
Ausarbeitung zum Softwarepraktikum Verfahren und Anwendung der Feldsimulation

Denis Andric
Marc Bodem
Theodor Sperling



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Inhaltsverzeichnis

1	Einführung in Numerische Methoden	2
1.1	Vorbereitungsaufgaben	2
1.1.1	Differenzenverfahren	2
1.1.2	Dreidimensionale Darstellung	7
1.2	Aufgaben während der Praktikumssitzung	10
1.2.1	Differenzenverfahren	10
1.2.2	Dreidimensionale Darstellung	12
1.3	Fazit	13
2	Grundlagen der Methode der Finiten Integration 1	15
2.1	Vorbereitungsaufgaben	15
2.1.1	Überzählige Kanten	15
2.1.2	Dreiecksgitter	16
2.1.3	Duale Gitter	19
2.1.4	Ab hier sollen beide Gitter aus Abb. 2.2 betrachtet werden.	19
2.2	Aufgaben während der Praktikumssitzung	20
2.2.1	Datenstruktur, Visualisierung des Gitters	21
2.2.2	Die topologischen Matrizen \mathbf{C} , $\tilde{\mathbf{C}}$, \mathbf{S} und $\tilde{\mathbf{S}}$	22
2.2.3	Unbelegte Kantenelemente	23
2.2.4	Einprägen gegebener Feldverteilungen	24
2.3	Fragen zur Ausarbeitung	26
2.4	Fazit	26
3	Grundlagen der Methode der Finiten Integration 2	28
3.1	Vorbereitungsaufgaben	28
3.2	Aufgaben während der Praktikumssitzung	29
3.2.1	Materialmatrizen	29
3.2.2	Interpolation und Visualisierung	30
3.3	Fragen zur Ausarbeitung	31
3.4	Fazit	31

1 Einführung in Numerische Methoden

1.1 Vorbereitungsaufgaben

1.1.1 Differenzenverfahren

1. Zeigen Sie, dass der zentrale Differenzenquotient (1.5) aus der Subtraktion zweier Taylor-Entwicklungen zu den Punkten x_{i+1} und x_{i-1} folgt. Vergessen Sie dabei nicht, die Fehlerterme zu berücksichtigen und kommentieren Sie die Ordnung des Fehlers im Vergleich zu vorwärts- und rückwärts-Differenzenquotient.

Der zentrale Differenzenquotient

$$f^{(1)}(x_i) = \frac{f(x_{i+1}) - f(x_{i-1}))}{\Delta x_{i-1} + \Delta x_i} + \mathcal{O}(\Delta x^{1 \dots 2}) \quad (1.1)$$

setzt sich zusammen aus dem Vorwärts-Differenzenquotienten

$$f^{(1)}(x_i) = \frac{f(x_{i+1}) - f(x_i)}{\Delta x_i} + \mathcal{O}(\Delta x) \quad (1.2)$$

und dem Rückwärts-Differenzenquotienten

$$f^{(1)}(x_i) = \frac{f(x_i) - f(x_{i-1}))}{\Delta x_{i-1}} + \mathcal{O}(\Delta x) \quad (1.3)$$

Im folgenden wird die Herleitung des Zentralen Differenzenquotienten bei homogenen Abständen Δx dargestellt.

Durch Subtraktion der Taylor Entwicklung zweiter Ordnung zu 1.3 von der Entwicklung zu 1.2 ergibt sich

$$\begin{aligned} f(x_{i+1}) - f(x_{i-1}) &= f(x_i) + \Delta x_i \cdot f'(x_i) + \Delta x_i \mathcal{O}(\Delta x_i) - (f(x_i) - \Delta x_{i-1} \cdot f'(x_i) + \Delta x_{i-1} \mathcal{O}(\Delta x_{i-1})) \\ &= (\Delta x_i + \Delta x_{i-1}) f'(x_i) + \mathcal{O}(\Delta x_i \Delta x_{i-1}) \end{aligned} \quad (1.4)$$

Hieraus ergibt sich durch Umstellen nach $f'(x_i)$ der Zentrale Differenzenquotient 1.1.

Der Fehler liegt zwischen $\mathcal{O}(\Delta x)$ und $\mathcal{O}(\Delta x^2)$ Abhängig vom Verhältnis der Abstände. Für äquidistante Abstände ist der Fehler $\mathcal{O}(\Delta x^2)$.

2. Berechnen Sie ausgehend von einem Startpunkt $f(\tilde{x}_i)$ auf einem dualen Gitter mit Hilfe von Taylorentwicklungen analog zur Differenzenvorschrift (1.6) im Fall äquidistanter Gitter eine zentrale Differenzenvorschrift vierter Ordnung zur Berechnung der ersten Ableitung. Verwenden Sie hierfür $f(x_i), f(x_{i+1}), f(x_{i+2})$ sowie $f(x_{i-1})$.

Um eine Differentiationsvorschrift vierter Ordnung für einen Startpunkt \tilde{x}_i im dualen Gitter zu bestimmen werden nach der Vorschrift für die den Zentralen Differenzenquotienten die Gleichungen

$$\begin{aligned} f(x_i) &= f(\tilde{x}_i) - \frac{h}{2}f'(\tilde{x}_i) + \frac{h^2}{8}f''(\tilde{x}_i) - \frac{h^3}{48}f'''(\tilde{x}_i) + \mathcal{O}(h^4) \\ f(x_{i+1}) &= f(\tilde{x}_i) + \frac{h}{2}f'(\tilde{x}_i) + \frac{h^2}{8}f''(\tilde{x}_i) + \frac{h^3}{48}f'''(\tilde{x}_i) + \mathcal{O}(h^4) \\ f(x_{i+2}) &= f(\tilde{x}_i) + \frac{3h}{2}f'(\tilde{x}_i) + \frac{9h^2}{8}f''(\tilde{x}_i) + \frac{27h^3}{48}f'''(\tilde{x}_i) + \mathcal{O}(h^4) \\ f(x_{i-1}) &= f(\tilde{x}_i) - \frac{3h}{2}f'(\tilde{x}_i) + \frac{9h^2}{8}f''(\tilde{x}_i) - \frac{27h^3}{48}f'''(\tilde{x}_i) + \mathcal{O}(h^4) \end{aligned} \quad (1.5)$$

mit $\Delta x = h$ aufgestellt.

Hieraus folgt somit

$$f(x_{i+1}) - f(x_i) = hf'(\tilde{x}_i) + \frac{1}{24}h^3f'''(\tilde{x}_i) + \mathcal{O}(h^4) \quad (1.6)$$

und

$$f(x_{i+2}) - f(x_{i-1}) = 3hf'(\tilde{x}_i) + \frac{27}{24}h^3f'''(\tilde{x}_i) + \mathcal{O}(h^4) \quad (1.7)$$

Damit nun die 3. Ableitungen weggelassen wird ein Koeffizientenvergleich von a und c durchgeführt. Mit

$$\begin{aligned} a[f(x_{i+1}) - f(x_i)] &= a[hf'(\tilde{x}_i) + \frac{1}{24}h^3f'''(\tilde{x}_i)] + \mathcal{O}(h^4) \\ c[f(x_{i+2}) - f(x_{i-1})] &= c[3hf'(\tilde{x}_i) + \frac{27}{24}h^3f'''(\tilde{x}_i)] + \mathcal{O}(h^4) \end{aligned}$$

ergibt sich somit für $a = -27c$

$$-27[f(x_{i+1}) - f(x_i)] + [f(x_{i+2}) - f(x_{i-1})] = -24hf'(\tilde{x}_i)$$

Woraus sich die Differenzenvorschrift vierter Ordnung zur Berechnung der ersten Ableitung zu

$$f'(\tilde{x}_i) = \frac{27[f(x_{i+1}) - f(x_i)] - [f(x_{i+2}) - f(x_{i-1})]}{24h} \quad (1.8)$$

3. Bestimmen Sie aus der Differenzenvorschrift (1.10) die Matrix $\tilde{\mathbf{C}}\mathbf{C}$ für einen 1D-Resonator mit 5 Stützstellen für den Fall elektrischer bzw. magnetischer Randbedingungen an beiden Rändern (siehe Abb. 1.3, Gl. (1.15) und Gl. (1.16)).

Nach (1.10) aus dem Skript ist der Differenzenquotient vierter Ordnung gegeben durch

$$f^{(2)}(x_i) = \frac{16(f(x_{i+1}) + f(x_{i-1})) - (f(x_{i+2}) + f(x_{i-2})) - 30f(x_i)}{12\Delta x^2} + \mathcal{O}(\Delta x^2). \quad (1.9)$$

Betrachtet man weiterhin die eindimensionale Wellengleichung

$$\frac{d^2 \underline{E}_y}{dx^2} = -k_x^2 \underline{E}_y$$

kann man nun den Differenzenquotienten aus (1.9) in diese Gleichung einsetzen. Dazu werden die Funktionswerte in einen Vektor \mathbf{e} geschrieben. Die Koeffizienten der Funktionswerte aus (1.9) werden in der Matrix $\tilde{\mathbf{C}}\mathbf{C}$ gesammelt, sodass sich folgende Diskretisierung des Problems ergibt:

$$\frac{1}{12}\tilde{\mathbf{C}}\mathbf{C}\mathbf{e} = -\Delta x^2 k_x^2 \mathbf{e} \quad (1.10)$$

mit

$$\mathbf{e} = \begin{pmatrix} \vdots \\ f(x_{i-1}) \\ f(x_i) \\ f(x_{i+1}) \\ \vdots \end{pmatrix}$$

$$\tilde{\mathbf{C}}\mathbf{C} = \begin{pmatrix} -30 & 16 & -1 & 0 & 0 & 0 & \dots \\ 16 & -30 & 16 & -1 & 0 & 0 & \dots \\ -1 & 16 & -30 & 16 & -1 & 0 & \dots \\ 0 & -1 & 16 & -30 & 16 & -1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & -1 & 16 & -30 & 16 \\ \vdots & \vdots & \vdots & 0 & -1 & 16 & -30 \end{pmatrix}$$

Die Matrix $\tilde{\mathbf{C}}\mathbf{C}$ beinhaltet aber noch keine Randbedingungen. Wir können zwei unterschiedliche Arten von Randbedingungen ansetzen. Einmal elektrische Randbedingungen und einmal magnetische Randbedingungen. Bei elektrischen Randbedingungen gilt eine ungerade Symmetrie, das heißt, dass wenn sich $f(x_0)$ am Rand befindet ($f(x_{-1}) = -f(x_1)$) gilt. Bei magnetischen Randbedingungen gilt im Gegensatz dazu eine gerade Symmetrie mit ($f(x_{-1}) = f(x_1)$). Will man nun die Randbedingungen einpflegen, so muss man jeweils die ersten und letzten beiden Zeilen anpassen.

Für elektrische Randbedingungen in den ersten beiden Zeilen gilt folgendes:

$$-\frac{1}{12}f(x_{-2}) + \frac{16}{12}f(x_{-1}) - \frac{30}{12}f(x_0) + \frac{16}{12}f(x_1) - \frac{1}{12}f(x_2) = -\Delta x^2 k_x^2 f(x_0)$$

$$-\frac{30}{12}f(x_0) \stackrel{\text{ung.Symm.}}{=} -\Delta x^2 k_x^2 f(x_0)$$

$$-\frac{1}{12}f(x_{-1}) + \frac{16}{12}f(x_0) - \frac{30}{12}f(x_1) + \frac{16}{12}f(x_2) - \frac{1}{12}f(x_3) = -\Delta x^2 k_x^2 f(x_1)$$

$$\frac{16}{12}f(x_0) - \frac{29}{12}f(x_1) + \frac{16}{12}f(x_2) - \frac{1}{12}f(x_3) \stackrel{\text{ung.Symm.}}{=} -\Delta x^2 k_x^2 f(x_1)$$

Bei magnetischen Randbedingungen gilt dagegen:

$$\begin{aligned}
-\frac{1}{12}f(x_{-2}) + \frac{16}{12}f(x_{-1}) - \frac{30}{12}f(x_0) + \frac{16}{12}f(x_1) - \frac{1}{12}f(x_2) &= -\Delta x^2 k_x^2 f(x_0) \\
-\frac{30}{12}f(x_0) + \frac{32}{12}f(x_1) - \frac{2}{12}f(x_2) &\stackrel{g.Symm.}{=} -\Delta x^2 k_x^2 f(x_0) \\
\\
-\frac{1}{12}f(x_{-1}) + \frac{16}{12}f(x_0) - \frac{30}{12}f(x_1) + \frac{16}{12}f(x_2) - \frac{1}{12}f(x_3) &= -\Delta x^2 k_x^2 f(x_1) \\
\frac{16}{12}f(x_0) - \frac{31}{12}f(x_1) + \frac{16}{12}f(x_2) - \frac{1}{12}f(x_3) &\stackrel{g.Symm.}{=} -\Delta x^2 k_x^2 f(x_1)
\end{aligned}$$

Das Ganze gilt analog für die letzten Beiden Zeilen der $\tilde{\mathbf{C}}\mathbf{C}$ -Matrix, jedoch gespiegelt für den rechten Rand. Damit lassen sich die Matrizen für elektrische bzw. magnetische Randbedingungen aufstellen:

$$\begin{aligned}
\tilde{\mathbf{C}}\mathbf{C}_{elek} &= \begin{pmatrix} -30 & 0 & 0 & 0 & 0 & 0 & \dots \\ 16 & -29 & 16 & -1 & 0 & 0 & \dots \\ -1 & 16 & -30 & 16 & -1 & 0 & \dots \\ 0 & -1 & 16 & -30 & 16 & -1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & -1 & 16 & -29 & 16 \\ \vdots & \vdots & \vdots & 0 & 0 & 0 & -30 \end{pmatrix} \\
\\
\tilde{\mathbf{C}}\mathbf{C}_{magn} &= \begin{pmatrix} -30 & 32 & -2 & 0 & 0 & 0 & \dots \\ 16 & -31 & 16 & -1 & 0 & 0 & \dots \\ -1 & 16 & -30 & 16 & -1 & 0 & \dots \\ 0 & -1 & 16 & -30 & 16 & -1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & -1 & 16 & -31 & 16 \\ \vdots & \vdots & \vdots & 0 & -2 & 32 & -30 \end{pmatrix}
\end{aligned}$$

4. Betrachtet werden soll exemplarisch die Berechnung der diskreten Wellenzahlen bei vorgegebener Länge des Rechengebietes nach Gl. (1.14). Geben Sie den Abbruchfehler in Abhängigkeit von der Anzahl der verwendeten Stützstellen n bzw. der Diskretisierungsschrittweite Δx an, wenn der Differenzenquotient nach Gl. (1.9) bzw. Gl. (1.10) verwendet wird. Wie müssen Sie ein entsprechendes Diagramm (Abbruchfehler vs. Anzahl der Stützstellen) skalieren, um einen geradlinigen Verlauf zu erhalten?

Ergänzt man (1.10) mit dem Fehler $\mathcal{O}(\Delta x^2)$ so erhält man

$$\frac{1}{12\Delta x^2} \tilde{\mathbf{C}}\mathbf{C}\mathbf{e} + k_x^2 \mathbf{e} + \mathcal{O}(\Delta x^4) = 0 \tag{1.11}$$

Der Fehler dieser Gleichung liegt also in der Ordnung $\mathcal{O}(\Delta x^4)$. Mit $\Delta x = \frac{L}{n-1}$, wobei L die Länge des Rechengebietes und n die Anzahl der Stützstellen ist. Damit kann man die Ordnung ganz einfach in

Abhängigkeit der Stückstellenanzahl schreiben. $\mathcal{O}(\frac{1}{(n-1)^4})$ ist hierbei nicht von L anhängig, da dies nur eine Konstante darstellt.

Diese Rechnung beschreibt das Vorgehen mit dem in (1.9) gezeigt Differenzenquotient vierter Ordnung. Das Ganze funktioniert analog mit dem Differenzenquotienten zweiter Ordnung

$$f^{(2)}(x_i) = \frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$

wobei sich hier dann die Ordnung $\mathcal{O}(\frac{1}{(n-1)^2})$ ergibt.

Will man einen gradlinigen Verlauf des Fehlers erhalten, so sollte man ihn doppel-logarithmisch plotten, weil dadurch der Exponent des Terms $(n-1)$ nur als einfacher Faktor berücksichtigt wird.

5. Stellen Sie eine Formel auf, mit der die analytischen Wellenzahlen $k_{x, \text{ana}}$ für die eindimensionale Wellengleichung und einem Resonator der Länge L einmal mit rein elektrischer Berandung und einmal mit unterschiedlichen Randbedingungen (eine Seite elektrisch – eine Seite magnetisch) berechnet werden kann. Bestimmen Sie dabei auch die jeweils kleinste Wellenzahl. Geben Sie bei einer numerischen Berechnung eine Formel für den relativen Wellenzahlfehler Δk_x an.

Zur Bestimmung der Wellenzahlen $k_{x, \text{ana}}$ betrachten wir wieder

$$\frac{d\underline{E}_y}{dx^2} + k_x^2 \underline{E}_y = 0.$$

Zur Lösung dieser gewöhnlichen Differentialgleichung 2. Ordnung wählt man den Ansatz

$$\underline{E}_y = c_1 \cos(k_x x) + c_2 \sin(k_x x) \quad (1.12)$$

Im Fall beidseitiger elektrischer Randbedingungen ergeben sich folgende Randbedingungen für \underline{E}_y :

$$\begin{aligned} \text{(I)} \quad \underline{E}_y(0) &= 0 \\ \text{(II)} \quad \underline{E}_y(L) &= 0 \end{aligned}$$

Mit (I) und (1.12) folgt direkt $c_1 = 0$. Mit (II) und (1.12) folgt $c_2 \sin(k_x L) = 0$. Damit dies erfüllt wird muss $k_x = \frac{n\pi}{L}$ mit $n \in \mathbb{Z}$ gelten. Mit $c_2 = 0$ wird die triviale Nulllösung erreicht, aber man will alle möglichen Lösungen finden. Die kleinste Wellenzahl ist damit $k_x = \frac{\pi}{L}$.

Im Fall, das eine Seite eine elektrische und eine Seite eine magnetische Randbedingung besitzt, sehen die Randbedingungen folgendermaßen aus:

$$\begin{aligned} \text{(I)} \quad \underline{E}_y(0) &= 0 \\ \text{(II)} \quad \underline{H}_x(L) &= 0 \end{aligned}$$

Für (II) kann man das Faraday'sche Gesetz im Frequenzbereich $\text{rot} \underline{E}_y = -j\omega\mu \underline{H}_x$ verwenden. Da die Rotation im eindimensionalen nur eine einfache Ableitung beschreibt, kann man $\frac{d\underline{E}_y}{dx} \propto \underline{H}_x$ schreiben. Damit folgt aus (II)

$$\text{(III)} \quad \frac{d\underline{E}_y(L)}{dx} = 0.$$

Verwendet man nun wieder (1.12) und (I) ergibt sich $c_1 = 0$. Mit (III) und (1.12) ergibt sich $c_2 k_x \cos(k_x L) = 0$ mit $n \in \mathbb{Z}$. Damit dies erfüllt wird muss $k_x = \frac{\pi}{L}(\frac{n}{2} + 1)$ gelten. $c_2 = 0$ wäre die einfache triviale Lösung. Die kleinste Wellenzahl ist damit $k_x = \frac{\pi}{L}$.

6. Wie kann man die Orthogonalität zweier Eigenvektoren testen und was sagen orthogonale Eigenvektoren über die Lösungen eines Eigenwertproblems (Moden) aus?

Die Eigenvektoren können durch Bildung des Skalarproduktes und Überprüfung $\vec{A} \cdot \vec{B} = 0$ auf Orthogonalität getestet werden.

Orthogonalität der Eigenvektoren bedeutet, dass alle Moden senkrecht zueinander stehen. Darüber hinaus wird durch die Vektoren eine Ortoqonalbasis gebildet mit welcher durch Linearkombination alle Lösungen abgebildet werden können.

1.1.2 Dreidimensionale Darstellung

Gegeben sei in Abbildung 1.1 die Diskretisierung eines Kreiszylinders der Höhe h und des Radius r mit n_D Deckflächendreiecken der Grundlänge $2a_D$ und Höhe h_D .

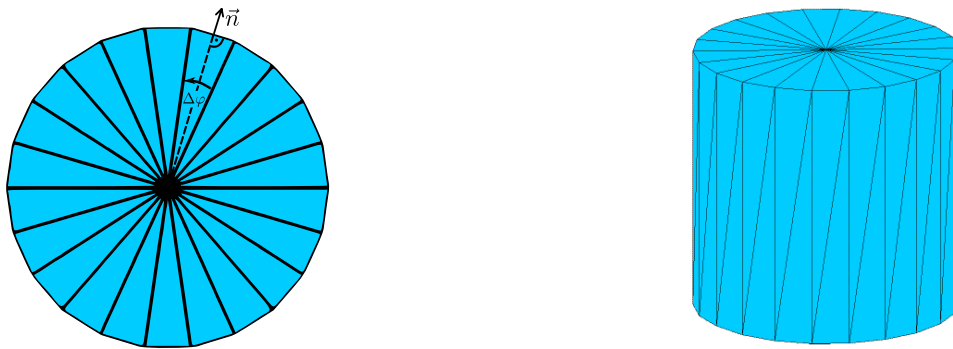


Abbildung 1.1: Diskretisierung einer Kreiszylinderoberfläche. Dargestellt sind die Dreiecksgitter der Diskretisierung der Deckelfläche (links) und des diskretisierten Zylinders in Perspektivansicht (rechts).

7. Überlegen Sie sich einen Pseudocode mit For-Schleifen, mit welchem Sie die Koordinaten aller Dreiecke der Oberflächendiskretisierung (siehe Abb. 1.1) bestimmen können. Die Anzahl der Dreiecke n_D soll dabei beliebig sein. Beachten Sie, dass zur Oberfläche Deck-, Mantel- und Bodenfläche gehören. Außerdem besitzen die Dreiecke auf der Mantelfläche „unterschiedliche Orientierungen“.

Um die Koordinaten aller Dreiecke in Abhängigkeit von n_D zu bestimmen, wird zunächst der Deckel betrachtet. Ein Punkt am Rand des Deckels hat immer die Koordinaten $\begin{pmatrix} r \cos(i \Delta \varphi) \\ r \sin(i \Delta \varphi) \\ h \end{pmatrix}$. Dabei beschreibt

$\Delta \varphi$ den Winkel zwischen 2 Dreiecksseiten. Dieser ergibt sich aus $\Delta \varphi = \frac{2\pi}{n_D}$. Das i beschreibt den Laufindex für die for-Schleife. Hat das Dreieck als Koordinaten von 2 Punkten $\begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix}, \begin{pmatrix} r \cos(i \Delta \varphi) \\ r \sin(i \Delta \varphi) \\ h \end{pmatrix}$ gegeben, so ist

der dritte Punkt gegeben durch $\begin{pmatrix} r \cos(i \Delta \varphi + \Delta \varphi) \\ r \sin(i \Delta \varphi + \Delta \varphi) \\ h \end{pmatrix}$. Diese Punkte werden nun mit einer for-Schleife

abgetastet und gespeichert. Das gleiche geschieht auch mit den Dreiecken am Boden des Zylinders, nur, dass die z-Koordinate hier immer 0 ist.

Für den Mantel gilt es jetzt nur die schon berechneten Punkte wiederzuverwerten. Dreiecke bei denen 2 Punkte am Boden und einer am Deckel ist, müssen einen Punkt von Boden, den darauf folgenden Punkt sowie den Punkt, der in der Höhe h im Vergleich zum ersten Punkt liegt abspeichern. Für die Dreiecke, die andersherum orientiert sind gilt das gleiche, nur sind hier Deckel und Boden vertauscht.

8. In der Theorie wurde erwähnt, dass eine feinere Diskretisierung die Genauigkeit der Darstellung steigert¹, aber dass auch der Rechenaufwand und der geforderte Speicher zunehmen. Berechnen Sie den Speicherplatz, der zur Speicherung des oben abgebildeten, diskretisierten Zylinders im STL-Format notwendig ist, als Funktion der Anzahl der Dreiecke n_D (Normalenvektoren müssen auch berücksichtigt werden). Beachten Sie hierbei lediglich die notwendige Anzahl der double-Zahlen, wobei eine double-Zahl 8 Byte benötigt.

Im STL Format ist jedes Dreieck durch 3 Punkte und einen Normalenvektor beschrieben. Jeder Punkt und jeder Normalenvektor hat 3 Koordinaten. Dazu besteht jeder Zylinder aus $4 \cdot n_D$ Dreiecken. Damit kann man den benötigte Speicherplatz zu

$$\begin{aligned}\text{Speicher}(n_D) &= 4n_D(3 \text{ Punkte} + \text{Normalenvektor}) \cdot 3 \text{ Koordinaten} \cdot 8 \text{ Bytes} \\ &= 384n_D \text{ Bytes}\end{aligned}$$

bestimmen.

9. Gegeben ist folgender Zusammenhang für die Fläche eines Deckflächendreiecks:

$$A_D = 2 \left[\frac{1}{2} a_D h_D \right] = 2 \left[\frac{1}{2} \cdot r \cos\left(\frac{\Delta\varphi}{2}\right) \cdot r \sin\left(\frac{\Delta\varphi}{2}\right) \right] = \frac{1}{2} r^2 \sin(\Delta\varphi) \quad (1.13)$$

Überlegen sie sich geometrisch, wie diese Formel zustande kommt und dokumentieren Sie die einzelnen Schritte. Leiten Sie anschließend eine Formel zur Berechnung des Volumens und der Fläche des gezeigten Zylinders in Abhängigkeit von der Anzahl der Deckflächendreiecke n_D her. Bestimmen Sie den relativen Oberflächenfehler ΔA bzw. Volumenfehler ΔV dieser Diskretisierung. Bei welchen Körpern wäre eine solche Oberflächendiskretisierung mit Dreiecken exakt?

Ein Dreieck der Deckfläche ist ein gleichschenkliges Dreieck. Um die Fläche des gleichschenkligen Dreiecks zu bestimmen, soll man es mithilfe der Höhe h_D in zwei rechtwinklige Dreiecke teilen. Die Fläche der rechtwinkligen Dreieck errechnet sich als Multiplikation zweier Katheten geteilt durch 2. Damit bekommt man die Formel:

$$A_D = 2 \left[\frac{1}{2} a_D h_D \right]$$

Mit der Pythagoras Formel rechnet ergibt sich aus h_D und a_D :

$$a_D = r \sin\left(\frac{\Delta\varphi}{2}\right)$$

$$h_D = r \cos\left(\frac{\Delta\varphi}{2}\right)$$

Jetzt setzt man alles zusammen:

$$A_D = 2 \left[\frac{1}{2} \cdot r \cos\left(\frac{\Delta\varphi}{2}\right) \cdot r \sin\left(\frac{\Delta\varphi}{2}\right) \right]$$

Mithilfe der Sinusdoppelwinkelformel

$$\sin(\Delta\varphi) = 2 \cos\left(\frac{\Delta\varphi}{2}\right) \sin\left(\frac{\Delta\varphi}{2}\right)$$

kann man der Formel noch vereinfachen auf:

$$A_D = \frac{1}{2} r^2 \sin(\Delta\varphi)$$

Wenn man den relativen Oberflächen- und Volumenfehler bestimmen wollte, sollte man die Oberfläche und das Volumen des wahren und des diskreten Zylinders ausrechnen.

Eine Formel für die Oberfläche des diskreten Zylinders(A_{disc}) erhält man mit der Summation über alle Dreiecksflächen. Die Oberfläche besteht aus 2 Grundflächen ($2 \cdot n_D A_D$) und einem Mantel ($2n_D \cdot \frac{1}{2} a_D h$). Eine Formel für das Volumen des diskreten Zylinders(V_{disc}) erhält man mit der Multiplikation mit der Grundbasis ($n_D A_D$) und der Höhe (h).

$$A_{\text{disc}} = 2n_D r \left[\frac{1}{2} \cdot r \sin\left(\frac{2\pi}{n_D}\right) + \sin\left(\frac{\pi}{n_D}\right) h \right]$$

$$V_{\text{disc}} = \frac{1}{2} n_D r^2 \sin\left(\frac{2\pi}{n_D}\right) h$$

$$A = 2r\pi(r+h)$$

$$V = r^2\pi h$$

Die Formel für den absolute Fehler lautet:

$$\Delta X = X - X_{\text{disc}}$$

Damit bekommt man

$$\Delta A = 2r\pi(r+h) - 2n_D r \left[\frac{1}{2} \cdot r \sin\left(\frac{2\pi}{n_D}\right) + \sin\left(\frac{\pi}{n_D}\right) h \right]$$

$$\Delta V = r^2\pi h - \frac{1}{2} n_D r^2 \sin\left(\frac{2\pi}{n_D}\right) h$$

Den relative Fehler bekommt man mit der Formel:

$$\varepsilon_x = \frac{\Delta X}{X}$$

Damit bekommt man:

$$\varepsilon_A = \frac{\pi(r+h) - n_D \left[\frac{1}{2} \cdot r \sin\left(\frac{2\pi}{n_D}\right) + \sin\left(\frac{\pi}{n_D}\right) h \right]}{\pi(r+h)}$$

$$\varepsilon_V = \frac{\pi - \frac{1}{2} n_D \sin\left(\frac{2\pi}{n_D}\right)}{\pi}$$

10. Bei einigen Anwendungen ist es wichtig, dass die diskretisierte Fläche möglichst glatt bleibt. Ein wichtiges Beispiel hierbei ist die Streuung hochfrequenter elektromagnetischer Felder, wobei Kanten und Ecken das gestreute Feld verändern können. Angenommen bei der oberen Diskretisierung des Zylinders ist der Winkel zwischen den Normalenvektoren benachbarter Dreiecke der Deckflächen $\Delta\varphi \leq 5^\circ$ gefordert. Berechnen Sie die minimale Anzahl an Dreiecken zur Erfüllung dieser Forderung.

Dieser Forderung rechnet man mit der schon bekannten Formel 1.1:

$$\Delta\varphi = \frac{360^\circ}{n_D}$$

aus. Realisierung der Forderung:

$$5^\circ \geq \frac{360^\circ}{n_D}$$

$$n_D \geq 72$$

Es werden minimal 72 Dreiecke benötigt.

1.2 Aufgaben während der Praktikumssitzung

1.2.1 Differenzenverfahren

1. Implementieren Sie eine Methode

$$[cc] = \text{createCC}(n, \text{ord}, bc) \quad (1.14)$$

welche die $\check{C}C$ -Matrix mit der Stützstellenanzahl n , der Ordnung des Differenzenverfahrens ord ($2 = \text{zweite}$ und $4 = \text{vierte Ordnung}$) und der für beide Ränder identischen Art der Randbedingung bc ($0 = \text{fehlende}$, $1 = \text{elektrische}$ und $2 = \text{magnetische}$) erstellt. Rückgabewert ist die $\check{C}C$ -Matrix cc . Nutzen Sie hierfür das vorgefertigte Template `createCC.m`.

2. Ein einfacher Solver soll in

$$[kx, \text{modes}] = \text{solveCC}(cc, dx) \quad (1.15)$$

implementiert werden, wobei die Schrittweite dx als zusätzlicher Eingabeparameter übergeben wird. kx ist hier ein Vektor mit den geordneten Wellenzahlen, angefangen mit der kleinsten Wellenzahl. In gleicher Reihenfolge sollen auch die Eigenvektoren in der Matrix modes zurückgegeben werden. Das Eigenwertproblem lässt sich durch die MATLAB[®]-Funktion `eig` lösen, das Sortieren kann mit `sort` erfolgen.

3. Verwenden Sie die Routine `createCC` mit $n=6$, $ord=2$ und $bc=0$ und anschließend `solveCC`. Überprüfen Sie die Orthogonalität der Eigenvektoren. Wie viele Eigenmoden können bei dieser Parameterwahl bestimmt werden? Nutzen Sie hierfür die bereits gegebene Datei `check_orth.m`.

Hinweis: Wenn Sie die Eigenvektoren geschickt miteinander multiplizieren, erhalten Sie eine Matrix, in der jeder Eintrag einem Produkt zweier Eigenvektoren entspricht. Diese sich ergebene Matrix lässt sich dann bequem mit dem Befehl `imagesc` (siehe Abschnitt 1.1.3) darstellen.

Mit der Routine `check_orth` und den gegebenen Parametern konnten 6 Eigenmoden bestimmt werden.

4. Stellen Sie die zwei niedrigsten Moden in einem Skript `plotModes` grafisch dar. Verwenden Sie $n=100$, $bc=1$ und $ord=4$ sowie die Länge des eindimensionalen Gebietes $L=5$ m.

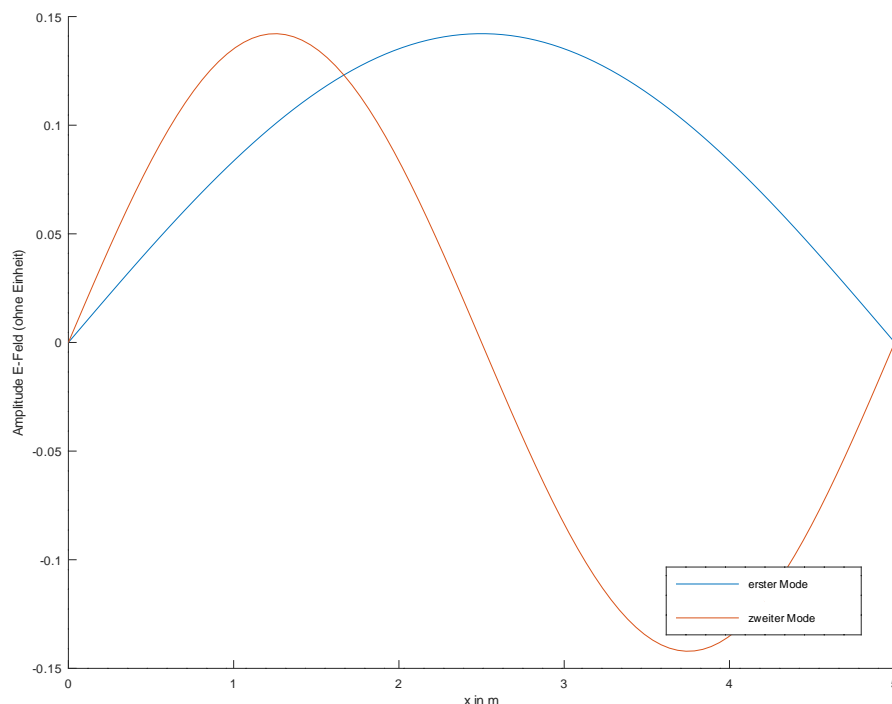


Abbildung 1.2: Plot der zwei Moden mit der geringsten Frequenz.

5. Als nächstes sollen Sie das Konvergenzverhalten betrachten. Schreiben Sie ein Skript `plotConv`, welches das Konvergenzverhalten in Abhängigkeit von der Stützstellenanzahl n Ihrer verschiedenen Implementierungen in zwei Grafiken dokumentiert:

1. Lineare Darstellung der Wellenzahl des Grundmodes über der Stützstellenanzahl n im Fall elektrischer Randbedingungen, sowohl analytisch als auch zweite und vierte Ordnung.
2. Doppelt-logarithmische Darstellung des relativen Wellenzahlfehlers des Grundmodes über die Gitterschrittweite bei elektrischen und fehlenden Randbedingungen für jeweils beide Ordnungen.

Verwenden Sie die Grafiken um die Ordnung der verschiedenen Implementierungen graphisch zu bestimmen. Vergleichen Sie Ihr Ergebnis mit Aufgabe 1.1 aus der Vorbereitung. Wie verändert sich das Konvergenzverhalten, wenn keine Randbedingungen implementiert sind?

In Abbildung 1.3 wird eine Approximation der Wellenzahl an Abhängigkeit der Stützstellenanzahl des Gitters mit verschiedenen Verfahren dargestellt. In Abbildung 1.4 wird der Wellenzahlfehler in Abhängigkeit der Gitterschrittweite gezeigt. Das Konvergenzverhalten wird durch die nicht gegebenen Randbedingungen schlechter. Dies ist auch logisch, da durch die vorgegebenen Werte an den Rändern nicht mehr so viele Möglichkeiten für Lösungen existieren. Dadurch konvergiert ein Verfahren mit Randbedingungen schneller als ohne.

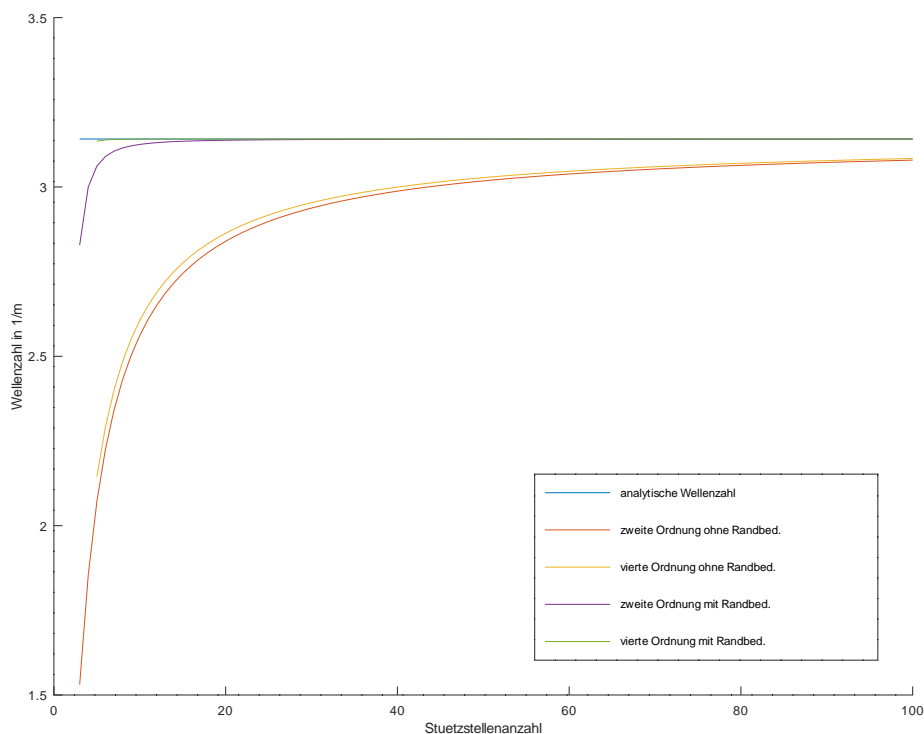


Abbildung 1.3: Approximation der Wellenzahl in Abhängigkeit der Stützstellenanzahl des Gitters.

1.2.2 Dreidimensionale Darstellung

6. Schreiben Sie ein Skript `plotCyl`, welches den Zylinder aus der Vorbereitung visualisiert. Bauen Sie hierzu auf der `MATLAB`[®]-Funktion `patch` auf. Verwenden Sie die Anzahl der Dreiecksflächen in einer Deckelfläche `nd=20`, den Radius `r=1` und die Höhe `h=1`.

Zur Lösung dieser Aufgabe wurde der Pseudocode aus Aufgabe 7 umgesetzt und die dabei gewählten Verfahren implementiert. Der Zylinder ist in Abbildung 1.5 zu sehen.

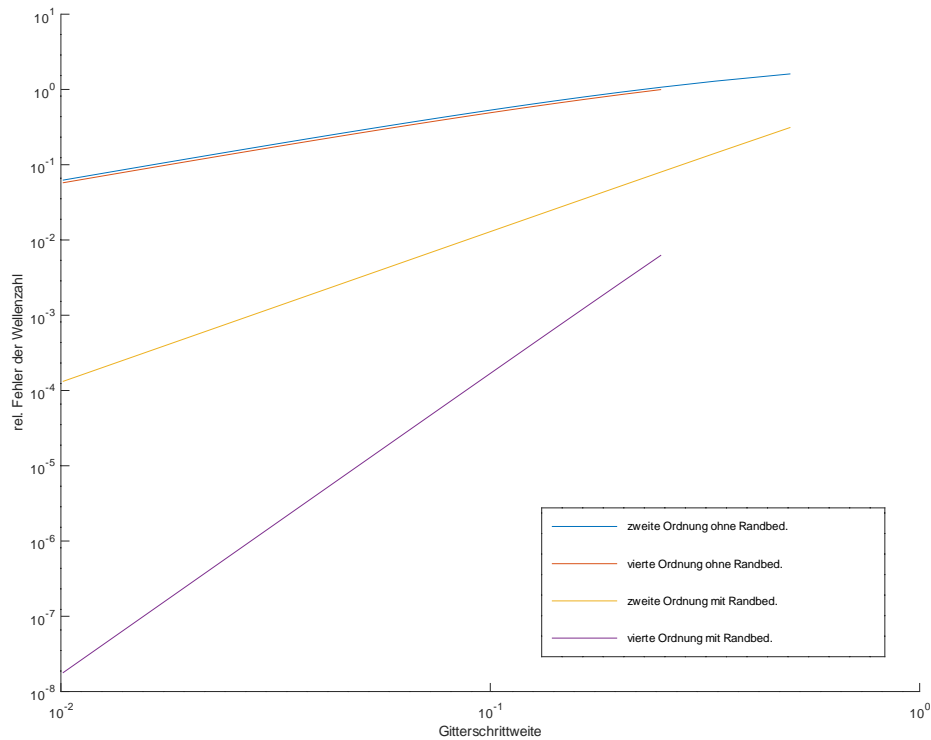


Abbildung 1.4: Wellenzahlfehler in Abhängigkeit der Gitterschrittweite.

7. Stellen Sie für einen Zylinder Ihrer Wahl den Oberflächenfehler ΔA sowie den Volumenfehler ΔV in Abhängigkeit von der Anzahl der Dreiecksflächen je Deckel n_D (Vorbereitungsaufgabe 1.1) in einem Skript `plotVisErr` doppelt-logarithmisch dar. Mit welcher Ordnung konvergieren die Fehler? Aus der Vorbereitung wissen Sie zusätzlich, wie der Speicherbedarf der Darstellung skaliert. Wie viele Dreiecke sind notwendig um einen Diskretisierungsfehler kleiner als 10^{-5} zu garantieren?

In Abbildung 1.6 ist der relative Oberflächen- und Volumenfehler in Abhängigkeit der Dreiecke auf der Deckelfläche dargestellt. Um ein Fehler kleiner von 10^{-5} zu garantieren, benötigt man 812 Dreiecke.

8. Verwenden Sie die bereitgestellte Methode `read_stl` um zwei der bereitgestellten Geometrien im STL-Format (vgl. Abb. 1.4) einzulesen und dann erneut mit `patch` darzustellen. Nennen Sie Ihr Skript `plotStl`.

1.3 Fazit

Wie man in den Aufgaben sehen konnte, kann man durch die Bildung von Matrizen (Diskretisierungen) für die Ableitungsoperatoren die gegebenen Gleichungen einfach numerisch lösen. Auch bei der Diskretisierung von Oberflächen wurde erkannt, dass Matrizen hier ein einfach zu handhabendes Speichermedium bilden. Die Diskretisierung kann man schnell, aber auch relativ genau bestimmen.

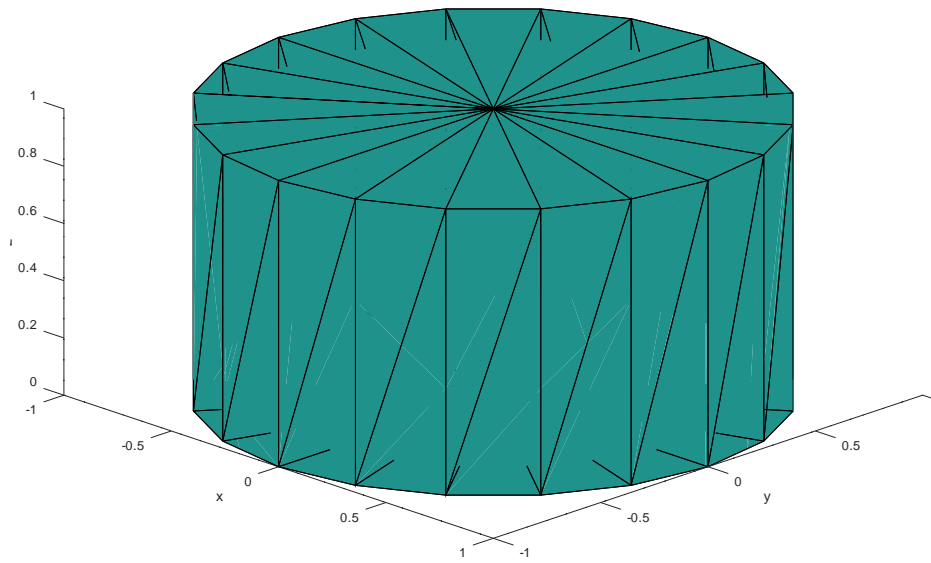


Abbildung 1.5: Nach den Vorgaben geplotteter Zylinder.

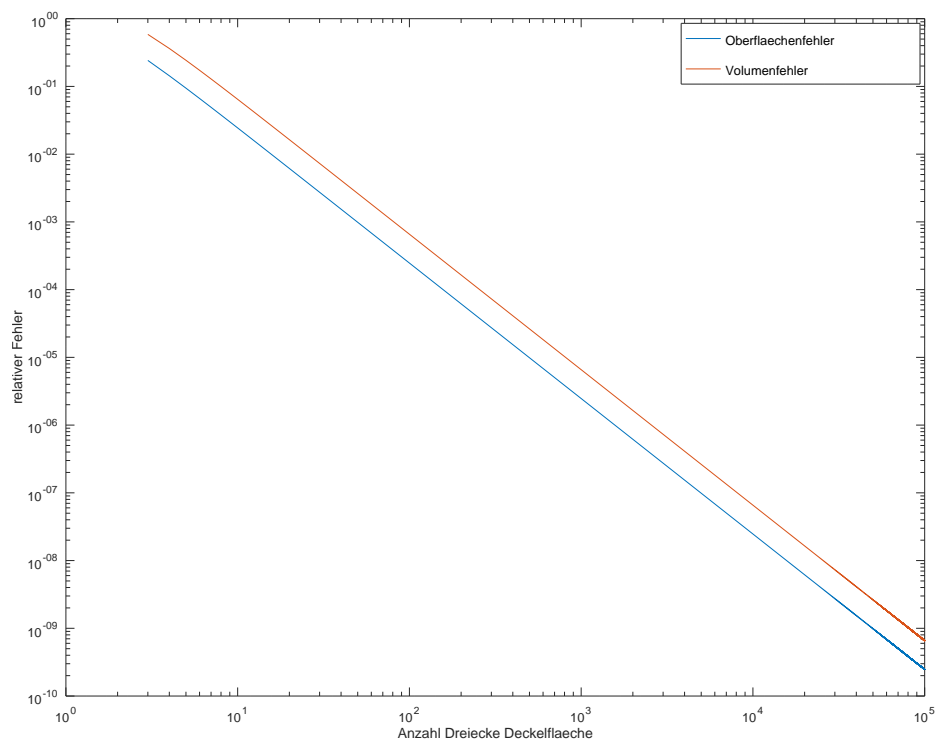


Abbildung 1.6: Relativer Oberflächen- und Volumenfehler in Abhängigkeit der Dreiecke auf der Deckelfläche.

2 Grundlagen der Methode der Finiten Integration 1

2.1 Vorbereitungsaufgaben

2.1.1 Überzählige Kanten

1. Skizzieren Sie ein zweidimensionales kartesisches Gitter mit 3×4 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.

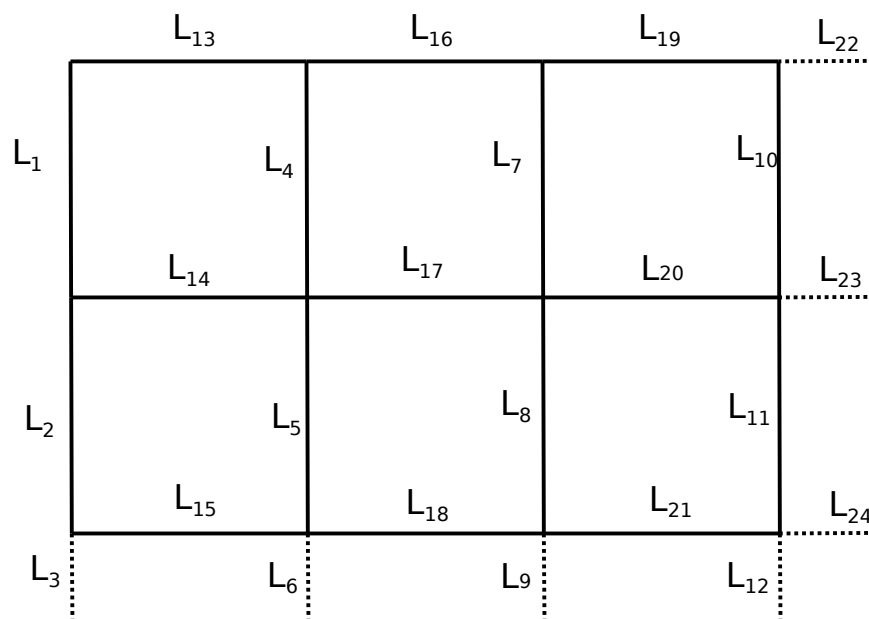


Abbildung 2.1: 3×4 Gitter mit kanonischer Kantenindizierung.

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 $N_p = N_{xy}^2$ 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 $\sqrt{N_p} \cdot 2$. Die Indizes aller Geisterkanten nach dem kanonischen Indizierungsschema lassen sich einfach berechnen. Dazu erst die Geisterkanten in x-Richtung:

$$n \in [N_x, 2N_x, \dots, N_x N_y].$$

und dann in y-Richtung:

$$n \in [N_x(N_y - 1) + 1, N_x(N_y - 1) + 2, \dots, 2N_p].$$

2.1.2 Dreiecksgitter

Gegeben sind die beiden Dreiecksgitter in Abb. 2.2, wobei zunächst nur das linke Gitter betrachtet werden soll.

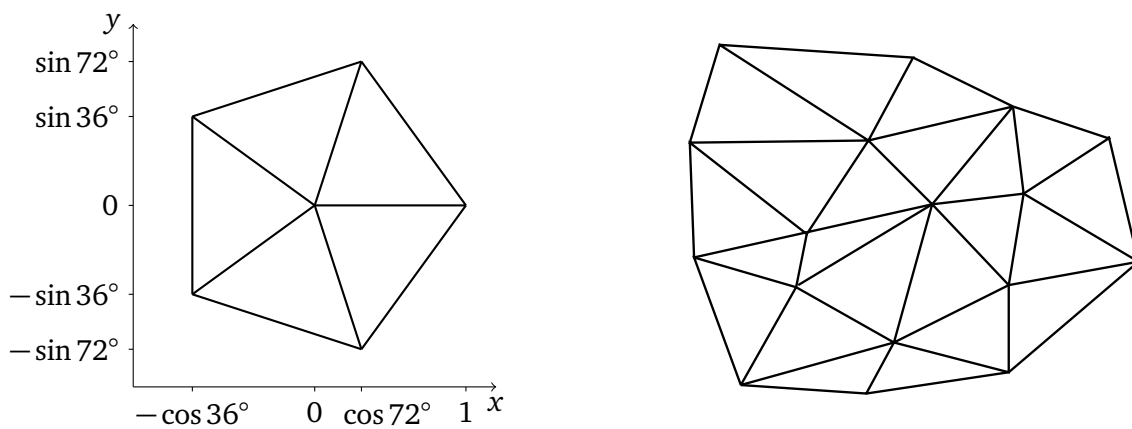


Abbildung 2.2: Dreiecksgitter von zwei verschiedenen Rechengebieten.

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

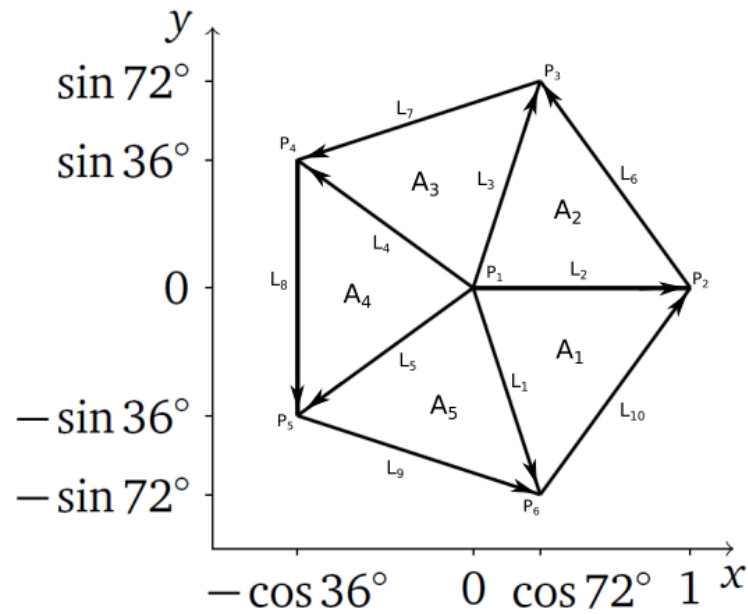


Abbildung 2.3: Gegebenes Gitter mit beliebig gewählter Indizierung für Punkte, Kanten und Flächen.

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.

Mithilfe der Abbildung 2.3 bekommt man folgende Tabellen:

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

Tabelle 2.1: Punkteliste

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 2.2: 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 2.3: Flächen-zu-Kanten Indextabelle

5. Erstellen Sie aus der Kanten-zu-Knoten-Inzidenz die Gradientenmatrix \mathbf{G} . Gehen Sie von einem Potentialvektor φ 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.

Aus der Kanten-zu-Knoten Tabelle 2.2 bekommt man Gradientenmatrix \mathbf{G} :

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

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

Aus der Flächen-zu-Kanten Tabelle 2.3 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 $\text{rot grad} = 0$ auch für die aufgestellten diskreten Matrizen $\mathbf{CG} = \mathbf{0}$ gilt.

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

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

2.1.3 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.

2.1.4 Ab hier sollen beide Gitter aus Abb. 2.2 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.

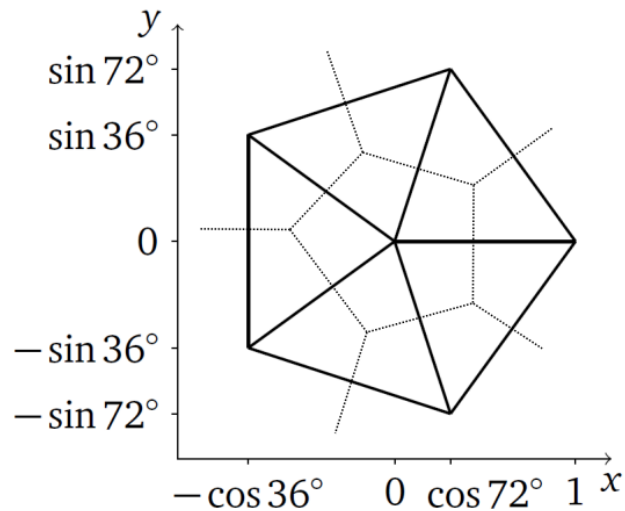


Abbildung 2.4: Duales Gitter für das erste gegebene Gitter.

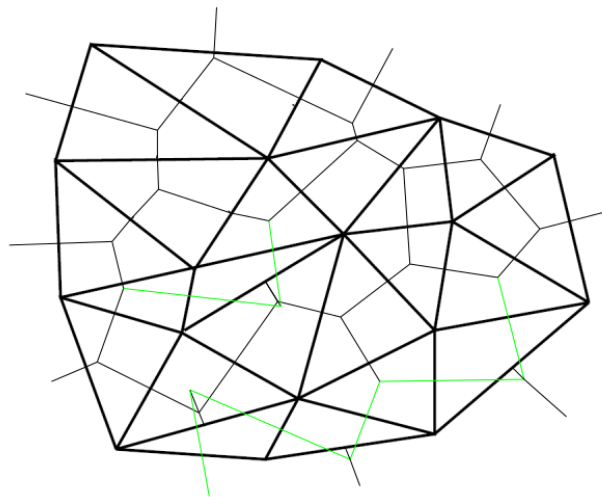


Abbildung 2.5: Duales Gitter für zweites gegebenes Gitter. Kanten, die die Orthogonalität nicht erfüllen, sind grün markiert.

Die dualen Gitter der gegebenen primären Gitter sind in den Abbildungen 2.4 und 2.5 gegeben.

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?

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 . 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.

2.2 Aufgaben während der Praktikumssitzung

2.2.1 Datenstruktur, Visualisierung des Gitters

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

$$[\text{msh}] = \text{cartMesh}(\text{xmesh}, \text{ymesh}, \text{zmesh}) \quad (2.1)$$

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.

2. Implementieren Sie die Methode

$$\text{plotMesh}(\text{msh}), \quad (2.2)$$

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

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.

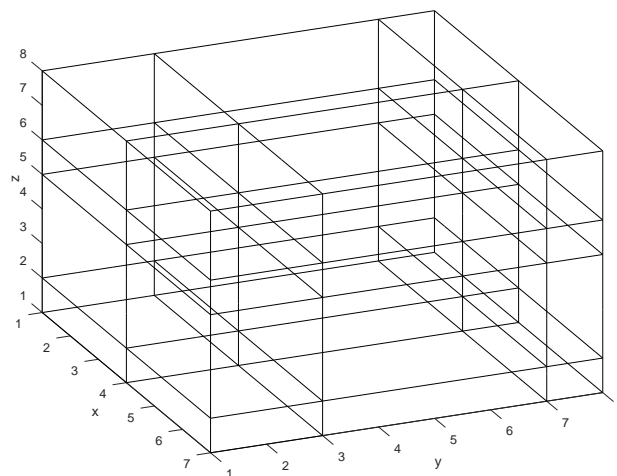


Abbildung 2.6: exampleMesh Plot

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

4. Schreiben Sie eine Methode

$$[c, s, st] = \text{geoMats}(\text{msh}), \quad (2.3)$$

die die Operatormatrizen für ein kanonisches, kartesisches Gitter msh erzeugt. Die Rückgabewerte c , s und st sind die Matrizen \mathbf{C} , \mathbf{S} und $\tilde{\mathbf{S}}$ und entsprechend Abschnitt 2.2.7 definiert. Diese werden mithilfe der \mathbf{P}_ξ -Matrizen erzeugt. Wieso ist es nicht sinnvoll, $\tilde{\mathbf{C}}$ 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 $\tilde{\mathbf{C}}$ ist nicht sinnvoll, da diese durch $\tilde{\mathbf{C}} = \mathbf{C}^\top$ und die Rückgabe von \mathbf{C} bereits einfach erzeugt werden kann. Selbiges gilt für die primäre Gradientenmatrix \mathbf{G} die aus \mathbf{S}^\top durch $\mathbf{G} = -\mathbf{S}^\top$ erzeugt werden kann.

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[®] 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`.

Wie in Abbildung 2.7 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 2.4 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.

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 2.4: Speicherverbrauch einer C-Matrix in unterschiedlichen Speicherformaten

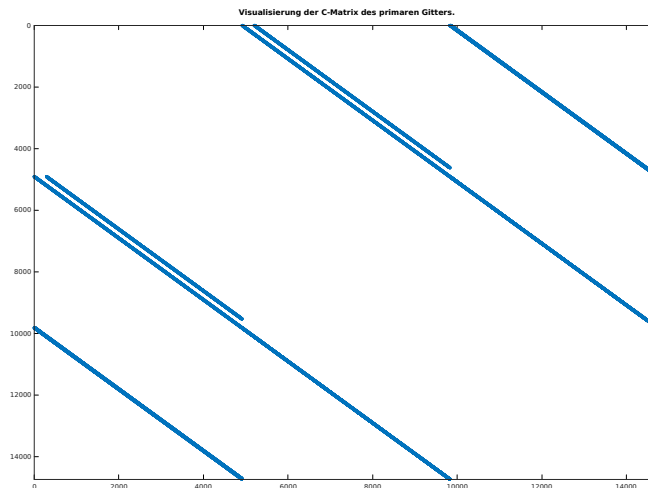


Abbildung 2.7: Speicherverbrauch einer C-Matrix in unterschiedlichen Speicherformaten

6. Berechnen Sie

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

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

$$\begin{aligned}\mathbf{S}\mathbf{C} &= 0 \sim \nabla \cdot \nabla \times = 0 \\ \mathbf{C}(-\tilde{\mathbf{S}}^T) &= \mathbf{C}(\mathbf{G}) = 0 \sim \nabla \times \nabla = 0\end{aligned}$$

einander entsprechen.

2.2.3 Unbelegte Kantenelemente

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

$$\text{edg} = \text{boundEdg}(\text{msh}) \quad (2.4)$$

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 $\text{nz}=1$, 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 2.8 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.

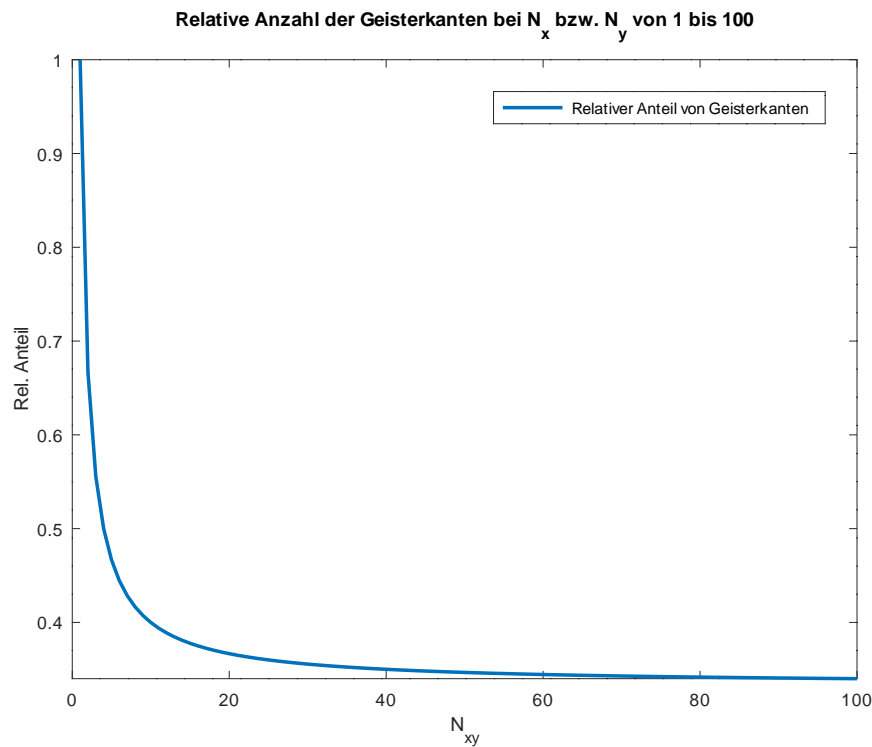


Abbildung 2.8: Anteil der Geisterkanten relativ zur Gesamtanzahl der Kanten.

2.2.4 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:

$$[\text{fieldBow}] = \text{impField}(\text{msh}, \text{field}) \quad (2.5)$$

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

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.

Die vorgegebenen Feldverteilungen sind in den Abbildungen 2.9 und 2.10 dargestellt.

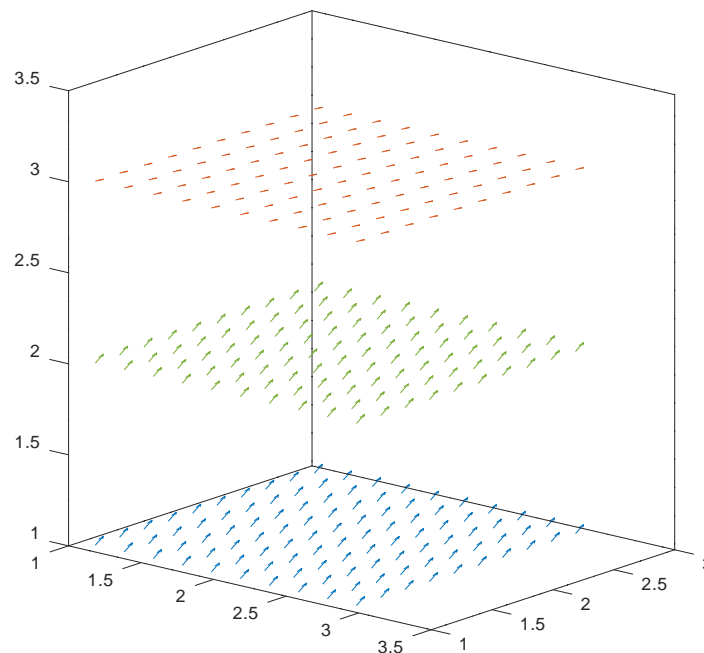


Abbildung 2.9: Feldverteilung aus 10.1

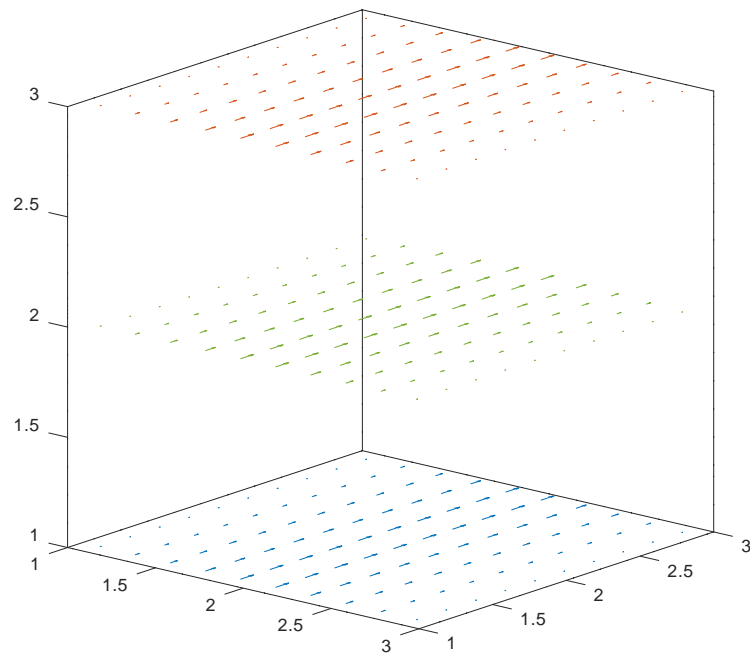


Abbildung 2.10: Feldverteilung aus 10.2

2.3 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 zweidimensionales kartesisches Gitter sowie für das Dreiecksgitter aus Bild 2.2 a). Ist diese Wahl des dualen Gitters am Rand immer notwendig?

Das duale Gitter sollte am Rand so gewählt werden, dass der Teil der Kante, der aus dem Rechengebiet herausragt, genauso lang ist, wie der Teil der noch innerhalb des Rechengebietes liegt. Die dualen Gitter für die in der Aufgabe geforderten primären Gitter sind in den Abbildungen 2.11 und 2.12 zu sehen.

2.4 Fazit

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

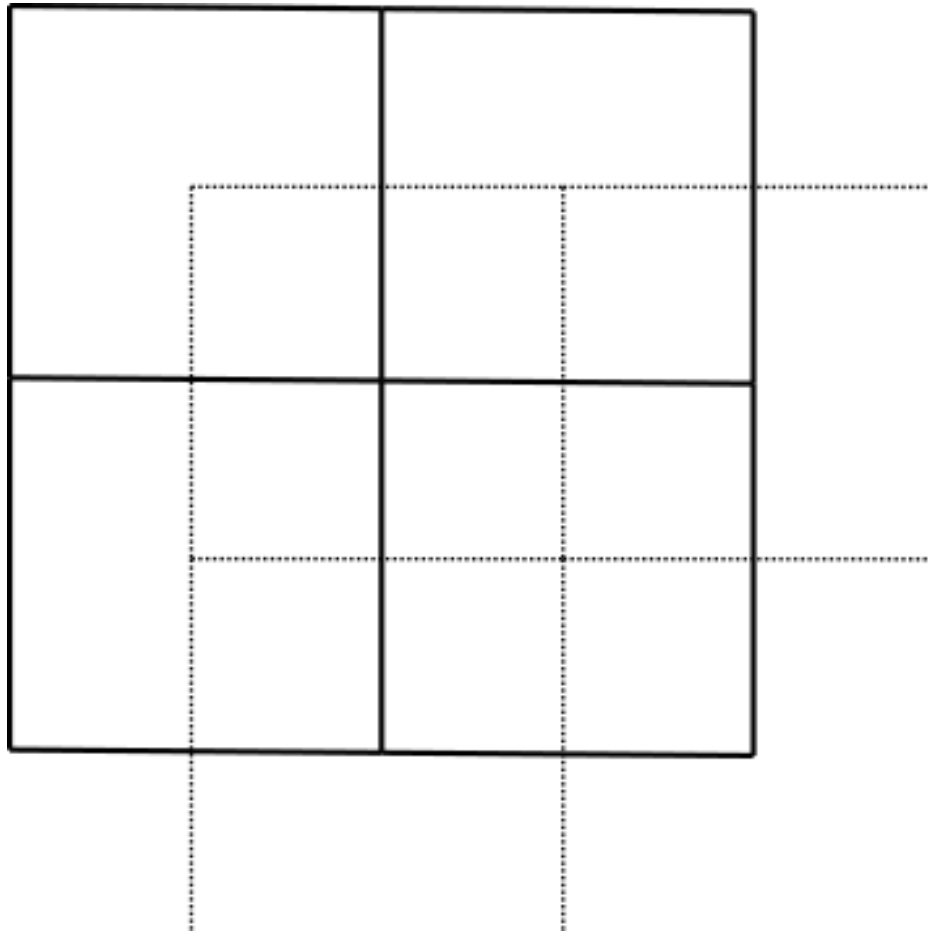


Abbildung 2.11: Primäres und duales Gitter für Erfüllung der magnetischen Randbedingung.

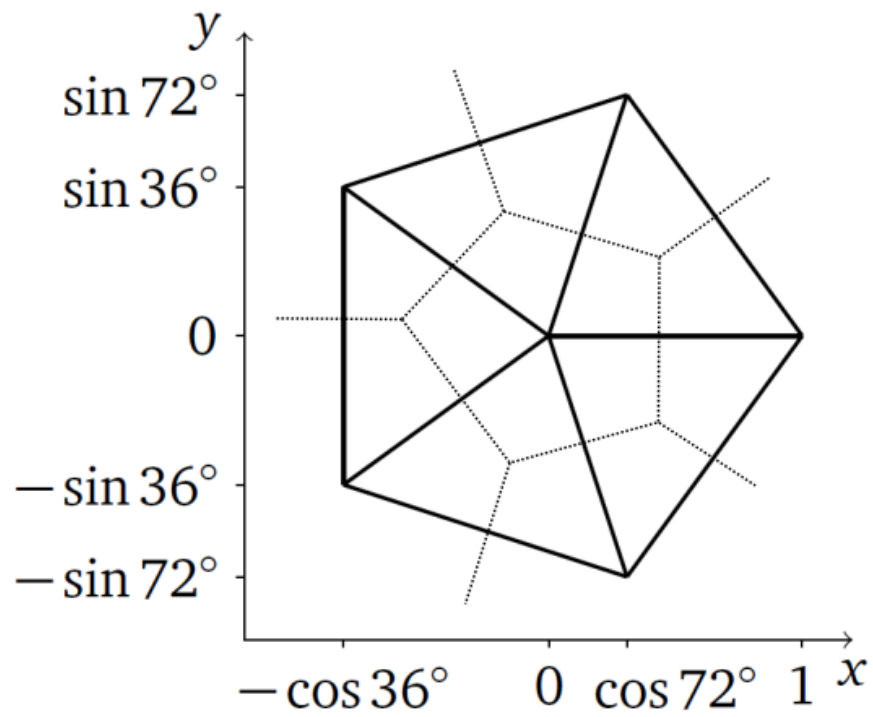


Abbildung 2.12: Primäres und duales Gitter für Erfüllung der magnetischen Randbedingung.

3 Grundlagen der Methode der Finiten Integration 2

3.1 Vorbereitungsaufgaben

1. Überlegen Sie sich, wie man ausgehend vom 3-fach Index i, j, k (vgl. Gl. (3.1)) die Randpunkte eines kartesischen Rechengebietes im kanonischen Indizierungsschema bestimmt (eine Skizze ist hilfreich). Schreiben Sie hierfür ein Schleifenkonstrukt in Pseudocode.

Ausgehend von einem 3-fach Index i, j, k lassen sich die Randpunkte eines Kartesischen Rechengitters im kanonischen Indizierungsschema durch ablaufen der Außenseiten bestimmen. Hierbei werden nacheinander i, j und k zu Null gesetzt und die beiden anderen Variablen variiert. Anschließend wird selbiges mit einem Festsetzen beim Maximalwert der jeweiligen Variable wiederholt. Alle hierbei erreichten Punkte liegen auf den Außenseiten des Gitters.

Im Pseudocode lässt sich diese Verfahren für eine Richtung i darstellen durch

```
1:  $c \leftarrow 1$ 
2:  $d \leftarrow 1$ 
3: while  $c \leq j_{\max}$  do
4:   while  $d \leq k_{\max}$  do
5:     AddToListBoundaryElementAt(1,  $c, d$ )
6:      $d \leftarrow d + 1$ 
7:   end while
8:    $c \leftarrow c + 1$ 
9: end while
10:  $c \leftarrow 1$ 
11:  $d \leftarrow 1$ 
12: while  $c \leq j_{\max}$  do
13:   while  $d \leq k_{\max}$  do
14:     AddToListBoundaryElementAt( $i_{\max}, c, d$ )
15:      $d \leftarrow d + 1$ 
16:   end while
17:    $c \leftarrow c + 1$ 
18: end while
```

Die Funktion *AddToListBoundaryElementAt* führt eine Liste mit den kanonischen Indizes, welche unter Zuhilfenahme von M_x , M_y und M_z bestimmt werden, der Randpunkte.

2. Wie sehen für ein äquidistantes, kartesisches Gitter die Geometriematrizen D_S , \tilde{D}_S , D_A und \tilde{D}_A aus? Was ist bei den Rändern zu beachten? Welche Dimensionen besitzen die Matrizen?

Für eine äquidistantes, kartesisches Gitter bildet die Matrix D_S eine Diagonalmatrizen mit dem Abstand Δx auf der Diagonalen. Die duale Matrix \tilde{D}_S unterscheidet sich hiervon nur dadurch, dass die Randelemente mit $\frac{1}{2}$ multipliziert werden.

Für die flächenzentrierten D_A gilt simultan, dass sie Diagonalmatrizen mit den Flächeninhalt Δx^2 sind. Bei der Dualen Flächenmatrix \tilde{D}_A muss nun bei den Randelementen allerdings unterschieden werden

zwischen denen die an einer Kante liegen und mit $\frac{1}{2}$ multipliziert werden und denen an einer Ecke die mit $\frac{1}{4}$ multipliziert werden.

Falls hier bereits Geisterkanten entfernt worden sind, sind bei den primären Matrizen D_S und D_A die Geisterelemente jeweils Null.

Die Matrizen besitzen immer die Dimension $[np, 3np]$ mit np als Anzahl der Gitterpunkte.

3. Skizzieren Sie kurz, wie sich die Materialmatrizen zusammenstellen. Wie sind hierbei die Randbedingungen (elektrisch & magnetisch) einzuarbeiten bzw. muss überhaupt eine Änderung vorgenommen werden?

Fügen Sie hier Ihre Lösung ein

4. Um die im Versuch zu implementierende Visualisierung zu testen, soll ein vorgegebenes rotationssymmetrisches Feld in Zylinderkoordinaten nach der analytischen Formel

$$\vec{D}(r, \varphi, z) = \frac{1}{r^2} \vec{e}_r \quad (3.1)$$

visualisiert werden. Es soll ein äquidistantes Gitter benutzt werden, dessen Mitte genau dem Koordinatenursprung entspricht.

Bestimmen Sie die diskreten Größen $\hat{d}(n)$ und $\hat{e}(n)$ des vorgegebenen Feldes. Zur Vereinfachung soll bei der hierfür notwendigen Integration der Feldwert in der Mitte der Strecke bzw. Fläche als repräsentativ gelten und damit als konstant über dem gesamten Element angenommen werden.

Hinweis: Transformieren Sie zuerst zur Bestimmung der notwendigen Feldwerte das gegebene Feld in kartesische Koordinaten $\vec{D}(x, y, z)$.

Fügen Sie hier Ihre Lösung ein

3.2 Aufgaben während der Praktikumssitzung

3.2.1 Materialmatrizen

1. Zuerst sollen zwei Funktionen zum Bestimmen der Geometriematrizen D_S , \tilde{D}_S und D_A geschrieben werden:

$$[DS, DSt] = \text{createDS}(msh) \quad (3.2)$$

$$[DA] = \text{createDA}(DS) \quad (3.3)$$

Wie kann mit der zweiten Funktion auch \tilde{D}_A bestimmt werden?

Fügen Sie hier Ihre Lösung ein

2. Nun sollen die Funktionen

$$[\text{Deps}] = \text{createDeps}(\text{msh}, \text{DA}, \text{DA}t, \text{eps_r}, \text{bc}) \quad (3.4)$$

$$[\text{Meps}] = \text{createMeps}(\text{DA}t, \text{Deps}, \text{DS}) \quad (3.5)$$

geschrieben werden, um die \mathbf{M}_ε -Matrix Meps aus der \mathbf{D}_ε -Matrix Deps der gemittelten Permittivitäten zu bestimmen. $\text{bc} = 1$ soll dabei elektrische und $\text{bc} = 2$ magnetische Randbedingungen bedeuten. Die Materialverteilung auf dem Gitter msh soll inhomogen und isotrop bezüglich der Raumrichtungen sein. Zur besseren Übersicht sollen bei der Übergabe relative Permittivitäten verwendet werden. eps_r soll damit als $N_p \times 1$ Matrix übergeben werden, also für jedes der N_p primären Volumen ein ε_r -Wert.

Hinweis: Für das Invertieren von \mathbf{D}_S ist die Methode nullInv vorgegeben.

Fügen Sie hier Ihre Lösung ein

3. Die Funktion (3.5) soll nun mit den Parametern $\text{xmesh} = [-2 \ 0 \ 2]$, $\text{ymesh} = [-1 \ 0 \ 1]$, $\text{zmesh} = [0 \ 1]$ und isotropem $\varepsilon = \varepsilon_0$ die Materialmatrix \mathbf{M}_ε für elektrische Randbedingungen berechnen und ausgeben. Vervollständigen Sie hierfür das bereits gegebene Skript exampleMeps.m

Fügen Sie hier Ihre Lösung ein

3.2.2 Interpolation und Visualisierung

4. Programmieren Sie eine Routine

$$\text{eField} = \text{fitInt}(\text{msh}, \text{eBow}), \quad (3.6)$$

die die Komponenten von $\hat{\mathbf{e}}$ als \vec{E} -Feld auf die primären Punkte interpoliert.

Fügen Sie hier Ihre Lösung ein

5. Schreiben sie eine Methode

$$\text{plotEBow}(\text{msh}, \text{eBow}, \text{indz}), \quad (3.7)$$

die auf Methode (3.6) aufbauend $\hat{\mathbf{e}}$ interpoliert und den Betrag des \vec{E} -Feldes mit dem MATLAB[®]-Befehl surf in einer x - y -Ebene mit Index indz grafisch darstellt. Verwenden Sie hierfür bitte elektrische Randbedingungen.

Hinweis: Nutzen Sie auch für das Invertieren von \mathbf{M}_ε die vorgegebene Methode nullInv.

Fügen Sie hier Ihre Lösung ein

6. Geben Sie das rotationssymmetrische Feld aus der Vorbereitung als Vektor $\hat{\mathbf{d}}$ vor, berechnen Sie daraus mit Hilfe der Materialmatrix $\mathbf{M}_\varepsilon^{-1}$ das Feld $\hat{\mathbf{e}}$ und wenden Sie dann Methode (3.7) an. Visualisieren Sie außerdem die selbe Schnittebene mit der in Versuch 2 vorgestellten Methode `plotEdgeVoltage`. Vervollständigen Sie hierfür den ersten Teil des bereits gegebenen Skripts `exampleVisualEfield.m`

Fügen Sie hier Ihre Lösung ein

7. Überlegen Sie sich, welche Änderungen an den bisher implementierten Methoden vorgenommen werden müssen, um ein anisotropes Material zu verwenden. Ändern Sie Ihre Implementierung entsprechend und verwenden Sie ein anisotropes Material mit unterschiedlichen Permittivitäten in x - und y -Richtung (z. B. $\varepsilon_x/\varepsilon_y = 4$) sowie elektrische Randbedingungen. Interpolieren und visualisieren Sie das Feld $\hat{\mathbf{e}}$ wie in der Aufgabe zuvor. Visualisieren Sie auch hier das Ergebnis zusätzlich mit der Methode `plotEdgeVoltage`. Vervollständigen Sie hierfür den zweiten Teil des bereits gegebenen Skripts `exampleVisualEfield.m`

Fügen Sie hier Ihre Lösung ein

3.3 Fragen zur Ausarbeitung

1. Erstellen Sie eine 2D-Skizze einer dualen Gitterfläche mit den zugehörigen primären Gitterzellen, welche zur Mittelung der Permittivität notwendig sind (siehe (3.10)).

Fügen Sie hier Ihre Lösung ein

2. Häufig werden für die Visualisierung der magnetischen Feldstärke \vec{H} die entsprechenden Komponenten ebenfalls auf den Punkten des primären Gitters gemittelt und nicht auf den dualen Punkten. Beschreiben Sie für diese Mittelung *kurz* eine geeignete Vorgehensweise (kleine Skizze sinnvoll) und gehen Sie dabei auch auf die Randbedingungen ein.

Fügen Sie hier Ihre Lösung ein

3.4 Fazit

Fügen Sie hier Ihre Lösung ein

Abbildungsverzeichnis

1.1	Diskretisierung einer Kreiszylinderoberfläche. Dargestellt sind die Dreiecksgitter der Diskretisierung der Deckelfläche (links) und des diskretisierten Zylinders in Perspektivansicht (rechts).	7
1.2	Plot der zwei Moden mit der geringsten Frequenz.	11
1.3	Approximation der Wellenzahl in Abhängigkeit der Stützstellenanzahl des Gitters.	12
1.4	Wellenzahlfehler in Abhängigkeit der Gitterschrittweite.	13
1.5	Nach den Vorgaben geplotteter Zylinder.	14
1.6	Relativer Oberflächen- und Volumenfehler in Abhängigkeit der Dreiecke auf der Deckelfläche.	14
2.1	3×4 Gitter mit kanonischer Kantenindizierung.	15
2.2	Dreiecksgitter von zwei verschiedenen Rechengebieten.	16
2.3	Gegebenes Gitter mit beliebig gewählter Indizierung für Punkte, Kanten und Flächen. . . .	17
2.4	Duales Gitter für das erste gegebene Gitter.	20
2.5	Duales Gitter für zweites gegebenes Gitter. Kanten, die die Orthogonalität nicht erfüllen, sind grün markiert.	20
2.6	exampleMesh Plot	21
2.7	Speicherverbrauch einer C-Matrix in unterschiedlichen Speicherformaten	23
2.8	Anteil der Geisterkanten relativ zur Gesamtanzahl der Kanten.	24
2.9	Feldverteilung aus 10.1	25
2.10	Feldverteilung aus 10.2	26
2.11	Primäres und duales Gitter für Erfüllung der magnetischen Randbedingung.	27
2.12	Primäres und duales Gitter für Erfüllung der magnetischen Randbedingung.	27