

### § 13 Aspekte der Implementierung

Die Lösung einer (linearen) PDE mit der FEM gliedert sich in folgende Unterpunkte:

- (1) Modellierung des Gebietes (z. B. mit CAD) und Erzeugung eines Gitters (mit einem Gittergenerator)
- (2) Entscheidung für einen FE-Typ
- (3) Entscheidung für die Äquivalenzrelation unter den Freiheitsgraden
- (4) Organisation der globalen Freiheitsgrade
- (5) Assemblierung der Steifigkeitsmatrix und des Lastvektors
- (6) Einarbeiten von Dirichlet-Randbedingungen und evtl. weiterer Bedingungen (z. B. aus der Behandlung von hängenden Freiheitsgraden)<sup>102</sup>
- (7) Lösen des linearen Gleichungssystems
- (8) Abschätzung des Fehlers
- (9) ggf. Gitterverfeinerung und zurück zu (4)
- (10) Postprocessing und Darstellung der Lösung

Verschiedene FE-Bibliotheken setzen diese Schritte sehr unterschiedlich um. Manchmal sind z. B. (2) und (3) untrennbar verbunden, etwa bei Auswahl von Lagrange-Elementen  $\mathbb{P}_k$  oder  $\mathbb{Q}_k$  immer die Standard- $\mathbb{P}_k$ -Räume bzw. Standard- $\mathbb{Q}_k$ -Räume verwendet werden, also  $H^1$ -konforme (stetige) globale Basisfunktionen, vgl. [Beispiel 12.12](#). Bei nichtlinearen Aufgaben (PDEs) sind die Schritte (5)–(7) noch in eine Newton-Schleife eingeschlossen.

Es liege aus Schritt (1)–(4) bereits ein (nicht notwendig konformes) Gitter  $\mathcal{T}$  und (für die Vereinfachung der folgenden Beschreibung) eine affine Familie  $\{(K, P_K, \Sigma_K)\}_{K \in \mathcal{T}}$  von Elementen auf Simplex-Zellen vor (z. B.  $\mathbb{P}_k$ -Elemente). Es sei  $V_h$  ein FE-Raum wie in [Definition 12.11](#) unter Beachtung von [\(MAX1\)](#).

#### § 13.1 Assemblierung der schwachen Form

Wir betrachten nun Punkt (5) genauer. Im Fall der Poisson-Gleichung lautet die Steifigkeitsmatrix (vgl. [§ 10](#))

$$(A)_{ij} = a[\varphi_j, \varphi_i] = \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i \, dx = \sum_{K \in \mathcal{T}} \int_K \nabla \varphi_j \cdot \nabla \varphi_i \, dx,$$

wobei  $\{\varphi_i\}_{i=1}^M$  die globalen Formfunktionen (Basisfunktionen von  $V_h$ ) sind. Nach [Lemma 12.13](#) stimmt  $\varphi_i$  auf jeder Zelle  $K$  mit genau einer der lokalen Formfunktion  $p_{K,1}, \dots, p_{K,s}$  überein oder ist dort null. Umgekehrt trägt jede lokale Formfunktion  $p_{K,1}, \dots, p_{K,s}$  auf  $K$  zu genau einer globalen Formfunktion  $\varphi_i$  bei. Zu welcher, das

<sup>102</sup>Dieser Schritt kann auch durch spezielle Vorkehrungen im iterativen Gleichungssystemlöser realisiert werden.

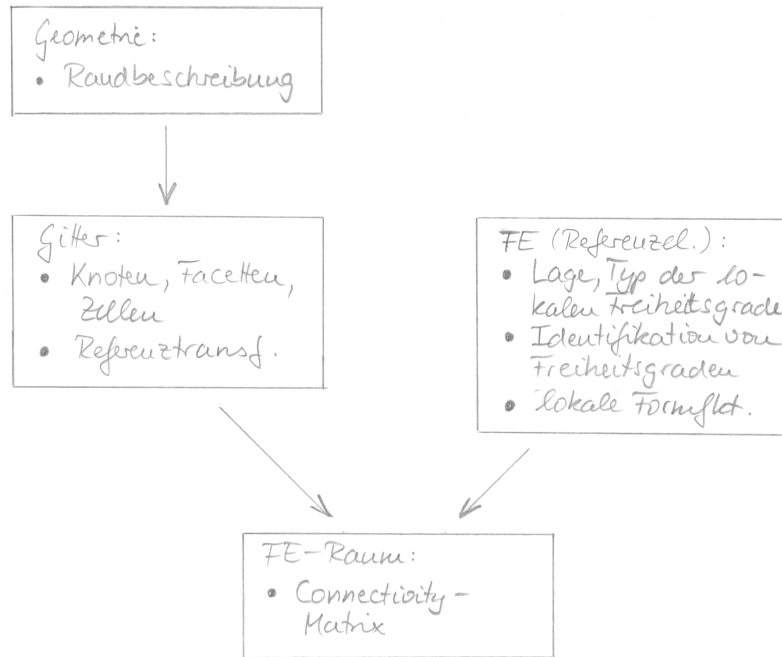


ABBILDUNG 13.1. typische Datenstrukturen in FE-Bibliotheken

ist in der sogenannten **Connectivity-Matrix**  $C_K$  der Zelle  $K$  festgehalten, die in Unterpunkt (4) aufgestellt wird:<sup>103</sup> Es gilt  $C_K \in \{0, 1\}^{M \times s}$  und

$$C_K(n_{\text{global\_dof}}, n_{\text{dof}}) = 1 \Leftrightarrow \begin{cases} \text{der lokale Freiheitsgrad/die lokale Formfkt.} \\ \text{mit der Nummer } n_{\text{dof}} \text{ auf der Zelle } K \text{ ist der} \\ \text{globale Freiheitsgrad/die globale Formfkt.} \\ \text{der Nummer } n_{\text{global\_dof}} \end{cases}$$

$$\Leftrightarrow \varphi_{n_{\text{global\_dof}}|K} = p_{n_{\text{dof}}}.$$

Jede Matrix  $C_K$  hat insgesamt  $s$  Einsen und wegen (MAX1) in jeder Zeile höchstens eine. Die restlichen Einträge sind null.

**Beachte:**  $C_K$  realisiert die Zuordnung der Freiheitsgrade lokal  $\mapsto$  global,  $C_K^\top$  dagegen global  $\mapsto$  lokal für die Zelle  $K$ . Es gilt (in MATLAB-Notation)

$$\sigma_{K,i} = \underbrace{C_K(:, i)}_{i\text{-te Spalte von } C_K}^\top \sigma, \quad i = 1, \dots, s,$$

wobei  $\sigma \in [V_h^*]^M$  der Vektor der globalen Freiheitsgrade ist, und außerdem

$$C_K^\top C_K = I \in \mathbb{R}^{s \times s}$$

$$C_K C_K^\top = \text{diag}(d) \in \mathbb{R}^{M \times M},$$

<sup>103</sup>Beispiele dazu in der Übung

mit

$$d_{n_{\text{global\_dof}}} = \begin{cases} 1, & \text{falls } n_{\text{global\_dof}} \text{ auf der Zelle } K \text{ vertreten ist} \\ 0 & \text{sonst.} \end{cases}$$

Die Wahl anderer Einträge als eins in  $C_K$  ermöglicht eine Skalierung.

Bei der Assemblierung berechnet man die **lokale Steifigkeitsmatrix** oder **Elementsteifigkeitsmatrix**  $A_K$  auf jeder Zelle  $K$

$$A_K = (a_{k,\ell})_{k,\ell=1}^s = \int_K \nabla p_\ell \cdot \nabla p_k \, dx \quad (13.1)$$

und addiert das Ergebnis zur globalen Steifigkeitsmatrix:<sup>104</sup>

$$A = \sum_{K \in \mathcal{T}} C_K A_K C_K^\top. \quad (13.2)$$

Analog gilt für den **lokalen Lastvektor** (**Elementlastvektor**)  $F_K$

$$F_K = (f_k)_{k=1}^s = \int_K f p_k \, dx, \quad (13.3)$$

den man zum globalen Lastvektor addiert:

$$\vec{F} = \sum_{K \in \mathcal{T}} C_K F_K. \quad (13.4)$$

### Bemerkung 13.1 (Alternative Form der Connectivity-Matrix)

Eine alternative Form der Organisation der Freiheitsgrade verwendet nur eine *einzige* Connectivity-Matrix  $\hat{C} \in \mathbb{N}^{s \times N_{\text{cells}}}$  für das gesamte Gitter:

$$\begin{aligned} \hat{C}(n_{\text{dof}}, n_{\text{cell}}) = i & \Leftrightarrow \begin{cases} \text{der lokale Freiheitsgrad/die lokale Formfkt.} \\ \text{mit der Nummer } n_{\text{dof}} \text{ auf der Zelle } n_{\text{cell}} \text{ ist der} \\ \text{globale Freiheitsgrad/die globale Formfkt. der Nummer } i \end{cases} \\ & \Leftrightarrow \varphi_{i|K_{n_{\text{cell}}}} = p_{K_{n_{\text{cell}}}, n_{\text{dof}}}. \end{aligned} \quad (13.5)$$

Die Matrix  $\hat{C}$  ist vollbesetzt. Die elementweisen Beiträge auf der Zelle  $K = K_{n_{\text{cell}}}$  werden in diesem Fall wie folgt zur globalen Steifigkeitsmatrix bzw. zum globalen Lastvektor addiert:

$$\begin{aligned} A_{\hat{C}(k, n_{\text{cell}}), \hat{C}(l, n_{\text{cell}})} &:= A_{\hat{C}(k, n_{\text{cell}}), \hat{C}(l, n_{\text{cell}})} + a_{k,\ell}, & k, \ell = 1, \dots, s \\ F_{\hat{C}(k, n_{\text{cell}})} &:= F_{\hat{C}(k, n_{\text{cell}})} + f_k, & k = 1, \dots, s. \end{aligned} \quad \diamond$$

**Frage:** Wie berechnet man die elementweisen Beiträge (13.1) und (13.3)?

Es sei  $T_K : \hat{K} \ni \hat{x} \mapsto B_K \hat{x} + b_K \in K$  wieder eine bijektive affine Abbildung. Aufgrund der Substitutionsregel gilt

$$\int_K g(x) \, dx = \int_{\hat{K}} g(T_K(\hat{x})) |\det T'_K(\hat{x})| \, d\hat{x} = |\det B_K| \int_{\hat{K}} g(T_K(\hat{x})) \, d\hat{x} \quad (13.6)$$

<sup>104</sup>Dies kann in MATLAB elegant mit dem `sparse`-Befehl realisiert werden. Es gilt weiterhin:  $\text{diag}(A) = \sum_{K \in \mathcal{T}} C_K \text{diag}(A_K) C_K^\top$ .

für integrierbare Funktionen  $g$ . Die Anwendung auf (13.1) ergibt<sup>105</sup>

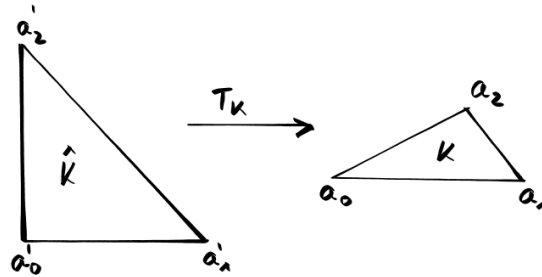
$$\begin{aligned}
 & \int_K \nabla p_\ell(x) \cdot \nabla p_k(x) \, dx \\
 &= \int_K \nabla(\hat{p}_\ell \circ T_K^{-1})(x) \cdot \nabla(\hat{p}_k \circ T_K^{-1})(x) \, dx && \text{Konstruktion der } p_k \\
 &= \int_K [B_K^{-\top} \hat{\nabla} \hat{p}_\ell(T_K^{-1}(x))] \cdot [B_K^{-\top} \hat{\nabla} \hat{p}_k(T_K^{-1}(x))] \, dx && \text{Kettenregel} \\
 &= |\det B_K| \int_{\hat{K}} [B_K^{-\top} \hat{\nabla} \hat{p}_\ell(\hat{x})] \cdot [B_K^{-\top} \hat{\nabla} \hat{p}_k(\hat{x})] \, d\hat{x} && \text{Substitutionsregel.}
 \end{aligned}$$

Der besseren Unterscheidung wegen bezeichnen wir den Gradienten einer Funktion auf der Referenzzelle  $\hat{K}$  mit  $\hat{\nabla}$ . Die Transformationsmatrix  $B_K$  auf das Einheitssimplex  $\hat{K}$  wird für jede (Simplex-)Zelle  $K$  wie folgt bestimmt: Ist

$$K = \text{conv}\{a_0, a_1, \dots, a_d\} \subset \mathbb{R}^d, \quad a_i \in \mathbb{R}^d$$

dann gilt

$$B_K = [a_1 - a_0, a_2 - a_0, \dots, a_d - a_0] \in \mathbb{R}^{d \times d}, \quad b_K = a_0 \in \mathbb{R}^d.$$



Aufgrund der (häufig anzutreffenden) Polynomeigenschaft von  $\hat{p}_k$  kann das obige Integral exakt berechnet werden. Bei Operatoren mit nicht-konstanten Koeffizienten, etwa

$$a[u, v] = \int_{\Omega} \nabla v(x)^\top A(x) \nabla u(x) \, dx, \quad (*)$$

ist das aber i. d. R. nicht möglich. Deshalb setzt man eine **Quadraturformel (Q-Formel)**

$$\int_{\hat{K}} \hat{g}(\hat{x}) \, d\hat{x} \approx \sum_{m=1}^q \omega_{\hat{K},m} \hat{g}(\xi_{\hat{K},m})$$

auf  $\hat{K}$  ein mit **Gewichten**  $\omega_{\hat{K},m}$  und **Stützstellen**  $\xi_{\hat{K},m}$ . Die **Ordnung**  $r \in \mathbb{N}$  der Formel ist der maximale Grad, sodass für alle Polynome  $\hat{g} \in P_r(\hat{K})$  Gleichheit gilt (analog für Randintegrale).

<sup>105</sup>**Beachte:** Die Ableitung (Jacobimatrix) ist  $J(\hat{p}_k \circ T_K^{-1})(x) = J(\hat{p}_k)(T_K^{-1}(x)) B_K^{-1}$ . Der Gradient ist die Transposition davon.

Die Anwendung auf (\*) ergibt:

$$\begin{aligned}
 \int_K \nabla p_\ell(x)^\top A(x) \nabla p_k(x) dx &= \dots \\
 &= |\det B_K| \int_{\hat{K}} [B_K^{-\top} \hat{\nabla} \hat{p}_\ell(\hat{x})]^\top A(T_K(\hat{x})) [B_K^{-\top} \hat{\nabla} \hat{p}_k(\hat{x})] d\hat{x} \\
 &\approx |\det B_K| \sum_{m=1}^q \omega_{\hat{K},m} [B_K^{-\top} \hat{\nabla} \hat{p}_\ell(\xi_{\hat{K},m})]^\top A(T_K(\xi_{\hat{K},m})) [B_K^{-\top} \hat{\nabla} \hat{p}_k(\xi_{\hat{K},m})].
 \end{aligned}$$

schwache Form	lokale Beiträge der Zelle $K$
$\int_\Omega \nabla \varphi_j^\top A(x) \nabla \varphi_i dx$	$ \det(B_K)  \sum_{m=1}^q \omega_{\hat{K},m} [B_K^{-\top} \hat{\nabla} \hat{p}_\ell(\xi_{\hat{K},m})]^\top A(T_K(\xi_{\hat{K},m})) [B_K^{-\top} \hat{\nabla} \hat{p}_k(\xi_{\hat{K},m})]$
$\int_\Omega \nabla \varphi_j \cdot \beta(x) \varphi_i dx$	$ \det(B_K)  \sum_{m=1}^q \omega_{\hat{K},m} [B_K^{-\top} \hat{\nabla} \hat{p}_\ell(\xi_{\hat{K},m})] \cdot \beta(T_K(\xi_{\hat{K},m})) \hat{p}_k(\xi_{\hat{K},m})$
$\int_\Omega \varphi_j c_0(x) \varphi_i dx$	$ \det(B_K)  \sum_{m=1}^q \omega_{\hat{K},m} \hat{p}_\ell(\xi_{\hat{K},m}) c_0(T_K(\xi_{\hat{K},m})) \hat{p}_k(\xi_{\hat{K},m})$
$\int_\Omega f(x) \varphi_i dx$	$ \det(B_K)  \sum_{m=1}^q \omega_{\hat{K},m} f(T_K(\xi_{\hat{K},m})) \hat{p}_k(\xi_{\hat{K},m})$

**Beachte:** Es gilt

$$|\det(B_K)| = \frac{|K|}{|\hat{K}|}, \quad \text{denn} \quad \int_K 1 dx = \int_{\hat{K}} 1 |\det(B_K)| d\hat{x}. \quad (13.7)$$

Folgende Daten können tabelliert (vorberechnet) werden:

- Transformationsmatrix  $B_K$  und Vektor  $b_K$  für jede Zelle sowie  $B_K^{-\top} \in \mathbb{R}^{d \times d}$  und evtl.  $\det B_K$  (gehören zum Gitter)

Speicheraufwand:  $N_{\text{cells}} \times (2d + 1)$  Vektoren des  $\mathbb{R}^d$

- die Funktionswerte  $\hat{p}_k(\xi_{\hat{K},m})$  und  $\hat{\nabla} \hat{p}_k(\xi_{\hat{K},m})$  der lokalen Formfunktionen *auf der Referenzzelle  $\hat{K}$* ,  $k = 1, \dots, s$  und  $m = 1, \dots, q$  (gehört zum FE bzw. zur Q-Formel)

Speicheraufwand:  $q \times s$  Skalare und  $q \times s$  Vektoren des  $\mathbb{R}^d$

**Beachte:** Es bietet sich nicht an,  $B_K^{-\top} \hat{\nabla} \hat{p}_k(\xi_{\hat{K},m})$  zu speichern, denn dies sind  $N_{\text{cells}} \times q \times s$  Vektoren des  $\mathbb{R}^d$ ! (Ersparnisse ergeben sich nur, wenn viele Zellen nur Translate voneinander sind.)

Die Randintegrale in der schwachen Formulierung werden analog berechnet. Wir betrachten als Beispiel

$$\int_\Gamma \varphi_j(s) \alpha(s) \varphi_i(s) ds = \sum_{K \in \mathcal{T}} \int_{\partial K \cap \Gamma} \varphi_j(s) \alpha(s) \varphi_i(s) ds.$$

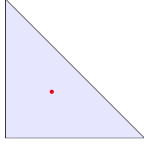
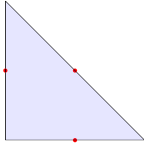
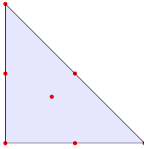
$r$	$q$		Koordinaten	Anzahl	Gewichte
1	1		$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	1	$ \widehat{K} $
2	3		$(\frac{1}{2}, \frac{1}{2}, 0)$	3	$\frac{1}{3} \widehat{K} $
3	7		$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ $(\frac{1}{2}, \frac{1}{2}, 0)$ $(1, 0, 0)$	1 3 3	$\frac{9}{20} \widehat{K} $ $\frac{2}{15} \widehat{K} $ $\frac{1}{20} \widehat{K} $

TABELLE 13.1. Quadraturformeln auf Dreiecken mit baryzentrischen Koordinaten. Bei Anzahlen  $> 1$  ergeben sich die Koordinaten der weiteren Quadraturpunkte durch zyklische Vertauschung.

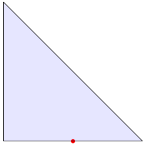
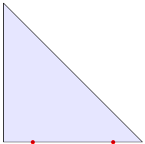
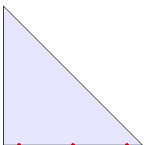
$r$	$q$		Koordinaten	Anzahl	Gewichte
1	1		$(\frac{1}{2}, \frac{1}{2}, 0)$	1	$ \widehat{F}_j $
3	2		$(\frac{1}{2} + \frac{1}{6}\sqrt{3}, \frac{1}{2} - \frac{1}{6}\sqrt{3}, 0)$ $(\frac{1}{2} - \frac{1}{6}\sqrt{3}, \frac{1}{2} + \frac{1}{6}\sqrt{3}, 0)$	1 1	$\frac{1}{2} \widehat{F}_j $ $\frac{1}{2} \widehat{F}_j $
5	3		$(\frac{1}{2} + \frac{1}{2}\sqrt{\frac{3}{5}}, \frac{1}{2} - \frac{1}{2}\sqrt{\frac{3}{5}}, 0)$ $(\frac{1}{2} - \frac{1}{2}\sqrt{\frac{3}{5}}, \frac{1}{2} + \frac{1}{2}\sqrt{\frac{3}{5}}, 0)$ $(\frac{1}{2}, \frac{1}{2}, 0)$	1 1 1	$\frac{5}{18} \widehat{F}_j $ $\frac{5}{18} \widehat{F}_j $ $\frac{8}{18} \widehat{F}_j $

TABELLE 13.2. Quadraturformeln auf Dreiecksanten mit baryzentrischen Koordinaten. Formeln für die anderen Kanten entstehen jeweils durch zyklisches Vertauschen der Koordinaten.

Man berechnet wieder die lokalen Beiträge auf der Zelle  $K$ <sup>106</sup>

$$(a_{k,\ell})_{k,\ell=1}^s = \int_{\partial K \cap \Gamma} p_\ell(s) \alpha(s) p_k(s) \, ds.$$

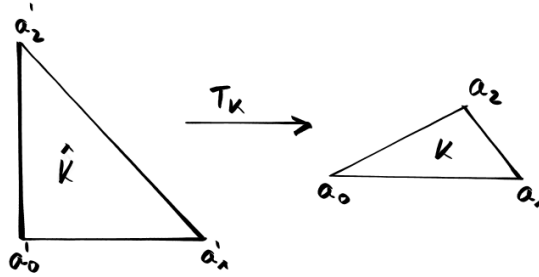
<sup>106</sup>Man könnte hier auch facettenweise vorgehen.

**Beachte:**  $\partial K \cap \Gamma$  ist entweder  $\emptyset$  oder besteht aus einer (oder mehreren) Facetten  $F_j$  von  $K$ . Dies muss ebenfalls in einer zum Gitter gehörenden Datenstruktur festgehalten werden.

Man transformiert das Integral (z. B. mittels  $T_K$ ) auf eine Facette des Referenzelements und setzt eine Q-Formel<sup>107</sup> ein:

$$\begin{aligned} \int_{F_j} p_\ell(s) \alpha(s) p_k(s) ds &= \frac{|F_j|}{|\widehat{F}_j|} \int_{\widehat{F}_j} \widehat{p}_\ell(\widehat{s}) \alpha(T_K(\widehat{s})) \widehat{p}_k(\widehat{s}) d\widehat{s} \\ &\approx \frac{|F_j|}{|\widehat{F}_j|} \sum_{m=1}^{q'} \omega_{\widehat{F}_j,m} \widehat{p}_\ell(\xi_{\widehat{F}_j,m}) \alpha(T_K(\xi_{\widehat{F}_j,m})) \widehat{p}_k(\xi_{\widehat{F}_j,m}), \end{aligned}$$

wobei  $|F_j|$  bzw.  $|\widehat{F}_j|$  die  $(d-1)$ -dimensionalen Volumina der Facetten bezeichnen.



schwache Form	lokale Beiträge der Zelle $K$
$\int_{\Gamma} \varphi_j \alpha(s) \varphi_i ds$	$\frac{ F_j }{ \widehat{F}_j } \sum_{m=1}^{q'} \omega_{\widehat{F}_j,m} \widehat{p}_\ell(\xi_{\widehat{F}_j,m}) \alpha(T_K(s)) \widehat{p}_k(\xi_{\widehat{F}_j,m})$
$\int_{\Gamma} g(s) \varphi_i ds$	$\frac{ F_j }{ \widehat{F}_j } \sum_{m=1}^{q'} \omega_{\widehat{F}_j,m} g(T_K(s)) \widehat{p}_k(\xi_{\widehat{F}_j,m})$

Wie bereits in § 10 gesehen, führt die Assemblierung auf ein LGS<sup>108</sup>

$$A \vec{u} = \vec{F} \quad (13.8)$$

mit

$$\begin{cases} A = (a[\varphi_j, \varphi_i])_{i,j=1}^M, & \vec{F} = F(\varphi_i)|_{i=1}^M & \text{ohne Q-Fehler} \\ A = (a_h[\varphi_j, \varphi_i])_{i,j=1}^M, & \vec{F} = F_h(\varphi_i)|_{i=1}^M & \text{mit Q-Fehler.} \end{cases}$$

### § 13.2 Behandlung von Dirichlet-Randbedingungen

Wir betrachten nun Punkt (6) genauer. Zur Behandlung der Aufgabe mit gemischten RB

$$\begin{aligned} -\Delta u + c_0 u &= f & \text{in } \Omega \\ \frac{\partial}{\partial n} u + \alpha u &= g & \text{auf } \Gamma_N \\ u &= u_D & \text{auf } \Gamma_D = \Gamma \setminus \Gamma_N \end{aligned}$$

bestehen folgende Möglichkeiten:

<sup>107</sup>Man benötigt mehrere Q-Formeln pro Zelle, für jede Facette eine.

<sup>108</sup>Implementation für  $\mathbb{P}_1$ - und  $\mathbb{P}_2$ -Elemente mit verschiedenen Q-Formeln in der Übung

- (a) Approximation der Dirichlet-Randbedingungen durch Robin-Randbedingungen

$$\frac{\partial}{\partial n} u + \alpha u = \alpha u_D \quad \text{auf } \Gamma_D$$

mit einem großen  $\alpha > 0$ . Man spricht auch von **Stiff-Spring-** oder **Penalty-Randbedingungen**. Diese Bezeichnung ist dadurch motiviert, dass sich diese Gleichung im Fall einer symmetrischen Bilinearform ergibt, wenn man die Bedingung  $u = u_D$  nicht strikt fordert, sondern sie in der Energieminimierungsaufgabe (8.7) durch Anfügen des Terms

$$\frac{\alpha}{2} \int_{\Gamma_d} |u - u_D|^2 ds$$

quadratisch penalisiert.

Vorteile:

- Es treten nur natürliche Randbedingungen im Problem auf.

Nachteile:

- große Konditionszahl der Steifigkeitsmatrix für große  $\alpha$
- zusätzlicher Konsistenz- und damit Diskretisierungsfehler

- (b) Es sei  $A \vec{u} = \vec{F}$  das LGS, das man bei Assemblierung über *alle* globalen Freiheitsgrade erhält.<sup>109</sup> Es sei  $D \subset \{1, \dots, M\}$  die Indexmenge derjenigen globalen Freiheitsgrade, die „auf  $\Gamma_D$  liegen“ („**Dirichlet-Freiheitsgrade**“) und  $N = \{1, \dots, M\} \setminus D$ .

**Beachte:** Das System  $A \vec{u} = \vec{F}$  enthält falsche Gleichungen, denn als Raum der Testfunktionen ist nur  $H_{\Gamma_D}^1 = \{u \in H^1(\Omega) : u|_{\Gamma_D} = 0\}$  zugelassen bzw. in der diskreten Aufgabe  $H_{\Gamma_D}^1 \cap V_h$ , was durch  $\{\varphi_i\}_{i \in N}$  realisiert wird.

Es sei  $u_D$  hinreichend glatt, sodass alle  $\sigma_i$ ,  $i \in D$ , auf  $u_D$  anwendbar sind. Die bisherigen Gleichungen, die durch Testfunktionen  $\varphi_i$ ,  $i \in D$  erzeugt wurden, sollen durch

$$\underbrace{\sigma_i(u)}_{=u_i} = \sigma_i(u_D), \quad i \in D \quad (13.9)$$

ersetzt werden. Aus algorithmischer Sicht gibt es dazu mehrere äquivalente Möglichkeiten:<sup>110</sup>

- (i) Ersetzen der Zeilen  $i \in D$  von  $A$  durch den  $i$ -ten Einheitsvektor und der rechten Seite  $F_i$  durch  $\sigma_i(u_D)$ .

Nachteil: unsymmetrisches LGS selbst bei symmetrischer Bilinearform

- (ii) Zusätzlich Ersetzen der Spalten  $j \in D$  von  $A$  durch den  $j$ -ten Einheitsvektor und entsprechende Anpassung der rechten Seite.

Nachteil: Zeilen- und Spaltenmanipulation notwendig<sup>111</sup>

<sup>109</sup>Dazu muss man die Randdaten  $\alpha$  und  $g$  z. B. durch null auf  $\Gamma_D$  fortsetzen.

<sup>110</sup>Die vier Varianten werden in der Übung getestet.

<sup>111</sup>schwierig bei sparser Speichertechnik; jedoch muss man beim iterativen Lösen die Matrix nicht tatsächlich manipulieren, siehe Übung



(iii) Reduktion des LGS auf die Nicht-Dirichlet-Freiheitsgrade:

$$\sum_{j \in N} A_{ij} u_j = F_i - \sum_{j \in D} A_{ij} \sigma_j(u_D), \quad i \in N \quad (13.10)$$

und anschließendes Setzen von  $u_j = \sigma_j(u_D)$  für  $j \in D$ .

(iv) Aufnahme der Gleichungen (13.9) und zusätzlicher Variablen  $\vec{\lambda} \in \mathbb{R}^{|D|}$  ins LGS:

$$\begin{pmatrix} A & \chi_D^\top \\ \chi_D & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} \vec{F} \\ \sigma_i(u_D)|_{i \in D} \end{pmatrix} \quad (13.11)$$

Hierbei besteht  $\chi_D \in \{0, 1\}^{|D| \times M}$  aus denjenigen Zeilen der Einheitsmatrix, deren Indizes zu  $D$  gehören. Eine Aufgabe der Gestalt (13.11) bzw. allgemeiner

$$\begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} \vec{f} \\ \vec{g} \end{pmatrix} \quad (13.12)$$

heißt ein **Sattelpunktproblem**. Es ist symmetrisch, wenn  $A = A^\top$  ist.

**Beachte:** Die beim iterativen Lösen von (i)–(iv) benötigten Matrix-Vektor-Produkte können durch einfache Manipulationen der Matrix-Vektor-Produkte mit  $A$  erzeugt werden.<sup>112</sup>

**Frage:** Was motiviert (13.11) bzw. (13.12)?

### Bemerkung 13.2 (Zum Sattelpunktproblem)

- (a) Das System  $A\vec{u} = \vec{F}$  enthält falsche Gleichungen. Wir haben in (13.11) zunächst die fehlenden  $m = |D|$  Gleichungen  $\chi_D \vec{u} = \sigma_i(u_D)|_{i \in D}$  hinzugefügt und dann die falschen Gleichungen  $[A\vec{u} = \vec{F}]_{i \in D}$  durch Hinzunahme der Variablen  $\vec{\lambda}$  annulliert. Dies geschieht durch den Term  $\chi_D^\top \vec{\lambda}$ .
- (b) Falls  $A$  symmetrisch und positiv semidefinit ist, dann sind (13.12) gerade die notwendigen und hinreichenden Optimalitätsbedingungen der Aufgabe

$$\begin{aligned} &\text{Minimiere} \quad \frac{1}{2} \vec{u}^\top A \vec{u} - \vec{f}^\top \vec{u} \\ &\text{unter} \quad B \vec{u} = \vec{g}. \end{aligned}$$

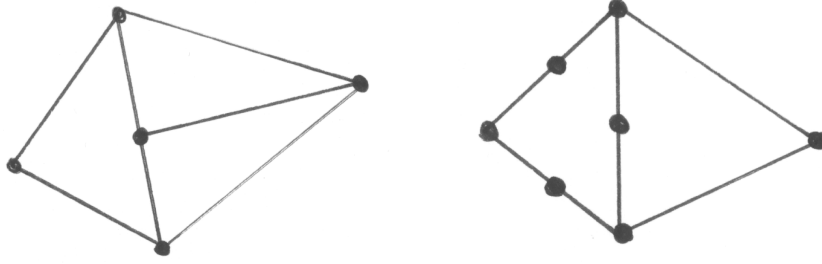
Dies entspricht einer Diskretisierung der Aufgabe (8.7) mit zusätzlichen Gleichungsnebenbedingungen.<sup>113</sup> Die Variablen  $\vec{\lambda}$  können als Lagrange-Multiplikatoren verstanden werden. Die lineare Unabhängigkeit von Zeilen von  $B$  entspricht der LICQ (Linear Independence Constraint Qualification).  $\diamond$

Neben Dirichlet-RB kann man auf dieselbe Art und Weise auch „hängende“ Freiheitsgrade behandeln, die etwa bei Gitterverfeinerung oder Verwendung ungleicher Polynomgrade<sup>114</sup> entstehen können.

<sup>112</sup>siehe Übung

<sup>113</sup>Suche in einem affinen Unterraum von  $V_h$

<sup>114</sup>bei sogenannten  $hp$ -Methoden



nicht-konformes Gitter (hängen-  
der Knoten)

keine affine Familie (ungleiche Po-  
lynomgrade)

Außerdem kann man die Sattelpunktidee auch benutzen, um Defekte aufgrund nicht-koerziver Bilinearformen  $a[\cdot, \cdot]$  auszugleichen, etwa beim Poisson-Problem mit reinen Neumannbedingungen (8.13), wie das folgende Lemma zeigt.

**Lemma 13.3** (Sattelpunktlemma<sup>115</sup>)

Es sei  $A \in \mathbb{R}^{n \times n}$ ,  $A = A^\top \succeq 0$  und  $B \in \mathbb{R}^{m \times n}$  mit  $m \leq n$ . Es sei

$$\mathcal{K} = \begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix}.$$

Dann gilt:

(a)

$$\mathcal{K} \text{ ist invertierbar} \Leftrightarrow \begin{cases} \text{rank}(B) = m \text{ (voller Zeilenrang) und} \\ \ker(A) \cap \ker(B) = \{0\}. \end{cases}$$

(b) Es sei  $\text{rank}(B) = m$ . Dann ist  $BB^\top$  symmetrisch positiv definit und  $\vec{u}_0 = B^\top(BB^\top)^{-1}\vec{g}$  eine Lösung von  $B\vec{u} = \vec{g}$ , und zwar die in  $\ker(B)^\perp$  eindeutige.

(c) Es seien die Voraussetzungen von (a) erfüllt. Es sei  $Z \in \mathbb{R}^{n \times (n-m)}$  eine Matrix, deren Spalten eine Basis von  $\ker(B)$  bilden.<sup>116</sup> Dann ist  $Z^\top AZ$  symmetrisch positiv definit. Weiter sei  $\vec{u}_0$  irgendeine Lösung von  $B\vec{u} = \vec{g}$ . Dann gilt:

$$\begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} \vec{f} \\ \vec{g} \end{pmatrix} \Leftrightarrow \begin{cases} Z^\top AZ \vec{v} = Z^\top (\vec{f} - A\vec{u}_0) \\ \vec{u} = \vec{u}_0 + Z\vec{v} \\ BB^\top \vec{\lambda} = B(\vec{f} - A\vec{u}) \end{cases} \quad (13.13)$$

<sup>115</sup>Wir beschränken uns hier der Einfachheit halber auf den Fall  $A = A^\top \succeq 0$  und verweisen auf [Benzi et al., 2005, Theorem 3.4] und [Gansterer et al., 2003, Theorem 3.1] für den allgemeinen Fall.

<sup>116</sup>Im Fall von Dirichlet-RB sind die Gleichungen  $B\vec{u} = \vec{g}$  von sehr einfacher Gestalt, sie stehen schon oben (13.11).  $B$  hat in jeder Zeile genau eine eins. Deshalb können wir  $Z$  (eine spaltenweise Basis des Nullraumes) direkt angeben:  $Z$  besteht aus den Einheitsvektoren, die gerade zu den Nicht-Dirichlet-Freiheitsgraden  $N$  gehören, somit  $\vec{v}$  aus den Koeffizienten  $(\vec{u}_j)_{j \in N}$ . Auch eine Partikulärlösung von  $B\vec{u} = \vec{g}$  lässt sich leicht angeben: Einfach die Koeffizienten  $u_D$  richtig setzen, die restlichen auf null. Damit besteht die Matrix im reduzierten System  $Z^\top AZ \vec{v} = Z^\top (\vec{f} - A\vec{u}_0)$  gerade aus den Zeilen und Spalten von  $A$ , die zu den  $N$ -Freiheitsgraden gehören, und wir bekommen dasselbe wie bei (13.10)!

*Beweis:* (a): „ $\Leftarrow$ “: Es seien die Voraussetzungen rechts erfüllt. Wir betrachten die Matrix

$$\mathcal{K}_- := \begin{pmatrix} A & B^\top \\ -B & 0 \end{pmatrix},$$

die genau dann invertierbar ist, wenn  $\mathcal{K}$  es ist. Es sei  $(\vec{u}, \vec{\lambda})$  ein Vektor im Kern von  $\mathcal{K}_-$ :

$$\begin{pmatrix} A & B^\top \\ -B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} \vec{0} \\ \vec{0} \end{pmatrix} \Rightarrow (\vec{u}^\top \quad \vec{\lambda}^\top) \begin{pmatrix} A & B^\top \\ -B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{\lambda} \end{pmatrix} = \vec{u}^\top A \vec{u} = 0.$$

Wegen  $\vec{x}^\top A \vec{x} \geq 0$  für alle  $\vec{x} \in \mathbb{R}^n$  muss  $\vec{u} \in \ker(A)$  liegen (benutze z. B. die Singulärwertzerlegung  $A = U\Sigma U^\top$  von  $A$ ). Wegen der zweiten Gleichung liegt außerdem  $\vec{u} \in \ker(B)$ , nach Voraussetzung ist also  $\vec{u} = \vec{0}$ . Daraus folgt  $B^\top \vec{\lambda} = \vec{0}$ , also  $\vec{\lambda} = \vec{0}$ , denn  $B^\top$  hat vollen Spaltenrang. Somit ist  $\ker(\mathcal{K}_-) = \{0\}$ , also sind  $\mathcal{K}_-$  und  $\mathcal{K}$  invertierbar.

„ $\Rightarrow$ “: Es sei eine der Voraussetzungen rechts *nicht* erfüllt. Falls  $B$  nicht vollen Zeilenrang hat, dann existiert  $\vec{\lambda} \neq \vec{0}$  mit  $B^\top \vec{\lambda} = \vec{0}$ , also auch

$$\begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{0} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} \vec{0} \\ \vec{0} \end{pmatrix},$$

d. h.,  $\mathcal{K}$  ist nicht invertierbar. Falls  $\ker(A) \cap \ker(B) \neq \{0\}$  ist, dann existiert  $\vec{u} \neq \vec{0}$  mit

$$\begin{pmatrix} A & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{u} \\ \vec{0} \end{pmatrix} = \begin{pmatrix} \vec{0} \\ \vec{0} \end{pmatrix},$$

d. h.,  $\mathcal{K}$  ist wiederum nicht invertierbar.

**Beachte:** Für „ $\Rightarrow$ “ wurde  $A = A^\top \succeq 0$  nicht gebraucht.

(b):  $BB^\top \succeq 0$  ist klar. Es sei  $\vec{w}$  ein Vektor im Kern von  $BB^\top$ . Es folgt  $0 = \vec{w}^\top BB^\top \vec{w} = \|B^\top \vec{w}\|^2$  und damit  $\vec{w} = \vec{0}$ , denn  $B^\top$  hat vollen Spaltenrang. Also hat  $BB^\top$  nur positive Eigenwerte.

Jeder Vektor  $\vec{u} \in \mathbb{R}^n$  kann eindeutig zerlegt werden in

$$\vec{u} = \underbrace{\vec{u}_1}_{\in \ker(B)} + \overbrace{\vec{u}_2}^{\in \ker(B)^\perp = \text{range}(B^\top)} = \vec{u}_1 + B^\top \vec{w}.$$

Mit dieser Zerlegung gilt

$$B\vec{u} = \vec{g} \Leftrightarrow \overbrace{B\vec{u}_1}^{=0} + B\vec{u}_2 = \vec{g} \Leftrightarrow BB^\top \vec{w} = \vec{g}.$$

Also ist  $\vec{u}_0 = B^\top \vec{w} = B^\top (BB^\top)^{-1} \vec{g}$  die in  $\ker(B)^\perp$  eindeutige Lösung von  $B\vec{u} = \vec{g}$ . Die allgemeine Lösung ist  $\vec{u}_0 + \ker(B)$ .

(c):  $Z^\top AZ \succeq 0$  ist klar. Es sei  $\vec{v}$  beliebig und betrachte  $\vec{v}^\top Z^\top AZ \vec{v}$ . Wie bei (a) „ $\Leftarrow$ “ folgt  $Z\vec{v} \in \ker(A)$ . Jedoch liegt  $Z\vec{v}$  auch in  $\ker(B)$ , d. h.,  $Z\vec{v} = \vec{0}$  und damit  $\vec{v} = \vec{0}$ , denn die Spalten von  $Z$  sind linear unabhängig. Also hat  $Z^\top AZ$  nur positive Eigenwerte.

„ $\Rightarrow$ “: Nach (b) haben die Lösungen von  $B\vec{u} = \vec{g}$  die Darstellung  $\vec{u} = \vec{u}_0 + \underbrace{Z\vec{v}}_{\in \ker(B)}$  mit  $\vec{v} \in \mathbb{R}^{n-m}$  beliebig. Einsetzen in die erste Gleichung ergibt

$$\begin{aligned} A\vec{u} + B^\top \vec{\lambda} &= \vec{f} \\ \Rightarrow AZ\vec{v} + B^\top \vec{\lambda} &= \vec{f} - A\vec{u}_0 \\ \Rightarrow Z^\top AZ\vec{v} &= B^\top (\vec{f} - A\vec{u}_0) \end{aligned}$$

Aus der ersten Gleichung folgt jetzt noch  $B^\top \vec{\lambda} = \vec{f} - A\vec{u}$ , also  $BB^\top \vec{\lambda} = B(\vec{f} - A\vec{u})$ .

„ $\Leftarrow$ “: Beide Systeme sind eindeutig lösbar. Wir haben bereits gezeigt, dass die Lösung des linken System in (13.13) notwendig auch das rechte löst. Wegen der Eindeutigkeit müssen beide Lösungen gleich sein.  $\square$

### § 13.3 Speichertechnik

Aufgrund der lokalen Träger der globalen Basisfunktionen  $\{\varphi\}_{i=1}^M$  ist die Steifigkeitsmatrix  $A$  dünn besetzt (sparse<sup>117</sup>). Zur Speicherung solcher Matrizen stehen verschiedene Formate zur Verfügung, die wir am Beispiel

$$A = \begin{pmatrix} 1 & \cdot & \cdot & 2 & \cdot \\ 3 & 4 & \cdot & 5 & \cdot \\ 6 & \cdot & 7 & 8 & 9 \\ \cdot & \cdot & 10 & 11 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 12 \end{pmatrix}$$

erläutern. Im Wesentlichen werden die Nicht-Null-Einträge der Matrix und ihre Positionen gespeichert.

**Compressed Sparse Column (CSC):**

Die Nicht-Null-Einträge stehen spaltenweise in einem Vektor  $a$ , der Vektor  $i$  enthält die dazugehörigen Zeilenindizes und der Vektor  $j$  die Indizes in  $a$ , an denen eine neue Spalte beginnt.

$a$	<span style="border: 1px solid black; padding: 0 2px;">1</span>	3	6	<span style="border: 1px solid black; padding: 0 2px;">4</span>	<span style="border: 1px solid black; padding: 0 2px;">7</span>	10	<span style="border: 1px solid black; padding: 0 2px;">2</span>	5	8	11	<span style="border: 1px solid black; padding: 0 2px;">9</span>	12	Einträge
$i$	1	2	3	2	3	4	1	2	3	4	3	5	Zeilen
$j$	1	4	5	7	11	13							Index in $a$

**Compressed Sparse Row (CSR):**

Dieses Format ist analog zu CSC, jedoch zeilenorientiert.

$a$	<span style="border: 1px solid black; padding: 0 2px;">1</span>	2	<span style="border: 1px solid black; padding: 0 2px;">3</span>	4	5	<span style="border: 1px solid black; padding: 0 2px;">6</span>	7	8	9	<span style="border: 1px solid black; padding: 0 2px;">10</span>	11	<span style="border: 1px solid black; padding: 0 2px;">12</span>	Einträge
$j$	1	4	1	2	4	1	3	4	5	3	4	5	Spalten
$i$	1	3	6	10	12	13							Index in $a$

**Modified Sparse Row (MSR):**

Beim MSR stehen zunächst die Diagonaleinträge der Matrix (i. d. R. keine Nullen) in  $a$  und dann zeilenweise die restlichen Nicht-Null-Einträge. Die ersten Indizes in  $i$

<sup>117</sup>Die MATLAB-Befehle für dünn besetzte Matrizen erhält man mit `doc sparsfun`.

geben an, an welcher Stelle in  $a$  eine neue Zeile beginnt. Die späteren Indizes sind die Spaltenindizes der zugehörigen Einträge in  $a$  wie bei CSR.

$a$	1	4	7	11	12	*	<span style="border: 1px solid black;">2</span>	<span style="border: 1px solid black;">3</span>	5	<span style="border: 1px solid black;">6</span>	8	9	<span style="border: 1px solid black;">10</span>	Einträge
$i$	7	8	10	13	14	14	4	1	4	1	4	5	3	Index in $a$