Basic Numerical Routines in Parallel

Kapitel 5

aus dem Buch:

„A Tutorial on Elliptic PDE Solvers and Their Parallelization“

von Craig C. Douglas, Gundolf Haase, Ulrich Langer

Kapitel 5
Basic Numerical Routines in Parallel

1. Speicherung schwach besetzter Matrizen
2. Zerlegung nicht überlappender Elemente
3. Vektor-Vektor Operationen
4. Vektor-Matrix Operatione

5.1 Speicherung schwach besetzter Matrizen

Eigenschaften der Matrizen, die wir hier betrachten wollen:

- Schwach besetzt
- Unstrukturiert

5.1 Speicherung schwach besetzter Matrizen

Compressed row storage (CRS):

– Nur die nicht null Elemente der Matrix werden gespeichert

– Zur Speicherung werden nur 2 ganzzahlige Vektoren und ein reeller benötigt

– Aufwand ist dabei nicht abhängig von der Größe der Matrix sondern von der Anzahl der Einträge

5.1 Speicherung schwach besetzter Matrizen

Beispielmatrix: $$K_{n\times m} = \begin{pmatrix} 10 & 0 & 0 & -2 \\ 3 & 9 & 0 & 0 \\ 0 & 7 & 8 & 7 \\ 3 & 0 & 8 & 7 \end{pmatrix}$$

Values : *val* =	10	-2	3	9	7	8	7	3	8	7
Comumn index : *col_ind* =	1	4	1	2	2	3	4	1	3	4
Row pointer : *row_ptr* =	1	3	5	8	11					

5.1 Speicherung schwach besetzter Matrizen

Andere Speicher Schemata:

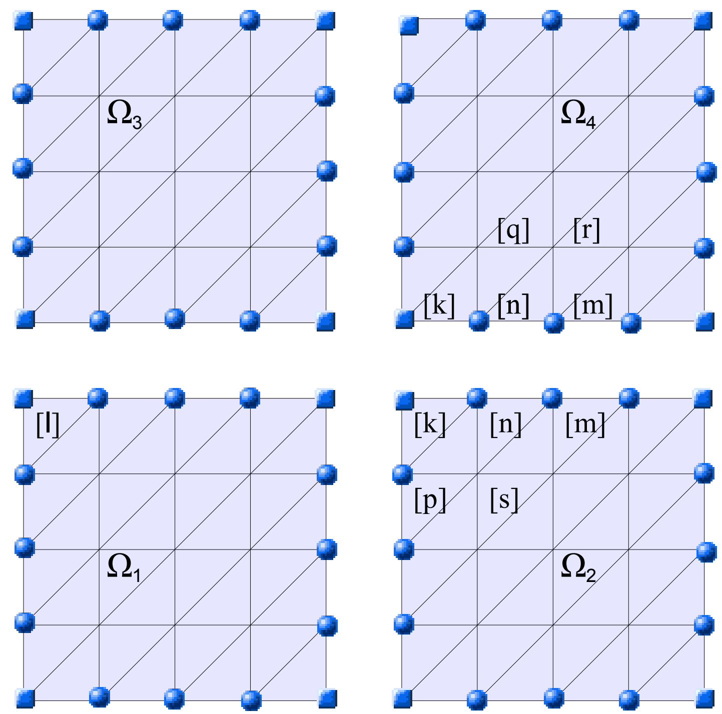
- Compressed column storage (CCS)
- Compressed diagonal storage (CDS)
- Jagged diagonal storage (JDS)

5.2 Zerlegung nicht überlappender Elemente

„I“ bezeichnet Knoten im Inneren des Gebietes

- „E“ sind Knoten auf den Kanten

- „V“ sind die Eckknoten



5.2 Zerlegung nicht überlappender Elemente

Nummerierung der Knoten:

zuerst die Eckpunkte, dann die Knoten auf den Kanten und dann die Innenknoten. Knoten eines Types werden fortlaufend Nummeriert.

Struktur eines Vektors:

$$v^T = (v_V, v_{E,1}, \ldots, v_{E,n_e}, v_{I,1}, \ldots, v_{I,P})$$

5.2 Zerlegung nicht überlappender Elemente

Anwendung der Blockstruktur bei der Gleichung $Ku = f$

$$\begin{pmatrix} K_V & K_{VE} & K_{VI} \\ K_{EV} & K_E & K_{EI} \\ K_{IV} & K_{IE} & K_I \end{pmatrix} \begin{pmatrix} u_V \\ u_E \\ u_I \end{pmatrix} = \begin{pmatrix} f_V \\ f_E \\ f_I \end{pmatrix}$$

5.2 Zerlegung nicht überlappender Elemente

Koinzidenz Matrix:

A_i ist eine boolesche Matrix, die den globalen Vektor v auf den lokalen Vektor v_i abbildet.

Eigenschaften:

- Einträge für Innere Punkte erscheinen genau einmal pro Zeile und Spalte.
- A_i enthält Eintrag für einen Eckknoten, wenn dieser zu Prozess i gehört.

5.2 Zerlegung nicht überlappender Elemente

Vektortypen :

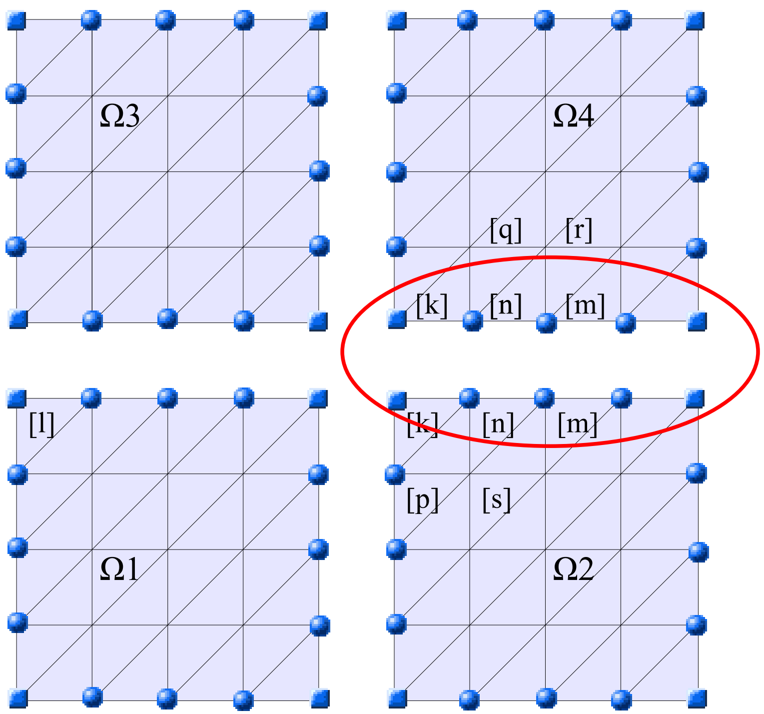
Typ I (accumulated vector):

$$u_i = A_i u$$

Typ II (distributed vector):

$$r = \sum_{i=1}^{P} A_i^T r_i$$

5.2 Zerlegung nicht überlappender Elemente



5.2 Zerlegung nicht überlappender Elemente

Matrixtypen:

analoges Vorgehen:

Typ II Matrix: $K = \sum A_i^T K_i A_i$

Typ I Matrix: $M_i = \sum A_i M A_i^T$

wobei für $K = M$ nicht gilt $K_i = M_i$

5.2 Zerlegung nicht überlappender Elemente

Diagonalmatrix:

$$R = \sum A_i^T A_i$$

enthält für jeden Knoten die Anzahl der Teilgebiete zu denen er gehört.

5.3 Vektor-Vektor Operationen

Welche Operationen benötigen Datenaustausch?

- Umwandlung von Typ II zu Typ I

$$w_i = A_i \sum_{i=1}^{P} A_i^T r_i \rightarrow r = \sum_{i=1}^{P} A_i^T r_i$$

- Umkehrung ist nicht eindeutig, eine Möglichkeit:

$$r_i = R^{-1} w_i$$

5.3 Vektor-Vektor Operationen

- Inneres Produkt verschiedener Vektortypen:

$$(w,r)=w^T r=w^T \sum_{i=1}^{P} A_i^T r_i=\sum_{i=1}^{P} (A_i w)^T r_i=\sum_{i=1}^{P} (w_i, r_i)$$

5.4 Matrix-Vektor Operationen

1. *Typ II Matrix * Typ I Vektor ergibt einen Typ II Vektor*

   (aus Definition 5.1)

$$K\,w = \sum_{i=1}^{P} A_i^T K_i A_i \cdot w = \sum_{i=1}^{P} A_i^T K_i \cdot w_i = r$$

2. *Typ II Matrix * Typ II Vektor*

   benötigt eine Umwandlung des Vektors

5.4 Matrix-Vektor Operationen

3. *Die Operation Typ I Matrix * Typ I Vektor kann nicht mit einer allgemeinen Typ I Matrix M durchgeführt werden*

Analyse:

Untersuche $u = M \cdot w$ am Knoten n

Lokal sollte das für $u_i = M_i w_i$ einen Typ I Vektor ergeben.

5.4 Matrix-Vektor Operationen

Ergebnis:

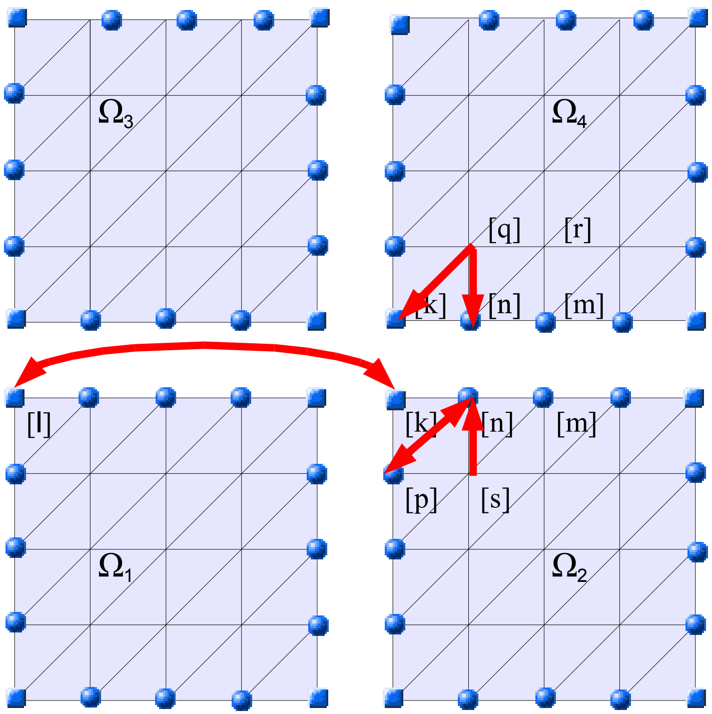
$$u_2^{[n]} = M_2^{[n,n]} w_2^{[n]} + M_2^{[n,m]} w_2^{[m]} + M_2^{[n,k]} w_2^{[k]} + M_2^{[n,p]} w_2^{[p]} + M_2^{[n,s]} w_2^{[s]}$$

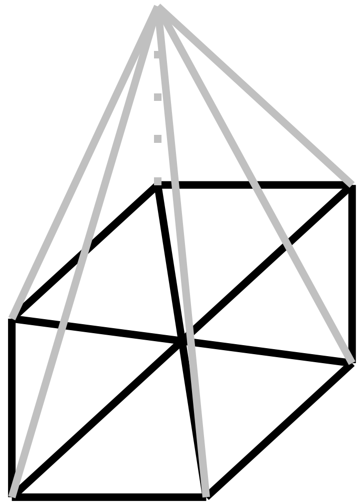
$$u_4^{[n]} = M_4^{[n,n]} w_4^{[n]} + M_4^{[n,m]} w_4^{[m]} + M_4^{[n,k]} w_4^{[k]} + M_4^{[n,q]} w_4^{[q]} + M_4^{[n,r]} w_4^{[r]}$$

Sei $\sigma^{[i]} = \{ s : x_i \in \overline{\Omega_s} \}$ die Menge der Prozessoren, zu denen der Knoten i gehört. Dann ist der Transport von Informationen von j nach i nur erlaubt, wenn gilt:

$$\sigma^{[i]} \subseteq \sigma^{[j]}$$

5.4 Matrix-Vektor Operationen





5.4 Matrix-Vektor Operationen

Das führt zu folgender Einschränkung der Matrix:

$$M = \begin{bmatrix} M_V & 0 & 0 \\ M_{EV} & M_E & 0 \\ M_{IV} & M_{IE} & M_I \end{bmatrix} \Rightarrow \qquad u = M \cdot w$$

mit Blockdiagonalmatrizen M_I, M_E, M_V

5.4 Matrix-Vektor Operationen

Wie kommt man auf diese Struktur?

- Die Blockstrukturen von M_I, M_{IE} und M_{IV} werden durch die entsprechende Datenzerlegung erreicht

- Für die anderen Eigenschaften müssen wir das zugehörige Netz abändern, damit es folgende Eigenschaften erfüllt:

5.4 Matrix-Vektor Operationen

a) Es darf keine Verbindung zwischen Eckpunkten von verschiedenen Teilgebieten und zwischen Eckpunkten auf gegenüberliegenden Seiten geben.

$\Rightarrow$ Struktur von M_V und M_{EV} ist sichergestellt.

b) Keine Verbindung zwischen Kantenknoten auf verschiedenen Kanten.

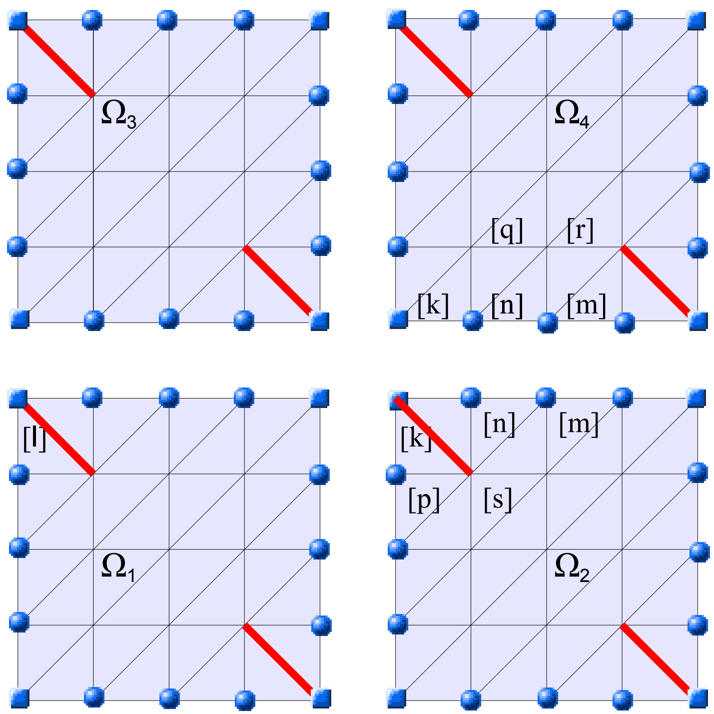
$\Rightarrow$ Sichert die Blockdiagonalität von M_E

5.4 Matrix-Vektor Operationen

Problem:

- Die Strategie kann nur auf Dreiecks oder Tetraeder Finite Elemente angewendet werden
- Generell erfüllen rechteckige Elemente nicht Bedingung b)

5.4 Matrix-Vektor Operationen



5.4 Matrix-Vektor Operationen

4. *Typ I Matrix * Typ II Vektor kann nicht mit einer allgemeinen Typ I Matrix durchgeführt werden.*

   Ein Typ II Vektor wird erwartet.

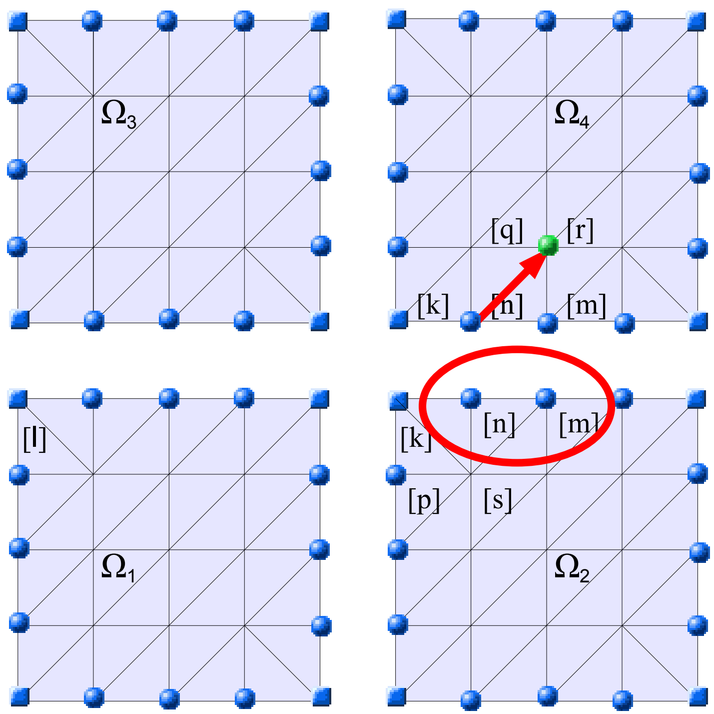
   Betrachte : $f_I = M_{IC} \cdot r_C$ (Wobei C knoten V und E vereinigt)

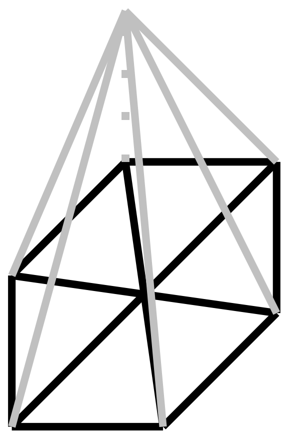
   lokal auf dem Prozessor 4 und dem Knoten r ergibt das:

$$f_4^{[r]} = M_{IC,4}^{[r,n]} \cdot r_4^{[n]} + M_{IC,4}^{[r,m]} \cdot r_4^{[m]}$$

Dabei fehlen die Einträge: $M_{IC,2}^{[r,n]} \cdot r_2^{[n]}$ und $M_{IC,2}^{[r,m]} \cdot r_2^{[m]}$

5.4 Matrix-Vektor Operationen





5.4 Matrix-Vektor Operationen

Nur Matrizen von der Form:

$$M = \begin{pmatrix} M_V & M_{V,E} & M_{V,I} \\ 0 & M_E & M_{E,I} \\ 0 & 0 & M_I \end{pmatrix} \Rightarrow f = \sum_{i=1}^{P} A_i^T f_i = \sum_{i=1}^{P} A_i^T (M_i r_i)$$

mit Blockdiagonalmatrizen M_I, M_E, M_V können für diese Art von Matrix-Vektor Multiplikation verwendet werden.

5.4 Matrix-Vektor Operationen

Bemerkung:

Man kann Punkt 3 und 4 auch zusammenfassend behandeln: Seien M_L, M_U und M_D der strikte untere, obere und diagonal Teil der Matrix M.

$$w = M \cdot u = (M_L + M_D) \cdot u + \sum_{i=1}^{P} A_i^T M_{U,i} R_i^{-1} \cdot u_i$$

$$w = M \cdot r = (M_L + M_D) \cdot \sum_{i=1}^{P} A_i^T \cdot r + \sum_{i=1}^{P} A_i^T M_{U,i} \cdot r_i$$

$$\Rightarrow$$

$$f = M \cdot u = R^{-1} (M_L + M_D) \cdot u + M_U R^{-1} \cdot u$$

$$f = M \cdot r = R^{-1} (M_L + M_D) \sum_{i=1}^{P} A_i^T r_i + M_U \cdot r$$

5.4 Matrix-Vektor Operationen

Falls eine Faktorisierung der Matrix $M = L\,U$ existiert, dann ergibt sich:

$$w = L\,U \cdot r = L \sum_{i=1}^{P} A_i^T U_i \cdot r_i$$

$$f = L\,U \cdot u = R^{-1} L \sum_{i=1}^{P} A_i^T U_i R^{-1} \cdot u_i$$

Analog für $M = U\,L$