



---

**Forschungszentrum Karlsruhe**  
Technik und Umwelt

---

**Wissenschaftliche Berichte**  
FZKA 5999

**Das VISART-Konzept einer  
standardisierten Schnittstelle  
zwischen Codes und  
Auswerteprogrammen**

**Teil 5:  
VISARVOL – ein Programm zur  
Visualisierung von Volumenanteilen  
aus VISART-Dateien**

**S. Kleinheins**

Institut für Neutronenphysik und Reaktortechnik  
Projekt Nukleare Sicherheitsforschung

**Mai 1999**

---



Forschungszentrum Karlsruhe  
Technik und Umwelt  
Wissenschaftliche Berichte  
FZKA 5999

Das VISART-Konzept einer standardisierten  
Schnittstelle zwischen Codes und Auswerteprogrammen

Teil 5: VISARVOL—ein Programm zur Visualisierung  
von Volumenanteilen aus VISART-Dateien

Siegfried Kleinheins

Institut für Neutronenphysik und Reaktortechnik  
Projekt Nukleare Sicherheitsforschung

Forschungszentrum Karlsruhe GmbH, Karlsruhe  
1999

**Als Manuskript gedruckt**  
**Für diesen Bericht behalten wir uns alle Rechte vor**  
**Forschungszentrum Karlsruhe GmbH**  
**Postfach 3640, 76021 Karlsruhe**  
**Mitglied der Hermann von Helmholtz-Gemeinschaft**  
**Deutscher Forschungszentren (HGF)**  
**ISSN 0947-8620**

## **Zusammenfassung**

VISART ist ein standardisiertes Format für Postprocessor-Dateien mit Daten, wie sie von Fluidodynamik- und ähnlichen Codes über einer mehrdimensionalen Geometrie und der Zeit errechnet werden. Das Programm VISARVOL visualisiert Daten im VISART-Format, die Volumenanteile von Materialien in den Maschen zweidimensionaler Netze oder Netzquerschnitte und die Durchlässigkeit von Wänden auf den Maschenhüllen beschreiben, durch Umsetzung in ein von dem kommerziellen Visualisierungssystem Tecplot-7 erwartetes Format. Über die VISARVOL-Eingabe gesteuert, können dabei aus den VISART-Daten entsprechende Größen ausgewählt werden und für die Darstellung als Volumenanteile (codiert mit Farben oder Symbolen, entweder in Schichten oder über die jeweilige Masche verteilt) oder als Wanddurchlässigkeiten (codiert mit Farbsättigung, Strichstärke oder Lückenweite) aufbereitet werden. Die von VISARVOL erstellte Ausgabedatei kann von Tecplot-7, gesteuert über eine bereitgestellte Script-Datei, unmittelbar eingelesen und visualisiert werden.

## **The VISART Concept of a Standardized Interface between Codes and Evaluation Programs**

### **Part 5: VISARVOL—a Program for Visualizing Volume Fractions from VISART Files**

#### **Abstract**

VISART is a standardized format for postprocessor files with data calculated by fluid-dynamics and similar codes over a multi-dimensional geometry and time. The VISARVOL program visualizes VISART format data describing material volume fractions in the cells of two-dimensional meshes or mesh cross sections and wall permeabilities on the cell boundaries, by transforming them into a format expected by the Tecplot-7 commercial visualization system. Controlled by VISARVOL input, in this process corresponding quantities from the VISART file can be selected and prepared for the representation as volume fractions (coded by colors or symbols, either in layers or distributed over the cell) or as wall permeabilities (coded by color saturation, line thickness or gap width). The output file generated by VISARVOL can directly be read in and visualized by Tecplot-7, controlled by a supplied script file.



# Übersicht über das VISART-Konzept und seine Dokumentation

## Welche Aufgabe ist zu lösen ?

Die rechnerische Simulation von Problemen der Fluidodynamik, Elektrodynamik, Neutronik usw. reduziert sich meist auf die numerische Lösung von zeitabhängigen partiellen Differentialgleichungen mittels räumlicher und zeitlicher Diskretisierung. Die dafür verwendeten Computer-Codes errechnen dabei eine Fülle von Daten über der Geometrie und der Zeit. Zu ihrer Auswertung sind diese Daten in geeigneter Form auf einer Postprocessor-Datei bereitzustellen.

## Was ist VISART ?

VISART ist in erster Linie der Name eines (zunächst im FZK-Bereich) standardisierten Formats für Postprocessor-Dateien mit nach obiger Art errechneten Daten, die zur Auswertung erstellt werden, aber noch keine Steuerinformation für Grafik usw. enthalten. Im weiteren Sinne ist VISART der Name eines auf dieser Standardisierung aufbauenden Konzepts einer einheitlichen Schnittstelle zwischen Codes und Auswerteprogrammen, die es erlaubt, mit minimalem Implementierungsaufwand einer Vielzahl von dafür geeigneten Codes eine Vielzahl von Diensten zur Weiterbehandlung der berechneten Daten zur Verfügung zu stellen.

## Welche Codes erstellen VISART-Dateien ?

Postprocessor-Ausgabe im VISART-Format wurde (Stand Dezember 1998) bereits in die folgenden, im FZK betreuten oder entwickelten Codes eingebaut: SIMMER-II.12, AFDM, SIMMER-III, HYDSOL, MATTINA, FLUTAN, KADI2D, MAX3D, BFCPIC, GASFLOW.

## Was kann man mit VISART-Dateien anfangen ?

VISART-Dateien können zunächst einmal mit einem Dienstprogramm in wählbarem Umfang und Detail „durchblättert“ werden. Mit einem anderen Programm können die Werte ausgewählter Größen gezielt über der Geometrie und/oder der Zeit in Zahlenform „angeschaut“ werden.

Die graphische und anderweitige Auswertung von Größen einer VISART-Datei ist sodann mit einigen im Hause oder im Auftrag entwickelten Auswerteprogrammen möglich, die das VISART-Dateiformat direkt lesen können.

Die Visualisierung der Daten einer VISART-Datei mit kommerziellen Visualisierungssystemen (bis jetzt AVS/Express, Gsharp, Tecplot-7), die jeweils eigene Formate für ihre Eingabedateien verlangen, geschieht über das Zwischenschalten eines für VISART angefertigten Adapterprogramms, das die Eingabedatei des jeweiligen Systems für die gerade gewählten Größen und Darstellungskordinaten (z.B. über der Geometrie—auch Schnitten oder Ausschnitten daraus—und/oder der Zeit) erstellt.

Schließlich steht ein mächtiges Werkzeug zum Bearbeiten von VISART-Dateien und zum Ableiten neuer Größen aus den in der VISART-Datei enthaltenen Daten zur Verfügung.

Alle erwähnten Dienste können über eine zentrale graphische Oberfläche auf X-Terminals für Workstations aufgerufen und gesteuert werden.

## **Wo findet man was ?**

Teil 1 der VISART-Dokumentation führt in das VISART-Konzept ein und stellt die graphische Benutzeroberfläche zum Aufruf der Dienste vor. Das Programm VISARTUL zum Durchblättern und Prüfen von VISART-Dateien wird beschrieben, und es wird ein Überblick über die anderen zur Verfügung stehenden Dienste gegeben. (Dieser Teil ist die Grundlage für die nachfolgenden Teile.)

Teil 2 der VISART-Dokumentation definiert den VISART-Standard für den Aufbau der Postprocessor-Dateien.

Teil 3 der VISART-Dokumentation beschreibt das Adapterprogramm VISAPTER zum Erstellen von Eingabedateien für die Visualisierung mit AVS/Express, Gsharp, Tecplot-7 usw. und zum Anschauen ausgewählter Werte in Zahlenform.

Teil 4 der VISART-Dokumentation beschreibt das Werkzeug VISARTOP zum Bearbeiten von VISART-Dateien und zum Ableiten neuer Größen.

Teil 5 der VISART-Dokumentation beschreibt das Adapterprogramm VISARVOL zum Erstellen von Eingabedateien für die zweidimensionale Darstellung von Volumenanteilen mit Tecplot usw..

Auswerteprogramme, die VISART-Dateien direkt lesen und verarbeiten können, werden nicht in dieser Reihe, sondern eigenständig dokumentiert. Das gleiche gilt für den Einbau der VISART-Schnittstelle in die einzelnen Codes.



## Inhalt

1. Einleitung	1
2. Fähigkeiten des Programms	3
2.1 Visualisierbare Größen	3
2.2 Visualisierungssysteme	3
2.3 Darstellungskordinaten	3
2.4 Darstellungsweisen	3
3. Steuerung des Programms	4
3.1 Steuereingabe	4
3.2 Anmerkungen zu den Eingabeparametern	8
3.3 Eingabebeispiele	9
3.4 Eingabe über die Benutzeroberfläche	10
4. Ein- und Ausgabedateien des Programms	12
4.1 VISART-Dateistruktur und -inhalt	12
4.2 Ausgabedatei	13
5. Technik des Programms	14
5.1 Programmoptionen	14
5.2 Programmaufruf und -argumente	15
5.3 Programmaufbau	17
5.4 Einige Programmdetails	18
5.5 Einschränkungen usw.	19
5.6 Aufbau der Benutzeroberfläche	19
6. Visualisierung mit Tecplot-7	20
6.1 Besonderheiten	20
6.2 Aufbau der Tecplot-Datei	20
6.3 Tecplot-Aufruf	23
6.4 Beispiele	24
Literatur	27

## 1. Einleitung

VISART ist der Name eines (zunächst im FZK-Bereich) standardisierten Formats für Postprocessor-Dateien [1b] mit über der Geometrie und der Zeit errechneten Daten, wie sie z.B. von Fluidodynamik-Codes zur anschließenden grafischen oder sonstigen Auswertung erstellt werden.

Im VISART-Konzept [1a] einer einheitlichen Schnittstelle zwischen Codes (Fluidodynamik- und verwandte Programme) und Auswerteprogrammen schreiben die Codes die errechneten auszuwertenden Daten auf eine Postprocessor-Datei im VISART-Format. Zur Auswertung der Daten mit einem kommerziellen Visualisierungssystem müssen sie zunächst in das vom jeweiligen System erwartete Eingabedateiformat umgesetzt werden. In diesem Prozess kann man auch eine Auswahl aus den in der Datei enthaltenen Daten in Bezug auf Größen und Darstellungskordinaten treffen.

Das hier vorgestellte Programm VISARVOL<sup>1</sup> dient zur Visualisierung von Größen aus VISART-Dateien, die Volumenanteile von Materialien in den Maschen von Netzen und (Un)Durchlässigkeiten von Wänden auf den Maschenhüllen beschreiben. Die Darstellung ist über zweidimensionalen, ebenen, kartesischen Netzen oder Teilgebieten möglich (bei Wänden allein auch über Polarkordinaten) und wird mit dem kommerziellen Visualisierungssystem Tecplot-7 realisiert. Nach Maßgabe der Steuereingabe von VISARVOL können geeignete Größen aus der VISART-Datei ausgewählt und ggf. ein geeignetes Teilgebiet des Netzes für einen bestimmten Problemzeitpunkt ausgeschnitten werden. Die so selektierten Daten werden dann für Tecplot aufbereitet und im von diesem System geforderten Format auf eine Datei ausgegeben, die als Eingabe für das Visualisierungssystem dient (siehe Bild 1).

VISARVOL ist ein Fortran-Programm, das im Stapelbetrieb aufgerufen wird, und zwar am bequemsten über eine grafische Benutzeroberfläche, mit der die Steuereingabe zusammengestellt wird. Die grafische Benutzeroberfläche wurde bereits im Teil 1 der VISART-Dokumentation [1a] vorgestellt. Im vorliegenden Bericht, dem Teil 5 der VISART-Dokumentation, werden VISARVOL, seine Steuereingabe, seine Ein- und Ausgabedateien und die Anwendung für Tecplot beschrieben.

Zum Verstehen des Berichts ist die Vertrautheit mit dem einführenden Teil 1 der Dokumentation [1a] Voraussetzung, besonders bezüglich der verwendeten Terminologie. Allerdings haben sich seit dessen Abfassung einige Änderungen ergeben, die besonders beim Vergleich der Bilder des Teils 1 und dieses Berichts auffallen. So ist das in [1a] zugrunde gelegte Visualisierungssystem Tecplot, Version 6, inzwischen durch die Nachfolgeversion 7 ersetzt worden (daneben wurde der Code IVA-KA in MATTINA umbenannt). Die Darstellung von Volumenanteilen wurde seither um die Darstellung von Wänden erweitert. Das Programm VISARVOL und der zugehörige Teil der VISART-Oberfläche ist nun hier auf aktuellem Stand dokumentiert.

---

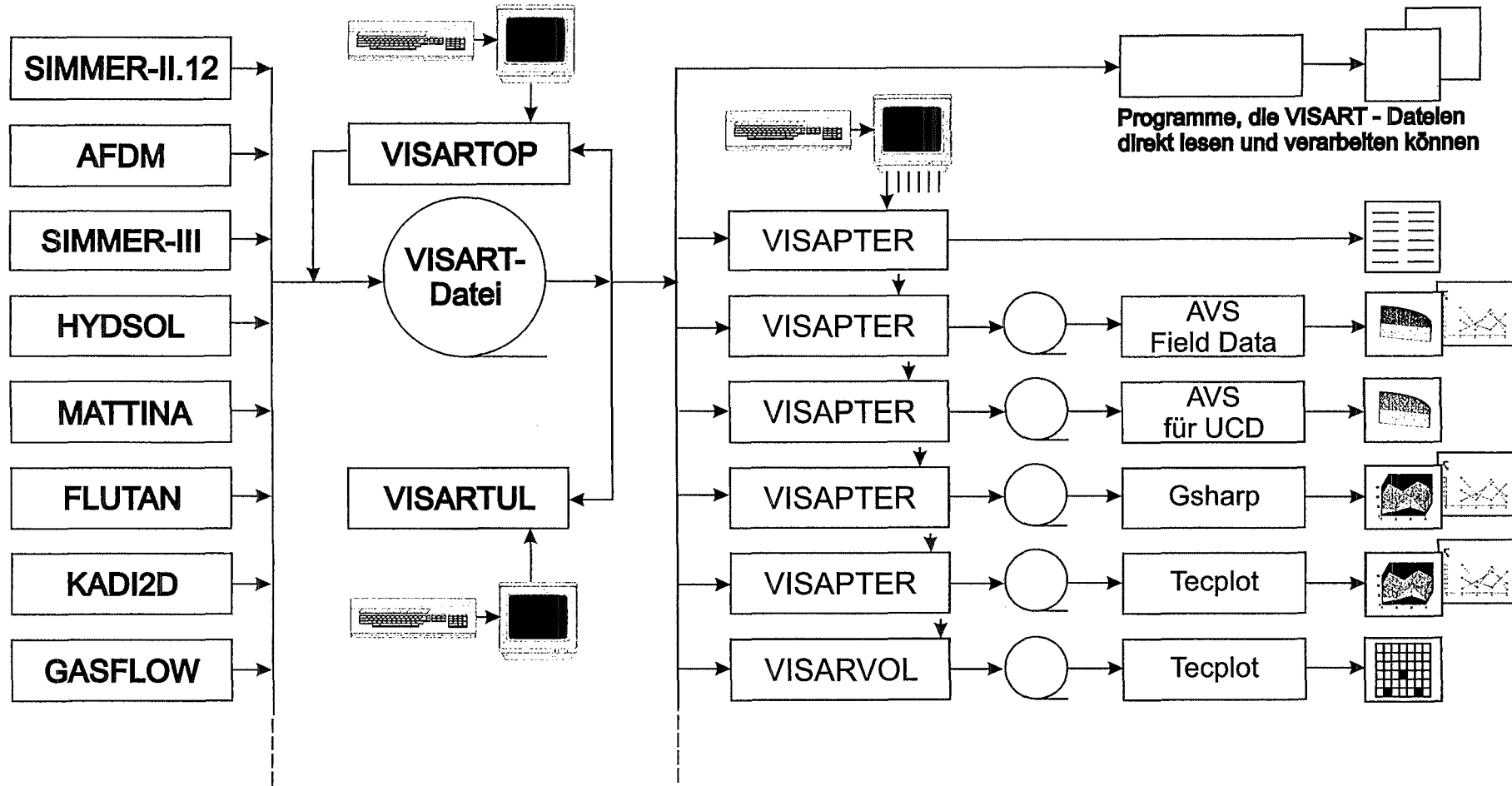
<sup>1</sup> von „VISART-Volumenanteile“

# Codes

# VISART - Werkzeuge

# Adapter

# Visualisierungssysteme



**Bild 1 : Datenfluß über die VISART - Schnittstelle**

## 2. Fähigkeiten des Programms

In diesem Kapitel werden die Visualisierungsoptionen des Programms VISARVOL vorgestellt.

### 2.1 Visualisierbare Größen

Mit VISARVOL sind Größen visualisierbar, die die **Volumenanteile von Materialien** in den Maschen der Netze und die **(Un)Durchlässigkeit von Wänden** auf den Maschenhüllen beschreiben. Die Werte dieser Größen müssen in jeder Masche im Bereich zwischen 0. und 1. liegen. Falls die Volumenanteile mehrerer Materialien zusammen dargestellt werden sollen, darf ihre Summe in keiner Masche den Wert 1. übersteigen. Der Wert 1. für die Undurchlässigkeit entspricht einer geschlossenen Wand, der Wert 0. einer fehlenden Wand; der Wert 1. für die Durchlässigkeit (oder Permeabilität) entspricht einer fehlenden Wand, der Wert 0. einer geschlossenen Wand.

### 2.2 Visualisierungssystem

Im Gegensatz zum Adapterprogramm VISAPTER [1c], das die Visualisierung von Größen aus VISART-Dateien mit den Visualisierungssystemen AVS/Express, Gsharp und Tecplot-7 ermöglicht, arbeitet VISARVOL nur mit Tecplot-7 zusammen.

### 2.2 Darstellungskordinaten

Mit VISARVOL können Netzgrößen (Gruppen 5/15) und Maschengrößen (Subgruppen 7/17) über Geometrien verarbeitet werden, die auf regulären, vollen oder nicht-vollen, **Netzen** (siehe [1a, 1b]) beruhen. Volumenanteile können nur über zweidimensionalen, ebenen und kartesischen Netzen oder Teilgebieten solcher Geometrien dargestellt werden, Wände allgemeiner über zweidimensionalen, ebenen Netzen oder Teilgebieten solcher Geometrien (d.h. auch über Polarkoordinaten). VISARVOL kann, je nach Steuereingabe, aus den Netzen solche geeignete Teilgebiete, falls vorhanden, ausschneiden.

Teilgebiete sind hier analog zu den Teilnetzen (siehe [1a, 1b]) definiert. Während Teilnetze zur Zusammensetzung eines Netzes in der VISART-Datei gebraucht werden, dienen Teilgebiete zur Zerlegung des Netzes für die Visualisierung. Ein **Teilgebiet** ist ein Ausschnitt aus einem (vollen oder nicht-vollen) Netz, der durch Festlegung von Anfangs- und Endmaschenindizes innerhalb des Netzes in einer oder mehreren Koordinatenrichtungen entsteht. Fallen Anfangs- und Endmaschenindex eines Teilgebiets in einer oder mehreren Koordinatenrichtungen zusammen, so sprechen wir von einem degenerierten Teilgebiet. Ein degeneriertes Teilgebiet läßt sich auch als ein **Subgebiet** auffassen. Ein Subgebiet entsteht durch einen Querschnitt (oder Querschnitte) durch das Netz, d.h. durch Konstanthalten einer oder mehrerer Koordinaten des Systems. Ein Subgebiet hat deshalb eine „Subdimension“ kleiner als die Netzdimension (im Extremfall die Subdimension Null).

Die Netzkoordinaten des ausgewählten Teilgebiets können mit VISARVOL ggf. bearbeitet werden. Netze mit krummlinigen Koordinaten lassen sich, wenn dies vom Visualisierungssystem her erforderlich ist, für die Darstellung auf kartesische Koordinaten **umrechnen**. 2-dimensionale  $\vartheta, z$ -Netze/Teilgebiete/Subgebiete lassen sich auf eine Ebene **abwickeln**. 2-dimensionale  $r, z$ -Netze/Teilgebiete lassen sich an der  $z$ -Achse **spiegeln**.

### 2.4 Darstellungsweisen

Den Ort, in dem eine Größe der Gruppen 5/15 oder der Subgruppen 7/17 in Bezug auf die Maschen dargestellt werden soll, entnimmt VISARVOL der in der Gruppe/Subgruppe stehenden Lokationsangabe [1a, 1b]. Volumenanteile sind über Lokationen in den Maschenmitten (Lokationsindikator 0) gegeben, Wände üblicherweise über Lokationen auf den Maschenhüllen (Lokationsindikatoren 11, 22, 44, 01, 02, 04, 10, 20, 40) <sup>1</sup>. VISARVOL stellt das Netzgitter und die Netzgeometrie geometrisch richtig und vollständig dar, und entspricht in dieser Hinsicht einer „cell-based“ Darstellung des Netzes in VISAPTER [1c].

<sup>1</sup> VISARVOL akzeptiert auch Lokationsindikatoren 0 für Wände, die dann als skalare oder vektorielle Größen definiert sein können. Siehe dazu 3.2 und 5.5.)

### 3. Steuerung des Programms

In diesem Kapitel wird die Steuereingabe des Programms VISARVOL beschrieben.

#### 3.1 Steuereingabe

Das Programm ist für den Stapelbetrieb entworfen; es erwartet eine komplette Steuereingabe auf einer Datei und läuft vom Start bis zum Programmende oder Fehlerabbruch ohne Wechselwirkung mit dem Benutzer.

Hier wird der Aufbau der Steuereingabe beschrieben, wie sie vom Benutzer zu erstellen ist bzw. wie sie von der grafischen Benutzeroberfläche (siehe 3.4) aus der Einstellung der „Widgets“ erzeugt wird.

In der Beschreibung der Eingabe entspricht jede Zeile *einem* (logischen) Satz der Steuereingabe. Die Eingabe wird listengesteuert gelesen; mit I, . . . , N beginnende Namen bezeichnen INTEGER-Variable, mit C beginnende Namen bezeichnen CHARACTER-Variable aus bis zu 8 Zeichen, alle anderen Namen bezeichnen REAL-Variable.

1. MODE IKRUM

(Das folgende Zeilenarrangement 2 bis 5 wird für jede VISART-Datei—siehe 4.1—wiederholt:)

2. NF NP NT NN

Wenn  $NT \neq 0$  ( $\wedge |NT| \leq 20$ ): eine Zeile 3:

3. T1(1) T2(1) . . . T1(|NT|) T2(|NT|)

4. NG KOR1(1) KOR2(1) KOR1(2) KOR2(2) KOR1(3) KOR2(3)

NN Zeilen 5:

5. CN CG IPHASE ISYMB ICLOR IGRADE

Das Programm VISARVOL erwartet die Steuereingabe als Standardeingabe. Wenn die Steuereingabe direkt mit der Tastatur eingegeben wird, springt das Programm nach dem Verarbeiten der letzten Eingabezeile wieder auf die Zeile 2 und erwartet Eingabe. Dort ist dann die Zeile

-1 0 0 0

für das Eingabeende einzutippen. Wenn die Steuereingabe von einer umgelenkten Datei kommt, wird das Dateiende (EOF) als Ende der Eingabe erkannt.

Bedeutung der Eingabeparameter:

MODE	wählt das Visualisierungssystem, für das die Ausgabe erstellt werden soll: 5: Tecplot
IKRUM	gibt bei krummlinigen Koordinaten die Art der Koordinatenumrechnung bzw. bei $r, z$ -Netzen/Teilgebieten die Spiegelungsoption an: 0: keine Umrechnung bzw. keine Spiegelung; 1: Umrechnung der Vektorkomponenten und Koordinaten auf kartesische Koordinaten; 2: bei Zylindermantel und Kreislinie: Umrechnung der Koordinaten auf die (kartesischen) Koordinaten der Abwicklung; -1: Spiegelung des rechteckigen $r, z$ -Netzes/Teilgebiets an der $z$ -Achse.
NF	> 0: Anzahl der zu verkettenden Fortsetzungs-Files der VISART-Datei (siehe 4.1); -1: für das Ende der Steuereingabe (s.u.).
NP	$\geq 0$ : Maximale Anzahl der auszugebenden Rumpf-Pakete; < 0: die Rumpf-Pakete, deren Zyklusnummern Vielfache von $ NP $ sind, werden ausgegeben.
NT	0: Alle in der Datei enthaltene Rumpf-Pakete werden (nach Maßgabe von NP) ausgegeben; $\neq 0 \wedge  NT  \leq 20$ : nur die Rumpf-Pakete in den durch T1, T2 spezifizierten Problemzeitintervallen werden (nach Maßgabe von NP) ausgegeben; $> 20 \vee < -20$ : nur das letzte Rumpf-Paket in der Datei wird ausgegeben.
NN	> 0: Anzahl der durch Zeile 5 spezifizierten auszugebenden Größen.
T1( $m$ )	( $m=1, \dots,  NT $ ;) spezifizieren die ausgewählten Problemzeitintervalle:
T2( $m$ )	bei NT > 0: T1( $m$ ) und T2( $m$ ) begrenzen das $m$ -te Intervall; bei NT < 0: T1( $m$ ) und T2( $m$ ) definieren eine gleichmäßige Folge von Intervallen (siehe 3.2).
NG	Gruppen-Kennzahl der auszugebenden Größe: 5/15: Netzgröße; 7/17: Maschengröße.
KOR1( $n$ )	( $n = 1, 2, 3$ ;) geben die Maschenindizes jeder Koordinatenrichtung ( $i, j, k$ ) für die untere bzw. obere Grenze des auszugebenden Teilgebiets an: -1, -1: wenn keine Grenzen gesetzt sind oder die Koordinate fehlt; $N, -1$ : wenn beide Grenzen auf $N$ zusammenfallen; $N, M$ : wenn die Grenzen $N \leq M$ gesetzt werden.
KOR2( $n$ )	
CN	Identifikation der auszugebenden Größe (Volumenanteil eines Materials oder Durchlässigkeit einer Wand).
CG	bei NG = 7/17: Identifikation der Referenzgruppe 6 oder 16, auf die sich die Subgruppe für CN bezieht; andernfalls: leer.

- IPHASE gibt an, ob die Größe ein Material oder eine Wand beschreibt, und gibt die Darstellung der Phase des Materials an bzw. bezeichnet für Wände ggf. die Lokation der Wand und die Interpretation der Durchlässigkeit:
- 0: Material über den von den Schichten nicht eingenommenen Raum der Masche verteilt;
  - 1: Material in einer dem Volumenanteil entsprechenden Schicht, im untersten Teil der Masche;
  - 2: Material in einer dem Volumenanteil entsprechenden Schicht, im nächstuntersten Teil der Masche;
  - ...
  - 11: Wand an der durch die Lokation gegebenen Maschenhülle (bei Lokation 0 an der rechten Maschenhülle (skalar) bzw. an der rechten und oberen Maschenhülle (vektoriell)), wobei die Undurchlässigkeit von 0 bis 1 wächst.
  - 12: (Wand bei Lokation 0 an der oberen Maschenhülle (skalar), wobei die Undurchlässigkeit von 0 bis 1 wächst.)
  - 11: Wand an der durch die Lokation gegebenen Maschenhülle (bei Lokation 0 an der rechten Maschenhülle (skalar) bzw. an der rechten und oberen Maschenhülle (vektoriell)), wobei die Durchlässigkeit von 0 bis 1 wächst.
  - 12: (Wand bei Lokation 0 an der oberen Maschenhülle (skalar), wobei die Durchlässigkeit von 0 bis 1 wächst.)
- ISYMB gibt die Darstellung des Materials bzw. der Wand durch eine Symbolik an:
- 1: Material: durch gleichmäßige Einfärbung mit der gewählten Farbe (bei verteilter Phase: Farbsättigung<sup>1</sup> proportional zum Volumenanteil);  
Wand: durch durchgezogene Gerade in der gewählten Farbe (Farbsättigung<sup>1</sup> oder Liniendicke<sup>2</sup> proportional zur Undurchlässigkeit);
  - 2: Material: durch Schraffur l.u./r.o. mit der gewählten Farbe (bei verteilter Phase: Schraffurdichte proportional zum Volumenanteil);  
Wand: durch unterbrochene Gerade in der gewählten Farbe (Lückenweite proportional zur Durchlässigkeit);
  - 3: Material: durch Schraffur l.o./r.u. mit der gewählten Farbe (bei verteilter Phase: Schraffurdichte proportional zum Volumenanteil);  
Wand: durch zentrierte durchgezogene Gerade in der gewählten Farbe (Weite proportional zur Undurchlässigkeit);
  - 4: Material: durch kreuzweise Schraffur mit der gewählten Farbe (bei verteilter Phase: Schraffurdichte proportional zum Volumenanteil);
  - 5: Material: durch ausgefüllte Tropfen mit der gewählten Farbe (bei verteilter Phase: Tropfenanzahl proportional zum Volumenanteil);
  - 6: Material: durch hohle Blasen mit der gewählten Farbe (bei verteilter Phase: Blasenanzahl proportional zum Volumenanteil).

<sup>1</sup> Proportionale Farbsättigung ist z.Z. nur möglich für die Farben rot und blau.

<sup>2</sup> Proportionale Liniendicke ist z.Z. nur für die Farbe schwarz implementiert.

ICLOR gibt die Farbe der Symbolik zur Darstellung des Materials bzw. der Wand an:

- 1: schwarz:
- 2: rot
- 3: grün
- 4: blau
- 5: türkis
- 6: gelb
- 7: violett
- 8: weiß

IGRADE gibt (als Exponent zur Basis 1.41...) einen Faktor zur Änderung der automatisch gesetzten Codierung der Volumenanteile bzw. Wand(un)durchlässigkeit durch Farbsättigung bzw. Schraffurdichte bzw. Tropfen/Blasengröße bzw. Strichstärke an.



### 3.2 Anmerkungen zu den Eingabeparametern

IKRUM > 0 wirkt für Größen der Gruppe 15 und Subgruppe 17 und bezieht sich immer auf das Koordinatensystem des jeweiligen Netzes oder Teilgebiets der auszugebenden Größe.

Bei IKRUM = 1 (Koordinatenumrechnung) und krummlinigen Netzen/Teilgebieten/Subgebieten werden die Koordinaten ggf. auf kartesische Koordinaten umgerechnet.

Bei IKRUM = 2 (Abwicklung) und einem (abwickelbaren)  $\vartheta$ ,  $z$ -Teilgebiet/Subgebiet oder Einzelmaschen daraus wird die Koordinate  $\vartheta$  auf die Koordinate  $u$  der Abwicklung umgerechnet ( $u = \rho_0 \vartheta$ , wo  $\rho_0$  die festgelegte Koordinate des Teilgebiets/Subgebiets ist); die Reihenfolge der Vektorkomponenten (falls vorhanden) ist dann  $\rho, u, z$ .

IKRUM < 0 (Spiegelung) wirkt nur für Größen der Gruppe 15 (aber nicht bei Teilnetzen und nicht bei defektiven Netzen) und bezieht sich immer auf das Koordinatensystem des jeweiligen Netzes oder Teilgebiets der auszugebenden Größe. Hier wird bei einem 2dimensionalen (ebenen, rechteckigen)  $r, z$ -Netz/Teilgebiet (das voll an die  $z$ -Achse anschließen muß) das Netz/Teilgebiet um das an der  $z$ -Achse gespiegelte Netz/Teilgebiet ergänzt; der  $r$ -Koordinatenwert kehrt sich in der gespiegelten Ebene um.

Wenn NT = 0 ist, werden alle Rumpf-Pakete ausgegeben. Wenn NT = 1 ist, werden alle der im durch T1(1), T2(1) spezifizierten Problemzeitrahmen liegenden Rumpf-Pakete ausgegeben.

Im Falle NT < 0 sind die Problemzeitintervalle durch

$$T2(m - 1) + n \times T1(m) \pm 0.1 \times T1(m)$$

gegeben, wo  $m = 1, \dots, |NT|$  ist;  $T2(0) = 0$  und  $n = 0, 1, \dots$ , solange wie  $T2(m - 1) + n \times T1(m) < T2(m)$  bleibt. Falls mehrere Rumpf-Pakete in einem Problemzeitintervall liegen, wird nur das erste davon ausgewählt.

Die Eingabeparameter KOR1 und KOR2 beziehen sich auch bei degenerierten Netzen auf die Netzdimension. Die Zählung der Maschen bezieht sich immer auf die Lokationen der Maschenmitte, d.h. sie beginnt mit 1. Wände auf den zugehörigen Maschenhüllen werden den Maschen automatisch richtig zugeordnet, unabhängig von der von den Lokationsindikatoren abhängigen evtl. unterschiedlichen Zählung. Für die in degenerierten Netzen konstant gehaltenen Koordinatenrichtungen müssen KOR1 und KOR2 = -1 sein oder die Werte der Maschenindizes der konstant gehaltenen Koordinaten haben.

Die Schichtgrenzen in den Maschen verlaufen entlang der horizontalen Bildachse; die Schichten werden vertikal, von unten nach oben, angeordnet. Die in den Maschen verbleibende Fläche über den Schichten steht für die verteilte Phase zur Verfügung. Die Darstellung der Materialien in den Maschen geschieht in der Reihenfolge: Material in der untersten Schicht, Material in der nächstuntersten Schicht, ..., verteilte Materialien in der verbleibenden Fläche so, wie sie in der Eingabe (Zeilen 5) aufeinanderfolgen. Anschließend werden ggf. Wände so, wie sie in der Eingabe aufeinanderfolgen, dargestellt. Die jeweils zuletzt dargestellten Symbole überdecken u.U. vorherige Darstellungen, was besonders für die verteilte Phase zu beachten ist.

Bei getrennter Phase eines Materials (IPHASE = 1, 2, ...) ist die Symbolik für die Darstellung des Materials in der Schicht unabhängig von dessen Volumenanteil.

Bei verteilter Phase eines Materials (IPHASE = 0) ist die Symbolik für die Darstellung des Materials in der freien Fläche der Masche proportional zu dessen Volumenanteil. Bei Tropfen oder Blasen als Symbolik entspricht das Produkt aus Tropfen/Blasenfläche und Tropfen/Blasenanzahl dem Volumenanteil in der Masche.

Wenn mit IGRADE die automatisch gewählte Größe der Tropfen oder Blasen verändert wird, wird gegenläufig dazu die Anzahl der Tropfen bzw. Blasen angepaßt.

Wenn als Wände zu interpretierende Größen in der VISART-Datei mit der Lokation 0 stehen, so werden sie von VISARVOL wie über der Lokation 01 (für skalare Größen) bzw. wie über den

Lokationen 01 und 02 (für vektorielle Größen) wiedergegeben, falls IPHASE =  $\pm 11$  spezifiziert wurde, dagegen wie über der Lokation 02 (für skalare Größen), falls IPHASE =  $\pm 12$  spezifiziert wurde.

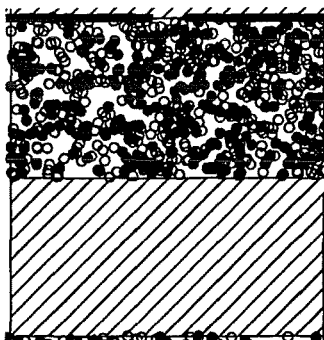
### 3.3 Eingabebeispiele

Im folgenden werden zwei Fälle der Steuereingabe anhand von Beispielen erläutert. Die Beispiele sind konsistent mit den im Teil 2 der VISART-Dokumentation [1b], Anhang A, aufgeführten Beispielen zum Aufbau von VISART-Dateien. ALPLK\_1, ALPLK\_2 und ALPLK\_3 sollen die Volumenanteile dreier Materialien beschreiben, XWALL und YWALL sollen die Durchlässigkeit der Wände auf den Maschenhüllen in  $x$ -Richtung (Lokation 11) bzw. in  $y$ -Richtung (Lokation 22) beschreiben. Die Abbildungen zeigen (vergrößert) eine Masche aus den zugehörigen Darstellungen. Weitere Eingabebeispiele findet man im Abschnitt 6.4.

**Netzgrößen** (Gruppen 5/15) werden durch die Angabe ihrer Identifikation in CN eindeutig bestimmt.

Beispiel für drei Materialien und Wände in beiden Richtungen über dem vollen Netz, zu einem bestimmten Problemzeitpunkt (hier Material 1 in getrennter Schicht, Material 2 und 3 als verteilte Phase):

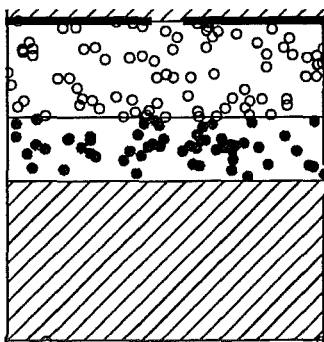
```
5 0
1 1 1 5
99.9 100.1
15 -1 -1 -1 -1 -1 -1
'ALPLK 1' '' 1 2 1 0
'ALPLK 2' '' 0 5 2 0
'ALPLK 3' '' 0 6 4 0
'XWALL' '' -11 2 1 2
'YWALL' '' -11 2 1 2
```



**Maschengrößen** (Subgruppen 7/17) werden durch die Angabe ihrer Identifikation in CN und der Referenz auf die zugehörige Referenzgruppe 6/16 in CG eindeutig bestimmt.

Beispiel für drei Materialien und Wände in beiden Richtungen über dem defektiven Netz, zum letzten vorhandenen Problemzeitpunkt (hier alle drei Materialien in getrennten Schichten):

```
5 0
1 1 99 5
17 -1 -1 -1 -1 -1 -1
'ALPLK 1' 'INDEX' 1 2 1 0
'ALPLK 2' 'INDEX' 2 5 2 0
'ALPLK 3' 'INDEX' 3 6 4 0
'XWALL' 'INDEXX' -11 2 1 2
'YWALL' 'INDEXY' -11 2 1 2
```



### 3.4 Eingabe über die Benutzeroberfläche

Wenn das Programm VISARVOL über die grafische Benutzeroberfläche [1a] aufgerufen wird, wird die Steuereingabe durch das Tcl/Tk-Skript der Benutzeroberfläche aus den Einstellungen der „Widgets“ im VISARVOL-Fenster aufgebaut und nach Betätigung der Tasten „Ausführen“ oder „*n* Eingaben“ auf die Datei VISART.IN im jeweiligen Arbeitsverzeichnis geschrieben. Diese Datei wird dem Programm beim Aufruf als Eingabedatei übergeben.

Die Widgets (das sind Tasten, Wähltasten, Schalttasten, Menütasten, Eingabekästchen, Anzeigefelder) sind meist einem bestimmten Eingabeparameter zugeordnet und setzen dessen Wert nach der Stellung der Taste(n) oder auf den im Kästchen oder Anzeigefeld eingegebenen Wert. Die Funktion der Widgets im VISARVOL-Fenster ist durch in die VISART-Benutzeroberfläche eingebaute Hilfetexte dokumentiert. In diesen Hilfen werden auch die Zuordnung der Widgets zu den Eingabeparameter und die den Tastenstellungen entsprechenden Werte angegeben.

Bild 2 zeigt das VISARVOL-Fenster, das für den Aufruf von VISARVOL in Frage kommt. Alle Widgets unterhalb der Anzeigefelder für die Dateien, mit Ausnahme der Schalttaste „ASCII“ (siehe 5.2), dienen zur Eingabe von Steuerparametern. Mit der Wähltastenleiste für das Visualisierungssystem wird der Parameter MODE gesetzt (hier nur für Tecplot möglich).

Mit der Wähltastenleiste „keine...“, „Umrechnung“, „Abwicklung“, „Spiegelung“ wird der Parameter IKRUM spezifiziert. In den Kästchen der nachfolgenden Zeile wird die „Anzahl“ NP der Rumpf-Pakete, die Anzahl NT der spezifizierten Problemzeit-„Intervalle“ und die entsprechenden „Intervallgrenzen“ T1, T2, schließlich in einem Kästchen die „Anzahl“ NN der spezifizierten Größen eingesetzt. (Der Parameter NF wird implizit über die Spezifizierung der VISART-Datei(en) im linken oberen Anzeigefeld gesetzt.) In den Kästchen der nächsten Zeile werden die Parameter NG, KOR1(1), KOR2(1), KOR1(2), KOR2(2), KOR1(3), KOR2(3) eingesetzt, die für alle Größen gültig sind. In den Kästchen des letzten Feldes werden die Parameter für die NN Größen eingesetzt, und zwar für jede Größe eine Zeile mit CN, CG, IPHASE, ISYMB, ICLOR, IGRADE.

Da im VISARVOL-Fenster nur Platz für die Spezifizierung von maximal 4 Größen ist, kann durch Doppelklicken auf das Kästchen für die Anzahl der Größen ein Unterfenster für die Spezifizierung von bis zu 20 Größen eröffnet werden.

Identifikationen (für CN, CG) werden in die Kästchen der Benutzeroberfläche ohne die bei der listengesteuerten Eingabe unter Fortran benötigten Apostrophe eingesetzt. In den Identifikationen stehende Leerzeichen werden deshalb als Unterstriche eingegeben. Sind Unterstriche Teil der Identifikation, so sind sie bei der Eingabe durch einen vorgesetzten Backslash zu „quoten“.

Für nicht ausgefüllte Kästchen werden den Parametern Default-Werte zugewiesen, falls dies möglich ist.

Die folgenden Eingabeparameter sind wegen des begrenzten Platzes in den Fenstern in der Größe eingeschränkt:

$$\begin{aligned} |NT| &\leq 3 \\ |NN| &\leq 20 \end{aligned}$$

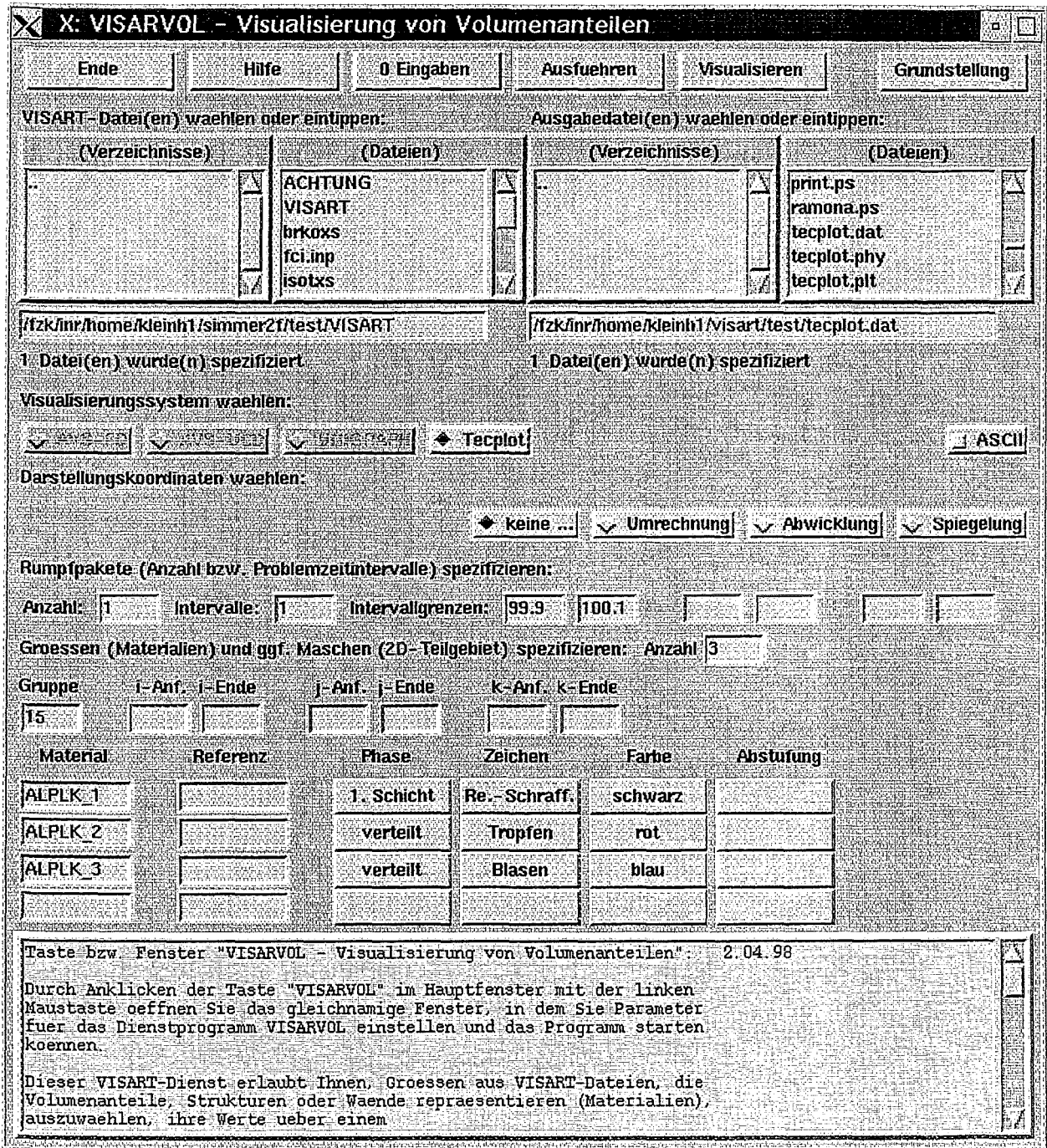


Bild 2: VISARVOL-Fenster

## 4. Ein- und Ausgabedateien des Programms

In diesem Kapitel werden die VISART-Eingabedatei(en) des Programms VISARVOL und seine Ausgabedateien für die Visualisierungssysteme beschrieben.

### 4.1 VISART-Dateistruktur und -inhalt

Im VISART-Standard ist nur die Form, nicht aber der Inhalt der Postprocessor-Dateien festgelegt. Allerdings werden von VISARVOL einige wenige Einschränkungen der Struktur der Dateien und einige Konventionen vorausgesetzt, die im folgenden aufgeführt werden.

Bezüglich Anzahl und Format der Dateien gilt:

- Die von VISARVOL erwarteten VISART-Dateien können aus einer oder mehreren Anwendungen eines Codes stammen, und jeweils aus einem Erstlauf und beliebig vielen Restart-Läufen einer Anwendung bestehen. Solche Fortsetzungsdateien werden hier als „Files“ einer VISART-„Datei“ bezeichnet. VISARVOL akzeptiert eine oder mehrere „Dateien“ im obigen Sinne, mit jeweils einem oder mehreren „Files“, als Eingabe (siehe 3.1).
- Die von VISARVOL erwarteten VISART-Dateien dürfen formatiert oder unformatiert sein und im letzteren Fall einfache oder doppelte Genauigkeit der REAL-Daten haben. Wenn sie formatiert, d.h. in ASCII geschrieben sind, sind sie ganz unabhängig von ihrer Herkunft lesbar. Wenn sie unformatiert, d.h. binär sind, sind sie im allgemeinen nur lesbar, wenn VISARVOL unter dem gleichen Betriebssystem und dem gleichen Processor-Typ läuft, wie diejenigen, mit denen die Dateien geschrieben wurden.
- Da VISARVOL in Fortran geschrieben ist, müssen unformatierte VISART-Dateien, die z.B. mit C-Programmen erstellt werden, auch die interne Darstellung der Fortran-Records nachbilden (mit einem 4-Byte-Wort am Anfang und Ende jedes Records, in dem die Anzahl der Bytes des Records steht), wenn sie von VISARVOL gelesen werden sollen.

Bezüglich der Struktur gilt:

- VISARVOL erwartet, daß alle Rumpf-Pakete einer VISART-Datei gleichartig aufgebaut sind, d.h. aus der gleichen Anzahl und Anordnung gleicher Gruppen bestehen. Dies gilt auch über alle Fortsetzungs-Files (aus Restart-Läufen eines Codes) einer VISART-Datei hinweg. Bei mehreren VISART-Dateien (im obigen Sinne) als Eingabe brauchen die Dateien untereinander nicht gleichartig aufgebaut sein.
- VISARVOL kann Größen der Gruppen/Subgruppen 5/15 und 7/17 aus VISART-Dateien verarbeiten. Größen der Gruppen/Subgruppen 51/151 und 71/171 werden dagegen bisher noch nicht akzeptiert.
- In einem Kopf- oder Rumpf-Paket können mehrere Referenzgruppen 6/16, mit jeweils zugehörigen Subgruppen 7/17, auftreten.
- Bei der Angabe von Größen der Gruppen 5/15 über Teilnetzen [1a, 1b] müssen die Identifikationen einer Größe in allen Gruppen, die jeweils ein Teilnetz überdecken, die gleichen sein. Die Reihenfolge der Teilnetze, gegeben durch die Reihenfolge der Gruppen, muß für alle Größen die gleiche sein. Die Teilnetze können, aber müssen sich nicht, überlappen.

Bezüglich der Konventionen gilt:

- VISARVOL erwartet die Geometrie-Gruppe 4 unter der Identifikation GEOMETRIE.
- Bei „schwach defektiven“ Netzen müssen die Gruppen 5/15 mit der Angabe der nicht belegten Maschen („Löcher“) des Netzes Identifikationen mit dem Namensbestandteil `..DEFCT..` tragen. In diesen Gruppen (Lokation 0, Typ Integer) stehen für die belegten Maschen Nullen, für die Löcher Einsen.

- Bei defektiven Netzen müssen die Referenzgruppen 6/16 mit den Maschenindexpaaren/tripel der belegten Maschen oder Maschenhüllen Identifikationen mit dem Namensbestandteil ..INDEX.. tragen.

Bezüglich der Verteilung auf Kopf-Paket und Rumpf-Pakete der VISART-Datei gilt:

- Für VISARVOL können die Gruppen mit den Angaben zur Netzgeometrie (Maschenspezifizierung, -belegung) in der VISART-Datei an verschiedener Stelle stehen: 5 bzw. 6/7 im Kopf-Paket, 15 bzw. 16/17 im ersten Rumpf-Paket oder 15 bzw. 16/17 in jedem Rumpf-Paket. Im einzelnen gilt:
- Die Referenzgruppen 6/16 mit der Angabe der belegten Maschen oder Maschenhüllen bei defektiven Netzen können sowohl im Kopf-Paket, als auch in den Rumpf-Paketen stehen. Stehen sie als Gruppen 6 im Kopf-Paket, so dürfen die (syntaktisch erforderlichen) entsprechenden Gruppen 16 in den Rumpf-Paketen leer sein (siehe Tab. B.3 in [1b]).
- Die Gruppen mit der Angabe der nicht belegten Maschen bei „schwach defektiven“ Netzen (Gruppen 5/15) können sowohl im Kopf-Paket als auch in den Rumpf-Paketen stehen (siehe Tab. B.2 in [1b]).

#### 4.2 Ausgabedatei

Die Ausgabedatei von VISARVOL dient als Eingabedatei für das Visualisierungssystem Tecplot. Sie ist formatiert, d.h. in ASCII, geschrieben oder auch unformatiert, d.h. in binärer Form (siehe 5.1 und 5.2).

Die Ausgabedatei enthält die Koordinaten der Maschen und die Darstellung der aus der VISART-Datei ausgewählten Volumenanteile und Wände über dem spezifizierten Teilgebiet und den spezifizierten Problemzeiten.

In der Beschreibung des Programms und in der Ausgabedatei wird mit „Komponente“ schlechthin eine skalare Größe und jede Vektorkomponente einer vektoriellen Größe bezeichnet. Den Komponenten werden die auszugebenden Größen in der spezifizierten Reihenfolge zugeordnet. Diese Reihenfolge der Größen gilt auch auf den Ausgabedateien für das Visualisierungssystem. Bei vektoriellen Größen erscheinen jeweils die Vektorkomponenten der Größen nacheinander.

## 5. Technik des Programms

In diesem Kapitel werden Aufruf, Aufbau und Charakteristiken des Programms VISARVOL beschrieben.

### 5.1 Programmoptionen

Das Programm VISARVOL ist in Fortran geschrieben und damit in jedem Betriebssystem implementierbar. Es läuft zur Zeit unter UNIX auf IBM-RS/6000-Workstations.

Das Programm verwendet im wesentlichen Fortran-77; aus Fortran-90 wurden für eine dynamische Speicherverwaltung einige Element im Zusammenhang mit ALLOCATE-Befehlen und das Module-Konzept übernommen.

Die Versionsführung geschieht mit dem Werkzeug HISTORIAN [2]. Die aktuelle HISTORIAN-Source von VISARVOL (z.Z. Release D.4) steht unter DCE/DFS in der Datei

```
/fzk/inr/@sys/VISART/sources90/visarvol90
```

Aus ihr wird in einem HISTORIAN-Lauf zunächst eine compilierbare Fortran-Source, unter Berücksichtigung der spezifizierten Optionen, erstellt. Die folgenden Optionen können kombiniert werden:

- UNIX: Falls diese Option „on“ ist, wird die Fortran-Source für UNIX erzeugt (andernfalls für MVS).
- ENGLISH: Falls diese Option „off“ ist, wird eine Fortran Source mit deutschsprachigen Nachrichten und Fehlermeldungen im Ausführungsprotokoll erzeugt; andernfalls mit englischsprachigen Texten.
- DELPP: Zum Lesen von unformatierten VISART-Dateien mit einfacher Wortlänge der REAL-Variablen und von formatierten VISART-Dateien [1b] muß diese Option „off“ sein (das Programm erkennt aus der ersten Beschreibungsgruppe der VISART-Datei, ob es sich um eine unformatierte oder um eine formatierte Datei handelt, und stellt sich demgemäß darauf ein). Zum Lesen und Verarbeiten von unformatierten VISART-Dateien mit doppelter Wortlänge der REAL-Variablen muß diese Option „on“ sein.
- BINARY: Diese Option betrifft das Format der Ausgabedatei für das Visualisierungssystem Tecplot. Wenn die Option „off“ ist, wird die Datei im ASCII-Format erstellt und bei der Verarbeitung in Tecplot dort in ein binäres Format umgewandelt. Andernfalls wird die Datei bereits von VISARVOL als binäre Datei erstellt, was das Einlesen in Tecplot beträchtlich beschleunigt.

Zum automatischen Erzeugen der Fortran-Source mit unterschiedlichen Kombinationen von Optionen stehen unter DCE/DFS Shell-Scripts unter den Namen

```
/fzk/inr/@sys/VISART/sources90/visarvol??s
```

zur Verfügung, wo ?? die Optionenkombination bezeichnet (siehe 5.2). (Die Shell-Scripts setzen auch Datum und Uhrzeit der Fortran-Source-Erstellung in die dafür vorgesehenen DATA-Statements in VISARVOL ein, deren Inhalt bei der Ausführung des Programms ins Protokoll gedruckt wird.)

## 5.2 Programmaufruf und -argumente

Ausführbare Binärprogramme von VISARVOL stehen unter DCE/DFS im Verzeichnis

```
/fzk/inr/@sys/VISART/bin
```

und zwar als Datei

- visarvol für die HISTORIAN-Option UNIX;
- visarvol-e für die HISTORIAN-Optionen UNIX, ENGLISH;
- visarvol-b für die HISTORIAN-Optionen UNIX, BINARY;
- visarvol-v für die HISTORIAN-Optionen UNIX, ENGLISH, BINARY.

Beim Aufruf von VISARVOL aus der VISART-Benutzeroberfläche werden bei Sprachauswahl „deutsch“ (über die Menütaste „Sprache“ im Hauptfenster der Benutzeroberfläche) die Binärprogramme ohne die Option ENGLISH ausgeführt, bei Sprachauswahl „english“ die Binärprogramme mit der Option ENGLISH. Bei gedrückter Taste „ASCII“ im VISARVOL-Fenster werden die Binärprogramme ohne die Option BINARY ausgeführt, bei nicht gedrückter Taste die Binärprogramme mit der Option BINARY.

Das Programm VISARVOL erwartet die Steuereingabe auf seiner Standardeingabe (Unit 5) und schreibt das Ausführungsprotokoll auf seine Standardausgabe (Unit 6). Das Ausführungsprotokoll besteht aus der Wiedergabe der Steuereingabe, dem Ausdruck der Kesssätze der ausgewählten Gruppen, und eventuellen Fehlermeldungen.

Beim Aufruf von VISARVOL über die grafische Benutzeroberfläche wird die Steuereingabe der von den Tcl/Tk-Skripts der Benutzeroberfläche aufgebauten Datei VISART\_IN entnommen (siehe 3.4). Die Protokollausgabe wird auf die Datei VISART\_OUT im jeweiligen Verzeichnis gelegt und im Textfeld des VISARVOL-Fensters angezeigt. Eventuelle Fehlerausgabe wird auf die Datei VISART\_ERR im jeweiligen Verzeichnis gelegt und in eigenen Dialogfenstern angezeigt.

Neben der Steuereingabe erwartet VISARVOL eine oder mehrere VISART-Datei(en) als Eingabe, jeweils aus einem oder mehreren Files bestehend (siehe 4.1), und erstellt eine Datei für das Visualisierungssystem (siehe 4.2). Die Namen der laut Steuereingabe benötigten Ein- und Ausgabedateien werden als Argumente beim Programmaufruf angegeben:

```
visarvol [ 10/1 [ 10/2 ... ] [ 20 ] ]
```

wo die Unit-Nummern für die Datei-/File-Namen laut nachfolgender Tabelle 5.2 stehen. Beim Aufruf von VISARVOL über die grafische Benutzeroberfläche werden die im VISARVOL-Fenster eingestellten VISART-Datei(en) bzw. -Fortsetzungsfiles und die eingestellte Ausgabedatei als Argumente übergeben.



Falls keine Datei-/File-Namen als Argumente angegeben werden, verwendet VISARVOL die in der folgenden **Tabelle 5.2** stehenden:

MODE	Unit	Name	Status	Format	Bedeutung
	10/1	VISART1	old	form./unform.	1. Datei, 1. File
	10/2	VISART2	old	form./unform.	1. Datei, 2. File
	...	...	...	...	...
	10/n	VISARTn	old	form./unform.	1. Datei, n. File
	11/1	VISARTn+1	old	form./unform.	2. Datei, 1. File
	11/2	VISARTn+2	old	form./unform.	2. Datei, 2. File
	...	...	...	...	...
	11/m	VISARTn+m	old	form./unform.	2. Datei, m. File
	...	...	...	...	...
5	20	DAT	unknown	form./unform.	Tecplot Data File

### 5.3 Programmaufbau

Das Hauptprogramm von VISARVOL steuert den Programmablauf in folgenden Phasen:

(Initialisierung:)

- Initialisieren und Einlesen der Zeile 1 der Steuereingabe (Subroutine EING1).

- (1) Einlesen der Zeilen 2 bis 5 der Steuereingabe für die jeweilige VISART-Datei (Subroutine EING2); falls Eingabeende: weiter bei (9).
- (2) Öffnen des anstehenden Files der VISART-Datei; Lesen der Beschreibungsgruppen; Auswerten der Geometrie-Gruppe (mit Subroutine CGEOM); Allokieren der dimensionsabhängigen Felder.

(Für das Kopf-Paket des VISART-Files:)

- Lesen des Kennsatzes der nächsten Gruppe vom VISART-File.

Falls Gruppe 5: Aufruf von Subroutine MESH. (Dort Aufruf von Subroutine CDEFCT, falls die Gruppe Fehlstellen definiert.) Falls die Gruppe übernommen werden soll: Einlesen der Daten, ggf. Ausschneiden des Teilgebiets, Umrechnen und Abspeichern als „Komponente(n)“ auf eine Hilfsdatei.

Falls Gruppe 6: Aufruf von Subroutine INDEX. Dort, falls die Gruppe Referenzgruppe für Subgruppe(n) 7 der Eingabe ist: Einlesen der Daten und Abspeichern als Indexvektor auf eine Hilfsdatei.

Falls Subgruppe 7: Aufruf von Subroutine CELL. Dort, falls die Subgruppe übernommen werden soll: Einlesen der Daten, ggf. Ausschneiden des Teilgebiets, Umrechnen und Abspeichern als „Komponente(n)“ auf eine Hilfsdatei.

(Für jedes Rumpf-Paket des VISART-Files:)

- Lesen der Zyklus-Gruppe des nächsten Rumpf-Pakets vom VISART-File.

Falls der Problemzyklus/zeitpunkt nicht übernommen werden soll: Überlesen des gesamten Rumpf-Pakets.

Falls der Problemzyklus/zeitpunkt übernommen werden soll:

- Lesen des Kennsatzes der nächsten Gruppe vom VISART-File.

Falls Gruppe 15: Aufruf von Subroutine MESH. (Dort Aufruf von Subroutine CDEFCT, falls die Gruppe Fehlstellen definiert.) Falls die Gruppe übernommen werden soll: Einlesen der Daten, ggf. Ausschneiden des Teilgebiets, Umrechnen und Abspeichern als „Komponente(n)“ auf eine Hilfsdatei.

Falls Gruppe 16: Aufruf von Subroutine INDEX. Dort, falls die Gruppe Referenzgruppe für Subgruppe(n) 17 der Eingabe ist: Einlesen der Daten und Abspeichern als Indexvektor auf eine Hilfsdatei.

Falls Subgruppe 17: Aufruf von Subroutine CELL. Dort, falls die Subgruppe übernommen werden soll: Einlesen der Daten, ggf. Ausschneiden des Teilgebiets, Umrechnen und Abspeichern als „Komponente(n)“ auf eine Hilfsdatei. Letzteres geschieht durch Verzweigung nach MESH (und anschließende Rückkehr nach CELL) für das Visualisierungssystem Tecplot, wobei die Daten für die belegten Maschen um solche für die leeren Maschen ergänzt werden.

(Für das Dateiende des VISART-Files:)

- Falls Dateiende innerhalb eines Rumpf-Pakets: Rücksetzen auf den Stand vor dem Lesen der Zyklus-Gruppe dieses Pakets und weiter nach (9).

Falls laut Steuereingabe Fortsetzungsfiles vorhanden sind: zurück nach (2).

Andernfalls: Schließen der jeweiligen VISART-Datei und zurück nach (1).

(Nach dem Lesen aller VISART-Dateien:)

(9) Öffnen der Ausgabedatei und Aufruf der Routine für das Visualisierungssystem:

Falls Eingabeparameter `MODE = 5`: Subroutine `MODE5` (für Tecplot).

- In dieser Routine werden die Koordinaten des Teilgebiets für die Ausgabe aufbereitet und die Werte der Komponenten darüber von den Hilfsdateien eingelesen und auf die Ausgabedatei geschrieben.

(Programmende.)

#### 5.4 Einige Programmdetails

Für das Abspeichern der Komponenten auf Hilfsdateien gilt:

Für jede Komponente (siehe 4.2) werden die Werte über den Koordinaten  $i, j, k$  (soweit vorhanden) fortlaufend auf temporären Hilfsdateien (ab Dateinummer 51) zwischengespeichert, wobei die erste Koordinate als erste variiert, dann die zweite Koordinate, usw..

Für das Abspeichern der Indexvektoren auf Hilfsdateien gilt:

Für jeden Indexvektor werden dessen Werte auf temporären Hilfsdateien (von Dateinummer `MAX-DAT+50` abwärts gezählt) zwischengespeichert.

## 5.5 Einschränkungen usw.

Die Tabellen für die Eingabeparameter und die Geometrie-abhängigen Felder werden in VISARVOL dynamisch dimensioniert, soweit dies am Programmanfang möglich ist.

Die anderen Felder sind zur Zeit wie folgt dimensioniert:

- Maximal MAXDAT = 949 temporäre Hilfsdateien für Komponenten und Indexvektoren.
- Maximal MAXNT = 20 Problemzeitintervalle in der Eingabe.
- Maximal MAXT = 5000 auszugebende Problemzyklusnummern/zeitpunkte.
- Maximal MAXLA1 = 6 unterschiedliche Lokationen der auszugebenden Größen (Lokationen von Größen aus dem Kopf-Paket gelten als verschieden von den gleichen Lokationen der Größen aus dem Rumpf-Paket).
- Maximal MAXMU = 3 Teilnetze in den VISART-Dateien.
- Maximal MAXNU = 2 (zu leeren Gruppen in den Rumpf-Paketen) komplementäre Gruppen im Kopf-Paket.
- Maximal MAXND = 2 VISART-Dateien (die Anzahl der jeweiligen Fortsetzungsfiles ist jedoch unbegrenzt).
- Maximal MAXCLB = 24 Zeichen in den Labels (einschließlich später entfernter Leerzeichen).

Bei der derzeitigen Implementierung von VISARVOL gilt für die Indikatoren (siehe [1b], Tabellen 4.3.4, ..., 4.3.7) der VISART-Dateien:

- Es können nur reguläre Maschennetze (IZGEO = 1) behandelt werden.
- Es können nur kartesische und zylindrische Koordinatensysteme IZSYS = 200, 210, 220, 300, 302 behandelt werden.
- Es können nur die „üblichen“ Reihenfolgen IxORD = 1 bzw. = 12 bzw. = 123 behandelt werden.
- Es können nur die Lokationen im Netzzinnern 0 (für Volumenanteile von Materialien) bzw. 0, 11, 22, 44, 01, 02, 04, 10, 20, 40 (für Wand(un)durchlässigkeiten) behandelt werden.
- Zur Zeit kann nur ein Problemzeitpunkt pro Aufruf verarbeitet werden.

## 5.6 Aufbau der Benutzeroberfläche

Die grafische Benutzeroberfläche für die VISART-Dienste unter UNIX wurde bereits im Teil 1 der VISART-Dokumentation [1a] vorgestellt. Sie ist in der Skriptsprache Tcl/Tk implementiert und setzt die Installierung des entsprechenden Interpreters voraus, der u.a. die Tk-Funktionen mit der Xlib-Library realisiert.

Die Tcl/Tk-Skripte der VISART-Benutzeroberfläche stehen unter DCE/DFS im Verzeichnis

```
/fzk/inr/@sys/VISART/tcltk
```

Der Aufruf des Skripts `visart` öffnet das Hauptfenster der Benutzeroberfläche. Von dort gelangt man durch Anklicken der Tasten „VISARVOL“ in das Skript `visarvol.tcl`, das das entsprechende Fenster (siehe 3.4) öffnet. Beim Aufruf des Visualisierungssystems aus dem VISARVOL-Fenster übergibt (ggf. nach Bestätigung in einem Dialogfenster) das Skript `visualize.tcl` die Ausgabedatei von VISARVOL zusammen mit einem Tecplot-Macro an das Visualisierungssystem.

## 6. Visualisierung mit Tecplot-7

In diesem Kapitel werden Besonderheiten der Ein- und Ausgabe für das Visualisierungssystem Tecplot-7 beschrieben.

### 6.1 Besonderheiten

MODE = 5 wählt die Ausgabe für Tecplot, Version 7 [3a].

Falls Teilnetze in der VISART-Datei definiert sind, bereitet VISARVOL deren Darstellung als „Zones“ in Tecplot vor.

Für schwach defektive Netze, definiert durch Maschenbelegungsgruppen (siehe 4.1) in der VISART-Datei, konstruiert VISARVOL aus der Maschenbelegung eine „Blanking Variable“ für Tecplot zur Ausblendung der „Löcher“ (siehe 6.2). Stark defektive Netze, definiert durch Gruppen 6/16 in der VISART-Datei, expandiert VISARVOL für Tecplot auf volle Netze mit Löchern und verfährt dann wie für schwach defektive Netze.

Die zugrunde liegenden Netze/Teilgebiete müssen kartesisch sein, außer im Falle, daß nur Wände dargestellt werden sollen, wo auch krummlinig orthogonale Netze/Teilgebiete zugelassen sind. Bei krummlinigen Koordinatensystemen müssen die Koordinaten auf kartesische Koordinaten umgerechnet werden (IKRUM = 1 oder IKRUM = 2).

Es können mehrere Größen, auch skalar und vektoriell gemischt, über dem gleichen Netz/Teilgebiet, nicht unbedingt mit gleichen Lokationen und daher nicht unbedingt mit gleicher Anzahl von Koordinatenwerten, ausgewählt werden. Es sind Größen des Typs REAL und INTEGER, auch gemischt, erlaubt (letztere werden auf erstere umdefiniert).

### 6.2 Aufbau der Tecplot-Datei

Tecplot erwartet die Eingabedaten in einem festen Format, das aber sehr flexibel ist. Ob die Eingabedatei als „ASCII-Data-File“ (Default-Suffix .dat) oder als „Binary-Data-File“ (Default-Suffix .plt) erstellt wird, wird durch die Auswahl des auszuführenden VISARVOL-Binärprogramms (siehe 5.2) festgelegt. ASCII-Dateien werden innerhalb von Tecplot nach dem Einlesen in binäre Form übersetzt, was zeitaufwendig sein kann, und sie beanspruchen mehr Platz als Binärdateien. Andererseits wird die Erstellung von Binary-Data-Files von den bisherigen Tecplot-Releases nur halbherzig unterstützt und empfohlen. Hier wird der Aufbau der ASCII-Data-Files beschrieben; auf die Binary-Data-Files werden die gleichen Daten geschrieben, wenn auch in anderer Anordnung und über die von Tecplot bereitgestellten Funktionen.

Ein von VISARVOL erstellter ASCII-Data-File ist aus File Header, Text Record und Ordered Zone Records für das Netz, sowie aus einer variablen Anzahl von Geometry Records mit den Symbolen für die Materialien und Wände, in der unten beschriebenen Anordnung, aufgebaut [3a]. Die Records werden mit Hilfe von Schlüsselworten und Schlüsselparametern gegliedert (die Aufteilung auf Zeilen ist beliebig). Letztere werden in den folgenden Beschreibungen in Schreibmaschinenschrift angegeben, während Variable kursiv geschrieben werden.

Es können beliebig viele Komponenten vorkommen. Da sie als Geometry Records geschrieben werden, lassen sie sich unter Tecplot nicht mehr als Komponenten identifizieren. Deshalb können mit dieser Anordnung keine Bildserien zur Verfilmung erstellt werden.

1. TITLE = "Name des Problems"
2. VARIABLES = "vname<sub>1</sub>" "vname<sub>2</sub>"
3. TEXT T = "ttime" X = xorigin Y = yorigin C = color H = height

Für jedes Teilnetz  $\mu = 1, \dots$  die folgenden Records 4, ..., 7.b:

4. ZONE T = " $\mu$ " I =  $\bar{i}_1$  J =  $\bar{i}_2$  F = BLOCK
5. 1. Koordinate (über alle  $\bar{i}_1 \times \bar{i}_2$  Lokationen des  $\mu$ -ten Teilnetzes)  
2. Koordinate (über alle  $\bar{i}_1 \times \bar{i}_2$  Lokationen des  $\mu$ -ten Teilnetzes)

Für jede als Schicht dargestellte Komponente  $\nu$  und für jede Masche die folgenden Records 6.a, 6.b, 6.c:

falls Einfärbung als Symbolik spezifiziert:

- 6.a GEOMETRY T = RECTANGLE M = GRID C = color <sub>$\nu$</sub>  FC = color <sub>$\nu$</sub>  X = ... Y = ...  
Koordinaten und Weiten der Schicht  $\nu$

falls Schraffur als Symbolik spezifiziert:

- 6.a GEOMETRY T = RECTANGLE M = GRID C = color <sub>$\nu$</sub>  X = ... Y = ...  
Koordinaten und Weiten der Schicht  $\nu$
- 6.b GEOMETRY T = LINE M = GRID C = color <sub>$\nu$</sub>   
Anzahl, Definition und Koordinaten der Geradensegmente der Schraffur

falls Blasen oder Tropfen als Symbolik spezifiziert:

- 6.a GEOMETRY T = RECTANGLE M = GRID C = color <sub>$\nu$</sub>  X = ... Y = ...  
Koordinaten und Weiten der Schicht  $\nu$
- für jede Blase oder jeden Tropfen:
- 6.c GEOMETRY T = CIRCLE M = GRID C = color <sub>$\nu$</sub>  [ FC = color <sub>$\nu$</sub>  ]  
Ursprung (Zufallsgröße) und Radius von Blasen oder Tropfen

Für jede in der verbleibenden Schicht verteilt dargestellte Komponente  $\nu$  und für jede Masche die folgenden Records 6.a, 6.b, 6.c:

falls Einfärbung als Symbolik spezifiziert:

- 6.a GEOMETRY T = RECTANGLE M = GRID C = colorgrade <sub>$\nu$</sub>  X = ... Y = ...  
Koordinaten und Weiten der Schicht

falls Schraffur als Symbolik spezifiziert:

- 6.a GEOMETRY T = RECTANGLE M = GRID C = color <sub>$\nu$</sub>  X = ... Y = ...  
Koordinaten und Weiten der Schicht
- 6.b GEOMETRY T = LINE M = GRID C = color <sub>$\nu$</sub>   
Anzahl, Definition und Koordinaten der Geradensegmente der Schraffur  
(Schraffurabstand umgekehrt proportional dem Wert der Komponente  $\nu$ )

falls Blasen oder Tropfen als Symbolik spezifiziert:

- 6.a GEOMETRY T = RECTANGLE M = GRID C = color <sub>$\nu$</sub>  X = ... Y = ...  
Koordinaten und Weiten der Schicht
- für jede Blase oder jeden Tropfen:
- 6.c GEOMETRY T = CIRCLE M = GRID C = color <sub>$\nu$</sub>  [ FC = color <sub>$\nu$</sub>  ]  
Ursprung (Zufallsgröße) und Radius von Blasen oder Tropfen  
(Radius proportional dem Wert der Komponente  $\nu$ )

Falls keine Wände mit durchgezogenen Linien als Symbolik vorliegen, für jede Masche der folgende Record 6:

6. GEOMETRY T = RECTANGLE M = GRID C = BLACK X = ... Y = ...  
Koordinaten und Weiten der Masche

Für jede als Wand dargestellte Komponente  $\nu$  und für jede Masche die folgenden Records 7.a, 7.b:

falls durchgezogene Gerade als Symbolik spezifiziert:

7.a GEOMETRY T = LINE M = GRID { C = *colorgrade $\nu$*  |  
C = *color $\nu$* , LT = *linethickness $\nu$*  }  
Definition und Koordinaten der Geraden

falls unterbrochene Gerade als Symbolik spezifiziert:

7.b GEOMETRY T = LINE M = GRID C = *color $\nu$*   
Anzahl, Definition und Koordinaten der Geradensegmente der Wand  
(Lückenweite proportional zur Wanddurchlässigkeit)

$vname_{1...2}$  sind die Labels zur Achsenbeschriftung (z.B. x, y, z, r, ..., t);

$ttime$  ist der Label zur Bezeichnung der Problemzeit;

$\bar{i}_1, \bar{i}_2$  sind die Anzahl der Maschen des  $\mu$ -ten Teilnetzes in den Koordinatenrichtungen.

*color $\nu$*  ist die mit dem Eingabeparameter ICLOR spezifizierte Farbe der Symbolik für das Material bzw. die Wand (als CHARACTER-Konstante BLACK, RED, GREEN, ...).

*colorgrade $\nu$*  ist die mit dem Eingabeparameter ICLOR spezifizierte Farbe der Schichteinfärbung bzw. der Wandgeraden in proportional zum Volumenanteil des Materials bzw. zur Wanddurchlässigkeit abgestufter Farbsättigung (u.a. über die Custom-Farben, als CHARACTER-Konstante CUST1, ..., CUST8, die im Tecplot-Configuration-File entsprechend definiert sein müssen<sup>1</sup>);

*linethickness $\nu$*  ist die proportional zur Wanddurchlässigkeit abgestufte Dicke der Wandgeraden.

---

<sup>1</sup> Wegen der lediglich acht zur Verfügung stehenden Custom-Farben ist die Farbsättigungsskala nur für zwei Farben (rot und blau) implementiert. Für jede dieser Farben stehen damit vier Werte zwischen der voll gesättigten Farbe und weiß zur Verfügung:

BLUE, CUST1, CUST2, CUST3, CUST4, WHITE bzw.

RED, CUST5, CUST6, CUST7, CUST8, WHITE

### 6.3 Tecplot-Aufruf

Die von VISARVOL für Tecplot im obigen Format erstellte Datei kann von Tecplot eingelesen und interpretiert werden. In Ergänzung zu diesem Dateiformat wird ein dazu kompatibler Tecplot-Macro-File zur Verfügung gestellt, der beim Aufruf des Visualisierungssystems als Parameter übergeben wird und unter Tecplot die Datei einliest und den Aufbau einer dem Fall angemessenen Standarddarstellung steuert. Unter DCE/DFS steht er im Verzeichnis

```
/fzk/inr/@sys/VISART/tecplot-macros
```

unter den Dateinamen

```
tecplot4.mcr
```

Tecplot wird hier mit dem Kommando

```
tecplot -p Macro-File
```

aufgerufen. Der Macro-File bewirkt das automatische Einlesen der von VISARVOL erzeugten Datei unter dem Namen `VINPUT.5.dat` und steuert anschließend die Darstellung der Größe(n). Anschließend kann der Benutzer die Steuerung von Tecplot zur Manipulation der eingelesenen Daten und Darstellungsparameter selbst übernehmen.

Das Visualisierungssystem Tecplot kann auch über die grafische Benutzeroberfläche [1a] aufgerufen werden. Im Anschluß an das „Ausführen“ des Programms VISARVOL über das VISARVOL-Fenster (Bild 2) der grafischen Benutzeroberfläche wird ein Dialogfenster zum Aufruf von Tecplot angeboten. Alternativ kann Tecplot auch durch Betätigen der Taste „Visualisieren“ im VISARVOL-Fenster eröffnet werden, wenn eine Datei `VINPUT.5.dat` bereits vorhanden ist. Nach Beendigung der Tecplot-Sitzung wird in das VISARVOL-Fenster der Benutzeroberfläche zurückgekehrt.



## 6.4 Beispiele

Bild 3.a und 3.b zeigen Beispiele für die Darstellung von Volumenfraktionen und Wanddurchlässigkeiten aus VISART-Postprocessor-Dateien mit Tecplot-7 über VISARVOL. Die zugehörigen Steuereingaben für das Programm VISARVOL stehen in Liste 6.a und 6.b.

Bild 3.a stammt aus dem MATTINA-Code. Das gezeigte PWR-Problem beruht auf einem quasi-zweidimensionalen, regulären, vollen  $20 \times 1 \times 24$  -Netz in Zylindergeometrie. Dargestellt werden die Volumenanteile von drei Materialkomponenten (Gas, Wasser, Schmelze), sowie die Permeabilitäten der Wände (beschrieben mit Lokation 0 für die rechten bzw. oberen Maschenhüllen), und zwar an der  $z$ -Achse gespiegelt.

Bild 3.b stammt aus dem GASFLOW-Code. Das gezeigte Problem beruht auf einem dreidimensionalen, regulären  $10 \times 23 \times 39$  -Netz mit Löchern in Zylindergeometrie. Dargestellt werden die Wandstrukturen auf den Maschenhüllen entlang eines axialen Querschnittes (Index  $k=10$ ). Sie sind hier als Undurchlässigkeiten durch die Gruppen IWALL, JWALL, Gruppenkennzahl 5, für die Lokationen 11 bzw. 22 beschrieben.

### Liste 6.a: VISARVOL-Steuereingabe zu Bild 3.a

```
5 -1
1 1 1 5
0.15 0.16
15 -1 -1 -1 -1 -1 -1
'AL____1' '' 3 5 3 1
'AL____2' '' 1 1 4 0
'AL____3' '' 2 1 2 0
'GARE' '' -11 2 1 1
'GAZT' '' -12 2 1 1
```

### Liste 6.b: VISARVOL-Steuereingabe zu Bild 3.b

```
5 1
1 1 0 2
5 -1 -1 -1 -1 10 -1
'IWALL' '' 11 1 1 2
'JWALL' '' 11 1 1 2
```

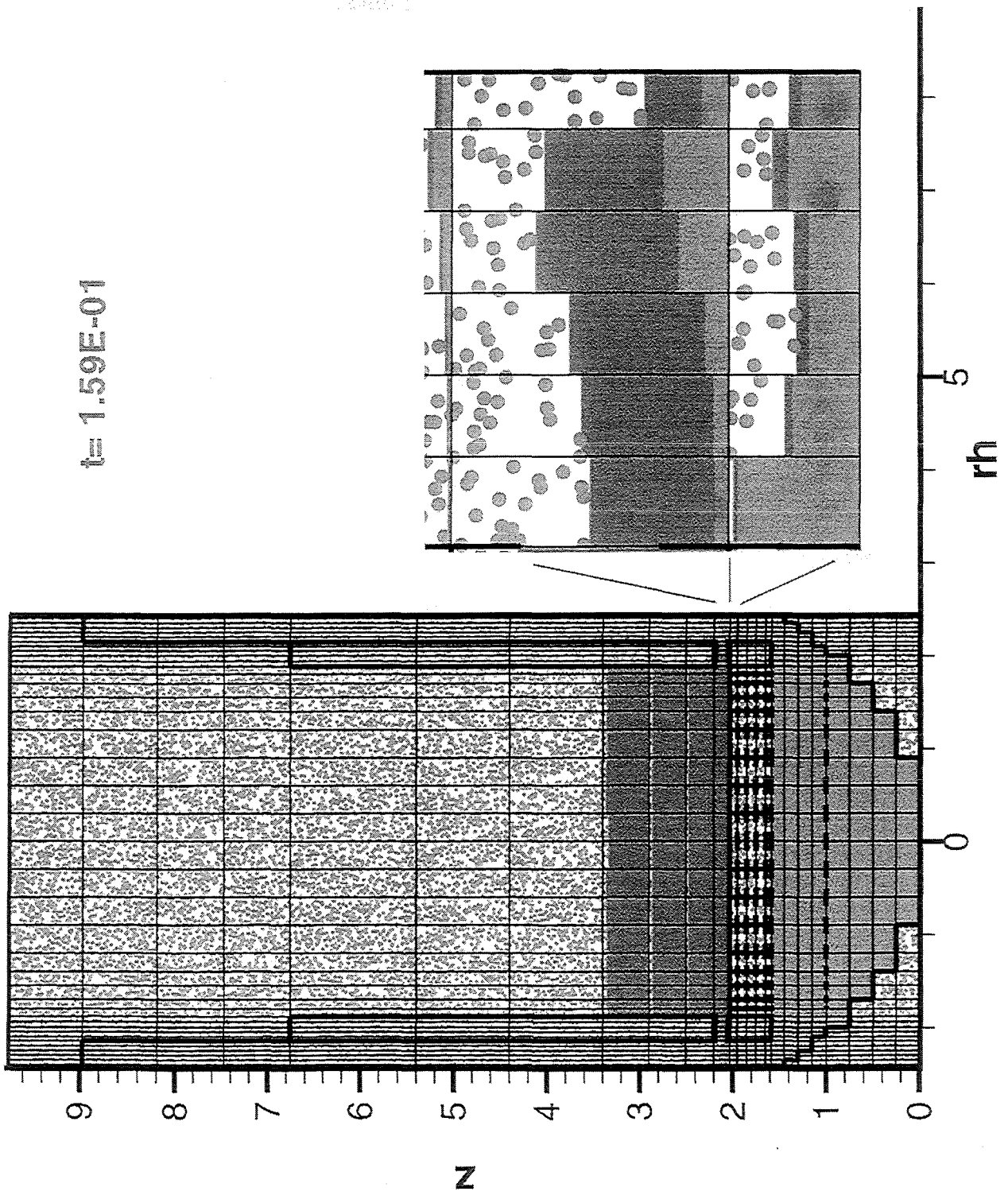


Bild 3.a: PWR-Problem aus dem MATTINA-Code

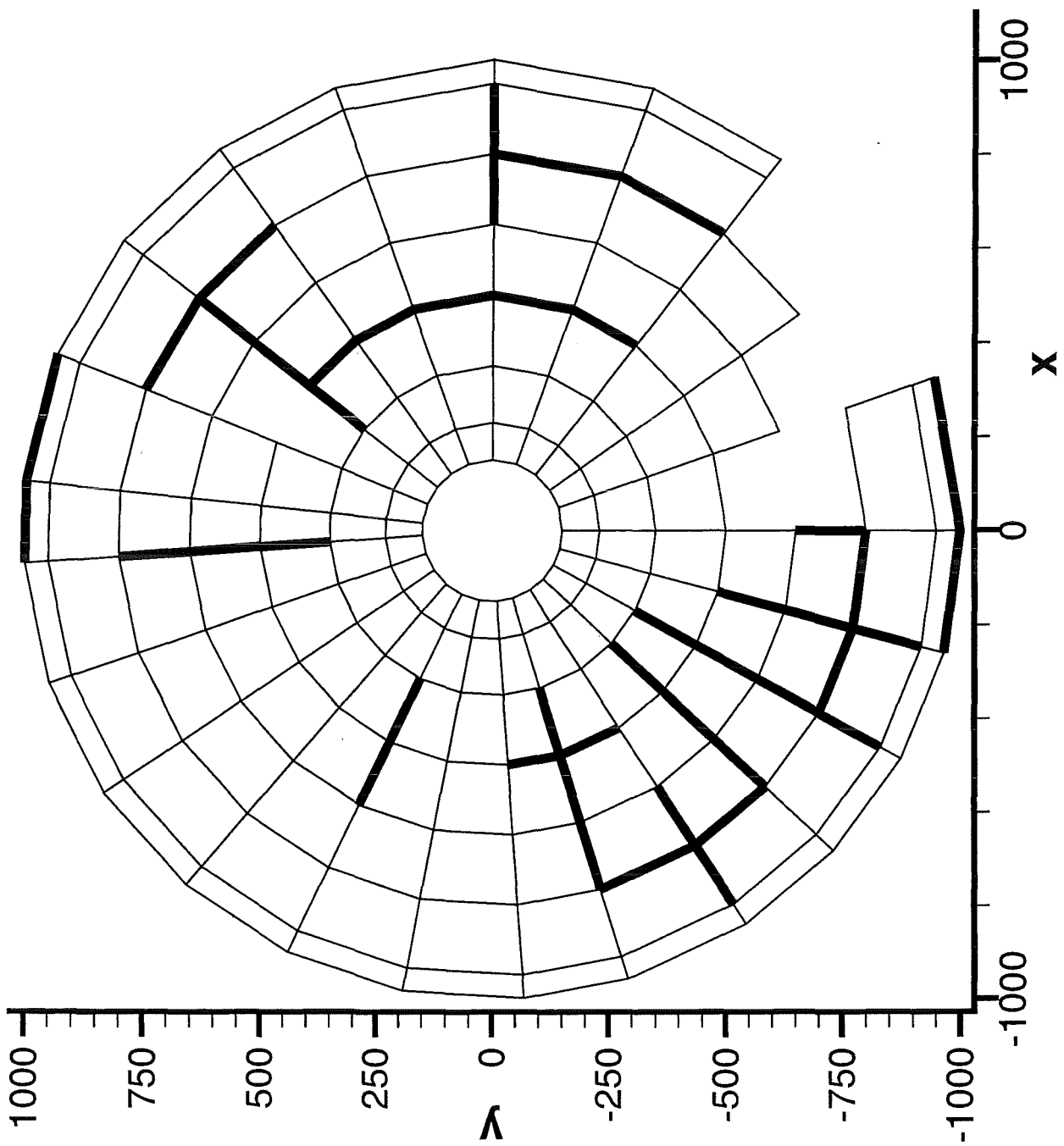


Bild 3.b: Wandstrukturen aus dem GASFLOW-Code

## Literatur

- [1a] S. Kleinheins:  
Das VISART-Konzept einer standardisierten Schnittstelle zwischen Codes und Auswerteprogrammen  
Teil 1: Einführung und Überblick  
FZKA 5995, 1997
- [1b] S. Kleinheins:  
Das VISART-Konzept einer standardisierten Schnittstelle zwischen Codes und Auswerteprogrammen  
Teil 2: VISART — ein standardisiertes Postprocessor-Dateiformat  
FZKA 5996, 1997
- [1c] S. Kleinheins:  
Das VISART-Konzept einer standardisierten Schnittstelle zwischen Codes und Auswerteprogrammen  
Teil 3: VISAPTER — ein Programm zum Anschluß einiger Visualisierungssysteme an VISART-Dateien  
FZKA 5997, 1999
- [1d] S. Kleinheins:  
Das VISART-Konzept einer standardisierten Schnittstelle zwischen Codes und Auswerteprogrammen  
Teil 4: VISARTOP — ein Programm zum Bearbeiten von VISART-Dateien  
FZKA 5998, 1999
  
- [2] Historian Plus User's Manual  
Release 4.3.137  
HPCSA, Austin, Texas, August 1991
  
- [3a] Tecplot User's Manual, Version 7  
Amtec Engineering, Inc., Bellevue WA, USA, August 1996
- [3b] Tecplot Reference Manual, Version 7  
Amtec Engineering, Inc., Bellevue WA, USA