Computing Power Spectra from MDI and GONG Datacubes

Doug Braun

2005 July 26

This hands-on exercise is designed to lead the student through an examination of the general properties of Doppler time-series observations as sampled from the Global Oscillations Network Group (GONG), and the Solar Oscillations Investigation – Michelsen Doppler Image (SOI-MDI, or often simply MDI) an instrument onboard the NASA-ESA satellite *SOHO*. The instruments are described in Harvey et al. (1998) and Scherrer, et al. (1995). Additional information, as well as instructions on how to download data from either source may be found in the websites listed in the references.

It should be recognized that, although traditional helioseismology has placed great emphasis on the construction and analysis of power spectra there are many tools of helioseismology (particularly recent local techniques) which work directly in the spatial or temporal domains, as opposed to the wavenumber and frequency domains (and some use other combinations of these). In general, a familiarity of the properties of solar oscillations in the spatial, temporal, as well as wavenumber and frequency domains, is critical for a helioseismologist to develop.

How to use this guide:

This instruction guide has numbered sections which correspond to data products computed and displayed when running the numbered idl *batch jobs* found in your working directories. These jobs may be run without modification as follows (idl uses an ampersand before the filename to run a batch job):

```
IDL>@01_read_data
```

The tasks may also be carried by copying and pasting individual lines of the batch job into idl. Each job will perform various processing and display some of the results. Students are advised to view the batch jobs in an editor in another window to understand what is being done. A series of **questions in bold type** will be presented in each section. It will be necessary in some instances to edit, modify and rerun the batch jobs (or create new ones) to address some of the questions raised. Alternatively, this may be accomplished by working in interactive mode (entering individual or sequences of commands directly in idl). There are also stand-alone movie programs which can be run at any time, which use the idl `xinteranimate` interface. Appendix II lists and explains the software supplied with these exercises. IDL also has a built-in help system (enter '?' at an idl prompt). Students are also strongly encouraged to explore on their own, and satisfy their own curiosity.

Note that if you need to exit idl and restart (if for example, something goes wrong) you do not need to rerun all of the batch jobs to catch up to where you left off. After reading the data (`@01_read_data`) you may run any of the batch jobs in any order without problems (this is because many of the jobs perform redundant computations). However, if you wish to run the stand–alone movies (`z_movie` or `row_movie`) on the power spectra you will need to type `@07_power_spectra` first.

1 Reading in the data (general properties of the datacubes)

Running the batch job

```
IDL>@01_read_data
```

will read in two datacubes, called 'data1' and 'data2', as well as read in and display sample images from the MDI instrument taken at during the same interval (2002 March 31 15:00 – 23:32 UT). The datacubes will be the focus of the remainder of the exercises. However, for now, examine the sample Dopplergram, magnetogram, and continuum-intensity image from the MDI instrument. It is probable that you have seen similar looking images as the intensity image, and magnetogram. The Dopplergram shows the *line-of-sight* velocity of the photosphere at each pixel. A positive velocity indicates motion away from the observer. Notice that one side of the Dopplergram is darker than the other.

If the units of the Dopplergram are in m/s, and the Sun has a radius of 696 Mm, can you determine the solar rotation period?

Also notice the patterns of light and dark structures which increase in contrast towards the limb. This is *supergranulation*.

Why do you think the supergranulation is more visible near the limb in these Dopplergrams?

Can you see evidence of supergranulation in the continuum image or magnetogram? Try adjusting the minimum and maximum values of the `range` keyword in the call to `tvim` to increase the contrast.

2 A comparison of single image frames

The two datasets (data1 and data2) are three-dimensional arrays (read in from the original FITS format). One of these simultaneous datacubes is from the MDI instrument and the other is from the ground-based GONG network. Run the batch job

```
IDL>@02_compare_frames
```

to compare sample images from each of the datacubes, as well as displays a scatter plot of the velocity values between the two frames.

Can you guess which is GONG and which is MDI? How? If you cannot, more clues will accumulate along the way.

Both of the datasets are calibrated in units of m/s. However the scatter plot does not have a slope of unity, even though the Dopplergrams were taken at the same time.

Can you understand why the scatter plot looks the way it does?

The spatial area covered by the datacubes represent only a small 256 by 256 pixel subarray of the full disk images you displayed in exercise 1 . The pixels here are each .002 solar radii, or 1.39 Mm in size. The area is centered on a sunspot, and the datacube is tracked along with a constant solar rotation rate (known as the *Carrington* rate). Sunspots have outflows called *Evershed* flows.

Why does the Evershed outflow look the way it does here?

3 Acoustic signals

As you may know by now, the solar acoustic oscillations we are studying have periods predominately around five minutes. The images in the datacubes here are taken at a rate of one per minute. A quick way to "see" the oscillations is to subtract two frames taken about two minutes apart (**why is this?**). The next batch job (`@03_acoustic_signal`) displays both the average and the difference of two frames taken two minutes apart for data1. After running it as is and examining the results, modify this or copy to another batch file to display the same for data2.

Why does the Evershed outflow disappear in the difference image?

What is appearing in the difference image?

How do the average and difference images compare between datasets 1 and 2?

4 Time slices

Now let's see how the data varies with time (which is the third dimension of the datacubes). The 512 images in each cube, at a rate of one image per minute, represent a total span of 8.53 hours. The next batch job (`@04_time_slices`) displays how a given pixel row (or "y" value, in this case row 20) varies with time (vertical axis).

Can you see five-minute oscillations?

Can you guess what the vertical patterns are? (You may be able to answer this after completing the next exercise).

If you look closely you may see grey horizontal lines or bands in each image. These represent missing data, which in these datacubes have been replaced with zeros. If you have trouble seeing these gaps, try displaying a row in the middle of the sunspot (repeat batchjob 2 or 3 to identify a row there).

From what you know about the general properties and strategies of GONG and MDI, **Can you guess why there are more gaps in one of the datasets?** (The frequency of gaps shown here are typical for both sets.)

Now run a movie of the datacubes in time (the "z" dimension), using the provided program `z_movie` (see Appendix II). The other movie program, `row_movie` will sequentially show all x-z slices, incrementing the y (row) number through the image. Try these with the two datacubes. Note that the contrast can be increased by using the *sfact* parameter (Appendix II discusses the supplied software in more detail).

Bring up the time-slices again (batch job `@04_time_slices`). If you changed the row number, then set it back to 20 before running the next job.

5 Time averages

Run `@05_time_averages` which will compute the temporal average of the entire 8.3 hour sequences and display the results. This is performed by a supplied idl routine call `time_avg`. The two time-slices from exercise 4 should still remain on the screen.

Can you now identify the vertical dark features in the time-slices?

Can you estimate the typical velocity of supergranulation?

6 Subtracting time-averages: residuals

Run `@06_residuals` which will subtract the temporal average shown in the last exercise from each frame of data2. The residual images are in a cube called data2res. The batch job will display three images; an original "raw" frame, the "residual" frame (with the mean subtracted), and an image showing the variance of the residual signal over time. Look first at a comparison of the raw and residual frames.

Does the residual image still show the Evershed flow?

Are the fluctuations of the remaining signal in the residual image random over the entire frame? The third frame shows the variance of the residual signal over time, which is the sum of the square of the residual images. Notice that the batch job uses the same routine `time_avg` to compute the. Notice the appearance of the variance in the sunspot and around the frame.

Where is the variance smaller (darker) then usual?

It is well known that acoustic waves have suppressed amplitudes in magnetic fields. You may want to bring up the magnetogram again (see `01_read_data` – you may have to adjust the range in `tvim`). The variance image you are looking at is a simple version of an "acoustic power map", (e.g. Braun et al. 1992; Toner & LaBonte 1993; Hindman & Brown 1998), although it includes other contributions and artifacts from other things besides *p*-modes.

Try to identify the sunspot and the surrounding region, by looking for a correspondence between the suppressed variance signal and the magnetic fields.

Can you guess what the cause of the enhanced (whiter) regions around the sunspot in the variance image?

To answer the above, you may want to run the movie of the residual datacube, datares2, and the original datacube, data2, again (be sure to adjust the sfact to increase the contrast if necessary).

Does the sunspot remain in the same place for the 8 hours?

The computation of helioseismic signals from datacubes spanning several hours is routine and often necessary for collecting sufficient statistics. During this time, solar features may evolve and change position. Coping with this is a fundamental challenge for many research goals in local helioseismology, as some unwanted effects can be reduced but not all. Examine and compare the temporal changes in the Doppler signal for single pixels, in and out of various regions, using simple idl commands like `plot` and `oplot` (see the online help or ask).

Can you think of ways to reduce some of the unwanted (i.e. non-acoustic) signal in the variance image?

The removal of unwanted background signals (both in space and time) is common in the reduction and analysis of helioseismic data. In the case of temporal variation of solar features, a larger issue remains:

How much can these or other corrections really substitute for a Sun with no temporal evolution?

In general corrections or adjustments, like filtering, smearing, or subtracting slowly varying functions, are often peformed by trial and error – there are no fixed rules. If you do much data analysis, you should always explicitly state what you have done. With corrections that are critical to achieving your results, be prepared to justify them and explore the effects with different (or no) corrections.

7 Computing power spectra

Run `@07_power_spectra` which will recompute the residuals of the datacubes and their three-dimensional power spectra. The batch job simply computes the spectra (called pow1 and pow2), as well as the azimuthally summed spectra (called knu1, and knu2; these are discussed more below). It is entirely up to you to view and explore the 3-D spectra. You may use `z_movie` and `row_movie` (see appendix) to display sequential slices of (and interactively slide through) various dimensions of the power spectra, or `tvim` to display individual slices with the ability to quantify the scales and dimensions (which is not part of the xinteranimate framework). Read the appendix to understand the axes of the spectra, as well as the symmetry that you observe as you run the movies. A classic paper discussed similar three-dimensional power spectra of solar oscillations in analogies of “rings” and “trumpets” (Hill 1988)

Can you see the rings and trumpets in the power spectra?

How high in frequency or wavenumber can you see them?

(Note you can convert the pixel number from the movie software to actual wavenumbers or frequencies, using formula in the appendix.) The dimensions of these datacubes are similar to those routinely used in *ring-diagram analysis*, which is widely used form of local helioseismology (e.g. Hill 1988; Patrón et al. 1995; Haber et al. 2002). Perhaps now you see where this term comes from and can develop a feeling of what ring-diagram analysts have to work with.

What do you see at low temporal frequencies (around the middle of the frequency range)? What do you think this might be?

Finally, display using tvim the azimuthally averaged power spectra (which is defined in Appendix I), which are called knu1 and knu2. You will probably want to adjust the range of the greyscale. Put labels and span the xrange and yrange to cover the full range in the images. Put the units in ℓ and ν.

Now do you know which dataset is which?

In one of the datasets, there appears a feature at a frequency of around 1.3 mHz and at high spatial wavenumbers. See if it is present in both datasets.

What does this tell you about whether the feature is solar?

Often features ("artifacts") appear in the power spectra which are not solar in origin. They may sometimes be instrumental, but frequently are due to effects of the data analysis themselves. Go back to the 3-D power spectra and look for this feature. Notice anything unusual about it. From the temporal frequency of this feature, estimate a corresponding temporal period.

How much does the Sun rotate in this time? Can you guess what this might be?

Now let's compare these observations with some theory. We know the the *fundamental*, or f-mode, has a simple dispersion relation, given by

$$\omega^2 = g k_h \tag{1}$$

where ω is the temporal frequency and k_h is the horizontal wavenumber as defined in Appendix I. The gravitational acceleration g at the solar surface is 2.74 x 10^4 cm/s^2.

Overplot the dispersion relation for the f-mode in the correct units of your $\ell - \nu$ plot.

Use your idl skills to accomplish this, or consult the manual or others. (Hint: you will need to generate a set of ℓ values with `findgen`, and use `oplot` to overlay your derived function of ℓ on the output of `tvim`.)

Now admire both $\ell - \nu$ diagrams, and all you've accomplished. You may want to look up (on the web now, or someday soon) some of the $k - \nu$ diagrams published in the early days of helioseismology (e.g. Deubner 1975; 1983). Reflect that these datacubes represent only 8 hours of observations of a small segment of the visible solar disk. Imagine what might be accomplished with decades of continuous full-disk high-resolution Dopplergrams. Congratulations! You are now a seasoned helioseismologist!

Data web sites (where to browse and download data)

Global Oscillations Network Group http://gong.nso.edu

The Michelson Doppler Imager http://soi.stanford.edu

References

Bracewell, R. 1986, *The Fourier Transform and its Applications,* (New York: McGraw-Hill)

Braun, D. C., Lindsey, C., Fan, Y., & Jefferies, S. M. 1992, "Local acoustic diagnostics of the solar interior," ApJ, 392, 739

Deubner, F.L. 1975, "Observations of low wavenumber nonradial eigenmodes of the sun," Astron. & Astrophys. 44, 371

Deubner, F.L. 1983, "Helioseismology with high degree p-modes," Solar Phys., 82, 103

Haber, D. A., Hindman, B. W., Toomre, J., Bogart, R. S., and Larsen, R. M. 2002, "Evolving Submerged Meridional Circulation Cells within the Upper Convection Zone Revealed by Ring-Diagram Analysis," ApJ, 570, 855

Harvey, J., Tucker, R., & Britanik, L. 1998, "High Resolution Upgrade of the GONG Instruments," in *Structure and Dynamics of the Interior of the Sun and Sun-like Stars SOHO 6/GONG 98* ,eds. S. G.

Korzennik and A. Wilson, **ESA SP-418**, (Noordwijk: ESA), 209

Hill, F. 1988, "Rings and trumpets - Three-dimensional power spectra of solar oscillations," ApJ, 333, 996

Hindman, B. W., & Brown, T. M. 1998, "Acoustic Power Maps of Solar Active Regions," ApJ, 504, 1029

Patrón, J., Hill, F., Rhodes, E. J.,Jr., Korzennik, S. G., and Cacciani, A. 1995, "Velocity Fields Within the Solar Convection Zone: Evidence from Oscillation Ring Diagram Analysis of Mount Wilson Dopplergrams," ApJ, 455, 746

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P 1992, *Numerical Recipes in C,* (Cambridge: Cambridge University Press)

Scherrer, P. H. et al. 1995, "The Solar Oscillations Investigation - Michelson Doppler Imager," Sol. Phys., 162, 129

Toner, C. G., & Labonte, B. J. 1993, "Direct Mapping of Solar Acoustic Power,"ApJ, 415, 847

Appendix I

Three-dimensional power spectra

Over the sphere of the Sun, global acoustic, or p-mode oscillations are traditionally decomposed into functions proportional to spherical harmonics described by integer quantum numbers. For much of *local helioseismology*, however, we consider local regions of the Sun and waves of sufficiently small wavelengths compared to the size of the region studied, such that the use of Cartesian coordinates is appropriate. Considerable advantages are to be gained by this approximation; for example some properties of the waves may be extracted by the use of fast Fourier transforms. The general properties of Fourier transforms and power spectra are discussed elsewhere (e.g. Bracewell, 1986; Press et al. 1992). This appendix briefly discusses the properties of the Cartesian power spectra computed in these exercises using the `compute_power` routine.

The discrete Fourier transform V of a one-dimensional function v of the coordinate x sampled at discrete locations $x_n = n\delta x$, where n is an integer and δx is the sampling interval (e.g. pixel size), is defined by

$$V(k) = \sum_{n=0}^{N_x-1} v(n\delta x)e^{-ikn\delta x}. \tag{2}$$

N_x is the number of discrete values of x_n. The wavenumber k is typically sampled at discrete values

$$k_i = \frac{i2\pi}{N_x\delta x}, i = -\frac{N_x}{2}, ... \frac{N_x}{2}. \tag{3}$$

The Fourier transforms are computed by means of FFTs (Fast Fourier Transforms). In this exercise, we will not be viewing or saving the Fourier transforms directly, but instead looking at the power spectra. The one-dimensional power spectra P of v is given by

$$P(k) = |V(k)|^2 \tag{4}$$

The extreme values of i above correspond to the lower and upper limits of the Nyquist critical sampling range (See Press et al. 1992), and the highest spatial wavenumber is given by the Nyquist wavenumber

$$k_{nyq} = \frac{\pi}{\delta x} \tag{5}$$

With two spatial and one temporal dimensions, there are three discrete Fourier transform operations performed, analogous to Eq. 2 with the resulting power spectra given by

$$P(k_x, k_y, \omega) = |V(k_x, k_y, \omega)|^2 \tag{6}$$

where each spatial wavenumber k_x and k_y take discrete values analogous to Eq. 3:

$$k_{x,i} = \frac{i2\pi}{N_x\delta x}, i = -\frac{N_x}{2}, ... \frac{N_x}{2}, \tag{7}$$

$$k_{y,j} = \frac{j2\pi}{N_y\delta y}, j = -\frac{N_y}{2}, ... \frac{N_y}{2}. \tag{8}$$

The formalism above allows for different sampling intervals in the two spatial dimensions (e.g. rectangular pixels), but the data used in these exercises has equal sampling intervals in x and y, so there is an identical set of wavenumbers for these dimensions. The temporal frequency ω has a set of discrete values, set by the temporal sampling δt (here, 60 sec):

$$\omega_q = \frac{q2\pi}{N_t\delta t}, q = -\frac{N_t}{2}, ... \frac{N_t}{2}. \tag{9}$$

For labeling solar oscillation spectra, observers (as opposed to theorists) often use the cyclic frequency ν which is related by

$$\nu = \omega/2\pi. \tag{10}$$

What is the temporal Nyquist frequency in terms of the temporal interval δt ?

Take particular note of the range of wavenumbers and frequencies given by Eqs. 3 and 9. The index of the three-dimensions of the power spectra computed by `compute_power` sequentially span these ranges *exactly*. This is not always the case with other FFT or power spectra routines, which sometimes start with zero wavenumber (or frequency), and cycle through the positive, and then negative wavenumbers (or frequencies). In fact, `compute_power` is a wrapper around an internal idl FFT routine, and rearranges the output specifically to accommodate this format.

You will see from this exercise, that solar oscillation spectra exhibit particular spatial and temporal symmetry in the (spatial) wavenumber and (temporal) frequency domains. One thing you should take particular care to note is the *azimuthal* symmetry. In fact, there are properties in the solar interior which can break this symmetry, slightly, and these are intensely studied by helioseismologists. However, the overall azimuthal symmetry is so pronounced that it is customary to construct two-dimensional power spectra by summing the the power in rings in the (k_x, k_y) plane, such that we define a total *horizontal wavenumber*

$$k_h = \sqrt{k_x^2 + k_y^2}, \tag{11}$$

and look at azimuthally summed power spectra as a function of k_h and ν. Helioseismologists who look at local, or high k power spectra, may refer to azimuthally summed power spectra as $k-\nu$ diagrams. In some ways, these are similar to so-called m-averaged power in the $\ell-\nu$ spherical decomposition of the mode power over a sphere. We can even connect the two types of spectra in the limit of high k_h, by expressing the horizontal wavenumber in terms of the degree ℓ of a spherical harmonic in global coordinates. The relationship is approximately

$$k_h \approx \ell/R_\odot \tag{12}$$

where $R_\odot$ is the solar radii (696 Mm). **What is the ℓ value of the spatial Nyquist frequency in terms of the pixel size in units of solar radii, $\delta x/R_\odot$?**

Appendix II

Supplied IDL software

`aspect`
Used by some plotting routines (e.g. `scatter` below) to set the aspect ratio. Not used by itself in these exercises. Written by David Fanning (see `http://www.dfanning.com/`).

`color_key`
Used by `tvim` to draw numbered color scale. Not used by itself in these exercises. Written by Paul Ricchiazzi (see `tvim`).

`compute_power,filename,input,azimuth_sum_power,power`
Computes the power spectra of a 3D real function (input). Returns the 3D power spectra (power) and a 2D array representing the power summed over rings in the k_x, k_y plane (`azimuth_sum`). Written by Juwei Zhao for use by the summerschool.

`readfits,filename,array [,/silent]`
This and many other FITS image reading/writing/manipulating routines are found in the `idl_fits` subdirectory. We will primarily use the `readfits` routine which reads a multi-dimensional image (filename) stored in FITS (Flexible Image Transport System) format into an idl array (array). The `/silent` keyword suppresses some (possibly informative) status messages of the read operation. See the README file in the

directory for more information on these useful routines. Part of the IDL Astronomy User's Library (see `http://idlastro.gsfc.nasa.gov/homepage.html`).

`row_movie,array,[sfact=` *num*`]`
Like `z_movie` below, but shows sequential slices in the x (column) and z (time or frequency) domain as the y (row) index is incremented. This takes the same arguments as `z_movie`. Written by Doug Braun for use by the summerschool.

`scatter,im1,im2,[title=`*title*`,xtitle=`*xtitle*`,ytitle=`*ytitle*`]`
A scatter plot of two arrays (of arbitrary dimension, but 2-D in the example used in these exercises). It uses the aspect routine (see below) to fix equal x and y scales, set by the extreme minimum and maximum of the two arrays. Takes optional arguments for titles. Written by Doug Braun for use by the summerschool.

`subtract_avg,input,output`
Takes a 3-D datacube (input) and computes and subtracts the average over the third dimension. Does not perform the subtraction for input frames that are all zeros (e.g. representing data gaps). Does not return the mean average as a separate array (see `time_avg` which does). Written by Doug Braun for use by the summerschool.

`time_avg,input,output`
Takes a 3-D datacube (input) and returns the average (output) over the third dimension. Not to be confused with `subtract_avg`. Written by Doug Braun for use by the summerschool.

`tvim,image,[title=`*title*`,xtitle=`*xtitle*`,ytitle=`*ytitle* `/scale, range=[`*num1*`,`*num2* `],`
`xrange=[`*x1*`,`*x2*`], yrange=[`*y1*`,`*y2*`]]`
Displays a grey scale of an image with optional titles, and a color bar if the parameter `/scale` is set. The range of the grey scale is set by the range keyword. Normally, tvim displays the x and y scale in pixels. This can be calibrated to real units by specifying the xrange and yrange limits. This routine is written by Paul Ricchiazzi and is part of the Earth Space Research Group (ESRG) UCSB library (see `http://www.astro.washington.edu/deutsch/idl/htmlhelp/library28.html`)

`transform,input,output`
Sorts the output from the idl FFT routine so that the output indices range from negative to positive wavenumbers (or frequencies). Used by `compute_power`, but not used by itself in these exercises. Written by Juwei Zhao for use by the summerschool.

`z_movie,array,[sfact=` *num*`]`
Runs a movie (as a wrapper to the idl "xinteranimate" routines) of a 3-D array, sequentially displaying each 2-D frame as the third dimension is incremented. For the datacubes in these exercises, `z_movie` shows the data frames in time. For power spectra, it shows sequential slices as the temporal frequency is incremented. `z_movie` takes only two parameters, the data array (which must be a 3-D array), and an optional scale factor, sfact, which can be used to increase the contrast. Setting sfact=1 (the default) sets the scale to the minimum and maximum of the array; setting sfact to a number less than one multiplies the minimum and maximum by this factor and will increase the contrast. Written by Doug Braun for use by the summerschool.