ASIMPTOTE Advanced SIMulation Programs for TOTal Energy systems

User Subroutines

A program for thermodynamic modeling
and optimization of energy conversion systems

Table of Contents

Chapter 1 Introduction

1.1 Introduction 1-1
1.2 Hardware and software requirements 1-2
1.3 Installation 1-2

Chapter 2 Working with User Subroutines

2.1 Creating user subroutines 2-3
2.2 Compiling and linking 2-3
2.3 Input for user subroutines in Cycle-Tempo 2-4
2.4 Performing calculations using user subroutines 2-5

Chapter 3 Geneal Observations

3.1 Available user subroutines 3-1
3.2 Calculation procedure 3-2
3.3 FORTRAN 3-3
3.4 Input and output 3-5

Chapter 4 User Subroutines in Detail

4.1 Memory capacity to be stated by the user (CTMAIN) 4-1
4.2 APSUB subroutine 4-3
4.2.1 Introduction 4-3
4.2.2 Argument list and declarations of variables 4-4
4.2.3 Classification and overview of the argument list 4-4
4.2.4 Explanation of the variables in the argument list 4-6
4.2.5 Examples 4-11
4.3 PIPSUB subroutine 4-13
4.3.1 Introduction 4-13
4.3.2 Argument list and declarations of variables 4-14
4.3.3 Classification and overview of the argument list 4-14
4.3.4 Explanation of the variables in the argument list 4-15
4.3.5 Example 4-18
4.4 USREAC subroutine 4-19

4.4.1 Introduction 4-19
4.4.2 Argument list and declarations of variables 4-19
4.4.3 Explanation of the variables 4-20
4.4.4 The operation of USREAC 4-20
4.4.5 Example: COS conversion 4-21
4.5 FUNCOP user function 4-22
4.5.1 Application 4-22
4.5.2 Argument list and declarations of variables 4-22
4.5.3 Explanation of the variables in the argument list 4-22
4.6 EPILOG subroutine 4-26
4.6.1 Introduction 4-26
4.6.2 Argument list and declaration of variables 4-26
4.6.3 Classification and overview of the argument list 4-27
4.6.4 Explanation of the variables in the argument list 4-29

Chapter 5 Available Functions

5.1 Introduction 5-1
5.2 Useful functions for working with apparatuses 5-2
5.3 Useful functions for working with pipes 5-5
5.4 Functions for thermodynamic properties 5-7

Appendix A Apparatus Variables

Chapter 1

Introduction

1.1 Introduction

In process calculations, it is possible to characterize variables of apparatuses that depend on calculated mass flows or compositions. We will explain this with the following examples:

- The isentropic efficiency of a turbine can depend on the mass flow through the turbine.
- The condenser outlet pressure can depend on the quantity of heat transferred.
- The pressure drop in a pipe can depend on the mass flow through the pipe.
- A valve joining two flows can be controlled so as to produce a desired outlet composition.
- For a separation apparatus, for example a cyclone that can be modeled using a valve, the outlet compositions depend, among other things, on inlet composition and volume flow.

All these dependencies differ per system. Besides that, we would like the user to be able to add his own relations to the program in a simple manner; this is possible with a “user subroutine”. The program has six user subroutines. The examples referred to above can be performed using the APSUB and PIPSUB user subroutines. The other four user subroutines are: EPILOG for post-processing facilities, USREAC for defining chemical reactions, FUNCOP for defining an optimization function, and CTMAIN for modifying the memory required for the calculations.

The calculation procedure for Cycle-Tempo is a closed module in which openings have been created at some places. User subroutines can be linked to these openings. The user subroutines are bits of FORTRAN code in which the user himself programs the desired behavior. If no user subroutines are applied, dummy user subroutines have been linked to the openings in the module.

1.2 Hardware and software requirements

In order to use user subroutines, the user should have a FORTRAN compiler, i.e. at least:

Intel® Visual Fortran Compiler 9.0, Standard Edition, for Windows

The hardware requirements for using user subroutines are the same as for Cycle-Tempo itself (see the "Introduction" part). Note, however, that additional disk space is required for the above compiler.

1.3 Installation

During the installation of Cycle-Tempo, the user is free to determine in which directory the program is selected. This manual assumes *C:\Program Files\Cycle-Tempo* as default directory.

During the Cycle-Tempo installation, the object files of the Cycle-Tempo calculation module are saved in the directory *C:\Program Files\Cycle-Tempo\System*. The object files are clustered in the files *WinTempo.lib* and *WinTempo.res*. Furthermore, the source code (extension .FOR) of the following files is available:

CTMAIN
APSUB
PIPSUB
USREAC
FUNCOP
EPILOG

In the file CTMAIN.FOR, the user can modify, if necessary, the memory space reserved for Cycle-Tempo, in connection with array sizes. The user himself can modify the APSUB.FOR, PIPSUB.FOR, USREAC.FOR, FUNCOP.FOR and EPILOG.FOR files so as to create user subroutines himself.
The file COMPILE.BAT is used to compile and link user subroutines with the Cycle-Tempo object files.

Chapter 2

Working with User Subroutines

2.1 Creating user subroutines

You can create a user subroutine using any text editor. This may be either the editor supplied with the compiler or any other editor.
The easiest way is to copy a supplied template (for example *C:\Program Files\Cycle-Tempo\System*) directory to the desired work directory and then add your own code to it.

2.2 Compiling and linking

Subroutines created by you should be compiled and linked with the Cycle-Tempo object files.

Although other ways are possible the user is advised to make use of the files COMPILE.BAT, WINTEMPO.LIB and WINTEMPO.RES supplied with the program, and present in the Cycle-Tempo subdirectory SYSTEM after installation. You can copy the APSUB.FOR, PIPSUB.FOR, USREAC.FOR, FUNCOP.FOR and EPILOG.FOR files from the system directory (default: *C:\Program Files\Cycle-Tempo\System*) to your work directory under the same or other names. The names of the files do not matter; however, it is required that the names of the subroutines remain the same, i.e., APSUB, PIPSUB, USREAC, FUNCOP and EPILOG, respectively. You can create your own versions of the APSUB, PIPSUB, USREAC, FUNCOP and EPILOG subroutines. The files in which these subroutines are found, as well as files that contain subroutines opened from APSUB, PIPSUB, USREAC, FUNCOP and EPILOG, respectively, should also be compiled.

To compile a Fortran source file, for example APSUB.FOR, use the Windows command interpreter (open *Run* in the Windows Start Menu and type *cmd*) and go to directory where the user subroutine resides, for example:

```
cd C:\My Documents\MyUserSubs
```

Next use the COMPILE.BAT file in the Windows command interpreter by giving the command:

```
compile apsub
```

Then the batch file will ensure that the file APSUB.FOR is compiled using the correct compiler switches. Note that the COMPILE.BAT file assumes that the Fortran compiler has been installed in the C:\PROGRAM FILES\INTEL\COMPILER\FORTRAN\9.1 directory. If the compiler is present in a different directory or you have a different version of the compiler, you should change the data in the COMPILE.BAT file accordingly. Usually you can find this data in the file IFORTVARS.BAT, which is in one of the subdirectories where the compiler has been installed.

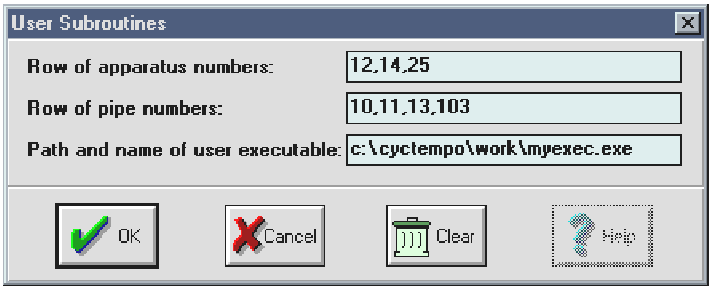
When you compiled your own Fortran source files, the batch file creates an executable file named USERSUB.EXE. The executable USERSUB.EXE is placed in the directory where the compile command is issued. You can refer to this executable in the input window for user subroutines (see next chapter).

2.3 Input for user subroutines in Cycle-Tempo

In the input window for user subroutines, you can state what apparatuses and pipes must be controlled by user subroutines.

To open the input window for user subroutines:

- Select: Calculation | User Subroutines


User Subroutines

Row of apparatus numbers: 12,14,25

Row of pipe numbers: 10,11,13,103

Path and name of user executable: c:\cyctempo\work\myexec.exe

Figure 2-1: input window for user subroutines

As an illustration, several apparatus and pipe numbers have been entered in Figure 2-1. Enter the name of the user executable into the lower text box made during the compiling and linking of the user subroutine itself.

If you use the user subroutine APSUB, the row with apparatus numbers (maximally 50) has to be filled in. Commas must separate the apparatus numbers.
The order in which the apparatus numbers are listed in the input window determines the order in which they can be opened from APSUB. This can be important if, for example, data need to be transferred from one apparatus to another. The apparatus supplying data must be listed in front of (the number of) the apparatus receiving data.

If you use the PIPSUB subroutine, the row with pipe numbers (maximally 100) should be entered. Commas must separate the pipe numbers.
Just as with the apparatus numbers, the order of the pipe numbers is important if data from the one pipe need to be passed to another.

If only user subroutines other than APSUB and PIPSUB are used, it is sufficient to enter the name of the user executable.

2.4 Performing calculations using user subroutines

You can perform calculations using user subroutines simply by entering the name of the user executable in the input window for user subroutines (see also section 2.3). You can then make the calculation in exactly the same manner as a calculation without user subroutines:

- Select: Calculation | Run!

Chapter 3

General Observations

3.1 Available user subroutines

The program has six user subroutines available. The overview below states the available subroutines and their objectives.

Name of user subroutine	Description of objective
CTMAIN	Modification of required memory allocation
APSUB	Modification of the behavior of an apparatus
PIPSUB	Modification of the behavior of a pipe
USREAC	Input of a user-defined reaction
FUNCOP	Input of a user-defined optimization function
EPILOG	Input of a user-defined 'post processing' procedure

The routines are found in the files that have the same name as the routine name, only with the extension ".FOR".

3.2 Calculation procedure

The PROGRAM CTMAIN routine is the first routine to be activated for calculations and also the place to reserve memory for the calculations.

In the main iterations, the apparatus subroutines of the apparatuses in the process diagram are activated again and again. If in a chemical reactor (type 27) the TRUSER parameter has been entered, the apparatus subroutine of the chemical reactor will load the USREAC user subroutine, in which you can define the desired chemical reaction yourself.

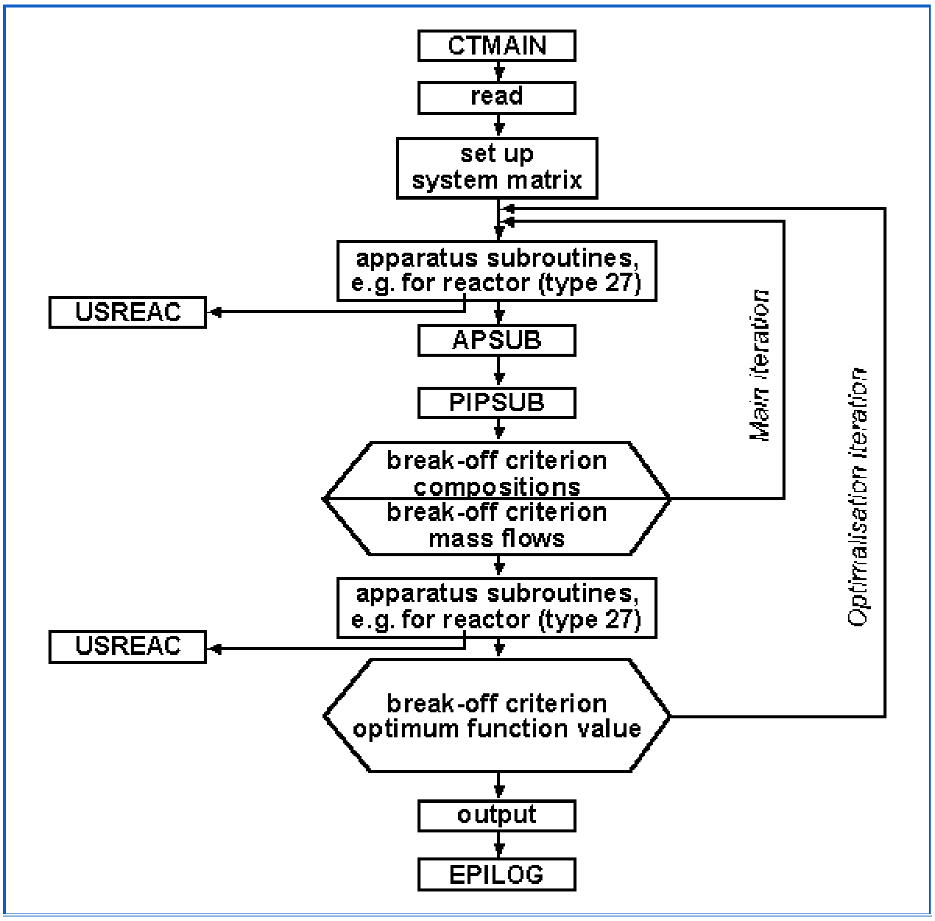

CTMAIN
read
set up system matrix
apparatus subroutines, e.g. for reactor (type 27)
USREAC
APSUB
PIPSUB
break-off criterion compositions
break-off criterion mass flows
Main iteration
Optimalisation iteration
apparatus subroutines, e.g. for reactor (type 27)
USREAC
break-off criterion optimum function value
output
EPILOG

Figure 3-1: place of user subroutines in the calculation procedure

After each main iteration in the calculation, the actual mass flows, compositions, pressures, temperatures and enthalpies are available for the APSUB and PIPSUB user subroutines.

When an optimization calculation has been performed, the system efficiency is calculated every time after convergence has been reached for the compositions and the mass flows. If the efficiency differs from previous optimization iterations, the parameters to be optimized are modified and a series of main iterations is started again. Using the FUNCOP user subroutine, with the data available in FUNCOP, you can establish an optimization function yourself, the function value of which should be optimized.

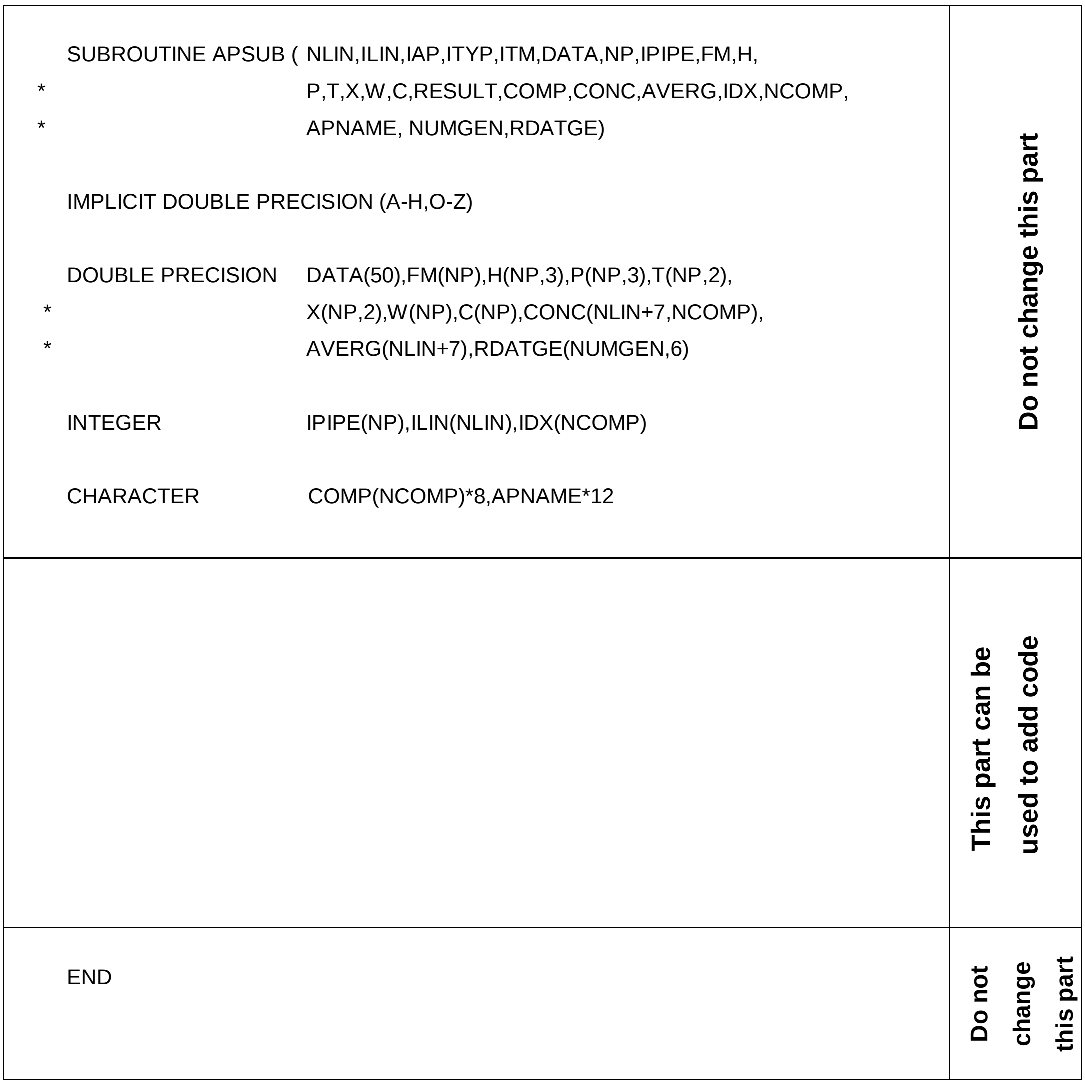
The sixth user subroutine, the EPILOG subroutine, is not activated during the main iterations. This subroutine serves for ‘post-processing’ the results and is activated at the very end of the program.

Figure 3-1 gives a graphical presentation of the places of the user subroutines in the program.

3.3 FORTRAN

The Cycle-Tempo calculation procedure has been written in standard FORTRAN 90, meaning that user subroutines must also be written in this language. The templates of user subroutines supplied with the program have been designed accordingly. Although various tricks with Argument lists and declarations are possible in FORTRAN, it is recommended not to make changes in the Argument lists and declaration parts (which together form the subroutine specification). Changes in these parts may disrupt the operation of the entire calculation procedure. The code fragment on the next page states in which part of the user subroutine code can be added ‘safely’.

Finally, it should be stated that the user should be reasonably familiar with FORTRAN in order to successfully apply user subroutines.



```
      SUBROUTINE APSUB ( NLIN,ILIN,IAP,ITYP,ITM,DATA,NP,IPIPE,FM,H,
    *                    P,T,X,W,C,RESULT,COMP,CONC,AVERG,IDX,NCOMP,
    *                    APNAME, NUMGEN,RDATGE)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION   DATA(50),FM(NP),H(NP,3),P(NP,3),T(NP,2),
     *                   X(NP,2),W(NP),C(NP),CONC(NLIN+7,NCOMP),
     *                   AVERG(NLIN+7),RDATGE(NUMGEN,6)

      INTEGER            IPIPE(NP),ILIN(NLIN),IDX(NCOMP)

      CHARACTER          COMP(NCOMP)*8,APNAME*12



      END
```

Figure 3-2: place where code can be added to a user subroutine

3.4 Input and output

In making user subroutines, you should bear in mind that, earlier in the program, files have been opened for reading and writing data. In Cycle-Tempo, the following unit numbers are used for these files:

4:	gas data input file (TEMPOGAS)
7:	standard output file of the calculation (Text output)
9:	standard input file for the calculation
12:	output file for off-design data, only used in design calculations
13:	output file for showing internal iteration processes and error messages (Message window)

Note! Unit numbers associated with standard input and output devices (unit numbers 5 and 6) are not allowed. This means that statements like READ(5,…), WRITE(6,…) are not allowed, nor usage of PRINT and PAUSE statements. Usage of these statements will lead to a premature termination of the calculation process.
Furthermore, the unit numbers 1, 10 and 22 are reserved for internal use and may not be used in user subroutines.
In a user subroutine, these unit numbers can ***not*** be used for matters other than those stated above. However, you can make use of the fact that these files are already open.

For example, by including the following statement:

```
WRITE(13,*) 'This text will appear in the message window…'
```

in a user subroutine, the text “This text will appear in the message window…” will be written in the message window after completion of the calculation. With `WRITE(7,*)`, you can cause text to appear in the Text output.

Design data for off-design calculations are written in unit number 12. These are files with a “.PLD”. extension. It is not advisable to save data in these files. This could also cause problems in pasting design data into the input of off-design calculations.

User Defined Input

Loading data from the standard input file, unit number 9, can be handy if extra data need to be loaded in a user subroutine. Using the General Data | User Defined Input, you can enter extra data. These data are pasted behind the other data in the input file. Subsequently, these data can be read with `READ(9,…)`.
An example of this construction can be found in the "Examples" part. In example 5 (EX5, EX5A, EX5B en EX5C) of this part, extra data that have been entered with General Data | User Defined Input are read via the APSUB user subroutine.

If you want to open a file yourself for reading or writing, you can use a unit number not yet in use. You can open the file using an `OPEN` statement. If, for example, you want to write data to a file with the name MYFILE, you can do this as follows:

```
OPEN( UNIT=14, FILE='MYFILE', STATUS='UNKNOWN')
```

Then you can write everything to this file, using `WRITE(14,…)`. The file is closed again using the `CLOSE(14)` statement.

Comment:

Note! In a user subroutine, it is not allowed to re-open or close the files with the reserved unit numbers stated in this section.

Chapter 4

User Subroutines in Detail

4.1 Memory capacity to be stated by the user (CTMAIN)

The memory capacity, required not only for drafting the system matrix but also for internal bookkeeping of apparatuses, pipes, working fluids, and cycles, strongly depends on the relative process diagram. It is not practical to work with fixed upper limits for the number of apparatuses, pipes, etc., since this would lead to limitations in the program's application or to over-use of memory capacity. In standard FORTRAN it is not possible to reserve dynamic memory. Therefore, you can modify the required memory capacity in Cycle-Tempo's main program.

```
      PROGRAM CTMAIN

*                 User supplied storage reservation.
*                                       |       Increase this number if you
*                                       v       don't have enough storage

      PARAMETER (LCSIZE = 500000)
      COMMON Z (LCSIZE)
      REAL Z

*     Character variables are stored in a separate area
      CHARACTER C(20000)
      COMMON /CARR/ C

*     Start main control subroutine of Cycle-Tempo
      CALL TEMPO (LCSIZE)

      END
```

The exact memory capacity required by the program is printed in a number of 4-byte words, folllowing the output on system capacity (see “CONFIGURATION OF THE INSTALLATION” in the Text Output). If the required memory capacity is larger than the reserved memory capacity, the program will say so in the Message window by way of an error message and terminate the calculation.

You can specify the required memory capacity in Cycle-Tempo’s main program, the CTMAIN**.**FOR. file, by changing the number stated with LCSIZE in the line:

```
PARAMETER (LCSIZE = 300000)
```

according to your wishes. By default, the reserved memory capacity is 300000 words of 4 bytes. An example of a main program is presented on the previous page.

Comment

Note! *Any other change in the program text of the main program may lead to the program not working properly.*

Although CTMAIN is not a real user subroutine, the file CTMAIN**.**FOR *should be treated like a user subroutine in order to activate a change in the reserved memory capacity.*

4.2 APSUB subroutine

4.2.1 Introduction

After each main iteration, the APSUB subroutine calculates values to be modified for the apparatus data (saved in DATA array, see Appendix A).

The argument list of this subroutine contains all thermodynamic and characteristic apparatus data, calculated by Cycle-Tempo or stated for the relevant apparatus. The argument list also contains the real gas composition of each pipe.

To activate APSUB for specific apparatuses, the apparatus numbers should be entered into the input window for user subroutines (Calculation | User Subroutines) (see also section 2.3). The order in which the apparatus numbers are listed in the input window determines the order in which they can be opened from APSUB. This can be important if, for example, data need to be transferred from one apparatus to another. The apparatus supplying data must be listed in front of (the number of) the apparatus receiving data.

4.2.2 Argument list and declarations of variables

```
      SUBROUTINE APSUB ( NLIN,ILIN,IAP,ITYP,ITM,DATA,NP,IPIPE,FM,H,
    *                    P,T,X,W, C,RESULT,COMP,CONC,AVERG,IDX,NCOMP,
    *                    APNAME,NUMGEN,RDATGE)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION   DATA(50),FM(NP),H(NP,3),P(NP,3),T(NP,2),
     *                   X(NP,2),W(NP),C(NP),CONC(NLIN+7,NCOMP),
     *                   AVERG(NLIN+7),RDATGE(NUMGEN,6)

      INTEGER            IPIPE(NP),ILIN(NLIN),IDX(NCOMP)

      CHARACTER          COMP(NCOMP)*8,APNAME*12
```

Warning

The above program text is the subroutine specification. It is recommended that you do not make changes in it. A change in the above program text may lead to the program not working properly.

4.2.3 Classification and overview of the argument list

The variables in the argument list can be classified as follows:

- Input variables

NLIN	:	number of pipes
ILIN	:	pipe numbers
IAP	:	apparatus number
ITYP	:	type of apparatus
ITM	:	iteration number
NP	:	number of pipes connected with IAP
IPIPE	:	pipes connected with IAP

FM	:	mass flows in the connected pipes
H	:	enthalpies in the connected pipes
P	:	pressures in the connected pipes
T	:	temperatures in the connected pipes
X	:	vapor quality in the connected pipes
W	:	mass fraction in the connected pipes
C	:	medium type in the connected pipes
APNAME	:	apparatus name
NUMGEN	:	number of generators in the system
RDATGE	:	generator data

- Input variables in the form of composition data of working fluids for the types FUEL and GASMIX:

AVERG	:	average mole mass in the pipes
COMP	:	components in the system
CONC	:	concentrations of components in the pipes
IDX	:	identification number of the components
NCOMP	:	number of different components in the system

- Input and output variables

DATA	:	apparatus input data
RESULT	:	apparatus input data for certain types of apparatuses

Although all relevant thermodynamic data of the apparatus are offered, *not* all changes are transferred to the program from which you are operating (see statements made at input description of the relative apparatus in the “Reference” part).
Modifications entered via RESULT have a higher priority than modifications entered in the DATA data.

4.2.4 Explanation of the variables in the argument list

The overview below describes the meaning of the variables in the argument list, in alphabetical order.

APNAME = name of apparatus IAP (maximum of 12 positions)

AVERG = double precision array with NLIN+7 elements
Contains the average mole mass of the working fluid in a pipe in the process diagram (kg/mole).
The element number of the array is an identification number of a pipe in the process diagram. See also under CONC for more information about identification numbers.

C = double precision array with NP elements
Contains the medium types of the medium in the pipes connected with apparatus IAP (-). Table 4-1 presents the meaning of the value of C in relation to the medium type.
The sequence of saving is the same as that of the pipe identification numbers in IPIPE.

Table 4-1: meaning of the contents of the C array

C	Medium type
2	FUEL
1	GASMIX
-1	WATERSTM
-2	POTASIUM
-4<C<-3	DUPONT-database
-5<C<-4	REFPROP-database
-5	NH3-H2O
-8<C<-7	LIQUIDS-database

COMP = character*8 array with NCOMP elements
Contains the names of all components of GASMIX and FUEL medium type used in a process diagram.

CONC = double precision array with (NLIN+7)*NCOMP elements divided over NLIN+7 rows and NCOMP columns.

Each row contains the concentrations of components in a pipe of the process diagram (mole component / mole mixture).

The 7 extra rows can be used for virtual pipes (dummy pipes for calculation purposes).

The row number in the array is the pipe identification number in the process diagram. The third row, for example, refers to the pipe whose pipe number is identical to ILIN(3). Obviously, the virtual pipe identification numbers do not refer to a pipe in the process diagram.

The column number is the identification number of a component in a process diagram. The name of the component to which, for example, the first column refers, is identical to COMP(1).

DATA = array with the input data (maximally 50) of apparatus IAP.

If input data have been stated with “Normal Input” for apparatus IAP in the input window for apparatus data, you can modify these data here.

Appendix A states which input data are entered at which places in the DATA array for apparatus type ITYP.

FM = double precision array with NP elements.

Contains the mass flows in the pipes that are connected with apparatus IAP (kg/s).

The order of storage is the same as that of the pipe identification numbers in IPIPE.

H = double precision array with NP*3 elements divided over NP rows and 3 columns.

Contains the specific enthalpies in the pipes connected with apparatus IAP (kJ/kg).

Each row i (where i = 1,2,...,NP) contains the following pipe data:

H (i,1): enthalpy at pipe inlet;

H (i,2): enthalpy at pipe outlet;

H (i,3): enthalpy drop in the pipe.

The order of the stored pipes (rows) is the same as that of the row identification numbers in IPIPE.

IAP = the apparatus number for which the APSUB subroutine is opened.
IAP is one of the apparatus numbers stated in the input window for user subroutines (Calculation | User Subroutines).

IDX = integer array with NCOMP elements
Contains identification numbers of the components in a process diagram in relation to the gas library, i.e. the places of the components in the gas library.
The element numbers themselves are identification numbers of the components in the process diagram. This variable is used to open other subroutines and subfunctions.

ILIN = integer array with NLIN elements
Contains the pipe numbers of a process diagram as stated by the user in the order of increasing pipe number.

IPIPE = integer array with NP elements
Contains the identification numbers of the pipes in ILIN connected with apparatus IAP. The sequence of storage is as follows:

IPIPE(1........n_i): identification numbers of the inlet pipes of the primary (heated) medium;
IPIPE(n_i+1...n_j): identification numbers of the outlet pipes of the primary (heated) medium;
IPIPE(n_j+1...n_k): identification numbers of the inlet pipes of the secondary (cooled) medium;
IPIPE(n_k+1...NP):identification number of the outlet pipes of the secondary (cooled) medium.

If only one medium is involved, IPIPE(n_j+1...n_k) and IPIPE(n_k+1...NP) are not applicable, and n_j = NP.

If more inlet or outlet pipes are connected to one medium of an apparatus, they are stored in numerical order.

The identification number of a pipe number in ILIN (also called internal pipe number) indicates where the relative pipe number has been saved in the ILIN array. The actual pipe number (also called external pipe number) of, for example, the first primary outlet pipe is identical to ILIN(IPIPE(n_i+1)).

ITM = number of the main iteration in which the APSUB subroutine is opened by the program.
Usually, ITM is applied to perform certain operations either exclusively in the zeroth and/or first main iteration, or in all iterations except the zeroth and/or first. The zeroth iteration is the very first calculation round; the first iteration is the second calculation round, etc.

ITYP = type number of apparatus IAP

NCOMP = the total number of different components in a process diagram

NLIN = the total number of pipes in a process diagram

NP = the total number of pipes connected with apparatus IAP

P = double precision array with NP*3 elements divided over NP rows and 3 columns.
Contains the pressures in the pipes connected with apparatus IAP (bar).
Each row i (where i = 1,2,...,NP) contains the following pipe data:
P (i,1): pressure at pipe inlet
P (i,2): pressure at pipe outlet
P (i,3): pressure drop in the pipe
The sequence of the stored pipes (rows) is the same as that of the identification numbers of the pipes in IPIPE.

RESULT = FACUA, U*A value for apparatus type = 5, 6 or 12 (kW/K)
ETHAI, isentropic efficiency for apparatus type = 3 (turbines)
DELTAP pressure drop for apparatus type = 8; this value must be negative in order to achieve a pressure rise (bar)

T = double precision array with NP*2 elements divided over NP rows and 2 columns.

Contains the temperatures in the pipes connected with apparatus IAP (°C).

Each row i (with i = 1,2,...,NP) contains the following pipe data:

T (i,1): temperature at pipe inlet

T (i,2): temperature at pipe outlet

The sequence of the stored pipes (rows) is the same as that of the identification numbers of the pipes in IPIPE.

W = double precision array with NP elements.

Contains the mass fractions in the pipes connected with apparatus IAP (-). Only relevant for certain mixtures (ammonia-water, solutions). The sequence of the stored pipes is the same as that of the identification numbers of the pipes in IPIPE.

X = double precision array with NP*2 elements divided over NP rows and 2 columns.

Contains the vapor fractions in the pipes connected with apparatus IAP (-).

Each row i (with i = 1,2,...,NP) contains the following pipe data:

X (i, 1): vapor fraction at pipe inlet

X (i, 2): vapor fraction at pipe outlet

The sequence of the stored pipes (rows) is the same as that of the identification numbers of the pipes in IPIPE.

Comment: the elements in X have the value 0 for the FUEL and GASMIX medium types.

4.2.5 Examples

Exampe 1

In this example, the value of the RESULT variable is calculated as a function of the mass flow.

```
      SUBROUTINE APSUB ( NLIN,ILIN,IAP,ITYP,ITM,DATA,NP,IPIPE,FM,H,
     *                   P,T,X,W,C,RESULT,COMP,CONC,AVERG,IDX,NCOMP,
     *                   APNAME,NUMGEN,RDATGE)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION   DATA(50),FM(NP),H(NP,3),P(NP,3),T(NP,2),
     *                   X(NP,2),W(NP),C(NP),CONC(NLIN+7,NCOMP),
     *                   AVERG(NLIN+7),RDATGE(NUMGEN,6)

      INTEGER            IPIPE(NP),ILIN(NLIN),IDX(NCOMP)

      CHARACTER          COMP(NCOMP)*8,APNAME*12

      IF (ITM.LT.2)   RETURN
      IF (IAP.EQ.k)   RESULT = $f_1$(FM)
      IF (IAP.EQ.m)   RESULT = $f_2$(FM)

      END
```

The ’k’ and ’m’ variables are the numbers of the apparatuses stated in the row with apparatus numbers in the input window for user subroutines (Calculation | User Subroutines). For initialization, estimates must be entered into the input for the accompanying apparatus with respect to the isentropic efficiency and the pressure increase of the relative apparatuses. The estimate for the U*A value should be stated in the off-design input data of the relative apparatus.

Example 2

Another construction that is used quite often is transferring a variable from one apparatus to another, without the apparatuses being directly connected with a pipe. For this, the apparatus number (supplying the variable) must first be entered into the row with apparatus numbers in the input window for user subroutines (Calculation | User Subroutines). Then the apparatus number (receiving the variable) must be entered. In the following example a variable prop(ip,a) of apparatus 'k' is transferred to variable number 'j' of apparatus 'm'. The latter must have been entered in the input window for apparatus data of apparatus 'm'. In it, prop(ip), for example, is a mass flow (FM), pressure (P), temperature (T) or an enthalpy (H) in the connected pipe 'ip'.

If, for example, you would transfer the temperature at the end of a pipe connected with apparatus 'k', prop(ip) is identical to T(ip,2).

```
      SUBROUTINE APSUB ( NLIN,ILIN,IAP,ITYP,ITM,DATA,NP,IPIPE,FM,H,
     *                   P,T,X,W,C,RESULT,COMP,CONC,AVERG,IDX,NCOMP,
     *                   APNAME,NUMGEN,RDATGE)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION   DATA(50),FM(NP),H(NP,3),P(NP,3),T(NP,2),
     *                   X(NP,2),W(NP),C(NP),CONC(NLIN+7,NCOMP),
     *                   AVERG(NLIN+7),RDATGE(NUMGEN,6)

      INTEGER            IPIPE(NP),ILIN(NLIN),IDX(NCOMP)

      CHARACTER          COMP(NCOMP)*8,APNAME*12

      SAVE PROPTY

      IF (IAP.EQ.k) PROPTY = prop(ip)
      IF (IAP.EQ.m) DATA(j) = PROPTY

      END
```

Note: Usually, this manoeuvre is easier to perform with the PIPSUB user subroutine.

4.3 PIPSUB subroutine

4.3.1 Introduction

In this subroutine, after each main iteration, you can modify characteristic pipe data (such as pressure drop and specific enthalpy drop) that have been entered into the input. Furthermore, you can use this subroutine to calculate or modify thermodynamic variables such as pressure, temperature, and specific enthalpy in a pipe.
The argument list of this subroutine contains all relevant pipe data, as well as the real gas composition of each pipe.

To activate PIPSUB for specific pipes, enter the pipe numbers in the input window for user subroutines (Calculation | User Subroutines) (see also section 2.3).
The order in which the pipe numbers are listed in the input window determines the order in which they can be opened in PIPSUB. This can be important if data from one pipe need to be transferred to another pipe. The number of the pipe supplying data should be listed in front of the number of the pipe receiving data.

The variables to be modified in each pipe should be stated in the input as “extra condition” of the relative pipe. The modifications to be made in PIPSUB are *not* performed when the variables have also been entered with the apparatus input.
Compositions in the pipes can *only* be modified if the composition of the relative pipes is explicitly stated in the input (and not as an estimate).

4.3.2 Argument list and declarations of variables

```
     SUBROUTINE PIPSUB (IPIPE,ITM,VOLFLO,FM,H,DH,P,DP,T,X,W,C,
*                                 COMP,CONC,AVERG,IDX,NCOMP,NLIN,ILIN)
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)

     DOUBLE PRECISION    CONC(NLIN+7,NCOMP),AVERG(NLIN+7)
     INTEGER             ILIN(NLIN),IDX(NCOMP)
     CHARACTER           COMP(NCOMP)*8
```

Warning

The above program text is the subroutine specification. It is recommended not to make changes in it. A change in the above program text may lead to the program not working properly.

4.3.3 Classification and overview of the argument list

The variables in the argument list can be classified as follows:

- Input variables

IPIPE	:	pipe number
ITM	:	iteration number
VOLFLO	:	volume flow
FM	:	mass flow
W	:	mass fraction
C	:	type of medium
NLIN	:	number of pipes in the system
ILIN	:	pipe numbers in the system

- Input variables in the form of composition data of media for the FUEL and GASMIX medium types:

AVERG	:	average mole flow in the pipes
COMP	:	components in the system
CONC	:	concentrations of components in the pipes
IDX	:	identification numbers of the components
NCOMP	:	number of different components in the system

- Input and output variables

H	:	enthalpy
DH	:	enthalpy difference
P	:	pressure
DP	:	pressure drop
T	:	temperature
X	:	vapor quality

4.3.4 Explanation of the variables in the argument list

The overview below describes the meaning of the variables in the argument list, in alphabetical order.

AVERG = double precision array with NLIN+7 elements
Contains the average mole mass of the medium in a pipe in the process diagram (kg/mole).
The element number of the array is an identification number of a pipe in the process diagram. See also under CONC for more information about the identification numbers.

C = Medium type of the medium in pipe IPIPE
Table 4-2 presents the meaning of the value of C in relation to the medium type.

Table 4-2: meaning of the contents of the C array

C	Medium type
2	FUEL
1	GASMIX
-1	WATERSTM
-2	POTASIUM
-4<C<-3	DUPONT-database
-5<C<-4	REFPROP-database
-5	NH3-H2O
-8<C<-7	LIQUIDS-database

COMP = character*8 array with NCOMP elements
Contains the names of all components of GASMIX and FUEL medium types used in a process diagram.

CONC = double precision array with (NLIN+7)*NCOMP elements divided over NLIN+7 rows and NCOMP columns.
Each row contains the concentrations of components in a pipe of the process diagram (mole component / mole mixture).
The 7 extra rows can be used for virtual pipes (dummy pipes for calculation purposes).
The row number in the array is the identification number of a pipe in the process diagram. For example, the third row refers to the pipe whose pipe number is identical to ILIN(3). The virtual pipe identification numbers naturally do not refer to a pipe in the process diagram.
The column number is the identification number of a component in a process diagram. The name of the component to which, for example, the first column refers, is identical to COMP(1).

DH = specific enthalpy drop in pipe IPIPE (kJ/kg)

DP = pressure drop in pipe IPIPE (bar) or (-)
If DP is larger than or identical to 500, the pressure drop is an absolute pressure drop and the actual pressure drop is
= DP – 1000.
If DP is smaller than 500, the pressure drop is the relative pressure drop.

This is defined as (pin – pout)/pin.
A correct value for a relative pressure drop is then between 0 and 1.

FM = mass flow in pipe IPIPE (kg/s)

H = specific enthalpy at the inlet of pipe IPIPE (kJ/kg)

IDX = integer array with NCOMP elements
Contains identification numbers of the components in a process diagram in relation to the gas library, or the places of the components in the gas library.
The element numbers themselves are identification numbers of the components in the process diagram.
This variable is used in calling other subroutines and subfunctions.

ILIN = integer array with NLIN elements.
Contains the pipe numbers of a process diagram stated by the user, in the sequence of increasing pipe number.

IPIPE = the pipe number for which the PIPSUB subroutine is called.
IPIPE is one of the pipe numbers stated in the input window for user subroutines (Calculation | User Subroutines).

ITM = number of the main iteration in which the APSUB subroutine is opened by the program.
Usually, ITM is applied to perform certain operations either exclusively in the zeroth and/or first main iteration, or in all iterations except the zeroth and/or first. The zeroth iteration is the very first calculation round; the first iteration is the second calculation round, etc.

NCOMP = the total number of different components in a process diagram.

NLIN = the total number of pipes in a process diagram.

P = pressure at the inlet of pipe IPIPE (bar)

T = temperature at the inlet of pipe IPIPE (°C)

VOLFLO = volume flow at the inlet of pipe IPIPE (m3/s)

W = mass fraction in pipe IPIPE (-)
Only relevant for certain mixtures (ammonia-water, solutions)

X = vapor fraction at the inlet of pipe IPIPE (-)
Note: the elements in X have the value 0 for the FUEL and GASMIX medium types.

4.3.5 Example

For pipe k, the pressure drop is a function of the mass flow: $\Delta p = f_1 (\Phi_m)$.
For pipe m, the enthalpy drop is a function of the mass flow: $\Delta h = f_2 (\Phi_m)$. This leads to the following PIPSUB subroutine:

```
      SUBROUTINE PIPSUB (IPIPE,ITM,VOLFLO,FM,H,DH,P,DP,T,X,W,C,
     *                   COMP,CONC,AVERG,IDX,NCOMP,NLIN,ILIN)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION   CONC(NLIN+7,NCOMP),AVERG(NLIN+7)
      INTEGER            ILIN(NLIN),IDX(NCOMP)
      CHARACTER          COMP(NCOMP)*8

        IF (ITM.LT.2)  RETURN
        IF (L.EQ.k)    DP = f1(FM)
        IF (L.EQ.m)    DH = f2(FM)

      END
```

4.4 USREAC subroutine

4.4.1 Introduction

In the chemical reactor (type 27), you can calculate a new equilibrium for a chemical reaction based on an equilibrium temperature, using the equilibrium constants. Several reactions can also be brought to equilibrium simultaneously, at different equilibrium temperatures.

The available reactions are:

- Water gas shift reaction: $CO + H_2O \leftrightarrow CO_2 + H_2$
- CH_4-reforming reaction: $CH_4 + H_2O \leftrightarrow CO + 3\ H_2$
- User-programmed reactions

The USREAC subroutine is used for entering the user-programmed reactions.

4.4.2 Argument list and declarations of variables

```
SUBROUTINE USREAC (IAP, TRUSER, PPART)

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

DOUBLE PRECISION    PPART(100)
CHARACTER           NAME*8
COMMON /GASDAT/     COEF(2,7,100),WMOL(100),SIGMA(100),EPSS(100),NAME(100)
```

4.4.3 Explanation of the variables

The meaning of the variables in the argument list is as follows:

IAP = apparatus number of the reactor

TRUSER = reaction temperature stated by the user (°C)

PPART = array with concentrations of ambient partial pressures in the reactor of all components available in the Cycle-Tempo gas library [bar]

The COMMON-area GASDAT is stated here only for the purpose of having the NAME array available. The meaning of the other variables is not important here.

NAME = character*8 array with the names of the components in the gas library

4.4.4 The operation of USREAC

The intention is that you modify the PPART array in such a way that equilibrium is reached for the reaction(s). In this array, you will find the partial pressures of all substances available in Cycle-Tempo's gas library, as they are found in the reactor (total pressure is PREACT, or if this pressure is not stated, the outlet pressure). If a substance does not occur in the reactor, its partial pressure is zero. In the `NAME` array you can look up where substances can be found.
The programmed reaction may also be a terminating reaction.

For further information about calculating equilibrium, see the "Technical Notes" part.

4.4.5 Example: COS conversion

In this example, all COS of a gas mixture is converted in conformity with the following reaction:

$$COS + H_2O \leftrightarrow CO_2 + H_2S$$

The USREAC subroutine looks as follows:

```
      SUBROUTINE USREAC (IAP, TRUSER, PPART)

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      DOUBLE PRECISION     PPART(100)
      CHARACTER            NAME*8
      COMMON /GASDAT/      COEF(2,7,100),WMOL(100),SIGMA(100),EPSS(100),NAME(100)

*   Select species concerning the reaction
      DO 10 I=1,100
      IF (NAME(I).EQ.'COS') ISPEC1=I
      IF (NAME(I).EQ.'H2O') ISPEC2=I
      IF (NAME(I).EQ.'H2S') ISPEC3=I
      IF (NAME(I).EQ.'CO2') ISPEC4=I
10    CONTINUE

*     Get reaction coordinate (all COS is converted)
      X=PPART(ISPEC1)

*     Calculate new partial pressures
      PPART(ISPEC1) = PPART(ISPEC1) – X
      PPART(ISPEC2) = PPART(ISPEC2) – X
      PPART(ISPEC3) = PPART(ISPEC3) + X
      PPART(ISPEC4) = PPART(ISPEC4) + X

      END
```

In the user subroutine, the numbers of the components are first looked up using the NAME array, and saved in ISPEC1, ISPEC2, ISPEC3 and ISPEC4, respectively. Subsequently, the reaction coordinate X is determined. Finally, the new partial pressures are determined.

4.5 FUNCOP user function

4.5.1 Application

In this user subroutine, you can define a function. An optimization is made according to the function value produced by this function, if the “Use user-defined optimization function” is selected in the input window for optimization data.

4.5.2 Argument list and declarations of variables

The first statements should look like this:

```
 FUNCTION FUNCOP (     NLIN,NAPP,DATAP,PH,MKGS,HKJK,PBAR,TCEL,XFAC,
&                      IAPP,ILIN)

   IMPLICIT DOUBLE PRECISION (A-H,O-Z)

   DOUBLE PRECISION      DATAP(NAPP,50), PH(NAPP), MKGS(NLIN),
 *                HKJK(NLIN,3), PBAR(NLIN,3), TCEL(NLIN,2),
 *                XFAC(NLIN,2)

    INTEGER                IAPP(NAPP),ILIN(NLIN)
```

4.5.3 Explanation of the variables in the argument list

The overview below describes the meaning of the variables in the argument list in alphabetical order.

DATAP = double precision array with NAPP*50 elements divided among NAPP rows 50 columns.
Contains the input data of all apparatuses entered into the window for apparatus input data at "Normal Input". Appendix A states at which places

in the DATAP array which input data are found for a specific type of apparatus.

Each row contains the data of an apparatus. The row number is the identification number of the apparatus in IAPP. The data of the first row number, for example, belong to apparatus number IAPP(1).

HKJK = double precision array with NLIN*3 elements divided among NLIN rows and 3 columns.

Contains the specific enthalpies in the pipes in the process diagram (kJ/kg).

Each row i (with i = 1,2,...,NLIN) contains the following data about the pipe:

- HKJK (i, 1): enthalpy at the pipe inlet
- HKJK (i, 2): enthalpy at the pipe outlet
- HKJK (i, 3): enthalpy drop in a pipe.

The row number is the identification number of a pipe in ILIN.

IAPP = integer array with NAPP elements

Contains the user-stated apparatus numbers of a process diagram in the following sequence:

1. Turbines (type 3) in numerical order
2. Condensers (type 4) in numerical order
3. Feedwater pre-heaters (type 5) in numerical order
4. Heat exchangers (EEQCOD = 2, type 6) in numerical order
5. Heat exchangers (EEQCOD = 1, type 12) in numerical order
6. Moisture separators (type 22) in numerical order
7. Other types of apparatus, in numerical order

ILIN = integer array with NLIN elements

Contains the user-stated pipe numbers of a process diagram in numerical order

MKGS = double precision array with NLIN elements

Contains the mass flows in the pipes in the process diagram (kg/s).

The element number is the identification number of a pipe in ILIN.

NAPP = the total number of apparatuses in a process diagram

NLIN = the total number of pipes in a process diagram

PBAR = double precision array with NLIN*3 elements divided among NLIN rows and 3 columns.
Contains the pressures in the pipes in the process diagram (bar).
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:
PBAR (i,1): pressure at the pipe inlet
PBAR (i,2): pressure at the pipe outlet;
PBAR (i,3): pressure drop in the pipe
The row number is the identification number of a pipe in ILIN.

PH = double precision array with NAPP elements
Contains the energy exchange of the apparatuses with the surrounding area, as a result of the calculations (kW).
The element number is the identification number of an apparatus in IAPP.

TCEL = double precision array with NLIN*2 elements divided among NLIN rows and 2 columns.
Contains the temperatures in the pipes in the process diagram (°C).
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:
TCEL (i,1): temperature at the pipe inlet
TCEL (i,2): temperature at the pipe outlet
The row number is the identification number of a pipe in ILIN.

XFAC = double precision array with NLIN*2 elements divided among NLIN rows and 2 columns.
Contains the vapor fractions in the pipes in the process diagram (-).
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:
XFAC (i,1): vapor fraction at the pipe inlet
XFAC (i,2): vapor fraction at the pipe outlet.
The row number is the identification number of a pipe in ILIN.

Comment: the elements in XFAC have the value 0 for the FUEL and GASMIX types of working fluid.

4.6 EPILOG subroutine

4.6.1 Introduction

The last program statement to be performed concerns the EPILOG. subroutine. With the argument list all available data are passed on. This produces the possibility of “post processing" such as printing tables or plotting graphs. After this routine, no other operations are performed so that you are completely free in using the data.

4.6.2 Argument list and declaration of variables

The first statements should look as follows:

```
SUBROUTINE EPILOG ( NAPP,NLIN,NCYCLE,NTURB,NTDP,NUMGEN,NCOMP,
*                   IAPP,ILIN,DATAP,MED,PBAR,HKJK,TCEL,XFAC,
*               XGAS,MKGS,PH,PTTT,FACUA,DELTDF,RDATGE,
*               ITYPE,NAMAP,NLIK,M,COMP,CONC,AVERG,IDX,
*               NUMGEO,EXKJKG,WMAS,HENV,HUC,IGEO,IDATGM,
*               RDATGM,EXDIFA,EXDIFP,HUCH,PPEXCH)

  IMPLICIT DOUBLE PRECISION (A-H,O-Z)

  DOUBLE PRECISION     DATAP(NAPP,50),PBAR(NLIN,3),
*               HKJK(NLIN,3),TCEL(NLIN,2),XFAC(NLIN,2),
*               XGAS(NLIN),MKGS(NLIN),PH(NAPP),
*               PTTT(NAPP),FACUA(3,*),DELTDF(2,*),
*               RDATGE(NUMGEN,6),CONC(NLIN+7,NCOMP),
*               AVERG(NLIN+7),RDATGM(NUMGEO,10),
*               EXKJKG(NLIN,2),WMAS(NLIN),HENV(NLIN,2),
*               HUC(NLIN),EXDIFA(NAPP,2),EXDIFP(NLIN,2),
*                       HUCH(NLIN),PPEXCH(NLIN)

  INTEGER              IAPP(NAPP),ILIN(NLIN),MED(NLIN),
*               ITYPE(NAPP),NLIK(NAPP,2),M(NLIN,2),
*               IDX(NCOMP),IGEO(NUMGEO),IDATGM(NUMGEO,5)

  CHARACTER            COMP(NCOMP)*8,NAMAP(NAPP)*12
```

Warning

The above program text is the subroutine specification. It is recommended that no changes be made in it. A change in the above program text may lead to the program not working properly.

4.6.3 Classification and overview of the argument list

The variables in the argument list can be classified as follows:

- Apparatus data

DATAP	:	apparatus input data
DELTDF	:	temperature differences at apparatus inlet and outlet
FACUA	:	off-design data of heat exchangers
IAPP	:	apparatus numbers
ITYPE	:	apparatus types
NAPP	:	number of apparatuses in the system
NAMAP	:	apparatus names
NTURB	:	number of turbines in the system
NUMGEN	:	number of generators in the system
PH	:	energy exchange with environment
PTTT	:	transmitted heat
RDATGE	:	generator data

- Exergy data in the system

EXKJKG	:	total exergy of the working fluids in the pipes
EXDIFA	:	exergy difference of apparatuses
EXDIFP	:	exergy difference of pipes
PPEXCH	:	chemical exergy of the working fluids in the pipes

- Geometry data of apparatuses

IGEO	:	numbers of apparatuses with geometry data
IDATGM	:	integer input data for geometry
NUMGEO	:	number of apparatuses with geometry data
RDATGM	:	double precision input data for geometry

- Pipe data

HKJK	:	specific enthalpy
HENV	:	specific enthalpy with respect to the environment
HUC	:	lower heating value
HUCH	:	higher heating value
ILIN	:	pipe numbers
MED	:	cycle numbers
MKGS	:	mass flow
NLIN	:	number of pipes
NTDP	:	number of mechanical axes
PBAR	:	pressure
TCEL	:	temperature
WMAS	:	mass fraction
XFAC	:	vapor fraction
XGAS	:	type of working fluid

- Composition data of working fluids for the medium types FUEL and GASMIX:

AVERG	:	average mole mass in the pipes
COMP	:	components in the system
CONC	:	concentration of components in the pipes
IDX	:	identification numbers of the components
NCOMP	:	number of different components in the system

- Other data

M	:	apparatuses at inlet and outlet of pipes
NLIK	:	pipes at the primary medium of apparatuses
NCYCLE	:	number of cycles within the system

4.6.4 Explanation of the variables in the argument list

The overview below describes the meaning of the variables in the argument list in alphabetical order.

AVERG = double precision array with NLIN+7 elements.
Contains the average mole mass of the working fluid in a pipe in the process diagram (kg/mole).
The element number of the array is the identification number of a pipe in the process diagram. See also under CONC for more information about identification numbers.

Note After completing the calculation and thus loading the EPILOG routine, the average mole masses of all pipes are stored in AVERG array, i.e., not only those of the pipes with medium type FUEL and GASMIX.

COMP = character*8 array with NCOMP elements.
Contains the names of all used components of the medium type GASMIX and FUEL in a process diagram.

CONC = double precision array with (NLIN+7)*NCOMP elements divided among NLIN+7 rows and NCOMP columns.
Each row contains the concentrations of components in a pipe of the process diagram (mole component / mole mixture).
The 7 extra rows can be used for virtual pipes (dummy pipes for calculation purposes).
The row number in the array is the identification number of a pipe in the

process diagram. For example, the third row refers to the pipe whose pipe number is identical to ILIN(3). The virtual pipe identification numbers naturally do not refer to a pipe in the process diagram.
The column number is the identification number of a component in a process diagram. The name of the component to which for example the first column refers is identical to COMP(1).

DATAP = double precision array with NAPP*50 elements divided among NAPP rows 50 columns.
Contains the input data of all apparatuses entered into the window for apparatus input data with "Normal Input". Appendix A states which data of a specific apparatus type are found at which places in the DATAP array. Each row contains the data of an apparatus. The row number is the identification number of the apparatus in IAPP. The data of the first row number, for example, belong to the apparatus number IAPP(1).

DELTDF = double precision array with 2*k elements divided among 2 rows and "k" columns. The number of “k” columns is identical to the number of heat exchangers present (types 4, 5, 6, 12 and 22). Contains the inlet and outlet temperature differences for the heat exchangers in the process diagram.

The rows contain the following data in case of counter-flow:

- DELTDF (1, *): temperature difference between the secondary inlet and the primary outlet side (ΔT_h);
- DELTDF (2, *): temperature difference between the secondary outlet and the primary inlet side (ΔT_l).

The rows contain the following data in case of co-flow (only for types 6 and 12):

- DELTDF (1, *): temperature difference between the secondary inlet and the primary inlet side (ΔT_h);
- DELTDF (2, *): temperature difference between the secondary outlet and the primary outlet side (ΔT_l).

See FACUA above, for how to store.

EXDIFA = double precision array with NAPP*2 elements divided among NAPP rows and 2 columns.

Contains the exergy differences between all ingoing and all outgoing process flows of an apparatus (kJ/s).

Each row i (where i = 1,2,...,NAPP) contains the following data about the apparatus:

- EXDIFA (i, 1): exergy difference over apparatus calculated using ingoing and outgoing exergy flows;
- EXDIFA (i, 2): same, now calculated via enthalpy and entropy values of the ingoing and outgoing process flows.

The row number is the identification number of an apparatus in IAPP.

EXDIFP = double precision array with NLIN*2 elements divided among NLIN rows and 2 columns.

Contains the exergy differences between the inlet and the outlet of the pipes in the process diagram (kW).

Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:

- EXDIFP (i, 1): exergy difference over pipe calculated using ingoing and outgoing exergy flows;
- EXDIFP (i,2): same, now calculated via enthalpy and entropy values at the inlet and outlet of the pipe.

The row number is the identification number of a pipe in ILIN.

EXKJKG = double precision array with NLIN*2 elements divided among NLIN rows and 2 columns.

Contains the total (= chemical + thermomechanical) exergy of the working fluids in the pipes of the process diagram (kJ/kg).

Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:

- EXKJKG (i, 1): total exergy at the inlet of a pipe;
- EXKJKG (i, 2): total exergy at the outlet of a pipe.

The row number is the identification number of a pipe in ILIN.

FACUA = double precision array with 3*k elements divided among 3 rows and "k" columns. The number of "k" columns is identical to the number of heat exchangers present (types 4, 5, 6, 12 and 22).

The order of storage the heat exchangers is:

1. Condensers (type 4) in numerical order
2. Feedwater pre-heaters (type 5) in numerical order
3. Heat exchangers (EEQCOD = 2, type 6) in numerical order
4. Heat exchangers (EEQCOD = 1, type 12) in numerical order
5. Moisture separators (type 22) in numerical order

The rows contain the following data about condensers (type = 4):

- FACUA (1, *): heat-exchanging surface A (m^2);

The second and third rows are not used with condensers.

The rows contain the following data about feedwater pre-heaters (type = 5):

- FACUA (2, *): design mass flow (kg/s);

The first and third rows are not used with feedwater pre-heaters.

The rows contain the following data about heat exchangers (type = 6: EEQCOD = 2, or type = 12: EEQCOD = 1):

- FACUA (1, *): U ∗ A value (kW/K);
- FACUA (2, *): design mass flow (kg/s);
- FACUA (3, *): exponent ETHA for U * A correction.

HENV = double precision array with NLIN*2 elements divided among NLIN rows and 2 columns.
Contains the specific enthalpy values with respect to the environment in the pipes in the process diagram (kJ/kg).
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:

- HENV (i, 1): enthalpy with respect to the environment at the inlet of a pipe;

- HENV (i, 2): enthalpy with respect to the environment at the outlet of a pipe.

The row number is the identification number of a pipe in ILIN.

HKJK = double precision array with NLIN*3 elements divided among NLIN rows and 3 columns.

Contains the specific enthalpies in the pipes in the process diagram (kJ/kg). Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:

- HKJK (i, 1): enthalpy at the inlet of a pipe;
- HKJK (i, 2): enthalpy at the outlet of a pipe;
- HKJK (i, 3): enthalpy drop in a pipe.

The row number is the identification number of a pipe in ILIN.

HUC = double precision array with NLIN elements.

Contains the lower heating value of the working fluids in the pipes in the process diagram (kJ/kg). The element number is the identification number of a pipe in ILIN.

HUCH = double precision array with NLIN elements.

Contains the higher heating value of the working fluids in the pipes in the process diagram (kJ/kg). The element number is the identification number of a pipe in ILIN.

IAPP = integer array with NAPP elements.

Contains the user-stated apparatus numbers of a process diagram in the following order:

1. Turbines (type 3) in numerical order
2. Condensers (type 4) in numerical order
3. Feedwater pre-heaters (type 5) in numerical order
4. Heat exchangers (EEQCOD = 2, type 6) in numerical order
5. Heat exchangers (EEQCOD = 1, type 12) in numerical order
6. Moisture separators (type 22) in numerical order
7. Other apparatus types, in numerical order

IDATGM = integer array with NUMGEO*5 elements divided among NUMGEO rows and 5 columns.
Contains the integer geometry data entered with the apparatuses. Table 4-3 presents the contents of this array.

Table 4-3: the contents of the "IDATGM" array for the relevant apparatus types

Type	Name	1	2	3	4	5
4	condenser	NPIPES	NEDGE		PRISEC	
21	fuel cell		FLOW	NITER	NCELL	PGFC

IDX = integer array with NCOMP elements.
Contains identification numbers of the components in a process diagram in relation to the gas library, or the places of the components in the gas library.
The element numbers themselves are identification numbers of the components in the process diagram.

IGEO = integer array with NUMGEO elements.
Contains the numbers of the apparatuses for which geometry data have been entered in numerical order.

ILIN = integer array with NLIN elements.
Contains the user-stated pipe numbers of a process diagram in numerical order.

ITYPE = integer array with NAPP elements.
Contains the type numbers of the apparatuses in the process diagram.
The element number is the identification number of an apparatus in IAPP.

M = integer array with NLIN*2 elements divided among NLIN rows and 2 columns.
This array is the connection matrix of the process diagram.
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:

- M (i,1) = element number of the apparatus in IAPP at the pipe inlet ILIN(i);
- M (i,2) = element number of the apparatus in IAPP at the pipe outlet ILIN(i);

The row number is therefore the identification number of a pipe in ILIN. If, for example, M(5,2) = 3, this means that the apparatus number at the pipe outlet with pipe number ILIN(5) is identical to IAPP(3).

MED = integer array with NLIN elements.
Contains the numbers of the cycles of which the pipes are part.
The element number is the identification number of the pipe in ILIN.

MKGS = double precision array with NLIN elements.
Contains the mass flows in the pipes in the process diagram (kg/s).
The element number is the identification number of a pipe in ILIN.

NAMAP = character*12 array with NAPP elements.
Contains the names of the apparatuses in the process diagram.
The element number is the identification number of an apparatus in IAPP.

NAPP = the total number of apparatuses in a process diagram

NCOMP = total number of different components in the process diagram

NCYCLE = the total number of independent cycles in a process diagram

NLIK = integer array with NAPP*2 elements divided among NAPP rows and 2 columns.
Contains the pipe identification numbers of the pipes that are connected to the primary working fluid of an apparatus.
Each row i (where i = 1,2,...,NAPP) contains the following data:

- NLIK (i, 1): pipe number of the outgoing pipe;
- NLIK (i, 2): pipe number of the ingoing pipe.

The row number is the identification number of an apparatus in IAPP.

NLIN = the total number of pipes in a process diagram

NTDP = the total number of turbine / pump or compressor combinations in the process diagram, i.e. the number of mechanical axes

NTURB = the total number of turbines in the process diagram, or the number of apparatuses for which it applies *type = 3*.

NUMGEN = the total number of generators in the process diagram

NUMGEO = the number of apparatuses for which geometry data have been entered (only possible with condenser, type = 4 and fuel cell, type = 21)

PBAR = double precision array with NLIN*3 elements divided among NLIN rows and 3 columns.
Contains the pressures in the pipes in the process diagram (bar).
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:
PBAR (i,1): pressure at the inlet of the pipe;
PBAR (i,2): pressure at the outlet of the pipe;
PBAR (i,3): pressure drop in the pipe.
The row number is the identification number of a pipe in ILIN.

PH = double precision array with NAPP elements.
Contains the energy exchange with the environment apparatuses, as a result of the calculations (kJ/s).
The element number is the identification number of an apparatus in IAPP.

PPEXCH = double precision array with NLIN elements.
Contains the chemical exergy of the working fluids in the pipes in the process diagram (kJ/kg). The element number is the identification number of a pipe in ILIN.

PTTT = double precision array with NAPP elements.
Contains the heat transferred in the apparatuses in the process diagram (kW).
The element number is the identification number of an apparatus in IAPP.

RDATGE = double precision array with NUMGEN*6 elements divided among NUMGEN rows and 6 columns.
Contains the data of the generators in the process diagram.
Each row i (where i = 1,2,...,NUMGEN) contains the following data about the genereator:

- RDATGE (i,1): efficiency (ETAGEN)
- RDATGE (i,2): cosinus ϕ (COSPHI)
- RDATGE (i,3): capacity (GENMVA)
- RDATGE (i,4): capacity / power ratio (CPRATI)
- RDATGE (i,5): mechanical losses (calculated)
- RDATGE (i,6): electrical losses (calculated)

The order of saving generators (rows) is identical to the order of decreasing generator number.

RDATGM = double precision array with NUMGEO*10 elements divided among NUMGEO rows and 10 columns. The columns 9 and 10 are not used at the moment.
Contains the double precision geometry input data entered with the apparatuses. Table 4-4 contains the contents of this array.

Table 4-4: the contents of the "RDATGM" array for the relevant types of apparatuses

Type	Name	1	2	3	4	5	6	7	8
4	condenser	DIAIN	DIAOUT	PITCH	LAMBDW	AIRFAC	RFOUL		
21	fuel cell	RCELL	VCELL	DELU	CDENS	ACELL		ESTVLT	ESTCDN

TCEL = double precision array with NLIN*2 elements divided among NLIN rows and 2 columns.
Contains the temperatures in the pipes in the process diagram (°C).
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:
TCEL (i,1): temperature at the inlet of a pipe;
TCEL (i,2): temperature at the outlet of a pipe.
The row number is the identification number of a pipe in ILIN.

WMAS = double precision array with NLIN elements.
Contains the mass fractions in the pipes in the process diagram (-).
Only relevant for certain mixtures (ammonia-water, solutions).
The element number is the identification number of a pipe in ILIN.

XFAC = double precision array with NLIN*2 elements divided among NLIN rows and 2 columns.
Contains the vapor fractions in the pipes in the process diagram (-).
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:
XFAC (i,1): vapor fraction at the inlet of the pipe;
XFAC (i,2): vapor fraction at the outlet of the pipe.
The row number is the identification number of a pipe in ILIN.

Comment: the elements in XFAC have the value 0 for the medium types FUEL and GASMIX.

XGAS = double precision array with NLIN elements.
Contains the medium types of the working fluid in the pipes in the process diagram. Table 4-5 gives the meaning of the value of XGAS in relation to the medium type.
The element number is the identification number of a pipe in ILIN.

Table 4-5: meaning of the contents of the XGAS array

XGAS	**Medium type**
2	FUEL
1	GASMIX
-1	WATERSTM
-2	POTASIUM
-4<XGAS<-3	DUPONT-database
-5<XGAS<-4	REFPROP-database
-5	NH3-H2O
-8<XGAS<-7	LIQUID-database

If the user adds no EPILOG routine, an internal dummy routine is activated.

Chapter 5

Available Functions

5.1 Introduction

This chapter contains an overview of a number of functions that can be applied from a user subroutine:

1. Useful functions for working with apparatuses
2. Useful functions for working with pipes
3. Functions for calculating thermodynamic properties

All variables stated meet the default FORTRAN convention, unless stated otherwise. This means that variables starting with the letters I through N are variables of the INTEGER type. The others are floating point variables. Since Cycle-Tempo fully calculates in double precision, these are of the DOUBLE PRECISION type.

5.2 Useful functions for working with apparatuses

When working with user subroutines, you often need the serial number of an apparatus in the IAPP array. With this serial number, you can, for example, use data from the DATAP array. The first possibility to determine the serial number is to write a DO loop in which IAPP(I) is compared with the relative apparatus number. If the condition is true, I is the required serial number. This is displayed in the code fragment below.

```
          SUBROUTINE … (…, IAPP, NAPP, …)

          INTEGER IAPP(NAPP)

          ………

*         Find apparatus number 25
          IAPNO = 0
          DO 10 I = 1, NAPP
             IF ( IAPP(I).EQ.25) THEN
                IAPNO = I
          GO TO 20
             ENDIF
   10     CONTINUE

   20     ………

          END
```

The second method is to use the INAPNO function:

```
    INTEGER FUNCTION INAPNO (IAPP,NAPP,IAPNO)
```

This directly produces the serial number of apparatus with IAPNO number. Instead of a DO loop, the internal INAPNO function is used in the code fragment below.

```
          SUBROUTINE … (…, IAPP, NAPP, …)

          INTEGER IAPP(NAPP)

          ………

*         Find apparatus number 25
          IAPNO =INAPNO( IAPP, NAPP, 25)

          ………

          END
```

If the apparatus involved does not exist, the INAPNO function will return the zero value.

Overview of the available useful functions for working with apparatuses:

- `INTEGER FUNCTION INAPNO (IAPP,NAPP,IAPNO)`

  If IAPNO exists in the process diagram, INAPNO will return the serial number of IAPNO in IAPP. If IAPNO does not exist, INAPNO will return the value 0.

- `LOGICAL FUNCTION ISAPNO (IAPP,NAPP,IAPNO)`

  If IAPNO exists in the process diagram, ISAPNO will return the value.TRUE.; otherwise,.FALSE.

- `CHARACTER*12 FUNCTION APNAME (NAMAP,IAPP,NAPP,IAPNO)`

  If IAPNO exists in the process diagram, APNAME will return the name of IAPNO. If IAPNO does not exist, APNAME will return the value ‘?’.

- `INTEGER FUNCTION APTYPE (ITYPE,IAPP,NAPP,IAPNO)`

  If IAPNO exists in the process diagram, APTYPE will give the apparatus type of IAPNO. If IAPNO does not exist, APTYPE will return the value 0.

In the functions stated above, IAPNO is always the number of the apparatus in the diagram. The meaning of the other variables is:

IAPP = integer array with NAPP elements.
Contains the apparatus numbers of a process diagram in the following order:

1. Turbines (type 3) in numerical order
2. Condensers (type 4) in numerical order
3. Feedwater pre-heaters (type 5) in numerical order
4. Heat exchangers (EEQCOD = 2, type 6) in numerical order
5. Heat exchangers (EEQCOD = 1, type 12) in numerical order
6. Moisture separators (type 22) in numerical order
7. Other types of apparatuses in numerical order

ITYPE = integer array with NAPP elements.
Contains the type numbers of the apparatuses in the process diagram.
The element number is the identification number of an apparatus in IAPP.

NAMAP = character*12 array with NAPP elements.
Contains the names of the apparatuses in the process diagram.
The element number is the identification number of an apparatus in IAPP.

NAPP = the total number of apparatus in a process diagram

5.3 Useful functions for working with pipes

When working with user subroutines, you often need the serial number of a pipe in the ILIN array. The first possibility to determine the serial number is to write a DO loop in which ILIN(L) is compared with the relevant pipe number, as is displayed in the first code fragment on page 5-2. The second method is to use the INPINO function:

```
INTEGER FUNCTION INPINO (ILIN,NLIN,IPINO)
```

This directly produces the serial number of the pipe with IPINO number. The code fragment below contains an example of the use of INPINO.

```
            SUBROUTINE … (…, ILIN, NLIN, …, PBAR, …)

            DOUBLE PRECISION PBAR(NLIN,3)
            INTEGER ILIN(NLIN)

            ………

*           Find pipe number 19
            L = INPINO( ILIN, NLIN, 19)


*           Get pressure at entrance of pipe L
            P = PBAR(L,1)
            ………

            END
```

If the relevant pipe does not exist, the INPINO function will return the value zero.

Overview of the available useful functions for working with pipes:

- `INTEGER FUNCTION INPINO (ILIN,NLIN,IPINO)`

  If IPINO exists in the process diagram, INPINO will return the serail number of IPINO in ILIN. If IPINO does not exist, INPINO will return the value 0.

- `LOGICAL FUNCTION ISPINO (ILIN,NLIN,IPINO)`

  If IPINO exists in the process diagram, ISPINO will return the value .TRUE., otherwise .FALSE.

- `LOGICAL FUNCTION ISAPPI( IAPP, ILIN, NAPP, NLIN, IAPNO, IPINO, M)`

  If apparatus IAPNO and pipe IPINO are connected with each other, ISAPPI will return the value .TRUE., otherwise .FALSE..
  If IAPNO or IPINO do not exist, ISAPPI will also return .FALSE..

In the above functions, IPINO is always the number of the pipe in the diagram and IAPNO is always the number of the apparatus in the diagram. The meaning of the other variables is:

IAPP = integer array with NAPP elements.
Contains the apparatus numbers of a process diagram in the following order:

1. Turbines (type 3) in numerical order
2. Condensers (type 4) in numerical order
3. Feedwater pre-heaters (type 5) in numerical order
4. Heat exchangers (EEQCOD = 2, type 6) in numerical order
5. Heat exchangers (EEQCOD = 1, type 12) in numerical order
6. Moisture separators (type 22) in numerical order
7. Other types of apparatuses in numerical order

ILIN = integer array with NLIN elements.
Contains the user-stated pipe numbers of a process diagram in numerical order.

M = integer array with NLIN*2 elements divided among NLIN rows and 2 columns.
This array is the connection matrix of the process diagram.
Each row i (where i = 1,2,...,NLIN) contains the following data about the pipe:

- M (i,1) = element number of the apparatus in IAPP at the pipe inlet (i);
- M (i,2) = element number of the apparatus in IAPP at the pipe outlet ILIN(i);

Thus, the row number is the identification number of a pipe in ILIN.
If, for example, M(5,2) = 3, this means that the apparatus number at the pipe outlet with pipe number ILIN(5) is identical to IAPP(3).

NAPP = the total number of apparatuses in a process diagram

NLIN = the total number of pipes in a process diagram

5.4 Functions for thermodynamic properties

This section contains an overview of functions that can be called to calculate thermodynamic properties. The meaning of the variables used in the argument lists in the column ‘way of utilization’ is as follows:

H	=	enthalpy [kJ/kg]
P	=	pressure [bar]
S	=	entropy [kJ/kg K]
T	=	temperature [°C]
W	=	mass fraction [-]
X	=	vapor fraction [-]
XGAS	=	medium type (see Table 5-1)

Table 5-1: meaning of the contents of the XGAS variable

XGAS	Medium type
2	FUEL
1	GASMIX
-1	WATERSTM
-2	POTASIUM
-4<XGAS<-3	DUPONT-database
-5<XGAS<-4	REFPROP-database
-5	NH3-H2O
-8<XGAS<-7	LIQUID-database

See section 4.6 "EPILOG" for the meaning of COMP, CONC, AVERG, IDX, NCOMP, NLIN and HUC. These variables are used to transfer the composition of a gas mixture (GASMIX or FUEL) to the relevant function. The same applies to W. This is also a composition parameter, but for certain mixtures (NH3-H2O, solutions). The L variable is the serial number of a pipe in ILIN (see also section 4.6 "EPILOG"). L is used for error messages and also to transfer the composition of a gas mixture (GASMIX or FUEL).

If a specific function is applied to a non-valid medium type, the program will generate an error message and the calculation is terminated.

Function	Description	Usage	Valid medium types
cp = f(p,t,comp.)	c_p value as a function of pressure, temperature, and composition	CP = Cp_PT(P, T, W, XGAS, L)	WATERSTM Refrigerants NH3-H2O
dv = f(p,t,comp.)	dynamic viscosity as a function of pressure, temperature and composition	DV = DV_PT(P ,T, W, XGAS, L)	WATERSTM Refrigerants [1)] NH3-H2O
dv = f(t,x,comp.)	dynamic viscosity as a function of temperature, vapor fraction and composition	DV = DV_TX (T, X, W, XGAS, L)	WATERSTM POTASIUM Refrigerants [1)] NH3-H2O
h = f(p,s,comp.)	enthalpy as a function of pressure, entropy and composition	H = HPSW(P, S ,W, XGAS, L, COMP, CONC, AVERG, IDX, NCOMP, NLIN)	WATERSTM GASMIX POTASIUM Refrigerants NH3-H2O

[1)] Only REFDROP

Function	Description	Usage	Valid medium types
h = f(p,x,comp.)	enthalpy as a function of pressure, vapor fraction and composition	H = HPXW(P, X, W, XGAS, L)	WATERSTM POTASIUM Refrigerants NH3-H2O
h = f(p,t,comp.)	enthalpy as a function of pressure, temperature and composition	H = HPTW(P, T, W, XGAS, L,COMP, CONC, AVERG, IDX, NCOMP, NLIN, HUC)	WATERSTM GASMIX FUEL POTASIUM Refrigerants NH3-H2O
h = f(t,x,comp.)	enthalpy as a function of temperature, vapor fraction and composition	H = HTXW(T, X, W, XGAS, L)	WATERSTM POTASIUM Refrigerants NH3-H2O
p = f(t,x,comp.)	saturation pressure as a function of temperature, vapor fraction and composition	P = PTXW(T, X, W, XGAS, L)	WATERSTM POTASIUM Refrigerants NH3-H2O
ρ = f(p,t,comp.)	Specific mass a a function of pressure, temperature and composition	SM = SMPTW(P, T, W, XGAS, L)	Refrigerants [1)] NH3-H2O
s = f(p,h,comp.)	Entropy as a function as a function of pressure, enthalpy and composition	S = SPHW(P, H, W, XGAS, L, COMP, CONC, AVERG, IDX, NCOMP, NLIN, HUC)	WATERSTM GASMIX FUEL POTASIUM Refrigerants NH3-H2O
s = f(p,t,comp.)	Entropy as a function of pressure, temperature and composition	S = SPTW(P, T, W, XGAS, L, COMP, CONC, AVERG, IDX, NCOMP, NLIN, HUC)	WATERSTM GASMIX FUEL POTASIUM Refrigerants NH3-H2O
tc = f(p,t,comp.)	Thermal conductivity as a function of pressure, temperature, and composition	TC = TC_PT(P, T, W, XGAS, L)	WATERSTM Refrigerants[1)] NH3-H2O

[1)] Only REFDROP

Function	Description	Usage	Valid medium types
tc = f(t,x,comp.)	Thermal conductivity as a function of temperature, vapor fraction and composition	TC = TC_TX(T, X, W, XGAS, L)	WATERSTM POTASIUM Refrigerants [1)]
t = f(p,h,comp.)	Temperature as a function of pressure, enthalpy and composition	T = TPHW(P, H, W, XGAS, L, COMP, CONC, AVERG, IDX, NCOMP, NLIN, HUC)	WATERSTM GASMIX FUEL POTASIUM Refrigerants NH3-H2O
t = f(p,x,comp.)	Saturation temperature as a function of pressure, vapor fraction and composition	T = TPXW(P, X, W, XGAS, L)	WATERSTM POTASIUM Refrigerants NH3-H2O
v = f(p,h,comp.)	Specific volume as a function of pressure, enthalpy and composition	V = VPHW(P, H, W, XGAS, L, COMP, CONC, AVERG, IDX, NCOMP, NLIN)	WATERSTM GASMIX FUEL Refrigerants POTASIUM NH3-H2O
v = f(p,t,comp.)	Specific volume as a function of pressure, temperature, and composition	V = VPTW(P, T, W, XGAS, L, COMP, CONC, AVERG, IDX, NCOMP, NLIN)	WATERSTM GASMIX FUEL Refrigerants POTASIUM NH3-H2O
v = f(p,x,comp.)	Specific volume as a function of pressure, vapor fraction and composition	V = VPXW(P, X, W, XGAS, L)	WATERSTM POTASIUM Refrigerants NH3-H2O
v = f(t,x,comp.)	Specific volume as a function of temperature, vapor fraction and composition	V = VTXW(T, X, W, XGAS, L)	WATERSTM POTASIUM Refrigerants NH3-H2O
x = f(p,h,comp.)	Vapor fraction as a function of pressure, enthalpy and composition	X = XPHW(P, H, W, XGAS, L)	WATERSTM POTASIUM Refrigerants NH3-H2O

[1)] Only REFPROP

Appendix A

Apparatus Variables

Contents of the DATA(P) array for all apparatus types (elements 1 through 6)

Type no.	Name	1	2	3	4	5	6
1	boiler	PIN	POUT	DELP	TIN	TOUT	DELT
2	reheater	PIN	POUT	DELP	TIN	TOUT	DELT
3	turbine	PIN	PINCND		TIN	TOUT	DELT
4	condenser	PIN1	POUT1	DELP1	TIN1	TOUT1	DELT1
5	flash. heat.	PIN1	POUT1	DELP1	TIN1	TOUT1	DELT1
6	heatexchng.	PIN1	POUT1	DELP1	TIN1	TOUT1	DELT1
7	deaerator	PIN	POUT	DELP	TIN	TOUT	DELT
8	pump	PIN	POUT	DELP	TIN	TOUT	DELT
9	node			DELP			
10	sink/source	PIN	POUT	DELP	TIN	TOUT	DELT
11	node+e			DELP			
12	evaporator	PIN1	POUT1	DELP1	TIN1	TOUT1	DELT1
13	combustor	PIN	POUT	DELP	TIN	TOUT	
14	valve			DELP			
15	drum	PIN	POUT				
20	reformer	PIN1	POUT1	DELP1	TIN1	TOUT1	DELT1
21	fuel cell	PIN1	POUT1	DELP1	TIN1	TOUT1	DELT1
22	moist. sep.	PIN1	POUT1	DELP1	TIN1	TOUT1	DELT1
23	gasifier	PIN1	POUT1	DELP1	TIN1	TOUT1	DELT1
25	scrubber	PING	POUT	DELPG	TING		
26	separator	PIN	POUT5	DELP5	TIN	TOUT5	DELT5
27	ch. reactor	PIN	POUT	DELP	TIN	TOUT	
28	saturator	PING	POUTG	DELPG	TING	TOUTG	DELTG
29	compressor	PIN	POUT	DELP	TIN	TOUT	DELT

Contents of the DATA(P) array for all apparatus types (elements 7 through 12) (continued)

Type no.	Name	7	8	9	10	11	12
1	boiler	DELE	(DELM)		ETHAB		
2	reheater	DELE	(DELM)		ETHAB		
3	turbine	ETHAI	ETHAM	TUCODE	DIAIN	DIAOUT	POUTDS
4	condenser	DELE	RPSM	PIN2	POUT2	DELP2	TIN2
5	flash. heat.	DELE		PIN2	POUT2	DELP2	TIN2
6	heat exchng.	DELE	RPSM	PIN2	POUT2	DELP2	TIN2
7	deaerator	DELE					
8	pump	ETHAI	ETHAM		ETHAE		
9	node				RMASS(1)	RMASS(2)	RMASS(3)
10	sink/source	DELE	DELM	HIN	HOUT	DELH	XIN
11	node+e	DELE					
12	evaporator		RPSM	PIN2	POUT2	DELP2	TIN2
13	combustor	DELE	ESTPOU			DPREAC	
14	valve				RMASS(1)	RMASS(2)	RMASS(3)
15	drum	DELE	CRATIO				
20	reformer	DELE	RPSM	PIN2	POUT2	DELP2	TIN2
21	fuel cell	DELE		PINCA	POUTCA	DELPCA	TINCA
22	moist. sep.	DELE	RPSM	PIN2	POUT2	DELP2	TIN2
23	gasifier	DELE		PIN2	POUT2	DPREAC	TIN2
25	scrubber	DELE	MAXIT	PINW		DELPW	TINW
26	separator	DELE			POUT6	DELP6	
27	ch. reactor	DELE	MAXIT		RMASS(1)	RMASS(2)	RMASS(3)
28	saturator	DELE	DELMW	PINW	POUTW	DELPW	TINW
29	compressor	ETHAI	ETHAM	PRATI	ETHAE		

Contents of the DATA(P) array for all apparatus types (elements 13 through 18) (continued)

Type no.	Name	13	14	15	16	17	18
1	boiler			EXFUEL	ESTMAS	LHV	
2	reheater			EXFUEL	ESTMAS	LHV	
3	turbine	DESMAS	POUTRT	ETHAID	GDCODE		
4	condenser	TOUT2	DELT2	DELTH	DELTL/SATCOD		
5	flash. heat.	TOUT2	DELT2	DELTH	DELTL/SATCOD		
6	heatexchng.	TOUT2	DELT2	DELTH	DELTL		
7	deaerator						
8	pump			(ETHAIC)	(ETHATT)		
9	node	RMASS(4)	RMASS(5)	RMASS(6)			
10	sink/source	XOUT	PIPE	(IMEQ)	ESTMAS	LHV	SUBTYP
11	node+e						
12	evaporator	TOUT2	DELT2	DELTH	DELTL	PRISEC	
13	combustor		DTREAC	(IMEQ1)	(IMEQ2)	PREACT	TREACT
14	valve	RMASS(4)	PIPE	(IMEQ)	RVOL(1)	RVOL(2)	RVOL(3)
15	drum						
20	reformer	TOUT2	DELT2			PREACT	TREACT
21	fuelcell	TOUTPS	DELTCA	(IMEQ)	(IEEQ)	PREACT	TREACT
22	moist. sep.	TOUT2	DELT2	(IMEQ)	DELTL		ESTTGS
23	gasifier	TOUT2	DTREAC	(IMEQ)	SFRATI	PREACT	TREACT
25	scrubber					RELHUM	ESTTEM
26	separator	TOUT6	DELT6	(IMEQ)	TEMDIF		
27	ch. reactor	RMASS(4)	RMASS(5)		ESTMAS	PREACT	
28	saturator	TOUTW	PIPE	(IMEQ)	DELTL	RELHUM	ESTTEM
29	compressor		(COCODE)	(ETHAIC)	(ETHATT))	ETHAID	(VOLFLD)

Contents of the DATA(P) array for all apparatus types (elements 19 through 25) (continued)

Type no.	Name	19	20	21	22	23	24	25
1	boiler							
2	reheater							
3	turbine							(POWER)
4	condenser	DTSUBC						
5	flash. heat.							
6	heatexchng.							
7	deaerator							
8	pump							(POWER)
9	node							
10	sink/source	DTSUBC	DTSUPH	WFOT	ESTTIN	ESTTOU	DELV	DELVN
11	node+e							
12	evaporator							
13	combustor	ESTOFR	LAMBDA	PASH	DPASH	TASH	DTASH	
14	valve	RVOL(4)	RVOLN(1)	RVOLN(2)	RVOLN(3)	RVOLN(4)		
15	drum							
20	reformer	SFRATI	SFMOL					
21	fuelcell	UFL	UOX	IPUFL	IPUOX	DELEP	DCAC	POWER
22	moist. sep.	CGRATI	DELTH	ESTPGS				
23	gasifier	ESTOFR	OFRATI	PASH	DPASH	TASH	DTASH	AFRATI
25	scrubber	ESTMLF		ESTPGS		DSPLIT		
26	separator							
27	ch. reactor			ESTPGS		TWGS	TCH4R	TRUSER
28	saturator	ESTMLF	DELTH	WFOTEB			DELTW	MLFH2O
29	compressor	PRATID	PCTRPM					(POWER)

The elements 26 through 50 are used only for the fuel cell (type 21), and are currently not being used for any other apparatus type.

Contents of the DATA(P) array for apparatus type 21 (elements 26 through 50)

element			Element			element	
26	PFCELL		35	(RMK)		43	
27	TFCELL		36	(OFRAT)		44	(UC)
28	ICCUFL		37	(UFL)		45	ESTMFL
29	(URATIO)		38	(UOX)		46	ESTMOX
30	TH2OOS		39	PINAN		47	ESTUFL
31			40	POUTAN		48	ESTUOX
32			41	DELPAN		49	DELTAN
33	(RMA1)		42	TINAN		50	XSHIFT
34	(RMA2)						