Universität Konstanz
Fachbereich Mathematik und Statsitik
Johannes Schropp
Jan Rohleff

Sommersemester 2024

Numerik partieller Differentialgleichungen II

https://www.math.uni-konstanz.de/~rohleff/numpde2.html

1. Übungsblatt

Ausgabe: 08.04.2024, Abgabe: 15.04.2024, bis 10.00 Uhr

Vorbemerkung zu den Programmieraufgaben

Die Programmieraufgaben können in Zweiergruppen bearbeitet werden. Dabei besteht die Option, die Aufgaben entweder in Matlab oder Python zu lösen. Die fertigen Programme sollten anschließend per E-Mail an jan.rohleff@uni-konstanz.de gesendet werden. Es ist wichtig, in den Programmen alle Schritte angemessen zu kommentieren, um die Nachvollziehbarkeit und Verständlichkeit zu gewährleisten.

Aufgabe 1.1 (Theorie - 6 Punkte)

Vorgelegt sei die Neumannsche Randwertaufgabe

$$\begin{aligned} -\Delta u + \lambda u &= g \text{ in } \Omega, \\ \frac{\partial u}{\partial n} &= 0 \text{ auf } \partial\Omega \end{aligned} \tag{1}$$

mit $\lambda \geq 0$, $g \in L^2(\Omega)$, $\Omega \subset \mathbb{R}^2$ beschränktes Gebiet mit stückweise glattem Rand $\partial\Omega$ und strikter Kegeleigenschaft. Dabei ist n der äußere Normaleneinheitsvektor.

a) Leiten Sie eine schwache Formulierung für die Aufgabe (1) her.

b) Zeigen Sie: Ist $\lambda > 0$, so existiert genau eine schwache Lösung $u \in H^1(\Omega)$ von (1).
Hinweise: Satz von Lax-Milgram.

Aufgabe 1.2 (Theorie - 6 Punkte)

Gegeben sei erneut das Problem (1), dieses Mal aber mit $\lambda = 0$. Zeigen Sie:

(i) Es existiert genau eine schwache Lösung

$$u \in V := \left\{ u \in H^1(\Omega) : \int_\Omega u \, dx = 0 \right\}$$

von (1). Hinweis: Wenden Sie den Satz von Lax-Milgram an auf den Raum V mit der Norm $\|u\|_V := \left(\int_\Omega |\nabla u|^2 \, dx \right)^{1/2}$; dafür benötigen Sie die zweite Poincarésche Ungleichung: Für $u \in H^1(\Omega)$ existiert ein $C_P > 0$ mit

$$\|u\|_{L^2} \leq C_P \left(\left(\int_\Omega |\nabla u|^2 \, dx \right)^{1/2} + \left| \int_\Omega u(x) \, dx \right| \right).$$

(ii) Ist u eine klassische Lösung von (1), d. h. $u \in C^2(\Omega) \cap C^1(\overline{\Omega})$, so gilt

$$\int_\Omega g \, dx = 0.$$

Aufgabe 1.3 **(Programm - 9 Punkte)**

Wir betrachten die inhomogene Helmholtz-Gleichung

$$-\Delta u + \lambda u = g \text{ in } \Omega \tag{2}$$

versehen mit den Dirichletschen Randbedingungen auf $\partial\Omega$

$$u|_{\partial\Omega} = \gamma. \tag{3}$$

Das Gebiet Ω sei durch $\Omega = A \backslash B$ definiert, wobei

$$A = \{(x,y) \in \mathbb{R}^2 : x = (r + \sin(6\varphi))\cos(\varphi), y = (r + \cos(6\varphi))\sin(\varphi), \varphi \in [0, 2\pi), r \in [0, 5)\},$$
$$B = \{(x,y) \in \mathbb{R}^2 : 0 \leq x^2 + y^2 \leq 4\}.$$

Ferner gelte $\lambda = 1/2$ und

$$g(x,y) = \cos(x/2)\sin(y/2) + 1/80(x^2 + y^2) - 1/10,$$
$$\gamma(x,y) = \cos(x/2)\sin(y/2) + 1/40(x^2 + y^2).$$

Die exakte Lösung zu (2)–(3) lautet

$$\bar{u}(x,y) = \cos(x/2)\sin(y/2) + 1/40(x^2 + y^2).$$

Finden Sie mit Hilfe der Finite-Elemente-Methode die numerische Lösung u_h von (2)–(3). Benutzen Sie hierfür die PDE-Toolbox von Matlab oder das Python package FEniCS.

Visualisieren Sie jeweils in einem 3D-Plot

- die numerische Lösung u_h,
- die exakte Lösung $\bar{u}$,
- den Fehler $|\bar{u} - u_h|$.

Hinweise zum Vorgehen mit der Matlab PDE-Toolbox:

Schreibe ein Skript `Main.m`, das die folgenden Anweisungen ausführt:

1. Erstellen Sie zunächst eine Struktur `prob`, welche die Gebiet-Spezifikationen A und B, die maximale Gittergröße h, Parameter λ und die rechten Seiten g und γ enthält.
2. Rufe die Funktion `solveHelmholtz.m` auf, die als Eingabe die Struktur `prob` annimmt und als Ausgabe die Struktur `prob` um die numerische Lösung von (2)-(3) und das PDE-Modell erweitert. Die Funktion `solveHelmholtz.m` muss die folgenden Schritte enthalten:
   - Erstelle ein PDE-Modell mit der Funktion `createpde.m` von Matlab.
   - Konstruieren Sie das Gebiet Ω für das Modell. Approximieren Sie dazu A als ein geschlossenen Polygonzug und ziehen Sie davon das Kreisgebiet B ab. Die folgenden Funktionen könnten hier nützlich sein: `decsg.m`, `pdepoly` und `pdecirc`
   - Generieren Sie ein Gitter mit maximaler Elementgröße h.
   - Spezifizieren Sie die PDE-Parameter und Randbedingungen.
   - Lösen Sie die PDE.
   - Geben Sie als Ausgabe die Struktur `prob` mit der Lösung u und dem PDE-Modell als zusätzliche Felder zurück.

Hinweise zum Vorgehen mit Python:

Schreibe ein Skript `Main.py`, das die folgenden Anweisungen ausführt:

1. Erstellen Sie zunächst eine Klasse `prob`, welche die Gebiet-Spezifikationen A und B, die maximale Gittergröße h, Parameter λ und die rechten Seiten g und γ enthält.

2. Rufe die Funktion `solveHelmholtz(prob)` auf, die als Eingabe die Klasse `prob` annimmt und als Ausgabe die numerische Lösung von (2)-(3) zurückgibt. Die Funktion `solveHelmholtz(prob)` muss die folgenden Schritte enthalten:

   - Konstruieren Sie das Gebiet Ω für das Modell. Approximieren Sie dazu A als ein geschlossenen Polygonzug und ziehen Sie davon das Kreisgebiet B ab. Die folgenden Fenics/Dolphin Funktionen könnten hier nützlich sein: `PolygonalMeshGenerator` und `Circle`.
   - Generieren Sie ein Gitter mit maximaler Netzgröße h.
   - Spezifizieren Sie den Funktionenraum V als stückweise stetige Lagrange-Elemente erster Ordnung (d.h. stetige lineare Polynome).
   - Spezifizieren Sie die bilineare Form a, das lineare Funktional l und die Randbedingungen entsprechend der schwachen Formulierung von (2)-(3).
   - Definieren und lösen Sie das lineare variationelle Problem $a(u, \phi) = l(\phi)$ für alle $\phi \in V$.
   - Geben Sie als Ausgabe die Lösung u zurück.