# Ausarbeitung zum Softwarepraktikum Verfahren und Anwendung der Feldsimulation

**Denis Andric**
**Marc Bodem**
**Theodor Sperling**

TECHNISCHE
UNIVERSITÄT
DARMSTADT

## Inhaltsverzeichnis

# 1 Grundlagen der Methode der Finiten Integration 1

## 1.1 Vorbereitungsaufgaben

### 1.1.1 Überzählige Kanten

**1.** Skizzieren Sie ein zweidimensionales kartesisches Gitter mit $3 \times 4$ Punkten und tragen Sie alle Kantenindizes für die $x$- und $y$-Kanten nach dem kanonischen Indizierungsschema aus Gl. (2.5) ein. Machen Sie sich klar, welche Indizes zu nicht existierenden Kanten gehören und markieren Sie diese.

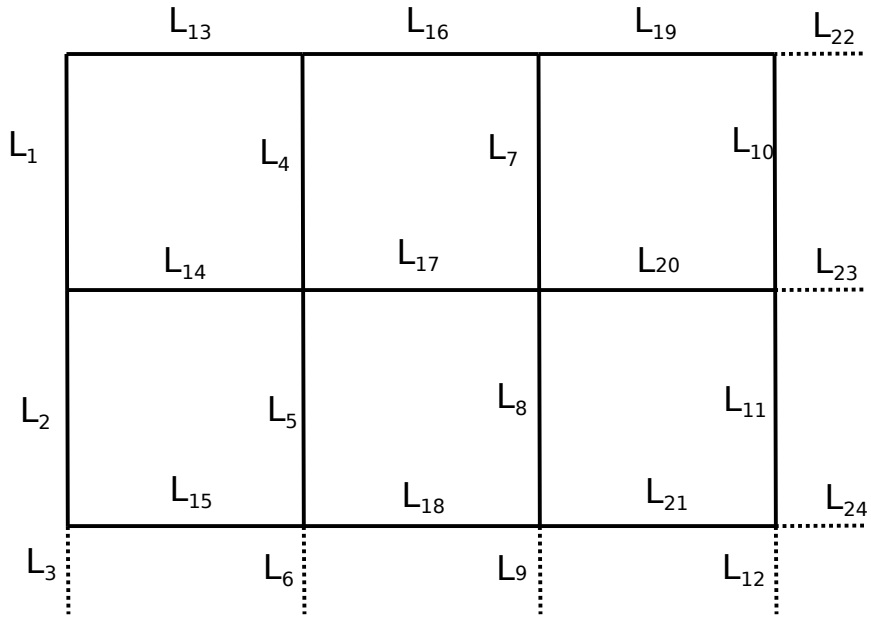


**Abbildung 1.1:** 3×4 Gitter mit kanonischer Kantenindizierung.

**2.** Überlegen Sie sich für ein $N_x \times N_y$-Gitter eine Formel für die Anzahl der Indizes, zu denen keine Kanten gehören. Geben Sie diese Formel auch für den Sonderfall $N_{xy} = N_x = N_y$ in Abhängigkeit von $N_\mathrm{P} = N_{xy}^2$ an. Geben Sie darüber hinaus auch eine Formel an, um die Indizes aller Geisterkanten nach dem kanonischen Indizierungsschema zu berechnen.

Die Anzahl der Geisterkanten eines $N_x \times N_y$-Gitters ergibt sich zu $N_x + N_y$ da jeweils an den in positiver Richtung liegenden Aussenseiten eine Reihe an Geisterkanten besitzt.
Für den Fall von $N_x = N_y$ ergibt sich die Anzahl der Geisterkanten zu $\sqrt{N_p} \cdot 2$.

### 1.1.2 Dreiecksgitter

Gegeben sind die beiden Dreiecksgitter in Abb. 1.2, **wobei zunächst nur das linke Gitter betrachtet werden soll.**

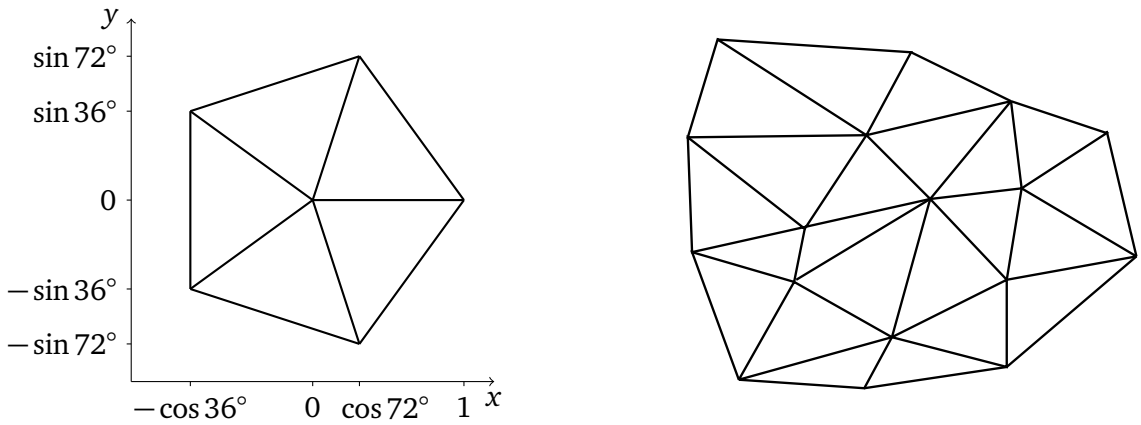


**Abbildung 1.2:** Dreiecksgitter von zwei verschiedenen Rechengebieten.

**3.** Nummerieren Sie die Flächen und Kanten des Gitters beliebig und ordnen Sie den Kanten eine Orientierung zu.

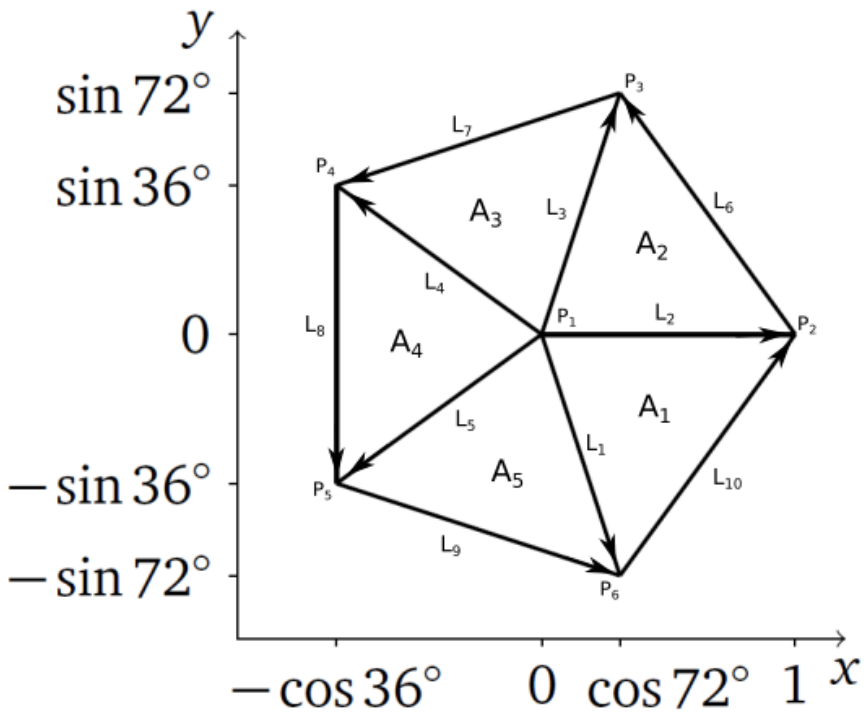


**Abbildung 1.3:** Gegebenes Gitter mit beliebig gewählter Indizierung für Punkte, Kanten und Flächen.

**4.** Erstellen Sie die Punkteliste (3-spaltige Tabelle mit Index, $x$-Koordinaten und $y$-Koordinaten). Stellen Sie auch die Indexlisten Kanten-zu-Knoten und Flächen-zu-Kanten auf (auch Inzidenzen genannt). Beachten Sie dabei die Orientierung der Kanten und Flächen. Die Kanten sind von Punkt 1 zu Punkt 2 gerichtet. Bei der Flächen-zu-Kanten-Inzidenz werden Kanten, die gegen die Umlaufrichtung der Fläche zeigen, mit einem negativen Vorzeichen vor dem Index gekennzeichnet.

| $P_i$ | X | Y |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 1 | 0 |
| 3 | $\cos 72°$ | $\sin 72°$ |
| 4 | $-\cos 36°$ | $\sin 36°$ |
| 5 | $-\cos 36°$ | $-\sin 36°$ |
| 6 | $\cos 72°$ | $-\sin 72°$ |

**Tabelle 1.1:** Punktliste

| $L_i$ | $P_{in}$ | $P_{out}$ |
|---|---|---|
| 1 | 1 | 6 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| 4 | 1 | 5 |
| 5 | 1 | 6 |
| 6 | 2 | 3 |
| 7 | 3 | 4 |
| 8 | 4 | 5 |
| 9 | 5 | 6 |
| 10 | 6 | 2 |

**Tabelle 1.2:** Kanten-zu-Knoten Indextabelle

| $A_i$ | $L_1$ | $L_2$ | $L_3$ |
|---|---|---|---|
| 1 | 1 | -2 | 10 |
| 2 | 2 | -3 | 6 |
| 3 | 3 | -4 | 7 |
| 4 | 4 | -5 | 8 |
| 5 | -1 | 5 | 9 |

**Tabelle 1.3:** Flächen-zu-Kanten Indextabelle

**5.** Erstellen Sie aus der Kanten-zu-Knoten-Inzidenz die Gradientenmatrix **G**. Gehen Sie von einem Potentialvektor $\boldsymbol{\varphi}$ der Dimension $N_\mathrm{P}$ aus, der die Werte einer Potentialfunktion in allen Gitterpunkten enthält. Legen Sie die Matrix **G** so fest, dass die Multiplikation $-\mathbf{G}\boldsymbol{\varphi}$ gerade den Vektor $\hat{\mathbf{e}}$ ergibt, was der kontinuierlichen Formel $\vec{E} = -\operatorname{grad}\varphi$ entspricht.

Aus der Kanten-zu-Knoten Tabelle **??** bekommt man Gradientenmatrix **G**:

$$\mathbf{G} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

*Fügen Sie hier Ihre Lösung ein*

**6.** Konstruieren Sie mithilfe der Flächen-zu-Kanten-Inzidenz die Curlmatrix **C**.
Zur Erinnerung: $\mathbf{C}\widehat{\mathbf{e}} = -\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\widehat{\mathbf{b}}}$.

*Fügen Sie hier Ihre Lösung ein* Aus der Flächen-zu-Kanten Tabelle **??** bekommt man:

$$\mathbf{C} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

**7.** Überprüfen Sie, ob genau wie im kontinuierlichen Fall die Beziehung $\mathrm{rot}\,\mathrm{grad} = 0$ auch für die aufgestellten diskreten Matrizen $\mathbf{CG} = \mathbf{0}$ gilt.

$$\mathbf{CG} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} -1+1 & 1-1 & 1-1 & 0 & 0 & 1-1 \\ -1+1 & 1-1 & -1+1 & 0 & 0 & 0 \\ -1+1 & 0 & 1-1 & -1+1 & 0 & 0 \\ -1+1 & 0 & 0 & 1-1 & -1+1 & 0 \\ 1-1 & 0 & 0 & 0 & 1-1 & -1+1 \end{bmatrix} = \mathbf{0}$$

### 1.1.3 Duale Gitter

Ein mögliches Gestaltungsprinzip für das duale Gitter eines Dreiecksgitters resultiert aus der Forderung, dass die dualen Kanten die normalen Flächen (2D = normale Kanten) orthogonal durchstoßen. Es wird daher versucht, die dualen Kanten aus den *Mittelsenkrechten* der Dreiecke zu konstruieren, die sich bekanntermaßen in einem Punkt – dem neuen dualen Gitterpunkt – schneiden. Voraussetzung für dieses Vorgehen ist jedoch, dass der Schnittpunkt der Mittelsenkrechten auch innerhalb des Dreiecks liegt, was nicht immer erfüllt ist.

### 1.1.4 Ab hier sollen beide Gitter aus Abb. 1.2 betrachtet werden.

**8.** Zeichnen Sie das orthogonale duale Gitter ein, wenn möglich nach der oben beschriebenen Konstruktionsvorschrift. Markieren Sie die dualen Gitterkanten, die die Eigenschaft der Orthogonalität nicht mehr erfüllen.

Die dualen Gitter der gegebenen primären Gitter sind in den Abbildungen 1.4 und 1.5 gegeben.

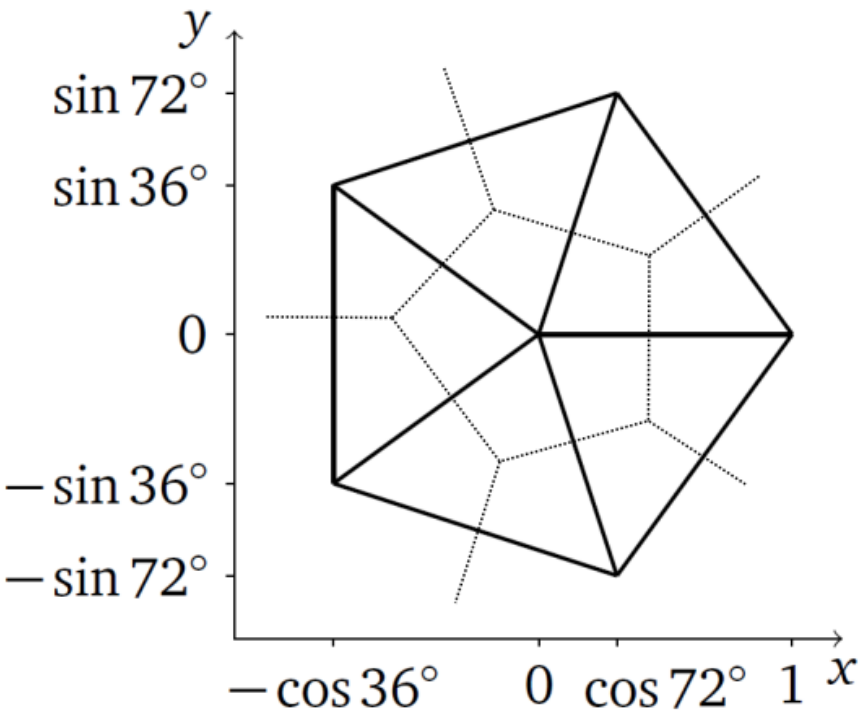


**Abbildung 1.4:** Duales Gitter für das erste gegebene Gitter.

**9.** Überlegen Sie sich, wie $N_\mathrm{V}$, $N_\mathrm{A}$, $N_\mathrm{L}$ und $N_\mathrm{P}$ mit den entsprechenden Größen des dualen Gitters $\tilde{N}_\mathrm{V}$, $\tilde{N}_\mathrm{A}$, $\tilde{N}_\mathrm{L}$ und $\tilde{N}_\mathrm{P}$ im Fall von 3D-Gittern zusammenhängen. Besonderheiten am Rand sind hierzu zu vernachlässigen. Wie verhalten sich die Größen im Fall von 2D-Gittern?

Beim dualen Gitter im dreidimensionalen entspricht die Anzahl der dualen Punkte $\tilde{N_P}$, aufgrund der Definition der dualen Gitterpunkte in den primären Volumen, der Anzahl der primären Volumen $N_V$.

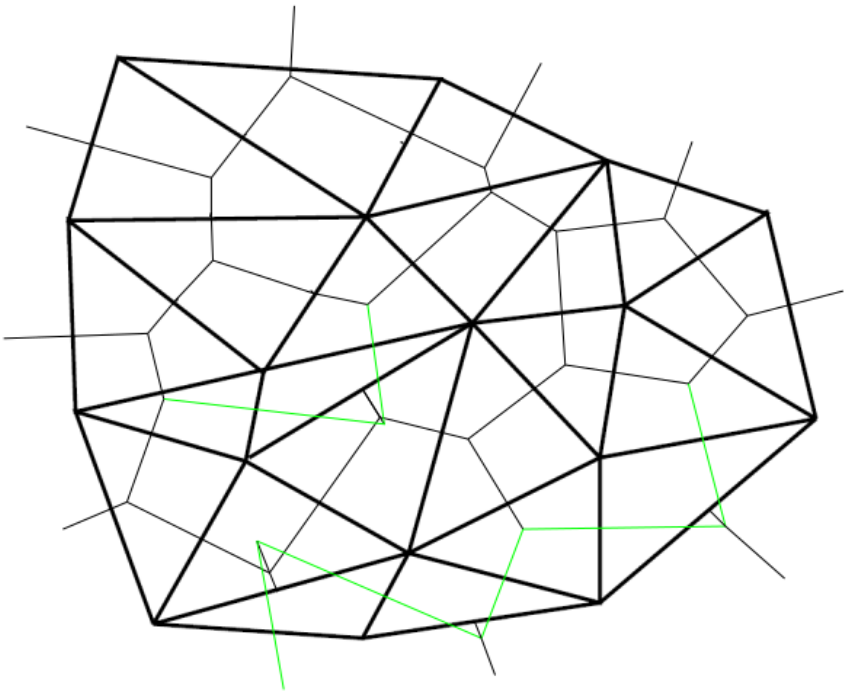

**Abbildung 1.5:** Duales Gitter für zweites gegebenes Gitter. Kanten, die die Orthogonalität nicht erfüllen, sind grün markiert.

Durch die Definition der dualen Kante, die durch die primären Flächen verlaufen, ergibt sich $\tilde{N}_L = N_A$. Die dualen Volumen befinden sich jeweils um einen primären Punkt herum weshalb die Anzahl $\tilde{N}_V$ der dualen Volumen mit der Anzahl $N_P$ der primären Punkte übereinstimmt.

## 1.2 Aufgaben während der Praktikumssitzung

### 1.2.1 Datenstruktur, Visualisierung des Gitters

**1.** Schreiben Sie eine Methode zur Abspeicherung dreidimensionaler, kartesischer Gitter in einem struct

$$[\text{msh}] = \text{cartMesh}(\text{xmesh}, \text{ymesh}, \text{zmesh}) \tag{1.1}$$

und verwenden Sie die Definitionen der Eingangsparameter aus Abschnitt 2.2.3. Die Struktur msh hält nach Aufruf dieser Funktion das durch xmesh, ymesh und zmesh definierte Gitter. Für spätere Routinen muss in msh auch die Gitterpunkteanzahl in jede Raumrichtung, d. h. nx, ny und nz, abgespeichert werden.

*Fügen Sie hier Ihre Lösung ein*

**2.** Implementieren Sie die Methode

$$\text{plotMesh(msh)}\,, \tag{1.2}$$

welche ein übergebenes kartesisches Gitter msh visualisiert. Verwenden Sie hierzu den line-Befehl und eine 3-fach Schleife über die Indizes $i, j, k$.

*Fügen Sie hier Ihre Lösung ein*

**3.** Nutzen Sie cartMesh zur Erzeugung eines nicht äquidistanten Gitters mit {3,4,5} Punkten in $\{x,y,z\}$-Richtung und visualisieren Sie es mit plotMesh. Nutzen Sie hierfür die bereits gegebene Datei exampleMesh.m.

### 1.2.2 Die topologischen Matrizen C, $\widetilde{\mathbf{C}}$, S und $\widetilde{\mathbf{S}}$

**4.** Schreiben Sie eine Methode

$$[\text{c},\ \text{s},\ \text{st}]\ =\ \text{geoMats(msh)}\,, \tag{1.3}$$

die die Operatormatrizen für ein kanonisches, kartesisches Gitter msh erzeugt. Die Rückgabewerte c, s und st sind die Matrizen $\mathbf{C}$, $\mathbf{S}$ und $\widetilde{\mathbf{S}}$ und entsprechend Abschnitt 2.2.7 definiert. Diese werden mithilfe der $\mathbf{P}_\xi$-Matrizen erzeugt. Wieso ist es nicht sinnvoll, $\widetilde{\mathbf{C}}$ und $\mathbf{G}$ zurückzugeben?

**Hinweis:** Schon bei mittleren Problemgrößen muss hier unbedingt mit MATLAB®s speziellem Speicherformat für *dünnbesetzte* Matrizen gearbeitet werden (Befehle wie sparse, speye, usw.) Im Allgemeinen geben MATLAB®-Befehle immer dann Matrizen im sparse-Format zurück, wenn *alle* ihre Argumente ebenfalls sparse sind. Mehr zu diesem Thema ist in der MATLAB®-Dokumentation zu finden.

Die Rückgabe der dualen Rotationsmatrix $\tilde{C}$ ist nicht sinnvoll, da diese durch $\tilde{C} = C^T$ und die Rückgabe von $C$ bereits einfach erzeugt werden kann. Selbiges gilt für die primäre Gradientenmatrix $G$ die aus $S^T$ durch $G = -\tilde{S^T}$ erzeugt werden kann.

**5.** Lassen Sie sich die Matrizen für eine kleine Problemgröße ($N_\text{P} < 50$) direkt ausgeben und visualisieren Sie die Matrizen für eine mittlere Problemgröße ($N_\text{P} < 5000$) mit dem Befehl spy. Welche speichertechnisch günstige Eigenschaft würde ohne das kanonische Indizierungsschema verloren gehen? Ermitteln Sie wie viel Speicherplatz jeweils von MATLAB® benötigt wird (sparse und full-Format). Legen Sie für die Ausarbeitung eine Tabelle mit dem jeweils benötigten Speicherplatz an. Nutzen Sie für diese Tests die bereits gegebene Datei exampleSparse.m.

*Fügen Sie hier Ihre Lösung ein*

| Anzahl der Elemente | Speicherbedarf im full-Format | Speicherbedarf im Sparse-Format |
|---|---|---|
| 40 | 115.200 Byte | 3.632 Byte |
| 4913 | 1.737.904.968 Byte | 466.736 Byte |

**Tabelle 1.4:** Speicherverbrauch einer C-Matrix in unterschiedlichen Speicherformaten

Wie in Abb. 1.7 zu sehen entsteht durch die kanonische Indizierung eine Matrix, die sich aus einer Mehrzahl Elementen auf den Nebendiagonalen zusammen setzt. Dies ermöglicht eine sehr günstige Speicherung.
Wie in Tabelle **??** zu erkennen ist, werden für die Speicherung einer mittleren C-Matrix mit 4913 Elementen im Vollformat 1,7 GB an Speicherplatz benötigt. Die selbe Matrix kann im Sparse-Format jedoch mit nur 0,4 MB gespeichert werden.

**6.** Berechnen Sie

1. $\mathbf{C}(-\widetilde{\mathbf{S}}^{\mathrm{T}})$ und
2. $\mathbf{S}\,\mathbf{C}$ bzw. $\widetilde{\mathbf{S}}\,\widetilde{\mathbf{C}}$ .

Was bedeutet das für die topologischen Matrizen in Hinblick auf die jeweiligen analytischen Operatoren? Erinnern Sie sich, welche analytischen Operatoren den jeweiligen Matrizen entsprechen.

Bei der Berechnung von $C(-\tilde{S^T})$ als auch $SC$ ist das Ergebnis eine Nullmatrix. Dies zeigt die Konsistenz der Diskreten Operationen mit den analytischen, da diese mit

$$SC = 0 \sim \nabla \cdot \nabla \times = 0$$
$$C(-\tilde{S^T}) = C(G) = 0 \sim \nabla \times \nabla = 0$$

einander entsprechen.

### 1.2.3 Unbelegte Kantenelemente

**7.** Als Fortführung von Aufgabe 2 aus der Vorbereitung konstruieren Sie eine Routine, die die überzähligen Kanten erfasst.

$$\text{edg} = \text{boundEdg(msh)} \tag{1.4}$$

gibt demnach für ein gegebenes Gitter msh einen Vektor edg zurück, der entsprechend der kanonischen Indizierung true für normale und false für die überzähligen Kanten enthält.

**Hinweis:** Benötigt wird in diesem Versuch nur der zweidimensionale Fall nz=1, jedoch ist es für spätere Versuche hilfreich auch den dreidimensionalen Fall zu implementieren. Zusätzlich ist es sinnvoll, Erfahrungen mit Vektoroperationen zu sammeln, da diese in MATLAB® in der Regel schneller sind als Schleifen. Das logical-Format (in anderen Programmiersprachen auch als boolean bekannt) hat den Vorteil, dass nur 1 Byte (im Vergleich zu 8 Bytes für double) pro Eintrag benötigt wird.

**8.** Zählen Sie mit boundEdg die unbelegten Kanten und vergleichen Sie Ihr Ergebnis mit der Formel aus der 2. Vorbereitungsaufgabe, indem Sie die relative Anzahl der unbelegten Kanten (inkl. Geisterkanten in $z$-Richtung) über die Anzahl aller Kanten für ein zweidimensionales Gitter msh mit $N_{xy} = N_x = N_y$ darstellen. plotBoundEdg soll diese Aufgaben dann in einem Skript zusammenfassen.

Der Funktionsverlauf aus Abbildung 1.8 ist proportional zu $\frac{\sqrt{x}+x}{x}$. Aus Vorbereitungsaufgabe 2 wurde klar, dass bei einem Quadratischen Gitter die Anzahl aller Geisterkanten gleich $2\sqrt{N_p}$ ist, wobei hier noch keine Geisterkanten in z-Richtung betrachtet wurden. Betrachtet man auch die Geisterkanten in z-Richtung, wobei die Höhe in z-Richtung 1 beträgt, so kommt man auf eine Gesamtanzahl von $2\sqrt{N_p}+N_p$. Dies kann man nun auf die Gesamtanzahl der Kanten beziehen und kommt auf $\frac{2\sqrt{N_p}+N_p}{3N_p}$. Die Werte aus Vorbereitungsaufgabe 2 bestätigen also das Diagramm.

### 1.2.4 Einprägen gegebener Feldverteilungen

**9.** Schreiben Sie eine Methode, die für ein vorgegebenes kontinuierliches $\vec{E}$-Feld field die entsprechenden integralen Zustandsgrößen fieldBow in einem 3D-Gitter msh berechnet und in einem Vektor gemäß Gl. (2.8) abspeichert. Implementieren Sie:

$$[\text{fieldBow}] = \text{impField}(\text{msh}, \text{field}) \tag{1.5}$$

**Hinweis:** field soll hierbei eine *anonymous function* sein, welche den Punkt mit $x$-,$y$- und $z$-Koordinate übergeben bekommt und einen Vektor mit $x$-,$y$- und $z$-Komponente zurückgibt. Zum Beispiel:

field = @(x,y,z)([1./x.^2, 0.01*x, y+z])
Aufruf mit field (1, 3, 4.5) oder field ([3,6]', [1,3]', [2,4]') .

Werten Sie für die notwendige Integration über eine Kante das gegebene Feld an den Kantenmittelpunkten aus und multiplizieren Sie den Wert mit der Kantenlänge anstatt das Feld tatsächlich zu integrieren.

**10.** Verwenden Sie Ihre Methode impField um folgende Felder zu diskretisieren:

1. $\vec{E}(\vec{r}) = \frac{5}{2}\,\vec{e}_x - 1{,}3\,\vec{e}_y + 2\,\vec{e}_z$,
2. $\vec{E}(\vec{r}) = 3\,\sin\Big(\frac{\pi}{x_{\max}-x_{\min}}\,(x - x_{\min})\Big)\,\vec{e}_y$ ,

wobei die Einheiten hier vernachlässigt werden. Mit Hilfe der vorgegebenen Routine plotEdgeVoltage sollen Sie Ihre Implementation optisch verifizieren. Fassen Sie diese Aufgabe in einem Skript plotImpField zusammen.

Die vorgegebenen Feldverteilungen sind in den Abbildungen 1.9 und 1.10 dargestellt.

## 1.3 Fragen zur Ausarbeitung

**1.** In den Vorbereitungsaufgabe zum dualen Gitter wurden Besonderheiten am Rand des Rechengebietes vernachlässigt.

Wie sollte das duale Gitter am Rand gewählt werden, damit die magnetische Randbedingung automatisch erfüllt ist. Machen Sie eine kleine Skizze für ein einfaches zweidimensionalen kartesisches Gitter sowie für das Dreiecksgitter aus Bild 1.2 a). Ist diese Wahl des dualen Gitters am Rand immer notwendig?

*Fügen Sie hier Ihre Lösung ein*

## 1.4 Fazit

Wie aus den Aufgaben ersichtlich wird, ist die Generierung von Mesh Matrizen relativ einfach. Weiterhin können mit einfachen Operationen die Matrizen $\mathbf{C}$, $\widetilde{\mathbf{C}}$, $\mathbf{S}$, $\widetilde{\mathbf{S}}$ und $\mathbf{G}$ erstellt und verwendet werden, um in Zukunft Feldprobleme effizient lösen zu können.

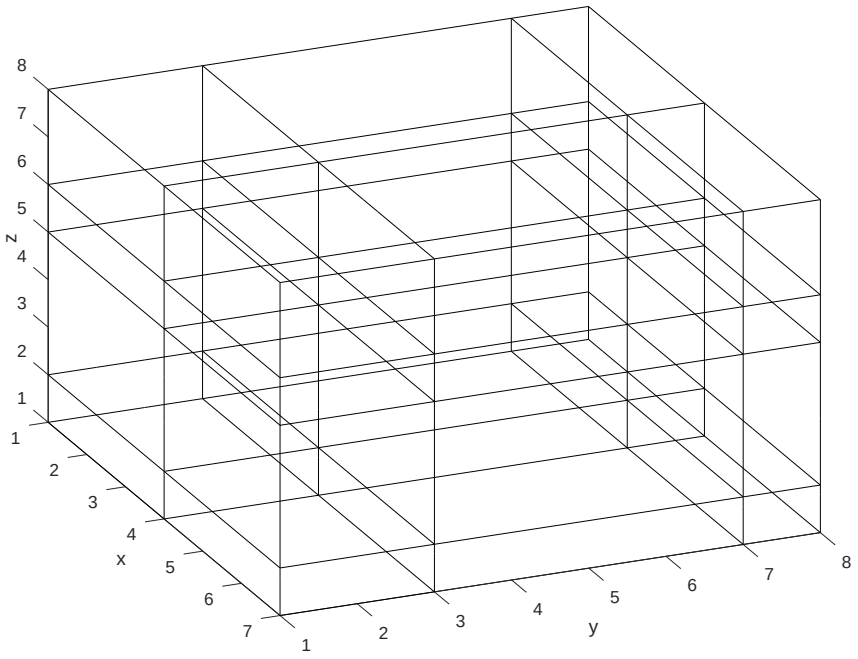


**Abbildung 1.6:** exampleMesh Plot

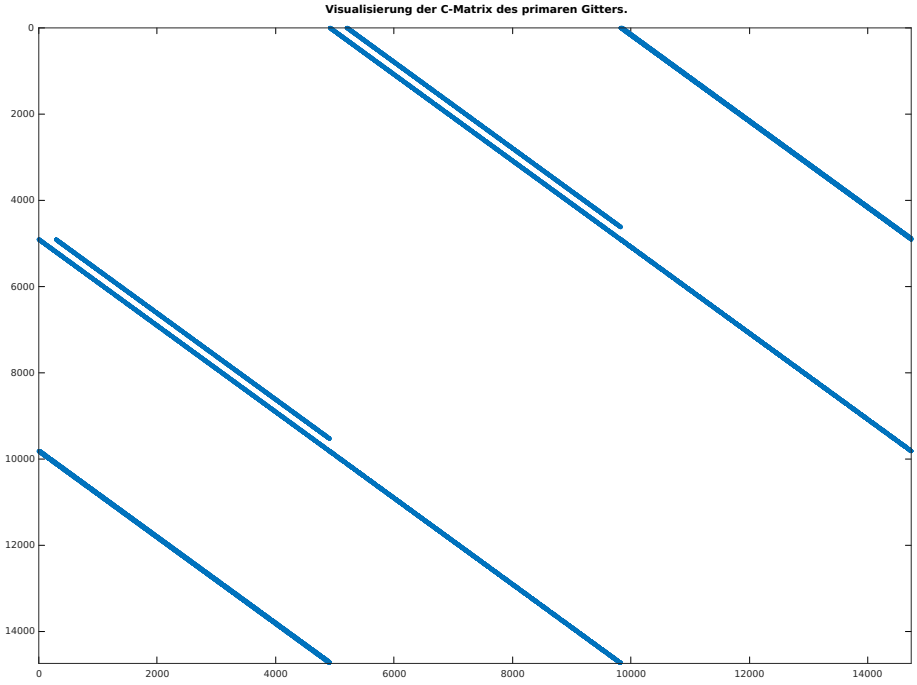


**Abbildung 1.7:** Speicherverbrauch einer C-Matrix in unterschiedlichen Speicherformaten

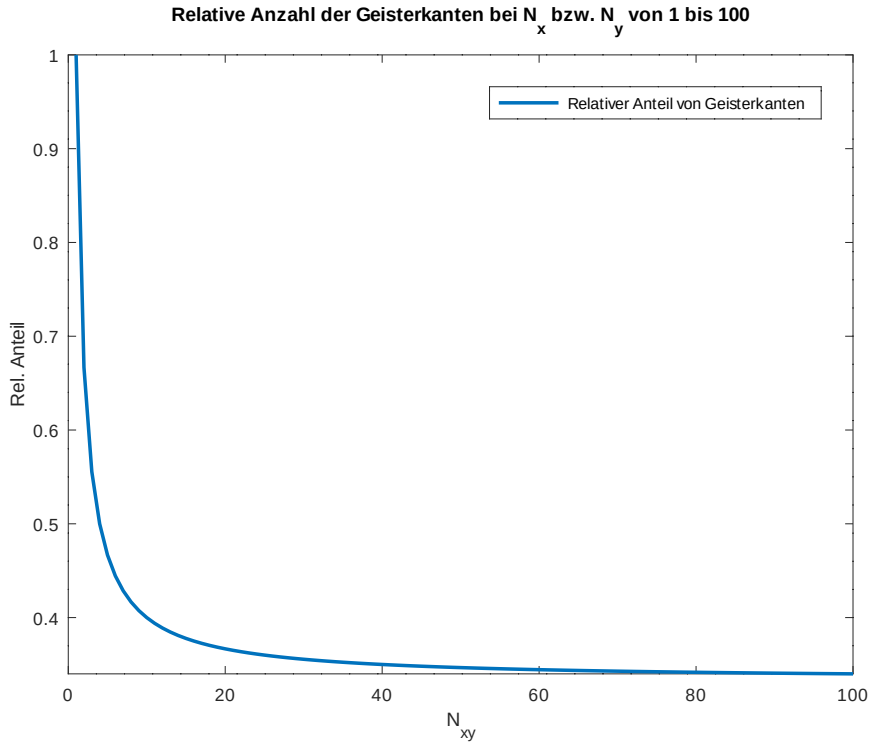


**Abbildung 1.8:** Anteil der Geisterkanten relativ zur Gesamtanzahl der Kanten.

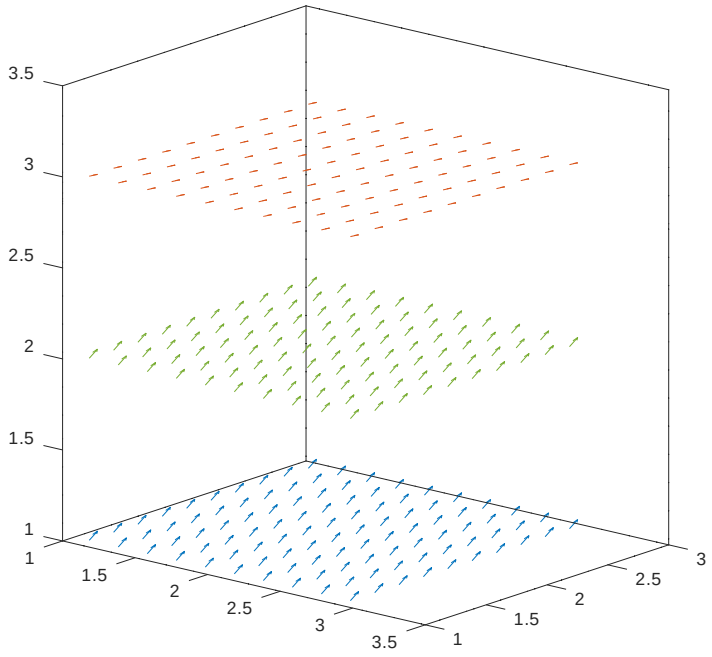


**Abbildung 1.9:** Feldverteilung aus 10.1

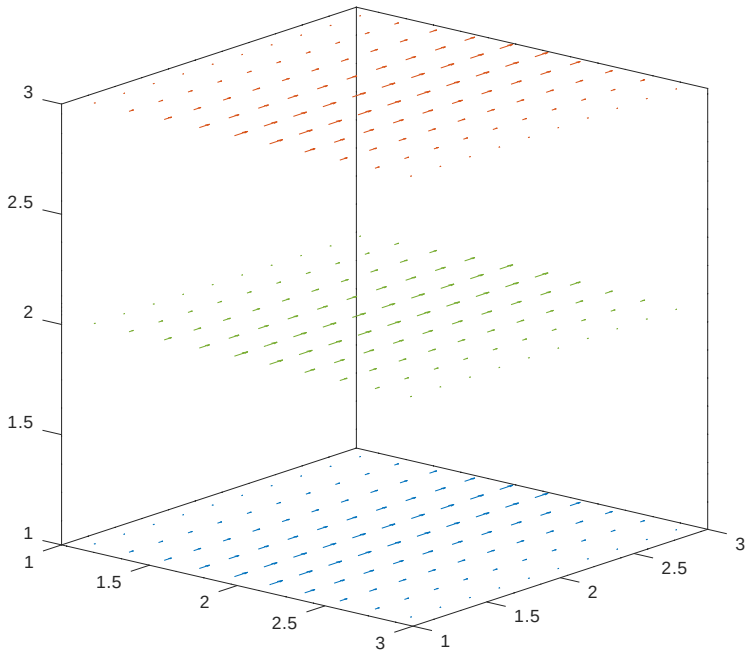


**Abbildung 1.10:** Feldverteilung aus 10.2

## Abbildungsverzeichnis