

Arbeitsstreffen „Adaptive Multilevel-FEM“
Berlin, 29. – 31.5.1996

Das Programm SPC-PM Po 3D
zur Lösung von Potentialproblemen
auf MIMD-Parallelrechnern

Thomas Apel
Michael Jung
Arnd Meyer
Frank Milde
Matthias Pester
Michael Theß
u.a.

Inhalt:

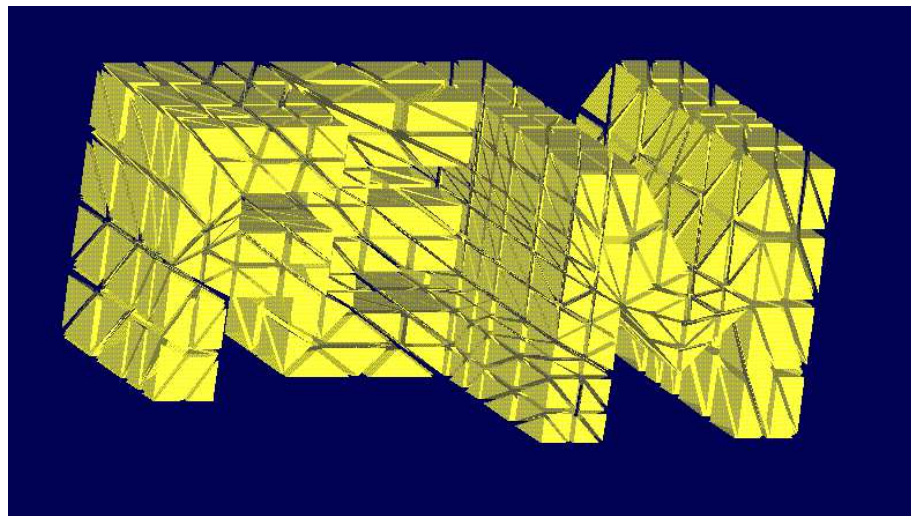
1. Allgemeines zu SPC-PM Po 3D
2. Realisierung der Kommunikation
3. Verteilung des Gebietes auf die Prozessoren
4. Lastbalancierung bei Adaptivität (Ideen)

Programm-Modul SPC-PM Po 3D

Probleme: Poisson-Gleichung und Lamé-System (3D);

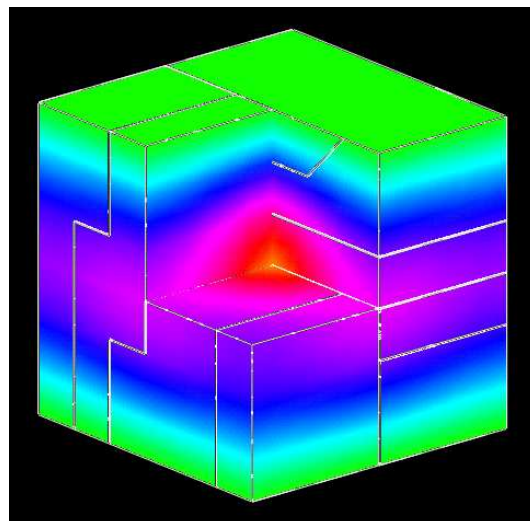
Elementtypen: 4- und 10-Knoten-Tetraeder,
8-, 20- und 27-Knoten-Hexaeder;

Netzgenerierung: Grobnetzgenerierung durch Spezialprogramme unterschiedlicher Allgemeinheit,
hierarchische Netzverfeinerung;



Löser: CG mit Vorkonditionierung (Yserentant, BPX),
Multigrid;

Postprocessing: Berechnung von Fehlernormen bei vorhandener exakter Lösung (Fehlerschätzer in Vorbereitung),
Visualisierung mit GRAPE.



Parallelisierung durch Gebietszerlegung

Lastbalance,
schnelle Algorithmen,
wenig Kommunikation



Kommunikation

Wesentliches Merkmal des parallelen Rechnens:
Minimierung der Kommunikation

Auftreten von Kommunikation:

- Verteilung des Netzes auf die Prozessoren
- Setzen von Dirichlet-Randbedingungen
- Iterativer Löser (PCG, MG) – **Effektivität wesentlich**
- Fehlerschätzer
- Grafikausgabe

Parallelisierung durch Gebietszerlegung

$$\text{Gebietszerlegung: } \bar{\Omega} = \bigcup_{s=1}^p \bar{\Omega}_s$$

2 Typen von Vektoren:

- Vektoren mit Funktionswerten (z.B. u, w):

$$u_s = H_s u,$$

- Vektoren mit Funktionalwerten (z.B. f, r):

$$f = \sum_{s=1}^p H_s^T f_s.$$

Pro CG-Iteration gibt es folgende Kommunikation:

- 2 Skalarprodukte von Vektoren verschiedenen Typs

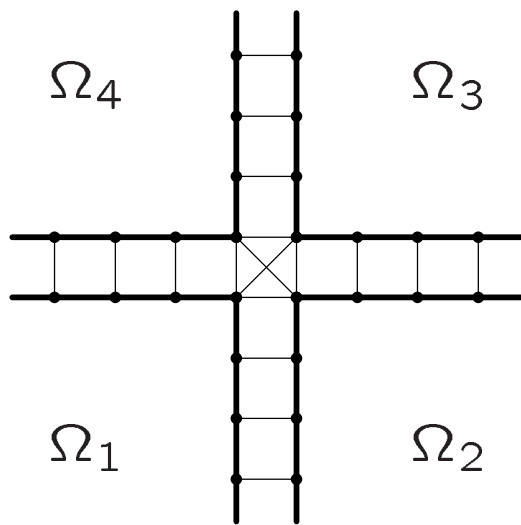
$$w^T r = w^T \sum_{s=1}^p H_s^T r_s = \sum_{s=1}^p (H_s w)^T r_s = \sum_{s=1}^p w_s^T r_s$$

⇒ jeweils nur eine globale Summe zu bilden.

- Typumwandlung bei $w = C^{-1}r$:

Koppelrandkommunikation (aufwendig!)

Koppelrandkommunikation 2D



Eckknoten der Teilgebiete
(Crosspoints):

- nur $\mathcal{O}(1)$ viele,
- globale Summe.

Alle anderen Koppelknoten liegen in genau 2 Teilgebieten:

- $\mathcal{O}(h^{-1})$ viele Knoten.
- Um Sortieraufwand zu sparen, seien sie auf beiden Prozessoren in der gleichen Reihenfolge angeordnet:

Kette.

Eine Kette wird lokal beschrieben durch:

- Startpointer
- Länge
- Information über den Zielprozessor
- Anfangs-/Endpunkt in globaler Numerierung

Akkumulation besteht aus:

- Kopiere Beschreibungsdaten und Werte der Kettenpunkte in Sendefeld.
- Sende/empfange paarweise.
- Addiere auf die lokale Kette.

Koppelrandkommunikation 3D

Besonderheit: 1D- und 2D-Ketten

2D-Ketten:

- $\mathcal{O}(h^{-2})$ viele Knoten,
- ansonsten wie die Ketten im 2D-Fall (Paarbildung etc.)

1D-Ketten:

- $\mathcal{O}(h^{-1})$ viele Knoten,
- diese gehören i.allg. zu mehreren Prozessoren
- Routine, die auf Denkmodell Hypercube basiert, aber Ketten nur in Teilen des Cubes versendet [Apel/Haase/Meyer/Pester:95].
- 2D-Ketten können genauso behandelt werden.

Länge der Datenströme hängt von der Zuordnung der Teilgebiete auf die Prozessoren ab.

Beispielrechnung

- 1: Verarbeiten der 2D-Ketten durch Paarbildung,
- 2: Gleichbehandlung aller Ketten,
- 3: wie 2, aber Protokollieren während der 1. Iteration das Auswerten der empfangenen Daten
⇒ weniger Sortieraufwand.

Würfelgebiet, Poissonproblem, BPX

Nutzernetz: 192 Elemente, lineare Verteilung

Zeit für PCG, 16 Knoten auf GCPP-128:

Level	Elemente	Knoten	Var. 1	Var. 2	Var. 3
1	1536	369	0.31	0.33	0.32
2	12288	2465	0.62	0.52	0.47
3	98304	17985	1.27	1.07	0.92
4	768432	137345	3.70	3.48	3.14

Gleiches Beispiel, 128 Prozessoren auf GCEI-192:

Level	Elemente	Knoten	Var. 1	Var. 2	Var. 3
1	1536	369	2.73	2.90	2.43
2	12288	2465	6.36	7.99	5.53
3	98304	17985	12.76	18.53	11.40
4	768432	137345	29.05	41.47	27.82

Verteilung des Nutzernetzes auf die Prozessoren

Ziele:

1. Lastbalance,
2. wenig Koppelflächen,
3. kurze Wege im Hypercube für die Ketten.

1. und 2. sind eine Frage der Partitionierung,
3. hängt von der Zuordnung ab.

Partitionierung:

- linear,
 - Spektrale Bisektion.
-

Zuordnung einbeziehen:

- Oktasektion,
- Nachbearbeitung der Partitionierung.

Tests zur Partitionierung

Würfelgebiet, Poissonproblem, BPX

Nutzernetz: 768 Elemente

Zeit für PCG, 16/32/64 Knoten auf GCPP-128:

Verteilung	Anzahl der Prozessoren		
	16	32	64
zufällig	40.96	44.99	56.79
linear	7.62	6.35	7.38
Spektralbisektion (SB)	10.08	9.68	11.77
SB mit KL	7.97	7.98	9.19
SB mit Term. Prop.			6.03
Quadrisektion mit KL	7.67	–	9.10
... + Hypercubeadapt.	–	–	7.94
Oktasektion mit KL	–	–	9.28

Adaptivität

Initialisierungsphase auf 1 Prozessor:

- Nutzernetz einlesen
- Schleife:
 - Assemblieren/Lösen
 - Fehlerschätzen/Markieren
 - Netz verfeinern

bis Hauptnetz erreicht.

Verteilung auf Prozessoren.

Schleife:

- Assemblieren/Lösen
- Fehlerschätzen/Markieren/Verfeinern vorbereiten
- Dynamische Lastbalancierung
- Verschieben von Hauptnetzelementen (+ alle Daten)
- Netz verfeinern

bis gewünschte Genauigkeit erreicht

(oder Speicher voll).

Hauptnetz:

- definiert 1D- und 2D-Ketten,
- Motivation: $n < p$,
 - geringe Dimension des Lastbalancierungsproblems,
 - Verschieben einfach,
- ca $20 * p$ viele Elemente.

Zusätzliche Option: bei geringer Problemgröße nur wenige Prozessoren verwenden!