Thermodynamik-Schnittstelle

für

CAPE-Anwendungen

Projekt der IK-CAPE

Programmer's Guide

Version vom 09.10.98

Gliederung **Seite**

1 Einleitung

In diesem Handbuch sind neben den allgemein zugänglichen Schnittstellenprogrammen auch die Unterprogramme und Strukturen beschrieben, die bekannt sein müssen, um die Schnittstelle weiterzuentwickeln.

2 Die internen Speicherstrukturen

Alle Common-Blöcke der Schnittstelle werden über INCLUDE-Anweisungen in die Sources der Programme eingebunden.
Unter VMS wird über das Logical CAPECOMMON die Verknüpfung zum Directory hergestellt.

2.1 Die Speicherfelder

Als Speicherfelder stehen die Common-Bereiche /PROP_CHAR/ und /PROP_DATA/ zur Verfügung.

COMMON /PROP_DATA/
IFELD(100000)

Die Länge des Vektors IFELD wird im UP T_INIT gesetzt und über den Verwaltungs-Common zur Verfügung gestellt.

Um dieses Feld wahlweise als Integer-, double precision-, real-, logical- oder Character-Feld benutzen zu können, sind Felder dieser Typen überlagert.

EQUIVALENCE (IFELD(1), R8FELD(1), RFELD(1), C4FELD(1), LFELD(1))

Die Belegung dieser Speicherfelder richtet sich nach der Anzahl der eingegebenen Daten. Sie wird quasi dynamisch verwaltet. Der Begriff "Zeiger" bedeutet hierbei die Speicherposition innerhalb des Common-Feldes, je nach Definition in 4- oder 8-Byte Größen gerechnet (Zeiger ist also nicht als Pointer im Sinne von C zu verstehen).

Die Anzahl der Character des Feldes C4FELD wird über den Parameter NCCHAR gesetzt. Für 32-Bit-Rechner ist NCCHAR=4 zu setzen; für 64-Bit-Wort-Rechner ist NCCHAR=8 zu setzen.

Außerdem enthält der Include-File für diesen Common-Block den Parameter NSDWORD. Wird das Feld R8FELD als doppelt genaue Größe benutzt, ist NSDWORD=2 zu setzen; wird bei 64-Bit-Rechnern mit der Option doppelt genau = einfach genau kompiliert, ist NSDWORD=1 zu setzen. Dieser Parameter wird in den Unterprogrammen, die Zeigerrechnungen durchführen, benutzt.

COMMON /PROP_CHAR/
CLABEL, CFELD*5000

CLABEL ist der Labelname des aktuellen Stoffsystems.

Die Anzahl Character in CFELD wird im UP T_INIT gesetzt und über den Verwaltungs-Common zur Verfügung gestellt.

Hinweis:
Es hat sich gezeigt, daß bei einigen Compilern Sonderparameter angegeben werden müssen, um die Characterlänge von 5000 Byte verarbeiten zu können.

2.2 Allgemeine Verwaltungsgrößen

Allgemeine Verwaltungsgrößen für die Speicherbereiche der Schnittstelle befinden sich im Common /PROP_VERWALT/.

COMMON /PROP_VERWALT/
IPFELDB, NPFELDE, ICFELDB, NCFELDE, IPBL1,
NDAT_KONST, NZEIGER,
NSINGAKT, NSINGMAX, NSINGLAN,
NWPAKT, NWPMAX, NWPLAN,
NPARAKT, NPARMAX,
NMATAKT, NMATMAX,
NBIAKT, NBIMAX, NBILAN,
CHEMFEST, CHEMNUM, CHEMFUNC

IPFELDB	Anzahl bereits besetzter Worte im Daten-Common
NPFELDE	Maximale Anzahl zu speichernder Worte im Daten-Common
ICFELDB	Anzahl bereits besetzter Zeichen im Character-Common
NCFELDE	Maximale Anzahl zu speichernder Zeichen im Character-Common
IPBL1	Zeiger auf die Position des Steuerblockes des ersten gespeicherten Stoffsystems
NDAT_KONST	Anzahl Worte der Steuergrößen für Stoffkonstanten und Stoff-Funktionen
NZEIGER	Anzahl Zeiger im Common /PROP_ZEIGER/
NSINGAKT	aktuelle Anzahl belegter Stoffkonstantenzeiger
NSINGMAX	maximale Anzahl Stoffkonstantenzeiger
NSINGLAN	Anzahl Worte je Komponente für Stoffkonstanten (diese Größe ist abhängig von dem Parameter NSDWORD)
NWPAKT	aktuelle Anzahl belegter Zeiger für Stoff-Funktionen
NWPMAX	maximale Anzahl Zeiger für Stoff-Funktionen
NWPLAN	Anzahl Worte je Komponente für Stoff-Funktionen (ohne Koeffizienten)
NPARAKT	aktuelle Anzahl belegter Zeiger für Stoffdatenparameter
NPARAMAX	maximale Anzahl Zeiger für Stoffdatenparameter
NMATAKT	aktuelle Anzahl belegter Zeiger für Stoffdatenmatrizen
NMATMAX	maximale Anzahl Zeiger für Stoffdatenmatrizen
NBIAKT	aktuelle Anzahl belegter Zeiger für Stoff-Funktionen bei binären Daten
NBIMAX	maximale Anzahl Zeiger für Stoff-Funktionen bei binären Daten
NBILAN	Anzahl Worte je Komponente für binäre Stoff-Funktionen
CHEMFEST	Anzahl fester Größen je Chemieblock
CHEMNUM	Anzahl Größen je Reaktion in einem Chemieblock
CHEMFUNC	maximale Anzahl F(T)- bzw. PHI-Funktionen einer kinetisch kontrollierten Reaktion

Alle diese Größen werden im UP T_INIT initialisiert.

2.3 Allgemeine problemabhängige Verwaltungsgrößen

Allgemeine problemabhängige Größen zur Verwaltung sind im Common /PROP_ALLGEMEIN/ gespeichert.

COMMON /PROP_ALLGEMEIN/
NKOMPMAX, NZLANG, NZPROT, NZCASR, NREAMAX,
VERSION, NKOMP

allgemeine Größen:

NKOMPMAX	maximale Anzahl Komponenten in den verschiedenen Stoffsystemen
NZLANG	maximale Anzahl Zeichen der Komponentenlangnamen
NZPROT	maximale Anzahl Zeichen der Komponentennamen
NZCASR	maximale Anzahl Zeichen der Cas-Nummer
NREAMAX	maximale Anzahl Reaktionen in einem Chemieblock

Größen, die für das gerade aktive Stoffsystem gesetzt werden:

VERSION	Versionsnummer des gerade aktiven Systems
NKOMP	Komponentenzahl des gerade aktiven Systems

2.4 Zeiger auf gespeicherte Stoffdaten

Die Zeiger auf die verschiedenen Stoffdatentypen in den Arbeitsfeldern sind im Common /PROP_ZEIGER/ enthalten. Dieser wird beim Aktivieren eines Stoffdatenblockes aktuell mit dessen Zeigern gefüllt und steht dann den Berechnungsprogrammen zur Verfügung.

Dieser Common-Block gliedert sich z.Zt. in 7 Teilbereiche:

- Identifikationsdaten
- Stoffkonstanten
- Stoff-Funktionen
- Stoffdatenparameter

- Stoffmatrizen
- Berechnungslabel
- Stoff-Funktionen für binäre Daten
- Informationen für Inkrementmethoden

Für Identifikationsdaten gibt es folgende Zeiger, die in den Character-Common weisen:

Zeigername	Stoffdatentyp	Länge
INAMES	Komponentennamen	NKOMP*NZLANG-Zeichen
ICASNR	Cas-Nummer	NKOMP*NZCASR-Zeichen
IPROTN	Komponentenkurznamen	NKOMP*NZPROT-Zeichen

Für Stoffkonstanten gibt es folgende Zeiger, die alle auf 4-Byte-Größen im Daten-Common weisen:

Zeigername	Stoffdatentyp	Länge
IMOLW	Molekulargewicht	NKOMP*NSINGLAN+NDAT_KONST-Worte
ITC	krit. Temperatur	NKOMP*NSINGLAN+NDAT_KONST-Worte
IPC	krit. Druck	NKOMP*NSINGLAN+NDAT_KONST-Worte
IRHOC	krit. Dichte	NKOMP*NSINGLAN+NDAT_KONST-Worte
IAC	azentrischer Faktor	NKOMP*NSINGLAN+NDAT_KONST-Worte
IMP	Schmelzpunkt	NKOMP*NSINGLAN+NDAT_KONST-Worte
IMH	Schmelzwärme	NKOMP*NSINGLAN+NDAT_KONST-Worte
IVO25	spez. Volumen bei 25 Cels.	NKOMP*NSINGLAN+NDAT_KONST-Worte
IBETS	Bildungsenthalpie Feststoff	NKOMP*NSINGLAN+NDAT_KONST-Worte
IBETL	Bildungsenthalpie flüssig	NKOMP*NSINGLAN+NDAT_KONST-Worte
IBETG	Bildungsenthalpie Gas	NKOMP*NSINGLAN+NDAT_KONST-Worte
IFETS	freie Bildungsenthalpie Feststoff	NKOMP*NSINGLAN+NDAT_KONST-Worte
IFETL	freie Bildungsenthalpie flüssig	NKOMP*NSINGLAN+NDAT_KONST-Worte
IFETG	freie Bildungsenthalpie Gas	NKOMP*NSINGLAN+NDAT_KONST-Worte
IETRS	Entropie des Feststoffs	NKOMP*NSINGLAN+NDAT_KONST-Worte
IETRL	Entropie flüssig	NKOMP*NSINGLAN+NDAT_KONST-Worte
IETRG	Entropie Gas	NKOMP*NSINGLAN+NDAT_KONST-Worte
IPARC	Parachor	NKOMP*NSINGLAN+NDAT_KONST-Worte
IDIVO	Diffusionsvolumen	NKOMP*NSINGLAN+NDAT_KONST-Worte
ISINGL	6 freie Zeigerplätze	

Für Stoff-Funktionen gibt es folgende Zeiger, die alle auf 4-Byte-Größen im Daten-Common weisen:

Zeigername	Stoffdatentyp	Länge
ICS	spez. Wärme des Feststoffes	NKOMP*NWPLAN+NDAT_KONST-Worte
ICL	spez. Wärme der Flüssigkeit	NKOMP*NWPLAN+NDAT_KONST-Worte
ICPID	spez. Wärme des idealen Gases	NKOMP*NWPLAN+NDAT_KONST-Worte
IHSUB	Sublimationswärme	NKOMP*NWPLAN+NDAT_KONST-Worte
IHVAP	Verdampfungswärme	NKOMP*NWPLAN+NDAT_KONST-Worte
IDENS	Feststoffdichte	NKOMP*NWPLAN+NDAT_KONST-Worte
IDENL	Flüssigdichte	NKOMP*NWPLAN+NDAT_KONST-Worte
IVISL	Viskosität der Flüssigkeit	NKOMP*NWPLAN+NDAT_KONST-Worte
IVISV	Viskosität des Gases	NKOMP*NWPLAN+NDAT_KONST-Worte
IKLIQ	Wärmeleitfähigkeit der Flüssigkeit	NKOMP*NWPLAN+NDAT_KONST-Worte
IKVAP	Wärmeleitfähigkeit des Gases	NKOMP*NWPLAN+NDAT_KONST-Worte
IVPS	Sublimationsdruck	NKOMP*NWPLAN+NDAT_KONST-Worte
IVP	Dampfdruck	NKOMP*NWPLAN+NDAT_KONST-Worte
IST	Oberflächenspannung	NKOMP*NWPLAN+NDAT_KONST-Worte
IVIR	Virialkoeffizienten	NKOMP*NWPLAN+NDAT_KONST-Worte
IWEPA	10 freie Zeigerplätze	

Für Stoff-Parameter gibt es folgende Zeiger, die alle auf 8-Byte-Größen im Daten-Common weisen:

Zeigername	Stoffdatentyp	Länge
IUNIQ	UNIQUAC-Parameter	NKOMP*3 + 4-Doppelworte
IRKS	RKS-Parameter	NKOMP*3 + 4-Doppelworte
IPRABM	RR-Parameter	NKOMP*3 + 4-Doppelworte
IFLORY	Flory-Huggins-Parameter	NKOMP*3 + 4-Doppelworte
IDMER	Dimerisations-Parameter	NKOMP*2 + 4-Doppelworte
ITMER	Tetramerisations-Parameter	NKOMP*2 + 4-Doppelworte
IHMER	Hexamerisations-Parameter	NKOMP*2 + 4-Doppelworte
IPARAM	8 freie Zeigerplätze	

Für Stoff-Matrizen gibt es folgende Zeiger, die alle auf 8-Byte-Größen im Daten-Common weisen:

Zeigername	Stoffdatentyp	Länge
INRTL	NRTL-Matrizen L-V	NKOMP*NKOMP*3 + 4 Doppelworte
INRTLL	NRTL-Matrizen L-L	NKOMP*NKOMP*3 + 4 Doppelworte
IUNIQV	UNIQUAC-Matrix L-V	NKOMP*NKOMP + 4 Doppelworte
IUNIQL	UNIQUAC-Matrix L-L	NKOMP*NKOMP + 4 Doppelworte
ISRK	Redlich-Kwong-Soave-Matrix	NKOMP*NKOMP*2 + 4 Doppelworte
IWILS	Wilson-Matrizen	NKOMP*NKOMP*2 + 4 Doppelworte
IHNRY	Henry-Matrizen	NKOMP*NKOMP*4 + 4 Doppelworte
IPR	Peng-Robinson-Matrix	NKOMP*NKOMP + 4 Doppelworte
IMATR	18 freie Zeigerplätze	

Für Berechnungslabel gibt es folgende Zeiger, die alle auf 4-Byte-Größen im Daten-Common weisen:

Zeigername	Stoffdatentyp	Länge je Block
IVLEQ	L-V-Gleichgewicht	VL-MAXMET + 2 Worte je Label
ILLEQ	L-L-Gleichgewicht	LL-MAXMET + 2 Worte je Label
ICHEMIE	Reaktionen	CHEMFEST + Anz.Label*Anz.Reakt.*CHEMNUM-Worte
IENTHAL	Enthalpie	8 Worte je Label
IENTROP	Entropie	kommt noch
IKPHAS	Phasenschlüssel	NKOMP*4+4 Worte
IDENV	Methode für Gasdichte	3+1 Wort
IMETH	3 freie Plätze	

Für Binärdaten, die elementweise über Stoff-Funktionen gespeichert werden, gibt es folgende Zeiger:

Zeigername	Stoffdatentyp	Länge je Block
ICROSS	Kreuzvirialkoeffizienten	(NKOMP*(NKOMP1))/2*NBILAN+NDAT_KONST -Worte
IHMIXA	Mischungswärme; Temperaturabhängigkeit A	(NKOMP*(NKOMP1))/2*NBILAN+NDAT_KONST -Worte
IHMIXB	Mischungswärme; Temperaturabhängigkeit B	(NKOMP*(NKOMP1))/2*NBILAN+NDAT_KONST -Worte
IHMIXC	Mischungswärme; Temperaturabhängigkeit C	(NKOMP*(NKOMP1))/2*NBILAN+NDAT_KONST -Worte
IHMIXD	Mischungswärme; Temperaturabhängigkeit D	(NKOMP*(NKOMP1))/2*NBILAN+NDAT_KONST -Worte
IHMIXE	Mischungswärme; Temperaturabhängigkeit E	(NKOMP*(NKOMP1))/2*NBILAN+NDAT_KONST -Worte
IHMIXF	Mischungswärme; Temperaturabhängigkeit F	(NKOMP*(NKOMP1))/2*NBILAN+NDAT_KONST -Worte
IBINA	18 freie Speicherplätze	

Für Inkrementinformationen gibt es folgende Zeiger, die alle auf 4-Byte-Größen im Daten-Common weisen:

Zeigername	Stoffdatentyp	Länge je Block
IUNIF_KO	Komponentenabhängige Inkremntinformationen	NKOMP*3 + 8 Worte
IUNIF_GR	Inkrement- und komponentenabhängige Informationen	MAXGRP*NKOMP*5 + 8 Worte
IUNIF_WP		MAXGRP * 3 + 8 Worte
IUNIF_MAT		Anz. Gruppen*Anz. Gruppen *2 + 8 Worte

2.5 Aktuelle Berechnungsmethoden

Im Common /PROP_THERMO/ sind die Berechnungsmethoden gespeichert, nach denen die Programme zur Berechnung von Phasengleichgewichten, Enthalpien etc. aktuell arbeiten.
Gespeichert werden diese Daten mit einem entsprechenden T_S_...-Programm.

COMMON /PROP_THERMO/
LVEQ_LABEL, LLEQ_LABEL, CHEM_LABEL, ENTH_LABEL, ENTR_LABEL

2.6 Arbeitsfelder für die allgemeinen Berechnungsprogramme

Der Arbeitsspeicher für die allgemeinen Berechnungsprogramme wird über Zeiger im Common **/PROP_CALC_ZEIGER/** verwaltet. Diese werden im UP T_ZEIGER aktiviert. Für die Berechnung von Reinstoff-Funktionen (T_PURE), Mittelwerten (T_AVER), k-Werten (T_LVEQ, T_GAMMA, T_POYNT, T_FUGA, T_HENRY, T_ZUST), L-L-Gleichgewichten (T_LLEQ, T_LL_GAMMA), Kompressibilitäten (T_COMPR), Enthalpien (T_ENTH, T_H_EX, T_H_ISO) und chemischen Reaktionen (T_CHEM, T_CH) sind bereits Zeiger vorgesehen.
In Abhängigkeit von ihrer Funktion ist ihre Länge im UP T_ZEIGER als Vektor oder Matrix über die maximale Komponentenzahl gesetzt.

2.7 Arbeitsfelder für Methodenberechnungsprogramme

Der Common /PROP_WORK/ enthält Zeiger, die auf allgemein als Hilfsfelder benutzbare Speicherbereiche im Speicherfeld des Common /PROP_DATA/ weisen (z.B. für Zwischenspeicher in den Berechnungs-unterprogrammen). IVEKTOR, IMATRIX und IGRPKOMP weisen ins R8FELD, INTVEKT in IFELD bzw. C4FELD.

COMMON /PROP_WORK/
IVEKTOR, IMATRIX, INTVEKT, IGRPKOMP

Im Unterprogramm T_ZEIGER werden diese Positionen in folgender Größe angelegt:

IVEKTOR	14 * max. Komponentenzahl
IMATRIX	5 * max. Komponentenzahl * max. Komponentenzahl
INTVEKT	3 * max. Komponentenzahl
IGRPKOMP	max. Gruppen * max. Komponenten

2.8 Parametergrößen der Include-Datei

Die Datei PROP_INCLUDE.FOR enthält Parametergrößen, die allgemeine Gültigkeit haben.

Es sind folgende Größen:

Parameter	Wert	Bedeutung
RJOULE	8314.3	Gaskonstante in J/(Kmol*Kelv)
MAXEXP	85	maximaler Exponent der e-Funktion unter VMS
ZHALB	5	z-Parameter für UNIQUAC (= 0.5 * KZ)
NCOEFF	10	Dimensionierung von Hilfsfeldern zur Speicherung von Koeffizienten
VL_MAXMET	5	Anzahl Felder des Methodenvektors für Flüssig-Dampf-Gleichgewichte
LL_MAXMET	1	Anzahl Felder des Methodenvektors für Flüssig-flüssig-Gleichgewichte
NONVAL	-1.D30	Kennzeichnung für nicht besetzte Größen (z.B. nicht berechnete Ableitungen)
EPS_GAUSS	1.D-6	Epsilon für Abweichungen des Funktionswertes bei Gauss
DIFAK_T	1.D-3	relativer Fehler zur Berechnung von Differenzenquotienten (Temperatur)
DIFAK_P	1.D-3	relativer Fehler zur Berechnung von Differenzenquotienten (Druck)
DIFAK_Y	1.D-6	relativer Fehler zur Berechnung von Differenzenquotienten (y-Anteile)
MAXGRP	25	maximale Anzahl Inkremente je Komponente bzw. im Gemisch

3 Die Unterprogramme zum Laden der Datei und Auffüllen von Stoffsystemen

- T_LOAD
  - T_INIT
  - T_P_COMP_NUMBER
  - T_P_NAME_LENGTH
  - T_P_BLOCK_NAME
  - T_ASK_TYPE
  - T_P_IDEN_COMP
  - T_P_SINGLE_COMP
  - T_P_PURE_COEFF
  - T_P_PARAM
  - T_P_MATRIX
  - T_P_AVER
  - T_P_LVEQ
  - T_P_LLEQ
  - T_P_LABEL_ENTH
  - T_P_EXCE_ENTH
  - T_P_ISOT_ENTH
  - T_P_PHASE_ENTH
  - T_P_HREF_ENTH
  - T_P_TREF_ENTH
  - T_P_TPHAS_ENTH
  - T_P_BINA_COEFF
  - T_P_CH_LABEL
  - T_P_CH_TYPE
  - T_P_CH_CONV
  - T_P_CH_ZETA
  - T_P_CH_DHR
  - T_P_CH_STOE
  - T_P_CH_FVT
  - T_P_CH_KI_FVT
  - T_P_CH_PHI
  - T_P_INKR
  - T_P_MIXT_INKR
  - T_FI_SYSTEM
    - T_FI_UNIQUAC
    - T_FI_RKS
    - T_FI_SRK
    - T_FI_PR
    - T_FI_ENTH
  - T_P_SYSTEM
  - T_ZEIGER
  - T_G_SYSTEM_NAME
  - T_S_SYSTEM

Die Parameterliste der FILL-Unterprogramme ist wie folgt definiert:

NKOMP, FILL-Stoffdaten, RVAL

Name	Typ	Zugriff
NKOMP	I4	Lesen
FILL-Stoffdaten	R8	Lesen/Schreiben
RVAL	I4	Fehlerkennung

ARGUMENTE

NKOMP

aktuelle Komponentenzahl

FILL-Stoffdaten

Felder der Stoffdaten, deren Hilfsgrößen aufgefüllt werden sollen.

RVAL

Fehlerkennung

= 0 : alles ok.

= >0 : Fehler

T_LOAD

Laden der neutralen Stoffdatenfiles

FORMAT

T_LOAD(CHANNEL, RVAL)

Name	Typ	Zugriff
CHANNEL	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

CHANNEL

Kanalnummer, unter der die Datei geöffnet ist

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Speicherplatz erschöpft, Datei nicht komplett eingelesen

= 10: Reihenfolge der Steuersätze nicht ok.

= 20: Stoffsystemreihenfolge nicht möglich

= 30: nicht identifizierbarer Datentyp

= 40: maximale Inkrementzahl überschritten

< 0 : Dateilesefehler, RVAL = negativem IOSTAT-Wert

FUNKTION

Das Stoffdaten-File muß bei Aufruf des Unterprogramms T_LOAD bereits geöffnet sein und wird von diesem auch nicht geschlossen.

Alle Daten der Übertragungsdatei werden mit Fortran-Readbefehlen eingelesen und durch Unterprogramm-Aufrufe in die internen Strukturen für die Schnittstellenprogramme gespeichert.

Das UP T_ASK_TYPE identifiziert die Satzkennung und steuert so die Speicherung.

Das zuerst eingelesene Stoffsystem wird automatisch als aktives Stoffsystem gespeichert, so daß es den Berechnungsprogrammen ohne explizite Aktivierung zur Verfügung steht.

T_FI_SYSTEM

Auffüllen von Hilfsgrößen, die von den Zustandsgrößen der Berechnung unabhängig sind

FORMAT

T_FI_SYSTEM(RVAL)

Name	Typ	Zugriff
RVAL	I4	Schreiben

ARGUMENTE

RVAL

Fehlerkennung

= 0 : alles ok.

> 0 : Fehler aus Unterprogrammen

FUNKTION

Das Unterprogramm T_FI_SYSTEM steuert die Berechnung von Hilfsvektoren und Hilfsmatrizen, die von den Zustandsgrößen während der Berechnung unabhängig sind.

Zur Zeit sind implementiert:

- die Berechnung des Vektors l_i der Uniquac-Gleichung.
- die Berechnung des Vektors a_i b_i m_i für Peng-Robinson
- die Berechnung des Vektors m_i für Redlich-Kwong-Soave
- die Berechnung der Gemischmatrix für Redlich-Kong-Soave ohne Berücksichtigung der Temperaturabhängigkeit
- die Berechnung der konstanten Terme der Reinstoffenthalpie bei vorgegebener Umwandlungstemperatur.

Der Aufruf dieses Programmes erfolgt im UP T_LOAD, bevor das Stoffsystem mit T_P_SYSTEM gespeichert wird.

4 Unterprogramme zum Speichern und Holen von Daten

4.1 Allgemeine Unterprogramme

- T_P_POINTER
- T_INIT
- T_ZEIGER

T_P_POINTER

Setzen von Zeigern; Platzverwaltung für die Common-Blöcke /PROP_DATA/ und /PROP_CHAR/

FORMAT

T_P_POINTER(KENNUNG, ANZAHL, LENGTH, ZEIGER, RVAL)

Name	Typ	Zugriff
KENNUNG	C1	Lesen
ANZAHL	I4	Lesen
LENGTH	I4	Lesen
ZEIGER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

KENNUNG

Kennung, in welchem Common-Block Speicherplatz belegt werden soll.

= 'C' : Character-Common

= 'D' : Daten-Common

ANZAHL

Anzahl zu reservierender Größen

LENGTH

Anzahl Bytes je zu belegender Größe

gültige Längenangaben sind:

1 : für Größen im Character-Common

4 : für einfach genaue Größen im Daten-Common

8 : für doppelt genaue Größen im Daten-Common

ZEIGER

Anfangsposition des belegten Speicherplatzes; bestimmt entsprechend der Angabe in LENGTH und dem Parameter NSDWORD

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

=-1 : fehlerhafte Kennung

=-2 : fehlerhafte Länge

=-3 : fehlerhafte Anzahl

FUNKTION

T_P_POINTER reserviert Speicherplatz entweder im Character-Common /PROP_CHAR/ (Kennung = 'C') oder im Daten-Common /PROP_DATA/ (Kennung = 'D').

Die Verwaltungsgrößen im Common /PROP_VERWALT/ NCFELDB für den Character-Common und NPFELDB für den Daten-Common werden entsprechend dem nun verbrauchten Platz neu gesetzt.

Die belegten Speicherplätze werden bei Charactergrößen mit Blank, bei Datengrößen mit 0 initialisiert.

Ist der Parameter NSDWORD auf 1 gesetzt, wird bei LENGTH = 8 die Länge in Worten gerechnet.

T_INIT

Initialisieren der internen Speicherbereiche der Thermodynamik-Schnittstelle

FORMAT T_INIT

FUNKTION

Die Common-Bereiche

- PROP_VERWALT
- PROP_ALLGEMEIN
- PROP_ZEIGER
- PROP_THERMO
- PROP_CHAR

werden initialisiert.

Im besonderen werden in diesem Unterprogramm die Feldlängen für die Common-Buffer /PROP_DATA/ (NPFELDE) und /PROP_CHAR/ (NCFELDE) gesetzt.

T_ZEIGER

Aktivieren von Hilfsspeicher für die Berechnungsprogramme der Thermodynamik-Schnittstelle

FORMAT

T_ZEIGER(RVAL)

Name	Typ	Zugriff
RVAL	I4	Schreiben

ARGUMENTE

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

FUNKTION

Es werden die Zeiger des Common /PROP_CALC_ZEIGER/ aktiviert, die den Berechnungsprogrammen zur Aufnahme von Ergebnissen und Ableitungen dienen.

Außerdem wird der Speicherplatz für die Hilfsfelder der methodenspezifischen Berechnungsprogramme im Common /PROP_WORK/ aktiviert.

4.2 Verwaltung von Dimensions- und Identifikationsdaten

Die Verwaltung der Dimensionierungsdaten des Steuersatzes und der Systemsteuersätze sowie der Identifikationsdaten der Komponenten erfolgt in folgenden Programmen:

- T_P_COMP_NUMBER
- T_P_NAME_LENGTH
- T_P_IDEN_COMP
  - T_P_POINTER
- T_G_COMP_NUMBER
- T_G_NAME_LENGTH
- T_G_IDEN_COMP

T_P_COMP_NUMBER

Speichern der maximalen bzw. der aktuellen Komponentenzahl

FORMAT

T_P_COMP_NUMBER(TASK, NUMBER, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
NUMBER	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

= 1 : speichern der maximalen Komponentenzahl

= 2 : speichern der aktuellen Komponentenzahl

NUMBER

Komponentenzahl

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : maximale Komponentenzahl verletzt den Gültigkeitsbereich

= 2 : aktuelle Komponentenzahl verletzt den Gültigkeitsbereich

=-1 : ungültige TASK

FUNKTION

Die Daten werden im Common /PROP_ALLGEMEIN/ abgelegt.

Der Gültigkeitsbereich der maximalen Komponentenzahl : 0 < NUMBER < 101

Der Gültigkeitsbereich der aktuellen Komponentenzahl : 0 < NUMBER < maximale Komponentenzahl

T_P_NAME_LENGTH

Speichern der Anzahl Zeichen der Komponentenidentifikationsdaten

FORMAT

T_P_NAME_LENGTH(TASK, NUMBER, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
NUMBER	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

= 1 : Speichern der Anzahl Zeichen des Namen

= 2 : Speichern der Anzahl Zeichen des Langnamens

= 3 : Speichern der Anzahl Zeichen der Cas-Nummer

NUMBER

Anzahl Zeichen

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Zeichenzahl ungültig

=-1 : ungültige TASK

FUNKTION

Die Daten werden in den Common /PROP_ALLGEMEIN/ gespeichert.

T_P_IDEN_COMP

Speichern von Komponentenidentifikationsdaten

FORMAT

T_P_IDEN_COMP(TASK, COMP_NR, COMP_ID, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
COMP_NR	I4	Lesen
COMP_ID	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

= 1 : Speichern des Namen

= 2 : Speichern des Langnamens

= 3 : Speichern der Cas-Nummer

COMP_NR

Komponentennummer

COMP_ID

zu speichernde Identifikationsdate

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

= 2 : Datenlänge nicht gesetzt

= 3 : zu speichernde Date ist Blank

= 4 : Länge des übergebenen Strings > zu speichernde Anzahl Zeichen

=-1 : ungültige TASK

=-2 : Komponentennummer nicht gültig

FUNKTION

Die Daten werden in den Common /PROP_CHAR/ gespeichert.

Die Zeiger im Common /PROP_ZEIGER/

IPROTN für die Komponentennamen

INAMES für die Komponentenlangnamen

ICASNR für die CAS-Nummern

werden, falls noch keine Date gespeichert ist, mit dem UP T_P_POINTER gesetzt.

Die Anzahl der reservierten Zeichen je Datentyp ist:

Anzahl Komponenten * Anzahl Zeichen

Hierbei bedeutet "Anzahl Zeichen" NZLANG oder NZCASR oder NZPROT.

Der Zeiger weist hierbei auf das erste gespeicherte Zeichen der ersten Komponente.

Ist die Länge des übergebenen Characterstring > als die Anzahl Zeichen wird RVAL = 4 gesetzt und der Characterstring beim Speichern abgeschnitten.

Bei kürzerer Länge wird mit Blank aufgefüllt.

T_G_COMP_NUMBER

Holen der maximalen bzw. aktuellen Komponentenzahl

FORMAT

T_G_COMP_NUMBER(TASK, NUMBER, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
NUMBER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1: holen der maximalen Komponentenzahl

=2: holen der aktuellen Komponentenzahl

NUMBER

gewünschte Komponentenzahl

RVAL

Fehlerkennung

= 0: alles ok.

= 1: maximale Komponentenzahl nicht gesetzt

= 2: aktuelle Komponentenzahl nicht gesetzt, d.h. es ist kein Stoffdatenblock aktiv

=-1: ungültige TASK

FUNKTION

Die im Common /PROP_ALLGEMEIN/ gespeicherten Daten werden zurückgegeben.

T_G_NAME_LENGTH

Holen der Anzahl Zeichen der Komponentenidentifikationsdaten

FORMAT

T_G_NAME_LENGTH(TASK, NUMBER, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
NUMBER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1: Holen der Anzahl Zeichen des Namens

=2: Holen der Anzahl Zeichen des Langnamens

=3: Holen der Anzahl Zeichen der Cas-Nummer

NUMBER

gewünschte Zeichenzahl

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Zeichenzahl = 0

=-1: ungültige TASK

FUNKTION

Die Zeichenzahl der gewünschten Identifikationsdate wird aus dem Common /PROP_ALLGEMEIN/ zurückgegeben.

T_G_IDEN_COMP

Holen der Stoffidentifikationsdaten
- Name
- Langname
- Cas-Nummer

FORMAT

T_G_IDEN_COMP(TASK, COMP_NR, COMP_ID, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
COMP_NR	I4	Lesen
COMP_ID	C(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

= 1: Holen des Namens

= 2: Holen des Langnamens

= 3: Holen der Cas-Nummer

COMP_NR

Komponentennummer

> 0: Nummer der gewünschten Komponente

= 0: alle Komponenten

COMP_ID

je nach TASK gefüllte Ausgabegröße

RVAL

Fehlerkennung

= 0: alles ok

= 1: Daten nicht gespeichert

= 2: ungültige Komponentennummer

= 4: Länge des Ausgabefeldes zu klein

=-1: ungültige TASK

FUNKTION

Für den Fall COMP_NR = 0 werden die gewünschten Identifikationsdaten für alle Komponenten im Vektor COMP_ID zurückgegeben. Die Dimension von COMP_ID muß also mindestens der aktuellen Komponentenzahl entsprechen. Für den Fall COMP_NR > 0 wird eine einzelne Date zurückgegeben.

Die Identifikationsdaten sind im Common /PROP_CHAR/ hintereinander gespeichert.

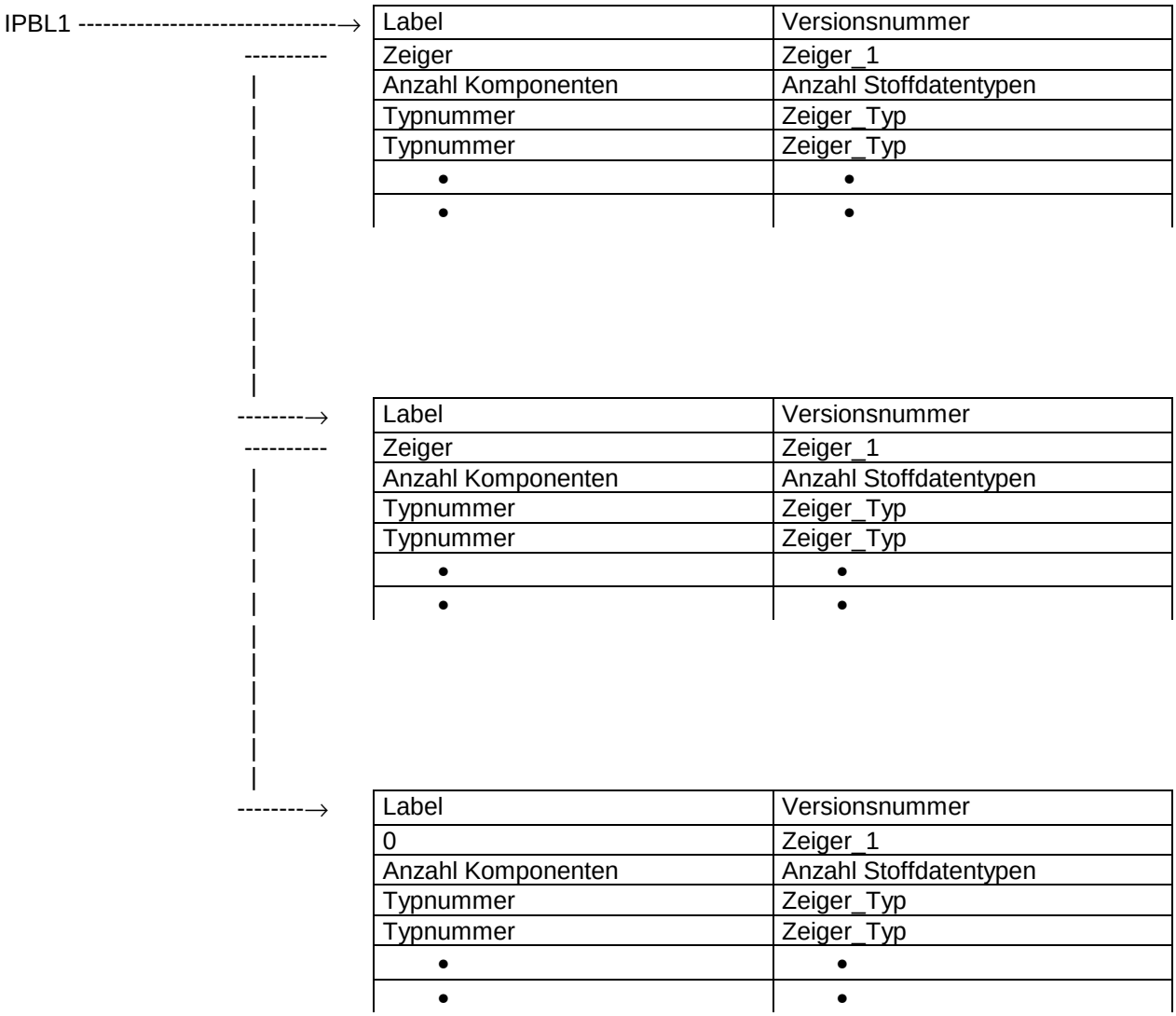
Die Positionsangaben

- IPROTN für die Namen
- INAMES für die Langnamen
- ICASNR für die Cas-Nummern

befinden sich im Common /PROP_ZEIGER/.

4.3 Stoffsystemverwaltung

Die Stoffsysteme werden in verketteten Steuerblöcken verwaltet. Anfangszeiger ist die Größe IPBL1 im Common /PROP_VERWALT/. Dieser Zeiger weist auf die 1. gespeicherte 4-Byte-Größe des ersten Steuerblocks im Common /PROP_DATA/. Der Aufbau der Steuerblöcke sieht wie folgt aus:



Jeder Steuerblock enthält alle beschreibenden und inhaltlichen Informationen, die für eine eindeutige Zuordnung notwendig sind:

- Label
- Versionsnummer
- Anzahl Komponenten
- Anzahl Stoffdatentypen

Da die Stoffdaten im Common /PROP_DATA/ permanent vorhanden sind, werden hier nur die Zeigerpositionen gespeichert. Die Typnummer ist gleich der Zeigernummer des entsprechenden Stoffdatentyps im Common /PROP_ZEIGER/. Zeiger_Typ ist die Speicherposition des Stoffdatentyps im Common /PROP_DATA/. Es werden hierbei nur die Typen gespeichert, für die Daten vorhanden sind. Die Größe Zeiger_1 ist für alle Basissysteme (Version 1) = 0. Bei höheren Versionen weist Zeiger_1 auf denjenigen Steuerblock, der die Daten des zugehörenden Basissystems enthält. Anzahl Stoffdatentypen enthält in diesem Fall nur die Anzahl der für die neue Version geänderten bzw. neuen Datentypen.

Für die Verwaltung (Speichern, Holen, Aktivieren) von Stoffsystemen sind folgende Unterprogramme vorhanden:

- T_P_SYSTEM_NAME
- T_P_SYSTEM
  - T_FIND_SYSTEM
  - T_P_POINTER
- T_G_SYSTEM_NAME
- T_S_SYSTEM
  - T_FIND_SYSTEM
- T_S_VERSION
  - T_FIND_SYSTEM

Nachfolgend werden die Unterprogramme detaillierter beschrieben (die Beschreibung von T_P_POINTER erfolgt im Kapitel "Allgemeine Unterprogramme").

T_P_BLOCK_NAME

Speichern des Labels und der Versionsnummer eines neu anzulegenden Stoffsystems als aktuelles System

FORMAT

T_P_BLOCK_NAME(LABEL, VERSION, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
VERSION	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Name des Stoffsystems

VERSION

Versionsnummer des Stoffsystems

RVAL

Fehlerkennung

= 0: alles ok.

FUNKTION

Die Eingabedaten werden den entsprechenden Größen in den Common-Blöcken /PROP_CHAR/ und /PROP_ALLGEMEIN/ zugewiesen.

Diese Zuweisung ist notwendig, um allen anderen Speicherprogrammen mitzuteilen, daß sie mit dem aktuellen Stoffsystem arbeiten.

Für Version = 1 werden zusätzlich alle Zeiger-Größen des Common /PROP_ZEIGER/ auf -1, die aktuelle Komponentenzahl auf 0 und alle Berechnungslabel auf Blank gesetzt.

T_P_SYSTEM

Speichern der Steuerdaten für ein Stoffsystem

FORMAT

T_P_SYSTEM(LABEL, VERSION, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
VERSION	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Name des Stoffsystems

VERSION

Versionsnummer des Stoffsystems

RVAL

Fehlerkennung

= 0: alles ok.

= 1: kein Speicherplatz zur Speicherung des Steuerblockes

= 2: Versionsnummer > 1 und Basissystem nicht gefunden

FUNKTION

Bei der Speicherung der Stoffsysteme müssen im wesentlichen 3 Fälle unterschieden werden:

- es ist noch kein Steuerblock gespeichert
- der zu speichernde Block war bereits vorhanden, d.h. ändern der Daten
- der Block ist neu, es sind aber noch andere Blöcke vorhanden, d.h. anhängen

Für alle 3 Fälle gilt, daß gezählt wird, wieviel Stoffdatentypen gespeichert werden müssen. Diese Anzahl bestimmt die Größe des Steuerblocks:

6 beschreibende Größen + 2*Anzahl Stoffdatentypen

Die Anzahl der Stoffdatentypen ergibt sich aus den Daten im Common /PROP_ZEIGER/ wie folgt:

- bei Basissystemen: Anzahl der Stoffdatenzeiger > 0
- bei höheren Versionen: Anzahl der Stoffdatenzeiger > 0, aber ungleich dem Basissystem

War der Block bereits vorhanden, wird geprüft, ob der vorhandene Platz für den Steuerblock ausreicht; wenn nicht, wird auch hier der Speicherplatz neu reserviert und der Block durch Neuverkettung der anderen Blöcke aus der Kette herausgenommen. Dadurch wird der Zustand eines ganz neuen Blockes erreicht.

Nach dem Einspeichern der allgemeinen Blockdaten und Stoffdatenzeiger werden die Zeiger eingespeichert:

Verkettungszeiger des neuen Blockes = 0 setzen

Verkettungszeiger im vorhergehenden Block auf Blockanfang positionieren

Für das erste System wird noch der Startzeiger gesetzt.

Handelt es sich um eine höhere Versionsnummer, wird die Verkettung auf das Basissystem eingetragen.

Wurde ein Basissystem, zu dem höhere Versionen vorhanden sind, geändert und mußte neu gespeichert werden, muß außerdem die Verkettung der höheren Versionen zu ihrem Basissystem aktualisiert werden.

T_FIND_SYSTEM

Finden des Steuerblockes von Stoffsystemen

FORMAT

T_FIND_SYSTEM(TASK, LABEL, VERS, ZEIGER, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen
VERS	I4	Lesen
ZEIGER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

= 1: das angegebene System suchen

= 2: das letzte gespeicherte System suchen

= 3: das Vorgängersystem des angegebenen Systems suchen

LABEL

Name des Stoffsystems

Eingabe notwendig bei TASK = 1, 3

VERSION

Versionsnummer des Stoffsystems

ZEIGER

Position des Steuerblockes

RVAL

Fehlerkennung

= 0: alles ok.

= 1: nicht gefunden

FUNKTION

Ausgehend vom Startzeiger IPBL1 im Common /PROP_VERWALT/ wird je nach TASK die Position des gewünschten Stoffsystems gesucht.

Im Fall TASK = 1 wird der Steuerblock für genau dieses System und diese Versionsnummer gesucht.

Für den Fall TASK = 2 wird der Steuerblock gesucht, dessen Verkettungsfeld = 0 ist.

Für den Fall TASK = 3 wird in Zeiger die Position des Steuerblockes zurückgegeben, dessen Verkettungsfeld auf den angebenen Stoffdatenblock zeigt.

T_G_SYSTEM_NAME

Namen und Versionsnummer von Stoffsystemen holen

FORMAT

T_G_SYSTEM_NAME(TASK, LABEL, VERSION, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Schreiben
VERSION	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

= 1 : Name des gerade aktiven Stoffsystems

= 2 : Name des ersten gespeicherten Stoffsystems

= 3 : Name des nächsten Stoffsystems

LABEL

Name des Stoffsystems

VERSION

Versionsnummer des Stoffsystems

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : bei TASK=3: kein weiteres System gespeichert

bei TASK=2: noch kein Stoffsystem gespeichert

= 2 : bei TASK=3: noch kein Aufruf mit TASK=2

=-1 : ungültige TASK

FUNKTION

Der Label und die Versionsnummer des aktuellen Stoffsystems sind im Common /PROP_CHAR/ bzw. /PROP_ALLGEMEIN/ gespeichert.

Die Steuerblöcke für die gespeicherten Stoffsysteme sind im Common /PROP_DATA/ gespeichert; auf den ersten Block weist der Zeiger IPBL1 im Common /PROP_VERWALT/.

T_S_SYSTEM

Aktivieren von Stoffsystemen

FORMAT

T_S_SYSTEM(LABEL, VERSION, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
VERSION	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Name des zu aktivierenden Stoffsystems

VERSION

Versionsnummer des zu aktivierenden Stoffsystems

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Stoffsystem nicht gefunden

FUNKTION

Wird T_S_SYSTEM mit dem Namen und der Versionsnummer des bereits aktiven Stoffsystems aufgerufen, braucht nicht mehr aktiviert zu werden.

Ansonsten wird wie folgt gearbeitet:

1. Mit dem UP T_FIND_SYSTEM wird gesucht, ob das gewünschte Stoffsystem gespeichert ist.
2. Der Common /PROP_ZEIGER/ wird in den Anfangszustand versetzt, d.h. alle Zeiger = -1.
3. Ist die Versionsnummer > 1 müssen zuerst die Zeiger des Basissystems umgespeichert werden (der Verkettungszeiger = 4. Größe des Steuerblocks weist auf den Steuerblock des Basissystems).
4. Umspeichern der Zeiger des aktuellen Systems.
5. Aktualisieren der aktuellen Komponentenzahl, des aktuellen Labels und der aktuellen Version.
6. Zurücksetzen der Berechnungslabel im Common /PROP_THERMO/

T_S_VERSION

Aktivieren der Daten, die ausschliesslich zu einem Stoffsystem mit Versionsnummer > 1 gehören

FORMAT

T_S_VERSION(LABEL, VERS, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
VERS	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Name des zu aktivierenden Stoffsystems

VERS

Versionsnummer des zu aktivierenden Stoffsystems

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Stoffsystem nicht gefunden

=-1: Versionsnummer > 1

FUNKTION

Mit T_S_VERSION werden nur die Daten aktiviert, die zu diesem Stoffsystem mit Versionsnummer > 1 gehören. Die T_G-Programme der Schnittstelle können dann diese Daten liefern.

Für die Berechnung ist es notwendig nach einem Aufruf von T_S_VERSION durch einen Aufruf von T_S_SYSTEM das Gesamtsystem wieder zu aktivieren.

Es wird wie folgt gearbeitet:

1. Mit dem UP T_FIND_SYSTEM wird gesucht, ob das gewünschte Stoffsystem gespeichert ist.
2. Der Common /PROP_ZEIGER/ wird in den Anfangszustand versetzt, d.h. alle Zeiger = -1.
3. Umspeichern der Zeiger des aktuellen Systems.
4. Aktualisieren der aktuellen Komponentenzahl; der aktuellen Label wird ‘ ‘,die aktuelle Version = 0 gesetzt.
5. Zurücksetzen der Berechnungslabel im Common /PROP_THERMO/

4.4 Stoffkonstanten

Für jeden Typ von Stoffkonstanten gibt es einen Zeiger im Common /PROP_ZEIGER/.
Dieser Zeiger weist auf 4-Byte-Größen im Common /PROP_DATA/.
Vorgelagert sind jedem Datentyp 4 Steuergrößen mit folgender Belegung:

C4	I4	C4	I4
Kürzel des Stofftyps	Anzahl gespeicherter Daten in Worten	Typ der Mittelwertbildung	Versionskennung = 0: Basisversion = 1: höhere Version

Die Daten sind als Block der Länge Anzahl Komponenten * NSINGLAN in folgender Form gespeichert:

	I4	C4	R8
Zeiger ----------→	Speicherkennung	Quelle	Stoffkonstante 1. Komponente
	Speicherkennung	Quelle	Stoffkonstante 2. Komponente
	•	•	•
	•	•	•

Die Speicherkennung kann folgende Werte annehmen:
= 1 : Wert ist gespeichert
= 0 : Wert ist nicht gespeichert, wird aber gebraucht
=-1 : Wert wird nicht gebraucht.

Für die Verwaltung von Stoffkonstanten sind folgende Unterprogramme vorhanden:

- T_P_SINGLE_COMP
  - T_ASK_SINGLE
  - T_ACT_SINGLE
    - T_P_POINTER
- T_G_SINGLE_COMP
  - T_ASK_SINGLE

T_P_SINGLE_COMP

Speichern von Stoffkonstanten

FORMAT

T_P_SINGLE_COMP(PROPTYP, COMP_NR, SOURCE, VALUE, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
SOURCE	C	Lesen
VALUE	R8	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffkonstantenkennung

COMP_NR

Komponentennummer

SOURCE

Quelleninformation

VALUE

Stoffkonstante in internen Einheiten

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

=-1 : ungültiger Stoffdatentyp

=-2 : ungültige Komponentennummer

FUNKTION

T_P_SINGLE_COMP ruft das Unterprogramm T_ACT_SINGLE zur eventuellen Aktivierung des für den Stoffdatentyp gültigen Zeigers.

Danach speichert es die Daten in der allgemeinen Form. Der Zeiger wird über das Unterprogramm T_ASK_SINGLE mit seiner Nummer identifiziert.

T_ASK_SINGLE

Stofftypkennung für Stoffkonstanten prüfen

FORMAT

T_ASK_SINGLE(PROPTYP, NUMMER, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

NUMMER

Nummer des Stoffdatentyps

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : angegebener Stofftyp ist keine Stoffkonstante

FUNKTION

T_ASK_SINGLE prüft den angegebenen Stoffdatentyp auf Gültigkeit.

Die Anzahl und Reihenfolge der in T_ASK_SINGLE initialisierten Stoffkonstantentypen muß mit der Anzahl und der Reihenfolge der Zeiger im Common /PROP_ZEIGER/ übereinstimmen.

Der Parameter NSINGLE gibt die Anzahl an; dieser Parameter entspricht der Common-Größe NSINGAKT im Common /PROP_VERWALT/.

T_ACT_SINGLE

Aktivieren von Zeigern für Stoffkonstanten

FORMAT

T_ACT_SINGLE(PROPTYP, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : angegebener Stofftyp ist keine Stoffkonstante

= 1 : Zeiger konnte nicht gesetzt werden

FUNKTION

T_ACT_SINGLE unterscheidet 2 Fälle, für die der benötigte Zeiger angelegt werden muß:

1. der Zeiger ist noch nicht aktiviert
2. bei einem Stoffsystem höherer Version ist der Zeiger für die Basisversion aktiviert, für die höhere Version aber noch kein eigener Zeiger angelegt.

Im ersten Fall wird der Zeiger durch Aufruf von T_P_POINTER in seiner Länge angelegt: Anzahl Komponenten * NSINGLAN + NDAT_KONST-Worte.

Die Steuerdaten werden eingetragen, wobei für die Mittelwertbildung als Standard die Kennung 'MOLA' gesetzt wird.

Im zweiten Fall muß der Zeiger ebenfalls neu gesetzt werden. Doch werden hier alle Daten der Basisversion umgespeichert. Im Steuerblock wird die Kennung eingetragen, daß der Zeiger für die höhere Version neu angelegt wurde.

T_G_SINGLE_COMP

Holen von Stoffkonstanten

FORMAT

T_G_SINGLE_COMP(PROPTYP, COMP_NR, SOURCE, VALUE, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
SOURCE	C(*)	Schreiben
VALUE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffkonstantenkennung

COMP_NR

Komponentennummer

= 0 : alle Komponenten

> 0 : eine spezielle Komponente

SOURCE

Quelleninformation

VALUE

Stoffkonstante(n) in internen Einheiten

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : Stoffdatentyp nicht gespeichert

=-2 : ungültige Komponentennummer

=-3 : ungültiger Stoffdatentyp

> 0 : Stoffdatum fehlt mindestens für Komponente RVAL

FUNKTION

T_G_SINGLE_COMP ruft das UP T_ASK_SINGLE, das die Zeigernummer des Stoffdatentyps im Common /PROP_ZEIGER/ zurückgibt. Die Position der ersten zurückzugebenden Date wird bestimmt; einmal in einfach genauen Größen und einmal in doppelt genauen Größen, wobei berücksichtigt wird, daß es Maschinen gibt, bei denen immer einfach genau gerechnet wird.

Danach werden die Daten je nach Angabe in COMP_NR umgespeichert.

Für den Fall COMP_NR = 0 muß die Dimension von SOURCE und VALUE mindestens der aktuellen Komponentenzahl entsprechen.

4.5 Stoff-Funktionen

Für jeden Typ von Stoff-Funktionen gibt es einen Zeiger im Common /PROP_ZEIGER/.
Dieser Zeiger weist auf 4-Byte-Größen im Common /PROP_DATA/.
Vorgelagert sind jedem Datentyp 4 Steuergrößen mit folgender Belegung:

C4	I4	C4	I4
Kürzel des Stofftyps	Anzahl gespeicherter Daten in Worten	Typ der Mittelwertbildung	Versionskennung = 0: Basisversion = 1: höhere Version

Die Daten sind als Block der Länge Anzahl Komponenten * NWPLAN in folgender Form gespeichert:

	I4	C4	I4	C4	I4	I4
Zeiger ------→	R8-Zeiger	Gleichung	Koeff.zahl	Quelle	Temp-Zeiger	Extrapolations-kennung
	R8-Zeiger	Gleichung	Koeff.zahl	Quelle	Temp-Zeiger	Extrapolations-kennung
	•	•	•	•	•	•
	•	•	•	•	•	•

R8-Zeiger weist auf die Speicherposition des ersten Koeffizienten der entsprechenden Komponente im Datenfeld. Hierbei bedeutet:

R8-Zeiger > 0 : Koeffizienten sind gespeichert
R8-Zeiger = 0 : Koeffizienten sind nicht gespeichert, werden aber gebraucht
R8-Zeiger =-1 : Koeffizienten werden nicht gebraucht.

Für die Verwaltung von Stoff-Funktionen sind folgende Unterprogramme vorhanden:

- T_P_PURE_COEFF
  - T_ASK_PURE
  - T_ASK_EQUA
  - T_ACT_PURE
    - T_P_POINTER
  - T_P_POINTER
- T_P_PURE_EXTR
  - T_ASK_PURE
  - T_ACT_PURE
    - T_P_POINTER
  - T_P_POINTER
- T_G_PURE_COEFF
  - T_ASK_PURE
- T_G_PURE_EXTR
  - T_ASK_PURE

T_P_PURE_COEFF

Speichern von Daten für Stoff-Funktionen

FORMAT

T_P_PURE_COEFF(PROPTYP, COMP_NR, EQUATION,
COEFF_NO, SOURCE, COEFFICIENTS,
TMIN, TMAX, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
EQUATION	C	Lesen
COEFF_NO	I4	Lesen
SOURCE	C	Lesen
COEFFICIENTS	R8(*)	Lesen
TMIN	R8	Lesen
TMAX	R8	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffkonstantenkennung

COMP_NR

Komponentennummer

EQUATION

Gleichungskennung

COEFF_NO

Anzahl zu speichernder Koeffizienten

SOURCE

Quelleninformation

COEFFICIENTS

Koeffizienten

TMIN

untere Temperaturgrenze für den Gültigkeitsbereich der Koeffizienten

TMAX

obere Temperaturgrenze für den Gültigkeitsbereich der Koeffizienten

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

=-1 : ungültiger Stoffdatentyp

=-2 : ungültige Komponentennummer

FUNKTION

T_P_PURE_COEFF ruft das Unterprogramm T_ACT_PURE zur eventuellen Aktivierung des für den Stoffdatentyp gültigen Zeigers.

Die Gültigkeit der Funktionskennung wird mit dem Unterprogramm T_ASK_EQUA geprüft.

Durch Aufruf des Unterprogramms T_P_POINTER wird der Speicherplatz für die Koeffizienten reserviert. Die Speicherung der Daten erfolgt in der oben beschriebenen Form.

T_ASK_PURE

Stofftypkennung für Stoff-Funktionen prüfen

FORMAT

T_ASK_PURE(PROPTYP, NUMMER, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

NUMMER

Nummer des Stoffdatentyps

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : angegebener Stofftyp ist keine Stoff-Funktion

FUNKTION

T_ASK_PURE prüft den angegebenen Stoffdatentyp auf Gültigkeit.

Die Anzahl und Reihenfolge der in T_ASK_PURE initialisierten Typen für Stoff-Funktionen muß mit der Anzahl und der Reihenfolge der Zeiger im Common /PROP_ZEIGER/ übereinstimmen.

Der Parameter NPURE gibt die Anzahl an; dieser Parameter entspricht der Common-Größe NWPAKT im Common /PROP_VERWALT/.

T_ASK_EQUA

Kennung für Gleichungstypen von Stoff-Funktionen prüfen

FORMAT

T_ASK_EQUA(EQUATYP, NUMMER, RVAL)

Name	Typ	Zugriff
EQUATYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

EQUATYP

Kennung des Gleichungstyps

NUMMER

Nummer des Gleichungstyps

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : angegebener Gleichungstyp unbekannt

FUNKTION

T_ASK_EQUA prüft den angegebenen Gleichungstyp auf Gültigkeit.

Der Parameter NEQUA gibt die Anzahl der bekannten Gleichungstypen an.

T_ACT_PURE

Aktivieren von Zeigern für Stoff-Funktionen

FORMAT

T_ACT_PURE(PROPTYP, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : angegebener Stofftyp ist keine Stoff-Funktion

= 1 : Zeiger konnte nicht gesetzt werden

FUNKTION

T_ACT_PURE unterscheidet 2 Fälle, für die der benötigte Zeiger angelegt werden muß:

1. der Zeiger ist noch nicht aktiviert
2. bei einem Stoffsystem höherer Version ist der Zeiger für die Basisversion aktiviert, für die höhere Version aber noch kein eigener Zeiger angelegt.

Im ersten Fall wird der Zeiger durch Aufruf von T_P_POINTER in seiner Länge angelegt: Anzahl Komponenten * NWPLAN + NDAT_KONST-Worte.

Die Steuerdaten werden eingetragen, wobei für die Mittelwertbildung als Standard die Kennung 'MOLA' gesetzt wird.

Im zweiten Fall muß der Zeiger ebenfalls neu gesetzt werden. Doch werden hier alle Daten der Basisversion umgespeichert. Die Zeiger auf die gespeicherten Koeffizienten müssen ebenfalls neu gesetzt und für jede Komponente neu eingetragen werden. Im Steuerblock wird die Kennung gesetzt, daß der Zeiger für die höhere Version neu angelegt wurde.

T_P_PURE_EXTR

Speichern der Extrapolationskennung für Stoff-Funktionen

FORMAT

T_P_PURE_EXTR(PROPTYP, COMP_NR, EXTKENN, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
EXTKENN	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentypkennung

COMP_NR

Komponentennummer

EXTKENN

Extrapolationskennung

=0: keine speziellen Extrapolationsgleichungen

=1: spezielle Extrapolationsgleichungen

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

=-1 : ungültiger Stoffdatentyp

=-2 : ungültige Komponentennummer

FUNKTION

T_P_PURE_EXTR ruft das Unterprogramm T_ACT_PURE zur eventuellen Aktivierung des für den Stoffdatentyp gültigen Zeigers und speichert die Extrapolationskennung.

T_G_PURE_COEFF

Holen von Daten für Stoff-Funktionen

FORMAT

T_G_PURE_COEFF(PROPTYP, COMP_NR, EQUATION,
COEFF_NO, SOURCE, COEFFICIENTS,
TMIN, TMAX, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
EQUATION	C	Schreiben
COEFF_NO	I4	Schreiben
SOURCE	C	Schreiben
COEFFICIENTS	R8(*)	Schreiben
TMIN	R8	Schreiben
TMAX	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffkonstantenkennung

COMP_NR

Komponentennummer

EQUATION

Gleichungskennung

COEFF_NO

Anzahl gespeicherter Koeffizienten

SOURCE

Quelleninformation

COEFFICIENTS

Koeffizienten

TMIN

untere Temperaturgrenze für den Gültigkeitsbereich der Koeffizienten

TMAX

obere Temperaturgrenze für den Gültigkeitsbereich der Koeffizienten

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : Stoffdatum nicht gespeichert

=-2 : ungültige Komponentennummer

=-3 : Stoffdatentyp nicht gültig

> 0 : Stoffdatum für diese Komponente nicht gespeichert

FUNKTION

T_G_PURE_COEFF ruft das UP T_ASK_PURE auf, das die Zeigernummer des Stoffdatentyps im Common /PROP_ZEIGER/ zurückgibt. Die Position der ersten zurückzugebenden Date wird bestimmt; einmal in einfach genauen Größen und einmal in doppelt genauen Größen.

Ist der Verkettungszeiger noch nicht gesetzt, wird RVAL = der Komponentennummer gesetzt.

Ansonsten werden die Stammdaten und die Koeffizienten für die gewünschte Komponente umgespeichert.

T_G_PURE_EXTR

Holen der Extrapolationskennung für Stoff-Funktionen

FORMAT

T_G_PURE_EXTR(PROPTYP, COMP_NR, EXTKENN, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
EXTKENN	I	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffkonstantenkennung

COMP_NR

Komponentennummer

EXTKENN

Extrapolationskennung

=0: keine spezielle Extrapolationsfunktion

=1: spezielle Extrapolationsfunktion

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : Stoffdatum nicht gespeichert

=-2 : ungültige Komponentennummer

=-3 : Stoffdatentyp nicht gültig

> 0 : Stoffdatum für diese Komponente nicht gespeichert

FUNKTION

T_G_PURE_EXTR ruft das UP T_ASK_PURE auf, das die Zeigernummer des Stoffdatentyps im Common /PROP_ZEIGER/ zurückgibt. Die Position der zurückzugebenden Date wird bestimmt und die Extrapolationskennung umgespeichert.

4.6 Stoffdatenparameter

Als Stoffdatenparameter werden diejenigen Größen bezeichnet, die mehrere Anzahl Komponenten lange Vektoren zur Speicherung der Daten benötigen.
Für jeden Typ von Stoffdatenparametern gibt es einen Zeiger im Common /PROP_ZEIGER/.
Dieser Zeiger weist auf 8-Byte-Größen im Common /PROP_DATA/.
Vorgelagert sind jedem Datentyp 8 Steuergrößen mit zur Zeit folgender Belegung:

C4	I4	I4	I4	I4	I4	I4	I4
Kürzel des Stofftyps	Anzahl Daten in 8-Byte-Gr.	Hilfsgr.: 0:zuberechnen 1:berechnet	Version 0: Basis 1: Höher	noch nicht belegt	Anzahl gespeicherter Vektoren	Anzahl Hilfs-vektoren	noch nicht belegt

Die Daten sind als Vektoren der Länge Anzahl Komponenten hintereinander abgelegt , als letztes die Hilfsvektoren.

Für die Verwaltung von Stoffkonstanten sind folgende Unterprogramme vorhanden:

- T_P_PARAM
  - T_ASK_PARAM
  - T_P_UNIRQ
    - T_ACT_PARAM
      - T_P_POINTER
  - T_P_RKS
    - T_ACT_PARAM
      - T_P_POINTER
  - T_P_PRAB
    - T_ACT_PARAM
      - T_P_POINTER
  - T_P_FLORY
    - T_ACT_PARAM
      - T_P_POINTER
  - T_P_DMER
    - T_ACT_PARAM
      - T_P_POINTER
  - T_P_TMER
    - T_ACT_PARAM
      - T_P_POINTER
  - T_P_HMER
    - T_ACT_PARAM
      - T_P_POINTER
- T_G_PARAM
  - T_ASK_PARAM

Da die T_P_Programme für die einzelnen Datentypen identisch aufgebaut sind, wird hier als Beispiel nur das Programm T_P_RKS beschrieben.

T_P_PARAM

Speichern von Stoffdatenparametern

FORMAT

T_P_PARAM(PROPTYP, COMP_NR, ELEMENTS, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
ELEMENTS	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffkonstantenkennung

COMP_NR

Komponentennummer

ELEMENTS

zu speichernde Parameter

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

> 1 : Element RVAL-1 ungültig; Warnung bei der trotzdem gespeichert wird

=-1 : ungültiger Stoffdatentyp

=-2 : ungültige Komponentennummer

FUNKTION

T_P_PARAM ruft die für den Stoffdatentyp spezifischen Unterprogramme auf, um die eingegebenen Elemente (Anzahl abhängig vom Parametertyp) zu prüfen und die Anzahl zu bestimmen.

Danach speichert das Unterprogramm die Parameter in die Vektoren ein.

T_ASK_PARAM

Stofftypkennung für Parameter prüfen

FORMAT

T_ASK_PARAM(PROPTYP, NUMMER, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

NUMMER

Nummer des Stoffdatentyps

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : angegebener Stofftyp ist kein Stoffparameter

FUNKTION

T_ASK_PARAM prüft den angegebenen Stoffdatentyp auf Gültigkeit.

Die Anzahl und Reihenfolge der in T_ASK_PARAM initialisierten Typen für Stoffparameter muß mit der Anzahl und der Reihenfolge der Zeiger im Common /PROP_ZEIGER/ übereinstimmen.

Der Parameter NPARAM gibt die Anzahl an; dieser Parameter entspricht der Common-Größe NPARAKT im Common /PROP_VERWALT/.

T_P_RKS

Speichervorbereitung für Redlich-Kwong-Soave-Parameter

FORMAT

T_P_RKS(PROPTYP, COMP_NR, ELEMENTS, IRKS, IPOS, NELEM, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
ELEMENTS	R8(*)	Lesen
IRKS	I4	Ändern
IPOS	I4	Schreiben
NELEM	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Parametertyp

COMP_NR

Komponentennummer

ELEMENTS

zu speichernde Elemente

IRKS

Speicherposition des ersten Elements der Parametervektoren

IPOS

Position des ersten zu speichernden Elements

NELEM

Anzahl zu speichernder Elemente

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

= 2 : A <= 0 gegeben

= 3 : B <= 0 gegeben

FUNKTION

Durch Aufruf von T_ACT_PARAM wird der Speicherplatz, falls notwendig, aktiviert und die Kennung gesetzt, daß die Hilfsvektoren neu gerechnet werden müssen.

Danach werden die Elemente auf Gültigkeit geprüft.

Die Ausgabegröße NELEM wird auf die Anzahl Vektoren für RKS-Parameter gesetzt.

Die Speicherposition IPOS wird als absolute Position, d.h. IRKS aufaddiert, bestimmt.

T_ACT_PARAM

Aktivieren von Zeigern für Stoffparameter

FORMAT

T_ACT_PARAM(PROPTYP, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : angegebener Stofftyp ist keine Stoff-Funktion

= 1 : Zeiger konnte nicht gesetzt werden

FUNKTION

T_ACT_PARAM unterscheidet 2 Fälle, für die der benötigte Zeiger angelegt werden muß:

1. der Zeiger ist noch nicht aktiviert
2. bei einem Stoffsystem höherer Version ist der Zeiger für die Basisversion aktiviert, für die höhere Version aber noch kein eigener Zeiger angelegt.

Im ersten Fall wird der Zeiger durch Aufruf von T_P_POINTER in seiner Länge in Doppelworten angelegt:

Anzahl Komponenten * (Anzahl Vektoren + Anzahl Hilfsvektoren) + 4

und die Daten im Steuerblock eingetragen.

Im zweiten Fall muß der Zeiger ebenfalls neu gesetzt werden. Doch werden hier alle Daten der Basisversion umgespeichert. Im Steuerblock wird die Kennung gesetzt, daß der Zeiger für die höhere Version neu angelegt wurde.

T_G_PARAM

Holen von Stoffparametern

FORMAT

T_G_PARAM(PROPTYP, COMP_NR, ELEMENTS, NELEM, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
ELEMENTS	R8(*)	Schreiben
NELEM	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffkonstantenkennung

COMP_NR

Komponentennummer

ELEMENTS

gespeicherte Parameter

NELEM

Anzahl gespeicherter Elemente

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : ungültiger Stoffdatentyp

=-2 : keine Daten gespeichert

=-3 : ungültige Komponentennummer

FUNKTION

Mit dem Programm T_ASK_PARAM wird die Zeigernummer im Common /PROP_ZEIGER/ bestimmt.

Die Anzahl der gespeicherten Vektoren bestimmt die Anzahl Elemente, die für eine Komponente zurückgegeben werden müssen.

4.7 Matrizen

Für jeden Typ von Stoffdatenmatrizen gibt es einen Zeiger im Common /PROP_ZEIGER/.
Dieser Zeiger weist auf 8-Byte-Größen im Common /PROP_DATA/.
Vorgelagert sind jedem Datentyp 8 Steuergrößen mit zur Zeit folgender Belegung:

C4	I4	I4	I4	I4	I4	I4	I4
Kürzel des Stofftyps	Anzahl Daten in 8-Byte-Gr.	Hilfsgr.: 0: zu ber. 1:berechn.	Version 0: Basis 1: Höher	Anzahl Matrizen	noch nicht belegt	Anzahl Hilfs-vektoren	Anzahl Hilfs-matrizen

Die Daten sind als Matrizen der Länge Anzahl Komponenten * Anzahl Komponenten hintereinander abgelegt.

Für die Verwaltung von Stoffdatenmatrizen sind folgende Unterprogramme vorhanden:

- T_P_MATRIX
  - T_ASK_MATRIX
  - T_P_NRTL
    - T_ASK_NRTL
    - T_ACT_MATRIX
      - T_P_POINTER
  - T_P_UNIQUAC
    - T_ACT_MATRIX
      - T_P_POINTER
  - T_P_SRK
    - T_ACT_MATRIX
      - T_P_POINTER
  - T_P_PR
    - T_ACT_MATRIX
      - T_P_POINTER
    - T_ACT_PARAM
  - T_P_WILSON
    - T_ASK_WILSON
    - T_ACT_MATRIX
      - T_P_POINTER
  - T_P_HNRY
    - T_ASK_HNRY
    - T_ACT_MATRIX
      - T_P_POINTER
- T_G_MATRIX
  - T_ASK_MATRIX
  - T_G_NRTL
    - T_ASK_NRTL
  - T_G_UNIQUAC
  - T_G_SRK
  - T_G_PR
  - T_G_WILSON
    - T_ASK_WILSON
  - T_G_HNRY
    - T_ASK_HNRY

Da die T_G_ und T_P_Programme für die einzelnen Matrixtypen identisch aufgebaut sind, werden als Beispiel nur die Programme T_G_NRTL, T_P_NRTL und T_ASK_NRTL beschrieben.

T_P_MATRIX

Speichern von Stoffdatenmatrizen

FORMAT

T_P_MATRIX(PROPTYP, PROPSPEZ, COMP_I, COMP_J,
ELEMENTS, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
PROPSPEZ	C	Lesen
COMP_I	I4	Lesen
COMP_J	I4	Lesen
ELEMENTS	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Matrixkennung

PROPSPEZ

spezieller Matrixtyp, falls es mehrere Matrizen zu PROPTYP gibt

COMP_I

Komponentennummer der ersten Komponente

COMP_J

Komponentennummer der zweiten Komponente

ELEMENTS

zu speichernde Elemente W(i,j) und W(j,i)

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

> 1 : fehlerhafte Elemente; Warnung bei der trotzdem gespeichert wird

=-1 : ungültige Kennung in PROPSPEZ

=-2 : ungültiger Stoffdatentyp

=-3 : ungültige Komponentennummer

FUNKTION

T_P_MATRIX ruft die für den Matrixtyp spezifischen Unterprogramme auf, um die eingegebenen Elemente zu prüfen und die Speicherpositionen zu bestimmen.

Danach speichert das Unterprogramm die Matrixelemente in den Daten-Common.

T_ASK_MATRIX

Kennung für Matrizen prüfen

FORMAT

T_ASK_MATRIX(PROPTYP, NUMMER, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Matrixtyp

NUMMER

Nummer des Matrixtyps

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : angegebener Stofftyp ist keine Matrix

FUNKTION

T_ASK_Matrix prüft den angegebenen Stoffdatentyp auf Gültigkeit.

Die Anzahl und Reihenfolge der in T_ASK_MATRIX initialisierten Typen für Stoffmatrizen muß mit der Anzahl und der Reihenfolge der Zeiger im Common /PROP_ZEIGER/ übereinstimmen.

Der Parameter NMATRIX gibt die Anzahl an; dieser Parameter entspricht der Common-Größe NMATAKT im Common /PROP_VERWALT/.

T_P_NRTL

Speichervorbereitung für NRTL-Matrizen

FORMAT

T_P_NRTL(PROPTYP, PROPSPEZ, NKOMP, COMP_I, COMP_J, ELEMENTS, INRTL, IPOS_I, IPOS_J, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
PROPSPEZ	C	Lesen
NKOMP	I4	Lesen
COMP_I	I4	Lesen
COMP_J	I4	Lesen
ELEMENTS	R8(*)	Lesen
INRTL	I4	Ändern
IPOS_I	I4	Schreiben
IPOS_J	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Matrixtyp

PROPSPEZ

spez. Matrixkennung

NKOMP

aktuelle Komponentenzahl

COMP_I

Komponentennummer der 1. Komponente

COMP_J

Komponentennummer der 2. Komponente

ELEMENTS

zu speichernde Elemente

INRTL

Speicherposition des ersten Elements der NRTL-Matrizen

IPOS_I

Position des ersten zu speichernden Elements

IPOS_J

Position des ersten zu speichernden Elements

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : ungültiger Typ in PROPSPEZ

= 1 : Speicherplatz erschöpft

= 2 : bei Alpha-Matrix: ein Element war 0 und wurde gleich dem anderen gesetzt

= 3 : bei Alpha-Matrix: Elemente sind unsymmetrisch

FUNKTION

Für NRTL werden 4 Matrizen gespeichert und zwar in der Reihenfolge:

- ALPHA
- BETA
- A
- B

Durch Aufruf von T_ACT_MATRIX wird der Speicherplatz, falls notwendig, aktiviert und die Kennung gesetzt, daß eventuell Hilfsvektoren neu gerechnet werden müssen.

Nach Identifizierung des Matrixtyps werden die Elemente entsprechend auf Gültigkeit geprüft.

Die Speicherpositionen IPOS_I und IPOS_J werden als absolute Positionen, d.h. INRTL aufaddiert, bestimmt.

T_ASK_NRTL

Kennung für NRTL-Matrizen prüfen

FORMAT

T_ASK_NRTL(PROPTYP, NUMMER, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Matrixtyp

NUMMER

Nummer des Matrixtyps

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : angegebener Stofftyp ist keine NRTL-Untermatrix

FUNKTION

T_ASK_NRTL prüft den angegebenen Stoffdatentyp auf Gültigkeit.

T_ACT_MATRIX

Aktivieren von Zeigern für Stoffmatrizen

FORMAT

T_ACT_MATRIX(PROPTYP, ZEIGER, ANZAHL_MATRIZEN,
ANZAHL_HVEKTOREN, ANZAHL_HMATRIZEN,
RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
ZEIGER	I4	Ändern
ANZAHL_MATRIZEN	I4	Lesen
ANZAHL_HVEKTOREN	I4	Lesen
ANZAHL_HMATRIZEN	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

ZEIGER

Stoffdatenzeiger

ANZAHL_MATRIZEN

Anzahl zu speichernder Matrizen

ANZAHL_HVEKTOREN

Anzahl zu speichernder Hilfsvektoren

ANZAHL_HMATRIZEN

Anzahl zu speichernder Hilfsmatrizen

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Zeiger konnte nicht gesetzt werden

FUNKTION

T_ACT_MATRIX unterscheidet 2 Fälle, für die der benötigte Zeiger angelegt werden muß:

1. der Zeiger ist noch nicht aktiviert
2. bei einem Stoffsystem höherer Version ist der Zeiger für die Basisversion aktiviert, für die höhere Version aber noch kein eigener Zeiger angelegt.

Im ersten Fall wird der Zeiger durch Aufruf von T_P_POINTER in seiner Länge in Doppelworten angelegt:

Anzahl Komponenten * Anzahl Komponenten * (Anzahl Matrizen + Anzahl_Hilfsmatrizen) + Anzahl Komponenten * Anzahl Vektoren + 4

und die Daten im Steuerblock eingetragen.

Im zweiten Fall muß der Zeiger ebenfalls neu gesetzt werden. Doch werden hier alle Daten der Basisversion umgespeichert. Im Steuerblock wird die Kennung gesetzt, daß der Zeiger für die höhere Version neu angelegt wurde.

T_G_MATRIX

Holen von Matrixelementen

FORMAT

T_G_MATRIX(PROPTYP, PROPSPEZ, COMP_I, COMP_J,
ELEMENTS, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
PROPSPEZ	C	Lesen
COMP_I	I4	Lesen
COMP_J	I4	Lesen
ELEMENTS	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Matrixtyp

PROPSPEZ

spez. Matrixkennung

COMP_I

Komponentennummer der 1. Komponente

COMP_J

Komponentennummer der 2. Komponente

ELEMENTS

gespeicherte Elemente

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : ungültiger Typ in PROPSPEZ

=-2 : Komponentennummer fehlerhaft

= 1 : Datentyp nicht gespeichert

FUNKTION

Zu unterscheiden sind 3 Fälle:

a) COMP_I, COMP_J > 0 : Rückgabe von W(i,j) und W(j,i)

b) COMP_I = 0, COMP_J > 0 : Rückgabe von W(i,j), j= 1, Anz. Komponenten

c) COMP_I > 0, COMP_J = 0 : Rückgabe von W(i,j), i= 1, Anz. Komponenten

Die Speicherposition der gewünschten Elemente wird in einem zu der entsprechenden Matrix gehörenden T_G_-Programm bestimmt und dann entsprechend den Angaben über die Komponentennummern umgespeichert.

T_G_NRTL

Bestimmen der Position von in T_G_MATRIX gewünschten Elementen innerhalb der NRTL-Matrizen.

FORMAT

T_G_NRTL(PROPSPEZ, NKOMP, COMP_I, COMP_J, INRTL, IPOS_I, IPOS_J, RVAL)

Name	Typ	Zugriff
PROPSPEZ	C	Lesen
NKOMP	I4	Lesen
COMP_I	I4	Lesen
COMP_J	I4	Lesen
INRTL	I4	Lesen
IPOS_I	I4	Schreiben
IPOS_J	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPSPEZ

spez. Matrixkennung

NKOMP

aktuelle Komponentenzahl

COMP_I

Komponentennummer der 1. Komponente

=0 bedeutet eine ganze Spalte

COMP_J

Komponentennummer der 2. Komponente

=0 bedeutet eine ganze Zeile

INRTL

Zeiger auf die Position der ersten gespeicherten NRTL-Date (R8)

IPOS_I

Position des Elements i,j der gewünschten NRTL-Matrizen

IPOS_J

Position des Elements j,i der gewünschten NRTL-Matrizen

RVAL

Fehlerkennung

=0: alles ok.

=1: ungültiger Typ in PROPSPEZ

FUNKTION

T_G_NRTL bestimmt die Position der umzuspeichernden Elemente.

Der übergebene Zeiger INRTL weist auf das erste gespeicherte Element.

Die Positionen müssen in Abhängigkeit vom speziellen Matrixtyp bestimmt werden.

4.8 Stoff-Funktionen für Binärdaten

Für jeden Typ von Stoff-Funktionen für Binärdaten gibt es einen Zeiger im Common /PROP_ZEIGER/.
Dieser Zeiger weist auf 4-Byte-Größen im Common /PROP_DATA/.
Vorgelagert sind jedem Datentyp 4 Steuergrößen mit folgender Belegung:

C4	I4	I4	I4
Kürzel des Stofftyps	Anzahl ge-speicherter Daten in Worten		Versionskennung = 0: Basisversion = 1: höhere Version

Die Daten sind als Block der Länge Anzahl Komponenten * NBILAN in folgender Form gespeichert:

	I4	I4	I4	C4	I4	C4
Zeiger ------→	R8-Zeiger	COMP_I	COMP_J	Gleichung	Koeff.zahl	Quelle
	R8-Zeiger	COMP_I	COMP_J	Gleichung	Koeff.zahl	Quelle
	•			•	•	•
	•			•	•	•

R8-Zeiger weist auf die Speicherposition des ersten Koeffizienten der entsprechenden Komponente im Datenfeld. Hierbei bedeutet:

R8-Zeiger > 0 : Koeffizienten sind gespeichert
R8-Zeiger = 0 : Koeffizienten sind nicht gespeichert, werden aber gebraucht
R8-Zeiger =-1 : Koeffizienten werden nicht gebraucht.

Für die Verwaltung von Stoff-Funktionen sind folgende Unterprogramme vorhanden:

- T_P_BINA_COEFF
  - T_ASK_BINA
  - T_ASK_EQUA
  - T_ACT_BINA
    - T_P_POINTER
  - T_P_POINTER
- T_G_BINA_COEFF
  - T_ASK_BINA

T_P_BINA_COEFF

Speichern von Daten für Stoff-Funktionen binärer Daten

FORMAT

T_P_BINA_COEFF(PROPTYP, COMP_I, COMP_J, EQUATION, COEFF_NO, SOURCE, COEFFICIENTS, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_I	I4	Lesen
COMP_J	I4	Lesen
EQUATION	C	Lesen
COEFF_NO	I4	Lesen
SOURCE	C	Lesen
COEFFICIENTS	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatenkennung

COMP_I

Komponentennummer

COMP_J

Komponentennummer

EQUATION

Gleichungskennung

COEFF_NO

Anzahl zu speichernder Koeffizienten

SOURCE

Quelleninformation

COEFFICIENTS

Koeffizienten

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

=-1 : ungültiger Stoffdatentyp

=-2 : ungültige Komponentennummer

FUNKTION

T_P_BINA_COEFF ruft das Unterprogramm T_ACT_BINA_COEFF zur eventuellen Aktivierung des für den Stoffdatentyp gültigen Zeigers.

Die Gültigkeit der Funktionskennung wird mit dem Unterprogramm T_ASK_EQUA geprüft.

Durch Aufruf des Unterprogramms T_P_POINTER wird der Speicherplatz für die Koeffizienten reserviert. Die Speicherung der Daten erfolgt in der oben beschriebenen Form.

T_ASK_BINA

Stofftypkennung für Stoff-Funktionen binärer Daten prüfen

FORMAT

T_ASK_BINA(PROPTYP, NUMMER, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

NUMMER

Nummer des Stoffdatentyps

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : angegebener Stofftyp ist keine Stoff-Funktion

FUNKTION

T_ASK_BINA prüft den angegebenen Stoffdatentyp auf Gültigkeit.

Die Anzahl und Reihenfolge der in T_ASK_BINA initialisierten Typen für Stoff-Funktionen muß mit der Anzahl und der Reihenfolge der Zeiger im Common /PROP_ZEIGER/ übereinstimmen.

Der Parameter NBINA gibt die Anzahl an; dieser Parameter entspricht der Common-Größe NBIAKT im Common /PROP_VERWALT/.

T_ACT_BINA_COEFF

Aktivieren von Zeigern für Stoff-Funktionen für binäre Daten

FORMAT

T_ACT_BINA_COEFF(PROPTYP, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : angegebener Stofftyp ist keine Stoff-Funktion

= 1 : Zeiger konnte nicht gesetzt werden

FUNKTION

T_ACT_BINA_COEFF unterscheidet 2 Fälle, für die der benötigte Zeiger angelegt werden muß:

1. der Zeiger ist noch nicht aktiviert
2. bei einem Stoffsystem höherer Version ist der Zeiger für die Basisversion aktiviert, für die höhere Version aber noch kein eigener Zeiger angelegt.

Im ersten Fall wird der Zeiger durch Aufruf von T_P_POINTER in seiner Länge angelegt: (Anzahl Komponenten * (Anzahl Komponenten-1)/2 * NBILAN + NDAT_KONST-Worte.

Die Steuerdaten werden eingetragen.

Im zweiten Fall muß der Zeiger ebenfalls neu gesetzt werden. Doch werden hier alle Daten der Basisversion umgespeichert. Die Zeiger auf die gespeicherten Koeffizienten müssen ebenfalls neu gesetzt und für jede Komponente neu eingetragen werden. Im Steuerblock wird die Kennung gesetzt, daß der Zeiger für die höhere Version neu angelegt wurde.

T_G_BINA_COEFF

Holen von Daten für Stoff-Funktionen für binäre Daten

FORMAT

T_G_BINA_COEFF(PROPTYP, COMP_I, COMP_J, EQUATION, COEFF_NO, SOURCE, COEFFICIENTS, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_I	I4	Lesen
COMP_J	I4	Lesen
EQUATION	C	Schreiben
COEFF_NO	I4	Schreiben
SOURCE	C	Schreiben
COEFFICIENTS	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatenkennung

COMP_I

Komponentennummer

COMP_J

Komponentennummer

EQUATION

Gleichungskennung

COEFF_NO

Anzahl gespeicherter Koeffizienten

SOURCE

Quelleninformation

COEFFICIENTS

Koeffizienten

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : Stoffdatum nicht gespeichert

=-2 : ungültige Komponentennummer

=-3 : Stoffdatentyp nicht gültig

> 0 : Stoffdatum für diese Komponente nicht gespeichert

FUNKTION

T_G_BINA_COEFF ruft das UP T_ASK_BINA auf, das die Zeigernummer des Stoffdatentyps im Common /PROP_ZEIGER/ zurückgibt. Die Position der ersten zurückzugebenden Date wird bestimmt.

Ist der Verkettungszeiger noch nicht gesetzt, wird RVAL = der Komponentennummer gesetzt.

Ansonsten werden die Stammdaten und die Koeffizienten für die gewünschte Komponente umgespeichert.

4.9 Inkremente

Z. Zt. ist nur die Speicherung von UNIFAC-Inkrementen möglich. Hierfür sind im Common /PROP_ZEIGER/ folgende Zeiger angelegt:

- IUNIF_KO zur Speicherung von Reinstoffinformationen für jede Komponente. Dieser Zeiger weist auf 3 Anzahl Komponenten lange Vektoren, die folgende Information für jede Komponente enthalten: Anzahl Inkremente, r-Parameter, q-Parameter. Diese Vektoren sind einfach genau gespeichert.
- IUNIF_GR zur Speicherung von inkrementabhängigen Informationen für jede Komponente. Dieser Zeiger weist auf 5 einfach genau gespeicherte Matrizen der Dimensionierung Max. Anzahl Inkremente ∗ Anzahl Komponenten. Gespeichert werden für jede Komponente die Informationen über jedes vorkommente Inkrement: Inkrementnummer, Inkrementhäufigkeit, Hauptgruppennummer des Inkrements, r-Parameter des Inkrements, q-Parameter des Inkrements
- IUNIF_WP zur Speicherung von inkrementabhängigen Informationen für das Gemisch. Dieser Zeiger weist auf 3 einfach genaue Vektoren der Länge Max. Anzahl Gruppen. Gespeichert werden: die Inkrementnummern der unterschiedlichen Inkremente, die dazugehörenden Hauptgruppennummern und die q-Parameter dieser Inkremente. Die Anzahl der unterschiedlichen Inkremente wird als letzte Informationsgröße gespeichert, um die Wechselwirkungsmatrix richtig dimensionieren zu können.
- IUNIF_MAT zur Speicherung der Wechselwirkungsmatrix. Dieser Zeiger weist auf 2 Matrizen der Dimensionierung Anz. unterschiedlicher Gruppen * Anzahl unterschiedlicher Gruppen. In der ersten Matrix werden die Wechselwirkungsparameter gespeichert, die zweite dient als Hilfsfeld für das Berechnungsprogramm.

Vorgelagert sind jedem Datentyp 8 Steuergrößen mit zur Zeit folgender Belegung:

C4	I4	I4	I4	I4	I4	I4	I4
Kürzel des Inkrement-typs	Anzahl Daten in 4-Byte-Gr.	Hilfsgr.: 0: zu ber. 1:berechn.	Version 0: Basis 1: Höher	Anzahl Matrizen	Anzahl Vektoren	Anzahl Hilfs-vektoren	Anzahl unterschied licher Gruppen

Die 8. Größe ist nur beim Zeiger IUNIF_WP besetzt.

Für die Verwaltung von Inkrementen sind folgende Unterprogramme vorhanden:

- T_P_INKR
  - T_ASK_INKR
  - T_P_REIN_UNIF
    - T_ACT_INKR
      - T_P_POINTER
  - T_P_GROUP_UNIF
    - T_ACT_WP_INKR
      - T_P_POINTER
- T_P_MIXT_INKR
  - T_ASK_INKR
  - T_P_UFMIXT
    - T_ACT_MAT_INKR
      - T_P_POINTER
- T_G_INKR
  - T_ASK_INKR
  - T_G_REIN_UNIF

T_P_INKR

Speichern von Inkrementinformationen eines Reinstoffes

FORMAT

T_P_INKR(INKTYP, COMP, ANZINKR, INKR_NO, NUMBER, HAUPT, ELEMENTS, RVAL)

Name	Typ	Zugriff
INKTYP	C	Lesen
COMP	I4	Lesen
ANZINKR	I4	Lesen
INKR_NO	I4(∗)	Lesen
NUMBER	I4(∗)	Lesen
HAUPT	I4(∗)	Lesen
ELEMENTS	R4(∗,∗)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

INKTYP

Inkrementkennung

COMP

Komponentennummer

ANZINKR

Anzahl der Inkremente

INKR_NO

Vektor der vorkommenden Inkrementnummern

NUMBER

Vektor der Häufigkeit der vorkommenden Inkremente

HAUPT

Vektor der Hauptgruppenzuordnung der vorkommenden Inkremente

ELEMENTS

Matrix von zusätzlichen Inkrementinformationen; die Anzahl Zeilen dieser Matrix ist durch den Parameter MAXGRP in der Datei PROP_INCLUDE festgelegt.

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

= 2 : maximale Anzahl Inkremente überschritten

=-1 : ungültiger Stoffdatentyp

=-2 : ungültige Komponentennummer

FUNKTION

T_P_INKR ruft die Unterprogramme auf, die für einen Inkrementtyp die Inkrementdaten für eine Komponente speichern.

T_ASK_INKR

Inkrementtypkennung prüfen

FORMAT

T_ASK_INKR(INKTYP, NUMMER, RVAL)

Name	Typ	Zugriff
INKTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Inkrementtyp

NUMMER

Nummer des Inkrementtyps

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : angegebener Typ ist nicht identifizierbar

FUNKTION

T_ASK_INKR prüft den angegebenen Inkrementtyp auf Gültigkeit.

T_P_REIN_UNIF

Speichern von UNIFAC-Inkrementinformationen eines Reinstoffes

FORMAT

T_P_REIN_UNIF(MAXGRP, INKTYP, COMP, ANZINKR, INKR_NO, NUMBER, HAUPT, ELEMENTS, RVAL)

Name	Typ	Zugriff
MAXGRP	I4	Lesen
INKTYP	C	Lesen
COMP	I4	Lesen
ANZINKR	I4	Lesen
INKR_NO	I4(∗)	Lesen
NUMBER	I4(∗)	Lesen
HAUPT	I4(∗)	Lesen
ELEMENTS	R4(∗,∗)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

MAXGRP

max. Anzahl zu speichernder Gruppen

INKTYP

Inkrementkennung

COMP

Komponentennummer

ANZINKR

Anzahl der Inkremente

INKR_NO

Vektor der vorkommenden Inkrementnummern

NUMBER

Vektor der Häufigkeit der vorkommenden Inkremente

HAUPT

Vektor der Hauptgruppenzuordnung der vorkommenden Inkremente

ELEMENTS

Matrix der r- und q-Parameter der Inkremente; die Anzahl Zeilen dieser Matrix ist durch den Parameter MAXGRP in der Datei PROP_INCLUDE festgelegt, die Spaltenzahl ist 2.

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

= 2 : maximale Anzahl Inkremente überschritten

=-1 : ungültiger Stoffdatentyp

=-2 : ungültige Komponentennummer

FUNKTION

T_P_REIN_UNIF aktiviert durch Aufruf des Unterprogramms T_ACT_INKR die Zeiger, die die Reinstoffinformationen für UNIFAC aufnehmen.

Der Zeiger IUNIF_GR im Common /PROP_ZEIGER/ weist auf 5 einfach genau gespeicherte Matrizen der Dimensionierung MAXGRP∗Anzahl Komponenten. Sie enthalten die Inkrementnummern, die Häufigkeitsangabe, die Hauptgruppennummern, die r-Parameter und die q-Parameter der Inkremente.

Der Zeiger IUNIF_KO im Common /PROP_ZEIGER/ weist auf 3 einfach genau angelegte Vektoren der Länge Anz. Komponenten. Sie enthalten die Anzahl Inkremente , die r -Parameter und die q-Parameter für jede Komponente.

T_P_REIN_UNIF speichert die eingegebenen Daten. Gleichzeitig werden die r- und q-Parameter der Inkremente aufsummiert und für diese Komponente gespeichert.

T_ACT_INKR

Aktivieren von Zeigern für Reinstoffinformationen bei Inkrementen

FORMAT

T_ACT_INKR(INKTYP, INKR_ZEIGER, ANZAHL_MATRIX,
ANZAHL_VEKTOR, ANZAHL_HVEKTOR, RVAL)

Name	Typ	Zugriff
INKTYP	C	Lesen
INKR_ZEIGER	I4	Lesen/Schreiben
ANZAHL_MATRIX	I4	Lesen
ANZAHL_VEKTOR	I4	Lesen
ANZAHL_HVEKTOR	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

INKTYP

Inkrementtyp

INKR_ZEIGER

anzulegender Zeiger

ANZAHL_MATRIX

Anzahl anzulegender Matrizen

ANZAHL_VEKTOR

Anzahl anzulegender Vektoren

ANZAHL_HVEKTOR

Anzahl anzulegender Hilfsvektoren

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Zeiger konnte nicht gesetzt werden

FUNKTION

T_ACT_INKR aktiviert den angegebenen Zeiger, falls dieser noch nicht aktiv ist.

Durch Aufruf von T_P_POINTER wird der Zeiger angelegt :

MAX. Gruppenzahl * Anzahl Komponenten bei Matrizen

Anzahl Komponenten für Vektoren

Die so aktivierten Zeiger weisen auf einfach genaue Größen.

T_P_GROUP_UNIF

Speichern von UNIFAC-Informationen für Gemischbehandlung

FORMAT

T_P_GROUP_UNIF(MAXGRP, ANZINKR, INKR_NO, HAUPT, QI, RVAL)

Name	Typ	Zugriff
MAXGRP	I4	Lesen
ANZINKR	I4	Lesen
INKR_NO	I4(∗)	Lesen
HAUPT	I4(∗)	Lesen
QI	R4(∗)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

MAXGRP

max. Anzahl zu speichernder Gruppen

ANZINKR

Anzahl der Inkremente

INKR_NO

Vektor der vorkommenden Inkrementnummern

HAUPT

Vektor der Hauptgruppenzuordnung der vorkommenden Inkremente

QI

Vektor der q-Parameter der Inkremente

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

= 2 : maximale Anzahl Inkremente überschritten

FUNKTION

T_P_GROUP_UNIF aktiviert durch Aufruf des Unterprogramms T_ACT_WP_INKR den Zeiger zur Speicherung der Informationen zum Aufbau der Gemischinformation für UNIFAC.

Der Zeiger IUNIF_WP im Common /PROP_ZEIGER/ weist auf 3 einfach genau gespeicherte Vektoren der Dimensionierung MAXGRP. Sie enthalten die Inkrementnummern, die Hauptgruppennummern und die q-Parameter der Inkremente. Hierbei werden nur noch die unterschiedlichen Inkremente gespeichert. Die Anzahl unterschiedlicher Inkremente ergibt die Dimensionierung der Wechselwirkungsmatrix.

T_ACT_WP_INKR

Aktivieren von Zeigern für Gemischinformationen bei Inkrementen

FORMAT

T_ACT_WP_INKR(INKTYP, INKR_ZEIGER, ANZAHL_MATRIX, ANZAHL_VEKTOR, RVAL)

Name	Typ	Zugriff
INKTYP	C	Lesen
INKR_ZEIGER	I4	Lesen/Schreiben
ANZAHL_MATRIX	I4	Lesen
ANZAHL_VEKTOR	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

INKTYP

Inkrementtyp

INKR_ZEIGER

anzulegender Zeiger

ANZAHL_MATRIX

Anzahl anzulegender Matrizen

ANZAHL_VEKTOR

Anzahl anzulegender Vektoren

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Zeiger konnte nicht gesetzt werden

FUNKTION

T_ACT_WP_INKR aktiviert den angegebenen Zeiger, falls dieser noch nicht aktiv ist.

Durch Aufruf von T_P_POINTER wird der Zeiger angelegt :

Max. Gruppenzahl * Max. Gruppenzahl bei Matrizen

Max. Gruppenzahl für Vektoren

Die so aktivierten Zeiger weisen auf einfach genaue Größen.

T_P_MIXT_INKR

Speichern von Elementen der Gemischmatrix bei Inkrementmethoden

FORMAT

T_P_MIXT_INKR(INKTYP, INKR_NO1, INKR_NO2, ELEMENTS, RVAL)

Name	Typ	Zugriff
INKTYP	C	Lesen
INKR_NO1	I4	Lesen
INKR_NO2	I4	Lesen
ELEMENTS	R4(2)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

INKTYP

Inkrementkennung

INKR_NO1

Inkrementnummer 1

INKR_NO1

Inkrementnummer 1

INKR_NO2

Inkrementnummer 2

ELEMENTS

zu speichernde Elemente

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

= 2 : maximale Anzahl Inkremente überschritten

=-1 : ungültiger Inkrementtyp

FUNKTION

T_P_MIXT_INKR ruft die Unterprogramme auf, die für einen Inkrementtyp die Wechselwirkungsparameter speichern.

T_P_UFMIXT

Speichern von UNIFAC-Wechselwirkungsparametern

FORMAT

T_P_UFMIXT(MAXGRP, INKR_NO1, INKR_NO2, ELEMENTS, RVAL)

Name	Typ	Zugriff
MAXGRP	I4	Lesen
INKR_NO1	I4	Lesen
INKR_NO2	I4	Lesen
ELEMENTS	R4(2)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

MAXGRP

max. Anzahl zu speichernder Gruppen

INKR_NO1

Nummer der ersten Hauptgruppe

INKR_NO2

Nummer der zweiten Hauptgruppe

ELEMENTS

Wechselwirkungsparameter der Hauptgruppen

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

= 2 : maximale Anzahl Inkremente überschritten

FUNKTION

T_P_UFMIXT aktiviert durch Aufruf des Unterprogramms T_ACT_MAT_INKR den Zeiger zur Speicherung der Wechselwirkungsmatrix für UNIFAC.

Der Zeiger IUNIF_MAT im Common /PROP_ZEIGER/ weist auf 2 einfach genau gespeicherte Matrizen der Dimensionierung Anzahl unterschiedlicher Gruppen * Anzahl unterschiedlicher Gruppen.

Die Anzahl unterschiedlicher Gruppen wird beim Speichern der Reinstoffinformation mitbestimmt. T_P_UFMIXT speichert die Wechselwirkungsmatrix, indem es die zu den Inkrementen gehörenden Hauptgruppen prüft und die Wechselwirkungsparameter entsprechend der Inkrementposition in die Wechselwirkungsmatrix einträgt.

T_ACT_MAT_INKR

Aktivieren des Zeigers für die Wechselwirkungsmatrix bei Inkrementmethoden

FORMAT

T_ACT_MAT_INKR(INKTYP, INKR_ZEIGER, NDIM,
ANZAHL_MATRIX, ANZAHL_VEKTOR, RVAL)

Name	Typ	Zugriff
INKTYP	C	Lesen
INKR_ZEIGER	I4	Lesen/Schreiben
NDIM	I4	Lesen
ANZAHL_MATRIX	I4	Lesen
ANZAHL_VEKTOR	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

INKTYP

Inkrementtyp

INKR_ZEIGER

anzulegender Zeiger

NDIM

Dimension der anzulegenden Vektoren bzw. Matrizen

ANZAHL_MATRIX

Anzahl anzulegender Matrizen

ANZAHL_VEKTOR

Anzahl anzulegender Vektoren

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Zeiger konnte nicht gesetzt werden

FUNKTION

T_ACT_MAT_INKR aktiviert den angegebenen Zeiger, falls dieser noch nicht aktiv ist.

Durch Aufruf von T_P_POINTER wird der Zeiger in der Dimension NDIM*NDIM für Matrizen und NDIM für Vektoren angelegt. Er weist auf einfach genaue Größen.

T_G_INKR

Holen von Inkrementinformationen eines Reinstoffes

FORMAT

T_G_INKR(INKTYP, COMP, ANZINKR, INKR_NO, NUMBER, RVAL)

Name	Typ	Zugriff
INKTYP	C	Lesen
COMP	I4	Lesen
ANZINKR	I4	Schreiben
INKR_NO	I4(*)	Schreiben
NUMBER	I4(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

INKTYP

Inkrementkennung

COMP

Komponentennummer

ANZINKR

Anzahl der Inkremente

INKR_NO

Vektor der vorkommenden Inkrementnummern

NUMBER

Vektor der Häufigkeit der vorkommenden Inkremente

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : nicht gespeichert

=-1 : ungültiger Inkrementtyp

=-2 : ungültige Komponentennummer

FUNKTION

T_G_INKR ruft die Unterprogramme auf, die für einen Inkrementtyp die Inkrementdaten für eine Komponente holen.

T_G_REIN_UNIF

Holen von UNIFAC-Inkrementinformationen eines Reinstoffes

FORMAT

T_G_REIN_UNIF(MAXGRP, COMP, ANZINKR, INKR_NO, NUMBER, RVAL)

Name	Typ	Zugriff
MAXGRP	I4	Lesen
COMP	I4	Lesen
ANZINKR	I4	Lesen
INKR_NO	I4(*)	Lesen
NUMBER	I4(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

MAXGRP

max. Anzahl zu speichernder Gruppen

COMP

Komponentennummer

ANZINKR

Anzahl der Inkremente

INKR_NO

Vektor der vorkommenden Inkrementnummern

NUMBER

Vektor der Häufigkeit der vorkommenden Inkremente

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : nicht gespeichert

FUNKTION

T_G_REIN_UNIF holt aus dem internen Speicher (1. Vektor des Zeigers IUNIF_KO) die Anzahl Inkremente für diese Komponente. Die ersten beiden Matrizen des Zeigers IUNIF_GR weisen auf die Inkremente und deren Häufigkeit für die Komponenten.

4.10 Phasenschlüssel für Komponenten

Im Common /PROP_ZEIGER/ steht der Zeiger IKPHAS zur Verfügung. Er. weist auf 4-Byte-Größen im Common /PROP_DATA/.
Vorgelagert sind jedem Datentyp 4 Steuergrößen mit folgender Belegung:

C4	I4	I4	I4
Kürzel des Stofftyps	Anzahl gespeicherter Daten in Worten		Versionskennung = 0: Basisversion = 1: höhere Version

Die Daten sind als Block der Länge Anzahl Komponenten * Anzahl Phasen(=4) in folgender Form gespeichert:

	I4-LIQUID	I4-LIQUID2	I4-VAPOR	I4-SOLID
Zeiger ------→	0=nicht präsent 1=präsent	0=nicht präsent 1=präsent	0=nicht präsent 1=präsent	0=nicht präsent 1=präsent
	0=nicht präsent 1=präsent	0=nicht präsent 1=präsent	0=nicht präsent 1=präsent	0=nicht präsent 1=präsent
	•			•
	•			•

Je Komponente ist eine Zeile gespeichert.

Für die Verwaltung von Phaseninformationen sind folgende Unterprogramme vorhanden:

- T_P_INFO_COMP
  - T_P_POINTER
- T_G_INFO_COMP

T_P_INFO_COMP

Speichern von Daten für Phaseninformationen

FORMAT

T_P_INFO_COMP(TASK, PHASE, ANZAHL, VECTOR, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
PHASE	C	Lesen
ANZAHL	I4	Lesen
VECTOR	I4(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenkennung

=1 : speichern

=2 : löschen

PHASE

Phasenkennung

LIQU : Flüssigkeit bzw. 1. flüssige Phase

LIQ2 : 2. flüssige Phase

VAPO : Dampf

SOLI : Feststoff

ANZAHL

Anzahl Komponenten

VECTOR

Vektor mit den Komponentennummern, gefüllt von 1 bis ANZAHL

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Speicherplatz erschöpft

=-1 : ungültige Phasenkennung

=-2 : ungültige Komponentennummer

=-3 : ungültige TASK

FUNKTION

T_P_INFO_COMP ruft das Unterprogramm T_P_POINTER zur eventuellen Aktivierung des Zeigers.

T_G_INFO_COMP

Holen von Daten für Stoff-Funktionen für binäre Daten

FORMAT

T_G_INFO_COMP(PHASE, ANZAHL, VECTOR, RVAL)

Name	Typ	Zugriff
PHASE	C	Lesen
ANZAHL	I4	Schreiben
VECTOR	I4(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PHASE

Phasenkennung

= LIQU : Flüssigkeit bzw. 1. flüssige Phase

= LIQ2 : 2. flüssige Phase

= VAPO: Dampf

= SOLI : Feststoff

ANZAHL

Anzahl Komponenten in dieser Phase

VECTOR

Vektor mit den Komponentennummern, gefüllt von 1 bis ANZAHL

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Daten nicht gespeichert

= 2 : keine Komponente in dieser Phase gespeichert

=-1 : Phase ungültig

FUNKTION

T_G_INFO_COMP holt die Informationen aus den internen Speichern und gibt sie an das aufrufenden Programm zurück.

5 Programme zum Speichern, Holen und Aktivieren von Berechnungsdaten

5.1 Mittelwertbildungen

Für jedes Reinstoffdatum ist im Steuerblock die Kennzeichnung der Mittelwertbildung vorgesehen. Der Standardeintrag, der beim Anlegen des Zeigers gespeichert wird, kann jederzeit überschrieben werden.
Für die Gasdichte wird die Mittelwertbildung über ein speziellen Methodenzeiger verwaltet.

Für die Verwaltung von Mittelwertbildungen sind folgende Unterprogramme vorhanden:

- T_P_AVER
  - T_ASK_ASPEZ
  - T_ASK_SINGLE
  - T_ASK_PURE
  - T_ASK_AVER
  - T_P_ASPEZ
- T_G_AVER
  - T_ASK_ASPEZ
  - T_ASK_SINGLE
  - T_ASK_PURE
  - T_G_ASPEZ

T_P_AVER

Speichern der Mittelwertmethode für einen Stoffdatentyp

FORMAT

T_P_AVER(PROPTYP, AVERTYP, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
AVERTYP	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

AVERTYP

Typ der Mittelwertbildung

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Stoffdatentyp noch nicht gespeichert

=-1 : Stoffdatentyp nicht gültig

=-2 : Mittelwertkennung ungültig

FUNKTION

Der Typ der Mittelwertbildung ist für alle Stoffkonstanten und Stoff-Funktionen auf Position 3 des Steuerblockes gespeichert.

Durch Aufruf von T_ASK_SINGLE bzw. T_ASK_PURE wird der gültige Zeiger für den Stoffdatentyp bestimmt. Nachdem mit T_ASK_AVER sichergestellt ist, daß der Mittelwerttyp gültig ist, wird die Speicherung vorgenommen.

T_ASK_AVER

Kennung für Mittelwertbildung prüfen

FORMAT

T_ASK_AVER(AVERTYP, NUMMER, RVAL)

Name	Typ	Zugriff
AVERTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

AVERTYP

Kennung des Mittelwertbildungstyps

NUMMER

Nummer des Typs

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : angegebener Typ unbekannt

FUNKTION

T_ASK_AVER prüft den angegebenen Mittelwertbildungstyp auf Gültigkeit.

Der Parameter NAVER gibt die Anzahl der bekannten Typen an.

T_ASK_ASPEZ

Prüfen der Kennung für spezielle Stoffwerttypen wie Gasdichte

FORMAT

T_ASK_ASPEZ(PROPTYP, NUMMER, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
NUMMER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Kennung des Stoffwerts

NUMMER

Nummer des Typs

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : angegebener Typ unbekannt

FUNKTION

T_ASK_ASPEZ prüft, ob es sich bei dem angegebenen Stoffwerttyp um einen Typ handelt, der nicht über die Reinstoffmittelwertbildung berechnet wird.

Der Parameter NASPEZ gibt die Anzahl der bekannten Typen an.

T_P_ASPEZ

Speichert die Methode zur Mittelwertbildung für Stoffdaten, die nicht nach Reinstoffmethoden behandelt werden.

FORMAT

T_P_ASPEZ(PROPTYP, METHVEK, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
METHVEK	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Kennung des Stoffwerts

METHVEK

Vektor mit den Methoden

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Speicherplatz erschöpft

=-1: PROPTYP nicht gültig

FUNKTION

T_P_ASPEZ speichert die Berechnungsmethode auf den für den Stoffdatentyp gültigen Zeiger.

Es gibt 3 Steuergrößen:

Typkennung	C4
Anzahl Methoden	I4
Kennung, ob Basisversion	I4

T_G_AVER

Holen der Methode für die Mittelwertbildung

FORMAT

T_G_AVER(PROPTYP, AVERTYP, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
AVERTYP	C	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

AVERTYP

Typ der Mittelwertbildung

RVAL

Fehlerkennung

= 0 : alles o.K.

=-1 : Stoffdatentyp nicht gültig

=-2 : Stoffdatentyp noch nicht gespeichert

FUNKTION

Für jeden Stoffdatentyp (Konstanten und Funktionen) ist im Steuerblock als 3. Größe die Mittelwertbildung gespeichert.

Ist der Stoffdatentyp identifiziert, wird diese Größe auf AVERTYP übertragen.

T_G_ASPEZ

Holt die Methode zur Mittelwertbildung für Stoffdaten, die nicht nach Reinstoffmethoden behandelt werden.

FORMAT

T_G_ASPEZ(PROPTYP, METHVEK, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
METHVEK	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Kennung des Stoffwerts

METHVEK

Vektor mit den Methoden

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : keine Methode gespeichert

=-1: PROPTYP nicht gültig

FUNKTION

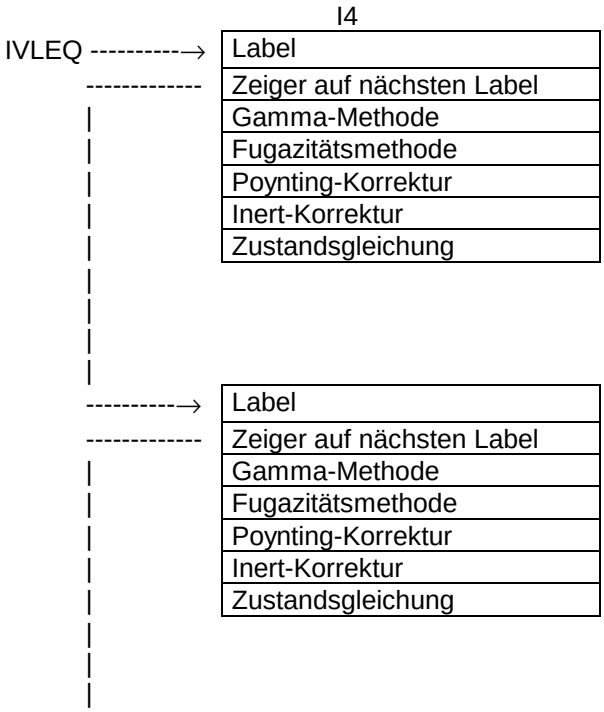
T_G_ASPEZ holt die Berechnungsmethode(n) aus dem internen Speicher.

5.2 Flüssig-Dampf-Gleichgewichts-Label

Für die Speicherung der Phasengleichgewichtslabel für Flüssig-Dampf-Gleichgewichte steht der Zeiger IVLEQ im Common /PROP_ZEIGER/.
Dieser Zeiger weist auf 4-Byte-Größen im Common /PROP_DATA/.
Vorgelagert sind 2 Steuergrößen mit folgender Belegung:

C4	I4
LVEQ	Versionskennung = 0: Basisversion = 1: höhere Version

Der Zeiger weist auf Speicherblöcke, in denen die Daten wie folgt abgelegt sind:



Für die Verwaltung von Flüssig-Dampf-Phasengleichgewichtslabeln sind folgende Unterprogramme vorhanden:

- T_P_LVEQ
  - T_P_POINTER
- T_G_LVEQ
- T_S_LVEQ

T_P_LVEQ

Speicherung von Labels zur Phasengleichgewichtsberechnung Flüssig-Dampf

FORMAT

T_P_LVEQ(LABEL, METHODEN, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
METHODEN	C(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

METHODEN

Vektor der zu speichernden Methodenkennungen

(1) = Aktivitätsmethode

(2) = Fugazitätsmethode

(3) = Poynting-Korrektur

(4) = Inert-Korrektur

(5) = Zustandsgleichungen

RVAL

=0 : alles o.K.

=1 : Speicherplatz erschöpft

FUNKTION

T_P_LVEQ speichert einen Label zur Beschreibung der Phasengleichgewichtsberechnung Flüssig-Dampf.

Ist bisher noch keine Methode gespeichert (Zeiger IVLEQ <= 0), wird der Speicherplatz mit dem UP T_P_POINTER angelegt und die Daten wie oben beschrieben gespeichert. Der Verkettungszeiger wird dabei auf 0 gesetzt, um das Ende der Verkettung anzuzeigen.

Sind bereits Label gespeichert, ist zu unterscheiden, ob ein bereits gespeicherter Label überschrieben werden oder ein neuer angehängt werden soll.

Beim Überschreiben werden der Label und die dazugehörenden Methodenkennungen neu gespeichert; die Verkettung braucht nicht verändert zu werden.

Beim Neuspeichern wird der Speicherplatz neu angelegt. Die Position wird als Verkettungszeiger in den letzten gespeicherten Block eingetragen. Die Daten auf die neue Position gespeichert und der Verkettungszeiger als Kennung des letzten Blockes auf 0 gesetzt.

T_G_LVEQ

Holen der Methoden für Flüssig-Dampf-Gleichgewichte

FORMAT

T_G_LVEQ(TASK, LABEL, METHODEN, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen/Schreiben
METHODEN	C(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1: Hole ersten Label mit Aufschlüsselung

=2: Hole nächsten Label mit Aufschlüsselung

=3: Hole Aufschlüsselung zu vorgegebenem Label

LABEL

Labelname

wird geschrieben bei TASK=1,2

wird gelesen bei TASK=3

METHODEN

Methodenkennung für die Gleichgewichtsberechnung

(1) = Aktivitätsmethode

(2) = Fugazitätsmethode

(3) = Poynting-Korrektur

(4) = Inert-Kennungen

(5) = Zustandsgleichungen

RVAL

Fehlerkennung

= 0: alles o.K.

= 1: kein Label gespeichert oder "NONE"-Label

=-1: bei TASK=3 : Label nicht gefunden

bei TASK=2 : kein weiterer Label gespeichert

=-2: fehlerhafte TASK

FUNKTION

Ausgehend vom Zeiger IVLEQ wird die Verkettung der Label je nach Aufgabenstellung verfolgt und die gewünschte Information zurückgegeben.

Ist kein Label gespeichert (IVLEQ <= 0) oder ist die Methodenkennung im Label auf NONE gesetzt, wird RVAL=1 gesetzt. Dies ist die Information für die übergeordneten Programme, daß kein Gleichgewicht gerechnet werden kann.

T_S_LVEQ

Aktivieren eines Labels zur Berechnung von Flüssig-Dampf-Gleichgewichten

FORMAT

T_S_LVEQ(LABEL, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Name des zu aktivierenden Labels

RVAL

Fehlerkennung

=0 : alles o.K.

=1 : Label nicht gefunden

FUNKTION

T_S_LVEQ verfolgt die Verkettung der gespeicherten Flüssig-flüssig-Phasengleichgewichtslabel.

Wird der Label gefunden, wird er im Common /PROP_THERMO/ als aktuelle Methode gespeichert. (LVEQ_LABEL)

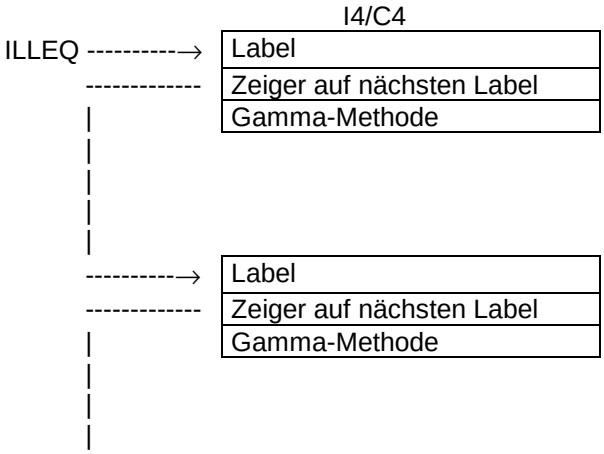
Wird er nicht gefunden, wird die RVAL-Kennung gesetzt; der aktuelle Inhalt von LVEQ_LABEL wird nicht geändert.

5.3 Flüssig-Flüssig-Gleichgewichts-Label

Für die Speicherung der Phasengleichgewichtslabel für Flüssig-Flüssig-Gleichgewichte steht der Zeiger ILLEQ im Common /PROP_ZEIGER/.
Dieser Zeiger weist auf 4-Byte-Größen im Common /PROP_DATA/.
Vorgelagert sind 2 Steuergrößen mit folgender Belegung:

C4	I4
LLEQ	Versionskennung = 0: Basisversion = 1: höhere Version

Der Zeiger weist auf Speicherblöcke, in denen die Daten wie folgt abgelegt sind:



Für die Verwaltung von Flüssig-flüssig-Phasengleichgewichtslabeln sind folgende Unterprogramme vorhanden:

- T_P_LLEQ
  - T_P_POINTER
- T_G_LLEQ
- T_S_LLEQ

T_P_LLEQ

Speichern eines Labels zur Phasengleichgewichtsberechnung Flüssig-flüssig

FORMAT

T_P_LLEQ(LABEL, METHODEN, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
METHODEN	C(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

METHODEN

Vektor der zu speichernden Methodenkennungen

(1) : Aktivitätsmethode

RVAL

=0 : alles o.K.

=1 : Speicherplatz erschöpft

FUNKTION

T_P_LLEQ speichert einen Label zur Beschreibung der Flüssig-flüssig Phasengleichgewichtsberechnung.

Ist bisher noch keine Methode gespeichert (Zeiger ILLEQ <= 0), wird der Speicherplatz mit dem UP T_P_POINTER angelegt und die Daten wie oben beschrieben gespeichert. Der Verkettungszeiger wird dabei auf 0 gesetzt, um das Ende der Verkettung anzuzeigen.

Sind bereits Label gespeichert, ist zu unterscheiden, ob ein bereits gespeicherter Label überschrieben oder ein neuer angehängt werden soll.

Beim Überschreiben werden der Label und die dazugehörenden Methodenkennungen neu gespeichert; die Verkettung braucht nicht verändert zu werden.

Beim Neuspeichern wird der Speicherplatz neu angelegt. Die Position wird als Verkettungszeiger in den letzten gespeicherten Block eingetragen. Die Daten auf die neue Position gespeichert und der Verkettungszeiger als Kennung des letzten Blockes auf 0 gesetzt.

T_G_LLEQ

Methoden für FlÜssig-Flüssig-Gleichgewichte holen

FORMAT

T_G_LLEQ(TASK, LABEL, METHODEN, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen/Schreiben
METHODEN	C(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1: Hole ersten Label mit Aufschlüsselung

=2: Hole nächsten Label mit Aufschlüsselung

=3: Hole Aufschlüsselung zu vorgegebenem Label

LABEL

Labelname

wird geschrieben bei TASK=1,2

wird gelesen bei TASK=3

METHODEN

Methodenkennung für die Gleichgewichtsberechnung

(1) = Aktivitätsmethode

RVAL

Fehlerkennung

= 0: alles o.K.

= 1: kein Label gespeichert oder "NONE"-Label

=-1: bei TASK=3 : Label nicht gefunden

bei TASK=2 : kein weiterer Label gespeichert

=-2: fehlerhafte TASK

FUNKTION

Ausgehend vom Zeiger ILLEQ wird die Verkettung der Label je nach Aufgabenstellung verfolgt und die gewünschte Information zurückgegeben.

Ist kein Label gespeichert (ILLEQ <= 0) oder ist die Methodenkennung im Label auf NONE gesetzt, wird RVAL=1 gesetzt. Dies ist die Information für die übergeordneten Programme, daß kein Gleichgewicht gerechnet werden kann.

T_S_LLEQ

Aktivieren eines Labels zur Berechnung von Flüssig-flüssig-Gleichgewichten

FORMAT

T_S_LLEQ(LABEL, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

RVAL

Fehlerkennung

=0 : alles o.K.

=1 : Label nicht gefunden

FUNKTION

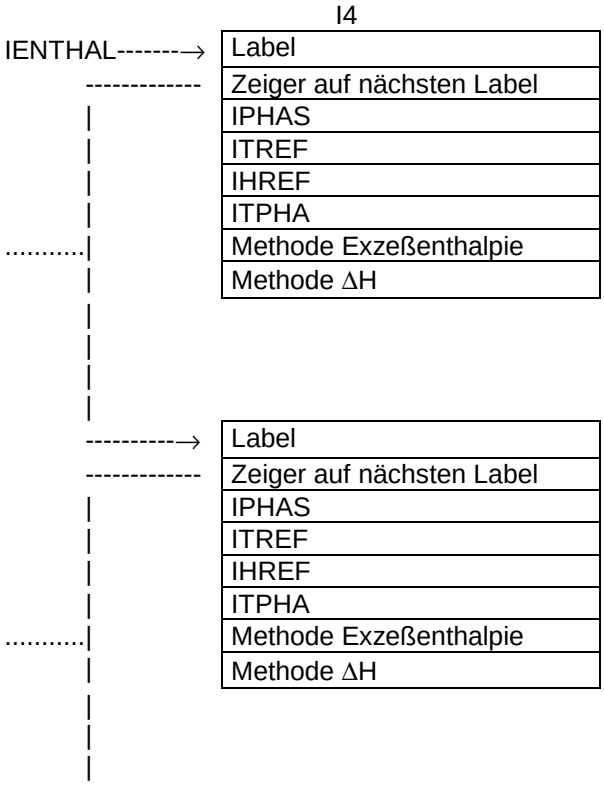
T_S_LLEQ verfolgt die Verkettung der gespeicherten Flüssig-flüssig-Phasengleichgewichtslabel.

Wird der Label gefunden, wird er im Common /PROP_THERMO/ als aktuelle Methode gespeichert. (LLEQ_LABEL)

Wird er nicht gefunden, wird die RVAL-Kennung gesetzt; der aktuelle Inhalt von LLEQ_LABEL wird nicht geändert.

5.4 Enthalpie-Label

Für die Speicherung der Enthalpie-Label steht der Zeiger IENTHAL im Common /PROP_ZEIGER/zur Verfügung.
Der Zeiger weist auf Speicherblöcke, in denen die Daten wie folgt abgelegt sind:



Hierbei bedeuten:

IPHAS Zeiger auf den Vektor der Referenzphasen; zeigt auf 4-Byte-Größen im Characterfeld des Daten-Common. Diese Größen werden standardmäßig mit "LIQU" initialisiert.

ITREF Zeiger auf den Vektor der Referenztemperaturen; zeigt auf 8-Byte-Größen des Daten-Common. Die Größen werden standardmäßig mit 273.15 Kelvin initialisiert.

IHREF Zeiger auf den Vektor der Referenzenthalpien; zeigt auf 8-Byte-Größen des Daten-Common. Die Größen werden standardmäßig mit 0 J/kmol initialisiert.

ITPHA Zeiger auf den Vektor der Phasenumwandlungstemperaturen und der von den Zustandsgrößen unabhängigen Summanden der Reinstoffenthalpie; zeigt auf 8-Byte-Größen des Daten-Common. Dieser Zeiger wird erst aktiviert, wenn die erste Umwandlungstemperatur angegeben wird. Die Standardinitialisierung für Komponenten, deren Phasenumwandlung bei Systemtemperatur geschehen soll, ist -1.D30 (NONVAL aus PROP_INCLUDE-Datei).

Für die Verwaltung von Enthalpielabeln sind folgende Unterprogramme vorhanden:

- T_P_LABEL_ENTH
  - T_P_POINTER
- T_P_PHASE_ENTH
  - T_FIND_ENTH_POSI
- T_P_HREF_ENTH
  - T_FIND_ENTH_POSI

- T_P_TREF_ENTH
  - T_FIND_ENTH_POSI
- T_P_TPHAS_ENTH
  - T_P_POINTER
  - T_FIND_ENTH_POSI
- T_P_EXCE_ENTH
  - T_FIND_ENTH_POSI
- T_P_ISOT_ENTH
  - T_FIND_ENTH_POSI
- T_G_ENTH
  - T_G_LABEL_ENTH
  - T_FIND_ENTH_POSI
- T_S_ENTH
  - T_FIND_ENTH_POSI

T_P_LABEL_ENTH

Speicherung von Labels zur Enthalpieberechnung

FORMAT

T_P_LABEL_ENTH(LABEL, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

RVAL

=0 : alles o.K.

=1 : Speicherplatz erschöpft

FUNKTION

T_P_LABEL_ENTH speichert einen Label zur Beschreibung der Enthalpieberechnung.

Ist bisher noch kein Label gespeichert (Zeiger IENTHAL <= 0), wird der Speicherplatz mit dem UP T_P_POINTER angelegt und die Standardinitialisierung wie oben beschrieben gespeichert. Hierbei werden auch die Methoden zur Berechnung der Exzeßenthalpie und der isothermen Druckabhängigkeit im Gas auf "NONE" gesetzt. Der Verkettungszeiger wird auf 0 gesetzt, um das Ende der Verkettung anzuzeigen.

Sind bereits Label gespeichert, ist zu unterscheiden, ob ein bereits gespeicherter Label überschrieben werden oder ein neuer angehängt werden soll.

Beim Überschreiben bleiben alle alten Daten erhalten; die Verkettung braucht nicht verändert zu werden.

Beim Neuspeichern wird der Speicherplatz neu angelegt. Die Position wird als Verkettungszeiger in den letzten gespeicherten Block eingetragen. Die neue Position wird wie beim ersten Label mit den Standardwerten initialisiert. Der Verkettungszeiger wird als Kennung des letzten Blockes auf 0 gesetzt.

T_P_PHASE_ENTH

Speicherung der Information über die Startphase des Enthalpieweges für einen Label zur Enthalpieberechnung

FORMAT

T_P_PHASE_ENTH(LABEL, COMP_NR, PHASE, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
COMP_NR	I4	Lesen
PHASE	C(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

COMP_NR

=0 : alle Komponenten

>0 : Nummer einer Komponente

PHASE

Vektor mit Kennungen der Startphase

'LIQU'	Flüssigkeit
'GAS'	Gas
'SOLI'	Feststoff

RVAL

=0 : alles o.K.

=1 : Label nicht gefunden

FUNKTION

T_P_PHASE_ENTH speichert zu einem Enthalpie-Label, der vorher mit dem Programm T_P_LABEL_ENTH angelegt wurde, die Informationen zur Startphase des Berechnungsweges.

Für COMP_NR = 0 ist PHASE als Vektor mit der Länge = der aktuellen Komponentenzahl einzugeben.

Für COMP_NR > 0 wird der Inhalt von PHASE(1) als Wert für diese Komponente gespeichert.

T_P_HREF_ENTH

Speicherung der Referenzenthalpien für einen Label zur Enthalpieberechnung

FORMAT

T_P_HREF_ENTH(LABEL, COMP_NR, HREF, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
COMP_NR	I4	Lesen
HREF	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

COMP_NR

=0 : alle Komponenten

>0 : Nummer einer Komponente

HREF

Referenzenthalpien

RVAL

=0 : alles O.K.

=1 : Label nicht gefunden

FUNKTION

T_P_HREF_ENTH speichert zu einem Enthalpie-Label, der vorher mit dem Programm T_P_LABEL_ENTH angelegt wurde, die Informationen über die Referenzenthalpien.

Für COMP_NR = 0 ist HREF als Vektor mit der Länge = der aktuellen Komponentenzahl einzugeben.

Für COMP_NR > 0 wird der Inhalt von HREF(1) als Wert für diese Komponente gespeichert.

T_P_TREF_ENTH

Speicherung der Referenztemperaturen für einen Label zur Enthalpieberechnung

FORMAT

T_P_TREF_ENTH(LABEL, COMP_NR, TREF, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
COMP_NR	I4	Lesen
TREF	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

COMP_NR

=0 : alle Komponenten

>0 : Nummer einer Komponente

TREF

Referenztemperaturen

RVAL

=0 : alles O.K.

=1 : Label nicht gefunden

FUNKTION

T_P_TREF_ENTH speichert zu einem Enthalpie-Label, der vorher mit dem Programm T_P_LABEL_ENTH angelegt wurde, die Informationen über die Referenztemperaturen.

Für COMP_NR = 0 ist TREF als Vektor mit der Länge = der aktuellen Komponentenzahl einzugeben.

Für COMP_NR > 0 wird der Inhalt von TREF(1) als Wert für diese Komponente gespeichert.

T_P_TPHAS_ENTH

Speicherung der Phasenumwandlungstemperaturen für einen Label zur Enthalpieberechnung

FORMAT

T_P_TPHAS_ENTH(LABEL, COMP_NR, TPHASE, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
COMP_NR	I4	Lesen
TPHASE	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

COMP_NR

=0 : alle Komponenten

>0 : Nummer einer Komponente

TPHASE

Phasenumwandlungstemperaturen

RVAL

=0 : alles O.K.

=1 : Speicherplatz erschöpft oder Label nicht gefunden

FUNKTION

T_P_TPHAS_ENTH speichert zu einem Enthalpie-Label, der vorher mit dem Programm T_P_LABEL_ENTH angelegt wurde, die Informationen über die Phasenumwandlungstemperaturen.

Für COMP_NR = 0 ist TPHASE als Vektor mit der Länge = der aktuellen Komponentenzahl einzugeben.

Für COMP_NR > 0 wird der Inhalt von TPHASE(1) als Wert für diese Komponente gespeichert.

Beim erstmaligen Aufruf von T_P_TPHASE_ENTH wird der Zeiger für die Umwandlungstemperaturen mit der Länge 2*aktuelle Komponentenzahl angelegt. Im ersten Vektor werden die eingegebenen Temperaturen gespeichert; der zweite Vektor dient zur Aufnahme der konstanten Terme des Enthalpieweges (im Bereich TREF bis Umwandlungstemperatur). Die Berechnung hierfür erfolgt im UP T_FI_ENTH vor Speicherung des Stoffsystems.

T_P_EXCE_ENTH

Speicherung der Methodenkennzeichnung zur Berechnung der Exzeßenthalpie für einen Label zur Enthalpieberechnung

FORMAT

T_P_EXCE_ENTH(LABEL, EXCESS, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
EXCESS	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

EXCESS

Methodenkennzeichnung zur Berechnung der Exzeßenthalpie

RVAL

=0 : alles O.K.

=1 : Label nicht gefunden

FUNKTION

T_P_EXCE_ENTH speichert zu einem Enthalpie-Label, der vorher mit dem Programm T_P_LABEL_ENTH angelegt wurde, die Informationen über die Methode zur Berechnung der Exzeßenthalpie.

T_P_ISOT_ENTH

Speicherung der Methodenkennzeichnung zur Berechnung der isothermen Druckabhängigkeit der Enthalpie im Gas für einen Label zur Enthalpieberechnung

FORMAT

T_P_ISOT_ENTH(LABEL, ISOT, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
ISOT	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Labelname

ISOT

Methodenkennzeichnung zur Berechnung der isothermen Druckabhängigkeit

RVAL

=0 : alles O.K.

=1 : Label nicht gefunden

FUNKTION

T_P_ISOT_ENTH speichert zu einem Enthalpie-Label, der vorher mit dem Programm T_P_LABEL_ENTH angelegt wurde, die Informationen über die Methode zur Berechnung der isothermen Druckabhängigkeit der Enthalpie im Gas.

T_G_ENTH

Holen der Daten zur Enthalpieberechnung

FORMAT

T_G_ENTH(TASK, LABEL, EXCESS, ISOT, PHASE, TREF, HREF, TPHASE, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen/Schreiben
EXCESS	C	Schreiben
ISOT	C	Schreiben
PHASE	C(*)	Schreiben
TREF	R8(*)	Schreiben
HREF	R8(*)	Schreiben
TPHASE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1: Hole ersten Label mit Aufschlüsselung

=2: Hole nächsten Label mit Aufschlüsselung

=3: Hole Aufschlüsselung zu vorgegebenem Label

=4: Hole Aufschlüsselung zu vorgegebenem Label einschließlich Hilfsvektoren

LABEL

Labelname

wird geschrieben bei TASK=1,2

wird gelesen bei TASK=3,4

EXCESS

Methodenkennung für die Berechnung der Exzeßenthalpie

ISOT

Methodenkennung für die Berechnung der isothermen Druckabhängigkeit

PHASE

Kennung für die Startphase des Berechnungsweges

TREF

Referenztemperaturen

HREF

Referenzenthalpien

TPHASE

Phasenumwandlungstemperaturen

für TASK=4: Phasenumwandlungstemperaturen und Teilsummen der Reinstoffenthalpie im Bereich TREF bis TPHASE

RVAL

Fehlerkennung

= 0: alles O.K.

= 1: kein Label gespeichert

= 2: keine Phasenumwandlungstemperaturen gegeben; alle Werte auf -1.D30 gesetzt

=-1: bei TASK=3 : Label nicht gefunden

bei TASK=2 : kein weiterer Label gespeichert

=-2: fehlerhafte TASK

FUNKTION

T_G_ENTH ruft für TASK=1 oder TASK=2 das UP T_G_LABEL_ENTH, das den Label einer gespeicherten Enthalpiebeschreibung zurückgibt.

Für TASK= 3 oder TASK=4 wird mit dem UP T_FIND_ENTH_POSI geprüft, ob der angegebene Label gespeichert ist.

Mit dem UP T_FIND_ENTH_POSI werden die Positionen der Enthalpiedaten bestimmt und dann aus dem Daten-Common in die Parametergrößen umgespeichert.

Für TASK=4 werden zusätzlich die mit T_FI_ENTH berechneten Daten zurückgegeben.

Die Dimension der Vektoren muß mindestens der aktuellen Komponentenzahl entsprechen. Die Länge der Character muß mindestens 4 sein, um alle Informationen eindeutig aufnehmen zu können.

T_G_LABEL_ENTH

Holen der Labels zur Enthalpieberechnung

FORMAT

T_G_LABEL_ENTH(TASK, LABEL, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1: Hole ersten Label

=2: Hole nächsten Label

LABEL

Labelname

RVAL

Fehlerkennung

= 0: alles o.K.

= 1: kein Label gespeichert

=-1: bei TASK=2 : kein weiterer Label gespeichert

=-2: fehlerhafte TASK

FUNKTION

Das UP T_G_LABEL_ENTH gibt den Label einer gespeicherten Enthalpiebeschreibung zurück.

Es verfolgt die Verkettung der Label beginnend mit dem Startzeiger IENTHAL im Common /PROP_ZEIGER/.

T_FIND_ENTH_POSI

Finden von Positionen von Enthalpielabeln bzw. Positionen der Daten innerhalb eines Enthalpielabels

FORMAT

T_FIND_ENTH_POSI(TASK, LABEL, POSITION, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen
POSITION	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1: Position des Labels

=2: Position des Phasenvektors

=3: Position des Vektors der Referenztemperaturen

=4: Position des Vektors der Referenzenthalpien

=5: Position des Vektors der Phasenumwandlungstemperaturen

=6: Position der Methode für die Exzeßenthalpie

=7: Position der Methode für die isotherme Druckabhängigkeit

LABEL

Labelname

POSITION

Zeiger

bei TASK = 1 auf IFELD

bei TASK = 2, 6, 7 auf C4FELD

bei TASK = 3, 4, 5 auf R8FELD

RVAL

Fehlerkennung

= 0: alles o.K.

= 1: kein Label gespeichert

= 2: bei TASK = 5: Zeiger noch nicht gesetzt

FUNKTION

Das UP T_FIND_ENTH_POSI verfolgt die Verkettung der Enthalpie-Label.

Je nach TASK wird POSITION entsprechend der Speicherung des Steuerblockes gesetzt.

T_S_ENTH

Aktivieren eines Labels zur Berechnung von Enthalpien

FORMAT

T_S_ENTH(LABEL, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Name des zu aktivierenden Labels

RVAL

Fehlerkennung

=0 : alles o.K.

=1 : Label nicht gefunden

FUNKTION

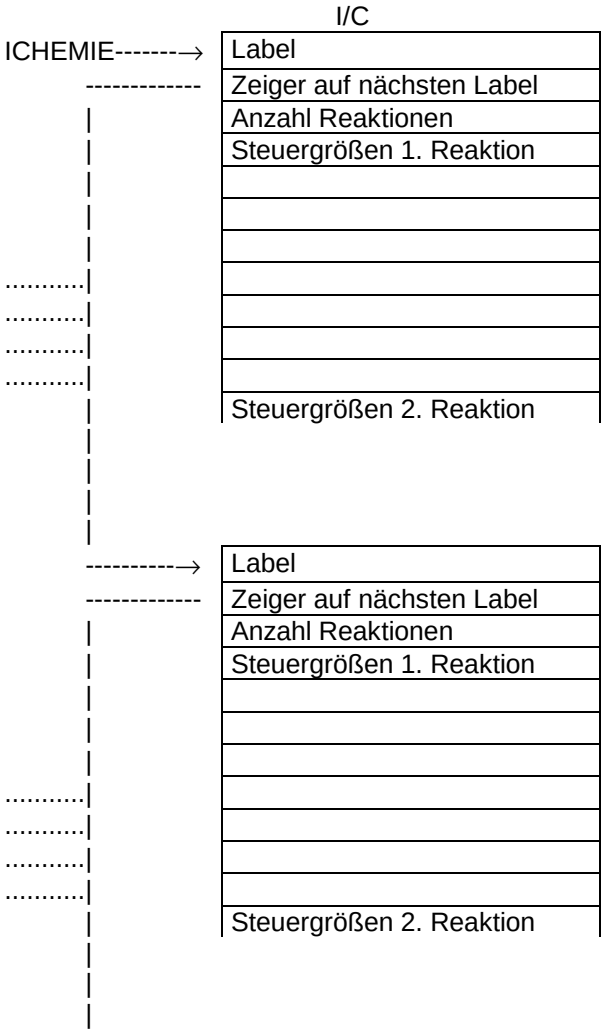
Durch einen Aufruf von T_FIND_ENTH_POSI wird geprüft, ob der gewünschte Enthalpie-Label gespeichert ist.

Wird der Label gefunden, wird er im Common /PROP_THERMO/ als aktuelle Methode gespeichert. (ENTH_LABEL)

Wird er nicht gefunden, wird die RVAL-Kennung gesetzt; der aktuelle Inhalt von ENTH_LABEL wird nicht geändert.

5.5 Chemieblöcke

Für die Speicherung der Chemieblocklabels steht der Zeiger ICHEMIE im Common /PROP_ZEIGER/ zur Verfügung.
Der Zeiger weist auf Speicherblöcke, in denen für jeden Chemieblock CHEMFEST (z.Zt. = 3) feste Größen und Anzahl Reaktionen * CHEMNUM (z.Zt. = 8) Steuergrößen gespeichert sind.



Die Belegung der Steuergrößen ist abhängig vom Reaktionstyp und in 3 Gruppen aufgeteilt.

Für Gleichgewichtsreaktionen EQxx wird wie folgt gespeichert:

I / C
Reaktionstyp
Phase
Zeiger auf die stöchiometrischen Koeff.
Zeiger auf Reaktionsenthalpie
Zeiger auf die Koeff. der F(T)-Funktion
Anzahl Koeff. der F(T)-Funktion
nicht benutzt
nicht benutzt

Für kinetisch kontrollierte Reaktionen KIxx wird wie folgt gespeichert:

I / C
Reaktionstyp
Phase
Zeiger auf die stöchiometrischen Koeff.
Zeiger auf Reaktionsenthalpie
Zeiger auf die Daten der F(T)-Funktionen und Alpha-Werte
Zeiger auf die Label der F(T)-Funktionen
Zeiger auf die Daten der Phi-Funktionen und Gamma-Werte
Zeiger auf die Label der Phi-Funktionen

Für KI-Reaktionen können max. 6 F(T)- bzw. 6 Phi-Funktionen gespeichert werden.

Für allg. Reaktionen (CONV, STAT, COOR) wird wie folgt gespeichert:

I / C
Reaktionstyp
nicht benutzt
Zeiger auf die stöchiometrischen Koeff.
Zeiger auf Reaktionsenthalpie und Umsatz bzw. Reaktionsrate
für CONV, STAT Nr. der Referenzkomponente
nicht benutzt
nicht benutzt
nicht benutzt

Für die Verwaltung von Chemieblocklabeln sind folgende Unterprogramme vorhanden:

- T_P_NO_REAC
- T_G_NO_REAC
- T_P_CH_LABEL
  - T_P_POINTER
- T_P_CH_TYPE
  - T_FIND_CHEM_POSI
  - T_ASK_CHEM
- T_P_CH_STOE
  - T_FIND_CHEM_POSI
- T_P_CH_DHR
  - T_FIND_CHEM_POSI
- T_P_CH_CONV
  - T_FIND_CHEM_POSI
- T_P_CH_FVT
  - T_FIND_CHEM_POSI

    - T_ASK_CHEM
    - T_P_POINTER
- T_P_CH_KI_FVT
    - T_FIND_CHEM_POSI
    - T_ASK_CHEM
    - T_P_POINTER
- T_P_CH_PHI
    - T_FIND_CHEM_POSI
    - T_ASK_CHEM
    - T_P_POINTER
- T_P_CH_ZETA
    - T_FIND_CHEM_POSI
- T_G_CHEMICS
- T_G_REACTION
    - T_FIND_CHEM_POSI
- T_G_CH_ALLG
    - T_FIND_CHEM_POSI
- T_G_CH_EQ
    - T_FIND_CHEM_POSI
- T_G_CH_KI
    - T_FIND_CHEM_POSI
- T_G_CH_PHASE
    - T_FIND_CHEM_POSI
- T_G_CH_TYPE
    - T_FIND_CHEM_POSI
- T_S_CHEMICS
    - T_FIND_CHEM_POSI

T_P_NO_REAC

Speichern der maximalen Anzahl Reaktionen, die in einem Chemieblock definiert werden können

FORMAT

T_P_NO_REAC(NUMBER, RVAL)

Name	Typ	Zugriff
NUMBER	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

NUMBER

Maximale Anzahl Reaktionen

RVAL

Fehlerkennung

= 0 : alles o.K.

FUNKTION

Die Daten werden im Common /PROP_ALLGEMEIN/ abgelegt.

T_G_NO_REAC

Holen der maximalen Anzahl Reaktionen, die in einem Chemieblock definiert werden können

FORMAT

T_G_NO_REAC(NUMBER, RVAL)

Name	Typ	Zugriff
NUMBER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

NUMBER

Maximale Anzahl Reaktionen

RVAL

Fehlerkennung

= 0 : alles o.K.

FUNKTION

Die Daten werden aus dem Common /PROP_ALLGEMEIN/ geholt.

T_P_CH_LABEL

Anlegen des Speicherplatzen für einen neuen Chemieblock und Speichern der Steuerdaten

FORMAT

T_P_CH_LABEL(LABEL, REAC_NO, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NO	I4	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NO

Maximale Anzahl Reaktionen

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Speicherplatz erschöpft

=-1: Label ist gespeichert, aber gespeicherte Reaktionszahl ungleich REAC_NO

=-2: Versionsnummer ungleich 1

=-3: keine chemischen Reaktionen vorgesehen

FUNKTION

T_P_CH_LABEL speichert und verwaltet die Steuerdaten für Chemieblöcke.

Chemieblocklabel werden nur für Basisversionen der Stoffsysteme gespeichert. Ist ein Stoffsystem mit höherer Versionsnummer aktiviert, wird nicht gespeichert und das Programm mit einer Fehlerkennung verlassen.

Ist bisher noch kein Label gespeichert (Zeiger ICHEMIE <= 0), wird der Speicherplatz für die allgemeinen Größen und die Steuergrößen der einzelnen Reaktionen mit dem UP T_P_POINTER angelegt. Der Verkettungszeiger wird auf 0 gesetzt, um das Ende der Verkettung anzuzeigen.

Sind bereits Label gespeichert, ist zu unterscheiden, ob ein bereits gespeicherter Label überschrieben oder ein neuer angehängt werden soll. Ist der Label bereits gespeichert, mup dessen Reaktionszahl mit der eingegebenen übereinstimmen.

Beim Überschreiben bleiben alle alten Daten erhalten; die Verkettung braucht nicht verändert zu werden.

Beim Neuspeichern wird der Speicherplatz angelegt. Die Position wird als Verkettungszeiger in den letzten gespeicherten Block eingetragen. Der Verkettungszeiger wird als Kennung des letzten Blockes auf 0 gesetzt. Außerdem werden die Zeiger der für alle Reaktionen gleich behandelten Daten gesetzt: für stöchiometrische Koeffizienten der Länge aktuelle Komponentenzahl, für Reaktionswärmen und spezielle doppelt genau zu speichernde Größen 2 Doppelworte.

T_P_CH_TYPE

Speichern des Reaktionstyps und der Reaktionsphase für eine Reaktion

FORMAT

T_P_CH_TYPE(LABEL, REAC_NR, TYPE, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
TYPE	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

TYPE

Reaktionstyp

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Label nicht gespeichert

=-1 : ungültiger Reaktionstyp

FUNKTION

T_P_CH_TYPE speichert den Reaktionstyp für die angegebene Reaktion.

Der Reaktionstyp ist als erste beschreibende Größe für eine Reaktion.

Aus dem Reaktionstyp ergibt sich, identifiziert durch einen Aufruf von T_ASK_CHEM, die Phase, in der die Reaktion laufen soll. Diese wird als 2. Steuergröße bei EQ- und KI-Reaktionen gespeichert.

T_FIND_CHEM_POSI

Finden von Positionen von Chemieblocklabeln bzw. Positionen der Daten innerhalb einer Reaktion eines Chemieblockes

FORMAT

T_FIND_CHEM_POSI(TASK, LABEL, REAC_NR, POSITION, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen
REAC_NR	I4	Lesen
POSITION	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1 : Position des Labels holen

=2 : Position einer Reaktion holen

=3 : Position des Reaktionstyps holen

=4 : Position der Reaktionsphase holen

=5 : Position der stöchiometrischen Koeffizienten holen

=6 : Position der Reaktionswärme holen

=7 : Position der F(T)-Funktion holen

=8 : Position des Umsatzes bzw. der Reaktionsrate holen

=9 : Position der Phi-Funktion holen

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

POSITION

Speicherposition der entsprechenden Date im Datenfeld

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Label nicht gespeichert

=-1 : Position noch nicht definiert

FUNKTION

T_FIND_CHEM_POSI verfolgt die Verkettung der Chemielabel und identifiziert innerhalb eines Chemieblockes die gewünschten Speicherpositionen.

T_P_CH_STOE

Speichern der stöchiometrischen Koeffizienten für eine Reaktion

FORMAT

T_P_CH_STOE(LABEL, REAC_NR, COMP_NR, STOE, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
COMP_NR	I4	Lesen
STOE	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

COMP_NR

Komponentennummer

= 0 : alle Komponenten

> 0 : Nummer der Komponente

STOE

stöchiometrischen Koeffizienten

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Label nicht gespeichert

FUNKTION

T_P_CH_STOE speichert die zu einer Reaktion gehörenden stöchiometrischen Koeffizienten.

Für COMP_NR > 0 wird STOE(1) für die angegebene Komponente gespeichert. Ist COMP_NR = 0 angegeben, muß die Länge von STOE der aktuellen Komponentenzahl entsprechen; es wird für alle Komponenten gespeichert.

Im Steuerblock einer Reaktion ist als 3. Größe der Zeiger auf die stöchiometrischen Koeffizienten dieser Reaktion gespeichert. T_FIND_CHEM_POSI liefert diese Position, die bereits im UP T_P_CH_LABEL gesetzt wird.

T_P_CH_DHR

Speichern der Reaktionwärme für eine Reaktion

FORMAT

T_P_CH_DHR(LABEL, REAC_NR, DHR, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
DHR	R8	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Nummer der Reaktion

DHR

Reaktionswärme

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Chemieblock nicht gefunden

=-1 : ungültiger Reaktionstyp

FUNKTION

T_P_CH_DHR speichert für eine Reaktion die Reaktionswärme.

Der Zeiger für die doppelt genau zu speichernde Reaktionswärme ist als 4. Größe im Steuerblock abgelegt. Dieser zeiger wird bereits im UP T_P_CH_LABEL aktiviert.

T_P_CH_CONV

Speichern der Umsatzes für eine Reaktion vom Typ CONV/STATE

FORMAT

T_P_CH_CONV(LABEL, REAC_NR, REF_COMP, CONV, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
REF_COMP	I4	Lesen
CONV	R8	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Nummer der Reaktion

REF_COMP

Nummer der Referenzkomponente

CONV

Umsatz

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Chemieblock nicht gefunden

=-1 : ungültiger Reaktionstyp

FUNKTION

T_P_CH_CONV speichert für eine Reaktion den Umsatz und die Referenzkomponente.

Der Umsatz wird als 2. doppeltgenau zu speichernde Größe über den Zeiger für die Reaktionswärmen abgelegt. Dieser Zeiger wird bereits im UP T_P_CH_LABEL aktiviert, da die Reaktionswärme für jeden Reaktionstyp zur Verfügung steht.

Die Referenz-(Bezugs-) Komponente wird als 5. Größe im Steuerblock gespeichert.

T_P_CH_FVT

Speichern der Koeffizienten der F(T)-Funktion für eine Gleichgewichtsreaktion

FORMAT

T_P_CH_FVT(LABEL, REAC_NR, COEFF_NO, FVT, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
COEFF_NO	I4	Lesen
FVT	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

COEFF_NO

Anzahl Koeffizienten der F(T)-Funktion

F(T)

Koeffizienten der F(T)-Funktion

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Label nicht gespeichert

FUNKTION

T_P_CH_FVT speichert die zu einer Gleichgewichtsreaktion gehörenden Koeffizienten der F(T)-Funktion und deren Anzahl.

Im Steuerblock einer Reaktion ist als 5. Größe der Zeiger auf die Koeffizienten der F(T)-Funktion dieser Reaktion gespeichert, als 6. Größe die Anzahl der Koeffizienten. Ist der Zeiger noch nicht gesetzt (RVAL < 0 im UP T_FIND_CHEM_POSI), so wird dieser mit dem UP T_P_POINTER auf eine maximale Länge von 6 Koeffizienten angelegt.

T_ASK_CHEM

Identifizieren des Reaktionstyps

FORMAT

T_ASK_CHEM(TYP, NUM, RVAL)

Name	Typ	Zugriff
TYP	C	Lesen
NUM	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

Typ

Reaktionstyp

NUM

Kennziffer des Reaktionstyps

> 200 : kinetisch kontrollierte Reaktion

> 100 : Gleichgewichtsreaktion

> 0 : allgemeine Reaktion

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : ungültige Kennung

FUNKTION

T_ASK_CHEM identifiziert den Reaktionstyp.

Reaktionen werden in 3 verschiedene Typklassen aufgeteilt:

1. Gleichgewichtsreaktionen mit den Kennungen EQLM, EQVM, EQLA, EQVP, EQLF, EQVF, EQLC, EQVC
2. kinetisch kontrollierte Reaktionen mit den Kennungen KILM, KIVM, KILC, KIVC, KILW, KIVW
3. allgemeine Reaktionen mit den Kennungen CONV, STAT, COOR

Die Initialisierung im Programm erfolgt so, daß Reaktionen in der Flüssigphase eine ungerade Kennziffer, Reaktionen in der Dampfphase eine gerade Kennziffer erhalten.

T_P_CH_KI_FVT

Speichern der Koeffizienten der F(T)-Funktion bzw. der Alpha-Werte für eine kinetisch kontrollierte Reaktion

FORMAT

T_P_CH_KI_FVT(TASK, LABEL, REAC_NR, FVT_LABEL, COMP_NR, FAK, VEKTOR, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen
REAC_NR	I4	Lesen
FVT_LABEL	C	Lesen
COMP_NR	I4	Lesen
FAK	R8	Lesen
VEKTOR	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1 : F(T) speichern

=2 : Alpha speichern

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

FVT_LABEL

Label der F(T)-Funktion

COMP_NR

für TASK = 2 : Komponentennummer

= 0 : alle Alpha-Werte speichern

> 0 : Alpha-Wert für die angegebene Komponente speichern

FAK

Vorzeichenfaktor für die F(T)-Funktion:

1.D0 für die Hinreaktion

-1.D0 für die Rückreaktion

VEKTOR

je nach TASK Koeffizienten der F(T)-Funktion oder Alpha-Werte

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Label nicht gespeichert

=-1 : ungültiger Reaktionstyp

=-2 : Anzahl Label erschöpft

FUNKTION

T_P_CH_KI_FVT speichert die zu einer kinetisch kontrollierten Reaktion gehörenden Koeffizienten der F(T)-Funktion und die dazu gehörenden Alpha-Werte.

Die Zusammengehörigkeit von Koeffizienten und Alpha-Werten wird durch den F(T)-Label sichergestellt.

Es können z. Zt. max. 6 Funktionen gespeichert werden.(Größe CHEMFUNC im COMMON /PROP_ALLGEMEIN/, initialisiert im UP T_INIT).

Im Steuerblock einer kinetisch kontrollierten Reaktion ist als 5. Größe der Zeiger auf die Werte dieser Reaktion gespeichert, als 6. Größe der Zeiger auf die Label der Funktionen. Ist dieser Zeiger noch nicht gesetzt (RVAL < 0 im UP T_FIND_CHEM_POSI) , wird er durch einen Aufruf von T_P_POINTER auf die Maximallänge (5+Anz. Komp.) * max. Anz. Funktionen gesetzt. Der Zeiger auf die Label wird auf maximale Anz. Funktionen + 1 gesetzt und mit Blank vorinitialisiert.(Abfragemöglichkeit auf Blank, um die Anzahl der gespeicherten Funktionen festzustellen).

Je Label werden gespeichert: Faktor, a, b, c, d, Alpha(i)

T_P_CH_PHI

Speichern der Koeffizienten der PHI-Funktion bzw. die Gamma-Werte für eine kinetisch kontrollierte Reaktion

FORMAT

T_P_CH_PHI(TASK, LABEL, REAC_NR, PHI_LABEL, COMP_NR, VEKTOR, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen
REAC_NR	I4	Lesen
PHI_LABEL	C	Lesen
COMP_NR	I4	Lesen
VEKTOR	R8(*)	Lesen
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1 : PHI speichern

=2 : Gamma speichern

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

PHI_LABEL

Label der PHI-Funktion

COMP_NR

für TASK = 2 : Komponentennummer

= 0 : alle Gamma-Werte speichern

> 0 : Gamma-Wert für die angegebene Komponente speichern

VEKTOR

je nach TASK Koeffizienten der PHI-Funktion oder Gamma-Werte

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Label nicht gespeichert

=-1 : ungültiger Reaktionstyp

=-2 : Anzahl Label erschöpft

FUNKTION

T_P_CH_PHI speichert die zu einer kinetisch kontrollierten Reaktion gehörenden Koeffizienten der PHI-Funktion und die dazu gehörenden Gamma-Werte.

Die Zusammengehörigkeit von Koeffizienten und Gamma-Werten wird durch den PHI-Label sichergestellt.

Es können z. ZT. max. 6 Funktionen gespeichert werden.(Größe CHEMFUNC im COMMON /PROP_ALLGEMEIN/, initialisiert im UP T_INIT).

Im Steuerblock einer kinetisch kontrollierten Reaktion ist als 7. Größe der Zeiger auf die Werte dieser Reaktion gespeichert, als 8. Größe der Zeiger auf die Label der Funktionen. Ist dieser Zeiger noch nicht gesetzt (RVAL < 0 im UP T_FIND_CHEM_POSI) , wird er durch einen Aufruf von T_P_POINTER auf die Maximallänge (4+Anz. Komp.) * max. Anz. Funktionen gesetzt. Der Zeiger auf die Label wird auf maximale Anz. Funktionen + 1 gesetzt und mit Blank vorinitialisiert.(Abfragemöglichkeit auf Blank, um die Anzahl der gespeicherten Funktionen festzustellen).

Je Label werden gespeichert: a, b, c, d, Gamma(i)

T_P_CH_ZETA

Speichern der gewünschten Reaktionsrate für eine Reaktion vom Typ COOR.

FORMAT

T_P_CH_ZETA(LABEL, REAC_NR, ZETA, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
ZETA	R8	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

ZETA

Reaktionsrate

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Label nicht gespeichert

FUNKTION

T_P_CH_ZETA speichert die zu einer Reaktion vom Typ COOR gehörende Reaktionsrate.

Die Reaktionsrate wird als 2. doppeltgenau zu speichernde Größe über den Zeiger für die Reaktionswärmen abgelegt. Dieser Zeiger wird bereits im UP T_P_CH_LABEL aktiviert, da die Reaktionswärme für jeden Reaktionstyp zur Verfügung steht.

T_G_CHEMICS

Label und Anzahl Reaktionen für Chemieblöcke holen

FORMAT

T_G_CHEMICS(TASK, LABEL, NUMBER, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen/Schreiben
NUMBER	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1 : Hole ersten Label mit Reaktionszahl

=2 : Hole nächsten Label mit Reaktionszahl

=3 : Hole Reaktionszahl zu vorgegebenem Label

LABEL

Label des Chemieblockes

NUMBER

Anzahl der Reaktionen des Chemieblockes

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : kein Chemielabel gespeichert

=-1 : bei TASK = 3: Label nicht gefunden

bei TASK = 2: kein weiterer Label gefunden

=-2 : fehlerhafte TASK

FUNKTION

Das UP T_G_CHEMICS gibt den Label und die Anzahl Reaktionen eines gespeicherten Chemieblockes zurück.

Es verfolgt die Verkettung der Label beginnend mit dem Startzeiger ICHEMIE im Common /PROP_ZEIGER/.

T_G_CH_ALLG

Daten für allgemeine Reaktionen holen

FORMAT

T_G_CH_ALLG(LABEL, REAC_NR, VALUE, COMP_NR, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
VALUE	R8	Schreiben
COMP_NR	I4	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Nummer der Reaktion

VALUE

Für Reaktionen vom Typ COOR : Reaktionsrate

Für Reaktionen vom Typ CONV/STAT : Umsatz

COMP_NR

Nummer der Referenzkomponente für CONV/STAT

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : kein Chemielabel gespeichert

=-1 : ungültiger Reaktionstyp

FUNKTION

Das UP T_G_CH_ALLG gibt die für allgemeine Reaktionen spezifischen Daten zurück.

T_G_CH_EQ

Spezielle Daten für Gleichgewichtsreaktionen holen

FORMAT

T_G_CH_EQ(LABEL, REAC_NR, COEFF_NO, COEFFICIENTS, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
COEFF_NO	I4	Schreiben
COEFFICIENTS	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

COEFF_NO

Anzahl gespeicherter Koeffizienten

COEFFICIENTS

Koeffizienten der F(T)-Funktion, gefüllt von 1 bis COEFF_NO

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Chemielabel nicht gespeichert

=-1 : ungültiger Reaktionstyp

FUNKTION

Das UP T_G_CH_EQ gibt für Gleichgewichtsreaktionen die Koeffizienten und die Anzahl Koeffizienten der F(T)-Funktion zurück.

T_G_CH_KI

spezielle Daten für kinetisch kontrollierte Reaktionen holen

FORMAT

T_G_CH_KI(TASK, LABEL, REAC_NR, FUNC_LABEL, FAK, FUNC, VECTOR, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
LABEL	C	Lesen
REAC_NR	I4	Lesen
FUNC_LABEL	C	Schreiben
FAK	R8	Schreiben
FUNC	R8(*)	Schreiben
VECTOR	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1 : Hole erste F(T)-Funktion mit Alpha-Werten

=2 : Hole nächste F(T)-Funktion mit Alpha-Werten

=3 : Hole erste PHI-Funktion mit Gamma-Werten

=4 : Hole nächste PHI-Funktion mit Gamma-Werten

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

FUNC_LABEL

Label der Funktion

FAK

Faktor bei F(T); 1 für Hinreaktionen, -1 für Rückreaktionen

FUNC

Koeffizienten der Funktion, Länge: 4

VECTOR

Vektor mit den Alpha- bzw. Gamma-Werten

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : kein Chemielabel gespeichert

=-1 : ungültiger Reaktionstyp

=-2 : ungültige Aufrufreihenfolge

=-3 : kein weiterer Label gespeichert

FUNKTION

Das UP T_G_CH_KI gibt die speziell für kinetisch kontrollierte Reaktionen gespeicherten Daten zurück.

T_G_CH_PHASE

Holen der Phasenkennung für eine Reaktion im gerade aktiven Chemieblock

FORMAT

T_G_CH_PHASE(NUMBER, PHASE, RVAL)

Name	Typ	Zugriff
NUMBER	I4	Lesen
PHASE	C	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

NUMBER

Nummer der Reaktion dieses Chemieblockes

PHASE

Reaktionphase

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Label nicht gefunden

FUNKTION

Die für die Reaktion mit der angegebenen Nummer im gerade aktiven Chemieblock gespeicherte Reaktionsphase ('LIQU' oder 'VAPO') wird zurückgegeben.

T_G_CH_TYPE

Holen von Reaktionstyp und Phase

FORMAT

T_G_CH_TYPE(LABEL, REAC_NR, TYP, PHASE, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
REAC_NR	I4	Lesen
TYP	C	Schreiben
PHASE	C	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Label des Chemieblockes

REAC_NR

Reaktionsnummer

TYP

Reaktionstyp

PHASE

Reaktionsphase

RVAL

Fehlerkennung

= 0 : alles o.K.

= 1 : Chemielabel nicht gespeichert

FUNKTION

Das UP T_G_CH_TYPE gibt für eine Reaktion den gespeicherten Reaktionstyp und die Kennung der Phase, in der die Reaktion laufen soll, zurück.

T_G_REACTION

Holen der allgemeinen, vom Reaktionstyp unabhängigen Daten, für eine Reaktion in einem Chemieblock

FORMAT

T_G_REACTION(LABEL, NUMBER, TYPE, PHASE, DHR, STOE, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
NUMBER	I4	Lesen
TYPE	C	Schreiben
PHASE	C	Schreiben
DHR	R8	Schreiben
STOE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Chemieblocklabel

NUMBER

Nummer der gewünschten Reaktion

TYPE

Reaktionstyp

PHASE

Reaktionsphase

DHR

Reaktionsenthalpie

STOE

Vektor der stöchiometrischen Koeffizienten

RVAL

Fehlerkennung

= 0 : alles o.K.

FUNKTION

Die Speicherpositionen der Daten werden über das Programm T_FIND_CHEM_POSI bestimmt.

Der Länge des Vektors STOE muß mindestens der aktuellen Komponentenzahl entsprechen, um alle stöchiometrischen Koeffizienten aufnehmen zu können.

Die Länge der Ausgabecharacter muß mindestens 4 sein, um alle Kennungen eindeutig umspeichern zu können.

T_S_CHEMICS

Aktivieren eines Chemieblockes

FORMAT

T_S_CHEMICS(LABEL, RVAL)

Name	Typ	Zugriff
LABEL	C	Lesen
RVAL	I4	Schreiben

ARGUMENTE

LABEL

Name des zu aktivierenden Chemieblockes

RVAL

Fehlerkennung

=0 : alles o.K.

=1 : Label nicht gefunden

FUNKTION

Durch einen Aufruf von T_FIND_CHEM_POSI wird geprüft, ob der gewünschte Chemieblock gespeichert ist.

Wird der Label gefunden, wird er im Common /PROP_THERMO/ als aktuelle Methode gespeichert. (CHEM_LABEL)

Wird er nicht gefunden, wird die RVAL-Kennung gesetzt; der aktuelle Inhalt von CHEM_LABEL wird nicht geändert.

6 Die Unterprogramme zur Berechnung

Die Berechnungsprogramme sind so aufgebaut, daß jeweils ein Unterprogramm die Berechnung der Werte und eventuell der Ableitungen übernimmt. Ein zweites Unterprogramm mit dem Postfix _DERIVATIVE liefert die Werte der Ableitungen, die intern gespeichert worden sind. Hier wird also nicht mehr gerechnet.

Der Parameter WHAT in der Parameterliste der Berechnungsprogramme steuert die Berechnung der Ableitungen. Die Kennungen, die in den Unterprogrammen gültig sind, werden je Unterprogramm in der Parameterlistenbeschreibung angegeben. Es ist die Angabe jeder Kombination dieser Kennungen möglich. Sollen keine Ableitungen gebildet werden, ist der Parameter WHAT = Blank zu setzen.

In den Programmen, die die Ableitungen zurückgeben, hat der Parameter WHAT die Länge 1. Hier führt jede Eingabe eines nicht definierten Zeichen zum Setzen der RVAL-Marke.

6.1 Stoff-Funktionen

Stoff-Funktionen werden im Unterprogramm T_PURE ausgewertet. z.Zt. sieht der Aufrufbaum wie folgt aus:

- T_PURE
  - T_CA_PURE
    - T_CA_PURE_FUNC
    - T_CA_PURE_EXTR
- T_PURE_DERIVATIVE

Die Funktionsunterprogramme werden sowohl in T_CA_PURE_FUNC als auch in T_CA_PURE_EXTR aufgerufen.

- T_CA_PURE_FUNC
  - T_WATSON
  - T_WAGNER
  - T_POLYNOM
  - T_EPOLYNOM
  - T_ANTOINE
  - T_ANT1
  - T_KHOFF
  - T_SUTHERLAND
  - T_CPL
  - T_ICPL
  - T_VISC
  - T_RACKETT
  - T_ALY_LEE
  - T_KHOFF1
  - T_DIP4
  - T_DIP5

Neue Programme zur Auswertung neuer Ansätze sind im UP T_CA_PURE_FUNC bzw. im Unterprogramm T_CA_PURE_EXTR aufzurufen.

Die Parameterliste der Berechnungsunterprogramme ist wie folgt festgelegt:

LTABL, COEFF_NO, COEFFICIENTS, T, VALUE, VAL_DERI

Name	Typ	Zugriff
LTABL	L	Lesen
Daten	R8	Lesen
COEFF_NO	I4	Lesen
COEFFICIENTS	R8(*)	Lesen
T	R8	Lesen
VALUE	R8	Schreiben
VAL_DERI	R8	Schreiben

ARGUMENTE

LTABL

=.TRUE. : T-Ableitung soll gebildet werden

Daten

Basisstoffdaten zur Berechnung (falls für die Gleichung erforderlich)

COEFF_NO

Anzahl der in COEFFICIENTS übergebenen Koeffizienten

COEFFICIENTS

Koeffizienten des Ansatzes

T

Temperatur in Kelvin

VALUE

Wert der Funktion

VAL_DERI

Wert der Ableitung

T_PURE

Berechnung von Reinstoffdaten als Funktion von T und/oder P

FORMAT

T_PURE(PROPTYP, WHAT, COMP_NR, T, P, VALUE, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
WHAT	C	Lesen
COMP_NR	I4	Lesen
T	R8	Lesen
P	R8	Lesen
VALUE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

COMP_NR

Komponentennummer

=0 : alle Komponenten

>0 : Nummer der gewünschten Komponente

T

Temperatur in Kelvin

P

Druck in Pascal

VALUE

Vektor mit den Funktionswerten

RVAL

Fehlerkennung

= 0 : alles ok.

=-1: Stoffdatum nicht gespeichert

=-2: ungültige Komponentennummer

=-3: Stoffdatentyp nicht gültig

=-4: Ansatz nicht identifizierbar in T_CA_PURE_FUNC

>0 : Stoffdatum für diese Komponente nicht gespeichert

>1000 : Temperaturbereich verletzt

FUNKTION

Für COMP_NR > 0 wird in VALUE der Funktionswert für die gewünschte Komponente zurückgegeben. Für COMP_NR > 0 werden die Funktionswerte und Ableitungen für alle Komponenten des aktuellen Stoffdatenblocks bestimmt.

Die gespeicherten Informationen werden durch Aufruf von T_G_PURE_COEFF abgefragt. Im Programm T_CA_PURE_FUNC werden die Funktionswerte und deren Ableitungen entsprechend dem gespeicherten Ansatztyp berechnet.

Zur Speicherung der Ableitungen stehen im Common /PROP_CALC_ZEIGER/ folgende R8-Zeiger, die Platz im Daten-Common belegen, zur Verfügung.

Name	Länge in Doppelworten	Nutzung für
IPURET	Anz. Komponenten	T-Ableitungen
IPUREP	Anz. Komponenten	P-Ableitungen

Die Vektoren R8FELD(IPURET) und R8FELD(IPUREP) werden für die Ableitungen mit -1.D30 vorinitialisiert. Diese Initialisierung ist die Kennzeichnung für das Programm T_PURE_DERIVATIVE, daß die Ableitungen nicht gebildet wurden. Wurden in T_CA_PURE_FUNC die entsprechenden Ableitungen gebildet, wird diese Vorinitialisierung überschrieben.

Bei Aufruf des Programms für alle Komponenten (COMP_NR > 0) werden bei nicht vorhandenem Stoffwerttyp (RVAL-Flagge in T_G_PURE_COEFF) alle Werte und Ableitungen = 0 gesetzt. Ist der Stoffwerttyp für einzelne Komponenten nicht gespeichert, wird für diese Komponenten ebenfalls 0 eingetragen.

Bei Berechnung für eine spezielle Komponente (COMP_NR > 0) wird der berechnete Wert in VALUE(1) zurückgegeben. In R8FELD(IPURET) und R8FELD(IPUREP) werden, falls gewünscht, die entsprechenden Ableitungen gespeichert. Um für das Programm T_PURE_DERIVATIVE zu kennzeichnen, daß T_PURE nur für eine bestimmte Komponente aufgerufen wurde, wird R8FELD(IPURET+1) und R8FELD(IPUREP+1) auf -1.D30 gesetzt.

Sind für diese Komponente keine Daten gespeichert, wird VALUE(1) auf 0 gesetzt; ebenso die Ableitungsgrößen, falls Ableitungen gewünscht waren.

T_CA_PURE_FUNC

Aufrufsteuerung zur Berechnung von Funktionswerten und Ableitungen für Reinstoff-Funktionen

FORMAT

T_CA_PURE_FUNC(WHAT, EQUATION, COEFF_NO, COEFFICIENTS, T, P, VALUE, VALUE_T, VALUE_P, RVAL)

Name	Typ	Zugriff
WHAT	C	Lesen
EQUATION	C	Lesen
COEFF_NO	I4	Lesen
COEFFICIENTS	R8(*)	Lesen
T	R8	Lesen
P	R8	Lesen
VALUE	R8	Schreiben
VALUE_T	R8	Schreiben
VALUE_P	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

EQUATION

Characterstring mit der Gleichungskennung

COEFF_NO

Anzahl der Koeffizienten

COEFFICIENTS

Koeffizienten

T

Temperatur in Kelvin

P

Druck in Pascal

VALUE

Wert der Funktion

VALUE_T

Wert der Temperatur-Ableitung

VALUE_P

Wert der Druck_Ableitung

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : ungültige Ansatzkennung

=-2 : Basisstoffdaten fehlen

FUNKTION

Je nach übergebener Gleichungskennung wird das entsprechende Berechnungsunterprogramm zur Berechnung des Wertes und der Ableitung aufgerufen.

Wird eine neue Reinstoff-Funktion implementiert, muß dies in einen neuen ELSE-Zweig eingebaut werden. Das Kürzel für den Gleichungstyp ist zusätzlich im Programm T_ASK_EQUA aufzunehmen. Soll von dieser Funktion auch der Integralwert bestimmt werden, ist der Aufruf zur Integralberechnung zusätzlich in T_CA_INT_PURE aufzunehmen.

T_PURE_DERIVATIVE

Rückgabe der Ableitungen von Reinstoffdaten

FORMAT

T_PURE_DERIVATIVE(WHAT, DERIVATIVE, RVAL)

Name	Typ	Zugriff
WHAT	C1	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

DERIVATIVE

Ableitung(en)

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_PURE gebildet und intern auf den Zeigern IPURET und IPUREP im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in R8FELD(IPURET) bedeutet, daß beim letzten Aufruf von T_PURE keine Ableitungen gebildet wurden. Der Wert -1.D30 in R8FELD(IPURET+1) bedeutet, daß beim letzten Aufruf von T_PURE der Wert nur für eine Komponente berechnet wurde.

6.2 Integrale von Stoff-Funktionen

Integrale von Stoff-Funktionen werden im Unterprogramm T_PURE_INTEGRAL ausgewertet. Z.Zt. sieht der Aufrufbaum wie folgt aus:

- T_PURE_INTEGRAL
  - T_CA_INT_PURE
    - T_INT_POLYNOM
    - T_INT_CPL
    - T_INT_ICPL
    - T_INT_ALY_LEE

Neue Programme zur Integralbildung neuer Ansätze sind also im UP T_CA_INT_PURE aufzurufen.

Die Parameterliste der Berechnungsunterprogramme ist wie folgt festgelegt:

COEFF_NO, COEFFICIENTS, T_UNTEN, T_OBEN, H_COEFF, VALUE

Name	Typ	Zugriff
COEFF_NO	I4	Lesen
COEFFICIENTS	R8(*)	Lesen
T_UNTEN	R8	Lesen
T_OBEN	R8	Lesen
H_COEFF	R8(*)	Schreiben
VALUE	R8	Schreiben

ARGUMENTE

COEFF_NO

Anzahl der in COEFFICIENTS übergebenen Koeffizienten

COEFFICIENTS

Koeffizienten des Ansatzes

T_UNTEN

Temperatur in Kelvin als untere Integralsgrenze

T_OBEN

Temperatur in Kelvin als obere Integralsgrenze

H_COEFF

Hilfsfeld der Länge max. Koeffizientenzahl

VALUE

Wert des Integrals

T_PURE_INTEGRAL

Berechnung des Integrals von Reinstoff-Funktionen

FORMAT

T_PURE_INTEGRAL(PROPTYP, COMP_NR, T_OBEN, T_UNTEN, P, VALUE, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
COMP_NR	I4	Lesen
T_OBEN	R8	Lesen
T_UNTEN	R8	Lesen
P	R8	Lesen
VALUE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

COMP_NR

Komponentennummer

=0 : alle Komponenten

>0 : Nummer der gewünschten Komponente

T_OBEN

Temperatur in Kelvin als obere Integralgrenze

T_UNTEN

Temperatur in Kelvin als untere Integralgrenze

P

Druck in Pascal

VALUE

Vektor der Funktionswerte

RVAL

Fehlerkennung

= 0 : alles ok.

=-1: Stoffdatum nicht gespeichert

=-2: ungültige Komponentennummer

=-3: Stoffdatentyp nicht gültig

=-4: Ansatz nicht identifizierbar in T_CA_INT_PURE

>0 : Stoffdatum für diese Komponente nicht gespeichert

FUNKTION

Für COMP_NR > 0 wird in VALUE der Integralwert für die gewünschte Komponente zurückgegeben. Für COMP_NR = 0 werden die Integralwerte für alle Komponenten des aktuellen Stoffdatenblocks bestimmt.

Die gespeicherten Informationen werden durch Aufruf von T_G_PURE_COEFF abgefragt. Im Programm T_CA_INT_PURE werden die Integralwerte entsprechend dem gespeicherten Ansatztyp berechnet.

Bei Aufruf des Programms für alle Komponenten (COMP_NR > 0) werden bei nicht vorhandenem Stoffwerttyp (RVAL-Flagge in T_G_PURE_COEFF) alle Werte gesetzt. Ist der Stoffwerttyp für einzelne Komponenten nicht gespeichert, wird für diese Komponenten ebenfalls 0 eingetragen.

Bei Berechnung für eine spezielle Komponente (COMP_NR > 0) wird der berechnete Wert in VALUE(1) zurückgegeben.

Sind für diese Komponente keine Daten gespeichert, wird VALUE(1) auf 0 gesetzt; ebenso die Ableitungsgrößen, falls Ableitungen gewünscht waren.

T_CA_INT_PURE

Aufrufsteuerung zur Berechnung von Integralen für Reinstoff-Funktionen

FORMAT

T_CA_INT_PURE(EQUATION, COEFF_NO, COEFFICIENTS,
T_UNTEN, T_OBEN, P, VALUE, RVAL)

Name	Typ	Zugriff
EQUATION	C	Lesen
COEFF_NO	I4	Lesen
COEFFICIENTS	R8(*)	Lesen
T_UNTEN	R8	Lesen
T_OBEN	R8	Lesen
P	R8	Lesen
VALUE	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

EQUATION

Characterstring mit der Gleichungskennung

COEFF_NO

Anzahl der Koeffizienten

COEFFICIENTS

Koeffizienten

T_UNTEN

Temperatur in Kelvin als untere Integralgrenze

T_OBEN

Temperatur in Kelvin als obere Integralgrenze

P

Druck in Pascal

VALUE

Wert des Integrals

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : ungültige Ansatzkennung

FUNKTION

Je nach übergebener Gleichungskennung wird das entsprechende Berechnungsunterprogramm zur Berechnung des Integrals aufgerufen.

Wird eine neue Reinstoff-Funktion implementiert, muß dies in einen neuen ELSE-Zweig eingebaut werden. Das Kürzel für den Gleichungstyp ist zusätzlich im Programm T_ASK_EQUA aufzunehmen und die Verzweigung im UP T_CA_PURE_FUNC einzubauen.

6.3 Mittelwertbildungen

Mittelwerte werden mit dem Unterprogramm T_AVER berechnet. Der Aufrufbaum sieht z. Zt. wie folgt aus:

- T_AVER
  - T_CA_AVER
    - T_WILKE
    - T_WASSI_MASON
    - T_DIP_ST
    - T_DIP_KLIQ
- T_AVER_DERIVATIVE

Neue Programme zur Berechnung neuer Mittelwertbildungen sind im UP T_CA_AVER aufzurufen.

Die Parameterliste der Berechnungsunterprogramme ist wie folgt festgelegt:

NKOMP, WHAT, T, MOLES, Hilfsgrößen, STOFF, STOFF_T, VALUE, VALUE_T, VALUE_M, RVAL

Name	Typ	Zugriff
NKOMP	I4	Lesen
WHAT	C	Lesen
T	R8	Lesen
MOLES	R8(*)	Lesen
Hilfsgrößen	R8	Lesen
STOFF	R8(*)	Lesen
STOFF_T	R8(*)	Lesen
VALUE	R8	Schreiben
VALUE_T	R8	Schreiben
VALUE_M	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

NKOMP

Anzahl Komponenten

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

T

Temperatur in Kelvin

MOLES

Vektor mit den Molanteilen

Hilfsgrößen

Basisstoffdaten zur Berechnung (falls für die Gleichung erforderlich) und Felder, die als Zwischenspeicher benötigt werden

STOFF

Stoffwerte, für die der Mittelwert gebildet werden soll

STOFF_T

T-Ableitung der Stoffwerte, falls im Parameter WHAT gekennzeichnet ist, daß diese gebildet werden soll.

VALUE

Mittelwert

VALUE_T

T-Ableitung des Mittelwertes

VALUE_M

Vektor mit den M-Ableitungen des Mittelwertes

RVAL

Fehlerkennung

= 0 : alles ok.

= -1: Wert nicht berechenbar

T_AVER

Berechnung von Mittelwerten

FORMAT T_AVER(PROPTYP, WHAT, T, P, MOLES, VALUE, RVAL)

Name	Typ	Zugriff
PROPTYP	C	Lesen
WHAT	C	Lesen
T	R8	Lesen
P	R8	Lesen
MOLES	R8(*)	Lesen
VALUE	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

PROPTYP

Stoffdatentyp

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen

T

Temperatur in Kelvin

P

Druck in Pascal

MOLES

Molanteile der Komponenten in der Phase für die der Stoffwerttyp gilt

VALUE

Mittelwert

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : ungültiger Stoffwerttyp oder Stofftyp nicht gespeichert

FUNKTION

Die Berechnung der Mittelwerte und deren Ableitungen erfolgt im Moment komplett im UP T_CA_AVER.

Im Common /PROP_CALC_ZEIGER/ stehen für die Mittelwertbildung folgende Zeiger als Zwischenspeicherplatz zur Verfügung:

Name	Länge in Doppelworten	Nutzung für
IAVER	max. Anz. Komp. + 2	Ableitungen Temp. Druck Mole
IAV_PROP	max. Anz. Komp.	Reinstoffdaten
IAV_VEC	max. Anz. Komp.	T-Abl. der Reinst.
IAV_VEC1	max. Anz. Komp.	P-Abl. der Reinst.

Der Vektor R8FELD(IAVER) wird in den Positionen 1:3 mit -1.D30 vorinitialiert, um für T_AVER_DERIVATIVE zu kennzeichnen, daß die Ableitung nicht berechnet ist. Diese Werte werden überschrieben, sobald mit T_CA_AVER die Ableitung berechnet wurde.

T_CA_AVER

Mittelwerte und ihre Ableitungen berechnen

FORMAT

T_CA_AVER(NKOMP, AVERTYP, WHAT, T, P, MOLES, STOFF, T_STOFF, P_STOFF, VALUE, VALUE_T, VALUE_P, VALUE_M, RVAL)

Name	Typ	Zugriff
NKOMP	I4	Lesen
AVERTYP	C	Lesen
WHAT	C	Lesen
T	R8	Lesen
P	R8	Lesen
MOLES	R8(*)	Lesen
STOFF	R8(*)	Lesen
T_STOFF	R8(*)	Lesen
P_STOFF	R8(*)	Lesen
VALUE	R8	Schreiben
VALUE_T	R8	Schreiben
VALUE_P	R8	Schreiben
VALUE_M	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

NKOMP

Anzahl Komponenten

AVERTYP

Kennung der Mittelwertbildung

WHAT

Ableitungskennung

T : T-Ableitung bilden

P : P-Ableitung bilden

M : M-Ableitung bilden

T

Temperatur in Kelvin

P

Druck in Pascal

MOLES

Molanteile der Komponenten in der Phase für die der Stoffwerttyp gilt

STOFF

Vektor mit den Reinstoffdaten

STOFF_T

Vektor mit den T-Ableitungen der Reinstoffdaten

STOFF_P

Vektor mit den P-Ableitungen der Reinstoffdaten

VALUE

Mittelwert

VALUE_T

T-Ableitung des Mittelwertes

VALUE_P

P-Ableitung des Mittelwertes

VALUE_M

Vektor mit den Ableitungen des Mittelwertes nach den Molanteilen

RVAL

Fehlerkennung

= 0 : alles ok.

=-1 : AVERTYP nicht gültig

FUNKTION

T_CA_AVER berechnet je nach angegebenem AVERTYP den Mittelwert und falls gewünscht dessen Ableitungen.

T_AVER_DERIVATIVE

Rückgabe der Ableitungen von Mittelwerten

FORMAT

T_AVER_DERIVATIVE(WHAT, DERIVATIVE, RVAL)

Name	Typ	Zugriff
WHAT	C1	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen

DERIVATIVE

Ableitung(en)

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_AVER gebildet und intern auf dem Zeiger IAVER im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in R8FELD(IAVER), R8FELD(IAVER+1), R8FELD(IAVER+2) bedeutet, daß beim letzten von T_AVER keine Ableitungen gebildet wurden.

6.4 Flüssig-Dampf-Gleichgewichte

K-Werte werden mit dem Unterprogramm T_LVEQ berechnet. Der Aufrufbaum sieht z.Zt. wie folgt aus:

- T_LVEQ
  - T_G_LVEQ
  - T_ZUST
    - T_CA_RKS
      - T_RKS
        - T_RKS_ALPHA
        - T_RKS_P0
          - T_RKS_ZREIN
            - T_IK_CARDAN
        - T_RKS_PLV
          - T_RKS_ZMIXT
            - T_IK_CARDAN
    - T_CA_PR
      - T_PR
        - T_PR_ALPHA
        - T_PR_PH0
          - T_PR_ZREIN
            - T_PR_CARDAN
        - T_PR_PHLV
          - T_PR_ZMIXT
            - T_PR_CARDAN
  - T_PURE
  - T_PURE_DERIVATIVE
  - T_GAMMA
    - T_CA_UNIQUAC
      - T_UNIQUAC
    - T_CA_NRTL
      - T_NRTL
    - T_CA_WILSON
      - T_WILSON
    - T_CA_FLORY
      - T_FLORY
    - T_CA_UNFAC
      - T_UNIFAC
  - T_CA_HENRY
    - T_HENRY
  - T_CA_POYNT
    - T_POYNT
  - T_FUGA
    - T_CA_RKS
      - T_RKS
        - T_RKS_ALPHA
        - T_RKS_P0
          - T_RKS_ZREIN

- T_IK_CARDAN
- T_RKS_PLV
- T_RKS_ZMIXT
- T_IK_CARDAN
- T_CA_PR
- T_PR
- T_PR_ALPHA
- T_PR_PH0
- T_PR_ZREIN
- T_PR_CARDAN
- T_PR_PHLV
- T_PR_ZMIXT
- T_PR_CARDAN
- T_CA_VIRI
- T_VIRI
- T_VIR_P0
- T_VIR_PV
- T_CA_PMER
- T_PMER
- T_PM_PH0
- T_PM_ZIZREIN
- T_GAUSS
- T_PM_PHV
- T_PM_ZI_ZMIXT
- T_GAUSS
- T_GAMMA_DERIVATIVE
- T_HENRY_DERIVATIVE
- T_POYNT_DERIVATIVE
- T_FUGA_DERIVATIVE
- T_LVEQ_DERIVATIVE
- T_GAMMA_DERIVATIVE
- T_FUGA_DERIVATIVE

Neue Programme zur **Berechnung von Aktivitätskoeffizienten** sind also im UP T_GAMMA in der Form

- T_CA_Methode
  - Methode

aufzurufen. Das T_CA_Methode-Programm hat Zugriff auf die internen Datenstrukturen und Include-Dateien der Schnittstelle. Das Methode-Unterprogramm ist hingegen völlig unabhängig von Strukturen der Schnittstelle und erhält seine Informationen nur aus der Parameter-Liste.

Die Parameterliste des T_CA_Methode-Programms ist wie folgt festgelegt:

PHASE, WHAT, NKOMP, T, LIQUID, GAMMA, GAMMA_DT, GAMMA_DX

Name	Typ	Zugriff
PHASE	I4	Lesen
WHAT	C	Lesen
NKOMP	I4	Lesen
T	R8	Lesen
LIQUID	R8	Lesen
GAMMA	R8(*)	Schreiben
GAMMA_DT	R8(*)	Schreiben
GAMMA_DX	R8(NKOMP,*)	Schreiben

ARGUMENTE

PHASE

Kennung, für welches Gleichgewicht die Aktivitätskoeffizienten berechnet werden sollen.

=1 : flüssig-Dampf

=2 : flüssig-flüssig

Dieser Parameter kann entfallen, wenn der Aufruf nur für ein bestimmtes Gleichgewicht möglich ist.

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'X' : Ableitung nach den Molanteilen

NKOMP

Anzahl Komponenten

T

Temperatur in Kelvin

LIQUID

Molanteile in der Flüssigkeit

GAMMA

Aktivitätskoeffizienten

GAMMA_DT

Temperatur-Ableitung der Aktivitätskoeffizienten

GAMMA_DX

Matrix der Ableitungen der Aktivitätskoeffizienten nach den Molanteilen

Die **Parameterliste des Methode-Programms** ist wie folgt aufgebaut:

WHAT, allg. Größen, NKOMP, T, LIQUID, Stoffdaten, Hilfsvektoren, Hilfsmatrizen, GAMMA, GAMMA_DT, GAMMA_DX

Name	Typ	Zugriff
WHAT	C	Lesen
allg. Größen		Lesen
NKOMP	I4	Lesen
T	R8	Lesen
LIQUID	R8	Lesen
Stoffdaten	R8	Lesen
Hilfsvektoren	R8	Schreiben
Hilfsmatrizen	R8	Schreiben
GAMMA	R8(*)	Schreiben
GAMMA_DT	R8(*)	Schreiben
GAMMA_DX	R8(NKOMP,*)	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'X' : Ableitung nach den Molanteilen

allg. Größen

allgemeine Größen sind z.B. die Gaskonstante oder der Wert des maximalen Exponenten der e-Funktion, die in der Parameter-Includedatei der Schnittstelle gespeichert sind.

NKOMP

Anzahl Komponenten

T

Temperatur in Kelvin

LIQUID

Molanteile in der Flüssigkeit

Stoffdaten

alle für die Berechnung nach dieser Methode notwendigen Stoffdaten

Hilfsvektoren

Vektoren für Zwischenergebnisse

Hilfsmatrizen

Matrizen für Zwischenergebnisse

GAMMA

Aktivitätskoeffizienten

GAMMA_DT

Temperatur-Ableitung

GAMMA_DX

Matrix der Ableitungen nach den Molanteilen

Neue Programme zur **Berechnung von Fugazitätskoeffizienten oder k-Werten aus Zustandsgleichungen** sind im UP T_FUGA bzw. im UP T_ZUST in der Form

- T_CA_Methode
  - Methode

aufzurufen. Das T_CA_Methode-Programm hat Zugriff auf die internen Datenstrukturen und Include-Dateien der Schnittstelle. Das Methode-Unterprogramm ist hingegen völlig unabhängig von Strukturen der Schnittstelle und erhält seine Informationen nur aus der Parameter-Liste.

Die Parameterliste des T_CA_Methode-Programms ist wie folgt festgelegt:

ITASK, WHAT, NKOMP, T, P, LIQUID, VAPOR, PHI, PHI_T, PHI_P, PHI_X, PHI_Y, RVAL

Name	Typ	Zugriff
ITASK	I4	Lesen
WHAT	C	Lesen
NKOMP	I4	Lesen
T	R8	Lesen
P	R8	Lesen
LIQUID	R8(*)	Lesen
VAPOR	R8(*)	Lesen
PHI	R8(*)	Schreiben
PHI_T	R8(*)	Schreiben
PHI_P	R8(*)	Schreiben
PHI_X	R8(NKOMP,*)	Schreiben
PHI_Y	R8(NKOMP,*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

ITASK

Kennung, welche Art Fugazitätskoeffizienten berechnet werden sollen.

=1 : $\frac{\varphi_i^0}{\varphi_i^V}$

=2 : φ_i^V

=3 : φ_i^L

=4 : φ_i^0

=5 : $\frac{\varphi_i^L}{\varphi_i^V}$

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'X' : Ableitung nach den Molanteilen in der Flüssigkeit

'Y' : Ableitung nach den Molanteilen im Dampf

NKOMP

Anzahl Komponenten

T

Temperatur in Kelvin

P

Druck in Pascal

LIQUID

Molanteile in der Flüssigkeit

VAPOR

Molanteile im Dampf

PHI

Fugazitätskoeffizienten

PHI_T

Temperatur-Ableitung

PHI_P

Druck-Ableitung

PHI_X

Matrix der Ableitungen nach den Molanteilen in der Flüssigkeit
(wird je nach TASK bestimmt)

PHI_Y

Matrix der Ableitungen nach den Molanteilen im Dampf
(wird je nach TASK bestimmt)

RVAL

Fehlerkennung
=0: alles ok.
≠0: Fehlerkennung

Die **Parameterliste des Methode-Programms** ist wie folgt aufgebaut:

ITASK, WHAT, allg. Größen, NKOMP, T, P, LIQUID, VAPOR, Stoffdaten, Hilfsvektoren, Hilfsmatrizen, PHI, PHI_T, PHI_P, PHI_X, PHI_Y, RVAL

Name	Typ	Zugriff
ITASK	I4	Lesen
WHAT	C	Lesen
allg. Größen		Lesen
NKOMP	I4	Lesen
T	R8	Lesen
P	R8	Lesen
LIQUID	R8(*)	Lesen
VAPOR	R8(*)	Lesen
Stoffdaten	R8	Lesen
Hilfsvektoren	R8	Schreiben
Hilfsmatrizen	R8	Schreiben
PHI	R8(*)	Schreiben
PHI_T	R8(*)	Schreiben
PHI_P	R8(*)	Schreiben
PHI_X	R8(NKOMP,*)	Schreiben
PHI_Y	R8(NKOMP,*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

ITASK

Kennung, welche Art Fugazitätskoeffizienten berechnet werden sollen.

=1 : $\frac{\varphi_i^0}{\varphi_i^V}$

=2 : φ_i^V

=3 : φ_i^L

=4 : φ_i^0

=5 : $\frac{\varphi_i^L}{\varphi_i^V}$

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'X' : Ableitung nach den Molanteilen in der Flüssigkeit

'Y' : Ableitung nach den Molanteilen im Dampf

allg. Größen

allgemeine Größen sind z.B. die Gaskonstante oder der Wert des maximalen Exponenten der e-Funktion, die in der Parameter-Includedatei der Schnittstelle gespeichert sind.

NKOMP

Anzahl Komponenten

T

Temperatur in Kelvin

P

Druck in Pascal

LIQUID

Molanteile in der Flüssigkeit

VAPOR

Molanteile im Dampf

Stoffdaten

alle für die Berechnung nach dieser Methode notwendigen Stoffdaten

Hilfsvektoren

Vektoren für Zwischenergebnisse

Hilfsmatrizen

Matrizen für Zwischenergebnisse

PHI

Fugazitätskoeffizienten

PHI_T

Temperatur-Ableitung

PHI_P

Druck-Ableitung

PHI_X

Matrix der Ableitungen nach den Molanteilen in der Flüssigkeit

PHI_Y

Matrix der Ableitungen nach den Molanteilen im Dampf

RVAL

Fehlerkennung

=0 : alles ok.

≠0 : Fehlerkennung

T_LVEQ

Berechnung von Daten für Flüssig-Dampf-Gleichgewichte

FORMAT T_LVEQ(TASK, WHAT, T, P, LIQUID, VAPOR, VALUE, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
WHAT	C	Lesen
T	R8	Lesen
P	R8	Lesen
LIQUID	R8(*)	Lesen
VAPOR	R8(*)	Lesen
VALUE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

= 1 : k-Werte berechnen

= 2 : nur Aktivitätskoeffizienten berechnen

= 3 : nur Fugazitätskoeffizienten berechnen

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'X' : Ableitung nach den Molanteilen in der Flüssigkeit

'Y' : Ableitung nach den Molanteilen im Dampf

T

Temperatur in Kelvin

P

Druck in Pascal

LIQUID

Molanteile in der Flüssigkeit

VAPOR

Molanteile im Dampf

VALUE

berechnete Daten (je nach Aufgabenstellung)

RVAL

Fehlerkennung

= 0 : alles ok.

FUNKTION

Das Programm berechnet je nach Aufgabenstellung

- k-Werte : $k_i = \frac{H_i}{P} * \Phi_i * Fp_i$ oder

$$k_i = \frac{p^s{}_i}{P} * \gamma_i * \Phi_i * Fp_i \text{ oder}$$

$$k_i = \frac{\phi_i{}^L}{\phi_i{}^V}$$

- Aktivitätskoeffizienten : γ_i
- Henry-Koeffizienten : H_i
- Fugazitätskoeffizienten: Φ_i
- Poynting-Faktor : Fp

Die Methoden, nach denen gerechnet werden soll, werden durch Aufruf von T_G_LVEQ geholt. Die Einzelterme für die Berechnung werden durch Aufruf der Steuerprogramme der entsprechenden Aufgaben bestimmt.

Zur Speicherung der k-Werte und ihrer Ableitungen stehen in /PROP_CALC_ZEIGER/ folgende Zeiger zur Verfügung:

- IKVALUE - für die k-Werte
- IKVALUET - für die T-Ableitungen der k-Werte
- IKVALUEP - für die P-Ableitungen der k-Werte
- IKVALUEX - für die X-Ableitungen der k-Werte
- IKVALUEY - für die Y-Ableitungen der k-Werte

Bei Rechnung ohne Phasengleichgewicht werden die gewünschten Felder = 1, die Ableitungen = 0 gesetzt.

T_LVEQ_DERIVATIVE

Rückgabe der Ableitungen von k-Werten, Aktivitätskoeffizienten, Fugazitätskoeffizienten

FORMAT

T_LVEQ_DERIVATIVE(TASK, WHAT, COMP_NR, DERIVATIVE, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
WHAT	C1	Lesen
COMP_NR	I4	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

=1 : Holen der Ableitung für k-Werte

=2 : Holen der Ableitung für Aktivitätskoeffizienten

=3 : Holen der Ableitung für Fugazitätskoeffizienten

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'X' : Ableitung nach den Molanteilen in der Flüssigkeit

'Y' : Ableitung nach den Molanteilen im Dampf

COMP_NR

Nummer der Komponente, für die die X- bzw. Y-Ableitung als Vektor zurückgegeben werden soll.

DERIVATIVE

Ableitungen

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_LVEQ gebildet und intern auf den Zeigern IKVALUET, IKVALUEP, IKVALUEX, und IKVALUEY im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in der ersten Größe der Vektoren bedeutet, daß beim letzten Aufruf von T_LVEQ keine Ableitungen gebildet wurden.

Die Ableitungen der k-Werte werden direkt über die oben genannten Zeiger angesprochen. Die Ableitungen der Aktivitäts- bzw. Fugazitätskoeffizienten werden durch Aufruf der entsprechenden Ableitungsprogramme (T_GAMMA_DERIVATIVE; T_FUGA_DERIVATIVE) geholt.

6.5 Flüssig-flüssig-Gleichgewichte

Flüssig-flüssig-Gleichgewichte werden mit dem Programm T_LLEQ berechnet. Der Aufrufbaum sieht z.Zt. wie folgt aus:

- T_LLEQ
  - T_G_LLEQ
  - T_LL_GAMMA
    - T_CA_UNIQUAC
      - T_UNIQUAC
    - T_CA_NRTL
      - T_NRTL
    - T_CA_UNFAC
      - T_UNIFAC
- T_LLEQ_DERIVATIVE

T_LLEQ

Berechnung von Daten für Flüssig-Flüssig-Gleichgewichte

FORMAT

T_LLEQ(TASK, WHAT, T, P, LIQUID1, LIQUID2, VALUE1, VALUE2, RVAL)

Name	Typ	Zugriff
WHAT	C	Lesen
T	R8	Lesen
P	R8	Lesen
LIQUID1	R8(*)	Lesen
LIQUID2	R8(*)	Lesen
VALUE1	R8(*)	Schreiben
VALUE2	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'X' : Ableitung nach den Molanteilen in der Flüssigkeit

T

Temperatur in Kelvin

P

Druck in Pascal

LIQUID1

Molanteile in der 1. flüssigen Phase

LIQUID2

Molanteile in der 2. flüssigen Phase

VALUE1

Aktivitätskoeffizient in der 1. Phase

VALUE2

Aktivitätskoeffizient in der 2. Phase

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : alle Gammas = 1, da keine Methode gegeben

FUNKTION

Das Programm steuert die Berechnung der Aktivitätskoeffizienten der beiden flüssigen Phasen sowie deren Ableitungen.

Die Methoden, nach denen gerechnet werden soll, werden durch Aufruf von T_G_LLEQ geholt. Die Berechnung wird durch Aufruf des UP T_LL_GAMMA durchgeführt.

Zur Speicherung der k-Werte und ihrer Ableitungen stehen in /PROP_CALC_ZEIGER/ folgende Zeiger zur Verfügung:

- ILLGAM1 - für die Werte der 1. Phase
- ILLGAM2 - für die Werte der 2. Phase
- ILLGAM1T - für die T-Ableitungen der 1. Phase
- ILLGAM2T - für die T-Ableitungen der 2. Phase
- ILLGAM1X - für die X-Ableitungen der 1. Phase
- ILLGAM2X - für die X-Ableitungen der 2. Phase

Bei Rechnung ohne Phasengleichgewicht werden die gewünschten Felder = 1, die Ableitungen = 0 gesetzt.

T_LLEQ_DERIVATIVE

Rückgabe der Ableitungen der Aktivitätskoeffizienten

FORMAT

T_LLEQ_DERIVATIVE(WHAT, PHASE_NR, COMP_NR,
DERIVATIVE, RVAL)

Name	Typ	Zugriff
WHAT	C1	Lesen
PHASE_NR	I4	Lesen
COMP_NR	I4	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'X' : Ableitung nach den Molanteilen in der Flüssigkeit

PHASE_NR

Nummer der Phase, für die Ableitung als Vektor zurückgegeben werden soll.

COMP_NR

Nummer der Komponente, für die die X- bzw. Y-Ableitung als Vektor zurückgegeben werden soll.

DERIVATIVE

Ableitungen

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_LLEQ gebildet und intern auf den Zeigern ILLGAM1T, ILLGAM2T, ILLGAM1X, und ILLGAM2X im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in der ersten Größe der Vektoren bedeutet, daß beim letzten von T_LLEQ keine Ableitungen gebildet wurden.

6.6 Kompressibilitäten

Kompressibilitäten werden mit dem Programm T_COMPR berechnet. Der Aufrufbaum sieht z.Zt. wie folgt aus:

- T_COMPR
  - T_G_LVEQ
  - T_CA_ZRKS
    - T_RKS_ALPHA
    - T_RKS_ZMIXT
      - T_IK_CARDAN
    - T_RKS_ZREIN
      - T_IK_CARDAN
  - T_CA_ZPR
    - T_PR_ALPHA
    - T_PR_ZMIXT
      - T_PR_CARDAN
    - T_PR_ZREIN
      - T_PR_CARDAN
  - T_CA_ZVIRI
    - T_VIR_ZMIXT
  - T_CA_ZPMER
    - T_PM_ZMIXT
      - T_PM_ZI_ZMIXT
      - T_PM_Z_ZMIXT
    - T_PM_ZREIN
      - T_PM_ZIZREIN
      - T_PM_ZZREIN
- T_COMPR_DERIVATIVE

T_COMPR

Steuerprogramm zur Berechnung der Kompressibilität in der Gasphase

FORMAT

T_COMPR(METHOD, WHAT, T, P, VAPOR, Z, RVAL)

Name	Typ	Zugriff
METHOD	C	Lesen
WHAT	C	Lesen
T	R8	Lesen
P	R8	Lesen
VAPOR	R8(*)	Lesen
Z	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

METHOD

Kennung der Methode, nach der die Kompressibilität berechnet werden soll.

Sonderfall : METHOD = 'VLEQ' bedeutet, daß die Methode aus der Zustandsgleichungs- bzw. Fugazitätskennung des gerade gültigen Flüssig-Dampf-Phasengleichgewichtslabels bestimmt werden soll.

WHAT

Steuerparameter mit der Kennung, welche Ableitungen gebildet werden sollen.

'T' : Temperaturableitung

'P' : Druckableitung

'Y' : Ableitung nach den Molanteilen

T

Temperatur in Kelvin

P

Druck in Pascal

VAPOR

Molanteile im Dampf

Z

berechnete Kompressibiltität

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Z=1., alle Ableitungen = 0

=-1 : Methodenkennzeichnung nicht gültig

FUNKTION

Das Unterprogramm T_COMPR steuert die Berechnung die Berechnung der Kompressibilität.

Gerechnet wird entweder nach der in der Parameterliste angegebenen Methode oder bei Angabe der kennung 'VLEQ' nach der im gerade aktiven Berechnungslabel für Flüssig-Dampf-Gleichgewichte vorgegebenen Methode für Zustandsgleichungen bzw. Fugazitätskoeffizienten. Wird keine Methode gefunden, so wird Z = 1 gesetzt und alle Ableitungen = 0.

Ist im Parameter WHAT gekennzeichnet, daß die partiellen Ableitungen gebildet werden sollen, so werden diese direkt mitberechnet und zur Abfrage mit dem Programm T_COMPR_DERIVATIVE zur Verfügung gestellt.

Zur Speicherung der Ableitungen stehen im Common /PROP_CALC_ZEIGER/ folgende Zeiger zur Verfügung:

ICOMPRTP für T- und P-Ableitung

ICOMPRY für Y-Ableitungen

T_COMPR_DERIVATIVE

Rückgabe der Ableitungen der Kompressibilität

FORMAT

T_COMPR_DERIVATIVE(WHAT, DERIVATIVE, RVAL)

Name	Typ	Zugriff
WHAT	C1	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, welche Ableitung zurückgegeben werden soll.

'T' : Temperaturableitung

'P' : Druckableitung

'X' : Ableitung nach den Molanteilen

DERIVATIVE

enthält die gewünschte Ableitung

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Ableitung wurde nicht gebildet

=-1 : ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_COMPR gebildet und intern auf den Zeigern ICOMPRTP und ICOMPRY im Daten-Common doppelt genau gespeichert. Der Wert -1.D30 kennzeichnet, daß beim letzten Aufruf vom T_COMPR keine Ableitungen gebildet wurden.

6.7 Enthalpie

Enthalpien werden mit dem Programm T_ENTH berechnet. Der Aufrufbaum sieht z.Zt. wie folgt aus:

- T_ENTH
  - T_ENTH_LIQUID
    - T_G_ENTH
    - T_PURE_INTEGRAL
    - T_PURE
    - T_PURE_DERIVATIVE
    - T_H_ISO
      - T_CA_H_RKS
        - T_H_RKSMIXT
          - T_RKS_ZMIXT
            - T_IK_CARDAN
        - T_H_RKSREIN
          - T_RKS_ZREIN
            - T_IK_CARDAN
      - T_CA_H_PR
        - T_H_PR_MIXT
          - T_PR_ZMIXT
            - T_PR_CARDAN
        - T_H_PR_REIN
          - T_PR_ZREIN
            - T_PR_CARDAN
      - T_CA_HPMER
        - T_H_PM_MIXT
          - T_PM_ZI_ZMIXT
            - T_GAUSS
          - T_PM_H_ZMIXT
        - T_H_PM_REIN
          - T_PM_ZIZREIN
            - T_GAUSS
          - T_PM_HZREIN
    - T_H_ISO_DERIVATIVE
    - T_H_EX
      - T_CA_H_NRTL
        - T_H_NRTL
      - T_CA_H_UNIQUAC
        - T_H_UNIQUAC
      - T_CA_H_WILSON
        - T_H_WILSON
      - T_CA_H_FLORY
        - T_H_FLORY
      - T_CA_H_HMIX
        - T_H_MISCH
    - T_H_EX_DERIVATIVE
  - T_ENTH_VAPOR

- T_G_ENTH
- T_PURE_INTEGRAL
- T_PURE
- T_PURE_DERIVATIVE
- T_H_ISO
  - T_CA_H_RKS
    - T_H_RKSMIXT
      - T_RKS_ZMIXT
      - T_RKS_T_ABL
      - T_RKS_M_ABL
    - T_H_RKSREIN
      - T_RKS_ZREIN
  - T_CA_H_PR
    - T_H_PR_MIXT
      - T_PR_ZMIXT
      - T_PR_T_ABL
      - T_PR_S_ABL
    - T_H_PR_REIN
      - T_PR_ZREIN
  - T_CA_HPMER
    - T_H_PM_MIXT
      - T_PM_ZI_ZMIXT
        - T_GAUSS
      - T_PM_H_ZMIXT
    - T_H_PM_REIN
      - T_PM_ZIZREIN
        - T_GAUSS
      - T_PM_HZREIN
  - T_H_ISO_DERIVATIVE
- T_ENTH_SOLID
  - T_G_ENTH
  - T_PURE_INTEGRAL
  - T_PURE
- T_ENTH_DERIVATIVE

Neue Programme zur **Berechnung von Exzeßenthalpien** sind also im UP T_H_EX in der Form

- T_CA_H_Methode
  - T_H_Methode

aufzurufen. Das T_CA_H_Methode-Programm hat Zugriff auf die internen Datenstrukturen und Include-Dateien der Schnittstelle. Das T_H_Methode-Unterprogramm ist hingegen völlig unabhängig von Strukturen der Schnittstelle und erhält seine Informationen nur aus der Parameter-Liste.

Die Parameterliste des T_CA_H_Methode-Programms ist wie folgt festgelegt:

WHAT, NKOMP, T, MOLES, H_E, H_E_DT, H_E_DX

Name	Typ	Zugriff
WHAT	C	Lesen
NKOMP	I4	Lesen
T	R8	Lesen
MOLES	R8	Lesen
H_E	R8	Schreiben
H_E_DT	R8	Schreiben
H_E_DX	R8(*)	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'M' : Ableitung nach den Molanteilen

NKOMP

Anzahl Komponenten

T

Temperatur in Kelvin

MOLES

Molanteile in der Flüssigkeit

H_E

Exzeßenthalpie

H_E_DT

Temperatur-Ableitung der Exzeßenthalpie

H_E_DX

Vektor der Ableitungen der Exzeßenthalpie nach den Molanteilen

Die **Parameterliste des H_Methode-Programms** ist wie folgt aufgebaut:

WHAT, allg. Größen, NKOMP, T, MOLES, Stoffdaten, Hilfsmatrizen, Hilfsvektoren, H_E, H_E_DT, H_E_DX

Name	Typ	Zugriff
WHAT	C	Lesen
allg. Größen		Lesen
NKOMP	I4	Lesen
T	R8	Lesen
MOLES	R8	Lesen
Stoffdaten	R8	Lesen
Hilfsvektoren	R8	Schreiben
Hilfsmatrizen	R8	Schreiben
H_E	R8	Schreiben
H_E_DT	R8	Schreiben
H_E_DX	R8(*)	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'M' : Ableitung nach den Molanteilen

allg. Größen

allgemeine Größen sind z.B. die Gaskonstante oder der Wert des maximalen Exponenten der e-Funktion, die in der Parameter-Includedatei der Schnittstelle gespeichert sind.

NKOMP

Anzahl Komponenten

T

Temperatur in Kelvin

MOLES

Molanteile in der Flüssigkeit

Stoffdaten

alle für die Berechnung nach dieser Methode notwendigen Stoffdaten

Hilfsvektoren

Vektoren für Zwischenergebnisse

Hilfsmatrizen

Matrizen für Zwischenergebnisse

H_E

Exzeßenthalpie

H_E_DT

Temperatur-Ableitung der Exzeßenthalpie

H_E_DX

Vektor der Ableitungen der Exzeßenthalpie nach den Molanteilen

Neue Programme **zur Berechnung der isothermen Druckabhängigkeit der Enthalpie** sind also im UP T_H_ISO in der Form

- T_CA_H_Methode
  - T_H_Methode_MIXT
  - T_H_Methode_REIN

aufzurufen. Das T_CA_H_Methode-Programme haben Zugriff auf die internen Datenstrukturen und Include-Dateien der Schnittstelle. Die T_H_Methode-Unterprogramme sind hingegen völlig unabhängig von Strukturen der Schnittstelle und erhalten ihre Informationen nur aus der Parameter-Liste.

Die Parameterliste des T_CA_H_Methode-Programms ist wie folgt festgelegt:

WHAT, NKOMP, JKOMP, T, P, MOLES, H_P, H_P_DT, H_P_DP, H_P_DX

Name	Typ	Zugriff
WHAT	C	Lesen
NKOMP	I4	Lesen
JKOMP	I4	Lesen
T	R8	Lesen
P	R8	Lesen
MOLES	R8(*)	Lesen
H_P	R8	Schreiben
H_P_DT	R8	Schreiben
H_P_DP	R8	Schreiben
H_P_DX	R8(*)	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen

NKOMP

Anzahl Komponenten

JKOMP

Komponentennummer

=0 : ΔH für das Gemisch

>0 : ΔH für die angegebene Komponente

T

Temperatur in Kelvin

P

Druck in Pascal

MOLES

Molanteile

H_P

isotherme Druckabhängigkeit

H_E_DT

Temperatur-Ableitung

H_E_DP

Druck-Ableitung

H_E_DX

Vektor der Ableitungen nach den Molanteilen

Die **Parameterliste des T_H_Methode_MIXT-Programms** ist wie folgt aufgebaut:

WHAT, allg. Größen, NKOMP, T, MOLES, Stoffdaten, Hilfsmatrizen, Hilfsvektoren, H_P, H_P_DT, H_P_DP, H_P_DX, RVAL

Name	Typ	Zugriff
WHAT	C	Lesen
allg. Größen		Lesen
NKOMP	I4	Lesen
T	R8	Lesen
MOLES	R8(*)	Lesen
Stoffdaten	R8	Lesen
Hilfsvektoren	R8(*)	Schreiben
Hilfsmatrizen	R8(NKOMP,*)	Schreiben
H_P	R8	Schreiben
H_P_DT	R8	Schreiben
H_P_DP	R8	Schreiben
H_P_DX	R8(*)	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen

allg. Größen

allgemeine Größen sind z.B. die Gaskonstante oder der Wert des maximalen Exponenten der e-Funktion, die in der Parameter-Includedatei der Schnittstelle gespeichert sind.

NKOMP

Anzahl Komponenten

T

Temperatur in Kelvin

P

Druck in Pascal

MOLES

Molanteile

Stoffdaten

alle für die Berechnung nach dieser Methode notwendigen Stoffdaten

Hilfsvektoren

Vektoren für Zwischenergebnisse

Hilfsmatrizen

Matrizen für Zwischenergebnisse

H_P

isotherme Druckabhängigkeit

H_P_DT

Temperatur-Ableitung

H_P_DP

Druck-Ableitung

H_P_DX

Vektor der Ableitungen nach den Molanteilen

Die **Parameterliste des T_H_Methode_REIN-Programms** ist wie folgt aufgebaut:

WHAT, allg. Größen, T, Stoffdaten, H_P, H_P_DT

Name	Typ	Zugriff
WHAT	C	Lesen
allg. Größen		Lesen
T	R8	Lesen
MOLES	R8(*)	Lesen
Stoffdaten	R8	Lesen
H_P	R8	Schreiben
H_P_DT	R8	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

allg. Größen

allgemeine Größen sind z.B. die Gaskonstante oder der Wert des maximalen Exponenten der e-Funktion, die in der Parameter-Includedatei der Schnittstelle gespeichert sind.

T

Temperatur in Kelvin

Stoffdaten

alle für die Berechnung nach dieser Methode notwendigen Stoffdaten

H_P

isotherme Druckabhängigkeit

H_P_DT

Temperatur-Ableitung

T_ENTH

Berechnung der Enthalpie

FORMAT

T_ENTH(WHAT, PHASE, T, P, MOLES, ENTHALPY, RVAL)

Name	Typ	Zugriff
WHAT	C	Lesen
PHASE	C	Lesen
T	R8	Lesen
P	R8	Lesen
MOLES	R8(*)	Lesen
ENTHALPY	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen der entsprechenden Phase

PHASE

Phasenkennzeichung

LIQU : Enthalpie der Flüssigkeit

VAPO : Enthalpie des Gases

SOLI : Enthalpie des Feststoffs

T

Temperatur in Kelvin

P

Druck in Pascal

MOLES

Molanteile der Komponenten in der entsprechenden Phase

ENTHALPY

Enthalpie

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Rechnung ohne Enthalpie

FUNKTION

Das Programm T_ENTH ruft je nach PHASE das für die entsprechende Berechnung zuständige Unterprogramm auf:

T_ENTH_LIQUID Flüssigkeitsenthalpie

T_ENTH_VAPOR Gasenthalpie

T_ENTH_SOLID Feststoffenthalpie

Die Methoden, nach denen gerechnet werden soll, werden durch Aufruf von T_G_ENTH in den einzelnen Unterprogrammen abgefragt.

Zur Speicherung der Enthalpien und ihrer Ableitungen stehen in /PROP_CALC_ZEIGER/ folgende Zeiger der Länge NKOMPMAX + 2 zur Verfügung:

- ILH_ABL - Ableitungen der Flüssigkeitsenthalpie
- IVH_ABL - Ableitungen der Gasenthalpie
- ISH_ABL - Ableitungen der Feststoffenthalpie

Bei Rechnung ohne Enthalpie wird der Wert = 0 gesetzt und die Fehlerkennung auf 1.

T_ENTH_DERIVATIVE

Rückgabe der Ableitungen der Enthalpie

FORMAT

T_ENTH_DERIVATIVE(WHAT, PHASE, DERIVATIVE, RVAL)

Name	Typ	Zugriff
WHAT	C1	Lesen
PHASE	C	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen

PHASE

Kennung der Phase, für die die Ableitung zurückgegeben werden soll.

DERIVATIVE

Ableitungen

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_ENTH gebildet und intern im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in der ersten Größe bedeutet, daß beim letzten Aufruf von T_ENTH keine Ableitungen gebildet wurden.

Die Ableitungen sind über die Zeiger ILH_ABL, IVH_ABL bzw. ISH_ABL im Daten-Common doppeltgenau gespeichert.

T_H_EX

Berechnung der Exzeßenthalpie

FORMAT

T_H_EX(METHODE, WHAT, T, MOLES, H_E, RVAL)

Name	Typ	Zugriff
METHODE	C	Lesen
WHAT	C	Lesen
T	R8	Lesen
MOLES	R8(*)	Lesen
H_E	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

METHODE

Kennzeichnung der Berechnungsmethode für die Exzeßenthalpie

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'M' : Ableitung nach den Molanteilen

T

Temperatur in Kelvin

MOLES

Molanteile der Komponenten

H_E

Exzeßenthalpie

RVAL

Fehlerkennung

= 0 : alles ok.

FUNKTION

Das Programm T_H_EX ruft je nach angegebener Methode das für die entsprechende Berechnung zuständige Unterprogramm auf.

Zur Speicherung der Ableitungen der Exzeßenthalpie steht in /PROP_CALC_ZEIGER/ der Zeiger IH_MIXT der Länge NKOMPMAX + 1 zur Verfügung.

T_H_EX_DERIVATIVE

Rückgabe der Ableitungen der Exzeßenthalpie

FORMAT

T_H_EX_DERIVATIVE(WHAT, DERIVATIVE, RVAL)

Name	Typ	Zugriff
WHAT	C1	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'M' : Ableitung nach den Molanteilen

DERIVATIVE

Ableitungen

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_H_EX gebildet und intern im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in der ersten Größe bedeutet, daß beim letzten Aufruf von T_H_EX keine Ableitungen gebildet wurden.

Die Ableitungen sind über den Zeiger IH_MIXT im Daten-Common doppeltgenau gespeichert:

R8FELD(IH_MIXT) - T-Ableitung

R8FELD(IH_MIXT+1) bis R8FELD(IH_MIXT+NKOMP+1) - M-Ableitungen

T_H_ISO

Berechnung der isothermen Druckabhängigkeit der Enthalpie des Gases

FORMAT T_H_ISO(METHODE, WHAT, JKOMP, T, P, MOLES, H_P, RVAL)

Name	Typ	Zugriff
METHODE	C	Lesen
WHAT	C	Lesen
JKOMP	I4	Lesen
T	R8	Lesen
P	R8	Lesen
MOLES	R8(*)	Lesen
H_P	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

METHODE

Methodenkennzeichung

WHAT

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen der entsprechenden Phase

JKOMP

Komponentennummer

=0 : Berechnung für das Gemisch

>0 : Berechnung für eine Komponente

T

Temperatur in Kelvin

P

Druck in Pascal

MOLES

Molanteile der Komponenten

H_P

ΔH

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : keine Lösung für die Kompressibilität gefunden

FUNKTION

Das Programm T_H_ISO ruft je nach Methode das für die entsprechende Berechnung zuständige Unterprogramm auf.

Zur Speicherung der Ableitungen der Enthalpieänderung steht in /PROP_CALC_ZEIGER/ der Zeiger IH_ISOT der Länge NKOMPMAX + 2 zur Verfügung.

T_H_ISO_DERIVATIVE

Rückgabe der Ableitungen der Enthalpieänderung

FORMAT

T_H_ISO_DERIVATIVE(WHAT, DERIVATIVE, RVAL)

Name	Typ	Zugriff
WHAT	C1	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen

DERIVATIVE

Ableitungen

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_H_ISO gebildet und intern im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in der ersten Größe bedeutet, daß beim letzten Aufruf von T_H_ISO keine Ableitungen gebildet wurden.

Die Ableitungen sind über den Zeiger IH_ISOT im Daten-Common doppeltgenau gespeichert:

R8FELD(IH_ISOT) - T-Ableitung

R8FELD(IH_ISOT+1) - P-Ableitung

R8FELD(IH_ISOT+2) bis R8FELD(IH_ISOT+NKOMP+2) - M-Ableitungen

6.8 Chemische Reaktionen

Die Gleichung für eine chemische Reaktion wird mit dem Programm T_CHEM berechnet. Z.Zt. sieht der Aufrufbaum wie folgt aus:

- T_CHEM
  - T_G_REACTION
  - T_G_CH_ALLG
  - T_C_CONV
  - T_G_CH_ALLG
  - T_C_STAT
  - T_G_CH_ALLG
  - T_C_COOR
  - T_CA_EQ
    - T_G_CH_EQ
    - T_FVT_VALUE
    - T_LVEQ
    - T_LVEQ_DERIVATIVE
    - T_PURE
    - T_PURE_DERIVATIVE
    - T_COMPR
    - T_COMPR_DERIVATIVE
    - T_POYNT
    - T_POYNT_DERIVATIVE
    - T_FUGA
    - T_FUGA_DERIVATIVE
    - T_C_EQM
    - T_C_EQLA
    - T_C_EQVP
    - T_C_EQLC
    - T_C_EQVC
    - T_C_EQLF
    - T_C_EQVF
  - T_CA_KI
    - T_G_CH_KI
    - T_FVT_VALUE
    - T_PURE
    - T_PURE_DERIVATIVE
    - T_COMPR
    - T_COMPR_DERIVATIVE
    - T_AVER
    - T_AVER_DERIVATIVE
    - T_G_SINGLE_COMP
    - T_C_KILM
    - T_C_KIVM
    - T_C_KILC
    - T_C_KIVC
    - T_C_KILW

    - T_C_KIVW
- T_CHEM_DERIVATIVE

Zur Berechnung der Reaktionsterme der Massen- bzw. Enthalpiebilanzgleichungen steht das Unterprogramm T_CH zur Verfügung.

- T_CH
- T_CH_DERIVATIVE

Für neue **Gleichgewichtsreaktionen** sind die Berechnungsprogramme im UP T_CA_EQ einzubinden.
Die Parameterliste ist in folgender Form vorgegeben:

WHAT, NKOMP,allg. Größen, T, P, MOLES, STOE, FVT, FVT_T, Hilfsgrößen, VALUE, VALUE_T, VALUE_P, VALUE_M, RVAL

Name	Typ	Zugriff
WHAT	C	Lesen
allg. Größen		Lesen
NKOMP	I4	Lesen
T	R8	Lesen
P	R8	Lesen
MOLES	R8(*)	Lesen
STOE	R8(*)	Lesen
FVT	R8	Lesen
FVT_T	R8	Lesen
Hilfsgrößen		Lesen/Schreiben
VALUE	R8	Schreiben
VALUE_T	R8	Schreiben
VALUE_P	R8	Schreiben
VALUE_M	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen

NKOMP

Anzahl Komponenten

allg. Größen

allgemeine Größen sind z.B. die Gaskonstante , die in der Parameter-Includedatei der Schnittstelle gespeichert sind.

T

Temperatur in Kelvin

P

Druck in Pascal

MOLES

Molanteile

STOE

Vektor der stöchiometrischen Koeffizienten

FVT

Wert der F(T)-Funktion

FVT_T

T-Ableitung der F(T)-Funktion

Hilfsgrößen

alle für die Berechnung nach dieser Methode notwendigen Stoffdaten und Hilfsfelder zur Zwischenspeicherung

VALUE

berechneter Wert

VALUE_T

Temperatur-Ableitung

VALUE_P

Druck-Ableitung

VALUE_M

Vektor der Ableitungen nach den Molanteilen

RVAL

Fehlerkennung

= 0 : alles ok.

Für neue **kinetisch kontrollierte Reaktionen** sind die Berechnungsprogramme im UP T_CA_KI einzubinden.
Die Parameterliste ist in folgender Form vorgegeben:

WHAT, NKOMP,allg. Größen, T, P, ZETA, VOL, MOLES, Hilfsgrößen, NFVT, FVT, FVT_T, ALPHA, NPHI, PHI, PHI_T, GAMMA, VALUE, VALUE_T, VALUE_P, VALUE_Z, VALUE_V, VALUE_M, RVAL

Name	Typ	Zugriff
WHAT	C	Lesen
NKOMP	I4	Lesen
allg. Größen		Lesen
T	R8	Lesen
P	R8	Lesen
ZETA	R8	Lesen
VOL	R8	Lesen
MOLES	R8(*)	Lesen
Hilfsgrößen		Lesen/Schreiben
NFVT	I4	Lesen
FVT	R8(*)	Lesen
FVT_T	R8(*)	Lesen
ALPHA	R8(NKOMP,*)	Lesen
NPHI	I4	Lesen
PHI	R8(*)	Lesen
PHI_T	R8(*)	Lesen
GAMMA	R8(NKOMP,*)	Lesen
VALUE	R8	Schreiben
VALUE_T	R8	Schreiben
VALUE_P	R8	Schreiben
VALUE_Z	R8	Schreiben
VALUE_V	R8	Schreiben
VALUE_M	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, der die zu bildenden Ableitungen kennzeichnet

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'M' : Ableitung nach den Molanteilen

NKOMP

Anzahl Komponenten

allg. Größen

allgemeine Größen sind z.B. die Gaskonstante , die in der Parameter-Includedatei der Schnittstelle gespeichert sind.

T

Temperatur in Kelvin

P

Druck in Pascal

ZETA

Reaktionsrate

VOL

Volumen

MOLES

Molanteile

Hilfsgrößen

alle für die Berechnung nach dieser Methode notwendigen Stoffdaten und Hilfsfelder zur Zwischenspeicherung

NFVT

Anzahl der F(T) - Funktionen

FVT

Werte der F(T)-Funktionen

FVT_T

T-Ableitungen der F(T)-Funktionen

NPHI

Anzahl der Phi - Funktionen

PHI

Werte der Phi-Funktionen

PHI_T

T-Ableitungen der Phi-Funktionen

VALUE

berechneter Wert

VALUE_T

Temperatur-Ableitung

VALUE_P

Druck-Ableitung

VALUE_Z

Zeta-Ableitung

VALUE_V

Volumen-Ableitung

VALUE_M

Vektor der Ableitungen nach den Molanteilen

RVAL

Fehlerkennung

= 0 : alles ok.

= 1 : Reaktion kann nicht laufen; --> Reaktionsrate = 0

T_CHEM

Berechnen einer Reaktionsgleichung

FORMAT

T_CHEM(REAC_NR, WHAT, T, P; ZETA, VOL, MOLES, VALUE, RVAL)

Name	Typ	Zugriff
REAC_NR	I4	Lesen
WHAT	C1	Lesen
T	R8	Lesen
P	R8	Lesen
ZETA	R8	Lesen
VOL	R8	Lesen
MOLES	R8(*)	Lesen
VALUE	R8	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

REAC_NR

Nummer der Reaktion

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'V' : Ableitung nach dem Volumen

'Z' : Ableitung nach der Reaktionsrate

'M' : Ableitung nach den Molanteilen

T

Temperatur

P

Druck

ZETA

Reaktionsrate

VOL

Volumen

MOLES

Vektor mit den Molanteilen in der Phase, in der die Reaktion stattfindet

VALUE

berechneter Wert

RVAL

Fehlerkennung

= 0: alles ok.

FUNKTION

Das UP T_CHEM ruft identifiziert den Reaktionstyp der angegebenen Reaktion und ruft das entsprechende Unterprogramm zur Berechnungssteuerung auf.

Ist im Parameter WHAT gekennzeichnet, daß Ableitungen gebildet werden sollen, werden diese direkt mitberechnet und auf den Feldern ICHALLG und ICHM im Daten-Common doppeltgenau gespeichert:

R8FELD(ICHALLG) T-Ableitung

R8FELD(ICHALLG+1) P-Ableitung

R8FELD(ICHALLG+2) Z-Ableitung

R8FELD(ICHALLG+3) V-Ableitung

R8FELD(ICHM) bis R8FELD(ICHM+NKOMP-1) - M-Ableitungen.

Nicht berechnete Ableitungen erhalten die Kennung -1.D30 (NONVAL in der Include-Datei).

T_CHEM_DERIVATIVE

Rückgabe der Ableitungen der Reaktionsgleichungen

FORMAT

T_CHEM_DERIVATIVE(WHAT, DERIVATIVE, RVAL)

Name	Typ	Zugriff
WHAT	C1	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

WHAT

Steuerparameter, welche Ableitungen zurückgegeben werden sollen

'T' : Ableitung nach der Temperatur

'P' : Ableitung nach dem Druck

'V' : Ableitung nach dem Volumen

'Z' : Ableitung nach der Reaktionsrate

'M' : Ableitung nach den Molanteilen

DERIVATIVE

Ableitungen

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_CHEM gebildet und intern im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in der ersten Größe bedeutet, daß beim letzten Aufruf von T_CHEM keine Ableitungen gebildet wurden.

Die Ableitungen sind über die Zeiger ICHALLG und ICHM im Daten-Common doppeltgenau gespeichert:

R8FELD(ICHALLG) T-Ableitung

R8FELD(ICHALLG+1) P-Ableitung

R8FELD(ICHALLG+2) Z-Ableitung

R8FELD(ICHALLG+3) V-Ableitung

R8FELD(ICHM) bis R8FELD(ICHM+NKOMP-1) - M-Ableitungen

T_CH

Berechnung der Massen- und Enthalpiebilanzterme für Reaktionen

FORMAT

T_CH(TASK, WHAT, ZETA, VALUE, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
WHAT	C	Lesen
ZETA	R8	Lesen
VALUE	R8(*)	Schreiben
RVAL	R8	Schreiben

ARGUMENTE

TASK

Aufgabenstellung

= 1 : Massenbilanz

= 2 : Enthalpiebilanz

WHAT

Steuerparameter, der die zu bildenden partiellen Ableitungen kennzeichnet

gültige Kennzeichnungen sind:

'Z' : Ableitung nach der Reaktionskoordinate

ZETA

Reaktionskoordinate

VALUE

gewünschte Werte

RVAL

Fehlerkennung

= 0: alles ok.

FUNKTION

Es werden Werte zum aktivierten Reaktionssystem (s. T_S_CHEMICS) berechnet.

Für TASK = 1 enthält VALUE als Vektor der Länge Anzahl Komponenten die Massenbilanzsummen je Komponente.

Für TASK = 2 enthält VALUE die Summe der Reaktionswärmen.

Ist im Parameter WHAT gekennzeichnet, daß die partiellen Ableitungen gebildet werden sollen, so werden diese direkt mitberechnet und zur Abfrage mit dem Programm T_CH_DERIVATIVE intern gespeichert. Ein erneuter Aufruf von T_CH überschreibt die Ableitungen.

T_CH_DERIVATIVE

Rückgabe von Ableitungen der Reaktionsterme für Massen- bzw. Enthalpiebilanz.

FORMAT

T_CH_DERIVATIVE(TASK, WHAT, COMP_NR, DERIVATIVE, RVAL)

Name	Typ	Zugriff
TASK	I4	Lesen
WHAT	C1	Lesen
COMP_NR	I4	Lesen
DERIVATIVE	R8(*)	Schreiben
RVAL	I4	Schreiben

ARGUMENTE

TASK

=1: Massenbilanzableitungen

=2: Enthalpiebilanzableitungen

WHAT

Steuerparameter, welche Ableitung zurückgegeben werden soll

'Z' : Ableitung nach Zeta

COMP_NR

Komponentennummer für TASK=1

DERIVATIVE

enthält die gewünschte Ableitung

RVAL

Fehlerkennung

= 0: alles ok.

= 1: Ableitung wurde nicht gebildet

=-1: ungültige Ableitungskennung

FUNKTION

Die Ableitungen wurden im UP T_CH gebildet.und intern im Daten-Common doppeltgenau gespeichert. Die Wert -1.D30 in der ersten Größe bedeutet, daß beim letzten Aufruf von T_CHEM keine Ableitungen gebildet wurden.

Die Ableitungen sind über die Zeiger ICHEMZ und ICHEMHZ im Daten-Common doppeltgenau gespeichert:

R8FELD(ICHEMZ)	Matrix der Ableitungen der Massenbilanzgleichungen; Dimension Anz. Komponenten * Anzahl Reaktionen
R8FELD(ICHEMHZ)	Vektor der Ableitungen der Enthalpiebilanzterme; Dimension Anz. Reaktionen

Für TASK = 1 werden im Ausgabevektor die Ableitungen der j-ten Komponente für alle Reaktionen geliefert. Für Task =2 enthält der Ausgabevektor die Ableitungen des Enthalpiebilanzterms je Reaktion.

7 Hinweise zur Erweiterung

Um die Schnittstelle mit neuen Funktionen und Stoffdaten zu versorgen, sind spezielle Schritte notwendig. Diese sind abhängig davon, welche neuen Funktionen oder Stoffdatentypen aufgenommen werden sollen.

Die folgenden Beschreibungsteile sollen ein Hinweis darauf sein, in welchen Unterprogrammen und Common-Blöcken beim Einbau einer speziellen neuen Funktion oder eines Stoffdatentyps geändert werden muß.

7.1 Erweitern der Speicherfelder

Die Speicherfelder /PROP_CHAR/ bzw. /PROP_DATA/ müssen je nach Stoffdatenbedarf eventuell erweitert werden. Hierzu sind folgende Aktionen notwendig:

- Eintragen der neuen Größe in die INCLUDE-Datei für den Common-Block
- Ändern der Common-Größe NPFELDE bzw. NCFELDE im UP T_INIT

7.2 Einfügen neuer Stoffkonstanten

Zum Einfügen neuer Typen von Stoffkonstanten sind folgende Aktionen notwendig:

- Eintragen der Identifikationskennung im UP T_ASK_SINGLE und Vergrößern des Parameters NSINGLE
- Eintragen eines neuen Zeigers in die INCLUDE-Datei für den Common /PROP_ZEIGER/
- Ändern der Common-Größe NSINGAKT im UP T_INIT

7.3 Einfügen neuer Stoff-Funktionen

Zum Einfügen neuer Typen von Stoff-Funktionen sind folgende Aktionen notwendig:

- Eintragen der Identifikationskennung im UP T_ASK_PURE und Vergrößern des Parameters NPURE
- Eintragen eines neuen Zeigers in die INCLUDE-Datei für den Common /PROP_ZEIGER/
- Ändern der Common-Größe NWPAKT im UP T_INIT

7.4 Einfügen neuer Ansätze für Stoff-Funktionen

Zum Einfügen neuer Ansätze für Stoff-Funktionen sind folgende Aktionen notwendig:

- Eintragen der Identifikationskennung im UP T_ASK_EQUA und Vergrößern des Parameters NEQUA
- Anbinden des entsprechenden Berechnungsunterprogramms im UP T_CA_PURE_FUNC
- Anbinden des entsprechenden Berechnungsunterprogramms im UP T_CA_PURE_EXTR
- Anbinden des entsprechenden Integrationsunterprogramms im UP T_CA_INT_PURE

Die Parameterliste des Berechnungsunterprogramms ist wie folgt vorgegeben:
Subroutine XYZ (LTABL, NCOEFF, COEFFICIENTS, T, VALUE, VALUE_T)

Die Parameterliste des Integrationsunterprogramms ist wie folgt vorgegeben:
Subroutine INT_XYZ (NCOEFF, COEFFICIENTS, T_UNTEN, T_OBEN, H_COEFF, VALUE)

7.5 Einfügen neuer Stoffparameter

Zum Einfügen neuer Typen von Stoffparameter sind folgende Aktionen notwendig:

- Eintragen der Identifikationskennung im UP T_ASK_PARAM und Vergrößern des Parameters NPARAM

- Eintragen eines neuen Zeigers in die INCLUDE-Datei für den Common /PROP_ZEIGER/
- Ändern der Common-Größe NPARAKT im UP T_INIT
- Einbinden des spezifischen Programms zur Speichervorbereitung in das UP T_P_PARAM
- Falls ein FILL-Programm notwendig ist, Einbinden des Aufrufs in das UP T_FI_SYSTEM

7.6 Einfügen neuer Matrizen

Zum Einfügen neuer Matrizen sind folgende Aktionen notwendig:

- Eintragen der Identifikationskennung im UP T_ASK_MATRIX und Vergrößern des Parameters NMATRIX
- Falls es mehrere Matrizen gibt, erstellen des Programms T_ASK_Typ, um die spezielle Kennung zu identifizieren.
- Eintragen eines neuen Zeigers in die INCLUDE-Datei für den Common /PROP_ZEIGER/
- Ändern der Common-Größe NMATAKT im UP T_INIT
- Einbinden des spezifischen Programms zur Speichervorbereitung in das UP T_P_MATRIX
- Einbinden des spezifischen Programms zum Holen der Daten in das UP T_G_MATRIX
- Falls ein FILL-Programm notwendig ist, Einbinden des Aufrufs in das UP T_FI_SYSTEM

7.7 Einfügen neuer Mittelwertbildungen

Zum Einfügen neuer Berechnungsmethoden für Mittelwertbildungen sind folgende Aktionen notwendig:

- Eintragen der Identifikationskennung im UP T_ASK_AVER und Vergrößern des Parameters NAVER
- Anbinden des entsprechenden Berechnungsunterprogramms im UP T_CA_AVER

Die Parameterliste des Berechnungsprogramms hat folgenden Aufbau:
Subroutine MITT (NKOMP, WHAT, T, MOLES, Hilfsgrößen, STOFF, STOFF_T, VALUE, VALUE_T,
VALUE_M, RVAL)

7.8 Einfügen neuer Methoden für Aktivitätskoeffizienten

Um neue Methoden für Aktivitätskoeffizienten einfügen zu können, muß zuerst dafür gesorgt werden, daß die entsprechenden Basisstoffdaten auch gespeichert sind (Parameter, Matrizen, etc.).
Ist dies geklärt, sind 2 Programme zu erstellen:

- T_CA_Methode
  - T_Methode

Das Programm *Methode* enthält die Berechnung der Aktivititätskoeffizienten und deren Ableitungen. **Alle** benötigten Größen werden über Parameterliste eingegeben.
Das Programm T_*CA_Methode* kennt die internen Strukturen der Thermodynamik-Schnittstelle und kann so die Stoffdaten, Hilfsvektoren und Hilfsmatrizen für das Unterprogramm zur Verfügung stellen.

Die Parameterliste des Methodenprogramms ist in folgender Form zu halten:

WHAT, NKOMP, allg. Größen, T, LIQUID, Stoffdaten, Hilfsvektoren, GAMMA, GAMMA_DT, GAMMA_DX

Das Programm T_CA_Methode muß dann im UP T_GAMMA aufgerufen werden. Gilt diese Berechnungsmethode auch für Flüssig-Flüssig-Gleichgewichte, muß der Aufruf auch in T_LL_GAMMA erfolgen.

7.9 Einfügen neuer Methoden für Exzeßenthalpien

Um neue Methoden für Exzeßenthalpien einfügen zu können, muß zuerst dafür gesorgt werden, daß die entsprechenden Basisstoffdaten auch gespeichert sind (Parameter, Matrizen, etc.).
Ist dies geklärt, sind 2 Programme zu erstellen:

- T_CA_H_Methode
  - T_H_Methode

Das Programm T_H_*Methode* enthält die Berechnung der Exzeßenthalpie und deren Ableitungen. **Alle** benötigten Größen werden über Parameterliste eingegeben.
Das Programm T_*CA_H_Methode* kennt die internen Strukturen der Thermodynamik-Schnittstelle und kann so die Stoffdaten, Hilfsvektoren und Hilfsmatrizen für das Unterprogramm zur Verfügung stellen.

Die Parameterliste des Methodenprogramms ist in folgender Form zu halten:

WHAT, NKOMP, allg. Größen, T, LIQUID, Stoffdaten, Hilfsvektoren, H_E, H_E_DT, H_E_DX

Das Programm T_CA_H_Methode muß dann im UP T_H_EX aufgerufen werden.

7.10 Einfügen neuer Methoden für Zustandsgleichungen, Fugazitätskoeffizienten und Kompressibilitätsfaktoren

Um neue Methoden für Zustandsgleichungen, Fugazitätskoeffizienten und Kompressibilitätsfaktoren einfügen zu können, muß zuerst dafür gesorgt werden, daß die entsprechenden Basisstoffdaten auch gespeichert sind (Parameter, Matrizen, etc.).
Ist dies geklärt, sind 2 Programme zu erstellen:

- T_CA_ZMethode
  - T_ZMethode_MIXT
  - T_ZMethode_REIN
- T_CA_Methode
  - T_Methode

Das Programm T_*Methode* berechnet je nach Aufgabenstellung die entsprechenden Fugazitätskoeffizienten und deren Ableitungen. **Alle** benötigten Größen werden über Parameterliste eingegeben.

Das Programm T_*CA_Methode* kennt die internen Strukturen der Thermodynamik-Schnittstelle und kann so die Stoffdaten, Hilfsvektoren und Hilfsmatrizen für das Unterprogramm zur Verfügung stellen.

Die Parameterliste des Methodenprogramms ist in folgender Form zu halten:

ITASK, WHAT, NKOMP, allg. Größen, T, P, LIQUID, VAPOR, Stoffdaten, Hilfsvektoren, PHI, PHI_DT, PHI_DP, PHI_DX, PHI_DY, RVAL

Die Aufrufe der T_CA-Programme sind entsprechend den Anforderungen in den Programmen T_ZUST, T_FUGA, T_COMPR aufzunehmen.