Technische Universität Chemnitz-Zwickau

DFG-Forschergruppe „SPC“ · Fakultät für Mathematik

Ulrich Groh

Ein technologisches Konzept zur Erzeugung adaptiver hierarchischer Netze für FEM-Schemata

Preprint-Reihe der Chemnitzer DFG-Forschergruppe „Scientific Parallel Computing“

SPC 95_6 **Mai 1995**

Summary:

Adaptive finite element methods for the solution of partial differential equations require effective methods of mesh refinement and coarsening, fast multilevel solvers for the systems of FE equations need a hierarchical structure of the grid.
In the paper a technology is presented for the application of irregular hierarchical triangular meshes arising from refinement by only dividing elements into four congruent triangles. The paper describes the necessary data structures and data structure management, the principles and algorithms of refining and coarsening the mesh, and also a specific assembly technique for the FE equations system. Aspects of the parallel implementation on MIMD computers with a message passing communication are included.

Zusammenfassung:

Adaptive Finite-Elemente-Methoden für die Lösung partieller Differentalgleichungen erfordern effektive Techniken der Netzverfeinerung und -vergröberung, schnelle Multilevel-Lösungsmethoden für die FEM-Gleichungssysteme benötigen eine hierarchische Netzstruktur.
In der Arbeit wird eine Technologie für die Anwendung irregulärer hierarchischer Dreiecksnetze vorgestellt, die bei Verfeinerung nur durch Elementzerlegung in vier kongruente Dreiecke entstehen. Die Arbeit beschreibt die dafür notwendigen Datenstrukturen und deren Verwaltung, die Prinzipien und Algorithmen von Netzverfeinerung und -vergröberung sowie eine spezifische Assemblierungstechnik für das FEM-Gleichungssystem. Fragen der parallelen Implementation auf MIMD-Computern mit Botschaftenaustausch werden einbezogen.

Ulrich Groh
Technische Universität Chemnitz–Zwickau
Fakultät für Mathematik
D – 09107 Chemnitz

Tel.: +49 - 0371 - 531 - 4685
e-mail: `ugroh@imech.tu-chemnitz.de`

Ein technologisches Konzept zur Erzeugung adaptiver hierarchischer Netze für FEM-Schemata

Inhaltsverzeichnis

1 Einleitung

Beim Auftreten von Grenzschichten und Singularitäten der Lösung, insbesondere wenn sich diese im Rahmen instationärer Probleme noch mit der Zeit bewegen, sind adaptive Netzgeneratoren für die FEM wünschenswert. In Kombination mit Fehlerschätzern und mit einer geeigneten Fehlerreduzierungsstrategie kann dann durch schrittweise bessere Anpassung des Netzes an die Lösung der Fehler der FEM-Näherungslösung bis unter eine vorgegebene Schranke verringert werden, vgl. zum Beispiel LEINEN [5] oder RÜDE [7].

Werden moderne Multilevel-Vorkonditionierungen für die schnelle iterative Auflösung der entsprechenden FEM-Gleichungssysteme eingesetzt, zum Beispiel die hierarchische Vorkonditionierung nach YSERENTANT [9], die BPX-Vorkonditionierung [1] oder auch Multigrid-Verfahren, so ist überdies eine hierarchische Netzstruktur notwendig. Bei Erzeugung derselben durch einfachste lokale Verfeinerungstechniken (wie die Zerlegung eines Dreieckselementes in vier kongruente Teildreiecke) können irreguläre Knoten (hanging nodes) und damit irreguläre Netze entstehen. Entsprechende Programme benutzen deshalb oft nur gleichmäßig über einem Grobnetz erzeugte Netzhierarchien oder ungleichmäßige hierarchische Netzstrukturen, bei denen durch zusätzliche Abschließungsoperationen der durch die irregulären Knoten gestörte reguläre Charakter des Netzes wiederhergestellt wird, wie etwa auch in [5], [7].

Als Alternative dazu kann eine Vorgehensweise benutzt werden, die das Vorhandensein von irregulären Knoten im Netz als „normal" akzeptiert. Der Erzeugung einer Hierarchie von Basisfunktionen wird die Priorität eingeräumt vor der Erzeugung eines regulären Netzes, bei dem Elemente nur einzelne Netzknoten oder ganze Kanten gemeinsam haben. Auch müssen Knoten des Netzes nicht unbedingt Basisfunktionen zugeordnet werden. Die hierarchische Struktur der FEM-Basis wird durch – möglicherweise ungleichmäßige – hierarchische Verfeinerung der Kanten eines regulären Netzes erzeugt und kann zur Adaption der Lösung genutzt werden. Die Entartung des Netzes durch irreguläre Knoten wird nicht als pathologischer Fall angesehen, für den im Rahmen der klassischen Technik Sondermaßnahmen ergriffen werden müssen, sondern als der Normalfall ungleichmäßiger Hierarchien. Dieser alternative Weg wird aus Sicht der FEM in der Literatur nur wenig behandelt.

Gegenstand der vorliegenden Arbeit ist deshalb eine FEM-Technologie der Nutzung irregulärer hierarchischer Dreiecksnetze im adaptiven Lösungsprozeß. Diese beinhaltet den Aufbau und die dynamische Verwaltung der Datenstrukturen des Netzes, die Algorithmen des Netzumbaus bei schrittweiser Anpassung des Netzes an die Lösung sowie die sich von der üblichen Assemblierung etwas unterscheidende Technik zur Erzeugung des FEM-Gleichungssystems.

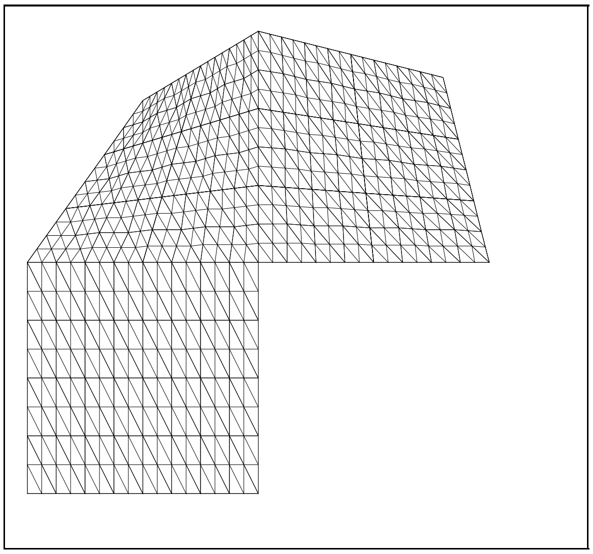
Die Untersuchungen erfolgen unter der Voraussetzung, daß beim Netzumbau bekannt ist, wo eine Verfeinerung oder Vergröberung des Netzes erfolgen soll. Die Problematik der Fehlerschätzer und Fehlersteuerung wird in dieser Arbeit somit ausgeklammert, wobei die Verwendung der hierarchischen Fehlerschätzer von VERFÜRTH [8] aus Sicht der hier betrachteten Datenstrukturen und Algorithmen naheliegend und deshalb im weiteren beabsichtigt ist. Im folgenden werden nur Dreieckselemente betrachtet, was aber keine Einschränkung von prinzipieller Bedeutung darstellt.

Im Hintergrund stehend, wird auch die Parallelisierung der Algorithmen auf einem MIMD-Parallelrechner mit verteiltem Speicher und Kommunikation über Botschaftenaustausch ins Auge gefaßt. Für die Parallelisierung durch Datenaufteilung mittels nichtüberlappender Gebietszerlegung, vgl. etwa [6], [4], [2], [3], ergeben sich keine wesentlich neuen Aspekte. Entsprechende Gesichtspunkte der Realisierung sind unmittelbar in den Text eingearbeitet.

Die Darstellungen beziehen sich im wesentlichen auf elliptische Probleme zweiter Ordnung und sind auf entsprechende instationäre Aufgaben übertragbar. An einigen Stellen sind jedoch auch Hinweise für die Anwendung bei Strömungsproblemen eingearbeitet, d.h. die Behandlung des Paares Geschwindigkeit – Druck.

2 Grundkonzept

Ausgangspunkt und damit Hierarchielevel 0 ist ein reguläres Netz, mit dem bereits die Geometrie des Problemgebietes erfaßt werde und das im Falle der Nutzung eines MIMD-Parallelrechners schon auf die Prozessoren aufgeteilt sei. Dieses Netz werde als *Hauptnetz* bezeichnet. Es stellt bei allen Netzveränderungen den *stationären Netzanteil* dar, Verfeinerungen werden vom Hauptnetz aus hierarchisch durchgeführt, Vergröberungen können nicht weiter als bis zum Level des Hauptnetzes gehen. Die auf das Hauptnetz aufgesetzten, lokal möglicherweise unterschiedlich tief gestaffelten Hierarchielevel bilden als *Zusatznetz* den *dynamischen Netzanteil*, der adaptiv gesteuert werden kann.



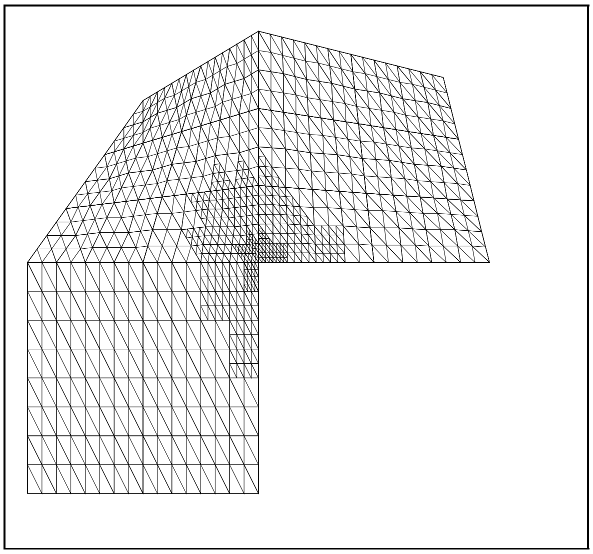


Abbildung 1: Hauptnetz (Level 0) und Hauptnetz mit 2 Levels Zusatznetz

Das Hauptnetz selbst kann mittels eines anderen Netzgenerators aus einem *Urnetz* erzeugt und parallelisiert worden sein. Ist dabei bereits eine kantenorientierte Knotenhierarchie entstanden, wird diese im weiteren für die hierarchische Vorkonditionierung mit in die entsprechende Datenstruktur eingebaut.

Die Grundmoduln der Verfeinerung bzw. Vergröberung des Netzes realisieren jeweils einen Lauf über sämtliche Elemente, bei dem lokal die Hierarchie maximal um ein Level auf- bzw. abgebaut werden kann.

Lokale Verfeinerung und Vergröberung des Netzes werden über *kantenorientierte Kriterien* gesteuert. Es wird jeweils entschieden, ob auf einer *aktivierbaren* (d.h. zulässigen) *Kante* eine neue, dem Kantenmittelpunkt zugeordnete Basisfunktion eingeführt werden soll bzw. ob umgekehrt eine solche Entscheidung wieder rückgängig zu machen ist. Aktivierbarkeit einer Kante bedeutet dabei, daß ihren beiden Endpunkten bereits Basisfunktionen zugeordnet wurden (die üblichen Hütchenfunktionen bei der Verwendung linearer Elemente). Die neue Basisfunktion wird aus dem nächsthöheren Hierarchielevel mit entsprechend kleinerem Träger gewählt. Die alleinige Verwendung aktivierbarer Kanten für den Ausbau der Basis ermöglicht die Anwendbarkeit kantenorientierter Kriterien sowie eine einfache levelweise faktorisierte Transformation aus der hierarchischen Basis in die FEM-Basis.

Bei Einführung der neuen Basisfunktion wird aus dem Kantenmittelpunkt ein Netzknoten mit zugeordneter Basisfunktion (*aktiver Knoten*), aus beiden Teilkanten werden deshalb wieder aktivierbare Kanten. Diese können in späteren Läufen durch aktive Knoten weiter unterteilt werden. Wird aus dem Kantenmittelpunkt der aktivierbaren Kante bei der aktuellen Verfeinerung kein aktiver Knoten, weil das Verfeinerungskriterium nicht erfüllt ist, kann die Verfeinerung auf dieser Kante bei einem späteren Verfeinerungslauf erneut versucht werden, falls die Kante dann noch aktivierbar ist.

Von den Dreiecken werden genau diejenigen verfeinert, auf deren Kanten mindestens eine neue Basisfunktion eingeführt wurde. Sie werden durch Verbindung der Kantenmittelpunkte in vier kongruente Teildreiecke zerlegt. Dabei können auch Eckknoten entstehen, die keine Basisfunktion tragen (*passive Knoten*). Bei mehreren Verfeinerungsläufen über alle Elemente ergeben sich somit im allgemeinen ungleichmäßige Elemente- und Kantenhierarchien.

3 Die Hierarchien und ihre Datenstrukturen

3.1 Die Elementehierarchie

Die primär erzeugte Hierarchie ist die Hierarchie der Elemente, hier als Dreiecke angenommen. Jedes Element des Hauptnetzes erzeugt bei hierarchischer Verfeinerung einen Elementebaum. Bei dynamischer Adaptivität muß dieser auf- und abbaubar sein und deshalb als doppelt verkettete Liste gespeichert werden, bezeichnet als hierarchische Elementeliste **HiEl**. Jede Zeile von **HiEl** repräsentiert ein Element. Außer den *Hauptelementen*, den Elementen des Hauptnetzes, besitzt jedes Element genau einen Vater und, wenn überhaupt vorhanden, vier Söhne. Die Söhne werden entsprechend dem Schema von Abb.1 geordnet.

Die Blätter der Elementebäume werden als *aktuelle Elemente* bezeichnet und in der Liste der aktuellen Elemente **Elem** genauer beschrieben. Über diese Liste laufen Verfeinerungs- und Vergröberungszyklen des Netzes sowie die Assemblierung der Steifigkeitsmatrix. Im Detail sind die Listen wie folgt aufgebaut.

Die hierarchische Elementeliste HiEl

Dimension	**HiEl**(5,NELX)
Bedeutung des Zeilenindex IEL	Elementnummer
Zeilenindex IEL=NELX	maximal mögliche Elementnummer
Anordnung der Listenelemente	S1 | S2 | S3 | S4 | V in jeder Zeile
Bedeutung von S1	Elementnummer des Sohnes S1
Bedeutung von S2	Elementnummer des Sohnes S2
Bedeutung von S3	Elementnummer des Sohnes S3
Bedeutung von S4	Elementnummer des Sohnes S4
Bedeutung von V	Elementnummer des Vaters
Besonderheiten:	
S1=0, S4=I>0	aktuelles Element, hat Index I in **Elem**
S4=0, S1=PT>0	Lücke, die Zeile repräsentiert kein Element, PT ist Zeiger auf vorhergehende Lücke
V=0	Hauptelement, kein Vater
Anzahl NELH	Anzahl der Hauptelemente
IEL≤NELH	Zeilen der Hauptelemente

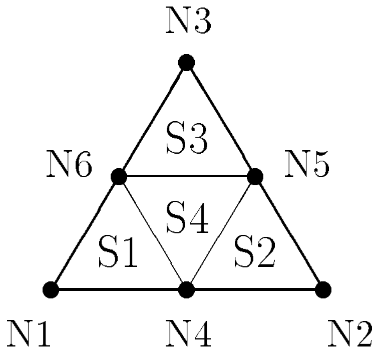


Abbildung 2: Numerierungsschema für die Sohn-Elemente und Knoten der Dreiecke

Die Elemente behalten ihre Nummer, d.h. ihren Platz in **HiEl**, während der gesamten Zeit ihrer Existenz. Bei Rückbau der Hierarchie entstehen Lücken. Diese werden als Teilliste von **HiEl** mit einfacher Verkettung über einen Pointer verwaltet. Die Hauptelemente stehen lückenlos auf den ersten NELH Zeilen des Feldes.

Die Liste der aktuellen Elemente Elem

Eine genauere Beschreibung der Elemente erfolgt nur für die jeweils aktuellen Elemente in der Liste **Elem**. Die Beschreibung besteht dann aus der für Netzgenerierung, Netzumbau und Matrixassemblierung notwendigen Element-Information ElInfo, die zum Beispiel die zugehörigen Knotennummern, Kantennummern und den Elementtyp enthalten kann, und der Elementnummer IEL als Verweis auf den Platz in der hierarchischen Elementeliste **HiEl**.

Die Element-Information der Söhne muß sich aus der Element-Information des Vaterelementes erzeugen lassen, umgekehrt muß die Element-Information des Vaters bei Rückbau der Hierarchie aus der Element-Information der Söhne rekonstruierbar sein, und zwar möglichst einfach, also ohne aufwendige Suchprozesse. Im Falle der hier benutzten Dreieckselemente bestehe die Element-Information aus den Knotennummern der drei Eckknoten und der drei Kantenmittelknoten, die auch als Kantennummern (s.u.) verstanden werden können. Bei Rückbau der Hierarchie werden Lücken in **Elem** sofort mit der Eintragung des Vaters und verlagerten Eintragungen vom Ende der Liste aufgefüllt.

allgemein:	
Dimension	**Elem**(dim(ElInfo)+1,NEL)
Bedeutung des Zeilenindex I	Listenplatz des aktuellen Elementes
Anzahl NEL	Gesamtzahl aktueller Elemente
Anordnung der Listenelemente	$\boxed{\text{ElInfo}}\boxed{\text{IEL}}$ in jeder Zeile
Bedeutung von ElInfo	Element-Information des aktuellen Elementes IEL
Bedeutung von IEL	Elementnummer des aktuellen Elementes
speziell:	ElInfo = $\boxed{\text{N1}}\boxed{\text{N2}}\boxed{\text{N3}}\boxed{\text{N4}}\boxed{\text{N5}}\boxed{\text{N6}}$
Dimension	**Elem**(7,NEL)
Bedeutung des Zeilenindex I	Listenplatz des aktuellen Dreiecks
Anzahl NEL	Gesamtzahl aktueller Dreiecke
Anordnung der Listenelemente	$\boxed{\text{N1}}\boxed{\text{N2}}\boxed{\text{N3}}\boxed{\text{N4}}\boxed{\text{N5}}\boxed{\text{N6}}\boxed{\text{IEL}}$ in jeder Zeile
Bedeutung von N1, N2, N3	Nummern der Eckknoten von Dreieck IEL
Bedeutung von N4, N5, N6	Nummern der Kantenmittelknoten von Dreieck IEL
Bedeutung von IEL	Elementnummer des aktuellen Dreiecks

Erzeugung und Rekonstruktion der Element-Information

Die Element-Information wird nur für die aktuellen Elemente gespeichert. Daraus sind bei Erweiterung der Hierarchie die Element-Informationen der Söhne zu erzeugen bzw. es ist bei Rückbau der Hierarchie die Element-Information des Vaters zu rekonstruieren.

Für die angegebenen 6-Knoten-Dreieckselemente werde nachfolgendes Schema zur Erzeugung der Element-Informationen der Söhne benutzt. Wenn das Vater-Element V zerlegt wird und seine Knoten sowie seine vier Söhne wie in Abb. 2 numeriert sind, dann werden bei den Söhnen die Knoten dem Schema entsprechend eingetragen. Die Nij bezeichnen dabei jeweils die Kantenmittelpunkte zwischen Ni und Nj, die als Knoten neu zu generieren sind.

Elem(1,.)	Elem(2,.)	Elem(3,.)	Elem(4,.)	Elem(5,.)	Elem(6,.)	Elem(7,.)
Vater:						
N1	N2	N3	N4	N5	N6	V
Söhne:						
N1	N4	N6	N14	N46	N61	S1
N2	N5	N4	N25	N54	N42	S2
N3	N6	N5	N36	N65	N53	S3
N4	N5	N6	N54	N65	N46	S4

Aus diesem Schema der erzeugten Element-Information der Söhne läßt sich umgekehrt sofort die Element-Information des Vaters ablesen. Dies ist die Liste der Knotennummern N1 bis N6, wenn nur die Söhne nach dem Schema von Abb. 2 numeriert und entsprechend in **HiEl** eingetragen wurden.

3.2 Die Knoten-/Kantenhierarchie

Aus der Elementehierarchie wird eine Knoten- bzw. Kantenhierarchie erzeugt. Bei der Verfeinerung eines Dreieckselementes in vier Teildreiecke werden einerseits die Seitenkanten des Dreiecks durch die Kantenmittelpunkte jeweils in zwei Teilkanten zerlegt, die bestehende Kantenbäume fortsetzen. Wenn überhaupt, besitzt eine Kante also genau zwei Söhne bzw. einen Vater. Andererseits entstehen als Verbindung von jeweils zwei Kantenmittelpunkten im Dreieck drei Wurzeln von neuen Kantenbäumen.

Wird der Knoten N in der Kantenmitte mit der Kante selbst identifiziert, ergibt sich eine weitgehende Isomorphie der Knoten- und der Kantenstruktur. Bei gleicher Numerierung sich entsprechender Knoten und Kanten ist durch die Kantenteilung die Nummer der Vaterkante, soweit vorhanden, immer mit der Knotennummer eines der beiden Endknoten identisch, der als Vaterknoten V1 des Knotens N bezeichnet werde. Die Kante kann aber auch durch beide Endknoten beschrieben werden, wie es für die hierarchische Vorkonditionierungstechnik zweckmäßig ist. Der Vaterknoten bildet dann mit dem hinzukommenden Endknoten V2, gewissermaßen der Mutter, ein Paar. Es werden auch beide als Vaterknoten des Kantenmittelknoten bezeichnet. Die beiden Endknoten V1, V2 gibt es auch an den Wurzeln der Kantenbäume, für die keine Väter im Sinne der Kantenstruktur existieren. Insofern ist die Knotenstruktur weiter ausgeprägt als die Kantenstruktur.

Die Knotenstruktur ist am unteren Ende um die Eckknoten der Hauptnetz-Dreiecke (*Hauptknoten*) zu erweitern, die im Sinne der Isomorphie keine Kanten darstellen, sondern zusätzliche Mutterknoten. Werden die Hauptknoten dem Knotenlevel 0 zugeordnet, so haben die Kantenmittelknoten des Hauptnetzes das Knotenlevel 1, wegen der Isomorphie zu den jeweiligen Kanten ist es also sinnvoll, das Kantenlevel der Hauptnetzkanten auf 1 festzulegen.

Die hierarchische Knoten-/Kantenliste HiNo

Die Knoten-/Kantenhierarchie wird durch die Liste **HiNo** repräsentiert. Am Anfang dieser Liste in den ersten NNOH Zeilen stehen die Hauptknoten. Sie haben weder Väter noch Söhne, weil sie im Rahmen der betrachteten Hierarchie keinen Kanten entsprechen. Sind sie jedoch bereits mit einem anderen Netzgenerator als entsprechende hierarchische Struktur erzeugt worden, so können ihre Positionen V1 und V2 für die hierarchische Vorkonditionierung mit den Vätern aus dieser Struktur belegt werden. Die Vorkonditionierung läuft dann über mehr Levels als die Adaption.

Alle Zeilen von **HiNo**, die nach denen der Hauptknoten stehen, entsprechen Kanten der Hierarchie. Bei diesen ist nur der Vater V1 auch Vater im Sinne der Kantenstruktur und auch das gilt nur, solange es sich nicht um eine Wurzelkante handelt. Andererseits ist V2 immer nur Vater im Sinne der Knotenstruktur (Mutter), auf Wurzelkanten gilt dies auch für V1. Die Söhne S1, S2 sind immer im Sinne der Kantenstruktur zu verstehen. Dabei werden die Endknoten den Kanten dadurch eindeutig zugeordnet, daß per definitionem S1 die auf der Seite von V1 gelegene Teilkante von Kante N0 ist. S1 besitzt somit die Vaterknoten V1(S1)=N0, V2(S1)=V1, S2 die Vaterknoten V1(S2)=N0, V2(S2)=V2.

Dimension	**HiNo**(6,NNO)
Bedeutung des Zeilenindex I	Listenplatz des Knotens / der Kante N0
Anzahl NNO	Gesamtzahl von Knoten (aktive und passive)
Ordnungsprinzip	Vaterkante steht vor Sohnkante
Anordnung der Listenelemente	N0 | V1 | V2 | S1 | S2 | TV in jeder Zeile
Bedeutung von N0	Knotennummer, ggf. auch Kantennummer
Bedeutung von V1	Knotennummer von Vater 1, ggf. Vater-Kantennummer
Bedeutung von V2	Knotennummer von Vater 2, ggf. Mutter-Knotennummer
Bedeutung von S1	Kantennummer des Sohnes 1 (V1 ist Endknoten)
Bedeutung von S2	Kantennummer des Sohnes 2 (V2 ist Endknoten)
Bedeutung von TV	Teilverhältnis (Knoten N0 nicht in Kantenmitte)
Besonderheiten:	
N0=0	Lücke (vor Zusammenschieben)
Anzahl NNOH	Anzahl der Hauptknoten
I≤NNOH	Zeilen der Hauptknoten
I>NNOH	Zeilen der Kantenmittelknoten bzw. Kanten
I>NNOH, S1=0	Kante besitzt keine Söhne

Die beim Rückbau der Hierarchie in **HiNo** entstehenden Lücken werden zunächst durch N0=0 markiert und müssen dann durch Zusammenschieben der Liste von hinten nach vorn beseitigt werden, damit das *Ordnungsprinzip* der Liste „Vater steht vor Sohn“ nicht durchbrochen wird. Dieses Ordnungsprinzip sichert die korrekte, levelweise Basistransformation in der hierarchischen Vorkonditionierung ab.

Die Knotenliste Node

Für die Beschreibung der aktuellen Elemente werden sämtliche Knoten mit ihren Lage-Informationen benötigt, sowohl aktive als auch passive. In der Liste **Node** wird jedem Knoten eine Zeile zugeordnet. In dieser stehen zuerst die Koordinaten des Knotens, danach ein Anzeiger, dessen einzelne Bits oder Bit-Kombinationen das Vorhandensein bestimmter Eigenschaften im Knoten markieren und dann ein Zeiger für die Einordnung des Knotens bzw. der zugehörigen Kante in **HiNo**. Als letztes folgt eine Zahl, die zur Unterscheidung aktiver Knoten (positiv) und passiver Knoten (negativ) dient und die für aktive Knoten die Nummer in einer (gedachten) Liste darstellt, die nur aus den aktiven Knoten besteht (für die Vektoren der Unbekannten und der rechten Seiten im FEM-Gleichungssystem).

Dimension	**Node**(5,NNOX)
Bedeutung des Zeilenindex INO	Knotennummer
Zeilenindex INO=NNOX	maximal mögliche Knotennummer
Anordnung der Listenelemente	| XN | YN | IANZ | IHI | IANO | in jeder Zeile
Bedeutung von XN	x-Koordinate des Knotens
Bedeutung von YN	y-Koordinate des Knotens
Bedeutung von IANZ	Anzeiger, seine Bits markieren Eigenschaften
Bedeutung von IHI	Zeiger auf Platz des Knotens in **HiNo** bzw.
	Zeiger auf die der Lücke vorhergehende Lücke
Bedeutung von IANO	Nummer als aktiver Knoten
Besonderheiten:	
NNCX	maximal mögliche Koppelknotenanzahl
INO$\leq$NNCX	Koppelknoten
INO>NNCX	Knoten, aber kein Koppelknoten
IANO > 0	Markierung als aktiver Knoten, zeitweise
	Nummer in (gedachter) Liste der aktiven Knoten,
IANO < 0	Markierung als passiver Knoten
IANO $= 0$	Lücke, die Zeile repräsentiert keinen Knoten
IANZ, Bit 29 = 1	nur Geschwindigkeitsknoten, kein Druck

Die bei der Parallelisierung auftretenden Koppelknoten werden im vorderen Bereich von **Node** in den Zeilen von 1 bis NNCX untergebracht, für die anderen Knoten stehen die Zeilen oberhalb NNCX zur Verfügung.

Auch die Knoten behalten ihre Nummer, d.h. ihren Platz in **Node**, während der gesamten Zeit ihrer Existenz, weil andernfalls unangenehme Umnumerierungen in **HiNo** und **Elem** die Folge wären. Die beim Rückbau der Hierarchie entstehenden Lücken werden wie bei **HiEl** als Teillisten von **Node** mit einfacher Verkettung über einen Pointer verwaltet, getrennt nach Koppelknoten-Lücken und übrigen.

Die Numerierung aktiver Knoten entsprechend einer gedachten lückenlose Liste aller aktiven Knoten kann durch Herauslassen der Lücken und passiven Knoten aus **Node** und anschließendes Zusammenschieben der Liste von hinten nach vorn erfolgen. Bei diesem Vorgehen stehen die Koppelknoten dann am Anfang der Liste der aktiven Knoten.

3.3 Die Verwaltung der Datenstrukturen

Die lokale Erweiterung bzw. der entsprechende Rückbau der Hierarchien im Verlaufe des Adaptionsprozesses führen zur Erweiterung der Datenstrukturen bzw. zum Entstehen von Lücken in ihnen. Das Vorgehen ist für die einzelnen Strukturen spezifisch und von ihrer Rolle in den Teilalgorithmen abhängig, von denen insbesondere zu berücksichtigen sind:

- Untersuchung der notwendigen Feinheit der Kanten mittels Fehlerschätzern (Zyklus über **HiNo**),
- Verfeinerung oder Vergröberung des Netzes (Zyklen über **Elem**),
- Assemblieren der Steifigkeitsmatrix für alle Knoten (Zyklus über **Elem**),
- Bereinigen der Steifigkeitsmatrix von Lücken und passiven Knoten (Zyklen über **HiNo** und **Node**),
- Komprimieren von Vektoren (Unbekannte, rechte Seiten) durch Entfernen von Lücken und Werten in passiven Knoten, entsprechend auch Expandieren (Zyklen über **Node** und **HiNo**),
- hierarchische Vorkonditionierung (Zyklen über **HiNo**),
- Netz- und Lösungsdarstellungen (Zyklen über **Elem**).

Während die Algorithmen beim Durchlaufen von **Elem** und **Node** die Beliebigkeit der Anordnung der Zeilen der Liste zulassen, ist beim Durchlaufen von **HiNo** in den meisten Fällen das Ordnungsprinzip „Vaterkante steht vor Sohnkante“ im Zusammenhang mit dem Durchlaufsinn (Liste vorwärts oder rückwärts durchlaufen) von grundlegender Bedeutung für die korrekte Durchführung des Algorithmus und muß auch beim Schließen von Lücken gewahrt bleiben.

Weiterhin besteht aus Sicht der Effektivität der Verwaltung der Listen die Forderung, daß die Basisobjekte der Listen, die Elemente und Knoten, nicht ständig ihre Nummern ändern dürfen, was andernfalls zu aufwendigen Umnumerierungen führen würde, zum Teil verbunden mit Suchprozessen.

Aus diesen Gesichtspunkten heraus ergeben sich die nachfolgenden Festlegungen für die Listenverwaltung.

HiEl: Der Index repräsentiert die Elementnummer, die stabil zu halten ist. Deshalb werden beim lokalen Rückbau der Hierarchie entstehende Lücken nicht durch Umspeichern gefüllt, sondern markiert (S4=0). Der entsprechende Index IEL der Lücke wird in eine Pointervariable **FreeHiEl** eingetragen, nachdem der vorhergehende Wert des Pointers in der Lücke IEL auf S1 abgelegt wurde. Somit entsteht in **HiEl** eine einfach verkettete Teilliste von Lücken, die über den Pointer zugänglich ist. Zu Beginn besteht diese Teilliste aus allen Zeilen von **HiEl** oberhalb der Hauptelemente bis zur letzten Zeile NELX einschließlich.

Node: Der Index repräsentiert die Knotennummer. Auch diese ist stabil zu halten, es wird deshalb ein analoges Konzept angewendet wie bei **HiEl**. Da jedoch die bei der Parallelisierung auftretenden Koppelknoten für verschiedene Vorkonditionierungen von den übrigen Knoten getrennt benötigt werden, wird **Node** in zwei getrennt verwaltete Speicherbereiche aufgeteilt. Die Zeilen 1 bis NNCX stehen für Koppelknoten zur Verfügung und werden mit dem Lückenpointer **FreeNc** verwaltet, in den Zeilen NNCX+1 bis NNOX dürfen nur solche Knoten untergebracht werden, die keine Koppelknoten sind, der zugehörige Lückenpointer heißt **FreeNode**. Die Lücken werden mit IANO=0 markiert, der vorangegangene Wert der Pointervariablen wird auf IHI abgelegt. Zu Beginn sind die Hauptknoten entsprechend in beiden Bereichen einzutragen, die Reste beider Bereiche sind als Lückenlisten zu verketten.

Elem: Da diese Liste relativ häufig durchlaufen wird, ist es wünschenswert, ständige Tests auf Vorliegen von Lücken zu vermeiden. Andererseits gibt es keinerlei Ordnungsanforderungen an die aktuellen Elemente. Es ist deshalb sinnvoll, neu entstehende aktuelle Elemente sofort in neu entstandene Lücken einzufügen, überzählige aktuelle Elemente am Schluß der Liste abzuspeichern und überzählige Lücken gleich durch Umspeicherungen von aktuellen Elementen vom Ende der Liste zu füllen.

HiNo: Diese Liste wird bei Anwendung der hierarchischen Vorkonditionierung besonders häufig durchlaufen, so daß Tests auf Lücken hier erst recht zu vermeiden sind. Da das Ordnungsprinzip „Vater steht vor Sohn" zu bewahren ist, dürfen Lücken nicht einfach mit neuen Söhnen gefüllt werden. Es empfiehlt sich deshalb, globalen Vergröberungs- und Verfeinerungslauf zu trennen. Nach jedem Vergröberungslauf, in dem Lücken entstanden, ist dann die Liste **HiNo** von hinten nach vorn zusammenzuschieben, wodurch die Ordnung nicht gestört wird. Die in einem Verfeinerungslauf neu entstehenden Söhne sind an das Ende von **HiNo** zu speichern, wodurch die Ordnung auch wieder erhalten bleibt.

4 Die Dynamik der Hierarchien

4.1 Prinzipien und Organisation des Netzumbaus

4.1.1 Der Basisumbau

Das Ergebnis jedes Netzumbaues soll in Analogie zur gleichmäßigen hierarchischen Vernetzung eine *kantenorientierte hierarchische FEM-Basis* aus den üblichen stückweise linearen Funktionen sein (Hütchenfunktionen). Kantenorientierte Hierarchie bedeute dabei, daß den Ausgangspunkt für den schrittweisen Aufbau die übliche FEM-Basis auf dem regulären Hauptnetz bilde, das auf irgendeine Weise aus Dreiecken generiert worden sei, und daß der elementare lokale Schritt des Ausbaus der Hierarchie nachfolgender Regel gehorche.

Basis-Ausbauregel:
Die Hinzunahme einer weiteren Basisfunktion ist nur möglich, indem zu zwei bereits vorhandenen, den Endpunkten einer Kante zugeordneten Basisfunktionen die dem Kantenmittelpunkt zugeordnete Hütchenfunktion vom nächsthöheren Feinheits-Level hinzugefügt wird.

Der lokale Abbau der Hierarchie ist die entsprechende Umkehrung dieses Vorgehens. Da zu einem Kantenmittelpunkt erst dann eine hierarchische Basisfunktion existieren darf, *nachdem* auch zu beiden Endpunkten Basisfunktionen existieren, kann die zu einem Knoten gehörende Basisfunktion auch nicht entfernt werden, solange vom Knoten noch eine Kante ausgeht, deren Mittelpunkt eine Basisfunktion zugeordnet ist. Damit muß der lokale Schritt des Abbaus der Basishierarchie der folgenden Regel entsprechen.

Basis-Abbauregel:
Die Entfernung einer Basisfunktion ist nur möglich, wenn sie zu keinem Hauptknoten gehört und zu keinem Vaterknoten einer Kante, deren Mittelpunkt noch eine Basisfunktion zugeordnet ist.

Während die Zulässigkeitsbedingung für den Basis-Ausbau ohne Kommunikation überprüfbar ist, da Koppelkanten immer mit allen ihren Knoten zu den jeweiligen Prozessoren gehören, wird zum Prüfen der Zulässigkeit des Entfernens einer Basisfunktion auf dem Koppelrand in der Regel Kommunikation nötig sein.

Die Entwicklung einer geeigneten Adaptionstechnik besteht nun darin, begründet festzulegen, in welchem Umfang, in welcher Reihenfolge und unter Einbeziehung welcher Lösungsinformationen elementare lokale Schritte zu definierten globalen Schritten des Basis-Umbaus zusammengefaßt werden. Für die Bestimmung eines tatsächlich die Lösung adaptierenden Algorithmus gibt es hier noch erheblichen Untersuchungsbedarf. Damit die Technologie der Netzgenerierung relativ unabhängig von der konkreten Adaptionsstrategie gestaltet werden kann, werden die elementaren globalen Schritte des Basis-Umbaus so bestimmt, daß in jedem von ihnen ein lokaler Aus- oder Abbau der Basishierarchie höchstens um ein Level erfolgt, wobei für Ausbau und Abbau separate Moduln zur Verfügung stehen sollten. Im Rahmen der Adaptionsstrategie können solche elementaren globalen Schritte dann auf unterschiedliche Weise mit Neuberechnungen der Näherungslösung kombiniert werden, wofür wiederum eine theoretische Fundierung wünschenswert ist.

4.1.2 Verfeinerungs- und Vergröberungskriterium

Für die Moduln zur Realisierung eines elementaren globalen Netzumbau-Schrittes (*Verfeinerungsmodul und Vergöberungsmodul*) wird die Existenz sowohl eines *Kantenverfeinerungskriteriums* als auch eines *Kantenvergröberungskriteriums* vorausgesetzt. Für jede Kante mit Basisfunktionen in beiden Endpunkten muß eindeutig und individuell entscheidbar sein, ob sie zu verfeinern ist in dem Sinne, daß dem Kantenmittelpunkt eine noch nicht zugeordnete Basisfunktion nun zugeordnet wird. Analog ist über das Entfernen der Basisfunktion von der Kante mittels eines Kriteriums zu entscheiden, wenn dies entsprechend

Basis-Abbauregel zulässig ist. Die Resultate beider Kriterien müssen dabei widerspruchsfrei sein. Die Entscheidung wird auf Grund einer auf der Kante bzw. in den Netzknoten definierten Bewertungsfunktion und eines Parameters getroffen, über den die Feinheit bzw. Grobheit des Netzes indirekt zu steuern ist. Bewertungsfunktionen können im Rahmen adaptiver Strategien zum Beispiel Fehlerschätzer sein, im Rahmen heuristischer Verfeinerungstechniken auch Lösungsiterierte oder Lösungen von der vorangegangenen Zeitschicht oder aus solchen gebildete Funktionen. Auch mehrere Bewertungsfunktionen und Steuerparameter in einem Kriterium sind denkbar, die Bewertungsfunktionen für Verfeinerung und Vergröberung müssen nicht die gleichen sein.

Im weiteren werde vorausgesetzt, daß Kantenvergröberungs- und Kantenverfeinerungskriterium rein kantenorientiert wirken, d.h. nur unter Einbeziehung der Bewertungsfunktionswerte auf der Kante selbst. Bei Parallelisierung ist die Entscheidung über Hinzu- oder Fortnehmen von Basisfunktionen auf Kanten des Koppelrandes nach Prüfung von deren Zulässigkeit dann ohne weitere Kommunikation mit dem Nachbarprozessor möglich, wenn die Bewertungsfunktion in den Koppelknoten verfügbar ist. Bis dahin kann jedoch auch Kommunikation zur Berechnung der Bewertungsfunktion selbst in den Koppelknoten erforderlich sein, etwa bei Benutzung von Residuen-Fehlerschätzern. Führen zum anderen Entscheidungen auf inneren Kanten eines Prozessor-Teilgebietes zur Verfeinerung/Vergröberung des zugehörigen Elementes, so sind auch ggf. zum gleichen Element gehörende Koppelkanten zu verfeinern/vergröbern. Darüber ist der Nachbarprozessor zu informieren.

Obige Kriterien könnten im einfachsten Fall wie folgt als Funktionen vereinbart werden:

logical function finer (edge, func, tol)
logical function coarser (edge, func, tol)

wobei edge die Nummer der Kante bzw. ihres Mittelpunktes in **HiNo** darstellt, func den Namen der Bewertungsfunktion (als Vektor von Knotenpunktwerten vorliegend) und tol den Steuerparameter. Der Wert **true** der beiden Funktionen entspricht der Entscheidung für Verfeinerung bzw. Vergröberung. Eine einfache heuristische Bedingung für die Verfeinerung könnte zum Beispiel das Überschreiten einer Toleranzgrenze durch die absolute oder relative Funktionsschwankung auf der Kante darstellen, für die Vergröberung analog das Unterschreiten, zum Beispiel:

```
logical function finer (edge, func, tol)
    finer = false ; v1 = HiNo(2, edge) ; v2 = HiNo(3, edge)
    test = abs( func(v1) – func(v2) )
    if (test > tol) then finer = true
```

Denkbar ist auch, daß Kantenvergröberungs- und Kantenverfeinerungskriterium elementorientiert arbeiten, daß also die Entscheidung über die Kante abhängig von den Informationen getroffen wird, die auf den beiden anliegenden Elementen zur Verfügung stehen. Der Zugriff auf diese Informationen ist jedoch nicht so einfach, da es sich nicht nur um aktuelle Elemente handelt, andererseits erfordert die Entscheidung bei Parallelisierung einen Informationsaustausch an den Koppelrändern. Diese Problematik kann auch in die Berechnung der Bewertungsfunktion verdrängt werden.

Der Modul zum Netzausbau wird so strukturiert, daß zunächst beim Durchlaufen der Kantenbäume (über **HiNo**) die zur Hinzunahme einer Basisfunktion zulässigen Mittelknoten markiert werden, danach folgt bei einem Lauf über **Node** in allen diesen zulässigen Knoten die Entscheidung über die Aktivierung der Basisfunktion mittels des Kantenverfeinerungskriteriums. Das Aktivieren der Basisfunktion ist in dem Sinne zu verstehen, daß die Information über das Vorhandensein einer Basisfunktion im Kantenmittelpunkt aktualisiert wird. Der Modul zum Netzabbau enthält analoge Schritte.

4.1.3 Der Umbau von Elemente- und Knoten-/Kantenhierarchie

Nach Abschluß der Entscheidungen über die von Knoten zu entfernenden oder ihnen neu zuzuordnenden Basisfunktionen wird mit Läufen über **Elem** die Elementehierarchie entsprechend aktualisiert, analog auch die Knoten-/Kantenhierarchie. Die Elementehierarchie muß dabei über das folgende Prinzip mit der Hierarchie der Basisfunktionen verbunden sein:

Genau dann, wenn mindestens einem Kantenmittelknoten des Dreieckselementes eine Basisfunktion zugeordnet wird, ist das Element durch Verbindung der Kantenmitten in vier kongruente Teilelemente zu zerlegen.

Dieses Prinzip für den Umbau der Elementehierarchie ist einerseits darin begründet, daß die neue Basisfunktion innerhalb des betrachteten Elementes auf eben diesen vier Teildreiecken jeweils linear ist, die Teildreiecke unter dem Gesichtspunkt der Bereitstellung von Informationen also notwendige elementare Einheiten bilden, aus denen der Träger der neuen Basisfunktion dargestellt werden kann. Bei Nichtvorhandensein der Basisfunktionen zu den drei Kantenmittelknoten des Elementes entfällt andererseits die Notwendigkeit solcher Darstellungseinheiten, Basisfunktionen mit feinerem Träger können innerhalb des Elementes wegen der Basis-Abbauregel sowieso nicht vorkommen. Auf Vergröberung und Verfeinerung separat angewendet, liefert das Umbauprinzip dann die folgenden Regeln:

ELEMENT-VERFEINERUNGSREGEL:
Erhielt mindestens einer der Kantenmittelknoten des aktuellen Dreieckselementes eine Basisfunktion zugeordnet, ist das Element durch Verbindung der Kantenmitten in vier kongruente Teilelemente zu zerlegen, die in **HiEl** *einzutragen sind. Das Element ist als aktuelles Element zu streichen, die vier Teilelemente sind als neue aktuelle Elemente in* **Elem** *aufzunehmen.*

ELEMENT-VERGRÖBERUNGSREGEL:
Sind allen drei Kantenmittelknoten des Vaterelementes zum betrachteten aktuellen Dreieckselement keine Basisfunktionen zugeordnet, werden die vier Söhne des Vaterelementes in **HiEl** *und als aktuelle Elemente gestrichen, das Vaterelement ist als neues aktuelles Element in* **Elem** *aufzunehmen.*

Die Basis-Ausbauregel sichert die Widerspruchsfreiheit der eben aufgeführten Regeln ab. Sind nämlich allen Kantenmitten des Vaterelementes keine Basisfunktionen zugeordnet, so entsprechen keiner Kante der Söhne zwei Basisfunktionen in den Vaterknoten, ist also die Situation für das Vergröbern gegeben, ist das Verfeinern ausgeschlossen. Das Umgekehrte folgt analog.

Im Zusammenhang mit der Element-Verfeinerung bzw. -Vergröberung sind auch die entsprechenden Knoten und Kanten in **Node** und **HiNo** zu erzeugen bzw. zu streichen, soweit sie nicht – durch Nachbarelemente veranlaßt – schon vorhanden bzw. noch nötig sind.

Die in den Netzknoten definierten FEM-Funktionen sind vor der weiteren Rechnung auf das neue Netz umsetzen. Beim Vergröbern ist dies trivial, die Werte der Funktionen in den gelöschten Knoten verfallen. Beim Verfeinern werden die Werte der FEM-Funktionen in den neuen Knoten mittels linearer Interpolation auf den eben verfeinerten Kanten erzeugt. Das entspricht der Benutzung des Kombinationsköffizienten null in der hierarchischen Basis für die bei der Verfeinerung neu hinzugekommenen Basisfunktionen.

4.1.4 6-Knoten-Dreiecke und das Paar Geschwindigkeit–Druck

Im Rahmen der gegenwärtig genutzten Datenstrukturen werden die 6-Knoten-Dreiecke nicht verwendet, um quadratische Dreieckselemente darzustellen, sondern als Makroelemente, um in Berechnungen inkompressibler Strömungen lineare Elemente für Druck und Geschwindigkeit auf Netzen unterschiedlicher Feinheit zu realisieren, vgl. die Abb. 3. Auf klassischen Netzen erfüllen diese Elemente dann die LBB-Bedingung. Bei den ungleichmäßig hierarchi-

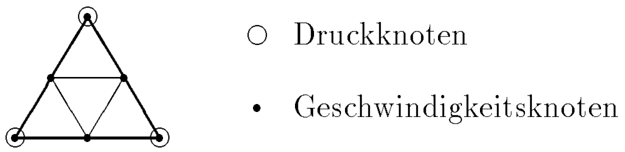


Abbildung 3: Benutzte Dreieckselemente

schen Netzen werde in Analogie zum klassischen Netz durchgängig mit einem *zusätzlichen Knotenlevel für die Ansatzfunktionen der Geschwindigkeit* gearbeitet. Die bisher beschriebene Hierarchie von Basisfunktionen entspreche deshalb zunächst dem Druck. Da bei der Erzeugung von Basisfunktionen in Kantenmittelknoten aktueller Dreiecke diese Dreiecke anschließend verfeinert werden, so sind nach dem Umbau der Hierarchien den Kantenmittelknoten aktueller Dreiecke keinesfalls Basisfunktionen des Druckes zugeordnet, sie stehen also für die Erzeugung eines zusätzlichen lokalen Hierarchielevels zur Verfügung. Es sind deshalb keine neue Knoten einzuführen, wenn mit dem gleichen Vorgehen wie beim Basisausbau ein zusätzliches Level von Basisfunktionen für die Geschwindigkeit aufgebaut wird. Die Erzeugung einer zusätzlichen Basisfunktion für die Geschwindigkeit in einem zulässigen Kantenmittelknoten erfolgt dabei unbedingt, wird also nicht von der Erfüllung einer Verfeinerungsbedingung abhängig gemacht.

Der Aufbau des zusätzlichen Levels von Basisfunktionen für die Geschwindigkeit geschieht somit wie folgt. Nach dem Umbau der Hierarchien wird ein weiterer Lauf über **HiNo** durchgeführt, bei dem Knoten ohne Basisfunktion, deren beide Väter eine solche besitzen, ausnahmslos auch eine Basisfunktion zugeordnet erhalten, wobei in IANZ von **Node** markiert wird, daß diese für die Geschwindigkeit, nicht aber für den Druck gilt. Diese zusätzlichen Basisfunktionen sind bis unmittelbar vor dem nächsten Vergröberungs- bzw. Verfeinerungslauf über **HiNo** verfügbar, müssen dann aber zusammen mit der Markierung gelöscht werden, die sie als nur zur Geschwindigkeit gehörend kennzeichnet, weil sie sonst den Umbau der Druckbasis behindern. Das Löschen geschieht bei einem Lauf über **Node** mit Abfragen der erwähnten Markierung.

Die zusätzlichen Basisfunktionen für die Geschwindigkeit müssen in den Datenstrukturen nicht notwendig zur expliziten Einführung eines weiteren Elemente- und Kantenlevels führen. Es reicht aus, sie nur zu markieren und erst bei der Assemblierung auf den Macro-Elementen der Abb. 3 einzubeziehen. Ein noch zu untersuchendes Problem ist, ob die so konstruierten ungleichmäßig hierarchischen Basen für Druck und Geschwindigkeit die LBB-Bedingung entsprechender bilinearer Funktionale erfüllen.

Für weitere unbekannte Funktionen der zu lösenden Aufgabe, etwa skalare Transportgrößen, kann zum Beispiel auf die Basis der Geschwindigkeit zurückgegriffen werden.

4.2 Algorithmus des Netzumbaus

Mit bereits in Abschnitt 2 benutzten Begriffsbildungen wird das beschriebene Vorgehen formalisiert. Ein *aktiver Knoten* sei ein Netzknoten, dem eine FEM-Basisfunktion zugeordnet wurde. Die anderen Netzknoten heißen *passive Knoten*. Die Markierung als aktiver oder passiver Knoten erfolgt in **Node** durch einen positiven oder negativen Wert des Listenelementes IANO. Beim Paar Druck–Geschwindigkeit bezieht sich diese Markierung während des Netzumbaus auf den Druck. Werden zusätzliche Knoten für Basisfunktionen der Geschwindigkeit aktiviert, erhalten diese eine Markierung im Listenelement IANZ von **Node**. Insbesondere sind die Hauptknoten alles aktive Knoten, die Kantenmittelknoten der Hauptnetzdreiecke sind zu Beginn nur aktive Knoten für die Geschwindigkeit.

4.2.1 Netzverfeinerung

Eine Kante mit passivem Mittelknoten, deren beide Endknoten aktiv sind, heiße *aktivierbare Kante* bzw. der Mittelknoten selbst *aktivierbarer Knoten*. Die Basis-Ausbauregel hat zum Inhalt, daß nur die aktivierbaren Knoten als passive Knoten aktiv gesetzt werden dürfen, was dann als Aktivieren des Knotens bzw. der Kante bezeichnet werde. Das Aktivieren einer Kante führt dazu, daß aus ihren beiden Teilkanten wiederum aktivierbare Kanten werden, sich die Ausgangssituation also auf einem um eins höheren Hierarchielevel reproduziert. Dieser elementare lokale Schritt des Hierarchieausbaus erfolgt auf einer aktivierbaren Kante nicht notwendig, sondern nur auf Grund des erfüllten Verfeinerungskriteriums finer.

Der Umbau der Elementehierarchie wird im Zyklus über die aktuellen Elemente in **Elem** realisiert, nachdem die Markierung der aktiven Knoten durch Aktivierung aktualisiert wurde. Ausgangspunkt ist der lokale Umbau der Elementehierarchie um höchstens ein Level auf Grundlage der Element-Verfeinerungsregel. Dementsprechend wird auch die Knoten-/Kantenhierarchie ausgebaut. Die neuentstandenen Knoten sind zu markieren, damit sie bei der Umsetzung der FEM-Funktionen auf die neue Netzstruktur erkannt werden. Der Verfeinerungsalgorithmus hat dann die weiter unten ab Seite 18 dargestellte grundlegende Struktur.

4.2.2 Netzvergröberung

Entsprechend der Basis-Abbauregel werden analog die Bezeichnungen *deaktivierbarer Knoten* usw. eingeführt und das Deaktivieren werde analog mit dem Vergröberungskriterium coarser durchgeführt.

Das Vergröbern der Elemente wird dann zwar als Zyklus über **Elem** organisiert, de facto werden aber immer vier zu einem gemeinsamen Vater gehörende aktuelle Elemente abgearbeitet, sobald das erste dieser vier in **Elem** erreicht wurde. Diesem Vorgriff in der Abarbeitung von **Elem** wird Rechnung getragen, indem die weiter hinten stehenden restlichen drei Elemente als bereits abgearbeitet markiert werden.

Der Rückbau der Knoten-/Kantenhierarchie ist erst nach vollzogenem Rückbau der Elementehierarchie im gesamten Netz realisierbar. Durch Element-Vergröberung rückt zwar das lokale Wegfallen eines Kantenlevels in den Bereich der Möglichkeiten, dies muß aber immer mit den auf den anderen Seiten der Kanten anliegenden aktuellen Elementen und ggf. deren Veränderungen verträglich sein, worüber erst nach Bearbeitung sämtlicher aktuellen Elemente Klarheit besteht. Dann gibt **Elem** über die noch benötigten Knoten/Kanten Aufschluß. Die beim Rückbau in **HiNo** entstehenden Lücken müssen durch Zusammenschieben dieser Liste beseitigt werden. Der Vergröberungsalgorithmus hat dann die ab Seite 20 dargestellte grundlegende Struktur.

4.2.3 Verwaltungsfunktionen

Weiter vorn wurde bereits erklärt, daß die Hauptdatenstrukturen im Hinblick auf das Streichen in ihnen gespeicherter Objekte bzw. das Neueinspeichern solcher zu verwalten sind. Dazu werden entsprechende Moduln mit Verwaltungsfunktionen bereitgestellt. Das betrifft nicht die Lücken in **HiNo**, die durch Zusammenschieben der Liste beseitigt werden, sowie das Platzieren neuer Kanten in **HiNo** und neuer aktueller Elemente in **Elem**, das immer am Ende dieser Listen vorgenommen wird. Für die übrigen Aufgaben sind folgende Moduln nötig, vgl. auch Abschnitt 3.3 :

MODUL GAPHIEL (IEL, HiEl, FreeHiEl)

Funktion:	Löschen des Elementes IEL in **HiEl**
Input:	IEL, HiEl, FreeHiEl
Output:	HiEl, FreeHiEl

MODUL LOCHIEL (IEL, HiEl, NELX, FreeHiEl)

Funktion:	Umwandlung der nächsten Lücke in **HiEl** in eine freie Elementnummer IEL
Input:	HiEl, NELX, FreeHiEl
Output:	IEL, FreeHiEl

Modul GapElem (I, Elem, NEL)

Funktion:	Überspeichern des aktuellen Elementes I in **Elem** mit aktuellem Element vom Ende von **Elem**
Input:	I, Elem, NEL
Output:	Elem, NEL

Modul GapNode (INO, Node, NNCX, FreeNc, FreeNode)

Funktion:	Löschen des Knotens INO in **Node**
Input:	INO, Node, NNCX, FreeNc, FreeNode
Output:	Node, FreeNc, FreeNode

Modul LocNode (INO, IEQ, Node, NNCX, NNOX, FreeNc, FreeNode)

Funktion:	Umwandeln der nächsten Lücke in **Node** in eine freie Knotennummer INO in **Node** ohne Kennzeichnung als aktiver oder passiver Knoten, aber ggf. mit Kennzeichnung als Koppelknoten äquivalent zu Knoten IEQ
Input:	IEQ, Node, NNCX, NNOX, FreeNc, FreeNode
Output:	INO, FreeNc, FreeNode

4.2.4 Umbau-Moduln

Modul NetFiner (elementarer globaler Verfeinerungsschritt):

Vorbereitung:

Zyklus (Node)

- Knoten zusätzlicher Basisfunktionen der Geschwindigkeit deaktivieren

Teilmodul Basishierarchie-Ausbau:

Markierung aller aktivierbaren passiven Knoten:

Zyklus (HiNo)

- passive Knoten mit zwei aktiven Vätern als aktivierbar markieren

Aktivierungsentscheidung in aktivierbaren passiven Knoten:

Zyklus (Node)

- aktivierbare passive Knoten bei erfüllter lokaler Verfeinerungsbedingung (finer) aktiv setzen

TEILMODUL ELEMENTE-/KNOTEN-/KANTENHIERARCHIE-AUSBAU:

Elementverfeinerung nach Element-Verfeinerungsregel
und zugehöriger Ausbau der Knoten-/Kantenhierarchie:

Zyklus (Elem)
- Elementverfeinerungsregel:
  falls wenigstens ein Kantenmittelknoten des aktuellen Elementes aktiv ist:
  Verfeinerung:
  - Element als aktuelles Element in Elem freiziehen
  - vier Söhne des Elementes in HiEl und als aktuelle Elemente in Elem erzeugen, Informationen über ihre Eckknoten einordnen
  - Zyklus über die neun Kanten der vier Söhne:
    - falls ein Kantenmittelknoten bereits existiert:
      - Feststellen des Kantenmittelknotens,

      sonst:
      - Erzeugung des Kantenmittelknotens in Node und als Kante in HiNo
    - Kantenmittelknoten in die Elementinformationen der Söhne in Elem eintragen
  - Markierung der verfeinerten passiven Koppelkanten (Mittelknoten!) des aktuellen Elementes

Kantenverfeinerungen, die nur durch Nachbarprozessor veranlaßt sind:

Kommunikation: Verfeinerung der passiven Koppelkanten abstimmen
Zyklus (Node), aber nur über die Koppelknoten:
- falls Koppelkante (Mittelknoten!) als zu verfeinernde Kante markiert ist, aber (auf dem Prozessor) keine Söhne besitzt,
  Verfeinerung:
  - Erzeugung beider Söhne durch Erzeugung ihrer Kantenmittelknoten in Node und als Kanten in HiNo

Nachbereitung:

Zyklus (Node)
- Knoten zusätzlicher Basisfunktionen der Geschwindigkeit markieren

FEM-FUNKTIONEN AUF NEUES NETZ UMSETZEN:

Zyklus (HiNo)
- Werte der FEM-Funktionen in den neuen Knoten mittels linearer Interpolation erzeugen

NETZABSCHLUSS:

Zyklus(Node)
- Aktivierung der zusätzlichen Geschwindigkeitsknoten

Feststellen der neuen Netzparameter
(diverse Knoten-, Kanten- bzw. Elementeanzahlen usw.)

Kommunikation auf neuem Netz vorbereiten

MODUL NETCOARSER (elementarer globaler Vergröberungsschritt):

Vorbereitung:

Zyklus (Node)

- Knoten zusätzlicher Basisfunktionen der Geschwindigkeit deaktivieren

TEILMODUL BASISHIERARCHIE-RÜCKBAU:

Markierung aller nicht deaktivierbaren aktiven Knoten:

Zyklus (HiNo)

- Hauptknoten als nicht deaktivierbar markieren
- aktive Väter aktiver Knoten als nicht deaktivierbar markieren

Kommunikation: Deaktivierbarkeit der Koppelknoten abstimmen

Deaktivierungsentscheidung in deaktivierbaren aktiven Knoten:

Zyklus (Node)

- deaktivierbare aktive Knoten bei erfüllter lokaler Vergröberungsbedingung (coarser) passiv setzen

TEILMODUL ELEMENTEHIERARCHIE-RÜCKBAU:

Elementvergröberung nach Element-Vergröberungsregel:

Zyklus (Elem), solange unbearbeitete aktuelle Elemente vorhanden sind

- falls Hauptelement, keine Vergröberung möglich
- Vater- und Brüder-Elemente ermitteln
- Kantenmittelknoten des Vaters ermitteln
- Element-Vergröberungsregel:
  falls alle drei Kantenmittelknoten des Vaters passiv sind,
  Vergröberung:
    - Rekonstruktion des Vaters als aktuelles Element in Elem und HiEl
    - Löschen der Söhne als aktuelle Elemente in Elem und in HiEl
  sonst
  keine Vergröberung:
    - aktuelles Element und seine unverfeinerten Brüder als bearbeitet markieren

TEILMODUL KNOTEN-/KANTENHIERARCHIE-RÜCKBAU:

Markierung aller nicht löschbaren Knoten:

Zyklus (Elem)

- Markierung aller Knoten der verbliebenen aktuellen Elemente als nicht löschbar

Kommunikation: Löschbarkeit der Koppelknoten abstimmen

Rückbau der Knoten-/Kantenhierarchie:

Zyklus (HiNo)
- falls Knoten (=Kante) löschbar:
  - Vater, falls Kantenvater, verliert in HiNo den Sohn
  - Knoten wird in HiNo und Node gelöscht

Zyklus (HiNo)
- Zusammenschieben von HiNo

Nachbereitung:

Zyklus (Node)
- Knoten zusätzlicher Basisfunktionen der Geschwindigkeit markieren

FEM-FUNKTIONEN AUF NEUES NETZ UMSETZEN:

Zyklus (HiNo)
- Werte der FEM-Funktionen in den gelöschten Knoten verfallen

NETZABSCHLUSS:

Zyklus(Node)
- Aktivierung der zusätzlichen Geschwindigkeitsknoten

Feststellen der neuen Netzparameter
(diverse Knoten-, Kanten- bzw. Elementeanzahlen usw.)

Kommunikation auf neuem Netz vorbereiten

5 Die Erzeugung des FEM-Gleichungssystems

5.1 Verschiedene Basen

Im Rahmen des Netzaufbaus und -umbaus wird jeweils eine aktuelle kantenorientierte hierarchische Basis erzeugt. Die einzelnen Basisfunktionen sind entsprechend Abschnitt 4.1 definiert

- als den Hauptknoten zugeordnete übliche stückweise lineare Hütchenfunktionen auf dem Hauptnetz bzw.
- als den aktiven Kantenmittelknoten von Dreiecken der Elementhierarchie zugeordnete stückweise lineare Hütchenfunktionen mit einem dem Hierarchielevel entsprechenden kleineren Träger.

Bei einer Kante mit aktivem Mittelknoten im Inneren von Ω (bzw. Ω_s bei Parallelisierung) ist durch die sukzessive Anwendung von Basis-Ausbauregel und Element-Verfeinerungsregel abgesichert, daß dies die gemeinsame Kante beiderseits anliegender Dreieckselemente ist. Der Träger der dem Kantenmittelknoten zugeordneten Basisfunktion wird durch die sechs um den Mittelknoten gruppierten Sohnelemente der beiden beteiligten Elemente gebildet, bei Randkanten gilt Analoges mit nur einem beteiligten Element. Diese hier nochmals beschriebene Basis werde als *aktive hierarchische Basis* bezeichnet.

Von den günstigen Auswirkungen auf die Kondition des FEM-Gleichungssystems abgesehen, führt die Verwendung einer solchen hierarchischen Basis auf Grund erheblicher Überlappungen der Träger von Basisfunktionen zu einem unakzeptablen Auffüllen der Steifigkeitsmatrix mit Nichtnullelementen. Bei gleichmäßiger hierarchischer Verfeinerung wird die Steifigkeitsmatrix deshalb unter Benutzung der üblichen FEM-Knotenbasis erzeugt, wobei eine einfach zu organisierende, billige, nach Levels faktorisierte Transformation der hierarchischen in die FEM-Basis verfügbar ist (Yserentant [9]). Mit analoger Transformation, aber die Ungleichmäßigkeit der hierarchischen Verfeinerung berücksichtigend, kann aus der aktiven hierarchischen Basis eine Basis mit zum Teil deutlich kleineren Trägern erzeugt werden. Diese werde *aktive FEM-Basis* genannt und letzten Endes zur Erzeugung der Steifigkeitsmatrix benutzt.

Während die Hierarchie der Elemente und der damit verbundenen Eck- und Kantenmittelknoten noch relativ homogen bearbeitet werden kann, d.h. levelweise einheitlich, führt die bezüglich des Netzes ziemlich willkürliche Auswahl aktiver Knoten zu erheblichen Inhomogenitäten im Algorithmus der Assemblierung der Steifigkeitsmatrix, also zu aufwendigen Fallunterscheidungen im Hinblick auf die Gestalt der Basisfunktionen bzw. die Überlappungsgebiete von deren Trägern. Diesem Problem kann man begegnen, indem zunächst auch den passiven Knoten Basisfunktionen zugeordnet werden und damit auf jedem Element eine homogene Behandlung der Knoten stattfindet. Nach der Assemblierung sind die Wirkungen dieser passiven Basisfunktionen wieder aus der Steifigkeitsmatrix zu entfernen. Die Basen, die sämtliche erzeugten Knoten berücksichtigen, aktive und passive, sollen entsprechend *verfügbare hierarchische Basis* und *verfügbare FEM-Basis* genannt werden.

Zu berücksichtigen ist, daß im Inneren von Ω passive Kantenmittelknoten nicht notwendig beiderseits der Kante zu einem Element als dessen Kantenmittelknoten gehören. In diesem Fall ist es sinnvoll, mit „halben“ Hütchenfunktionen zu arbeiten, die zwar entlang der Kante unstetig sind, letztendlich jedoch sowieso wieder eliminiert werden.

Entsprechend ihrer Stufe in der Hierarchie werden die Elemente und Knoten nach Levels eingeteilt, wobei Level 0 den Hauptelementen entspricht und Level J die höchste in **HiEl** vorkommende Hierarchiestufe darstellt. Dabei enthalten die aktuellen Elemente mit den Kantenmittelknoten implizit die Informationen eines weiteren lokalen Hierarchielevels, das aber erst unmittelbar bei der Assemblierung und nur für die Geschwindigkeit genutzt wird, in **HiEl** jedoch nicht auftaucht. Die folgenden Darstellungen beziehen sich auf die Basen ohne Nutzung des zusätzlichen Hierarchielevels, also für den Druck, während für die Geschwindigkeit analog unter Hinzuziehung dieses Levels zu verfahren ist. Zunächst werden einige Bezeichnungen eingeführt.

Elementlevels:

E_0^*	=	Menge der Hauptelemente (Level 0–Elemente),
E_k^*	=	Menge der Level k–Elemente, $k = 1(1)J$, wobei jedes $e \in E_k^*$ bei der Zerlegung eines $e_* \in E_{k-1}^*$ in vier kongruente Dreiecke mit entstanden ist,
E_k	=	Menge der aktuellen Level k–Elemente, $k = 0(1)J$.

Offenbar sind sowohl die E_k^* als auch die E_k paarweise disjunkt, wobei immer $E_k \subset E_k^*$ gilt, speziell $E_J = E_J^*$. Die Menge $E_k^* \setminus E_k$ ist die Menge aller Level k–Elemente, die weiter verfeinert wurden. $E = \cup_{k=0}^J E_k$ bildet die Menge sämtlicher aktuellen Elemente. Weiterhin seien

$$E^{(k)} = \left(\bigcup_{l=0}^{k-1} E_l \right) \cup E_k^*, \quad k = 0(1)J.$$

Dies wären die Mengen aktueller Elemente, wenn die Verfeinerungen oberhalb des Levels k rückgängig gemacht würden. Speziell ist natürlich $E^{(J)} = E$.

Knotenlevels:

M_0	=	Menge der Hauptknoten, d.h. Eckknoten der Level 0–Dreiecke, (Level 0–Knoten),
M_k	=	Menge derjenigen Eckknoten der Level k–Dreiecke, die keine Eckknoten von Dreiecken niedrigeren Levels sind, (Level k–Knoten), $k = 1(1)J$.

$M = \cup_{k=0}^J M_k$ ist dann die Menge aller Eckknoten der Elemente, weil mit Ausnahme der Kantenmittelknoten der aktuellen Elemente genau diese Knoten im Prozeß der Konstruktion der Elementhierarchie systematisch erzeugt wurden. M_k enthält für $k \geq 1$ nur Kantenmittelknoten von Level $(k-1)$–Dreiecken. Weiterhin seien

$$M^{(k)} = \bigcup_{l=0}^{k} M_l, \quad k = 0(1)J,$$

speziell ist $M^{(J)} = M$. Mit dem zusätzlichen Index a oder p werden die Durchschnitte der entsprechenden Knotenmengen mit der Menge $M_{(a)}$ aller aktiven oder $M_{(p)} = M \setminus M_{(a)}$ aller passiven Eckknoten bezeichnet, also zum Beispiel

$$M_{k,p} = M_k \cap M_{(p)}, \qquad M_a^{(k)} = M^{(k)} \cap M_{(a)}.$$

Hierarchische Basisfunktionen:

Für $i \in M_0$ seien die θ_i^0 die Hütchenfunktionen der üblichen FEM-Basis auf dem Hauptnetz. Für $i \in M_k$, $k = 1(1)J$, gehört Knoten i mindestens einem Element $e \in E_{k-1}$ als Kantenmittelknoten an, aber höchstens einem weiteren $e' \in E_{k-1}$. Dann sei $\theta_{i,e}^k$ die „halbe" Hütchenfunktion, die außerhalb von e identisch null ist, im Knoten i den Wert eins hat und zu den beiden benachbarten Kantenendpunkten sowie den beiden anderen Kantenmittelpunkten von e linear auf null abfällt. Entlang der Kante von e, auf der i selbst liegt, ist diese Funktion also unstetig, soweit die Kante nicht zu $\partial\Omega$ gehört.

Die Basisfunktion θ_i^k sei dann identisch mit der „halben“ Hütchenfunktion $\theta_{i,e}^k$, falls i nur einem Element e als Kantenmittelknoten angehört, andernfalls mit der entsprechenden „ganzen“ Hütchenfunktion, d.h. für $i \in M_k,\ k = 1(1)J$ sei

$$\begin{aligned} \theta_i^k &= \theta_{i,e}^k, && \text{falls } i \text{ Kantenmittelknoten nur von } e, \\ \theta_i^k &= \theta_{i,e}^k + \theta_{i,e'}^k, && \text{falls } i \text{ Kantenmittelknoten von } e, e'. \end{aligned}$$

Liegt Knoten i auf einer aktivierbaren Kante, so ist θ_i^k offenbar diejenige Basisfunktion, die beim aktiv-Setzen von i zur Basis hinzugenommen wird. Wie oben bemerkt, sichern Basis-Ausbauregel und Element-Verfeinerungsregel ab, daß dies außer auf $\partial\Omega$ eine „ganze“, also immer eine stetige Hütchenfunktion ist.

Im Hinblick auf die Darstellung von Basistransformationen werden Basen im weiteren *immer als Zeilenvektoren* verstanden, in denen die Basisfunktionen entsprechend der Reihenfolge ihrer zugeordneten Knoten nach folgendem Schema geordnet sind:

$$M_{0,a} \;\; M_{0,p} \;\; M_{1,a} \;\; M_{1,p} \;\; \cdots \;\; M_{J,a} \;\; M_{J,p} \tag{1}$$

bzw. bei nur auf aktive Knoten bezogenen Basisfunktionen

$$M_{0,a} \;\; M_{1,a} \;\; \cdots \;\; M_{J,a},$$

wobei die Ordnung innerhalb von $M_{k,a}$ und $M_{k,p}$ beliebig, aber fest sei.

Verfügbare hierarchische Basis Θ :

$$\begin{aligned} \Theta^0 &= \{\theta_i^0 \mid i \in M_0\}, && \text{übliche FEM-Basis auf dem Hauptnetz,} \\ \Theta^k &= \{\theta_i^k \mid i \in M_k\}, && k = 1(1)J, \\ \Theta &= (\,\Theta^0\,\Theta^1\,\cdots\,\Theta^J\,) = (\,\theta_1^0\,\cdots\,\theta_{\mathrm{NNO}}^J) && \text{verfügbare hierarchische Basis.} \end{aligned}$$

Aktive hierarchische Basis Θ_a :

Ist I_a die *Matrix der Basiseinschränkung auf die aktiven Knoten*, also eine Rechteckmatrix, die aus der entsprechenden Einheitsmatrix entsteht durch Streichen der Spalten der passiven Knoten, so sei

$$\Theta_a = \Theta\, I_a. \tag{2}$$

Aus der hier definierten verfügbaren bzw. aktiven hierarchischen Basis kann die verfügbare bzw. aktive FEM-Basis durch eine zu [9] analoge Transformation erzeugt werden, die jedoch keine Gleichmäßigkeit der Behandlung der Basisfunktionen ein und desselben Levels benutzt, d.h. es gibt solche, die transformiert werden, und andere, bei denen dies nicht der Fall ist, weil dort lokal bereits das maximale Hierarchielevel erreicht wurde.

Sei $K(i,k)$ die Menge der Kantenmittelknoten aller von Knoten i ausgehenden Seitenkanten von Dreiecken aus $E_k^* \backslash E_k$, das heißt von weiter verfeinerten Dreiecken des Levels k. Offenbar gilt $K(i,k) \subset M_{k+1} \subset M \setminus M^{(k)}$. Dann sei die *verfügbare FEM-Basis* $\Psi = \{\psi_i \mid i \in M\}$ durch folgende levelweise faktorisierte Transformation definiert:

Mit Θ *als Ausgangsbasis*

$$\psi_i^0 = \theta_i^l, \quad i \in M_l,\ l = 0(1)J,$$

werde für $k = 0(1)J-1$ *berechnet*

$$\begin{aligned} \psi_i^{k+1} &= \psi_i^k, & i \in M \setminus M^{(k)}, \quad (3)\\ \psi_i^{k+1} &= \psi_i^k - \frac{1}{2} \sum_{j \in K(i,k)} \psi_j^k, & i \in M^{(k)}, \quad (4) \end{aligned}$$

das Ergebnis des letzten Schrittes sei die verfügbare FEM-Basis

$$\psi_i = \psi_i^J, \quad i \in M.$$

Diese levelweise faktorisierte Transformation der hierarchischen Basis kann auch in einer Formel zusammengefaßt werden.

$$\textit{Für}\, i \in M_k,\ k = 0(1)J:$$

$$\begin{aligned} \psi_i &= \psi_i^0 - \frac{1}{2} \sum_{l=k}^{J-1} \sum_{j \in K(i,l)} \psi_j^0 \\ &= \theta_i^k - \frac{1}{2} \sum_{l=k}^{J-1} \sum_{j \in K(i,l)} \theta_j^{l+1}. \end{aligned} \quad (5)$$

Die obige Definition der verfügbaren FEM-Basis werde in Matrixform geschrieben.

Verfügbare FEM-Basis Ψ :

$$\Psi = \Theta\, T \quad \text{mit} \quad T = T^{(0)}\, T^{(1)} \cdots T^{(J-1)}, \quad (6)$$

wobei die levelbezogenen Transformationsmatrizen $T^{(k)}$ offensichtlich invertierbare Blockmatrizen der nachfolgenden Struktur darstellen. Mit I werden dabei Einheitsmatrizen, mit O Nullmatrizen unterschiedlichen Typs symbolisiert, deren Größe sich aus den angegebenen Knotenmengen ergibt.

$$T^{(k)} = \begin{array}{cl} \begin{pmatrix} I & O & O \\ G^{(k)} & I & O \\ O & O & I \end{pmatrix} & \begin{array}{l} \} \ M_0 \ldots M_k \\ \} \ M_{k+1} \\ \} \ M_{k+2} \ldots M_J \end{array} \\ \begin{array}{ccc} \underbrace{\quad}_{M_0 \ldots M_k} & \underbrace{\quad}_{M_{k+1}} & \underbrace{\quad}_{M_{k+2} \ldots M_J} \end{array} & \end{array} \quad (7)$$

wobei

$$G^{(k)} = \begin{array}{cl} \begin{pmatrix} G^{k+1,0} & \cdots & G^{k+1,k} \end{pmatrix} & \} \ M_{k+1}. \\ \begin{array}{ccc} \underbrace{\qquad}_{M_0} & \cdots & \underbrace{\qquad}_{M_k} \end{array} & \end{array}$$

Jede Zeile von $G^{(k)}$ entspricht einem Knoten von M_{k+1}, dessen Basisfunktion in (4) zur Transformation der Basisfunktion seines Vaters benutzt wird. Da Kantenmittelknoten genau zwei Väter besitzen, besteht jede Zeile von $G^{(k)}$ wie im Spezialfall gleichmäßiger Hierarchien aus zwei Elementen $-\frac{1}{2}$ in den Spalten beider Väter und sonst aus lauter Nullen. Weil die Väter aktiver Knoten nach Basis-Ausbauregel auch immer selbst aktiv sein müssen, haben die $G^{k+1,l}$ folgende Blockstruktur:

$$G^{k+1,l} \;=\; \begin{array}{cc} \begin{pmatrix} G_1^{k+1,l} & O \\ G_2^{k+1,l} & G_3^{k+1,l} \end{pmatrix} & \begin{array}{l} \} \;\; M_{k+1,a} \\ \} \;\; M_{k+1,p}. \end{array} \\ \begin{array}{cc} \underbrace{\qquad\quad}_{M_{k+1,a}} & \underbrace{\qquad\quad}_{M_{k+1,p}} \end{array} & \end{array} \tag{8}$$

Wie sich noch zeigen wird, ist diese Struktur von entscheidender Bedeutung für eine einfache Assemblierungstechnologie der Steifigkeitsmatrix.

Die *aktive FEM-Basis* Φ_a kann analog zu Ψ definiert werden, jedoch unter Nutzung nur der aktiven Knoten, also von $M_{(a)}$, $M_a^{(k)}$, $K_a(i,k)$ anstelle M, $M^{(k)}$, $K(i,k)$, ausgehend von Θ_a. Dann ergeben sich zu (3), (4) bzw. (5) analoge Darstellungen, zum Beispiel:

$$\Phi_a \;=\; \{\varphi_i \mid i \in M_{(a)}\}, \qquad \text{wobei für } i \in M_{k,a},\ k = 0(1)J:$$

$$\varphi_i \;=\; \theta_i^k - \frac{1}{2} \sum_{l=k}^{J-1} \sum_{j \in K_a(i,l)} \theta_j^{l+1}. \tag{9}$$

In der Matrixdarstellung entfallen dann die zu den passiven Knoten gehörenden Zeilen und Spalten. Es kann jedoch auch der gleiche Matrixtyp wie in (6) verwendet werden, indem von Θ ausgegangen wird, die Basisfunktionen zu passiven Knoten jedoch weder geändert, noch bei der Transformation anderer Funktionen benutzt, am Schluß aber mit I_a gestrichen werden.

Aktive FEM-Basis Φ_a :

$$\Phi \;=\; \Theta\, T_a \quad \text{mit} \quad T_a = T_a^{(0)}\, T_a^{(1)} \cdots T_a^{(J-1)}, \tag{10}$$

$$\Phi_a \;=\; \Phi\, I_a, \tag{11}$$

wobei $T_a^{(k)}$ aus $T^{(k)}$ entsteht, indem die $G^{(k)}$, $G^{k+1,l}$ durch entsprechende $G_a^{(k)}$, $G_a^{k+1,l}$ ersetzt werden mit

$$G_a^{k+1,l} = \begin{pmatrix} G_1^{k+1,l} & O \\ O & O \end{pmatrix}. \tag{12}$$

5.2 Die Assemblierung von Matrix und rechter Seite

Modellaufgabe

Es werde ein elliptisches Problem zweiter Ordnung betrachtet, das als Variationsgleichung in $V = W_2^1(\Omega)$ für ein polygonales Gebiet Ω vorliege:

$$u \in V : \quad a(u,v) \;=\; f(v) \quad \forall v \in V\,. \tag{13}$$

Das bilineare Funktional $a(.,.)$ und das lineare Funktional $f(.)$ mögen die Bedingungen des Lemmas von Lax-Milgram erfüllen, so daß Existenz und Eindeutigkeit von exakter und FEM-Lösung gesichert sind. Auch Dirichlet-Randbedingungen könnten wie üblich einbezogen werden und wurden nur der Übersichtlichkeit wegen nicht berücksichtigt.

Assemblieren in der verfügbaren FEM-Basis

Für das FEM-Schema soll die aus den C^0-Funktionen φ_i bestehende aktive FEM-Basis Φ_a verwendet werden. Da *supp* φ_i von der Positionierung der aktiven Nachbarknoten abhängt, führt die Assemblierung der entsprechenden Steifigkeitsmatrix und der rechten Seite des FEM-Gleichungssystems zu unangenehmen Fallunterscheidungen. Günstiger ist eine Assemblierung unter Verwendung der verfügbaren FEM-Basis mit anschließender Transformation, wie die weiteren Überlegungen zeigen.

Die ψ_i^{k+1} aus (3), (4) haben folgende leicht nachprüfbaren Eigenschaften:

a) Für $i \in M \setminus M^{(k)}$ handelt es sich um unveränderte Funktionen aus der verfügbaren hierarchischen Basis:

$$\psi_i^{k+1} = \psi_i^0 = \theta_i^l, \quad i \in M_l,\; l = k{+}1(1)J.$$

b) Für $i \in M^{(k)}$ ist ψ_i^{k+1} auf jedem $e \in E^{(k+1)}$ linear und $\psi_i^{(k+1)}(x_j) = \delta_{ij}$ in allen Eckknoten j von e, wobei ψ_i^{k+1} auf den von Knoten i ausgehenden Kanten dann nicht eindeutig definiert sein muß.

Eigenschaft a) ergibt sich aus (3). Eigenschaft b) folgt aus (4), weil dadurch die Funktion nur auf Elementen $e \in E_k^* \setminus E_k$ und dort wegen $j \in K(i,k) \subset M_{k+1}$ wie im Falle einer gleichmäßigen Hierarchie verändert wird. Insbesondere ergibt sich für $k+1 = J$ eine Charakterisierung der verfügbaren FEM-Basis:

Verfügbare FEM-Basis Ψ *:*

$$\begin{array}{lcl} \Psi & = & \{\psi_i \,|\, i \in M\}, \qquad \text{wobei} \\ \psi_i & : & \text{auf jedem } e \in E \text{ ist } \psi_i \text{ linear und } \psi_i(x_j) = \delta_{ij} \text{ in allen Eckknoten } j \text{ von } e. \end{array}$$

Somit besteht *supp* ψ_i aus allen aktuellen Elementen, die Knoten i als Eckknoten besitzen. Die ψ_i sind auf den von Knoten i ausgehenden Kanten aktueller Elemente allerdings *nicht notwendig eindeutig* bestimmt und besitzen dort ggf. *Sprungstellen.*

Im weiteren sollen K_Ψ, K_{Φ_a} die Steifigkeitsmatrizen und f_Ψ, f_{Φ_a} die rechten Seiten des FEM-Gleichungssystems bei Verwendung der FEM-Basen Ψ, Φ_a darstellen. Dabei bedeute Assemblierung in der verfügbaren FEM-Basis die Verwendung von entsprechenden Ersatzfunktionalen $\tilde{a}$, $\tilde{f}$ für a, f, welche elementweise mit den Einschränkungen u_e, v_e der Funktionen u, v arbeiten und deshalb auch für die Funktionen der verfügbaren FEM-Basis und deren Linearkombinationen definiert sind:

$$a(u,v) = \sum_{e \in E} a_e(u_e, v_e) =: \tilde{a}(u,v), \quad \forall u, v \in V,$$

$$f(v) = \sum_{e \in E} f_e(v_e) =: \tilde{f}(v), \quad \forall v \in V.$$

Aus obiger Charakterisierung ergibt sich, daß bei der Assemblierung der Steifigkeitsmatrix der verfügbaren FEM-Basis auf den aktuellen Elementen die gleiche Situation vorliegt wie bei einer klassischen Basis aus Hütchenfunktionen. Entsprechende Assemblierungsmoduln können also weitgehend übernommen werden, allerdings ist zusätzlicher Speicherplatz für Matrixelemente mit beteiligten passiven Knoten nötig. Die so entstandene Steifigkeitsmatrix der verfügbaren FEM-Basis Ψ ist anschließend in die der aktiven FEM-Basis Φ_a zu transformieren. Die nachfolgenden Überlegungen zeigen, daß eine relativ problemlose Erzeugung von Φ aus Ψ möglich ist, woraus Φ_a dann durch Weglassen der zu den passiven Knoten gehörenden Basisfunktionen entsteht. Aus (6), (10) ergibt sich zunächst noch mit Θ als unerwünschter Zwischenstation

$$\Phi = \Psi T^{-1} T_a = \Psi \left(T^{(J-1)}\right)^{-1} \cdots \left(T^{(0)}\right)^{-1} T_a^{(0)} \cdots T_a^{(J-1)}. \tag{14}$$

Eigenschaften der Transformationsmatrizen

Die Matrizen $T^{(k)}$, $T_a^{(k)}$ besitzen die Struktur (7). Allgemeine Matrizen $A^{(k)}$ dieser Struktur, also ohne weiter spezifizierte $G^{(k)}$, besitzen folgende leicht nachprüfbaren Eigenschaften, stellen also insbesondere bei festem k bezüglich der Multiplikation eine *abelsche Gruppe* dar:

1. *$A^{(k)}$ ist regulär und $\left(A^{(k)}\right)^{-1}$ besitzt die gleiche Struktur (7), wobei $A^{(k)}$ und $\left(A^{(k)}\right)^{-1}$ sich nur im Vorzeichen des Blockes $G^{(k)}$ unterscheiden, also*

$$\left(A^{(k)}\right)^{-1} = 2I - A^{(k)}.$$

2. *Zwei Matrizen $A^{(k)}$, $B^{(k)}$ der Struktur (7) sind vertauschbar, wobei ihr Produkt wieder die Struktur (7) besitzt, und es gilt*

$$A^{(k)} B^{(k)} = B^{(k)} A^{(k)} = A^{(k)} + B^{(k)} - I. \tag{15}$$

3. *Zwei Matrizen $A^{(k)}$, $B^{(l)}$ der Struktur (7) mit $k \neq l$ sind im allgemeinen nicht vertauschbar, wobei jedoch gilt*

$$A^{(k)} B^{(l)} = A^{(k)} + B^{(l)} - I, \quad \textit{falls} \quad k < l. \tag{16}$$

Aus (15) folgt zunächst, daß sich $T^{(k)}$ in zwei vertauschbare Matrizen der Struktur (7) faktorisieren läßt

$$T^{(k)} \;=\; T_a^{(k)}\, T_p^{(k)} \;=\; T_p^{(k)}\, T_a^{(k)}, \tag{17}$$

wovon $T_a^{(k)}$ bereits oben erklärt wurde und $T_p^{(k)} = T^{(k)} - T_a^{(k)} + I$ auf analoge Weise wie $T_a^{(k)}$ aus $T^{(k)}$ entsteht, indem die $G^{(k)}$, $G^{k+1,l}$ durch entsprechende $G_p^{(k)}$, $G_p^{k+1,l}$ ersetzt werden mit

$$G_p^{k+1,l} = \begin{pmatrix} O & O \\ G_2^{k+1,l} & G_3^{k+1,l} \end{pmatrix}. \tag{18}$$

Die $T_a^{(k)}$ bzw. $T_p^{(k)}$ realisieren die Transformation mittels der Basisfunktionen zu aktiven bzw. passiven Knoten des Levels $k+1$.

Durch Nachrechnen unter Benutzung von (12), (18) läßt sich bestätigen, daß über (16) hinaus folgende Vertauschbarkeitseigenschaften gelten:

$$T_p^{(k)}\, T_a^{(l)} \;=\; T_a^{(l)}\, T_p^{(k)} \;=\; T_p^{(k)} + T_a^{(l)} - I \quad \text{für} \quad k \le l,$$

und wegen obiger Eigenschaft *1.* der inversen Matrix analog

$$\left(T_p^{(k)}\right)^{-1} T_a^{(l)} \;=\; T_a^{(l)} \left(T_p^{(k)}\right)^{-1} \;=\; I - T_p^{(k)} + T_a^{(l)} \quad \text{für} \quad k \le l. \tag{19}$$

Diese Eigenschaften sind Folge des Verschwindens des rechten oberen Blockes in den Matrizen $G^{k+1,l}$ und damit Folge der Regel, aktive Knoten nur auf aktivierbare Kanten, also Kanten mit zwei aktiven Vätern zu setzen.

Basistransformation $\Phi_a = \Psi\, Q$

Mit (17) und (19) kann (14) vereinfacht werden. Damit ergibt sich schließlich die zu nutzende Basistransformation

$$\Phi_a \;=\; \Phi\, I_a \;=\; \Psi\, \left(T_p^{(J-1)}\right)^{-1} \cdots \left(T_p^{(0)}\right)^{-1} I_a \;=\; \Psi\, Q\,. \tag{20}$$

Für die $\left(T_p^{(k)}\right)^{-1}$ gilt im allgemeinen wieder obige Eigenschaft *3.*, so daß die Basistransformation (20) tatsächlich levelweise von oben nach unten realisiert werden muß.

Transformation von Steifigkeitsmatrix und rechter Seite

Es wird nun der Zusammenhang beider Steifigkeitsmatrizen betrachtet, wenn eine Basistransformation $\Phi_a = \Psi\, Q$ wie in (20) vorliegt. Ist n die Anzahl der aktiven Knoten, so führt die übliche Erklärung der Steifigkeitsmatrix auf die für beliebige $\underline{u}, \underline{v} \in \mathsf{R}^n$ gültige Beziehung

$$(K_{\Phi_a}\underline{u}, \underline{v}) = a(\Phi_a\underline{u}, \Phi_a\underline{v}) = \tilde{a}(\Phi_a\underline{u}, \Phi_a\underline{v}) = \tilde{a}(\Psi\, Q\underline{u}, \Psi\, Q\underline{v}) = (K_\Psi\, Q\underline{u}, Q\underline{v}) = (Q^\top K_\Psi\, Q\underline{u}, \underline{v}),$$

also folgt für die Matrizen und analog für die rechten Seiten

$$K_{\Phi_a} \;=\; Q^\top K_\Psi\, Q\,, \quad f_{\Phi_a} \;=\; Q^\top f_\Psi\,.$$

Dann ergibt die Transformation (20) den Zusammenhang

$$K_{\Phi_a} \;=\; I_a^\top \left(T_p^{(0)}\right)^{-\top} \cdots \left(T_p^{(J-1)}\right)^{-\top} K_\Psi \;\left(T_p^{(J-1)}\right)^{-1} \cdots \left(T_p^{(0)}\right)^{-1} I_a\,, \tag{21}$$

die Matrix K_Ψ kann somit auch durch einfache levelweise Transformationen in K_{Φ_a} überführt werden, die rechte Seite analog,

$$\begin{aligned} K_\Phi^{(J)} &= K_\Psi, \\ K_\Phi^{(k)} &= \left(T_p^{(k)}\right)^{-\top} K_\Phi^{(k+1)} \left(T_p^{(k)}\right)^{-1}, \quad k = J-1(-1)\,0, \\ K_{\Phi_a} &= I_a^\top K_\Phi^{(0)}\, I_a. \end{aligned} \tag{22}$$

Nutzung elementarer Transformationsmatrizen

Soll dieses Vorgehen durch Abarbeitung der hierarchischen Liste **HiNo** vom oberen Ende her realisiert werden, müßten die Knoten in dieser Liste immer nach Levels geordnet sein, was für die dynamische Verwaltung der Liste beim Umbau der Hierarchien zusätzlichen Aufwand in Form von Umspeicherungen bedeutete. Durch weitere Faktorisierung der Transformationsmatrizen $T_p^{(k)}$ bis hin zu den elementaren Transformationen einzelner Basisfunktionen und anschließendes geeignetes Vertauschen dieser kann nun gezeigt werden, daß das Prinzip „Vater steht vor Sohn" zur Ordnung von **HiNo** ausreichend ist für eine schrittweise Transformation der Steifigkeitsmatrix K_Ψ in K_{Φ_a} an Hand von **HiNo**.

Zu jedem Knoten $i \in M \setminus M_0$ werde eine *elementare Transformationsmatrix* T_i erklärt, und zwar unterscheide sie sich von der Einheitsmatrix nur dadurch, daß in der entsprechend Reihenfolge (1) zu Knoten i gehörenden Zeile der Wert $-\frac{1}{2}$ in den Spalten der beiden Vaterknoten von i stehe. Für $i \in M_{k+1}$ gehören die T_i der abelschen Gruppe von Matrizen der Struktur (7) an. Die $T^{(k)}$ usw. können wegen (15) dann folgendermaßen faktorisiert werden, wobei die Reihenfolge der Faktoren keine Rolle spielt

$$T^{(k)} \;=\; \prod_{i \in M_{k+1}} T_i\,, \qquad T_a^{(k)} \;=\; \prod_{i \in M_{k+1,a}} T_i\,, \qquad T_p^{(k)} \;=\; \prod_{i \in M_{k+1,p}} T_i\,.$$

Damit ist die Transformation (20) der verfügbaren in die aktive FEM-Basis auch darstellbar als

$$\Phi_a \;=\; \Psi \left(\prod_{i \in M_{J,p}} T_i^{-1} \right) \cdots \left(\prod_{i \in M_{1,p}} T_i^{-1} \right) I_a\,, \tag{23}$$

die Reihenfolge der T_i^{-1} bei Anwendung auf die Basis ist dabei eine mögliche Realisierung der Knotenordnung „Sohn vor Vater".

Es kann leicht nachgeprüft werden, daß zwei elementare Transformationen T_i, T_j genau dann vertauschbar sind, wenn weder i Vater von j noch j Vater von i ist. Deshalb kann in (23) die Reihenfolge der T_i^{-1} weiter verändert werden, solange die bestehende Ordnung „Sohn vor Vater" nicht gestört wird.

$$\Phi_a \;=\; \Psi \left(\prod_{\substack{i \in M_{(p)} \\ \text{Sohn vor Vater}}} T_i^{-1} \right) I_a\,. \tag{24}$$

Somit kann der Algorithmus (22) zur Transformation der Steifigkeitsmatrix ersetzt werden durch

$$
\begin{aligned}
& & K_\Phi^{(NNO)} &= K_\Psi\,, \\
\textsf{Zyklus} & & \text{für } m &= NNO-1\,(-1)\,NNOH, \\
& & \text{mit } \ i &= \mathbf{HiNo}\,(1,m+1) \quad \text{(Knotennummer)}: \\
& & K_\Phi^{(m)} &= \begin{cases} K_\Phi^{(m+1)} & \text{für i aktiv,} \\[1ex] T_i^{-\top} K_\Phi^{(m+1)}\, T_i^{-1} & \text{für i passiv,} \end{cases} \qquad (25) \\
& & K_{\Phi_a} &= I_a^\top K_\Phi^{(NNOH)}\, I_a\,,
\end{aligned}
$$

wobei NNO die Anzahl der Knoten und $NNOH$ die Anzahl der Hauptknoten ist. Für die rechten Seiten ist der Algorithmus entsprechend zu modifizieren.

6 Beispiel

Zur Demonstration der beschriebenen Technologie werde folgendes Beispiel betrachtet. Das Gebiet Ω sei ein Polygon, sein Rand $\partial\Omega$ werde durch den Polygonzug mit den Eckknoten

$$(0,0), \quad (1,0), \quad (0.8,0.8), \quad (0,1), \quad (-0.5,0.7), \quad (-1,0), \quad (-1,-1), \quad (0,-1), \quad (0,0)$$

gebildet. In Ω ist ein Dirichletproblem für $u = u(x,y)$ zu lösen, wobei (r,φ) die Polarkoordinaten darstellen:

$$
\begin{aligned}
\Delta\, u &= 0 & & \text{in } \Omega, \\
u &= r^{\frac{2}{3}} \sin\left(\tfrac{2}{3}\varphi\right) & & \text{auf } \partial\Omega.
\end{aligned}
$$

Die exakte Lösung lautet

$$u = r^{\frac{2}{3}} \sin\left(\tfrac{2}{3}\varphi\right) \qquad\qquad \text{in } \overline{\Omega}.$$

Die numerische Lösung erfolgte auf einem Transputersystem mit 16 Prozessoren. Die FE-Gleichungssysteme für die Näherungen der Funktionswerte in den aktiven Knoten wurden mit einer vorkonditionierten parallelen Realisierung der Methode der konjugierten Gradienten gelöst, wobei der parallelisierte Yserentant-Vorkonditionierer ohne Grobgitterlöser Verwendung fand. Der Iterationsprozeß wurde bei einer Fehlerreduktion um den Faktor $\varepsilon = 10^{-15}$ in einer zur energetischen Norm äquivalenten Norm abgebrochen. In die Basen für die FEM wurde immer das zusätzliche Level von Basisfunktionen mit einbezogen, welches für die Geschwindigkeit vorgesehen ist, auch auf dem Hauptnetz. In den Abbildungen sind die aktuellen 6-Knoten-Elemente deshalb durch ihre vier Teildreiecke dargestellt. In Abb. 1 ist links das Hauptnetz für die FE-Diskretisierung zu sehen, in Abb. 4 oben die Isolinien der exakten Lösung ($u_{min} = 0$, $u_{max} = 0.899$), unten der absolute Fehler e der FEM-Näherung auf dem Hauptnetz ($e_{min} = -1.81 \cdot 10^{-2}$, $e_{max} = 0$). Der betragsmaximale Fehler wird wie erwartet in der Nähe der Ecksingularität angenommen.

In Abb. 5 sind Netz und absoluter Fehler bei Nutzung des weiter oben beschriebenen Verfeinerungsmoduls NetFiner zu sehen, wobei zum Hauptnetz zusätzlich bis zu vier weitere Levels erzeugt wurden. Das genutzte heuristische Verfeinerungskriterium orientierte sich an der Gestalt der Näherungslösung und signalisierte eine Verfeinerung der Kante, wenn die Richtungsableitung der Näherungslösung auf der Kante einen Steuerparameterwert überschritt. Als Parameterwerte wurden nacheinander die Werte 1.0, 1.5, 2.0, 2.5 verwendet (Verfeinerungstechnik 1).

In Abb. 6 sind in der Ausschnittsvergrößerung des Netzes die aktiven Knoten als schwarze Punkte markiert. Dabei ist eine teilweise unvertretbare Richtungsabhängigkeit der Verdichtung aktiver Knoten zu beobachten, die daraus resultiert, daß in das Verfeinerungskriterium nur Informationen von der Kante selbst einfließen. Übliche Fehlerschätzer hingegen integrieren in der Regel die Fehleranteile, zum Beispiel über die an der Kante anliegenden Elemente. Ist der Fehler in einem Element hinreichend groß, führt das zur Verfeinerung aller drei Elementkanten in dem Sinne, daß in deren drei Mittelpunkte neue Basisfunktionen gesetzt werden.

Eine derartige elementorientierte Verfeinerungstechnik auf Grundlage kantenorientierter Verfeinerungsentscheidungen wird als Verfeinerungstechnik 2 realisiert. Diese Technik läuft zweistufig ab: in der ersten Stufe wird die Verfeinerung wie bei Technik 1 durchgeführt, in der zweiten Stufe werden zusätzlich in allen aktuellen Elementen der ersten Stufe mit drei aktiven Eckknoten und einem aktiven Kantenmittelknoten auch die übrigen beiden Kantenmittelknoten aktiviert, falls sie nicht schon aktiv sind. Diese zusätzliche Aktivierung ist unbedingt und betrifft auch nur aktuelle Elemente, die in der ersten Stufe wegen des aktiven Kantenmittelknotens sowieso verfeinert wurden. Das maximale Level des Netzes ändert sich durch die zweite Stufe nicht, die Gesamtzahl der Knoten sowie die Zahl der aktiven Knoten vergrößern sich jedoch. Netz und Fehler bei vier Schritten Verfeinerungstechnik 2 mit dem gleichen Verfeinerungskriterium und den gleichen Parameterwerten wie bei Technik 1 sind in Abb. 7 dargestellt, der Netzausschnitt in Abb. 8 zeigt eine Verdichtung der aktiven Knoten um die Singularität, die eher der Richtungsunabhängigkeit nahekommt. Bemerkenswert sind die ausgeprägten Oszillationen des Fehlers entlang der Levelgrenzen.

Nachfolgende Tabelle gibt eine Übersicht über Resultate der Berechnungen. V1, V2 bezeichnen die beiden Verfeinerungstechniken, V3 steht für die gleichmäßige Verfeinerung des Hauptnetzes um vier Levels. L1, L2 usw. bezeichnen, bis zu welchem Level mit obigen Parameterwerten gerechnet wurde, L1 ist dabei das Level des Hauptnetzes. Größenordnungsmäßig werden die Fehler an der Grenze zum gröbsten Level durch den maximalen Fehler charakterisiert, der (negative) minimale Fehler entstammt der unmittelbaren Umgebung der Singularität. Die Anzahl der CG-Iterationen ist hinsichtlich obiger Genauigkeit von $\varepsilon = 10^{-15}$ gemeint.

Variante	aktive Knoten	alle Knoten	Iterationen	Fehler e_{min}	Fehler e_{max}
V1-L1	565	565	74	$-1.81 \cdot 10^{-2}$	0
V1-L2	594	694	74	$-1.30 \cdot 10^{-2}$	$+7.90 \cdot 10^{-4}$
V1-L3	602	724	74	$-1.12 \cdot 10^{-2}$	$+8.63 \cdot 10^{-4}$
V1-L4	610	754	74	$-1.06 \cdot 10^{-2}$	$+8.91 \cdot 10^{-4}$
V1-L5	618	784	74	$-1.04 \cdot 10^{-2}$	$+9.01 \cdot 10^{-4}$
V2-L1	565	565	74	$-1.81 \cdot 10^{-2}$	0
V2-L2	695	755	76	$-1.15 \cdot 10^{-2}$	$+1.45 \cdot 10^{-4}$
V2-L3	739	834	76	$-7.27 \cdot 10^{-3}$	$+3.27 \cdot 10^{-4}$
V2-L4	769	892	77	$-4.62 \cdot 10^{-3}$	$+4.79 \cdot 10^{-4}$
V2-L5	799	950	77	$-2.94 \cdot 10^{-3}$	$+5.43 \cdot 10^{-4}$
V3-L5	131905	131905	154	$-2.90 \cdot 10^{-3}$	$+2.54 \cdot 10^{-7}$

Literatur

[1] J.H. Bramble, J.E. Pasciak, J. Xu
Parallel multilevel preconditioners. Mathematics of Computation, 55(191):1-22, 1990.

[2] U. Groh
Lokale Realisierung von Vektoroperationen auf Parallelrechnern. Preprint SPC 94_2, Preprint–Reihe der Chemnitzer DFG-Forschergruppe „Scientific Parallel Computing“, 1994.

[3] U. Groh, S. Meinel, A. Meyer
On the Numerical Simulation of Coupled Transient Problems on MIMD Parallel Systems. In: Proceedings of the Tenth GAMM–Seminar Kiel 1994, Notes on Numerical Fluid Mechanics, to appear, Vieweg 1995.

[4] G. Haase, U. Langer, A. Meyer
Parallelisierung und Vorkonditionierung des CG – Verfahrens durch Gebietszerlegung. In: G. Bader, R. Rannacher, G. Wittum (Hrsg.); Numerische Algorithmen auf Transputersystemen, Teubner-Skripten zur Numerik, Teubner–Verlag, Stuttgart 1993.

[5] P. Leinen
Ein schneller adaptiver Löser für elliptische Randwertprobleme auf Seriell- und Parallelrechnern. Dissertation, Universität Dortmund, 1990.

[6] A. Meyer
A parallel preconditioned conjugate gradient method using domain decomposition and inexact solvers on each subdomain. Computing, 45:217-234, 1990.

[7] U. Rüde
Mathematical and Computational Techniques for Multilevel Adaptive Methods. Series: Frontiers in Applied Mathematics, Vol. 13, SIAM, Philadelphia, 1993.

[8] R. Verfürth
A review of a posteriori error estimation and adaptive mesh refinement techniques. Report, Inst. f. Angew. Mathem., Univ. Zürich, 1993

[9] H. Yserentant
On the multilevel-splitting of finite element spaces. Numer. Math., *49*, 379–412 (1986).

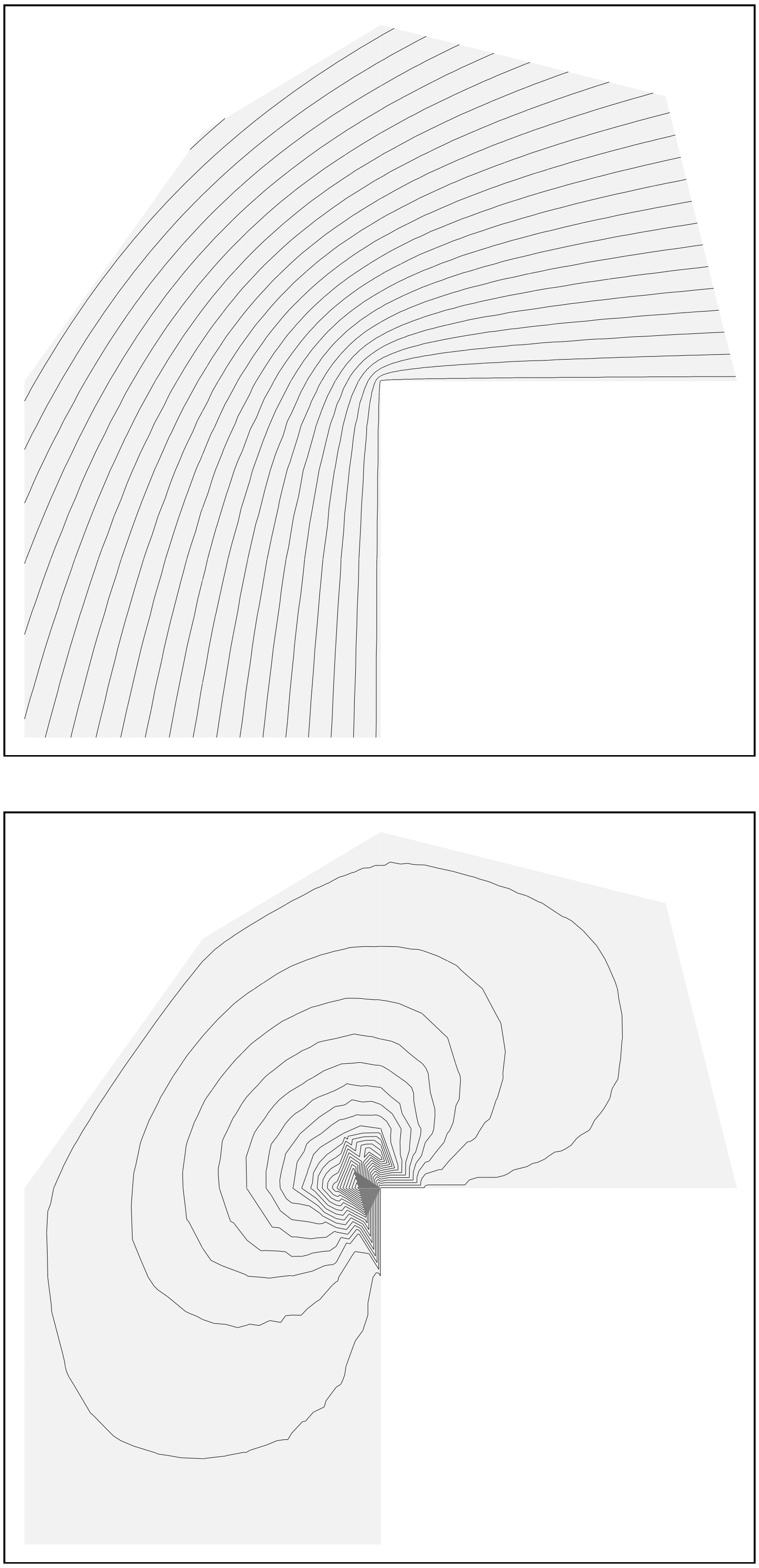


Abbildung 4: Exakte Lösung und Fehler der Näherungslösung auf dem Hauptnetz

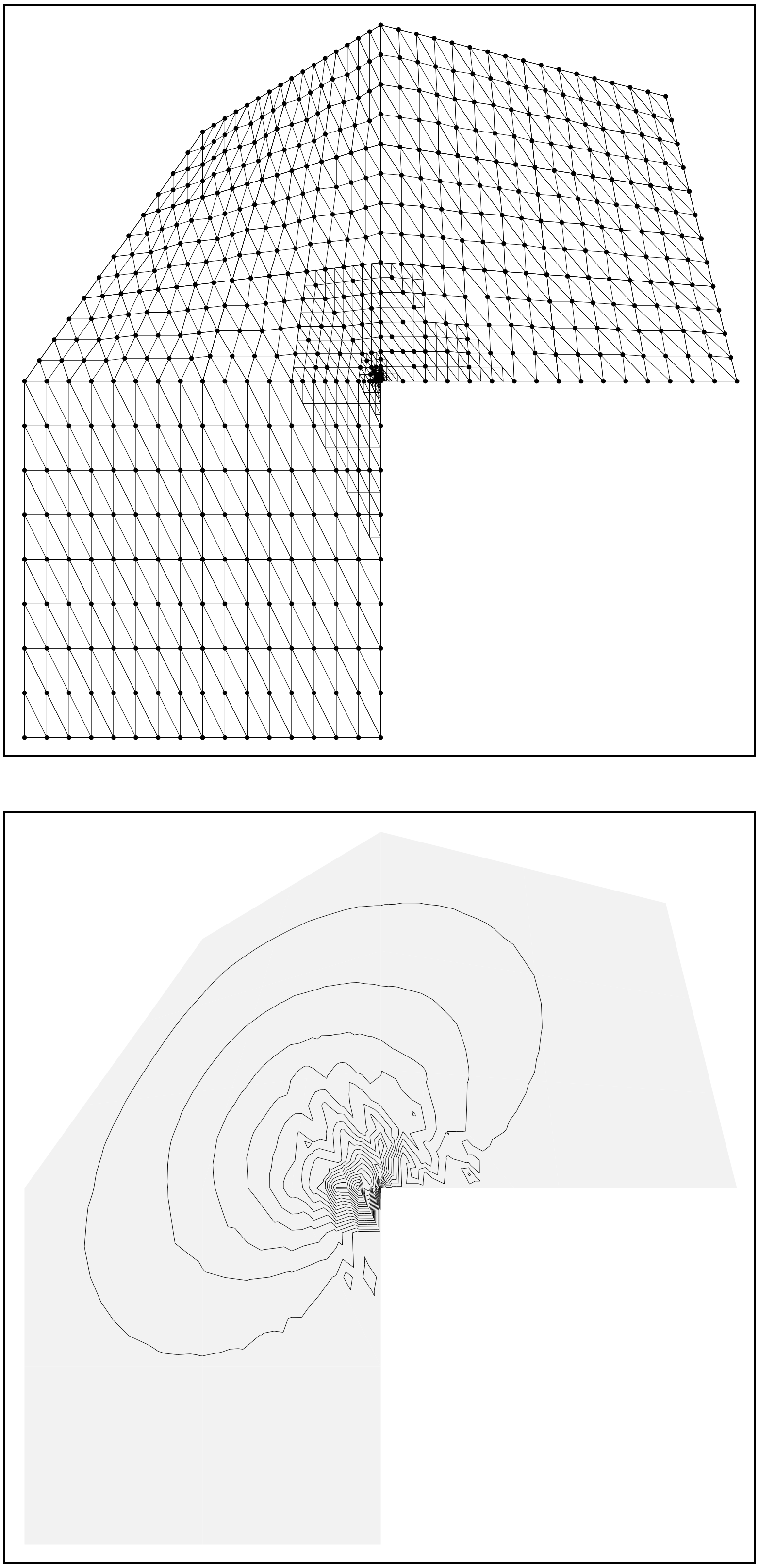


Abbildung 5: Netz und Fehler, Verfeinerungstechnik 1, maximal 4 Zusatzlevels

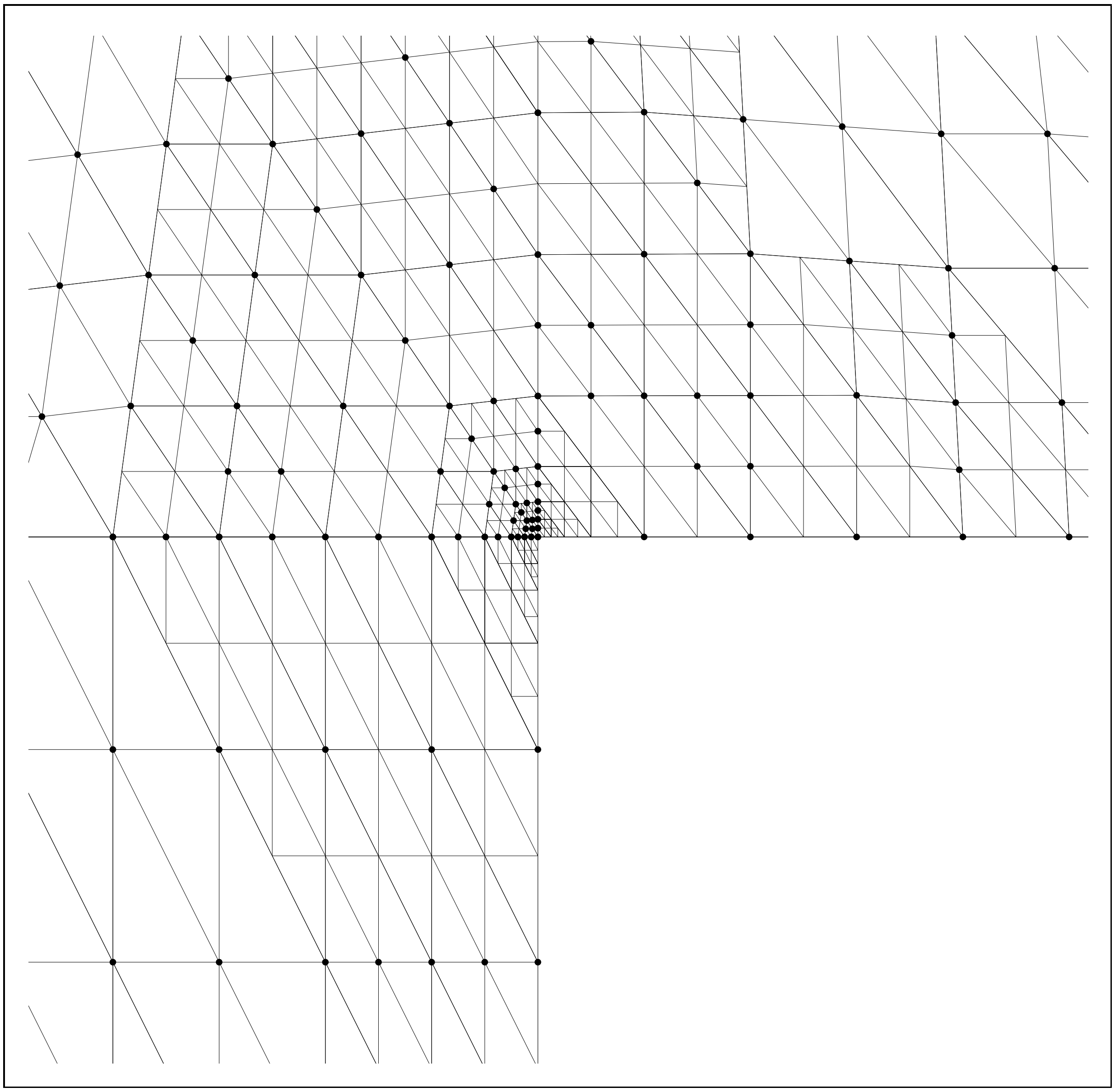


Abbildung 6: Netzausschnitt, Verfeinerungstechnik 1, maximal 4 Zusatzlevels

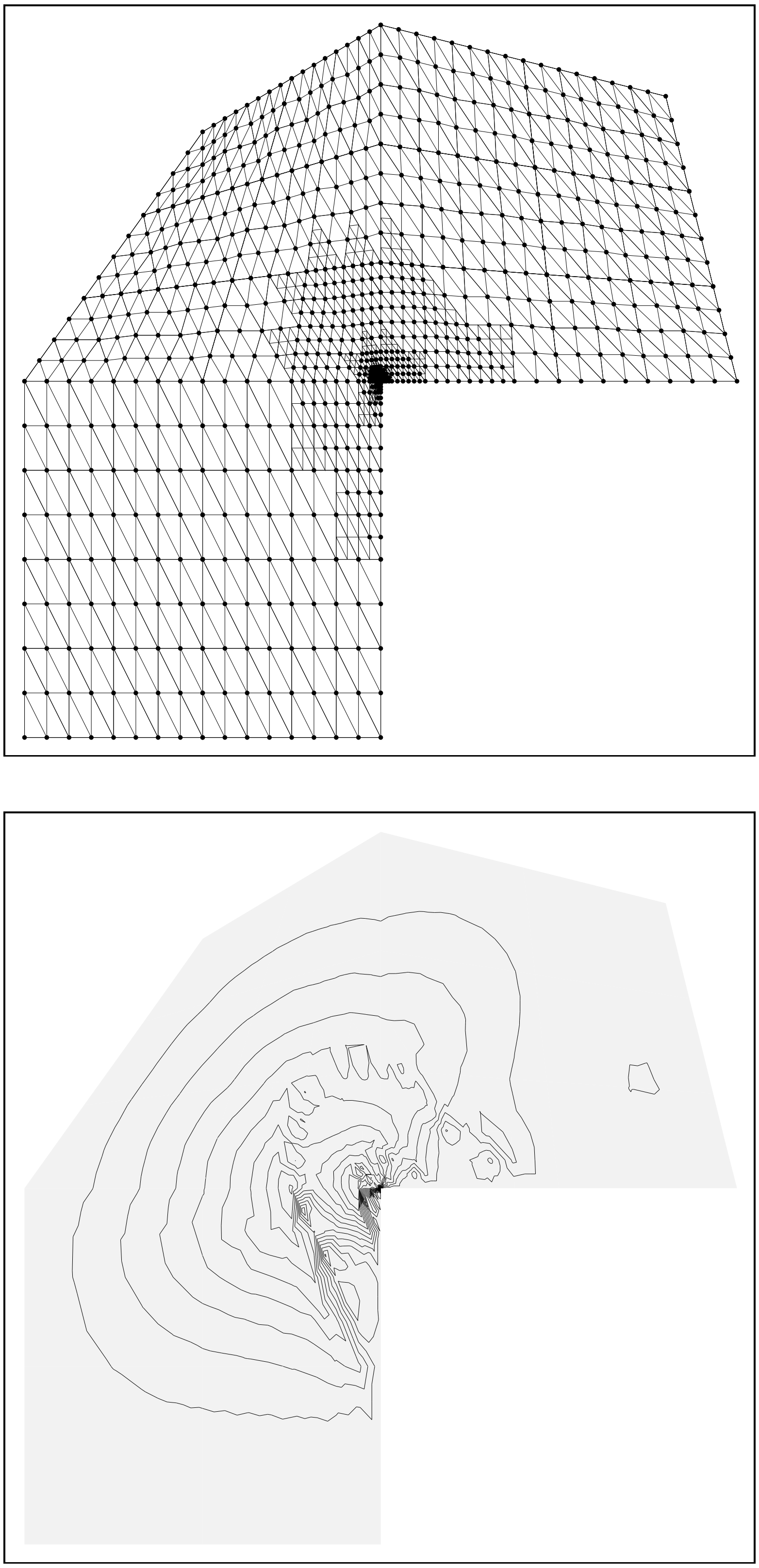


Abbildung 7: Netz und Fehler, Verfeinerungstechnik 2, maximal 4 Zusatzlevels

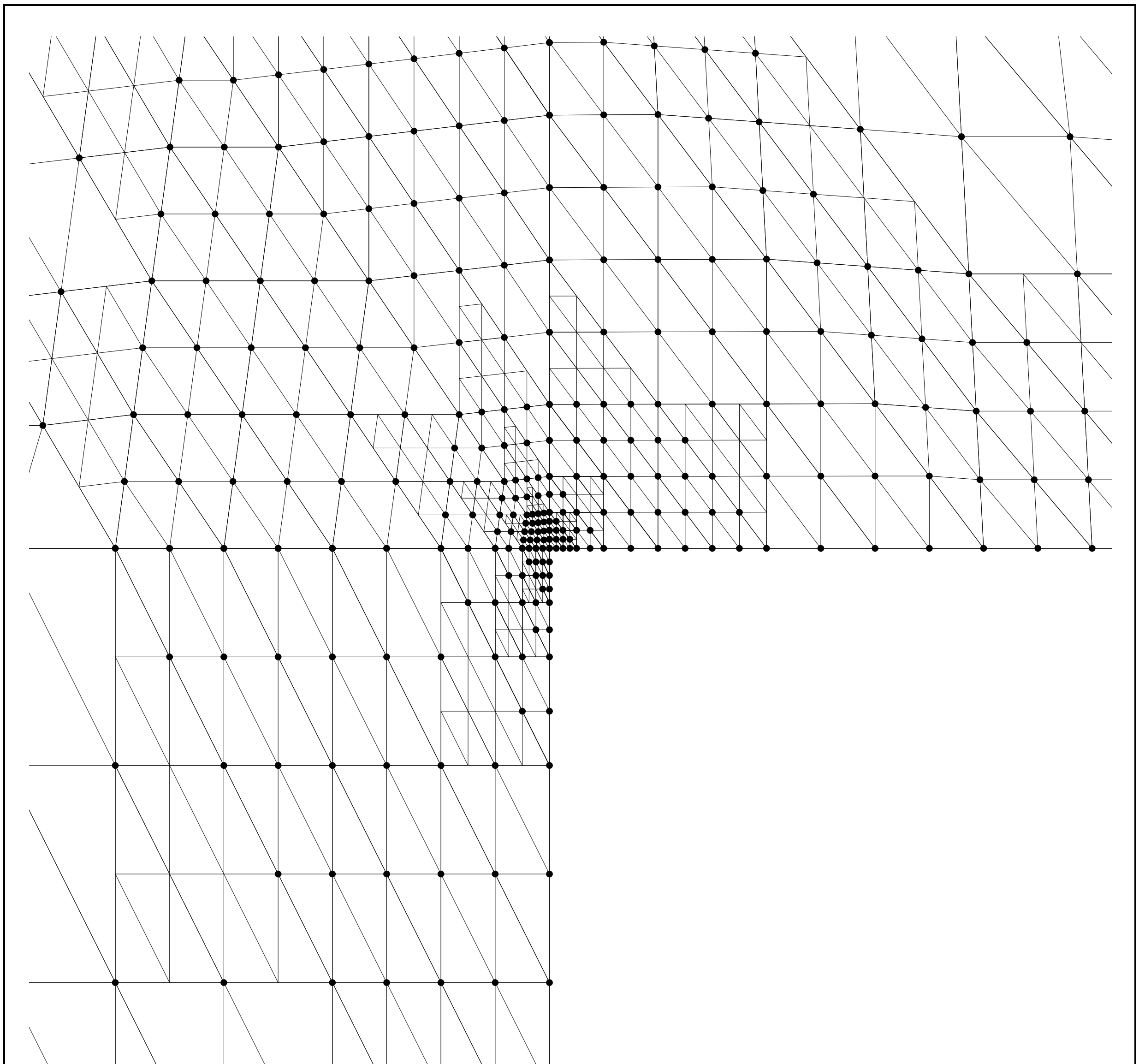


Abbildung 8: Netzausschnitt, Verfeinerungstechnik 2, maximal 4 Zusatzlevels