MATHEMATISCHES INSTITUT
PROF. DR. ACHIM SCHÄDLE

Heinrich Heine Universität Düsseldorf

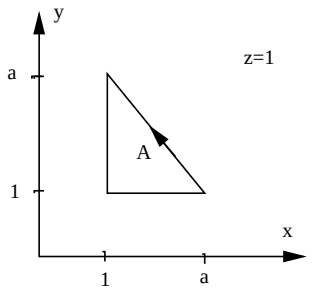
Numerik für Maxwellgleichungen – 7. Übungsblatt

Aufgabe 24:

Berechnen Sie für die 1-Form

$$\omega = yz\mathrm{d}x + xz\mathrm{d}y + xy\mathrm{d}z$$

und die Fläche A



das Integral

$$\int_{\partial A} \omega$$

Aufgabe 25:

Seien U, $V \subset \mathbb{R}^3$ und $\varphi : V \to U$, $\tilde{x} \mapsto x = \varphi(\tilde{x})$ eine C^1-invertierbare Abbildung. Zeigen Sie:

i) Für $\omega \in \Omega^1(U)$ mit $\omega(x) = \sum_{j=1}^3 f_j(x)\mathrm{d}x_j = f(x) \cdot d\vec{s}$ gilt:

$$\varphi^*\omega(\tilde{x}) = \sum_{j=1}^3 \tilde{f}_j(\tilde{x})\mathrm{d}\tilde{x}_j \quad \text{mit} \quad \tilde{f}(\tilde{x}) = D\varphi(\tilde{x})^T f(\varphi(\tilde{x})),$$

wobei $D\varphi$ die Jacobimatrix von φ ist.

ii) Für $\omega \in \Omega^2(U)$ mit $\omega(x) = \sum_{j=1}^3 f_j(x)(-1)^{j-1}\mathrm{d}x_1 \wedge \widehat{\mathrm{d}x_j} \wedge \mathrm{d}x_3$ gilt:

$$\varphi^*\omega(\tilde{x}) = \tilde{f}_1(\tilde{x})\mathrm{d}\tilde{x}_2 \wedge \mathrm{d}\tilde{x}_3 + \tilde{f}_2(\tilde{x})\mathrm{d}\tilde{x}_3 \wedge \mathrm{d}\tilde{x}_1 + \tilde{f}_3(\tilde{x})\mathrm{d}\tilde{x}_1 \wedge \mathrm{d}\tilde{x}_2$$

mit $\tilde{f}(\tilde{x}) = det\,(D\varphi(\tilde{x}))\, D\varphi(\tilde{x})^{-1} f(\varphi(\tilde{x}))$.

b.w

Aufgabe 26:

Modifizieren das Programm aus Aufgabe 21 so, dass Sie die Lösung der Wellengleichung auch für eine ortsabhängige Geschwindigkeit $c = c(x)$ berechnen können. Stellen Sie den Verlauf der Lösung graphisch dar. Der Funktion `wave1d` soll also nun eine Funktion `c` übergeben werden.

Testen Sie Ihr Programm für das Geschindigkeitsprofil $c(x) = 2+\text{sign}(x)$ und wählen Sie Anfangsdaten mit Träger in $[-a, 0]$.

Aufgabe 27:

Schreiben Sie ein Python- oder Matlab-Programm, welches die Lösung folgender Gleichung mit Hilfe der finiten Differenzen Methode der Vorlesung berechnet.

$$\begin{aligned}
\frac{1}{c^2}\partial_{tt}u = \partial_{xx}u + \partial_{yy}u + \partial_{zz}u \qquad & (x,y,z) \in (-a,a)^3,\ t \in (0,T) \\
u(x,y,z,t) = 0 \qquad & (x,y,z) \in \partial((-a,a)^3),\ t \in (0,T) \\
u(x,0) = u^0(x,y,z) \qquad & (x,y,z) \in (-a,a)^3 \\
\partial_t u(x,y,z,t)|_{t=0} = v^0(x,y,z) \qquad & (x,y,z) \in (-a,a)^3
\end{aligned}$$

Schreiben Sie dazu eine Funktion `u = diff2D3D(u,dx,dy,dz)` welche für ein Array $u \in \mathbb{R}^{K+2} \times \mathbb{R}^{L+2} \times \mathbb{R}^{M+2}$ die Anwendung der linearen Abbildung $\delta_x^2 + \delta_y^2 + \delta_z^2$ mit Schrittweiten `dx`, `dy` und `dz` implementiert.

Schreiben Sie weiter eine Funktion `u = wave3d(x,y,z,c,u,v,dt,t,tend)` , welche für

- die Gittervektoren `x`, `y` und `z`
- die Geschwindigkeit c,
- die Funktion der Anfangswerte `u`,
- die Funktion der Anfangsgeschwindigkeiten `v`,
- die Zeitschrittweite `dt`

auf dem Zeitintervall [`t`, `tend`], die Lösung `u` an der Stelle `tend` berechnet und in jedem Zeitschritt die Lösung in der (x, y)-, (y, z)- und (x, z)-Ebene darstellt.

Testen Sie Ihr Programm mit $a = 2$, $K = L = M = 39$, $T = 10$, $v^0(x) = 0$, für $u^0(x) = \mathrm{e}^{-2(x^2+y^2+z^2)}$ und `dt = dx/3` .

Hinweis: Die Matlabfunktionen `ndgrid` und `squeeze` könnten hilfreich sein. In Python erzeugt `numpy.mgrid` ein mehrdimensionales Gitter.

Abgabe der Übungsaufgaben bis 04.06.2021, 8:00 Uhr über ILIAS.
Besprechung in der Übung am 07.06.2021.