

# Allgemeine und spezielle Lösungskonzepte für großdimensionierte Probleme

Dipl.-Ing.Dr. Gundolf Haase



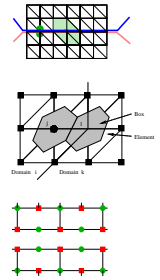
Salzburg, 4. Mai 2001

1

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Inhalt

- Multigrid
  - Grundprinzip
  - Algebraisches Multigrid (AMG)
- Die nichtüberlappende Gebietszerlegungsmethode
  - Datenverteilung, parallele CG Iteration
  - Erweiterung des Datenkonzeptes: akkumulierte Matrizen
  - Operationen mit akkumulierten Matrizen
  - Anwendungen in Algorithmen: ILU, Multigrid
- Theorie als Konstruktionsprinzip für parallele Algorithmen
  - Triple Matrix Produkt.
  - Setupschritt in parallelem algebraischem Multigrid (AMG)
- Anwendungen - weitere Arbeiten - Schlußfolgerungen



2

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Betrachtete Problemklassen

Finde  $u$  so daß 
$$\begin{aligned} Lu(x) &= f(x) & \forall x \in \Omega \\ lu(x) &= g(x) & \forall x \in \partial\Omega \end{aligned}$$

Variations  $\Downarrow$  Formulierung

Finde  $u \in \mathbb{V}$ : 
$$a(u, v) = \langle F, v \rangle \quad \forall v \in \mathbb{V}$$

FEM, FDM  $\Downarrow$  FVM, FIT

Löse 
$$K_h \cdot \underline{u}_h = \underline{f}_h \quad \underline{u}_h \in \mathbb{R}^{N_h}$$

- (lineares) Problem 2. Ordnung.
  - Poissongleichung (Temperatur)
  - Lamgleichungen (Verformungen)
  - Maxwellgleichungen (Magnetfeld)
- Eulergleichungen (dynamische Gase)
- kompressible Navier-Stokesgleichungen (Flüssigkeiten)
- Matrix  $K_h$  ist dünnbesetzt, positiv definit (symmetrisch, großdimensioniert)

3

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

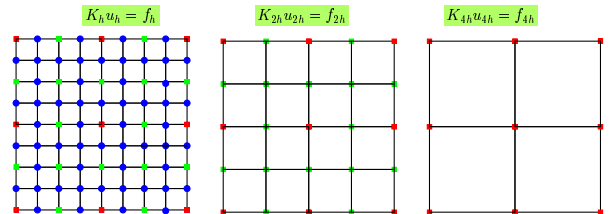
## Das Prinzip von Multigrid

Löse das Gleichungssystem

$$K_h u_h = f_h$$

mit Hilfe einer Folge verschiedener Diskretisierungen, d.h., durch

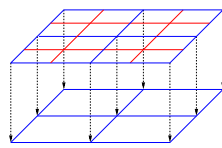
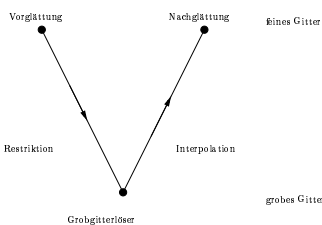
### Kombination verschiedener Genauigkeitsstufen.



4

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Zweigiterverfahren (Two grid method)



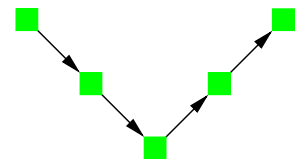
**Gewinn:** Optimaler iterativer Löser (Speicher, CPU-Zeit).  
**Bemerkung:** Eine rekursive Anwendung führt zum Multigridzyklus.

5

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## (Klassischer) Multigrid $V(\nu_F, \nu_B)$ -zyklus: $MG(u_l, f_l, l)$

if  $l = COARSELEVEL$  then  
 $u_l = (K_l)^{-1} f_l$  wird mit direktem Löser gelöst  
 else  
 Glätte  $\nu_F$ -mal mit  $K_l u_l = f_l$   
 Berechne den Defekt  $d_l = f_l - K_l u_l$   
 Restringiere den Defekt  $d_{l+1} = P_l^T d_l$   
 Sei  $w_{l+1} \equiv 0$   
 $MG(w_{l+1}, d_{l+1}, l+1)$   
 Interpoliere die Korrektur  $w_l = P_l w_{l+1}$   
 Aktualisiere die Lösung  $u_l = u_l + w_l$   
 Glätte  $\nu_B$ -mal mit  $K_l u_l = f_l$   
 end if



6

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Strategien in Multigrid

Jede MG-Iteration erfordert:  
 Matrizen  $K_l$ , Glätter, Interpolation  $P_l$ , (Restriktion  $R_l = P_l^T$ ) für jede Diskretisierungsstufe.

### • Klassisches MG:

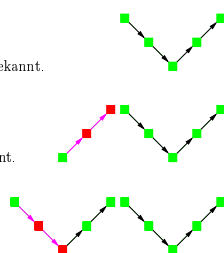
Netze/Gitter, Matrizen, Interpolation sind für alle Stufen bekannt.

### • Full MG - adaptives MG:

Grobnetz/-gitter, Grobmatrix und Interpolation sind bekannt.

### • Algebraisches MG (AMG):

Nur die Feingittermatrix (+Feingitter) ist bekannt.



7

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Hauptbestandteile von AMG

(i) Vergrößerung:

$$\omega^h = \omega_C^h \cup \omega_F^h$$

(ii) Interpolationsgewichte:

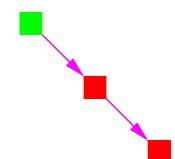
$$P = \{\alpha_{ij}\}_{i \in \omega^h, j \in \omega_C^h} : \mathbb{R}^H \mapsto \mathbb{R}^h$$

(iii) Grobmatrix:

$$K^H = P^T \cdot K \cdot P$$

(iv) (i)-(iii) rekursiv anwendbar.

(v) Anwendung der standard MG-procedure.



8

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Vergrößerung

9

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Beispiel: Magnetventil

$\int_{\Omega} v \text{curl} \text{curl} v + \sigma \mathbf{u} v \, dx$   
 $\sigma = 10^{-4}$   
 $\varepsilon = 10^{-8}$

$N_h$	it	setup	solver
8714	14	0.53	2.49
65219	28	4.17	44.01
504246	63	34.49	792.49

10

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Medizin: Kern des inversen Problems

$$-\nabla^T a(x) \nabla u(x) = f(x) \quad x \in \Omega$$

$$u(x) = g(x) \quad x \in \Gamma = \partial\Omega$$

$a(x)$  ist aus Computertomographie bekannt.  
 $g(x)$  wird gemessen.  
 $f(x)$  ist Ursache der Anomalie.

FEM Diskretisierung, etc.  $\Downarrow$  Inverser Löser

Löse mit CG und AMG als Vorkonditionierer

$$K \cdot \underline{u}_s = \underline{f}_s \quad s = 1, \dots, 10000$$

$\Downarrow$   
Quellen  $f(x)$

11

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Medizin: Simultane rechte Seiten

$K \cdot \underline{u}_s = \underline{f}_s \quad s=1, \dots, 10000 \Rightarrow 1^* \text{Setup(AMG)} + 10.000^* \text{PCG(AMG)} \approx 10.000 \times 70 \text{ sec.} \approx 8 \text{ days}$

Löse  $n \text{ sim}$  Gleichungssysteme simultan mit PCG(AMG).  
 AMG: V-zyklus mit einem Gauß-Seidel Schritt in Vor- und Nachglättung.

$$\text{blockdiag}\{K\} \cdot (\underline{u}_k)_{k=s}^{s+n \text{ sim}-1} = (\underline{f}_k)_{k=s}^{s+n \text{ sim}-1} \quad s=1, 1+n \text{ sim}, 1+2n \text{ sim}, \dots, 10000,$$

- Speicherhierarchien (Caches, Memory, NUMA): Speicherzugriffe sind kritisch.
- Anzahl der arithmetischen Operationen bleibt gleich.
- Ziel:  $t_{n \text{ sim}} < n \text{ sim} \cdot t_1$

12

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Medizin: Simultane rechte Seiten - Ergebnisse

Testbeispiel (90.000 d.o.f.)

$$\int_{\Omega} \nabla^T v \nabla u \, dx = \int_{\Omega} f v \, dx$$

$\Omega = [0, 1]^2$   
 $\varepsilon = 10^{-8}$   
 Händisches loop unrolling!!

13

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Verbinden von Multigriddkomponenten I

- Verbinden von **Glättung** und Defektberechnung: SMOOTHFDEFECT.

Gauss-Seidel vorwärts ( $\forall i = 1, \dots, n$ ):

$$\hat{u}_i = K_{ii}^{-1} \left( f_i - \sum_{j=1}^{i-1} K_{ij} \hat{u}_j - \sum_{j=i+1}^n K_{ij} u_j \right) = K_{ii}^{-1} v_i$$

$$d_i = f_i - \sum_{j=1}^n K_{ij} \hat{u}_j = f_i - \sum_{j=1}^{i-1} K_{ij} \hat{u}_j - \underbrace{K_{ii} \hat{u}_i}_{=v_i} - \sum_{j=i+1}^n K_{ij} \hat{u}_j$$

$$= h_i - v_i - \sum_{j=i+1}^n K_{ij} \hat{u}_j$$

14

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Verbinden von Multigriddkomponenten II

- $u = 0$  auf dem Grobgitter und Kombination von Glättung und Defektberechnung für *genau einen* Vorglättungsschritt: SMOOTHFODEFFECT

$$\hat{u}_i = K_{ii}^{-1} \left( f_i - \sum_{j=1}^{i-1} K_{ij} \hat{u}_j \right)$$

$$d_i = - \sum_{j=i+1}^n K_{ij} \hat{u}_j$$

- Verbinden von **Interpolation** und **Korrekturschritt**. (10% Gewinn)
- CRS-Format (Compressed row storage) für Systemmatrizen und Interpolationsmatrix (15% Gewinn).

15

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Medizin: Simultane rechte Seiten - Ergebnisse II

Testbeispiel (90.000 d.o.f.)

$$\int_{\Omega} \nabla^T v \nabla u \, dx = \int_{\Omega} f v \, dx$$

$\Omega = [0, 1]^2$   
 $\varepsilon = 10^{-8}$

5-20 Systeme simultan  
 $\Downarrow$   
 Speedup von 3.6 - 4  
 $\Downarrow$   
 Inverses Problem  $\approx 2 \text{ days}$

16

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

### Nichtüberlappende Gebietszerlegung

akkumuliert

$$\underline{u}_s = A_s \underline{u}$$

$$\mathcal{M}_s = A_s \mathcal{M} A_s^T$$

verteilt

$$\underline{r} = \sum_{s=1}^P A_s^T \underline{r}_s$$

$$K = \sum_{s=1}^P A_s^T K_s^{\text{FEM}} A_s$$

17

### Grundlegende Operationen

ohne Kommunikation

Globales Reduce

Next Neighbor Komm.

$$\underline{v} \leftarrow K \cdot \underline{f}$$

$$\underline{r} \leftarrow \underline{f} + \alpha \cdot \underline{v}$$

$$\underline{w} \leftarrow \underline{u} + \alpha \cdot \underline{f}$$

$$\underline{r} \leftarrow R^{-1} \cdot \underline{w}$$

$$\alpha \leftarrow \langle \underline{w}, \underline{r} \rangle = \sum_{s=1}^P \langle \underline{w}_s, \underline{r}_s \rangle$$

$$\underline{w} \leftarrow \underline{r} = \sum_{s=1}^P A_s^T \underline{r}_s$$

mit  $R = \text{diag}\{R_s\}_{s=1}^P = \text{diag}\{\# \text{ TG zu denen } x_i \text{ gehört}\} = \sum_{s=1}^P A_s^T \cdot A_s$

18

### Paralleler CG : $\text{PCG}(K, \underline{u}, \underline{f})$

repeat

$$\underline{v} \leftarrow K \cdot \underline{f}$$

$$\alpha \leftarrow \sigma / \langle \underline{f}, \underline{v} \rangle$$

$$\underline{u} \leftarrow \underline{u} + \alpha \underline{v}$$

$$\underline{r} \leftarrow \underline{r} - \alpha \underline{v}$$

$$\underline{w} \leftarrow C^{-1} \cdot \underline{r}$$

$$\sigma \leftarrow \langle \underline{w}, \underline{r} \rangle$$

$$\beta \leftarrow \sigma / \sigma_{\text{old}}, \quad \sigma_{\text{old}} \leftarrow \sigma$$

$$\underline{f} \leftarrow \underline{w} + \beta \underline{f}$$

until termination

19

### Gebietszerlegung: Blockstruktur der Matrix

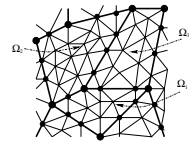
$$\begin{pmatrix} K_C & K_{CI} \\ K_{IC} & K_I \end{pmatrix} \cdot \begin{pmatrix} \underline{u}_C \\ \underline{u}_I \end{pmatrix} = \begin{pmatrix} \underline{f}_C \\ \underline{f}_I \end{pmatrix}$$

$$\begin{pmatrix} K_C & K_{CI,1} & K_{CI,2} & \dots & K_{CI,P} \\ K_{IC,1} & K_{I,1} & 0 & 0 & 0 \\ K_{IC,2} & 0 & K_{I,2} & 0 & 0 \\ \vdots & 0 & 0 & \ddots & 0 \\ K_{IC,P} & 0 & 0 & 0 & K_{I,P} \end{pmatrix} \cdot \begin{pmatrix} \underline{u}_{I,1} \\ \underline{u}_{I,2} \\ \vdots \\ \underline{u}_{I,P} \end{pmatrix} = \begin{pmatrix} \underline{f}_C \\ \underline{f}_{I,1} \\ \underline{f}_{I,2} \\ \vdots \\ \underline{f}_{I,P} \end{pmatrix}$$

↓

$$C_h \cdot \underline{u}_h = \underline{r}_h$$

ASM und MSM Vorkonditionierer



20

### Nichtüberlappende Gebietszerlegung

akkumuliert

$$\underline{u}_s = A_s \underline{u}$$

$$\mathcal{M}_s = A_s \mathcal{M} A_s^T$$

verteilt

$$\underline{r} = \sum_{s=1}^P A_s^T \underline{r}_s$$

$$K = \sum_{s=1}^P A_s^T K_s^{\text{FEM}} A_s$$

21

### Überlappende Gebietszerlegung

akkumuliert

$$\underline{u}_s = A_s \underline{u}, \quad \mathcal{M}_s = A_s \mathcal{M} A_s^T$$

verteilt

$$\underline{r} = \sum_{s=1}^P A_s^T \underline{r}_s, \quad K = \sum_{s=1}^P A_s K_s A_s^T$$

ABER, wie wählt man  $K_s$  ?

$$K_s := \sum_{\delta^{(r)} \in \Omega_s} \frac{1}{W^{(r)}} \cdot K^{\text{FEM},r}$$

$W^{(r)} := \# \Omega_s \text{ zu denen Element } \delta^{(r)} \text{ gehört.}$

22

### Eulergleichungen

Sei  $\vec{u} = (\varrho, \varrho u, \varrho v, e)^T$

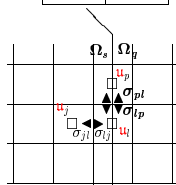
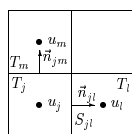
$$\partial_t \vec{u} + \partial_x f_1(\vec{u}) + \partial_y f_2(\vec{u}) = 0 \quad \text{in } \Omega \times (0, T]$$

+ A.B.    + R.B.

Explizites Schema, Approximation vom flux ( $j=1, \dots, N, \quad \forall K=0, 1, \dots$ ):

$$u_j^{K+1} = u_j^K - \frac{\tau}{|T_j|} \sum_{l=1}^{n_e(j)} \int_{S_{jl}} f(\underline{u}^K(s)) \vec{n}_{jl} ds$$

$$\approx u_j^K - \frac{\tau}{|T_j|} \sum_{l=1}^{n_e(j)} |S_{jl}| (\sigma_{jl} - \sigma_{lj})$$



[Haase]

23

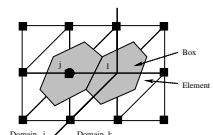
### Kompressible Navier-Stokes Gleichungen

Kompressible Navier-Stokes Gleichungen  $\rightarrow$  Eulergleichungen + 2. Ordnung PDE

$$\frac{1}{2} \partial_t \begin{pmatrix} \varrho \\ \varrho u \\ \varrho v \\ e \end{pmatrix} + \partial_x \begin{pmatrix} \varrho u \\ \varrho u^2 + p \\ \varrho u v \\ u \cdot (e + p) \end{pmatrix} + \partial_y \begin{pmatrix} \varrho v \\ \varrho u v \\ \varrho v^2 + p \\ v \cdot (e + p) \end{pmatrix} = 0$$

$$\frac{1}{2} \partial_t \begin{pmatrix} \varrho \\ \varrho u \\ \varrho v \\ e \end{pmatrix} + \partial_x \begin{pmatrix} 0 \\ \tau_{11} \\ \tau_{12} \\ \theta_1 \end{pmatrix} + \partial_y \begin{pmatrix} 0 \\ \tau_{21} \\ \tau_{22} \\ \theta_2 \end{pmatrix} = 0$$

Diskretisierung:  
Eulergleichungen  $\rightarrow$  Boxes, FVM  
2. Ordnung PDE  $\rightarrow$  FEM  
Duale Vernetzung.



[Haase]

24

## Grundlegende Operationen

ohne Kommunikation

Globales Reduce

Next Neighbor Komm.

$$\begin{aligned} \underline{v} &\leftarrow K \cdot \underline{u} \\ \underline{r} &\leftarrow \underline{f} + \alpha \cdot \underline{v} \\ \underline{w} &\leftarrow \underline{u} + \alpha \cdot \underline{r} \\ \underline{r} &\leftarrow R^{-1} \cdot \underline{w} \end{aligned}$$

$$\alpha \leftarrow \langle \underline{w}, \underline{r} \rangle = \sum_{s=1}^P \langle \underline{w}_s, \underline{r}_s \rangle$$

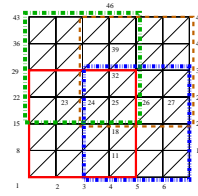
$$\underline{w} \leftarrow \underline{r} = \sum_{s=1}^P A_s^T \underline{r}_s$$

Was ist mit  $\mathfrak{M} \cdot \underline{v}$  möglich??

25

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Einige Definitionen



subdomain 1  
subdomain 2  
subdomain 3  
subdomain 4

Mengen der Teilgebiete:

$$\sigma^{[i]} = \{s : x^{[i]} \in \bar{\Omega}_s\}$$

Mengen von Indizes/Knoten:

$$\omega(\sigma) := \{i \in \omega : \sigma^{[i]} = \sigma\}$$

26

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Matrixmuster und -anwendungen

Die folgenden Operationen können parallel (ohne Komm.) ausgeführt werden;

$$\underline{f} = K \cdot \underline{u}$$

Matrix  $\mathfrak{M}$  erfüllt die Musterbedingung:

$$\forall i, j \in \omega : \sigma^{[i]} \not\subseteq \sigma^{[j]} \Rightarrow \mathfrak{M}^{[i,j]} = 0$$

$$\underline{u} = \mathfrak{M} \cdot \underline{w}$$

$$\underline{f} = \mathfrak{M}^T \cdot \underline{r}$$

$$K^H = \mathfrak{M}^T \cdot K \cdot \mathfrak{M}$$

Theoreme/Beweise [Haase]

27

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Zulässige Matrixoperationen

Vertex/Eckknoten, Edge/Kantenknoten, Innere Knoten

$$\mathfrak{M} = \begin{pmatrix} \mathfrak{M}_V & 0 & 0 \\ \mathfrak{M}_{EV} & \mathfrak{M}_E & 0 \\ \mathfrak{M}_{IV} & \mathfrak{M}_{IE} & \mathfrak{M}_I \end{pmatrix} = \mathfrak{M}_L + \mathfrak{M}_D \Rightarrow \underline{u} = \mathfrak{M} \cdot \underline{w}$$

Musterbedingung  $\sigma^{[i]} \not\subseteq \sigma^{[j]} \Rightarrow \mathfrak{M}^{[i,j]} = 0$  muß in allen Teilmatrizen erfüllt sein !! Erlaubt Operationen wie (Parallel ADI [DouHaar])

$$\underline{w} = \mathfrak{M} \cdot \underline{u} := (\mathfrak{M}_L + \mathfrak{M}_D) \cdot \underline{u} + \sum_{s=1}^P A_s^T \mathfrak{M}_{E,s} R_s^{-1} \cdot \underline{u}_s$$

oder, für  $\mathfrak{M} = \mathfrak{L}^{-1} \cdot \mathfrak{U}^{-1}$  ([Haase])

$$\underline{w} = \mathfrak{L}^{-1} \mathfrak{U}^{-1} \cdot \underline{r} := \mathfrak{L}^{-1} \sum_{s=1}^P A_s^T \mathfrak{U}_s^{-1} \cdot \underline{r}_s$$

28

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## UL Faktorisierung [Haase]

Start	$K$
Berechne	
$U_I, L_I$	$K_I = U_I \cdot L_I - P_I$
$L_{IE}$	$K_{IE} = U_I \cdot L_{IE} - P_{IE}$
$L_{IV}$	$K_{IV} = U_I \cdot L_{IV} - P_{IV}$
$U_{EI}$	$K_{EI} = U_{EI} \cdot L_I - P_{EI}$
$U_{VI}$	$K_{VI} = U_{VI} \cdot L_I - P_{VI}$
Modifiziere	
$K_E$	$K_E := K_E - U_{EI} \cdot L_{IE}$
$K_{EV}$	$K_{EV} := K_{EV} - U_{EI} \cdot L_{IV}$
$K_{VE}$	$K_{VE} := K_{VE} - U_{VI} \cdot L_{IE}$
$K_V$	$K_V := K_V - U_{VI} \cdot L_{IV}$
Akkumuliere	$\mathfrak{R}_E, \mathfrak{R}_{EV}, \mathfrak{R}_{VE}$
e.g.	$\mathfrak{R}_{EV} := \sum_{i=1}^P A_{E,i}^T K_{EV,i} A_{V,i}$
Bestimme	
$\mathfrak{U}_E, \mathfrak{L}_E$	$\mathfrak{R}_E = \mathfrak{U}_E \cdot \mathfrak{L}_E - P_E$
$\mathfrak{U}_{VE}$	$\mathfrak{R}_{VE} = \mathfrak{U}_{VE} \cdot \mathfrak{L}_E - P_{VE}$
$\mathfrak{L}_{EV}$	$\mathfrak{R}_{EV} = \mathfrak{U}_E \cdot \mathfrak{L}_{EV} - P_{EV}$
Modifiziere	$K_V := K_V - \mathfrak{U}_{VE} \cdot \mathfrak{R}_E^{-1} \cdot \mathfrak{L}_{EV}$
Akkumuliere	$\mathfrak{R}_V$

31

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Parallele Iterationsverfahren zum Lösen von $K \cdot \underline{u} = \underline{f}$

• Richardson-Iteration:

$$\underline{u}_s^{k+1} := \underline{u}_s^k + \tau \sum_{q=1}^P (\underline{f} - K \cdot \underline{u}^k)_q$$

• Jacobi-Iteration mit  $\mathfrak{D} = \sum_{s=1}^P \text{diag}\{K_s\}$ :

$$\underline{u}_s^{k+1} := \underline{u}_s^k + \omega \mathfrak{D}_s^{-1} \sum_{q=1}^P (\underline{f} - K \cdot \underline{u}^k)_q$$

• Unvollständige Faktorisierung  $\mathfrak{R} = \mathfrak{U} \cdot \mathfrak{D} + \mathfrak{R}$ :

$$\underline{u}_q^{k+1} := \underline{u}_q^k + \mathfrak{L}_q^{-1} \sum_{q=1}^P \mathfrak{U}_q^{-1} (\underline{f} - K \cdot \underline{u}^k)_q$$

30

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Paralleles Multigrid : PMG( $K, \underline{u}, \underline{f}, \ell$ )

if  $\ell == 1$  then

$$\text{Löse } \sum_{s=1}^P A_s^T K A_s \cdot \underline{u} = \underline{f}$$

else

$$\underline{\hat{u}} \leftarrow \text{SMOOTH}(K, \underline{u}, \underline{f}, \nu)$$

$$\underline{d} \leftarrow \underline{f} - K \cdot \underline{\hat{u}}$$

$$\underline{d}^H \leftarrow \mathfrak{P}^T \cdot \underline{d}$$

$$\underline{m}^H \leftarrow 0$$

$$\text{PMG}^{\gamma}(K^H, \underline{m}^H, \underline{d}^H, \ell - 1)$$

$$\underline{m} \leftarrow \mathfrak{P} \cdot \underline{m}^H$$

$$\underline{\hat{u}} \leftarrow \underline{\hat{u}} + \underline{m}$$

$$\underline{u} \leftarrow \text{SMOOTH}(K, \underline{\hat{u}}, \underline{f}, \nu)$$

end if

## Parallelisierungs-idee für AMG

Paralleles MG  $\Leftrightarrow$  Interpolation  $\mathfrak{P}$  muß die Musterbedingung erfüllen

$$K^H = \mathfrak{P}^T \cdot K \cdot \mathfrak{P} \text{ kann in PMG}(K^H, \underline{u}^H, \underline{f}^H, \ell - 1) \text{ benutzt werden.}$$

Idea

Steuerung von Vergrößerung und Interpolation derart, daß die Musterbedingung für  $\mathfrak{P}$  erfüllt ist.

[Haase]

32

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Vergrößerung im sequentiellen AMG

$$(\omega_C, \omega_F) \leftarrow \text{COARSE}(\{S_h^{i,T}\}, \omega_C, \omega_F)$$

```

while  $\omega_C \cup \omega_F \not\supseteq \omega_h$  do
   $k \leftarrow \text{PICK}(\{S_h^{i,T}\}, \omega_h \setminus (\omega_C \cup \omega_F))$ 
  if  $|S_h^{k,T}| + |S_h^{k,T} \cap \omega_F| = 0$  then
     $\omega_F \leftarrow \omega_F \cup (\omega_h \setminus \omega_C)$ 
  else
     $\omega_C \leftarrow \omega_C \cup \{k\}$ 
     $\omega_F \leftarrow \omega_F \cup (S_h^{k,T} \setminus \omega_C)$ 
  end if
end while

```

33

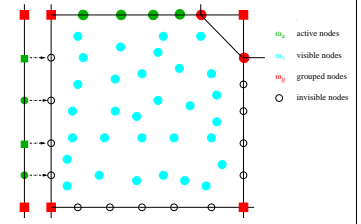
Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Steuerung des Musters bei der Vergrößerung

$$\text{COARSEP}(\{S^i\}, \omega_m, \omega_a, \omega_v, \omega_C, \omega_F)$$

•  $K_i \xrightarrow{\text{pat tern}} \tilde{K}_i$

- active nodes:  $\omega_a := \{i \in \omega_s : \sigma^{|i|} \equiv \sigma_k\}$ ,
- visible nodes:  $\omega_v := \{i \in \omega_s : \sigma^{|i|} \subset \sigma_k\}$ ,
- grouped nodes:  $\omega_g := \{i \in \omega_s : \sigma^{|i|} \supset \sigma_k\}$ ,
- invisible nodes:  $\omega_s \setminus (\omega_a \cap \omega_v \cap \omega_g)$ .



34

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Vergrößerung auf Teilmenge von Knoten im parallelen AMG

$$(\omega_C, \omega_F) \leftarrow \text{COARSEP}(\{S_h^{i,T}\}, \omega_m, \omega_a, \omega_v, \omega_C, \omega_F)$$

```

while  $\omega_C \cup \omega_F \not\supseteq \omega_h$  do
   $k \leftarrow \text{PICK}(\{S_h^{i,T}\}, \omega_m \setminus (\omega_C \cup \omega_F), \omega_F)$ 
  if  $|S_h^{k,T}| + |S_h^{k,T} \cap \omega_F| = 0$  then
     $\omega_F \leftarrow \omega_F \cup (\omega_h \setminus \omega_C)$ 
  else
     $\omega_C \leftarrow \omega_C \cup \{k\}$ 
     $\omega_F \leftarrow \omega_F \cup ((S_h^{k,T} \setminus \omega_C) \cap \omega_v)$ 
  end if
end while

```

35

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

$$\text{PARCOARSE}(\{S^i,T\}, \omega_s, \omega_C, \omega_F)$$

Bestimme die Liste  $[\sigma_k]_{k=1, \dots, m_c}$  (= Kommunikationsgruppen).

```

for all  $k = 1, \dots, m_c$  do
   $\omega_a \leftarrow \{i \in \omega_s : \sigma^{|i|} \equiv \sigma_k\}$ 
   $\omega_v \leftarrow \{i \in \omega_s : \sigma^{|i|} \subset \sigma_k\}$ 
   $\omega_g \leftarrow \{i \in \omega_s : \sigma^{|i|} \supset \sigma_k\}$ 
  if  $s = \text{ROOT}(\sigma_k)$  then
     $\text{COARSEP}(\{S^i,T\}, \omega_a, \omega_a, \omega_v, \omega_C, \omega_F)$ 
     $\omega_m \leftarrow \omega_C \cap \omega_a$ 
  end if
   $\omega_m \leftarrow \text{BROADCAST}(\sigma_k, \omega_m)$ 
  if  $s = \text{ROOT}(\sigma_k)$  then
     $\text{COARSEP}(\{S^i,T\}, \omega_m, \omega_a, \omega_v, \omega_C, \omega_F)$ 
  end if
  for all  $i \in \omega_a$  do
    if  $i \in \omega_F$  then
       $S^{i,T} \leftarrow S^{i,T} \cap (\omega_a \cup \dots)$ 
    else
       $S^{i,T} \leftarrow S^{i,T} \cap (\omega_a \cup \omega_v)$ 
    end if
  end for
end for

```

## Paralleles AMG: $\{K^j, \mathfrak{P}^j\}_{j=1}^{\text{COARSELEVEL}} \leftarrow \text{PARSETUP}(K_h, \omega_h, \ell)$

```

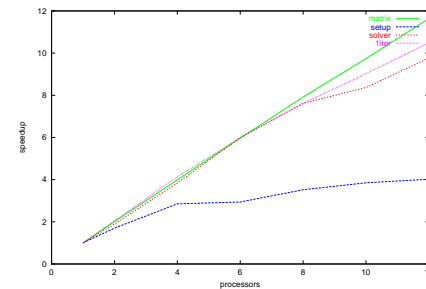
if  $\|\omega\| > \text{COARSEGRID}$  then
   $K^\ell \leftarrow K_h$ 
   $\tilde{\mathfrak{R}} \leftarrow (\sum_{s=1}^p (A_s^T)^T \tilde{K}_s A_s^T)$  pattern
   $\{S^{i,T}\}^\ell \leftarrow \text{GETSTRONG}(\tilde{\mathfrak{R}})$ 
   $(\{S^{i,T}\}^\ell, \omega_C, \omega_F) \leftarrow \text{PARCOARSE}(\{S^{i,T}\}^\ell, \omega^\ell)$ 
   $\mathfrak{P}^\ell \leftarrow \text{WEIGHTS}(\{S^{i,T}\}^\ell, \tilde{\mathfrak{R}}, \omega_C, \omega_F)$ 
   $K_H \leftarrow (\mathfrak{P}^\ell)^T K^\ell \mathfrak{P}^\ell$ 
   $\omega_H \leftarrow \omega_C$ 
   $\text{PARSETUP}(K_H, \omega_H, \ell + 1)$ 
else
   $\text{COARSELEVEL} \leftarrow \ell$ 
end if

```

37

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Medizin: Parallelrechner (AMG<sub>1</sub>)



$\int_{\Omega} \nabla^T v \nabla u dx = \int_{\Omega} f v dx$   
 $\Omega = \text{menschlicher Kopf}$   
 $\varepsilon = 10^{-8}$   
 SGI ORIGIN 200  
 Speedup 10 bei 12 Proz.  
 $\downarrow$   
 8 days / 4 / 10  $\approx$  5 h !!

38

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Aktuelle Projekte

- **Medizin: Quellenidentifikation** [MPI Leipzig]  
→ AMG, Cache, Parallel
- **Form- und Dickenoptimierung** [ENGEL Schwertberg, Land, Sparkasse]  
→ Large-scale Optimierung, adaptive Löser
- **Ozeanmodellierung** [NSF, University of Miami, Kentucky State University]  
→ ASM/MSM Löser (instationär), Parallel, Cache
- **Mathematik: Schnelle Löser für Randelementgleichungen** [Uni Stuttgart]  
→ AMG, parallel
- **Informatik: Parallelisierungsstrategien** [Land, BMWV ??]  
→ parallel, FEM-Ingenieurssoftware
- **Physik: Schnelle Lösung der Kohn-Sham Gleichungen in 3D** [Uni Linz, FWF-Antrag]  
→ nichtlineare EVP, Multigrid, Parallel
- **SFB F013 Numerical and Symbolic Scientific Computing** [FWF, Land, Stadt, Uni]

39

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Geplante Arbeiten

- **Robust AMG and its Parallelization** [FWF-Antrag]  
→ weitere Problemklassen, Theorie AMG, Parallel, Effizienz
- Buch über **Gebietszerlegungsmethoden** [+ SIAM], **Summer School** in Montreal  
→ Theorie ASM/MSM, Effizienz
- **Parallelisierung Ingenieursoftware** [Steyr-Daimler-Puch, FFF, Minister, AVL]  
→ Parallel, schnelle Löser, UNIX/Win-Cluster
- **Form- und Dickenoptimierung mit geom. Oberflächenbeschreibung**  
[FWF-Antrag (Geometrie: Prof. Jüttler), ENGEL Schwertberg]  
→ Parametrisierungen, Optimierung
- **Parallele Optimierung** [Uni Graz (Prof. Kunisch)]  
→ Parallel, Optimierungsalgorithmen
- **SFB F013, Industrieprojekte** [SDP, AVL, IBM, ...]

40

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Weitere Kontakte

- Uni Bonn, Uni Chemnitz, Uni Dresden, Uni Heidelberg, Uni Erlangen
- TH Győr, ELTE Universität Budapest
- CERFACS Toulouse
- Old Dominion University (Norfolk), Washington State University, McGill University Montreal
- Lawrence Livermore National Labs (LLNL), SANDIA Labs Livermore
- MPI Leipzig, GMD St. Augustin
- BMW Steyr, Bosch Linz, EBI Linz

41

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Schlußbemerkungen

- Parallelisierung, Mehrgitter, Multilevel, ASM, MSM, ...
  - sind für viele Aufgabenklassen **verfügbar** und **erweiterbar**.
  - sind in Software **umgesetzt** (TRANSCG, PEBBLES, FEPP, DDDCOM).
  - werden für reale, große Probleme **angewendet** und **angepaßt**.
  - führen zu **neuen sequentiellen** Algorithmen (Patch-AMG, Patch-Glätter)
- **Wissenschaftliches Rechnen** besteht aus
  - Mathematischer Theorie (Was kann ich machen?)
  - Informatik (Wie kann ich es machen?)
  - Anwendungswissenschaften (Warum mache ich es?)

**Spaß an der Arbeit!**

42

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Ozeanmodellierung - NSF Projekt mit USA-Partnern

Problem ist **zeitabhängig**, nichtlinear, nichtsymmetrisch und großdimensioniert [38+428 tUS-\$].  
Modellierung von M. Iskandarani [Miami] : 3D Ozean  
⇒ Shallow Water Equations + 5 Schichten für Smagorski-Filterung (Turbulenzen)

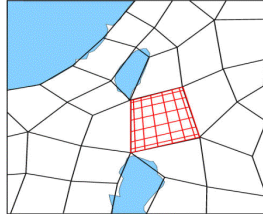
### Zeitdiskretisierung (AB3):

Nichtlineare, nichtsymm. Anteile → explizit  
Lineare Anteile → implizit

### Ortsdiskretisierung:

Spektralelemente mit  
8 × 8 Geschwindigkeits-Knoten  
6 × 6 Druck-Knoten

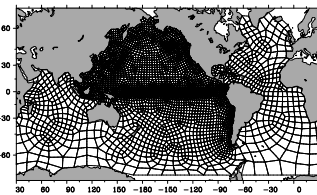
↓  
Ausnutzung der Elementstruktur in Lösem.



43

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Nordostpazifik



3552 Elemente  
176.377 Geschwindigkeitsknoten  
90.459 Druckknoten  
Simulationszeitraum: **1982-1998**  
Zeitschrittweite: **150 sec.**  
Rechenzeit: **50.000 h**

↓  
Bessere numerische Verfahren  
Algorithmenbeschleunigung  
Parallelisierung

44

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Ozeanmodellierung: Gekoppeltes System

$$\begin{aligned} \frac{1}{\tau} M^v u + \frac{1}{2} g G^x \zeta &= a & M^v \text{ ist Diagonalmatrix !!} \\ \frac{1}{\tau} M^v v + \frac{1}{2} g G^y \zeta &= b & \rightarrow \text{einfacher Uzawa Alg.} \\ -\frac{1}{2} D^x u - \frac{1}{2} D^y v + \frac{1}{\tau} M^p \zeta &= c & \rightarrow \text{Schurkomplement} \end{aligned}$$

Löse (CG)  $S \zeta := \left[ \frac{M^p}{\tau} + \frac{1}{4} \left( (G^x)^T H (M^v)^{-1} G^x + (G^y)^T H (M^v)^{-1} G^y \right) \right] \zeta = f$

**Vorkonditionierung von S:** Alt: Diagonalmatrix mit gelumpem S.

Neu: Patchglätter (überlappende ASM)

⇒ Reduktion der Iterationszahlen um Faktor 4.

⇒ Reduktion der Rechenzeit um Faktor 2.8.

Neu\*: Twolevel-Methode mit Patchglätter

⇒ **Robustheit** bzgl.  $\tau$  (Zeitdiskretisierung)

45

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics

## Ozeanmodellierung: Filterung

Filter: Reihenentwicklung und Boyden-Vandeven Filter  
 $\delta$  - gefiltertes Divergenzfeld,  $\omega$  - gefiltertes Geschwindigkeitsfeld  
In jeder der 5 Ozeanschichten muß gelöst werden:

$$-\Delta u(x) = f(x) \text{ in } \Omega$$

mit  $f := \nabla \delta - \nabla \times \omega$   
und Randbedingungen

- Semi-uniforme Vernetzung ⇒ Schur-Komplement  $S_C$
- Diagonal-VK für  $S_C$  im CG ⇒ Beschleunigung
- BPS-VK für  $S_C$  im CG ⇒ **zu teuer** für gewünschte Genauigkeit
- GIS (+ RB) für alle 5 Schichten und für  $u_x, u_y$  gleich!  
⇒ **Simultanes Lösen** mit 10 rechten Seiten  
⇒ Beschleunigung um Faktor 1.5

46

Johannes Kepler University Linz  
Institute of Analysis and Computational Mathematics