Goethe-Center for Scientific Computing (G-CSC)
Goethe-Universität Frankfurt am Main

Modeling and Simulation I

(Practical SIM1, WS 2017/18)

Dr. A. Nägel, Dr. S. Reiter, Dr. M. Hoffer

Aufgabenblatt 10 (Abgabe: Di., 30.1.2018, 23:59h)

Mittels der Finite Elemente Diskretisierung soll der Laplace Operator in 2d diskretisiert werden. Dazu sei $\Omega \subset \mathbb{R}^2$ ein offenes, zusammenhängendes Gebiet und $\mathcal{T} := \{t_0, t_1, ..., t_{m-1}\}$ eine Überdeckung von Ω mit Dreiecken $t_i \subset \mathbb{R}^2$ und es gelte:

- Falls $t_i \cap t_j$ genau aus einem Punkt besteht, so ist dieser ein Eckpunkt sowohl von t_i als auch t_j.
- Falls $t_i \cap t_j$ für $i \neq j$ aus mehr als einem Punkt besteht, so ist $t_i \cap t_j$ eine Kante sowohl von t_i als auch von t_j.

Die Menge der Eckpunkte der Dreiecke $t_i,\ i = 0, \ldots, m-1$ sei durch die Menge $\mathcal{V}_h := \{v_0, v_1, \ldots, v_{n-1}\} \subset \mathbb{R}^2$ gegeben.
Die Lösung des Randwertproblems

$$\begin{aligned} -\Delta u &= 0 && \text{auf } \Omega, \\ u &= g && \text{auf } \partial\Omega, \end{aligned}$$

kann dann für lineare Ansatzfunktionen auf $\mathcal{T}$ mittels des Finite Elemente Verfahrens genähert werden. Zunächst ist

$$\begin{aligned} & -\Delta u = 0 \\ \Leftrightarrow & -\int_\Omega \varphi \cdot \Delta u \; dx = 0 \quad \forall \varphi \in C_0^\infty(\Omega) \\ \Leftrightarrow & \int_\Omega \nabla\varphi \cdot \nabla u \; dx = 0 \quad \forall \varphi \in C_0^\infty(\Omega) \end{aligned}$$

Im Folgenden betrachten wir statt $C_0^\infty(\Omega)$ nun den Raum $S_h(\mathcal{T})$ mit auf jedem Dreieck t_i linearen Basisfunktionen $\psi_k,\ k = 0, \ldots, n-1$ mit $\psi_k(v_k) = 1$ und $\psi_k(v_j) = 0$ falls $k \neq j$. Auf t_i gilt dann $\psi_k(x, y) = a + bx + cy$ mit von i und k abhängigen Koeffizienten $a, b, c \in \mathbb{R}$.

Statt $u \in H_0^1(\Omega)$ suchen wir nun $u_h \in S_h(\mathcal{T})$. u_h lässt sich über die Basisfunktionen von S_h als $u_h(x) = \sum_{i=0}^{n-1} z_k \psi_k(x)$ mit Koeffizienten $z_k \in \mathbb{R}$ schreiben. Dann ergibt sich:

$$\begin{aligned}
& \int_{\mathcal{T}} \nabla\varphi \cdot \nabla u_h \; dx = 0 \quad \forall \varphi \in S_h \\
\Leftrightarrow & \int_{\mathcal{T}} \nabla\psi_k \cdot \nabla u_h \; dx = 0 \quad k = 0, \ldots, n-1 \\
\Leftrightarrow & \int_{\mathcal{T}} \nabla\psi_k \cdot \nabla \sum_{i=0}^{n-1} z_i \psi_i \; dx = 0 \quad k = 0, \ldots, n-1 \\
\Leftrightarrow & \sum_{i=0}^{n-1} z_i \int_{\mathcal{T}} \nabla\psi_k \cdot \nabla\psi_i \; dx = 0 \quad k = 0, \ldots, n-1 \\
\Leftrightarrow & \sum_{i=0}^{n-1} z_i \sum_{j=0}^{m-1} \int_{t_j} \nabla\psi_k \cdot \nabla\psi_i \; dx = 0 \quad k = 0, \ldots, n-1
\end{aligned}$$

Hieraus ergeben sich n Gleichungen mit n Unbekannten. Dies lässt sich als Gleichungssystem auffassen:

$$Az = b$$

mit einer Matrix $A \in \mathbb{R}^{n \times n}$, dem Vektor der Unbekannten $z = (z_0, \ldots, z_{n-1})^T \in \mathbb{R}^n$ und rechter Seite $b \in \mathbb{R}^n$

Aufgabe 1 (Finite Elemente Diskretisierung, 20P)
Laden Sie sich von der Vorlesungshomepage (*https://gcsc.uni-frankfurt.de/simulation-and-modelling/lectures-courses*) die Datei 'quad_4x4.2df' sowie die verlinkten Quelltexte zu 'File2DReader' sowie 'LUSolver' herunter. Erstellen Sie ein neues VRL Projekt und fügen Sie die Quellcodes zu File2DReader und LUSolver in jeweils eine Groovy-Code Komponente ein. Mittels 'File2DReader' können Sie die Datei 'quad4x4.2df' öffnen. Der File2DReader liefert Ihnen

- ein Array `vrts` von Koordinaten der Form $double[n][2]$, wobei `vrts[i][0]` und `vrts[i][1]` die $x-$ bzw $y-$Koordinate des i-ten Vertex sind.
- ein Array `tris` von Dreiecksindizes der Form $int[m][3]$, wobei `vrts[tris[i][0]]`, `vrts[tris[i][1]]` und `vrts[tris[i][2]]` die Koordinaten der Eckknoten des i-ten Dreiecks sind.

- ein Array `vrtMarks` von Markierungen der Form $int[n]$, wobei `vrtMarks[i] == 0`, falls das i-te Vertex ein innerer Knoten ist und `vrtMarks[i] == 1`, falls das i-te Vertex ein Randknoten ist.

Schreiben Sie eine Klasse `LaplaceDiscFE`, zur Diskretisierung der Matrix A und der rechten Seite b mittels Finite Elemente Methode für $-\Delta u = 0$ auf Ω und $u(x) = x_0 \cdot x_1$ auf $\partial\Omega$.
Falls `vrtMarks[i] == 1`, so setzen Sie die i-te Zeile von A als Einheitszeile (1 auf der Diagonalen, Rest 0). Setzen Sie entsprechende Einträge in b auf den zugehörigen Randwert.
Geben Sie die Matrix als $double[][]$ Array und b als $double[]$ Array aus. Visualisieren Sie A und b über den Matrix-Plotter (für b: stride 4).
Lösen Sie das Gleichungssystem und Plotten Sie die Lösung ebenfalls über den Matrix-Plotter (stride 4).

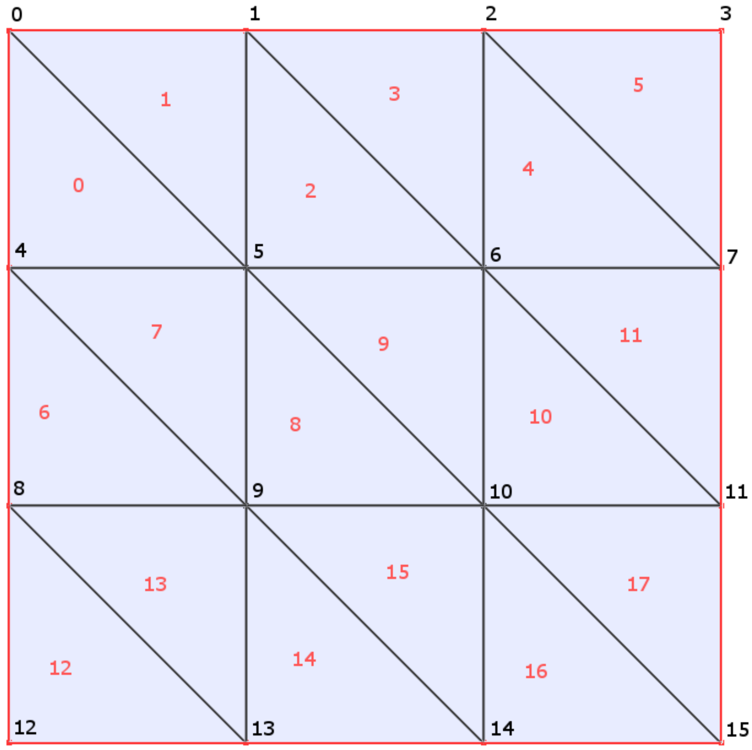
Hinweise:

- Die Gradienten der einzelnen Basisfunktionen sind auf jedem Dreieck konstant. Dies lässt sich nutzen um die Auswertung des Integrals über das Skalarprodukt der Gradienten auf einem Dreieck zu vereinfachen!

- Der zur Verfügung gestellte LUSolver (neue Version!) nutzt zeilenweise Pivotisierung um zufällige Auslöschungen des Diagonalelements im Zuge der LU-Zerlegung zu vermeiden. Dies ist wichtig um die Koeffizienten der Basisfunktionen auf einzelnen Dreiecken zuverlässig zu berechnen.

- Die resultierende Matrix sollte für dieses Beispiel bis auf den Faktor h^2 genau der Fünfpunktsternmatrix plus Einheitszeilen für Randwerte bei Finite-Differenzen Diskretisierung auf einem Vierecksgitter auf den gegebenen Knoten entsprechen.

- Bei der Programmierung in der VRL kann es dazu kommen, dass Änderungen an Quelltexten in anderen Komponenten nicht sofort berücksichtigt werden. Die Berücksichtigung lässt sich nach Rechtsklick auf den Header des Fensters einer Instanz einer abhängigen Komponente und anschließender Auswahl von 'Reset Instance' erzwingen.



Das Gitter 'quad_4x4.2df' mit Knotenindizes (schwarz) und Dreiecksindices (rot).

Anmerkung: Senden Sie den Quelltext als VRL-Studio Projekt (.vrlp Datei) und die Antworten zu den Fragen als E-Mail. Schicken Sie die PDFs/PNG-Dateien, die Sie mit den Plottern erstellt haben, als Anhang mit.
Senden Sie Ihre Lösungen an `practical.sim1@gcsc.uni-frankfurt.de`.
Abgabe bis spätestens Dienstag, 30.1.2018, 23:59h.