APPENDIX C
DESCRIPTION OF THE TRAC-M MODULE AND INCLUDE FILE (COMMON BLOCK) VARIABLES

The number of common blocks in TRAC-M has been reduced significantly relative to the number previously utilized in TRAC-P by placing the relevant data within modules. The remaining common blocks are now incorporated in the various TRAC routines through the use of Fortran 90 include statements. Each included filename consists of the common block name followed by the extension .h (e.g., `bandw.h`). The F90 source code for the modules and include files may be accessed directly using the TRAC PathFinder.

Note: XTV/XMGR5 Graphics System. Module `Xtv`, which implements the XTV/XMGR5 graphics system, is to be replaced in a future version of TRAC-M/F90 by modules `CXtvXFaces`, `XtvComps`, `XtvData`, `XtvDump`, and `XtvSetup`. The new implementation of the XTV/XMGR5 logic will include many arrays and derived types, all of which will be defined in module `XtvData`.

C.1. Module `Bad`

INTEGER(sik) nbd
PARAMETER (nbd = 74)
INTEGER(sik), PARAMETER :: bdFull = 0
INTEGER(sik), PARAMETER :: bdInit = 1
INTEGER(sik), PARAMETER :: bdPrep = 2

INTEGER VARIABLES:

nbd	Used to dimension the bd array.
bdFull	Used in bd-array logic at VESSEL source (3D-1D) junctions.
bdInit	Used in bd-array logic at VESSEL source junctions.
bdPrep	Used in bd-array logic at VESSEL source junctions.

C.2. Module `BadInput`

INTEGER(sik) jflag

INTEGER VARIABLES:

jflag	Flag that is set to 1 when an input-data error is encountered and TRAC-M is to abort the calculation after all input data have been processed.

C.3. Module `Bits`

INTEGER(sik) satLineCrossVap

```
PARAMETER (satLineCrossVap=1)
!
INTEGER(sik) satLineCrossLiq
PARAMETER (satLineCrossLiq=2)
!
INTEGER(sik) oneVoidFrReit
PARAMETER (oneVoidFrReit=3)
!
INTEGER(sik) meanEqnSet
PARAMETER (meanEqnSet=4)
!
INTEGER(sik) condensing
PARAMETER (condensing=5)
!
INTEGER(sik) evapOrFlashing
PARAMETER (evapOrFlashing=6)
!
INTEGER(sik) freezeXvset
PARAMETER (freezeXvset=7)
!
INTEGER(sik) tinyBubbles
PARAMETER (tinyBubbles=8)
!
INTEGER(sik) triedVoidFrReset
PARAMETER (triedVoidFrReset=9)
!
INTEGER(sik) netAirFlow
PARAMETER (netAirFlow=10)
!
INTEGER(sik) packAtLeftFace
INTEGER(sik) pack3D
PARAMETER (packAtLeftFace=11)
PARAMETER (pack3D=11)
!
INTEGER(sik) packAtRightFace
PARAMETER (packAtRightFace=12)
!
```

```
INTEGER(sik) stretch
INTEGER(sik) stretch3D
PARAMETER (stretch=13)
PARAMETER (stretch3D=13)
!
INTEGER(sik) newVoidFrUp
PARAMETER (newVoidFrUp=14)
!
INTEGER(sik) oldVoidFrUp
PARAMETER (oldVoidFrUp=15)
!
INTEGER(sik) netMassOut
PARAMETER (netMassOut=16)
!
INTEGER(sik) specEqnSteamP
PARAMETER (specEqnSteamP=17)
!
INTEGER(sik) negVapVel
PARAMETER (negVapVel=20)
!
INTEGER(sik) negLiqVel
PARAMETER (negLiqVel=21)
!
INTEGER(sik) significantVapFlow
INTEGER(sik) significantVapFlowxr
PARAMETER (significantVapFlow=22)
PARAMETER (significantVapFlowxr=22)
!
INTEGER(sik) significantLiqFlow
INTEGER(sik) significantLiqFlowxr
PARAMETER (significantLiqFlow=23)
PARAMETER (significantLiqFlowxr=23)
!
INTEGER(sik) significantVapFlowz
PARAMETER (significantVapFlowz=24)
!
INTEGER(sik) significantVapFlowyt
```

```
PARAMETER (significantVapFlowyt=25)
!
INTEGER(sik) changeVapVel
INTEGER(sik) changeVapVelyt
PARAMETER (changeVapVel=26)
PARAMETER (changeVapVelyt=26)
!
INTEGER(sik) changeVapVelz
PARAMETER (changeVapVelz=27)
!
INTEGER(sik) significantLiqFlowyt
PARAMETER (significantLiqFlowyt=28)
!
INTEGER(sik) changeVapVelxr
PARAMETER (changeVapVelxr=29)
!
INTEGER(sik) chokedFlowOn
PARAMETER (chokedFlowOn=30)
!
INTEGER(sik) significantLiqFlowz
PARAMETER (significantLiqFlowz=31)
!
INTEGER(sik) userChokeControl
PARAMETER (userChokeControl=32)
!
```

INTEGER VARIABLES:

satLineCrossVap	Variable used to indicate that bit 1 of the `bit` and `bitn` arrays is to be accessed.
satLineCrossLiq	Variable used to indicate that bit 2 of the `bit` and `bitn` arrays is to be accessed.
oneVoidFrReit	Variable used to indicate that bit 3 of the `bit` and `bitn` arrays is to be accessed.
meanEqnSet	Variable used to indicate that bit 4 of the `bit` and `bitn` arrays is to be accessed.
condensing	Variable used to indicate that bit 5 of the `bit` and `bitn` arrays is to be accessed.
evapOrFlashing	Variable used to indicate that bit 6 of the `bit` and `bitn` arrays is to be accessed.

freezeXvset	Variable used to indicate that bit 7 of the `bit` and `bitn` arrays is to be accessed.
tinyBubbles	Variable used to indicate that bit 8 of the `bit` and `bitn` arrays is to be accessed.
triedVoidFrReset	Variable used to indicate that bit 9 of the `bit` and `bitn` arrays is to be accessed.
netAirFlow	Variable used to indicate that bit 10 of the `bit` and `bitn` arrays is to be accessed.
packAtLeftFace	Variable used to indicate that bit 11 of the `bit` and `bitn` arrays is to be accessed.
pack3D	Variable used to indicate that bit 11 of the 3D `bit` and `bitn` arrays is to be accessed.
packAtRightFace	Variable used to indicate that bit 12 of the `bit` and `bitn` arrays is to be accessed.
stretch	Variable used to indicate that bit 13 of the `bit` and `bitn` arrays is to be accessed.
stretch3D	Variable used to indicate that bit 13 of the 3D `bit` and `bitn` arrays is to be accessed.
newVoidFrUp	Variable used to indicate that bit 14 of the `bit` and bitn arrays is to be accessed.
oldVoidFrUp	Variable used to indicate that bit 15 of the `bit` and `bitn` arrays is to be accessed.
netMassOut	Variable used to indicate that bit 16 of the `bit` and `bitn` arrays is to be accessed.
specEqnSteamP	Variable used to indicate that bit 17 of the `bit` and `bitn` arrays is to be accessed.
negVapVel	Variable used to indicate that bit 20 of the `bit` and `bitn` arrays is to be accessed.
negLiqVel	Variable used to indicate that bit 21 of the `bit` and `bitn` arrays is to be accessed.
significantVapFlow	Variable used to indicate that bit 22 of the `bit` and `bitn` arrays is to be accessed.
significantVapFlowxr	Variable used to indicate that bit 22 of the `bit` and `bitn` arrays is to be accessed.
significantLiqFlow	Variable used to indicate that bit 23 of the `bit` and `bitn` arrays is to be accessed.
significantLiqFlowxr	Variable used to indicate that bit 23 of the `bit` and `bitn` arrays is to be accessed.
significantVapFlowz	Variable used to indicate that bit 24 of the `bit` and `bitn` arrays is to be accessed.
significantVapFlowyt	Variable used to indicate that bit 25 of the `bit` and `bitn` arrays is to be accessed.

changeVapVel	Variable used to indicate that bit 26 of the `bit` and `bitn` arrays is to be accessed.
changeVapVelyt	Variable used to indicate that bit 26 of the `bit` and `bitn` arrays is to be accessed.
changeVapVelz	Variable used to indicate that bit 27 of the `bit` and `bitn` arrays is to be accessed.
significantLiqFlowyt	Variable used to indicate that bit 28 of the `bit` and `bitn` arrays is to be accessed.
changeVapVelxr	Variable used to indicate that bit 29 of the `bit` and `bitn` arrays is to be accessed.
chokedFlowOn	Variable used to indicate that bit 30 of the `bit` and `bitn` arrays is to be accessed.
significantLiqFlowz	Variable used to indicate that bit 31 of the `bit` and `bitn` arrays is to be accessed.
userChokeControl	Variable used to indicate that bit 32 of the `bit` and `bitn` arrays is to be accessed.

C.4. Module `Boundary`

```
REAL(sdk),DIMENSION(:),POINTER :: vsi
REAL(sdk),DIMENSION(:,:),POINTER :: bd
INTEGER(sik) :: bdOffset
```

REAL VARIABLES:

vsi	Velocity sign indicator array for bd array.
bd	bd array data for the components.

INTEGER VARIABLES:

bdOffset

C.5. Module `BreakArray`

```
TYPE breakArrayT
    REAL(sdk),  POINTER, DIMENSION(:)  :: alptb
    REAL(sdk),  POINTER, DIMENSION(:)  :: contb
    REAL(sdk),  POINTER, DIMENSION(:)  :: patb
    REAL(sdk),  POINTER, DIMENSION(:)  :: ptb
    REAL(sdk),  POINTER, DIMENSION(:)  :: rftb
    REAL(sdk),  POINTER, DIMENSION(:)  :: tltb
    REAL(sdk),  POINTER, DIMENSION(:)  :: tvtb
!
END TYPE breakArrayT
```

!

TYPE (breakArrayT),DIMENSION(maxComps) :: breakAr

DERIVED TYPE VARIABLES:

breakAr	Variables specific to the BREAK component.

REAL VARIABLES:

!Elements of DERIVED TYPE breakArrayT

alptb	Pointer to the gas-volume-fraction table.
contb	Pointer to the ratio of solute-mass to liquid-mass table.
ptab	Pointer to the noncondensable-gas, partial-pressure table.
ptb	Pointer to the total pressure table.
rftb	Pointer to the rate-factor table.
tltb	Pointer to the liquid temperature table.
tvtb	Pointer to the gas temperature table.

C.6. Module `BreakVlt`

TYPE breakTabT

REAL(sdk) alpoff
REAL(sdk) bsa
REAL(sdk) bsmass
REAL(sdk) bxa
REAL(sdk) bxmass
REAL(sdk) conoff
REAL(sdk) deltl
REAL(sdk) deltv
REAL(sdk) paoff
REAL(sdk) poff
REAL(sdk) poffs
REAL(sdk) rbmx
REAL(sdk) tin
REAL(sdk) tloff
REAL(sdk) tvoff
REAL(sdk) type1
INTEGER(sik) ibasv
INTEGER(sik) ibcnsv
INTEGER(sik) ibf
INTEGER(sik) ibp

```
    INTEGER(sik) ibpasv
    INTEGER(sik) ibpsv
    INTEGER(sik) ibsv
    INTEGER(sik) ibtlsv
    INTEGER(sik) ibtr
    INTEGER(sik) ibtvsv
    INTEGER(sik) ibty
    INTEGER(sik) inexti
    INTEGER(sik) ioff
    INTEGER(sik) ionoff
    INTEGER(sik) isat
    INTEGER(sik) js1
    INTEGER(sik) jun1
    INTEGER(sik) nbrf
    INTEGER(sik) nbsv
    INTEGER(sik) nbtb
    INTEGER(sik) js1get
    INTEGER(sik) js1put
  END TYPE breakTabT
!
TYPE(breakTabT),DIMENSION(maxComps) :: breakTab
!
INTEGER(sik) breakDumpSize
PARAMETER (breakDumpSize=38)
```

DERIVED TYPE VARIABLES:

breakTAB — The VLT for the BREAK component.

REAL VARIABLES:

!Elements of DERIVED TYPE breakTabT

alpoff	Gas-volume fraction when the trip is off after it was on.
bsa	Time-integrated, noncondensable-gas mass from the BREAK.
bsmass	Time-integrated mass flow from the BREAK.
bxa	Noncondensable-gas mass flow from the BREAK.
bxmass	Current mass flow from the BREAK.
conoff	Ratio of solute mass to liquid mass when the trip is off after it was on.
deltl	Liquid temperature offset from the saturation temperature.
deltv	Gas temperature offset from the saturation temperature.

paoff — Noncondensable-gas partial pressure when the trip is off after it was on.

poff — Total pressure when the trip is off after it was on.

poffs — Saved value of total pressure when the trip is off after it was on that has not been adjusted by a CSS type-5 controller.

rbmx — Maximum rate of change of total pressure at the BREAK.

tin — Fluid temperature at the BREAK.

tloff — Liquid temperature when the trip is off after it was on.

tvoff — Gas temperature when the trip is off after it was on.

type1 — Variable no longer used.

INTEGER VARIABLES:

!Elements of DERIVED TYPE breakTabT

ibasv — Signal-variable or control-block ID number defining the gas-volume fraction in the BREAK cell for the `ibty` = 6 option.

ibcnsv — Signal-variable or control-block ID number defining the ratio of the solute mass to the liquid mass in the BREAK cell for the `ibty` = 6 option.

ibf — Last interpolated interval in the rate-factor table.

ibp — Last interpolated interval in the BREAK composition parameter tables.

ibpasv — Signal-variable or control-block ID number defining the noncondensable-gas partial pressure in the BREAK cell for the `ibty` = 6 option.

ibpsv — Signal-variable or control-block ID number defining the total pressure in the BREAK cell for the `ibty` = 6 option.

ibsv — Signal-variable or control-block ID number defining the BREAK-table, abscissa-coordinate variable.

ibtlsv — Signal-variable or control-block ID number defining the liquid temperature in the BREAK cell for the `ibty` = 6 option.

ibtr — Trip ID number that controls evaluation of the BREAK tables.

ibtvsv — Signal-variable or control-block ID number defining the gas temperature in the BREAK cell for the `ibty` = 6 option

ibty — BREAK-table input option.

inexti — Variable no longer used.

ioff — Fluid-state option when the trip is off after it was on.

ionoff — Number of timesteps the trip is on.

isat — BREAK-table use option.

js1 — Junction sequence number.

jun1 — Junction number for connection to the BREAK.

nbrf — Number of data pairs in the rate-factor table.

nbsv	Signal-variable or control-block ID number defining the rate-factor table's abscissa-coordinate variable.
nbtb	Number of data pairs in the BREAK table.
js1get	Index for getting information from bd array.
js1put	Index for putting information into bd array.

!

breakDumpSize Number of variables in the derived-type breakTabT.

C.7. Module Ccfl

```
PARAMETER (mccfl=10)
  REAL(sdk) ccflm(mccfl),ccflc(mccfl),cbeta(mccfl),ctrans(mccfl)
  REAL(sdk) diah(mccfl)
  INTEGER(sik) nccfl,nholes(mccfl)
```

REAL VARIABLES:

ccflm	Slope of the CCFL correlation.
ccflc	Constant of the CCFL correlation.
cbeta	Bankoff interpolation constant for interpolating between Wallis and Kutalatze characteristic-length dimensions.
ctrans	Bond number above which the CCFL constant is independent of the Bond number.
diah	Diameter of one hole in the perforated plate.

INTEGER VARIABLES:

nccfl	Number of CCFL parameter sets.
nholes	Number of holes in the perforated plate.

C.8. Module CompTyp

```
REAL(sdk)     breakh,fillh,pipeh,plenh,prizrh,pumph,teeh,turbh,  &
& valveh,vsslh,rodh,slabh,sepdh
!
PARAMETER (pipeh =   1.0d0)
PARAMETER (vsslh =   2.0d0)
PARAMETER (rodh =    3.0d0)
PARAMETER (slabh =   4.0d0)
PARAMETER (pumph =   5.0d0)
PARAMETER (teeh =    6.0d0)
PARAMETER (valveh =  7.0d0)
PARAMETER (prizrh =  8.0d0)
```

```
PARAMETER (turbh =  9.0d0)
PARAMETER (sepdh =  10.0d0)
PARAMETER (breakh =  11.0d0)
PARAMETER (fillh =  12.0d0)
PARAMETER (plenh =  13.0d0)
```

REAL VARIABLES:

breakh	Identifies the PIPE-component-type number.
fillh	Identifies the FILL-component-type number.
plenh	Identifies the PLENUM-component-type number.
prizrh	Identifies the PRESSURIZER-component-type number.
pumph	Identifies the PUMP-component-type number.
teeh	Identifies the TEE-component-type number.
turbh	Identifies the TURBINE-component-type number.
valvh	Identifies the VALVE-component-type number.
vsslh	Identifies the VESSEL-component-type number.
rodh	Identifies the ROD-component-type number.
slabh	Identifies the SLAB-component-type number.
sepdh	Identifies the SEPARATOR-component-type number.

C.9. Module `ControlDat`

```
!   Global Data
TYPE csGlT
     INTEGER(sik) :: ntsv
     INTEGER(sik) :: ntcb
     INTEGER(sik) :: ntcf
     INTEGER(sik) :: ntrp
     INTEGER(sik) :: ntcp
     INTEGER(sik) :: ntse
     INTEGER(sik) :: ntct
     INTEGER(sik) :: ntsf
     INTEGER(sik) :: ntdp
     INTEGER(sik) :: ntsd
     REAL(sdk)    :: etime
END TYPE csGlT
!
    TYPE (csGlT) :: csGl
```

```
    TYPE (csGlT) :: csrGl
!   Control Parameter Evaluation Pass Data
TYPE csCPEDT
      INTEGER(sik) :: isv1
      INTEGER(sik) :: isv2
      INTEGER(sik) :: icb1
      INTEGER(sik) :: icb2
      INTEGER(sik) :: itp1
      INTEGER(sik) :: itp2
END TYPE csCpedT
!   Dynamically dimensioned to csGl%ntcp
TYPE (csCPEDT),ALLOCATABLE,DIMENSION(:) :: csCPED
!   Signal Variable Data
TYPE csSigT
      INTEGER(sik) :: idsv
      INTEGER(sik) :: isvn
      INTEGER(sik) :: ilcn
      INTEGER(sik) :: icn1
      INTEGER(sik) :: icn2
      REAL(sdk)    :: prevVal
      REAL(sdk)    :: presVal
END TYPE csSigT
!   Dynamically dimensioned to csGl%ntsv
TYPE (csSigT),ALLOCATABLE,DIMENSION(:) :: csSig
TYPE (csSigT),ALLOCATABLE,DIMENSION(:) :: csrSig
!   Control Block Data
TYPE csCBT
      INTEGER(sik)          :: idcb
      INTEGER(sik)          :: icbn
      INTEGER(sik),DIMENSION(3) :: icb
      REAL(sdk)             :: cbgain
      REAL(sdk)             :: cbxmin
      REAL(sdk)             :: cbxmax
      REAL(sdk)             :: cbcon1
      REAL(sdk)             :: cbcon2
      REAL(sdk)             :: flagl
      REAL(sdk)             :: flags
```

```
      REAL(sdk),DIMENSION(3)   :: cbin
      REAL(sdk)                :: prevVal
      REAL(sdk)                :: presVal
!
      REAL(sdk)                :: cbwt
      INTEGER(sik)             :: tableIndex
END TYPE csCBT
!   Dynamically dimensioned to csGl%ntcb
TYPE (csCBT),ALLOCATABLE,DIMENSION(:) :: csCB
TYPE (csCBT),ALLOCATABLE,DIMENSION(:) :: csrCB
!   Control Block Tabular Data
!   Dynamically dimensioned to csGl%ntcf
    REAL(sdk),ALLOCATABLE,DIMENSION(:) :: csCBTD
    REAL(sdk),ALLOCATABLE,DIMENSION(:) :: csrCBTD
!   Control Block Units Labels
TYPE csULCBT
      REAL(sdk),DIMENSION(6) :: data
      INTEGER(sik), DIMENSION(6) :: index
END TYPE csULCBT
!   Dynamically dimensioned to csGl%ntcb
TYPE (csULCBT),ALLOCATABLE,DIMENSION(:) :: csULCB
TYPE (csULCBT),ALLOCATABLE,DIMENSION(:) :: csrULCB
!   Trip Units Labels
TYPE csULTRT
      REAL(sdk),DIMENSION(1) :: data
      INTEGER(sik), DIMENSION(1) :: index
END TYPE csULTRT
!   Dynamically dimensioned to csGl%ntrp
TYPE (csULTRT),ALLOCATABLE,DIMENSION(:) :: csULTR
TYPE (csULTRT),ALLOCATABLE,DIMENSION(:) :: csrULTR
!   Signal Variable Units Labels
TYPE csULSET
      REAL(sdk),DIMENSION(5) ::data
      INTEGER(sik), DIMENSION(5) :: index
END TYPE csULSET
!   Dynamically dimensioned to csGl%ntse
TYPE (csULSET),ALLOCATABLE,DIMENSION(:) :: csULSE
```

```
TYPE (csULSET),ALLOCATABLE,DIMENSION(:) :: csrULSE
!   Trip Data
TYPE csTripT
      INTEGER(sik)          :: idtp
      INTEGER(sik)          :: isrt
      INTEGER(sik)          :: iset
      INTEGER(sik)          :: itst
      INTEGER(sik)          :: idsg
      REAL(sdk),DIMENSION(4)   :: setp
      REAL(sdk),DIMENSION(4)   :: dtsp
      INTEGER(sik),DIMENSION(4) :: ifsp
      REAL(sdk)             :: delsv
      REAL(sdk)             :: prevVal
      INTEGER(sik)          :: prevSub
      INTEGER(sik),DIMENSION(5) :: subRange
      REAL(sdk),DIMENSION(5)   :: setTime
      REAL(sdk),DIMENSION(10,5) :: timeData
END TYPE csTripT
!   Dynamically dimensioned to csGl%ntrp
TYPE (csTripT),ALLOCATABLE,DIMENSION(:) :: csTrip
TYPE (csTripT),ALLOCATABLE,DIMENSION(:) :: csrTrip
!   Trip Signal Expression Signal Data
TYPE csTSET
      INTEGER(sik)          :: idse
      INTEGER(sik)          :: inse
      INTEGER(sik)          :: incn
      INTEGER(sik),DIMENSION(3,10) :: ids
      REAL(sdk),DIMENSION(5)    :: constants
END TYPE csTSET
!   Dynamically dimensioned to csGl%ntse
TYPE (csTSET),ALLOCATABLE,DIMENSION(:) :: csTSE
TYPE (csTSET),ALLOCATABLE,DIMENSION(:) :: csrTSE
!   Trip Controlled Trip Signal Data
TYPE csTCTT
      INTEGER(sik)          :: idtn
      INTEGER(sik)          :: intn
      INTEGER(sik),DIMENSION(10)  :: tripIDs
```

```
END TYPE csTCTT
!   Dynamically dimensioned to csGl%ntct
TYPE (csTCTT),ALLOCATABLE,DIMENSION(:) :: csTCT
TYPE (csTCTT),ALLOCATABLE,DIMENSION(:) :: csrTCT
!   Trip Set Point Factor Table Data
TYPE csTSFT
    INTEGER(sik)              :: idft
    INTEGER(sik)              :: idsg
    INTEGER(sik)              :: inft
    REAL(sdk),DIMENSION(2,10)   :: setp
END TYPE csTSFT
!   Dynamically dimensioned to csGl%ntsf
TYPE (csTSFT),ALLOCATABLE,DIMENSION(:) :: csTSF
TYPE (csTSFT),ALLOCATABLE,DIMENSION(:) :: csrTSF
!   Set Point Factor Table
TYPE csTDPT
    INTEGER(sik)                :: ndmp
!   Dynamically dimensioned to csGl%ntdp
    INTEGER(sik),POINTER,DIMENSION(:) :: tripIDs
END TYPE csTDPT
!   Only One of These
TYPE (csTDPT) :: csTDP
TYPE (csTDPT) :: csrTDP
!   Trip Initiated Time Step Data
TYPE csTSDT
    INTEGER(sik)          :: ndid
    INTEGER(sik)          :: ntid
    INTEGER(sik),DIMENSION(5) :: tripIDs
    REAL(sdk)             :: dtmin
    REAL(sdk)             :: dtmax
    REAL(sdk)             :: dtend
    REAL(sdk)             :: dtsof
    REAL(sdk)             :: edint
    REAL(sdk)             :: gfint
    REAL(sdk)             :: dmpint
    REAL(sdk)             :: sedint
END TYPE csTSDT
```

```
!   Dynamically dimensioned to csGl%ntsd
TYPE (csTSDT),ALLOCATABLE,DIMENSION(:) :: csTSD
TYPE (csTSDT),ALLOCATABLE,DIMENSION(:) :: csrTSD
!   Constrained Steady-State Data
TYPE cssGlT
    INTEGER(sik)        :: ncontr
    INTEGER(sik)        :: nconts
    INTEGER(sik)        :: ncontt
    INTEGER(sik)        :: ncontp
END TYPE cssGlT
TYPE (cssGlT) :: cssGl
TYPE cssDatT
    INTEGER(sik)        :: num
    REAL(sdk)           :: amn
    REAL(sdk)           :: amx
    INTEGER(sik)        :: nmp
    INTEGER(sik)        :: nap
END TYPE cssDatT
!   Dynamically dimensioned to cssGl%ncontr
TYPE (cssDatT),ALLOCATABLE,DIMENSION(:) :: cssDat
TYPE cssTPT
    INTEGER(sik)            :: num
    REAL(sdk)               :: fac
    INTEGER(sik),DIMENSION(18)  :: numAr
END TYPE cssTPT
!   Dynamically dimensioned to cssGl%ncontp
TYPE (cssTPT),ALLOCATABLE,DIMENSION(:) :: cssTP
!
REAL(sdk) cpv(42),dsv(2)
```

DERIVED TYPE VARIABLES:

csGl	Control-system input data used to dimension variable storage.
csrGl	Control-system restart data used to dimension variable storage.
csCPED	Control parameter evaluation pass data.
csSig	Signal-variable input data.
csrSig	Signal-variable restart data.
csCB	Control-block input data.
csrCB	Control-block restart data.

csULCB	Control-block, units-labels input data.
csrULCB	Control-block, units-labels restart data.
csULTR	Trip units-labels input data.
csrULTR	Trip units-labels restart data.
csULSE	Signal-variable, units-labels input data.
csrULSE	Signal-variable, units-labels restart data.
csTrip	Trip input data.
csrTrip	Trip restart data.
csTSE	Trip signal-expression input data.
csrTSE	Trip signal-expression restart data.
csTCT	Trip-controlled-trip input data.
csrTCT	Trip-controlled trip restart data.
csTSF	Trip setpoint-factor input data.
csrTSF	Trip setpoint-factor restart data.
csTDP	Trip-initiated restart dump input data.
csrTDP	Trip-initiated restart dump restart data.
csTSD	Trip-initiated timestep input data.
csrTSD	Trip-initiated timestep restart data.
cssGl	Number of CSS controllers.
cssDat	CSS controller data.
cssTP	

REAL VARIABLES:

!Elements of DERIVED TYPE csGlT	
etime	problem time (seconds)
!Elements of DERIVED TYPE csSigT	
prevVal	
presVal	
!Elements of DERIVED TYPE csCBT	
cbgain	User's Guide Control Block Card 3—Control-block gain factor.
cbxmin	User's Guide Control Block Card 3—Control-block minimum value of its output parameter.
cbxmax	User's Guide Control Block Card 3—Control-block maximum value of its output parameter.
cbcon1	User's Guide Control Block Card 3—Control-block first constant.
cbcon2	User's Guide Control Block Card 3—Control-block second constant.
flagl	

flags

cbin(3)

prevVal

presVal

cbwt — User's Guide Control Block Card 6—The first constant `Wt` value (–) for the weighted summing function operation.

!

csCBTD — Control-block tabular input data.

csrCBTD — Control-block tabular restart data.

!Elements of DERIVED TYPE csULCBT

data — User's Guide Control Block Card 2—Used to input the control-block units labels.

!Elements of DERIVED TYPE csULTRT

data — User's Guide Trip Card 2—Used to input the trip units labels.

!Elements of DERIVED TYPE csULSET

data — Used to input the signal-variable units labels

!Elements of DERIVED TYPE csTripT

setp(*) — User's Guide Trip Card 3—The trip signal setpoint values.

dtsp(*) — User's Guide Trip Card 4—The setpoint delay times (s) after the trip signal crosses the setpoint value to when the trip set status is changed.

ifsp(*) — User's Guide Trip Card 5—The setpoint-factor-table ID numbers.

delsv

prevVal

setTime

timeData

!Elements of DERIVED TYPE csTSET

constants

!Elements of DERIVED TYPE csTSFT

setp(2,10) — User's Guide Trip Set Point Factor Table Card 2—Setpoint-factor-table data

!Elements of DERIVED TYPE csTSDT

dtmin — User's Guide Trip Initiated Time Step Data Card 3—The minimum timestep size (s).

dtmax — User's Guide Trip Initiated Time Step Data Card 3—The maximum timestep size (s).

dtend — User's Guide Trip Initiated Time Step Data Card 3—The problem time interval (s) during which these timestep data are used.

dtsof — User's Guide Trip Initiated Time Step Data Card 3—The next timestep `dtsof` (s) or the factor –`dtsof` (–) to be applied to the existing timestep in defining the timestep to be used at the start of the `dtend` time interval when implementing these special timestep data.

edint — User's Guide Trip Initiated Time Step Data Card 4—Long-print edit interval (s).

gfint — User's Guide Trip Initiated Time Step Data Card 4—Graphics edit interval (s).

dmpint — User's Guide Trip Initiated Time Step Data Card 4—Restart-dump edit interval (s).

sedint — User's Guide Trip Initiated Time Step Data Card 4—Short-print edit interval (s).

!Elements of DERIVED TYPE cssDatT

amn — User's Guide CSS-Controller Card—Minimum value to which the parameter action can be adjusted by the CSS controller

amx — User's Guide CSS-Controller Card—Maximum value to which the parameter action can be adjusted by the CSS controller

!Elements of DERIVED TYPE cssTPT

fac

!

cpv — Control-panel vector for storing the values of signal-variable parameter numbers 1 through 6 for the global parameters and 7 through 15 for up to four coolant loops (variable not used).

dsv — Dummy signal-variable vector for storing the values of signal-variable parameter numbers 16 and 17 (variable not used).

INTEGER VARIABLES:

!Elements of DERIVED TYPE csGIT

ntsv — User's Guide Main Data Card 7—Number of signal variables from input and the restart file.

ntcb — User's Guide Main Data Card 7—Number of control blocks from input and the restart file.

ntcf — User's Guide Main Data Card 7—Total number of table entries for the tabular control blocks from input and the restart file.

ntrp — User's Guide Main Data Card 7—Number of trips from input and the restart file.

ntcp — User's Guide Main Data Card 7—Number of passes made each timestep through the control-parameter evaluation of signal variables, control blocks, and trips.

ntse — User's Guide Trip Dimensions Variables Card—The number of different signal-expression trip signals from input and the restart file.

ntct — User's Guide Trip Dimensions Variables Card—The number of different trip-controlled-trip signals from input and the restart file.

ntsf — User's Guide Trip Dimensions Variables Card—The number of different setpoint-factor tables referenced by trips from input and the restart file.

ntdp — User's Guide Trip Dimensions Variables Card—The number of trips from input and the restart file that generate a restart dump and possible problem termination when they are set on.

ntsd — User's Guide Trip Dimensions Variables Card—The number of trip-controlled timestep data sets from input and the restart file that are used for timestep and edit control when their defined trips are set on.

!Elements of DERIVED TYPE csCPEDT

isv1 — User's Guide Control-Parameter List Card—The smallest signal-variable ID number evaluated during the first control-parameter evaluation pass.

isv2 — User's Guide Control-Parameter List Card—The largest signal-variable ID number evaluated during the first control-parameter evaluation pass.

icb1 — User's Guide Control-Parameter List Card—The smallest (in absolute value) control-block ID number evaluated during the first control-parameter evaluation pass.

icb2 — User's Guide Control-Parameter List Card—The largest (in absolute value) control-block ID number evaluated during the first control-parameter evaluation pass.

itp1 — User's Guide Control-Parameter List Card—The smallest (in absolute value) trip ID number evaluated during the first control-parameter evaluation pass.

itp2 — User's Guide Control-Parameter List Card—The largest (in absolute value) trip ID number evaluated during the first control-parameter evaluation pass.

!Elements of DERIVED TYPE csSigT

idsv — User's Guide Signal-Variable Card—Signal-variable ID number.

isvn — User's Guide Signal-Variable Card—Signal-variable parameter number.

ilcn — User's Guide Signal-Variable Card—The component number or the trip ID number of the signal-variable parameter.

icn1 User's Guide Signal-Variable Card—The cell, interface, or node number of the first location in component ILCN where the signal-variable parameter is defined.

icn2 User's Guide Signal-Variable Card—The cell, interface, or node number of the second location in component ILCN where the signal-variable parameter is defined.

!Elements of DERIVED TYPE csCBT

idcb User's Guide Control Block Card 1—Control-block ID number.

icbn User's Guide Control Block Card 1—Control-block operation number.

icb(1) User's Guide Control Block Card 1—ID number for the first input parameter to the control block.

icb(2) User's Guide Control Block Card 1—ID number for the second input parameter to the control block.

icb(3) User's Guide Control Block Card 1- ID number for the third input parameter to the control block.

tableIndex

!Elements of DERIVED TYPE csTripT

idtp User's Guide Trip Card 1—The trip ID number.

isrt User's Guide Trip Card 1—The signal-range-type number.

iset User's Guide Trip Card 1—The initial trip-set-status number.

itst User's Guide Trip Card 1—The trip-signal-type number.

idsg User's Guide Trip Card 1—The ID number for the trip signal variable when `itst` = ±1.

prevSub

subRange

!Elements of DERIVED TYPE csTSET

idse User's Guide Trip Signal-Expression Signal Card 1—The trip signal-expression signal ID number.

inse User's Guide Trip Signal-Expression Signal Card 1—The number of subexpressions defining the trip signal expression.

incn User's Guide Trip Signal-Expression Signal Card 1—The number of different constants referenced in the subexpressions defining the trip signal expression.

ids(1,10) User's Guide Trip Signal-Expression Signal Card 2—The arithmetic-operator ID number of the J^{th} arithmetic subexpression.

ids(2,10) User's Guide Trip Signal-Expression Signal Card 2—The first argument ID number of the J^{th} arithmetic subexpression.

ids(3,10) User's Guide Trip Signal-Expression Signal Card 2—The second argument ID number of the J^{th} arithmetic subexpression.

!Elements of DERIVED TYPE csTCTT

idtn User's Guide Trip-Controlled Trip-Signal Card 1—The trip-controlled-trip signal ID number.

intn User's Guide Trip-Controlled Trip-Signal Card 1—The number of trip ID numbers whose ISET set-status values are summed or multiplied to evaluate this trip-controlled-trip signal.

tripIDs(10) User's Guide Trip-Controlled Trip-Signal Card 2—The trip ID numbers where the ISET set-status values are summed (`idtn` > 0) or multiplied (`idtn` < 0) to evaluate this trip-controlled-trip signal.

!Elements of DERIVED TYPE csTSFT

idft User's Guide Trip Set Point Factor Table Card 1—The setpoint-factor-table ID number.

idsg User's Guide Trip Set Point Factor Table Card 1—The signal-variable or control-block ID number defining the setpoint-factor, table-independent variable.

inft User's Guide Trip Set Point Factor Table Card 1—The number of setpoint-factor table entry pairs.

!Elements of DERIVED TYPE csTDPT

ndmp User's Guide Trip-Initiated Restart Dump Card 1—The total number of trips from the input file and the restart dump that generate a restart dump and possible problem termination when any one of the trips is set to $ON_{reverse}$ or $ON_{forward}$.

tripIDs(:) User's Guide Trip-Initiated Restart Dump Card 2—The absolute value of the trip ID numbers that generate a restart data dump when any one of the trips is set to $ON_{reverse}$ or $ON_{forward}$.

!Elements of DERIVED TYPE csTSDT

ndid User's Guide Trip-Initiated Time Step Data Card 1.

ntid User's Guide Trip-Initiated Time Step Data Card 1.

tripIDs(:)

!Elements of DERIVED TYPE cssGlT

ncontr User's Guide Main-Data Card 6—Number of CSS controllers.

nconts

ncontt

ncontp

!Elements of DERIVED TYPE cssDatT

num User's Guide CSS-Controller Card—Component number where the parameter action is adjusted by the CSS controller.

nmp User's Guide CSS-Controller Card—Monitored-parameter number that is defined for each CSS controller type.

nap User's Guide CSS-Controller Card—Adjusted-parameter number that is defined for the last two controller types.

```
!Elements of DERIVED TYPE cssTPT
num
numArr
```

C.10. Module `EngUnits`

```
MODULE EngUnits
!
!  BEGIN MODULE USE
   USE IntrType
!
   IMPLICIT NONE
!
!  This is necessary to hide dummy indices for DATA statements
!  on the Cray
   public
   private :: i,j
   INTEGER(sik) i,j
!
!  To use clearer names
!
   INTEGER(sik) :: uTypeList,vNameList,allLists
   PARAMETER (uTypeList=1,vNameList=2, allLists=3)
   INTEGER(sik) ils,ilun,ilss
!
!  These parameters define the number of real variable labels (ils),
!  units (ilun), and signal-variable labels (ilss). They are also
!  used to dimension the associated arrays; therefore, these parameters must be
!  changed if units and labels are either added or deleted. The LabPrg
```

```
!   program will make this change automatically.
!
    PARAMETER (ils=777,ilun=51,ilss=105)
!
!   From labelv.h:
!
    CHARACTER*2 lud,lutp,lus
    CHARACTER*3 lut,luz
    CHARACTER*4 luar,lue,lum,luvo
    CHARACTER*5 lup,lupd,luv,luvf
    CHARACTER*6 luis
    CHARACTER*7 luen,lumf,lupt,lupw
    CHARACTER*8 luid,lur,lusp,lusz,lutm
    CHARACTER*9 luha
    CHARACTER*10 lucp
    CHARACTER*11 luph
    CHARACTER*12 luhx,lutc
    CHARACTER*13 luh
    CHARACTER*19 ludh
    CHARACTER*26 alpbet
    INTEGER(sik) ih(26),ilu,iOld,uOld
!
    Type UnitsT
     CHARACTER*8  :: Labun
     CHARACTER*13, DIMENSION(2)  :: Luncb
     CHARACTER*12, DIMENSION(2)  :: Runcb
     CHARACTER*13, DIMENSION(2)  :: Lupcb
     REAL(sdk)  :: Factor
     REAL(sdk)  :: Offset
    End Type UnitsT
```

```
!
   Type VarLabelsT
    CHARACTER*8  :: Labels
    CHARACTER*8  :: LabUnits
    INTEGER(sik) :: Itls
   End Type VarLabelsT
!
   Type SigLabelsT
    CHARACTER*14, DIMENSION(2)  :: Labsv
    CHARACTER*8  :: SigUnits
    INTEGER(sik) :: Itsv
   End Type SigLabelsT
!
   DATA ih,iOld,uOld/26*0,1,1/
   DATA alpbet/'abcdefghijklmnopqrstuvwxyz'/
   DATA luar,lucp,lud,ludh,lue,luen,luh,luha/8*' '/
   DATA luhx,luid,luis,lum,lumf,lup,lupd,luph/8*' '/
   DATA lupt,lupw,lur,lus,lusp,lusz,lut/7*' '/
   DATA lutc,lutm,lutp,luv,luvf,luvo,luz/7*' '/
!
   TYPE (UnitsT), DIMENSION(:), ALLOCATABLE :: Units
!
   TYPE (VarLabelsT), DIMENSION(ils) :: VarLabels
!
   TYPE (SigLabelsT), DIMENSION(ilss) :: SigLabels
```

DERIVED TYPE VARIABLES:

Units	Defines the units type name, units labels, and conversion factors (`Factor` and `Offset`).
VarLabels	Defines the units type and definition for real variables.
SigLabels	Defines the units type, units labels, and definition for signal variables.

REAL VARIABLES:

!Elements of DERIVED TYPE UnitsT

Factor	Conversion factor from SI units to English units.
Offset	Offset shift value from SI units to English units after the conversion factor is applied to the SI-units value.

INTEGER VARIABLES:

uTypeList	Parameter indicating that subroutine `LuMatch` should search the database for a units-type name.
vNameList	Parameter indicating that subroutine `LuMatch` should search the database for a real-variable label name.
allLists	Parameter indicating that subroutine `LuMatch` should search the databases for both a units-type and real-variable label name.
ils	Total number of Fortran I/O real-variable names in TRAC-M.
ilun	Index of the last units-name label in TRAC-M for the pre-defined units.
ilss	Total number of Fortran I/O signal-variable names in TRAC-M.
ih	`ih(i)` defines the index of the first Fortran I/O real-variable name in TRAC-M, beginning with the ith letter of the alphabet.
ilu	Index of the last units-name label defined internally in TRAC-M after user-defined, units-name labels are input.
iOld	Index of the units-name label for the last Fortran I/O real-variable name processed for possible units conversion by subroutine `uncnvt`.
uOld	Index of the units-name label for the last Fortran I/O units-type name processed for possible units conversion by subroutine `uncnvt`.

!Elements of DERIVED TYPE VarLabelsT

Itls	`Itls` defines the index of the units-name label defining the units of the Fortran I/O real-variable name in TRAC-M.

!Elements of DERIVED TYPE SigLabelsT

Itsv	`Itsv` defines the index of the units-name label defining the units of the signal-variable parameter.

CHARACTER VARIABLES:

lud	Commonly used units symbol "–" for a dash indicating no units.
lutp	Commonly used units symbol "k" or "f" for temperature.
lus	Commonly used units symbol "s" for time.
lut	Commonly used string "**".
luz	Commonly used units symbol "m" or "ft" for length.
luar	Commonly used units symbol "m2" or "ft2" for area.
lue	Commonly used units symbol "w*s" or "btu" for energy.
lum	Commonly used units symbol "kg" or "lbm" for mass.

luvo	Commonly used units symbol "m3" or "ft3" for volume.
lup	Commonly used units symbol "pa" or "psia" for absolute pressure.
lupd	Commonly used units symbol "pa" or "psid" for a pressure difference.
luv	Commonly used units symbol "m/s" or "ft/s" for velocity.
luvf	Commonly used units symbol "m3/s" or "gpm" for volumetric flow.
luis	Commonly used units symbol "rad/s" or "rpm" for the pump-impeller rotational speed.
luen	Commonly used string "end" or "end (s)".
lumf	Commonly used units symbol "kg/s" or "lbm/hr" for mass flow.
lupt	Commonly used units symbol "pa*m3" or "lbf*ft" for torque.
lupw	Commonly used units symbol "w" or "btu/hr" for power.
luid	Commonly used units symbol "kg/m4" or "lbm/ft4" for interfacial drag.
lur	Commonly used units symbol "kg/m3" or "lbm/ft3" for density.
lusp	Commonly used string "step" or "step (s)".
lusz	Commonly used string "size" or "size (s)"
lutm	Commonly used string "time" or "time (s)".
luha	Commonly used units symbol "w/k" or "btu/f/hr" for interfacial-heat-transfer total flux.
lucp	Commonly used units symbol "w*s/kg/k" or "btu/lbm/f" for specfic heat.
luph	Commonly used units symbol "m2/s2 " or "lbf*ft/lbm" for pump head.
luhx	Commonly used units symbol "w/m2" or "btu/ft2/hr" for heat flux.
lutc	Commonly used units symbol "w/m/k' or 'btu/ft/f/hr' for thermal conductivity.
luh	Commonly used units symbol 'w/m2/k" or "btu/ft2/f/hr" for a HTC.
ludh	Commonly used units symbol "—— w/m2/k ———" or "— btu/ft2/f/hr —" with dashes on each side for a HTC.
alpbet	String of the 26 letters of the alphabet.

!Elements of DERIVED TYPE UnitsT

Labun	Names of the units-name labels.
Luncb	Left-justified names of the SI-units and English-units symbols of the units-name labels.

Runcb	Right-justified names of the SI-units and English-units symbols of the units-name labels.
Lupcb	Names surrounded by parentheses of the SI-units and English-units symbols of the units-name labels.

!Elements of DERIVED TYPE VarLabelsT

Labels	Names of the FORTRAN I/O real variables in TRAC-M.
LabUnits	Units-type names of the FORTRAN I/O real variables.

!Elements of DERIVED TYPE SigLabelsT

Labsv	Names of the signal-variable parameters.
SigUnits	Units-type names of the FORTRAN I/O real variables.

C.11. Module `EosData`

```
REAL(sdk)  ceoslp(40)
REAL(sdk)  aeos14,ceos1,ceos2,ceos3
INTEGER(sik)  igas
DATA igas/1/
!
LOGICAL lh2o
INTEGER(sik) id2o
!
DATA id2o/0/
!
INTEGER(sik) istrt3,iend3,nvthm,ndimv1,nixnj,nsthm
!
PARAMETER (pkhe1=2.774d-03,pkhe2=0.701d0)
```

REAL VARIABLES:

aeos14	Constant in expression for saturation-temperature calculation at intermediate pressures (defined in subroutine `thermo`).
ceos1	First constant in expression for saturation-temperature calculation at intermediate pressures (defined in subroutine `thermo`).
ceos2	Second constant in expression for saturation-temperature calculation at intermediate pressures (defined in subroutine `thermo`).
ceos3	Third constant in expression for saturation-temperature calculation at intermediate pressures (defined in subroutine `thermo`).
ceoslp	EOS array for low pressures (defined in subroutine `seteos`).

pkhe1 Constant used in calculating nonideal helium conductivity in subroutines `mgap` and `seteos`.

pkhe2 Constant used in calculating nonideal helium conductivity in subroutines `mgap` and `seteos`.

INTEGER VARIABLES:

igas Noncondensable-gas type option (namelist variable).

1 = air;

2 = hydrogen;

3 = helium (ideal gas);

4 = helium (nonideal gas).

!

id2o Option for fluid properties.

1 = H2O properties;

2 = D2O properties.

!

istrt3 First cell number (`ico`) in the VESSEL component r- or x-direction.

iend3 Last cell number (`icx`) in the VESSEL component r- or x-direction.

nvthm Number of different array parameters in the `equiv` common block for a VESSEL component.

ndimv1 `nvthm` times the total number of r- or x-direction calculation plus pseudo cells dimensioned for.

nixnj `ndimv1` times the total number of θ- or y-direction calculation plus pseudo cells dimensioned for.

nsthm

LOGICAL VARIABLES:

lh2o Logic flag for fluid properties.

.TRUE. = H2O properties;

.FALSE. = D2O properties.

C.12. Module `EosInline`

INTEGER(sik),PARAMETER,PRIVATE :: sdkx=sdk

INTEGER VARIABLES:

sdkx Required to declare real KIND functions with Nagware Fortran 90 compiler.

C.13. Module `EosNoInline`

INTEGER(sik),PARAMETER,PRIVATE :: sdkx=sdk

INTEGER VARIABLES:

sdkx	Required to declare real KIND functions with Nagware Fortran 90 compiler.

C.14. Module `FailDat`

```
REAL(sdk) antest,atest1,dara,darl,darv,ddvl,ddvv,dtll,dtllm,     &
& dtlu,dtlum,dtvl,dtvlm,dtvu,dtvum,timdl,timdu,tsdlt,tsdut
INTEGER(sik)  iatest,ichga,ilreit,iptest,ivtest,jatest,jdara,    &
& jdarl,jdarv,jddvl,jddvv,jdtll,jdtlu,jdtvl,jdtvu,jptest,jvtest,  &
& kptest,ndara,ndarl,ndarv,nddvl,nddvv,ndtll,ndtlu,ndtvl,ndtvu,   &
& nptest,nsdl,nsdu,tsdls,tsdus
!
DATA tsdls,tsdus,tsdlt,tsdut/-1,-1,-1.0d0,-1.0d0/
DATA nsdl,nsdu,timdl,timdu/-1,-1,-1.d0,-1.d0/
DATA ddvv,ddvl/0.d0,0.d0/
DATA dtvlm,dtvum,dtllm,dtlum/2.d0,2.d0,2.d0,2.d0/
DATA ilreit,ichga/0,0/
DATA darv,darl/1.d-20,1.d-20/
```

REAL VARIABLES:

antest	End-of-timestep gas-volume fraction that is outside its 0.0- to 1.0-value range in mesh cell jatest of component `iatest`.
atest1	Beginning-of-timestep gas-volume fraction in mesh cell `jatest` of component `iatest`.
dara	Maximum change in $\alpha\rho_a$.
darl	Measure of the maximum difference in $(1-\alpha)\rho_\ell$ between the basic and stabilizer steps.
darv	Measure of the maximum difference in $\alpha\rho_g$ between the basic and stabilizer steps.
ddvl	Measure of the maximum difference in V_ℓ between the basic and stabilizer steps.
ddvv	Measure of the maximum difference in V_g between the basic and stabilizer steps.
dtll	Largest decrease in T_ℓ from the current iteration.
dtllm	`dtvlm` and `dtllm` are limits on `dtvl` and `dtll` beyond which another iteration must be performed.

dtlu	Largest increase in T_ℓ from the current iteration.
dtlum	`dtvlm` and `dtllm` are limits on `dtvl` and `dtll` beyond which another iteration must be performed.
dtvl	Largest decrease in gas temperature in a given iteration.
dtvlm	`dtvlm` and `dtllm` are limits on `dtvl` and `dtll` beyond which another iteration must be performed.
dtvu	Largest increase in T_g from the current iteration.
dtvum	`dtvlm` and `dtllm` are limits on `dtvl` and `dtll` beyond which another iteration must be performed.
timdl	If `timdl` ≤ `timet` ≤ `timdu` for the problem time, details of `darv`, etc., should be output.
timdu	If `timdl` ≤ `timet` ≤ `timdu` for the problem time, details of `darv`, etc., should be output.
tsdlt	Starting time at which detailed timestep-diagnostic information on the logic used to evaluate the timestep size is output to file TRCMSG.
tsdut	Ending time at which detailed timestep-diagnostic information on the logic used to evaluate the timestep size is output to file TRCMSG.

INTEGER VARIABLES:

iatest	Component number with an out-of-range, gas-volume-fraction value.
ichga	Flag to print the maximum gas-volume-fraction changes to the TRCMSG file.
ilreit	Flag that allows reiteration messages when equation set changes.
iptest	VESSEL radial-r or x direction mesh-cell number having maximum $\lvert \delta p/p \rvert$.
ivtest	Component number having a velocity that changed its numerical sign during the last outer iteration.
jatest	Mesh-cell number in component iatest where the gas-volume fraction has an out-of-range value.
jdara	Cell number where `dara` occurred.
jdarl	Cell number where `darl` occurred.
jdarv	Cell number where `darv` occurred.
jddvl	Cell number where `ddvl` occurred.
jddvv	Cell number where `ddvv` occurred.
jdtll	Cell number where `dtll` occurred.
jdtlu	Cell number where `dtlu` occurred.
jdtvl	Cell number where `dtvl` occurred.

jdtvu	Cell number where dtvu occurred.
jptest	VESSEL azimuthal θ or y direction mesh-cell number or 1D component mesh-cell number with maximum $\lvert \delta p/p \rvert$.
jvtest	Mesh-cell interface number in component ivtest with a velocity that changed numerical sign during the last outer iteration.
kptest	VESSEL axial-z direction mesh-cell number with maximum $\lvert \delta p/p \rvert$.
ndara	Component number where dara occurred.
ndarl	Component number where darl occurred.
ndarv	Component number where darv occurred.
nddvl	Component number where ddvl occurred.
nddvv	Component number where ddvv occurred.
ndtll	Component number where dtll occurred.
ndtlu	Component number where dtlu occurred.
ndtvl	Component number where dtvl occurred.
ndtvu	Component number where dtvu occurred.
nptest	Component number with maximum $\lvert \delta p/p \rvert$.
nsdl	If nsdl ≤ nstep ≤ nsdu for the timestep number, a detailed diagnostic of darv, etc., should be output to the TRCOUT file and idiag = 3 diagnostics to the TRCMSG file.
nsdu	If nsdl ≤ nstep ≤ nsdu for the timestep number, a detailed diagnostic of darv, etc., should be output to the TRCOUT file and idiag = 3 diagnostics to the to TRCMSG.
tsdls	First timestep number where detailed timestep-diagnostic information on the logic used to evalaute the timestep size is output to file TRCMSG.
tsdus	Last timestep number where detailed timestep-diagnostic information on the logic used to evalaute the timestep size is output to file TRCMSG.

C.15. Module FillArray

```
TYPE fillArrayT
    REAL(sdk), POINTER, DIMENSION(:) :: alptb
    REAL(sdk), POINTER, DIMENSION(:) :: contb
    REAL(sdk), POINTER, DIMENSION(:) :: patb
    REAL(sdk), POINTER, DIMENSION(:) :: ptb
    REAL(sdk), POINTER, DIMENSION(:) :: rftb
    REAL(sdk), POINTER, DIMENSION(:) :: tltb
```

```
    REAL(sdk),   POINTER, DIMENSION(:)  :: tvtb
    REAL(sdk),   POINTER, DIMENSION(:)  :: vmtb
    REAL(sdk),   POINTER, DIMENSION(:)  :: vvtb
  END TYPE fillArrayT
!
TYPE (fillArrayT),DIMENSION(maxComps) :: fillAr
```

DERIVED TYPE VARIABLES:

fillAr	Pointer table for the FILL component.

REAL VARIABLES:

!Elements of DERIVED TYPE fillArrayT

alptb	Pointer to the gas-volume-fraction table.
contb	Pointer to the ratio of solute-mass-to-liquid-mass table.
ptab	Pointer to the noncondensable-gas, partial-pressure table.
ptb	Pointer to the total-pressure table.
rftb	Pointer to the rate-factor table.
tltb	Pointer to the liquid-temperature table.
tvtb	Pointer to the gas-temperature table.

C.16. Module `FillVlt`

```
TYPE fillTabT
    REAL(sdk) alpoff
    REAL(sdk) conoff
    REAL(sdk) flowin
    REAL(sdk) flwoff
    REAL(sdk) fsmass
    REAL(sdk) fxmass
    REAL(sdk) paoff
    REAL(sdk) poff
    REAL(sdk) rfmx
    REAL(sdk) tloff
    REAL(sdk) tvoff
    REAL(sdk) twtold
    REAL(sdk) vloff
    REAL(sdk) vvoff
    REAL(sdk) type1
    INTEGER(sik) icj
```

```
      INTEGER(sik) iff
      INTEGER(sik) ifp
      INTEGER(sik) ifsv
      INTEGER(sik) iftr
      INTEGER(sik) ifty
      INTEGER(sik) inexti
      INTEGER(sik) ioff
      INTEGER(sik) ifmlsv
      INTEGER(sik) ifmvsv
      INTEGER(sik) iftlsv
      INTEGER(sik) iftvsv
      INTEGER(sik) ifasv
      INTEGER(sik) ifpsv
      INTEGER(sik) ifpasv
      INTEGER(sik) ifcnsv
      INTEGER(sik) ionoff
      INTEGER(sik) js1
      INTEGER(sik) jun1
      INTEGER(sik) nfrf
      INTEGER(sik) nfsv
      INTEGER(sik) nftb
      INTEGER(sik) js1get
      INTEGER(sik) js1put
    END TYPE fillTabT
!
TYPE(fillTabT),DIMENSION(maxComps) :: fillTab
!
!   This is used to calculate the total dump size and must
!   be adjusted when changes are made to fillTab.
!
INTEGER(sik) fillDumpSize
PARAMETER (fillDumpSize=39)
```

DERIVED TYPE VARIABLES:

fillTAB The VLT for the FILL component.

REAL VARIABLES:

!Elements of DERIVED TYPE fillTabT

alpoff Gas-volume fraction when the trip is off after it was on.

conoff	Ratio of solute mass to liquid mass when the trip is off after it was on.
flowin	Initial fluid mass flow to or from adjacent component.
flwoff	Fluid mass flow when the trip is off after it was on.
fsmass	Time-integrated fluid mass flow out of the FILL.
fxmass	Current fluid mass-flow rate out of the FILL.
paoff	Noncondensable-gas partial pressure when the trip is off after it was on.
poff	Total pressure when the trip is off after it was on.
rfmx	Maximum rate of change of FILL velocity or mass flow.
tloff	Liquid temperature when the trip is off after it was on.
tvoff	Gas temperature when the trip is off after it was on.
twtold	Fraction of a previous FILL fluid-dynamic-state parameter that is averaged with the FILL table's defined parameter to define the FILL parameter value for this timestep ($0.0 \leq$ TWTOLD ≤ 1.0).
vloff	Liquid velocity when the trip is off after it was on.
vvoff	Gas velocity when the trip is off after it was on.
type1	Variable no longer used.

INTEGER VARIABLES:

!Elements of DERIVED TYPE fillTabT

icj	Iteration index of adjacent component.
iff	Last interpolated interval in the rate-factor table.
ifp	Last interpolated interval in the FILL table.
ifsv	The signal-variable ID number, which defines the FILL table's independent variable.
iftr	FILL trip ID number.
ifty	FILL-type option.
inexti	Variable no longer used.
ioff	FILL fluid-state option when the trip is off after it was on.
ifmlsv	Signal-variable or control-block ID number defining liquid mass flow in the FILL cell for the `ifty` = 10 option.
ifmvsv	Signal-variable or control-block ID number defining gas mass flow in the FILL cell for the `ifty` = 10 option.
iftlsv	ID number of the signal variable or control block defining the liquid temperature in the FILL cell for the `ifty` = 10 option.
iftvsv	ID number of the signal variable or control block defining the vapor temperature in the FILL cell for the `ifty` = 10 option.
ifasv	Signal-variable or control-block ID number defining the gas-volume fraction in the FILL cell for the `ifty` = 10 option.

ifpsv — ID number of the signal variable or control block defining the total pressure in the FILL cell for the `ifty` = 10 option.

ifpasv — ID number of the signal variable or control block defining the noncondensable-gas partial pressure in the FILL cell for the `ifty` = 10 option.

ifcnsv — Signal-variable or control-block ID number defining the ratio of solute mass to liquid mass in the FILL cell for the `ifty` = 10 option.

ionoff — The number of timesteps the trip has been on.

js1 — Junction sequence number at JUN1.

jun1 — Junction number where the FILL is located.

nfrf — Number of rate-factor table data pairs where the rate factor is applied to the FILL table's independent variable.

nfsv — Signal-variable or control-block ID number defining the rate-factor table's abscissa-coordinate variable.

nftb — Number of data pairs in the FILL table.

js1get — Index for getting information from `bd` array.

js1put — Index for putting information into `bd` array.

!

fillDumpSize — Number of variables in the derived-type `fillTabT`.

C.17. Module `Flt`

TYPE genTabT
- REAL(sdk) htlsci
- REAL(sdk) htlsco
- REAL(sdk) pinteg
- REAL(sdk) title(4)
- INTEGER(sik) icflg
- INTEGER(sik) id
- INTEGER(sik) irest
- INTEGER(sik) typeIndex
- INTEGER(sik) lenvlt
- INTEGER(sik) lextra
- INTEGER(sik) ncellt
- INTEGER(sik) numbm1
- INTEGER(sik) numbm2
- INTEGER(sik) numbm3
- INTEGER(sik) numbn1

```
    INTEGER(sik) numbn2
    INTEGER(sik) numbn3
    INTEGER(sik) nodes
    INTEGER(sik) num
    REAL(sdk) type
END TYPE genTabT
!
TYPE(genTabT),DIMENSION(maxComps) :: genTab
!
!   This is used in calculating the total dump length and must
!   be adjusted when adding variables to genTab
!
INTEGER(sik) genDumpSize
PARAMETER (genDumpSize=23)
```

DERIVED TYPE VARIABLES:

genTAB	The FLT that is generic to all components.

REAL VARIABLES:

!Elements of DERIVED TYPE genTabT

htlsci	Component-wall, inner-surface HTC.
htlsco	Component-wall, outer-surface HTC.
pinteg	Energy (such as resistance heating) into a 1D hydraulic component wall.
title	Component description.
type	Component type.

INTEGER VARIABLES:

!Elements of DERIVED TYPE genTabT

icflg	Cell-edge, choked-flow model option.
id	Component identification number.
irest	Component restart indicator.
typeIndex	
lenvlt	Length of the VLT.
lextra	Length of nonstandard dump for components.
ncellt	Total number of cells.
numbm1	Index to access large numerals for printing component num and iorder.
numbm2	Index to access large numerals for printing component num and iorder.

numbm3	Index to access large numerals for printing component num and iorder.
numbn1	Index to access large numerals for printing component num and iorder.
numbn2	Index to access large numerals for printing component num and iorder.
numbn3	Index to access large numerals for printing component num and iorder.
nodes	Number of heat-transfer nodes.
num	Component number.
maxComp	Maximum number of components.
!	
genDumpSize	Number of variables in derived-type genTabT.

C.18. Module Gen1DArray

```
INTEGER(sik),PARAMETER :: sdkx=sdk
!
!  These indices are used as arguments to the Gen1DArray
!  data interface:
!
   INTEGER(sik), PARAMETER ::  hgamInd= 1
   INTEGER(sik), PARAMETER ::   hlaInd= 2
   INTEGER(sik), PARAMETER ::   hvaInd= 3
   INTEGER(sik), PARAMETER :: hlatwInd= 4
   INTEGER(sik), PARAMETER :: hvatwInd= 5
   INTEGER(sik), PARAMETER ::   watInd= 6
   INTEGER(sik), PARAMETER :: finanInd= 7
   INTEGER(sik), PARAMETER ::   alpInd= 8
   INTEGER(sik), PARAMETER ::   alvInd= 9
   INTEGER(sik), PARAMETER ::   cplInd= 10
   INTEGER(sik), PARAMETER ::   cpvInd= 11
   INTEGER(sik), PARAMETER ::    clInd= 12
   INTEGER(sik), PARAMETER ::    cvInd= 13
   INTEGER(sik), PARAMETER ::  hdhtInd= 14
   INTEGER(sik), PARAMETER ::  chtiInd= 15
   INTEGER(sik), PARAMETER ::     pInd= 16
   INTEGER(sik), PARAMETER ::   rovInd= 17
```

```
INTEGER(sik), PARAMETER ::  rolInd= 18
INTEGER(sik), PARAMETER ::  volInd= 19
INTEGER(sik), PARAMETER ::  hfgInd= 20
INTEGER(sik), PARAMETER ::  sigInd= 21
INTEGER(sik), PARAMETER ::   tlInd= 22
INTEGER(sik), PARAMETER ::  tlnInd= 23
INTEGER(sik), PARAMETER :: tsatInd= 24
INTEGER(sik), PARAMETER ::   tvInd= 25
INTEGER(sik), PARAMETER ::  tvnInd= 26
INTEGER(sik), PARAMETER :: vislInd= 27
INTEGER(sik), PARAMETER :: visvInd= 28
INTEGER(sik), PARAMETER ::   vvInd= 29
INTEGER(sik), PARAMETER ::   vlInd= 30
INTEGER(sik), PARAMETER ::   vmInd= 31
INTEGER(sik), PARAMETER ::   evInd= 32
INTEGER(sik), PARAMETER ::   elInd= 33
INTEGER(sik), PARAMETER ::   eaInd= 34
INTEGER(sik), PARAMETER ::  roaInd= 35
INTEGER(sik), PARAMETER ::   paInd= 36
INTEGER(sik), PARAMETER :: tssnInd= 37
INTEGER(sik), PARAMETER :: concInd= 38
INTEGER(sik), PARAMETER ::    sInd= 39
INTEGER(sik), PARAMETER :: rmvmInd= 40
INTEGER(sik), PARAMETER ::  vmnInd= 41
INTEGER(sik), PARAMETER :: cifnInd= 42
INTEGER(sik), PARAMETER ::   snInd= 43
INTEGER(sik), PARAMETER :: gamnInd= 44
INTEGER(sik), PARAMETER :: alpnInd= 45
INTEGER(sik), PARAMETER ::  hivInd= 46
INTEGER(sik), PARAMETER ::  hilInd= 47
INTEGER(sik), PARAMETER ::  panInd= 48
INTEGER(sik), PARAMETER ::  eanInd= 49
INTEGER(sik), PARAMETER ::  elnInd= 50
INTEGER(sik), PARAMETER ::  evnInd= 51
INTEGER(sik), PARAMETER :: rolnInd= 52
INTEGER(sik), PARAMETER :: rovnInd= 53
INTEGER(sik), PARAMETER ::  romInd= 54
```

```
  INTEGER(sik), PARAMETER ::  roanInd= 55
  INTEGER(sik), PARAMETER ::    amInd= 56
  INTEGER(sik), PARAMETER ::    hdInd= 57
  INTEGER(sik), PARAMETER :: concnInd= 58
  INTEGER(sik), PARAMETER ::   vlnInd= 59
  INTEGER(sik), PARAMETER ::   vvnInd= 60
  INTEGER(sik), PARAMETER ::  rvmfInd= 61
  INTEGER(sik), PARAMETER ::    pnInd= 62
  INTEGER(sik), PARAMETER ::    dxInd= 63
  INTEGER(sik), PARAMETER :: chtinInd= 64
  INTEGER(sik), PARAMETER ::  alvnInd= 65
  INTEGER(sik), PARAMETER ::    faInd= 66
  INTEGER(sik), PARAMETER :: faLiqFracInd = 67
  INTEGER(sik), PARAMETER :: faVapFracInd = 68
  INTEGER(sik), PARAMETER :: faArvInd = 69
  INTEGER(sik), PARAMETER :: faArlInd = 70
  INTEGER(sik), PARAMETER :: faArevInd = 71
  INTEGER(sik), PARAMETER :: faArelInd = 72
  INTEGER(sik), PARAMETER :: faAraInd = 73
  INTEGER(sik), PARAMETER :: faArcInd = 74
  INTEGER(sik), PARAMETER :: faWlPInd = 75
  INTEGER(sik), PARAMETER :: faWlMInd = 76
  INTEGER(sik), PARAMETER :: faWvPInd = 77
  INTEGER(sik), PARAMETER :: faWvMInd = 78
  INTEGER(sik), PARAMETER :: faLiqInd = 79
  INTEGER(sik), PARAMETER :: faVapInd = 80
!
  INTEGER(sik),PARAMETER :: num1DFaceArrays=80
!
  TYPE array1DPtrT
   LOGICAL :: isAssociated
   REAL(sdk),POINTER,DIMENSION(:) :: array1DPtr
  END TYPE array1DPtrT
!
  TYPE arrayNodeT
!  Array(comp index) of Pointers to this 1-D Array
   TYPE(array1DPtrT),DIMENSION(maxComps) :: array1DPtrs
```

```
   END TYPE arrayNodeT
!
   TYPE(arrayNodeT),DIMENSION(num1DFaceArrays) :: faceArs
!
!
!  Generic 1-D arrays
!
   TYPE g1DArrayT
    REAL(sdk), POINTER, DIMENSION(:)   :: driv
!
!    The following are currently only associated in
!    module PLENUM—in other 1D, offsets into driv
!    are used.
!
     REAL(sdk), POINTER, DIMENSION(:) :: dtsdp
     REAL(sdk), POINTER, DIMENSION(:) :: deldp
     REAL(sdk), POINTER, DIMENSION(:) :: devdp
     REAL(sdk), POINTER, DIMENSION(:) :: deldt
     REAL(sdk), POINTER, DIMENSION(:) :: devdt
     REAL(sdk), POINTER, DIMENSION(:) :: drolp
     REAL(sdk), POINTER, DIMENSION(:) :: drovp
     REAL(sdk), POINTER, DIMENSION(:) :: drolt
     REAL(sdk), POINTER, DIMENSION(:) :: drovt
     REAL(sdk), POINTER, DIMENSION(:) :: hvst
     REAL(sdk), POINTER, DIMENSION(:) :: hlst
     REAL(sdk), POINTER, DIMENSION(:) :: dhvsp
     REAL(sdk), POINTER, DIMENSION(:) :: dhlsp
     REAL(sdk), POINTER, DIMENSION(:) :: dtssp
     REAL(sdk), POINTER, DIMENSION(:) :: devat
     REAL(sdk), POINTER, DIMENSION(:) :: devap
     REAL(sdk), POINTER, DIMENSION(:) :: drvap
     REAL(sdk), POINTER, DIMENSION(:) :: drvat
!
!    Balance of hydro pointers:
!
     REAL(sdk), POINTER, DIMENSION(:) :: cfz
     REAL(sdk), POINTER, DIMENSION(:) :: dfldp
```

```
REAL(sdk), POINTER, DIMENSION(:) :: dfvdp
REAL(sdk), POINTER, DIMENSION(:) :: cl
REAL(sdk), POINTER, DIMENSION(:) :: cpl
REAL(sdk), POINTER, DIMENSION(:) :: cpv
REAL(sdk), POINTER, DIMENSION(:) :: cv
REAL(sdk), POINTER, DIMENSION(:) :: dx
REAL(sdk), POINTER, DIMENSION(:) :: fa
REAL(sdk), POINTER, DIMENSION(:) :: fric
REAL(sdk), POINTER, DIMENSION(:) :: fricr
REAL(sdk), POINTER, DIMENSION(:) :: grav
REAL(sdk), POINTER, DIMENSION(:) :: alpo
REAL(sdk), POINTER, DIMENSION(:) :: am
REAL(sdk), POINTER, DIMENSION(:) :: h1
REAL(sdk), POINTER, DIMENSION(:) :: h2
REAL(sdk), POINTER, DIMENSION(:) :: h3
REAL(sdk), POINTER, DIMENSION(:) :: arc
REAL(sdk), POINTER, DIMENSION(:) :: hd
REAL(sdk), POINTER, DIMENSION(:) :: hdht
REAL(sdk), POINTER, DIMENSION(:) :: hfg
REAL(sdk), POINTER, DIMENSION(:,:) :: qppp
REAL(sdk), POINTER, DIMENSION(:) :: rarl
REAL(sdk), POINTER, DIMENSION(:) :: rarv
REAL(sdk), POINTER, DIMENSION(:) :: rhs
REAL(sdk), POINTER, DIMENSION(:) :: rmem
REAL(sdk), POINTER, DIMENSION(:) :: rmvm
REAL(sdk), POINTER, DIMENSION(:) :: rom
REAL(sdk), POINTER, DIMENSION(:) :: rvmf
REAL(sdk), POINTER, DIMENSION(:) :: sig
REAL(sdk), POINTER, DIMENSION(:,:) :: trid
REAL(sdk), POINTER, DIMENSION(:) :: tsat
REAL(sdk), POINTER, DIMENSION(:) :: tssn
REAL(sdk), POINTER, DIMENSION(:) :: visl
REAL(sdk), POINTER, DIMENSION(:) :: visv
REAL(sdk), POINTER, DIMENSION(:) :: vol
REAL(sdk), POINTER, DIMENSION(:) :: vrv
REAL(sdk), POINTER, DIMENSION(:) :: vr
REAL(sdk), POINTER, DIMENSION(:) :: wa
```

```
    REAL(sdk), POINTER, DIMENSION(:) :: vlx
    REAL(sdk), POINTER, DIMENSION(:) :: vvx
    REAL(sdk), POINTER, DIMENSION(:) :: wfl
    REAL(sdk), POINTER, DIMENSION(:) :: wfv
    REAL(sdk), POINTER, DIMENSION(:) :: wfmfl
    REAL(sdk), POINTER, DIMENSION(:) :: wfmfv
    REAL(sdk), POINTER, DIMENSION(:) :: favol
    REAL(sdk), POINTER, DIMENSION(:) :: vvvol
    REAL(sdk), POINTER, DIMENSION(:) :: vlvol
    REAL(sdk), POINTER, DIMENSION(:) :: grvol
    REAL(sdk), POINTER, DIMENSION(:) :: alpmn
    REAL(sdk), POINTER, DIMENSION(:) :: alpmx
    REAL(sdk), POINTER, DIMENSION(:) :: vlvc
    REAL(sdk), POINTER, DIMENSION(:) :: fsmlt
    REAL(sdk), POINTER, DIMENSION(:) :: vlalp
    REAL(sdk), POINTER, DIMENSION(:) :: elev
    REAL(sdk), POINTER, DIMENSION(:) :: qp3f
    REAL(sdk), POINTER, DIMENSION(:) :: dalva
    REAL(sdk), POINTER, DIMENSION(:) :: hgam
    REAL(sdk), POINTER, DIMENSION(:) :: hla
    REAL(sdk), POINTER, DIMENSION(:) :: hva
    REAL(sdk), POINTER, DIMENSION(:) :: hlatw
    REAL(sdk), POINTER, DIMENSION(:) :: hvatw
    REAL(sdk), POINTER, DIMENSION(:) :: finan
    REAL(sdk), POINTER, DIMENSION(:) :: wat
    REAL(sdk), POINTER, DIMENSION(:) :: regnm
!   The following variables ending in "sm" are used for
!   DOE proprietary updates.  Please do not remove.
    REAL(sdk), POINTER, DIMENSION(:) :: xsm
    REAL(sdk), POINTER, DIMENSION(:) :: ysm
    REAL(sdk), POINTER, DIMENSION(:) :: zsm
    REAL(sdk), POINTER, DIMENSION(:) :: rsm
    REAL(sdk), POINTER, DIMENSION(:) :: r0sm
    REAL(sdk), POINTER, DIMENSION(:) :: nfvsm
    REAL(sdk), POINTER, DIMENSION(:) :: nf1sm
    REAL(sdk), POINTER, DIMENSION(:) :: nf2sm
    REAL(sdk), POINTER, DIMENSION(:) :: nf3sm
```

```
      REAL(sdk), POINTER, DIMENSION(:) :: nfv4sm
      REAL(sdk), POINTER, DIMENSION(:) :: nflsm
      REAL(sdk), POINTER, DIMENSION(:) :: nfl4sm
      REAL(sdk), POINTER, DIMENSION(:) :: uvsm
      REAL(sdk), POINTER, DIMENSION(:) :: nfcvsm
      REAL(sdk), POINTER, DIMENSION(:) :: nfclsm
      REAL(sdk), POINTER, DIMENSION(:) :: vvsm
      REAL(sdk), POINTER, DIMENSION(:) :: vlsm
!
      REAL(sdk), POINTER, DIMENSION(:) :: qrl
      REAL(sdk), POINTER, DIMENSION(:) :: qrv
      REAL(sdk), POINTER, DIMENSION(:) :: faLiq
      REAL(sdk), POINTER, DIMENSION(:) :: faVap
      REAL(sdk), POINTER, DIMENSION(:) :: faLiqFrac
      REAL(sdk), POINTER, DIMENSION(:) :: faVapFrac
      REAL(sdk), POINTER, DIMENSION(:) :: faArl
      REAL(sdk), POINTER, DIMENSION(:) :: faArv
      REAL(sdk), POINTER, DIMENSION(:) :: faArel
      REAL(sdk), POINTER, DIMENSION(:) :: faArev
      REAL(sdk), POINTER, DIMENSION(:) :: faAra
      REAL(sdk), POINTER, DIMENSION(:) :: faArc
      REAL(sdk), POINTER, DIMENSION(:) :: faWvM
      REAL(sdk), POINTER, DIMENSION(:) :: faWvP
      REAL(sdk), POINTER, DIMENSION(:) :: faWlM
      REAL(sdk), POINTER, DIMENSION(:) :: faWlP
!
!     Dual-time pointers:
!
      REAL(sdk), POINTER, DIMENSION(:) :: alp
      REAL(sdk), POINTER, DIMENSION(:) :: alpd
      REAL(sdk), POINTER, DIMENSION(:) :: alv
      REAL(sdk), POINTER, DIMENSION(:) :: alve
      REAL(sdk), POINTER, DIMENSION(:) :: ara
      REAL(sdk), POINTER, DIMENSION(:) :: arel
      REAL(sdk), POINTER, DIMENSION(:) :: arev
      REAL(sdk), POINTER, DIMENSION(:) :: arl
      REAL(sdk), POINTER, DIMENSION(:) :: arv
```

```
REAL(sdk), POINTER, DIMENSION(:) :: chti
REAL(sdk), POINTER, DIMENSION(:) :: chtia
REAL(sdk), POINTER, DIMENSION(:) :: conc
REAL(sdk), POINTER, DIMENSION(:) :: ea
REAL(sdk), POINTER, DIMENSION(:) :: el
REAL(sdk), POINTER, DIMENSION(:) :: ev
REAL(sdk), POINTER, DIMENSION(:) :: gam
REAL(sdk), POINTER, DIMENSION(:) :: p
REAL(sdk), POINTER, DIMENSION(:) :: pa
REAL(sdk), POINTER, DIMENSION(:) :: roa
REAL(sdk), POINTER, DIMENSION(:) :: rol
REAL(sdk), POINTER, DIMENSION(:) :: rov
REAL(sdk), POINTER, DIMENSION(:) :: s
REAL(sdk), POINTER, DIMENSION(:) :: td
REAL(sdk), POINTER, DIMENSION(:) :: tl
REAL(sdk), POINTER, DIMENSION(:) :: tv
REAL(sdk), POINTER, DIMENSION(:) :: vm
REAL(sdk), POINTER, DIMENSION(:,:) :: tw
REAL(sdk), POINTER, DIMENSION(:) :: vl
REAL(sdk), POINTER, DIMENSION(:) :: vv
REAL(sdk), POINTER, DIMENSION(:) :: bit
REAL(sdk), POINTER, DIMENSION(:) :: cif
REAL(sdk), POINTER, DIMENSION(:) :: hilo
REAL(sdk), POINTER, DIMENSION(:) :: hivo
REAL(sdk), POINTER, DIMENSION(:) :: higo
REAL(sdk), POINTER, DIMENSION(:) :: qppco
REAL(sdk), POINTER, DIMENSION(:) :: vlto
REAL(sdk), POINTER, DIMENSION(:) :: vvto
REAL(sdk), POINTER, DIMENSION(:) :: twa
REAL(sdk), POINTER, DIMENSION(:) :: twe
REAL(sdk), POINTER, DIMENSION(:) :: tce
REAL(sdk), POINTER, DIMENSION(:) :: alpn
REAL(sdk), POINTER, DIMENSION(:) :: alpdn
REAL(sdk), POINTER, DIMENSION(:) :: alvn
REAL(sdk), POINTER, DIMENSION(:) :: alven
REAL(sdk), POINTER, DIMENSION(:) :: aran
REAL(sdk), POINTER, DIMENSION(:) :: areln
```

```
  REAL(sdk), POINTER, DIMENSION(:) :: arevn
  REAL(sdk), POINTER, DIMENSION(:) :: arln
  REAL(sdk), POINTER, DIMENSION(:) :: arvn
  REAL(sdk), POINTER, DIMENSION(:) :: chtin
  REAL(sdk), POINTER, DIMENSION(:) :: chtan
  REAL(sdk), POINTER, DIMENSION(:) :: concn
  REAL(sdk), POINTER, DIMENSION(:) :: ean
  REAL(sdk), POINTER, DIMENSION(:) :: eln
  REAL(sdk), POINTER, DIMENSION(:) :: evn
  REAL(sdk), POINTER, DIMENSION(:) :: gamn
  REAL(sdk), POINTER, DIMENSION(:) :: pn
  REAL(sdk), POINTER, DIMENSION(:) :: pan
  REAL(sdk), POINTER, DIMENSION(:) :: roan
  REAL(sdk), POINTER, DIMENSION(:) :: roln
  REAL(sdk), POINTER, DIMENSION(:) :: rovn
  REAL(sdk), POINTER, DIMENSION(:) :: sn
  REAL(sdk), POINTER, DIMENSION(:) :: tdn
  REAL(sdk), POINTER, DIMENSION(:) :: tln
  REAL(sdk), POINTER, DIMENSION(:) :: tvn
  REAL(sdk), POINTER, DIMENSION(:) :: vmn
  REAL(sdk), POINTER, DIMENSION(:,:) :: twn
  REAL(sdk), POINTER, DIMENSION(:) :: vln
  REAL(sdk), POINTER, DIMENSION(:) :: vvn
  REAL(sdk), POINTER, DIMENSION(:) :: bitn
  REAL(sdk), POINTER, DIMENSION(:) :: cifn
  REAL(sdk), POINTER, DIMENSION(:) :: hil
  REAL(sdk), POINTER, DIMENSION(:) :: hiv
  REAL(sdk), POINTER, DIMENSION(:) :: hig
  REAL(sdk), POINTER, DIMENSION(:) :: qppc
  REAL(sdk), POINTER, DIMENSION(:) :: vlt
  REAL(sdk), POINTER, DIMENSION(:) :: vvt
  REAL(sdk), POINTER, DIMENSION(:) :: twan
  REAL(sdk), POINTER, DIMENSION(:) :: twen
  REAL(sdk), POINTER, DIMENSION(:) :: tcen
END TYPE g1DArrayT
!
TYPE (g1DArrayT),DIMENSION(maxComps) :: g1DAr
```

DERIVED TYPE VARIABLES:

array1DPtrT	Pointers to the generic 1D arrays.
arrayNodeT	Array (comp index) of pointers to the generic 1D arrays.
array1DPtrs	Array (comp index) of pointers to the generic 1D arrays.
faceArs	Generic 1D face-centered component arrays.
g1DAr	Generic 1D cell-centered component arrays.

REAL VARIABLES:

!Elements of DERIVED TYPE array1DPtrT

array1DPtr

!Elements of DERIVED TYPE g1DArrayT

driv	Storage for thermodynamic derivatives and enthalpies that are individually stored in the following 18 arrays. See Section 3.2.2.b.4 (subsection on the 1D/3D common data interface) for a description of the use of array driv.
dtsdp	Derivative of tsat wrt pressure.
deldp	Derivative of el wrt pressure.
devdp	Derivative of vapor int. energy wrt pressure.
deldt	Derivative of el wrt liquid temperature.
devdt	Derivative of vapor internal energy wrt vapor temperature.
drolp	Derivative of rol wrt pressure.
drovp	Derivative of vapor density wrt pressure.
drolt	Derivative of liquid density wrt liquid temperature.
drovt	Derivative of vapor density wrt vapor temperature.
hvst	Vapor enthalpy at tssn and p – pa in (j/kg).
hlst	Liquid enthalpy at tssn and p in (j/kg).
dhvsp	Derivative of vapor enthalpy wrt p – pa.
dhlsp	Derivative of liquid enthalpy wrt p.
dtssp	Derivative of tssn wrt p – pa.
devat	Derivative of eva wrt vapor temperature.
devap	Derivative of eva wrt pa.
drvap	Derivative of rova wrt pa.
drvat	Derivative of rova wrt vapor temperature.

! Hydro variables

cfz	Variable not currently implemented.
dfldp	Derivative of liquid velocity with respect to pressure.
dfvdp	Derivative of gas velocity with respect to pressure.
cl	Liquid thermal conductivity.
cpl	Liquid specific heat at constant pressure.

cpv	Gas specific heat at constant pressure.
cv	Gas thermal conductivity.
dx	Cell length.
fa	Cell-edge (interface) flow area.
fric	Additive friction factor.
grav	Gravitation term (cosine θ).
alpo	Gas-volume fraction at the start of the previous step (α^{n-1}).
am	Noncondensable-gas mass.
h1	Weighting factor for the stratified flow regime.
h2	Stratified interfacial area.
h3	Gravitational head caused by gas-volume-fraction gradient.
arc	Density of solute in cell, $c(1-\alpha)\rho_\ell$. `isolut` = 0 or 1.
hd	Hydraulic diameter.
hdht	Heat-transfer hydraulic diameter.
hfg	Latent heat of vaporization.
qppp	QPPP spatial-distribution factor applied to the wall heat source.
rarl	Variable not currently implemented.
rarv	Variable not currently implemented.
rhs	Implicit vs explicit weighting factor, g'.
rmem	Variable not currently implemented.
rmvm	Mixture density times mixture velocity.
rom	Mixture density.
rvmf	Gas mass flow.
sig	Surface tension.
trid	Storage for stabilizer linear system.
tsat	Saturation temperature at total pressure.
tssn	Saturation temperature at steam (vapor) pressure.
visl	Liquid viscosity.
visv	Gas viscosity.
vol	Cell volume.
vrv	Cell-averaged relative velocity.
vr	Relative (gas–liquid) velocity.
wa	Wall area.
vlx	Variable not currently implemented.
vvx	Variable not currently implemented.
wfl	Wall-friction factor for the liquid.

wfv	Wall-friction factor for the gas.
wfmfl	Wall-friction multiplier factor for the liquid.
wfmfv	Wall-friction multiplier factor for the gas.
favol	Cell flow area used in choked-flow model.
vvvol	Choked-flow, model-cell gas velocity.
vlvol	Choked-flow, model-cell liquid velocity.
grvol	Cell-averaged gravitation term.
alpmn	Minimum value of the gas-volume fraction among a cell and all of its neighbors.
alpmx	Maximum value of the gas-volume fraction among a cell and all of its neighbors.
vlvc	Liquid velocity at a neighboring cell edge where the donor-celled liquid fraction is maximum.
fsmlt	Interphasic-area multiplier during condensation.
vlalp	Liquid mass flux that enters the cell from the cell edges located above the cell.
elev	Cell-centered elevations (used only if `ielv = 1` in namelist input).
qp3f	QPPP spatial-distribution factor applied to the wall heat source.
dalva	Variable not currently implemented.
hgam	Energy contribution to phase change from subcooled boiling.
hla	Sum of all products of liquid HTC with heat-transfer area.
hva	Sum of all products of gas HTC with heat-transfer area.
hlatw	Similar to `hla`, except that the product includes wall temperature.
hvatw	Similar to `hva`, except that the product includes wall temperature.
finan	Inverted annular regime factor.
wat	Total heat-transfer area.
regnm	Flow-regime number.
xsm	Special-purpose, DOE-model parameter.
ysm	Special-purpose, DOE-model parameter.
zsm	Special-purpose, DOE-model parameter.
r0sm	Special-purpose, DOE-model parameter.
rsm	Special-purpose, DOE-model parameter.
nfvsm	Special-purpose, DOE-model parameter.
nf1sm	Special-purpose, DOE-model parameter.
nf2sm	Special-purpose, DOE-model parameter.

nf3sm	Special-purpose, DOE-model parameter.
nfv4sm	Special-purpose, DOE-model parameter.
nflsm	Special-purpose, DOE-model parameter.
nfl4sm	Special-purpose, DOE-model parameter.
uvsm	Special-purpose, DOE-model parameter.
nfcvsm	Special-purpose, DOE-model parameter.
nfclsm	Special-purpose, DOE-model parameter.
vvsm	Special-purpose, DOE-model parameter.
vlsm	Special-purpose, DOE-model parameter.
qrl	Radiation heat flux to the liquid.
qrv	Radiation heat flux to the gas.
faLiq	
faVap	
faLiqFrac	
faVapFrac	
faArl	
faArv	
faArel	
faArev	
faAra	
faArc	
faWvM	
faWvP	
faWlM	
faWlP	

! Dual timestep variables.

alp	Old gas-volume fraction.
alpd	Variable not currently implemented.
alv	Old value of the flashing interfacial HTC times interfacial area.
alve	Old value of the liquid-side interfacial HTC times interfacial area.
ara	Old stabilizer value for $\alpha\rho_a$.
arel	Old stabilizer value for $(1-\alpha)\rho_\ell e_\ell$.
arev	Old stabilizer value for $\alpha\rho_v e_v$.
arl	Old stabilizer value for $(1-\alpha)\rho_\ell$.
arv	Old stabilizer value for $\alpha\rho_v$.

chti	Old value of the vapor-side interfacial HTC times the interfacial area.
chtia	Old value of the noncondensable-gas interfacial HTC times the interfacial area.
conc	Old solute-mass-to-liquid-mass ratio, where `isolut` = 0 or 1.
ea	Old noncondensable-gas internal energy.
el	Old liquid internal energy.
ev	Old gas internal energy.
gam	Old vapor generation rate per unit volume.
p	Old total pressure.
pa	Old noncondensable-gas partial pressure.
roa	Old noncondensable-gas density.
rol	Old liquid density.
rov	Old gas density.
s	Old solute mass plated on structure surface. `isolut` = 0 or 1.
td	Variable not currently implemented.
tl	Old liquid temperature.
tv	Old gas temperature.
vm	Old mixture velocity.
tw	Old wall temperature.
vl	Old liquid velocity.
vv	Old gas velocity.
bit	Bit flags from the previous timestep.
cif	Old interfacial-drag coefficients.
hilo	Old HTC between inside wall and liquid.
hivo	Old HTC between inside wall and gas.
higo	Old HTC between inside wall and gas.
qppco	Old CHF.
vlto	Old stabilizer liquid velocity $\left(\tilde{V}_{\ell}^{n}\right)$.
vvto	Old stabilizer gas velocity $\left(\tilde{V}_{g}^{n}\right)$.
twa	Old absolute total conduction.
twe	Old effective total conduction.
tce	Old total convective energy.
alpn	New gas-volume fraction.
alpdn	Variable not currently implemented.

alvn	New value of the flashing interfacial HTC times interfacial area.
alven	New value of the liquid-side interfacial HTC times interfacial area.
aran	New stabilizer value for $\alpha\rho_a$.
areln	New stabilizer value for $(1-\alpha)\rho_\ell e_\ell$.
arevn	New stabilizer value for $\alpha\rho_v e_v$.
arln	New stabilizer value for $(1-\alpha)\rho_\ell$.
arvn	New stabilizer value for $\alpha\rho_v$.
chtin	New value of the vapor-side interfacial HTC times the interfacial area.
chtan	New value of the noncondensable-gas interfacial HTC times the interfacial area.
concn	New solute-mass-to-liquid-mass ratio, where `isolut` = 0 or 1.
ean	New noncondensable-gas internal energy.
eln	New liquid internal energy.
evn	New gas internal energy.
gamn	New vapor generation rate per unit volume.
pn	New total pressure.
pan	New noncondensable-gas partial pressure.
roan	New noncondensable-gas density.
roln	New liquid density.
rovn	New gas density.
sn	New solute mass plated on structure surface. `isolut` = 0 or 1.
tdn	Variable not currently implemented.
tln	New liquid temperature.
tvn	New gas temperature.
vmn	New mixture velocity.
twn	New wall temperature.
vln	New liquid velocity.
vvn	New gas velocity.
bitn	Bit flags for the current timestep.
cifn	New interfacial-drag coefficients.
hil	New HTC between inside wall and liquid.
hiv	New HTC between inside wall and gas.
hig	New HTC between inside wall and gas.
qppc	New CHF.

vlt	New stabilizer liquid velocity $\left(\tilde{V}_e^{n+1}\right)$.
vvt	New stabilizer gas velocity $\left(\tilde{V}_g^{n+1}\right)$.
twan	New absolute total conduction.
twen	New effective total conduction.
tcen	New total convective energy.

LOGICAL VARIABLES:

!Elements of DERIVED TYPE array1DPtrT	
isAssociated	Logical flag indicating pointer-association status.

INTEGER VARIABLES:

The following 80 index variables (`xxxInd`) are used as arguments to the `Gen1DArray` data interface:

hgamInd	Pointer to `hgam` in the generic 1D array.
hlaInd	Pointer to `hla` in the generic 1D array.
hvaInd	Pointer to `hva` in the generic 1D array.
hlatwInd	Pointer to `hlatw` in the generic 1D array.
hvatwInd	Pointer to `hvatw` in the generic 1D array.
watInd	Pointer to `wat` in the generic 1D array.
finanInd	Pointer to `finan` in the generic 1D array.
alpInd	Pointer to `alp` in the generic 1D array.
alvInd	Pointer to `alv` in the generic 1D array.
cplInd	Pointer to `cpl` in the generic 1D array.
cpvInd	Pointer to `cpv` in the generic 1D array.
clInd	Pointer to `cl` in the generic 1D array.
cvInd	Pointer to `cv` in the generic 1D array.
hdhtInd	Pointer to `hdht` in the generic 1D array.
chtiInd	Pointer to `chti` in the generic 1D array.
pInd	Pointer to `p` in the generic 1D array.
rovInd	Pointer to `rov` in the generic 1D array.
rolInd	Pointer to `rol` in the generic 1D array.
volInd	Pointer to `vol` in the generic 1D array.
hfgInd	Pointer to `hfg` in the generic 1D array.
sigInd	Pointer to `sig` in the generic 1D array.
tlInd	Pointer `totl` in the generic 1D array.
tlnInd	Pointer to `tln` in the generic 1D array.
tsatInd	Pointer to `tsat` in the generic 1D array.

tvInd	Pointer to `tv` in the generic 1D array.
tvnInd	Pointer to `tvn` in the generic 1D array.
vislInd	Pointer to `visl` in the generic 1D array.
visvInd	Pointer to `visv` in the generic 1D array.
vvInd	Pointer to `vv` in the generic 1D array.
vlInd	Pointer to `vv` in the generic 1D array.
vmInd	Pointer to `vm` in the generic 1D array.
evInd	Pointer to `ev` in the generic 1D array.
elInd	Pointer to `el` in the generic 1D array.
eaInd	Pointer to `ea` in the generic 1D array.
roaInd	Pointer to `roa` in the generic 1D array.
paInd	Pointer to `pa` in the generic 1D array.
tssnInd	Pointer to `tssn` in the generic 1D array.
concInd	Pointer to `conc` in the generic 1D array.
sInd	Pointer to `s` in the generic 1D array.
rmvmInd	Pointer to `rmvm` in the generic 1D array.
vmnInd	Pointer to `vmn` in the generic 1D array.
cifnInd	Pointer to `cifn` in the generic 1D array.
snInd	Pointer to `sn` in the generic 1D array.
gamnInd	Pointer to `gamn` in the generic 1D array.
alpnInd	Pointer to `alpn` in the generic 1D array.
hivInd	Pointer to `hiv` in the generic 1D array.
hilInd	Pointer to `hil` in the generic 1D array.
panInd	Pointer to `pan` in the generic 1D array.
eanInd	Pointer to `ean` in the generic 1D array.
elnInd	Pointer to `eln` in the generic 1D array.
evnInd	Pointer to `evn` in the generic 1D array.
rolnInd	Pointer to `roln` in the generic 1D array.
rovnInd	Pointer to `rovn` in the generic 1D array.
romInd	Pointer to `rom` in the generic 1D array.
roanInd	Pointer to `roan` in the generic 1D array.
amInd	Pointer to `am` in the generic 1D array.
hdInd	Pointer to `hd` in the generic 1D array.
concnInd	Pointer to `conc` in the generic 1D array.
vlnInd	Pointer to `vln` in the generic 1D array.
vvnInd	Pointer to `vvn` in the generic 1D array.
rvmfInd	Pointer to `rvm` in the generic 1D array.

pnInd	Pointer to `pn` in the generic 1D array.
dxInd	Pointer to `dx` in the generic 1D array.
chtinInd	Pointer to `chtin` in the generic 1D array.
alvnInd	Pointer to `alvn` in the generic 1D array.
faInd	Pointer to `fa` in the generic 1D array.
faLiqFracInd	Pointer to `faLiqFrac` in the generic 1D array.
faVapFracInd	Pointer to `faVapFrac` in the generic 1D array.
faArvInd	Pointer to `faArv` in the generic 1D array.
faArlInd	Pointer to `faArl` in the generic 1D array.
faArevInd	Pointer to `faArev` in the generic 1D array.
faArelInd	Pointer to `faArel` in the generic 1D array.
faAraInd	Pointer to `faAra` in the generic 1D array.
faArcInd	Pointer to `faArc` in the generic 1D array.
faWlPInd	Pointer to `faWlP` in the generic 1D array.
faWlMInd	Pointer to `faWlM` in the generic 1D array.
faWvPInd	Pointer to `faWvP` in the generic 1D array.
faWvMInd	Pointer to `faWvM` in the generic 1D array.
faLiqInd	Pointer to `faLiq` in the generic 1D array.
faVapInd	Pointer to `faVap` in the generic 1D array.
num1DFaceArrays	Number of variables available to the `Gen1DArray` data interface.

C.19. Module `Global`

```
REAL(sdk), POINTER, DIMENSION(:) :: RunTitle
REAL(sdk), DIMENSION(2047*2) :: Buffer
INTEGER(sik) ifreeIG,ifreeAG
INTEGER(sik) igSize
PARAMETER (igSize=10000)
INTEGER(sik) ig(igSize)
REAL(sdk)  ag(igSize)
INTEGER(sik) lentitle
INTEGER(sik) cci
INTEGER(sik) cco
INTEGER(sik) h1Ind,r1Ind,rh1Ind
INTEGER(sik) compIndices(maxComps)
DATA h1Ind,r1Ind,rh1Ind/0,0,0/
```

REAL VARIABLES:

RunTitle	Title for the TRAC run.
Buffer	Container array for the VESSEL component.
ag(igSize)	

INTEGER VARIABLES:

ifreeIG	
ifreeAG	
igSize	Parameter used to dimension `ig` and `ia`.
ig(igSize)	
lentitle	Length of the title.
cci	Component indices as input.
cco	Component indices as reordered in subroutine `asign`.
h1Ind	Component index of first heat structure in `tracin`.
r1Ind	Component index of first restart component.
rh1Ind	Component index of first heat structure in restart file.
compIndices	
vessPtrSize	Size of the VESSEL pointer array.
vessPtrTable	VESSEL pointer table.

C.20. Module `GlobalDat`

```
REAL (sdk) ::  recipDt
INTEGER(sik) :: iSegment, iMat3D
LOGICAL :: newSolver = .FALSE.
REAL(sdk)         cpuflg,dammc,damx,delt,deltht,difmin,dprmx,dtlmx      &
  &,dtmax,dtmin,dto,dtrat,dtrmx,dtsmx,dtvmx,encmax,eps1,eps2,epso    &
  &,ercemx,epss,etime,htlosi,htloso,odelt,powerc,pssmn,pssmx,rfat    &
  &,rvmax,tend,tercmx,timec,timet,tmmax,varer,vcmn,vcmx,vmaxo,vmaxt  &
  &,vmaxt3,vmcon,vmnew,vmold,vmxt3o,xtable,x0sm,y0sm,z0sm,omsasm     &
  &,wsasm,wdsasm,t0sm(3),xvsm,yvsm,zvsm,rmatsm(3,3),omsm(3),wsm(3)   &
  &,wdsm(3),errsm,dtsm
INTEGER(sik)      dstep,iadded,iblaus,iccmx,icmp,icmpmx,idiag,  &
  & idiags,ieccpx,ieos,iff3d,ifprep,igeom3,im100,im100x,imfr,invan,  &
  & iofftk,ipak,ipak3d,ipakon,ipkpmp,ireset,irsflg,irstfl,isolut,    &
  & issflg,istdy,isttc,ithd,itmin,itpako,jfat,kccmx,lcmptr,levstg,   &
  & llvflg,ncmn,ncmx,ncrg,ndia1,nencl,newrfd,                        &
  & nfrc1,nfrc3,nitav,nitmn,nitmx,nloopp,nosets,nsend,nseo,nsmn,     &
  & nsmx,nspl,nspu,nsso,nstab,nstp,nvgrav,nvpow,oitmax,sitmax,       &
```

```
   & stdyst,transi,motsm,statsm,saxsm
!
REAL(sdk) dtend,hdump,hedit,hgraf,hsedit
INTEGER(sik) nalt,ndid
!
REAL(sdk) dtstrt
INTEGER(sik) icdelt
!
REAL(sdk) edint,gfint,sedint,tedit,tgraf,tsedit
INTEGER(sik) nsgo
!
DATA edint,gfint,sedint,tedit,tgraf,tsedit/6*0.0d0/
!
INTEGER(sik) noair,nsmec,nthrmc,nvtc
!
DATA noair/1/
DATA nsmec/0/
DATA nthrmc,nvtc/0,0/
!
INTEGER(sik) ipreit,itdon,jdonp,ncomdp
LOGICAL lbckv,lreit,lreitv
!
DATA itdon/4/,ipreit/0/
!
REAL(sdk) bkpall,bkpalu,bkpsta,bkpstp,bkpstt
INTEGER(sik) ibkpst,jbkpst
LOGICAL lbkpst,lbkcyl
!
DATA bkpalu/0.03d0/,bkpall/1.0d-10/,bkpsta/0.5d0/,bkpstp/2.5d0/   &
&,bkpstt/200.0d0/
!
INTEGER(sik) ntsprn
!
DATA ntsprn/0/
!
REAL(sdk) adate,atime,cput,timcpu,timei,timiom,timsys,timtot
INTEGER(sik) nstept
```

```
!
INTEGER(sik) iftp,itfl1,nfl1,nfl3
!
DATA iftp,itfl1,nfl1,nfl3/0,0,0,0/
!
REAL(sdk) varerm,verr
INTEGER(sik) iott,nstep,oitno
!
REAL(sdk) csf,csf1d,csf3d,csf3dl,csf3dh,fncif
INTEGER(sik) ivmn,ivmx,jiv,noldv

DATA csf1d/1000.0d0/
DATA csf,csf3dl,csf3dh,fncif/1.0d0,1.0d0,1000.0d0,0.7/
DATA ivmn,ivmx,jiv,noldv/120,140,0,0/
DATA nstep/0/
DATA oitno,varerm/0,0.0d0/
!
REAL(sdk) fmax(7)
REAL(sdk) cf,eps,epspow,fflw,maxfln,rpcf,rtwfp,stime,tpowr
INTEGER(sik) lok(7,2)
INTEGER(sik) ipovel,ipowr,isscvt,ncores,nef,net,nopow
```

REAL VARIABLES:

recipDt	
cpuflg	Option for eliminating the CPU time from being output to files TRCMSG and TRCOUT and the terminal so that a DIFF file comparison between TRAC-P versions will not include the CPU-time differences between calculations (NAMELIST variable).
dammc	Maximum gas-volume-fraction change during the timestep (not used).
damx	Error caused by the relative change in the gas-volume fraction (not used).
delt	Current timestep size for advancement in time of the finite-difference-equation solution.
deltht	Heat-transfer timestep size.
difmin	Minimum diffusion number required for stability of the ROD or SLAB conduction solution.
dprmx	Maximum pressure change during the timestep.

dtlmx	Maximum liquid-temperature change during the timestep.
dtmax	Maximum allowable timestep size for the time interval.
dtmin	Minimum allowable timestep size for the time interval.
dto	Previous timestep size.
dtrat	Ratio of the previous timestep size to the reduced timestep size that results in a trip (assigned special timestep data) crossing its setpoint at the end of the timestep.
dtrmx	Maximum ROD or SLAB temperature change during the timestep.
dtsmx	Maximum metal-temperature change during the timestep.
dtvmx	Maximum gas-temperature change during the timestep.
encmax	Worst-case, convection-power difference from a timestep.
eps1	The lower-bound criterion for increasing the Kaganove-method integration timestep for solving the point-reactor kinetics equations.
eps2	The upper-bound criterion for decreasing the Kaganove-method integration timestep for solving the point-reactor kinetics equations.
epso	Convergence criterion for the outer iteration.
ercemx	Worst-case, convection-power difference during a calculation.
epss	Convergence criterion for the steady-state calculation.
etime	Current calculation time.
htlosi	Wall inner-surface heat loss by 1D components only (total system heat loss from the fluid to the wall inner surface for 1D hydraulic components only).
htloso	Wall outer-surface heat loss by 1D components only (total system heat loss from the wall outer surface to the exterior surroundings for 1D hydraulic components only).
odelt	Previous timestep size.
powerc	Maximum convection-power difference between what goes into the fluid and what comes from the wall in convection heat-transfer from HTSTRs.
pssmn	Minimum steam-generator, secondary-side pressure.
pssmx	Maximum steam-generator, secondary-side pressure.
rfat	Maximum ratio of the interface flow area to the adjacent-mesh-cell average flow area.
rvmax	Maximum ratio of the adjacent-mesh-cell average flow areas when their interface does not have an additive loss coefficient specified.
tend	End time for the timestep data domain.

tercmx	Time at which the worst-case power difference occurred during a calculation.
timec	Clock time in seconds.
timet	Current calculation time.
tmmax	Time at which the worst-case, convection-power difference occurred during a calculation.
varer	Variable error.
vcmn	Final convergence for component ncmn at step nsmn.
vcmx	Final convergence for component ncmx at step nsmx.
vmaxo	1D-component maximum ratio of the Courant number to the timestep size at the beginning of the previous timestep.
vmaxt	1D-component maximum ratio of the Courant number to the timestep size at the beginning of the present timestep.
vmaxt3	3D VESSEL component maximum ratio of the Courant number to the timestep size at the beginning of the present timestep.
vmcon	Net water mass (liquid plus vapor) convected into VESSEL component(s) during the time interval $t^{n+1} - t^n$.
vmnew	VESSEL water mass (liquid plus vapor) at t^{n+1}.
vmold	VESSEL water mass (liquid plus vapor) at t^n.
vmxt3o	3D-VESSEL-component maximum ratio of the Courant number to the timestep size at the beginning of the previous timestep.
xtable	Abscissa-coordinate value from the last axial-power-shape table evaluation.
x0sm	Special-purpose, DOE-model parameter.
y0sm	Special-purpose, DOE-model parameter.
z0sm	Special-purpose, DOE-model parameter.
omsasm	Special-purpose, DOE-model parameter.
wsasm	Special-purpose, DOE-model parameter.
wdsasm	Special-purpose, DOE-model parameter.
t0sm(3)	Special-purpose, DOE-model parameter.
xvsm	Special-purpose, DOE-model parameter.
yvsm	Special-purpose, DOE-model parameter.
zvsm	Special-purpose, DOE-model parameter.
rmatsm(3,3)	Special-purpose, DOE-model parameter.
omsm(3)	Special-purpose, DOE-model parameter.
wsm(3)	Special-purpose, DOE-model parameter.
wdsm(3)	Special-purpose, DOE-model parameter.
errsm	Special-purpose, DOE-model parameter.
dtsm	Special-purpose, DOE-model parameter.

!

dtend	Time interval during which the special timestep data are used.
hdump	Saved value of the next data-dump-edit time from the regular timestep data when the special timestep data are used.
hedit	Saved value of the next large-edit time from the regular timestep data when the special timestep data are used.
hgraf	Saved value of the next graphics-edit time from the regular timestep data when the special timestep data are used.
hsedit	Saved value of the next small-edit time from the regular timestep data when the special timestep data are used.

!

dtstrt	Timestep that can be set as the initial timestep size for a restart calculation (namelist variable; –1.0 default value).

!

edint	Large-edit interval for the time domain.
gfint	Graphics-edit interval for the time domain.
sedint	Small-edit interval for the time domain.
tedit	Time of next large edit.
tgraf	Time of next graphics edit.
tsedit	Time of next small edit.

!

bkpall	Maximum lower limits on the gas-volume fraction such that a backup is forced if the gas-volume fraction lies within these limits.
bkpalu	Maximum upper limits on the gas-volume fraction such that a backup is forced if the gas-volume fraction lies within these limits.
bkpsta	Gas-volume-fraction variation that is allowed in the POST stage. If the gas-volume-fraction change exceeds `bkpsta`, a backup is forced.
bkpstp	Maximum fractional pressure change that is allowed in the POST stage. If the fractional pressure change exceeds `bkstp`, a backup is forced.
bkpstt	Maximum variation in liquid and gas temperatures that is allowed in the POST stage. If the temperature change exceeds `bkpstt`, a backup is forced.

!

adate	Date obtained from calling system routine `date`.
atime	Time obtained from calling system routine `date`.

cput	Cumulative CPU time from previous jobs in a restarted series of calculations; `cput` is set to 0.0 s at time 0.0 s.
timcpu	CPU time obtained from calling system routine `timing`.
timei	Time limit of the current job obtained from calling system routine GETJTL.
timiom	I/O time obtained from calling system routine `timing`.
timsys	System time obtained from calling system routine `timing`.
timtot	Total of CPU, I/O, and system times obtained from calling system routine `timing`.
!	
varerm	Maximum variable error.
verr	Velocity error at component junction.
!	
csf	A factor (1.0) applied to `csf1d` and `csf3d` to define the maximum material Courant number.
csf1d	Maximum material Courant number (1000.0) for the 1D hydraulic components.
csf3d	Maximum material Courant number (1000.0) for the 3D VESSEL components.
csf3dl	Maximum material Courant number (1.0) for the 3D VESSEL components when the SET3D equations are not evaluated.
csf3dh	Maximum material Courant number (1000.0) for the 3D VESSEL components when the SET3D equations are evaluated.
fncif	Constant 0.7 (variable not used).
!	
cf	Fluid mass flow through the reactor-core region.
eps	Tolerance on calculation time for editing and terminating the problem.
epspow	Convergence criterion on the fractional change in liquid velocity per second for turning on the steady-state power when all reactor-core inlet interfaces satisfy this criterion.
fflw	Fraction of the steady-state power level defined by the coolant mass flow through the reactor core times `rpcf`.
fmax	Array of maximum normalized errors.
maxfln	Maximum 1D mass flow at this steady-state convergence test.
rpcf	Ratio of reactor-core power to coolant mass flow based on the difference in internal energies from the core inlet and outlet temperatures that are input.
rtwfp	Ratio of heat-transfer to fluid-dynamics timestep sizes.
stime	Steady-state calculation time.

tpowr — Steady-state calculation time when the reactor-core power is set on.

(1) INTEGER VARIABLES:

iSegment

iMat3D

dstep — Timestep number of the data dump to be used for the restart calculation.

iadded — Number-of-timesteps interval for printing calculation summary to the terminal and TRCMSG file (0 suppresses this printout).

iblaus — Option to apply the Blasius interfacial-drag correlation in the downcomer and lower PLENUM of the VESSEL components (namelist variable).

iccmx — Component number in the IORDER array having the most severe timestep limit for numerical stability of the calculation.

icmp — Component indicator.

icmpmx — Component number in which the worst-case, convection-power difference occurred during the timestep.

idiag — Option that defines different levels of debugging information of appropriate parameter values (namelist variable).

idiags — Option to select alternate variables to be written in a large edit to the TRCOUT file for 1D hydraulic components when `idiag` > 0 (namelist variable).

ieccpx — Component number in which the worst-case, convection-power difference occurred during the calculation.

ieos — Air-water option

0 = steam water, noncondensable gas, and liquid water in fluid;

1 = noncondensable gas and liquid water (no steam water) in fluid.

iff3d — Outer-iteration, VESSEL-evaluation flag.

0 = evaluate the VESSEL-coefficient matrix equation;

1 = back-substitute the VESSEL matrix-equation solution.

ifprep — Flag that indicates sections of PREPER to be executed (nonzero only for 1D cores).

igeom3 — VESSEL-geometry flag.

0 = flow areas between the downcomer and inside of the VESSEL set to zero (default value);

1 = flow areas between the downcomer and inside of the VESSEL maintained at the user input values.

Note: The vent-valve option overrides the IGEOM3 = 1 option in cells that have vent-valve connections.

im100	Flag that indicates if the backup occurred during previous timestep (used for mass check on logic).
im100x	Flag that indicates whether the previous timestep that failed was obtained from a restart.
imfr	Calculates the azimuthal-θ or y, axial-z, and radial-r or x mass flows for both liquid and gas and outputs them to the TRCGRF graphics file (namelist variable). 1 = outputs no phasic mass flows (default); 3 = outputs 3D VESSEL mass flow.
invan	Option to select either T_{CHF} or T_{sat} for control of the inverted annular-flow regime.
iofftk	Option to select the TEE offtake model. 0 = offtake model off; 1 = offtake model on.
ipak	1D hydraulic-component, water-packing option. 0 = off; 1 = on.
ipak3d	3D VESSEL water-packing option. 0 = off; 1 = on.
ipakon	Flag that indicates if water-packing logic is on during the timestep.
ipkpmp	Flag that indicates if water-packing corrections are made at a pump momentum-source interface. 0 = no (default); 1 = yes.
ireset	Option to reinitialize the energy error to zero at the start of a restart calculation. 0 = no (allow the energy error to accumulate from the previous calculation. 1 = yes.
irsflg	Composite number of the number of signal variables, control blocks, and trips that needs to be read from the TRCRST restart file.
irstfl	Variable not used.
isolut	Solute-tracking option for the entire system model. 0 = off; 1 = on.
issflg	Flag that controls steady-state convergence editing.

istdy — Flag that indicates the type of calculation.

0 = transient;

1 = steady state.

isttc — Static-check flag.

0 = normal mode;

1 = a static-balance check was requested when STDYST = 5 was input.

ithd — Option for inputting heat-transfer diameters for HTSTRs.

0 = no (heat-transfer diameters defined by hydraulic diameters);

1 = yes.

itmin — Minimum stable-film-boiling option.

itpako — Iteration number at which water packing was detected.

jfat — Flow-area-ratio, test-results flag.

0 = flow-area ratios are appropriate;

1 = one or more ratios of the interface flow area to the adjacent mesh-cell, volume-average flow area are invalid.

2 = one or more ratios of the adjacent mesh-cell, volume-average flow areas are invalid.

3 = one or more of both types of flow-area ratios are invalid.

kccmx — Component number of cell that limits stability.

lcmptr — Pointer to end of component data for last component read.

levstg — Gas-volume-fraction averaging flag.

0 = no gas-volume-fraction averaging is performed in HTCOR for steam-generator secondaries (default value);

1 = special gas-volume-fraction averaging is performed in HTCOR for steam-generator secondaries.

llvflg — Switch that determines averaging procedure used in subroutine `htif`.

ncmn — Element number in the `iorder` array for the component that was last to converge at timestep `nsmn`.

ncmx — Element number in the `iorder` array for the component that was last to converge at timestep `nsmx`.

ncrg — Variable (not documented elsewhere and defaulted to zero) that could be used to force the input of `icrrg` (see the VESSEL VLT, Appendix C). Logic is incomplete.

ndia1 — Heat-transfer diameter input option for 1D components (namelist variable).

1 = no heat-transfer diameter input for 1D components (default value);

2 = heat-transfer diameter input for 1D components.

nencl — Total number of radiation enclosures in the radiation heat-transfer model (namelist variable).

newrfd — Option that activates the reflood-model calculation for HTSTR components coupled to VESSEL components when internal test criteria are satisfied (namelist variable).

0 = off;

1 = on.

nfrc1 — Additive-loss-coefficient defining form option for 1D components (namelist variable).

1 = `fric` additive loss coefficents are input for both flow directions (default);

2 = `fric` and `rfric` forward- and reverse-flow additive loss coefficients are input.

nfrc3 — Additive-loss-coefficient defining form option for VESSEL components (namelist variable).

nitav — Average number of outer iterations since the last small edit.

nitmn — Minimum number of outer iterations since the last small edit.

nitmx — Maximum number of outer iterations since the last small edit.

nloopp — Flag to indicate inconsistent source connections of a 1D component loop to different directional cell faces of VESSEL component(s).

nosets — Option for evaluating the sets3d equations for all VESSEL components (namelist variable).

0 = evaluate the `sets3d` equations when the timestep size exceeds 0.8 times the VESSEL Courant-limit timestep size;

1 = do not evaluate the `sets3d` equations;

2 = evaluate the `sets3d` equations every timestep (default).

nsend — End the calculation at this timestep number (namelist variable).

nseo — Timestep number of last completed edit.

nsmn — Last timestep at which `nitmn` outer iterations occurred.

nsmx — Last timestep at which `nitmx` outer iterations occurred.

nspl — Debug print output if `nspl` < `nstep` < `nspu`.

nspu — Debug print output if `nspl` < `nstep` < `nspu`.

nsso — Timestep number of last completed small edit.

nstab — SETS3D-equations evaluation flag for all VESSEL components.

nstp — Number of timesteps since the last edit.

nvgrav — Option to allow the orientation of each VESSEL component to be input specified (namelist variable).

nvpow	Number of VESSEL components in the input and restart data files that are coupled to powered HTSTR components (namelist variable).
oitmax	Maximum number of outer iterations for a transient calculation.
sitmax	Maximum number of outer iterations for a steady-state calculation.
stdyst	Steady-state calculation indicator.
transi	Transient calculation indicator.
motsm	Special-purpose, DOE-model parameter.
statsm	Special-purpose, DOE-model parameter.
saxsm	Special-purpose, DOE-model parameter.
!	
nalt	Constant used to determine if gas-volume-fraction adjustments are needed when the interfacial drag is calculated at a 1D junction connected to a BREAK.
ndid	ID number of the special timestep data that are being used.
!	
icdelt	Option that overrides the evaluation of delt at the beginning of an initial calculation. 0 = `delt` is set to `dtmin`; 1 = `delt` is evaluated.
!	
nsgo	Timestep number of last completed graphics edit.
!	
noair	Variable that turns off noncondensable gas calculations.
nsmec	Variable that turns off stabilizer mass and energy equations.
nthrmc	Variable that turns off (debugs) basic equation set.
nvtc	Variable that turns off stabilizer motion equations.
!	
ipreit	Flag to print messages on forced reiteration.
itdon	If flow reversals occur for `oitno > itdon`, the timestep is backed up.
jdonp	Cell number in `ncomdp`.
ncomdp	Component number of flow reversal, forcing backup.
!	
ibkpst	Component that forces a backup.
jbkpst	Cell number that forces a backup.
!	

ntsprn	Flag for printing extra thermal-hydraulic parameter information to file TRCOUT.
!	
nstept	Cumulative number of timesteps from previous jobs in a restarted series of calculations; `nstept` is set to 0 at time 0.0 s.
!	
iftp	Flag that prevents thermal failure messages if a message has come from `tf1sd3` or `ff3d`.
itfl1	Iteration number of the last `tf1ds3` failure.
nfl1	Total number of `tf1ds3` failures in the current timestep.
nfl3	Total number of `ff3d` failures in the current timestep.
!	
iott	Temporary storage for `iitno`.
nstep	Number of timesteps evaluated during the TRAC-M calculation.
oitno	Outer-iteration number.
!	
ivmn	Minimum timestep number for debug outputing interface `jiv` velocities.
ivmx	Maximum timestep number for debug outputing interface `jiv` velocities.
jiv	Mesh-cell interface number for debug outputing gas and liquid tilde and basic velocities in subroutine `tf1ds1`.
noldv	Flag for setting the beta factor in the momentum-convection term to zero. 0 = no; 1 = yes.
ipovel	Number of reactor-core inlet interfaces that satisfy the `epspow` criterion based on the date-of-change of the liquid velocity.
ipowr	Flag that turns on the steady-state power.
isscvt	Option for evaluating the EPSS steady-state convergence test during a `transi` = 1 transient calculation (namelist variable).
ncores	Total number of reactor-core region inlet interfaces.
nef	Number of timesteps (100) between steady-state convergence check printouts to the terminal and message files.
net	Number of timesteps (5) between steady-state convergence checks.
nopow	Steady-state power flag. 0 = on; 1 = off.

LOGICAL VARIABLES:

newSolver

lbckv	If .TRUE., then variable forces a timestep backup.
lreit	If .TRUE., then variable forces a reiteration.
lreitv	If .TRUE., then variable forces a reiteration.

!

lbkpst	If .TRUE., then a timestep backup is forced from POST.
lbkcyl	If .TRUE., then a timestep backup is forced because heat-transfer energy conservation is not satisified.

C.21. Module `GlobalDim`

```
INTEGER(sik) maxComps
PARAMETER (maxComps=500)
INTEGER(sik) jnvssl,kvel1t,kvel2t,kvel3t,ldim,lendim,lm1dp,     &
& lver,mdim,memflg,moff,ncomp,ncompt,nhtstr,njnmx,njnt,         &
& njun,nloops,nmvssl,npx,nstgj,nthm,numtcr,nvcon,               &
& nvelx,nvely,nvelz
!
INTEGER(sik) lasti
```

INTEGER VARIABLES:

maxComps	Maximum number of components allowed in calculation.
jnvssl	Maximum number of VESSEL junctions in a loop.
kvel1t	Order of the r- or x-direction stabilizer motion-equation VESSEL matrix.
kvel2t	Order of the θ- or y-direction stabilizer motion-equation VESSEL matrix.
kvel3t	Order of the z-direction stabilizer motion-equation VESSEL matrix.
ldim	Maximum size order of the capacitance matrix.
lendim	Variable that dimensions the component VLTs.
lm1dp	Pointer variable for array that stores `m1d` input-data values for HTSTR components.
lver	Location of version information data.
mdim	Maximum order of the banded VESSEL matrix.
memflg	Flag for monitoring dynamic computer-memory expansion.
moff	Array row number of the main diagonal elements from the banded VESSEL matrix.
ncomp	Number of components.

ncompt	Total number of components.
nhtstr	Total number of HTSTR components (namelist variable).
njnmx	Maximum number of network junctions.
njnt	Total number of network junctions for all loops.
njun	Number of junctions.
nloops	Number of 1D loops in the system model.
nmvssl	Number of VESSELs.
npx	Number of pointers in the PTRS COMMON block.
nstgj	Variable not used.
nthm	Number of elements per cell in the DRIV array.
numtcr	Number of title cards.
nvcon	Total number of VESSEL connections.
nvelx	Order of the r- or x-direction stabilizer motion equation VESSEL matrix.
nvely	Order of the θ- or y-direction stabilizer motion equation VESSEL matrix.
nvelz	Order of the z-direction stabilizer motion equation VESSEL matrix.
!	
lasti	Last i index used in VESSEL arrays.

C.22. Module `GlobalPtr`

```
INTEGER(sik) licvs,ldpmax,lijvs,lilcmp,liou,lisvf,livcon,      &
  & livljn,ljout,ljseq,ljun,llcon,lloopn,lmatb,lmcmsh,lmsct,   &
  & lnbr,lnjn,lnsig,lnsigp,lnvcnl,lorder,lptbln,ltitle
INTEGER(sik) lidpcv
INTEGER(sik) lilprb,livlfc,livvto,livlto
INTEGER(sik)   nmat,nvcell
```

INTEGER VARIABLES:

licvs	Pointer for a temporary array that contains a list of all VESSEL composite-cell numbers that have a source connection to one of their cell faces.
ldpmax	
lijvs	Pointer for a temporary array that contains a list of all junction numbers that link to a VESSEL.
lilcmp	Component LCM pointers stored in the order in which the components were read.

liou	Network junction numbers for the junctions of all components, excluding BREAKs and FILLs.
lisvf	Pointer to an array of flags indicating whether a particular component is used to evaluate one or more signal variables (–1, no signal variable; +1, signal variable); this array uses the same order in which the component data are processed.
livcon	Pointer to network junction numbers that connect to a VESSEL.
livljn	ivljn(i) is the VESSEL junction number that corresponds to the network junction number given by ivcon(i).
ljout	Storage area for pointers that locate the beginning of each system loop within data for `iou`.
ljseq	Junction numbers in the order in which junctions occur in the junction-component array.
ljun	Junction-component pair array pointer.
llcon	Number of times each component was the last to converge since the last edit.
lloopn	`ia(lloopn+il–1)` gives the element of the `iorder` array that begins the il^{th} loop pass.
lmatb	Pointer for additional material-property ID numbers.
lmcmsh	Storage for number of coarse-mesh VESSEL source cells or absolute cell index if direct VESSEL solution is used.
lmsct	Temporary storage for VESSEL pressure changes adjacent to sources.
lnbr	Component numbers stored in the order in which components were read.
lnjn	`njn(il)` is the number of network junctions in loop il.
lnsig	`nsig(il)` is the total number of components excluding BREAKs and FILLs in a loop.
lnsigp	`nsigp(il)` is `nsig(il)`.
lnvcnl	`ia(lnvcnl+il–1)` points to the elements of `ivcon` and `ivljn` that begin the il^{th} loop.
lorder	Component numbers stored in the order used for iteration.
lprptb	Pointer to user-defined, material-property tables.
lptbln	Pointer for the number of entry groups in the user-defined, material-property table.
ltitle	Problem title and version information (stored using only the first four bytes of each word).
lidpcv	Pointers to coefficients stored in `dpcv`.
lilprb	Pointer that defines if each hydraulic loop has VESSEL predictor velocities coupled in different directions.

livlfc	Pointer that defines the face-connect number for all junction connections to VESSELs for a given hydraulic loop.
livvto	Pointer that defines the gas tilde velocity at a source-connection junction to a VESSEL for a hydraulic loop.
livlto	Pointer that defines the liquid tilde velocity at a source-connection junction to a VESSEL for a hydraulic loop.
nmat	Number of additional material-property tables provided by the user through input.
nvcell	Total number of cells in all VESSELs.

C.23. Module `HSArray`

```
INTEGER(sik),PARAMETER :: sdkx=sdk
INTEGER(sik),PARAMETER :: sikx=sik
!
TYPE hsArrayT
    REAL(sdk), POINTER, DIMENSION(:)   :: rdpwr
    REAL(sdk), POINTER, DIMENSION(:)   :: rs
    REAL(sdk), POINTER, DIMENSION(:)   :: cpowr
    REAL(sdk), POINTER, DIMENSION(:)   :: hs
    REAL(sdk), POINTER, DIMENSION(:)   :: zpwzt
    REAL(sdk), POINTER, DIMENSION(:)   :: rpwrt
    REAL(sdk), POINTER, DIMENSION(:)   :: zpwtb
    REAL(sdk), POINTER, DIMENSION(:)   :: zpwrf
    REAL(sdk), POINTER, DIMENSION(:)   :: zpw
    REAL(sdk), POINTER, DIMENSION(:,:) :: zpwf
    REAL(sdk), POINTER, DIMENSION(:)   :: zpwfb
    REAL(sdk), POINTER, DIMENSION(:)   :: zs
    REAL(sdk), POINTER, DIMENSION(:)   :: nrdx
    REAL(sdk), POINTER, DIMENSION(:)   :: rdz
!
    REAL(sdk), POINTER, DIMENSION(:,:) :: lchci
    REAL(sdk), POINTER, DIMENSION(:,:) :: lchco
    REAL(sdk), POINTER, DIMENSION(:)   :: hceli
    REAL(sdk), POINTER, DIMENSION(:)   :: hcelo
    REAL(sdk), POINTER, DIMENSION(:)   :: hcomi
    REAL(sdk), POINTER, DIMENSION(:)   :: hcomo
    REAL(sdk), POINTER, DIMENSION(:)   :: htmli
```

```
      REAL(sdk), POINTER, DIMENSION(:)   :: htmlo
      REAL(sdk), POINTER, DIMENSION(:)   :: htmvi
      REAL(sdk), POINTER, DIMENSION(:)   :: htmvo
      REAL(sdk), POINTER, DIMENSION(:)   :: gravr
      REAL(sdk), POINTER, DIMENSION(:)   :: powli
      REAL(sdk), POINTER, DIMENSION(:)   :: powlo
      REAL(sdk), POINTER, DIMENSION(:)   :: powvi
      REAL(sdk), POINTER, DIMENSION(:)   :: powvo
!
      REAL(sdk), POINTER, DIMENSION(:)   :: rpkf
      REAL(sdk), POINTER, DIMENSION(:)   :: idrod
      INTEGER(sik), POINTER, DIMENSION(:) :: ntsxx
      REAL(sdk), POINTER, DIMENSION(:)   :: radrd
      REAL(sdk), POINTER, DIMENSION(:)   :: tc
      REAL(sdk), POINTER, DIMENSION(:)   :: matrd
      REAL(sdk), POINTER, DIMENSION(:)   :: rpwtb
      REAL(sdk), POINTER, DIMENSION(:)   :: rpwrf
      REAL(sdk), POINTER, DIMENSION(:)   :: rctf
      REAL(sdk), POINTER, DIMENSION(:)   :: rctc
      REAL(sdk), POINTER, DIMENSION(:)   :: rcal
      REAL(sdk), POINTER, DIMENSION(:)   :: rcbm
!
      REAL(sdk), POINTER, DIMENSION(:)   :: srp
      REAL(sdk), POINTER, DIMENSION(:)   :: rcn
      REAL(sdk), POINTER, DIMENSION(:)   :: xn
      REAL(sdk), POINTER, DIMENSION(:)   :: xo
      REAL(sdk), POINTER, DIMENSION(:)   :: beta
      REAL(sdk), POINTER, DIMENSION(:)   :: lamda
      REAL(sdk), POINTER, DIMENSION(:)   :: edh
      REAL(sdk), POINTER, DIMENSION(:)   :: lamdh
      REAL(sdk), POINTER, DIMENSION(:)   :: nfax
      REAL(sdk), POINTER, DIMENSION(:)   :: fpuo2
      REAL(sdk), POINTER, DIMENSION(:)   :: ftd
      REAL(sdk), POINTER, DIMENSION(:)   :: gmix
      REAL(sdk), POINTER, DIMENSION(:)   :: gmles
      REAL(sdk), POINTER, DIMENSION(:)   :: pgapt
      REAL(sdk), POINTER, DIMENSION(:)   :: plvol
```

```
      REAL(sdk), POINTER, DIMENSION(:)   :: pslen
!
!     Time-Dependent Data
      REAL(sdk), POINTER, DIMENSION(:)   :: cdg
      REAL(sdk), POINTER, DIMENSION(:)   :: cdh
      REAL(sdk), POINTER, DIMENSION(:)   :: clen
      REAL(sdk), POINTER, DIMENSION(:)   :: cdgn
      REAL(sdk), POINTER, DIMENSION(:)   :: cdhn
      REAL(sdk), POINTER, DIMENSION(:)   :: clenn
!
!     Rod and slab dependent data
      REAL(sdk), POINTER, DIMENSION(:,:)   :: burn
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: cnd
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: cndr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: cpnd
      REAL(sdk), POINTER, DIMENSION(:,:)   :: zht
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: emis
      REAL(sdk), POINTER, DIMENSION(:,:)   :: hgap
      REAL(sdk), POINTER, DIMENSION(:,:)   :: pgap
      REAL(sdk), POINTER, DIMENSION(:,:)   :: pint
      REAL(sdk), POINTER, DIMENSION(:,:)   :: pldv
      REAL(sdk), POINTER, DIMENSION(:,:)   :: qwrx
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: rnd
      REAL(sdk), POINTER, DIMENSION(:,:)   :: idht
      INTEGER(sik), POINTER, DIMENSION(:)  :: noht
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: cpdr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: rndr
      REAL(sdk), POINTER, DIMENSION(:,:)   :: rpowf
!
!     Time dependent rod data
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: radr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: radrn
      REAL(sdk), POINTER, DIMENSION(:,:)   :: drz
      REAL(sdk), POINTER, DIMENSION(:,:)   :: drzn
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: rft
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: rftn
!
```

```
!   Surface dependent rod data
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: alpr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: alvr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: clr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: cvr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: cplr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: cpvr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: concr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hdr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hfgr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: chtir
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: ihtf
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: pr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: rolr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: rovr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: romr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: volr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: stnu
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tld
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: sr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: sigr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tchfr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tchff
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hgamr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hlar
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hvar
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hlatr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hvatr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: finar
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: watr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tlr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tvr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tlnr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tvnr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tsatr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: vislr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: visvr
```

```
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlzr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvzr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvcr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: vmzr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: elr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: evr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: drvdt
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: drldt
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: ear
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: roar
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: par
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: tssnr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hlsr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hvsr
!
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrfg
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrfgo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrfl
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrflo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrfv
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrfvo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrlg
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrlgo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrll
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrllo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrlv
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hrlvo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hqrad
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: hqrdo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: qchff
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: qchfo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: qchfr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: qchro
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdhlr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdhlo
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdhvr
      REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdhvo
```

```
!
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tcefn
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tcefo
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: twaen
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: twaeo
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tween
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: tweeo
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: cepwn
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: cepwo
!
END TYPE hsArrayT
!
TYPE (hsArrayT),TARGET,DIMENSION(maxComps) :: hsAr
TYPE (hsArrayT),POINTER:: chs
```

DERIVED TYPE VARIABLES:

hsAr	HTSTR component-specific arrays.
chs	Pointer table for the HTSTR component-specific arrays.

REAL VARIABLES:

!Elements of DERIVED TYPE hsArrayT

rdpwr	ROD or SLAB relative radial or thickness power density.
rs	Relative ROD-radial or SLAB-thickness, power-density distribution used to average reactivity feedback parameters over the reactor-core volume.
cpowr	Relative power per average rod.
hs	Pointer variable for the horizontal-plane, shape-weight function used.
zpwzt	Axial locations where the axial-power-shape relative power densities are defined.
rpwrt	ROD-radial or SLAB-thickness locations where the power shape's relative power densities are defined.
zpwtb	Relative power density axial-power-shape table.
zpwrf	Axial-power-shape, rate-factor table.
zpw	Last interpolated axial power.
zpwf	2D axial-r or axial-x power shape after numerical integration over the node and hydro-cell lengths.
zpwfb	Subroutine `zpwhci` evaluated axial-power shape at `ncrz+1` nodes based on the input axial-power shape defined at `nzpwz` node locations.

zs	Relative axial-power-shape, power-density distribution used to volume average the reactivity-feedback parameters over the reactor-core volume.
nrdx	Number of actual rods or slabs modeled by the calculational ROD or SLAB element.
rdz	Axial node positions.
!	
lchci	The hydro-cell parameters for heat-transfer coupling to the HTSTR inner surface.
lchco	The hydro-cell parameters for heat-transfer coupling to the HTSTR outer surface.
hceli	Cell number coupled to the HTSTR nodes at the inner surface.
hcelo	Cell number coupled to the HTSTR nodes at the outer surface.
hcomi	Component number of the hydro cell coupled to the HTSTR inner surface.
hcomo	Component number of the hydro cell coupled to the HTSTR outer surface.
htmli	Liquid-phase wall heat-transfer multiplier factor for the inner surface.
htmlo	Liquid-phase wall heat-transfer multiplier factor for the outer surface.
htmvi	Gas-phase wall heat-transfer multiplier factor for the inner surface.
htmvo	Gas-phase wall heat-transfer multiplier factor for the outer surface.
gravr	Cosine of the angle between a vector pointing upward and a vector from the lower-to-higher numbered axial cells.
powli	Total power across the HTSTR inner surface to the liquid.
powlo	Total power across the HTSTR outer surface to the liquid.
powvi	Total power across the HTSTR inner surface to the gas.
powvo	Total power across the HTSTR outer surface to the gas.
!	
rpkf	Supplemental rod power-peaking factor.
idrod	Cell-coupling identifier for rods.
radrd	Rod node radii (cold).
tc	Thermocouple-model input parameters.
matrd	ROD or SLAB material identification numbers.
rpwtb	Power or reactivity table.
rpwrf	Rate-factor table for the power or reactivity table.

rctf	Fuel temperature reactivity-coefficient table. The symbol π_i indicates the product of the following variable taken over the i subscript.
rctc	Coolant temperature reactivity-coefficient table. The symbol π_i indicates the product of the following variable taken over the i subscript.
rcal	Gas volume fraction reactivity-coefficient table. The symbol π_i indicates the product of the following variable taken over the i subscript.
rcbm	Boron reactivity-coefficient table. The symbol π_i indicates the product of the following variable taken over the i subscript.
!	
rcn	Reactivity-coefficient values at the beginning of the previous timestep.
srp	Summed programmed and feed-back reactivity changes.
xn	New reactivity-feedback parameter values.
xo	Old reactivity-feedback parameter values.
beta	Delayed-neutron group fraction.
lamda	Decay constant of delayed-neutron groups.
edh	Energy-yield fraction of decay-heat group.
lamdh	Decay constant of decay-heat groups.
nfax	Rod fine-mesh noding factor.
fpuo2	Fraction of plutonium oxide in mixed-oxide fuel fraction.
ftd	Fuel density (fraction of theoretical).
gmix	Mole fraction of gap-gas constituent.
gmles	Moles of gap gas.
pgapt	Gap-gas total pressure.
plvol	Rod plenum volume.
pslen	Pellet stack length.
! Time-Dependent Data.	
cdg	Old concentration of delayed-neutron group.
cdh	Old concentration of decay-heat group.
clen	Old total cladding length.
cdgn	New concentration of delayed-neutron group.
cdhn	New concentration of decay-heat group.
clenn	New total cladding length.
! Rod and slab dependent data.	

burn	Fuel burnup.
cnd	ROD or SLAB thermal conductivity.
cndr	ROD or SLAB thermal conductivity to the right of the interface.
cpnd	ROD or SLAB specific heat.
zht	Axial location of the heat-transfer node.
emis	ROD or SLAB surface emissivity.
hgap	Gap-gas conductance.
pgap	Gap-gas pressure.
pint	Pellet-cladding contact pressure.
pldv	Pellet dish volume.
qwrx	Metal-water reaction heat source.
rnd	ROD or SLAB density.
idht	ROD or SLAB node identifier.
cpdr	ROD or SLAB specific heat to the right of the interface.
rndr	ROD or SLAB density to right of the material interface.
rpowf	ROD or SLAB power density.
! Time dependent rod data.	
radr	Old radial-node positions.
radrn	New radial-node positions.
drz	Old zirconium-dioxide reaction depth.
drzn	New zirconium-dioxide reaction depth.
rft	Old fine-mesh ROD or SLAB temperatures.
rftn	New fine-mesh ROD or SLAB temperatures.
! Surface dependent rod data.	
alpr	Gas-volume fraction.
alvr	Liquid HTC times the interfacial area.
clr	Liquid thermal conductivity.
cvr	Gas thermal conductivity.
cplr	Liquid specific heat.
cpvr	Gas specific heat.
concr	Mass ratio of dissolved solute in the liquid (kg solute/kg liquid).
hdr	Rod-bundle hydraulic diameter.
hfgr	Latent heat of vaporization of the fluid.
chtir	Gas HTC times interfacial area.
ihtf	Fine-mesh, heat-transfer-regime flag.
pr	Total pressure.
rolr	Liquid density.

rovr	Gas density.
romr	Mixture density.
volr	Fluid volume in hydrodynamic mesh cells.
stnu	Stanton number.
tld	Liquid temperature at bubble departure.
sr	Density of plated-out solute.
sigr	Surface tension.
tchfr	Wall temperature at the CHF point.
tchff	Fine-mesh wall temperature at the CHF point.
hgamr	Energy contribution to subcooled boiling.
hlar	Sum of the products of the liquid HTC and the heat-transfer area.
hvar	Sum of the products of the gas HTC and the heat-transfer area.
hlatr	Sum of the products of the liquid HTC, the heat-transfer area, and the wall temperature.
hvatr	Sum of the products of the gas HTC, the heat-transfer area, and the wall temperature.
finar	Variable not used.
watr	ROD or SLAB total heat-transfer area.
tlr	Old liquid temperature.
tvr	Old gas temperature.
tlnr	New liquid temperature.
tvnr	New gas temperature.
tsatr	Saturation temperature.
vislr	Liquid viscosity.
visvr	Gas viscosity.
vlzr	Axial liquid velocity.
vvzr	Axial gas velocity.
vvcr	Gas cross-flow velocity.
vmzr	Axial mixture velocity.
elr	Liquid internal energy.
evr	Gas internal energy.
drvdt	Derivative of the gas density with respect to the gas temperature.
drldt	Derivative of the liquid density with respect to the liquid temperature.
ear	Specific internal energy of the noncondensable-gas component.
roar	Noncondensable-gas density.

par	Noncondensable-gas partial pressure.
tssnr	Saturation temperature corresponding to the partial pressure of steam.
hlsr	Specific enthalpy of the liquid phase at saturation (corresponding to the saturation temperature at the partial pressure of steam).
hvsr	Specific enthalpy of the steam (not gas) at saturation (at the partial pressure of steam and saturation temperature).
!	
hrfg	New subcooled-boiling HTC.
hrfgo	Old subcooled-boiling HTC.
hrfl	New fine-mesh liquid HTC.
hrflo	Old fine-mesh liquid HTC.
hrfv	New fine-mesh gas HTC.
hrfvo	Old fine-mesh gas HTC.
hrlg	New fine-mesh, subcooled-boiling HTC.
hrlgo	Old fine-mesh, subcooled-boiling HTC.
hrll	New liquid HTC for the lower half node.
hrllo	Old liquid HTC for the lower half node.
hrlv	New gas HTC for the lower half node.
hrlvo	Old gas HTC for the lower half node.
hqrad	New radiation power absorbed by the coolant.
hqrdo	Old radiation power absorbed by the coolant.
qchff	New CHF.
qchfo	Old CHF.
qchfr	New fine-mesh CHF.
qchro	Old fine-mesh CHF.
rdhlr	Liquid HTC.
rdhlo	Variable not currently implemented.
rdhvr	Gas HTC.
rdhvo	Variable not currently implemented.
!	
tcefn	New total convective power.
tcefo	Old total convective power.
twaen	New absolute total conduction.
twaeo	Old absolute total conduction.
tween	New effective total conduction.
tweeo	Old effective total conduction.

cepwn	New total convective power.
cepwo	Old total convective power.

INTEGER VARIABLES:

sdkx	Required to declare real KIND functions with the Nagware F90 compiler.
sikx	Required to declare integer KIND functions with the Nagware F90 compiler.

!Elements of DERIVED TYPE `hsArrayT`

ntsxx	Number of mesh cells in the plane transverse to the axial direction.
noht	Number of rows of heat-transfer nodes for each ROD or SLAB.

C.24. Module `HeatArray`

```
TYPE heatArrayT
    REAL(sdk),  POINTER, DIMENSION(:,:)  :: cpw
    REAL(sdk),  POINTER, DIMENSION(:,:)  :: cw
    REAL(sdk),  POINTER, DIMENSION(:)    :: dr
    REAL(sdk),  POINTER, DIMENSION(:)    :: emis
    REAL(sdk),  POINTER, DIMENSION(:)    :: hol
    REAL(sdk),  POINTER, DIMENSION(:)    :: hov
    REAL(sdk),  POINTER, DIMENSION(:)    :: rn
    REAL(sdk),  POINTER, DIMENSION(:)    :: rn2
    REAL(sdk),  POINTER, DIMENSION(:,:)  :: row
    REAL(sdk),  POINTER, DIMENSION(:)    :: tchf
    REAL(sdk),  POINTER, DIMENSION(:)    :: tol
    REAL(sdk),  POINTER, DIMENSION(:)    :: tov
  END TYPE heatArrayT
!
TYPE (heatArrayT),DIMENSION(maxComps) :: heatAr
```

DERIVED TYPE VARIABLES:

heatAr	Arrays specific to embedded-wall, heat-transfer calculations.

REAL VARIABLES:

!Elements of DERIVED TYPE `heatArrayT`

cpw	Specific heat of wall.
cw	Wall thermal conductivity.
dr	Radial mesh size.
emis	Wall-surface emissivity.

hol	HTC between wall and outside liquid.
hov	HTC between wall and outside gas.
rn	Radii at nodes.
rn2	Radii at node centers.
row	Wall density.
tchf	CHF temperature.
tol	Liquid temperature, outside wall.
tov	Gas temperature, outside wall.

C.25. Module `HpssDat`

```
TYPE hpsT
!     Dynamically allocated to nloops in SUB icomp:
      REAL(sdk),POINTER,DIMENSION(:) :: masi
      REAL(sdk),POINTER,DIMENSION(:) :: masm
      REAL(sdk),POINTER,DIMENSION(:) :: masn
      REAL(sdk),POINTER,DIMENSION(:) :: mast
!     Dynamically allocated to npaths in SUB input:
      INTEGER(sik),POINTER,DIMENSION(:) :: dcinf
      INTEGER(sik),POINTER,DIMENSION(:) :: dcloc
      INTEGER(sik),POINTER,DIMENSION(:) :: dcouf
      INTEGER(sik),POINTER,DIMENSION(:) :: dcpwi
      INTEGER(sik),POINTER,DIMENSION(:) :: dcpwo
      INTEGER(sik),POINTER,DIMENSION(:) :: ddinf
      INTEGER(sik),POINTER,DIMENSION(:) :: ddloc
      INTEGER(sik),POINTER,DIMENSION(:) :: ddouf
      INTEGER(sik),POINTER,DIMENSION(:) :: ddpwi
      INTEGER(sik),POINTER,DIMENSION(:) :: ddpwo
!
      REAL(sdk),POINTER,DIMENSION(:)   :: phm
      REAL(sdk),POINTER,DIMENSION(:)   :: pmvl
      REAL(sdk),POINTER,DIMENSION(:)   :: pmvv
      REAL(sdk),POINTER,DIMENSION(:)   :: power
      REAL(sdk),POINTER,DIMENSION(:)   :: pp
      REAL(sdk),POINTER,DIMENSION(:)   :: ppa
      REAL(sdk),POINTER,DIMENSION(:)   :: ptl
      REAL(sdk),POINTER,DIMENSION(:)   :: ptv
!
```

```
END TYPE hpsT
!
TYPE(hpsT) :: hps
!
INTEGER(sik)  nic(200),iic,nfpi,npaths,ntpi
DATA iic/0/
```

DERIVED TYPE VARIABLES:

hps — Arrays specific to embedded wall heat-transfer calculations.

REAL VARIABLES:

!Elements of DERIVED TYPE hpsT

! Dynamically allocated to nloops in SUBROUTINE icomp.

masi — Pointer variable for the input-specified coolant mass in each hydraulically coupled region of the system model.

masm — Pointer variable for the two-phase coolant mass based on $\alpha = \alpha_m$ and $T_G = T_{sat} = T_L$ in each hydraulically coupled region of the system model.

masn — Pointer variable for the two-phase coolant mass based on $\alpha = \alpha_n$ and $T_G = T_{sat} = T_L$ in each hydraulically coupled region of the system model.

mast — Pointer variable for the coolant mass based on $\alpha = 0, \alpha_n$, or 1 and $T_G = T_L$ in each hydraulically coupled region of the system model.

!

phm — Pointer variable for the initial liquid mass flow or velocity at the location defining the hydraulic condition.

pmvl — Pointer variable for the initial liquid mass flow or velocity at the interface location defining the hydraulic condition.

pmvv — Pointer variable for the initial gas mass flow or velocity at the interface location defining the hydraulic condition.

power — Pointer variable for the total heat source or sink power between and including cells `a(lddinf)` to `a(lddouf)`.

pp — Pointer variable for the total pressure in all cells along the hydraulic path (when `ntpi` = 0).

ppa — Pointer variable for the noncondensable-gas pressure in all cells along the hydraulic path (when `ntpi` = 0).

ptl — Pointer variable for the initial liquid temperature at the donor-cell location defining the hydraulic condition.

ptv — Pointer variable for the initial gas temperature at the donor-cell location defining the hydraulic condition.

INTEGER VARIABLES:

!Elements of DERIVED TYPE hpsT

! Dynamically allocated to npaths in SUBROUTINE input.

dcinf	Pointer variable for the 1D-component ID number of the hydraulic-path inflow location of the hydraulic path.
dcloc	Pointer variable for the 1D-component ID number of the hydraulic-path condition location defining the hydraulic condition.
dcouf	Pointer variable for the 1D-component ID number of the hydraulic-path outflow location of the hydraulic path.
dcpwi	Pointer variable for the 1D-component ID number of the hydraulic-path inflow location of the first cell having a heat source or sink.
dcpwo	Pointer variable for the 1D-component ID number of the hydraulic-path inflow location of the last cell having a heat source or sink.
ddinf	Pointer variable for the interface number of the hydraulic-path inflow location of the hydraulic path.
ddloc	Pointer variable for the interface number of the hydraulic-path conditon location defining the hydraulic condition.
ddouf	Pointer variable for the interface number of the hydraulic-path outflow location of the hydraulic path.
ddpwi	Pointer variable for the cell number of the hydraulic-path inflow location of the first cell having a heat source or sink.
ddpwo	Pointer variable for the cell number of the hydraulic-path inflow location of the last cell having a heat source or sink.
!	
iic	Total number of components that are not being initialized by the hydraulic-path, steady-state initialization procedure.
nfpi	Mass-flow or velocity input option. 0 = input liquid and gas mass flows; 1 = input liquid and gas velocities.
nic	Component ID numbers that are not being initialized by the hydraulic-path, steady-state initialization procedure.
npaths	Number of 1D hydraulic paths defined in the system model.
ntpi	Total pressure and noncondensable-gas pressure input option. 0 = input pressures for all hydraulic-path cells; 1 = define the hydraulic-condition, donor-cell pressures for all hydraulic-path cells; 2 = define pressure from the component input data.

C.26. Module `IntArray`

```
TYPE intArrayT
    REAL(sdk),  POINTER, DIMENSION(:)  :: idr
    REAL(sdk),  POINTER, DIMENSION(:)  :: matid
    REAL(sdk),  POINTER, DIMENSION(:)  :: nff
    REAL(sdk),  POINTER, DIMENSION(:)  :: lccfl
  END TYPE intArrayT
!
TYPE (intArrayT),DIMENSION(maxComps) :: intAr
```

DERIVED TYPE VARIABLES:

intAr	Integer component-specific arrays.

REAL VARIABLES:

!Elements of DERIVED TYPE intArrayT

idr	Heat-transfer regime.
matid	Structural material identification.
nff	Friction-correlation option.
lccfl	CCFL flag.

C.27. Module `IntrType`

```
INTEGER, PARAMETER :: sdk = selected_real_kind (13,307)
INTEGER, PARAMETER :: sik = kind (10000000)
```

INTEGER VARIABLES:

sdk	The real KIND required to declare variables with the desired precision and range.
sik	The integer KIND required to declare variables with the desired precision and range.

C.28. Module `Io`

```
CHARACTER*100 card
!
INTEGER(sik) ibfadd,ibfadg,ibfadr,ibflnd,ibflng,ibflnr,idout,    &
& ieeeg,igout,ikeybd,imout,in,inlab,inproc,ioall,iodone,ioerr,    &
& iogrf,ioinp,iolab,ioout,ioskip,iout,irstrt,itty,iunlab,iunout,  &
& lcmcpd,nittab,npwtab,nrdy ,xtvres
!
```

```
DATA in,iout,itty,igout,idout,irstrt,imout/35,36,6,11,12,13,7/
DATA ikeybd/5/
DATA ioall,iogrf,ioinp,iolab,ioout,iunlab,iunout/6*0,1/
DATA ioerr,iodone,ioskip,inproc/0,0,0,0/
DATA inlab,nittab,npwtab/0,0,0/
DATA ieeeg/0/
DATA xtvres/4/
```

CHARACTER VARIABLES:

card	Variable that contains the current input-card data.

INTEGER VARIABLES:

ibfadd	Pointer to the beginning of dump LCM buffer.
ibfadg	Pointer to the beginning of graphics LCM buffer.
ibfadr	Pointer to the beginning of restart LCM buffer.
ibflnd	Length of dump buffer.
ibflng	Length of graphics buffer.
ibflnr	Length of restart buffer.
idout	I/O unit number for dump output file (currently set to unit 12).
ieeeg	Option to specify the format of the TRCGRF file for graphics output (namelist variable). 0 = unformatted binary file; 1 = IEEE standard-format binary file.
igout	I/O unit number for graphics output file (currently set to unit 11).
ideybd	I/O unit number for the terminal keyboard (currently set to unit 59 for –DEF,IBM and unit 5 for DEF,IBM).
imout	I/O unit number for warning messages (currently set to unit 7).
in	I/O unit number for input to TRAC-M (initially set to unit 5 to point to file TRCINP; if the input does not invoke free format, IN is changed to 1 to point to file TRACIN).
inlab	I/O unit number for TRAC to generate a labeled input-data file (currently set to unit 3).
inproc	Flag set during input to indicate whether component data are being processed.
ioall	ioall = |iogrf| + |ioinp| + |iolab| + |ioout|.
iodone	Flag that indicates if the current input card has been read.
ioerr	Input error flag.
iogrf	SI/English-units flag for writing graphics data to file TRCGRF (namelist variable).

ioinp	SI/English-units flag for reading input data from file TRACIN (namelist variable).
iolab	SI/English-units flag for writing comment-labeled input data to file INLAB (namelist variable).
ioout	SI/English-units flag for echoing input and restart data, writing small and large edits to file TRCOUT, and writing calculative information to file TRCMSG and the terminal (namelist variable).
ioskip	Flag that turns input processing off and on.
iout	I/O unit number for the printed-output file (currently set to unit 6).
irstrt	I/O unit number for the restart-input file (currently set to unit 13).
itty	I/O unit number for terminal output (currently set to unit 59).
iunlab	Option for inputting user-defined, units-name labels required for defining the units of control-block or trip signal-expression parameters (namelist variable).
iunout	Option for writing SI/English units to file TRCOUT (namelist variable).
lcmcpd	Storage for the beginning address for reading from or writing to LCM with calls to subroutines `rdlcm` and `wrlcm`.
nittab	Flag for printing the timestep-data-table heading label to the terminal (–1) or message file (–2) because a warning message(s) has been printed since the last table values were printed.
npwtab	Flag for printing the power/reactivity feedback table heading label to the message file (–1) because a warning message(s) has been printed since the last table values were printed.
nrdy	Flag for reading the input-data files. 0 = reading the TRCRST restart-data file; 1 = reading the TRACIN user-input-data file.
xtvres	This toggles graphics resolution between 4 and 8 bytes.

C.29. Module `JunTerms`

Derived Types

junValuesT	At each segment junction, this provides information necessary to evaluate system matrix coefficients, linking system variables across that junction.

Structure:

```
TYPE, PRIVATE :: junValuesT
```

```
  REAL(sdk), POINTER :: faLiqFrac, faVapFrac, faArv, faArl
  REAL(sdk), POINTER :: faWlOut, faWlIn, faWvIn, faWvOut
  REAL(sdk), POINTER :: faAra, faArev, faArel, faLiq, faVap
  REAL(sdk), POINTER :: vl, vv, vln, vvn, dvldp, dvvdp, xvset
  REAL(sdk), POINTER :: alpn, rovn, roln, roan, evn, eln, p, pn
  REAL(sdk), POINTER :: arl, arv, ara, arel, arev, conc
  INTEGER(sik) :: ivarOut, ivarIn, iaob, vsign
END TYPE junValuesT
```

Real components:

alpn - new-time void fraction in the junction cell

ara - old-time product of vapor fraction and noncondensable density in the junction cell

arel - old-time product of liquid fraction, liquid density, and specific liquid internal energy in the junction cell

arev - old-time product of vapor fraction, vapor density, and specific vapor internal energy in the junction cell

arl - old-time product of liquid fraction and liquid density in the junction cell

arv - old-time product of vapor fraction and gas density in the junction cell

conc - old-time solute concentration in the junction cell

dvldp - derivative of the junction liquid velocity with respect to pressure (evaluated from the side with doEdge=.TRUE.)

dvvdp - derivative of the junction gas velocity with respect to pressure (evaluated from the side with doEdge=.TRUE.)

eln - new-time liquid specific internal energy in the junction cell

evn - new-time vapor specific internal energy in the junction cell

faAra - product of the junction effective vapor flow area and the edge-average macroscopic noncondensable density (void fraction times noncondensable density)

faArl - product of the junction effective liquid flow area and the edge-average macroscopic liquid density (liquid fraction times liquid density)

faArv - product of the junction effective vapor flow area and the edge-average macroscopic gas density (void fraction times density of the mixed steam and noncondensable gas)

faArel - product of the junction effective liquid flow area and the edge-average macroscopic liquid-energy density (liquid fraction times liquid density)

faArev - product of the junction effective vapor flow area and the edge-average macroscopic gas-energy density (void fraction times density of the mixed steam and noncondensable gas)

faLiq - effective area for liquid flow at the junction (normally geometric flow area)

faLiqFrac - product of the junction effective liquid flow area and the edge-average liquid fraction

faVap - effective area for vapor flow at the junction (normally geometric flow area)

faVapFrac - product of the junction effective flow area and the edge-average gas fraction

faWlIn - product of the junction effective liquid flow area and the weight factor used on the liquid variable in this junction cell to compute the junction face average

faWlOut - product of the junction effective liquid flow area and the weight factor used on the liquid variable across the junction from this junction cell to compute the junction face average

faWvIn - product of the junction effective vapor flow area and the weight factor used on the gas variable in this junction cell to compute the junction face average

faWvOut - product of the junction effective vapor flow area and the weight factor used on the gas variable across the junction from this junction cell to compute the junction face average

roan - new-time noncondensable gas density in the junction cell

p - old-time total pressure in the junction cell

pn - new-time total pressure in the junction cell

roln - new-time liquid density in the junction cell

rovn - new-time gas density in the junction cell

vl - junction old-time liquid velocity (evaluated from the side with doEdge=.TRUE.)

vln - junction new-time liquid velocity (evaluated from the side with doEdge=.TRUE.)

vv - junction old-time gas velocity (evaluated from the side with doEdge=.TRUE.)

vvn - junction new-time gas velocity (evaluated from the side with doEdge=.TRUE.)

xvset - value of the special nonconservative adjustment coefficient for fluxes exiting a cell in the presence of phase change

INTEGER Components:

iaob - index in `at%aob` to which the above junction averages contribute (zero if this not an off-band connection)

ivarIn - index of the system variable in this cell (one weighted by `faWlIn` or `faWvIn`)

ivarOut - index of the system variable on the other side of this junction (one weighted by `faWlOut` or `faWvOut`)

vsign - number by which `vln` or `vvn` must be multiplied (+1 or -1) to conform to a convention that positive velocity is directed out of this junction cell.

Variables

REAL:

scaleDp - a real parameter set to the value 999,999, used to scale effects of pressure variations during water packing corrections. This should be moved once a standard module (or set of modules) is established for storage of basic constants.

Derived Type

junVal - an array with derived type `junValues` containing pointers and index information to obtain cell edge averages at intersegment junctions. Information is stored separately for both cells adjacent to the junction. There is a one-to-one correspondence between elements in this array and those in `junCells` (see Module `SysConfig` below)

C.30. Module `Linear`

INTEGER(sik),PARAMETER :: sikx=sik

INTEGER(sik),PARAMETER :: sdkx=sdk

INTEGER VARIABLES:

sikx Required to declare integer KIND functions with the Nagware F90 compiler.

sdkx Required to declare real KIND functions with the Nagware F90 compiler.

C.31. Module `Matrices`

Derived Types

adjEdgIndT a derived type containing index information for sparse matrix coefficients for the two edges immediately adjacent to the edge represented by the given element of this derived type.

Structure:

```
TYPE adjEdgIndT
  INTEGER(sik) :: p,m
  LOGICAL :: offBandP, offBandM
  REAL(sdk) :: signM, signP
END TYPE adjEdgIndT
```

LOGICAL Components:

offBandP - when this logical variable is true, coupling to the next edge in the mesh segment's positive direction is off the main matrix band

offBandM - when this logical variable is true, coupling to the next edge in the mesh segment's negative direction is off the main matrix band

INTEGER Components:

p - index for the adjacent edge in the positive direction of the mesh segment responsible for computation of this edge. If `offbandP` is false, this gives the index in `al%a` or `av%a`; otherwise, it contains the index in `al%aob` or `av%aob`.

m - index for the adjacent edge in the negative direction of the mesh segment responsible for computation of this edge. If `offbandP` is false, this gives the index in `al%a` or `av%a`; otherwise, it contains the index in `al%aob` or `av%aob`.

REAL Components:

signM - relative sign convention of the velocity at the adjacent cell in the negative direction

signP - relative sign convention of the velocity at the adjacent cell in the positive direction

blockmatrixT a derived type containing coefficients and constants associated with the block matrix for semi-implicit equations in a given cell. One restriction on this structure is that elements associated with a given subnetwork matrix are contiguous.

Structure:

```
TYPE blockMatrixT
  REAL(sdk) :: a(nivars,nivars), b(nivars), bp(nivars)
  REAL(sdk) :: fluxSum(nivars), fluxltSum, fluxvtSum
```

```
  REAL(sdk) :: liqVolFluxSum, vapVolFluxSum
  REAL(sdk) :: faWvInVvSum, faWlInVlSum
 LOGICAL :: sourceLiq, reDoJacobian, singlePhase, tempEquilib
 LOGICAL :: changeVvSign, changeVlSign, noLiqVel, noVapFlow
 LOGICAL :: airEqn, tvLimit
 INTEGER(sik) :: nfaces
 INTEGER(sik), POINTER, DIMENSION(:) :: index
 REAL(sdk),  POINTER, DIMENSION(:,:) :: cDp, cDpp
 LOGICAL, POINTER, DIMENSION(:) :: wpack
END TYPE blockMatrixT
```

REAL Components:

a - coefficient matrix for the cell

b - right-hand-side array associated with the linearized equations

bp - reduced form of the constants in component "b" (result of multiplying the inverse of "a" times "b")

cDp - contains coefficients of pressure differences associated with faces of the cell

cDpp - reduced form of the coefficients in `cDp` (result of multiplying the inverse of "a" times `cDp`)

FaWlInVlSum - sum of the coefficients multiplying the liquid macroscopic density or energy in the current cell contributing to mass or energy flux terms (used in `xvset` calculation)

FaWvInVvSum - sum of the coefficients multiplying the gas macroscopic density or energy in the current cell contributing to mass or energy flux terms (used in `xvset` calculation)

fluxSum - an array with `nivars` values containing the sum of all basic fluxes in and out of the cell for the conservation equation

fluxltSum - sum of all liquid mass fluxes in and out of the cell, based on the velocities with `vlx` (see logic in `tf1d`)

fluxvtSum - sum of all gas mass fluxes in and out of the cell, based on the velocities associated with `vvx` (see logic in `tf1d`)

liqVolFluxSum - sum of liquid volumetric fluxes to a cell from side junctions

vapVolFluxSum - sum of gas (vapor plus noncondensable) volumetric fluxes to a cell from side junctions

INTEGER Components:

index - index of the element in array "blocks" to which the corresponding coefficient in cDpp couples

nfaces - total number of cell faces connected to this computational volume (includes main mesh and side junctions)

LOGICAL Components:

airEqn - set TRUE if the normal air mass equation is evaluated

changeVlsign - set to TRUE if the liquid velocity at any face on this cell changes sign

changeVvsign - set to TRUE if the gas velocity at any face on this cell changes sign

noLiqVel - set to TRUE if the magnitudes of liquid velocities are less than or equal to 1.e-10 at all faces to this cell

noVapFlow - set to TRUE if no vapor flow occurs at any face of this cell

reDoJacobian - set to TRUE if a negative void fraction estimate requires recalculation of the Jacobian determinant based on mean flow equations

singlePhase - set TRUE if single-phase equations are used

sourceLiq - logical variable set to TRUE if no sources of liquid are found to this cell in side-junction fluxes

tempEquilib - set to TRUE if the temperature of one phase is driven to equilibrium

tvLimit - set TRUE if the mean energy equation is replaced by an equation limiting the vapor temperature

wpack - a logical array (length nfaces) indicating if a water pack has occurred in this cell at any of the faces with coefficients in the cDp array

NOTE: By convention, elements in cDp are ordered so that the first is the coupling coefficient to the cell with the next lower index in the primary direction, and the second is the coefficient coupling to the cell with the next higher index in the primary direction. The sign convention of the elements of cDp is chosen so that each is the coefficient multiplying the difference between the pressure on the opposite side of the connecting face and the pressure in the current cell.

coef3Dto1DT — derived-type array containing coefficients for rows in a 3D submatrix that couple to 1D variables. Coefficients labeled "a" are those that directly multiply variables in the 1D region, and the location of these variables is obtained from the corresponding elements of `iNet` and `i1D` contained in derived-type `jun3Dto1DT`.

Structure:

```
TYPE coef3Dto1DT
 REAL(sdk), POINTER :: a(:)
END TYPE coef3Dto1DT
```

REAL Components:

a - pointer array containing the `n3Dto1D` matrix coefficients (see `jun3Dto1DT`), which multiply network variables from 1D regions

jun3Dto1DT — derived-type array containing information on elements of a given row in a 3D submatrix that couple to 1D variables. Actual coefficients are stored in the derived-type `coef3Dto1DT`.

Structure:

```
TYPE jun3Dto1DT
 INTEGER(sik),POINTER :: i1D(:), iNet(:), junCell(:)
 INTEGER(sik) :: n3Dto1D, i3D
END TYPE jun3Dto1DT
```

INTEGER Components:

i1D - pointer array containing the `n3Dto1D` indices of variables multiplied by the corresponding coefficients in `coef3Dto1DT`. These indices point directly to elements of the Network matrix solution in `rhsNet%b`

i3D - pointer array containing the index of the row (equation) in the 3D matrix block to which these terms contribute

iNet - pointer array component giving the subNetwork index for the element of the derived-type `netRHS` containing necessary information for the variables multiplied by the coefficient in `coef3Dto1DT`.

junCell - index in the `junCells` array with information on the vessel side of this junction

n3Dto1D - number of coefficients in this row multiplying Network variables from the 1D regions

momentumTransT This derived type is responsible for gathering momentum transport terms, particularly for the stabilizer motion equations. Because face connections transport momentum across volume cells, elements of this type are directly associated with a system volume.

Structure:

```
TYPE momentumTransT
 INTEGER(sik), POINTER :: nfaces
 INTEGER(sik), POINTER :: iface(:,:)
 REAL(sdk), POINTER :: coefl(:,:), coefv(:,:)
 REAL(sdk), POINTER :: srcl(:), srcv(:)
END TYPE momentumTransT
```

Real Components:

coefl - a 2D matrix containing coupling coefficients for liquid-momentum transport terms. `coefl(i,j)` provides a coefficient in the equation associated with the edge with system index `momTran%iface(i,i)`. This coefficient multiplies the yet-to-be-determined system variable `vltS(momTran%iface(j,j)`. If the coefficient is on the main band of the matrix system solution matrix (`al` or `av`), then it is stored in `al(momTran%iface(i,i))%a`. If the coefficient is off the main band, then it is stored as `al(momTran%iface(i,i))%aob(ABS(iface(i,j)`

coefv - a 2D matrix containing coupling coefficients for vapor momentum transport terms.

srcl - a 1D array collecting the constant portions of the liquid momentum sources for the equations at each corresponding face

srcv - a 1D array collecting the constant portions of the vapor momentum sources for the equations at each corresponding face

INTEGER Components:

nfaces - number of cells faces exchanging momentum across this volume

iface - indices of faces exchanging momentum. This is a `2Dl` matrix, with

each row providing information for the momentum equation associated with the index on the matrix diagonal. Columns give indices of faces coupling to each momentum equation, and a negative index in this matrix gives the position where this transport contributes to an off-band coefficient in the sparse matrix data [e.g., `al(i)%aob(iface(2,3)]`

netMatT — a derived type containing the network matrices. Two instances of this are declared, one for the cell center and one for the cell edge.

Structure:

```
TYPE netMatT
 REAL(sdk), POINTER, DIMENSION(:,:) :: a, c3D
 INTEGER(sik), POINTER :: ipvt(:)
END TYPE netMatT
```

REAL Components:

a - pointer array containing the subNetwork matrix, dimensioned `a(netInd%nvar,netInd%nvar)`

c3D - doubly dimensioned array containing the coefficients on the right-hand side of the network equations, multiplying unknowns within 3D regions, dimensioned `c3D("net%nvar","net%n3d")`

INTEGER Components:

ipvt - pivot index array associated with LU decomposition of array "a"

netMatIndT — derived type containing indices and sizes associated with the subNetwork matrices. Two instances of this are declared, one for the cell center and one for the cell edge.

Structure:

```
TYPE netMatIndT
 INTEGER(sik) :: nvar, n3D
 INTEGER(sik), POINTER :: i3Dvar(:), ivar(:)
END TYPE netMatIndT
```

INTEGER Components:

i3Dvar - pointer array containing indices of 3D variables within the system arrays (`araS`, `arlS`, etc.). `i3Dvar(i)` is the index of the 3D variable multiplying a coefficient such as `rhsNet(iNet)%c3D(j,i)`.

ivar - pointer array containing indices of variables within the systemwide arrays (araS, argS, etc.) corresponding to the network variables in this derived type (dimension is nvar)

n3D - number of connections between this 1D equation system and the 3D equations (number of physical connections between the 1D loop and 3D components)

nvar - number of network variables in this system

netRHST a derived type containing the right-hand-side information for the network equations and returning final solutions.

Structure:

```
TYPE netRHST
  REAL (sdk), POINTER :: b(:)
END TYPE netRHST
```

REAL Components:

b - pointer array receiving the right-hand sides for the nvar network variable equations and returning the solution for the equations, dimensioned araNet(net%nvar)

netVarIndT a derived type providing indices to elements attached to the "positive" and "negative" sides of a network junction.

Structure:

```
TYPE netVarIndT
 INTEGER(sik) :: pos, neg
END TYPE netVarIndT
```

INTEGER Components:

pos - index of an array element to the positive side the network junction

neg - index of an array element to the negative side of the network junction

sparseIndicesT derived-type array containing all indices associated with the sparseMatrixT derived type.

Structure:

```
TYPE sparseIndicesT
 INTEGER(sik), POINTER, DIMENSION(:) :: aob
 INTEGER(sik) :: nOffBand, iTri
END TYPE sparseIndicesT
```

INTEGER Components:

aob - pointer array component of sparsematrix containing the `nOffBand` indices of variables multiplied by the corresponding coefficients in `aob`. If the index is negative, it references a variable in a 3D region giving its index in the `i3Dvar` component of the `netMatInd` derived type.

iTri - index in the `triIndices` type array for information on the associated tridiagonal submatrix. Rows in the matrix that are not in a tridiagonal submatrix contain coefficients needed for the network matrix, and `iTri` is set to the negative of the network equation number for the current subNetwork matrix.

nOffBand - number of off-band coefficients in this row (size of `aob` and `index`)

sparseMatrixT derived-type array containing all coefficients for the i^{th} row of the matrix. One restriction on this structure is that elements associated with a given subNetwork matrix are contiguous. Information content of this type is designed for 1D flow networks. Equation information for equations contained in 3D regions is contained in `vessMatrixT` and `vessMatIndT`.

Structure:

```
TYPE sparseMatrixT
 REAL(sdk) :: a(bandWidth)
 REAL(sdk) :: clow,chigh
 REAL(sdk), POINTER, DIMENSION(:) :: aob
END TYPE sparseMatrixT
```

REAL Components:

a - array component of sparsematrix containing the `bandWidth` coefficients associated with connections along the primary 1D mesh

aob - pointer array component of `sparsematrix` containing the `nOffBand` coefficients linking variables not in the band for the primary mesh connections (TEE, connections, etc.)

chigh - this contains the coefficient multiplying the network variable at the

high end of the submatrix, when the variable associated with this row of the sparse matrix is obtained as a linear function of the network variables

clow - this contains the coefficient multiplying the network variable at the low end of the submatrix, when the variable associated with this row of the sparse matrix is obtained as a linear function of the network variables

triIndicesT a derived type providing information about indices needed to locate tridiagonal submatrices within systemwide coefficient and constant arrays and to couple them to the appropriate network matrix.

Structure:

```
TYPE triIndicesT
 INTEGER(sik) :: ilow, ihigh, netNum, netLow, netHigh
END TYPE triIndicesT
```

INTEGER Components:

ilow - index in the systemwide array that starts the submatrix

ihigh - index in the systemwide array that ends the submatrix

netNum - index of the network matrix (or loop index) associated with this submatrix

netLow - index in the network variable associated with the network variable directly coupled to the low end of this tridiagonal submatrix

netHigh - index in the network variable associated with the network equation and variable directly coupled to the high end of this tridiagonal submatrix

vssMatrixT a derived type storing equation coefficients and constants necessary for the solution within the 3D regions of the full linear system. Arrays are specified based on use of the Capacitance Matrix method described in the TRAC Theory Manual**need ref.** .

Structure:

```
TYPE vssMatrixT
 REAL(sdk), POINTER, DIMENSION(:,:) :: a,emat, rmat, dmat
 REAL(sdk), POINTER, DIMENSION(:) :: bt
 INTEGER(sik), POINTER :: ipvt(:)
 TYPE (coef3Dto1DT), POINTER :: jn(:)
 INTEGER(sik) :: lr
END TYPE vssMatrixT
```

REAL Components:

a - an array containing coefficients for the banded portion of the 3D matrix; storage is described in subroutine `dgbfa` contained in the module `LinearM`]dimensioned `a(mdim, nrows3D)`]

bt - an intermediate rank-1 array used in the capacitance matrix method

dmat - an intermediate matrix generated during the capacitance matrix solution

emat - the matrix defining the rows containing off-band coupling

rmat - the matrix containing off-band coefficients for each row flagged in `emat`

INTEGER Components:

ipvt - an array for pivot indices used for communicating partial pivoting information between `dgbfa` and `dgbsl`

lr - the number of 3D equations containing off-band coupling to other 3D equations after substitution of the partially solved 1D equations

DERIVED TYPE Components:

jn - a pointer array of derived-type `coef3Dto1D` containing coefficients for those rows containing coupling to 1D junction variables (has `nJunRows` elements)

vssMatIndT — a derived type storing indices and sizes necessary for the solution within the 3D regions of the full linear system. Arrays are specified based on use of the capacitance matrix method described in the TRAC Theory Manual**need ref.** .

Structure:

```
TYPE vssMatIndT
 INTEGER(sik) :: nvar, nJunRows, bandwidth, mu, ldim, mdim
 INTEGER(sik) :: istartS
 TYPE (jun3Dto1DT), POINTER :: jn(:)
END TYPE vssMatIndT
```

INTEGER Components:

bandWidth - bandwidth of a3D%a (2*mu+1)

istartS - starting point for these 3D variables in the full-system arrays [e.g., arlS(istartS)]

ldim - first dimension of rmat, second dimension of emat

mdim - first dimension of a3D%a (at least 3*mu+1)

mu - number of upper diagonals in the band (also equal to the number of lower diagonals in the band)

nJunRows - number of rows (equations) containing junctions with 1D regions (this is the size of the array jn of coefficients to 1D junction variables)

nvar - number of variables (matrix order) in this 3D region

DERIVED TYPE Components:

jn - a pointer array of derived-type jun3Dto1D containing indices for those rows with coupling to 1D junction variables (has nJunRows elements)

Variables

REAL:

```
REAL(sdk), ALLOCATABLE, TARGET, DIMENSION (:) :: arlS, arvS,     &
&     arelS, arevS, araS, arcS, vvtS, pS, vltS, arlRHS,      &
&     arvRHS, arelRHS, arevRHS, araRHS, arcRHS, pRHS,        &
&     vvtRHS, vltRHS
REAL(sdk), POINTER, DIMENSION (:) :: rhs(:), rhs3D(:)
```

For each independent variable in an equation solution (p, arl, arv, arc, ara, arel, arev, vlt, and vvt), arrays are allocated to store the variable values in the full-system solution. The names for these arrays begin with the independent variable name and end with "S". In addition, variables are available, but not currently used, to store the right-hand sides (residuals) of equations associated with these variables. Names for these Fortran variables end with "RHS". The pointer array rhs is associated with the array containing the right-hand side of the equation set currently being solved. The pointer array rhs3D is associated with the portion of those equations originating in 3D regions.

INTEGER:

```
INTEGER(sik) :: nrowsC1D, nrowsE1D, nrows1D, nrows3D, nrowsC
INTEGER(sik) :: nsplitsC, nsplitsE, nsplits
```

```
INTEGER(sik) :: n3DmatC, n3DmatE, n3Dmat
INTEGER(sik) :: nTridiag, nTridiagC , nTridiagE , nSubNet
INTEGER(sik), PARAMETER :: bandWidth=3, nivars=5,            &
&     ibdiag = bandWidth/2 + 1
INTEGER(sik) :: nuvars=nivars
INTEGER(sik), ALLOCATABLE, TARGET, DIMENSION (:) :: splitRowsC,   &
&         splitRowsE
INTEGER(sik), POINTER, DIMENSION (:) :: splitRows
```

bandWidth - primary bandwidth of the matrix associated with the chosen finite difference method (fixed to 3 for the current method)

ibdiag - index of the element in component "a" (row coefficients) of a `sparseMatrix` data type that represents the diagonal element for that row

n3Dmat - number of 3D matrix blocks in the current full system of equations (either `n3DmatC` or `n3DmatE`)

n3DmatC - number of 3D matrix blocks in a cell-centered system of equations

n3DmatE - number of 3D matrix blocks in a cell-edge system of equations

nivars - number of independent variables associated with each computational volume

nrows1D - number of rows in the current 1D systemwide matrix (either `nrowsC` or `nrowsE`)

nrows3D - number of rows in the current 3D systemwide matrix

nrowsC1D - total number of 1D systemwide equations associated with the cell centers

nrowsE1D - total number of 1D systemwide equations associated with the cell edges

nrowsC - number of rows in a full cell-centered linear system (equal to the total number of system volumes).

nsplits - number of splitting rows in the current systemwide matrix (either `nsplitsC` or `nsplitsE`)

nsplitsC - number of rows that define splits between submatrixes in the cell-centered matrix (number of network junction variables)

nsplitsE - number of rows that define splits between submatrices in the cell

face matrix (number of network junction variables)

nSubNet - number of network submatrices in the system (dimension of `netMat` and `netRHS` derived-type arrays)

nTridiag - number of tridiagonal submatrices in the full system of equations (either `nTridiagC` or `nTridiagE`)

nTridiagC - number of tridiagonal submatrices in a full cell-centered system of equations

nTridiagE - number of tridiagonal submatrices in a full cell-edge system of equations

splitRows - an integer pointer array associated with either `splitRowsC` or `SplitRowsE`

splitRowsC - an integer array listing the rows in the 1D portion of a cell-centered system matrix that split pure 1D submatrices from each other

splitRowsE - an integer array listing the rows in the 1D portion of a cell-edge system matrix that split pure 1D submatrices from each other

Logical:

```
LOGICAL :: anyPack
```

anyPack - a logical scalar set to .TRUE if water packing occurs at any face in this timestep

DERIVED TYPE:

```
TYPE (adjEdgIndT), ALLOCATABLE :: adjEdg(:)
TYPE (momentumTransT), ALLOCATABLE :: momTran(:)
TYPE (sparseMatrixT), ALLOCATABLE, TARGET :: al(:), av(:)
TYPE (sparseMatrixT), POINTER :: at(:)
TYPE (sparseIndicesT), ALLOCATABLE, TARGET :: aIndC(:), aIndE(:)
TYPE (sparseIndicesT), POINTER :: atInd(:)
TYPE (vssMatrixT), ALLOCATABLE, TARGET :: a3Dl(:), a3Dv(:)
TYPE (vssMatrixT), ALLOCATABLE, TARGET :: a3DlE(:), a3DvE(:)
TYPE (vssMatrixT), POINTER :: a3D(:)
TYPE (vssMatIndT), ALLOCATABLE, TARGET :: i3DC(:), i3DE(:)
TYPE (vssMatIndT), POINTER :: i3D(:)
TYPE (blockMatrixT), ALLOCATABLE, TARGET :: blocks(:)
```

```
TYPE (triIndicesT), ALLOCATABLE, TARGET, DIMENSION(:) ::        &
& triBlock, iTridiagC, iTridiagE
TYPE (triIndicesT), POINTER :: iTrid(:)
TYPE (netVarIndT), ALLOCATABLE :: junVars(:), junCoef(:)

TYPE (netMatT), ALLOCATABLE, TARGET :: netlC(:), netgC(:),      &
&   netlE(:), netgE(:)
TYPE (netMatT), POINTER :: net(:)
TYPE (netMatIndT), ALLOCATABLE, TARGET :: netIndC(:), netIndE(:)
TYPE (netMatIndT), POINTER :: netInd(:)
TYPE (netRHST), ALLOCATABLE, TARGET, DIMENSION(:) :: araNet,    &
&   arcNET, arelNet, arevNet, arlNet, arvNet, pNet, vltNet,     &
&   vvtNet
TYPE (netRHST), POINTER :: rhsNet(:)
```

a3D - a pointer array of derived-type `vssMatrixT` associated with `a3Dl`, `a3Dg`, `a3DlE`, or `a3DgE`

a3Dl - a pointer array of derived-type `vssMatrixT` associated with the liquid-field 3D coefficient matrices

a3Dv - a pointer array of derived-type `vssMatrixT` associated with the gas-field 3D coefficient matrices

a3DlE - a pointer array of derived-type `vssMatrixT` associated with the liquid-velocity 3D coefficient matrices

a3DvE - a pointer array of derived-type `vssMatrixT` associated with the vapor-velocity 3D coefficient matrices

adjEdg - an allocated array of derived-type `adjEdjInd`, with each element corresponding to a system-edge variable index and components giving the sparse matrix indices for adjacent edges

aIndC - a pointer array of derived-type `sparseIndicesT` containing index information for cell-centered 1D matrices (`al` or `av`)

aIndE - a pointer array of derived-type `sparseIndicesT` containing index information for cell-edge 1D matrices (`al` or `av`)

al - a coefficient array for the full system's liquid stabilizer equations having derived-type `sparseMatrix` (same storage used for momentum and mass-energy equations)

at - a pointer array associated with `av` or `al`

atInd - a pointer array of derived-type `vssMatIndT` containing index information for `at` (points to either `aIndC` or `aIndE` as appropriate)

av - a coefficient array for the full system's gas stabilizer equations having derived-type `sparseMatrix` (same storage used for momentum and mass-energy equations)

araNet - an array of derived-type `netRHS` containing right-hand-side information for the noncondensable-gas stabilizer mass equations

arcNet - an array of derived-type `netRHS` containing right-hand-side information for the solute stabilizer mass equations

arelNet - an array of derived-type `netRHS` containing right-hand-side information for the liquid stabilizer energy equations

arevNet - an array of derived-type `netRHS` containing right-hand-side information for the gas stabilizer energy equations

arlNet - an array of derived-type `netRHS` containing right-hand-side information for the liquid stabilizer mass equations

arvNet - an array of derived-type `netRHS` containing right-hand-side information for the gas stabilizer mass equations

blocks - an array with derived-type `blockMatrix` containing one element for each computational volume in the problem

i3D - a pointer array of derived-type `vssMatIndT` associated with `i3DC`, `i3DE`

i3DC - a pointer array of derived-type `vssMatIndT` containing index information for cell-centered vessel matrices

i3DE - a pointer array of derived-type `vssMatIndT` containing index information for cell-edge vessel matrices

iTrid - a pointer associated with either `iTridiagC` or `iTridiagE`

iTridiagC - a `triIndices` derived-type array associated with cell-centered (mass-energy) stabilizer equations or semi-implicit pressure equations

iTridiagE - a `triIndices` derived-type array associated with cell-edge (momentum) stabilizer equations

junVars - an array of derived-type `netVarInd` containing index information

on the elements of "blocks" associated with the i^{th} network junction

momTran(i) - an array of derived-type `momentumTransT` containing side-connection, momentum-transfer information

net - a pointer array of derived-type `netMat` associated with either `netC` or `netE`

netgC - an array of derived-type `netMat` containing network matrices for cell-centered gas variables

netgE - a pointer array of derived-type `netMat` containing network matrices for cell-edge gas variables

netInd - a pointer array of derived-type `netMatInd` associated with either `netIndC` or `netIndE`

netIndC - a pointer array of derived-type `netMatInd` containing indices necessary to move information in and out of network matrices for cell-centered variables

netIndE - a pointer array of derived-type `netMatInd` containing indices necessary to move information in and out of network matrices for cell edge variables

netlC - an array of derived-type `netMat` containing network matrices for cell-centered liquid variables

netlE - a pointer array of derived-type `netMat` containing network matrices for cell-edge liquid variables

pNet - array of derived-type `netRHS` containing right-hand-side information for the basic pressure equations

rhs - a real pointer array associated with the appropriate array storing the right-hand side of an equation system currently being solved (contains the solution for the equations on completion of the solution procedure)

rhs3D - a `real` pointer array associated with the appropriate array storing the right-hand side of a 3D subsystem currently being solved (contains the solution for the equations on completion of the solution procedure)

rhsNet - a pointer array of derived-type `netRHST` associated with the array containing the current right-hand side for the system of network variable equations (points to `arcNET`, `arvNET`, etc., as needed)

triBlock - an array of derived-type `triIndices` containing information for submatrices within the array "blocks"

vltNet - array of derived-type `netRHS` containing right-hand-side information for the liquid stabilizer momentum equations

vvtNet - array of derived-type `netRHS` containing right-hand-side information for the gas stabilizer momentum equations

C.32. Module `Network`

NOTE: Most of the logic supported by module `Network` is no longer in TRAC. Currently, logic is retained only in the initialization stage to enforce a currently required restriction on 1D loop connections to VESSELs; i.e., all VESSEL connections in a given 1D loop must be to 3D faces of the same kind (all radial or x, all azimuthal or y, or all axial or z). Array `rnet` will be removed from TRAC after a planned parallel implementation of subroutine `Solver` is completed.

```
INTEGER(sik) ldrl,ldrv,ldrel,ldrev,ldra,ldrc
INTEGER(sik) ldpvc,laov,ldpvcv,lod,laol
!  This is a length, not a pointer
INTEGER(sik) lnldpv
REAL(sdk), DIMENSION(:),ALLOCATABLE :: rnet
```

REAL VARIABLES:

rnet - (Former) container array for network-solution arrays; see note above.

INTEGER VARIABLES:

ldrl - Variable to rework solution of ARL and VLT (contains right-hand side of linear equations).

ldrv - Variable to rework solution of ARV and VVT (contains right-hand side of linear equations).

ldrel - Storage for right-hand side of the liquid stabilizer equation.

ldrev - Storage for right-hand side of the vapor stabilizer equation.

ldra - Storage for right-hand side of the noncondensable stabilizer mass equation.

ldrc - Pointers for network variables for the solute-tracking option.

ldpvc - Locator that shows the beginning of coefficients to evaluate the derivatives of junction velocities with respect to VESSEL pressures.

laov - Variable to rework solution of ARV, AREV, and VVT (contains rework matrix).

ldpvcv	Pointer for reordered coupling coefficients between the VESSEL and the 1D network solution.
lod	Temporary storage for intercomponent coupling information.
laol	Variable to rework solution of ARL, AREL, and VLT (contains rework matrix).
lnldpv	Pointer variable for the network matrix equation right-hand-side vector.

C.33. Module `OneDDat`

```
REAL(sdk)      ardmin,arn,ary,alpst,c1a,c1av,c2a,c2av,ct,          &
& ctp,dvjp,fl1,fl2,fv1,fv2,havlv,qtp,r1l,r1v,r2l,r2v,s01,s02,salt,  &
& savt,ssac,sse,ssmc,ssmom,ssvc,ssve,vjs
 INTEGER(sik)          iacc2,ibks,icme,icorl,icoru                  &
&,il,iphsep,islb,isrb,ivpvlv,jstart,msc,nc2,nstg,ntee,njn          &
&,isflg,iclflg,lpindx
!
   DATA ardmin,arn,ary/0.0d0,0.0d0,1.0d0/
!
   INTEGER(sik) nosrce
!
   DATA nosrce/0/
!
!  Number of separator stages from sepcb.h
   INTEGER(sik) istagc
```

REAL VARIABLES:

ardmin	Minimum value of the difference between the flow-area ratios that are one mesh-cell distant from a junction interface with a PLENUM component and at the junction interface with a PLENUM component for flow from the PLENUM component.
arn	No factor for applying flow-area ratios in the momentum-convection term. 0.0 = apply area ratios; 1.0 = do not apply area ratios.
ary	Yes factor for applying flow-area ratios in the momentum-convection term. 1.0 = apply area ratios; 0.0 = do not apply area ratios.

alpst	The `jcell` fluid gas-volume fraction to be convected into the TEE side tube by the TEE offtake model.
c1a	Fraction of the liquid velocity at the left face of the TEE `jcell` that contributes to momentum transfer into the TEE side tube.
c1av	Fraction of the gas velocity at the left face of the TEE `jcell` that contributes to momentum transfer into the TEE side tube.
c2a	Fraction of the liquid velocity at the right face of the TEE `jcell` that contributes to momentum transfer into the TEE side tube.
c2av	Fraction of the gas velocity at the right face of the TEE `jcell` that contributes to momentum transfer into the TEE side tube.
ct	Momentum source coefficient.
ctp	AMAX1(0.0, COST).
dvjp	Pressure derivative of source velocity.
fl1	Temporary storage for liquid mass-flow corrections for mass-conservation checks at low-numbered cell face.
fl2	Temporary storage for liquid mass-flow corrections for mass-conservation checks at high-numbered cell face.
fv1	Temporary storage for gas mass-flow corrections for mass-conservation checks at low-numbered cell face.
fv2	Temporary storage for gas mass-flow corrections for mass-conservation checks at high-numbered cell face.
havlv	Temporary storage for the hydraulic diameter when the valve is open.
qtp	Total direct power input.
r1l	Coefficient of the SEPD or TEE side-tube coupled momentum-convection term at the left interface of JCELL for liquid.
r1v	Coefficient of the SEPD or TEE side-tube coupled momentum-convection term at the left interface of JCELL for gas.
r2l	Coefficient of the SEPD or TEE side-tube coupled momentum-convection term at the right interface of JCELL for liquid.
r2v	Coefficient of the SEPD or TEE side-tube coupled momentum-convection term at the right interface of JCELL for gas.
s01	Sign of IOU(1,current component).
s02	Sign of IOU(2,current component).
salt	Source term to liquid for compressible work.
savt	Source term to gas for compressible work.
ssac	Noncondensable-gas source.
sse	Energy source.
ssmc	Mass source.
ssmom	Momentum source to left-hand cell boundary.

ssvc	Gas mass source.
ssve	Gas energy source.
vjs	Source velocity.

INTEGER VARIABLES:

iacc2	Flag for PIPE to model an accumulator.
ibks	Indicator for network solution.
icme	Component index for referencing `iou` array.
icorl	Reactor-core-region lower boundary.
icoru	Reactor-core-region upper boundary.
il	Loop number index.
iphsep	Phase-separation evaluation flag of the TEE offtake model.
islb	Left-hand boundary switch.
isrb	Right-hand boundary switch.
ivpvlv	Interface number of the adjustable-valve flow area.
jstart	Cell number at the left end of the 1D segment.
msc	Cell number for the source terms.
nc2	Cell number that begins a SEPD or TEE side tube.
nstg	Variable not used.
ntee	Counter for a SEPD or TEE.
njn	Number of network matrix junctions.
isflg	Variable not used.
iclflg	Variable not used.
lpindx	Loop index that indicates the loop in the system.
nosrce	Option to turn off momentum-source coupling between the main tube and side tube of a SEPD or TEE component. 0 = evaluate momentum-source coupling; 1 = turn off momentum-source coupling.
istagc	Number of separator stages.

C.34. Module `PipeArray`

```
TYPE pipeArrayT
    REAL(sdk),  POINTER, DIMENSION(:)  :: powrf
    REAL(sdk),  POINTER, DIMENSION(:)  :: powtb
    REAL(sdk),  POINTER, DIMENSION(:)  :: qp3rf
    REAL(sdk),  POINTER, DIMENSION(:)  :: qp3tb
END TYPE pipeArrayT
!
```

TYPE (pipeArrayT),DIMENSION(maxComps) :: pipeAr

DERIVED TYPE VARIABLES:

pipeAr	Variables specific to the PIPE component.

REAL VARIABLES:

!Elements of DERIVED TYPE pipeArrayT

powrf	Rate-factor table for the power-deposited-in-the-coolant table.
powtb	Power-deposited-in-the-coolant table.
qp3rf	Rate-factor table for the QPPP-factor table.
qp3tb	QPPP-factor table.

C.35. Module `PipeVlt`

TYPE pipeTabT

REAL(sdk) bsmass
REAL(sdk) cpow
REAL(sdk) eninp
REAL(sdk) epsw
REAL(sdk) fl(2)
REAL(sdk) fv(2)
REAL(sdk) houtl
REAL(sdk) houtv
REAL(sdk) plent
REAL(sdk) powin
REAL(sdk) powoff
REAL(sdk) qint
REAL(sdk) qout
REAL(sdk) qp3in
REAL(sdk) qp3off
REAL(sdk) radin
REAL(sdk) rpowmx
REAL(sdk) rqp3mx
REAL(sdk) th
REAL(sdk) toutl
REAL(sdk) toutv
REAL(sdk) vflow
REAL(sdk) z
REAL(sdk) type1

```
    REAL(sdk) type2
    INTEGER(sik) iacc
    INTEGER(sik) ichf
    INTEGER(sik) icj1
    INTEGER(sik) icj2
    INTEGER(sik) iconc
    INTEGER(sik) ionoff
    INTEGER(sik) ipf
    INTEGER(sik) ipow
    INTEGER(sik) ipowsv
    INTEGER(sik) ipowtr
    INTEGER(sik) ipp
    INTEGER(sik) iqf
    INTEGER(sik) iqp
    INTEGER(sik) iqp3sv
    INTEGER(sik) iqp3tr
    INTEGER(sik) isollb
    INTEGER(sik) isolrb
    INTEGER(sik) js1
    INTEGER(sik) js2
    INTEGER(sik) jun1
    INTEGER(sik) jun2
    INTEGER(sik) ncells
    INTEGER(sik) nonoff
    INTEGER(sik) npowrf
    INTEGER(sik) npowsv
    INTEGER(sik) npowtb
    INTEGER(sik) nqp3rf
    INTEGER(sik) nqp3sv
    INTEGER(sik) nqp3tb
    INTEGER(sik) js1get
    INTEGER(sik) js1put
    INTEGER(sik) js2get
    INTEGER(sik) js2put
END TYPE pipeTabT
!
   TYPE(pipeTabT),DIMENSION(maxComps) :: pipeTab
```

!

INTEGER(sik) pipeDumpSize

PARAMETER (pipeDumpSize=60)

DERIVED TYPE VARIABLES:

pipeTAB	The VLT for the PIPE component.

REAL VARIABLES:

!Elements of DERIVED TYPE pipeTabT

bsmass	Time-integrated fluid mass flow from the PIPE.
cpow	Special PIPE power input.
eninp	Total (time-integrated) energy directly input to the PIPE.
epsw	Wall surface roughness.
fl(2)	Liquid mass-flow corrections for mass-conservation checks.
fv(2)	Gas mass-flow corrections for mass-conservation checks.
houtl	HTC between outer boundary of the PIPE wall and liquid.
houtv	HTC between outer boundary of the PIPE wall and gas.
plent	Total length of the PIPE.
powin	Initial power deposited in the liquid.
powoff	Power deposited in the liquid when the trip is off after it was on.
qint	Initial liquid volume in the PIPE.
qout	Volume of liquid that has been discharged from the PIPE modeled as an accumulator.
qp3in	Initial QPPP factor.
qp3off	QPPP factor when its trip is off after it was on.
radin	Inner radius of the PIPE wall.
rpowmx	Maximum rate of change of power deposited in the coolant.
rqp3mx	Maximum rate of change of the QPPP factor.
th	Thickness of the PIPE wall.
toutl	Liquid temperature outside the PIPE.
toutv	Gas temperature outside the PIPE.
vflow	Volume flow rate at fluid discharged from the PIPE modeled as an accumulator.
z	Water height above discharge.
type1	Type of adjacent component at `jun1`.
type2	Type of adjacent component at `jun2`.

INTEGER VARIABLES:

!Elements of DERIVED TYPE pipeTabT

iacc	PIPE modeled as an accumulator option.

ichf	CHF calculation option.
icj1	Variable not used.
icj2	Variable not used.
iconc	Presence of solute in the liquid option.
ionoff	Number of timesteps that the power-deposited-in-the-coolant trip has been on.
ipf	Last interpolated interval in the power-deposited-in-the-coolant's rate-factor table.
ipow	Presence of power deposited in the coolant option.
ipowsv	Signal-variable or control-block ID number defining the power-deposited-in-the-coolant table's abscissa-coordinate variable.
ipowtr	Trip ID number that controls the evaluation of the power-deposited-in-the-coolant table.
ipp	Last interpolated interval in the power-deposited-in-the-coolant table.
iqf	Last interpolated interval in the QPPP-factor table's rate-factor table.
iqp	Last interpolated interval in the QPPP-factor table.
iqp3sv	Signal-variable or control-block ID number defining the QPPP-factor table's abscissa-coordinate variable.
iqp3tr	Trip ID number that controls evaluation of the QPPP-factor table.
isollb	Indicator for velocity update at `jun1`.
isolrb	Indicator for velocity update at `jun2`.
js1	Junction sequence number at cell 1 of the PIPE.
js2	Junction sequence number at cell `ncells` of the PIPE.
jun1	Junction number at cell 1.
jun2	Junction number at cell `ncells`.
ncells	Number of fluid cells in the PIPE.
nonoff	Number of timesteps the QPPP-factor table's controlling trip has been on.
npowrf	Number of data pairs in the power-deposited-in-the-coolant table's rate-factor table.
npowsv	Signal-variable or control-block ID number defining the power-deposited-in-the-coolant rate-factor table's abscissa-coordinate variable.
npowtb	Number of data pairs in the power-deposited-in-the-coolant table.
nqp3rf	Number of data pairs in the QPPP-factor table's rate-factor table.

nqp3sv	Signal-variable or control-block ID number defining the QPPP-factor table's rate-factor table's abscissa-coordinate variable.
nqp3tb	Number of data pairs in the QPPP-factor table.
js1get	Index for getting information from bd array for left junction.
js1put	Index for putting information into bd array for left junction.
js2get	Index for getting information from bd array for right junction.
js2put	Index for putting information into bd array for right junction.

!

pipeDumpSize Number of variables in the derived-type pipeTabT.

C.36. Module PlenArray

```
TYPE plenumArrayT
    INTEGER(sik), POINTER, DIMENSION(:)   :: i0j
    INTEGER(sik), POINTER, DIMENSION(:)   :: jsn
    INTEGER(sik), POINTER, DIMENSION(:)   :: jsnget
    INTEGER(sik), POINTER, DIMENSION(:)   :: jsnput
    REAL(sdk),  POINTER, DIMENSION(:)     :: junj
    REAL(sdk),  POINTER, DIMENSION(:)     :: alw
    REAL(sdk),  POINTER, DIMENSION(:)     :: avw
    REAL(sdk),  POINTER, DIMENSION(:,:)   :: dbnd
    REAL(sdk),  POINTER, DIMENSION(:)     :: dnfl
    REAL(sdk),  POINTER, DIMENSION(:)     :: dnfv
    REAL(sdk),  POINTER, DIMENSION(:)     :: pak
    REAL(sdk),  POINTER, DIMENSION(:)     :: sgn
    REAL(sdk),  POINTER, DIMENSION(:)     :: dalp
    REAL(sdk),  POINTER, DIMENSION(:)     :: dxvol
    REAL(sdk),  POINTER, DIMENSION(:)     :: gravol
    REAL(sdk),  POINTER, DIMENSION(:)     :: fasmlt
    REAL(sdk),  POINTER, DIMENSION(:)     :: vrvpl
    REAL(sdk),  POINTER, DIMENSION(:)     :: vvvul
    REAL(sdk),  POINTER, DIMENSION(:)     :: vlvul
    REAL(sdk),  POINTER, DIMENSION(:)     :: favul
END TYPE plenumArrayT
!
TYPE (plenumArrayT),DIMENSION(maxComps) :: plenAr
```

DERIVED TYPE VARIABLES:

plenAr Variables specific to the PLENUM component.

REAL VARIABLES:

!Elements of DERIVED TYPE plenArrayT

junj	PLENUM junction numbers.
alw	Temporary storage for the right-hand side of the liquid stabilizer mass and energy equations.
avw	Temporary storage for the right-hand side of the gas stabilizer mass and energy equations.
dbnd	Donor-cell quantities $\alpha\rho_v$, $(1-\alpha)\rho_\ell$, $\alpha\rho_a$, $\alpha\rho_v e_v$, and $(1-\alpha)\rho_\ell e_\ell$
dnfl	Donor-cell flag for liquid. 0.0 = defines flow to the plenum cell; 1.0 = defines flow from the plenum cell.
dnfv	Donor-cell flag for gas. 0.0 = defines flow to the plenum cell; 1.0 = defines flow from the plenum cell.
pak	BIT array for the plenum junctions (used only for storing the water packing and stretching bits).
sgn	Junction flow-reversal indicators.
dalp	Weighting factor XVSET.
dxvol	Plenum-cell average length.
gravol	Plenum-cell average GRAV.
fasmlt	Interfacial-area multiplier during condensation.
vrvpl	Volume average relative velocity in the plenum cell.
vvvul	Plenum-cell average gas velocity.
vlvul	Plenum-cell average liquid velocity.
favul	Plenum-cell average flow area.

INTEGER VARIABLES:

!Elements of DERIVED TYPE plenArrayT

i0j	Network-junction numbers.
jsn	PLENUM junction-sequence numbers.
jsnget	
jsnput	

C.37. Module `PlenVlt`

TYPE plenTabT

REAL(sdk) bl

REAL(sdk) bsmass

REAL(sdk) bv

```
    REAL(sdk) fas1
    REAL(sdk) fas2
    REAL(sdk) flxa
    REAL(sdk) flxal
    REAL(sdk) flxav
    REAL(sdk) flxc
    REAL(sdk) flxel
    REAL(sdk) flxev
    REAL(sdk) flxl
    REAL(sdk) flxv
    REAL(sdk) rxcl
    REAL(sdk) rxcv
    REAL(sdk) xl
    REAL(sdk) xv
    INTEGER(sik) iconc
    INTEGER(sik) ipow
    INTEGER(sik) juns1
    INTEGER(sik) juns2
    INTEGER(sik) ncells
    INTEGER(sik) npljn
END TYPE plenTabT
!
   TYPE(plenTabT),DIMENSION(maxComps) :: plenTab
!
!  This is used to calculate the total dump size and must
!  be adjusted when changes are made to plenTab
!
INTEGER(sik) plenDumpSize
PARAMETER (plenDumpSize=23)
```

DERIVED TYPE VARIABLES:

plenTAB	The VLT for the PLENUM component.

REAL VARIABLES:

!Elements of DERIVED TYPE plenTabT

bl	Temporary storage for liquid mass-conservation checks.
bsmass	Time-integrated fluid mass flow from the plenum.
bv	Temporary storage for gas mass-conservation checks.
epsw	Wall surface roughness.

fas1	Summed flow area of all junctions on side 1 of the PLENUM cell.
fas2	Summed flow area of all junctions on side 2 of the PLENUM cell.
flxa	Total noncondensable-gas mass flow into the PLENUM cell during a timestep.
flxal	Total liquid volumetric flow into the PLENUM cell during a timestep.
flxav	Total gas volumetric flow into the PLENUM cell during a timestep.
flxc	Total solute mass flow into the PLENUM cell during a timestep.
flxel	Total liquid internal-energy flow into the PLENUM cell during a timestep.
flxev	Total gas internal-energy flow into the PLENUM cell during a timestep.
flxl	Total liquid mass flow into the PLENUM cell during a timestep.
flxv	Total gas mass flow into the PLENUM cell during a timestep.
rxcl	Temporary storage for the right-hand side of the liquid stabilizer mass and energy equations.
rxcv	Temporary storage for the right-hand side of the gas stabilizer mass and energy equations.
xl	Gross total liquid volumetric flow from the PLENUM cell during a timestep.
xv	Gross total gas volumetric flow from the PLENUM cell during a timestep.

INTEGER VARIABLES:

!Elements of DERIVED TYPE plenTabT

iconc	Presence of solute in the liquid option.
ipow	Presence of power-deposited-in-the-coolant option.
juns1	Number of junctions on side 1 of the PLENUM cell that convect momentum across the cell.
juns2	Number of junctions on side 2 of the PLENUM cell that convect momentum across the cell.
ncells	Number of fluid cells (1 for a PLENUM).
npljn	Number of plenum junctions.

!

plenDumpSize Number of variables in the derived-type plenTabT.

C.38. Module `Plenum`

```
PARAMETER (nvp=111)
PARAMETER (loldp=1,lold1p=24,lnewp=27)
```

!

PARAMETER (lentp=lnewp-loldp,lent1p=lold1p-loldp)

!

INTEGER(sik) currentPlenumInd

INTEGER VARIABLES:

nvp	Total number of PLENUM hydrodynamic database variables.
loldp	Index of first variable in the "old-time" class for which new-time values ARE calculated before the start of the OUTER hydrodynamic step.
lold1p	Index of first variable in the "old-time" class for which new-time values ARE NOT calculated before the start of the OUTER hydrodynamic step.
lnewp	Index of first variable in the "new-time" class for which new-time values ARE calculated before the start of the OUTER hydrodynamic step.
lentp	lnewp-loldp
lent1p	lold1p-loldp

currentPlenumInd

C.39. Module `PrizeVlt`

TYPE prizeTabT

REAL(sdk) bsmass
REAL(sdk) bsmssp
REAL(sdk) bxmass
REAL(sdk) dpmax
REAL(sdk) epsw
REAL(sdk) fl(2)
REAL(sdk) flow
REAL(sdk) fv(2)
REAL(sdk) houtl
REAL(sdk) houtv
REAL(sdk) pset
REAL(sdk) qheat
REAL(sdk) qin
REAL(sdk) qint
REAL(sdk) qout
REAL(sdk) qp3in
REAL(sdk) radin

```
    REAL(sdk) th
    REAL(sdk) toutl
    REAL(sdk) toutv
    REAL(sdk) z
    REAL(sdk) zhtr
    REAL(sdk) type1
    REAL(sdk) type2
    INTEGER(sik) ichf
    INTEGER(sik) iconc
    INTEGER(sik) ict1
    INTEGER(sik) iuv1
    INTEGER(sik) iuv2
    INTEGER(sik) js1
    INTEGER(sik) js2
    INTEGER(sik) jun1
    INTEGER(sik) jun2
    INTEGER(sik) ncells
    INTEGER(sik) js1get
    INTEGER(sik) js1put
    INTEGER(sik) js2get
    INTEGER(sik) js2put
END TYPE prizeTabT
!
TYPE(prizeTabT),DIMENSION(maxComps) :: prizeTab
!
INTEGER(sik) prizeDumpSize
PARAMETER (prizeDumpSize=40)
```

DERIVED TYPE VARIABLES:

prizeTAB	The VLT for the PRESSURIZER component.

REAL VARIABLES:

!Elements of DERIVED TYPE prizeTabT

bsmass	Time-integrated mass flow from pressurizer.
bsmssp	Current fluid mass-flow rate during a transient calculation.
bxmass	Current fluid mass-flow rate during a steady-state calculation.
dpmax	Differential pressure at which heaters have maximum power.
epsw	Wall surface roughness.
fl(2)	Liquid mass-flow corrections for mass-conservation checks.

flow	Volume flow rate at discharge.
fv(2)	Gas mass-flow corrections for mass-conservation checks.
houtl	HTC between outer boundary of pressurizer wall and liquid.
houtv	HTC between outer boundary of pressurizer wall and gas.
pset	Pressurizer pressure setpoint for heater-spray control.
qheat	Total heater power.
qin	Heater power being input to the liquid.
qint	Initial liquid volume in pressurizer.
qout	Volume of liquid that has been discharged from the pressurizer.
qp3in	Initial QPPP factor.
radin	Inner radius of pressurizer wall.
th	Thickness of pressurizer wall.
toutl	Liquid temperature outside the pressurizer.
toutv	Gas temperature outside the pressurizer.
z	Liquid height above discharge.
zhtr	Liquid height for heater cutoff.
type1	Variable not used.
type2	Variable not used.

INTEGER VARIABLES:

!Elements of DERIVED TYPE prizeTabT

ichf	CHF calculation option.
iconc	Presence of solute in the liquid option.
ict1	The sequence number (position in the `iorder` array) of the component next to the junction of the pressurizer (this variable is computed but not used).
iuv1	Indicator for velocity update at `jun1`.
iuv2	Indicator for velocity update at `jun2`.
js1	Junction sequence number at cell 1 of the pressurizer.
js2	Junction sequence number at cell `ncells` of the pressurizer.
jun1	Junction number at cell 1.
jun2	Junction number at cell `ncells`.
ncells	Number of fluid cells.
js1get	Index for getting information from `bd` array for left junction.
js1put	Index for putting information into `bd` array for left junction.
js2get	Index for getting information from `bd` array for right junction.
js2put	Index for putting information into `bd` array for right junction.

!

prizeDumpSize Number of variables in the derived-type `prizeTabT`.

C.40. Module `PumpArray`

```
TYPE pumpArrayT
    REAL(sdk),  POINTER, DIMENSION(:)  :: hsp1
    REAL(sdk),  POINTER, DIMENSION(:)  :: hsp2
    REAL(sdk),  POINTER, DIMENSION(:)  :: hsp3
    REAL(sdk),  POINTER, DIMENSION(:)  :: hsp4
    REAL(sdk),  POINTER, DIMENSION(:)  :: htp1
    REAL(sdk),  POINTER, DIMENSION(:)  :: htp2
    REAL(sdk),  POINTER, DIMENSION(:)  :: htp3
    REAL(sdk),  POINTER, DIMENSION(:)  :: htp4
    REAL(sdk),  POINTER, DIMENSION(:)  :: tsp1
    REAL(sdk),  POINTER, DIMENSION(:)  :: tsp2
    REAL(sdk),  POINTER, DIMENSION(:)  :: tsp3
    REAL(sdk),  POINTER, DIMENSION(:)  :: tsp4
    REAL(sdk),  POINTER, DIMENSION(:)  :: ttp1
    REAL(sdk),  POINTER, DIMENSION(:)  :: ttp2
    REAL(sdk),  POINTER, DIMENSION(:)  :: ttp3
    REAL(sdk),  POINTER, DIMENSION(:)  :: ttp4
    REAL(sdk),  POINTER, DIMENSION(:)  :: hdm
    REAL(sdk),  POINTER, DIMENSION(:)  :: tdm
    REAL(sdk),  POINTER, DIMENSION(:)  :: pmprf
    REAL(sdk),  POINTER, DIMENSION(:)  :: pmptb
    REAL(sdk),  POINTER, DIMENSION(:)  :: qp3rf
    REAL(sdk),  POINTER, DIMENSION(:)  :: qp3tb
    REAL(sdk),  POINTER, DIMENSION(:)  :: idxcs
END TYPE pumpArrayT
!
TYPE (pumpArrayT),DIMENSION(maxComps) :: pumpAr
```

DERIVED TYPE VARIABLES:

pumpAr Variables specific to the PUMP component.

REAL VARIABLES:

!Elements of DERIVED TYPE pumpArrayT

hsp1 Single-phase head curve 1.

hsp2 Single-phase head curve 2.

hsp3	Single-phase head curve 3.
hsp4	Single-phase head curve 4.
htp1	Two-phase head curve 1.
htp2	Two-phase head curve 2.
htp3	Two-phase head curve 3.
htp4	Two-phase head curve 4.
tsp1	Single-phase torque curve 1.
tsp2	Single-phase torque curve 2.
tsp3	Single-phase torque curve 3.
tsp4	Single-phase torque curve 4.
ttp1	Two-phase torque curve 1.
ttp2	Two-phase torque curve 2.
ttp3	Two-phase torque curve 3.
ttp4	Two-phase torque curve 4.
hdm	Head-degradation multiplier curve.
tdm	Torque-degradation multiplier curve.
pmprf	Rate-factor table for the pump-speed table.
pmptb	Pump-impeller, rotational-speed table.
qp3rf	Rate-factor table for the QPPP-factor table.
qp3tb	QPPP-factor table.
idxcs	Curve-set index array.

C.41. Module `PumpVlt`

TYPE pumpTabT
- REAL(sdk) aeffmi
- REAL(sdk) alpha
- REAL(sdk) alphao
- REAL(sdk) beffmi
- REAL(sdk) bsmass
- REAL(sdk) ceffmi
- REAL(sdk) delp
- REAL(sdk) dsmom
- REAL(sdk) effmi
- REAL(sdk) effmi1
- REAL(sdk) epsw
- REAL(sdk) fl(2)
- REAL(sdk) flow

REAL(sdk) fv(2)
REAL(sdk) head
REAL(sdk) houtl
REAL(sdk) houtv
REAL(sdk) mflow
REAL(sdk) omega
REAL(sdk) omegan
REAL(sdk) omgoff
REAL(sdk) omtest
REAL(sdk) qp3in
REAL(sdk) qp3off
REAL(sdk) radin
REAL(sdk) rflow
REAL(sdk) rhead
REAL(sdk) rho
REAL(sdk) romega
REAL(sdk) romgmx
REAL(sdk) rqp3mx
REAL(sdk) rrho
REAL(sdk) rtork
REAL(sdk) smom
REAL(sdk) tfr0
REAL(sdk) tfr1
REAL(sdk) tfr2
REAL(sdk) tfr3
REAL(sdk) tfrb
REAL(sdk) tfrl0
REAL(sdk) tfrl1
REAL(sdk) tfrl2
REAL(sdk) tfrl3
REAL(sdk) th
REAL(sdk) torque
REAL(sdk) toutl
REAL(sdk) toutv
REAL(sdk) type1
REAL(sdk) type2
INTEGER(sik) ichf

INTEGER(sik) icj1
INTEGER(sik) icj2
INTEGER(sik) iconc
INTEGER(sik) indxhm
INTEGER(sik) indxtm
INTEGER(sik) ionoff
INTEGER(sik) ipf
INTEGER(sik) ipm
INTEGER(sik) ipmps
INTEGER(sik) ipmps2
INTEGER(sik) ipmpsv
INTEGER(sik) ipmptr
INTEGER(sik) ipmpty
INTEGER(sik) ipp
INTEGER(sik) iqf
INTEGER(sik) iqp
INTEGER(sik) iqp3sv
INTEGER(sik) iqp3tr
INTEGER(sik) irp
INTEGER(sik) isol1
INTEGER(sik) isol2
INTEGER(sik) js1
INTEGER(sik) js2
INTEGER(sik) jun1
INTEGER(sik) jun2
INTEGER(sik) ncells
INTEGER(sik) ndata(16)
INTEGER(sik) ndmax
INTEGER(sik) nhdm
INTEGER(sik) nonoff
INTEGER(sik) npmprf
INTEGER(sik) npmpsd
INTEGER(sik) npmpsv
INTEGER(sik) npmptb
INTEGER(sik) nqp3rf
INTEGER(sik) nqp3sv
INTEGER(sik) nqp3tb

```
    INTEGER(sik) ntdm
    INTEGER(sik) option
    INTEGER(sik) js1get
    INTEGER(sik) js1put
    INTEGER(sik) js2get
    INTEGER(sik) js2put
END TYPE pumpTabT
!
TYPE(pumpTabT),DIMENSION(maxComps) :: pumpTab
!
INTEGER(sik) pumpDumpSize
PARAMETER (pumpDumpSize=110)
```

DERIVED TYPE VARIABLES:

pumpTAB The VLT for the PUMP component.

REAL VARIABLES:

!Elements of DERIVED TYPE pumpTabT

aeffmi	The coefficient for the (omega/romega)**2 term in the calculation of the variable moment of inertia ($kg \cdot m^2$).
alpha	Gas-volume fraction.
alphao	Gas-volume fraction used on previous timestep for the pump-head calculation.
beffmi	The coefficient for the (omega/romega) term in the calculation of the variable moment of inertia ($kg \cdot m^2$).
bsmass	Time-integrated mass flow from the PUMP.
ceffmi	The constant term in the calculation of the variable moment of inertia ($kg \cdot m^2$).
delp	Pressure difference across the pump-impeller interface.
dsmom	Derivative of the pump head with respect to velocity.
effmi	Moment of inertia.
effmi1	The alternate effective moment of inertia.
epsw	Wall surface roughness.
fl(2)	Liquid mass-flow corrections for mass-conservation checks.
flow	PUMP volumetric fluid-flow rate.
fv(2)	Gas mass-flow corrections for mass-conservation checks.
head	Pump head.
houtl	HTC between outer boundary of the PUMP wall and liquid.
houtv	HTC between outer boundary of the PUMP wall and gas.

mflow	PUMP fluid mass-flow rate.
omega	Pump-impeller rotational speed at old time.
omegan	Pump-impeller rotational speed at new time.
omgoff	Pump-impeller rotational speed when its controlling trip is off after it was on.
omtest	The pump-impeller rotational speed below which `effmi1` (the alternate effective moment of inertia) is used.
qp3in	Initial QPPP factor.
qp3off	QPPP factor when its controlling trip is off after it was on.
radin	Inner radius of wall.
rflow	Rated fluid flow.
rhead	Rated head.
rho	PUMP mixture density.
romega	Rated pump-impeller rotational speed.
romgmx	Maximum rate of change of the pump-impeller rotational speed.
rqp3mx	Maximum rate of change of the QPPP factor.
rrho	Rated fluid density.
rtork	Rated torque.
smom	Momentum source.
tfr0	Frictional-torque constant coefficient.
tfr1	Frictional-torque linear coefficient.
tfr2	Frictional-torque quadratic coefficient.
tfr3	Frictional-torque third-order coefficient.
tfrb	Pump-impeller rotational speed that defines the low-speed regime.
tfrl0	Low-speed, frictional-torque constant coefficient.
tfrl1	Low-speed, frictional-torque linear coefficient.
tfrl2	Low-speed, frictional-torque quadratic coefficient.
tfrl3	Low-speed, frictional-torque, third-order coefficient.
th	Wall thickness.
torque	Pump torque.
toutl	Liquid temperature outside the PUMP wall.
toutv	Gas temperature outside the PUMP wall.
type1	Type of adjacent component at `jun1`.
type2	Type of adjacent component at `jun2`.

INTEGER VARIABLES:

!Elements of DERIVED TYPE pumpTabT

ichf	CHF calculation option.

icj1	Variable not used.
icj2	Variable not used.
iconc	Presence of solute in the coolant option.
indxhm	Index on head-degradation multiplier curve.
indxtm	Index on torque-degradation multiplier curve.
ionoff	Number of timesteps that the pump-speed controlling trip has been on.
ipf	Last interpolated interval in the pump-speed table's rate-factor table.
ipm	Two-phase indicator. 0 = use single-phase curves; 1 = use two-phase curves.
ipmps	Flag that indicates whether the pump-impeller rotational speed previously has dropped below `omtest`. 0 = pump speed always has been greater than `omtest`; 1 = pump speed has dropped below `omtest` at some time.
ipmps2	Flag that indicates the evaluation of variable pump inertia in subroutine RPUMP.
ipmpsv	Signal-variable or control-block ID number defining the pump-speed table's independent variable.
ipmptr	PUMP trip ID number.
ipmpty	Pump-type number (1 or 2).
ipp	Last interpolated interval in the pump-speed table.
iqf	Last interpolated interval in the QPPP-factor table's rate-factor table.
iqp	Last interpolated interval in the QPPP-factor table.
iqp3sv	Signal-variable or control-block ID number defining the QPPP-factor table's rate-factor table's abscissa-coordinate variable.
iqp3tr	Trip ID number that controls evaluation of the QPPP-factor table.
irp	Reverse speed indicator. 0 = reverse speed not allowed; 1 = reverse speed allowed.
isol1	Indicator for velocity update at `jun1`.
isol2	Indicator for velocity update at `jun2`.
js1	Junction sequence number at cell 1 of the PUMP.
js2	Junction sequence number at cell `ncells` of the PUMP.
jun1	Junction number at cell 1.
jun2	Junction number at cell `ncells`.

ncells	Number of fluid cells.
ndata(16)	Number of sets of points in head and torque curves.
ndmax	Size of scratch storage array.
nhdm	Number of data pairs in the head-degradation multiplier curve.
nonoff	Number of timesteps that the QPPP-factor table's controlling trip has been on.
npmprf	The number of rate-factor table data pairs where the rate factor is applied to the pump-speed table's independent variable.
npmpsd	Signal-variable or control-block ID number defining the pump-impeller rotational speed when the pump-speed controlling trip is initially off.
npmpsv	Signal-variable or control-block ID number defining the pump-speed, rate-factor table's abscissa-coordinate variable.
npmptb	Number of data pairs in the pump-speed table.
nqp3rf	Number of data pairs in the QPPP-factor table's rate-factor table.
nqp3sv	Signal-variable or control-block ID number defining the QPPP-factor rate-factor table's abscissa-coordinate variable.
nqp3tb	Number of data pairs in the QPPP-factor table.
ntdm	Number of data pairs in the torque-degradation multiplier curve.
option	Pump-curve option.
js1get	Index for getting information from bd array for left junction.
js1put	Index for putting information into bd array for left junction.
js2get	Index for getting information from bd array for right junction.
js2put	Index for putting information into bd array for right junction.

!

pumpDumpSize Number of variables in the derived-type pumpTabT.

C.42. Module Restart

```
INTEGER(sik) ictrlr(8)
INTEGER(sik) dncomp,dlnflt
REAL(sdk) ddate,ddtime
!
INTEGER(sik) ictrld(8),dmpflg,nsdo
REAL(sdk) dmpint,tdump,ltdump
!
DATA ictrld/8*0/
DATA dmpflg/0/,nsdo/9999/
```

DATA tdump,dmpint/2*0.d0/

REAL VARIABLES:

ddate	Date the restart file was created.
ddtime	Time the restart file was created.
!	
dmpint	Dump interval for time domain.
ltdump	CPU time when last data dump was taken.
tdump	Calculation time when next data dump will be taken.

INTEGER VARIABLES:

dlnflt	Length of the FLTs read from the restart file.
dncomp	Number of components in the restart file.
ictrlr(8)	Array that contains buffering information about the restart file.
!	
dmpflg	Flag that signals whether the dump output file has been initialized. 0 = uninitialized; 1 = initialized.
ictrld	Array that contains buffering information about the dump output file.
nsdo	Timestep number of the last completed data dump.

C.43. Module `RodCrunch`

INTEGER(sik),PARAMETER :: sdkx=sdk

INTEGER VARIABLES:

sdkx	Required to declare real KIND functions with Nagware F90 compiler.

C.44. Module `RodGlobal`

INTEGER, DIMENSION(:),ALLOCATABLE :: wp

REAL(sdk), DIMENSION(:),ALLOCATABLE :: prptb

REAL VARIABLES:

prptb

INTEGER VARIABLES:

wp

C.45. Module RodHtcref1

```
REAL(sdk) alpag2(nxryt),alpcf2(nxryt),alprw(nxryt),alpsm      &
& (nxryt),alptb(nxryt),funh(nxryt),zags(nxryt),zchfl(nxryt),zdfs  &
& (nxryt),zrws(nxryt),zsms(nxryt),ztb(nxryt),nhsca(nxryt)
REAL(sdk) qchf,zslab
INTEGER(sik) ij,nnodes
```

REAL VARIABLES:

alpag2	Array of gas-volume fractions at the top of the agitated section for a given (r, θ) or (x, y) cell.
alpcf2	Array of gas-volume fractions at the CHF location for a given (r, θ) or (x, y) cell.
alprw	Array of gas-volume fractions at the top of the rough wavy section for a given (r, θ) or (x, y) cell.
alpsm	Array of gas-volume fractions at the top of the smooth section for a given (r, θ) or (x, y) cell.
alptb	Array of gas-volume fractions at the transition boiling location for a given (r, θ) or (x, y) cell.
funh	Array of the fraction of each heat-structure surface that is unheated.
zags	Array of the elevation where agitated inverted annular flow ends for a given (r, θ) or (x, y) cell.
zchfl	Array of the elevation of the CHF point for a given (r, θ) or (x, y) cell.
zdfs	Array of the elevation where highly dispersed flow begins for a given (r, θ) or (x, y) cell.
zrws	Array of the elevation where rough-wavy inverted annular flow ends for a given (r, θ) or (x, y) cell.
zsms	Array of the elevation where smooth inverted annular flow ends for a given (r, θ) or (x, y) cell.
ztb	Array of the elevation of the transition boiling point for a given (r, θ) or (x, y) cell.
qchf	CHF.
zslab	Elevation of the heat-transfer node being considered.
nhsca	Array of HTSTR-component numbers that defines the principal powered RODs or SLABs.

INTEGER VARIABLES:

ij	(r,θ) or (x, y) horizontal-plane, hydraulic-cell number.
nnodes	Number of nodes in a given ROD or SLAB.

C.46. Module `RodVlt`

TYPE rodTabT
- REAL(sdk) amh2
- REAL(sdk) bcr0
- REAL(sdk) bcr1
- REAL(sdk) beff
- REAL(sdk) bpp0
- REAL(sdk) bpp1
- REAL(sdk) drfb
- REAL(sdk) dri
- REAL(sdk) drio
- REAL(sdk) dtnht(2)
- REAL(sdk) dtpk
- REAL(sdk) dtxht(2)
- REAL(sdk) dznht
- REAL(sdk) eneff
- REAL(sdk) extsou
- REAL(sdk) fsi
- REAL(sdk) fso
- REAL(sdk) ftci
- REAL(sdk) ftcm
- REAL(sdk) ftco
- REAL(sdk) fucrac
- REAL(sdk) hdri
- REAL(sdk) hdro
- REAL(sdk) hgapo
- REAL(sdk) hli
- REAL(sdk) hlo
- REAL(sdk) hvi
- REAL(sdk) hvo
- REAL(sdk) pdrat
- REAL(sdk) pldr
- REAL(sdk) powexp
- REAL(sdk) reac
- REAL(sdk) reacn
- REAL(sdk) react

REAL(sdk) rmck
REAL(sdk) rmckn
REAL(sdk) rpower
REAL(sdk) rpowpf
REAL(sdk) rpowr
REAL(sdk) rpowri
REAL(sdk) rpowrn
REAL(sdk) rpowro
REAL(sdk) rpowto
REAL(sdk) rpwoff
REAL(sdk) rpwscl
REAL(sdk) rrpwmx
REAL(sdk) rzpwmx
REAL(sdk) sa(2)
REAL(sdk) saf
REAL(sdk) sdt
REAL(sdk) shelv
REAL(sdk) shtd
REAL(sdk) stimet
REAL(sdk) tk(3)
REAL(sdk) tli
REAL(sdk) tlo
REAL(sdk) tneut
REAL(sdk) tpowi
REAL(sdk) tpowo
REAL(sdk) tramax
REAL(sdk) trhmax
REAL(sdk) tvi
REAL(sdk) tvo
REAL(sdk) width
REAL(sdk) zpwin
REAL(sdk) zpwoff
REAL(sdk) zuptop
REAL(sdk) zupbot
REAL(sdk) zlptop
REAL(sdk) zlpbot
INTEGER(sik) iaf

INTEGER(sik) iaxcnd
INTEGER(sik) ibu(4)
INTEGER(sik) idbci
INTEGER(sik) idbco
INTEGER(sik) iext
INTEGER(sik) ionoff
INTEGER(sik) ipatch
INTEGER(sik) ipwdep
INTEGER(sik) ipwrad
INTEGER(sik) irc(4)
INTEGER(sik) ircjfm(4)
INTEGER(sik) ircjtb(4,4)
INTEGER(sik) irf
INTEGER(sik) irftr
INTEGER(sik) irftr2
INTEGER(sik) irp
INTEGER(sik) irpwsv
INTEGER(sik) irpwtr
INTEGER(sik) irpwty
INTEGER(sik) isnotb
INTEGER(sik) ittcs
INTEGER(sik) izf
INTEGER(sik) izp
INTEGER(sik) izpwsv
INTEGER(sik) izpwtr
INTEGER(sik) liqlev
INTEGER(sik) iis
INTEGER(sik) m1d
INTEGER(sik) m1dt
INTEGER(sik) ncrx
INTEGER(sik) ncrz
INTEGER(sik) ndg
INTEGER(sik) ndgx
INTEGER(sik) ndh
INTEGER(sik) ndhx
INTEGER(sik) nfbpwt
INTEGER(sik) nfci

```
    INTEGER(sik) nfcil
    INTEGER(sik) nhist
    INTEGER(sik) nint
    INTEGER(sik) nmwrx
    INTEGER(sik) nonoff
    INTEGER(sik) nopowr
    INTEGER(sik) nramax
    INTEGER(sik) nrfd
    INTEGER(sik) nrhmax
    INTEGER(sik) nridr
    INTEGER(sik) nrods
    INTEGER(sik) nrpwi
    INTEGER(sik) nrpwr
    INTEGER(sik) nrpwrf
    INTEGER(sik) nrpwsv
    INTEGER(sik) nrpwtb
    INTEGER(sik) nrts
    INTEGER(sik) nset
    INTEGER(sik) nset2
    INTEGER(sik) nzmax
    INTEGER(sik) nzpwrf
    INTEGER(sik) nzpwsv
    INTEGER(sik) nzpwtb
    INTEGER(sik) nzpwi
    INTEGER(sik) nzpwz
    INTEGER(sik) nzznhc
END TYPE rodTabT
!
TYPE(rodTabT),DIMENSION(maxComps) :: rodTab
!
INTEGER(sik) rodDumpSize
PARAMETER (rodDumpSize=163)
```

DERIVED TYPE VARIABLES:

rodTAB The VLT of the HTSTR component.

REAL VARIABLES:

!Elements of DERIVED TYPE rodTabT

amh2 Hydrogen mass generated from metal-water reaction.

bcr0	Zero-order coefficient of the first-order polynomial that defines the effective core-averaged concentration of control-rod-pin boron.
bcr1	First-order coefficient of the first-order polynomial that defines the effective core-averaged concentration of control-rod-pin boron.
beff	Total delayed-neutron fraction.
bpp0	Zero-order coefficient of the first-order polynomial that defines the effective core-averaged concentration of burnable-poison-pin boron.
bpp1	First-order coefficient of the first-order polynomial that defines the effective core-averaged concentration of burnable-poison-pin boron.
drfb	Reactivity-feedback change in K_{eff} over the last timestep.
dri	Estimated change in power or reactivity over the previous timestep.
drio	Old value of `dri`; the old value of the power or reactivity-estimated correction.
dtnht(2)	Delta temperature minimums used in the reflood calculation.
dtpk	Kaganove-method integration timestep for solving the point-reactor kinetics equations.
dtxht(2)	Delta temperature maximums used in the reflood calculation.
dznht	Delta Z_{min}.
eneff	Total decay-heat fraction.
extsou	Thermal power produced by external source neutrons in the reactor core.
fsi	Inner-surface area (or HTC) adjustment factor from a CSS type-5 controller.
fso	Outer-surface area (or HTC) adjustment factor from a CSS type-5 controller.
ftci	Inner-surface node thermal-conductivity adjustment factor from a CSS type-5 controller.
ftcm	Internal-nodes thermal-conductivity adjustment factor from a CSS type-5 controller.
ftco	Outer-surface node thermal-conductivity adjustment factor from a CSS type-5 controller.
fucrac	Fraction of uncracked fuel.
hdri	Thermal diameter (m) for the inner surface of the heat-structure ROD or SLAB element (used only when namelist variable `ithd` = 1).

hdro	Thermal diameter (m) for the outer surface of the heat-structure ROD or SLAB element (used only when namelist variable `ithd` = 1).
hgapo	Rod gap-conductance coefficient (for `matrd` = 3).
hli	Constant liquid HTC at the inner surface (used when the inner-surface boundary condition flag `idbci` = 1, indicating constant HTCs and external temperatures).
hlo	Constant liquid HTC at the outer surface (used when the outer-surface boundary condition flag `idbco` = 1, indicating constant HTCs and external temperatures).
hvi	Constant gas HTC at the inner surface (used when the inner-surface boundary condition flag `idbci` = 1, indicating constant HTCs and external temperatures).
hvo	Constant gas HTC at the outer surface (used when the outer-surface boundary condition flag `idbco` = 1, indicating constant HTCs and external temperatures).
pdrat	Rod pitch-to-diameter ratio.
pldr	Pellet dish radius. 0.0 = no pellet dish calculation; 1.0 = pellet dish calculation.
powexp	Exponent value to which the power distribution is raised to define the weighting function for averaging the reactivity-feedback parameters over the reactor-core volume.
qrdtot	Total rod heat flux.
reac	Reactivity feedback at the beginning of the previous timestep.
reacn	Reactivity-feedback estimate at the end of the present timestep.
react	Total reactivity at the beginning of the present timestep.
rmck	Reactor multiplication constant at the beginning of the present timestep.
rmckn	Reactor multiplication constant estimate at the end of the present timestep.
rpower	Average reactor-core power over the timestep.
rpowpf	Prompt-fission power.
rpowr	Beginning-of-timestep, reactor-core power.
rpowri	Initial reactor-core power.
rpowrn	End-of-timestep, reactor-core power.
rpowro	End-of-timestep, reactor-core power of the previous timestep.
rpowto	Beginning-of-timestep, reactor-core power of the previous timestep.

rpwoff	Programmed reactivity or reactor-core power when the controlling trip is off after it was on.
rpwscl	Reactivity-power-table scale factor for programmed reactivity or reactor-core power.
rrpwmx	Maximum rate of change of programmed reactivity or reactor-core power.
rzpwmx	Maximum rate of change of the axial power shape.
sa(2)	Values of the inner- and outer-surface areas (or HTCs) adjusted by a CSS type-5 controller.
saf	Adjustment factor evaluated by a CSS type-5 controller.
sdt	Time interval/s since the last reactivity change printout.
shelv	Axial elevation of the first (bottom) node row.
shtd	Numerical sign of the heat-transfer direction.
stimet	Problem time at which the last reactivity change was summed to variable storage for later printout.
tk(3)	Values of the inner-node, internal-node, and outer-node thermal conductivity adjusted by a CSS type-5 controller.
tli	Constant liquid temperature at the inner surface (used when the inner-surface, boundary-condition flag `idbci` = 1, indicating constant HTCs and external temperatures).
tlo	Constant liquid temperature at the outer surface (used when the outer-surface, boundary-condition flag `idbco` = 1, indicating constant HTCs and external temperatures).
tneut	Neutron generation time.
tpowi	Total power across the inner surface of the HTSTR.
tpowo	Total power across the outer surface of the HTSTR.
tramax	Average-rod, peak-cladding temperature.
trhmax	Maximum supplemental rod temperature.
tvi	Constant gas temperature at the inner surface (used when the inner-surface, boundary-condition flag `idbci` = 1, indicating constant HTCs and external temperatures).
tvo	Constant gas temperature at the outer surface (used when the outer-surface, boundary-condition flag `idbco` = 1, indicating constant HTCs and external temperatures).
width	Width of the SLAB surface (used to compute surface area).
zpwin	Axial-power-shape table's abscissa-coordinate variable value corresponding to the initial axial-power shape.
zpwoff	Axial-power-shape table's abscissa-coordinate variable value that corresponds to the axial-power shape that is used when the controlling trip is off after it was on.
zuptop	Axial location (m) of the top of the upper hot patch.

zupbot Axial location (m) of the bottom of the upper hot patch.

zlptop Axial location (m) of the top of the lower hot patch.

zlpbot Axial location (m) of the bottom of the lower hot patch.

INTEGER VARIABLES:

!Elements of DERIVED TYPE rodTabT

iaf

iaxcnd Axial conduction indicator.

- 0 = no axial heat-transfer conduction calculated;
- 1 = axial heat-transfer conduction calculated in the heat-structure ROD or SLAB element.

ibu(4) Boron-unit flag for the J^{th} reactivity coefficient.

idbci Boundary condition option for the inner surface of the heat-structure ROD or SLAB element.

- 0 = adiabatic boundary condition;
- 1 = constant HTCs and external temperatures;
- 2 = coupled to specified cells in one or more hydro components.

idbco Boundary condition option for the outer surface of the heat-structure ROD or SLAB element.

- 0 = adiabatic boundary condition;
- 1 = constant HTCs and external temperatures;
- 2 = coupled to specified cells in one or more hydro components.

iext Specifies if this HTSTR input was generated by the post-processor EXTRACT.

- 0 = no;
- 1 = yes.

ionoff Number of timesteps that the reactivity-power table's controlling trip has been on.

ipatch Hot-patch modeling indicator.

- 0 = no modeling;
- 1 = modeling of hot patches.

ipwdep Power-shape table-dependence option.

- –1 = power-shape-table power is defined by a signal variable or control block for each node with no normalization;
- 0 = power-shape-table independent variable is defined by signal-variable or control-block ID number `izpwsv`;
- 1 = power-shape-table power is defined by a signal variable

or control block for each node with normalization.

ipwrad — Spatial power-shape option.

- 0 = 1D axial-power-shape table;
- 1 = 2D axial-r or axial-x power-shape table.

irc(4) — Number of data values that define the fuel-temperature (1), coolant-temperature (2), gas-volume-fraction (3), and solute-mass-concentration (4) reactivity-coefficient tables.

ircjfm(4) — Form number of reactivity coefficient for the argument number reactivity-feedback parameter.

ircjtb(4,4) — Number of first argument reactivity-feedback parameter value entries for the second argument reactivity-coefficient table.

irf — Last interpolated interval number in rate-factor table for the reactivity-power table.

irftr — Trip ID number that controls the axial fine-mesh calculation.

irftr2 — Trip ID number for evaluating the core reflood model when namelist variable `newrfd` = 1.

irp — Last interpolated interval number in the reactivity-power table.

irpwsv — Signal-variable or control-block ID number defining the reactivity-power table's abscissa-coordinate variable.

irpwtr — Trip ID number that controls evaluation of the reactivity-power table.

irpwty — Neutronic point-reactor kinetics or reactor-core power option.

- 1 = point-reactor kinetics with constant programmed reactivity;
- 2 = point-reactor kinetics with table-defined programmed reactivity;
- 3 = point-reactor kinetics with trip-initiated constant programmed reactivity;
- 4 = point-reactor kinetics with initial constant programmed reactivity and trip-initiated, table-defined programmed reactivity;
- 5 = constant reactor-core power;
- 6 = table defined reactor-core power;
- 7 = initial constant reactor-core power with trip-initiated, table-defined, reactor-core power.

Add 10 to the above values to evaluate reactivity feedback.

isnotb — A flag variable that is defined if the solute is boron for the reactivity-feedback calculation.

- 0 = solute is boron;
- 1 = solute is not boron.

ittcs	Saved value of `ittc`, the specification of an external thermocouple on the ROD- or SLAB-element surface.
izf	Last interpolated interval number in the rate-factor table for the axial-power-shape table.
izp	Last interpolated interval number in the axial-power-shape table.
izpwsv	Signal-variable or control-block ID number defining the axial-power-shape table's abscissa-coordinate variable.
izpwtr	Trip ID number that controls evaluation of the axial-power-shape table.
liqlev	
iis	
m1d	Multiple 1D hydraulic-component coupling option.
m1dt	Type of HTSTR for the purpose of a neutronics calculation. 0 = Not part of a neutronics calculation. 1 = First HTSTR coupled to a neutronics calculation. 2 = Between the first and last HTSTR coupled to a neutronics calculation. 3 = Last HTSTR coupled to a neutronics calculation that evaluates the point-reactor kinetics calculation for all of the coupled HTSTRs.
ncrx	Number of average ROD or SLAB elements that affect fluid dynamics.
ncrz	Number of (course) axial intervals between temperature node rows.
ndg	Input-specified number of delated-neutron groups.
ndgx	Number of delayed-neutron groups.
ndh	Input-specified number of decay-heat groups.
ndhx	Number of decay-heat groups.
nfbpwt	Flag that defines the spatial distribution used to weight the averaging of the reactivity-feedback parameters over the reactor-core volume.
nfci	`fci` flag. 0 = no calculation; 1 = calculation.
nfcil	Limit on `fci` calculations per timestep.
nhist	Number of data pairs in the power-history table.
nint	Maximum possible number of interfaces between dissimilar materials in ROD or SLAB elements.

nmwrx — Metal-water reaction flag.

- 0 = no calculation;
- 1 = calculation.

nonoff — Number of timesteps that the trip-controlling evaluation of the axial-power-shape table has been on.

nopowr — Specification of whether a power source is present in the heat-structure ROD or SLAB element.

- 0 = power source present in the ROD or SLAB;
- 1 = no power source present in the ROD or SLAB.

nramax — Location of average-rod peak-cladding temperature used in the reflood calculation.

nrfd — Reflood flag.

- 0 = takes no action;
- 1 = turns on axial fine-mesh flag if it is off.

nrhmax — Location of the supplemental-rod, peak-cladding temperature.

nridr — Specification of the hydro-cell location that is coupled to the inner and/or outer surfaces of the heat-structure ROD or SLAB element.

- 0 = the `idrod` array is input for only the supplemental RODs or SLABs;
- 1 = the `idrod` array is input for all RODs or SLABs;
- 2 = the `idrod` array is input for all RODs or SLABs for both surfaces of the HTSTR.

nrods — Number of computational (average plus supplemental) rods, including "hot" rods. See `ncrx`.

nrpwi — Radial- or thickness-direction, power-shape integration option.

- −1 = histogram with step changes at the r or x locations;
- 0 = histogram with step changes midway between the r or x locations;
- 1 = trapezoidal integration.

nrpwr — Number of radial or thickness locations that define the 2D axial-r or axial-x power shape.

nrpwrf — Number of rate-factor table data pairs where the rate factor is applied to the power or reactivity table's independent variable.

nrpwsv — Signal-variable or control-block ID number defining the reactivity-power rate-factor table's abscissa-coordinate variable.

nrpwtb — Number of data pairs in the reactivity-power table.

nrts — Number of timesteps over which programmed reactivity and reactivity-feedback changes are summed for printout.

nset — Absolute value of the reflood axial fine-mesh trip set-status number during the previous timestep.

nset2 — Absolute value of the core-reflood trip set-status number.

nzmax — Maximum number of rows of heat-transfer nodes used in reflood calculation.

nzpwrf — Number of data pairs in the axial-power-shape, rate-factor table.

nzpwsv — Signal-variable or control-block ID number defining the axial-power-shape, rate-factor table's abscissa-coordinate variable.

nzpwtb — Number of axial power shapes in the axial-power-shape table.

nzpwi — Axial power-shape integration option for the heat-transfer calculation.

- –1 = histogram with step changes at the axial locations;
- 0 = histogram with step changes midway between the axial locations;
- 1 = trapezoidal integration.

nzpwz — Number of axial locations defining the axial-power shape.

nzznhc — Number of hydro-cell, axial-direction channels to which this powered HTSTR is coupled.

!

rodDumpSize — Number of variables in the derived-type `rodTabT`.

C.47. Module `SemiSolver`

Derived Types --- None

Variables

REAL

scaleDp — a real parameter set to the value 999,999, used to scale effects of pressure variations during water-packing corrections. This should be moved once a standard module (or set of modules) is established for storage of basic constants.

C.48. Module `Sepd`

```
TYPE sepdArrayT
    REAL(sdk),  POINTER, DIMENSION(:)  :: aa
    REAL(sdk),  POINTER, DIMENSION(:)  :: ads
    REAL(sdk),  POINTER, DIMENSION(:)  :: bb
    REAL(sdk),  POINTER, DIMENSION(:)  :: bd4
```

```
  REAL(sdk),  POINTER, DIMENSION(:)  :: cks
  REAL(sdk),  POINTER, DIMENSION(:)  :: dds
  REAL(sdk),  POINTER, DIMENSION(:)  :: effld
  REAL(sdk),  POINTER, DIMENSION(:)  :: hbs
  REAL(sdk),  POINTER, DIMENSION(:)  :: hsk
  REAL(sdk),  POINTER, DIMENSION(:)  :: rws
  REAL(sdk),  POINTER, DIMENSION(:)  :: rrs
 END TYPE sepdArrayT
!
 TYPE (sepdArrayT),DIMENSION(maxComps) :: sepdAr
!

   INTEGER(sik),PARAMETER,PRIVATE :: sdkx=sdk
```

DERIVED TYPE VARIABLES:

sepdAr	Arrays specific to the SEPD component.

REAL VARIABLES:

!Elements of DERIVED TYPE sepdArrayT

aa	Void profile coefficient inside water-layer radius.
ads	Flow area of discharge path.
bb	Void profile coefficient within water layer.
bd4	bd4-array pointer (boundary data for side tube at JCELL).
cks	Loss coefficient for discharge passage.
dds	Hydraulic diameter of discharge passage.
effld	Effective L/D coefficient at pick-off ring.
hbs	Length of the separator band.
hsk	Axial distance between discharge and swirling vane.
rws	Inner radius of separator wall.
rrs	Inner radius of the pickoff ring.

INTEGER VARIABLES:

sdkx	Required to declare real KIND functions with the Nagware F90 compiler.

C.49. Module `Sepdvlt`

```
 TYPE sepdTabT
     REAL(sdk) ai
     REAL(sdk) alpd
     REAL(sdk) alps
```

```
  REAL(sdk) an
  REAL(sdk) deldim
  REAL(sdk) dpsep
  REAL(sdk) dpss
  REAL(sdk) rh
  REAL(sdk) rr1
  REAL(sdk) theta
  REAL(sdk) vdryl
  REAL(sdk) vdryu
  REAL(sdk) wli0
  REAL(sdk) xco
  REAL(sdk) xcu
  INTEGER(sik) icbs1
  INTEGER(sik) icbs2
  INTEGER(sik) idry
  INTEGER(sik) isep
  INTEGER(sik) isoln
  INTEGER(sik) istage
  INTEGER(sik) ncsep
  INTEGER(sik) ndryr
  INTEGER(sik) nseps
 END TYPE sepdTabT
!
 TYPE(sepdTabT), DIMENSION(maxComps) :: sepdTab
!
 INTEGER(sik) sepdDumpSize
 PARAMETER (sepdDumpSize=24)
```

DERIVED TYPE VARIABLES:

sepdTab	The VLT for the SEPD component.

REAL VARIABLES:

!Elements of DERIVED TYPE sepdTabT

ai	Standpipe flow area.
alpd	`jcell` gas-volume fraction for the separator component.
alps	Side-arm separator gas-volume fraction for the SEPD component.
an	Separator nozzle exit area.

deldim	Constant in the dryer model (variable not used).
dpsep	Pressure drop across the separator.
dpss	Desired pressure drop across the separator.
rh	Radius of the separator hub at the inlet.
rr1	Radius of larger pickoff ring at first stage of two-stage separator.
theta	Angle between swirling vane and horizontal plane.
vdryl	Lower limit for dryer velocity (currently not available).
vdryu	Upper limit for dryer velocity (currently not available).
wli0	Liquid flow rate into the separator from the previous timestep.
xco	Carryover ratio of liquid mass flow to total mass flow.
xcu	Carryunder ratio of gas mass flow to total mass flow.

INTEGER VARIABLES:

!Elements of DERIVED TYPE teeTabT

icbs1	Control-block ID number that defines the separator carryover (the liquid mass flow divided by the total mass flow at the `jcell` + 1 interface).
icbs2	Control-block ID number that defines the separator carryunder (the gas mass flow divided by the total mass flow at the `jcell` + 1 interface).
idry	Dryer option flag (currently not available).
isep	Separator flag.
isoln	Advanced separator flag.
istage	Separator-type option.
ncsep	Separator flag.
ndryr	Dryer option flag (dryer not available).
nseps	Number of physical separators modeled.

sepdDumpSize Number of variables in the derived-type `sepdTabT`.

C.50. Module `SysConfig`

Derived Types

adjIndVolT	This provides the system volume indices for each of the two volumes adjacent to a face

Structure:

```
TYPE adjIndValT
 INTEGER(sik) p,m
END TYPE adjIndValT
```

INTEGER Components:

p - volume index on the "plus" (in direction of increasing volume index) side of the face

m - volume index on the "minus" (in direction of decreasing volume index) side of the face

junctionCellsT - For each component junction in any cell, this stores needed information about that junction, the associated cell, and that cell's component. For each junction between mesh segments, an array of this type has two entries: one for the cell on each side of the junction. A cell with multiple junctions will have multiple entries in the array of this type, one for each junction

Structure:

```
TYPE junctionCellsT
 INTEGER(sik):: ioc, icmp, compNum, cellNum, junNum, adjEdg
 INTEGER(sik):: vOutSign, otherSide, ncAdj, iEndAdj, iSeg
 INTEGER(sik) :: ivarC, ivarE, icDp
 INTEGER(sik) :: ix, iy, iz
 REAL(sdk) :: compType
 REAL(sdk) :: theta, phi, cosTheta, dist
 LOGICAL :: is3D, doEdge, side
END TYPE junctionCellsT
```

REAL Components:

compType - component type for the component containing this junction cell

cosTheta - cosine of the connecting angle (theta) between mesh segments at the junction

dist - the distance between the cell center and the junction face

phi - the angle (degrees) between an inwardly directed normal to the junction face and a reference vector perpendicular to the primary positive direction of motion within the component

theta - the angle (degrees) between an inwardly directed normal to the junction face and the primary positive direction of motion within the component

INTEGER Components:

adjEdg - system variable index at the volume edge "adjacent" to the junction face (next edge into this mesh segment along the normal to the junction face). As currently configured, a side junction has no such

adjacent edge, and the value is set to zero.

cellNum - cell number within the component for this junction cell

compNum - component number assigned in the input deck for the component containing this junction cell

icDp - index giving position of coefficient information in `blocks(ivar)%cDp`

icmp - component index in the `compSeg` array for the component containing this junction cell

ioc - integer giving the position of the component containing this junction cell within the computational order (will be obsolete when the code is compatible with parallel processing)

iEndAdj - index of the junction cell (element in the `junCells` array) associated with the opposite end of the 1D mesh in this component that is a continuation of the mesh associated with the other side of this junction. Set to zero if the mesh segment ends in this component.

iSeg - segment index for the `seg1D` or `seg3D` segment within `compSeg` that contains this junction cell

ivarC - system variable index associated with the center of this junction cell

ivarE - system variable index associated with the cell edge of this junction

ix - for a 3D cell, this gives the x (or radial) cell index

iy - for a 3D cell, this gives the y (or azimuthal cell index)

iz - for a 3D cell, this gives the axial cell index

jcTblOrd - denotes order in which the current junction cell is registered into the boundary pointer tables within the System Service. Provides ability to determine bounding indices in the pointer table for a given junction cell's information

junNum - junction number assigned in the input deck for this junction

ncAdj - number of cells adjacent to this junction (including this junction cell) within this component and along the continuation of the 1D mesh or 3D mesh direction associated with the junction face

otherSide - index of the junction cell (element in the `junCells` array) on the other side of this junction

vOutSign - the sign of the velocity associated with flow out from the cell through this junction face (+1 or -1).

LOGICAL Components:

doEdge - optional argument that when set to TRUE, gives the component containing this cell control over the evaluation of edge-based quantities at the junction

is3D - logical variable indicating whether the junction cell is in a 3D region

side - logical variable indicating whether this is a side junction

rangeT defines a range of integer indices

Structure:

```
TYPE rangeT
 INTEGER(sik) :: iLB, iUB
END TYPE rangeT
```

INTEGER Components:

iLB - lower bound of the range

iUB - upper bound of the range

segment1DT stores information on the extent of data segments in 1D regions

Structure:

```
TYPE segment1DT
 INTEGER(sik) :: cellLB, cellUB,  junLB, junUB
 INTEGER(sik) :: ivarLBC, ivarLBE, ivarUBC, ivarUBE
 INTEGER(sik) :: inc, iNet
 INTEGER(sik) :: junCellLB, junCellUB, matSeq, compNum
 INTEGER(sik) :: nSideJun
 INTEGER(sik), POINTER :: iSideJun(:)
 LOGICAL :: continuesLB, continuesUB
END TYPE segment1DT
```

INTEGER Components:

cellLB - component cell number at the lower boundary of the 1D segment

cellUB - component cell number at the upper boundary of the 1D segment

compNum - input component number containing this mesh segment

inc - increment between system variable indices within this segment (+1 or -1)

iNet - index of the subnetwork containing this segment

iSideJun - integer pointer array with length `nSideJun` containing the indices of the elements in the `junCells` array with information on the side junctions and associated junction cells for this 1D mesh segment

ivarLBC - system variable index for the cell center at the lower boundary of the 1D segment.

ivarUBC - system variable index for the cell center at the upper boundary of the 1D segment.

ivarLBE - system variable index for the cell edge at the lower boundary of the 1D segment. If stored as a negative number, the absolute value can be used to deduce variable indices between the lower and upper bounding mesh cells, but the true system-variable index for this cell must be obtained with an indirect reference to the array `junCells`

ivarUBE - system variable index for the cell edge at the upper boundary of the 1D segment. If stored as a negative number, the absolute value can be used to deduce variable indices between the lower and upper bounding mesh cells, but the true system-variable index for this cell must be obtained with an indirect reference to the array `junCells`

junCellLB - index of the element in the `junCells` array containing information on the junction and junction cell at the lower boundary of this contiguous mesh segment (set to zero if no such junction exists)

junCellUB - index of the element in the `junCells` array containing information on the junction and junction cell at the upper boundary of this contiguous mesh segment (set to zero if no such junction exists)

junLB - input junction number at the lower boundary of the 1D segment

junUB - input junction number at the upper boundary of the 1D segment

matSeq - sequential order in which this segment is incorporated into the full-system matrix

nSideJun - number of side junctions connected to this mesh segment

LOGICAL Components:

continuesLB - if TRUE, the mesh numbering continues sequentially beyond the lower bound of this mesh segment

continuesUB - if TRUE, the mesh numbering continues sequentially beyond the upper bound of this mesh segment

segment3DT stores information on the extent of data segments in 3D regions. This data structure and operations on it assume a logically rectangular mesh (with provisions for wraparound in a cylindrical geometry)

Structure:

```
TYPE segment3DT
 CHARACTER*16 :: geometry
 INTEGER(sik) :: njun, nx, ny, nz, ncells, compNum
 INTEGER(sik) :: nvarxE, nvaryE, nvarzE
 INTEGER(sik) :: ivarLBC, ivarUBC, ivarLBxE, ivarUBxE
 INTEGER(sik) :: ivarLByE, ivarUByE, ivarLBzE, ivarUBzE, matSeq
 INTEGER(sik), POINTER :: junInfo(:)
END TYPE segment3DT
```

INTEGER Components:

compNum - input component number containing this mesh segment

ivarLBC - system variable index for the first cell-centered variable in this 3D segment.

ivarLBxE - system variable index for the first cell radial- (x) edge variable in this 3D segment.

ivarLByE - system variable index for the first cell theta- (y) edge variable in this 3D segment.

ivarLBzE - system variable index for the first cell axial- (z) edge variable in this 3D segment.

ivarUBC - system variable index for the last cell-centered variable in this 3D segment.

ivarUBxE - system variable index for the last cell radial- (x) edge variable in this 3D segment.

ivarUByE - system variable index for the last cell theta- (y) edge variable in this 3D segment.

ivarUBzE - system variable index for the last cell axial- (z) edge variable in this 3D segment.

junInfo - a pointer array containing the indices of the elements in `junCells` containing information about the junctions connecting to the 3D region

matSeq - sequential order in which this segment is incorporated into the full-system matrix

ncells - number of computational volumes in this segment

njun - number of junctions to other mesh segments

nvarxE - number of variables at radial- (x) cell edges in this segment

nvaryE - number of variables at theta- (y) cell edges in this segment

nvarxE - number of variables at axial- (z) cell edges in this segment

nx - number of radial (r) or x cells

ny - number of azimuthal (theta) or y cells

nz - number of cells in the z direction

CHARACTER Components:

geometry - mesh geometry (either "cartesian" or "cylindrical")

segmentT an array with `ncomp` elements storing information on all mesh segments in each component

Structure:

```
TYPE segmentT
 INTEGER(sik) :: nSeg1D, nSeg3D
 TYPE (segment1DT), POINTER :: seg1D(:)
 TYPE (segment3DT), POINTER :: seg3D(:)
END TYPE segmentT
```

INTEGER Components:

nSeg1D - number of 1D mesh segments in this component

nSeg3D - number of 3D mesh segments in this component

DERIVED TYPE Components:

seg1D - an array with derived-type `segment1DT`, containing 1D mesh segment information for this component

seg3D - an array with derived-type `segment3DT`, containing 3D mesh segment information for this component

segUseT stores information about mesh segments being used. When implemented as an array, it stores the order in which the segments appear in the system-variable matrix

Structure:

```
TYPE segUseT
  INTEGER(sik) :: icmp, iseg, loop, junUsed
END TYPE segUseT
```

INTEGER Components:

icmp - component index [e.g., `gen1DAr(icmp)`] associated with this segment

iseg - segment index within its component

loop - index of the subnet (or loop) containing this segment

junUsed - tracks the side junctions already used to trace through the system as system-variable order is established

Variables

DERIVED TYPE:

```
TYPE (junctionCellsT), ALLOCATABLE, TARGET :: junCells(:)
TYPE (rangeT), ALLOCATABLE :: junComp(:)
TYPE (segmentT), ALLOCATABLE :: compSeg(:)
TYPE (segUseT), ALLOCATABLE :: segUse(:)
TYPE(adjIndValT), ALLOCATABLE :: iVol(:)
```

compSeg - an array with derived-type `segmentT`, with one element for each component in the system

iVol - an array with derived-type `adjVolT`, containing indices of volumes adjacent to all 1D cell edges

junCells - an array with derived-type `junctionCellsT`, containing information for all junction cells in the system. For each junction connecting mesh segments, there are two entries in `junCells`, one for each of the two cells connected by the junction face.

junComp - an array with type `rangeT`. For a given component index (`cco`), the element in `junComp` with that index gives the range of indices in `junCells` of junction information for that component

segUse - an array with derived-type `segUseT`, with one element for each mesh segment in the system, storing the order in which the segments appear in the system-variable matrix

INTEGER:

```
INTEGER(sik), PRIVATE :: intJunNum=0
INTEGER(sik) :: nJunCells=0, nSeg1D, nSeg3D, nStarts
INTEGER(sik), PRIVATE :: iCompSeg=0, iSeg1D, iSeg3D
INTEGER(sik) :: nVol1D, nVol3D, nVolTot, nEdg1D, nEdg3D, nEdgTot
INTEGER(sik) :: nEdg3Dx, nEdg3Dy, nEdg3Dz
INTEGER(sik), PRIVATE :: ivLastx, ivLasty, ivLastZ
INTEGER, ALLOCATABLE, DIMENSION(:) :: cmpInd, segInd
```

intJunNum - integer tracking the last value of an interior junction number assigned by the function `InteriorJunNum`

iSeg1D - counter of the number of 1D mesh segments processed

iSeg3D - counter of the number of 3D mesh segments processed

nJunCells - number of active elements in the `junCells` (or `junAvg`) array. Equal to twice the number of junctions between mesh segments.

nEdg1D - total number of cell edges in 1D regions (gives a count of the 1D cell-edge variables)

nEdg3D - total number of cell edges in 3D regions (gives a count of the 3D cell-edge variables)

nEdg3Dx - total number of x or radially facing cell edges in all 3D regions

nEdg3Dy - total number of y or azimuthally facing cell edges in all 3D regions

nEdg3Dz - total number of z axially facing cell edges in all 3D regions

nEdgTot - total number of cell edges in the system (= `nEdg1D` + `nEdg3D`)

nSeg1D - total number of 1D mesh segments in the calculation

nSeg3D - total number of 3D mesh segments in the calculation

nStarts - total number of times that a trace through a contiguous 1D mesh was started. This is the number of subnetworks ("loops") in the system

nVol1D - total number of computational volumes in 1D regions (gives a count of the 1D cell-centered variables)

nVol3D - total number of computational volumes in 3D regions (gives a count of the 3D cell-centered variables)

nVolTot - total number of computational volumes in the system (= `nVol1D` + `nVol3D`)

C.51. Module `SysService`

Derived Types:

TransferRealT a table containing information used to transfer real variables between specified locations

Structure:

```
TYPE transferRealT
  REAL(sdk), POINTER :: from, to
  LOGICAL :: flipSign
END TYPE transferRealT
```

REAL Components:

from - real pointer to the memory location that contains data that are to be copied

to - real pointer to the memory location to which the "from" data are to be copied into

LOGICAL Components:

flipSign - logical variable indicating whether the data that are being copied are to be flipped in sign

TransferIntT table containing a information used to transfer `INTEGER` variables between specified locations

Structure:

```
TYPE transferIntT
 INTEGER(sik), POINTER :: from, to
 LOGICAL :: flipSign
END TYPE transferIntT
```

INTEGER Components:

from - integer pointer to the memory location that contains data that are to be copied

to - integer pointer to the memory location to which the "from" data are to be copied into

LOGICAL Components:

flipSign - logical variable indicating whether the data that are being copied are to be flipped in sign. Needed when adjacent components have different sign conventions at the connecting face

Variables:

REAL:

```
REAL(sdk), TARGET :: nul = 0.0_sdk
REAL(sdk), TARGET,DIMENSION(:),ALLOCATABLE :: vSign
REAL(sdk), TARGET,DIMENSION(:),ALLOCATABLE,PRIVATE :: xnum
```

nul - target variable set to 0.0 so that System Service table pointers have a place to point to when a zero is always transferred (used to set certain bd array values to 0)

vSign - this array contains the same information as the `vsi` array in `icomp`

xnum - this real array contains input component numbers

INTEGER:

```
INTEGER(sik) :: tableOrder = 0, bdIndex = 0
INTEGER(sik), PRIVATE:: ccoAdjJun, otherSide, jco
INTEGER(sik), PRIVATE :: cellIndex, face1Index
INTEGER(sik), PRIVATE :: face1AdjIndex, face2Index
INTEGER(sik), PRIVATE :: ix, jy, kz, ixf2, jyf2, kzf2, is
```

bdIndex - counter index for keeping track of current location in the pointer table array

cellIndex - index to the element in the g1DAr component array that contains the information for the junction cell adjacent to the cell being operated upon

ccoAdjJun - cco index for the adjacent junction cell.

face1AdjIndex - index used in the special logic for plenums to get at the information for the current junction, but via the current component (normally face1Index gets that information via the component on the opposite side of the junction being operated upon. This value is needed only for the PLENUM special logic because the PLENUM data structure has no knowledge of face information for its own junctions (because it does not control the calculation of these terms)

face1Index - index to the element in the g1DAr component array for the first face away from the current junction cell (this is just the junction face itself)

face2Index - index to the element in the g1DAr component array for the second face away from the current junction cell

jco - for a given junction cell, index to the element in the junCells array containing that junction cell's information jco stands for junction cell order.

otherSide - for a given junction cell, index to the element in the junCells array containing the opposing junction cell's information

tableOrder - order that current junction cell is registered into pointer table

LOGICAL:

```
LOGICAL, PRIVATE :: flipSign, isSideJunCell, isPlenum
LOGICAL, PRIVATE :: is3D, isAdjJcell
```

flipSign - logical variable indicating whether the boundary data that are being initialized are to be flipped in sign

isSideJunCell - logical variable indicating whether the junction cell being operated on is an internal TEE junction (on the side-tube side)

isPlenum - logical variable indicating whether the junction cell being operated on is next to a PLENUM

is3D - logical variable indicating whether the current junction cell sits next

to a 3D region

isAdjJcell - logical variable indicating whether the current junction cell sits next to the `jcell` of a TEE

NOTE: In post-3.0 versions, variable `isPlenum` is renamed `isAdjPlenum`, and variable `is3D` is renamed `isAdj3D`.

Derived TYPE:

TYPE (transferRealT), DIMENSION(:), ALLOCATABLE :: table

table - pointer table used for transferring data to and from the `bd` array

C.52. Module `SysTime`

INTEGER(sik),PARAMETER,PRIVATE :: sikx=sik
INTEGER(sik),PARAMETER,PRIVATE :: sdkx=sdk
!
INTEGER(sik) countInit,countRate,countNow,count1,count2
INTEGER(sik) :: clockCount=0

INTEGER VARIABLES:

sikx	Required to declare integer KIND functions with the Nagware F90 compiler.
sdkx	Required to declare real KIND functions with the Nagware F90 compiler.
countInit	
countRate	
countNow	
count1	
count2	
clockCount	

C.53. Module `Tee`

INTEGER(sik),PARAMETER,PRIVATE :: sdkx=sdk

INTEGER VARIABLES:

sdkx	Required to declare real KIND functions with the Nagware F90 compiler.

C.54. Module `TeeArray`

```
REAL(sdk),   DIMENSION(nbd)  :: bd4
   TYPE teeArrayT
    REAL(sdk),   POINTER, DIMENSION(:)  :: bd4
    REAL(sdk),   POINTER, DIMENSION(:)  :: powrf
    REAL(sdk),   POINTER, DIMENSION(:)  :: powtb
    REAL(sdk),   POINTER, DIMENSION(:)  :: qp3rf
    REAL(sdk),   POINTER, DIMENSION(:)  :: qp3tb
   END TYPE teeArrayT
!
   TYPE (teeArrayT),DIMENSION(maxComps) :: teeAr
```

DERIVED TYPE VARIABLES:

teeAr	Arrays specific to the TEE component.

REAL VARIABLES:

bd4	Holds boundary data for TEE side tube at JCELL.
!Elements of DERIVED TYPE teeArrayT	
bd4	bd4-array pointer.
powrf	Rate-factor table for the power-deposited-in-the-coolant table.
powtb	Power-deposited-in-the-coolant table.
qp3rf	Rate-factor table for the QPPP-factor table.
qp3tb	QPPP-factor table.

C.55. Module `TeeVlt` <<<--- update numbers from here on

```
TYPE teeTabT
    REAL(sdk) ai
    REAL(sdk) alpot
    REAL(sdk) alpotn
    REAL(sdk) alpoto
    REAL(sdk) bsmass
    REAL(sdk) ca
    REAL(sdk) ca1
    REAL(sdk) ca1v
    REAL(sdk) cav
    REAL(sdk) cost
    REAL(sdk) epsw
    REAL(sdk) enin1
```

REAL(sdk) enin2
REAL(sdk) fl(4)
REAL(sdk) fv(4)
REAL(sdk) houtl1
REAL(sdk) houtl2
REAL(sdk) houtv1
REAL(sdk) houtv2
REAL(sdk) powr1
REAL(sdk) powr2
REAL(sdk) pwin1
REAL(sdk) pwin2
REAL(sdk) pwoff1
REAL(sdk) pwoff2
REAL(sdk) qpin1
REAL(sdk) qpin2
REAL(sdk) qpoff1
REAL(sdk) qpoff2
REAL(sdk) radin1
REAL(sdk) radin2
REAL(sdk) rpwmx1
REAL(sdk) rpwmx2
REAL(sdk) rqpmx1
REAL(sdk) rqpmx2
REAL(sdk) rt1l
REAL(sdk) rt1v
REAL(sdk) rt2l
REAL(sdk) rt2v
REAL(sdk) th1
REAL(sdk) th2
REAL(sdk) tlen1
REAL(sdk) tlen2
REAL(sdk) toutl1
REAL(sdk) toutl2
REAL(sdk) toutv1
REAL(sdk) toutv2
REAL(sdk) type1
REAL(sdk) type2

REAL(sdk) type3
INTEGER(sik) ichf
INTEGER(sik) icj1
INTEGER(sik) icj2
INTEGER(sik) icj3
INTEGER(sik) iconc1
INTEGER(sik) iconc2
INTEGER(sik) ientrn
INTEGER(sik) ionof1
INTEGER(sik) ionof2
INTEGER(sik) ipf1
INTEGER(sik) ipf2
INTEGER(sik) ipow1
INTEGER(sik) ipow2
INTEGER(sik) ipp1
INTEGER(sik) ipp2
INTEGER(sik) ipwsv1
INTEGER(sik) ipwsv2
INTEGER(sik) ipwtr1
INTEGER(sik) ipwtr2
INTEGER(sik) iqf1
INTEGER(sik) iqf2
INTEGER(sik) iqp1
INTEGER(sik) iqp2
INTEGER(sik) iqpsv1
INTEGER(sik) iqpsv2
INTEGER(sik) iqptr1
INTEGER(sik) iqptr2
INTEGER(sik) isol1
INTEGER(sik) isol2
INTEGER(sik) isol3
INTEGER(sik) jcell
INTEGER(sik) js1
INTEGER(sik) js2
INTEGER(sik) js3
INTEGER(sik) jun1
INTEGER(sik) jun2

```
    INTEGER(sik) jun3
    INTEGER(sik) ncell1
    INTEGER(sik) ncell2
    INTEGER(sik) ncells
    INTEGER(sik) nonof1
    INTEGER(sik) nonof2
    INTEGER(sik) npwrf1
    INTEGER(sik) npwrf2
    INTEGER(sik) npwsv1
    INTEGER(sik) npwsv2
    INTEGER(sik) npwtb1
    INTEGER(sik) npwtb2
    INTEGER(sik) nqprf1
    INTEGER(sik) nqprf2
    INTEGER(sik) nqpsv1
    INTEGER(sik) nqpsv2
    INTEGER(sik) nqptb1
    INTEGER(sik) nqptb2
    INTEGER(sik) js1get
    INTEGER(sik) js1put
    INTEGER(sik) js2get
    INTEGER(sik) js2put
    INTEGER(sik) js3get
    INTEGER(sik) js3put
  END TYPE teeTabT
!

  TYPE teeJcellT
    REAL(sdk), DIMENSION(2) :: coeffVltn
    REAL(sdk), DIMENSION(2) :: coeffVvtn
    REAL(sdk), DIMENSION(2) :: thirdVltn
    REAL(sdk), DIMENSION(2) :: thirdVvtn
    REAL(sdk), DIMENSION(2) :: ellFlag
  END TYPE teeJcellT

  TYPE(teeTabT), DIMENSION(maxComps) :: teeTab
  TYPE(teeJcellT), DIMENSION(maxComps), TARGET :: teeJCellAr
```

!

INTEGER(sik) teeDumpSize
PARAMETER (teeDumpSize=110)

DERIVED TYPE VARIABLES:

teeTAB	The VLT for the TEE component.
teeJCellAr	Stores TEE momentum source coefficients for "ell" (or "I") configuration; acts as target location for certain bd-array elements.

REAL VARIABLES:

!Elements of DERIVED TYPE teeTabT

ai	Standpipe flow area.
alpot	Old offtake gas-volume fraction.
alpotn	New offtake gas-volume fraction.
alpoto	Old-old offtake gas-volume fraction.
bsmass	Time-integrated fluid mass flow from the SEPD or TEE.
ca	Fraction of the liquid velocity at the left face of `jcell` that contributes to the momentum transfer into the SEPD or TEE side tube.
ca1	Fraction of the liquid velocity at the right face of `jcell` that contributes to the momentum transfer into the SEPD or TEE side tube.
ca1v	Fraction of the gas velocity at the right face of `jcell` that contributes to the momentum transfer into the SEPD or TEE side tube.
cav	Fraction of the gas velocity at the left face of jcell that contributes to the momentum transfer into the SEPD or TEE side tube.
cost	Cosine of the angle between the low-numbered cells of the SEPD or TEE main tube and side tube.
epsw	Wall surface roughness.
fl(4)	Liquid mass-flow corrections for mass-conservation checks.
fv(4)	Gas mass-flow corrections for mass-conservation checks.
houtl1	HTC to liquid at the outer boundary of the SEPD or TEE main-tube wall.
houtl2	HTC to liquid at the outer boundary of the SEPD or TEE side-tube wall.
houtv1	HTC to gas at the outer boundary of the SEPD or TEE main-tube wall.
houtv2	HTC to gas at the outer boundary of the SEPD or TEE side-tube wall.

powr1	Power per length added to the SEPD or TEE main tube.
powr2	Power per length added to the SEPD or TEE side tube.
pwin1	Initial power deposited in the coolant of the SEPD or TEE main tube.
pwin2	Initial power deposited in the coolant of the SEPD or TEE side tube.
pwoff1	Power deposited in the coolant of the SEPD or TEE main tube when its controlling trip is off after it was on.
pwoff2	Power deposited in the coolant of the SEPD or TEE side tube when its controlling trip is off after it was on.
qpin1	Initial QPPP factor for the SEPD or TEE main tube.
qpin2	Initial QPPP factor for the SEPD or TEE side tube.
qpoff1	QPPP factor for the SEPD or TEE main tube when its controlling trip is off after it was on.
qpoff2	QPPP factor for the SEPD or TEE side tube when its controlling trip is off after it was on.
radin1	Inner radius of the SEPD or TEE main tube.
radin2	Inner radius of the SEPD or TEE side tube.
rpwmx1	Maximum rate of change of power deposited in the coolant for the SEPD or TEE main tube.
rpwmx2	Maximum rate of change of power deposited in the coolant for the SEPD or TEE side tube.
rqpmx1	Maximum rate of change of the QPPP factor for the SEPD or TEE main tube.
rqpmx2	Maximum rate of change of the QPPP factor for the SEPD or TEE side tube.
rt1l	Coefficient of the SEPD or TEE side-tube coupled momentum-convection term at the left interface of `jcell` for liquid.
rt1v	Coefficient of the SEPD or TEE side-tube coupled momentum-convection term at the left interface of `jcell` for gas.
rt2l	Coefficient of the SEPD or TEE side-tube coupled momentum-convection term at the right interface of `jcell` for liquid.
rt2v	Coefficient of the SEPD or TEE side-tube coupled momentum-convection term at the right interface of `jcell` for gas.
th1	Wall thickness of the SEPD or TEE main tube.
th2	Wall thickness of the SEPD or TEE side tube.
tlen1	Length of the SEPD or TEE main tube.
tlen2	Length of the SEPD or TEE side tube.
toutl1	Temperature of liquid outside the SEPD or TEE main-tube wall.
toutl2	Temperature of liquid outside the SEPD or TEE side-tube wall.

toutv1 Temperature of gas outside the SEPD or TEE main-tube wall.

toutv2 Temperature of gas outside the SEPD or TEE side-tube wall.

type1 Type of adjacent component at `jun1`.

type2 Type of adjacent component at `jun2`.

type3 Type of adjacent component at `jun2`.

!Elements of DERIVED TYPE teeJcellT

Boundary data at left face of primary tube:

coeffVltn(1) Coefficient of VLTN(JCELL+1) in momentum source term for VLTN(JCELL).

coeffVvtn(1) Coefficient of VVTN(JCELL+1) in momentum source term for VVTN(JCELL).

thirdVltn(1) Third term in momentum source term for VLTN(JCELL).

thirdVvtn(1) Third term in momentum source term for VVTN(JCELL).

ellFlag(1) Flag indicating that an ELL is present

Boundary data at right face of primary tube:

coeffVltn(2) Coefficient of VLTN(JCELL) in momentum source term for VLTN(JCELL+1).

coeffVvtn(2) Coefficient of VVTN(JCELL) in momentum source term for VVTN(JCELL+1).

thirdVltn(2) Third term in momentum source term for VLTN(JCELL+1).

thirdVvtn(2) Third term in momentum source term for VVTN(JCELL+1).

ellFlag(2) Flag indicating an ELL is present

INTEGER VARIABLES:

!Elements of DERIVED TYPE teeTabT

ichf CHF calculation option.

icj1 Iteration index of adjacent component to SEPD or TEE at `jun1`.

icj2 Iteration index of adjacent component to SEPD or TEE at `jun2`.

icj3 Iteration index of adjacent component to SEPD or TEE at `jun3`.

iconc1 Indicator for presence of solute in the liquid of the SEPD or TEE main tube.

iconc2 Indicator for presence of solute in the liquid of the SEPD or TEE side tube.

ientrn Offtake model option.

0 = model OFF;

1 = model ON (side-tube, internal-junction mass flow determined using offtake model).

ionof1	Number of timesteps that the power-deposited-in-the-coolant table for the SEPD or TEE main tube's controlling trip has been on.
ionof2	Number of timesteps that the power-deposited-in-the-coolant table for the SEPD or TEE side tube's controlling trip has been on.
ipf1	Last interpolated interval number of the rate-factor table for the power-deposited-in-the-coolant table of the SEPD or TEE main tube.
ipf2	Last interpolated interval number of the rate-factor table for the power-deposited-in-the-coolant table of the SEPD or TEE side tube.
ipow1	Presence of power-deposited-in-the-coolant option for the SEPD or TEE main tube.
ipow2	Presence of power-deposited-in-the-coolant option for the SEPD or TEE side tube.
ipp1	Last interpolated interval number of the power-deposited-in-the-coolant table for the SEPD or TEE main tube.
ipp2	Last interpolated interval number of the power-deposited-in-the-coolant table for the SEPD or TEE side tube.
ipwsv1	Signal-variable or control-block ID number defining the power-deposited-in-the-coolant table's abscissa-coordinate variable for the SEPD or TEE main tube.
ipwsv2	Signal-variable or control-block ID number defining the power-deposited-in-the-coolant table's abscissa-coordinate variable for the SEPD or TEE side tube.
ipwtr1	Trip ID number that controls the evaluation of the power-deposited-in-the-coolant table for the SEPD or TEE main tube.
ipwtr2	Trip ID number that controls the evaluation of the power-deposited-in-the-coolant table for the SEPD or TEE side tube.
iqf1	Last interpolated interval number of the rate-factor table for the QPPP-factor table of the SEPD or TEE main tube.
iqf2	Last interpolated interval number of the rate-factor table for the QPPP-factor of the SEPD or TEE side tube.
iqp1	Last interpolated interval number of the QPPP-factor table for the SEPD or TEE main tube.
iqp2	Last interpolated interval number of the QPPP-factor table for the SEPD or TEE side tube.
iqpsv1	Signal-variable or control-block ID number defining the QPPP-factor table's abscissa-coordinate variable for the SEPD or TEE main tube.

iqpsv2	Signal-variable or control-block ID number defining the QPPP-factor table's abscissa-coordinate variable for the SEPD or TEE side tube.
iqptr1	Trip ID number that controls the evaluation of the QPPP-factor table for the SEPD or TEE main tube.
iqptr2	Trip ID number that controls the evaluation of the QPPP-factor table for the SEPD or TEE side tube.
isol1	Indicator for velocity update at junction `jun1`.
isol2	Indicator for velocity update at junction `jun2`.
isol3	Indicator for velocity update at junction `jun3`.
jcell	Cell number of the internal-junction cell within the SEPD or TEE main tube.
js1	Junction sequence number at cell 1 of the SEPD or TEE main tube.
js2	Junction sequence number at cell `ncell1` of the SEPD or TEE main tube.
js3	Junction sequence number at cell `ncell2` of the SEPD or TEE side tube.
jun1	Junction number at cell 1 of the SEPD or TEE main tube.
jun2	Junction number at cell `ncell1` of the SEPD or TEE main tube.
jun3	Junction number at cell `ncell2` of the SEPD or TEE side tube.
ncell1	Number of fluid cells in the SEPD or TEE main tube.
ncell2	Number of fluid cells in the SEPD or TEE side tube.
ncells	Total number of main- and side-tube cells plus the internal pseudo cell of the SEPD or TEE (`ncells` = `ncell1` + `ncell2` + 1).
nonof1	Number of timesteps that the QPPP-factor table controlling trip for the SEPD or TEE main tube has been on.
nonof2	Number of timesteps that the QPPP-factor table controlling trip for the SEPD or TEE side tube has been on.
npwrf1	Number of data pairs in the rate-factor table for the power-deposited-in-the-coolant table of the SEPD or TEE main tube.
npwrf2	Number of data pairs in the rate-factor table for the power-deposited-in-the-coolant table of the SEPD or TEE side tube.
npwsv1	Signal-variable or control-block ID number defining the rate-factor table's abscissa-coordinate variable for the power-deposited-in-the-coolant table of the SEPD or TEE main tube.
npwsv2	Signal-variable or control-block ID number defining the rate-factor table's abscissa-coordinate variable for the power-deposited-in-the-coolant table of the SEPD or TEE side tube.

npwtb1	Number of data pairs in the power-deposited-in-the-coolant table for the SEPD or TEE main tube.
npwtb2	Number of data pairs in the power-deposited-in-the-coolant table for the SEPD or TEE side tube.
nqprf1	Number of data pairs in the rate-factor table for the QPPP-factor table of the SEPD or TEE main tube.
nqprf2	Number of data pairs in the rate-factor table for the QPPP-factor table of the SEPD or TEE side tube.
nqpsv1	Signal-variable or control-block ID number defining the rate-factor table's abscissa-coordinate variable for the QPPP-factor table of the SEPD or TEE main tube.
nqpsv2	Signal-variable or control-block ID number defining the rate-factor table's abscissa-coordinate variable for the QPPP-factor table of the SEPD or TEE side tube.
nqptb1	Number of data pairs in the QPPP-factor table for the SEPD or TEE main tube.
nqptb2	Number of data pairs in the QPPP-factor table for the SEPD or TEE side tube.
js1get	Index for getting information from the bd array for left junction of primary tube.
js1put	Index for putting information into the bd array for left junction of primary tube.
js2get	Index for getting information from the bd array for right junction of primary tube.
js2put	Index for putting information into the bd array for right junction of primary tube.
js3get	Index for getting information from the bd array for right junction of secondary tube.
js3put	Index for putting information into the bd array for right junction of secondary tube.

!

teeDumpSize Number of variables in the derived-type teeTabT.

C.56. Module Temp

```
INTEGER(sik) scrSize
PARAMETER (scrSize=25000)
REAL(sdk) scratch(scrSize)
REAL(sdk) scratch1(scrSize)
INTEGER(sik) iscratch(scrSize)
```

REAL VARIABLES:

scratch	real scratch array.
scratch1	real scratch array.

INTEGER VARIABLES:

scrSize	parameter used to dimension `scratch` array.
iscratch	integer `scratch` array.

C.57. Subroutine `tf3ds`

```
REAL(sdk)  aflux(nk),arlck(nk),arvck(nk),s2a(nk),s2b(nk)  &
&,s2c(nk),s2d(nk),s3a(nk),s3b(nk),s3c(nk),s3d(nk),s5a(nk),stder(nk) &
&,stprs(nk),xvoll(nk),xvolv(nk)
LOGICAL lifeq(nk)
```

REAL VARIABLES:

aflux	Net noncondensable-gas mass flow into the nk–nzbcm-level mesh cell.
arlck	Net liquid mass flow into the nk–nzbcm-level mesh cell.
arvck	Net gas mass flow into the nk–nzbcm-level mesh cell.
s2a	Vectorization mask factor for defining the gas mass equation.
s2b	Vectorization mask factor for defining the gas-volume fraction equal to 1.0.
s2c	Vectorization mask factor for defining the gas-volume fraction equal to 0.0.
s2d	Vectorization mask factor for defining the vapor pressure equal to the saturation pressure based on the gas temperature.
s3a	Vectorization mask factor for defining the gas energy equation.
s3b	Vectorization mask factor for defining the liquid temperature equal to the gas temperature.
s3c	Vectorization mask factor for defining the liquid temperature equal to the saturation temperature based on the vapor pressure.
s3d	Vectorization mask factor for defining the gas temperature equal to the saturation temperature based on the vapor pressure.
s5a	Vectorization mask factor for defining the noncondensable-gas mass equation.
stder	Derivative of the saturation temperature with respect to the total pressure based on the saturation temperature and saturation pressure.
stprs	Saturation pressure based on the liquid temperature.

xvoll	Fluid volume (nstab=0) or fluid volume minus liquid volume outflow during the timestep (nstab=1) in the nk–nzbcm-level mesh cell.
xvolv	Fluid volume (nstab=0) or fluid volume minus gas volume outflow during the timestep (nstab=1) in the nk–nzbcm-level mesh cell.

LOGICAL VARIABLES:

lifeq	Fluid-phase flag that is false when two-phase fluid may become single phase. If this flag is false on the second pass through the linearization, the cell will be relinearized.

C.58. Module Thermocple

INTEGER(sik) ittc

REAL(sdk) ntc,diatc,atc,vtc,aw,atw,ckw

REAL VARIABLES:

ntc	Number of thermocouples per ROD or SLAB element.
diatc	Diameter of thermocouple.
atc	Area per unit length of thermocouple.
vtc	Volume per unit length of thermocouple.
aw	Area of ROD or SLAB element to thermocouple weld.
atw	Thickness of ROD or SLAB element to thermocouple weld.
ckw	Thermal conductivity of the ROD or SLAB element-to-thermocouple weld.

INTEGER VARIABLES:

ittc: Thermocouple flag.

0 = no thermocouple on ROD or SLAB element;

1 = thermocouple present on ROD or SLAB element.

C.59. Module TimeStepDat

INTEGER(sik) ntltst,jtltst,ltltst,ntvtst,jtvtst,ltvtst,ntmtst, &

& jtmtst,ltmtst,nprtst,jprtst,lprtst,ndamx,kdamx,ldamx,idiag2

INTEGER VARIABLES:

ntltst	Component number that controls the timestep size, which is due to the liquid-temperature change limit.
jtltst	Horizonal-plane cell number of component ntltst that controls the timestep size, which is due to the liquid-temperature change limit.

ltltst	Axial-z cell number of component `ntltst` that controls the timestep size, which is due to the liquid-temperature change limit.
ntvtst	Component number that controls the timestep size, which is due to the gas-temperature change limit.
jtvtst	Horizonal-plane cell number of component `ntvtst` that controls the timestep size, which is due to the gas-temperature change limit.
ltvtst	Axial-z cell number of component `ntvtst` that controls the timestep size, which is due to the gas-temperature change limit.
ntmtst	Component number that controls the timestep size, which is due to the metal-temperature change limit.
jtmtst	Structure node number of component ntmtst that controls the timestep size, which is due to the "metal"-temperature change limit.
ltmtst	Axial-z cell number of component `ntmtst` that controls the timestep size, which is due to the "metal"-temperature change limit.
nprtst	Component number that controls the timestep size, which is due to the pressure change limit.
jprtst	Horizonal-plane cell number of component `nprtst` that controls the timestep size, which is due to the pressure change limit.
lprtst	Axial-z cell number of component `nprtst` that controls the timestep size, which is due to the pressure change limit.
ndamx	Variable not used.
kdamx	Variable not used.
ldamx	Variable not used.
idiag2	Flag that allows skipping of certain diagnostics generated in subroutine `newdlt` by the `idiag` option (default is on).

C.60. Module `Util`

INTEGER(sik),PARAMETER :: sikx=sik

INTEGER(sik),PARAMETER :: sdkx=sdk

INTEGER VARIABLES:

sikx	Required to declare integer KIND functions with the Nagware F90 compiler.
sdkx	Required to declare real KIND functions with the Nagware F90 compiler.

C.61. Module `ValveArray`

```
TYPE valveArrayT
    REAL(sdk),  POINTER, DIMENSION(:)  :: qp3rf
    REAL(sdk),  POINTER, DIMENSION(:)  :: qp3tb
    REAL(sdk),  POINTER, DIMENSION(:)  :: vrf
    REAL(sdk),  POINTER, DIMENSION(:)  :: vtb1
    REAL(sdk),  POINTER, DIMENSION(:)  :: vtb2
END TYPE valveArrayT
!
TYPE (valveArrayT),DIMENSION(maxComps) :: valveAr
```

DERIVED TYPE VARIABLES:

valveAr	Variables specific to the VALVE component.

REAL VARIABLES:

!Elements of DERIVED TYPE valveArrayT

qp3rf	Rate-factor table for the QPPP-factor table.
qp3tb	QPPP-factor table.
vrf	Rate-factor table for the VALVE table(s).
vtb1	First VALVE table.
vtb2	Second VALVE table.

C.62. Module `ValveVlt`

```
TYPE valveTabT
    REAL(sdk) avlve
    REAL(sdk) bsmass
    REAL(sdk) epsw
    REAL(sdk) favlve
    REAL(sdk) fl(2)
    REAL(sdk) fmaxov
    REAL(sdk) fminov
    REAL(sdk) fric0
    REAL(sdk) fric0r
    REAL(sdk) fv(2)
    REAL(sdk) hdrdx
    REAL(sdk) houtl
    REAL(sdk) houtv
    REAL(sdk) hvlve
```

REAL(sdk) qp3in
REAL(sdk) qp3off
REAL(sdk) radin
REAL(sdk) rqp3mx
REAL(sdk) rvmx
REAL(sdk) rvov
REAL(sdk) th
REAL(sdk) toutl
REAL(sdk) toutv
REAL(sdk) xpos
REAL(sdk) type1
REAL(sdk) type2
INTEGER(sik) ichf
INTEGER(sik) icj1
INTEGER(sik) icj2
INTEGER(sik) iconc
INTEGER(sik) ionoff
INTEGER(sik) iqf
INTEGER(sik) iqp
INTEGER(sik) iqp3sv
INTEGER(sik) iqp3tr
INTEGER(sik) isollb
INTEGER(sik) isolrb
INTEGER(sik) ivf
INTEGER(sik) ivp1
INTEGER(sik) ivp2
INTEGER(sik) ivps
INTEGER(sik) ivsv
INTEGER(sik) ivtr
INTEGER(sik) ivtrov
INTEGER(sik) ivty
INTEGER(sik) ivtyov
INTEGER(sik) js1
INTEGER(sik) js2
INTEGER(sik) jun1
INTEGER(sik) jun2
INTEGER(sik) mode

```
    INTEGER(sik) ncells
    INTEGER(sik) nonoff
    INTEGER(sik) nqp3rf
    INTEGER(sik) nqp3sv
    INTEGER(sik) nqp3tb
    INTEGER(sik) nvrf
    INTEGER(sik) nvsv
    INTEGER(sik) nvtb1
    INTEGER(sik) nvtb2
    INTEGER(sik) js1get
    INTEGER(sik) js1put
    INTEGER(sik) js2get
    INTEGER(sik) js2put
END TYPE valveTabT
!
TYPE(valveTabT),DIMENSION(maxComps) :: valveTab
INTEGER(sik) valveDumpSize
PARAMETER (valveDumpSize=66)
```

DERIVED TYPE VARIABLES:

valveTAB The VLT for the VALVE component.

REAL VARIABLES:

!Elements of DERIVED TYPE valveTabT

avlve	VALVE-interface, open-flow area.
bsmass	Time-integrated fluid mass flow from the VALVE.
epsw	Wall surface roughness.
favlve	Fraction of the fully open-flow area `avlve` to which the adjustable-valve interface is set.
fl(2)	Liquid mass-flow corrections for mass-conservation checks.
fmaxov	Maximum flow area fraction or relative valve-stem position during VALVE-interface adjustment by the overriding trip.
fminov	Minimum flow area fraction or relative valve-stem position during VALVE-interface adjustment by the overriding trip.
fric0	Fully open VALVE-interface, form-loss FRIC for forward flow.
fric0r	Fully open VALVE-interface, form-loss FRIC for reverse flow.
fv(2)	Gas mass-flow corrections for mass-conservation checks.
hdrdx	Fully open VALVE-interface hydraulic diameter over `dx`.
houtl	HTC between outer boundary of the VALVE wall and liquid.

houtv — HTC between outer boundary of the VALVE wall and gas.

hvlve — VALVE-interface open hydraulic diameter.

qp3in — Initial QPPP factor.

qp3off — QPPP factor when the controlling trip is off after it was on.

radin — Inner radius of the VALVE wall.

rqp3mx — Maximum allowed rate of change of the QPPP factor.

rvmx — Maximum rate of change of the VALVE-interface flow area fraction or relative valve-stem position.

rvov — Rate of change of the VALVE-interface flow area fraction or relative valve-stem position when controlled by the overriding trip that is on.

th — Thickness of the VALVE wall.

toutl — Liquid temperature outside the VALVE wall.

toutv — Gas temperature outside the VALVE wall.

xpos — Adjustable VALVE-interface relative valve-stem position.

type1 — Type of adjacent component at `jun1`.

type2 — Type of adjacent component at `jun2`.

INTEGER VARIABLES:

!Elements of DERIVED TYPE valveTabT

ichf — CHF calculation option.

icj1 — Iteration index of adjacent component at junction `jun1`.

icj2 — Iteration index of adjacent component at junction `jun2`.

iconc — Presence of solute in the liquid option.

ionoff — Number of timesteps that the VALVE table's controlling trip has been on.

iqf — Last interpolated interval number in the rate-factor table for the QPPP-factor table.

iqp — Last interpolated interval number in the QPPP-factor table.

iqp3sv — Signal-variable or control-block ID number defining the QPPP-factor table's abscissa-coordinate variable.

iqp3tr — Trip ID number that controls evaluation of the QPPP-factor table.

isollb — Indicator for velocity update at junction `jun1`.

isolrb — Indicator for velocity update at junction `jun2`.

ivf — Last interpolated interval number in the rate-factor table for the VALVE table.

ivp1 — Last interpolated interval number in the first VALVE table.

ivp2 — Last interpolated interval number in the second VALVE table.

ivps — Adjustable VALVE-interface number.

ivsv	Signal-variable or control-block ID number that defines the VALVE table's independent variable.
ivtr	Trip ID number that controls evaluation of the VALVE table(s).
ivtrov	Trip ID number that overrides trip `ivtr` control of VALVE-interface adjustment.
ivty	Valve-type option.
ivtyov	Type of VALVE-interface adjustment by the overriding trip `ivtrov`.
js1	Junction sequence number at cell 1 of the VALVE.
js2	Junction sequence number at cell `ncells` of the VALVE.
jun1	Junction number at cell 1 of the VALVE.
jun2	Junction number at cell `ncells` of the VALVE.
mode	Indicator for valve movement over the previous timestep. –1 = closing; 0 = no movement; 1 = opening.
ncells	Total number of fluid cells in the VALVE.
nonoff	Number of timesteps that the QPPP-factor table's controlling trip has been on.
nqp3rf	Number of data pairs in the rate-factor table for the QPPP-factor table.
nqp3sv	Signal-variable or control-block ID number defining the rate-factor table's abscissa-coordinate variable for the QPPP-factor table.
nqp3tb	Number of data pairs in the QPPP-factor table.
nvrf	Number of rate-factor table data pairs where the rate factor is applied to the VALVE table's independent variable.
nvsv	Signal-variable or control-block ID number defining the rate-factor table's abscissa-coordinate variable for the VALVE table(s).
nvtb1	Number of data pairs in the first VALVE table.
nvtb2	Number of data pairs in the second VALVE table.
js1get	Index for getting information from the `bd` array for left junction.
js1put	Index for putting information into the `bd` array for left junction.
js2get	Index for getting information from the `bd` array for right junction.
js2put	Index for putting information into the `bd` array for right junction.

!

valveDumpSize Number of variables in the derived-type `valveTabT`.

C.63. Module `VectDrag`

```
INTEGER(sik) tmpvl
INTEGER(sik)      iwrtpt,ntmpv,lvt1,lvt2,lvt3,lvt4,                &
& lvt5,lvt6,lvt7,lvt8,lvt9,lvt10,lvt11,lvt12,lvt13,lvt14,lvt15,    &
& lvt16,lvt17,lvt18,lvt19,lvt20,lvt21,lvt22,lvt23,lvt24,lvt25,     &
& lvt26,lvt27,lvt28,lvt29,lvt30,lvt31,lvt32,lvt33,lvt34,lvt35,     &
& lvt36,lvt37,lvt38,lvt39,lvt40,lvt41,lvt42,lvt43,lvt44,lvt45,     &
& lvt46,lvt47,lvt48,lvt49,lvt50,lvt51,lvt52,lvt53,lvt54,lvt55,     &
& lvt56,lvt57,lvt58,lvt59,lvt60,lvt61,lvt62,lvt63,lvt64,lvt65,     &
& lvt66,lvt67,lvt68,lvt69,lvt70,lvt71,lvt72,lvt73,lvt74,lvt75,     &
& lvt76,lvt77,lvt78,lvt79,lvt80,lvt81,lvt82,lvt83,lvt84,lvt85,     &
& lvt86,lvt87,lvt88,lvt89,lvt90,lvt91,lvt92,lvt93,lvt94,lvt95,     &
& lvt96,lvt97,lvt98,lvt99,lvt100,lvt101,lvt102,lvt103,lvt104,      &
& lvt105,lvt106,lvt107,lvt108,lvt109,lvt110,lvt111,lvt112,lvt113,  &
& lvt114,lvt115,lvt116,lvt117,lvt118,lvt119,lvt120,lvt121,lvt122,  &
& lvt123,lvt124
INTEGER(sik) lvt125,lvt126,lvt127,lvt128,lvt129,lvt130
!
DATA iwrtpt/0/
PARAMETER (ntmpv=130)
```

INTEGER VARIABLES:

iwrtpt	Flag to set up temporary pointers for subroutines `prefwd` and `preifd`.
lastp1	LAST+1 pointer for the first free location in memory for the temporary storage arrays set up to vectorize the 3D wall-shear and interfacial-drag coefficient evaluations.
ntmpv	Number of temporary storage arrays (130) in subroutines `prefwd` and `preifd` set up to vectorize the 3D wall-shear and interfacial-drag coefficient evaluations.
tmpvl	Number of calculative mesh cells in the 3D VESSEL component.
lvt#	Pointer variable for the # th (# = 1 to 130) temporary storage array set up to vectorize the 3D wall-shear and interfacial-drag-coefficient evaluations.

C.64. Module `VessArray`

```
TYPE vessArrayT
    REAL(sdk), POINTER, DIMENSION(:) :: z
    REAL(sdk), POINTER, DIMENSION(:) :: dz
    REAL(sdk), POINTER, DIMENSION(:) :: rad
    REAL(sdk), POINTER, DIMENSION(:) :: dr
    REAL(sdk), POINTER, DIMENSION(:) :: th
    REAL(sdk), POINTER, DIMENSION(:) :: dth
    REAL(sdk), POINTER, DIMENSION(:) :: avent
    REAL(sdk), POINTER, DIMENSION(:) :: dpcvn
    REAL(sdk), POINTER, DIMENSION(:) :: dpovn
    REAL(sdk), POINTER, DIMENSION(:) :: frcvn
    REAL(sdk), POINTER, DIMENSION(:) :: frovn
    REAL(sdk), POINTER, DIMENSION(:) :: locvn
    REAL(sdk), POINTER, DIMENSION(:) :: vvtb
    REAL(sdk), POINTER, DIMENSION(:) :: zsgrd
    INTEGER(sik), POINTER, DIMENSION(:) :: isrl
    INTEGER(sik), POINTER, DIMENSION(:) :: isrc
    INTEGER(sik), POINTER, DIMENSION(:) :: isrf
    INTEGER(sik), POINTER, DIMENSION(:) :: juns
    INTEGER(sik), POINTER, DIMENSION(:) :: jsn
    INTEGER(sik), POINTER, DIMENSION(:) :: jsnget
    INTEGER(sik), POINTER, DIMENSION(:) :: jsnput
    INTEGER(sik), POINTER, DIMENSION(:) :: msc
    INTEGER(sik), POINTER, DIMENSION(:) :: nsrl
    REAL(sdk), POINTER, DIMENSION(:) :: svc
    REAL(sdk), POINTER, DIMENSION(:) :: sac
    REAL(sdk), POINTER, DIMENSION(:) :: scc
    REAL(sdk), POINTER, DIMENSION(:) :: slc
    REAL(sdk), POINTER, DIMENSION(:) :: sve
    REAL(sdk), POINTER, DIMENSION(:) :: sle
    REAL(sdk), POINTER, DIMENSION(:) :: dvvdp
    REAL(sdk), POINTER, DIMENSION(:) :: dvldp
!
!   Constant sources from 1-D to 3-D velocity equations
!
```

```
      REAL(sdk), POINTER, DIMENSION(:) :: svvx
      REAL(sdk), POINTER, DIMENSION(:) :: svlx
      REAL(sdk), POINTER, DIMENSION(:) :: svvy
      REAL(sdk), POINTER, DIMENSION(:) :: svly
      REAL(sdk), POINTER, DIMENSION(:) :: svvz
      REAL(sdk), POINTER, DIMENSION(:) :: svlz
!
!     Coefficients coupling 1-D to 3-D velocity equations
!
      REAL(sdk), POINTER, DIMENSION(:) :: scvvx
      REAL(sdk), POINTER, DIMENSION(:) :: scvlx
      REAL(sdk), POINTER, DIMENSION(:) :: scvvy
      REAL(sdk), POINTER, DIMENSION(:) :: scvly
      REAL(sdk), POINTER, DIMENSION(:) :: scvvz
      REAL(sdk), POINTER, DIMENSION(:) :: scvlz
      REAL(sdk), POINTER, DIMENSION(:) :: scvv
      REAL(sdk), POINTER, DIMENSION(:) :: scvl
!
!     Original velocity equation 1-D to 3-D source terms
!     need to be cleaned from code (still used for tf3ds1)
!
      REAL(sdk), POINTER, DIMENSION(:) :: smoml
      REAL(sdk), POINTER, DIMENSION(:) :: smomv

      REAL(sdk), POINTER, DIMENSION(:) :: velsv
      REAL(sdk), POINTER, DIMENSION(:) :: velsl
      REAL(sdk), POINTER, DIMENSION(:) :: psold
      REAL(sdk), POINTER, DIMENSION(:) :: psnew
      REAL(sdk), POINTER, DIMENSION(:) :: gravt
      REAL(sdk), POINTER, DIMENSION(:) :: gravr
      REAL(sdk), POINTER, DIMENSION(:) :: refld
      REAL(sdk), POINTER, DIMENSION(:) :: funh
      REAL(sdk), POINTER, DIMENSION(:) :: nhsca
      REAL(sdk), POINTER, DIMENSION(:) :: alpag
      REAL(sdk), POINTER, DIMENSION(:) :: alprw
      REAL(sdk), POINTER, DIMENSION(:) :: alpsm
      REAL(sdk), POINTER, DIMENSION(:) :: zags
```

```
      REAL(sdk), POINTER, DIMENSION(:) :: zdfs
      REAL(sdk), POINTER, DIMENSION(:) :: zrws
      REAL(sdk), POINTER, DIMENSION(:) :: zsms
      REAL(sdk), POINTER, DIMENSION(:) :: alpan
      REAL(sdk), POINTER, DIMENSION(:) :: alprn
      REAL(sdk), POINTER, DIMENSION(:) :: alpsn
      REAL(sdk), POINTER, DIMENSION(:) :: zagsn
      REAL(sdk), POINTER, DIMENSION(:) :: zdfsn
!     The following variables ending in 'sm' are used for
!     DOE proprietary updates.  Please don't remove.
      REAL(sdk), POINTER, DIMENSION(:) :: rsm
      REAL(sdk), POINTER, DIMENSION(:) :: rvsm
      INTEGER(sik), POINTER, DIMENSION(:) :: nfvsm
      INTEGER(sik), POINTER, DIMENSION(:) :: nf1sm
      INTEGER(sik), POINTER, DIMENSION(:) :: nf2sm
      INTEGER(sik), POINTER, DIMENSION(:) :: nf3sm
      INTEGER(sik), POINTER, DIMENSION(:) :: nfv4sm
      INTEGER(sik), POINTER, DIMENSION(:) :: nflsm
      INTEGER(sik), POINTER, DIMENSION(:) :: nfl4sm
      REAL(sdk), POINTER, DIMENSION(:) :: esm
      REAL(sdk), POINTER, DIMENSION(:) :: evsm
      INTEGER(sik), POINTER, DIMENSION(:) :: nfcvsm
      INTEGER(sik), POINTER, DIMENSION(:) :: nfclsm
      REAL(sdk), POINTER, DIMENSION(:) :: tpsm
      REAL(sdk), POINTER, DIMENSION(:) :: tmsm
      REAL(sdk), POINTER, DIMENSION(:) :: radsm
      REAL(sdk), POINTER, DIMENSION(:) :: vvsm
      REAL(sdk), POINTER, DIMENSION(:) :: vlsm
!
      REAL(sdk), POINTER, DIMENSION(:) :: zrwsn
      REAL(sdk), POINTER, DIMENSION(:) :: zsmsn
      REAL(sdk), POINTER, DIMENSION(:) :: alpcn
      REAL(sdk), POINTER, DIMENSION(:) :: alptn
      REAL(sdk), POINTER, DIMENSION(:) :: zchfn
      REAL(sdk), POINTER, DIMENSION(:) :: ztbn
!
    END TYPE vessArrayT
```

```
!
   TYPE (vessArrayT),DIMENSION(maxComps) :: vsAr
!
   TYPE vsSrcArT
    REAL(sdk), POINTER, DIMENSION(:) :: dx
    REAL(sdk), POINTER, DIMENSION(:) :: vln2
    REAL(sdk), POINTER, DIMENSION(:) :: vvn2
    REAL(sdk), POINTER, DIMENSION(:) :: faFrac
    REAL(sdk), POINTER, DIMENSION(:) :: vlt2
    REAL(sdk), POINTER, DIMENSION(:) :: vvt2
    REAL(sdk), POINTER, DIMENSION(:) :: vln1
    REAL(sdk), POINTER, DIMENSION(:) :: vvn1
    REAL(sdk), POINTER, DIMENSION(:) :: dfldp
    REAL(sdk), POINTER, DIMENSION(:) :: dfvdp
    REAL(sdk), POINTER, DIMENSION(:) :: fa1
    REAL(sdk), POINTER, DIMENSION(:) :: hd
    REAL(sdk), POINTER, DIMENSION(:) :: vlto2
    REAL(sdk), POINTER, DIMENSION(:) :: vvto2
    REAL(sdk), POINTER, DIMENSION(:) :: vvvol
    REAL(sdk), POINTER, DIMENSION(:) :: vlvol
    REAL(sdk), POINTER, DIMENSION(:) :: cifn
    REAL(sdk), POINTER, DIMENSION(:) :: bitn
    REAL(sdk), POINTER, DIMENSION(:) :: grav
    REAL(sdk), POINTER, DIMENSION(:) :: vl
    REAL(sdk), POINTER, DIMENSION(:) :: vv
    REAL(sdk), POINTER, DIMENSION(:) :: fa2
    REAL(sdk), POINTER, DIMENSION(:) :: vlt1
    REAL(sdk), POINTER, DIMENSION(:) :: vvt1
    REAL(sdk), POINTER, DIMENSION(:) :: vlto1
    REAL(sdk), POINTER, DIMENSION(:) :: vvto1
    REAL(sdk), POINTER, DIMENSION(:) :: wfmfl
    REAL(sdk), POINTER, DIMENSION(:) :: wfmfv
!   The following variables ending in 'sm' are used for
!   DOE proprietary updates.  Please don't remove.
    REAL(sdk), POINTER, DIMENSION(:) :: xsm
    REAL(sdk), POINTER, DIMENSION(:) :: ysm
    REAL(sdk), POINTER, DIMENSION(:) :: zsm
```

!

END TYPE vsSrcArT

TYPE (vsSrcArT),DIMENSION(maxComps) :: vsSrcAr

DERIVED TYPE VARIABLES:

vsAr	Arrays specific to the VESSEL component (VESSEL mesh arrays are in derived-type array variable `VsAr3`).
vsSrcAr	Target location for `bd` array pointers that point to the VESSEL data structure.

REAL VARIABLES:

!Elements of DERIVED TYPE vessArrayT

z	Axial-direction cell's upper elevation.
dz	Axial-direction cell's length (Δz).
rad	Radial cell's outer radius.
dr	Radial- or x-direction cell's length (Δr or Δx).
th	Theta-direction cell's angle.
dth	Theta- or y-direction cell's length ($\Delta\theta$ or Δy).
avent	Pointer for vent-valve area.
dpcvn	Pointer for vent-valve maximum ΔP to be closed.
dpovn	Pointer for vent-valve minimum ΔP to be open.
frcvn	FRIC value when the vent valve is closed.
frovn	FRIC value when the vent valve is open.
locvn	Vent-valve location.
vvtb	Multiple-point, vent-valve table.
zsgrd	New location of grid spacer.
svc	Gas mass source.
sac	Noncondensable-gas mass source.
scc	Solute-concentration mass source. `isolut` = 0 or 1.
slc	Liquid mass source.
sve	Vapor energy source.
sle	Liquid energy source.
dvvdp	Derivative of gas velocity with respect to pressure.
dvldp	Derivative of the liquid velocity with respect to pressure.
smomv	Gas momentum source.
smoml	Liquid momentum source.
velsv	Gas source velocity.
velsl	Liquid source velocity.

psold	Old source pressure.
psnew	New source pressure.
gravt	Theta- or y-direction component of the gravity unit vector on each θ- or y-direction interface of a VESSEL cell.
gravr	Radial- or x-direction component of the gravity unit vector on each r- or x-direction interface of a VESSEL cell.
refld	Reflood flag.
funh	Fraction of the heat-structure surface in each horizontal-plane cell that is unheated.
nhsca	HTSTR component number for average rod.
alpag	Old gas-volume fraction at the agitated inverted annular flow (IAF).
alprw	Old gas-volume fraction at the rough-wavy IAF.
alpsm	Old gas-volume fraction at the smooth IAF.
zags	Old location of agitated IAF.
zdfs	Old location of dispersed IAF.
zrws	Old location of rough-wavy IAF.
zsms	Old location of smooth IAF.
alpan	New gas-volume fraction at the agitated IAF.
alprn	New gas-volume fraction at the rough-wavy IAF.
alpsn	New gas-volume fraction at the smooth IAF.
zagsn	New location of agitated IAF.
zdfsn	New location of disposed IAF.
rsm	Special-purpose, DOE-model parameter.
rvsm	Special-purpose, DOE-model parameter.
esm	Special-purpose, DOE-model parameter (`nsizesm = nxrv*ntsx *nasx+nrsx*nytv*nasx+ nrsx*ntsx*nzzv`).
evsm	Special-purpose, DOE-model parameter.
tpsm	Special-purpose, DOE-model parameter.
tmsm	Special-purpose, DOE-model parameter.
radsm	Special-purpose, DOE-model parameter.
vvsm	Special-purpose, DOE-model parameter.
vlsm	Special-purpose, DOE-model parameter.
zrwsn	New location of rough-wavy IAF.
zsmsn	New location of smooth IAF.
alpcn	New gas-volume fraction at the CHF point.
alptn	New gas-volume fraction at the transition boiling.
zchfn	New location of CHF point.

ztbn New location of transition boiling.

!Elements of DERIVED TYPE vsSrcArT

dx

vln2

vvn2

faFrac

vlt2

vvt2

vln1

vvn1

dfldp

dfvdp

fa1

hd

vlto2

vvto2

vvvol

vlvol

cifn

bitn

grav

vl

vv

fa2

vlt1

vvt1

vlto1

vvto1

wfmfl

wfmfv

xsm Special-purpose, DOE-model parameter.

ysm Special-purpose, DOE-model parameter.

zsm Special-purpose, DOE-model parameter.

Integer VARIABLES:

!Elements of DERIVED TYPE vessArrayT

isrl Level number associated with the source connection.

isrc Relative cell number associated with the source connection.

isrf	Cell face number associated with the source connection.
juns	Junction number associated with the source connection.
jsn	Junction sequence number associated with the source connection.
jsnget	Index for getting boundary data at source connection.
jsnput	Index for putting boundary data at source connection.
msc	Absolute cell number of source connection.
nsrl	Number of source connections on an axial level.
nfvsm	Special-purpose, DOE-model parameter.
nf1sm	Special-purpose, DOE-model parameter.
nf2sm	Special-purpose, DOE-model parameter.
nf3sm	Special-purpose, DOE-model parameter.
nfv4sm	Special-purpose, DOE-model parameter.
nflsm	Special-purpose, DOE-model parameter.
nfl4sm	Special-purpose, DOE-model parameter.
nfcvsm	Special-purpose, DOE-model parameter.
nfclsm	Special-purpose, DOE-model parameter.

C.65. Module `VessArray3`

```
TYPE vessArray3T
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hla
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hva
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: q3drl
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: q3drv
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: wat
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hlatw
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: hvatw
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: finan
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: rmem
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: rom
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: qrd
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: sig
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: am
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: qsl
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: arc
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: vol
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: volg
```

```
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vmfrl
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vmfrv
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cpl
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cpv
REAL(sdk), POINTER, DIMENSION(:,:,:) :: tsn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: tssn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cl
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cv
REAL(sdk), POINTER, DIMENSION(:,:,:) :: visl
REAL(sdk), POINTER, DIMENSION(:,:,:) :: visv
REAL(sdk), POINTER, DIMENSION(:,:,:) :: hfg
REAL(sdk), POINTER, DIMENSION(:,:,:) :: hgam
REAL(sdk), POINTER, DIMENSION(:,:,:) :: lccfl
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fayt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: faz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: faxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fagyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fagz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fagxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vmyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vmz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vmxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: hdyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: hdz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: hdxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wflyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wflz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wflxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wfvyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wfvz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wfvxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vwfmly
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vwfmlz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vwfmlx
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vwfmvy
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vwfmvz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vwfmvx
```

```
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvlyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvlz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvlxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfzlyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfzlz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfzlxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfrlyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfrlz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfrlxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfzvyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfzvz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfzvxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfrvyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfrvz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cfrvxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dtsdp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: deldp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: degdp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: deldt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: degdt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: drldp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: drgdp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: drldt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: drgdt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: hvs
REAL(sdk), POINTER, DIMENSION(:,:,:) :: hls
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dhvs
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dhls
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dtssdp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: deadt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: deadp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dradp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dradt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: drlast
```

```
REAL(sdk), POINTER, DIMENSION(:,:,:) :: oryt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: orz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: orxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wmyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wmz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: wmxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dzz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rmean
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdyta
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdza
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rdxra
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rddyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rddz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rddxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: alpo
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dalva
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dalp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: favyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: favz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: favxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: falyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: falz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: falxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frvyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frvz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frvxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fevyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fevz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fevxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frayt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fraz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fraxr
```

```
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frlyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frlz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frlxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: felyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: felz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: felxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c1p1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c1p2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c1p3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c1p4
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c1p5
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c1p6
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c2p1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c2p2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c2p3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c2p4
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c2p5
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c2p6
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c3p1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c3p2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c3p3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c3p4
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c3p5
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c3p6
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c4p1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c4p2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c4p3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c4p4
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c4p5
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c4p6
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c5p1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c5p2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c5p3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c5p4
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c5p5
REAL(sdk), POINTER, DIMENSION(:,:,:) :: c5p6
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dprhs
```

```
REAL(sdk), POINTER, DIMENSION(:,:,:) :: darhs
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dtvrhs
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dtlrhs
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dparhs
REAL(sdk), POINTER, DIMENSION(:,:,:) :: fbit
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvs1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvs1m
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvls1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvls1m
REAL(sdk), POINTER, DIMENSION(:,:,:) :: sc1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: sc1m
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvs3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvs3m
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvls3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvls3m
REAL(sdk), POINTER, DIMENSION(:,:,:) :: sc3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: sc3m
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvs2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvls2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: sc2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: scd1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: scd1m
REAL(sdk), POINTER, DIMENSION(:,:,:) :: scd2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: scd3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: scd3m
REAL(sdk), POINTER, DIMENSION(:,:,:) :: bit
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frci1
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frci2
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frci3
REAL(sdk), POINTER, DIMENSION(:,:,:) :: ciyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: ciz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cixr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: chti
REAL(sdk), POINTER, DIMENSION(:,:,:) :: chtia
REAL(sdk), POINTER, DIMENSION(:,:,:) :: alv
REAL(sdk), POINTER, DIMENSION(:,:,:) :: alve
REAL(sdk), POINTER, DIMENSION(:,:,:) :: arv
```

```
REAL(sdk), POINTER, DIMENSION(:,:,:) :: conco
REAL(sdk), POINTER, DIMENSION(:,:,:) :: pa
REAL(sdk), POINTER, DIMENSION(:,:,:) :: roa
REAL(sdk), POINTER, DIMENSION(:,:,:) :: ea
REAL(sdk), POINTER, DIMENSION(:,:,:) :: alp
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rov
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rol
REAL(sdk), POINTER, DIMENSION(:,:,:) :: s
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: ev
REAL(sdk), POINTER, DIMENSION(:,:,:) :: el
REAL(sdk), POINTER, DIMENSION(:,:,:) :: tv
REAL(sdk), POINTER, DIMENSION(:,:,:) :: tl
REAL(sdk), POINTER, DIMENSION(:,:,:) :: gam
REAL(sdk), POINTER, DIMENSION(:,:,:) :: p
REAL(sdk), POINTER, DIMENSION(:,:,:) :: arev
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvtyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvtz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvtxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: arl
REAL(sdk), POINTER, DIMENSION(:,:,:) :: arel
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vltyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vltz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vltxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: ara
REAL(sdk), POINTER, DIMENSION(:,:,:) :: owvyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: owvz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: owvxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: owlyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: owlz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: owlxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: bitn
```

```
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frci1n
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frci2n
REAL(sdk), POINTER, DIMENSION(:,:,:) :: frci3n
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cinyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cinz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: cinxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: chtin
REAL(sdk), POINTER, DIMENSION(:,:,:) :: chtan
REAL(sdk), POINTER, DIMENSION(:,:,:) :: alvn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: alven
REAL(sdk), POINTER, DIMENSION(:,:,:) :: arvn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: conc
REAL(sdk), POINTER, DIMENSION(:,:,:) :: pan
REAL(sdk), POINTER, DIMENSION(:,:,:) :: roan
REAL(sdk), POINTER, DIMENSION(:,:,:) :: ean
REAL(sdk), POINTER, DIMENSION(:,:,:) :: alpn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: rovn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: roln
REAL(sdk), POINTER, DIMENSION(:,:,:) :: sn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvnyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvnz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvnxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlnyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlnz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlnxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: evn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: eln
REAL(sdk), POINTER, DIMENSION(:,:,:) :: tvn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: tln
REAL(sdk), POINTER, DIMENSION(:,:,:) :: gamn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: pn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: arevn
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvntyt
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvntz
REAL(sdk), POINTER, DIMENSION(:,:,:) :: vvntxr
REAL(sdk), POINTER, DIMENSION(:,:,:) :: arln
REAL(sdk), POINTER, DIMENSION(:,:,:) :: areln
```

```
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlntyt
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlntz
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: vlntxr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: aran
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: wvyt
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: wvz
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: wvxr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: wlyt
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: wlz
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: wlxr
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: spifz
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvvs2m
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: dvls2m
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: sc2m
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: scd2m
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xcl
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xl1
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xl2
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xl3
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xl4
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xl5
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xl6
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xls
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xcv
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xv1
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xv2
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xv3
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xv4
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xv5
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xv6
    REAL(sdk), POINTER, DIMENSION(:,:,:) :: xvs
END TYPE vessArray3T
!
TYPE (vessArray3T),DIMENSION(maxComps) :: vsAr3
```

DERIVED TYPE VARIABLES:

vsAr3 Variables specific to the 3D VESSEL component.

REAL VARIABLES:

!Elements of DERIVED TYPE vessArray3T

hla	Sum of the products of the liquid HTC and the heat-transfer area.
hva	Sum of the products of the gas HTC and the heat-transfer area.
q3drl	Radiation power absorbed by the liquid.
q3drv	Radiation power absorbed by the gas.
wat	Total heat-transfer area.
hlatw	Similar to HLA, except that the product includes wall temperature.
hvatw	Similar to HVA, except that the product includes wall temperature.
finan	Inverted-annular regime factor.
rmem	Mixture energy.
rom	Mixture density.
qrd	Variable not used.
sig	Surface tension.
am	Noncondensable-gas mass.
qsl	Wall heat flux.
arc	Density of solute in cell, $c(1-\alpha)\rho_\ell$.
vol	Cell fluid volume.
volg	Cell geometric volume.
vmfrl	Liquid mass flux in the axial direction.
vmfrv	Gas mass flux in the axial direction.
cpl	Liquid specific heat at constant pressure.
cpv	Gas specific heat at constant pressure.
tsn	Saturation temperature at total pressure.
tssn	Saturation temperature at steam pressure.
cl	Liquid thermal conductivity.
cv	Gas thermal conductivity.
visl	Liquid viscosity.
visv	Gas viscosity.
hfg	Latent heat of vaporization.
hgam	Energy contribution to phase change from subcooled boiling.
lccfl	CCFL flag.
fayt	Actual flow area of the azimuthal θ or y face.
faz	Actual flow area of the axial z face.
faxr	Actual flow area of the radial r or x face.

fagyt	Geometric flow area of the azimuthal θ or y face.
fagz	Geometric flow area of the axial z face.
fagxr	Geometric flow area of the radial r or x face.
vmyt	Mixture velocity at the azimuthal θ or y face.
vmz	Mixture velocity at the axial z face.
vmxr	Mixture velocity at the radial r or x face.
hdyt	Hydraulic diameter at the azimuthal θ or y face.
hdz	Hydraulic diameter at the axial z face.
hdxr	Hydraulic diameter at the radial r or x face.
wflyt	Wall-friction factor for liquid at the azimuthal θ or y face.
wflz	Wall-friction factor for liquid at the axial z face.
wflxr	Wall-friction factor for liquid at the radial r or x face.
wfvyt	Wall-friction factor for gas at the azimuthal θ or y face.
wfvz	Wall-friction factor for gas at the axial z face.
wfvxr	Wall-friction factor for gas at the radial r or x face.
vwfmly	Wall-friction multiplier factor for the liquid at the azimuthal θ or y face.
vwfmlz	Wall-friction multiplier factor for the liquid at the axial z face.
vwfmlx	Wall-friction multiplier factor for the liquid at the radial r or x face.
vwfmvy	Wall-friction multiplier factor for the gas at the azimuthal θ or y face.
vwfmvz	Wall-friction multiplier factor for the gas at the axial z face.
vwfmvx	Wall-friction multiplier factor for the gas at the radial r or x face.
dvvyt	Derivative of the gas velocity with respect to pressure at the azimuthal θ or y face.
dvvz	Derivative of the gas velocity with respect to pressure at the axial z face.
dvvxr	Derivative of the gas velocity with respect to pressure at the radial r or x face.
dvlyt	Derivative of the liquid velocity with respect to pressure at the azimuthal θ or y face.
dvlz	Derivative of the liquid velocity with respect to pressure at the axial z face.
dvlxr	Derivative of the liquid velocity with respect to pressure at the radial r or x face.
cfzlyt	Liquid forward-flow-direction additive friction-loss coefficient at the azimuthal θ or y face.

cfzlz	Liquid forward-flow-direction additive friction-loss coefficient at the axial z face.
cfzlxr	Liquid forward-flow-direction additive friction-loss coefficient at the radial r or x face.
cfrlyt	Liquid reverse-flow-direction additive friction-loss coefficient at the azimuthal θ or y face.
!	
cfrlz	Liquid reverse-flow-direction additive friction-loss coefficient at the axial z face.
cfrlxr	Liquid reverse-flow-direction additive friction-loss coefficient at the radial r or x face.
cfzvyt	Gas forward-flow-direction additive friction-loss coefficient at the azimuthal θ or y face.
cfzvz	Gas forward-flow-direction additive friction-loss coefficient at the axial z face.
cfzvxr	Gas forward-flow-direction additive friction-loss coefficient at the radial r or x face.
cfrvyt	Gas reverse-flow-direction additive friction-loss coefficient at the azimuthal θ or y face.
cfrvz	Gas reverse-flow-direction additive friction-loss coefficient at the axial z face.
cfrvxr	Gas reverse-flow-direction additive friction-loss coefficient at the radial r or x face.
dtsdp	Derivative of TSAT with respect to pressure.
deldp	Derivative of the liquid internal energy with respect to pressure at constant temperature.
degdp	Derivative of the steam internal energy with respect to pressure at constant temperature.
deldt	Derivative of the liquid internal energy with respect to temperature at constant pressure.
degdt	Derivative of the steam internal energy with respect to temperature at constant pressure.
drldp	Derivative of the liquid density with respect to pressure at constant temperature.
drgdp	Derivative of the steam density with respect to pressure at constant temperature.
drldt	Derivative of the liquid density with respect to temperature at constant pressure.
drgdt	Derivative of the steam density with respect to temperature at constant pressure.
hvs	Enthalpy of the steam at TSAT.

hls	Enthalpy of the liquid at TSAT.
dhvs	Derivative of the enthalpy of the gas at TSAT with respect to pressure.
dhls	Derivative of the enthalpy of the liquid at TSAT with respect to pressure.
dtssdp	Derivative of the saturation temperature corresponding to the steam pressure with respect to pressure.
deadt	Derivative of the noncondensable-gas internal energy with respect to temperature at constant pressure.
deadp	Derivative of the noncondensable-gas internal energy with respect to pressure at constant temperature.
dradp	Derivative of the noncondensable-gas density with respect to pressure at constant temperature.
dradt	Derivative of the noncondensable-gas density with respect to temperature at constant pressure.
drlast	Variable not used.
oryt	Scale factor used to reduce cross-flow at the azimuthal θ or y face to simulate the presence of an orifice (currently set to 1).
orz	Scale factor used to reduce cross-flow at the axial z face to simulate the presence of an orifice (currently set to 1).
orxr	Scale factor used to reduce cross-flow at the radial r or x face to simulate the presence of an orifice (currently set to 1).
wmyt	Fraction of the momentum cell at the azimuthal θ or y face that is associated with flow from the upstream cell.
wmz	Fraction of the momentum cell at the axial z face that is associated with flow from the upstream cell.
wmxr	Fraction of the momentum cell at the radial r or x face that is associated with flow flow the upstream cell.
dyt	Cell length in the y direction or azimuthal θ sector angle in radians.
dzz	Cell length in the axial z direction.
dxr	Cell length in the radial r or x direction.
rdyt	Reciprocal of DYT.
rdz	Reciprocal of DZZ.
rdxr	Reciprocal of DXR.
rmean	Radius to the cell center.
rdyta	Reciprocal of the momentum cell length in the azimuthal θ or y direction.
rdza	Reciprocal of the momentum cell's length in the axial z direction.

rdxra	Reciprocal of the momentum cell's length in the radial r or x direction.
rddyt	The maximum of RDYTA and FA/VOL of the momentum cell in the azimuthal θ or y direction.
rddz	The maximum of RDZA and FA/VOL of the momentum cell in the axial z direction.
rddxr	The maximum of RDXRA and FA/VOL of the momentum cell in the radial r or x direction.
alpo	Gas-volume fraction at the start of the previous step (α^{n-1}).
dalva	Variable not used.
dalp	Weighting factor for the new-time level contribution to outflow in the basic mass and energy equations.
favyt	Donor-cell-averaged, gas-volume fraction at the azimuthal θ or y face.
favz	Donor-cell-averaged, gas-volume fraction at the axial z face.
favxr	Donor-cell-averaged, gas-volume fraction at the radial r or x face.
falyt	Donor-cell-averaged, liquid-volume fraction at the azimuthal θ or y face.
falz	Donor-cell-averaged, liquid-volume fraction at the axial z face.
falxr	Donor-cell-averaged, liquid-volume fraction at the radial r or x face.
frvyt	Product of the donor-cell-averaged gas macroscopic density, flow area, and timestep size at the azimuthal θ or y face.
frvz	Product of the donor-cell-averaged gas macroscopic density, flow area, and timestep size at the axial z face.
frvxr	Product of the donor-cell-averaged gas macroscopic density, flow area, and timestep size at the radial r or x face.
fevyt	Product of the donor-cell-averaged gas internal energy, flow area, and timestep size at the azimuthal θ or y face.
fevz	Product of the donor-cell-averaged gas internal energy, flow area, and timestep size at the axial z face.
!	
fevxr	Product of the donor-cell-averaged gas internal energy, flow area, and timestep size at the radial r or x face.
frayt	Product of the donor-cell-averaged, noncondensable-gas macroscopic density, flow area, and timestep size at the azimuthal θ or y face.
fraz	Product of the donor-cell-averaged, noncondensable-gas macroscopic density, flow area, and timestep size at the axial z face.

fraxr	Product of the donor-cell-averaged, noncondensable-gas macroscopic density, flow area, and timestep size at the radial r or x face.
frlyt	Product of the donor-cell-averaged liquid macroscopic density, flow area, and timestep size at the azimuthal θ or y face.
frlz	Product of the donor-cell-averaged liquid macroscopic density, flow area, and timestep size at the axial z face.
frlxr	Product of the donor-cell-averaged liquid macroscopic density, flow area, and timestep size at the radial r or x face.
felyt	Product of the donor-cell-averaged liquid internal energy, flow area, and timestep size at the azimuthal θ or y face.
felz	Product of the donor-cell-averaged liquid internal energy, flow area, and timestep size at the axial z face.
felxr	Product of the donor-cell-averaged liquid internal energy, flow area, and timestep size at the radial r or x face.
c#pm	Variables used as temporaries in a number of routines. Also the coefficient of the change in pressure across the m^{th} cell face in the equation for the $\#^{th}$ primary dependent variable in the basic step. The variables in order for #= 1,2,3,4,5 are total pressure, gas temperature, liquid temperature, gas-volume fraction, and partial pressure of noncondensable gas. The faces in order from m = 1,2,3,4,5,6 are the lower-numbered radial r or x face, the higher-numbered radial r or x face, the lower-numbered azimuthal θ or y face, the higher-numbered azimuthal θ or y face, the lower-numbered axial z face, and the higher-numbered axial z face.
dprhs	Iterate change in pressure during the basic step before inclusion of effects due to the relative change in pressure across the cell faces.
darhs	Iterate change in gas-volume fraction during the basic step before inclusion of effects due to the relative change in pressure across the cell faces.
dtvrhs	Iterate change in gas temperature during the basic step before inclusion of effects due to the relative change in pressure across the cell faces.
dtlrhs	Iterate change in liquid temperature during the basic step before inclusion of effects due to the relative change in pressure across the cell faces.
dparhs	Iterate change in the partial pressure of the noncondensable gas during the basic step before inclusion of effects due to the relative change in pressure across the cell faces.
fbit	Time-independent bit flags.

dvvs1	Scale factor applied to the derivative of the gas velocity at the outer radial r or x face with respect to cell pressure for the water-packing model.
dvvs1m	Scale factor applied to the derivative of the gas velocity at the inner radial r or x face with respect to cell pressure for the water-packing model.
dvls1	Scale factor applied to the derivative of the liquid velocity at the outer radial r or x face with respect to cell pressure for the water-packing model.
dvls1m	Scale factor applied to the derivative of the liquid velocity at the inner radial r or x face with respect to cell pressure for the water-packing model.
sc1	Area-ratio scale factor applied to the outer radial r or x face convecting velocities for cross-term contribution to the azimuthal-θ or y and axial-z motion equations.
sc1m	Area-ratio scale factor applied to the inner radial r or x face convecting velocities for cross-term contribution to the azimuthal-θ or y and axia-z motion equations.
dvvs3	Scale-factor applied to the derivative of the gas velocity at the upper axial z face with respect to cell pressure for the water-packing model.
dvvs3m	Scale factor applied to the derivative of the gas velocity at the lower axial z face with respect to cell pressure for the water-packing model.
dvls3	Scale factor applied to the derivative of the liquid velocity at the upper axial z face with respect to cell pressure for the water-packing model.
dvls3m	Scale factor applied to the derivative of the liquid velocity at the lower axial z face with respect to cell pressure for the water-packing model.
sc3	Area-ratio scale factor applied to the upper axial z face convecting velocities for cross-term contribution to the radial-r or x and azimuthal-θ or y motion equations.
sc3m	Area-ratio scale factor applied to the lower axial z face convecting velocities for cross-term contribution to the radial-r or x and azimuthal-θ or y motion equations.
dvvs2	Scale factor applied to the derivative of the gas velocity at the forward azimuthal-θ or y face with respect to cell pressure for the water-packing model.
dvls2	Scale factor applied to the derivative of the liquid velocity at the forward azimuthal-θ or y face with respect to cell pressure for the water-packing model.

sc2	Area-ratio scale factor applied to the forward azimuthal-θ– or y-face convecting velocities for cross-term contribution to the radial-r or x and axial-z motion equations.
!	
scd1	Area-ratio scale factor associated with the outer face used in the diagonal V del V term in the radial-r or x motion equation.
scd1m	Area-ratio scale factor associated with the inner face used in the diagonal V del V term in the radial-r or x motion equation.
scd2	Area-ratio scale factor associated with the forward face used in the diagonal V del V term in the azimuthal-θ or y motion equation.
scd3	Area-ratio scale factor associated with the upper face used in the diagonal V del V term in the axial-z motion equation.
scd3m	Area-ratio scale factor associated with the lower face used in the diagonal V del V term in the axial-z motion equation.
bit	Bit flags from the previous timestep.
frci1	Variable not used.
frci2	Variable not used.
frci3	Variable not used.
ciyt	Old interfacial-drag coefficient at the azimuthal θ or y face.
ciz	Old interfacial-drag coefficient at the axial z face.
cixr	Old interfacial-drag coefficient at the radial r or x face.
!	
chti	Old vapor interfacial HTC times the interfacial area.
chtia	Old noncondensable-gas interfacial HTC times the interfacial area.
alv	Old flashing interfacial HTC times the interfacial area.
alve	Old liquid interfacial HTC times the interfacial area.
arv	Old stabilizer macroscopic gas density, ar_v.
conco	Old ratio of the solute mass to the liquid mass.
pa	Old noncondensable-gas partial pressure.
roa	Old noncondensable-gas density.
ea	Old noncondensable-gas internal energy.
alp	Old gas-volume fraction.
rov	Old gas density.
rol	Old liquid density.
s	Old solute mass plated out on structure.
vvyt	Old basic gas velocity at the azimuthal q or y face.
vvz	Old basic gas velocity at the axial z face.

vvxr	Old basic gas velocity at the radial r or x face.
vlyt	Old basic liquid velocity at the azimuthal q or y face.
vlz	Old basic liquid velocity at the axial z face.
vlxr	Old basic liquid velocity at the radial r or x face.
ev	Old gas internal energy.
el	Old liquid internal energy.
tv	Old gas temperature.
tl	Old liquid temperature.
gam	Old vapor generation rate per unit volume.
p	Old total pressure.
arev	Old stabilizer gas internal energy, $\alpha\rho_v e_v$..
vvtyt	Old stabilizer gas velocity at the azimuthal q or y face.
vvtz	Old stabilizer gas velocity at the axial z face.
vvtxr	Old stabilizer gas velocity at the radial r or x face.
arl	Old stabilizer $(1-\alpha)\rho_\ell$.
arel	Old stabilizer $(1-\alpha)\rho_\ell e_\ell$.
vltyt	Old stabilizer liquid velocity at the azimuthal q or y face.
vltz	Old stabilizer liquid velocity at the axial z face.
vltxr	Old stabilizer liquid velocity at the radial r or x face.
ara	Old stabilizer $\alpha\rho_a$.
owvyt	Old donor-cell factor at the azimuthal q or y face for gas.
owvz	Old donor-cell factor at the axial z face for gas.
owvxr	Old donor-cell factor at the radial r or x face for gas.
owlyt	Old donor-cell factor at the azimuthal q or y face for liquid.
owlz	Old donor-cell factor at the axial z face for liquid.
owlxr	Old donor-cell factor at the radial r or x face for liquid.
bitn	Bit flags for the current timestep.
frci1n	Variable not used.
frci2n	Variable not used.
frci3n	Variable not used.
cinyt	New interfacial-drag coefficient at the azimuthal q or y face.
cinz	New interfacial-drag coefficient at the axial z face.
cinxr	New interfacial-drag coefficient at the radial r or x face.
chtin	New vapor interfacial HTC times the interfacial area.
chtan	New noncondensable-gas interfacial HTC times the interfacial area.

!

alvn	New flashing interfacial HTC times the interfacial area.
alven	New liquid interfacial HTC times the interfacial area.
arvn	New stabilizer $\alpha\rho_v$.
conc	New ratio of solute mass to liquid mass.
pan	New noncondensable-gas partial pressure.
roan	New noncondensable-gas density.
ean	New noncondensable-gas internal energy.
alpn	New gas-volume fraction.
rovn	New gas density.
roln	New liquid density.
sn	New solute mass plated out on structure surfaces.
vvnyt	New basic gas velocity at the azimuthal q or y face.
vvnz	New basic gas velocity at the axial z face.
vvnxr	New basic gas velocity at the radial r or x face.
vlnyt	New basic liquid velocity at the azimuthal q or y face.
vlnz	New basic liquid velocity at the axial z face.
vlnxr	New basic liquid velocity at the radial r or x face.
evn	New gas internal energy.
eln	New liquid internal energy.
tvn	New gas temperature.
tln	New liquid temperature.
gamn	New vapor generation rate per unit volume.
pn	New total pressure.
arevn	New stabilizer $\alpha\rho_v e_v$.
vvntyt	New stabilizer gas velocity at the azimuthal q or y face.
vvntz	New stabilizer gas velocity at the axial z face.
vvntxr	New stabilizer gas velocity at the radial r or x face.
arln	New stabilizer $(1-\alpha)\rho_\ell$.
areln	New stabilizer $(1-\alpha)\rho_\ell e_\ell$.
vlntyt	New stabilizer liquid velocity at the azimuthal q or y face.
vlntz	New stabilizer liquid velocity at the axial z face.
vlntxr	New stabilizer liquid velocity at the radial r or x face.
aran	New stabilizer $\alpha\rho_a$.
wvyt	New donor-cell factor at the azimuthal q or y face for gas.
wvz	New donor-cell factor at the axial z face for gas.

wvxr	New donor-cell factor at the radial r or x face for gas.
wlyt	New donor-cell factor at the azimuthal q or y face for liquid.
wlz	New donor-cell factor at the axial z face for liquid.
wlxr	New donor-cell factor at the radial r or x face for liquid.
spifz	Stratified-flow weighting factor for the interfacial heat-transfer correlations.
dvvs2m	Scale-factor applied to the derivative of the gas velocity at the backward azimuthal q or y face with respect to cell pressure for the water-packing model.
dvls2m	Scale factor applied to the derivative of the liquid velocity at the backward azimuthal q or y face with respect to cell pressure for the water-packing model.
sc2m	Area-ratio scale factor applied to the backward azimuthal q or y face convecting velocities for cross-term contribution to the radial-r or x and axial-z motion equations.
scd2m	Area-ratio scale factor associated with the backward face used in the diagonal V del V term in the azimuthal-q or y motion equation.
xcl	
xl1	
xl2	
xl3	
xl4	
xl5	
xl6	
xls	
xcv	
xv1	
xv2	
xv3	
xv4	
xv5	
xv6	
xvs	

C.66. Module VessCon

```
INTEGER(sik) nxrmx,nytmx
PARAMETER(nxrmx=21,nytmx=12)
INTEGER(sik) nrfmx,nzfmx
```

```
PARAMETER(nrfmx=20,nzfmx=250)
INTEGER(sik) nms,ndms
PARAMETER(nms=10,ndms=7+nms)
!
INTEGER(sik) nxbcm,nybcm,nzbcm
PARAMETER(nxbcm=2,nybcm=2,nzbcm=2)
INTEGER(sik) nxbcp,nybcp,nzbcp
PARAMETER(nxbcp=1,nybcp=1,nzbcp=1)
INTEGER(sik) nxryt
PARAMETER (nxryt=nxrmx*nytmx)
INTEGER(sik) nrfmx1,nrzfmx
PARAMETER(nrfmx1=nrfmx+1,nrzfmx=nrfmx*nzfmx)
PARAMETER(jc0p=1+nybcm)
PARAMETER(kc0p=1+nzbcm)
PARAMETER(jc0mp=jc0p-1,kc0mp=kc0p-1)
PARAMETER(jc0mmp=jc0p-nybcm,kc0mmp=kc0p-nzbcm)
PARAMETER(iseq=1,imfreq=1,idrpeq=0,nfr3eq=2)
```

INTEGER VARIABLES:

nxrmx	
nytmx	
nrfmx	
nzfmx	
nms	
ndms	
nv	
nxbcm	
nybcm	
nzbcm	
nxbcp	
nybcp	
nzbcp	
nxryt	nxrmx*nytmx
nrfmx1	nrfmx+1
nrzfmx	nrfmx*nzfmx
jc0p	1+nybcm
kc0p	1+nzbcm
jc0mp	jc0p-1

kc0mp	kc0p-1
jc0mmp	jc0p-nybcm
kc0mmp	kc0p-nzbcm
iseq	
imfreq	
idrpeq	
nfr3eq	

C.67. Module `VessMat`

INTEGER(sik) lbvec,lbw,ldmat,lemat,lrmat,lvmat,lvssc,lvssip

INTEGER(sik) nclear

!

REAL(sdk), DIMENSION(:),ALLOCATABLE :: vmap

REAL VARIABLES:

lbvec	Pointer for storing in the a array the capacitance-matrix equation right-hand-side vector.
lbw	Number of element rows in the array that stores the VESSEL-banded coefficient matrix.
ldmat	Pointer for storing in the a array the capacitance coefficient matrix.
lemat	Pointer for storing in the a array the E matrix of the capacitance-matrix method.
lrmat	Pointer for storing in the a array the R matrix of the capacitance-matrix method.
lvmat	VESSEL matrix storage for coarse-mesh rebalance or direct inversion.
lvssc	Right-hand side of equation associated with `lvmat`.
lvssip	Pivoting information for `lvmat`.
vmap	

INTEGER VARIABLES:

nclear	Number of values in the a (`lvmat`) array storing the VESSEL-banded coefficient matrix.

C.68. Module `VessTf3dc`

INTEGER(sik) insct,iz,kabso,kcmsh,kl,klev,ku,kvel1,kvel2,kvel3

INTEGER VARIABLES:

insct	Variable used to obtain a displacement into network arrays involving VESSEL junctions when there is more than one VESSEL.
iz	VESSEL level number currently being evaluated.
kabso	Storage offset to obtain an absolute cell number when multiple VESSELs are used.
kcmsh	Offset for coarse-mesh indexing with multiple VESSELs.
kl	Displacement of level (iz–1) from level (iz) in A-array storage for the VESSEL 3D data array.
klev	VESSEL component axial-direction K index [the axial-level number IZ plus NZBCM (for two lower pseudo-cell levels)].
ku	Displacement of level (iz+l) from level (iz) in A-array storage for the VESSEL 3D data array.
kvel1	Order of the r- or x-direction stabilizer motion-equation matrix for the present VESSEL component.
kvel2	Order of the θ- or y-direction stabilizer motion-equation matrix for the present VESSEL component.
kvel3	Order of the z-direction stabilizer motion-equation matrix for the present VESSEL component.

C.69. Module `VessVlt`

TYPE vessTabT
- REAL(sdk) bsmass
- REAL(sdk) bsmssn
- REAL(sdk) cimfr
- REAL(sdk) cimfrl
- REAL(sdk) cimfrv
- REAL(sdk) comfr
- REAL(sdk) comfrl
- REAL(sdk) comfrv
- REAL(sdk) corelq
- REAL(sdk) dcflow
- REAL(sdk) dclqvl
- REAL(sdk) epsw
- REAL(sdk) gcc
- REAL(sdk) geomfc
- REAL(sdk) gravz
- REAL(sdk) pcore
- REAL(sdk) pdc

REAL(sdk) plp
REAL(sdk) pup
REAL(sdk) qhstot
REAL(sdk) shelv
REAL(sdk) tcilmf
REAL(sdk) tcivmf
REAL(sdk) tcolmf
REAL(sdk) tcore
REAL(sdk) tcovmf
REAL(sdk) tdc
REAL(sdk) tlp
REAL(sdk) tscore
REAL(sdk) tsdc
REAL(sdk) tslp
REAL(sdk) tsup
REAL(sdk) tup
REAL(sdk) vbmass
REAL(sdk) vbmssn
REAL(sdk) vcore
REAL(sdk) vdclq
REAL(sdk) vfmass
REAL(sdk) vfmssn
REAL(sdk) vlcore
REAL(sdk) vlpliq
REAL(sdk) vlplm
REAL(sdk) vlplq
REAL(sdk) vlqmss
REAL(sdk) voldc
REAL(sdk) vollp
REAL(sdk) volup
REAL(sdk) vsflow
REAL(sdk) vupliq
REAL(sdk) vuplm
! REAL(sdk) vrmatsm(3,3)
! REAL(sdk) r0vsm(3)
! REAL(sdk) x0vsm
! REAL(sdk) y0vsm

```
!     REAL(sdk) z0vsm
!     REAL(sdk) tx0vsm
!     REAL(sdk) ty0vsm
!     REAL(sdk) tz0vsm
      INTEGER(sik) iall
      INTEGER(sik) ic0
      INTEGER(sik) ic0m
      INTEGER(sik) ic0mm
      INTEGER(sik) iconc
      INTEGER(sik) icrl
      INTEGER(sik) icrr
      INTEGER(sik) icru
      INTEGER(sik) icx
      INTEGER(sik) icxp
      INTEGER(sik) idcl
      INTEGER(sik) idcr
      INTEGER(sik) idcu
      INTEGER(sik) iext
      INTEGER(sik) if0
      INTEGER(sik) ifx
      INTEGER(sik) igbc
      INTEGER(sik) igbcxr
      INTEGER(sik) igbcyt
      INTEGER(sik) igbcz
      INTEGER(sik) igeom
      INTEGER(sik) ilcsp
      INTEGER(sik) iucsp
      INTEGER(sik) iuhp
      INTEGER(sik) ivssbf
      INTEGER(sik) jall
      INTEGER(sik) jc0
      INTEGER(sik) jc0m
      INTEGER(sik) jc0mm
      INTEGER(sik) jcx
      INTEGER(sik) jcxp
      INTEGER(sik) jf0
      INTEGER(sik) jfx
```

```
      INTEGER(sik) kall
      INTEGER(sik) kc0
      INTEGER(sik) kc0m
      INTEGER(sik) kc0mm
      INTEGER(sik) kcx
      INTEGER(sik) kcxp
      INTEGER(sik) kf0
      INTEGER(sik) kfx
      INTEGER(sik) lenld
      INTEGER(sik) lenldo
      INTEGER(sik) locvsp
      INTEGER(sik) lstvsp
      INTEGER(sik) ltempl
      INTEGER(sik) nasx
      INTEGER(sik) ncells
      INTEGER(sik) nclx
      INTEGER(sik) ncrx
      INTEGER(sik) ncsr
      INTEGER(sik) nijt
      INTEGER(sik) nrsx
      INTEGER(sik) nsgrid
      INTEGER(sik) ntsx
      INTEGER(sik) nvent
      INTEGER(sik) nvvtb
      INTEGER(sik) nxrv
      INTEGER(sik) nytv
      INTEGER(sik) nzzv
!     INTEGER(sik) ncshm
!     INTEGER(sik) ntism
!     INTEGER(sik) nzism
END TYPE vessTabT
!
TYPE(vessTabT),DIMENSION(maxComps) :: vessTab
!
INTEGER(sik) vessDumpSize
PARAMETER (vessDumpSize=110)
  DERIVED TYPE VARIABLES:
```

vessTAB	The VLT for theVESSEL component.

REAL VARIABLES:

!Elements of DERIVED TYPE vessTabT

bsmass	Integrated fluid mass flow from the VESSEL at the start of the timestep.
bsmssn	Integrated fluid mass flow from the VESSEL at the end of the timestep.
cimfr	Core-inlet fluid mass-flow rate.
cimfrl	Core-inlet liquid mass-flow rate.
cimfrv	Core gas-volume fraction.
comfr	Core-outlet fluid mass-flow rate.
comfrl	Core-outlet liquid mass-flow rate.
comfrv	Core-outlet gas mass-flow rate.
corelq	Core liquid volume fraction.
dcflow	Downcomer-fluid, mass-flow rate.
dclqvl	Downcomer-liquid-volume fraction.
epsw	Wall surface roughness.
gcc	Gravity-acceleration constant.
geomfc	Geometry factor (1.0 = cylindrical, 0.0 = Cartesian).
gravz	GRAV component in the z direction.
pcore	Core average pressure.
pdc	Downcomer average pressure.
plp	Lower-plenum average pressure.
pup	Upper-plenum average pressure.
qhstot	Total heat flux from HTSTR components coupled to the VESSEL.
shelv	An addition to the input Z coordinates to give elevations for computing GRAV in 1D.
tcilmf	Integrated core-inlet liquid mass flow.
tcivmf	Integrated core-inlet gas mass flow.
tcolmf	Integrated core-outlet liquid mass flow.
tcore	Core average liquid temperature.
tcovmf	Integrated core-outlet gas mass flow.
tdc	Downcomer average liquid temperature.
tlp	Lower-plenum average liquid temperature.
tscore	Core average saturation temperature.
tsdc	Downcomer average saturation temperature.
tslp	Lower-plenum average saturation temperature.

tsup	Upper-plenum average saturation temperature.
tup	Upper-plenum average liquid temperature.
vbmass	Old fluid mass flow to internal BREAKs.
vbmssn	New fluid mass flow to internal BREAKs.
vcore	Total liquid mass in the core.
vdclq	Total liquid mass in the downcomer.
vfmass	Old fluid mass flow from internal FILLs.
vfmssn	New fluid mass flow from internal FILLs.
vlcore	Core liquid mass.
vlpliq	Lower-plenum liquid volume fraction.
vlplm	Lower-plenum liquid mass.
vlplq	Total liquid mass in the lower plenum.
vlqmss	Total liquid mass in the VESSEL.
voldc	Downcomer volume.
vollp	Lower-plenum volume.
volup	Upper-plenum volume.
vsflow	VESSEL mass flow.
vupliq	Upper-plenum liquid volume fraction.
vuplm	Upper-plenum liquid mass.
vrmatsm(3,3)	Special-purpose, DOE-model parameter.
r0vsm(3)	Special-purpose, DOE-model parameter.
x0vsm	Special-purpose, DOE-model parameter.
y0vsm	Special-purpose, DOE-model parameter.
z0vsm	Special-purpose, DOE-model parameter.
tx0vsm	Special-purpose, DOE-model parameter.
ty0vsm	Special-purpose, DOE-model parameter.
tz0vsm	Special-purpose, DOE-model parameter.

INTEGER VARIABLES:

!Elements of DERIVED TYPE vessTabT

iall	`icxl` + `nxbcp*nv`.
ic0	`ic0mm` + `nxbcm*nv`.
ic0m	`ic0mm` + `(nxbcm-1)*nv`.
ic0mm	A array starting location for the VESSEL 3D arrays.
iconc	Presence of solute in the liquid option.
icrl	Core lower-boundary segment number, `z(icrl)`.
icrr	Core outer radial-boundary segment number, `rad(icrr)`.
icru	Core upper-boundary segment number, `z(icru)`.

icx	ic0mm + (nxbcm + nxr-2)*nv.
icxp	icxl + nv.
idcl	Downcomer lower-boundary segment number, z(idcl).
idcr	Downcomer radial-boundary segment number, rad(idcr).
idcu	Downcomer upper-boundary segment number, z(idcu).
iext	Specifies if the VESSEL input was generated by the post-processor EXTRACT. 0 = no; 1 = yes.
if0	ic0ml if igeom.eq.1.and.igbcxr.eq.1, else ic01.
ifx	if01 + (nxrv-1)*nv.
igbc	igbcxr + igbcyt + igbcz.
igbcxr	Flag (0 or 1) for internal pressure/velocity boundary conditions on the x- or r-direction walls of the VESSEL.
igbcyt	Flag (0 or 1) for internal pressure/velocity boundary conditions on the y- or θ- direction walls of the VESSEL.
igbcz	Flag (0 or 1) for internal pressure/velocity boundary conditions on the z-direction walls of the VESSEL.
igeom	Vessel-geometry option. 0 = cylindrical geometry; 1 = Cartesian geometry.
ilcsp	Lower-core support-plate axial segment number.
iucsp	Upper-core support-plate axial segment number.
iuhp	Upper head-plate axial segment number.
ivssbf	Internal VESSEL boundary condition. 0 = no internal boundaries (default); 2 = first axial level acts as a FILL, last axial level acts as a BREAK; 20 = first axial level acts as a BREAK, last axial level acts as a FILL; 22 = both the first and last axial levels act as BREAKs.
jall	jcx + nybcp.
jc0	nybcm + 1.
jc0m	nybcm.
jc0mm	nybcm – 1.
jcx	jc0 + nyt – 1.
jcxp	jcx + 1.
jf0	jc0m if igeom.eq.1 .and. igbcyt.eq.1, else jc0.

jfx	jf0 + nyt – 1.
kall	kcx + nzbcp.
kc0	nzbcm + 1.
kc0m	nzbcm.
kc0mm	nzbcm – 1.
kcx	kc0 + nzz – 1.
kcxp	kcx + 1.
kf0	kc0m if igbcz.eq.1, else kc0.
kfx	kf0 + nzz – 1.
lenld	Length of level data.
lenldo	Defined to be zero (variable not used).
locvsp	Beginning offset for the VESSEL pointer table.
lstvsp	Length of the VESSEL pointer table.
ltempl	Location of temporary space in the A array to contain one level of data for level-data editing (calculated, but variable not used).
nasx	Number of axial segments (levels).
ncells	Number of fluid cells.
nclx	Number of fluid cells per level.
ncrx	Number of core cells per level.
ncsr	Number of source connections to VESSEL cells.
nijt	ni * jall.
nrsx	Number of radial segments.
nsgrid	Number of spacer grids present in the core region (spacer grids are modeled only when the reflood model has been selected by inputting namelist variable newrfd = 1).
ntsx	Number of θ segments.
nvent	Number of cells with vent valves in the outer radial surface.
nvvtb	Number of input data pairs in the multiple-point, vent-valve table.
nxrv	nrsx if igeom.eq.0 .and. igbcxr.ne.0, else nrsxh if igeom. ne.0 .and. igbcxr.ne.0, else nrsx–1.
nytv	ntsx–1 if igeom.eq.0 .and. igbcyt.eq.0, else ntsx+1 if igeom.ne.0 .and. igbcyt.ne.0, else 0 if igeom.eq.0 .and. ntsx.eq.1, else ntsx.
nzzv	nasx+1 if igbcz.ne.0, else nasx–1.
nzism	Special-purpose, DOE-model parameter.
ncshm	Special-purpose, DOE-model parameter.
ntism	Special-purpose, DOE-model parameter.

!

vessDumpSize Number of variables in the derived-type vessTabT.

C.70. Module xtv

```
INTEGER(sik) nvname1,nvname3,nvnameh,nvnamep
PARAMETER (nvname1=15,nvname3=24,nvnameh=12,nvnamep=10)
CHARACTER*30 vname1(nvname1),vname3(nvname3)
CHARACTER*30 vnameh(nvnameh)
CHARACTER*30 vnamep(nvnamep)
!
INTEGER(sik) uxtvh,ndmp,fflag1(nvname1),fflag3(nvname3),fflagh    &
& (nvnameh)
```

INTEGER VARIABLES:

nvname1	Parameter used to to dimension vname1.
nvname3	Parameter used to to dimension vname3.
nvnameh	Parameter used to to dimension vnameh.
nvnamep	Parameter used to to dimension vnamep.
uxtvh	
ndmp	
fflag1	
fflag3	
fflagh	

CHARACTER VARIABLES:

vname1	XTV-graphics variable names for 1D hydraulic components.
vname3	XTV-graphics variable names for 3D VESSEL components.
vnameh	XTV-graphics variable names for HTSTR ROD or SLAB components.
vnamep	XTV-graphics variable names for PLENUM components.

C.71. Module xvol

```
REAL(sdk) bgss,dawl,daxvl,daxvu,dgss,frev
INTEGER(sik) ifvt,ifvtu,ldax
!
DATA frev/0.001d0/,dgss,bgss/0.99d0,0.001d0/
DATA daxvl,daxvu/0.d0,1.d0/
DATA dawl/.5d0/,ifvt/0/,ldax/1/
```

REAL VARIABLES:

bgss	Limits on special gas-volume fraction prediction logic.
dawl	Weighting factors in special `tf1ds` flux logic.
daxvl	Lower-velocity limit on special `tf1ds` flux logic.
daxvu	Upper-velocity limit on special `tf1ds` flux logic.
dgss	Limits on special gas-volume-fraction prediction logic.
frev	Sensitivity level for reiteration on flow reversal.

INTEGER VARIABLES:

ifvt	Flag for setting velocities passed to `tf1ds` for special flux logic.
ifvtu	Time-of-velocity controller. 0 = xvset logic uses the old-time velocity; 1 = xvset logic uses the the new-time velocity.
ldax	Bypass switches on special `tf1ds` flux logic.

C.72. Include File (Common-Block) bandw

COMMON/bandw/ mux, muy, muz

INTEGER VARIABLES:

mux	The number of diagonal rows above and below the main diagonal lying within the mux+1+mux bandwidth of the VESSEL matrix for the x- or θ-directional stabilizer motion equation.
muy	The number of diagonal rows above and below the main diagonal lying within the muy+1+muy bandwidth of the VESSEL matrix for the y- or θ-direction stabilizer motion equation.
muz	The number of diagonal rows above and below the main diagonal lying within the muz+1+muz bandwidth of the VESSEL matrix for the z-direction stabilizer motion equation, pressure semi-implicit equation, and the stabilizer mass and energy equations.

C.73. Include file **bignum**

The include file `bignum.h` is NOT a common block, but rather contains simple data statements to initialize the arrays used to display the TRAC-M big numbers.

C.74. Include file (common-block) **cflow**

DIMENSION chm1(5),chm2(5)

COMMON /cflow/chm1,chm2,chmlt1,chmlt2

COMMON /cflow/icflow,ihor,ihorg

REAL VARIABLES:

chm1 Five sets of choked-flow multipliers for subcooled critical flow.

chm2 Five sets of choked-flow multipliers for two-phase critical flow.

chmlt1 Default multiplier for subcooled critical flow.

chmlt2 Default multiplier for two-phase critical flow.

INTEGER VARIABLES:

icflow Choked-flow option (namelist variable).

0 = model turned off;

1 = model using default multipliers turned on only for components connected to a BREAK (default value);

2 = model using optional multipliers turned on at cell edges defined by component input (note that this option requires additional array data for all 1D hydrodynamic components).

ihor Wall-drag form option (namelist variable).

0 = uses dispersed drag only;

1 = uses stratified drag in one dimension if conditions are met (default value);

2 = always uses stratified drag;

3 = turns off head gradient force.

ihorg Variable not used.

C.75. Include file (common-block) `chfint`

COMMON/chfint/ alpchf

REAL VARIABLE:

alpchf Gas-volume fraction at the CHF location.

C.76. Include file (common-block) `chgalp`

COMMON/chgalp/ dal, dau, oal, oau, xdal, xdau, xoal, xoau

COMMON/chgalp/ jdal, jdau, joal, joau, ndal, ndau, noal, noau

REAL VARIABLES:

dal Maximum decrease in the gas-volume fraction over the timestep.

dau Maximum increase in the gas-volume fraction over the timestep.

oal Maximum decrease in the gas-volume fraction immediately following an increase.

oau Maximum increase in the gas-volume fraction immediately following a decrease.

xdal Limit on `dal` beyond which the timestep is reduced.

xdau Limit on dau beyond which the timestep is reduced.

xoal Limit on oal beyond which the timestep is reduced.

xoau Limit on oau beyond which the timestep is reduced.

INTEGER VARIABLES:

jdal Cell where dal occurred.

jdau Cell where dau occurred.

joal Cell where oal occurred.

joau Cell where oau occurred.

ndal Component where dal occurred.

ndau Component where dau occurred.

noal Component where oal occurred.

noau Component where oau occurred.

C.77. Include file (common-block) `ciflim`

COMMON/ciflim/ fifi, fifr

REAL VARIABLES:

fifi Maximum decrease factor for the time-constant constraint on the interfacial-drag coefficient (0.4).

fifr Maximum increase factor for the time-constant constraint on the interfacial-drag coefficient (2.0).

C.78. Include file (common-block) `cnrslv`

COMMON/cnrslv/aa(nrfmx1,nrzfmx),bb(nrzfmx),w(nrzfmx)

COMMON/cnrslv/key,m,m1,n,nrslv

COMMON /cnrslv/err

REAL VARIABLES:

aa Coefficient matrix.

bb Right-hand-side (known) vector.

w Working-area vector.

INTEGER VARIABLES:

nrfmx1

nrzfmx

key Evaluation-flag option.

1 = solves the linear matrix equation by forward-elimination and backward-substitution.

2 = performs the forward-elimination only.

3 = performs the backward-substitution only.

m — Number of r- or x-direction nodes in the heat-transfer mesh that defines the matrix aa bandwidth of m+1+m.

m1 — m + 1.

n — Order of matrix a that is stored in matrix aa.

nrslv — Axial-direction, heat-transfer-calculation numerics option (namelist variable).

0 = evaluate axial direction explicitly (default);

1 = evaluate axial direction implicitly.

LOGICAL VARIABLE:

err — Error flag from subroutine bansol that indicates a singular matrix when .TRUE.

C.79. Include file (common-block) concck

COMMON/concck/ jflagc

INTEGER VARIABLE:

jflagc — Flag that indicates an error in specifying the 1D-component input-parameter values.

C.80. Include file (common-block) condht

COMMON/condht/ ylv, yll

REAL VARIABLES:

ylv — Axial distance above node row jl where the gas-liquid interface is located.

yll — Axial distance above node row jl where the gas-liquid interface is located.

C.81. Include file (common-block) constant

COMMON/const/ pi, gc, zero, one, epsalp, explim

REAL VARIABLES:

pi — Constant pi (3.1415926535898).

gc — Gravitational constant (9.80665 $m \cdot s^{-2}$).

zero — Real constant zero.

one — Real constant one.

epsalp — Gas-volume-fraction cutoff for thermodynamic vapor properties.

explim	Maximum value of the exponent for time-constant constraint of the heat-transfer coefficients.

C.82. Include file (common-block) `decayc`

COMMON/decayc/ fisphi, fp235, fp238, fp239, qavg, q235, q238, q239, rans, r239pf, topate

COMMON/decayc/ ians79

REAL VARIABLES:

fisphi	Fissions per initial fissile atom.
fp235	Fraction of reactor-core power from ^{235}U fissions.
fp238	Fraction of reactor-core power from ^{238}U fissions.
fp239	Fraction of reactor-core power from ^{239}Pu fissions.
qavg	Average energy per fission.
q235	Energy per fission from ^{235}U.
q238	Energy per fission from ^{238}U.
q239	Energy per fission from ^{239}Pu.
rans	Multiplier applied to the ANS79 decay heat.
r239pf	Atoms of ^{239}U produced per fission.
topate	Four years in seconds units.

INTEGER VARIABLE:

ians79 — ANS79 decay-heat standard evaluation flag.

- 0 = not evaluated;
- 1 = evaluate the 69-group ANS79 decay-heat standard;
- 2 = evaluate the ANS79 decay-heat standard and the heavy-metal decay for ^{239}U and ^{239}Np.

C.83. Include file (common-block) `defval`

COMMON/ defval/ alpq,hstnq,pq,paq,qpppq,tlq,tvq,twq,vlq,vvq,cfz3q, hd3q

COMMON/ defval/ istopt

REAL VARIABLES:

alpq	Default value for initial gas-volume fractions input through namelist and used to specify gas-volume fractions when `istopt` = 1 or 2.
hstnq	Default value for initial HTSTR temperatures input through namelist and used to specify the HTSTR temperatures when `istopt` = 1 or 2.

pq — Default value for initial pressures input through namelist and used to specify pressures when `istopt` = 1 or 2.

paq — Default value for initial noncondensable-gas partial pressures input through namelist and used to specify noncondensable-gas partial pressures when `istopt` = 1 or 2.

qpppq — Default value for initial volumetric heat sources in flow channel walls input through namelist and used to specify volumetric heat sources when `istopt` = 1 or 2.

tlq — Default value for initial liquid temperatures input through namelist and used to specify liquid temperatures when `istopt` = 1 or 2.

tvq — Default value for initial gas temperatures input through namelist and used to specify gas temperatures when `istopt` = 1 or 2.

twq — Default value for initial wall temperatures input through namelist and used to specify wall temperatures when `istopt` = 1 or 2.

vlq — Default value for initial liquid velocities input through namelist and used to specify liquid velocities when `istopt` = 1 or 2.

vvq — Default value for initial gas velocities input through namelist and used to specify gas velocities when `istopt` = 1 or 2.

cfz3q — Default value for 3D VESSEL-component additive loss coefficients input through namelist and used to specify VESSEL additive loss coefficients when `istopt` = 1 or 2.

hd3q — Default value for 3D VESSEL component hydraulic diameters input through namelist and used to specify VESSEL hydraulic diameters when `istopt` = 1 or 2.

INTEGER VARIABLE:

istopt — Option for defining thermal-hydraulic parameter default values through namelist input (namelist variable).

C.84. Include file (common-block) diddle

COMMON/diddle/ alpcc, afct, alpshl, alpshu, encut, enfac1, enfac2, alw1, alw2, farea1, fareah, fareav, fse5, vrtcut, alpbct, vecvct, veclct, vintf, alplvu, alplvl, cbmin, calv2, vrbcut, vdrpf, vdrpmx, vlvcmx, enmin, scinan, tgrav

COMMON/diddle/ nifslb

REAL VARIABLES:

alpcc — Gas-volume fraction that gives the minimum value for the bubble condensation rate.

afct — Area scaling for waves on inverted annular interface.

alpshl — Gas-volume fraction below which the interface sharpener is off.

alpshu	Gas-volume fraction above which the interface sharpener is on.
encut	Minimum droplet-entrainment fraction.
enfac1	Scaling factor for minimum-entrainment velocity.
enfac2	Scaling factor for entrainment-correlation exponent.
alw1	Gas-volume fraction lower limit for transition from bubbly slug (at `alw1`= 0.5) to annular-mist (at `alw2` = 0.75) flow regimes.
alw2	Gas-volume fraction upper limit for transition from bubbly slug (at `alw1` = 0.5) to annular-mist (at `alw2` = 0.75) flow regimes.
farea1	Scale factor for 1D plug-flow condensation area.
fareah	Scale factor for 3D separated plug-flow horizontal condensation area.
fareav	Scale factor for 3D separated plug-flow vertical condensation area.
fse5	Scale factor for pool entrainment.
vrtcut	Minimum allowed relative speed for computing interfacial coefficients (except during boiling).
alpbct	Lower bound on bubble gas-volume fraction used to compute interfacial heat-transfer rates (and resulting Γ) when boiling.
vecvct	Lowest allowed gas speed when computing interfacial coefficients.
veclct	Lowest allowed liquid speed when computing interfacial coefficients.
vintf	Factor in expression for mean circulation speed in a moving drop.
alplvu	Highest value of the minimum adjacent gas-volume fraction allowed for calculating a plug interfacial area.
alplvl	Lowest value of the maximum adjacent gas-volume fraction allowed for calculating a plug interfacial area.
cbmin	Minimum allowed mean bubble diameter.
calv2	The liquid-side HTC for superheated drops.
vrbcut	Minimum allowed relative velocity for computing ALV during boiling.
vdrpf	Scale factor in the expression for the limiting circulation velocity in a drop.
vdrpmx	Maximum allowed internal circulation velocity in a drop.
vlvcmx	Maximum liquid velocity used for the condensation ALV correlation.
enmin	Variable not used.
scinan	Scale factor for inverted annular drag.

tgrav — Absolute value of grav (0.75) above which horizontal stratified flow can not exist.

INTEGER VARIABLE:

nifslb — If nonzero, then slabs should be used to test for inverted annular flow.

C.85. Include File (Common-Block) diddlh

COMMON/diddlh/ alp2, alp3, alpbr, alpcut, alpctr, alow, aup, aflml, aflmu, alpag, alpdf, fliler, fliles, fber, fbex, fdfhl, hgf, hgvmn, facthl3, fdalva, freq1, freq2, fudge1, fudge2, limflg

COMMON/diddlh/ ihtav, ihtcn, nshtcn

REAL VARIABLES:

alp2 — Gas-volume fraction (1.0) above which the gas is in forced convection.

alp3 — Gas-volume fraction (0.9999) above which there is no liquid heat transfer.

alpbr — Gas-volume fraction (0.99) above which liquid convection is linearly faired off and gas convection is faired on.

alpcut — Gas-volume fraction (0.98) above which nucleate boiling is not permitted (if other criteria are met).

alpctr — Gas-volume fraction (0.995).

alow — Lowest value of the gas-volume fraction in adjacent cells.

aup — Highest value of the gas-volume fraction in adjacent cells.

aflml — Gas-volume fraction below which Bromley film boiling contributes fully to the liquid.

aflmu — Gas-volume fraction above which no Bromley coefficient is added to the liquid.

alpag — Gas-volume fraction at the agitated-to-post-agitated, inverted-annular-flow transition boundary.

alpdf — Gas-volume fraction describing the beginning of the highly dispersed inverted annular flow.

fliler — Constant used to adjust the wall-to-liquid HTC obtained by modified Bromley correlation in reflood.

fliles — Same as fliler for nonreflood cases.

fber — Variable not used.

fbex — Power of $(1 - \alpha)$ weighting of the Bromley correlation.

fdfhl — A scaling factor for the wall-to-droplet, heat-transfer correlation.

hgf — Function of nucleate-boiling heat transfer, which contributed to subcooled boiling.

hgvmn	Cutoff velocity for condensation used to suppress subcooled nucleate boiling.
facthl3	Power to which (aloff2–alp)/(aloff2–alpx2) is raised.
fdalva	Multiplier of dalva(j), which is the rate of change of alw with respect to gas-volume fraction and is currently set to zero.
freq1	Time-constant constraint frequency for the maximum increase in interfacial heat-transfer and drag coefficients.
freq2	Time-constant constraint frequency for the maximum decrease in interfacial heat-transfer and drag coefficients.
fudge1	Time-constant constraint factor of maximum increase when the timestep size is 1.0/freq1.
fudge2	Time-constant constraint factor of maximum decrease when the timestep size is 1.0/freq2.

INTEGER VARIABLES:

limflg	Flag for evaluating time-constant constraint of the evaporation and condensation rate coefficients. 0 = no; 1 = yes.
ihtav	Variable is normally 1. When ihtav is 0, there is no time averaging of HTCs.
ihtcn	Variable is normally 0. When ihtcn is 1, HTCs are forced to remain constant.
nshtcn	Variable is normally 10 000 000. If nstep > nshtcn, then ihtch is set to 1 (for debugging only).

C.86. Include File (Common-Block) diddli

COMMON/diddli/ smivx, nscool, iiabk

REAL VARIABLE:

smivx	Constant value 1.5 (variable not used).

INTEGER VARIABLES:

nscool	Flag (when having its default value of 1) that constrains the subcooled boiling heat flux to not exceed the wall heat flux to the liquid.
iiabk	Constant value 1 (variable not used).

C.87. Include File dlimit (Common-Block dlim)

DIMENSION nlim(9), nlim2(6)

COMMON/dlim/ delamx, delcmx, deldmx, delemx, delpmx, delrmx, delvmx, delxmx, dtbkup, fpmax, fxmax, gxmax, svmaxt, svmxt3

COMMON/dlim/ nlim, nlim2

REAL VARIABLES:

delamx	Timestep limit caused by gas-volume-fraction change.
delcmx	Timestep limit caused by maximum changes in pressures and temperatures.
deldmx	Timestep limit caused by numerical considerations in the ROD and SLAB heat-transfer calculation.
delemx	Timestep limit caused by VESSEL mass errors.
delpmx	Timestep limit that results in a maximum 10% change in reactor-core power.
delrmx	Timestep limit caused by final value of the percentage variation in pressure from iteration to iteration.
delvmx	Material Courant stability limit (computed only in VESSELs).
delxmx	Timestep limit that results in the maximum allowed adjustment of VALVE components.
dtbkup	Timestep limit defined by `delpmx` or `delxmx` when a backup calculation is required after the prep-stage calculation.
fpmax	Maximum fractional change (0.1) in reactor-core power per timestep.
fxmax	VALVE-adjustment algorithm parameter (0.4).
gxmax	Minimum fractional change (0.05) in the VALVE maximum flow-area fraction change over a timestep.
svmaxt	Reciprocal of the material Courant timestep size in all 1D hydraulic components.
svmxt3	Reciprocal of the material Courant timestep size in all 3D VESSEL components.

INTEGER VARIABLES:

nlim	Array that stores the number of timesteps that were constrained by each of the timestep limits since the last small or large edit.
nlim2	Array that stores the number of timesteps that were constrained by each of the six different timestep limits defining `delcmx` since the last small or large edit [the sum of all six `nlim2(i)` equals `nlim`(5), which is the number of times `delcmx` controls the timestep size].

C.88. Include File (Common-Block) dmpck

COMMON/dmpck/ lvck

INTEGER VARIABLE:

lvck	Summed number of values over the VESSEL component that were written to the TRCDMP dump file (summed by subroutine DLEVEL but not used).

C.89. Include File (Common-Block) `dtinfo`

COMMON/dtinfo/ ddi, deltit, deltnc, delv3x, dprsv, dtddi, dtfl, dtlsv, dtrsv, dtssv, dtvsv, prmxsv, svdelc, tlmxsv, trmxsv, tsmxsv, tvmxsv

COMMON/dtinfo/ idcdal, idcdau, idcoal, idcoau, itrpdt, izdal, izdau, izoal, izoau, kccmxt

REAL VARIABLES:

ddi	Timestep-size, maximum-increase factor.
deltit	Timestep size from the iteration-count logic that results in scaling back the timestep size.
deltnc	Timestep size from the iteration-count logic that results in no change to the timestep size.
delv3x	Timestep size from the 3D material Courant limit in the VESSEL components.
dprsv	Pressure change `delcmx` limiter.
dtddi	Timestep size from the iteration-count logic that results in an increase in the timestep size by the factor `ddi`.
dtfl	Value of 0.5*`delt` used in the delcmx timestep-size control logic.
dtlsv	Liquid-temperature change `delcmx` limiter.
dtrsv	HTSTR-temperature change `delcmx` limiter.
dtssv	1-D component wall-temperature change `delcmx` limiter.
dtvsv	Gas-temperature change `delcmx` limiter.
prmxsv	Maximum pressure change used in the `delcmx` logic.
svdelc	Current value of `delcmx` (before 0.5*`delt` minimum applied).
tlmxsv	Maximum liquid-temperature change used in the `delcmx` logic.
trmxsv	Maximum HTSTR wall-temperature change used in the `delcmx` logic.
tsmxsv	Maximum 1D component wall-temperature change used in the `delcmx` logic.
tvmxsv	Maximum gas-temperature change used in the `delcmx` logic.

INTEGER VARIABLES:

idcdal	Cell number in the 3D level for gas-volume-fraction change timestep-size control variable `dal`.
idcdau	Cell number in the 3D level for gas-volume-fraction change timestep-size control variable `dau`.

idcoal	Cell number in the 3D level for gas-volume-fraction change timestep-size control variable `oal`.
idcoau	Cell number in the 3D level for gas-volume-fraction change timestep-size control variable `oau`.
itrpdt	Flag to indicate that trip-controlled, timestep-size logic was used in subroutine TRPSET.
izdal	3D level number for gas-volume-fraction change timestep-size control variable `dal`.
izdau	3D level number for gas-volume-fraction change timestep-size control variable `dau`.
izoal	3D level number for gas-volume-fraction change timestep-size control variable `oal`.
izoau	3D level number for gas-volume-fraction change timestep-size control variable `oau`.
kccmxt	Cell-face-index absolute value for 1D material Courant limit logic.

C.90. Include File (Common-Block) `elvkf`

COMMON/elvkf/ ielv, iinl, ikfac, mwfl, mwfv

INTEGER VARIABLES:

ielv	Option for inputting cell-centered elevations to the `grav` array (namelist variable).
iinl	Index for the two passes through INIT.
ikfac	Option for inputting K factors to the additive form-loss coefficient array (namelist variable).
mwfl	Option for inputting wall-to-liquid, wall-friction multiplier factors (namelist variable).
mwfv	Option for inputting wall-to-gas, wall-fraction multiplier factors (namelist variable).

C.91. Include File (Common-Block) `film`

COMMON/film/ conflm, filmu, filml, xfdcon, fdmax, alpf1, ffunh1, alpf2, ffunh2, xpfunh

REAL VARIABLES:

conflm	Constant used in film-thickness calculation.
filmu	Upper bound on film thickness.
filml	Lower bound on film thickness.
xfdcon	Multiplier on wet-wall film drag.

fdmax	Factor indicating how much larger film drag may be than wall drag.
alpf1	Gas-volume fraction when factor indicating cross-channel, cold-wall effect begins.
ffunh1	Factor indicating minimum cross-channel, cold-wall effect.
alpf2	Gas-volume fraction when factor indicating cross-channel, cold-wall effect reaches full on.
ffunh2	Factor indicating maximum cross-channe,l cold-wall effect.
xpfunh	Decay power for cross-channel, cold-wall effect.

C.92. Include File (Common-Block) `h2fdbk`

COMMON/h2fdbk/ ih2src

INTEGER VARIABLE:

ih2src	Hydrogen-source flag (namelist variable; when nonzero, TRAC sets namelist variables `igas` = 2 and `noair` = 0).

C.93. Include File (Common-Block) `htcav`

COMMON/htcav/ fhtcu, fhtcl, owhtd

REAL VARIABLES:

fhtcu	Maximum factor of increase (2.0) in the liquid and gas HTCs.
fhtcl	Minimum factor of decrease (0.0) in the liquid and gas HTCs.
owhtd	Fraction (0.55) of the previous time-averaged liquid or gas HTC that is averaged together with the fraction 1.0 – `owhtd` (0.45) of the present coefficient to define the present time-averaged value.

C.94. Include File (Common-Block) `htcref2`

COMMON/htcref2/ tvz(nzfmx), twz(nzfmx), znodes(nzfmx)

REAL VARIABLES:

tvz	Array of gas temperatures for a given ROD or SLAB.
twz	Array of wall temperatures for a given ROD or SLAB.
znodes	Array of node-center elevations.

INTEGER VARIABLES:

nzfmx

C.95. Include File (Common-Block) `htcref3`

COMMON/htcref3/ ifrez,nrefld(nxryt)

INTEGER VARIABLES:

ifrez	Flag used to turn interfacial vapor heat transfer off; i.e., freeze the drop size.
nrefld	Flag indicating the reflood model is on (set in subroutine CORE1).
nxryt	

C.96. Include File (Common-Block) `htcs`

COMMON/htcs/ hvap, hliq, slip, qsteam, htcwl, htcwv
COMMON/htcs/ iconht, mhtli, mhtlo, mhtvi, mhtvo

REAL VARIABLES:

hvap	Enthalpy of the gas.
hliq	Enthalpy of the liquid.
slip	Slip ratio between phasic velocities.
qsteam	Wall-to-wall heat flux.
htcwl	Constant wall-to-liquid HTC (namelist variable).
htcwv	Constant wall-to-gas HTC (namelist variable).

INTEGER VARIABLES:

iconht	Heat-transfer option (namelist variable). 0 = HTCs evaluated (default); 1 = constant HTCs defined by namelist variables.
mhtli	Option for inputting wall-to-liquid, heat-transfer multiplier factors for the inner surface of all HTSTRs (namelist variable).
mhtlo	Option for inputting wall-to-liquid, heat-transfer multiplier factors for the outer surface of all HTSTRs (namelist variable).
mhtvi	Option for inputting wall-to-gas, heat-transfer multiplier factors for the inner surface of all HTSTRs (namelist variable).
mhtvo	Option for inputting wall-to-gas, heat-transfer multiplier factors for the outer surface of all HTSTRs (namelist variable).

C.97. Include File (Common-Block) `ifcrs`

COMMON/ifcrs/ almax, almin, alpbcd, alpbch, alpbcw, alpdch, alpgs, alpmct, alpts1, alpts2, alptp, alptm, alpvs, alvcn, alvcn1, alvcn2, alvefx, alvev, alvev1, alvev2, alvfax, al01, aupct, aupdrg, ccfll, ccfvlm, ccful, chtabh, chtacc, chtafx, chtcn1, chtcn2, chtev1, chtev2, chtfax, chticn, chtiev, cndbs, cndfl, cndpl, cndro, cndst, chtinv, chtanv, d1x, d2x, dcalw1, dcalw2, dcdgm1, dcdgm2, dtvht, epmax, epmin, evfax,

f2mx, fcsub, fdis1, fdis2, fdisv1, fdisv2, fifam, fifbl, fifbs, fifcr, fifep, fifst, fifwl, fishi, flmin, flshf, flsh1, flsh2, fri1, fri2, fui1, fui2

COMMON/ifcrs/ fsb, ffs, fsm, frw, fmdis, fcdrop, fdis, ffd, voids1, voids2, voids3, voidd1, voidd2, xmdis, xnb, xhvdis, harmx, hamin, hcmin, hcamin, hdmax, hfvl, hfvu, himfac, h0, pc24, pcrit, regmn, rdmax, rdmin, slp1, stfrl, stfru, ststrt, tlgts, tvltl, tvlts, twdfac, twdfak, vlacc, vlmax, vrcmin, vrfmin, vr2min

COMMON/ifcrs/ iepri, iwils, ihotp, lpdrgx

REAL VARIABLES:

almax	Maximum gas-volume fraction (0.9999) to use in calculation of interfacial drag.
almin	Minimum gas-volume fraction (0.00001) to use in calculation of interfacial drag.
alpbcd	Minimum gas-volume fraction (0.00001) to use in calculation of bubbly interfacial drag.
alpbch	Minimum gas-volume fraction (0.00001) to use in calculation of bubbly interfacial heat transfer.
alpbcw	Minimum gas-volume fraction (0.00001) to use in the Wilson model (upper plenum) calculation of interfacial drag.
alpdch	Maximum gas-volume fraction (0.9995) for calculation of droplet diameter in the annular-mist regime.
alpgs	Variable not used.
alpmct	Variable not used.
alpts1	Variable not used.
alpts2	Variable not used.
alptp	Minimum gas-volume fraction (0.9) in cell above for vertical stratified flow in the VESSEL.
alptm	Maximum gas-volume fraction (0.1) in cell below for vertical stratified flow in the VESSEL.
alpvs	Gas-volume fraction constant (0.3) in model for bubbly flow below a stratified level.
alvcn	Time constant in rate model for change in condensing ALVE.
alvcn1	Constant used in determining the upper bound at the liquid-side HTC for subcooled liquids.
alvcn2	Constant used in determining the lower bound of the liquid-side HTC for subcooled liquids.
alvefx	Constant used in determining the limit of the liquid-side HTC between timesteps.
alvev	Constant used in determining the limit of the liquid-side HTC between timesteps.

alvev1	Constant used in determining the upper bound of the liquid-side HTC for saturated or superheated liquids.
alvev2	Constant used in determining the lower bound of the liquid-side HTC for saturated or superheated liquids.
alvfax	Constant used in determining the limit of flashing HTC between timesteps.
al01	Constant in subcooled boiling model.
aupct	Constant in stratified flow model for ACCUMs (variable not used).
aupdrg	Constant in stratified flow model for ACCUMs (variable not used).
ccfll	Variable not used.
ccfvlm	Variable not used.
ccful	Variable not used.
chtabh	Gas-side HTC for noncondensable gas in bubbly slug flow.
chtacc	Gas-side HTC for ACCUMs (variable not used).
chtafx	Constant used in determining the limit of the gas-side HTC for noncondensable gas.
chtcn1	Constant used in determining the upper bound of the gas-side HTC for subcooled vapor.
chtcn2	Constant used in determining the lower bound of the gas-side HTC for subcooled vapor.
chtev1	Constant used in determining the upper bound of the gas-side HTC for saturated or superheated vapor.
chtev2	Constant used in determining the lower bound of the gas-side HTC for saturated or superheated vapor.
chtfax	Constant used in determining the limit of the gas-side HTC between timesteps.
chticn	Constant used in determining the limits of the gas-side HTC.
chtiev	Constant used in determining the limits of the gas-side HTC.
cndbs	Constant to adjust the interfacial area for condensing bubble.
cndfL	Constant to adjust the liquid-side HTC in annular-mist flow.
cndpl	Constant to adjust the liquid-side HTC for condensation in plug flows.
cndro	Constant to adjust the liquid-side HTC in annular-mist flows.
cndst	Multiplier for stratified-flow condensation interfacial heat transfer.
chtinv	Liquid-side HTC for smooth, rough-wavy, and agitated inverted-annular flows.

chtanv	Liquid-side HTC of noncondensable gas for smooth, rough-wavy, and agitated inverted-annular flow.
d1x	Constant in EPRI model.
d2x	Constant in EPRI model.
dcalw1	Minimum gas-volume fraction (0.0001) for averaging the bubbly flow-regime axial drag and separated-flow-regime Blausis axial drag in the downcomer and lower plenum.
dcalw2	Maximum gas-volume fraction (0.05) for averaging the bubbly flow-regime axial drag and separated-flow-regime Blausis axial drag in the downcomer and lower plenum.
dcdgm1	Multiplier (0.5) for Blausis axial drag in the downcomer and lower plenum for the bubbly flow regime.
dcdgm2	Multiplier (0.5) for Blausis axial drag in the downcomer and lower plenum for the separated-flow regime.
dtvht	Variable not used.
epmax	Maximum drag on EPRI model for CORE-component interfacial drag (variable not used).
epmin	Minimum drag on EPRI model for CORE-component interfacial drag (variable not used).
evfax	Constant in the evaporation model.
f2mx	Factor in the droplet vapor to interface heat-transfer model.
fcsub	Multiplication constant in the subcooled-boiling condensation model.
fdis1	Constant in the dispersed-droplet, interfacial-drag model.
fdis2	Constant in the dispersed-droplet, interfacial-drag model.
fdisv1	Constant in the dispersed-droplet, interfacial-drag model.
fdisv2	Constant in the dispersed-droplet, interfacial-drag model.
fifam	Factor (1.0) applied to annular-mist interfacial drag.
fifbl	Multiplier for downcomer interfacial-drag model.
fifbs	Factor (1.0) applied to bubbly slug interfacial drag.
fifcr	Variable not used.
fifep	Multiplier (1.0) in EPRI interfacial-drag model (variable not used).
fifst	Factor (1.0) applied to stratified-flow interfacial drag.
fifwl	Multiplier (1.0) for Wilson-model interfacial drag in the upper plenum of the VESSEL.
fishi	Variable not used.
flmin	Minimum film thickness for annular flow.
flshf	Multiplier on liquid superheat for flashing.
flsh1	Maximum-flash multiplier.

flsh2	Minimum-flash multiplier.
fri1	Time constant for rate of decrease in C_i.
fri2	Time constant for rate of increase in C_i.
fui1	Minimum allowed change in C_i.
fui2	Maximum allowed change in C_i.
fsb	Constant to adjust the interfacial-drag coefficient for the subcooled nucleate-boiling regime.
ffs	Constant to adjust the interfacial-drag coefficient for the free-stream contribution in subcooled nucleate-boiling regime.
fsm	Constant to adjust the interfacial-drag coefficient for the smooth inverted-annular flow.
frw	Constant to adjust the interfacial-drag coefficient for the rough-wavy, inverted-annular flow.
fmdis	Constant to adjust the interfacial-drag coefficient for the post-agitated, inverted-annular flow.
fcdrop	Constant to adjust the droplet interfacial-drag coefficient for lightly dispersed inverted-annular flow.
fdis	Constant to adjust the interfacial-drag coefficient for highly dispersed inverted-annular flow.
ffd	Constant to adjust the interfacial-drag coefficient for liquid flow in highly dispersed inverted-annular flow.
voids1	Lower limit of the gas-volume fraction (0.05) for smooth inverted-annular flow.
voids2	Upper limit on the gas-volume fraction for rough-wavy, inverted-annular flow.
voids3	Upper limit on the gas-volume fraction (0.3) for smooth inverted-annular flow.
voidd1	Variable not used.
voidd2	Gas-volume fraction limit in the rough-wavy reflood flow regime.
xmdis	Constant to adjust the weighting for the post-agitated, inverted-annular-flow, interfacial-drag coefficient.
xnb	Constant to adjust the weighting for the bubbly flow interfacial-drag coefficient in the intermediate gas-volume-fraction region.
xhvdis	Constant to adjust the weighting for rough-wavy- and smooth-inverted-annular-flow, interfacial-drag coefficient in the intermediate gas-volume-fraction region.
harmx	Variable not used.
hamin	Product of the HTC times the interfacial area.
hcmin	Variable not used.

hcamin	Variable not used.
hdmax	Hydraulic diameter used in the VESSEL component if user-input hydraulic diameter is less than 10^{-5}.
hfvl	Constant used in dispersed-droplet, interfacial-drag model.
hfvu	Constant used in dispersed-droplet, interfacial-drag model.
himfac	Multiplication constant used in the calculation of the minimum gas-interface, heat-transfer rate.
h0	Constant in subcooled-boiling condensation model.
pc24	Pressure constant (1.95187E+15 Pa^2).
pcrit	Critical-point pressure (2.209E+07 Pa).
regmn	Minimum Reynold's number in stratified flow.
rdmax	Maximum droplet radius in annular-mist flow.
rdmin	Minimum droplet radius in annular-mist flow.
slp1	Constant in subcooled-boiling model.
stfrl	Stratified-flow, lower-velocity limit multiplier.
stfru	Stratified-flow, upper-velocity limit multiplier.
ststrt	Multiplier (1.0) on stratified-flow interfacial heat transfer.
tlgts	Maximum liquid superheat used to calculate limit on interfacial heat transfer.
tvltl	Maximum gas temperature less than liquid temperature used to calculate limit on interfacial heat transfer.
tvlts	Maximum vapor subcooling used to calculate limit on interfacial heat transfer.
twdfac	Constant in subcooled-boiling model.
twdfak	Constant in subcooled-boiling model.
vlacc	Maximum liquid velocity in ACCUM for calculation of interfacial heat transfer (variable not used).
vlmax	Maximum liquid velocity in annular film for calculation of interfacial heat transfer.
vrcmin	Minimum relative velocity (0.1) used to calculate a run.
vrfmin	Minimum relative velocity to be used in the bubbly slug C_i calculation.
vr2min	Variable not used.

INTEGER VARIABLES:

iepri	EPRI interfacial-drag-model flag used for rod bundles in the core region when set to 1.
iwils	Wilson interfacial-drag-model flag for use in the upper plenum when set to 1.
ihotp	Hot-patch modeling option.

0 = off;

1 = on.

lpdrgx — Lower-plenum Blasius interfacial-drag off flag.

0 = apply Blasius interfacial drag in the lower plenum when IBLAUS = 1.

1 = do not apply Blasius interfacial drag in the lower plenum when IBLAUS = 1.

C.98. Include File (Common-Block) `infohl`

COMMON/infohl/ dropd, fhlf, qden, qfr, qtotal, qwebb, vr2

REAL VARIABLES:

dropd	Calculated drop diameter used in the Forsland-Rohsenow correlation.
fhlf	Factor carried along to separate the Denham and Forsland-Rohsenow regions.
qden	Heat flux calculated using the Denham correlation.
qfr	Heat flux calculated using the Forsland-Rohsenow correlation.
qtotal	Total heat flux calculated, including radiation.
qwebb	Heat flux calculated using the Webb-Chen correlation.
vr2	Local relative velocity minus quench-front relative velocity.

C.99. Include File (Common-Block) `junction`

COMMON/junct/ jptr, jmatch

INTEGER VARIABLES:

jptr	Number of junction-component pairs.
jmatch	Number of bad junction numbers detected during the network trace in SRTLP.

C.100. Include File (Common-Block) `massck`

COMMON/massck/ nstabo

INTEGER VARIABLE:

nstabo	Old value of `nstab` (flag for evaluating the SETS3D equations) from the previous timestep.

C.101. Include File (Common-Block) `nrcmp`

COMMON/nrcmp/ ncmpmx, nhtsmn, nrcomp

INTEGER VARIABLES:

ncmpmx	Maximum hydraulic-component number.
nhtsmn	Minimum heat-structure component number.
nrcomp	Number of components defined from the TRCRST restart-data file.

C.102. Include File (Common-Block) `pmpstb`

COMMON/pmpstb/ ipmpcn
COMMON/pmpstb/ fwpa

INTEGER VARIABLE:

ipmpcn	Flag for not defining the donor-celled mixture density and gas-volume fraction across the pump-impeller interface.

REAL VARIABLE:

fwpa	Fraction 0.1 of the present donor-celled, gas-volume fraction across the pump-impeller interface that is averaged with the fraction (1.0 – `fwpa` = 0.9) of its previous gas-volume fraction average to define the gas-volume fraction for evaluating the PUMP-curve HDM table.

C.103. Include File (Common-Block) `refhti`

COMMON/refhti/ agalp, agsz, chfalp, chfhv, chfz, dfalp, dfsz, rwalp, rwsz, smalp, smsz, tbalp, tbz, unhf, cafj, vlag, vvag

REAL VARIABLES:

agalp	Gas-volume fraction at the agitated section of inverted-annular flow.
agsz	Elevation of the agitated section of inverted-annular flow.
chfalp	Gas-volume fraction at the CHF point.
chfhv	Vapor heat transfer at CHF.
chfz	Elevation of CHF.
dfalp	Gas-volume fraction at the highly dispersed section elevation.
dfsz	Elevation of highly dispersed section of inverted-annular flow.
rwalp	Gas-volume fraction of rough-wavy section elevation.
rwsz	Elevation of rough-wavy section of inverted-annular flow.
smalp	Gas-volume fraction at the smooth section elevation.
smsz	Elevation of smooth section of inverted-annular flow.
tbalp	Gas-volume fraction at transition-boiling point.
tbz	Elevation of transition boiling.

unhf	Fraction of heated surface that is unheated.
cafj	Capillary number.
vlag	Liquid velocity at the agatiated level.
vvag	Gas velocity at the agatiated level.

C.104. Include File (Common-Block) `refhti2`

COMMON/refhti2/ alptsl, alptrl, alptal, alptsu, alptru, alptau

REAL VARIABLES:

alptsl	Minimum gas-volume fraction allowed for the end of the smooth-inverted flow regime.
alptrl	Minimum gas-volume fraction allowed for the end of the rough-wavy-inverted flow regime.
alptal	Minimum gas-volume fraction allowed for the end of the agitated-inverted flow regime.
alptsu	Maximum gas-volume fraction allowed for the end of the smooth-inverted flow regime.
alptru	Maximum gas-volume fraction allowed for the end of the rough-wavy-inverted flow regime.
alptau	Maximum gas-volume fraction allowed for the end of the agitated-inverted flow regime.

C.105. Include File (Common-Block) `rows`

COMMON/rows/ iscl

INTEGER VARIABLE:

iscl	Flag that has TRAC-P divide by the largest matrix element in each matrix row for all four or five matrix elements and three right-hand-side elements in each row of the 4 x 4 or 5 x 5 outer-iteration, mesh-cell matrix equation. 0 = no; 1 = yes.

C.106. Include File (Common-Block) `sepcb`

COMMON /sepcb/ alpspc,alpdrc,dpsepc,isepcb,idrycb,ncsepc,ndryrc, nsepsc,istagc

REAL VARIABLES:

alpspc	Separator gas-volume fraction.
alpdrc	Gas-volume fraction to be convected from the dryer.
dpsepc	Separator pressure drop.

INTEGER VARIABLES:

isepcb	Separator flag.
idrycb	Dryer flag.
ncsepc	Cell number for separator.
ndryrc	Cell number for dryer.
nsepsc	Number of separators modeled.
istagc	Separator-option type.

C.107. Include File (Common-Block) `solcon`

COMMON/solcon/ cnt, cnc, cntlmn, cnmin, cntlmx, cnmax

REAL VARIABLES:

cnt	Coefficient of liquid temperature (kg solute/kg liquid K, lb_m solute/lb_m liquid F) in linear fit to solubility.
cnc	Constant term (kg solute/kg liquid, lb_m solute/lb_m liquid) in linear fit to solubility.
cntlmn	Minimum liquid temperature (K, °F) of linear fit.
cnmin	Solubility (kg solute/kg liquid, lb_m solute/lb_m liquid) when the liquid temperature is at or below cntlmn.
cntlmx	Maximum liquid temperature (K, °F) of linear fit.
cnmax	Solubility (kg solute/kg liquid, lb_m solute/lb_m liquid) when the liquid temperature is at or above cntlmx.

C.108. Include File (Common-Block) `stncom`

COMMON/stncom/ stnmax, tmstnu, tldmin, tmtld

COMMON/stncom/ istnu, jstnu, kstnu, nstnu

COMMON/stncom/ itldm, jtldm, ktldm, ntldm

REAL VARIABLES:

stnmax	Largest Stanton number evaluated in this calculation.
tmstnu	Time when `stnmax` was evaluated.
tldmin	The minimum liquid temperature (for any heat structure) when subcooled boiling begins based on the Saha-Zuber correlation.
tmtld	Time when `tldmin` was found.

INTEGER VARIABLES:

istnu	3D r- or x-cell number where `stnmax` was evaluated.
jstnu	3D θ- or y-cell number where `stnmax` was evaluated.
kstnu	3D z-level number where `stnmax` was evaluated.
nstnu	Component number where `stnmax` was evaluated.

itldm	Variable not used.
jtldm	Axial node number where `tldmin` was found.
ktldm	Variable not used.
ntldm	Component number where `tldmin` was found.

C.109. Include File (Common-Block) `strtnt`

COMMON/strtnt/ sdtint, stfvl, stfvu, stfll, stflu, fstrv, fstrl

REAL VARIABLES:

sdtint	Variable not used.
stfvl	Variable not used.
stfvu	Variable not used.
stfll	Constant used to determine stratified-flow weighting factors.
stflu	Constant used to determine stratified-flow weighting factors.
fstrv	Variable not used.
fstrl	Multiplier on the liquid velocity check for stratified flow in subroutine `cella3`.

C.110. Include File (Common-Block) `supres`

COMMON/supres/ s

REAL VARIABLE:

s	Factor in nucleate-boiling HTC evaluation in subroutine chen.

C.111. Include File (Common-Block) `syssum`

COMMON/syssum/ alqcor, alqprz, alqup, corwm, pmx, tlmx, tlncor, tshcor, tsncor, tvmx, volcor, xlqcor, xtshcr

COMMON/syssum/ jpmx, jtlmx, jtvmx, npmx, ntlmx, ntvmx

REAL VARIABLES:

alqcor	Core-region mean liquid-volume fraction.
alqprz	PRIZER (pressurizer) mean liquid-volume fraction.
alqup	Upper-plenum mean liquid-volume fraction (evaluated only for 3D VESSELs).
corwm	Core-region water mass.
pmx	Maximum pressure.
tlmx	Maximum liquid temperature.
tlncor	Core-region mean liquid temperature.
tshcor	Core region mean superheat.

tsncor	Core-region mean saturation temperature.
tvmx	Maximum gas temperature.
volcor	Core-region volume.
xlqcor	Minimum core-region, liquid-volume fraction.
xtshcr	Maximum core-region superheat.

INTEGER VARIABLES:

jpmx	Cell number for the maximum pressure.
jtlmx	Cell number for the maximum liquid temperature.
jtvmx	Cell number for the maximum gas temperature.
npmx	Component number for the maximum pressure.
ntlmx	Component number for the maximum liquid temperature.
ntvmx	Component number for the maximum gas temperature.

C.112. Include File (Common-Block) `totals`

COMMON/totals/ tlen, tvol

REAL VARIABLES:

tlen	Total length of a component.
tvol	Total fluid volume of a component.

C.113. Include File (Common-Block) `tst3d`

COMMON/tst3d/ ccif,i1d, nifht, nifsh, noboil, noimp, nwsh, imoml

REAL VARIABLE:

ccif	Constant value for the interfacial-drag coefficient when `nifsh` = 1 (namelist variable).

INTEGER VARIABLES:

i1d	Flag to convert mean-mass and gas-mass equations to gas-mass and liquid-mass equations for evaluation by subroutine `tf3ds`.
nifht	Flag for defining a constant 10.0 value to the `alve`, chti, `alv`, and `chtia` evaporation and condensation coefficients.
nifsh	Interfacial-shear (drag) option flag (namelist variable).
noboil	Flag for not evaluating evaporation and condensation when IEOS = 0.
noimp	Flag for not evaluating the gas-volume-fraction, time-derivative term in the motion equation. 0 = evaluate the gas-volume-fraction, time-derivative term; 1 = do not evaluate the gas-volume-fraction, time-derivative term (default).

nwsh — Flag for defining the gas fric by its gas-field value rather than by the liquid-field value.

imoml — Option to improve momentum conservation where the gas-volume-fraction gradient is large.

0 = no (default);

1 = yes.

C.114. Include File (Common-Block) vckdat

COMMON/vckdat/ dontol

COMMON/vckdat/ iprvck, iskip, itvkmx

REAL VARIABLE:

dontol — Tolerance for density difference requiring redonor celling in the VESSEL.

INTEGER VARIABLES:

iprvck — Flag to print information about redonor celling in the VESSEL (normally set to 0 for no print).

iskip — Flag to skip redonor-cell logic in the VESSEL component (normally set to 0 for no skip).

itvkmx — Maximum iteration count to check for need to redonor cell in the VESSEL.

C.115. Include File (Common-Block) vdvmod

COMMON/vdvmod/ ivdvs1, ivdvs2

INTEGER VARIABLES:

ivdvs1 — Flag for scaling the $V \cdot \nabla V$ terms.

0 = no;

1 = yes.

ivdvs2 — Flag for scaling the $\beta V \cdot \nabla V$ terms.

0 = no;

1 = yes.

C.116. Include File (Common-Block) vellim

COMMON/vellim/ vvub, vvlb, vlub, vllb, dfvub, dfvlb, dflub, dfllb, jvlim

REAL VARIABLES:

vvub — Pump-impeller interface, gas-velocity, upper-limit value.

vvlb — Pump-impeller interface, gas-velocity, lower-limit value.

vlub	Pump-impeller interface, liquid-velocity, upper-limit value.
vllb	Pump-impeller interface, liquid-velocity, lower-limit value.
dfvub	Derivative of the pump-impeller interface gas velocity (at its upper limit) with respect to total pressure.
dfvlb	Derivative of the pump-impeller interface gas velocity (at its lower limit) with respect to total pressure.
dflub	Derivative of the pump-impeller interface liquid velocity (at its upper limit) with respect to total pressure.
dfllb	Derivative of the pump-impeller interface liquid velocity (at its lower limit) with respect to total pressure.

INTEGER VARIABLE:

jvlim:	For PUMP type `ipmpty` = 0, the pump-impeller interface number (`jvlim` = 2) when the PUMP component-action table defines the fluid velocity.

C.117. Include File (Common-Block) **webnum**

COMMON/webnum/ alvfcp, alvfcs, bmin, chtfcp, chtfcs, chtibc, chtibh, cndfc, dmin, pentl, pentu, vlspr, vvlow, vvup, web, wed, wedu

COMMON/webnum/ ichvol

REAL VARIABLES:

alvfcp	Multiplier on `alv` for low-velocity vertical components.
alvfcs	Multiplier on `alv` under spray conditions.
bmin	Minimum allowed bubble size.
chtfcp	Multiplier on `chti` for low-velocity vertical components.
chtfcs	Multiplier on `chti` under spray conditions.
chtibc	Vapor-bubble interfacial HTC when `tv` > `tsat`.
chtibh	Vapor-bubble interfacial HTC when `tv` < `tsat`.
cndfc	Condensation-rate scaling factor.
dmin	Minimum allowed drop size.
pentl	Lower bound on entrained gas-volume fraction α.
pentu	Upper bound on entrained gas-volume fraction α.
vlspr	Lower limit on the quantity $(1-\alpha)V_\ell$ at the top of the cell above which the spray condition is assumed to exist.
vvlow	Lower limit on gas velocity for special condensation model for low-velocity vertical components.
vvup	Upper limit on gas velocity for special condensation model for low-velocity vertical components. Note: For liquid velocity greater than `vlup`, the regular condensation model is used. For liquid velocity less than `vllow`, the special condensation model

is used. For liquid velocity between `vllow` and `vlup`, a linear interpolation between the two models is used.

web — Bubble Weber number.

wed — Droplet Weber number.

wedu — Droplet Weber number during core-region upflow (not implemented).

INTEGER VARIABLE:

ichvol — Flag that invokes a minimum value on the interfacial HTC.

0 = no effect (default);

1 = sets the minimum value to the cell volume times 1.0×10^7.