DFG-Forschergruppe „SPC“ · Fakultät für Mathematik

Ulrich Groh

Lokale Realisierung von Vektoroperationen auf Parallelrechnern

Preprint-Reihe der Chemnitzer DFG-Forschergruppe „Scientific Parallel Computing“

SPC 94_2 **März 1994**

Lokale Realisierung von Vektoroperationen auf Parallelrechnern

Summary: For the basic algebraic vector operations several variants of a local implementation on distributed memory parallel computers are presented and discussed systematically. In particular necessary and sufficient conditions are shown for the local realizability of the multiplication matrix by vector.

1 Einleitung

FEM–Schemata zur Lösung partieller Differentialgleichungen im Gebiet Ω werden auf MIMD–Parallelrechnern mit verteiltem Speicher oft mittels nichtüberlappender Gebietsdekompositionstechniken realisiert, vgl. zum Beispiel [2], [1]. Die Zerlegung des Gesamtgebietes Ω in paarweise disjunkte Teilgebiete Ω_s, $s=1(1)p$,

$$\overline{\Omega} = \bigcup_{s=1}^{p} \overline{\Omega}_s, \qquad \Omega_s \cap \Omega_t = \emptyset \quad \text{für} \quad s \neq t,$$

und deren Abbildung auf die p Prozessoren P_s entsprechend $\Omega_s \to P_s$, $s=1(1)p$, führt zu einer Verteilung der N Knoten des FEM–Netzes auf die Prozessoren. Auf die gleiche Weise werden auch die Knotenwerte der FEM-Ansatzfunktionen auf die Prozessoren aufgeteilt, das heißt die Koordinaten x_i von Vektoren $\underline{x} \in \mathbf{R}^N$.

Bei der Implementierung schneller Lösungsalgorithmen für die zugehörigen Gleichungssysteme der FEM ist es wünschenswert, die grundlegenden Vektoroperationen (vor allem Skalarprodukt, Matrix mal Vektor) unter Ausnutzung der Datenverteilung weitgehend lokal zu realisieren, das heißt durch voneinander unabhängige Operationen auf den einzelnen Prozessoren und mit möglichst wenig zusätzlicher Kommunikation zwischen den Prozessoren. Probleme entstehen durch Knoten und damit Knotenwerte, die sich im Besitz mehrerer Prozessoren befinden.

Für diese Problematik wurde von Meyer [2] ein Konzept entwickelt, das auf zwei unterschiedlichen lokalen Speichertypen von Vektoren beruht, die einerseits den Vektoren der Unbekannten entsprechen (Näherungen, Korrekturen, Suchrichtungen), andererseits den linearen Funktionalen in der Variationsformulierung des diskreten Problems (rechte Seiten, Defekte). Dieses Konzept wird in der vorliegenden Arbeit systematisch aufbereitet. Einen Schwerpunkt stellt die Herleitung notwendiger und hinreichender Bedingungen für die lokale Realisierbarkeit der Multiplikation Matrix mal Vektor dar.

2 Lokale Speicherung von Vektoren

Bei der Verteilung der Knoten des FEM–Netzes auf die Prozessoren ist zu unterscheiden zwischen inneren Knoten der Teilgebiete und Koppelknoten. Innere Knoten gehören zu genau einem Teilgebiet und damit Prozessor, Koppelknoten im allgemeinen zu mehreren, im Spezialfall auch nur zu einem (Randknoten des Gesamtgebietes mit natürlichen Randbedingungen im Rahmen der Dirichlet-Dekompositionstechnik).

Es seien insgesamt m Koppelknoten und n innere Knoten, d.h. $N = m + n$, wobei Prozessor P_s genau m_s Koppelknoten und n_s innere besitze, zusammen also $N_s = m_s + n_s$. Offensichtlich gilt dann

$$n = n_1 + \ldots + n_p, \qquad m \leq m_1 + \ldots + m_p, \qquad N \leq N_1 + \ldots + N_p.$$

Vektoren $\underline{x} \in \mathsf{R}^N$, die durch ihre Knotenwerte x_i in den N Knoten des FEM–Netzes bestimmt sind, seien durch eine globale Knotennummerierung wie folgt geordnet:

$$\underline{x} = \begin{pmatrix} \underline{x}_C \\ \underline{x}_I \end{pmatrix},$$

wobei $\underline{x}_C \in \mathsf{R}^m$ der Teilvektor der Koppelknotenwerte sei und $\underline{x}_I \in \mathsf{R}^n$ der Teilvektor der Werte in den inneren Knoten, nach Prozessoren geordnet:

$$\underline{x}_I = \begin{pmatrix} \underline{x}_{I,1} \\ \vdots \\ \underline{x}_{I,p} \end{pmatrix}$$

mit $\underline{x}_{I,s} \in \mathsf{R}^{n_s}$ als Teilvektor der Werte in den inneren Knoten von Prozessor P_s. Insgesamt ist also

$$\underline{x} = \begin{pmatrix} \underline{x}_C \\ \underline{x}_{I,1} \\ \vdots \\ \underline{x}_{I,p} \end{pmatrix} \in \mathsf{R}^N. \tag{1}$$

Die Vektoren $\underline{x} \in \mathsf{R}^N$ existieren praktisch jedoch nur in auf die p Prozessoren verteilter Form, d.h. Prozessor P_s besitzt die Werte von

$$\underline{x}_s = \begin{pmatrix} \underline{x}_{C,s} \\ \underline{x}_{I,s} \end{pmatrix} \in \mathsf{R}^{N_s}, \tag{2}$$

wobei $\underline{x}_{C,s} \in \mathsf{R}^{m_s}$ und $\underline{x}_{I,s} \in \mathsf{R}^{n_s}$ die Teilvektoren der Werte in den zu P_s gehörenden Koppelknoten bzw. inneren Knoten darstellen, letztere lokal in der gleichen Reihenfolge geordnet wie global in $\underline{x}$.

Da auf Prozessor P_s in der Regel nur ein Teil der Gesamtmenge der Koppelknoten vorkommt, seien die Werte in $\underline{x}_{C,s}$ entsprechend einer lokalen Nummerierung angeordnet, in die global nummerierte Werte aus $\underline{x}_C$ mittels einer Zusammenhangsmatrix $A_{C,s} \in \mathsf{R}^{m_s \times m}$ als $A_{C,s}\, \underline{x}_C$

eingeordnet werden können. Die Matrix $A_{C,s}$ ist eine Boolesche Matrix und besitzt in jeder der m_s Zeilen genau eine Eins und sonst lauter Nullen, in jeder der m Spalten höchstens eine Eins. Die Einordnung der Werte von $\underline{x}_{C,s}$ in die globale Nummerierung erfolgt dann als $A_{C,s}^{\top}\,\underline{x}_{C,s}$.

Für die Darstellung globaler Vektoren $\underline{x} \in \mathbf{R}^N$ entsprechend (1) durch lokale Vektoren $\underline{x}_s \in \mathbf{R}^{N_s}$ entsprechend (2), $s=1(1)p$, gibt es zumindest zwei unterschiedlich zu behandelnde Möglichkeiten in den Koppelknoten: entweder besitzt jeder Prozessor die Werte von $\underline{x}$ in den ihm gehörenden Koppelknoten, oder er besitzt jeweils nur einen Anteil dieser Werte, und die Werte selbst ergeben sich durch Assemblierung über die den Knoten besitzenden Prozessoren. Letzteres ist bei verteilter Berechnung der rechten Seite im FEM–Gleichungssystem entsprechend der Gebietsdekomposition der Fall. Im weiteren werden deshalb wie in [2] folgende beiden Darstellungstypen betrachtet:

Typ I: $\underline{x}$ dargestellt durch $\underline{x}_s$, $s=1(1)p$, mit $\underline{x}_{C,s} = A_{C,s}\,\underline{x}_C$,

Typ II: $\underline{x}$ dargestellt durch $\underline{x}_s$, $s=1(1)p$, mit $\underline{x}_C = \sum\limits_{s=1}^{p} A_{C,s}^{\top}\,\underline{x}_{C,s}$.

Die Darstellung vom Typ II ist offensichtlich nicht eindeutig.

3 Eigenschaften der Zusammenhangsmatrizen

Auch für die Einordnung der Werte von $\underline{x}$ in $\underline{x}_s$ kann eine entsprechende Zusammenhangsmatrix $A_s \in \mathbf{R}^{N_s \times N}$ angegeben werden, vgl. [2].

$$A_s = \underbrace{\begin{pmatrix} A_{C,s} & \mathbf{O} & \multicolumn{3}{c}{\cdots\cdots\cdots\cdots} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{I} & \cdots & \mathbf{O} \end{pmatrix}}_{N} \begin{array}{l} \} \; m_s \\ \} \; n_s \end{array} \Bigg\} \; N_s$$

$$\underbrace{\quad}_{m}\ \underbrace{\quad}_{n_1}\ \cdots\ \underbrace{\quad}_{n_s}\ \cdots\ \underbrace{\quad}_{n_p}$$

A_s besitzt die nachfolgenden, mehr oder weniger offensichtlichen Eigenschaften.

1°. In jeder Zeile von A_s steht genau eine Eins und sonst lauter Nullen, in jeder Spalte höchstens eine Eins.

2°. Für $\underline{x} \in \mathbf{R}^N$ (vgl. (1)) folgt

$$A_s\,\underline{x} = \begin{pmatrix} A_{C,s}\,\underline{x}_C \\ \underline{x}_{I,s} \end{pmatrix}, \tag{3}$$

für $\underline{x}_s \in \mathbf{R}^{N_s}$ (vgl. (2)) folgt

$$\sum_{s=1}^{p} A_s^{\top}\underline{x}_s = \begin{pmatrix} \sum\limits_{s=1}^{p} A_{C,s}^{\top}\,\underline{x}_{C,s} \\ \underline{x}_I \end{pmatrix}. \tag{4}$$

3°. Die Darstellungstypen lassen sich mit (3) bzw. (4) auch wie folgt äquivalent definieren:

$$\begin{array}{llll} \text{Typ I:} & \underline{x} \text{ dargestellt durch } \underline{x}_s,\ s=1(1)p, & \text{mit} & \underline{x}_s = A_s\,\underline{x}, \\ \text{Typ II:} & \underline{x} \text{ dargestellt durch } \underline{x}_s,\ s=1(1)p, & \text{mit} & \underline{x} = \sum\limits_{s=1}^{p} A_s^\top\,\underline{x}_s. \end{array}$$

4°. Es gilt

$$\begin{aligned} A_{C,s}\,A_{C,s}^\top &= I \in \mathbf{R}^{m_s\times m_s}, \\ A_s\,A_s^\top &= I \in \mathbf{R}^{N_s\times N_s}. \end{aligned} \tag{5}$$

5°. Die Matrizen

$$\begin{aligned} R_{C,s} &= A_{C,s}^\top\,A_{C,s} \in \mathbf{R}^{m\times m}, \\ R_s &= A_s^\top\,A_s \in \mathbf{R}^{N\times N} \end{aligned}$$

sind Diagonalmatrizen, deren Hauptdiagonalen nur aus Einsen und Nullen bestehen und die deshalb trivialerweise symmetrisch und positiv semidefinit sind. Die Einsen stehen in den Zeilen mit den Nummern, die den zu P_s gehörenden Koppelknoten entsprechen bei $R_{C,s}$ bzw. allen zu P_s gehörenden Knoten bei R_s. Insbesondere folgen die einfachen Beziehungen

$$\begin{array}{rcl} R_{C,s}\,A_{C,s}^\top = A_{C,s}^\top & , & A_{C,s}\,R_{C,s} = A_{C,s}, \\ R_s\,A_s^\top = A_s^\top & , & A_s\,R_s = A_s. \end{array} \tag{6}$$

6°. Weiterhin ist

$$R_{C,s}{}^2 = R_{C,s}, \qquad R_s{}^2 = R_s.$$

Somit stellen die R_s im euklidischen Raum $\mathbf{R}^N$ Orthoprojektoren dar. Sie projizieren dabei jeweils genau auf den Teilraum von Vektoren $\underline{x}$, deren Koeffizienten x_i in allen nicht zu Prozessor P_s gehörenden Knoten verschwinden. Die R_s sind paarweise miteinander vertauschbar

$$R_s\,R_t = R_t\,R_s,$$

so daß auch $R_s\,R_t$ einen Orthoprojektor im $\mathbf{R}^N$ darstellt, denn

$$(R_s\,R_t)^2 = R_s\,R_t = (R_s\,R_t)^\top.$$

Er projiziert auf den Teilraum von Vektoren $\underline{x}$, deren Koeffizienten x_i in allen nicht Prozessor P_s und P_t gemeinsamen Knoten verschwinden. Die Prozessoren P_i mit $R_s\,R_i \neq \mathbf{O}$ sind also die unmittelbaren Nachbarn von Prozessor P_s.

7°. Auch folgende Matrizen sind Diagonalmatrizen:

$$\begin{aligned} R_C &= \sum_{s=1}^{p} R_{C,s} \in \mathbf{R}^{m\times m}, \\ R &= \sum_{s=1}^{p} R_s \in \mathbf{R}^{N\times N}. \end{aligned}$$

Sei $\underline{w} \in \mathbf{R}^N$ der Vektor, dessen i-te Komponente w_i die Anzahl der Prozessoren angibt, auf denen der i-te Knoten vertreten ist, wobei offenbar immer gilt

$$1 \leq w_i \leq w_{max} \leq p, \qquad \underline{w}_I = (1\,1\ldots 1)^\top.$$

Für $w_i > 2$ handelt es sich bei den Knoten um sogenannte Crosspoints. Dann ist

$$R_C = diag(\underline{w}_C), \qquad R = diag(\underline{w}),$$

und im Sinne der positiven Definitheit gilt

$$\begin{aligned} I &\leq R_C \leq w_{max}\, I \in \mathbf{R}^{m\times m}, \\ I &\leq R \leq w_{max}\, I \in \mathbf{R}^{N\times N}. \end{aligned}$$

4 Gleichheit von Vektoren, Typwechsel

Es wird nun betrachtet, wie sich die Gleichheit zweier Vektoren $\underline{x}, \underline{y} \in \mathbf{R}^N$ in der lokalen Darstellung widerspiegelt.

$\underline{x}$ und $\underline{y}$ vom Typ I: Aus $\underline{x} = \underline{y}$ ergibt sich sofort $\underline{x}_s = A_s\,\underline{x} = A_s\,\underline{y} = \underline{y}_s$, $s = 1(1)p$. Auch die Umkehrung gilt, da bei Typ I die Werte in jedem Koppelknoten auf unterschiedlichen Prozessoren übereinstimmen.

$\underline{x}$ und $\underline{y}$ vom Typ II: $\underline{x} = \underline{y}$ ist äquivalent zu

$$\sum_{s=1}^{p} A_s^\top\,\underline{x}_s = \sum_{s=1}^{p} A_s^\top\,\underline{y}_s.$$

$\underline{x}$ vom Typ I, $\underline{y}$ vom Typ II: Aus $\underline{x} = \underline{y}$ folgt mit (5), vgl. auch [2],

$$\underline{x}_s = A_s\,\underline{x} = A_s\,\underline{y} = \sum_{i=1}^{p} A_s\,A_i^\top\,\underline{y}_i = \underline{y}_s + \sum_{\substack{i=1\\ i\neq s}}^{p} A_s A_i^\top\,\underline{y}_i, \qquad s = 1(1)p. \tag{7}$$

Wie im ersten Fall gilt auch hier die Umkehrung. Die Bedingungen lassen sich auch getrennt nach Koppelknoten und inneren Knoten aufschreiben.

$\underline{x}$	$\underline{y}$	$\underline{x} = \underline{y}$ ist äquivalent zu	
I	I	$\underline{x}_s = \underline{y}_s,\quad s=1(1)p$	$\underline{x}_{C,s} = \underline{y}_{C,s},\ \underline{x}_{I,s} = \underline{y}_{I,s},\quad s=1(1)p$
II	II	$\sum_{s=1}^{p} A_s^\top\,\underline{x}_s = \sum_{s=1}^{p} A_s^\top\,\underline{y}_s$	$\sum_{s=1}^{p} A_{C,s}^\top\,\underline{x}_{C,s} = \sum_{s=1}^{p} A_{C,s}^\top\,\underline{y}_{C,s},\quad \underline{x}_{I,s} = \underline{y}_{I,s},\quad s=1(1)p$
I	II	$\underline{x}_s = \underline{y}_s + \sum_{\substack{i=1\\ i\neq s}}^{p} A_s\,A_i^\top\,\underline{y}_i,\quad s=1(1)p$	$\underline{x}_{C,s} = \underline{y}_{C,s} + \sum_{\substack{i=1\\ i\neq s}}^{p} A_{C,s}\,A_{C,i}^\top\,\underline{y}_{C,i},\quad \underline{x}_{I,s} = \underline{y}_{I,s},\quad s=1(1)p$

Übergang Typ II → Typ I

Mit Formel (7) kann auch praktisch der Übergang vom Darstellungstyp II zu Typ I realisiert werden. Er ist mit Kommunikation zwischen den unmittelbar benachbarten Prozessoren über die Werte in den Koppelknoten verbunden.

Übergang Typ I → Typ II

Der Übergang vom Typ I in die nicht eindeutige Darstellung des Typs II ist zum Beispiel dadurch möglich, daß in jedem Koppelknoten der Wert durch die Anzahl w_i der diesen Koppelknoten benutzenden Prozessoren geteilt wird:

$$\underline{y}_s = A_s \, R^{-1} \, \underline{x},$$

denn es gilt dann

$$\sum_{s=1}^{p} A_s^\top \, \underline{y}_s = \left(\sum_{s=1}^{p} A_s^\top \, A_s \right) \, R^{-1} \, \underline{x} = R \, R^{-1} \, \underline{x} = \underline{x}.$$

Sind die lokal benötigten Anzahlen w_i dem Prozessor P_s erst einmal bekannt, d.h. $\underline{w}_{C,s} = A_{C,s} \, \underline{w}_C$, so kann der Übergang immer ohne weitere Kommunikation vollzogen werden:

$$\underline{y}_s = \begin{pmatrix} A_{C,s} \, R_C^{-1} \underline{x}_C \\ \underline{x}_{I,s} \end{pmatrix} = \begin{pmatrix} (diag(\underline{w}_{C,s}))^{-1} \underline{x}_{C,s} \\ \underline{x}_{I,s} \end{pmatrix}.$$

5 Addition, Multiplikation mit Skalar

Leicht nachzuprüfen ist, daß sich die Grundoperationen Addition $\underline{x} + \underline{y}$ und Multiplikation von $\underline{x}$ mit einem Skalar λ innerhalb ein und desselben Typs problemlos lokal, also ohne Kommunikation, realisieren lassen. Dazu sind nur die entsprechenden lokalen Darstellungen $\underline{x}_s, \underline{y}_s$ zu addieren bzw. $\underline{x}_s$ ist mit dem Skalar λ zu multiplizieren. Allerdings muß der Skalar allen Prozessoren bekannt sein.

6 Skalarprodukt

Typ von $\underline{x}$ und $\underline{y}$ unterschiedlich

In [2] wird gezeigt, daß sich das Skalarprodukt $(\underline{x}, \underline{y})$ in $\mathbf{R}^N$ besonders einfach realisieren läßt, wenn $\underline{x}$ und $\underline{y}$ von unterschiedlichem Typ sind, z.B. $\underline{x}$ vom Typ I und $\underline{y}$ vom Typ II. Dann gilt

$$(\underline{x}, \underline{y})_{\mathbf{R}^N} \;\; = \;\; \underline{x}^\top \underline{y} = \underline{x}^\top \sum_{s=1}^{p} A_s^\top \, \underline{y}_s = \sum_{s=1}^{p} (A_s \, \underline{x})^\top \underline{y}_s = \sum_{s=1}^{p} \underline{x}_s^\top \underline{y}_s,$$

also

$$(\underline{x}, \underline{y})_{\mathsf{R}^N} = \sum_{s=1}^{p} (\underline{x}_s, \underline{y}_s)_{\mathsf{R}^{N_s}}. \tag{8}$$

Das Skalarprodukt ist also durch lokale Skalarproduktbildung auf jedem Prozessor und Summierung dieser Skalarprodukte über die Prozessoren realisierbar, nur letztere erfordert eine geringe Kommunikation.

Typ von $\underline{x}$ und $\underline{y}$ gleich

Bei gleichen Darstellungstypen von $\underline{x}$ und $\underline{y}$ ist zusätzlich für einen der beiden Vektoren die Überführung in den anderen Darstellungstyp notwendig. Sind $\underline{x}$ und $\underline{y}$ vom Typ II, so ist dafür Kommunikation zwischen den unmittelbar benachbarten Prozessoren nötig, sind beide vom Typ I, ist dies nicht der Fall.

Zunächst scheint beliebig festlegbar, für welchen der an einem Skalarprodukt beteiligten Vektoren der Darstellungstyp I bzw. II gewählt wird. Im Rahmen der Assemblierungstechnik der FEM bei Gebietsdekomposition liegen jedoch die Funktionale erzeugenden Vektoren der rechten Seiten bereits auf natürliche Weise in Form von Typ II vor.

7 Produkt Matrix mal Vektor

Bei der Multiplikation Matrix mal Vektor ist davon auszugehen, daß nicht nur die Vektoren, sondern erst recht auch die Matrix in einer auf die Prozessoren verteilten Form vorliegt, etwa in Gestalt lokaler Matrizen.

Es wird nun untersucht, unter welchen Bedingungen sich die Multiplikation $\underline{y} = K\underline{x}$ einer Matrix $K \in \mathsf{R}^{N\times N}$ mit einem Vektor $\underline{x} \in \mathsf{R}^N$ lokal auf den Prozessoren realisieren läßt, ohne irgendwelche Kommunikation, und zwar im folgenden Sinn. Seien $A, B \in \{I, II\}$.

Definition: *Die Multiplikation $\underline{y} = K\underline{x}$ heißt lokal realisierbar von Typ A nach Typ B, wenn Matrizen $K_s \in \mathsf{R}^{N_s\times N_s}$ existieren, $s=1(1)p$, so daß für beliebige $\underline{x} \in \mathsf{R}^N$ gilt $\underline{y}_s = K_s\,\underline{x}_s$, $s=1(1)p$, sobald $\{\underline{x}_s\}_{s=1}^p$ die Darstellung von $\underline{x}$ vom Typ A ist und $\{\underline{y}_s\}_{s=1}^p$ die Darstellung von $\underline{y}$ vom Typ B.*

Satz: *Sei $K \in \mathsf{R}^{N\times N}$. Die Multiplikation $\underline{y} = K\underline{x}$ ist genau dann lokal realisierbar, wenn entsprechend den vier möglichen Fällen die in nachfolgender Tabelle angegebenen Bedingungen gelten. In den Fällen 2. bis 4. sind die K_s eindeutig bestimmt:*

$$K_s = A_s\,K A_s^{\top}, \qquad s=1(1)p. \tag{9}$$

Fall	$\underline{x}$	$\underline{y}$	Bedingung
1.	I	II	$\exists K_s \in \mathbf{R}^{N_s \times N_s},\ \ s=1(1)p: \quad K = \sum\limits_{s=1}^{p} A_s^\top K_s A_s$
2.	I	I	$R_s K (I - R_s) = \mathbf{O},\ \ s=1(1)p \qquad (\text{bzw.} \quad R_s K = R_s K R_s)$
3.	II	II	$(I - R_s) K R_s = \mathbf{O},\ \ s=1(1)p \qquad (\text{bzw.} \quad K R_s = R_s K R_s)$
4.	II	I	$R_s K R_i = \mathbf{O}, \qquad s, i = 1(1)p,\ s \neq i.$

Bemerkung 1: Die Bedingungen des Satzes bedeuten folgendes:

1. Die Matrix K muß eine vorgegebene Struktur haben, die durch Assemblierung der lokalen Matrizen auf den Prozessoren zu einer Gesamtmatrix erzeugt wird. Diese Struktur ist für die Matrizen der FEM-Technik typisch und stellt deshalb keine Einschränkung dar. Eine Matrix dieser Gestalt werde im weiteren als *Matrix vom Assemblierungstyp* bezeichnet.

2. Die Werte von $K\underline{x}$ in den Knoten von P_s sind von den Werten von $\underline{x}$ außerhalb von P_s unabhängig („außen“ bez. des Prozessors wirkt nicht nach „innen“).

3. Die Werte von $\underline{x}$ in den Knoten von Prozessor P_s beeinflussen nur die Werte von $K\underline{x}$ in den Knoten von P_s („innen“ wirkt nicht nach „außen“).

4. Die Werte von $\underline{x}$ in den Knoten, die auch einem fremden Prozessor P_i gehören (d.h. Werte in Koppelknoten), wirken nicht auf Werte von $K\underline{x}$ in Knoten, die Prozessor P_s gehören. Diese Bedingung ist besonders hart. Sobald ein Prozessor nur Koppelknoten besitzt, die mindestens zwei Prozessoren gemeinsam sind, bestehen die zu diesen Koppelknoten gehörenden Zeilen und Spalten von K nur aus Nullen, die Matrix muß also auf jeden Fall singulär sein. Demzufolge kann in dieser Koppelknoten–Situation eine Multiplikation $K\underline{x}$ von Typ II nach Typ I mit regulärer Matrix K nicht rein lokal vollzogen werden, sondern nur durch Zurückführung auf einen der anderen drei Fälle lokaler Realisierbarkeit und zusätzlichen Typwechsel. In jedem Falle ist dabei ein Typübergang von II nach I notwendig, der Kommunikation der Prozessoren erfordert.

Die vier Fälle lokaler Realisierbarkeit sind offenbar nicht disjunkt.

Bemerkung 2: Ist K wie im Fall *1.* vom Assemblierungstyp

$$K = \sum_{s=1}^{p} A_s^\top K_s A_s,$$

so ist die Darstellung durch lokale Matrizen K_s, $s=1(1)p$, nicht eindeutig. Wird die globale Matrix K entsprechend den globalen Koppelknoten und inneren Knoten in eine Blockstruktur zerlegt

$$K = \begin{pmatrix} K_C & K_{CI} \\ K_{IC} & K_I \end{pmatrix}$$

mit $K_C \in \mathbf{R}^{m\times m}$, $K_{CI} \in \mathbf{R}^{m\times n}$, $K_{IC} \in \mathbf{R}^{n\times m}$, $K_I \in \mathbf{R}^{n\times n}$, und analog die lokalen Matrizen entsprechend den lokalen Knoten

$$K_s = \begin{pmatrix} K_{C,s} & K_{CI,s} \\ K_{IC,s} & K_{I,s} \end{pmatrix} \tag{10}$$

mit $K_{C,s} \in \mathbf{R}^{m_s\times m_s}$, $K_{CI,s} \in \mathbf{R}^{m_s\times n_s}$, $K_{IC,s} \in \mathbf{R}^{n_s\times m_s}$, $K_{I,s} \in \mathbf{R}^{n_s\times n_s}$, so liefert einfaches Ausrechnen, vgl. [2], die Darstellung

$$\begin{aligned} K_C &= \sum_{s=1}^{p} A_{C,s}^\top K_{C,s} A_{C,s}, \\ K_{CI} &= \left(A_{C,1}^\top K_{CI,1} \ \ldots \ A_{C,p}^\top K_{CI,p} \right), \\ K_{IC} &= \left(K_{IC,1} A_{C,1} \ \ldots \ K_{IC,p} A_{C,p} \right), \\ K_I &= diag\left(K_{I,1} \ \ldots \ K_{I,p} \right). \end{aligned}$$

Gibt es eine weitere Darstellung von K durch lokale Matrizen $\hat{K}_s$ mit analoger Blockstruktur

$$K = \sum_{s=1}^{p} A_s^\top \hat{K}_s A_s,$$

so folgt aus einem Vergleich der beiden Darstellungen

$$\sum_{s=1}^{p} A_{C,s}^\top \hat{K}_{C,s} A_{C,s} = \sum_{s=1}^{p} A_{C,s}^\top K_{C,s} A_{C,s}, \tag{11}$$

$$A_{C,s}^\top \hat{K}_{CI,s} = A_{C,s}^\top K_{CI,s}, \quad \hat{K}_{IC,s} A_{C,s} = K_{IC,s} A_{C,s}, \quad \hat{K}_{I,s} = K_{I,s}, \quad s=1(1)p.$$

Durch Multiplikation mit $A_{C,s}$ von links bzw. $A_{C,s}^\top$ von rechts ergibt sich in der letzten Formelzeile

$$\hat{K}_{CI,s} = K_{CI,s}, \qquad \hat{K}_{IC,s} = K_{IC,s} \qquad s=1(1)p.$$

Somit ist die Darstellung von K durch lokale Matrizen K_s in den Blöcken K_I, K_{IC}, K_{CI} eindeutig, für den Block K_C gilt die Bedingung (11), der die unterschiedlichen Darstellungen gehorchen müssen. Hauptdiagonalelemente von K in Koppelknoten bzw. Nebendiagonalelemente für Koppelknotenpaare können also mit einer gewissen Willkür aus lokalen Anteilen gebildet worden sein.

Bemerkung 3: Für den Aufwand der Multiplikation kann neben der lokalen Realisierbarkeit weiter von Bedeutung sein, ob die K_s auf den jeweiligen Prozessoren unmittelbar zur Verfügung stehen bzw. lokal erzeugt werden können, oder ob für ihre Erzeugung ein zusätzlicher Informationsaustausch zwischen den Prozessoren notwendig ist. Bei häufiger Nutzung ein und derselben Matrix tritt diese Frage allerdings in den Hintergrund.

Folgerungen:

1. *Für die lokale Realisierbarkeit von $K\underline{x}$ und $K^\top \underline{x}$ gelten die folgenden Äquivalenzen:*

$K\underline{x}$ lokal realisierbar			$\Longleftrightarrow$	$K^\top\underline{x}$ lokal realisierbar		
Typ I	$\to$	Typ II		Typ I	$\to$	Typ II
Typ I	$\to$	Typ I		Typ II	$\to$	Typ II
Typ II	$\to$	Typ II		Typ I	$\to$	Typ I
Typ II	$\to$	Typ I		Typ II	$\to$	Typ I

Dabei trifft immer $(K^\top)_s = (K_s)^\top$ zu.

Beweis: Dieser ist mittels des Satzes und $R_s^\top = R_s$ elementar, zum Beispiel ist für den Fall *2.* $R_s K = R_s K R_s$ äquivalent zu $(R_s K)^\top = (R_s K R_s)^\top$, das heißt $K^\top R_s = R_s K^\top R_s$, außerdem gilt $(K^\top)_s = A_s K^\top A_s^\top = (A_s K A_s^\top)^\top = (K_s)^\top$.

2. *$K\underline{x}$ ist lokal realisierbar von Typ I nach Typ I* **und** *von Typ II nach Typ II genau dann, wenn gilt*

$$R_s K = K R_s, \qquad s = 1(1)p. \tag{12}$$

Beweis: Aus den notwendigen Bedingungen der Fälle *2.* und *3.* folgt sofort (12). Umgekehrt ist mit (12) $\quad R_s K = R_s{}^2 K = R_s(R_s K) = R_s K R_s$.

3. *Wenn $K \in \mathbf{R}^{N\times N}$ vom Assemblierungstyp ist und die $\underline{x}$, $\underline{y} \in \mathbf{R}^N$ vom Typ I sind, so gilt*

$$(K\underline{x}, \underline{y})_{\mathbf{R}^N} = \sum_{s=1}^{p} (K_s \underline{x}_s, \underline{y}_s)_{\mathbf{R}^{N_s}}.$$

Beweis: $K\underline{x}$ ist vom Typ II mit $(K\underline{x})_s = K_s \underline{x}_s$ und es gilt (8) mit $K\underline{x}$ anstelle von $\underline{x}$.

Bemerkung 4: Ist K im Satz eine rechteckige Matrix, so ergeben sich keine wesentlichen Änderungen. $\underline{x}$ und $\underline{y}$ gehören dann unterschiedlichen Räumen an, und es ist nur zu berücksichtigen, daß die Typdefinitionen für $\underline{x}$ und $\underline{y}$ mit unterschiedlichen Zusammenhangsmatrizen erfolgen. Entsprechend sind auch die zugehörigen Projektoren zu unterscheiden.

Beweis des Satzes:

1. Sei $\underline{x} \in \mathbf{R}^N$ beliebig und vom Typ I, $\underline{y} = K\underline{x}$ vom Typ II. Dann gilt bei lokaler Realisierbarkeit von $K\underline{x}$

$$K\underline{x} = \underline{y} = \sum_{s=1}^{p} A_s^\top \underline{y}_s = \sum_{s=1}^{p} A_s^\top K_s \underline{x}_s = \left(\sum_{s=1}^{p} A_s^\top K_s A_s\right) \underline{x},$$

wegen der Beliebigkeit von $\underline{x}$ folgt also

$$K = \sum_{s=1}^{p} A_s^\top K_s A_s.$$

Hat andererseits K eben diese Gestalt und werden lokal die Vektoren $\underline{y}_s = K_s \underline{x}_s$ berechnet, so folgt umgekehrt

$$\sum_{s=1}^{p} A_s^\top \underline{y}_s = \sum_{s=1}^{p} A_s^\top K_s \underline{x}_s = \left(\sum_{s=1}^{p} A_s^\top K_s A_s \right) \underline{x} = K\underline{x} = \underline{y},$$

die $\underline{y}_s$ realisieren also gerade die Darstellung vom Typ II für $\underline{y}$.

2. Sei $\underline{x} \in \mathbf{R}^N$ beliebig, $\underline{y} = K\underline{x}$, $\underline{x}$ und $\underline{y}$ vom Typ I. Dann gilt bei lokaler Realisierbarkeit von $K\underline{x}$

$$A_s K\underline{x} = A_s \underline{y} = \underline{y}_s = K_s \underline{x}_s = K_s A_s \underline{x},$$

wegen der Beliebigkeit von $\underline{x}$ und $A_s A_s^\top = I$ folgt

$$A_s K = K_s A_s, \tag{13}$$
$$K_s = K_s A_s A_s^\top = A_s K A_s^\top. \tag{14}$$

Wird (14) in (13) eingesetzt und von links mit $A_s^\top$ multipliziert, ergibt sich schließlich die Bedingung von *2.*

$$R_s K = A_s^\top A_s K = A_s^\top (A_s K A_s^\top) A_s = R_s K R_s.$$

Gilt umgekehrt $R_s K = R_s K R_s$, $s = 1(1)p$, und werden mit $K_s = A_s K A_s^\top$ die Vektoren $\underline{y}_s = K_s \underline{x}_s$ berechnet, so folgt für diese mit (6)

$$\underline{y}_s = A_s K A_s^\top \underline{x}_s = A_s R_s K A_s^\top (A_s \underline{x}) = A_s R_s K R_s \underline{x} = A_s R_s K \underline{x} = A_s K \underline{x} = A_s \underline{y},$$

die $\underline{y}_s$ realisieren also gerade die Darstellung vom Typ I für $\underline{y}$.

3. Sei $\underline{x} \in \mathbf{R}^N$ beliebig, $\underline{y} = K\underline{x}$, $\underline{x}$ und $\underline{y}$ vom Typ II. Beliebigkeit von $\underline{x}$ bedeutet insbesondere Beliebigkeit der lokalen Darstellungen $\underline{x}_s \in \mathbf{R}^{N_s}$, $s = 1(1)p$. Dann gilt bei lokaler Realisierbarkeit von $K\underline{x}$

$$\sum_{s=1}^{p} K A_s^\top \underline{x}_s = K \sum_{s=1}^{p} A_s^\top \underline{x}_s = K\underline{x} = \underline{y} = \sum_{s=1}^{p} A_s^\top \underline{y}_s = \sum_{s=1}^{p} A_s^\top K_s \underline{x}_s,$$

wegen der Beliebigkeit von $\underline{x}_s$ und $A_s A_s^\top = I$ folgt

$$K A_s^\top = A_s^\top K_s, \tag{15}$$
$$K_s = A_s A_s^\top K_s = A_s K A_s^\top. \tag{16}$$

Wird (16) in (15) eingesetzt und von rechts mit A_s multipliziert, ergibt sich die Bedingung von *3.*

$$K R_s = K A_s^\top A_s = A_s^\top (A_s K A_s^\top) A_s = R_s K R_s.$$

Gilt umgekehrt $K\,R_s = R_s\,K\,R_s$, $s=1(1)p$, und werden mit $K_s = A_s\,K\,A_s^\top$ die Vektoren $\underline{y}_s = K_s\,\underline{x}_s$ berechnet, so folgt für diese mit (6)

$$\begin{aligned}\sum_{s=1}^{p} A_s^\top\,\underline{y}_s &= \sum_{s=1}^{p} A_s^\top(A_s\,K\,A_s^\top)\,\underline{x}_s = \sum_{s=1}^{p} R_s\,K\,R_s\,A_s^\top\,\underline{x}_s \\ &= \sum_{s=1}^{p} K\,R_s\,A_s^\top\,\underline{x}_s = K\sum_{s=1}^{p} A_s^\top\,\underline{x}_s = K\underline{x} = \underline{y},\end{aligned}$$

die $\underline{y}_s$ realisieren also gerade die Darstellung vom Typ II für $\underline{y}$.

4. Sei $\underline{x} \in \mathbf{R}^N$ beliebig und vom Typ II, das heißt insbesondere die lokale Darstellung $\underline{x}_s \in \mathbf{R}^{N_s}$, $s=1(1)p$, sei beliebig. Sei weiterhin $\underline{y} = K\underline{x}$ und $\underline{y}$ vom Typ I. Dann gilt bei lokaler Realisierbarkeit von $K\underline{x}$

$$K_s\,\underline{x}_s = \underline{y}_s = A_s\,\underline{y} = A_s\,K\,\underline{x} = A_s\,K\sum_{i=1}^{p} A_i^\top\,\underline{x}_i. \tag{17}$$

Wegen der Beliebigkeit der $\underline{x}_i$ ist für $\underline{x}_i = \underline{0}$, $i \neq s$

$$\begin{aligned}K_s\,\underline{x}_s &= A_s\,K\,A_s^\top\,\underline{x}_s, \\ K_s &= A_s\,K\,A_s^\top.\end{aligned}$$

Eingesetzt in (17) liefert dies

$$\begin{aligned}A_s\,K\,A_s^\top\,\underline{x}_s &= A_s\,K\sum_{i=1}^{p} A_i^\top\,\underline{x}_i, \\ \sum_{\substack{i=1\\ i\neq s}}^{p} A_s\,K\,A_i^\top\,\underline{x}_i &= \underline{0}\,.\end{aligned}$$

Wegen der Beliebigkeit der $\underline{x}_i$ führt dies auf

$$A_s\,K\,A_i^\top = \mathbf{O}, \quad s \neq i,$$

bzw. nach Multiplikation mit $A_s^\top$ von links und A_i von rechts auf

$$R_s\,K\,R_i = \mathbf{O}, \quad s \neq i.$$

Gilt umgekehrt letztere Gleichung für $s \neq i$ und werden mit $K_s = A_s\,K\,A_s^\top$ die Vektoren $\underline{y}_s = K_s\,\underline{x}_s$ berechnet, so folgt für diese

$$\begin{aligned}\underline{y}_s &= A_s\,K\,A_s^\top\,\underline{x}_s = A_s\,R_s\,K\,R_s\,A_s^\top\,\underline{x}_s = \sum_{i=1}^{p} A_s\,R_s\,K\,R_i\,A_i^\top\,\underline{x}_i \\ &= A_s\,K\sum_{i=1}^{p} A_i^\top\,\underline{x}_i = A_s\,K\underline{x} = A_s\underline{y},\end{aligned}$$

die $\underline{y}_s$ realisieren also gerade die Darstellung vom Typ I für $\underline{y}$.

Die eindeutige Gestalt (9) folgt in den Fällen *2.* bis *4.* notwendig aus dem jeweils ersten Teil des Beweises. □

Beispiele:

1. *Diagonalmatrizen* $K = diag(\underline{k})$ erfüllen generell die Bedingungen des Satzes für die Fälle *1.* bis *3.* Einerseits gilt für die Fälle *2.* und *3.* immer $R_s\,K = K\,R_s$, vgl. Folgerung 2., da auch die R_s Diagonalmatrizen sind. Wird $\underline{k}$ als Vektor vom Typ I dargestellt, das heißt $\underline{k}_s = A_s\,\underline{k}$, so folgt insbesondere

$$K_s = A_s\,K\,A_s^\top = diag(\underline{k}_s), \qquad s=1(1)p.$$

Andererseits ist K auch entsprechend Fall *1.* vom Assemblierungstyp. Wird nämlich $\underline{k}$ als Vektor vom Typ II dargestellt, d.h. $\underline{k} = \sum\limits_{s=1}^{p} A_s^\top\,\underline{k}_s$, und $K_s = diag(\underline{k}_s)$ gewählt, so ergibt sich

$$\begin{aligned}\sum_{s=1}^{p} A_s^\top\,K_s\,A_s &= \sum_{s=1}^{p} A_s^\top\,diag\,(\underline{k}_s)\;A_s = \sum_{s=1}^{p} diag\left(A_s^\top\,\underline{k}_s\right)\\ &= diag\left(\sum_{s=1}^{p} A_s^\top\,\underline{k}_s\right) = diag(\underline{k}) = K.\end{aligned}$$

Im Fall *4.* bedeutet die Bedingung $R_s\,K\,R_i = \mathbf{O},\ s \neq i$, daß Diagonalelemente von K zu solchen Koppelknoten gleich null sein müssen, die zu mindestens zwei Prozessoren gehören.

2. *FEM-Matrizen:* Wie bereits bemerkt, führt bei der parallelen Realisierung der FEM mittels Gebietsdekompositionsmethoden die Assemblierungstechnik im Falle linearer Probleme automatisch auf eine Systemmatrix K vom Assemblierungstyp, d.h.

$$K = \sum_{s=1}^{p} A_s^\top\,K_s\,A_s,$$

so daß bei Multiplikationen $\underline{y} = K\underline{x}$ mit dieser Matrix im Rahmen von Iterationsverfahren von vornherein die Nutzung von Typ I für $\underline{x}$ und Typ II für $\underline{y}$ das Nächstliegende ist, da die K_s nach lokaler Assemblierung sofort lokal zur Verfügung stehen und dann auch für die Multiplikation keine Kommunikation nötig ist, vgl. zum Beispiel [2], [1].

Jede der drei anderen Varianten würde für die lokale Realisierbarkeit der Multiplikation Einschränkungen der Allgemeinheit von K entsprechend den Bedingungen des Satzes zufolge haben. Dazu käme die Berechnung der lokalen Matrizen $\hat{K}_s = A_s\,K\,A_s^\top$, die sich von den K_s im allgemeinen unterscheiden

$$\hat{K}_s = A_s\left(\sum_{i=1}^{p} A_i^\top\,K_i\,A_i\right)\,A_s^\top = K_s + \sum_{\substack{i=1\\ i\neq s}}^{p} A_s\,A_i^\top\,K_i\,A_i\,A_s^\top$$

und zur Bereitstellung wiederum Kommunikation erfordern.

3. *Vorkonditionierung von FEM-Matrizen:* Es sei ein lineares Gleichungssystem mit der Systemmatrix K vom Assemblierungstyp iterativ zu lösen, wobei die Multiplikation mit K wie oben von Typ I nach Typ II lokal realisiert werde, zum Beispiel bei der Berechnung der Defekte der Iterierten. Die Iterierten sind dazu vom Typ I, die Defekte vom Typ II. Wird eine Vorkonditionierung des Systems mit einer Matrix $C \approx K$

vorgenommen, so ist durch Anwendung von C^{-1} auf die Defekte eine geeignete Korrektur für die Iterierten zu berechnen, die Multiplikation mit C^{-1} muß also von Typ II nach Typ I erfolgen.

Bei Verwendung der hierarchischen Vorkonditionierung von Yserentant [3]

$$C^{-1} = V\,V^\top \quad \text{mit} \quad V = V^{(J)}\,V^{(J-1)}\ldots V^{(1)}$$

hat die Multiplikation mit $V^{(k)}$ die Eigenschaft, daß nur die Werte in den Knoten des k-ten Levels eines hierarchischen Netzes verändert werden, und zwar nur in Abhängigkeit von den Werten in jeweils zwei Vaterknoten aus dem $(k-1)$-ten Level, die bei einer dem Grobnetz des 0-ten Levels entsprechenden Gebietsdekomposition mit zum gleichen Prozessor P_s gehören. Werte, die anderen Prozessoren, aber nicht P_s gehören, haben also keinen Einfluß, werden aber beeinflußt, und zwar in den Knoten des k-ten Levels, die in unmittelbarer Nachbarschaft der Koppelknoten des $(k-1)$-ten Levels von Prozessor P_s liegen:

$$R_s\,V^{(k)} = R_s\,V^{(k)}\,R_s \neq V^{(k)}\,R_s, \qquad s=1(1)p.$$

Somit folgt, daß die $V^{(k)}$ von Typ I nach Typ I lokal realisierbar sind, nicht aber von Typ II nach Typ II. Umgekehrt ist $V^{(k)\top}$ lokal von Typ II nach Typ II realisierbar, nicht aber von Typ I nach Typ I. Analoges gilt auch für V und $V^\top$. Die Multiplikation eines Vektors vom Typ II mit

$$V\,V^\top = V^{(J)}\,V^{(J-1)}\ldots V^{(1)}\,V^{(1)\top}\ldots V^{(J-1)\top}\,V^{(J)\top}$$

erfordert somit genau zwischen $V^{(1)\top}$ und $V^{(1)}$ einen zusätzlichen Übergang von Typ II zu Typ I.

4. Matrizenprodukt $K^\top K$: Das vorige Beispiel ist nur ein Spezialfall der Berechnung von $\underline{y} = K^\top K\underline{x}$. Aus Folgerung 1. ergibt sich dafür die folgende allgemeine Situation. Ist $\underline{z} = K\underline{x}$ in einem der vier möglichen Fälle lokal realisierbar, so ist $\underline{y} = K^\top\underline{z}$ anschließend garantiert auch lokal realisierbar. Die Garantie bezieht sich dabei aber immer nur auf einen Fall, der einen vorherigen Typwechsel von $\underline{z}$ erfordert. In den Fällen *2.* und *4.* des Satzes ist dieser Typwechsel unproblematisch, weil von I nach II und damit im wesentlichen ohne Kommunikation ausführbar.

Ist die Multiplikation mit $K^\top K$ häufig auszuführen, so entsteht die Frage, ob nicht zur Vermeidung der ständigen doppelten Multiplikation, ggf. unter Einsatz von Kommunikation, lokale Matrizen berechnet werden sollten, die $K^\top K$ repräsentieren und die Multiplikation mit $K^\top K$ lokal realisieren, ggf. mit weiterer Kommunikation. Prinzipiell tritt dabei aber folgendes Problem auf. Im allgemeinen werden Koppelknotenwerte von $\underline{z}$ durch Werte von $\underline{x}$ in inneren Knoten aller den Koppelknoten besitzenden Prozessoren beeinflußt. Diese Wirkung wird dann an $\underline{y}$ weitergegeben, so daß $\underline{y}_s$ auch von Werten von $\underline{x}$ in inneren Knoten der Nachbarprozessoren abhängt. Die somit notwendige Kommunikation würde das Prinzip der Parallelisierung durch nichtüberlappende Gebietsdekomposition verletzen, Prozessorkommunikation nur über Koppelknotenwerte auszuführen. Speziell für K vom Assemblierungstyp, also auch $K^\top$, sowie

$\underline{x}$ vom Typ I, $\underline{y}$ vom Typ II, erhält man durch Ausrechnen

$$\underline{y}_s \;=\; K_s^\top A_s \sum_{i=1}^{p} A_i^\top K_i\,\underline{x}_i \;=\; K_s^\top K_s\,\underline{x}_s \;+\; \sum_{\substack{i=1\\ i\neq s}}^{p} (K_s^\top A_s A_i^\top K_i)\,\underline{x}_i .$$

5. Im Rahmen der Projektionstechnik zur Berechnung inkompressibler Strömungen ist bei der iterativen Lösung der Gleichung für die Druckkorrektur immer wieder

$$\underline{y} = B^\top D B\,\underline{x}$$

mit einer rechteckigen Matrix B vom Assemblierungstyp und einer Diagonalmatrix D zu berechnen. Unter Berücksichtigung von Bemerkung 4. und Beispiel 1. lassen sich hierauf die Überlegungen des vorangegangenen Beispiels übertragen.

Literatur

[1] Haase, G., Langer, U., Meyer, A.: Parallelisierung und Vorkonditionierung des CG-Verfahrens durch Gebietszerlegung. In: Bader, Rannacher, Wittum (Hrsg.), Numerische Algorithmen auf Transputersystemen, B. G. Teubner Stuttgart 1993.

[2] Meyer, A.: A parallel preconditioned conjugate gradient method using domain decomposition and inexact solvers on each subdomain. Computing, *45*, 217–234 (1990).

[3] Yserentant, H.: On the multilevel-splitting of finite element spaces. Numer. Math., *49*, 379–412 (1986).

Adresse des Autors:

Dr. Ulrich Groh, Technische Universität Chemnitz-Zwickau, Fakultät für Mathematik, Postfach 964, 09009 Chemnitz