Technische Universität Chemnitz-Zwickau

DFG-Forschergruppe "SPC" · Fakultät für Mathematik

Matthias Pester

Grafik-Ausgabe vom Parallelrechner für 2D-Gebiete

Fakultät für Mathematik
TU Chemnitz-Zwickau
D-09107 Chemnitz, FRG
(0371)-531-2656
(0371)-531-2657 (fax)
m.pester@mathematik.tu-chemnitz.de
01.12.94

Preprint-Reihe der Chemnitzer DFG-Forschergruppe
"Scientific Parallel Computing"

SPC 94_24 **November 1994**

Grafik-Ausgabe vom Parallelrechner für 2D-Gebiete

Matthias Pester

1 Einleitung

Mit dieser Dokumentation werden die an der TU Chemnitz entwickelten Grafikroutinen vorgestellt, mit deren Hilfe eine schnelle Visualisierung (*on-line*) von 2D-Netzen und Ergebnissen vom Parallelrechner aus ermöglicht wird. Grundlage für die Realisierung sind die in [3] dokumentierten Routinen zur Nutzung der X11-Bibliothek ([1]) und zur Kommunikation im Hypercube ([9]), sowie eine Reihe von Routinen zur numerischen Lösungsauswertung, ursprünglich für PC-Grafik von A. Meyer, C. Israel und S. Meinel entwickelt.

Die bisher verwendeten Routinen `gebgraf` (für Potential- und Deformationsprobleme) bzw. `firegraf` (für Strömungssimulationen) sind unter einer einheitlichen Benutzeroberfläche vereinigt worden, deren Handhabung im folgenden beschrieben wird.

2 Programmschnittstellen

Die entsprechende Grafikroutine (`gebgraf`, `gebmgraf` oder `firegraf`) kann im wesentlichen als Black-Box betrachtet werden; die Auswahl von Darstellungsformen und -parametern erfolgt nach dem Aufruf des Unterprogramms interaktiv unter Verwendung von Maus und Tastatur.

Voraussetzung ist, daß zuvor ein Grafikfenster initialisiert wurde. Hierzu wird das Unterprogramm `ginitx` empfohlen, das nur auf Prozessor 0 aufzurufen ist:

```
call ginitx ( mode, ier )
```

Parameter:

`mode` : Initialisierungsparameter für die Farbpalette
0 = `autodetect`, passend zum verwendeten Bildschirm;
1 = `16 colors`, entspricht etwa der EGA-Palette eines PC
2 = `70 colors`, Regenbogenpalette blau-grün-gelb-rot
3 = `99 colors`, Regenbogenpalette violett-blau-grün-gelb-rot-weiß
4 = `B & W`, schwarz-weiß
5 = `Grayscale`, 14 Graustufen

`mode` kann um den Wert `10` höher angegeben werden, um für dieses Fenster eine eigene Farbpalette anlegen zu lassen und damit von den Farben in anderen Fenstern unabhängig zu sein. (s. u.)

`ier` : Fehleranzeiger: $\neq 0$ bedeutet, daß kein Grafikfenster geöffnet werden konnte.

Bemerkungen zu `ginitx`: Um ein Grafikfenster auf dem Bildschirm des Nutzers zu öffnen, sind die üblichen Voraussetzungen zu schaffen:

`xhost +`*parhost*	auf dem lokalen Rechner *myhost*,
`setenv DISPLAY` *myhost*`:0`	auf dem Host des Parallelrechners *parhost*.

Das Setzen der `DISPLAY`-Variablen kann auch durch das Anwenderprogramm selbst realisiert werden, indem das Unterprogramm

```
call gsetdisplay ('myhost:0')
```

<u>vor</u> `ginitx` aufgerufen wird, bzw. als Reaktion auf `ier`$\neq 0$ mit nochmaligem Aufruf von `ginitx`.

Wird beim Starten des parallelen Programms eine vollständige (!) `geometry`-Spezifikation angegeben, z. B. (unter PARIX):

```
run ... myprogram.px -geometry 500x500+20-20
```

so wird diese von `ginitx` für das Grafikfenster verwendet, anderenfalls gilt als Standard: `400x320-100+100`. Die Fenstergröße gilt im weiteren Programmablauf als Mindestgröße, so daß manuell nur noch eine Vergrößerung möglich ist.

Bei der Verteilung der Farben durch X11 können unter Umständen Nebeneffekte auftreten, die folgendermaßen zu erklären sind. Zunächst existiert eine Standardfarbpalette, die von allen Fenstern gemeinsam genutzt wird. Dort können (je nach Farbtüchtigkeit des X-Servers) i. a. bis zu 256 verschiedene Farben eingetragen werden, denen jeweils ein Farbindex (0…255) zugeordnet wird. Fordert ein Programm einen neuen Index für eine (über `RGB`-Anteile) gewünschte Farbe an, so wird bei Bedarf ein neuer Index vergeben oder ein vorhandener Index nochmals verwendet, wenn die gewünschte Farbe *„nahe“* an einer schon (von anderen Programmen) definierten Farbe liegt. Ist beides nicht möglich, liefert X11 keinen gültigen Farbindex, was sich in unserem Fall dadurch äußert, daß die gewünschte Farbpalette unvollständig ist. Ein Hinweis dazu erscheint durch das von `ginitx` benutzte Unterprogramm `setxcolors` jeweils in der Standardausgabe.

Als Ausweg bietet X11 die Möglichkeit, einem Fenster eine individuelle Farbpalette zuzuordnen, die dann nicht mehr durch fremde Anwendungsprogramme beeinflußt wird. Der X-Server schaltet dann stets zwischen den jeweiligen Farbpaletten um, wenn das betreffende Fenster aktiviert bzw. deaktiviert wird. Im jeweils anderen Bereich des Bildschirms treten somit Falschfarben auf. Beim Aufruf von `ginitx` kann diese individuelle Palette durch Angabe von (`mode+10`) anstelle von `mode` angefordert werden.

Da die von X11 vergebenen Farbindizes in keiner Beziehung zur Reihenfolge ihrer

Zuordnung für ein bestimmtes Programm stehen, verwenden wir eine nochmalige Indizierung für die Zuordnung zwischen Programm- und X11-Farbnummern. Darin bedeuten:

`0,1`	:	Text-/Hintergrund (weiß/schwarz, vertauschbar)
`2,3,4`	:	Grundfarben rot, grün, blau
`5,...,ncolmax`	:	restliche Farben der Palette

Die Gesamtzahl `ncolmax` verwendbarer Farbnummern wird als `common`-Variable gespeichert (in `Grafics.inc`). Sie muß nicht exakt mit den obigen Angaben (70 bzw. 99) übereinstimmen. In der Graustufen-Palette werden die Farben 2...15 mit zunehmender Helligkeit verwendet.

Nachdem ein Grafikfenster geöffnet wurde, können die Grafikroutinen `gebgraf` bzw. `firegraf` jederzeit gerufen werden, um aktuelle Daten zu visualisieren. Diese Unterprogramme sind so konzipiert, daß sie zugleich auf allen Prozessoren (nicht nur auf Prozessor 0) gerufen werden müssen. Sie enthalten die notwendigen Kommunikationen, um die Grafikdaten über Prozessor 0 auf den Bildschirm zu bringen.

```
call gebgraf  (IDoF,Nel,Ne,EL,Nkn,Nodes,U,H)
call gebmgraf (IDoF,Nel,Ne,EL,Nkn,Nodes,U,Mat,H)
call firegraf (IDoF,Nel,Ne,EL,Nkn,Nodes,U,Xi,H)
```

Parameter:

`IDoF`	:	Auswahl einer Darstellungsart: 0 = nur Netzdarstellung (keine Lösungsdaten) sonst: Nummer des darzustellenden Freiheitsgrades (bei `firegraf`: Stromlinien; bei `gebgraf`: filled areas)
`Nel`	:	Anzahl der FEM-Elemente in `EL`
`Ne`	:	Anzahl der Knoten pro Element
`EL`	:	Elementliste `EL(Ne,Nel)`, enthält für jedes Element die Knotennummern; zu beachten: Wenn vorhanden, werden die Seitenmittelknoten innerhalb eines Elements hinter den Eckknoten angeordnet.
`Nkn`	:	Anzahl der Knoten in `Nodes`
`Nodes`	:	Knotenliste, `Nodes(3,Nkn)` (x, y)-Koordinaten und Randbedingungscode;
`U`	:	Lösungsvektor, aufeinanderfolgend für die einzelnen Freiheitsgrade je ein `REAL*8`-Wert pro Knoten, z. B. für vektorielle Lösungen zunächst alle x-Werte, dahinter alle y-Werte.
`Mat`	:	Vektor der Länge `Nel` mit den Materialbereichsnummern der einzelnen FEM-Elemente (nur bei `gebmgraf`)
`Xi`	:	Vektor mit den Werten der Stromfunktion (nur bei `firegraf`).
`H`	:	Hilfsfeld *hinreichender* Länge, dient dem Aufbau von Pixeldaten für die Visualisierung und der Kommunikation zwischen den Prozessoren.

Es sei nochmals darauf verwiesen, daß das Feld `U` bei mehreren Freiheitsgraden wie ein zweidimensionales Feld `U(Nkn,*)` interpretiert wird, d. h. jeder Freiheitsgrad entspricht einem zusammenhängenden Teilvektor aus `U`. Die Anzahl und Bezeichnung der Freiheitsgrade erfährt das Grafikprogramm durch den Aufruf eines **vom Nutzer zu schreibenden** Unterprogramms:

```
call getdofs (nDoFs,DoFs)
```

Parameter:

`nDoFs`	:=	Anzahl der darstellbaren Freiheitsgrade
`DoFs`	:=	`character*6`-Feld; Kurznamen für die Freiheitsgrade

Im Falle von `firegraf` wurde zur Darstellung spezieller Funktionen folgende Konvention zur Anordnung der Freiheitsgrade festgelegt:

`1...nDoFs-3`	:	beliebige Freiheitsgrade aus dem Feld `U`
`nDoFs-2`	:	Druckverteilung (letzter Teilvektor auf `U`) (wird nur an Eckknoten ausgewertet)
`nDoFs-1`	:	Stromfunktion aus dem Feld `Xi`
`nDoFs`	:	Geschwindigkeit, berechnet aus den ersten beiden Freiheitsgraden, die als x- und y-Anteil des Geschwindigkeitsfeldes betrachtet werden; Darstellung des Betrages oder Vektordarstellung sind möglich.

Wenn im Falle von `geb[m]graf` eine vektorielle Lösung vorliegt, so sind ebenfalls die ersten beiden Freiheitsgrade als x- und y-Komponenten zu verwenden und der letzte Freiheitsgrad (Nr. `nDoFs`) zur Visualisierung der Vektorlösung (Darstellung von Pfeilen bzw. *Länge* der Vektoren).

Für alle Programmvarianten wird zusätzlich zu den über `getdofs` definierten Freiheitsgraden folgende Darstellung angeboten:

`nDoFs+1`	:	Visualisierung der Gebietsaufteilung auf die Prozessoren (Teilgebiete der Prozessoren werden mit unterschiedlichen Farben angezeigt)
`nDoFs+2`	:	(bei `gebmgraf`) Darstellung der verschiedenen Materialbereiche durch Farben.

Für beide sind die Darstellungsmodi „`Net-2D`“ oder „`Filled`“ sinnvoll (s. u.).

3 Interaktive Benutzerschnittstelle

Nach dem Aufruf von `...graf` erfolgt die weitere Auswahl von Darstellungsformaten und anderen Optionen interaktiv, teils im Grafikfenster unter Verwendung von Maus oder „Hotkeys“, teils im Textfenster zur Eingabe zusätzlicher Parameterwerte. Das Grafikfenster enthält dazu eine einfache Menüzeile mit Pull-Down-Menüs, die mit Hilfe der Maus aktiviert werden können. Eine Übersicht der bisher vorgesehenen Menüfunktionen ist in Abbildung 1 dargestellt. Die rechte obere Ecke enthält ein Feld zum Ein- und Ausblenden der Farbskala am rechten Rand des Fensters.

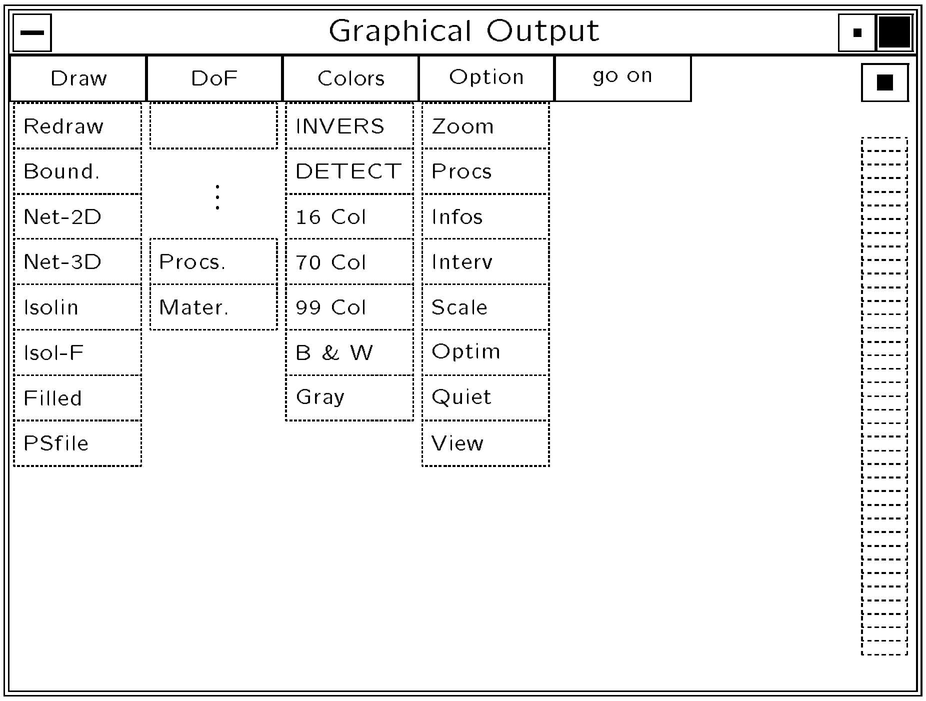

Abbildung 1: Übersicht der Pull-Down-Menüs im Grafik-Fenster

Je nach Aufrufparametern und aktuellem Zustand des Programms können einzelne Teile der Menüs unterdrückt sein.

Hinsichtlich der Verwendung der Maus muß darauf hingewiesen werden, daß das Programm auf dem Parallelrechner den Status der Maustasten im Polling-Verfahren abfragt, so daß unter Umständen ein länger andauernder Mausclick erforderlich ist, wo sonst ein kurzer Click genügt. Die bisherige Implementierung der Maussteuerung unterscheidet nicht, ob das Grafikfenster auf dem Bildschirm im Vordergrund (also aktiv) ist oder nicht. Das Programm reagiert daher auch auf Mausclicks, die für ein darüberliegendes Fenster bestimmt sind.

- Das erste Untermenü `Draw` ermöglicht die Auswahl einer Darstellungsart für einen zuvor ausgewählten Freiheitsgrad.

  `Redraw` Bild löschen und letzte Darstellungsart wiederholen;

  `Bound.` bei 2D-Darstellungen werden nur die Kanten mit vorgegebenen Randbedingungen eingezeichnet (Dirichlet: grün, Neumann: rot); Koppelkanten zwischen Prozessoren werden dabei ebenfalls i. a. rot dargestellt, wenn keine anderen Randbedingungen vorgegeben sind.
  Wenn zuvor `Net-3D` angezeigt wurde, bewirkt `Bound.` die Anzeige des umschreibenden Quaders zur besseren räumlichen Orientierung.

  `Net-2D` Darstellung der Vernetzung;
  bei vektoriellen Lösungen (Freiheitsgradnummer `nDoFs`, wenn `nDoFs`>2) erscheint an dieser Stelle `Vector` zur Anzeige von Geschwindigkeiten bzw. Verschiebungen oder Gradienten durch Vektoren (Pfeile).

`Net-3D` 3D-Gebirgsdarstellung einer skalaren Lösung über dem Gebiet. Der Blickwinkel kann über das links oben erscheinende kleine Koordinatensystem variiert werden. Dazu wird dieses mit der Maus angeclickt oder der Menüpunkt `Option-View` gewählt. Obwohl auch eine Maussteuerung vorgesehen ist, geschieht die Wahl der richtigen Darstellungsform für `Net-3D` zuverlässiger über die Tastatur (siehe Tabelle 1).

`Isolin` Lösungsdarstellung mittels Isolinien (25 Stufen), die Linien überlagern die vorherige Ausgabe;

`Isol-F` Isolinien auf einfarbigem Hintergrund;

`Filled` Lösungsdarstellung als Isoflächen (entsprechend der gewählten Farbpalette; bei schwarz-weiß: 30 Stufen mit alternierender Farbe);

`PSfile` Eröffnung eines Files für Postscript-Ausgabe: nachfolgende Darstellungen, einschließlich Farbskala am rechten Rand, werden zugleich auf Bildschirm und PS-File ausgegeben. Anschließend ist das PS-File wieder zu schließen (an gleicher Stelle im Menü), da es nicht vorgesehen ist, mehrere Abbildungen nebeneinander oder mehrere Seiten in eine Datei auszugeben. Das Postscript-File entspricht dem EPS-Format (Encapsulated Postscript), das sehr gut für die Weiterbearbeitung geeignet ist (z. B. mit LaTeX, Skalierung auf beliebige Größe).

- Das zweite Untermenü `DoF` enthält die mit dem Unterprogramm `getdofs` (s. o.) definierten Namen der Freiheitsgrade. Für Aufgaben mit nur einem Freiheitsgrad entfällt natürlich die Auswahlmöglichkeit, ebenso wenn das Programm `...graf` mit `IDoF=0` aufgerufen wurde.
  Zusätzlich zu den vom Nutzer zu bestimmenden Freiheitsgraden wurden in dieses Menü die Punkte `Procs.` und `Mater.` (letzteres bei `gebmgraf`) aufgenommen, um die Verteilung der Teilgebiete über die Prozessoren bzw. die einzelnen Materialbereiche zu visualisieren.

- Mit dem dritten Untermenü (`Colors`) kann nachträglich eine andere Farbpalette gewählt werden als die bei `ginitx` voreingestellte:

`INVERS` Austausch von Text- und Hintergrundfarbe (weiß $\leftrightarrow$ schwarz);

`DETECT` Auswahl einer für den aktuellen Monitor geeignet erscheinenden Palette, dies kann sein: schwarz-weiß, Graustufen oder 16 Farben;

`16 Col` Palette mit 16 Farben, einschließlich schwarz, weiß, rot, grün, blau;

`70 Col` Palette mit insgesamt 76 Farben, wobei die Regenbogenpalette von blau bis rot 71 Abstufungen umfaßt;

`99 Col` Palette mit insgesamt 105 Farben, wobei die Regenbogenpalette von violett bis weiß 100 Abstufungen entspricht.

`B & W` Schwarz-Weiß-Darstellung;

`Gray` Palette mit 14 Graustufen von schwarz bis weiß.

Tabelle 1: Einstellung des Koordinatensystems im Grafikfenster

X , ⇧ X	– Drehung um die x–Achse des Weltkoordinatensystems (vor-/rückwärts)
Y , ⇧ Y	– Drehung um die y–Achse des Weltkoordinatensystems (vor-/rückwärts)
Z , ⇧ Z	– Drehung um die z–Achse des Weltkoordinatensystems (vor-/rückwärts)
W , ⇧ W	– Drehung um die horizontale Bildachse (vor-/rückwärts)
U , ⇧ U	– Drehung um die vertikale Bildachse (vor-/rückwärts)
B , ⇧ B	– Drehung um den Blickvektor (vor-/rückwärts)
− , +	– Umschalten zwischen kleiner und großer Schrittweite für die Drehungen ($2\frac{1}{2}^\circ$ bzw. 10°)
0 , S	– Standardeinstellung auf Blickrichtung $(1, -1, 1)$ mit z–Achse nach oben
1	– Blick aus x–Richtung auf yz–Ebene
2	– Blick aus y–Richtung auf zx–Ebene
3	– Blick aus z–Richtung auf xy–Ebene
4	– Blickrichtung $(1, 1, 1)$ mit z–Achse nach oben
P	– Umschalten zwischen Parallel- und Perspektivprojektion
V , ⇧ V	– Nur bei Perspektivdarstellung: Abstand des Betrachters verkleinern oder vergrößern.
H	– Umschalten zwischen der Darstellung eines Hauses oder eines Koordinatendreibeins.
*	– Eingabe eines Skalierungsfaktors für die Darstellung. Die Skalierung `1.0` entspricht einer Abbildungsgröße, die bei jeder Lage im Raum die Raumdiagonalen des umschreibenden Quaders noch innerhalb des Grafikfensters darstellt, wodurch bei nahezu achsenparalleler Sicht i. a. eine relativ kleine Darstellung entsteht.
ESC	– Wiederherstellen der ursprünglichen Transformationsmatrix, mit der das Programm gerufen wurde. Aber: Nach dem Umschalten mit H gilt der zu diesem Zeitpunkt aktuelle Zustand als „ursprünglich“.
ENTER	– Beenden des Programms und Rückgabe der aktualisierten Transformationsmatrix

- Das vierte Untermenü $\boxed{\texttt{Option}}$ ist eine Sammlung weiterer mehr oder weniger nützlicher Funktionen:

  $\boxed{\texttt{Zoom}}$ Eingabe eines Zoom-Faktors für die Darstellung des Gebietes; es sind anzugeben: Mittelpunkt (x, y) und Radius des darzustellenden Ausschnitts (in Weltkoordinaten). Werden alle drei Werte Null gesetzt, so berechnet das Programm wieder einen Standardausschnitt anhand der Minima und Maxima der Koordinatenwerte.

  $\boxed{\texttt{Procs}}$ Auswahl von Prozessoren, die ihre Teilgebiete ausgeben sollen. Die Ausgabe aller übrigen Prozessoren wird dann unterdrückt.

  $\boxed{\texttt{Infos}}$ Anzeige verschiedener Statistiken zu Testzwecken (z. B. Gesamtkommunikationsaufwand für die grafische Darstellung; Balkendiagramm der Rechen- und Kommunikationszeiten, falls zuvor mittels `Init_Time` und `Get_Times` eine Zeitmessung erfolgt war).

  $\boxed{\texttt{Interv}}$ Eingabe eines Teilintervalls zwischen 0 und 1 für die Darstellung von Isolinien bzw. Stromlinien, Standard ist $(0, 1)$; damit können z. B. bestimmte Details deutlicher gemacht werden.

  $\boxed{\texttt{Scale}}$ Vorgabe einer Skala (`min,max`) für eine konstante Zuordnung von Farben zu absoluten Werten eines Freiheitsgrades (z. B. Temperatur) bei zeitlich fortschreitenden Lösungsdarstellungen; normalerweise wird die Farbskala bei jedem neuen Aufruf an das aktuelle Intervall der Lösungswerte angepaßt.

  $\boxed{\texttt{Optim}}$ Ein- und Ausschalter für eine Komprimierung der auszugebenden Grafikdaten. Die vielen kleinen Polygone gleicher Farbe werden zu größeren zusammengefaßt, wodurch der Datenumfang auf 30-50 % reduziert werden kann. Da die Berechnung relativ lange dauern kann, wird diese Option nur für die Postscript-Ausgabe empfohlen, da dort die Filegröße normalerweise im Megabyte-Bereich liegt.

  $\boxed{\texttt{Quiet}}$ Einschalten des *Quiet-Mode*, wodurch bei wiederholtem Aufruf von `firegraf` keine Abfragen mehr zu beantworten sind und auch die Menüzeile nicht mehr erscheint. Beim Start dieses *Quiet-Mode* kann gewählt werden, ob bei den weiteren Aufrufen von `firegraf` die Grafikanzeige erfolgen soll oder nicht. Dieser Modus kann jederzeit beendet werden, indem eine beliebige Taste (nicht Maustaste!) bei aktivem Grafikfenster gedrückt wird.

  $\boxed{\texttt{View}}$ Bei der Darstellungsart `Net-3D` kann damit das Koordinatensystem zur Manipulation des Betrachterstandpunktes aktiviert werden (s.o.).

- Der letzte Menüpunkt $\boxed{\texttt{go on}}$ veranlaßt die Fortsetzung des Programms, das `geb[m]graf` oder `firegraf` gerufen hatte.

- Folgende zusätzlichen Bedienfunktionen sind vorgesehen:

  - Ein Mausclick am oberen Rand *neben* der Menüzeile schaltet diese vorübergehend aus, was z. B. für eine Bildschirmkopie mittels `xv` oder anderen

Programmen zweckmäßig sein kann. Die Menüzeile erscheint wieder, nachdem die mittlere Maustaste gedrückt wurde.

- Die Ausgabe einer Darstellung kann abgebrochen werden, indem die mittlere Maustaste gedrückt gehalten wird, bis als Mauscursor zwei ineinander verschlungene Pfeile erscheinen. Diese sollen symbolisieren, daß die Grafikdaten trotzdem noch im Prozessorring weitergesendet werden (zur Vermeidung von Verklemmungen!), aber nicht mehr ausgegeben. Der gleiche Effekt wird auch mit Hilfe der `ESC`-Taste erreicht.
- Im Normalzustand (Menüzeile oben sichtbar) bewirkt auch die `ESC`-Taste wie `go on` das Verlassen des Programms.
- Es gibt eine Reihe von *„Hotkeys“*, mit denen im wesentlichen die gleichen Funktionen wie über die Menüpunkte aktiviert werden können. Da diese *„Geheimtasten“* als Überbleibsel zur Kompatibilität mit früheren Programmversionen anzusehen sind und für zukünftige Versionen nicht garantiert sind, werden sie auch hier nicht im einzelnen aufgeführt.

4 Beispiele

Als Anhang werden auf den folgenden Seiten einige Beispiele der verschiedenen Darstellungsvarianten gezeigt. Die Berechnungen erfolgten mit den in der Chemnitzer Forschergruppe „SPC“ entstandenen Testprogrammen für verschiedene Aufgabenstellungen ([2], [3], [4], [5], [6]).

Die Abbildungen wurden mittels der Postscript-Ausgabefunktion des Programms erzeugt und mit Hilfe des LaTeX-Stylefiles `epsfig.sty` dargestellt.

Literatur

[1] J. Gettys and R. W. Scheifler. Xlib - C language X interface, MIT Consortium Standard, X Version 11, Release 5. First revision, Digital Equipment Corporation and Massachussetts Institute of Technology, August 1991.

[2] U. Groh, C. Israel, S. Meinel, A. Meyer. On the numerical simulation of coupled transient problems on MIMD parallel systems. Preprint SPC 94_5, TU Chemnitz-Zwickau, 1994.

[3] G. Haase, T. Hommel, A. Meyer, M. Pester. Bibliotheken zur Entwicklung paralleler Algorithmen. Preprint SPC 94_4, TU Chemnitz-Zwickau, 1994.

[4] G. Haase, U. Langer, A. Meyer, S. V. Nepomnyaschikh. Hierarchical extension and local multigrid methods in domain decomposition preconditioners. Preprint SPC 94_8, TU Chemnitz-Zwickau, 1994.

[5] A. Meyer. Preconditioning the Pseudo-Laplacian for finite element simulation of incompressible flow. Preprint SPC 94_19, TU Chemnitz-Zwickau, 1994.

[6] A. Meyer and M. Pester. Verarbeitung von Sparse-Matrizen in Kompaktspeicherform (KLZ/KZU). Preprint SPC 94_12, TU Chemnitz-Zwickau, 1994.

[7] A. Nye. *Xlib Programming Manual for Version X11.* O'Reilly & Associates, Inc., 1990.

[8] Y. Saad and M. H. Schultz. Topological properties of hypercubes. Research Report 389, Yale University, Dept. Computer Science, 1985.

[9] Y. Saad and M. H. Schultz. Data communication in hypercubes. *Journal of parallel and distributed computing*, 6 : 115–135, 1989.

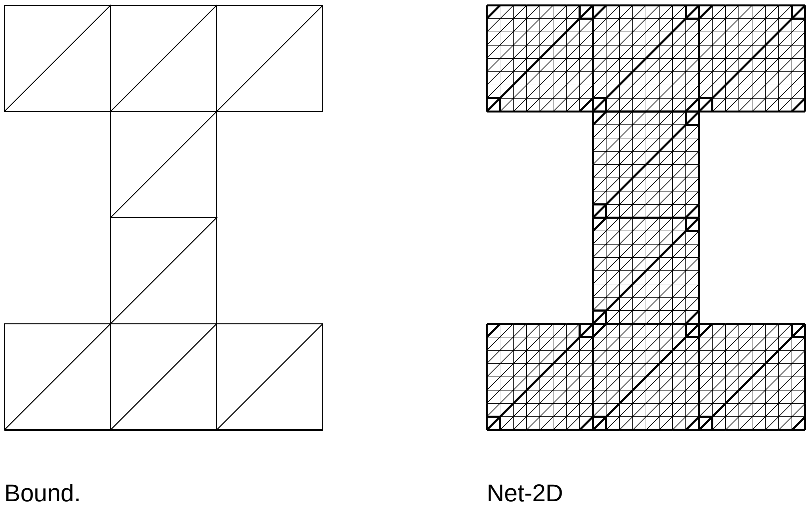

Abbildung 2: Netzdarstellungen

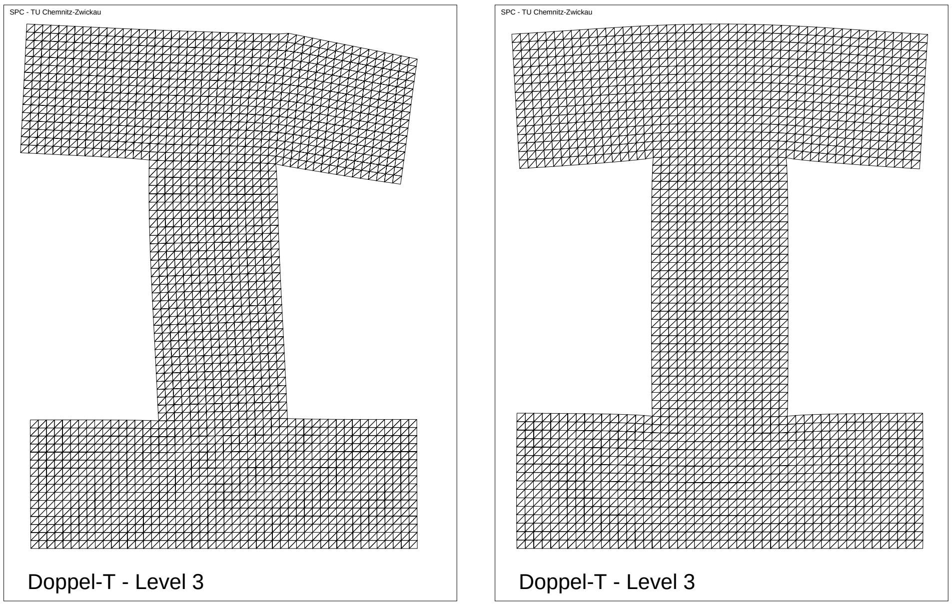

Abbildung 3: Deformiertes Netz bei verschiedenen Belastungen

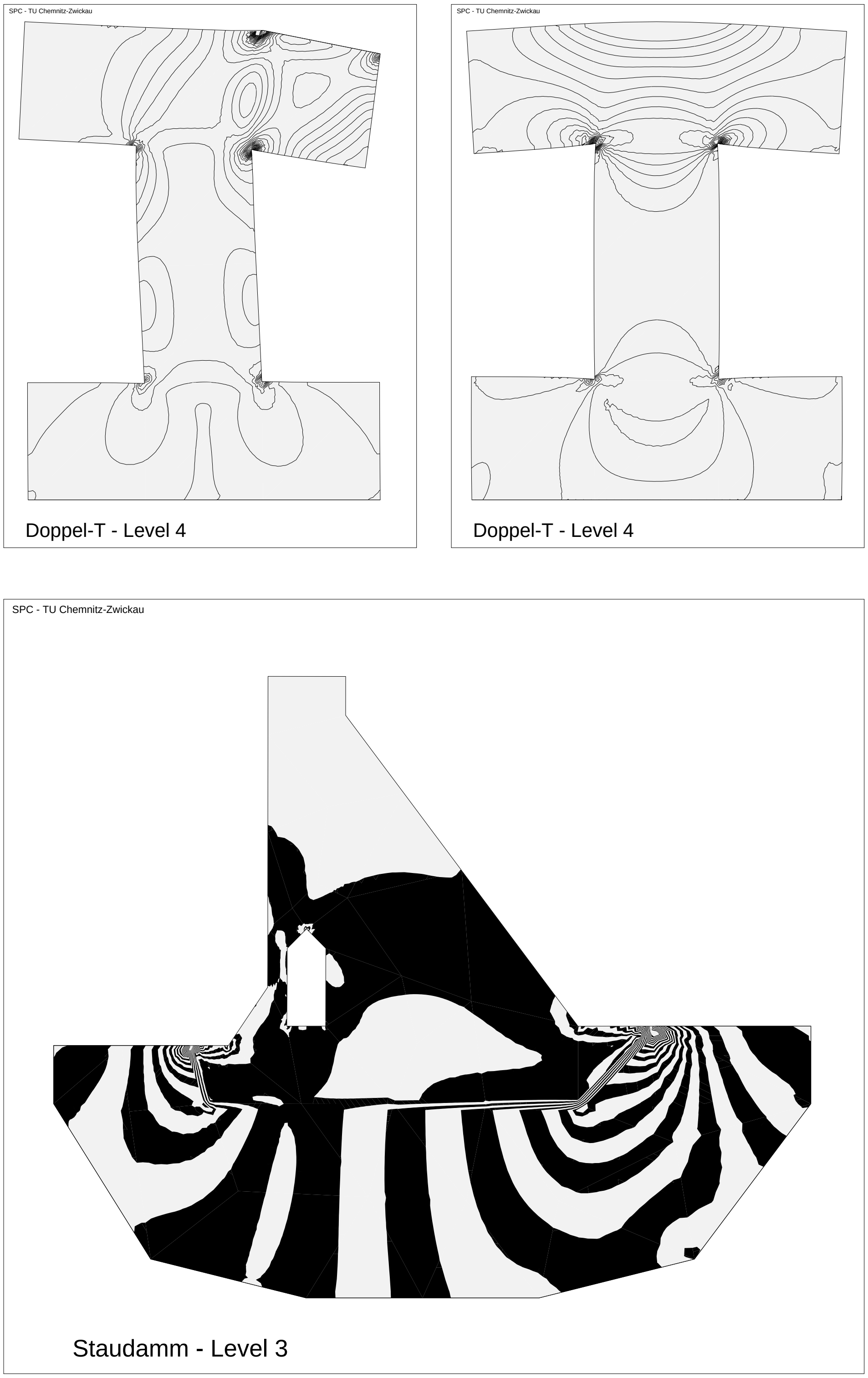

Abbildung 4: Spannungen und Verzerrungen

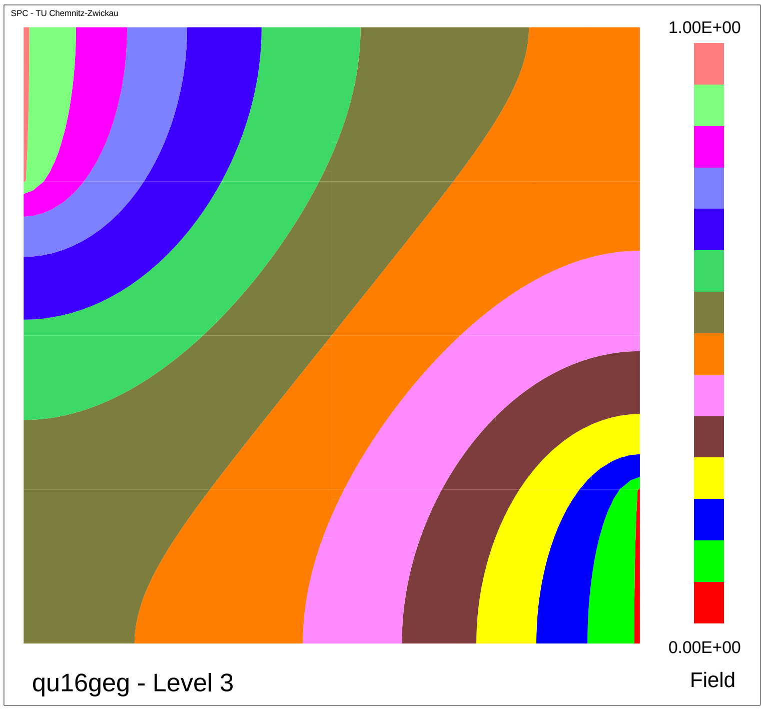

Abbildung 5: Isoflächen (`Filled`)

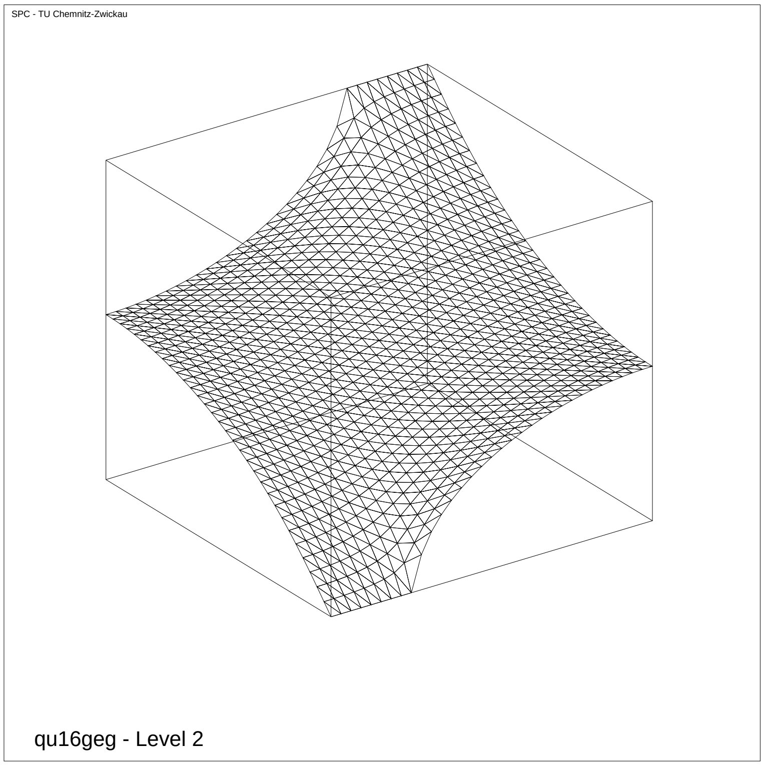

Abbildung 6: Gebirgsdarstellung (`Net-3D`)

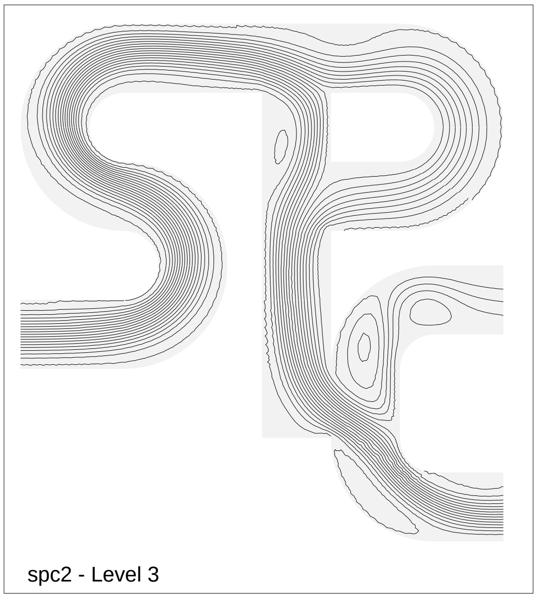

Abbildung 7: Stromliniendarstellung

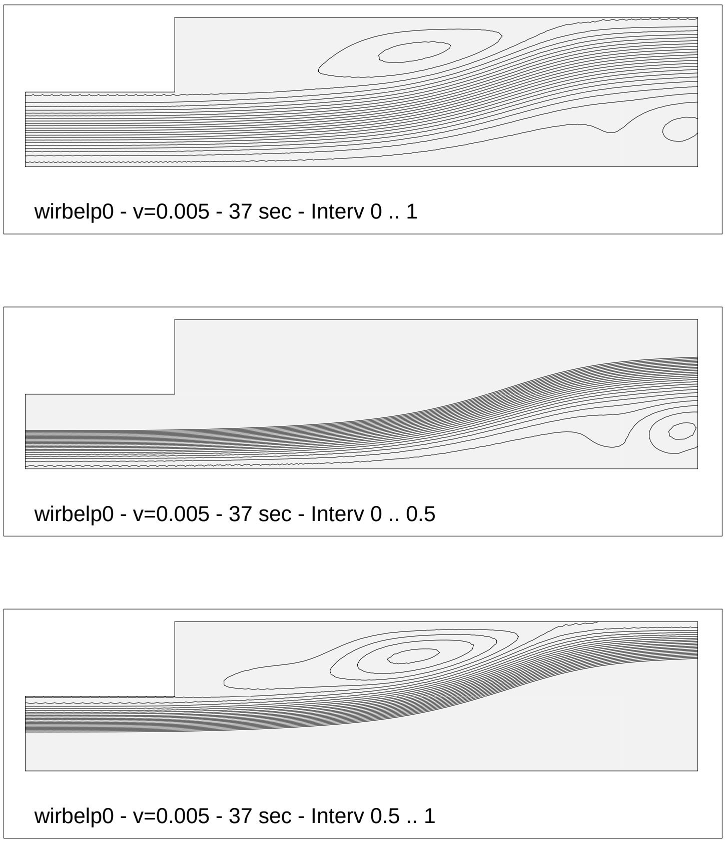

Abbildung 8: Stromlinien in verschiedenen Bereichen

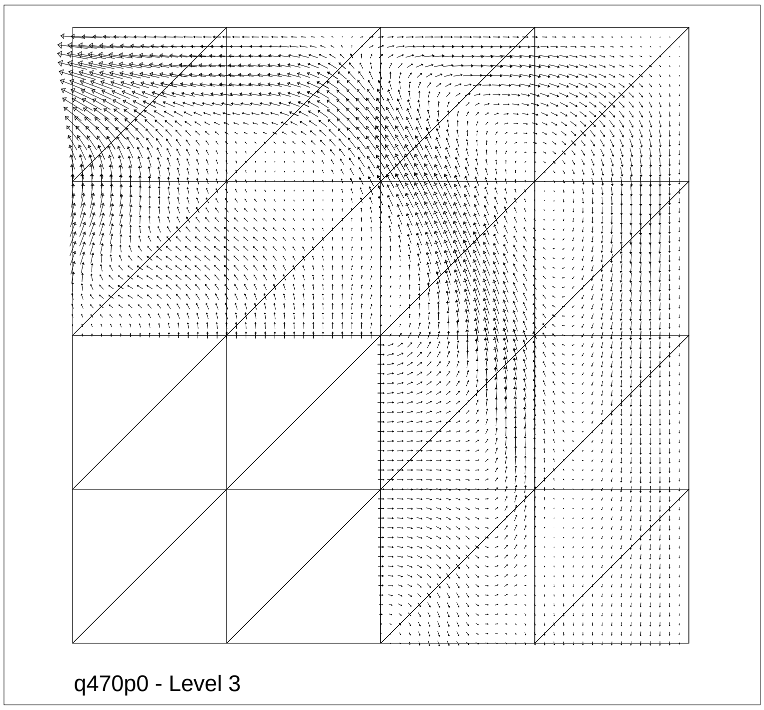

Abbildung 9: Geschwindigkeitsfeld

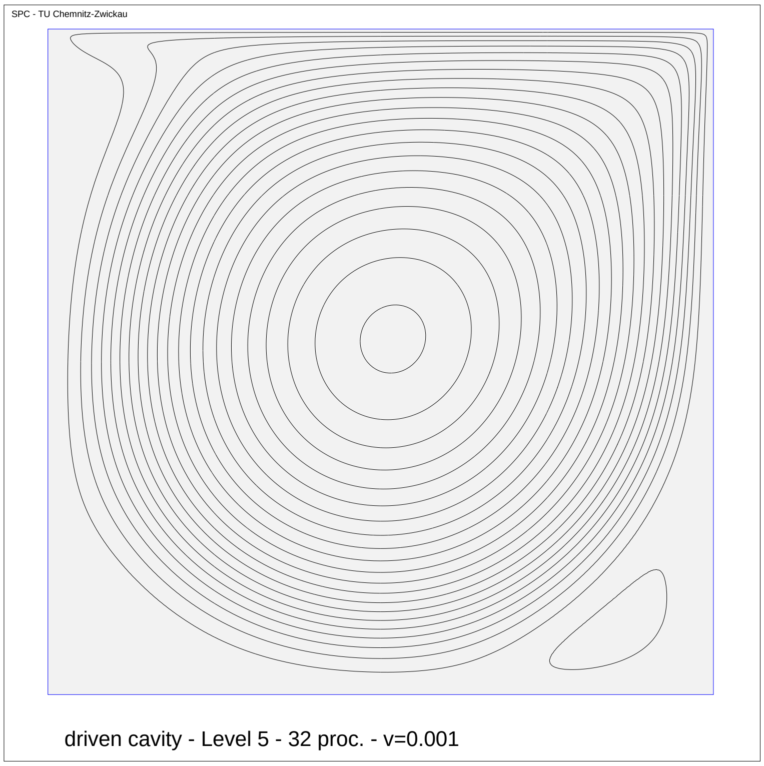

Abbildung 10: normale Stromliniendarstellung

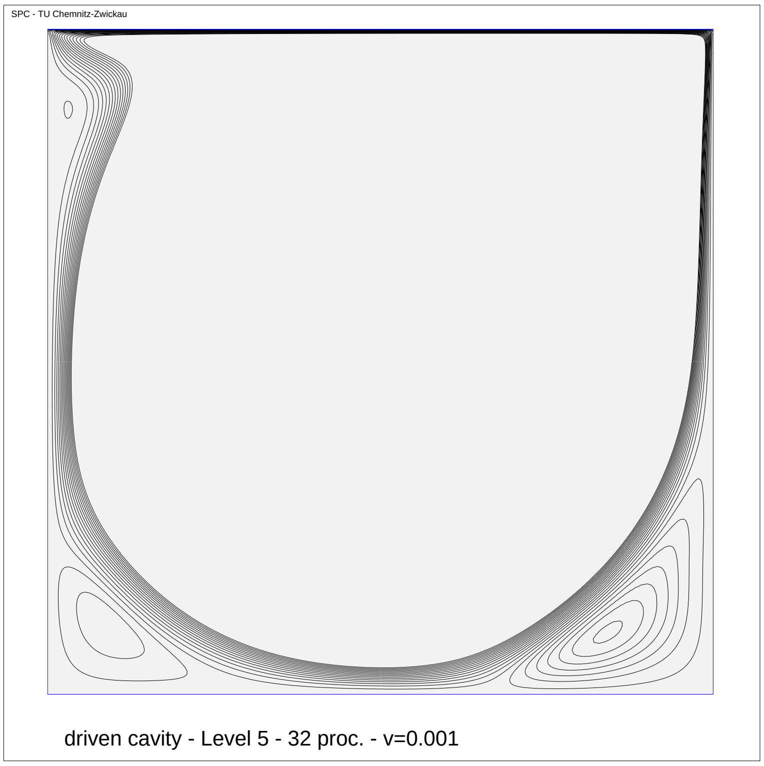

Abbildung 11: Detailhervorhebung durch `Interv`

Abbildung 12: Detailhervorhebung durch `Interv` und `Zoom`

Other titles in the SPC series:

93_1 G. Haase, T. Hommel, A. Meyer and M. Pester. Bibliotheken zur Entwicklung paralleler Algorithmen. May 1993.

93_2 M. Pester and S. Rjasanow. A parallel version of the preconditioned conjugate gradient method for boundary element equations. June 1993.

93_3 G. Globisch. PARMESH – a parallel mesh generator. June 1993.

94_1 J. Weickert and T. Steidten. Efficient time step parallelization of full-multigrid techniques. January 1994.

94_2 U. Groh. Lokale Realisierung von Vektoroperationen auf Parallelrechnern. March 1994.

94_3 A. Meyer. Preconditoning the Pseudo-Laplacian for Finite Element simulation of incompressible flow. February 1994.

94_4 M. Pester. Bibliotheken zur Entwicklung paralleler Algorithmen. (aktualisierte Fassung). March 1994.

94_5 U. Groh, Chr. Israel, St. Meinel and A. Meyer. On the numerical simulation of coupled transient problems on MIMD parallel systems. April 1994.

94_6 G. Globisch. On an automatically parallel generation technique for tetrahedral meshes. April 1994.

94_7 K. Bernert. Tauextrapolation – theoretische Grundlagen, numerische Experimente und Anwendungen auf die Navier-Stokes-Gleichungen. June 1994.

94_8 G. Haase, U. Langer, A. Meyer and S. V. Nepomnyaschikh. Hierarchical extension and local multigrid methods in domain decomposition preconditoners. June 1994.

94_9 G. Kunert. On the choice of the basis transformation for the definition of DD Dirichlet preconditioners. June 1994.

94_10 M. Pester and T. Steidten. Parallel implementation of the Fourier Finite Element Method. June 1994.

94_11 M. Jung and U. Rüde. Implicit Extrapolation Methods for Multilevel Finite Element Computations: Theory and Applications. June 1994.

94_12 A. Meyer and M. Pester. Verarbeitung von Sparse-Matrizen in Kompaktspeicherform (KLZ/KZU). June 1994.

94_13 B. Heinrich and B. Weber. Singularities of the solution of axisymmetric elliptic interface problems. June 1994.

94_14 K. Gürlebeck, A. Hommel and T. Steidten. The method of lumped masses in cylindrical coordinates. July 1994.

94_15 Th. Apel and F. Milde. Realization and comparison of various mesh refinement strategies near edges. August 1994.

94_16 Th. Apel and S. Nicaise. Elliptic problems in domains with edges: anisotropic regularity and anisotropic finite element meshes. August 1994.

94_17 B. Heinrich. The Fourier-finite-element method for Poisson's equation in axisymmetric domains with edges. August 1994.

94_18 M. Pester and S. Rjasanow. A parallel preconditioned iterative realization of the panel method in 3D. September 1994.

94_19 A. Meyer. Preconditoning the Pseudo-Laplacian for Finite Element simulation of incompressible flow. October 1994.

94_20 V. Mehrmann. A step towards a unified treatment of continous and discrete time control problems. October 1994.

94_21 C. He, V. Mehrmann. Stabilization of large linear systems. October 1994.

94_22 B. Heinrich and B. Weber. The Fourier-finite-element method for three-dimensional elliptic problems with axisymmetric interfaces. November 1994.

94_23 M. Pester. On-line visualization in parallel computations. November 1994.

Some papers can be accessed via anonymous ftp from server `ftp.tu-chemnitz.de`, directory `pub/Local/mathematik/SPC`. (Note the capital L in Local!)