Solar Physics Group, Department of Mathematics, Physics and Electrical Engineering,
Northumbria University

CoMP Wave Tracking Code Document

Richard Morton*

July 9, 2019

Version 1

* Contact: richard.morton@northumbria.ac.uk

History

Version 1 - 05/2019 - Basic operating information

If you spot any mistakes or suggestions please email *richard.morton@northumbria.ac.uk* .

1 Introduction

This document contains information on how to use the Coronal Multi-Channel Polarimeter (CoMP) Wave Tracking Code (WTC), which is designed to align CoMP L2 data products (core intensity, Doppler Velocity and Doppler width) and fill in any missing frames to produce evenly sampled time-series. The CoMP Doppler velocity data is ideal for the measurement of coronal Alfvénic waves. The WTC can calculate the propagation orientation of the waves to produce a wave angle map - this has been demonstrated to show good agreement with the plane-of-sky magnetic field orientation derived from CoMP polarimetric measurements. Using the wave angles, the WTC can then make measurements of the propagation speed of the waves throughout the corona. This is useful for magneto-seismology if CoMP has also taken measurements in the 10798 Å line, whose ratio with the 10747 Å line is sensitive to density - hence there is the possibility to provide some measure of the coronal magnetic field.

We note that the current version of the code is unix centric! Certain features will not work with a windows machine. Let us know any problems and we can try to fix (or if you can provide a fix, even better!).

1.1 Required files

Current working version - v0.3

1.1.1 Main files

wave_tracking.pro, find_man_date.pro, comp_load_files.pro, wave_angle_calc.pro, compute_speed.pro, dejitter.pro, comp_config.pro

1.1.2 Minor files

do_apod.pro, plot_wave_maps.pro

1.1.3 External files

fg_rigidalign2.pro, tr_get_disp_2b.pro, shift_img.pro
uses routines in ssw/gen

2 Initial configuration

The majority of hard-coded variables in the WTC are located in the *comp_config.pro* file. It is suggested that these are not altered unless you are sure you know what the consequences are!

The only variable that will initially need altering is your personal directory for CoMP data analysis (root_dir). The folder system assumes that you have L2 data located at 'root_dir/CoMP/year/month/day/' and that there is a folder for data products located at 'root_dir/CoMP/wave_tracking_output/'.

3 Initial data load

The main routine is *wave_tracking* which will read in L2 data from fits files, and produce cubes of intensity, velocity and Doppler width with a functional index structure and saves in idlsave files.

The L2 fits files should be kept in folders with format 'CoMP/year/month/day/' where year, month, day are numbers. At present, the path to the CoMP folder is hard-coded into the *wave_tracking* file and needs to be edited to the user.

When working with L2 files for the first time, you need to make at least the following basic call:

IDL>date='20120327' ; year/month/day
IDL>wave_tracking, date, /init_load

The routine calls comp_load_files, which is designed to read in fits files, works out cadence (generally 30 seconds) and fill in any gaps (due to missing frames) via linear interpolation. There is no maximum to the number of contiguous missing frames that can be filled. However, any more than 2 or 3 is likely to contain to much 'imaginary' data, and will probably require some more advanced approach, e.g., maximum entropy method. Also created is the basic

index file from information in the fits headers (note this index does not follow the standard header fits notation). The index is furnished with additional information related to data and it its processing with the WTC.

As part of this stage, we also mask out some of the pixels based on certain conditions. First we calculate the median value of each pixel in the intensity cube from all time-frames. Any pixel whose median is zero will have more than 50% of time-frames with no signal. When then apply a spatial median filter to the time-median intensity image, any pixel in the resulting image with a zero value of intensity is then isolated from pixels with signal. Any pixel that have zeros in both these median filters is considered poor quality and masked out. This process removes a significant number of the pixels with poor quality time-series.
A final filtering step is performed. The intensity time-series of each pixel that has passed the first stage of filtering is then examined, and the number of zero values it contains is calculated. If this value is greater than the number of missing frames then the time-series is considered poor quality and masked out. This final stage of filtering can be turned off by using the */no_hard_mask* keyword.

As mentioned, the default gap filling procedure is linear interpolation. If you are wanting to fill gaps longer than 2 or 3 contiguous frames, then a more advanced technique is required. There is the option to implement maximum entropy gap filling (reference). This can be turned on via the */max_ent* keyword. The time-series of each pixel is examined in turn and the maximum entropy calculation is undertaken. Hence, this process is slower than the linear interpolation but is still reasonably quick. At present, the number of coefficients used in the calculation can be set in the *find_man_date.pro* file. In theory, it should be possible to apply a model selection criteria, e.g. AIC, to find the 'best' number of coefficients - however, there are some issues with this. It is likely that the ideal number of coefficients for intensity, velocity and Doppler width are different, given the different nature of the time-series. At present, a single value is used for all. Hence, it is suggested that some initial tests are performed with the time-series to find the most suitable value.

The data is then saved to 'CoMP/wave_tracking_output/' - the folder will be automatically generated if it doesn't exist.

The basic call given above can be furnished with additional keywords and calls to achieve various tasks (see below).

4 Aligning data

The data cubes can be aligned via an FFT-based implementation that measures shifts in intensity images. If the keyword /cross_corr is used then data cubes will be aligned via 2d FFT cross-correlation (CC), with the neighbourhood of the CC peak fit with a surface to find sub-pixel accuracy for CC (tests have shown this accuracy is comparable to Taylor expansion techniques 0.1 pix). The wrapper calls dejitter.pro for this, which is requires fg_rigidalign2.pro and tr_get_disp.pro (both of which are alignment routines developed by A. De Wijn and T. Tarbell respectively, modified by R. J. Morton). The alignment process CC's four contiguous frames and calculates alignment vectors for them. The next four frames are also self aligned and then additional vectors are calculated to align to the last frame of the previous sequence. The alignment is carried out by with shift_img (T. Metcalf) using poly_2d.
The alignment process repeats itself on the aligned images multiple times to achieve the best alignment. The process is repeated until returned frame-to-frame shifts are less than a threshold value or a maximum number of iterations have been performed (both of which can be set in find_man_date).

If you have already performed an initial load of the data, you can restore and align the data with the following:

IDL> wave_tracking, date,/cross_corr,/choose_corr_box

Using /choose_corr_box allows a manual selection of the CC box through an interactive window - normally useful when examining data for the first time.

It is advised to use the /dosob keyword if doing alignment, which uses the Sobel gradient filter to calculate 2d gradients in the image. The intensity gradient turns out to provide better contrast and improves cross-correlation.

IDL>wave_tracking, date,/cross_corr,/choose_corr_box,/dosob

The choose_corr keyword enables a window-based point and click method for choosing a region that will be used for correlation. From experience, you will probably have to modify the region used for cross-correlation by hand. A well-chosen region will only take a few iterations (< 20) of the alignment to reach the threshold (if using the Sobel filter). The coordinates of the region used for cross-correlation can also be entered in find_man_date.

Note: All the additional keywords can be given during the initial load stage.

4.1 Wave Angle Calculation

IDL> wave_tracking,'20120327',/compute_waveang

4.2 Computing propagation speeds

IDL> wave_tracking,'20120327',/compute_speeds

Text below here needs to be sorted.
Also calculates the Alfvenic wave propagation angle from velocity data.

5 find_man_date

First call is to find_man_date . This is essentially a large list of dates for which data sets have already been processed and processing values are known (e.g., correlation box). The information is stored in a short section labelled by date,
'20120327': begin

config.coordinatesCCBox=[[55,220],[85,260]] ; Cross-correlation box [x1,y1],[x2,y2]

config.startFile = 0 ;fits file to start with

config.numberFilesToProcess = 164 ;Number of fits files to use

config.lowerMaskRadius = 228. ;Upper and lower radial indices that marks FOV

config.upperMaskRadius = 280. ;boundaries for CoMP

end

Undefined dates are given default values which are probably no good!

6 text that needs working on

With /debug set, .
After alignment the routine stops so that you can play movies of the data cubes to check quality of alignment procedure.
Type .c to continue.
Basic pixel mask is defined based on inner and outer radius of occulter. This is not waste computational time on pixels with no signal.
The next stage is to calculate the wave propagation angle. This is done as default. The velocity data is mean subtracted and FFT'd. The transformed data is then sent to wave_angle_calc.pro for processing.
Wave Angle Calculation Starting at a pixel, a 40 by 40 is generated around the pixel. Each time-series in then cross-correlated in Fourier space with the central pixel to calculate the cross-spectral coherence. The results are filtered in frequency space, focusing on a narrow frequency band centred on 3.5 mHz with a width of 1.5 mHz (is this the best choice?) in order to select the power enhancement seen in power spectra (e.g., Tomczyk & McIntosh 2009; Morton et al 2016). If a time-series has a coherence of gt 0.5 with the central pixel, it is saved. Otherwise, it is rejected.
Not entirely sure if this best selection criteria for coherence, seems arbitrary but works well enough.
If there is enough coherent points (¿10) then the a straight line is fit to the minimise the perpendicular distance between all points. This is an analytic solution calculated under the constraint that the line does through the central pixel. Error on angle calculated using analytic perturbation by pm 2 degrees (not sure where this comes from).
No account for quality of points is considered - may benefit from applying some rejection criteria, e.g, double fitting process or weighting based on coherence. Calculation of error on angle may or may not be appropriate. Also, unclear how this performs near occulting disk with half empty boxes!

A measure of coherence 2 dimensionally is also calculated by fitting ellipses to the data points, assuming a coherent island of points. Returning the length of the major and minor axis.
Once performed for every pixel inside mask, returns to main programme.
The elliptical coherence measure is used to define a new mask, that removes pixels with no significant coherence to at least 10 other pixels.

7 Keywords

7.1 wave_tracking

Keyword related to reading/writing in data

init_load - use this keyword for initial data load and gap filling. This keyword is required for first load of data. If not provided, default is to look for existing, gap-filled data (saved as cube_ivw_date).

frameLimitInterp - sets maximum allowed number of missing frames (gaps) in a row, default is 2. Default gap filling method is linear interpolation.

max_ent - Uses maximum entropy method for gap filling

no_hard_mask - turns off the hard masking of the data. Hard mask removes any pixel that has at least one frame with no signal

Keyword related to cross-correlation of data

cross_corr - Remove jitter from the CoMP data. Default operation is to use coordinate locations in find_man_date. If you do this, make sure to set the debug keyword for the first run, which will show you the box for which the cross-correlation is computed. If the default box does not look correct, set the values x1, x2, y1, y2 manually in find_man_date.

choose_corr_box - Select the cross-correlation box with the mouse.

dosob - uses a sobel filtered image for alignment (typically better & faster results)

Keyword related to wave angle calculation

compute_waveang - will compute the angles of wave propagation, i.e. orientation of magnetic field. Best to align data before calculating wave angle.

compute_speeds - compute phase speeds after the computation of the wave propagation angle.

plot_maps - plot ps-files with all quantities after the computation.

save_coh - Create a sav-file containing a measure of the coherence for each pixel. This is done by fitting an ellipse to the coherence island. The coherence for a pixel is good when the ellipse is long and slim, i.e. the normalized ratio of the major and minor axis of the ellipse is a measure of the coherence. The sav-file will contain an array coh_measure, which is just the major (coh_measure[*,*,0]) and minor (coh_measure[*,*,1]) axis of the ellipse. One possible way to compute a quantity for the coherence quality would then be coherence_quality = (coh_measure[*,*,0]-coh_measure[*,*,1])/(coh_measure[*,*,0]+coh_measure[*,*,1])

debug - set this if you want to visually check what's going on. *MUCH MUCH* slower computation though, so don't use this as the default.

wr_speeds - This is mainly intended for checking the computed speeds afterwards. It will write out *very* large arrays, so don't do this on a computer with insufficient memory or disk space, especially in combination with the save_coh keyword. Not really needed for everyday use, mostly for debugging.

altangle - Use alternative estimate for wave angle based on sixlin & choosing

splt_cube - save cubes as separate files