*National Aeronautics and Space Administration*
*Goddard Earth Science*
*Data Information and Services Center (GES DISC)*

# README Document for the Nimbus-4 Backscatter Ultraviolet Spectrometer (BUV) v5 Level-2 Compressed Ozone Product

BUVN4L2CPOZ

Last Revised 09/25/2015

Goddard Earth Sciences Data and Information Services Center (GES DISC)
http://disc.gsfc.nasa.gov
NASA Goddard Space Flight Center
Code 610.2
Greenbelt, MD 7271 USA

Prepared By:

James E. Johnson | 09/25/2015
Name | Date
GES DISC
GSFC Code 610.2

Reviewed By:

Name | mm/dd/yyyy
Name | Date
GSFC Code xxx

Name | mm/dd/yyyy
Name | Date
GSFC Code xxx

Goddard Space Flight Center
Greenbelt, Maryland

# Revision History

| ***Revision Date*** | ***Changes*** | ***Author*** |
|---|---|---|
| 09/25/2015 | Original | James E. Johnson |

# Table of Contents

# 1. Introduction

This document provides basic information on using the old Nimbus-4 Backscatter Ultraviolet Spectrometer (BUV) Level-2 Compressed Ozone Profile data product produced using the v5 algorithm. More recently the Nimbus-4 BUV data have been reprocessed using the updated v8.6 algorithm under the NASA MEaSUREs project, and users are encouraged to use those now.

## 1.1 Data Product Description

The Nimbus-4 Backscatter Ultraviolet Spectrometer (BUV) Level-2 Compressed Ozone Profile Data, or CPOZ, product contains the vertical distribution and total column amount of ozone (note, for the full set of data including all ancillary information, see the related HDBUV product). Each file typically contains one day of data. Each file contains total ozone, reflectivities, ozone mixing ratios, and layer ozone amounts measured every 32 seconds. The mixing ratios are given at 19 pressure levels: 0.3, 0.4, 0.5, 0.7, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 7.0, 10, 15, 20, 30, 40, 50, 70, and 100 mbar. The layer ozone amounts are given for 12 layers: 0-0.24, 0.24-0.49, 0.49-0.99, 0.99-1.98, 1.98-3.96, 3.96-7.92, 7.92-15.8, 15.8-31.7, 31.7-63.3, 63.3-127, 127-253, and 253-1013 mbar. Spatial coverage is mostly global. The data are available from April 10, 1970 (day of year 100) through May 6, 1977 (day of year 126).

> This product was previously available from the NASA National Space Science Data Center (NSSDC) under the name Backscatter Ultraviolet Spectrometer (BUV) Compressed Ozone Profile Data (CPOZ) with the identifier ESAC-00010 (old id 70-025A-05P).

### 1.1.1 The Backscatter Ultraviolet Spectrometer

The Backscatter Ultraviolet Spectrometer (BUV) was designed to monitor the vertical distribution and total column amount of atmospheric ozone on a global scale by measuring the intensity of ultraviolet radiation backscattered by the atmosphere in the 250 to 340 nm spectral region. The primary instrumentation consisted of a double monochromator containing all reflective optics and a photomultiplier detector. The double monochromator was composed of two Ebert-Fastie-type monochromators in tandem. Each monochromator had a 52 x 52-mm grating with 2400 lines per mm. Light from a 0.05-sr solid angle (subtending approximately a 222 $km^2$ area on the earth's surface from a satellite height of approximately 1100 km) entered the nadir-pointing instrument through a depolarizing filter. A motor-driven cam step rotated the gratings to monitor the intensity of 12 ozone absorption wavelengths. The detector was a photomultiplier tube. For background readings, a filter photometer measured the reflected UV radiation in an ozone-free absorption band at 380 nm.

The Nimbus-4 BUV mission was succeeded by the SBUV instrument flown later on the Nimbus-7 satellite, and subsequently the SBUV/2 instruments on a series of NOAA Polar orbiting Operational Environmental satellites. The BUV experiment was successful and returned data from April 10, 1970 through May 6, 1977. The instrument operated mostly continuously until July 1972 when the Nimbus-4 spacecraft solar array partially failed. After this time data collection had to be curtailed, particularly in the later years.

The original principal investigator for the BUV experiment was Dr.Donald F. Heath.

### 1.1.2 Nimbus-4 Overview

The Nimbus-4 satellite was successfully launched on April 8, 1970. The spacecraft included nine experiments: (1) an Image Dissector Camera System (IDCS) for providing daytime cloud cover pictures, both in real-time and recorded modes (2) a Temperature-Humidity Infrared Radiometer (THIR) for measuring daytime and nighttime surface and cloudtop temperatures, as well as the water vapor content of the upper atmosphere, (3) an Backscatter Ultraviolet Spectrometer (BUV) for measuring the emission spectra of the earth/atmosphere system, (4) a Satellite Infrared Spectrometer (SIRS) for determining the vertical profiles of temperature and water vapor in the atmosphere, (5) a Monitor of Ultraviolet Solar Energy (MUSE) for detecting solar UV radiation, (6) a Backscatter Ultraviolet (BUV) detector for monitoring the vertical distribution and total amount of atmospheric ozone on a global scale, (7) a Filter Wedge Spectrometer (FWS) for accurate measurement of IR radiance as a function of wavelength from the earth/atmosphere system, (8) a Selective Chopper Radiometer (SCR) for determining the temperatures of six successive 10-km layers in the atmosphere from absorption measurements in the 15-micrometer $CO_2$ band, and (9) an Interrogation, Recording, and Location System (IRLS) for locating, interrogating, recording, and retransmitting meteorological and geophysical data from remote collection stations.

The orbit of the satellite can be characterized by the following:

- circular orbit at 1100 km
- inclination of 80 degrees
- period of an orbit is about 107 minutes
- orbits cross the equator at 26 degrees of longitude separation
- sun-synchronous

## 1.2 Algorithm Background

The Nimbus-4 BUV data were generated using the v5 algorithm from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 360 computers using a 32-bit architecture. The intensity of UV radiation at 12 wavelengths between 250 and 340 nm are measured by the monochromator and photomultiplier detector along with geolocation, ancillary and housekeeping data. Further information on the BUV data processing can be found in the Nimbus-4 Users' Guide Section 7.

## 1.3 Data Disclaimer

The data should be used with care and one should first read the Nimbus-4 User's Guide, section 7 describing the BUV experiment. Users should cite this data product in their research.

# 2. Data Organization

The Nimbus-4 Backscatter Ultraviolet Spectrometer CPOZ Level-2 Compressed Ozone Profile Data spans the time period from April 10, 1970 to May 6, 1977. Each file typically contains one day of data.

## 2.1 File Naming Convention

The data product files are named according to the following convention:

<Instrument>-<Platform>_<Level>-<Type>_<Date>_<OrbitRange>.<Suffix>

where:

- Instrument = name of the instrument (always BUV)
- Platform = name of the platform or satellite (always Nimbus4)
- Level = processing level of data (always L2)
- Type = the data type indentifier (always CPOZ)
- Date = Data start date and time in UTC in format <YYYY>m<MMDD>t<hhmm> where
    1. YYYY = 4 digit year (1970 or 1971)
    2. MM = 2 digit month (01-12)
    3. DD = 2 digit day of month (01-31)
    4. hh = 2 digit hour of day (01-23)
    5. mm = 2 digit minute of hour (01-59)
- OrbitRange = range of orbit numbers when the data were collected (preceded by the letter ‘o’)
- Suffix = the file format (always dat, indicating binary data)

`File name example: BUV-Nimbus4_L2-CPOZ_1970m1001t0438_o2360-2371.dat`

## 2.2 File Format and Structure

The data are stored as they were originally written in IBM binary (big-endian) record oriented structured files. The files were written on the original 9-track tapes using a blocked FORTRAN form. The data archived have the tape FORTRAN leading and trailing block record size words removed, and have equal size records of seventy-two 4-byte words (288 bytes). The first record is the header, which is followed by a series of data records. The file is then padded with a set of last records to fill the original tape block sizes. The format is identical to the later Nimbus-7 SBUV HDSBUV data product.

**Figure 1:** BUV File Structure

| Record | Object |
|---|---|
| 1 | Header Record |
| 2 – L-1 | Data Records |
| L – N | Last Records |

## 2.3 Key Science Data Fields

The primary science data fields in this data product are the ozone mixing ratio profiles at 19 pressure levels, 12 layer ozone amounts, and the total column amount in Dobson units (DU).

**Figure 2:** Typical data coverage for a Nimbus-4 BUV Level-2 CPOZ daily file.

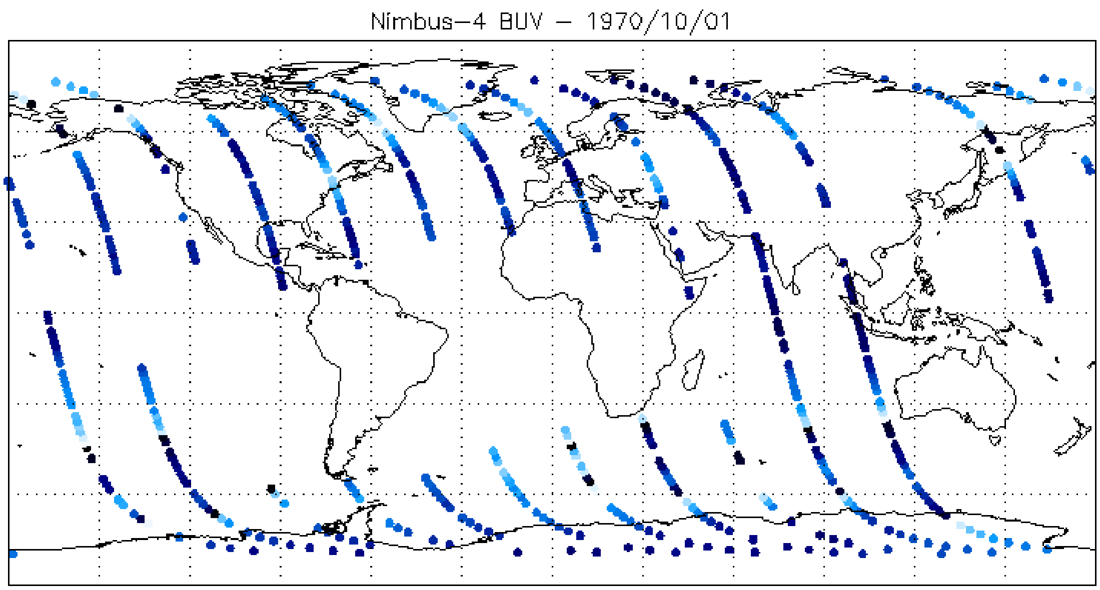
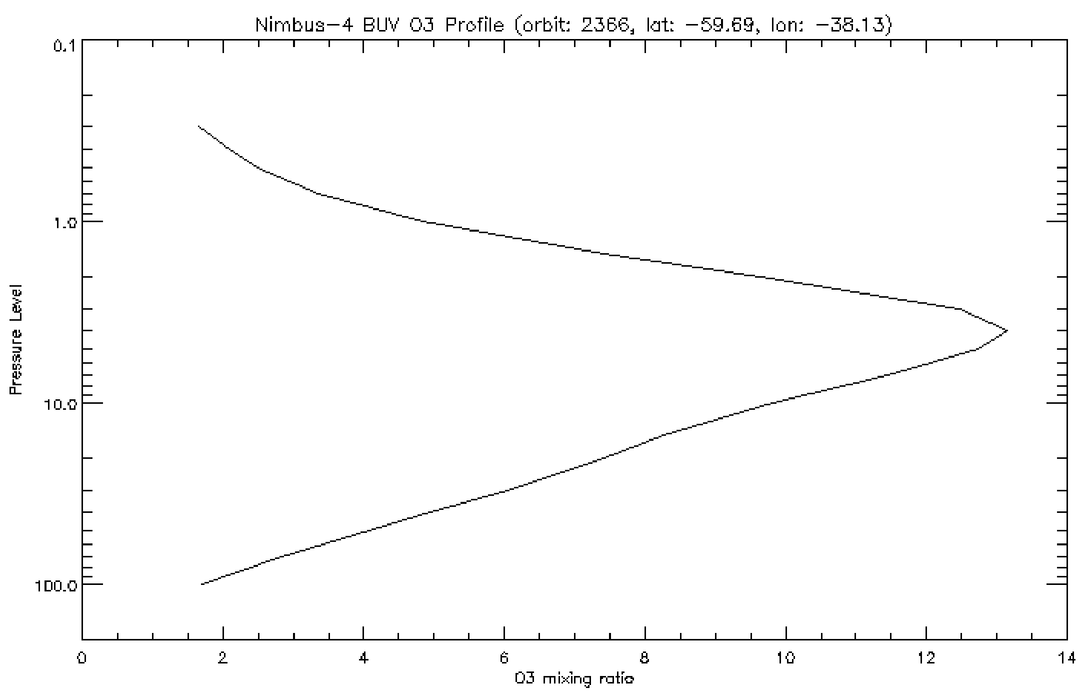



**Figure 3:** Example of a BUV $O_3$ mixing ratio profile for Oct 1, 1970 (orbit 2366).



# 3. Data Contents

The granularity of this data collection is daily.

## 3.1 Data Records

The CPOZ data files contain a single header record, a series of data records, and a number of last records (for padding out the original tape file block size). The type of record is identified by the 2nd byte in the block identifier. Each record consists of 72 4-byte words (288 bytes). See the Nimbus-7 SBUV Ozone Products User's Guide – Section 9.2 for a description of the CPOZ data file format. A sample FORTRAN program to read the data files can be found in the Appendix.

**Table 3-1-1:** Header Record

| Word | Field Name | Units | Type | Comments |
|---|---|---|---|---|
| 1 | Block Identifier | - | I*4 | |
| 2 | Logical sequence number | - | R*4 | always 1.0 |
| 3 | Orbit number | - | R*4 | |
| 4 | Year of first good scan on file | | R*4 | (minus 1900) |
| 5 | Day of first good scan on file | - | R*4 | (1-366) |
| 6 | GMT of first good scan on file | s | R*4 | |
| 7 | Subsatellite latitude for first scan | degrees | R*4 | south is negative |
| 8 | Subsatellite longitude for first scan | degrees | R*4 | west is negative |
| 9 | Spare | | R*4 | -77. |
| 10-13 | Date of job run | - | 2I*4 | EBCDIC text |
| 14-26 | 13 instrument wavelengths | nm | 13R*4 | |
| 27-39 | Solar flux at 13 wavelengths | $W/cm^3$ | 13R*4 | |
| 40-52 | Absorption coefficients at 13 wavelengths | $(atm*cm)^{-1}$ | 13R*4 | |
| 53-65 | Count to radiance conversion constants | $W/cm^3/sr$ | 37R*4 | |
| 66-72 | Spare | | 7R*4 | -77. |

Block identifier bits

- 1-2: block number
- 13-16: spare on file
- 17: set if last block
- 18: set if last file
- 19-24: record id (05 = header, 20 = data, 55 = last, 61 = trailer file record)

**Table 3-1-2:** Data Record

| Word | Field Name | Units | Type | Comments |
|---|---|---|---|---|
| 1 | Block Identifier | - | I*4 | |
| 2 | Logical sequence number | - | R*4 | |
| 3 | Orbit number | - | R*4 | |
| 4 | Year | - | R*4 | (minus 1900) |
| 5 | Day of year | - | R*4 | (1-366) |
| 6 | GMT time of day | s | R*4 | |
| 7 | View latitude | degrees | R*4 | south is negative |
| 8 | View longitude | degrees | R*4 | west is negative |
| 9 | Solar zenith angle | degrees | R*4 | |
| 10 | Data quality flag (3 digits) | | R*4 | |
| 11 | Reflectivity | | R*4 | |
| 12 | Total ozone(not using THIR Infrared cloud height info) | (matm*cm) | R*4 | |
| 13 | Total ozone (using THIR Infrared cloud height info) | (matm*cm) | R*4 | |
| 14 | Volcano Contamination Index (VCI) | | R*4 | |
| 15-26 | Monochromator N-values (244-340 nm) | | 12R*4 | |
| 27-38 | Ozone amounts – in 12 layers | matm*cm | 12R*4 | |
| 39-50 | Estimated uncertainty of the 12 layer ozone amounts | | 12R*4 | |

| | | | | |
|---|---|---|---|---|
| 51-69 | Mixing ratios at 19 levels (0.3 to 100 mbar) | µgm/gm | 19R*4 | |
| 70 | Spare | | R*4 | -77. |
| 71 | Starting solar zenith angle for scan | degrees | R*4 | |
| 71 | Ending solar zenith angle for scan | degrees | R*4 | |

Note: -77. indicates a fill value

19 Pressure levels: 0.3, 0.4, 0.5, 0.7, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 7.0, 10, 15, 20, 30, 40, 50, 70, 100

12 Umkehr layers:

| BUV Layer No. | Umkehr Layer No | Layer Pressure (mb) | Pressure at Midpoint (mb) | Layer Midpoint (km) |
|---|---|---|---|---|
| 1 | - | 0.0 - 0.247 | - | - |
| 2 | - | 0.246 - 0.495 | 0.350 | 56.5 |
| 3 | - | 0.495 - 0.990 | 0.700 | 51.0 |
| 4 | 9 | 0.990 - 1.98 | 1.40 | 45.5 |
| 5 | 8 | 1.98 - 3.96 | 2.80 | 40.2 |
| 6 | 7 | 3.96 - 7.92 | 5.60 | 35.2 |
| 7 | 6 | 7.92 - 15.8 | 11.2 | 30.4 |
| 8 | 5 | 15.8 - 31.7 | 22.4 | 25.8 |
| 9 | 4 | 31.7 - 63.3 | 44.8 | 21.3 |
| 10 | 3 | 63.3 - 127.0 | 89.6 | 17.0 |
| 11 | 2 | 127.0 - 253.0 | 179.0 | 12.5 |
| 12 | - | 253.0 - 1013.0 | 507.0 | 5.5 |

**Table 3-1-3:** Last Record

| Word | Field Name | Units | Type | Comments |
|---|---|---|---|---|
| 1 | Block Identifier | - | I*4 | |
| 2 | Logical sequence number | - | R*4 | negative |
| 3-72 | Spare | | 70R*4 | -77. |

## 3.2 Metadata

The metadata are contained in a separate XML formatted file having the same name as the data file with .xml appended to it.

**Table 3-2:** Metadata attributes associated with the data file.

| Name | Description |
|---|---|
| LongName | Long name of the data product. |
| ShortName | Short name of the data product. |
| VersionID | Product or collection version. |
| GranuleID | Granule identifier, i.e. the name of the file. |
| Format | File format of the data file. |
| CheckSumType | Type of checksum used. |
| CheckSumValue | The value of the calculated checksum. |
| SizeBytesDataGranule | Size of the file or granule in bytes. |
| InsertDateTime | Date and time when the granule was inserted into the archive. The format for date is YYYY-MM-DD and time is hh-mm-ss. |
| ProductionDateTime | Date and time the file was produced in format YYYY-MM-DDThh:mm:ss.ssssssZ |
| RangeBeginningDate | Begin date when the data was collected in YYYY-MM-DD format. |
| RangeBeginningTime | Begin time of the date when the data was collected in hh-mm-ss format. |
| RangeEndingDate | End date when the data was collected in YYYY-MM-DD format. |
| RangeEndingTime | End time of the date when the data was collected in hh-mm-ss format. |
| WestBounding Coordinate | The westernmost longitude of the bounding rectangle(-180.0 to +180.0) |
| NorthBounding Coordinate | The northernmost latitude of the bounding rectangle(-90.0 to +90.0) |
| EastBounding Coordinate | The easternmost longitude of the bounding rectangle(-180.0 to +180.0) |
| SoutthBounding Coordinate | The southernmost latitude of the bounding rectangle (-90.0 to +90.0) |

| | |
|---|---|
| PlatformShortName | Short name or acronym of the platform or satellite |
| InstrumentShortName | Short name or acronym of the instrument |
| SensorShortName | Short name or acronym of the sensor |
| FirstOrbitNumber | Number of the first orbit in the file |
| LastOrbitNumber | Number of the last orbit in the file |
| TapeFileName | Sequence number of file from the original tape preceded with ‘f’ and ending in ‘.dat’ |

# 4. Reading the Data

The data are written in a binary record-oriented format. Using the record format specification in the section above, users can write software to read the data files. Please note that the data were originally written using a big-endian format, therefore users on little-endian machines will need to swap bytes for the words. Also, the floating point data were written using IBM 360 machines, and must be converted if reading on a machine that understands IEEE floats (integers are not affected).

A sample FORTRAN program is included in the Appendix section which will read the data files. Additionally three FORTRAN functions are included to perform byte swapping, conversion from IBM float to IEEE float, and translation of EBCDIC to ASCII text.

# 5. Data Services

## 5.1 Reverb

The GES DISC provides basic temporal and advanced (event) searches through the EOSDIS Reverb data search and download interface:

http://reverb.echo.nasa.gov

Reverb allows users the ability to search on keywords, spatial region, and time period on datasets archived and various data centers. It offers various download options that suit users with different preferences and different levels of technical skills. To search for the CPOZ data enter GES_DISC_BUVN4L2CPOZ_V005 into the keyword field.

## 5.2 FTP

The Nimbus data products are available for users to download directly using anonymous FTP:

ftp://acdisc.gsfc.nasa.gov/data/s4pa/Nimbus4_BUV_Level2/BUVN4L2CPOZ.005/

The data are organized in directories by year with subdirectories by day of year. README, User's Guide and other documentation are located under the doc directory.

# 6. More Information

## 6.1 Web Resources

For other Nimbus data products, please see the GES DISC's Nimbus heritage data web page at:
http://disc.gsfc.nasa.gov/nimbus/

To search for other related data, please visit NASA's Global Change Master Directory at:
http://gcmd.nasa.gov.

## 6.2 Point of Contact

Name: GES DISC Help Desk
URL: http://disc.gsfc.nasa.gov/
E-mail: gsfc-help-disc@lists.nasa.gov
Phone: 301-614-5224
Fax: 301-614-5228
Address: Goddard Earth Sciences Data and Information Services Center
Attn: Help Desk
Code 610.2
NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA

## 6.3 References

D. F. Heath, and A. J. Kruger, "The Nimbus-4 User's Guide - Section 7: The Backscatter Ultraviolet Spectrometer (BUV) Experiment", NASA Goddard Space Flight Center, March 1970, Pages 149-172

A.J. Fleig and R. D. McPeters, et al, "Nimbus 7 Solar Backscatter Ultraviolet (SBUV) Ozone Products User's Guide – Section 9.2: CPOZ Tape Format", NASA Goddard Space Flight Center, 1990, Pages 73-81

# 7. Appendices

## Acknowledgements

The Nimbus data recovery task at the GES DISC is funded by NASA's Earth Science Data and Information System program.

## Acronyms

*EOS:* Earth Observing System

*ESDIS:* Earth Science and Data Information System

*GES DISC*: Goddard Earth Sciences Data and Information Services Center

*GSFC*: Goddard Space Flight Center

*BUV*: Backscatter Ultraviolet Spectrometer

*CPOZ*: Compressed Backscatter Ultraviolet Spectrometer Data Product

*L1:* Level-1 Data

*NASA:* National Aeronautics and Space Administration

*Reverb:* ECHO's Next Generation Metadata and Service Discovery Tool

*QA:* Quality Assessment

*UT:* Universal Time

# FORTRAN Code

```
C-------------------------------------------------------------------
C ^NAME: READ_CPOZ
C
C ^DESCRIPTION:
C     This program opens and reads a Nimbus-4 CPOZ level-2 data file
C     and prints the contents of the file to the screen. Data files
C     consist of a single header, a number of data, and a final trailer
C     record, each of size 288 bytes (72 4-byte words). See the Nimbus-7
C     SBUV Users Guide Section 9.2 for the file specification of the
C     Nimbus-4 CPOZ product.
C
C ^MAJOR VARIABLES:
C     FNAME  - name of input file
C
C ^NOTES:
C     Compile: gfortran -o READ_CPOZ.EXE READ_CPOZ.FOR
C
C ^ORGANIZATION: NASA/GSFC, Code 610.2
C
C ^AUTHOR:  James Johnson
C
C ^ADDRESS: james.johnson@nasa.gov
C
C ^CREATED: September 23, 2015
C-------------------------------------------------------------------

      CHARACTER          FNAME*1024    ! Name of input file
      INTEGER*4          IREC(72)      ! Record is 72 4-byte words

C     Get the name of the input data file to read
      PRINT *, 'Enter the name of the input file:'
      READ (5,'(A)') FNAME

C     Open the specified input file
      OPEN (UNIT=1, FILE=FNAME, STATUS='OLD', ACCESS='DIRECT',
     &      FORM='UNFORMATTED', RECL=4*72, ERR=99, IOSTAT=IOS)
      PRINT '("FILE: ",A80)', FNAME

C     Initialize N (block number)
      N=1

C     Loop through the file reading all blocks of data
   10 DO

        READ (1, REC=N, IOSTAT=IOS, ERR=90) IREC

C       Check the record type. This is byte 2 (3 unswapped) in the first
C       4-byte word. Value 05 = header, 20 = data, 55 = last record
        ITYPE = ISHFT(IAND(I4SWAP(IREC(1)), Z'00003F00'), -8)
        IF (ITYPE .EQ. 05) THEN
          CALL PRHREC(IREC, N)
        ELSE IF (ITYPE .EQ. 20) THEN
          CALL PRDREC(IREC, N)
        ELSE IF (ITYPE .EQ. 55) THEN
          CALL PRLREC(IREC, N)
        ELSE
          PRINT '("UNKNOWN RECORD TYPE: ", I3,X,I11)', ITYPE, N
        ENDIF
```

```
        N=N+1

      END DO

C     Close the input file
   90 CLOSE(1)
      GOTO 100

   99 PRINT '("ERROR: OPENING FILE, IOSTAT: ",I6)', IOS

  100 STOP
      END

C-----------------------------------------------------------------------
C ^SUBROUTINE: PRHREC
C
C     This subroutine will print the header record to the screen
C-----------------------------------------------------------------------

      SUBROUTINE PRHREC(IREC, N)

      INTEGER*4           IREC(72)      ! Data record
      INTEGER*4           IEBC(4)       ! Temporary array for EBC text
      CHARACTER           EBCTMP(16)    ! Temporary EBCDIC character
      CHARACTER           ASCVAL*16     ! Array for ASCII text
      CHARACTER           EBC2ASC       ! Function to convert EBC to ASCII
      EQUIVALENCE         (IEBC,EBCTMP)

C Word 1 contains hexidecimal data; words 4-7 contain EBCDIC data, and
C the remaining words contain data in IBM floating-point format (REAL4).

      PRINT '("HEADER RECORD")'
      DO 10 I=1,72

        IF (I .EQ. 1) THEN
          CALL IBLKID(I4SWAP(IREC(I)), IBLKNO, ILASTB, ILASTF, IRECID)
          PRINT '("WORD ",I3," = {",I4,X,L,X,L,X,I4,"}")',
     &            I,IBLKNO,ILASTB,ILASTF,IRECID
        ELSE IF (I .GE. 10 .AND. I .LT. 14) THEN
          IF (I .NE. 10) GOTO 10
          IEBC = IREC(I:I+3)
          DO 20 J=1,16
            ASCVAL(J:J) = EBC2ASC(ICHAR(EBCTMP(J)))
   20     CONTINUE
          PRINT '("WORD ",I3," = ",A16)', I,ASCVAL
        ELSE
          PRINT '("WORD ",I3," = ",G12.6)', I,R4IBM(I4SWAP(IREC(I)))
        END IF

   10 CONTINUE
      PRINT '("")'

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C ^SUBROUTINE: PRDREC
C
C     This subroutine will print the data records to the screen
C-----------------------------------------------------------------------

      SUBROUTINE PRDREC(IREC, N)

      INTEGER*4          IREC(72)      ! Data record

C All words are in IBM floating-point format (REAL*4) except word 1,
C which contains hexidecimal data. Any word in the record may contain
C -77., indicating fill data.

      PRINT '("DATA RECORD")'
      DO 10 I=1,72

        IF (I .EQ. 1) THEN
          CALL IBLKID(I4SWAP(IREC(I)), IBLKNO, ILASTB, ILASTF, IRECID)
          PRINT '("WORD ",I3," = {",I4,X,L,X,L,X,I4,"}")',
     &            I,IBLKNO,ILASTB,ILASTF,IRECID
        ELSE
          PRINT '("WORD ",I3," = ",G12.6)', I,R4IBM(I4SWAP(IREC(I)))
        END IF

   10 CONTINUE
      PRINT '("")'

      RETURN
      END

C-----------------------------------------------------------------------
C ^SUBROUTINE: PRLREC
C
C     This subroutine will print the last records to the screen
C-----------------------------------------------------------------------

      SUBROUTINE PRLREC(IREC, N)

      INTEGER*4          IREC(72)      ! Data record

C All words are in IBM floating-point format (REAL*4) except word 1,
C which contains hexidecimal data.

      PRINT '("LAST RECORD")'
      DO 10 I=1,72

        IF (I .EQ. 1) THEN
          CALL IBLKID(I4SWAP(IREC(I)), IBLKNO, ILASTB, ILASTF, IRECID)
          PRINT '("WORD ",I3," = {",I4,X,L,X,L,X,I4,"}")',
     &            I,IBLKNO,ILASTB,ILASTF,IRECID
        ELSE
          PRINT '("WORD ",I3," = ",G12.6)', I,R4IBM(I4SWAP(IREC(I)))
        END IF

   10 CONTINUE
      PRINT '("")'

      RETURN
      END
```

```
C------------------------------------------------------------------------
C ^SUBROUTINE: IBLKID
C
C     This subroutine will decode the block identifier (word 1)
C------------------------------------------------------------------------

      SUBROUTINE IBLKID(IWORD, IBLKNO, ILASTB, ILASTF, IRECID)
                                                      !  bits
          IBLKNO = ISHFT(IWORD,-20)                   !  1-12 Block Number
          ILASTB = IAND(IWORD,'1000000000000000'B)    !    17 Last Block of File
          ILASTF = IAND(IWORD,'0100000000000000'B)    !    18 Last File of Tape
          IRECID = ISHFT(ISHFT(IWORD,18),-26)         ! 25-32 Record Identifier

      RETURN
      END

C------------------------------------------------------------------------
C ^FUNCTION: I4SWAP
C
C     This subroutine will swap the bytes of a data element
C------------------------------------------------------------------------

      FUNCTION I4SWAP(INPUT)

      INTEGER*4          IWORD          ! Input 4-byte word
      INTEGER*4          IDROW          ! Byte-swapped 4-byte word
      CHARACTER          DATBUF(4)      ! Input data buffer
      CHARACTER          SWPBUF(4)      ! Output swapped buffer
      EQUIVALENCE        (IWORD, DATBUF)
      EQUIVALENCE        (IDROW, SWPBUF)

      IWORD = INPUT
      DO 10 K=1,4
        SWPBUF(K) = DATBUF(4-K+1)
   10 CONTINUE
      I4SWAP = IDROW

      RETURN
      END

C------------------------------------------------------------------------
C ^FUNCTION: R4IBM
C
C     This function will convert an input word to an IBM float
C------------------------------------------------------------------------

      FUNCTION R4IBM(IWORD)

      INTEGER*4      IDROW              ! reverse the bits of input word
      REAL*8         A /16.0/           ! base number
      INTEGER*4      B /64/             ! exponent offset
      REAL*8         C /0.0/            ! fraction offset
      INTEGER*1      S                  ! sign flag
      INTEGER*2      E                  ! binary exponent
      REAL*8         F                  ! binary fraction
      REAL*8         M                  ! mantissa
      REAL*8         V                  ! float value

      S = ISHFT(IWORD, -31)

      E = 0
      DO 10 I=0,6
        E = E + IAND(ISHFT(IWORD,-24),ISHFT(1,I))
   10 END DO
```

```
      IDROW = 0
      DO 11 I=0,31
        IF (IAND(IWORD,ISHFT(1,I)) .NE. 0) THEN
          IDROW = IOR(IDROW,(ISHFT(1,31-I)))
        END IF
   11 END DO

      F = 0.0
      DO 12 I=0,31
        IF (ISHFT(IAND(ISHFT(IDROW,-8),ISHFT(1,I)),1) .NE. 0) THEN
          F = F + 1 / FLOAT(ISHFT(IAND(ISHFT(IDROW,-8),ISHFT(1,I)),1))
        END IF
   12 END DO

      M = C + F                         ! calculate the mantissa
      V = (-1)**S * M * A**(E - B)      ! calculate the float value
      IF (ABS(V) .LT. 2.0**(-149)) THEN
        V = (-1)**S * 0.0               ! avoid underflow
      END IF

      R4IBM = V
      RETURN
      END

C------------------------------------------------------------------------
C ^FUNCTION: EBC2ASC
C
C     This function will convert an EBCDIC character to ASCII
C------------------------------------------------------------------------

      CHARACTER FUNCTION EBC2ASC(N)

      INTEGER            EBCTAB(256)

C  EBCDIC to ASCII table
      DATA EBCTAB/
     & 000,001,002,003,156,009,134,127,151,141,142,011,012,013,014,015, ! 0
     & 016,017,018,019,157,133,008,135,024,025,146,143,028,029,030,031, ! 1
     & 128,129,130,131,132,010,023,027,136,137,138,139,140,005,006,007, ! 2
     & 144,145,022,147,148,149,150,004,152,153,154,155,020,021,158,026, ! 3
     & 032,160,161,162,163,164,165,166,167,168,213,046,060,040,043,124, ! 4
     & 038,169,170,171,172,173,174,175,176,177,033,036,042,041,059,094, ! 5
     & 045,047,178,179,180,181,182,183,184,185,229,044,037,095,062,063, ! 6
     & 186,187,188,189,190,191,192,193,194,096,058,035,064,039,061,034, ! 7
     & 195,097,098,099,100,101,102,103,104,105,196,197,198,199,200,201, ! 8
     & 202,106,107,108,109,110,111,112,113,114,203,204,205,206,207,208, ! 9
     & 209,126,115,116,117,118,119,120,121,122,210,211,212,091,214,215, ! A
     & 216,217,218,219,220,221,222,223,224,225,226,227,228,093,230,231, ! B
     & 123,065,066,067,068,069,070,071,072,073,232,233,234,235,236,237, ! C
     & 125,074,075,076,077,078,079,080,081,082,238,239,240,241,242,243, ! D
     & 092,159,083,084,085,086,087,088,089,090,244,245,246,247,248,249, ! E
     & 048,049,050,051,052,053,054,055,056,057,250,251,252,253,254,255/ ! F
C       0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F

      EBC2ASC = CHAR(EBCTAB(N+1))

      RETURN
      END
```