Northumbria University: Department of Mathematics, Physics & Electrical Engineering

Auto-NUWT User Guide

Micah Weberg
Richard Morton*
James McLaughlin

June 29, 2018

Version 2

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

Contents

History

Version 2 - 2018/04 - Pre-public release of Auto-NUWT.
Version 2 - 2018/06 - Fixed a couple of typos and added more program headers to Appendix B

1 Introduction

The Northumbria Wave Tracking (NUWT) code is a program designed to identify, track, and measure transverse wave motions within a series of images. This document describes, in short, the setup and operation of the automated version of NUWT. High-level details about how NUWT works and our efforts to validate the code can be found in the paper Weberg et al., 2018, ApJ, 852, 57

NUWT contains a comprehensive suite of programs and include procedures for extracting td-diagrams from calibrated data cubes, tracking a measuring transverse waves, and then plotting the results. NUWT also includes convenience functions for downloading SDO /AIA data and analyzing a large number of data slits.

This is a pre-public version of the code and the documentation (as well as this guide) may be incomplete in places. If you spot any mistakes or have any questions or suggestions please email *micah.weberg@northumbria.ac.uk* or *richard.morton@northumbria.ac.uk* .

2 Setup and Installation

NUWT is almost entirely self-contained and only requires IDL v8.4 (or higher) to run. To "install", simply unzip the NUWT source code to a location where IDL can find it (or add the NUWT folder to your IDL path as part of your startup script). All core NUWT scripts are included in the */NUWT* subdirectory.

If you wish to take advantage of the convenience functions for downloading and preprocessing SDO data, you will also need a working installation of the SolarSoft (SSW) IDL library which is available online at http://www.lmsal.com/solarsoft/.

3 Extracting data slits

This version of NUWT contains two procedures for extracting data slits from an input 3D datacube, *nuwt_diag_slit.pro* and *nuwt_arc_slit.pro*. Here are a couple of example uses:

```
   IDL>nuwt_diag_slit, datacube, output_slit, x1=100, x2=500, y1=100, y2=500
IDL>nuwt_arc_slit, datacube, output_slit, radius=1750, start_ang=85, subtend=10
```

Angles are measured in units of degrees. If any of the coordinate values are missing, the programs will enter an interactive mode where the user may select the slit for an example image of the input datacube. You may also set the */noopen* option to suppress plotting altogether.

Important note: Since NUWT is generalized to work with any set of input images, *nuwt_arc_slit.pro* uses the standard definition of polar coordinates in which angles are measured anticlockwise from the positive x-axis. This is different from the "position angle" commonly used in solar physics which is measured anticlockwise from the solar north pole.

4 Running NUWT

All NUWT actually needs to run is a time-distance diagram extracted from a series of images. For best results, it is recommended to also input error estimations for the td-diagram as well as the values for the spatial resolution and temporal cadence of the instrument which collected the data.

The main NUWT procedure is called *run_nuwt.pro*. A basic call will look like this:

```
   IDL>run_nuwt, td, errors=err_img, res=data_res, cad=data_cad
```

Where *td* is a time-distance diagram (or a stack of related td-diagrams), *err_img* is an image containing the estimated error values, *res* is the spatial resolution (in [arcsec]), and *cad* is the temporal cadence (in [s]) of the input data. If td is a stack of identically sized td-diagrams, such as a series of slits taken at different locations along a feature, NUWT will automatically loop over each slit.

Below are a few important parameters and keywords for *run_nuwt.pro*:

grad Gradient cut-off value used to select local maxima (or minima) in the td-diagram. The default value is 0.5 (good for unsharp masked data)

/invert If set, will invert the input td-diagram and track the local MINIMA in the image instead of the local maxima. Please set any invalid intensity values to a value of -999 or less before using running NUWT with the /invert option

/gauss If set, will fit the intensity cross-section of each peak to find the sub-pixel location.

min_tlen Minimum "thread" length used to filter the features found by NUWT. The default value is 20 data points.

/pad_fft If set, will use zero padding when running an FFT on each thread

pad_length If set, will use zero padding when running an FFT on each thread

slit_meta Metadata structure outputted by one of the NUWT slit extraction procedures. For SDO/AIA data, this may include res and cad values calculated from the data (thereby removing the need to directly track and pass the values to run_nuwt.pro yourself.

Please see the header text in run_nuwt.pro (in the /automation subfolder) for more information and additional parameters.

By default, NUWT will save all of the results to a file in your current working directory. The default filename will be of the form,

```
nuwt_results_run_YYYYMMDD_hhmm.sav
```

Where *YYYYMMDD_hhmm* is the date and time of when the program was run (note, if NUWT is run twice within the same minute, the output file will be overwritten by default). NUWT also stores the results from the most recent run in a set of IDL COMMON blocks. These COMMON blocks make plotting easy and reduces the need to juggle multiple output data structures.

5 Plotting Results

NUWT includes a number of procedures for quickly plotting the results. All of the plotting procedures make use of COMMON block data. Therefore, you must either run the plotting scripts in the same IDL session as you ran *run_nuwt.pro* (and before running a second set of data) or load the results into the COMMON blocks. The program, *restore_nuwt_common_data.pro* may be used to load results from a previous run. Typically you will use the procedure as such,

```
IDL>restore_nuwt_common_data, ’filename’
```

If the results file you wish to load is not in your current working directory, you will need to give the full path and filename. The *restore_nuwt_common_data.pro* procedure also has optional outputs for each of the NUWT data structures (see section 6 and appendix A for more details concerning the structure of the output variables).

Below is a list of the most useful plotting procedures:

plot_nuwt_peaks Plots the input td-diagram along with the located peaks and the tracked threads.

plot_nuwt_wave_hist Plots histograms of the waves found by NUWT. You may also input a structure of simulated wave values to compare with. By default, log-normal parameter distributions calculated from the data will be overplotted (can be hidden by setting the */hide_log_norm_dist keyword*).

plot_nuwt_power_spec Plots the amplitude, velocity amplitude, and power spectral densities of all waves found by NUWT. Note, the power values can be normalized to show the median or mean power observed.

plot_nuwt_fft_results Makes a series of multi-panel plots showing the showing the maxima locations, fit residuals, FFT spectrum, and wave parameters for each thread detected by NUWT. WARNING: this program will take a long time plot all of the results you can use the *plot_indices*, *first_thread*, & *last_thread* parameters to only plot a subset of the results.

plot_nuwt_compare_runs Produces a set of plots comparing the primary wave results of two separate NUWT runs or a set of simulated waves and the corresponding results from NUWT.

Please see the header text of each program (in Appendix B) for a more complete list of available parameters and keyword options. All plotting programs can be found in the */plotting_results* subfolder. Note, the default plot filenames DO NOT append the date and time. Please be sure to rename your files before making plots for a new run (unless you wish to overwrite the previous plot). All of the plotting routines have both *save_folder* and *filename* as optional parameters.

6 Extracting results for further analysis

6.1 Output Variables

Each *nuwt_results* save file contains 6 variables. With the exception of the *nuwt_meta structure*, each variable is actually a list of multiple arrays, lists, or structures. The first element (index 0) of each list contains the results for the first slit in the given NUWT run, the second element of each list corresponds to the second slit, and so on and so forth. The NUWT output variables will always be given as a list at the very top level, even if there is only a single slit in your NUWT run.

The NUWT output variables can be loaded using either a standard IDL, `restore, ’filename’` command or by using the optional outputs from *restore_nuwt_common_data.pro*. It is not recommended to try accessing the COMMON data blocks directly unless you are familiar with how to use them (IDL can be a bit temperamental with COMMON block definitions)
The output variables are as follows:

nuwt_meta Metadata about the NUWT run including data and time ran, key parameter values, and information about the data slit (when available)

nuwt_located Structure of 2D arrays with the intensity values and sub-pixel locations of the local maxima (or minima) found in the input td-diagram (note: locations are given in units of pixels)

nuwt_threads Array of structures with the time series with the positions of the ”threads” connected and tracked by NUWT.

nuwt_fft_spec List of structures with the frequency spectra output by the FFT for each thread (note: will automatically convert the spectra to physically sensible units if *res* and *cad* are given to NUWT at runtime).

nuwt_fft_peaks Array of structures with the wave parameters for all of the significant peaks found in the FFT spectra. Up to four different waves may be returned for each thread. (again, if NUWT knows [res] and [cad], it will convert the values to physical units).

nuwt_bulk_stats Ordered hash table containing the bulk statistics (such as number of threads and waves found) as well as the mean, median, and stddev of the basic wave parameters (log-normal values are also included). See appendix A.6 for more information concerning using the ordered hash table.

More information about the keys available in the output structure can be found by typing,

```
IDL>help, STRUCTURE_NAME
```

in the IDL command line (remember, you may need extract the desired element from the top-level list first). You may also get more information by reading the header text of the *nuwt_locate_things.pro*, *nuwt_follow_threads.pro*, and *nuwt_apply_fft.pro* programs in the */NUWT subfolder*. See appendix A for more details concerning the structure of the output results.

6.2 Flattening Structures

Since NUWT can return multiple (and differing numbers of) waves for each thread, the format of the *nuwt_fft_peaks* structure is somewhat unwieldy and can make it difficult to produce certain summary plots (such as a scatter plot of ALL non-zero wave amplitudes, regardless of thread number or wave order). Therefore, the *flatten_nuwt_results* function is provided to, as the name suggests, flatten the wave parameters into 1D arrays. You do not need to restore the NUWT results to the COMMON blocks before calling *flatten_nuwt_results*.

Example use,

```
IDL>nuwt_params = flatten_nuwt_results(’filename’)
```

If *filename* is omitted, the results for the currently active NUWT run will be returned instead. *Flatten_nuwt_results* can also convert the wave parameters into the units of your choice be setting the *amp_units*, *freq_units*, and *period_units* keywords (defaults are ”km”, ”Hz”, and ”s”, respectively) The output structure includes arrays of thread indices and wave order numbers in order to easily reference the extra details in the original NUWT output structures and allow for filtering to be done with a simple *where()* function call.

7 To do

Below is a list of upgrades that are under consideration for future implementation. This list is not exclusive and items on the list may or may not actually be included in new versions of the code. If you find any bugs or have suggestions for improvements, please let us know so that we may update this list.

- Included the Lomb-Scargle periodogram as an alternative to the FFT method when analysing data with uneven temporal sampling
- Apply FFTs to smaller, overlapping subsections of each thread to test for time variance in the wave parameters (note: this will reduce the accuracy of the wave frequencies)
- Expand the goodness-of-fit testing and improve the automatic quality filtering.
- Add a tutorial section to this manual.

A Appendix A: NUWT Output Structures

A.1 located

Structure of 2D arrays with the intensity values and sub-pixel locations of the local maxima (or minima) found in the input td-diagram (note: locations are given in units of pixels). Created by the *nuwt_locate_things.pro* routine. The format is as follows:

.peaks [nx,nt,2] or [nx,nt,6] array with sub-pixel locations and maximum values at each located peak (in [pixels])
`[*,*,0]` - sub-pixel location of each peak
`[*,*,1]` - maximum intensity value at each peak
`[*,*,2]` - Gaussian width (/full_gauss only)
`[*,*,3]` - constant coeff (/full_gauss only)
`[*,*,4]` - slope of linear term (/full_gauss only)
`[*,*,5]` - i index of maximum data point (/full_gauss only)

.errs [nx,nt] or [nx,nt,6] array with errors on the fit values. Format is similar to .peaks (but for the errors instead)

.allpeaks [nx,nt] array containing integer flags for ALL local maxima. The values are as follows:
1 - local peak rejected by chosen gradient
2 - Gaussian fit failed. Defaulted to whole pixels
3 - Gaussian fit obtained for subpixel resolution

.grad_left [nx,nt] array with the gradient values on the lefthand side of ALL potential peaks

.grad_right same as as the above but for right-hand side gradients

.td_img td-diagram AFTER any inversion or despiking operations are applied

.inverted binary flag indicating if the image was inverted .despiked - binary flag indicating id the image was despiked .spike_sigma

.smoothed binary flag indicating if the image was smoothed over time before locating peaks

.sm_width bin widths used for smoothing in each dimension. A value of 1 indicates no smoothing in the corresponding dimension

.dx [pixels] to [.units_dx] conversion factor

.units_dx desired distance units after after multiplying the pixel location by .dx

.dt [timesteps] to [.units_dt] conversion factor

.units_dx desired time units resulting from after multiplying the timestep number by .dt

.res & .cad source data resolution and cadence values given to NUWT by the user

.km_per_arcsec [arcsec] to [km] conversion factor. Defaults to 725.27 km/arcsec, which is the mean scale of the solar surface as viewed from 1 AU.

A.2 threads

Array of structures with the time series with the positions of the ”threads” connected and tracked by NUWT. Created by the *nuwt_follow_threads.pro* routine. Each thread structure (array element) has the following format:

.pos [nt] long array with the thread position at each timestep. Values of -1 indicate timesteps outside of the thread and a 0 indicates timesteps skipped due to no nearby peaks

.err_pos [nt] long array with the thread position errors

.bin_flags [nt] array with flags indicating the type of peak found in each timestep of the thread. Possible values are:
-1 : time-step not part of thread
0 : data gap inside thread
1 : lower quality data found (not currently used but may be used in the future for filling with rejected peaks)
2 : higher quality data found

.start_bin timestep bin with the first thread position

.end_bin timestep bin with the last thread position

.length total length (in timesteps) of the thread

If *run_nuwt.pro* was run with the */FULL_GAUSS* option set, the output structures will also include the following tags:

.inten [nt] long array with the peak intensity values

.err_inten [nt] long array with the estimated intensity errors

.wid [nt] long array with the Gaussian width found by the subpixel fitting method of *locate_things.pro*

.err_wid [nt] long array with the Gaussian fit errors

A.3 fft_spec

List of structures with the frequency spectra output by the FFT for each thread (note: will automatically convert the spectra to physically sensible units if *res* and *cad* are given to NUWT at runtime). Created by the *nuwt_apply_fft.pro* routine. The format of each list element (i.e. substructure) is described below). Please note: **(a)** *nf* is the number of frequency bins in the FFT spectrum for the given thread. The number of frequency bins depends on the number of datapoints in the thread and the amount of zero padding (if used), therefore *nf* will be different for each thread. **(b)** error values are only calculated is *run_nuwt.pro* is ran with the */BOOTSTRAP* option set. Otherwise, all errors in both *fft_spec* and *fft_peaks* will default to values of 0.

.power [nf] array of the corrected power spectral density in each FFT bin

.err_power [nf] array with the power errors computed using bootstrapping

.power_units Output power units. Defaults to [km2̂ s] if *res* & *cad* are known and [pixels2̂ timesteps] if they are not.

.power_to_pxls May be used to convert the power values back to units of [pixels2̂ timesteps] (in case you wish to recompute the units yourself)

.amplitude [nf] array with the amplitude value calculated from the power spectral density above

.err_amplitude [nf] array with the amplitude error values computed using bootstrapping

.amp_units Output amplitude units. Defaults to [km] is *res* is known and [pixels] if it is not.

.amp_to_pxls May be used to convert amplitude to units of [pixels]

.freq [nf] array with the frequency in each FFT bin

.freq_units Output frequency units. Defaults to [Hz] is *cad* is known and [timesteps] if it is not.

.freq_to_timesteps May be used to convert frequencies to units of [timesteps]

.phase [nf] array with the phase value in each FFT bin

.err_phase Phase errors computed using bootstrapping

.trend [thread length] size array with the trend removed from the thread positions before applying the FFT. By default, the "trend" removed will simply be the mean position value (this is necessary since the FFT requires values to be given as perturbations from a zero)

.trend_poly_degree Polynomial order of the removed trend. "0" indicates a constant mean value was removed and "1" indicates a linear fit.

.apod_window [thread length] size array with the apodization (also known as "tapering") window values. Apodization is important as it helps reduce spectral leakage between FFT bins.

.window_func Name of the window function used. Defaults to the "split_cosine_bell" (also known as the "hann") window. Other window functions may be added in future versions of the code.

.window_param Parameter value(s) that modify the window function.

.signif_vals [nf] array with the significant power threshold used for selecting waves.

.fft_length Length of the array input to the FFT (including zeros). Note, this value is not the same as *nf*

.signif_test Method used to compute the significance values. Defaults to a method adapted from Torrence & Compo 1998, *Bul. Amer. Met. Soc. (BAMS)* which uses a white noise spectrum computed from the input data

.signif_level Significance level used. Defaults to 0.95 (i.e. the 95% confidence level)

.bin_flags [nf] array indicating which frequency bins have values above the significance threshold. Key:
-2 : invalid thread (too little real data)
-1 : empty or invalid frequency bin
0 : power below selected significance level
1 : significant value above the selected threshold
2 : local maximum among adjacent significant values (these are the waves selected by NUWT)

.enbw Effective Noise Bandwidth. Used to correctly scale the power spectrum and calculate amplitude values.

.cpg Coherent Power Gain of the apodization window

A.4 fft_peaks

Array of structures with the wave parameters for all of the significant peaks found in the FFT spectra. Up to four different waves may be returned for each thread. (again, if NUWT knows */*res] and */*cad], it will convert the values to physical units). Created by the *nuwt_apply_fft.pro* routine.

.analysis_method Method used to apply the FFT and identify wave parameters. Currently, "FFT" is the only analysis method used (eventually, new methods will be added)

.peak_power Peak power of the significant waves in order of magnitude. The first value corresponds to the largest peak, the second value represents the second largest peak, and so on and so forth. Values of 0.0 indicate that no wave of that particular order was found.

.err_peak_power Peak power errors computed using bootstrapping

.power_units Output power units. Defaults to [km2̂ s] if *res* & *cad* are known and [pixels2̂ timesteps] if they are not.

.power_to_pxls May be used to convert amplitude to units of [pixels]

.peak_amplitude Peak wave displacement amplitudes

.err_peak_amplitude Peak amplitude errors computed using bootstrapping

.amp_units Output amplitude units. Defaults to [km] is *res* is known and [pixels] if it is not.

.amp_to_pxls May be used to convert amplitude to units of [pixels]

.peak_freq Peak wave frequencies

.freq_units Output frequency units. Defaults to [Hz] is *cad* is known and [timesteps] if it is not.

.freq_to_timesteps May be used to convert frequencies to units of [timesteps]

.peak_vel_amp Peak wave velocity amplitudes. Calculated using the equation $v = 2\pi\xi f$ where ξ is the displacement amplitude and f is the frequency.

.vel_amp_units Output velocity amplitude units. Defaults to [km]/[s] if res and cad are known and [pixels]/[timesteps] if they are not.

.peak_phase Peak wave phase

.err_peak_phase Peak phase errors computed using bootstrapping

.peak_bin FFT bin number of each significant peak.

.adjacent_peaks [EXPERIMENTAL] Binary flag indicating if adjacent FFT bins may be selected as separate waves. By default, only local maxima in the FFT spectrum may be selected as a wave (assuming it is above the significance threshold)

.num_signif_peaks Total number of significant peaks found in the FFT spectrum. Note: this number may include peaks that fall below the minium frequency cutout value.

.num_saved_waves Number of waves selected and saved in this structure

.signif_level Significance level used. Defaults to 0.95 (i.e. the 95% confidence level)

.KS_stat KolmogorovSmirnov one-sample test statistic. The stat computed from the residuals after removing the first N waves combined is stored in the i = N position. In other words, the 1st result (index 0) is the stats for the null case of no waves, the 2nd (index 1) results is for the first wave, the 3rd (index 2) is for the first two waves inclusive, and so on and so forth...

.KS_prob Probability that the residuals are normally distributed as detected by the KS test statistic (which is sensitive to skewness and shifts in the median value and may yield incorrect probabilities).

.AD_stat Anderson-Darling test statistic. Alternative to the KS-test which also tests for normally distributed residuals. Order of values is similar to the *KS_stat*

.AD_crit Critical value for the Anderson-Darling test

.LB_stat Ljung-Box statistic which tests for autocorrelation in the residuals

.LB_chisqrd Ljung-Box critical value (which is based on a χ^2 distribution)

.enbw Effective Noise Bandwidth. Used to correctly scale the power spectrum and calculate amplitude values.

.user_qual_flag User quality flag set using the */INTERACTIVE* mode of *set_nuwt_qual_flags.pro*

.auto_qual_flag Automatically generated quality flag. Currently, only flagged based on the percent of data gaps in the thread (this may change in future versions of the code).
1 : 35% - 50% data points are missing (ok quality) . Note: threads with >50% data gaps are rejected by NUWT.
2 : <35% data points are missing (best quality)

A.5 bulk_stats

Ordered hash table containing the bulk statistics (such as number of threads and waves found) as well as the mean, median, and stddev of the basic wave parameters (log-normal values are also included). Note: the use of a hash table may be unfamiliar to some IDL users, the values can be accessed by using the syntax `TABLE_NAME['stat_name']` (this allows for *stat_name* to be constructed programmatically while looping over different parameters and summary statistics). Keys included in the *bulk_stats* hash table are given below.

Miscellaneous counts and values:

'num_threads' Total number of threads found by NUWT (including threads without waves)

'num_waves' Total number of waves identified by NUWT, regardless of quality flags

'filtered_num_waves' Number of waves in the selected quality flag range

'wave_counts' Array giving the number of threads with N waves or more. That is, the first (index 0) is the number of threads with zero waves, the second value (index 1) is the number of threads with at least one wave, the third value (index 2) is the number of threads with two or more waves, and so on and so forth.

'user_flag_range' Range of user quality flags included 'auto_flag_range' Range of auto quality flags included

'wave_order_range' Range of wave order numbers included in the calculations

Short variable names: (must be used in conjunction with the full statistic names)

'amp' Wave displacement amplitude

'period' Wave periods

'freq' Wave frequency

'vel_amp' Wave velocity amplitude calculated using the equation $v = 2\pi\xi f$ where ξ is the displacement amplitude and f is the frequency.

Full statistic names: To reference the actual values of the statistics in the table, replace "VAR" in the list below with one of the short variable names given above. **Note well:** only non-zero waves of the selected quality range are used to calculate the statistics below. Detected threads without significant waves are not included in the calculations.

'VAR_mean' Arithmetic mean of the selected variable for all good quality waves

'VAR_stddev' Standard deviation

'VAR_median' Median value of the selected variable

'VAR_MAD' Median absolute deviation

'log_norm_VAR_mean' Log-normal mean calculated using the equation $EXP(\mu + (\sigma^2)/2.0)$ where $\mu = \Sigma ln(x)/N$ and $\sigma = ((N-1)/N) * STDDEV(ln(x))$, and N is the total number of values

'log_norm_VAR_stddev' Log-normal standard deviation calculated using $SQRT((EXP(\sigma^2)-1)*EXP(2*\mu+\sigma^2))$

'log_norm_VAR_mode' Log-normal mode calculated using $EXP(\mu - \sigma^2)$

'VAR_units' Units of the selected variable

B Appendix B: Program Headers

B.1 *run_nuwt.pro*

```
;+
;NAME: RUN_NUWT
;
;PURPOSE:
;   One of the top level Northumbria University Wave Tracking (NUWT) run scripts.
;   Takes a time distance diagram, picks out peaks in the intensity (also called
;   'threads'), then feeds them into an FFT to calculate the power and frequencies
;   of oscillation (if any). See the user guide (once available) and the code of
;   each sub-program for more information.
;
;INPUTS:
;   input_data - an array of time-distance (t-d) diagrams of the form (x,t,z)
;
;OPTIONAL INPUTS:
;   errors - estimated errors on intensity values, supplied to gaussian fitting routine.
;            if not supplied a default value of 10% of intensity values is used
;   /invert - If set, will invert the intensity values of the td-diagram before
;             finding peak locations. This results in the code finding the local
;             MINIMA in the original image.
;   grad - gradient limit used by "locate_things.pro". Default is 0.5 (good for
;          unsharp masked data)
;   min_tlen - minimum thread length used by "follow_threads.pro". Default is 20
;   max_dist_jump - maximum allowable distance (in pixels) between two peaks in
;                   the same thread. Used by "follow_threads.pro". Default is 3 pixels
;   max_time_skip - maximum allowable timesteps beweewn two peaks in the same
;                   thread. Used BY "follow_threads.pro". Default is 4 time steps.
;   res - spatial resolution of data in [arcsec]. Defaults to a value of 1 [pixel]
;   cad - temporal cadence of the data in [s]. Defaults to a value of 1 [timestep]
;   km_per_arcsec - ratio of [km]/[arcsec]. Defaults to 725.27 which is the
;                   mean scale of the solar surface.
;   /gauss - uses a gaussian fit to locate thread centres; provides sub-pixel measurements
;   /full_gauss - similar to the above. Will also return the fitted gaussian widths (for debu
;   /pad_fft - if set, will pad the array with zeros to improve the precision
;              of peak frequencies returned by the fft. Note, this does NOT
;              actually increase the resolution and can results in extra spectral
;              leakage. Use with care.
;   /fill_pad - if set, will fill the array with extra zeros until the length equals
;               "pad_length" instead of appending a set number zeros (default).
;               Note: if the pad_length is set too low, using /fill_pad may
;               introducing "banding" in the possible frequency values returned.
;   pad_length - number of zeros to pad to the end of the array when /pad_fft is set.
;                If /fill_pad is ALSO set, will instead pad the array until the
;                total length equals pad_length (or just pad with 1 zero if the
;                array is already longer than pad_length). Default is 1000.
;   /bootstrap - if set, will perform bootstrapping to resample the thread positions
;                before running the FFT. This gives a rough error estimates on the
;                FFT spectrum and output wave paramters
;   num_bootstrap - number of bootstrap resamples to use. Default is 1000
;   /vel_amp_mode - [EXPERIMENTAL] alternate mode for "nuwt_apply_fft.pro" (see
;                   that code for more information)
;   slit_meta - metadata structure outputted by "nuwt_arc_slit.pro" or "nuwt_diag_slit.pro"
;   /aia - if set, will assume input data came from SDO / AIA and proceed to use
;          a few convenience functions.
;   wavelength - wavelength of input AIA data from calculating intensity errors.
;   slit_norm_num - width of data slit. Alos used for calcuating AIA intensity errors
;   /interactive - if set, will allow for interatively picking gradient and min thread length
;   save_folder - folder in which to save the results. Defaults to current folder.
;                 Will also append a '/' to the end if not included.
;   filetag - unique tag to append to the START of the output file
;   default_plot_header - string that can be optioanlly defined an used when
;                         plotting the data using the NUWT plot procedures
;
;OUTPUTS:
;   common blocks contain the results and are used for all internal passing of
;   data between subprograms.
;
;EXTERNAL CALLS - locate_things.pro, follow_thread.pro, patch_up.pro, nuwt_apply_fft.pro
;
;HISTORY: Name---------Date---------Description
;         M Weberg  JUNE, 2016  Initial coding.
;         M weberg  SEPT, 2016  Made some quality-of-life improvements
;                               - Added a master COMMON block that compiles the
;                                 results from all of the data slits into lists.
;                                 Can also output this block to a '.save' file
;                                - Added an interactive mode for selecting gradients
;                                  and minimum thread lengths
;         M Weberg   FEB, 2017   Added vel_amp_mode options
;         M Weberg   MAR, 2017   Updated all NUWT procedures to automatically
;                                calculate parameters in physical units when given
;                                the "res" and "cad" for the input data. Also
;                                added some convenient defaults when using aia data.
;         M Weberg   MAY, 2017   Added slit_meta functions
;         M Weberg  JUNE, 2017   Added the "nuwt_meta_dat" COMMON block and structure
;         M Weberg  JULY, 2017   Exposed the "max_dist_jump" & "max_time_skip" options
;                                from "follow_thread.pro" to be changable via inputs
;                                to this script
;         M Weberg  SEPT, 2017   Renamed "nuwt_fft_results" to "nuwt_fft_spec" in
;                                all NUWT programs (more appropriate name)
;         M Weberg   JAN, 2018   Renamed a number of subprograms and output structure
;                                to have more intuitive names (most importanly,
;                                "nuwt_fft_stats" is now "nuwt_fft_peaks" and
;                                "plot_nuwt_fft_peaks" is now "plot_nuwt_wave_hist")
;
;TO DO / RESTRICTIONS:
;   - Add funcationality for passing a structure with the desired keyword
;     arguments (kwargs) for each subprogram.
;   - Add options for plotting the data after processing
;-
```

B.2 *plot_nuwt_peaks.pro*

```
;+
;NAME: PLOT_NUWT_PEAKS
;
;PURPOSE:
;   Plots the intensity peaks found by 'locate_things.pro' and the threads found
;   by 'follow_thread.pro'. Will also plot the input td-diagram as well as histograms
;   of the gradients on both the left and right sides (as an aid for selecting
;   a good gradient. Can plot either to the screen or a PDF (for later refference)
;
;INPUTS:
;   input_data - [optional positional input] image that was processed by NUWT
;                (typically a td-diagram). If no image is given, the program will
;                default to a copy of the image stored within the 'located' structures
;                produced by "locate_things.pro" (note: this image may be despiked
;                or have inverted values).
;
;   All other required inputs are loaded from the 'all_nuwt_dat' COMMON block
;   containing the results from 'locate_things.pro' and 'follow_threads.pro'
;
;OPTIONAL INPUTS:
;   res - spatial resolution of data (in [arcsec]). Default is 1
;   cad - temporal cadence of the data (in [s]). Default is 1
;   km_per_arcsec - ratio of [km]/[arcsec]. Defaults to 725.27 which is the
;                   mean scale of the solar surface.
;   grad - gradient used to filter peaks. If not set, will estimate from the data
;   /final_units - [DEPRECIATED] if set, will load "res", "cad", and "_units" values from the
;                  NUWT "located" structure.
;   dist_units - string indicating what units to use for distance in the plot
;                labels. Defaults to units of [pixels] unless "res" is set,
;                in which case the default is [arcsec]
;    time_units - string indicating what units to use for time in the plot
;                 labels. Defaults to units of [steps] unless "cad" is set,
;                 in which case the default is [s]
;    /underplot_td - if set, will underplot the td-diagram beneath the "Peak locations"
;                    and "Threads found" images. Note: plots may become harder to read.
;    /show_rejected_peaks - if set, will plot all of the peaks rejected by the gradient
;    /normalized - if set, will normalize the gradient histograms by the number of peaks
;    /plot_td_only - if set, will only plot the input td-diagram (useful mainly for
;                    presentations or debugging)
;    /plot_peaks - if set, will plot input td, located peaks, and histograms of gradients
;    /plot_threads - if set, will plot located peaks and found threads
;    /screen - if set, will plot to the screen rather than a file
;        Note: Defaults to plotting everything if neither /plot_peaks or /plot_threads is set
;              For consistancy, will also plot everything if saving to a file.
;    /use_temp_common_blocks - if set, will use the temporary common blocks rather
;                              than the all_nuwt_dat block. Use with care!
;    slitnum - virtual slit number for header text. By defualt assumes 1
;    header - custom header text. Useful for identifying the source data.
;             Defaults to 'NUWT FFT stats'.
;    run_tag - custom string that will be appended to the end of the header.
;              Normally used to keep track of differnt runs of the same dataset.
;              There is no default run_tag string.
;    save_folder - folder in which to save the plots. Defaults to the user's home folder.
;                  Will also append a '/' to the end if not included.
;    filename - name for the output PDF file. Default is 'NUWT_selected_peaks'
;
;OUTPUTS:
;    PDF_file - multi-page PDF containing (1) the input td-diagram, (2) selected
;               peaks, (3) threads found, and (4) histograms of gradients
;
;HISTORY: Name—————Date—————Description
;         M Weberg   MAY, 2016  Initial coding. Was part of a modified version
;                               of 'locate_things.pro'
;         M Weberg  JULY, 2016  Moved to seperate program and extended options
;         M Weberg   AUG, 2016  Fixed gradient estimation & added basic unit conversions
;         M Weberg  SEPT, 2016  Large update:
;                                - more plots (better PDF output, can also show threads)
;                                - more diagnostic information
;                                - more options (for interatively exploring parameters)
;                                - reconfigured to use NUWT master COMMON block
;         M Weberg   JAN, 2017  Added the "underplot_td" option
;         M Weberg   MAR, 2017  Now can load unit information from the NUWT COMMON
;                               block structures
;
;TO-DO / LIMITATIONS:
;   - Add inputs for metadata indicating the data and source of observations.
;-
```

B.3 *plot_nuwt_wave_hist.pro*

```
;+
;NAME: PLOT_NUWT_WAVE_HIST
;
;PURPOSE:
;    Plots histograms of the peak FFT wave values from 'nuwt_apply_fft.pro' for all
;    significant waves. Returns one set of plots for each order (primary, secondary,
;    etc.) of waves as well as the combined distibution of all waves. Additionally,
;    a calculated log-normal normal distribution of each parameter will be overplotted
;    The vertical scale of the log-nromal distributions will be adjusted to match
;    the y-axis units of the histograms
;
;INPUTS:
;    None directly. Loads in the common blocks containing the results from
;    'locate_things.pro' (not currently used), 'follow_thread.pro', and
;    'nuwt_apply_fft.pro'
;
;OPTIONAL INPUTS:
;    res - spatial resolution of the data (in [arcsec]). If not given by the user,
;          the program will load the 'res' value stored by NUWT for the current
;          set of results. If NUWT does not have a valid res value, then all
;          distance values will revert to units of [pixels]
;    cad - temporal cadence of the data (in [s]). If not given by the user, the
;          program will load the 'cad' value stored by NUWT for the current set of
;          results. If NUWT does not have a valid cad value, then all time values
;          will revert to units of [timesteps]
;    km_per_arcsec - ratio of [km]/[arcsec]. Defaults to 725.27 which is the
;                    mean scale of the solar surface as seen from 1 AU.
;    /final_units - [DEPRECIATED] (will be removed in the near future)
;    min_user_flag - minimum user quality flag to plot. These flags are set with
;                    'set_nuwt_qual_flags.pro'. Default is -1 (all waves)
;    max_user_flag - maximum user quality flag to plot (see above). Default is 1000
;    VVV_range - range to plot for amp, period, or vel amp. Defaults are [0,2000] km,
;                [0,2000] s, & [0,50] km/s respectively
;    VVV_units - string with the plot units for amp or period (vel_amp units are
;                automatically determined from the other two). Default is [km]
;                for amp_units and [s] for period_units
;    VVV_binsize - size of histogram bins for amp, period, or vel amp in the selected
;                  units. Defaults are 25 km, 25 s, & 1 km/s for amp, period,
;                  and vel_amp respectively.
;    VVV_nbins - number of histogram bins for amp, period, or vel amp. By default,
;                VVV_nbins =  1 + (VVV_range[1] - VVV_range[0]) / VVV_binsize
;    /normalized - if set, will normalize the histograms by the number of waves
;    /hide_log_norm_dist - if set, will hide the lines showing the calculated
;                          log-normal distributions for each parameter.
;    /clean - if set, will suppress plotting symbols and text showing the mean
;             and median parameter values
;    ref_waves - stucture containing either input simulated wave parameters (from
;                'generate_kink_waves.pro') or the 'fft_peaks' output from a different
;                run of NUWT. Will be histogrammed and compared to the current results.
;                Used for testing and validation.
;    /simulated - if set, will assume the ref_waves structure contains input
;                 simulated parameters (default)
;    /nuwt - if set, will assume the ref_waves structure contains output from
;            a seperate NUWT run. Note: if both /simulated and /nuwt are set, the
;            program will give precedence to the /simulated option.
;    slitnum - virtual slit number to plot. By default, will show slit number "0".
;              Note: can also set to 'all' to plot a combined histogram of all slits.
;    header - custom header text. Useful for identifying the source data.
;             Defaults to 'NUWT FFT stats'
;    run_tag - custom string that will be appended to the end of the header.
;              Normally used to keep track of differnt runs of the same dataset.
;              There is no default run_tag string.
;    save_folder - folder in which to save the plots. Defaults to the user's home folder.
;                  Will also append a '/' to the end if not included.
;   filename - name for the output PDF file. Default is "NUWT_fft_peaks"
;
;OUTPUTS:
;   PDF_file - multi-page PDF containing FFT stats and diagnostics for each
;              ordered group of waves (primary, secondary, etc.) The first page
;              will show the distributions of all waves combined.
;   bulk_stats_out - stucture with the calculated summary statistics for the
;                    selected NUWT waves
;   ref_stats_out - stucture with the calculated summary statistics for the
;                   reference data
;
;HISTORY: Name----------Date----------Description
;         M Weberg  8 Sept, 2016  Initial coding
;         M Weberg 14 SEPT, 2016  reconfigured to use NUWT master COMMON block
;         M Weberg 18  OCT, 2016   added output structure for summary stats
;         M Weberg 09  JAN, 2017   added the "all" option to "slitnum"
;         M Weberg ??  MAR, 2017  Now can load unit information directly from
;                                 NUWT common block structures
;         M Weberg ?? JUNE, 2017  corrected how log-normal values are calculated
;         M Weberg 25 JULY, 2017  Modified slitnum="all" mode to allow for merged
;                                 nuwt results from runs with different number of
;                                 timesteps.
;         M Weberg     FEB, 2017  Reworked how NUWT data is loaded and unit
;                                 calculations performed (cleaner code with more options)
;                                 Also changed the default to always plot the calculated
;                                 log-normal distributions (can be hidden with
;                                 /hide_log_norm_dist)
;
;TO-DO / LIMITATIONS:
;   - more options for the 'compare_waves' distibution that would allow comparisions
;     to other NUWT results and not just the simulated waves input for testing
;   - Automatic metadata handling to denote dataset and source of observations.
;     (currently managed via user defined text strings)
;-
```

B.4 *plot_nuwt_fft_results.pro*

```
;+
;NAME: PLOT_NUWT_FFT_RESULTS
;
;PURPOSE:
;   Plots the FFT wave results from 'nuwt_apply_fft.pro' for each thread as a
;   seperate multi-panel plot in a single PDF. Also prints various parameters
;   for diagnostic purposes.
;
;INPUTS:
;   None directly. Loads in the common blocks containing the results from
;   'locate_things.pro' (not currently used), 'follow_thread.pro', and
;   'nuwt_apply_fft.pro'
;
;OPTIONAL INPUTS:
;   res - spatial resolution of the data (in [arcsec]). If not given by the user,
;         the program will load the 'res' value stored by NUWT for the current
;         set of results. If NUWT does not have a valid res value, then all
;         distance values will revert to units of [pixels]
;   cad - temporal cadence of the data (in [s]). If not given by the user, the
;          program will load the 'cad' value stored by NUWT for the current set of
;          results. If NUWT does not have a valid cad value, then all time values
;          will revert to units of [timesteps]
;    km_per_arcsec - ratio of [km]/[arcsec]. Defaults to 725.27 which is the
;                    mean scale of the solar surface as seen from 1 AU.
;    dist_units - string indicating what units to use on the distance axis of plot
;                 Defaults to units of 'arcsec' if 'res' is known or 'pixels' if
;                 'res' is unknown.
;    time_units - string indicating what units to use on the time axis of plots.
;                 Defaults to units of 's' if 'cad' is known or 'timesteps' if
;                 'cad' is unknown.
;    VVV_units - string with the plot units for amp, freq, or period (vel_amp and
;                power units are automatically determined). The defaults are
;                [km] for amp_units, [Hz] for freq_units, and [s] for period_units
;    /final_units - [DEPRECIATED] (will be removed in the near future)
;    first_thread - index of the first thread to plot. Default is 0
;    last_thread - index of the last thread to plot. Default is (num_threads-1)
;    plot_indices - indices of the threads to plot (overides first_thread and
;                   last_thread). This allows more complicated plotting
;    slitnum - virtual slit number for header text. By defualt assumes 1
;    header - custom header text. Useful for identifying the source data.
;             Defaults to 'NUWT FFT Results'
;    run_tag - custom string that will be appended to the end of the header.
;              Normally used to keep track of differnt runs of the same dataset.
;              There is no default run_tag string.
;    save_folder - folder in which to save the plots. Defaults to the user's home folder.
;                  Will also append a '/' to the end if not included.
;    filename - name for the output PDF file. Default is 'NUWT_FFT_results'
;
;OUTPUTS:
;    PDF_file - multi-page PDF containing FFT results and diagnostics for each
;               thread / set of waves found. If there are multiple significant
;               wave results, the program will plot the combined waveform of
;               all of the waves with power above the significance threshold.
;
;HISTORY: Name---------Date---------Description
;         M Weberg  30 JUNE, 2016  Initial coding
;         M Weberg  26  AUG, 2016  Expanded options and updated plot format
;                                  - Added basic unit conversions
;                                  - Will now plot / list the four largest waves
;                                  - Can not be passed an arbitary header text
;                                  - Other visual tweaks and small improvements
;         M Weberg  14 SEPT, 2016  reconfigured to use NUWT master COMMON block
;         M Weberg  ??  MAR, 2017  Now can load unit information directly from
;                                  NUWT common block structures
;
;TO-DO / LIMITATIONS:
;    - Automatic metadata handling to denote dataset and source of observations.
;      (currently managed via user defined text strings)
;-
```