Technische Universität Graz

Automatische Positionierung von FEM-Netzen

P. Urthaler, G. Of, O. Steinbach

Berichte aus dem
Institut für Numerische Mathematik

Bericht 2006/8

Technische Universität Graz

Automatische Positionierung von FEM-Netzen

P. Urthaler, G. Of, O. Steinbach

Berichte aus dem
Institut für Numerische Mathematik

Bericht 2006/8

Technische Universität Graz
Institut für Numerische Mathematik
Steyrergasse 30
A 8010 Graz

WWW: `http://www.numerik.math.tu-graz.at`

Automatische Positionierung von FEM-Netzen

Peter Urthaler, Günther Of und Olaf Steinbach

Zusammenfassung

Im Automobilbau sind einerseits globale Simulationen wie beispielsweise Crashberechnungen und lokale Simulationen zur Berechnung der elastischen Verformungen notwendig. Dies verlangt die Verwendung unterschiedlicher Netze und folglich die Abbildung der physikalischen Größen zwischen den verschiedenen Netzen. Dies erfordert die Zuordnung verschiedener teils verformter Netze.

1 Problemstellung und grundlegende Idee

Zwei Oberflächennetze eines Werkstückes sollen zueinander möglichst optimal positioniert werden. Die beiden Netze entstehen folgendermaßen: Zuerst wird das Werkstück vernetzt, im Anschluss leicht verformt, wobei auch Ränder abgeschnitten werden können. Nun wird das verformte und möglicherweise etwas kleinere Werkstück erneut vernetzt. Zusätzlich kann eine der beiden Vernetzungen stark adaptiv sein, während im anderen Fall das Werkstück gleichmässig vernetzt wird. Abbildung 1 zeigt zwei solche Netze eines Bauteils. Dabei beschreibt das linke Netz nur einen Ausschnitt des Bauteils.

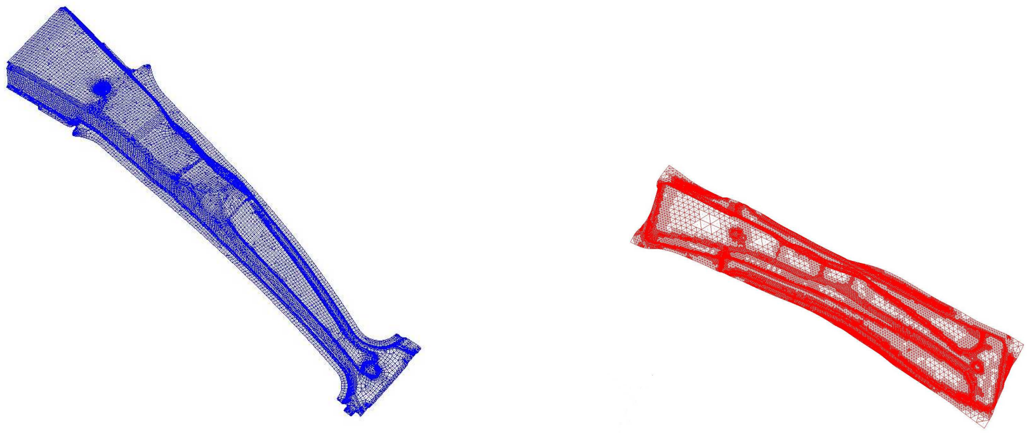


Abbildung 1: Zwei Beispielnetze

Um die optimale Lage der beiden Netze zueinander zu finden, muss zuerst ein geeignetes Kriterium definiert werden, mit Hilfe dessen die Lage der beiden Netze beurteilt werden kann. Die grundlegende Idee dabei ist, zuerst eine grobe Vorpositionierung vorzunehmen und im Anschluss kann unter Verwendung eines geeigneten Abschlusses, siehe Abbildung 2, das Volumen zwischen den Oberflächen als Kriterium verwendet werden. Die Minimierung des Volumens erfolgt durch Translation und Rotation des zweiten Netzes, bis die optimale Lage gefunden wird. Für die Minimierung wird hier ein modifiziertes Gradientenverfahren eingesetzt.

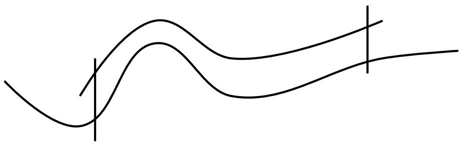


Abbildung 2: Bildung eines Abschluss

2 Bestimmung des Volumens

2.1 Vorpositionierung

Um eine Vorpositionierung durchführen zu können, müssen pro Netz drei Knoten vom Benutzer vorgegeben werden. Daraufhin wird der quadratische Abstand dieser Knoten minimiert und somit beide Netze optimal positioniert. Es bleibt also

$$\min \sum_{n=1}^{3} \|Tx_n - y_n\|_2^2$$

zu minimieren, wobei T die Abbildung, die sich aus den Translationen und Rotationen zusammensetzt, beschreibt. x_1, x_2, x_3 sind drei Punkte aus dem ersten Oberflächennetz und y_1, y_2, y_3 die Punkte aus dem zweiten Oberflächennetz. Zuerst wird dabei nur die Translation betrachtet, und es ergibt sich als Bedingung für das Minimum:

$$\sum_{n=1}^{3} x_n = \sum_{n=1}^{3} y_n.$$

Diese Bedingung lässt sich erfüllen, indem man beide Schwerpunkte übereinander legt. Zusätzlich wird der Schwerpunkt noch in den Ursprung verschoben.
Durch die drei Punkte pro Netz werden zwei Ebenen vorgegeben. Indem man beide Ebenen in die xy-Ebene dreht, eliminiert man zwei weitere Freiheitsgrade. Der letzte Drehwinkel für die Abstandsminimierung ergibt sich als

$$\alpha = \sum_{i=1}^{3} \frac{x_{i2}y_{i1} - x_{i1}y_{i2}}{x_{i1}y_{i1} + x_{i2}y_{i2}}.$$

2.2 Bildung des Abschlusses

Nachdem beide Netze nun vorpositioniert wurden, fehlt noch ein geeigneter Abschluss, um ein Volumen bestimmen zu können. Dieser Abschluss wird ebenfalls durch den Benutzer in Form von Punkten auf einem der beiden Netze vorgegeben. Dieser Bereich wird auf das andere Netz projiziert. Als Projektionsrichtung wird aufgrund der Vorpositionierung die z-Richtung verwendet. Die Abschlussflächen werden nicht vernetzt, da sie bei der Berechnung des Volumens später nicht benötigt werden.

2.3 Berechnung des Volumens

Nachdem ein Abschluss gefunden wurde, kann das Volumen berechnet werden. Dazu wird der Integralsatz von Gauss

$$\int_\Omega \operatorname{div} f(x)\ dx = \int_\Gamma f(x) \cdot n(x)\ ds_x$$

verwendet. Durch diesen Satz kann ein Volumenintegral auf ein Oberflächenintegral zurückgeführt werden. $f(x)$ ist dabei eine beliebige zweimal stetig differenzierbare Funktion und $n(x)$ ist der Normalvektor auf dem Rand, der nach außen zeigt.
Setzt man für $f(x) = (0, 0, z)^\top$, so ergibt sich für die Divergenz der Funktion der Wert eins. Aus dem Integralsatz von Gauss folgt also

$$\int_\Omega dx = \int_\Gamma z\, n_z ds_x = \sum_\ell \int_{\Delta_\ell} z\, n_z ds_x.$$

Auf diese Weise kann das Volumenintegral auf eine Summe von Oberflächenintegrale zurückgeführt werden. Bei Elementen mit einem Normalvektor der orthogonal zur z-Achse steht, verschwindet das Integral, und so können diese Elemente bei der Berechnung der Summe weggelassen werden. Nach Konstruktion steht der gesamte Abschluss orthogonal auf der z-Achse. Daher muss der Abschluss nicht vernetzt werden.
Für ebene Dreiecke kann dieses Integral exakt berechnet werden. Bei der Berechnung der Summe ist die Richtung des Normalvektors zu beachten, da dieser im Integralsatz von Gauss immer nach außen zeigen muss. Betrachtet man ein Oberflächennetz, so sind somit bei benachbarten Dreiecken die Vorzeichen dann gleich, wenn die Kante zwischen den Dreiecken keine Schnittkante ist. Entlang einer Schnittkante dreht sich die Richtung des Normalvektors um. Die Schnittkanten müssen also exakt bestimmt werden.

Zur Berechnung des Volumens müssen zuerst die Integrale aller Dreiecke berechnet werden, und im Anschluss müssen diese Integrale mit richtigem Vorzeichen aufsummiert werden. Der Algorithmus wird im folgenden Abschnitt etwas genauer beschrieben.

2.3.1 Aufsummierung über die Dreiecke

Zuerst startet man an einer Schnittkante und sucht die vier passenden Elemente zu dieser Kante. In diesen Dreiecken kann die Richtungen der Normalvektoren bestimmt werden.

Ein Beispiel für diese Bestimmung sieht man in Abbildung 3. Ausgehend von diesen Elementen werden nun die Vorzeichen der Nachbarelemente durch Übergang über die Kanten bestimmt. Dabei können drei Fälle auftreten:

- Falls die Kante eine Schnittkante ist, ändert sich die Richtung des Normalvektors.
- Im Falle einer Kante, die durch die Bildung des Abschlusses entstanden ist, muss gestoppt werden.
- Falls die Kante weder eine Schnitt- noch eine Abschlusskante ist, bleibt die Richtung des Normalvektors gleich.

Sind alle Elemente berücksichtigt worden, ist das Volumen komplett berechnet.

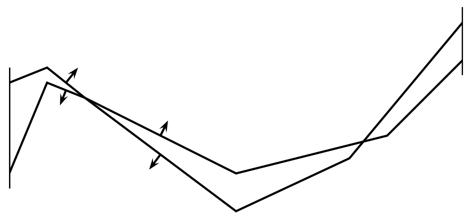


Abbildung 3: Zwei Oberflächenetze mit Normalvektoren

2.4 Bestimmung der Schnitte der Netze

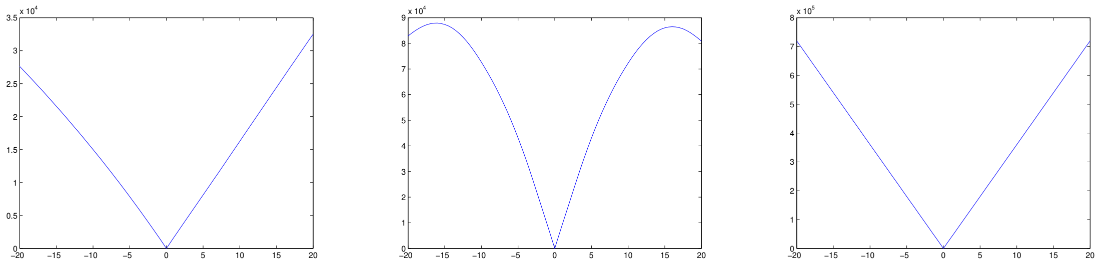
Um das Volumen berechnen zu können, müssen beide Netze miteinander geschnitten werden. Wenn zwei Elemente geschnitten werden, wird zuerst die Schnittkante ermittelt und im Anschluss werden beide Dreiecke neu vernetzt. Dreiecke zerfallen dabei in Unterdreiecke. Wird das Dreieck noch von anderen Dreiecken geschnitten, muss mit allen Unterdreiecken getestet und möglicherweise auch geschnitten werden.
Alle bisher beschriebenen Teile des Algorithmus können in einer Laufzeit von $O(N_1 + N_2)$ realisiert werden, wobei N_i die Anzahl der Dreiecke des i-ten Netzes bezeichnet. Dies ist beim Schneiden nicht ohne weiteres mehr möglich, da jedes Element des ersten Netzes mit jedem Element des zweiten Netzes auf einen Schnitt getestet werden muss. Da der eigentliche Schneidealgorithmus äußerst komplex und somit auch rechenzeitaufwändig ist, ist eine effiziente Auswahlregel notwendig.
Jedes Element wir dazu in eine Kugel eingeschlossen. Anstatt die Dreiecke auf Schnitt zu testen, testet man zuerst die zugehörigen Kugeln auf Schnitt. Dazu muss überprüft werden, ob der Abstand der beiden Mittelpunkte kleiner ist als die Summe der Radien der beiden

Kugeln. Falls die Bedingung erfüllt ist, wird der Schneidealgorithmus auf diese beiden Dreiecke angewandt. Somit muss der eigentliche Schneidealgorithmus pro Element nur mehr auf eine beschränkte Anzahl von Elementen angewendet werden, was zu einer enormen Laufzeitverbesserung führt. In der Vorauswahl bleibt die Abhängigkeit von $O(N_1 N_2)$ jedoch erhalten. Nur durch Einführung einer geeigneten hierarchischen Struktur könnte eine fast lineare Laufzeit realisiert werden.

3 Minimierung

3.1 Funktion in der Umgebung des Minimums

Um einen geeigneten Algorithmus zur Lösung dieses Problems zu finden, wird zunächst die Funktion in der Umgebung des Minimums entlang der sechs Freiheitsgrade betrachtet. Es wurde das Beispiel aus Abschnitt 4.3 verwendet. In diesem Beispiel wird zweimal dasselbe Oberflächennetz verwendet. Dadurch ist das Minimum exakt bekannt.



(a) Translation in x (b) Translation in y (c) Translation in z

Abbildung 4: Funktion in Umgebung des Minimums

Wie sich in der Abbildung 4 zeigt, ist die Funktion weder im Minimum differenzierbar noch in dessen Umgebung konvex. Die Freiheitsgrade der Rotation zeigen ein nahezu identisches Verhalten, weshalb die Funktionen in die Richtungen dieser Freiheitsgrade nicht dargestellt werden. Obwohl es scheint, dass die Funktion nur ein Minimum hat, haben Beispiele gezeigt, dass dies nicht immer der Fall ist, sondern sehr wohl auch mehrere lokale Minima auftreten können. Bei üblicherweise verwendeten Verfahren besteht außerdem noch ein viel schwerwiegenderes Problem. Üblicherweise wird ein Minimum gesucht, indem die Gleichung $\nabla f(x) = 0$ gelöst wird. Diese Verfahren brechen ab, falls diese Gleichung genügend gut gelöst ist, also die Norm des Gradienten klein genug ist. Bei dieser Funktion wird der Gradient jedoch nie Null. Somit können Verfahren wie das Newton-Verfahren oder Varianten des Newton-Verfahrens, siehe beispielsweise [1, 2], also nicht ohne Modifikationen angewandt werden. Ein Verfahren, das verwendet werden kann, ist das Gradientenverfahren, siehe Algorithmus 1.

Algorithm 1 Gradientenverfahren

Wähle $\sigma \in (0,1), \beta \in (0,1), \varepsilon \geq 0$
while $\|\nabla f(\underline{0})\| > \varepsilon$ **do**
$d := -\nabla f(\underline{0})$ {Mit $f()$ wird die Funktion des Volumens bezeichnet.}
$t := \max\{\beta^\ell | \ell = 0, 1, 2, \cdots\}$ mit

$$f(t\underline{d}) \leq f(\underline{0}) + \sigma t \nabla f(\underline{x})^T d$$

move($t\underline{d}^k$) {Es wird das zweite Oberflächennetz verschoben, wodurch sich der Nullpunkt ändert.}
end while

3.2 Modifikationen des Gradientenverfahren

Das Gradientenverfahren kann jedoch nicht direkt angewendet werden. Es stellen sich einige Probleme:

1. Der Gradient ist nicht analytisch berechenbar.
2. Die Startsuchlänge $t_k = 1$ kann außerhalb des erlaubten Bereiches liegen.
3. Die Schrittweitenstrategie ist zufällig, und es wird nicht das Minimum entlang des Gradienten genommen. Dies sorgt für viele zusätzliche Iterationsschritte.
4. Das übliche Abbruchkriterium wird nie erreicht.

Um nun diese Probleme zu beseitigen, sind einige Modifikationen notwendig:

1. Der Gradient wird numerisch angenähert.
2. Die maximale Suchlänge wird eingeschränkt.
3. Die optimale Schrittweite wird durch Intervallschachtelung bestimmt.
4. Falls die Norm der Änderung der Verschiebung kleiner als ein vorgegebenes ε ist, wird das Gradientenverfahren abgebrochen.

Selbst durch diese Modifikationen konvergiert das Gradientenverfahren nicht immer. Es sind weitere Modifikationen notwendig.

Weitere Modifikationen: Skalierung und Elimination

Man stelle sich als einfaches Beispiel zwei Ebenen vor, die aufeinander gelegt werden sollen. Betrachtet man die sechs Freiheitsgrade, fällt auf, dass die Volumensänderung durch Translation um einen vorgegebenen Wert viel kleiner ist als ein Volumensänderung durch eine Rotation um denselben Wert. Dies muss ausgeglichen werden, da sonst immer versucht wird in der dominierenden Richtung zu optimieren.

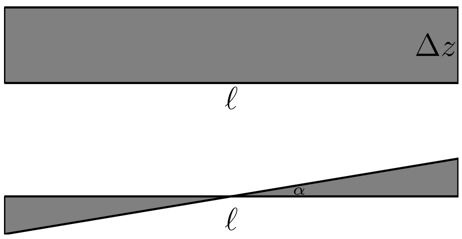


Abbildung 5: Vergleich der Volumensänderung

Wie aus Abbildung 5 ersichtlich ist, muss für den Winkel $\alpha = \arctan(4\ \Delta z/\ell)$ gelten, um dieselbe Volumensänderung zu erreichen. Somit kann die Dominanz der Rotation behoben werden.
Jedoch können weiterhin gewisse Freiheitsgrade über andere dominieren. Betrachtet man Abbildung 4, so wird ersichtlich, dass der Gradient in diesem Beispiel in Richtung der Translation in z etwa 10-mal größer ist, als in Richtung der beiden anderen Translationen. Diese Unterschiede lassen sich aber nicht mehr so einfach beheben. Daher wird versucht das Gradientenverfahren entsprechend anzupassen.

Wie bereits bekannt wird der Gradient nie zu Null. Das bedeutet, das bereits optimierte Richtungen nicht aus dem Gradienten verschwinden. Das Gradientenverfahren versucht das Volumen in einer bereits optimierten Richtung zu verbessern. Da es in diese Richtung jedoch kaum noch Verbesserungsmöglichkeiten gibt, bricht der Algorithmus laut Abbruchkriterium ab, obwohl nur die Hauptrichtung optimiert worden ist. Dies kann umgangen werden, indem die Hauptrichtung aus dem Gradienten eliminiert wird, falls sich das Volumen nicht mehr genügend ändert. Es wird der betragsmäßig größte Wert im Gradienten gleich Null gesetzt. Dies kann so lange geschehen, bis der Gradient gleich Null ist. Hat der Algorithmus fünfmal eliminiert und das Volumen dabei nicht verkleinern können, bricht der Algorithmus ab. In numerischen Beispielen konnte so Konvergenz erreicht werden.

3.3 Weitere Methoden

Es wurden neben dem Gradientenverfahren noch weitere Methoden getestet:

- Es wurde die "Sequential Response Surface Method" aus dem Programm Hyperstudy verwendet, siehe beispielsweise [3].
- Durch eine Quadratur der Funktionswerte konnte das Newtonverfahren angewendet werden. Auf dieselbe Weise wie das Gradientenverfahren, wurde auch das Newtonverfahren modifiziert. Ohne diese Modifikationen konnten keine Konvergenz festgestellt werden.

4 Numerische Beispiele

4.1 Sequential Response Surface Method

Es wurde die "Sequential Response Surface Method", die im Programm Hyperstudy zur Verfügung stand, getestet. Zum Test wurden die beiden Oberflächennetze aus Abbildung 1 verwendet.

Schritt	Volumen	Schritt	Volumen	Schritt	Volumen
1	1435010	7	3140180	60	342560
2	3306090	8	1427780	61	276610
3	1464740	9	1107880	75	262525
4	2467870	20	982583	80	367154
5	2931080	40	786267	90	291644
6	1139020	50	682200	100	278872

Tabelle 1: Sequential Response Surface Method

Zu Beginn der Optimierung springen die Zielfunktionswerte des Algorithmus, siehe Tabelle 1. Ab der zehnten Iteration etwa beginnt er jedoch stetig in kleinen Schritten das Volumen zu minimieren. Diese Minimierung endet jedoch im Schritt 61. Ab diesem Schritt bewegt sich der Wert des Volumens nur mehr zwischen ca. 250000 und 350000. Da der Optimalwert jedoch bei ca. 95000 liegt, ist die Ungenauigkeit doch sehr hoch. Auch durch Erhöhung der programminternen Genauigkeit konnte kein besseres Ergebnis erzielt werden. Auch andere Beispiele zeigen bei diesem Optimierungsverfahren ein ähnliches Verhalten.

4.2 Frühzeitige Elimination

In diesem Beispiel wurden zwei Ebenen mit verschiedenen Vernetzungen verwendet. In der Ausgangslage ist eine der beiden Ebenen leicht verdreht, wie in Abbildung 6 zu sehen ist.

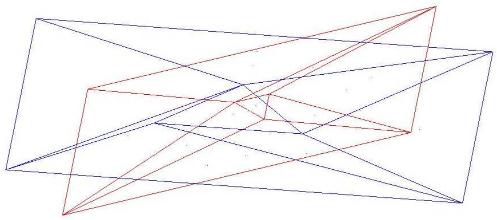


Abbildung 6: Zwei Ebenen

Der Minimalwert ist somit bekannt und beträgt Null. Dieses Beispiel zeigt, wie wichtig Eliminieren der Hauptrichtung nicht nur für das Bestimmen eines Minimums ist, sondern auch wie dramatisch die Elimination die Konvergenzgeschwindigkeit erhöhen kann. Bei der ersten Berechnung, siehe Tabelle 2 links, wird erst eliminiert, sobald ein mögliches Minimum erreicht wird. Im zweiten Anlauf, siehe Tabelle 2 rechts, wird bereits eliminiert, sobald sich das Volumen in einem Schritt nicht genügend verkleinert.

Schritt	Volumen	Schritt	Volumen
1	260.3	7	61
2	259.4	8	44
3	175	9	39
4	131	10	33
5	99		
6	76	40	0.28

Schritt	Volumen
1	260.3
2	96
3	0.74
4	0.06
5	0.06
6	0.001

Tabelle 2: Zwei Ebenen ohne und mit frühzeitiger Elimination

Es zeigt sich eine dramatische Beschleunigung der Iterationszahlen des Gradientenverfahren von 40 auf 6 durch das frühzeitige Eliminieren. Auch in sämtlichen anderen Beispielen zeigt sich ein schnellere Konvergenz durch frühzeitiges Eliminieren, jedoch kann nicht immer eine derart große Verbesserung festgestellt werden.

4.3 Beispiel mit bekanntem Minimum

In folgendem Beispiel wurde zweimal dasselbe Oberflächennetz aus Abbildung 7 verwendet. Zur Vorpositionierung werden verschiedene Punkte verwendet, wodurch die beiden Netze in der Ausgangslage leicht verschoben und verdreht zueinander liegen. Da dasselbe Netz zweimal verwendet wird, beträgt der Minimalwert Null. In diesem Beispiel wurde mit frühzeitiger Elimination gerechnet.

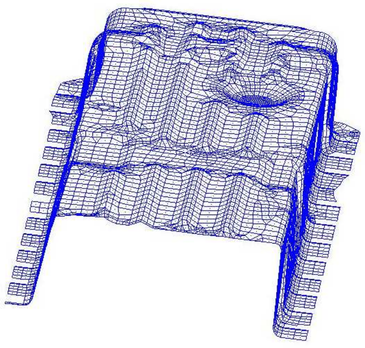


Abbildung 7: Oberflächennetz

Schritt	Volumen	Schritt	Volumen	Schritt	Volumen
1	81260	7	18760	30	268
2	61260	8	12432	31	256
3	39104	9	10753	36	234
4	29515	19	670	40	230
5	23979	20	491	41	229
6	19587	21	414	50	229

Tabelle 3: Beispiel mit bekanntem Minimum

Aufgrund der komplexeren Geometrie als in Abschnitt 4.2 erhöht sich in diesem Beispiel die Iterationszahl, vergleiche Tabelle 3. Allerdings könnte aber schon ab Iteration 36 abgebrochen werden, da sich im Anschluss der Aufwand für die relativ geringe Verbesserung nicht mehr rechtfertigen lässt. Der erreichte Minimalwert scheint noch etwas weit vom Minimum Null entfernt zu sein. Allerdings besitzt die Geometrie eine Oberfläche von ca. 20000 und somit liegt der durchschnittliche Abstand zwischen den beiden Oberflächen bei ca. 0.01.

4.4 Anwendungsbeispiel

In diesem Beispiel wurden die zwei verschiedenen Oberflächennetze aus Abbildung 1 betrachtet.

Schritt	Volumen	Schritt	Volumen	Schritt	Volumen
1	432661	7	107425	13	95531
2	255867	8	100334	14	95223
3	192096	9	99375	15	94985
4	179654	10	96530	16	94483
5	148308	11	96086	17	93832
6	117258	12	95799		

Tabelle 4: Anwendungsbeispiel

Hier scheint die Funktion etwas glatter zu sein. Besonders in den ersten 3 Schritten konvergiert das Verfahren schnell und bereits nach 8 Schritten ist der Algorithmus in der Nähe des Minimums, siehe Tabelle 4. In diesem Beispiel beträgt die Ausdehnung der Geometrie ca. 100000, wodurch nach 8 Schritten ein durchschnittlicher Abstand von ca. 0.1 erreicht wird. In der Nähe des Minimums verlangsamt sich die Konvergenz dramatisch. In diesem Beispiel ist das minimale Volumen nicht Null, da die beiden verwendeten Netze nicht genau aufeinander passen.

4.5 Newton-Verfahren

Um die Eigenschaften der zu minimierenden Funktion zu verbessern, wurde das Quadrat des Volumens minimiert. Somit konnte ein modifiziertes Newton-Verfahren eingesetzt werden. Als Test wurden die beiden Netze aus Abbildung 1 verwendet. Die Werte aus Tabelle 5 zeigen eine höhere Konvergenzrate. Allerdings wird nur ein lokales Minimum in der Nähe des Minimums, das mit Hilfe des Gradientenverfahrens bestimmt wurde, siehe Abschnitt 4.4, gefunden.

Schritt	Volumen	Schritt	Volumen
1	432661	7	157173
2	210012	8	153229
3	187317		
4	179654		
5	166318		
6	159510		

Tabelle 5: Anwendungsbeispiel

Zusammenfassung und Ausblick

Es wurde ein Verfahren zur optimalen Positionierung zweier Oberflächennetze entwickelt und getestet. Dazu wurde das zwischen den Oberflächen eingeschlossene Volumen minimiert. Aufgrund der Eigenschaften des zu minimierenden Funktionals wurden verschiedene Verfahren getestet und geeignet angepasst. Mit Hilfe eines angepassten Gradientenverfahrens konnte in allen Test- und Anwendungsbeispielen eine Minimierung erreicht werden, während beispielsweise die "Sequential Response Surface Method" zu keinem zufriedenstellenden Ergebnis führte.
Die noch erhöhten Rechenzeiten könnten sich mit Hilfe einer hierarchischen Strategie zur Bestimmung der Schnittelemente deutlich reduzieren lassen.

Acknowledgement

Wir danken Herrn Dipl.-Ing. Dr. Christoph Lind für die gute Zusammenarbeit und dem Kompetenzzentrum "Das virtuelle Fahrzeug" für die finanzielle Unterstützung.

Literatur

[1] Geiger, C. und Kanzow, C.: *Theorie und Numerik restringierter Optimierungsaufgaben*, Berlin: Springer, 2002.

[2] Geiger, C. und Kanzow, C.: *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*, Berlin: Springer, 1999.

[3] Myers, R.H. und Montgomery, D.C.: *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*, New York: John Wiley & Sons, Inc., 1995.

Erschienene Preprints ab Nummer 2005/1

2005/1	O. Steinbach	Numerische Mathematik 1. Vorlesungsskript.
2005/2	O. Steinbach	Technische Numerik. Vorlesungsskript.
2005/3	U. Langer G. Of O. Steinbach W. Zulehner	Inexact Fast Multipole Boundary Element Tearing and Interconnecting Methods
2005/4	U. Langer G. Of O. Steinbach W. Zulehner	Inexact Data–Sparse Boundary Element Tearing and Interconnecting Methods
2005/5	U. Langer O. Steinbach W. L. Wendland	Fast Boundary Element Methods in Industrial Applications Söllerhaus Workshop, 25.–28.9.2005, Book of Abstracts.
2005/6	U. Langer A. Pohoata O. Steinbach	Dual–Primal Boundary Element Tearing and Interconnecting Methods
2005/7	O. Steinbach (ed.)	Jahresbericht 2004/2005
2006/1	S. Engleder O. Steinbach	Modified Boundary Integral Formulations for the Helmholtz Equation
2006/2	O. Steinbach (ed.)	2nd Austrian Numerical Analysis Day, Book of Abstracts
2006/3	B. Muth G. Of P. Eberhard O. Steinbach	Collision Detection for Complicated Polyhedra Using the Fast Multipole Method or Ray Crossing
2006/4	G. Of, B. Schneider	Numerical Tests for the Recovery of the Gravity Field by Fast Boundary Element Methods
2006/5	U. Langer O. Steinbach W. L. Wendland	4th Workshop on Fast Boundary Element Methods in Industrial Applications, Book of Abstracts
2006/6	O. Steinbach (ed.)	Jahresbericht 2005/2006
2006/7	G. Of	The All-floating BETI Method: Numerical Results