

# Simulation einer magnetischen Rastertunnelmikroskop - Spitze

Emanuel Gull  
Betreuer: Kersten Schmidt

8. Juni 2004

## Zusammenfassung

Dieser kurze Bericht beschreibt die Semesterarbeit, die ich im März 2004 am Seminar für Angewandte Mathematik geschrieben habe. Die Simulation beschreibt ein neues Gerät, das zur Zeit in der Oberflächenphysik entwickelt wird. Das magnetische Feld einer Spule wird durch eine ferromagnetische Spitze gebündelt und auf eine mikroskopische Probe fokussiert. Wieviel des magnetischen Feldflusses bricht aus der Spitze aus, und wieviel wird gebündelt und erreicht die Probe? Die numerischen Resultate können eventuell helfen, die Physik von dünnen ferromagnetischen Inseln besser zu verstehen.

Betreut wurde ich in dieser Zeit von Kersten Schmidt. Vielen Dank für die Hilfe mit Concepts, die anregenden Diskussionen und die zahlreichen Design-Vorschläge!

## 1 Die Physikalischen Aspekte der Simulation

### 1.1 STM – Scanning Tunneling Microscope oder Rastertunnelmikroskop

Das Scanning Tunneling Microscope (auf Deutsch: Rastertunnelmikroskop) wurde von Gerd Binnig und Heinrich Rohrer Ende der Siebziger Jahre am IBM-Labor in Rüschlikon entwickelt. 1986 erhielten beide je einen Viertel Nobelpreis für ihre Arbeit – die andere Hälfte ging an Ernst Ruska für die Entwicklung des Elektronenmikroskops in den Jahren 1931 - 1939. Das Scanning Tunneling Microscope ist in der Lage, die Elektronenstruktur und Topologie von Oberflächen aufzulösen, und kann im günstigsten Fall sogar einzelne Atome abbilden. Eine Metallspitze (normalerweise Wolfram oder Platin) wird dazu auf ein Potential gebracht, das leicht verschieden von demjenigen der Oberfläche ist. Darauf wird die Spitze mit Hilfe von Piezokristallen zur Probe geführt, bis ein Tunnelstrom einsetzt. Wiederum werden piezoelektrische Effekte benutzt, um die Spitze über die Oberfläche schwingen zu lassen. Je nach Abstand zur Oberfläche und Elektronendichte fließt dabei ein unterschiedlicher Tunnelstrom. Da dieser exponentiell von der Distanz abhängig ist, lassen sich “Höhenunterschiede” auf der Oberfläche sehr genau messen. Aus dem millionenfach verstärkten Tunnelstrom können schliesslich physikalisch relevante Informationen gewonnen und Bilder (s. Abbildung 1) generiert werden.

### 1.2 STM mit magnetischer Spitze

Die Idee des magnetischen STMs ist etwas verschieden von derjenigen des konventionellen Rastertunnelmikroskops. Hier soll nicht atomare Auflösung erreicht werden,

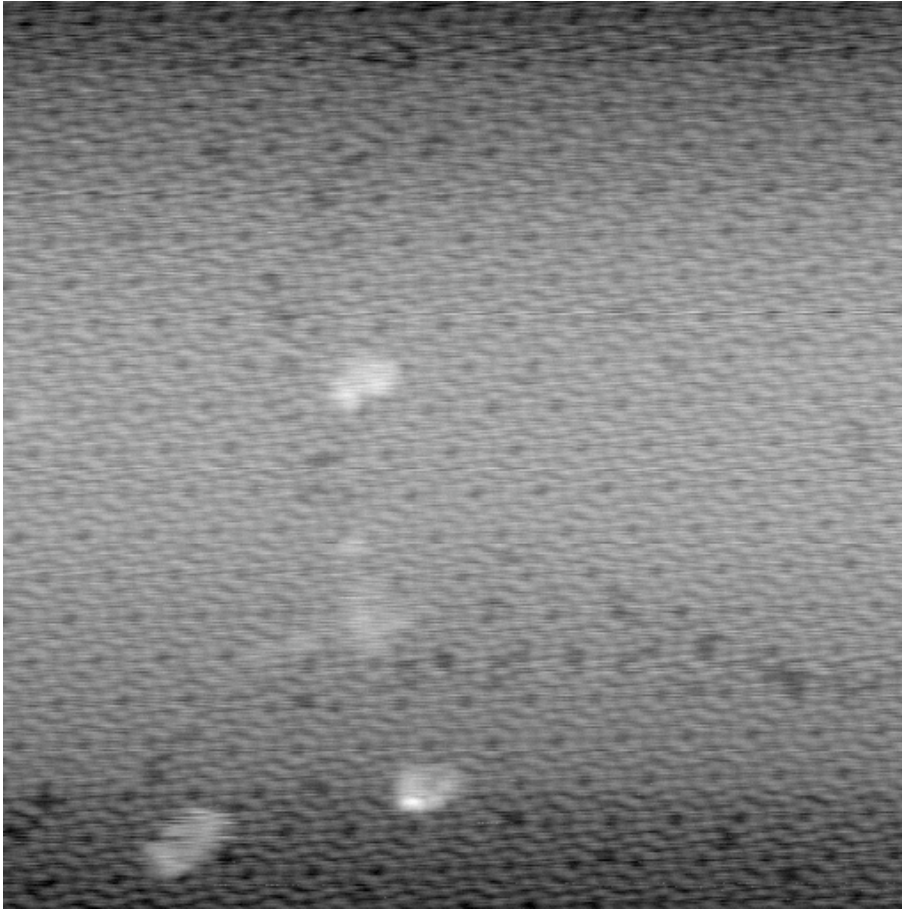


Abbildung 1: Atomare Auflösung: Diese STM-Bild zeigt eine  $7 \times 7$  - Oberflächenrekonstruktion eines Si-111-Wafers. Die weissen Punkte sind einzelne Si-Atome. Die schwarzen Punkte sind Löcher in der Oberflächenstruktur (die Atome an diesen Plätzen befinden sich in tieferen Gitterlagen). Ebenfalls sichtbar sind Verunreinigungen auf der Oberfläche (vermutlich Sauerstoff) sowie mehrere Domänengrenzen und Fehlstellen. Aufgenommen von Thomas Michlmayr und mir im Sommer 2003.

sondern es soll eine Möglichkeit geben, die Magnetisierung von dünnen ferromagnetischen Inseln auf einer Probe mit Stromstößen durch eine Spule gezielt umzupolen. Deshalb besteht die Spitze des STMs nicht aus Wolfram oder Platin, sondern aus dem ferromagnetischen Material Cobalt.

Das Instrument kann selbstverständlich ebenfalls wie ein konventionelles STM benutzt werden. Dazu wird das Magnetfeld abgeschaltet. Darauf bringt man wiederum die Spitze auf ein leicht höheres Potential als die Probe und “rastert” über die Oberfläche. Aufgrund der Beschaffenheit der Cobaltdrähte, aus denen die Spitze hergestellt wird, ist es jedoch kaum möglich, atomare Auflösung zu erreichen. Dazu müsste das vorderste Ende der Spitze aus einem einzigen Atom bestehen, die Spitze also “monoatomar” scharf sein. Im Fall von Wolfram wird dies durch einen speziellen Ätzprozess in einer Lauge erreicht. Dieses Verfahren ist jedoch für Cobalt nicht anwendbar, und die Spitze deshalb von weit geringerer Qualität. Trotzdem sollte die Auflösung gut genug sein, um die auf der Oberfläche aufgedampften ferromagnetischen Inseln zu sehen.

Zusätzlich zum “Scannen” kann das Gerät jedoch auch die Magnetisierung der ferromagnetischen Inseln umkehren. Um die Cobalt-Spitze ist eine kleine Spule gewickelt, durch die ein starker Strom fließt. Dieser generiert ein Magnetfeld entlang der Spitze, das das ferromagnetische Material magnetisiert. Der grösste Teil des magnetischen Flusses bleibt dabei innerhalb des Gebietes mit hoher Permeabilität (d.h. innerhalb der Spitze) und wird auf die Probe gelenkt. Es ist jedoch nicht a priori klar, wieviel Fluss auf der Probe ankommt, und wieviel aus der Spitze “ausbricht” und damit für das Experiment verloren geht.

### 1.3 SEMPA

Zusätzlich zum STM befindet sich ein zweites Instrument in der Vakuumkammer: ein SEMPA (Scanning Electron Microscope with Polarization Analysis). Dieses Instrument analysiert die Polarisation der Sekundärelektronen, die ein Primärelektronenstrahl aus der Probenoberfläche schlägt. Damit kann es sowohl topologische Information (analog einem Elektronenmikroskop), als auch magnetische Information, die aus der Polarisation der Sekundärelektronen extrahiert wird, messen. Der Setup mit diesen zwei Instrumenten erlaubt es, die Magnetisierungsänderung der Inseln “live” zu beobachten.

Die Abbildungen 2 und 3 zeigen den Aufbau und den ersten Entwurf des Apparats. Leider muss das Gerät zur Zeit wegen mechanischen Problemen umgebaut werden. Abbildung 4 zeigt, wie die Spitze vorne in Domänen zerfällt.

### 1.4 Ziel der Simulation

Das Ziel der Simulation ist es, die genaue Feldstärke des Magnetfeldes an der Spitze des STMs zu bestimmen. Da einerseits der Fluss bevorzugt innerhalb der Region mit hoher Permeabilität bleiben sollte, andererseits aber dieses Gebiet mikroskopisch klein wird, ist nicht klar, wie stark das Feld auf der Probe ist. Es ist jedoch wichtig, dies zu wissen, da damit magnetische Eigenschaften der zu untersuchenden Inseln genau bestimmt werden können.

## 2 Modellierung des STMs– Spitze und Spule

Ein STM ist ein sehr empfindliches und kompliziertes Gerät mit unzähligen Teilen. Im Fall dieser magnetostatischen Simulation müssen jedoch nur diejenigen Gebiete berücksichtigt werden, die entweder Strom leiten (d.h. hier die Spule) oder eine Permeabilität verschieden von derjenigen des Vakuums haben (d.h. die Spitze). An

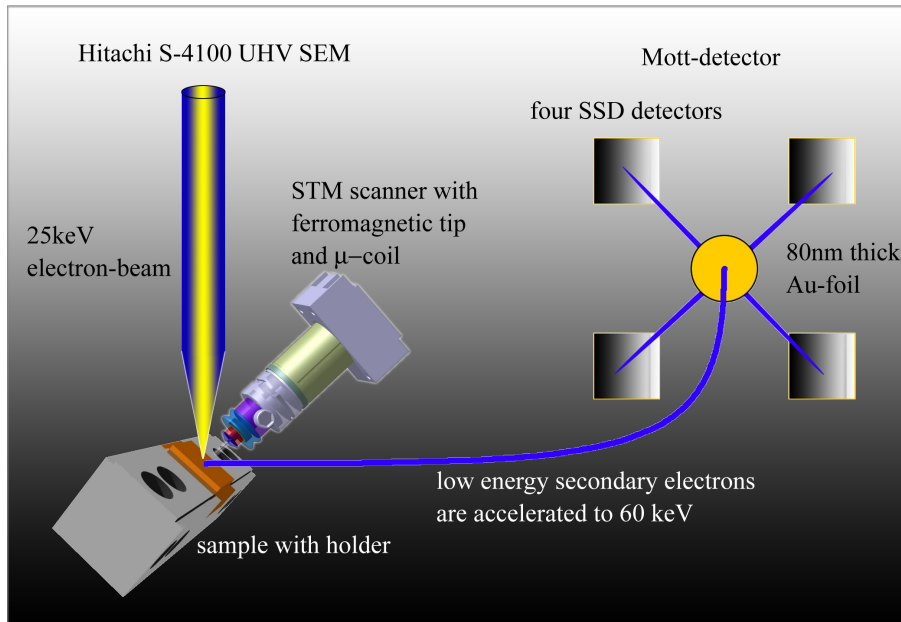


Abbildung 2: Skizze Versuchsaufbau

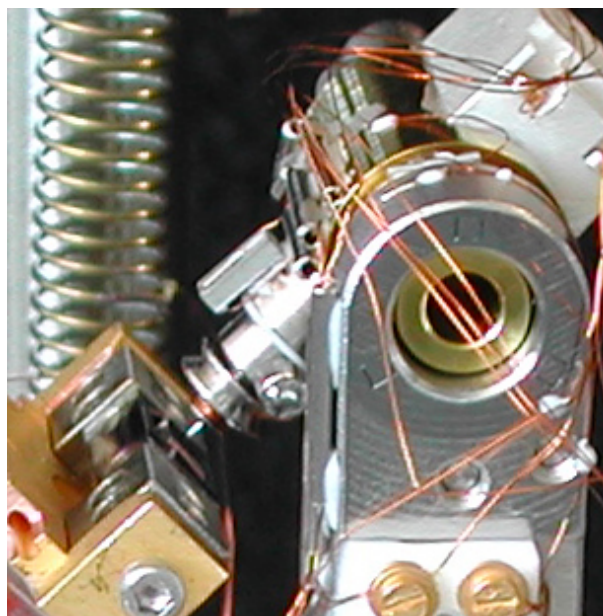


Abbildung 3: Photographie des STMs. Gut sichtbar sind der schiefe Probenhalter und die Spitze links unten im Bild.

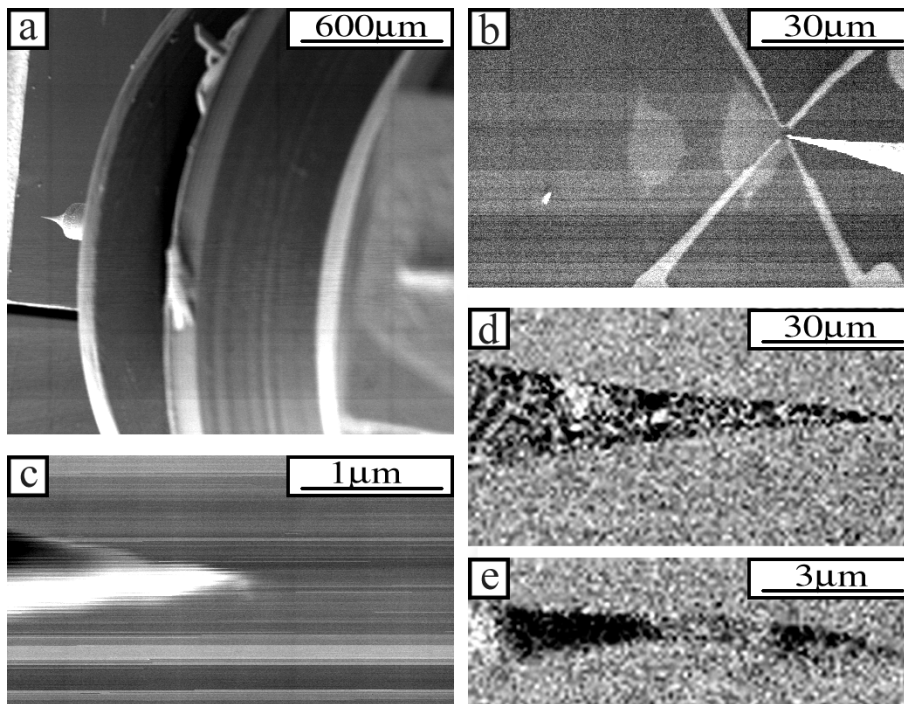


Abbildung 4: Elektronenmikroskop / SEMPA - Aufnahme der Spitze. Man sieht, wie sie in verschiedene Domänen zerfällt, also ferromagnetisch ist (Ausschnitt “3µm” unten rechts).

die anderen Teile - insbesondere Probe, Probenhalter und Spitzenhalter – muss das Gitter nicht extra angepasst werden, denn sie sind magnetisch unsichtbar. Zudem hat das System eine sehr einfache Symmetrie: es ist rotationsinvariant um die zentrale Achse. Deshalb kann es in einer zweidimensionalen Magnetostatik - FEM - Simulation behandelt werden. Jedoch gibt es eine zusätzliche Schwierigkeit: auf der einen Seite befindet sich eine makroskopische Spule mit einem Durchmesser von etwa einem Quadratmillimeter; auf der anderen Seite möchte man gerne Größen im mikroskopischen Bereich berechnen (Feld an der Spitze). Deshalb muss sich das Gitter über etwa sieben Größenordnungen erstrecken ( $10^{-3}m - 10^{-10}m$ ). Daran scheitern die kommerziellen Gittergeneratoren zur Zeit noch.

Die ungefähre Geometrie der Spitze kann zum Beispiel mit einem Elektronenmikroskop abgebildet werden. Diese Aufnahmen zeigen eine komplett radialsymmetrische Spitze mit zwei kreisförmig gebogenen Abschnitten – einem nach innen gebogenen “Schaft” und einem runden, unteren Teil (vgl. Abbildungen später, z.B. 11 und 14). Die Abbildung 5 zeigt eine Skizze der in Concepts modellierten Geometrie.

### 3 Die Maxwellgleichungen und Magnetostatik – das magnetische Skalarpotential

Die Maxwellgleichungen und die konstituierenden Gleichungen im ferromagnetischen Medium für statische Phänomene lauten folgendermassen:

$$\vec{B} = \mu_0(\vec{H} + \vec{M}) \quad (1)$$

$$\vec{\nabla} \times \vec{H} = \vec{j} \quad (2)$$

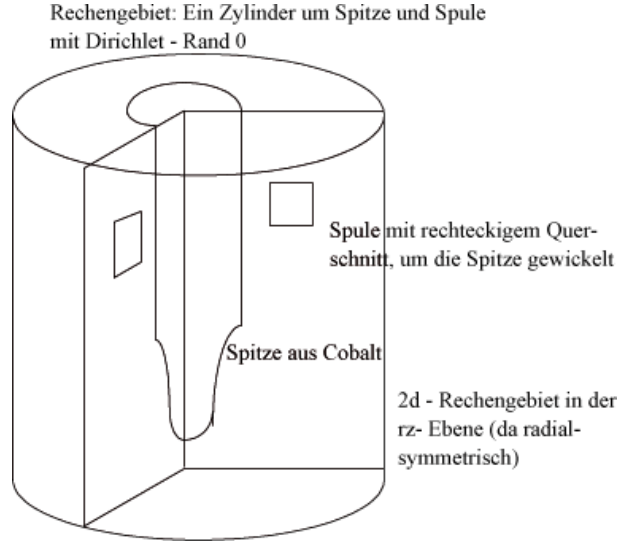


Abbildung 5: Skizze der Geometrie des Problems

$$\vec{\nabla} \cdot \vec{B} = 0, \quad (3)$$

wobei  $\vec{j}$  die (externe) Stromdichte bezeichnet.

Teilen wir das Magnetfeld auf in einen Anteil  $\vec{H}^*$ , der vom Strom der Spule herrührt, und einen Anteil, der durch das magnetische Medium hervorgerufen wird, so können wir schreiben:

$$\vec{\nabla} \times \vec{H}^* = \vec{j} \quad (4)$$

$$\vec{\nabla} \times (\vec{H} - \vec{H}^*) = 0 \rightarrow \vec{H} - \vec{H}^* = -\vec{\nabla}\Psi \quad (5)$$

Die letzte Gleichung definiert das magnetische Skalarpotential  $\Psi$ . Einsetzen von (1) und (5) in (3) führt auf

$$0 = \vec{\nabla} \cdot (\mu_0(\vec{H} + \vec{M})) = \vec{\nabla} \cdot (\mu_0(-\vec{\nabla}\Psi + \vec{H}^* + \vec{M})). \quad (6)$$

Die freien Randbedingungen (Spitze und Spule sind sehr viel kleiner als die Vakuumkammer, die restlichen Leiter werden vernachlässigt) werden approximiert durch Dirichlet - Randbedingungen  $\Psi = 0$  auf  $\Gamma = \partial\Omega$  einer Box  $\Omega$ , die genügend gross gewählt werden muss. Deshalb lautet das Problem:

$$\vec{\nabla}(\vec{\nabla}\Psi) = \vec{\nabla}(\vec{H}^* + \vec{M}) \text{ in } \Omega \quad (7)$$

$$\Psi = 0 \text{ auf } \Gamma = \partial\Omega. \quad (8)$$

Es ergibt sich das zu lösende Problem durch multiplizieren mit Testfunktion und partielle Integration:

$$\int_{\Omega} (\vec{\nabla}\Psi)(\vec{\nabla}\Phi) dx = - \int_{\Omega} (\vec{H}^* + \vec{M}) \cdot (\vec{\nabla}\Phi) dx. \quad (9)$$

Der Term über die Ränder verschwindet aufgrund der Dirichlet - Randbedingungen. Unter Ausnützung der vollständigen Rotationssymmetrie können wir auf Zylinderkoordinaten transformieren und über die Winkelkoordinate  $\phi$  integrieren. Die Integration um die Achse ergibt einen Faktor von  $2\pi$  auf beiden Seiten und zusätzliche Neumannrandbedingungen im Zentrum. Diese kann man dadurch motivieren, dass

für  $\nabla\Psi \cdot \vec{n}$  bei  $r = 0$  das B-Feld im Zentrum eine Quelle hätte und damit nicht mehr divergenzfrei wäre. Die zu lösende Gleichung ist deshalb:

$$\int_{\Omega_{rad}} (\vec{\nabla}\Psi)(\vec{\nabla}\Phi)rdrdz = \int_{\Omega_{rad}} (\vec{H}^* + \vec{M}) \cdot (\vec{\nabla}\Phi)rdrdz, \quad (10)$$

mit Neumannrand 0 auf der Achse  $r = 0$  und Dirichletrand 0 an allen anderen Rändern des 2d-Rechengebietes  $\Omega_{rad}$  zu vorgegebener Magnetisierung  $\vec{M}$  und externem Feld  $\vec{H}^*$  (s. auch Abb 5).

Aus der Lösung  $\Psi$  erhält man anschliessend das gesuchte Magnetfeld  $\vec{H}$  mit Hilfe der Gleichung

$$\vec{H} = -\vec{\nabla}\Psi + \vec{H}^*. \quad (11)$$

In einer Finite Elemente - Simulation wird die exakte Lösung des Problems angenähert durch Funktionen in einem endlich dimensionalen Raum. Diese Funktionen haben kleinen Träger. Die Integrale in (10) werden berechnet durch numerische Quadratur von Basisfunktionen, die auf einer Zelle – in unserem Fall viereckig – definiert sind. Dadurch lässt sich das Problem zurückführen auf das Lösen eines linearen Gleichungssystems, dessen Matrix nur dünn besetzt ist.

## 4 Feld der Spule

Als erstes haben wir eine Funktion entworfen, die das Feld der Spule an jedem Quadraturpunkt berechnet. Der Artikel [Mar01] gibt dazu einen guten Überblick. Für eine gegebene "externe" Stromdichte  $\vec{J}_s$  ist das Magnetfeld an jedem Punkt im Raum gegeben durch das Gesetz von Biot-Savart:

$$\vec{H}_s = \frac{1}{4\pi} \int_{\Omega_s} \vec{J}_s \times \vec{\nabla} \left( \frac{1}{|\vec{r} - \vec{r}'|} \right) d\vec{r}'. \quad (12)$$

Hier bezeichnet  $\Omega_s$  das Gebiet der Spule. Vollkommen äquivalent zu dieser Gleichung ist die Formulierung

$$\vec{H}_s(\vec{r}) = \frac{1}{4\pi} \int_{\Omega_s} \vec{J}_s \times \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^3} d\vec{r}'. \quad (13)$$

Für Spulen mit rechteckigem Querschnitt (und eine solche Spule liegt hier vor) lassen sich diese Formeln mit Hilfe von elliptischen Integralen exakt integrieren. Die Formeln sind jedoch relativ kompliziert, und die numerische Implementation divergiert für bestimmte Punkte im Rechengebiet (z.B. auf den Rändern der Spule). Deshalb haben wir uns dafür entschieden, die Formeln analytisch in r- und z- Richtung zu integrieren und anschliessend numerisch die Integration über den Winkel durchzuführen. Auch hier kommen Singularitäten vor, deren Integral jedoch endlich ist. Mit einer geeigneten Integrationroutine der C-Bibliothek GSL (GNU Scientific Library), die integrierbare Singularitäten behandeln kann, gelang es, das Feld der Spule in Zylinderkoordinaten zu berechnen (s. Abbildung 6).

Die Formeln (siehe [Mar01] für Einzelheiten) sind

$$\vec{H}_s(r, \phi, z) = \frac{J_s}{4\pi} (\vec{H}(R_2, \phi, Z_2) - \vec{H}(R_1, \phi, Z_2) + \vec{H}(R_2, \phi, Z_1) + \vec{H}(R_1, \phi, Z_1)) \quad (14)$$

mit den zylindrischen Komponenten von  $\vec{H}$

$$H_r(r, \phi, z) = \int_{-\pi}^{\pi} \cos(D(\theta) + r \cos \theta \sinh^{-1} \beta_1(\theta)) d\theta \quad (15)$$

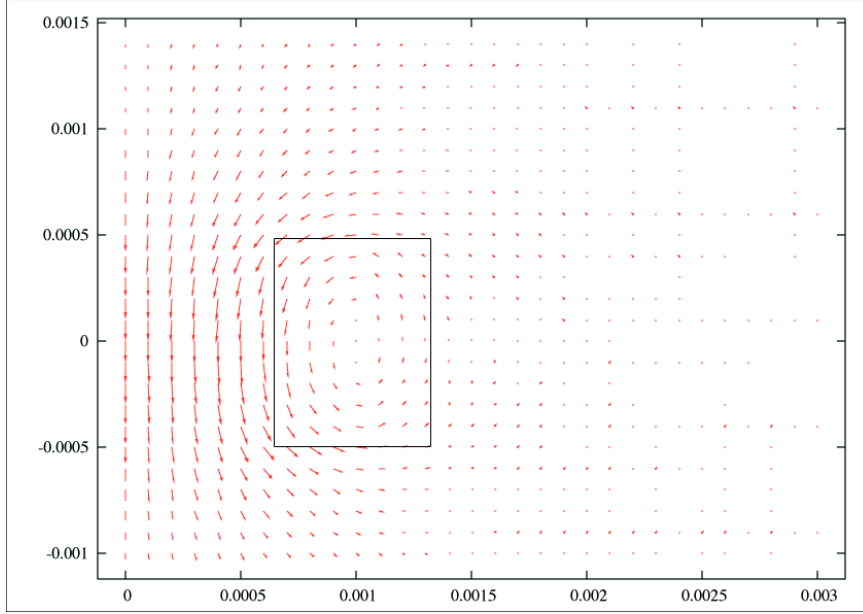


Abbildung 6: Feld der rechteckigen Spule – Feldlinien. Die Feldstärke nimmt ausserhalb der Spule stark ab. r- und z- Achseneinheit in Meter, Feldstärke relativ. Die Position der Spule ist mit einem Rechteck angedeutet – der Übersicht halber ist die Spitze (die sich links von der Spule befinden würde) nicht eingezeichnet.

$$H_\phi(r, \phi, z) = \int_{-\pi}^{\pi} \sin(D(\theta) + r \cos \theta \sinh^{-1} \beta_1(\theta)) d\theta = 0 \quad (16)$$

$$H_z(r, \phi, z) = \int_{-\pi}^{\pi} (\gamma \sinh^{-1} \beta_1(\theta) - r \cos \theta \sinh^{-1} \beta_2(\theta) - r \sin \theta \arctan \beta_3(\theta)) d\theta, \quad (17)$$

mit den Funktionen

$$\gamma = z' - z, \quad \theta = \phi' - \phi, \quad \theta_i = \phi_i - \phi \quad (i = 1, 2), \quad (18)$$

$$B^2(\theta) = r'^2 + r^2 - 2rr' \cos(\theta), \quad D^2(\theta) = \gamma^2 + B^2(\theta), \quad (19)$$

$$G^2(\theta) = \gamma^2 + r^2 \sin^2 \theta, \quad \beta_1(\theta) = (r' - r \cos \theta)/G(\theta), \quad (20)$$

$$\beta_2(\theta) = \gamma/B(\theta), \quad \beta_3(\theta) = \gamma(r' - r \cos \theta)/(r \sin \theta \arctan \beta_3(\theta)) d\theta. \quad (21)$$

Diese Formeln wurden ursprünglich in [Ura82] hergeleitet, enthalten dort aber laut [Mar01] einige grobe Fehler.

## 5 Implementierung in Concepts

Concepts ([FL02], [Con03]) ist ein Open-Source Programm für partielle Differentialgleichungen, das zur Zeit an der ETH entwickelt wird. Unter anderem sind auch Berechnungen mit der Finite Element - Methode mit adaptiver Gitterverfeinerung und adaptiver Verfeinerung des Polynomgrades der Ansatzfunktionen (hp-FEM) möglich. Wir haben uns entschieden, Concepts für die Implementierung zu benutzen, da das Problem mit Matlab und dem auf Matlab basierenden Femlab - FEM-Toolkit nicht lösbar ist, und da Concepts am SAM entwickelt wurde, und deshalb



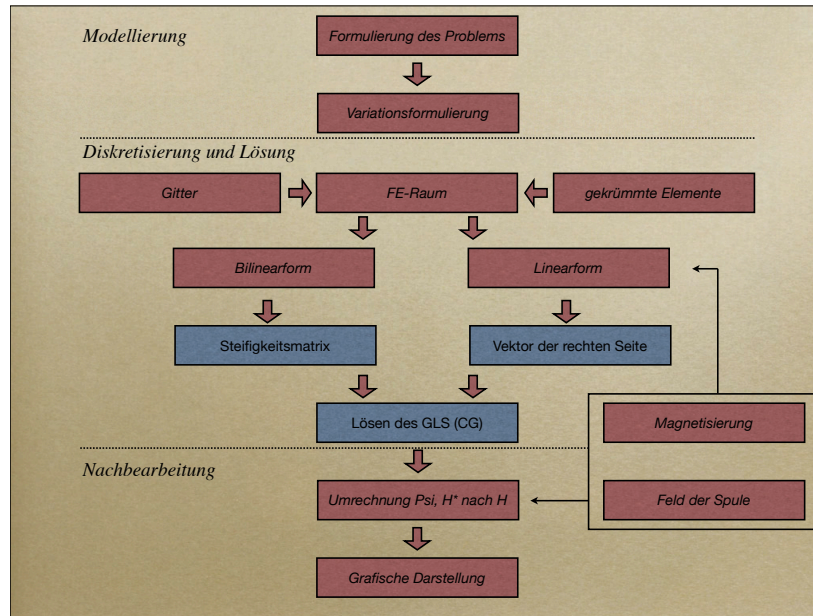


Abbildung 7: Implementierung in Concepts – Arbeitsablauf

das Wissen vorhanden ist. Ausserdem sind hohe Polynomgrade besonders gut geeignet für die “glatten” Strukturen der Spitze, die zwar gekrümmte Elemente, aber keinerlei Ecken aufweist. Die Graphik (3) zeigt die wichtigsten Schritte auf dem Weg zu einer funktionierenden Simulation. Da es in Concepts bereits funktionierende Finite - Element - Beispiele gibt, wie das zweidimensionale Laplace - Problem, war es für uns naheliegend, diese Abläufe und Programmstrukturen zu übernehmen und auf unsere neuen Bedürfnisse umzubauen.

## 5.1 Bilinearform

Als Bilinearform wird die linke Seite der Gleichung (10) bezeichnet, denn

$$b(\Phi, \Psi) = \int_{\Omega_{rad}} (\vec{\nabla}\Psi)(\vec{\nabla}\Phi)rdrdz \quad (22)$$

ist sowohl in  $\Psi$  als auch in  $\Phi$  linear. Die Implementierung der Bilinearform konnten wir aus der existierenden Laplace - Bilinearform für zweidimensionale kartesische Koordinaten aus Concepts erzeugen, denn bis auf die Jakobideterminante  $r$  sind die beiden gleich. Aus der Bilinearform wird die Steifigkeitsmatrix generiert, die darauf mit dem Vektor der rechten Seite dem CG<sup>1</sup> - Gleichungslöser übergeben wird.

## 5.2 Linearform

Als Linearform wird die rechte Seite der Gleichung (10) bezeichnet, da

$$l(\Phi) = \int_{\Omega_{rad}} (\vec{H}^* + \vec{M}) \cdot (\vec{\nabla}\Phi)rdrdz \quad (23)$$

linear in  $\Phi$  ist. Diese Linearform mussten wir von Grund auf neu programmieren, denn einerseits muss dort das Feld der Spule  $\vec{H}^*$  eingefügt werden, andererseits

<sup>1</sup>Conjugate Gradient – Methode der konjugierten Gradienten. Ein schnelles Verfahren zum Lösen von grossen Gleichungssystemen, siehe z.B. [Sch97]

muss aber auch die Magnetisierung  $\vec{M}$  je nach Element (Spitze? Spule? Vakuum?) anders gesetzt werden. Diese Information muss zuerst für jedes Element eingelesen und gespeichert sowie später aus den Elementen extrahiert werden.

### 5.3 Materialkonstanten

Materialkonstanten können in Concepts als Attribute einer Gitterzelle abgespeichert werden. In unserem Modell gibt es bloss Regionen mit Magnetisierung (d.h. die Spitze) und solche ohne Magnetisierung (wie das Vakuum, die Probe oder die Spule). Die Magnetisierung entspricht aufgrund der hohen Feldstärke immer der Sättigungsmagnetisierung, d.h. hat für alle Gebiete denselben Betrag. Jedoch ist die Richtung ortsabhängig (nämlich entlang der Feldlinien). Wir haben uns dafür entschieden, vorerst nur für konstante Magnetisierungsrichtung zu simulieren. Da die Feldlinien innerhalb der magnetisierten Spitze ungefähr in dieselbe Richtung laufen, ist diese Annahme vorerst gerechtfertigt. In einem späteren Schritt wollen wir eine Iteration einbauen, die die Magnetisierung den Feldlinien entlang ausrichtet. Dazu sind wir jedoch bis jetzt nicht gekommen.

### 5.4 Gekrümmte Elemente – Blending

Die Integrale der Linear- und der Bilinearform müssen numerisch möglichst einfach berechenbar sein. Man definiert deshalb Ansatzfunktionen auf einem Referenzelement  $\hat{K}$  (typischerweise  $[0, 1] \times [0, 1]$ ) und eine sogenannte Elementabbildung, die dieses Referenzelement  $\hat{K}$  auf das Element im Raum  $K$  (d.h. an der richtigen Stelle, richtig gedreht und - wie später erklärt - mit gekrümmten Kanten – siehe Abb. (9)) abbildet. Die grundlegende Idee ist nun, dass man die Integration auf dem Element  $K$  mit Hilfe der Jakobimatrix der Elementabbildung auf eine Integration auf dem Referenzelement  $\hat{K}$  zurückführt. Auf diesem Referenzelement kennt man die numerischen Werte der Ansatzfunktionen an sogenannten Quadraturpunkten und kann deshalb die Integration schnell und bequem durchführen. Benötigt werden also an jedem Quadraturpunkt  $(\xi, \eta)$  des Referenzelementes  $\hat{K}$  die Elementabbildung

$$F_K(\xi, \eta) = \begin{pmatrix} r(\xi, \eta) \\ z(\xi, \eta) \end{pmatrix} : \hat{K} \rightarrow K \quad (24)$$

sowie die Jakobimatrix

$$J(\xi, \eta) = \begin{pmatrix} \partial_\xi r & \partial_\eta r \\ \partial_\xi z & \partial_\eta z \end{pmatrix} \quad (25)$$

bzw. deren Inverse und Determinante, die daraus folgen.

Den Hauptteil meiner Zeit verbrachte ich damit, Elemente mit gekrümmten Kanten in Concepts einzubauen. Diese sind für eine gute Approximation der Geometrie der Spitze zwingend notwendig. Die Klassenstruktur ist sehr allgemein gehalten und erlaubt es, zu einem späteren Zeitpunkt eine allgemeine Kantenfunktion und nicht nur die kreisförmige gekrümmten Kanten, die zur Zeit implementiert sind, zu behandeln. Ausserdem funktionieren die in Concepts eingebaute automatische Verfeinerung und beliebig hohe Polynomgrade der Formfunktionen (s. auch Abbildungen 12 und 14 am Ende des Berichts) mit den neuen Elementen.

Das Blending funktioniert dabei folgendermassen: die vier Kanten des Vierecks werden durch Kantenfunktionen  $\vec{x}_j$  beschrieben, für die gilt

$$\vec{x}_j(0) = p_j, \quad \vec{x}_j(1) = p_{j+1}, \quad (26)$$

wobei die  $\vec{p}_j$  die Eckpunkte beschreiben. Dazwischen krümmt sich die Kante mit konstantem Radius. Geometrisch lässt sich die Formel einfach herleiten:

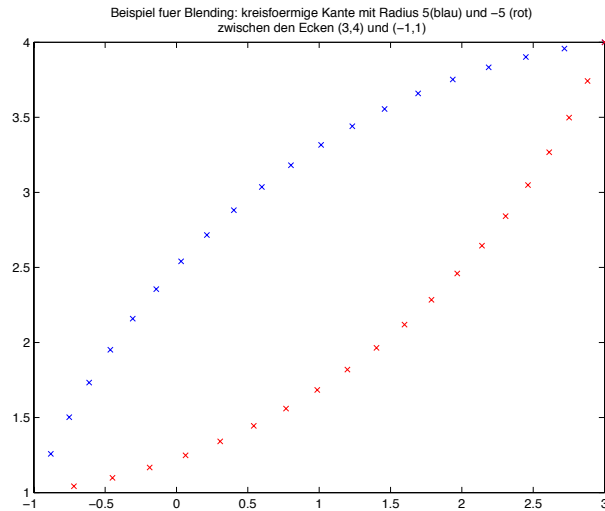


Abbildung 8: Beispiel für die kreisförmig gekrümmten Kanten: Die Kante ist von rechts oben nach links unten gerichtet. Für positives Vorzeichen des Radius befindet sie sich auf der rechten Seite der direkten Verbindungsstrecke, für negatives Vorzeichen auf der linken Seite.

```
x-Koordinate: (1-pos)*vtx_[0][x_coord] +
              (pos)*vtx_[1][x_coord]-sin(alpha_)*arc/sf_;
y-Koordinate: (1-pos)*vtx_[0][y_coord] +
              (pos)*vtx_[1][y_coord]+cos(alpha_)*arc/sf_;
```

mit

```
arc = (r/fabs(r))*(sqrt(r*r-(0.5-pos) * (0.5-pos))-sqrt(r*r-0.25));
```

Die Formel interpoliert also linear zwischen den beiden Endpunkten und addiert eine Korrektur für die Krümmung (“arc”) darüber. Das Resultat: eine kreisförmig gekrümmte Kante wie in Abbildung 8.

Aus den Kanten erhält man die Elementabbildung vom Referenzelement auf das Element im Raum. Dabei wird “linear interpoliert”: Die Korrektur soll 1 sein an der gekrümmten Kante und gegen 0 auf der gegenüberliegenden Seite abfallen. Die Formel dafür lautet:

$$\begin{pmatrix} r(\xi, \eta) \\ z(\xi, \eta) \end{pmatrix} = (1 - \eta)\vec{x}_1(\xi) + \xi\vec{x}_2(\eta) + \eta\vec{x}_3(1 - \xi) + (1 - \xi)\vec{x}_4(1 - \eta) \quad (27)$$

$$-((1 - \xi)(1 - \eta)\vec{p}_0 + \xi(1 - \eta)\vec{p}_1 + \xi\eta\vec{p}_2 + (1 - \xi)\eta\vec{p}_3),$$

Das Resultat ist ein Viereck, dessen vier Kanten in alle Richtungen gebogen sein dürfen - eine Illustration gibt Abbildung 9.

## 6 Details der Implementation

Dieser Abschnitt behandelt die Details der C++ - Implementation meines Projekts. Er wird vermutlich nur für jemanden interessant sein, der direkt mit meinem Code weiterarbeitet.

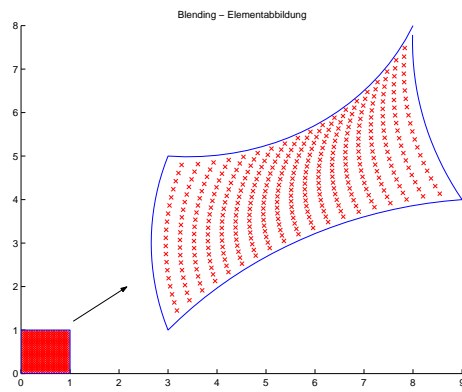


Abbildung 9: Elementabbildung: das Referenzelement wird abgebildet auf ein Viereck mit gekrümmten Kanten. Gezeichnet sind ebenfalls äquidistante Punkte im Referenzelement und deren Bild.

## 6.1 Linear- und Bilinearform

Die Linear- und Bilinearform sind in den Klassen `Int_HplusMgradPhi` sowie `Laplace` in den Dateien `laplace.cc`, `-.h` und `Int_HplusMgradPhi.cc`, `-.h` programmiert.

`Laplace` entspricht `hp2D::Laplace`, bis auf ein weiteres Argument im Konstruktor, das entweder auf `Laplace::cylindrical` oder `Laplace::cartesian` gesetzt wird. Standardmässig ist es `Laplace::cartesian`, d.h. wird die Funktion so aufgerufen wie `hp2D::Laplace` bisher, verhält sie sich genau gleich.

Die Linearform ist eine Ergänzung zum bisherigen Code. Sie ist sehr speziell für das Problem der magnetischen Spitze programmiert. Im Konstruktor kann ebenfalls ein Argument übergeben werden, das die Geometrie festlegt (`Int_HplusMgradPhi::cylindrical` resp. `cartesian`). In der Simulation wird nur der zylindersymmetrische Teil gebraucht, die kartesische Funktionalität ist eingebaut zur Kontrolle und für Testzwecke. Beim Berechnen der Linearform muss der Gradient der Funktion  $\Phi$  bestimmt werden. Dieser wird aus den Formfunktionen und deren Ableitungen, die schon bekannt sind, an jedem Quadraturpunkt des Elements berechnet. Die Magnetisierung wird als Attribut aus der Zelle ausgelesen:

```
Real My=0;
const concepts::Attribute *attrib =
  &elm.cell().connector().attrib();
if(*attrib!=0){
  std::cout<<"attribut value"<<
    *((Real*)(attrib->attrib()))<<std::endl;
  My=MAGNETIZATION;
}
```

Das Feld der Spule wird an jedem Quadraturpunkt berechnet durch

```
Real current=-5.0;
coil_params coil(1,1.5,0.5,0.7);
coilfield(r,z,Hr,Hz,coil, current);
```

wobei  $x$  und  $y$  die Punkte im Raum sind, an denen das Feld berechnet werden soll,  $H_r$  und  $H_z$  die Felder, `current` die Stromdichte und `coil` eine Struktur, die die Parameter der Spule beschreibt. Für gegebene Spulengeometrie, Strom und Ort im Raum wird dadurch das Magnetfeld  $\vec{H} = (H_r, H_z)$  berechnet.

## 6.2 Gekrümmte Elemente

Die gekrümmten Elemente sind in Concepts in der Klasse `MapExactBlending2d` programmiert. Diese Klasse implementiert die Elementabbildung vom Referenzelement  $[0, 1] \times [0, 1]$  auf das Element im Raum, sowie die Jakobimatrix für gegebene Kantenfunktionen der vier Kanten. Kantenfunktionen können dabei beliebige von `MappingEdge2d` abgeleitete Klassen sein, die die pure virtual Funktionen der Klasse implementieren. Ein Beispiel dafür ist `CircleMappingEdge2d`.

### 6.2.1 Die Elementabbildung für gekrümmte Kanten

Die vektorwertige Elementabbildung für gekrümmte Kanten vom Referenzelement auf das Element im Raum ist gegeben durch

$$\begin{pmatrix} r(\xi, \eta) \\ z(\xi, \eta) \end{pmatrix} = (1 - \eta)\vec{x}_1(\xi) + \xi\vec{x}_2(\eta) + \eta\vec{x}_3(1 - \xi) + (1 - \xi)\vec{x}_4(1 - \eta) \quad (28)$$

$$-((1 - \xi)(1 - \eta)\vec{p}_0 + \xi(1 - \eta)\vec{p}_1 + \xi\eta\vec{p}_2 + (1 - \xi)\eta\vec{p}_3),$$

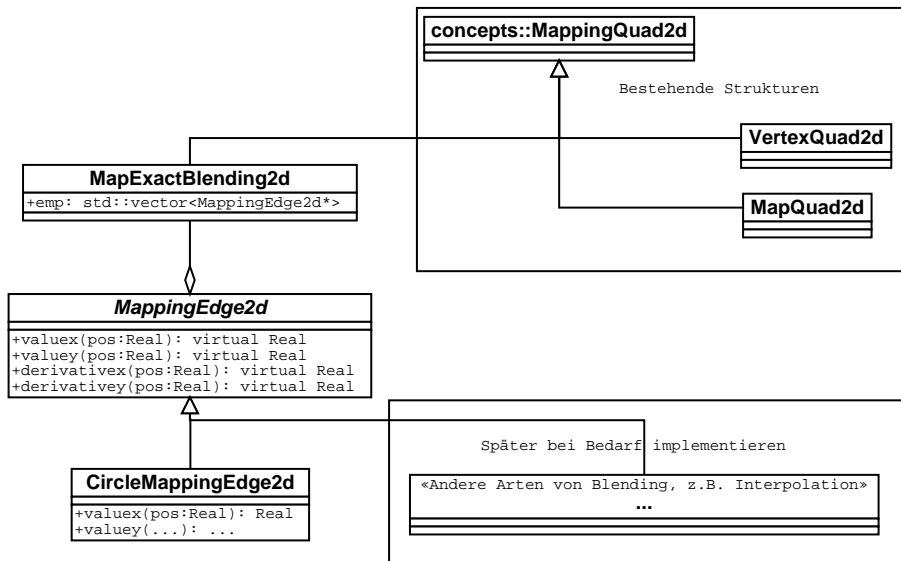


Abbildung 10: Klassen in Concepts – Blending

wobei die vektorwertigen Funktionen  $\vec{x}_j$  die Krümmung der Kanten beschreiben, und die  $\vec{p}_j$  die Eckpunkte sind. Die in (28) beschriebene Abbildung ist in `MapExactBlending2d::operator()` implementiert.

### 6.2.2 Kreisförmige Kanten

Jedes Element hat vier Kanten, davon dürfen alle vier nach innen oder aussen kreisförmig gekrümmt sein. Dies ist in `CircleMappingEdge2d` implementiert. Im Konstruktor werden zwei Eckpunkte und ein Radius übergeben. Je nach Vorzeichen des Radius geht die Krümmung in andere Richtungen (Konvention siehe Abb 8). Das Berechnen der gekrümmten Kanten geschieht in der Funktion `valuex()` und `valuey()`. Im Fall von `valuex()` sieht dies folgendermassen aus:

```

//Value of the function (x,y)=(x,y)(pos) with 0<=pos<=1
// - x-coordinate, for a circular arc.
Real valuex(Real pos) const{
    Real deltapos;
    if (r_==0){ //straight line: linear interpolation
        return (1-pos)* vtx_[0][x_coord] + (pos)*vtx_[1][x_coord];
    }
    else{ //circular arc: Linear interpolation and arc
        deltapos=arc(pos, r_);
        return (1-pos)*vtx_[0][x_coord] + (pos)*vtx_[1][x_coord]
        -sin(alpha_)*deltapos/sf_;
    }
}
}

```

Im Code wird zunächst linear zwischen den Endpunkten interpoliert und darauf die Korrektur für den gekrümmten Bogen addiert.

### 6.3 Einlesen der Materialkonstanten und der gekrümmten Elemente

Um Gitter und Gitterparameter einfach ändern zu können, haben wir uns entschieden, eine neue Einleseroutine zu schreiben. Das Beispiel beschreibt ein simples Gitter, das die wesentlichen Eigenschaften der STM-Geometrie aufweist.

Wie bisher werden zuerst die Eckpunkte eingelesen (Nummer, x- und y- Koordinaten) – Datei `coords.dat`:

```
1  0.000000 -1.000000
2  0.300000 -1.000000
3  0.000000  0.000000
4  2.000000 -1.000000
5  0.000000 -0.650000
6  0.176777 -0.576777
7  0.500000 -0.600000
8  2.000000 -0.600000
9  0.000000 -0.400000
10 0.250000 -0.400000
11 0.750000 -0.400000
12 2.000000 -0.400000
13 0.000000  0.000000
14 0.500000  0.000000
15 2.000000  0.000000
16 0.000000  1.000000
17 0.500000  1.000000
18 2.000000  1.000000
```

Darauf die Elemente aus den Ecken zusammengesetzt (Elementnummer und die vier Eckpunktindices) – Datei `elements.dat`:

```
1  1 5 6 2
2  2 6 10 7
3  2 4 8 7
4  5 6 10 9
5  7 8 12 11
6  9 10 14 13
7  7 10 14 11
8  11 12 15 14
9  13 14 17 16
10 14 15 18 17
```

Die Neumann - Randbedingungen werden in der Datei `neumann.dat` definiert (Nummer, Kante von Eckpunkt A zu Eckpunkt B, und Neumannrand 0):

```
1 1 5 0
2 5 9 0
3 9 13 0
4 13 16 0
```

Schliesslich müssen neu die gekrümmten Kanten eingelesen werden (Nummer, Eckpunkt A und Eckpunkt B, Krümmungsradius – Datei `radius.dat`):

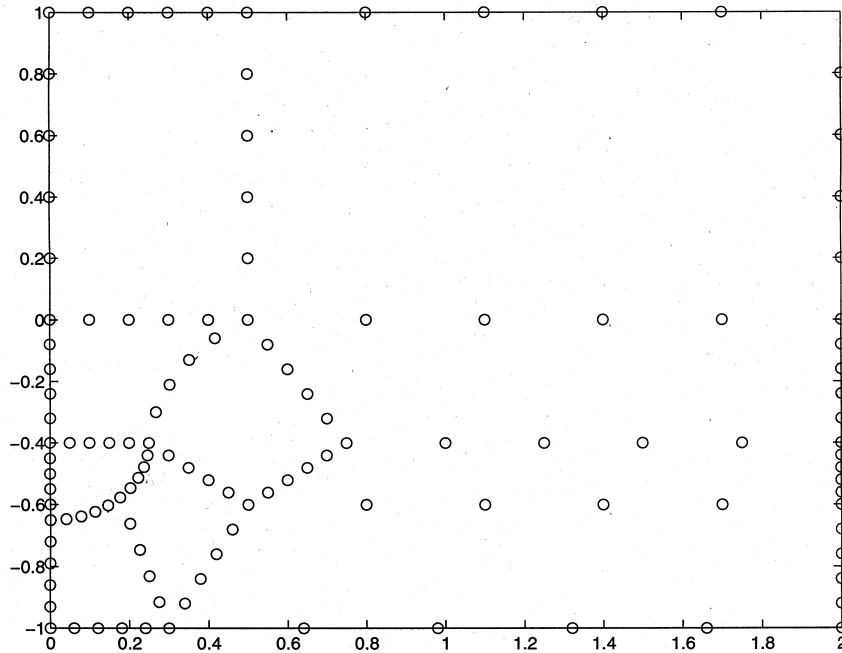


Abbildung 11: Gitter wie in Abschnitt 6.3. Das Gitter weist keine hängenden Knoten auf – der Eckpunkt  $(0.2, -0.6)$  ist Eckpunkt eines Vierecks.

```
1 6 5 0.25
2 6 10 -0.25
3 10 14 0.5
```

und die Materialkonstanten (vom Typ `concepts::Real`, d.h. normalerweise double precision floating point number) den Elementen zugeordnet werden (Nummer, Elementnummer, Materialkonstante) – Datei `material.dat`:

```
1 4 1
2 6 1
3 9 1
```

Nicht alle Elemente müssen Material besitzen. Da zur Zeit die Grösse der Magnetisierung einheitlich ist, wird nur überprüft, ob “Material” (d.h. hier Magnetisierung) vorhanden ist, und nicht, wie gross der Parameter ist. Die Elemente 4, 6, und 9 bilden im Gitter die Spitze und haben  $\vec{M} \neq \vec{0}$ .

Aus dieser Information generiert das Programm darauf das Gitter mit gekrümmten Elementen – in diesem Fall in Abbildung 11 wiedergegeben.

## 7 Stand des Projekts und Schlussbemerkung

In den drei Wochen, die ich am SAM verbracht habe, konnten wir die Simulation leider nicht beenden. Dies lag vor allem daran, dass ich mich nicht schnell genug in



Concepts einarbeiten konnte. Wir konnten aber für das abgebildete Testgitter die Lösung rechnen und ausgeben, jedoch kamen wir nicht mehr dazu, das Postprocessing und die graphische Darstellung zu programmieren. Ich werde in nächster Zeit versuchen, zusammen mit Thomas Michlmayr jemanden zu finden, der das Projekt beenden möchte - doch die Chancen stehen nicht gut, denn leider gibt es nur wenige Studenten, die sich sowohl in der Oberflächenphysik als auch mit finiten Elementen ein wenig auskennen und sich für beides interessieren. Ich rechne mit einem Arbeitsaufwand von weiteren zwei Wochen (inklusive Einarbeiten und Schlussbericht), um das Projekt zu beenden.

Die Zeit mit Kersten wahr äusserst spannend und lehrreich und sehr angenehm. Zwar habe ich in einer Vorlesung schon einiges über Finite Elemente gelernt, aber selbst einen solchen Code kennenzulernen ist doch etwas ganz anderes als nur über Theorie zu sprechen. Ausserdem gaben mir diese drei Wochen einen guten Einblick in die Arbeitsweise an einem Mathematikinstitut, von der man als "normaler" Student leider kaum etwas mitbekommt. Die Betreuung war hervorragend - besonders die angeregten Diskussionen an der Wandtafel und das aufgestellte Team werden mir lange in Erinnerung bleiben. Vielen Dank, Kersten!

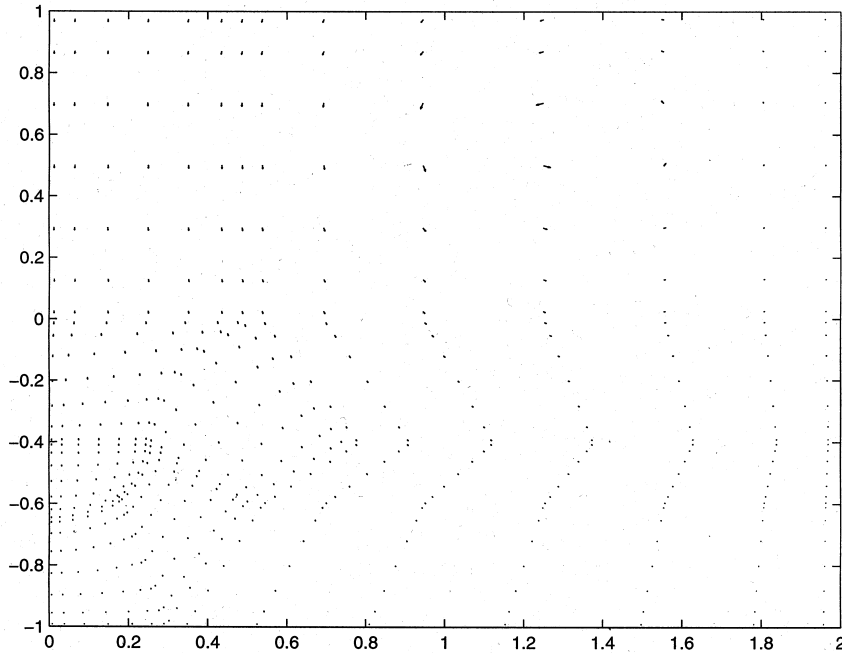


Abbildung 12: An jedem Quadraturpunkt des Gitters ist das Feld der Spule (die sich oben rechts im Bild befindet) abgebildet. Die Magnetisierung der Spitze ist hier noch nicht berücksichtigt.

## Literatur

- [Con03] Concepts Development Team. Concepts homepage. Internet, 2003. <http://www.concepts.math.ethz.ch/>.
- [FL02] Philipp Frauenfelder and Christian Lage. Concepts—an object-oriented software package for partial differential equations. *M2AN Math. Model. Numer. Anal.*, 36(5):937–951, 2002.
- [Mar01] Marzia Fontana. Integration methods for the calculation of the magnetostatic field due to coils. unpublished, 2001.
- [Sch97] H.R. Schwarz. *Numerische Mathematik*. Teubner, 1997.
- [Ura82] L. U. Urankar. Vector potential and magnetic field of current-carrying finite arc segment in analytical form. part iii: Exact computation for rectangular cross section. *IEEE Trans. Magn.*, 18:1860 – 1867, 1982.

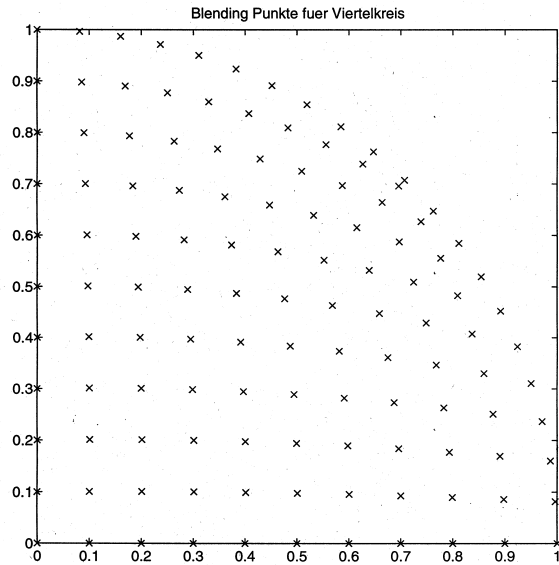


Abbildung 13: Ein Beispiel für die Randapproximation: Ein Viereck mit zwei gekrümmten Kanten bildet einen Viertelkreis.

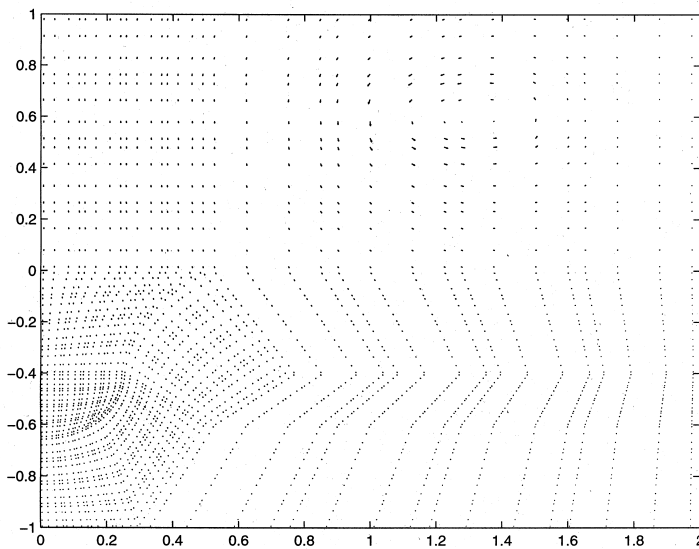


Abbildung 14: Gitterverfeinerung – Dieses Bild zeigt ebenfalls das Feld an jedem Quadraturpunkt, allerdings wurde hier bei kleinem Polynomgrad zweimal automatisch verfeinert. Die Spule liegt oben rechts (Spulenmitte ca. bei (1.3, 0.6); die Spitze unten links).