Using the ARPS WSR-88D Data Remapper

Keith Brewster
Center for Analysis and Prediction of Storms
University of Oklahoma
Norman, OK 73019
kbrewster@ou.edu

Revised: 9 August 2004[1]
ARPS version 5.1.2

1. Introduction

The ARPS program 88d2arps reads raw radar data in polar coordinates and remaps the data onto an ARPS terrain-following Cartesian grid. The program reads data from WSR-88D archive level-II tapes, from disk files, or from the RIDDS live circular buffer. The program uses a local least-squares fit of the observed data to interpolate. The interpolation fits each data type to a function that is quadratic in the horizontal and linear in the vertical. One output file is produced for each radar data "volume scan." Data returned are reflectivity factor (dBZ), radial velocity (ms^{-1}), spectrum width (ms^{-1}) and local Nyquist velocity (ms^{-1}) interpolated to the three-dimensional ARPS grid. Optionally, the user may also direct the program to write files containing the lowest tilt of reflectivity and velocity interpolated to the ARPS grid in two dimensions. The data are checked for velocity folding, transient echoes in clear air are removed (image is despeckled), and the data are screened for anomalous propagation (AP).

The ARPS grid used for remapping is specified through the regular ARPS input file (e.g., `arps.input`). The namelist variables dealing with the grid spacing, grid location, vertical stretching, terrain and map projection are used to establish the grid just as for an ARPS model run. The namelist specifying the ARPS initial file (which can be an ARPS forecast, external model data interpolated to the ARPS grid, or an ARPS-formatted sounding) is used to establish the mean wind profile for wind data quality control. The remaining namelist variable blocks are not used.

Most controls for the remapping itself are specified in the command line. Some controls may be specified through environment variables; see the instructions for running 88d2arps.

2. Files

The radar remapper is contained in several source code files (C and Fortran) that are distributed with the official CAPS source code in the directory:

[1] Revised on July 31, 2007 for arps5.2.8 to read China netrad (CINRAD) data by Yunheng Wang.

./src/88d2arps

The main driver source is 88d2arps.c. The directory also contains files needed for building the executable and a short script for setting the optional environment variables.

The ARPS distribution also contains a library of I/O routines (written in C) for reading the NEXRAD Level-II data. The makearps make command will automatically build these libraries from the source code located in the directory: ./src/88d2arps/a2io.

You may need to install the zlib.a which is used for uncompressing radar files. If this library is not available on your system, the source code may be downloaded for free. Information about this can be found on the web at URL:
http://www.info-zip.org/pub/infozip/zlib/

Information about the location of the NEXRAD radars is contained in an ASCII formatted data file `radarinfo.dat`. This file is distributed in the `./data/adas` directory. If you have not already done so, copy the file to the directory in which you are running `88d2arps`. Additional radars can be added to the `radarinfo.dat` file using a text editor, should that be necessary. The information in the file is radar name, site name, latitude (degrees N, minutes and seconds), longitude (degrees W, minutes and seconds), and elevation above sea level (m).

3. Building the Executable

The 88d2arps executables can be built using the makearps command. The executable file is built using the UNIX make utility.

Check the locations for the libraries specified in the `makearps` csh script to see that they match the actual locations. Also be sure the zlib is specified (-lz) as this was not needed for earlier ARPS releases. The makearps script library statements should read something like:

```
case 88d2arps:
   set LIBS = '-lz'
breaksw
```

The file `setREMAPenv` is a csh script for setting environment variables, it is only needed if you are running in real time and redirecting the output to another machine.

To build the executable

```
makearps 88d2arps
```

4. Command Line Options

The user may control some aspects of the data processing and output through the use of the following command line options.

`-dir directory_name`: Write the remapping output file to the specified directory.

`-hdf n` : Store the output in HDF-4 format using compression level n, where n is an integer from 0 to 5:

HDF-4 is a standardized portable data format from the National Center for Supercomputing Applications (NCSA, http://hdf.ncsa.uiuc.edu/hdf4.html). This option is recommended if you will be running 88d2arps on one machine and copying it to another machine for use in other applications.

Consult the HDF website for more information about their compression routines. Generally we have found that good compression with high efficiency is obtained with compression levels 1 or 2.

`-vad:` In addition to the standard output file, write a Velocity-Azimuth-Display VAD wind profile file for use as an ADAS data file. The file will have a name Kxxx.YYMMDD.HHMM.vad

`-reffile` In addition to the standard output file, write a 3d binary file of the reflectivities. This file can be used with arpsplt and the 3d "arbitrary variable" plotting option of the arpsplt plotting program. The file will have a name that looks like
`dirname/runname.refl__000000`

`-velfile`: In addition to the standard output file, write a 3d binary file of the radial velocities. This file can be used with arpsplt and the 3-d "arbitrary variable" plotting option of the arpsplt plotting program. The file will have a name that looks like
`dirname/runname.radv__000000`

`-ref2d`: In addition to the standard output file write a 2d binary file containing the lowest tilt of reflectivity data (generally 0.5 degrees) interpolated to the ARPS grid. This file can be used with arpsplt and the 2-d "arbitrary variable" plotting option of the arpsplt plotting program. The file will have a name that looks like
`dirname/runname.refl2d000000`

`-vel2d`: In addition to the standard output file write a 2d binary file containing the lowest tilt of radial velocity data (generally 0.5 degrees) interpolated to the ARPS grid. This file can be used with arpsplt and the 3-d "arbitrary variable" plotting option of the arpsplt plotting program. The file will have a name that looks like
`dirname/runname.radv2d000000`

`-novel`: Do not process the velocity data. This saves CPU time if turnaround time is critical and you only desire the remapped reflectivity data. The velocity data are set to "missing" in the output file. Some processing is done on the velocity data

in the quality control routines for the purposes of anomalous propagation detection.

`-noqc`: Do not execute the quality control routines. This is useful for checking and tuning the quality control routines (comparing before-and-after results) and is not recommended for general use.

`-rad98`: To read CINRAD file (98d_data – without 24 byte).

4. Running the Program

a) To run 88d2arps to read WSR-88D Archive-II data tape or disk file:

1) Edit an ARPS input file (e.g., `arps.input`, to specify desired grid parameters, including the terrain file (if desired). A map projection option other than "zero" must be selected. Also an input grid (inifile) or sounding MUST be specified. The input gridded data or sounding data are used in the velocity unfolding process. The unfolding routine computes a mean velocity profile from the gridded data or sounding file in the vicinity of the radar. Contact Keith Brewster at CAPS (kbrewster@ou.edu) for information about programs to convert some common archived sounding formats to the ARPS sounding format.

2) If running in a real-time mode and you want your files copied to another machine or directory as they are created: Edit the environment script file, setREMAPenv, to set the proper radar name, and rcp user and destination directory. The setenv commands for rcp are optional and if they are set to a blank value, they will cause the program to skip over this feature.

setREMAPenv lines:

```
setenv RADARNAME KTLX   [4-character name of radar]
setenv REMAP_DIR ./     [destination directory for output files]
setenv REMAP_USER user  [optional user name for rcp command, used
                                to copy output to remote system
                               ..leave blank to disable rcp action]
setenv REMAP_REMOTE stratus
                [optional: destination machine for rcp command]
setenv REMAP_DEST /scratch/stratus/user
                [optional: destination directory for rcp]
setenv REMAP_COMPRESS gzip
                [optional: desired compression utility for output files:
                 gzip, compress or nothing]
```

Once you are done editing, set the environment variables

```
source setREMAPenv
```

4) Be sure you have the file "radarinfo.dat" in the directory of execution. This is distributed in the ./data/adas directory in the normal ARPS distribution.

5) If reading from an archive tape, insert tape in drive (in the following, drive named /dev/rmt2 is used as an example). The tape drive name used MUST be non-rewinding on close.

6) If reading from an archive tape, advance the tape to desired file on tape. Each volume scan (5-10 minutes of data) constitutes a file on the tape. The program processes all files it encounters, so to save time you need to skip the files you do not wish to process. You may use the `mt` command.

```
mt -f /dev/rmt2.1 fsf 100
```

7) Run the program

If reading from an archive tape:

```
88d2arps Kxxx -f /dev/rmt2.1 [other options] < arps.input
```

where Kxxx is the radar name.

The program will run until the end of tape is reached, so use control-C to stop it. The program may also be stopped, and later restarted, if you find you need to reposition the tape to a different file.

```
<ctrl>C
```

If reading from a single file:

```
88d2arps Kxxx -diskf full_filename [other options] < arps.input
```

where Kxxx is the radar name.

b) To run 88d2arps to read from the RIDDS circular buffer

1) Edit ARPS input file to specify desired grid parameters

2) Edit the script, setREMAPenv, to set the proper radar name, and rcp user and destination directory. See instructions in section 4b), above, for guidance.

3) Set the environment variables

```
source setREMAPenv
```

4) Run the program

```
88d2arps [other options] < arps.input
```

5) Stop the program. The program will run indefinitely, so use control-C to stop it. The program may also be stopped, and later restarted, if you find you need to reset the environment variables, change the grid, etc.

```
<ctrl>C
```

c) To run 88d2arps to read from the CINRAD radar file, please read CINRAD.pdf in directory docs/ of the ARPS package.

5. Reading the output file(s)

The data may be read into your application program using the subroutine `rdradcol`, which is contained in the file `./src/adas/rdradcol.f`. The data are stored and read-in as columns of data which are identified by their i,j indexes and their latitude and longitude. See the `rdradcol` source code for details.

The VAD data are stored in an ASCII-formatted file that can be read by ADAS.

The output from `–reffile`, `-velfile`, `-vel2d` and `–ref2d` command line switches are binary files containing the gridded data. They can be read into `arpsplt` for plotting (see Section 6).

6. Plotting the output

A program to examine the output of 88d2arps is provided for your convenience. It requires the NCARgraphics libraries.

1) Create the program pltradcol.

```
~~ncargf77 -o pltradcol pltradcol.f maproj3d.f pltmap.f timelib3d.f~~
      makearps -zxncar pltradcol
or
      makearps -zxpost pltradcol
```

2) Run pltradcol, it will prompt for filename.

```
      pltradcol
```

3) examine the gmeta file

```
      idt gmeta
```

Parameters in the source code in pltradcol.f may be modified to change which data are plotted.

The output from the `–reffile, -velfile, -vel2d` and `–ref2d` command line switches are binary files containing the gridded data. They can be read into `arpsplt` for plotting.

To plot the data in these files using the plotting program `arpsplt`:

1) Open the `arpsplt.input` file in a text editor.
2) Set the ARPS gridded file to be the same gridded file used as `inifile` in `arps.input` for `88d2arps`.
3) Set the input variable `arbvaropt=1`
4) To plot the 3-D data, set `var3dnum=n` where `n` is the number of 3d files.

5) Set `var3d(i)='reflct'` to plot the output from the command line switch `-reffile`.
   Set `var3d(i)='radvel'` to plot the output from the command line switch `-velfile`
6) Set the directory name, plotting contours and colors as needed. As with other 3-D data, the plotting of 2-D slices of 3-D data are controlled by the namelist variables `slice_xy`, `slice_xz`, and `slice_yz`.
7) To plot the 2-D data, set `var2dnum=n` where `n` is the number of 2-d files.
8) Use `var2d(i)='refl2d'` to plot the output from the command line switch `-ref2d`
   Use `var2d(i)='radv2d'` to plot the output from the command line switch `-vel2d`
9) Set the directory name, plotting contours and colors as needed.
10) Run `arpsplt` normally, e.g.

```
arpsplt < input/arpsplt.input
```