

Programmierblatt 4.

Bearbeiten bis: **Sonntag, 15.12.2019**

Auf diesem letzten Blatt dieser Serie von Programmierblättern wenden wir uns dem numerischen Lösen der elliptischen Differentialgleichung zu, welche die Verformung eines linear elastischen Körpers, siehe Kapitel 12 der Vorlesung, modelliert:

$$\begin{aligned} -\operatorname{div}(\boldsymbol{\sigma}(\mathbf{u}))(\mathbf{x}) &= \mathbf{f}(\mathbf{x}) \quad \text{für } \mathbf{x} \in \Omega, \\ \mathbf{u}(\mathbf{x}) &= \mathbf{g}(\mathbf{x}) \quad \text{für } \mathbf{x} \in \Gamma_D, \\ \boldsymbol{\sigma}(\mathbf{u})(\mathbf{x})\mathbf{n}(\mathbf{x}) &= \mathbf{0} \quad \text{für } \mathbf{x} \in \Gamma_N. \end{aligned}$$

Hierbei ist

$$\boldsymbol{\sigma}(\mathbf{u})(\mathbf{x}) = \mathfrak{C}(\boldsymbol{\epsilon}(\mathbf{u})(\mathbf{x}))$$

der Spannungstensor, wobei die lineare Abbildung $\mathfrak{C}: \mathbb{R}_{\text{sym}}^{2 \times 2} \rightarrow \mathbb{R}_{\text{sym}}^{2 \times 2}$ den homogenen Elastizitätstensor darstellt und

$$\boldsymbol{\epsilon}(\mathbf{u})(\mathbf{x}) = \frac{1}{2} \left[\nabla \mathbf{u}(\mathbf{x}) + (\nabla \mathbf{u}(\mathbf{x}))^\top \right]$$

der lineare Verzerrungstensor ist. Im folgenden sei der Elastizitätstensor durch

$$\mathfrak{C} \left(\begin{bmatrix} a & c \\ c & b \end{bmatrix} \right) = \lambda \begin{bmatrix} a+b & 0 \\ 0 & a+b \end{bmatrix} + 2\mu \begin{bmatrix} a & c \\ c & b \end{bmatrix}$$

gegeben, wobei λ und μ die Lamé-Parameter sind.

Die Funktion $\mathbf{u}: \Omega \rightarrow \mathbb{R}^2$ beschreibt das Verschiebungsfeld, welches den Körper von dem undeformierten Referenzzustand Ω auf den deformierten Zustand,

$$\{\mathbf{x} + \mathbf{u}(\mathbf{x}) : \mathbf{x} \in \Omega\},$$

abbildet. Die rechte Seite \mathbf{f} entspricht den Volumenkräften, zum Beispiel Gravitationskräfte, die auf den Körper wirken. Auf dem Rand Γ_D betrachten wir eine durch \mathbf{g} vorgegebene Verschiebung und auf dem Rand Γ_N die Situation, wo keine Oberflächenkräfte wirken.

Um das resultierende Gleichungssystem effizient zu lösen, greifen wir hier wie auf Blatt 3 auf das Mehrgitterverfahren zurück. Dabei können einige der bereits implementierten Funktionen direkt verwendet und andere als Startpunkt für die neu zu implementierenden Funktionen benutzt werden.

Schwache Formulierung

Wir nehmen an, dass zu den Dirichlet-Randdaten \mathbf{g} eine Erweiterung $\mathbf{u}_g \in H^1(\Omega; \mathbb{R}^2)$ bekannt ist. Der Ansatz $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}_g$ liefert dann die schwache Formulierung von \mathbf{u}_0 : Finde $\mathbf{u}_0 \in H_{\Gamma_D}^1(\Omega; \mathbb{R}^2)$, so dass

$$\int_{\Omega} \left\langle \mathfrak{C}(\boldsymbol{\epsilon}(\mathbf{u}_0)(\mathbf{x})), \boldsymbol{\epsilon}(\mathbf{v})(\mathbf{x}) \right\rangle_F \mathrm{d}\mathbf{x} = \int_{\Omega} \langle \mathbf{f}(\mathbf{x}), \mathbf{v}(\mathbf{x}) \rangle \mathrm{d}\mathbf{x} - \int_{\Omega} \left\langle \mathfrak{C}(\boldsymbol{\epsilon}(\mathbf{u}_g)(\mathbf{x})), \boldsymbol{\epsilon}(\mathbf{v})(\mathbf{x}) \right\rangle_F \mathrm{d}\mathbf{x}$$

für alle $\mathbf{v} \in H_{\Gamma_D}^1(\Omega; \mathbb{R}^2)$ gilt. Hierbei ist $\langle \cdot, \cdot \rangle_F$ das Frobenius-Skalarprodukt für Matrizen und $H^1(\Omega; \mathbb{R}^2)$ der Unterraum der H^1 -Funktionen, die auf Γ_D null sind.

FEM-Diskretisierung

Zu jeder Triangulierung \mathcal{T}_l des Mehrgitter-Dreiecksnetzes assoziieren wir den Raum der vektorwertigen, linearen Finiten Elemente, der durch $V_l^2 = \text{span}\{\varphi_1, \dots, \varphi_{2n}\}$ gegeben ist. Dabei sind

$$\varphi_{2k-1} = \begin{bmatrix} \varphi_k \\ 0 \end{bmatrix} \quad \text{und} \quad \varphi_{2k} = \begin{bmatrix} 0 \\ \varphi_k \end{bmatrix}$$

für alle $k = 1, \dots, n$, wo φ_k die zum Eckpunkt \mathbf{p}_k assoziierte, skalare, lineare Finite-Elemente-Funktion der Triangulierung \mathcal{T}_l ist.

Die Steifigkeitsmatrix \mathbf{A}_l für die lineare Elastizität auf der Triangulierung \mathcal{T}_l ergibt sich also durch die Einträge

$$a_{i,j} = \int_{\Omega} \langle \mathfrak{C}(\epsilon(\varphi_j)(\mathbf{x})), \epsilon(\varphi_i)(\mathbf{x}) \rangle_F \, d\mathbf{x}$$

und die Massenmatrix \mathbf{M}_l durch die Einträge

$$m_{i,j} = \int_{\Omega} \langle \varphi_j(\mathbf{x}), \varphi_i(\mathbf{x}) \rangle \, d\mathbf{x}.$$

Bezeichnen wir mit $\vec{\psi}_l$ jeweils die Koeffizienten der nodalen Interpolation der Funktion $\psi: \Omega \rightarrow \mathbb{R}^2$ mit der obigen Basis von V_l^2 , so entspricht dies für Finite-Element-Funktionen genau wieder den Koeffizienten. Weiter sind dann

$$\vec{\mathbf{v}}_l^T \mathbf{M}_l \vec{\mathbf{f}}_l \approx \int_{\Omega} \langle \mathbf{f}(\mathbf{x}), \mathbf{v}(\mathbf{x}) \rangle \, d\mathbf{x}$$

und

$$\vec{\mathbf{v}}_l^T \mathbf{A}_l \vec{\mathbf{u}}_{\mathbf{g},l} \approx \int_{\Omega} \langle \mathfrak{C}(\epsilon(\mathbf{u}_{\mathbf{g}})(\mathbf{x})), \epsilon(\mathbf{v})(\mathbf{x}) \rangle_F \, d\mathbf{x}$$

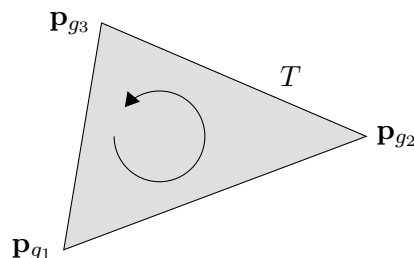
die durch die Trapezregel erhaltenen Approximationen der Terme der rechten Seite.

Damit ist schliesslich die Finite-Element-Approximation der Lösung \mathbf{u}_0 in V_l^2 gegeben durch den Koeffizientenvektor $\vec{\mathbf{u}}_{0,l}$, der die Gleichung

$$\mathbf{A}_l \vec{\mathbf{u}}_{0,l} = \mathbf{M}_l \vec{\mathbf{f}}_l - \mathbf{A}_l \vec{\mathbf{u}}_{\mathbf{g},l},$$

– nach dem Streichen der Zeilen und Unbekannten die durch den Dirichlet-Rand fixiert sind – erfüllt. Der Koeffizientenvektor $\vec{\mathbf{u}}_l$ der Finite-Element-Approximation der Lösung \mathbf{u} berechnet sich dann gemäss $\vec{\mathbf{u}}_l = \vec{\mathbf{u}}_{0,l} + \vec{\mathbf{u}}_{\mathbf{g},l}$.

Assemblierung der FEM-Matrizen



Die Assemblierung der Massen- und Steifigkeitsmatrix erfolgt wie auf Blatt 2 und Blatt 3 mittels der lokalen Elementmatrizen, welche wegen der Vektorwertigkeit nun (6×6) -Matrizen sind. Für das oben dargestellte Dreieck T definieren wir dazu

$$\mathbf{J} = [\mathbf{p}_{g2} - \mathbf{p}_{g1} \quad \mathbf{p}_{g3} - \mathbf{p}_{g1}]$$

und weiter $\gamma \in \mathbb{R}^{3 \times 6}$ mit den Nichtnull-Einträgen

$$\begin{bmatrix} \gamma_{1,1} & \gamma_{1,3} & \gamma_{1,5} \\ \gamma_{3,1} & \gamma_{3,3} & \gamma_{3,5} \end{bmatrix} = \mathbf{J}^{-\top} \begin{bmatrix} -1 & 1 & \\ -1 & & 1 \end{bmatrix} \quad \text{und} \quad \begin{bmatrix} \gamma_{3,2} & \gamma_{3,4} & \gamma_{3,6} \\ \gamma_{2,2} & \gamma_{2,4} & \gamma_{2,6} \end{bmatrix} = \mathbf{J}^{-\top} \begin{bmatrix} -1 & 1 & \\ -1 & & 1 \end{bmatrix}.$$

Damit sind nach kurzer Rechnung die lokalen Elementmatrizen für das oben dargestellte Dreieck T durch

$$\mathbf{M}_T = \frac{|\det \mathbf{J}|}{24} \begin{bmatrix} 2 & 1 & 1 \\ & 2 & 1 & 1 \\ 1 & 2 & 1 \\ & 1 & 2 & 1 \\ 1 & 1 & 2 \\ & 1 & 1 & 2 \end{bmatrix} \quad \text{und} \quad \mathbf{A}_T = \frac{|\det \mathbf{J}|}{2} \gamma^\top \begin{bmatrix} 2\mu + \lambda & \lambda & \\ \lambda & 2\mu + \lambda & \\ & & \mu \end{bmatrix} \gamma$$

gegeben; die dazugehörigen globalen Indizes lauten $2g_1 - 1$, $2g_1$, $2g_2 - 1$, $2g_2$, $2g_3 - 1$ und $2g_3$.

V-Zyklus

Für die Umsetzung des V-Zyklus muss grundsätzlich gegenüber Blatt 3 nur die Prolongation und Restriktion angepasst werden. Aufgrund der gewählten Enumerierung der Basisfunktionen stehen an den ungeraden Stellen in einem Koeffizientenvektor die Koeffizienten zu den Basisfunktionen der Form $\varphi_j \mathbf{e}_1$ und an den geraden Stellen die zu den Basisfunktionen der Form $\varphi_j \mathbf{e}_2$. Daher können die Prolongation oder Restriktion des skalaren Falls einfach separat auf diese zwei Bereiche angewendet und die zwei resultierenden Vektoren geeignet zusammengefügt werden, um die vektorwertige Prolongation respektive Restriktion zu vollführen.

Aufgabe 1.

Schreiben Sie eine Funktion

```
g = vec_compute_nodal_interpolation_mg(gf, mgmesh, 1),
```

welche den Koeffizientenvektor $\vec{\mathbf{g}}$ der nodalen Approximation von \mathbf{g} für das l -te Dreiecksnetz in dem übergebenen Mehrgitter-Dreiecksnetz berechnet. Die Funktion \mathbf{g} wird dabei als `function handle gf` übergeben.

Implementieren sie weiter eine Funktion

```
function vec_mgmesh_function_plot(g, mgmesh, 1),
```

welche die durch den Koeffizientenvektor $\vec{\mathbf{g}}$ und dem l -ten Dreiecksnetz in dem übergebenen Mehrgitter-Dreiecksnetz definierte Funktion als Deformation visualisiert. Dabei verschieben Sie die Eckpunkte der Dreiecke mit der Funktion \mathbf{g} und setzen für den Farbwert an dem jeweiligen Eckpunkte die 2-Norm der Verschiebung.

Aufgabe 2.

Schreiben Sie eine Funktion

```
function [M, A] = ...
    assemble_lin_elast_mass_and_stiffness_mg(mgmesh, lambda, mu),
```

welche die Massen- und Steifigkeitsmatrizen im `sparse`-Format für alle Dreiecksnetze in dem übergebenen Mehrgitter-Dreiecksnetz berechnet und in den Cell-Arrays \mathbf{M} und \mathbf{A} zurückgibt.

Schreiben Sie weiter Funktionen

```
function v = vec_prolong(T, v),  
function v = vec_restrict(T, v),
```

welche die Prolongation respektive Restriktion in dem Fall der vektorwertigen, linearen Finiten Elemente ausführen. Dabei wird in T die Prolongationsmatrix für den skalaren Fall übergeben.

Aufgabe 3.

Schreiben Sie eine Funktion

```
function u = vec_vcycle(u, mgmesh, A, b, Bv, Bn, j, K),
```

die einen Schritt des Mehrgitterverfahrens ausführt. Hierbei bezeichnen Bv und Bn Cell-Arrays der Matrizen, die den Glätter definieren, wobei die ersten für die K Vor- und die zweiten für die K Nachglättungen benutzt werden.

Aufgabe 4.

Schreiben Sie eine Funktion

```
function [us, ts] = ...  
solve_lin_elast_problem_mgcascade(ff, ugf, lambda, mu, mgmesh, K, R),
```

welche das durch die Function-Handles ff und ugf und den Lamé-Parametern λ und μ definierte lineare Elastizitätsproblem auf allen Dreiecksnetzen in dem übergebenen Mehrgitter-Dreiecksnetz löst. Die Lösungen werden dabei als Cell-Array us zurückgegeben, in dem Vektor ts steht die für jede Lösung benötigte Laufzeit.

Benutzen Sie zum Lösen auf dem j -ten Dreiecksnetz R Iterationen des V-Zyklus mit K Vor- und Nachglättungen. Dabei sei jeweils die Prolongation der Lösung zum ein größeren Dreiecksnetz der Startwert. Als Glätter soll der Gauß-Seidel-Glätter benutzt werden.

Aufgabe 5.

Schreiben Sie ein Skript, welches das lineare Elastizitätsproblem mit $\lambda = 1$ und $\mu = 1$ sowie

$$\mathbf{f}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{und} \quad \mathbf{u}_g\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} (1-y)^2 \\ 0.27x \end{bmatrix},$$

auf den Dreiecksnetzen löst, die Sie durch sechsmaliges Verfeinern des durch

```
mgmesh_initialize(mesh_refine(mesh_generation_square()))
```

generierten Mehrgitter-Dreiecksnetz erhalten. Wählen Sie dazu $K = 2$, $R = 2$.

Plotten Sie für jedes Dreiecksnetz die approximative Lösung. Plotten Sie ebenso die jeweilige Laufzeit gegen das Verfeinerungslevel.

Wiederholen Sie dies für

$$\mathbf{f}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \text{und} \quad \mathbf{u}_g\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0.27x \end{bmatrix}$$

und ebenso für

$$\mathbf{f}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{und} \quad \mathbf{u}_g\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0.2 \cos(2\pi x) \end{bmatrix}.$$

Was ändert sich, wenn λ oder μ auf 0.6 oder 3 gesetzt werden?

