Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

universitätbonn

# Übungen zu Algorithmische Mathematik II (V1G6)

Sommersemester 2015

Prof. Dr. Martin Rumpf — Dr. Orestis Vantzos — Dipl.-Math. Behrend Heeren

## Übungsblatt 4 — Abgabe: 1. Juli 2015

### Aufgabe 1 (Dirichletproblem in 1D) — 4 Punkte

Für ein $N \in \mathbb{N}$ definiere Stützstellen $x_i = ih$ mit $h = N^{-1}$ und $i = 0, \dots, N$. Eine Funktion $u \in C([0,1])$ kann approximiert dargestellt werden durch eine Folge $(u_h^i)_{i=0,\dots,N}$ mit $u_h^i \approx u(x_i)$. Definiere nun den diskreten Differenzenoperator

$$\partial_h^- u_h^i = \frac{u_h^i - u_h^{i-1}}{h} \,.$$

Für ein $f \in C([0,1])$ ist eine diskrete Energie (*Dirichlet*-Energie) definiert durch

$$E_h[u_h] = \sum_{i=1}^{N} \tfrac{1}{2} \, |\partial_h^- u_h^i|^2 \, h - f(x_i) \, h \, u_h^i \,.$$

(*Bemerkung:* Dies stellt eine Approximation an die kontinuierliche Dirichlet-Energie $E[u] = \int_0^1 \frac{1}{2}|u'(x)|^2 - f(x)u(x) \,\mathrm{d}x$ für $u \in C^1([0,1])$ dar.)

(a) Zeigen Sie: Unter den Randbedingungen $u_h^0 = u_h^N = 0$ existiert ein Minimierer von $E_h$.

(b) Geben Sie ein Gleichungssytem an, um den Minimierer von $E_h$ zu berechnen.

(c) Vergleichen Sie das Gleichungssytem aus (b) mit dem für Federketten in Kapitel 2.1.

**Lösung**

(a) Zeige zunächst, dass $E_h$ nach unten beschränkt ist: Da $u_h^0 = 0$ folgt $u_h^k = \sum_{j=1}^k (u_h^j - u_h^{j-1}) = h \sum_{j=1}^k \partial_h^- u_h^i$. Quadrieren ergibt

$$(u_h^k)^2 = h^2 \Big( \sum_{j=1}^k \partial_h^- u_h^i \Big)^2 \leq h^2 k \sum_{j=1}^k \Big( \partial_h^- u_h^i \Big)^2 \leq N \sum_{j=1}^N \tfrac{1}{2} h \Big( \partial_h^- u_h^i \Big)^2 .$$

Wähle $\alpha > 0$ so, dass $\alpha > h \max_i f(x_i)$, d.h. für jedes $k$ gilt

$$E_h[u_h] = \sum_{i=1}^N \tfrac{1}{2} |\partial_h^- u_h^i|^2 h - f(x_i)\, h\, u_h^i \geq N^{-1} (u_h^k)^2 - \alpha \sum_{i=1}^N u_h^i \tag{1}$$

Angenommen, für jedes $C > 0$ gibt es ein $u_h$ mit $E_h[u_h] < -C$. Aus (1) folgt, dass $\max_i u_h^i$ mit $C$ steigt. Wähle also $C > 0$ so, dass $u_h^k := \max_i u_h^i \geq N^2 \alpha$. Dann folgt aber mit

$$E_h[u_h] \geq N^{-1} (u^k)^2 - \alpha \sum_{i=1}^N u_h^i \geq N^{-1} (u^k)^2 - \alpha N u_h^k \geq N^{-1} u_h^k N^2 \alpha - \alpha N u_h^k = 0$$

ein Widerspruch, weswegen $E_h$ nach unten beschränkt ist.

Sei nun $(u_n)_n \subset \mathbb{R}^{N+1}$ mit $u_n = (u_h^{n,i})_i$ eine Minimalfolge, d.h. $E_h[u_n] \to \inf_u E_h[u]$. Mit $|u_h^k| = \max_i |u_h^i| = \|u_h\|_\infty$ folgt aus (1):

$$|u_h^k|^2 \leq N\, E_h[u_h] + \alpha \sum_{i=1}^N u_h^i \leq N\, E_h[u_h] + \alpha N^2 |u_h^k| ,$$

also die Beschränktheit von $(u_n)_n$. Somit besitzt $(u_n)_n$ nach Bolzano-Weierstraß eine konvergente Teilfolge. Konvergenz im $\mathbb{R}^{N+1}$ bedeutet komponentenweise Konvergenz, daher erfüllt auch der Grenzwert die Randwerte.

(b) Die notwendigen Bedingungen für ein Minimum von $E_h : \mathbb{R}^{N+1} \to \mathbb{R}$ sind

$$\begin{aligned} 0 = \partial_i E_h[u_h] &= \Big( \frac{u_h^i - u_h^{i-1}}{h} - f(x_i) h \Big) + \Big( \frac{u_h^{i+1} - u_h^i}{h} \cdot (-1) \Big) \\ \Leftrightarrow \quad h^2 f(x_i) &= 2\, u_h^i - u_h^{i-1} - u_h^{i+1} , \qquad \text{für } i = 1, \dots, N-1 \end{aligned}$$

und entsprechend $h^2 f(x_0) = u_h^0 - u_h^1$ und $h^2 f(x_N) = u_h^N - u_h^{N-1}$. Das ergibt das System unter Berücksichtigung der Randwerte $u_h^0 = u_h^N = 0$ das System

$$h^{-2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \cdot \begin{pmatrix} u_h^1 \\ \\ \vdots \\ \\ u_h^{N-1} \end{pmatrix} = \begin{pmatrix} f(x_1) + u_h^0 \\ \\ \vdots \\ \\ f(x_{N-1}) + u_h^N \end{pmatrix} = \begin{pmatrix} f(x_1) \\ \\ \vdots \\ \\ f(x_{N-1}) \end{pmatrix}$$

(c) Die Matrix stimmt mit der Matrix $A_h$ aus der Vorlesung überein, jedoch ist hier die Skalierung $h^{-2}$. Die rechten Seiten unterscheiden sich, bei den Federketten wirkte die konstante (!) Gravitationskraft.

**Aufgabe 2 (Schwingung einer Federkette)** **4 Punkte**

Die Schwingung einer Federkette wird beschrieben durch die Gleichung

$$\partial_t^2 \mathbf{U} + A_h \mathbf{U} = \mathbf{0},$$

für $\mathbf{U} = \mathbf{U}(t) = (u^1(t), \ldots, u^N(t)) \in \mathbb{R}^N$, wobei $u^0(t) = u^{N+1}(t) = 0$. Die Eigenwerte bzw. Eigenvektoren von $A_h$ seien gegeben durch $\lambda_1, \ldots, \lambda_N$ und $\mathbf{U}_1, \ldots, \mathbf{U}_N$. Leiten Sie aus dem Ansatz

$$\mathbf{U}(t) = \sum_{i=1}^{N} \phi_i(t) \mathbf{U}_i$$

und gegebenen Anfangswerten $\mathbf{U}(0)$ und $\mathbf{U}'(0)$ eine Darstellung der Lösung her.

**Lösung**

Ansatz für $\mathbf{U}(t) = \sum_{i=1}^{N} \phi_i(t) \mathbf{U}_i$:

$$0 = (\partial_t^2 + A_h)\,\mathbf{U}(t) = \sum_{i=1}^{N} \Big(\partial_t^2 \phi_i(t) + \lambda_i \phi_i(t)\Big) \mathbf{U}_i$$

Für $\partial_t^2 \phi_i(t) + \lambda_i \phi_i(t) = 0$ haben wir die Lösung

$$\phi_i(t) = a_i \sin(\sqrt{\lambda_i} t) + b_i \cos(\sqrt{\lambda_i} t)\,.$$

Die Darstellung $\mathbf{U}(0) = \sum_i \alpha_i \mathbf{U}_i$ und $\mathbf{U}'(0) = \sum_i \beta_i \mathbf{U}_i$ der Anfangswerte liefert

$$\mathbf{U}(t) = \sum_i \Big(\alpha_i \cos(\sqrt{\lambda_i} t) + \tfrac{\beta_i}{\sqrt{\lambda_i}} \sin(\sqrt{\lambda_i} t)\Big) \mathbf{U}_i$$

**Aufgabe 3 (Differenzenquotienten höherer Ordnung)** **4 Punkte**

(a) Sei $f \in C^2(\mathbb{R})$ und $x \in \mathbb{R}$. Zeigen Sie mit Hilfe des Satzes von Taylor:

$$\lim_{h \to 0} \frac{(-f(x+2h) + 4f(x+h) - 3f(x))}{2h} = f'(x)$$

(b) Zeigen Sie, dass mit der Formel aus (a) Polynome vom Grad 2 exakt differenziert werden.

**Lösung**

(a) $f: \mathbb{R} \to \mathbb{R}$ sei 2-mal stetig differenzierbar. Taylor ergibt:

$$\begin{aligned} f(x+h) &= f(x) + f'(x) \cdot h + O(h^2) \\ f(x+2h) &= f(x) + f'(x) \cdot 2h + O(h^2) \end{aligned}$$

$$\begin{aligned} \Rightarrow \quad & (-f(x+2h) + 4f(x+h) - 3f(x)) \\ = \quad & (\underbrace{-f(x) + 4f(x) - 3f(x)}_{=0}) + (-f'(x) + 2f'(x)) \cdot 2h + O(h^2) \end{aligned}$$

$$\Rightarrow \quad \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} = f'(x) + O(h)$$

(b) Sei $p(x) = a_0 + a_1 x + a_2 x^2 \in \mathcal{P}_2$, also $p'(x) = a_1 + 2a_2 x$.

$$\begin{aligned} p(x+h) &= a_0 + a_1(x+h) + a_2(x+h)^2 \\ &= a_0 + a_1 x + a_2 x^2 + a_1 h + a_2 2xh + a_2 h^2 \\ &= p(x) + (a_1 + 2a_2 x)h + a_2 h^2 \\ p(x+2h) &= a_0 + a_1(x+2h) + a_2(x+2h)^2 \\ &= p(x) + (a_1 + 2a_2 x)2h + a_2 4h^2 \end{aligned}$$

Dann folgt

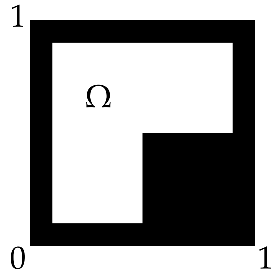
$$\begin{aligned} \frac{-p(x+2h) + 4p(x+h) - 3p(x)}{2h} &= \frac{1}{2h}(\underbrace{-p(x) + 4p(x) - 3p(x)}_{=0}) \\ &\quad + \frac{1}{2h}\left(-(a_1 + 2a_2 x)2h + 2(a_1 + 2a_2 x)2h\right) \\ &\quad + \frac{1}{2h}(\underbrace{-4a_2 h^2 + 4a_2 h^2}_{=0}) \\ &= a_1 + 2a_2 x = p'(x) \end{aligned}$$

**Programmieraufgabe 2 (Abgabe in der Woche 6. - 10. Juli 2015)**

Sei $\Omega \subset [0,1]^2$ ein $L$-förmiges Gebiet und $f \equiv 1$. Lösen Sie

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{auf } \partial\Omega \end{aligned}$$

unter Verwendung einer Finite-Differenzen-Diskretisierung.




Verfahren Sie wie folgt:

1. Das Rechengebiet $D = [0,1]^2$ wird uniform durch die Knotenmenge $x_{ij} = (ih, jh)$ für $0 \leq i, j \leq N$ und $h = N^{-1}$, $N \in \mathbb{N}$, diskretisiert.

2. Das (weiße) Gebiet $\Omega$ wird diskretisiert durch eine Teilmenge der Knoten $x_{ij}$, wobei $0 < i, j < N$ gelten soll. Genauer: $\Omega$ wird beschrieben durch ein Feld $\chi_{ij}$ mit $\chi_{ij} = 1$ falls $x_{ij} \in \Omega$ und $\chi_{ij} = 0$ sonst.

3. Die diskrete Lösung $U$ wird beschrieben durch ein Feld $U_{ij}$ mit $i, j = 0, \ldots, N$. Dises wird mit $U_{ij} = 0$ initialisiert, damit sind die Randwerte vorgegeben.

4. Zu implementieren ist die **Anwendung der Diskretisierungsmatrix** zu $-\Delta$ auf einen Knotenvektor $U$ (ohne die Matrix zu assemblieren!). Dabei iteriert man über das gesamte Feld $U$, identifiziert innere Knoten (d.h. $\chi_{ij} = 1$) und nutzt auf diesen den Differenzenstern

$$\frac{1}{h^2} \begin{bmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{bmatrix}.$$

5. Zur Lösung des Gleichungssystems implementieren Sie das **CG-Verfahren**.

6. Stellen Sie die Lösung $U$ graphisch dar.

*Hinweis:* Auf der Homepage finden Sie ein Progammfragment in C++, welches die Gitterverwaltung und Erstellung der Maske $\chi$ (d.h. 1. - 3.), sowie Output-Routinen zur Verfügung stellt. Insbesondere kann die als ppm-File herausgeschriebene Lösung durch Aufruf der Mathematica Routine `plot.nb` visualisiert werden.