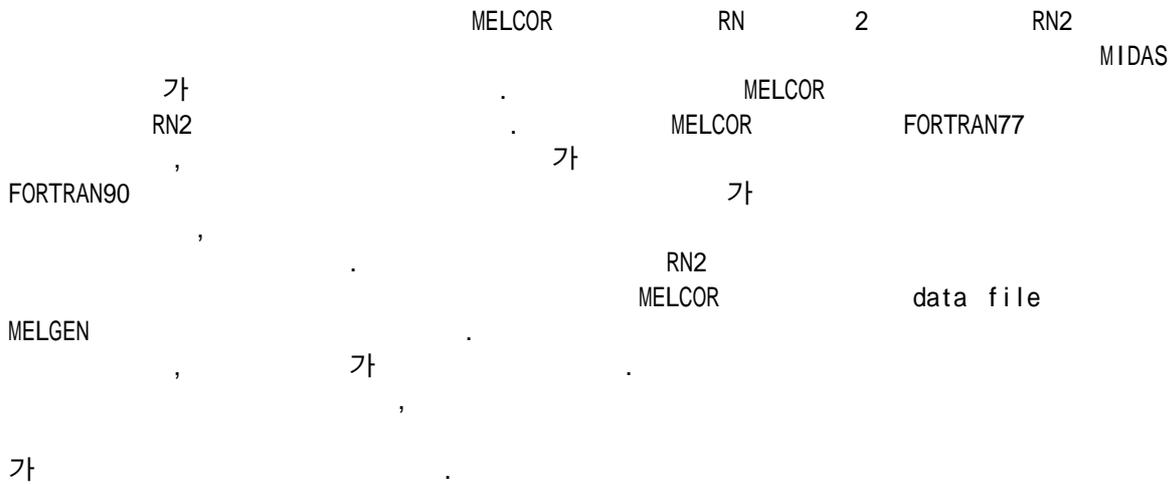


(MIDAS) RN2

A Restructuring of RN2 Package for MIDAS Computer Code

150



Abstract

RN2 package, which is one of two fission product-related package in MELCOR, has been restructured for the MIDAS computer code. MIDAS is being developed as an integrated severe accident analysis code with a user-friendly graphical user interface and data structure. To do this, data transferring methods of current MELCOR code are modified and adopted into the RN2 package. The data structure of the current MELCOR code using FORTRAN77 causes a difficult grasping of meaning of the variables as well as waste of memory. New features of FORTRAN90 make it possible to allocate the storage dynamically and to use the user-defined data type, which lead to an efficient memory treatment and an easy understanding of the code. Restructuring of the RN2 package addressed in this paper includes module development, subroutine modification, and treats MELGEN, which generates data file, as well as MELCOR, which is processing a calculation. The validation has been done by comparing the results of the modified code with those from the existing code. As the trends are the similar to each other, it hints that the same approach could be extended to the entire code package. It is expected that code restructuring will accelerate the code domestication thanks to direct understanding of each variable and easy implementation of modified or newly developed models.

1.

MELCOR MIDAS MELCOR 가
 MELCOR 가
 , 가
 , 가
 FORTRAN90 가
 가 data type .[1,2,3]
 , (readability) (DMM)
 , 가 subroutine (derived type variables)
 [4,5].
 [6,7,8],
 2) RN2 (RN
 restart file MELGEN , MELCOR ,

2.

MELCOR code 3 restart file
 MELGEN, restart file , log file plot file
 MELCOR, PLOT
 가 , data COR, HT, SPR, TF, RN1, RN2 20
 subroutine , message subroutine, code
 subroutine subroutine [9].

2.1 Restart file

data MELCOR
 RESTART file read write subroutine MXXRS MXXRSW ,
 read write , subroutine
 xyzPRS 1, 2 , subroutine xyzRIO
 coefficient , real, integer, logical,
 character 4 array . (xyz :
)

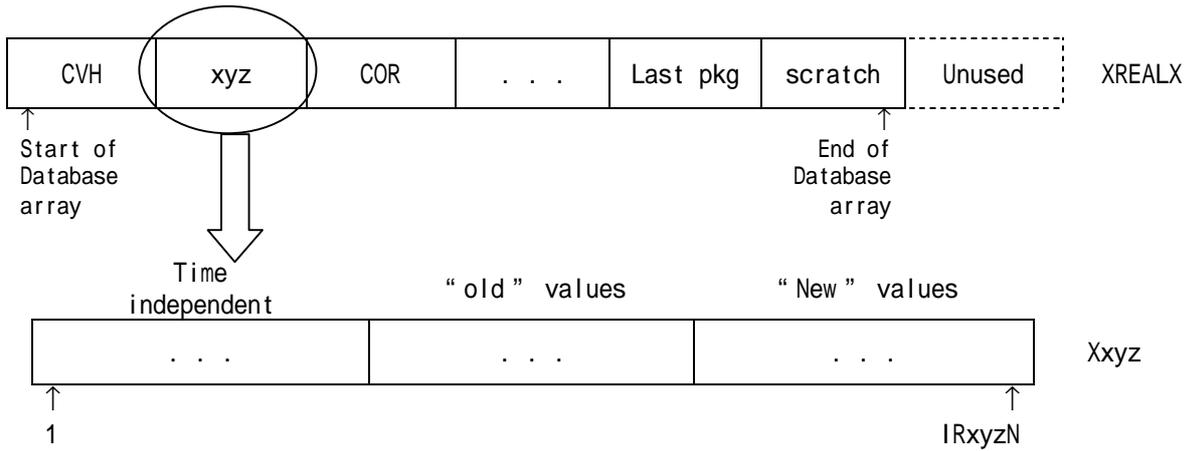
2.2 Database

MELCOR data
 4가

- XREALX : floating point variables (500,000 array)

- INTEGE : integer variables (50,000 array)
- LOGICA : logical variables (5,000 array)
- CHARAC : character variables (30,000 array)

(1),



1.

Database array subroutine 274 2 . Database array argument 1 ,

```

SUBROUTINE xyzDBC
*- INCLUDE BLANK
COMMON /DBREAL/ NEDREL, IRELCS, IRELES
COMMON /      / DREALX(NUMREL/2)
DOUBLE PRECISION DREALX C
DIMENSION XREALX(NUMREL)
EQUIVALENCE (XREALX(1), DREALX(1))
DOUBLE PRECISION VREALX(NUMREL/2)
EQUIVALENCE (VREALX(1), DREALX(1))
*- INCLUDE xyzDB
COMMON /xyzDB/ IRxyzF , IRxyzN , IxyzF , IxyzN ,
1 ILxyzF , ILxyzN , ICxyzF , ICxyzN
CALL xyzDBD(
1 IRxyzN , XREALX(IRxyzF) ,
2 IxyzN , INTEGE(IxyzF) ,
3 ILxyzN , LOGICA(ILxyzF) ,
4 ICxyzN , CHARAC(ICxyzF) , . . . )
C
RETURN
END

```

2. 1 database subroutine

Database 2
array argument , subroutine 3 .

```

SUBROUTINE xyzDBD (
1 NxyzR , Rxyz ,
2 NxyzI , Ixyz ,
3 NxyzL, Lxyz,
4 NxyzC, Cxyz, . . . )
C
  DIMENSION Rxyz(NxyzR), Ixyz(NxyzI), . . .
C
*- INCLUDE xyzPNT
COMMON /xyzPNT/ Nvar1, Nvar2, . . .
C
CALL xyzRUN (Nvar1, Rxyz(Ivar1), Nvar2,Ixyz(Ivar2), . . . )
C
RETURN
END

```

3. 2 database subroutine

Database argument ,
subroutine 4 .

```

SUBROUTINE xyzRUN(
1 Nvar1, var1, Nvar2, var2, . . . )
C
  DIMENSION var1(Nvar1), . . .
C
FROM HERE ON, SIMPLY USE var1 AND var2.
FORGETTING ALL DETAILS OF WHERE AND HW THEY ARE STORED.
C
CALL xyz. . .
C
RETURN
END

```

4. database subroutine

2.3

subroutine xyzDBD subroutines database comment argument pass subroutine xyzDBC . RN2

(1) RN2 1
1 (5),
subroutine ' RN2DB ' common block (6), 4 data
type 2

(2) RN2 2
2 , subroutine
' xyzPNT ' common block , 4 data type
database 가 RN2

7 .

, restart file

MELCOR

```

* - INCLUDE RN2PNT
C
C   RN2 DATABASE POINTERS
C
COMMON/RN2PNT/ IT2ARG, IT2ARL, IT2AFG, IT2AFL,
+             IT2VPG, IT2VPL, IT2VFG, IT2VFL,
+             IR2ARG, IR2ARL, IR2AFG, IR2AFL,
+             IR2VPG, IR2VPL, IR2VFG, IR2VFL,
+             IORRN2, IOIRN2
* -
* - INCLUDE RN2PLS
C
C   RN2 DATABASE FOR POOL SCRUBBING AEROSOL REMOVAL
C
COMMON/RN2PLS/ IAVENT, IMVENT, INVENT, INTYPE, INUMFP, NUMPLS, NPLS
+             , INUMDF, IDFBUB, IMCAPO, IMESCO, IMCAPN, IMESCN, IOBUB
* -
* - INCLUDE RN2FLT
C
C   RN2 DATABASE FOR FILTER AEROSOL AND FP VAPOR REMOVAL
C
COMMON/RN2FLT/ INMFLT, IDF, IFLTFP, IFLTCV, IDFTFA, IDFTFV,
+             IMXFCA, IMXFCA, IMXFCA, IMXFCA,
+             IAEFTO, IAEFTN, IRAFTO, IRAFTN, IVPFTO, IVPFTN, IRVFTO, IRVFTN, IOFLT,
+             NUMFLT, NFLT, ISRFLO, ISRFLN, IOSRFL
* -
* - INCLUDE RN2SP
C
C   RN2 DATABASE FOR SPRAY AEROSOL AND FP VAPOR REMOVAL
C
COMMON/RN2SP/ IICLS, IHSPR, ILMDAV, ILMDA
* -

```

7. RN2 2

MELCOR 1.8.4

FORTRAN90

가

MELCOR

FORTRAN90

(derived data type)

8

RN 2

RN2 , RN2

. MELCOR

subroutine RN2

60

subroutine

subroutine

[4,5].

3.1


```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!! RN2 (Radio Nuclide 2) package data !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! IMPLICIT NONE
! MODULE RN2_MDL
!
! ***** RadioNuclide-2 GLOBAL DATA *****
INTEGER :: NUMPLS, NPLS, NUMFLT, NFLT, IICLS
INTEGER :: NRRN2S, NIRN2S, NRN2_FLAG
INTEGER :: RN2_NNVOL, RN2_NNFL, RN2_NSEC, RN2_NCLS
INTEGER :: RN2_NDPSR, RN2_MXSPSZ, RN2_MXSPJN
!
! ***** RN2_NC(NFLT) *****
TYPE RN2_NO ; SEQUENCE
REAL :: DFCTFA, DFCTFV, XMXFCA, XMXFCV, &
VAPFTO, RDVFTO, AERFTO, RDAFTO, &
VAPFTN, RDVFTN, AERFTN, RDAFTN
END TYPE
TYPE (RN2_NO), ALLOCATABLE :: RN2_NNC(:, :)
!
! TYPE RN2_N1 ; SEQUENCE
REAL :: SRFLT0, SRFLT1
END TYPE
TYPE (RN2_N1), ALLOCATABLE :: RN2_NNS(:, :)
!
! . . . . .
!
! TYPE RN2_ND ; SEQUENCE
REAL :: XLMDAA
END TYPE
TYPE (RN2_ND), ALLOCATABLE :: RN2_NSSC(:, :, :, :)
!
! END MODULE RN2_MDL
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

9. RN2

1. Array

DFCTFA(NFLT, NUMCLS)	RN2_NNC(NFLT, NUMCLS)%DFCTFA
DFCTFV(NFLT, NUMCLS)	RN2_NNC(NFLT, NUMCLS)%DFCTFV
XXMFCA(NFLT, NUMCLS)	RN2_NNC(NFLT, NUMCLS)%XXMFCA
XXMFCV(NFLT, NUMCLS)	RN2_NNC(NFLT, NUMCLS)%XXMFCV
AER2L(NUMCLS, NNVOL)	RN2_NCNV(NUMCLS, NNVOL)%AER2L
VAP2G(NUMCLS, NNVOL)	RN2_NCNV(NUMCLS, NNVOL)%VAP2G
VAP2L(NUMCLS, NNVOL)	RN2_NCNV(NUMCLS, NNVOL)%VAP2L
RDA2L(NUMCLS, NNVOL)	RN2_NCNV(NUMCLS, NNVOL)%RDA2L
RDV2G(NUMCLS, NNVOL)	RN2_NCNV(NUMCLS, NNVOL)%RDV2G
RDV2L(NUMCLS, NNVOL)	RN2_NCNV(NUMCLS, NNVOL)%RDV2L
AVENT(NPLS)	RN2_NP(NPLS)%AVENT
MVENT(NPLS)	RN2_NP(NPLS)%MVENT
NVENT(NPLS)	RN2_NP(NPLS)%NVENT
NTYPE(NPLS)	RN2_NP(NPLS)%NTYPE
I NUMFP(NPLS)	RN2_NP(NPLS)%I NUMFP
I NUMDF(NPLS)	RN2_NP(NPLS)%I NUMDF

subroutine

<> subroutine RN2DBD

```
SUBROUTINE RN2DBD (NRRN2, XRN2, NIRN2, IRN2, NRRN1, XRN1, NIRN1, &
  IRN1, NRCVH, XCV, NICVH, ICV, NRFL, XFL, NIFL, IFL, NICAV, ICAV, &
  NRSPPR, XSPR, NISPR, ISPR, NLSPPR, LSPR, NCSPPR, CSPR, NRSC, XSC, &
  NISC, ISC)
. . . . .
!      COPY RN1 MASSES INTO RN2 ARRAYS
!
CALL RN2RNO (NNVOL, NUMSEC, NUMCLS, XRN2 (IT2ARG), XRN2 (IT2ARL), &
  XRN2 (IT2VPG), XRN2 (IT2VPL), XRN2 (IR2ARG), XRN2 (IR2ARL), &
  XRN2 (IR2VPG), XRN2 (IR2VPL), XRN1 (IT1AGN - J1), XRN1 (IT1ALN - &
  J1), XRN1 (IT1VGN - J1), XRN1 (IT1VLN - J1), XRN1 (IR1AGN - J1), &
  XRN1 (IR1ALN - J1), XRN1 (IR1VGN - J1), XRN1 (IR1VLN - J1), &
  XCV (KCVVON - IC), ICV (KCVNUM) )
. . . . .
RETURN
END SUBROUTINE RN2DBD
```

<> subroutine RN2DBD

```
!      SUBROUTINE RN2DBD (NRRN2, XRN2, NIRN2, IRN2, NRRN1, XRN1, NIRN1, &
!      IRN1, NRCVH, XCV, NICVH, ICV, NRFL, XFL, NIFL, IFL, NICAV, ICAV, &
!      NRSPPR, XSPR, NISPR, ISPR, NLSPPR, LSPR, NCSPPR, CSPR, NRSC, XSC, &
!      NISC, ISC)
!! NOTICE: Converted by MELtoMID
SUBROUTINE RN2DBD (      NRRN1, XRN1, NIRN1, &
  IRN1, NRCVH, XCV, NICVH, ICV, NRFL, XFL, NIFL, IFL, NICAV, ICAV, &
  NRSPPR, XSPR, NISPR, ISPR, NLSPPR, LSPR, NCSPPR, CSPR, NRSC, XSC, &
  NISC, ISC)
!
USE RN2_MDL
. . . . .
!! NOTICE(2): Converted by hand
!      CALL RN2RNO (NNVOL, NUMSEC, NUMCLS, XRN2 (IT2ARG), XRN2 (IT2ARL), &
!      XRN2 (IT2VPG), XRN2 (IT2VPL), XRN2 (IR2ARG), XRN2 (IR2ARL), &
!      XRN2 (IR2VPG), XRN2 (IR2VPL), XRN1 (IT1AGN - J1), XRN1 (IT1ALN - &
!      J1), XRN1 (IT1VGN - J1), XRN1 (IT1VLN - J1), XRN1 (IR1AGN - J1), &
!      XRN1 (IR1ALN - J1), XRN1 (IR1VGN - J1), XRN1 (IR1VLN - J1), &
!      XCV (KCVVON - IC), ICV (KCVNUM) )
!! NOTICE: Converted by MELtoMID
CALL RN2RNO (NNVOL, NUMSEC, NUMCLS, &
  XRN1 (IT1AGN - J1), XRN1 (IT1ALN - &
  J1), XRN1 (IT1VGN - J1), XRN1 (IT1VLN - J1), XRN1 (IR1AGN - J1), &
  XRN1 (IR1ALN - J1), XRN1 (IR1VGN - J1), XRN1 (IR1VLN - J1), &
  XCV (KCVVON - IC), ICV (KCVNUM) )
. . . . .
!
RETURN
END SUBROUTINE RN2DBD
```

<> subroutine RN2RNO

```
      SUBROUTINE RN2RNO (NNVOL, NSEC, NCLS, AER2G, AER2L, VAP2G, VAP2L, &
      RDA2G, RDA2L, RDV2G, RDV2L, AER1G, AER1L, VAP1G, VAP1L, RDA1G, &
      RDA1L, RDV1G, RDV1L, VOL, ICVNUM)
      . . . . .

      DO 1000, J = 1, NNVOL
        IF (VOL (2, J) + VOL (3, J) .GT.PZERO) THEN
          IF (VOL (1, J) .GT.PZERO) THEN
! -----
! THERE IS POOL AND ATMOSPHERE
! -----
            DO 200, I = 1, NCLS
              DO 100, K = 1, NSEC
! TRANSFER AEROSOLS IN ATMOSPHERE
                AER2G (K, I, J) = AER1G (K, I, J)
                RDA2G (K, I, J) = RDA1G (K, I, J)
100          END DO
            . . . . .

            AER2L (NCLSW, J) = PZERO
            RDA2L (NCLSW, J) = PZERO
            VAP2L (NCLSW, J) = PZERO
            RDV2L (NCLSW, J) = PZERO
1000 END DO
!
      RETURN
      END SUBROUTINE RN2RNO
```

<> subroutine RN2RNO

```
      SUBROUTINE RN2RNO (NNVOL, NSEC, NCLS, &
      AER1G, AER1L, VAP1G, VAP1L, RDA1G, &
      RDA1L, RDV1G, RDV1L, VOL, ICVNUM)

      USE RN2_MDL
      . . . . .

      DO 1000, J = 1, NNVOL
        IF (VOL (2, J) + VOL (3, J) .GT.PZERO) THEN
          IF (VOL (1, J) .GT.PZERO) THEN
! -----
! THERE IS POOL AND ATMOSPHERE
! -----
            DO 200, I = 1, NCLS
              DO 100, K = 1, NSEC
! TRANSFER AEROSOLS IN ATMOSPHERE
                RN2_NCVS(K, I, J)%AER2G = AER1G (K, I, J)
                RN2_NCVS(K, I, J)%RDA2G = RDA1G (K, I, J)
100          END DO
            . . . . .

            RN2_NCNV(NCLSW, J)%AER2L = PZERO
            RN2_NCNV(NCLSW, J)%RDA2L = PZERO
            RN2_NCNV(NCLSW, J)%VAP2L = PZERO
            RN2_NCNV(NCLSW, J)%RDV2L = PZERO
1000 END DO
!
      RETURN
      END SUBROUTINE RN2RNO
```

RN2 data RN2 CVH(1 subroutine), FL(1 subroutine), RN1
 (9 subroutine) SPR (3 subroutine) ,
 subroutine . subroutine .

- common block

RN2CFA.F90	RN2CFB.F90	RN2DB1.F90	RN2DB2.F90	RN2DB3.F90
RN2DB4.F90	RN2DB5.F90	RN2DB6.F90	RN2DBC.F90	RN2DBD.F90
RN2DBE.F90	RN2DBF.F90	RN2DBM.F90	RN2DBN.F90	RN2DBY.F90
RN2DBZ.F90	RN2MOV.F90	RN2MOW.F90	RN2PBD.F90	RN2PRS.F90
CVHDB1.F90	FLDB1.F90	RN1DBC.F90	RN1DBD.F90	RN1DBE.F90
RN1DBF.F90	RN1DBQ.F90	RN1DBR.F90	SPRDBC.F90	SPRDBD.F90

- module 가

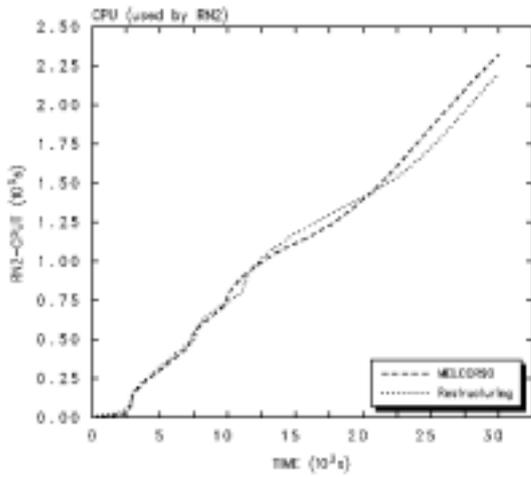
RN2CFC.F90	RN2EDT.F90	RN2MDS.F90	RN2MOX.F90	RN2PS2.F90
RN2PSA.F90	RN2PSB.F90	RN2RNO.F90	RN2RN1.F90	RN2RN2.F90
RN2SPR.F90	(RN2PBD.F90)	RN1ED1.F90	RN1ED5.F90	RN1RN6.F90
SPRRUN.F90				

- MEGEDH.F90 MEXEDH.F90 MXXRS.F90

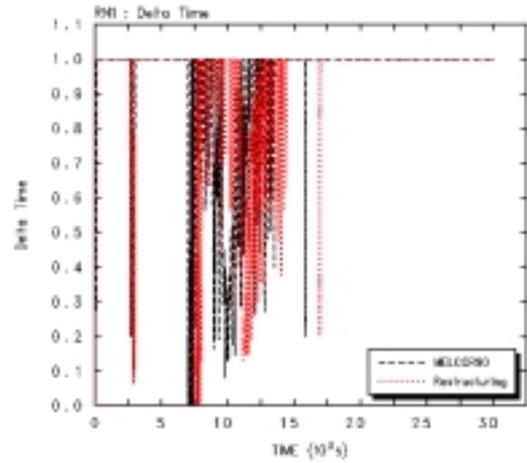
- 가
 RN2FLG.F90 RN2MDL.F90 RN2RES.F90

4.

FORTRAN90 ,
 RN2 FORTRAN77 MELCOR FORTRAN90
 , library execution file . FORTRAN90
 MELCOR RN2MELCOR RN2
 .
 RN2 , ,
 가 , flow path convection , RN2
 convection switch . PWR type SBO , RN
 가 RN1000 card . 30
 , 100 , 30,000 .
 MELGEN MELCOR subroutine data
 file(restart file) read/write . MELGEN restart file
 write가 , MELCOR data file(restart file) read ,
 RN2
 Unformatted print 가 subroutine RN2
 ,
 file write UNIX diff command text file . text data
 file(restart file) read/write .
 , FORTRAN90 MELCOR
 RN2MELCOR RN2 , 10
 13 .

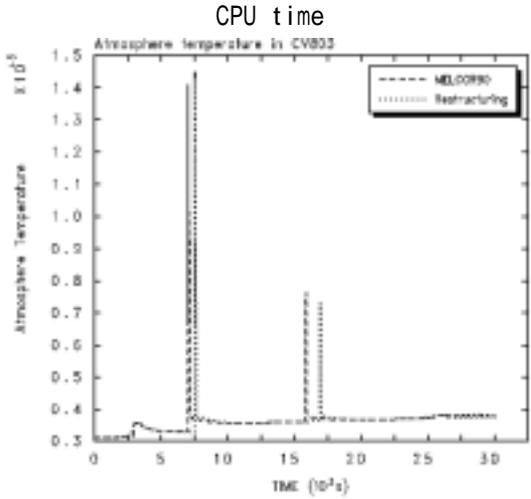


10. RN2



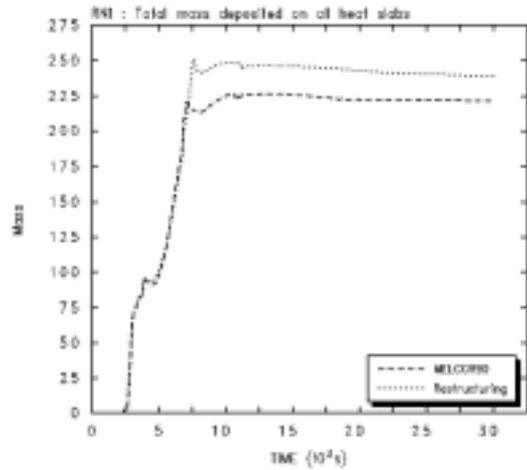
11.

(dt)



12. RN2

CV803



13.

10 RN2 CPU time
 가 array /
 even/odd cycle (delta time) 2500

11 7500 15000
 가
 12 803 , RN2 7000 8000 16000 17000
 가 가 10% 500 1500 7000

unformatted print text file
 (aerosol mass in gas, aerosol mass in liquid, vapor mass in gas, vapor mass in liquid) , 2777 cycle(2300 sec)
 0.0000003 × 10⁻¹⁰ 가 , 11
 가

5.

MIDAS, MELCOR
 RN2, RN2
 RN2, subroutine, RN2
 graph
 BH, COR, ESF, HS, double precision, single
 precision
 (dt), (dt)
 (dt), 가
 graph, 가
 benchmarking, problem set, RN1
 RN, /
 RN2, 가
 RN2
 [11], 가 가

[1] A Multi-Dimensional Thermal-Hydraulic System Analysis Code, MARS 1.3.1, Vol.31, Number 3, pp.344-363, June 1998.
 [2] Realistic Thermal-Hydraulic System Code Development Workshop, '98 Fall KNS Conference, 1998.
 [3] 3rd MARS Users Group Meeting, '99 Dec. 2 KAIST Conference, 1999.
 [4] , , Development of a Computer Program for Automatic Variable Conversion in MELCOR Code, , 2000
 [5] , , MIDAS , KAERI/TR-2220/2002, Jun, 2002.
 [6] , , A Restructuring Proposal for MIDAS, , 2000.
 [7] S.H.Park, H.D.Kim, D.H.Kim, Y.M.Song, B.D.Chung, Development of Restructuring Template for MELCOR, 5th PSAM Conference, Japan, 2000.11.26 2000.12.2
 [8] , , A Restructuring of TF package for MIDAS Computer Code, , 2002.
 [9] , , MELCOR , KAERI/TR-1543/00, June, 2000.
 [10] , , MELCOR , KAERI/TR-1536/2000, March, 2000

[11] , 12 ,
2216/2001, pp 302-321, May, 2002.

, KAERI/RR-