

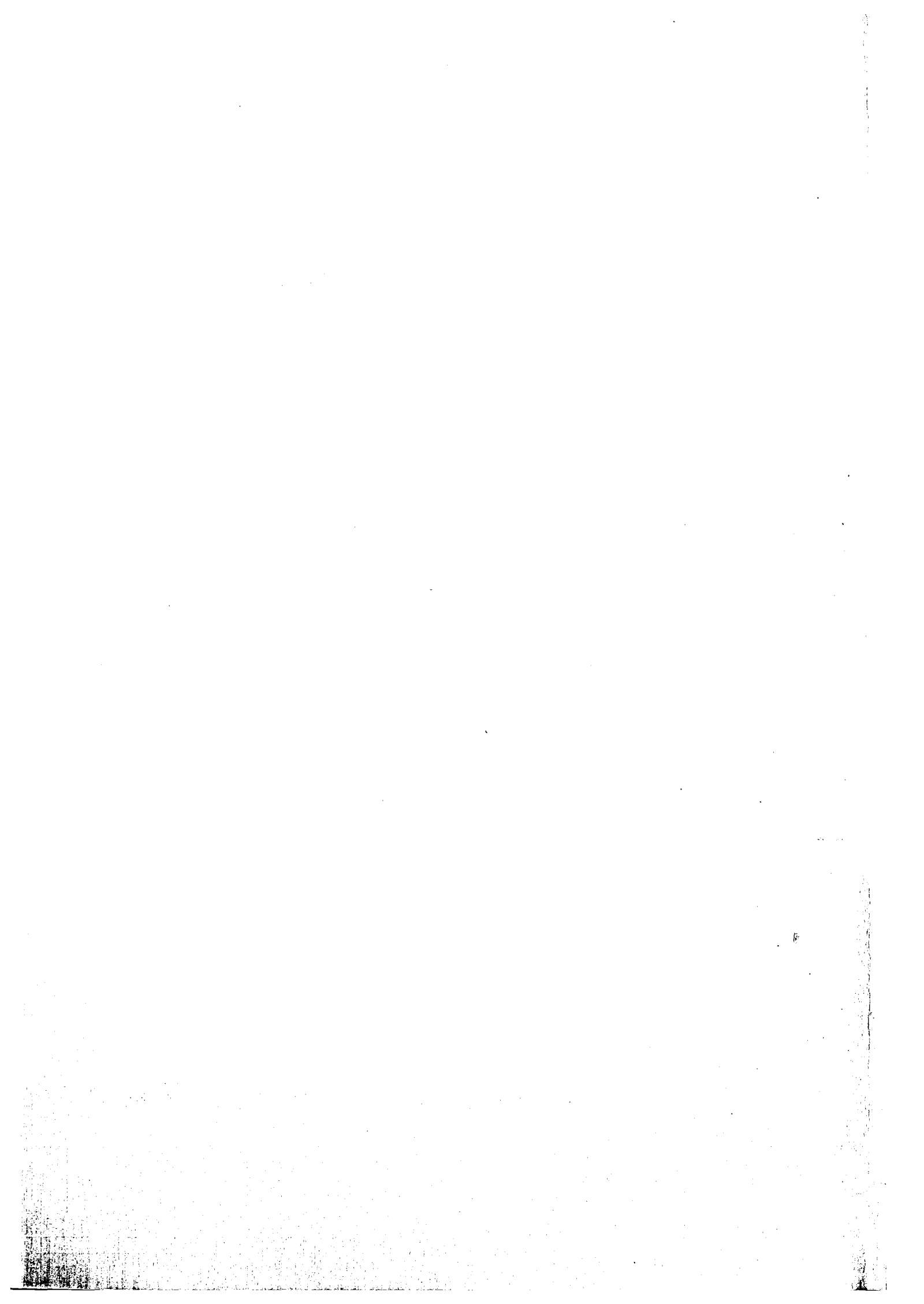


KfK 2679  
Januar 1979

**FLUST-2D — Ein Programm  
zur Berechnung der  
zweidimensionalen Strömung  
eines kompressiblen Mediums  
in aneinandergesetzten  
Rechteckbereichen**

G. Enderle  
Institut für Reaktorentwicklung  
Projekt Nukleare Sicherheit

**Kernforschungszentrum Karlsruhe**



KERNFORSCHUNGSZENTRUM KARLSRUHE  
Institut für Reaktorentwicklung  
Projekt Nukleare Sicherheit

KfK 2679

FLUST-2D - Ein Programm zur Berechnung der zweidi-  
mensionalen Strömung eines kompressiblen Mediums  
in aneinandergesetzten Rechteckbereichen

G. Enderle



## FLUST-2D - Ein Programm zur Berechnung der zweidimensionalen Strömung eines kompressiblen Mediums in aneinandergeschlossenen Rechteckbereichen

### Zusammenfassung:

Das Programm FLUST-2D erlaubt die Berechnung zweidimensionaler Strömungen eines kompressiblen Fluids in beliebig aneinander geschlossenen rechteckigen Bereichen. Durch ein Finite-Differenzen-Verfahren wird der zeitliche Verlauf von Druck, Dichte, innerer Energie und Geschwindigkeit berechnet. Ausgehend von den Grundgleichungen werden die Differenzengleichungen in einem rechtwinkligen Maschennetz entwickelt und der Ablauf des Integrationsschrittes für gekoppelte Fluidbereiche beschrieben. Anhand von Testbeispielen werden die Rechenergebnisse mit analytischen Lösungen verglichen, der Einfluß von Zeitschrittweite und Maschenweite wird untersucht.

Mit dem Programm wurden Vorausrechnungen der Blowdown-Versuche des HDR-Versuchsprogramms durchgeführt. Dabei wurden Ringraum, Innenraum, Plena, Bruchstützen und Containment zweidimensional modelliert. Die wesentlichen Ergebnisse dieser Vorausrechnungen werden vorgestellt. Darüberhinaus enthält dieser Bericht eine Beschreibung des Programmaufbaus und eine Bedienungsanleitung.

FLUST-2D - A Computercode for the Calculation of  
the two-dimensional Flow of a Compressible Medium in Coupled rectangular Areas

---

Abstract

The computer-code FLUST-2D is able to calculate the two-dimensional flow of a compressible fluid in arbitrary coupled rectangular areas. In a finite-difference scheme the program computes pressure, density, internal energy and velocity. Starting with a basic set of equations, the difference equations in a rectangular grid are developed. The computational cycle for coupled fluid areas is described. Results of test calculations are compared to analytical solutions and the influence of time step and mesh size are investigated.

The program was used to precalculate the blowdown experiments of the HDR experimental program. Downcomer, plena, internal vessel region, blowdown pipe and a containment area have been modelled two-dimensionally. The major results of the precalculations are presented. This report also contains a description of the code structure and user information.

<u>Inhaltsverzeichnis</u>	<u>Seite</u>
1. Einleitung	1
2. Formelzeichen	3
3. Lösungsverfahren	5
3.1 Grundgleichungen	5
3.2 Maschennetz und Differenzgleichungen	5
3.3 Randbedingungen	9
3.4 Zustandsgleichung	12
3.5 Ablauf des Integrationsschrittes	13
4. Geometrie und Topologie	24
4.1 Verbindung von Fluidbereichen	24
4.2 Terminologie	26
4.3 Ränder und Nähte	29
4.4 Geometrische Angaben	34
4.5 Topologische Angaben	35
5. Programmaufbau	37
5.1 Programmstruktur	37
5.2 Datenstruktur	44
5.3 Der Save-File	48
5.4 FLUST als REGENT-Subsystem	49
6. Benutzungsanleitung	52
6.1 Anwendersprache	52
6.2 Modellbeschreibung	62
7. Testrechnungen	71
7.1 Stoßrohr mit idealem Gas	71
7.2 Ausströmung von Wasser aus einem Behälter	74
7.3 Vergleich mit einem Zweiphasenprogramm	79
8. Anwendungsrechnungen	82
8.1 Blowdown-Versuchsprogramm am HDR	82
8.2 Modellierung des HDR	84
8.3 Ergebnisse	91
9. Erweiterung von FLUST um neue Modelle	134

	Seite
9.1 Anschluß eines Programms an einem Rand	134
9.2 Einbau eines neuen FLUST-Modells	137
9.3 Mögliche Vereinfachungen in FLUST	141
Literatur	144
Anhang A1 Einzelschritte der Lösung	147
A2 Programmstruktur	157
A3 Datenstruktur	170



## 1. Einleitung

Ziel der Programmentwicklung FLUST ist die Berechnung der Strömung eines Fluids in verschiedenen aneinandergeschlossenen Bereichen mit unterschiedlicher Modellierung. Bereiche unterschiedlicher Dimensionalität, mit unterschiedlichen Koordinatensystemen, mit und ohne Berücksichtigung der Zweiphasenströmung sollen verbunden werden können, die Koppelung mit einer Strukturberechnung soll möglich sein. Dieser Bericht beschreibt den gegenwärtigen Stand der Entwicklung. Innerhalb eines Code-Rahmens zur Aufnahme und Kopplung verschiedener Modelle wurde ein Satz von Programmen zur zweidimensionalen Berechnung kompressibler Fluide erstellt. Dieses Modell, FLUST-2D, ist durch folgende Eigenschaften gekennzeichnet:

- Differenzenverfahren in einem Eulerschen Maschennetz, kartesische Koordinaten
- Staggered Grid und Donor-Cell-Verfahren
- Verschiedene aneinandergeschlossene rechteckige zweidimensionale Fluidbereiche
- Rechteckige Maschen mit variablen Maschenweiten
- Feste Wände oder Flüssigkeitszufluß oder -abfluß am Rand
- Berücksichtigung einer konstanten Zähigkeit in der Impulsgleichung (keine Wandreibung, keine Drossel-, Stoß- und Umlenkverluste)
- Vernachlässigung von Volumenkräften (Schwerkraft) in der Impulsgleichung und von Reibung und Wärmeleitung in der Energiegleichung
- Berücksichtigung einer Zweiphasen-Strömung im mechanischen und thermodynamischen Gleichgewicht.

Die Vorgehensweise orientiert sich an den ursprünglich vom Los Alamos Scientific Laboratory übernommenen Programmen REMAC [1,23], YAQUI [2] und an dem Code KACHINA [3,4] und stützt sich ab auf den Arbeiten zur eindimensionalen FLUST-Version [5].

In diesem Bericht werden die benutzten numerischen Verfahren, der Programmaufbau und die Anwendung beschrieben. Für ein eindimensionales Stoßrohr mit idealem Gas werden die Rechenergebnisse mit einer analytischen Lösung verglichen. Anhand dieses Stoßrohres sowie mit der Berechnung einer Ausströmung von Wasser aus einem Rohr wird der Einfluß von Zeitschrittweite und Maschenweite auf die Ergebnisse untersucht. Die erste Anwendung des Codes FLUST-2D war die Vorausrechnung der Blowdown-Versuche des HDR-Programms [6].

Die zweidimensionale Modellierung von Ringraum, Plena, Innenraum, Brechstutzen und Containmentbereich sowie die wichtigsten Ergebnisse der Rechnungen werden vorgestellt.

## 2. Formelzeichen

e	spezifische innere Energie	$[\bar{\text{J/kg}}]$
u	Geschwindigkeit in x-Richtung	$[\bar{\text{m/s}}]$
v	Geschwindigkeit in y-Richtung	$[\bar{\text{m/s}}]$
p	Druck	$[\bar{\text{Pa}}]$
$\rho$	Dichte	$[\bar{\text{kg/m}^3}]$
$\alpha$	Modellkennzeichnung (S. 38)	
$\alpha_o$	Donor-Cell-Parameter	
$\beta_o$		
$\beta$	$\left(\frac{\partial D}{\partial p}\right)^{-1}$	Änderung der Quellstärke mit dem Druck (Kehrwert)
$\gamma_1$	Schwellen zur Zeitschritt-Halbierung bzw. Verdoppelung	
$\gamma_2$		
$\Delta x$	Breite einer Masche	$[\bar{\text{m}}]$
$\Delta y$	Höhe einer Masche	$[\bar{\text{m}}]$
$\Delta t$	Zeitschrittweite	$[\bar{\text{s}}]$
$\eta$	dynamische Zähigkeit	$[\bar{\text{kg/m}\cdot\text{s}}]$
$\epsilon$	Iterations-Abbruchschwelle	
$\xi$	Donor-Cell-Faktor	
$\omega$	Überrelaxationsparameter	
D	Quellstärke	$[\bar{\text{kg/m}^3\cdot\text{s}}]$
R	maximale Änderung von D bei der Iteration	
c	Schallgeschwindigkeit	$[\bar{\text{m/s}}]$
cycle	Anzahl Zeitschritte	
iter	Anzahl Iterationen	
konv	logische Variable, 'true', wenn die Iterationsabbruchbedingung erfüllt ist	
t	Zeit	$[\bar{\text{s}}]$
x	Dampfgehalt	
< Qu >	Donor-Cell-Interpolation zwischen $Qu_{\text{links}}$ und $Qu_{\text{rechts}}$	
$\delta Q$	Änderung einer Größe Q bei der Iteration	
$\tilde{Q}$	Vorläufiger Schätzwert der Größe Q vor und während der Iteration	

Indizierung

$Q_{i,j}$	Größe Q in der Maschenmitte
$Q_{i+1/2}^j$	Größe Q an der rechten Maschenbegrenzung
$Q_{i-1/2}^j$	Größe Q an der linken Maschenbegrenzung
$Q_i^{j+1/2}$	Größe Q an der oberen Maschenbegrenzung
$Q_i^{j-1/2}$	Größe Q an der unteren Maschenbegrenzung
$n_Q$	Größe Q zum Zeitschritt n
$Q_{in}$	Größe Q im Fluid eine Masche vom Rand entfernt

### 3. Lösungsverfahren

#### 3.1 Grundgleichungen

Masse:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \bar{u}) = 0 \quad (1)$$

Impuls:

$$\frac{\partial \rho \bar{u}}{\partial t} + \nabla(\rho \bar{u} \bar{u})^{*)} = -\nabla p + \eta \Delta \bar{u}, \quad \eta = \text{const.} \quad (2)$$

Energie:

$$\frac{\partial \rho e}{\partial t} + \nabla(\rho e \bar{u}) = -p \nabla \bar{u} \quad \text{umformen für Diferenzenbildung:} \quad (3)$$

$$\rho \frac{\partial e}{\partial t} + \underbrace{e \frac{\partial \rho}{\partial t} + \rho \bar{u} \nabla e}_{= 0 \text{ wegen (1)}} + \rho \bar{u} \nabla e = -p \nabla \bar{u} \quad (4)$$

$$\rho \left( \frac{\partial e}{\partial t} + \nabla(\bar{u} e) - e \nabla \bar{u} \right) = -p \nabla \bar{u} \quad (5)$$

Zustandsgleichung:

$$p = p(\rho, e) \quad (6)$$

#### 3.2 Maschennetz und Differenzgleichungen

Für die Differenzenform der Gleichungen wird ein rechtwinkeliges Eulersches Koordinatensystem zugrunde gelegt. Die Stützpunkte der Zustandsgrößen Druck, Dichte und innere Energie werden in die Maschenmitte gelegt, während die Geschwindigkeiten und Impulse an den Maschengrenzen definiert sind, auf denen sie senkrecht stehen (s. Abb. 1). Diese versetzte Anordnung der

\*) dyadisches Produkt

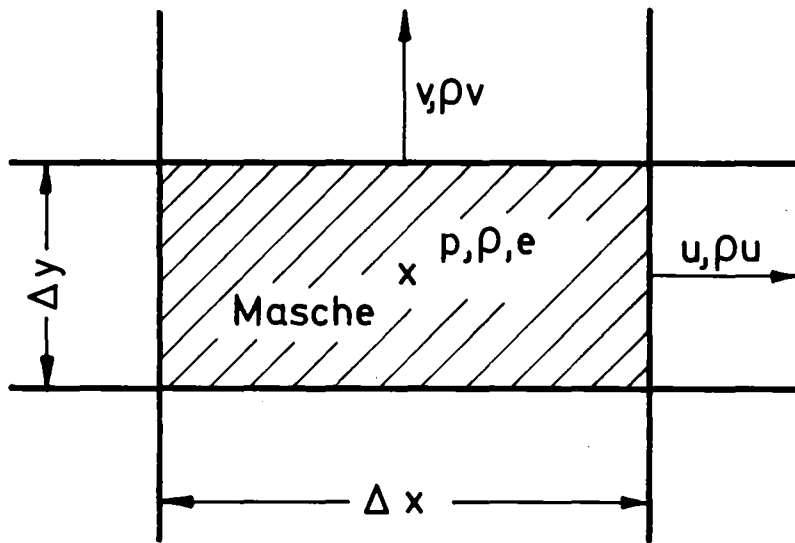


Abb. 1: Stützpunkte der Feldgrößen im Maschennetz

Größen im Maschennetz wird als "staggered grid" bezeichnet. Die Indizierung der Maschen und Maschengrenzen wird durch Abb. 2 verdeutlicht.

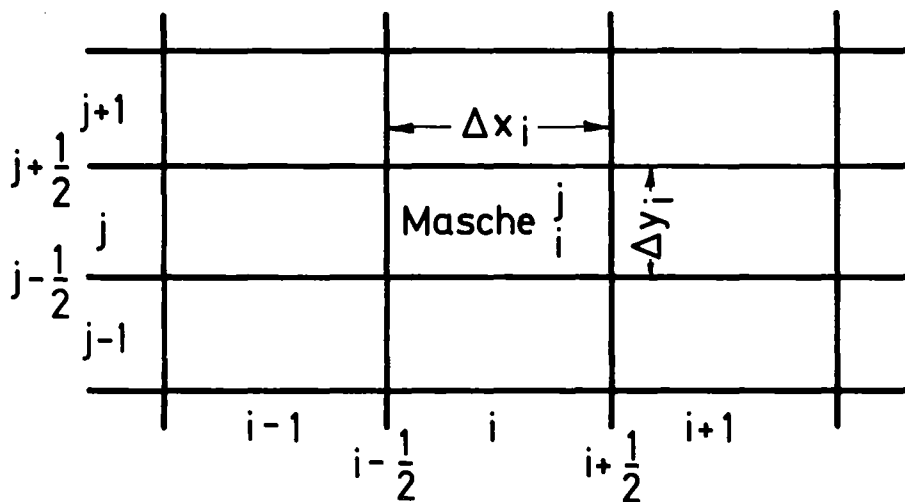


Abb. 2: Indizierung im Maschennetz

Zur Berechnung von Produkten einer Zustandsgröße mit einer Geschwindigkeit an einer Maschengrenze (z.B.  $S \cdot u_{i+1/2}^j$ ) muß zwischen den Werten der Größe in der linken und rechten Masche interpoliert werden. Eine lineare Interpolation macht das Integrationsverfahren numerisch instabil, es wird daher der Wert, der stromauf von der Maschengrenze liegt, stärker berücksichtigt. Diese Methode wird "donor-cell-interpolation" genannt.

Donor-Cell-Interpolation:

Q: Größe in der Maschenmitte

u: Geschwindigkeit an der Maschenbegrenzung

$$\langle Qu \rangle_{i+1/2}^j = u_{i+1/2}^j \left\{ \xi Q_i^j + (1-\xi) Q_{i+1}^j \right\} \quad (7)$$

$$\xi = (1-\alpha_0) \frac{\Delta x_{i+1} + \beta_0 \Delta t u_{i+1/2}^j}{\Delta x_i + \Delta x_{i+1}} + \frac{\alpha_0}{2} \left\{ \text{sign}(u_{i+1/2}^j) + 1 \right\} \quad (8)$$

Die Art der Interpolation wird durch die Faktoren  $\alpha_0$  und  $\beta_0$  bestimmt. Für  $\alpha_0$  und  $\beta_0 = 0$  ergibt sich eine reine von der Geschwindigkeit unabhängige lineare Interpolation zwischen  $Q_i$  und  $Q_{i+1}$ , die numerisch instabil sein kann. Für  $\alpha_0 = 1$  und  $\beta_0 = 0$  wird entweder der linke oder der rechte Wert der Größe genommen, je nach dem Vorzeichen der Geschwindigkeit (full donor cell). Bei  $\alpha_0 = 0$  und  $\beta_0 = 1$  hängt die Gewichtung zwischen dem linken und rechten Wert von der Größe der Geschwindigkeit ab. Ist  $u = \frac{\Delta x_i}{\Delta t}$  wird nur der linke Wert benutzt, bei  $u = -\frac{\Delta x_{i+1}}{\Delta t}$  nur der rechte Wert, dazwischen wird linear interpoliert.

Mit dem zugrunde gelegten Maschennetz und der Donor-Cell-Interpolation ergeben sich folgende Differenzgleichungen:

Masse:

$$\begin{aligned} {}^{n+1}D_i^j &\equiv \frac{{}^{n+1}S_i^j - {}^n S_i^j}{\Delta t} + \frac{{}^{n+1}\langle Su \rangle_{i+1/2}^j - {}^{n+1}\langle Su \rangle_{i-1/2}^j}{\Delta x_i} \\ &+ \frac{{}^{n+1}\langle Sv \rangle_{i+1/2}^j - {}^{n+1}\langle Sv \rangle_{i-1/2}^j}{\Delta y_i} \stackrel{!}{=} 0 \end{aligned} \quad (9)$$

Energie:

$${}^{n+1}e_i^j = {}^n e_i^j \left\{ 1 + \frac{\Delta t}{\Delta x_i} (u_{i+1/2}^j - u_{i-1/2}^j) + \frac{\Delta t}{\Delta y_j} (v_{i+1/2}^j - v_{i-1/2}^j) \right\}$$

$$- \frac{\Delta t}{\Delta x_i} (\langle eu \rangle_{i+1/2}^j - \langle eu \rangle_{i-1/2}^j) - \frac{\Delta t}{\Delta y_j} (\langle ev \rangle_{i+1/2}^j - \langle ev \rangle_{i-1/2}^j)$$

$$- \frac{p_i^j \Delta t}{s_i^j} \left( \frac{u_{i+1/2}^j - u_{i-1/2}^j}{\Delta x_i} + \frac{v_{i+1/2}^j - v_{i-1/2}^j}{\Delta y_j} \right) \quad (10)$$

Impuls:

(2) in Komponenten zerlegt:

$$\frac{\partial s u}{\partial t} + \frac{\partial s u^2}{\partial x} + \frac{\partial s v u}{\partial y} = - \frac{\partial p}{\partial x} + \eta \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (11)$$

$$\frac{\partial s v}{\partial t} + \frac{\partial s u v}{\partial x} + \frac{\partial s v^2}{\partial y} = - \frac{\partial p}{\partial y} + \eta \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (12)$$

daraus:

$$\begin{aligned} & \frac{{}^{n+1}(su)_{i+1/2}^j - {}^n(su)_{i+1/2}^j}{\Delta t} + \frac{2(\langle su^2 \rangle_{i+1/2}^j - \langle su^2 \rangle_i^j)}{\Delta x_i + \Delta x_{i+1}} + \frac{\langle svu \rangle_{i+1/2}^j - \langle svu \rangle_{i-1/2}^j}{\Delta y_j} \\ &= - \frac{2(p_{i+1}^j - p_i^j)}{\Delta x_i + \Delta x_{i+1}} + \frac{2\eta}{\Delta x_i + \Delta x_{i+1}} \left( \frac{u_{i+3/2}^j - u_{i+1/2}^j}{\Delta x_{i+1}} - \frac{u_{i+1/2}^j - u_{i-1/2}^j}{\Delta x_i} \right) \\ & \quad + \frac{2\eta}{\Delta y_j} \left( \frac{u_{i+1/2}^{j+1} - u_{i+1/2}^j}{\Delta y_j + \Delta y_{j+1}} - \frac{u_{i+1/2}^j - u_{i+1/2}^{j-1}}{\Delta y_{j-1} + \Delta y_j} \right) \end{aligned} \quad (13)$$



$$\begin{aligned}
 & \frac{n+1 (\rho v)_i^{j+1/2} - n (\rho v)_i^{j+1/2}}{\Delta t} + \frac{\langle \rho u v \rangle_{i+1/2}^{j+1/2} - \langle \rho u v \rangle_{i-1/2}^{j+1/2}}{\Delta x_i} + \frac{2 (\langle \rho v^2 \rangle_i^{j+1} - \langle \rho v^2 \rangle_i^j)}{\Delta y_j + \Delta y_{j+1}} \\
 &= - \frac{2 (p_i^{j+1} - p_i^j)}{\Delta y_j + \Delta y_{j+1}} + \frac{2 \eta}{\Delta x_i} \left( \frac{v_{i+1}^{j+1/2} - v_i^{j+1/2}}{\Delta x_{i+1} + \Delta x_i} - \frac{v_{i,j+1/2} - v_{i-1}^{j+1/2}}{\Delta x_i + \Delta x_{i-1}} \right) \\
 & \quad + \frac{2 \eta}{\Delta y_{j+1} + \Delta y_j} \left( \frac{v_{i,j+3/2} - v_{i,j+1/2}}{\Delta y_{j+1}} - \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y_j} \right) \quad (14)
 \end{aligned}$$

In diesen Gleichungen sind alle Größen, die nicht besonders gekennzeichnet sind, zum neuen Zeitschritt n+1 anzuschreiben, wenn das Verfahren voll implizit sein soll. Aus Gründen der Effektivität werden jedoch einige Werte zum alten Zeitpunkt benutzt, im Abschnitt 3.5 wird die Vorgehensweise bei der Zeitintegration im Detail erläutert. Das verwendete Differenzenverfahren hat eine Genauigkeit zweiter Ordnung im Raum und erster Ordnung in der Zeit.

### 3.3 Randbedingungen

Die Fluidbereiche sind nach außen hin von Rändern abgeschlossen. Als Randbedingungen sind feste Wände oder Zu- und Abflüsse möglich. Die Randbedingungen werden im Maschennetz dadurch realisiert, daß fiktive Randmaschen, die außerhalb der Ränder die Fluidbereiche umgeben, mit bestimmten Werten gefüllt werden. Abb. 3 zeigt einen waagerechten Rand mit den fiktiven Randmaschen und den Randwerten. Folgende Randbedingungen werden betrachtet: Feste Wand mit und ohne Haftung, erzwungener Zufluß, freier Abfluß und zyklische Randbedingung.

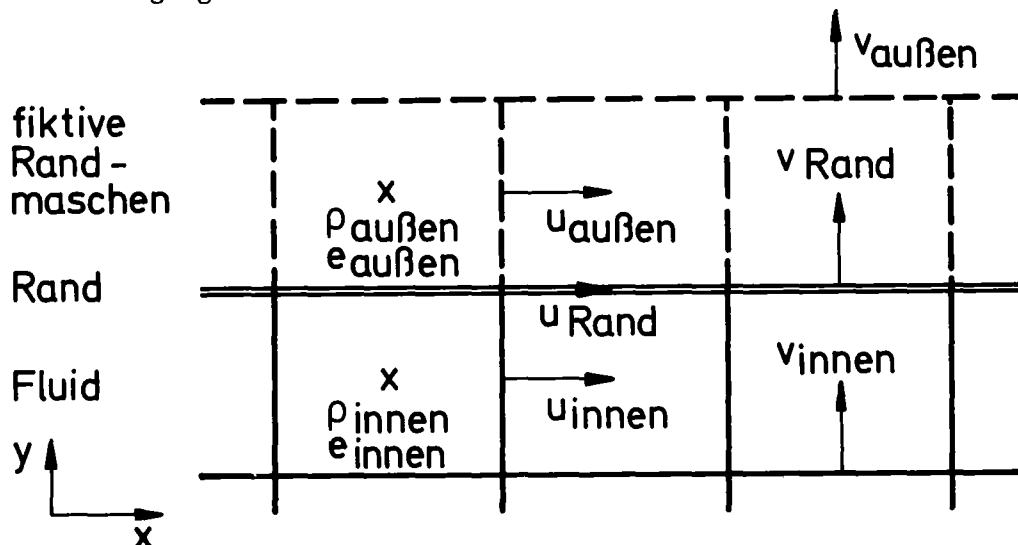


Abb. 3: Senkrechter Rand mit fiktiven Randmaschen

Randbedingungen:

(u ist die Geschwindigkeitskomponente parallel, v senkrecht zum Rand)

Werte der Randmaschen:

Feste Wand mit Haftung

$$u_{\text{Rand}} = 0$$

$$u_{\text{außen}} = -u_{\text{innen}}$$

$$v_{\text{Rand}} = 0$$

$$v_{\text{Rand}} = 0$$

$$\frac{\partial \mathcal{S}}{\partial y} \text{ Rand} = 0$$

$$\mathcal{S}_{\text{außen}} = \mathcal{S}_{\text{innen}}$$

$$\frac{\partial e}{\partial y} \text{ Rand} = 0$$

$$e_{\text{außen}} = e_{\text{innen}}$$

Feste Wand ohne Haftung

$$v_{\text{Rand}} = 0$$

$$v_{\text{Rand}} = 0$$

$$\frac{\partial u}{\partial y} \text{ Rand} = 0$$

$$u_{\text{außen}} = u_{\text{innen}}$$

$$\frac{\partial \mathcal{S}}{\partial y} \text{ Rand} = 0$$

$$\mathcal{S}_{\text{außen}} = \mathcal{S}_{\text{innen}}$$

$$\frac{\partial e}{\partial y} \text{ Rand} = 0$$

$$e_{\text{außen}} = e_{\text{innen}}$$

Erzwungener Zufluß

$$\mathcal{S}^{v_{\text{Rand}}} = f_1(x, y, t)$$

$$\mathcal{S}^{v_{\text{Rand}}} = f_1(i, j, t)$$

$$\mathcal{S}^{u_{\text{Rand}}} = 0$$

$$\mathcal{S}^{u_{\text{außen}}} = 0$$

$$\mathcal{S}_{\text{außen}} = f_2(x, y, t)$$

$$\mathcal{S}_{\text{außen}} = f_2(i, j, t)$$

$$e_{\text{außen}} = f_3(x, y, t)$$

$$e_{\text{außen}} = f_3(i, j, t)$$

damit

$$p_{\text{außen}} = p(\mathcal{S}_{\text{außen}}, e_{\text{außen}})$$

Freier Abfluß

$$u_{\text{Rand}} = 0$$

$$u_{\text{außen}} = 0$$

$$\mathcal{S}_{\text{außen}} = f_1(x, y, t)$$

$$\mathcal{S}_{\text{außen}} = f_1(i, j, t)$$

$$e_{\text{außen}} = f_2(x, y, t)$$

$$e_{\text{außen}} = f_2(i, j, t)$$

damit

$$p_{\text{außen}} = p(\mathcal{S}_{\text{außen}}, e_{\text{außen}})$$

vorgegeben.

Die Randbedingung  $g_{\text{Rand}}^v = 0$  und  $v_{\text{Rand}} = 0$  für erzwungenen Zufluß und freien Abfluß werden durch Setzen der Werte eine halbe Maschenweite außerhalb auf Null nicht genau erfüllt. Diese Vorgehensweise wurde aus Gründen der Einfachheit gewählt.

### Zyklische Randbedingung

Hierbei wird der rechte Rand eines Fluides mit dem linken verbunden, so daß der zweidimensionale Bereich zu einer Zylinderoberfläche zusammengebogen gedacht werden kann (Abb. 4). Diese Randbedingung ist hilfreich bei der Modellierung dünner zylindrischer Räume, wie sie etwa beim Ringraum eines Reaktors gegeben sind.

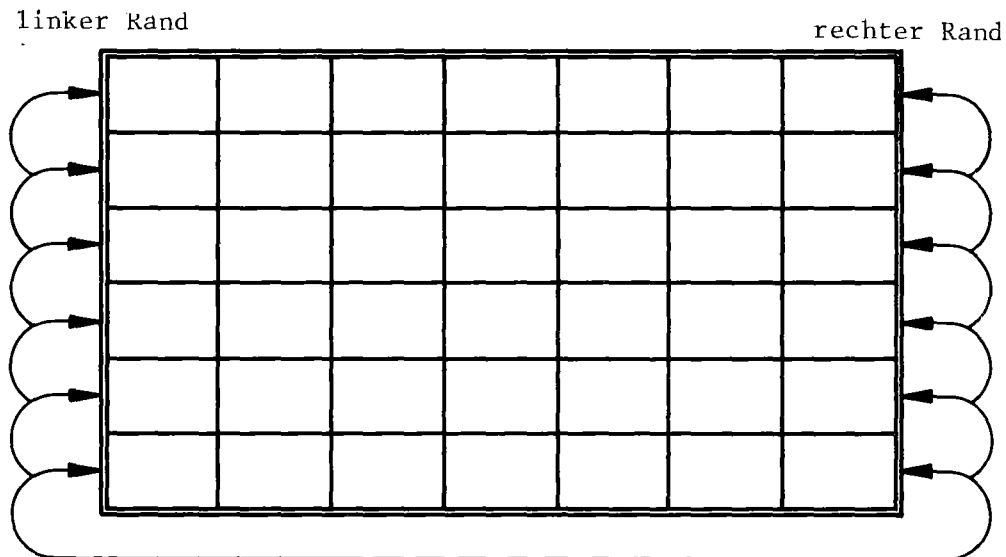


Abb. 4: Zyklische Randbedingung

Randbedingung:

rechter Rand = linker Rand

Werte der Randmaschen:

$$u_{\text{Rand}}^{\text{rechts}} = u_{\text{Rand}}^{\text{links}}$$

$$v_{\text{außen}}^{\text{rechts}} = v_{\text{innen}}^{\text{links}}$$

$$v_{\text{außen}}^{\text{links}} = v_{\text{innen}}^{\text{rechts}}$$

ebenso für  $g$  und  $e$

### 3.4 Zustandsgleichung

Die Zustandsgleichung des betrachteten Fluids liefert den Druck in Abhängigkeit von Dichte und innerer Energie. Eine beliebige Funktion  $p = p(\varrho, e)$  kann als Unterprogramm realisiert dem Programm angegliedert werden. Schon vorhanden sind die Zustandsgleichung für ideales Gas und zwei Zustandsgleichungen für Wasser.

#### Ideales Gas:

$$p = (\gamma - 1)\varrho e \quad (15)$$

#### Lineare Wassergleichung:

$$p = p_0 + c^2(\varrho - \varrho_0) \quad (16)$$

$p_0, \varrho_0$  und  $c^2$  sind dabei Konstanten, die noch je nach betrachtetem Problem angepaßt werden können.  $p_0$  ist in den meisten Fällen der Anfangsdruck des Fluids,  $\varrho_0$  die zu  $p_0$  und der Anfangstemperatur  $T_0$  gehörende Dichte.  $c$  ist der als Schallgeschwindigkeit bezeichnete Ausdruck  $\sqrt{\frac{\partial p}{\partial \varrho}}$  bei konstanter Entropie  $s$ . Der Druck ist unabhängig von der inneren Energie. Das bedeutet, daß die Temperatur des Fluids keinen Einfluß auf die Strömung hat.

#### "Genaue" Wassergleichung

Diese Gleichung soll die Zustandswerte von Wasser möglichst so wiedergeben, wie sie anhand von Versuchen bestimmt und in Wassertafeln [7] wiedergegeben sind. Dazu wurde ein Unterprogramm H2OZUST [8] in FLUST integriert, das den Druck abhängig von Dichte und innerer Energie anhand von aus der Wassertafel entnommenen Tabellen ermittelt. Dieses Programm deckt einen Temperaturbereich von  $5^\circ\text{C}$  bis  $370^\circ\text{C}$  und einen Druckbereich von 1000 Pa bis 22 MPa ab. Außer Wasser und reinem Dampf wird auch das Zweiphasengebiet erfaßt. Dadurch ist die Betrachtung einer homogenen Zweiphasenströmung in FLUST möglich. Allerdings wird dabei das mechanische Ungleichgewicht (Schlupf) und das thermodynamische Ungleichgewicht (endliche Verdampfungs- und Kondensationsrate) der Zweiphasenströmung nicht berücksichtigt. Außer dem Druck liefert die Gleichung auch den Dampfgehalt  $x$  abhängig von Dichte und innerer Energie:

$$x = x(\varrho, e) \quad (17)$$

Neben den Zustandsgleichungen selbst wird für das verwendete Lösungsverfahren auch die Schallgeschwindigkeit benötigt.

$$c^2 = \frac{\partial p}{\partial \rho} \Big|_{s=\text{const.}} \quad (18)$$

Auch die Schallgeschwindigkeit wird als Unterprogramm bereitgestellt, dadurch kann sie eine beliebige Funktion von Dichte und innerer Energie sein.

Ideales Gas [10]:

$$c^2 = \gamma(\gamma-1)e \quad (19)$$

Lineare Wassergleichung:

$$c^2 = \text{const.}$$

c wird dabei etwa zu 1000 m/s gewählt.

"Genaue" Wassergleichung:

Wie der Druck, wird auch die Schallgeschwindigkeit aus Tabellenwerten ermittelt. Das Programm H2OZUST liefert zusätzlich auch  $c^2(\rho, e)$ .

### 3.5 Ablauf des Integrationssschrittes

Die zeitliche Integration der Größen  $p, \rho, e, u$  und  $v$  um die Zeitschrittweite  $\Delta t$  ist in drei wesentliche Phasen gegliedert. In einer ersten expliziten Phase sind alle expliziten Berechnungen zusammengefaßt. Anhand der Kontinuitätsgleichung, der Impulsgleichung und der Zustandsgleichung werden explizit erste Schätzwerte für die Größen  $p, \rho, u$  und  $v$  zum neuen Zeitpunkt errechnet. In einer zweiten Iterationsphase werden mit einem Newton-Iterationsverfahren diese Größen unter Benutzung der gleichen Gleichungen so lange verändert, bis die Kontinuitätsgleichung zum neuen Zeitpunkt erfüllt ist.

Während der ersten zwei Phasen wird die Energie, die in die Zustandsgleichung eingeht, stets zum alten Zeitschritt betrachtet, während die Größen  $p, \rho, u$  und  $v$  am Ende der Iteration die Werte zum neuen Zeitpunkt besitzen. In einer dritten Phase wird mit Hilfe der Energiegleichung mit den neuen Werten von  $p, \rho, u$  und  $v$  die Energie zum neuen Zeitpunkt errechnet. Das Verfahren ist bezüglich der Energie explizit, hat jedoch nicht eine Courant-Stabilitätsbedingung der Form  $(|u|+c)\frac{\Delta t}{\Delta x} < 1$ , wie voll explizite Verfahren. Das Stabilitätskriterium für die hier benutzte Methode ist im wesentlichen gegeben durch  $|u|\frac{\Delta t}{\Delta x} < \frac{1}{2}$ . (Gl. 43 gibt die lineare Stabilitätsbedingung bei Berücksichtigung der Viskosität näherungsweise an.)

Zwischen der expliziten Phase und der Iteration erfolgt eine Überprüfung der Zeitschrittweite. Wird durch die gewählte Zeitschrittweite das Stabilitätskriterium verletzt, muß mit reduzierter Zeitschrittweite die explizite Phase wiederholt werden. In einer Akzeptiere-Phase nach der Iteration werden die alten Werte  $p, g, u$  und  $v$  durch die iterierten Werte zum neuen Zeitpunkt ersetzt.

Das hier angewandte Verfahren ist als ICE-(Implicit Continuous Eulerian-) Technik bekannt, es wird in ähnlicher Form auch in YAQUIR [2], KACHINA [3], DRIX-2D [11] benutzt.

Es folgt nun eine Beschreibung der einzelnen Phasen im Detail.

### Explizite Rechnung:

Die Schätzwerte zum neuen Zeitpunkt werden durch eine Tilde über dem Formelzeichen gekennzeichnet:  $\tilde{p}, \tilde{g}, \tilde{u}, \tilde{v}, (\tilde{g}\tilde{u}), (\tilde{g}\tilde{v})$ . Alle Größen ohne Angabe des Zeitpunktes sind zum Zeitpunkt  $n$  zu benutzen.

$\tilde{g}$  aus (9)

$$\begin{aligned} \tilde{g}_i^j = & g_i^j - \frac{\Delta t}{\Delta x_i} \left( \langle g u \rangle_{i+1/2}^j - \langle g u \rangle_{i-1/2}^j \right) \\ & - \frac{\Delta t}{\Delta y_i} \left( \langle g v \rangle_i^{j+1/2} - \langle g v \rangle_i^{j-1/2} \right) \end{aligned} \quad (20)$$

$\tilde{p}$ : Zustandsgleichung

$$\tilde{p}_i^j = p(\tilde{g}_i^j, e_i^j) \quad (21)$$

$\tilde{u}, \tilde{v}$ :

aus (13), (14) mit Abkürzungen  $\overline{gu}, \overline{gv}$ :

$$\begin{aligned}
 \overline{su}_{i+1/2}^j &= (su)_{i+1/2}^j - 2\Delta t \frac{\langle su^2 \rangle_{i+1}^j - \langle su^2 \rangle_i^j}{\Delta x_i + \Delta x_{i+1}} \\
 &\quad - \Delta t \frac{\langle svu \rangle_{i+1/2}^{j+1/2} - \langle svu \rangle_{i+1/2}^{j-1/2}}{\Delta y_j} \\
 &\quad + \frac{2\eta\Delta t}{\Delta x_i + \Delta x_{i+1}} \left( \frac{u_{i+3/2}^j - u_{i+1/2}^j}{\Delta x_{i+1}} - \frac{u_{i+1/2}^j - u_{i-1/2}^j}{\Delta x_i} \right) \\
 &\quad + \frac{2\eta\Delta t}{\Delta y_j} \left( \frac{u_{i+1/2}^{j+1} - u_{i+1/2}^j}{\Delta y_j + \Delta y_{j+1}} - \frac{u_{i+1/2}^j - u_{i+1/2}^{j-1}}{\Delta y_j + \Delta y_{j-1}} \right) \quad (22)
 \end{aligned}$$

$$\begin{aligned}
 \overline{sv}_i^{j+1/2} &= (sv)_i^{j+1/2} - \Delta t \frac{\langle svu \rangle_{i+1/2}^{j+1/2} - \langle svu \rangle_{i-1/2}^{j+1/2}}{\Delta x_i} \\
 &\quad - 2\Delta t \frac{\langle sv^2 \rangle_i^{j+1} - \langle sv^2 \rangle_i^j}{\Delta y_j + \Delta y_{j+1}} \\
 &\quad + \frac{2\eta\Delta t}{\Delta x_i} \left( \frac{v_{i+1}^{j+1/2} - v_i^{j+1/2}}{\Delta x_{i+1} + \Delta x_i} - \frac{v_i^{j+1/2} - v_{i-1}^{j+1/2}}{\Delta x_i + \Delta x_{i-1}} \right) \\
 &\quad + \frac{2\eta\Delta t}{\Delta y_{j+1} + \Delta y_j} \left( \frac{v_i^{j+3/2} - v_i^{j+1/2}}{\Delta y_{j+1}} - \frac{v_i^{j+1/2} - v_i^{j-1/2}}{\Delta y_j} \right) \quad (23)
 \end{aligned}$$

damit aus (13), (14):

$${}^{n+1}(su)_{i+1/2}^j = \overline{su}_{i+1/2}^j - 2 \Delta t \frac{{}^{n+1}p_{i+1}^j - {}^{n+1}p_i^j}{\Delta x_i + \Delta x_{i+1}} \quad (24)$$

$${}^{n+1}(sv)_i^{j+1/2} = \overline{sv}_i^{j+1/2} - 2 \Delta t \frac{{}^{n+1}p_i^{j+1} - {}^{n+1}p_i^j}{\Delta y_j + \Delta y_{j+1}} \quad (25)$$

daraus Anfangswert für  $\tilde{u}$ ,  $\tilde{v}$  mit:

$${}^{n+1}(su)_{i+1/2}^j \rightarrow \tilde{S}_{i+1/2}^j \cdot \tilde{u}_{i+1/2}^j$$

$${}^{n+1}(sv)_i^{j+1/2} \rightarrow \tilde{S}_i^{j+1/2} \cdot \tilde{v}_i^{j+1/2}$$

$${}^{n+1}p \rightarrow \tilde{p}$$

$$\tilde{u}_{i+1/2}^j = \frac{1}{\tilde{S}_{i+1/2}^j} \left\{ \overline{su}_{i+1/2}^j - 2 \Delta t \frac{\tilde{p}_{i+1}^j - \tilde{p}_i^j}{\Delta x_i + \Delta x_{i+1}} \right\} \quad (26)$$

$$\tilde{v}_i^{j+1/2} = \frac{1}{\tilde{S}_i^{j+1/2}} \left\{ \overline{sv}_i^{j+1/2} - 2 \Delta t \frac{\tilde{p}_i^{j+1} - \tilde{p}_i^j}{\Delta y_j + \Delta y_{j+1}} \right\} \quad (27)$$

$$\tilde{S}_{i+1/2}^j = \frac{\Delta x_{i+1} \tilde{S}_i^j + \Delta x_i \tilde{S}_{i+1}^j}{\Delta x_i + \Delta x_{i+1}} \quad (28)$$

$$\tilde{S}_i^{j+1/2} = \frac{\Delta y_{j+1} \tilde{S}_i^j + \Delta y_j \tilde{S}_i^{j+1}}{\Delta y_j + \Delta y_{j+1}} \quad (29)$$



Es fehlen noch die Donor-Cell-Ausdrücke für  $\langle \mathcal{S}u^2 \rangle$ ,  $\langle \mathcal{S}uv \rangle$ ,  $\langle \mathcal{S}uv^2 \rangle$ :

$$\langle \mathcal{S}u^2 \rangle_i^j = u_i^j \left\{ \mathcal{F}_1 (su)_{i-1/2}^j + (1 - \mathcal{F}_1) (su)_{i+1/2}^j \right\} \quad (30)$$

$$\mathcal{F}_1 = (1 - \alpha_0) \frac{\Delta x_i + \beta_0 \Delta t u_i^j}{2 \Delta x_i} + \frac{\alpha_0}{2} \left\{ \text{sign}(u_i^j) + 1 \right\} \quad (31)$$

$$u_i^j = \frac{1}{2} \left( u_{i+1/2}^j + u_{i-1/2}^j \right) \quad (32)$$

$$\langle \mathcal{S}vu \rangle_{i+1/2}^{j+1/2} = v_{i+1/2}^{j+1/2} \left\{ \mathcal{F}_2 (su)_{i+1/2}^j + (1 - \mathcal{F}_2) (su)_{i+1/2}^{j+1/2} \right\} \quad (33)$$

$$\mathcal{F}_2 = (1 - \alpha_0) \frac{\Delta y_{j+1} + \beta_0 \Delta t v_{i+1/2}^{j+1/2}}{\Delta y_{j+1} + \Delta y_j} + \frac{\alpha_0}{2} \left\{ \text{sign}(v_{i+1/2}^{j+1/2}) + 1 \right\} \quad (34)$$

$$v_{i+1/2}^{j+1/2} = \frac{\Delta x_{i+1} v_i^{j+1/2} + \Delta x_i v_{i+1}^{j+1/2}}{\Delta x_i + \Delta x_{i+1}} \quad (35)$$

$$\langle \mathcal{S}uv \rangle_{i+1/2}^{j+1/2} = u_{i+1/2}^{j+1/2} \left\{ \mathcal{F}_3 (sv)_i^{j+1/2} + (1 - \mathcal{F}_3) (sv)_{i+1}^{j+1/2} \right\} \quad (36)$$

$$\mathcal{F}_3 = (1 - \alpha_0) \frac{\Delta x_{i+1} + \beta_0 \Delta t u_{i+1/2}^{j+1/2}}{\Delta x_i + \Delta x_{i+1}} + \frac{\alpha_0}{2} \left\{ \text{sign}(u_{i+1/2}^{j+1/2}) + 1 \right\} \quad (37)$$

$$u_{i+1/2}^{j+1/2} = \frac{\Delta y_{j+1} u_{i+1/2}^j + \Delta y_j u_{i+1/2}^{j+1}}{\Delta y_j + \Delta y_{j+1}} \quad (38)$$

$$\langle \mathcal{S}v^2 \rangle_i^j = v_i^j \left\{ \mathcal{F}_4 (sv)_i^{j-1/2} + (1 - \mathcal{F}_4) (sv)_i^{j+1/2} \right\} \quad (39)$$

$$\mathcal{F}_4 = (1 - \alpha_0) \frac{\Delta y_j + \beta_0 \Delta t v_i^j}{2 \Delta y_j} + \frac{\alpha_0}{2} \left\{ \text{sign}(v_i^j) + 1 \right\} \quad (40)$$

$$V_i^j = \frac{1}{2} \left( V_i^{j+1/2} + V_i^{j-1/2} \right) \quad (41)$$

Es werden die Ausdrücke  $\langle \rho u^2 \rangle$  und  $\langle \rho v^2 \rangle$  also in der Maschenmitte berechnet, die gemischten Ableitungen  $\langle \rho uv \rangle$  und  $\langle \rho vu \rangle$  dagegen an den Kreuzungspunkten des Maschennetzes.

D: aus (9)

$$D_i^j = \frac{\tilde{S}_i^j - {}^n S_i^j}{\Delta t} + \frac{\langle \tilde{S} \tilde{u} \rangle_{i+1/2}^j - \langle \tilde{S} \tilde{u} \rangle_{i-1/2}^j}{\Delta x_i} + \frac{\langle \tilde{S} \tilde{v} \rangle_{i,j+1/2} - \langle \tilde{S} \tilde{v} \rangle_{i,j-1/2}}{\Delta y_j} \quad (42)$$

Automatische Zeitschrittkontrolle:

Die Stabilität des Verfahrens ist nur gewahrt [12], wenn gilt:

$$\Delta t \left\{ \frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + 2\eta \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right\} < 0.5 \quad (43)$$

für alle Maschen. Sind diese Beträge sehr klein, kann der Zeitschritt vergrößert werden. Zeitschrittkontrolle wird durch die Eingabegrößen  $\gamma_1$  und  $\gamma_2$  gesteuert:

$\Delta t$  halbieren, wenn

$$\Delta t \cdot \max_{i,j} \left\{ \frac{\max(|u_{i-\frac{1}{2}}^j|, |u_{i+\frac{1}{2}}^j|)}{\Delta x_i} + \frac{\max(|v_{i,j-\frac{1}{2}}|, |v_{i,j+\frac{1}{2}}|)}{\Delta y_j} + 2\eta \left( \frac{1}{\Delta x_i^2} + \frac{1}{\Delta y_j^2} \right) \right\} > \gamma_1 \quad (44)$$

$\Delta t$  verdoppeln, wenn

$$\Delta t \cdot \max_{i,j} \left\{ \frac{\max(|u_{i-\frac{1}{2}}^j|, |u_{i+\frac{1}{2}}^j|)}{\Delta x_i} + \frac{\max(|v_{i,j-\frac{1}{2}}|, |v_{i,j+\frac{1}{2}}|)}{\Delta y_j} + 2\eta \left( \frac{1}{\Delta x_i^2} + \frac{1}{\Delta y_j^2} \right) \right\} < \gamma_2 \quad (45)$$

Die Zeitschrittüberprüfung müßte eigentlich am Ende der Iteration erfolgen. Da sich jedoch die Geschwindigkeiten während der Iteration nicht mehr stark ändern, kann die Stabilitätskontrolle schon nach der expliziten Phase erfolgen. Somit muß im Falle einer Zeitschritthalbierung nur die explizite Phase mit dem kleineren Zeitschritt wiederholt werden.

Iteration:

Die Quellstärke  $D$  soll zu Null gemacht werden. Iterationsverfahren mit einem Überrelaxationsparameter  $\omega$ , der etwa gleich 1 ist (Eingabewert):

$$D(\tilde{p}) \stackrel{!}{=} 0$$

Schätzwert:  $\tilde{p}_0$

Dann

$$\tilde{p}_1 = \tilde{p}_0 - \omega \frac{D(\tilde{p}_0)}{D'(\tilde{p}_0)} \quad D' = \frac{\partial D}{\partial p} = \beta^{-1} \quad 0 < \omega < 2 \quad (46)$$

$$\beta^{-1} = \frac{\partial D}{\partial p} = \frac{\partial}{\partial p} \left( \frac{\partial s}{\partial t} \right) + \frac{\partial}{\partial p} (\nabla s \bar{u}) \quad (47)$$

in Differenzenform:

$$\beta^{-1} = \frac{1}{\Delta t} \frac{\partial s}{\partial p}(s, e) + \frac{1}{\Delta x_i} \frac{\partial}{\partial p} \left( \langle s u \rangle_{i+1/2}^j - \langle s u \rangle_{i-1/2}^j \right) \quad (48)$$

$$+ \frac{1}{\Delta y_j} \frac{\partial}{\partial p} \left( \langle s v \rangle_i^{j+1/2} - \langle s v \rangle_i^{j-1/2} \right)$$

$$\approx \frac{1}{\Delta t} \frac{\partial p}{\partial s}(\tilde{s}_{0i}^j, e_i^j) + \frac{1}{\Delta x_i} \frac{\partial}{\partial p} \left( {}^n (s u)_{i+1/2}^j - {}^n (s u)_{i-1/2}^j \right)$$

$$+ \frac{1}{\Delta y_j} \frac{\partial}{\partial p} \left( {}^n (s v)_i^{j+1/2} - {}^n (s v)_i^{j-1/2} \right) \quad (49)$$

$\frac{\partial p}{\partial s}$ : Schallgeschwindigkeit  $c^2(s, e)$ , gegebene Funktion

aus (13), (14):

$$\frac{\partial}{\partial p} (s u)_{i+1/2}^j = \frac{\partial}{\partial p_i^j} \left\{ \frac{-2 \Delta t}{\Delta x_i + \Delta x_{i+1}} \left( p_{i+1}^j - p_i^j \right) \right\} \quad (50)$$

$$= \frac{2 \Delta t}{\Delta x_i + \Delta x_{i+1}} \quad (51)$$

$$\frac{\partial}{\partial p} (su)_{i-1/2}^j = \frac{\partial}{\partial p_i^j} \left\{ \frac{-2 \Delta t}{\Delta x_{i-1} + \Delta x_i} (p_i^j - p_{i-1}^j) \right\} \quad (52)$$

$$= \frac{-2 \Delta t}{\Delta x_{i-1} + \Delta x_i} \quad (53)$$

$$\frac{\partial}{\partial p} (sv)_i^{j+1/2} = \frac{\partial}{\partial p_i^j} \left\{ \frac{-2 \Delta t}{\Delta y_j + \Delta y_{j+1}} (p_i^{j+1} - p_i^j) \right\} \quad (54)$$

$$= \frac{2 \Delta t}{\Delta y_j + \Delta y_{j+1}} \quad (55)$$

$$\frac{\partial}{\partial p} (sv)_i^{j-1/2} = \frac{\partial}{\partial p_i^j} \left\{ \frac{-2 \Delta t}{\Delta y_{j-1} + \Delta y_j} (p_i^j - p_i^{j-1}) \right\} \quad (56)$$

$$= \frac{-2 \Delta t}{\Delta y_{j-1} + \Delta y_j} \quad (57)$$

aus (49), (51), (53), (55) und (57):

$$\beta^{-1} = \frac{1}{\Delta t \cdot c^2 (\tilde{g}_i^j, e_i^j)} + \frac{2 \Delta t}{\Delta x_i} \left( \frac{1}{\Delta x_i + \Delta x_{i+1}} + \frac{1}{\Delta x_{i-1} + \Delta x_i} \right) + \frac{2 \Delta t}{\Delta y_j} \left( \frac{1}{\Delta y_j + \Delta y_{j+1}} + \frac{1}{\Delta y_{j-1} + \Delta y_j} \right) \quad (58)$$

Da ein kleiner Fehler in  $\beta$  auf das Iterationsergebnis keinen Einfluß hat (allenfalls auf die Konvergenzrate), ist die Annäherung von Gl. (48) durch Gl. (49) zulässig. Außerdem braucht  $\beta$  nur einmal vor dem Start der Iteration bestimmt werden.

Neuer Wert für  $\tilde{p}$ :

$$\delta \tilde{p}_i^j = -\omega \beta D_i^j \quad (59)$$

$$\text{neu } \tilde{p}_i^j = \text{alt } \tilde{p}_i^j + \delta \tilde{p}_i^j \quad (60)$$

Neuer Wert für  $\tilde{g}$ :

$$\delta \tilde{g}_i^j = \frac{1}{c^2(\tilde{g}_i^j, e_i^j)} \cdot \delta \tilde{p}_i^j \quad (61)$$

$$\text{neu } \tilde{g}_i^j = \text{alt } \tilde{g}_i^j + \delta \tilde{g}_i^j \quad (62)$$

Neuer Wert für  $(\tilde{g}\tilde{u})$ ,  $(\tilde{g}\tilde{v})$ :

$$\delta (\tilde{g}\tilde{u})_{i+1/2}^j = \frac{2\Delta t}{\Delta x_i + \Delta x_{i+1}} (\delta \tilde{p}_i^j - \delta \tilde{p}_{i+1}^j) \quad (63)$$

$$\text{neu } (\tilde{g}\tilde{u})_{i+1/2}^j = \text{alt } (\tilde{g}\tilde{u})_{i+1/2}^j + \delta (\tilde{g}\tilde{u})_{i+1/2}^j \quad (64)$$

$$\delta (\tilde{g}\tilde{v})_i^{j+1/2} = \frac{2\Delta t}{\Delta y_j + \Delta y_{j+1}} (\delta \tilde{p}_i^j - \delta \tilde{p}_i^{j+1}) \quad (65)$$

$$\text{neu } (\tilde{g}\tilde{v})_i^{j+1/2} = \text{alt } (\tilde{g}\tilde{v})_i^{j+1/2} + \delta (\tilde{g}\tilde{v})_i^{j+1/2} \quad (66)$$

Neuer Wert für  $\tilde{u}, \tilde{v}$ :

$$\text{neu } \tilde{u}_{i+1/2}^j = \frac{\text{neu } (\tilde{g}\tilde{u})_{i+1/2}^j}{\text{neu } \tilde{g}_{i+1/2}^j} \quad (67)$$

$$= \frac{\text{neu}(\tilde{\rho} \tilde{u})_{i+1/2}^j (\Delta X_i + \Delta X_{i+1})}{\Delta X_i \text{neu} \tilde{\rho}_{i+1}^j + \Delta X_{i+1} \text{neu} \tilde{\rho}_i^j} \quad (68)$$

$$\text{neu} \tilde{v}_{j+1/2}^j = \frac{\text{neu}(\tilde{\rho} \tilde{v})_i^{j+1/2}}{\text{neu} \tilde{\rho}_i^{j+1/2}} \quad (69)$$

$$= \frac{\text{neu}(\tilde{\rho} \tilde{v})_i^{j+1/2} (\Delta y_j + \Delta y_{j+1})}{\Delta y_j \text{neu} \tilde{\rho}_i^{j+1} + \Delta y_{j+1} \text{neu} \tilde{\rho}_i^j} \quad (70)$$

Neues D:

$$\text{neu } D_i^j = \frac{\text{neu} \tilde{\rho}_i^j - \text{neu} \tilde{\rho}_i^j}{\Delta t} + \frac{\text{neu} \langle \tilde{\rho} \tilde{u} \rangle_{i+1/2}^j - \text{neu} \langle \tilde{\rho} \tilde{u} \rangle_{i-1/2}^j}{\Delta X_i} + \frac{\text{neu} \langle \tilde{\rho} \tilde{v} \rangle_i^{j+1/2} - \text{neu} \langle \tilde{\rho} \tilde{v} \rangle_i^{j-1/2}}{\Delta y_j} \quad (71)$$

Abbruchkriterium für die Iteration:

$$\text{ende} = |D_i^j| \leq \varepsilon \cdot \frac{\tilde{\rho}_i^j \max}{\Delta t} \quad \& \quad |\delta \tilde{p}_i^j| \leq \varepsilon \cdot \tilde{p}_i^j \quad (72)$$

für alle Maschen

oder: iter  $\geq$  maxiter, Erreichen einer maximalen Iterationszahl

Da der Überrelaxationsparameter stets  $\approx 1$  ist, wurde er nicht in das Abbruchkriterium einbezogen.

Explizite Rechnung der Energie:

nach Gl. (10) mit den neuen Werten von u, v, p und  $\rho$ .

Übernehmen der Tilde-Werte als neue Werte für den folgenden Zeitschritt:

$${}^{n+1} \rho_i^j = \tilde{\rho}_i^j \quad (73)$$

$${}^{n+1} p_i^j = \tilde{p}_i^j \quad (74)$$

$${}^{n+1} (\rho u)_{i+1/2}^j = (\tilde{\rho} \tilde{u})_{i+1/2}^j \quad (75)$$

$${}^{n+1} (g v)_i^{j+1/2} = (\tilde{g} \tilde{v})_i^{j+1/2} \quad (76)$$

$${}^{n+1} u_{i+1/2}^j = \tilde{u}_{i+1/2}^j \quad (77)$$

$${}^{n+1} v_i^{j+1/2} = \tilde{v}_i^{j+1/2} \quad (78)$$

#### 4. Geometrie und Topologie

##### 4.1 Verbindung von Fluidbereichen

In vielen Finite-Differenzen-Programmen ist der Kontrollbereich von fiktiven Randmaschen umgeben. Durch Vorgeben der Werte dieser Randmaschen können Randbedingungen erfüllt werden. So kann eine Tangentialgeschwindigkeit Null am Rand dadurch erreicht werden, daß die Geschwindigkeit eine halbe Maschenweite außen auf den negativen Wert der Tangentialgeschwindigkeit eine halbe Maschenweite innen gesetzt wird. Ein vorgegebener Druck am Rand wird durch Setzen des Randmaschendruckes auf den gewünschten Wert erreicht. Soll an einem derartigen Rand ein zweiter Fluidbereich angeschlossen werden, so muß sichergestellt sein, daß stets die Randmaschenwerte des einen Fluids gleich den Werten der letzten Masche innerhalb des anderen Fluids sind (s. Abb. 5). Die Geschwindigkeiten und Impulse senkrecht zur Kontaktfläche der beiden Fluidbereiche können dann auf beiden Fluidumrandungen berechnet werden. Um die doppelte Berechnung der Kontaktflächengrößen zu vermeiden, wurden die Randmaschen von den beiden Fluiden gelöst und zu einer neuen Komponente, einem Rand, zusammengefaßt (Abb. 6).

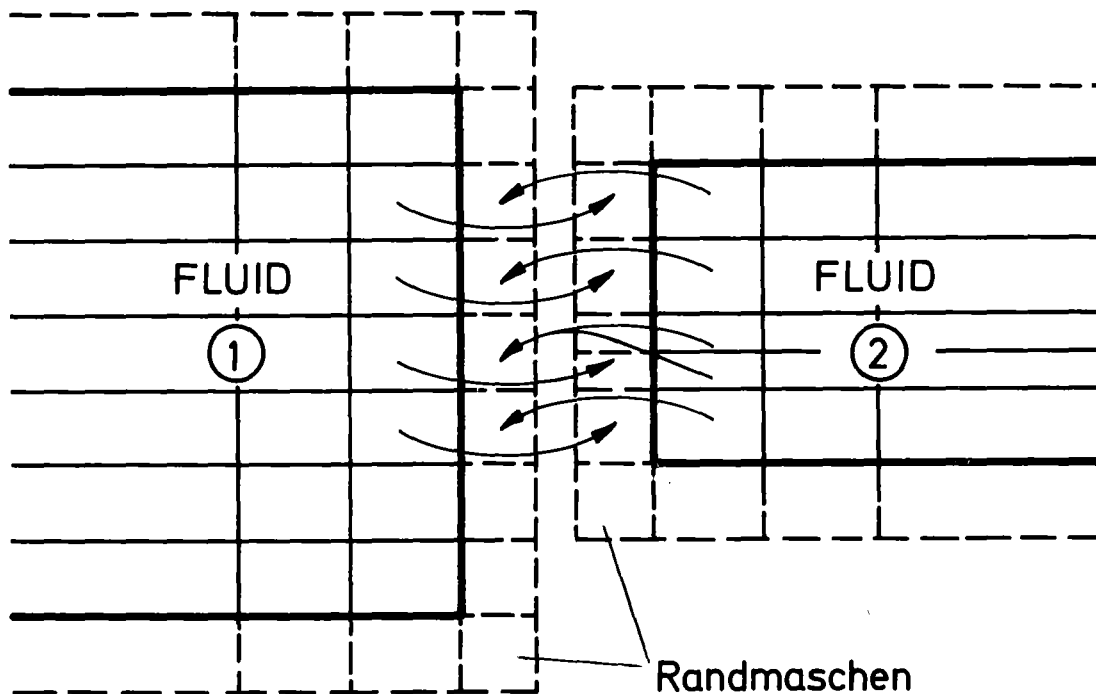


Abb. 5: Verbindung zweier Fluide mit fiktiven Randmaschen



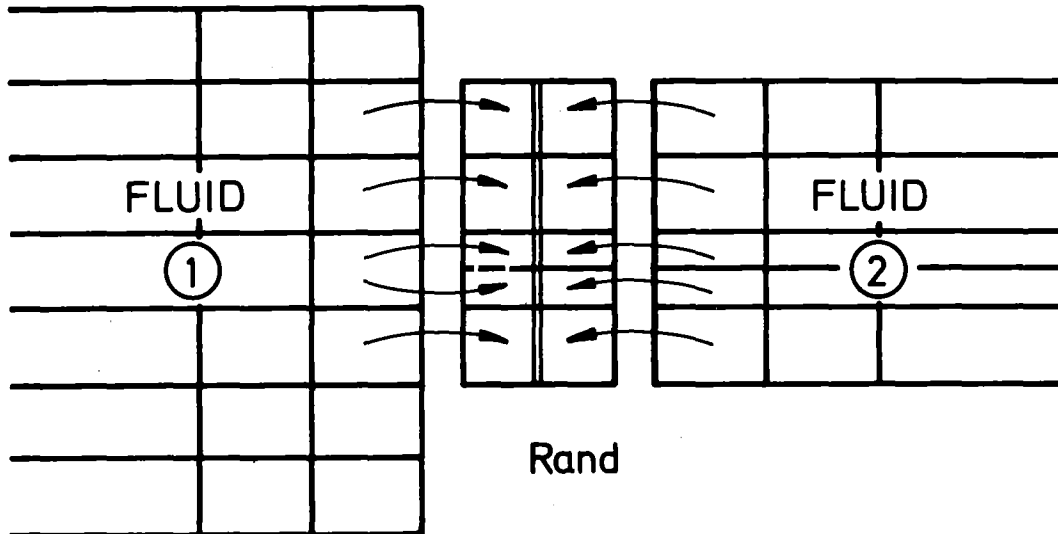


Abb. 6: Rand zwischen zwei Fluiden

Ein Rand enthält die Werte der angrenzenden Fluide. Alle Größen, deren Berechnung die Kenntnis der Werte von zwei verschiedenen Fluiden voraussetzt, werden auf dem Rand gerechnet. Es sind dies nicht nur die Normalgeschwindigkeiten und -impulse auf der Grenzfläche, sondern auch die Tangentialgeschwindigkeiten  $u$  (waagerechter Rand) und  $v$  (senkrechter Rand) und gemischte Ableitungen  $\langle \rho u v \rangle$  und  $\langle \rho v u \rangle$ .

An Stellen des Modells, wo mehrere Ränder aufeinanderstoßen, z.B. am Berührungspunkt von 3 oder 4 Fluiden, erfordert die Berechnung bestimmter Größen die Kenntnis von Werten von mehr als zwei Fluiden. Um diese Größen einfach berechnen zu können, wurde als Verbindung von Rändern Komponenten in die Modellstruktur eingefügt, die die Werte aller an sie anstoßenden Ränder enthalten. Damit enthalten sie auch die Werte der angrenzenden Fluide. Die Komponenten zur Verbindung von Rändern heißen Nähte. Auf Nähten werden z.B. die Geschwindigkeiten  $\tilde{u}$ ,  $\tilde{v}$  und die gemischten Ableitungen  $\langle \rho u v \rangle$  und  $\langle \rho v u \rangle$  errechnet.

Abb. 7: zeigt, welche Werte zur Berechnung von  $\langle \rho u v \rangle$  am Berührungspunkt der Fluide 1, 2 und 3 erforderlich sind. Diese Werte werden vor der Berechnung

auf die Naht übertragen. Die Größe  $\langle p_{uv} \rangle$  kann dann auf alle angrenzenden Fluide zurückgespeichert werden, so daß sie dort für die Berechnung der Geschwindigkeit  $\tilde{u}$ ,  $\tilde{v}$  zur Verfügung steht.

Nachdem die Verwendung von Rändern und Nähten verständlich gemacht wurde, soll im folgenden Abschnitt zunächst die zur Beschreibung der Modellgeometrie und -topologie benutzte Terminologie erläutert werden.

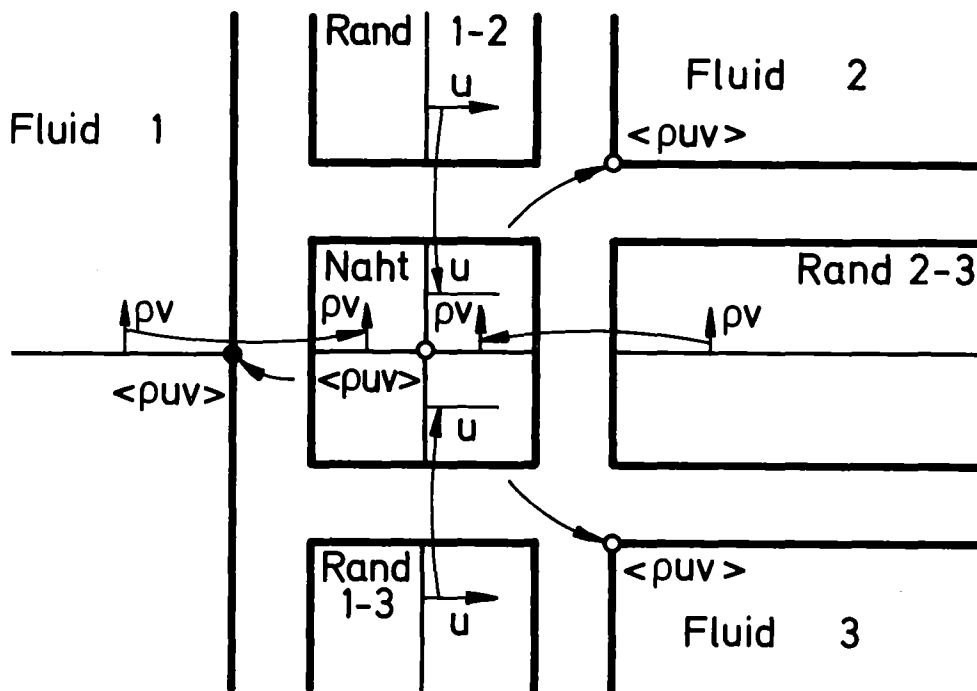


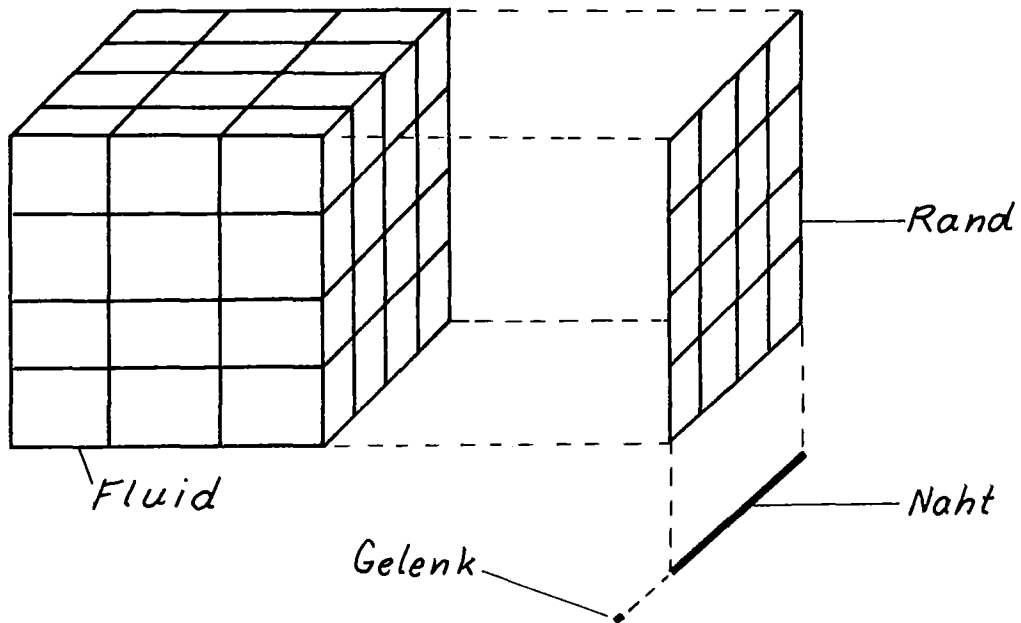
Abb. 7: Berechnung von  $\langle p_{uv} \rangle$  auf der Naht

#### 4.2 Terminologie

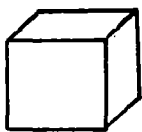
Da FLUST auch dreidimensionale Modelle aufnehmen soll, orientiert sich die Wortwahl für die einzelnen Modellbestandteile am dreidimensionalen Fall, um die Konsistenz in einem Gesamtsystem zu wahren. Die zweidimensionalen Bereiche werden so bezeichnet, als wären sie dreidimensionale Bereiche mit konstanter Ausdehnung in z-Richtung. Daher heißt z.B. die rechte Begrenzung einer Masche nicht Kante, sondern Fläche.

Abb. 8 zeigt die 3D-Komponenten, Abb. 9 die Einzelteile eines 2D-Modells.

Gesamtbereiche:



Einzelteile:



Masche



Fläche

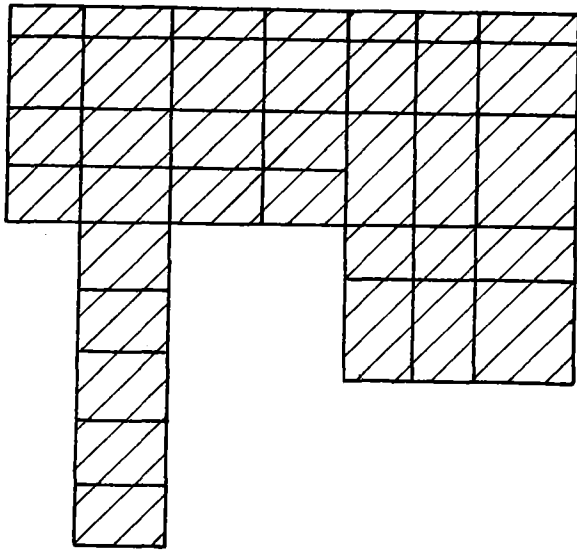


Kante



Knoten

Abb. 8: Teile eines 3D-Modells



Modell

Kontrollraum schraffiert. Ein Modell wird aufgebaut aus rechteckigen Fluidbereichen, zwischen ihnen befinden sich Ränder. Das Fluid wird nach außen durch Außenränder begrenzt. Ränder werden durch Nähte verbunden.

Aufteilung in Einzelbereiche:

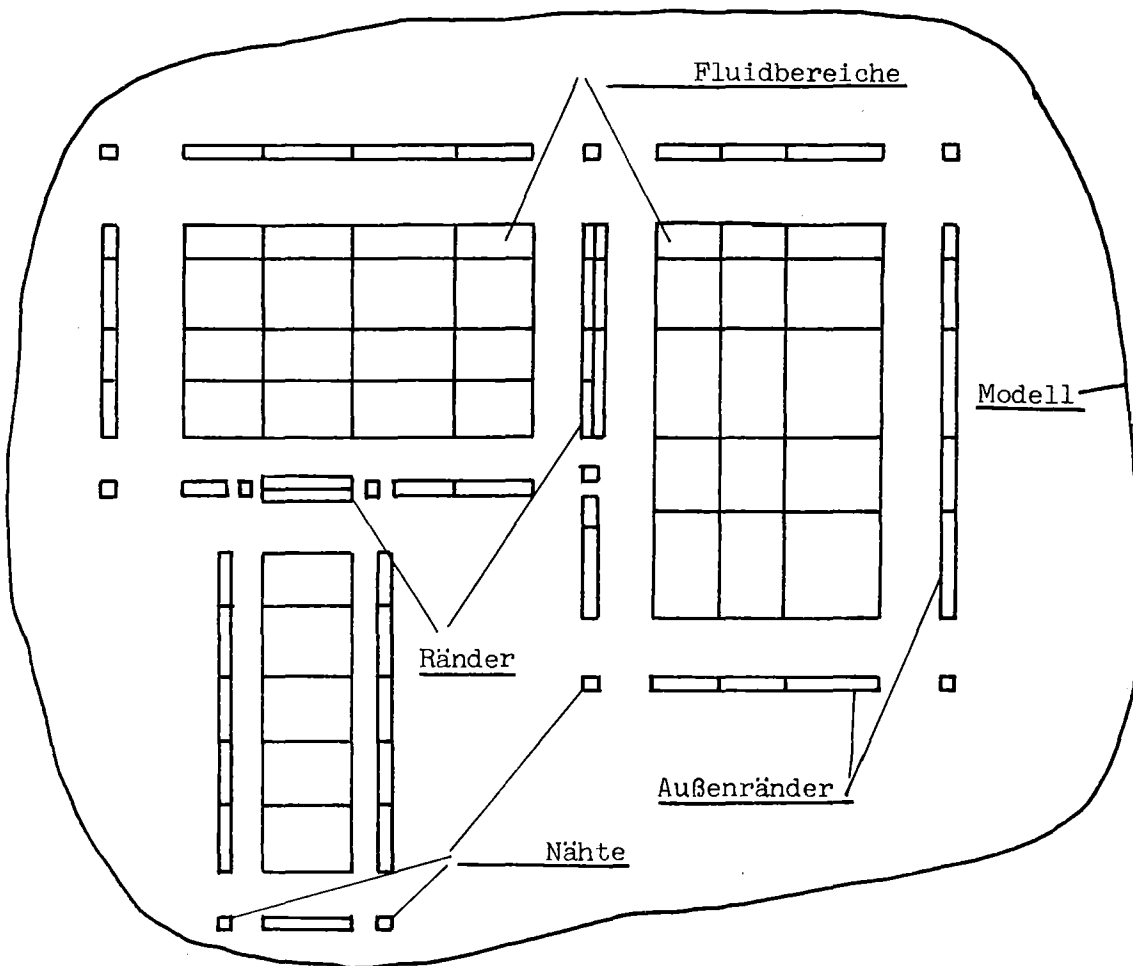


Abb. 9: Teile eines 2D-Modells

Tabelle der Bezeichnungen:

<u>deutsch</u>	<u>englisch</u>
Modell	model
Fluid(bereich)	fluid
Masche	mesh
Fläche	face
Kante	edge
Knoten	node
Rand	boundary
Naht	joint
Außenrand	exterior boundary

4.3 Ränder und Nähte

Ränder zwischen Fluiden enthalten auf ihren beiden Seiten die Werte der angrenzenden zwei Fluide, die für die Berechnungen auf dem Rand erforderlich sind. Abb.10 zeigt das Beispiel eines senkrechten Randes. Die Pfeile deuten die Werte an, die vom Fluid zum Rand oder umgekehrt übertragen werden können. Die Randmasche mit dem Index  $i_2$  enthält auf der linken Seite  $(i_2,1)$  die Werte der Fluidmasche mit den Indizes  $(i_1, j_1)$ . Die rechte Seite  $(i_2,2)$  enthält die Werte aus der Masche  $(i_3, j_3)$  des rechten Fluides. Sowohl die Werte auf der Maschenmitte  $(p_i^j, \rho_i^j, \text{etc.})$  als auch die auf den Maschenflächen senkrecht stehenden Werte  $(u_{i+\frac{1}{2}}^j, v_i^{j+\frac{1}{2}}, \rho u_{i+\frac{1}{2}}^j, \text{etc.})$  und

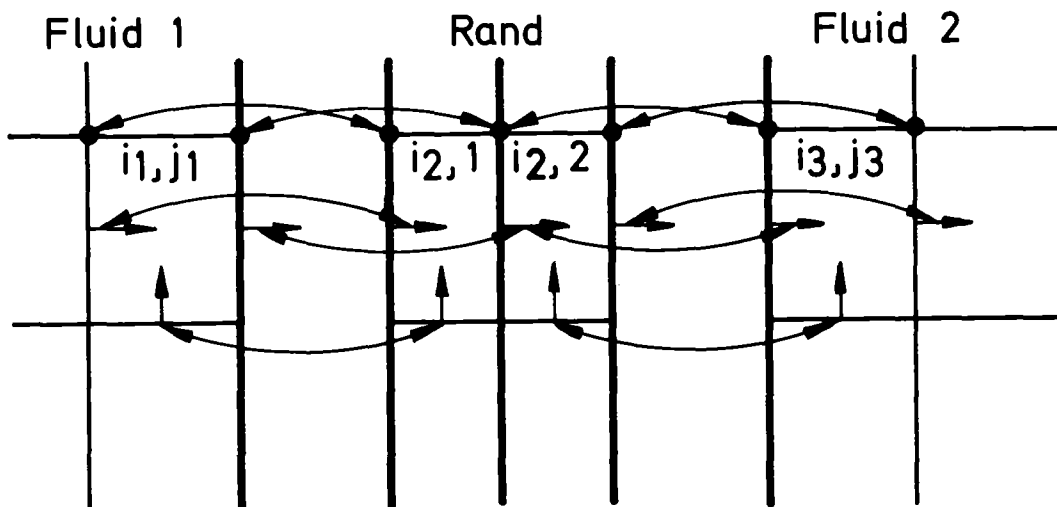


Abb.10: Fluidwerte auf einem Rand

die Werte  $\langle \rho v \rangle_i^j$  auf den Kanten werden auf dem Rand benötigt. Wie in Abb. 10 gezeigt, müssen auch einige Werte der Maschenflächen und -kanten eine Maschenweite innerhalb des Fluids auf dem Rand bekannt sein. Die Größen, die im Fluid in der zweiten Reihe vom Rand entfernt liegen, werden mit dem tiefgestellten Zeichen 'in' gekennzeichnet, z.B.  $u_{in}$ ,  $\langle \rho v \rangle_{in}$ .

Diejenigen Größen, die auf dem Rand errechnet werden, z.B. die Geschwindigkeiten auf der Grenzfläche zweier Fluide, werden anschließend in die Fluide zurückübertragen. Werte, die im Fluid bekannt sind, müssen vor ihrer Verwendung auf dem Rand vom Fluid übertragen werden. Bei einem Übergang von mehreren Maschen zu einer am Rand muß beim Übertragen der Werte der Durchschnitt gebildet werden.

Außenränder sind ebenso aufgebaut wie Ränder zwischen Fluiden, statt der Übertragung von Werten aus einem Fluid werden bei ihnen die Werte der Außenmaschen entsprechend den Randbedingungen gesetzt (s. Kap. 3.3).

Nähte enthalten die Werte der umgebenden Maschen. Die Werte werden von den Rändern auf die Nähte übertragen, wenn sie dort benötigt werden. Auf den Nähten errechnete Größen werden zurück auf den Rand übertragen und von dort ins Fluid. Für alle Übertragungsvorgänge müssen bei den Außenmaschen der Fluide Verweise auf die zugehörigen Ränder und an den Enden der Ränder Verweise auf die zugehörigen Nähte vorhanden sein.

Die folgende Abb. 11 zeigt die wesentlichen Vorgänge beim Berechnen der Geschwindigkeiten  $\tilde{u}$  und  $\tilde{v}$  während der expliziten Phase. Zunächst erfolgt die Bereitstellung der Werte  $\rho$  und  $\langle \rho u^2 \rangle$  aus dem Fluid, danach wird  $u$  und  $v$  auf der Naht gerechnet. Gleichzeitig zur Berechnung auf dem Rand werden an Außenrändern die Randbedingungen eingesetzt. Zuletzt werden die Geschwindigkeiten im Fluid berechnet.

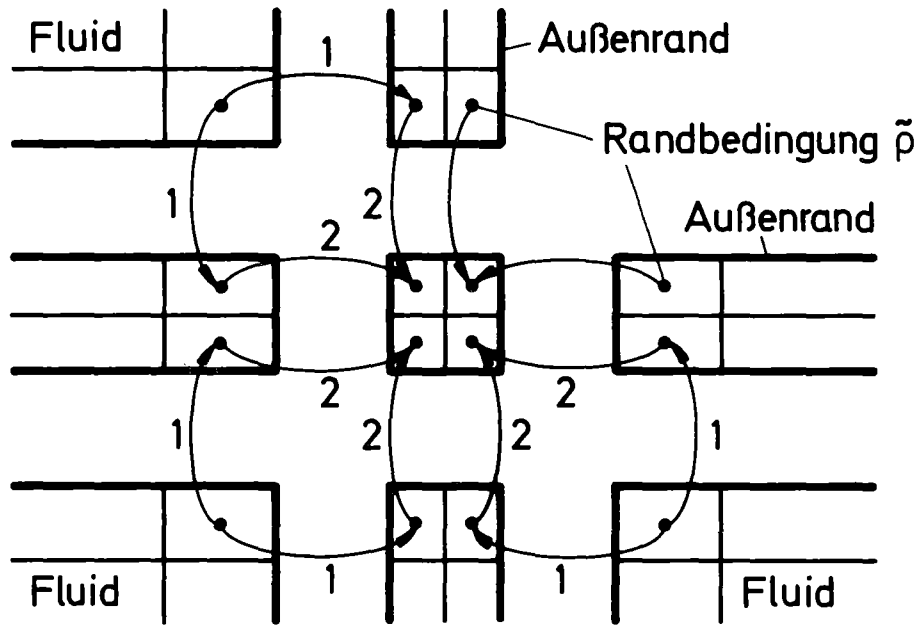


Abb. 11a) Übertragen  $\tilde{p}$ ,  $\langle \rho u^2 \rangle$  aus dem Fluid auf Ränder und Nähte

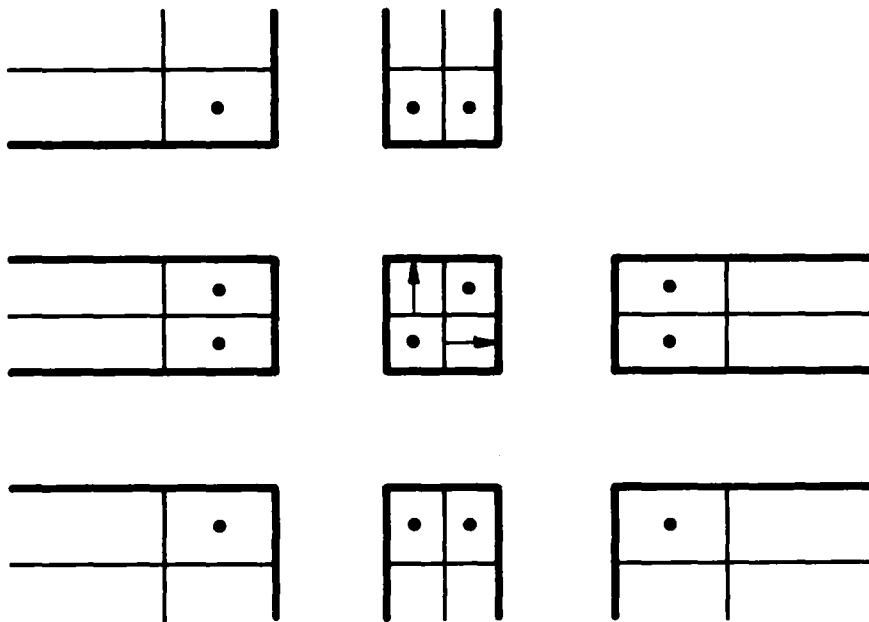


Abb. 11b) Rechnen von  $\tilde{u}$  und  $\tilde{v}$  auf der Naht

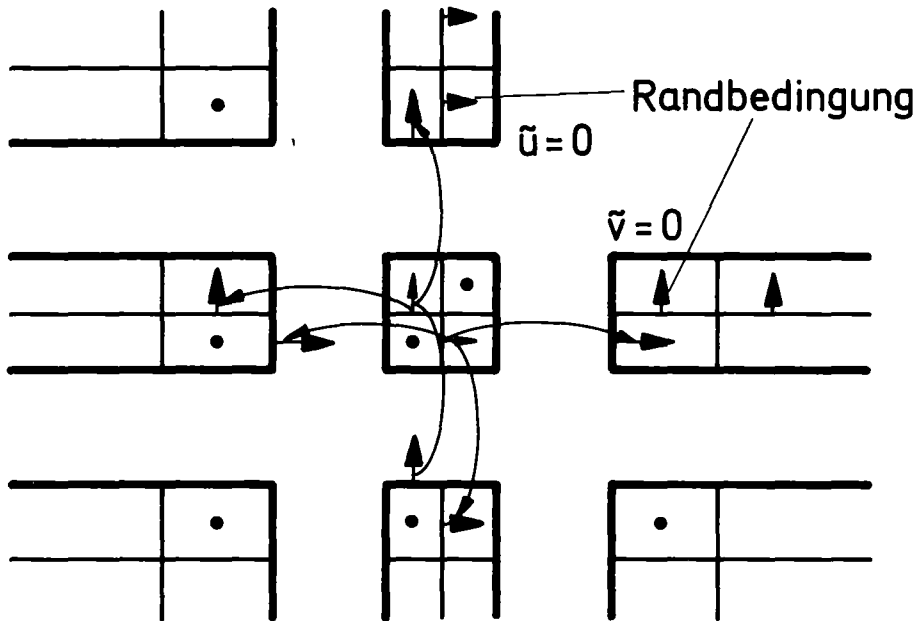


Abb. 11c) Übertragen  $u, v$  von den Nähten auf die Ränder und Randbedingung für  $u, v$  auf den Außenrändern (Pfeile kennzeichnen neu gesetzte Größen)

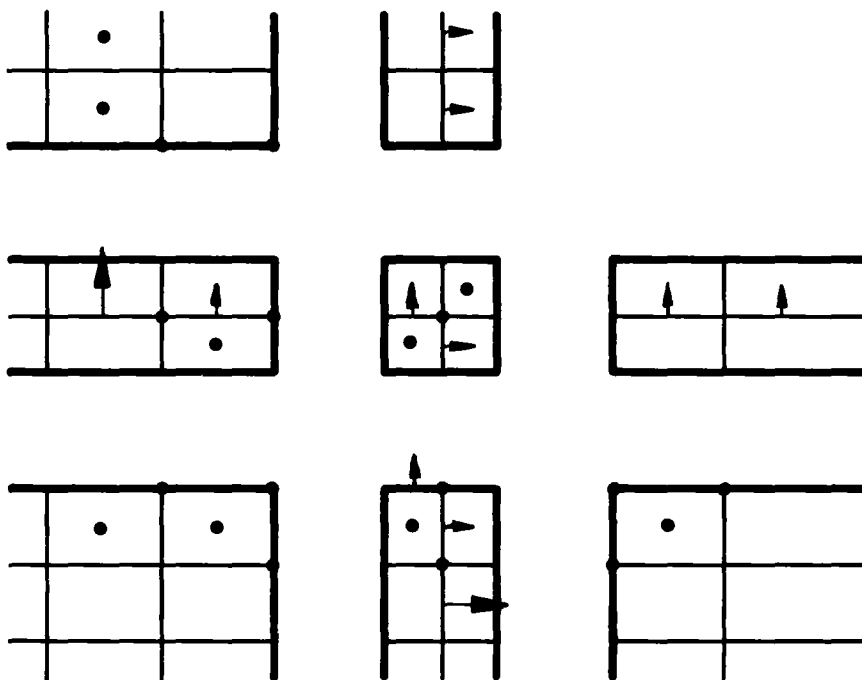


Abb. 11d) Berechnen von  $\tilde{u}$  und  $\tilde{v}$  auf den Rändern



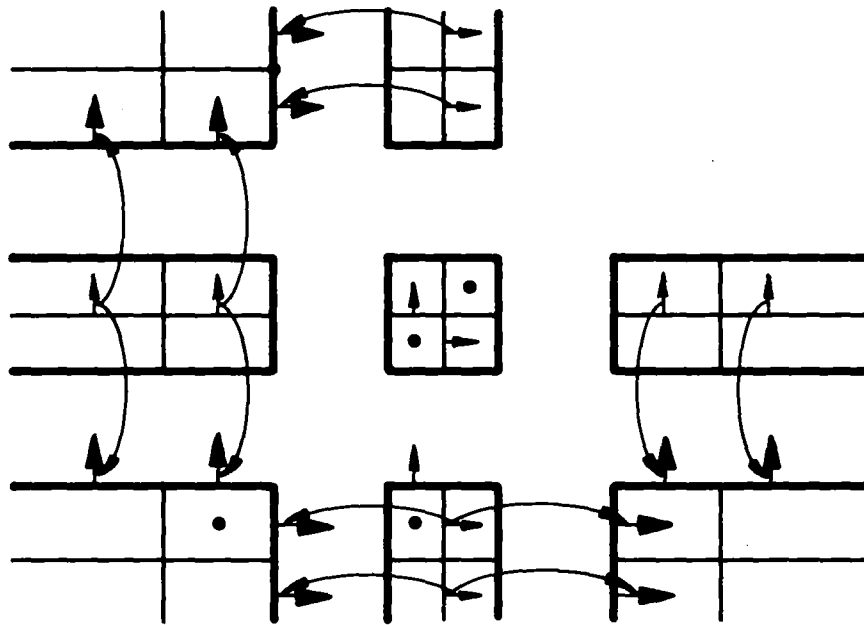


Abb. 11e) Holen der Werte von den Rändern ins Fluid

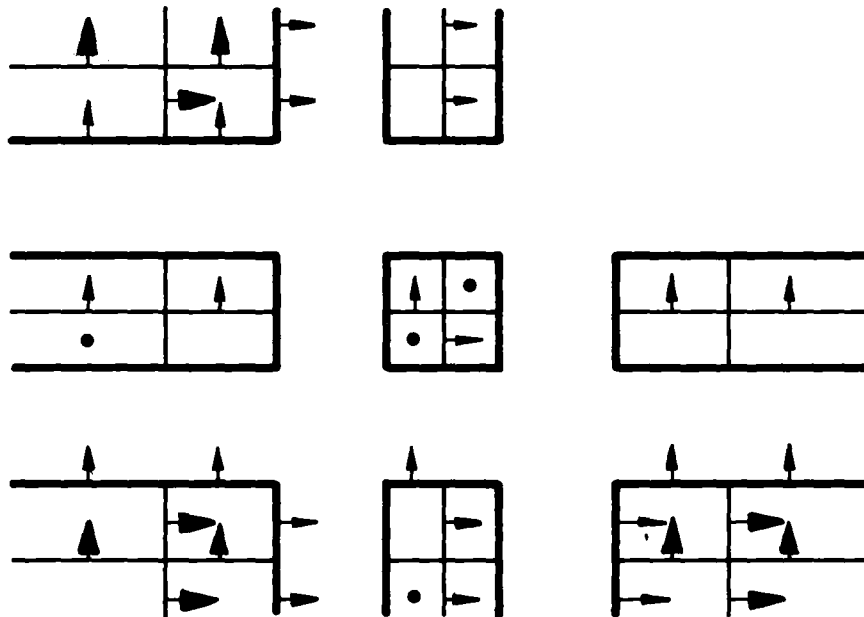


Abb. 11f) Berechnen der Werte im Fluid

Abb. 11: Beispiel zur Berechnung von Werten auf Nähten und Rändern

#### 4.4 Geometrische Angaben

Die Angaben zur Modellgeometrie beschreiben die Größe der einzelnen Komponenten, ihre Form und ihre Lage im Raum. Für jede Komponente wird ein lokales Koordinatensystem definiert. Die Lage dieses lokalen Koordinatensystems in einem globalen Koordinatensystem wird durch einen Vektor zum Ursprung und zwei Richtungsvektoren festgelegt, siehe Abb. 12. Die beiden Richtungen sind die ersten beiden Basisvektoren des lokalen Koordinatensystems, der dritte Basisvektor steht rechtsdrehend senkrecht auf ihnen. Die Art des lokalen Koordinatensystems (kartesische, zylindrische oder Kugelkoordinaten) wird durch eine Kennzeichnung vermerkt. Auch die Dimensionalität des Modells wird festgehalten.

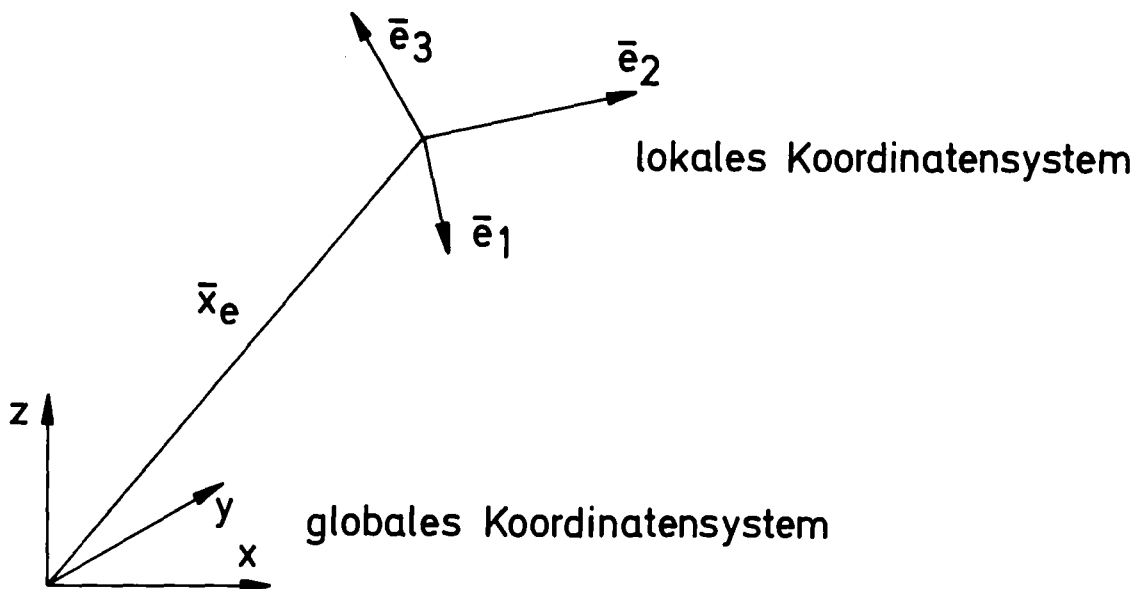


Abb. 12: Lage des lokalen Koordinatensystems

Im lokalen Koordinatensystem wird die Größe der Komponente und die Maschenweiten beschrieben. Das globale System ist für alle Komponenten kartesisch und dreidimensional.

Die Geometrischen Angaben sind unabhängig vom verwendeten numerischen Modell, mit ihrer Hilfe kann die Modellgeometrie, z.B. graphisch, dargestellt werden. Für FLUST-2D sind die lokalen Koordinatensysteme kartesisch und zweidimensional,  $\bar{e}_1$  und  $\bar{e}_2$  sind die Richtungen der x- und y-Achsen. Die lokalen Systeme können dabei um Vielfache von  $90^\circ$  gegenüber dem globalen gedreht sein.

#### 4.5. Topologische Angaben

Die Modelltopologie beschreibt die Nachbarschaftsverhältnisse der Komponenten untereinander. Zur Übertragung von Werten muß bekannt sein, welche Maschenindizes in welcher Komponente zur Nachbarmasche gehören. Es wird hier auf die in [13] beschriebenen Grundlagen aufgebaut. Dabei werden in jeder Komponente folgende Angaben über die Nachbarn vermerkt (s. Abb. 13):

- Indifikation der Nachbarkomponenten,
- Maschen-Indizes der ersten Nachbarmaschen (IREF)
- Zusammenhang der Zählrichtungen der Indizes (LZ)
- Angaben über Änderung der Maschenweiten

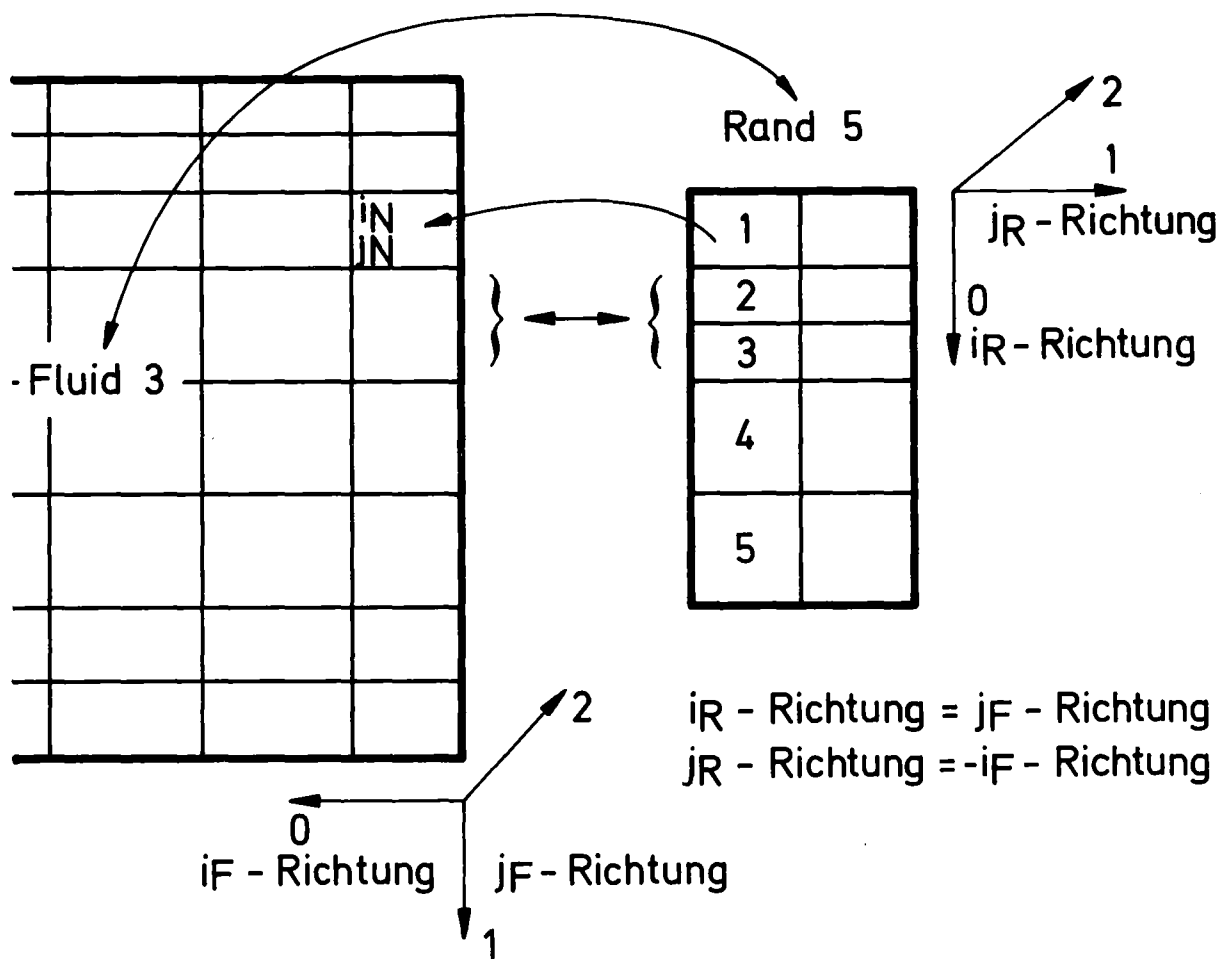


Abb. 13: Topologische Angaben

Die Identifikation des angrenzenden Randes wird in jeder Außenmasche eines Fluides gespeichert. Da ein Rand auf jeder Seite nur je ein Fluid als Nachbar haben kann, enthält er nur für jede Seite eine Fluidkennzeichnung. Ein Rand enthält außerdem die Identifikation der zwei angrenzenden Nähte, eine Naht die der vier angrenzenden Ränder.

Die Angaben über Maschen-Indizes, Zählrichtung und Maschenweitenänderung sind nur auf dem Rand für das Nachbarschaftsverhältnis Fluid-Rand und auf der Naht für die Nachbarschaft Rand-Naht festgehalten. Sie sind erforderlich, um jeder Masche einer Komponente die richtige Nachbarmasche zuzuordnen zu können. Die Angabe der Nachbar-Indizes zur ersten Masche und eine Aussage darüber, welche Zählrichtungen der Nachbarkomponente zu den Zählrichtungen der betrachteten Komponente gehören, reichen für die Zuordnung aller Maschen aus.

Die Zählrichtungen der Maschen in Richtung der drei Basisvektoren des Koordinatensystems werden als 0-, 1- und 2-Richtung bezeichnet. Die Angabe  $LZ = (1, -0, 2)$  auf einem Rand als Zählrichtung des Fluids besagt, daß der 0-Richtung auf dem Rand die 1-Richtung im Fluid entspricht, der 1-Richtung auf dem Rand die negative 0-Richtung im Fluid und der 2-Richtung auch die 2-Richtung im Fluid (s. Abb. 13).

Vorzeichenbehaftete Größen, wie Geschwindigkeit und Impuls, werden im Maschennetz dann positiv gerechnet, wenn ihre Richtung in Zählrichtung des Maschennetzes zeigt. Wechselt beim Übergang von einem Maschennetz zum andern die Zählrichtung, muß auch das Vorzeichen dieser Größen geändert werden. Eine in Abb. 13 nach rechts gerichtete Geschwindigkeit ist z.B. im Fluid negativ, auf dem Rand dagegen positiv. Sind einer Masche einer Komponente mehrere Maschen einer Nachbarkomponente zugeordnet, muß dies bei der Maschenzuordnung berücksichtigt werden. Maschenweitenänderungen werden daher auf den Rändern vermerkt, da nur auf ihnen Wechsel des Maschennetzes zulässig sind.

## 5. Programmaufbau

### 5.1 Programmstruktur

Entsprechend der in Abschnitt 3.5 beschriebenen Phasen beim Ablauf der Lösung ist das FLUST-Programm in die fünf Hauptmodule

EXPLIC	-	Explizite Phase
TIME	-	Zeitschrittkontrolle
ITERAT	-	Iteration
ACCEPT	-	Neue Werte $q$ = iterierte Werte $\tilde{q}$
ENERGY	-	Berechnung der inneren Energie

gegliedert. Dazu kommt ein Modul

SETUP	-	für die Initialisierung,
FPRINT	-	zum Ausdrucken von Werten
FSAVE	-	zum Retten von Werten auf Band und
ENDE	-	Abschluß der Rechnung.

Alle diese Module werden vom Steuermodul CONTRL gerufen, der durch die Steueranweisung "SOLVE;" aus dem Anwender-Hauptprogramm gestartet wird. Der Modul CONTRL ist folgendermaßen aufgebaut:

```
CALL SETUP;
DO WHILE(¬stop);
  IF print THEN CALL FPRINT;
  IF save THEN CALL FSAVE;
  CALL EXPLIC;
  CALL TIME;
  IF ¬timestephalf THEN DO;
    CALL ITERAT;
    CALL ACCEPT;
    CALL ENERGY;
    cycle = cycle + 1;
    t      = t      + Δt;
  END;
  IF timestepdouble THEN Δt = 2 · Δt;
END;
CALL ENDE;
```

Die Bedingungen "stop", "print" und "save" werden durch Programm-Steueranweisungen gesetzt. "timestephalf" und "timestepdouble" werden in dem Modul

TIME anhand von Gleichung (44) und (45) errechnet.

Die bisher aufgeführten Module werden für alle Modelle, 2- oder 3-dimensional, unabhängig vom Formelsatz, verwendet. Innerhalb dieser Module werden zur Berechnung der Feldgrößen in Durchgängen über alle Fluide, Ränder oder Nähte modellspezifische Module aufgerufen. Abb. 14 zeigt die Aufrufhierarchie. Während der Initialisierungsphase wird in einem Durchgang durch das Modell der Datenstruktur jeder Komponente eine aus zwei Buchstaben bestehende Kennzeichnung  $\alpha$  entnommen. Aus dieser Kennzeichnung und einem die Teilrechnung identifizierenden Namen wird ein Programmname zusammengesetzt. Das Programm (der Modul) dieses Namens berechnet die betreffende Feldgröße für die Komponente mit der Kennzeichnung  $\alpha$ . Elf Module für Fluide ( $\alpha$  F1 bis  $\alpha$  F11), zehn für Ränder ( $\alpha$  R1 bis  $\alpha$  R6 und  $\alpha$  R8 bis  $\alpha$  R11) und drei für Nähte ( $\alpha$  N3,  $\alpha$  N4,  $\alpha$  N6) müssen für jede Modellkennzeichnung  $\alpha$  vorhanden sein. Das FLUST-2D-Modell hat die Kennzeichnung AA, die Module heißen demnach AAF1, AAR1, AAN3 usw. Während der Initialisierung (in SETUP) werden die Modulnamen in einer Matrix ENTRIES (nk, nid) gespeichert. "nk" ist die Anzahl der Komponenten, "nid" die Anzahl der Modelle. Beim Durchgang durch das Modell wird aus dieser Matrix für jede Teilrechnung das richtige Programm ausgewählt und aufgerufen. Der Modulaufruf erfolgt dynamisch, d.h., wenn erforderlich, von einer Bibliothek. Es ist jedoch möglich, alle Module oder eine Teilgruppe vor Beginn der Rechnung in den Arbeitsspeicher zu laden und dort während dem ganzen Lösungsablauf festzulegen. Dann enthält die Matrix ENTRIES die Anfangsadressen (Entry-Points) der Module, ihr Aufruf kann dann sehr effektiv erfolgen, jedoch mit höherem Arbeitsspeicherbedarf.

Außer den "Rechenmoduln"  $\alpha$  F1 usw. benötigen die Hauptmodule noch die drei Module

- $\alpha$  TRFB - zum Übertragen von Werten  
Fluid-Rand und zurück
- $\alpha$  TRBS - zum Übertragen von Werten  
Rand-Naht und zurück
- $\alpha$  BCND - zum Setzen der Randbedingungen auf dem Rand

Die folgende Tabelle 1 enthält die Hauptmodule und die von ihnen aufgerufenen Rechenmodule sowie die von den Moduln berechneten bzw. übertragenen Größen. Eine ausführliche Beschreibung aller Module befindet sich im Anhang A2.

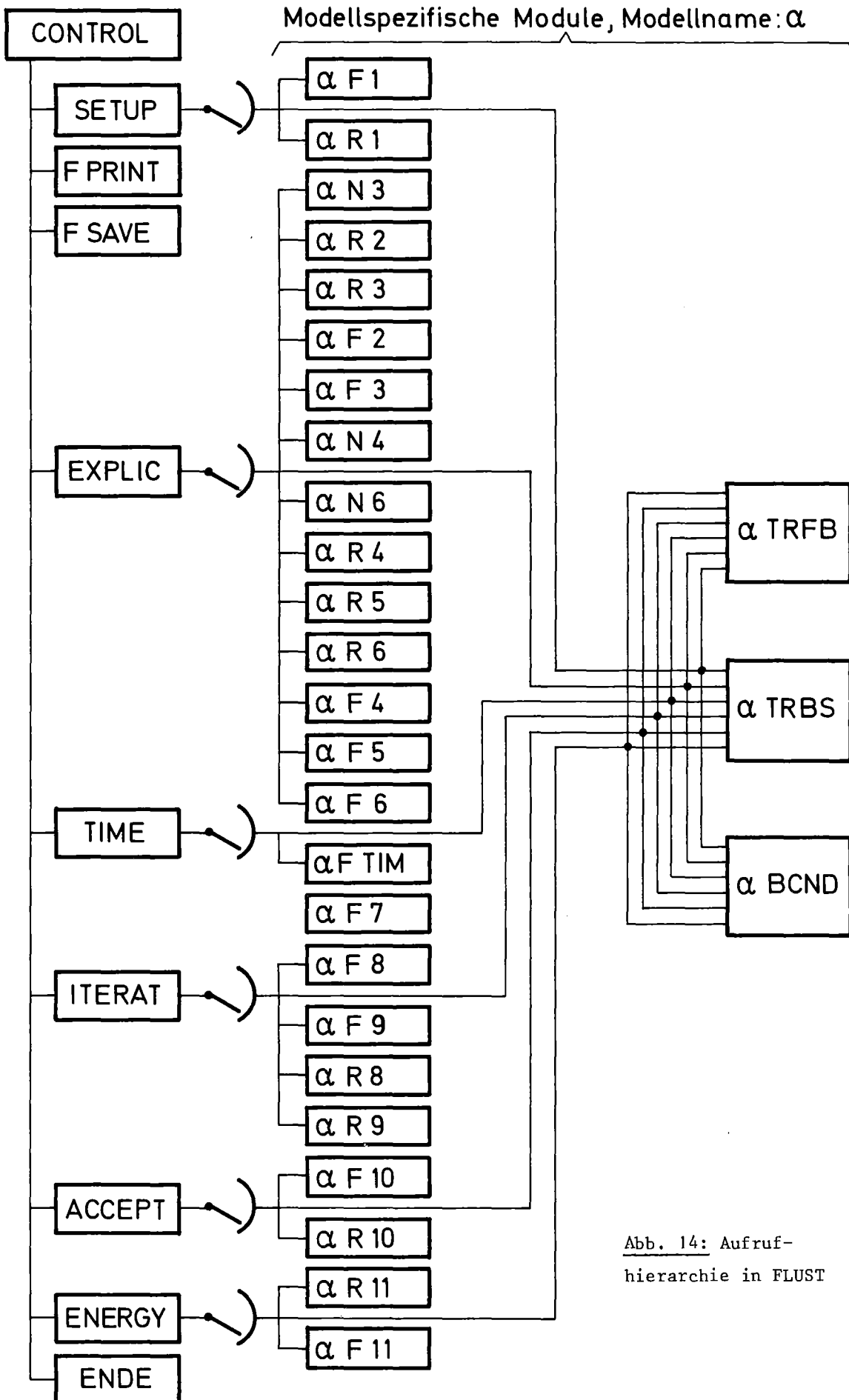


Abb. 14: Aufruf-  
hierarchie in FLUST

Tabelle 1: FLUST-Module

Erläuterung:

- F → R: übertrage vom Fluid auf den Rand
- R → F: übertrage vom Rand ins Fluid
- R → N: übertrage vom Rand auf die Naht
- N → R: übertrage von der Naht auf den Rand
- B.C.: Randbedingung
- u auf dem Rand ist die Geschwindigkeitskomponente senkrecht zum Rand,
- v ist auf dem Rand die Tangentialkomponente

Haupt-modul	Errechnete oder übertragene Größen	Modul
SETUP	<p><u>Bei Restart:</u></p> <p>Für alle Fluide</p> <p>rechne <math>\rho u, \rho v, p</math> Anfang</p> <p>F → R <math>\rho u, \rho v, \rho, e, u, v</math></p> <p style="padding-left: 40px;"><math>u_{in}, v_{in}</math></p> <p>Für alle Ränder</p> <p>B.C. <math>\rho, e, \rho v, u</math></p> <p>rechne <math>\rho u, \rho v</math></p> <p>B.C. <math>u_{in}, v_{in}, \langle \rho uv \rangle_{in},</math>  <math>\langle \rho vu \rangle_{in}, \langle \rho uv \rangle, \langle \rho vu \rangle</math></p> <p>R → N <math>u, v, u_{in}, v_{in}, \rho u, \rho v, e</math></p> <p>Für alle Fluide</p> <p>R → F <math>\rho u</math></p>	<p><math>\alpha</math> F1</p> <p><math>\alpha</math> TRFB</p> <p><math>\alpha</math> BCND</p> <p><math>\alpha</math> R1</p> <p><math>\alpha</math> BCND</p> <p><math>\alpha</math> TRBS</p> <p><math>\alpha</math> TRFB</p>
	<p><u>Bei Neustart:</u></p> <p>Für alle Fluide</p> <p>F → R <math>\rho u, \rho v, \rho, e</math></p> <p>Für alle Ränder</p> <p>B.C. <math>\rho, e, \rho u, \rho v</math></p> <p>rechne <math>u, v</math></p>	<p><math>\alpha</math> TRFB</p> <p><math>\alpha</math> BCND</p> <p><math>\alpha</math> R1</p>



Tab. 1 (Forts.)

Haupt-modul	Errechnete oder übertragene Größe	Modul
SETUP	<p>Für alle Fluide</p> <p><math>R \rightarrow F \quad u</math></p> <p>rechne <math>u, v, p</math></p> <p><math>F \rightarrow R \quad v, u_{in}, v_{in}</math></p> <p>Für alle Ränder</p> <p>B.C. <math>v, u_{in}, v_{in}, \langle \rho uv \rangle_{in},</math>  <math>\langle \rho vu \rangle_{in}, \langle \rho uv \rangle, \langle \rho vu \rangle</math></p> <p><math>R \rightarrow N \quad u, v, u_{in}, v_{in},</math>  <math>\rho u, \rho v, e</math></p>	<p><math>\alpha</math> TRFB</p> <p><math>\alpha</math> F1</p> <p><math>\alpha</math> TRFB</p> <p><math>\alpha</math> BCND</p> <p><math>\alpha</math> TRBS</p>
	EXPLICIT	<p>Für alle Nähte</p> <p>rechne <math>\langle \rho uv \rangle, \langle \rho vu \rangle</math></p> <p><math>\alpha</math> N3</p> <p>Für alle Ränder</p> <p>rechne <math>\langle \rho u \rangle</math></p> <p><math>N \rightarrow R \quad \langle \rho uv \rangle, \langle \rho vu \rangle</math></p> <p>rechne <math>\langle \rho uv \rangle, \langle \rho vu \rangle</math></p> <p><math>\alpha</math> R2</p> <p><math>\alpha</math> TRBS</p> <p><math>\alpha</math> R3</p> <p>Für alle Fluide</p> <p><math>R \rightarrow F \quad \langle \rho uv \rangle, \langle \rho vu \rangle, \rho u</math></p> <p>rechne <math>\langle \rho u \rangle, \langle \rho v \rangle</math></p> <p>rechne <math>\langle \rho uv \rangle, \langle \rho vu \rangle, \langle \rho u^2 \rangle,</math>  <math>\langle \rho v^2 \rangle, \tilde{p}, \tilde{P}</math></p> <p><math>F \rightarrow R \quad \tilde{p}, \tilde{\rho}, \langle \rho u^2 \rangle, \langle \rho v^2 \rangle,</math>  <math>\langle \rho uv \rangle_{in}, \langle \rho vu \rangle_{in}</math></p> <p><math>\alpha</math> TRFB</p> <p><math>\alpha</math> F2</p> <p><math>\alpha</math> F3</p> <p><math>\alpha</math> TRFB</p> <p>Für alle Ränder</p> <p>B.C. <math>\langle \rho u^2 \rangle, \langle \rho v^2 \rangle, u, \rho, p</math></p> <p><math>R \rightarrow N \quad p, \rho, \langle \rho u^2 \rangle, \langle \rho v^2 \rangle,</math>  <math>\langle \rho uv \rangle_{in}, \langle \rho vu \rangle_{in}, u_{B.C.}</math></p> <p><math>\alpha</math> BCND</p> <p><math>\alpha</math> TRBS</p> <p>Für alle Nähte</p> <p>rechne <math>u, v</math></p> <p>rechne <math>\beta</math></p> <p><math>\alpha</math> N4</p> <p><math>\alpha</math> N6</p>

Tab. 1 (Forts.)

Haupt- modul	Errechnete oder übertragene Größe	Modul
EXPLIC	<p>Für alle Ränder  <math>N \rightarrow R \quad \tilde{u}, \beta</math>                      rechne <math>\overline{\rho u}, \tilde{u}, \overline{\rho v}, \tilde{v}</math>                      rechne <math>\langle \tilde{\rho u} \rangle</math>                      rechne <math>\beta</math>                      B.C. <math>(\tilde{\rho u})</math></p> <p>Für alle Fluide  <math>R \rightarrow F \quad \tilde{u}, \beta</math>                      rechne <math>\overline{\rho u}, \tilde{u}, \overline{\rho v}, \tilde{v}</math>                      rechne <math>\langle \tilde{\rho u} \rangle, \langle \tilde{\rho v} \rangle</math>                      rechne <math>(\tilde{\rho u}), (\tilde{\rho v}), \frac{1}{c}, \beta, \rho_{\max}</math>  <math>F \rightarrow R \quad (\tilde{\rho v})</math></p> <p>Für alle Ränder                      B.C. <math>(\tilde{\rho v})</math>                      iter = 0  <math>R = \epsilon \cdot \rho_{\max} / \Delta t</math></p>	<p><math>\alpha</math> R4  <math>\alpha</math> R5  <math>\alpha</math> R6  <math>\alpha</math> BCND    <math>\alpha</math> TRFB  <math>\alpha</math> F4  <math>\alpha</math> F5  <math>\alpha</math> F6  <math>\alpha</math> TRFB    <math>\alpha</math> BCND</p>
TIME	<p>Für alle Fluide                      WHILE ( <math>\neg</math> timestephalf )                      rechne timestephalf, timestepdouble</p>	<p><math>\alpha</math> FTIM</p>
ITERAT	<p>WHILE ( <math>\neg</math> ende &amp; iter &lt; maxiter )                      iter = iter + 1                      Für alle Fluide  <math>R \rightarrow F \quad \langle \tilde{\rho u} \rangle</math>                      rechne <math>D, \delta \tilde{p}, \tilde{p}, \tilde{\rho}, \text{ende}</math>                      rechne <math>(\tilde{\rho u}), (\tilde{\rho v}), \tilde{u}, \tilde{v}</math>                      rechne <math>\langle \tilde{\rho u} \rangle, \langle \tilde{\rho v} \rangle</math>  <math>F \rightarrow R \quad \langle \tilde{\rho u} \rangle, \delta \tilde{p}, \tilde{\rho}</math></p>	<p><math>\alpha</math> TRFB  <math>\alpha</math> F7  <math>\alpha</math> F8  <math>\alpha</math> F9  <math>\alpha</math> TRFB</p>

Tab. 1 (Forts.)

Haupt-modul	Errechnete oder übertragene Größen	Modul
ITERAT	<p>Für alle Ränder</p> <p>B.C. <math>\delta p, \rho, \rho u</math></p> <p>rechne <math>(\rho u), u</math></p> <p>rechne <math>\langle \rho u \rangle</math></p> <p>Für alle Fluide</p> <p>R <math>\rightarrow</math> F <math>(\tilde{\rho u}), \tilde{u}</math></p> <p>F <math>\rightarrow</math> R <math>(\tilde{\rho v}), \tilde{v}</math></p>	<p><math>\alpha</math> BCND</p> <p><math>\alpha</math> R8</p> <p><math>\alpha</math> R9</p> <p><math>\alpha</math> TRFB</p> <p><math>\alpha</math> TRFB</p>
ACCEPT	<p>Für alle Fluide</p> <p>rechne <math>u, v, p, \rho, \rho u, \rho v</math></p> <p>R <math>\rightarrow</math> F <math>u_{in}</math></p> <p>Für alle Ränder</p> <p>rechne <math>u, v, \rho u, \rho v, \rho</math></p> <p>B.C. <math>\rho u, \rho v, u, v, \rho, u_{in}, v_{in}</math></p> <p>R <math>\rightarrow</math> N <math>\rho u, u, u_{in}</math></p>	<p><math>\alpha</math> F10</p> <p><math>\alpha</math> TRFB</p> <p><math>\alpha</math> R10</p> <p><math>\alpha</math> BCND</p> <p><math>\alpha</math> TRBS</p>
ENERGY	<p>Für alle Ränder</p> <p>rechne <math>\langle eu \rangle</math></p> <p>Für alle Fluide</p> <p>R <math>\rightarrow</math> F <math>\langle eu \rangle</math></p> <p>rechne <math>\langle eu \rangle, \langle ev \rangle, e</math></p> <p>F <math>\rightarrow</math> R <math>e</math></p> <p>Für alle Ränder</p> <p>B.C. <math>e</math></p> <p>R <math>\rightarrow</math> N <math>e</math></p>	<p><math>\alpha</math> R11</p> <p><math>\alpha</math> F11</p> <p><math>\alpha</math> TRFB</p> <p><math>\alpha</math> BCND</p> <p><math>\alpha</math> TRBS</p>
ENDE	<p>Gebe ENTRIES frei</p> <p>CALL FPRINT (drucke Werte)</p> <p>CALL FSAVE (rette Werte)</p>	<p>FPRINT</p> <p>FSAVE</p>

Tabelle 1: FLUST-Module

## 5.2 Datenstruktur

Die Datenstruktur des FLUST-Programms besteht aus 3 Ebenen:

- die Modellstruktur, die es nur einmal gibt, die an jeden Modul übergeben wird und die globale, das Modell und seinen Zustand betreffende Daten enthält, z.B. Anzahl Fluide oder Zeit  $t$ . Die Modellstruktur wird beim Programmstart angegeben und während der Initialisierungsphase initialisiert.
- die Komponentenstrukturen, die je einmal für jedes Fluid, jeden Rand und jede Naht vorhanden sind und Daten enthalten, die eine Komponente näher beschreiben, z.B. Maschenanzahl, geometrische Daten, topologische Daten. Sie werden in der Initialisierungsphase angelegt und initialisiert, sie werden über Zeiger in der Modellstruktur adressiert. Nur diejenigen Komponentenstrukturen, die ein Modul benötigt, sind für ihn zugreifbar.
- Die Feldgrößen-Arrays, die zu den Komponenten gehören, über die Komponentenstrukturen adressierbar sind und die Feldgrößen enthalten, z.B.  $p$ ,  $\rho$ ,  $e$ ,  $u$ ,  $v$ .

Inhalt der Modellstruktur:

Fluid-Kennzeichnung,  $\eta$ ,  $\alpha_0$ ,  $\beta_0$ ,  $\gamma_1$ ,  $\gamma_2$ ,  $\omega$ ,  $\epsilon$ ,  $\Delta t$ ,  $t$ ,  $R$ ,  $cycle$ ,  $iter$ ,  $maxiter$ ,  $timestephalf$ ,  $timestepdouble$ ;

Entry-Variable für  $p(\rho, e)$ ,  $c_2(\rho, e)$ ,  $stop$ ,  $print$ ,  $save$ ;

Anzahl Fluide, Anzahl Ränder, Anzahl Nähte;

Zeiger auf Matrix der Module ENTRIES, Filevariable für den SAVE-RESTART-File und Zeiger auf:

alle Fluid-Strukturen

alle Rand-Strukturen

alle Naht-Strukturen

Inhalt jeder Komponentenstruktur:

Kennzeichnung  $\alpha$ , Dimension;

Geometrische Daten: Ursprung, Richtung, Größe;

Maschenzahlen  $imax$ ,  $jmax$ ,  $kmax$  und Zeiger auf die Feldgrößen-Arrays, deren Aufbau und Inhalt von  $\alpha$  abhängig ist. Die folgenden Angaben beziehen sich auf das FLUST-2D-Modell mit  $\alpha = AA$ .

Inhalt der Feldgrößen-Arrays für 2D-Fluide:

$$\tilde{p}, \tilde{\rho}, \tilde{e}, \left\{ \frac{\langle \rho u^2 \rangle}{\delta \tilde{p}} \right\}, \langle \rho v^2 \rangle, \beta, \frac{1}{c^2},$$

$$(\rho u), (\rho v), u, v, \tilde{u}, \tilde{v}, \left\{ \begin{array}{l} \langle \tilde{\rho} \tilde{u} \rangle \\ \langle \tilde{\rho} u \rangle \\ \langle e u \rangle \end{array} \right\}, \left\{ \begin{array}{l} \langle \tilde{\rho} \tilde{v} \rangle \\ \langle \tilde{\rho} v \rangle \\ \langle e v \rangle \end{array} \right\}, (\tilde{\rho} \tilde{u}), (\tilde{\rho} \tilde{v}),$$

$$\langle \rho u v \rangle, \langle \rho v u \rangle$$

Geometrische Angaben:  $\Delta x, \Delta y$

Topologische Angaben: Nummer und Seite des Nachbarrandes

Inhalt der Feldgrößen-Arrays für 2D-Ränder:

Auf den Rändern und Nähten müssen nicht alle Variablen, sondern nur die für die Rechnungen auf Rändern und Nähten notwendigen Variablen gespeichert werden.

$$\tilde{p}, \tilde{\rho}, \tilde{e}, \left\{ \frac{\langle \rho u^2 \rangle}{\delta \tilde{p}} \right\}, \langle \rho v^2 \rangle, \beta, \frac{1}{c^2},$$

$$(\rho u), (\rho v), (\tilde{\rho} \tilde{u}), (\tilde{\rho} \tilde{v}), u, v, u_{in}, v_{in}, \tilde{u}, \tilde{v},$$

$$\left\{ \begin{array}{l} \langle \tilde{\rho} \tilde{u} \rangle \\ \langle \tilde{\rho} u \rangle \\ \langle e u \rangle \end{array} \right\}, \left\{ \begin{array}{l} \langle \tilde{\rho} \tilde{v} \rangle \\ \langle \tilde{\rho} v \rangle \\ \langle e v \rangle \end{array} \right\}, \langle \rho u v \rangle, \langle \rho v u \rangle, \langle \rho u v \rangle_{in}, \langle \rho v u \rangle_{in}$$

Geometrische Angaben:  $\Delta x, \Delta y, \Delta y_{in}$

Topologische Angabe: Nummer und Seite der Nachbarnähte,  
Fluidindizes IREF, Zählrichtungsangabe LZ, Nummer der  
Nachbarfluide,

Art des Randes für jede Seite des Randes,

für offene Außenränder Zeiger auf Randbedingung.

Inhalt der Feldgrößen-Arrays für 2D-Nähte:

$$\tilde{p}, \tilde{\rho}, \tilde{e}, \langle \rho u^2 \rangle, \langle \rho v^2 \rangle, \beta$$

$$(\rho u), (\rho v), u, v, u_{in}, v_{in}, \tilde{u}, \tilde{v},$$

$$\langle v u v \rangle, \langle \rho v u \rangle, \langle \rho u v \rangle_{in}, \langle \rho v u \rangle_{in}$$

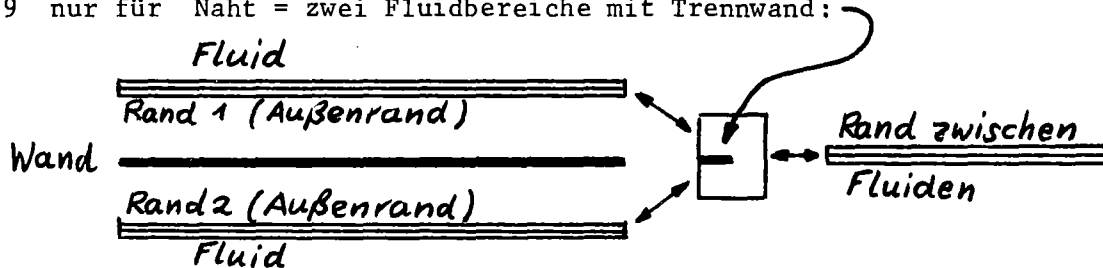
Geometrische Angaben:  $\Delta x, \Delta y, \Delta x_{in}, \Delta y_{in}$

Geometrische Angaben: Nummer der  
Nachbarränder, Nahtindizes IREF  
Zählrichtungsangabe LZ,

Art des Randes für alle Seiten.

Art des Randes:

- 1 = Fluid
- 2 = Wand mit Haftung
- 3 = Wand ohne Haftung
- 4 = Kontinuierlicher Abfluß
- 6 = Erzwungener Zufluß
- 7 nur für Naht = Randseite ohne Nachbarrand zum Fluid, z.B. linke Randseite in Abb. 7, S.26
- 8 = Verbindung zweier 2D-Fluide unter 90°-Drehung.
- 9 nur für Naht = zwei Fluidbereiche mit Trennwand:



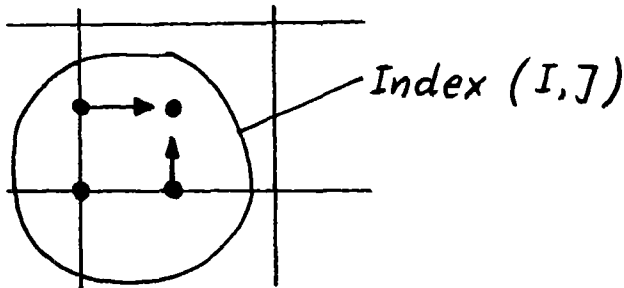
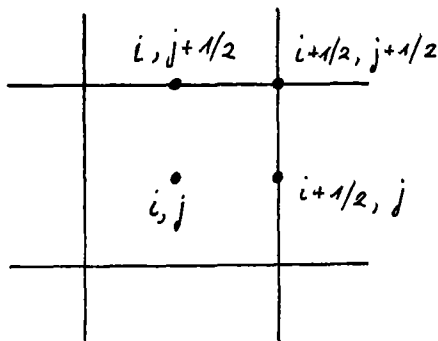
Randbedingung (für Art = 4 und 6): Durch Eingabe können konstante oder zeitlich veränderliche Randbedingungen gewählt werden.

Konstante Bedingungen:  $\rho_u, \rho, e$  auf dem Rand als konstante Werte.

Zeitabhängige Bedingungen:  $\rho_u(\mathbf{t}), \rho(\mathbf{t}), e(\mathbf{t})$  auf dem Rand als Entry-Variable, die durch die Eingabe zur Verfügung gestellt werden.

Indizierung:

Formeln:



Für Rechnung:

- $i-1/2 \rightarrow I$
- $i+1/2 \rightarrow I+1$
- $i \rightarrow I$
- $i+3/2 \rightarrow I+2$
- $j-1/2 \rightarrow J$
- $j+1/2 \rightarrow J+1$
- $j \rightarrow J$
- $j+3/2 \rightarrow J+2$

Das bedeutet, daß die Indizes (I, J) für die auf der Fläche liegenden Größen von 1 bis IMAX+1 und 1 bis JMAX+1 laufen und die Indizes für die in der Maschenmitte liegenden Größen von 1 bis IMAX und 1 bis JMAX.

Abb. 15 gibt einen Überblick über den Zusammenhang der einzelnen Teile der Datenstruktur. Die genaue Realisierung der Datenstruktur in Feldern und Listen sowie die verwendeten Variablennamen werden im Anhang A3 angegeben.

Gegenwärtig sind alle Real-Größen in doppelt genauen Variablen gespeichert, die Rechengenauigkeit wird über den Compilezeitparameter PREC gesteuert (Membername PREC). %PREC=53 bedeutet BIN FLOAT(53), also doppelt genau.

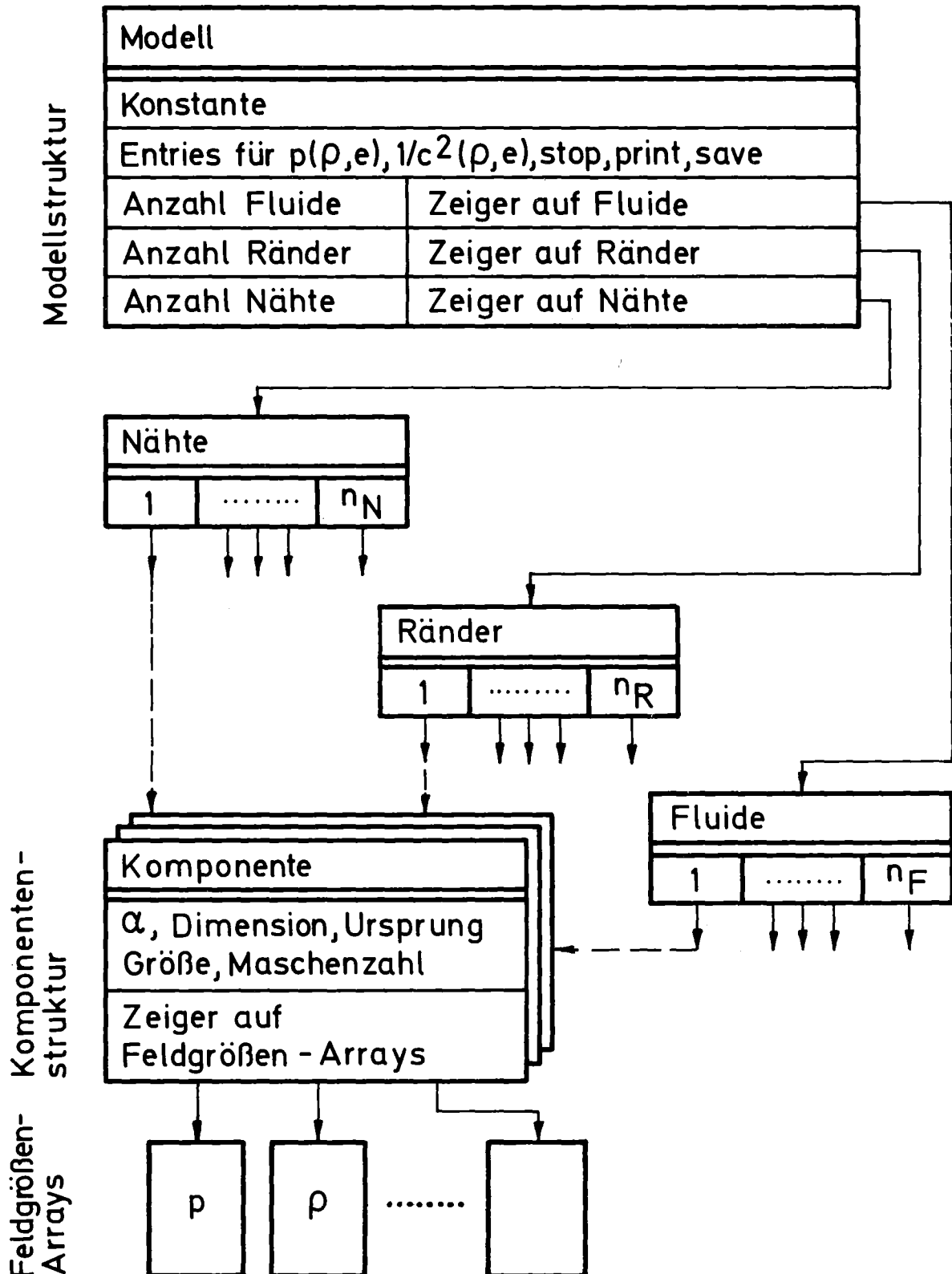


Abb. 15: Übersicht über die Datenstruktur

### 5.3 Der Save-File

Anmerkung: Die Unterprogramme zum Retten der Daten auf Band, zum Wiedereinlesen bei Restart und das graphische Auswerteprogramm FLUSTPLOT wurden von Herrn K.H. Bechler erstellt.

#### Save-Restart

Abhängig von Zeit  $t$  oder Zyklus können nach bestimmten Iterationsschritten Zustandsgrößen wie Zeit und Zyklus und die Feldgrößen  $p, \rho, e, u$  und  $v$  auf eine Datei geschrieben werden. Durch Steueranweisungen werden die Zeitpunkte des Rettens der Daten bestimmt. Angaben, die das Modell beschreiben, werden nur einmal auf die Datei gespeichert. Mit Hilfe dieser Save-Datei ist es möglich, einen unterbrochenen Rechenlauf ab einem beliebigen gespeicherten Zeitpunkt fortzusetzen. Parameter wie Zähigkeit und Donor-Cell-Parameter können beim Restart-Lauf geändert werden, jedoch nicht die Modellkonfiguration.

#### Graphische Ausgabe

Der Save-File ist nicht nur erforderlich für einen Restart, er ist auch die Grundlage der gesamten Auswertung. Für die graphische Aufbereitung der errechneten und auf dem Save-File gespeicherten Daten wurde das Programm FLUSTPLOT [14] erstellt. Es erzeugt:

- Diagramme  $f(t)$  für die Größen  $p, \rho, e, u$ , Dampfgehalt  $x$ , Temperatur  $T$  für beliebige Zeitintervalle in beliebigen Maschen der Fluide,
- Vektorplots  $u(x, y)$  für die Geschwindigkeit zu beliebigen Zeitpunkten für beliebige Fluide,
- Reliefdarstellungen  $f(x, y)$  für die Größen  $p, \rho, e, u, v$ , Dampfgehalt  $x$ ,  $T$  zu beliebigen Zeitpunkten für einzelne oder aneinandergefügte Fluide mit beliebiger Zentral- oder Parallelprojektion,
- Höhenlinien  $f(x, y)$  für die Größen  $p, \rho, e, u, v$ , Dampfgehalte  $x$ ,  $T$  zu beliebigen Zeitpunkten für beliebige Fluide,
- Für die drei letzteren Möglichkeiten ist außer der Ausgabe auf Zeichengeräte auch die Erzeugung von Filmen möglich.

Die Abbildungen zu den Testbeispielen und Anwendungsrechnungen zeigen Ergebnisse des Programms FLUSTPLOT.



#### 5.4 FLUST als REGENT-Subsystem

FLUST wurde ebenso wie FLUSTPLOT als Subsystem des integrierten CAD-Systems REGENT [15,16,17,18,19] realisiert. REGENT erwies sich als hervorragend geeignete Basis für ein Programm mit einer so komplexen Programm- und Datenstruktur wie FLUST, das darüber hinaus noch leicht erweiterbar und modifizierbar sein soll. Folgende REGENT-Eigenschaften waren von besonderem Nutzen:

##### Modularer Aufbau

REGENT ermöglicht und fördert die Gliederung von Programmen in einzelne unabhängig voneinander entwickelbare Einheiten, in REGENT Module genannt. Ebenso unterstützt REGENT die Aufgliederung von Datenbereichen in einzelne Blöcke und sichert die Konsistenz von Datenstrukturen über Modulgrenzen hinweg. Dies führte zu einer klar strukturierten Entwicklung von FLUST in Schritten von oben nach unten ("top down design").

##### Module

Ein REGENT-Modul besteht aus einer oder mehreren externen Prozeduren, die unabhängig von anderen Modulen übersetzt, gebunden und auf eine Systembibliothek geladen werden können. Nach der Änderung eines Moduls ist keine Änderung an andern Systemteilen und kein Neubinden erforderlich. Bei der Subsystem-Ausführung werden Module von einer Modulverwaltung bei Bedarf dynamisch in den Arbeitsspeicher geladen und bei Platzmangel wieder freigegeben. Alle im Abschnitt Programmstruktur (5.1) angesprochenen Module sind REGENT-Module. Ein Umschalten von einem Modell auf das andere für verschiedene Komponenten durch Auswahl eines Moduls anhand einer Kennzeichnung  $\alpha$  wäre ohne die REGENT-MODUL-Verwaltung nur mit erheblichem Zusatzaufwand und Assembler-Programmierung möglich gewesen.

##### Dynamische Datenstrukturen

Alle Komponentendaten in FLUST sind in dynamischen Datenstrukturen gespeichert. Die REGENT-Datenverwaltung allokiert Platz für die Daten, ermöglicht den Zugriff und gibt Platz wieder frei. Die Datenfelder sind die unterste Ebene einer Baumstruktur, sowohl die Größe von Datenfeldern als auch die Zugriffsstruktur kann dynamisch geändert werden. Bei Platzmangel in Hauptspeicher werden die Datenfelder auf externe Speicher ausgelagert und bei Bedarf wieder zurückgelesen. Die Allokierung von Platz und das Initialisieren der Felder geschieht in der Initialisierungsphase von FLUST. Für alle Zugriffe

in den Rechen-Modulen werden die benötigten Felder im Arbeitsspeicher festgelegt (REGENT-Anweisung FIX). Der Zugriff erfolgt dann sehr effektiv über PL/1-BASED-Felder. Nach der Bearbeitung einer Komponente werden die Felder wieder freigegeben (LOOSE-Anweisung), damit sie bei Bedarf ausgelagert werden können. Die Speicherung der Feldgrößen in Feldern variabler Dimensionen für eine beliebige Anzahl von Komponenten konnte durch Verwendung der dynamischen Datenstrukturen von REGENT in einfacher und übersichtlicher Weise erfolgen.

#### Anpassung an Hauptspeichergröße

Die REGENT-Modulverwaltung und die Datenverwaltung erlauben es, die vorhandene Hauptspeichergröße möglichst optimal auszunutzen. Geringe Speichergröße (z.B. interaktive FLUST-Anwendung):

- Beschränkter Modul-Speicherbereich
- Laden der Module bei Bedarf von einer externen Bibliothek
- Die maximale Länge der gleichzeitig aktiven Module (die sich gegenseitig rufen) bestimmt den erforderlichen Speicherplatz für Module
- Beschränkter Datenbereich
- Auslagern von nicht benötigten Daten auf externe Speicher
- Die maximale Länge der Datenfelder einer Komponente, die gleichzeitig verfügbar sein müssen, bestimmt den erforderlichen Speicherplatz für die Daten
- Hohe Ein-/Ausgaberraten, erhöhte Rechenzeit.

#### Große Speichergröße:

- Großer Modulbereich zur Aufnahme aller Rechen-Module
- Einmaliges Laden von der externen Bibliothek (REGENT-LOAD-Anweisung) und schneller Aufruf
- Großer Datenbereich zur Aufnahme aller Daten
- Kein Auslagern von Daten auf externe Speicher
- Geringe Ein-/Ausgabertätigkeit, kurze Rechenzeit.

Damit sind sowohl kurze Testläufe, auch interaktiv, als auch lange Produktionsläufe effektiv möglich.

### Eingabesprache

Wie alle REGENT-Subsysteme wird auch FLUST durch eine Anwendersprache gesteuert. Die Einbettung der Anwendersprache in P1/1 erlaubt auch Zusatzrechnungen im Anwenderprogramm. Die FLUST-Sprachanweisungen werden im folgenden Abschnitt beschrieben.

## 6. Benutzungsanleitung

### 6.1 Anwendersprache

Eine FLUST-Rechnung wird durch ein Anwenderprogramm gesteuert. Der Programmrahmen ist für alle REGENT-Subsysteme gleich, hier wird nur auf die wichtigsten Einzelheiten eingegangen. Beispiel für ein Anwenderprogramm:

```
// EXEC REGENT
//P.SYSIN DD *

R1: PROC OPTIONS(MAIN) REGENT(MPOOL=100000,
    DPOOL=120000);
    ENTER FLUSH HDR2;
    .
    .   FLUST-Anweisungen
    .
    END FLUSH;
    MESSAGE ACTIVE DEBUG;
    FINISH;
END R1;
```

Die erste Programmzeile setzt den Arbeitsspeicherbereich für die Module auf 100000 Bytes und für die Daten auf 120000 Bytes. Zwischen ENTER und END FLUSH sind FLUST-Steueranweisungen einzufügen. Auf der ENTER-Anweisung wird der Name des zu rechnenden Modells (hier HDR2) angegeben. Der Modell-Aufbau wird in Abschnitt 6.2 beschrieben. Die Anweisungen MESSAGE ACTIVE DEBUG und FINISH veranlassen die Ausgabe einer Statistik über Modul- und Datenbereiche und Ein-/Ausgabeaktivitäten nach Abschluß der Rechnung. Diese Angaben erleichtern die Auswahl einer angemessenen Speicherstrategie. Im folgenden werden die FLUST-Anweisungen aufgeführt und erläutert.

#### Zugreifbare Variable:

VISCOSITY	Zähigkeit $\eta$
DONOR_CELL_FACTOR	Donor-Cell-Faktor $\alpha$
PREDICTOR_FACTOR	Donor-Cell-Faktor $\beta$
HALF_TIME_LIMIT	Zeitgrenze $\gamma_1$
DOUBLE_TIME_LIMIT	Zeitgrenze $\gamma_2$
OVER_RELAXATION_FACTOR	Überrelaxationsfaktor $\omega$
ITERATION_LIMIT1	Abbruchschwelle $\epsilon$
TIME_STEP	Zeitschritt $\Delta t$

TIME	Zeit t
ITER	Iterationszahl iter
ITERATION_MAXIMUM	maximale Iterationszahl maxiter
SAVE_FILE	Datei (P1/1-File) für Save
SAVE_TITLE	DD-Name für SAVE, 8 Zeichen

Erläuterung:

Diese Variable können im Anwenderprogramm sowohl abgefragt als auch gesetzt werden.

Beispiel:

```
PUT LIST(TIME,      TIME_STEP);
VISCOSITY = 1.0E-4;
SAVE_TITLE = 'SAVE';SAVE_FILE=F1;
```

CONDITION-Anweisung

```
CONDITION name;
.
.
.
.
RETURN(cond);
END name;
```

Erläuterung:

Zwischen CONDITION und END werden Bedingungen geprüft und im RETURN entweder '1'B oder '0'B zurückgegeben. Die Bedingungen, die entweder erfüllt oder nicht erfüllt sein können, werden geprüft, wenn festgestellt werden soll, ob an einem bestimmten Zeitpunkt gedruckt, Daten auf Band gerettet oder die Rechnung beendet werden soll. Insbesondere können zwischen COND und END die Variablen TIME und CYCLE überprüft werden.

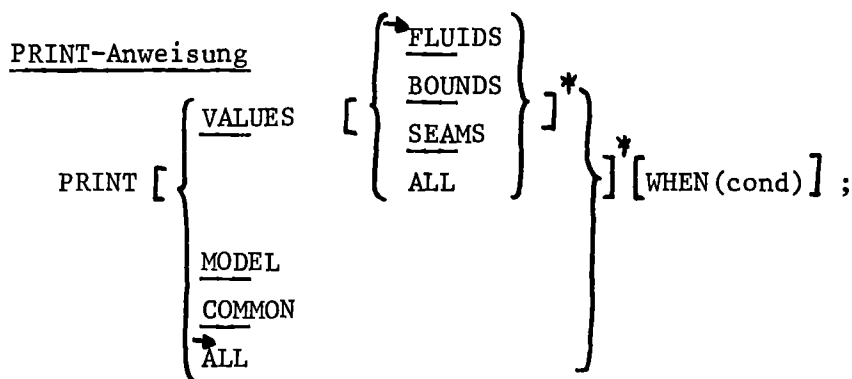
"name" ist der Name der Bedingung  
(max. 32 Zeichen)

"cond" muß vom Datentyp BIT(1) sein,  
'1'B, wenn die Bedingung erfüllt ist.

Beispiel:

```
COND PR1;  
  RETURN (MOD(CYCLE,10)=0);  
END PR1;  
PRINT WHEN(PR1);
```

Hier wird in der PRINT-Anweisung die Condition PR1 angegeben. Sie liefert immer dann den Wert '1'B, wenn CYCLE durch 10 ohne Rest teilbar ist, also alle 10 Integrationsschritte. Nur dann wird gemäß der PRINT-Anweisung gedruckt.



Erläuterung:

Die PRINT-Anweisung dient zum Drucken von Werten. Ohne die WHEN-Option wird die Anweisung sofort ausgeführt. Bei Verwendung der WHEN-Option wird immer dann gedruckt, wenn die Bedingung in der CONDITION mit Namen "cond" erfüllt ist. PRINT MODEL druckt für alle Komponenten geometrische und topologische Daten. PRINT COMMON druckt einige wichtige Common-Variable wie  $\Delta t$ , t, cycle,  $\epsilon$ ,  $\alpha$ ,  $\beta$ , iter. PRINT VALUES druckt die Werte aller Feldgrößen für Fluide (bei FLUIDS), Ränder (BOUNDS) oder Nähte (SEAMS). PRINT VALUES ALL veranlaßt das Drucken der Feldgrößen aller Komponenten. PRINT ALL druckt Modell, Common und Werte. Bei mehrfachem PRINT WHEN gilt die zuletzt ausgeführte Anweisung.

Beispiel:

```
PRINT MODEL;  
C1: CONDITION;  
  DCL PTIME STATIC BIN FLOAT(21) INIT(0);  
  IF TIME>PTIME THEN DO;  
    PTIME=PTIME+2.OE-3;  
    RETURN('1'B);  
  END;
```

```
RETURN('O'B);  
END C1;  
PRINT COMMON VALUES FLUIDS WHEN(C1);
```

Die erste PRINT-Anweisung druckt die Modelldaten. Da sich das Model während der Rechnung nicht ändert, ist ein mehrfaches Drucken nicht sinnvoll. Dann folgt die Bedingung C1, die alle 2 ms den Wert '1'B liefert und 'O'B sonst. Die PRINT-Anweisung veranlaßt das Drucken der Common-Werte und der Fluid-Werte immer dann, wenn C1 erfüllt ist, also nach jeweils 2 ms Problemzeit.

#### SAVE-Anweisung

```
SAVE WHEN cond;
```

#### Erläuterung:

Die SAVE-Anweisung veranlaßt die Ausgabe aller wichtigen Common-Variablen und Feldgrößen auf eine Datei immer dann, wenn die Bedingung "cond" erfüllt ist. Diese Datei wird später zur graphischen Auswertung und zum Restart des Programms benutzt. Bei mehrfachem SAVE gilt die zuletzt ausgeführte SAVE-Anweisung.

#### Beispiel:

```
S1: CONDITION;  
RETURN(MOD(CYCLE,2)=0);  
END S1;  
SAVE WHEN S1;
```

Alle 2 Integrationsschritte ist die Bedingung S1 erfüllt, dann werden die Werte auf eine Datei übertragen.

#### PLOT-Anweisung

```
PLOT MODEL [option]*  
  
option ::= HEIGHT expr | WIDTH expr |  
          TYP { FLUID  
              BOUNDARY } | DIM expr |  
          NAME expr
```

#### Erläuterung:

Die PLOT-Anweisung dient zum Zeichnen der Modellkomponenten. Alle anderen

Zeichenfähigkeiten sind im Subsystem FLUST\_PLOT zusammengefaßt:

HEIGHT: Höhe der Zeichnung in cm, Standard:20 cm  
WIGTH: Breite in cm, Standard 30 cm  
DIM: Dimension, bisher geht nur 2 (2D)  
NAME: Name eines einzelnen Fluids  
TYP: Angabe, ob Fluid, Boundary, Seam, Standard:Fluid  
expr: Pl/1-Ausdruck

Beispiel:

```
PLOT MODEL HEIGHT 21,WIDTH 29.0 TYP FLUID;
```

RESTART-Anweisung

```
RESTART AT { CYCLE (cycle) } FROM  
            { TIME (time) }
```

```
FILE(file) TITLE (ddname);
```

Erläuterung:

Es soll ein Restart eines Rechenlaufes vorgenommen werden. Die Daten befinden sich auf einer Datei mit DD-Namen "ddname", die über den Pl/1-Dateinamen "file" angesprochen wird. Der Restart wird mit demjenigen Datensatz durchgeführt der als erster auf der Datei die Bedingung Zyklus  $\geq$  "cycle" oder Zeit  $\geq$  "time" erfüllt. Die Common-Variablen und die Feldgrößen werden aus der Datei entnommen und als Anfangswerte des neuen Rechenlaufes verwendet.

Beispiel:

```
DCL R1 FILE INPUT;  
RESTART AT TIME (50.0E-3)FROM  
FILE(R1) TITLE('RESTART1');
```

Die ersten Datensätze des Restart-Bandes werden vor Beginn der Rechnung auf das Save-Band kopiert, danach werden die neuen Save-Daten auf das Save-Band geschrieben.

TERMINATE-Anweisung

```
TERMINATE WHEN cond;
```



Erläuterung:

Der Programmfluss wird dann abgebrochen, wenn die Bedingung "cond" zum ersten Mal erfüllt ist.

Beispiele:

```
TERMINATE WHEN C1;  
C1: CONDITION; RETURN(TIME > 0.1); END C1;
```

Ende der Rechnung, wenn  $t > 0.1$  .

```
C2: CONDITION; RETURN(CYCLE >=100); END C2;  
TERM WHEN C2;
```

Ende der Rechnung nach 100 Integrationsschritten.

Die TERM-Anweisung wird nach jedem Integrationsschritt einmal aufgerufen, um die Ende-Bedingung zu testen. Daher können in der zugehörigen Bedingung auch Zustandsvariablen abgefragt oder verändert werden, z.B.:

```
C3: CONDITION;  
  PUT LIST('TIME=',TIME,'DT=', TIME_STEP);  
  IF TIME > 0.01 THEN VISCOSITY=0.003;  
  RETURN(TIME >=0.1);  
END C3;  
TERMN WHEN(C3);
```

Diese Anweisungen dienen dazu, Zeit und Zeitschritt nach jedem Integrations-schritt auszudrucken. Beim Zeitpunkt  $t=0.01$  wird die Zähigkeit auf 0.003 verändert. Das Ende der Rechnung wird bei  $t=0.1$  erreicht.

MATERIAL-Anweisung

```
MATERIAL { IDEAL GAS  
            WATER  
            FLUXH2O } ;
```

Erläuterung:

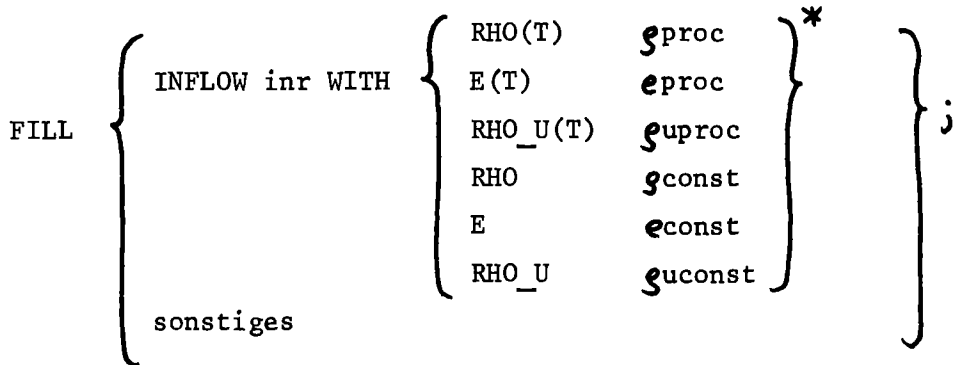
Die MATERIAL-Anweisung dient zur Auswahl der Zustandsgleichung  $p(\rho, e)$  und der Gleichung für die Schallgeschwindigkeit  $c^2(\rho, e)$ , siehe Abschnitt 3.4.

- MATERIAL IDEALGAS: ideales Gas  $\gamma = \frac{5}{3}$ ,
- MATERIAL WATER: genaue Wassergleichung
- MATERIAL FLUXH2O: lineare Wassergleichung

Beispiel:

MATERIAL WATER;

FILL-Anweisung



Erläuterung:

Die FILL-Anweisung dient zum Setzen von Randbedingungen und Modellparametern. Gegenwärtig ist lediglich FILL INFLOW zur Beschreibung von Zu- und Abflüssen an Rändern verfügbar. Alle anderen Parameter des Modells werden in der Modellbeschreibung (Abschnitt 6.2) angegeben.

"inr": Nr. einer Randbedingung, sie ist auf dem zugehörigen Rand eingetragen.

RHO,E,RHO\_U: Zeitlich konstante Werte für  $g, e$  und/oder  $g_u$  werden angegeben.

"gconst", "econst" und "guconst" sind Ausdrücke, die in der Initialisierungsphase ausgewertet werden und nicht von der Zeit abhängen dürfen.

RHO(T),E(T),RHO\_U(T): Zeitabhängige Randbedingungen für  $g, e$  und/oder  $g_u$ .

"gproc", "eproc" und "guproc" sind P1/1-Prozeduren, die bei ihrem Aufruf abhängig von Zeit oder Zyklus variable Werte liefern. Die Prozeduren müssen das Attribut RETURNS(BIN FLOAT(53)) besitzen.

Beispiel:

```
FILL INFLOW 1 WITH E 0.18 RHO (RHO1+0.3)/F1;
RHO2: PROC RETURNS(BIN FLOAT(53));
  I = FLOOR (TIME*1000.0);
  RETURN(TAB(I,1));
END RHO2;
E2: PROC RETURNS(BIN FLOAT(53));
  I = FLOOR(TIME*1000.0);
```

```
RETURN(TAB(I,2));  
END E2;  
FILL INFLOW 2 WITH RHO(T) RHO2 E(T) E2;
```

Die erste FILL-Anweisung beschreibt einen Rand mit konstanten Werten  $e = 0.18$  und  $\mathcal{S} = (\text{RHO1}+0.3)/\text{F1}$ . Die zweite FILL-Anweisung verbindet die Randbedingung für  $\mathcal{S}$  mit der Prozedur RHO2 und die für  $e$  mit der Prozedur E2. In diesen Prozeduren wird die Zeit in ms umgerechnet und daraus ein Index I abgeleitet. Dieser Index wird benutzt, um aus einer Tabelle TAB die Werte für  $\mathcal{S}(t)$  bzw.  $e(t)$  herauszulesen.

LOAD-Anweisung:

```
LOAD { SERVICE  
      ITERATION  
      ALL } ;
```

Erläuterung:

LOAD wird benutzt, um Module teilweise oder vollständig in den Arbeitsspeicher zu laden. Danach ist der Zugriff über CALL genau so effektiv wie normale Unterprogrammaufrufe. Die Modulverwaltung wird nicht mehr benutzt.

LOAD SERVICE lädt nur die Übertragungs- und Randbedingungsmodule TRFB, TRBS und BCND. LOAD ITERATION lädt zusätzlich alle bei der Iterationsphase benötigten Module, LOAD ALL lädt alle Module.

Empfehlung: LOAD ALL für Produktionsläufe, Modulpool für FLUST-2D muß dann  $> 380\ 000$  Bytes sein. Kein LOAD für interaktive Anwendung.

Beispiel:

```
LOAD ALL;
```

SOLVE-Anweisung:

```
SOLVE;
```

Erläuterung:

SOLVE startet die Problemlösung. Der Hauptmodul CONTRL wird aufgerufen, der die Berechnung der Strömung in den Modellkomponenten steuert.

Beispiel:

```
SOLVE;
```

Vollständiges Beispiel:

```
//IRE147F JOB (acct),name,          JOB-Karte
// REGION=1000K,TIME=100            Produktionslauf
// EXEC REGENT,GPARM='PRINT,       REGENT-Aufruf
// MAP/ISA(250K),REPORT'           ISA im Go-Step groß wählen!

//P.SYSIN DD *                      Anwenderprogramm
H1: PROC OPTIONS(MAIN)              Modul- und Datenpool groß wählen!
  REGENT(MPOOL=380000,DPOOL=
  200000);
  ENTER FLUSH HDR2;                Aufruf FLUST

  ON ERROR BEGIN;
    CALL PLIDUMP('CTF','H');       Fehlerbehandlung
    PRINT VALUES FLUIDS;
    PRINT COMMON;
    MESSAGE ACTIVE DEBUG ;
    FINISH;STOP;
  END;

  DCL ROAUS BIN FLOAT(53) STATIC    sAußen }
    INIT(0.572738);                 Rand
  DCL EAUS BIN FLOAT(53) STATIC     eAußen }
    INIT(2.52EC);                   PRINT- und
  DCL PTIME STATIC BIN FLOAT(53);   SAVE-Zeit
  DCL STIME STATIC BIN FLOAT(53);
  STIME=FLUSH.TIME;
  PTIME=FLUSH.TIME;
  FILL INFLOW 1 E EAUS RHO ROAUS;   Randbedingung
  MATERIAL WATER;                  Fluid ist Wasser
  STOP1 :PROC;

  PUT LIST('XXXXXXXX CYCLE,ITER,
  TIME XXXXXXXX',CYCLE,ITER,TIME)SKIP;
  END STOP1;
  STOP: CONDITION;
  CALL STOP1;
  RETURN(FLUSH.TIME>=/x 100 msec=x/0.100);
  END STOP;
  TERMINATE WHEN STOP;
```

sAußen }  
eAußen } Rand

PRINT- und  
SAVE-Zeit

Randbedingung  
Fluid ist Wasser

Abbruchsteuerung mit  
Ausdrucken cycle,  
iter, t

```

PRI:CONDITION;
IF TIME>=PTIME THEN DO
    PTIME=PTIME+0.0025;
    RETURN('1'B);
END;
RETURN('0'B);
END PRI;

PRINT WHEN(PRI) VALUES FLUIDS COMMON;
S: CONDITION;
IF TIME>=STIME THEN DO;
    STIME=TIME+0.00025/π 1/4 MSEC π/;
    RETURN('1'B);
END;
RETURN('0'B);
END S;

SAVE WHEN S;
    FLUSH.TIME_STEP=1.0E-4;
    FLUSH.VISCOSITY=0.0002;
    FLUSH.DOUBLE_TIME_LIMIT=0.0005;
    FLUSH.HALF_TIME_LIMIT=0.5;
    FLUSH.OVER_RELAXATION_FACTOR=0.95;
    FLUSH.DONOR_CELL_FACTOR=1.0;
    FLUSH.PREDICTOR_FACTOR=0.0;
    FLUSH.ITERATION_LIMIT1=5.0E-4;
PRINT MODEL;
LOAD ALL;
SOLVE;

END FLUSH;
MESSAGE ACTIVE DEBUG;
FINISH;
END;
```

Ausdrucksteuerung  
Drucken alle  
2.5 ms

SAVE-Steuerung  
Retten der Werte  
alle 0.25 ms

Setzen von  
Parametern

Drucke Modell  
Laden aller Module  
Start der Rechnung

## 6.2 Modellbeschreibung

In der ENTER-Anweisung eines FLUST-Programms wird der Name eines Modells angegeben. Der Modellaufbau kann bisher nicht über Anweisungen im Programm erfolgen, er wird in einem bereitzustellenden Modul vorgenommen, der den Namen des Modells hat. Der Modul wird vor der Anwendung auf eine Bibliothek geladen. Da es sich um einen normalen REGENT-Modul handelt, wird er vom Modulgenerator erzeugt und geladen. Beispiel:

```
// EXEC QQPCL                Aufruf Modulgenerator
//P.SYSIN DD *
*SUBSYSTEM FLUSH;           Subsystemname
*MODULE M1;                 Modulname
M1:PROC;
    Modellaufbau,           PLR-Programm
    Initialisierung
END M1;
```

In FLUST könnte danach das in M1 aufgebaute Modell mit ENTER FLUSH M1; angesprochen werden. Zur Modellbeschreibung wird die Programmiersprache PLR benutzt, die als P1/1-Erweiterung die Fähigkeiten des REGENT-Systems verfügbar macht. Im einzelnen sind folgende Schritte für den Modellaufbau erforderlich:

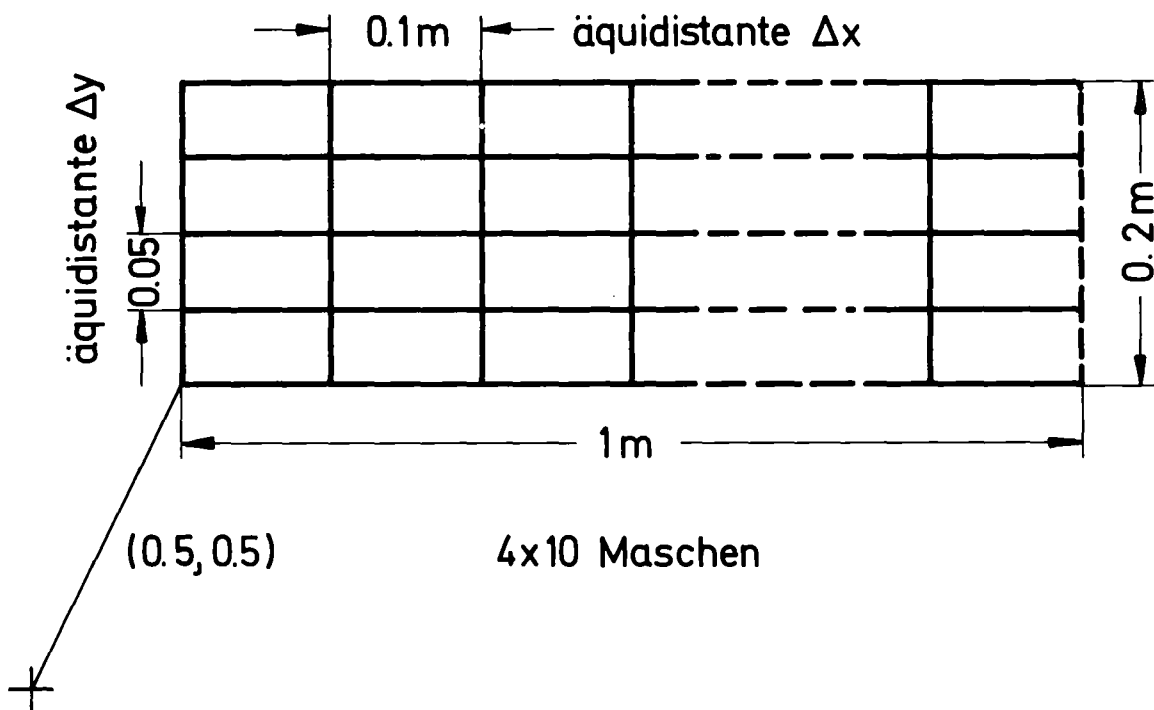
- Allokieren der Datenstrukturen für Fluide, Ränder und Nähte
- Zuweisen der Werte aller geometrischen Größen
- Wertezuweisung für alle topologischen Angaben
- Setzen aller Feldgrößen auf einen undefinierten Wert
- Setzen der Anfangsbedingungen für  $\mathcal{S}$ ,  $e$ ,  $\mathcal{S}_u$  und  $\mathcal{S}_v$ .

Das Allokieren muß entsprechend der im Anhang A3 beschriebenen Datenstruktur erfolgen. Die geometrischen Daten sind entsprechend der Geometrie des zu berechnenden Fluids zu setzen, die topologischen Daten beschreiben den Komponentenzusammenhang, insbesondere die Nachbarkomponenten und die Angaben IREF und LZ. Zur besseren Fehlererkennung werden alle Größen zunächst auf den Wert 6.66666E66 "undefiniert" gesetzt. Das Unterprogramm FILL erledigt diese Aufgabe bei fertig allokiertem Modell automatisch. Danach werden die Variablen  $\mathcal{S}$ ,  $e$ ,  $\mathcal{S}_u$  und  $\mathcal{S}_v$  entsprechend der Anfangsbedingungen gesetzt.

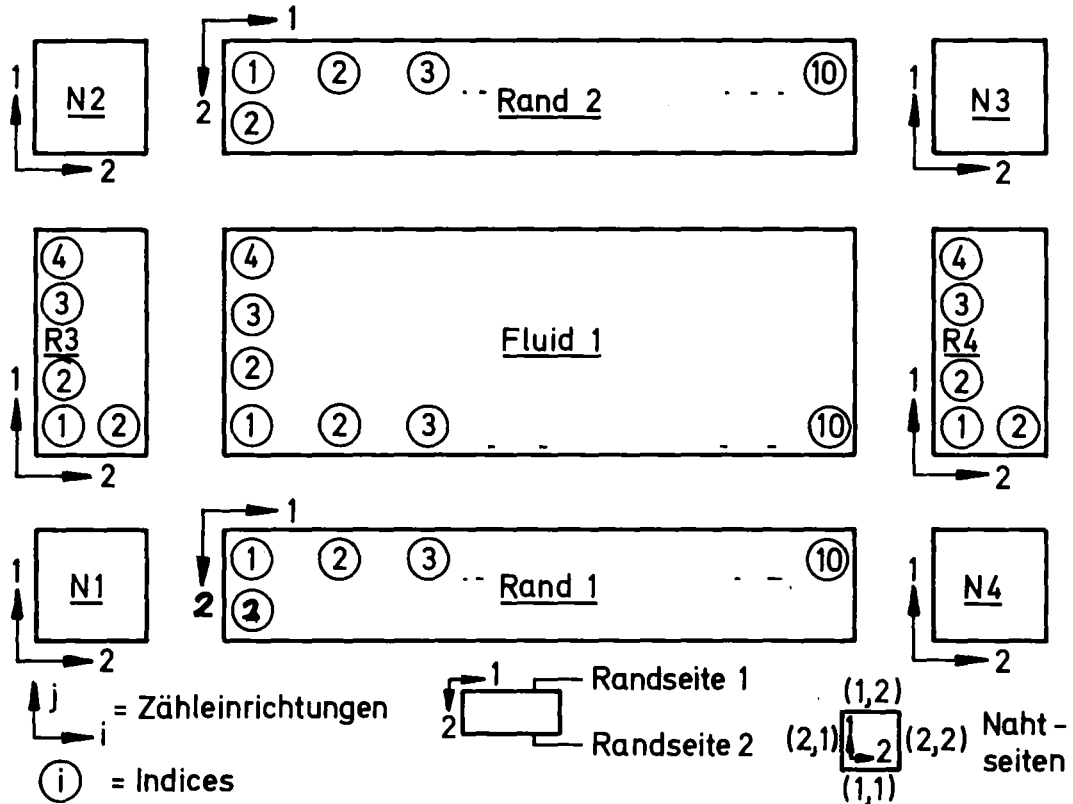
Dies ist lediglich für Fluide nötig, die Werte werden vor Beginn der Rechnung im Modul SETUP auf Ränder und Nähte übertragen. Anhand eines einfachen Beispiels soll der Modellaufbau erläutert werden. Vor Beginn der Programmierung ist festzulegen:

- Modellgeometrie
- Topologie
- Randbedingungen
- Anfangsbedingungen.

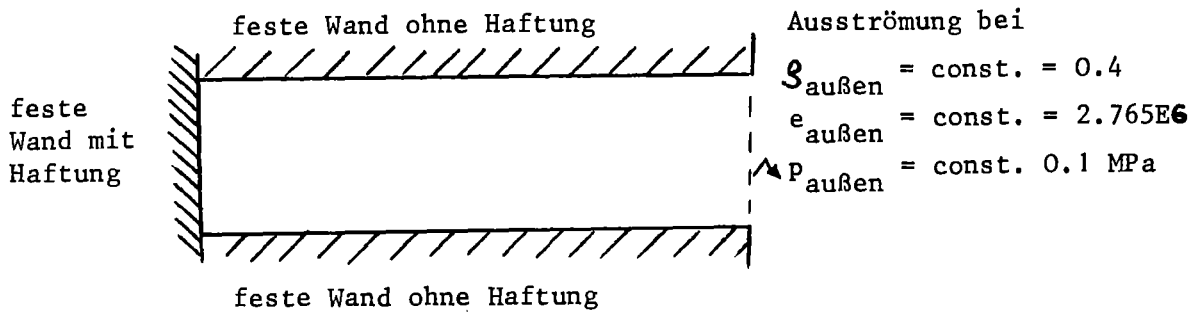
Modellgeometrie des Beispiels



Topologie des Beispiels



Randbedingungen



Anfangsbedingungen

Konstante Anfangsbedingungen

$$s = s_0 = 777.08$$

$$e = e_0 = 1.165 \cdot 10^6$$

Bei Anwendung der Wasser-Zustandsgleichung ergibt sich dafür ein Anfangsdruck von

$P_0 = 11 \text{ MPa}$  und eine Anfangstemperatur  
 von  $T_0 = 269 \text{ C}$



Aus der Modellgeometrie ist abzulesen:

Ursprung des Fluides	(0.5, 0.5)
R1, R3, N1	(0.5, 0.5)
R3, N2	(0.5, 0.7)
R4, N4	(1.5, 0.5)
N3	(1.5, 0.7)

Abmessungen des Fluides	(1.0, 0.2)
R1, R2	(1.0, 0.0)
R3, R4	(0.0, 0.2)

Maschenweiten des Fluides

0-Richtung	alle 0.1
1-Richtung	alle 0.05
R1,R2	
1-Richtung	0.1
2-Richtung	0.05
R3,R4	
1-Richtung	0.05
2-Richtung	0.1

Aus der Modelltopologie ist abzulesen:

Die untere Seite des Fluids ist mit der Seite 1 von Rand 1 verbunden.

Verbindungen: Fluid oben - Seite 2 R2  
Fluid links - Seite 2 R3  
Fluid rechts- Seite 1 R4

Der Rand R1 ist links verbunden mit Seite (2,2) der Naht N1, usw.

Zählrichtungen LZ:

Den Zählrichtungen (0,1,2) der Ränder R1 und R2 entsprechen (-2,0,-1) des Fluides, Ränder R3 und R4 haben LZ =(-2,1,0).

Nachbarindizes IREF:

Der Masche (1,1) des Randes R1 entspricht die Masche (1,1) des Fluides, für R2 ist IREF=(1,4), für R3 (1,1) und für R4 (10,1).

Die Werte LZ und IREF für alle Seiten einer Naht, die mit einem Rand verbunden sind, können ebenfalls abgelesen werden.

Die Randbedingungen sind zu berücksichtigen:

Rand 1, Seite 2 (außen)	}	Randart = 3
Rand 2, Seite 1 (außen)		Wand ohne Haftung
Rand 3, Seite 1 (außen)		Randart = 2, mit Haftung
Rand 4, Seite 2 (außen)		Randart 6, Abfluß

Für die Randart 6 muß eine Datenstruktur für die Werte  $s_{\text{außen}}$ ,  $e_{\text{außen}}$  allokiert werden. Durch die Eingabeanweisung FILL INFLOW wird sie später gefüllt.

Anfangsbedingungen sind durch Wertezuweisungen auf die Feldvariablen des Fluids zu setzen.

Damit hat der Modul für das Beispielmmodell M1 folgendes Aussehen. Die genaue Bezeichnung und Bedeutung der Variablen der Datenstruktur ist im Anhang A3 beschrieben.

```
MODULE M1;
M1: PROC REORDER;
DCL (I,J,K) BIN FIXED(15);
%INCLUDE FLMAC(PREC);
%INCLUDE FLMAC(FLUID2D),FLMAC(BOUND2D),
      FLMAC(BHEAD),FLMAC(SHEAD),
      FLMAC(SEAM2D),FLMAC(FHEAD),
      FLMAC(PARTCODE),FLMAC(INFLOW);
DCL FILL ENTRY DYNAMIC;
  #IFLUIDS=1;
  #IBOUNDS=4; /* NUMBER OF COMPONENTS */
  #ISEAMS=4;
  #IJOINTS=0;
  #FLUID=WATER;
DCL (IMAXF,JMAXF,KMAXF) BIN FIXED(15);
IMAXF=10;
JMAXF=4;
KMAXF=0; * SIZE OF COMPONENTS */
/* DEFINE COMPONENTS */
  DEFINE #FLUID_DESC,(IFLUIDS,12) DESCRIPTOR;
  DEFINE #BOUND_DESC,(IBOUNDS,8) DESCRIPTOR;
  DEFINE #SEAM_DESC, (ISEAMS ,6) DESCRIPTOR;
  DEFINE #INFLOW_DESC,(1,4) DESCRIPTOR ;
  DEFINE #INFLOW_DESC(1,1),(1) AS(INFLOW_TYPES);
DO K=1 TO 1;
/* FOR EVERY FLUID */
  DEFINE #FLUID_DESC(K,HEAD),(1)AS(F_HEAD);
  DEFINE #FLUID_DESC(K,EDGE2),
    (IMAXF+1,JMAXF+1) AS(F2_EDGE);
  DEFINE #FLUID_DESC(K,FACE0),
    (IMAXF+1,JMAXF) AS(F2_FACE);
  DEFINE #FLUID_DESC(K,FACE1),
    (IMAXF,JMAXF+1) AS(F2_FACE);
  DEFINE #FLUID_DESC(K,MESH),
    (IMAXF,JMAXF) AS(F2_MESH);
  DEFINE #FLUID_DESC(K,EXTMESH0),
    (JMAXF) AS(F2_EXTMESH);
  DEFINE #FLUID_DESC(K,EXTMESH1),
    (IMAXF) AS(F2_EXTMESH);
  DEFINE #FLUID_DESC(K,COL0),(IMAXF) AS(F2_COL);
  DEFINE #FLUID_DESC(K,COL1),(JMAXF) AS(F2_COL);
/* VALUES FOR FLUID-HEAD */
  F_HEAD(K,HEAD,1).IDENT=AA;
  F_HEAD(K,HEAD,1).DIMENSION=2;
  F_HEAD(K,HEAD,1).REFERENCE_SYSTEM=0;
  F_HEAD(K,HEAD,1).ORIGIN(0)=0.5;
  F_HEAD(K,HEAD,1).ORIGIN(1)=0.5;
  F_HEAD(K,HEAD,1).ORIGIN(2)=0.0;
  F_HEAD(K,HEAD,1).DIRECTION(0)=1.0;
  F_HEAD(K,HEAD,1).DIRECTION(1),
  F_HEAD(K,HEAD,1).DIRECTION(2)=0.0;
  F_HEAD(K,HEAD,1).SIZE(0)=1.0;
  F_HEAD(K,HEAD,1).SIZE(1)=0.2;
  F_HEAD(K,HEAD,1).SIZE(2)=0.0;
  F_HEAD(K,HEAD,1).IMAX=IMAXF;
  F_HEAD(K,HEAD,1).JMAX=JMAXF;
  F_HEAD(K,HEAD,1).KMAX=KMAXF;
END;
```

Modulkopf

Bereitstellen der  
Datenstruktur-  
deklarationen

Anzahl Fluide  
Anzahl Ränder  
Anzahl Nähte

Fluid-Name

Dimensionen

Allokierung der  
Datenstrukturen  
für alle  
Komponenten

Allokieren  
Fluid

Geometrische  
Daten für  
Fluid

```
/*FOR EVERY BOUNDARY */ DO I=1 TO 4;
  IF I<=2 THEN IMAXB=IMAXF;
    ELSE IMAXB=JMAXF;
  DEFINE #BOUND_DESC(I,HEAD),(1) AS(B_HEAD);
  DEFINE #BOUND_DESC(I,EDGE0),
  (IMAXB+1) AS(B2_EDGE);
  DEFINE #BOUND_DESC(I,FACE1),
  (IMAXB+1) AS(B2_FACE);
  DEFINE #BOUND_DESC(I,FACE2),
  (IMAXB) AS(B2_FACE);
  DEFINE #BOUND_DESC(I,MESH),
  (IMAXB) AS(B2_MESH);
  DEFINE #BOUND_DESC(I,EXTMESH1),
  (1) AS(B2_EXTMESH);
  DEFINE #BOUND_DESC(I,COL1),(IMAXB) AS(B2_COL);
  DEFINE #BOUND_DESC(I,SIDE),(1) AS(B2_SIDE);
  END;
/* VALUES FOR BOUNDARY HEADS */
DO K=1 TO 4;
  IF K<=2 THEN IMAXB=IMAXF;
    ELSE IMAXB=JMAXF;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.IDENT=AA;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.DIMENSION=2;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.REFERENCE_SYSTEM=0;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.ORIGIN(0)=
  (K=1|K=2|K=3)*0.5+(K=4)*1.5;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.ORIGIN(1)=
  (K=1|K=3|K=4)0.5+(K=2)*0.7;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.ORIGIN(2)=0.0;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.DIRECTION(0)=(K<=2)*1.0;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.DIRECTION(1)=(K>2)*1.0;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.DIRECTION(2)=0.0;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.SIZE(0)=(K<=2)*1.0;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.SIZE(1)=(K>2)*0.2;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.SIZE(2)=0.0;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.JMAX=IMAXB;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.IMAX=0;
  #BOUND_DESC(K,HEAD,1)->B_HEAD.KMAX=0;
  END;
/* FOR EVERY SEAM */
DO I=1 TO 4;
  DEFINE #SEAM_DESC(I,HEAD),(1) AS(HEAD);
  DEFINE #SEAM_DESC(I,EDGE0),(1) AS(S2_EDGE);
  DEFINE #SEAM_DESC(I,FACE1),(1) AS(S2_FACE);
  DEFINE #SEAM_DESC(I,FACE2),(1) AS(S2_FACE);
  DEFINE #SEAM_DESC(I,SIDE),(1) AS(S2_SIDE);
  DEFINE #SEAM_DESC(I,MESH),(1) AS(S2_MESH);
  END;
DO K=1 TO 4;
  /* VALUES FOR SEAM HEADS */
  #SEAM_DESC(K,HEAD,1)->HEAD.IDENT=AA;
  #SEAM_DESC(K,HEAD,1)->HEAD.DIMENSION=2;
  #SEAM_DESC(K,HEAD,1)->HEAD.REFERENCE_SYSTEM=0;
  #SEAM_DESC(K,HEAD,1)->HEAD.ORIGIN(0)=
  (K=1|K=2)*0.5+(K=2|K=3)*0.7;
  #SEAM_DESC(K,HEAD,1)->HEAD.ORIGIN(1)=
  (K=1|K=4)*0.5+(K=3|K=4)*1.5;
  #SEAM_DESC(K,HEAD,1)->HEAD.ORIGIN(2)=0.0;
  #SEAM_DESC(K,HEAD,1)->HEAD.DIRECTION(0)=0.0;
  #SEAM_DESC(K,HEAD,1)->HEAD.DIRECTION(1)=1.0;
  #SEAM_DESC(K,HEAD,1)->HEAD.DIRECTION(2)=0.0;
```

Allokieren der  
Ränder

Geometrische Daten  
für die Ränder

Allokieren der  
Nähte

Geometrische Daten  
für die  
Nähte

```
#SEAM_DESC(K,HEAD,1)->HEAD.SIZE(0)=0.0;
#SEAM_DESC(K,HEAD,1)->HEAD.SIZE(1)=0.0;
#SEAM_DESC(K,HEAD,1)->HEAD.SIZE(2)=0.0;
#SEAM_DESC(K,HEAD,1)->HEAD.IMAX=0;
#SEAM_DESC(K,HEAD,1)->HEAD.JMAX=0;
#SEAM_DESC(K,HEAD,1)->HEAD.KMAX=0;
END;

CALL FILL;

/* VALUES FOR FLUIDS */
DO K=1 TO 1;
/* FOR EVERY FLUID */
DO I=1 TO IMAXF+1;
DO J=1 TO JMAXF;
F2_FACE(K,FACE0,I,J).RHO_U=0.0;
END;
END;
DO I=1 TO IMAXF;
DO J=1 TO JMAXF+1;
F2_FACE(K,FACE1,I,J).RHO_U=0.0;
END;
END;
DO I=1 TO IMAXF;
DO J=1 TO JMAXF;
F2_MESH(K,MESH,I,J).E=1.165E6;
F2_MESH(K,MESH,I,J).RHO=777.08;
END;
END;
DO I=1 TO JMAXF;
F2_EXTMESH(K,EXTMESH0,I).INDEX_BOUNDARY(1)=3;
F2_EXTMESH(K,EXTMESH0,I).SIDE_BOUNDARY(1)=2;
F2_EXTMESH(K,EXTMESH0,I).INDEX_BOUNDARY(2)=4;
F2_EXTMESH(K,EXTMESH0,I).SIDE_BOUNDARY(2)=1;
F2_COL(K,COL1,I).DX=0.2E0/JMAXF;
END;
DO I=1 TO IMAXF;
F2_EXTMESH(K,EXTMESH1,I).INDEX_BOUNDARY(1)=1;
F2_EXTMESH(K,EXTMESH1,I).SIDE_BOUNDARY(1)=1;
F2_EXTMESH(K,EXTMESH1,I).INDEX_BOUNDARY(2)=2;
F2_EXTMESH(K,EXTMESH1,I).SIDE_BOUNDARY(2)=2;
F2_COL(K,COL0,I).DX=0.1;
END;
END;
/* VALUES FOR BOUNDARIES */
DO K=1 TO 4;
IF K<=2 THEN IMAXB=IMAXF;
ELSE IMAXB=JMAXF;
DO I=1 TO IMAXB;
B2_EXTMESH(K,EXTMESH1,1).INDEX_SEAM(1)=
(K=1)*1+(K=2)*2+(K=3)*1+(K=4)*3;
B2_EXTMESH(K,EXTMESH1,1).INDEX_SEAM(2)=
(K=1)*4+(K=2)*3+(K=3)*2+(K=4)*3;
B2_EXTMESH(K,EXTMESH1,1).SIDE_SEAM(1)=(K<=2)*5+(K>2)*3;
B2_EXTMESH(K,EXTMESH1,1).SIDE_SEAM(2)=(K<=2)*4+(K>2)*2;
END;
DO I=1 TO IMAXB;
IF K<=2 THEN ZWI=0.1; ELSE ZWI=0.05;
#BOUND_DESC(K,COL1,I)->B2_COL.DX=ZWI;
B2_COL(K,COL1,I).CHAIN=0B;
END;
END;
```

Initialisieren aller  
Feldwerte aller  
Komponenten mit  
6.666666E66

Anfangswerte für  
alle Fluide

Topologische Daten  
für alle  
Fluide

Topologische Daten  
für alle  
Ränder

dx und dy

```
DO I=1 TO IMAXB;
B2_MESH(K,MESH,1).WALLTYPE(1)=(K=1|K=4)*1+(K=3)*2+   Randarten
(K=2)*3;
B2_MESH(K,MESH,1).WALLTYPE(2)=(K=4)*6+(K=2|K=3)*1+(K=1)*3;
B2_MESH(K,MESH,1).INFLOW_INDEX=0;                       Zufluß-Nr.
IF K=4 THEN B2_MESH(K,MESH,1).INFLOW_INDEX(2)=1;
END;
B2_SIDE(K,SIDE,1).JMAXF=IMAXB;
IF K<=2 THEN ZWI=0.05; ELSE ZWI=0.1;
B2_SIDE(K,SIDE,1).DX=ZWI;
B2_SIDE(K,SIDE,1).DX_IN=ZWI;
B2_SIDE(K,SIDE,1).FLUID_BACK=0;
B2_SIDE(K,SIDE,1).LZ=0;
B2_SIDE(K,SIDE,1).IREF(1,0)=(K=4)*IMAXF;
B2_SIDE(K,SIDE,1).IREF(1,1)=(K=1|K=2)*1;
B2_SIDE(K,SIDE,1).IREF(2,0)=(K=3)*1;
B2_SIDE(K,SIDE,1).IREF(2,1)=(K=2)*JMAXF;
B2_SIDE(K,SIDE,1).IREF(*,2)=0;
END;
B2_SIDE(1,SIDE,1).LZ(1,0)=2;
B2_SIDE(1,SIDE,1).LZ(1,2)=-1;
B2_SIDE(1,SIDE,1).FLUID_BACK(1)=1;
```

/\*\*\*\*\* ETC. \*\*\*\*\*/

```
DO K=1 TO 4;
/* VALUES FOR SEAMS */
S2_SIDE(K,SIDE,1).DX(1,*,*)=0.1;
S2_SIDE(K,SIDE,1).DX(2,*,*)=0.05;
S2_SIDE(K,SIDE,1).DX_IN(1,*,*)=0.1;
S2_SIDE(K,SIDE,1).DX_IN(2,*,*)=0.05;
S2_SIDE(K,SIDE,1).LZ=0;
S2_SIDE(K,SIDE,1).IREF=0;
S2_SIDE(K,SIDE,1).BOUND_BACK(*,*)=0;
S2_MESH(K,MESH,1).WALLTYPE=3;
END;
S2_MESH(1,MESH,1).WALLTYPE(1,1,1)=2;
S2_MESH(1,MESH,1).WALLTYPE(2,2,2)=1;
S2_MESH(1,MESH,1).WALLTYPE(1,2,2)=1;
S2_MESH(1,MESH,1).WALLTYPE(2,1,1)=2;
S2_MESH(1,MESH,1).WALLTYPE(2,1,2)=2;
S2_SIDE(1,SIDE,1).LZ(2,2,2)=1;
S2_SIDE(1,SIDE,1).BOUND_BACK(2,2)=1;
S2_SIDE(1,SIDE,1).LZ(1,2,2)=2;
S2_SIDE(1,SIDE,1).LZ(1,2,1)=1;
S2_SIDE(1,SIDE,1).BOUND_BACK(1,2)=3;
S2_SIDE(1,SIDE,1).LZ(2,2,1)=-2;
```

Topologische Daten  
für Nähte und  
dx, dy

Randarten

Nachbarschafts-  
verhältnisse

/\*\*\*\*\* ETC. \*\*\*\*\*/

END M1;

## 7. Testrechnungen

### 7.1 Stoßrohr mit idealem Gas

Für ein eindimensionales Stoßrohr mit idealem Gas existiert eine analytische Lösung, dieses Problem diente daher als ersten Test für das Programm FLUST. Die Zeitschrittweite und die Maschenweite wurde variiert, um den Einfluß auf das Ergebnis festzustellen. Abb. 16 zeigt die Geometrie und den Anfangszustand des Stoßrohres.

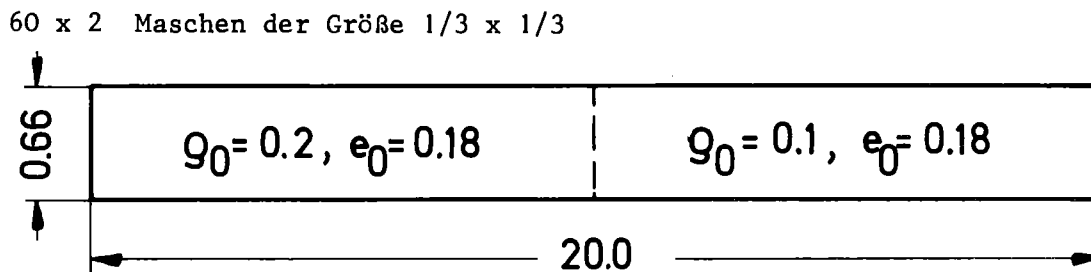


Abb. 16: Eindimensionales Stoßrohr

Abb. 17 zeigt den errechneten Verlauf von Druck, Dichte, innerer Energie und Geschwindigkeit nach 10 s im Vergleich zur analytischen Lösung. Die analytische Lösung ist u.a. in [10] beschrieben.  $\Delta t$  war 0.1 s.

Abb. 18 zeigt den Druckverlauf nach 10 s für 240, 120, 60, 20 und 10 Maschen in x-Richtung. Deutlich ist festzustellen, daß der Druckverlauf sich mit höheren Maschenzahlen der analytischen Lösung nähert. Insbesondere starke Gradienten können durch ein Differenzenverfahren nur dann wiedergegeben werden, wenn die Maschenweite klein ist gegen die Länge, in der sich eine Größe ändert.

Abb. 19 zeigt den Druckverlauf bei verschiedenen Zeitschrittweiten  $\Delta t = 0.1, 0.2, 0.4, 0.8$  und 1.4. Die größte Schrittweite 1.4 liegt dicht unterhalb der Stabilitätsgrenze. Durch eine kleinere Zeitschrittweite erhöht sich hier deutlich die Steilheit der Rampen, während sich eine große Zeitschrittweite dämpfend auswirkt und die Übergänge über mehrere Maschen verschmiert.

Interessant ist, daß die Überschwingung links des Stoßes mit abnehmender Zeitschrittweite zwar schmaler wird, die Amplitude aber unverändert bleibt.

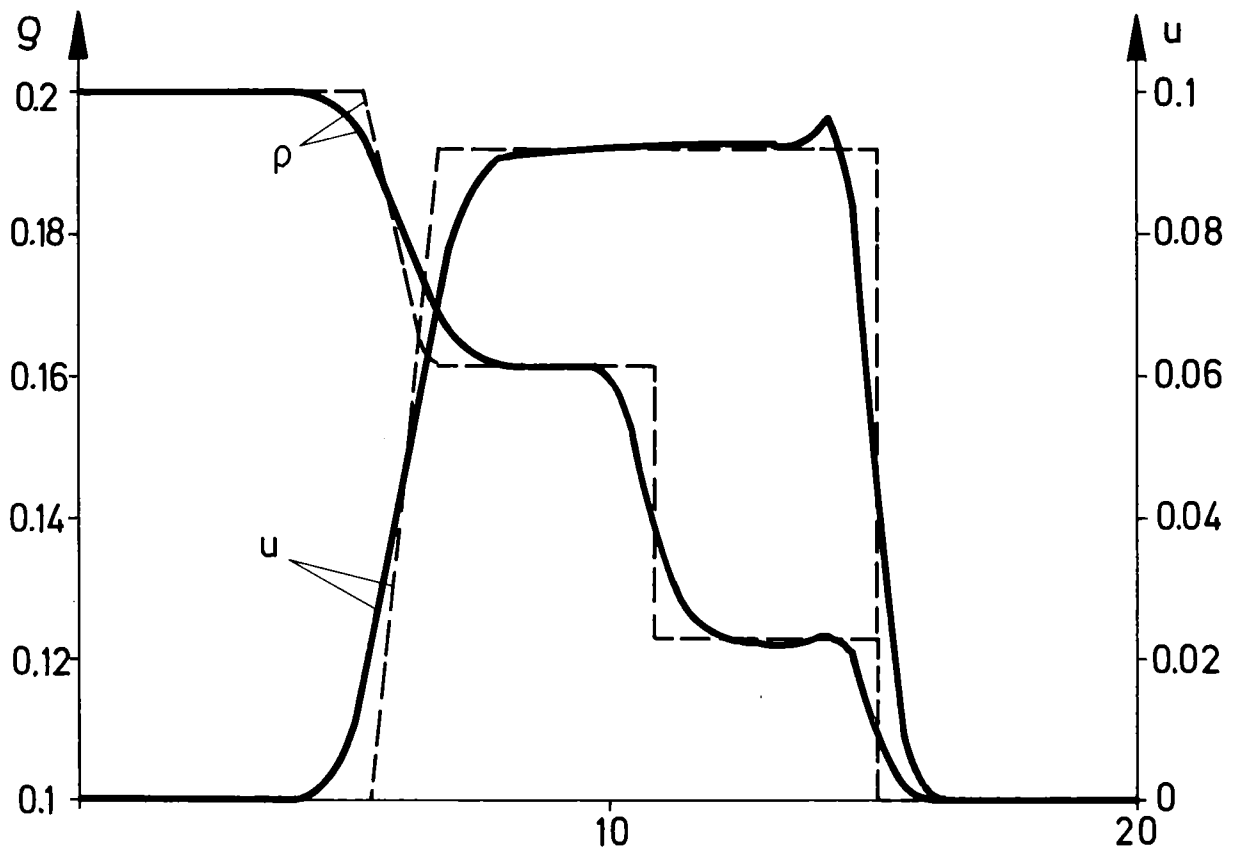
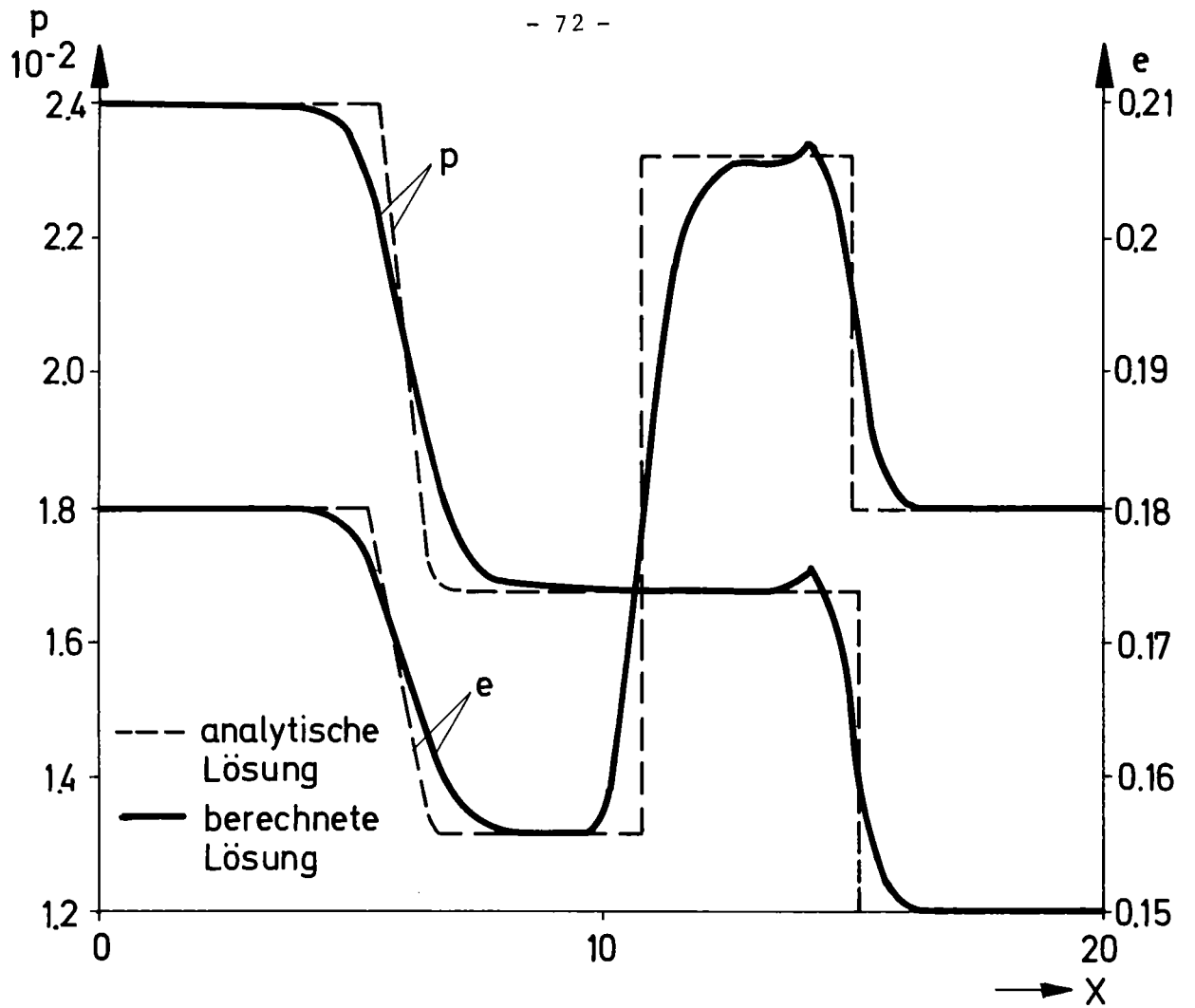


Abb. 17: Vergleich der analytischen und berechneten Lösung für ein ein-dimensionales Stoßrohr



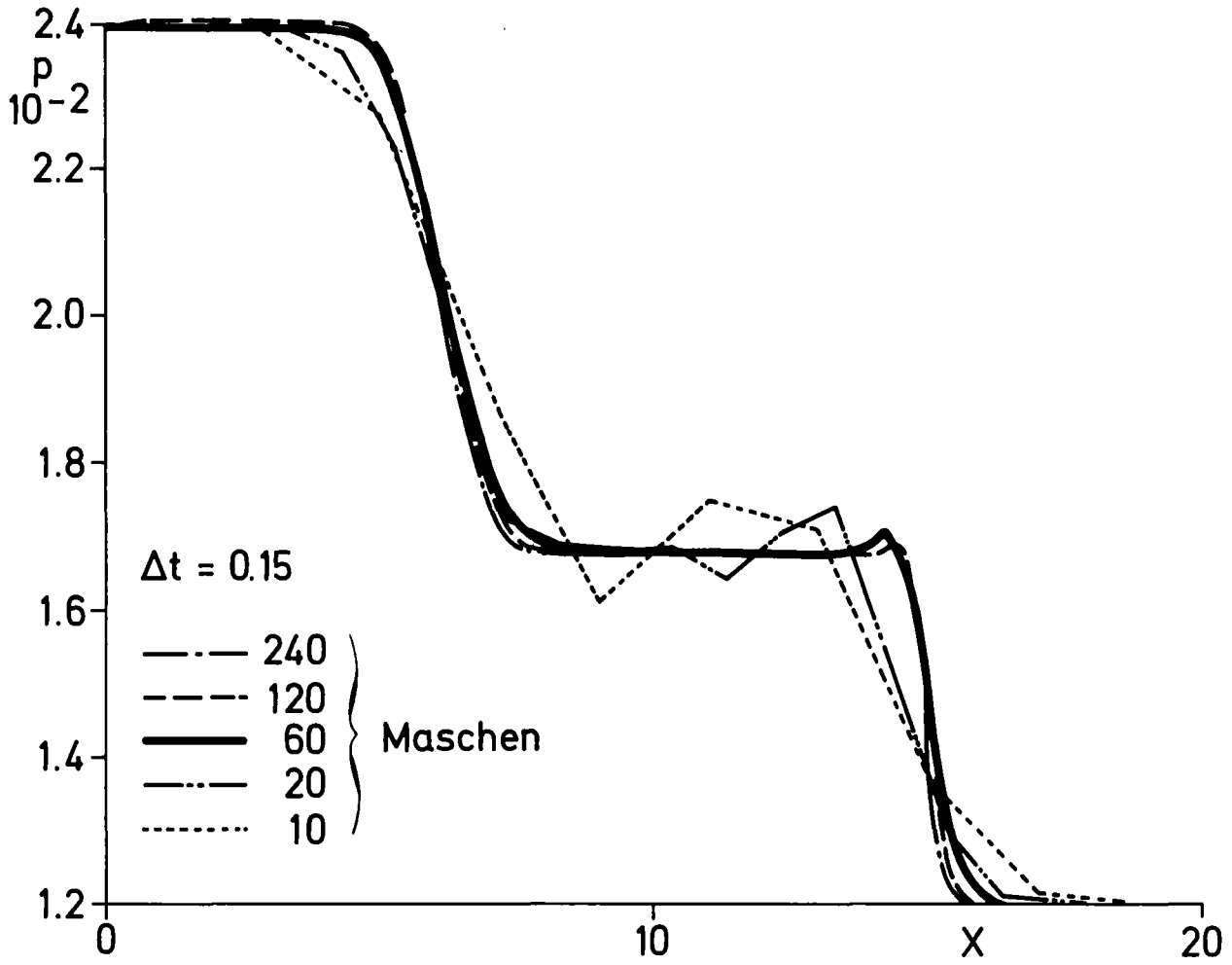


Abb. 18: Variation der Maschenweite

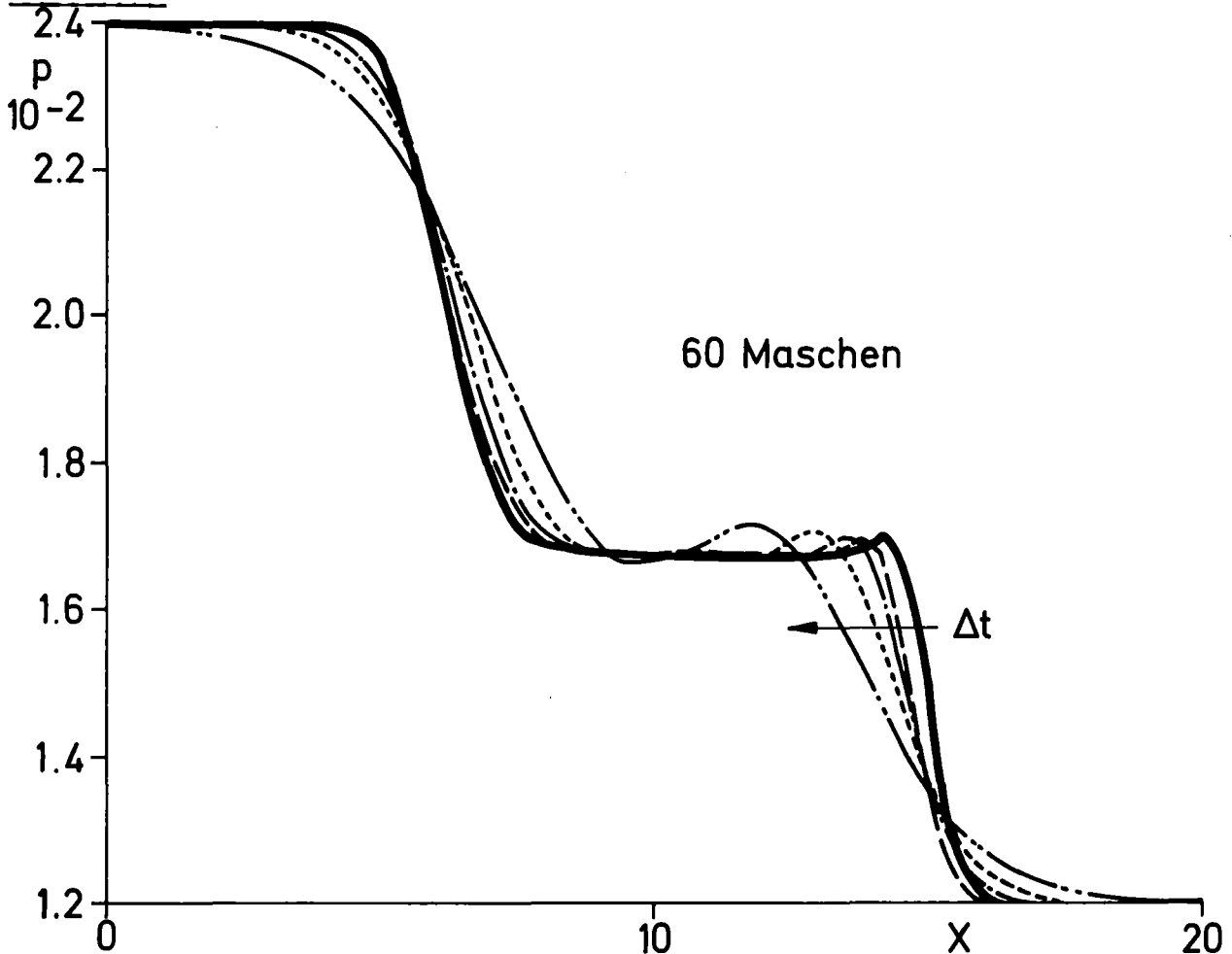
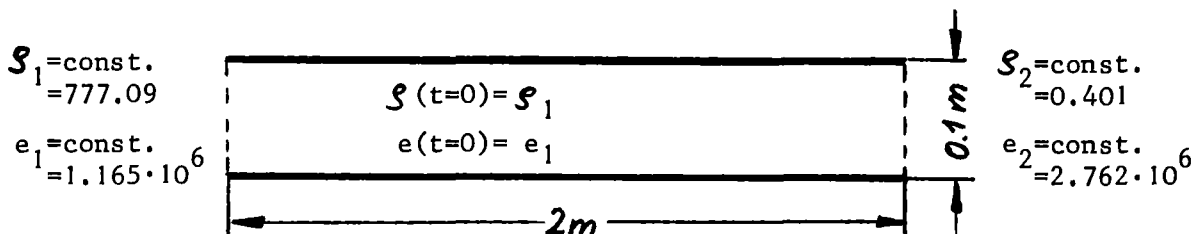


Abb. 19: Variation der Zeitschrittweite

## 7.2 Ausströmung von Wasser aus einem Behälter

Eine Strömung in einem Rohr wurde simuliert. An der Einströmseite wurde als Randbedingung der Druck 1.1 MPa bei  $T = 270\text{ C}$  vorgegeben, an der Ausströmseite ein Druck von 0.1 MPa. Abb. 20 zeigt die Rohrgeometrie und die Anfangsbedingungen.



das entspricht

$$p_1 = \text{const.} = 11.0 \text{ MPa}$$

$$T_1 = \text{const.} = 270^\circ\text{C}$$

$$x_1 = \text{const.} = 0$$

das entspricht

$$p_2 = \text{const.} = 0.1 \text{ MPa}$$

$$T_2 = \text{const.} = 270^\circ\text{C}$$

$$x_2 = \text{const.} = 1$$

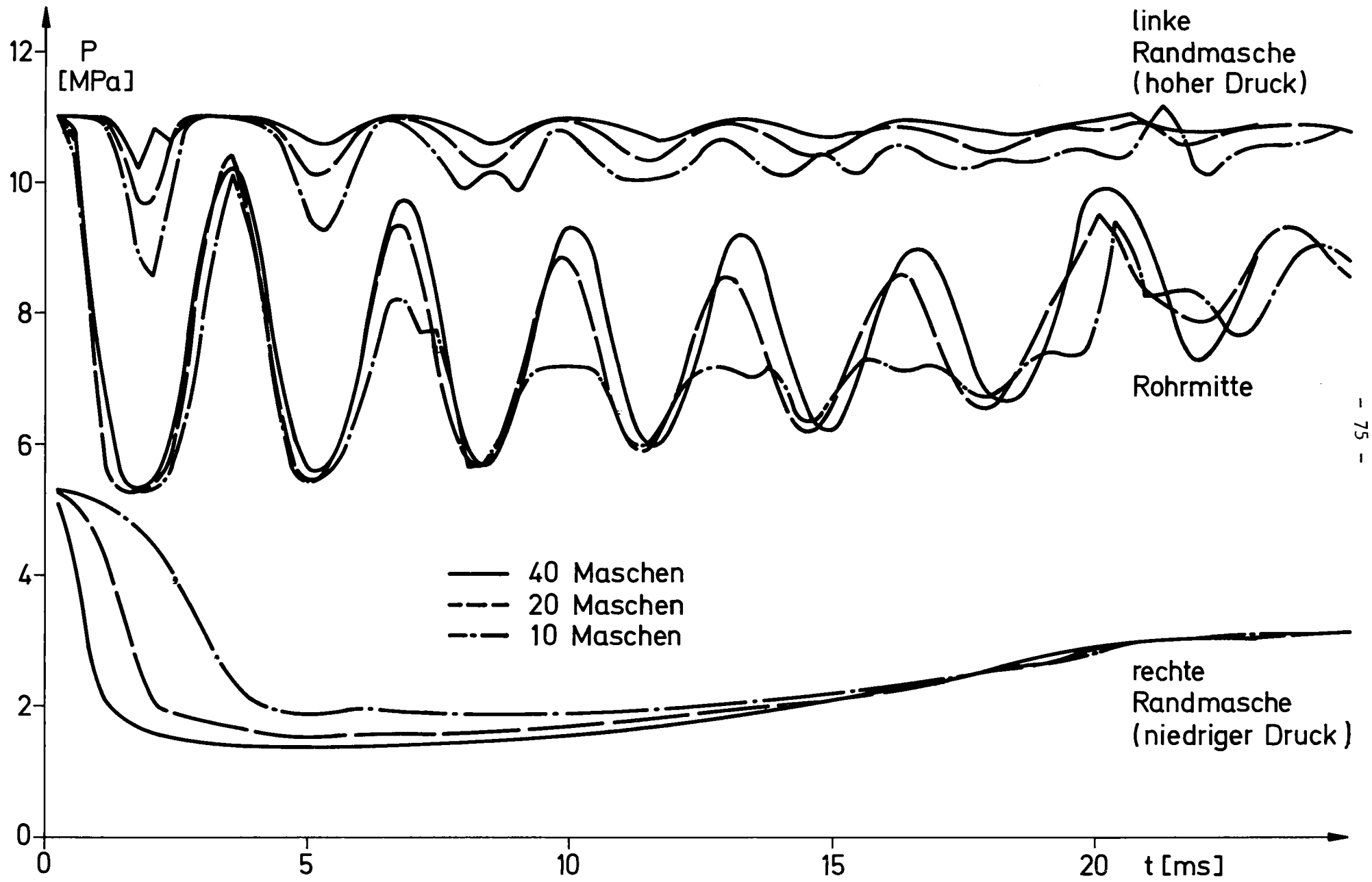
Abb. 20: Strömung in einem Rohr

Durch diese Randbedingungen wird eine stark beschleunigte Zweiphasenströmung erzeugt. Der Einfluß von Maschenweite und Zeitschrittweite wurde untersucht.

Abb. 21 zeigt den Einfluß der Maschenweite auf den Druckverlauf.

Der Druck in den beiden Randmaschen und in der Rohrmitte ist dargestellt für ein Rohr mit 10, 20 und 40 Maschen in x-Richtung, entsprechend Maschenweiten von 20, 10 und 5 cm. Die Zeitschrittweite wurde in allen drei Fällen zu  $\Delta t = 5 \cdot 10^{-5}$  s gewählt. Für diese Zeitschrittweite ist  $c \cdot \Delta t \approx 5$  cm, d.h. Schallwellen laufen während eines Zeitschrittes  $1/4$ ,  $1/2$  oder 1 Masche weit. Die Druckverläufe in der Rohrmitte zeigen deutlich die mit zunehmender Maschenweite stärker werdende Dämpfung der im Rohr hin- und herlaufenden Druckwellen. Die mit zunehmender Maschenweite stärker werdende Druckschwankung in den Randmaschen ist darauf zurückzuführen, daß die Entfernung zur Stelle konstanten Druckes auf dem Rand größer wird (sie beträgt eine Maschenweite).

Abb. 22 zeigt den Einfluß einer Variation der Maschenweite auf die Geschwindigkeiten im Rohr. Ein nennenswerter Einfluß ist lediglich für die Geschwindigkeit an der Ausströmstelle des Rohres festzustellen, sie nimmt mit



- 75 -

Abb. 21: Druck bei verschiedenen Maschenweiten

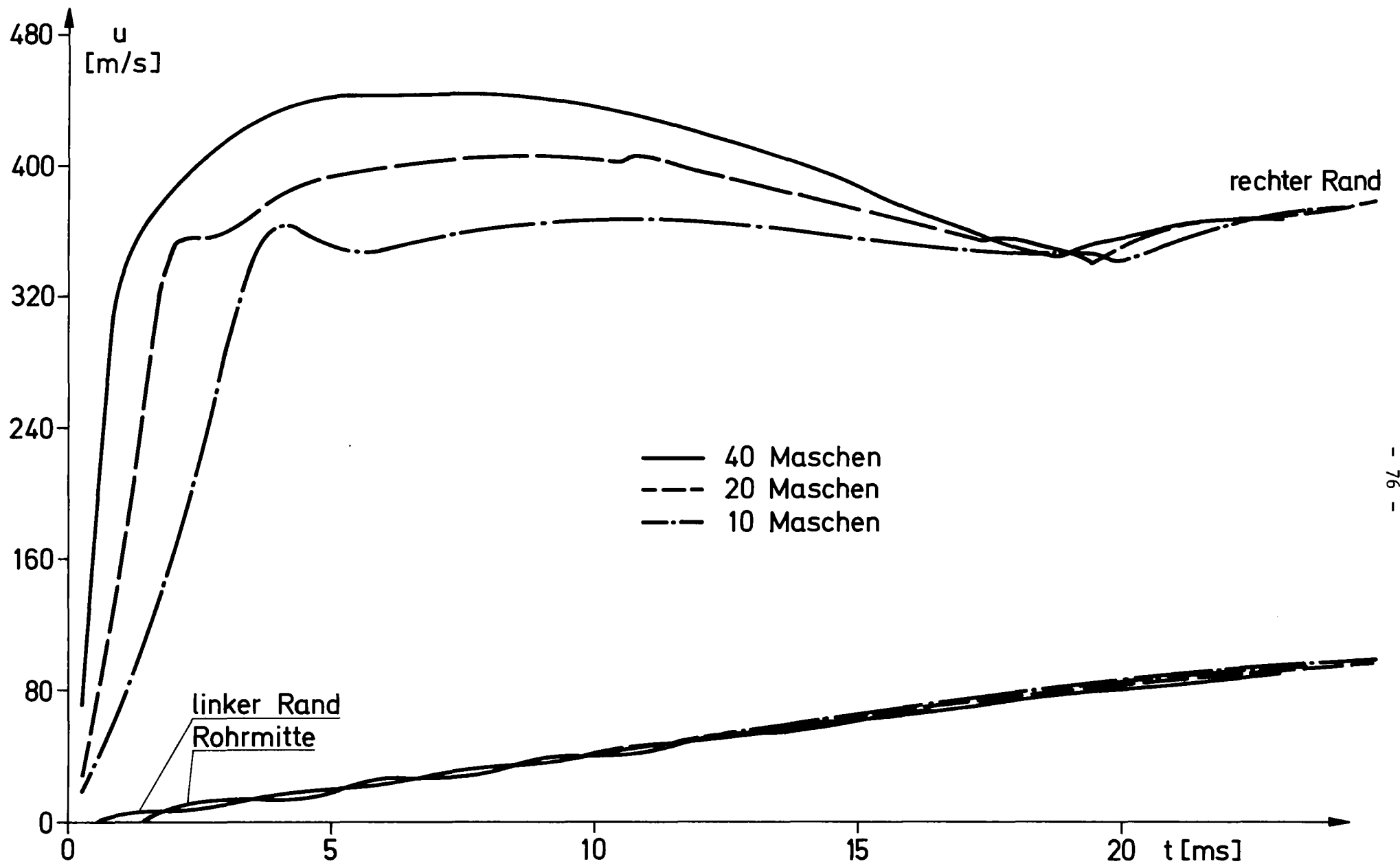


Abb. 22: Geschwindigkeit bei verschiedenen Maschenweiten

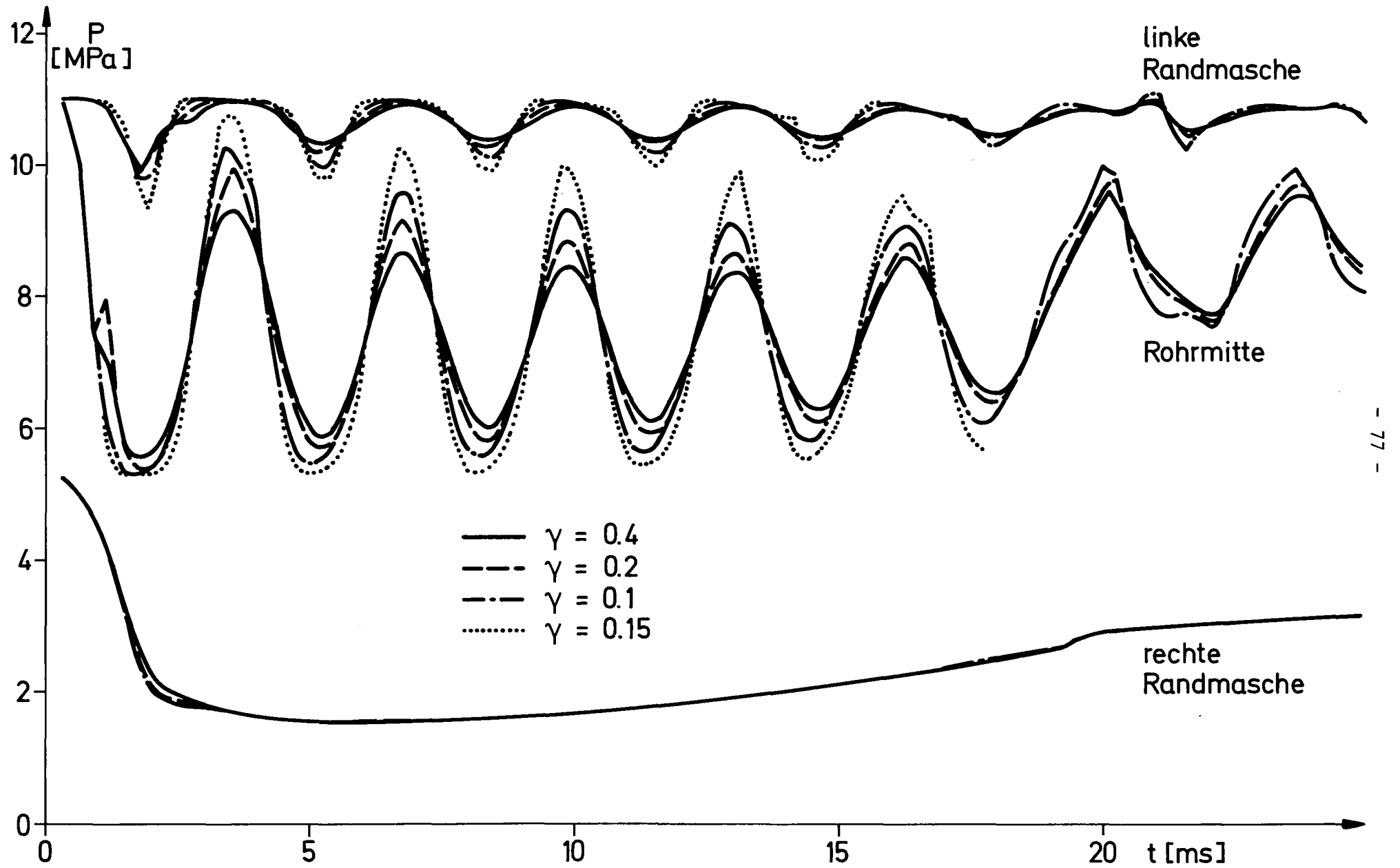


Abb. 23: Druck bei verschiedenen Zeitschrittweiten

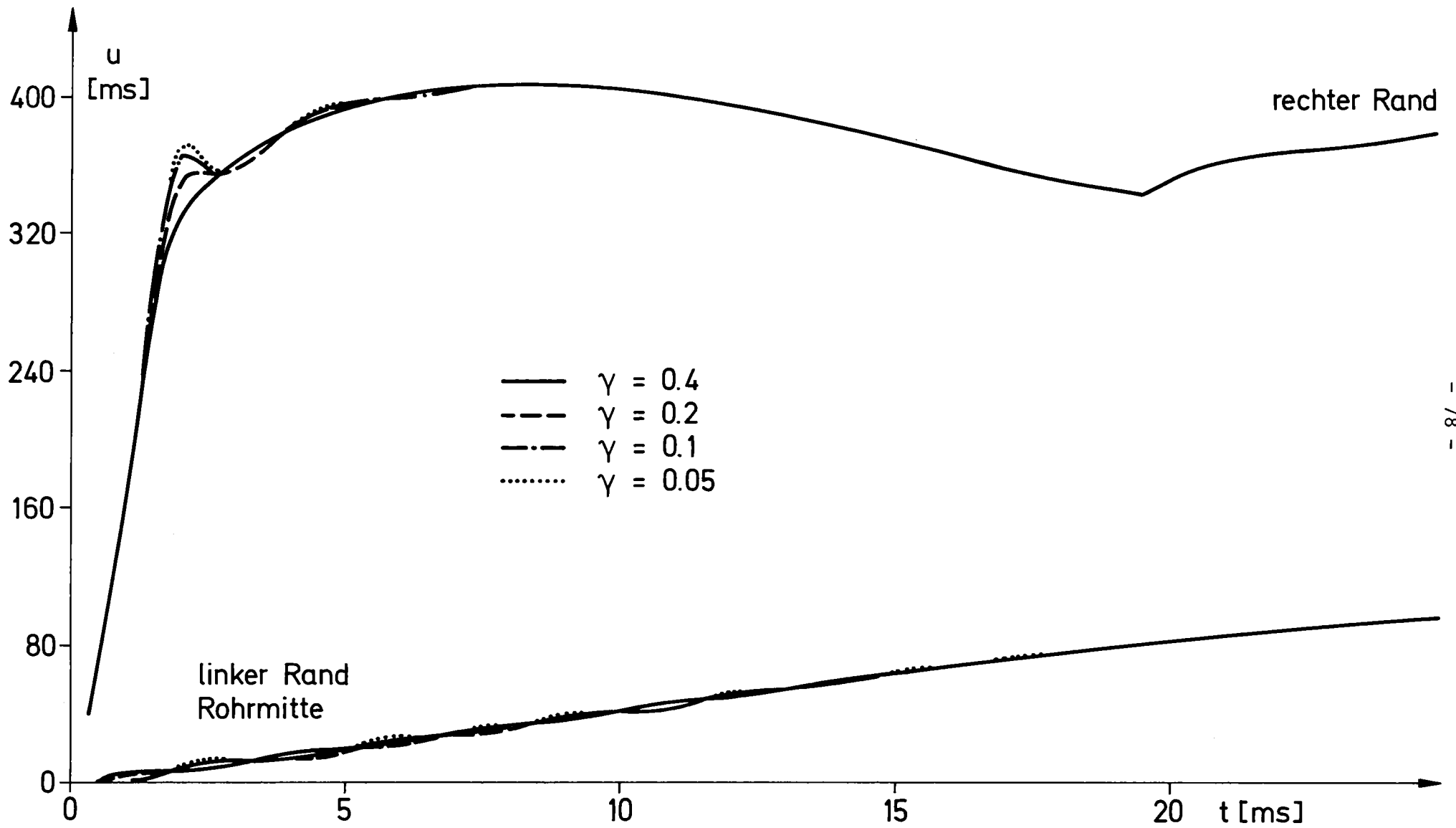


Abb. 24: Geschwindigkeit bei verschiedenen Zeitschrittweiten

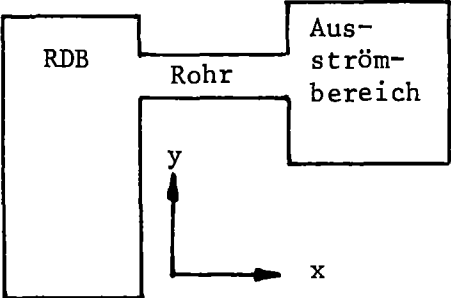
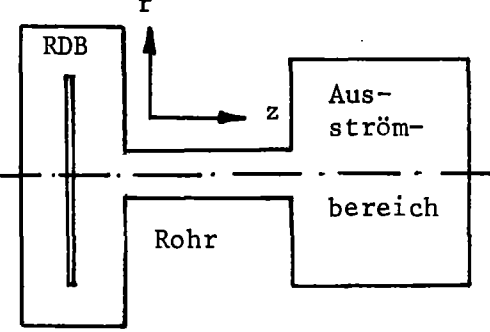
kleiner werdender Maschenweite zu. Die Geschwindigkeiten im Rohr ändern sich nur wenig mit der Maschenweite.

Abb. 23 und Abb. 24 zeigen den Einfluß der Zeitschrittweite auf Druck- und Geschwindigkeitsverlauf im Rohr. Es wurde der Parameter  $\gamma_1$  (vgl. Gl.(44), S.18) variiert, der die maximale Transportweite eines Teilchens innerhalb eines Zeitschrittes bestimmt. Die Rechnung wurde mit  $\gamma_1 = 0.4, 0.2, 0.1$  und  $0.05$  durchgeführt. Die Zeitschrittweite betrug zu Beginn stets  $\Delta t = 10^{-4}$  s. Durch die automatische Zeitschrittkontrolle erfolgte bei zunehmender Strömungsgeschwindigkeit eine Reduzierung. Ab 6 ms blieb die Zeitschrittweite in allen Fällen konstant, für die 4 Rechnungen war  $\Delta t$  ab diesem Zeitpunkt gleich  $5 \cdot 10^{-5}$  s,  $2.5 \cdot 10^{-5}$  s,  $1.25 \cdot 10^{-5}$  s und  $0.625 \cdot 10^{-5}$  s. Die Maschenzahl war 20. Im Druckverlauf ist die mit zunehmender Zeitschrittweite steigende Dämpfung deutlich zu beobachten. Die Geschwindigkeiten unterscheiden sich dagegen nur wenig, die Abweichung ist zwischen 3 und 20 ms immer kleiner als 2%. Lediglich die Spitze bei etwa 2 ms, die etwa 0.6 ms lang ist, kann bei großem  $\gamma_1$  nicht mehr aufgelöst werden.

### 7.3 Vergleich mit einem Zweiphasenprogramm

Um den Einfluß der Modellierung der Zweiphasenströmung auf die Rechenergebnisse zu zeigen, wurden die Ergebnisse der Berechnung eines Blowdown-Rohres mit den Resultaten des Zweiphasenprogramms DRIX-2D [11] verglichen. Das Problem ist dem HDR-Versuchsprogramm entnommen, die genaue Beschreibung der Modellgeometrie und der Anfangsbedingungen findet sich im folgenden Abschnitt 8.1 (S. 82). Die beim Bruch eines Stützens eines Druckwasserreaktors auftretende beschleunigte Zweiphasenströmung im Bruchstutzen und die Anströmung in das Containment wurde mit DRIX-2D und FLUST-2D berechnet.

Die Anfangsbedingungen waren Druck = 11 MPa und Temperatur = 270 °C. Am Außenrand des Ausströmereiches wurde ein Druck von 0.1 MPa als Randbedingung vorgegeben. Sowohl die Programme als auch die Modellierung des Problems unterscheiden sich erheblich:

FLUST-2D	DRIX-2D
<p>homogene Zweiphasenströmung, Gleichgewicht</p>	<p>Zweiphasenströmung im mecha- nischen und thermodynamischen Ungleichgewicht, 6-Gleichungsmodell</p>
 <p>Reaktordruckbehälter (RDB), Rohr und Ausströmbereich als zweidimensionale Recht- eckbereiche</p>	 <p>RDB, Rohr und Ausströmbereich als Zylinder, zweidimensional, Zylinderkoordinaten</p>
<p>Gleichgewichts-Wassergleichung mit <math>c^2 = c^2(g, e)</math> und <math>p = p(g, e)</math></p>	<p>je eine Zustandsgleichung im unterkühlten und Dampfgebiet, "gefrorene" Schallgeschwindig- keit im 2-Phasen-Bereich</p>

Trotz dieser Unterschiede zeigen die Ergebnisse eine zufriedenstellende Übereinstimmung. Abb. 25 zeigt den Druckverlauf im Rohr bei  $t = 0.6$  ms, 1.5 ms, 3 ms und 5 ms.

Die deutlichen Druckminima bei 1.5 und 3 ms, die durch die verzögerte Verdampfung in DRIX-2D entstehen, können naturgemäß durch die Gleichgewichts-Zweiphasenströmung in FLUST nicht dargestellt werden. Zu späteren Zeitpunkten nähern sich die Druckverläufe einander an, lediglich die verschiedenen Schallgeschwindigkeiten führen zu einer zunehmenden Phasenverschiebung. So deckt sich der von FLUST errechnete Verlauf bei  $t = 5.17$  ms gut mit dem Ergebnis von DRIX-2D bei  $t = 5$  ms.



Ringraum

offenes  
Rohrende

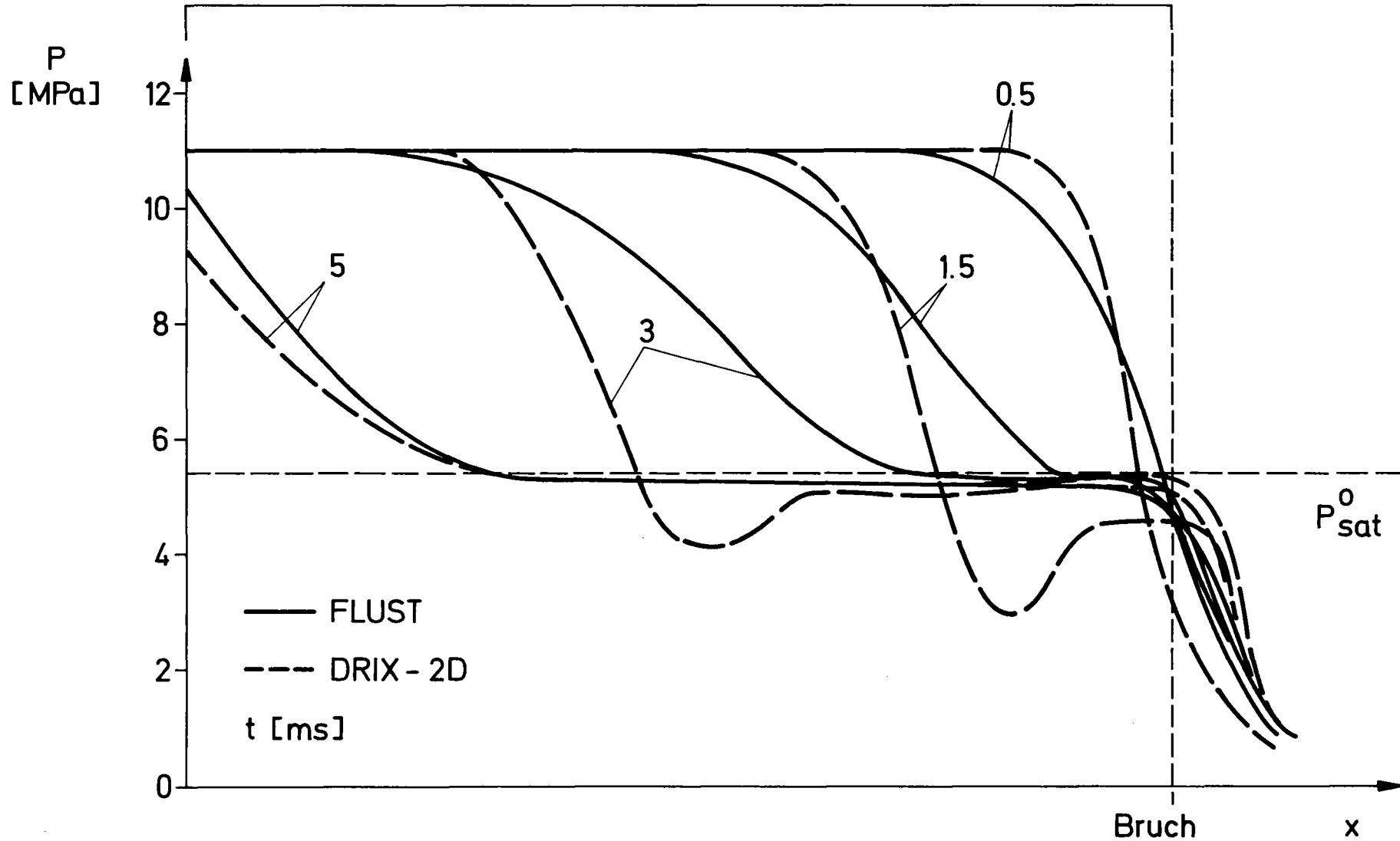


Abb. 25: Druckverlauf im Stutzen, berechnet von DRIX-2D und FLUST

## 8. Anwendungsrechnungen

### 8.1 Blowdown-Versuchsprogramm am HDR

Am stillgelegten Heißdampfreaktor HDR in Großwelzheim wird ein umfangreiches Versuchsprogramm zur Sicherheit von Druckwasserreaktoren (DWR) durchgeführt [6]. Ein Teil der Versuche befaßt sich mit den Vorgängen, die beim Bruch einer Hauptkühlmittelleitung an einem DWR auftreten. Der RDB des HDR hat etwa die gleiche Höhe wie ein DWR, der Durchmesser des RDB beträgt etwa 60% des DWR-Wertes. Wegen ihrer DWR-Ähnlichkeit sind daher die HDR-Versuche ein idealer Testfall für Programme zur Berechnung von Vorgängen in Druckwasserreaktoren. Die stark transiente Ausströmung von Wasser und Dampf aus dem Reaktor-druckbehälter durch den Bruchstutzen in das Containment wird auch kurz als Blowdown bezeichnet. Die wesentlichen Blowdownexperimente sind die Versuche 1 und 3, deren Anfangsbedingungen in Tabelle 2 zusammengefaßt sind. Die Temperaturverteilung wurde in [20] vorausberechnet, die im folgenden beschriebenen Blowdownrechnungen wurden mit der berechneten Anfangsbedingung für Temperatur durchgeführt. Die Erstellung der Temperaturschichtung soll die Temperaturverteilung in einem DWR bei Vollast simulieren. Der Versuch 1 mit der höheren Temperatur ist wegen des höheren Sättigungsdruckes der mildere. Bei beiden Versuchen ist die Temperatur im Innenraum am höchsten, so daß bei der Druckabsenkung während des Blowdowns ein Aufsieden im Innenraum auftritt, während im unteren Plenum und im Ringraum noch reine Wasserströmung herrscht.

Versuch 1	Versuch 3
Anfangsdruck $p^{\circ}$	
11 MPa	11 MPa
mittlere Temperatur im Ringraum $T_R^{\circ}$	
289 °C	268 °C
mittlere Temperatur im Innenraum $T_I^{\circ}$	
308 °C	305 °C
Anfangstemperatur im unteren Plenum $T_{UP}^{\circ}$	
288 °C	267 °C
Anfangstemperatur im oberen Plenum $T_{OP}^{\circ}$	
310 °C	310 °C
Anfangstemperatur im Stutzen $T_{St}^{\circ}$	
289.4 °C	269 °C
Sättigungsdruck zu T: $P_{sat}^{\circ}$	
$P_{R sat}^{\circ} = 7.35$ MPa	$P_{R sat}^{\circ} = 5.34$ MPa
$P_{I sat}^{\circ} = 9.6$ MPa	$P_{I sat}^{\circ} = 9.21$ MPa
$P_{UP sat}^{\circ} = 7.23$ MPa	$P_{UP sat}^{\circ} = 5.25$ MPa
$P_{OP sat}^{\circ} = 9.87$ MPa	$P_{OP sat}^{\circ} = 9.87$ MPa
$P_{St sat}^{\circ} = 7.38$ MPa	$P_{St sat}^{\circ} = 5.42$ MPa
Stutzenlänge $l$	
5.43 m	1.10 m
Stutzendurchmesser $d$	
0.2 m	0.2 m

Tabelle 2: Versuchsdaten für Blowdownversuch 1 und 3

## 8.2 Modellierung des HDR

Abb. 26 zeigt einen Querschnitt durch den RDB des HDR. Der HDR enthält während der Blowdownversuche zwar einen Kernmantel mit einem die Masse des Kerns simulierenden Massering am unteren Ende, jedoch keinen Reaktorkern. Alle Stutzen bis auf den Bruchstutzen sind verschlossen. Abb. 27 zeigt die Modellierung von Ringraum und Bruchstutzen. Im Ringraum überwiegen zweidimensionale Effekte in der Umfangsrichtung und in axialer Richtung, er wurde daher in seiner Abwicklung modelliert. Wegen der Abwesenheit intakter Stutzen ist das Problem symmetrisch, es muß also nur eine Hälfte des Ringraumes berechnet werden. Der Stutzen wird in die Ringraumebene geklappt und ebenfalls zweidimensional kartesisch modelliert. Die durchströmte Fläche am Übergang Ringraum - Stutzen muß in der Wirklichkeit und im Modell gleich sein:

$$h \cdot b = \frac{1}{2} r^2 \pi \quad (79)$$

h = Stutzhöhe im 2D-Modell

b = Ringraumbreite

r = Stutzenradius (real)

Mit dieser Gleichung wird die Stutzhöhe im x-y-Stutzen-Modell bestimmt. Der Faktor 1/2 berücksichtigt, daß die zweite Ringraumhälfte, die aus Symmetriegründen nicht mitgerechnet wird, ebenfalls einen Stutzen besitzt.

Das Ringraummodell besteht aus 11 x 20 Maschen. Die kleinsten Maschen in Stutzennähe sind 53 mm weit (= 1/2 h). Die größten Maschen sind 500 mm weit. Der Stutzen ist in der Höhe durch 2 Maschen aufgelöst, in seiner Länge durch 16 Maschen für Versuch 1 und durch 8 Maschen für Versuch 3. Die Maschen am Stutzenende, deren Größe einen Einfluß auf die Ausströmung hat (vgl. Abb. 22), sind 100 mm groß. Das Stutzenende ist mit einem Ausströmereich verbunden, das untere Ringraumende ist an das Modell des unteren Plenums angeschlossen.

Die zweidimensionalen Vorgänge im Ringraum können durch die beschriebene Modellierung gut aufgelöst werden, die Strömung im Rohr wird zufriedenstellend dargestellt. Am Übergang Stutzen-Ringraum kann jedoch die in der Wirklichkeit bestehende Umlenkung der Strömung aus dem Ringraum in den Stutzen nicht modelliert werden.

In Abb. 28 ist der Aufbau des Modells von Innenraum und Plena gezeigt. Zunächst werden aus unterem und oberem Plenum Zylinder gleicher Volumina abgeleitet.

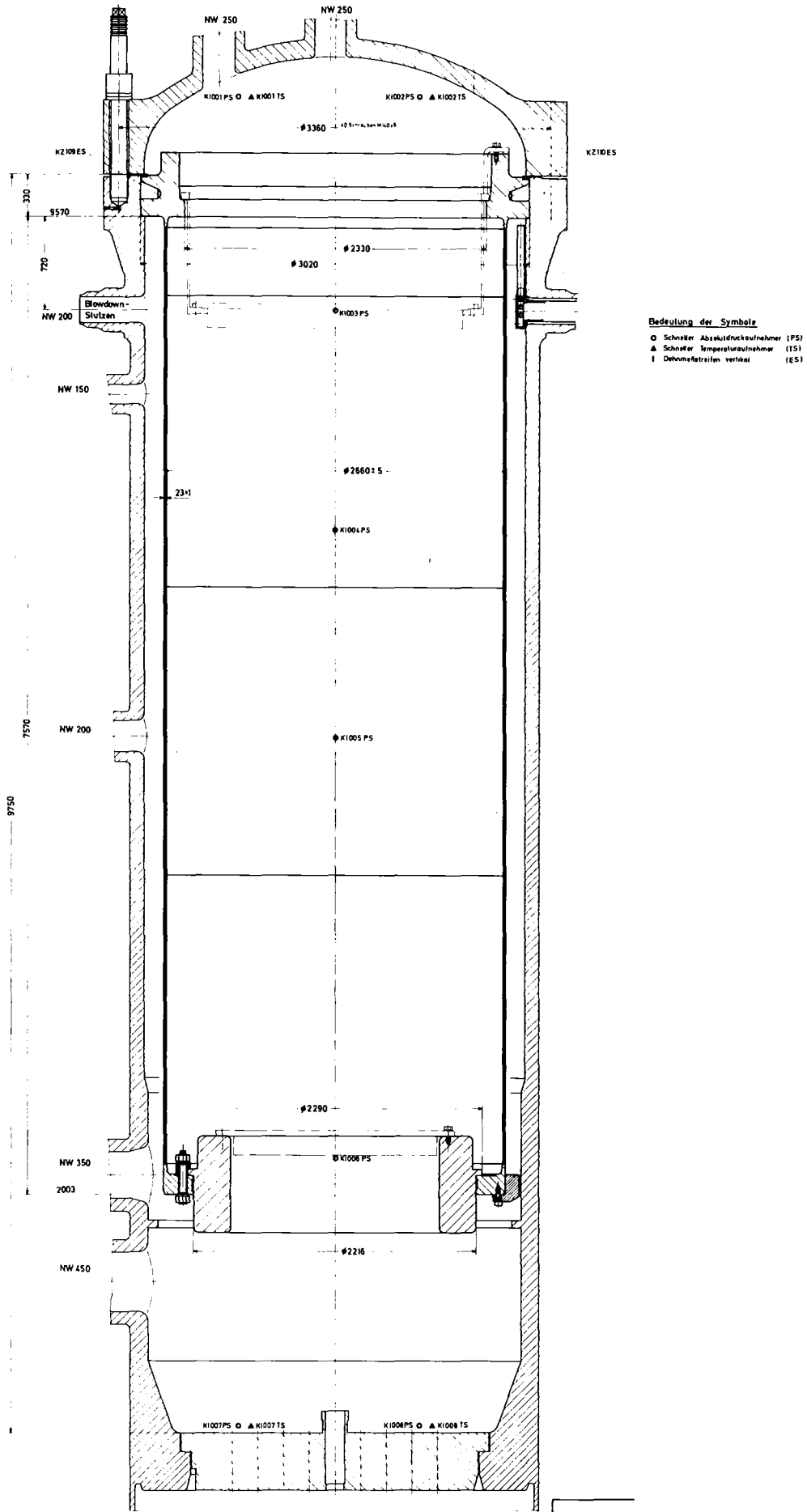
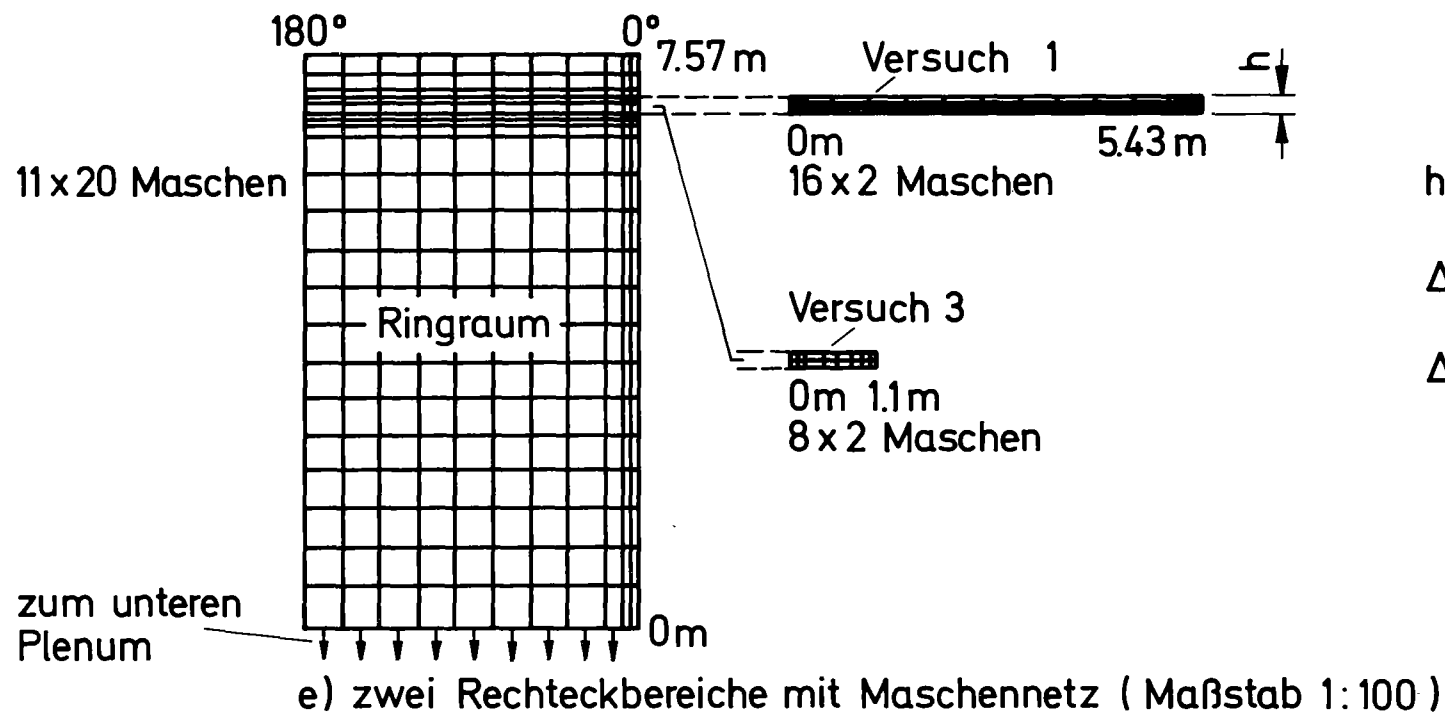
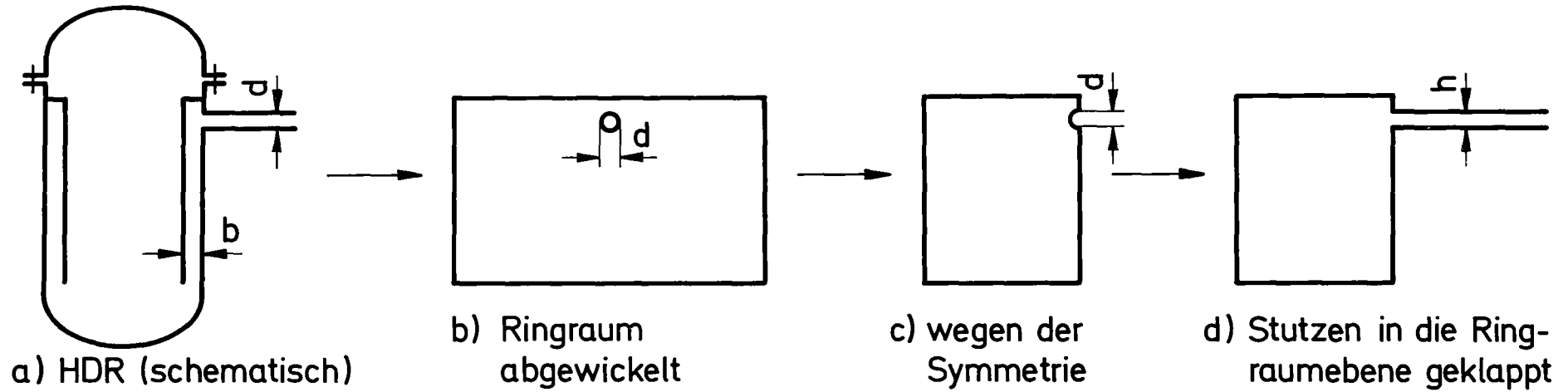


Abb. 26: Querschnitt durch den Reaktordruckbehälter des HDR

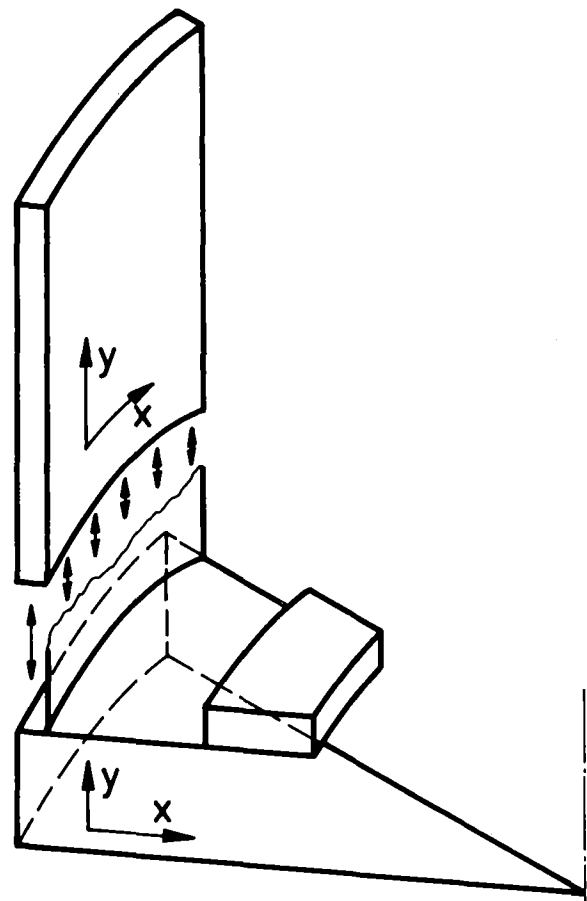
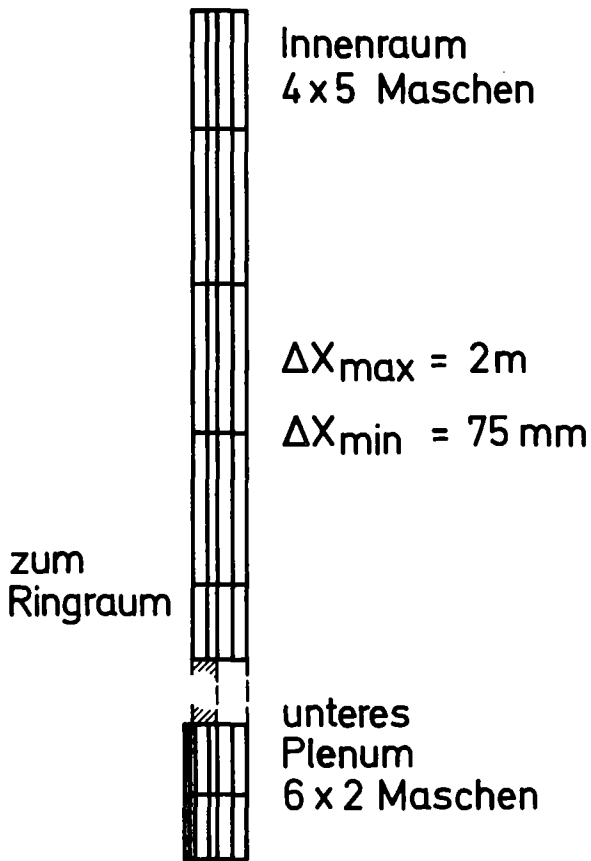
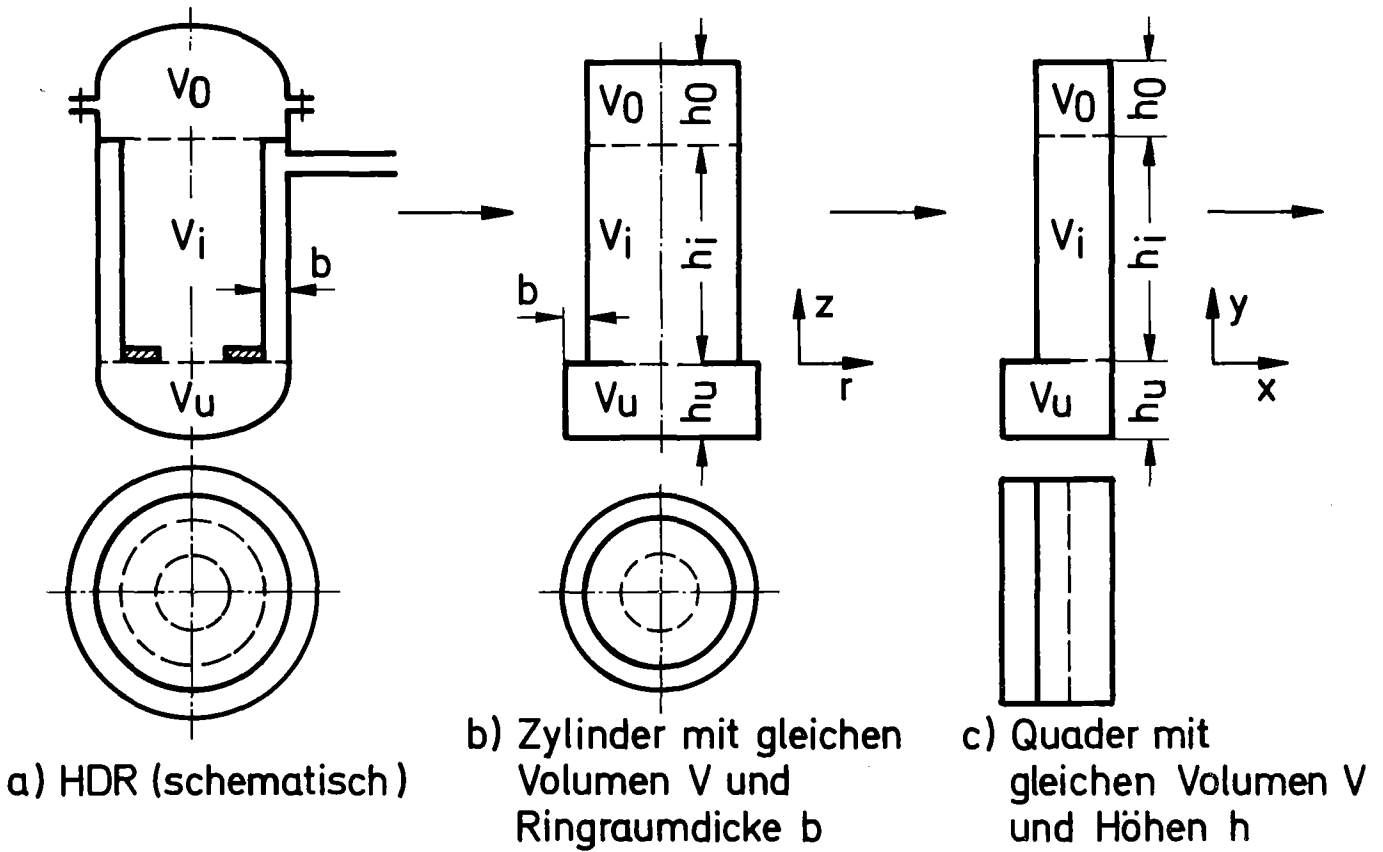


$$h = \frac{1}{2} \cdot \frac{\left(\frac{d}{2}\right)^2 \pi}{b}$$

$$\Delta X_{\min} = 53 \text{ mm}$$

$$\Delta X_{\max} = 500 \text{ mm}$$

Abb. 27: Modellierung von Ringraum und Bruchstützen



d) Zwei Rechteckbereiche (Maßstab 1:100)

e) Verbindung Ringraum - unteres Plenum

Abb. 28: Modellierung von Innenraum und Plena

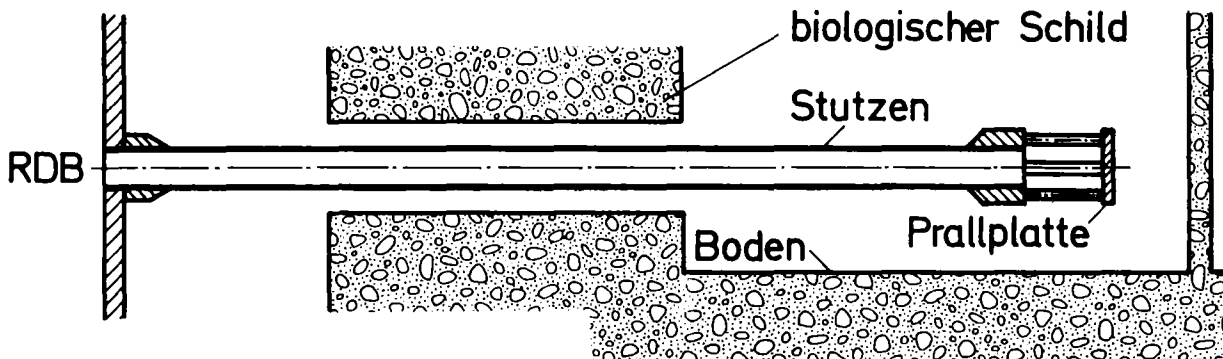
leitet. Das obere Plenum erhält den gleichen Radius wie der Innenraum, das untere Plenum einen um die Ringraumdicke größeren. Da FLUST-2D nur kartesische Koordinaten enthält, werden die Zylinder in Quader gleicher Volumina überführt, die Tiefe der Quader entspricht dabei dem Umfang des Ringraumes. Um Laufzeiten in axialer Richtung wiedergeben zu können, wurden die Höhen unvermindert beibehalten. Aus Innenraum und oberem Plenum wurde ein Bereich mit  $4 \times 5$  Maschen gebildet, aus dem unteren Plenum ein zweiter Bereich mit  $6 \times 2$  Maschen. Zwischen Innenraum und unterem Plenum befindet sich beim HDR der Massenring, der die Masse des Kerns darstellt. Dieser Massenring verschließt einen Teil der Öffnung vom Innenraum zum unteren Plenum. Zwischen den beiden Bereichen wurde daher eine feste Wand (mit Haftung) eingefügt, die einen Teil der Verbindung verschließt. Innenraum und Plena sind so durch ein zweidimensionales Maschennetz dargestellt, das in radialer und axialer Richtung ausgedehnt ist. Das Ringraummodell, das in axialer und Umfangsrichtung ausgedehnt ist, muß mit einer Drehung von  $90^\circ$  angeschlossen werden. Im Ringraum wird die Variation in radialer Richtung, im Innenraum und in den Plena die Variation in Umfangsrichtung vernachlässigt. Die Verbindung geschieht durch einen besonderen Typ von Rand.

Beim Übergang Plenum  $\rightarrow$  Ringraum wird in jeder Ringraummasche als Nachbarwert der Durchschnitt der beiden Randmaschen im unteren Plenum benutzt. Beim Übergang Ringraum  $\rightarrow$  unteres Plenum wird der Durchschnitt der 11 Randmaschen unten im Ringraum als Nachbarwert für die beiden Randmaschen im Plenum benutzt.

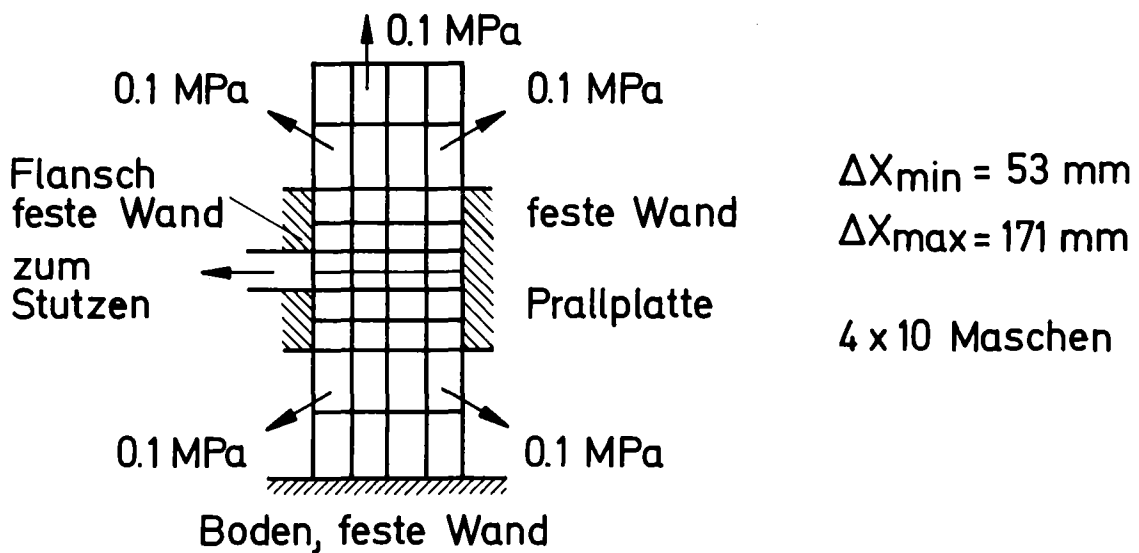
Abb. 29 zeigt die Diskretisierung des Ausströmbereiches für Versuch 1. Dabei wird der Stutzen durch eine Bohrung im biologischen Schild geführt. 400 mm vom Stutzenende entfernt ist eine Prallplatte angebracht, das Stutzenende befindet sich 550 mm über dem Boden. Das Modell berücksichtigt die Prallplatte, den Flansch am Stutzenende und den Boden als feste Wände, an den offenen Stellen wurde ein Druck von 0.1 MPa bei einer Temperatur von 100 C als Randbedingung festgehalten. In Abb. 30 ist das Modell des Außenbereiches für Versuch 3 dargestellt. Der Stutzen endet bei diesem Versuch 140 mm vor der Bohrung im biologischen Schild. Der Schild ist als feste Wand im Modell berücksichtigt. An den offenen Stellen, insbesondere am Ende der Bohrung, wurde als Randbedingung ( $P_R = 0.1 \text{ MPa}$ ,  $T = 100^\circ\text{C}$ ) festgehalten. Die Anfangsbedingung im Ausströmbereich ist gleich der Randbedingung.

Versuche haben gezeigt, daß die Modellierung des Außenbereiches keinen für die Vorgänge im RDB und Stutzen wesentlichen Einfluß ausübt. Ein rechteckförmiger Ausströmbereich von  $3 \times 3 \text{ m}$  Größe mit der Randbedingung  $P = 0.1 \text{ MPa}$ ,



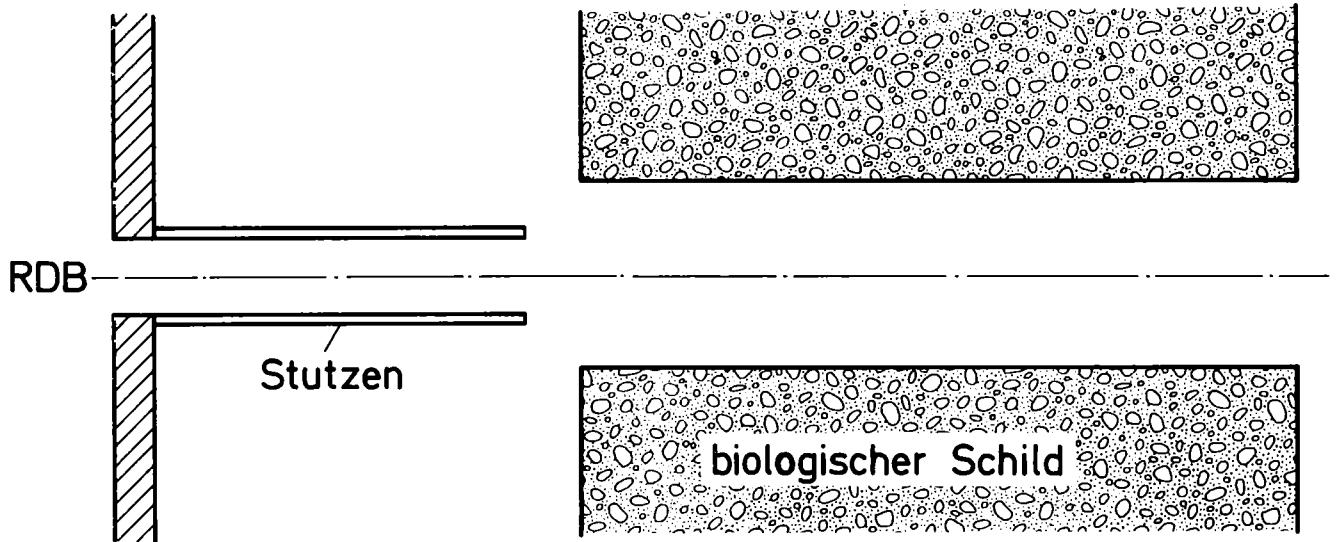


Anordnung des Bruchstützens (schematisch)

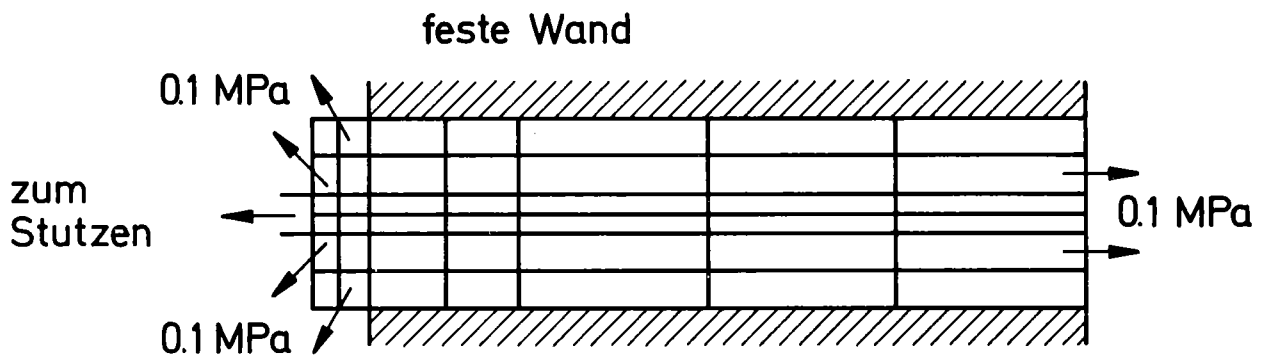


Maschennetz des Außenbereiches (Maßstab 1:20)

Abb. 29: Modellierung des Ausströmbereiches für Versuch 1



Anordnung des Bruchstutzens (schematisch)



Maschennetz des Außenbereiches ( Maßstab 1:20 )  
 $\Delta X_{\min} = 70 \text{ mm}$        $\Delta X_{\max} = 500 \text{ mm}$

Abb. 30: Modellierung des Ausströmbereiches für Versuch 3

$T = 100^{\circ}\text{C}$ ) an allen Seiten ist für die Vorgänge in RDB und Stutzen gleichwertig. Auch eine Variation der Randbedingung ergab nur Unterschiede, die in den Computerzeichnungen der Zeitverläufe nicht feststellbar waren. Die Randwerte für die Vergleichsläufe waren ( $P_R = 0.5 \text{ MPa}$ ,  $T = T_{\text{sat}}(P_R)$ ) und ( $P_R = 2 \text{ MPa}$ ,  $T = T_{\text{sat}}(P_R)$ ). Lediglich die Ausströmgeschwindigkeit in der ersten Stutzenmasche war bei 2 MPa bis etwa  $t = 10 \text{ ms}$  geringer, die Differenz betrug bei Versuch 3 maximal 7%. Dieser Unterschied ist darauf zurückzuführen, daß mit der geänderten Randbedingung auch die Anfangsbedingung im Außenbereich auf die Randwerte gesetzt wurde.

Als Zustandsgleichung wurde die genaue Wassergleichung (s. Kap. 3.4, 5.12) benutzt. Eine Vergleichsrechnung mit der linearen Wassergleichung zeigt erhebliche Unterschiede. Die Ergebnisse der Vorausrechnungen von HDR-Blowdownversuch 1 und 3 sind im folgenden Abschnitt dokumentiert.

Es wurde in allen Fällen mit der maximal möglichen Zeitschrittweite gerechnet ( $\gamma_1 = 0.5$ ).

### 8.3 Ergebnisse

In den Abbildungen 31 bis 68 sind die Ergebnisse der HDR-Vorausrechnungen in Form von Zeitverläufen  $f(t)$  für verschiedene Meßpunkte und in Form von Reliefs  $f(x,y)$  für verschiedene Zeitpunkte dargestellt. Tabelle 3 gibt einen Überblick über die Abbildungen zur Ergebnisdarstellung.

Die Abbildungen 31 bis 47 zeigen Ergebnisse des Versuches 3 mit den Anfangsbedingungen  $p_0 = \text{MPa}$  und  $T_0 = 270^{\circ}\text{C}$ .

#### Abb. 31: Druckverlauf im Blowdownstutzen.

In der letzten Masche herrscht praktisch von Anfang an Sättigungsdruck. Der Sättigungsdruck liegt hier wegen der Temperaturabnahme tiefer als in den anderen Rohrmaschen, in denen bei etwa 35 ms Sättigungszustand erreicht wird. Ab diesem Zeitpunkt ist im gesamten Rohr eine Zweiphasenströmung vorhanden. Die Schwingungen bis etwa 10 ms rühren von der Druckwelle her, die jeweils als Entlastungswelle in den Stutzen hineinläuft und als Welle mit zunehmendem Druck zurückkehrt. Bei 55 und 85 ms sind Druckspitzen zu erkennen, die nicht auf einen physikalischen Effekt, sondern auf das numerische Verfahren zurückzuführen sind. Die für die Lösung erforderliche Schallgeschwindigkeit im Gleichgewicht hat beim Übergang vom Wasser- zum Zweiphasengebiet einen Sprung, bei den vorliegenden Bedingungen von etwa 1000 m/s auf 50 m/s. Da während der Druckiteration mit der vorher einmal berechneten Schallgeschwindigkeit

Abbildung	Versuch	Größe	Zeitverlauf/ Relief	Ort/ Zeitpunkt
31	3	$p(t)$	Z	Stützen
32	3	$p(t)$	Z	Ringraum
33	3	$p(t)$	Z	unt. Plenum+Innenr.
34	3	$u(t)$	Z	Stützen
35	3	$x(t)$	Z	Stützen
36	3	$x(t)$	Z	Innenraum
37	3	$T(t)$	Z	Stützen
38	3	$T(t)$	Z	Innenraum
39	3	$p(x,y)$	R	$t=0$ ms
40	3	$p(x,y)$	R	2,6 ms
41	3	$p(x,y)$	R	3,8 ms
42	3	$p(x,y)$	R	4,4 ms
43	3	$p(x,y)$	R	6,8 ms
44	3	$p(x,y)$	R	9,8 ms
45	3	$p(x,y)$	R	24,5 ms
46	3	$p(x,y)$	R	49,7 ms
47	3	$p(x,y)$	R	99,5 ms
48	1	$p(t)$	Z	Stützen
49	1	$p(t)$	Z	Ringraum
50	1	$p(t)$	Z	unt. Plenum+Innenr.
51	1	$u(t)$	Z	Stützen
52	1	$x(t)$	Z	Stützen
53	1	$T(t)$	Z	Stützen
54	1	$T(t)$	Z	Innenraum
55	1	$p(x,y)$	R	$t=0$
56	1	$p(x,y)$	R	$t=1,6$ ms
57	1	$p(x,y)$	R	$t=3$ ms
58	1	$p(x,y)$	R	$t=5$ ms
59	1	$p(x,y)$	R	$t=8$ ms
60	1	$p(x,y)$	R	$t=16,4$ ms
61	1	$p(x,y)$	R	$t=25$ ms
62	1	$p(x,y)$	R	$t=49,2$ ms
63	1	$p(x,y)$	R	$t=65,4$ ms
64	1	$p(x,y)$	R	$t=90$ ms
65	3	$p(t)$	Z	Stützen
66	3	$p(t)$	Z	Ringraum
67	3	$p(t)$	Z	Innenraum
68	3	$u(t)$	Z	Stützen

Tabelle 3: Abbildungen mit Rechenergebnissen

gerechnet wird, kann es vorkommen, daß während der Iteration von einem in den anderen Bereich hineiniteriert wird. Dies führt dann im folgenden Zeitschritt zu einer erheblichen Druckdifferenz. Der Vergleich mit Abb. 35 zeigt, daß die Druckspitzen immer dann auftreten, wenn der Dampfgehalt auf Null zurückgeht, also beim Übergang vom Zweiphasengebiet ins Wasser. Eine Rechnung mit einer um den Faktor 10 kleineren Zeitschrittweite brachte die numerischen Druckspitzen fast zum Verschwinden.

Abb. 32: Druck im Ringraum.

In der Stutzennähe ist noch deutlich die Druckschwingung zu sehen, die von der im Stutzen hin- und herlaufenden Entlastungswelle herrührt. Auch die im Stutzen erzeugten numerischen Druckspitzen wirken sich hier noch aus. Im übrigen wird der Druckverlauf im Ringraum dominiert von der auf- und ablaufenden Druckwelle, die an der Ringraumoberseite an der geschlossenen Wand und am unteren offenen Ende reflektiert wird. Die Schwingungsdauer ist etwa 30 ms, das entspricht  $4 \cdot h_{\text{Ringraum}}/c$ .

Abb. 33: Druck im unteren Plenum und Innenraum.

Hier ist noch deutlich der Einfluß der Druckschwingung im Ringraum zu beobachten. Die Schwankungen um etwa 0,1 MPa am Anfang sind darauf zurückzuführen, daß zwar oben im Innenraum und wieder am unteren Plenum der Anfangsdruck von 11 MPa als Anfangsbedingung richtig eingestellt wurde, aber in der Ringraummitte der Anfangsdruck tiefer lag. Die Anfangsbedingungen sind gegeben durch den Druck und die Temperaturschichtung, das Programm erwartet jedoch Dichte und innere Energie als Anfangswerte. Durch lineare Interpolation von  $\rho_0$  und  $\rho_0$  oben und unten im Innenraum ergab sich die Druckabweichung in der Innenraummitte, die für den späteren Verlauf unbedeutend ist. Ab 50 ms beginnt das Wasser oben im Ringraum aufzusieden, der Druck bleibt konstant auf Sättigungsdruck. Zeitlich danach wird dadurch auch die Druckabnahme unten im Ringraum und im unteren Plenum aufgehalten und der Druck schwingt um den durch das Sieden aufgeprägten Sättigungsdruck.

Abb. 34: Geschwindigkeit im Stutzen.

Nur direkt an der Bruchstelle und eine Masche weiter links liegt die Geschwindigkeit über der Schallgeschwindigkeit, an allen anderen Stellen im Rohr ergibt sich der gleiche Geschwindigkeitsverlauf wie der in der Rohrmitte zeigte, der etwa der vom Programm benutzten Schallgeschwindigkeit im Zweiphasengebiet entspricht.

Abb. 35: Massendampfgehalt im Stutzen.

Hier ist deutlich zu sehen, daß nur in der letzten Masche stets der Dampfgehalt größer Null ist. In den anderen Maschen herrscht zeitweise reine Wasserströmung. In der Rohrmitte ergibt sich erstmals ab 40 ms eine Zweiphasenströmung.

Abb. 36: Massendampfgehalt im Innenraum.

Nur in der obersten Masche des Innenraumes ist ab 55 ms ein Aufsieden des Wassers zu erkennen, der Dampfanteil steigt ab diesem Zeitpunkt stetig an. Die Druckverläufe sind dann entscheidend von diesem Siedevorgang geprägt.

Abb. 37 zeigt die Temperatur im Blowdown-Stutzen, Abb. 38 die Temperatur im Innenraum, die dort mit einer Rate von etwa 10 K/s abnimmt.

Die Abbildungen 39 bis 47 zeigen das Druckfeld in Reliefdarstellungen im gesamten System zu verschiedenen Zeitpunkten. Die dynamischen Vorgänge werden dadurch besonders hervorgehoben.

Die Abbildungen 48 bis 64 zeigen Ergebnisse von HDR-Blowdown-Versuch 1 ( $p_0 = 11 \text{ MPa}$ ,  $T_0 = 290^\circ\text{C}$ ). Durch die mildereren Anfangsbedingungen geschieht die Druckabnahme langsamer. Der lange Stutzen bei diesem Versuch hat ebenfalls einen prägenden Einfluß auf die Ergebnisse.

Abb. 48: Die Druckschwingung im Stutzen dauert länger an als bei Versuch 3. Mehrfach geht der Druck auf Sättigung und nimmt dann wieder zu.

Abb. 49: Druck im Ringraum.

Die bis 100 ms andauernden Druckschwingungen sind zu Beginn durch die Druckentlastungswelle im Rohr induziert. Da die Zeitdauer dieser Schwingung etwa mit der Dauer der im Ringraum in Umfangsrichtung laufenden Welle übereinstimmt, bleibt sie lange zu beobachten. Zu Beginn des Blowdowns ist deutlich die Laufzeit der Entlastungswelle zu verfolgen.

Abb. 50: Druck im Innenraum und unterem Plenum.

Bei Versuch 1 wird erst bei etwa 95 ms oben im Innenraum der Sättigungsdruck von 9,8 MPa erreicht, der zu einem Aufsieden des Wassers führt.

Abb. 51 zeigt die Geschwindigkeit im Stutzen.

In den letzten Maschen liegt sie über der Schallgeschwindigkeit.

Abb. 52: Massendampfgehalt im Stutzen.

Hier ist das Einlaufen der Zweiphasenfront in den Stutzen zu beobachten.

Zwischen 3 und 15 ms wird die Front teilweise durch die Zunahme der Geschwindigkeit wieder zurückgedrängt.

Die Abbildungen 53 und 54 zeigen die Temperatur im Stutzen und Innenraum. In Abb. 55 bis 64 ist das Druckfeld als Relief dargestellt. Insbesondere die Druckentwicklung im Stutzen wird hier deutlich. Die Abb. 65 bis 68 zeigen Vergleiche zwischen einer isothermen Rechnung mit linearer Wassergleichung und Sättigungsdruck an der Bruchöffnung (durchgezogene Linien) und der Rechnung mit genauer Wassergleichung für Versuch 3. Der Druck im Stutzen (Abb. 65) liegt im Falle der linearen Wassergleichung höher. Der Druck im Ringraum (Abb. 66) zeigt daher für die lineare Wassergleichung kleinere Schwankungen. Ab 50 ms nimmt er weiter ab, während bei Anwendung der genauen Wassergleichung der Einfluß des Aufsiedens den Druck festhält. In Abb. 67 ist der Druck im Innenraum und unterem Plenum dargestellt, hier wird die fehlende Modellierung der Siedevorgänge durch die lineare Wassergleichung deutlich. Abb. 68 zeigt den Geschwindigkeitsverlauf im Stutzen. Lediglich in den Maschen nahe der Bruchöffnung ist der Unterschied groß. Die ab 60 ms deutlich höhere Geschwindigkeit bei genauer Wassergleichung ist auf den höheren Ringraumdruck zurückzuführen.

Aus Abb. 65 und 66 wird deutlich, daß die Übereinstimmung in der Anfangsphase durch einen niedrigeren Druck an der Bruchöffnung für die lineare Wassergleichung verbessert werden könnte. Insbesondere sollte geprüft werden, ob nicht der zum Anfangszustand gehörige isentropen Sättigungszustand als Randbedingung geeigneter ist als der hier verwendete Sättigungszustand bei der Anfangstemperatur.

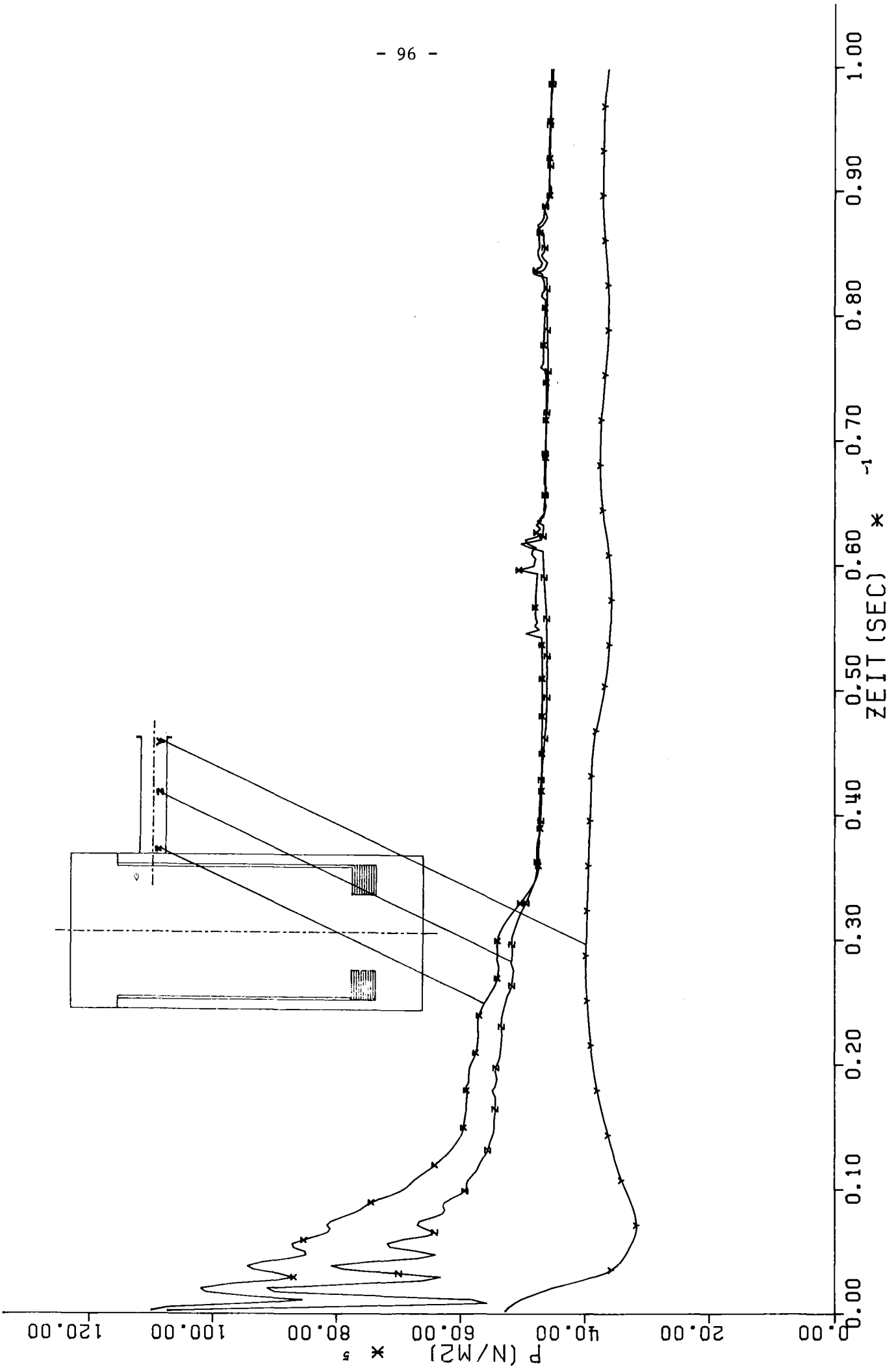
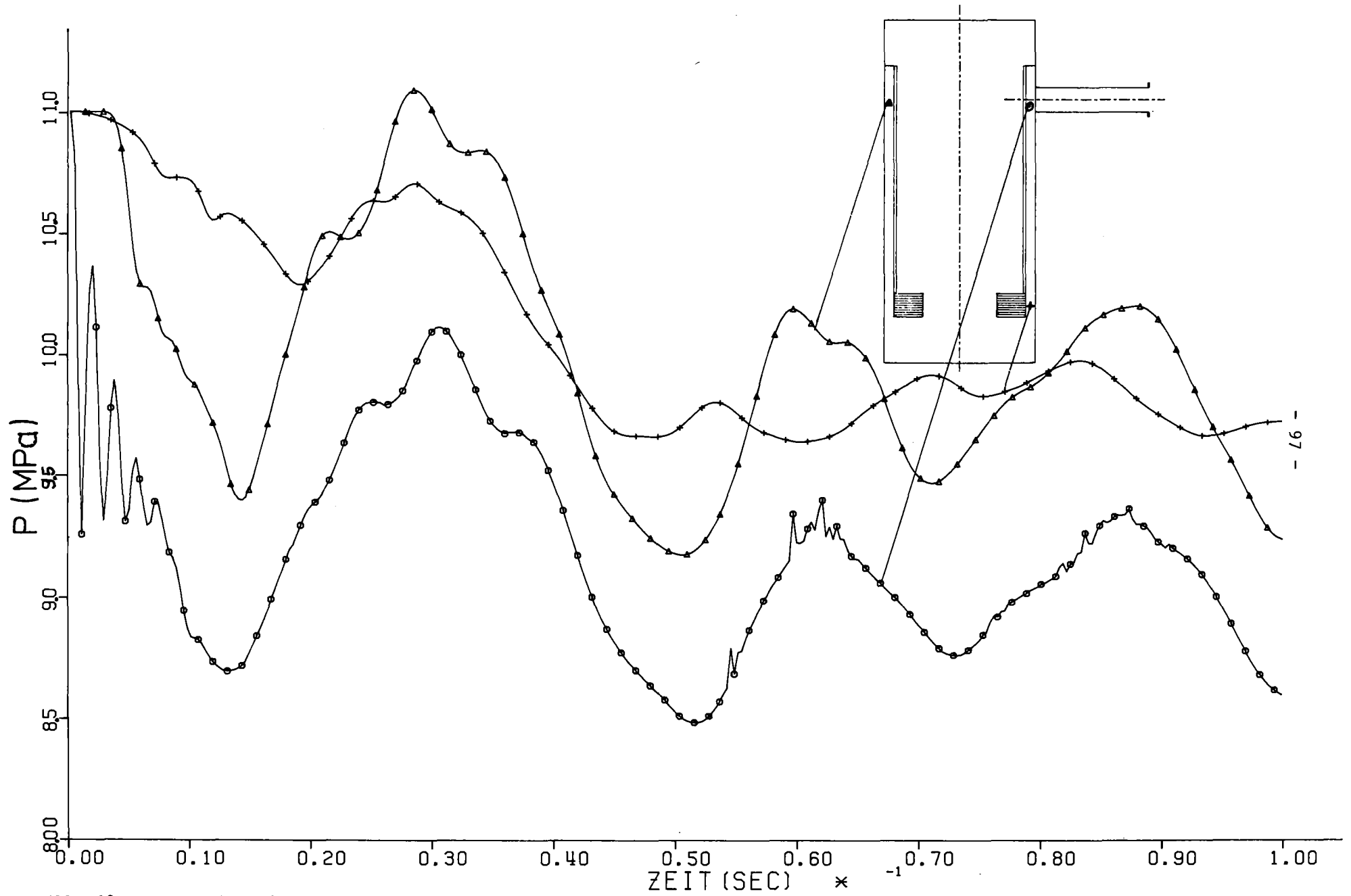


Abb. 31: Druckverlauf im Stutzen, Versuch 3





- 97 -

Abb. 32: Druckverlauf im Ringraum, Versuch 3

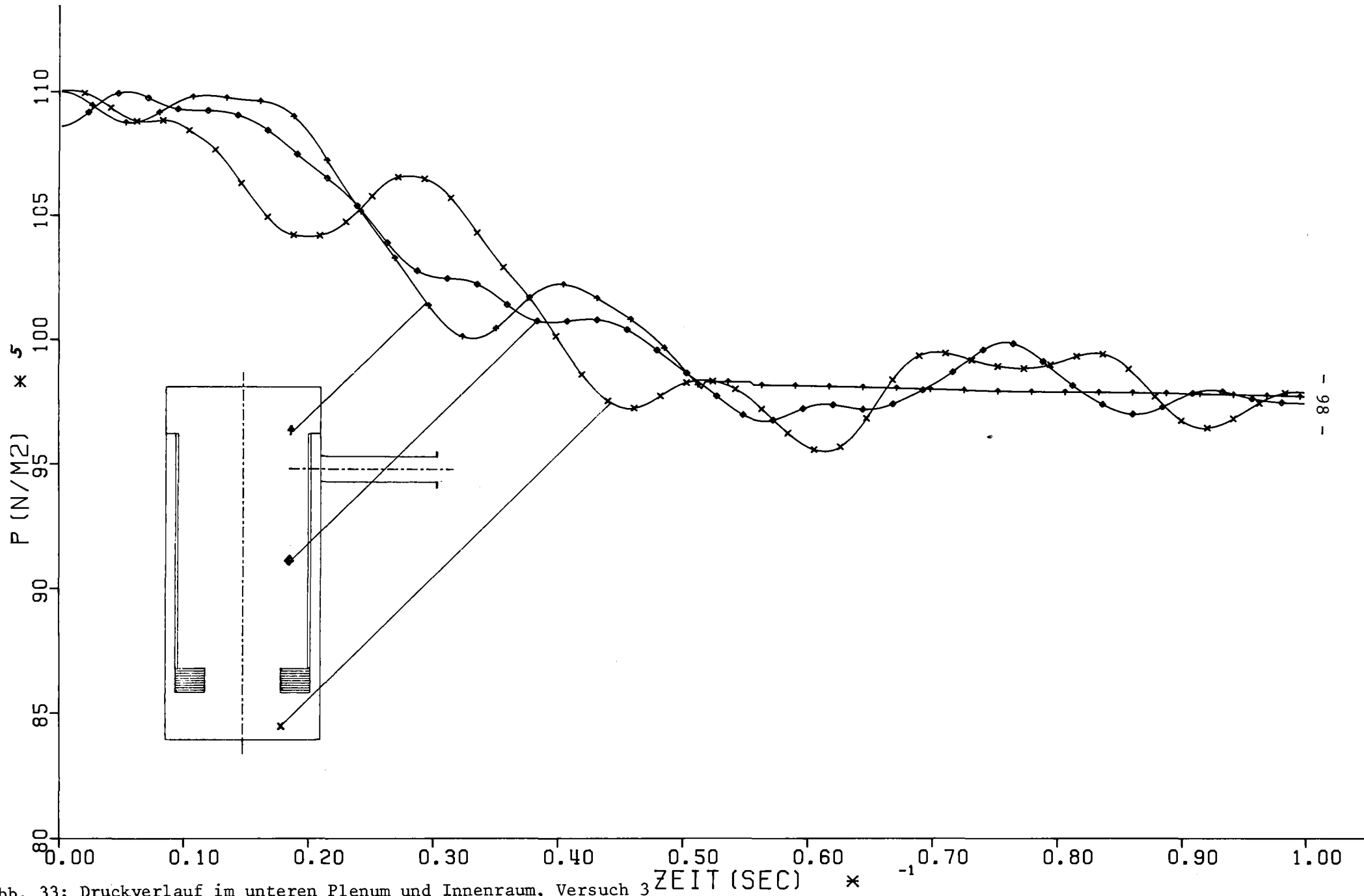


Abb. 33: Druckverlauf im unteren Plenum und Innenraum, Versuch 3

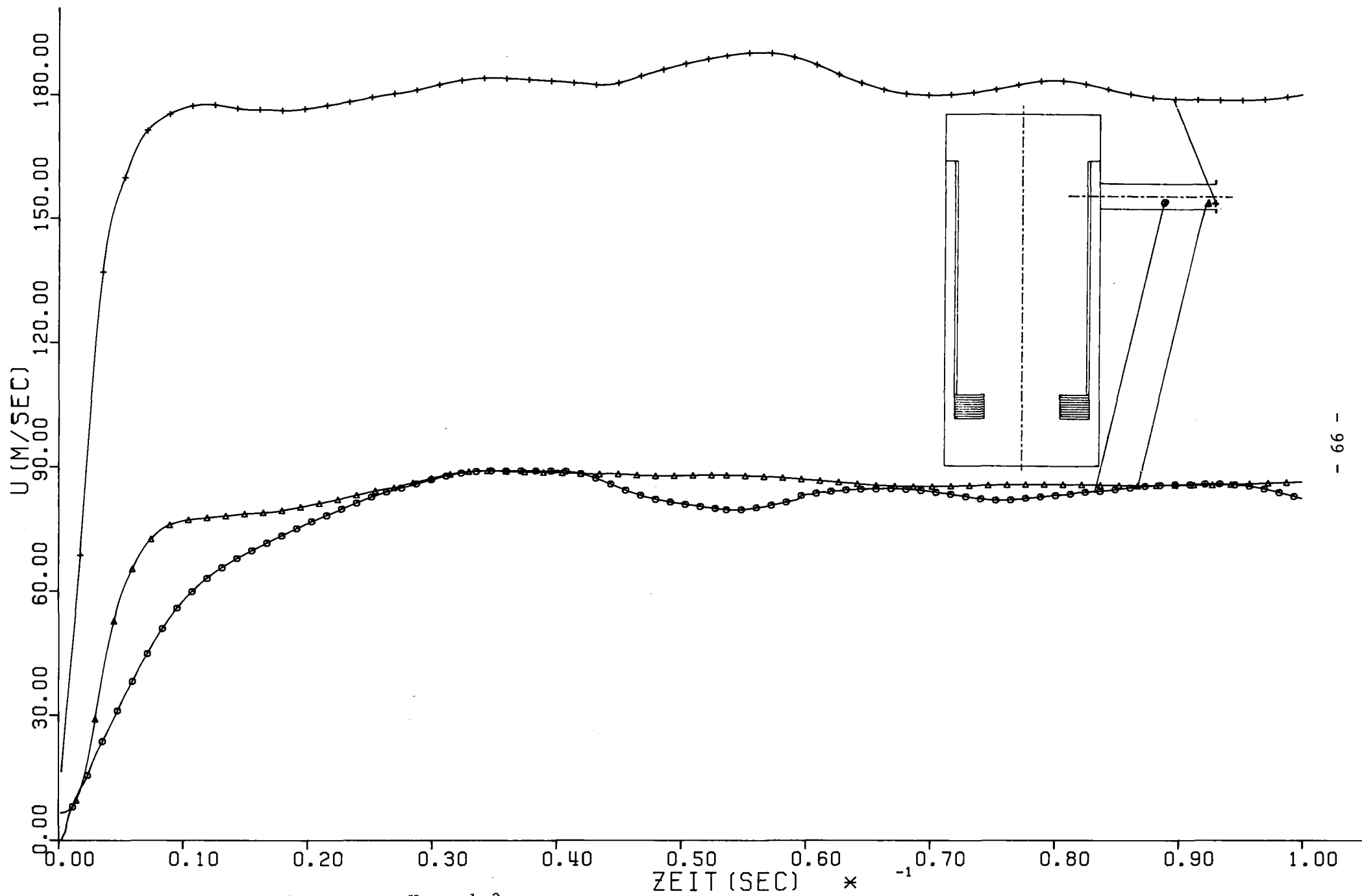


Abb. 34: Geschwindigkeit im Stutzen, Versuch 3

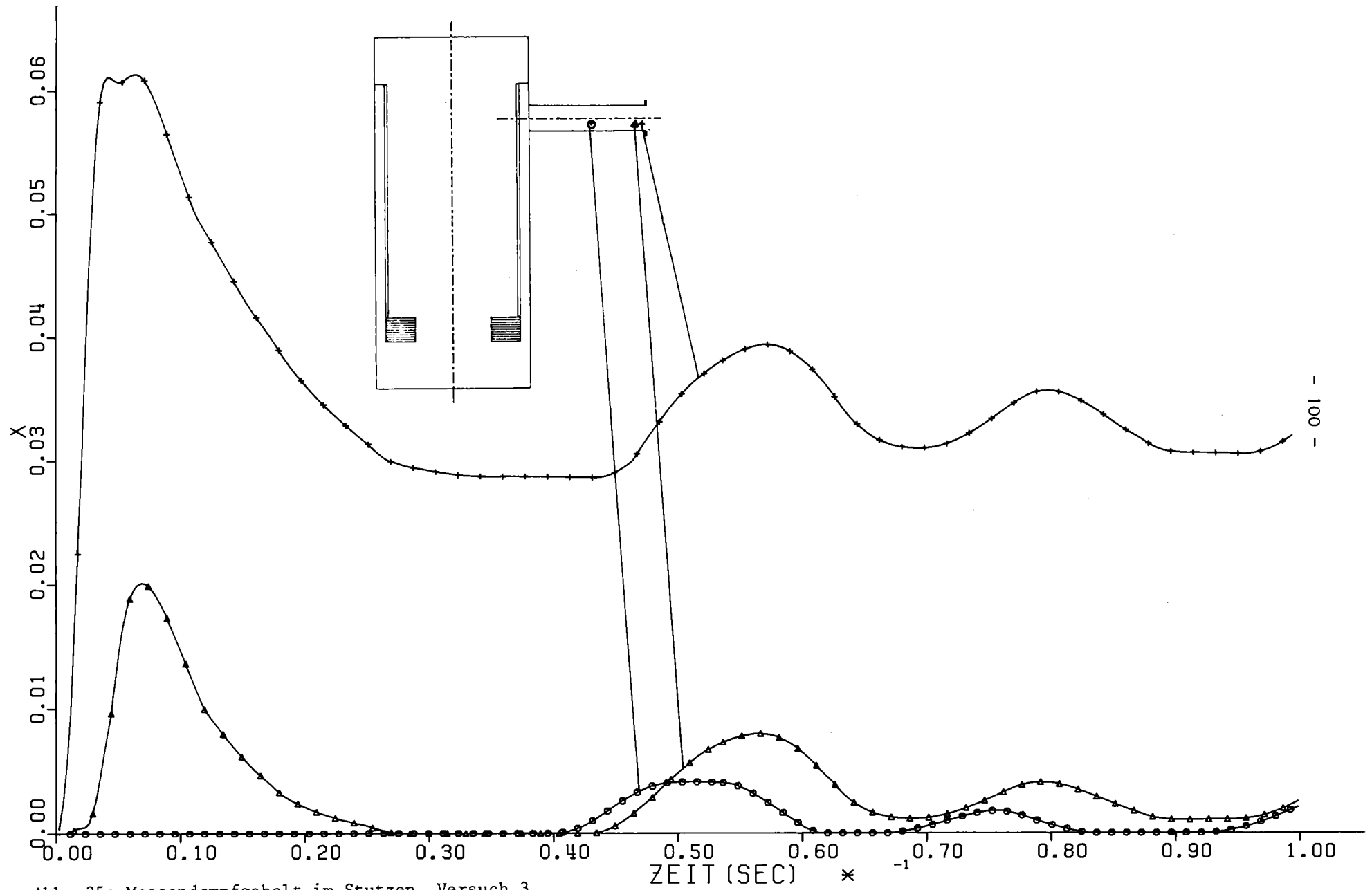


Abb. 35: Massendampfgehalt im Stutzen, Versuch 3

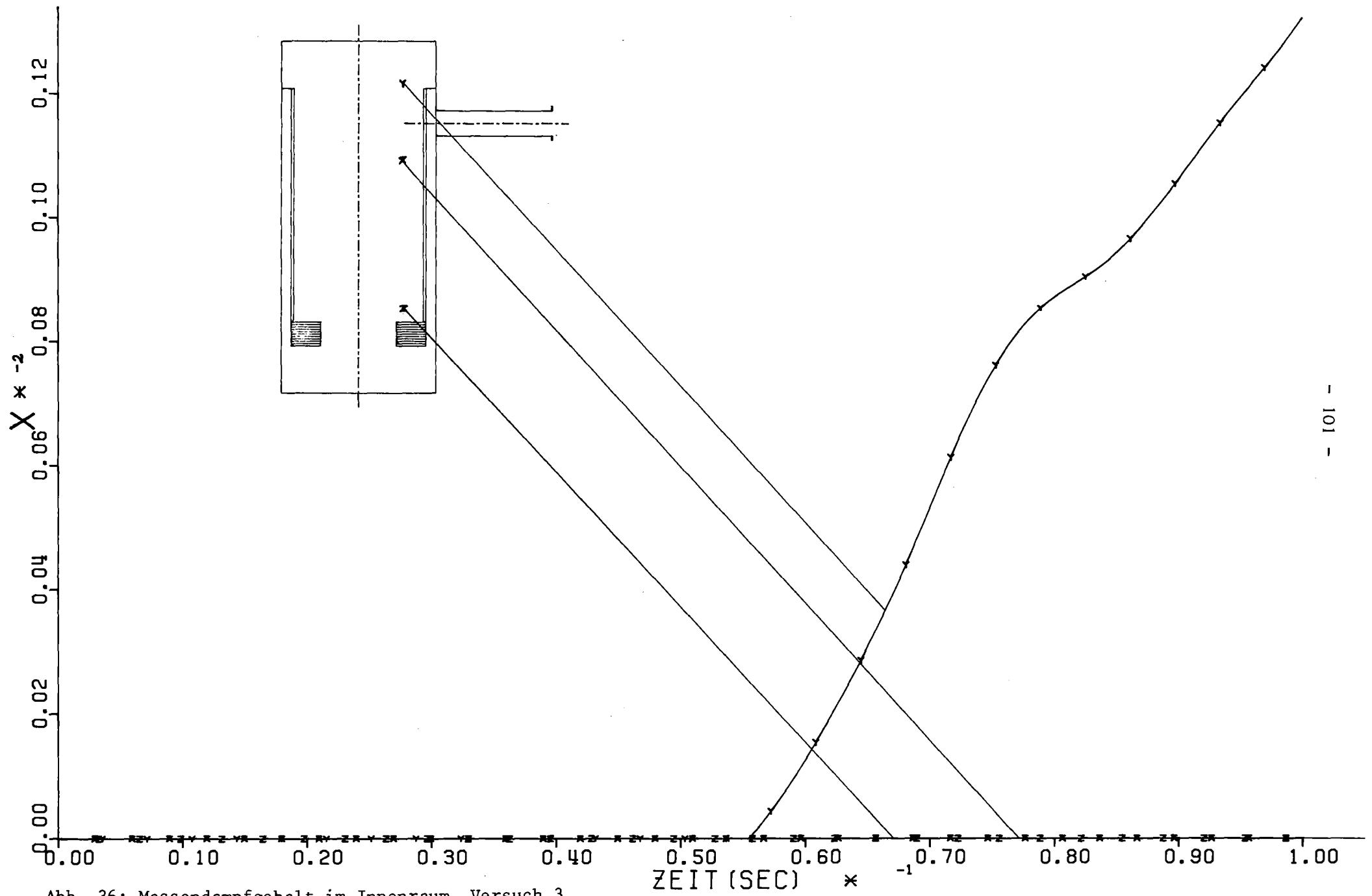


Abb. 36: Massendampfgehalt im Innenraum, Versuch 3

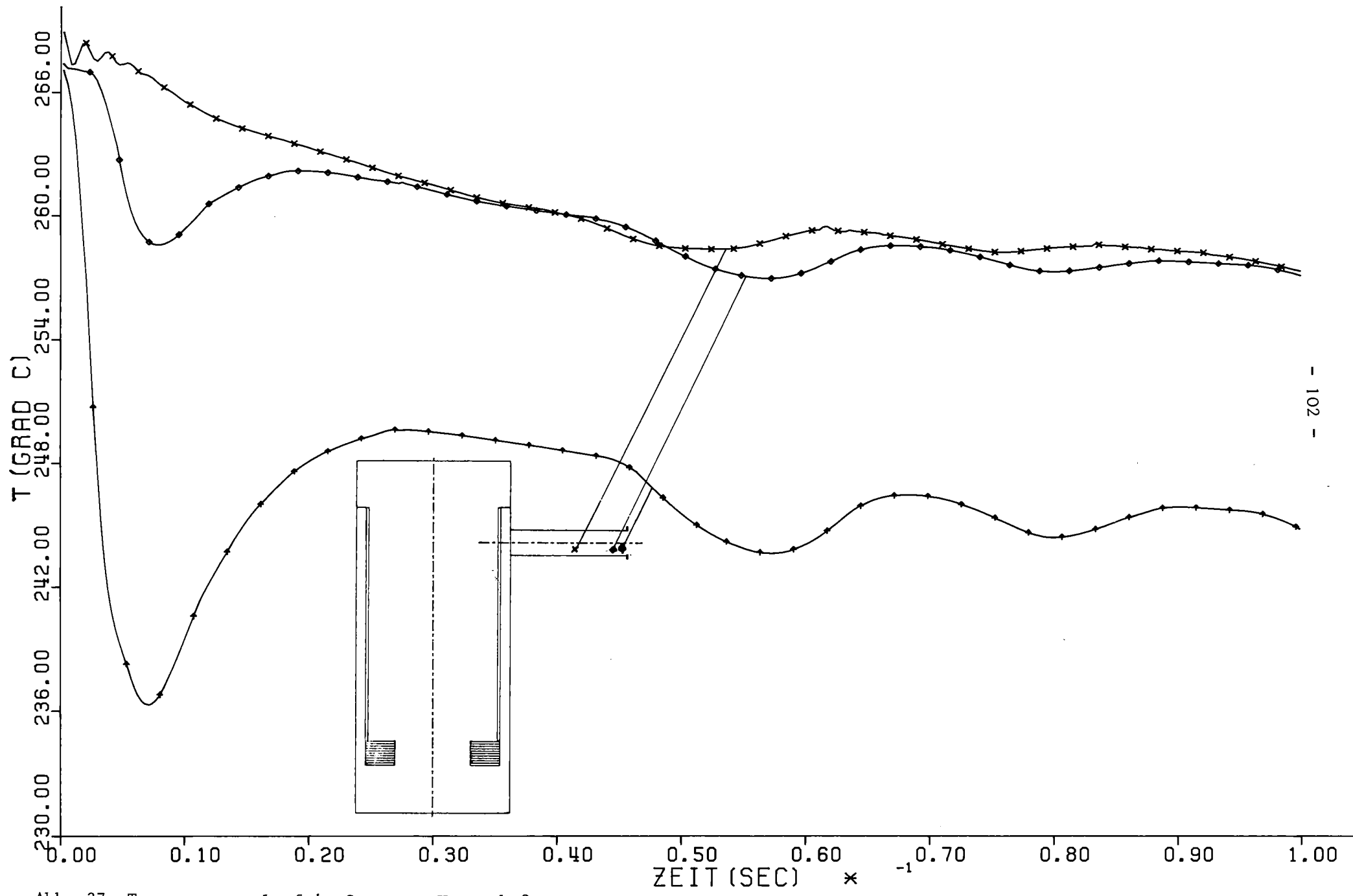


Abb. 37: Temperaturverlauf im Stutzen, Versuch 3

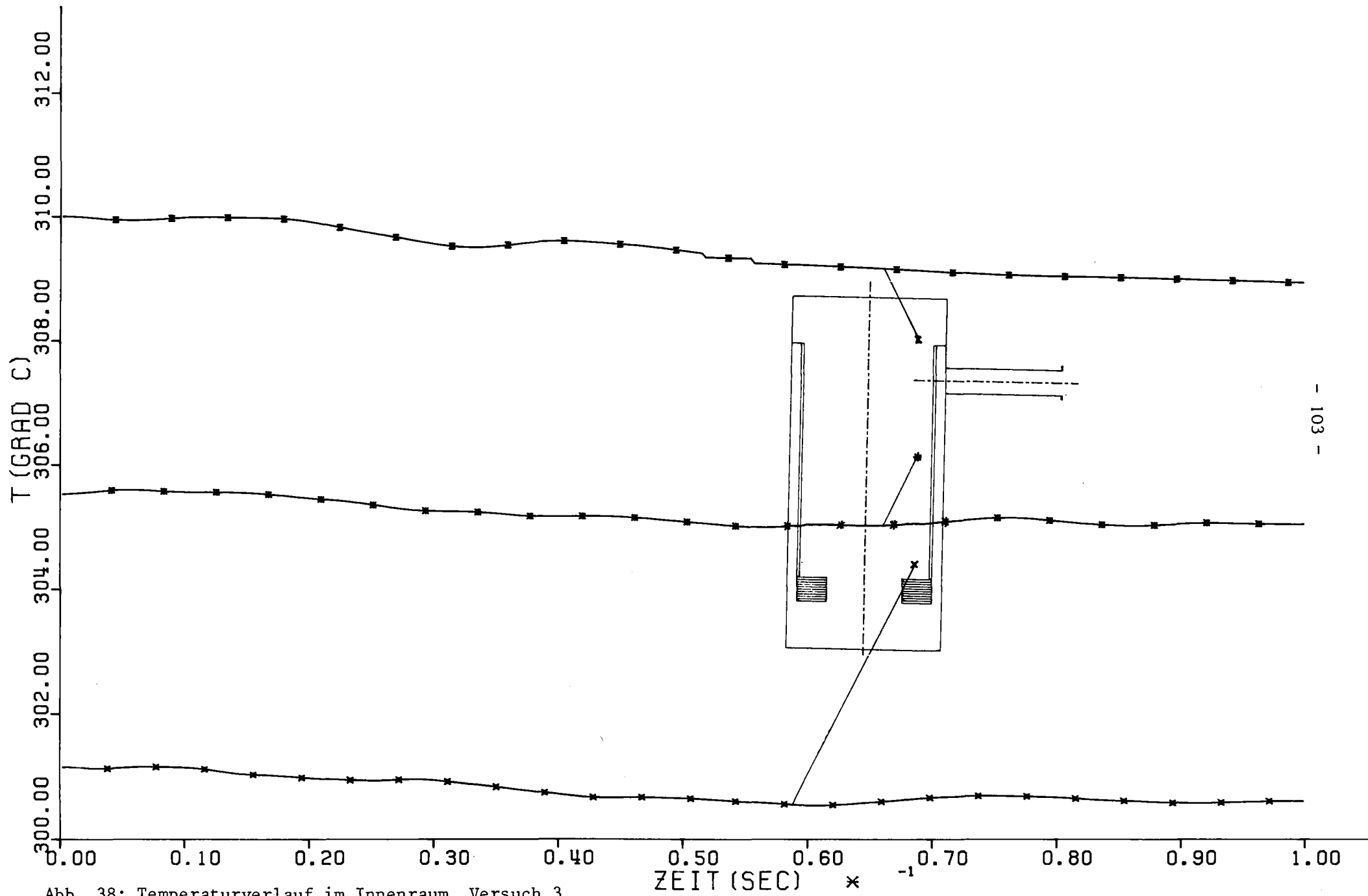


Abb. 38: Temperaturverlauf im Innenraum, Versuch 3

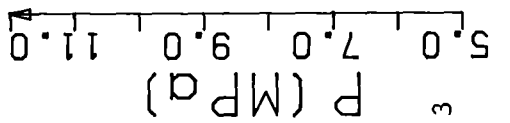
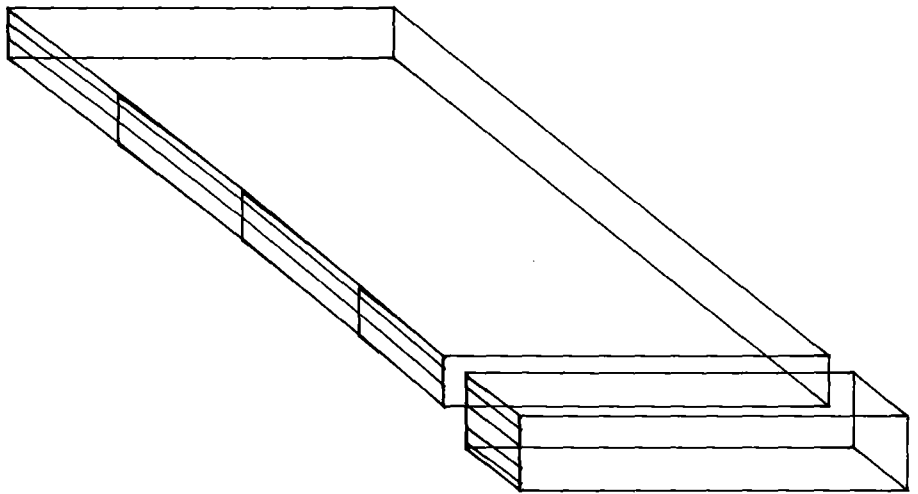
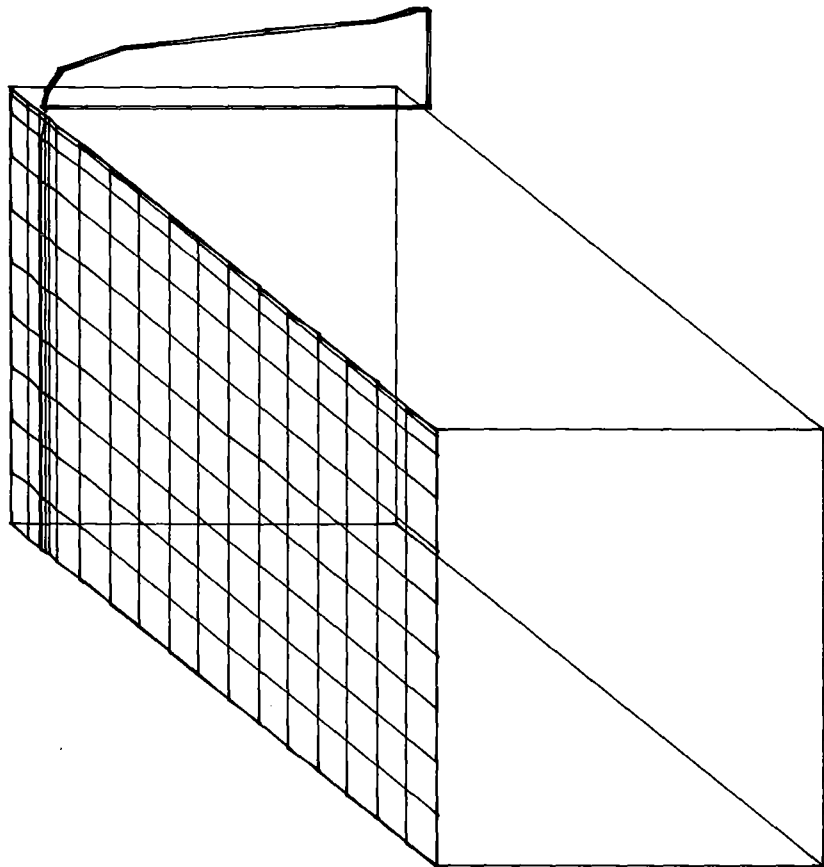
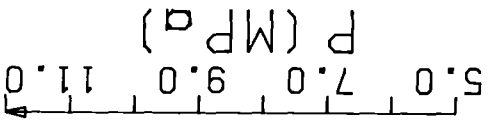
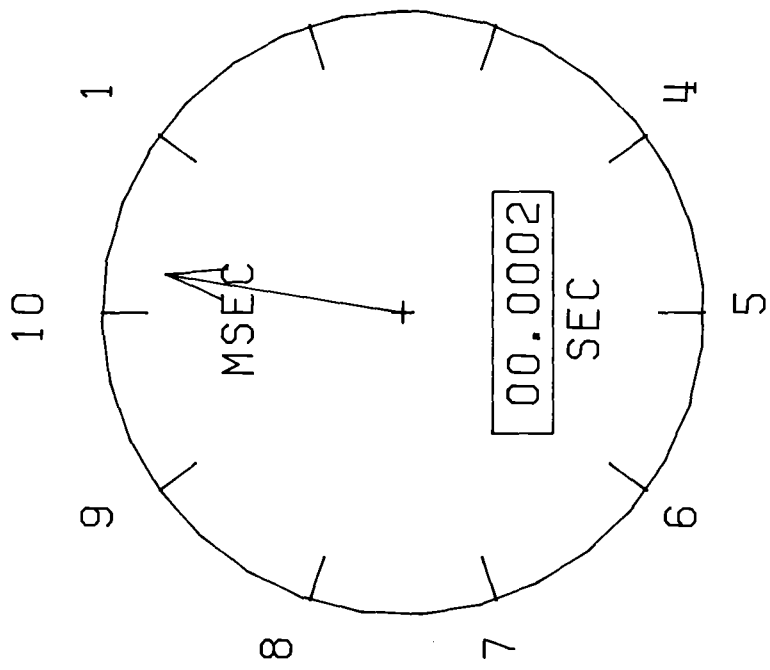
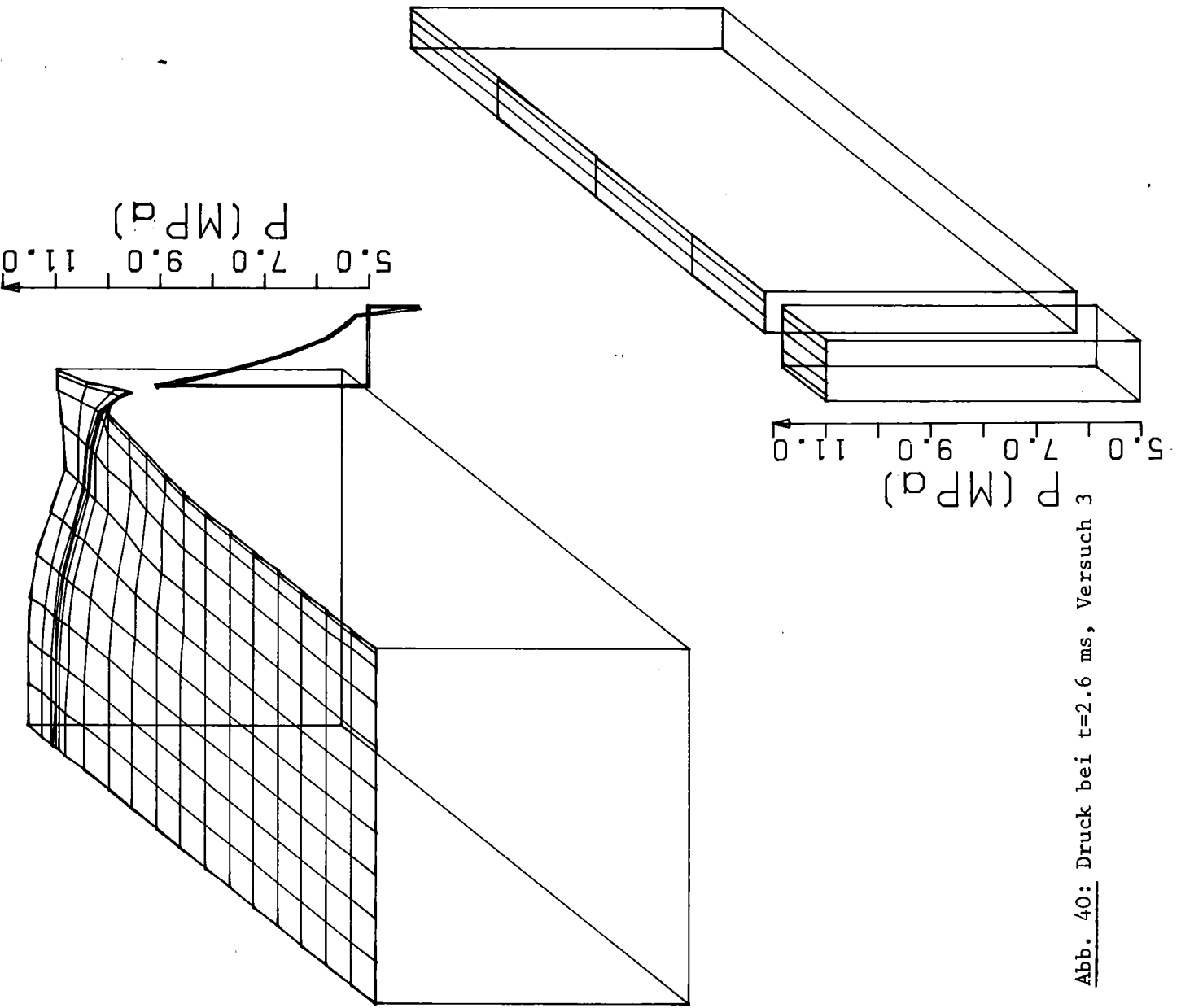


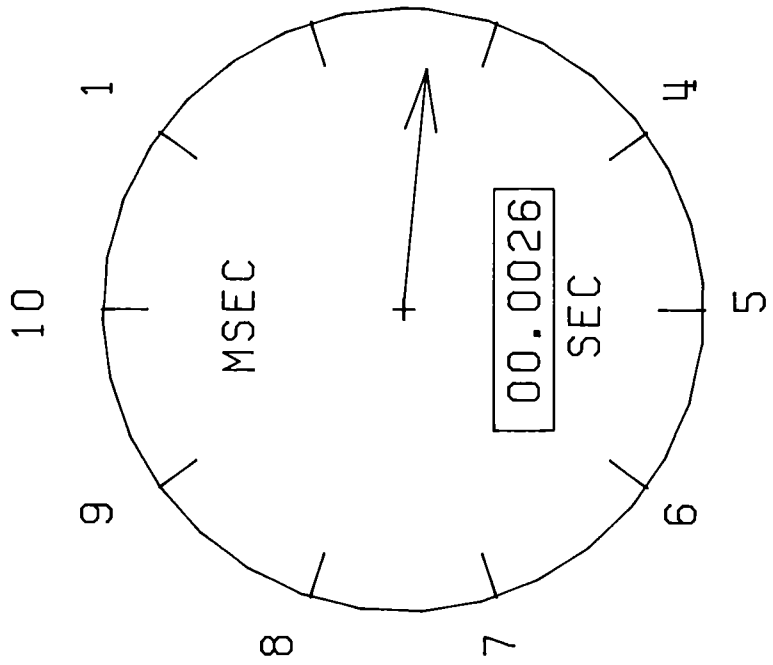
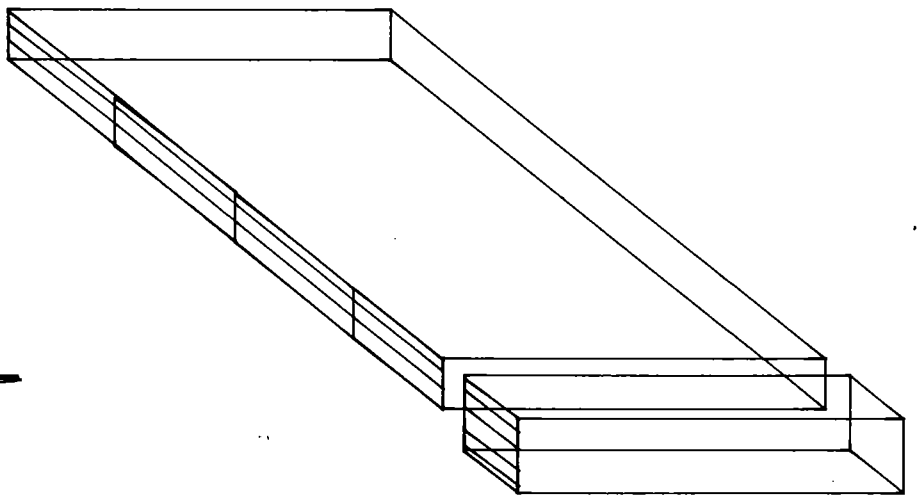
Abb. 39: Druck bei  $t=0.2$  ms, Versuch 3





5.0 7.0 9.0 11.0  
 $P$  (MPa)

5.0 7.0 9.0 11.0  
 $P$  (MPa)



- 105 - 2

3

Abb. 40: Druck bei  $t=2.6$  ms, Versuch 3

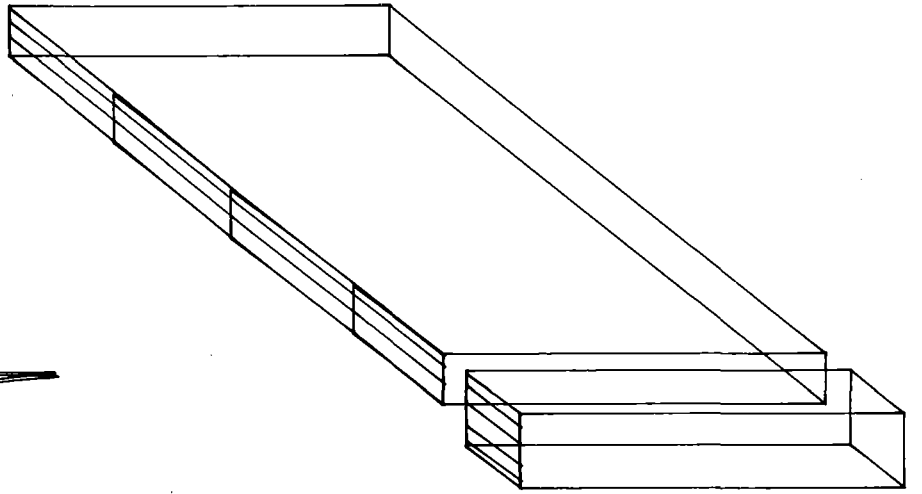
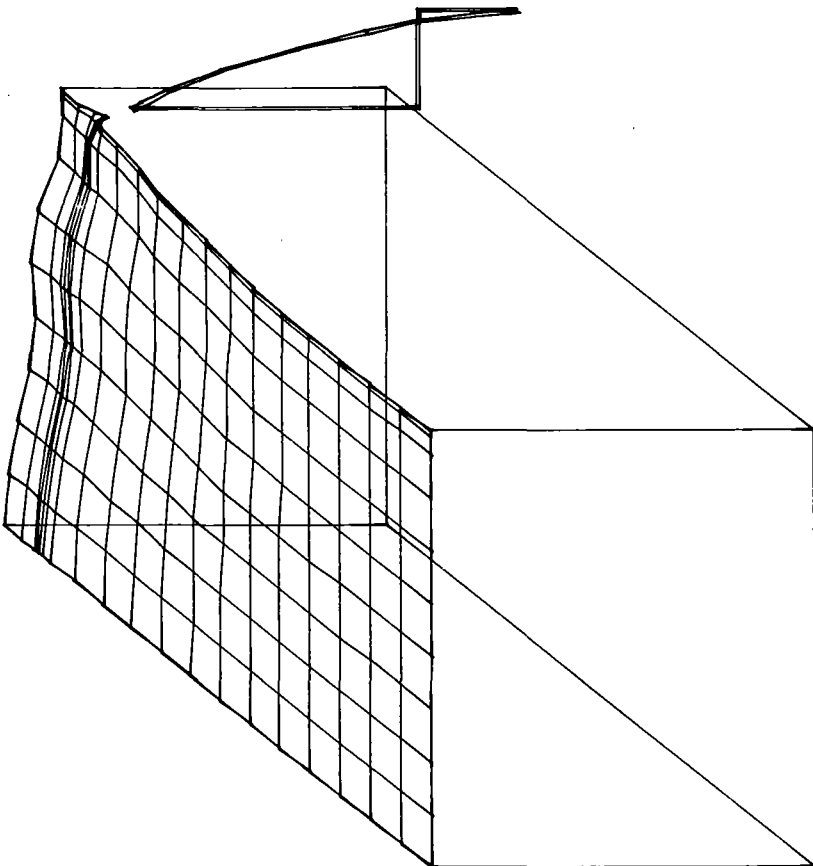
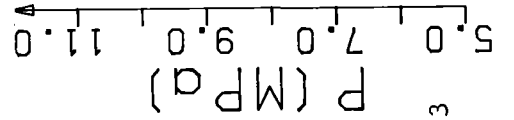
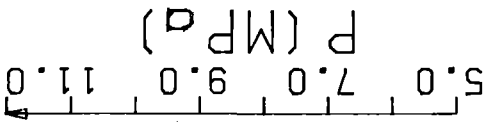
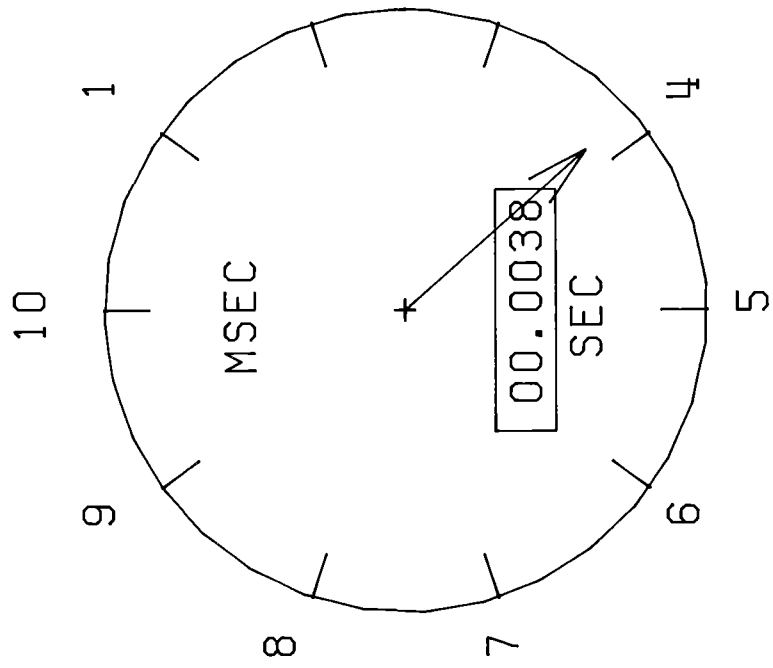
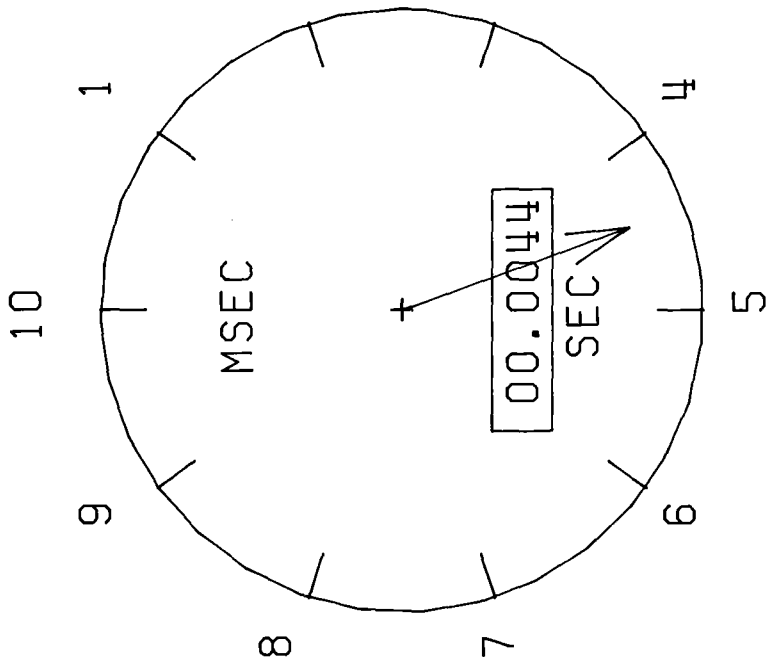
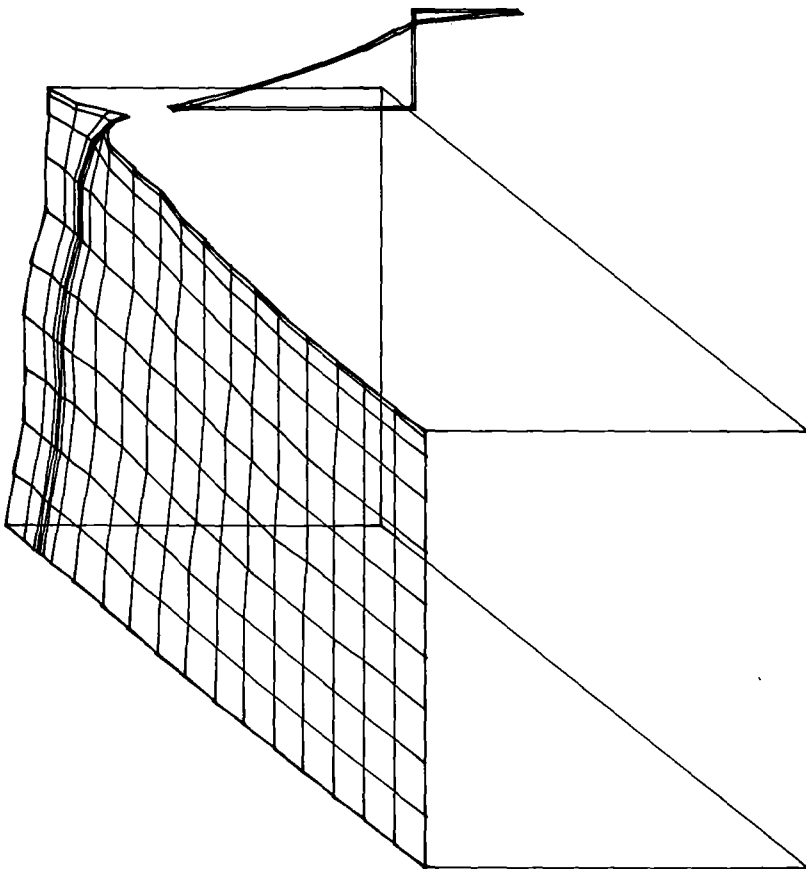


Abb. 41: Druck bei  $t=3.8$  ms, Versuch 3



P (MPa)  
5.0 7.0 9.0 11.0



P (MPa)  
5.0 7.0 9.0 11.0

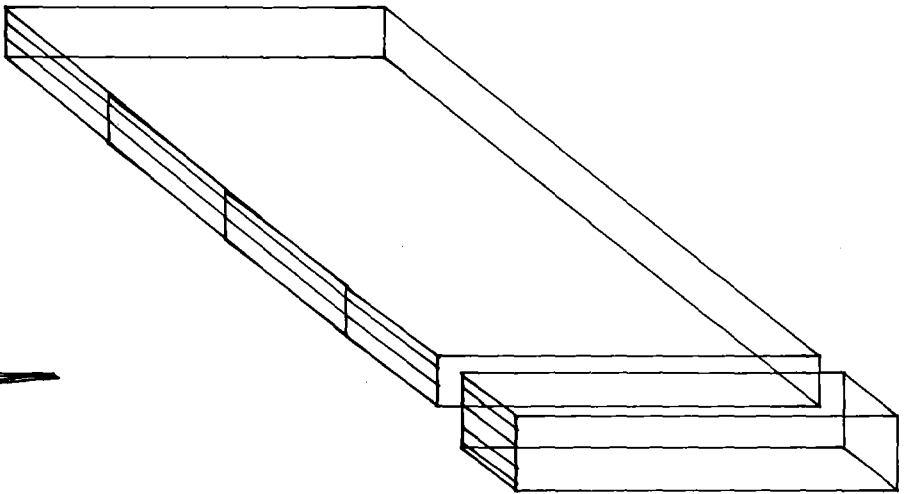


Abb. 42: Druck bei  $t=4.4\text{ms}$ , Versuch 3

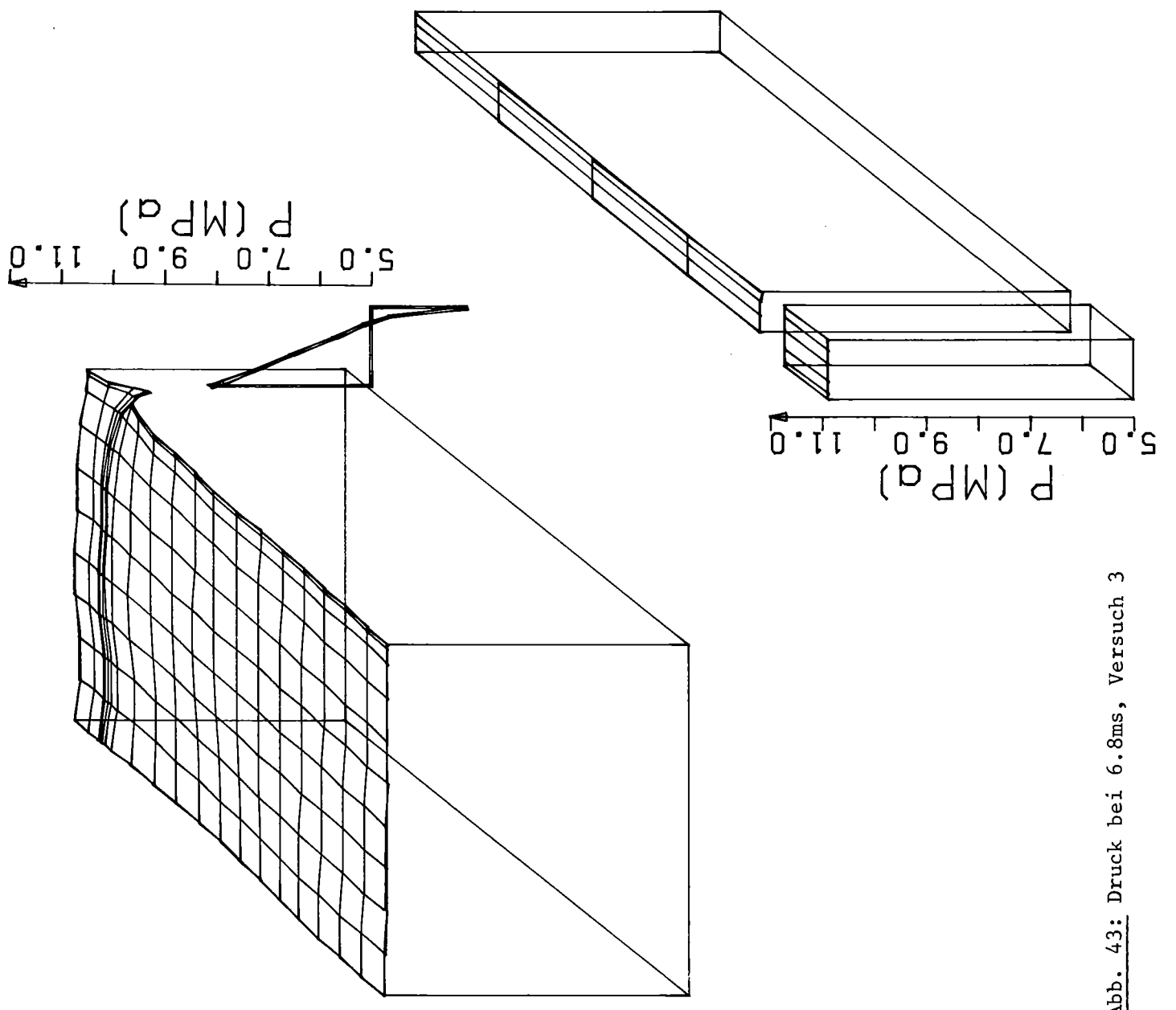
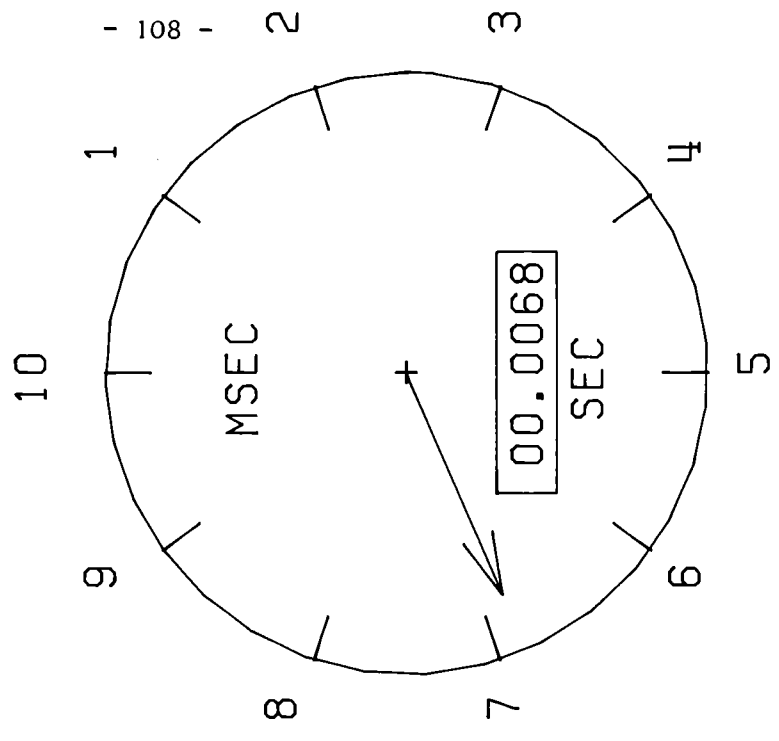


Abb. 43: Druck bei 6.8ms, Versuch 3

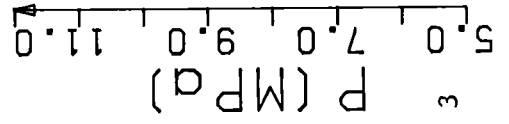
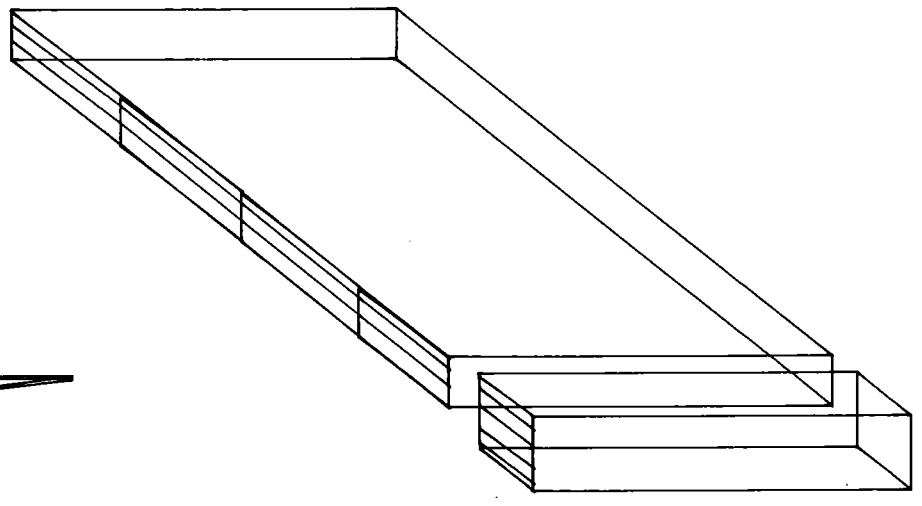
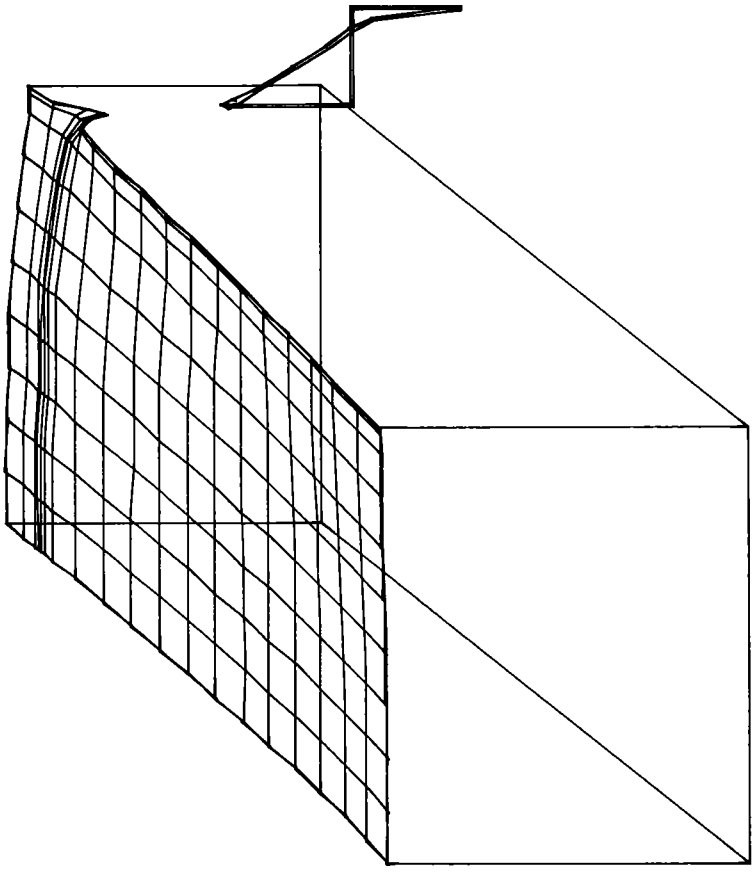
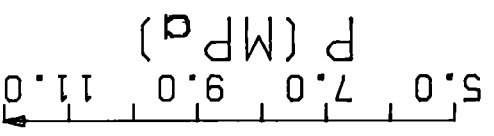
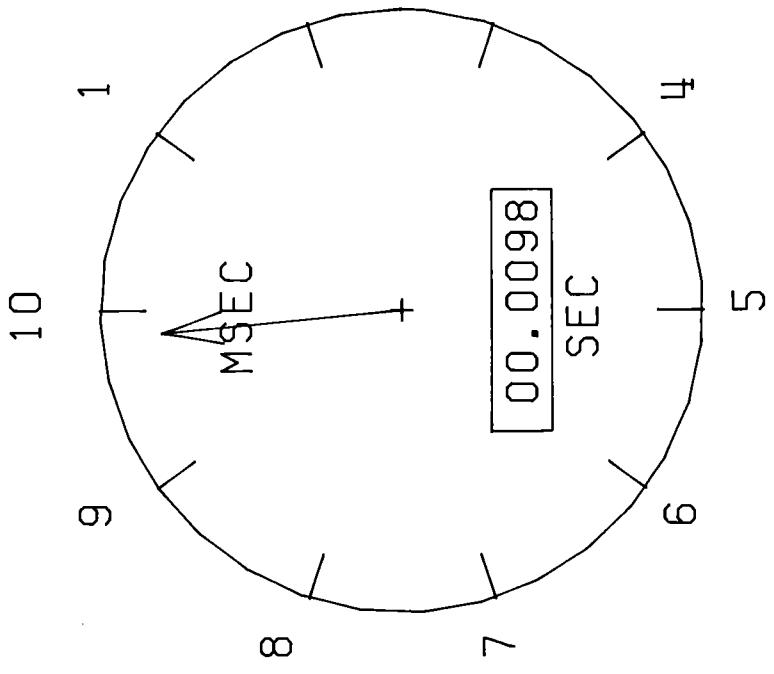
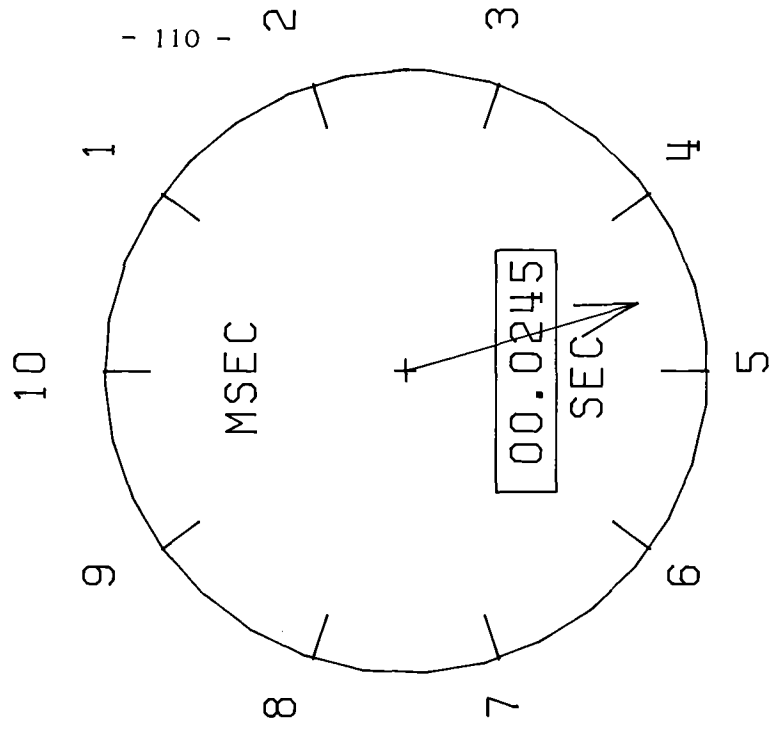
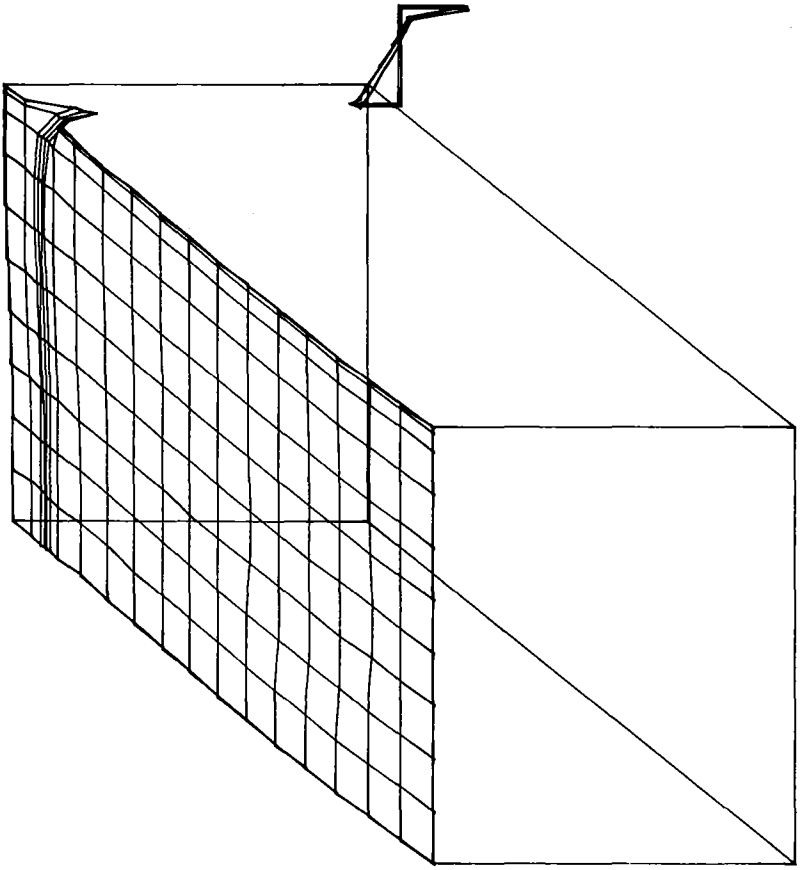


Abb. 44: Druck bei  $t=9.8$  ms, Versuch 3



P (MPa)  
5.0 7.0 9.0 11.0



P (MPa)  
5.0 7.0 9.0 11.0

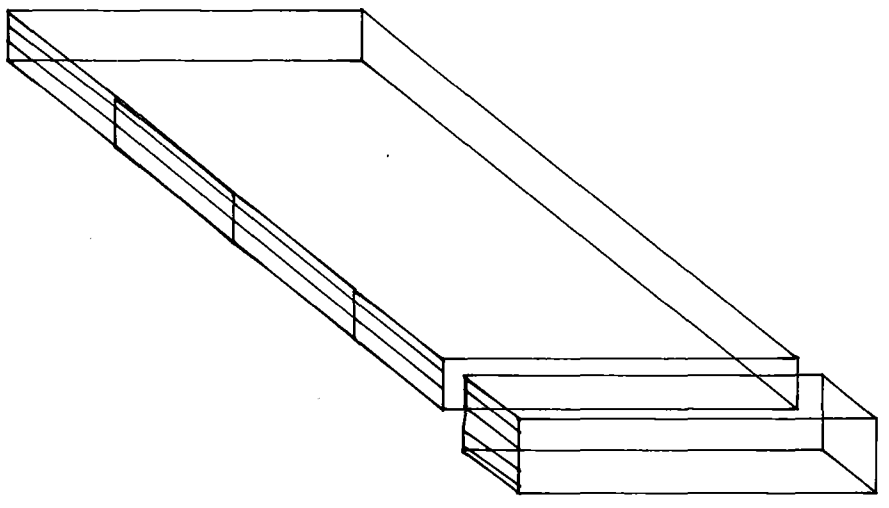


Abb. 45: Druck bei  $t=24.5$  ms, Versuch 3

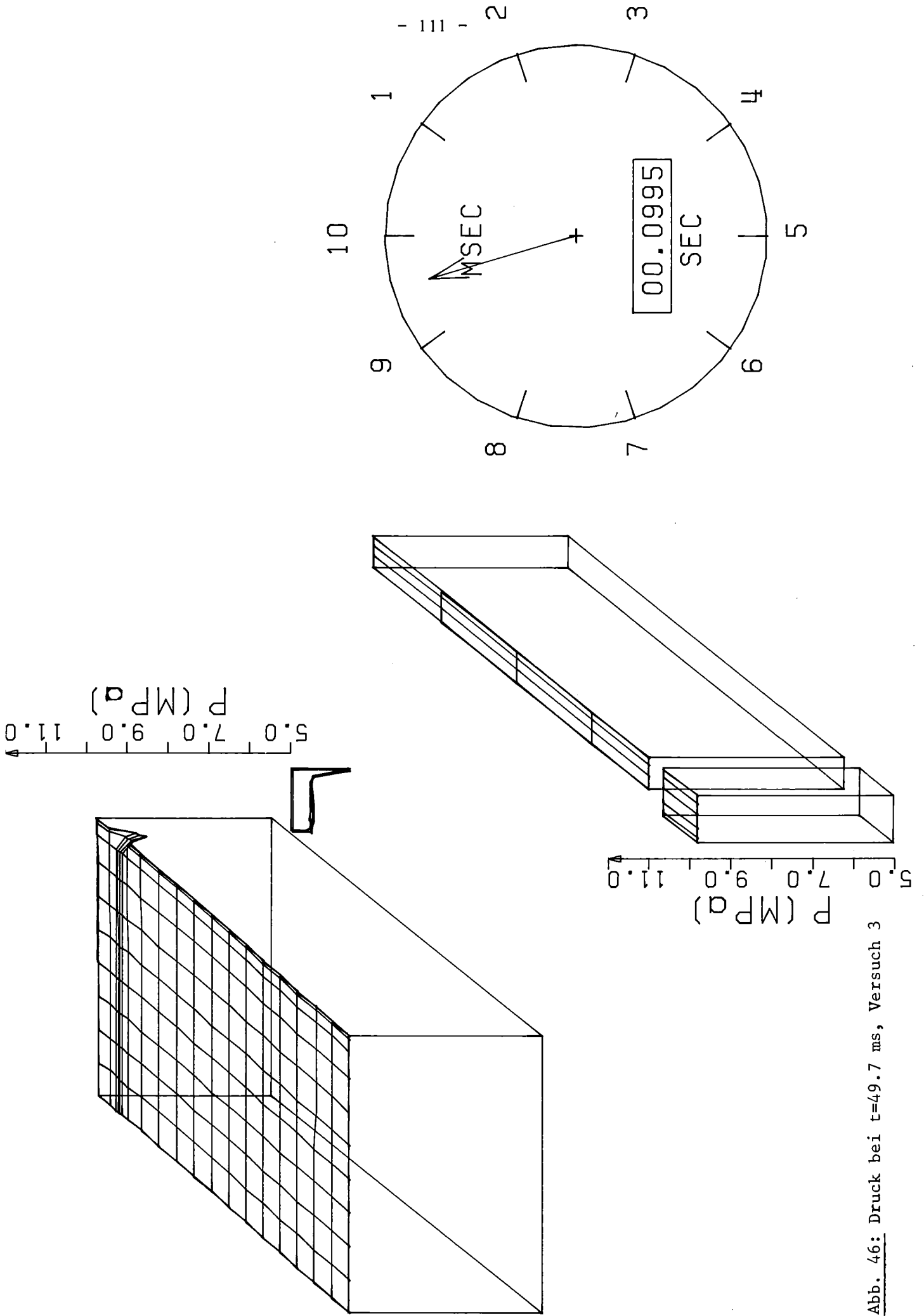


Abb. 46: Druck bei  $t=49.7$  ms, Versuch 3

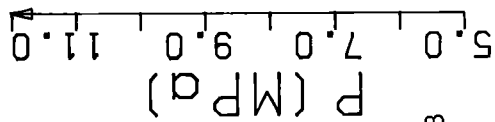
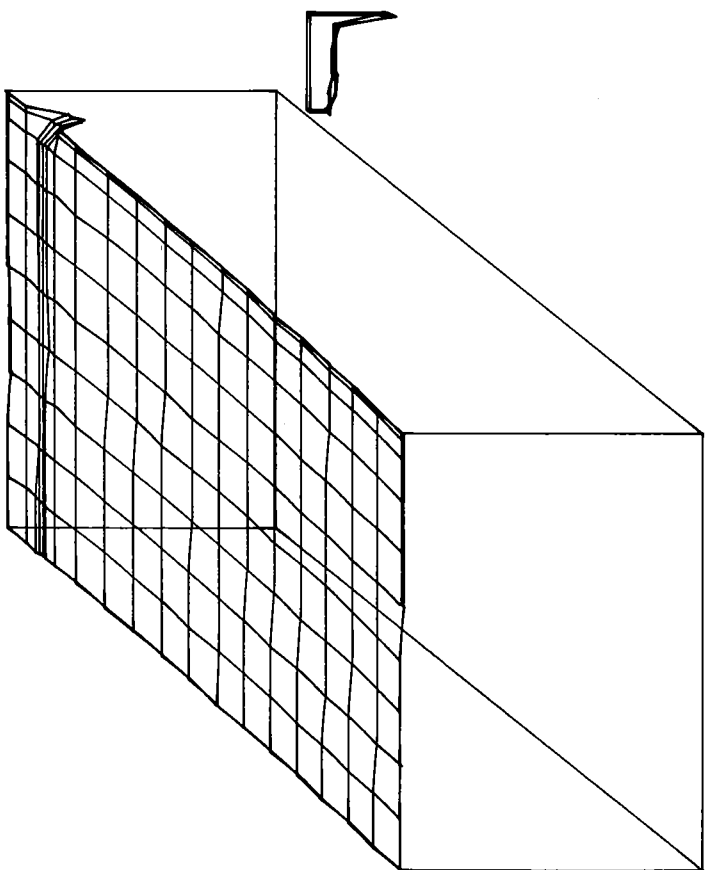
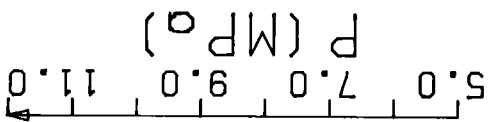
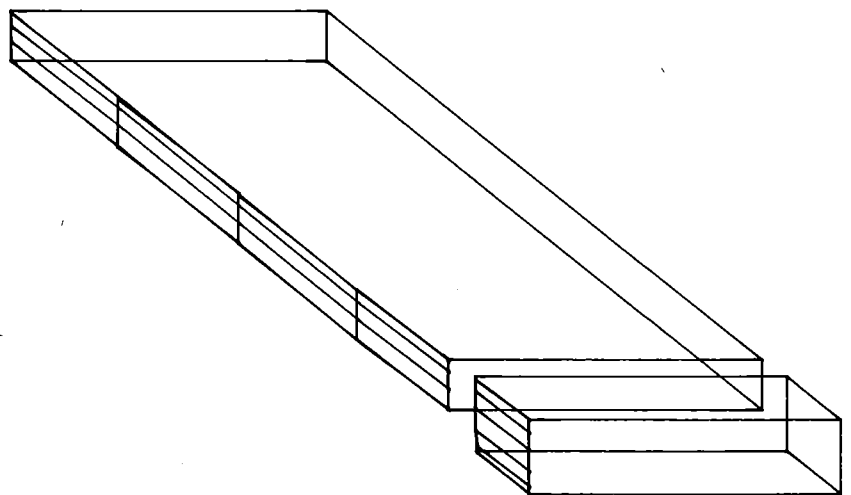
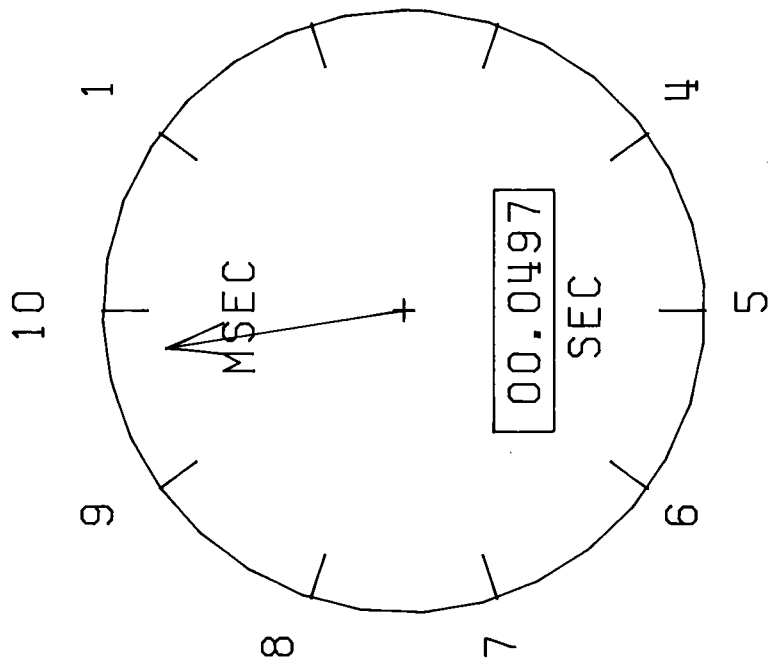


Abb. 47: Druck bei  $t=99.5 \text{ ms}$ , Versuch 3



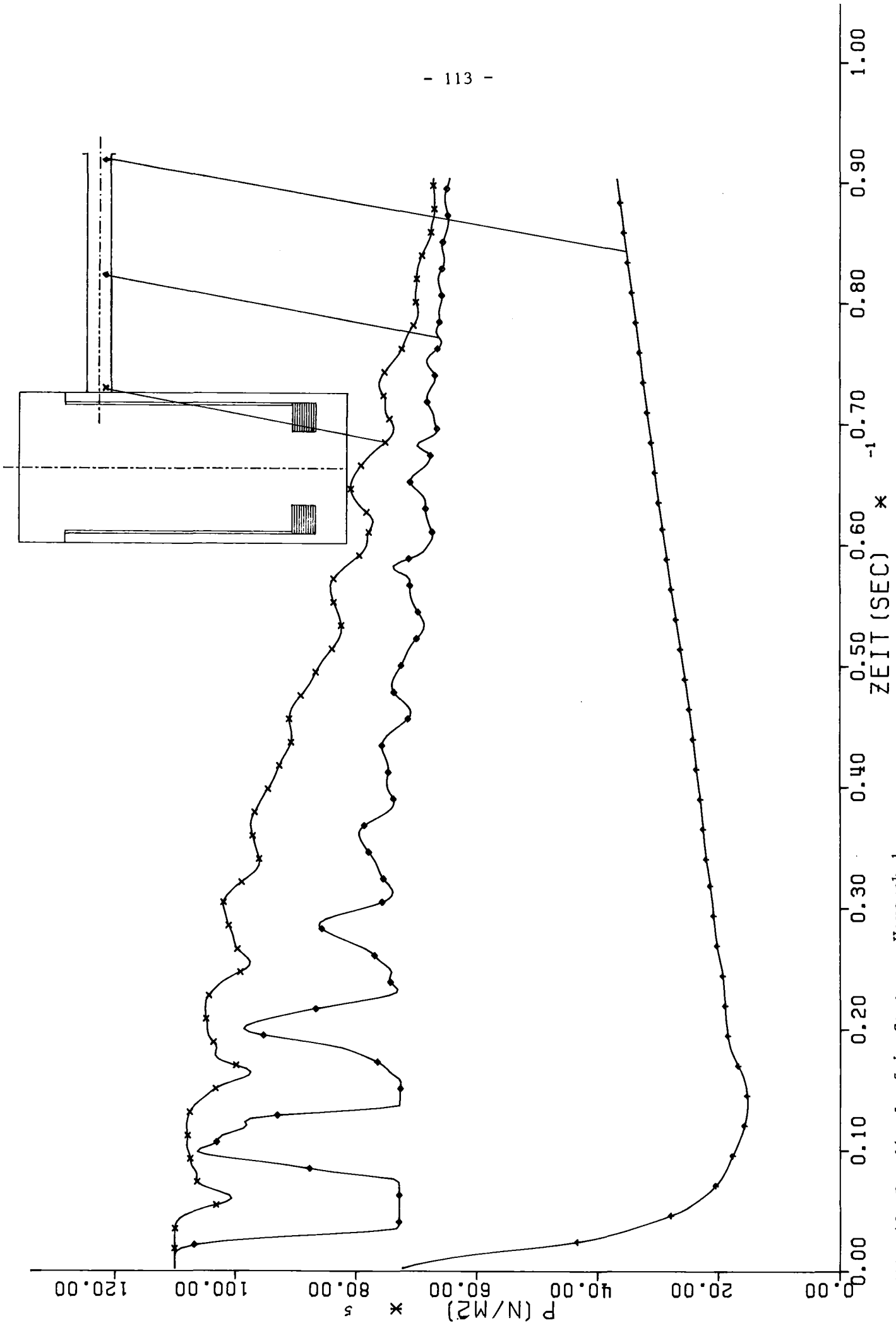


Abb. 48: Druckverlauf im Stutzen, Versuch 1

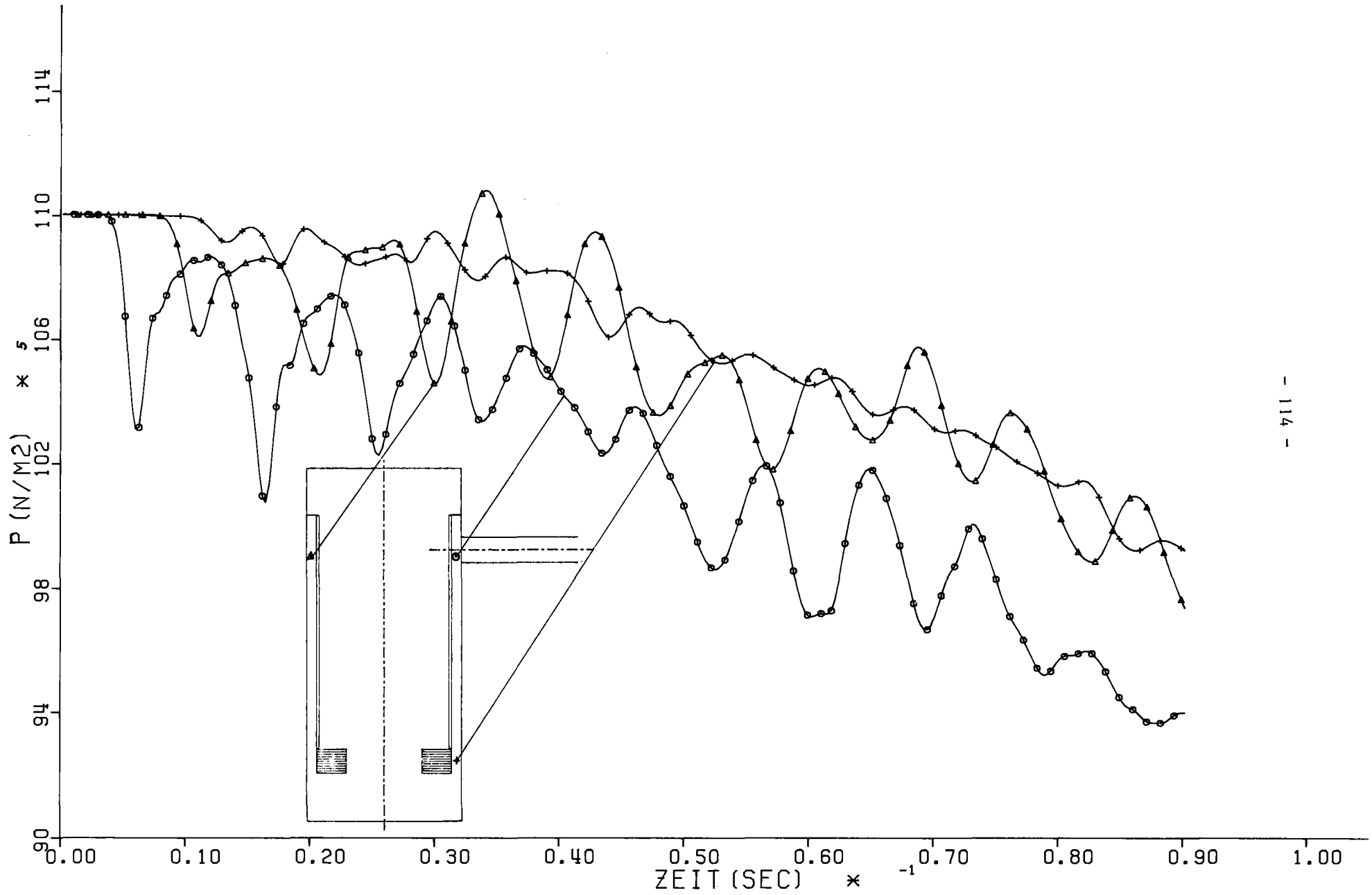


Abb. 49: Druckverlauf im Ringraum, Versuch 1

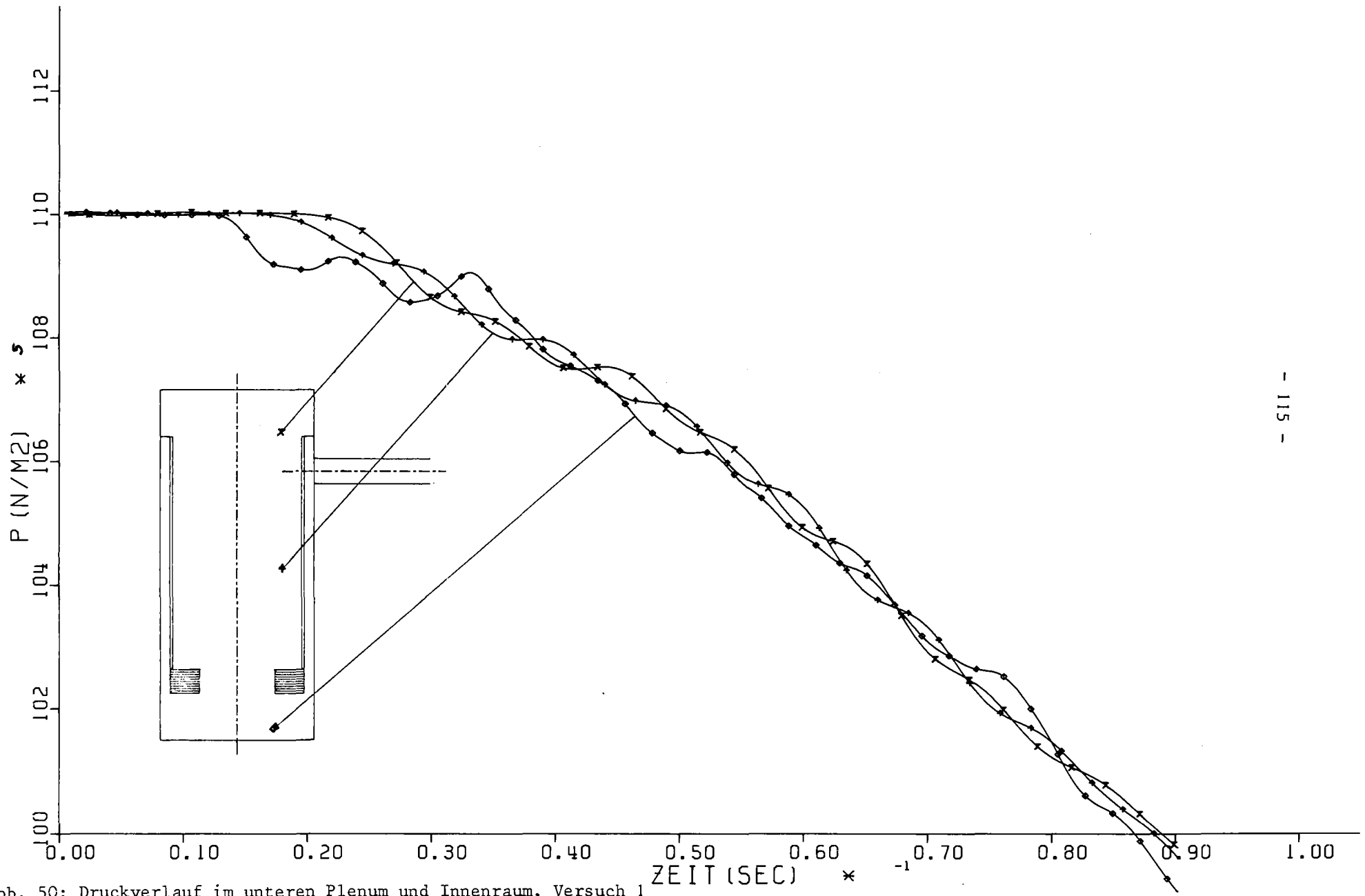


Abb. 50: Druckverlauf im unteren Plenum und Innenraum, Versuch 1

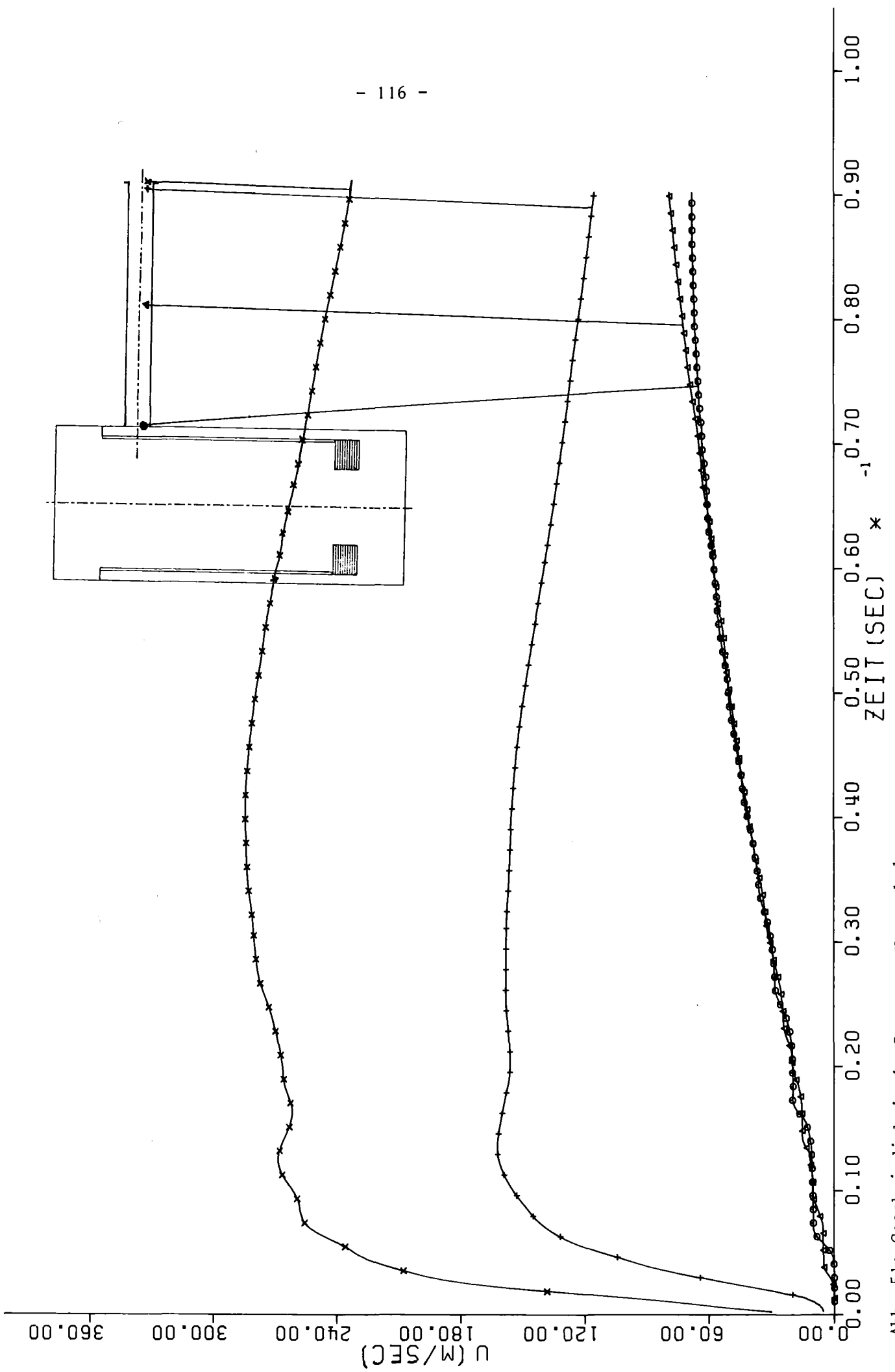


Abb. 51: Geschwindigkeit im Stutzen, Versuch 1

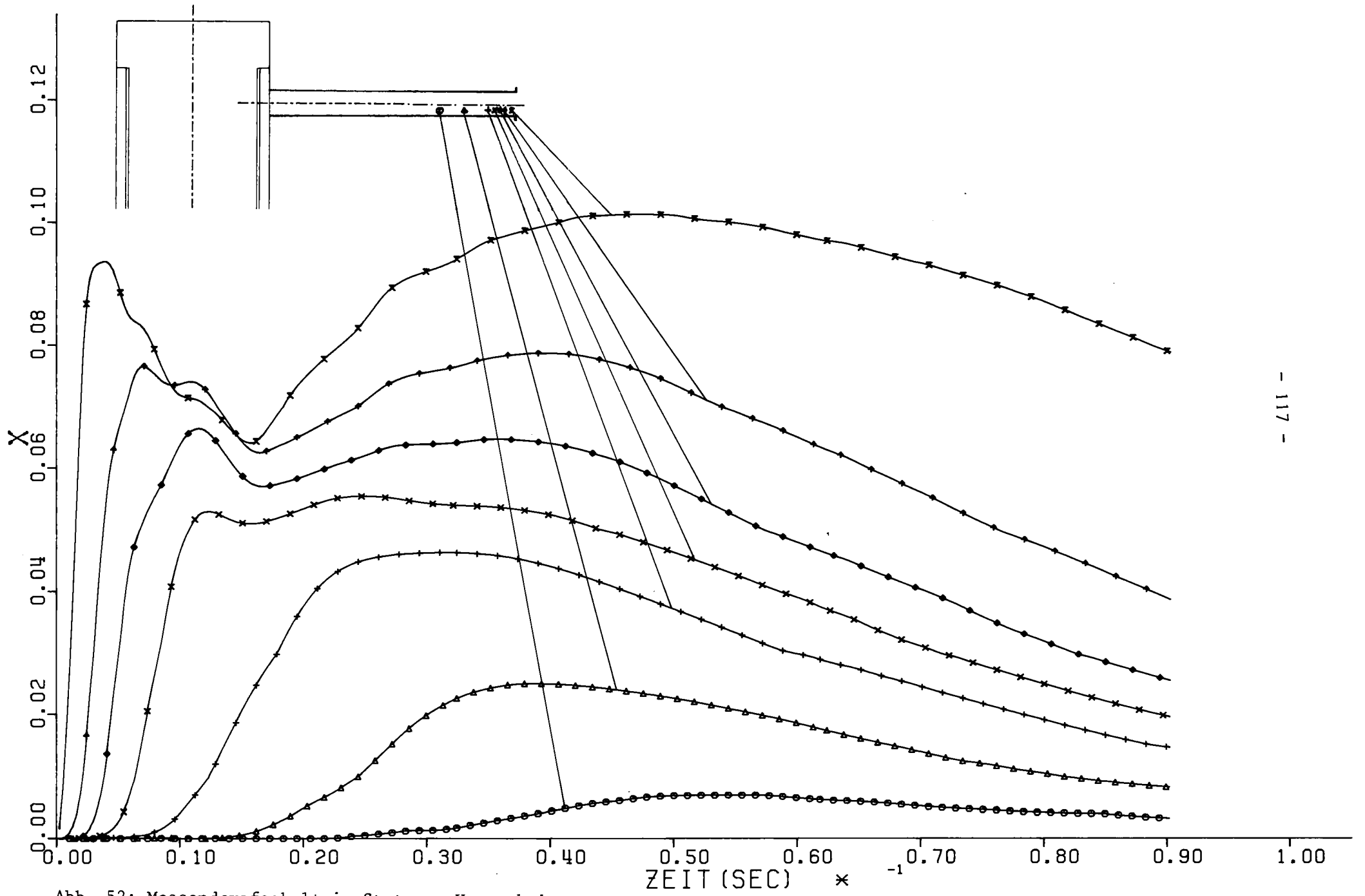


Abb. 52: Massendampfgehalt im Stutzen, Versuch 1

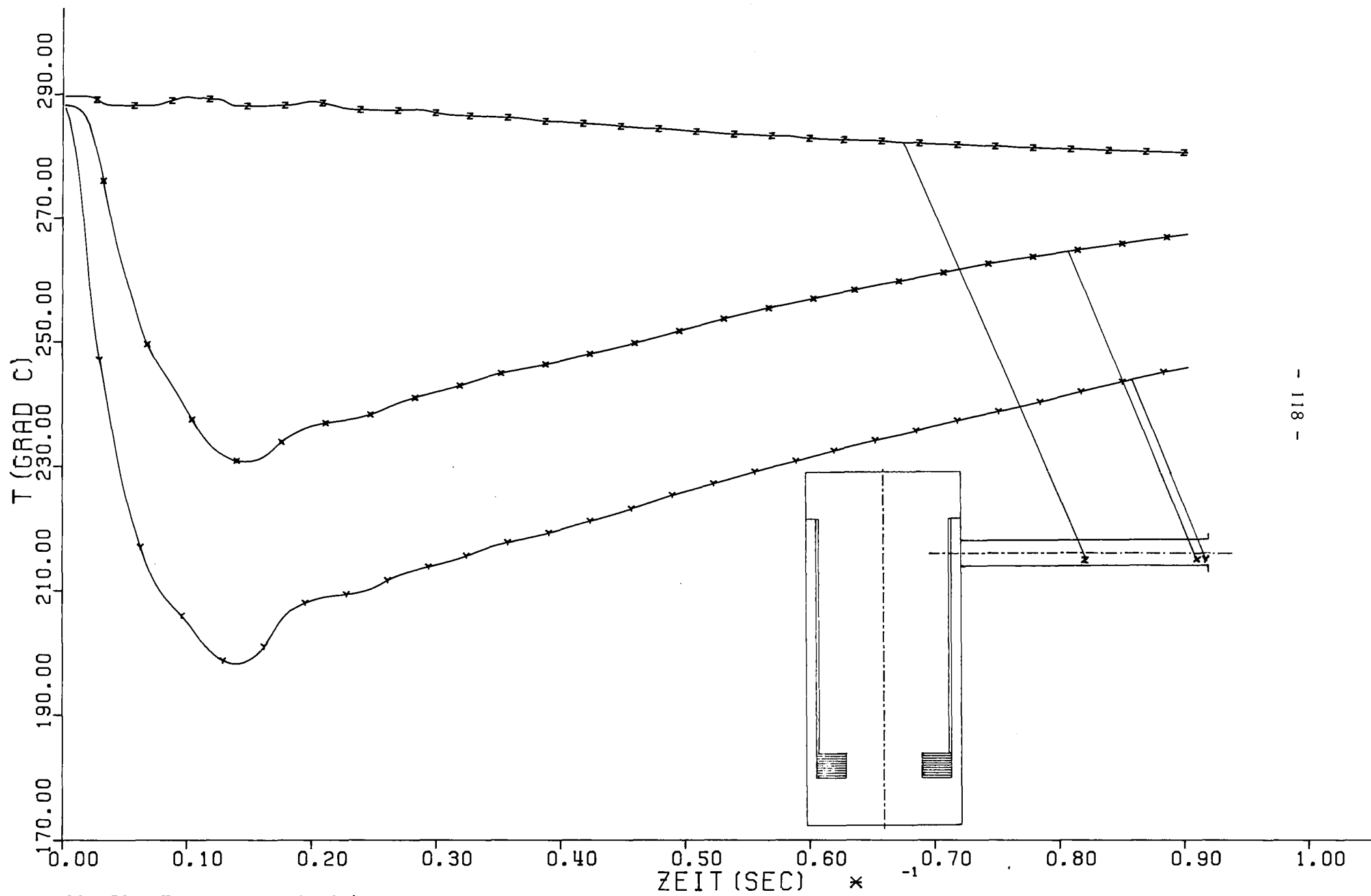


Abb. 53: Temperaturverlauf im Stutzen, Versuch 1

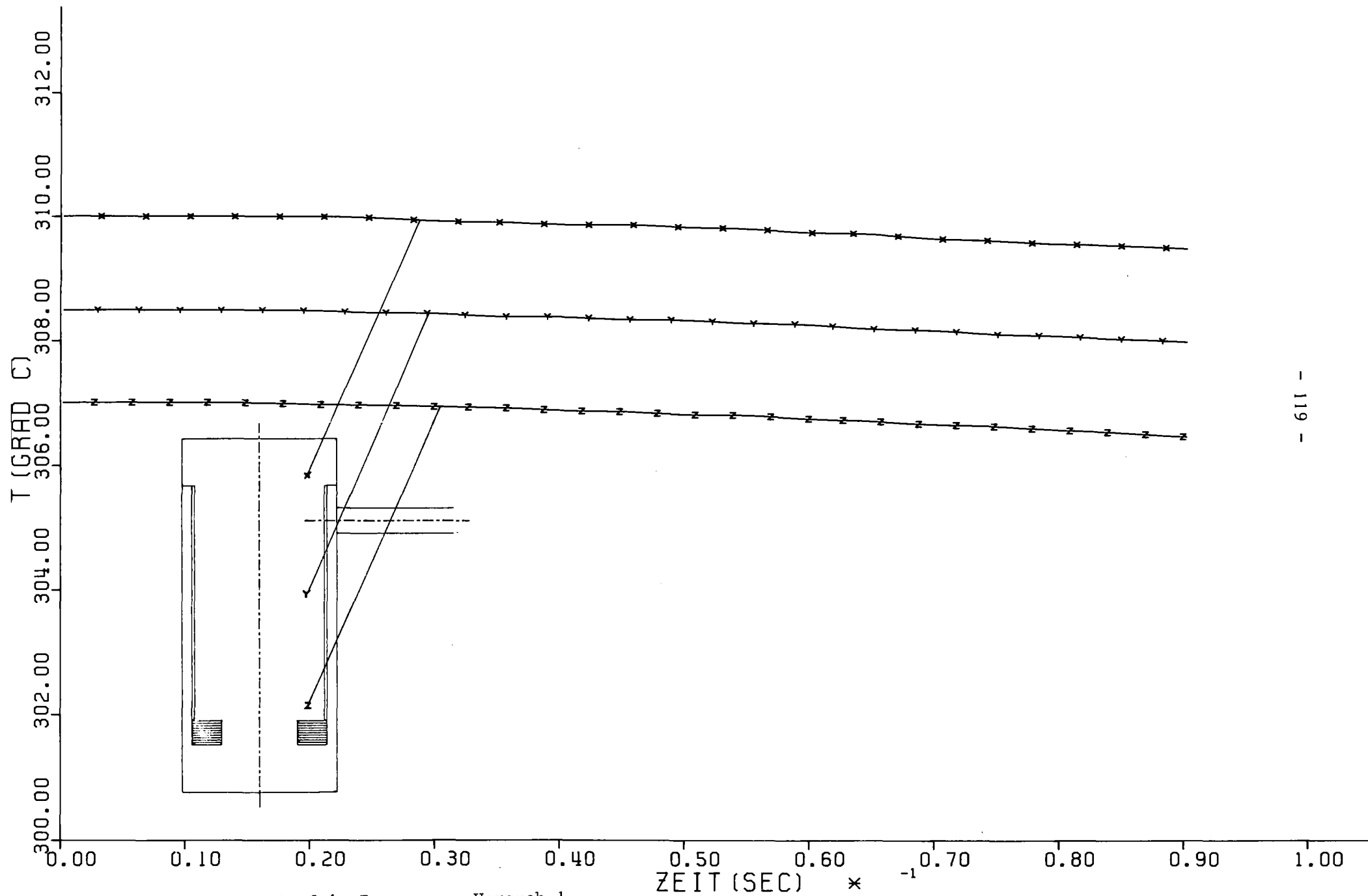


Abb. 54: Temperaturverlauf im Innenraum, Versuch 1

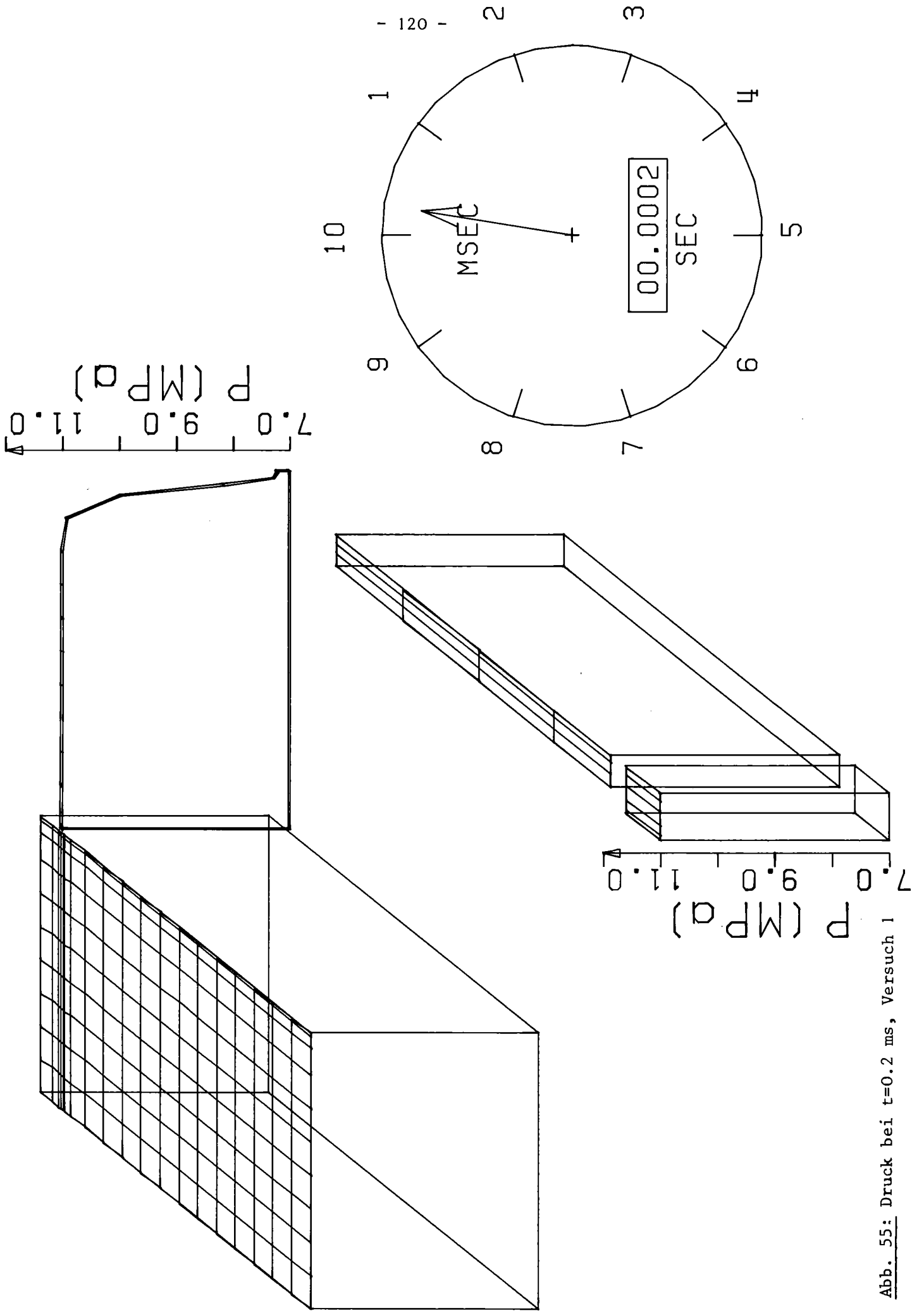


Abb. 55: Druck bei  $t=0.2$  ms, Versuch 1



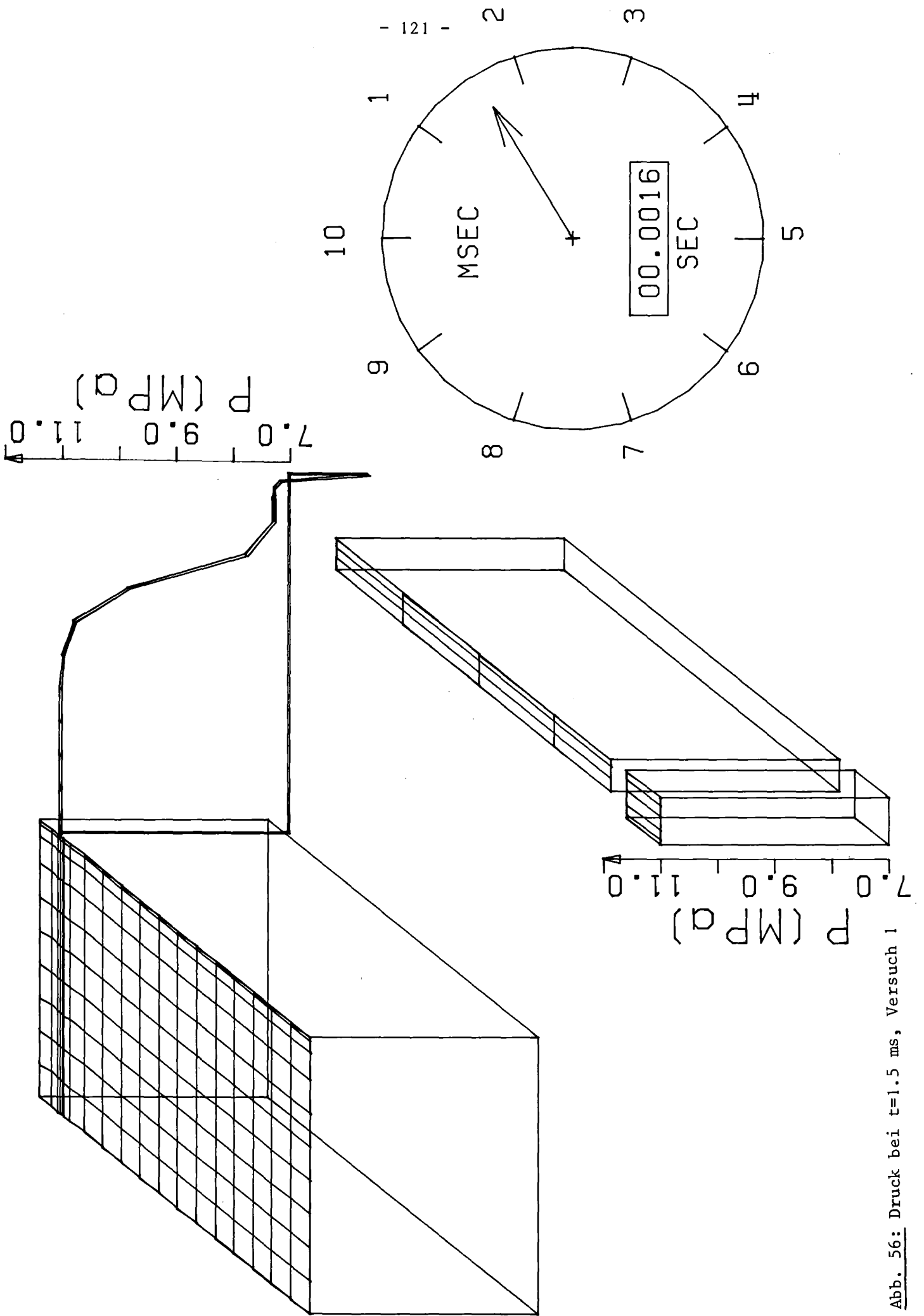


Abb. 56: Druck bei  $t=1.5$  ms, Versuch 1

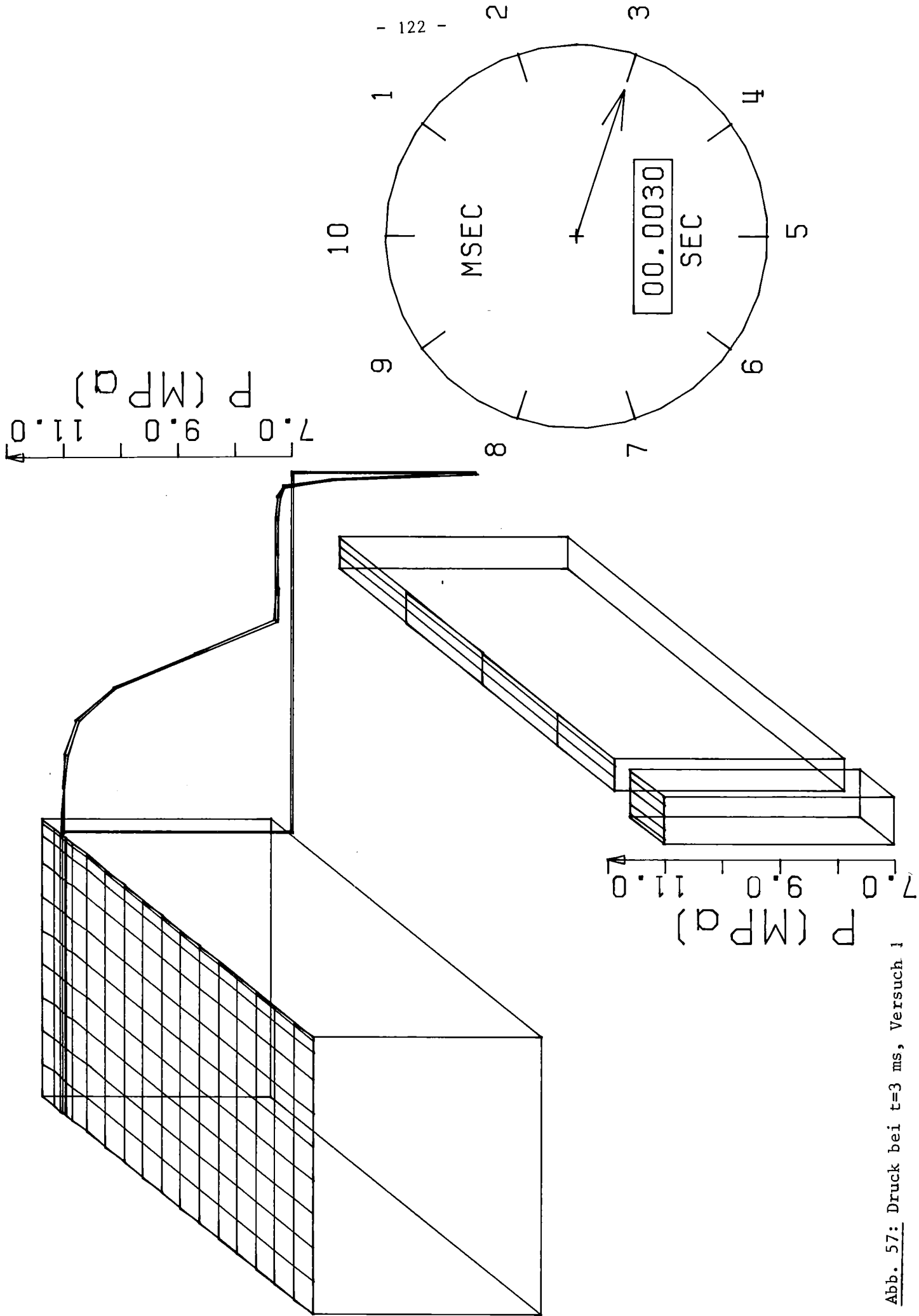


Abb. 57: Druck bei t=3 ms, Versuch 1

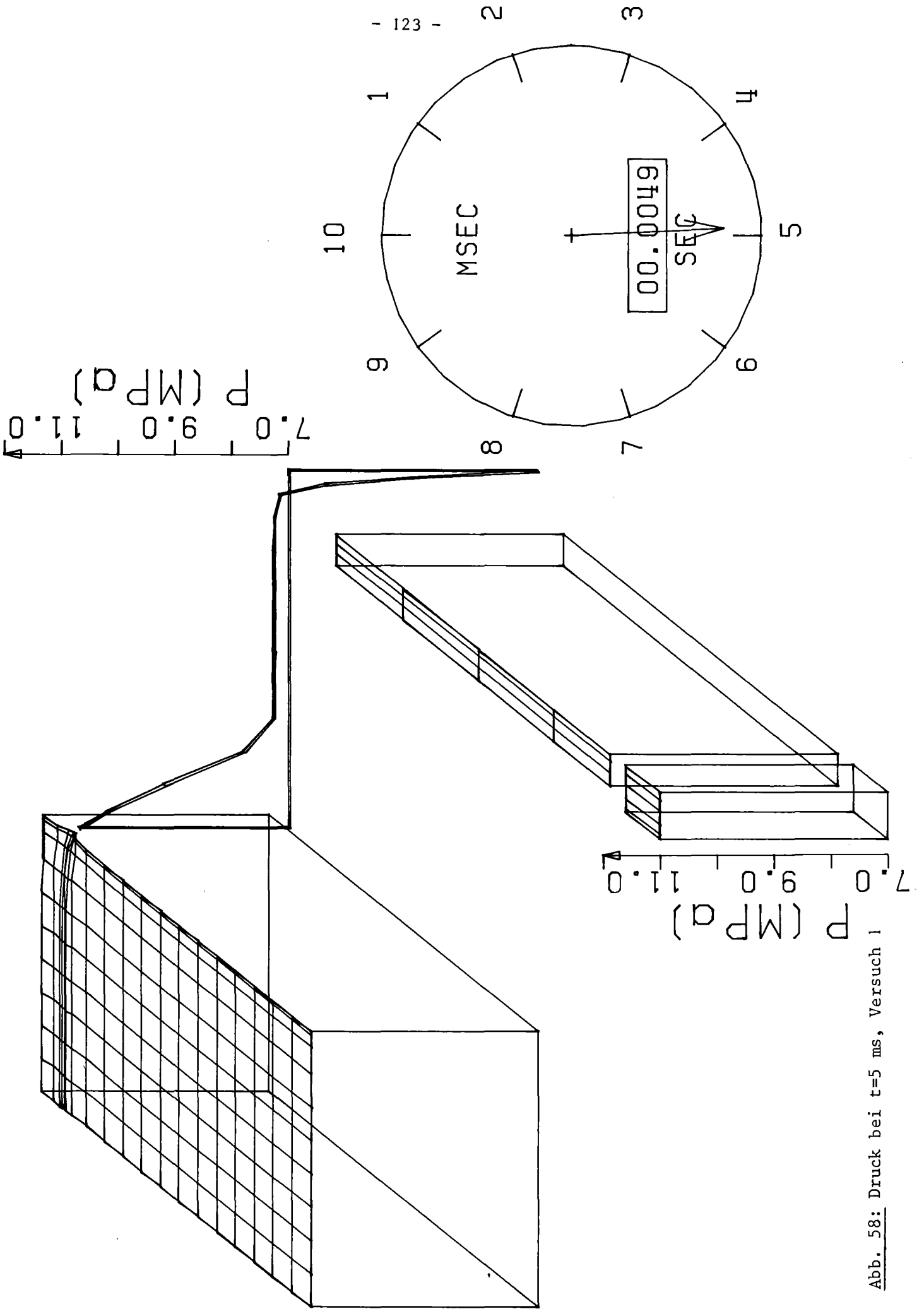


Abb. 58: Druck bei  $t = 5$  ms, Versuch 1

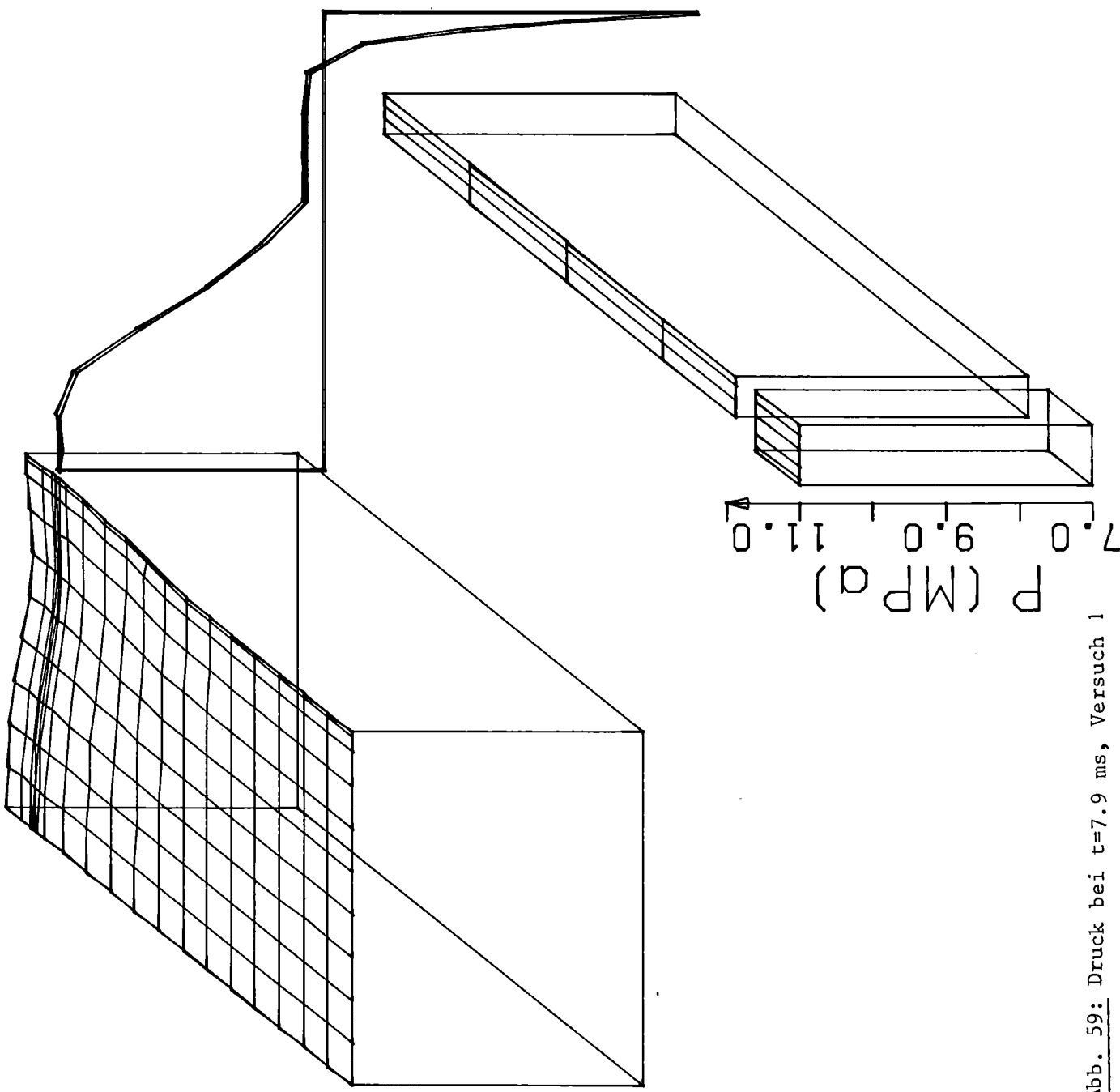
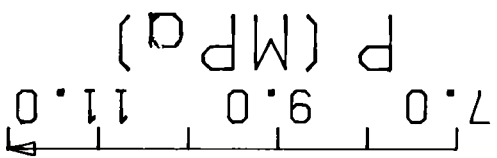
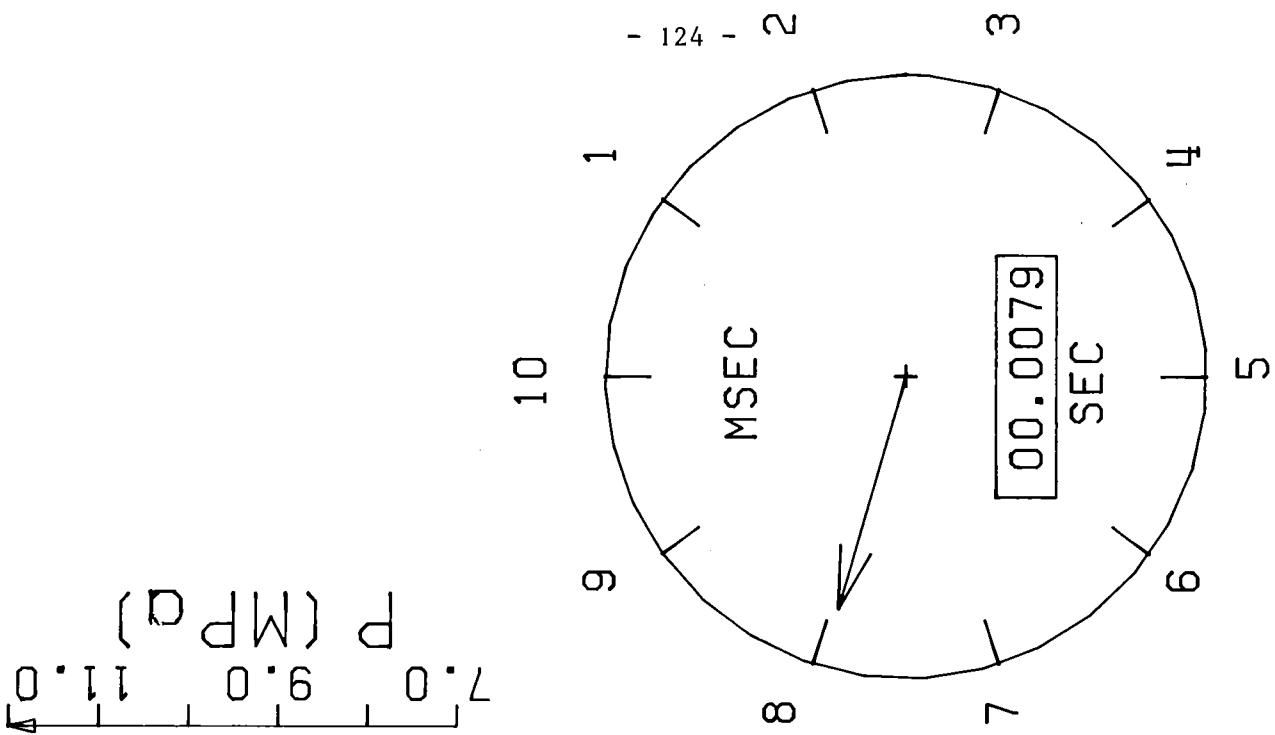


Abb. 59: Druck bei  $t=7.9$  ms, Versuch 1

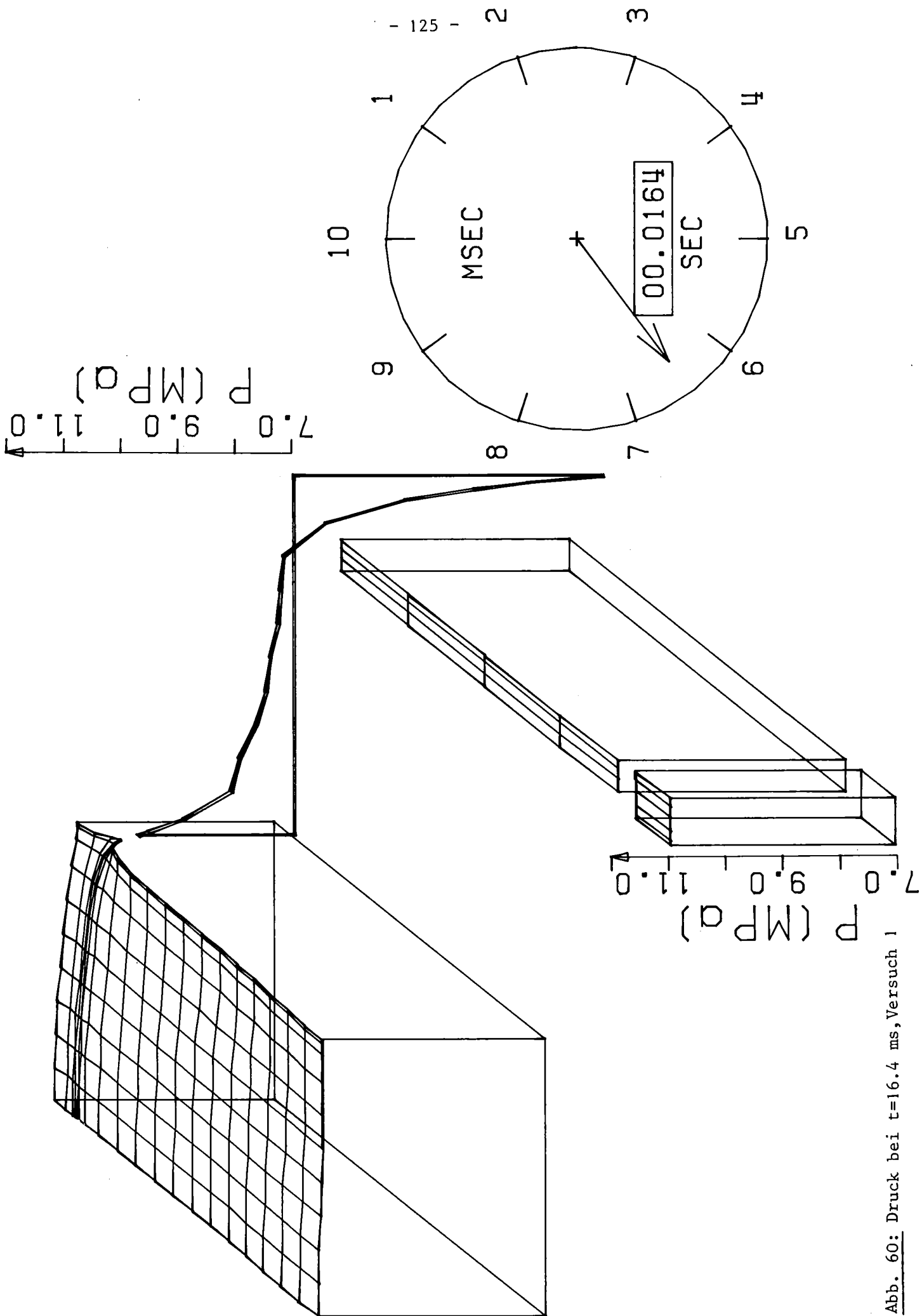


Abb. 60: Druck bei  $t=16.4$  ms, Versuch 1

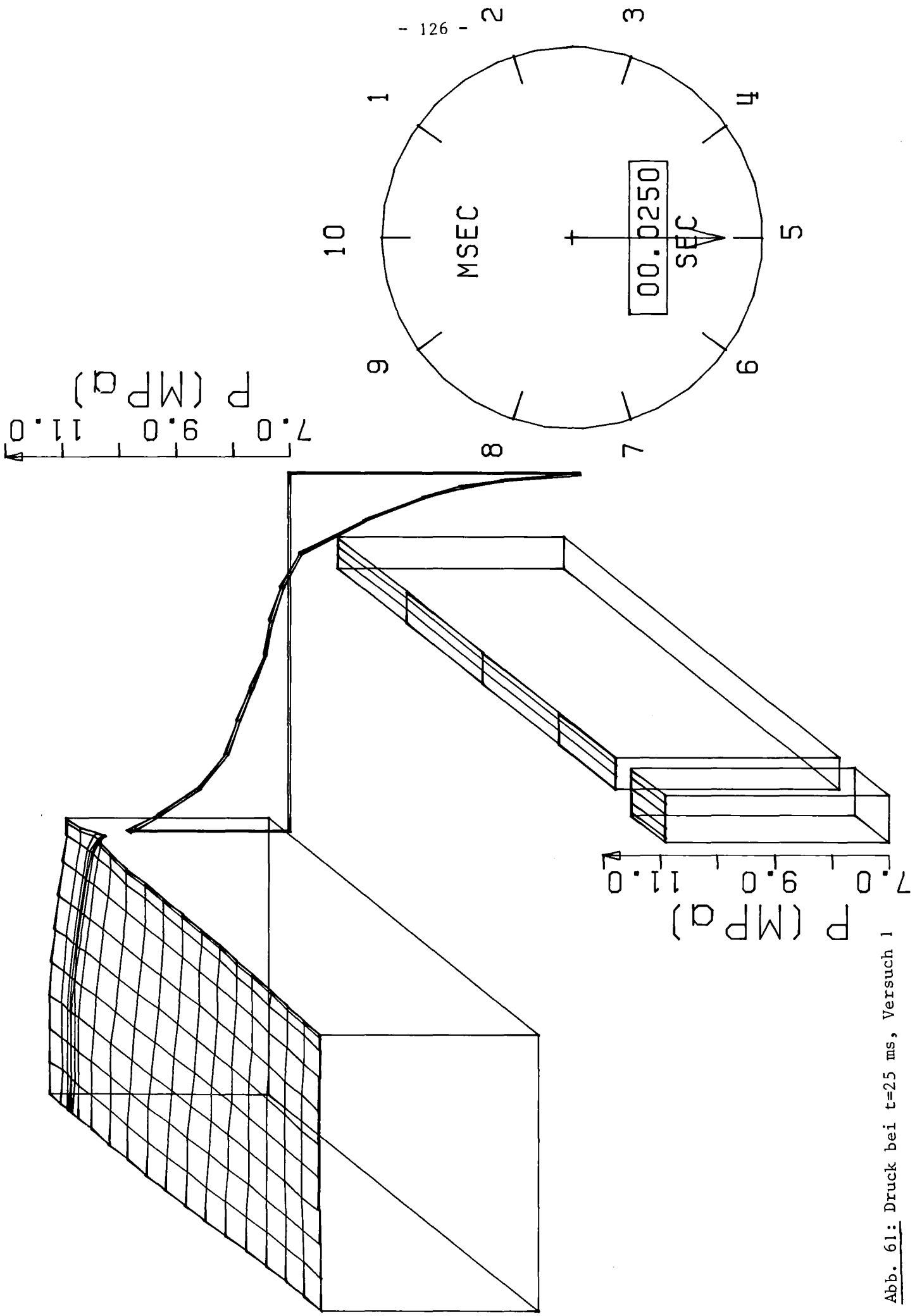


Abb. 61: Druck bei  $t=25$  ms, Versuch 1

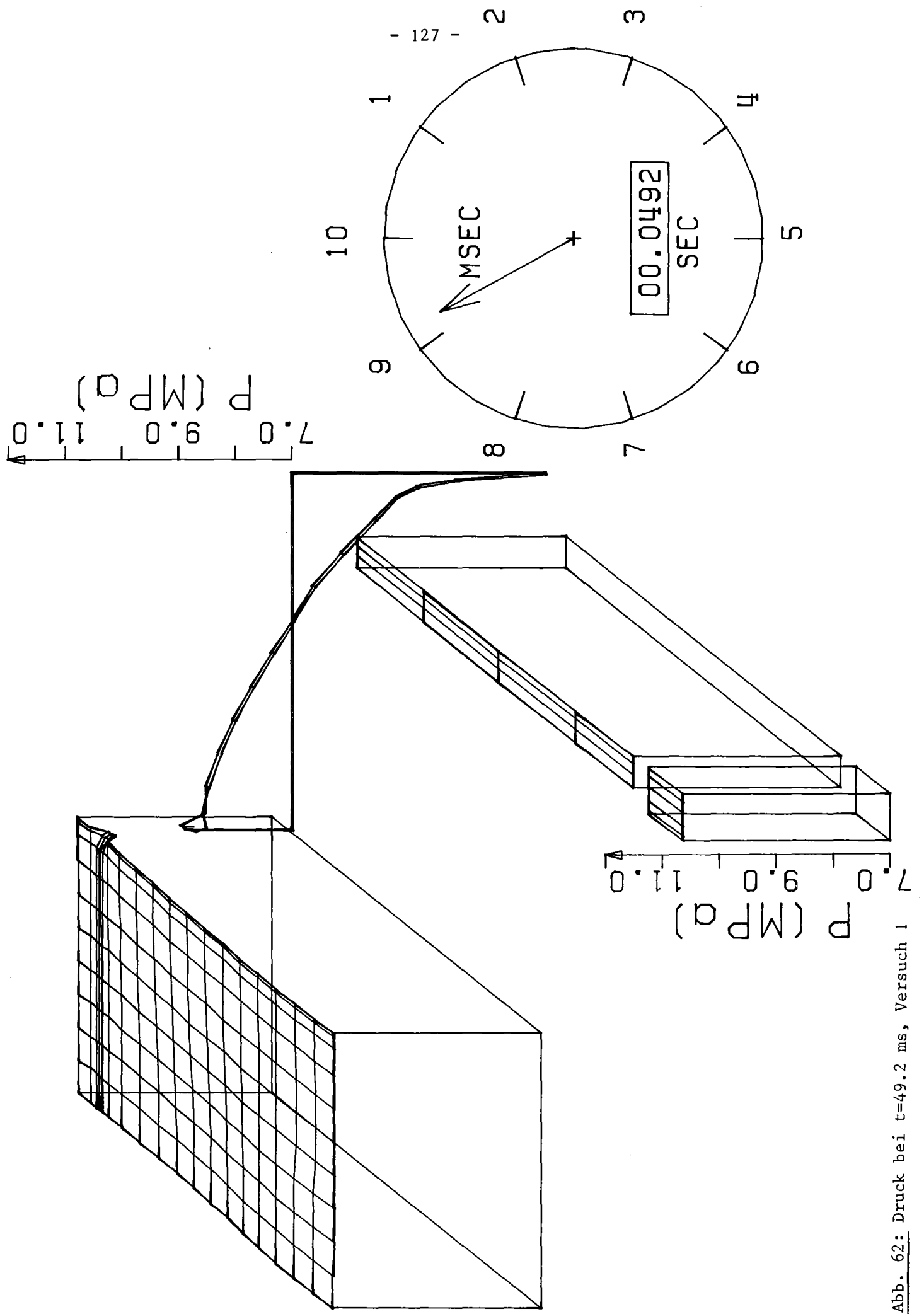


Abb. 62: Druck bei  $t=49.2$  ms, Versuch 1

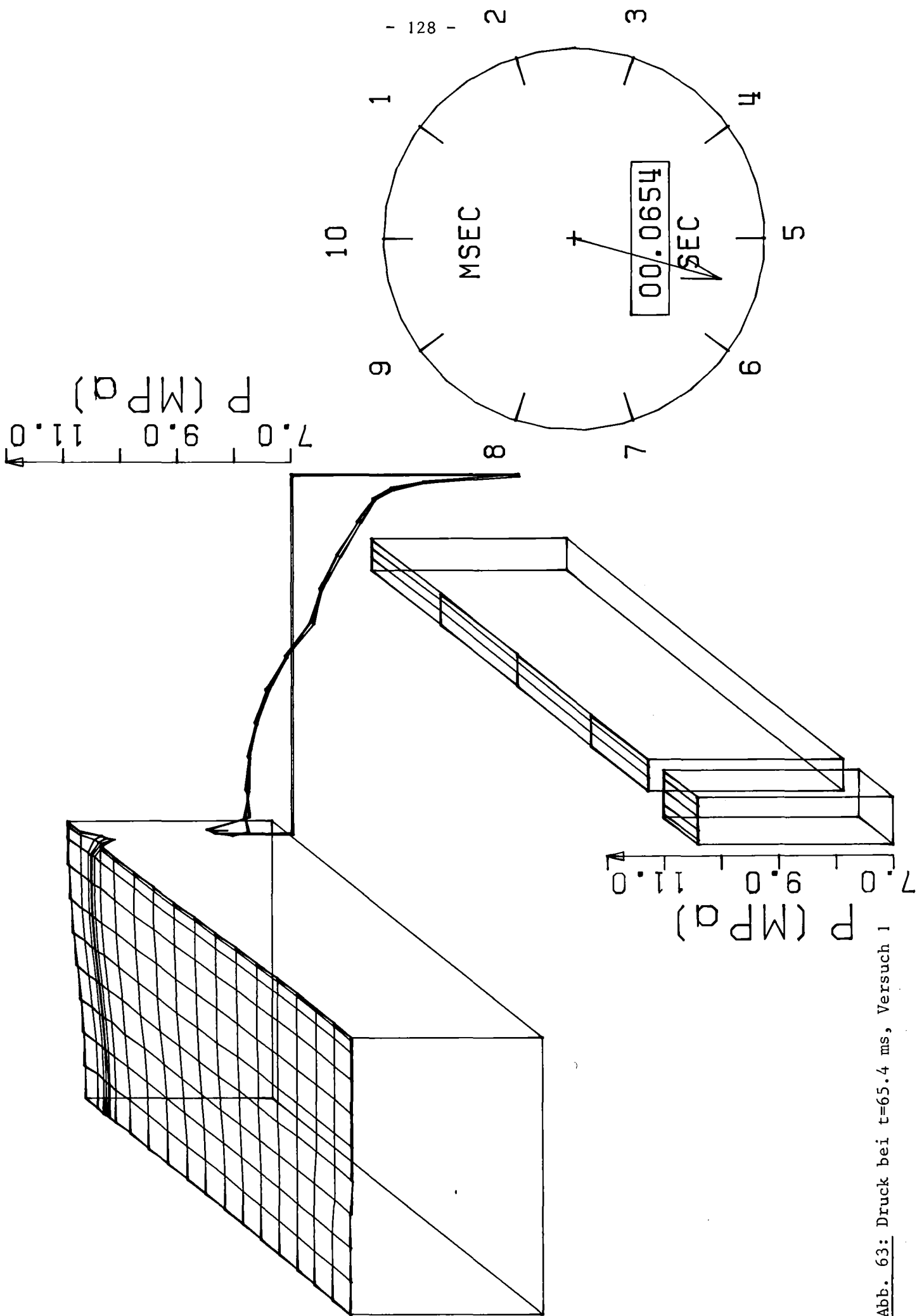


Abb. 63: Druck bei  $t=65.4$  ms, Versuch 1



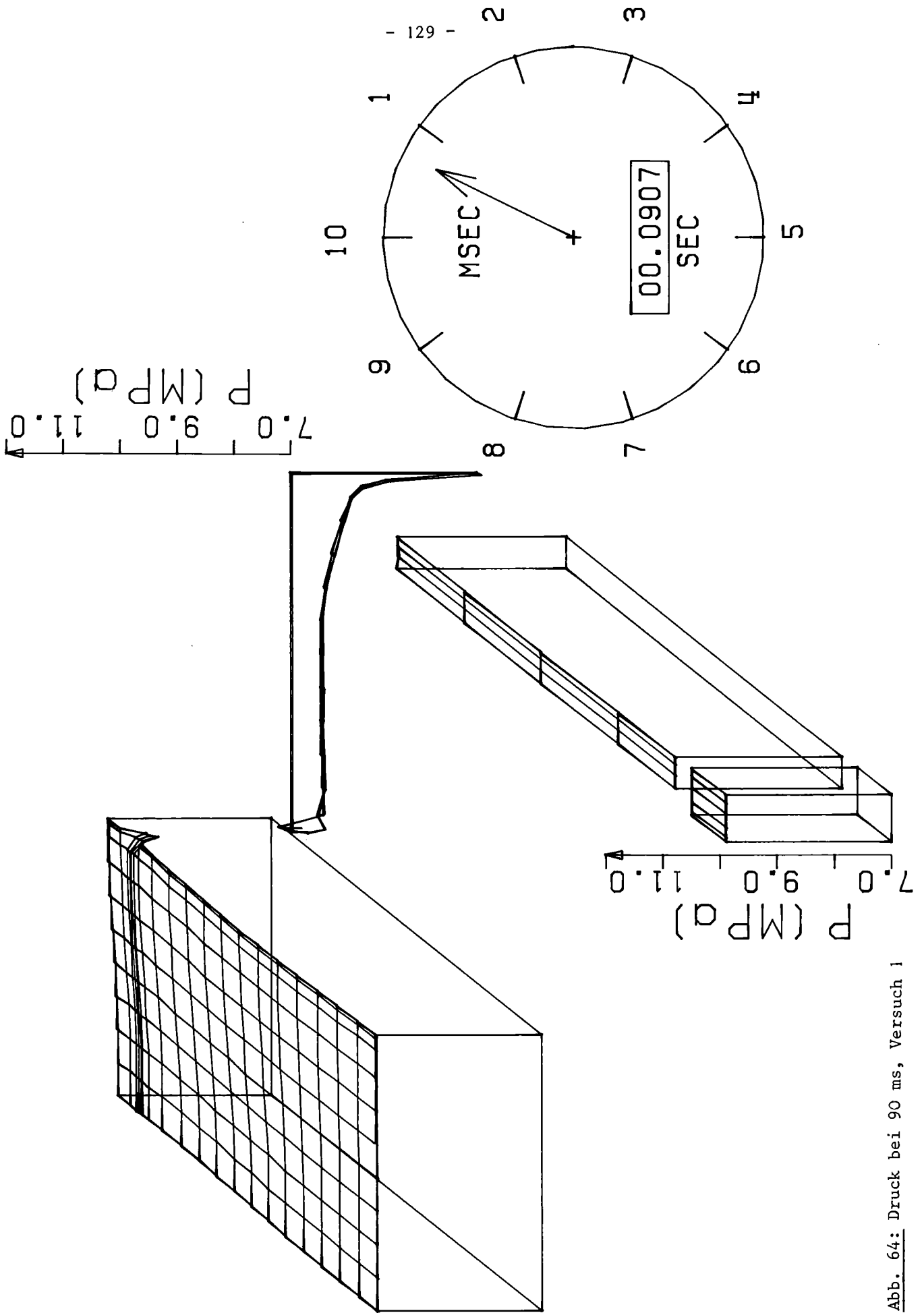


Abb. 64: Druck bei 90 ms, Versuch 1

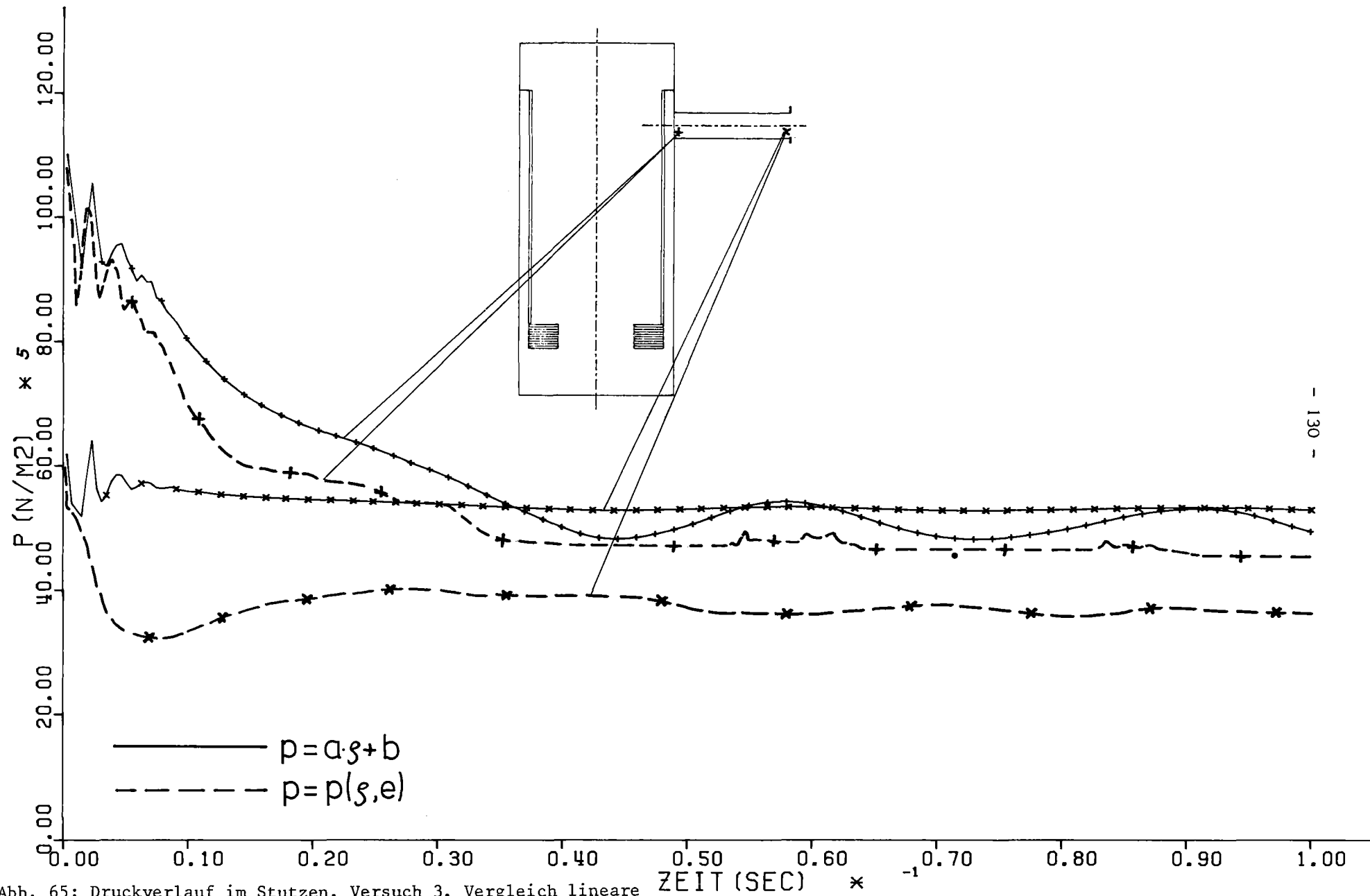


Abb. 65: Druckverlauf im Stutzen, Versuch 3, Vergleich lineare  
und genaue Wassergleichung

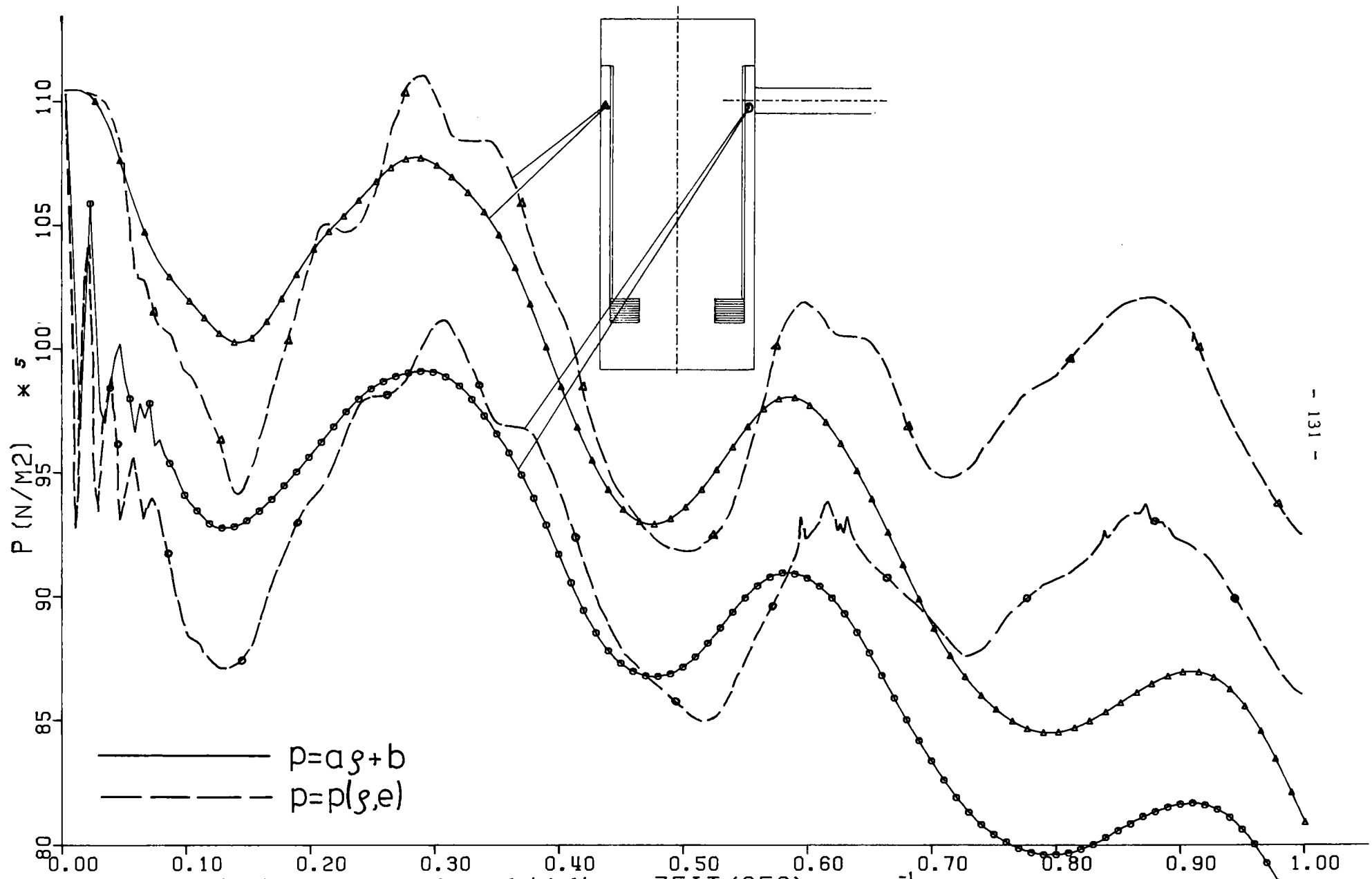


Abb. 66: Druckverlauf im Ringraum, Versuch 3, Vergleich lineare, genaue Wassergleichung

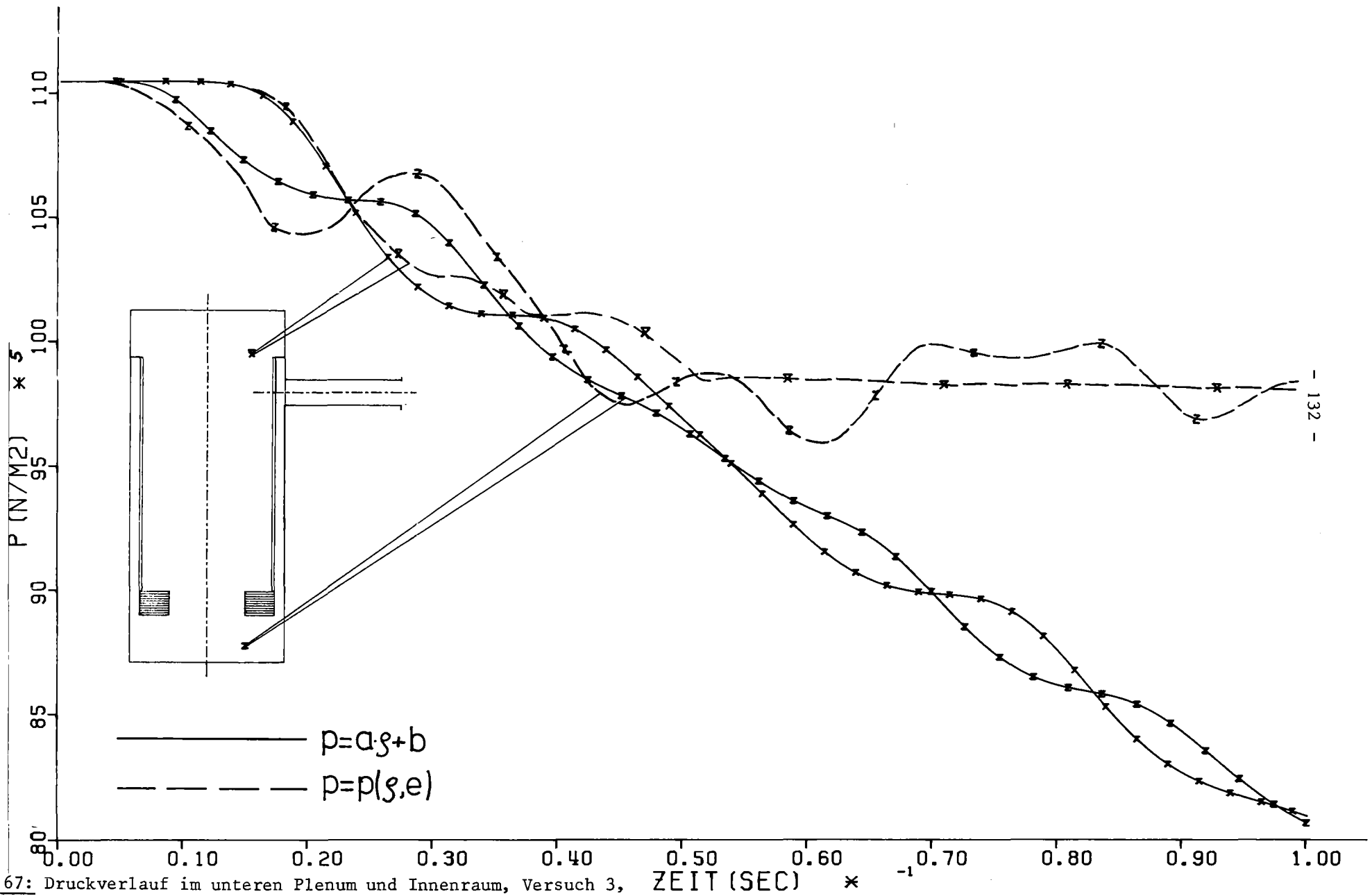


Abb. 67: Druckverlauf im unteren Plenum und Innenraum, Versuch 3, Vergleich lineare und genaue Wassergleichung

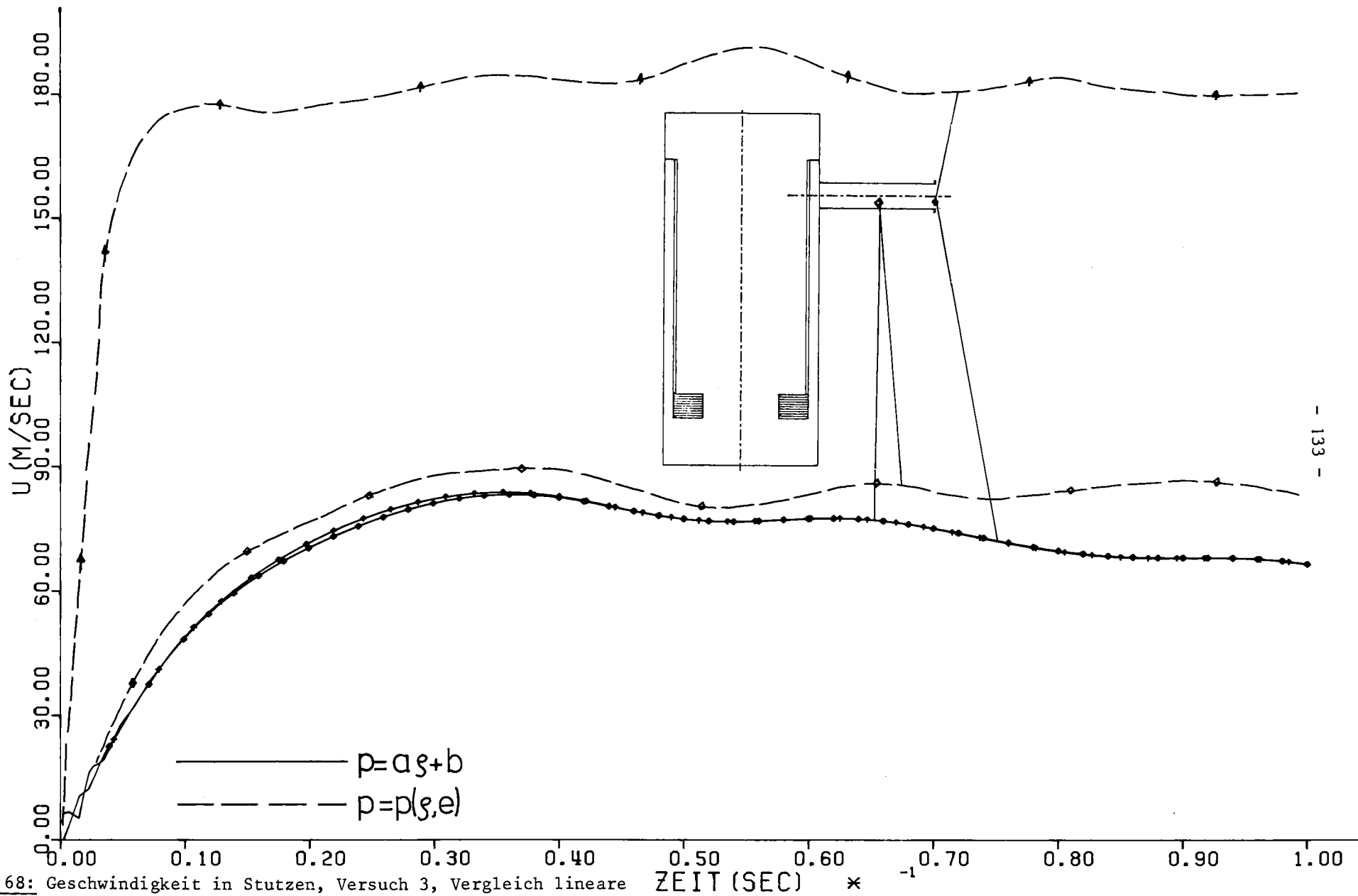


Abb. 68: Geschwindigkeit in Stutzen, Versuch 3, Vergleich lineare und genaue Wassergleichung

## 9. Erweiterung von FLUST um neue Modelle

Das Programm FLUST wurde von vornherein so angelegt, daß eine Koppelung zwischen verschiedenen Modellen für das Fluid möglich ist. Die Schritte, die erforderlich sind, um ein neues Modell in FLUST einzubringen, werden im folgenden beschrieben.

Die Erweiterung von FLUST um ein neues Modell ist auf zwei verschiedene Arten möglich:

- 1) Ankoppelung eines bestehenden Programms oder eines zu entwickelnden Modells an einem Rand, wobei die Programm- und Datenstrukturen nur an diesem Rand an die bestehende Form angepaßt werden müssen.
- 2) Beibehaltung der ICE-Technik und der Aufteilung der Berechnung in Module, die in FLUST schon existieren, lediglich Änderung der Gleichungen in den Moduln und der Feldgrößen-Arrays der Datenstruktur.

Die erste Methode ist dann sinnvoll, wenn ein komplettes schon existierendes Programm angeschlossen werden soll, daher wird dieser Weg für die Ankoppelung von FLUX [21] an FLUST eingeschlagen. Der RDB mit Fluid und Struktur wird dabei von FLUX berechnet, am Übergang zum Stutzen werden FLUST-Fluidbereiche angeschlossen. Die zweite Methode ist zur Erweiterung von FLUST innerhalb der ICE-Technik angemessen, z.B. zur Einbringung von dreidimensionalen Modellen, Zweiphasenströmungen oder Zylinderkoordinaten. Der zweite Weg ist für den Einbau des DRIX-2D-Modells [11] (zweiphasig, karthesische oder Zylinder-Koordinaten) vorgesehen.

### 9.1 Anschluß eines Programms an einem Rand

Die Grenze zwischen dem Teil des Fluids, der durch das neue Programm berechnet wird und dem FLUST-2D-Modell wird durch einen besonderen Rand dargestellt (s. Abb. 69). Lediglich der Kopf der Datenstruktur für das neue Modell und der Name und der Aufruf eines Moduls oder mehrerer Module muß sich an das FLUST-Schema halten. Die interne Programm- und Datenstruktur ist für FLUST ein "schwarzer Kasten". Von dem anzuschließenden Programm muß lediglich gefordert werden, daß es in der Lage ist, anhand der von FLUST zur Verfügung gestellten Zustandswerte diesseits des Randes zum alten und/oder neuen Zeitpunkt die Zustandswerte jenseits des Randes zum neuen Zeitpunkt zu berechnen.

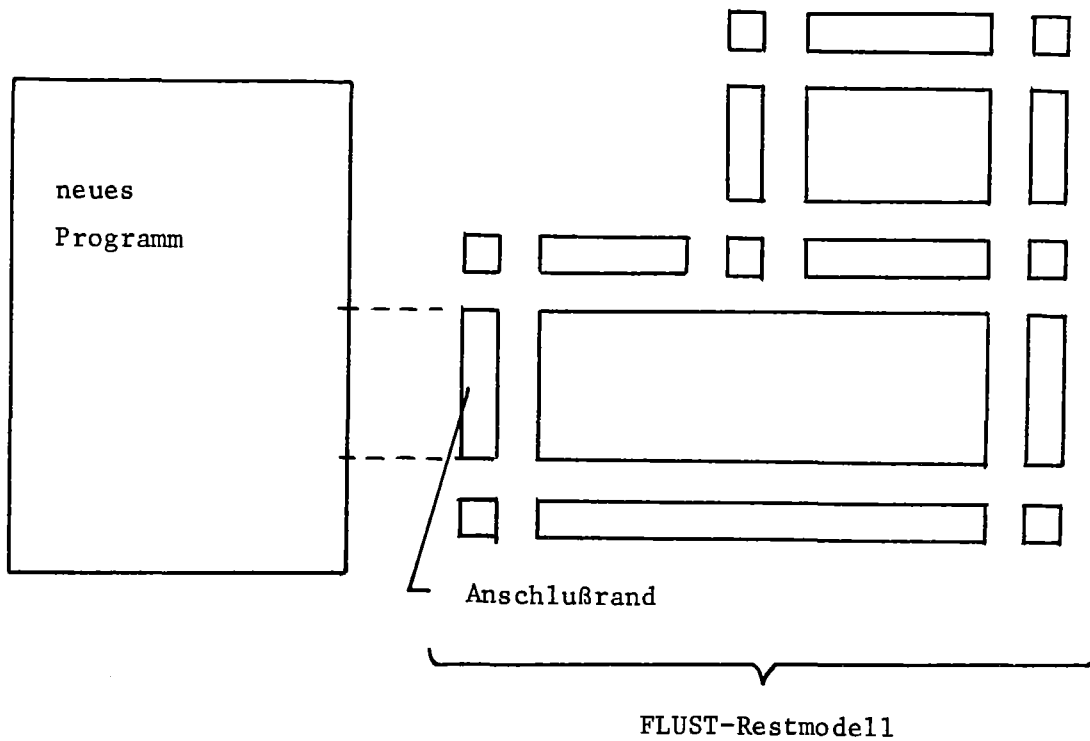


Abb. 69 : Anschluß eines vollständigen Programms

Die folgenden Schritte sind zur Anpassung erforderlich:

#### Datenstruktur

Alle Daten, die über einen Zeitschritt hinaus benötigt werden, müssen über die FLUST-Modellstruktur adressierbar gemacht werden (vgl. Abb. 15, S.47). Von der Komponentenstruktur des neuen Modells ist die Fluid-Kennzeichnung  $\alpha$  unbedingt erforderlich. Der Rest der Komponentenstruktur kann beliebig gewählt werden und u.a. dazu dienen, die Felder mit den Zustandsgrößen über Zeigervariable zu adressieren. Die Fluid-Kennzeichnung für das neue Modell darf nicht mit einer bereits bestehenden Kennzeichnung übereinstimmen.

#### Anpassen des Programms

Das Programm ist so zu ändern, daß es in einen Initialisierungsteil und einen oder mehrere Module pro Zeitschritt aufgeteilt werden kann. Jeder Modul hat über die Komponentenstruktur Zugriff zu den Daten. Der Initialisierungsmodul hat den Namen  $\alpha F1$  ( $\alpha$ =Fluidkennzeichnung), die Rechenmodule die Namen  $\alpha F2$  bis  $\alpha F11$ . Zur Aufruffolge siehe Abb. 14, S.39. Wenn das neue Programm nicht so viele Rechenmodule wie die ICE-Technik benötigt, werden

die nicht benötigten Plätze der Entry-Tabelle mit dem Routinen-Namen DUMMY gefüllt, der Aufruf dieser Module wird beim Durchgang durch das Gesamtmodell übergangen. Das Allokieren der Datenstruktur und das Setzen von Anfangsbedingungen erfolgt in dem Modul, dessen Name in der ENTER FLUSH-Anweisung angegeben wird und der das gesamte Modell festlegt.

### Rand

Der Rand zwischen den beiden Modellen kann wie ein Rand zwischen zwei FLUST-2D-Fluiden gestaltet sein. Dann sind für die Übertragung von Werten zwischen Rand und Naht und zum Setzen von Randbedingungen die erforderlichen Module schon vorhanden. Lediglich zum Übertragen von Werten vom Rand zum neuen Modell und zurück ist eine Routine  $\alpha$ TRFB zu erstellen ( $\alpha$ =Kennzeichnung des neuen Modells). Diese Routine kann nötige Anpassungen wie Änderung des Maschennetzes, Umrechnung von Zustandsgrößen oder Mittelwertbildungen übernehmen.

### Ablauf der Rechnung

Die Reihenfolge, in der die beiden Modelle vom Zeitpunkt  $n$  ausgehend die neuen Zustandsgrößen zum Zeitpunkt  $n+1$  errechnen, kann durch die Fluid-Numerierung festgelegt werden. Jedes Modell kennt (über den Rand) die alten Zustandsgrößen im Nachbarmodell. Die Geschwindigkeit an der Grenzfläche, die von jedem Modell errechnet wird, muß nicht unbedingt übereinstimmen, da beide Modelle verschiedene Verfahren benutzen. Um die Kontinuitätsgleichung nicht zu verletzen ist es daher von Vorteil, wenn das an zweiter Stelle gerechnete Modell die vom Nachbarmodell schon ermittelte Geschwindigkeit als Randbedingung benutzt.

### Beispiel

Folgende Tabelle 4 enthält die Module, die beim Anschluß von FLUX an FLUST neu erstellt werden müßten, mit ihren Aufgaben. Die Kennzeichnung für das FLUX-Modell sei "XX".



Modulname	Aufgabe
Gesamtmodell	Allokieren der Datenstrukturen für beide Modelle. Setzen von Anfangsbedingungen für $\rho, e, u, v$
XXF1	Initialisieren der restlichen Zustandsgrößen
XXF2	Errechnen der Zustandsgrößen $p, \rho, u, v$ zum neuen Zeitpunkt
XXTRFB	Übertragen von Werten aus dem FLUX-Modell auf den Rand und zurück, Setzen einer konstanten Energie auf dem Rand

Tabelle 4: Module zum Ankoppeln von FLUX an FLUST

### 9.2 Einbau eines neuen FLUST-Modells

Der Ablauf der Rechnung und die Aufteilung in Einzelschritte wird vom FLUST-2D-Modell übernommen. Dadurch wird die Flexibilität von FLUST, beliebige Bereiche aneinanderzukoppeln, beibehalten. Für jeden in FLUST-2D vorhandenen Rechenmodul muß ein entsprechender Modul für das neue Modell erstellt werden. Abb. 70 verdeutlicht die Koppelung zweier Modelle, die beide auf der FLUST-Grundstruktur aufbauen.

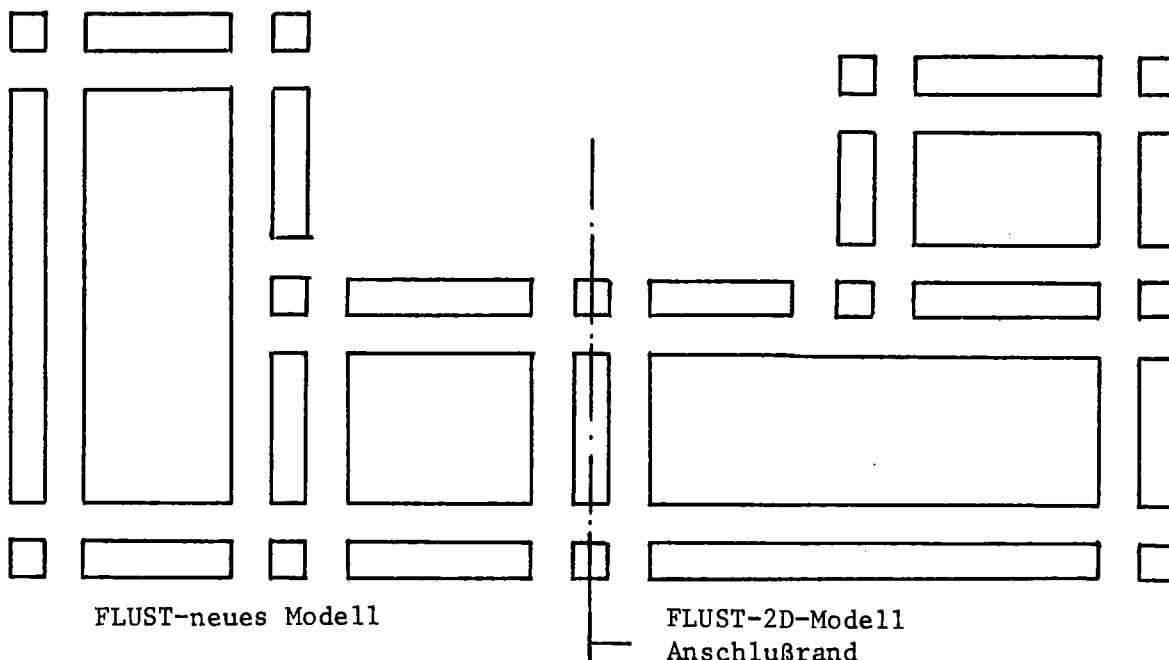


Abb. 70 : Anschluß eines neuen FLUST-Modells

Die folgenden Schritte sind zum Einbau eines neuen Modells in FLUST erforderlich:

#### Datenstruktur

Die Feldgrößen-Arrays zur Aufnahme der Zustandsgrößen und Hilfsgrößen müssen entsprechend dem neuen Modell erweitert oder geändert werden. Die Modellstruktur und die Komponentenstrukturen bleiben unverändert. Das neue Modell erhält eine neue Fluid-Kennzeichnung  $\alpha$ .

#### Programmstruktur

Die Aufteilung in Module bleibt unverändert. Es muß je ein Modul  $\alpha$ TRFB,  $\alpha$ TRBS,  $\alpha$ BCND,  $\alpha$ Fi,  $\alpha$ Ri und  $\alpha$ Ni für das neue Modell erstellt werden. Der Durchgang durch das Gesamtmodell wird wie beim FLUST-2D-Modell durchgeführt.

#### Rand

Auch beim neuen Modell sind die Fluide von Rändern und Nähten umgeben, die von den Moduln  $\alpha$ Ri und  $\alpha$ Ni behandelt werden. Der Rand an der Grenzfläche beider Module müßte die Eigenschaften beider Seiten berücksichtigen. Eine besondere Art von Rand und Naht wäre für diesen Zweck bereitzustellen. Der dafür erforderliche Aufwand kann reduziert werden, wenn an der Grenzfläche zwei Ränder aneinandergesetzt werden, je einer für jedes Modell. Die Übertragung von Werten zwischen den beiden Rändern wird durch die Randbedingung-Routine  $\alpha$ BCND vorgenommen. Diese Methode wird in FLUST-2D benutzt, um zwei Fluide im Winkel von  $90^\circ$  zueinander zu koppeln.

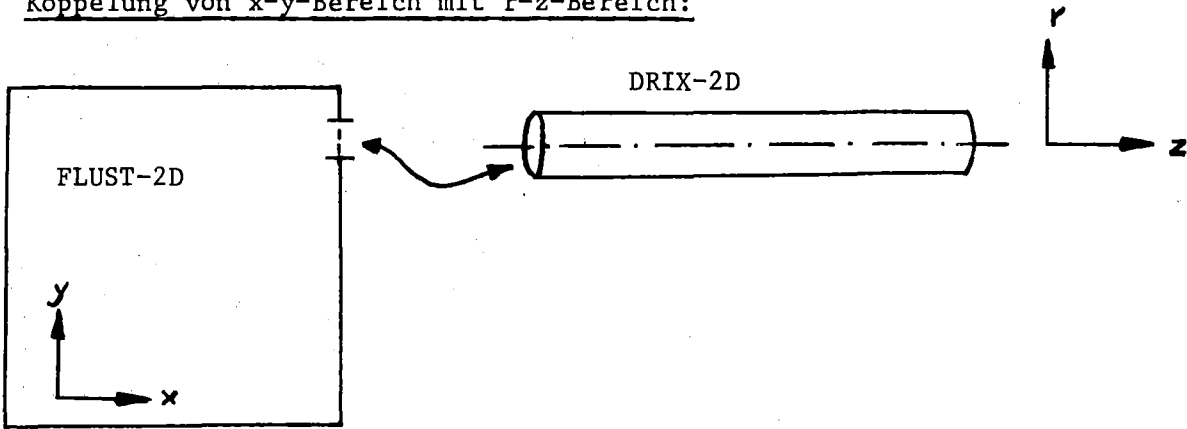
#### Beispiel

Die beim Einbau des DRIX-2D-Modells in FLUST erforderlichen Module sind in Tabelle 5 mit ihren Aufgaben zusammengestellt. Abb. 71 verdeutlicht die Gestaltung des Randes beim Übergang von einem x-y-Bereich zu dem neuen r-z-Modell. Die Fluid-Kennzeichnung sei "DD".

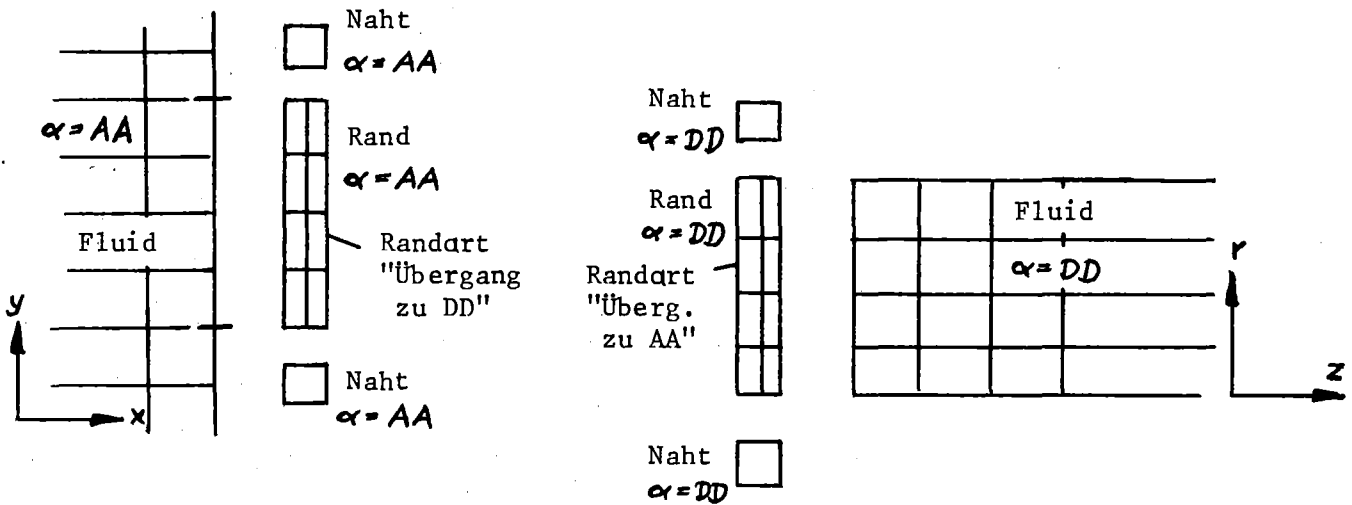
Modulname	Aufgabe
Gesamtmodell	Allokieren der Datenstrukturen für beide Modelle, Setzen der Anfangsbedingungen für $\rho, e, u, v$
DDR1 } DDF1 }	Initialisieren der restlichen Zustandsgrößen
DDN3 } DDR2 } DDR3 }	neue Schätzwerte $\tilde{p}, \tilde{\xi}$
DDF2 } DDF3 }	
DDN4 } DDN6 }	neue Schätzwerte $\tilde{u}, \tilde{v}, \tilde{\xi u}, \tilde{\xi v}$
DDR4 } DDR5 } DDR6 }	neues $\beta, 1/c^2$
DDF4 } DDF5 } DDF6 }	
DDFTIM	timestephalf, timestepdouble
DDF7 } DDF8 } DDF9 }	Iteration $\tilde{p}, \tilde{\xi}, \tilde{u}, \tilde{v}, \tilde{\xi v}$ (siehe [11_7])
DDR8 } DDR9 }	Berechnung von $u_r, v_r$
DDR10 } DDF10 }	$p=\tilde{p}, \xi=\tilde{\xi}, u=\tilde{u}, v=\tilde{v}$
DDR11 } DDF11 }	neue Energie $e$ , Temperatur, Reibkräfte
DDTRFB	Übertragen der Werte Fluid ↔ Rand
DDTRBS	Übertragen der Werte Rand ↔ Naht
DDBCND	Randbedingungen

Tabelle 5: Module zum Eingliedern des DRIX-2D-Modells in FLUST

Koppelung von x-y-Bereich mit r-z-Bereich:



Maschennetze an der Grenzfläche:



Module zum Rechnen und Übertragen von Werten:

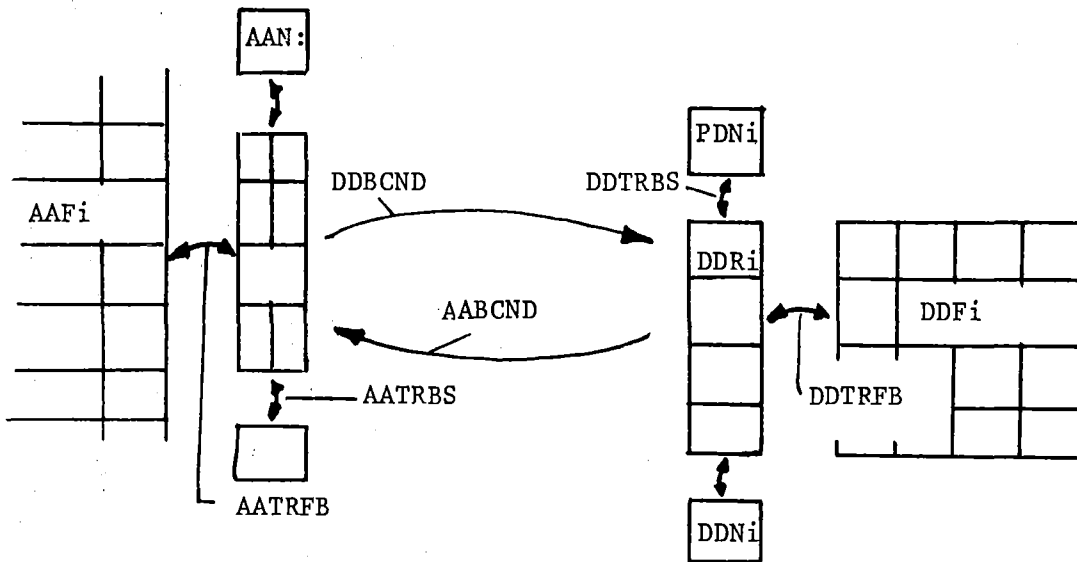


Abb. 71: Übergang zwischen zwei FLUST-Modellen

An der Grenzfläche sind beide Modelle durch je einen Rand abgeschlossen. An der Außenseite hat dieser Rand die Randart "Übergang zum anderen Modell". Das Übertragen von Werten vom Fluid auf den Rand geschieht durch die Module AATRFB und DDTRFB. Auch die angrenzenden Nähte werden wie andere zum gleichen Modell gehörende Nähte behandelt. Die eigentliche Ankoppelung erfolgt durch die Routinen AABCND und DDBCND, die sonst zum Setzen von Randbedingungen benutzt werden. Da das x-y-Modell am Grenzrand eine Variation der Zustandsgrößen in y-Richtung zuläßt, das r-z-Modell aber Rotationssymmetrie voraussetzt, muß beim Übergang jeweils über die Grenzmaschen gemittelt werden. Der Mittelwert des einen Modells wird dann dem anderen Modell als Randbedingung in allen Maschen angeboten. Die einzigen Module, die so etwas von der Struktur des anderen Modells wissen müssen, sind die Randbedingungsmodule AABCND und DDBCND.

### 9.3 Mögliche Vereinfachungen in FLUST

Vor einer Erweiterung von FLUST sollte geprüft werden, ob nicht zunächst das hier beschriebene FLUST-2D-Modell vereinfacht oder in Richtung auf höhere Effektivität optimiert werden sollte. Im folgenden werden dazu vier konkrete Vorschläge gemacht, von denen die ersten beiden sofort problemlos verwirklicht werden können. Für die anderen Vorschläge sind außer dem Umstellungsaufwand noch zusätzliche theoretische Überlegungen erforderlich.

#### 1. Zusammenfassung von Moduln

Wie aus Tabelle 1 (S. 40) ersichtlich ist, werden an mehreren Stellen in FLUST-2D zwei oder drei Module für die gleiche Komponente hintereinander aufgerufen, z.B. AAF2 und AAF3. Durch Zusammenfassen dieser Aufrufe kann die Modulanzahl und damit die Anzahl Aufrufe pro Zeitschritt reduziert werden. Tabelle 6 enthält eine Liste der Module, die zusammengefaßt werden kann. Die Zahl der Rechenmodule geht durch diese Maßnahme von 25 auf 16 zurück. Darüber hinaus werden in FLUST-2D die Routinen AATRFB und AABCND für verschiedene Zustandsgrößen nacheinander mehrfach aufgerufen. Eine Zusammenfassung mehrerer Aufrufe zu einem kann die Aufrufanzahl pro Zeitschritt von 58 auf 26 verringern.

Alte Modulnamen	Neuer Name
$\alpha F2, \alpha F3$	$\alpha F2$
$\alpha F4, \alpha F5, \alpha F6$	$\alpha F3$
$\alpha F7, \alpha F8, \alpha F9$	$\alpha F4$
$\alpha F10$	$\alpha F5$
$\alpha F11$	$\alpha F6$
$\alpha R4, \alpha R5, \alpha R6$	$\alpha R4$
$\alpha R8, \alpha R9$	$\alpha R5$
$\alpha R10$	$\alpha R6$
$\alpha R11$	$\alpha R7$
$\alpha N4, \alpha N6$	$\alpha N4$

Tabelle 6: Zusammenfassung von Moduln

## 2. Optimierung des Dynamic-Array-Zugriffs

Alle Feldgrößen in FLUST werden in den dynamischen Datenfeldern des REGENT-Systems gespeichert. Beim Durchgang über alle Komponenten werden die benötigten Teilfelder im Speicher festgelegt (FIX-Anweisung in PLR) und wieder freigegeben (LOOSE-Anweisung). Das Festlegen und Freigeben erfolgt in jedem Modul für die benötigten Feldgrößen. Zwischen FIX und LOOSE werden die Daten über PL/1-BASED-Felder adressiert, der Zugriff ist dadurch sehr schnell. FIX und LOOSE führen jedoch jeweils zu einem Aufruf der REGENT-Routinen QQDFIX und QQDLOS, je ein Aufrufpaar erfordert eine Rechenzeit von etwa  $4 \cdot 10^{-4}$  s. Wenn genügend Speicherplatz zur Verfügung steht um alle Datenfelder im Hauptspeicher zu halten, kann das Festlegen der Daten im Speicher einmal vor Beginn der Rechnung erfolgen, die Adressierung der Daten kann über Zeigervariable erfolgen, die der Modellstruktur ähnlich wie die ENTRIES-Matrix (vgl. S.38) angegliedert ist. Diese Programmieretechnik wird im einzelnen in [22] beschrieben. Für das HDR-Modell erfolgt etwa 8000 Mal pro Zeitschritt ein FIX/LOOSE, das entspricht 3.25 s CPU-Zeit. Durch das Einsparen dieser Aufrufe wird eine Reduzierung der Rechenzeit auf höchstens 60% des jetzigen Wertes erwartet.

### 3. Eliminieren der Außenränder

Ein Großteil der Rechenzeit wird in Komponenten verbraucht, die das Fluid nach außen hin abgrenzen und lediglich die Randbedingung "feste Wand" darstellen. Könnte man auf alle Ränder und Nähte verzichten, die nicht der Verbindung zweier Fluide dienen, könnte z.B. beim HDR-Modell die Anzahl Ränder von 26 auf 5 und die Anzahl Nähte von 24 auf 0 (Null!) zurückgehen. Dazu wäre es jedoch erforderlich, die Randbedingung in der Fluidkomponente festzulegen. Man könnte an eine Matrix denken, die für jede Maschengrenze im Fluid vor allem also auch an den Außenflächen, die zur Verfügung stehende Durchfluß-Fläche angibt. Damit könnten nicht nur die Außenwände angegeben werden, es könnten auch leicht Hindernisse und Wände im Fluid modelliert werden. Die Formeln zur Strömungsberechnung im Fluid müßten jedoch an diese Vorgehensweise angepaßt werden. Diese Maßnahme würde nicht nur mindestens 50% Rechenzeit sparen, sie würde darüber hinaus die Flexibilität beim Modellieren von Einbauten im Fluidbereich erhöhen.

### 4. Verschiedene Zeitschrittweiten für Fluidbereiche

Im gegenwärtigen FLUST-Modell werden alle Fluidbereiche mit einer gemeinsamen Zeitschrittweite berechnet. Diese Zeitschrittweite wird durch die Stabilitätsbedingung (Gl. 43, S.18) begrenzt. Für die HDR-Rechnungen wird die maximale Zeitschrittweite durch die Maschen außerhalb der Bruchstelle bestimmt, dort herrschen bei kleinster Maschenweite (0.05 m) die größten Geschwindigkeiten (einige 100 m/s). Der Ringraum, der Innenraum und die Plena könnten mit einer wesentlich größeren Zeitschrittweite gerechnet werden, was eine erhebliche Rechenzeit-Einsparung bedeuten würde. Es sollte daher die Möglichkeit geben, jeden Fluidbereich entsprechend der für ihn geltenden Stabilitätsbedingung mit einer anderen Zeitschrittweite zu berechnen. Wenn alle Zeitschrittweiten das  $2^n$ -fache eines kleinsten Wertes betragen, werden zu bestimmten Zeitpunkten alle Fluidbereiche wieder synchronisiert. In denjenigen Bereichen, die zu einem Zeitpunkt nicht berechnet werden, würde der zuletzt berechnete Wert als Randbedingung für die Nachbarbereiche festgehalten.

Literatur

- [1] Bechler, K.H., Schumann, U.:  
REMAC - Ein REGENT-Subsystem zur Berechnung inkompressibler Strömungen mit freien Oberflächen und MHD-Kräften, 1977, unveröffentlicht
- [2] Katz, F., Krieg, R., Ludwig, A., Schlechtendahl, E.G., Stölting, K.:  
2D Fluid Flow in the Downcomer and Dynamic Response of the Core Barrel during PWR blowdown. SMIRT Conf (4.), Aug. 1977
- [3] Amsden, A., Harlow, H.:  
KACHINA: an Eulerian Computer Program for Multifield Fluid Flows, LA 5680, 1974
- [4] Harlow, H., Amsden, A.:  
Numerical Calculation of Multiphase Fluid Flow, J.Comp.Physics 17, No. 1, Jan. 75, pp. 19 - 52
- [5] Schlechtendahl, E.G.:  
Gleichungen für ein eindimensionales Fluidodynamikmodell im Code FLUSTO und Spezifikation des Fluidodynamikcodes FLUSTO. 1975, unveröffentlicht
- [6] Schlechtendahl, E.G., Krieg, R., Schumann, U., Hirt, C.W.:  
Analyse der fluid-strukturdynamischen Wechselwirkung von Reaktordruckbehältereinbauten beim Blowdown. Jahreskolloquium 1977 des PNS, Karlsruhe, KfK 2570, 1977
- [7] Schmidt, E.:  
VDI-Wasserdampfatafeln, Springer-Verlag Berlin/Göttingen/Heidelberg, 6. Aufl., 1963
- [8] Freuschle, D.:  
Programm zur Bestimmung von Temperatur, Druck und Dampfgehalt für Wasser in allen drei Phasen bei Eingabe der inneren Energie und der Dichte, Studienarbeit, Uni. Karlsruhe, 1977
- [9] Bechler, K.H., Enderle, G.:  
H2OPC2 - Ein Programm zur Bestimmung der Schallgeschwindigkeit in Wasser, 1978, unveröffentlicht



- [10\_] Harlow, F.H., Amsden, A.A.:  
Fluid Dynamics, A LASL Monograph,  
LA-4700, June 1971
- [11\_] Mössinger, H.:  
Assessment of a drift-flux approximation for a strongly  
transient two-phase flow.  
Specialists' Meeting on transient two-phase flow, Paris, 1978
- [12\_] Schumann, U.:  
Linear Stability of Finite Difference Equations for Three-  
Dimensional Flow Problems.  
J.Comp. Physics 18, No. 4, 1975, pp. 465-470
- [13\_] Schlechtendahl, E.G.:  
Überlegungen zur Datenstruktur dreidimensionaler Modelle  
im CODE FLUST unter topologischen und geometrischen Ge-  
sichtspunkten, 1976, unveröffentlicht
- [14\_] Bechler, K.H.:  
FLUSTPLOT, ein Programm zur Auswertung der Ergebnisse des  
Fluidynamik-Codes FLUST, 1978, unveröffentlicht
- [15\_] Schlechtendahl, E.G.:  
Grundzüge des integrierten CAD-Systems REGENT.  
Angewandte Informatik (1976) 11, S.490-496
- [16\_] Enderle, G.:  
Problemorientierte Sprachen im REGENT-System,  
Angewandte Informatik, (1976) 12, S.543-549
- [17\_] Leinemann, K.:  
Dynamische Datenstrukturen des integrierten CAD-Systems  
REGENT. Angewandte Informatik, (1977) 1, S.26-32
- [18\_] Schlechtendahl, E.G., Leinemann, K., Enderle, G., Schuster, R.:  
Das Programmsystem REGENT im praktischen Einsatz. In:  
Gnatz, R., Samelson, K. (Hrsg.):  
Methoden der Informatik für Rechnergestütztes Entwerfen  
und Konstruieren, GI-Fachtagung, München, 19.-21.Okt. 1977.  
(Informatik-Fachberichte, Bd. 11, S. 287-301)

- [ 19 ] Schlechtendahl, E.G., Bechler, K.H., Enderle, G., Leinemann, K.,  
Olbrich, W.: REGENT-Handbuch, KfK 2666 und KfK-CAD71, 1978
- [ 20 ] Mösinger, H., Schumann, U.:  
Berechnung der Anfangstemperaturverteilung im HDR für zwei  
vorgegebene Austrittstemperaturen zur Vorausberechnung von  
Blowdown-Versuchen, 1976, unveröffentlicht
- [ 21 ] Schumann, U.:  
Effektive Berechnung dreidimensionaler Fluid-Struktur-Wirkung  
beim Kühlmittelverluststörfall eines Druckwasserreaktors - FLUX.  
KfK 2645 (1978)
- [ 22 ] Bechler, K.H.:  
Effektives Arbeiten in REGENT, 1978, unveröffentlicht
- [ 23 ] Enderle, G.:  
Definition, Übersetzung und Anwendung benutzerorientierter Sprachen  
als PL/1-Erweiterung in dem System für das rechnerunterstützte  
Entwickeln und Konstruieren REGENT, KfK 2204, 1975 und Diss. Uni-  
versität Karlsruhe, 1975

Anhang A1

Einzel Schritte der Lösung

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe	
SETUP	1				$\Delta x_i, \Delta y_i$ $\eta, \omega$ $\alpha_0, \beta_0$ $\delta_1, \delta_2$ $\epsilon, \Delta t$ $s_i^j, e_i^j$ cycle, t  Neustart: $(su)_{i+\frac{1}{2}}^j$ $(sv)_i^{j+\frac{1}{2}}$  Restart: $u_{i+\frac{1}{2}}^j$ $v_i^{j+\frac{1}{2}}$	
	Neustart					
	2	$u_{i+\frac{1}{2}}^j = \frac{(su)_{i+\frac{1}{2}}^j (\Delta x_i + \Delta x_{i+1})}{\Delta x_{i+1} s_i^j + \Delta x_i s_{i+1}^j}$	$(su)_{i+\frac{1}{2}}^j$ $s_i^j$	1  1	$u_{i+\frac{1}{2}}^j$	
	3	$v_i^{j+\frac{1}{2}} = \frac{(sv)_i^{j+\frac{1}{2}} (\Delta y_i + \Delta y_{i+1})}{\Delta y_{i+1} s_i^j + \Delta y_i s_{i+1}^{j+1}}$	$(sv)_i^{j+\frac{1}{2}}$ $s_i^j$	1  1	$v_i^{j+\frac{1}{2}}$	
	Restart					
	2	$(su)_{i+\frac{1}{2}}^j = \frac{u_{i+\frac{1}{2}}^j (\Delta x_{i+1} s_i^j + \Delta x_i s_{i+1}^j)}{\Delta x_i + \Delta x_{i+1}}$	$u_{i+\frac{1}{2}}^j$ $s_i^j$	1  1	$(su)_{i+\frac{1}{2}}^j$	
	3	$(sv)_i^{j+\frac{1}{2}} = \frac{v_i^{j+\frac{1}{2}} (\Delta y_{i+1} s_i^j + \Delta y_i s_{i+1}^{j+1})}{\Delta y_i + \Delta y_{i+1}}$	$v_i^{j+\frac{1}{2}}$ $s_i^j$	1  1	$(sv)_i^{j+\frac{1}{2}}$	
	4	$p_i^j = p(s_i^j, e_i^j)$	$s_i^j$ $e_i^j$	1  1	$p_i^j$	

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe
EXPLIZITE RECHNUNG	5	$\xi_u = (1-\alpha_0) \frac{\Delta X_{i+1} + \beta_0 \Delta t u_{i+1/2}^i}{\Delta X_i + \Delta X_{i+1}} + \frac{\alpha_0}{2} \left\{ \text{sign}(u_{i+1/2}^i) + 1 \right\} \quad (8)$	$\alpha_0$ $\beta_0$ $\Delta X_i$ $\Delta t$ $u_{i+1/2}^i$	1 1 1 1,41,75 2,67	$\xi_u$
	6	$\langle su \rangle_{i+1/2}^j = u_{i+1/2}^i \left\{ \xi_u s_i^j + (1-\xi_u) s_{i+1}^j \right\}$ <p style="text-align: center;">nach (7)</p>	$u_{i+1/2}^i$ $s_i^j$ $\xi_u$	2,67 1,63 5	$\langle su \rangle_{i+1/2}^j$
	7	$\xi_v = (1-\alpha_0) \frac{\Delta y_{j+1} + \beta_0 \Delta t v_{i+1/2}^{j+1/2}}{\Delta y_j + \Delta y_{j+1}} + \frac{\alpha_0}{2} \left\{ \text{sign}(v_{i+1/2}^{j+1/2}) + 1 \right\} \quad \text{nach (8)}$	$\alpha_0$ $\beta_0$ $\Delta y_j$ $\Delta t$ $v_{i+1/2}^{j+1/2}$	1 1 1 1,41,75 3,68	$\xi_v$
	8	$\langle sv \rangle_i^{j+1/2} = v_i^{j+1/2} \left\{ \xi_v s_i^j + (1-\xi_v) s_{i+1}^j \right\}$ <p style="text-align: center;">nach (7)</p>	$v_i^{j+1/2}$ $s_i^j$ $\xi_v$	3,68 1,63 7	$\langle sv \rangle_i^{j+1/2}$
	9	$v_{i+1/2}^{j+1/2} = \frac{\Delta X_{i+1} v_i^{j+1/2} + \Delta X_i v_{i+1}^{j+1/2}}{\Delta X_i + \Delta X_{i+1}} \quad (35)$	$\Delta X_i$ $v_i^{j+1/2}$	1 3,68	$v_{i+1/2}^{j+1/2}$
	10	$\xi_2 = (1-\alpha_0) \frac{\Delta y_{j+1} + \beta_0 \Delta t v_{i+1/2}^{j+1/2}}{\Delta y_{j+1} + \Delta y_j} + \frac{\alpha_0}{2} \left\{ \text{sign}(v_{i+1/2}^{j+1/2}) + 1 \right\} \quad (34)$	$\alpha_0$ $\beta_0$ $\Delta t$ $\Delta y_j$ $v_{i+1/2}^{j+1/2}$	1 1 1,41,75 1 9	$\xi_2$
	11	$\langle svu \rangle_{i+1/2}^{j+1/2} = v_{i+1/2}^{j+1/2} \left\{ \xi_2 (su)_{i+1/2}^j + (1-\xi_2) (su)_{i+1/2}^{j+1} \right\} \quad (33)$	$\xi_2$ $v_{i+1/2}^{j+1/2}$ $(su)_{i+1/2}^j$	10 9 1,65	$\langle svu \rangle_{i+1/2}^{j+1/2}$

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe
EXPLIZIT	12	$u_{i+1/2}^{j+1/2} = \frac{\Delta y_{j+1} u_{i+1/2}^j + \Delta y_j u_{i+1/2}^{j+1}}{\Delta y_j + \Delta y_{j+1}} \quad (38)$	$\Delta y_j$ $u_{i+1/2}^j$	1 2,67	$u_{i+1/2}^{j+1/2}$
	13	$\xi_3 = (1-\alpha_0) \frac{\Delta x_{i+1} + \beta_0 \Delta t u_{i+1/2}^{j+1/2}}{\Delta x_i + \Delta x_{i+1}} + \frac{\alpha_0}{2} \left\{ \text{sign}(u_{i+1/2}^{j+1/2}) + 1 \right\} \quad (37)$	$\alpha_0$ $\beta_0$ $\Delta t$ $u_{i+1/2}^{j+1/2}$ $\Delta x_i$	1 1 1,41,75 12 1	$\xi_3$
	14	$\langle suv \rangle_{i+1/2}^{j+1/2} = u_{i+1/2}^{j+1/2} \left\{ \xi_3 (sv)_i^{j+1/2} + (1-\xi_3) (sv)_{i+1}^{j+1/2} \right\} \quad (36)$	$u_{i+1/2}^{j+1/2}$ $\xi_3$ $(sv)_i^{j+1/2}$	12 13 1,66	$\langle suv \rangle_{i+1/2}^{j+1/2}$
	15	$u_i^j = \frac{1}{2} (u_{i+1/2}^j + u_{i-1/2}^j) \quad (32)$	$u_{i+1/2}^j$	2,71	$u_i^j$
	16	$\xi_1 = (1-\alpha_0) \frac{\Delta x_i + \beta_0 \Delta t u_i^j}{2 \Delta x_i} + \frac{\alpha_0}{2} \left\{ \text{sign}(u_i^j) + 1 \right\} \quad (31)$	$\alpha_0$ $\beta_0$ $\Delta t$ $\Delta x_i$ $u_i^j$	1 1 1,41,75 1 15	$\xi_1$
	17	$\langle su^2 \rangle_i^j = u_i^j \left\{ \xi_1 (su)_{i-1/2}^j + (1-\xi_1) (su)_{i+1/2}^j \right\} \quad (30)$	$\xi_1$ $u_i^j$ $(su)_{i+1/2}^j$	16 15 1,65	$\langle su^2 \rangle_i^j$
	18	$v_i^j = \frac{1}{2} (v_i^{j+1/2} + v_i^{j-1/2}) \quad (41)$	$v_i^{j+1/2}$	3,68	$v_i^j$
	19	$\xi_4 = (1-\alpha_0) \frac{\Delta y_i + \beta_0 \Delta t v_i^j}{2 \Delta y_i} + \frac{\alpha_0}{2} \left\{ \text{sign}(v_i^j) + 1 \right\} \quad (40)$	$\alpha_0$ $\beta_0$ $\Delta y_i$ $\Delta t$ $v_i^j$	1 1 1 41,75 18	$\xi_4$

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe
EXPLIZIT	20	$\langle sv^2 \rangle_i^j = v_i^j \left\{ \xi_4 (sv)_i^{j-1/2} + (1 - \xi_4) (sv)_i^{j+1/2} \right\} \quad (39)$	$v_i^j$ $\xi_4$ $(sv)_i^{j+1/2}$	18 19 1,66	$\langle sv^2 \rangle_i^j$
	21	$\tilde{g}_i^j = {}^n g_i^j - \frac{\Delta t}{\Delta x_i} ({}^n \langle gu \rangle_{i+1/2}^j - {}^n \langle gu \rangle_{i-1/2}^j) - \frac{\Delta t}{\Delta y_j} ({}^n \langle sv \rangle_i^{j+1/2} - {}^n \langle sv \rangle_i^{j-1/2}) \quad (20)$	$g_i^j$ $\Delta t$ $\Delta x_i$ $\Delta y_j$ $\langle gu \rangle_{i+1/2}^j$ $\langle sv \rangle_i^{j+1/2}$	1,63 1,41,75 1 1 7 8	$\tilde{g}_i^j$
	22	$\tilde{p}_i^j = p(\tilde{g}_i^j, e_i^j)$	$\tilde{g}_i^j$ $e_i^j$	21 1,73	$\tilde{p}_i^j$
	23	$\overline{gu}_{i+1/2}^j = \text{siehe (22)}$	$(gu)_{i+1/2}^j$ $\Delta x_i \Delta y_j$ $\Delta t$ $\eta$ $\langle gu^2 \rangle_i^j$ $\langle svu \rangle_{i+1/2}^{j+1/2}$ $w_{i+1/2}$	1,65 1 1,41,75 1 17 11 2,67	$\overline{gu}_{i+1/2}^j$
	24	$\overline{sv}_i^{j+1/2} = \text{siehe (23)}$	$(sv)_i^{j+1/2}$ $\Delta x_i$ $\Delta y_j$ $\Delta t$ $\eta$ $\langle svu \rangle_{i+1/2}^{j+1/2}$ $\langle sv^2 \rangle_i^j$ $v_i^{j+1/2}$	1,66 1 1 1,41,75 1 14 20 3,68	$\overline{sv}_i^{j+1/2}$
25	$\tilde{g}_{i+1/2}^j = \frac{\Delta x_{i+1} \tilde{g}_i^j + \Delta x_i \tilde{g}_{i+1}^j}{\Delta x_i + \Delta x_{i+1}} \quad (28)$	$\Delta x_i$ $\tilde{g}_i^j$	1 21	$\tilde{g}_{i+1/2}^j$	

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe
EXPLIZIT	26	$\tilde{u}_{i+1/2}^i = \frac{1}{\tilde{S}_{i+1/2}^i} \left\{ \overline{su}_{i+1/2}^i - 2 \Delta t \frac{\tilde{p}_{i+1}^i - \tilde{p}_i^i}{\Delta x_i + \Delta x_{i+1}} \right\} \quad (26)$	$\tilde{S}_{i+1/2}^i$ $\Delta t$ $\overline{su}_{i+1/2}^i$ $\Delta x_i$ $\tilde{p}_i^i$	25 1,41,75 23 1 22	$\tilde{u}_{i+1/2}^i$
	27	$\tilde{S}_i^{j+1/2} = \frac{\Delta y_{j+1} \tilde{S}_i^j + \Delta y_j \tilde{S}_i^{j+1}}{\Delta y_j + \Delta y_{j+1}} \quad (29)$	$\Delta y_j$ $\tilde{S}_i^j$	1 21	$\tilde{S}_i^{j+1/2}$
	28	$\tilde{v}_i^{j+1/2} = \frac{1}{\tilde{S}_i^{j+1/2}} \left\{ \overline{sv}_i^{j+1/2} - 2 \Delta t \frac{\tilde{p}_i^{j+1} - \tilde{p}_i^j}{\Delta y_j + \Delta y_{j+1}} \right\} \quad (27)$	$\tilde{S}_i^{j+1/2}$ $\Delta t$ $\overline{sv}_i^{j+1/2}$ $\Delta y_j$ $\tilde{p}_i^j$	27 1,41,75 24 1 22	$\tilde{v}_i^{j+1/2}$
	29	$\mathcal{F}_5 = (1 - \alpha_0) \frac{\Delta x_{i+1} + \beta_0 \Delta t \tilde{u}_{i+1/2}^i}{\Delta x_i + \Delta x_{i+1}} + \frac{\alpha_0}{2} \left\{ \text{sign}(\tilde{u}_{i+1/2}^i) + 1 \right\} \quad (8)$	$\alpha_0$ $\beta_0$ $\Delta x_i$ $\Delta t$ $\tilde{u}_{i+1/2}^i$	1 1 1 1,41,75 26	$\mathcal{F}_5$
	30	$\langle \tilde{S} \tilde{u} \rangle_{i+1/2}^i = \tilde{u}_{i+1/2}^i \left\{ \mathcal{F}_5 \tilde{S}_i^j + (1 - \mathcal{F}_5) \tilde{S}_{i+1}^j \right\}$ nach (7)	$\tilde{u}_{i+1/2}^i$ $\mathcal{F}_5$ $\tilde{S}_i^j$	26 29 21	$\langle \tilde{S} \tilde{u} \rangle_{i+1/2}^i$
	31	$\mathcal{F}_6 = (1 - \alpha_0) \frac{\Delta y_{j+1} + \beta_0 \Delta t \tilde{v}_i^{j+1/2}}{\Delta y_j + \Delta y_{j+1}} + \frac{\alpha_0}{2} \left\{ \text{sign}(\tilde{v}_i^{j+1/2}) + 1 \right\}$ nach (8)	$\alpha_0$ $\beta_0$ $\Delta y_j$ $\Delta t$ $\tilde{v}_i^{j+1/2}$	1 1 1 1,41,75 28	$\mathcal{F}_6$
	32	$\langle \tilde{S} \tilde{v} \rangle_i^{j+1/2} = \tilde{v}_i^{j+1/2} \left\{ \mathcal{F}_6 \tilde{S}_i^j + (1 - \mathcal{F}_6) \tilde{S}_i^{j+1} \right\}$ nach (7)	$\tilde{v}_i^{j+1/2}$ $\mathcal{F}_6$ $\tilde{S}_i^j$	28 31 21	$\langle \tilde{S} \tilde{v} \rangle_i^{j+1/2}$

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe
EXPLIZIT	33	$(\tilde{\sigma} \tilde{u})_{i+1/2}^j = \tilde{\sigma}_{i+1/2}^j \tilde{u}_{i+1/2}^j$	$\tilde{\sigma}_{i+1/2}^j$ $\tilde{u}_{i+1/2}^j$	25 26	$(\tilde{\sigma} \tilde{u})_{i+1/2}^j$
	34	$(\tilde{\sigma} \tilde{v})_i^{j+1/2} = \tilde{\sigma}_i^{j+1/2} \tilde{v}_i^{j+1/2}$	$\tilde{\sigma}_i^{j+1/2}$ $\tilde{v}_i^{j+1/2}$	27 28	$(\tilde{\sigma} \tilde{v})_i^{j+1/2}$
	35	$(\frac{1}{c^2})_i^j = 1/c^2 (\tilde{\sigma}_i^j, e_i^j)$	$\tilde{\sigma}_i^j$ $e_i^j$	21 73	$(\frac{1}{c^2})_i^j$
	36	$\beta_i^j = \left\{ \frac{1}{\Delta t} (\frac{1}{c^2})_i^j + \frac{2\Delta t}{\Delta x_i} \left( \frac{1}{\Delta x_i + \Delta x_{i+1}} + \frac{1}{\Delta x_{i-1} + \Delta x_i} \right) + \frac{2\Delta t}{\Delta y_i} \left( \frac{1}{\Delta y_i + \Delta y_{i+1}} + \frac{1}{\Delta y_{i-1} + \Delta y_i} \right) \right\}^{-1}$	$\Delta x_i$ $\Delta y_i$ $\Delta t$ $(\frac{1}{c^2})_i^j$	1 1 1,41,75 35	$\beta_i^j$
	37	$R = \epsilon \cdot \frac{(\tilde{\sigma}_i^j)_{\max}}{\Delta t}$	$\tilde{\sigma}_i^j$ $\epsilon$ $\Delta t$	21 1 1,41,75	R
	38	iter = 0			iter
	TIME	39	$\text{timestephalf} = \Delta t \cdot \max_{i,j} \left\{ \frac{\max( \tilde{u}_{i-\frac{1}{2}}^j ,  \tilde{u}_{i+\frac{1}{2}}^j )}{\Delta x_i} + \frac{\max( \tilde{v}_i^{j-\frac{1}{2}} ,  \tilde{v}_i^{j+\frac{1}{2}} )}{\Delta y_i} + 2\eta \left( \frac{1}{\Delta x_i^2} + \frac{1}{\Delta y_i^2} \right) \right\} > \gamma_1$ (44)	$\tilde{u}_{i+\frac{1}{2}}^j$ $\tilde{v}_i^{j+\frac{1}{2}}$ $\Delta x_i, \Delta y_i$ $\Delta t$ $\gamma_1$	26 28 1 1,41,75 1
40		wenn $\neg$ timestephalf gehe zum Schritt 43	time-step-half	39	
41		$\Delta t = \Delta t / 2$	$\Delta t$	1,41,75	$\Delta t$
42		neuer Versuch, gehe zum Schritt 5			



Phase	Schritt Nr.	Gleichung	Eingabe	aus Schritt	Ausgabe
TIME	43	$\text{timestepdouble} = \Delta t \cdot \max_{i,j} \left\{ \frac{\max( u_{i-\frac{1}{2}}^j ,  u_{i+\frac{1}{2}}^j )}{\Delta x_i} + \frac{\max( v_{i-\frac{1}{2}}^j ,  v_{i+\frac{1}{2}}^j )}{\Delta y_j} + 2\eta \left( \frac{1}{\Delta x_i^2} + \frac{1}{\Delta y_j^2} \right) \right\} < \gamma_2$	$\tilde{u}_{i+\frac{1}{2}}^j$ $\tilde{v}_{i+\frac{1}{2}}^j$ $\Delta x_i, \Delta y_j$ $\Delta t$ $\gamma_2$	26 28 1 1,41,75 1	time-step-double
	44	$\text{iter} = \text{iter} + 1$	iter	38,44	iter
ITERATION	45	$D_i^j = \frac{\tilde{g}_i^j - n s_i^j}{\Delta t} + \frac{\langle \tilde{g} \tilde{u} \rangle_{i+1/2}^j - \langle \tilde{g} \tilde{u} \rangle_{i-1/2}^j}{\Delta x_i} + \frac{\langle \tilde{g} \tilde{v} \rangle_{i+1/2}^j - \langle \tilde{g} \tilde{v} \rangle_{i-1/2}^j}{\Delta y_j} \quad (42)$	$\tilde{g}_i^j$ $s_i^j$ $\Delta t$ $\Delta x_i$ $\Delta y_j$ $\langle \tilde{g} \tilde{u} \rangle$ $\langle \tilde{g} \tilde{v} \rangle$	21,51 1,63 1,41,75 1 1 30,59 32,61	$D_i^j$
	46	$\delta \tilde{p}_i^j = -\omega \beta_i^j D_i^j \quad (59)$	$\omega_i^j$ $\beta_i^j$ $D_i^j$	1 36 45	$\delta \tilde{p}_i^j$
	47	$\text{konv}_i^j =  D_i^j  \leq R \ \& \  \delta \tilde{p}_i^j  \leq \epsilon \cdot \tilde{p}_i^j \quad (72)$	$D_i^j$ $R$ $\delta \tilde{p}_i^j$ $\tilde{p}_i^j$	45 37 46 22,49	konv <sub>i</sub> <sup>j</sup>
	48	$\text{ende} = \left( \&_{i,j} \text{konv}_i^j \right) \mid (\text{iter} \geq \text{maxiter})$ <p>nach (72)</p>	iter maxiter konv <sub>i</sub> <sup>j</sup>	44 1 47	ende
	49	$\tilde{p}_i^j = \tilde{p}_i^j + \delta \tilde{p}_i^j \quad (60)$	$\tilde{p}_i^j$ $\delta \tilde{p}_i^j$	22,49 46	$\tilde{p}_i^j$
	50	$\delta \tilde{g}_i^j = \left( \frac{1}{c^2} \right)_i^j \delta \tilde{p}_i^j \quad (61)$	$\left( \frac{1}{c^2} \right)_i^j$ $\delta \tilde{p}_i^j$	35 46	$\delta \tilde{g}_i^j$
	51	$\tilde{g}_i^j = \tilde{g}_i^j + \delta \tilde{g}_i^j \quad (62)$	$\tilde{g}_i^j$ $\delta \tilde{g}_i^j$	21,51 50	$\tilde{g}_i^j$

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe
ITERATION	52	$\delta(\tilde{g}\tilde{u})_{i+1/2}^j = \frac{2\Delta t}{\Delta x_i + \Delta x_{i+1}} (\delta\tilde{p}_i^j - \delta\tilde{p}_{i+1}^j) \quad (63)$	$\Delta x_i$ $\Delta t$ $\delta\tilde{p}_i^j$	1 1,41,75 46	$\delta(\tilde{g}\tilde{u})_{i+1/2}^j$
	53	$(\tilde{g}\tilde{u})_{i+1/2}^j = (\tilde{g}\tilde{u})_{i+1/2}^{j-1} + \delta(\tilde{g}\tilde{u})_{i+1/2}^j \quad (64)$	$(\tilde{g}\tilde{u})$ $\delta(\tilde{g}\tilde{u})$	33,53 52	$(\tilde{g}\tilde{u})_{i+1/2}^j$
	54	$\tilde{u}_{i+1/2}^j = \frac{(\tilde{g}\tilde{u})_{i+1/2}^j (\Delta x_i + \Delta x_{i+1})}{\Delta x_i \tilde{g}_{i+1}^j + \Delta x_{i+1} \tilde{g}_i^j} \quad (68)$	$(\tilde{g}\tilde{u})$ $\Delta x_i$ $\tilde{g}_i^j$	53 1 51	$\tilde{u}_{i+1/2}^j$
	55	$\delta(\tilde{g}\tilde{v})_i^{j+1/2} = \frac{2\Delta t}{\Delta y_j + \Delta y_{j+1}} (\delta\tilde{p}_i^j - \delta\tilde{p}_i^{j+1}) \quad (65)$	$\Delta y_j$ $\Delta t$ $\delta\tilde{p}_i^j$	1 1,41,75 46	$\delta(\tilde{g}\tilde{v})_i^{j+1/2}$
	56	$(\tilde{g}\tilde{v})_i^{j+1/2} = (\tilde{g}\tilde{v})_i^{j+1/2} + \delta(\tilde{g}\tilde{v})_i^{j+1/2} \quad (66)$	$(\tilde{g}\tilde{v})$ $\delta(\tilde{g}\tilde{v})$	34,56 55	$(\tilde{g}\tilde{v})_i^{j+1/2}$
	57	$\tilde{v}_i^{j+1/2} = \frac{(\tilde{g}\tilde{v})_i^{j+1/2} (\Delta y_j + \Delta y_{j+1})}{\Delta y_j \tilde{g}_i^{j+1} + \Delta y_{j+1} \tilde{g}_i^j} \quad (70)$	$(\tilde{g}\tilde{v})$ $\Delta y_j$ $\tilde{g}_i^j$	56 1 51	$\tilde{v}_i^{j+1/2}$
	58	$\mathcal{F}_7 = \text{wie Schritt 29} \quad (8)$	$\alpha_0$ $\beta_0$ $\Delta x_i$ $\Delta y_j$ $\Delta t$ $\tilde{u}_{i+1/2}^j$	1 1 1 1 1,41,75 54	$\mathcal{F}_7$
	59	$\langle \tilde{g}\tilde{u} \rangle_{i+1/2}^j = \text{wie Schritt 30}$  nach (7)	$\tilde{u}_{i+1/2}^j$ $\mathcal{F}_7$ $\tilde{g}_i^j$	54 58 51	$\langle \tilde{g}\tilde{u} \rangle_{i+1/2}^j$
	60	$\mathcal{F}_8 = \text{wie Schritt 31}$  nach (8)	$\alpha_0$ $\beta_0$ $\Delta y_j$ $\Delta t$ $\tilde{v}_i^{j+1/2}$	1 1 1 1,41,75 57	$\mathcal{F}_8$

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe
ITERATION	61	$\langle \tilde{g} \tilde{v} \rangle_i^{j+1/2} = \text{wie Schritt 32}$ nach (7)	$\tilde{v}_i^{j+1/2}$ $\tilde{g}_i^j$	57 60 51	$\langle \tilde{g} \tilde{v} \rangle_i^{j+1/2}$
	6	wenn → ende wiederhole die Iteration ab Schritt 42	ende	48	
ACCEPT	63	$s_i^j = \tilde{s}_i^j$ (73)	$\tilde{s}_i^j$	51	$s_i^j$
	64	$p_i^j = \tilde{p}_i^j$ (74)	$\tilde{p}_i^j$	50	$p_i^j$
	65	$(su)_{i+1/2}^j = (\tilde{s} \tilde{u})_{i+1/2}^j$ (75)	$(\tilde{s} \tilde{u})$	53	$(su)_{i+1/2}^j$
	66	$(sv)_i^{j+1/2} = (\tilde{s} \tilde{v})_i^{j+1/2}$ (76)	$(\tilde{s} \tilde{v})$	56	$(sv)_i^{j+1/2}$
	67	$u_{i+1/2}^j = \tilde{u}_{i+1/2}^j$ (77)	$\tilde{u}_{i+1/2}^j$	54	$u_{i+1/2}^j$
	68	$v_i^{j+1/2} = \tilde{v}_i^{j+1/2}$ (78)	$\tilde{v}_i^{j+1/2}$	57	$v_i^{j+1/2}$
ENERGIE	69	$\xi_g = (1-\alpha_0) \frac{\Delta X_{i+1} + \beta_0 \Delta t u_{i+1/2}^j}{\Delta X_i + \Delta X_{i+1}} + \frac{\alpha_0}{2} \{ \text{sign}(u_{i+1/2}^j) + 1 \}$ (8)	$\alpha_0$ $\beta_0$ $\Delta X_i$ $\Delta t$ $u_{i+1/2}^j$	1 1 1 1,41,75 67	$\xi_g$
	70	$\langle eu \rangle_{i+1/2}^j = u_{i+1/2}^j \{ \xi_g e_i^j + (1-\xi_g) e_{i+1}^j \}$ nach (7)	$e_i^j$ $u_{i+1/2}^j$ $\xi_g$	1,73 67 69	$\langle eu \rangle_{i+1/2}^j$
	71	$\xi_{10} = (1-\alpha_0) \frac{\Delta y_{j+1} + \beta_0 \Delta t v_{j+1/2}^j}{\Delta y_j + \Delta y_{j+1}} + \frac{\alpha_0}{2} \{ \text{sign}(v_{j+1/2}^j) + 1 \}$ nach (8)	$\alpha_0$ $\beta_0$ $\Delta y_j$ $\Delta t$ $v_{j+1/2}^j$	1 1 1 1,41,75 68	$\xi_{10}$

Phase	Schritt Nr.	Gleichung	Ein-gabe	aus Schritt	Aus-gabe
ENERGIE	72	$\langle ev \rangle_i^{j+1/2} = v_i^{j+1/2} \left\{ \beta_{10} e_i^j + (1 - \beta_{10}) e_i^{j+1} \right\}$ <p style="text-align: center;">nach (7)</p>	$v_i^{j+1/2}$ $\beta_{10}$ $e_i^j$	68 71 1, 73	$\langle ev \rangle_i^{j+1/2}$
	73	$e_i^j =$ siehe Gl. (10)	$\Delta x_i$ $\Delta y_j$ $\Delta t$ $u_{i+1/2}^j$ $v_i^{j+1/2}$ $\langle eu \rangle$ $\langle ev \rangle$ $e_i^j$ $p_i^j$ $s_i^j$	1 1 1,41,75 67 68 70 72 73 64 63	$e_i^j$
	74	$cycle = cycle + 1$	cycle	1,74	cycle
	76	$t = t + \Delta t$	$t$ $\Delta t$	1,76 41,75	$t$
	75	wenn timestepdouble $\Delta t = 2 \Delta t$	timestep-double $\Delta t$	43 41,75	$\Delta t$

Anhang A2

Programmstruktur

Für jede Routine des FLUST-Programms werden im folgenden Name, Parameterliste und Eingabe/Ablauf/Ausgabe in Form von HIPO-Diagrammen aufgeführt (HIPO=Hierarchical Input Process Output). Zusätzlich sind die Rechenschritte aus Anhang A1 angegeben.

Steuermodule

CONTROL			
Keine Parameterliste, gerufen aus Anwenderprogramm beim SOLVE-Statement			
Eingabe	Ablauf	Schritt	Ausgabe
Modell	SETUP		Druck-Liste
Konstanten	WHILE ( $\neg$ stop)	77	Save-Datei
Funktionen	IF print THEN PRINT		Fehlermeldungen
Anfangswerte	IF save THEN SAVE		
Randwerte	EXPLIC	5-38	
	TIME	39-43	
	IF $\neg$ timestephalf		
	ITERAT	44-62	
	ACCEPT	63-68	
	ENERGY	69-73	
	cycle+1	74	
	t=t+ $\Delta$ t	76	
	IF timestepdouble		
	THEN $\Delta$ t=2 $\cdot$ $\Delta$ t	75	
	ENDE		

SETUP			
Keine Parameterliste			
Neustart:			
Eingabe	Ablauf	Schritt	Ausgabe
Anfangs- werte (Fluid) Rand- werte $\xi_u, \xi_v,$ $\xi, e$	Allokriere und initiali- siere ENTRIES-Matrix I=1... IFLUIDS F→R $\xi_u, \xi_v, \xi, e$ I=1... IBOUNDS R.C. $\xi, e, \xi_u, \xi_v$ AAR1(I) I=1... IFLUIDS R→F u AAF1(I) F→R $v, u_{in}, v_{in}$ I=1... IBOUNDS B.C. $v, u_{in}, v_{in}, <\xi_{uv}>_{in}$ $<\xi_{vu}>_{in}, <\xi_{uv}>, <\xi_{vu}>$ R→N $u, v, u_{in}, v_{in},$ $\xi_u, \xi_v, e$	               2-4      2-4	u v p
Restart:			
$u, v,$ $\xi, e$	Allokriere und initiali- siere ENTRIES-Matrix I=1... IFLUIDS AAF1(I) F→R $\xi_u, \xi_v, \xi, e,$ $u, v, u_{in}, v_{in}$ I=1... IBOUNDS B.C. $\xi, e, \xi_v, u$ AAR1(I) B.C. $v, u_{in}, v_{in}, <\xi_{uv}>_{in}$ $<\xi_{vu}>_{in}, <\xi_{uv}>, <\xi_{vu}>$ R→N $u, v, u_{in}, v_{in}, \xi_u$ $\xi_v, e$ I=1... IFLUIDS R→F $\xi_u$	              2-4      2-4	$\xi_u$  $\xi_v$ p

EXPLIC			
Keine Parameterliste			
Eingabe	Ablauf	Schritt	Ausgabe
Modell	I=1... ISEAMS		
u	AAN3(I)	9-14	$\tilde{u}$
v	I=1... IBOUNDS		$\tilde{v}$
$\mathcal{E}u$	AAR2(I)	5-8	$(\tilde{\mathcal{E}}u)$
$\mathcal{E}v$	N→R $\langle \mathcal{E}uv \rangle, \langle \mathcal{E}vu \rangle$		$(\tilde{\mathcal{E}}v)$
p	AAR3(I)	9-14	$\tilde{p}$
e	I=1... FLUIDS		$\tilde{\mathcal{E}}$
$\mathcal{E}$	R→F $\langle \mathcal{E}uv \rangle, \langle \mathcal{E}vu \rangle, \mathcal{E}u$		B
	AAF2(I)	5-8	R
	AAF3(I)	9-22	$\frac{1}{c^2}$
	F→R $\tilde{p}, \tilde{\mathcal{E}}, \langle \mathcal{E}u^2 \rangle,$ $\langle \mathcal{E}v^2 \rangle, \langle \mathcal{E}uv \rangle_{in}, \langle \mathcal{E}vu \rangle_{in}$		
	I=1... IBOUNDS		
	B.C. $\langle \mathcal{E}u^2 \rangle, \langle \mathcal{E}v^2 \rangle, \tilde{u}, \tilde{\mathcal{E}}$ $, \tilde{p}$		
	R→N $\tilde{p}, \tilde{\mathcal{E}}, \langle \mathcal{E}u^2 \rangle, \langle \mathcal{E}v^2 \rangle,$ $\langle \mathcal{E}uv \rangle_{in}, \langle \mathcal{E}vu \rangle_{in}, \tilde{u}_{B.C.}$		
	I=1... ISEAMS		
	AAN4(I)	23-28	
	AAN6(I)	35-36	
	I=1... IBOUNDS		
	N→R $\tilde{u}, \beta$		
	AAR4(I)	23-28	
	AAR5(I)	29-30	
	AAR6(I)	35-36	
	B.C. $(\tilde{\mathcal{E}}\tilde{u})$		
	I=1... IFLUIDS		
	R→F $\tilde{u}, \beta$		
	AAF4(I)	23-28	
	AAF5(I)	29-32	
	AAF6(I)	33-36	
	F→R $(\tilde{\mathcal{E}}\tilde{v})$		

Eingabe	Ablauf	Schritt	Ausgabe
	I=1... IBOUNDS B.C. ( $\tilde{g} \tilde{v}$ ) iter=0 $R = \epsilon \cdot S_{\max} / \Delta t$	38 37	
TIME			
Keine Parameterliste			
Eingabe	Ablauf	Schritt	Ausgabe
$\tilde{u}$ $\tilde{v}$ $\Delta t$ $\gamma_1$ $\gamma_2$	I=1.. IFLUIDS WHILE ( $\neg$ timestephalf) AAFTIM(I)	39-43	timestep- half timestep- double
ITERAT			
Keine Parameterliste			
Eingabe	Ablauf	Schritt	Ausgabe
$\tilde{p}$ $\tilde{S}$ $\tilde{u}$ $\tilde{v}$ $(\tilde{g}\tilde{u})$ $(\tilde{g}\tilde{v})$ $B$ $S$ $e$ iter	WHILE ( $\neg$ ende) iter=iter+1 I=1... IFLUIDS R→F $\langle \tilde{g}\tilde{u} \rangle$ AAF7(I) AAF8(I) AAF9(I) F→R $\langle \tilde{g}\tilde{u} \rangle, \delta\tilde{p}, \tilde{p}$ I=1... IBOUNDS B.C. $\delta\tilde{p}, \tilde{g}, (\tilde{g}\tilde{u})$ AAR8(I) AAR9(I) I=1... IFLUIDS R→F $(\tilde{g}\tilde{u}), \tilde{u}$ F→R $(\tilde{g}\tilde{v}), \tilde{v}$	47  45-51 52-57 58-61  52-57 58-59	$\tilde{p}$ $\tilde{S}$ $\tilde{u}$ $\tilde{v}$ $(\tilde{g}\tilde{u})$ $(\tilde{g}\tilde{v})$



ACCEPT			
Eingabe	Ablauf	Schritt	Ausgabe
$\tilde{p}, \tilde{\xi},$ $\tilde{u}, \tilde{v},$ $(\tilde{\xi u}), (\tilde{\xi v})$	$I=1 \dots$ IFLUIDS AAF10(I) $F \rightarrow R$ $u_{in}$	63-68	$p, \xi,$ $u, v,$ $(\xi u), (\xi v)$
	$I=1 \dots$ IBOUNDS AAR10(I) B.C. $\xi u, \xi v, u, v, \xi,$ $u_{in}, v_{in}$ $R \rightarrow N$ $\xi u, u, u_{in}$	63-68	
ENERGY			
Keine Parameterliste			
Eingabe	Ablauf	Schritt	Ausgabe
$e$ $u$ $v$ $p$ $\xi$	$I=1 \dots$ IBOUNDS AAR11(I)	69-70	$e$
	$I=1 \dots$ IFLUIDS $R \rightarrow F$ $\langle eu \rangle$ AAF11(I) $F \rightarrow R$ $e$ $I=1 \dots$ IBOUNDS B.C. $e$ $R \rightarrow N$ $e$	69-73	
ENDE			
Keine Parameterliste			
	Ablauf		
	Gebe ENTRIES-Matrix frei IF print THEN FPRINT IF save THEN FSAVE		

Übertragungs- und Randbedingungs-Module

AATRFB (NR, ICODE)		Fluid ↔ Rand	
NR:	Nummer des Fluides		
ICODE:	1 = F→R $v$	2 = F→R $u$	
	3 = F→R $\rho u, \rho v$	4 = R→F $\rho u$	
	5 = F→R $\langle \rho v \rangle$	6 = R→F $\langle \rho u \rangle$	
	7 = F→R $\tilde{p}$	8 = F→R $\tilde{\xi}$	
	9 = F→R $\langle \rho uv \rangle_{in}, \langle \rho vu \rangle_{in}$		
	10 = R→F $\tilde{u}, \tilde{v}$	11 = R→F $\langle \tilde{\xi} \tilde{u} \rangle$	
	12 = F→R $(\tilde{\xi} \tilde{v})$	13 = R→F $(\tilde{\xi} \tilde{u})$	
	14 = F→R $\delta \tilde{p}$	15 = R→F $\langle eu \rangle$	
	16 = F→R $e$	16 = F→R $\xi$	
	18 = R→F $\langle \rho uv \rangle$	$\langle \rho vu \rangle$	
	19 = F→R $\langle \rho u^2 \rangle$	20 = F→R $u_{in}, v_{in}$	
	21 = R→F $\beta$	22 = R→F $\tilde{u}$	
	23 = F→R $\tilde{v}$	F→R $u, v$	
	25 = R→F $(\rho u), (\rho v)$		
Eingabe	Ablauf	Schritt	Ausgabe
Fluid- werte Rand- werte	GOTO M(ICODE)  M(ICODE): AAFTRN2 (Parameterliste für zu übertragende Werte)		Rand- werte Fluid- werte
AAFTRN2 (NR, IND, IVAR, IBVAR, ILEN, IBLEN, GET, ICODE)			
NR:	Nummer des Fluids		
IND:	Komponententeil, z.B. Maschenmitte, senkrechte Fläche (s. Anhang A3)		
IVAR:	Position der zu übertragenden Variablen in der Daten- struktur des Komponententeils des Fluids		
IBVAR:	Position in der Datenstruktur des Randes		
ILEN:	Länge des Datenblockes im Fluid (Anzahl Werte)		
IBLEN:	Länge des Datenblockes für den Rand		

Get:	'1'B = hole vom Rand ins Fluid '0'B = speichere vom Fluid auf den Rand
ICODE:	0 = übertrage Variable 1 = übertrage nur Variable auf Grenzfläche senkrecht zum Rand, z.B. $u, \mathcal{S}u$ 2 = übertrage nur Variable auf Grenzfläche parallel zum Rand, z.B. $v, \mathcal{S}v$
AATRBS(NR, CODE) <span style="float: right;">Rand ↔ Naht</span>	
NR:	Nummer des Randes
ICODE:	1 = R→N ( $\mathcal{S}u$ ), $u, u_{in}$ , ( $\mathcal{S}v$ ), $v, v_{in}$ 2 = R→N $e$ 3 = R→N $\tilde{p}, \tilde{\mathcal{S}}, \langle \mathcal{S}uv \rangle, \langle \mathcal{S}vu \rangle, \langle \mathcal{S}uv \rangle_{in}, \langle \mathcal{S}vu \rangle_{in}, \tilde{u}_{B.C.}$ 4 = N→R $\langle \mathcal{S}uv \rangle \langle \mathcal{S}vu \rangle$ 5 = N→R $\tilde{u}$ 6 = N→R $\beta$
Ablauf siehe Programm	
AABCND(NR, ICODE) <span style="float: right;">Randbedingung</span>	
NR:	Nummer des Randes
ICODE:	1 = $\mathcal{S}u$ 2 = $\mathcal{S}v$ 3 = $\mathcal{S}$ 4 = $e$ 5 = $u$ 6 = $v$ 7 = $\tilde{u}$ 8 = $\tilde{v}$ 9 = $\tilde{\mathcal{S}}$ 10 = $(\tilde{\mathcal{S}}\tilde{u})$ 11 = $p$ 12 = $\tilde{p}$ 13 = $u_{in_2}$ 14 = $\langle \mathcal{S}uv \rangle_{in}, \langle \mathcal{S}vu \rangle_{in}$ 15 = $\langle \mathcal{S}u^2 \rangle$ 16 = $(\tilde{\mathcal{S}}\tilde{v})$ 17 = $v_{in}$ 18 = $\delta\tilde{p}$
Die Werte ICODE = 11 bis 18 werden nur wirksam bei Randart = 6 (Zufluß/Abfluß) und 8 (90°-Koppelung von Fluid)	
Ablauf siehe Programm	

Rechenmodule

Eingabe	Ablauf	Schritt	Ausgabe
AAF1 (NR)			
$\mathfrak{g}u, \mathfrak{g}v$ (Neustart) $u, v$ (Restart) $\mathfrak{s}, e$	Für alle Flächen $u, v$ (Neustart) oder $\mathfrak{g}u, \mathfrak{g}v$ (Restart) Für alle Maschen $p$	2-3  4	$u, v$ (Neustart) $\mathfrak{g}u, \mathfrak{g}v$ (Restart) $p$
AAF2 (NR)			
$\mathfrak{s},$ $u, v$	Für alle Flächen $\langle \mathfrak{g}u \rangle, \langle \mathfrak{g}v \rangle$	5-8	$\langle \mathfrak{g}u \rangle$ $\langle \mathfrak{g}v \rangle$
AAF3 (NR)			
$u, v,$ $\mathfrak{s}, e$	Für alle Kanten $\langle \mathfrak{g}uv \rangle, \langle \mathfrak{g}vu \rangle$ Für alle Maschen $\langle \mathfrak{g}u^2 \rangle \langle \mathfrak{g}v^2 \rangle$ $\tilde{p}, \tilde{\mathfrak{g}}$	9-14  15-22	$\langle \mathfrak{g}uv \rangle$ $\langle \mathfrak{g}vu \rangle$ $\langle \mathfrak{g}u^2 \rangle$ $\langle \mathfrak{g}v^2 \rangle$ $\tilde{p}, \tilde{\mathfrak{g}}$
AAF4 (NR)			
$u, v,$ $\langle \mathfrak{g}u^2 \rangle, \langle \mathfrak{g}v^2 \rangle$ $(\mathfrak{g}u), (\mathfrak{g}v)$ $\tilde{\mathfrak{g}}, \tilde{p}$ $\langle \mathfrak{g}uv \rangle,$ $\langle \mathfrak{g}vu \rangle$	Für alle Flächen $\tilde{u}, \tilde{v}$	23-28	$\tilde{u}$  $\tilde{v}$
AAF5 (NR)			
$\tilde{u}, \tilde{v},$ $\tilde{\mathfrak{g}}$	Für alle Flächen $\langle \tilde{\mathfrak{g}}\tilde{u} \rangle, \langle \tilde{\mathfrak{g}}\tilde{v} \rangle$	29-32	$\langle \tilde{\mathfrak{g}}\tilde{u} \rangle$ $\langle \tilde{\mathfrak{g}}\tilde{v} \rangle$

Eingabe	Ablauf	Schritt	Ausgabe
AAF6 (NR)			
$\tilde{u}, \tilde{v},$ $\tilde{s}, e$	Für alle Flächen $(\tilde{s}\tilde{u}), (\tilde{s}\tilde{v})$ Für alle Maschen $1/c^2, \beta$	33-34	$(\tilde{s}\tilde{u}), (\tilde{s}\tilde{v})$ $1/c^2$ $\beta$
AAF7 (NR)			
$\langle \tilde{s}\tilde{u} \rangle, \langle \tilde{s}\tilde{v} \rangle,$ $s, \tilde{s},$ $\beta, \tilde{p}$	Für alle Maschen $D, \delta\tilde{p}, \tilde{p}, \delta\tilde{s}, \tilde{s}$	45-52	$\tilde{p}$ $\tilde{s}$ $\delta\tilde{p}$
AAF8 (NR)			
$(\tilde{s}\tilde{u}), (\tilde{s}\tilde{v}),$ $\delta\tilde{p}, \tilde{s}$	Für alle Flächen $(\tilde{s}\tilde{u}), (\tilde{s}\tilde{v}), \tilde{u}, \tilde{v}$	52-57	$(\tilde{s}\tilde{u}), (\tilde{s}\tilde{v})$ $\tilde{u}, \tilde{v}$
AAF9 (NR)			
$\tilde{s}$ $\tilde{u}, \tilde{v}$	Für alle Flächen $\langle \tilde{s}\tilde{u} \rangle, \langle \tilde{s}\tilde{v} \rangle$	58-61	$\langle \tilde{s}\tilde{u} \rangle$ $\langle \tilde{s}\tilde{v} \rangle$
AAF10 (NR)			
$\tilde{p}, \tilde{s},$ $\tilde{u}, \tilde{v},$ $(\tilde{s}\tilde{u}), (\tilde{s}\tilde{v})$	Für alle Maschen $p, s$ Für alle Flächen $u, v, (su), (sv)$	63-68	$p, s$ $u, v$ $(su), (sv)$
AAF11 (NR)			
$u, v,$ $s, p,$ $e$	Für alle Flächen $\langle eu \rangle, \langle ev \rangle$ Für alle Maschen $e$	69-72  73	$e$
AAFTIM (NR)			
$u, v$	Für alle Maschen  timestephalf, -double	39-40  43	timestep- halt,  timestep- double

Eingabe	Ablauf	Schritt	Ausgabe
AAR1 (NR)			
$(\xi_u), (\xi_v)$ (Neustart)  $u, v$ (Restart)	Für alle Flächen  $u, v$ (Neustart)  $\xi_u, \xi_v$ (Restart)	2-4	$u, v$ (Neustart)  $(\xi_u), (\xi_v)$ (Restart)
AAR2 (NR)			
$u$ $\xi$	Für alle Randflächen (Fluid - Fluid) $\langle \xi u \rangle$	5-6	$\langle \xi u \rangle$
AAR3 (NR)			
$u, v,$ $(\xi_u),$ $(\xi_v)$	Für alle Kanten (Fluid - Fluid) $\langle \xi_{uv} \rangle, \langle \xi_{vu} \rangle$	9-14	$\langle \xi_{uv} \rangle$ $\langle \xi_{vu} \rangle$
AAR4 (NR)			
$u, v,$ $(\xi_u), (\xi_v),$ $\tilde{\xi}, \tilde{p},$ $\langle \xi u^2 \rangle \langle \xi v^2 \rangle,$ $\langle \xi_{uv} \rangle \langle \xi_{vu} \rangle$	Für alle Flächen $\tilde{u}$ (Fluid-Fluid) $\tilde{v}$ (Fluid)	23-26 27-28	$\tilde{u}$ $\tilde{v}$
AAR5 (NR)			
$\tilde{u}$ $\tilde{\xi}$	Für alle Randflächen $\langle \tilde{\xi} \tilde{u} \rangle$ (Fluid-Fluid)	29-30	$\langle \tilde{\xi} \tilde{u} \rangle$
AAR6 (NR)			
$\tilde{\xi}$ $e$	Für alle Maschen (Fluid) $1/c^2, \beta$	35-36	$1/c^2$ $\beta$

Eingabe	Ablauf	Schritt	Ausgabe
AAR8(NR)			
$\tilde{\xi}, \delta\tilde{p}$ $\tilde{u}, (\xi\tilde{u})$	Für alle Randflächen (Fluid - Fluid) $(\xi\tilde{u}), \tilde{u}$	52-54	$(\xi\tilde{u})$ $\tilde{u}$
AAR9(NR)			
$\tilde{\xi}, \tilde{u}$	Für alle Randflächen (Fluid - Fluid) $\langle \xi\tilde{u} \rangle$	58-59	$\langle \xi\tilde{u} \rangle$
AAR10(NR)			
$\tilde{p}, \tilde{\xi},$ $\tilde{u}, \tilde{v},$ $(\xi\tilde{u}), (\xi\tilde{v})$	Für alle Maschen $\xi$ Für alle Flächen $u, v, (\xi u), (\xi v)$	63-68	$\xi,$ $u, v,$ $(\xi u), (\xi v)$
AAR11(NR)			
$u, e$	Für alle Randflächen (Fluid - Fluid) $\langle e u \rangle$	69-70	$\langle e u \rangle$
AAN3(NR)			
$\xi, u, v$	Nur für Fluid an allen Seiten: $\langle \xi uv \rangle$ $\langle \xi vu \rangle$ sonst $\langle \xi uv \rangle, \langle \xi vu \rangle = 0$	9-14	$\langle \xi uv \rangle$ $\langle \xi vu \rangle$
AAN4(NR)			
$u, v, \tilde{\xi}, \tilde{p},$ $\langle \xi u^2 \rangle, \langle \xi v^2 \rangle,$ $\langle \xi uv \rangle,$ $\langle \xi vu \rangle,$ $(\xi u), (\xi v)$	Für alle Flächen $\tilde{u}, \tilde{v}$	23-28	$\tilde{u}$ $\tilde{v}$

AAN6 (NR)			
$\tilde{g}$ $e$	Für alle Maschen $1/c^2, \beta$	35-36	$1/c^2$ $\beta$

Zustandsgleichungen

QSPFLX(RHO,E)		
$\rho$	Zust.-Gl. $p(\rho)$ lineare Wassergl.	p
QSPFLX(RHO,E)		
	Schallgeschw. $\frac{1}{c^2}$ lineare Wassergl.	$1/c^2$
QSPH20(RHO,E)		
$\rho, e$	Zust.-Gl. $p(\rho, e)$ Wasser	p
QSCH20(RHO,E)		
$\rho, e$	Schallgeschw. $\frac{1}{c^2}(\rho, e)$ Wasser	$1/c^2$
QSPIG(RHO,E)		
$\rho, e$	Zust.-Gl. $p(\rho, e)$ ideales Gas	p
QSCIG(RHO,E)		
$\rho, e$	Schallgeschw. $\frac{1}{c^2}(\rho, e)$ ideales Gas	$1/c^2$



Eingaberoutine

INPUT(CODE,POOL,BIT,BF1,BF2,BF3,BF4,FLOAT,ENTRY)

CODE: 2 = Setzen der Zustandsgleichung

BF1 = 1: ideales Gas

3: lineare Wassergleichung

Sonst: Wasser

14 = Setzen einer konstanten Randbedingung  
bei Zufluß/Abfluß

BF2 = 1: Setze für Zufluß Nr. BF1

$s_{\text{Rand}} = \text{FLOAT}$

2: Setze für Zufluß Nr. BF1

$e_{\text{Rand}} = \text{FLOAT}$

3: Setze für Zufluß Nr. BF1

$(\mathcal{G}u)_{\text{Rand}} = \text{FLOAT}$

15 = einer zeitabhängigen Randbedingung bei Zufluß/Abfluß

Wie CODE = 14, jedoch ENTRY-Variable in ENTRY

Andere Parameter noch unbenutzt.

Anhang A3

Datenstruktur

Aufbau, Elemente, Indices,  
Zugriff auf Strukturelemente,  
PL/I-Deklarationen mit schnellem Zugriff.

Subsystem-Common

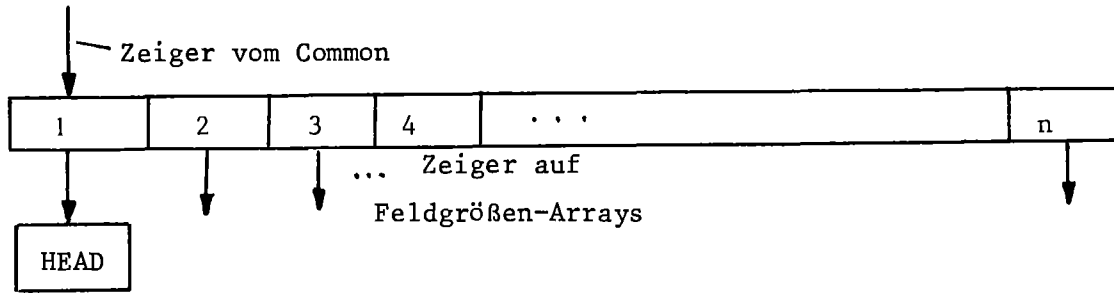
FLUID-Kennzeichnung $\alpha$		$\eta$	$\alpha_o$	$\beta_o$
$\gamma_1$	$\gamma_2$	$\omega$	$\epsilon$	iter
maxiter	$\Delta t$	t	cycle	timestep- half
timestep- double	$p(s, e)$	$1/c^2(s, e)$	ende	stop
print	Print- Kontrolle	save	IFLUIDS	IBOUNDARIES
ISEAMS	IJOINTS	FLUID_DESC		
BOUNDARY_DESC				
SEAM_DESC				
JOINT_DESC				
INFLOW_DESC				
SAVE_FILE				

PL/1-Deklaration des Common:

```
DCL 1 FLUSH,
  2 #FLUID CHAR(8),
  2(VISCOSITY INIT(1.0E-3),
    DONOR_CELL_FACTOR INIT(1.),
    PREDICTOR_FACTOR INIT(0.0),
    HALF_TIME_LIMIT INIT(0.5),
    DOUBLE_TIME_LIMIT INIT(0.01),
    OVER_RELAXATION_FACTOR INIT(1.0),
    ITERATION_LIMIT1 INIT(1.0E-3),
    ITERATION_LIMIT2 INIT(0.0)) BIN FLOAT(PREC),
  2(TIME_STEP INIT(0.01),
    TIME INIT(0.0),
    RESERVE1) BIN FLOAT(PREC),
  2 #R BIN FLOAT(PREC) INIT(6.6666E66),
  2(CYCLE INIT(0),
    ITER INIT(0)) BIN FIXED(15),
  2 ITERATION_MAXIMUM BIN FIXED(15) INIT(100),
  2 #LOADMODS BIN FIXED(15) INIT(0),
  2 (#PRINT CONTROL, #PRINT CONTROL1) BIN FIXED(15),
  2 (TIME_STEP_HALF, TIME_STEP_DOUBLE) BIT(1) ALIGNED INIT(0B),
  2 (#VALPRINT, #MODPRINT, #COMPRINT, #END_OF_ITERATION,
    #RBIT1, #RBIT2) BIT(1)
    ALIGNED INIT(0B),
  2 #RBIT3 BIT(8),
  2 (#F DE_LIST_PTR, #B DE_LIST_PTR, #S DE_LIST_PTR,
    #J DE_LIST_PTR) PTR INIT(NULL()), /* Zeiger auf ENTRIES */
  2(#P_FUNC ENTRY(BIN FLOAT(PREC), BIN FLOAT(PREC)),
    #DRHO_OVER_DP ENTRY(BIN FLOAT(PREC), BIN FLOAT(PREC)),
    #C_SQUARE_ENTRY(BIN FLOAT(PREC), BIN FLOAT(PREC)))
    DYNAMIC_VARIABLE RETURNS(BIN FLOAT(PREC)),
  2(#STOP, #PRINT, #PLOT, #SAVE) ENTRY VARIABLE RETURNS(BIT(1))
    INIT(DUMMY_FUN),
  2(#IFLUIDS, #IBOUNDS, #ISEAMS, #IJOINTS) BIN FIXED(15) INIT(0),
  2(#FLUID_DESC,
    #BOUND_DESC,
    #SEAM_DESC,
    #JOINT_DESC, #REF_DESC, #INFLOW_DESC) DESCRIPTOR INIT(NULL()),
  2 #SAVE_FILE FILE VARIABLE INIT(FSAVE),
  2 #SAVE_TITLE CHAR(8) INIT(SAVE),
  2 #SAVE_BIT BIT(1) INIT(0B);
DCL FSAVE_FILE RECORD;
DUMMY_FUN: PROC RETURNS(BIT(1));
RETURN(0B);
END DUMMY_FUN;
```

```
%DCL (PREC, MAX_INDEX) FIXED;
%PREC=53;
%MAX_INDEX=32767;
```

Komponenten



Indices für die einzelnen Teilkomponenten:

- 1 = HEAD = Zeiger auf Komponentenkopf
- 2 = MESH = Zeiger auf Maschinenmitten-Variable
- 3 = SIDE = Seiten von Rändern und Nähten
- 4 = EDGE0 = Kanten || 0 (parallel 0-Richtung)
- 5 = FACE1 = Flächen ⊥ 1 (senkrecht 1-Richtung)
- 6 = FACE2 = Flächen ⊥ 2
- 7 = EXTMESH1 = Außenmaschen ⊥ 1
- 8 = COL1 = Spalten ⊥ 1
- 9 = FACE0 = Flächen ⊥ 0
- 10 = EDGE2 = Kanten || 2
- 11 = COLO = Spalten ⊥ 0
- 12 = EXTMESHO = Außenflächen ⊥ 0
- 13 = EDGE1 = Kanten || 1
- 14 = EXTMESH2 = Außenflächen ⊥ 2 (nur 3D)
- 15 = COL2 = Spalten ⊥ 2 (nur 3D)
- 16 = NODE = Knoten (nur 3D)

Über jeden Index der Datenstruktur werden also die auf einer Masche, einer Fläche, einem Knoten oder sonstigen Teilkomponente liegenden Zustandsvariablen erreicht. Die Indices werden zur Übersetzungszeit festgelegt, so daß in den Programmen nicht die Zahlen, sondern die Namen der Teilkomponenten verwendet werden können. Die Zuordnung ist im Member PARTCODE der FLUST-Makrobibliothek enthalten:

```
%DCL(HEAD,NODE,EDGE0,EDGE1,
      EDGE2,FACE0,FACE1,FACE2,
      MESH,EXTMESH0,EXTMESH1,EXTMESH2,
      COL0,COL1,COL2,SIDE) FIXED;
%HEAD=1;
%NODE=16;
%EDGE0=4;
%EDGE1=13;
%EDGE2=10;
%FACE0=9;
%FACE1=5;
%FACE2=6;
%MESH=2;
%EXTMESH0=12;
%EXTMESH1=7;
%EXTMESH2=14;
%COL0=11;
%COL1=8;
%COL2=15;
%SIDE=3;
```

HEAD für alle Komponenten, 1-, 2- und 3-dimensional

IDENT (Kennzeichnung)		
DIMENSION	REFERENCE, SYSTEM	
ORIGIN (0)	(1)	(2)
(0) DIRECTION	(1)	(2)
SIZE (0)	(1)	(2)
IMAX	JMAX	KMAX

Zugriff:           #FLUID\_DESC (fluidnumber,HEAD,1)  
                  →IDENT,   etc  
                  #BOUND\_DESC (boundary-number,HEAD,1)  
                  →IDENT  
                  #SEAM\_DESC (seam-number,HEAD,1)  
                  →IDENT  
                  #JOINT\_DESC (joint-number,HEAD,1)  
                  →IDENT

Fluid, 2-dim:       DIMENSION=2  
                  SIZE(2)=0  
                  KMAX=0

Fluid, 1-dim:      DIMENSION=1  
                  SIZE(1),SIZE(2)=0  
                  JMAX,KMAX=0

Boundary, 3-dim:   DIMENSION=3  
                  SIZE(2),KMAX=0

Boundary, 2-dim:   DIMENSION=2  
                  SIZE(1),SIZE(2),JMAX,KMAX=0

Boundary, 1-dim:   DIMENSION=1  
                  SIZE(0),SIZE(1),SIZE(2)=0  
                  IMAX,JMAX,KMAX=0

Seam, 3-dim:       DIMENSION=3  
                  SIZE(1),SIZE(2)=0  
                  JMAX,KMAX=0

Seam, 2-dim:       DIMENSION=2  
                  SIZE=0  
                  IMAX,JMAX,KMAX=0

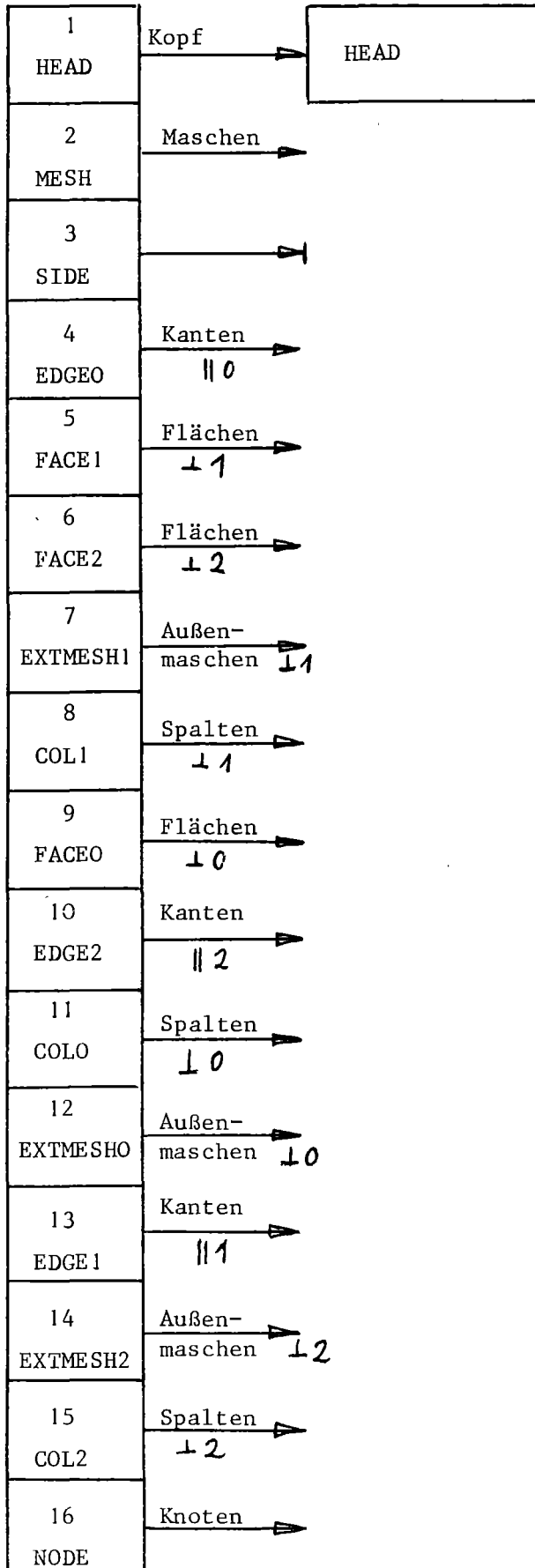
Joint, 3-dim:      DIMENSION=3  
                  SIZE=0  
                  IMAX,JMAX,KMAX=0

Globales Bezugssystem:   REFERENCE\_SYSTEM=0

PL/1-Deklaration

```
DCL 1 HEAD DYNAMIC(#FLUID_DESC), /* oder (#BOUND_DESC) oder */
  2 IDENT CHAR(2),             /*      (#SEAM_DESC)      */
  2 DIMENSION BIN FIXED(15),
  2 REFERENCE SYSTEM BIN FIXED(15),
  2 ORIGIN(0:2) BIN FLOAT(PREC),
  2 DIRECTION(0:2) BIN FLOAT(PREC),
  2 SIZE(0:2) BIN FLOAT(PREC),
  2 IMAX BIN FIXED(15),
  2 JMAX BIN FIXED(15),
  2 KMAX BIN FIXED(15);
```

Fluid, 3-dimensional



Zugriff:

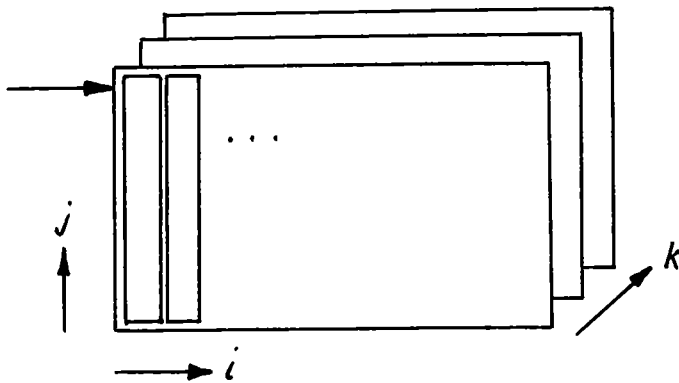
#FLUID\_DESC(fluid\_number,  
HEAD, .....

#FLUID\_DESC(fluid\_number,  
MESH, .....

usw.



Knoten zu Fluid 3-dim



Anzahl:  
(IMAX+1, JMAX+1, KMAX+1)

Zugriff: #FLUID\_DESC(fluid-number,  
          NODE,  
          i,j) → Wert(k)

Kanten || 0 zu Fluid 3-dim

Anzahl: (IMAX, JMAX+1, KMAX+1)  
Zugriff: #FLUID\_DESC(fluid-number,  
          EDGE0,  
          i,j) → Wert(k)

Kanten || 1

Anzahl: (IMAX+1, JMAX, KMAX+1)  
#FLUID\_DESC(fluid-number,  
          EDGE1,  
          i,j) → Wert(k)

Kanten || 2, Anzahl: (IMAX+1, JMAX+1, KMAX)

Flächen ⊥ 0, Anzahl: (IMAX+1, JMAX, KMAX)

Flächen ⊥ 1, Anzahl: (IMAX, JMAX+1, KMAX)

Flächen ⊥ 2, Anzahl: (IMAX, JMAX, KMAX+1)

Maschen , Anzahl: (IMAX, JMAX, KMAX)

Außenmaschen ⊥ 0, Anzahl: (JMAX, KMAX)

" ⊥ 1, Anzahl: (IMAX, KMAX)

" ⊥ 2, Anzahl: (IMAX, JMAX)

Spalten  $\perp$  0, Anzahl: (IMAX)  
           $\perp$  1, Anzahl: (JMAX)  
           $\perp$  2, Anzahl: (KMAX)

Weitere Komponenten werden nur für den zweidimensionalen Fall beschrieben.

Fluid, 2-dimensional

Abb. 72 gibt eine Übersicht über die Teilkomponenten eines zweidimensionalen Fluides.

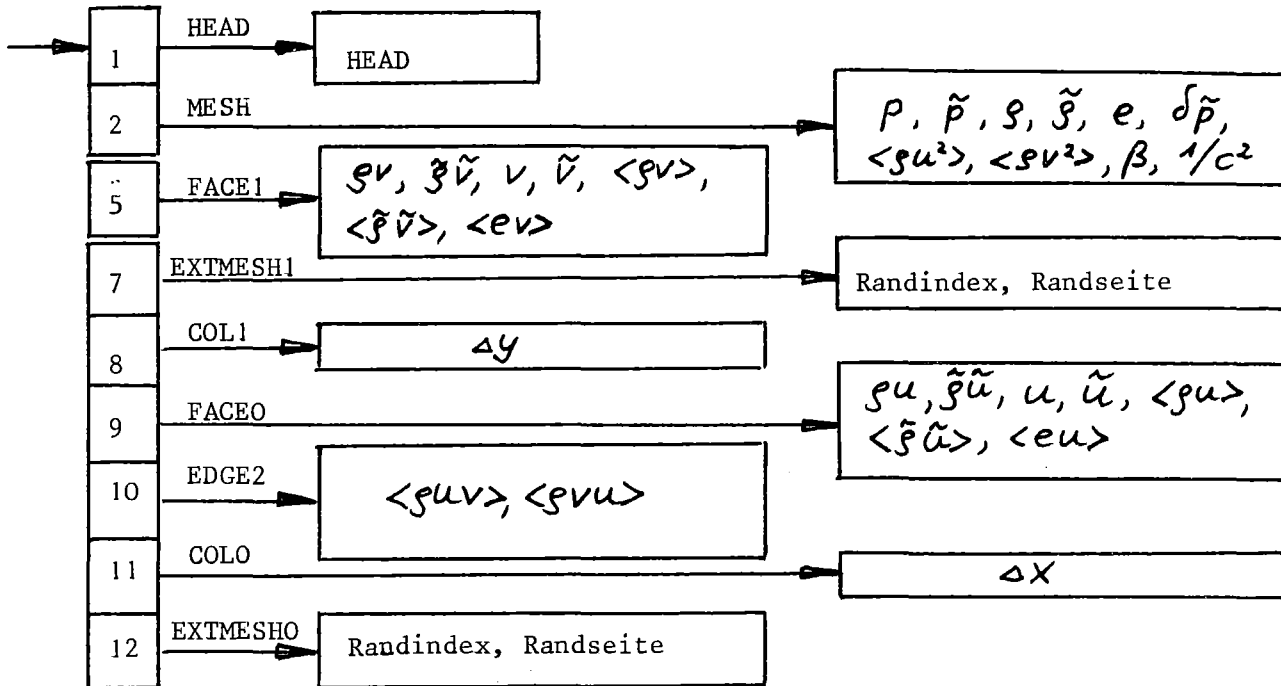


Abb.72 : Teilkomponenten eines 2D-Fluids

Kanten || 2 zu Fluid 2-dim

Anzahl: (IMAX+1, JMAX+1)

Zugriff: #FLUID\_DESC(fluid-number,  
EDGE2,  
i) → Wert(j)

Flächen ⊥ 0, Anzahl (IMAX+1), JMAX)

Flächen ⊥ 1, Anzahl (IMAX, JMAX+1)

Maschen , Anzahl (IMAX, JMAX)

Außenmaschen ⊥ 0, Anzahl (JMAX)

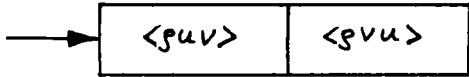
⊥ 1, Anzahl (IMAX)

Spalten ⊥ 0, Anzahl (IMAX)

⊥ 1, Anzahl (JMAX)

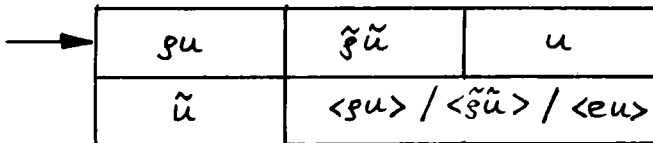
Deklarationen zu Fluid, 2D

Kanten



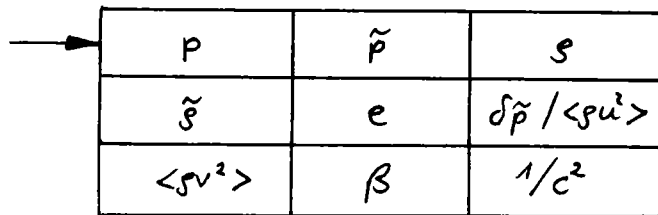
```
DCL 1 F2_EDGE DYNAMIC(#FLUID_DESC),
    2 RHOU1P2 BIN FLOAT(PREC), /* <math>\langle \rho v u \rangle</math> */
    2 RHOU2P1 BIN FLOAT(PREC); /* <math>\langle \rho v u \rangle</math> */
DCL 1 QF2_EDGE(MAX_INDEX) BASED(F2_EDGE_PTR) LIKE F2_EDGE;
```

Flächen



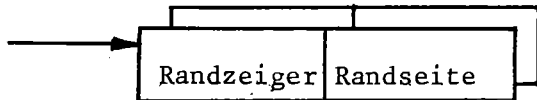
```
DCL 1 F2_FACE DYNAMIC(#FLUID_DESC),
    2 (RHO_U,
    RHO_UTIL,
    U,
    UTIL,
    RHO_U_DON) BIN FLOAT(PREC);
    /* = E U DON = RHO_UTIL DON */
DCL 1 QF2_FACE(MAX_INDEX) BASED(F2_FACE_PTR) LIKE F2_FACE;
%DCL RHO_UTIL_DON=CHAR;
%RHO_UTIL_DON=RHO_U_DON;
%DCL E_U_DON=CHAR;
%E_U_DON=RHO_U_DON;
```

Maschen



```
DCL 1 F2_MESH DYNAMIC(/#FLUID_DESC),
  2 (P,
    PTIL,
    RHO,
    RHOTIL,
    E,
    DP, /* = RHO_U2_DON */
    RHO_V2_DON,
    BETA,
    ONE OVER CSQUARE) BIN FLOAT(PREC);
DCL 1 QF2_MESH(MAX_INDEX) BASED(F2_MESH_PTR) LIKE F2_MESH;
%DCL RHO_U2_DON CHAR;
% RHO_U2_DON=DP;
```

Außenmaschen



```
DCL 1 F2_EXTMESH(2) DYNAMIC(/#FLUID_DESC),
  2 INDEX_BOUNDARY BIN FIXED(15), /* Index des Randes */
  2 SIDE_BOUNDARY BIN FIXED(15); /* Seite des Randes */
DCL 1 QF2_EXTMESH(MAX_INDEX,2) BASED(F2_EXTMESH_PTR) LIKE
F2_EXTMESH;
```

Maschenspalten



```
DCL 1 F2_COL DYNAMIC(/#FLUID_DESC),
  2 DX BIN FLOAT(PREC);
DCL 1 QF2_COL(MAX_INDEX) BASED(F2_COL_PTR) LIKE F2_COL;
```

Boundary, 2-dimensional

Teilkomponenten s. Abb. 73

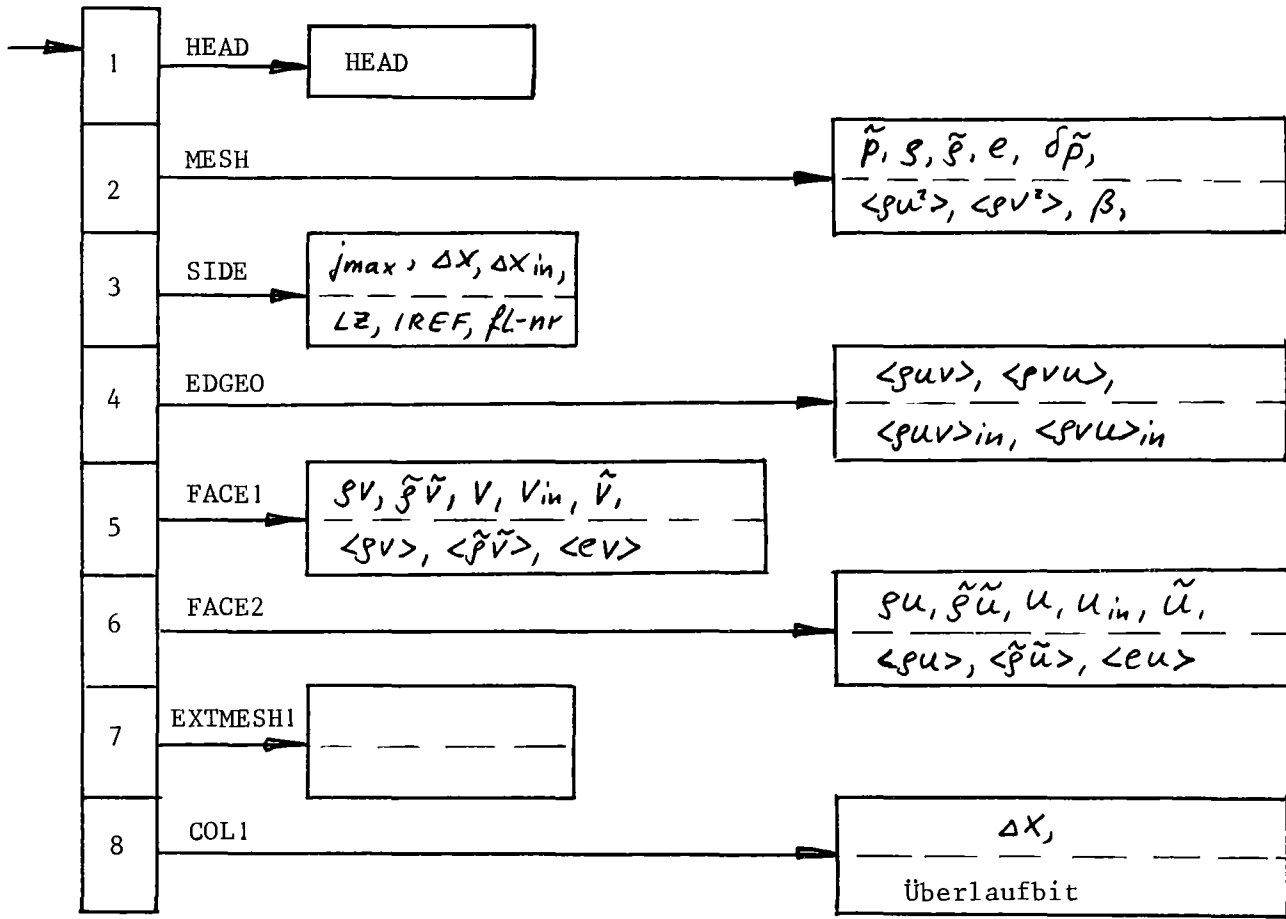


Abb. 73 : Teilkomponenten eines 2D-Randes

Kanten || 0 zu Boundary 2-dim

Anzahl (JMAX+1) (2)

Zugriff #BOUND\_DESC(boundary-number,

EDGE0, i)  $\rightarrow$  B2\_EDGE( $\begin{smallmatrix} 1 \\ 2 \end{smallmatrix}$ )....

Flächen  $\perp$  1: Anzahl (JMAX+1) (2)

Flächen  $\perp$  2: (JMAX) (2)

Maschen (JMAX) (2)

Außenmaschen  $\perp$  1: (1) (2)

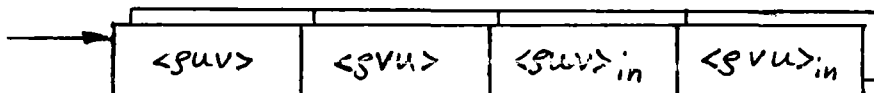
Spalten  $\perp$  1: (JMAX) (2)

Seiten (2)

JMAX = max (JMAX<sub>links</sub>, JMAX<sub>rechts</sub>)

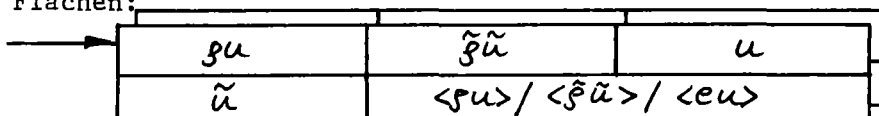
Deklarationen der Feldgrößen-Arrays:

Kanten:



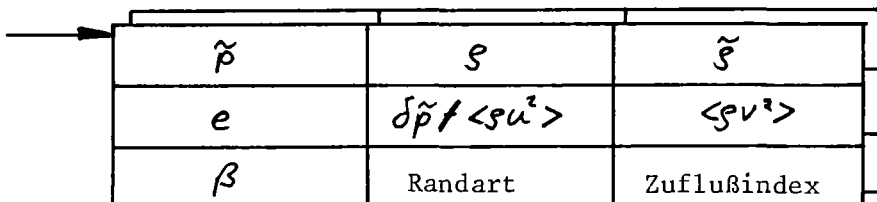
```
DCL 1 B2_EDGE(2) DYNAMIC(#BOUND_DESC),
  2 RHOU1P2 BIN FLOAT(PREC), /* <RHO U V> */
  2 RHOU2P1 BIN FLOAT(PREC), /* <RHO V U> */
  2 RHOU1P2_IN BIN FLOAT(PREC),
  2 RHOU2P1_IN BIN FLOAT(PREC);
DCL QB2_EDGE(MAX_INDEX,2) BASED(B2_EDGE_PTR) LIKE B2_EDGE;
```

Flächen:



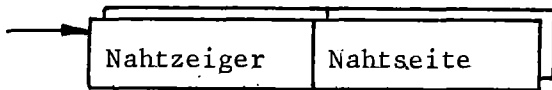
```
DCL 1 B2_FACE(2) DYNAMIC(#BOUND_DESC),
  2 (RHO U,
  RHO_UTIL,
  U,
  U_IN,
  UTIL,
  RHO_U_DON) BIN FLOAT(PREC);
/* = E U_DON = RHO_UTIL_DON */
DCL QB2_FACE(MAX_INDEX,2) BASED(B2_FACE_PTR) LIKE B2_FACE;
%DCL RHO_UTIL_DON CHAR;
%RHO_UTIL_DON=RHO_U_DON;
%DCL E_U_DON CHAR;
%E_U_DON=RHO_U_DON;
```

Maschen:

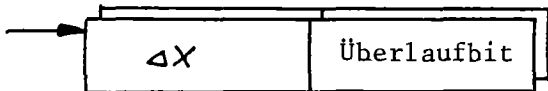


```
DCL 1 B2_MESH(2) DYNAMIC(#BOUND_DESC),
  2 (PTIL,
  RHO,
  RHOTIL,
  E,
  DP, /* = RHO_U2_DON */
  RHO_V2_DON,
  BETA) BIN FLOAT(PREC),
  2 WALLTYPE BIN FIXED(15),
  2 INFLOW_INDEX BIN FIXED(15);
DCL QB2_MESH(MAX_INDEX,2) BASED(B2_MESH_PTR) LIKE B2_MESH;
%DCL RHO_U2_DON CHAR;
%RHO_U2_DON=DP;
```

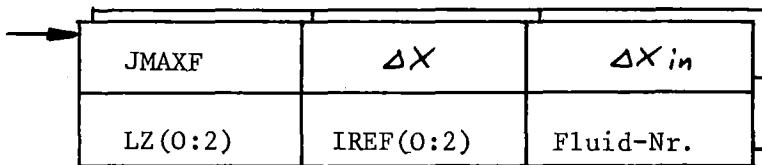
Außenmaschen:



```
DCL 1 B2_EXTMESH(2) DYNAMIC(#BOUND_DESC),
  2 INDEX_SEAM BIN FIXED(15), /* Index der Naht */
  2 SIDE_SEAM BIN FIXED(15); /* Seite der Naht */
```



```
DCL 1 B2_COL(2) DYNAMIC(#BOUND_DESC),
  2 DX BIN FLOAT(PREC),
  2 CHAIN BIT(1)ALIGNED; /* Wenn 1B, dann gehört */
                          /* diese Masche logisch zur */
                          /* nächsten (Änderung der */
                          /* Maschenweite am Rand) */
DCL QB2_COL(MAX_INDEX,2) BASED(B2_COL_PTR) LIKE B2_COL;
```

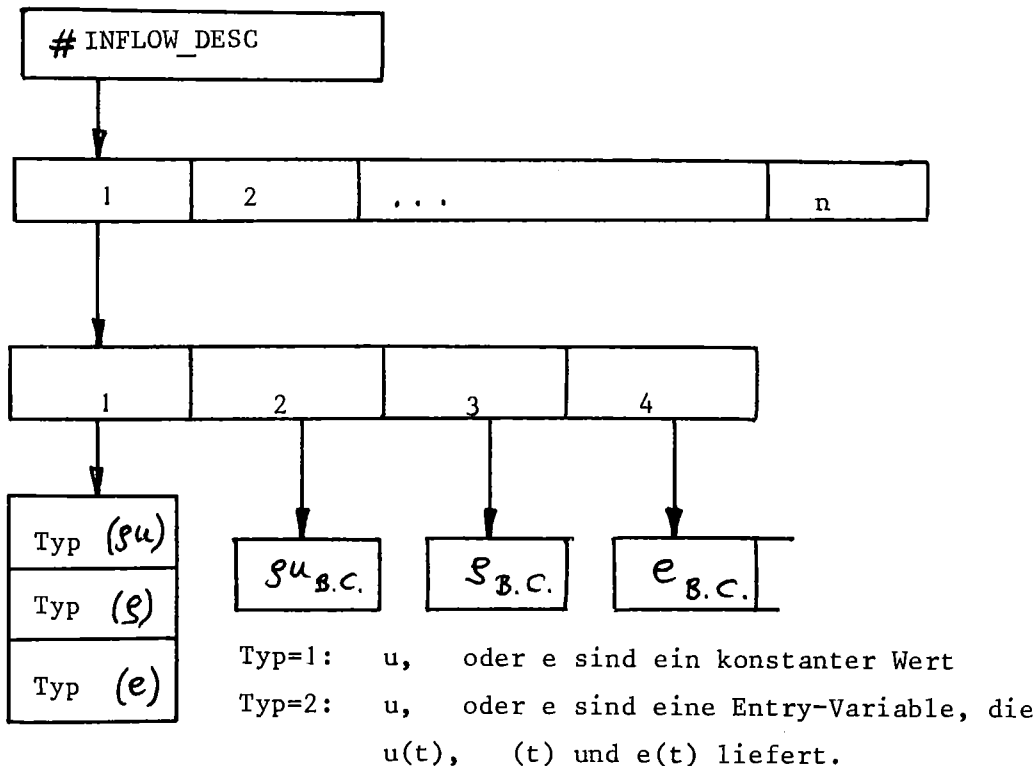


```
DCL 1 B2_SIDE(2) DYNAMIC(#BOUND_DESC),
  2 JMAXF BIN FIXED(15),
  2 DX BIN FLOAT(PREC),
  2 DX_IN BIN FLOAT(PREC),
  2 LZ(0:2) BIN FIXED(15),
  2 IREF(0:2) BIN FIXED(15),
  2 FLUID_BACK BIN FIXED(15);
DCL (B2_EDGE_PTR,B2_FACE_PTR,B2_MESH_PTR,B2_COL_PTR) PTR;
```



Zufluß/Abfluß

Der Zeiger #INFLOW\_DESC des Common zeigt auf eine Datenstruktur zur Beschreibung von Zuflüssen und Abflüssen. Die Variable #INFLOW\_INDEX bei den Maschengrößen des Randes gibt bei Rändern mit Randart = 6 (Zufluß/Abfluß) den Index der zugehörigen Datenstruktur an.



Deklaration:

```

/* DEFINE INFLOW(I), (4) DESCRIPTOR */
/* 1=INFLOW-TYPES */
/* 2=RHO_U, 3=RHO, 4=E */
DCL 1 INFLOW_TYPES DYNAMIC(#INFLOW_DESC),
    2 (RHO_U_TYP,RHO_TYP,E_TYP) BIN FIXED(15);
DCL (INCONST_RHO_U,INCONST_RHO,INCONST_E) DYNAMIC
    (#INFLOW_DESC) BIN FLOAT(PREC);
DCL 1 INFLOW_RHO U DYNAMIC
    (#INFLOW_DESC),
    2 INFLOW_ENTRY ENTRY(BIN FLOAT(PREC)) VARIABLE
    RETURNS(BIN FLOAT(PREC));
DCL 1 INFLOW_RHO DYNAMIC(#INFLOW_DESC) , 2 INFLOW_ENTRY ENTRY
    (BIN FLOAT(PREC)) VARIABLE RETURNS(BIN FLOAT(PREC));
DCL 1 INFLOW_E DYNAMIC(#INFLOW_DESC) , 2 INFLOW_ENTRY ENTRY
    (BIN FLOAT(PREC)) VARIABLE RETURNS(BIN FLOAT(PREC));
    
```

Naht, 2-dimensional

Teilkomponenten s. Abb.74

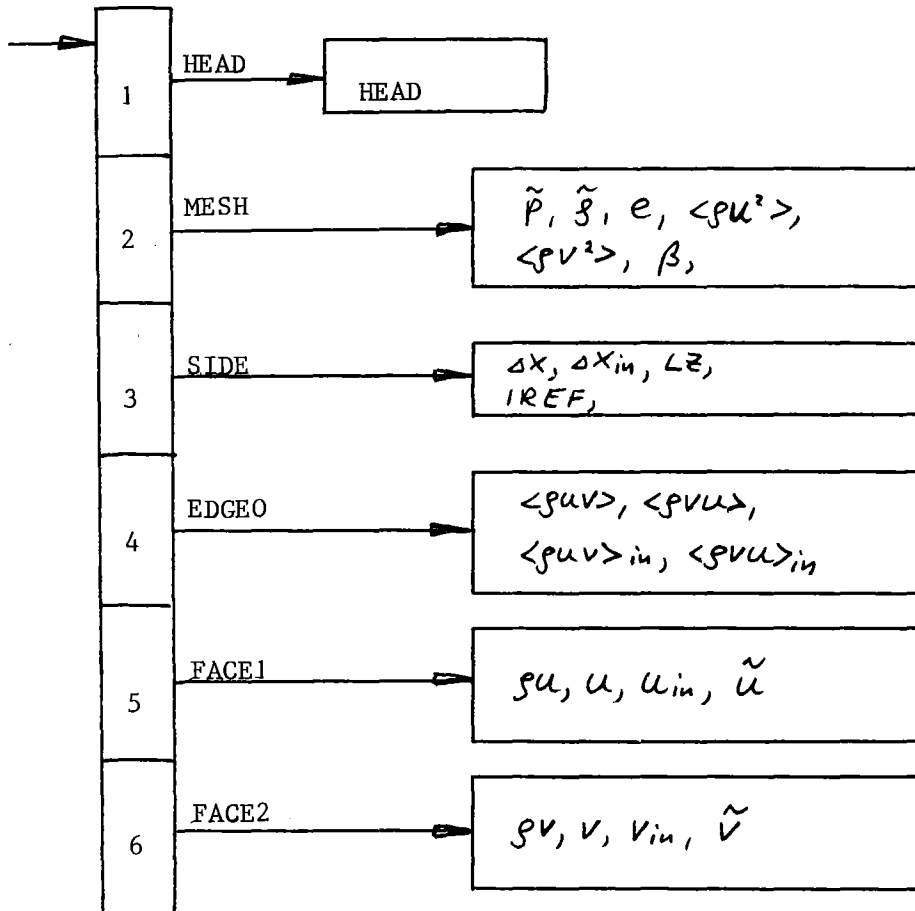


Abb.74 : Teilkomponenten einer 2D-Naht

Kanten || 0 zu Seam 2-dim

Anzahl (2,2,2)

- (1. Index: x- oder y-Richtung,
- 2. Index: links oder rechts bzw. oben oder unten
- 3. Index: linke oder rechte Randseite)

Zugriff #SEAM\_DESC(seam-number,EDGE0)  
 ->S2\_EDGE(ir,i12,ib12).....

Flächen || 1 (2,2,2)

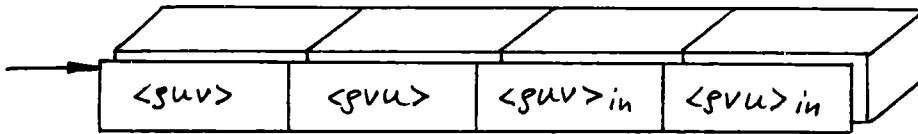
Flächen || 2 (2,2,2)

Maschen (2,2,2)

Seiten (2,2)

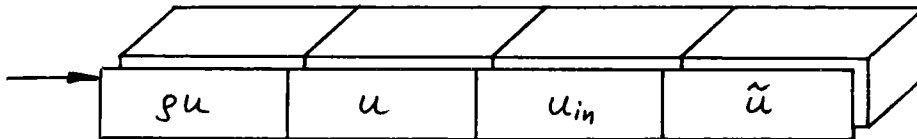
Deklarationen für 2D-Naht:

Kanten:



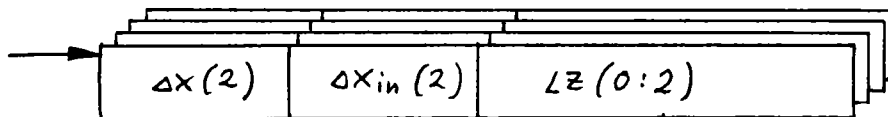
```
DCL 1 S2_EDGE(2,2,2) DYNAMIC(#SEAM_DESC),
  2  $\bar{\rho}uP1P2$  BIN FLOAT(PREC), /* <math>\langle \rho u v \rangle</math> */
  2  $\bar{\rho}uP2P1$  BIN FLOAT(PREC), /* <math>\langle \rho v u \rangle</math> */
  2  $\bar{\rho}uP1P2_{in}$  BIN FLOAT(PREC),
  2  $\bar{\rho}uP2P1_{in}$  BIN FLOAT(PREC);
```

Flächen:



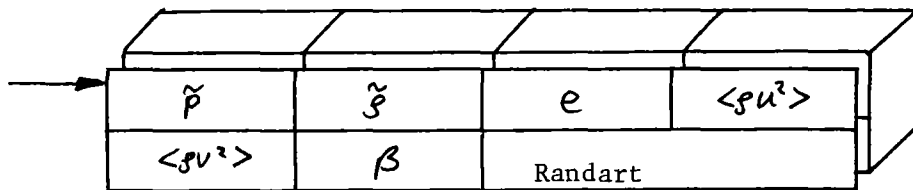
```
DCL 1 S2_FACE(2,2,2) DYNAMIC(#SEAM_DESC),
  2  $\bar{\rho}u$  BIN FLOAT(PREC),
  2  $\bar{u}$  BIN FLOAT(PREC),
  2  $\bar{u}_{in}$  BIN FLOAT(PREC),
  2  $\tilde{u}$  BIN FLOAT(PREC);
```

Seiten:



```
DCL 1 S2_SIDE(2,2) DYNAMIC(#SEAM_DESC),
  2  $\bar{\Delta x}(2)$  BIN FLOAT(PREC),
  2  $\Delta x_{in}(2)$  BIN FLOAT(PREC),
  2  $Lz(0:2)$  BIN FIXED(15),
  2  $Iref(0:2)$  BIN FIXED(15),
  2  $BOUND\_BACK$  BIN FIXED(15);
```

Maschen:



```
DCL 1 S2_MESH(2,2,2) DYNAMIC(#SEAM_DESC),
      2 PTIL BIN FLOAT(PREC),
      2 RHOTIL BIN FLOAT(PREC),
      2 E BIN FLOAT(PREC),
      2 RHO_U2_DON BIN FLOAT(PREC),
      2 RHO_V2_DON BIN FLOAT(PREC),
      2 BETA BIN FLOAT(PREC),
      2 WALLTYPE BIN FIXED(15);
```

### Schnellzugriff

Bei den Quick-Zugriffs-Strukturen muß jeweils eine Dimension mehr deklariert werden [19]. Statt DYNAMIC(...DESC) steht BASED (name\_PTR) name = Name der Teilkomponente. Die BASED-Strukturen tragen ein Q vor dem Namen. Beispiel:

```
DCL 1 F3_MESH DYNAMIC(#FLUID_DESC),
      2 (P,PML,RHO,RHOTIL,E,DP,D,BETA);
```

Quick-Struktur:

```
DCL 1 QF3_MESH(MAX_INDEX)BASED(F3_MESH_PTR)
      LIKE F3_MESH; (MAX_INDEX siehe S.171)
```

### Anwendung, Bsp:

```
FIX F3_MESH(i,j) SET(F3_MESH_PTR) ON(1);
DØ K=1 TO KMAX;
```

```
QF3_MESH(K).P = ....
```

```
END;
```

```
LOOSE F3_MESH(i,j);
```

Die Deklarationen der BASED-Strukturen sind bei den Datenstruktur-Deklarationen mit angegeben.