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

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

setzt sich zusammen aus dem Vorwärts-Differenzenquotienten

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

und dem Rückwärts-Differenzenquotienten

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

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

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

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 wird ein lineares Gleichungssystem

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

mit $\Delta x = h$ aufgestellt.
Hieraus folgt somit

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

und

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

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

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

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

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

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

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

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 einsetzten. Dazu werden die Funktionswerte in einen Vektor **e** geschrieben. Die Koeffizienten der Funktionswerte aus (1.9) werden in der Matrix $\widetilde{\mathbf{CC}}$ gesammelt, sodass sich folgende Diskretisierung des Problems ergibt:

$$\frac{1}{12}\widetilde{\mathbf{CC}}\mathbf{e} = -\Delta x^2 k_x^2 \mathbf{e} \tag{1.10}$$

mit

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

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

Die Matrix $\widetilde{\mathbf{CC}}$ 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:

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

$$\begin{aligned} -\frac{1}{12}f(x_{-1}) + \frac{16}{12}f(x_0) - \frac{30}{12}f(x_1) + \frac{16}{12}f(x_2) - \frac{1}{12}f(x_3) &= -\Delta x^2 k_x^2 f(x_1) \\ \frac{16}{12}f(x_0) - \frac{29}{12}f(x_1) + \frac{16}{12}f(x_2) - \frac{1}{12}f(x_3) &\overset{ung.Symm.}{=} -\Delta x^2 k_x^2 f(x_1) \end{aligned}$$

Bei magnetischen Randbedingungen gilt dagegen:

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

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

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

$$\widetilde{\mathbf{CC}}_{elek} = \begin{pmatrix} -30 & 0 & 0 & 0 & 0 & 0 & \dots \\ 16 & -29 & 16 & -1 & 0 & 0 & \dots \\ -1 & 16 & -30 & 16 & -1 & 0 & \dots \\ 0 & -1 & 16 & -30 & 16 & -1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & -1 & 16 & -29 & 16 \\ \vdots & \vdots & \vdots & 0 & 0 & 0 & -30 \end{pmatrix}$$

$$\widetilde{\mathbf{CC}}_{magn} = \begin{pmatrix} -30 & 32 & -2 & 0 & 0 & 0 & \dots \\ 16 & -31 & 16 & -1 & 0 & 0 & \dots \\ -1 & 16 & -30 & 16 & -1 & 0 & \dots \\ 0 & -1 & 16 & -30 & 16 & -1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \vdots & \ddots & -1 & 16 & -31 & 16 \\ \vdots & \vdots & \vdots & 0 & -2 & 32 & -30 \end{pmatrix}$$

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

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

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 Differenzenquotieten 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

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

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

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

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 $c_2 = 0$ wäre die triviale Nulllösung erreicht. Die kleinste Wellenzahl ist damit $k_x = 0$.

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 die 1. Maxwell'sche Gleichung im Frequenzbereich $\mathrm{rot}\underline{E}_y = -j\omega\mu\underline{H}_x$ verwenden. Da die Rotation im eindimensionalen nur eine einfach Ableitung beschreibt, kann man $\frac{\mathrm{d}\underline{E}_y}{\mathrm{d}x} \propto \underline{H}_x$ schreiben. Damit folgt aus (II)

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

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 triviale Lösung. Die kleinste Wellenzahl ist damit $k_X = 0$.

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 das alle Moden voneinander unabhängige Lösungen sind.

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 $2a_\mathrm{D}$ und Höhe h_D.

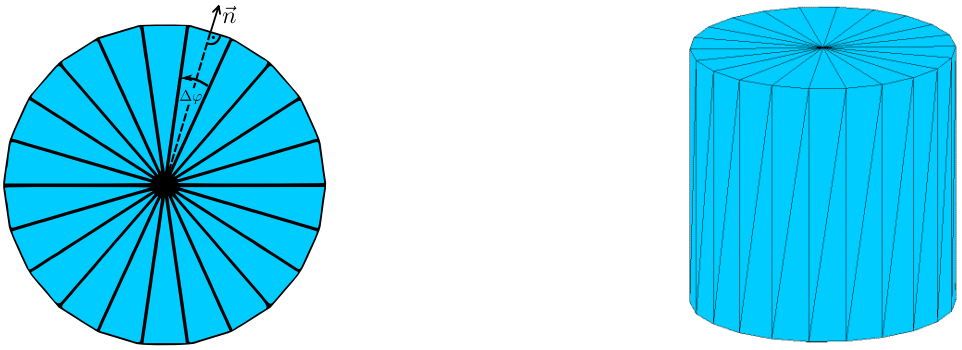

Abbildung 1.1: Diskretisierung einer Kreiszylinderoberfläche. Dargestellt sind die Dreiecksgitter der Diskretisierung der Deckelfläche (links) und des diskretisierten Zylinders in Perspektivansicht (rechts).

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“.

Fügen Sie hier Ihre Lösung ein

8. In der Theorie wurde erwähnt, dass eine feinere Diskretisierung die Genauigkeit der Darstellung steigert[1], 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.

In einer STL Format ist jeder Dreieck mit 3 Punkte und Normalenvektor beschrieben. Jeder Punkt und Normalenvektor hat 3 Koordinaten. Dazu hat Zylinder 4 n_D Dreiecke.
Damit kann man benötigte Speicherplatz bestimmen:

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

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

$$A_{\mathrm{D}} = 2\left[\frac{1}{2}a_{\mathrm{D}}h_{\mathrm{D}}\right] = 2\left[\frac{1}{2}\cdot r\cos\left(\frac{\Delta\varphi}{2}\right)\cdot r\sin\left(\frac{\Delta\varphi}{2}\right)\right] = \frac{1}{2}r^2\sin(\Delta\varphi) \tag{1.13}$$

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 gleichschenkliges Dreieck zu bestimmen, soll man es mithilfe der Höhe h_{D} auf zwei rechtwinklige Dreiecke teilen. Fläche der rechtwinkliger Dreieck rechnet man als Multiplikation zweier Katheten geteilt durch 2. Damit bekommt man die Formel:

$$A_{\mathrm{D}} = 2\left[\frac{1}{2}a_{\mathrm{D}}h_{\mathrm{D}}\right]$$

Mit der Pythagoras Formel rechnet man h_{D} und a_{D} aus:

$$\begin{aligned} a_{\mathrm{D}} &= r\sin\left(\frac{\Delta\varphi}{2}\right) \\ h_{\mathrm{D}} &= r\cos\left(\frac{\Delta\varphi}{2}\right) \end{aligned}$$

Jetzt setzt man alles zusammen:

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

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:

$$A_{\mathrm{D}} = \frac{1}{2}r^2\sin(\Delta\varphi)$$

Wenn man relativen Oberflächen- und Volumenfehler bestimmen wollte, soll man Oberfläche und Volumen des wahres und diskretisiertes Zylinder ausrechnen.

$$\begin{aligned} A_{\mathrm{disc}} &= 2n_{\mathrm{D}}r\left[\frac{1}{2}\cdot r\sin\left(\frac{2\pi}{n_{\mathrm{D}}}\right) + \sin\left(\frac{\pi}{n_{\mathrm{D}}}\right)h\right] \\ V_{\mathrm{disc}} &= \frac{1}{2}n_{\mathrm{D}}r^2\sin\left(\frac{2\pi}{n_{\mathrm{D}}}\right)h \\ A &= 2r\pi(r+h) \\ V &= r^2\pi h \end{aligned}$$

Formel für absolute Fehler lautet:

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

Damit bekommt man:

$$\begin{aligned} \Delta A &= 2r\pi(r+h) - 2n_{\text{D}} r \left[\frac{1}{2} \cdot r \sin\left(\frac{2\pi}{n_{\text{D}}}\right) + \sin\left(\frac{\pi}{n_{\text{D}}}\right) h \right] \\ \Delta V &= r^2 \pi h - \frac{1}{2} n_{\text{D}} r^2 \sin\left(\frac{2\pi}{n_{\text{D}}}\right) h \end{aligned}$$

relative Fehler bekommt man mit der Formel:

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

Damit bekommt man:

$$\begin{aligned} \varepsilon_A &= \frac{\pi(r+h) - n_{\text{D}} \left[\frac{1}{2} \cdot r \sin\left(\frac{2\pi}{n_{\text{D}}}\right) + \sin\left(\frac{\pi}{n_{\text{D}}}\right) h \right]}{\pi(r+h)} \\ \varepsilon_V &= \frac{\pi - \frac{1}{2} n_{\text{D}} \sin\left(\frac{2\pi}{n_{\text{D}}}\right)}{\pi} \end{aligned}$$

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 $\Delta\varphi \leq 5°$ gefordert. Berechnen Sie die minimale Anzahl an Dreiecken zur Erfüllung dieser Forderung.

Dieser Forderung rechnet man mit schon bekannten Formel aus 1.1:

$$\Delta\varphi = \frac{360°}{n_{\text{D}}}$$

Realisierung der Forderung:

$$\begin{aligned} 5° &\geq \frac{360°}{n_{\text{D}}} \\ n_{\text{D}} &\geq 72 \end{aligned}$$

Es ist minimal 72 Dreiecke benötigt.

1.2 Aufgaben während der Praktikumssitzung

1.2.1 Differenzenverfahren

1. Implementieren Sie eine Methode

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

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

```
%Methode zur Erstellung der 1D Curl-Curl-Matrix eines einfachen
%Eigenwertproblems einer eindimensionalen Wellengleichung (TE-Mode).
%
%   Eingabe
%   n       Stuetzstellenanzahl (n>=3 fuer ord=2, n>=5 fuer ord=4)
%   ord     Ordnung des Verfahrens (ord=2 zweite oder ord=4 vierte Ordnung)
%   bc      Randbedingungen (bc=0 keine, bc=1 elektrisch, bc=2 magnetisch)
%
%   Rueckgabe
%   cc      1D Curl-Curl-Matrix

function [cc]=createCC(n, ord, bc)

    % Aufstellen der Matrix
    cc = sparse(n,n);

    % Eintraege eintragen
    if ord==2
        % Bestimmen der Eintraege fuer Ordnung 2 ohne Randbedingungen
        for k=1:1:n
          if (k==1)
            cc(k,k)=-2;
            cc(k,k+1)=1;
          elseif (k==n)
            cc(k,k-1)=1;
            cc(k,k)=-2;
          else
            cc(k,k-1)=1;
            cc(k,k)=-2;
            cc(k,k+1)=1;
           endif
        endfor
        if bc==1
            % Aenderung der Matrix bei elektrischem Rand
            cc(1,2)=0;
            cc(n,n-1)=0;
        elseif bc == 2
            cc(1,2)=2;
            cc(n,n-1)=2;
```

```
40          % Aenderung der Matrix bei magnetischem Rand
41      elseif bc ~= 0
42          error('bc kann nur die Werte 0 (elektrisch) oder 1 (magnetisch)
43          annehmen.')
44      end
45      elseif ord==4
46      % Bestimmen der cc Matrix fuer Ordnung 4 ohne Randbedingungen
47      for k=1:1:n
48        if (k==1)
49          cc(k,k)=-30;
50          cc(k,k+1)=16;
51          cc(k,k+2)=-1;
52        elseif (k==2)
53          cc(k,k-1)=16;
54          cc(k,k)=-30;
55          cc(k,k+1)=16;
56          cc(k,k+2)=-1;
57        elseif (k==n)
58          cc(k,k-2)=-1;
59          cc(k,k-1)=16;
60          cc(k,k)=-30;
61        elseif (k==n-1)
62          cc(k,k-2)=-1;
63          cc(k,k-1)=16;
64          cc(k,k)=-30;
65          cc(k,k+1)=16;
66        else
67          cc(k,k-2)=-1;
68          cc(k,k-1)=16;
69          cc(k,k)=-30;
70          cc(k,k+1)=16;
71          cc(k,k+2)=-1;
72         endif
73      endfor
74
75      if bc==1
76          % Aenderung der Matrix bei elektrischem Rand
77          cc(1,2)=0;
78          cc(1,3)=0;
79          cc(2,2)=-29;
80          cc(n,n-1)=0;
81          cc(n,n-2)=0;
82          cc(n-1,n-1)=-29;
83      elseif bc==2
84          % Aenderung der Matrix bei magnetischem Rand
85          cc(1,2)=32;
86          cc(1,3)=-2;
87          cc(2,2)=-31;
88          cc(n,n-1)=32;
89          cc(n,n-2)=-2;
```

```
90              cc(n-1,n-1)=-31;
91          elseif bc~=0
92              error('bc kann nur die Werte 0 (elektrisch) oder 1 (magnetisch) annehm
93          end
94          cc=1/12.*cc;
95          else
96                  error('Ordnung %d ist noch nicht implementiert.', n)
97      end
98
99  end
```

2. Ein einfacher Solver soll in

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

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.

```
1  %Loest das Eigenwertproblem einer Matrix und gibt Wellenzahlen,
2  %berechnet aus Eigenwerten und Gitterschrittweite, und
3  %Eigenvektoren geordnet aus. Bereinigt _nicht_ von unphysikalischen Moden.
4  %
5  %    Eingabe
6  %    cc            Curl-Curl-Matrix
7  %    dx            Gitterschrittwete
8  %
9  %    Rueckgabe
10 %    kx            Wellenzahlen (aufsteigend)
11 %    modes         Eigenmoden (geordnet entsprechend kx)
12
13 function [kx ,modes]=solveCC(cc, dx)
14
15     % Bestimmen der Eigenwerte und -vektoren mit eig
16     [eigenvectors,eigenvalues] = eigs(cc,rows(cc));
17
18
19     % Bestimmen der Wellenzahlen aus den Eigenwerten
20     for i=1:1:rows(cc)
21       k(i)=(sqrt(-eigenvalues(i,i)/dx));
22     endfor
23
24     % Sortieren der Wellenzahlen. Sortierindex mit zurueckgeben lassen !
25     [kx,m]=sort(k);
26     % Sortieren der Eigenvektoren mithilfe des Sortierindexes und damit
27     eigenvectors_sort=sort(eigenvectors',m);
```

```
28      % Bestimmung der Moden.
29      modes_dummy=eigenvectors_sort';
30       for l=1:rows(cc)
31        modes_dummy(:,rows(cc)-l+1)=eigenvectors_sort'(:,l);
32      endfor
33      modes=modes_dummy;
34
35  end
```

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.

```
1  %Skript zur Ueberpruefung der Orthogonalitaet
2
3  % Setzen der parameter n, ord, bc, L
4  n=5;
5  ord=4;
6  bc=2;
7  L=1;
8
9  % Erstellen der CC matrix
10    cc=createCC(n,ord,bc);
11
12 % Gitterschrittweite bestimmen
13    dx=L/(n-1);
14
15 % Loesen der Eigenwertgleichung mit solveCC
16    [kx,modes] = solveCC(cc,dx);
17
18 % Ueberpruefung der Orthogonalitaet der Eigenvektoren
19    orthogonal=modes*modes';
20    imagesc(orthogonal);
```

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.

```
1  %Skript zur Darstellung der ersten beiden Eigenmoden. Verwendet createCC
2  %(Erstellung der Systemmatrix) und solveCC (Loesen des Eigenwertproblems).
3
4  % setzen der parameter n, ord, bc, L
5  n=100;
6  ord=4;
```

```
bc=0;
L=5;

% Erstellen der CC matrix
  cc=createCC(n, ord, bc);

% Gitterschrittweite bestimmen
  dx=L/(n-1);
% Loesen der Eigenwertgleichung mit solveCC
  [kx, modes] = solveCC(cc, dx);

% Sonderbetrachtung fuer magnetische Randbedingungen
if bc==2
    % Loesche ersten Mode, denn er ist nur die statische Loesung (konstant)
    modes=modes(:,2);
end

% x-Koordinaten fuer jede Stuetzstelle in einen Vektor schreiben
  for a=1:1:n
    xKoord(a) = -dx+a*dx;
  endfor
% Grundmode und 2.Mode plotten
figure(1)
hold all
plot(xKoord,modes(:,1));
plot(xKoord,modes(:,2));

hold off
legend({'erster Mode','zweiter Mode'},'Location','Southeast')
xlabel('x in m')
ylabel('Amplitude E-Feld (ohne Einheit)')
set(1,'papersize',[12,9])
set(1,'paperposition',[0,0,12,9])
print -dpdf modes.pdf
```

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?

```
%Skript fuer die Darstellung des Konvergenzverhaltens der Loesung der
%eindimensionalen Wellengleichung. Verwendet createCC (Erstellung der
%Systemmatrix) und solveCC (Loesen des Eigenwertproblems).
%
%   Eingabe
%   L           Abmessung/Laenge des eindimensionalen Gebietes
%   nmax        Maximale Stuetzstelleanzahl
%   kxind       Betrachtete Mode (Grundmode=1)
%
%   Rueckgabe
%   figure(1)   Plot Konvergenzverhalten kx, linear (wird abgespeichert in
%               plotConv.pdf)
%   figure(2)   Plot Konvergenzverhalten fehler, doppelt-logarithmisch (wird
%               abgespeichert in plotConvloglog.pdf)

% Setzen der Parameter
nmax = 100;
L = 1;
kxind = 1;
nOrd2 = 3:nmax;
nOrd4 = 5:nmax;
dx=@(n)(L./(n.-1));

% Konvergenzstudie fuer Ordnung 2 und keine Randbedingung
disp('Konvergenzstudie fuer Ordnung 2 und keine Randbedingung')
kxOrd2bc0 = zeros(length(nOrd2),1);
for i=1:length(nOrd2)
  n = nOrd2(i);
  kx=solveCC(createCC(n,2,0),dx(n));
  kxOrd2bc0(i) =kx(kxind) ;
end;


% Konvergenzstudie fuer Ordnung 4 und keine Randbedingung
disp('Konvergenzstudie fuer Ordnung 4 und keine Randbedingung')
kxOrd4bc0 = zeros(length(nOrd4),1);
for i=1:length(nOrd4)
  n = nOrd4(i);
kx=solveCC(createCC(n,4,0),dx(n));
  kxOrd4bc0(i) =kx(kxind) ;
end;

% Konvergenzstudie fuer Ordnung 2 und elektrische Randbedingung
disp('Konvergenzstudie fuer Ordnung 2 und elektrische Randbedingung')
kxOrd2bc1 = zeros(length(nOrd2),1);
for i=1:length(nOrd2)
  n = nOrd2(i);
kx=solveCC(createCC(n,2,1),dx(n));
  kxOrd2bc1(i) =kx(kxind) ;
end;
```

```
51
52 % Konvergenzstudie fuer Ordnung 4 und elektrische Randbedingung
53 disp('Konvergenzstudie fuer Ordnung 4 und elektrische Randbedingung')
54 kxOrd4bc1 = zeros(length(nOrd4),1);
55 for i=1:length(nOrd4)
56   n = nOrd4(i);
57 kx=solveCC(createCC(n,4,1),dx(n));
58   kxOrd4bc1(i) =kx(kxind) ;
59 end;
60
61
62 % Formel fuer analytische Luesung
63  kxAna=@(n)(n.*pi./L) ;
64
65
66 % Plot fuer die Wellenzahl ueber Stuetzstellenanzahl
67 figure(1)
68 hold on
69 plot(nOrd2,kxAna(nOrd2));
70 plot(nOrd2,kxOrd2bc0);
71 plot(nOrd4,kxOrd4bc0);
72 plot(nOrd2,kxOrd2bc1);
73 plot(nOrd4,kxOrd4bc1);
74 hold off
75 legend({'analytische Wellenzahl',...
76         'zweite Ordnung ohne Randbed.','vierte Ordnung ohne Randbed.',...
77         'zweite Ordnung mit Randbed.','vierte Ordnung mit Randbed.'
78        },...
79         'Location','Southeast')
80 xlabel('Stuetzstellenanzahl')
81 ylabel('Wellenzahl in 1/m')
82 set(1,'papersize',[12,9])
83 set(1,'paperposition',[0,0,12,9])
84 print -dpdf plotConv.pdf
85
86 % Plot fuer den relativen Wellenzahlfehler ueber Gitterschrittweite (loglog)
87 figure(2)
88
89 %%%
90 %%% Ich weiss nicht
91
92 legend({'zweite Ordnung ohne Randbed.','vierte Ordnung ohne Randbed.',...
93         'zweite Ordnung mit Randbed.','vierte Ordnung mit Randbed.'
94        }, 'Location','Southeast')
95 xlabel('Gitterschrittweite')
96 ylabel('rel. Fehler der Wellenzahl')
97 set(2,'papersize',[12,9])
98 set(2,'paperposition',[0,0,12,9])
99 print -dpdf plotConvloglog.pdf
```

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.

```
% Skript fuer die Visualisierung eines diskretisierten Zylinders.
%
%   Eingabe
%   nd          Anzahl Dreiecksflaechen je Deckel
%   h           Hoehe des Zylinders
%   r           Radius des Zylinders
%
%   Rueckgabe
%   figure(1)   Plot Zylinder (wird abgespeichert in cyl.pdf)

% Parameter setzen
nd=20;
h=1;
r=1;

% Berechnung von delta phi
dphi=2*pi/nd;

% Bestimmung der Arrays fuer die X, Y, Z Koordinate,
% jeweils 3 Zeilen (Punkt 1, 2, 3 des Dreiecks)
% und nd Spalten (Anzahl der Dreiecke)
% und das fuer die vier Dreiecke Deckel, Boden, Mantel 1, Mantel 2
XDeckel = zeros(3, nd);
YDeckel = zeros(3, nd);
ZDeckel = zeros(3, nd);
XBoden = zeros(3, nd);
YBoden = zeros(3, nd);
ZBoden = zeros(3, nd);
XMantel1 = zeros(3, nd);
YMantel1 = zeros(3, nd);
ZMantel1 = zeros(3, nd);
XMantel2 = zeros(3, nd);
YMantel2 = zeros(3, nd);
ZMantel2 = zeros(3, nd);

for i=1:nd
    % Eintrag i in XDeckel, YDeckel, ZDeckel
    XDeckel(1:3,i)=[0; r*cos(i*dphi); r*cos(i*dphi+dphi);];
    YDeckel(1:3,i)=[0; r*sin(i*dphi); r*sin(i*dphi+dphi);];
    ZDeckel(1:3,i)=[h; h; h;];
    % Eintrag i in XBoden, YBoden, ZBoden
    XBoden(1:3,i)=[0; r*cos(i*dphi); r*cos(i*dphi+dphi);];
```

```
    YBoden(1:3,i)=[0; r*sin(i*dphi); r*sin(i*dphi+dphi);];
    ZBoden(1:3,i)=[0; 0; 0;];
    % Eintrag i in XMantel1, YMantel1, ZMantel1
    XMantel1(1:3,i)=[XBoden(2,i); XBoden(3,i); XBoden(3,i);];
    YMantel1(1:3,i)=[YBoden(2,i); YBoden(3,i); YBoden(3,i);];
    ZMantel1(1:3,i)=[ZBoden(2,i); ZBoden(3,i); ZDeckel(3,i);];
    % Eintrag i in XMantel2, YMantel2, ZMantel2
    XMantel2(1:3,i)=[XBoden(2,i); XBoden(2,i); XBoden(3,i);];
    YMantel2(1:3,i)=[YBoden(2,i); YBoden(2,i); YBoden(3,i);];
    ZMantel2(1:3,i)=[ZBoden(2,i); ZDeckel(2,i); ZDeckel(3,i);];

end

% Plotten und Speichern der Dreiecke mithilfe von Patch
figure(1)

patch(XBoden,YBoden,ZBoden,zeros(3,nd));
patch(XDeckel,YDeckel,ZDeckel,zeros(3,nd));
patch(XMantel1,YMantel1,ZMantel1,zeros(3,nd));
patch(XMantel2,YMantel2,ZMantel2,zeros(3,nd));

view([1,-1,0.5])
xlabel('x')
ylabel('y')
zlabel('z')
set(1,'papersize',[12,9])
set(1,'paperposition',[0,0,12,9])
set(gca,'DataAspectRatio',[1 1 1])
print -dpdf cyl.pdf
```

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?

```
%Skript zur Darstellung der Diskretisierungsfehler eines Zylinders.
%
%   Eingabe
%   nmax        Maximale Stuetzstellenanzahl
%
%   Rueckgabe
%   figure(1)   Plot Fehler doppelt-logarithmisch (wird abgespeichert
%               in plotVisErrloglog.pdf)

% Parameter setzen
nmax=100000;    %100000
r=1;
```

```
13 h=5;
14
15 % Vektor mit verschiedener Anzahl von Dreicken
16 nd=3:nmax;
17
18 % Berechnung von diskreter Oberflaeche und Volumen
19  oberflaecheDiskrete = 2.*nd.*r.*(1/2.*r.*sin(2.*pi./nd).+sin(pi./nd).*h);
20  volumenDiskrete = 1/2.*r.^2.*sin(2.*pi./nd).*nd.*h;
21
22 % Berechnung von analytischer Oberflaeche und Volumen
23  oberflaecheAnalytisch = 2.*pi.*r.*(r.+h);
24  volumenAnalytisch = pi.*r.^2.*h;
25
26 % Berechnung von Oberflaechen- und Volumenfehler
27  oberflaechenFehler = abs(oberflaecheDiskrete-oberflaecheAnalytisch)...
28  ./oberflaecheAnalytisch;
29  volumenFehler = abs(volumenDiskrete-volumenAnalytisch)./volumenAnalytisch;
30
31
32 % Plotten der beiden Fehler ueber nd
33 figure(1)
34      loglog(nd,oberflaechenFehler,nd,volumenFehler)
35      xlabel('Anzahl Dreiecke Deckelflaeche')
36      ylabel('relativer Fehler')
37      legend({'Oberflaechenfehler',...
38              'Volumenfehler'},...
39              'Location','Northeast')
40      set(1,'papersize',[12,9])
41      set(1,'paperposition',[0,0,12,9])
42      print -dpdf plotVisErrloglog.pdf
43
44
45 % nd fuer Fehler kleiner als 10^(-5) finden
46 %Marc
47 %for i=1:columns(nd)
48 %  if (volumenFehler(i)<10^-5)
49 %    Dreiecke=i+2
50 %    break;
51 %  endif
52 %endfor
53 for i=1:columns(nd)
54   if oberflaechenFehler(i)<10^(-5) && volumenFehler(i)<10^(-5)
55     Dreiecke = i+2
56     break;
57     endif
58 endfor
```

Um ein Fehler kleiner von 10^{-5} zu garantieren, braucht 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.

```
%% Skript to plot .stl file in octave

filename='zylinder.stl'; % .stl file
%filename='w210.stl';
%filename='coaxial.stl'; % .stl file
%filename='cut.stl'; % .stl file
%filename='cube.stl'; % .stl file
%filename='kugel.stl'; % .stl file
%filename='kugel2.stl';
%filename='microstrip.stl';
%%reading data from .stl file and converting to octave
[X,Y,Z,normals,stlname]=read_stl(filename);
%%patching stuff together and visualising
figure(1)

patch(X,Y,Z,zeros(3,columns(X)));
title(filename);
view([1,-1,0.5])
xlabel('x')
ylabel('y')
zlabel('z')
set(1,'paperposition',[0,0,12,9])
set(1,'paperposition',[0,0,12,9])
print -dpdf strcat(stlname,"cyl.pdf")
```

1.3 Fazit

Fügen Sie hier Ihre Lösung ein