die Ungleichung (20).
(Analogon zu $p+1$ bei $p+1$ Orthoprojektoren)

zu $\underline{\gamma}$:

$$\underline{\gamma} \inf \sum_j a(u_j,u_j) \leq a(u,u)$$

ist vorausgesetzt. Mit einem Vergleich zum Lemma von Lions (Lemma 1)

$$\sum a(u_j, u_j) \leq c_L^2 a(u, u) \qquad \forall u = \sum u_j$$

folgt die Behauptung mit $\underline{\gamma} = \frac{1}{c_L^2}$ wie vorher.

$\square$

Bemerkung 21. *Die Verallgemeinerung / Einführung der b_j in $\mathbb{V}_j$ ist der direkte Einbau der vorkonditionierten ASM.*

Bei Anwendung auf hierarchisch definierte Räume $\mathbb{V}_l = span(\Phi^{(l)})$ folgt

$$w = \sum_{l=0}^{L} \sum_j d_j^{(l)} \langle r, \varphi_j^{(l)} \rangle \varphi_j^{(l)}$$

mit $d_j^{(l)} = b_l(\varphi_j^{(l)}, \varphi_j^{(l)})^{-1}$.
Es ergeben sich zwei Möglichkeiten:

$$\begin{array}{rcll} b_l(u_l, u_l) & = & 2^{2l} \langle u_l, u_l \rangle_{L_2(\Omega)} & \Longrightarrow \text{ klassisches BPX} \\ b_l(u_l, u_l) & = & a(u_l, u_l) & \Longrightarrow \text{ MDS - BPX} \end{array}$$

Zur <u>Implementierung</u>:

$$\begin{array}{rcl} \underline{w} & := & C^{-1} \underline{r} \\ w & = & \Phi \underline{w} \\ \underline{r} & = & [\langle r, \varphi_i \rangle]_{i=1}^N \\ & & \\ w & = & \sum_{l=0}^{L} \Phi^{(l)} D_l \underline{r}^{(l)} \\ \underline{r}^{(l)} & = & \left[\langle r, \varphi_i^{(l)} \rangle \right]_{i=1}^{N_l} \\ & & \\ \Phi^{(l)} & = & \Phi Q_l \\ \Longrightarrow \quad \underline{r}^{(l)} & = & Q_l^T \underline{r} \\ w & = & \Phi \left(\sum_{l=0}^{L} Q_l D_l Q_l^T \right) \underline{r} \\ \Longrightarrow \quad \underline{w} & = & \sum_{l=0}^{L} Q_l D_l Q_l^T \underline{r} \\ C^{-1} & = & \sum_{l=0}^{L} Q_l D_l Q_l^T \end{array}$$

Zur Vereinfachung wurde dabei D_l verwendet, welches eine von der Raumdimension abhängige Konstante ist:

$$\begin{array}{rcll} D_l & = & 2^l \cdot I & \text{in 3D} \\ D_l & \sim & I & \text{in 2D} \end{array}$$

Dieses wird dann in 2 Schritten implementiert:

1. berechne $\underline{r}^{(l)} = Q_l^T \underline{r}$ für $l = L, L-1, \ldots, 0$

$$\begin{aligned}
\underline{r}^{(L)} &= \underline{r} \\
\underline{r}^{(L-1)} &= \begin{bmatrix} I \\ P_{L-1} \end{bmatrix}^T \underline{r}^{(L)} \\
&= \begin{bmatrix} I & P_{L-1}^T \end{bmatrix} \underline{r} \\
\underline{r}^{(L-2)} &= \begin{bmatrix} I & P_{L-2}^T \end{bmatrix} \underline{r}^{(L-1)} \\
\vdots \quad &\quad \vdots \\
\underline{r}^{(l-1)} &= \begin{bmatrix} I & P_{l-1}^T \end{bmatrix} \underline{r}^{(l)} \qquad l = L, \ldots, 1
\end{aligned}$$

2. summieren $\underline{w} = \sum_{l=0}^{L} Q_l(D_l \underline{r}^{(l)})$; $Q_l = \begin{bmatrix} I \\ P_{L-1} \end{bmatrix} \cdots \begin{bmatrix} I \\ P_{l-1} \end{bmatrix}$ mit dem Horner-schema:

$$\begin{aligned}
\underline{w}^{(0)} &:= D_0 \underline{r}^{(0)} \\
\underline{w}^{(1)} &:= \begin{bmatrix} I \\ P_0 \end{bmatrix} \underline{w}^{(0)} + D_1 \underline{r}^{(1)} \\
\vdots \quad &\quad \vdots \\
\underline{w}^{(l)} &:= \begin{bmatrix} I \\ P_{l-1} \end{bmatrix} \underline{w}^{(l-1)} + D_l \underline{r}^{(l)} \qquad l = 1, \ldots, L
\end{aligned}$$

Bemerkung 22. *Der Nachteil dieser Methode ist, das man Hilfsfelder benötigt. (Größe $\ll 2N$) Aber der Vorteil dürfte dies aufwiegen. Die Anzahl der benötigten Operationen pro Schritt ist $\sim N$.*

4 Das Fictitious Space Lemma (FSL) und seine Anwendungen

Der Sinn dieses Kapitels soll sein, einen Weg aufzuzeigen, wie man die für lineare Elemente entwickelten effizienten hierarchischen Techniken für Elemente höherer Ordnung verallgemeinert.

4.1 Das Fictitious Space Lemma

Satz 2 (Das FSL in algebraischer Form).
Voraussetzungen:

1. *Sei* $\mathbb{H} = \{\mathbb{R}^n, \langle .,. \rangle\}$ *ein Hilbertraum,* A *eine Abbildung* $A: \quad \mathbb{H} \longmapsto \mathbb{H}$, A *symmetrisch, positiv definit bezüglich* $\langle .,. \rangle$

2. *Sei* $\widetilde{\mathbb{H}} = \{\mathbb{R}^N, \langle .,. \rangle_\sim\}$ *ein Hilbertraum, der sogenannte Fictitious Space (i.A.* $N > n$*),* $\widetilde{A}$ *eine Abbildung* $\widetilde{A}: \quad \widetilde{\mathbb{H}} \longmapsto \widetilde{\mathbb{H}}$, $\widetilde{A}$ *symmetrisch, positiv definit bezüglich* $\langle .,. \rangle_\sim$

3. *Sei* R *eine Abbildung* $\widetilde{\mathbb{H}} \longmapsto \mathbb{H}$ *mit der Eigenschaft*
$$\langle AR\widetilde{x}, R\widetilde{x} \rangle \leq c_R \langle \widetilde{A}\widetilde{x}, \widetilde{x} \rangle_\sim \quad \forall \widetilde{x} \in \widetilde{\mathbb{H}}$$
für eine Konstante $c_R \in \mathbb{R}$

4. *Es existiert eine Abbildung* $T: \mathbb{H} \longmapsto \widetilde{\mathbb{H}}$ *mit* $RTx = x \quad \forall x \in \mathbb{H}$*, und*
$$c_T \langle \widetilde{A}Tx, Tx \rangle_\sim \leq \langle Ax, x \rangle \qquad \forall x \in \mathbb{H}$$
für eine Konstante $c_T \in \mathbb{R}$

Dann gilt:

$$c_T \langle A^{-1}x, x \rangle \leq \langle R\widetilde{A}^{-1}R^*x, x \rangle c_R \leq \langle A^{-1}x, x \rangle \qquad \forall x \in \mathbb{H} \tag{21}$$

Bemerkung 23.

$$\langle R\widetilde{x}, x \rangle = \langle \widetilde{x}, R^*x \rangle_\sim \qquad \forall\, \widetilde{x} \in \widetilde{\mathbb{H}}, x \in \mathbb{H}$$

Folgerung 2. *Wenn* $\widetilde{C}^{-1}$ *guter Vorkonditionierer für* $\widetilde{A}$ *ist, so ist* $C^{-1} = R\widetilde{C}^{-1}R^*$ *guter Vorkonditionierer für* A*.*

Beweis. (zum FSL)

1. rechte Ungleichung in (21)

$$\begin{aligned}
\langle R\widetilde{A}^{-1}R^*x,x\rangle^{\frac{1}{2}} &=: ||\widetilde{A}^{-\frac{1}{2}}R^*x||_\sim \\
&= \max_{\widetilde{y}\in\widetilde{\mathbb{H}}} \frac{\langle \widetilde{A}^{-\frac{1}{2}}R^*x,\widetilde{y}\rangle_\sim}{||\widetilde{y}||_\sim} \qquad |\widetilde{v} := \widetilde{A}^{-\frac{1}{2}}\widetilde{y} \\
&= \max_{\widetilde{v}\in\widetilde{\mathbb{H}}} \frac{\langle R^*x,\widetilde{v}\rangle_\sim}{\langle \widetilde{A}\widetilde{v},\widetilde{v}\rangle_\sim^{\frac{1}{2}}} \\
&= \max_{\widetilde{v}\in\widetilde{\mathbb{H}}} \frac{\langle A^{-\frac{1}{2}}x,A^{\frac{1}{2}}R\widetilde{v}\rangle}{\langle \widetilde{A}\widetilde{v},\widetilde{v}\rangle_\sim^{\frac{1}{2}}} \\
&\leq ||A^{-\frac{1}{2}}x|| \max_{\widetilde{v}\in\widetilde{\mathbb{H}}} \frac{\langle AR\widetilde{v},R\widetilde{v}\rangle^{\frac{1}{2}}}{\langle \widetilde{A}\widetilde{v},\widetilde{v}\rangle_\sim^{\frac{1}{2}}} \\
&\leq \sqrt{c_R}\langle A^{-1}x,x\rangle^{\frac{1}{2}}
\end{aligned}$$

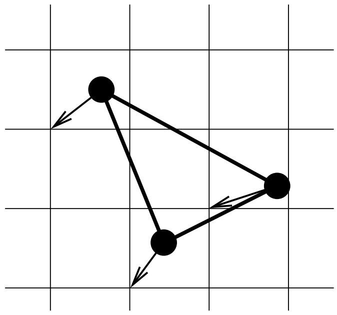
2. linke Ungleichung in (21)

$$\begin{aligned}
||A^{-\frac{1}{2}}x|| &= \max_{y\in\mathbb{H}} \frac{\langle A^{-\frac{1}{2}}x,y\rangle}{||y||} \qquad |z = A^{-\frac{1}{2}}y \\
&= \max_{z\in\mathbb{H}} \frac{\langle x,z\rangle}{||A^{\frac{1}{2}}z||} \\
&= \max_{z\in\mathbb{H}} \frac{\langle x,RTz\rangle}{||A^{\frac{1}{2}}z||} \\
&= \max_{z\in\mathbb{H}} \frac{\langle \widetilde{A}^{-\frac{1}{2}}R^*x,\widetilde{A}^{\frac{1}{2}}Tz\rangle_\sim}{||A^{\frac{1}{2}}z||} \\
&\leq ||\widetilde{A}^{-\frac{1}{2}}R^*x||_\sim \max_{z\in\mathbb{H}} \frac{||\widetilde{A}^{\frac{1}{2}}Tz||_\sim}{||A^{\frac{1}{2}}z||} \\
&\leq \frac{1}{\sqrt{c_T}}\langle R\widetilde{A}^{-1}R^*x,x\rangle^{\frac{1}{2}}
\end{aligned}$$

□

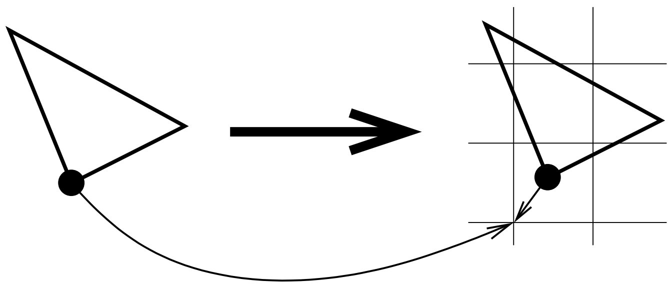
4.2 Anwendung bei gegebenen nichthierarchischen Netzen

Es sei ein beliebiges Netz gegeben. Wir definieren ein gleichmäßiges Vierecksnetz (fictitious space) so, das in jeder Gitterzelle maximal ein Knoten des Ausgangsnetzes enthalten ist.



Die noch benötigten Abbildungen R und T werden wie folgt definiert:

R: Der Wert der Gitterzelle ist gleich dem Wert des enthaltenen Knotens, und gleich 0 falls kein Knoten enthalten ist.



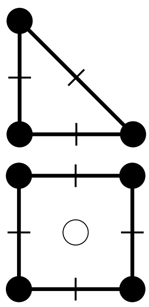
T: Der Wert im Gitterknoten ist gleich dem der zugehörigen Gitterzelle.

Mit diesem Trick kann man im fictitious space hierarchische Techniken anwenden.

4.3 Hierarchische Techniken für Elemente mit höherem Polynomgrad

Wir gehen hier von den geschachtelten Teilräumen $\mathbb{V}_0 \subset \mathbb{V}_1 \subset \ldots \subset \mathbb{V}_L$ und der zugehörigen Verfeinerungsformel aus.

4.3.1 Quadratische Elemente



6-Knoten Dreieck

8 bzw. 9-Knoten Viereck

Die durch diese Elemente definierten FEM Ansatzfunktionen sind (stückweise) quadratisch. Weiterhin gilt:

$$\varphi_i(x^{(j)}) = \delta_{ij}$$

Die geschachtelten Teilräume werden so definiert, das sich folgendes Schema ergibt:

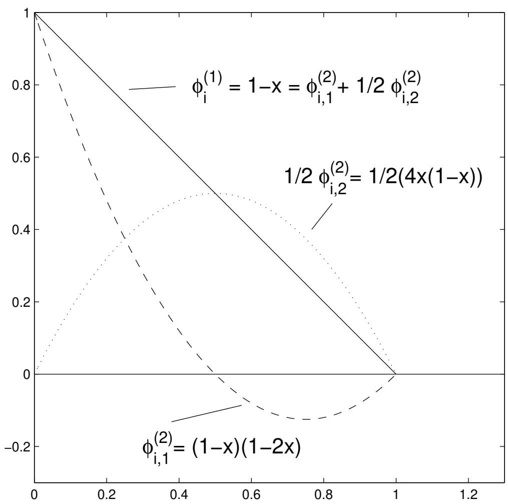
$$\underbrace{\mathbb{V}_0^{(1)} \subset \mathbb{V}_1^{(1)} \subset \ldots \subset \mathbb{V}_L^{(1)}}_{\text{zu linearen Funktionen}} \subset \quad \mathbb{V}_L^{(2)}$$

Wenn wir für die linearen Funktionen z.B. die zu 3-Knoten Dreiecken gehörenden Funktionen verwenden, so wissen wir bereits wie wir auf diesem Teil hierarchisch arbeiten können. Interessant ist hier nur der Übergang $\mathbb{V}_L^{(1)} \subset \mathbb{V}_L^{(2)}$.

Hier gilt die Verfeinerungsgleichung:

$$\varphi_i^{(1)}(x) = \varphi_i^{(2)} + \sum_j \alpha_{ij}\varphi_j^{(2)}$$

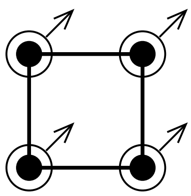
mit $|\alpha_{ij}| = \frac{1}{2}$, z.B. in 1D, in $[0, 1]$:



$$\begin{aligned}
\varphi_i^{(2)}(x) + \frac{1}{2}\varphi_j^{(2)}(x) &= (1-x)(1-2x) + \frac{1}{2}(4x(1-x)) \\
&= (1-x)(1-2x+2x) \\
&= 1-x \\
&= \varphi_i^{(1)}(x)
\end{aligned}$$

Damit hat man also auch die Verfeinerungsformel, und die hierarchischen Techniken funktionieren ohne Änderung.

4.3.2 Bogner - Fox - Schmidt - Elemente



Hier geht es um bikubische Ansatzfunktionen für Rechteckelemente. Jedem Knoten werden die vier Freiheitsgrade $[u, u_x, u_y, u_{xy}]$ zugeordnet. D.h., es gibt auch vier Basisfunktionen pro Knoten.

$$\Phi = [\varphi_{0,1}, \varphi_{1,1}, \varphi_{2,1}, \varphi_{3,1}, \varphi_{0,2}, \ldots]$$

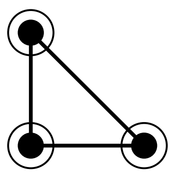
Mit der üblichen Definition der Teilräume $\mathbb{V}_0 \subset \ldots \subset \mathbb{V}_L$ ergibt sich eine Verfeinerungsformel der Form:

$$\left[\varphi_{0,i}^{(l-1)}, \varphi_{1,i}^{(l-1)}, \varphi_{2,i}^{(l-1)}, \varphi_{3,i}^{(l-1)}\right] = \left[\varphi_{0,i}^{(l)}, \varphi_{1,i}^{(l)}, \varphi_{2,i}^{(l)}, \varphi_{3,i}^{(l)}\right] + \sum_j \left[\varphi_{0,j}^{(l)}, \varphi_{1,j}^{(l)}, \varphi_{2,j}^{(l)}, \varphi_{3,j}^{(l)}\right] B_j$$

Damit sind die hierarchischen Techniken wieder analog anwendbar.

4.3.3 Nichtgeschachtelte Teilräume am Beispiel der Hermite Dreiecke

Es geht also um eine Dreiecksvernetzung der Ebene mit drei Freiheitsgraden pro Knoten:



$$u_i = u(x^{(i)}) \qquad \Theta_i = \nabla u|_{x^{(i)}} \in \mathbb{R}^2$$

Die Ansatzfunktionen sind also stückweise bikubisch. Der Gradient ist in jedem Knoten stetig.
Damit gilt $\mathbb{V}_{L-1}^{HC} \not\subset \mathbb{V}_L^{HC}$ und auch $\mathbb{V}_L^{(1)} \not\subset \mathbb{V}_L^{HC}$. Aber man kann auf folgende Schachtelung das FSL anwenden:

$$\mathbb{V}_L^{HC} \subset \mathbb{V}_L^{(3)}$$

$\mathbb{V}_L^{(3)}$ sind die stückweise bikubischen Funktionen mit Freiheitsgraden als Funktionswerte und Ableitungen in Kantenrichtung, jeweils in den Knoten(Funktionswert u_i und $\frac{\partial u}{\partial \tau_{i,j}}|_i \quad \forall$ Kanten(i,j)).

Also gilt $dim(\mathbb{V}_L^{(3)}) \sim 7n$ und $dim(\mathbb{V}_L^{HC}) \sim 3n$.
Natürlich gilt $\mathbb{V}_L^{(1)} \subset \mathbb{V}_L^{(3)}$ mit einer entsprechenden Verfeinerungsformel.

Es läßt sich also über die Raumschachtelung

$$\mathbb{V}_0^{(1)} \subset \mathbb{V}_1^{(1)} \subset \ldots \subset \mathbb{V}_L^{(1)} \subset \mathbb{V}_L^{(3)}$$

ein Vorkonditionierer für $\mathbb{V}_L^{(3)}$ finden. Mit dem FSL dann auch für $\mathbb{V}_L^{HC}$.