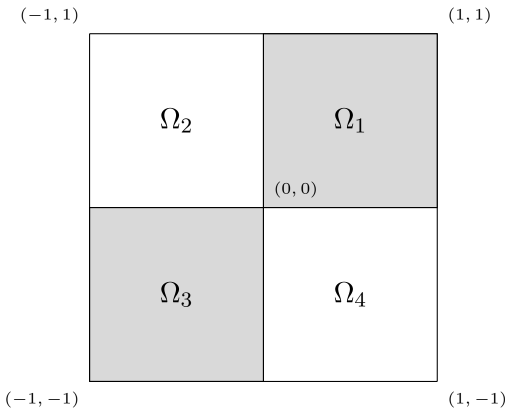
Numerik Partieller Differentialgleichungen, Wintersemester 2010/2011 — Aufgabenblatt 2
Prof. Peter Bastian — Abgabe 5. November 2010
IWR, Universität Heidelberg

ÜBUNG 1 ANALYTISCHE LÖSUNG FÜR HETEROGENE WÄRMELEITUNG



Auf dem oben dargestellten zwei-dimensionalen Gebiet solle die Wärmeleitungsgleichung gemäß

$$\nabla\left(-\lambda \nabla u\right) = 0, \qquad \forall x \in \Omega, \text{ mit } \Omega = \bigcup_{i=1\ldots 4} \Omega_i,$$

gelöst werden. Dabei sei λ stückweise konstant und es gelte

$$\lambda = \begin{cases} \lambda_1 & x \in \Omega_1 \cup \Omega_3 \\ \lambda_2 & x \in \Omega_2 \cup \Omega_4 \end{cases}.$$

1. Zeigen Sie, dass die in Polarkoordinaten gegebene Funktion

   $$p_i(r,\theta) = r^\delta (a_i \sin(\delta\theta) + b_i \cos(\delta\theta))$$

   auf $\Omega \backslash (0,0)$ für Konstanten $a_i, b_i, \delta \in \mathbb{R}$ harmonisch ist, also $\Delta u = 0$ gilt.

2. Die Funktion $p : \Omega \to \mathbb{R}$ sei stückweise durch

   $$p(r,\theta)\big|_{\Omega_i} = p_i(r,\theta), \qquad (i = 1\ldots 4),$$

   definiert. Welche Übgergangsbedingungen müssen an den Rändern

   $$\Omega_1 \bigcap \Omega_2, \quad \Omega_2 \bigcap \Omega_3, \quad \Omega_3 \bigcap \Omega_4, \quad \Omega_4 \bigcap \Omega_1,$$

   gelten, damit p den physikalischen Anforderungen an die Erhaltungseigenschaften des Wärmetransports genügt. Für gegebenes δ entsprechen diese Forderungen einem linearen Gleichungssystem in den a_i und b_i. Stellen Sie dieses System auf.

3. *(Bonus)* Bestimmen Sie explizit (mit Matlab, Maple, Mathematica oder eigenem Programm) die Koeffizienten a_i, b_i, δ unter der zusätzlichen Forderung $\delta = 0.5354409455$.

5 (+ 2) Punkte

ÜBUNG 2 EIGENSCHAFTEN DES ENERGIE-FUNKTIONALS IM DISKRETEN FEDER-SYSTEM

In der Vorlesung wurde gemäß der Darstellung

$$J^{(n)}(u) = J_{\text{el}}^{(n)} + J_{\text{f}}^{(n)} = \sum_{i=0}^{n} \frac{\kappa_i}{2}(\|u_{i+1} - u_i\|) - l_i)^2 - \sum_{i=1}^{n} u_i \cdot f_i$$

die Funktion $J^{(n)} : U \to \mathbb{R}$ vorgestellt, wobei

$$U = \underbrace{\mathbb{R}^3 \times \mathbb{R}^3 \times \cdots \times \mathbb{R}^3}_{n+1 \text{ mal}}.$$

Dies entspricht einer diskreten Approximation der elastischen und (durch Wirkung externer Kräfte bedingten) potentiellen Energie in einem gespannten Faden.

Es gebe ein $\epsilon \in (0, 1)$ so dass

$$\frac{\epsilon}{2} \sum_{i=0}^{n} \|u_{i+1} - u_i\| \geq \sum_{i=0}^{n} l_i.$$

Zeigen Sie, dass dann das Funktional $J^{(n)}(u)$ nach unten beschränkt ist, also

$$\exists C \in \mathbb{R} : J^{(n)}(u) \geq C, \quad \forall u \in U.$$

Gehen Sie wie folgt vor:

1. Zeigen Sie zunächst:

$$J_{\text{el}}^{(n)} \geq \alpha \left(\sum_{i=0}^{n} \|u_{i+1} - u_i\| \right)^2 + \beta \qquad (\alpha, \beta > 0).$$

2. Zeigen Sie auch:

$$\|u\| \leq \sqrt{n+2} \left(\sum_{i=0}^{n} \|u_{i+1} - u_i\| + \|u_0\| \right)$$

3. Verwenden Sie beide Ergebnisse, um

$$J_{\text{el}}^{(n)} \geq \frac{\alpha^*}{2} \|u\|^2 + \beta^*, \qquad (\alpha^* > 0, \beta^* \in \mathbb{R})$$

   zu zeigen. Kombiniert mit einer Abschätzung für $J_{\text{f}}^{(n)}(u)$ führt dies zur Behauptung.

Hilfreiche Ungleichung:

$$\left(\sum_{i=1}^{n} a_i \right)^2 \leq 2^{\log_2 n} \sum_{i=1}^{n} a_i^2$$

10 Punkte

ÜBUNG 3 SIMULATION DES DISKRETEN FEDER-SYSTEMS

Die Lösung des $u \in \mathbb{R}^{3(n+1)}$ des diskreten Energie-Funktionals $J^{(n)}(u)$, welche

$$J^{(n)}(u) \leq J^{(n)}(v) \qquad \forall v \in U$$

erfüllt soll numerisch bestimmt werden. Hierzu ist die Lösung der nicht linearen algebraischen Gleichung

$$\nabla J^{(n)}(u) = 0$$

zu bestimmen. Es gilt:

$$\frac{\partial J(u)}{\partial (u_k)_l} = \kappa_{k-1}(\|u_k - u_{k-1}\| - l_{k-1})\frac{(u_k)_l - (u_{k-1})_l}{\|u_k - u_{k-1}\|} + \kappa_k(\|u_{k+1} - u_k\| - l_k)\frac{(u_k)_l - (u_{k+1})_l}{\|u_{k+1} - u_k\|} - (f_k)_l.$$

Im *dune-npde* Modul ist im Verzeichnis *dune-npde/uebungen/uebung02* ein Programm verfügbar, welches bereits fast alle notwendigen Schritte durchführt, um dieses nicht-lineare Problem durch eine Iteration nach dem Schema

$$\frac{\partial J(u^i, u^{i-1})}{\partial (u_k)_l} = \kappa_{k-1}(\|u_k^{i-1} - u_{k-1}^{i-1}\| - l_{k-1})\frac{(u_k^i)_l - (u_{k-1}^i)_l}{\|u_k^{i-1} - u_{k-1}^{i-1}\|} + \kappa_k(\|u_{k+1}^{i-1} - u_k^{i-1}\| - l_k)\frac{(u_k^i)_l - (u_{k+1}^i)_l}{\|u_{k+1}^{i-1} - u_k^{i-1}\|} - (f_k)_l.$$

zu lösen. Augehend von einem Startwert $u^0 \in \mathbb{R}^{3(n+1)}$ muss in jeder Iteration ein lineares Problem zur Bestimmung der u^i gelöst werden. Lediglich die Funktionen `assembleMatrix(..)` und `assembleRhs(..)`, welche die Matrix und rechte Seite der linearen System aufstellen, sind noch nicht (bzw. unvollständig) implementiert.

1. Vervollständigen Sie diese Implementierung und testen Sie diese. Das Programm wird durch die Datei *uebung02.ini* konfiguriert. Die Voreinstellung entspricht einem Silikon-Kautschuk Faden mit einem Quadratmillimeter Querschnittsfläche, der um das zweieinhalbfache seiner Ruhelänge auf 1000 Meter gespannt wurde.

2. Testen Sie ihre Lösung und erweitern Sie das Programm so, dass es von den y-Koordinaten der Feder-Knoten sowohl den Mittelwert als auch den niedrigsten Wert bestimmt und ausgibt.

3. *(Bonus):* Verwenden Sie in der Matrix-Iterator Schleife in der Methode `assembleMatrix(..)` keine *conditionals*, also Anweisungen, die im kompilierten Code bedingte Sprünge erzeugen könnten (wie `if`, `switch`, `?:`, `std::max(..)`, etc.).

10 (+ 3) Punkte