HMI Global Helioseismology Data Analysis Pipeline

Tim Larson
tplarson@sun.stanford.edu

Hansen Experimental Physics Laboratory
Stanford University

The HMI global helioseismology data analysis pipeline is based largely on the MDI medium-l program. All of the modules that ran in the SOI Science Support Center have been ported for use in the JSOC and given greater functionality. Many errors and approximations which are present in the standard MDI pipeline have been corrected and improvements have been added. Scripts have been written to automate the submission of compute jobs to our local cluster; it is now possible to go from dopplergrams to mode parameters with the push of a button. JSOC dataseries have been created to hold all intermediate data products, timeseries, window functions, and mode parameters. Modules and scripts have the capability to save processing metadata in an ancillary dataseries which will enable the automatic regeneration of data that has aged off disk.

Modules

- jv2helio – remaps dopplergrams to heliographic coordinates. input can be of any resolution, output are maps equally spaced in longitude and sin latitude.
- jhelio2mlat – performs fourier transform in longitude and transposes data.
- jqdotprod – performs projection onto spherical harmonics. output are timeseries chunked in l.
- jv2ts – combines the above 3 modules to save on I/O and is the one actually used.
- jv2hmod – jv2helio modified to output dopplergrams corrected for distortion and interpolated to any resolution. useful for making HMI-like data from MDI data.
- jretile – input and output are timeseries, changes tiling in time and chunking in l.
- jtsfiddle – detrends and gapfills timeseries. output can be timeseries, fourier transforms, power spectra, etc.
- jpkbgn – extracts mode parameters from fourier transforms. currently input are timeseries, output are ascii tables.

What's Available

- Source for all modules, build tested with intel compiler.
- JSOC dataseries definitions for all dataseries used by modules and scripts.
- Shell scripts to automate spherical harmonic transforms, retiling, detrending/gapfilling, and peakbagging.
- Parameter file templates used by scripts for processing options.
- All of the above are available online in our CVS repository.
- The contents of all JSOC dataseries (data products) can be browsed online at http://jsoc.stanford.edu/ajax/lookdata.html

To Do

- Remove IDL from scripts where appropriate, likewise replace shell scripts with perl scripts.
- Measure load and run times to determine how to store data. Floats are likely, but we also have the option of using scaled ints.
- Automate the detection of gaps and discontinuities in timeseries. We hope to derive this information from the housekeeping data, if not from QUALITY keyword.
- Write a module to simulate MDI vw_V data from HMI dopplergrams, as well as arbitrary binning.
- Write script to do inversions for sound speed and internal rotation.
- Leakage matrix, needs measurement of PSF from HMI.
- Document all modules and scripts.

More Codes

- Ridge fitting - Rabello-Soares & Korzennik
- MPTS & WMLTP codes - Rhodes & Reiter
- Long timeseries code - Korzennik
- GONG PEAKFIND

The WMLTP code, capable of fitting m-averaged spectra to above l=1000, is running in the JSOC computing cluster at Stanford. The MPTS code, capable of fitting tesseral modes to above l=1000, is still under development, but can run in the same software infrastructure as the WMLTP code. The ridge-fitting and long timeseries codes are functional and pending integration with JSOC. GONG PEAKFIND is waiting to be ported to C, after which it can also be ported to JSOC.

Tentative Processing Plan

- Run the first part of the pipeline once a day, producing 1 day timeseries with l=0 to l=lmax. Do not make any intermediate data products except by request. These may be archived.
- Every 36 or 72 days, retile the above timeseries into longer timeseries containing one l each. Detrend and gapfill these timeseries. These may be archived.
- Run peakbagging and whatever other algorithms are in the pipeline. Create fourier transforms or power spectra as needed. Archive mode parameters.
- Run inversions for sound speed and rotation.

FYI:
l=0-1500 => 1,127,251 m's
1 day = 1920 records at 45 sec cadence
storing floats => 17,314,575,360 bytes/day ~ 17GB/day
36 days = 612 GB