NSCNG2

Ein Programm zur räumlich zweidimensionalen, instationären Strömungssimulation thermisch perfekter, aber kalorisch nicht perfekter Gasgemische im chemischen Nichtgleichgewicht

Stand: Mai 1994

Siegfried Müller[1]
Institut für Geometrie und Praktische Mathematik
RWTH Aachen

E-Mail: mueller@igpm.rwth-aachen.de

Zusammenfassung

In dieser Arbeit wird ein Programm zur Simulation von räumlich zweidimensionalen bzw. rotationssymmetrischen kompressiblen Strömungen dokumentiert. Das betrachtete Fluid ist ein thermisch perfektes aber kalorisch nicht perfektes Gasgemisch. Dieses befindet sich im chemischen Nichtgleichgewicht. Reibung, Wärmeleitung und Diffusion werden berücksichtigt. Das zugrundeliegende physikalische Modell wird durch die Navier-Stokes-Gleichungen beschrieben. Diese müssen um einen Produktionsterm zur Beschreibung der chemischen Produktionsraten erweitert werden. Dieses System partieller Differentialgleichungen wird mittels eines expliziten Shock-Capturing-Verfahrens diskretisiert. In jedem Zeitschritt wird der Quellterm der chemischen Produktionsraten vom restlichen System abgespalten. Dies führt auf ein Sytem partieller Differentialgleichungen und ein gewöhnliches Differentialgleichungssytem. Zur Diskretisierung des reibungsfreien Anteils der Navier-Stokes-Gleichungen werden ENO-Verfahren und approximative Riemannlöser verwendet. Die Terme zweiter Ordnung — reibungsbehafteter Anteil — werden durch zentrale Differenzen diskretisert. Erweiterungen auf ein zonales Konzept und eine daraus resultierende Parallelisierung des Programms sind darauf aufbauend beschrieben.

Key Words: Navier-Stokes equations, chemically reacting flows, finite difference methods, applications of mathematical programming

AMS subject classification: 35Q30, 80A32, 76M20, 90C90

[1]Der Autor wurde von der Deutschen Forschungsgemeinschaft im Sonderforschungsbereich 253, Teilprojekt B5, gefördert.

Vorwort

Die vorliegende Dokumentation des Programms **NSCNG2** (**N**avier–**S**tokes–Gleichungen für **c**hemisches **N**icht**g**leichgewicht in **2** Raumdimensionen) soll dem Benutzer eine schnelle Einarbeitung ermöglichen und dem Programmierer einen Überblick über die Programmstruktur, die verwendeten Parameter, Variablen und Flags sowie die Strukturen der einzulesenden Dateien vermitteln. Dem Benutzer werden insbesondere die Kapitel 1, 2, 3 und 5 zur Einarbeitung empfohlen, der Programmierer sollte darüber hinaus die Kapitel 4, 6 und 7 lesen.

An dieser Stelle sei darauf hingewiesen, daß das Programm im Rahmen des Sonderforschungsbereiches *„Grundlagen zum Entwurf von Raumfahrzeugen“* im Teilprojekt B5 *„Berechnung hypersonischer Staupunktströmungen bei hohen Temperaturen unter Berücksichtigung chemischer Reaktionen“* im Zeitraum Juli 89 – Juni 93 entwickelt und validiert wurde.

Um auch komplexere Geometrien berechnen zu können, wurde das Programm im Frühjahr 1994 auf ein zonales Konzept erweitert. Dieses baut wesentlich auf das ursprüngliche Programm auf. Änderungen an der Programmstruktur wurden vermieden. Dieses Konzept ermöglicht auch eine Parallelisierung, die auf einem Workstation–Cluster durchgeführt wurde.

Der Chemnitzer Forschergruppe *Scientific Parallel Computing* sei hier insbesondere für ihre Programmbibliothek **PAR-LIB** zur Entwicklung paralleler Algorithmen gedankt, die sie mir freundlicherweise zur Benutzung überlassen hat.

Mein besonderer Dank gilt Stephan Mannheim für seinen Einsatz bei der Erstellung des Layouts dieser Arbeit sowie allen Kollegen vom

- Institut für Geometrie und Praktische Mathematik, RWTH Aachen,
- Lehr– und Forschungsgebiet für Mechanik, RWTH Aachen und
- Seminar für Angewandte Mathematik, ETH Zürich,

die mir in den zurückliegenden Jahren bei der Entwicklung des Programms behilflich waren.

Inhaltsverzeichnis

1 Einleitung **1**

2 Die inhomogenen Navier-Stokes-Gleichungen **4**

3 Numerische Methoden **8**

4 Programmstruktur **14**
4.1 NSCNG2 (Hauptprogramm) . . . 14
4.2 INPUT . . . 14
4.3 INITIAL . . . 16
4.4 CHEMIE . . . 17
4.5 CFD . . . 18
4.5.1 BOUND . . . 18
4.5.2 MWQUAN . . . 21
4.5.3 REKON . . . 22
4.5.4 FLINV . . . 23
4.5.4.1 RPN . . . 24
4.5.4.2 FIBOU . . . 25
4.5.4.3 ROELSG . . . 25
4.5.5 FLVIS . . . 26
4.5.5.1 VFLPAR . . . 27
4.5.5.2 VFLWAL . . . 28
4.5.5.3 RQVXI, RQVETA, RQWVXI, RQWVETA . . . 28
4.5.6 MWSTEP . . . 29
4.5.7 TEST . . . 30
4.5.7.1 PTEST . . . 30
4.5.7.2 STEADY . . . 31
4.6 OUTPUT . . . 31

5 Programmsteuerung über Optionen und Dateien **32**
5.1 Optionen . . . 32
5.2 Dateistrukturen . . . 33
5.2.1 Eingabedatei . . . 33
5.2.2 Referenzgrößen der Anströmung . . . 35
5.2.3 Gitter . . . 35
5.2.4 Mittelwerte der Bilanzgrößen . . . 36
5.2.5 Tabellen . . . 36
5.2.6 Protokoll . . . 39

6 Schnittstellen **41**
6.1 Energiemodell . . . 41
6.1.1 Schnittstelle IET . . . 41
6.1.2 Schnittstelle CVT . . . 42
6.1.3 Schnittstelle IEI . . . 43

6.1.4 Schnittstelle CVI 43
6.1.5 Schnittstelle TEMP 44
6.2 Modellierung der chemischen Produktionsraten 45
6.2.1 Schnittstelle CHEPRO 45
6.3 Transport– und Diffusionsmodell 46
6.3.1 Schnittstelle KOEFF 48
6.3.2 Schnittstelle MDIFF 49
6.3.3 Schnittstelle VISCOE 50
6.3.4 Schnittstelle DVISCO 51

7 Globale Variablen **53**
7.1 INCLUDE–Datei FIXDEK 53
7.2 INCLUDE–Datei FLDEK 56
7.3 INCLUDE–Datei NETDEK 59
7.4 INCLUDE–Datei REKDEK 60
7.5 INCLUDE–Datei RPDEK 61
7.6 INCLUDE–Datei CHEDEK 62

8 Warnungen **65**

9 Mögliche Änderungen durch den Benutzer **66**

10 Compilieren, Linken und Programmstart **68**

11 Zonales Konzept **69**
11.1 Programmstruktur 70
11.1.1 ZNSCNG2 (Hauptprogramm) 70
11.1.2 ZINPUT 71
11.1.3 ZINITIAL 72
11.1.4 ZCHEMIE 73
11.1.5 ZCFD 74
11.1.6 ZBOUND 75
11.1.7 ZREKON 75
11.1.8 ZOUTPUT 75
11.2 Kommunikation zwischen den Zonen 76
11.3 Programmsteuerung über Optionen und Dateien 77
11.3.1 Optionen 77
11.3.2 Dateistrukturen 78
11.3.2.1 Eingabedatei 78
11.3.2.2 Referenzgrößen der Anströmung 79
11.3.2.3 Zonales Gitter 80
11.3.3 Mittelwerte der Bilanzgrößen 80
11.3.4 Protokoll 80
11.4 Globale Variablen 81
11.4.1 INCLUDE–Datei ZONDEK 81
11.5 Warnungen 84

11.6 Mögliche Änderungen durch den Benutzer 84
11.7 Compilieren, Linken und Programmstart 85

12 Paralleles Konzept **86**
12.1 Programmstruktur 87
12.1.1 PNSCNG2 (Hauptprogramm) 87
12.1.2 PCHEMIE 89
12.1.3 PCFD 89
12.2 Kommunikation zwischen den Zonen 91
12.3 Kommunikation zwischen den Prozessoren 93
12.4 Globale Variablen 93
12.4.1 INCLUDE–Datei trnet.inc 94
12.5 Compilieren, Linken und Programmstart 94

13 Anhang **96**
13.1 Abbildungen 96
13.2 Zuordnung Datei — Unterprogramm für NSCNG2 109
13.3 Zuordnung Datei — Unterprogramm für ZNSCNG2 110
13.4 Zuordnung Datei — Unterprogramm für PNSCNG2 111

1 Einleitung

Im Rahmen des Sonderforschungsbereichs 253 wurde im Teilprojekt B5 ein FORTRAN 77-Programm entwickelt, das die numerische Simulation von kompressiblen Strömungen ermöglicht. Dieses wurde dazu verwendet, um Strömungen um einen stumpfen Körper bei einer Überschallanströmung ($Ma_\infty \gg 1$) zu untersuchen. Diese sind an räumlich zweidimensionalen Geometrien beziehungsweise an rotationssymmetrischen Körpern (z.B. Kugel mit Radius r) durchgeführt worden. Die Ergebnisse sind in [12], [13] dargestellt.
Aufgrund der für solche Probleme typisch hohen Anströmungsgeschwindigkeit und der relativ großen Höhen (30–80 km) kommt es hinter dem Frontstoß zu Temperaturen von teilweise weit über 3000 K. Dies führt zur Dissoziation von molekularem Sauerstoff und Stickstoff. Bei Temperaturen von über 8000 K kommt es ferner zur Ionisation der chemischen Komponenten (z.B. Stickoxid). Da der Nasenradius r in der Regel sehr klein ist (z.B. ELAC-1: $r = 0.18$m, [14]), befindet sich der Frontstoß sehr nahe am Körper. Dadurch haben die chemischen Reaktionen nicht genügend Zeit, um in den Gleichgewichtszustand zu relaxieren. Deshalb wurde chemisches Nichtgleichgewicht modelliert. Dazu wurde das 5-Komponenten-Modell von Park [16] für N, N_2, O, O_2 und NO, das die wichtigsten 17 Reaktionen berücksichtigt, verwendet.
Da auf eine Modellierung der Ionisationseffekte verzichtet wurde, brauchen auch keine elektrischen Kräfte berücksichtigt zu werden. Diese dürfen jedoch bei einem hohen Ionisationsgrad des Gasgemisches nicht vernachlässigt werden.
Auf eine Modellierung von thermischem Nichtgleichgewicht wurde verzichtet. Bei den hier untersuchten Strömungen kann davon ausgegangen werden, daß sich die Translations– und die Rotationstemperatur im Gleichgewicht befinden. Bei Anströmungsgeschwindigkeiten über etwa 3000 m/s ist die Vibrationstemperatur ebenfalls im Gleichgewicht (siehe [4]).
Um quantitative Aussagen über den Wärmetransport an der Wand machen zu können, wurden Reibungs– und Diffusionseffekte berücksichtigt. Katalytische Wandeffekte wurden jedoch vernachlässigt. Diese viskosen Effekte hängen wesentlich von der Modellierung der Transport– und Diffusionskoeffizienten ab. Da diese aber noch Gegenstand laufender Forschungsarbeiten sind (z.B. Modellierung des intermolekularen Potentials, experimentell zu bestimmende Parameter, etc.), wurde auf die explizite Implementierung eines speziellen Modells verzichtet. Stattdessen wurden die entsprechenden Koeffizienten für jede einzelne chemische Komponente mittels einer Least-Square-Approximation über den Temperaturbereich gefittet (siehe [8]). In einem Gasgemisch hängen diese Koeffizienten jedoch zusätzlich von der Zusammensetzung des Gases ab. Um die Gemischkoeffizienten realistisch zu modellieren, wurden die Mischungsregeln von Obermeier und Schaber (D_{ij}), Wilke (μ) sowie Mason (λ) verwendet. Näheres über die Qualität der Modellierung findet sich in [9]. Dort werden die Transport– und Diffusionskoeffizienten – ausgehend von der kinetischen Gastheorie – betrachtet und verschiedene Alternativen speziell für das 5-Komponenten-Modell diskutiert.
Zur Modellierung der massenspezifischen inneren Energie e und der massenspezifischen Wärmekapazitäten bei konstantem Volumen c_V wurde analog vorgegangen. Hier sind die entsprechenden Größen e_i und $c_{V,i}$ für eine einzelne chemische Komponente i gefittet worden. Die Größen für das Gemisch können daraus additiv durch Gewichtung mit der entsprechenden Massenkonzentration bestimmt werden. Dadurch bleibt es dem Benutzer

vorbehalten, zu entscheiden, ob ein harmonischer bzw. ein anharmonischer Oszillator verwendet werden soll, ob eine Kopplung zwischen Rotations- und Vibrationsenergie bestehen soll oder welche Parameter verwendet werden sollen.
Diese Strategie der Kurvenfits für physikalische Größen der einzelnen chemischen Komponenten hat mehrere Vorteile:

- Flexibilität bei Änderung des Modells
- effiziente Auswertung aufwendiger analytischer Modelle bei Abweichungen, die im Rahmen des analytischen Modells liegen
- Vermeidung von Fallunterscheidungen zur Laufzeit
- hoher Vektorisierungsgrad

Zur Modellierung des bisher beschriebenen pkysikalischen Modells wurden die Navier-Stokes-Gleichungen (NSGen), bestehend aus den Komponentengleichungen für die Partialdichten, den Impulsgleichungen und der Energiegleichung, auf einem strukturierten krummlinigen Gitter, das sich aus regelmäßigen Vierecken zusammensetzt, diskretisiert.
Dazu wurde ein explizites Shock-Capturing-Verfahren implementiert. In jedem Zeitschritt wird der Quellterm der chemischen Reaktionen mittels eines Operatorsplittings von den NSGen getrennt. Daraus resultiert ein System gewöhnlicher Differentialgleichungen (GDGen) und ein homogenes System partieller Differentialgleichungen (PDGen).
Die Lösung der GDGen wird mittels eines expliziten Eulerschrittes approximiert. Die homogenen NSGen werden durch die Finite-Volumen-Methode, d.h. Integration über ein räumlich-zeitliches Kontinuum, diskretisiert. Die räumliche Integration wird mittels des Satzes von Gauß auf die Integration entlang des Zellrandes überführt.
Bei der Diskretisierung der Randflüsse wird zwischen dem reibungsfreien und dem viskosen Anteil unterschieden. Die im viskosen Fluß auftretenden räumlichen Ableitungen 1.Ordnung werden durch zentrale Differenzen approximiert. Zur Diskretisierung des reibungsfreien Flusses werden in den PDGen die viskosen Terme vernachlässigt. Dies führt auf die Eulergleichung (EGen) in Integraldarstellung.
Auf diese wird die Dimensionssplitting-Technik angewendet, d.h. die Erhaltungsgrößen werden auf jeder Zellkante konstant angenommen, und somit auch der Randfluß. Letzterer muß nur noch in den Zellrandmittelpunkten bestimmt werden.
Dazu werden die Mittelpunkte zweier gegenüberliegender Zellkanten miteinander verbunden. Dadurch wird eine Schar einparametriger Kurven konstruiert. Entlang jeder dieser Kurven werden die räumlich eindimensionalen EGen mittels eines Godunov-Typ-Verfahrens – dem sogenannten Essentially Non-Oscillatory-Verfahren (ENO) [1] – diskretisiert.
Dieses basiert auf einer Rekonstruktionstechnik (hier: RP-Methode, [1]), die aus den gegebenen Mittelwerten einer Funktion an diskreten Stellen eine kontinuierliche, stückweise glatte Funktion, die an den Zellrändern springen darf, approximiert. In den Zellrandmittelpunkten werden nun eindimensionale Riemannprobleme gelöst. Dies geschieht aus Rechenzeitgründen nicht exakt, sondern approximativ mit dem verallgemeinerten Löser von Liu und Vinohur [11] für die EGen im thermischen Gleichgewicht und chemischen Nichtgleichgewicht.

Daraus ergibt sich eine Approximation des reibungsfreien Randflusses.
Ziel dieser Arbeit ist es nun, das oben in groben Zügen beschriebene Programm zu dokumentieren. Damit sollen Fremdbenutzer in die Lage versetzt werden, eigene Simulationen durchzuführen. Dazu sind folgende Kenntnisse erforderlich:

- Programmstruktur, z.B.
  - Inhalt der Unterprogramme,
  - globale Variablen
- Ein– und Ausgabe
- Schnittstellen für die physikalischen Modelle, z.B.
  - Kurvenfits für Transport- und Diffusionskoeffizienten,
  - Kurvenfits für die innere Energie und die isochoren Wärmen,
  - Kurvenfits für die Gleichgewichtskonstanten,
  - Diffusionsfluß,
  - Randbedingungen.

Zunächst werden deshalb die zugrunde liegenden Gleichungen explizit angegeben (siehe Kap. 2) und die sich daraus durch das Operatorsplitting und Dimensionssplitting ergebenden Teilprobleme beschrieben. Daran schließt sich die Beschreibung der numerischen Methoden zur Diskretisierung der Teilprobleme an (siehe Kap. 3). Hier werden jedoch **keine** expliziten Algorithmen angegeben. Diese können in [12] nachgeschlagen werden. Auf die entsprechenden Stellen wird verwiesen.
Darauf aufbauend kann die Programmstruktur nachvollzogen werden.

2 Die inhomogenen Navier-Stokes-Gleichungen

Das in der Einleitung beschriebene physikalische Modell einer kompressiblen Strömung im chemischen Nichtgleichgewicht unter Vernachlässigung äußerer Kräfte wird durch die inhomogenen NSGen beschrieben (hier: räumlich d-dimensional):

$$\begin{array}{rcl} \boldsymbol{u}_{,t} & + & \text{div } \left(\boldsymbol{F}_1^I(\boldsymbol{u}), \ldots, \boldsymbol{F}_d^I(\boldsymbol{u})\right) \\ & - & \text{div } \left(\boldsymbol{F}_1^V(\boldsymbol{u}, \boldsymbol{u}_{,x_1}, \ldots, \boldsymbol{u}_{,x_d}), \ldots, \boldsymbol{F}_d^V(\boldsymbol{u}, \boldsymbol{u}_{,x_1}, \ldots, \boldsymbol{u}_{,x_d})\right) = \boldsymbol{S}(\boldsymbol{u}). \end{array} \tag{1}$$

Die Notation $(\cdot)_{,x_i}$ bedeutet dabei die partielle Ableitung der Größe $(\cdot)$ nach der Komponente x_i. Die Größen sind wie folgt definiert:

- Vektor der Bilanzgrößen:

$$\boldsymbol{u} := (\rho_1, \ldots, \rho_K, \rho\boldsymbol{v}, \rho E)^T \tag{2}$$

- Quellterm der chemischen Reaktionen:

$$\begin{array}{rcl} \boldsymbol{S}(\boldsymbol{u}) & := & (s_1, \ldots, s_K, 0, \ldots, 0)^T \\ s_i & = & s_i(\rho_1, \ldots, \rho_K, T) \end{array} \tag{3}$$

- reibungsfreier Fluß in Richtung x_i:

$$\begin{array}{rcl} F_i^I(\boldsymbol{u}) & := & (\rho_1 v_i, \ldots, \rho_K v_i, \rho\boldsymbol{v}v_i + p\boldsymbol{e}_i, \rho v_i H)^T \\ \boldsymbol{e}_i & = & i\text{–ter Einheitsvektor des } R^{K+d+1} \end{array} \tag{4}$$

- viskoser Fluß in Richtung x_i:

$$F_i^V(\boldsymbol{u}, \boldsymbol{u}_{,x_1}, \ldots, \boldsymbol{u}_{,x_d}) := (-q_{1,i}^D, \ldots, -q_{K,i}^D, (\boldsymbol{\tau})^i, (\boldsymbol{\tau}\boldsymbol{v})_i - q_i)^T \tag{5}$$

- Diffusionsfluß in Richtung x_i (nach [7], S. 516):

$$\begin{array}{rcl} \boldsymbol{q}_i^D & := & \dfrac{c^2}{\rho}\displaystyle\sum_{j\neq i} M_i M_j D_{ij}\boldsymbol{d}_j - D_i^T \frac{1}{T}\,\mathbf{grad}\ T \\ \boldsymbol{d}_i & := & \mathbf{grad}\ \dfrac{c_i}{c} + \left(\dfrac{c_i}{c} - \dfrac{\rho_i}{\rho}\right)\dfrac{1}{p}\,\mathbf{grad}\ p \end{array} \tag{6}$$

- Tensor der Reibungsspannungen:

$$\begin{array}{rcl} \boldsymbol{\tau} & := & \mu(\ \mathbf{grad}\ \boldsymbol{v} + (\ \mathbf{grad}\ \boldsymbol{v})^T - \frac{2}{3}(\ \text{div}\ \boldsymbol{v})\,\mathbf{I}) \\ (\boldsymbol{\tau})^i & = & i\text{-te Spalte von } \boldsymbol{\tau} \\ (\boldsymbol{\tau}\boldsymbol{v})_i & = & i\text{-te Komponente von } \boldsymbol{\tau}\boldsymbol{v} \end{array} \tag{7}$$

- Wärmefluß:

$$\begin{array}{rcl} \boldsymbol{q} := -\lambda\ \mathbf{grad}\ T & + & \displaystyle\sum_i h_i \boldsymbol{q}_i^D \\ & + & \dfrac{RT}{c}\displaystyle\sum_i\sum_{j\neq i} \frac{c_j}{M_i}\frac{D_i^T}{\mathcal{D}_{ij}}\left(\frac{\boldsymbol{q}_i^D}{\rho_i} - \frac{\boldsymbol{q}_j^D}{\rho_j}\right) \end{array} \tag{8}$$

Die physikalischen Größen sind wie folgt definiert:

- Massendichte:

$$\rho := \sum_{i=1}^{K} \rho_i \tag{9}$$

- Massenbruch der Komponenete i:

$$Y_i := \frac{\rho_i}{\rho} \tag{10}$$

- molare Konzentration der Komponente i:

$$c_i := \frac{\rho_i}{M_i} \tag{11}$$

- Summe der molaren Konzentrationen:

$$c := \sum_{i=1}^{K} c_i \tag{12}$$

- Molenbruch der Komponente i:

$$X_i := \frac{c_i}{c} \tag{13}$$

- massenspezifische innere Energie (kalorische Zustandsgleichung):

$$e := \sum_{i=1}^{K} Y_i \left(e_i(T) + e_i^0 \right) \tag{14}$$

- massenspezifische Totalenergie:

$$E := e + \frac{1}{2}\boldsymbol{v}^2 \tag{15}$$

- Druck (thermische Zustandsgleichung):

$$p := \rho RT \sum_{i=1}^{K} Y_i / M_i \tag{16}$$

- massenspezifische Enthalpie der Komponente i:

$$h_i := e_i(T) + \frac{RT}{M_i} \tag{17}$$

- massenspezifische Enthalpie:

$$h := \sum_{i=1}^{K} Y_i \left(h_i + e_i^0 \right) = e + p/\rho \tag{18}$$

- massenspezifische Totalenthalpie:

$$H := E + p/\rho = h + \frac{1}{2}\boldsymbol{v}^2 \tag{19}$$

Die Gleichung (1) wird vervollständigt durch die Transport- und Diffusionskoeffizienten:

- Viskositätskoeffizient des Gasgemisches:

$$\mu = \mu(T, X_1, \ldots, X_K) \tag{20}$$

- Wärmeleitungskoeffizient des Gasgemisches:

$$\lambda = \lambda(T, X_1, \ldots, X_K) \tag{21}$$

- thermischer Diffusionskoeffizient der Komponente i:

$$D_i^T = D_i^T(T, p, X_1, \ldots, X_K) \tag{22}$$

- binärer Diffusionskoeffizient der Komponenten i und j:

$$\mathcal{D}_{ij} = \mathcal{D}_{ij}(T, p) \tag{23}$$

- Koeffizient der Mehrkomponentendiffusion der Komponenten i und j:

$$D_{ij} = D_{ij}(T, p, X_1, \ldots, X_K) \tag{24}$$

Die NSGen (1) sind hier für den räumlichen d-dimensionalen Fall dargestellt. Sie wurden jedoch nur für $d = 2$ implementiert. Diese können auch für den Fall einer rotationssymmetrischen Strömung (hier: Rotationsachse $x_2 = 0$) verwendet werden. Aus der Rotationssymmetrie, d.h.

$$\boldsymbol{u}_{,x_3} = \boldsymbol{0} \quad \text{und} \quad v_3 = 0, \tag{25}$$

können die räumlich dreidimensionalen NSGen (1) überführt werden in

$$\begin{aligned} \boldsymbol{u}_{,t} \;+\; \operatorname{div}\left(\boldsymbol{F}_1^I(\boldsymbol{u}), \boldsymbol{F}_2^I(\boldsymbol{u})\right) - \operatorname{div}\left(\boldsymbol{F}_1^V(\boldsymbol{u}, \boldsymbol{u}_{,x_1}, \boldsymbol{u}_{,x_2}), \boldsymbol{F}_2^V(\boldsymbol{u}, \boldsymbol{u}_{,x_1}, \boldsymbol{u}_{,x_2})\right) \\ = \boldsymbol{S}(\boldsymbol{u}) - \sigma\left[\boldsymbol{R}^I(\boldsymbol{u}) - \boldsymbol{R}^V(\boldsymbol{u}, \boldsymbol{u}_{,x_1}, \boldsymbol{u}_{,x_2})\right] \Big/ x_2 = 0\,, \end{aligned} \tag{26}$$

d.h. die zweidimensionalen Gleichungen müssen um einen zusätzlichen Quellterm erweitert werden:

$$\boldsymbol{R}^I = \begin{pmatrix} F_{2,1}^I \\ \vdots \\ F_{2,K}^I \\ \rho v_1 v_2 \\ \rho v_2 v_2 \\ F_{2,K+3}^I \end{pmatrix} \tag{27}$$

$$\boldsymbol{R}^V = \begin{pmatrix} F^V_{2,1} \\ \vdots \\ F^V_{2,K} \\ \mu(v_{2,x_1} + v_{1,x_2}) - \frac{2}{3}(v_1 v_2)_{,x_1} \\ \left[2\mu\left(\frac{v_2}{x_2}\right)_{,x_2} - \frac{2}{3}\left(\frac{\mu v_2}{x_2}\right)_{,x_2}\right] \Big/ x_2 \\ F^V_{2,K+3} - \frac{2}{3}\left[(\mu v_1 v_2)_{,x_1} + (\mu v_2 v_2)_{,x_2}\right] \end{pmatrix}, \tag{28}$$

$$\sigma = \begin{cases} 1, \text{ rotationssymmetrische Strömung} \\ 0, \text{ ebene Strömung} \end{cases}.$$

3 Numerische Methoden

Im Rahmen des Forschungsprojektes wurden die Gleichungen (26) diskretisiert. Dazu wurden folgende Techniken benutzt:

- Operatorsplitting (Quellterm der chemischen Produktionsraten)
- Dimensionssplitting
- Finite–Volumen–Technik
- explizites Shock–Capturing–Verfahren (reibungsfreier Anteil der NSGen)
- zentrale Differenzen (viskoser Term der NSGen, Quellterm der Rotationssymmetrie).

Zu jedem expliziten Zeitschritt wird zunächst der Quellterm $\boldsymbol{S}(\boldsymbol{u})$ der chemischen Produktionsraten von den restlichen Gleichungen getrennt. Dies führt auf ein System von GDGen und ein System von PDGen, die sukzessive gelöst werden:

$$\text{Schritt 1:} \quad \frac{d\boldsymbol{u}^1}{dt} = \boldsymbol{S}(\boldsymbol{u}^1) \quad , \; \boldsymbol{u}^1(\boldsymbol{x}, t_n) = \boldsymbol{u}(\boldsymbol{x}, t_n) \tag{29}$$

$$\begin{aligned} \text{Schritt 2:} \quad \boldsymbol{u}^2_{,t} \; &+ \; \text{div} \left(\boldsymbol{F}^I_1(\boldsymbol{u}^2), \boldsymbol{F}^I_2(\boldsymbol{u}^2) \right) \\ &- \; \text{div} \left(\boldsymbol{F}^V_1(\boldsymbol{u}^2, \boldsymbol{u}^2_{,x_1}, \boldsymbol{u}^2_{,x_2}), \boldsymbol{F}^V_2(\boldsymbol{u}^2, \boldsymbol{u}^2_{,x_1}, \boldsymbol{u}^2_{,x_2}) \right) \\ &+ \; \sigma \left[\boldsymbol{R}^I(\boldsymbol{u}^2) - \boldsymbol{R}^V(\boldsymbol{u}^2, \boldsymbol{u}^2_{,x_1}, \boldsymbol{u}^2_{,x_2}) \right] \Big/ x_2 = 0 \\ \boldsymbol{u}^2(\boldsymbol{x}, t_n) &= \boldsymbol{u}^1(\boldsymbol{x}, t_{n+1}) \; . \end{aligned} \tag{30}$$

Das System (29) geht mit (2) und (3) über in

$$\begin{aligned} \text{a)} \quad \frac{d\rho_i}{dt} &= s_i(\rho_i, \ldots, \rho_N, T), \quad \rho_i(t_n) = \rho_i^n \qquad , \; i = 1, \ldots, K \\ \text{b)} \quad T &= T(e) \overset{(15)}{=} T\left(e(\rho_1, \ldots, \rho_N, (\rho \boldsymbol{v})^n, (\rho E)^n)\right) \; . \end{aligned} \tag{31}$$

Bei der Diskretisierung von (31) ist darauf zu achten, daß die Atomerhaltung lokal gewährleistet ist (siehe [13], S. 18), d.h. es soll keine elementare Entmischung stattfinden. Diese zusätzlichen algebraischen Gleichungen werden dazu verwendet, um das System (31a) entsprechend zu reduzieren. Das reduzierte System wurde hier mittels eines expliziten Eulerschrittes approximiert (siehe [13], S. 64).
Da die Flüsse $\boldsymbol{F}^I_i$ und $\boldsymbol{F}^V_i$ zwei unterschiedliche physikalische Effekte beschreiben – Advektion bzw. dissipative Effekte (Diffusion, Reibung, Wärmeleitung) – und im Sinne der kinetischen Gastheorie von unterschiedlicher Ordnung sind (siehe Chapman-Enskog Entwicklung, [7]), wurden diese unterschiedlich diskretisiert. Dies entspricht einem weiteren Operatorsplitting.
Da die Rotationssymmetrie optional ist, wird auch der dadurch bedingte Quellterm – der bei einer $3D$-Rechnung nicht auftritt – von den übrigen Termen separiert. Die Gleichung (30) wird somit in drei voneinander unabhängige Systeme aufgespalten:

$$\text{Schritt 2.1:} \quad \begin{aligned} &\boldsymbol{u}^{21}_{,t} + \text{ div } \left(\boldsymbol{F}^I_1(\boldsymbol{u}^{21}), \boldsymbol{F}^I_2(\boldsymbol{u}^{21})\right) = 0 \ , \\ &\boldsymbol{u}^{21}(\boldsymbol{x}, t_n) = \boldsymbol{u}^1(\boldsymbol{x}, t_{n+1}) \ ; \end{aligned} \tag{32}$$

$$\text{Schritt 2.2:} \quad \begin{aligned} &\boldsymbol{u}^{22}_{,t} - \text{ div } \left(\boldsymbol{F}^V_1(\boldsymbol{u}^{22}, \boldsymbol{u}^{22}_{,x_1}, \boldsymbol{u}^{22}_{,x_2}), \boldsymbol{F}^V_2(\boldsymbol{u}^{22}, \boldsymbol{u}^{22}_{,x_1}, \boldsymbol{u}^{22}_{,x_2})\right) = 0 \ , \\ &\boldsymbol{u}^{22}(\boldsymbol{x}, t_n) = \boldsymbol{u}^1(\boldsymbol{x}, t_{n+1}) \ ; \end{aligned} \tag{33}$$

$$\text{Schritt 2.3:} \quad \begin{aligned} &\boldsymbol{u}^{23}_{,t} + \sigma \left[\boldsymbol{R}^I(\boldsymbol{u}^{23}) - \boldsymbol{R}^V(\boldsymbol{u}^{23}, \boldsymbol{u}^{23}_{,x_1}, \boldsymbol{u}^{23}_{,x_2})\right] / x_2 = 0 \ , \\ &\boldsymbol{u}^{23}(\boldsymbol{x}, t_n) = \boldsymbol{u}^1(\boldsymbol{x}, t_{n+1}) \ . \end{aligned} \tag{34}$$

Diese werden in einem Zeitschritt simultan gelöst, d.h.

$$\boldsymbol{u}^2_{,t} = \boldsymbol{u}^{21}_{,t} + \boldsymbol{u}^{22}_{,t} + \boldsymbol{u}^{23}_{,t} \ . \tag{35}$$

Die Systeme (32), (33) und (34) werden mittels der Finiten-Volumen-Technik in Integralgleichungen überführt. Dazu werden sie über ein räumlich–zeitliches Kontinuum $[t_n, t_{n+1}] \times V$, $V \subset \mathrm{I\!R}^2$ integriert und der Satz von Gauß angewendet. Daraus resultieren die Gleichungen

$$\begin{aligned} \text{a)} \quad \widehat{\boldsymbol{u}^{21}}(\boldsymbol{x}_M, t_{n+1}) &= \widehat{\boldsymbol{u}^{21}}(\boldsymbol{x}_M, t) - \frac{\Delta t}{|V|} \widehat{\boldsymbol{F}^I}_{\boldsymbol{N},\partial V}(\boldsymbol{u}^{21}) \\ \text{b)} \quad \widehat{\boldsymbol{u}^{22}}(\boldsymbol{x}_M, t_{n+1}) &= \widehat{\boldsymbol{u}^{22}}(\boldsymbol{x}_M, t) + \frac{\Delta t}{|V|} \widehat{\boldsymbol{F}^V}_{\boldsymbol{N},\partial V}(\boldsymbol{u}^{22}) \\ \text{c)} \quad \widehat{\boldsymbol{u}^{23}}(\boldsymbol{x}_M, t_{n+1}) &= \widehat{\boldsymbol{u}^{23}}(\boldsymbol{x}_M, t) - \sigma \Delta t \left[\widehat{\boldsymbol{R}^I}(\boldsymbol{u}^{23}) - \widehat{\boldsymbol{R}^V}(\boldsymbol{u}^{23})\right] \ . \end{aligned} \tag{36}$$

mit den Notationen

- räumlicher Mittelwert von $\boldsymbol{u}$:

$$\widehat{\boldsymbol{u}}(\boldsymbol{x}_{M,t}) := \frac{1}{|V|} \iint\limits_V \boldsymbol{u}(\boldsymbol{x}, t) \, dV \tag{37}$$

- reibungsfreier Fluß in Richtung der äußeren Normalen $\boldsymbol{N}$ zum Rand ∂V:

$$\boldsymbol{F}^I_{\boldsymbol{N},\partial V}\left(\boldsymbol{u}^{21}(\boldsymbol{x}, t)\right) := \left(\boldsymbol{F}^I_1\left(\boldsymbol{u}^{21}(\boldsymbol{x}, t)\right), \boldsymbol{F}^I_2\left(\boldsymbol{u}^{21}(\boldsymbol{x}, t)\right)\right) \cdot \boldsymbol{N}(\boldsymbol{x}) \tag{38}$$

- viskoser Fluß in Richtung der äußeren Normalen $\boldsymbol{N}$ zum Rand ∂V:

$$\begin{aligned} &\boldsymbol{F}^V_{\boldsymbol{N},\partial V}\left(\boldsymbol{u}^{22}(\boldsymbol{x}, t), \boldsymbol{u}^{22}_{,x_1}(\boldsymbol{x}, t), \boldsymbol{u}^{22}_{,x_2}(\boldsymbol{x}, t)\right) := \\ &\qquad \left(\boldsymbol{F}^V_1\left(\boldsymbol{u}^{22}(\boldsymbol{x}, t), \boldsymbol{u}^{22}_{,x_1}(\boldsymbol{x}, t), \boldsymbol{u}^{22}_{,x_2}(\boldsymbol{x}, t)\right), \right. \\ &\qquad\qquad \left. \boldsymbol{F}^V_2\left(\boldsymbol{u}^{22}(\boldsymbol{x}, t), \boldsymbol{u}^{22}_{,x_1}(\boldsymbol{x}, t), \boldsymbol{u}^{22}_{,x_2}(\boldsymbol{x}, t)\right)\right) \cdot \boldsymbol{N}(\boldsymbol{x}) \end{aligned} \tag{39}$$

- zeitlicher Mittelwert des reibungsfreien Flusses über den Rand ∂V:

$$\widehat{\boldsymbol{F}^I}_{\boldsymbol{N},\partial V}(\boldsymbol{u}^{21}) := \frac{1}{\Delta t} \int\limits_{t_n}^{t_{n+1}} \int\limits_{\partial V} \boldsymbol{F}^I_{\boldsymbol{N},\partial V}\left(\boldsymbol{u}^{21}(\boldsymbol{x}(s), t)\right) \, ds \, dt \tag{40}$$

- zeitlicher Mittelwert des viskosen Flusses über den Rand ∂V:

$$\widehat{\boldsymbol{F}^V}_{\boldsymbol{N},\partial V}(\boldsymbol{u}^{22}) := \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \int_{\partial V} \boldsymbol{F}^V_{\boldsymbol{N},\partial V} \left(\boldsymbol{u}^{22}(\boldsymbol{x}(s),t)\right) \, ds \, dt \tag{41}$$

- räumlich-zeitlicher Mittelwert des Quellterms $\boldsymbol{R}^I$:

$$\widehat{\boldsymbol{R}^I}(\boldsymbol{u}^{23}) := \frac{1}{\Delta t \cdot |V|} \int_{t_n}^{t_{n+1}} \iint_{V} \boldsymbol{R}^I \left(\boldsymbol{u}^{23}(\boldsymbol{x},t)\right) \, dV \, dt \tag{42}$$

- räumlich-zeitlicher Mittelwert des Quellterms $\boldsymbol{R}^V$:

$$\widehat{\boldsymbol{R}^V}(\boldsymbol{u}^{23}) := \frac{1}{\Delta t \cdot |V|} \int_{t_n}^{t_{n+1}} \iint_{V} \boldsymbol{R}^V \left(\boldsymbol{u}^{23}(\boldsymbol{x},t), \boldsymbol{u}^{23}_{,x_1}(\boldsymbol{x},t), \boldsymbol{u}^{23}_{,x_2}(\boldsymbol{x},t)\right) \, dV \, dt \; . \tag{43}$$

Dabei gibt $\Delta t := t_{n+1} - t_n > 0$ die Länge des Zeitintervalls, $|V|$ die Fläche von V und $\boldsymbol{x}_M \in V$ den Schwerpunkt von V an.
Aus (35) folgt dann die Approximation für (30)

$$\begin{aligned} \widehat{\boldsymbol{u}^2}(\boldsymbol{x}_M, t_{n+1}) \;=\; \widehat{\boldsymbol{u}^2}(\boldsymbol{x}_M, t_n) \;&-\; \frac{\Delta t}{|V|} \left[\widehat{\boldsymbol{F}^I}_{\boldsymbol{N},\partial V}(\boldsymbol{u}^{21}) - \widehat{\boldsymbol{F}^V}_{\boldsymbol{N},\partial V}(\boldsymbol{u}^{22})\right] \\ &-\; \sigma \cdot \Delta t \left[\widehat{\boldsymbol{R}^I}(\boldsymbol{u}^{23}) - \widehat{\boldsymbol{R}^V}(\boldsymbol{u}^{23})\right] \; . \end{aligned} \tag{44}$$

Wendet man die Finite–Volumen–Technik direkt auf die Gleichung (30) an, so ergibt sich (44) mit $\boldsymbol{u}^{21} = \boldsymbol{u}^{22} = \boldsymbol{u}^{23} = \boldsymbol{u}^2$. Somit bewirkt das Aufspalten von (30) in (36a), (36b) und (36c) nur, daß die Flüsse an unterschiedlichen Stellen ausgewertet werden.
Zur Diskretisierung der Gleichung (32) — diese entsprechen den Euler–Gleichungen — wird ein explizites Shock-Capturing Verfahren verwendet. Es handelt sich dabei um das Essentially–Non–Oscillatory–Verfahren (ENO) von Harten u.a. [1]. Dieses basiert darauf, daß die Gleichung (32) über jede Zelle des diskretisierten Strömungsgebietes integriert bzw. (36a) berechnet wird.
In der hier durchgeführten numerischen Simulation wird der Strömungsbereich in allgemeine Vierecke unterteilt. Diese sollen so gewählt sein, daß die Seiten eines Viereckes möglichst senkrecht aufeinander stehen und etwa gleiche Seitenlängen haben. Ein Ausschnitt eines derart diskretisierten Gebietes ist in Abb. 1 dargestellt.
Dabei sind die Punkte $\boldsymbol{P}_i^{j,k}$ die Mittelpunkte der vier Seiten $\Gamma_i^{j,k}$ der Zelle $V_{j,k}$ mit der konstanten äußeren Normalen $\boldsymbol{N}_i^{j,k}$. Man beachte, daß die äußeren Normalen zu einer Seite, die zwei benachbarten Zellen angehören, sich nur durch das Vorzeichen unterscheiden, d.h.

$$N_1^{j,k} = -N_3^{j,k+1} \quad \text{und} \quad N_2^{j,k} = -N_4^{j-1,k} \; . \tag{45}$$

Nun wird (36a) für jede Zelle $V_{j,k}$ berechnet, d.h. die Mittelwerte $\widehat{\boldsymbol{u}^{21}}$ zum neuen Zeitpunkt ergeben sich aus denen des alten Zeitniveaus und der Differenz der Flüsse, d.h. dessen,

was in dem Zeitintervall über die Zellgrenzen fließt. Zur Auswertung der Flüsse wird aber $\boldsymbol{u}^{21}$ und nicht der Mittelwert benötigt, d.h. aus den diskreten Mittelwerten $\widehat{\boldsymbol{u}^{21}}$ muß $\boldsymbol{u}^{21}$ rekonstruiert werden.
Somit erhält man für das ENO-Verfahren folgende Schrittfolge eines expliziten Zeitschrittes auf obigem Strömungsgebiet:

a) **Initialisierung:**
$\boldsymbol{v}_{j,k}^n \approx \widehat{\boldsymbol{u}^{21}}(\boldsymbol{x}_{j,k}, t_n)$;

b) **Rekonstruktion von Ordnung N:**
$\boldsymbol{R}^N(\boldsymbol{x}, \boldsymbol{v}_{j,k}^n) = \boldsymbol{u}^{21}(\boldsymbol{x}, t_n) + \mathrm{O}\left(||\Delta x||^N\right)$,
$\boldsymbol{R}_{j,k}^N := \boldsymbol{R}^N|_{V_{j,k}} \in \mathcal{P}_2^N := \left\{ x_1^{l_1} x_2^{l_2} \;\middle|\; 0 \le l_1 + l_2 \le N \right\}$;

c) **Numerischer Fluß:**
$$\widehat{\boldsymbol{F}^I}_{i,j,k} := \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \int_{\Gamma_I^{j,k}} \boldsymbol{F}^I_{\boldsymbol{N}_i^{j,k}, \Gamma_i^{j,k}}(\boldsymbol{w}(\boldsymbol{x}(s), t))\, ds\, dt \tag{46}$$
$\boldsymbol{F}^I_{\boldsymbol{N}_i^{j,k}, \Gamma_i^{j,k}} := (\boldsymbol{F}_1^I, \boldsymbol{F}_2^I) \cdot \boldsymbol{N}_i^{j,k}$
$\boldsymbol{w}$ Lösung von (32) zu den Anfangsbedingungen $\boldsymbol{w}(\boldsymbol{x}, t_n) = \boldsymbol{R}^N(\boldsymbol{x}, \boldsymbol{v}_{j,k}^n)$;

d) **Mittelwertbildung:**
$$\begin{aligned} \boldsymbol{v}_{j,k}^{n+1} &= \boldsymbol{v}_{j,k}^n - \frac{\Delta t}{|V_{j,k}|} \sum_{i=1}^4 \widehat{\boldsymbol{F}^I}_{i,j,k} \\ &\overset{(45)}{=} \boldsymbol{v}_{j,k}^n - \frac{\Delta t}{|V_{j,k}|} \left\{ \left[\widehat{\boldsymbol{F}^I}_{3,j,k} - \widehat{\boldsymbol{F}^I}_{1,j,k-1} \right] + \left[\widehat{\boldsymbol{F}^I}_{2,j,k} - \widehat{\boldsymbol{F}^I}_{4,j-1,k} \right] \right\} . \end{aligned}$$

Die Berechnung der Rekonstruktionspolynome $\boldsymbol{R}_{j,k}^N$ und der Flüsse $\widehat{\boldsymbol{F}^I}_{i,j,k}$ wird nun mittels der Dimensions-Splitting-Technik auf den räumlich eindimensionalen Fall zurückgeführt. Dazu wird angenommen, daß die Größe $\boldsymbol{u}^{21}$ entlang jeder Zellkante konstant ist. Dadurch muß der Fluß über den Zellrand nur in den Zellrandmittelpunkten ausgewertet werden. Die Zeitintegration wird durch die Mittelpunktsregel approximiert:

$$\widehat{\boldsymbol{F}^I}_{i,j,k} \approx \left|\Gamma_i^{j,k}\right| \; \boldsymbol{F}^I_{\boldsymbol{N}_I^{j,k}, \Gamma_I^{j,k}} \left(\boldsymbol{w}(\boldsymbol{P}_i^{j,k}, t_n + \Delta t/2) \right) . \tag{47}$$

Zur Berechnung von $\boldsymbol{w}$ an der Stelle $\boldsymbol{P}_i^{j,k}$ reicht aufgrund des Dimensionssplittings ebenfalls eine eindimensionale Rekonstruktionstechnik. Durch Verbinden der Punkte $\boldsymbol{P}_2^{j,k}$ und $\boldsymbol{P}_4^{j,k}$ bzw. $\boldsymbol{P}_1^{j,k}$ und $\boldsymbol{P}_3^{j,k}$ für alle Zellen $V_{j,k}$ ergeben sich einparametrige Kurven $\boldsymbol{z}_k(\zeta)$ bzw. $\boldsymbol{z}_j(\eta)$ entlang denen $\boldsymbol{u}^{21}$ mittels der RP-Methode (siehe [1]) rekonstruiert wird (siehe Abb. 2).
In den Stellen $\boldsymbol{P}_i^{j,k}$ haben die Rekonstruktionspolynome $\boldsymbol{R}_j^N(\boldsymbol{z}_k(\zeta))$ bzw. $\boldsymbol{R}_k^N(\boldsymbol{z}_j(\eta))$ Unstetigkeiten. Um den physikalisch sinnvollen Zustand zu bestimmen, werden eindimensionale Riemannprobleme in Richtung der äußeren Normale zum Zellrand gelöst:

$$\boldsymbol{w}_t(\xi,t) + \Big((\boldsymbol{F}_1^I(\boldsymbol{w}(\xi,t)), \boldsymbol{F}_2^I(\boldsymbol{w}(\xi,t))) \cdot \boldsymbol{N}_i^{j,k}\Big)_{,\xi} = 0\,, \quad \xi = \boldsymbol{N}_i^{j,k} \cdot \boldsymbol{x}$$

$$\boldsymbol{w}(\xi, t_n + \tfrac{\Delta t}{2}) = \begin{cases} \boldsymbol{w}^N\left(\xi_i - 0, t_n + \dfrac{\Delta t}{2}\right), & \xi < \xi_i \\ \boldsymbol{w}^N\left(\xi_i + 0, t_n + \dfrac{\Delta t}{2}\right), & \xi > \xi_i \end{cases} \tag{48}$$

zu den Zuständen

$$\boldsymbol{w}(\xi,t_n) = \boldsymbol{R}_j^N(\boldsymbol{z}_k(\xi)) \quad \text{bzw.} \quad \boldsymbol{w}(\xi,t_n) = \boldsymbol{R}_k^N(\boldsymbol{z}_j(\xi))\,. \tag{49}$$

Zur Lösung des Riemannproblems (48) wird der approximative Riemannlöser von Liu und Vinokur [11] für ein Gasgemisch, das sich im chemischen Nichtgleichgewicht befindet (siehe [13], S. 55 ff.) adaptiert. Er wird jedoch nicht explizit der Zwischenzustand sondern der approximative Fluß ausgerechnet.
Bei der Berechnung der Flüsse am Rand des Strömungsgebietes werden keine Riemannprobleme gelöst, sondern die Randwerte explizit in den Fluß eingesetzt.
Die Diskretisierung der Eulergleichungen (32) ist damit beschrieben. Sie stellt wohl den aufwendigsten Teil des Algorithmus' dar.
In der Berechnung von (36b) wird der Mittelwert des viskosen Flusses durch die Zellmittelwerte auf dem alten Zeitniveau approximiert. Analog zu (47) ergibt sich

$$\widehat{\boldsymbol{F}^V}_{i,j,k} \approx \left|\Gamma_i^{j,k}\right| \; \boldsymbol{F}^V_{\boldsymbol{N}_I^{j,k},\Gamma_i^{j,k}}(\boldsymbol{u}^{22}(\boldsymbol{P}_i,t_n), \boldsymbol{u}^{22}_{,x_1}(\boldsymbol{P}_i,t_n), \boldsymbol{u}^{22}_{,x_2}(\boldsymbol{P}_i,t_n))\,. \tag{50}$$

Um $\boldsymbol{F}^V_{\boldsymbol{N}_i^{j,k},\Gamma_i^{j,k}}$ im Punkt $\boldsymbol{P}_i$ auswerten zu können, werden sowohl die Größen $\boldsymbol{u}^{22}(\boldsymbol{P}_i)$ als auch die Ableitungen des Geschwindigkeitsvektors $\boldsymbol{v}_\xi(\boldsymbol{P}_i)$ und $\boldsymbol{v}_\eta(\boldsymbol{P}_i)$ benötigt, wobei (x_1,x_2) die kartesischen und (ξ,η) beliebige krummlinige Koordinaten sind (vgl. (39)). Für die Ableitungen gilt lokal:

$$\textbf{grad}\ \boldsymbol{v}(x_1,x_2) \approx \begin{pmatrix} N_1 & -N_2 \\ N_2 & N_1 \end{pmatrix} \textbf{grad}\ \boldsymbol{v}(\xi,\eta)\,. \tag{51}$$

Diese lassen sich durch zentrale Differenzen

$$\boldsymbol{v}_\xi(\boldsymbol{P}_i) \approx \frac{\boldsymbol{v}(\boldsymbol{P}_O) - \boldsymbol{v}(\boldsymbol{P}_W)}{||\boldsymbol{P}_W - \boldsymbol{P}_O||} \quad \text{bzw.} \quad \boldsymbol{v}_\eta(\boldsymbol{P}_i) \approx \frac{\boldsymbol{v}(\boldsymbol{P}_N) - \boldsymbol{v}(\boldsymbol{P}_S)}{||\boldsymbol{P}_N - \boldsymbol{P}_S||} \tag{52}$$

mit

$$\begin{aligned} \boldsymbol{v}(\boldsymbol{P}_N) &= \frac{1}{4}\Big(\boldsymbol{v}(\boldsymbol{P}_{NW}) + \boldsymbol{v}(\boldsymbol{P}_W) + \boldsymbol{v}(\boldsymbol{P}_O) + \boldsymbol{v}(\boldsymbol{P}_{NO})\Big) \\ \boldsymbol{v}(\boldsymbol{P}_S) &= \frac{1}{4}\Big(\boldsymbol{v}(\boldsymbol{P}_{SO}) + \boldsymbol{v}(\boldsymbol{P}_O) + \boldsymbol{v}(\boldsymbol{P}_W) + \boldsymbol{v}(\boldsymbol{P}_{SW})\Big) \end{aligned}$$

diskretisieren (vgl. Abb. 3).
Analog gilt dies auch für alle anderen Größen und deren Ableitungen bzgl. ξ und η. Die Werte in den Zellmittelpunkten sind gerade die Werte auf dem alten Zeitniveau.
An der Körperwand muß man zwei Fälle unterscheiden (vgl. Abb. 4, 5):

Im Fall von Abb. 4 müssen in den Punkten $\boldsymbol{P}_{SW}$ und $\boldsymbol{P}_{SO}$ die Wanddaten verwendet werden. Durch Mittellung dieser Werte erhält man dann eine Diskretisierung für den Punkt $\boldsymbol{P}_S$. Bei der Diskretisierung des Flusses normal zur Körperwand (5) müssen in den Punkten $\boldsymbol{P}_{SW}, \boldsymbol{P}_W = \boldsymbol{P}_i$ und $\boldsymbol{P}_{NW}$ die Wanddaten verwendet werden. Durch Mittellung erhält man dann wiederum approximative Werte in den Punkten $\boldsymbol{P}_N$ und $\boldsymbol{P}_S$. Beachte, daß in diesem Fall der Fluß eine andere Darstellung hat, da hier eventuell die Transportvorgänge an der Wand berücksichtigt werden müssen.
In Schritt 2.3 (vgl. (34)) müssen im Fall der Rotationssymmetrie ($\sigma = 1$) die Terme (42) und (43) diskretisiert werden. Um den Aufwand möglichst gering zu halten, soll dazu auf die schon vorhandenen Informationen aus den Schritten 2.1 und 2.2 (vgl. (32), (33)) zurückgegriffen werden. Dazu werden zum alten Zeitpunkt die räumlichen Integrale approximiert. Für den reibungsfreien Anteil wird dieses im Zellmittelpunkt ausgewertet, für den viskosen Anteil dagegen in den vier Zellrandmittelpunkten, d.h.

$$\widehat{\boldsymbol{R}^I}_{j,k} = \boldsymbol{R}^I\left(\boldsymbol{u}^{23}(\boldsymbol{x}_{j,k}, t_n)\right) \quad \text{bzw.} \tag{53}$$

$$\widehat{\boldsymbol{R}^V}_{j,k} = \frac{1}{4}\sum_{i=1}^{4} \boldsymbol{R}^V\left(\boldsymbol{u}^{23}(\boldsymbol{P}_i^{j,k}, t_n), \boldsymbol{u}^{23}_{,x_1}(\boldsymbol{P}_i^{j,k}, t_n), \boldsymbol{u}^{23}_{,x_2}(\boldsymbol{P}_i^{j,k}, t_n)\right) . \tag{54}$$

Abschließend sei noch bemerkt, daß die Schritte 2.1 – 2.3 simultan berechnet werden und nicht sukzessive. Wahlweise kann auch auf das Separieren der chemischen Produktionsterme verzichtet werden und diese Schritte auch simultan zu 2.1 – 2.3 durchgeführt werden. Dies ist für instationäre Probleme sinnvoll. Die Wahl des physikalischen und somit auch numerischen Modells erfolgt über Flags.

4 Programmstruktur

4.1 NSCNG2 (Hauptprogramm)

Im vorausgegangenen Kapitel wurden die numerischen Methoden ausführlich beschrieben. Die Diskretisierung der einzelnen Schritte (29), (32), (33) und (34) wurde dabei im wesentlichen in der Reihenfolge vorgenommen, in der sie auch im Programm ablaufen. Schwerpunkt des Programms bildet der Eulerlöser zur Diskretisierung von (32).
Die Berechnung von (29), (33) und (34) geschieht optional. Die Programmstruktur ist in einem Flußdiagramm (Abb. 11) dargestellt.
Die Unterroutinen zu den einzelnen Schritten befinden sich in den gleichnamigen Dateien.
Die einzelnen Schritte sind

INPUT

Eingabe der strömungs– und diskretisierungsspezifischen Parameter;

INITIAL

Einlesen der Daten und des Gitters;

Initialisierung der charakteristischen Anströmwerte für die durchzuführende Strömungssimulation;

Initialisierung der physikalischen Größen;

Initialisierung der Anströmwerte;

Initialisierung aller gitterspezifischen Größen;

Initialisierung der rekonstruktionsspezifischen Größen;

CHEMIE (optional, falls chemisches Nichtgleichgewicht modelliert werden soll)

Berechnung der Diskretisierung von (29) bzw. (31);

CFD

Berechnung der Strömung;

Berechnung der Bilanzgrößen (2) auf dem neuen Zeitlevel;

BOUND

Berechnung der Bilanzgrößen (2) in den Randzellen des Gitters;

OUTPUT

Ausgabe der neuen Mittelwerte nach **NSTEPS** durchgeführten Zeitschritten.

4.2 INPUT

Die Programmeingabe der zur Laufzeit benötigten Parameter und Dateien erfolgt in der Routine **INPUT**. Diese wird zu Programmbeginn aufgerufen. Die Eingabe erfolgt über eine Datei, deren Name zum Programmstart eingegeben wird. Sie muß folgendermaßen strukturiert sein:

OFLAG ohne/mit Operatorsplitting (siehe Kap. 5)

WFLAG adiabate/nicht adiabate Wand (siehe Kap. 5)

KWFLAG voll katalytische/nicht katalytische Wand (siehe Kap. 5)

GFLAG ebene/rotationssymmetrische/achsensymmetrische Strömung (siehe Kap. 5)

NSFLAG ohne/mit Reibung, Diffusion und Wärmeleitung (siehe Kap. 5)

CFLAG inertes Gas/chemisches Nichtgleichgewicht (siehe Kap. 5)

RFLAG(i) Randtyp des Randes Γ_i, $i = 1,4$ (siehe Kap. 5)

KFLAG(i) fester Körperrand oder freier Strömungsrand des Randes Γ_i, $i = 1,4$ (siehe Kap. 5)

PFLAG ohne/mit Überprüfung auf positiven Druck (siehe Kap. 5)

ENINP Dateiname für Tabellierung von $\tilde{e}_i(T) := e_i(T)M_i/R/T$

CVINP Dateiname für Tabellierung von $\tilde{c}_{V,i}(T) := c_{V,i}(T)M_i/R$

MUINP Dateiname für Tabellierung von $\mu_i(T)$

BDINP Dateiname für Tabellierung von $\widetilde{\mathcal{D}}_{ij}(T) := p\mathcal{D}_{ij}$

AKINP Dateiname für Tabellierung von $\tilde{K}_g^i(\ln T) := \ln\left(K_g^i(\ln T)\right)$

POLGRA Rekonstruktionsordnung

NSTEPS Anzahl der zu berechnenden Zeitschritte

DELTAT Zeitschrittweite

DATZWI Dateiname für Zwischenspeicherung der Daten

ZSTEPS Anzahl der Zeitschritte, nach denen die Daten zwischengespeichert werden sollen

SSTEPS Anzahl der Zeitschritte, nach denen überprüft werden soll, ob die Strömung stationär ist

GITTER Dateiname für Gitter

DATINP Dateiname für Eingabedaten

DATOUT Dateiname für Ausgabedaten

REFQUA Dateiname für Anströmungsgrößen

4.3 INITIAL

Nach der Eingabe der strömungsspezifischen Parameter erfolgt die Initialisierung der zur Laufzeit benötigten konstanten Größen. Diese wird über die Routine **INITIAL** gesteuert. Sie erfolgt in der nachstehenden Reihenfolge:

DATINI

Einlesen des Gitters und der Strömungsdaten (Bilanzgrößen), d.h. Vorbelegung der Variablen (siehe Kap. 5.2 und 7.2):

MZAHLX, **MZAHLY**

GRID(j,k,i)	$j=-2,$**MZAHLX**$+1,$	$k=-2,$**MZAHLY**$+1,$	$i=1,2$
MW(j,k,i)	$j=-1,$**MZAHLX**$+1,$	$k=-1,$**MZAHLY**$+1,$	$i=1,$**DIMMAX**
MWRHO(j,k,i)	$j=-1,$**MZAHLX**$+1,$	$k=-1,$**MZAHLY**$+1,$	$i=1,$**IMAX**

PHYINI

Einlesen der Koeffizienten der Tabellierung von $\tilde{e}_i(T), \tilde{c}_{V,i}(T), \mu_i(T)$, $\widetilde{\mathcal{D}}_{ij}(T)$ und $\tilde{K}_g^i(\ln T)$, d.h. Vorbelegung der Variablen (siehe Kap. 7.6):

TMIN, **TMAX**, **DTP**, **JANZ**

W(i), $\quad i=1,$**IMAX**

FENTH(i), $\quad i=1,$**IMAX**

IEA(l,j,k), $\quad l=1,4$, $j=1,$**IMAX** , $k=1,$**JANZ**

CVA(l,j,k), $\quad l=1,4$, $j=1,$**IMAX** , $k=1,$**JANZ**

TMINMU, **TMAXMU**, **DTPMU**, **JANZMU**

MUA(l,j,k), $\quad l=1,5$, $j=1,$**IMAX** , $k=1,$**JANZMU**

TMINBD, **TMAXBD**, **DTPBD**, **JANZBD**

BDA(l,i,j,k), $\quad l=1,5$, $i,j=1,$**IMAX** , $k=1,$**JANZBD**

TMINAK, **TMAXAK**, **DTPAK**, **JANZAK**

AKOEFF(l,j,k), $\quad l=1,4$, $j=1,$**IMAX** , $k=1,$**JANZAK**

REFINI

Einlesen der Anströmungsgrößen (siehe Kap. 7.1):

XREF(i) $\quad i=1,$**IMAX**

RHOREF, **MREF**, **NREF**, **PREF**, **FREF**

TWALL(i) (falls **NSFLAG**$=1$, **WFLAG**$=1$ und **KFLAG**$(i)=1$), $i=1,4$

XWALL(i) (falls **NSFLAG**$=1$, **KWFLAG**$=0$), $i=1,$**IMAX**

NETINI

Initialisierung der gitterabhängigen Größen (siehe Kap. 7.3):

1. Normale in den Zellrandmittelpunkten (**NORMAL**)
   NXI(j,k,i) $j=-1$,**MZAHLX**, $k=-1$,**MZAHLY**+1 $i=1,2$
   NETA(j,k,i) $j=-1$,**MZAHLX**+1, $k=-1$,**MZAHLY**, $i=1,2$
2. Flächeninhalt der einzelnen Zellen (**VOLUME**)
   VOL(j,k) $j=-1$,**MZAHLX**+1, $k=-1$, **MZAHLY**+1
3. Bogenlänge der Zellkanten (**ARCLEN**)
   ARCXI(j,k) $j=-1$,**MZAHLX**+1, $k=0$,**MZAHLY**
   ARCETA(j,k) $j=0$,**MZAHLX**, $k=-1$,**MZAHLY**+1

REKINI

Initialisierung der für die Rekonstruktion benötigten Größen, hier: Berechnung der Stützstellen auf den eindimensionalen parametrisierten Kurven entlang denen rekonstruiert wird (siehe Kap. 7.4):

MPKXI(j,k) $j=-$**NMAX**,**MZAHLX** + **NMAX**, $k=0$,**MZAHLY**

MPKETA(j,k) $j=0$,**MZAHLX**, $k=-$**NMAX**,**MZAHLY** + **NMAX**

4.4 CHEMIE

Soll chemisches Nichtgleichgewicht modelliert werden (**CFLAG**= 1), so müssen die chemischen Produktionsraten berechnet werden. Diese beschreiben die zeitliche Änderung der Partialdichten gemäß der chemischen Produktion. Dies geschieht in der Routine **CHEMIE**. Dazu werden die drei folgenden Schritte ausgeführt:

MWQUAN

Berechnung der Massenbrüche, der Bindungsenergie $\sum Y_i e_i^0$ und der Temperatur zu den Mittelwerten der Zellen im Strömungsgebiet **und** auf dem Rand sowie Abspeichern der alten Erhaltungsgrößen (siehe Kap. 7.2):

MWX(j,k) $j=-1$,**MZAHLX**+1, $k=-1$,**MZAHLY**+1
MWFE(j,k) $j=-1$,**MZAHLX**+1, $k=-1$,**MZAHLY**+1
MWT(j,k) $j=-1$,**MZAHLX**+1, $k=-1$,**MZAHLY**+1
MWALT(j,k) $j=-1$,**MZAHLX**+1, $k=-1$,**MZAHLY**+1

CHEPRO

Berechnung der chemischen Produktionsraten s_i zu den Zellen $\{V_{j,k}\}_{j=0,\mathbf{MZAHLX}}$ entlang der Kurven $k=0$,**MZAHLY**:

QUELLE(j,k,i) $j=0$,**MZAHLX**, $k=0$,**MZAHLY**, $i=1$,**IMAX**

Näheres zu dieser Routine findet man in Kap. 6.2.1.

EULER

Falls der Quellterm der chemischen Produktionsraten von den übrigen Gleichungen gesplittet werden soll (**OFLAG**= 1), dann werden die Partialdichten zu jeder Zelle **im** Strömungsgebiet aufdatiert mittels eines expliziten Eulerschrittes:

MWRHO(j,k,i) $j=0$,**MZAHLX**, $k=0$,**MZAHLY**, $i=1$,**IMAX**

4.5 CFD

Das Kernstück des Programms bildet die Berechnung der Strömung, d.h. die Diskretisierung von (29). Diese wird von der Prozedur **CFD** gesteuert und gliedert sich im wesentlichen in die Berechnung von (33) und (34). Die Struktur ist in einem Flußdiagramm (Abb. 12) dargestellt.

BOUND

Berechnung der Bilanzgrößen (2) in den Randzellen des Gitters;

MWQUAN

Berechnung der physikalischen Größen ($\sum Y_i e_i^0, \rho_i/\rho, T$) aus den Bilanzgrößen zum alten Zeitpunkt $t = t_n$, und Abspeichern der alten Erhaltungsgrößen;

REKON

Berechnung der Rekonstruktion entlang der Kurven in ξ-Richtung – Kurve $\boldsymbol{z}_k(\xi)$ – und in η-Richtung – Kurve $\boldsymbol{z}_j(\eta)$ – über die einparametrige RP-Methode ([13], S. 33 ff.), und Auswertung der Polynome in den Zellrandmittelpunkten;

FLINV

Lösung der Riemannprobleme in den Zellrandmittelpunkten entlang der Kurven $\boldsymbol{z}_k(\xi)$ und $\boldsymbol{z}_j(\eta)$ mittels des Riemannlösers von Liu und Vinokur [11], und Berechnung des reibungsfreien numerischen Flusses (47);

FLVIS (optional, falls viskose Effekte modelliert werden sollen)

Berechnung der viskosen Flüsse (5) und optional – bei einer rotationssymmetrischen Rechnung – des Quellterms (55) über zentrale Differenzen (51), (52);

MWSTEP

Berechnung der Mittelwerte $\widehat{\boldsymbol{u}^{21}}$ (36a), $\widehat{\boldsymbol{u}^{22}}$ (36b) bzw. $\widehat{\boldsymbol{u}^{23}}$ (36c), sowie optional – bei einer rotationssymetrischen Rechnung – die Berechnung des Quellterms (53) und des Mittelwertes $\widehat{\boldsymbol{u}^{23}}$ (36c);

TEST

Überprüfung (optional), ob aus den Mittelwerten auf dem neuen Zeitniveau keine negativen Drücke resultieren und eventuell erforderliche Korrekturmaßnahmen; Überprüfung, ob Strömung stationär ist über zeitliche Änderung der Erhaltungsgrößen $(\rho, \rho\boldsymbol{v}, \rho E)^T$ in einer Zelle;

4.5.1 BOUND

Aufgrund der Verwendung von strukturierten Gittern setzt sich der Rand des Diskretisierungsgebietes aus vier Teilstücken zusammen:

$$
\begin{aligned}
\Gamma_1 &:= \{\boldsymbol{x}(\mathbf{MZAHLX}+1,k)|k=0,\mathbf{MZAHLY}\},\\
\Gamma_2 &:= \{\boldsymbol{x}(j,\mathbf{MZAHLY}+1)|j=0,\mathbf{MZAHLX}\},\\
\Gamma_3 &:= \{\boldsymbol{x}(-1,k)|k=0,\mathbf{MZAHLY}\},\\
\Gamma_4 &:= \{\boldsymbol{x}(j,-1)|j=0,\mathbf{MZAHLX}\}
\end{aligned}
$$

und den vier Endpunkten

$$
\begin{aligned}
E_{12} &:= \boldsymbol{x}(\mathbf{MZAHLX}+1,\mathbf{MZAHLY}+1),\\
E_{23} &:= \boldsymbol{x}(-1,\mathbf{MZAHLY}+1),\\
E_{34} &:= \boldsymbol{x}(-1,-1),\\
E_{41} &:= \boldsymbol{x}(\mathbf{MZAHLX}+1,-1).
\end{aligned}
$$

Da das Verfahren auf dem Dimensionssplitting beruht, wird auf die Endpunkte E_{jk} nur bei der Diskretisierung von (41), (43) zugegriffen.
Bei der Berechnung der Randdaten muß grundsätzlich unterschieden werden zwischen

- einem freien Strömungsrand und
- einer festen Wand.

Da die Modellierung der Ränder wesentlich das zu simulierende Strömungsproblem bestimmt, z.B.

- Staupunktströmung
- Stoßwellenkanal
- Düsenströmung, etc ,

wurde den vier Rändern kein bestimmter Randtyp zugeordnet, sondern der Benutzer kann diese über den Flag **RFLAG**(i), $i = 1,4$, frei wählen. Dadurch können unterschiedliche Problemstellungen mit dem Programm simuliert werden. Der Benutzer kann zwischen den folgenden Randtypen wählen:

HYPIN (hypersonischer Inflow)

hier werden die Anströmdaten als Randdaten verwendet; dies ist nur dann physikalisch sinnvoll, wenn die Strömung hypersonisch ist.

HYPOUT (hypersonischer Outflow)

hier werden die Daten vom Strömungsgebiet auf den Rand fortgesetzt; dies ist nur dann physikalisch sinnvoll, wenn die Strömung hypersonisch ist.

WALLS (Haftbedingung an der Wand)

$\boldsymbol{v}_W = 0$

$T_W = \text{const.}$ (**WFLAG**= 1) oder $\dfrac{\partial T}{\partial \boldsymbol{N}} = 0$ (**WFLAG**= 0)

$\dfrac{\partial p}{\partial \boldsymbol{N}} = 0$ oder $\dfrac{\partial p}{\partial \boldsymbol{N}}$ gemäß [13], S. 72

$\dfrac{\partial Y_i}{\partial \boldsymbol{N}} = 0$ (**KWFLAG**= 1) oder $Y_{i,W} = \text{const.}$ (**KWFLAG**= 0)

WALLV (Gleitbedingung)

Berechnung der Zustände an der Wand über eine isotrope Expansions– bzw. Kompressionswelle derart, daß gilt (vgl. [13], S. 69 ff):

$$v_W^N = 0$$

WALLR (Reflexionsbedingung)

Berechnung der Wanddaten durch Reflexion des Massenflusses, d.h. (vgl. [13], S. 71):

$$v_W^N = -\boldsymbol{v}_W \cdot \boldsymbol{N}$$

SYMROT (Rotationsachse)

Spiegelung der Daten an der Rotationsachse (hier nur Γ_3, d.h. $x_2 = 0$)

$$\begin{aligned} \rho(x_1, x_2) &= \rho(x_1, -x_2) \\ E(x_1, x_2) &= E(x_1, -x_2) \\ v_1(x_1, x_2) &= v_1(x_1, -x_2) \\ v_2(x_1, x_2) &= -v_2(x_1, -x_2) \end{aligned}$$

SYMACH (Symmetrieachse)

analog zu **SYMROT**, man beachte aber, daß sich in Abhängigkeit von **GFLAG** die Gitterstruktur ändert (siehe Kap. 5.2.1).

Die Parameterübergabe ist bei allen Routinen identisch:

XIMIN, **ETAMIN** } 1. Punkt der Randkurve

NORMAL Gibt die Richtung der inneren Normale des Randpunktes an in Relation zur Kurve von der die Wanddaten abgetragen werden sollen ((1) für Γ_1, Γ_3 bzw. (-1) für Γ_2, Γ_4)

ANZ Anzahl der zu berechnenden Randdaten

XI Randzellen liegen auf ξ-Kurve (1), sonst (0)

ETA Randzellen liegen auf η-Kurve (1), sonst (0)

Nur bei der Routine **WALLS** muß zusätzlich noch die Nummer **NR** des Randes Γ_i, $i = 1, 4$ mitübergeben werden.
Die Berechnung der Randdaten erfolgt in der Routine **BOUND** in vier Schritten:

PRABOU

Vorbesetzung von gewissen Randzellen, um bei der Berechnung der Randdaten auf vorbesetzte Werte zuzugreifen;

RANDDATEN

Berechnung der Randdaten bezüglich des gewählten Randtyps (siehe **RFLAG**);

EPIBOU

Belegung der Eckzellen;

REKBOU

Randwerte zur Berechnung der Rekonstruktion von Ordnung > 1.

4.5.2 MWQUAN

In jeder Zelle werden außer den Mittelwerten der Bilanzgrößen $\rho_i, \rho\boldsymbol{v}, \rho E$ noch andere physikalische Größen benötigt, auf die in einem Zeitschritt wiederholt zugegriffen wird. Diese werden aus den Erhaltungsgrößen auf dem alten Zeitniveau zu jeder Zelle im Strömungsgebiet und auf dem Rand berechnet. Zur Charakterisierung aller Größen, die sich auf die Mittelwerte einer Zelle zum alten Zeitniveau beziehen, haben die Namen der Variablen die Struktur
MW$*$ (z.B. MW, MWRHO, MWX, MWT, MWFE).
In der Routine **MWQUAN** sind alle benötigten Größen zusammengefaßt:

MWMFRA

Berechnung der Massenbrüche $Y_i := \rho_i/\rho$, $i = 1,$**IMAX**

MWENTH

Berechnung der Bindungsenergie $\sum_{i=1}^{\mathbf{IMAX}} Y_i e_i^0$

MWTEMP

Berechnung der Temperatur aus $e = e(Y_1, \ldots, Y_{\mathbf{IMAX}}, T)$

MWALT

Erhaltungsgrößen $(\rho, \rho\boldsymbol{v}, \rho E)$ auf dem alten Zeitniveau

4.5.3 REKON

Das Ziel der Routine **REKON** ist die Rekonstruktion der Bilanzgrößen $\rho_1, \ldots, \rho_{\mathbf{IMAX}}, \rho\boldsymbol{v}$, ρE aus den gegebenen Mittelwerten und der Bestimmung der Riemannprobleme in den Zellrandmittelpunkten (siehe (48),(49)). Zur Reduzierung des Aufwandes werden jedoch nur die Größen $\rho, \rho\boldsymbol{v}, \rho E$ rekonstruiert. Zur Rekonstruktion der Partialdichten wird angenommen, daß die Zusammensetzung in jeder Zelle lokal gefroren ist. Dann ergibt sich:

$$R(\cdot, \rho_i) = Y_i \cdot R(\cdot, \rho).$$

Der Rekonstruktionsaufwand ist somit unabhängig von der Anzahl der Komponenten des Gasgemisches. Eine weitere Vereinfachung besteht darin, daß keine zweidimensionale Rekonstruktion berechnet wird, sondern die Größen nur entlang einer eindimensionalen Parameterkurve $\boldsymbol{z}_k(\xi) := \boldsymbol{x}(\xi, k-1/2)$ bzw. $\boldsymbol{z}_j(\eta) := \boldsymbol{x}(j-1/2, \eta)$ rekonstruiert werden. Dies ist in [13], Kap. 2.2, 2.3 und 3.2 ausführlich beschrieben. Das Kernstück der eindimensionalen Rekonstruktion bildet die Routine **KOMREK**. In ihr wird die komponentenweise RP–Rekonstruktion der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ entlang einer Parameterkurve berechnet. Diese setzt sich aus drei Schritten zusammen:

- Berechnung der ENO–Interpolationspolynome der Stammfunktion W (siehe [13], Kap. 2.2 und 3.2) in Newtondarstellung zu jeder Zelle;
- Taylorentwicklung der Interpolationspolynome um die Zellrandmittelwerte (siehe [1]);
- Ableiten der Interpolationspolynome ergibt die gesuchte Rekonstruktion.

In einem zweiten Schritt werden die Riemannprobleme (48) durch Taylorentwicklung von $\boldsymbol{u}(\boldsymbol{z}_k(\xi), t_{n+1/2})$ bzw. $\boldsymbol{u}(\boldsymbol{z}_j(\eta), t_{n+1/2})$ um den Punkt $(\boldsymbol{z}_k(\xi_{j+1/2}), t_n)$ bzw. $(\boldsymbol{z}_j(\eta_{k+1/2}), t_n)$ ermittelt (vgl. [13], Kap. 3.3). Dies wird über die Routine **TSTATE** gesteuert:

HORN$*$

Berechnung der i-ten Ableitung der Rekonstruktionspolynome in dem linken bzw. rechten Zellrand;

JACOBI

Berechnung des Druckes p sowie der Ableitung des Druckes, des Flusses $\boldsymbol{F}_1^I$ (vgl. 4)) und des Flusses $\boldsymbol{F}_2^I$ (vgl. 4)) für gefrorenes Gas nach den Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$;

TAYLOR

Berechnung der Koeffizienten der Taylorentwicklung (vgl. [13], Kap. 3.3);

STATE$*$

Berechnung der Zustände der Riemannprobleme im linken bzw. rechten Zellrand.

Hierbei steht $*$ für **L** bzw. **R**.
Die Berechnung der Rekonstruktion und der Riemannprobleme entlang einer Parameterkurve wird über die übergeordnete Routine **REKON** gesteuert:

REK$**$

Rekonstruktion entlang der Kurve $\boldsymbol{z}_k(\xi)$ bzw. $\boldsymbol{z}_j(\eta)$;

STA$**$

Berechnung der Riemannprobleme in den Zellrändern entlang der Kurve $\boldsymbol{z}_k(\xi)$ bzw. $\boldsymbol{z}_j(\eta)$.

Hier steht $**$ für XI bzw. ETA.
Es bleibt dem Benutzer des Programms freigestellt, andere Rekonstruktionsarten zu implementieren (siehe z.B. [1]). Es muß nur darauf geachtet werden, daß nach Verlassen der Routine **REKON** die Variablen **UL**$**$ bzw. **UR**$**$ mit den Zuständen links und rechts eines Zellrandmittelpunktes besetzt sind.
Zur besseren Vektorisierung und somit kürzeren Rechenzeiten wurden drei Routinen implementiert, in denen die RP–Rekonstruktion wie oben beschrieben in Abhängigkeit der Rekonstruktionsordnung durchgeführt wird. Die unterschiedlichen Versionen stehen in den Files

REKON1.F Rekonstruktion von Ordnung $N = 1$

REKON2.F Rekonstruktion von Ordnung $N = 2$

REKONN.F Rekonstruktion von beliebiger Ordnung $N \geq 1$

Für eine schnelle Simulationsrechnung empfiehlt sich $N = 1$, da hier keine Rekonstruktion durchgeführt werden muß. In diesem Fall ist die Rekonstruktion in einer Zelle identisch mit dem Zellmittelwert.

4.5.4 FLINV

In den Zellrandmittelpunkten müssen Riemannprobleme gelöst werden. Die Anfangszustände ergeben sich durch Auswertung der Rekonstruktionspolynome zu den benachbarten Zellen in diesem Punkt. Es interessiert hier nicht der Zwischenzustand $\tilde{\boldsymbol{u}}$ in einem Punkt, sondern der Fluß über den Zellrand in Normalenrichtung, also $\boldsymbol{F}^I_{\boldsymbol{N}}(\tilde{\boldsymbol{u}}) := \left(\boldsymbol{F}^I_1(\tilde{\boldsymbol{u}}), \boldsymbol{F}^I_2(\tilde{\boldsymbol{u}})\right) \boldsymbol{N}$.
Die Berechnung dieser Größe wird über die Routine **FLINV** gesteuert:

RPN

Berechnung aller reibungsfreien Flüsse über den Zellrand innerhalb des Strömungsbereiches und am Rand

FIBOU

Berechnung aller reibungsfreien Flüsse über die Zellränder des Diskretisierungsrandes außer auf der Symmetrieachse

ROELSG

Berechnung der Lösung der eindimensionalen Riemannprobleme in Normalenrichtung

4.5.4.1 RPN

In dieser Routine werden alle numerischen reibungsfreien Flüsse in Normalenrichtung (siehe (42)) in den Zellrandmittelpunkten

$$\begin{array}{llll} \boldsymbol{P}_4^{j,k}, & j=-1,\mathbf{MZAHLX}, & k=0,\mathbf{MZAHLY} & (\xi\text{–Parameterkurve) bzw.} \\ \boldsymbol{P}_1^{j,k}, & j=0,\mathbf{MZAHLX}, & k=-1,\mathbf{MZAHLY} & (\eta\text{–Parameterkurve}) \end{array}$$

bestimmt (vgl. Abb. 1).
Die Richtung der Parameterkurven wird über die Parameterliste gesteuert.

Input:

XI	Lösung aller Riemannprobleme entlang der Parameterkurve in ξ–Richtung (1), sonst (0)
ETA	Lösung aller Riemannprobleme entlang der Parameterkurve in η–Richtung (1), sonst (0)
JDIM	maximale Anzahl von Punkten in ξ–Richtung
KDIM	maximale Anzahl von Punkten in η–Richtung
URPL(j,k,i)	$j=1$,**JDIM**, $k=1$,**KDIM**, $i=1$,**DIMMAX** rechter Zustand der Größen $\rho, \rho\boldsymbol{v}, \rho E$ im Punkt $\boldsymbol{P}_1^{j,k}$ bzw. $\boldsymbol{P}_4^{j,k}$
URPR(j,k,i)	$j=1$,**JDIM**, $k=1$,**KDIM**, $i=1$,**DIMMAX** linker Zustand der Größen $\rho, \rho\boldsymbol{v}, \rho E$ im Punkt $\boldsymbol{P}_1^{j,k}$ bzw. $\boldsymbol{P}_4^{j,k}$
JU	erster Punkt in ξ–Richtung
JO	letzter Punkt in ξ–Richtung
KU	erster Punkt in η–Richtung
KO	letzter Punkt in η–Richtung
ARC(j,k)	$j=1$,**JDIM**, $k=1$,**KDIM** Länge des Zellrandes, über den das Riemannproblem gelöst wird
N(j,k,i)	$j=1$,**JDIM**, $k=1$,**KDIM**, $i=1,2$ Normalenrichtung, in der das Riemannproblem gelöst werden soll

Output:

NFL(j,k,i)	$j=1$,**JDIM**, $k=1$,**KDIM**, $i=1$,**DIMMAX** reibungsfreier numerischer Fluß $\boldsymbol{F}_N^I$ in Normalenrichtung $\boldsymbol{N}$ der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ (vgl. (47))
CNF(j,k,i)	$j=1$,**JDIM**, $k=1$,**KDIM**, $i=1$,**DIMMAX** reibungsfreier numerischer Fluß $\boldsymbol{F}_N^I$ in Normalenrichtung $\boldsymbol{N}$ der Größen $(\rho_1,\ldots,\rho_{\mathbf{IMAX}})$ (vgl. (47))

4.5.4.2 FIBOU

Hier werden alle Flüsse in Normalenrichtung zu den Zellrändern, die auf dem Rand des Diskretisierungsbereiches liegen, berechnet. Dazu werden die in **BOUND** berechneten Werte in den Fluß $\boldsymbol{F}_{\boldsymbol{N}}^{l}$ eingesetzt. Dies gilt jedoch nicht, falls der Rand eine Symmetrieachse ist. In diesem Fall wird dort ein Riemannproblem gelöst (siehe **RPN**).
Die Berechnung erfolgt über jeden Zellrand getrennt. Der Rand wird über eine zu den Randroutinen aus Kap. 4.5.1 identischen Parameterliste gesteuert.

4.5.4.3 ROELSG

In dieser Routine erfolgt die eigentliche Berechnung der Lösung der Riemannprobleme und der Auswertung der reibungsfreien Flüsse in Normalenrichtung (siehe (38)). Dies geschieht jeweils für eine **einzige** Parameterkurve. Letztere wird in **RPN** festgelegt. Von dort erfolgt auch der Aufruf des Riemannlösers. Die Übergabe der Anfangszustände erfolgt über die folgende Parameterliste:

Input:

U$*(l,i)$	$l = 1$,**DIM**, $i = 1,3$ Zustand $*$ der Größen $\rho\boldsymbol{v}, \rho E$
RHO$*(l)$	$l = 1$,**DIM** Zustand $*$ der Gesamtdichte ρ
X$*(l,i)$	$l = 1$,**DIM**, $i = 1$,**IMAX** Zustand $*$ der Massenbrüche Y_i
NORM$*(l,i)$	$l = 1$,**DIM**, $i = 1,2$ Normalenrichtung, in der das Riemannproblem gelöst werden soll
DIM	Anzahl der zu lösenden Riemannprobleme

Der Platzhalter $*$ steht für den linken (**L**) bzw. rechten (**R**) Zustand.

Output:

F$*(j,i)$	$j = 1$,**DIM**, $i = 1$,**IMAX**+3 reibungsfreier Fluß in Normalenrichtung gemäß (38)

Zur Zeit existieren zwei Riemannlöser, die auf dem Ansatz von Liu und Vinokur [11] basieren. Diese stehen in den Dateien:

LVCNG2.F

Riemannlöser gemäß [11] für chemisches Nichtgleichgewicht. Hier wird das Riemannproblem approximativ für die Größen $\rho_1, \ldots, \rho_{\mathbf{IMAX}}, \rho\boldsymbol{v}, \rho E$ gelöst. Diese Routine ist jedoch nur für ein 5-Komponenten-Gasgemisch inplementiert.

LVFRO2.F

Riemannlöser gemäß [5] für gefrorenes Gas als Vereinfachung von [11]. Das Riemannproblem wird approximativ für die Größen $\rho, \rho\boldsymbol{v}, \rho E$ gelöst. Die Flüsse in den Partialdichten ergeben sich aus den Massenbrüchen Y_i des linken bzw. rechten Zustandes und dem approximativen Fluß in ρ. Hier ist die Anzahl der Komponenten des Gasgemisches nicht eingeschränkt.

Vor dem Programmstart muß der Benutzer eine der Alternativen auswählen und in die Datei **ROELSG.F** kopieren und das Programm durch Aufruf der MAKE–Datei **nscng2.m** neu übersetzen (siehe auch Kap. 10).

4.5.5 FLVIS

Im Falle einer reibungsbehafteten Strömung (**NSFLAG** = 1) muß in den Zellrandmittelpunkten der viskose Fluß $\boldsymbol{F}_{\boldsymbol{N}}^{V}$ in Normalenrichtung $\boldsymbol{N}$ gemäß (39) berechnet werden. Dies geschieht analog zur Routine **FLINV** für die Punktemengen $\boldsymbol{P}_1^{j,k}$, $j = 0, \mathbf{MZAHLX}$, $k = -1, \mathbf{MZAHLY}$ bzw. $\boldsymbol{P}_4^{j,k}$, $j = -1, \mathbf{MZAHLX}$, $k = 0, \mathbf{MZAHLY}$ getrennt; erstere liegen alle auf Parameterkurven in η–Richtung, letztere auf solchen in ξ–Richtung. Dabei wird zwischen solchen Punkten, die in der Strömung liegen und solchen auf einer festen Körperwand, unterschieden:

VFLPAR

Berechnung aller viskosen Flüsse in den Zellrandmittelpunkten, die alle auf einer Parameterkurve in einer Richtung (ξ bzw. η) liegen und nicht auf einer Körperwand;

VFLWAL

Berechnung aller viskosen Flüsse in den Zellrandmittelpunkten, die alle auf einer einzigen Parameterkurve in ξ– bzw. η–Richtung liegen und mit einer festen Körperwand zusammenfallen.

Falls die Strömung rotationssymmetrisch ist (**GFLAG** = 1), dann werden hier zusätzlich die Anteile **Q** des rotationsbedingten Quellterms R^V (vgl. (28)) berechnet, der sich nicht als $\boldsymbol{F}_2^V / Y_2$ darstellen läßt, d.h.

$$\boldsymbol{Q} := \boldsymbol{R}^V - \left(\boldsymbol{F}_{2,1}^V, \ldots, \boldsymbol{F}_{2,\mathbf{IMAX}}^V, 0, 0, \boldsymbol{F}_{2,\mathbf{IMAX}+3}^V\right)^T . \tag{55}$$

Vergleicht man diese Differenz mit (28), so zeigt sich, daß nur drei Terme — für die Impuls– und die Energiegleichung — zu berechnen sind.
Dies geschieht von **VFLPAR** und **VFLWAL** aus, durch Aufruf der Routinen:

RQVXI

Berechnung von $\boldsymbol{Q}$ (vgl.(55)) in den Zellrandmittelpunkten $\boldsymbol{P}_4^{j,k}$, die auf Parameterkurven in ξ–Richtung liegen;

RQVETA

Berechnung von $\boldsymbol{Q}$ (vgl.(55)) in den Zellrandmittelpunkten $\boldsymbol{P}_1^{j,k}$, die auf Parameterkurven in η–Richtung liegen;

RQWVXI

Berechnung von $\boldsymbol{Q}$ (vgl.(55)) in den Zellrandmittelpunkten, die auf einer Parameterkurven in ξ–Richtung liegen und mit einer festen Wand zusammenfallen;

RQWVETA

Berechnung von $\boldsymbol{Q}$ (vgl.(55)) in den Zellrandmittelpunkten, die auf einer Parameterkurven in η–Richtung liegen und mit einer festen Wand zusammenfallen;

Diese stehen in der Datei **RQUELL.F**.

4.5.5.1 VFLPAR

In dieser Routine werden die viskosen Flüsse $\boldsymbol{F}_i^V$ in den Punkten $\boldsymbol{P}_1^{j,k}$ bzw. $\boldsymbol{P}_4^{j,k}$ berechnet, die nicht auf dem Körperrand liegen. Die Punkte liegen dabei immer auf Parameterkurven in ξ–Richtung bzw. η–Richtung. Die Richtung wird über folgende Parameterliste gesteuert:

Input:

XI	Berechnung aller viskosen Flüsse in den Punkten auf einer Parameterkurve in ξ–Richtung (1), sonst (0)
ETA	Berechnung aller viskosen Flüsse in den Punkten auf einer Parameterkurve in η–Richtung (1), sonst (0)
JDIM	maximale Anzahl von Punkten in ξ–Richtung
KDIM	maximale Anzahl von Punkten in η–Richtung
JU	erster Punkt in ξ–Richtung
JOB	letzter Punkt in ξ–Richtung
KU	erster Punkt in η–Richtung
KOB	letzter Punkt in η–Richtung
ARC(j,k)	j = 1,**JDIM**, k = 1,**KDIM** Länge des Zellrandes, über den die Flüsse berechnet werden
NORM(j,k,i)	j = 1,**JDIM**, k = 1,**KDIM**, i = 1,2 Normalenrichtung im Zellrandmittelpunkt, in dem die Flüsse berechnet werden sollen

Output:

VF(j,k,i)	j = 1,**JDIM**, k = 1,**KDIM**, i = 1,**DIMMAX** viskoser Fluß in x_1–Richtung der Erhaltungsgrößen $(\rho, \rho\boldsymbol{v}, \rho E)$
VG(j,k,i)	j = 1,**JDIM**, k = 1,**KDIM**, i = 1,**DIMMAX** viskoser Fluß in x_2–Richtung der Erhaltungsgrößen $(\rho, \rho\boldsymbol{v}, \rho E)$
VFD(j,k,i)	j = 1,**JDIM**, k = 1,**KDIM**, i = 1,**IMAX** viskoser Fluß in x_1–Richtung der Partialdichten ρ_i
VGD(j,k,i)	j = 1,**JDIM**, k = 1,**KDIM**, i = 1,**IMAX** viskoser Fluß in x_2–Richtung der Partialdichten ρ_i

4.5.5.2 VFLWAL

In Analogie zur Routine **VFLPAR** werden hier die entsprechenden viskosen Flüsse in den Punkten, die auf einer Körperwand liegen, berechnet. In diesen Punkten ergeben sich Änderungen in der Berechnung, die in der Haftbedingung, der Wanddiffusion und der Modellierung des Wärmeübergangs an der Wand begründet sind. Diesen Wandeffekten wird in der Routine **VFLWAL** Rechnung getragen.
Die Parameterliste ist identisch zu der von **VFLPAR**. Man beachte aber, daß vorher die Flüsse in allen Punkten entlang **aller** Parameterkurven in einer Richtung berechnet wurden. Dies entspricht einer zweidimensionalen Punktmenge. Hier werden jedoch die Flüsse in den Punkten entlang einer **einzigen** eindimensionalen Parameterkurve berechnet.
Diese werden über die Parameter **JU**, **JOB**, **KU**, **KOB** gesteuert:

Richtung der Parameterkurve	Rand Γ_i	**JU**	**JOB**	**KU**	**KOB**
ξ	1	**MZAHLX**	**MZAHLX**	0	**MZAHLY**
η	2	0	**MZAHLX**	**MZAHLY**	**MZAHLY**
ξ	3	-1	-1	0	**MZAHLY**
η	4	0	**MZAHLX**	-1	-1

4.5.5.3 RQVXI, RQVETA, RQWVXI, RQWVETA

Im Falle einer rotationssymmetrischen und reibungsbehafteten Strömung (**GFLAG** = 1, **NSFLAG** = 1) muß der Quellterm $\boldsymbol{R}^V$ (vgl. (54)) berechnet werden. Dieser setzt sich zusammen aus $\boldsymbol{F}_2^V/x_2$ und $\boldsymbol{Q}$ (vgl. (55)).
Während der erste Term in **VFLPAR** bzw. **VFLWAL** berechnet wird, muß der zweite Term zusätzlich berechnet werden. Dies geschieht in **RQV**$*$ bzw. **RQWV**$*$. Der Platzhalter $*$ steht dabei für die Richtung der Parameterkurven, also **XI** und **ETA**. Diese Routinen werden aus **VLFPAR** und **VFLWAL** aufgerufen. Dort werden schon die meisten, für die Berechnung von $\boldsymbol{Q}$ benötigten Größen berechnet, insbesondere die Transportkoeffizienten. Übergeben werden folgende Parameter:

Input:

XI	Berechnung von $\boldsymbol{Q}$ in den Punkten $\boldsymbol{P}_4^{j,k}$ auf einer Parameterkurve in ξ–Richtung (1), sonst (0)
ETA	Berechnung von $\boldsymbol{Q}$ in den Punkten $\boldsymbol{P}_1^{j,k}$ auf einer Parameterkurve in η–Richtung (1), sonst (0)
JDIM	maximale Anzahl von Punkten in ξ–Richtung
KDIM	maximale Anzahl von Punkten in η–Richtung
JU	erster Punkt in ξ–Richtung
JO	letzter Punkt in ξ–Richtung

KU	erster Punkt in η–Richtung
KO	letzter Punkt in η–Richtung
TPP(j)	j = 1,**DIM** Temperatur im Zellrandmittelpunkt
XP(j,i)	j = 1,**DIM**, i = 1,**IMAX** molare Konzentration im Zellrandmittelpunkt
UP(j)	j = 1,**DIM** Geschwindigkeitskomponente in x_1–Richtung im Zellrandmittelpunkt
VP(j)	j = 1,**DIM** Geschwindigkeitskomponente in x_2–Richtung im Zellrandmittelpunkt
VX(j)	j = 1,**DIM** partielle Ableitung der Geschwindigkeitskomponente in x_2–Richtung nach der x_1–Richtung im Zellrandmittelpunkt
TX(j)	j = 1,**DIM** partielle Ableitung der Temperatur nach der x_1–Richtung im Zellrandmittelpunkt
TY(j)	j = 1,**DIM** partielle Ableitung der Temperatur nach der x_2–Richtung im Zellrandmittelpunkt
DETA(j)	j = 1,**DIM** Länge des Zellrandes
DXI(j)	j = 1,**DIM** Abstand zwischen den Zellmittelpunkten der an den Zellrand angrenzenden Zellen
TAUXY(j)	j = 1,**DIM** Scherspannungskoeffizient τ_{12} gemäß (7) im Zellrandmittelpunkt
MUE(j)	j = 1,**DIM** Viskositätskoeffizient des Gasgemisches im Zellrandmittelpunkt
LK	Nummer der Parameterkurve in η–Richtung, auf der die Zellrandmittelpunkte liegen, entlang der $\boldsymbol{Q}$ bestimmt werden soll
LJ	Nummer der Parameterkurve in ξ–Richtung, auf der die Zellrandmittelpunkte liegen, entlang der $\boldsymbol{Q}$ bestimmt werden soll

Der Parameter **LJ** wird jedoch nur an **RQVWXI** übergeben, sonst nur der Parameter **LK**.

4.5.6 MWSTEP

Zur Berechnung des Mittelwertes der Bilanzgrößen $\rho_1, \ldots, \rho_{\mathbf{IMAX}}, \rho\boldsymbol{v}, \rho E$ auf dem neuen Zeitniveau $t = t_n +$ **DELTAT** müssen die Flüsse über den Zellgrenzen gemäß (44) aufaddiert werden. Dies geschieht für den reibungsfreien und den reibungsbehafteten Fluß separat in den Routinen **MWINV** und **MWVIS**:

MWVIS

Berechnung der numerischen viskosen Flüsse $\widehat{\boldsymbol{F}^V}_{\boldsymbol{N}}$ (41) aus den in **VFLPAR** und **VFLWAL** berechneten Flüsse in x_1– und x_2–Richtung sowie des Quellterms $\widehat{\boldsymbol{R}^V}$ (43) in den Zellrandmittelpunkten und anschließendes aufaddieren auf die alten Mittelwerte der Bilanzgrößen gemäß (44);

MWINV

Berechnung des Mittelwertes der Bilanzgrößen auf dem neuen Zeitniveau durch aufaddieren der numerischen inviskosen Flüsse $\widehat{\boldsymbol{F}^I}_{\boldsymbol{N}}$ (40) (siehe **FLINV**) und des Quellterms $\widehat{\boldsymbol{R}^I}$ (42) gemäß (44) aus den alten Mittelwerten.

4.5.7 TEST

Nach der Mittelwertbildung (44) der Bilanzgrößen werden die beiden folgenden Bedingungen in den Routinen überprüft:

PTEST (falls **PFLAG** = 1)

Positivität des Druckes auf dem neuen Zeitniveau;

STEADY

mittlere zeitliche Änderung der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ im Vergleich zu den Mittelwerten auf dem vorausgegangenen Zeitniveau.

4.5.7.1 PTEST

Die Mittelwertbildung (44) der Bilanzgrößen $\rho_1, \ldots, \rho_{\mathbf{IMAX}}, \rho\boldsymbol{v}, \rho E$ kann eventuell zu negativen Drücken führen. Diese würden zu einem sofortigen Programmabsturz führen. Bei den bisher untersuchten hypersonischen Staupunktproblemen trat dieses unphysikalische Phänomen lokal **vor** dem Frontstoß in der Nähe der Stromlinien durch den Staupunkt, also in der Anströmung, auf. Dort kam es aufgrund des starken Stoßes zu leichten Oszillationen der Rekonstruktion von Ordnung größer als 1. Diese führten zu einer Absenkung des Druckes in der Anströmung. Da dieser in der Regel sehr gering ist, resultierte daraus mit fortschreitender Integrationszeit ein negativer Druck. Um dies zu verhindern, wird der Impuls in der entsprechenden Zelle so skaliert, daß der Druck auf den Anströmwert hochgesetzt wird, aber die Totalenergie konstant gehalten wird. Dies entspricht physikalisch einer Umwandlung von innerer Energie in kinetische Energie bei Wahrung der Energieerhaltung. Dies hat sich als eine sinnvolle Korrekturmaßnahme in der Praxis bewährt.
Es kann aber geschehen, daß diese Korrekturmaßnahme nicht durchführbar ist, weil aus einem negativen Skalierungsfaktor die Wurzel zu ziehen ist. In diesem Fall wird die Energieerhaltung verletzt und soviel Energie auf den Mittelwert der Totalenergie aufaddiert wie erforderlich ist, um den Anströmdruck zu erreichen. Diese Maßnahme verhindert zwar den sofortigen Programmabsturz, führt aber langfristig zu unphysikalischen Ergebnissen. In diesem Fall ist eine erneute Rechnung mit kleinerer Zeitschrittweite **DELTAT** erforderlich. Zur Laufzeit des Programms werden entsprechende Warnungen ausgegeben, falls eine Korrekturmaßnahme erforderlich ist und welche durchgeführt wurde.

Warnung:

Es wird davor gewarnt, die Routine **PTENS** immer zu verwenden. So kann es geschehen, daß bei einer schrägen Anströmung eines Körpers auf dessen Leeseite ein Druckabfall unter den Anströmwert sehr wohl physikalisch sein kann. In diesem

Fall darf keine Korrektur vorgenommen werden. Man sollte deshalb diese nur dann durchführen, falls der Druck in der Anströmung auf einen Wert kleiner als dem kleinsten Wert hinter dem Stoß sinkt. In diesem Fall muß **PFLAG** = 0 gesetzt werden.

4.5.7.2 STEADY

Da man bei der Staupunktberechnung an der stationären Lösung interessiert ist, wird nach einer von dem Benutzer zur Laufzeit über die Eingabe festzulegende Anzahl von Integrationsschritten (**SSTEPS**) überprüft, ob der stationäre Zustand erreicht ist. Dazu wird die relative zeitliche Änderung in einer Zelle in den Größen $\rho, \rho\boldsymbol{v}, \rho E$ berechnet und über alle Zellen gemittelt. Diese Residuen werden in der Regel nicht gegen Null konvergieren, sondern ab einem gewissen Zeitpunkt konstant bleiben. Dies entspricht dann dem stationären Zustand. Das Phänomen kann durch die Diskretisierungsfehler begründet werden. Aufgrund der Mittelwertbildung werden ab einem gewissen Zeitpunkt nur noch diese Fehler zwischen den einzelnen Zellen lokal transportiert.

4.6 OUTPUT

Nach Beendigung der Zeitintegration werden die berechneten Daten ausgegeben und ein Protokoll mit den Parametern der Strömungssimulation erstellt. Der Filename des Protokolls hat die Endung **.PRO**. Es wird in der Routine **OUTPUT** erstellt. Von dort wird auch die Prozedur **OUTDAT** aufgerufen. In ihr werden die berechneten Mittelwerte gemäß Kap. 5.2.2 in das File, dessen Name in **DATOUT** abgespeichert wurde, rausgeschrieben.
Beim Aufruf beider Routinen muß die Zahl der in diesem Programmablauf schon durchgeführten Integrationsschritte übergeben werden. An die Routine **OUTDAT** muß zusätzlich noch der Dateiname übergeben werden, in den die Daten geschrieben werden sollen. Da in der Regel viele Zeitschritte in einem Programmlauf ausgeführt werden und dies in der Regel sehr viel Rechenzeit erfordert, die auf kontingentierten Rechneranlagen teuer sind, werden die Daten nach je **ZSTEPS** Zeitschritten auf ein File abgespeichert. Der Name dieses Files steht in der Variablen **DATZWI**.

5 Programmsteuerung über Optionen und Dateien

5.1 Optionen

Wie in den vorausgegangenen Kapiteln schon dargelegt wurde, hat der Benutzer relativ viele Freiheiten bezüglich

- der physikalischen Modellierung
- der numerischen Diskretisierung
- und der zugrundeliegenden Geometrie des Strömungsgebietes.

Zur Steuerung der Optionen müssen zur Laufzeit Flags gesetzt werden. Diese sind im folgenden aufgelistet:

NSFLAG

reibungsfreie Strömung (0)
reibungsbehaftete Strömung (1)

CFLAG

inertes Gas (0)
chemisches Nichtgleichgewicht (1)

OFLAG (nur falls **CFLAG**= 1)

ohne Operatorsplitting für chemische Produktionsraten (0)
mit Operatorsplitting für chemische Produktionsraten (1)

WFLAG (nur falls **NSFLAG**= 1)

adiabate Wand $\left(\frac{\partial T}{\partial \mathbf{N}} = 0\right)$ (0)
nicht adiabate Wand (T_W = const.) (1)

KWFLAG (nur falls **NSFLAG**= 1)

voll katalytische Wand ($Y_{i,W}$ = const.) (0)
nicht katalytische Wand $\left(\frac{\partial Y_i}{\partial \mathbf{N}} = 0\right)$ (1)

GFLAG

ebene Strömung (0)
rotationssymmetrische Strömung (1)
achsensymmetrische, ebene Strömung (2)

RFLAG(i)

Zuordnung eines der folgenden Randtypen zu den vier Strömungsrändern Γ_i, $i = 1, 4$:

hypersonischer Inflow	(1)
hypersonischer Outflow	(2)
Wand (Haftbedingung, **NSFLAG** = 1)	(3)
Wand (Verdünnungswelle, **NSFLAG** = 0)	(4)
Wand (Reflexion, **NSFLAG** = 0)	(5)
Symmetrieachse (Rotationssymmetrie, nur $x_2 = 0$)	(6)
Symmetrieachse (Achsensymmetrie, nur $x_2 = 0$)	(7)

KFLAG(i)

Charakterisierung des Randes Γ_i, $i = 1, 4$:

freier Strömungsrand	(0)
feste Körperwand	(1)

PFLAG

ohne Positivitätstest des Druckes	(0)
mit Positivitätstest des Druckes	(1)

Diese Optionen werden beim Programmstart über eine Datei, deren Name beim Programmaufruf vom Benutzer angegeben werden muß (vgl. Kap. 10), eingegeben.
Neben dieser Eingabedatei werden noch weitere Dateien benötigt, die das physikalische Modell festlegen und für die Diskretisierung benötigt werden:

- Referenzgrößen der Anströmung (**REFQUA**)
- Gitter (**GITTER**)
- Mittelwerte der Bilanzgrößen in den Zellen (**DATIN**, **DATZWI**, **DATOUT**)
- Tabellen für die Koeffizienten der Kurvenfits $\tilde{e}_i(T)$, $\tilde{c}_{V,i}(T)$, $\mu_i(T)$, $\widetilde{\mathcal{D}}_{ij}(T)$ und $\tilde{\mathcal{K}}_g^i(\ln T)$
  (**ENINP**, **CVINP**, **MUINP**, **BDINP**, **AKINP**)

Die Dateinamen sind in Klammern angeführt (vgl. auch Kap. 7.2).

5.2 Dateistrukturen

Im folgenden werden die Dateistrukturen der Dateien erklärt, die während des Programmablaufs eingelesen bzw. auf die Daten rausgeschrieben werden.

5.2.1 Eingabedatei

Die Datei, die beim Programmstart als Parameter mit angegeben werden muß (vgl. Kap. 10), hat folgende Struktur:

OFLAG : Parameter zur Steuerung des Operatorsplittings

WFLAG : Parameter zur Steuerung der thermischen Wandmodellierung

KWFLAG: Parameter zur Steuerung der katalytischen Wandmodellierung

GFLAG : Parameter zur Steuerung der Geometrie

NSFLAG : Parameter zur Steuerung der Differentialgleichung

CFLAG : Parameter zur Steuerung der Modellierung der chemischen Produktion

RFLAG(i): Randtypen zu Rand $\Gamma_i, i = 1, 4$:

hypersonischer Inflow	(1)
hypersonischer Outflow	(2)
Wand (Haftbedingung, Navier–Stokes)	(3)
Wand (Verdünnungswelle, Euler)	(4)
Wand (Reflexion, Euler)	(5)
Symmetrieachse (Rotationssym., nur $x_2 = 0$)	(6)
Symmetrieachse (Achsensym., nur $x_2 = 0$)	(7)

KFLAG(i): Charakterisierung des Gitterrandes $\Gamma_i, i = 1, 4$

PFLAG : Parameter zur Steuerung der Drucküberprüfung

ENINP : Dateiname für die Tabellierung der inneren Energie $\tilde{\epsilon}_i(T)$

CVINP : Dateiname für die Tabellierung der Wärmekapazität $\tilde{c}_{V,i}(T)$

MUINP : Dateiname für die Tabellierung von $\mu_i(T)$

BDINP : Dateiname für die Tabellierung von $\widetilde{\mathcal{D}}_{ij}(T)$

AKINP : Dateiname für die Tabellierung von $\tilde{K}_g^i(\ln T)$

POLGRA : Interpolationsordnung

NSTEPS : Anzahl durchzuführender Zeitschritte

DELTAT : Zeitschrittweite

DATZWI : Dateiname für Zwischenspeicher

ZSTEPS : Anzahl der Zeitschritte, nach denen die Daten zwischengespeichert werden sollen

SSTEPS : Anzahl der Zeitschritte, nach denen überprüft werden soll, ob die Strömung stationär ist

GITTER : Dateiname für Gitter

DATINP : Dateiname für Bilanzgrößen

DATOUT : Dateiname für Ausgabedaten

REFQUA : Dateiname für Referenzgrößen

Die Dateinamen dürfen dabei nicht länger als 14 Zeichen sein. Die Variablen sind in Kap. 7 erklärt.

5.2.2 Referenzgrößen der Anströmung

Die Strömung ist durch die Anströmwerte charakterisiert. Diese Werte werden in den Routinen **PTENS** und **HYPIN** benötigt. Der Benutzer hat diese Werte in einer Datei mit folgender Struktur zur Verfügung zu stellen:

XREF(i) Massenbruch Y_i der Anströmung, $i = 1, \mathbf{IMAX}$

RHOREF Massendichte der Anströmung

MREF x_1–Impuls der Anströmung

NREF x_2–Impuls der Anströmung

REREF Totalenergie der Anströmung

PREF Druck der Anströmung

TREF Temperatur der Anströmung

TWALL(1) (falls **NSFLAG** = 1, **WFLAG** = 1, **KFLAG**(1) = 1) Wandtemperatur

TWALL(2) (falls **NSFLAG** = 1, **WFLAG** = 1, **KFLAG**(2) = 1) Wandtemperatur

TWALL(3) (falls **NSFLAG** = 1, **WFLAG** = 1, **KFLAG**(3) = 1) Wandtemperatur

TWALL(4) (falls **NSFLAG** = 1, **WFLAG** = 1, **KFLAG**(4) = 1) Wandtemperatur

XWALL(i) (falls **NSFLAG** = 1, **KWFLAG** = 0) Massenbruch an der Wand, $i = 1, \mathbf{IMAX}$

5.2.3 Gitter

Die Strömungssimulation wird auf einem strukturierten Gitter durchgeführt. Das Gitter ist somit gegeben durch die Transformation eines zweidimensionalen Parameterbereiches P auf das Gebiet $G \subset \mathrm{IR}^2$, formal:

$$\boldsymbol{x} : P \to G \quad \text{def. durch} \quad \boldsymbol{x} = \boldsymbol{x}(\xi, \eta).$$

Zu jeder Zelle $V_{j,k}$ werden die Eckpunkte abgespeichert, d.h.

$$V_{j,k} = \square(\boldsymbol{x}(j \pm 1/2, k \pm 1/2)) \qquad \begin{array}{l} j = -1, \mathbf{MZAHLX} + 1, \\ k = -1, \mathbf{MZAHLY} + 1. \end{array}$$

Struktur: MZAHLX MZAHLY

$$\boldsymbol{x}(j + 1/2, k + 1/2),\ j = -2, \mathbf{MZAHLX} + 1,\ k = -2, \mathbf{MZAHLY} + 1$$

Diese Struktur gilt für alle Gitter, unabhängig von der zugrunde liegenden Geometrie. Wird insbesondere eine achsensymmetrische (**GFLAG** = 2) oder rotationssymmetrische Geometrie (**GFLAG** = 1) gewählt, so sind folgende Besonderheiten an der Symmetrieachse zu beachten:

1. Symmetrieachse darf nur $x_2 = 0$ sein

2. (a) **GFLAG** = 1:
$$\begin{aligned} x_1(-3/2, k+1/2) &= x_1(1/2, k+1/2) \\ x_2(-3/2, k+1/2) &= -x_2(1/2, k+1/2) \\ x_2(-1/2, k+1/2) &= 0 \end{aligned}$$
$$k = -2, \mathbf{MZAHLY} + 1$$

   (b) **GFLAG** = 2:
$$\begin{aligned} x_1(-1/2, k+1/2) &= x_1(1/2, k+1/2) \\ x_2(-1/2, k+1/2) &= -x_2(1/2, k+1/2) \\ x_1(-3/2, k+1/2) &= x_1(3/2, k+1/2) \\ x_2(-3/2, k+1/2) &= -x_2(3/2, k+1/2) \end{aligned}$$
$$k = -2, \mathbf{MZAHLY} + 1$$

5.2.4 Mittelwerte der Bilanzgrößen

Zum Start der Rechnung werden die Mittelwerte zu jeder Zelle im Strömungsgebiet und auf dem Rand des Diskretisierungsbereiches benötigt. Die Struktur für die einzulesenden als auch für die rauszuschreibenden Daten sind identisch:

Struktur:	**ASTEPS**	# alter Zeitschritt
	DUMMY	# Leerzeile
	(**MW**(j,k,i), $i = 2$,**DIMMAX**),	# $\rho\boldsymbol{v}, \rho E, \rho_1, \ldots, \rho_{\mathbf{IMAX}}$
	(**MWRHO**(j,k,i), $i = 1$,**IMAX**)	# $k = -1$,**MZAHLY**$+1$,
		$j = -1$,**MZAHLX**$+1$

Achtung:

Zur Zeit ist das 5–Komponenten–Modell für Luft mit den Komponenten N, N_2, O, O_2 und NO in gleicher Reihenfolge installiert.

5.2.5 Tabellen

Um nicht auf ein spezielles Modell für die Transport– und Diffusionskoeffizienten sowie die Energie eingeschränkt zu sein, werden für diese Größen Kurvenfits verwendet. Dadurch hat der Benutzer die Möglichkeit ein beliebiges Modell seiner Wahl zu verwenden, ohne im Programm selber etwas ändern zu müssen. Näheres zu dieser Strategie findet man in [9]. Folgende Größen werden gefittet:

1. innere Energie der Komponente i

$$\check{e}_i(T) := e_i(T) \cdot M_i/R/T = \sum_{l=1}^{4} a_{l,i,j} \cdot T^{l-1},\ j = INT\left(\frac{T - \mathbf{TMIN}}{\mathbf{DTP}} + 1\right);$$

2. Wärmekapazität bei konstantem Volumen der Komponente i

$$\check{c}_{V,i}(T) := c_{V,i}(T) \cdot M_i/R = \sum_{l=1}^{4} b_{l,i,j} \cdot T^{l-1},\ j = INT\left(\frac{T - \mathbf{TMIN}}{\mathbf{DTP}} + 1\right);$$

3. Komponentenviskosität der Komponente i

$$\mu_i(T) = \sum_{l=1}^{5} c_{l,i,j} \cdot T^{l-1},\ j = INT\left(\frac{T - \mathbf{TMINMU}}{\mathbf{DTPMU}} + 1\right);$$

4. binärer Diffusionskoeffizient der Komponenten i, j

$$\tilde{\mathcal{D}}_{ij}(T) := \mathcal{D}_{ij} \cdot p = \sum_{l=1}^{5} d_{l,i,j,k} \cdot T^{l-1},\ k = INT\left(\frac{T - \mathbf{TMINBD}}{\mathbf{DTPBD}} + 1\right),$$

5. Gleichgewichtskonstante der Reaktionsklasse i

$$\tilde{K}_g^i(\ln T) := K_g^i(T) = \sum_{l=1}^{4} e_{lik}(\ln T)^{l-1},\ k = INT\left(\frac{\ln T - \mathbf{TMINAK}}{\mathbf{DTPAK}} + 1\right).$$

Struktur 1:

anz	Anzahl der Kurvenfits
TMIN	minimale Temperatur des Diskretisierungsbereiches
TMAX	maximale Temperatur des Diskretisierungsbereiches
DTP	äquidistante Intervallänge
JANZ	Anzahl der Diskretisierungsintervalle
'xx'	Komponente i, z. B. N1, N2, O1, O2, NO
W(i)	Molgewichte der Komponente i
FENTH(i)	Formationsenthalpie der Komponente i
IEA(l,i,j)	Koeffizienten $a_{l,i,j}$ der Kurvenfits von $\tilde{e}_i(T)$, $l = 1,4$, $j = 1$,**JANZ**

Struktur 2:

anz	Anzahl der Kurvenfits
TMIN	minimale Temperatur des Diskretisierungsbereiches
TMAX	maximale Temperatur des Diskretisierungsbereiches
DTP	äquidistante Intervallänge
JANZ	Anzahl der Diskretisierungsintervalle
'xx'	Komponente i, z. B. N1, N2, O1, O2, NO
CVA(l,i,j)	Koeffizienten $b_{l,i,j}$ der Kurvenfits von $\tilde{c}_{V,i}(T)$, $l = 1,4$, $j = 1$,**JANZ**

Achtung: Die Größen **TMIN**, **TMAX**, **DTP**, **JANZ** müssen identisch sein!

Struktur 3:

anz	Anzahl der Kurvenfits

TMINMU minimale Temperatur des Diskretisierungsbereiches
TMAXMU maximale Temperatur des Diskretisierungsbereiches
DTPMU äquidistante Intervallänge
JANZMU Anzahl der Diskretisierungsintervalle
'xx' Komponente i, z. B. N1, N2, O1, O2, NO
MUA(l,i,j) Koeffizienten $c_{l,i,j}$ des Kurvenfits von $\mu_i(T)$, $l = 1,5$, $j = 1$,**JANZMU**

Struktur 4:

anz Anzahl der Kurvenfits
TMINBD minimale Temperatur des Diskretisierungsbereiches
TMAXBD maximale Temperatur des Diskretisierungsbereiches
DTPBD äquidistante Intervallänge
JANZBD Anzahl der Diskretisierungsintervalle
'xx' 'yy' Komponente i,j, z. B. N1, N2, O1, O2, NO mit $1 \leq j \leq i \leq$ **IMAX**
BDA(l,i,j,k) Koeffizienten $d_{l,i,j,k}$ des Kurvenfits $\widetilde{\mathcal{D}}_{ij}(T)$, $l = 1,5$, $k = 1$,**JANZBD**

Struktur 5:

anz Anzahl der Kurvenfits
TMINAK minimale Temperatur des Diskretisierungsbereiches
TMAXAK maximale Temperatur des Diskretisierungsbereiches
DTPAK äquidistante Intervallänge
JANZAK Anzahl der Diskretisierungsintervalle
'xx' Reaktionsklasse i, z. B.
AKOEFF(l,i,j) Koeffizienten $e_{l,i,j}$ des Kurvenfits $\tilde{K}_g^i(\ln T)$, $l = 1,4$, $j = 1$,**JANZAK**

Bemerkung:

Die Zuweisung der Strings 'xx' und 'yy' einer Komponente erfolgt in der Routine **NUMMER**. Zur Zeit ist folgende Zuweisung für die Komponenten des Gasgemisches und der Reaktionsklassen implementiert:

SPEZIE(1) = 'N1' $\longleftrightarrow$ 1
SPEZIE(2) = 'N2' $\longleftrightarrow$ 2
SPEZIE(3) = 'O1' $\longleftrightarrow$ 3
SPEZIE(4) = 'O2' $\longleftrightarrow$ 4
SPEZIE(5) = 'NO' $\longleftrightarrow$ 5
REAKT(1) = 'O2DIS' $\longleftrightarrow$ O_2 - Dissoziation
REAKT(2) = 'N2DIS' $\longleftrightarrow$ N_2 - Dissoziation
REAKT(3) = 'NODIS' $\longleftrightarrow$ NO - Dissoziation
REAKT(4) = 'O2REK' $\longleftrightarrow$ O_2 - Rekombination
REAKT(5) = 'NOREK' $\longleftrightarrow$ NO - Rekombination

Zur Zeit existieren folgende Dateien für das 5–Komponenten–Luftmodell:

EN_AIR5.DAT, **CV_AIR5.DAT**: harmonischer Oszillator, keine Kopplung von Rotation und Vi-) bration, thermisches Gleichgewicht aller inneren Freiheitsgrade, analytisches Energiemodell (siehe [9], Kap. 3.2, 8.1, 8.2

EN_ESS5.DAT, **CV_ESS5.DAT**, **AK_ESS5.DAT**: anharmonischer Oszillator, Kopplung von Rotation und Vibration, thermisches Gleichgewicht aller inneren Freiheitsgrade, analytisches Energiemodell (siehe [2])

MU_AIR5.DAT analytisches Modell (siehe [9], Kap. 4.3)

MU_AVE5.DAT Mittelung der Komponentenviskositätskoeffizienten über die Gleichgewichtszusammensetzung von 5–Komponenten–Luft; diese ist nur in Zusammenhang mit dem Modell **SPF** (siehe Kap. 6.3) zu verwenden

DB_AIR5.DAT Parameter des analytischen Modells (siehe [9], Kap. 4.3)

Näheres zur Implementierung der Modelle findet man in [9], Kap. 5.

5.2.6 Protokoll

Zum Abschluß des Programms wird eine Datei angelegt, die den Programmablauf protokolliert. Diese hat den Namen fn.**PRO** mit dem Filenamen, der in **DATOUT** steht.

Struktur:

NSFLAG	
CFLAG	
OFLAG	
XREF(i),	$i = 1$,**IMAX**
RHOREF	
MREF	
NREF	
REREF	
PREF	
TREF	
WFLAG	(falls **NSFLAG** $= 1$)
TWALL(i)	(falls **NSFLAG** $= 1$, **WFLAG** $= 1$, **KFLAG**$(i) = 1$, $i = 1, 4$)
XWALL(i)	(falls **NSFLAG** $= 1$, **KWFLAG** $= 0$) $i = 1$,**IMAX**
MUEREF	(falls **NSFLAG** $= 1$)

MREF/MUEREF (falls **NSFLAG** = 1) Reynoldszahl pro Länge

POLGRA

GFLAG

DELTAT

MZAHLX

MZAHLY

GITTER

6 Schnittstellen

Einige Routinen sind so angelegt, daß sie auf keine globalen Variablen zugreifen, d.h. es werden keine INCLUDE–Dateien eingebunden. Die gesamte Ein– und Ausgabe erfolgt über eine Parameterliste. Diese Vorgehensweise wurde nur bei solchen Unterprogrammen angewendet, in denen physikalische Größen berechnet werden, die auf Modellen beruhen, zu denen Alternativen existieren. In diesem Fall soll es dem Benutzer ermöglicht werden, eigene Modelle zu verwenden, ohne dazu Änderungen im ganzen Programm vornehmen zu müssen. Zu diesem Zweck wurde die Parameterliste so allgemein gestaltet, daß möglichst viele, zur Zeit bekannte Modelle, adaptierbar sind.

6.1 Energiemodell

6.1.1 Schnittstelle IET

Ziel: Berechnung der massenspezifischen inneren Energie

$$e = e(Y_1, \ldots, Y_{\mathbf{IMAX}}, T) := \sum_{i=1}^{\mathbf{IMAX}} Y_i e_i(T)$$

über einen stückweisen polynomialen Fit von

$$\tilde{e}_i(T) := e_i(T) \cdot M_i/R/T = \sum_{l=1}^{4} \mathbf{IEA}(l,i,j) \cdot T^{l-1},\ j = INT\left(\frac{T - \mathbf{TMIN}}{\mathbf{DTP}} + 1\right);$$

Input:

$\mathbf{T}(j)$	Temperatur, $j = 1$,**MANZ**
$\mathbf{X}(j,i)$	Massenbrüche, $i = 1$,**IMAX**, $j = 1$,**MANZ**
$\mathbf{IEA}(l,i,k)$	Koeffizienten der Kurvenfits für die inneren massenspezifischen Energien der einzelnen Komponenten, $l = 1,4$, $i = 1$,**IMAX**, $k = 1$,**JANZ**
$\mathbf{IA}(j)$	Hilfsvektor zum Abspeichern der Intervallnummern, $j = 1$,**MANZ**
MANZ	Anzahl der zu berechnenden Werte der massenspezifischen inneren Energien
MMAX	maximale Anzahl der zu berechnenden Werte der massenspezifischen inneren Energien
IMAX	Anzahl der Komponenten des Gasgemisches
TMIN	minimale Temperatur für Kurvenfits
TMAX	maximale Temperatur für Kurvenfits
DTP	äquidistante Schrittweite der Diskretisierungsintervalle
JANZ	Anzahl der Diskretisierungsintervalle für $\tilde{e}_i(T)$
JMAX	maximale Anzahl der Diskretisierungsintervalle für $\tilde{e}_i(T)$

RGK	universelle Gaskonstante
W(i)	Molmasse der Komponente i, $i = 1$,**IMAX**

Output:

IE(j)	gesamte massenspezifische innere Energie gemäß (14), $j = 1$,**MANZ**

6.1.2 Schnittstelle CVT

Ziel: Berechnung der gesamten massenspezifischen Wärmekapazität bei konstantem Volumen

$$c_V = c_V(Y_1, \ldots, Y_{\mathbf{IMAX}}, T) := \sum_{i=1}^{\mathbf{IMAX}} Y_i c_{V,i}(T)$$

über einen stückweise polynomialen Kurvenfit von

$$\tilde{c}_{V,i}(T) := c_{V,i}(T) \cdot M_i/R = \sum_{l=1}^{4} \mathbf{CVA}(l,i,j) \cdot T^{l-1},\ j = INT\left(\frac{T - \mathbf{TMIN}}{\mathbf{DTP}} + 1\right);$$

Input:

T(j)	Temperatur, $j = 1$,**MANZ**
X(j,i)	Massenbrüche, $i = 1$,**IMAX**, $j = 1$,**MANZ**
CVA(l,i,k)	Koeffizienten der Kurvenfits für die massenspezifischen Wärmekapazitäten der einzelnen Komponenten, $l = 1,4$, $i = 1$,**IMAX**, $k = 1$,**JANZ**
IA(j)	Hilfsvektor zum Abspeichern der Intervallnummern, $j = 1$,**MANZ**
MANZ	Anzahl der zu berechnenden Werte der massenspezifischen Wärmekapazitäten
MMAX	maximale Anzahl der zu berechnenden Werte der massenspezifischen Wärmekapazitäten
IMAX	Anzahl der Komponenten des Gasgemisches
TMIN	minimale Temperatur für Kurvenfits
TMAX	maximale Temperatur für Kurvenfits
DTP	äquidistante Schrittweite der Diskretisierungsintervalle
JANZ	Anzahl der Diskretisierungsintervalle für $\tilde{c}_{V,i}(T)$
JMAX	maximale Anzahl der Diskretisierungsintervalle für $\tilde{c}_{V,i}(T)$
RGK	universelle Gaskonstante
W(i)	Molmasse der Komponente i, $i = 1$,**IMAX**

Output:

CV(j)	gesamte massenspezifische Wärmekapazität, $j = 1$,**MANZ**

6.1.3 Schnittstelle IEI

Ziel: Berechnung der inneren massenspezifischen Energie der Komponente **L** über einen stückweisen polynomialen Fit

$$\tilde{e}_{\mathbf{L}}(T) := e_{\mathbf{L}}(T) \cdot M_{\mathbf{L}}/R/T = \sum_{l=1}^{4} \mathbf{IEA}(l, \mathbf{L}, j) \cdot T^{l-1},\ j = INT\left(\frac{T - \mathbf{TMIN}}{\mathbf{DTP}} + 1\right);$$

Input:

$\mathbf{T}(j)$	Temperatur, $j = 1$,**MANZ**
L	Komponente **L**
$\mathbf{IEA}(l, i, k)$	Koeffizienten der Kurvenfits für die massenspezifische innere Energie der Komponente **L**, $l = 1, 4$, $i = 1$,**IMAX**, $k = 1$,**JANZ**
MANZ	Anzahl der zu berechnenden Werte der massenspezifischen inneren Energie
MMAX	maximale Anzahl der zu berechnenden Werte der massenspezifischen inneren Energien
IMAX	Anzahl der Komponenten des Gasgemisches
TMIN	minimale Temperatur für Kurvenfits
TMAX	maximale Temperatur für Kurvenfits
DTP	äquidistante Schrittweite der Diskretisierungsintervalle
JANZ	Anzahl der Diskretisierungsintervalle für $\tilde{e}_i(T)$
JMAX	maximale Anzahl der Diskretisierungsintervalle für $\tilde{e}_i(T)$
RGK	universelle Gaskonstante
$\mathbf{W}(i)$	Molmasse der Komponente i, $i = 1$,**IMAX**

Output:

$\mathbf{IEK}(j)$	massenspezifische innere Energie der Komponente **L**, $j = 1$,**MANZ**

6.1.4 Schnittstelle CVI

Ziel: Berechnung der massenspezifischen Wärmekapazität bei konstantem Volumen der Komponente **L** über einen stückweisen polynomialen Kurvenfit

$$\tilde{c}_{V,\mathbf{L}}(T) := c_{V,\mathbf{L}}(T) \cdot M_{\mathbf{L}}/R = \sum_{l=1}^{4} \mathbf{CVA}(l, \mathbf{L}, j) \cdot T^{l-1},\ j = INT\left(\frac{T - \mathbf{TMIN}}{\mathbf{DTP}} + 1\right);$$

Input:

$\mathbf{T}(j)$	Temperatur, $j = 1$,**MANZ**

L	Komponente **L**
CVA(l,i,k)	Koeffizienten der Kurvenfits für die massenspezifische Wärmekapazität der Komponente **L**, $l = 1,4$, $i = 1$,**IMAX**, $k = 1$,**JANZ**
MANZ	Anzahl der zu berechnenden Werte der massenspezifischen inneren Energie
MMAX	maximale Anzahl der zu berechnenden Werte der massenspezifischen inneren Energie
IMAX	Anzahl der Komponenten des Gasgemisches
TMIN	minimale Temperatur für Kurvenfits
TMAX	maximale Temperatur für Kurvenfits
DTP	äquidistante Schrittweite der Diskretisierungsintervalle
JANZ	Anzahl der Diskretisierungsintervalle für $\tilde{c}_{V,i}(T)$
JMAX	maximale Anzahl der Diskretisierungsintervalle für $\tilde{c}_{V,i}(T)$
RGK	universelle Gaskonstante
W(i)	Molmasse der Komponente i, $i = 1$,**IMAX**

Output:

CVK(j)	massenspezifische Wärmekapazität der Komponente **L**, $j = 1$,**MANZ**

6.1.5 Schnittstelle TEMP

Ziel: Berechnung der Temperatur als Lösung des Nullstellenproblems

$$e(Y_1, \ldots, Y_{\mathbf{IMAX}}, T) - e = 0$$

über eine Newton-Iteration. Es werden jedoch nur drei Iterationsschritte durchgeführt. Dies ist für eine gute Näherung für die hier verwendeten Kurvenfits von $\tilde{e}_i(T)$ und $\tilde{c}_{V,i}(T)$ ausreichend. Als Starttemperatur hat sich

$$T = e/1200$$

als gut erwiesen.

Input:

X(j,i)	Massenbrüche der Komponenten i, $i = 1$,**IMAX**, $j = 1$,**MANZ**
IE(j)	totale massenspezifische innere Energie **ohne** Bindungsenergie
IEA(l,i,k)	Koeffizienten der Kurvenfits für die inneren massenspezifischen Energien der einzelnen Komponenten, $l = 1,4$, $i = 1$,**IMAX**, $k = 1$,**JANZ**
CVA(l,i,k)	Koeffizienten der Kurvenfits für die massenspezifischen Wärmekapazitäten der einzelnen Komponenten, $l = 1,4$, $i = 1$,**IMAX**, $k = 1$,**JANZ**

IEN(j), **CVN**(j)	Hilfsvektoren zur Berechnung des Newton–Schrittes bei der Temperaturiteration, j = 1,**MANZ**
IA(j)	Hilfsvektor zum Abspeichern der Intervallnummern, j = 1,**MANZ**
MANZ	Anzahl der zu berechnenden Werte der massenspezifischen inneren Energien
MMAX	maximale Anzahl der zu berechnenden Werte der massenspezifischen inneren Energien
IMAX	Anzahl der Komponenten des Gasgemisches
TMIN	minimale Temperatur für Kurvenfits
TMAX	maximale Temperatur für Kurvenfits
DTP	äquidistante Schrittweite der Diskretisierungsintervalle
JANZ	Anzahl der Diskretisierungsintervalle für $\tilde{e}_i(T)$ bzw. $\tilde{c}_{V,i}(T)$
JMAX	maximale Anzahl der Diskretisierungsintervalle für $\tilde{e}_i(T)$ bzw. $\tilde{c}_{V,i}(T)$
RGK	universelle Gaskonstante
W(i)	Molmasse der Komponente i, i = 1,**IMAX**

Output:

T(j)	Temperatur, j = 1,**MANZ**

6.2 Modellierung der chemischen Produktionsraten

6.2.1 Schnittstelle CHEPRO

Ziel: Berechnung von $s_i = s_i(\rho_1, \ldots, \rho_{\mathbf{IMAX}}, T)$

Input:

PRHO(j,i)	Partialdichten $\rho_1, \ldots, \rho_{\mathbf{IMAX}}$ zur j-ten Zusammensetzung j = 0,**ANZ**, i = 1,**IMAX**
T(j)	Temperatur T, j = 0,**ANZ**
ANZ	Anzahl der Zusammensetzungen
DIM	maximale Anzahl an Zusammensetzungen -1
IMAX	Anzahl der Komponenten des Gasgemisches
AKOEFF(l,i,k)	Koeffizienten der Kurvenfits für die Gleichgewichtskonstante, $l = 1, 4$, i = 1,**IMAX**, k = 1,**JANZAK**
TMINAK	minimale Temperatur für den Kurvenfit der Gleichgewichtskonstanten
TMAXAK	maximale Temperatur für den Kurvenfit der Gleichgewichtskonstanten
DTPAK	äquidistante Schrittweite der Diskretisierungsintervalle der Kurvenfits der Gleichgewichtskonstanten

JANZAK Anzahl der Diskretisierungsintervalle der Kurvenfits der Gleichgewichtskonstanten

JMAX maximale Anzahl der Diskretisierungsintervalle aller Kurvenfits

Output:

S(j,i) chemische Produktionsrate der Komponente i,
$j = 0$,**ANZ**, $i = 1$, **IMAX**

Bemerkung 1:

Zur Zeit existiert ein Modell:

CHPROL5 5–Komponenten–Luftmodell
Gasmodell: N, N_2, O, O_2, NO (**IMAX** = 5)
Reaktionen: Dissoziation von N_2, O_2 und NO
Rekombination von O_2 und NO
Gleichgewichtskonstante: Kurvenfit gemäß [10]
Vorwärtsreaktionsraten: Kurvenfit gemäß [15]

Bemerkung 2:

Hier wird die Atomerhaltung explizit ausgenutzt, d.h. es werden beispielsweise für das 5-Komponenten-Luftmodell nur die drei linear unabhängigen Produktionsraten der Moleküle N_2, O_2 und NO berechnet. Die fehlenden zwei werden über die Atomerhaltung ermittelt.

Bemerkung 3:

Die Gleichgewichtskonstanten werden über einen stückweisen polynomialen Kurvenfit berechnet (vgl. **AKOEFF**).

6.3 Transport– und Diffusionsmodell

Zur Berechnung der viskosen Flüsse $\boldsymbol{F}_i^V$ (5) und des Quellterms $\boldsymbol{R}^V$ (28) werden folgende Routinen benötigt:

KOEFF

Berechnung der Transport– und Diffusionskoeffizienten $\mathcal{D}_{ij}, \mu, \lambda$

MDIFF

Berechnung des Diffusionsflusses $\boldsymbol{q}_i^D$

VISCOE

Berechnung des Viskositätskoeffizienten des Gasgemisches μ

DVISCOE

Berechnung von $\frac{d\mu}{dT}$ bei gefrorener Gaszusammensetzung

Letztere beiden werden für die Berechnung des Wanddruckes bei einer reibungsbehafteten Strömung (vgl. Routine **WALLS** in Kap. 4.5.1) benötigt.
Die Berechnung der Transport– und Diffusionkoeffizienten basiert auf Kurvenfits gemäß Kap. 5.2.3 für

- den Viskositätskoeffizient $\mu_i(T)$ der Komponente i eines reinen Gases und
- den binären Diffusionskoeffizienten $\widetilde{\mathcal{D}}_{ik}/(T)$ des binären Gasgemisches bestehend aus den Komponenten i und k.

Damit ist der Wärmeleitungskoeffizient $\lambda_i(T)$ der Komponente i eines reinen Gases eindeutig festgelegt (siehe z.B. [7]).
Mit Hilfe der Mischungsregeln von Wilke für den Viskositätskoeffizienten μ und den Wärmeleitungskoeffizienten λ eines Gasgemisches sowie Obermeier und Schaber für die polynären Diffusionskoeffizienten D_{ik} werden daraus die Koeffizienten für ein Gasgemisch ermittelt.
Die allgemeine Darstellung des Diffusionsflusses $\boldsymbol{q}_i^D$ gemäß [7] (vgl. (6)) kann durch geeignete Annahmen vereinfacht werden (z.B. Gesetz von Fick).
Bisher wurden drei — auf obigem Konzept basierende — Modelle implementiert. Diese sind in [9] beschrieben:

MIXp

Mischungsregeln von Wilke sowie Obermeier und Schaber;

Diffusionsfluß gemäß (6), jedoch ohne thermische Diffusion;

MIX

Mischungsregeln von Wilke sowie Obermeier und Schaber;

Diffusionsfluß gemäß (6), jedoch ohne thermische Diffusion und Druckdiffusion;

SPW

Mischungsregel von Wilke für den Viskositätskoeffizienten μ;

konstante Schmidt– und Prandtl–Zahl zur Berechnung der Diffusionskonstanten D des Fickgesetzes und des Wärmeleitungskoeffizienten λ;

SPA

Kurvenfit für den Viskositätskoeffizienten $\mu = \mu(T)$ aus Mittelung der Kurvenfits für die Viskositätskoeffizienten $\mu_i(T)$ eines reinen Gases und der Gleichgewichtsverteilung der 5–Komponenten–Luft;

konstante Schmidt– und Prandtl–Zahl zur Berechnung der Diffusionskonstanten D des Fickgesetzes und des Wärmeleitungskoeffizienten λ;

Diffusionsfluß gemäß Fick.

Die entsprechenden Modelle sind in den Dateien

CKMIXP.F

CKMIX.F

CKSPW.F

CKSPA.F

abgespeichert. Vor dem Programmstart muß der Benutzer eine der Alternativen auswählen, in die Datei **TRKO.F** kopieren und das Programm durch Aufruf der MAKE-Datei **nscng2.m** neu übersetzen (siehe auch Kap. 10).

6.3.1 Schnittstelle KOEFF

Ziel: Berechnung der Größen

$\mathcal{D}_{ij}$	binärer Diffusionskoeffizient der Komponenten i und j, $i,j = 1$,**IMAX**
μ	Viskositätskoeffizient des Gasgemisches
λ	Wärmeleitungskoeffizient des Gasgemisches

Input:

DIM	Anzahl der zu berechnenden Koeffiziententupel
MMAX	maximale Anzahl an zu berechnenden Koeffiziententupeln
IMAX	Anzahl der Komponenten des Gasgemisches
T(j)	Temperatur, $j = 1$,**DIM**
P(j)	Druck, $j = 1$,**DIM**
C(j)	Summe der Molmassen, $j = 1$,**DIM**
RHO(j)	Massendichte, $j = 1$,**DIM**
X(j,i)	molare Konzentration der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
CPK(j,i)	massenspezifische Wärmekapazität bei konstantem Druck für die Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
MUEI(j,i)	Viskositätskoeffizient der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
KDTI(j,i)	Wärmeleitungskoeffizient der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
PHI(j,l,i)	Hilfsgröße Φ_{ij} bei der Berechnung von μ und λ gemäß der Mischungsregel von Wilke, $j = 1$,**DIM**, $l,i = 1$,**IMAX**
DUM(j,i)	Dummy–Variable zum Speichern von Zwischenergebnissen, $j = 1$,**DIM**, $i = 1$,**IMAX**
MUA(l,i,j)	Koeffizienten der Kurvenfits für die Viskositätskoeffizienten $\mu_i(T)$ gemäß Kap. 5.2.2, $j = 1$,**DIM**, $i = 1$,**IMAX**, $l = 1,5$

JMAX	maximale Anzahl an Diskretisierungsintervallen
JANZMU	tatsächliche Anzahl der Diskretisierungsintervalle für $\mu_i(T)$
TMINMU	minimale Temperatur für $\mu_i(T)$
TMAXMU	maximale Temperatur für $\mu_i(T)$
DTPMU	äquidistante Schrittweite für $\mu_i(T)$
BDA(l,i,k,j)	Koeffizienten der Kurvenfits für die binären Diffusionskoeffizienten $\widetilde{\mathcal{D}}_{ik}(T)$ gemäß Kap. 5.2.3, $j = 1$,**DIM**, $i,k = 1$,**IMAX**, $l = 1,5$
JANZBD	tatsächliche Anzahl an Diskretisierungsintervallen für $\widetilde{\mathcal{D}}_{ij}(T)$
TMINBD	minimale Temperatur für $\widetilde{\mathcal{D}}_{ij}(T)$
TMAXBD	maximale Temperatur für $\widetilde{\mathcal{D}}_{ij}(T)$
DTPBD	äquidistante Schrittweite für $\widetilde{\mathcal{D}}_{ij}(T)$
IA(j)	Hilfsvektor zur Auswertung von $\widetilde{\mathcal{D}}_{ij}(T), \mu_i(T)$, $j = 1$,**DIM**
RGK	universelle Gaskonstante
W(i)	Molmasse der Komponente i, $i = 1$,**IMAX**

Output:

MUE(j)	Viskositätskoeffizient $\mu(T, X_1, \ldots, X_{\mathbf{IMAX}})$ des Gasgemisches, $j = 1$,**DIM**
KDT(j)	Wärmeleitungskoeffizient $\lambda(T, X_1, \ldots, X_{\mathbf{IMAX}})$ des Gasgemisches, $j = 1$,**DIM**
BD(j,i,k)	binärer Diffusionskoeffizient $\mathcal{D}_{ik}$ der Komponenten i und k, $j = 1$,**DIM**, $i,k = 1$,**IMAX**
MD(j,i,k)	polynärer Diffusionskoeffizient D_{ik} der Komponenten i und k, $j = 1$,**DIM**, $i,k = 1$,**IMAX**

6.3.2 Schnittstelle MDIFF

Ziel: Berechnung der Diffusionsflüsse $\boldsymbol{q}_1^D, \boldsymbol{q}_2^D$ analog zu (6) $i = 1$,**IMAX**

Input:

DIM	Anzahl der zu berechnenden Diffusionsflüsse
MMAX	maximale Anzahl zu berechnender Diffusionsflüsse
IMAX	Anzahl der Komponenten des Gasgemisches
X(j,i)	molare Konzentration der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
PRHO(j,i)	Partialdichte der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
RHO(j)	Massendichte, $j = 1$,**DIM**
P(j)	Druck, $j = 1$,**DIM**

$\mathbf{C}(j)$	Summe der Molmassen, $j = 1$,**DIM**
$\mathbf{CX}(j,i)$	molares Konzentrationsgefälle in x_1–Richtung der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
$\mathbf{CY}(j,i)$	molares Konzentrationsgefälle in x_2–Richtung der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
$\mathbf{PX}(j)$	Druckgefälle in x_1–Richtung, $j = 1$,**DIM**
$\mathbf{PY}(j)$	Druckgefälle in x_2–Richtung, $j = 1$,**DIM**
$\mathbf{DX}(j)$	Hilfsvektor zur Berechnung von $\boldsymbol{q}^D_{1,i}$, $j = 1$,**DIM**
$\mathbf{DY}(j)$	Hilfsvektor zur Berechnung von $\boldsymbol{q}^D_{2,i}$, $j = 1$,**DIM**
$\mathbf{XERROR}(j)$	Abweichung der Summe der Diffusionsflüsse $\boldsymbol{q}^D_{1,i}$ von Null (Verletzung der Massenerhaltung), $j = 1$,**DIM**
$\mathbf{YERROR}(j)$	Abweichung der Summe der Diffusionsflüsse $\boldsymbol{q}^D_{2,i}$ von Null (Verletzung der Massenerhaltung), $j = 1$,**DIM**
$\mathbf{BD}(j,i,k)$	binärer Diffusionskoeffizient $\mathcal{D}_{ik}$ der Komponenten i und k, $j = 1$,**DIM**, $i,k = 1$,**IMAX**
$\mathbf{MD}(j,i,k)$	polynärer Diffusionskoeffizient D_{ik} der Komponenten i und k, $j = 1$,**DIM**, $i,k = 1$,**IMAX**

Output:

$\mathbf{QDX}(j,i)$	Diffusionsfluß $\boldsymbol{q}^D_{1,i}$ der Komponente i in x_1–Richtung, $j = 1$,**DIM**, $i = 1$,**IMAX**
$\mathbf{QDY}(j,i)$	Diffusionsfluß $\boldsymbol{q}^D_{2,i}$ der Komponente i in x_2–Richtung, $j = 1$,**DIM**, $i = 1$,**IMAX**

6.3.3 Schnittstelle VISCOE

Ziel: Berechnung des Viskositätskoeffizienten

$$\mu(T, X_1, \ldots, X_{\mathbf{IMAX}})$$

des Gasgemisches identisch zu der Berechnung in **KOEFF**

Input:

DIM	Anzahl der zu berechnenden Koeffiziententupel
MMAX	maximale Anzahl an zu berechnenden Koeffiziententupeln
IMAX	Anzahl der Komponenten des Gasgemisches
$\mathbf{T}(j)$	Temperatur, $j = 1$,**DIM**
$\mathbf{X}(j,i)$	molare Konzentration, $j = 1$,**DIM**, $i = 1$,**IMAX**
$\mathbf{MUEI}(j,i)$	Viskositätskoeffizient der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**

PHI(j,l,i)	Hilfsgröße Φ_{ij} bei der Berechnung von μ und λ gemäß der Mischungsregel von Wilke, $j = 1$,**DIM**, $l,i = 1$,**IMAX**
MUA(l,i,j)	Koeffizienten der Kurvenfits für die Viskositätskoeffizienten $\mu_i(T)$ gemäß Kap. 5.2.2, $j = 1$,**DIM**, $i = 1$,**IMAX**, $l = 1,5$
JMAX	maximale Anzahl an Diskretisierungsintervallen
JANZMU	tatsächliche Anzahl der Diskretisierungsintervalle für $\mu_i(T)$
TMINMU	minimale Temperatur für $\mu_i(T)$
TMAXMU	maximale Temperatur für $\mu_i(T)$
DTPMU	äquidistante Schrittweite für $\mu_i(T)$
IA(j)	Hilfsvektor zur Auswertung von $\widetilde{\mathcal{D}}_{ij}(T), \mu_i(T)$, $j = 1$,**DIM**
W(i)	Molmasse der Komponente i, $i = 1$,**IMAX**
DUM(j,i)	Dummy–Variable zum Speichern von Zwischenergebnissen, $j = 1$,**DIM**, $i = 1$,**IMAX**

Output:

MUE(j)	Viskositätskoeffizient $\mu(T, X_1, \ldots, X_{\mathbf{IMAX}})$ des Gasgemisches, $j = 1$,**DIM**

6.3.4 Schnittstelle DVISCO

Ziel: Berechnung der Ableitung des Viskositätskoeffizienten nach der Temperatur bei gefrorenen Konzentrationen

$$\left(\frac{\partial \mu}{\partial T}\right)_{X_i=\text{const.}}$$

Input:

DIM	Anzahl der zu berechnenden Koeffiziententupel
MMAX	maximale Anzahl an zu berechnenden Koeffiziententupeln
IMAX	Anzahl der Komponenten des Gasgemisches
T(j)	Temperatur, $j = 1$,**DIM**
X(j,i)	molare Konzentration der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
MUEI(j,i)	Viskositätskoeffizient der Komponente i, $j = 1$,**DIM**, $i = 1$,**IMAX**
PHI(j,l,i)	Hilfsgröße Φ_{ij} bei der Berechnung von μ und λ gemäß der Mischungsregel von Wilke, $j = 1$,**DIM**, $l,i = 1$,**IMAX**
MUA(l,i,j)	Koeffizienten der Kurvenfits für die Viskositätskoeffizienten $\mu_i(T)$ gemäß Kap. 5.2.2, $j = 1$,**DIM**, $i = 1$,**IMAX**, $l = 1,5$
JMAX	maximale Anzahl an Diskretisierungsintervallen
JANZMU	tatsächliche Anzahl der Diskretisierungsintervalle für $\mu_i(T)$

TMINMU	minimale Temperatur für $\mu_i(T)$
TMAXMU	maximale Temperatur für $\mu_i(T)$
DTPMU	äquidistante Schrittweite für $\mu_i(T)$
IA(j)	Hilfsvektor zur Auswertung von $\widetilde{\mathcal{D}}_{ij}(T), \mu_i(T)$, $j = 1$,**DIM**
W(i)	Molmasse der Komponente i, $i = 1$,**IMAX**
DUM(j,i)	Dummy–Variable zum Speichern von Zwischenergebnissen, $j = 1$,**DIM**, $i = 1$,**IMAX**
DMUEI(j,i)	Viskositätskoeffizient der Komponente i nach der Temperatur abgeleitet bei gefrorenen Konzentrationen, $j = 1$,**DIM**, $i = 1$,**IMAX**
DPHI(j,l,i)	Ableitung der Hilfsgröße Φ_{ij} bei der Berechnung von μ gemäß der Mischungsregel von Wilke, $j = 1$,**DIM**, $l,i = 1$,**IMAX**

Output:

DMUE(j)	Änderung des Viskositätskoeffizienten μ des Gasgemisches bezüglich der Temperatur bei gefrorenen Konzentrationen, $j = 1$,**DIM**

7 Globale Variablen

In FORTRAN 77 ist es nicht möglich, dynamisch Speicherplatz zu allokieren, d.h. die Dimensionierung von Arrays darf sich während des Programmablaufs nicht ändern. Das hat zur Folge, daß alle in einem Unterprogramm benötigten Arrays mit entsprechender Dimensionierung dort deklariert werden müssen. Daraus resultieren sehr lange Parameterlisten. Dies kann vermieden werden, in dem man die entsprechende Größen in einer Datei deklariert, diese in COMMON–Blöcken zusammenfaßt und dann als INCLUDE–Datei in das Unterprogramm einbindet. Es ist jedoch nicht sinnvoll, für jedes Unterprogramm eine solche Datei anzulegen. Dies würde bei großen Programmen, die in der Regel aus vielen Routinen bestehen, die Anzahl der Dateien unnötig erhöhen.
Es ist deshalb angebracht, nur solche Variablen und Parameter in einigen wenigen INCLUDE–Dateien zusammenzufassen, die in verschiedenen Routinen benötigt werden. Dadurch sind jedoch die Unterprogramme nicht mehr ohne weiteres austauschbar. Man muß sich also sehr genau überlegen, ob man alle Variablen über eine eventuell sehr lange Parameterliste übergibt oder über INCLUDE–Dateien einbindet.
In dem hier dokumentierten Programm wurden beide Strategien verwendet. Dabei wurde die Unterteilung nach folgenden Aspekten durchgeführt:

INCLUDE–Dateien:

Variablen und Parameter, auf die in verschiedenen Unterroutinen zugegriffen wird;

Variablen und Parameter, die zur numerischen Diskretisierung benötigt werden;

Parameterliste:

Variablen und Parameter, die den Programmablauf steuern;

Unterprogramme, die das thermodynamische Modell beschreiben, wie

- Energiemodell
- Modell der chemischen Produktionsraten
- Modell der Transport– und Diffusionsphänomene (vgl. Kap. 6)

Unterprogramme, die vom Benutzer leicht durch Alternativen zu ersetzen sein sollten.

Im folgenden sollen die globalen Variablen, nach INCLUDE–Dateien sortiert, beschrieben werden.

7.1 INCLUDE–Datei FIXDEK

In dieser Datei sind alle Parameter und Variablen zusammengefaßt, die

- zur Dimensionierung der Arrays
- zur Steuerung des physikalischen Modells zur Programmlaufzeit (∗**FLAG**)
- zur Festlegung der Strömung (∗**REF**)

benötigt werden. Diese Datei muß immer zusätzlich zu den anderen INCLUDE–Dateien eingebunden werden, da hier die Array–Dimensionierung festgelegt ist.
Die einzelnen Variablen und Parameter sind:

DIMMAX Anzahl der Komponenten des Erhaltungsvektors $(\rho, \rho\boldsymbol{v}, \rho E)$, (hier: 4, da 2D)

NMAX maximale Interpolationsordnung (≥ 1)

MXMAX maximale Anzahl von Punkten entlang der Parameterkurve in ξ–Richtung ($\geq$ **MZAHLX** $+ 1$)

MYMAX maximale Anzahl von Punkten entlang der Parameterkurve in η–Richtung ($\geq$ **MZAHLY** $+ 1$)

MMAX Maximum von **MXMAX** und **MYMAX**

IMAX Anzahl der Komponenten des Gasgemisches

RMAX Anzahl der verschiedenen Gleichgewichtskonstanten

IOFORM Ausgabeformat der Daten

SPEZIE(i) Komponente i des Gasgemisches, $i = 1$,**IMAX**

REAKT(i) Reaktionsklasse i, die mit der gleichen Gleichgewichtskonstante einhergeht, in der gleichen Reihenfolge abgespeichert, wie diese in **CHEPRO** berechnet werden, $i = 1$,**RMAX**

RGK universelle Gaskonstante

TREF Temperatur in der Anströmung

PREF Druck in der Anströmung

XREF(i) Massenbruch der Komponente i in der Anströmung, $i = 1$,**IMAX**

RHOREF Massendichte in der Anströmung

MREF Impuls in x_1–Richtung in der Anströmung

NREF Impuls in x_2–Richtung in der Anströmung

REREF Totalenergie in der Anströmung

FEREF Bindungsenergie in der Anströmung

$$\sum_{i=1}^{\mathbf{IMAX}} Y_i e_i^0$$

TWALL(i) Körpertemperatur an der nichtadiabaten Wand Γ_i, $i = 1, 4$ (vgl. Kap. 4.5.1)

XWALL(i) Massenbruch an der vollkatalytischen Wand, $i = 1, \mathbf{IMAX}$ (vgl. Kap. 4.5.1)

NSFLAG Wahl der Strömung:

reibungsfrei	(0)
reibungsbehaftet	(1)

CFLAG chemische Modellierung des Gasgemisches:

inert	(0)
chemisches Nichtgleichgewicht	(1)

OFLAG Operatorsplitting für den Quellterm der chemischen Produktionsraten (vgl. (29))

ohne Operatorsplitting	(0)
mit Operatorsplitting	(1)

GFLAG Wahl der Geometrie (vgl. Kap. 5.2.1)

eben	(0)
quasi–2D, rotationssymmetrisch	(1)
eben, achsensymmetrisch	(2)

WFLAG thermische Modellierung der Körperwand

adiabat	(0)
nicht adiabat	(1)

KWFLAG katalytische Modellierung der Körperwand

voll katalytisch	(0)
nicht katalytisch	(1)

RFLAG(i) Zuordnung eines Randtyps zu dem Rand Γ_i, i, $i = 1, 4$

Hypersonischer Inflow	(0)
Hypersonischer Outflow	(1)
Körperwand (Haftbedingung, Navier–Stokes)	(3)
Körperwand (Verdünnungswelle, Euler)	(4)
Körperwand (Reflexion, Euler)	(5)
Symmetrieachse (Rotationssymmetrie, nur Γ_3 und $x_2 = 0$)	(6)
Symmetrieachse (Achsensymmetrie, nur Γ_3 und $x_2 = 0$)	(7)

KFLAG(i) Charakterisierung des Randes Γ_i, $i = 1, 4$

freier Strömungsrand	(0)
feste Körperwand	(1)

PFLAG Überprüfung auf positiven Druck

ohne Test	(0)
mit Test	(1)

7.2 INCLUDE–Datei FLDEK

Hier sind alle globalen Variablen zusammengefaßt, die benötigt werden für:

- Mittelwerte
- numerische Flüsse
- Quellterm der chemischen Produktionsraten
- Dateinamen zur Steuerung der Ein– und Ausgabe.

ASTEPS	Anzahl der schon durchgeführten Zeitschritte
NSTEPS	Anzahl der in diesem Programmlauf auszuführenden Zeitschritte
ZSTEPS	Anzahl der Zeitschritte, nach denen die Daten zwischengespeichert werden sollen
SSTEPS	Anzahl der Zeitschritte, nach denen überprüft werden soll, ob die Strömung stationär ist
DATINP	Dateiname der Daten zum alten Zeitpunkt $t = t_{\mathbf{ASTEPS}}$
DATZWI	Dateiname zum Zwischenspeichern der Daten
DATOUT	Dateiname der Daten zum neuen Zeitpunkt $t = t_{\mathbf{ASTEPS+NSTEPS}}$
REFQUA	Dateiname für die Referenzgrößen der Strömung
MZAHLX	Anzahl der Zellen entlang der Parameterkurve in ξ–Richtung -3
MZAHLY	Anzahl der Zellen entlang der Parameterkurve in η–Richtung -3
DELTAT	Zeitschrittweite
MW(j,k,i)	Mittelwerte der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ in der Zelle $V_{j,k}$ $j = -1, \mathbf{MZAHLX}+1$, $k = -1, \mathbf{MZAHLY}+1$, $i = 1,4$
MWI(j,i,l)	Mittelwerte der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ am Rand l innerhalb des Rechengebietes (wird nur für das zonale Konzept benötigt) $j = 0, \mathbf{MMAX}$, $i = 1, \mathbf{DIMMAX}$, $l = 1,4$
MWA(j,i,l)	Mittelwerte der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ am Rand l außerhalb des Rechengebietes (wird nur für das zonale Konzept benötigt) $j = 0, \mathbf{MMAX}$, $i = 1, \mathbf{DIMMAX}$, $l = 1,4$
MWALT(j,k,i)	wie **MW**, aber zum alten Zeitpunkt
MWRHO(j,k,i)	Mittelwerte der Partialdichten ρ_i in der Zelle $V_{j,k}$ $j = -1, \mathbf{MZAHLX}+1$, $k = -1, \mathbf{MZAHLY}+1$, $i = 1$,**IMAX**

MWRHOI(j,i,l) Mittelwerte der Partialdichten ρ_i am Rand l innerhalb des Rechengebietes (wird nur für das zonale Konzept benötigt)

$j = 0, \mathbf{MMAX}$, $i = 1, \mathbf{IMAX}$, $l = 1, 4$

MWRHOA(j,i,l) Mittelwerte der Partialdichten ρ_i am Rand l außerhalb des Rechengebietes (wird nur für das zonale Konzept benötigt)

$j = 0, \mathbf{MMAX}$, $i = 1, \mathbf{IMAX}$, $l = 1, 4$

MWX(j,k,i) Mittelwerte der Massenbrüche Y_i in der Zelle $V_{j,k}$

$j = -1, \mathbf{MZAHLX} + 1$, $k = -1, \mathbf{MZAHLY} + 1$, $i = 1$,**IMAX**

MWT(j,k) Mittelwerte der Temperatur in der Zelle $V_{j,k}$

$j = -1, \mathbf{MZAHLX} + 1$, $k = -1, \mathbf{MZAHLY} + 1$

MWFE(j,k) Mittelwerte der Bildungsenergie $\sum_{i=1}^{\mathbf{IMAX}} Y_i e_i^0$ in der Zelle $V_{j,k}$

$j = -1, \mathbf{MZAHLX} + 1$, $k = -1, \mathbf{MZAHLY} + 1$

VFXI(j,k,i) i–te Komponente des viskosen Flusses zu $\rho, \rho\boldsymbol{v}, \rho E$ in x_1–Richtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_4^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in ξ–Richtung

$j = -1, \mathbf{MZAHLX}$, $k = 0, \mathbf{MZAHLY} + 1$, $i = 1$,**DIMMAX**

VGXI(j,k,i) i–te Komponente des viskosen Flusses zu $\rho, \rho\boldsymbol{v}, \rho E$ in x_2–Richtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_4^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in ξ–Richtung

$j = -1, \mathbf{MZAHLX}$, $k = 0, \mathbf{MZAHLY} + 1$, $i = 1$,**DIMMAX**

VFETA(j,k,i) i–te Komponente des viskosen Flusses zu $\rho, \rho\boldsymbol{v}, \rho E$ in x_1–Richtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_1^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in η–Richtung

$j = 0, \mathbf{MZAHLX}$, $k = -1, \mathbf{MZAHLY}$, $i = 1$,**DIMMAX**

VGETA(j,k,i) i–te Komponente des viskosen Flusses zu $\rho, \rho\boldsymbol{v}, \rho E$ in x_2–Richtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_1^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in η–Richtung

$j = 0, \mathbf{MZAHLX}$, $k = -1, \mathbf{MZAHLY}$, $i = 1$,**DIMMAX**

VFXID(j,k,i) i–te Komponente des viskosen Massenflusses von ρ_i in x_1–Richtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_4^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in ξ–Richtung

$j = -1, \mathbf{MZAHLX}$, $k = 0, \mathbf{MZAHLY}$, $i = 1$,**IMAX**

VGXID(j,k,i) i–te Komponente des viskosen Massenflusses von ρ_i in x_2–Richtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_4^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in ξ–Richtung

$j = -1,$**MZAHLX**, $k = 0,$**MZAHLY**, $i = 1,$**IMAX**

VFETAD(j,k,i) i–te Komponente des viskosen Massenflusses von ρ_i in x_1–Richtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_1^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in η–Richtung

$j = 0,$**MZAHLX**, $k = -1,$**MZAHLY**, $i = 1,$**IMAX**

VGETAD(j,k,i) i–te Komponente des viskosen Massenflusses von ρ_i in x_2–Richtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_1^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in η–Richtung

$j = 0,$**MZAHLX**, $k = -1,$**MZAHLY**, $i = 1,$**IMAX**

NFLXI(j,k,i) i–te Komponente des numerischen inviskosen Flusses zu $\rho, \rho\boldsymbol{v}, \rho E$ in Normalenrichtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_4^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in ξ–Richtung

$j = -1,$**MZAHLX**, $k = 0,$**MZAHLY**, $i = 1,$**DIMMAX**

NFLETA(j,k,i) i–te Komponente des numerischen inviskosen Flusses zu $\rho, \rho\boldsymbol{v}, \rho E$ in Normalenrichtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_1^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in η–Richtung

$j = 0,$**MZAHLX**, $k = -1,$**MZAHLY**, $i = 1,$**DIMMAX**

CNFXI(j,k,i) i–te Komponente des numerischen inviskosen Flusses zu ρ_i in Normalenrichtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_4^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in ξ–Richtung

$j = 0,$**MZAHLX**, $k = -1,$**MZAHLY**, $i = 1,$**IMAX**

CNFETA(j,k,i) i–te Komponente des numerischen inviskosen Flusses zu ρ_i in Normalenrichtung entlang den Zellrandmittelpunkten $\boldsymbol{P}_1^{j,k}$ der Zelle $V_{j,k}$ einer Parameterkurve in η–Richtung

$j = -1,$**MZAHLX**, $k = 0,$**MZAHLY**, $i = 1,$**IMAX**

QUELLE(j,k,i) Quellterm der chemischen Produktionsraten in der Zelle $V_{j,k}$

$j = 0,$**MZAHLX**, $k = 0,$**MZAHLY**, $i = 1,$**IMAX**

RQ(j,k,i) Quellterm $\boldsymbol{Q}$ gemäß (55) in den Punkten $\boldsymbol{P}_1^{j,k}$ bzw. $\boldsymbol{P}_4^{j,k}$

$j = -1,$**MZAHLX** $+ 1$, $k = -1,$**MZAHLY** $+ 1$, $i = 1, 3$

7.3 INCLUDE–Datei NETDEK

Zur Berechnung der Mittelwerte werden gitterabhängige Informationen wie

- Kantenlänge einer Zelle
- Eckpunkte einer Zelle
- Normale zu Zellrändern
- Volumen einer Zelle

benötigt. Diese sind hier zusammengefaßt:

GITTER Dateiname für das Gitter

GRID(j,k,i) Koordinate i eines Zelleckpunktes
$j = -2, \mathbf{MZAHLX}+1,\ k = -2, \mathbf{MZAHLX}+1,\ i = 1,2$

NXI(j,k,i) Koordinate i der äußeren Normalen im Zellrandmittelpunkt $\boldsymbol{P}_4^{j,k}$ zur Zelle $V_{j,k}$ einer Parameterkurve in ξ–Richtung
$j = -1, \mathbf{MZAHLX},\ k = 0, \mathbf{MZAHLY},\ i = 1,2$

NETA(j,k,i) Koordinate i der äußeren Normalen im Zellrandmittelpunkt $\boldsymbol{P}_1^{j,k}$ zur Zelle $V_{j,k}$ einer Parameterkurve in η–Richtung
$j = 0, \mathbf{MZAHLX},\ k = -1, \mathbf{MZAHLY},\ i = 1,2$

ARCXI(j,k) Länge des Zellrandes $\Gamma_4^{j,k}$,
$j = -1$,**MZAHLX**, $k = 0$,**MZAHLY**

ARCETA(j,k) Länge des Zellrandes $\Gamma_1^{j,k}$,
$j = 0$,**MZAHLX**, $k = -1$,**MZAHLY**

VOL(j,k) Volumen der Zelle $V_{j,k}$,
$j = -1, \mathbf{MZAHLX}+1,\ k = -1, \mathbf{MZAHLY}+1$

Es sei bemerkt, daß die Zelle $V_{j,k}$ durch die folgenden vier Gitterpunkte

$$\begin{array}{ll} \mathbf{GRID}(j-1,k-1,i) & \mathbf{GRID}(j-1,k,i) \\ \mathbf{GRID}(j,k-1,i) & \mathbf{GRID}(j,k,i) \end{array}$$

$$j = -1, \mathbf{MZAHLX}+1, \quad k = -1, \mathbf{MZAHLY}+1, \quad i = 1,2$$

festgelegt ist.

7.4 INCLUDE–Datei REKDEK

In dieser Datei stehen alle Variablen, die zur Bestimmung des Rekonstruktionspolynoms benötigt werden (vgl. Kap. 4.5.3). Hier wurde nur die eindimensionale, komponentenweise RP–Rekonstruktion entlang einer Parameterkurve durch die Zellrandmittelpunkte implementiert. Diese ist in [1], Kapitel 3 ausführlich beschrieben. Die Berechnung erfolgt für jede Parameterkurve getrennt. Die Variablen haben somit nur einen Raumindex:

ZWNMAX $2 * \mathbf{NMAX}$

MZAHL Anzahl der Zellen entlang einer Parameterkurve in ξ– bzw. η–Richtung -3, d.h.

$\mathbf{MZAHL} = \mathbf{MZAHLX}$ oder $\mathbf{MZAHLY}$

POLGRA Grad des Rekonstruktionspolynoms $+1$, d.h. Ordnung des Rekonstruktionspolynoms

MPKXI(j,k) Parametrisierung einer Parameterkurve in ξ–Richtung, d.h. $\eta = k =$ const. über den euklidischen Abstand der Zellrandmittelpunkte $\boldsymbol{P}_2^{j,k}$ und $\boldsymbol{P}_4^{j,k}$

$j = -\mathbf{POLGRA}, \mathbf{MZAHLX} + \mathbf{POLGRA}$

$k = -\mathbf{POLGRA}, \mathbf{MZAHLY} + \mathbf{POLGRA}$

MPKETA(j,k) Parametrisierung einer Parameterkurve in η–Richtung, d.h. $\xi = j =$ const. über den euklidischen Abstand der Zellrandmittelpunkte $\boldsymbol{P}_3^{j,k}$ und $\boldsymbol{P}_1^{j,k}$

$j = -\mathbf{POLGRA}, \mathbf{MZAHLX} + \mathbf{POLGRA}$

$k = -\mathbf{POLGRA}, \mathbf{MZAHLY} + \mathbf{POLGRA}$

APPROX(l,i) Mittelwerte der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ entlang einer Parameterkurve

$l = -\mathbf{POLGRA}, \mathbf{MZAHL} + \mathbf{POLGRA}$, $i = 1, \mathbf{DIMMAX}$

MPKTE(l) parametrische Stützstellen einer Parameterkurve gemäß **MPKXI** bzw. **MPKETA**

$l = -\mathbf{POLGRA}, \mathbf{MZAHL} + \mathbf{POLGRA}$

REKPOL(l,i,k) k–ter Koeffizient des Rekonstruktionspolynoms zum Intervall l der i–ten Komponente des Erhaltungsvektors $\rho, \rho\boldsymbol{v}, \rho E$

$l = -1, \mathbf{MZAHL} + 1$, $k = 0, \mathbf{POLGRA}$, $i = 1, \mathbf{DIMMAX}$

KMIN(l,i) kleinste Stützstelle des Rekonstruktionspolynoms zum Intervall l der i–ten Komponente des Erhaltungsvektors $\rho, \rho\boldsymbol{v}, \rho E$

$l = -1, \mathbf{MZAHL} + 1$, $i = 1, \mathbf{DIMMAX}$

7.5 INCLUDE–Datei RPDEK

Alle Variablen, die zur Berechnung der Anfangszustände für die Riemannprobleme in den Zellrandmittelpunkten benötigt werden (vgl. (48),(49)), sind in dieser INCLUDE–Datei deklariert.

NORMAL(l,i) Normalenvektor in den Zellrandmittelpunkten einer Parameterkurve in ξ– bzw. η–Richtung

$l=-1$,**MZAHL**, $i=1,2$

UREKR(l,i) rechtsseitiger Grenzwert der Rekonstruktionspolynome zu $\rho, \rho\boldsymbol{v}, \rho E$ in den Zellrandmittelpunkten einer Parameterkurve in ξ– bzw. η–Richtung

$l=-1$,**MZAHL**, $i=1$,**DIMMAX**

UREKL(l,i) linksseitiger Grenzwert der Rekonstruktionspolynome zu $\rho, \rho\boldsymbol{v}, \rho E$ in den Zellrandmittelpunkten einer Parameterkurve in ξ– bzw. η–Richtung

$l=-1$,**MZAHL**, $i=1$,**DIMMAX**

XREKR(l,i) zu **UREKR**$(\cdot,l)$ gehörender Massenbruch Y_i

$l=-1$,**MZAHL**, $i=1$,**IMAX**

XREKL(l,i) zu **UREKL**$(\cdot,l)$ gehörender Massenbruch Y_i

$l=-1$,**MZAHL**, $i=1$,**IMAX**

FERKR(l) zu **UREKR**$(\cdot,l)$, **XREKR**$(\cdot,l)$ gehörende Bindungsenergie $\sum Y_i e_i^0$

$l=-1$,**MZAHL**

FERKL(l) zu **UREKL**$(\cdot,l)$, **XREKL**$(\cdot,l)$ gehörende Bindungsenergie $\sum Y_i e_i^0$

$l=-1$,**MZAHL**

UREK(l,i,n) n–te Ableitung der i–ten Komponente des Vektors der Rekonstruktionspolynome in den Zellrandmittelpunkten entlang einer Parameterkurve in ξ– bzw. η–Richtung

$l=0,$**MZAHL** $+1$, $i=1$,**DIMMAX**, $n=0$,**POLGRA**

XREK(l,i) zu **UREK**$(\cdot,0,l)$ gehörender Massenbruch Y_i

$l=0,$**MZAHL** $+1$, $i=1$,**IMAX**

FFL1(l,i,n) erste partielle Ableitung der i–ten Komponente des reibungsfreien Flusses in x_1–Richtung nach der n–ten Komponente von $(\rho, \rho\boldsymbol{v}, \rho E)$

$l=0,$**MZAHL** $+1$, $i,n=1$,**DIMMAX**

GFL1(l,i,n) erste partielle Ableitung der i–ten Komponente des reibungsfreien Flusses in x_2–Richtung nach der n–ten Komponente von $(\rho, \rho\boldsymbol{v}, \rho E)$

$l=0,$**MZAHL** $+1$, $i,n=1$,**DIMMAX**

PR(l) Druck bezüglich **XREK**$(l,\cdot)$, **UREK**$(\cdot,0,l)$

PR1(l,i) erste partielle Ableitung des Druckes bei konstanten Konzentrationen (gefrorenes Gas) nach der i–ten Komponente von $\rho, \rho\boldsymbol{v}, \rho E$

$l = 0,$**MZAHL** $+1$, $i = 1,$**DIMMAX**

DUDT(l,i,n) n–te partielle Ableitung der i–ten Komponente der Erhaltungsvektors $\rho, \rho\boldsymbol{v}, \rho E$ des Problems (48) an der Stelle **UREK**$(\cdot,0,l)$ nach der Zeit (vgl. (49))

$l = 0,$**MZAHL** $+1$, $i = 1,$**DIMMAX**, $n = 0,1$

ULXI(j,k,i) i–te Komponente des linken Anfangszustandes der Größen $\rho, \rho\boldsymbol{v}, \rho E$ für das Riemannproblem in den Zellrandmittelpunkten, die auf einer Parameterkurve in ξ–Richtung liegen

$j = -1,$**MZAHLX**, $k = 0,$**MZAHLY**, $i = 1,$**DIMMAX**

URXI(j,k,i) i–te Komponente des rechten Anfangszustandes der Größen $\rho, \rho\boldsymbol{v}, \rho E$ für das Riemannproblem in den Zellrandmittelpunkten, die auf einer Parameterkurve in ξ–Richtung liegen

$j = -1,$**MZAHLX**, $k = 0,$**MZAHLY**, $i = 1,$**DIMMAX**

ULETA(j,k,i) i–te Komponente des linken Anfangszustandes der Größen $\rho, \rho\boldsymbol{v}, \rho E$ für das Riemannproblem in den Zellrandmittelpunkten, die auf einer Parameterkurve in η–Richtung liegen

$j = 0,$**MZAHLX**, $k = -1,$**MZAHLY**, $i = 1,$**DIMMAX**

URETA(j,k,i) i–te Komponente des rechten Anfangszustandes der Größen $\rho, \rho\boldsymbol{v}, \rho E$ für das Riemannproblem in den Zellrandmittelpunkten, die auf einer Parameterkurve in η–Richtung liegen

$j = 0,$**MZAHLX**, $k = -1,$**MZAHLY**, $i = 1,$**DIMMAX**

7.6 INCLUDE–Datei CHEDEK

Alle Größen, die zur Berechnung des physikalischen Modells benötigt werden, wie

- Koeffizienten der Kurvenfits für die massenspezifische innere Energie und die massenspezifische Wärmekapazität bei konstantem Volumen
- Koeffizienten der Kurvenfits für die Gleichgewichtskonstanten der einzelnen Reaktionsklassen
- Koeffizienten der Kurvenfits für den Viskositätskoeffizienten der Komponente i eines reinen Gases
- Koeffizienten der Kurvenfits für die binären Diffusionskoeffizienten der Komponenten i und j eines binären Gasgemisches

- Molmassen der einzelnen Komponenten eines Gasgemisches
- Formationsenthalpien der einzelnen Komponenten eines Gasgemisches

sind hier deklariert:

IEA(l,i,k) l–ter Koeffizient des stückweisen polynomialen Kurvenfits der massenspezifischen inneren Energie $\tilde{e}_i(T)$ zur i–ten Komponente des Gasgemisches im k–ten Temperaturintervall (vgl. Kap. 5.2.3)

$l = 1,4$, $i = 1$,**IMAX**, $k = 1$,**JANZ**

CVA(l,i,k) l–ter Koeffizient des stückweisen polynomialen Kurvenfits der massenspezifischen isochoren Wärmekapazitäten $\tilde{c}_{V,i}(T)$ zur i–ten Komponente des Gasgemisches im k–ten Temperaturintervall (vgl. Kap. 5.2.3)

$l = 1,4$, $i = 1$,**IMAX**, $k = 1$,**JANZ**

JANZ Anzahl an Temperaturintervallen der Kurvenfits für $\tilde{e}_i(T)$ bzw. $\tilde{c}_{V,i}(T)$; diese muß für beide Fits identisch sein

DTP äquidistante Intervallänge der Diskretisierung des Temperaturbereiches der Kurvenfits zu $\tilde{e}_i(T)$ bzw. $\tilde{c}_{V,i}(T)$

TMIN untere Temperatur des Diskretisierungsbereiches der Kurvenfits $\tilde{e}_i(T)$ bzw. $\tilde{c}_{V,i}(T)$

TMAX obere Temperatur des Diskretisierungsbereiches der Kurvenfits $\tilde{e}_i(T)$ bzw. $\tilde{c}_{V,i}(T)$

MUA(l,i,k) l–ter Koeffizient des stückweisen polynomialen Kurvenfits der Komponentenviskosität $\mu_i(T)$ zur i–ten Komponente des Gasgemisches im k–ten Temperaturintervall (vgl. Kap. 5.2.3)

$l = 1,5$, $i = 1$,**IMAX**, $k = 1$,**JANZMU**

JANZMU Anzahl an Temperaturintervallen der Kurvenfits für $\mu_i(T)$

DTPMU äquidistante Intervallänge der Diskretisierung des Temperaturbereiches der Kurvenfits zu $\mu_i(T)$

TMINMU untere Temperatur des Diskretisierungsbereiches der Kurvenfits $\mu_i(T)$

TMAXMU obere Temperatur des Diskretisierungsbereiches der Kurvenfits $\mu_i(T)$

BDA(l,i,j,k) l–ter Koeffizient des stückweisen polynomialen Kurvenfits der binären Diffusionskoeffizienten $\widetilde{\mathcal{D}}_{ij}(T)$ der Komponenten i und j des binären Gasgemisches im k–ten Temperaturintervall (vgl. Kap. 5.2.3)

$l = 1,5$, $i,j = 1$,**IMAX**, $k = 1$,**JANZBD**

JANZBD Anzahl an Temperaturintervallen der Kurvenfits für $\widetilde{\mathcal{D}}_{ij}(T)$

DTPBD äquidistante Intervallänge der Diskretisierung des Temperaturbereiches der Kurvenfits zu $\tilde{\mathcal{D}}_{ij}(T)$

TMINBD untere Temperatur des Diskretisierungsbereiches der Kurvenfits $\tilde{\mathcal{D}}_{ij}(T)$

TMAXBD obere Temperatur des Diskretisierungsbereiches der Kurvenfits $\tilde{\mathcal{D}}_{ij}(T)$

AKOEFF(l,i,k) l-ter Koeffizient des stückweise polynomialen Kurvenfits der Gleichgewichtskonstanten $\tilde{K}_g^l(T)$ der i-ten Reaktionsklasse im k-ten Temperaturintervall, $l = 1,4$, $i =$ 1,**IMAX**, $k =$ 1,**JANZAK**

JANZAK Anzahl an Temperaturintervallen der Kurvenfits für $\tilde{K}_g^i(T)$

DTPAK äquidistante Intervallänge der Diskretisierung des Temperaturbereiches der Kurvenfits zu $\tilde{K}_g^i(T)$

TMINAK untere Temperatur des Diskretisierungsbereiches der Kurvenfits $\tilde{K}_g^i(T)$

TMAXAK obere Temperatur des Diskretisierungsbereiches der Kurvenfits $\tilde{K}_g^i(T)$

JMAX Maximale Anzahl an Temperaturintervallen aller Kurvenfits

W(i) Molmasse der Komponente i des Gasgemisches, $i =$ 1,**IMAX**

FENTH(i) Formationsenthalpie der Komponente i des Gasgemisches, $i =$ 1,**IMAX**

MUINP Name der Datei, in der die Koeffizienten der Kurvenfits für $\mu_i(T)$ abgespeichert sind

BDINP Name der Datei, in der die Koeffizienten der Kurvenfits für $\tilde{\mathcal{D}}_{ij}(T)$ abgespeichert sind

ENINP Name der Datei, in der die Koeffizienten der Kurvenfits für $\tilde{e}_i(T)$ abgespeichert sind

CVINP Name der Datei, in der die Koeffizienten der Kurvenfits für $\tilde{c}_{V,i}(T)$ abgespeichert sind

AKINP Name der Datei, in der die Koeffizienten der Kurvenfits für $\tilde{K}_g^i(\ln T)$ abgespeichert sind

8 Warnungen

Ziel der Implementierung war unter anderem die Wahrung der Flexibilität bezüglich der zu simulierenden Strömungssituation. Das Programm **NSCNG2** ist so ausgelegt, daß damit unterschiedlichste ebene und rotationssymmetrische Strömungsprobleme wie

- Staupunktprobleme
- Rampenströmungen
- Plattenströmungen
- Kanal– und Düsenströmungen, etc.

untersucht werden können.
Dies wurde durch die Implementierung verschiedener Randbedingungen erreicht. Da bisher aber nur hypersonische Staupunktprobleme konkret untersucht wurden, sind nur einige wenige Kombinationen der zur Zeit sieben Randbedingungen (siehe Kap. 4.5.1) im Rahmen einer Strömungssimulation ausgetestet worden. Dies waren folgende Parametereinstellungen für

NSFLAG	**GFLAG**	**RFLAG**(1)	**RFLAG**(2)	**RFLAG**(3)	**RFLAG**(4)
1	1	2	1	6	3
1	2	2	1	7	3
0	0	2	1	2	4,5
0	2	2	1	7	4,5

Bei allen anderen Parameterkombinationen können eventuell Probleme auftreten. In diesem Fall müssen die entsprechenden Routinen in **BOUND.F** überprüft werden. Sie sind zwar lokal getestet worden, dies garantiert aber keinen fehlerfreien Einsatz bei einer Strömungssimulation.
Damit verbunden sind eventuelle Fehler bei der Lösung der Riemannprobleme am Gitterrand — hier können Probleme in den Routinen in **FLINV.F** auftreten — und der Diskretisierung der reibungsbehafteten Flüsse und des Quellterms der rotationssymmetrischen Gleichungen. Letztere kann Fehler in den Routinen in **FLVIS.F** und **RQUELL.F** sowie **MWINV** in **MWSTEP.F** führen.
Bei einem anderen Strömungsproblem als der hypersonischen Staupunktberechnung an stumpfen Körpern, kann die Verwendung der Routine **PTEST** zu Problemen führen (siehe Kap. 4.5.7.1). In diesem Fall sollte **PFLAG** auf 0 gesetzt werden.

9 Mögliche Änderungen durch den Benutzer

Bei der Implementierung wurde darauf geachtet, daß bestimmte Programmodule in entsprechenden Dateien zusammengefaßt wurden. Dies soll es dem Benutzer erleichtern, ganze Programmodule durch eigene Routinen zu ersetzen. Exemplarisch seien im folgenden einige Änderungsmöglichkeiten angeführt:

1. Änderung der Rekonstruktionsmethode:

   Austausch der Rekonstruktionsmethode in **REKON.F** [1] (siehe Routinen **KOMREK**, **HORNL**, **HORNR**);

   entsprechende Änderung der Variablendeklaration in **REKDEK.H**;

   bei Erhöhung der Rekonstruktionsordnung $N \geq 3$ muß **NMAX** in **FIXDEK.H** entsprechend angepaßt werden;

   Wichtig:

   URXI, **URETA**, **ULXI**, **ULETA** (vgl. Kap. 7.5) müssen nach Verlassen von **REKON** mit den Anfangsdaten für die Riemannprobleme in den Zellrandmittelpunkten belegt sein.

2. Änderung des Riemannlösers:

   Austausch der Routine **ROELSG** in **ROELSG.F** [2] ;

   Beibehaltung der Parameterliste;

   Wichtig:

   Der Vektor $\boldsymbol{F}$ muß mit dem Fluß in Normalenrichtung belegt sein (vgl. Kap. 4.5.4.3)

3. Änderung des Gasmodells:

   Passe **IMAX**, **RMAX**, **SPEZIE** , **REAKT** und **IOFORM** in **FIXDEK.H** an die aktuelle Anzahl an Komponenten des Gasgemisches an;

   tausche die Routine **CHEPRO** [3] in **CHEMIE.F** gegen eine solche aus, die die chemischen Produktionsraten des neuen Gasgemisches beschreibt (vgl. Kap. 4.4);

   berechne in **ROELSG** die Flüsse in Normalenrichtung zu den Partialdichten des neuen Gasgemisches, falls der Riemannlöser von Liu und Vinokur verwendet werden soll (siehe **LVCNG2.F**);

   aktualisiere die Dateien mit den Koeffizienten für die Kurvenfits gemäß 5.2.3;

   passe die Datenstruktur gemäß Kap. 5.2.4 an die veränderte Anzahl an Komponenten an

[1] Der Benutzer hat zur Zeit drei Alternativen zur Auswahl (siehe Kap. 4.5.3).
[2] Der Benutzer hat zur Zeit zwei Alternativen zur Auswahl (siehe Kap. 4.5.4.3).
[3] Der Benutzer hat zur Zeit eine Alternative zur Auswahl (siehe Kap. 6.2.1).

4. Änderung der Koeffizienten der Kurvenfits:

   Falls mehr als 15 Temperaturintervalle für die Kurvenfits gewählt werden, so passe **JMAX** in **CHEDEK.H** entsprechend an;

5. Änderung des Transport– und Diffusionsmodells

   Sollen die Diffusionsflüsse sowie die Transport– und Diffusionskoeffizienten nach anderen Modellen bzw. Mischungsregeln berechnet werden, so müssen die Routinen **MDIFF**, **KOEFF**, **VISCOE** und **DVISCO** in **TRKO.F** entsprechend geändert werden (vgl. Kap. 6.3).

10 Compilieren, Linken und Programmstart

Vor dem Programmstart müssen folgende Dateien compiliert und gelinkt werden:

NSCNG2.F	**MWQUAN.F**	**MWSTEP.F**
OUTPUT.F	**INPUT.F**	**REKON.F**[1]
TEST.F	**INITIAL.F**	**FLINV.F**
CHEMIE.F[2]	**FLVIS.F**	**RQUELL.F**
BOUND.F	**ROELSG.F**[3]	**TRKO.F**[4]
CFD.F		

Dies geschieht automatisch mit dem Aufruf

make -f nscng2.m

des MAKE-Files **nscng2.m**. Daraus resultiert das lauffähige Programm **nscng2.out**. Dieses wird aufgerufen mit

nscng2.out <'input'>.

Dabei steht **input** für den Namen der Datei, in der die Parameter für die Strömungssimulation und die erforderlichen Dateinamen stehen. Die Dateistruktur kann Kap 5.2.1 entnommen werden.
Desweiteren müssen vor dem Programmstart folgende Datein erzeugt werden, damit zur Laufzeit darauf zugegriffen werden kann:

- Referenzgrößen der Anströmung (Kap. 5.2.2)
- Gitter (Kap. 5.2.3)
- Mittelwerte (Kap. 5.2.4)
- Tabellen der Koeffizienten des Energie–, des Transport– und Diffusionsmodells sowie der Gleichgewichtskonstanten (Kap. 5.2.5)

Nach dem Programmablauf stehen die neuen Mittelwerte in der Datei mit dem Namen, der in **DATOUT** abgespeichert ist. Desweiteren wird der Ablauf der Strömungssimulation in einer Datei mit dem Namen

fn.**PRO**

dokumentiert, wobei fn identisch mit dem Namen in **DATOUT** ist.

[1] Der Benutzer hat zur Zeit drei Alternativen zur Auswahl (siehe Kap. 4.5.3).
[2] Der Benutzer hat zur Zeit zwei Alternativen zur Auswahl (siehe Kap. 6.2.1).
[3] Der Benutzer hat zur Zeit zwei Alternativen zur Auswahl (siehe Kap. 4.5.4.3).
[4] Der Benutzer hat zur Zeit drei Alternativen zur Auswahl (siehe Kap. 6.3).

11 Zonales Konzept

Der Einsatzbereich des in den vorangegangenen Kapiteln beschriebenen Programms wird wesentlich durch die Wahl eines strukturierten Gitters eingeschränkt. Dadurch können Geometrien mit lokalen Singularitäten, z. B. bei einer Kante oder einer Rampe, nicht hinreichend aufgelöst werden. Um dennoch nicht auf die Berechnung solcher Geometrien verzichten zu müssen, empfiehlt es sich, das Gitter aus mehreren strukturierten Teilzonen zusammenzusetzen. Die Berechnung der Strömung in den einzelnen Zonen kann dann wie bisher erfolgen. Problematisch sind hier allerdings die Zonenränder, an denen zwei Zonen aufeinander treffen. Man muß sich hier entscheiden, ob sich die Zonen überlappen sollen oder nicht. Es wurde hier das Konzept der Überlappung gewählt, da das Verfahren auf einer Finte–Volumen–Diskretisierung basiert. Dadurch kann die Gitterstruktur in den einzelnen Zonen beibehalten werden, d. h. es liegt jeweils eine Zelle außerhalb des Strömungsgebietes (vgl. Abb. 6). Zur Zeit wird gefordert, daß zwei angrenzende Zonen einen identischen Zonenrand haben (vgl. Abb. 7), d. h. die Ränder der beiden Zonen sind nicht versetzt (vgl. Abb. 8), und die Zonenaußenrandzelle der Zone A muß deckungsgleich mit der Zoneninnenrandzelle der Zone B sein (vgl. Abb. 9). Die Anzahl der Zellen entlang eines Zonenrandes ist also insbesondere auf beiden Seiten gleich. Die Situation aus Abbildung 10 ist demnach ausgeschlossen. Löst man nun am Zonenrand ein Riemannproblem, so ist gewährleistet, daß der Fluß aus der Zone A in die Zone B identisch mit dem Fluß aus Zone B in Zone A ist. Dies ist der kritische Punkt, falls man allgemeinere Situationen zulassen möchte. Zur Zeit läuft auf diesem Gebiet eine Diplomarbeit.

Aufgrund der Einschränkungen sind nur einige wenige Änderungen in dem Programm erforderlich. Sie betreffen im wesentlichen

- die Ein- und Ausgabe der Daten (**ZINPUT**, **ZOUTDAT**, **ZOUTPUT**),
- die Initialiserung (**ZDATINI**, **ZREFINI**, **ZNETINI**, **ZNORMAL**, **ZVOLUME**, **ZARCLEN**, **ZREKINI**) sowie
- die Schnittstellen zwischen den Zonen (**ZBINI**, **ZONEIN**, **ZONEOUT**, **ZRANDUP**, **ZBIN**, **ZBOUT**).

Zum Abspeichern der zonenabhängigen Daten und Parameter werden neue globale Variablen eingeführt. Diese sind in der Include-Datei **ZONDEK.H** zusammengefaßt. Die Variablennamen entsprechen den bisherigen bis auf den Anfangsbuchstaben **Z**. Dies gilt auch für geänderte oder neue Prozedur- und Filenamen. Einzige Ausnahme bildet die Variable **ZSTEPS**. Diese wurde schon in der ursprünglichen Version deklariert.

Die Programmteile zur Berechnung der chemischen Produktionsraten **CHEMIE** und der Strömungssimulation **CFD** werden somit zu Modulen, die für jede Zone identisch ist.

Im folgenden sollen diese beschrieben werden. Dazu wird die Struktur der Kapitel 4 bis 10 beibehalten.

11.1 Programmstruktur

11.1.1 ZNSCNG2 (Hauptprogramm)

Die Programmstruktur hat sich überhaupt nicht verändert (vgl. Kapitel 4.1). Allerdings müssen in einigen Routinen die entsprechenden Funktionen für alle Zonen statt bisher einer einzigen Zone durchgeführt werden:

ZINPUT

Eingabe der strömungs– und diskretisierungsspezifischen Parameter für alle Zonen;

ZINITIAL

Einlesen der Daten und des Gitters für alle Zonen;

Initialisierung der charakteristischen Anströmwerte für die durchzuführende Strömungssimulation für alle Zonen;

Initialisierung der physikalischen Größen;

Initialisierung der Anströmwerte für alle Zonen;

Initialisierung aller gitterspezifischen Größen für alle Zonen;

Initialisierung der rekonstruktionsspezifischen Größen für alle Zonen;

Initialisierung aller Zonenränder;

ZCHEMIE (optional, falls chemisches Nichtgleichgewicht mit Operatorsplitting modelliert werden soll)

Sequentielle Berechnung der Diskretisierung von (29) bzw. (31) für alle Zonen;

ZCFD

Sequentielle Berechnung der Strömung in allen Zonen;

Sequentielle Berechnung der Bilanzgrößen (2) auf dem neuen Zeitlevel in allen Zonen;

ZONEIN(l)

Umspeichern der zonenabhängigen Parameter und Daten der Zone l in die entsprechenden zonenunabhängigen Variablen;

BOUND

Berechnung der Bilanzgrößen (2) in den Randzellen des Gitters und der Zonenränder der Zone l;

ZONEOUT(l)

Umspeichern der Zonenrandwerte und der neuen Bilanzgrößen in die entsprechenden zonenabhängigen Variablen der Zone l;

ZOUTPUT

Ausgabe der neuen Mittelwerte nach **NSTEPS** durchgeführten Zeitschritten.

11.1.2 ZINPUT

Einige zur Laufzeit benötigten Parameter sind zonenabhängig und müssen zu Programmbeginn in der Routine **ZINPUT** zusätzlich zu Kapitel 4.2 eingelesen werden. Diese sind **ZONANZ**, **ZRAND**, **ZGFLAG**, **RZANZ**, **ZRFLAG** und **ZKFLAG**. Die Struktur der Eingabedatei muß wie folgt verändert werden:

ZONANZ Anzahl der Zonen (zur Zeit nur 2 oder 4)

OFLAG ohne/mit Operatorsplitting (siehe Kap. 5)

WFLAG adiabate/nicht adiabate Wand (siehe Kap. 5)

KWFLAG voll katalytische/nicht katalytische Wand (siehe Kap. 5)

ZGFLAG(l) ebene/rotationssymmetrische/achsensymmetrische Strömung (siehe Kap. 11.3) in den Zonen $l = 1$,**ZONANZ**

NSFLAG ohne/mit Reibung, Diffusion und Wärmeleitung (siehe Kap. 5)

CFLAG inertes Gas/chemisches Nichtgleichgewicht (siehe Kap. 5)

RZANZ Anzahl der aneinandergrenzenden Zonenränder

ZRFLAG(i, l) Randtyp des Randes Γ_i, $i = 1, 4$ jeder einzelnen Zone $l = 1$,**ZONANZ** (siehe Kap. 5, Kap. 11.3)

ZRAND(l, i) Zuordnung der Zonenränder $l = 1$,**RZANZ** zu den lokalen Rand– und Zonennummern der angrenzenden Zonen

ZKFLAG(i, l) fester Körperrand oder freier Strömungsrand des Randes Γ_i, $i = 1, 4$ jeder einzelnen Zone $l = 1$,**ZONANZ** (siehe Kap. 5, Kap. 11.3)

PFLAG ohne/mit Überprüfung auf positiven Druck (siehe Kap. 5)

ENINP Dateiname für Tabellierung von $\tilde{e}_i(T) := e_i(T)M_i/R/T$

CVINP Dateiname für Tabellierung von $\tilde{c}_{V,i}(T) := c_{V,i}(T)M_i/R$

MUINP Dateiname für Tabellierung von $\mu_i(T)$

BDINP Dateiname für Tabellierung von $\widetilde{\mathcal{D}}_{ij}(T) := p\mathcal{D}_{ij}$

AKINP Dateiname für Tabellierung von $\tilde{K}_g^i(\ln T) := \ln\left(K_g^i(\ln T)\right)$

POLGRA Rekonstruktionsordnung

NSTEPS Anzahl der zu berechnenden Zeitschritte

DELTAT Zeitschrittweite

DATZWI Dateiname für Zwischenspeicherung der Daten

ZSTEPS Anzahl der Zeitschritte, nach denen die Daten zwischengespeichert werden sollen

SSTEPS Anzahl der Zeitschritte, nach denen überprüft werden soll, ob die Strömung stationär ist

GITTER Dateiname für Gitter

DATINP Dateiname für Eingabedaten

DATOUT Dateiname für Ausgabedaten

REFQUA Dateiname für Anströmungsgrößen

11.1.3 ZINITIAL

Die Initialisierung der zur Laufzeit benötigten Parameter und Daten erfolgt in der Routine **ZINITIAL**. Bis auf die physikalischen Größen (vgl. Routine **PHYINI** in Kap. 4.3) sind diese zonenabhängig:

ZDATINI

Einlesen des Gitters und der Strömungsdaten (Bilanzgrößen), d.h. Vorbelegung der Variablen (siehe Kap. 11.3.2 und 11.4.1) für jede Zone:

ZMZAHLX(l)	$l = 1$,**ZONANZ**	
ZMZAHLY(l)	$l = 1$,**ZONANZ**	
ZGRID(j,k,i,l)	$j = -2, \mathbf{ZMZAHLX}(l)+1$,	$k = -2, \mathbf{ZMZAHLY}(l)+1$,
	$i = 1, 2$,	$l = 1$,**ZONANZ**
ZMW(j,k,i,l)	$j = -1, \mathbf{ZMZAHLX}(l)+1$,	$k = -1, \mathbf{MZAHLY}(l)+1$,
	$i = 1$,**DIMMAX**,	$l = 1$,**ZONANZ**
ZMWRHO(j,k,i,l)	$j = -1, \mathbf{ZMZAHLX}(l)+1$,	$k = -1, \mathbf{ZMZAHLY}(l)+1$,
	$i = 1$,**IMAX**,	$l = 1$,**ZONANZ**

PHYINI

unverändert (vgl. Kap. 4.3)

ZREFINI

Einlesen der Anströmungsgrößen (siehe Kap. 7.1 und 11.4.1):

XREF(i) $i = 1$,**IMAX**

RHOREF, **MREF**, **NREF**, **PREF**, **FREF**

ZTWALL(i,l) (falls **NSFLAG** $= 1$, **WFLAG** $= 1$ und **ZKFLAG**$(i,l) = 1$), $i = 1, 4$, $l = 1$,**ZONANZ**

XWALL(i) (falls **NSFLAG**$= 1$, **KWFLAG**$= 0$), $i = 1$,**IMAX**

ZNETINI

Initialisierung der gitterabhängigen Größen (siehe Kap. 7.3 und 11.4.1):

1. Normale in den Zellrandmittelpunkten (**ZNORMAL**) aller Zonen
   ZNXI(j,k,i,l) $j = -1$,**ZMZAHLX**(l), $i = 1,2$,
   $k = -1$,**ZMZAHLY**$(l)+1$, $l = 1$,**ZONANZ**
   ZNETA(j,k,i,l) $j = -1$,**ZMZAHLX**$(l)+1$, $i = 1,2$,
   $k = -1$,**ZMZAHLY**(l), $l = 1$,**ZONANZ**
2. Flächeninhalt der einzelnen Zellen (**ZVOLUME**) aller Zonen
   ZVOL(j,k,l) $j = -1$,**ZMZAHLX**$(l)+1$,
   $k = -1$, **ZMZAHLY**$(l)+1$, $l = 1$,**ZONANZ**
3. Bogenlänge der Zellkanten (**ZARCLEN**) aller Zonen
   ZARCXI(j,k,l) $j = -1$,**ZMZAHLX**$(l)+1$,
   $k = 0$,**ZMZAHLY**(l), $l = 1$,**ZONANZ**
   ZARCETA(j,k,l) $j = 0$,**ZMZAHLX**(l),
   $k = -1$,**ZMZAHLY**$(l)+1$, $l = 1$,**ZONANZ**

ZREKINI

Initialisierung der für die Rekonstruktion benötigten Größen, hier: Berechnung der Stützstellen auf den eindimensionalen parametrisierten Kurven entlang denen rekonstruiert wird (siehe Kap. 7.4, 11.1.7 und 11.4.1) für alle Zonen:

ZMPKXI(j,k,l) $j = -$**NMAX**,**MZAHLX** + **NMAX**,
$k = 0$,**MZAHLY**, $l = 1$,**ZONANZ**
ZMPKETA(j,k,l) $j = 0$,**MZAHLX**,
$k = -$**NMAX**,**MZAHLY** + **NMAX**, $l = 1$,**ZONANZ**

11.1.4 ZCHEMIE

Die Berechnung der chemischen Produktionsraten in einer Zone erfolgt wie bisher in der Routine **CHEMIE** (vgl. Kap. 4.4). Die Routine **ZCHEMIE** steuert die sequentielle Berechnung in allen Zonen $l = 1$,**ZONANZ**. Dazu müssen vor dem Aufruf von **CHEMIE** die zonenabhängigen Daten umgespeichert werden in zonenunabhängige Variablen (vgl. Routine **ZONEIN**) und nach jeder Zone die neu berechneten Daten wieder in die zonenabhängigen Variablen geschrieben werden (vgl. Routine **ZONEOUT**). Die Zonenränder müssen, nachdem die Produktionsterme in zwei aneinandergrenzenden Zonen berechnet wurden, ebenfalls auf den aktuellen Stand gebracht werden. Dies geschieht in der Routine **ZRANDUP** (vgl. Kap. 11.2).

ZONEIN(l)

Umspeichern der zonenabhängigen Parameter und Daten der Zone l in die entsprechenden zonenunabhängigen Variablen;

CHEMIE

Berechnung der chemischen Produktionsraten s_i in der Zone l;

Aufdatieren der Partialdichten zu jeder Zelle **im** Strömungsgebiet mittels eines expliziten Eulerschrittes (vgl. Kap. 4.4), falls der Quellterm der chemischen Produktionsraten von den übrigen Gleichungen gesplittet werden soll (**OFLAG** $= 1$). Diese Routine wird sequentiell für alle Zonen durchgeführt;

ZBOUT

Aufdatieren der Zonenränder (**MWI**, **MWRHOI**) der gerade bearbeiteten Zone l vor dem Verlassen des zonenunabhängigen Bereiches;

ZONEOUT(l)

Umspeichern der Zonenrandwerte und der neuen Bilanzgrößen in die entsprechenden zonenabhängigen Variablen der Zone l;

ZRANDUP(l)

Aktualisierung der Zonenränder nachdem die Produktionsterme in zwei aneinandergrenzenden Zonen berechnet wurden, d.h. die am Zonenrand anliegenden Werte einer Zone (**ZMWI**, **ZMWRHOI**) müssen als Randwerte (**ZMWA**, **ZMWRHOA**) in der angrenzenden Zone gesetzt werden.

11.1.5 ZCFD

Die Berechnung der Strömung in einer Zone erfolgt wie bisher in der Routine **CFD** (vgl. Kap. 4.5). Die Routine **ZCFD** steuert die sequentielle Berechnung in allen Zonen $l = 1$,**ZONANZ**. Dazu müssen vor dem Aufruf von **CFD** die zonenabhängigen Daten umgespeichert werden in zonenunabhängige Variablen (vgl. Routine **ZONEIN**) und nach jeder Zone die neu berechneten Daten wieder in die zonenabhängigen Variablen geschrieben werden (vgl. Routine **ZONEOUT**). Wird kein Operatorsplitting durchgeführt, so muß vor der Berechnung der Strömung die Routine **CHEMIE** aufgerufen werden, um den Quellterm **QUELLE** zu berechnen. Die Zonenränder müssen, nachdem die Produktionsterme in zwei aneinandergrenzenden Zonen berechnet wurden, ebenfalls auf den aktuellen Stand gebracht werden. Dies geschieht in der Routine **ZRANDUP** (vgl. 11.2).

ZONEIN(l)

Umspeichern der zonenabhängigen Parameter und Daten der Zone l in die entsprechenden zonenunabhängigen Variablen;

CHEMIE (optional, falls chemisches Nichtgleichgewicht modelliert werden soll ohne Operatorsplitting)

Berechnung der chemischen Produktionsraten s_i in der Zone l;

Aufdatieren der Partialdichten zu jeder Zelle **im** Strömungsgebiet mittels eines expliziten Eulerschrittes (vgl. Kap. 4.4), falls der Quellterm der chemischen Produktionsraten von den übrigen Gleichungen gesplittet werden soll (**OFLAG** = 1). Diese Routine wird sequentiell für alle Zonen durchgeführt;

CFD

Berechnung der Bilanzgrößen (2) auf dem neuen Zeitlevel in der Zone l;

ZBOUT

Aufdatieren der Zonenränder (**MWI**, **MWRHOI**) der gerade bearbeiteten Zone l vor dem Verlassen des zonenunabhängigen Bereiches;

ZONEOUT(l)

Umspeichern der Zonenrandwerte und der neuen Bilanzgrößen in die entsprechenden zonenabhängigen Variablen der Zone l;

ZRANDUP(l)

Aktualisierung der Zonenränder nachdem die Strömung in zwei aneinandergrenzenden Zonen berechnet wurden, d.h. die am Zonenrand anliegenden Werte einer Zone (**ZMWI**, **ZMWRHOI**) müssen als Randwerte (**ZMWA**, **ZMWRHOA**) in der angrenzenden Zone gesetzt werden.

11.1.6 ZBOUND

Neben den schon in Kapitel 4.5.1 erklärten Randtypen wird ein weiterer benötigt. Dieser besetzt den Randwert an einem Zonenrand mit den inneren Werten **MWI**, **MWRHOI** der angrenzenden Zone. Die entsprechende Routine heißt **ZBIN** und hat die selben Parameter wie die übrigen Randroutinen:

ZBIN (Zonenrand)

hier werden die Innenwerte **MWI**, **MWRHOI** der angrenzenden Zone als Randwerte verwendet.

11.1.7 ZREKON

Bezüglich der Rekonstruktion hat sich nur eine Änderung ergeben. Diese betrifft die Initialisierung der Stützstellen auf den eindimensionalen Kurven entlang denen rekonstruiert wird (vgl. Kap. 11.1.3). Die unterschiedlichen Files mit den verschiedenen Rekonstruktionsordnungen (vgl. Kap. 4.5.3) stehen nun in

ZREKON1.F Rekonstruktion von Ordnung $N = 1$

ZREKON2.F Rekonstruktion von Ordnung $N = 2$

ZREKONN.F Rekonstruktion von beliebiger Ordnung $N \geq 1$

11.1.8 ZOUTPUT

Nach Beendigung der Zeitintegration werden die berechneten Daten ausgegeben und ein Protokoll mit den Parametern der Strömungssimulation erstellt. Der Filename des Protokolls hat die Endung **.PRO**. Es wird in der Routine **ZOUTPUT** erstellt. Von dort wird auch die Prozedur **ZOUTDAT** aufgerufen. In ihr werden die berechneten Mittelwerte gemäß Kap. 11.3.3 in das File, dessen Name in **DATOUT** abgespeichert wurde, rausgeschrieben. Diese Routinen sind im wesentlichen identisch zu **OUTPUT** und **OUTDAT** (vgl. Kap. 4.6). In der Routine **ZOUTPUT** werden zusätzlich noch die zonenabhängigen Parameter protokolliert, und in der Routine **ZOUTDAT** wird eine weitere Datei mit Endung **.plt** erstellt, die zur Zeit für die Verarbeitung der Daten mit der schon existierenden graphischen Software benötigt wird.

11.2 Kommunikation zwischen den Zonen

Die Berechnung der chemischen Produktionsraten und der Strömung erfolgt für jede Zone sequentiell in den Modulen **CHEMIE** und **CFD**. Dazu müssen vor und nach dem Aufruf dieser Routinen die benötigten zonenabhängigen Daten und Parameter umgespeichert werden in die entsprechenden zonenunabhängigen Variablen. Da die Strömung in den einzelnen Zonen nicht unabhängig von den angrenzenden Zonen ist, muß gewährleistet werden, daß ein Austausch zwischen zwei benachbarten Zonen über den Zonenrand stattfindet. Dies wird durch die vier folgenden Routinen gewährleistet:

ZONEIN(l)

Vor der Berechnung der chemischen Produktionsraten oder der Strömung innerhalb der Zone l werden hier die folgenden zonenabhängigen Parameter und Daten

Zonengröße	**ZMZAHLX**(l)	**ZMZAHLY**(l)
Zonale Parameter	**ZGFLAG**(l)	**ZRFLAG**(i,l)
	ZKFLAG(i,l)	**ZTWALL**(i,l)
zonales Gitter	**ZGRID**(j,k,i,l)	**ZNXI**(j,k,i,l)
	ZNETA(j,k,i,l)	**ZARCXI**(j,k,l)
	ZARCETA(j,k,l)	**ZVOL**(j,k,l)
zonale Rekonstruktion	**ZMPKXI**(j,k,l)	**ZMPKETA**(j,k,l)
zonale Bilanzgrößen	**ZMW**(j,k,i,l)	**ZMWRHO**(j,k,i,l)
Zonenrandwerte	**ZMWI**(j,k,i,l)	**ZMWRHOI**(j,k,i,l)
	ZMWA(j,k,i,l)	**ZMWRHOA**(j,k,i,l)

in die entsprechenden zonenunabhängigen Variablen umgespeichert. Die Namen dieser Größen sind bis auf den Anfangsbuchstaben **Z** und den Index l identisch.

ZBOUT

Vor dem Verlassen des zonenunabhängigen Bereiches der Module **CHEMIE** und **CFD** müssen die Zonenrandwerte (**MWI**, **MWRHOI**) der gerade bearbeiteten Zone l aufdatiert werden, bevor diese an die übergeordnete zonenabhängige Variable zurück übergeben werden und die nächste Zone bearbietet wird.

ZONEOUT(l)

Nachdem die Bilanzgrößen in einer der beiden Modulen **CHEMIE** und **CFD** für die Zone l berechnet wurden, müssen diese an die übergeordneten zonenabhängigen Variablen

zonale Bilanzgrößen	**ZMW**(j,k,i,l)	**ZMWRHO**(j,k,i,l)
Zonenrandwerte	**ZMWI**(j,k,i,l)	**ZMWRHOI**(j,k,i,l)
	ZMWA(j,k,i,l)	**ZMWRHOA**(j,k,i,l)

zurück übergeben werden, bevor die nächste Zone bearbeitet wird.

ZRANDUP(l)

Nachdem die Produktionsterme oder die Strömung in zwei angrenzenden Zonen berechnet wurden, müssen die Zonenränder ebenfalls aktualisiert werden, d. h. die

am Zonenrand anliegenden Werte einer Zone (**ZMWI**, **ZMWRHOI**) müssen als Randwerte (**ZMWA**, **ZMWRHOA**) in der angrenzenden Zone gesetzt werden. Dazu wird die Variable **ZRAND** benötigt. Sie ordnet einem Zonenrand die Nummern der angrenzenden Zonen und der lokalen Randnummer der jeweiligen Zone zu (vgl. Kap. 11.3.1).

11.3 Programmsteuerung über Optionen und Dateien

11.3.1 Optionen

Zur Steuerung des zonalen Konzepts werden einige Parameter benötigt, die zonenabhängig sind. Sie haben die selben Funktionen wie die in Kapitel 5.1 beschriebenen:

ZGFLAG(l)

ebene Strömung in der Zone l	(0)
rotationssymmetrische Strömung in der Zone l	(1)
achsensymmetrische, ebene Strömung in der Zone l	(2)

ZRFLAG(i, l)

Zuordnung eines der folgenden Randtypen zu den vier Strömungsrändern Γ_i, $i = 1, 4$, der Zone l:

Zonenrand	(0)
hypersonischer Inflow	(1)
hypersonischer Outflow	(2)
Wand (Haftbedingung, **NSFLAG** $= 1$)	(3)
Wand (Verdünnungswelle, **NSFLAG** $= 0$)	(4)
Wand (Reflexion, **NSFLAG** $= 0$)	(5)
Symmetrieachse (Rotationssymmetrie, nur $x_2 = 0$)	(6)
Symmetrieachse (Achsensymmetrie, nur $x_2 = 0$)	(7)

ZRAND(l, i) Zuordnung des Zonenrandes $l = 1$,**RZANZ** zu den lokalen Randnummern und Zonennummern der angrenzenden Zonen:

ZRAND$(l, 1)$	Nummer der 1. angrenzenden Zone
ZRAND$(l, 2)$	Randnummer des Zonenrandes Γ_i, $i = 1, 4$, der 1. angrenzenden Zone
ZRAND$(l, 3)$	Nummer der 2. angrenzenden Zone
ZRAND$(l, 4)$	Randnummer des Zonenrandes Γ_i, $i = 1, 4$, der 2. angrenzenden Zone

ZKFLAG(i, l)

Charakterisierung des Randes Γ_i, $i = 1, 4$, der Zone l:

freier Strömungsrand	(0)
feste Körperwand	(1)

11.3.2 Dateistrukturen

Durch das zonale Konzept ergeben sich einige Veränderungen in den Dateistrukturen (vgl. Kap. 5.2).

11.3.2.1 Eingabedatei

Die Datei, die beim Programmstart als Parameter mit angegeben werden muß (vgl. Kap. 11.7), hat folgende Struktur:

ZONANZ : Anzahl verschiedener Zonen

OFLAG : Parameter zur Steuerung des Operatorsplittings

WFLAG : Parameter zur Steuerung der thermischen Wandmodellierung

KWFLAG : Parameter zur Steuerung der katalytischen Wandmodellierung

ZGFLAG(l) : Parameter zur Steuerung der Geometrie der Zone l

NSFLAG : Parameter zur Steuerung der Differentialgleichung

CFLAG : Parameter zur Steuerung der Modellierung der chemischen Produktion

ZRFLAG(i,l): Randtypen zu Rand $\Gamma_i, i = 1,4$, in der Zone l:

Zonenrand	(0)
hypersonischer Inflow	(1)
hypersonischer Outflow	(2)
Wand (Haftbedingung, Navier–Stokes)	(3)
Wand (Verdünnungswelle, Euler)	(4)
Wand (Reflexion, Euler)	(5)
Symmetrieachse (Rotationssym., nur $x_2 = 0$)	(6)
Symmetrieachse (Achsensym., nur $x_2 = 0$)	(7)

ZRAND(l,i) : Zuordnung der Randnummern und Zonennummern zweier benachbarter Zonen zu dem Zonenrand $l = 1$,**RZANZ**

ZRAND$(l,1)$	Nummer der 1. angrenzenden Zone
ZRAND$(l,2)$	Randnummer des Zonenrandes Γ_i, $i = 1,4$, der 1. angrenzenden Zone
ZRAND$(l,3)$	Nummer der 2. angrenzenden Zone
ZRAND$(l,4)$	Randnummer des Zonenrandes Γ_i, $i = 1,4$, der 2. angrenzenden Zone

ZKFLAG(i,l): Charakterisierung des Gitterrandes $\Gamma_i, i = 1,4$, der Zone l

PFLAG : Parameter zur Steuerung der Drucküberprüfung

ENINP : Dateiname für die Tabellierung der inneren Energie $\tilde{e}_i(T)$

CVINP : Dateiname für die Tabellierung der Wärmekapazität $\tilde{c}_{V,i}(T)$

MUINP : Dateiname für die Tabellierung von $\mu_i(T)$

BDINP : Dateiname für die Tabellierung von $\widetilde{\mathcal{D}}_{ij}(T)$

AKINP : Dateiname für die Tabellierung von $\tilde{K}_g^i(\ln T)$

POLGRA : Interpolationsordnung

NSTEPS : Anzahl durchzuführender Zeitschritte

DELTAT : Zeitschrittweite

DATZWI : Dateiname für Zwischenspeicher

ZSTEPS : Anzahl der Zeitschritte, nach denen die Daten zwischengespeichert werden sollen

SSTEPS : Anzahl der Zeitschritte, nach denen überprüft werden soll, ob die Strömung stationär ist

GITTER : Dateiname für Gitter

DATINP : Dateiname für Bilanzgrößen

DATOUT : Dateiname für Ausgabedaten

REFQUA : Dateiname für Referenzgrößen

11.3.2.2 Referenzgrößen der Anströmung

Als zusätzliche zonenabhängige Referenzgrößen wird der Parameter **ZTWALL** für die Wandtemperaturen in den einzelnen Zonen $l \in \{1, \ldots, \mathbf{ZONANZ}\}$ benötigt. Die Datei der Referenzgrößen hat somit folgende leicht veränderte Struktur (vgl. Kap. 5.2.2):

XREF(i) Massenbruch Y_i der Anströmung, $i = 1, \mathbf{IMAX}$

RHOREF Massendichte der Anströmung

MREF x_1–Impuls der Anströmung

NREF x_2–Impuls der Anströmung

REREF Totalenergie der Anströmung

PREF Druck der Anströmung

TREF Temperatur der Anströmung

ZTWALL$(1, l)$ (falls **NSFLAG** $= 1$, **WFLAG** $= 1$, **ZKFLAG**$(1, l) = 1$) Wandtemperatur

ZTWALL$(2, l)$ (falls **NSFLAG** $= 1$, **WFLAG** $= 1$, **ZKFLAG**$(2, l) = 1$) Wandtemperatur

ZTWALL$(3,l)$ (falls **NSFLAG** $= 1$, **WFLAG** $= 1$, **ZKFLAG**$(3,l) = 1$) Wandtemperatur

ZTWALL$(4,l)$ (falls **NSFLAG** $= 1$, **WFLAG** $= 1$, **ZKFLAG**$(4,l) = 1$) Wandtemperatur

XWALL(i) (falls **NSFLAG** $= 1$, **KWFLAG** $= 0$) Massenbruch an der Wand, $i =$ $1,$**IMAX**

11.3.2.3 Zonales Gitter

Das Gitter setzt sich nun aus mehreren benachbarten Zonen zusammen. Dadurch hat sich sich die Struktur des gesamten Gitters gegenüber Kapitel 5.2.3 wie folgt verändert:

Struktur: **ZMZAHLX**(l) **ZMZAHLY**(l), $l =$ 1,**ZONANZ**

$$\boldsymbol{x}(j-1/2, k-1/2, l),\ j = -2, \mathbf{MZAHLX}+1,\ k = -2, \mathbf{MZAHLY}+1$$

Sollte einer der Zonenränder eine Symmetrieachse sein, so gilt für diese dasselbe wie in Kapitel 5.2.3 beschrieben.

11.3.3 Mittelwerte der Bilanzgrößen

Die zu dem zonalem Gitter gehörenden Mittelwerte sind wie folgt abgespeichert (vgl. auch Kap. 5.2.4):

Struktur:	**ASTEPS**	# alter Zeitschritt
	Zone l	# Daten der Zone $l =$ 1,**ZONANZ**
	(**ZMW**(j,k,i,l), $i =$ 2,**DIMMAX**),	# $\rho\boldsymbol{v}, \rho E, \rho_1, \ldots, \rho_{\mathbf{IMAX}}$
	(**ZMWRHO**(j,k,i,l), $i =$ 1,**IMAX**)	# $k = -1$,**MZAHLY**+1,
		$j = -1$,**MZAHLX**+1

11.3.4 Protokoll

Im abschließenden Protokoll (vgl. Kap. 5.2.6) werden nun auch einige zonenabhängige Daten protokolliert:

Struktur:

NSFLAG
CFLAG
OFLAG
XREF(i), $i =$ 1,**IMAX**
RHOREF

MREF	
NREF	
REREF	
PREF	
TREF	
WFLAG	(falls **NSFLAG** $= 1$)
ZTWALL(i,l)	(falls **NSFLAG** $= 1$, **WFLAG** $= 1$, **ZKFLAG**$(i,l) = 1$, $i = 1,4,\ l = 1,$**ZONANZ**)
XWALL(i)	(falls **NSFLAG** $= 1$, **KWFLAG** $= 0$) $i = 1,$**IMAX**
MUEREF	(falls **NSFLAG** $= 1$)
MREF/MUEREF	(falls **NSFLAG** $= 1$) Reynoldszahl pro Länge
POLGRA	
ZGFLAG(l)	$l = 1,$**ZONANZ**
DELTAT	
ZMZAHLX(l)	$l = 1,$**ZONANZ**
ZMZAHLY(l)	
GITTER	
RZANZ	
ZRAND(l,i)	$l = 1,$**RZANZ**, $i = 1,4$

11.4 Globale Variablen

Die globalen zonenabhängigen Variablen sind in der INCLUDE–Datei **ZONDEK.H** zusammengefaßt. Es handelt sich dabei um Variablen, die in den schon existierenden INCLUDE–Dateien als zonenunabhängige Größen deklariert sind (vgl. Kap. 7.1, 7.2, 7.3, 7.4):

11.4.1 INCLUDE–Datei ZONDEK

Die zonenabhängigen Variablen und Parameter sind:

ZONMAX	maximale Anzahl der verschiedenen Zonen
ZONANZ	Anzahl der verschiedenen Zonen
RZMAX	maximale Anzahl an verschiedenen Zonenrändern
RZANZ	Anzahl der verschiedenen Zonenrändern
ZTWALL(i,l)	Körpertemperatur an der nichtadiabaten Wand Γ_i, $i = 1,4$, der Zone $l = 1,$**ZONANZ**

ZGFLAG(l) Wahl der Geometrie in der Zone $l = 1$,**ZONANZ**

eben	(0)
quasi–2D, rotationssymmetrisch	(1)
eben, achsensymmetrisch	(2)

ZRFLAG(i, l) Zuordnung eines Randtyps zu dem Rand Γ_i, i, $i = 1, 4$, der Zone $l = 1$,**ZONANZ**

Zonenrand	(0)
Hypersonischer Inflow	(1)
Hypersonischer Outflow	(2)
Körperwand (Haftbedingung, Navier–Stokes)	(3)
Körperwand (Verdünnungswelle, Euler)	(4)
Körperwand (Reflexion, Euler)	(5)
Symmetrieachse (Rotationssymmetrie, nur Γ_3 und $x_2 = 0$)	(6)
Symmetrieachse (Achsensymmetrie, nur Γ_3 und $x_2 = 0$)	(7)

ZRAND(l, i) Zuordnung der Randnummern und Zonennummern zweier benachbarter Zonen zu dem Zonenrand $l = 1$,**RZANZ**

ZRAND$(l, 1)$	Nummer der 1. angrenzenden Zone
ZRAND$(l, 2)$	Randnummer des Zonenrandes Γ_i, $i = 1, 4$, der 1. angrenzenden Zone
ZRAND$(l, 3)$	Nummer der 2. angrenzenden Zone
ZRAND$(l, 4)$	Randnummer des Zonenrandes Γ_i, $i = 1, 4$, der 2. angrenzenden Zone

ZKFLAG(i, l) Charakterisierung des Randes Γ_i, $i = 1, 4$, der Zone $l = 1$,**ZONANZ**

freier Strömungsrand	(0)
feste Körperwand	(1)

ZMZAHLX(l) Anzahl der Zellen entlang der Parameterkurve in ξ–Richtung -3 der Zone $l = 1$,**ZONANZ**

ZMZAHLY(l) Anzahl der Zellen entlang der Parameterkurve in η–Richtung -3 der Zone $l = 1$,**ZONANZ**

ZMW(j, k, i, l) Mittelwerte der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ in der Zelle $V_{j,k}$ der Zone l

$j = -1,$**MZAHLX** $+ 1$, $k = -1,$**MZAHLY** $+ 1$, $i = 1, 4$, $l =$ 1,**ZONANZ**

ZMWI(j, i, k, l) Mittelwerte der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ am Rand k innerhalb des Rechengebietes der Zone l

$j = 0,$**MMAX**, $i = 1,$**DIMMAX**, $k = 1, 4$, $l = 1$,**ZONANZ**

ZMWA(j,i,k,l) Mittelwerte der Erhaltungsgrößen $\rho, \rho\boldsymbol{v}, \rho E$ am Rand k außerhalb des Rechengebietes der Zone l

$j = 0,$**MMAX**, $i = 1,$**DIMMAX**, $k = 1,4$, $l = 1,$**ZONANZ**

ZMWRHO(j,k,i,l) Mittelwerte der Partialdichten ρ_i in der Zelle $V_{j,k}$ der Zone l

$j = -1,$**MZAHLX** $+ 1$, $k = -1,$**MZAHLY** $+ 1$, $i = 1,$**IMAX**, $l = 1,$**ZONANZ**

ZMWRHOI(j,i,k,l) Mittelwerte der Partialdichten ρ_i am Rand k innerhalb des Rechengebietes der Zone l

$j = 0,$**MMAX**, $i = 1,$**IMAX**, $k = 1,4$, $l = 1,$**ZONANZ**

ZMWRHOA(j,i,k,l) Mittelwerte der Partialdichten ρ_i am Rand k außerhalb des Rechengebietes der Zone l

$j = 0,$**MMAX**, $i = 1,$**IMAX**, $k = 1,4$, $l = 1,$**ZONANZ**

ZGRID(j,k,i,l) Koordinate i eines Zelleckpunktes der Zone l

$j = -2,$**MZAHLX** $+ 1$, $k = -2,$**MZAHLX** $+ 1$, $i = 1,2$, $l = 1,$**ZONANZ**

ZNXI(j,k,i,l) Koordinate i der äußeren Normalen im Zellrandmittelpunkt $\boldsymbol{P}_4^{j,k}$ zur Zelle $V_{j,k}$ einer Parameterkurve in ξ–Richtung der Zone l

$j = -1,$**MZAHLX**, $k = 0,$**MZAHLY**, $i = 1,2$, $l = 1,$**ZONANZ**

ZNETA(j,k,i,l) Koordinate i der äußeren Normalen im Zellrandmittelpunkt $\boldsymbol{P}_1^{j,k}$ zur Zelle $V_{j,k}$ einer Parameterkurve in η–Richtung der Zone l

$j = 0,$**MZAHLX**, $k = -1,$**MZAHLY**, $i = 1,2$, $l = 1,$**ZONANZ**

ZARCXI(j,k,l) Länge des Zellrandes $\Gamma_4^{j,k}$ der Zone l

$j = -1,$**MZAHLX**, $k = 0,$**MZAHLY**, $l = 1,$**ZONANZ**

ZARCETA(j,k,l) Länge des Zellrandes $\Gamma_1^{j,k}$ der Zone l

$j = 0,$**MZAHLX**, $k = -1,$**MZAHLY**, $l = 1,$**ZONANZ**

ZVOL(j,k,l) Volumen der Zelle $V_{j,k}$ der Zone l

$j = -1,$**MZAHLX** $+ 1$, $k = -1,$**MZAHLY** $+ 1$, $l = 1,$**ZONANZ**

ZMPKXI(j,k,l) Parametrisierung einer Parameterkurve in ξ–Richtung, d.h. $\eta = k =$ const. über den euklidischen Abstand der Zellrandmittelpunkte $\boldsymbol{P}_2^{j,k}$ und $\boldsymbol{P}_4^{j,k}$ der Zone l

$j =$ **−POLGRA**,**MZAHLX** $+$ **POLGRA**,

$k =$ **−POLGRA**,**MZAHLY** $+$ **POLGRA**,

$l = 1,$**ZONANZ**

ZMPKETA(j,k,l) Parametrisierung einer Parameterkurve in η–Richtung, d.h. $\xi = j = \text{const.}$ über den euklidischen Abstand der Zellrandmittelpunkte $\boldsymbol{P}_3^{j,k}$ und $\boldsymbol{P}_1^{j,k}$ der Zone l

$j = -$**POLGRA**,**MZAHLX** + **POLGRA**,

$k = -$**POLGRA**,**MZAHLY** + **POLGRA**,

$l = 1$,**ZONANZ**

11.5 Warnungen

Das zonale Konzept befindet sich zur Zeit noch in der Testphase. Erste Rechnungen zeigen gute Ergebnisse. Es kann allerdings zu Problemen kommen, wenn eine Stoßwelle gegen einen Zonenrand läuft und über diesen hinwegtransportiert werden muß. Bei Rekonstruktionsordnung größer als 1 kann es passieren, daß der Stoß vor dem Zonenrand stehen bleibt. Der Grund dafür liegt wohl darin, daß zwischen zwei benachbarten Zonen nur die unmittelbaren Nachbarzellen ausgetauscht werden. Bei der Rekonstruktion höherer Ordnung sind allerdings entsprechend mehr Informationen zwischen den Zonen auszutauschen, jeweils **POLGRA** Zellen auf beiden Seiten des Zonenrandes. Es empfiehlt sich deshalb zunächst mit einer Rechnung erster Ordnung zu beginnen, bis die Strömung fast stationär ist und dann zu höherer Ordnung und eventuell aufwendigeren physikalischen Modellen zu wechseln.

11.6 Mögliche Änderungen durch den Benutzer

Sollen andere Konstellationen als zwei oder vier Zonen betrachtet werden, so beachte man folgende möglicherweise erforderlichen Änderungen:

1. Änderung des zonalen Konzepts:

   Falls mehr als vier Zonen zugelassen werden sollen, so erhöhe **ZONMAX** entsprechend;

   falls mehr als vier Zonenränder zugelassen werden sollen, so erhöhe **RZMAX** entsprechend;

   ändere in der Routinie **ZOUTPUT** die Ausgabe des **plt**–Files;

   passe in der INPUT–Datei die Parameter **ZONANZ**, **RZANZ**, **ZRFLAG**, **ZKFLAG**, **ZGFLAG** und **ZRAND** entsprechend an;

   passe in der REFERENZ–Datei die Wandtemperaturen **ZTWALL** entsprechend an;

2. Änderung der Rekonstruktionsordnung:

   Der Benutzer hat zur Zeit drei Alternativen zur Auswahl (siehe Kap. 11.1.7). Vor dem Compilieren muß das entprechende File **ZREKON*.F** nach **ZREKON.F** kopiert werden.

11.7 Compilieren, Linken und Programmstart

Vor dem Programmstart müssen folgende Dateien compiliert und gelinkt werden:

ZNSCNG2.F	**MWQUAN.F**	**MWSTEP.F**
ZOUTPUT.F	**ZINPUT.F**	**ZREKON.F**[1]
TEST.F	**ZINITIAL.F**	**FLINV.F**
ZCHEMIE.F[2]	**FLVIS.F**	**RQUELL.F**
ZBOUND.F	**ROELSG.F**[3]	**TRKO.F**[4]
ZCFD.F	**ZCOMM.F**	**ZINOUT.F**

Dies geschieht automatisch mit dem Aufruf

make -f znscng2.m

des MAKE-Files **znscng2.m**. Daraus resultiert das lauffähige Programm **znscng2.out**. Dieses wird aufgerufen mit

znscng2.out <'input'>.

Dabei steht **input** für den Namen der Datei, in der die Parameter für die Strömungssimulation und die erforderlichen Dateinamen stehen. Die Dateistruktur kann Kap. 11.3.2.1 entnommen werden. Der Rest ist identisch zu Kapitel 10.

[1] Der Benutzer hat zur Zeit drei Alternativen zur Auswahl (siehe Kap. 11.1.7).
[2] Der Benutzer hat zur Zeit zwei Alternativen zur Auswahl (siehe Kap. 6.2.1).
[3] Der Benutzer hat zur Zeit zwei Alternativen zur Auswahl (siehe Kap. 4.5.4.3).
[4] Der Benutzer hat zur Zeit drei Alternativen zur Auswahl (siehe Kap. 6.3).

12 Paralleles Konzept

Die Rechenzeiten des in den Kapiteln 4 bis 10 beschriebenen Programms **NSCNG2** sind sehr hoch. Für eine stationäre Lösung einer Staupunktströmung werden bei einer Mach–26–Strömung etwa 20000 Zeitschritte mit einer Schrittweite von 10^{-8} sec benötigt. Auf einem aus 33×147 Punkten bestehenden Gitter bedeutet dies eine Laufzeit von etwa 45000 cpu sec auf einer CRAY–Y/MP. Dies liegt insbesondere an

- dem Verfahren hoher Ordnung,
- dem Riemannlöser und
- dem Transport– und Diffusionsmodell.

Um diesen Aufwand zu reduzieren, ist es erforderlich, möglichst viele Programmschritte parallel auszuführen. Um dabei die Programmstruktur nicht wesentlich zu verändern, ist es sinnvoll, daß Rechengebiet in viele kleinere Teilgebiete zu zerlegen und in jedem dieser Gebiete die Strömung gleichzeitig zu berechnen. Nach jedem Zeitschritt ist dann nur ein Austausch der Informationen über die Ränder der Teilgebiete erforderlich. Dieses zonale Konzept wurde in Kapitel 11 beschrieben. Es soll im folgenden gezeigt werden, wie dieses Konzept parallelisiert werden kann. Ein Problem, das sich hier dem Anwender stellt, ist die Vielzahl heute existierender paralleler Architekturen. Hier ist es sinnvoll, Schnittstellen zu entwickeln, die auf möglichst viele der gängigen Modelle adaptiert werden können. Im Rahmen des DFG–Forschungsprojektes *Scientific Parallel Computing* an der TU Chemnitz–Zwickau wurde eine Programmbibliothek zur Entwicklung paralleler Algorithmen erstellt (vgl. [6]). Mit diesen Routinen ist es möglich, das parallelisierte Programm auf den folgenden parallelen Archtikturen laufen zu lassen:

- Transputersysteme (3L, Helios, Parix),
- $n-$Cube,
- Workstation–Cluster.

Aus dieser Bibliothek werden die folgenden Routinen benutzt:

- **TRINIT**, **TRCLOSE**,
- **SEND_NODE_1**, **RECV_NODE_1**,
- **TREE_UP**, **TREE_DOWN**.

Diese steuern die Initialisierung der Prozessoren, das Versenden und Empfangen von Daten zwischen den Prozessoren und das ordnungsgemäße Abschließen der Prozessorverbindungen nach Beendigung der Rechnung.
Im folgenden soll die Parallelisierung des zonalen Konzeptes beschrieben werden. Dazu werden jedoch nur die gegenüber Kapitel 11.1 erforderlichen Änderungen beschrieben. Diese stehen in Files, die mit einem **P** statt einem **Z** beginnen (vgl. 13.3).

12.1 Programmstruktur

12.1.1 PNSCNG2 (Hauptprogramm)

Die Programmstruktur ist in Abbildung 16 dargestellt. Die Änderung gegenüber Kapitel 11.1.1 besteht darin, daß das Programm auf **NPROC**($= 2^{\mathbf{NCUBE}}$) Prozessoren gleichzeitig läuft. Jeder Prozessor ist für eine Zone zuständig. Somit muß die Initialisierung der Rechnung (vgl. Kap. 11.1.3) nur auf einem Prozessor — dem Hauptprozessor (**ICH** $= 0$) — durchgeführt werden. Die initialisierten Werte werden dann an die Unterprozessoren geschickt (vgl. Routine **SEND_NODE_1**) und von diesen empfangen (vgl. Routine **RECV_NODE_1**). Diese müssen nur einmal verschickt werden, da die Speicherbereiche der einzelnen Prozessoren für die gleiche Variable verschieden sind. Vor der Berechnung der chemischen Produktionsraten bzw. der Strömung müssen die aktuellen Zonenränder vom Hauptprozessor an alle Unterprozessoren verschickt werden. Nach der gleichzeitigen Berechnung auf allen Prozessoren für einen Zeitschritt, werden die Zonenränder an den Hauptprozessor zurückgeschickt und dort aktualisiert (vgl. Routine **ZRANDUP**).
Man beachte, daß der Hauptprozessor eine Doppelfunktion ausübt. Zum einen ist er für die Kommunikation zwischen den einzelnen Zonen (Prozessoren) zuständig, und zum anderen wird auf ihm die Berechnung für die erste Zone durchgeführt.
Die Struktur des Hauptprogramms ist somit wie folgt gegeben:

TRINIT

Initialisierung des parallelen Programms auf allen Prozessoren;

ZINPUT (nur auf dem Hauptprozessor (**ICH** $= 0$)

Eingabe der strömungs– und diskretisierungsspezifischen Parameter für alle Zonen;

ZINITIAL (nur auf dem Hauptprozessor (**ICH** $= 0$)

Einlesen der Daten und des Gitters für alle Zonen;

Initialisierung der charakteristischen Anströmwerte für die durchzuführende Strömungssimulation für alle Zonen;

Initialisierung der physikalischen Größen;

Initialisierung der Anströmwerte für alle Zonen;

Initialisierung aller gitterspezifischen Größen für alle Zonen;

Initialisierung der rekonstruktionsspezifischen Größen für alle Zonen;

Initialisierung aller Zonenränder;

PPARSND (nur auf dem Hauptprozessor (**ICH** $= 0$)

Senden der zonenabhängigen und zonenunabhängigen Variablen und Parameter an die Unterprozessoren;

Ablegen dieser Daten für die erste Zone in den entsprechenden zonenunabhängigen Größen auf dem Hauptprozessoren ab;

PPARRCV (nur auf den Unterprozessoren (**ICH** $\neq 0$)

Empfangen der zonenabhängigen und zonenunabhängigen Variablen und Parameter der Zone l von dem Hauptprozessor auf dem entsprechenden Unterprozessor (**ICH** $= l-1$);

Ablegen dieser Daten in den entsprechenden zonenunabhängigen Größen des Prozessors;

PCHEMIE (optional, falls chemisches Nichtgleichgewicht mit Operatorsplitting modelliert werden soll)

Parallele Berechnung der Diskretisierung von (29) bzw. (31) für alle Zonen;

PCFD

Parallele Berechnung der Strömung in allen Zonen;

Parallele Berechnung der Bilanzgrößen (2) auf dem neuen Zeitlevel in allen Zonen;

PZONEIN

Senden der Zonenränder der Zone l vom Hauptprozessor an den Unterprozessor (**ICH** $= l-1$);

Empfangen der Zonenränder der Zone l auf dem Unterprozessor (**ICH** $= l-1$);

Ablegen dieser Daten in die entsprechenden zonenunabhängigen Variablen;

BOUND

Parallele Berechnung der Bilanzgrößen (2) in den Randzellen des Gitters und den Zonenränder ;

PZONEOUT

Senden der Zonenränder der Zone l von dem Unterprozessor (**ICH** $= l-1$) an den Hauptprozessor;

Empfangen der Zonenränder der Zone l vom Unterprozessor (**ICH** $= l-1$) auf dem Hauptprozessor (**ICH** $= 0$);

Ablegen dieser Daten in die entsprechenden zonenabhängigen Variablen;

DATABACK (nur auf dem Hauptprozessor (**ICH** $= 0$)

Senden der neu berechneten Bilanzgrößen der Zone l von dem Unterprozessor (**ICH** $= l-1$) an den Hauptprozessor (**ICH** $= 0$);

Empfangen der neu berechneten Bilanzgrößen der Zone l von dem Unterprozessor (**ICH**$= l-1$) auf dem Hauptprozessor (**ICH**$= 0$);

Ablegen dieser Daten in die entsprechenden zonenabhängigen Variablen;

ZOUTPUT (nur auf dem Hauptprozessor (**ICH** $= 0$)

Ausgabe der neuen Mittelwerte nach **NSTEPS** durchgeführten Zeitschritten;

TRCLOSE

Beenden des parallelen Programms auf allen Prozessoren.

12.1.2 PCHEMIE

Die Berechnung der chemischen Produktionsraten in einer Zone erfolgt wie bisher in der Routine **CHEMIE** (vgl. Kap. 4.4). Die Routine **PCHEMIE** steuert die parallele Berechnung in allen Zonen $l = 1$,**ZONANZ**. Dazu müssen vor dem Aufruf von **CHEMIE** die Zonenrandwerte der Zone l vom Hauptprozessor an den Unterprozessor (**ICH** $= 0$) verschickt werden und dort in die entsprechenden zonenunabhängigen Variablen (vgl. Routine **PZONEIN**) abgelegt werden. Nach der Berechnung auf dem Unterprozessor werden die neuen Zonenrandwerte wieder an den Hauptprozessor zurückgeschickt und dort in die zonenabhängigen Variablen geschrieben werden (vgl. Routine **PZONEOUT**). Die Zonenränder müssen, nachdem die Produktionsterme in zwei aneinandergrenzenden Zonen berechnet wurden, ebenfalls auf den aktuellen Stand gebracht werden. Dies geschieht in der Routine **ZRANDUP** (vgl. Kap. 11.2).

CHEMIE

Berechnung der chemischen Produktionsraten s_i in der Zone l;

Aufdatieren der Partialdichten zu jeder Zelle **im** Strömungsgebiet mittels eines expliziten Eulerschrittes (vgl. Kap. 4.4), falls der Quellterm der chemischen Produktionsraten von den übrigen Gleichungen gesplittet werden soll (**OFLAG** $= 1$). Diese Routine wird parallel für alle Zonen durchgeführt;

ZBOUT

Aufdatieren der Zonenränder (**MWI**, **MWRHOI**) der Zone l vor Verlassen des zonenunabhängigen Bereiches;

PZONEOUT

Senden der Zonenränder der Zone l von dem Unterprozessor (**ICH** $= l - 1$) an den Hauptprozessor;

Empfangen der Zonenränder der Zone l vom Unterprozessor (**ICH**$= l - 1$) auf dem Hauptprozessor (**ICH** $= 0$);

Ablegen dieser Daten in die entsprechenden zonenabhängigen Variablen;

ZRANDUP(l) (nur auf dem Hauptprozessor (**ICH** $= 0$)

Aktualisierung der Zonenränder $l = 1,$**RZANZ** nachdem die Produktionsterme in zwei aneinandergrenzenden Zonen berechnet wurden, d.h. die am Zonenrand anliegenden Werte einer Zone (**ZMWI**, **ZMWRHOI**) müssen als Randwerte (**ZMWA**, **ZMWRHOA**) in der angrenzenden Zone gesetzt werden;

12.1.3 PCFD

Die Berechnung der Strömung in einer Zone erfolgt wie bisher in der Routine **CFD** (vgl. Kap. 4.5). Die Routine **PCFD** steuert die parallel Berechnung in allen Zonen $l = 1$,**ZONANZ**. Dazu müssen vor dem Aufruf von **CFD** die Zonenrandwerte der Zone l vom Hauptprozessor an den Unterprozessor (**ICH** $= 0$) verschickt werden und dort in die entsprechenden zonenunabhängigen Variablen (vgl. Routine **PZONEIN**) abgelegt

werden. Nach der Berechnung auf dem Unterprozessor werden die neuen Zonenrandwerte wieder an den Hauptprozessor zurückgeschickt und dort in die zonenabhängigen Variablen abgelegt (vgl. Routine **PZONEOUT**). Wird kein Operatorsplitting durchgeführt, so muß vor der Berechnung der Strömung die Routine **CHEMIE** aufgerufen werden, um den Quellterm **QUELLE** zu berechnen. Die Zonenränder müssen, nachdem die Produktionsterme in zwei aneinandergrenzenden Zonen berechnet wurden, ebenfalls auf den aktuellen Stand gebracht werden. Dies geschieht in der Routine **ZRANDUP** (vgl. Kap. 11.2).

PZONEIN

Senden der Zonenränder der Zone l vom Hauptprozessor an den Unterprozessor (**ICH** $= l - 1$);

Empfangen der Zonenränder der Zone l auf dem Unterprozessor (**ICH** $= l - 1$);

Ablegen dieser Daten in die entsprechenden zonenunabhängigen Variablen;

CHEMIE (optional, falls chemisches Nichtgleichgewicht modelliert werden soll ohne Operatorsplitting)

Berechnung der chemischen Produktionsraten s_i in der Zone l;

Aufdatieren der Partialdichten zu jeder Zelle **im** Strömungsgebiet mittels eines expliziten Eulerschrittes (vgl. Kap. 4.4), falls der Quellterm der chemischen Produktionsraten von den übrigen Gleichungen gesplittet werden soll (**OFLAG** $= 1$). Diese Routine wird parallel für alle Zonen durchgeführt;

CFD

Berechnung der Bilanzgrößen (2) auf dem neuen Zeitlevel in der Zone l;

ZBOUT

Aufdatieren der Zonenränder (**MWI**, **MWRHOI**) der Zone l vor Verlassen des zonenunabhängigen Bereiches;

PZONEOUT

Senden der Zonenränder der Zone l von dem Unterprozessor (**ICH** $= l - 1$) an den Hauptprozessor;

Empfangen der Zonenränder der Zone l vom Unterprozessor (**ICH** $= l - 1$) auf dem Hauptprozessor (**ICH** $= 0$);

Ablegen dieser Daten in die entsprechenden zonenabhängigen Variablen;

ZRANDUP(l) (nur auf dem Hauptprozessor (**ICH** $= 0$)

Aktualisierung der Zonenränder $l = 1, \mathbf{RZANZ}$ nachdem die Strömung in zwei aneinandergrenzenden Zonen berechnet wurden, d.h. die am Zonenrand anliegenden Werte einer Zone (**ZMWI**, **ZMWRHOI**) müssen als Randwerte (**ZMWA**, **ZMWRHOA**) in der angrenzenden Zone gesetzt werden;

12.2 Kommunikation zwischen den Zonen

Da die Initialisierung gemäß den Routinen **ZINPUT** und **ZINITIAL** nur auf dem Hauptprozessor stattfindet, müssen die initialisierten Werte an die Unterprozessoren gesendet werden und dort von den entsprechenden Variablen empfangen werden. Ferner erfolgt die Berechnung der chemischen Produktionsraten und der Strömung für jede Zone parallel in den Modulen **PCHEMIE** und **PCFD**. Dazu müssen vor dem Aufruf dieser Routinen die aktuellen Zonenränder an die Unterprozessoren geschickt werden und nach der Bearbeitung die neuen Werte an den Hauptprozessor zurückgesendet werden, damit ein Austausch zwischen zwei benachbarten Zonen über den Zonenrand stattfindet. Nach Beendigung der gesamten Simulation müssen die neuen Bilanzgrößen der einzelnen Zonen, die sich nur auf den Unterprozessoren befinden an den Hauptprozessor geschickt werden und dort von den zonenabhängigen Variablen empfangen werden. Dies alles wird durch die fünf folgenden Routinen gewährleistet:

PPARSND

Senden der folgenden zonenabhängigen und zonenunabhängigen Variablen und Parameter

Parameter	**JANZ**	**JANZMU**	**JANZBD**	**JANZAK**
	CFLAG	**OFLAG**	**WFLAG**	**PFLAG**
	KWFLAG	**NSFLAG**		
	NSTEPS	**SSTEPS**	**ASTEPS**	**ZSTEPS**
	DELTAT			
	TMIN	**TMINMU**	**TMINBD**	**TMINAK**
	TMAX	**TMAXMU**	**TMAXBD**	**TMAXAK**
	DTP	**DTPMU**	**DTPBD**	**DTPAK**
	MREF	**NREF**	**PREF**	**TREF**
	RHOREF	**REREF**	**FEREF**	**MUEREF**

Zonengröße	**ZMZAHLX**(l)	**ZMZAHLY**(l)
Zonale Parameter	**ZGFLAG**(l)	**ZRFLAG**(i,l)
	ZKFLAG(i,l)	**ZTWALL**(i,l)
zonales Gitter	**ZGRID**(j,k,i,l)	**ZNXI**(j,k,i,l)
	ZNETA(j,k,i,l)	**ZARCXI**(j,k,l)
	ZARCETA(j,k,l)	**ZVOL**(j,k,l)
zonale Rekonstruktion	**ZMPKXI**(j,k,l)	**ZMPKETA**(j,k,l)
zonale Bilanzgrößen	**ZMW**(j,k,i,l)	**ZMWRHO**(j,k,i,l)
Zonenrandwerte	**ZMWI**(j,k,i,l)	**ZMWRHOI**(j,k,i,l)
	ZMWA(j,k,i,l)	**ZMWRHOA**(j,k,i,l)

an die Unterprozessoren, und Ablegen dieser Daten für die erste Zone in den zonenunabhängigen Größen auf dem Hauptprozessoren ab;

PPARRCV

Empfangen der zonenabhängigen und zonenunabhängigen Variablen und Parameter der Zone l von dem Hauptprozessor auf dem entsprechenden Unterprozessor (**ICH** $= l-1$), und lege diese in den zonenunabhängigen Größen des Prozessors ab:

Parameter	**JANZ**,	**JANZMU**,	**JANZBD**,	**JANZAK**,
	CFLAG,	**OFLAG**,	**WFLAG**,	**PFLAG**,
	KWFLAG,	**NSFLAG**,		
	NSTEPS,	**SSTEPS**,	**ASTEPS**,	**ZSTEPS**,
	DELTAT,			
	TMIN,	**TMINMU**,	**TMINBD**,	**TMINAK**,
	TMAX,	**TMAXMU**,	**TMAXBD**,	**TMAXAK**,
	DTP,	**DTPMU**,	**DTPBD**,	**DTPAK**,
	MREF,	**NREF**,	**PREF**,	**TREF**,
	RHOREF,	**REREF**,	**FEREF**,	**MUEREF**

Zonengröße	**MZAHLX**	**MZAHLY**
Zonale Parameter	**GFLAG**	**RFLAG**(i)
	KFLAG(i)	**TWALL**(i)
zonales Gitter	**GRID**(j,k,i)	**NXI**(j,k,i)
	NETA(j,k,i)	**ARCXI**(j,k)
	ARCETA(j,k)	**VOL**(j,k)
zonale Rekonstruktion	**MPKXI**(j,k)	**MPKETA**(j,k)
zonale Bilanzgrößen	**MW**(j,k,i)	**MWRHO**(j,k,i)
Zonenrandwerte	**MWI**(j,k,i)	**MWRHOI**(j,k,i)
	MWA(j,k,i)	**MWRHOA**(j,k,i)

PZONEIN

Vor der Berechnung der chemischen Produktionsraten oder der Strömung innerhalb der Zone l werden hier die Zonenränder der Zone l

Zonenrandwerte	**ZMWI**(j,k,i,l)	**ZMWRHOI**(j,k,i,l)
	ZMWA(j,k,i,l)	**ZMWRHOA**(j,k,i,l)

vom Hauptprozessor an den Unterprozessor (**ICH** $= l-1$) gesendet. Dieser Unterprozessor empfängt die Zonenränder der Zone l und legt diese in die entsprechenden zonenunabhängigen Variablen ab:

Zonenrandwerte	**MWI**(j,k,i)	**MWRHOI**(j,k,i)
	MWA(j,k,i)	**MWRHOA**(j,k,i)

PZONEOUT

Nachdem die Bilanzgrößen in einer der beiden Modulen **CHEMIE** und **CFD** für die Zonen $l = 1$,**ZONANZ** berechnet wurden, müssen die Zonenränder aktualisiert werden. Dazu werden die Zonenränder der Zone l von dem Unterprozessor (**ICH** $= l-1$)

Zonenrandwerte	**MWI**(j,k,i)	**MWRHOI**(j,k,i)
	MWA(j,k,i)	**MWRHOA**(j,k,i)

an den Hauptprozessor verschickt. Der Hauptprozessor (**ICH** $= 0$) empfängt die Zonenränder der Zone l vom Unterprozessor (**ICH** $= l-1$) und legt diese dort in die entsprechenden zonenabhängigen Variablen;

Zonenrandwerte	**ZMWI**(j,k,i,l)	**ZMWRHOI**(j,k,i,l)
	ZMWA(j,k,i,l)	**ZMWRHOA**(j,k,i,l)

DATABACK

Sendet die neu berechneten Bilanzgrößen der Zone l von dem Unterprozessor (**ICH** = $l-1$) an den Hauptprozessor (**ICH** = 0). Diese Werte werden auf dem Hauptprozessor (**ICH** = 0) empfangen und dort in die entsprechenden zonenabhängigen Variablen der Zone l abgelegt;

12.3 Kommunikation zwischen den Prozessoren

Aus der Bibliothek zur Entwicklung paralleler Algorithmen (vgl. [6]) wurden folgende Routinen benutzt:

TRINIT(0)

Initialisierung des parallelen Programms auf allen Prozessoren;

TRCLOSE

Beenden des Hauptprogramms auf allen Prozessoren;

SEND_NODE_1($nWords, Data, NrTo$)

Senden der Daten $Data$ der Länge $nWords$ (1 Wort = 4 Byte) vom aktuellen Prozessor an den Prozessor $NrTo$;

RECV_NODE_1($nWords, Data, NrFrom$)

Empfangen der Daten der Länge $nWords$ (1 Wort = 4 Byte) vom aktuellen Prozessor von dem Prozessor $NrFrom$ und schreibe diese nach $Data$;

TREE_UP($Nlocal, Xin, Nout, Xout, MaxX$)

Empfangen der Daten $Xout$ der Länge $Nout$ (1 Wort = 4 Byte) auf dem Hauptprozessor (**ICH** = 0). Die maximale Länge von $Xout$ beträgt $MaxX$. Dieses Datenfeld setzt sich dabei aus den Daten Xin der Länge $Nlocal$, die lokal auf allen Prozessoren verteilt vorliegen, zusammen;

TREE_DOWN($nWords, Data$)

Senden der Daten $data$ der Länge $nWords$ (1 Wort = 4 Byte) vom Hauptprozessor (**ICH** = 0) an alle Unterprozessoren.

12.4 Globale Variablen

Zur Kommunikation zwischen den Prozessoren (vgl. Kap. 12.3) werden einige globale Variablen benutzt. Diese stehen in der INCLUDE–Datei **trnet.inc**, die zu den Routinen dazugelinkt werden muß, die auf Routinen der Bibliothek zur Entwicklung paralleler Algorithmen zugreifen.

12.4.1 INCLUDE–Datei trnet.inc

Die globalen Kommunikationsvariablen zwischen den Prozessoren sind:

NCUBE Hypercube–Dimension (Anzahl der Links je Prozessor)

NPROC Anzahl der Prozessoren ($= 2^{\mathbf{NCUBE}}$)

ICH eigene Prozessornummer ($0, \ldots, \mathbf{NPROC} - 1$)

ICHRING Position des Prozessors innerhalb des in den Hypercube eingebetteten Prozessorrings

LFORW Linknummer nach *vorn* im Prozessorring

LBACK Linknummer nach *hinten* im Prozessorring

NODENR

Die Beschreibung ist [6], S. 10, entnommen.

12.5 Compilieren, Linken und Programmstart

Vor dem Programmstart müssen folgende Dateien compiliert und gelinkt werden:

PNSCNG2.F	**MWQUAN.F**	**MWSTEP.F**
ZOUTPUT.F	**ZINPUT.F**	**ZREKON.F**[1]
TEST.F	**ZINITIAL.F**	**FLINV.F**
PCHEMIE.F[2]	**FLVIS.F**	**RQUELL.F**
ZBOUND.F	**ROELSG.F**[3]	**TRKO.F**[4]
PCFD.F	**PCOMM.F**	**ZINOUT.F**

Dies geschieht automatisch mit dem Aufruf

make -f pnscng2.m

des MAKE-Files **pnscng2.m**. Das lauffähige Programm heißt **pnscng2.out**. Dieses muß an alle Prozessoren verschickt werden.
Man beachte, daß hier die entsprechenden Bibliotheken mit den Routinen für die Parallelisierung dazu gelinkt werden müssen. Dies sind zum einen die maschinenabhängigen Routinen (z.B. PVM–Software für Workstation–Cluster, vgl. [3]) und zum anderen die Routinen der Chemnitzer Gruppe (vgl. [6] bzw. Kap. 12.3).

[1] Der Benutzer hat zur Zeit drei Alternativen zur Auswahl (siehe Kap. 11.1.7).
[2] Der Benutzer hat zur Zeit zwei Alternativen zur Auswahl (siehe Kap. 6.2.1).
[3] Der Benutzer hat zur Zeit zwei Alternativen zur Auswahl (siehe Kap. 4.5.4.3).
[4] Der Benutzer hat zur Zeit drei Alternativen zur Auswahl (siehe Kap. 6.3).

Bisher wurde das Programm nur auf einem Workstation–Cluster bestehend aus maximal vier IRIS–Maschinen laufen gelassen. Zur Konfiguration des Clusters muß eine Konfigurationsdatei (hier: **pvmconfig**), in der alle zu verbindenen Rechner wie folgt aufgeführt sind

$$\mathbf{IP} = hostname \qquad \mathbf{dx} = daemon \qquad \mathbf{ep} = pfad$$

hostname	Name oder FTP–Nummer des zu verbindenden Rechners
daemon	Dämon, der die Kommunikation zwischen den Rechnern ermöglicht
pfad	Suchpfad für das lauffähige Benutzerprogramms

mit

$$\mathbf{pvm} = \mathbf{pvmconfig}$$

aufgerufen werden.
Nach der Konfiguration des Clusters kann das Programm mit

$$\mathbf{pnscng2.out} \quad < \mathbf{NCUBE} > \quad <\text{'}\mathbf{input}\text{'}>.$$

gestartet werden. Dabei gibt **NCUBE** die Zweierpotenz der Prozessorenzahl an, und **input** steht für den Namen der Datei, in der die Parameter für die Strömungssimulation und die erforderlichen Dateinamen stehen. Die Dateistruktur kann Kap. 11.3.2.1 entnommen werden. Der Rest ist identisch zu Kapitel 10.

13 Anhang

13.1 Abbildungen

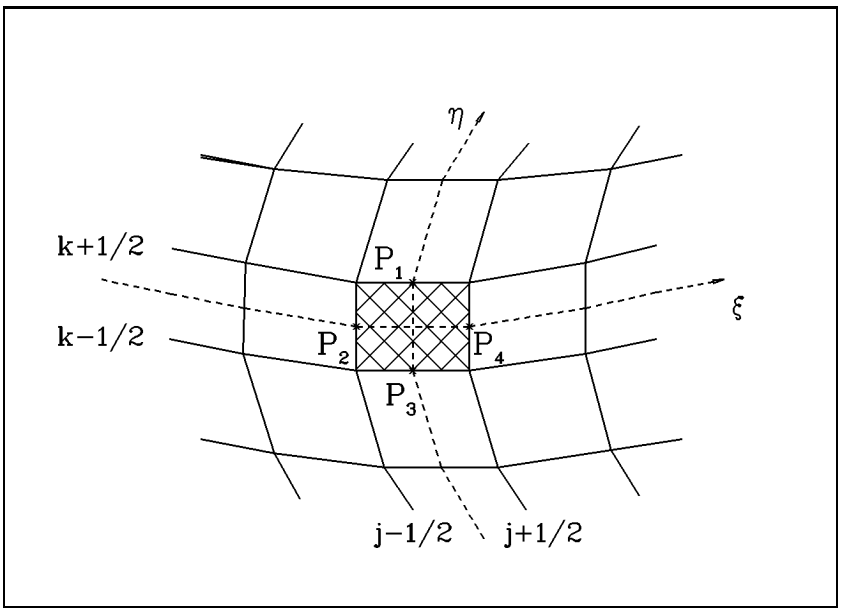

Abbildung 1: (entnommen aus [13], S. 33)

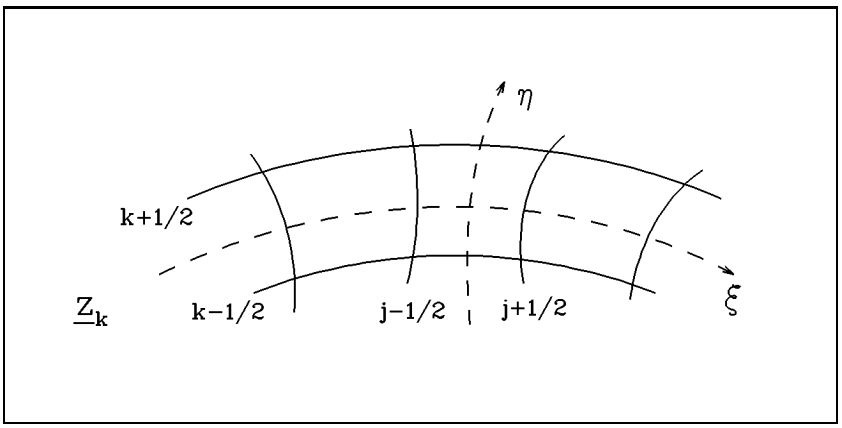

Abbildung 2: (entnommen aus [13])

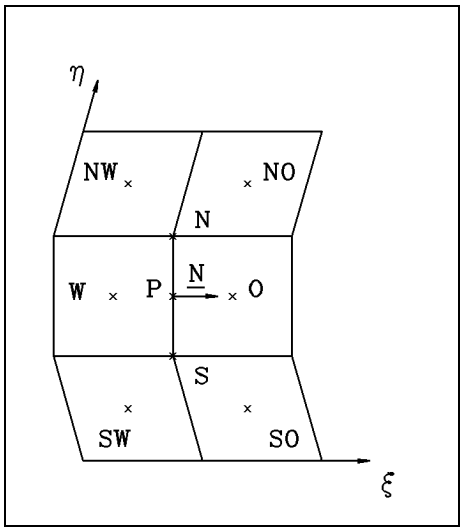

Abbildung 3: (entnommen aus [13])

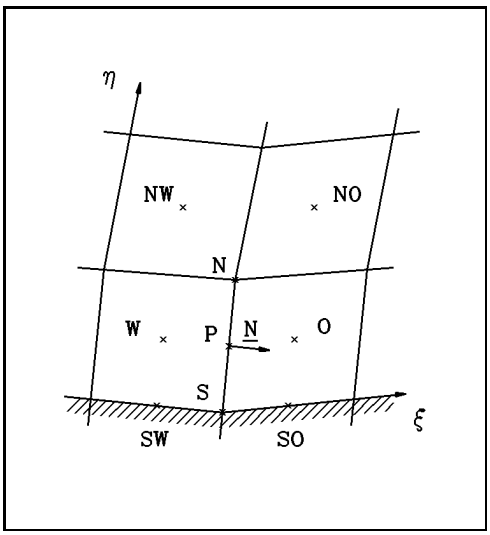

Abbildung 4: (entnommen aus [13])

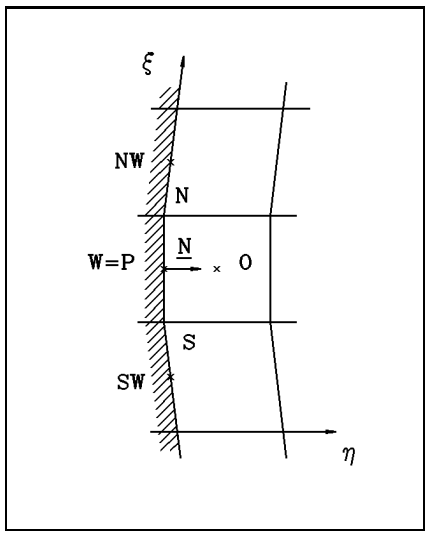

Abbildung 5: (entnommen aus [13])

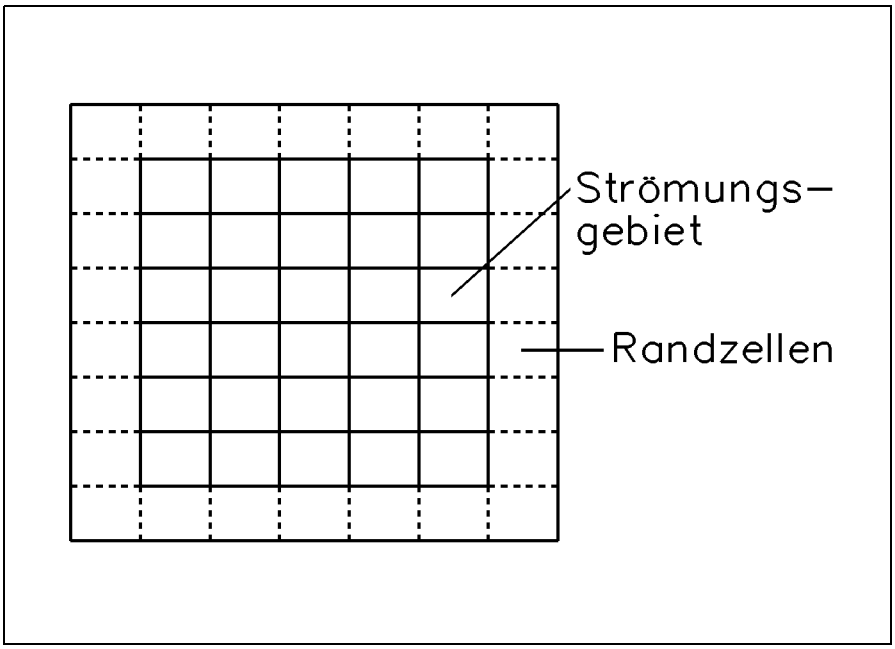

Abbildung 6: Aufbau einer Zone

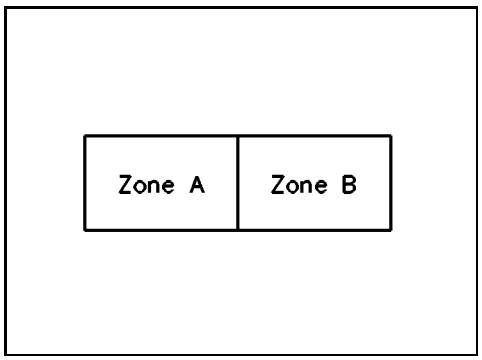

Abbildung 7: richtig

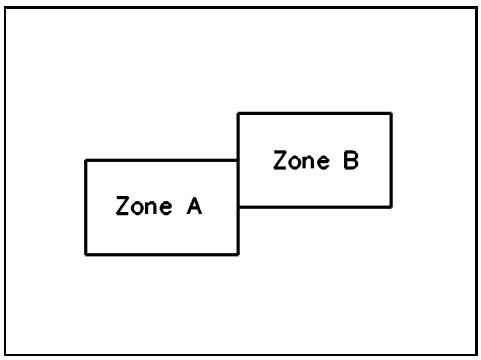

Abbildung 8: falsch

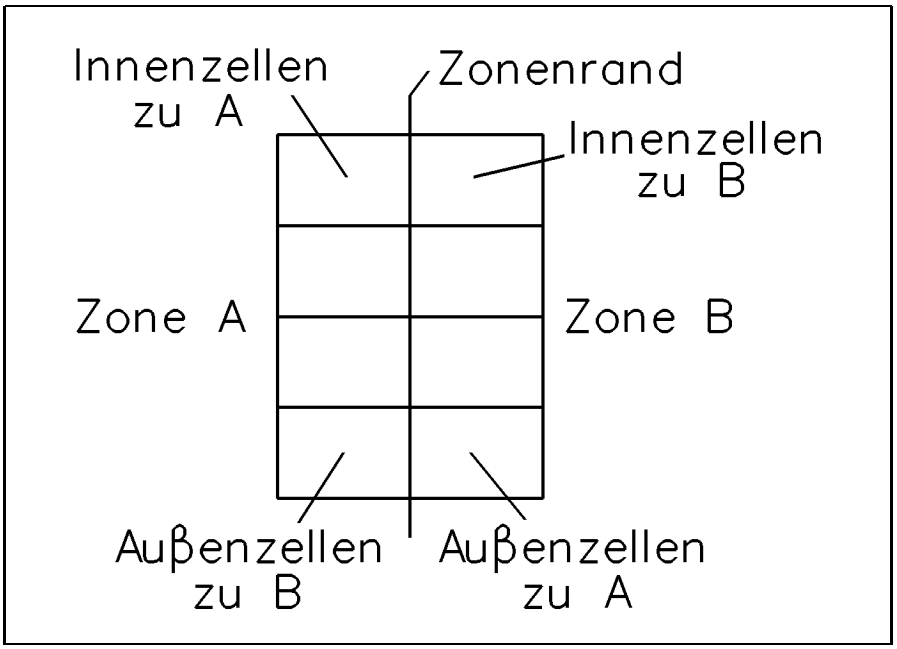

Abbildung 9: richtiger Aufbau eines Zonenrandes

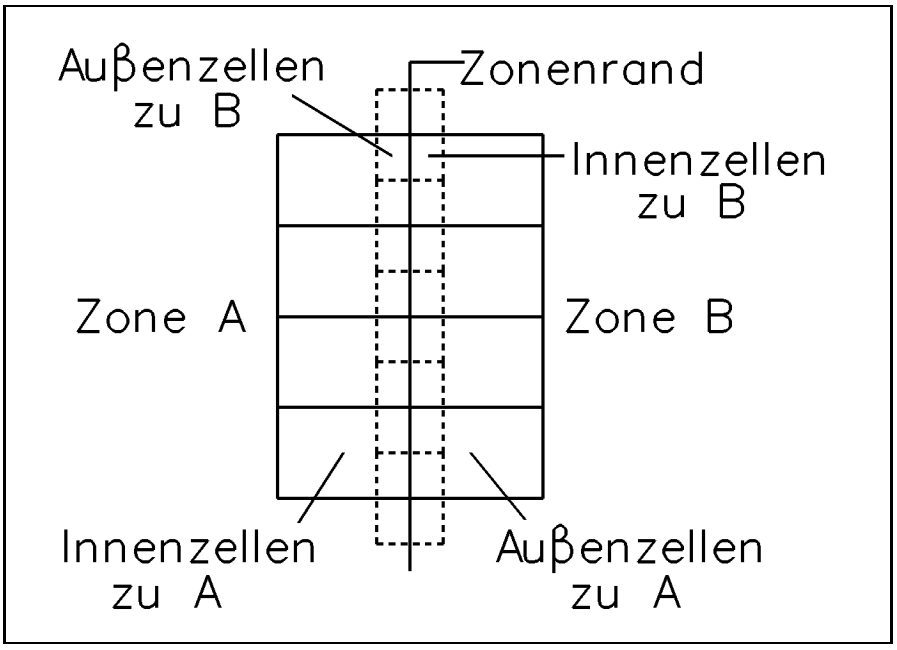

Abbildung 10: falscher Aufbau eines Zonenrandes

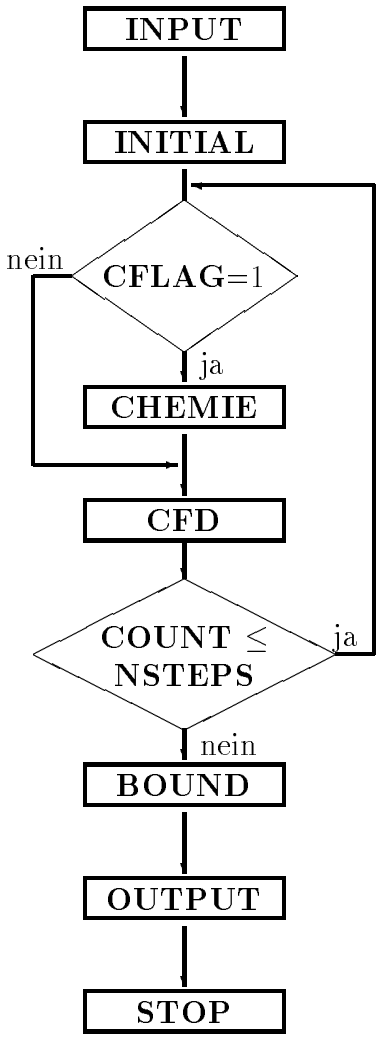

Abbildung 11: Flußdiagramm des Programms **NSCNG2**

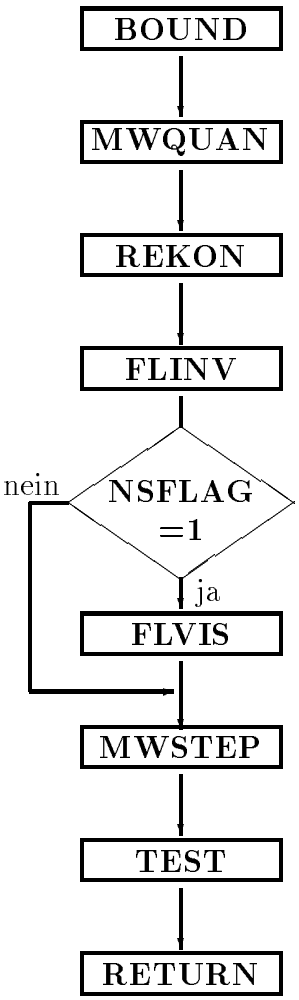

Abbildung 12: Flußdiagramm der Routine **CFD**

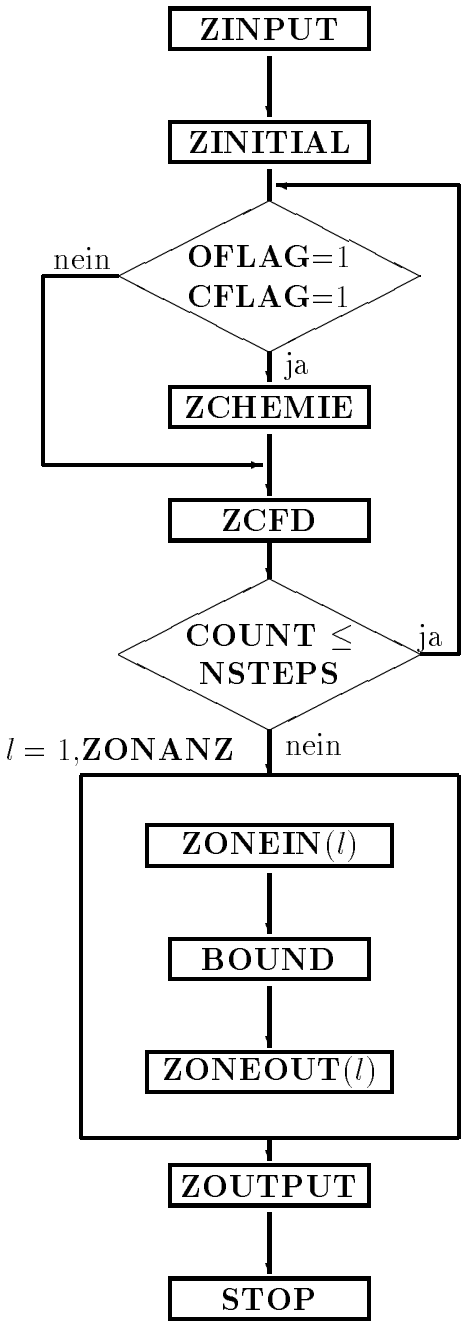

Abbildung 13: Flußdiagramm des Programms **ZNSCNG2**

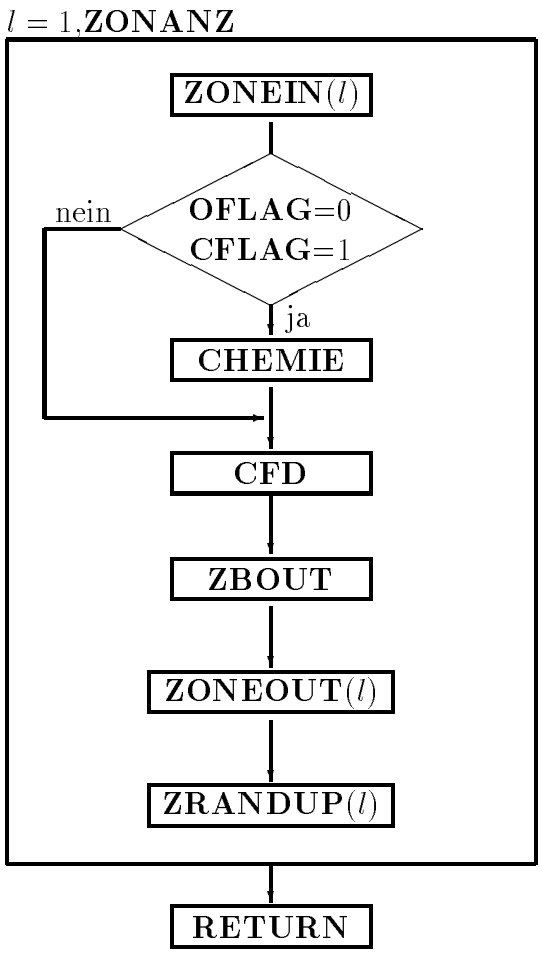

Abbildung 14: Flußdiagramm der Routine **ZCFD**

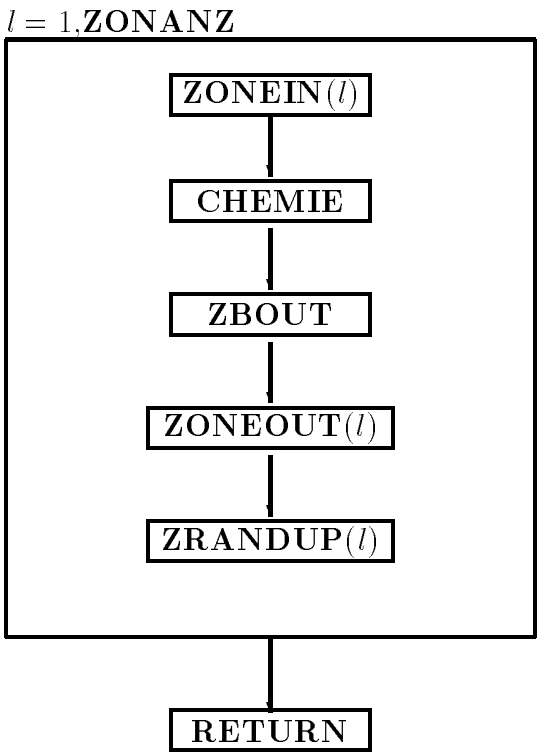

Abbildung 15: Flußdiagramm der Routine **ZCHEMIE**

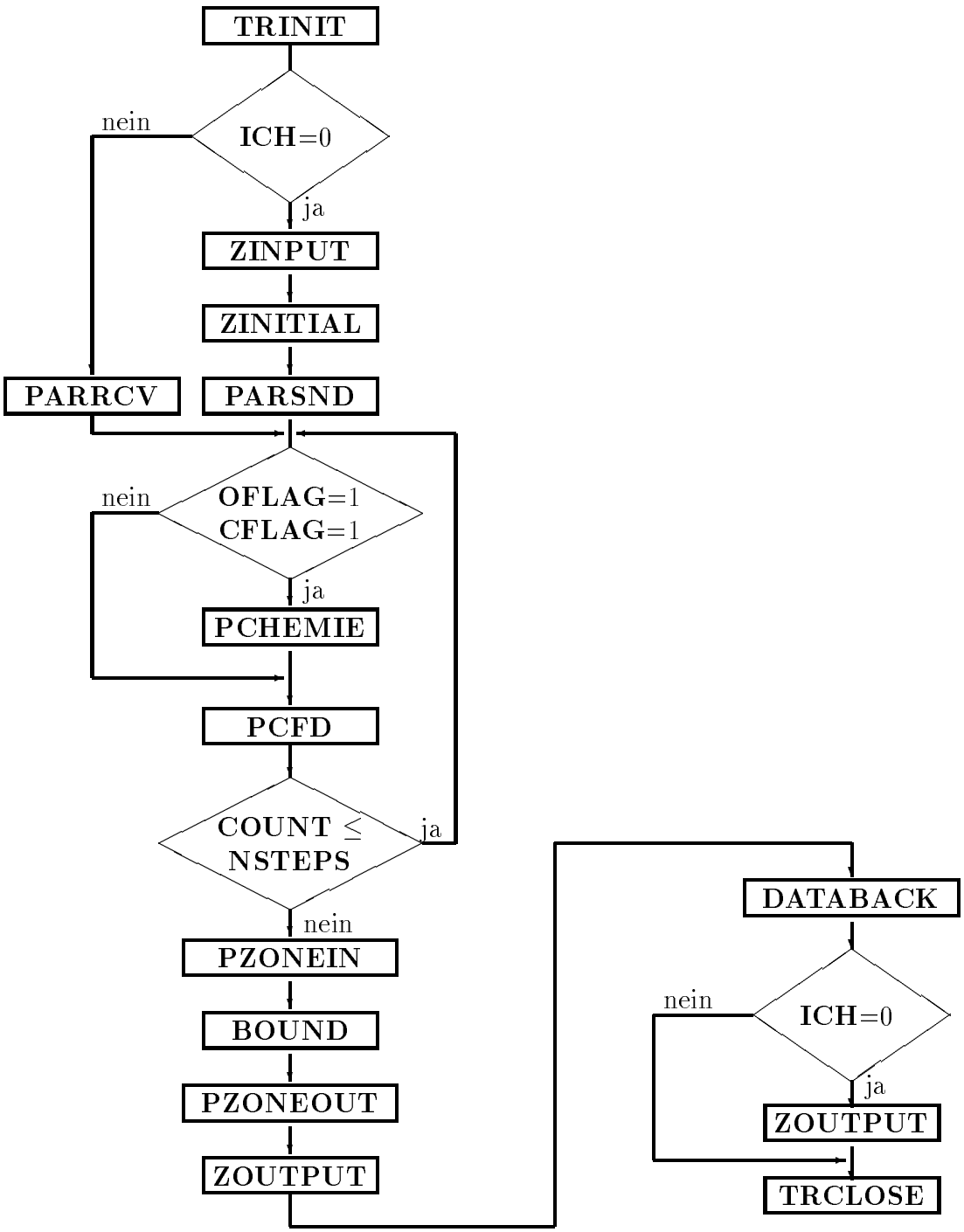

Abbildung 16: Flußdiagramm des Programms **PNSCNG2**

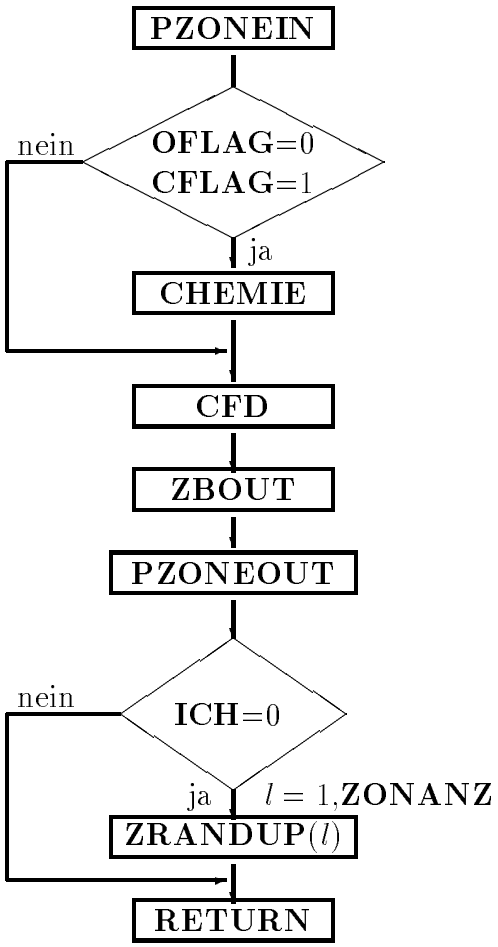

Abbildung 17: Flußdiagramm der Routine **PCFD**

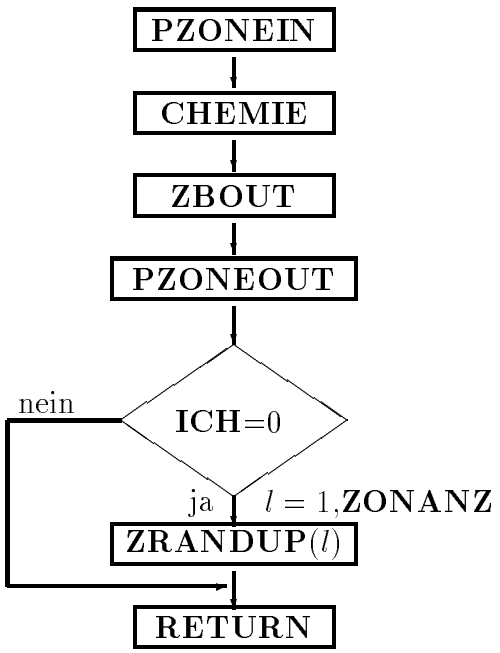

Abbildung 18: Flußdiagramm der Routine **PCHEMIE**

13.2 Zuordnung Datei — Unterprogramm für NSCNG2

Um dem Benutzer das Auffinden von Routinen zu erleichtern, sind im folgenden, nach Dateien sortiert, die darin enthaltenen Unterprogramme aufgeführt:

NSCNG2.F :	**NSCNG2**	(Hauptprogramm)	
CFD.F :	**CFD**		
INPUT.F :	**INPUT**		
INITIAL.F :	**INITIAL**	**DATINI**	**PHYINI**
	REFINI	**NETINI**	**ENORM**
	VPROD	**NORMAL**	**VOLUME**
	ARCLEN	**NUMMER**	
CHEMIE.F :	**CHEMIE**	**CHEPRO**	**EULER**
BOUND.F :	**BOUND**	**EPIBOU**	**WALLS**
	WALLR	**PRABOU**	**REKBOU**
	SYMROT	**WALLV**	**ECKE**
	HYPOUT	**SYMACH**	**HYPIN**
MWQUAN.F	**MWQUAN**	**MWENTH**	**MWMFRA**
	MWTEMP	**MWALTU**	
REKON.F :	**REKINI**	**STAXI**	**REKON**
	STAETA	**REKXI**	**REKETA**
	TAYLOR	**KOMREK**	**STATEL**
	STATER	**RPREK**	**HORNL**
	HORNR	**JACOBI**	
FLINV.F :	**RPS**	**RPSN**	**FIBOU**
RPSOLV.F :	**ROELSG**		
FLVIS.F :	**VFLPAR**	**VFLWAL**	
RQUELL.F :	**RQUXI**	**RQUETA**	**RQVWXI**
	RQWVETA		
PHYQUA.F:	**IET**	**CVT**	**TEMP**
	IEI	**CVI**	
MWSTEP.F:	**MWSTEP**	**MWVIS**	**MWINV**
TEST.F :	**TEST**	**PTEST**	**STEADY**
TRKO.F :	**MDIFF**	**KOEFF**	**VISCOE**
	DVISCO		
OUTPUT.F:	**OUTDAT**	**OUTPUT**	

13.3 Zuordnung Datei — Unterprogramm für ZNSCNG2

Hier werden nur noch die gegenüber Kapitel 13.2 veränderten oder neuen Files erfaßt:

ZNSCNG2.F	: **ZNSCNG2**	(Hauptprogramm)	
ZCFD.F	: **ZCFD**	**CFD**	
ZINPUT.F	: **ZINPUT**		
ZINITIAL.F	: **ZINITIAL**	**ZDATINI**	**PHYINI**
	ZREFINI	**ZNETINI**	**ENORM**
	VPROD	**ZNORMAL**	**ZVOLUME**
	ZARCLEN	**NUMMER**	**ZBINI**
ZCHEMIE.F	: **ZCHEMIE**	**CHEPRO**	**EULER**
	CHEMIE		
ZBOUND.F	: **BOUND**	**EPIBOU**	**WALLS**
	WALLR	**PRABOU**	**REKBOU**
	SYMROT	**WALLV**	**ECKE**
	HYPOUT	**SYMACH**	**HYPIN**
	ZBIN		
ZREKON.F	: **ZREKINI**	**STAXI**	**REKON**
	STAETA	**REKXI**	**REKETA**
	TAYLOR	**KOMREK**	**STATEL**
	STATER	**RPREK**	**HORNL**
	HORNR	**JACOBI**	
ZOUTPUT.F	: **ZOUTDAT**	**ZOUTPUT**	
ZCOMM.F	: **ZONEIN**	**ZONEOUT**	
ZINOUT.F	: **ZRANDUP**	**ZBOUT**	

13.4 Zuordnung Datei — Unterprogramm für PNSCNG2

Hier werden nur noch die gegenüber Kapitel 13.2 und 13.3 veränderten oder neuen Files erfaßt:

PNSCNG2.F:	**PNSCNG2**	(Hauptprogramm)	
PCFD.F :	**PCFD**	**CFD**	
PCHEMIE.F:	**PCHEMIE**	**CHEPRO**	**EULER**
	CHEMIE		
PCOMM.F :	**PZONEIN**	**PZONEOUT**	**PPARSND**
	PPARRCV	**DATABACK**	

Literatur

[1] Chakravarthy S.R., Engquist B., Harten A., Osher S., *"Uniformly High Order Accurate Essentially Non-Oscillatory Schemes III"*, J. Comp. Phys., Vol. 71, 1987, pp. 231 - 303

[2] Esser, B., *"Die Zustandsgrößen im Stoßwellenkanal als Ergebnisse eines exakten Riemannlösers"*, Dissertation, Stoßwellenlabor, RWTH Aachen, 1991

[3] Geist, A., Beguelin, A., Dongarra, J., Jiang,W., Manchek, R., Sunderam,V., *"PVM 3 User's Guide and Reference Manual"*, Oak Ridge National Laboratory, ORNL–Report TM–12187, Mai 1993

[4] Gupta R.N., Yos J.M., Thompson R.A., Lee K.-P., *"A Review of Reaction Rates and Thermodynamic and Transport Properties for an 11-Species Air Model for Chemical and Thermal Nonequilibrium Calculations to 30000 K"*, NASA Technical Memorandum C01528, 1989

[5] Haardt R., *"Erweiterung eines Godunov–Typ–Verfahrens zur Berechnung von Strömungen im chemischen und thermischen Nichtgleichgewicht"*, Studienarbeit, Lehr– und Forschungsgebiet für Mechanik, RWTH Aachen, 1993

[6] Haase, G., Hommel, T., Meyer, T., Pester, M., *"Bibliotheken zur Entwicklung paralleler Algorithmen"*, Preprint–Reihe der Chemnitzer DFG–Forschergruppe *Scientific Parallel Computing*, TU Chemnitz–Zwickau, SPC 94_4, März 1994

[7] Hirschfelder J.O., Curtiss Ch.F., Bird R.B., *"Molecular Theory of Gases and Liquids"*, John Wiley & Sons Inc., 4. Auflage, 1967

[8] Klomfaß A., *Unveröffentlichte Arbeitspapiere*

[9] Klomfaß A., Müller S., Ballmann J., *"Modellierung Hypersonischer Staupunktströmungen"*, Lehr– und Forschungsgebiet für Mechanik, Institut für Geometrie und Praktische Mathematik, RWTH Aachen, 1993

[10] Klomfaß A., Ballmann J., *"A Numerical Study on Hypersonic Non-Equilibrium Nitrogen Flow Modelling"*, Lehr– und Forschungsgebiet für Mechanik, RWTH Aachen, unveröffentlichte Arbeitspapiere (1993)

[11] Liu Y. , Vinokur M. , *"Nonequilibrium Flow Computations I. An Analysis of Numerical Formulations of Conservation Laws"*, J. Comp. Phys., Vol. 83, 1989, pp. 373 - 397

[12] Müller S., Dickopp Ch., Ballmann J., Jeltsch R., *"Computation of Viscous Hypersonic Nonequilibrium Blunt Body Flow"*, ZFW Bd. 17, 1993

[13] Müller S., *"Erweiterung von ENO–Verfahren auf zwei Raumdimensionen und Anwendung auf hypersonische Staupunktprobleme"*, Dissertation, RWTH Aachen 1993, Verlag Shaker, Aachen

[14] Neuwerth G., Staufenbiel R., *"Geometrie der Konfiguration ELAC-1"*, Interner Bericht Nr. 1, Teilprojekt A4, 1990

[15] Park C., *"On the Convergence of Computation of Chemically Reacting Flows"*, AIAA–Paper 085-0247, 1985

[16] Park C., *"Nonequilibrium Hypersonic Aerothermodynamics"*, John Wiley & Sons Inc., 1990