ProdeProperties Test File

for Prode Properties ver. 1.2b

This file may be used to insure that the Mathcad prode.dll file is in the right directory and

Directions for using this worksheet

The default archive from Prode, def.ppp, will be used as the starting file.This worksheet will

Operations that set a variable will show a result of 1 if successful or 0 if not.Results that

Automatic calculation has been turned off so the new user may read these instructions befor

Procedure:
Calculate the entire worksheet, ctrl-F9
Select dep.ppp archive when first window appears, and click Open
When second popup window, the Prode archive, appears, it should show stream 2.Click OK.
Scroll through the worksheet to check for errors.See "Errors" below.
If the Prode window pops up, it is allowing you to view a recently created stream.Click OK

Errors:
Mathcad will show errors in red as usual.Typical errors might be caused by syntax in the a
If the result is zero, then the Prode function had an error or could not return a value.Frequen
Rarely, a ppp.dll error window may appear with "Error accessing component's data archive".Thi

File open commands

`mc_AFOpen( "C:\ProgramData\prode\def.ppp" )= 1` disabled

This command sets the path to the archive file and directory.The path shown is the default

`mc_AOpen( "dummy" )= 1` browse for an archive in the default directory

The next two functions do not obey the normal Prode convention regarding the result returne

`mc_setErrFlag( 0)= 0` set to 0at start of calc's to clear flag

`wopt := 0`

`mc_defErrMsg( wopt )= 0` wopt = 0 turns off the Window Dialog messages
wopt = 1 turns on the Window Dialog messages

Open Properties window to view edit streams

`stream := 2`

`mc_edS( stream )= 1` edit the given stream in the current active archive

`mc_edSS( "dummy" )= 1` open to the first stream (disabled for the test to r

Chemical file operations

`mc_getFCNr( "dummy" )= 58` number of components in data file, should be 163

`id := 7732185` CAS number of water (use internet search to find

compcode is an integer from 1 to number of components in the data file

`compcode := mc_CompCID( id )= 21` given id=CAS#, return compcode from database

Note: The above statement shows that the functions may be used to define a variable

`mc_CompF( compcode )= "H2O"` givena component code, returns component formula

`mc_CompN( compcode )= "WATER"` component name

`mc_CompID( compcode )= 7.7322 ·10`^{6} CAS number of component, compare to id above

Note: The units are not returned by the Prode commands.Operations that show which

`mc_CompMw( compcode )= 18.0153` molecular weight

$T_c := $ `mc_CompTc( compcode )` $K = 647.096$ critical temperature

$T_c = 64709.6008$ $K\ s\ T\ R$ multiply the function by the current Prode units for

`mc_CompPc( compcode )= 2.2064 ·10`^{7} critical pressure

`mc_CompVc( compcode )= 0.0031` critical volume

`mc_CompAc( compcode )= 0.344` acentric factor

$mc\_CompDm(compcode) = 6.1782 \cdot 10^{-30}$ dipole moment

$mc\_CompRg(compcode) = 6.15 \cdot 10^{-11}$ radius of gyration

$mc\_CompSol(compcode) = 1511.8849$ solubility parameter

$mc\_CompHf(compcode) = -13422.8364$ heat of formation

$mc\_CompGf(compcode) = -12698.6729$ Gibbs energy of formation

$mc\_CompSf(compcode) = 333.4738$ enthalpy of fusion

$mc\_CompNb(compcode) = 373.15$ normal boiling point

$mc\_CompMp(compcode) = 273.15$ melting point temperature

The following provide non zero values only if the phase of interest is present at the te

$tgl := 300$ temperature for gas/liquids (above free

$ts := 260$ temperature for solids (below freezing)

$mc\_CompVP(compcode, tgl) = 3548.3262$ saturation pressure at temp tgl

$mc\_CompHV(compcode, tgl) = 2436.3108$ heat of vaporization at tgl

$mc\_CompLV(compcode, tgl) = 0.0009$ liquid viscosity at tgl

$mc\_CompGV(compcode, tgl) = 9.9253 \cdot 10^{-6}$ gas viscosity at tgl

$mc\_CompLD(compcode, tgl) = 995.4764$ liquid density at tgl

$mc\_CompSD(compcode, ts) = 918.6313$ solid density at ts

$mc\_CompLC(compcode, tgl) = 0.6162$ liquid thermal conductivity at tgl

$mc\_CompGC(compcode, tgl) = 0.0188$ gas thermal conductivity at tgl

$mc\_CompSC(compcode, ts) = 0$ solid thermal conductivity at ts (appear

$mc\_CompST(compcode, tgl) = 0.0718$ surface tension at tgl

integrated changes between two temperatures, t0 and t1 for pure compon

$t0 := 280$ $t1 := 290$

$mc\_CompHG(compcode, t0, t1) = 18.6114$ ideal gas enthalpy change

$mc\_CompSG(compcode, t0, t1) = 0.0653$ ideal gas entropy change

$mc\_CompHL(compcode, t0, t1) = 42.0177$ ideal liquid enthalpy change

$mc\_CompSL(compcode, t0, t1) = 0.1474$ ideal liquid entropy change

$ts0 := 260$ $ts1 := 270$ lower the temperature range < freez

mc_CompHS(compcode , ts0 , ts1) = 20.5241 ideal solid enthalpy change

mc_CompSS(compcode , ts0 , ts1) = 0.0774 ideal solid entropy change

Units commands

See "Units of Measurement" section in Prode manual for a list of the units and their numeric

UM := 15 pressure is used for an example

n_press := mc_getUMN(UM) = 20 no. of units avail. for UM

mc_getUMC(UM) = 1 present units code for UM

mc_getSUMS(UM) = "Pa.a" present units string for UM

sel := 5 select unit 5

mc_getUMS(UM , sel) = "KPa.a" units string for (UM, sel)

list all of the units for pressure

i_ := 1 .. n_press

P_units =

mc_getP(stream) Pa = 14.6963 psi multiply by current Prode pressure unit, then

sel := 11 select a new pressure unit

mc_setUMC(UM , sel) = 1 change to the 11th unit for pressure

mc_getSUMS(UM) = "psi.a" show current unit name for UM

mc_getP(stream)psi = 14.6965 psi now pressure results must be multiplied by ps

mc_setUMC(UM , 1)= 1 reset to original unit for remainder of worksheet

Routines UMCR, UMCS, and UMAU are not fully documented in the Prode manual so

mc_UMRAU(UM)= 1 removes all added units for (property no.)

Error message flags

last error message, mayb

mc_ErrMsg(" ")= "Error accessing component's d

dum := "dummy"

errflag := mc_getErrFlag(dum) 0 = no errors, 1 = errors found

errflag = 1 This flag only works if the Window Dialog message

At the time this test file was created, the thermal

mc_setErrFlag(0)= 0 set to 0at start of next calc's to clear flag

mc_defErrMsg(0)= 0 = turns off the Window Dialog messages
1 = turns on the Window Dialog messages

This function was demonstrated at start of workshee

Atmospheric pressure

patm := mc_getPatm("mc")= 1.0133 ·10^5 the pressure shouldbe 1.013105

This section is new for this version

Base values for enthalpy and entropy

The default values for the base temperature, enthalpy, and entropy may be found in the cor

Code Procedure
1 = initial values specified by user (values of tref and val)
2 = initial values are enthalpy of formation (or entropy of formation) and temperature 25 C
If code = 2, the tref and val inputs are ignored.

code := 1 tref := 298 val := 0

mc_setHB(code , tref , val) = 1 enthalpy references

mc_setSB(code , tref , val) = 1 entropy references

Read/write stream properties

If a write operation exists, it will appear under the read operation, using the value from the

The write operations in this section are in blue highlight.

stream := 1

phase := 2 the phase position (not the phase type)

cpos := 2 cpos is the component's numerical position

mc_isSDef(stream) = 1 given a stream returns TRUE (integer = 1) if

name := mc_StrN(stream) = "Test Case 1" stream name

mc_putN(stream , name) = 1

mc_setOp(stream , 150 , patm) = 1 This is an edit operation to lower the temperature

t := mc_getT(stream) = 150 temperature

mc_putT(stream , t) = 1

p := mc_getP(stream) = $1.0133 \cdot 10^{5}$ pressure

mc_putP(stream , p) = 1

pnr := mc_getPNr("mc") = 5 returns the maximum number of phases that procedure

mc_StrPt(stream , phase) = 1 given a stream and phase # in range 1- getPNr() returns

i_ := 0 .. pnr − 1

given a stream and phase # in range 1- getPNr() returns

phases =

only one liquid phase is present
later, the flash routine code will be

mc_StrLf(stream)= 0.2598 given a stream returns the total liquid fra

mc_StrPf(stream , phase)= 0.1611 given a stream and phase phase # in ra

w := mc_getW(stream , phase , cpos)= 0.2717 mole fraction of component (cpos #)

mc_putW(stream , phase , cpos , w mole fraction w of component cpos in a phase

rate := mc_getWm(stream)= 1 stream flow rate, mass/time

mc_setWm(stream , rate)= 1

zi := mc_getZ(stream , cpos)= 0.15 mole fraction of component cpos in

mc_putZ(stream , cpos , zi)= 1

mc_getCNr(stream)= 3 number of components in stream

mc_StrZv(stream)= 0.9845 returns the relevant compressibility factor

mc_StrMw(stream)= 22.9437 molecular weight of total stream

mc_StrGMw(stream)= 17.5627 molecular weight of gas phase

mc_StrLMw(stream)= 38.2718 molecular weight of liquid phase

mc_StrV(stream)= 0.3913 specific volume as sum of specific volume

enthalpy

h := mc_StrH(stream)= − 434.4133 total stream enthalpy

mc_StrGH(stream)= − 163.9053 gas phase enthalpy

mc_StrSGH(stream)= − 289.2938 gas specific enthalpy

mc_StrLH(stream)= − 270.5079 liquid enthalpy

mc_StrSLH(stream)= − 624.1104 liquid specific enthalpy

mc_StrSH(stream) = 0 solid enthalpy

mc_StrSSH(stream) = 0 solid specific enthalpy

entropy

entropy := mc_StrS(stream) = – 2.1368 total stream entropy

mc_StrGS(stream) = – 0.7565 gas phase entropy

mc_StrSGS(stream) = – 1.3352 gas specific entropy

mc_StrLS(stream) = – 1.3803 liquid entropy

mc_StrSLS(stream) = – 3.1847 liquid specific entropy

mc_StrSS(stream) = 0 solid entropy

mc_StrSSS(stream) = 0 solid specific entropy

heat capacity, mass basis

mc_StrGICp(stream) = 1.8869 ideal gas heat capacity

mc_StrGCp(stream) = 1.9145 gas constant pressure heat capacity

mc_StrGCv(stream) = 1.416 gas constant volume heat capacity

mc_StrLCp(stream) = 1.7373 liquid constant pressure heat capacity

mc_StrLCv(stream) = 1.2147 liquid constant volume heat capacity

mc_StrSCp(stream) = 0 solid constant pressure heat capacity

speed of sound

mc_StrMSS(stream) = mixed phase speed of sound HEM model

mc_StrGSS(stream) = 266.8638 gas phase

mc_StrLSS(stream) = 1894.1038 liquid phase

Joule Thomson coefficient

mc_StrGJT(stream) = $1.5277 \cdot 10^{-5}$ gas phase

mc_StrLJT(stream) = – $3.9923 \cdot 10^{-7}$ liquid phase

compressibility, expansivity

mc_StrGIC(stream) = $1.0026 \cdot 10^{-5}$ gas isothermal compressibility, 1VPV

mc_StrLIC(stream) = $5.9767 \cdot 10^{-10}$ liquid isothermal compressibility

mc_StrGVE(stream) = – 0.0069 gas volumetric expansivity 1VTV

mc_StrLVE(stream) = 0.0015 gas volumetric expansivity

mc_StrLVE(stream)= 0.0015 gas volumetric expansivity

density

mc_StrGD(stream)= 1.4494 gas density

mc_StrLD(stream)= 1112.6071 liquid density

thermal conductivity

mc_StrGC(stream)= 0.0155 gas conductivity

mc_StrLC(stream)= 0.2849 liquid conductivity

viscosity

mc_StrGV(stream)= $6.2611 \cdot 10^{-6}$ gas viscosity

mc_StrLV(stream)= 0.0002 liquid viscosity

surface tension

mc_StrST(stream)= 0.0345 liquid/gas

flammability

mc_StrFML(stream)= 4.9934 gas phase lean limit

mc_StrFMH(stream)= 15.0823 gas phase rich limit

other stream properties

mc_StrHC(stream)= 43324.9154 gas phase heat of combustion

compcode := mc_getCC(stream , cpos)= 2 component number for component = cpos

mc_putCC(stream , cpos , compcode)= 1

mc_getMCNr("dummy")= 10 maximum number of components in a stream

interactions

This section was changed for this version.

int_pos := 1 the interaction number (i.e. the row in the

mc_getMBPNr("dummy")= 250 maximum number of binary pairs in a stre

ci := mc_getCi(stream , int_pos)= 1 component index i in interaction list

mc_putCi(stream , int_pos , ci)= 1

cj := mc_getCj(stream , int_pos)= 2 component index j in interaction list

mc_putCj(stream , int_pos , cj)= 1

model := mc_getMB(stream , int_pos)= 50 returns model number for interaction # int

mc_putMB(stream , int_pos , model) = 1 sets the model number for a given stream

id := 0 This id is the BIP column (starting with 0 for BIP-

Kji := mc_getBIP(stream , int_pos , id) = 0 value of the interaction coefficient

mc_putBIP(stream , int_pos , id , Kji) = 1 specifies a value for an interaction co

thermodynamic models for streams

stream = 1 This section was changed for this version.

Kcode := 50 code for SRK standard model package, see manua

mc_setKM(stream , Kcode) = 1

mp codes for functions below

1fugacity
2enthalpy
3entropy
4molar volume
5viscosity

Examples for fugacity and enthalpy models below:

state := 0 vapor state

mp := 1 fugacity model for stream, state

fmodel := mc_getMP(stream , mp , state) = 50 retrieve model number

mc_setMP(stream , mp , fmodel , state) = 1 set model number

mp := 2 enthalpy model for stream, stat

hmodel := mc_getMP(stream , mp , state) = 50 retrieve model number

mc_setMP(stream , mp , hmodel , state) = 1 set model number

Thermodynamic calculations

stream := 5 use stream 5 for examples bel

t = 150 $p = 1.0133 \cdot 10^{5}$

state := 1 state (0=vapor, 1=liquid, 2=solid)

phase equilibria

n := 1 pf := .3 see below

n := 1 pf := 1.0 see below

mc_PfPF(stream , p , pf , state , n) = 292.2056 n th equilibrium temp at p, pf

mc_PfTF(stream , t , pf , state , n) = 0 n th equil. press at t, pf, state

lf := .2 set liquid fraction

mc_LfPF(stream , p , lf) = 300.6739 first equil. temp at liquid fraction, lf

mc_LfTF(stream , t , lf) = 0 first equil. pressure at liquid fraction, lf

mc_StrCPnr(stream) = 1 number of critical points found

cpn := 1 selected critical point

mc_StrPc(stream , cpn) = $5.2424 \cdot 10^{6}$ critical pressure for critical point #, cpn

mc_StrCBp(stream) = 0 cricondenBar pressure

mc_StrCBt(stream) = 0 cricondenBar temperature

mc_StrCTp(stream) = $4.9822 \cdot 10^{6}$ cricondenTherm pressure

mc_StrCTt(stream) = 457.5439 cricondenTherm temperature

mc_StrAc(stream) = 0.2077 acentric factor (mole fraction average)

phase diagrams

stream := 5

lnr := mc_PELNr(stream) Given a stream calculates the phase diagram and returns

lnr = 2

line types

line := 2

ltype := mc_PELT(stream , line) Given a stream and line number, returns the line
bubble line
dew line
three phase line
fractional phase

ltype = 2

line properties

lprop := mc_PELP(stream , line) Given a stream and line, returns the line properties
vapor-liquid
vapor-liquid-liquid
vapor-solid
liquid-solid
fractional phase

lprop = 1

equilibrium lines

The prode.dll has assumed a maximum number of points of 50 for the equilibrium lines

The mc_PELine function (see the first line in the program below) produces a matrix re

Given stream and equilibrium line number, the temperature and pressure vectors and the tot

(T1 P1 npts1) := PELine(stream , line)

The output is shown below.

npts1 = ▮

T1 = ▮ P1 = ▮

phase fraction lines

stream := 5
state := 0

fraction := .5

Given stream, state, and fraction of that state, computes the temperature and pressure vecto

$(Tf\ \ Pf\ \ nf) := PFLine(\ stream\ ,\ state\ ,\ fraction\ )$

$nf =$ ▮

$Tf =$ ▮ $Pf =$ ▮

The Mathcad program, "PhaseEnv" below obtains all of the equilibrium curves which ca

$stream := 5$

$(Tj\ \ Pj\ \ lnr\ \ nc\ \ type\ \ prop) := PhaseEnv(\ stream\ )$

$lnr = 2$

$type =$ ▮ $nc =$ ▮

As shown in the nc vector, the lines may have different number of points.In order to p

The legend labels in the above plot were input manually supplied based on the values

The PELine, PFLine, and PhaseEnv programs may be copied into or referenced by oth

hydrates

hydmodel := 1 thyd := 260

str_hyd := 2 stream for hydrate function below

hydmodel =
1 = assume free water present, this option produces conservative but safe values
2 = calculate amount of water in liquid phase
3 = solve as multiphase equilibria, solve phase equilibria including solids as ice

Since water is not present in the stream chosen for testing, the 2 and 3 hydmodels w

phyd := mc_HPFORM(str_hyd , thyd , hydmodel) = 1.8208 $\cdot 10^{6}$ returns the pressure that hydrates form at

The HTFORM Prode function is not available in the Basic version, but the HPFORM fu

flashes

stream := 5

mc_setSOp(stream) = 1 — flash at standard conditions

et := 0 — estimated temperature set to 0 for automat

ep := 0 — estimated pressure set to 0 for automatic

mc_setOp(stream , 150 , patm) = 1 — set new operating conditions and flash

t := mc_getT(stream) = 150 — temperature

p := mc_getP(stream) = 1.0133 · 10^5 — pressure

h := mc_StrH(stream) = −697.4662 — enthalpy obtained above

entropy := mc_StrS(stream) = −2.8657 — entropy obtained above

sv := mc_StrV(stream) = 0.0014 — volume obtained above

find temperature

mc_VPF(stream , p , sv , et) = 150 — volume-pressure flash, et=temp guess

mc_HPF(stream , p , h , et) = 150 — enthalpy-pressure flash, et=temp guess

mc_SPF(stream , p , entropy , et) = 150 — entropy-pressure flash, et=temp guess

find pressure

mc_VTF(stream , t , sv , ep) = 0 — volume-temp flash, ep=press guess

mc_HTF(stream , t , h , ep) = 0 — enthalpy-temp flash, ep=press guess

mc_STF(stream , t , entropy , ep) = 0 — entropy-temp flash, ep=press guess

The flashes that determine pressure have some difficulty converging for multiphase (liqu

Additional flashes for mixing and dividing streams are found at this section

Extended methods for accessing stream properties

These functions allow simultaneous setting of temperature and pressure followed by an isothe

stream := 2

mc_EStrGMw(stream , t , p) = 16.0473 — gas molecular weight

mc_EStrLMw(stream , t , p)= 55.2982	liquid molecular weight
mc_EStrLf(stream , t , p)= 0.001	mole fraction of liquid
phase := 1	position of phase, not the state code positions are usually vapor=1, liquid=2
mc_EStrPf(stream , phase , t , p)= 0.999	molar phase fraction of phase
mc_EStrZv(stream , t , p)= 0.9843	gas (vapor) compressibility factor
mc_EStrH(stream , t , p)= − 317.8398	total enthalpy
mc_EStrV(stream , t , p)= 0.7525	total specific volume
mc_EStrGCp(stream , t , p)= 2.103	gas constant pressure heat capacity
mc_EStrGCv(stream , t , p)= 1.5593	gas constant volume heat capacity
mc_EStrLCp(stream , t , p)= 1.7093	liquid constant pressure heat capacity
mc_EStrLCv(stream , t , p)= 1.3993	liquid constant volume heat capacity
mc_EStrGIC(stream , t , p)= $1.0028 \cdot 10^{-5}$	gas isothermal compressibility
mc_EStrLIC(stream , t , p)= $5.2108 \cdot 10^{-10}$	liquid isothermal compressibility
mc_EStrMSS(stream , t , p)= 317.7238	mixture speed of sound
mc_EStrGSS(stream , t , p)= 318.267	gas speed of sound
mc_EStrLSS(stream , t , p)= 3324.6268	liquid speed of sound
mc_EStrGJT(stream , t , p)= $1.4662 \cdot 10^{-5}$	gas Joule Thomson coefficient
mc_EStrLJT(stream , t , p)= $-6.9575 \cdot 10^{-7}$	liquid Joule Thomson coefficient
mc_EStrGVE(stream , t , p)= − 0.0069	gas volumetric expansivity coefficient
mc_EStrLVE(stream , t , p)= − 0.0009	liquidvolumetric expansivity coefficient
mc_EStrHC(stream , t , p)= 50008.9256	heat of combustion
mc_EStrFML(stream , t , p)= 4.999	lean flammability limit of gas
mc_EStrFMH(stream , t , p)= 14.9988	rich flammability limit of gas
mc_EStrS(stream , t , p)= − 1.4662	total entropy
mc_EStrGD(stream , t , p)= 1.3246	gas density
mc_EStrLD(stream , t , p)= 729.1205	liquid density
mc_EStrGC(stream , t , p)= 0.0161	gas thermal conductivity
mc_EStrLC(stream , t , p)= 0.1752	liquid thermal conductivity
mc_EStrGV(stream , t , p)= $6.0281 \cdot 10^{-6}$	gas viscosity

mc_EStrLV(stream , t , p) = 0.0012 liquid viscosity

mc_EStrST(stream , t , p) = 0.0285 surface tension

Fugacity and derivatives

This section was changed for this version.

The operations below behave like subroutines rather than functions because they return more

The prode.dll has assumed a maximum number of components of 50 for all vector and matr

stream := 1

NC := mc_getCNr(stream)

t = 150 These variables were defined above.

$p = 1.0133 \cdot 10^{5}$

mc_setOp(stream , t , p) = 1

i_ := 0 .. NC − 1

phase := 2

w = 0.2717

state := 1 The liquid state is being used.

New in version 1.2b: The fugacity, H, S, and V functions and their derivatives previously inc

process := 1 Up to 5 processes may be defined in the base P

mc_DPinit(process , stream) = 1

fugacity vector

$fg := mc\_StrFv(process, state, t, p, w, NC)$ This routine returns the fugacity vector alone.

$$fg = \begin{pmatrix} 0 \\ -6.2774 \cdot 10^{66} \ \frac{kg}{m\ s^2} \\ -6.2774 \cdot 10^{66} \ \frac{kg}{m\ s^2} \end{pmatrix} \quad fugacity = \blacksquare$$

The Prode routines define the "fugacity" variable a

fugacity vector plus derivatives wrt T, P, w

given the stream, state, temp, press, composition vector, w,and the number of components,

$(fg\ \ dfgt\ \ dfgp\ \ dfgw) := StrFvd(process, state, t, p, w, NC)$

Add the default Prode units as needed.

$$fg\ Pa = \begin{pmatrix} 0 \\ -6.2774 \cdot 10^{66} \ \frac{kg}{m^2\ s^4} \\ -6.2774 \cdot 10^{66} \ \frac{kg}{m^2\ s^4} \end{pmatrix} \quad dfgt \ \frac{Pa}{K^2} = \blacksquare \ \frac{Pa}{K} \quad dfgp = \blacksquare$$

$dfgw\ Pa = \blacksquare$

Other stream state variables and their derivatives

This section was changed for this version.

Functions were provided above (eg. mc_StrH)to obtain the enthalpy (H), entropy(S), and mc

$KJ := 1000\ J$ $Kmol := 1000\ mol$ define new units for Mathcad

$(H\ dHt\ dHp\ dHw) := StrXvd("H", process, state, t, p, w, NC)$

$H \cdot \frac{KJ}{Kmol} = 1\ H\ \frac{KJ}{Kmol}$ the default Prode units have been applied to the

$dHt \cdot \frac{\frac{KJ}{Kmol}}{K} = \blacksquare\ \frac{KJ}{Kmol\ K}$

$dHp \cdot \frac{\frac{KJ}{Kmol}}{Pa} = \blacksquare\ \frac{KJ}{Kmol\ Pa}$

$dHw \cdot \frac{KJ}{Kmol} = \blacksquare\ \frac{KJ}{Kmol}$

$(S\ dSt\ dSp\ dSw) := StrXvd("S", process, state, t, p, w, NC)$

$S \cdot \frac{KJ}{Kmol\ K} = 1\ S\ \frac{KJ}{Kmol\ K}$

$dSt \cdot \frac{KJ}{Kmol\ K^2} = \blacksquare\ \frac{KJ}{Kmol\ K^2}$

$dSp \cdot \frac{KJ}{Kmol\ K\ Pa} = \blacksquare\ \frac{KJ}{Kmol\ K\ Pa}$

$dSw \cdot \frac{KJ}{Kmol\ K} = \blacksquare\ \frac{KJ}{Kmol\ K}$

$(V\ dVt\ dVp\ dVw) := StrXvd("V", process, state, t, p, w, NC)$

$V\ \frac{m^3}{mol} = 1\ \frac{m^3\ V}{mol}$

$dVt\ \frac{m^3}{\blacksquare} = \blacksquare$

$mol \cdot K^{-1}$

$$dVp \quad \frac{m^3}{mol \cdot Pa} = \blacksquare \frac{m^3}{mol \cdot Pa}$$

$$dVw \quad \frac{m^3}{mol} = \blacksquare$$

Operations to set/retrieve the options needed for equation of

See the Prode manual (see paragraph "Codes used in Prode library") and also open the Pr

All even code values mean that only single liquid phases are allowed in the flash routines.

stream := 1

option := mc_getOM(stream) = 545 current option set
This should = 552 for stream 1 in the def.ppp arc

mc_setOM(stream , option + 1) = 1 This changes the flashes of "stream" to multiple li

mc_setOp(stream , t , p) = 1 redo the flash

given a stream and phase # in range 1- getPNr() returns

i_ := 0 .. pnr - 1

given a stream and phase # in range 1- getPNr() returns

phases = ■

Previously, only one liquid was present

mc_setOM(stream , option) = 1 reset the code to single liquids

Initializing a stream

This section was changed for this version.

The example will create a stream with water and methanol. The component numbers in the

methanol_id := 67561 CAS number of methanol

methanol code := mc CompCID(methanol id) = 24

water_id := 7732185 CAS number of water

water_code := mc_CompCID(water_id) = 21

stream := 11

mc_initS(stream) = 1 initialize a new stream

model := 50 SRK standard see Prode manual for model cod

mc_setKM(stream , model) = 1 set property model package

mc_putZ(stream , 1 , .5) = 1 set total stream mole fractions

mc_putZ(stream , 2 , .5) = 1

mc_putCC(stream , 1 , methanol_code) = 1 define components

mc_putCC(stream , 2 , water_code) = 1

mc_setS(stream) = 1 validate the stream

mc_loadSB(stream) = 1 load BIP coefficients

mc_setWm(stream , 1.3) = 1 set mass flow rate

temp := 300 pres := patm

mc_setOp(stream , temp , pres) = 1 set temp and pres and flash

mc_edS(stream) = 1 view the stream then press OK

Other stream operations

stream2 := 1

stream1 := 10

mc_StrCopy(stream1 , stream2) = 1 copy stream2 to stream1 (note the order!)

et := mc_getT(stream2) = 150

mc_getT(stream1) = 150

mc_MixF(stream1 , stream2 , et) = 1 flash at lower stream press, et=temp guess for mi

mc_getT(stream1) = 123.5373 mixed stream temperature

stream2 := 12 the new stream to be created by Divi

wdiv := .7

mc_Divi(stream1 , stream2 , wdiv) = 1 Given one stream (stream1) and a flowrate fraction (0-1) into two streams (stream1, stream2) of the same composi specified by wdiv (stream2 = wdiv, stream1 = 1- wdiv)

Only one new stream is created, NOT two.The sta

phase separation

stream1 := 5

stream2 := 13 the new stream to be created by PSep

phase := 1 phase number to separate, NOT the phase type

mc_PSep(stream1 , stream2 , phase) = 1 Given a stream (stream1) performs an isothermal flash t

gasstream := 14

mc_GSep(stream1 , gasstream) = 1 Given a stream (stream1) performs an isothermal flash t

liqstream := 15

mc_LSep(stream1 , liqstream) = 1 Given a stream (stream1) performs an isothermal flash t

Polytropic compressor/expander

rate compressor efficiency

pin := 10^6 pressure in Pa

pout := $2 \cdot 10^6$

tin := 300 temperature in K

tout := 370

model := 2 for a rating, model may be the following:
2 = Huntington method
4 = Paron method

stream := 2

mc_setOp(stream , tin , pin) = 1 set the inlet stream conditions

mc_PSPF(stream , pout , model , tout) = 0.743 efficiency rating and "stream" in archive now contai

design model

eff := .75 polytropic efficiency given

model := 1 for a design, model may be the following:
1 = Huntington
3 = Paron

mc_setOp(stream , tin , pin)= 1 reset the inlet stream conditions

mc_PSPF(stream , pout , model , entle= 39.2757 temperature and "stream" now contains the

Isentropic expansion, nozzles

stream := 5

tin := 340 $pin := 2 \cdot 10^{6}$

mc_setWm(stream , 1.23)= 1

mc_setOp(stream , tin , pin)= 1 set stream conditions

model := 2 model options::
1 = homogeneous, equilibrium
2 = homogeneous, non equilibrium
3 = homogeneous, non equilibrium ?
4 = nonhomogeneous, non equilibrium

pout := patm

parameter := .75 Prode manual does not explain

calculated orifice area, m2

mc_ISPF(stream , pout , model , parameter)= $4.1708 \cdot 10^{-5}$
mc_getErrFlag("")= 1

Pipe flow

The PIPE function is only available for users with an extended Prode license.

model := 1

stream := 1

diam := $\frac{1\ in}{m} = 0.0254$

rough := .00045

length := $\frac{100\ km}{m} = 1 \cdot 10^{5}$

dHeight := 0

dHeat := 0

mc_PIPE(stream , model , diam , rough , length , dHeight , dHeat)= 0

The result above will be 0 if the user has a Basic Prode license or 1 for an Extended

Parameters :
stream (int) inlet stream
model (int) model for fluid flow and phase equilibria (see below)
diam (double) pipe internal diameter
rough (double) parameter defining relative pipe roughness
length (double) length of this segment
dHeight (double) height difference (inlet, outlet)
dHeat (double) heat added, removed
Codes for models
1Beggs & Brill / Hazen-Williams / AGA
additional models on request to Prode

File save

`mc_AFSave( "C:\ProgramData\prode\test.ppp" )=1` save modifications to a new