536665

Sandia National Laboratories

Operated for the U.S. Department of Energy by
Sandia Corporation

4100 National Parks Highway-
Carlsbad< NM 88220

date: September 9, 2004

to: Yongliang Xiong (Org. 6822)

from: Joseph Kanney (Org. 6821) 9/9/04
William Zelinski (Org. 6821) William Zelinski 9/9/04

Technical Review: Eric Vugrin (Org. 6821) 9/9/04

QA Review: Mario Chavez (Org. 6820) Mario Chavez 9/10/04

Management Review: Dave Kessel (Org. 6821) 9/9/04

subject: Input for $CaCO_3$ Precipitation Modeling

Introduction

This memo provides an estimate for the brine volume and brine-to-rock volume ratio in the waste panel area and the surrounding damaged rock zone (DRZ) over the course of the 10,000-year regulatory period. These estimates were obtained by analyzing the results of BRAGFLO calculations performed by Stein and Zelinski (2003) for the WIPP 2004 Compliance Recertification Application (2004 CRA, U.S. DOE 2004).

Ranges for these quantities were calculated. The brine volume and brine-to-rock volume ratio varies with time due to changes in porosity and brine saturation. The time history of these quantities also varies from realization to realization under the influence of sampled parameters in performance assessment (PA) modeling.

This work was carried out under Analysis Plan AP-112, to support Recertification Response Committee Action Item 05/20/04T: response to comment C-23-5 in EPA's first set of CRA comments (Cotsworth 2004)

Brine Volume and Brine-to-Rock Volume Ratio

Two approaches were used to estimate the volumes and ratios of interest: 1) averaging over cell layers in the DRZ adjacent to the waste panel (the DRZ layer average); and 2) averaging over all cells comprising both the waste panel and the DRZ (the mixing cell average). The former approach is consistent with a scenario in which sufficient CO_2 has diffused into the DRZ such that the locus of the geochemical reactions is in close proximity to DRZ minerals. The latter approach is more consistent with the

WIPP:1.4.1.2:PA:QA-L:533999

conceptual model used in PA that views the repository geochemical system as a well mixed batch reactor.

The BRAGFLO grid used in the 2004 CRA calculations is specified in the GENMESH input file GM_BF_CRA1.INP, stored in CMS library CRA1_GM on the WIPP PA Alpha cluster. The BRAGFLO grid in the vicinity of the waste panel, detailing regions referenced in this analysis, is shown in Figure 1.

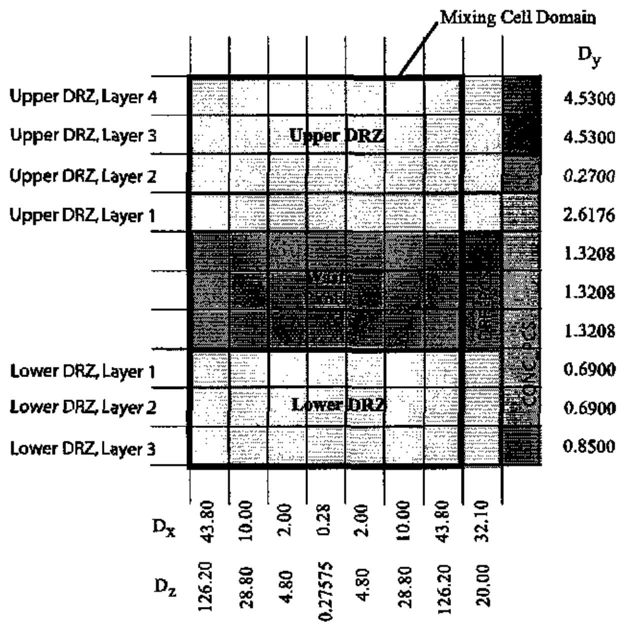

Figure 1. BRAGFLO Grid inVicinity of the Waste Panel

The cell ranges and thicknesses for the regions of interest are shown in Table 1. The upper DRZ is thicker than the lower DRZ because more disruption is expected above the repository as the ceiling closes into the open space of the waste panels.

Table 1. Regions and Cell Ranges in Vicinity of Waste Panel

Region	Symbol	Cell Range	Thickness (m)
Upper DRZ, 4th Layer	UD4	1189-1195	4.5300
Upper DRZ, 3rd Layer	UD3	1181-1187	4.5300
Upper DRZ, 2nd Layer	UD2	1173-1179	0.2700
Upper DRZ, 1st Layer	UD1	1165-1171	2.6167
Waste Panel	WP	1407-1427	3.9624
Lower DRZ, 1st Layer	LD1	1124-1130	0.6900
Lower DRZ, 2nd Layer	LD2	1116-1122	0.6900
Lower DRZ, 3rd Layer	LD3	1108-1114	0.8500

The brine volume in the l^{th} region is given by

$$V_b^l = \sum_{k \in C^l} V_k \phi_k (1 - S_k^g) \qquad (1)$$

where V_b^l is the brine volume in region l (m^3), C^l is the set of indices for cells that comprise region l, V_k is the volume of the k^{th} cell, ϕ_k is the porosity of the k^{th} cell, S_k^g is the gas saturation of the k^{th} cell.

The rock volume in the l^{th} region is given by

$$V_r^l = \sum_{k \in C^l} V_k (1 - \phi_k) \qquad (2)$$

where V_r^l is rock volume in region l (m^3). Finally, the brine-to-rock volume ratio in the l^{th} region is given by

$$\frac{V_b^l}{V_r^l} = \frac{\sum_{k \in C^l} V_k \phi_k (1 - S_k^g)}{\sum_{k \in C^l} V_k (1 - \phi_k)} \qquad (3)$$

Brine volumes and brine-to-rock volume ratios in each region were calculated for all 100 vectors in replicate R1, for scenarios S1 and S2. The volume, porosity, and gas saturation for individual cells were extracted from POSTBRAGFLO output files BF3_CRA1_R1_S*s*_V*vvv*.CDB, where $s \in \{1,2\}$, and $vvv \in \{001, 002, ..., 100\}$. The correspondence between the POSTBRAGFLO variable names and the variables in the equations presented here is shown in Table 2.

Table 2. POSTBRAGFLO Variable Name Crosswalk

Property	Variable Name	
	This Analysis	POSTBRAGFLO
Grid cell volume	V_k	GRIDVOL
Porosity	ϕ_k	POROS
Gas phase saturation	S_k^g	SATGAS

An ALGEBRA script was written to calculate brine volume, rock volume, and brine-to-rock volume ratio over the regions of interest at each time step for all vectors in each scenario. The ALGEBRA script also calculates the maximum and minimum values for these quantities over all time steps in each vector for each scenario. The SUMMARIZE code was used to compile all of the vector-wise minimum and maximum values into a table for each scenario. A FORTRAN utility code (MMOV) was used to sort through the table to identify the minimum and maximum values over all vectors. The details of these calculations are provided in Appendix A. The ALGEBRA script and output files, the SUMMARIZE input and output files, the MMOV source code and executable, and MMOV output files are stored in CMS library LIBCRA1V_BRVOL.

DRZ Layer Average Approach

Tables 3 and 4 summarize the computed range for brine volume and brine-to-rock volume ratio in each layer for scenarios S1 and S2, respectively.

Table 3. Brine Volume and Brine-to-Rock Volume Ratio Range in Each Layer (CRA1_R1_S1)

Layer	Brine Volume V_b^l (m^3)		Brine-to-Rock Volume Ratio V_b^l / V_r^l (-)	
	Min	Max	Min	Max
Upper DRZ, 4th Layer (UD4)	2.188E+00	1.329E+03	4.149E-05	2.583E-02
Upper DRZ, 3rd Layer (UD3)	2.348E+00	1.323E+03	4.452E-05	2.572E-02
Upper DRZ, 2nd Layer (UD2)	0.155E+00	7.369E+01	4.915E-05	2.403E-02
Upper DRZ, 1st Layer (UD1)	1.521E+00	7.043E+02	4.986E-05	2.367E-02
Lower DRZ, 1st Layer (LD1)	1.146E+00	3.350E+02	1.433E-04	4.349E-02
Lower DRZ, 2nd Layer (LD2)	1.461E+00	3.430E+02	1.834E-04	4.457E-02
Lower DRZ, 3rd Layer (LD3)	3.702E+00	4.225E+02	3.740E-04	4.457E-02

Table 4. Brine Volume and Brine-to-Rock Volume Ratio Range in Each Layer (CRA1_R1_S2)

Layer	Brine Volume V_b^l (m^3)		Brine-to-Rock Volume Ratio V_b^l / V_r^l (-)	
	Min	Max	Min	Max
Upper DRZ, 4th Layer (UD4)	2.188E+00	1.655E+03	4.149E-05	3.241E-02
Upper DRZ, 3rd Layer (UD3)	2.348E+00	1.723E+03	4.452E-05	3.376E-02
Upper DRZ, 2nd Layer (UD2)	0.155E+00	1.095E+02	4.915E-05	3.606E-02
Upper DRZ, 1st Layer (UD1)	1.521E+00	1.073E+03	4.986E-05	3.643E-02
Lower DRZ, 1st Layer (LD1)	1.345E+00	3.119E+02	1.680E-04	4.036E-02
Lower DRZ, 2nd Layer (LD2)	2.080E+00	3.119E+02	2.596E-04	4.037E-02
Lower DRZ, 3rd Layer (LD3)	3.703E+00	3.843E+02	3.740E-04	4.037E-02

Mixing Cell Average Approach

The range of calculated brine volumes for the mixing cell average approach is shown in Table 5.

Table 5. Brine Volume (Mixing Cell Average Approach)

Scenario	Min	Max
CRA1_R1_S1	3.787E+01	6.658E+03
CRA1_R1_S2	5.037E+01	1.635E+04

When calculating the brine to rock volume ratio for the mixing-cell average approach, the rock volume is estimated in two ways: 1) excluding the waste panel; and 2) including the waste panel. Because of room closure, the true rock volume should be somewhere between these two estimates (The BRAGFLO code uses a fixed computational grid with a time-varying porosity to model the effects of room closure). Both types of ratios are shown in Table 6.

Table 6. Brine-to-Rock Volume Ratio (Mixing Cell Average Approach)

Scenario	Waste Panel Included in Rock Volume		Waste Panel Omitted From Rock Volume	
	Min	Max	Min	Max
CRA1_R1_S1	1.887E-04	3.300E-02	2.305E-04	4.127E-02
CRA1_R1_S2	2.558E-04	8.443E-02	3.062E-04	1.021E-01

References

Cotsworth, E. 2004. First Set of CRA Comments (May 20, 2004 Letter to R. Paul Detweiler, Acting Manger, Carlsbad Field Office, U.S. Department of Energy). U.S. Environmental Protection Agency.

Stein, J. S., and W. Zelinski. 2003. Analysis Package for BRAGFLO: Compliance Recertification Application. Analysis Report ERMS 530163, Sandia National Laboratories, Carlsbad, NM.

U.S. DOE. 2004. Title 40 CFR Part 191 Compliance Recertification Application for the Waste Isolation Pilot. DOE/WIPP 2004-3231, U.S. Department of Energy Waste Isolation Pilot Plant, Carlsbad Field Office, Carlsbad, NM.

Appendix A. Calculation of Brine Volumes and Brine-to Rock Volume Ratios

This section describes the procedure for calculating the brine volumes and brine-to-rock volume ratios. The procedure can be summarized as follows (Steps 1-4 are repeated for each scenario):

Step 1. Extract BRAGFLO output files from CMS.
Step 2. Find the minimum and maximum values for volumes and ratios over all time steps for each vector using the ALGEBRA code.
Step 3. Combine the results for all vectors into a single table using the SUMMARIZE code.
Step 4. Extract the minimum and maximum values over all vectors from the SUMMARIZE table using MMOV, a FORTRAN utility code.

Extract BRAGFLO Output Files from CMS

Log onto the VMS system and create a working directory. The following commands will fetch the BRAGFLO output files for replicate R1, scenarios S1 and S2:

```
$ libcra1_bfr1s1
$ cms fetch bf3_cra1_r1_s1_v*.cdb
$ libcra1_bfr1s2
$ cms fetch bf3_cra1_r1_s2_v*.cdb
```

These commands extract all 100 CDB files for each scenario.

Minimum and Maximum Volumes and Ratios Over All Time Steps for Each Vector

Brine volumes and brine-to-rock volume ratios for the regions of interest were calculated with the ALGEBRA code using the input script file ALG_BFVOL_MINMAX_CRA1.INP shown below. This script also directs calculation of the minimum and maximum values of the volume and ratios over all time steps in each vector. Descriptions of the variables defined in the ALGEBRA script are shown in Table A1.

Table A1. Variable Names in ALGEBRA script

ALGEBRA Variable	Description
VB_UD4	Volume of Brine, Upper DRZ, Layer 4
VB_UD3	Volume of Brine, Upper DRZ, Layer 3
VB_UD2	Volume of Brine, Upper DRZ, Layer 2
VB_UD1	Volume of Brine, Upper DRZ, Layer 1
VB_WP	Volume of Brine, Waste Panel
VB_LD1	Volume of Brine, Lower DRZ, Layer 1
VB_LD2	Volume of Brine, Lower DRZ, Layer 2
VB_LD3	Volume of Brine, Lower DRZ, Layer 3
VB_TOT	Volume of Brine, Total
VR_UD4	Volume of Brine, Upper DRZ, Layer 4
VR_UD3	Volume of Brine, Upper DRZ, Layer 3
VR_UD2	Volume of Brine, Upper DRZ, Layer 2
VR_UD1	Volume of Brine, Upper DRZ, Layer 1
VR_WP	Volume of Brine, Waste Panel
VR_LD1	Volume of Brine, Lower DRZ, Layer 1
VR_LD2	Volume of Brine, Lower DRZ, Layer 2
VR_LD3	Volume of Brine, Lower DRZ, Layer 3
VR_TOT1	Volume of Rock, Total (including WP)
VR_TOT2	Volume of Rock, Total (excluding WP)
BR_UD4	Brine-to-Rock Volume Ratio, Upper DRZ, Layer 4
BR_UD3	Brine-to-Rock Volume Ratio, Upper DRZ, Layer 3
BR_UD2	Brine-to-Rock Volume Ratio, Upper DRZ, Layer 2
BR_UD1	Brine-to-Rock Volume Ratio, Upper DRZ, Layer 1
BR_LD1	Brine-to-Rock Volume Ratio, Lower DRZ, Layer 1
BR_LD2	Brine-to-Rock Volume Ratio, Lower DRZ, Layer 2
BR_LD3	Brine-to-Rock Volume Ratio, Lower DRZ, Layer 3
BRTOT1	Brine-to-Rock Volume Ratio, Total (including WP)
BRTOT2	Brine-to-Rock Volume Ratio, Total (excluding WP)

ALG_BFVOL_MINMAX_CRA1.INP

```
!========================================================================
!
! ALG_BFVOL_MINMAX_CRA1.INP
!
! ALGEBRA file (Post-Brag) for calculating brine and rock volumes
! in vicinity of the waste panel
!
! Author: Joseph Kanney, SNL Org. 6821
! Date  : 02 September, 2004
!========================================================================
!
!
! This ALGEBRA script computes brine volumes, rock volumes, and brine
! to rock volume ratios for several regions in the vicinity of the
! waste panel.  We look at layers within the DRZ as well as the
! region composed of the entire DRZ plus the waste panel.  The
! schematic below indicates the labeling convention adopted for these
! calculations.
!
!     -----------------
!     UPPER DRZ Layer 4 (UD4)
!     -----------------
!     UPPER DRZ Layer 3 (UD3)
!     -----------------
!     UPPER DRZ Layer 2 (UD2)
!     -----------------
!     UPPER DRZ Layer 1 (UD1)
!     -----------------
!
!      Waste Panel      (WP)
!
!     -----------------
!     LOWER DRZ Layer 1 (LD1)
!     -----------------
!     LOWER DRZ Layer 2 (LD2)
!     -----------------
!     LOWER DRZ Layer 3 (LD3)
!     -----------------
!
!---------------------------------------------------------------------
!---------------------------------------------------------------------
!
! Eliminate excess output
!
  DELETE ALL
!
!*****************************************************
! Compute brine and rock volume in 4th layer of upper drz
!*****************************************************
!
  LIMIT ELEMENT 1189 to 1195
!
  VB_UD4 = SUM( GRIDVOL*POROS*(1.-SATGAS) )
  VR_UD4 = SUM( GRIDVOL*(1.-POROS) )
!
!
!*****************************************************
! Compute brine and rock volume in 3rd layer of upper drz
```

```
!********************************************************
!
  LIMIT ELEMENT 1181 to 1187
!
  VB_UD3 = SUM( GRIDVOL*POROS*(1.-SATGAS) )
  VR_UD3 = SUM( GRIDVOL*(1.-POROS) )
!
!
!********************************************************
! Compute brine and rock volume in 2nd layer of upper drz
!********************************************************
!
  LIMIT ELEMENT 1173 to 1179
!
  VB_UD2 = SUM( GRIDVOL*POROS*(1.-SATGAS) )
  VR_UD2 = SUM( GRIDVOL*(1.-POROS) )
!
!
!********************************************************
! Compute brine and rock volume in 1st layer of upper drz
!********************************************************
!
  LIMIT ELEMENT 1165 to 1171
!
  VB_UD1 = SUM( GRIDVOL*POROS*(1.-SATGAS) )
  VR_UD1 = SUM( GRIDVOL*(1.-POROS) )
!
!
!********************************************************
! Compute brine and rock volume in waste panel
!********************************************************
!
  LIMIT ELEMENT 1407 to 1427
!
  VB_WP    = SUM( GRIDVOL*POROS*(1.-SATGAS) )
  VR_WP    = SUM( GRIDVOL*(1.-POROS) )
!
!
!********************************************************
! Compute brine and rock volume in 1st layer of lower drz
!********************************************************
!
  LIMIT ELEMENT 1124 to 1130
!
  VB_LD1 = SUM( GRIDVOL*POROS*(1.-SATGAS) )
  VR_LD1 = SUM( GRIDVOL*(1.-POROS) )
!
!
!********************************************************
! Compute brine and rock volume in 2nd layer of lower drz
!********************************************************
!
  LIMIT ELEMENT 1116 to 1122
!
  VB_LD2 = SUM( GRIDVOL*POROS*(1.-SATGAS) )
  VR_LD2 = SUM( GRIDVOL*(1.-POROS) )
!
!
!********************************************************
! Compute brine and rock volume in 3rd layer of lower drz
!********************************************************
```

```
!
  LIMIT ELEMENT 1108 to 1114
!
  VB_LD3 = SUM( GRIDVOL*POROS*(1.-SATGAS) )
  VR_LD3 = SUM( GRIDVOL*(1.-POROS) )
!
!
!
!********************************************************
! Compute brine to rock volume ratios layer by layer
!********************************************************
!
  BR_UD4 = VB_UD4/VR_UD4
  BR_UD3 = VB_UD3/VR_UD3
  BR_UD2 = VB_UD2/VR_UD2
  BR_UD1 = VB_UD1/VR_UD1
!
  BR_LD1 = VB_LD1/VR_LD1
  BR_LD2 = VB_LD2/VR_LD2
  BR_LD3 = VB_LD3/VR_LD3
!
!
!********************************************************
! Compute total volumes and overall brine to rock volume ratios
!********************************************************
!
! Total brine volume
!
  VB_UD  = VB_UD1 + VB_UD2 + VB_UD3 + VB_UD4
  VB_LD  = VB_LD1 + VB_LD2 + VB_LD3
  VB_TOT = VB_UD + VB_LD + VB_WP
!
! Total rock volume
!
! VR_TOT1 includes waste panel solids as rock
! VR_TOT2 exludes waste panel solids
!
  VR_UD   = VR_UD1 + VR_UD2 + VR_UD3 + VR_UD4
  VR_LD   = VR_LD1 + VR_LD2 + VR_LD3
  VR_TOT1 = VR_UD + VR_LD + VR_WP
  VR_TOT2 = VR_UD + VR_LD
!
! Brine to rock volume ratios
!
  BRTOT1 = VB_TOT/VR_TOT1
  BRTOT2 = VB_TOT/VR_TOT2
!
!
!***********************************************************
! Compute max/min for volumes and ratios over all time steps
!***********************************************************
!
! Max brine volumes
!
  VB_UD4MX = ENVMAX(VB_UD4)
  VB_UD3MX = ENVMAX(VB_UD3)
  VB_UD2MX = ENVMAX(VB_UD2)
  VB_UD1MX = ENVMAX(VB_UD1)
!
  VB_LD1MX = ENVMAX(VB_LD1)
  VB_LD2MX = ENVMAX(VB_LD2)
```

```
  VB_LD3MX = ENVMAX(VB_LD3)
!
  VB_WPMX  = ENVMAX(VB_WP)
  VB_TOTMX = ENVMAX(VB_TOT)
!
! Min brine volumes
!
  VB_UD4MN = ENVMIN(VB_UD4)
  VB_UD3MN = ENVMIN(VB_UD3)
  VB_UD2MN = ENVMIN(VB_UD2)
  VB_UD1MN = ENVMIN(VB_UD1)
!
  VB_LD1MN = ENVMIN(VB_LD1)
  VB_LD2MN = ENVMIN(VB_LD2)
  VB_LD3MN = ENVMIN(VB_LD3)
!
  VB_WPMN = ENVMIN(VB_WP)
  VB_TOTMN = ENVMIN(VB_TOT)
!
!
! Max rock volumes
!
  VR_UD4MX = ENVMAX(VR_UD4)
  VR_UD3MX = ENVMAX(VR_UD3)
  VR_UD2MX = ENVMAX(VR_UD2)
  VR_UD1MX = ENVMAX(VR_UD1)
!
  VR_LD1MX = ENVMAX(VR_LD1)
  VR_LD2MX = ENVMAX(VR_LD2)
  VR_LD3MX = ENVMAX(VR_LD3)
!
  VR_WPMX = ENVMAX(VR_WP)
!
  VRTOT1MX = ENVMAX(VR_TOT1)
  VRTOT2MX = ENVMAX(VR_TOT2)
!
! Min rock volumes
!
  VR_UD4MN = ENVMIN(VR_UD4)
  VR_UD3MN = ENVMIN(VR_UD3)
  VR_UD2MN = ENVMIN(VR_UD2)
  VR_UD1MN = ENVMIN(VR_UD1)
!
  VR_LD1MN = ENVMIN(VR_LD1)
  VR_LD2MN = ENVMIN(VR_LD2)
  VR_LD3MN = ENVMIN(VR_LD3)
!
  VR_WPMN = ENVMIN(VR_WP)
!
  VRTOT1MN = ENVMIN(VR_TOT1)
  VRTOT2MN = ENVMIN(VR_TOT2)
!
! Max brine to rock volume ratios
!
  BR_UD4MX = ENVMAX(BR_UD4)
  BR_UD3MX = ENVMAX(BR_UD3)
  BR_UD2MX = ENVMAX(BR_UD2)
  BR_UD1MX = ENVMAX(BR_UD1)
!
  BR_LD1MX = ENVMAX(BR_LD1)
  BR_LD2MX = ENVMAX(BR_LD2)
```

```
  BR_LD3MX = ENVMAX(BR_LD3)
!
  BRTOT1MX = ENVMAX(BRTOT1)
  BRTOT2MX = ENVMAX(BRTOT2)
!
! Min brine to rock volume ratios
!
  BR_UD4MN = ENVMIN(BR_UD4)
  BR_UD3MN = ENVMIN(BR_UD3)
  BR_UD2MN = ENVMIN(BR_UD2)
  BR_UD1MN = ENVMIN(BR_UD1)
!
  BR_LD1MN = ENVMIN(BR_LD1)
  BR_LD2MN = ENVMIN(BR_LD2)
  BR_LD3MN = ENVMIN(BR_LD3)
!
  BRTOT1MN = ENVMIN(BRTOT1)
  BRTOT2MN = ENVMIN(BRTOT2)
!
!
!
END
```

The ALGEBRA script is used to process all 100 vectors in scenarios S1 and S2. The commands for scenario S1 are shown below.

```
$ algebra   bf3_cra1_r1_s1_v001.cdb alg_bfvol_minmax_cra1_r1_s1_v001.cdb -
            alg_bfvol_minmax_cra1.inp alg_bfvol_minmax_cra1_r1_s1_v001.log

$ algebra   bf3_cra1_r1_s1_v002.cdb alg_bfvol_minmax_cra1_r1_s1_v002.cdb -
            alg_bfvol_minmax_cra1.inp alg_bfvol_minmax_cra1_r1_s1_v002.log
       .
       .
       .
$ algebra   bf3_cra1_r1_s1_v100.cdb alg_bfvol_minmax_cra1_r1_s1_v100.cdb -
            alg_bfvol_minmax_cra1.inp alg_bfvol_minmax_cra1_r1_s1_v100.log
```

The resulting CDB files are named ALG_BFVOL_MINMAX_CRA1_R1_S*s*_V*nnn*.CDB, where $s \in \{1,2\}$, and $nnn \in \{001,002,...,100\}$.

Combine Results From All Vectors Into a Single Table.

SUMMARIZE scripts SUM_BFVOL_MINMAX_CRA1_R1_S1.INP and SUM_BFVOL_MINMAX_CRA1_R1_S2.INP were used to combine the minimum and maximum values from all vectors into a single table for scenarios S1 and S2, respectively (one will need to edit the directory and name lines to reflect one's working directory):

The SUMMARIZE scripts are run using the following commands:

```
$ summarize  sum_bfvol_minmax_cra1_r1_s1.inp
$ summarize  sum_bfvol_minmax_cra1_r1_s2.inp
```

These commands will produce the following output files:

SUM_BFVOL_MINMAX_CRA1_R1_S1.TBL
SUM_BFVOL_MINMAX_CRA1_R1_S2.TBL

SUM_BFVOL_MINMAX_CRA1_R1_S1.INP

```
!=================================================================
!
! SUM_BFVOL_MINMAX_CRA1_R1_S1.INP
!
! This SUMMARIZE input file is used to compile the results
! of running the ALGEBRA script ALG_BFVOL_MINMAX_CRA1.INP
! on all 100 vectors for scenario S1 into a single table.
!
! Author: Joseph F. Kanney, SNL Org 6821
! Date  : 02 September 2004
!=================================================================
!
*input files
template = alg_bfvol_minmax_cra1_r1_s1_v###
disk = paa:
directory = [shared.jfkanne.wrk.cra_resp.caco3_inp.cdb_alg]
type = CDB

*vector
id = #
vector = 1 to 100

*times
read = seconds
input = years
output = years
steps = last

*items
type =global
name = vb_ud4mx, vb_ud4mn, &
       vb_ud3mx, vb_ud3mn, &
       vb_ud2mx, vb_ud2mn, &
       vb_ud1mx, vb_ud1mn, &
       vb_ld1mx, vb_ld1mn, &
       vb_ld2mx, vb_ld2mn, &
       vb_ld3mx, vb_ld3mn, &
       vb_wpmx,  vb_wpmn,  &
       vb_totmx, vb_totmn, &
       vr_ud4mx, vr_ud4mn, &
       vr_ud3mx, vr_ud3mn, &
       vr_ud2mx, vr_ud2mn, &
       vr_ud1mx, vr_ud1mn, &
       vr_ld1mx, vr_ld1mn, &
       vr_ld2mx, vr_ld2mn, &
       vr_ld3mx, vr_ld3mn, &
       vr_wpmx,  vr_wpmn,  &
       vrtot1mx, vrtot1mn, &
       vrtot2mx, vrtot2mn, &
       br_ud4mx, br_ud4mn, &
       br_ud3mx, br_ud3mn, &
       br_ud2mx, br_ud2mn, &
       br_ud1mx, br_ud1mn, &
       br_ld1mx, br_ld1mn, &
       br_ld2mx, br_ld2mn, &
       br_ld3mx, br_ld3mn, &
       brtot1mx, brtot1mn, &
```

```
       brtot2mx, brtot2mn
*output
driver = text
write = vector vs item
SINGLE_FILE = concatenate sets vertical
name = [.tbl]sum_bfvol_minmax_cra1_r1_s1
EXTENSION = TBL

*end
```

SUM_BFVOL_MINMAX_CRA1_R1_S2.INP

```
!==============================================================
!
! SUM_BFVOL_MINMAX_CRA1_R1_S2.INP
!
! This SUMMARIZE input file is used to compile the results
! of running the ALGEBRA script ALG_BFVOL_MINMAX_CRA1.INP
! on all 100 vectors for scenario S2 into a single table.
!
! Author: Joseph F. Kanney, SNL Org 6821
! Date  : 02 September 2004
!==============================================================
!
*input files
template = alg_bfvol_minmax_cra1_r1_s2_v###
disk = paa:
directory = [shared.jfkanne.wrk.cra_resp.caco3_inp.cdb_alg]
type = CDB

*vector
id = #
vector = 1 to 100

*times
read = seconds
input = years
output = years
steps = last

*items
type =global
name = vb_ud4mx, vb_ud4mn, &
       vb_ud3mx, vb_ud3mn, &
       vb_ud2mx, vb_ud2mn, &
       vb_ud1mx, vb_ud1mn, &
       vb_ld1mx, vb_ld1mn, &
       vb_ld2mx, vb_ld2mn, &
       vb_ld3mx, vb_ld3mn, &
       vb_wpmx,  vb_wpmn,  &
       vb_totmx, vb_totmn, &
       vr_ud4mx, vr_ud4mn, &
       vr_ud3mx, vr_ud3mn, &
       vr_ud2mx, vr_ud2mn, &
       vr_ud1mx, vr_ud1mn, &
       vr_ld1mx, vr_ld1mn, &
       vr_ld2mx, vr_ld2mn, &
       vr_ld3mx, vr_ld3mn, &
       vr_wpmx,  vr_wpmn,  &
       vrtot1mx, vrtot1mn, &
       vrtot2mx, vrtot2mn, &
       br_ud4mx, br_ud4mn, &
       br_ud3mx, br_ud3mn, &
       br_ud2mx, br_ud2mn, &
       br_ud1mx, br_ud1mn, &
       br_ld1mx, br_ld1mn, &
       br_ld2mx, br_ld2mn, &
       br_ld3mx, br_ld3mn, &
       brtot1mx, brtot1mn, &
```

```
       brtot2mx, brtot2mn
*output
driver = text
write = vector vs item
SINGLE_FILE = concatenate sets vertical
name = [.tbl]sum_bfvol_minmax_cra1_r1_s2
EXTENSION = TBL

*end
```

Extract the minimum and maximum values over all vectors from the SUMMARIZE table

The following FORTRAN utility code was written to extract minimum and maximum values from the columns of the SUMMARIZE output files. Since the MMOV code merely extracts data from the table, its correct operation was verified by visual inspection.

MMOV.F

```
C-------------------------------------------------------------------
C MMOV.F (MinMax over vectors)
C
C Author: Joseph F. Kanney, SNL Org 6821
C Date  : 02 Sept 2004
C-------------------------------------------------------------------
C
C This FORTRAN code is designed to process a SUMMARIZE output
C table written with the "text" output driver. The SUMMARIZE file
C has three lines of header information followed  by a data set
C of size N rows x M columns. The first two columns contain the
C vector and time, respectively. The remaining colomns contain the
C minimum and maximium values for several variables over all time
C for each vector. The total number of columns is M = 2 + 2K, where K
C is the number of variables for which summarize has reported on.
C The total number of rows is N = 3 + L, where L is the number of vectors
C
C The format of the summarize input file is
C
C >vector,time var1_mx var1_mn ......... varK_mx varK_mn
C > [G],[G]......[G]
C > ,,,......,,
C > vec1 time vec1_var1_mx  vec1_var1_mn ........vec1_varK_mx vec1_varK_mn
C > vec2 time vec2_var1_mx  vec2_var1_mn ........vec2_varK_mx vec2_varK_mn
C > .
C > .
C > .
C > vecL time vecL_var1_mx  vecL_var1_mn ........vecL_varK_mx vecL_varK_mn
C
C The main functions of the MMOV code are:
C
C  1) Extract the column headings from the first row of file header
C
C  2) Read the data set into an array
C
C  3) Find the min or max of each column (i.e. min/max for each variable
C     all vectors in the data set).
C
C  4) Report the range for each variable
C
C-----------------------------------------------------------------------
C
      PROGRAM MMOV
C
C
C     nhead - number of rows in header
C     nrow  - number of rows in the file (3 + number of vectors)
C     ncol  - number of columns in the file ( 2 + 2*number of variables)
C
C
C
```

```
      PARAMETER (nfiles = 3)
      PARAMETER (nhead = 3)
      PARAMETER (nrow = 100)
      PARAMETER (ncol = 58

      CHARACTER*80 filenm(nfiles)
      CHARACTER*80 chead(ncol), tmp, varnm
      CHARACTER*80 rdfmt
      CHARACTER*80 fnminp, fnmout, fnmdbg
      CHARACTER*1024 head(nhead), tmphead

      INTEGER ierr, idebug
      INTEGER iunscr, iuninp, iunout, iundbg
      INTEGER nfiler
      DOUBLE PRECISION, ALLOCATABLE :: sumdat(:,:)
      DOUBLE PRECISION, ALLOCATABLE :: exvalue(:)


C-----------------------------
C.....QA Setup
C-----------------------------

      CALL QASETUP('MMOV',
     +             'X-1.00',
     +             '09/02/04',
     +             'J. F. Kanney',
     +             'Nobody'
     +             )

C-----------------------------
C.....Assign i/o unit numbers
C-----------------------------

      iunscr = 6
      iuninp = 11
      iunout = 12
      iundbg = 13

C--------------------
C.....Allocate memory
C--------------------

      WRITE(iunscr,*) 'MMOV >> Allocating memory'

      ALLOCATE( sumdat(1:nrow,1:ncol), STAT=ierr )

      IF ( ierr .NE. 0 ) THEN
         WRITE(iunscr,*) 'Error allocating memory'
         CALL QAABORT ('Error allocating memory')
      ENDIF

      ALLOCATE( exvalue(1:ncol), STAT=ierr )

      IF ( ierr .NE. 0 ) THEN
         WRITE(iunscr,*) 'Error allocating memory'
         CALL QAABORT ('Error allocating memory')
      ENDIF


C-------------------------------------------
C.....Process command line (get file names)
```

```
C-----------------------------------------

      WRITE(iunscr,*) 'MMOV >> Processing command line'


C.....Required args  (1-3) are fnminp, fnmout, fnmdbg


      CALL filcmdlin( nfiles, nfiler, filenm )

C      write(iunscr,*) 'nfiler = ', nfiler

      IF ( nfiler .GT. nfiles ) THEN
         CALL QAABORT( 'MMOV >> Too many command line arguments')
      ELSE
         IF ( nfiler .LT. nfiles-1 ) THEN
            CALL QAABORT( 'MMOV >> Too few command line arguments')
         ENDIF
      ENDIF


C.....Assign file names read from command line. If no debug file
C     has been specified, redirect all diagnostic output to output file


      fnminp = filenm(1)
      fnmout = filenm(2)

      IF ( nfiler .EQ. nfiles) THEN
        idebug = 1
        fnmdbg = filenm(3)
      ELSE
        idebug = 0
        iundbg = iunout
        fnmdbg = 'NULL'
      ENDIF


C.....Echo files names

      WRITE(iunscr,20) fnminp, fnmout, fnmdbg

 20   FORMAT (1X, 'MMOV >> input file              = ',A80/
     +        1X, 'MMOV >> output file             = ',A80/
     +        1X, 'MMOV >> diagnostic/debug file   = ',A80/
     +       )


C-----------------------------------------
C.....Open files
C-----------------------------------------

      WRITE(iunscr,*) 'MMOV >> Opening files'

C.....Open input summarize data file

      OPEN (UNIT=iuninp, FILE=fnminp, STATUS='OLD',IOSTAT=ierr)
      IF ( ierr .NE. 0 ) THEN
         CALL QAABORT ('Error opening input file')
      ENDIF
```

```
C.....Open output file

      OPEN (UNIT=iunout, FILE=fnmout, STATUS='UNKNOWN', RECL=ncol*200,
     +      IOSTAT=ierr)
      IF ( ierr .NE. 0 ) THEN
         CALL QAABORT ('Error opening output file')
      ENDIF

C.....Open diagnostics/debug file

      IF ( idebug .NE. 0 ) THEN
        OPEN (UNIT=iundbg, FILE=fnmdbg, STATUS='UNKNOWN', RECL=ncol*200,
     +        IOSTAT=ierr)
        IF ( ierr .NE. 0 ) THEN
           CALL QAABORT ('Error opening debug file')
        ENDIF
      ENDIF

C.....Write Header to output file and debug file

      CALL QAPAGE(iunout, ' ')
      CALL QAPAGE(iundbg, ' ')

C.....Echo file names to debug file

      WRITE(iundbg,20) fnminp, fnmout, fnmdbg
      WRITE(iunout,20) fnminp, fnmout, fnmdbg

C-------------------------
C.....Read input file
C-------------------------

      WRITE(iunscr,*) 'MMOV >> Reading input file'
      WRITE(iundbg,*) 'MMOV >> Reading input file'

C.....Read in header lines

      DO i=1,nhead
        READ (iuninp,'(A)') head(i)
      END DO


C.....Read in data

      rdfmt = '(58e13.6,1x))'
      DO i=1,nrow
         READ(iuninp,rdfmt) (sumdat(i,j),j=1,ncol)
      END DO


C.....Echo input data to dubug file

      WRITE (iundbg,*) 'MMOV >> Begin Input Echo'


      DO i=1,nhead
        WRITE (iundbg,'(A)') head(i)
      END DO


      DO i=1,nrow
```

```
         WRITE(iundbg,rdfmt) (sumdat(i,j),j=1,ncol)
      END DO

      WRITE (iundbg,*) 'MMOV >> End Input Echo'


C.....Close input file

      WRITE (iundbg,*) 'MMOV >> Close input file'
      CLOSE (UNIT=iuninp,STATUS='KEEP')


C------------------------
C.....Process input data
C------------------------

      WRITE(iunscr,*) 'MMOV >> Processing input data'
      WRITE(iundbg,*) 'MMOV >> Processing input data'

C.....Extract vector column heading

      ichead = 1
      idxl = 0
      tmphead = head(1)
      idxr = index(tmphead,',')
      chead(ichead) = tmphead(idxl:idxr-1)

C.....Extract time column heading

      ichead = 2
      idxl = idxr
      idxr = idxl + index( tmphead(idxl+1:),' ')
      chead(ichead) = tmphead(idxl:idxr-1)

C.....Extract variable column headings

      DO i=3, ncol
        ichead = ichead+1
        idxl = idxr
        idxr = idxl + 9
        chead(ichead) = tmphead(idxl:idxr-1)
      END DO

      DO j=1,ncol
         WRITE(iundbg,10) j, chead(j)
      END DO

   10 FORMAT(' Column ',i3,' = ', A10)


C.....Find min and max for variables
C.....Find max over all vectors for odd number colums
C.....Find min over all vectors for even number colums


      DO j=3, ncol

        IF ( MOD(j,2) .EQ. 1 ) THEN

            exvalue(j) = 0.D0
            DO i=1,nrow
```

```
            IF ( sumdat (i,j) .GT. exvalue(j) ) THEN
               exvalue(j) = sumdat(i,j)
            ENDIF
          END DO

          WRITE(iundbg,100) chead(j), exvalue(j)
          WRITE(iunout,100) chead(j), exvalue(j)

        ELSE

           exvalue(j) = 1.D+300

           DO i=1,nrow
            IF ( sumdat (i,j) .LT. exvalue(j) ) THEN
               exvalue(j) = sumdat(i,j)
            ENDIF
          END DO

          WRITE(iundbg,200) chead(j), exvalue(j)
          WRITE(iunout,200) chead(j), exvalue(j)

        ENDIF

      END DO

  100 FORMAT(' Max of ', A10, ' = ', e13.6)
  200 FORMAT(' Min of ', A10, ' = ', e13.6)

C.....Report Ranges

      WRITE(iunout,'(A)') ' '
      WRITE(iundbg,'(A)') ' '

      DO k = 3,ncol-1,2
         j1 = k
         j2 = k+1
         idxr = INDEX(chead(j1),'MX')
         tmp = chead(j1)
         varnm = tmp(1:idxr-1)
         vmax = exvalue(j1)
         vmin = exvalue(j2)
         write(iunout,300) varnm, vmin, vmax
         write(iundbg,300) varnm, vmin, vmax
      END DO

  300 FORMAT(A10, 'range = [',e13.6, ',', e13.6,']' )

C-------------
C.....Clean up
C-------------

      WRITE(iunscr,*) 'MMOV >> Cleaning up'
      WRITE(iundbg,*) 'MMOV >> Cleaning up'

C.....Release allocated memory

      WRITE (iundbg,*) 'MMOV >> Deallocate Memory'

      DEALLOCATE(exvalue)
      DEALLOCATE(sumdat)
```

```
C.....Close output and debug files

      WRITE (iundbg,*) 'MMOV >> Close output file'

      CLOSE (UNIT=iunout,STATUS='KEEP')

      IF ( idebug .NE. 0 ) THEN
        WRITE (iundbg,*) 'MMOV >> Close debug file'
        CLOSE (UNIT=iundbg,STATUS='KEEP')
      ENDIF

C.....Signal normal completion

      STOP 'MMOV >> Normal Completion'
      END
```

The MMOV code can be built using the following commands:

```
$ for mmov.f
$ link /exe=mmov.exe mmov.obj camdat_lib/lib camcon_lib/lib camsupes_lib/lib
```

The SUMMARIZE tables can be processed with the MMOV code using the following series of commands (replace "disk:[dir_spec]" with one's current working directory):

```
$
$ mmov_exe :== "$disk:[dir_spec]mmov.exe"
$
$ input_file  :== "sum_bfvol_minmax_cra1_r1_s1.tbl"
$ output_file :== "mmov_bfvol_minmax_cra1_r1_s1.out"
$ debug_file  :== "mmov_bfvol_minmax_cra1_r1_s1.dbg"
$
$ mmov_exe 'input_file  'output_file 'debug_file
$
$
$ input_file  :== "sum_bfvol_minmax_cra1_r1_s2.tbl"
$ output_file :== "mmov_bfvol_minmax_cra1_r1_s2.out"
$ debug_file  :== "mmov_bfvol_minmax_cra1_r1_s2.dbg"
$
$ mmov_exe 'input_file  'output_file 'debug_file
```

The MMOV output files for scenarios S1 and S2 are shown below:

MMOV_BFVOL_MINMAX_CRA1_R1_S1.OUT

```
 MMOV X-1.00 09/02/2004                                      09/02/2004 14:44:31

 MMOV >> input file              =
PAA:[SHARED.JFKANNE.WRK.CRA_RESP.CACO3_INP.TBL]SUM_BFVOL_MINMAX_CRA1_R1_S1.TBL
 MMOV >> output file             =
PAA:[SHARED.JFKANNE.WRK.CRA_RESP.CACO3_INP]MMOV_BFVOL_MINMAX_CRA1_R1_S1.OUT
 MMOV >> diagnostic/debug file   =
PAA:[SHARED.JFKANNE.WRK.CRA_RESP.CACO3_INP]MMOV_BFVOL_MINMAX_CRA1_R1_S1.DBG

 Max of  VB_UD4MX   =  0.132888E+04
 Min of  VB_UD4MN   =  0.218839E+01
 Max of  VB_UD3MX   =  0.132343E+04
 Min of  VB_UD3MN   =  0.234789E+01
 Max of  VB_UD2MX   =  0.736918E+02
 Min of  VB_UD2MN   =  0.154502E+00
 Max of  VB_UD1MX   =  0.704318E+03
 Min of  VB_UD1MN   =  0.152097E+01
 Max of  VB_LD1MX   =  0.335029E+03
 Min of  VB_LD1MN   =  0.114607E+01
 Max of  VB_LD2MX   =  0.343018E+03
 Min of  VB_LD2MN   =  0.146084E+01
 Max of  VB_LD3MX   =  0.422549E+03
 Min of  VB_LD3MN   =  0.370256E+01
 Max of  VB_WPMX    =  0.554610E+04
 Min of  VB_WPMN    =  0.276150E-03
 Max of  VB_TOTMX   =  0.665778E+04
 Min of  VB_TOTMN   =  0.378661E+02
 Max of  VR_UD4MX   =  0.527560E+05
 Min of  VR_UD4MN   =  0.503922E+05
 Max of  VR_UD3MX   =  0.527562E+05
 Min of  VR_UD3MN   =  0.504319E+05
 Max of  VR_UD2MX   =  0.314441E+04
 Min of  VR_UD2MN   =  0.300711E+04
 Max of  VR_UD1MX   =  0.305125E+05
 Min of  VR_UD1MN   =  0.291872E+05
 Max of  VR_LD1MX   =  0.803577E+04
 Min of  VR_LD1MN   =  0.769375E+04
 Max of  VR_LD2MX   =  0.803577E+04
 Min of  VR_LD2MN   =  0.769459E+04
 Max of  VR_LD3MX   =  0.989913E+04
 Min of  VR_LD3MN   =  0.948001E+04
 Max of  VR_WPMX    =  0.427231E+05
 Min of  VR_WPMN    =  0.699648E+04
 Max of  VRTOT1MX   =  0.206525E+06
 Min of  VRTOT1MN   =  0.168041E+06
 Max of  VRTOT2MX   =  0.165140E+06
 Min of  VRTOT2MN   =  0.157887E+06
 Max of  BR_UD4MX   =  0.258299E-01
 Min of  BR_UD4MN   =  0.414911E-04
 Max of  BR_UD3MX   =  0.257240E-01
 Min of  BR_UD3MN   =  0.445208E-04
 Max of  BR_UD2MX   =  0.240319E-01
 Min of  BR_UD2MN   =  0.491465E-04
 Max of  BR_UD1MX   =  0.236700E-01
 Min of  BR_UD1MN   =  0.498589E-04
 Max of  BR_LD1MX   =  0.434895E-01
 Min of  BR_LD1MN   =  0.143345E-03
```

```
Max of  BR_LD2MX  =  0.445723E-01
Min of  BR_LD2MN  =  0.183355E-03
Max of  BR_LD3MX  =  0.445714E-01
Min of  BR_LD3MN  =  0.374028E-03
Max of  BRTOT1MX  =  0.330052E-01
Min of  BRTOT1MN  =  0.188726E-03
Max of  BRTOT2MX  =  0.412673E-01
Min of  BRTOT2MN  =  0.230489E-03

VB_UD4    range = [ 0.218840E+01, 0.132888E+04]
VB_UD3    range = [ 0.234789E+01, 0.132343E+04]
VB_UD2    range = [ 0.154501E+00, 0.736918E+02]
VB_UD1    range = [ 0.152097E+01, 0.704318E+03]
VB_LD1    range = [ 0.114607E+01, 0.335030E+03]
VB_LD2    range = [ 0.146084E+01, 0.343018E+03]
VB_LD3    range = [ 0.370256E+01, 0.422549E+03]
VB_WP     range = [ 0.276150E-03, 0.554610E+04]
VB_TOT    range = [ 0.378661E+02, 0.665779E+04]
VR_UD4    range = [ 0.503922E+05, 0.527560E+05]
VR_UD3    range = [ 0.504319E+05, 0.527562E+05]
VR_UD2    range = [ 0.300711E+04, 0.314442E+04]
VR_UD1    range = [ 0.291872E+05, 0.305125E+05]
VR_LD1    range = [ 0.769375E+04, 0.803577E+04]
VR_LD2    range = [ 0.769459E+04, 0.803577E+04]
VR_LD3    range = [ 0.948001E+04, 0.989913E+04]
VR_WP     range = [ 0.699648E+04, 0.427231E+05]
VRTOT1    range = [ 0.168041E+06, 0.206525E+06]
VRTOT2    range = [ 0.157887E+06, 0.165140E+06]
BR_UD4    range = [ 0.414911E-04, 0.258299E-01]
BR_UD3    range = [ 0.445208E-04, 0.257240E-01]
BR_UD2    range = [ 0.491465E-04, 0.240319E-01]
BR_UD1    range = [ 0.498589E-04, 0.236700E-01]
BR_LD1    range = [ 0.143346E-03, 0.434895E-01]
BR_LD2    range = [ 0.183355E-03, 0.445723E-01]
BR_LD3    range = [ 0.374028E-03, 0.445714E-01]
BRTOT1    range = [ 0.188726E-03, 0.330052E-01]
BRTOT2    range = [ 0.230489E-03, 0.412673E-01]
```

MMOV_BFVOL_MINMAX_CRA1_R1_S2.OUT

```
 MMOV X-1.00 09/02/2004                                        09/02/2004 14:44:37

 MMOV >> input file             =
PAA:[SHARED.JFKANNE.WRK.CRA_RESP.CACO3_INP.TBL]SUM_BFVOL_MINMAX_CRA1_R1_S2.TBL
 MMOV >> output file            =
PAA:[SHARED.JFKANNE.WRK.CRA_RESP.CACO3_INP]MMOV_BFVOL_MINMAX_CRA1_R1_S2.OUT
 MMOV >> diagnostic/debug file  =
PAA:[SHARED.JFKANNE.WRK.CRA_RESP.CACO3_INP]MMOV_BFVOL_MINMAX_CRA1_R1_S2.DBG

 Max of  VB_UD4MX  =  0.165492E+04
 Min of  VB_UD4MN  =  0.218839E+01
 Max of  VB_UD3MX  =  0.172251E+04
 Min of  VB_UD3MN  =  0.234789E+01
 Max of  VB_UD2MX  =  0.109475E+03
 Min of  VB_UD2MN  =  0.154502E+00
 Max of  VB_UD1MX  =  0.107291E+04
 Min of  VB_UD1MN  =  0.152097E+01
 Max of  VB_LD1MX  =  0.311888E+03
 Min of  VB_LD1MN  =  0.134507E+01
 Max of  VB_LD2MX  =  0.311935E+03
 Min of  VB_LD2MN  =  0.208004E+01
 Max of  VB_LD3MX  =  0.384264E+03
 Min of  VB_LD3MN  =  0.370256E+01
 Max of  VB_WPMX   =  0.135374E+05
 Min of  VB_WPMN   =  0.124280E-01
 Max of  VB_TOTMX  =  0.163465E+05
 Min of  VB_TOTMN  =  0.503695E+02
 Max of  VR_UD4MX  =  0.527560E+05
 Min of  VR_UD4MN  =  0.506848E+05
 Max of  VR_UD3MX  =  0.527562E+05
 Min of  VR_UD3MN  =  0.507004E+05
 Max of  VR_UD2MX  =  0.314441E+04
 Min of  VR_UD2MN  =  0.302236E+04
 Max of  VR_UD1MX  =  0.305125E+05
 Min of  VR_UD1MN  =  0.293309E+05
 Max of  VR_LD1MX  =  0.803577E+04
 Min of  VR_LD1MN  =  0.772683E+04
 Max of  VR_LD2MX  =  0.803577E+04
 Min of  VR_LD2MN  =  0.772683E+04
 Max of  VR_LD3MX  =  0.989913E+04
 Min of  VR_LD3MN  =  0.951857E+04
 Max of  VR_WPMX   =  0.432778E+05
 Min of  VR_WPMN   =  0.699648E+04
 Max of  VRTOT1MX  =  0.207046E+06
 Min of  VRTOT1MN  =  0.168041E+06
 Max of  VRTOT2MX  =  0.165140E+06
 Min of  VRTOT2MN  =  0.158711E+06
 Max of  BR_UD4MX  =  0.324083E-01
 Min of  BR_UD4MN  =  0.414911E-04
 Max of  BR_UD3MX  =  0.337455E-01
 Min of  BR_UD3MN  =  0.445208E-04
 Max of  BR_UD2MX  =  0.360591E-01
 Min of  BR_UD2MN  =  0.491465E-04
 Max of  BR_UD1MX  =  0.364322E-01
 Min of  BR_UD1MN  =  0.498589E-04
 Max of  BR_LD1MX  =  0.403642E-01
 Min of  BR_LD1MN  =  0.167981E-03
```

```
Max of  BR_LD2MX  =  0.403704E-01
Min of  BR_LD2MN  =  0.259574E-03
Max of  BR_LD3MX  =  0.403700E-01
Min of  BR_LD3MN  =  0.374028E-03
Max of  BRTOT1MX  =  0.844281E-01
Min of  BRTOT1MN  =  0.255789E-03
Max of  BRTOT2MX  =  0.102074E+00
Min of  BRTOT2MN  =  0.306207E-03

VB_UD4    range = [ 0.218840E+01, 0.165492E+04]
VB_UD3    range = [ 0.234789E+01, 0.172251E+04]
VB_UD2    range = [ 0.154501E+00, 0.109475E+03]
VB_UD1    range = [ 0.152097E+01, 0.107291E+04]
VB_LD1    range = [ 0.134507E+01, 0.311888E+03]
VB_LD2    range = [ 0.208004E+01, 0.311935E+03]
VB_LD3    range = [ 0.370256E+01, 0.384264E+03]
VB_WP     range = [ 0.124280E-01, 0.135374E+05]
VB_TOT    range = [ 0.503695E+02, 0.163465E+05]
VR_UD4    range = [ 0.506848E+05, 0.527560E+05]
VR_UD3    range = [ 0.507004E+05, 0.527562E+05]
VR_UD2    range = [ 0.302236E+04, 0.314442E+04]
VR_UD1    range = [ 0.293309E+05, 0.305125E+05]
VR_LD1    range = [ 0.772683E+04, 0.803577E+04]
VR_LD2    range = [ 0.772683E+04, 0.803577E+04]
VR_LD3    range = [ 0.951857E+04, 0.989913E+04]
VR_WP     range = [ 0.699648E+04, 0.432779E+05]
VRTOT1    range = [ 0.168041E+06, 0.207046E+06]
VRTOT2    range = [ 0.158711E+06, 0.165140E+06]
BR_UD4    range = [ 0.414911E-04, 0.324083E-01]
BR_UD3    range = [ 0.445208E-04, 0.337455E-01]
BR_UD2    range = [ 0.491465E-04, 0.360591E-01]
BR_UD1    range = [ 0.498589E-04, 0.364322E-01]
BR_LD1    range = [ 0.167981E-03, 0.403643E-01]
BR_LD2    range = [ 0.259574E-03, 0.403704E-01]
BR_LD3    range = [ 0.374028E-03, 0.403700E-01]
BRTOT1    range = [ 0.255789E-03, 0.844281E-01]
BRTOT2    range = [ 0.306207E-03, 0.102074E+00]
```