

## DMR Initial Product (IP) Documentation

### 1. Introduction

The Differential Microwave Radiometers (DMR) experiment, one of the three scientific experiments aboard the COsmic Background Explorer (COBE, see [1]), was designed to obtain definitive measurements of the anisotropy of the cosmic microwave background (CMB) radiation.

The Initial Product Data Sets include six DMR full sky maps at 31.5, 53, and 90 GHz with two independent channels at each frequency, and in addition differential data sorted by sky position to provide sky sampling information. The sky maps are in Ecliptic coordinates. Products for four of the channels (53 and 90 GHz, channel A and B) consist of data obtained during the first full year of the operation (1989 day 356 to 1990 day 355). Products for the other two channels (31 GHz, channel A and B) consist of data obtained during the same time period excluding the eclipse season (1990 day 141 to 1990 day 206). Here the eclipse season refers to the period when COBE moves through the Earth's shadow as the spacecraft passes over the South Pole. The data sets are in the Flexible Image Transport System (FITS) format.

### 2. DMR Instrument Description

#### A. The Differential Microwave Radiometers

The DMR instrument was designed to map the CMB intensity on large angular scales. The instrument is described in detail in [2]. The instrument consists of six differential microwave radiometers, two independent radiometers which operate at each of three frequencies: 31.5, 53, and 90 GHz. These frequencies were chosen to be near the minimum of the Galactic emission spectrum and near the CMB peak. The multiple frequency measurements provide the ability to separate Galactic emission from cosmic emission based on spectral shape and to measure the spectrum of detected anisotropies.

The DMR instrument measures the difference in power received from two directions separated by 60 degrees on the sky. Each horn antenna has a field of view which is approximately a seven degree FWHM Gaussian (see [3]). The COBE Satellite is in a 900 km altitude, near-polar (99 degree inclination) orbit which precesses to follow the terminator. Over six months the combination of the 0.8-rpm spacecraft spin, 103-minute orbit, and 1 degree-per-day orbital precession enable DMR antennae to make highly-redundant observations of the entire sky. Noise level per measurement (0.5 sec being the duration of a measurement) for each DMR channel can be found in Table 2.1.

Table 2.1 Noise Level (per 0.5 sec measurement) for Each DMR Channel

Channel	31A	31B	53A	53B	90A	90B
Nominal RMS (mK)	55.0	58.3	22.5	24.9	40.3	28.3

### B. Calibration

Three techniques are used to assure the accurate calibration of the DMR instrument. First, the signal difference between beam-filling hot and cold targets observed on the ground provides a primary calibration that is transferred to space by noise sources internal to the instrument. Second, the Moon is used in flight as an external calibration source. Third, the signal arising from the Doppler effect due to the Earth's motion around the barycenter of the solar system is used as an external calibration source. Preliminary analysis of the external source calibration techniques confirms the accuracy of the currently more precise ground-based calibration. A detailed description of various calibration techniques can be found in [4].

## 3. Data Reduction

### A. Overview

COBE data are digitized and stored in an on-board tape recorder before being telemetered daily to a ground station. A pre-processor strips the DMR data from the telemetry stream, merges the raw DMR data with spacecraft attitude and orbit information, and checks the quality of the data, flagging data known or suspected to be unusable. A slowly-varying baseline is removed and the data are calibrated. The calibrated data are then corrected for various instrumental and systematic effects and used to make sky maps.

### B. Processing Leading to Initial Products

We combine the original DMR telemetry data with spacecraft attitude and orbit information to create a Time-Ordered Data Set. The data are then scanned for various sources of contamination and data quality flags are set. We reject data of questionable telemetry quality, data without accompanying attitude information, or with the instrument science telemetry disabled. We also reject spikes and transients in the data, disregarding all points that lie more than five times the RMS scatter from the daily mean.

An instrumental offset is modeled using a slowly-varying baseline fit to the uncalibrated data and then subtracted. Three different fitting functions, a cubic spline, a Legendre polynomial, and a running mean, were used and the results were checked for consistency. Then a calibration factor is applied to convert the data from digitized telemetry units to antenna temperature.

The calibrated data were corrected for various known effects: the

satellite's motion around the Earth (7.4 km/s); the Earth's motion around the solar system barycenter (30 km/s); the magnetic susceptibility of the instrument; emission from the moon picked up in the antenna sidelobes and emission from Jupiter. For a detailed account of the systematic error corrections see [5] and [6].

The calibrated, corrected data are fitted to sky maps with 6144 pixels (2.8 degree angular size) whose independent antenna temperatures are chosen to minimize the Chi-squared sum,

$$\text{Chi-squared} = \text{SUM} (T_i - T_j - DT_{ij})^2 / (\text{sigma}_{ij})^2$$

where the sum is over all observations,  $T_i$  and  $T_j$  are the map temperatures at pixels  $i$  and  $j$ , and  $DT_{ij}$  and  $\text{sigma}_{ij}$  are the measured temperature differences between pixels  $i$  and  $j$  and their associated errors respectively. Since the antenna beam width is larger than the pixel size, there is some correlation between neighboring pixels for any real signal. Further details of the data processing algorithms are described in [7] and [8].

#### 4. Initial Products

##### A. DMR Pixelized Differential Data

The corrected, calibrated time-ordered data have been sorted according to the pixel number seen by horn 1, and then by the pixel number seen by horn 2. All differential measurements for a given pixel pair are then combined into four statistics: the sums of the first to fourth powers of the differential temperatures. These sums, the horn 1 and horn 2 pixel numbers, and the number of occurrences of each pixel permutation comprise the DMR pixelized differential data. For each channel, there are over a million pixel permutations. The data record format for each pixel permutation is given in Table 4.1.

A simple program PIXPRM, listed in Appendix 1, can be used to read the FITS binary table from the DMR pixelized Initial Product. This FORTRAN example uses the FITSIO library and should run, as written, on all platforms on which the library is supported. FITSIO was written by William Pence of NASA GSFC HEASARC, and is available by anonymous FTP from [tetra.gsfc.nasa.gov](http://tetra.gsfc.nasa.gov). For illustrative purposes, the example calculates some simple statistics from the data sets; users may wish to substitute their own code to perform the calculations in which they are interested.

Table 4.1 DMR Pixelized Differential Data Set

```

=====
Data field          Description
-----
PIX_PLUS           Pixel number (0 to 6143) for horn 1
PIX_MINU           Pixel number (0 to 6143) for horn 2
N_OBS              Number of observations
SUM                SUM (differential temperatures) (mK)
SUM_2              SUM (differential temperatures)^2 (mK^2)
SUM_3              SUM (differential temperatures)^3 (mK^3)
SUM_4              SUM (differential temperatures)^4 (mK^4)
=====

```

B. DMR Sky Maps

A sky temperature for each pixel on the sky maps is derived by solving a linear system of equations to the DMR observations by minimizing the chi-squared, while simultaneously fitting the coefficients of systematic error models. The resulting sky maps represent temperature differences; chi-squared is unchanged by the addition of an arbitrary constant to the map. The data record format for each pixel of the sky map is given in Table 4.2. The maximum, mean, and minimum sky map coverages and the corresponding pixel-to-pixel RMS noise levels are given in Table 4.3. More information about the type of baseline used and various systematic error corrections applied can be found in the FITS header.

A simple program SKYMAP, listed in Appendix 2, can be used to read the FITS binary table from the DMR sky map Initial Product. This FORTRAN program also uses the FITSIO library as described in 4.A.

Table 4.2 DMR Sky Maps Data Set

```

=====
Data field          Description
-----
PIXEL              Pixel number (0 to 6143)
SIGNAL             Sky antenna temperature (mK)
N_OBS              Number of observations
SERROR            Estimate of uncertainty as RMS/SQRT(N_OBS) (mK)
ECLON              Ecliptic longitude (degrees)
ECLAT              Ecliptic latitude (degrees)
GALON              Galactic longitude (degrees)
GALAT              Galactic latitude (degrees)
RA                 Right ascension (degrees)
DEC                Declination (degrees)
=====

```

Table 4.3 DMR Sky Map Coverage and Sensitivity Limits

DMR Channel	31A	31B	53A	53B	90A	90B
Max obs/pixel	36178	35720	41063	41218	40711	40713
Noise level (mK)	0.289	0.309	0.111	0.123	0.200	0.140
Mean obs/pixel	14612	14076	17433	17434	17337	17325
Noise level (mK)	0.455	0.492	0.170	0.223	0.306	0.215
Min obs/pixel	5306	4205	8881	8896	8825	8853
Noise level (mK)	0.755	0.900	0.238	0.263	0.429	0.300

References:

- [1] The COBE Mission: Its Design and Performance Two Years After Launch, N.W. Boggess et al. 1992, APJ,420
- [2] COBE Differential Microwave Radiometers: Instrument Design and Implementation, G.F.Smoot et al. 1990, APJ,360,685
- [3] Measurements of Very Low-sidelobe Conical Horn Antennas, Marco A. Toral et al. 1989, IEEE Transactions on Antennas and Propagation, Vol 37, No.2.
- [4] COBE Differential Microwave Radiometers(DMR): Calibration Techniques, C.L.Bennett et al. 1992, APJ,391
- [5] Daily Quality Assurance Software for a Satellite Radiometer System, P.B. Keegstra et al., Proceedings of the 1st Astronomical Data Analysis and Software Systems Conference, PASP Conference Series, 25(1991): 3 p.
- [6] COBE Differential Microwave Radiometers (DMR): Preliminary Systematic Error Analysis, A. Kogut et al. 1992, APJ,401
- [7] COBE Differential Microwave Radimeter (DMR) Data Processing Techniques, P.D. Jackson et al., Proceedings of the 1st Astronomical Data Analysis and Software Systems Conference, PASP Conference Series, 25(1991): 3 p.
- [8] Mapping the Sky with the COBE Differential Microwave Radiometers, M.A. Janssen and S. Gulkis, Proceedings of Infrared and Submillimetre Sky After COBE, 1992.

Appendix 1. FORTRAN source code to read the DMR pixelization file

```
program pixprm

C      A simple program to read the FITS binary table from the DMR
C      pixelized Initial Product.

integer maxdim
parameter (maxdim = 20)

character*30 errtxt, extnam,
&           ttype(maxdim),
&           tform(maxdim),
&           tunit(maxdim)

logical simple,extend,anyflg
integer iunit,status,bitpix,naxis,naxes(maxdim),pcount,gcount
integer group,fpixel,nelem,nrows
integer i,j,nobs,nperm,npoint
integer tfield,rwstat,bksize,vardat
integer colnum,frow,felem
integer hdutyp,inull
real    sum,sum2,enull
doubleprecision sumsum,sum_sq

status=0
iunit=15
C      open the existing FITS file with readonly access
      rwstat=0
      call ftopen(iunit,'dmr_pixperm_31a.fits',rwstat,bksize,status)
      if (status .ne. 0) goto 99

C      read the required primary array keywords
      call ftghpr(iunit,maxdim,simple,bitpix,naxis,naxes,
&               pcount,gcount,extend,status)
      if (status .ne. 0) goto 99

C      now move to the binary table extension
      call ftmahd(iunit,2,hdutyp,status)
      if (status .ne. 0) goto 99

C      get the binary table parameters
      call ftghbn(iunit,maxdim,nrows,tfield,ttype,tform,tunit,
&               extnam,vardat,status)
      if (status .ne. 0) goto 99

C      test that this is a DMR pixperm dataset, and its fields
C      are as expected.
      if (extnam .ne. 'DMR_PIXPERM' .or.
&        ttype(4) .ne. 'SUM' .or.
&        ttype(5) .ne. 'SUM_2' .or.
&        ttype(3) .ne. 'N_OBS') then
```

```

        print *, 'extnam   = ', extnam
        print *, 'ttype(4) = ', ttype(4)
        print *, 'ttype(5) = ', ttype(5)
        print *, 'ttype(3) = ', ttype(3)
        stop 'Not a DMR pixperm dataset.'
    endif

C      initialize statistical accumulators
    nperm = 0
    npoint = 0
    sumsum = 0.0d0
    sum_sq = 0.0d0
C      read each pix-perm and process
    do 10 frow = 1, nrows
        felem=1
        nelelem=1
        inull=0
        colnum=4
        call ftgcve(iunit,colnum,frow,felem,nelelem,enull,sum,
&                  anyflg,status)
        if (status .ne. 0) goto 99
        colnum=5
        call ftgcve(iunit,colnum,frow,felem,nelelem,enull,sum2,
&                  anyflg,status)
        if (status .ne. 0) goto 99
        colnum=3
        call ftgcvj(iunit,colnum,frow,felem,nelelem,inull,nobs,
&                  anyflg,status)
        if (status .ne. 0) goto 99

        if (nobs .gt. 0) then
            nperm = nperm + 1
            npoint = npoint + nobs
            sumsum = sumsum + sum
            sum_sq = sum_sq + sum2
        endif
    10 continue

C      print out statistics
    print 1000, nperm, npoint, sumsum/dbble(npoint),
&          sqrt(sum_sq/dbble(npoint))
1000    format(' Pix-perms read in = ',i9/
&          ' Total observations = ',i9/
&          '           Data mean = ',2x,f14.6/
&          '           Data RMS = ',2x,f14.6)

C      now close the table and quit
    call ftclos(iunit,status)

99     if (status .le. 0)then
        print *, '*** Program completed successfully ***'
    else
C      get the error text description

```



```
    call ftgerr(status,errtxt)
    print *,'*** ERROR - program did not run successfully ***'
    print *,'status =',status,': ',errtxt
end if
```

```
100  continue
     end
```

Appendix 2. FORTRAN source code to read the DMR sky maps

```
program skymap

C      A simple program to read the FITS binary table from the DMR
C      skymap Initial Product.

integer maxdim
parameter (maxdim = 20)

character*30 errtxt, extnam,
&           ttype(maxdim),
&           tform(maxdim),
&           tunit(maxdim)

logical simple,extend,anyflg
integer iunit,status,bitpix,naxis,naxes(maxdim),pcount,gcount
integer group,fpixel,nelem,nrows
integer i,j,nobs,npseen,tfield,rwstat,bksize,vardat
integer colnum,frow,felem
integer hdu typ,inull
real    temp,enull,tmax,tmin
doubleprecision sum,sum_sq

status=0
iunit=15
C      open the existing FITS file with readonly access
      rwstat=0
      call ftopen(iunit,'dmr_skymap_31a.fits',rwstat,bksize,status)
      if (status .ne. 0) goto 99

C      read the required primary array keywords
      call ftghpr(iunit,maxdim,simple,bitpix,naxis,naxes,
&               pcount,gcount,extend,status)
      if (status .ne. 0) goto 99

C      now move to the binary table extension
      call ftmahd(iunit,2,hdu typ,status)
      if (status .ne. 0) goto 99

C      get the binary table parameters
      call ftghbn(iunit,maxdim,nrows,tfield,ttype,tform,tunit,
&               extnam,vardat,status)
      if (status .ne. 0) goto 99

C      test that this is a DMR skymap dataset, and its fields
C      are as expected.
      if (extnam .ne. 'DMR_SKYMAP' .or.
&         ttype(2) .ne. 'SIGNAL' .or.
&         ttype(3) .ne. 'N_OBS') then
          print *, 'extnam = ', extnam
          print *, 'ttype(2) = ', ttype(2)
```

```

        print *, 'ttype(3) = ', ttype(3)
        stop 'Not a DMR skymap.'
    endif

C    initialize statistical accumulators
    npseen = 0
    sum     = 0.0d0
    sum_sq  = 0.0d0
    tmax    = -1e30
    tmin    = 1e30
C    read each pixel and process
    do 10 frow = 1, nrows
        felem=1
        nelelem=1
        inull=0
        colnum=2
        call ftgcve(iunit,colnum,frow,felem,nelem,enull,temp,
&                  anyflg,status)
        if (status .ne. 0) goto 99
        colnum=3
        call ftgcvj(iunit,colnum,frow,felem,nelem,inull,nobs,
&                  anyflg,status)
        if (status .ne. 0) goto 99

        if (nobs .gt. 0) then
            npseen = npseen + 1
            sum     = sum + temp
            sum_sq  = sum_sq + temp*temp
            tmax    = max(tmax,temp)
            tmin    = min(tmin,temp)
        endif
    10 continue

C    print out statistics
    print 1000, npseen,
&          tmax,
&          tmin,
&          sum/dbble(npseen),
&          sqrt( (dbble(npseen)*sum_sq - sum*sum)
&                /dbble(npseen)/dbble(npseen-1) )
1000    format(' Observed Pixels = ',2x,i5/
&          '           Max = ',f14.6/
&          '           Min = ',f14.6/
&          '           Avg = ',f14.6/
&          '           Sigma = ',f14.6)

C    now close the table and quit
    call ftclos(iunit,status)

99    if (status .le. 0)then
        print *, '*** Program completed successfully ***'
    else
C    get the error text description
        call ftgerr(status,errtxt)

```

```
    print *, '*** ERROR - program did not run successfully ***'  
    print *, 'status =', status, ': ', errtxt  
end if
```

```
100  continue  
     end
```