Unterlagen zu FEM

Andreas Stahel, HTA Biel

8. März 2002

Inhaltsverzeichnis

Verstehen der **F**inite **E**lement **M**ethode und ihrer Programmierung anhand von einfachen physikalischen Problemen, basierend auf mathematischen und physikalischen Grundlagen.

1 Einführung

Ziel dieses Kurzvortrages ist es die Zuhörer auf einige mögliche Hilfsmittel für ein Einführung in die FEM hinzuweisen.

- Kursunterlagen die (auch) direkt von Gymnasiasten lesbar sind.
- Unterlagen, die für Lehrer nützlich sein könnten für die Vorbereitung von Kursen.
- Ein Programmpakete zur Lösung von einfachen FEM Problemen.

Sowohl Kursunterlagen wie Programme haben zum Ziel die Studenten mit den Grundlagen der FEM vertraut zu machen. Die mathematische Grundlage der Algorithmen ist sorgfältig erklärt und die Implementation kommentiert. Konsequenterweise sind nur sehr wenig Elementtypen implementiert. Einige technisch–naturwissenschaftliche Probleme können mittels der Programme gelöst werden.

Die Programme und Unterlagen sind nicht geeignet um grössere Ingenieur–Probleme zu lösen. Die Benutzeroberflächen (sofern vorhanden) sind nicht für DAU's ausgelegt.

Die Programmpakete bauen auf zwei Grundprogrammen auf. Hier sind mögliche Vor– und Nachteile der Grundpakete aufgeführt:

- *Mathematica*[1] : ein kommerzieles Paket für symbolische Berechnungen und Visualisierungen
  - Vorteile: sehr viele Möglichkeiten, verbreitet, platformunabhängig, gute Graphik Unterstützung, gute Platform für Testen von Algorithmen
  - Nachteile: teuer, sehr viele Möglichkeiten, steile Lernkurve, langsam (für FEM)
  - FEM : langsam (< 1000 Gleichungen gut lösbar, grössere Probleme strapazieren Resourcen und Geduld), gute Graphik
- *Octave*[2] : ein Matrix orientiertes Paket, mit grossen Ähnlichkeiten zu *MATLAB*
  - Vorteile: **kostenlos**, leicht erlernbar, platformunabhängig, stabil, erweiterbar, `C++` Code kann leicht eingebunden werden (Geschwindigkeit)
  - Nachteile: keine spektakuläre Benutzeroberfläche, keine symbolischen Rechnungen, Graphiken etwas mühsam zu erstellen (Gnuplot)
  - FEM : schnell (einige tausend Gleichungen sind kein Problem), gut erweiterbar, Graphikmöglichkeiten sind nicht spektakulär
- Gittererzeugung : für die FEM müssen oft Gebiete in Elemente (z.B. Dreiecke) zerlegt werden. Diese Problem wird mit Hilfe von zwei Programmen gelöst: `EasyMesh` und **triangle**, siehe [1], [2]. Da die WWW–Seite von `EasyMesh` manchmal nicht erreichbar ist, kann die Information auch auf [3] gefunden werden.

Der Autor bevorzugt die *Octave* Umgebung, die Auswahl sollte aber Vorkenntnisse von Lehrer/Schüler in Betracht ziehen.

2 Unterlagen für Kurse und Vorbereitung

2.1 Ebene Stabsysteme

In [3] wird das Problem von ebenen Zug–Druck–Stäben mit Hilfe vom FEM diskutiert. Eine typische Situation ist in Abbildung 1 gezeigt. Aufgrund der notwendigen Vorkenntnisse in Physik und Mathematik könnten diese Unterlagen für Gymnasien auch geeignet sein.

Diese Notizen werden an der HTA Biel regelmässig eingesetzt um die Studenten mit den Grundlagen von FEM vertraut zu machen. Der Aufbau des ersten Abschnittes ist

- Konstruktion der Elementsteifigkeitsmatrix
- Konstruktion der Gesamtsteifigkeitsmatrix
- Berücksichtigen der Zwangsbedingungen
- Lösen und Interpretation der Resultate
- Codierung in *Octave*

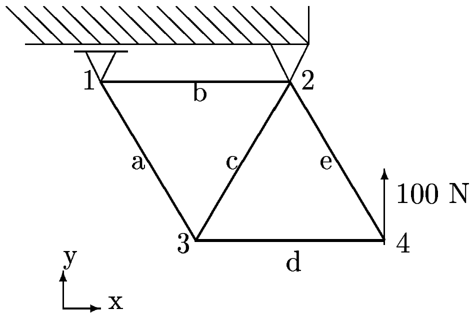

Abbildung 1: Struktur aus Zug– und Druckstäben

In weiteren Abschnitten werden die folgenden Aspekte präsentiert:

- Elementsteifigkeitsmatrix für einen Biegestab
- Lösen von 'grossen' Gleichungssystemen, Verfahren von Cholesky, Bandstruktur der Matrix, Numerierung der Knoten.

2.2 Calculus of Variations and Finite Elements

Diese englischen Kursunterlagen [3] stellen erheblich höhere Anforderungen an die mathematische und physikalische Vorbildung. In der gegebenen Form können sie als Informationsquelle für Dozenten hilfreich sein. Einige Abschnitte sind als Dokumentation zu den unten aufgeführten Programmen brachbar. Hier ist eine Kurzbeschreibung des Inhalts.

- einführende Beispiele (eventuell geeignet für Gymnasien)
  - Ebenes System von Zug–Druck–Stäben, gelöst mittels Energieminimierung
  - Strecken eines Stabes
- Variationsrechnung, vom Extremalproblem zur Differentialgleichung
- Finite Element Probleme in einer Variablen
  - Elemente erster Ordnung
  - Gauss Integration, Elemente zweiter Ordnung
  - Dokumentierte Codierung in *Mathematica*
  - von Eigenwerten zu Eigenschwingungen
  - Approximationsfehler, Konvergenz
- Finite Element Probleme in zwei Variablen
  - Dreieckselemente mit linearen Basisfunktionen
  - Aufbau des Gleichungssystems
  - Codierung in *Mathematica*
- Elastizität
- *MATLAB*–PDE Toolbox
- Matrix Berechnungen, Theorie und Code, Leistungsdaten von CPU's und Compilern.

[1] `http://www.wri.com`
[2] `http://www.octave.org`

3 *Mathematica* 1D

3.1 Beschreibung

Mit einem *Mathematica*–Paket [3] können Randwert–Probleme in einer Variablen gelöst werden, d.h. für gegebene Funktionen $a(x)$, $b(x)$ und $f(x)$ ist die Differentialgleichung

$$\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\, u(x)\right) - b(x)\, u(x) = f(x) \quad \text{für} \quad x_l < x < x_r$$

aufzulösen nach der unbekannten Funktion $u(x)$. An den beiden Randpunkten $x = x_l$ und $x = x_r$ kann je der Wert der Funktion oder die Steigung gegeben werden. Der Code basiert auf Elementen zweiter Ordnung und Gauss Integration, eine detailierte Beschreibung finden Sie in [4] .

3.2 Streckung eines Stabes

Als Beispiel sei hier die Streckung eines Stabes mit variablem Querschnitt angegeben, siehe Abbildung 2.

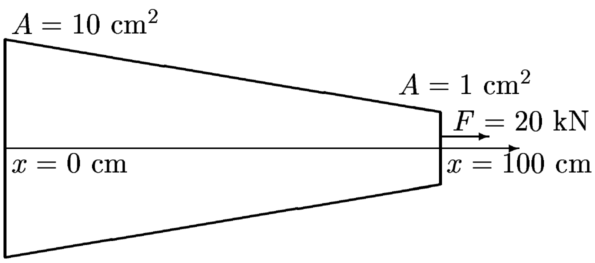

Abbildung 2: Stab mit variablem Querschnitt

Mathematica

```
<<BVP.m;
<<Interpol2.m;
Clear[A,b,f,x]
A = Function[x,-(10-0.09*x)*3*10^6];
b = Function[x,0];
f = Function[x,0];
n = 5;
x = Table[s,{s,0,100,100/n}];
data = BVP[A,b,f,x,{"D","N"},{0,2*10^4}];

uexact[x_] = -Integrate[2*10^4/A[s], {s, 0, x}];
Plot[{uexact[xt], Interpol2[xt, data]}, {xt, 0, 100}, PlotRange -> All,
AxesLabel -> {"x", "Displacement"},PlotStyle -> {Dashing[{0.03, 0.02}], Dashing[{}]}];

Duexact[xt_] = D[uexact[xt], xt];
Plot[{Duexact[xt], DInterpol2[xt, data]}, {xt, 0, 100}, PlotRange -> {0,0.007},
AxesLabel -> {"x", "Strain"},PlotStyle -> {Dashing[{0.03, 0.02}], Dashing[{}]}];
```

Als Resultat erhält man die Verschiebung und Spannung als Funktion des Ortes x in Abbildung 3.

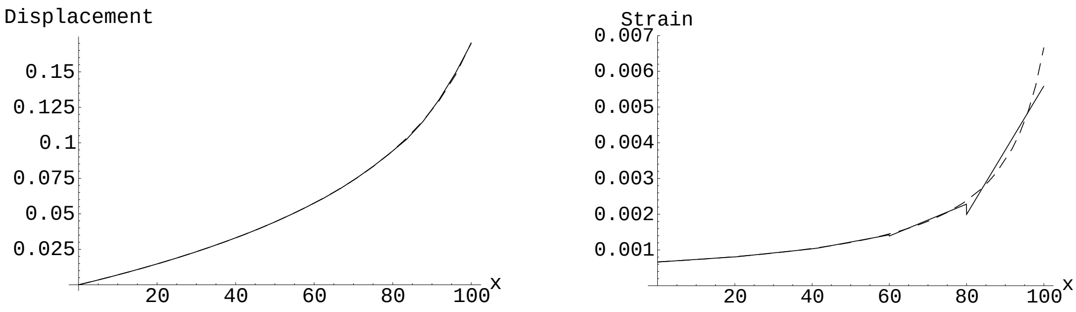

Abbildung 3: Verschiebung und Spannung mit 5 Elementen der Ordnung 2

4 *Mathematica* 2D

4.1 Beschreibung

Mit diesem *Mathematica*–Paket [3] können Randwert–Probleme in einer Variablen gelöst werden, d.h. für gegebene Funktionen a, b, f, g_1 und g_2 ist die partielle Differentialgleichung

$$\begin{aligned} \nabla\cdot(a\,\nabla u) - b\,u &= f && \text{für} \quad (x,y)\in\Omega \\ u &= g_1 && \text{für} \quad (x,y)\in\Gamma_1 \\ a\,\tfrac{\partial\,u}{\partial n} u &= g_2 && \text{für} \quad (x,y)\in\Gamma_2 \end{aligned}$$

aufzulösen nach der unbekannten Funktion $u(x)$. Der Code basiert auf Dreieck–Elementen erster Ordnung, wie beschrieben in [4] .

4.2 Beispiele

4.2.1 Eine einfache Gleichung auf einem Rechteck

Hier ist eine auf einem Rechteck definierte Gleichung.

$$\begin{aligned} u_{xx}+u_{yy} &= -1 && \text{für} \quad (x,y)\in(0,5)\times(0,4) \\ u(x,y) &= \tfrac{y}{10} && \text{für} \quad x\in(0,5) \text{ und } y\in\{0,4\} \\ \tfrac{\partial\,u}{\partial n}u(x,y) &= -1 && \text{für} \quad x\in\{0,5\} \text{ und } y\in(0,4) \end{aligned}$$

Mathematica

```
<<BVP2.m
{nodes,elements,segments}=ReadMesh["test1"];

a = Function[{x, y}, 1];
b = Function[{x, y}, 0];
f = Function[{x, y}, -1];
gDirichlet = Function[{x, y}, y/10];
gNeumann = Function[{x, y}, -1];

points = FEMSolve[a, b, f, gDirichlet, gNeumann, nodes, elements, segments];

Show[FEMSurface[points, elements],
     Axes -> True, AxesLabel -> {"x", "y", "u"},
     AspectRatio -> 1, ViewPoint -> {4, -4, 2}];
```

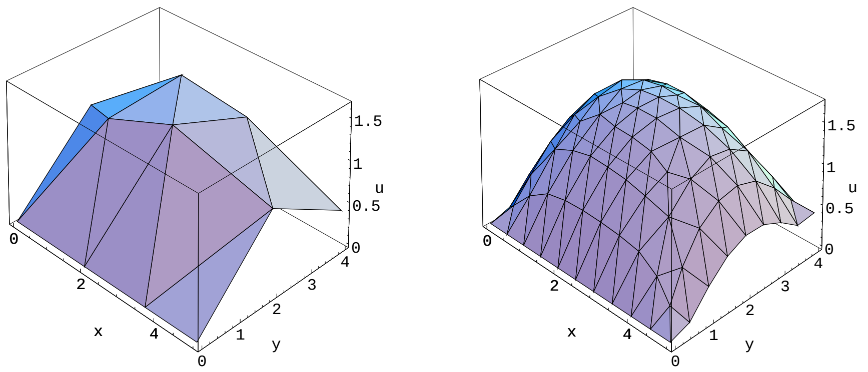

Abbildung 4: Lösung des Test–Problems mit wenigen und vielen Elementen

4.2.2 Laminare Strömung einer idealen Flüssigkeit

Das Notebook `Fluss.nb` enthält Code um einen laminaren Fluss durch einen Kanal mit Verengung zu berechnen. Das Geschwindigkeitsfeld ist in Abbildung 6 gezeigt. Es kann ein Strömungspotential berechnet werden, mit Hilfe der FEM auf einem von mit `EasyMesh` erzeugten Gitter. Gitter und Potential sind in Abbildung 5 gezeigt.

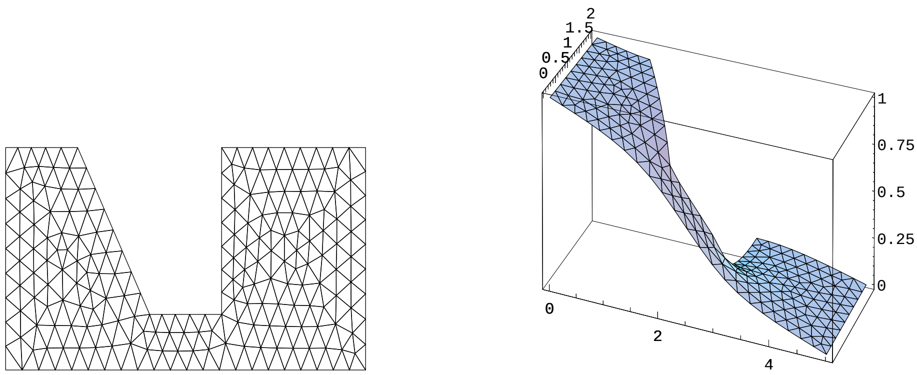

Abbildung 5: Gitter und Potential für eine laminare Strömung

5 FEMoctave

Das Paket basiert auf dem kostenlos erhältlichen Programm *Octave* . Die FEM Ergänzungen sind ebenfalls frei verwendbar. Der Autor hat nur auf Unix/Linux Platformen gearbeite. *Octave* kann auch (mit Mehraufwand) auf Win??–Systemen installiert werden. Tips sind zu finden ...

5.1 Beschreibung

Mit diesen Paket können elliptische Differentialgleichungen zweiter Ordnung auf Gebieten $\Omega \subset \mathbb{R}^2$ mit Rand $\Gamma = \Gamma_1 \dot\cup \Gamma_2$ untersucht werden. Für gegebene Funktionen a, b, f, gD und gN wird eine Approximation der Lösung u der partiellen Differentialgleichung

$$\begin{aligned} \operatorname{div}(a\,\operatorname{grad} u) - b\,u &= f && \text{in} \quad \Omega \\ u &= gD && \text{on} \quad \Gamma_1 \\ a\,\frac{\partial\, u}{\partial n} &= gN && \text{on} \quad \Gamma_2 \end{aligned}$$

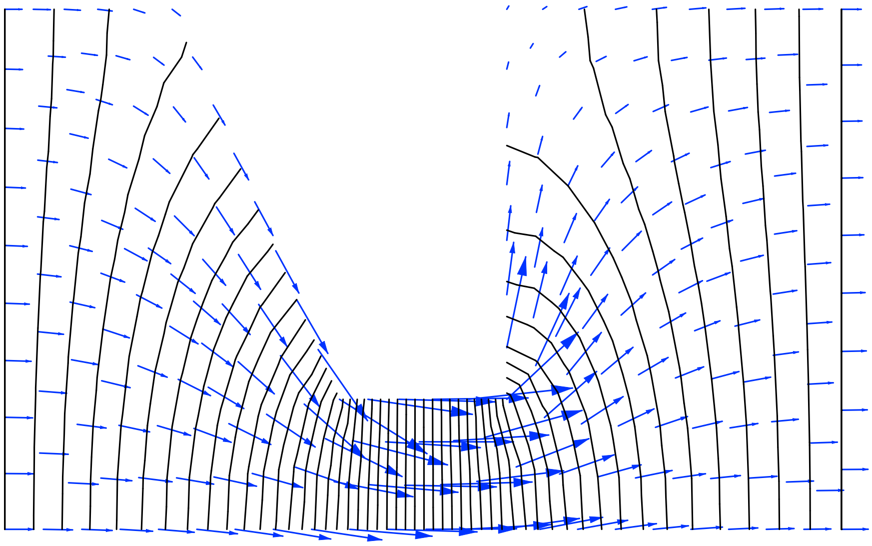

Abbildung 6: Geschwindigkeitsfeld der laminaren Strömung

berechnet. Das Gebiet Ω wird in Dreiecke zerlegt und auf jedem Dreieck wird eine lineare Funktion verwendet. In Tabelle 1 finden Sie einige der Anwendungen, die auf Gleichungen vom obigen Typ führen.

Dokumentation zum Programmpaket und etliche Beispiele finden Sie in [3]. Zur Erzeugung der Triangularisierung kann entweder `EasyMesh` [1] oder `triangle` [2] verwendet werden. Da die Web–Seite von `EasyMesh` oft nicht erreichbar ist habe ich die Files auch auf meiner Seite [3] zur Verfügung gestellt. Als Ergänzung zu `triangle` kann der Algorithmus von `Cuthill-McKee` verwendet werden um die Knotennummerierung für eine Bandstruktur der Matrix zu optimieren.

5.2 Beispiele

Beschreibung und Beispiele

5.2.1 Eine einfache Gleichung auf einem Rechteck, Steifigkeitsmatrizen

Nur mit Hilfe von Script Files kann die folgende Gleichung auf dem rechteckigen Bereich $\Omega = [0,1] \times [0,2]$ gelöst werden.

$$\begin{aligned} \Delta u &= -1 \quad \text{in} \quad \Omega \\ u &= 0 \quad \text{on} \quad \partial\Omega \end{aligned}$$

Octave

```
function res = aF(xy)   [n,m]=size(xy); res=ones(n,1);    endfunction
function res = zero(xy) [n,m]=size(xy); res=zeros(n,1);   endfunction
function res = fF(xy)   [n,m]=size(xy); res=-1*ones(n,1); endfunction

x=linspace(0,1,5); y=linspace(0,2,5);
[nodes,elem,edges]=CreateRectMesh(x,y,1,1,1,1);
[A,b,n2d]=FEMEquationM(nodes,elem,edges,'aF','zero','fF','zero','zero');
u=FEMSolve(nodes,A,b,n2d,'zero');
A
b'
```

Field of application	Primary variable	Material constant	Source variable	Secondary variables
General situation	u	a	f	q, $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial y}$
Heat transfer	Temperature T	Conductivity k	Heat source Q	Heat flow density $\vec{q}$ $\vec{q} = -k \nabla T$
Electrostatics	Scalar potential Φ	Dielectric constant ε	Charge density ρ	Electric flux density D
Magnetostatics	Magnetic potential Φ	Permeability ν	Charge density ρ	Magnetic flux density B
Transverse deflection of elastic membrane	Transverse deflection u	Tension of membrane T	Transversely distributed load	Normal force q
Torsion of a bar	Warping function ϕ	Constant 1	Constant 0	Stress τ $\tau_{xz} = \frac{E\,\alpha}{2\,(1+\nu)}\,(-y + \frac{\partial\,\phi}{\partial x})$ $\tau_{yz} = \frac{E\,\alpha}{2\,(1+\nu)}\,(x + \frac{\partial\,\phi}{\partial y})$
Irrotational flow of an ideal fluid	Stream function Ψ Velocity potential Φ	Density ρ	Mass production σ (usually zero)	Velocity $(u,v)^T$ $\frac{d\Psi}{\partial x} = -u$ $\frac{d\Psi}{\partial y} = v$ $\frac{d\Phi}{\partial x} = u$ $\frac{d\Phi}{\partial y} = v$
Ground-water flow	Piezometric head Φ	Permeability K	Recharge Q (or pumping $-Q$)	seepage $q = K\,\frac{\partial\Phi}{\partial n}$ velocities $u = -K\,\frac{d\Phi}{\partial x}$ $v = -K\,\frac{d\Psi}{\partial y}$

Tabelle 1: Einige Beispiele der Poisson Gleichung $-\nabla\,(a\,\nabla u) = f$

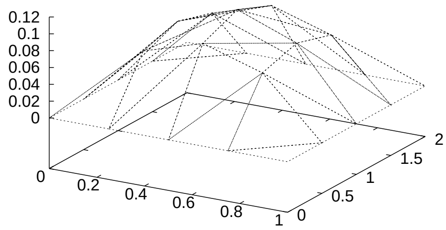

Abbildung 7: Lösung eines einfachen Problems

Der Code berechnet die globale Steifigkeitsmatrix `A` und den Vektor `b`

```
A =
   5.000  -2.000   0.000  -0.500   0.000   0.000   0.000   0.000   0.000
  -2.000   5.000  -2.000   0.000  -0.500   0.000   0.000   0.000   0.000
   0.000  -2.000   5.000   0.000   0.000  -0.500   0.000   0.000   0.000
  -0.500   0.000   0.000   5.000  -2.000   0.000  -0.500   0.000   0.000
   0.000  -0.500   0.000  -2.000   5.000  -2.000   0.000  -0.500   0.000
   0.000   0.000  -0.500   0.000  -2.000   5.000   0.000   0.000  -0.500
   0.000   0.000   0.000  -0.500   0.000   0.000   5.000  -2.000   0.000
   0.000   0.000   0.000   0.000  -0.500   0.000  -2.000   5.000  -2.000
   0.000   0.000   0.000   0.000   0.000  -0.500   0.000  -2.000   5.000
b'=
  -0.125  -0.125  -0.125  -0.125  -0.125  -0.125  -0.125  -0.125  -0.125
```

Man kann auch Elementsteifigkeitsmatrizen und entsprechende Vektoren bestimmen. Für das Dreieck mit den Ecken $(0,0,$ $(1,0)$ und $(0,1)$ führt der Code

Octave

```
[mat,vec]=ElementContribution([0,0;1,0;0,1],'aF','zero','fF')
```

zum Resultat

```
mat =   1.00000  -0.50000  -0.50000
       -0.50000   0.50000   0.00000
       -0.50000   0.00000   0.50000

vec =  -0.16667
       -0.16667
       -0.16667
```

Beispiele dieser Art können nützlich sein um die Grundlagen der Finite Element Methode zu verstehen.

5.2.2 L–förmiges Gebiet, Gradientenfeld, Eigenfunktion

Auf einem L–förmigen Gebiet soll die Gleichung

$$\begin{aligned} \Delta u &= 10\,(x-y) && \text{in} \quad \Omega \\ u &= 0 && \text{on} \quad \partial\Omega \end{aligned}$$

untersucht werden. Mit Hilfe von `EasyMesh` wird ein Gitter mit ca. 2500 Knoten erzeugt. Dann kann die Lösung der Gleichung, das durch den Gradienten erzeugte Vektorfeld, ein Schnitt entlang der Diagonalen und die dritte Eigenfunktion bestimmt werden. Das Resultat finden Sie in Abbildungen 8 und 9. Da kompilierter Code und die Bandstruktur der Matrix verwendet wird kann das gesamte Problem in wenigen Sekunden gelöst werden.

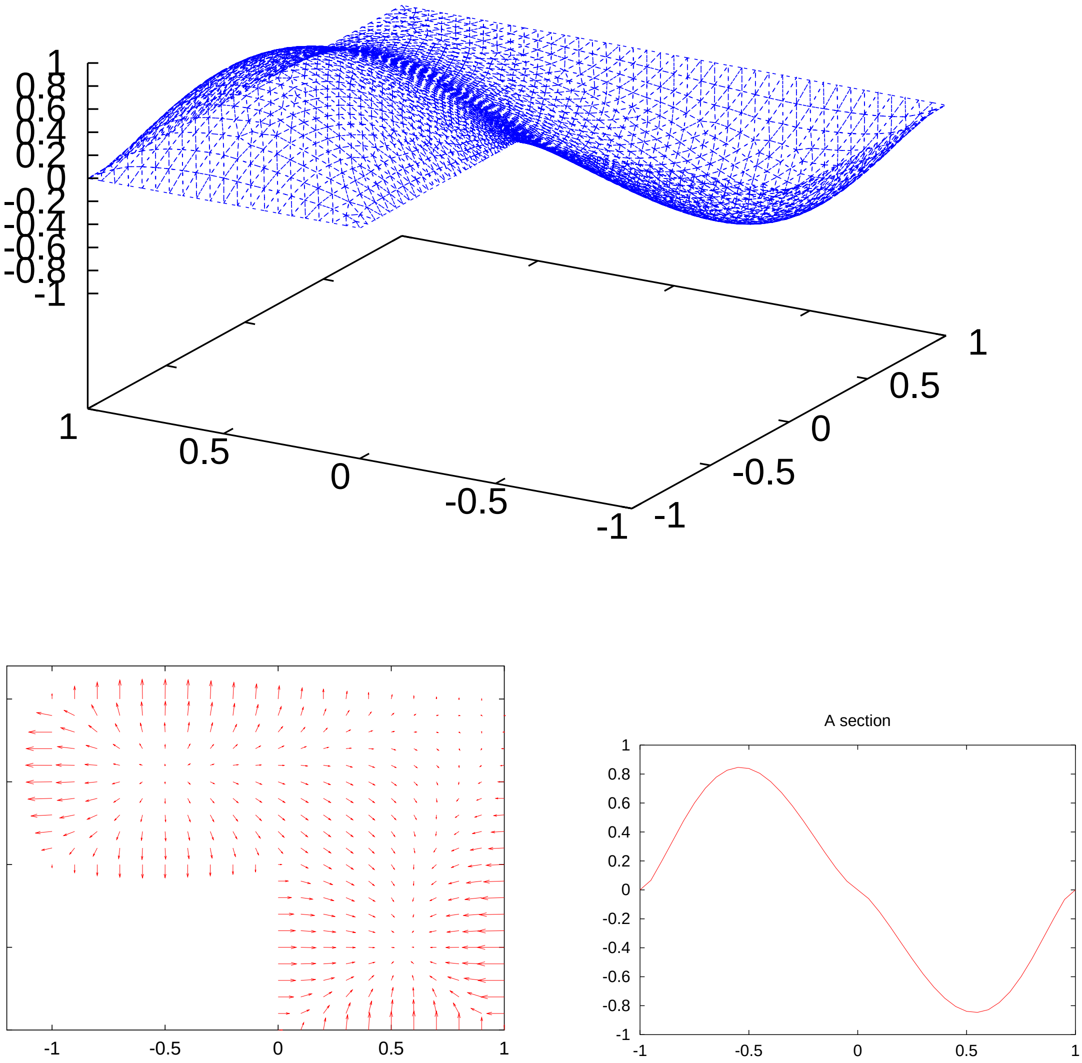

Abbildung 8: Lösung der PDE, das Gradienten–Feld und ein Schnitt durch die Lösungsfläche

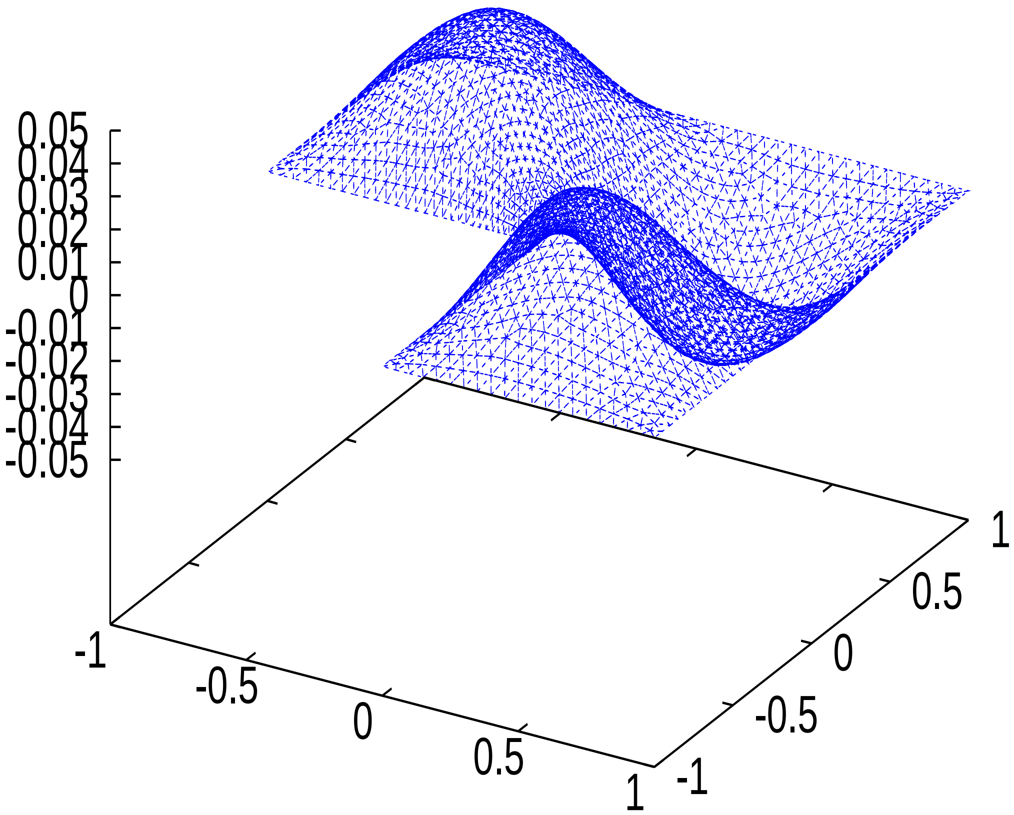

Abbildung 9: Graph der dritten Eigenfunktion

5.2.3 Ein Strömungsproblem

Das in Abschnitt 4.2.2 mit *Mathematica* gelöste Problem kann auch mit `FEMoctave` untersucht werden. Mittels `triangle` und `Cuthill-McKee` kann ein Gitter mit ca. 3100 Knoten und 6000 Elemente erzeugt werden. Die entstehende 3073×3073-Matrix hat eine halbe Bandbreite von nur 63 und die Gleichung kann in wenigen Sekunden gelöst werden. Ein Resultat ist in Abbildung 10 gezeigt.

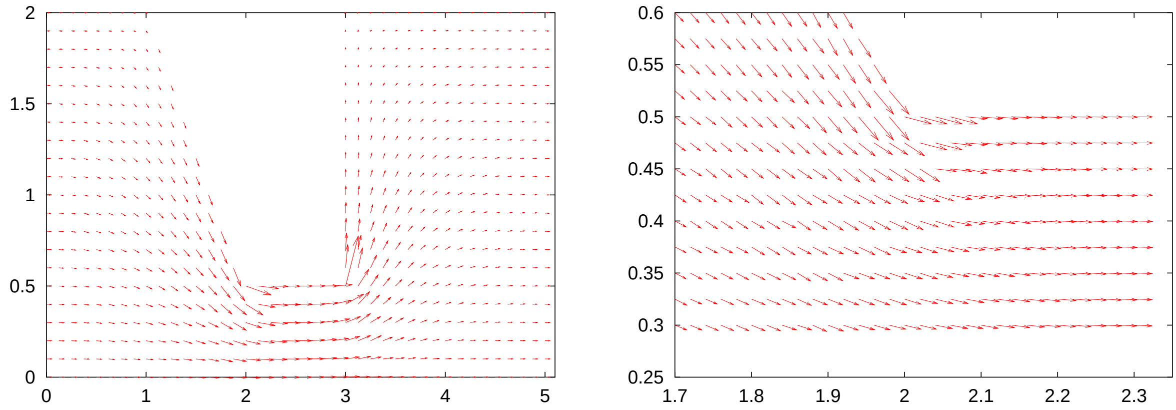

Abbildung 10: Das gesamte Geschwindigkeitsfeld und eine Detailansicht

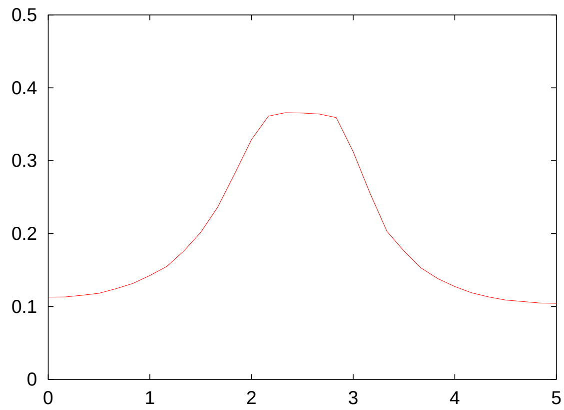

Abbildung 11: Die horizontale Geschwindigkeit entlang einer horizontalen Geraden

Literatur

[1] Bojan Niceno. http://www-dinma.univ.trieste.it/~nirftc/research/easymesh/.

[2] Jonathan Richard Shewchuk. http://www-2.cs.cmu.edu/~quake/triangle.html.

[3] Andreas Stahel. Web page. www.hta-bi.bfh.ch/~sha.

[4] Andreas Stahel. Calculus of Variations and Finite Elements. Lecture Notes used at HTA Biel, 2000.