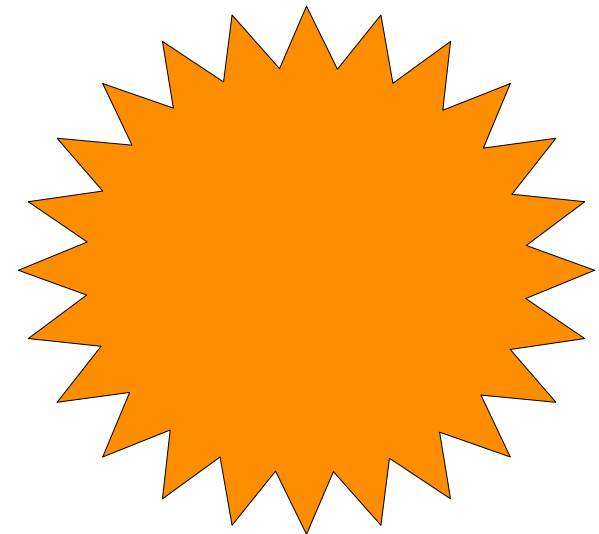
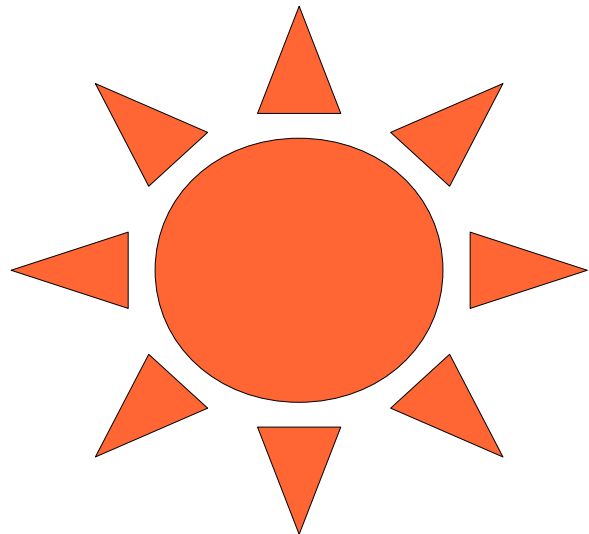


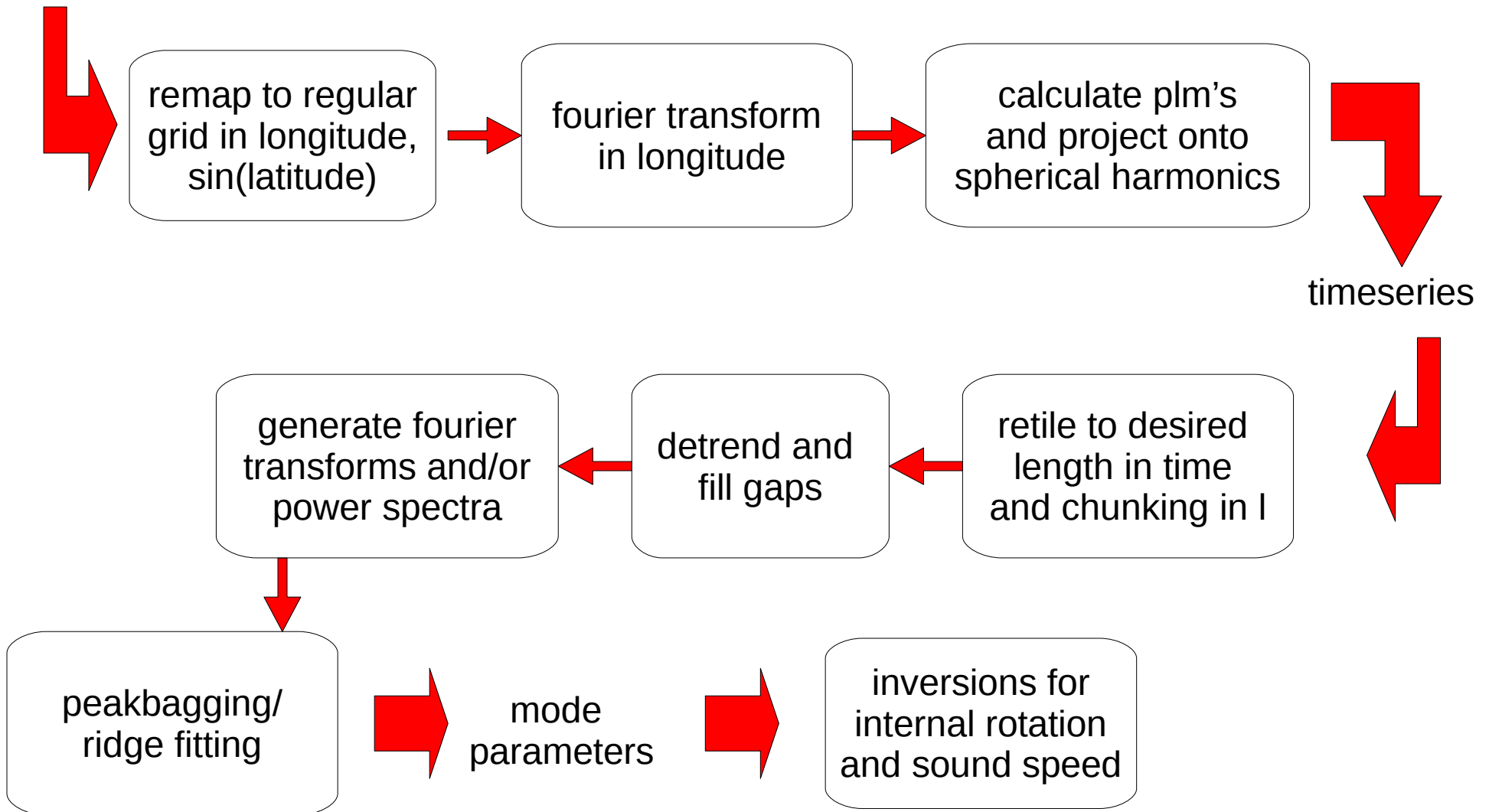
# HMI Global Helioseismology Data Analysis Pipeline

Tim Larson

[tplarson@sun.stanford.edu](mailto:tplarson@sun.stanford.edu)



dopplergrams



# Modules

ju2helio – 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.

ju2ts – combines the above 3 modules to save on I/O and is the one actually used.

juretile – input and output are timeseries, changes tiling in time and chunking in l.

ju2fiddle – detrends and gapfills timeseries. output can be timeseries, fourier transforms, power spectra, etc.

ju2slice – performs fourier transforms on a section of a timeseries to save on I/O.

ju2pkbgn – extracts mode parameters from fourier transforms. currently input are timeseries, output are ascii tables.

# Improvements

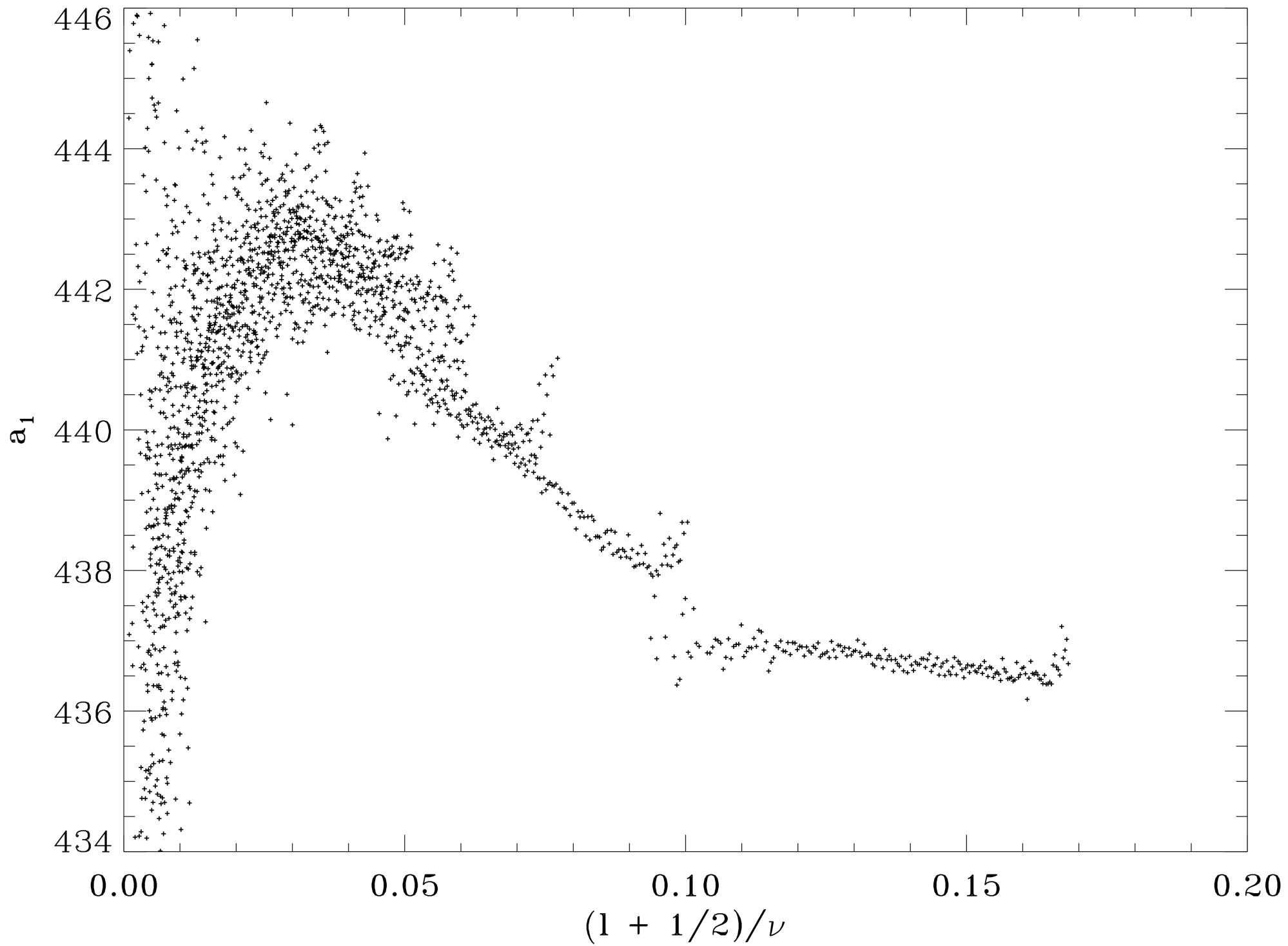
MDI: remapping takes into account change in plate scale, p-angle error, cubic distortion from optics, alleged CCD tilt.

HMI: dopplergrams already corrected for above, but may correct carrington inclination. new interpolation algorithms for remapping.

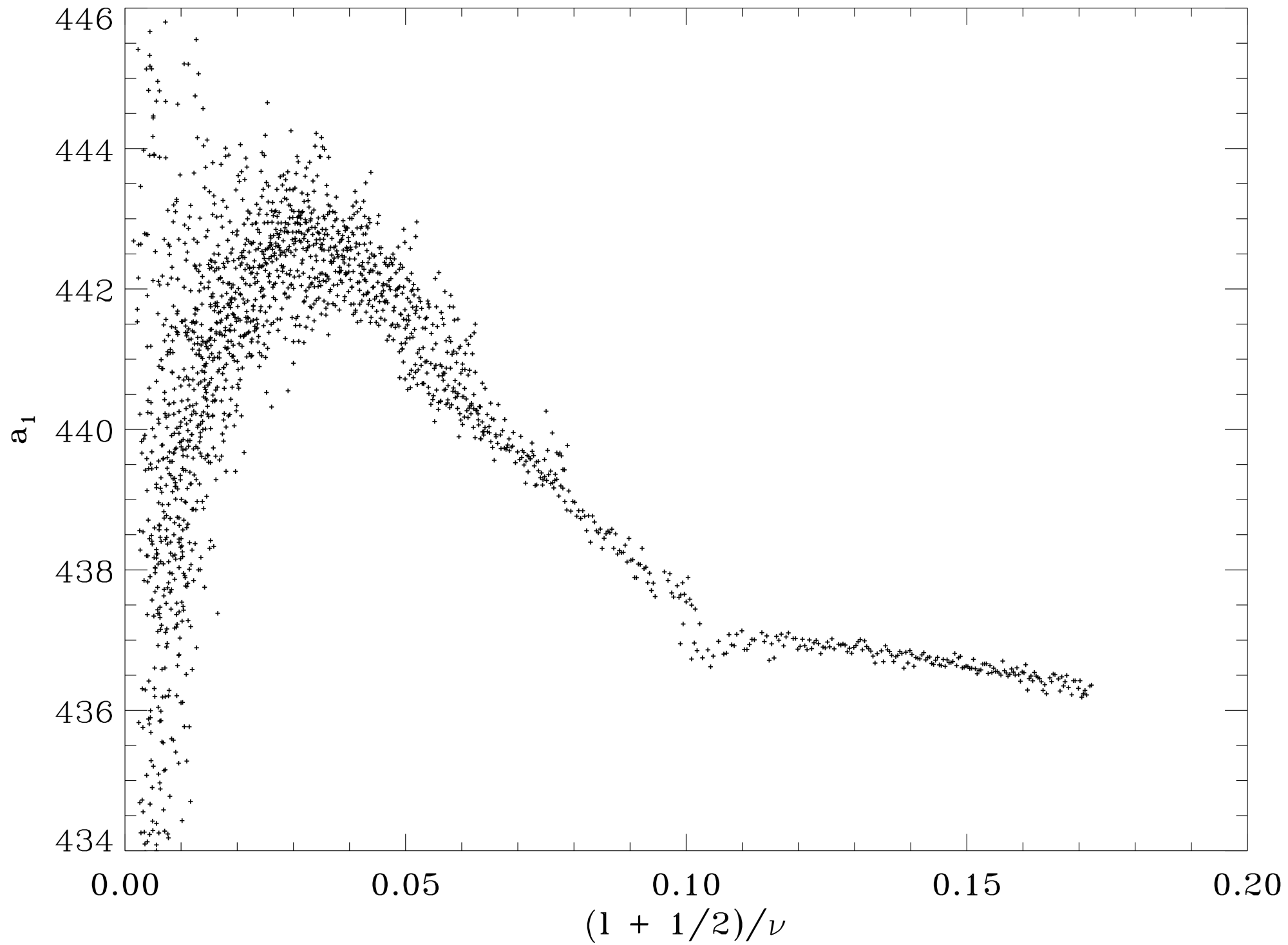
new gapfilling algorithm increases duty cycle and improves results at high frequencies.

peakbagging: fitting takes into account distortion of eigenfunctions by differential rotation, horizontal component of oscillations, and asymmetric line profiles.

# original



# all corrections



# What's Available

Source for modules, build tested with intel compiler.

JSOC dataserie definitions for all dataserie used by modules and scripts.

Shell scripts to automate spherical harmonic transforms, retiling, detrending/gapfilling, generation of power spectra, 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 dataserie (data products) can be browsed online at <http://jsoc2.stanford.edu/ajax/lookdata.html>

# Keywords

T\_START – the beginning of the time interval a record corresponds to. This will always be the first primekey of these dataseries. It can be specified as a date string or as an offset from an epoch, likely MDI epoch 1993.01.01\_TAI

NDT – number of time points represented.

T\_STEP – length of a time step in seconds, usually a constant. The length of the timeseries is then NDT times T\_STEP.

T\_STOP – the beginning of the following timeseries, or  $T\_START + NDT * T\_STEP$ .

T\_OBS – the midpoint of a timeseries as given by  $(T\_START + T\_STOP) / 2$ .

LMIN – minimum spherical harmonic degree represented.

LMAX – maximum spherical harmonic degree represented.

LMIN and LMAX will be the second and third primekeys when applicable.



# To Do

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 lev1.5 keywords, such as QUALITY.

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 improvements: correct for time dilation, difference in light travel time from different points on the disk, geometric error related to height of formation of spectral lines

# 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  $l=l_{\max}$ . 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:

$l=0-1500 \Rightarrow 1,127,251$  m's

1 day = 1920 records at 45 sec cadence

storing floats  $\Rightarrow 17,314,575,360$  bytes/day  $\sim 17$ GB/day

36 days = 612 GB