Parallele FEM

Wir betrachten das Dirichlet-Randwertproblem für die Poissongleichung

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

mit $f \equiv 1$, wobei

(i) $\Omega = [0,\, 1]^2$ (Einheitsquadrat),

(ii) Ω ein L-Gebiet, siehe Fig. 2.

Wir diskretisieren das Randwertproblem mittels linearer finiter Dreieckselemente auf einem Netz bestehend aus $8n^2$ bzw. $6n^2$ kongruenten gleichschenklig-rechtwinkligen Dreiecken, vgl. Fig. 2 ($n = 2$). Das zugehörige Gleichungssystem

$$K_h u = f_h$$

soll mit parallelen Methoden gelöst werden.

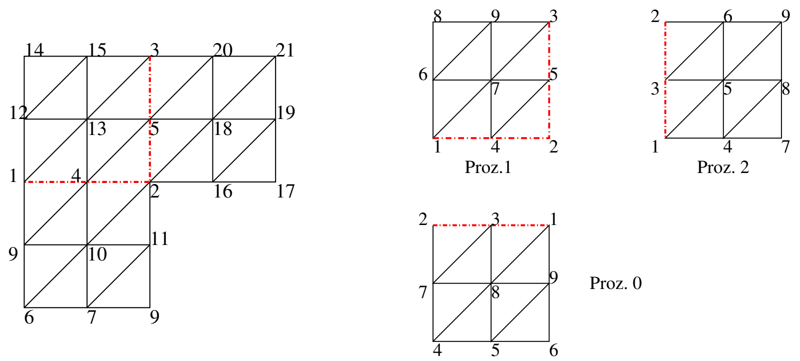


Figure 2: Links: Globales Netz für L-Gebiet. Rechts: Verteiltes Netz für L-Gebiet.

1. Beschreibe für das Netz in Fig. 2 die Booleschen Matrizen H_0, H_1, H_2 (siehe Vorlesung).

2. Stelle eine Datenstruktur für (FEM-Knoten-basierte) akkumulierte/verteilte Vektoren bereit. Programmiere die Akkumulations-Operation für Fall (i) mit mindestens 4 und Fall (ii) mit mindestens 3 Prozessoren, n beliebig. Der Einfachheit halber dürfen globale Vektoren von der Größe der Anzahl der globalen Koppelknoten im Zuge der Kommunikation aufgestellt werden.

   Hinweis: Sinnvolle Nummerierung der Unbekannten vornehmen (siehe Vorlesung).

3. Schreibe eine Routine zur *parallelen* Matrix-Vektor-Multiplikation $\vec{s} = K_h\, \underline{u}$ (hier $\underline{u}$ akkumuliert, $\vec{s}$ verteilt), wobei K_h die FEM-Steifigkeitsmatrix bezeichnet.

   Hinweise:

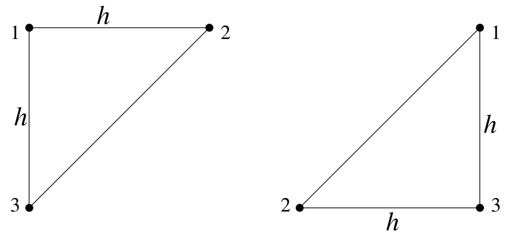
   - Die Matrix K_h soll *nicht* abgespeichert werden.

- Die Operation $K_h u$ auf Prozessor p kann im Wesentlichen durch Assemblierung über die Elemente dargestellt werden:

  $$\sum_{s=1}^{nel(p)} L_s K_s (L_s^\top u)\,,$$

  wobei K_s die Neumann-Element-Steifigkeitsmatrix auf Element R_s und L_s eine Connectivity-Operation (mit evtl. Umordnung der Unbekannten) bezeichnet (siehe Vorlesung).

- Für die Dreiecke

  mit der entsprechenden Element-basierten Knoten-Nummerierung gilt:

  $$K_{i,T} = \frac{1}{2}\begin{pmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \quad \text{bzw.} \quad K_{i,T} = \frac{1}{2}\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{pmatrix}.$$

- Um die korrekte Matrix-Vektor-Multiplikation zu erhalten, soll die Dirichlet-Randbedingung durch Penalty-Technik eingebaut werden.

4. Schreibe eine Routine zur Berechnung der rechten Seite $\vec{f}_h$.

   Hinweis: Die Einträge von $\vec{f}_h$ bestehen im Wesentlichen aus den Integralen $f|_{R_s} \int_{R_s} \varphi_j \, dx = f|_{R_s} \frac{h^2}{6}$ wenn man die Funktion f als stückweise konstant voraussetzt. Einbau der Dirichlet-Randbedingung durch Penalty-Technik.

5. Realisiere einen *parallelen* Jacobi-Vorkonditionierer $C = D_h$ zur Steifigkeitsmatrix K_h. Von welcher Art ist der Eingangs/Ausgangs-Vektor in Hinblick auf die folgende Aufgabe zu wählen?

6. Implementiere das PCG-Verfahren zur *parallellen* Lösung des obigen Randwertproblems mit dem Jacobi-Vorkonditionierer und untersuche die Anzahl der Iterationen für $n = 2, 4, 8, 16, \ldots$

7. **Bonus-Aufgabe:** Visualisiere die Lösung, z.B. mit `gnuplot`. Jeder Prozessor schreibt die Koordinaten der FEM-Knoten und die dort ausgewertete Lösung in ein *eigenes* Datenfile. Mit dem `gnuplot`-Befehl `splot` können Surface-Plots von einer beliebigen Anzahl von Datenfiles übereinandergelegt angezeigt werden:

```
splot 'proc0.dat' using 1:2:3 title '' w p, \
  'proc1.dat' using 1:2:3 title '' w p, \
  'proc2.dat' using 1:2:3 title '' w p
```