

---

## 2 Grundlagen der Methode der Finiten Integration 1

---

### 2.1 Inhalt und Ziel des Versuchs

---

Ziel des Versuchs ist eine Einführung in die Grundlagen der *FIT* (*Finite Integration Technique*). Insbesondere sollen die Definitionen und die Handhabung der Rechengitter und der verwendeten Datenstrukturen vermittelt werden. Am Beispiel von zweidimensionalen Dreiecksgittern und dreidimensionalen kartesischen Gittern werden die Operatormatrizen der FIT und einfache Gittervektoren erstellt und visualisiert.

---

### 2.2 Theorie

---

---

#### 2.2.1 Rechengitter

---

Grundlage der räumlichen Diskretisierung in der FIT wie auch in einigen anderen Methoden (*Finite Differenzen*, *Finite Elemente*) ist eine Zerlegung des endlichen Problemvolumens in eine Menge von Teilvolumina  $V_l$ . Im Dreidimensionalen werden diese Gitterzellen  $V_l$  (engl. *cells*) durch eine Menge von Gitterflächen (engl. *facets*) begrenzt, jede Gitterfläche  $A_k$  wiederum durch eine Menge von Gitterkanten (engl. *edges*) und jede Gitterkante  $L_j$  durch zwei Gitterpunkte (engl. *points* oder *vertices*). Durch diese geometrischen Objekte, die *Elementarfiguren*  $G$ , wird schließlich das *Rechengitter* (engl. *grid* oder *mesh*) definiert.  $N_p$ ,  $N_L$ ,  $N_A$  und  $N_V$  entsprechen jeweils der Gesamtanzahl der diskretisierten Elementarfiguren:

$G$ :	Gitterpunkte	$P_i$	$(i = 1 \dots N_p)$
	Gitterkanten	$L_j$	$(j = 1 \dots N_L)$
	Gitterflächen	$A_k$	$(k = 1 \dots N_A)$
	Gitterzellen	$V_l$	$(l = 1 \dots N_V)$

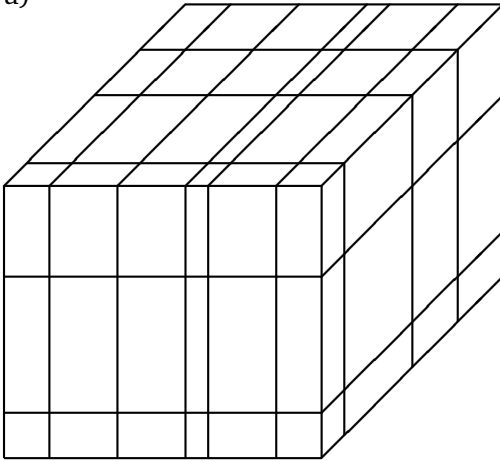
An die geometrischen Objekte des Gitters müssen zunächst nur einige wenige Anforderungen gestellt werden: Die Schnittmenge von zwei verschiedenen Zellen kann z.B. leer sein oder aus genau einer Gitterfläche bestehen. Ähnliches gilt für das Verhältnis von Flächen und Kanten und von Kanten und Punkten.

Es ist eine Reihe von verschiedenen Zelltypen zugelassen: In 3D sind dies z. B. Würfel, Quader, Tetraeder oder allgemeine  $n$ -flächige Körper; in 2D entsprechend Dreiecke, Rechtecke usw. Auch eine Mischung verschiedener Typen in einem Gitter ist möglich und bisweilen sinnvoll.

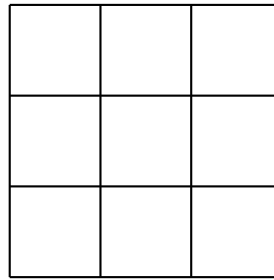
In der Praxis werden aufgrund ihrer kompakten und einfachen Datenstruktur vor allem strukturierte Gitter eingesetzt. Im dreidimensionalen Raum bestehen die Gitter insbesondere aus Tetraedern und allgemeinen *6-Flächnern* (Hexaeder). Bei letzteren spielen so genannte kartesische Gitter eine bedeutende Rolle. Hier sind die Gitterkanten parallel zu den kartesischen Koordinatenachsen ausgerichtet.

Von *strukturierten Gittern* spricht man, wenn alle Gitterzellen vom selben Typ sind und alle Elementarfiguren jeweils gleich viele Nachbarn besitzen (abgesehen von den Elementarfiguren am Rand des Rechengebiets).

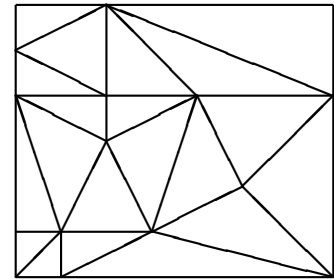
a)



b)



c)



**Abbildung 2.1:** Beispiele für 2D und 3D Gitter für den Fall eines a) kartesischen, nicht äquidistanten, dreidimensionalen Gitters; b) kartesischen, äquidistanten, zweidimensionalen Gitters; c) zweidimensionalen Dreiecksgitters.

## 2.2.2 Funktion des Rechengitters

Dem Rechengitter kommen in der numerischen Feldberechnung zwei wesentliche Funktionen zu:

1. Über den geometrischen Ort der Elementarfiguren wird festgelegt, an welchen Stellen des Raumes Feldgrößen berechnet werden. Dies hängt im Einzelnen von der verwendeten Methode ab. In der FIT werden z.B. Feldstärken bzw. Spannungen an Gitterkanten *alloziert* (berechnet), Flussdichten bzw. Flüsse auf Gitterflächen. In der Standardformulierung der Methode der *Finiten Elemente* werden sog. Ansatzfunktionen den Gitterpunkten oder -kanten zugeordnet.
2. In den meisten Methoden wird außerdem die Materialverteilung einschließlich der Randbedingungen durch eine Übertragung auf das Rechengitter modelliert, indem jeder Gitterzelle eine homogene Materialfüllung mit konstanten Materialparametern zugeordnet wird. Für eine realistische Modellbildung muss dazu die Lage der Gitterobjekte möglichst gut an Rand- und Grenzflächen der Anordnung angepasst werden (*konforme Gitter*).

## 2.2.3 Datenstrukturen für Rechengitter

Um eine numerische Feldsimulation mit Hilfe eines Computers durchführen zu können, muss das Rechengitter selbst durch eine geeignete Datenstruktur beschrieben werden.

### Kanonisches Indizierungsschema für kartesische Gitter

Für kartesische Gitter mit  $N_p = N_x \cdot N_y \cdot N_z$  Gitterpunkten müssen dazu lediglich die Koordinaten der Gitterlinien in jede Koordinatenrichtung abgespeichert werden, z.B. in den Feldern

$$\text{xmesh} = [x_1, x_2, \dots, x_{N_x}] \quad (2.1)$$

$$\text{ymesh} = [y_1, y_2, \dots, y_{N_y}] \quad (2.2)$$

$$\text{zmesh} = [z_1, z_2, \dots, z_{N_z}], \quad (2.3)$$

wodurch die Position jedes Gitterpunktes  $P_{ijk}$  eindeutig definiert ist durch

$$P_{ijk} = \{x_i \mid y_j \mid z_k\}. \quad (2.4)$$

Der dreidimensionale Index  $(ijk)$  wird nach dem so genannten **kanonischen Indizierungsschema** mit der Formel

$$n = 1 + (i - 1)M_x + (j - 1)M_y + (k - 1)M_z$$

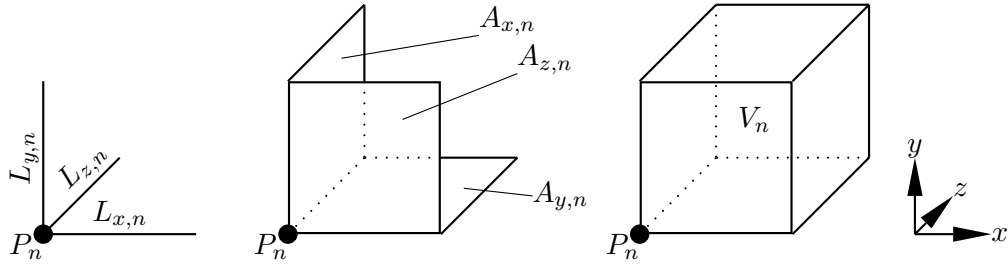
mit  $M_x = 1, M_y = N_x, M_z = N_x N_y$

(2.5)

auf einen eindimensionalen Index  $n$  transformiert. Dies entspricht anschaulich einer Nummerierung der Gitterpunkte zuerst in  $x$ -, dann in  $y$ - und zuletzt in  $z$ -Richtung (siehe auch Abb. 2.3).

Die Werte  $M_x$ ,  $M_y$  und  $M_z$  geben an, welcher Wert zum aktuellen Index  $n$  eines Punktes dazugaddiert werden muss, um den nächsten Punkt in  $x$ -,  $y$ - bzw.  $z$ -Richtung zu erhalten.

Jedem Punkt werden nun 3 Kanten und 3 Flächen zugeordnet, welche jeweils in die positive  $x$ -,  $y$ - bzw.  $z$ -Richtung zeigen (siehe Abb. 2.2).



**Abbildung 2.2:** Indizierung von Kanten und Flächen. Jeder Kante und Fläche wird in  $x$ -,  $y$ - und  $z$ -Richtung derselbe Index, den auch der gemeinsame Startpunkt hat, zugeordnet.

Damit lassen sich alle Punkte, die zu einer Kante, Fläche bzw. Volumen gehören, sehr einfach bestimmen:

$$L_{x,n} := (P_n, P_{n+M_x}), \quad A_{y,n} := (P_n, P_{n+M_x}, P_{n+M_z}, P_{n+M_z+M_x}), \quad (2.6)$$

$$V_n := (P_n, P_{n+M_x}, P_{n+M_y}, P_{n+M_z}, P_{n+M_x+M_y}, P_{n+M_x+M_z}, P_{n+M_y+M_z}, P_{n+M_x+M_y+M_z}). \quad (2.7)$$

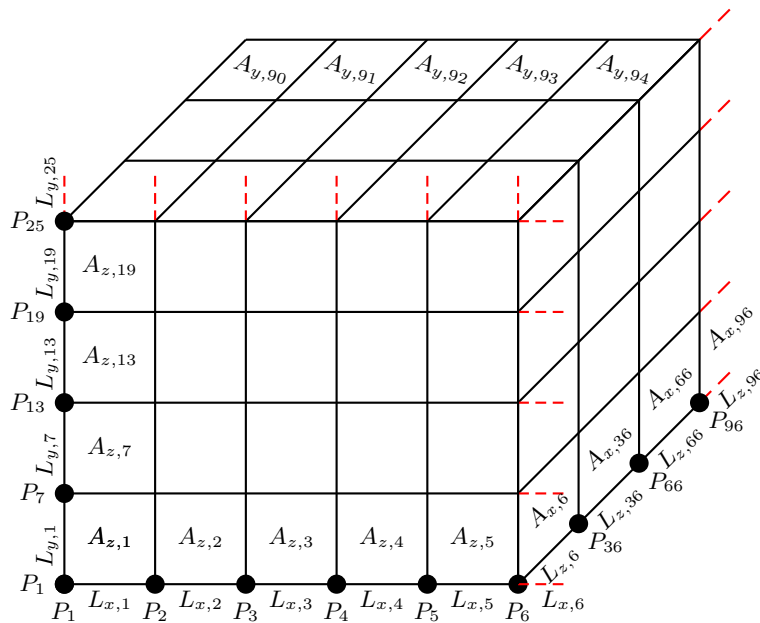
Um auch für Kanten und Flächen einen eindimensionalen Index zu erhalten, werden die einzelnen Komponenten der Kanten und Flächen in einem Vektor hintereinander angeordnet, sodass zum Beispiel für Kanten gilt

$$L_{x,n} \rightarrow L_n, \quad L_{y,n} \rightarrow L_{N_p+n}, \quad L_{z,n} \rightarrow L_{2 \cdot N_p+n}. \quad (2.8)$$

Bei diesem Indizierungsschema entstehen an den Rändern des Gitters Indizes, deren Kanten, Flächen und Zellen nicht mehr im Rechengebiet selbst liegen (siehe rot gestrichelte Kanten in Abb. 2.3). Diese irregulären Indizes werden zugunsten des beschriebenen einfachen Schemas bewusst in Kauf genommen. Häufig werden die zugehörigen überzähligen Kanten, Flächen und Volumina auch als Geisterkanten, -flächen bzw. -volumina bezeichnet. Bei der praktischen Implementierung werden diese dennoch in den Vektoren angelegt, jedoch durch Nullsetzen in einer der Matrizen (z.B. Materialmatrizen) ausgeblendet, oder alternativ am Ende komplett aus dem Gleichungssystem entfernt (Zeilen und Spalten in der Matrix streichen).

Die Anzahl der Kanten sei im folgenden  $N_L$ , die Anzahl der Flächen  $N_A$  und die Anzahl der Volumina  $N_V$ . Es gilt hierfür näherungsweise

$$N_L, N_A \approx 3 \cdot N_p, \quad N_V \approx N_p \quad (2.9)$$



**Abbildung 2.3:** Komplettes Rechengebiet mit  $5 \times 4 \times 3$  Zellen und Darstellung der Indizes von ausgewählten Punkten, Kanten und Flächen. Angedeutet sind auch einige Indizes, die irreguläre Kanten (rot gestrichelte Linien) besitzen.

## Allgemeine Gitter

Nicht-kartesische Gitter erfordern eine wesentlich aufwendigere Datenstruktur, da nicht nur die Koordinaten jedes einzelnen Gitterpunkts gespeichert werden müssen, sondern auch die Topologie des Gitters nicht mehr eindeutig aus den Punktlisten hervorgeht. Daher müssen für solche Gitter außerdem die Relationen (auch Inzidenzen genannt) zwischen den verschiedenen geometrischen Objekten abgespeichert werden. Die allgemeine Effizienz einer Datenstruktur ist bestimmt durch

- den verwendeten Speicherplatz
- die Rechenzeit zum Zugriff auf Gitterdaten
  - bei der Erstellung der Gitter
  - bei weiteren Berechnungen wie z.B. der Aufstellung der Matrizen, der eigentlichen Feldsimulation, der Visualisierung der Ergebnisse, usw.
- die Robustheit der Datenstruktur in Bezug auf mögliche Fehlerquellen, wie z.B. Rundungsfehler durch endliche Rechengenauigkeit oder inkonsistente Daten nach einer Modifikation des Gitters.

Auf heutigen Rechenanlagen spielt der Speicherplatz eine immer kleinere Rolle und die Gitterdaten können in aller Regel komplett im Speicher gehalten werden. Speicherkritisch sind vielmehr häufig die Systemmatrizen und die zu berechnenden Felddaten.

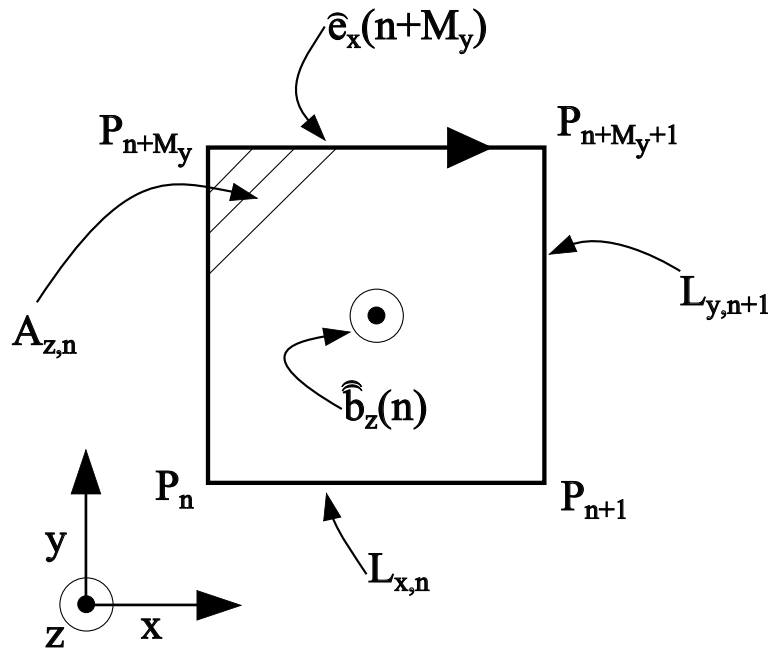
Bei der Abschätzung der Rechenzeiten spielt vor allem die Komplexität der verwendeten Algorithmen eine große Rolle: Der Aufwand soll bei einer Vergrößerung des Gitters möglichst linear mit der Anzahl der Elementarfiguren steigen. Dies ist vor allem bei Suchalgorithmen im Allgemeinen nicht erreichbar. Häufig ist daher eine Redundanz in der Datenstruktur, die mehrfaches Suchen vermeidet, von Vorteil.

Die in diesem Versuch zu verwendende Datenstruktur für zweidimensionale Dreiecksgitter soll deswegen folgende Eigenschaften aufweisen:

- Die Punktkoordinaten ( $x$ ,  $y$  und  $z$ ) werden in einer Punkteliste als Daten vom Zahlentyp *real* (in MATLAB<sup>®</sup>: `double`) in einer  $N_p \times 3$  Liste gespeichert.
- Um Rundungsfehler bei den Koordinaten oder Fehler beim Vergleichen von Punkten zu vermeiden, werden alle anderen Daten zur Beschreibung der Gittertopologie (und damit der Inzidenzen) als Integerzeiger (bzw. in MATLAB<sup>®</sup> ebenfalls `double`, jedoch mit Integerzahlen belegt) auf diese Punkteliste angelegt:
  - Kanten als ein  $N_L \times 2$ -Feld von Zeigern
  - Flächen als ein  $N_A \times 3$ -Feld von Zeigern.

#### 2.2.4 Die elektromagnetischen Größen im Rechengitter

Wie in Abschnitt 2.2.2 beschrieben, gibt das Rechengitter die Orte im Raum an, an denen die Felddaten ( $\vec{E}$ ,  $\vec{H}$ ,  $\vec{D}$ ,  $\vec{B}$ ) ausgewertet werden. In der FIT werden zum Beispiel Feldstärken als integrale Zustandsgrößen<sup>1</sup> entlang einer Gitterkante ausgewertet (siehe Abb. 2.4).



**Abbildung 2.4:** Allokation der integralen Zustandsgrößen  $\hat{e}$  und  $\hat{b}$  im Rechengitter. Die integrale Zustandsgröße  $\hat{e}_x(n+M_y)$  ist definiert als die elektrische Spannung zwischen dem Gitterpunkt  $P_{n+M_y}$  und dem Gitterpunkt  $P_{n+M_y+1}$ . Sie wird auf der Gitterkante  $L_{x,n+M_y}$  allokiert. Analog ist die integrale Zustandsgröße  $\hat{b}_z(n)$  gleich dem magnetischen Fluss durch die Gitterfläche  $A_{z,n}$ . Die Bezeichnungen der Knoten, Kanten und Flächen des Gitters erfolgt gemäß Abschnitt 2.2.3.

<sup>1</sup> Die integralen Zustandsgrößen sind im Skript, Abschnitt 4.3.1, ausführlich erläutert.

Zwischen der integralen Zustandsgröße  $\hat{e}_x(n + M_y)$  und der kontinuierlichen Feldgröße  $\vec{E}$  besteht der Zusammenhang

$$\hat{e}_x(n + M_y) = \int_{L_{x,n+M_y}} \vec{E}(\vec{r}) \cdot d\vec{s} = \int_{x_i}^{x_{i+1}} E_x(x, y_{j+1}, z_k) dx, \quad (2.10)$$

wobei die Koordinaten eines Gitterpunktes  $P_n$  gemäß Gl. (2.4) mit  $(x_i, y_j, z_k)$  bezeichnet werden. Analog zu Gl. (2.10) werden die übrigen integralen Zustandsgrößen für alle Gitterkanten berechnet. Diese werden dann wie in Gl. (2.8) beschrieben in einem Vektor, für die elektrische Feldstärke  $\hat{e}$ , zusammengefasst. Aufgrund des in Gl. (2.10) beschriebenen Zusammenhangs werden die Komponenten des Vektors  $\hat{e}$  auch als *elektrische Gitterspannungen* bezeichnet.

Ebenso ergibt sich die integrale Zustandsgröße  $\hat{b}_z(n)$  durch Integration

$$\hat{b}_z(n) = \int_{A_{z,n}} \vec{B}(\vec{r}) \cdot d\vec{A} = \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} B_z(x, y, z_k) dy dx. \quad (2.11)$$

Die Größe  $\hat{b}_z(n)$  ist also eine Flussgröße und wird daher als *magnetischer Gitterfluss* bezeichnet. Die Zusammenfassung der Flüsse in einem Vektor erfolgt im Vektor  $\hat{b}$ . Auch dieser Vektor ist entsprechend Gl.(2.8) aufgebaut.

An dieser Stelle ist festzuhalten:

1. Die **integrale Zustandsgröße**  $\hat{e}$  wird auf den **Gitterkanten** allokiert.
2. Die **integrale Zustandsgröße**  $\hat{b}$  wird auf den **Gitterflächen** allokiert.

## 2.2.5 Duales Gitter

Wie im Skript ausführlich erklärt wird, definiert man zur Allokation der restlichen Zustandsgrößen der FIT (der *magnetischen Gitterspannung*  $\hat{h}$ , des *elektrischen Gitterflusses*  $\hat{d}$  und des *elektrischen Gitterstroms*  $\hat{j}$ ) ein zweites, sogenanntes *duales Gitter*  $\tilde{G}$ . Es ist definiert durch die dualen Gitterelemente

$\tilde{G}$ :	duale Gitterpunkte	$\tilde{P}_i$	$(i = 1 \dots \tilde{N}_p),$
	duale Gitterkanten	$\tilde{L}_j$	$(j = 1 \dots \tilde{N}_l),$
	duale Gitterflächen	$\tilde{A}_k$	$(k = 1 \dots \tilde{N}_A),$
	duale Gitterzellen	$\tilde{V}_l$	$(l = 1 \dots \tilde{N}_v)$

und seine Beziehungen zum zuvor eingeführten *primären Gitter*  $G$ , den *Dualitätsbedingungen* (vgl. auch Abb. 2.5):

1. Zu jeder primären Gitterfläche  $A_k$  gibt es genau eine duale Gitterkante  $\tilde{L}_j$ , die die Gitterfläche in genau einem Punkt schneidet, und umgekehrt.
2. Jeder Gitterpunkt des primären Gitters  $P_i$  liegt innerhalb genau einer Zelle des dualen Gitters  $\tilde{V}_l$  und umgekehrt.

Die magnetische Gitterspannung

$$\hat{h}_x(n) = \int_{\tilde{L}_n} \vec{H}(\vec{r}) \cdot d\vec{s} = \int_{\tilde{x}_{i-1}}^{\tilde{x}_i} H_x(x, \tilde{y}_j, \tilde{z}_k) dx. \quad (2.12)$$



wendigerweise aus den gleichen geometrischen Figuren wie das primäre Gitter. Ein einfaches Beispiel für diese Tatsache sind zweidimensionale Dreiecksgitter, wie sie auch in diesem Versuch behandelt werden.

---

## Orthogonalität

---

Einige erhebliche Vereinfachungen in Formulierung und Implementierung der FIT ergeben sich, wenn sich Gitterkanten des primären und Gitterflächen des dualen Gitters (und umgekehrt) unter einem Winkel von  $90^\circ$  schneiden.<sup>2</sup>

Diese wichtige Eigenschaft wird als *Orthogonalität* bezeichnet. Sie kann nur für einige wenige Sonderfälle von Rechengittern erfüllt werden, darunter allerdings für die wichtige Klasse der kartesischen Gitter und anderer sog. Koordinatengitter. Häufig erfordert die Erfüllung der Orthogonalitätsbedingung auch speziellen Aufwand bei der Gittererzeugung. Für die Dualität zweier Gitter ist ihre Orthogonalität dagegen im Allgemeinen keine Voraussetzung.

---

### 2.2.6 Behandlung der Materialverteilung im Rechengebiet

---

Da die Behandlung der Materialverteilung<sup>3</sup> im Rechengebiet im nächsten Versuch ausführlich betrachtet wird, wird an dieser Stelle auf eine ausführliche Betrachtung verzichtet. Im Kontinuierlichen lauten die Materialbeziehungen

$$\vec{D}(\vec{r}) = \varepsilon(\vec{r}) \vec{E}(\vec{r}), \quad \vec{J}(\vec{r}) = \kappa(\vec{r}) \vec{E}(\vec{r}) \quad \text{und} \quad \vec{B}(\vec{r}) = \mu(\vec{r}) \vec{H}(\vec{r}) \quad (2.14)$$

In der diskreten Betrachtungsweise der FIT werden diese Materialbeziehungen<sup>4</sup> zu

$$\hat{\vec{d}} = \mathbf{M}_\varepsilon \hat{\vec{e}}, \quad \hat{\vec{j}} = \mathbf{M}_\kappa \hat{\vec{e}} \quad \text{und} \quad \mathbf{M}_{\mu^{-1}} \hat{\vec{b}} = \hat{\vec{h}}. \quad (2.15)$$

In den Materialmatrizen  $\mathbf{M}_\varepsilon$  und  $\mathbf{M}_{\mu^{-1}}$  wird u.a. eine unterschiedliche Materialverteilung in den einzelnen Gitterzellen, wie in Abb. 2.5 angedeutet, berücksichtigt.

Sind primäres und duales Gitter zueinander orthogonal, so können Materialmatrizen in Diagonalform gefunden werden, mit erheblichen Vorteilen für Theorie und Implementierung. Unter anderem reicht es dann aus, nur die Hauptdiagonale der Matrix im Speicher zu halten.

---

### 2.2.7 Die Matrizen $\mathbf{C}$ , $\tilde{\mathbf{C}}$ , $\mathbf{S}$ und $\tilde{\mathbf{S}}$

---

Bisher wurden die diskreten Entsprechungen ( $\hat{\vec{e}}$ ,  $\hat{\vec{b}}$ ,  $\mathbf{M}_\varepsilon$  usw.) der kontinuierlichen Größen der Maxwell'schen Gleichungen ( $\vec{E}$ ,  $\vec{B}$ ,  $\varepsilon$  usw.) eingeführt. Damit lassen sich wie in Gl. (2.15) die Materialbeziehungen in einer diskreten Formulierung angeben. Um nun auch die Maxwell'schen Gleichungen in einer diskreten Entsprechung formulieren zu können, müssen die durch die Gleichungen beschriebenen Beziehungen zwischen den Größen auf dem Gitter betrachtet werden. Dieses ist im Skript ausführlich für alle vier Maxwell'schen Gleichungen in den Abschnitten 4.3 - 4.7 dargestellt. Der Vollständigkeit halber wird diese Betrachtung hier anhand des Induktionsgesetzes vorgestellt:

Ausgangspunkt bildet dabei das Induktionsgesetz in seiner allgemeinen Form

$$\oint_{\partial A} \vec{E}(\vec{r}, t) \cdot d\vec{s} = -\frac{d}{dt} \iint_A \vec{B}(\vec{r}, t) \cdot d\vec{A}. \quad (2.16)$$

---

<sup>2</sup> siehe Skript 4.4.2

<sup>3</sup> siehe Skript 4.4

<sup>4</sup> Warum in der Matrix  $\mathbf{M}_{\mu^{-1}}$  die *inverse* Permeabilität verwendet wird, wird im Skript 4.4.2 und im nächsten Versuch erläutert.



Angewendet auf eine Gitterzelle (wie in Abb. 2.4 gezeigt) führt Gl. (2.16) auf

$$\oint_{\partial A_{z,n}} \vec{E} \cdot d\vec{s} = \int_{L_{x,n}} \vec{E} \cdot d\vec{s} + \int_{L_{y,n+1}} \vec{E} \cdot d\vec{s} + \int_{-L_{x,n+M_y}} \vec{E} \cdot d\vec{s} + \int_{-L_{y,n}} \vec{E} \cdot d\vec{s} = -\frac{d}{dt} \iint_{A_{z,n}} \vec{B} \cdot d\vec{A}. \quad (2.17)$$

Dabei ist der Umlaufsinn – mathematisch positiv – und das Vorzeichen des magnetischen Gitterflusses zu beachten. Weiter führt Gl. (2.16) auf

$$\underbrace{\int_{L_{x,n}} \vec{E} \cdot d\vec{s}}_{\widehat{e}_x(n)} + \underbrace{\int_{L_{y,n+1}} \vec{E} \cdot d\vec{s}}_{\widehat{e}_y(n+1)} - \underbrace{\int_{L_{x,n+M_y}} \vec{E} \cdot d\vec{s}}_{\widehat{e}_x(n+M_y)} - \underbrace{\int_{L_{y,n}} \vec{E} \cdot d\vec{s}}_{\widehat{e}_y(n)} = -\frac{d}{dt} \underbrace{\iint_{A_{z,n}} \vec{B} \cdot d\vec{A}}_{\widehat{\widehat{b}}_z(n)}. \quad (2.18)$$

Die Integrale in Gl. (2.18) sind wie dort gezeigt bereits aus Abschnitt 2.2.4 bekannt, so dass sich das Induktionsgesetz für eine Gitterzelle schließlich als

$$\widehat{e}_x(n) + \widehat{e}_y(n+1) - \widehat{e}_x(n+M_y) - \widehat{e}_y(n) = -\frac{d}{dt} \widehat{\widehat{b}}_z(n) \quad (2.19)$$

schreiben lässt.

Wie man sieht, hängt die zeitliche Ableitung des Gitterflusses durch eine Gitterfläche nur von den Gitterspannungen der die Fläche umrandenden Gitterkanten ab. Dies gilt für alle auftretenden Gitterflüsse. Es müssen also stets die passenden Gitterspannungen **vorzeichenrichtig** summiert werden, um die zeitlichen Ableitungen der Gitterflüsse zu bestimmen. Dementsprechend ergeben sich die zeitlichen Ableitungen aller Gitterflüsse im Rechengebiet als eine Matrix-Vektor-Multiplikation gegeben durch

$$\mathbf{C} \widehat{\mathbf{e}} = -\frac{d}{dt} \widehat{\widehat{\mathbf{b}}}. \quad (2.20)$$

Dabei enthält die Matrix  $\mathbf{C}$  die Vorzeichen der Summation – also nur +1 und –1. Aufgrund der Anordnung der integralen Zustandsgrößen im Vektor  $\widehat{\mathbf{e}}$  gemäß Gl. (2.8) hat die Matrix  $\mathbf{C}$  die nachfolgend gezeigte Struktur<sup>5</sup>:

$$\mathbf{C} = \begin{pmatrix} 0 & \begin{array}{c} \diagup +1 \diagdown -1 \end{array} & \begin{array}{c} \diagdown -1 \diagup +1 \end{array} \\ \begin{array}{c} \diagdown -1 \diagup +1 \end{array} & 0 & \begin{array}{c} \diagdown -1 \diagup +1 \end{array} \\ \begin{array}{c} \diagdown -1 \diagup +1 \end{array} & \begin{array}{c} \diagdown -1 \diagup +1 \end{array} & 0 \end{pmatrix} = \begin{pmatrix} 0 & -\mathbf{p}_z & \mathbf{p}_y \\ \mathbf{p}_z & 0 & -\mathbf{p}_x \\ -\mathbf{p}_y & \mathbf{p}_x & 0 \end{pmatrix}. \quad (2.21)$$

<sup>5</sup> Die Struktur der Matrix hängt im Allgemeinen von der Topologie des verwendeten Gitters ab. Für die hier betrachteten kartesischen Gitter gilt die gezeigte Struktur in jedem Fall.

Auch die Untermatrizen von  $\mathbf{C}$ , die  $N_p \times N_p$ -Matrizen  $\mathbf{P}_\xi$  ( $\xi = x, y, z$ ), haben eine einheitliche, aber jeweils vom Gitter abhängige Struktur:

$$\mathbf{P}_\xi = \begin{pmatrix} \overbrace{\quad\quad\quad}^{M_\xi} & & & \\ & 0 & & \\ & & +1 & \\ & & & 0 \\ & & -1 & \\ & 0 & & & \end{pmatrix},$$

Für die Komponenten der Matrix  $\mathbf{P}_\xi$  gilt dann:

$$P_{u,v} = \begin{cases} -1 & u = v \\ +1 & v = u + M_\xi \\ 0 & \text{sonst} \end{cases}$$

Es sei ausdrücklich darauf hingewiesen, dass die Matrizen in den oben angegebenen Formen auch auf Komponenten zugreifen, die es nicht gibt, z.B.  $\hat{e}_x(N_x)$ . Bei der Implementierung müssen diese Zugriffe durch das Einfügen von Nulleinträgen in den Matrizen vermieden werden. Ebenso wurden die Randbedingungen noch nicht berücksichtigt.

Wie gezeigt wurde, kann die Matrix  $\mathbf{C}$  als das diskrete Analogon zum Integral  $\oint_{\partial A} d\vec{s}$  bzw. in differentieller Form zum rot-Operator verstanden werden. Die Matrix  $\mathbf{C}$  wird jeweils angewandt auf die Größen, die auf dem primären Gitter allokiert sind. Ebenso lassen sich Matrizen für die übrigen Operationen  $\oint_{\partial A} d\vec{s}$  bzw. rot auf dem dualen Gitter,  $\oint_{\partial V} d\vec{A}$  bzw. div auf dem primärem und dualen Gitter sowie grad auf dem primärem Gitter finden. Alle diese Matrizen werden mit Hilfe der Matrix  $\mathbf{P}_\xi$  dargestellt.

**Matrizen, die auf dem dualen Gitter operieren, erhalten eine Tilde, z.B.  $\tilde{\mathbf{C}}$ .** So ergeben sich folgende Analogien:

$$\oint_{\partial A} d\vec{s}, \text{ rot} \longrightarrow \mathbf{C}, \tilde{\mathbf{C}} = \mathbf{C}^\top \quad (2.22)$$

$$\oint_{\partial V} d\vec{A}, \text{ div} \longrightarrow \mathbf{S} = (\mathbf{P}_x \mathbf{P}_y \mathbf{P}_z), \tilde{\mathbf{S}} = (\tilde{\mathbf{P}}_x \tilde{\mathbf{P}}_y \tilde{\mathbf{P}}_z) = (-\mathbf{P}_x^\top - \mathbf{P}_y^\top - \mathbf{P}_z^\top) \quad (2.23)$$

$$\text{grad} \longrightarrow \mathbf{G} = \begin{pmatrix} \mathbf{P}_x \\ \mathbf{P}_y \\ \mathbf{P}_z \end{pmatrix} = -\tilde{\mathbf{S}}^\top. \quad (2.24)$$

Die wichtigen *Dualitätseigenschaften*  $\tilde{\mathbf{C}} = \mathbf{C}^\top$  und  $\mathbf{G} = -\tilde{\mathbf{S}}^\top$  lassen sich im hier dargestellten, kartesischen Fall aus der Grundeigenschaft  $\tilde{\mathbf{P}}_\xi = -\mathbf{P}_\xi^\top$  ableiten, gelten aber auch für alle anderen Typen dualer Gittersysteme (Beweisidee dazu im Skript zur Vorlesung).

Mit den Gl. (2.22) - (2.24) lassen sich schließlich die Maxwell'schen Gleichungen in diskreter Form, die sog. **Gitter-Maxwell-Gleichungen (GMG)**, angeben:

$$\begin{aligned} \text{rot } \vec{E} = -\frac{d}{dt} \vec{B} &\longrightarrow \mathbf{C} \hat{\mathbf{e}} = -\frac{d}{dt} \hat{\mathbf{b}} \\ \text{rot } \vec{H} = \frac{d}{dt} \vec{D} + \vec{J} &\longrightarrow \tilde{\mathbf{C}} \hat{\mathbf{h}} = \frac{d}{dt} \hat{\hat{\mathbf{d}}} + \hat{\mathbf{j}} \\ \text{div } \vec{B} = 0 &\longrightarrow \mathbf{S} \hat{\hat{\mathbf{b}}} = \mathbf{0} \\ \text{div } \vec{D} = \varrho &\longrightarrow \tilde{\mathbf{S}} \hat{\hat{\mathbf{d}}} = \mathbf{q}. \end{aligned}$$

## 2.2.8 Besonderheiten in 2D

In vielen Anwendungsfällen reicht es aus, nur eine zweidimensionale Feldverteilung zu berechnen. Dazu wird ein zweidimensionales Rechengitter benötigt. Diese Situation ist i. Allg. immer dann gegeben, wenn die Abhängigkeit der Felder in eine Raumrichtung im Voraus bekannt ist: Im einfachsten Fall bei längshomogenen Feldern mit  $\frac{\partial}{\partial z} \equiv 0$ , beispielsweise aber auch im Falle einer Wellenausbreitung in Hohlleitern mit  $f(z) \sim \exp(-jk_z z)$  oder bei Moden in rotationssymmetrischen Strukturen mit  $f(\varphi) \sim \cos / \sin(m\varphi)$  ( $m = 0, 1, \dots$ ).

Auf den ersten Blick erscheinen zweidimensionale Gitter lediglich als eine Vereinfachung des allgemeinen 3D-Falls. Dies ist zumindest bei der Erzeugung und Verwaltung der Gitter selbst auch der Fall. Bei der Anwendung der FIT gibt es allerdings einige Besonderheiten zu beachten:

- 2D-Gitter bestehen nur aus einer Gitterebene in normaler Richtung (meist der  $z$ -Richtung). Formal kann in vielen Formeln schlicht  $N_z = 1$  gesetzt werden.
- Im Fall *längshomogener Felder* ( $\frac{\partial}{\partial z} \equiv 0$ ) wird auch die diskrete Differentiations-Matrix für diese Raumrichtung zu Null:

$$\mathbf{P}_z = 0 \quad \Rightarrow \quad \mathbf{C}_{2D} = \begin{pmatrix} 0 & 0 & \mathbf{P}_y \\ 0 & 0 & -\mathbf{P}_x \\ -\mathbf{P}_y & \mathbf{P}_x & 0 \end{pmatrix}. \quad (2.25)$$

Man erkennt daraus, dass – wie im Kontinuierlichen – auch bei den diskreten Feldlösungen der FIT zwei vollständig entkoppelte Lösungsräume entstehen: TE-Felder mit den Komponenten  $\hat{e}_x, \hat{e}_y, \hat{h}_z$  und TM-Felder mit den Komponenten  $\hat{h}_x, \hat{h}_y, \hat{e}_z$ .

Wird z.B. nur der TE-Fall betrachtet, so verkürzt man zweckmäßigerweise die Vektoren und Matrizen auf

$$\hat{\mathbf{e}}_{TE} = \begin{pmatrix} \hat{e}_x \\ \hat{e}_y \end{pmatrix}, \quad \hat{\mathbf{h}}_{TE} = (\hat{h}_z), \quad \mathbf{C}_{TE} = \begin{pmatrix} -\mathbf{P}_y & \mathbf{P}_x \end{pmatrix}. \quad (2.26)$$

Entsprechend auch für die übrigen Operatormatrizen bzw. für den TM-Fall.

- In anderen Fällen wird für  $\mathbf{P}_z$  häufig eine aus analytischen Überlegungen gewonnene Matrix eingesetzt, z.B. für Hohlleiter  $\mathbf{P}_z = -jk_z \mathbf{I}$  ( $\mathbf{I}$ =Einheitsmatrix). Die Trennung von TE- und TM-Feldern ist dann nicht mehr generell zulässig (gilt aber wieder in einigen wichtigen Sonderfällen). Dies soll hier nicht weiter behandelt werden.
- Während in 3D die Gitterlinien des primären und dualen Gitters in jede Raumrichtung versetzt angeordnet sind, gilt dies im 2D-Fall nicht für die normale Richtung, wo beide Gitter in der selben Ebene (der Zeichenebene) definiert sind. Formal entspricht dies einem Grenzübergang  $\Delta z \rightarrow 0$ .
- Wird der Zusammenhang zwischen einer Gitterspannung und dem über die Materialeigenschaft damit verknüpften Fluss betrachtet, so stellt sich die Frage, welches die relevante Fläche ist, auf der der entsprechende Fluss allokiert ist. Beispiel: Betrachtet man die in Abb. 2.4 eingetragene Gitterspannung  $\hat{e}_x(n + M_y)$  und bestimmt – durch Anwendung einer Zeile der Matrix  $\mathbf{M}_e$  – den zugehörigen, elektrischen Gitterfluss  $\hat{d}_x(n + M_y)$ , so ist dieser Fluss auf der dualen Gitterfläche  $\tilde{A}_x(n + M_y)$  zu allokiieren. Diese Gitterfläche ist allerdings in einem zweidimensionalen Gitter nicht vorhanden, da eine Ausdehnung in  $z$ -Richtung fehlt.

Dieses Problem wird in der Praxis gelöst, indem man in die – nicht vorhandene – dritte Raumrichtung eine Einheitsgitterschrittweite mit  $\Delta z = 1$  annimmt. Es kann gezeigt werden, dass dies kein Widerspruch zu dem oben eingeführten Konzept eines Grenzübergangs  $\Delta z \rightarrow 0$  darstellt, da sich  $\Delta z$  in allen relevanten Formeln herauskürzt.

---

## 2.3 Versuchsdurchführung

---

### 2.3.1 Vorbereitungsaufgaben

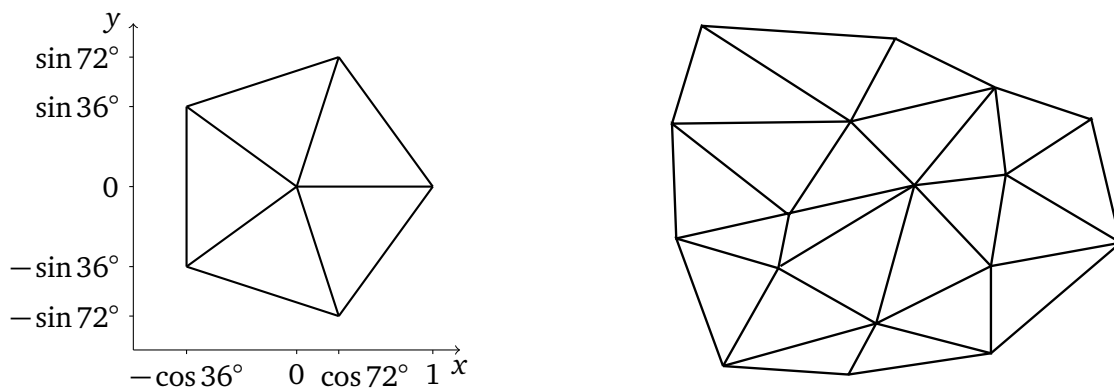
---

#### 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.
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  $N_p = N_{xy}^2$  an. Geben Sie darüber hinaus auch eine Formel an, um die Indizes aller Geisterkanten nach dem kanonischen Indizierungsschema zu berechnen.

#### Dreiecksgitter

Gegeben sind die beiden Dreiecksgitter in Abb. 2.6, wobei zunächst nur das linke Gitter betrachtet werden soll.



**Abbildung 2.6:** Dreiecksgitter von zwei verschiedenen Rechengebieten.

3. Nummerieren Sie die Flächen und Kanten des Gitters beliebig und ordnen Sie den Kanten eine Orientierung zu.
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.
5. Erstellen Sie aus der Kanten-zu-Knoten-Inzidenz die Gradientenmatrix  $\mathbf{G}$ . Gehen Sie von einem Potentialvektor  $\varphi$  der Dimension  $N_p$  aus, der die Werte einer Potentialfunktion in allen Gitterpunkten enthält. Legen Sie die Matrix  $\mathbf{G}$  so fest, dass die Multiplikation  $-\mathbf{G}\varphi$  gerade den Vektor  $\hat{\mathbf{e}}$  ergibt, was der kontinuierlichen Formel  $\vec{E} = -\text{grad } \varphi$  entspricht.
6. Konstruieren Sie mithilfe der Flächen-zu-Kanten-Inzidenz die Curlmatrix  $\mathbf{C}$ .  
Zur Erinnerung:  $\mathbf{C}\hat{\mathbf{e}} = -\frac{d}{dt}\hat{\mathbf{b}}$ .
7. Überprüfen Sie, ob genau wie im kontinuierlichen Fall die Beziehung  $\text{rot grad} = 0$  auch für die aufgestellten diskreten Matrizen  $\mathbf{C}\mathbf{G} = \mathbf{0}$  gilt.

---

## 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.

**Ab hier sollen beide Gitter aus Abb. 2.6 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.
9. Überlegen Sie sich, wie  $N_V$ ,  $N_A$ ,  $N_L$  und  $N_P$  mit den entsprechenden Größen des dualen Gitters  $\tilde{N}_V$ ,  $\tilde{N}_A$ ,  $\tilde{N}_L$  und  $\tilde{N}_P$  im Fall von 3D-Gittern zusammenhängen. Besonderheiten am Rand sind hierzu zu vernachlässigen. Wie verhalten sich die Größen im Fall von 2D-Gittern?

---

### 2.3.2 Aufgaben während der Praktikumssitzung

---

#### Datenstruktur, Visualisierung des Gitters

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

$$[msh] = \text{cartMesh}(xmesh, ymesh, zmesh) \quad (2.27)$$

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 Gitterpunktzahl in jede Raumrichtung, d. h.  $n_x$ ,  $n_y$  und  $n_z$ , abgespeichert werden.

2. Implementieren Sie die Methode

$$\text{plotMesh}(msh), \quad (2.28)$$

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

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`.

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

4. Schreiben Sie eine Methode

$$[c, s, st] = \text{geoMats}(msh), \quad (2.29)$$

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

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

5. Lassen Sie sich die Matrizen für eine kleine Problemgröße ( $N_p < 50$ ) direkt ausgeben und visualisieren Sie die Matrizen für eine mittlere Problemgröße ( $N_p < 5000$ ) mit dem Befehl `spy`. Welche speichertechnisch günstige Eigenschaft würde ohne das kanonische Indizierungsschema verloren gehen? Ermitteln Sie wie viel Speicherplatz jeweils von MATLAB<sup>®</sup> 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`.

6. Berechnen Sie

- $C(-\tilde{S}^T)$  und
- $SC$  bzw.  $\tilde{S}\tilde{C}$ .

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

### Unbelegte Kantenelemente

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

$$\text{edg} = \text{boundEdg}(\text{msh}) \quad (2.30)$$

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  $n_z=1$ , 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<sup>®</sup> 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.

### 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:

$$[\text{fieldBow}] = \text{impField}(\text{msh}, \text{field}) \quad (2.31)$$

**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:

a)  $\vec{E}(\vec{r}) = \frac{5}{2} \vec{e}_x - 1,3 \vec{e}_y + 2 \vec{e}_z,$

b)  $\vec{E}(\vec{r}) = 3 \sin\left(\frac{\pi}{x_{\max}-x_{\min}} (x - x_{\min})\right) \vec{e}_y,$

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.

---

### 2.3.3 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 2.6 a). Ist diese Wahl des dualen Gitters am Rand immer notwendig?

---

### 2.3.4 Nützliche MATLAB<sup>®</sup>/GNU OCTAVE Befehle und Hilfsroutinen

---

MATLAB <sup>®</sup> - Befehle	
Befehl	Beschreibung
<code>struct('name1', values1, 'name2', values2, ...)</code>	Erzeugt eine Struktur mit den Werten <code>values</code> und dem zugeordneten Namen <code>name</code> , worüber diese mit <code>structname.name</code> angerufen werden können.
<code>axis tight</code>	Passt die Achsen des Graphs den tatsächlichen Intervallen der dargestellten Daten an.
<code>hold on</code>	Die graphischen Ausgaben nachfolgender Befehle, die eine solche erzeugen, werden in denselben Graph gezeichnet. Außerdem werden bestimmte Eigenschaften des Bildes (z.B. Eigenschaften der Achsen) eingefroren. Der Befehl <code>hold off</code> schaltet dies wieder aus.
<code>line([x1 xh],[y1 yh],[z1 zh])</code>	Erzeugt eine Linie in dreidimensionalen Koordinaten. Die beiden Enden der Linie werden durch die Koordinaten $(x_1, y_1, z_1)$ bzw. $(x_h, y_h, z_h)$ definiert.
<code>speye(m,n)</code>	Erzeugt eine $m \times n$ -Matrix mit Einsen auf der Hauptdiagonalen.
<code>sparse(i,j,s,m,n)</code>	Erzeugt eine dünn besetzte $m \times n$ -Matrix $S$ , so dass $S(i(n), j(n)) = s(n)$ , wobei $n$ gleich die Länge der Vektoren $i$ , $j$ und $s$ ist, gilt.
<code>spy(matrix)</code>	Visualisiert das Besetzungsmuster (engl. <i>sparsity pattern</i> ) einer Matrix.



Hilfsroutinen		
Befehl	Beschreibung	
plotEdgeVoltage(...)	Stellt die Werte von volt als Vektoren in einer x-y-Ebene dar.	
	Parameter	
	msh	Struktur mit Gitterdaten: Anzahl der Gitterpunkte in jede Raumrichtung: msh.nx, msh.ny, msh.nz. Implementierung z.B.: msh.nx=Nx Koordinaten der Gitterlinien in jede Raumrichtung: msh.xmesh, msh.ymesh, msh.zmesh. Implementierung z.B.: msh.xmesh=[x1:dx:xh] mit x1/xh := kleinste/größte x-Koordinate und dx := Abstand zwischen 2 Gitterpunkten.
	vec	Darzustellende Daten des Feldes (z.B. E-Feld) gemäß dem kanonischen Indizierungsschema bzw. Gl. (2.5) und (2.8).
	izcut	Index der darzustellenden xy-Gitterebene.
	bc	Randbedingungen: [xlow xhigh ylow yhigh zlow zhigh], dabei bedeutet z.B. xlow = 1 einen elektrischen Rand am unteren x-Ende des Rechengebietes.

