

KfK 2818  
November 1979

# **STRUYA**

## **Ein Programm zur Berechnung zweidimensionaler Strömungsvorgänge mit und ohne Strukturkopplung**

F. W. Katz, E. G. Schlechtendahl, K. Stölting  
Institut für Reaktorentwicklung

**Kernforschungszentrum Karlsruhe**



KERNFORSCHUNGSZENTRUM KARLSRUHE  
Institut für Reaktorentwicklung

KFK 2818

S T R U Y A

Ein Programm zur Berechnung zweidimensionaler Strömungs-  
vorgänge mit und ohne Strukturkopplung.

F.W. Katz  
E.G. Schlechtendahl  
K. Stölting

Kernforschungszentrum Karlsruhe G.m.b.H., Karlsruhe

Als Manuskript vervielfältigt  
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH  
ISSN 0303-4003

## Zusammenfassung

STRUYA erlaubt die Berechnung zweidimensionaler Strömungsvorgänge im Unter- und Überschallbereich in Eulerscher und Lagrange-scher Behandlung. In der dritten Dimension kann der Strömungsbereich durch eine bewegte Wand begrenzt sein. Die Bewegung der Wand kann orts- oder zeitveränderlich vorgegeben werden oder durch ein Strukturmodell berechnet werden. STRUYA bietet eine allgemeine Systematik zum Einbau verschiedener Strukturmodelle an. Ein Modell für zylindrische Schalen (CYLDY2) gehört zur Grundausrüstung.

## STRUYA a Code for Two-Dimensional Fluid Flow Analysis With and Without Structure Coupling

### Abstract

STRUYA is a code for two-dimensional subsonic and supersonic flow analysis. Both Eulerian and Lagrangian grids are allowed. In the third dimension the flow domain may be bounded by a moving wall. The wall movement may be prescribed in a time- and space varying way or computed by a structural model. STRUYA offers a general scheme for adapting various structural models. As a standard feature it includes a cylindrical shell model (CYLDY2).

# I n h a l t

		Seite
1.	Einleitung	1
2.	YAQUIR in STRUYA	3
2.1	Die Lösungsmethode	3
2.2	Die Anwendung der ICED-ALE Methode	5
2.2.1	Maschengitter	6
2.2.2	Zustandsgrößen	9
2.2.3	Zustandsgleichung	10
2.2.4	Randbedingungen	11
2.2.5	Spannungstensor	11
2.2.6	Schrittweiten	12
2.2.7	Einheiten	13
3.	Modifikationen zur 2 1/2 D-Realisation	14
3.1	Ableitung der Fluidgleichungen	14
3.2	Impulsgleichung für die Wandbewegung	21
3.3	Ableitung der Gleichungen für die Wandbewegung	23
4.	Strukturkopplung	32
4.1	Kommunikationsroutinen	32
4.2	Initialisierung eines Struktursubsystems	35
4.3	Datenaustausch zwischen den Strukturroutinen	35
5.	Benutzerhandbuch	40
5.1	Aufruf von STRUYA	40
5.2	STRUYA Syntax	41
5.3	Das Anwendungsprogramm	87
5.3.1	Erzeugung von Problemdaten	87
5.3.2	Graphische Darstellung von Ergebnissen	91
5.3.3	Reparatur einer STRUYA Datei	93
5.3.4	Restart	94

## A n h a n g

	Seite	
A 1	Strukturmodule	105
A 2	Standard- und DUMMY-Routinen	114
A 3	STRUYA COMMON und STATUS	120
A 4	Behandlung der Viskosität in STRUYA	128
A 5	Syntaxnotation	132
Literatur		135

## 1. Einleitung

STRUYA (Struktur YAQUI) ist ein REGENT-Subsystem / 1 /, das erlaubt, Strukturdynamik-Codes mit einer 2 1/2 D-Version des im IRE entwickelten Fluidodynamik-Codes YAQUIR zu koppeln. STRUYA ist also eine Weiterentwicklung von YAQUIR und basiert wie dieses auf dem Los Alamos (LASL) Code YAQUI / 2 /. YAQUI ist ein Lagrange-Eulersches Rechenprogramm für Fluidströmungen mit beliebigen Geschwindigkeiten. In der LASL-Version wurde der Code in FORTRAN auf einer Rechenanlage vom Typ CDC 7600 programmiert, die REGENT-Versionen haben PL/1 als Programmiersprache und wurden auf Maschinen IBM 370 getestet.

Bei der Implementierung von YAQUIR als Subsystem von REGENT konnten die Vorteile, die REGENT bietet, ausgenutzt werden. So wurden u.a. folgende Punkte beachtet bzw. verbessert, um einerseits die Anwendungsmöglichkeiten zu erweitern, andererseits aber auch die Anwendung zu vereinfachen:

- Definition von subsystemspezifischen Sprachelementen zwecks Erweiterung des Sprachumfangs.
- einfache Anwendung durch verständliche Eingabeweisungen, dadurch Verminderung der Eingabevorschriften und möglicher Fehlerquellen.
- Austauschbarkeit von Programmteilen (z.B. Initialisierungsroutinen, Standardroutinen, Randbedingungen, Zustandsgleichung) zwecks Berechnung verschiedenartiger Fluid-Strukturen.
- Definition von Datenstrukturen, die im ganzen Subsystem verfügbar sind.

STRUYA enthält praktisch alle die Fähigkeiten von YAQUIR für fluiddynamische Probleme, ist aber im allgemeinen effektiver. Die wesentlichen zusätzlichen Fähigkeiten sind:

- Vorgabe nicht konstanter Fluidschichtdicke bei ebenen, bzw. nicht konstanter Radien bei Zylinderkoordinaten.

- Interface für die Kopplung eines Strukturmodells.
- Einen erweiterten Satz Zustandsgrößen zur Übernahme der Strukturbewegung.
- Einen Satz Dummy-Routinen, die bei Kopplung mit Struktur automatisch durch entsprechende Strukturroutinen ersetzt werden.

Möglichkeiten zur graphischen Ausgabe der Ergebnisse auf Papier, Bildschirm oder Film sind zur Darstellung des Fluidnetzes, der Fluidgeschwindigkeit, Druckverteilung und Strukturverbiegung in STRUYA vorhanden.

## 2. YAQUIR in STRUYA

Der hydrodynamische Teil in STRUYA benutzt ein zweidimensionales Gitter zur Lösung von Differenzgleichungen für Masse, Impuls und Energie. Das Lösungsverfahren entspricht der in YAQUIR angewandten und in YAQUI entwickelten ICED-ALE Methode. Diese wird im folgenden beschrieben.

### 2.1 Die Lösungsmethode

Im vergangenen Jahrzehnt wurden bei der Entwicklung von numerischen Lösungsverfahren mehrdimensionaler Probleme der Fluid-Dynamik erhebliche Fortschritte erzielt. Eine Reihe grundlegender Methoden hat sich gut durchgesetzt und wurde durch nützliche und praktische Anwendungen aus sich ständig erweiternden Problemkreisen vieler technischer Gebiete bestätigt.

Numerische Methoden behandeln eine Flüssigkeit, indem sie sich auf die Beschreibung einer endlichen, diskreten Menge von Fluid-elementen beziehen. Deshalb wird das Gebiet, das von Interesse ist, üblicherweise in ein endliches Gitter bzw. in Maschen berechenbarer Zonen unterteilt. Jeder Zone bzw. jedem Knoten wird ein lokaler Wert einer Zustandsgröße, wie z.B. Masse, Energie oder Geschwindigkeit, zugeordnet.

Die Differential-Gleichungen werden durch finite Differenzen in Bezug auf das Gitter angenähert. Um die zeitliche Änderung der Zustandsgrößen während eines Beobachtungszeitraumes zu erhalten, wird die Zeitspanne in mehrere Intervalle unterteilt und das Gleichungssystem zu diesen Zeiten gelöst.

Die verschiedenen numerischen Verfahren werden meist nach zwei Aspekten klassifiziert:

- Kompressibilität oder Inkompressibilität
- Relativbewegung von Gitter und Fluid

Verfahren für kompressible Strömungen werden eingesetzt, wenn die Fluid-Geschwindigkeit vergleichbar oder größer als die lokale Material-Schallgeschwindigkeit ist. In inkompressiblen Strömungen (oder Strömungen mit kleinen Geschwindigkeiten bzw. Mach-Zahlen)

sind die Fluid-Geschwindigkeiten wesentlich geringer als die Material-Schallgeschwindigkeit. Da viele Fluid-Dynamik-Probleme nicht in einen dieser beiden Extremfälle einzuordnen sind, können sie auch nicht ausreichend genau von Hoch- oder Niedergeschwindigkeits-Methoden gelöst werden. Beispiele hierfür sind Strömungen, die zu Anfang im Überschall liegen, aber sehr schnell in das Unterschall-Gebiet gelangen oder Strömungen, bei denen Teilbereiche im Überschall-Gebiet, andere Teilbereiche im Unterschall-Gebiet liegen. Ein ähnlicher Zustand ist beim Blowdown eines Druckwasserreaktors zu erwarten. Der Strömungsvorgang mit sehr großen Geschwindigkeiten im Bruchstutzen und niedrigen Geschwindigkeiten im unteren Ringraum ist ein wichtiger Punkt bei den Sicherheitsberechnungen für Reaktoren.

Der zweite Aspekt betrifft die Beziehung zwischen Fluid und dem Koordinaten-Gitter. Es gibt zwei grundlegende Betrachtungsweisen, sowohl für Überschall wie auch für Unterschall. Die erste ist die Lagrangesche Methode, bei der das Maschengitter sich mit dem Fluid bewegt. Bei diesem Verfahren wird die Genauigkeit der Berechnung begrenzt durch die Unfähigkeit des Gitters, sich mit großen Verformungen, die viele Strömungen charakterisieren, zu bewegen. Die zweite Methode ist die Eulersche Methode, die das Maschengitter als festgehaltenen Rahmen benutzt, durch den sich die Flüssigkeit bewegt.

Gegenwärtig ist eine erfolgreiche Methode, Strömungen bei allen Geschwindigkeiten zu berechnen, die implizite Technik für kontinuierliche Fluide nach Euler (Implicit Continuous-fluid Eulerian technique = ICE). Hier darf die Strömung von tiefen Unterschall bis Überschall variieren. Das wird erreicht durch die implizite Behandlung der Druckberechnung. Die Methode ist vielseitig anwendbar für Berechnungen in ein, zwei oder drei Raumdimensionen und willkürlichen Zustandsgleichungen.

Gleichzeitig wurden Methoden entwickelt, die in hohem Maße die Vorteile beider Lösungsverfahren (Lagrange und Euler) vereinen. Bei einigen Methoden werden Lagrange-Partikel verwandt, um Fluid-Trennlinien oder freie Oberflächen zu definieren oder das Fluid

selbst innerhalb eines Euler-Gitters zu vereinbaren. Es gibt noch andere Methoden, die nicht grundsätzlich auf Partikel basieren. Eine solche ist die beliebige Lagrange-Euler Methode (Arbitrary Lagrangian-Eulerian method=ALE), eine Niedergeschwindigkeits-Methode, die es erlaubt, die Knoten des Gitters mit dem Fluid zu bewegen (nach Lagrange) oder in dem Fluid festzuhalten (nach Euler) oder in irgend einer beliebig definierten Weise zu bewegen, um eine Rezoning-Fähigkeit zu erhalten. Es können größere Verzerrungen in der Fluid-Bewegung behandelt werden als bei der Lagrange-Methode, zudem mit einer besseren Lösung als bei der Euler-Methode.

Die in STRUYA implementierte ICED-ALE-Methode bietet die Möglichkeit, zwischen Lagrangescher oder Eulerscher Modellierung zu wählen. Damit ist STRUYA ein sehr flexibel einsetzbarer Code für viele zweidimensionale Fluidodynamik-Probleme. Im Falle zeitveränderlicher Fluidschichtdicke ist jedoch nur die Eulersche Modellierung erlaubt.

## 2.2 Die Anwendung der ICED-ALE Methode

Der eigentliche Kern des Codes, der die ICED-ALE-Methode beschreibt, ist der Hydrodynamik-Teil. Dieser Abschnitt wird für jeden Zyklus durchlaufen und besteht aus drei Berechnungs-Phasen:

- Phase 1: explizite Lagrange-Berechnung.
- Phase 2: Iteration, die die Drücke für die Impulsgleichungen und die Dichten für die Massengleichungen berechnet.
- Phase 3: Rezone-Abschnitt, der die Berechnung der fortschreitenden Strömung ausführt.

Um eine Berechnung durchführen zu können, muß das physikalische Problem in einem Modell wiedergegeben werden, das in STRUYA verarbeitet werden kann. Dazu werden in den folgenden Kapiteln die nötigen Informationen gegeben.

### 2.2.1 Maschengitter

Das benutzte Maschengitter besteht aus einem 2-dimensionalen Netz von viereckigen Zellen, mit denen Berechnungen entweder in Zylinder- oder in kartesischen Koordinaten durchgeführt werden. Berechnungen in Zylinderkoordinaten sind bezogen auf einen Einheitswinkel, so daß die Gleichungen ohne die Kreiszahl  $\pi$  geschrieben werden können. Im kartesischen Koordinatensystem wird den Maschen die Dicke 1 zugeordnet.

Die Radialkoordinate wird mit  $r$  oder  $x$  bezeichnet, die Axialkoordinate mit  $z$  oder  $y$ , wobei der Ursprung des Koordinatensystems im allgemeinen in der linken unteren Ecke des Maschennetzes liegt. Die Knoten des Gitters werden mit den Indizes  $i$  (radiale Richtung) und  $j$  (axiale Richtung) durchnummeriert. Die Numerierung einer Masche ist identisch mit der des linken unteren Knotens an der Masche (Abb.2.1)

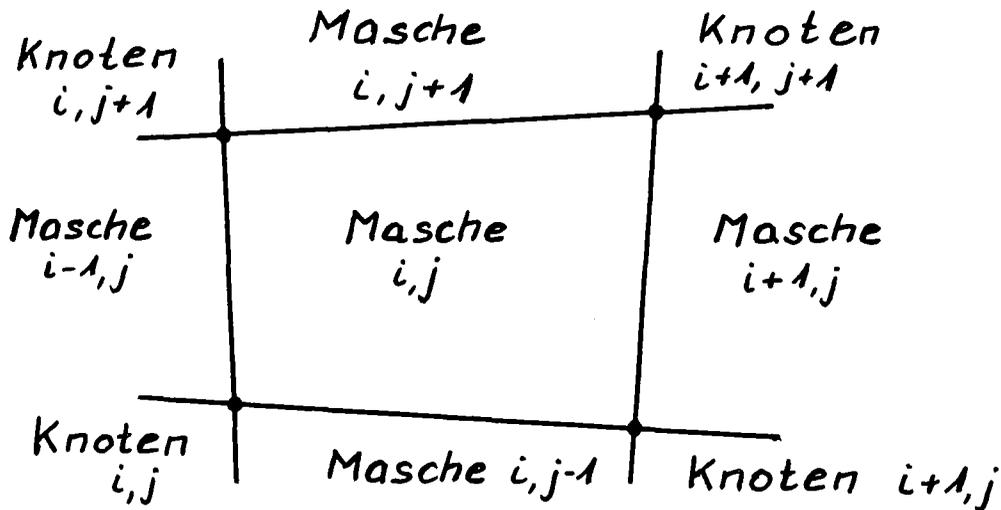


Abb.2.1: Maschen- und Knotenindizierung

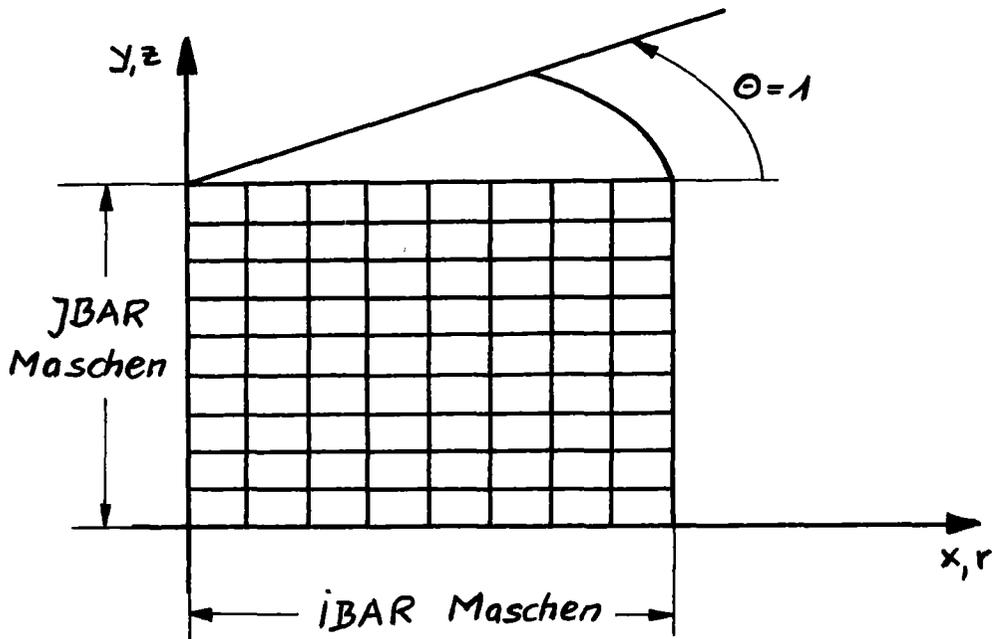


Abb.2.2: Ein ICED-ALE-Maschengitter in Zylinderkoordinaten

Das in Abb.2.2 gezeigte rechteckige Maschengitter in Zylinderkoordinaten besteht aus Ausschnitten rotations-symmetrischer Toroide. Das Gitter besteht in radialer Richtung aus IBAR Maschen, in axialer Richtung sind JBAR Maschen angeordnet.

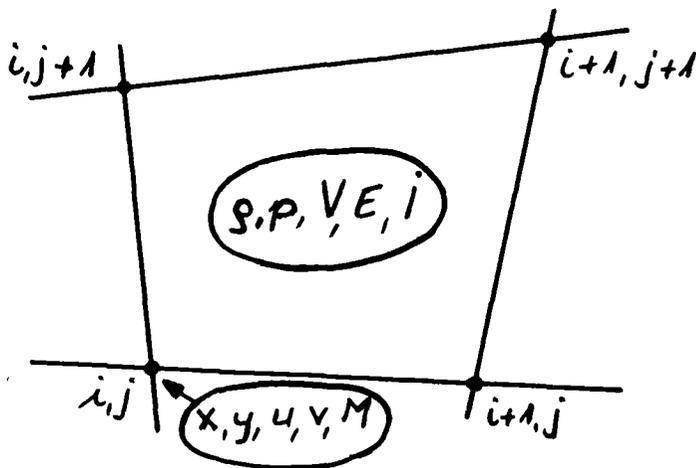


Abb.2.3: Eine ICED-ALE-Masche (i, j) mit den wichtigsten Zustandsgrößen für Knoten und Masche

Die Variablen der ICED-ALE-Gitter werden zwei Gruppen zugeordnet (s. Abb. 2.3), der Gruppe der Variablen, die an den Knoten definiert sind, das sind die Koordinaten  $x$  und  $y$ , die Geschwindigkeiten  $u$  und  $v$  und die Massen  $M$ , und der Gruppe der Variablen, die in den Maschenmitten definiert sind, hierzu gehören die Dichten  $\rho$ , die Drücke  $p$ , die Volumina  $V$  und die Energien  $E$  und  $I$ .  $E$  ist die spezifische Gesamtenergie und  $I$  die spezifische innere Energie.

Die spezifischen Energien sind auf die Massen bezogen, deshalb werden sie in der Dimension  $\text{Länge}^2/\text{Zeit}^2$ , z.B.  $\text{m}^2/\text{sec}^2$  angegeben.

Die Anzahl der Knoten in jeder Richtung ist um 1 größer als die Anzahl der Zellen. Die Indizierung bezieht sich auf die Maschenmitten bzw. auf die linken, unteren Knoten, so daß wir jeweils eine zusätzliche Maschenreihe an der rechten und an der oberen Begrenzung des Gitters zulassen können. YAQUIR erzeugt überdies eine Maschenreihe an der unteren Begrenzung. Das so erzeugte Maschengitter ist also  $(\text{IBAR}+1)$  Maschen breit und  $(\text{JBAR}+2)$  Maschen hoch. Die äußeren Maschen werden als fiktive Maschen bezeichnet und sind nur an drei Seiten vorhanden. Die fiktiven Maschen werden allerdings nicht am linken Rand benutzt. Da der Code ursprünglich für Berechnungen in Zylinderkoordinaten konzipiert wurde, ist der linke Rand als Symmetrieachse vorgesehen. In ebenen Koordinaten ist dieser Rand eine reibungsfreie Wand (free-slip wall) oder Symmetrieebene. Das Maschengitter aus Abb. 2.2 wird in STRUYA zu einem Maschengitter, wie es in Abb. 2.4 gezeigt ist.

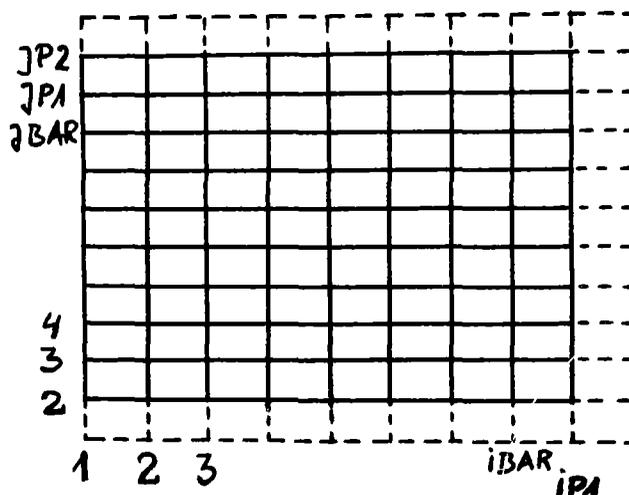


Abb. 2.4: STRUYA - Maschengitter

Deshalb haben die doppelten DO-Schleifen, die alle Knoten abarbeiten, die Grenzen

```
DO J = 2 TO JP2; und
```

```
DO I = 1 TO IP1;
```

Entsprechend haben die DO-Schleifen über alle Maschen die Grenzen

```
DO J = 2 TO JP1; und
```

```
DO I = 1 TO IBAR;
```

Ein solches Gitter kann die Fluid-Region und die Partikel-Region definieren.

### 2.2.2 Zustandsgrößen

Vor dem Start einer Berechnung müssen an den Knoten bzw. in den Maschenmitten alle Zustandsgrößen definiert werden (s.auch Abb.2.3). So werden die an den Knoten herrschenden Geschwindigkeiten mit den Komponenten in radialer und axialer Richtung (u bzw.v) benötigt. Den Maschenmitten müssen die Größen zugeordnet werden, die den Energie-Zustand definieren. Dieser Zustand wird durch die Dichte  $\rho$  und die spezifischen inneren Energien I festgelegt, die zur Berechnung der Drücke p in den Maschen dienen.

Neben diesen Zustandsgrößen, die der Benutzer für sein Beispiel angeben muß, werden in dem Code noch weitere Zustandsgrößen aus den gemachten Angaben berechnet.

An den Knotenpunkten werden Knotenpunktmassen definiert, und in den Maschenmitten werden die Drücke p, die Maschenvolumina V und die Maschen-Gesamtenergien E errechnet.

Bei der Berechnung von explosionsartigen Vorgängen ist darauf zu achten, daß die Druckgradienten im Anfangszustand nicht zu groß gewählt werden. Als Richtwert kann dienen, daß das Verhältnis

der Drücke von angrenzenden Maschen den Wert 10 nicht übersteigen sollte. Dies gilt ganz besonders für den Gradient zweier Drücke benachbarter Maschen mit einem gemeinsamen Knoten.

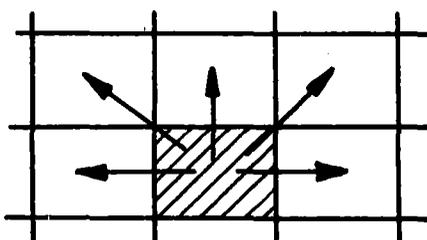


Abb.2.5: Druckgradient von benachbarten Maschen

Entsprechend sind die Dichten und die spezifischen Energien zu wählen, die für die Bestimmung des Druckes herangezogen werden können.

### 2.2.3 Zustandsgleichung

Die Maschen-Drücke werden am Anfang eines jeden Zyklus berechnet, indem die Zustandsgleichung

$$p = p(\rho, I)$$

benutzt wird.

Für kleine Mach-Zahlen, für inkompressible Strömungen, ist die Beschreibung einer Zustandsgleichung unnötig, weil die Änderungen im Druck sich direkt als Folge der Dynamik ergeben. In STRUYA ist jedoch die Zustandsgleichung in jedem Fall vorhanden, indem in der Phase 2 (implizite Iteration) das Quadrat der Schallgeschwindigkeit  $c^2$  benutzt wird. Die Zustandsgleichung wird direkt zur Iteration des Druckes herangezogen, um die Dichteänderung für die Phase 3 der Berechnung zu bestimmen.

#### 2.2.4 Randbedingungen

Verschiedene Randbedingungen können in einem ICED-ALE Programm verwandt werden. Standardmäßig ist der einfache Fall von starren, rechtwinklig angeordneten Rändern an allen 4 Seiten des Gitters vorgesehen.

Die reflektierenden Ränder werden als reibungsfrei angesehen, wobei der linke Rand bei Berechnungen in Zylinderkoordinaten zur Symmetrieachse wird.

Das Kriterium für eine vom Anwender gegebene Randbedingung ist, daß die Geschwindigkeiten der Randknoten in der gewünschten Weise festgelegt werden. Für den Standardfall mit reibungsfreien, festen Rändern (free-slip walls) bedeutet dies, daß die Normalkomponente der Knotengeschwindigkeit am Rand während der gesamten Berechnung zu Null gesetzt werden muß, während die Tangentialkomponenten erhalten bleiben.

An den Rändern können auch Beschleunigungsterme vorgesehen werden, die sich aus Drücken oder Kräften auf die Ränder errechnen lassen bzw. aus den Geschwindigkeiten resultieren.

#### 2.2.5 Spannungstensor

Der in der Phase 1 der ICED-ALE-Methode berechnete Spannungstensor  $P_{ij}$  hat folgende Komponenten:

$$\begin{aligned} P_{ii} &= 2 \cdot \mu \frac{\partial u}{\partial x} + \lambda \nabla \cdot \vec{u} \\ P_{jj} &= 2 \cdot \mu \frac{\partial v}{\partial y} + \lambda \nabla \cdot \vec{u} \\ P_{ij} &= \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \end{aligned}$$

Hier ist  $\mu$  die Schubspannung und  $\lambda = \zeta - \frac{2}{3}\mu$ , wobei  $\zeta$  der Koeffizient

für die Ausdehnungs-Viskosität ist. Z.B. ist für ideale Gase der Wert  $\zeta = 0$ .

Die Größen  $\mu$  und  $\lambda$  werden in vielen Veröffentlichungen auch als erster ( $\mu$ ) und zweiter ( $\lambda$ ) Viskositätskoeffizient bezeichnet.

### 2.2.6. Schrittweiten

Bei einer Berechnung von Zustandsänderungen hat man zwei Möglichkeiten, die Zeitabhängigkeit zu wählen.

Zum einen kann man während der Beobachtungsdauer konstante Zeitintervalle  $\delta t$  vorschreiben, zum anderen aber auch im Laufe der Rechnung größer werdende Schrittweiten erlauben.

Die selbständige Berechnung der Schrittweite  $\delta t$  ist in STRUYA als Option vorgesehen. Zwei Stabilitätsbedingungen bestimmen jeweils eine neue Schrittweite. Die eine Bedingung wird gegeben durch die Zähigkeitsspannungen und den Koeffizienten  $\mu$  und  $\lambda$ . Die andere ist verbunden mit der fortschreitenden Strömung in der Euler-Berechnung oder durch die Verhinderung von negativen Volumina in der Lagrange-Berechnung. / siehe 2, S 23 ff /

$$\delta t < \left[ \frac{2(\lambda + 2\mu)}{\rho} \cdot \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1}$$

Diese Bedingung verlangt, daß sich der Impuls innerhalb eines Zeitschrittes nur um weniger als eine Maschenweite ausbreiten darf.

$$\delta t < f \cdot \frac{\delta x}{\delta u}, \quad f < 1.0$$

erlaubt nur Strömungsgeschwindigkeiten von weniger als einer Maschenweite pro Zeitschritt, da von der Annahme ausgegangen wird, daß ein Austausch nur zwischen benachbarten Maschen stattfinden kann (Courant-Bedingung).

Ein neues Zeitintervall ergibt sich aus dem Minimalwert der beiden Bedingungen und dem aktuellen Zeitschritt, vergrößert um den Faktor 1.25.

### 2.2.7 Einheiten

Die dimensionsbehafteten Größen bei einem Anwendungsbeispiel sind:

Koordinaten X und Y bzw. R und Z

Geschwindigkeiten U und V

Dichte  $\rho_0$

spezifische innere Energie SIE

Druck P (bzw.  $f(\rho_0, SIE)$ )

Schallgeschwindigkeit CSQUARE (bzw.  $f(\rho_0, SIE)$ )

Zeitintervall DT

Beschleunigung bzw. Gravitation b

Da im Programm nicht zwischen verschiedenen Dimensionen umgerechnet wird, muß allen Größen ein einheitliches Maß-System zugrunde liegen. Empfohlen werden die Einheiten m, kg, sec.

### 3. Modifikationen zur 2 1/2 D - Realisation

#### 3.1 Ableitung der Fluidgleichungen

- 1) Betrachtet wird ein zweidimensionales Gebiet, das in vier-eckige Maschen eingeteilt ist. Jedem Gitterpunkt des Ma-schennetzes ist eine Höhe  $H$  (genauer  $H_{i,j}$ ) zugeordnet. Zwischen den Gitterpunkten wird ein bilinearer Verlauf der Höhe angenommen.
- 2) Geschwindigkeiten und deren Wirkungen senkrecht zur Haupt-strömungsebene werden vernachlässigt.
- 3) Es wird angenommen, daß Fluid und Struktur durch unter-schiedliche Modelle dargestellt sind.

Wir gehen aus von den Grundgleichungen gemäß / 3, Gl.6a-6c /, mit  $\bar{n}$  als äußerer Normalen.

$$\frac{d}{dt} \int_V \rho dV - \int_S \rho (\bar{U} - \bar{u}) \cdot \bar{n} dS = 0 \quad (1a)$$

$$\begin{aligned} \frac{d}{dt} \int_V \rho \bar{u} dV - \int_S \rho \bar{u} [(\bar{U} - \bar{u}) \cdot \bar{n}] dS \\ + \int_V \nabla \rho dV - \int_V \rho \bar{g} dV = 0 \end{aligned} \quad (1b)$$

$$\begin{aligned} \frac{d}{dt} \int_V \rho E dV - \int_S \rho E (\bar{U} - \bar{u}) \cdot \bar{n} dS \\ + \int_S \rho \bar{u} \cdot \bar{n} dS - \int_V \rho \bar{g} \cdot \bar{u} dV = 0 \end{aligned} \quad (1c)$$

Wir formulieren 1b noch um in

$$\frac{d}{dt} \int_V \rho \bar{u} dV - \int_S \rho \bar{u} [(\bar{U} - \bar{u}) \cdot \bar{n}] dS + \int_S \bar{n} p dS - \int_V \rho \bar{g} dV = 0 \quad (1b)$$

und erhalten für 2 1/2-dimensionale Probleme mit

$$\begin{aligned} dV &= H dF \\ dS &= H d\sigma && \text{entlang der Maschenränder } \sigma \\ \bar{n} dS &= \bar{n}_b dF && \text{an der beweglichen Wand} \\ \bar{n} dS &= \bar{n}_f dF && \text{an der festen Wand} \\ \bar{n} \bar{u} = \bar{n} \bar{U} &= -\dot{H} && \text{an der beweglichen Wand} \end{aligned}$$

$$\int_F \frac{d(\rho H)}{dt} dF + \int_{\sigma} (\rho H) \bar{u} \bar{n} d\sigma = 0 \quad (2a)$$

$$\begin{aligned} \int_F \frac{d}{dt} (\rho H \bar{u}) dF + \int_{\sigma} \rho H \bar{u} [\bar{u} \bar{n}] d\sigma + \int_{\sigma} \bar{n} p H d\sigma \\ - \int_F \rho H \bar{g} dF + \int_F (\bar{n}_b + \bar{n}_f) p dF = 0 \end{aligned} \quad (2b)$$

$$\begin{aligned} \int_F \frac{d}{dt} (\rho E H) dF + \int_{\sigma} \rho E H \bar{u} \cdot \bar{n} d\sigma + \int_{\sigma} \rho \bar{u} \cdot \bar{n} H d\sigma \\ - \int_F \rho H \bar{g} \cdot \bar{u} dF - \int_F \rho H \dot{H} dF = 0 \end{aligned} \quad (2c)$$

Gegenüber den in YAQUI enthaltenen Gleichungen stellt man formal fest, daß lediglich der Radius  $R$  der Zylinderkoordinaten durch die Höhe  $H$  ersetzt ist und in Impuls- bzw. Energiegleichung ein Zusatzterm auftritt.

Aus diesen Änderungen in den Grundgleichungen lassen sich für 2 1/2-dimensionale Probleme bei vorgegebener Wandbewegung  $\dot{H}$  die nötigen Änderungen in den Differenzgleichungen und schließlich im YAQUI-Code ableiten. Bei der Kopplung mit der Struktur muß jedoch auch die Impulsgleichung für die Richtung normal zu den Hauptströmungsrichtungen berücksichtigt werden, da sie eine Beziehung zwischen  $\dot{H}$  und  $p$  darstellt. Im folgenden werden die wesentlichen Änderungen in den Differenzgleichungen erläutert.

Für die Behandlung der Impulsgleichung wählen wir folgende Darstellung eines 1 1/2-dimensionalen Sonderfalles, da die graphische Repräsentation des allgemeinen 2 1/2-dimensionalen Falles unübersichtlich wird.

Wir benutzen ein Kontrollvolumen, das die dem Knoten  $K$  benachbarten Maschen fiktiv so ergänzt oder beschneidet, daß das Ersatzkontrollvolumen überall dieselbe Höhe hat.

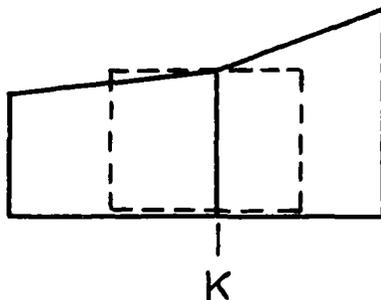


Abb.3.1:  
Ersatzkontrollvolumen  
für Knoten  $K$

Für diese Ersatzkontrollvolumen ist die Impulsbilanz und die Masse zu berechnen. Dazu betrachten wir den Knoten und seine Umgebung von oben.

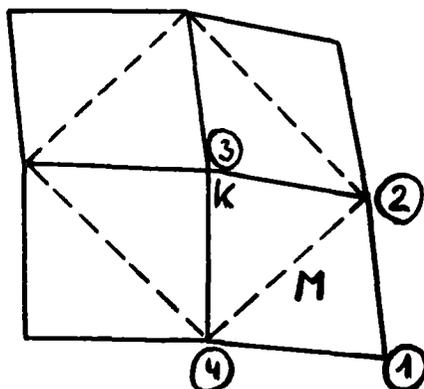


Abb.3.2:  
Knoten und benachbarte  
Maschen

In Abb.3.2 ist ein Bereich gestrichelt, der der doppelten Masse des Knotens K entspricht. Die Massen für alle Knoten erhalten wir durch einen Durchgang über alle Maschen M, wobei wir die Anteile der Maschenmassen den 4 Eckknoten zuschlagen (vorher alle Knotenmassen Null setzen).

$$\begin{aligned}
 M_1 &= M_1 + \sum_M \cdot ABR \cdot r_1 / 2 \\
 M_2 &= M_2 + \sum_M \cdot ATR \cdot r_2 / 2 \\
 M_3 &= M_3 + \sum_M \cdot ATL \cdot r_3 / 2 \\
 M_4 &= M_4 + \sum_M \cdot ABL \cdot r_4 / 2
 \end{aligned}
 \tag{3}$$

ATR und ABL sind aus / 2 / bekannt. ATL und ABR sind entsprechend zu berechnen.

$$\begin{aligned}
 ATR &= 0,5 \cdot [x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)] \\
 ABL &= 0,5 \cdot [x_1(y_3 - y_4) + x_3(y_4 - y_1) + x_4(y_1 - y_3)] \\
 ATL &= 0,5 \cdot [x_2(y_3 - y_4) + x_3(y_4 - y_2) + x_4(y_2 - y_3)] \\
 ABR &= 0,5 \cdot [x_4(y_1 - y_2) + x_1(y_2 - y_4) + x_2(y_4 - y_1)]
 \end{aligned}
 \tag{4}$$

Hierdurch wird die einfachere Gleichung

$$M_i^j = \frac{1}{4} \left( M_{i+1/2}^{j+1/2} + M_{i-1/2}^{j+1/2} + M_{i-1/2}^{j-1/2} + M_{i+1/2}^{j-1/2} \right)$$

in Phase 1 und Phase 3 ersetzt. Auch die Impulsgleichung ist hiermit konsistent mit dem Ersatzkontrollvolumen zu formulieren. Also z.B.

$$\tilde{u}_1 = \tilde{u}_1 + \frac{\delta t}{2M_1} \cdot \left\{ r_1 \left[ \bar{\pi}_{xx} \cdot (y_4 - y_2) + \bar{\pi}_{xy} \cdot (x_2 - x_4) + \rho (y_2 - y_4) \right] - PTH \right\} \quad (5)$$

etc. bis

$$\tilde{v}_4 = \tilde{v}_4 + \frac{\delta t}{2M_4} \cdot \left\{ r_4 \left[ \bar{\pi}_{xy} \cdot (y_3 - y_1) + (\bar{\pi}_{yy} - \rho)(x_1 - x_3) \right] \right\}$$

Anmerkung: Dieses Verfahren erlaubt keine Knoten, bei denen  $r = 0$  ist!

Diese Berechnung der Massen und Fluidgeschwindigkeiten wird jedoch nur für 2 1/2 D-Probleme eingeführt. Für Zylinderkoordinaten werden die ursprünglichen Gleichungen / 2 / benutzt. Das geschieht auf zwei verschiedene Arten: Die Art der Massenberechnung wird durch Abfrage eines Parameters #CYL entschieden, der mit dem Eingabestatement TYPE gesetzt wird. Für die Berechnung der Fluidgeschwindigkeiten kann #CYL als Faktor verwendet werden, wegen:

#CYL = 1. für Zylinderkoordinaten

#CYL = 0. für 2 1/2 D-Probleme.

Hierzu wird benutzt bei der Initialisierung und in Phase 3:

$$R1CYL = r_1 + \#CYL \cdot [0.5 \cdot (r_2 + r_4) - r_1] \quad (6)$$

Entsprechend lauten die Gleichungen in Phase 2

$$\begin{aligned} u_{L1} &= u_{L1} + \frac{\delta t}{2M_1} \cdot r_1 \cdot (y_2 - y_4) \cdot \delta p \\ &\vdots \\ v_{L1} &= v_{L1} + \frac{\delta t}{2M_1} \cdot R1CYL \cdot (x_4 - x_2) \cdot \delta p \end{aligned} \quad (7)$$

Die Behandlung der anderen 3 Knoten sowie der Randbedingungen erfolgt in ähnlicher Weise.

In Phase 3 wird (unter Verwendung der ausiterierten Werte für Druck und Geschwindigkeiten) der Zusatzterm  $-\int p H \dot{H} dF$  der Energiegleichung berücksichtigt. In den Differenzgleichungen wird daraus:

$$\frac{\tilde{E}_{i+1/2}^{j+1/2} - \langle \tilde{E} \rangle_{i+1/2}^{j+1/2}}{\delta t} \cdot M_{i+1/2}^{j+1/2} = \text{die in / 2 / enthaltenen Terme} \quad (8)$$

$$+ p_{i+1/2}^{j+1/2} \cdot \frac{1}{4} (\dot{H}_1 + \dot{H}_2 + \dot{H}_3 + \dot{H}_4) \cdot V_{i+1/2}^{j+1/2}$$

Hierin wurde für den Mittelwert von H in der Masche das arithmetische Mittel der Knotenpunktwerte benutzt. Dies läßt sich noch vereinfachen. Es wird

$$\tilde{E} = \tilde{E}_{YAQUI} + \frac{\delta t}{\rho} \cdot \frac{\rho}{4} \cdot (\dot{H}_1 + \dot{H}_2 + \dot{H}_3 + \dot{H}_4) \quad (9)$$

Für die Druckiteration benötigen wir zunächst den Ausdruck für die Massenquellstärke  $S_q$  (der durch Iteration auf fast Null gebracht werden soll).

In YAQUI ist dies mit der dort berechneten Divergenz D / 2 /:

$$S_q = \frac{1}{\delta t} (\rho_L - \bar{\rho}) + \rho_L \cdot D; \quad \text{mit } \rho_L \approx \bar{\rho} \quad (10)$$

In unserem Fall haben wir

$$\frac{d}{dt} \int_V \rho dV - \int_S \rho (\bar{U} - \bar{u}) \bar{n} dS = \int_V S_q dV \quad (11)$$

Es ist zu bemerken, daß im Code YAQUI gemäß / 2 / hier anders vorgegangen wird als in der zugehörigen Theoriebeschreibung / 3 /.

Die Theorie ist rein Lagrange, während der Code plötzlich eine Eulersche Betrachtung einführt, obwohl vorher und hinterher ein Lagrange-Modell benutzt wird. Wir halten uns an die Eulersche Vorstellung und benutzen ein Kontrollvolumen entsprechend  ${}^nV$ . Dann ist unter Verwendung von "advanced time values" für  $\rho$  und  $\dot{H}$ :

$$\frac{{}^n\bar{H} \cdot {}^nF \cdot ({}^{n+1}\rho - {}^n\rho)}{\delta t} + {}^n\bar{H} \cdot {}^nF \cdot {}^{n+1}\rho \cdot D + {}^{n+1}\rho \cdot {}^{n+1}\bar{H} \cdot {}^nF = S_q \cdot {}^n\bar{H} \cdot {}^nF \quad (12)$$

$$S_q = \frac{1}{\delta t} ({}^{n+1}\rho - {}^n\rho) + {}^{n+1}\rho \cdot D + {}^{n+1}\rho \frac{{}^{n+1}\bar{H}}{{}^n\bar{H}} \quad (13)$$

Die Mittelwerte  $\bar{H}$  und  $\bar{\dot{H}}$  in der Masche sind zu berechnen aus

$$\bar{H} = \frac{1}{4} \cdot (H_1 + H_2 + H_3 + H_4) \quad (14)$$

$$\bar{\dot{H}} = \frac{1}{4} \cdot (\dot{H}_1 + \dot{H}_2 + \dot{H}_3 + \dot{H}_4) \quad (15)$$

In (13) ist D die Divergenz gemäß / 2 / Seite 9. Wir müssen also lediglich in der Berechnung von  $S_q$  den dritten Term dieser Gleichung dem in YAQUI errechneten Wert hinzufügen, um daraus - in unveränderter Weise - die Druckänderung  $\delta p$  zu berechnen. Wir haben bisher angenommen, daß  $\dot{H}$  fest vorgegeben ist. Im Falle einer Fluid-Struktur-Kopplung ist  $\dot{H}$  das Ergebnis einer dynamischen Strukturberechnung.  $\dot{H}$  hängt dann vom Druck ab und muß auch in die Druckiteration einbezogen werden.

Wir nehmen an, daß die Wandverformung im Fluid durch das Wertefeld  $H_i$  ( $i=1$  bis  $N$ ) gekennzeichnet ist und in der Struktur durch das Wertefeld  $X_s$  ( $s=1$  bis  $M$ ) und so müssen wir - um die Modelle koppeln zu können - voraussetzen, daß für  $M \leq N$  eine eindeutige Funktion

$$H_i = T_i(X_s) \quad (16)$$

existiert. Wir nehmen an, diese sei linear, dann ist

$$H_i = \sum T_{is} \cdot X_s \quad (17)$$

Wegen  $M \leq N$  können wir nicht annehmen, daß diese Funktion eindeutig umkehrbar ist. Dies führt zu dem Schluß:

Die Geometrie muß in demjenigen Modell dargestellt und berechnet werden, das weniger Freiheitsgrade besitzt. Von dort kann sie jederzeit kinematisch kompatibel in das Modell mit mehr Freiheitsgraden umgerechnet werden.

Das bedeutet, die Geschwindigkeiten und Verformungen der Wand müssen im Strukturmodell berechnet und dann in das Fluidmodell übertragen werden.

### 3.2 Impulsgleichung für die Wandbewegung

Zur Lösung der Impulsgleichung für die Wandbewegung muß grundsätzlich folgende Gleichung bestimmt werden:

Beschleunigung = f (Druckdifferenzverteilung ,  
Verformung der Wand,  
Steifigkeitseigenschaften der Wand,  
effektiv bewegte Masse,  
bei Dämpfung auch Geschwindigkeit)

Diese Größen sind wie folgt den Modellen zugeordnet:

Struktur	Fluid
Beschleunigung	Druckverteilung
Geschwindigkeit	
Verformung	
Steifigkeitseigenschaften	
ein Anteil der Masse	ein Anteil der Masse

Bei einem expliziten numerischen Integrationsverfahren wäre also für einen Integrationsschritt wie folgt vorzugehen:

- Übertragen von Druck und Fluidmasse in das Strukturmodell
- Integration der Impulsgleichung in dem Strukturmodell
- Integration der Geschwindigkeit in der Struktur
- Rückübertragung von Geschwindigkeit und Verformung  
in das Fluidmodell.

Da wir aber davon ausgehen wollen, daß bei einer verschwindenden Strukturmasse das Verfahren in die für das Fluidmodell bewährte ICE-Technik / 2,3 / übergeht, muß die Integration der Impulsgleichung wenigstens teilweise implizit ablaufen. Der Grad an Implizität sei durch folgende Gleichungen charakterisiert.

$$\text{Beschleunigung} = f(\begin{matrix} {}^{n+1} & \text{Druckdifferenzverteilung,} \\ {}^n & \text{Verformung der Wand,} \\ {}^n & \text{Steifigkeitseigenschaften der Wand,} \\ {}^{n+1} & \text{effektiv bewegte Masse,} \\ {}^n & \text{Geschwindigkeit)} \end{matrix})$$

$${}^{n+1} \text{ Geschwindigkeit} = {}^n \text{ Geschwindigkeit} + \delta t \cdot \text{Beschleunigung}$$

$${}^{n+1} \text{ Verformung} = {}^n \text{ Verformung} + {}^{n+1} \text{ Geschwindigkeit} \cdot \delta t$$

Folgt man der Aufteilung gem. der in YAQUI benutzten Vorgehensweise in 3 Phasen, so ergibt sich ganz allgemein folgender Ablauf:

- Phase 1 enthält
- die Phase 1 der Fluiddynamik
  - Übergabe des Druckes an das Strukturmodell
  - Berechnung der Geschwindigkeit im Strukturmodell (1.Schätzung)
  - Rückübertragung der Geschwindigkeit an das Fluidmodell

Phase 2 (Iteration) enthält

- Berechnung eines Feldes von Druckkorrekturen und Massenkorrekturen im Fluid  
(Gln. 10-15)

- die Übertragung der Druckkorrekturen und Massekorrekturen an das Strukturmodell
- die Berechnung korrigierter Geschwindigkeiten im Strukturmodell
- die Rückübertragung der korrigierten Geschwindigkeiten in das Fluidmodell

Phase 3 enthält

- die Phase 3 der Fluiddynamik
- die Integration der Geschwindigkeiten im Strukturmodell (Verformung)
- die Übertragung der Verformung an das Fluidmodell

Man beachte, daß vor Beginn der Integration in der Vorbereitungsphase die Wandverformung und ihre zeitliche Ableitung im Strukturmodell definiert und in das Fluidmodell übertragen werden muß.

### 3.3 Ableitung der Gleichungen für die Wandbewegung

Die folgenden Ableitungen dienen nur als Grundlage für die Berechnung

einer effektiven mit der Wand bewegten Fluidmasse und der Vorschrift für die Druckiteration.

Sie dienen nicht zur Lösung der Impulsgleichung quer zur Wand selbst, da diese im Strukturmodell integriert wird. In Kapitel 3.1 und 3.2 wurde angenommen,  $H$  und  $\dot{H}$  seien vorgegeben. In Wirklichkeit hängen sie vom Druck ab. In Kap. 3.2 wurde gezeigt, daß die Impulsgleichung für die Wand im Strukturmodell gelöst werden muß. Im Fluidmodell müssen jedoch folgende Größen ermittelt werden:

- effektiver Anteil der Fluidmasse, der mit der Struktur bewegt wird,
- Korrektur des Druckfeldes.

Um diese Größe zu ermitteln, formulieren wir die Impulsgleichung für die Struktur und für einen geeigneten Ausschnitt aus dem Fluid.

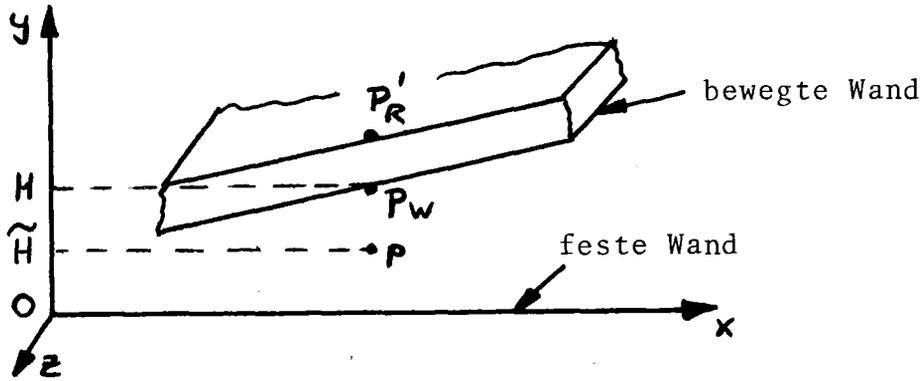


Abb. 3.3: Druck im Fluid und an der Wand

In obiger Abbildung sei

- $P_R'$  der Druck auf der Rückseite der bewegten Wand
- $P_W$  der Druck an der bewegten Wand
- $\tilde{H}$  die Höhe, in der der mittlere Flüssigkeitsdruck  $p$  herrscht

Wir setzen ferner

- $\mu$  Massenbelegung der Wand
- $w$  Geschwindigkeit in  $y$ -Richtung
- $u_n$  die aus der Masche gerichtete Normalgeschwindigkeit

und vernachlässigen Massenkräfte im Fluid. Dann gilt für die Wand

$$\int_F (\mu \ddot{H} - P_W + P_R' + \text{Struktureinfluß}) dF = 0 \quad (18)$$

bzw. für den Fluidbereich zwischen  $\tilde{H}$  und  $H$

$$\begin{aligned} & \frac{d}{dt} \int_F \int_{\tilde{H}}^H \rho w dy dF && \text{/x Trägheitskräfte x/} \\ & + \int_{\sigma} u_n \int_{\tilde{H}}^H \rho w dy d\sigma && \text{/x Impulstransport x/} \quad (19) \\ & - \int_F (p - P_W) dF && \text{/x Druck x/} \\ & - \int_F ([\rho w^2]_{\tilde{H}} - [\rho w^2]_H) dF = 0 && \text{/x Impulstransport x/} \end{aligned}$$

Wir nehmen an, die Felder  $u$ ,  $v$  und  $\rho$  seien über  $y$  konstant. Dann ist aus Gründen der Kontinuität  $w$  linear von  $y$  abhängig

$$w = \frac{\dot{H}}{H} \cdot y \quad (20)$$

Daraus folgt für die Druckverteilung in der Flüssigkeit über  $y$  näherungsweise gemäß

$$\frac{\partial p}{\partial y} + \rho \frac{\ddot{H}}{H} y = 0 \quad (21)$$

eine parabolische Verteilung für  $p$  mit  $\frac{\partial p}{\partial y} = 0$  an der festen Wand. Das bedeutet, daß der mittlere Druck  $\tilde{p}$  in der Höhe

$$\tilde{H} = \frac{1}{\sqrt{3}} \cdot H \quad (22)$$

auftritt. Damit erhalten wir aus (19)

$$\begin{aligned} & \frac{d}{dt} \int_F \frac{1}{3} \rho H \dot{H} dF \\ & + \int_{\sigma} u_n \frac{1}{3} \rho H \dot{H} d\sigma \\ & - \int_F (p - p_w) dF + \frac{2}{3} \int_F \rho \dot{H}^2 dF = 0 \end{aligned} \quad (23)$$

Wir benutzen nun die Beziehungen

$$\frac{d}{dt} (\rho H \dot{H}) = \rho H \ddot{H} + \dot{H} \frac{d}{dt} (\rho H) \quad (24)$$

$$\begin{aligned} \int_{\sigma} u_n \rho H \dot{H} d\sigma &= \int_F \operatorname{div} (\rho H \dot{H} \bar{u}) dF \\ &= \int_F [\dot{H} \operatorname{div} (\bar{u} \rho H) + \bar{u} \rho H \operatorname{grad} \dot{H}] dF \end{aligned} \quad (25)$$

und die Kontinuitätsgleichung (die für jedes beliebige F identisch erfüllt ist):

$$\int_F \left[ \frac{d}{dt} (\rho H) + \operatorname{div} (\bar{u} \rho H) \right] dF = 0 \quad (26)$$

Dann folgt daraus

$$\int_F \left( \frac{1}{3} \rho H \ddot{H} + \frac{1}{3} \rho H \bar{u} \operatorname{grad} \dot{H} - \rho + p_w + \frac{2}{3} \rho \dot{H}^2 \right) dF = 0 \quad (27)$$

Wir nehmen ferner an, daß sich das Fluid auf der Rückseite der Wand akustisch verhält. Dann ist bei einem Ruhedruck  $p_R$  und einer akustischen Impedanz  $\rho_R \cdot c_R$

$$p'_R = p_R + \rho_R \cdot c_R \cdot \dot{H} \quad (28)$$

Aus (18), (23) und (28) erhalten wir

$$\begin{aligned} & \int_F \left[ \left( \mu + \frac{1}{3} \rho H \right) \ddot{H} \right. \\ & \left. + p_R - p + \rho_R \cdot c_R \cdot \dot{H} + \frac{1}{3} \rho H \bar{u} \operatorname{grad} \dot{H} + \frac{2}{3} \rho \dot{H}^2 \right. \\ & \left. + \text{Struktureinfluß} \right] dF = 0 \end{aligned} \quad (29)$$

Wir stellen fest, daß diese Gleichung für das gekoppelte System in die Strukturgleichung ohne Fluid übergeht, wenn wir setzen

$$\text{a) eine effektive Masse} = \mu + \frac{1}{3} \rho H \quad (30)$$

In STRÜYA wird der Term  $\frac{1}{3} \rho \cdot H$  vorab einmal ermittelt und konstant gehalten.

b) eine effektive Druckdifferenz =

$$\begin{aligned}
 & P - P_R - \rho_R \cdot c_R \cdot \bar{H} - \frac{2}{3} \rho \bar{H}^2 \\
 & - \frac{1}{6} \rho \bar{H} \cdot |\bar{u}_x| \cdot \left[ (z_1 - z_3)(\dot{H}_4 - \dot{H}_2) + (z_2 - z_4)(\dot{H}_1 - \dot{H}_3) \right] / F \\
 & - \frac{1}{6} \rho \bar{H} \cdot |\bar{u}_z| \cdot \left[ (x_1 - x_3)(\dot{H}_4 - \dot{H}_2) + (x_2 - x_4)(\dot{H}_1 - \dot{H}_3) \right] / F
 \end{aligned} \tag{31}$$

Von diesem Ausdruck werden in STRUYA jedoch nur die ersten drei Terme  $(P - P_R - \rho_R c_R \bar{H})$  berücksichtigt,

Benutzen wir für die angegebenen Größen die Werte, die zum Zeitpunkt  ${}^n t$  vorliegen, so haben wir damit bereits die Vorschrift für Phase 1 der Integration.

Anschließend müssen wir das Druckfeld  $p$  iterativ verbessern.

Wir benutzen dazu wie in Kap.3.1 die Kontinuitätsgleichung (13):

$$S_q = \frac{1}{\delta t} ({}^{n+1} \rho - {}^n \rho) + {}^{n+1} \rho D + {}^{n+1} \rho \frac{{}^{n+1} \bar{H}}{{}^n \bar{H}} \tag{32}$$

Um alle  $S_{qi}$  auf Null zu bringen, müßte im Rahmen einer Newtonschen Näherung der Druck geändert werden um Werte  $\delta p_K$ , für die folgende Gleichung gilt:

$$\sum \left[ \frac{\partial S_i}{\partial p_K} \left( \frac{1}{\delta t} + D_i + \frac{\bar{H}_i}{\bar{H}_i} \right) + \rho_i \frac{\partial D_i}{\partial p_K} + \frac{\rho_i}{\bar{H}_i} \frac{\partial \bar{H}_i}{\partial p_K} \right] \delta p_K = - S_{qi} \tag{33}$$

Wir setzen

$$\frac{\partial \bar{H}_i}{\partial p_K} = \delta t \frac{\partial \bar{H}_i}{\partial p_K} = \frac{\delta t}{4} \cdot \sum_{\ell=1}^4 \frac{\partial \bar{H}_{i,\ell}}{\partial p_K} \tag{34}$$

Darin bedeutet  $i, \ell$  einen der vier zur Masche  $i$  gehörenden

Knoten. Entsprechend der für das Fluid gewählten Vorgehensweise / 2 / begnügen wir uns damit, die Koeffizienten der Gleichung (33) näherungsweise zu bestimmen, da wir ja ohnehin iterieren müssen. Wir vernachlässigen daher die Änderung der effektiv bewegten Masse während der Iteration und bestimmen den Ausdruck  $\frac{\partial \ddot{H}_{i,e}}{\partial p_k}$  nur einmal zu Beginn der Iteration. Wegen (17) ist

$$\frac{\partial \ddot{H}_{i,e}}{\partial p_k} = \sum_s T_{i,e,s} \cdot \frac{\partial \ddot{x}_s}{\partial p_k} \quad (35)$$

Diese Ausdrücke können im Strukturmodell berechnet werden. Für das anzuwendende Überrelaxationsverfahren benötigen wir allerdings nur die Terme

$$\frac{\partial \bar{\ddot{H}}_i}{\partial p_i} = \frac{1}{4} \sum_{e=1}^4 \frac{\partial \ddot{H}_{i,e}}{\partial p_i} \quad (36)$$

Dieser Vektor (von der Länge gleich der Anzahl der Fluidmaschen) muß vom Strukturmodell dem Fluidmodell mitgeteilt werden. Damit und mit der Vorgehensweise gemäß Kap. 3.1 erhalten wir Schätzwerte für  $\delta p_i$

$$\delta p_i \left[ \frac{1}{c_i^2} \left( \frac{1}{\delta t} + D_i + \frac{\bar{\ddot{H}}_i}{H_i} \right) + 2 \delta t \left( \frac{1}{\Delta r_i^2} + \frac{1}{\Delta z_i^2} \right) + \frac{S_i \delta t}{H_i} \cdot \frac{\partial \bar{\ddot{H}}_i}{\partial p_i} \right] = - S_{qi} \quad (37)$$

Obige Gleichung kann unter Einbeziehung eines Relaxationsfaktors  $\omega$  geschrieben werden als

$$\delta p_i = - \omega \delta t c_i^2 \frac{S_{qi}}{1 + K_{F,i} + K_{S,i}} \quad (38)$$

mit

$$K_{F,i} = \delta t \left( D_i + \frac{\bar{H}_i}{H_i} \right) + 2 c_i^2 \delta t^2 \left( \frac{1}{\Delta r_i^2} + \frac{1}{\Delta z_i^2} \right) \quad (39)$$

$$K_{S,i} = \frac{S_i c_i^2}{\bar{H}_i} \delta t^2 \left( \frac{\partial \bar{H}_i}{\partial \rho_i} \right) \quad (40)$$

Für kleine Werte  $K_{F,i}$  und  $K_{S,i}$  kann auch geschrieben werden

$$\delta \rho_i = - \omega \cdot \delta t \cdot c_i^2 \frac{S_{qi}}{(1 + K_{F,i})(1 + K_{S,i})} \quad (41)$$

Man kann also den Einfluß der Koppelung im Rahmen der Druckiteration als einen zusätzlichen Unterrelaxationsfaktor der Größe  $1/(1 + K_{S,i})$  auffassen.

Dieses Verfahren war für einfache Strukturmodelle erfolgreich. Bei der Koppelung des CYLDY2-Modells / 4 / für den HDR-Kernmantel / 5 / ergaben sich jedoch Stabilitätsprobleme (abgesehen von sehr kleinen Schrittweiten). Zwei Ursachen ließen sich feststellen: die Weichheit der Struktur und das Globalverhalten des Strukturmodells. Das CYLDY2-Modell benutzt zur Darstellung der Strukturverformung globale Formfunktionen. Daraus ergibt sich, daß eine lokale Druckvariation unmittelbar die gesamte Struktur beeinflusst. Besonders ausgeprägt ist dies in der Nähe starrer Teile der Struktur. Damit geht die Lokalität im physikalischen Verhalten und die ausgeprägte Bandstruktur der Gleichungsmatrix der Druckiteration verloren. Aus diesem Grunde kann in STRUYA die durch die Koppelung bewirkte Unterrelaxation in den Bereich des Strukturmodells verlagert werden. Dies geschieht folgendermaßen:

Schritt 1: Berechnung der Druckvariation nach (38), jedoch für starre Struktur ( $K_{s,i}=0$ )

Schritt 2: Transformation der Druckvariationen in den Raum der Formfunktionen  $X_{ni}$  der Struktur gemäß

$$\delta p_i = \sum \delta P_n \cdot X_{ni} + \delta p_{iR} \quad (42)$$

$\delta p_{iR}$  ist der verbleibende Teil, der durch die Formfunktionen nicht repräsentiert wird.

Schritt 3: Unterrelaxation gemäß

$$\delta \tilde{P}_n = \frac{\delta P_n}{1 + \tilde{K}_{s,n}} \quad (43)$$

Die Faktoren  $\tilde{K}_{s,n}$  werden vorab bestimmt (40).

Schritt 4: Ermittlung der lokalen  $\delta p_i$  nach Unterrelaxation

$$\delta \tilde{p}_i = \sum \delta \tilde{P}_n \cdot X_{ni} + \delta p_{iR} \quad (44)$$

Zur Vermeidung unnötigen Aufwandes wird die Unterrelaxation nur für  $\tilde{K}_{s,n} > 0.2$  durchgeführt. Die Druckiteration wird durchgeführt, solange

$$|\delta p_i| > \varepsilon \cdot \max(p)$$

$$|S_{qi} \cdot \delta t| > \varepsilon \cdot p_i$$

$$|D \cdot \delta t| > \frac{\alpha_0}{2(1 + \alpha_0^2)} + 10^{-10}$$

$$|\delta \bar{H} \cdot \delta t| > \max(\varepsilon \cdot \max(|\dot{H}|) \cdot \delta t, 10^{-10} \bar{H})$$

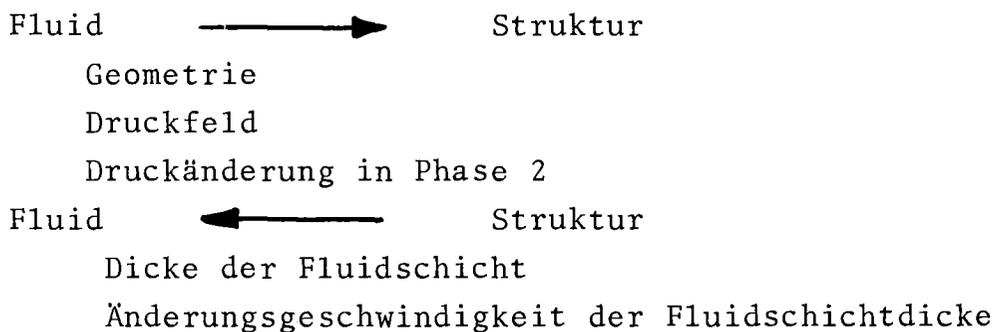
wobei  $\delta \bar{H}$  die Änderung der Wandgeschwindigkeit im letzten Iterationsschritt ist. Es ist jedoch eine Begrenzung auf

maximal 20 Iterationsschritte

festgelegt. Tritt diese Begrenzung auf, sind danach berechnete Problemwerte im allgemeinen fehlerhaft.

#### 4. Strukturkopplung

Die Kopplung eines Strukturprogrammpaketes erfolgt über den Aufruf eines Satzes von Routinen, die den notwendigen Datenaustausch zwischen dem Fluidbereich in STRUYA und dem Strukturbereich herstellen. Wichtige Kommunikationsdaten sind beispielsweise:



Die Steuerung des Strukturmodells und die Kommunikation werden von STRUYA durch Aufruf eines Satzes von StrukturROUTINEN bewerkstelligt. Für jedes neue Strukturmodell in STRUYA muß dieser Satz von Routinen (Module im REGENT-Sinne / 1 /) sowie die Initialisierung der Strukturmodelldatenstruktur neu erstellt werden.

Die Kommunikation der StrukturROUTINEN untereinander geschieht über einen Pointer im STRUYA Subsystem -COMMON(siehe / A3 /, der auf die Datenstruktur des Strukturmodells zeigt).

##### 4.1 KommunikationsROUTINEN

- TRXY (Transfer XY)  
vom Fluidgrid abhängige Strukturwerte werden initialisiert
- FSXO die Fluidschichtdicke H wird aus der Strukturverformung berechnet

- FSX1 Die Änderungsgeschwindigkeit der Fluidschichtdicke  $\dot{H}$  wird aus der Strukturverformungsgeschwindigkeit berechnet.
- INT1 Die in Phase 2 berechneten Strukturverformungsgeschwindigkeiten werden in die Verformung integriert.
- DHDP berechnet ( $\partial H / \partial p$ ), allgemein sind hier nach einem Restart einmalig zu berechnende fluidabhängige Strukturfunktionen zu setzen, z.B. im Kesselmodell wird aus der Fluidschichtdicke ein Massenzuschlag für die Struktur berechnet.
- P1 übergibt in Phase 1 das Fluiddruckfeld  $p$  an die Struktur.
- DXPT berechnet in Phase 2 die Änderung der Geschwindigkeit der Struktur durch Integration der Beschleunigungsänderung.
- SAVE schreibt Strukturwerte auf eine sequentielle Datei (Anlage 5.6)
- READ liest diese wieder (Anlage 5.8)
- PRNT für Druckausgabe
- DISP berechnet für graphische Ausgabe die Displacements des Kernmantels.
- FREE gibt die vom Struktursubsystem angelegte Datenstruktur frei.

Ähnlich wie in REGENT alle Subsystemmodule einen Prefix haben (bei STRUYA heißt er 'ST'), bekommen in STRUYA die Routinen, die einem Strukturmodell zugeordnet sind, einen zweiten Prefix.

Der Aufbau des Namens einer solchen Routine erfolgt also folgendermaßen:

Subsystemprefix || Strukturmodellprefix || Routinename

Das Einfügen des Modellprefix geschieht bei der Deklaration der StrukturROUTINEN mit Hilfe des Attributs ENTRYNAME automatisch. Z.B. kann eine Deklaration so aussehen:

```
DCL STRUCT_READ ENTRYNAME((STRU_MODEL
    ||READ)) ENTRY DYNAMIC;
```

Für den Strukturmodellprefix können beliebige 2-stellige Charakterstrings benutzt werden, mit Ausnahme der Kombination '\$\$'. Diese ist dem Fall vorbehalten, daß STRUYA für Rechnungen ohne Strukturkopplung benutzt wird. Hier wird eine Reihe von Dummy-Routinen benutzt mit eben diesem Prefix.

Den Zusammenhang zwischen den STRUYA-Fluidroutinen und den StrukturROUTINEN zeigen Anlagen 4.1 und 4.2. Anlage 4.1 zeigt ein Programmablaufschema für eine Restart-Save-Rechnung ohne Strukturkopplung. Anlage 4.2 zeigt eine Kopplung mit dem Strukturmodell CYLDY2. Hier gehören alle Routinen mit dem Prefix CD zum Struktur-subsystem.

Alle StrukturROUTINEN haben als Parameter den STATUS, das ist eine Struktur in STRUYA, die die Diskretisierung des Fluidmodells und die Zustandsgrößen des Fluids beschreibt. Außerdem einige Größen zur Übertragung der Strukturbewegung und Beschleunigung auf die Fluidschicht und der in Phase 2 vorausberechneten Fluiddruckänderung auf die Struktur. Die I/O-Programme haben einen zusätzlichen Parameter, der den Zustand des Rechenablaufes angibt, das heißt, ob Initialisierungsdaten transferiert werden oder ob es sich um Daten eines Rechenzyklus handelt (siehe A1).

#### 4.2 Die Initialisierung eines Struktursubsystems

Die Kopplung Fluid-Struktur erfolgt während der Ausführung einer STRUYA-Rechnung durch den Aufruf eines REGENT-Subsystems, bei dem die die Struktur beschreibenden Daten, das Strukturmodell, erzeugt werden.

Die Erzeugung eines solchen Subsystems erfolgt vor der Ausführung eines STRUYA-Rechenlaufs mit Hilfe der REGENT-Sprache PLS / 1 /. Ein Beispiel für die Generierung eines solchen Struktursubsystems zeigt Anlage 4.3. Wie man dort sehen kann, wird beim Subsysteminitialisieren folgendes festgelegt:

- Der Name für das Struktursubsystem (folgende Strukturmodelle wurden z.B. bisher erzeugt: KESSEL, BEAM, CYLDY)
- Eine Datenstruktur vom Typ BASED, die die für das Modell notwendigen Beschreibungsdaten zusammenfaßt.
- Ein Satz von Standardzuweisungen zum Füllen der Datenstruktur und der im STRUYA-COMMON enthaltenen Größen STRUDAT\_PTR und STRU\_MODEL. Diese Zuweisungen können auch später beim Aufruf des Subsystems geschrieben bzw. überschrieben werden.

#### 4.3 Datenaustausch zwischen den StrukturROUTINEN

Nachdem das Struktursubsystem initialisiert ist, kann es in STRUYA benutzt werden. Beim gezeigten Beispiel wird durch ein einfaches

```
ENTER KESSEL;
```

die INIT CLAUSE ausgeführt und dadurch die Datenstruktur für das KESSEL-Modell deklariert, allokiert und mit Standardwerten gefüllt. Ihre Adresse MODEL-PTR und ihr Prefix 'AA' werden im

STRUYA-DATA-Common abgespeichert, so daß in allen Moduls, in denen die Modellstruktur wieder deklariert ist, auf die Strukturgrößen zurückgegriffen werden kann. Damit ist die Möglichkeit, Daten zwischen den Strukturroutinen auszutauschen, gegeben. Der Datenaustausch zwischen Fluid und Struktur erfolgt also über den STRUYA-STATUS als Argument, zwischen Struktur und Struktur über den STRUYA-Common, in dem der Pointer zur Modellstruktur festgehalten ist. Das Verlassen des Struktursubsystems mit z.B.

END KESSEL;

bewirkt nicht mehr, als daß die STRUYA-Sprache wieder wirksam wird. Die Modelldatenstruktur wird beim END STRUYA freigegeben.

STRUYA PROZEDUREN

```

*****
#SOLV1**
  |--SET1****
  * |           *-----$$TRXY *
  * |           *-----$$FSX0 *
  * *****
  *
  |--QUI1****
  * |           *--READ   *---$$READ *
  * |           *--SAVE   *---$$SAVE *
  * |           *--READ   *---$$READ *
  * |           *--SAVE   *---$$SAVE *
  * |           *-----$$DHDP *
  * |           *--PRINT  *---$$PRNT *
  * |           *--SAVE   *---$$SAVE *
  * |           *
  * |           *-----$$P1   * <-|
  * |           *-----$$FSX1 *
  * |           *--UVTILQ *
  * |           *--DPINew**
  * |           *--PREPH2 *
  * |           *--PH2ITER*
  * |           * |           *---$$DXPT *
  * |           * *****---$$FSX1 *
  * |           *--ETILM  *
  * |           *--PHASE3**
  * |           * |           *---$$INT1 *
  * |           * |           *---$$FSX0 *
  * |           * *****---SETSTA *
  * |           *--PRINT  *---$$PRNT *
  * |           *--SAVE   *---$$SAVE *
  * |           *
  * |           *-----$$FREE * >-|
*****
MOFREE

```

Anlage 4.1: Programmablaufschaema für einen STRUYA-Restart-rechenlauf ohne Strukturkopplung



```

KESPOL: PROC OPTIONS(MAIN) REGENT(INIT,NODA,MPOOL=30000);
  ENTER PLS;
  INIT 'KESSEL' 'IRE6TEAM';
  SUBSYSTEM 'KESSEL' 'IRE6TEAM';
  CLAUSE INIT;
  EXEC;
    DCL 1 MODEL_AA BASED(MODEL_PTR),          /* KESSEL MODELL */
        2 FLUSTRU_DATA,
          3 RADSTRU BIN FLOAT(53) ,
          3 HSTRU   BIN FLOAT(53) ,
/*          3 CSTRU   BIN FLOAT(53) ,      */
          3 ROSTRU  BIN FLOAT(53) ,
          3 EMOD    BIN FLOAT(53) ,
        2 FLUID,
          3 #HNUL BIN FLOAT(53),
        2 FLURUECK,
          3 PRUECK  BIN FLOAT(53) ,
          3 RORUECK BIN FLOAT(53) ,
          3 CRUECK  BIN FLOAT(53) ,
        2 KES_CODE_NAME CHAR(2) ,
        2 XSNULL  BIN FLOAT(PREC),
        2 STRUSTAT,
          3 (XS,XSPT,XS2PT) BIN FLOAT(PREC),
        2 MASSZU BIN FLOAT(PREC);

/****** STANDARD ZUWEISUNGEN FUER KESSEL *****/
  ALLOCATE MODEL_AA ;
  STRU_DAT_PTR = MODEL_PTR;
  STRU_MODEL,KES_CODE_NAME = 'AA' ;
  RADSTRU = 2.66;
  HSTRU   = 23.E-3;
  ROSTRU  = 9.E3;
  EMOD    =2.E11;
  #HNUL   = .15;
  PRUECK  = PNULL;
  RORUECK = RONULL;
  CRUECK  = CNULL;
  XSNULL  = .15;
  XS      = XSNULL;
  XSPT    = 0.0;
  XS2PT   = 0.0;
  MASSZU  = 0.0;
/******
  END EXEC;
  END CLAUSE;
  END PLS;
  END KESPOL;

```

Anlage 4.3: Beispiel einer Struktursubsysteminitialisierung

## 5. Benutzerhandbuch

### 5.1 Aufruf von STRUYA

Zur Ausführung eines STRUYA-Jobs steht eine katalogisierte REGENT-Prozedur zur Verfügung:

```
// EXEC REGENT
```

Da die STRUYA-Module im Sinne von REGENT auf privaten Dateien / 1,S 5-31 / angelegt sind, muß dieser Prozedur Einheit und Name des Datenträgers sowie Name der STRUYA-Bibliothek mitgeteilt werden. Dies sind zur Zeit:

Einheit = 2314

Name der Einheit: GFK099

Name der Bibliothek: STRUYA

In der REGENT-Prozedur übernehmen die Parameter PUNIT, PVOL, PLIB diese Information. Damit lautet der Aufruf von STRUYA:

```
// EXEC REGENT,PUNIT=2314,PVOL=GFK099,PLIB=STRUYA
```

Die Eingabe des Programms erfolgt hinter //P.SYSIN DD \*

Die Prozedurkarte erhält die Option REGENT / 1,S 6-5 /.

Eine Prozedur, die in STRUYA z.B. Plotausgabe erzeugt, beginnt folgendermaßen:

```
BLOW:PROC OPTIONS(MAIN) REGENT(NODA,PLOT=STATOS,MPOOL=200000);
```

Der Eintritt in das Subsystem erfolgt mit

```
ENTER STRUYA CELLS(mn,nn);
```

Das Subsystem wird abgeschlossen durch

```
END STRUYA;
```

Die Kopplung von STRUYA mit einem Strukturmodell ist in Abschnitten 4 und 5.3 beschrieben.

Vor Beendigung des REGENT-Hauptprogramms sollten bzw. müssen die Anweisungen

```
MESSAGE ACTIVE DEBUG;  
FINISH;
```

im Programm eingefügt sein.

DEBUG erzeugt statistische Angaben über die Modulverwaltung, ausgegeben werden diese durch FINISH. Dieses FINISH ist unbedingt notwendig bei Verwendung von Plot-Software.

## 5.2 STRUYA-Syntax

Die STRUYA-Anwendersprache enthält folgende Befehle:

```
ANC  
BOTTOM      } { ACCELERATION  
RIGHT       } { STATUS  
TOP         } { VELOCITIES  
COPY  
CSQUARE  
END  
ENTER  
GRAVITY  
GRID  
  
INIT  
MEANHEIGHT  
MESH  
PLOT  
PRESSURE  
PRINT  
RESTART  
SAVE  
SOLVE  
STEP  
TITLE  
TYPE  
VISCOSITY
```

Ihre Syntax und ihre Bedeutung wird im folgenden beschrieben.

Syntax

ANC

ANC expr;

Erläuterung

Der Kopplungsfaktor (alternate node coupler, siehe / 2 /) stellt während der Berechnung der Fluidgeschwindigkeiten in Phase 1 eine Beziehung zu Nachbarmaschen her.

Der Standardwert ist 0., der Maximalwert 1. Er hat einen wesentlichen Einfluß auf auftretende Oszillationen / 6 /.

BOTTOM  
RIGHT  
TOP

Syntax

BOTTOM } { ACCELERATION }  
RIGHT } { STATUS } ;  
TOP } { VELOCITIES }

respekt.

END { BOTTOM } { ACCELERATION } ;  
{ RIGHT } { STATUS } ;  
{ TOP } { VELOCITIES }

Erläuterung:

1. Die Zustandsgrößen am Strukturrand werden mit dem beschriebenen Statement gesetzt, wenn das Rechenmodell von dem Standardfall (starre Berandung der Fluid-Region) abweicht.
2. Mit dem Attribut ACCELERATION können den Gitterpunkten am Rand Beschleunigungsterme aufgeprägt werden.  
Mit einem solchen Statement (z.B. BOTTOM ACCELERATION) wird der Kopf einer Subroutine erzeugt, in deren Argumentliste folgende Variablen stehen:  
I Numerierung des Randknotens bei den Routinen BOTTOM und TOP  
J Numerierung des Randknotens bei der Routine RIGHT

Ferner sind enthalten:

- U Geschwindigkeitskomponente U am Randknoten
- V Geschwindigkeitskomponente V am Randknoten
- X(I), Y(I) X- und Y-Koordinaten des Randes bei den Routinen BOTTOM und TOP

X(J), Y(J) X- und Y-Koordinaten des Randes bei der Routine RIGHT  
RM Reziproker Wert der Masse des Randknotens.

Diese Variablen werden deklariert und können dazu dienen, der  
Größe

ACC=Beschleunigungswert am Randknoten

einen Wert zuzuweisen.

Alle anderen hier aufgezählten Variablen dürfen keine neuen  
Werte erhalten.

3. Mit dem Attribut STATUS kann der Fall des kontinuierlichen  
Ein- bzw. Ausströmens von Flüssigkeit in die/aus der Fluid-  
Region simuliert werden. Dabei können die Randmaschen als  
Quellen bzw. Senken für die spez. innere Energie und die Masse  
(Dichte) angesehen werden.

Bei Aufruf des Statements 'boundary STATUS' werden folgende  
Deklarationen geliefert:

I Numerierung der Randmasche bei den Routinen BOTTOM und TOP.  
J Numerierung der Randmasche bei der Routine RIGHT.

In diesen Randmaschen können die deklarierten Größen

RO Dichte in der Randmasche

SIE spezifische innere Energie in der Randmasche

z.B. in Abhängigkeit von der Zeit festgelegt werden.

4. Das Attribut VELOCITIES dient zur Neufestlegung der Geschwin-  
digkeiten am Rand. In der generierten Subroutine stehen folgende  
Deklarationen:

I Numerierung des Randknotens bei den Routinen BOTTOM und TOP  
J Numerierung des Randknotens bei der Routine RIGHT

Ferner sind alle Koordinaten der Randknoten durch

X(I), Y(I) bei den Routinen BOTTOM und TOP bzw.

X(J), Y(J) bei der Routine RIGHT

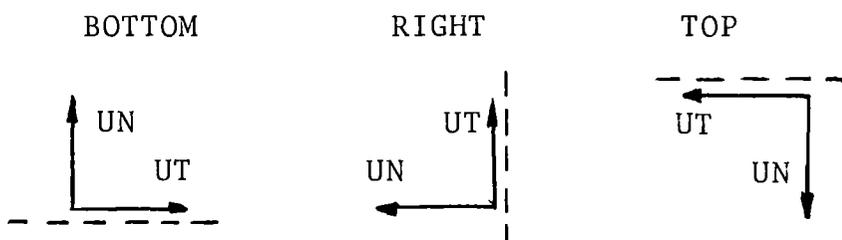
als Zahlenfelder deklariert.

Die Größen, die in dieser Routine festgelegt werden, sind

UN Geschwindigkeitskomponente in Normalrichtung  
am Randknoten

UT Geschwindigkeitskomponente in Tangentialrichtung  
am Randknoten

Positive Richtungen von UN, UT an den Rändern:



5. Die drei aufgeführten Unterprogrammtypen werden für jeden Randknoten bzw. jede Randmasche aufgerufen. Dabei werden jeweils die Numerierung, Koordinaten usw. in das Unterprogramm übergeben.
6. Jede Eingabe-Routine schließt mit dem entsprechenden END-Statement

END boundary typ;

ab.

Beispiele:

1. TOP ACCELERATION;  
ACC=-1.5\*I\*RM;  
END TOP ACC;

```
2.  RIGHT STATUS;
    IF J>20& J<25 THEN DO;
      RO = 1.0;
      SIE = 0.5;
    END;
    END RIGHT STATUS ;

3.  BOTTOM VELOCITIES;
    DCL PI BIN FLOAT(21) INIT(3.14159);
    DCL N BIN FIXED(15);
    N = 15;
    UT = 0.0;
    UN = 2.0* $\text{SIN}((I-1)/(N-1)*\text{PI})$ ;
    END BOTTOM VELOCI;
```

### Syntax

```
COPY file1 TO file2 [EVERY n STEP]
      UNTIL { CYCLE ncyc
             [TIME tim [MSEC]] } [ {and} PRINT ] ;
```

### Erläuterung

Mit COPY kann ein STRUYA-Savefile ganz oder teilweise kopiert werden. COPY ist vorgesehen zur Reparatur von nicht ordnungsgemäß abgeschlossenen Files (z.B. Savejob Abend wegen Zeitüberschreitung) und zur Verkleinerung von Dateien durch Aussortierung von Werten. File1 und file2 müssen physikalisch verschiedene Files sein.

EVERY n STEP: es wird nur jeder n-te Zyklus kopiert

CYCLE : erwartet eine Angabe über den letzten Zyklus bis zu dem kopiert wird

TIME : erwartet eine Zeitangabe in Sekunden (Problemzeit). Wird MSEC angegeben, wird der Wert programmintern umgerechnet

PRINT : Der kopierte File wird gelesen und jeder gelesene Zyklus durch Druckausgabe dokumentiert (5 Linien/Zyklus)

### Beispiele

```
DCL (INFIL, OUTFIL) FILE;
COPY INFIL TO OUTFIL UNTIL CYCLE 182;
COPY INFIL TO OUTFIL EVERY 2 STEP UNTIL CYCLE 2000;
```

Siehe auch Anlage 5.5

Syntax

CSQUARE expr;

Erläuterung

1. expr ist entweder ein Ausdruck der das Quadrat der adiabatischen Schallgeschwindigkeit c angibt

$$c^2 = \left( \frac{\partial p}{\partial \rho} \right)_s$$

oder der Name einer Funktion einschließlich den Argumenten RO, SIE, in der dieses Quadrat berechnet wird.

2. In die Funktionsroutine, die durch dieses Statement erzeugt wird, werden in der Argumentliste die Zustandsgrößen SIE und RO übergeben und innerhalb der Routine deklariert.
3. Dieses Statement muß in jedem Datenerzeugungslauf erscheinen, andernfalls erfolgt eine Fehlermeldung.

Beispiele

1. CSQUARE 2./3.\*SIE;
2. CSQUARE SIE\*(RO-1.);
3. CSQUARE CQUAD(RO,SIE); Dieses Beispiel wird übersetzt in:

```
CSQUARE:PROC(SIE,RO) RETURNS(BIN FLOAT(53));
```

```
DCL(SIE,RO,CSQ)BIN FLOAT(PREC);
```

```
CSQ=CQUAD(RO,SIE);
```

```
RETURN(CSQ);
```

```
END CSQUARE;
```

```
STRUYA,CSQUARE=CSQUARE;
```

Berechnet wird hier in der vom Benutzer definierten Funktion CQUAD.

Syntax

```
END { HEIGHT
    STATUS
    VELOCITIES
    MESH
    GRID
    BOTTOM } { ACCELERATION }
    RIGHT } { STATUS } ;
    TOP } { VELOCITIES }
```

Erläuterung

END dient neben seiner Verwendung zum Verlassen des Subsystems (siehe ENTER) auch zum Abschluß von mit Hilfe der STRUYA-POL zur Beschreibung des physikalischen Modells erzeugten internen Routinen (Anlage 2).

Beispiel

```
MESH;
  ⋮
END MESH;

RIGHT STATUS;
  ⋮
END RIGHT STATUS;
```

ENTER

STRUYA

### Syntax

```
ENTER STRUYA  [CELLS (ibar, jbar)]  
              [VERSION (version-nr)] ;  
              - - - STRUYA-Programm - - -  
END STRUYA;
```

### Erläuterung

1. Das ENTER-Statement bewirkt den Aufruf des REGENT-Subsystems, in diesem Fall den Aufruf des Subsystems STRUYA. Es muß am Anfang eines jeden Anwendungsprogrammes stehen.
2. Das Attribut CELLS gibt die Anzahl der Maschen der Fluid-Region längs der x- bzw. r-Achse (ibar) sowie längs der y- bzw. z-Achse (jbar) an. Die für IBAR und JBAR initialisierten Werte betragen 20 bzw. 32.
3. Mit dem Attribut VERSION wird die gewünschte Version des aufgerufenen Subsystems angegeben. Der Default-Wert ist die Version Nr.1.
4. Die Attribute können in beliebiger Reihenfolge hinter dem Aufruf ENTER STRUYA erscheinen.
5. Durch END STRUYA wird das Subsystem wieder verlassen.

### Beispiele

1. ENTER STRUYA VERSION(2) CELL(10,20);
2. ENTER STRUYA;(bei RESTART)

Syntax

$$\text{GRAVITY} \left[ \left[ \begin{array}{l} \underline{\text{X}}\text{RICHTUNG} \\ \underline{\text{R}}\text{ADIAL} \end{array} \right] = \text{gr}, \right. \\ \left. \left[ \left[ \begin{array}{l} \underline{\text{Y}}\text{RICHTUNG} \\ \underline{\text{Z}}\text{RICHTUNG} \end{array} \right] = \text{gz}, \right] \right. ;$$

Erläuterung

1. Die Berücksichtigung von globalen Beschleunigungen der Fluid-Region wird mit dem GRAVITY-Statement ausgedrückt.
2. Die Gravitationen können für x- (oder r-) Richtung, für y- (oder z-) Richtung angegeben werden.
3. Standardmäßig sind alle Beschleunigungen mit Null initialisiert.

Beispiel

GRA X = 0.5, Z = -9.81;

Syntax

GRID  $\left\{ \begin{array}{l} \underline{\text{EULER}} \\ \underline{\text{LAGRANGE}} \end{array} \right\};$

oder

GRID VELOCITY;

Setzen der Anfangswerte für U(I,J), V(I,J);

END GRID;

Erläuterung

1. Das GRID-Statement gibt an, ob es sich um ein Euler-Gitter, das räumlich festgelegt ist, oder ob es sich um ein Lagrange-Gitter, das mit der Strömung mitbewegt wird, handelt.
2. Standardmäßig wird das Gitter festgehalten (EULER).
3. Weicht die Gitterbewegung von den beiden genannten Fällen ab, so kann zwischen dem Statement GRID VELOCITY;  
und  
END GRID;  
die Vorschrift programmiert werden, nach der sich das Gitter bewegen soll.
4. Bei Strukturkopplung ist nur EULER erlaubt.

5. In dem von dem Statement

GRID VELOCITY

erzeugten Subroutinen-Kopf stehen folgende Variablen, so daß sie in der Routine benutzt werden können:

IP1 = Maschenzahl in X/R-Richtung  
JP2 = Maschenzahl in Y/Z-Richtung  
U(I,J) = Geschwindigkeitskomponenten in X/R-Richtung  
V(I,J) = Geschwindigkeitskomponenten in Y/Z-Richtung

Diesen Variablen dürfen keine neuen Werte zugewiesen werden.

Als Schleifen-Parameter kann der Anwender

I      und  
J

benutzen, die ebenfalls deklariert werden. Die genannten Variablen können benutzt werden, um allen Knotenpunkten in jedem Zyklus neue Zahlenwerte für die Koordinaten

X(I,J) = X-Koordinaten der Knotenpunkte  
Y(I,J) = Y-Koordinaten der Knotenpunkte

zuzuweisen.

Beispiele:

1. GRID EUL;
2. GRID LAGRA;
3. GRID VELOCITY;  
DO J=2 TO JP2;  
DO I=1 TO IP1;  
U(I,J)=X(I,J)\*X(I-1)\*0.2;  
V(I,J)=Y(I,J)\*X(J-2)\*0.1;  
END;  
END;  
END GRID;

Syntax

```
INIT STATUS;  
:  
END STATUS;
```

Erläuterung

1. In der Eingaberoutine INIT STATUS werden für alle Maschen der Fluid-Region die Dichte  $\rho$  und die spezifische innere Energie I für den Anfangszustand definiert.
2. Durch das Statement INIT STATUS wird eine PL/1-Subroutine generiert, in der folgende Deklarationen stehen:
  - X - X/R-Koordinate der Maschenmitte
  - Y - Y/Z-Koordinate der Maschenmitte
  - RO - Dichte  $\rho$  des Fluids in der Masche
  - SIE - spezifische innere Energie I des Fluids  
in der Masche
3. Die Routine wird für jede Masche aufgerufen. In der Argumentliste werden durch X und Y die Koordinaten der Maschenmitte übergeben, so daß diese für die Berechnung benutzt werden können.
4. Die Eingabe-Routine
  - INIT STATUS; schließt mit
  - END STATUS; ab.
5. In jedem Anwendungsbeispiel (nicht bei RESTART) muß eine Routine INIT STATUS stehen, in der Dichte und/oder Energie neu definiert werden. Denn andernfalls enthalten diese Zahlenfelder nur die Werte Null. Ein solches Beispiel ist sinnlos, da keine zeitabhängigen Zustandsänderungen auftreten.

INIT  
STATUS

Beispiele:

```
1.  INIT STATUS;  
    RO = 1.0;  
    SIE = 0.5;  
    END STATUS;
```

```
2.  INIT STATUS;  
    IF X > 5. THEN SIE = 2.;  
        ELSE SIE = 1.;  
    RO = 0.9x Y;  
    END STATUS;
```

INIT  
VELOCITIES

Syntax

```
INIT VELOCITIES;  
  :  
END   VELOCITIES;
```

Erläuterung

1. Mit der Eingabe-Routine INIT VELOCITIES werden für alle Knoten des Fluidgitters die Anfangsgeschwindigkeiten mit den Komponenten u und v definiert.
2. Durch das Statement INIT VELOCITIES wird eine PL/1-Subroutine generiert, in der folgende Deklarationen stehen:
  - X - X/R-Koordinate des Knotens
  - Y - Y/Z-Koordinate des Knotens
  - U - Geschwindigkeitskomponente u
  - V - Geschwindigkeitskomponente v
3. Die Subroutine wird für jeden Gitterknoten aufgerufen. In der Argumentenliste werden X und Y übertragen und können für die Festlegung von U und V benutzt werden.
4. Den Größen X und Y dürfen keine neuen Werte zugewiesen werden.
5. Die Eingabe-Routine

```
INIT VELOCITIES;   schließt mit  
END VELOCITIES;   ab.
```
6. Wird keine Routine INIT VELOCITIES geschrieben, so wird bei der Initialisierung des Anfangszustandes eine Standard-Routine aufgerufen, die das ganze Geschwindigkeitsfeld zu Null setzt.

INIT  
VELOCITIES

Beispiele:

1. INIT VELOCITIES;

U = 0.0;

V = 1.0;

END VELOC;

2. INIT VELOC;

U = (10.-X) $\times\times$ 2;

V = (10.-Y) $\times\times$ 2;

IF Y > 10 THEN V = -V;

END VELOC;

## MEANHEIGHT

### Syntax

MEANHEIGHT [OF FLUID] expr  $\left[ \begin{array}{c} \\ \text{MM} \end{array} \right]$  ;

### Erläuterung

MEANHEIGHT dient bei Erzeugung graphischer Ausgabe als Referenzhöhe der Fluidschicht. An die Plotroutinen wird es über die STRUYA-Datenstruktur in der Variablen #DTC(8) weitergegeben. Wird es nicht angegeben, dann wird die Fluidschichtdicke bei YMAX (in SET) als Referenzhöhe festgelegt.

Benutzt wird MEANHEIGHT in den Routinen KHIS, KREL, KNIV, §§DISP. Bei Kopplung mit einem Strukturmodell hat die Referenzhöhe keinen Einfluß auf die perspektive Darstellung des Modells.

### Beispiele:

MEANHEIGHT OF FLUID 0.15;

MEAN 150.MM;

Syntax

```
MESH { [RECTANGULAR (dr,dz)] } ;  
      { [HEIGHT] }  
END { MESH } ;  
    { HEIGHT }
```

Erläuterung

1. Mit dem Eingabe-Statement MESH werden die Maschenweiten des Maschengitters angegeben. Dabei ist dr die Maschenweite in x- oder r-Richtung, dz diejenige in y- oder z-Richtung.
2. Soll eine rechtwinkelige Fluid-Region mit konstanter Maschenbreite definiert werden, so genügt folgende Eingabe:

```
MESH RECTANGULAR(dr,dz);
```

3. Hat die Fluid-Region eine davon abweichende Form, so muß zwischen den Statements

```
MESH; und END MESH;
```

die Zuordnung der Knotenkoordinaten X und Y erfolgen.

4. Durch das Statement MESH wird die Subroutine INIT\_MESH erzeugt, die folgende Deklarationen bereits enthält:

```
IP1 - Knotenzahl in X/R-Richtung
```

```
JP2 - Knotenzahl in Y/Z-Richtung
```

Diesen Variablen werden ihre richtigen Zahlenwerte zugeordnet.

```
I - Schleifen-Parameter für X/R-Richtung
```

```
J - Schleifen-Parameter für Y/Z-Richtung
```

I muß alle Werte zwischen 1 und IP1 durchlaufen und J alle Werte zwischen 2 und JP2. Mit Hilfe dieser Integer-Variablen und möglicherweise mit Hilfe von Variablen, die der Benutzer selbst definiert, können die Knotenkoordinaten angegeben werden.

X(I,J) - X- oder R-Koordinaten  
 Y(I,J) - Y- oder Z-Koordinaten.

5. Durch das Statement MESH HEIGHT wird eine Subroutine INIT\_HEIGHT generiert, die es erlaubt, die Ringraumdicke an den Knotenpunkten festzulegen. Sie hat folgenden Kopf:

```
INIT_HEIGHT:PROC(X,Y,R);
DCL (X,Y,R) (x,x) BIN FLOAT(53);
```

MESH HEIGHT wird durch END HEIGHT abgeschlossen;

Beispiele:

1. MESH RECTANGULAR(2.,2.);
2. MESH RECT(1./3.,2./3.);
3. MESH;
 

```
DCL (RNULL,DR,ALFA,DALFA)
      BIN FLOAT(53);
DCL PI BIN FLOAT(53) INIT(3.14159);
DR = 1.0;
RNULL = 10.0;
ALFA = 0.0;
DALFA = PI/20.0;
DO J=2 TO JP2;
DO I=1 TO IP1;
X(I,J)=(DRx(I-1)+RNULL)xCOS(ALFA);
Y(I,J)=(DRx(I-1)+RNULL)xSIN(ALFA);
END;
ALFA = ALFA + DALFA;
END;
END MESH;
```

MESH

```
4. MESH HEIGHT;
   DCL (XMAX,DELTA,FJ) BIN FLOAT(53);
   XMAX = X(IP1,2);
   DELTA = 0.001;
       R = 0.015;
   DO J = 2 TO JP2;
       DO I = 1 TO IP1;
           FJ = (JP2-J+2)/JP2;
           R(I,J) = R(I,J) + COS(PI*X(I,J)/XMAX)*DELTA*FJ;
           PUT DATA(I,J,R(I,J));
       END;
   END;
END;
END HEIGHT;
```

Syntax

$$\text{PLOT } \left\{ \begin{array}{l} \text{OPEN} \\ \text{FILM} \end{array} \right\} \left\{ \text{figure } [\text{param}]^x \right\}^{10}, \text{ DRAW CLOSE};$$

Erläuterung

param:=

<u>ORIGIN</u>	expr	expr	
<u>PARALLEL</u>	expr	expr	expr
<u>PI_NORMAL</u>	expr	expr	expr
<u>FRAME</u>	expr	expr	
<u>NOFRAME</u>			
<u>SCALE</u>	expr		
<u>NOTEXT</u>			
<u>INTERPOL</u>	expr		
<u>WHEN</u>	expr	(cycle time)	
<u>FILE</u>	filename		

figure:=

- MESH
- VELOCITIES
- PNIVEAU
- PRELIEF
- PHISTORY, KHISTORY
- KGEOMETRY
- KNIVEAU
- KRELIEF
- CLOCK

Plotbeispiele in Anlagen 5.2 bis 5.4

PLOT kann sowohl als Kompaktstatement in der angegebenen Form verwendet werden, als auch als mehrfach aufgeteilter und durch Semicolon getrennter Befehl. Nach jedem Semicolon wird der PLOT-Befehl weitergeführt durch die Angabe PLOT, z.B.

## PLOT

```
PLOT OPEN MESH;  
PLOT FRAME 30. 40.;  
PLOT DRAW CLOSE;
```

OPEN, FILM initialisieren die graphische Ausgabe (mit ENTER GIPSY / 7 / im Modul). Durch OPEN wird Zeichnungsausgabe auf Papier, durch FILM Ausgabe im AGF-Plotfileformat / 8 / vorbereitet. Für den AGF-Plotfile stehen Programme zur Verfügung, zur Darstellung seines Inhalts auf Tektronix Bildsichtgeräten, oder für seine Weiterverarbeitung zu Filmen.

DRAW bewirkt die Ausführung des Plotbefehls durch einen PL1-Call auf die Plotmodule.

CLOSE beendet die Nutzung von GIPSY und schließt damit den Plotbefehl ab.

Die Parameter FRAME, NOFRAME, SCALE, PARALLEL, PI\_NORMAL, NOTEXT dienen zur Aufbereitung der Bilddarstellung. Werden sie nicht angegeben, hat jeder Bildtyp seine Standardwerte.

FRAME legt die Größe des Bildes in mm fest und zeichnet einen Rahmen mit diesen Abmessungen.

NOFRAME unterdrückt die Zeichnung eines Rahmens (z.B. mehrere Bilder in einem Rahmen)

SCALE gibt den Zeichnungsmaßstab an, zur Umrechnung der Problemkoordinaten in Zeichnungsgrößen.

ORIGIN fixiert den geometrischen Nullpunkt auf der Zeichnungsfläche.

PARALLEL, PI\_NORMAL sind Projektionsparameter. Ihre Bedeutung ist dieselbe wie im CHANGE-statement von GIPSY / 7 /.

## PLOT

NOTEXT unterdrückt das Schreiben eines Textes unter der Abbildung. Wird NOTEXT nicht angegeben, dann werden 3 Textlinien geschrieben:

1. Bildart (z.B.PREL),
2. der mit TITLE im vorausgegangenen STRUYA-Job erzeugte Text,
3. Problemzeit und Zyklusnummer.

INTERPOL dient bei der Filmerstellung zum Interpolieren zweier Momentaufnahmen.

WHEN legt die Bedingung für die Plotzeitpunkte fest.

FILE wird zusammen mit den Historydiagrammen benötigt. Es bezeichnet den Filenamem, auf dem die Meßorte definiert sind.

Die Bilddarstellungen werden im folgenden einzeln beschrieben. Es können bis zu 10 Bildanforderungen in einen Plotbefehl gepackt werden, egal ob diese alle in einen Rahmen oder getrennt gezeichnet werden. Jedem Bildtyp ist eine PL1-Struktur zugeordnet, die die Standardzeichnungsbeschreibung enthält. Sie ist für alle Typen gleich aufgebaut, für manche nicht voll ausgenutzt. Geändert werden die Standardwerte durch die weiter vorn beschriebenen Parameter.

PLOT  
CLOCK

Standardwerte:

```
DCL 1 PLOPCLOC,  
  2 PLOAX(2) DEC FLOAT(6) INIT(210.,150.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(90.,90.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 PROPIN(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(1.),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(10),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

```
PLOT OPEN CLOCK SCALE 0.5 DRAW CLOSE;
```

Erläuterung

Es wird eine Uhr gezeichnet, welche digital die Problemzeit in sec und analog in msec anzeigt.

Der Bereich der Digitalanzeige geht von  $10^{-4}$  sec bis 100 sec. Der Analogzeiger rotiert in 10 msec einmal um 360 Grad. Der Radius der Uhr beträgt standardmäßig 5 cm und kann mit SCALE verändert werden. Die Uhr ist in GIPSY ein 2D-Objekt, es ist deshalb nicht sinnvoll, die Projektionen zu verändern.

Standardwerte:

```
DCL 1 PLOPKGEO,
  2 PLOAX(2) DEC FLOAT(6) INIT(200.,250.),
  2 PRORIG(2) DEC FLOAT(6) INIT(80.,70.),
  2 PROPAR(3) DEC FLOAT(6) INIT(0.,1.,0.6),
  2 PROPIN(3) DEC FLOAT(6) INIT(0.,1.,0.),
  2 XYZAX(3) DEC FLOAT(6) INIT(2.,0.,0.),
  2 SCALE DEC FLOAT(6) INIT(0.02),
  2 HIFI FILE VARIABLE,
  2 PLTYP BIN FIXED(15) INIT(6),
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

```
PLOT FILM KGEO SCALE 0.01 NOFRAME DRAW CLOSE;
```

Erläuterung

KGEO erzeugt eine geometrische Darstellung der Struktur mit stark vergrößerter Darstellung eventuell vorhandener Abweichungen von einem mittleren Radius.

Die in STRUYA benutzte Abwicklung einer Zylindergeometrie wird wieder in einen Zylinder umgerechnet. Der Radius wird aus dem Minimal- und Maximalwert der x-Werte berechnet (Angenommen ist eine 180° Abwicklung). Diesem überlagert werden an den Netzknoten, bei Rechnungen mit CYLDY, die von CYLDY berechneten Auslenkungen u, v, w, sonst die Abweichung der Ringraumdicke von einem mittleren Wert (siehe MEANHEIGHT). Die realen Displacements sind mit dem Faktor 5 verstärkt. Wird der Standardwert der Skalierung von 0.02 benutzt, ergibt sich insgesamt eine relative Vergrößerung von

$$5/0.02 = 250.$$

PLOT

KHISTORY

Standardwerte:

```
DCL 1 PLOPKHIS,  
  2 PLOAX(2) DEC FLOAT(6) INIT(200.,250.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(0.,0.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 PROPIN(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(1.),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(9),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

```
PLOT OPEN FILE=MSTEL DRAW CLOSE;
```

Erläuterung

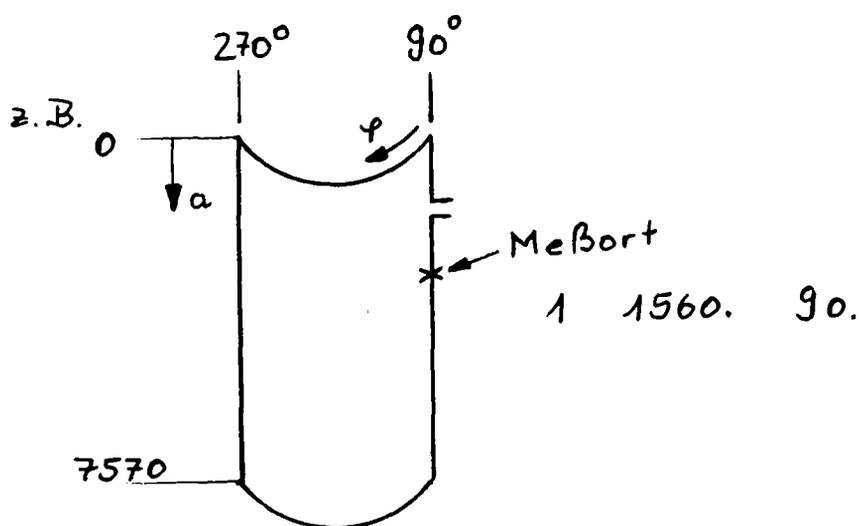
1. Dieser Befehl ist speziell für Anwendungen mit dem Strukturmodell CYLDY2 gedacht und nicht ohne weiteres verallgemeinerbar. Mit KHIS kann der zeitliche Verlauf der Auslenkung der Struktur an vorgegebenen Punkten (Meßstellen) aufgezeichnet werden.
2. Die Lage der Meßstellen wird in einem Dataset angegeben, der im obigen Beispiel den DD-Namen MSTEL hat. Wird kein File angegeben, ist der Standardname: KFIL. Die Deklaration wird generiert. Angaben für den Meßort sind folgende:

Nr, Winkel, Abstand

Der Winkel geht von  $90^{\circ}$  bis  $270^{\circ}$  ausgehend von der Stutzen-  
seite. Der Abstand wird von oben nach unten gemessen und in  
Millimeter angegeben.

PLOT

KHIS



3. Die Ordinatenachse reicht für den Standardfall: SCALE=1 von -3mm bis +3mm. Die Extremwerte können mit SCALE verändert werden. SCALE ist der einzige Parameter, der bei KHIS einen Einfluß auf die Plotdarstellung hat. Die Länge der Achse ist festgelegt. Die Skalierung der Abszisse erfolgt automatisch über die Feststellung des aufzuzeichnenden Zeitbereiches.
4. Maximal darf eine Kurve aus 1000 Punkten bestehen. Enthält ein Plotfile mehr Zeitschritte, kann die WHEN-Option benutzt werden; z.B. WHEN(MOD(CYCLE,n)=0 & CYCLE < m). Zeitverläufe können auch gezeichnet werden, wenn der Datenfile nicht ordnungsgemäß abgeschlossen ist.
5. Werden mehrere Kurven in ein Diagramm gezeichnet, dann sind diese durch verschiedene Symbole in unterschiedlichem Abstand markiert. Die Zuordnung der Symbole zu den Meßpunkten erfolgt über Textangabe.
6. Wird mit CYLDY gerechnet, dann werden die von CYLDY berechneten Displacements W aufgezeichnet, sonst die Abweichung von einem Mittelwert (siehe MEANHEIGHT).

PLOT  
KNIVEAU

Standardwerte

```
DCL 1 PLOPKNIV,  
  2 PLOAX(2) DEC FLOAT(6) INIT(210.,150.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(32.,130.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(0.,0.,1.),  
  2 PROPIN(3) DEC FLOAT(6) INIT(0.,0.,1.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(0.02),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(8),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

```
PLOT OPEN KNIV DRAW CLOSE;
```

Erläuterung

KNIV erzeugt eine Höhenliniendarstellung der Fluidschichtdicke, bezogen auf die Abweichung der Variablen R von einem mittleren Wert (siehe MEANHEIGHT). Die Höhenlinien erfassen einen Bereich von -3mm bis +3 mm mit Intervallen von 0.5 mm. Die Maschenknoten werden durch Symbole gekennzeichnet:

	Symbol-Nr.
$\Delta R > 0,25 \text{ mm}$	4
$-0.25\text{mm} \leq \Delta R \leq 0,25 \text{ mm}$	2
$\Delta R < -0,25 \text{ mm}$	1

Die Zuordnung der Symbolarten zu den Symbolnummern ist geräteabhängig.

Eine Veränderung der Projektionsparameter würde die Zuordnung der Höhenlinien zu den Knoten und zum Rahmen aufheben.

PLOT

KRELIEF

### Standardwerte

```
DCL 1 PLOPKREL,  
  2 PLOAX(2) DEC FLOAT(6) INIT(240.,250.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(80.,80.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(1.,1.,0.6),  
  2 PROPIN(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(0.02),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(7),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

### Beispiel:

```
PLOT OPEN KREL DRAW CLOSE;
```

### Erläuterung

Mit KRELIEF wird eine 3D-Reliefdarstellung der Fluidschichtdicke erzeugt. In Verbindung mit einem Struktursystem werden von diesem Abweichungen von einem Mittelwert in radialer Richtung berechnet, sonst werden die Abweichungen des Wertefeldes R von einem vorgegebenen Mittelwert (siehe MEANHEIGHT) dargestellt.

Mit SCALE können die X und Y Werte skaliert werden. Der Maßstab für R liegt fest mit

2 mm Abweichung von der mittleren Höhe pro 1 cm  
Zeichnungsstrecke

PLOT

MESH

Standardwerte:

```
DCL 1 PLOPMESH,  
  2 PLOAX(2) DEC FLOAT(6) INIT(200.,250.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(50.,50.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 PROPIN(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(0.02),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(1),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

```
PLOT OPEN MESH DRAW CLOSE;
```

Erläuterung

MESH zeigt die Aufteilung des Maschengitters in der X-Y-Ebene.

PLOT  
PHISTORY

Standardwerte:

```
DCL 1 PLOPPHIS,  
  2 PLOAX(2) DEC FLOAT(6) INIT(200.,250.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(0.,0.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 PROPIN(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(0.02),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(5),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

```
PLOT OPEN FILE=MSTEL DRAW CLOSE;
```

Erläuterung:

Mit PHIS kann der zeitliche Verlauf des Druckes im Fluid an vorgegebenen Punkten (Meßstellen) aufgezeichnet werden.

1. Die Lage der Meßstellen wird in einem Dataset angegeben, der im obigen Beispiel den Namen MSTEL hat (DD-Name). Wird kein File angegeben, ist der Standardname PFIL. Die File-declaration wird generiert. Angaben für einen Meßort sind folgende:

Nr, Winkel, Abstand.

Der Winkel geht von  $90^{\circ}$  am Bruchstutzen aus bis  $270^{\circ}$ . Der Abstand wird von oben nach unten gemessen und in Millimeter angegeben. (Siehe Abbildung bei KHISTORY).

PLOT  
PHISTORY

2. Die Länge der Achsen ist festgelegt. Die Ordinate erfaßt einen Druckbereich von 60 bis 140 bar. Die Skalierung der Abszisse erfolgt automatisch über die Feststellung des aufzuzeichnenden Zeitbereiches.
3. Maximal darf eine Kurve aus 1000 Punkten bestehen. Enthält ein Plotfile mehr Zeitschritte, kann die WHEN-Option benutzt werden, z.B.

WHEN(MOD(CYCLE,n)=0 & CYCLE < m)

Zeitverläufe können auch gezeichnet werden, wenn der Datenfile nicht ordnungsgemäß abgeschlossen ist.

4. Werden mehrere Kurven in ein Diagramm gezeichnet, dann sind diese durch verschiedene Symbole in unterschiedlichem Abstand markiert. Die Zuordnung der Symbole zu den Meßpunkten erfolgt durch Textangabe.

PLOT  
PNIVEAU

Standardwerte

```
DCL 1 PLOPPNIV,  
  2 PLOAX(2) DEC FLOAT(6) INIT(210.,150.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(32.,130.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(0.,0.,1.),  
  2 PROPIN(3) DEC FLOAT(6) INIT(0.,0.,1.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(0.02),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(4),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

```
PLOT OPEN PNIV DRAW CLOSE;
```

Erläuterung

PNIV erzeugt eine Höhenliniendarstellung des Druckfeldes bezogen auf die gemittelten Drücke der Maschen. Die Höhenlinien erfassen einen Bereich von 70 bar bis 130 bar mit einem Linienabstand von 2.5 bar. Die Maschen-Mitten sind durch Symbole gekennzeichnet:

	Symbol Nr.
70 bar $\leq$ P < 80	2
80 $\leq$ P < 90	3
90 $\leq$ P < 100	4
100 $\leq$ P < 110	5
110 $\leq$ P < 120	6
120 $\leq$ P < 130	7

Die Zuordnung der Symbolarten zu den Symbolnummern ist geräteabhängig.

Standardwerte:

```
DCL 1 PLOPPREL,  
  2 PLOAX(2) DEC FLOAT(6) INIT(240.,250.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(80.,80.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(1.,1.,0.6),  
  2 PROPIN(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(0.02),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(3),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

1. PLOT OPEN PREL DRAW CLOSE;
2. PLOT FILM PREL SCALE 0.015;  
    PLAX = '1'B;  
  
    PLOT DRAW CLOSE;

Erläuterung

Mit PREL wird eine 3D-Reliefdarstellung des Druckes in den Maschen erzeugt. Das Relief liegt auf einem Rahmen, dessen Grundlinie bei 50 bar liegt. Eine Skala von 50 bar bis 110 bar bezieht sich auf die vorderste Relieflinie. Mit SCALE können die X und Y-Werte skaliert werden. Der Maßstab für den Druck liegt mit 1.5 cm pro 10 bar fest. Für Druckdarstellungen in einem anderen Druckbereich (z.B. 1 bar) kann das Zeichnen der 50 bis 110 bar Skala mit

PLAX = '1'B;

unterdrückt werden.

PLOT  
VELOCITIES

Standardwerte:

```
DCL 1 PLOPVELO,  
  2 PLOAX(2) DEC FLOAT(6) INIT(200.,250.),  
  2 PRORIG(2) DEC FLOAT(6) INIT(50.,50.),  
  2 PROPAR(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 PROPIN(3) DEC FLOAT(6) INIT(1.,0.,0.),  
  2 XYZAX(3) DEC FLOAT(6) INIT(0.,0.,0.),  
  2 SCALE DEC FLOAT(6) INIT(0.02),  
  2 HIFI FILE VARIABLE,  
  2 PLTYP BIN FIXED(15) INIT(2),  
  2 (PLFRAME,PLTEXT) BIT(1) INIT('1'B),  
  2 PLAX BIT(1) INIT('0'B);
```

Beispiel:

```
PLOT OPEN VELO DRAW CLOSE;
```

Erläuterung:

Es werden die Geschwindigkeitsvektoren des Fluids im Ringraum gezeichnet. Die Länge der Vektoren ist normiert durch die maximal auftretende Geschwindigkeit. Der längste Vektor ist 2 cm lang. Die geometrischen Ausdehnungen können mit SCALE verändert werden.

### Syntax

PRESSURE expr;

### Erläuterung

1. Die Zustandsgleichung des Fluids wird in PRESSURE als Funktion von der Dichte  $\rho$  (RO) und der spez. in. Energie  $i$  (SIE) angegeben.

$$p = f(\rho, i)$$

2. Der Ausdruck expr stellt die rechte Seite der Zustandsgleichung, d.h.  $f(\rho, i)$ , dar.
3. Die angegebene Schreibweise für die Dichte und Energie (RO bzw. SIE) muß eingehalten werden, weil diese so in der Routine deklariert werden. Der Druck p (mit dem Variablennamen P) steht ebenfalls innerhalb des DECLARE-Statements.
4. In jedem Anwendungsbeispiel muß eine Zustandsgleichung angegeben werden, andernfalls erfolgt eine Fehlernachricht und ein abnormaler Abbruch der Berechnung.

### Beispiel:

1. PRESSURE 2./3.\*RO\*SIE;
2. PRESSURE 2.\*RO\*(SIE-1.)+760.;

## PRINT

### Syntax

PRINT WHENEVER cond;

### Erläuterung

1. Das PRINT-Statement dient zur Festlegung des gewünschten Output. Dies kann in Abhängigkeit von der Beobachtungszeit oder von der Zyklus-Zahl geschehen.
2. Unter cond ist eine gültige PL/1-Bedingung zu schreiben, in der die Beobachtungszeit (TIME) und/oder die Zyklus-Zahl (CYCLE) benutzt werden darf.
3. Als Bedingung eignet sich besonders die Modulo-Funktion (s.Beispiel), um jeden n-ten Zyklus zu drucken.

### Beispiel:

1. PRINT WHEN CYCLE > 10;
2. PRINT WHENEVER TIME = 2.;
3. PRINT WHENEV MOD(CYCLE,5) = 0;

## RESTART

### Syntax

RESTART FROM filename;

### Erläuterung

1. Das RESTART-Statement wird zum einen für einen Restart-Job, zum anderen aber auch für einen Plot-Job benötigt, um den Namen des Datenträgers anzugeben, von dem die gesuchten Zustandsgrößen gelesen werden sollen.
2. Der RESTART-Befehl sollte unmittelbar nach dem ENTER STRUYA stehen, damit neu zugewiesene Variablen nicht durch das Einlesen (READ) überschrieben werden.
3. filename muß z.B. mit  
DCL filename FILE;  
im Anwendungsprogramm deklariert werden.
4. Das DD-Statement in der Job Control Language kann hier z.B. folgendes Aussehen haben:

```
// G.SAVREX DD DSN=YASAVE.IRE313.REXCO,VOL=SER=TSTLIB,
```

```
//          DISP=(OLD,KEEP) , UNIT=3330
```

### Beispiel:

1. RESTART FROM SAVREX;

RIGHT

Syntax und Erläuterung siehe BOTTOM

## SAVE

### Syntax

```
SAVE [ EVERY n STEPS ] ON filename;
```

### Erläuterung

1. Sollen die errechneten Daten für Restart- oder Plot-Jobs abgespeichert werden, so definiert man in dem SAVE-Statement die Datei, in die geschrieben werden soll.
2. Außerdem steht die Möglichkeit offen, durch Angabe von EVERY n STEPS nur zu jedem n-ten Zeitschritt die Daten zu speichern.
3. Das Statement SAVE erfordert die Deklaration des 'filename' als FILE, z.B.: DECLARE filename FILE;
4. Ferner muß für diese Datei ein Data-Definition-Statement (DD-Statement) in der Job Control Language (JCL) vorhanden sein, z.B.:

```
// G.DAFILE DD DSN=YASAVE.IRE313.REXCD,UNIT=3330,VOL=SER=TSTLIB,  
//          DISP=(NEW,KEEP),SPACE=(TRK,(50,50)),  
//          DCB=(LRECL=32760,RECFM=VBS,BLKSIZE=13028)
```

### Beispiel:

1. SAVE ON DAFILE;
2. SAVE EV 10 STEPS ON PLOTBAND;

### Syntax

SOLVE [UNTIL cond] ;

### Erläuterung

1. Mit dem SOLVE-Statement wird der Modul aufgerufen, der je nach vorliegendem Beispiel (Rechenjob oder Restartjob) den Programmablauf steuert.

Das SOLVE-Statement muß deshalb am Ende des Anwendungsprogrammes stehen.

2. Unter cond kann eine Abbruchbedingung angegeben werden. Die Default-Abbruchbedingung ist: Beobachtungsdauer größer als 10 Sekunden.
3. Die wählbare Abbruchbedingung kann in Abhängigkeit von der Beobachtungsdauer (TIME) oder in der Abhängigkeit von der Anzahl der durchlaufenen Zeitschritte oder Zyklen (CYCLE) angegeben werden. Die Schreibweise TIME bzw. CYCLE muß hier eingehalten werden.

### Beispiel:

1. SOLVE;
2. SOL UNTIL TIME > 2.5;
3. SOLVE UNTIL CYCLE=50;

## STEP

### Syntax

$$\text{STEP } \left\{ \begin{array}{l} \underline{\text{CONSTANT}} \\ \underline{\text{VARIABLE}} \end{array} \right\} \text{ dt};$$

### Erläuterung

1. Mit diesem Eingabe-Statement wird festgelegt, ob während der numerischen Simulation des physikalischen Modells mit konstanten oder variablen Zeitschritten gerechnet werden soll.
2. Der Ausdruck dt gibt entweder den konstanten Zeitschritt an oder die Anfangs-Schrittweite; programmintern wird dann eine automatische Schrittweitenvergrößerung vorgenommen.
3. Das Zeitintervall dt ist initialisiert mit dem Wert 0.1. Standardmäßig wird mit sich verändernden Schrittweiten gerechnet.
4. Bei Anwendung des Strukturmodells CYLDY2 ist nur CONSTANT erlaubt.

### Beispiel:

1. STEP VAR 0.1;
2. STEP CONSTANT 1.E-3;

TITLE

TOP

### Syntax

```
TITLE 'text';
```

### Erläuterung

1. Mit TITLE wird die gewünschte Textüberschrift angegeben.  
Sie darf eine maximale Zeichenlänge von 80 Charakters haben.
2. Der Text wird in Apostrophs eingeschlossen.

### Beispiel:

```
TITLE 'FLUID- STRUCTURE CODE STRUYA';
```

TOP

Syntax und Erläuterung siehe BOTTOM

## TYPE

### Syntax

TYPE  $\left\{ \begin{array}{l} \underline{\text{CYLINDER}} \\ \underline{\text{RADIAL}} \\ \underline{\text{PLANE}} \\ \underline{\text{XYEBENE}} \end{array} \right\} ;$

### Erläuterung

1. Mit TYPE wird angegeben, ob die definierte Fluid-Struktur rotationssymmetrisch (CYLINDER, RADIAL) oder eben (PLANE, XYEBENE) sein soll.
2. Der Standardfall ist der Fall der Rotationssymmetrie.

### Beispiel:

1. TYPE PLANE;
2. TYPE R;

## VISCOSITY

### Syntax

```
VISCOSITY [ { AUTOMATIC
              { KINETIC
              { DYNAMIC } [, ]
            }
          [ LAMDA=expr,
          [ MUE = expr ] ] ;
```

### Erläuterung

1. Die Viskositätskoeffizienten  $\lambda$  und  $\mu$  werden mit dem Statement

VISCOSITY

angegeben.

2. Die Behandlung der Viscosität ist im Anhang 4 beschrieben.
3. Die Standardwerte sind  $\lambda = 10.0$  und  $\mu = 1.0$  und Automatic Viscosity.

### Beispiel

1. VISCOSITY LAMDA = 0.002, MUE = 0.0;
2. VIS MU = 1.0;

### 5.3 Das Anwendungsprogramm

Das STRUYA-Anwendungsprogramm kann für verschiedene Anwendungsklassen eingesetzt werden. Die Grundanwendungsart dient der Erzeugung von Problemdaten, daneben wurden Möglichkeiten zur graphischen Darstellung der STRUYA-Ergebnisse und zur Reparatur einer defekten Ergebnisdatei geschaffen. Die Befehle für diese drei Bereiche lauten

SOLVE  
PLOT  
COPY

Diese Befehle werden von REGENT in PL1-Calls übersetzt und sind damit die drei verschiedenen Ausgänge aus dem vom STRUYA-Anwender erstellten Hauptprogramm zu den Problembearbeitungsmodulen. Ein weiterer wichtiger Befehl, der dem PLOT immer vorausgeht, dem SOLVE nur dann, wenn ein bereits bearbeitetes Problem weitergerechnet werden soll, ist der

RESTART

Befehl. Seine Funktion und Anwendungsmöglichkeiten werden im Abschnitt 5.3.4 besprochen.

Eine grobe Zusammenstellung der wichtigsten Befehle für diese Anwendungen zeigt Anlage 5.1.

#### 5.3.1 Erzeugung von Problemdaten

Der Berechnung von Problemdaten voraus geht die Beschreibung dieses Problems durch ein physikalisch-geometrisches Modell und durch den zeitlichen Zustand der Problemlösung.

Im folgenden wird ein Programmbeispiel mit den wichtigsten Befehlen beschrieben. Zum weiteren Verständnis sei auf die Abschnitte 4. Strukturkopplung, 5.2 Syntaxbeschreibung und A2 STRUYA-COMMON und STATUS verwiesen.

Folgende Aspekte sind bei der Problembeschreibung zu beachten:

- 2-3 D Grid
- Kartesische Zylinderkoordinaten
- Ohne, mit Struktur
- Beginn, Weiterführung der Lösung
- Randbedingungen
- Physikalischer Zustand des Fluids
- Zeitliche Diskretisierung der Lösung
- Ausführungsart der Lösung (Euler, Lagrange)

Bereits beim Aufruf von STRUYA mit

```
ENTER STRUYA CELLS (ibar, jbar);
```

wird für ein zweidimensionales Maschengitter die Zahl der Maschen festgelegt. Die Angabe CELLS ist nur dann wirksam, wenn es sich nicht um einen Restartfall handelt.

Mit dem Befehl

```
TYPE
```

wird festgelegt, ob mit kartesischen oder mit Zylinderkoordinaten gerechnet wird. Zwei Dimensionen der Geometrie des Modells werden durch den Befehl MESH festgelegt, dessen einfachste Form

```
MESH RECT (dx, dy)
```

ein rechtwinkeliges Modell erzeugt, dessen Maschen eine Grundfläche  $dx \times dy$  haben (siehe Abschn.2.2.1). Die dritte Dimension oder auch alle drei, kann für ebene Geometrie ebenfalls mit dem MESH-Befehl festgelegt werden.

```
MESH HEIGHT;
```

```
STATUS ,R(I,J) = F(I,J);
```

```
END HEIGHT;
```

Hier kann die sonst standardmäßig 1 m große Höhe STATUS.R zwischen MESH HEIGHT und END HEIGHT programmiert werden. Wird ohne Strukturkopplung gerechnet, wird die Höhe während der Rechnung festgehalten. (Sie wird in jedem Zyklus von der Routine §§FSX0 in den Urzustand gesetzt). Bei Kopplung mit Struktur wird die Höhe vom Strukturmodell bestimmt.

Zur weiteren Beschreibung des physikalischen Modells ist es notwendig, den energetischen Anfangszustand mittels

INIT STATUS;

zu definieren. Hier wird der Dichte  $R_0$  und der inneren Energie  $SIE$  in den Maschen ein Anfangswert zugewiesen. Abgeschlossen wird diese Zuweisungsfolge durch ein

END STATUS;

Weichen die Fluidgeschwindigkeiten am Anfang vom Wert Null ab, so sind diese mit

INIT VELOCITIES;

vorzugeben. Diese Anfangsbedingungen sind nur beim Beginn einer Problemlösung zu programmieren. Die Randbedingungen für Dichte, Energie und Geschwindigkeit werden durch

BOTTOM STATUS;

RIGHT STATUS;

TOP STATUS;

BOTTOM VELOCITIES;

RIGHT VELOCITIES;

TOP VELOCITIES;

mit anschließender Angabe der Bedingungen festgelegt, z.B.

RIGHT VELOCITIES;

IF  $J > J_1$  &  $J < J_2$  THEN;

ELSE  $UN = 0$ ;

END RIGHT VELOCITIES;

Die Angabe von Randbedingungen ist nicht zwingend notwendig (z.B. bei Berechnung von Wellenbewegungen im geschlossenen Behälter). Da sie jedoch in jedem Zyklus abgefragt werden, müssen

einmal gegebene Randbedingungen auch bei Weiterführung einer Rechnung (RESTART) vorhanden sein.

Grundsätzlich immer vorhanden sein müssen Funktionen, i.a. interne Prozeduren, zur Berechnung des Druckes abhängig von Dichte und Energie und der quadrierten Schallgeschwindigkeit. In die Rechnung einbezogen werden diese Funktionen durch die Befehle PRESSURE und CSQUARE z.B.

```
PRESSURE DRUFU(RO,SIE);
```

```
CSQUARE CQUAD(RO,SIE);
```

Hier sind DRUFU und CQUAD zwei Prozeduren, die Druck und quadrierte Schallgeschwindigkeit als Funktionswert liefern und deren Namengebung dem Benutzer freigestellt ist.

Soll die Lösung nicht mit räumlich festem Eulergitter ausgeführt werden, dann werden die Angaben mit

```
GRID LAGRANGE;
```

(das Gitter bewegt sich mit der Strömung) oder hinter

```
GRID VELOCITY;
```

gemacht.

Für die zeitliche Diskretisierung wird der STEP-Befehl benutzt, für konstanten Zeitschritt mit Intervall dt

```
STEP CONSTANT dt;
```

und für variables Zeitintervall

```
STEP VARIABLE dt;
```

Hier ist dt der Anfangszeitritt. Nachdem durch die obigen Befehle das Problem beschrieben ist, kann mit dem SOLVE-Statement die Ausführung der Problemlösung gestartet werden. In diesem Statement kann auch die Jobabbruchbedingung i.a. die zu berechnende Problemzeit, aber auch die Anzahl der zu berechnenden Zyklen, z.B.

```
SOLVE UNTIL TIME = 0.1;
```

Das SOLVE-Statement bewirkt den Aufruf der #SOLVE-Routine und damit den Übergang in die Problemlösungsroutinen. In den MAIN-Modul wird erst nach Beendigung der Problemberechnung zurückgekehrt. Deshalb muß auch das SOLVE-Statement unbedingt am Ende eines Anwendungsprogrammes stehen. Dies gilt auch für die beiden folgenden Angaben zum Programm:

- Die berechneten Daten sollen ganz oder teilweise für Auswertung oder Problemweiterberechnung gerettet werden. Dazu wird der SAVE-Befehl benutzt. Die Anwendung dieses Befehles setzt voraus, daß ein File deklariert ist, dessen Name mit dem Namen der DD-Karte im JCL-Programm, die die Ausgabedatei beschreibt, übereinstimmt (Anlage 5.6), z.B.

```
    DCL OUTFILE FILE;  
    SAVE EVERY 2 STEP ON OUTFILE;  
    ⋮  
//OUTFILE DD DSN = .....
```

- Es soll ein gekoppeltes Fluid-Struktur-Problem berechnet werden. Das setzt voraus, daß für die Strukturseite ein Subsystem besteht, das mit

```
ENTER subsystem;
```

initialisiert wird (siehe Abschn. 4.3). Die Initialisierung wird abgeschlossen mit

```
END subsystem;
```

### 5.3.2 Graphische Darstellung von Ergebnissen

Mit dem PLOT-Befehl stellt STRUYA verschiedene Möglichkeiten zur Darstellung des benutzten Modells oder von Rechenergebnissen zur Verfügung. Für ein ebenes Modell kann die Aufteilung des

Gitters in der X-Y-Ebene gezeichnet werden. Weiterhin kann die Geschwindigkeit des Fluids abgebildet durch Vektorpfeile und die Druckverteilung in Relief und Niveaulinien dargestellt gezeichnet werden. Die Fluidschichtdicke kann ebenfalls als Relief und als Höhenlinienplot gezeichnet werden. Eine weitere Darstellungsmöglichkeit ist die schräge Ansicht eines Zylinders, dessen Oberfläche durch die stark vergrößerten Abweichungen der Strukturoberfläche von einem festen Radius verbogen wird.

Neben diesen Darstellungsformen von Momentaufnahmen des Problemzustandes können für den Druck und für die Fluidschichtdicke zeitabhängige Diagramme an beliebiger Stelle innerhalb des Gitters (Meßorte) gezeichnet werden.

Der PLOT-Befehl benutzt GIPSY / 8 /. Dadurch kann die Plotausgabe auf die verschiedenen Bildträger, auf die GIPSY zugreifen kann, erfolgen. Gegenwärtig sind in GIPSY folgende Einheiten bekannt:

Papier: CALCOMP, STATOS, XYNETICS

Bildschirm: TEKTRONIX 4014, TEKTRONIX 4081

Die Bildschirmdateien werden im AGF-Plotfileformat geschrieben / 9 /. Für diese Form existiert ein Postprozessorprogramm / 8/, das eine Datei erzeugt, die mit einem CALCOMP-Filmplotter auf einen Film übertragen werden kann.

Den Aufbau eines Plotprogramms zeigen die Beispiele 'Plot Film' und 'Plot Statos' (Anlagen 5.2, 5.3), er wird im folgenden kurz besprochen.

Der Befehl RESTART eröffnet den Zugriff auf eine STRUYA-SAVE-Datei (siehe Abschn. 5.3.4). Durch PLOT OPEN oder PLOT FILM wird GIPSY initialisiert und die Adresse der GIPSY-Datenstruktur in einen Pointer PGIPS eingetragen. Diese Adresse wird als Argument an die GIPSY-Ausföhrungsroutinen weitergegeben, wo auf die bereits allocierte GIPSY-Struktur durch

```
ENTER GIPSY REENTER(PGIPS);
```

wieder zugegriffen werden kann. Anschließend erfolgt die Definition des Plotauftrags im PLOT-Befehl.

In den Beispielen wird einmal mit OPEN ein Papierplot, zum anderen mit FILM eine Bildschirmdatei verlangt. Es werden drei Bilder: Druckrelief, Druckniveaulinien und eine Uhr in einen Rahmen (FRAME) gezeichnet. Die Angaben der weiteren Parameter überschreiben die zu jedem Bildtyp erzeugten Standardangaben (siehe Abschn. 5.2). Mit DRAW wird ein Aufruf einer den Auftrag abarbeitenden Routine (STPLGIPS) generiert. In einem Plotbefehl können bis zu 10 Teilbilder zusammengefaßt werden. CLOSE muß aus PL1-programmtechnischen Gründen den Abschluß eines Plotbefehles bilden. Es schließt einen Beginblock mit Labelnamen ab. Danach wird auch GIPSY wieder freigegeben.

Die Standardplotfilenamen sind in GIPSY für Papier: PLOTTAPE und für Film: PLOTLIB. Die Deklaration der PLOTLIB wird durch PLOT FILM generiert.

Anlage 5.4 zeigt ein Programm zur Erzeugung eines Druckdiagrammes für 3 Meßstellen.

### 5.3.3 Reparatur einer STRUYA-Datei

Mit dem Befehl COPY kann eine STRUYA-Savedatei ganz oder teilweise kopiert werden.

COPY dient hauptsächlich zur Reparatur von nicht ordnungsgemäß abgeschlossenen Files, d.h. Files, bei denen der letzte Zyklus durch einen Abend (z.B. Zeitüberschreitung) nicht vollständig geschrieben wurde. Die kopierte Datei muß physikalisch verschieden von der zu lesenden sein. Ihr Ende wird im COPY mit UNTIL festgelegt. (Anlage 5.5).

Eine andere Anwendungsmöglichkeit von COPY ist die Verkürzung einer Datei durch Aussortierung von Zyklen mit EVERY.

Z.B.

```
COPY INFILE TO OUTFILE EVERY 2 STEP UNTIL CYCLE 2000;
```

Hier enthält der OUTFILE nach dem Kopieren 1000 Zyklen. (Die Notwendigkeit dieser Anwendung kann z.B. bei der Erstellung eines Films auftreten.)

Mit der PRINT-Option kann schließlich der ordnungsgemäße Zustand der kopierten Datei dokumentiert werden. Durch PRINT wird die neue Datei vollständig gelesen, die Zyklusnummer und der zu jedem Zyklus gehörende Text 'GOON' sowie am Ende 'xEOF' geschrieben.

#### 5.3.4 Restart

Der Befehl RESTART erlaubt, eine Datei, die STRUYA Ergebnisse enthält, zu erweitern oder, wenn er dem PLOT-Befehl vorausgeht, diese graphisch darzustellen. Folgende Anwendungsgebiete gibt es für diesen Befehl:

- große Problemlösungszeiten
- abnormale Beendigung eines Rechenlaufes
- Änderung der Randbedingungen
- Änderung des Rechenzeitschrittes
- Erzeugung von Grafik-Dateien

Mit Hilfe des RESTART-Befehles kann die Berechnung eines Problems in mehrere Rechenläufe zerlegt werden. Dies wird dann notwendig, wenn sich bei feiner Diskretisierung des Rechenmodells sehr große Rechenzeiten ergeben oder wenn ein Rechenlauf abnormal beendet wurde. In diesem Fall muß jedoch die Problemdatei vor dem Restart restauriert werden (siehe COPY). Weiterhin kann Restart benutzt werden, um die Randbedingungen und die Zustandsgleichung zu ändern. Falls diese im Hauptprogramm abweichend von der Standardversion (Anhang 2) programmiert sind, müssen sie auch beim Restart definiert sein.

Durch Veränderung ihrer Funktion gegenüber dem vorangegangenen Rechenlauf kann damit die Problemstellung geändert werden. Neben den obengenannten Möglichkeiten, den Problemablauf zu steuern, kann auch ein Satz von Variablen zur Problem- und Ausführungsbeschreibung der im STRUYA-DATA-COMMON steht (Anhang 3), beim RESTART variiert werden.

Der RESTART-Befehl bewirkt neben dem Einlesen von zwei Namen auch die Auffüllung dieser COMMON-Daten noch während der Ausführung des MAIN-Programmes. Diese Daten können geändert werden durch Anschreiben der entsprechenden Anweisungen hinter dem RESTART-Befehl. Ausgenommen davon sind die Zyklusnummer #NCYC, die Zykluszeit # T und bei variabler Schrittweite # DT. Wie Anlage 5.7, 5.8 (Aufbau des SAVE-files und Ablauf RESTART) zeigen, werden diese Größen zusammen mit den Problemdaten für jeden Zyklus gespeichert und auch wieder gelesen.

Ein Beispiel für eine Ausführungssteuerung ist die Annahme, eine Problemlösung benötigt am Anfang eine feine Zeitdiskretisierung z.B.  $1.E-5/2$  und es genügt dann, wenn sich die Übergangsfunktion einem stationären Zustand nähert, ein größerer Zeitschritt, so kann man dies durch die Befehlsfolge:

```
RESTART FROM DAFILE;  
# DT = 1.E-4;
```

realisieren.

Wird RESTART benutzt, um neue Problemdaten zu erzeugen, dann werden zunächst alle auf einer STRUYA-Datei (dem Inputfile) befindlichen Zyklendaten bis zum letzten gelesen und sofort auf eine zweite Datei (den Outputfile) geschrieben. Von hier aus wird dann weitergerechnet und wenn gefordert, die Daten der Zyklen auf Outputfile geschrieben.

Bei Erzeugung von graphischer Ausgabe wird nach jedem gelesenen Zyklus geprüft, ob eine Plotaktion für diesen Zeitpunkt verlangt

wird, und gegebenenfalls die Aktionen für diesen Plot ausgeführt.

Anlagen 5.2 bis 5.4 sind Programm-Beispiele für RESTART. Problemdaten und Plots können nicht in einem Auftrag erzeugt werden.

1. START MIT STRUKTURSUBSYSTEM

---

```
ENTER STRUYA CELLS(NX,NY);
MESH RECT(DX,DY);
INIT STATUS;
INIT VELOCITIES;
BOTTOM,RIGHT,TOP STATUS,VELOCITIES;
PRESS PRESSFU(RO,SIE);
CSQUARE CQUAD(P,RO,SIE);
ENTER STRUKTUR;

.

END STRUKTUR;
TYPE;
STEP DT;
SAVE ON DAFILE;
SOLVE;
END STRUYA;
```

2. VORGABE DER FLUIDSCHICHTDICKE

---

```
ENTER STRUYA CELLS(NX,NY);
INIT STATUS;
INIT VEL;
BOTTOM,RIGHT,TOP STATUS,VELO;
TYPE;
MESH HEIGHT;

.

END HEIGHT;
PRESS PRESSFU;
CSQUARE CQUAD;
STEP;
SOLVE;
END STRUYA;
```

3. RESTART

---

```
ENTER STRUYA;
BOTTOM,RIGHT,TOP STATUS,VELOC;
PRES PRESSFU(RO,SIE);
CSQUARE;
RESTART FROM INFILE;
DCL (INFILE,OUTFILE) FILE;
SAVE ON OUTFILE;
SOLVE;
END STRUYA;
```

4. PLOT

---

```
ENTER STRUYA;
DCL DAFILE FILE;
RESTART FROM DAFILE;
PLOT .....DRAW CLOSE;
FINISH;
MESSAGE ACTIVE DEBUG;
END STRUYA;
```

5. COPY

---

```
ENTER STRUYA;
DCL (INFILE,OUTFILE) FILE;
COPY INFILE TO OUTFILE
EVERY NN STEP
UNTIL CYCLE > MM;
END STRUYA;
```

```
// EXEC REGENT,CPARM='NIS,C,SYN,MACRO,NS',PVOL=PNS001,
//   PLIB='TSO251.STRUYA',GPARM='PRINT,MAP/ISASIZE(200K)'
//P.SYSIN DD *
  BLOW:PROC OPTIONS(MAIN) REGENT(INIT,DA,PLOT=STATOS,MPOOL=600000,
    DPOOL=800000);

  ENTER STRUYA VERSION (1) ;
    PRINT WHEN CYCLE=8;
    DCL DAFILE FILE ;
    RESTART FROM DAFILE;
    PLOT OPEN
    WHEN CYCLE=2
      PREL FRAME 330. 260.          ORIGIN 130 140.
      PNIV NOFRAME                ORIGIN 79 101.
      CLOC NOFRAME                ORIGIN 270 60
      DRAW
    CLOSE;
  END STRUYA;
  MESSAGE ACTIVE DEBUG;
  FINISH;
END BLOW;
//C.SUBMAC DD DSN=TSO251.RING.CNTL,DISP=SHR
//G.DAFILE DD DSN=WT01A.DATA,UNIT=T6250,VOL=SER=DV0643,
//  DISP=(OLD,PASS),LABEL=(1,SL),DCB=DEN=4
//G.PLOTTAPE DD UNIT=T0800,VOL=SER=S251AL,LABEL=(1,NL),
//  DCB=DEN=2
```

Anlage 5.2: Beispielprogramm für Plot auf Statos

```
// EXEC REGENT,CPARM='NIS,C,SYN,MACRO,NS',PVOL=PNS001,
//   PLIB='TSO251.STRUYA',GPARM='PRINT,MAP/ISASIZE(200K)'
//P.SYSIN DD *
  BLOW:PROC OPTIONS(MAIN) REGENT(INIT,DA,PLOT=STATOS,MPOOL=600000,
    DPOOL=800000);

  ENTER STRUYA VERSION (1) ;
    PRINT WHEN CYCLE=8;
    DCL DAFILE FILE ;
    RESTART FROM DAFILE;
    PLOT FILM
    WHEN MOD(CYCLE,2)
      PREL FRAME 330. 260.          ORIGIN 130 140.
      PNIV NOFRAME                ORIGIN 79 101.
      CLOC NOFRAME                 ORIGIN 270 60
      DRAW
    CLOSE;
  END STRUYA;
  MESSAGE ACTIVE DEBUG;
  FINISH;
  END BLOW;
//C.SUBMAC DD DSN=TSO251.RING.CNTL,DISP=SHR
//G.DAFILE DD DSN=WT01A.DATA,UNIT=T6250,VOL=SER=DV0643,
//  DISP=(OLD,PASS),LABEL=(1,SL),DCB=DEN=4
//G.PLOTLIB DD DSN=FILM.DATA,DISP=(NEW,PASS),
//  UNIT=T6250,VOL=SER=FILMTP,LABEL=(1,SL)
//  DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB,DEN=4)
```

Anlage 5.3: Beispielprogramm für Plot auf AGF-Plotfile

```
// EXEC REGENT,CPARM='NS,NIS,C,SYN,MACRO',PLIB='TSO251.STRUYA',
// PVOL=PNS001,GPARM='PRINT,MAP/ISASIZE(400K),REPORT'
//P.SYSIN DD *
  BLOW:PROC OPTIONS(MAIN) REGENT(INIT,DA,PLOT=STATOS,MPOOL=500000,
    DPOOL=800000);
    ENTER STRUYA VERSION (1) ;
    DCL DAFILE FILE ;
    RESTART FROM DAFILE;
    PRINT WHEN CYCLE=4;
    PLOT OPEN PHIS FILE=PUPI SCALE 2.
      WHEN (MOD(CYCLE,8)=0)
        DRAW CLOSE;
    END STRUYA;
    MESSAGE ACTIVE DEBUG;
    FINISH;
  END BLOW;
//C.SUBMAC DD DSN=TSO251.RING.CNTL,DISP=SHR
//G.DAFILE DD DSN=WT01A.DATA,DISP=(OLD,PASS),
// UNIT=T6250,VOL=SER=DV0643,LABEL=(1,SL),DCB=DEN=4
//G.PLOTTAPE DD UNIT=T0800,VOL=SER=S251KK,LABEL=(1,NL),
// DCB=DEN=2
//G.PUPI DD *
  1 90 720.
  2 90 1593.
  3 90 7570.
```

Anlage 5.4: Beispiel Programm für History-Plot

```
// EXEC REGENT,CPARM='NIS,C,SYNTAX,MACRO',PLIB='TSO251.STRUYA',
//      PVOL=PNS001,GPARM='PRINT,MAP/ISASIZE(400K)'
//P.SYSIN DD *
  BLOW:PROC OPTIONS(MAIN) REGENT(INIT,DA,PLOT=STATOS,MPOOL=500000,
    DPOOL=800000);
    ENTER STRUYA VERSION (1) ;
  ON ERROR BEGIN;
    DCL DAFILE FILE ;
    DCL NEWFILE FILE;
    COPY DAFILE TO NEWFILE EVERY 2 STEP UNTIL CYCLE 6,PRINT;
    END STRUYA;
  MESSAGE ACTIVE DEBUG;
  FINISH;
  END BLOW;
//C.SUBMAC DD DSN=TSO251.RING.CNTL,DISP=SHR
//G.DAFILE DD DSN=WAFIGE.DATA,VOL=SER=DV0040,DISP=(OLD,PASS),
//  UNIT=T6250,LABEL=(1,SL)
//G.NEWFILE DD DSN=TSO251.FK1.DATA,DISP=SHR
//* UNIT=3330,VOL=SER=PNS002,SPACE=(TRK,(50,10)),
//* DCB=(BLKSIZE=13028,LRECL=32760,RECFM=VBS)
```

Anlage 5.5: Programmbeispiel für COPY

ABLAUF SAVE

```
*****  
INIT: | SAVE_PROC:  
      | WRITE #JNM  
      |       #NAME  
      |       #DATA  
      |       STRU_MODEL  
      |       CMMN  
      | CALL  STRUCT_SAVE('INIT')  
      |  
*****  
GOON: | SAVE_PROC:  
      | WRITE 'GOON'  
      |       #NCYC  
      |       #T  
      |       #DT  
      |       #TOUT  
      |       STATUS  
      | CALL  STRUCT_SAVE('GOON')  
      |  
*****  
END:  | QUI1_PROC:  
      | WRITE '*EOF'  
      |  
*****
```

AUFBAU SAVEDATEI

```
*****  
HEAD:  | #JNM           - |  
      | #NAME         | STRUYA COMMON |  
      | #DATA         |               |  
      | STRU_MODEL    - |  
      | HEAD OF STRUCTURE |  
      |               |  
*****  
CYCLE 0: | 'GOON'        |  
      | #NCYC        |  
      | #T           |  
      | #DT         |  
      | #TOUT       |  
      | STATUS      |  
      | STRUCTURE-DATA |  
      |               |  
*****  
CYCLE 1: | 'GOON'        |  
      | #NCYC        |  
      | #T           |  
      | #DT         |  
      | #TOUT       |  
      | STATUS      |  
      | STRUCTURE-DATA |  
      |               |  
*****  
CYCLE N: | 'GOON'        |  
      | #NCYC        |  
      | #T           |  
      | #DT         |  
      | #TOUT       |  
      | STATUS      |  
      | STRUCTURE-DATA |  
      |               |  
END:    | '*EOF'       |  
      |               |  
*****
```

ABLAUF RESTART

```
*****
INIT:  | MAIN PROC:
      |   READ #JNM
      |       #NAME
      |       #DATA
      |       STRU_MODEL
      | READ PROC:
      |   READ CMMN
      |   CALL STRUCT_READ('INIT')
      |   READ 'GOON'
      |
*****

GOON:  | READ PROC:
      |   READ #NCYC
      |       #T
      |       #DT(IF STEP_SIZE_VARIABLE)
      |       TOUT
      |       STATUS
      |   CALL STRUCT_READ('GOON')
      |   READ 'GOON'
      |
*****

GOON:  | READ PROC:
      |   READ #NCYC
      |       #T
      |       #DT(IF STEP_SIZE_VARIABLE)
      |       TOUT
      |       STATUS
      |   CALL STRUCT_READ('GOON')
      |   READ '*EOF'
      |
*****
```

Anlage 5.8: Ablauf RESTART

## A n h a n g A 1: Strukturmodule

Der folgende Abschnitt gibt eine Zusammenfassung der Dummy-StrukturROUTINEN mit Hinweisen für die Programmierung der Kopplung eines realen Strukturmodells. Die Kopplung ist in Abschnitt 4 beschrieben. Die Zuordnung der StrukturROUTINEN zu den sie aufrufenden zeigen Anlagen 4.1 und 4.2.

Der Informationsaustausch Fluid-Struktur erfolgt über den STATUS. (Siehe Anhang A3).

```
*SUBSYSTEM STRUYA;                                00000000
*MODUL $$DHDP;                                    00000000
*CPARM 'MACRO,OPT(2),A,AG,XREF,OF,NGS';           00000000
*PROCESS;                                          00000000
  $$DHDP:PROC(STATUS) REORDER;                    00000100
  /*                                                00000100
    DHDP BERECHNET ABLEITUNGEN VOM TYP DH/DP. DIE ERGEBNISSE
    WERDEN IN DIE FELDER DH2ODP UND DMH2ODP (SIEHE STATUS IN
    ANHANG A4) UEBERTRAGEN. ALLGEMEIN SIND HIER NACH EINEM
    RESTART EINMALIG ZU BERECHNENDE FLUIDABHAENGIGE STRUKTUR-
    FUNKTIONEN ZU SETZEN.                           00000100
    DIE DEKLARATION DES STATUS STECKT IN STFLDA.    00000100
    EINE SUBSYSTEMDATENSTRUKTUR WIRD UEBER DEN POINTER STRU_DAT_PTR
    MIT DEM HAUPTPROGRAMM DURCH FOLGENDE DEKLARATION VERBUNDEN:
    00000100
    DCL 1 SUBSYST_STRUKTUR BASED(STRU_DAT_PTR),     00000100
        2 .....                                    00000200
  /*                                                00000200
  %INCLUDE SUBMAC(STPREC);                          00000201
  %INCLUDE SUBMAC(STFLDA);                          00000200
  MESSAGE I TEXT('*** $$DHDP IST DUMMY ! ***');    00000200
END $$DHDP;                                         00000200
```

```

//IRE251DP JOB (0251,330,P4125),KATZ,REGION=1024K, 00000010
// NOTIFY=IRE251 00000020
// EXEC QQPCL,PLIB='TSO251.STRUYA',PVOL=PNS001 00000030
//SUBMAC DD DSN=TSO251.RING.CNTL,DISP=SHR 00000040
//P.SYSIN DD * 00000050
*SUBSYSTEM STRUYA; 00000060
*MODUL $$DISP; 00000070
*CPARM 'MACRO,OPT(2),A,AG,XREF,OF,NGS'; 00000080
*PROCESS; 00000090
  $$DISP:PROC(STATUS,TEXT) REORDER; 00000100
    DISP BERECHNET FUER GRAPHISCHE AUSGABE DIE DISPLACEMENTS DES 00000110
    KERNMANTELS. 00000120
    IN TEXT STEHT 'INIT' ODER 'GOON'. 00000130
    DIE DEKLARATION DES STATUS STECKT IN STFLDA. 00000140
    EINE SUBSYSTEMDATENSTRUKTUR WIRD UEBER DEN POINTER STRU_DAT_PTR 00000150
    MIT DEM HAUPTPROGRAMM DURCH FOLGENDE DEKLARATION VERBUNDEN: 00000160
    DCL 1 SUBSYST_STRUKTUR BASED(STRU_DAT_PTR), 00000170
      2 ..... 00000190
    HIER IN $$DISP IST SYMMETRIE VORRAUSGESETZT ZUR EBENE XMIN 00000200
    XMAX. 00000210
      VMAX = 1/2*(XMAX-XMIN) 00000220
      W = R - RO 00000230
  */ 00000240
%INCLUDE SUBMAC(STPREC); 00000250
%INCLUDE SUBMAC(STFLDA); 00000260
  ON ERROR CALL PLIDUMP('TFNB','HIST'); 00000270
  DCL TEXT CHAR(4); 00000280
  DCL RDELTA DEC FLOAT(6); 00000290
  DCL PI DEC FLOAT; 00000300
  PI = 2.*ACOS(0.); 00000310
  HPT,DH2ODP,DMH2ODP = 0.; 00000320
  RDELTA = #DTC(8); 00000330
  DO J = 2 TO #JBAR+2; 00000350
  DO I=1 TO #IBAR+1; 00000360
  DH2ODP(I,J) = - 0.5*(R(1,J)-R(#IBAR+1,J))*SIN(PI*X(I,J)/X(#IBAR+1,J)); 00000370
  DMH2ODP(I,J) = R(I,J) - RDELTA; 00000380
  00000390
  END; 00000400
  END; 00000410
  END $$DISP; 00000420

```



```

//IRE251$0 JOB (0251,330,P4125),KATZ,REGION=1024K      00000010
// EXEC QQPCL                                           00000020
//SUBMAC DD DSN=TSO251.RING.CNTL,DISP=SHR              00000030
//P.SYSIN DD *                                          00000050
*SUBSYSTEM STRUYA;                                     00000060
*MODUL $$FSX0 ;                                        00000070
*CPARM 'MACRO,OPT(2),A,AG,XREF,OF,NGS';               00000080
*PROCESS;                                              00000090
  $$FSX0:PROC(STATUS) REORDER;                         00000100
  /*                                                    00000110
    IN FSX0 WIRD DIE FLUIDSCHICHTDICKE H (IM STATUS : R) 00000120
    AUS DER STRUKTURVERFORMUNG BERECHNET.                00000130
    DIE DEKLARATION DES STATUS STECKT IN STFLDA.         00000140
    EINE SUBSYSTEMDATENSTRUKTUR WIRD UEBER DEN POINTER  STRU_DAT_PTR 00000150
    MIT DEM HAUPTPROGRAMM DURCH FOLGENDE DEKLARATION 00000160
    VERBUNDEN:                                           00000170
  DCL 1 SUBSYST_STRUKTUR BASED(STRU_DAT_PTR),           00000180
        2 .....                                         00000190
  FUER REINE FLUIDRECHNUNGEN WIRD DIE FLUIDSCHICHTDICKE 00000191
    UEBER #INIT_HEIGHT GESETZT.                          00000192
  /*                                                    00000193
  /*                                                    00000200
  %INCLUDE SUBMAC(STPREC);                              00000210
  %INCLUDE SUBMAC(STFLDA);                              00000220
    CALL #INIT_HEIGHT(X,Y,R);                            00000230
    MESSAGE I TEXT('*** IN $$FSX0 ***');                00000235
  END $$FSX0;                                           00000240

//IRE251$1 JOB (0251,330,P4223),KATZ,REGION=240K,TIME=3 00000010
// EXEC QQPCL                                           00000020
//SUBMAC DD DSN=TSO251.RING.CNTL,UNIT=3330,VOL=SER=REGICE, 00000030
//          DISP=SHR                                     00000040
//P.SYSIN DD *                                          00000050
*SUBSYSTEM STRUYA;                                     00000060
*MODUL $$FSX1;                                        00000070
*CPARM 'MACRO,OPT(2),A,AG,XREF,OF,NGS';               00000080
*PROCESS;                                              00000090
  $$FSX1:PROC(STATUS) REORDER;                         00000100
  /*                                                    00000110
    IN FSX1 WIRD DIE AENDERUNGSGESCHWINDIGKEIT DER FLUIDSCHICHT 00000120
    DICKE HPT (IN STATUS) AUS DER STRUKTURVERFORMUNGSGESCHWINDIG 00000130
    KEIT BERECHNET.                                     00000140
    DIE DEKLARATION DES STATUS STECKT IN STFLDA.         00000150
    EINE SUBSYSTEMDATENSTRUKTUR WIRD UEBER DEN POINTER  STRU_DAT_PTR 00000160
    MIT DEM HAUPTPROGRAMM DURCH FOLGENDE DEKLARATION 00000170
    VERBUNDEN:                                           00000180
  DCL 1 SUBSYST_STRUKTUR BASED(STRU_DAT_PTR),           00000190
        2 .....                                         00000200
  /*                                                    00000210
  /*                                                    00000220
  %INCLUDE SUBMAC(STPREC);                              00000220
  %INCLUDE SUBMAC(STFLDA);                              00000230
    MESSAGE I TEXT('*** $$FSX1 IST DUMMY ! ***');        00000240
  END $$FSX1;                                           00000250

```









## A2. Standard- und DUMMY-Routinen

### A2.1 Standard-Routinen

Um die Beschreibung eines physikalischen Modells zu vereinfachen, kann der Subsystem-Anwender sich der Standard-Routinen bedienen.

Die Standard-Routinen sind ausgelegt für einfache, oft wiederkehrende Problemstellungen und Randbedingungen.

Im folgenden werden die zur Verfügung stehenden Standard-Routinen beschrieben. Dazu wird auch angegeben, mit welchen Eingabe-Statements die Standard-Routinen übergangen bzw. durch andere Routinen ersetzt werden können.

#### STAND\_MESH

```
STAND_MESH:PROC(X,Y);
  DCL (X,Y) (*,*) BIN FLOAT(53);
  DCL (I ,J ,IP1,JP2)
      BIN FIXED(15);
  IP1 = #IBAR+1;
  JP2 = #JBAR+2;
  DO J=1 TO JP2;
    DO I=1 TO IP1;
      X(I,J) = (I-1)*#DR;
      Y(I,J) = (J-2)*#DZ;
    END;
  END;
END STAND_MESH;
```

#### Erläuterung

1. Die Standard-Routine STAND\_MESH erzeugt ein rechtwinkeliges Maschengitter mit den Maschenweiten, die in dem Eingabe-Statement

MESH RECT (dr, dz);

angegeben wurden, und den Maschenzahlen, die dem ENTER-Statement

```
ENTER STRUYA CELLS (ibar, jbar);
```

entnommen wurden.

2. In der Routine STAND\_MESH werden den Gitterpunkten die Koordinaten X und Y zugewiesen.
3. Soll ein Maschengitter generiert werden, das keine rechtwinkelig angeordneten Ränder hat, z.B. ein Rohrkrümmer, so sind die Statements

```
MESH;  
  ⋮  
END MESH;
```

zu benutzen.

### STAND\_HEIGHT

```
STAND_HEIGHT:PROC(X,Y,R);  
DCL(X,Y,R) (x,x) BIN FLOAT(53);  
  IF # CYL = 0 THEN  R = 1.;  
                      ELSE  R = X;  
END STAND_HEIGHT;
```

### Erläuterung

1. Die Standard-Routine STAND\_HEIGHT weist der Fluidschicht eine Dicke zu. Diese ist bei ebenem Gittertyp gleich eins, bei Zylindergeometrie ist  $R = X$ .

2. Eine davon abweichende Fluidschichtdicke kann mit dem Befehl MESH HEIGHT erzeugt werden (Abschnitt 5.2).
3. Bei Kopplung von STRUYA mit einem Strukturmodell wird die Dicke des Fluids von der Struktur bestimmt.

### STAND\_VELOCITIES

```
STAND_VELOCITIES:PROC(X,Y,U,V);
  DCL (X,Y,U,V) BIN FLOAT(53);
  U = 0.0;
  V = 0.0;
END STAND_VELOCITIES;
```

### Erläuterung

1. Die Routine STAND\_VELOCITIES ist die standardmäßige Routine zur Zuweisung der Geschwindigkeiten der Gitterknoten im Anfangszustand. Die Geschwindigkeiten U und V haben hier den Wert Null.
2. Durch die Eingabe-Statements

```
      INIT VELOCITIES;
      ⋮
      END VELOCITIES;
```

können von Null verschiedene Anfangsgeschwindigkeiten gesetzt werden.

### STAND\_BOUND

```
STAND_BOUND_ACC:PROC(U,V,X,Y,RM,ACC,I);
  DCL I BIN FIXED(15);
  DCL (U,V,RM,ACC) BIN FLOAT(53);
  DCL (X,Y) (*) BIN FLOAT(53);
  ACC = 0.0;
END STAND_BOUND_ACC;
STAND_BOUND_STATUS:PROC(RO,SIE,I);
  DCL (RO,SIE) BIN FLOAT(53);
  DCL I BIN FIXED(15);
END STAND_BOUND_STATUS;
STAND_BOUND_VEL:PROC(UN,UT,X,Y,I);
  DCL I BIN FIXED(15);
  DCL (UN,UT) BIN FLOAT(53);
  DCL (X,Y) (*) BIN FLOAT(53);
  UN = 0.0;
END STAND_BOUND_VEL;
```

Erläuterung

1. Die Routine `STAND_BOUND_ACC` wird standardmäßig aufgerufen und liefert den Beschleunigungsanteil 0 (Null) an den Knoten des Fluidrandes zurück. Es handelt sich also um eine starre Wand. Von dort wirken keine äußeren Kräfte auf die Flüssigkeit.
2. Mit `STAND_BOUND_STATUS` wird der energetische Zustand (Dichte, spez.innere Energie) in den Randmaschen nicht verändert. Es findet kein Massentransport über den Rand hinweg statt.
3. Die Geschwindigkeiten der Randknoten werden in der Routine `STAND_BOUND_VEL` dahingehend geändert, daß der Normalkomponente der Wert Null zugewiesen wird, während die Tangentialkomponente unbeeinflusst bleibt. Standardmäßig sind also 'Free Slip Walls' vorgesehen.
4. Die Standard-Routinen können ausgeschaltet werden, indem man

$$\left\{ \begin{array}{l} \text{BOTTOM} \\ \text{RIGHT} \\ \text{TOP} \end{array} \right\} \left\{ \begin{array}{l} \text{ACC} \\ \text{STATUS} \\ \text{VEL} \end{array} \right\} ;$$

⋮ Programm;

$$\text{END} \left\{ \begin{array}{l} \text{BOTTOM} \\ \text{RIGHT} \\ \text{TOP} \end{array} \right\} \left\{ \begin{array}{l} \text{ACC} \\ \text{STATUS} \\ \text{VEL} \end{array} \right\} ;$$

in dem Anwendungsprogramm angibt.

## A2.2 DUMMY-Routinen

Im Subsystem STRUYA gibt es mehrere DUMMY-Routinen. Sie werden immer dann in den Verarbeitungsmodulen aufgerufen, wenn im Anwendungsprogramm unerläßliche Angaben über das physikalische Modell fehlten.

Die DUMMY-Routinen drucken eine erklärende Meldung aus. Diese Routinen müssen auch beim RESTART vorhanden sein.

### DUMMY\_PRESSURE

```
DUMMY_PRESSURE:PROC(D1,D2,D3);  
  DC $\bar{L}$  (D1,D2,D3) BIN FLOAT(53);  
  PUT SKIP(2) LIST('*** MISSING INPUT STATEMENT PRESSURE');  
END DUMMY_PRESSURE;
```

### Erläuterung

1. Für jedes Rechenbeispiel muß eine Zustandsgleichung für das Fluid angegeben werden. Wird das PRESSURE-Statement vergessen oder enthält es einen Fehler, dann wird die Routine

DUMMY\_PRESSURE

aufgerufen, die die angegebene Fehlermeldung ausdrückt.

### DUMMY\_CSQUARE

```
DUMMY_CSQUARE:PROC(D1,D2) RETURNS(BIN FLOAT(53));  
  DC $\bar{L}$  (D1,D2) BIN FLOAT(53);  
  PUT SKIP(2) LIST('*** MISSING INPUT STATEMENT CSQUARE');  
  RETURN(0);  
END DUMMY_CSQUARE;
```

### Erläuterung

1. Für die Material-Schallgeschwindigkeit gilt dasselbe wie für die Zustandsgleichung. Das Quadrat der Material-Schallgeschwindigkeit ( $c^2$ ) muß immer angegeben werden. Unterbleibt diese Angabe, so schreibt die Routine

DUMMY\_CSQUARE

die Fehlernachricht heraus.

### DUMMY\_INIT\_STATUS

```
DUMMY_INIT_STATUS:PROC(D1,D2,D3,D4);  
  DCL (D1,D2,D3,D4) BIN FLOAT(53);  
  PUT SKIP(2) LIST('*** MISSING INPUT STATEMENT INIT STATUS');  
END DUMMY_INIT_STATUS;
```

### Erläuterung

1. Wird für ein Beispiel kein energetischer Anfangszustand mittels INIT\_STATUS; definiert, so sind für das Fluid keine zeitabhängigen Zustandsänderungen zu erwarten, denn die Zahlenfelder für die Dichte (RO) und die spezifische innere Energie (SIE) wurden am Anfang des Programms zu Null gesetzt.

Ohne eine Neudefinition dieser Zahlenfelder ist ein Beispiel also sinnlos. Werden keine neuen Werte für die Dichte oder die spezifische innere Energie gesetzt, so wird DUMMY\_INIT\_STATUS aufgerufen und die Fehlermeldung ausgedruckt.

A n h a n g 3

STRUYA DATA COMMON und STATUS

A3.1 Die Datenstruktur

Die von PLS erzeugte Datenstruktur des Subsystems STRUYA enthält alle Variablen-, Entry- und File-Namen, die in irgendeiner Weise der Problem- und der Ausführungsbeschreibung dienen. Diesen Größen kann also im Anwendungsprogramm ein neuer Ausdruck zugeordnet worden sein, so daß dieser jetzt mittels Datenstruktur auch in den ausführenden Modulen verfügbar sein muß.

```
DATA COMMON;
  DCL 1,
    2 #DATA,
      3 #A0      BIN FLOAT(53) INIT(0.75),
      3 #AOM     BIN FLOAT(53) INIT(0.75),
      3 #B0      BIN FLOAT(53) INIT(0.),
      3 #CYL     BIN FLOAT(53) INIT(1.),
      3 #DR      BIN FLOAT(53) INIT(1.),
      3 #DRPAR   BIN FLOAT(53) INIT(0.0),
      3 #DT      BIN FLOAT(53) INIT(0.1),
      3 #DTO (10) BIN FLOAT(53)
      INIT(0.1,0.5, 1.0,0.,0.,0.,0.,0.,0.,0.),
      3 #DTC (10) BIN FLOAT(53)
      INIT(1.0,5.0,10.0,0.,0.,0.,0.,0.,0.,0.),
      3 #DZ      BIN FLOAT(53) INIT(1.),
      3 #DZPAR   BIN FLOAT(53) INIT(0.0),
      3 #EPS     BIN FLOAT(53) INIT(1.E-4),
      3 #GR      BIN FLOAT(53) INIT(0.),
      3 #GRDVEL  BIN FLOAT(53) INIT(0.),
      3 #GZ      BIN FLOAT(53) INIT(0.),
      3 #GZP     BIN FLOAT(53) INIT(0.),
      3 #IBAR    BIN FIXED(15) INIT(20),
      3 #IBP     BIN FIXED(15) INIT(0),
      3 #IMOMX   BIN FIXED(15) INIT(1),
      3 #JBAR    BIN FIXED(15) INIT(32),
      3 #JBP     BIN FIXED(15) INIT(0),
      3 #JNM     CHAR(10)      INIT('STRUYA : '),
      3 #KXI     BIN FIXED(15) INIT(-1),
      3 #LAM     BIN FLOAT(53) INIT(10.),
      3 #MAXPAR  BIN FIXED(15) INIT(1),
```

```

3 #MU      BIN FLOAT(53) INIT(1.),
3 #NAME    CHAR(80)      INIT(' * REGENT-SUBSYSTEM: STRUYA
FLUID-DYNAMIC CODE * '),
3 #NCYC    BIN FIXED(15) INIT(0),
3 #NUMOUT  BIN FIXED(15) INIT(0),
3 #OM      BIN FLOAT(53) INIT(1.),
3 #PDR     BIN FLOAT(53) INIT(0.),
3 #PDZ     BIN FLOAT(53) INIT(0.),
3 #PRINT_OPTION
      BIT(1)          INIT('0'B),
3 #PYB     BIN FLOAT(53) INIT(0.),
3 #RESTART_REQUESTED
      BIT(1)          INIT('0'B),
3 #SAVE_STEP
      BIN FIXED(15)  INIT(1),
3 #STEP_SIZE_VARIABLE
      BIT(1)          INIT('1'B),
3 #T       BIN FLOAT(53) INIT(0.),
3 #YB      BIN FLOAT(53) INIT(0.),
3 #DX      DEC FLOAT(6)  INIT(7.),
3 #DY      DEC FLOAT(6)  INIT(10.),
3 #ZM      DEC FLOAT(6)  INIT(0.01),
2 #NOT_DATA,
3 #CSQUARE
      INIT(DUMMY_CSQUARE) ENTRY(BIN FLOAT(53),BIN FLOAT(53))
      RETURNS(BIN FLOAT(53)),
3 #PRESSURE
      BIN FLOAT(53)) ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      INIT(DUMMY_PRESSURE),
3 #FINAL_TIME
      ENTRY(BIN FLOAT(53),BIN FIXED(15))
      RETURNS(BIT),
3 #OUTPUT
      ENTRY(BIN FLOAT(53),BIN FIXED(15))
      RETURNS(BIT),
3 #GRID_VELOCITY
      ENTRY((*,*) BIN FLOAT(53),
      (*,*) BIN FLOAT(53), (*,*) BIN FLOAT(53)),
3 #INDOMAIN
      ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      BIN FLOAT(53),BIN FLOAT(53),BIN FLOAT(53),BIN FLOAT(53))
      INIT(STAND_DOMAIN) RETURNS(BIT),
3 #INIT_STATUS
      ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      BIN FLOAT(53),BIN FLOAT(53))
      INIT(DUMMY_INIT_STATUS),
3 #INIT_VELOCITIES
      ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      BIN FLOAT(53),BIN FLOAT(53))
      INIT(STAND_VELOCITIES),
3 #INIT_MESH
      ENTRY((*,*) BIN FLOAT(53),
      (*,*) BIN FLOAT(53)) INIT(STAND_MESH),
3 #PARTICLE_OPTION
      BIT(1)          INIT('0'B),
3 #PLOT_CYCLE
      ENTRY(BIN FIXED(15))
      INIT(STAND_PLOT) RETURNS(BIT),
3 #PLOT_REQUESTED
      BIT(1)          INIT('0'B),
3 #PLOT_MESH BIT(1)    INIT('1'B),
3 #PLOT_PART BIT(1)    INIT('0'B),
3 #PLOT_PRES BIT(1)    INIT('1'B),
3 #PLOT_VELO BIT(1)    INIT('1'B),
3 #USER_SCALE BIT(1)   INIT('0'B),
3 #USER_SHIFT BIT(1)   INIT('0'B),

```

```
3 #RESTART_FILE      FILE VARIABLE,
3 #SAVE_FILE         FILE VARIABLE,
3 #SAVE_REQUESTED   BIT(1)      INIT('0'B),
3 #BOTTOM_ACC       ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      (*) BIN FLOAT(53),(*) BIN FLOAT(53),BIN FLOAT(53),
      BIN FLOAT(53),BIN FIXED(15))
      INIT(STAND_BOUND_ACC),
3 #BOTTOM_STATUS    ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      BIN FIXED(15))
      INIT(STAND_BOUND_STATUS),
3 #BOTTOM_VEL       ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      (*) BIN FLOAT(53),(*) BIN FLOAT(53),BIN FIXED(15))
      INIT(STAND_BOUND_VEL),
3 #RIGHT_ACC        ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      (*) BIN FLOAT(53),(*) BIN FLOAT(53),BIN FLOAT(53),
      BIN FLOAT(53),BIN FIXED(15))
      INIT(STAND_BOUND_ACC),
3 #RIGHT_STATUS     ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      BIN FIXED(15))
      INIT(STAND_BOUND_STATUS),
3 #RIGHT_VEL        ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      (*) BIN FLOAT(53),(*) BIN FLOAT(53),BIN FIXED(15))
      INIT(STAND_BOUND_VEL),
3 #TOP_ACC          ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      (*) BIN FLOAT(53),(*) BIN FLOAT(53),BIN FLOAT(53),
      BIN FLOAT(53),BIN FIXED(15))
      INIT(STAND_BOUND_ACC),
3 #TOP_STATUS       ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      BIN FIXED(15))
      INIT(STAND_BOUND_STATUS),
3 #TOP_VEL          ENTRY(BIN FLOAT(53),BIN FLOAT(53),
      (*) BIN FLOAT(53),(*) BIN FLOAT(53),BIN FIXED(15))
      INIT(STAND_BOUND_VEL),
2 STRU_DATA,
3 STRU_DAT_PTR PTR   INIT(NULL()),
3 STRU_MODEL CHAR(2) INIT('$$'),
3 #INIT_HEIGHT      ENTRY((*,*) BIN FLOAT(53),
      (*,*) BIN FLOAT(53), (*,*) BIN FLOAT(53))
      INIT(STAND_HEIGHT);
END DATA;
```

Beschreibung der Variablen

#Ao	Koeffizient $\alpha_0$ in den Impuls-Gleichungen der Phase 3
#AoM	Koeffizient $\alpha_0$ in den Massen- und Energiegleichungen der Phase 3
#Bo	Koeffizient $\beta_0$ in den Massen- und Energiegleichungen der Phase 3
#CYL	0.0 für kartesische Koordinaten 1.0 für Zylinder-Koordinaten
#DR	Maschenweite $\delta_r$ in x/r-Richtung
#DRPAR	Partikelabstand $\delta_{r_p}$ in x/r-Richtung
#DT	Zeitschrittweite $\delta_t$
#DTO	Problemzeitintervalle, zu denen ausgedruckt wird
#DTOC	Zeitpunkt, an dem ein Problemzeitintervall geändert wird
#DZ	Maschenweite $\delta_z$ in y/z-Richtung
#DZPAR	Partikelabstand $\delta_{z_p}$ in y/z-Richtung
#EPS	Konvergenz-Kriterium der Iteration in Phase 2
#GR	Gravitation auf das Fluid in x/r-Richtung
#GRDVEL	"Grid-Velocity";=0.0 Euler;=1.0 Lagrange,=2. REZONE
#GZ	Gravitation auf das Fluid in y/z-Richtung
#GZP	Gravitation auf die Partikels in y/z-Richtung
#IBAR	Anzahl der Maschen in der Fluid-Region in x/r-Richtung
#IBP	Anzahl der Maschen in der Partikel-Region in x/r-Richtung
#IMOMX	= 1.0 für die Impulstausch-Option, andernfalls = 0.0
#JBAR	Anzahl der Maschen in der Fluid-Region in y/z-Richtung
#JBP	Anzahl der Maschen in der Partikel-Region in y/z-Richtung
#JNM	Job-Nummer
#KXI	$\xi$ , Exponent von $\rho$ , der die Art der Viskosität bestimmt
#LAM	Viskositätskoeffizient $\lambda$
#MAXPAR	maximal zulässige Partikelanzahl
#MU	Viskositätskoeffizient $\mu$

#NAME	Beschreibung des Beispiels
#NCYC	Zyklusnumerierung
#NUMOUT	Anzahl der ausgedruckten Zyklen
#OM	$\omega$ -Relaxation-Parameter in der Iteration der Phase 2 $\omega > 1$ Überrelaxation
#PDR	Maschenwerte $\mathcal{J}_r$ der Partikel-Region in x/r-Richtung
#PDZ	Maschenweite $\mathcal{J}_z$ der Partikel-Region in y/z-Richtung
#PRINT_OPTION	0 = Standardausdrucksfolge 1 = individuelle Ausdrucksfolge
#PYB	Y-Koordinate des linken unteren Knotens des Partikel-Gitters
#RESTART_REQUESTED	0 = neu startender Job 1 = RESTART-Job
#SAVE_STEP	Zyklusintervalle, bei denen die Zustands- größen gespeichert werden
#STEP_SIZE_VARIABLE	0 = konstante Schrittweite $\mathcal{J}_t$ 1 = variable Schrittweite $\mathcal{J}_t$
#T	Problemzeit, Beobachtungszeit
#YB	Y-Koordinate des linken unteren Knotens des Fluid-Gitters
#DX	Anfangsverschiebung dx beim Plotten
#DY	Anfangsverschiebung dy beim Plotten
#ZM	Zeichenmaßstab beim Plotten
#CSQUARE	Funktionsroutine für die Bestimmung der Material-Schallgeschwindigkeit
#PRESSURE	Subroutine zur Bestimmung des Druckes
#FINAL_TIME	Funktionsroutine, die die Abbruchbedingung enthält
#OUTPUT	Funktionsroutine zur Steuerung der Ausgabe
#GRID_VELOCITY	Subroutine zur Festlegung der Knotenge- schwindigkeit
#INDOMAIN	Routine zum Generieren von Partikels
#INIT_STATUS	Routine zum Initialisieren von Dichte und spez.in.Energie

```
#INIT_VELOCITIES Routine zum Initialisieren der Knotenge-
                    schwindigkeiten
#INIT_MESH        Routine zum Generieren des Maschengitters
#PARTICLE_OPTION  0 = keine Partikel
                  1 = mit Partikel oder Markers
#PLOT_CYCLE       Funktionsroutine zur Abfrage, ob eine
                  Zeichnung bei dem betreffenden Zyklus er-
                  wünscht ist
#PLOT_REQUESTED   0 = keine Zeichnungen
                  1 = Zeichnungen verlangt
#PLOT_MESH        für die Zeichnung des Gitters
#PLOT_PART        für die Zeichnung der Partikel
#PLOT_PRES        für die Zeichnung der Drucklinien
#PLOT_VELO        für die Zeichnung der Geschwindigkeiten
#USER_SCALE       Der Anwender hat den gewünschten Zeichen-
                  maßstab selbst festgelegt
#USER_SHIFT       Der Anwender hat die Verschiebung des Maschen-
                  gitters im Zeichenrahmen angegeben
#RESTART_FILE     Name des Restart-File
#SAVE_FILE        Name des Save-File
#SAVE_REQUESTED   0 = kein Schreiben auf Platte
                  1 = Schreiben auf Platte
#BOTTOM_ACC
    _STATUS
    _VEL
#RIGHT_ACC        Routinen für die Festlegung von Beschleu-
    _STATUS       nigungstermen, Zustandsgrößen ( f.e) und
    _VEL          Geschwindigkeiten an den drei Rändern
#TOP_ACC
    _STATUS
    _VEL
    STRU_DAT_PTR  Pointer an dem das Strukturmodell hängt
    STRU_MODEL    Prefix des Strukturmodells
#INIT_HEIGHT      Routine zur Festlegung der Anfangsfluid-
                  schichtdicke
```

### A3.2 Der Fluidstatus

Die Deklaration der Zustandsgrößen für Fluid sieht folgendermaßen aus:

```
DCL 1 STATUS(*,*) CONNECTED,  
    2 (X,R,Y,U,V,RO,SIE,E,RVOL,RM,P)  
    BIN FLOAT(PREC),  
    2 (HPT,DH2ODP,DMH2ODP,DPTIL)  
    BIN FLOAT(PREC);
```

Im ersten Deklarations-Statement STATUS werden die Zustandsgrößen für das Fluid definiert. Im einzelnen bedeuten:

X	X-Koordinate
R	R-Koordinate
Y	Y- oder Z-Koordinate
U	Geschwindigkeit in Richtung x oder r
V	Geschwindigkeit in Richtung y oder z
RO	Dichte des Fluids in der Masche
SIE	spezifische innere Energie in der Masche
E	Gesamtenergie in der Masche
RVOL	reziproker Wert des Maschenvolumens
RM	reziproker Wert der Knotenmasse
P	Druck in der Masche
HPT	Änderungsgeschwindigkeit der Maschenhöhe
DH2ODP	Durch Druckänderung bedingte Beschleunigung der Maschenhöhe (Struktur)
DMH2ODP	Durch Druckänderung bedingte Beschleunigung der strukturzugeschlagenen effektiven Fluidmasse
DPTIL	Druckänderung in Phase 2

Die Zeilen- und Spaltenzahl dieser Felder betragen einheitlich (IBAR+1, JBAR+2), obgleich einige Variablen für die Knoten und andere für die Maschen definiert sind.

Die bisher genannten Datenstrukturen müssen sowohl im Anwendungsprogramm wie auch in den Modulen verfügbar sein, weil hier die Eingabedaten übertragen werden.

Daneben gibt es noch die internen Strukturen, die von einem Modul zum anderen übergeben werden. Sie enthalten Konstanten und Hilfsgrößen für den codierten Rechenablauf. Diese internen Strukturen bzw. internen Deklarationen werden hier nicht beschrieben.

A n h a n g A4

Behandlung der Viskosität in STRUYA

Wenn die Viskosität in STRUYA als DYNAMIC oder KINEMATIC angegeben ist, so wird sie ebenso behandelt wie in YAQUIR. Wenn sie dagegen als AUTOMATIC angegeben ist, so wird eine Viskosität errechnet, die aus Stabilitätsgründen erforderlich ist. Diese Berechnung unterscheidet sich von YAQUIR insofern, als die Tatsache berücksichtigt wird, daß die Donor-Cell-Technik ohnehin stabilisierend wirkt.

1) DYNAMIC und KINEMATIC

Wir orientieren uns an / 2 /, Seite 6, und behandeln speziell die Gleichung

$$\pi_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda \nabla \vec{u} \quad (1)$$

bzw. die diskrete Form

$$\begin{aligned} \pi_{xx} = \frac{\mu}{2V} \cdot \left\{ \begin{aligned} &(r_2 + r_1)(u_2 + u_1)(y_2 - y_1) \\ &+ (r_3 + r_2)(u_3 + u_2)(y_3 - y_2) \\ &+ (r_4 + r_3)(u_4 + u_3)(y_4 - y_3) \\ &+ (r_1 + r_4)(u_1 + u_4)(y_1 - y_4) \\ &- \frac{CYL}{2} \cdot (u_1 + u_2 + u_3 + u_4) \cdot \left[ (x_1 - x_3)(y_2 - y_4) \right. \\ &\quad \left. + (x_2 - x_4)(y_3 - y_1) \right] \end{aligned} \right\} + \lambda \cdot D \end{aligned} \quad (2)$$

Die übrigen Gleichungen werden analog behandelt.

Im Programm selbst steht für

$\mu / 2$  die Variable MU02  
 $\lambda D$  die Variable LAMD

Diese Größen werden berechnet aus den Variablen

RO(I,J) Dichte  
#MU und #LAM Zähigkeit

in der Weise, daß bei kinematischer Zähigkeit

$$\begin{aligned}\mu &= \text{\#MU} \times \text{RO}(I,J) & (3) \\ \lambda &= \text{\#LAM} \times \text{RO}(I,J)\end{aligned}$$

bzw. bei dynamischer Zähigkeit

$$\begin{aligned}\mu &= \text{\#MU} & (4) \\ \lambda &= \text{\#LAM}\end{aligned}$$

gesetzt wird.

Abweichend von Gleichung (2) wird im Programm

$\lambda D = 0$  gesetzt, wenn  $D > 0$  ist.

Das heißt, daß bei Expansion dieser Anteil der Zähigkeit unterdrückt wird, während er bei Kompression wirkt. Siehe hierzu / 2 /, Seite 25.

## 2) AUTOMATIC

In / 2 /, Seite 24 wird als Stabilitätskriterium angegeben:

$$(\lambda + 2\mu) / \rho > \frac{1}{2} u^2 \int t + \frac{1}{2} u' \int x^2 \quad (5)$$

Für  $u$  und  $u' \cdot \int x^2$  werden Näherungsformeln benutzt.

$$u^2 \approx U(I,J) ** 2 + V(I,J) ** 2 \quad (6)$$

$$u' \cdot \Delta x^2 \approx \frac{1}{n} \max \{ |U(I,J)| \cdot \Delta r, |V(I,J)| \cdot \Delta z \} \quad (7)$$

Dabei gelten auch für  $\Delta r$  und  $\Delta z$  Näherungsansätze und  $n \geq 2$ .

Diese Näherungsformeln erweisen sich speziell im Fall (7) nicht als zweckmäßig, um den vielfältigen Strömungssituationen gerecht zu werden. Insbesondere sind diese Ausdrücke ungeeignet, wenn im Punkt (I,J) die Geschwindigkeit Null ist. In STRUYA wird daher benutzt:

$$u' \cdot \Delta x^2 = \max \left\{ (\max(u_k) - \min(u_k)) \cdot \Delta r, \right. \\ \left. (\max(v_k) - \min(v_k)) \cdot \Delta z \right\} (1 - 0.5\alpha) \\ - \alpha \sqrt{(U(I,J) \cdot \Delta r)^2 + (V(I,J) \cdot \Delta z)^2} \quad (8)$$

Darin sind

$u_k$  die Geschwindigkeiten in den vier Maschenecken

$\alpha$  der Donor-Cell-Faktor

Diese Gleichung berücksichtigt die dämpfende Wirkung des Donor-Cell-Verfahrens und bringt auch bei pathologischen Geschwindigkeitsfeldern noch brauchbare Ergebnisse. Gleichung (8) wurde allerdings mehr empirisch aufgestellt als analytisch abgeleitet.

### 3) Schrittweitenkriterium

Gemäß / 2 /, Seite 25 soll sein

$$\Delta t \leq \frac{\rho (1 - ANC) \Delta r^2 \Delta z^2}{2 (\lambda + 2\mu)_{eff} \cdot (\Delta r^2 + \Delta z^2)} \quad (9)$$

Im Falle der AUTOMATIC Viskosität wird dieses Kriterium nicht anhand derjenigen künstlichen Zähigkeit geprüft, die für die aktuelle Schrittweite gemäß (5) ermittelt wurde, sondern mit einer Zähigkeit, die für die aus anderen Gründen maximal mögliche Schrittweite DTPOS berechnet wird.

## A n h a n g A5

### Syntaxnotationen

In diesem Abschnitt erfolgt die Beschreibung der Syntaxnotation zu den Sprachelementen der problemorientierten Sprache.

#### 1. Großbuchstabile Texte und Sonderzeichen

Alle in den Syntaxbeschreibungen vorkommenden großbuchstabigen Texte und alle Sonderzeichen, die nicht Teile der Syntaxnotations-Sprache sind, sind konstante Elemente der beschriebenen Anweisung (oder eines Teils einer Anweisung). Sie müssen so kodiert werden, wie sie in der Syntaxbeschreibung angegeben sind.

Beispiel:   INIT STATUS;

          Diese Anweisung muß genau so kodiert werden,  
          wie sie hier steht.

Bezüglich der Leerzeichen gelten die PL/1-Regeln: Zwischen verschiedenen Sprachelementen (Benennungen, Operatoren und andere Begrenzungszeichen wie , ) ( ; und Konstanten) sind beliebig viele Leerzeichen erlaubt. Ein oder mehrere Leerzeichen müssen benutzt werden, um Benennungen oder Konstanten zu trennen, die nicht durch einen anderen Begrenzer getrennt sind. Im obigen Beispiel müssen also zwischen INIT und STATUS ein oder mehrere Leerzeichen stehen, zwischen STATUS und ; darf ein oder mehrere Leerzeichen stehen. Innerhalb von Benennungen und Konstanten sind keine Leerzeichen erlaubt.

          falsch: INI T STATUS;

## 2. Unterstreichung von großbuchstabigen Texten

Viele Schlüsselwörter in PLS können abgekürzt werden, d.h. nur die ersten  $i$  Buchstaben sind signifikant,  $i \geq 1$ . Danach dürfen beliebige Kombinationen von Buchstaben, Ziffern und der Zeichen #, \$, @, - stehen. Es gelten also für PLS-Schlüsselwörter die Regeln wie für PL/1-Benennungen.

Beispiel: VISCOSITY

Das Schlüsselwort VISCOSITY wird an den ersten drei Buchstaben VIS erkannt, die folgenden Zeichen sind ohne Bedeutung.

Gültig: VISC, VISSSS, VIS1

Nicht gültig: VI (zu wenig Zeichen), VI? (ungültiges Zeichen), VI SC (enthält Leerzeichen).

## 3) Kleinbuchstabige Texte

Alle mit kleinen Buchstaben geschriebenen Worte bezeichnen einen variablen Ausdruck. Anstelle des Wortes können z.B. Benennungen, Konstanten oder in einigen Fällen gültige PL/1-Expressions stehen. Die Bedeutung der Worte ist jeweils nach der Syntaxbeschreibung erläutert.

Beispiele: Benennung

SAVE ON filename;

"filename" ist hier ein variables Sprachelement, seine Zeichenkettenlänge richtet sich nach den Vorschriften für Namensgebung.

Expression

SOLVE UNTIL cond;

"cond" ist ein PL/1-Ausdruck, der nach den gültigen Regeln kodiert sein muß. Der Ausdruck darf Sonderzeichen enthalten.

PRESSURE expr;

"expr" kann hier ein arithmetischer Ausdruck sein oder aber auch einen Funktionsaufruf enthalten.

z.B.

PRESSURE 2./3.\*RO\*SIE;

PRESSURE DRUCK(RO,SIE);

Im 2. Fall muß aber die Funktion DRUCK im Anwendungsprogramm aufgeführt werden.

#### 4. Geschweifte Klammern

In geschweiften Klammern werden Alternativen aufgeführt, von denen eine ausgewählt werden muß.

Beispiel:    GRID    { EULER  
                          LAGRANGE } ;

Zulässige Alternativen sind:

GRID EULER;

GRID LAGRANGE;

#### 5. Eckige Klammern [ ]

Eckige Klammern umschließen eine Gruppe von Sprachelementen, die vorhanden sein können, aber nicht müssen.

Beispiel:    SAVE [EVERY n STEPS] ON filename;

Die Worte "EVERY n STEPS" können also entfallen.

R e f e r e n z e n

- / 1 / E.G.Schlechtendahl et al.;  
REGENT-Handbuch, KfK 2666, 1978
- / 2 / A.A.Amsden, C.W.Hirt:  
YAQUI: An Arbitrary Lagrangian-Eulerian Computer  
Program for Fluid Flow at all Speeds. LA-5100,  
Los Alamos Scien.Lab.,(March 73)
- / 3 / C.W.Hirt, A.A.Amsden, J.L.Cook:  
An Arbitrary Lagrangian-Eulerian Computing Method  
for all Flow Speeds. J.Comp.Phys.14,(1974) 227-253
- / 4 / A.Ludwig, R.Krieg:  
Dynamic Response of a Clamped/Ring-Stiffened Circular  
Cylindrical Shell under Non-axisymmetric Loading.  
Nucl.Eng.Design 43,(1977) 437-453
- / 5 / R.Krieg et al.:  
Blowdown loading and Dynamic Response of PWR-Vessel  
Internals. Nucl. Eng. Design 43,(1977) 419-435
- / 6 / PNS, 14. Halbjahresbericht / 2.Halbjahr 1978  
KfK 2750
- / 7 / R. Schuster:  
System und Sprache zur Behandlung graphischer In-  
formation im Rechnergestützten Entwurf.  
KfK 2305 (8.76)
- / 8 / G.Enderle et al.: GIPSY-Handbuch, KfK 2878 (1979)
- / 9 / G.Enderle et al.: AGF-Plotfile.  
Eine Datei zum Speichern und Transportieren graphischer  
Information. KfK 2776 (1979)