Allgemeine komponenten–orientierte Finite–Element–Modellierung

Markus König und Peter Milbradt

Kurzfassung: *In einer Studie am Institut für Bauinformatik der Universität Hannover beschäftigte man sich damit, Finite Elemente universell und komponenten–orientiert zu beschreiben, um dadurch eine gute allgemeine Einsetzbarkeit in vielen unterschiedlichen Bereichen zu ermöglichen. Hauptaugenmerk lag bei der Entwicklung darauf, daß eine einheitliche Formulierung für verschiedene Finite Elemente definiert wurde, um diese sehr einfach und übersichtlich anwenden zu können.*

Einleitung

Finite Elemente werden in vielen unterschiedlichen Bereichen des Bauwesens eingesetzt und gewinnen immer mehr an Bedeutung. Bei den meisten Anwendungen werden sehr häufig nur spezielle Geometrien und Topologien für Finite Elemente und ihre Netze verwendet. Eine gute Wiederverwendbarkeit und eine allgemeine Formulierung wird dadurch kaum gewährleistet. Aufgrund einer Studienarbeit am Institut für Bauinformatik der Universität Hannover, in der Simplexe beliebiger Ordnung als Finite Elemente definiert und getestet wurden, beschäftigt man sich nun mit einer allgemeinen komponenten–orientierten Finite–Element–Modellierung.

1 Mathematische Grundlagen

Definition:

Eine Menge heißt kompakt, wenn jede Folge $\{x_n\}$, $(x_n \in A)$, eine konvergente Teilfolge $\{x_{nj}\}$ mit Grenzpunkt $x \in A$ enthält.

Eigenschaften:

1. Jede Kompakte Menge ist abgeschlossen.
2. Jede abgeschlossene Teilmenge einer Kompakten Menge ist wieder kompakt.
3. Die Vereinigung endlich vieler und der Durchschnitt beliebig vieler Kompakter Mengen ist kompakt.

2 Finites Element

Ein grundlegendes Merkmal der Methode der Finiten Elemente ist es, daß ein zu untersuchendes Gebiet in geeignete Teilgebiete zerlegt wird. Die Dimension und Gestalt der Finiten Elemente hängt von der Dimension des Raumes, vom formulierten Problem und der gesuchten Lösung ab, die durch eine gewählte Interpolation der Werte an den Freiheitsgraden bestimmt wird.

can.–Ing. M. König und Dr.–Ing P. Milbradt, Institut für Bauinformatik, Universität Hannover

2.1 Attribute

Nach gründlichem Studium der Literatur und eigenen Ideen, gelangte man zu der Auffassung, daß ein Finites Element sich durch drei wesentliche Attribute beschreiben läßt:

- Durch eine Kompakte Menge von Elementen eines normierten Raumes. Die Kompakte Menge ist abgeschlossen und beschränkt. Sie beschreibt ein bestimmtes Gebiet des normierten Raumes und stellt die Geometrie des Finiten Elements da.
- Durch eine Menge von Interpolationsfunktionen. Mit Hilfe des Interpolationsraums wird die Lösung des gegebenen Problems im Finiten Element beschrieben.
- Durch eine Menge von Werten an bestimmten Knotenpunkten der Kompakten Menge. Diese werden allgemein Freiheitsgrade genannt. Sie dienen zum Setzen von Randbedingungen und enthalten nach Beendigung der Rechnung die Lösungswerte an den Knotenpunkten.

Um ein Finites Element aufstellen zu können, muß man folgende Beziehungen der Attribute untereinander beachten:

- Der Interpolationsraum ist auf der Kompakten Menge definiert.
- Die Menge der Freiheitsgrade hängt vom Interpolationsraum ab.

Mit Hilfe dieser Definition kann man ein Finites Element vollständig und allgemeingültig beschreiben.

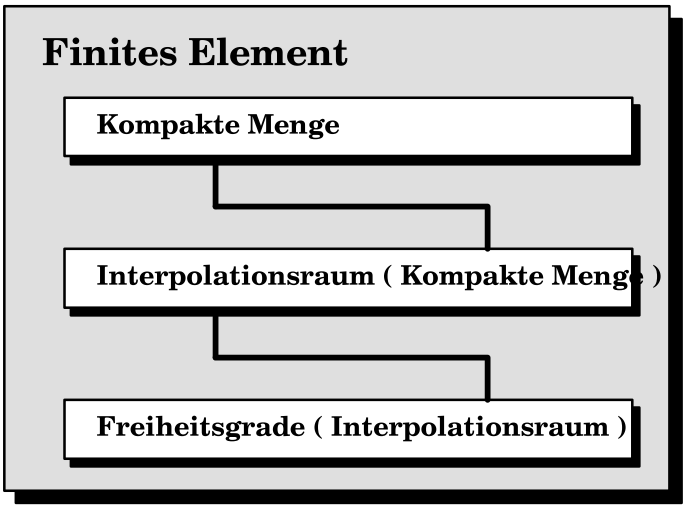


Bild 1: Definition eines Finiten Elements

Es stellte sich heraus, daß ein solches Finites Element nur dann erzeugt werden kann, wenn die drei definierten Attribute parametrisch sind. Das bedeutet, daß bestimmte Komponenten der Schnittstelle bei allen verwendeten Geometrien, Interpolationsräumen und Freiheitsgraden übereinstimmen müssen. Bei der Implementierung wurde daher die Programmiersprache C++ verwendet, die solche Möglichkeiten bietet.

2.1.1 Kompakte Menge

Auf dem Weg zu einer allgemeinen Definition wurden bei den bisherigen Betrachtungen zwei Arten von Kompakten Mengen untersucht. Dieses waren zum einen Simplexe beliebiger Ordnung und zum anderen Rechtecke bzw. Vierecke. Man erhielt durch diese Geometrien Kenntnisse darüber, welche Attribute und Methoden eine Geometrie besitzen muß, um sie als Kompakte Menge für ein Finites Element verwenden zu können. Dabei stellte sich als wichtigste Eigenschaft heraus:

- Eine Kompakte Menge muß aus einer Menge von Punkten eines linearen Raumes bestehen, für die eine Metrik definiert ist.

Mit wenigen eindeutigen Attributen, Methoden und einigen Standardoperatoren konnten diese Geometrien als Kompakte Mengen für das definierte Finite Element verwendet werden. Dabei wurde eine im Institut vorhandene Implementierung von Simplexen, die im Rahmen einer Diplomarbeit [10] entwickelt wurde, ohne weitere Schwierigkeit verwendet werden.

2.1.2 Interpolationsraum

Für die genannten Geometrien wurden im nächsten Schritt geeignete Interpolationsfunktionen durch die Knoten auf den Kanten der Elemente erzeugt. Bei Rechteck– bzw. Viereckelementen wurden für die Interpolationsfunktion Lagrang'sche Polynome der *SERENDIPITY*–Klasse bis zum Grad drei verwendet. Für die Simplex–Elemente der Ordnung n entschied man sich für eine allgemeine Formulierung der Interpolationsfunktion bis zum Grad zwei, die in einer Studienarbeit [5] am Institut entwickelt wurde.

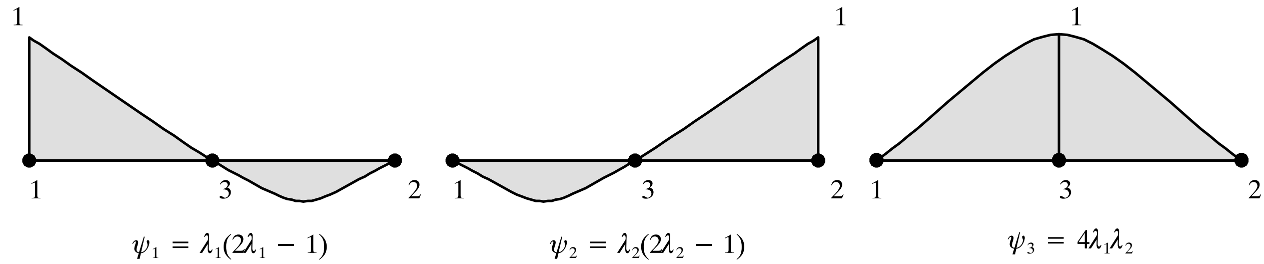


Bild 2: Interpolationsfunktionen vom Grad zwei für ein Simplex–Element der Ordnung eins

2.1.3 Freiheitsgrade

Freiheitsgrade werden in den meisten Fällen nur an den Knoten der Interpolationsfunktionen definiert. Sie enthalten skalare Werte, die nach dem Rechenlauf zur Interpolation der Lösung dienen. Da ein Interpolationsknoten mehrere Freiheitsgrade besitzen kann, wurde die Menge der Werte der Freiheitsgrade in Form einer Matrix innerhalb einer eigenen Klasse gespeichert.

2.2 Implementierung

Im folgenden Quellcode wird ein Einblick in die Erzeugung von Finiten–Element–Objekten begeben. Es wird exemplarisch ein Simplex–Element der Ordnung zwei (Dreieck) vereinbart.

```
simplex triangle (number_of_points=3, points);
interpolation_of_simplex interpol (triangle, degree=2);
degrees_of_freedom < interpolation_of_simplex > dof  (interpol, number_of_degrees_per_node=1, values
finit_element < simplex, interpolation_of_simplex, degrees_of_freedom
< interpolation_of_simplex > > fe (triangle, interpol, dof);
```

Bild 3: Erzeugen eines Dreieckselements mit der Interpolationsordnung zwei und je einem Freiheitsgrad pro Knoten

3 Zellkomplex als Finites–Element–Netz

Bei der überwiegenden Anzahl von Anwendungen werden Finite Elemente in Netzen verwendet. Um ein Finites–Element–Netz zu definieren, beschränkte man sich zuerst auf Finite Elemente vom gleichen Typ. Man stellte fest, daß ein Netz durch eine Menge von Elementen und durch eine Menge von Relationen zwischen diesen Elementen eindeutig bestimmt wird. Diese Eigenschaft wird durch einen Zellkomplex exakt erfüllt. Demzufolge ist eine Finites–Element–Netz ein Zellkomplex, der aus Finiten Elementen besteht. Die Theorie der Zellkomplexe wird in dem Forumsbeitrag “Entwicklung von Komponenten für den Einsatz bei bauingenieurspezifischen Problemen” ausführlich beschrieben. Mit der im Institut vorhandenen Klassenbibliothek wurden sehr einfach Netze erzeugt, die den bisherigen Anforderungen genügen.

```
complex < finit_element < simplex, interpolation_of_simplex, degrees_of_freedom
< interpolation_of_simplex > > > fenet;
finit_element < simplex, interpolation_of_simplex, degrees_of_freedom
< interplation_of_simplex > >fe = new finit_element < simplex, interpolation_of_simplex,
degrees_of_freedom < interplation_of_simplex > > [number_of_elements];
for  (long  i=0; i<number_of_elements; i++)
          fenet = fenet + fe[i];
```

Bild 4: Erzeugen eines Finiten–Element–Netzes mit Hilfe eines Zellkomplexes

4 Anwendungsbeispiele

An ausgewählten Beispielen wurde das Arbeiten mit der aufgestellten Klassenbibliothek getestet. Hierbei stellte sich heraus, daß ein Programm so allgemein aufgebaut werden kann, das ein Verwenden von beliebigen Finiten Elementen und Finiten–Element–Netzen ohne weitere Probleme möglich ist. Als erstes wurde die Interpolation einer Funktion auf einem Gebiet implementiert.

4.1 Interpolation einer Funktion

Mit Hilfe der Interpolation einer Funktion auf einem beliebigen Finiten–Element–Netz wurde überprüft, ob die definierten Interpolationsfunktionen auch die exakten Werte liefern. Es wurden Funktionsterme verschiedener Ordnung definiert und auf Dreiecks– und Rechtecksnetzen mit verschiedenen Ansatzordnungen interpoliert. Hierzu wurden die Werte an den Freiheitsgraden exakt berechnet und den Elementen übergeben. Danach wurde ein gleichmäßiges Raster über das Finite–Element–Netz gelegt und für die einzelnen Rasterpunkte die Funktionswerte interpoliert. Es zeigte sich, daß Funktionen der Ordnung n durch Interpolationsfunktionen der Ordnung n genau berechnet werden können.

```
data.function = hyperboloid
fen.forallcells (set_degrees_of_freedom,&data);
...
result.x = grid_point;
result.value = new vector (0);
fen.forallcells (interpolation_of_point, &result);
cout  <<  result.value;
...
```

```
int interpolation_of_point (fe& elem, void* v)
{
  argument *arg = (argument) v;
  if(!elem.is_inside(*arg->x))
      return 0;
  for(longi=0;i<elem.number_of_degrees_per_node();i++)
      for (longj=0;j<elem.number_of_interpolation_functions();j++)
          (*arg->value)[i]=(*arg->value)[i]+
          elem.interpolation_functions(*arg->x)[j]*
          elem.value_of_degrees_of_freedom()[j][i];
  return 1;
}
```

Bild 5: Interpolation eines Hyperboloids

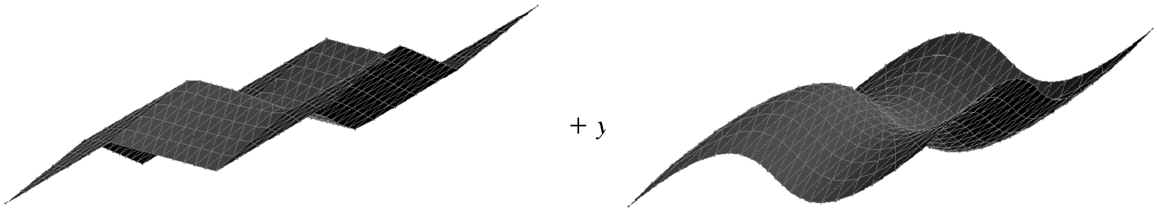
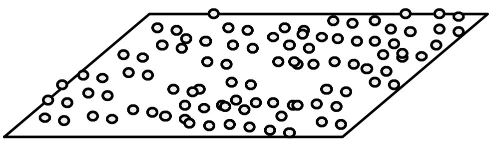


Bild 6: Hyperboloid mit linearen und kubischen Interpolationsfunkionen

4.2 Generieren eines Modellnetzes

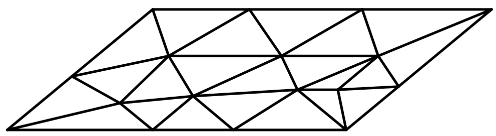
Als weiteres Beispiel wurde das Erzeugen von Modellnetzen ausgewählt. Man nimmt an, daß eine Menge von Meßdaten für ein bestimmtes Gebiet vorhanden sind. Die Triangulation dieser Meßdaten (Meßdatennetz) wird charakerisiert, zum einen durch die eingesetzten Meßverfahren und zum anderen durch die örtlichen Gegebenheiten bei den Messungen. In den meisten Fällen ist das erzeugte Meßdatennetz für die weiteren Berechnungen ungeeignet. Da numerische Verfahren hohe Anforderungen bzgl. der Regularität an das Gitternetz stellen, wird zunächst ein geeignetes Basisnetz generiert. Dem so erzeugten topologischen Modell wird im nächsten Schritt die geometrische Information durch Interpolation aus dem Meßdatennetz aufgeprägt. Mit dem entstandenen Modellnetz kann nun weitergearbeitet werden. Die Abbildung 7 zeigt die einfache und übersichtliche Realisierung mit Hilfe der entwickelten Klassenbibliothek.

lineares Vermaschen
⇒ Finites- -Element- -Netz *fenet1*



Meßdatennetz

Erzeugen einer Trägertopologie
⇒ Finites- -Element- -Netz *fenet2 (z=0)*

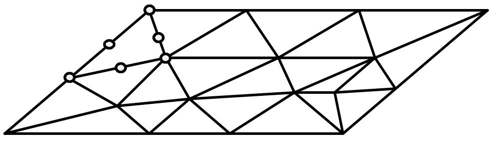


Basisnetz

Setzen der Freiheitsgrade

```
fenet fenet1, fenet2;
...
args.function      = fenetinterpolation;
args.function_args = fenet1;
fenet2.forallcells( set_degrees_of_freedom, &args );
```

Interpolieren
⇒ Finites- -Element- -Netz *fenet2* (z = f(*fenet1*))



Modellnetz

```
vector fenetintpolation (complex <felem>* fenet, vector* x)
{
  arguments result;

  result.x = x;
  result.value = new vector();

  if (x->dim() == fenet->coord_dim())
  fenet->forallcells (interpolatio_of_point, &result);

  return result.value;
}
```

Bild 7: Generieren eines Modellnetzes

5 Ausblick

Gegenstand der laufenden Untersuchungen ist die Realisierung einer Finiten–Element–Approximation für Differentialgleichungen, die auf dem definierten Finiten–Element–Modell basiert. Es wird angestrebt eine universelle Methode zum Aufbau der Elementsteifigkeitsmatrix und der globalen Steifigkeitsmatrix zu entwickeln. Inwieweit sich Schwierigkeiten bei der Formulierung mit unterschiedlichen Elementtypen ergeben, ist zum jetzigen Zeitpunkt noch nicht abzusehen.

Literatur

[1] Bathe, K–J.; Wilson, E. L.: Numerical Methods in Finite Element Analysis. New Jersey: Prentice Hall, 1976

[2] Bergan, P. G.: Finite Elemente in nonlinear Mechanics. Trondheim: Norwegain Institute of Technology, 1977

[3] Chung, T. J.: Finite Elemente in der Strömungsmechanik. München: Carl Hanser Verlag, 1982

[4] Heuser, H.: Gewöhnliche Differentialgleichungen. Karlsruhe: B.G. Teubner Stuttgart, 1989

[5] König, M.: Simpliziale Finite–Element–Approximation. Hannover: Institut für Bauinformatik Universität Hannover, 1996

[6] Meißner, U.; Menzel, A.: Die Methode der Finiten Elemente. Hannover: Springer–Verlag, 1989

[7] Milbradt, P.: Zur mathematischen Modellierung großräumiger Wellen– und Strömungsvorgänge. Hannover: Institut für Bauinformatik Universität Hannover, 1995

[8] Oden, J. T.; Reddy, J. N.: An Introduction to the Mathematical Theory of Finite Elements. Austin: John Wiley & Sons, 1976

[9] Rose, M.; Hüttermann, R.: Forum Bauinformatik – Junge Wissenschaftler forschen Hannover ’95 – Entwicklung einer Datenstruktur für Graphen zum Einsatz in Ingenieursanwendungen. Hannover: VDI Verlag, 1995

[10] Sellerhoff, F.: Objektorientiertes Modell für simpliziale Zerlegungen im n–dimensionalen Raum. Hannover: Institut für Bauinformatik Universität Hannover, 1995

[11] Zienkiewicz, O. C.: Methode der Finiten Elemente. München: Carl Hanser Verlag, 1984