

FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT • DEPARTMENT INFORMATIK

Lehrstuhl für Informatik 10 (Systemsimulation)



Akkurate Mehrgitteroperatoren für die Strukturmechanik

Niklas Schmitt

Bachelorarbeit

Akkurate Mehrgitteroperatoren für die Strukturmechanik

Niklas Schmitt

Bachelorarbeit

Aufgabensteller: Prof. Dr. C. Pflaum

Betreuer: Prof. Dr. C. Pflaum

Bearbeitungszeitraum: 2.11.2022 – 3.4.2023

Erklärung:

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Der Universität Erlangen-Nürnberg, vertreten durch den Lehrstuhl für Systemsimulation (Informatik 10), wird für Zwecke der Forschung und Lehre ein einfaches, kostenloses, zeitlich und örtlich unbeschränktes Nutzungsrecht an den Arbeitsergebnissen der Bachelorarbeit einschließlich etwaiger Schutzrechte und Urheberrechte eingeräumt.

Erlangen, den 30. März 2023



.....

Abstract

As the default multigrid operators prolongation and restriction fail to prolongate and restrict all the functions for the rigid body modes, the multigrid method converges rather slowly for the finite element method which is commonly used in structural mechanics.

Thus the goal of this thesis is to ameliorate the rate of convergence of the multigrid method for the finite element method in structural mechanics by adjusting its operators. For that purpose the points on the coarse grid get assigned weights, that are calculated relative to individual points on the fine grid by weighted interpolation which specific procedure slightly differs for one-, two-, and three-dimensional figures respectively. To make sure the weights can all be calculated unambiguously, the coordinates of the grid points get transformed into local coordinates.

After that, the new operators are integrated in code and tested for poisson's equation on different geometric figures. With the now calculated weights the rate of convergence could be slightly improved for poisson's equation, but should be even better for applications in structural mechanics.

Inhaltsverzeichnis

1	Einleitung	6
2	Ansatz der gewichteten Interpolation	7
2.1	Eindimensionale gewichtete Interpolation	7
2.2	Zweidimensionale gewichtete Interpolation	8
2.3	Dreidimensionale gewichtete Interpolation	9
3	Eigenschaften der Berechnung der Gewichte	11
4	Koordinatentransformation in lokale Koordinaten	13
4.1	Lokale Koordinaten im Zweidimensionalen	13
4.2	Lokale Koordinaten im Dreidimensionalen	14
4.3	Berechnung lokaler Koordinaten für vier Punkte in einer Ebene im \mathbb{R}^3	15
5	Beispielrechnung für vier Gewichte im \mathbb{R}^2	17
6	Integration in das Mehrgitterverfahren im Programm UGBlocks	18
7	Test der Konvergenzgeschwindigkeit anhand der Poisson-Gleichung	20
8	Zusammenfassung und Ausblick	23
9	Literaturverzeichnis	24

1 Einleitung

In der Strukturmechanik ist die Finite-Elemente-Methode eine grundlegende Vorgehensweise, Fragestellungen wie die Verformung von Festkörpern zu bearbeiten und zu lösen [7]. Durch die Diskretisierung der im Zuge der Finite-Elemente-Methode entstehenden partiellen Differentialgleichungen ergeben sich Gleichungssysteme mit sehr vielen Unbekannten, so dass herkömmliche Verfahren wie die Gaußelimination nicht mehr effizient durchführbar sind. Ein Verfahren zur Lösung dieser großen Gleichungssysteme ist das Mehrgitterverfahren. Der Fehler einer Näherungslösung wird iterativ auf immer gröbere Gitter restringiert, bis er auf dem größten Gitter direkt gelöst wird. Anschließend wird er sukzessive auf die feinen Gitter prolongiert, und schließlich damit die ursprüngliche Näherungslösung verbessert [3].

Die Operatoren des Mehrgitterverfahrens - Restriktion und Prolongation - sind dabei allgemein definiert. Das Problem an dieser allgemeinen Definition besteht darin, dass bei in der Strukturmechanik entstehenden Problemen, beispielsweise die Ausdehnung eines Körpers, nicht alle Funktionen der Rigid Body Modes korrekt prolongiert bzw. restringiert werden.

Beispielsweise wird ein Punkt auf einer Fläche immer mit den selben Gewichten für die Stützpunkte interpoliert/prolongiert. Allerdings können sich die Verhältnisse der Distanzen der Stützpunkte zum Interpolationspunkt verändern, zum Beispiel bei Ausdehnung des Körpers. Daher bleiben die x-, y- und z-Koordinaten nicht erhalten, wodurch die Funktionen

$$\begin{pmatrix} y \\ -x \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ z \\ -y \end{pmatrix} \text{ und } \begin{pmatrix} z \\ 0 \\ -x \end{pmatrix} \text{ ([4], S. 43)}$$

nicht korrekt prolongiert werden. Deshalb konvergiert das herkömmliche Mehrgitterverfahren für strukturmechanische Probleme nur langsam.

Allgemein fließen Stützpunkte im standardmäßigen Mehrgitterverfahren unabhängig von ihrer Lage bei der Interpolation mit festen Gewichten ein. Diese Arbeit stellt eine Anpassung des Prolongations- und Restriktionsoperators vor, mit welchen alle Funktionen korrekt prolongiert und restringiert werden sollen. Dafür erhält jeder Punkt auf dem groben Gitter ein individuell berechnetes Gewicht bezüglich des Interpolationspunktes auf dem feinen Gitter, mit welchem er in die Prolongation und Restriktion eingeht. Ziel ist es, damit eine Verbesserung der Konvergenzgeschwindigkeit des Mehrgitterverfahrens in der Strukturmechanik zu erreichen.

Das vorgestellte Verfahren wird anschließend in das Programm UGBlocks [5] integriert und die Konvergenzgeschwindigkeit für das Poisson-Problem getestet und der Geschwindigkeit der vorherigen, standardmäßigen Methode gegenübergestellt.

2 Ansatz der gewichteten Interpolation

Das Problem bei den herkömmlichen Mehrgitteroperatoren ist, dass alle Punkte **unabhängig** von ihrer Lage im Raum mit festen Gewichten in Prolongation und Restriktion eingehen. Es wird also ein Ansatz gewählt, bei dem jedem Punkt auf dem groben Gitter ein Gewicht ω zugewiesen wird. Diese Gewichte werden dann **abhängig** von ihrer Lage im Raum mittels gewichteter Interpolation berechnet.

Für einen Interpolationspunkt, welcher auf dem feinen Gitter liegt, gibt es jeweils zwei, vier oder acht Stützpunkte und damit zwei, vier oder acht zu berechnende Gewichte, abhängig davon, ob es sich um eine Kante, Fläche oder ein Volumen handelt.

Eine allgemeine Forderung ist, dass die Summe der Gewichte je Interpolationspunkt immer eins ergeben. Des Weiteren sollen die Koordinaten erhalten bleiben. Folgende Bedingungen werden also an die Gewichte gestellt:

$$(i) \sum_{i \in M} \omega_i = 1 \text{ und}$$

$$(ii) \sum_{i \in M} \omega_i \cdot x_i = x_Q, \text{ (iii) } \sum_{i \in M} \omega_i \cdot y_i = y_Q \text{ und (iv) } \sum_{i \in M} \omega_i \cdot z_i = z_Q,$$

wobei $P_i = (x_i, y_i, z_i)$ die Stützpunkte auf dem groben Gitter und $Q = (x_Q, y_Q, z_Q)$ den zu interpolierenden Punkt auf dem feinen Gitter bezeichnen.

Des Weiteren sei $M = \{W, E\}$, $M = \{SW, SE, NW, NE\}$ oder $M = \{SWD, SED, NWD, NED, SWT, SET, NWT, NET\}$, je nachdem, ob die Interpolation im \mathbb{R}^1 , \mathbb{R}^2 oder \mathbb{R}^3 durchzuführen ist. Für den eindimensionalen Fall sind dabei die Bedingungen (iii) und (iv) und für den zweidimensionalen Bedingung (iv) nicht zu erfüllen.

Ein Gewicht für den Stützpunkt ist größer bzw. kleiner, je näher bzw. ferner der Interpolationspunkt diesem ist, anstatt einen festen Wert zu besitzen.

Die Berechnung der Gewichte unterscheidet sich je nachdem, ob es sich um eine Kante, Fläche oder ein Volumen handelt. Im Folgenden werden diese Fälle unterschieden und das Vorgehen dargestellt, wie die Berechnung durchgeführt wird.

2.1 Eindimensionale gewichtete Interpolation

Für die Interpolation auf den Kanten der Gitter des Mehrgitterverfahrens kommt eine eindimensionale gewichtete Interpolation zum Einsatz.

Dafür seien zwei beliebige, aber verschiedene, grobe Gitterpunkte x_W und x_E gegeben, welchen die Gewichte ω_W bzw. ω_E zugeordnet werden, und der Interpolationspunkt x_Q auf dem feinen Gitter, der zwischen x_W und x_E liegt, also $x_W \leq x_Q \leq x_E$, $x_W \neq x_E$. Um ω_W und ω_E eindeutig berechnen zu können, benötigt man zwei Gleichungen, welche unmittelbar aus den zuvor genannten Bedingungen folgen:

$$\omega_W + \omega_E = 1 \tag{1}$$

$$x_W \cdot \omega_W + x_E \cdot \omega_E = x_Q \tag{2}$$

Als lineares Gleichungssystem (LGS) $A\omega = b$ ergeben die Gleichungen (1) und (2):

$$\begin{pmatrix} 1 & 1 \\ x_W & x_E \end{pmatrix} \begin{pmatrix} \omega_W \\ \omega_E \end{pmatrix} = \begin{pmatrix} 1 \\ x_Q \end{pmatrix}. \tag{3}$$

Satz 1 Ist $x_W \neq x_E$, so ist dieses System eindeutig lösbar.

Beweis. Das System (3) ist eindeutig lösbar, wenn $\det(A) \neq 0$ gilt ([6], S. 168).

$$\det(A) = \det \begin{pmatrix} 1 & 1 \\ x_W & x_E \end{pmatrix} = 1 \cdot x_E - x_W \cdot 1 \neq 0.$$

□

Die Gewichte können also eindeutig bestimmt werden. Das ist essenziell, da jeder Stützpunkt ein eindeutiges Gewicht benötigt, um anschließend Restriktion und Prolongation durchführen zu können.

2.2 Zweidimensionale gewichtete Interpolation

Im Zweidimensionalen wird eine Interpolation auf Flächen durchgeführt. Es sind also vier beliebige Punkte P_{SW}, P_{SE}, P_{NW} und P_{NE} mit den Koordinaten $P_i = (x_i, y_i), i \in M$ auf dem groben Gitter gegeben mit $M = \{SW, SE, NW, NE\}$. Zudem sei der Interpolationspunkt $Q = (x_Q, y_Q)$ auf der von den P_i aufgespannten Fläche auf dem feinen Gitter. Dabei gilt $P_i \neq P_j, \forall i, j \in M, i \neq j$.

Folgende drei Gleichungen ergeben sich direkt aus den zuvor genannten Bedingungen:

$$\omega_{SW} + \omega_{SE} + \omega_{NW} + \omega_{NE} = 1 \quad (4)$$

$$\omega_{SW}x_{SW} + \omega_{SE}x_{SE} + \omega_{NW}x_{NW} + \omega_{NE}x_{NE} = x_Q \quad (5)$$

$$\omega_{SW}y_{SW} + \omega_{SE}y_{SE} + \omega_{NW}y_{NW} + \omega_{NE}y_{NE} = y_Q. \quad (6)$$

Allerdings ergibt sich nun das Problem, dass vier Gewichte mit nur drei Gleichungen eindeutig zu bestimmen sind. Es ist also notwendig, eine weitere Bedingung aufzustellen, aus der eine zusätzliche Gleichung folgt.

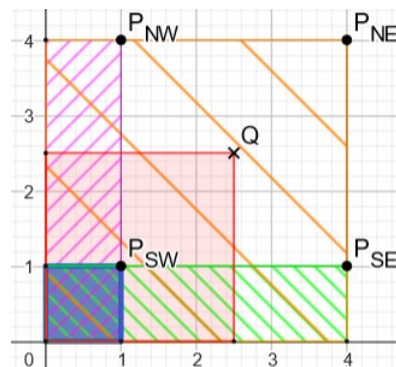


Abbildung 1: Die Summe der gewichteten Flächen ist die Fläche zwischen Ursprung und Q

Diese ist die Erhaltung der Flächen $A_i = x_i y_i$, also

$$\omega_{SW}A_{SW} + \omega_{SE}A_{SE} + \omega_{NW}A_{NW} + \omega_{NE}A_{NE} =$$

$$\omega_{SW}x_{SW}y_{SW} + \omega_{SE}x_{SE}y_{SE} + \omega_{NW}x_{NW}y_{NW} + \omega_{NE}x_{NE}y_{NE} = x_Q y_Q. \quad (7)$$

Das sind nun vier Gleichungen für vier Unbekannte. Damit erhält man als Berechnungsvorschrift für die Gewichte, wenn man die Gleichungen (4)-(7) als $A\omega = b$ schreibt:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ x_{SW} & x_{SE} & x_{NW} & x_{NE} \\ y_{SW} & y_{SE} & y_{NW} & y_{NE} \\ x_{SW}y_{SW} & x_{SE}y_{SE} & x_{NW}y_{NW} & x_{NE}y_{NE} \end{pmatrix} \begin{pmatrix} \omega_{SW} \\ \omega_{SE} \\ \omega_{NW} \\ \omega_{NE} \end{pmatrix} = \begin{pmatrix} 1 \\ x_Q \\ y_Q \\ x_Q y_Q \end{pmatrix}. \quad (8)$$

Auch hier gilt die Frage nach der Lösbarkeit des Gleichungssystems (8). Diese ist nicht allgemein gegeben, was später in Kapitel 3 gezeigt wird.

2.3 Dreidimensionale gewichtete Interpolation

Es bleibt noch die Interpolation im Dreidimensionalen, wofür eine erneute Anpassung notwendig ist. Es befinden sich acht beliebige Punkte $P_i = (x_i, y_i, z_i)$, $i \in M$, $M = \{SWD, SED, NWD, SWT, NED, SET, NWT, NET\}$ auf dem groben und $Q = (x_Q, y_Q, z_Q)$ im Inneren des Hexaeders, der von den P_i aufgespannt wird, auf dem feinen Gitter. Dabei gilt für die P_i : $P_i \neq P_j$ $\forall i, j \in M, i \neq j$.

Ziel ist es, acht Gewichte ω_i zu berechnen, eines für jeden Punkt P_i . Wieder müssen die Gewichte in Summe eins ergeben.

$$\omega_{SWD} + \omega_{SED} + \omega_{NWD} + \omega_{SWT} + \omega_{NED} + \omega_{SET} + \omega_{NWT} + \omega_{NET} = 1 \quad (9)$$

Des Weiteren sollen wie zuvor die Koordinaten erhalten bleiben:

$$\begin{aligned} \omega_{SWD}x_{SWD} + \omega_{SED}x_{SED} + \omega_{NWD}x_{NWD} + \omega_{SWT}x_{SWT} \\ + \omega_{NED}x_{NED} + \omega_{SET}x_{SET} + \omega_{NWT}x_{NWT} + \omega_{NET}x_{NET} = x_Q \end{aligned} \quad (10)$$

$$\begin{aligned} \omega_{SWD}y_{SWD} + \omega_{SED}y_{SED} + \omega_{NWD}y_{NWD} + \omega_{SWT}y_{SWT} \\ + \omega_{NED}y_{NED} + \omega_{SET}y_{SET} + \omega_{NWT}y_{NWT} + \omega_{NET}y_{NET} = y_Q \end{aligned} \quad (11)$$

$$\begin{aligned} \omega_{SWD}z_{SWD} + \omega_{SED}z_{SED} + \omega_{NWD}z_{NWD} + \omega_{SWT}z_{SWT} \\ + \omega_{NED}z_{NED} + \omega_{SET}z_{SET} + \omega_{NWT}z_{NWT} + \omega_{NET}z_{NET} = z_Q \end{aligned} \quad (12)$$

Analog zum zweidimensionalen Fall sind das zu wenige Gleichungen. Hier sind es sogar vier Gleichungen zu wenig - vier Gleichungen für acht gesuchte Gewichte. Es werden also vier weitere benötigt, um die ω_i eindeutig berechnen zu können. Daher stellt man auch hier zusätzliche Bedingungen auf.

Erneut kann man auf die Flächen zurückgreifen, wobei es in diesem Fall Flächen in drei Richtungen gibt, jeweils eine in xy-, xz- und yz-Richtung, sodass sich folgende drei Gleichungen ergeben:

$$\begin{aligned} \omega_{SWD}x_{SWD}y_{SWD} + \omega_{SED}x_{SED}y_{SED} + \omega_{NWD}x_{NWD}y_{NWD} + \omega_{SWT}x_{SWT}y_{SWT} \\ + \omega_{NED}x_{NED}y_{NED} + \omega_{SET}x_{SET}y_{SET} + \omega_{NWT}x_{NWT}y_{NWT} + \omega_{NET}x_{NET}y_{NET} = x_Q y_Q \end{aligned} \quad (13)$$

$$\begin{aligned} \omega_{SWD}x_{SWD}z_{SWD} + \omega_{SED}x_{SED}z_{SED} + \omega_{NWD}x_{NWD}z_{NWD} + \omega_{SWT}x_{SWT}z_{SWT} \\ + \omega_{NED}x_{NED}z_{NED} + \omega_{SET}x_{SET}z_{SET} + \omega_{NWT}x_{NWT}z_{NWT} + \omega_{NET}x_{NET}z_{NET} = x_Q z_Q \end{aligned} \quad (14)$$

$$\begin{aligned} \omega_{SWD}y_{SWD}z_{SWD} + \omega_{SED}y_{SED}z_{SED} + \omega_{NWD}y_{NWD}z_{NWD} + \omega_{SWT}y_{SWT}z_{SWT} \\ + \omega_{NED}y_{NED}z_{NED} + \omega_{SET}y_{SET}z_{SET} + \omega_{NWT}y_{NWT}z_{NWT} + \omega_{NET}y_{NET}z_{NET} = y_Q z_Q \end{aligned} \quad (15)$$

Mit (9)-(15) sind es sieben Gleichungen. Als achte wird nun noch die Erhaltung des Volumens $x_i \cdot y_i \cdot z_i$ vorausgesetzt:

$$\begin{aligned} \omega_{SWD}x_{SWD}y_{SWD}z_{SWD} + \omega_{SED}x_{SED}y_{SED}z_{SED} + \omega_{NWD}x_{NWD}y_{NWD}z_{NWD} + \omega_{SWT}x_{SWT}y_{SWT}z_{SWT} \\ + \omega_{NED}x_{NED}y_{NED}z_{NED} + \omega_{SET}x_{SET}y_{SET}z_{SET} + \omega_{NWT}x_{NWT}y_{NWT}z_{NWT} + \omega_{NET}x_{NET}y_{NET}z_{NET} \\ = x_Q y_Q z_Q \end{aligned} \quad (16)$$

Es ergibt sich das Gleichungssystem

$$A\omega = b \quad (17)$$

mit

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_{SWD} & x_{SED} & x_{NWD} & x_{SWT} & x_{NED} & x_{SET} & x_{NWT} & x_{NET} \\ y_{SWD} & y_{SED} & y_{NWD} & y_{SWT} & y_{NED} & y_{SET} & y_{NWT} & y_{NET} \\ z_{SWD} & z_{SED} & z_{NWD} & z_{SWT} & z_{NED} & z_{SET} & z_{NWT} & z_{NET} \\ x_{SWD}y_{SWD} & x_{SED}y_{SED} & x_{NWD}y_{NWD} & x_{SWT}y_{SWT} & x_{NED}y_{NED} & x_{SET}y_{SET} & x_{NWT}y_{NWT} & x_{NET}y_{NET} \\ x_{SWD}z_{SWD} & x_{SED}z_{SED} & x_{NWD}z_{NWD} & x_{SWT}z_{SWT} & x_{NED}z_{NED} & x_{SET}z_{SET} & x_{NWT}z_{NWT} & x_{NET}z_{NET} \\ y_{SWD}z_{SWD} & y_{SED}z_{SED} & y_{NWD}z_{NWD} & y_{SWT}z_{SWT} & y_{NED}z_{NED} & y_{SET}z_{SET} & y_{NWT}z_{NWT} & y_{NET}z_{NET} \\ (xyz)_{SWD} & (xyz)_{SED} & (xyz)_{NWD} & (xyz)_{SWT} & (xyz)_{NED} & (xyz)_{SET} & (xyz)_{NWT} & (xyz)_{NET} \end{pmatrix},$$

$$\omega = \begin{pmatrix} \omega_{SWD} \\ \omega_{SED} \\ \omega_{NWD} \\ \omega_{SWT} \\ \omega_{NED} \\ \omega_{SET} \\ \omega_{NWT} \\ \omega_{NET} \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} 1 \\ x_Q \\ y_Q \\ z_Q \\ x_Q y_Q \\ x_Q z_Q \\ y_Q z_Q \\ x_Q y_Q z_Q \end{pmatrix},$$

wobei $(xyz)_i$ jeweils $x_i y_i z_i$ entspricht und hier nur für die Übersichtlichkeit verwendet wird. Dieses Gleichungssystem ist nicht allgemein lösbar, wie das folgende Kapitel zeigt.

3 Eigenschaften der Berechnung der Gewichte

Es folgen zwei wichtige Aussagen über die zuvor beschriebene gewichtete Interpolation als Berechnungsvorschrift der Gewichte.

Satz 2 Die Berechnung der Gewichte entsprechend der zuvor genannten Vorgehensweise ist translationsinvariant.

Es folgt ein exemplarischer Beweis für den zweidimensionalen Fall, die Beweise für die ein- und dreidimensionalen Varianten verhalten sich analog.

Beweis. Sei $\omega^* = (\omega_{SW}^*, \omega_{SE}^*, \omega_{NW}^*, \omega_{NE}^*)$ die Lösung des linearen Gleichungssystems

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ x_{SW} & x_{SE} & x_{NW} & x_{NE} \\ y_{SW} & y_{SE} & y_{NW} & y_{NE} \\ x_{SW}y_{SW} & x_{SE}y_{SE} & x_{NW}y_{NW} & x_{NE}y_{NE} \end{pmatrix} \begin{pmatrix} \omega_{SW} \\ \omega_{SE} \\ \omega_{NW} \\ \omega_{NE} \end{pmatrix} = \begin{pmatrix} 1 \\ x_Q \\ y_Q \\ x_Q y_Q \end{pmatrix}$$

zu den vier beliebigen, unterschiedlichen Stützpunkten $P_i = (x_i, y_i)$. Dann ist ω^* auch Lösung des durch Translation um $\vec{\delta} = (\delta_x, \delta_y)$ entstehenden Gleichungssystems

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ x_{SW} + \delta_x & x_{SE} + \delta_x & x_{NW} + \delta_x & x_{NE} + \delta_x \\ y_{SW} + \delta_y & y_{SE} + \delta_y & y_{NW} + \delta_y & y_{NE} + \delta_y \\ [(x_{SW} + \delta_x) \cdot (y_{SW} + \delta_y)] & [(x_{SE} + \delta_x) \cdot (y_{SE} + \delta_y)] & [(x_{NW} + \delta_x) \cdot (y_{NW} + \delta_y)] & [(x_{NE} + \delta_x) \cdot (y_{NE} + \delta_y)] \end{pmatrix} \begin{pmatrix} \omega_{SW} \\ \omega_{SE} \\ \omega_{NW} \\ \omega_{NE} \end{pmatrix} = \begin{pmatrix} 1 \\ x_Q + \delta_x \\ y_Q + \delta_y \\ [(x_Q + \delta_x) \cdot (y_Q + \delta_y)] \end{pmatrix}$$

da

$$\begin{aligned} & (x_{SW} + \delta_x) \cdot \omega_{SW}^* + (x_{SE} + \delta_x) \cdot \omega_{SE}^* + (x_{NW} + \delta_x) \cdot \omega_{NW}^* + (x_{NE} + \delta_x) \cdot \omega_{NE}^* \\ &= x_{SW}\omega_{SW}^* + x_{SE}\omega_{SE}^* + x_{NW}\omega_{NW}^* + x_{NE}\omega_{NE}^* + \delta_x\omega_{SW}^* + \delta_x\omega_{SE}^* + \delta_x\omega_{NW}^* + \delta_x\omega_{NE}^* \\ &= x_Q + \delta_x \cdot (\omega_{SW}^* + \omega_{SE}^* + \omega_{NW}^* + \omega_{NE}^*) = x_Q + \delta_x \end{aligned}$$

und

$$\begin{aligned} & (y_{SW} + \delta_y) \cdot \omega_{SW}^* + (y_{SE} + \delta_y) \cdot \omega_{SE}^* + (y_{NW} + \delta_y) \cdot \omega_{NW}^* + (y_{NE} + \delta_y) \cdot \omega_{NE}^* \\ &= y_{SW}\omega_{SW}^* + y_{SE}\omega_{SE}^* + y_{NW}\omega_{NW}^* + y_{NE}\omega_{NE}^* + \delta_y\omega_{SW}^* + \delta_y\omega_{SE}^* + \delta_y\omega_{NW}^* + \delta_y\omega_{NE}^* \\ &= y_Q + \delta_y \cdot (\omega_{SW}^* + \omega_{SE}^* + \omega_{NW}^* + \omega_{NE}^*) = y_Q + \delta_y \end{aligned}$$

und

$$\begin{aligned} & (x_{SW} + \delta_x)(y_{SW} + \delta_y) \cdot \omega_{SW}^* + (x_{SE} + \delta_x)(y_{SE} + \delta_y) \cdot \omega_{SE}^* \\ &+ (x_{NW} + \delta_x)(y_{NW} + \delta_y) \cdot \omega_{NW}^* + (x_{NE} + \delta_x)(y_{NE} + \delta_y) \cdot \omega_{NE}^* \\ &= x_{SW}y_{SW}\omega_{SW}^* + x_{SE}y_{SE}\omega_{SE}^* + x_{NW}y_{NW}\omega_{NW}^* + x_{NE}y_{NE}\omega_{NE}^* \\ &+ x_{SW}\delta_y\omega_{SW}^* + x_{SE}\delta_y\omega_{SE}^* + x_{NW}\delta_y\omega_{NW}^* + x_{NE}\delta_y\omega_{NE}^* \\ &+ y_{SW}\delta_x\omega_{SW}^* + y_{SE}\delta_x\omega_{SE}^* + y_{NW}\delta_x\omega_{NW}^* + y_{NE}\delta_x\omega_{NE}^* \\ &+ \delta_x\delta_y\omega_{SW}^* + \delta_x\delta_y\omega_{SE}^* + \delta_x\delta_y\omega_{NW}^* + \delta_x\delta_y\omega_{NE}^* \\ &= x_Q y_Q + \delta_y \cdot (x_{SW}\omega_{SW}^* + x_{SE}\omega_{SE}^* + x_{NW}\omega_{NW}^* + x_{NE}\omega_{NE}^*) \\ &+ \delta_x \cdot (y_{SW}\omega_{SW}^* + y_{SE}\omega_{SE}^* + y_{NW}\omega_{NW}^* + y_{NE}\omega_{NE}^*) \\ &+ \delta_x\delta_y \cdot (\omega_{SW}^* + \omega_{SE}^* + \omega_{NW}^* + \omega_{NE}^*) \\ &= x_Q y_Q + \delta_y x_Q + \delta_x y_Q + \delta_x \delta_y = (x_Q + \delta_x)(y_Q + \delta_y). \end{aligned}$$

Damit erfüllt ω^* alle vier Gleichungen des durch Translation entstandenen Gleichungssystems und ist somit auch dessen eindeutige Lösung. \square

Satz 3 Die Berechnung der Gewichte nach beschriebener Methode ist **nicht** rotationsinvariant.

Zum Beweis von Satz 3 folgt nun ein Gegenbeispiel.

Seien die vier Stützpunkte $P_i = (x_i, y_i)$ als die Ecken des Einheitsquadrats gegeben, also $P_{SW} = (0, 0)$, $P_{SE} = (1, 0)$, $P_{NW} = (0, 1)$ und $P_{NE} = (1, 1)$. Das Gleichungssystem $A\omega = b$ ist dann:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \omega_{SW} \\ \omega_{SE} \\ \omega_{NW} \\ \omega_{NE} \end{pmatrix} = \begin{pmatrix} 1 \\ x_Q \\ y_Q \\ x_Q y_Q \end{pmatrix},$$

welches lösbar ist, da $\det(A) = 1 \neq 0$ ([6], S. 168).

Dreht man nun das Quadrat $P_{SW}P_{SE}P_{NW}P_{NE}$ um dessen Mittelpunkt mit der Rotationsmatrix $R = \begin{pmatrix} \cos(\frac{\pi}{4}) & -\sin(\frac{\pi}{4}) \\ \sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \end{pmatrix}$ um 45° gegen den Uhrzeigersinn, erhält man für A' :

$$A' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \frac{1}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1}{2} \\ \frac{1-\sqrt{2}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1+\sqrt{2}}{2} \\ \frac{1-\sqrt{2}}{4} & \frac{1+\sqrt{2}}{4} & \frac{1-\sqrt{2}}{4} & \frac{1+\sqrt{2}}{4} \end{pmatrix}$$

Da $\det(A') = 0$ ist, lassen sich die Gewichte im gedrehten Einheitsquadrat nicht eindeutig bestimmen. Daraus kann man folgern, dass die beschriebene Gewichtberechnung im Zweidimensionalen nicht rotationsinvariant ist.

Betrachtet man im \mathbb{R}^3 den Einheitswürfel, der um 45° gedreht ist, dann ist A' :

$$A' = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \frac{1-\sqrt{2}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1+\sqrt{2}}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1-\sqrt{2}}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1+\sqrt{2}}{2} \\ \frac{1-\sqrt{2}}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1+\sqrt{2}}{2} \\ 0 & 0 & 0 & 0 & \frac{1-\sqrt{2}}{2} & \frac{1+\sqrt{2}}{2} & \frac{1-\sqrt{2}}{2} & \frac{1+\sqrt{2}}{2} \end{pmatrix}$$

Die Determinante dieser Matrix A' ist gleich null, womit das Gleichungssystem $A'\omega = b'$ nicht (eindeutig) lösbar ist. Auch im Dreidimensionalen ist also das beschriebene Verfahren nicht rotationsinvariant.

Die Translationsinvarianz alleine reicht nicht aus, um die eindeutige Bestimmung der Gewichte zu gewährleisten. Um sicherzugehen, dass die Gewichte eindeutig und allgemein berechenbar sind, muss man die Koordinaten transformieren, was im folgenden Kapitel beschrieben wird.

4 Koordinatentransformation in lokale Koordinaten

Da die Gewichtberechnung nach Satz 3 nicht rotationsinvariant ist und die Lösbarkeit der Gleichungssysteme nicht allgemein vorausgesetzt werden kann, bedarf es weiterer Anpassungen, um alle Gewichte eindeutig berechnen zu können.

Das Problem besteht darin, dass die Koordinaten der Stützpunkte P_i und des Interpolationspunktes Q in einem globalen Koordinatensystem gegeben sind und somit durchaus in einer Konstellation stehen können, so dass mindestens zwei Zeilen der Matrix A linear abhängig sind, und das Gleichungssystem somit nicht lösbar ist.

Um dies zu umgehen, werden die Koordinaten der Punkte in ein lokales Koordinatensystem transformiert. Dabei wird ein P_i als neuer Koordinatenursprung festgelegt. Als Basis des lokalen Systems werden Vektoren berechnet, die diejenigen Kanten des Vierecks im \mathbb{R}^2 bzw. Hexaeders im \mathbb{R}^3 beschreiben, welche den gewählten Ursprungspunkt als Eckpunkt besitzen. Bezüglich dieser Basis erhalten die restlichen P_j , $i \neq j$, transformierte, lokale Koordinaten. Mit diesen wird dann das Gleichungssystem $A'\omega = b'$ aufgestellt und damit die Gewichte ω_i berechnet.

Im Folgenden wird die Vorgehensweise zur Koordinatentransformation in lokale Koordinaten dargestellt, jeweils für den ein-, zwei- und dreidimensionalen Fall [1].

4.1 Lokale Koordinaten im Zweidimensionalen

Gegeben seien die vier Stützpunkte $P_i = (x_i, y_i)$ mit $i \in \{SW, SE, NW, NE\}$ und der zu interpolierende Punkt $Q = (x_Q, y_Q)$. Ziel ist es, die P_i sowie Q bezüglich den lokalen Koordinaten η_i und ξ_i darzustellen. Sie werden dann als $P'_i = (\eta_i, \xi_i)$ und $Q' = (\eta_Q, \xi_Q)$ bezeichnet.

O.B.d.A. wird P'_{SW} als neuer Koordinatenursprung festgelegt, er liegt also immer bei $(\eta_{SW}, \xi_{SW}) = (0, 0)$. Zunächst werden die Vektoren $v_{SE} = P'_{SE} - P'_{SW}$ und $v_{NW} = P'_{NW} - P'_{SW}$ als Basis des lokalen Koordinatensystems berechnet, wobei v_S und v_W zwei der vier Kanten des Vierecks $P_{SW}P_{SE}P_{NW}P_{NE}$ darstellen, v_{SE} jene, welche P_{SW} und P_{SE} verbindet, und v_{NW} ist die Verbindung der Punkte P_{SW} und P_{NW} .

Für den Eckpunkt P_{SE} werden ebenfalls fixe lokale Koordinaten festgelegt, nämlich $P'_{SE} = (\eta_{SE}, \xi_{SE}) = (1, 0)$. Analog liegt P'_{NW} fest bei $(\eta_{NW}, \xi_{NW}) = (0, 1)$.

Es befinden sich also drei der vier Stützpunkte an immer gleichen lokalen Koordinaten. Nun gilt es η_{NE} und ξ_{NE} für den verbleibenden Stützpunkt P_{NE} und η_Q sowie ξ_Q für den Interpolationspunkt Q zu berechnen.

P'_{NE} und Q' sind also die einzigen Punkte, die veränderliche lokale Koordinaten haben. Die allgemeine Berechnung von η_O und ξ_O eines Punktes O verhält sich folgendermaßen:

$$P'_{SW} + \eta_O \cdot v_{SE} + \xi_O \cdot v_{NW} = O' \quad (18)$$

Für P'_{NE} bzw. Q' muss also gelten:

$$P'_{SW} + \eta_{NE} \cdot v_{SE} + \xi_{NE} \cdot v_{NW} = P'_{NE} \quad (19) \quad \text{bzw.} \quad P'_{SW} + \eta_Q \cdot v_{SE} + \xi_Q \cdot v_{NW} = Q' \quad (20)$$

Die Berechnung der lokalen Koordinaten für einen Punkt O ist also nichts anderes als die Darstellung des Vektors $v_O = O' - P'_{SW}$ bezüglich der Basis $V = (v_{SE}, v_{NW})$ [1].

Jene lässt sich als lineares Gleichungssystem $M \begin{pmatrix} \eta_O \\ \xi_O \end{pmatrix} = v_O$ schreiben:

$$M \begin{pmatrix} \eta_O \\ \xi_O \end{pmatrix} = \begin{pmatrix} v_{SE1} & v_{NW1} \\ v_{SE2} & v_{NW2} \end{pmatrix} \begin{pmatrix} \eta_O \\ \xi_O \end{pmatrix} = v_O \quad (21)$$

Dieses lässt sich beispielsweise mit dem Gaußschen Eliminationsverfahren lösen, um η_O und ξ_O zu erhalten. Hat man mit (21) die lokalen Koordinaten für $P'_{NE} = (\eta_{NE}, \xi_{NE})$ und $Q' = (\eta_Q, \xi_Q)$ berechnet, kann man diese nutzen, um die Gewichte zu errechnen. Es ergibt sich dann für $A'\omega = b'$:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ \eta_{SW} & \eta_{SE} & \eta_{NW} & \eta_{NE} \\ \xi_{SW} & \xi_{SE} & \xi_{NW} & \xi_{NE} \\ \eta_{SW}\xi_{SW} & \eta_{SE}\xi_{SE} & \eta_{NW}\xi_{NW} & \eta_{NE}\xi_{NE} \end{pmatrix} \begin{pmatrix} \omega_{SW} \\ \omega_{SE} \\ \omega_{NW} \\ \omega_{NE} \end{pmatrix} = \begin{pmatrix} 1 \\ \eta_Q \\ \xi_Q \\ \eta_Q\xi_Q \end{pmatrix}$$

und damit

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & \eta_{NE} \\ 0 & 0 & 1 & \xi_{NE} \\ 0 & 0 & 0 & \eta_{NE}\xi_{NE} \end{pmatrix} \begin{pmatrix} \omega_{SW} \\ \omega_{SE} \\ \omega_{NW} \\ \omega_{NE} \end{pmatrix} = \begin{pmatrix} 1 \\ \eta_Q \\ \xi_Q \\ \eta_Q\xi_Q \end{pmatrix}. \quad (22)$$

Satz 4 Das lineare Gleichungssystem (22) ist eindeutig lösbar.

Beweis. Die Determinante von A berechnet sich als

$$\det(A) = \eta_{NE}\xi_{NE} \cdot \det \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \eta_{NE}\xi_{NE}.$$

Die Determinante ist also ungleich 0, wenn sowohl η_{NE} als auch ξ_{NE} ungleich 0 sind. Dass dies der Fall ist, lässt sich an (21) erkennen. Wäre η_O oder ξ_O hier gleich 0, dann wäre v_O ein Vielfaches von einem der Basisvektoren v_{SE} oder v_{NW} und damit würde ursprünglich kein echtes Viereck zugrunde liegen, da der vierte Punkt auf einer Linie mit P_{SW} und P_{SE} oder P_{SW} und P_{NW} läge. Wäre der Punkt P'_{NE} bei $(\eta_{NE}, \xi_{NE}) = (0, 0)$, so wäre er identisch mit P_{SW} .

Also sind η_{NE} und ξ_{NE} beide ungleich null und damit $\det(A) \neq 0$, womit (22) eindeutig lösbar ist. \square

4.2 Lokale Koordinaten im Dreidimensionalen

An sich verfährt man im \mathbb{R}^3 analog zum \mathbb{R}^2 , nur dass es nun natürlich drei Koordinatenachsen gibt.

Wie im Zweidimensionalen sind die Eckpunkte $P_i = (x_i, y_i, z_i)$, $i \in \{SWD, SED, NWD, NED, SWT, SET, NWT, NET\}$, zusammen mit dem Interpolationspunkt $Q = (x_Q, y_Q, z_Q)$ gegeben. Es sind alle neun Punkte folgendermaßen mit lokalen Koordinaten darzustellen: $P'_i = (\eta_i, \xi_i, \rho_i)$ bzw. $Q' = (\eta_Q, \xi_Q, \rho_Q)$.

Auch hier wird wieder einer der P_i als Koordinatenursprung $(0, 0, 0)$ festgelegt, o.B.d.A. sei dies P_{SWD} . Zusätzlich wählt man drei weitere Punkte der P_i , nämlich diejenigen, welche sich eine Kante mit P_{SWD} teilen, also P_{SED} , P_{SWT} und P_{NWD} . Diese sind fix an den lokalen Koordinaten $P'_{SED} = (1, 0, 0)$, $P'_{NWD} = (0, 1, 0)$ und $P'_{SWT} = (0, 0, 1)$. Die restlichen vier P_i und Q sind zu berechnen.

Dazu ermittelt man die Vektoren $v_{SED} = P'_{SED} - P'_{SWD}$, $v_{NWD} = P'_{NWD} - P'_{SWD}$ und $v_{SWT} = P'_{SWT} - P'_{SWD}$, welche eine Basis V im \mathbb{R}^3 bilden. Bezüglich V werden dann die lokalen Koordinaten der übrigen Punkte berechnet.

Die allgemeine Berechnung ist wie im \mathbb{R}^2 nur mit der zusätzlichen ρ -Koordinate. Für η_O , ξ_O und ρ_O eines beliebigen Punktes O muss dabei gelten:

$$P'_{SWD} + \eta_O \cdot v_{SED} + \xi_O \cdot v_{NWD} + \rho_O \cdot v_{SWT} = O'. \quad (23)$$

Subtrahiert man bei (23) auf beiden Seiten P_{SWD}^T erhält man damit

$$\eta_O \cdot v_{SED} + \xi_O \cdot v_{NWD} + \rho_O \cdot v_{SWT} = v_O, \quad (24)$$

also die Darstellung des Vektors v_O bezüglich der Basis $V = (v_{SED}, v_{NWD}, v_{SWT})$.

Als lineares Gleichungssystem ist (24):

$$M \begin{pmatrix} \eta_O \\ \xi_O \\ \rho_O \end{pmatrix} = \begin{pmatrix} v_{SED1} & v_{NWD1} & v_{SWT1} \\ v_{SED2} & v_{NWD2} & v_{SWT2} \\ v_{SED3} & v_{NWD3} & v_{SWT3} \end{pmatrix} \begin{pmatrix} \eta_O \\ \xi_O \\ \rho_O \end{pmatrix} = v_O \quad (25)$$

In (25) setzt man für die übrigen P_i die jeweiligen $v_i = P_i^T - P_{SWD}^T$ ein und löst es beispielsweise mit dem Gauß-Verfahren. Damit erhält man schließlich die restlichen lokalen Koordinaten der übrigen Punkte. Zusammen mit den fixen Koordinaten für die Punkte P_{SWD} , P_{SED} , P_{NWD} und P_{SWT} ergibt sich das Gleichungssystem $A'\omega = b'$ aus (17):

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & \eta_{NED} & \eta_{SET} & \eta_{NWT} & \eta_{NET} & \omega_{SWD} \\ 0 & 0 & 1 & 0 & \xi_{NED} & \xi_{SET} & \xi_{NWT} & \xi_{NET} & \omega_{SED} \\ 0 & 0 & 0 & 1 & \rho_{NED} & \rho_{SET} & \rho_{NWT} & \rho_{NET} & \omega_{NWD} \\ 0 & 0 & 0 & 0 & \eta_{NED}\xi_{NED} & \eta_{SET}\xi_{SET} & \eta_{NWT}\xi_{NWT} & \eta_{NET}\xi_{NET} & \omega_{SWT} \\ 0 & 0 & 0 & 0 & \eta_{NED}\rho_{NED} & \eta_{SET}\rho_{SET} & \eta_{NWT}\rho_{NWT} & \eta_{NET}\rho_{NET} & \omega_{NED} \\ 0 & 0 & 0 & 0 & \xi_{NED}\rho_{NED} & \xi_{SET}\rho_{SET} & \xi_{NWT}\rho_{NWT} & \xi_{NET}\rho_{NET} & \omega_{SET} \\ 0 & 0 & 0 & 0 & (\eta\xi\rho)_{NED} & (\eta\xi\rho)_{SET} & (\eta\xi\rho)_{NWT} & (\eta\xi\rho)_{NET} & \omega_{NWT} \\ 0 & 0 & 0 & 0 & & & & & \omega_{NET} \end{pmatrix} \begin{pmatrix} \omega_{SWD} \\ \omega_{SED} \\ \omega_{NWD} \\ \omega_{SWT} \\ \omega_{NED} \\ \omega_{SET} \\ \omega_{NWT} \\ \omega_{NET} \end{pmatrix} = \begin{pmatrix} 1 \\ \eta_Q \\ \xi_Q \\ \rho_Q \\ \eta_Q\xi_Q \\ \eta_Q\rho_Q \\ \xi_Q\rho_Q \\ \eta_Q\xi_Q\rho_Q \end{pmatrix}. \quad (26)$$

Auch dieses System ist lösbar, wenn die P_i wirklich einen Hexaeder beschreiben, also kein v_i ein Vielfaches eines v_j , $i \neq j$ ist.

4.3 Berechnung lokaler Koordinaten für vier Punkte in einer Ebene im \mathbb{R}^3

Es gibt einen dritten Fall, der die Berechnung lokaler Koordinaten notwendig macht. Sind vier Stützpunkte P_i gegeben, die zwar im \mathbb{R}^3 liegen, aber trotzdem in einer Ebene, kann man nicht wie in (8) vorgehen. Beispielsweise wenn die P_i in der xz-Ebene liegen, da dort die y-Koordinate immer den selben Wert hat, weshalb die y-Zeile der Matrix nur ein Vielfaches der ersten Zeile ist und das System damit nicht lösbar ist. Allerdings können die lokalen Koordinaten auch nicht wie in 4.1 berechnet werden, da sich (18) zu Folgendem ändert:

$$\begin{pmatrix} x_{SW} \\ y_{SW} \\ z_{SW} \end{pmatrix} + \eta_O \cdot \begin{pmatrix} v_{SE1} \\ v_{SE2} \\ v_{SE3} \end{pmatrix} + \xi_O \cdot \begin{pmatrix} v_{NW1} \\ v_{NW2} \\ v_{NW3} \end{pmatrix} = \begin{pmatrix} x_O \\ y_O \\ z_O \end{pmatrix},$$

bzw.

$$M \begin{pmatrix} \eta_O \\ \xi_O \end{pmatrix} = \begin{pmatrix} v_{SE1} & v_{NW1} \\ v_{SE2} & v_{NW2} \\ v_{SE3} & v_{NW3} \end{pmatrix} \begin{pmatrix} \eta_O \\ \xi_O \end{pmatrix} = \begin{pmatrix} x_O - x_{SW} \\ y_O - y_{SW} \\ z_O - z_{SW} \end{pmatrix} = v_O. \quad (27)$$

Das sind drei Gleichungen für nur zwei Unbekannte, also ein überbestimmtes Gleichungssystem. Um dennoch die lokalen Koordinaten bestimmen zu können, wird hier die Least-Squares Methode angewandt [2]. Dazu wird bei (27) auf beiden Seiten von links die transponierte Matrix M^T multipliziert:

$$M^T M \begin{pmatrix} \eta_O \\ \xi_O \end{pmatrix} = \begin{pmatrix} v_{SE1} & v_{SE2} & v_{SE3} \\ v_{NW1} & v_{NW2} & v_{NW3} \end{pmatrix} \begin{pmatrix} v_{SE1} & v_{NW1} \\ v_{SE2} & v_{NW2} \\ v_{SE3} & v_{NW3} \end{pmatrix} = \begin{pmatrix} v_{SE1} & v_{SE2} & v_{SE3} \\ v_{NW1} & v_{NW2} & v_{NW3} \end{pmatrix} \begin{pmatrix} x_O - x_{SW} \\ y_O - y_{SW} \\ z_O - z_{SW} \end{pmatrix}. \quad (28)$$

Da $M^T M$ eine 2x2-Matrix und $M^T v_O \in \mathbb{R}^2$, lassen sich mit (28) eindeutige Werte für die lokalen Koordinaten (η_O, ξ_O) für einen Punkt O bestimmen.

Das Ganze ist allerdings nur dann nötig, wenn die P_i tatsächlich in einer Ebene liegen. Sind sie hingegen nicht in einer Ebene, sodass sie ein Volumen beschreiben, lässt sich die Gewichtberechnung von (8) auf folgendes vereinfachen.

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ x_{SW} & x_{SE} & x_{NW} & x_{NE} \\ y_{SW} & y_{SE} & y_{NW} & y_{NE} \\ z_{SW} & z_{SE} & z_{NW} & z_{NE} \end{pmatrix} \begin{pmatrix} \omega_{SW} \\ \omega_{SE} \\ \omega_{NW} \\ \omega_{NE} \end{pmatrix} = \begin{pmatrix} 1 \\ x_Q \\ y_Q \\ z_Q \end{pmatrix} \quad (29)$$

Dann ist es also nicht notwendig, lokale Koordinaten zu berechnen. Das ist hier möglich, weil man die z-Koordinate als zusätzliche Information hat, welche benutzt werden kann, um die Gewichte eindeutig zu bestimmen.

Hat man vier Punkte im \mathbb{R}^3 gegeben, kann man also überprüfen, ob sie in einer Ebene liegen. Ist dies der Fall, muss man nach (28) vorgehen, ansonsten genügt es, das System (29) zu lösen. Allerdings ist es auch möglich, immer die lokalen Koordinaten zu berechnen, was jedoch einen gewissen Mehraufwand bedeutet.

Zur Überprüfung, ob die vier Punkte in einer Ebene liegen, kann man die Vektoren v_{SE} und v_{NW} nutzen. Zuerst bildet man das Kreuzprodukt aus beiden, um einen Vektor zu erhalten, der orthogonal zu v_{SE} und v_{NW} liegt. Ist das Skalarprodukt aus diesem Orthogonalvektor und $v_{NE} = P_{NE}^T - P_{SW}^T$ gleich null, so ist v_{NE} senkrecht zum Orthogonalvektor. Damit sind v_{SE} , v_{NW} und v_{NE} in einer Ebene und somit alle vier P_i . Die Gleichung zum Prüfen, ob die P_i in einer Ebene liegen, ist also:

$$\langle v_{SE} \times v_{NW}, v_{NE} \rangle \stackrel{?}{=} 0. \quad (30)$$

Damit können alle benötigten lokalen Koordinaten berechnet werden.

Die Vorgehensweise zur Berechnung der Gewichte ist somit vollständig beschrieben. Nach einem nun folgenden Beispiel wird gezeigt, wie das Verfahren in einem bestehenden Code integriert wurde und welche Ergebnisse damit erzielt werden konnten.

5 Beispielrechnung für vier Gewichte im \mathbb{R}^2

Um zuvor Genanntes zu veranschaulichen, folgt ein konkretes Beispiel, wie man vorgeht, um vier Gewichte im \mathbb{R}^2 zu berechnen.

Hierfür seien die vier Stützpunkte $P_S = (\frac{1}{2}, \frac{1-\sqrt{2}}{2})$, $P_W = (\frac{1-\sqrt{2}}{2}, \frac{1}{2})$, $P_E = (\frac{1+\sqrt{2}}{2}, \frac{1}{2})$ und $P_N = (\frac{1}{2}, \frac{1+\sqrt{2}}{2})$ und des Weiteren $Q = (\frac{3}{4}, \frac{1}{2})$ gegeben.

Zuerst werden die lokalen Koordinaten berechnet. Sei P_S der Ursprung. Es müssen also die Vektoren $v_W = P_W^T - P_S^T$ und $v_E = P_E^T - P_S^T$ berechnet werden, welche dann die Basis $V = (v_W, v_E)$ für das lokale Koordinatensystem bilden.

$$v_W = \begin{pmatrix} \frac{1-\sqrt{2}}{2} \\ \frac{1}{2} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} \\ \frac{1-\sqrt{2}}{2} \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

$$v_E = \begin{pmatrix} \frac{1+\sqrt{2}}{2} \\ \frac{1}{2} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} \\ \frac{1-\sqrt{2}}{2} \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix}.$$

Außerdem gilt:

$$v_N = P_N^T - P_S^T = \begin{pmatrix} \frac{1}{2} \\ \frac{1+\sqrt{2}}{2} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} \\ \frac{1-\sqrt{2}}{2} \end{pmatrix} = \begin{pmatrix} 0 \\ \sqrt{2} \end{pmatrix}$$

$$v_Q = Q^T - P_S^T = \begin{pmatrix} \frac{3}{4} \\ \frac{1}{2} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} \\ \frac{1-\sqrt{2}}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{\sqrt{2}}{2} \end{pmatrix}.$$

Damit lässt sich P'_N und Q' berechnen.

$$M \begin{pmatrix} \eta_N \\ \xi_N \end{pmatrix} = \begin{pmatrix} v_{W1} & v_{E1} \\ v_{W2} & v_{E2} \end{pmatrix} \begin{pmatrix} \eta_N \\ \xi_N \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \eta_N \\ \xi_N \end{pmatrix} = \begin{pmatrix} 0 \\ \sqrt{2} \end{pmatrix}$$

$$\rightarrow (\eta_N, \xi_N) = (1, 1),$$

$$M \begin{pmatrix} \eta_Q \\ \xi_Q \end{pmatrix} = \begin{pmatrix} v_{W1} & v_{E1} \\ v_{W2} & v_{E2} \end{pmatrix} \begin{pmatrix} \eta_Q \\ \xi_Q \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \eta_Q \\ \xi_Q \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{\sqrt{2}}{2} \end{pmatrix}$$

$$\rightarrow (\eta_Q, \xi_Q) = \left(\frac{-\sqrt{2} + 4}{8}, \frac{\sqrt{2} + 4}{8} \right).$$

Schließlich kann man das Gleichungssystem $A'\omega = b'$ aufstellen:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \omega_S \\ \omega_W \\ \omega_E \\ \omega_N \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{-\sqrt{2}+4}{8} \\ \frac{\sqrt{2}+4}{8} \\ \frac{-\sqrt{2}+4}{8} \cdot \frac{\sqrt{2}+4}{8} \end{pmatrix}$$

$$\rightarrow (\omega_S, \omega_W, \omega_E, \omega_N) = \left(\frac{7}{32}, \frac{-4\sqrt{2} + 9}{32}, \frac{4\sqrt{2} + 9}{32}, \frac{7}{32} \right) \approx (0.22, 0.10, 0.46, 0.22).$$

Man sieht, dass das Gewicht ω_E für den Punkt P_E am größten ist, was die Tatsache widerspiegelt, dass Q am nächsten an P_E von den vier P_i liegt. Gleichzeitig ist das Gewicht ω_W am kleinsten, da Q zu P_W die größte Entfernung aufweist. Die übrigen beiden Gewichte ω_S und ω_N haben hingegen den gleichen Wert, da Q zu beiden äquidistant ist.

Dem gegenüber steht das unveränderte Mehrgitterverfahren, bei dem die $\omega_i = \frac{1}{4} \forall i$ sind, also ein deutlicher Unterschied zu den berechneten Gewichten.

6 Integration in das Mehrgitterverfahren im Programm UGBlocks

Da nun das Verfahren zur Berechnung der Gewichte dargestellt worden ist, gilt es dieses in das Mehrgitterverfahren zu integrieren. Dafür wird das bestehende Mehrgitterverfahren, wie es in dem Programm UGBlocks [5] durchgeführt wird, abgeändert, sodass in der Prolongation und Restriktion das neue Verfahren zum Berechnen von Gewichten für die Punkte genutzt wird, statt immer gleiche, feste Gewichte zu verwenden.

Das zuvor genannte Verfahren bezieht sich jeweils immer auf eine einzelne „Zelle“. Die im Mehrgitterverfahren vorkommenden Gitter bestehen aus mehreren Zellen, für welche jeweils die Gewichte für die Punkte auf dem groben Gitter bezüglich eines Punktes auf dem feinen Gitter zu berechnen sind.

Bevor also Restriktion und Prolongation durchgeführt werden können, müssen die Gewichte bereits berechnet sein. Diese müssen damit noch vor dem Aufruf der Prolongations- oder Restriktionsoperatoren berechnet und gespeichert sein.

In UGBlocks ist dafür eine neue Klasse erstellt worden, die die neuen Prolongations- und Restriktionsoperatoren beinhaltet. Sobald eine Instanz dieser Klasse erstellt wird, wird die Berechnung der Gewichte im Konstruktor der Klasse ausgeführt, zuerst für Hexaeder im Dreidimensionalen, dann für Flächen im Zweidimensionalen und schließlich für Kanten im Eindimensionalen. Das Vorgehen im Code ist dabei für alle drei Fälle immer das Gleiche.

Für einen Hexaeder müssen für jede grobe Zelle acht Gewichte von acht Eckpunkten auf dem groben Gitter bezüglich einem Punkt Q auf dem feinen Gitter berechnet werden. Es wird über alle groben Zellen des gegebenen Gitters iteriert, also eine Schleife für die x -, eine für die y - und eine für die z -Richtung. In jeder Iteration hat man also die Koordinaten der nächsten acht Eckpunkte auf dem groben Gitter und die des Punktes Q auf dem feinen Gitter gegeben. Diese werden zunächst abgerufen.

Dann werden die lokalen Koordinaten dieser neun Punkte berechnet, um sicher zu gehen, dass das folgende Gleichungssystem lösbar ist, welches als $A'\omega = b'$ wie in (26) aufgestellt und anschließend mit dem Gauß-Verfahren gelöst wird. Die Werte der jeweils acht Gewichte werden in einem Array gespeichert, sodass sie später bei der Restriktion und Prolongation abgerufen werden können.

Dann folgt die Iteration über die einzelnen Oberflächen der groben Zellen, wobei pro Iterationsschritt jeweils vier Gewichte zu berechnen sind. Dabei wird in der Richtung der Oberfläche unterschieden: Es werden nacheinander die Gewichte für die Oberflächen der x -, y - und schließlich der z -Richtung berechnet. Dabei wird jeweils wieder mit drei Schleifen über alle groben Zellen iteriert.

In jeder Iteration werden die Koordinaten der vier Eckpunkte auf dem groben Gitter und des Interpolationspunktes auf dem feinen Gitter aufgerufen. Dann folgt eine Unterscheidung wie in Kapitel 4.3 angedeutet: Es wird getestet (30), ob die vier Punkte tatsächlich in einer Ebene liegen, oder eigentlich ein Volumen beschreiben. Ist letzteres der Fall, kann das Gleichungssystem wie in (29) aufgestellt und gelöst werden. Sind die Punkte jedoch wirklich in einer Ebene, müssen auch hier die lokalen Koordinaten berechnet werden (siehe 4.3). Das mit diesen aufgestellte Gleichungssystem wird mit dem Gauß-Verfahren gelöst und die Gewichte gespeichert.

Dies wird so für jede Oberfläche durchgeführt, wonach lediglich noch die Gewichte auf den Kanten des Gitters zu bestimmen sind.

Dabei werden analog zu den Flächen nacheinander die Gewichte auf den Kanten in x -, y - und z -Richtung berechnet, wobei auch hier jeweils über alle groben Zellen iteriert wird.

Auf jeder Kante sind zwei grobe Gitterpunkte bzw. deren Koordinaten abzurufen, sowie die des feinen Gitterpunktes, bezüglich welchem die Gewichte jeweils zu berechnen sind. Dies wird wie in Kapitel 2.1 beschrieben durchgeführt, also das System aufgestellt und mit dem Gauß-Verfahren gelöst. Die Gewichte werden gespeichert, damit sie später für die Prolongation und Restriktion aufgerufen werden können.

Das beschließt die Gewichtberechnung auf einem dreidimensionalen Gitter. Auf dem zweidimensionalen Gitter geht man analog vor, nur dass der erste Schritt, also die Interpolation des Hexaeders wegfällt und man bei der Fläche beginnt, wobei dann für die lokalen Koordinaten das entsprechende Kapitel 4.1 die Vorgehensweise darstellt. Ebenso fällt die Interpolation auf der Fläche bei der schließlich folgenden Berechnung auf dem eindimensionalen Gitter weg.

Die Berechnung der Gewichte ist damit abgeschlossen. Da das Verfahren sowie die Prolongation den feinen Gitterpunkt interpolieren, können für die Prolongation die Gewichte ganz einfach für jeden Punkt abgerufen werden, analog dazu wie sie berechnet wurden.

Bei der Restriktion hingegen, werden die Gewichte in die entgegengesetzte Richtung benötigt. Das heißt ein Gewicht ω_i , welches für den Punkt P_i bezüglich dem Interpolationspunkt Q berechnet wurde, ist nun auf den Punkt Q anzuwenden. Mit etwas mehr Rechenaufwand können auch die Gewichte explizit für die Restriktion berechnet werden, wodurch sich deren Aufruf einfacher gestaltet. Dabei sei jedoch angemerkt, dass dabei keine neuen Gewichte berechnet werden, sie sind identisch mit den zuvor berechneten, die bei der Prolongation angewandt werden. Die erneute Berechnung dient nur der Einfachheit beim Aufruf.

Im Zuge dieser erneuten Berechnung sind also die Interpolationspunkte, bezüglich derer die Gewichte berechnet werden, auf dem groben und die Stützpunkte P_i auf dem feinen Gitter. Ansonsten ist die Vorgehensweise identisch mit der zuvor beschriebenen.

Auch bei der Restriktion der lokalen Steifigkeitsmatrizen werden die berechneten Gewichte verwendet.

7 Test der Konvergenzgeschwindigkeit anhand der Poisson-Gleichung

Mit dem Integrieren der Gewichtberechnung in das Mehrgitterverfahren von UGBlocks [5] kann nun die Konvergenzgeschwindigkeit getestet und mit dem herkömmlichen Mehrgitterverfahren mit konstanten Gewichten verglichen werden. Dieser Test wird in diesem letzten Kapitel für verschiedene geometrische Figuren anhand der Poisson-Gleichung durchgeführt.

Dafür sei die Poisson-Gleichung

$$-\Delta u = f \quad (31)$$

mit der Randbedingung

$$u|_{\partial\Omega} = g \quad (32)$$

gegeben [4]. Um die Konvergenzgeschwindigkeit zu testen, wird $f = 0$ gesetzt und des Weiteren gilt für den Anfangswert der Näherungslösung u^0 :

$$u^0 = 1 \quad \text{und} \quad u^0|_{\partial\Omega} = 0.$$

Also setzt man die Anfangs-Näherungslösung gleich eins im Inneren und gleich null auf dem Rand, womit die Lösung für u gegen null konvergieren muss.

Die Konvergenzgeschwindigkeit δ_{MG} wird folgendermaßen berechnet:

$$\lim_{k \rightarrow \infty} \frac{\|u^{k+1}\|}{\|u^k\|} = \delta_{MG}. \quad (33)$$

Im Folgenden wird die Poisson-Gleichung auf verschiedenen geometrischen Figuren mit dem Mehrgitterverfahren gelöst und die Konvergenzgeschwindigkeit berechnet.

Zuerst für den Einheitswürfel, für welchen kein Unterschied zum vorherigen Standardmehrgitterverfahren erwartet wird, da die Stützpunkte auf dem groben Gitter jeweils alle äquidistant zu den zu interpolierenden Punkten sind und damit die berechneten Gewichte identisch zu den herkömmlichen sein sollten.

In der Tat sind die Gewichte exakt gleich mit den nicht berechneten, was auf die Äquidistanz der Punkte zurückzuführen ist. Die Konvergenzgeschwindigkeit bleibt also unverändert.

Für ein Zylinder mit ungekrümmten Innenflächen ergeben sich für Eckgewichte in der Mitte des Gitters beispielsweise folgende Gewichte:

$$\begin{pmatrix} \omega_{SWD} \\ \omega_{SED} \\ \omega_{NWD} \\ \omega_{NED} \\ \omega_{SWT} \\ \omega_{SET} \\ \omega_{NWT} \\ \omega_{NET} \end{pmatrix} = \begin{pmatrix} 0.1642259581 \\ 0.0644781599 \\ 0.1030062861 \\ 0.1642259581 \\ 0.1682895959 \\ 0.0644781599 \\ 0.1030062861 \\ 0.1682895959 \end{pmatrix}.$$

Mit den herkömmlichen Mehrgitteroperatoren sind alle Gewichte $\frac{1}{8}$.

Iteration	mit festen Gewichten	mit berechneten Gewichten
1	0.0514960789	0.0514916726
2	0.0927755719	0.0919682127
3	0.1159110709	0.1158416169
4	0.1256696584	0.1257077630
5	0.1329329374	0.1329297454
6	0.1385334730	0.1385051424
7	0.1424131728	0.1423766836
8	0.1449180342	0.1448860693
9	0.1465266349	0.1465052976
10	0.1476197132	0.1476102871

Tabelle 1: Werte für $\frac{\|u^{k+1}\|}{\|u^k\|}$ für die Poisson-Gleichung auf einem Zylinder mit ungekrümmten Innenflächen

Die Konvergenzgeschwindigkeit ist also etwas geringer mit den neuen Operatoren.

Für die Gewichte eines Slabs (3D-Trapez) in einer Ecke des Gitters ergeben sich für die Eckpunkte zum Beispiel folgende Gewichte:

$$\begin{pmatrix} \omega_{SWD} \\ \omega_{SED} \\ \omega_{NWD} \\ \omega_{NED} \\ \omega_{SWT} \\ \omega_{SET} \\ \omega_{NWT} \\ \omega_{NET} \end{pmatrix} = \begin{pmatrix} 0.1169738423 \\ 0.1169738423 \\ 0.1321127490 \\ 0.1330261577 \\ 0.1321127490 \\ 0.1330261577 \\ 0.1178872510 \\ 0.1178872510 \end{pmatrix}.$$

Die Gewichte auf einer Oberfläche einer groben Zelle in der Mitte des Gitters sind beispielsweise:

$$\begin{pmatrix} \omega_{SD} \\ \omega_{ND} \\ \omega_{ST} \\ \omega_{NT} \end{pmatrix} = \begin{pmatrix} 0.2419738423 \\ 0.2575418980 \\ 0.2580261577 \\ 0.2424581020 \end{pmatrix}.$$

Iteration	mit festen Gewichten	mit berechneten Gewichten
1	0.0861987936	0.0862018001
2	0.3471465719	0.3471326141
3	0.3962114904	0.3962090840
4	0.4122894130	0.4122867263
5	0.4206397574	0.4206368370
6	0.4266757961	0.4266717943
7	0.4324637821	0.4324567092
8	0.4388178199	0.4388060707
9	0.4458381210	0.4458214828
10	0.4531531508	0.4531333645

Tabelle 2: Werte für $\frac{\|u^{k+1}\|}{\|u^k\|}$ für die Poisson-Gleichung auf einem Slab

Auch für einen Slab konvergiert das Verfahren mit den neuen Operatoren etwas schneller für die Poisson-Gleichung.

Zusammenfassend kann man also feststellen, dass die Konvergenzgeschwindigkeit ein wenig verbessert werden konnte. Es sei angemerkt, dass bei der Poisson-Gleichung keine großen Unterschiede zu erwarten sind, da es sich nicht um ein strukturmechanisches Problem handelt. Allerdings ist zu erkennen, dass die neuen Operatoren funktionieren und etwas passendere Gewichte berechnet werden.

8 Zusammenfassung und Ausblick

Damit sind die Inhalte dieser Arbeit abgeschlossen. Ziel war es, die Mehrgitteroperatoren Prolongation und Restriktion so anzupassen, dass alle Funktionen der Rigid Body Modes korrekt prolongiert und restringiert werden, um dadurch eine Steigerung der Konvergenzgeschwindigkeit des Mehrgitterverfahrens für die Strukturmechanik zu erreichen.

Dafür wurde ein Ansatz gewählt, der für Punkte auf dem groben Gitter Gewichte bezüglich eines Punktes auf dem feinen Gitter berechnet, die abhängig von der Lage der Punkte im Raum sind.

Es wird dabei unterschieden, ob es sich um zwei Gewichte für zwei Punkte (eindimensionaler Fall), um vier Gewichte für vier Punkte (zweidimensionaler Fall), oder um acht Gewichte für acht Punkte (dreidimensionaler Fall) handelt. Abhängig davon muss die Berechnung der Gewichte angepasst werden.

Man benötigt also zwei, vier oder acht Gleichungen und erhält dadurch verschieden große Gleichungssysteme, die es zu lösen gilt. Diese sind dabei nicht allgemein lösbar, weshalb eine Möglichkeit vorgestellt wurde, wie man lokale Koordinaten berechnet, mit Hilfe derer angepasste Gleichungssysteme aufgestellt werden können, welche lösbar sind, womit man dann alle Gewichte berechnen kann, die für Restriktion und Prolongation benötigt werden.

Anschließend wurde das Verfahren in das Programm UGBlocks integriert, um für verschiedene geometrischen Figuren die Konvergenzgeschwindigkeit des angepassten Mehrgitterverfahrens testen zu können. Diese wurde mit der des herkömmlichen Verfahrens mit dessen Prolongation und Restriktion verglichen.

Bereits für die Poisson-Gleichung konnten kleinere Verbesserungen in der Konvergenzgeschwindigkeit mit leicht abweichenden Gewichten gegenüber den nicht berechneten erzielt werden. Größere Unterschiede sind bei tatsächlichen Problemen der Strukturmechanik zu erwarten, was eine Weiterführung des Themas angehen könnte.

9 Literaturverzeichnis

- [1] Ganster, Maximilian (2019): Lineare Abbildungen und Koordinatentransformationen, https://www.math.tugraz.at/~ganster/lv_mathematik_1_bau_ws_2019/09_lineare_abbildungen.pdf, zuletzt abgerufen am: 22.03.23
- [2] Margalit, Dan; Rabinoff, Joseph (2019): Interactive Linear Algebra, <https://textbooks.math.gatech.edu/ila/ila.pdf>, zuletzt aufgerufen am: 22.03.23
- [3] Pflaum, Christoph (2005): Multigrid Methods, <https://www.cs10.tf.fau.de/files/2018/06/pflaum-script-mg.pdf>, zuletzt abgerufen am: 22.03.23
- [4] Pflaum, Christoph (2017): Simulation und wissenschaftliches Rechnen II, <https://www.cs10.tf.fau.de/files/2018/06/pflaum-script-siwir2.pdf>, zuletzt abgerufen am: 26.03.2023
- [5] Pflaum, Christoph (2010): <https://www.cs10.tf.fau.de/research/software/expde/unstructured-block-grids/>
- [6] Şanal, Ziya (2015): Mathematik für Ingenieure: Grundlagen - Anwendungen in Maple, 3. Aufl., Wiesbaden: Springer Vieweg
- [7] Wissmann, Johannes; Sarnes, Klaus-Dieter (2006): Finite Elemente in der Strukturmechanik, Berlin Heidelberg: Springer