IDL AACGM DLM

(Author: Haje Korth, JHU/APL, haje.korth@jhuapl.edu)

Introduction:

IDL AACGM DLM is a Dynamic Link Module (DLM) for the Interactive Data Language (IDL) by Research Systems, Inc. The purpose of this library is to provide easy access to the AACGM library by R. J. Barnes, which is based on algorithms by K. Baker and S. Wing. I do not claim any rights to those routines. My only intellectual assets are in the interface itself. The interface allows calling of the AACGM functions from the IDL command line without any knowledge of the C language. All commands behave like native IDL procedures and are faster than the native IDL implementation of the library.

Installation:

The IDL AACGM DLM consists of the files idl_aacgm.dlm and idl_aacgm.dll. For installation, simply copy the above two files into the IDL executable directory, or into a directory in the search path of the IDL system variable !DLM_PATH. Then set the environment variable AACGM_PATH to the directory that holds the coefficient files plus the prefix of the file name up to the year. Use a double backslash as path separator. For example: c:\\aacgm\\coeff\\. The coefficient files must be named aacgm_coeffsYYYY.asc, where YYYY is the four digit year.

Implemented Procedures:

0. AACGM_HELP

   Description: List AACGM DLM functions and procedures.

   Calling Sequence: aacgm_help.

   Inputs: None.

   Outputs: None.

1. AACGM_SET_PATH

Description: Set AACGM_PATH environment variable.

Calling Sequence: aacgm_set_path, path.

Inputs: path: File path to aacgm_coeffsYYYY.asc files. Use "\\" as path separator.

Outputs: None.

Keywords: quiet: Suppress informational messages.

2. AACGM_LOAD_COEF

Description: Load AACGM coefficient file.

Calling Sequence: aacgm_load_coef, year.

Inputs: year: Year of coefficients to be used. Allowed values are 1975, 1980, 1985, 1990, 1995, 2000.

Outputs: None.

Keywords: quiet: Suppress informational messages.

3. AACGM_CONV_COORD

Description: Convert from geographic to AACGM coordinates and vice versa. Coordinates may be given as vectors. Use this feature to avoid slow IDL loops.

Calling Sequence: aacgm_conv_coord, glat, glon, hgt, mlat, mlon, err, /to_aacgm or
aacgm_conv_coord, mlat, mlon, hgt, glat, glon, err, /to_geo.

Inputs: glat, glon: Geographic latitude and longitude.
mlat, mlon: Geomagnetic latitude and longitude.
hgt: Altitude for conversion.

Outputs: glat, glon: Geographic latitude and longitude.
mlat, mlon: Geomagnetic latitude and longitude.
err: Error status of coordinate transformation.

Keywords: to_aacgm: Convert from geographic to AACGM coordinates.
to_geo: Convert from AACGM to geographic coordinates.
order: Specify the order of spherical harmonics used in the conversion.

4. AACGM_MLT

Description: Convert magnetic longitude to magnetic local time.

Calling Sequence: mlt=aacgm_mlt(epoch, mlong).

Inputs: epoch: CDF epoch time,
mlong: Magnetic longitude.

Outputs: mlt: Magnetic local time.

Keywords: mslong: Variable set to magnetic longitude of the sun.

5. AACGM_MLONG

Description: Convert magnetic local time to magnetic longitude.

Calling Sequence: mlong=aacgm_mlong(epoch, mlt).

Inputs: epoch: CDF epoch time,
mlt: Magnetic local time.

Outputs: mlong: Magnetic longitude.

6. AACGM_CONV_VEC

Description: Convert theta and phi component vectors from geographic to AACGM.

Calling Sequence: aacgm_conv_vec, glat, glon, hgt, gvth, gvph,
mlat, mlon, mvth1, mvph1, mvth2, mvph2, err, /to_aacgm.

Inputs: year: glat: array of geographic latitudes (-90< Lat < 90)
glon: array of geographic longitudes,
hgt: array of altitudes (km) for AACGM calculation
gvth: array of Theta component of vectors (geographic)
gvph: array of Phi component of vectors (geographic)

Outputs: mlat, mlong; arrays for the aacgm lats and lons
mvth1: aacgm theta vector component for a shift in geographic theta
mvph1: aacgm phi vector component for a shift in geographic theta
mvth2: aacgm theta vector component for a shift in geographic phi
mvph2: aacgm phi vector component for a shift in geographic phi

Further notes on aacgm_conv_vec (C. L. Waters)

Given the locations (lat, lon) in geographic (GEOG) coordinates and the associated vector components, it might appear strange that the routine returns the associated AACGM locations (mlat, mlon) plus four output arrays. The basic reason is the fact that an orthogonal shift in the geographic coordinate system does not result in an orthogonal shift in AACGM. In fact, for an orthogonal vector in geographic, the angle between the mapped components in AACGM varies with position. We are dealing with a non-orthogonal coordinate mapping. The mathematics of the geometry involves contravariant and covariant components with associated Jacobians, that vary with position.

The four output vector arrays provide the user with the choice of components to use. There are three cases:
Case 1:
Map a theta vector component in GEOG (Lat, 0) to AACGM to give some (mlat, mlon) vector in AACGM. Then calculate R cross this AACGM vector to get the AACGM Phi component. This will be an orthogonal vector pair in AACGM. In terms of the routine outputs the vector arrays are (mth_vec_gth, mph_vec_gth).

Case 2:
Map a phi vector component in GEOG (0, Lon) to AACGM to give some (mlat, mlon) vector in AACGM. Then calculate this AACGM vector cross R to get the AACGM theta component. This will be an orthogonal vector pair in AACGM (but different components to Case 1). In terms of the routine outputs the vector arrays are (mth_vec_gph, mph_vec_gph)

Case 3:
Map a theta vector component in GEOG (Lat, 0) to AACGM to get a AACGM (mlat, mlon)_dth vector. Map the orthogonal phi vector component in GEOG (0, Lon) to AACGM to get the AACGM (mlat, mlon)_dph vector. These two AACGM vectors will not be orthogonal. In terms of the routine outputs the output vector arrays are (mth_vec_gth, mph_vec_gph).

Warning: If you wish to perform the usual orthogonal operations (e.g. vector magnitude calculations based on theta^2+phi^2 components) DO NOT use Case 3.

[Please email colin.waters@newcastle.edu.au if you discover any problems with this routine]

License:

The IDL AACGM DLM is BEERWARE. If use it, like it, adore it, or even worship it, buy me a beer. ☺

History:

v1.0: Initial release.

v1.1: Give error message if both conversion directions specified in AACGM_CONV_COORD.

v1.2: Height parameter in AACGM_CONV_COORD is now vector. The AACGM_MLT routine has been vectorized.

v1.3: Functions renamed to idl_aacgm for consistency with other packages.

v1.4: Fixed clean-up problem with height_in variable in aacgm_conv_coord.

v1.5: Added AACGM_MLONG and error checking of input parameters.

v2.0: Project switched to Visual Studio 2005. New AACGM source code v1.09.

v2.1: Added AACGM_HELP procedure.

v3.0: Renaming IDL32.DLL to IDL.DLL in IDL 6.3 broke the old DLM code. To remain current, the project is switched to IDL 6.3. Thus v2.1 is the final version for IDL versions <=6.2 unless RSI finds a workaround for the problem.

v3.1: Error message for missing coefficient file changed to indicate name of missing file.

v3.2: Output array dimensions now identical with input array dimensions.

v3.3: Output values are returned as scalars if the input values are scalars.

v3.4: Added QUIET keyword to AACGM_LOAD_COEF.

v3.5: Fixed crash if AACGM_DAT_PREFIX environment variable not set.

v3.6: Changed environment variable to AACGM_PATH. Added routine AACGM_SET_PATH to reset the path to the coefficient files.

v3.7: Updated AACGM_HELP routine.

v3.8: Added routine AACGM_CONV_VEC to convert vectors from GEO to AACGM.

v3.9: Updated help routine to include AACGM_CONV_VEC.

v4.0: Version number adjusted to sync with UNIX version. A bug was fixed in the UNIX DLM, where a segmentation fault resulted when an IDL session was reset after changing the AACGM_PATH environment variable.

v5.0: Uses updated routines from Rob Barnes' RST package. AACGM_MLT and AACGM_MLONG now use CDF epoch time instead of year and seconds of year.

v5.1: Replaced EPOCH_F Fortran code with C code.