

## PURPOSE OF CODE

BRFDUMP is a companion program to HDFDUMP designed to specifically calculate the radiances and BRF's for Level 1B2 Ellipsoid and Terrain Projected data. It works in much the same way as HDFDUMP and has an almost identical interface. Please read the HDFDUMP documentation first before moving onto this document.

## CALCULATION OF BRF AND RADIANCE

$$r(-\mu, \mu_0, \Phi - \Phi_0) = \frac{\pi L(-\mu, \mu_0, \Phi - \Phi_0) \cdot d^2}{\mu_0 E_{0,b}^{std}}$$

where

- r = BRF as a function of viewing and solar angles
- L = measured radiance from L1B2 Radiance Product
- $\mu$  = cosine of viewing zenith angle
- $\mu_0$  = cosine of solar zenith angle
- $\Phi - \Phi_0$  = difference between solar and viewing azimuth angles
- $E_{0,b}^{std}$  = standardized band-weighted solar irradiance
- d = solar distance in AU

The solar zenith angles (originally provided at 17.6km resolution) are linearly interpolated to the same resolution as the radiance data.

## RUNNING THE PROGRAM

When you run BRFDUMP you are prompted for the names of the three necessary data files - the Level 1B2 Radiance file, the corresponding Geometric Parameters file that contains the solar zenith angles (both of these vary orbit by orbit) and the PreFlightCalibration Ancillary Radiometric Product (ARP) file. This latter one contains several scale factors necessary for the conversion of the L1B2 data to actual radiances and is constant.

Then, like HDFDUMP, you will be prompted for the number of the grid and field that you want to dump, and the block range that you desire. As before, absolute block numbers are always used. Currently the data for each camera is contained in a separate file and there are separate grids for each band so there is no need to specify the index numbers for any additional dimensions. You will be asked to confirm both the data selection and its resolution.

Dumping an entire orbit's worth of data at 1.1km resolution will need about 300 Meg in main memory. Trying to do this task at 275m resolution will require a huge amount of memory, beyond the capacity of most computers.

A sample session is included below:

```
catherin@khamzin:/kham2/MISR_data/path109 [28]>brfdump
enter name of radiance file
MISR_AM1_GRP_ELLIPSOID_GM_P109_O003316_AF_01.hdf
enter name of arp file
MISR_AM1_ARP_PRFLTCAL_F01_003.hdf
enter name of geometric parameters file
MISR_AM1_GP_GMP_P109_O003316_01.hdf
```

PLEASE CHOOSE GRID/FIELD TO DUMP

GRID 1 BlueBand

FIELDS:

1 Blue Radiance/RDQI

GRID 2 GreenBand

FIELDS:

1 Green Radiance/RDQI

GRID 3 RedBand

FIELDS:

1 Red Radiance/RDQI

GRID 4 NIRBand

FIELDS:

1 Infrared Radiance/RDQI

ENTER GRID NUMBER: 1 - 4

4

ENTER FIELD NUMBER: 1 - 1

1

YOU HAVE SELECTED GRID/FIELD = NIRBand : Infrared Radiance/RDQI

NAMES AND SIZES OF DIMENSIONS

YDim 512

XDim 128

SOMBlockDim 180

ENTER FIRST AND LAST BLOCKS TO READ IN

USE ABSOLUTE BLOCK NUMBERS

VALID DATA EXISTS FOR BLOCKS: 2 - 145

2 145

DETERMINING DATA RESOLUTION

LIKELY RESOLUTION = (LTRACK,XTRACK) = 1100.0 1100.0

IS THIS CORRECT? (Y/N)

Y

```
IS DATA SELECTION OK? (Y/N)
Y
IPAD = 1536
```

```
DATA TYPE IS 32-BIT FLOATING POINT
WRITING OUT CONTENTS OF DATA FILES
NLINE = 18432
NSMP = 512
NSMP+IPAD = 2048
```

## OUTPUT OF PROGRAM

Like HDFDUMP, BRFDUMP produces output files in both the aligned and stacked format. It actually produces four output files, the radiances and the BRF's in aligned and stacked format. The corresponding filenames are '*img.align.brf.bin*', '*img.stack.brf.bin*', '*img.align.rad.bin*', and '*img.stack.rad.bin*'.

The datatype is always 32-bit floating point, and the sizes of the files are the same as those produced from HDFDUMP. The *img.align* files are NLINE lines and NSMP+IPAD columns, and the *img.stack* files are NLINE lines by NSMP columns.

## COMPILING CODE

The source code is provided for you free of charge. To compile the code you need a Fortran90 compiler (that has the option of extending the source code lines to 80 characters), and the HDF and HDF-EOS libraries. A copy of the executable compiled and run on an SGI IRIX system (with the -n32 flag) has also been provided.

Set the following environmental variables:

```
HDFINC = location of HDF include files
HDFLIB = location of HDF libraries
HDFEOS_LIB = location of HDF-EOS libraries
```

```
prompt> f90 -col80 -o hfdump hfdump_public.f \
-I$HDFINC -L$HDFLIB -lmfhdf -ldf -ljpeg -lz \
-L$HDFEOS_LIB -lhdfEOS -lGctp
```

The HDF and HDFEOS libraries can be found at the following locations:

```
HDF: http://hdf.ncsa.uiuc.edu
HDFEOS:http://newsroom.gsfc.nasa.gov/sdptoolkit/toolkit.html
```

For reasons explained in the HDFDUMP readme file, the compilation will probably fail on the first time through. Compiling the code two or three times in succession should remove all of those problems.

## **CONTACT INFORMATION**

If you have any questions about BRFDUMP or problems getting it to run, please contact the author at the following address:

Catherine Moroney  
Jet Propulsion Laboratory  
Mail Stop 169-315  
4800 Oak Grove Drive  
Pasadena, CA 91109-8099

(818) 393-3392  
Catherine.M.Moroney@jpl.nasa.gov

## **COPYRIGHT INFORMATION**

The accompanying software may be freely copied and distributed provided this notice in its entirety is included. No responsibility is assumed for the selection, performance, or use of this software.

*Last Updated: November 9, 2000*