Universität Konstanz
Fachbereich Mathematik und Statistik
S. Trenz, Prof. Dr. S. Volkwein

WS 15/16

# ÜBUNGEN ZU Theorie und Numerik partieller Differentialgleichungen

http://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching

## Blatt 11

**Abgabe: Dienstag, 26.01.2016, in der Vorlesung!**

**Aufgabe 1** (Theorie) (6 Punkte)

Sei $\Omega = (a,b) \subset \mathbb{R}$ mit Rand $\Gamma = \{a,b\}$ und äquidistanter Ortsdiskretisierung $\Omega^h = \{x_0, x_1, ..., x_n, x_{n+1}\} \subset \overline{\Omega}$ mit $a = x_0 < x_2 < ... < x_{n+1} = b$ und der Schrittweite $h = \frac{b-a}{n+1}$.

a) Definieren Sie jeweils *stückweise quadratische Ansatzfunktionen* $\{\varphi_i\}_{i=1}^n \subset H^1(\Omega)$ an den inneren Diskretisierungspunkten sowie $\{\varphi_{i+\frac{1}{2}}\}_{i=0}^n \subset H^1(\Omega)$ an zusätzlichen Diskretisierungspunkten $x_{j+\frac{1}{2}} := x_j + \frac{h}{2}$ in der Mitte der jeweiligen Trägerintervalle $[x_j, x_{j+1}]$, so dass gilt:

$$\varphi_i(x_j) = \delta_{ij} \qquad \text{und} \qquad \varphi_i(x_{j+\frac{1}{2}}) = 0,$$
$$\varphi_{i+\frac{1}{2}}(x_j) = 0 \qquad \text{und} \qquad \varphi_{i+\frac{1}{2}}(x_{j+\frac{1}{2}}) = \delta_{ij}.$$

*Hinweis:* Eine quadratische Funktion ist durch ihre Werte an drei Interpolationspunkten eindeutig festgelegt.

b) Berechnen Sie die Ableitungen $\varphi_i'$ und $\varphi_{i+\frac{1}{2}}'$.

c) Zeichnen (oder plotten) Sie jeweils $\varphi_i$ und $\varphi_{i+\frac{1}{2}}$ sowie die Ableitungen $\varphi_i'$ und $\varphi_{i+\frac{1}{2}}'$ auf einem geeigneten Intervall. Achten Sie in jedem Fall auf eine klare und nachvollziehbare Achsenbeschriftung und eine saubere Darstellung.

**Aufgabe 2** (Matlab) (16 Punkte)

Gegeben sie die im Ort zweidimensionale (parabolische) Wärmeleitungsgleichung mit homogener Neumann-Randbedingung

$$\begin{aligned} \frac{\partial}{\partial t}u(x,t) - \Delta u(x,t) &= 0 && \text{in } \Omega \times (0,T), \\ \frac{\partial}{\partial \nu}u(s,t) &= 0 && \text{in } \Gamma \times (0,T), \\ u(x,0) &= u_\circ(x) && \text{in } \Omega, \end{aligned} \tag{1}$$

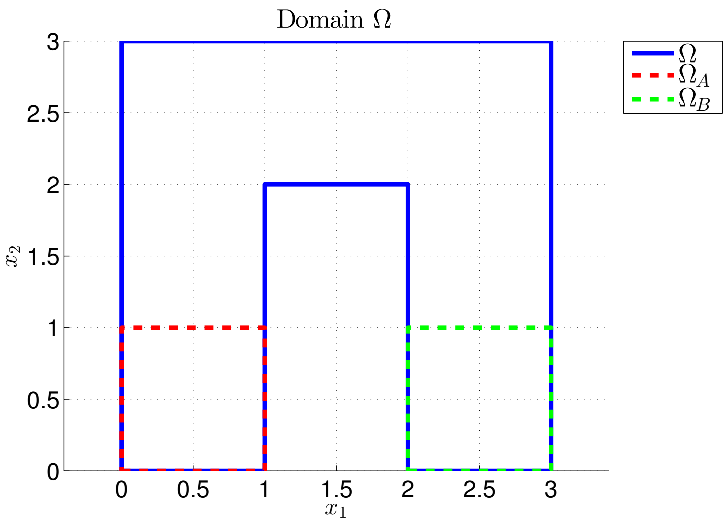


Abbildung 1.

auf dem Ortsgebiet $\Omega \subset \mathbb{R}^2$ mit zugehörigem Rand $\Gamma := \partial\Omega$, sowie den Teilgebieten $\Omega_A, \Omega_B \subset \Omega$, siehe Abb. 1. Lösen Sie (1) numerisch mittels Matlab, in dem Sie zur Ortsdiskretisierung die Finite-Elemente-Methode und zur Zeitdiskretisierung das *implizite* Euler-Verfahren anwenden. Bezüglich der Ortsdiskretisierung mittels FE werden wir uns der Routinen aus der Matlab PDE Toolbox bedienen.

*Hinweis:* Bei Bedarf finden Sie sämtliche Befehle und Routinen neben der Hilfe in Matlab auch im zugehörigen Handbuch *Partial Differential Equation Toolbox User's Guide* detailliert erläutert:

`http://www.mathworks.com/help/pdf_doc/pde/pde.pdf`

Gehen Sie wie folgt vor:

1. Starten Sie mittels dem Befehl `pdetool` die Matlab PDE Toolbox und erzeugen Sie ein Geometry- und Boundary-File, indem Sie zunächst das Gebiet $\Omega$ aus Abb. 1 zeichnen und die Randbedingungen entsprechend der Angabe setzen. Exportieren Sie nun die Geometry- und Boundary-Daten in den Workspace (siehe Menu "Draw" und "Boundary"). Zum Erstellen der entsprechenden m-Files verwenden Sie nun die Matlab-Funktionen `decsg` und `wgeom` sowie `wbound` auf den exportierten Daten (siehe Matlab-Hilfe).

2. Schreiben Sie ein Haupt-File `main.m`, in welchem Sie zunächst ein Gitter mittels `initmesh` erzeugen und mit `refinemesh` verfeinern.

3. Assemblieren Sie nun mittels `assema` die im Weiteren benötigte Steifigkeits- und Massematrix $\mathbf{S}$ und $\mathbf{M}$, indem Sie hierzu die PDE Toolbox Koeffizienten `c` $= 1.0$ und `a` $= 1.0$ sowie die rechte Seite `f` $= 0.0$ setzen.

   *Hinweis:* Im Falle nicht-homogener Randbedingungen müssen diese unter Verwendung von `assemb` und der Erzeugung einer *spring constant* `spc` in der sogenannten *Systemmatrix* sowie in der rechten Seite entsprechend berücksichtigt werden (siehe Handbuch zur PDE Toolbox)!

4. Implementieren Sie das implizite Euler-Verfahren für beliebige Endzeit $T > 0$ sowie einer beliebigen Anzahl an diskreten Zeitschritten $n_t$ und einem Anfangswert der

Gestalt

$$u_\circ(x) = \begin{cases} \alpha & \text{falls } x \in \Omega_A, \\ \beta & \text{falls } x \in \Omega_B, \\ 0 & \text{sonst,} \end{cases}$$

mit $\alpha, \beta \in \mathbb{R}$.

Aufgabenstellung:

a) Setzen Sie $T = 2$, $n_t = 100$, $\alpha = 200$, $\beta = -75$ und lösen Sie (1) mittels Ihres Programms approximativ. Visualisieren Sie anschließend das Ergebnis in einer Animation, indem Sie den Anfangswert $u^{(1)} := u_\circ(\cdot)$ sowie die Lösung $u^{(j)} := u(t_j, \cdot)$ für alle diskreten Zeitpunkten $t_j \in [0, T]$, $j = 2, ..., n_t$, in einer `for`-Schleife innerhalb einer `figure`-Umgebung nacheinander plotten.

   *Hinweis:* Hier können die Matlab-Befehle `hold on/off`, `pdesurf`, `drawnow`, `pause` sowie `clf` hilfreich sein.

b) Was beobachten Sie, wenn Sie den PDE Toolbox Koeffzienten `c` $\geq 5$ setzen?

c) Was passiert, wenn Sie für die Zeitdiskretisierung das *explizite* Euler-Verfahren verwenden?

d) Lösen Sie (1) mit dem Matlab Solver `parabolic` und vergleichen Sie das Ergebnis. Tun Sie dies, indem Sie die Animation in a) zu jedem Zeitpunkt $t_j \in [0, T]$, $j = 2, ..., n_t$, um zwei Plots erweitern, wobei der mittlere die *betragsmäßige Abweichung der Differenz* der beiden Lösung `U1(:,j)` aus a) und `U2(:,j)` aus d) darstellen soll und der letzte die Lösung aus d) selbst (siehe Matlab `subplot`):

   [Plot 1: `U1(:,j)`] [Plot 2: `abs(U1(:,j)-U2(:,j))`] [Plot 3: `U2(:,j)`]

   Was beobachten Sie im zeitlichen Verlauf?

   *Hinweis:* Achten Sie auf eine konsistente Belegung der PDE Toolbox Koeffizienten `c`, `a`, `f`, und `d` entsprechend der Problemstellung aus (1).