Updates to cBathy, version 3.0

Rob Holman

07/24/19

This short paper documents changes made to cBathy to accommodate short records from drones. The algorithm is based on version 2.0 which uses adaptive tile sizes. The codes are stored in the directory V3ShortRecordscBathy and are parallel to version 2.0 but with the name Short appended on routines that have been substantially changed. Rather than adapting the code to handle either short or long records, the codes are kept separate, leaving it to the user to decide which version should be used (i.e. there needs to be a wrapper that makes this determination and calls the appropriate toolbox).

The functional definition of ‘short records’ is that the record length is too short for cross-spectral analysis. In a cross-spectrum, functions of Fourier coefficients are averaged over a set of N adjacent frequencies in a frequency band. Since the number of available frequencies depends on record length, short records will simply not have enough frequencies for sensible band-averaging while maintaining sufficient spectral resolution. As a rule of thumb, you should not have fewer than three Fourier frequencies per band to use cross-spectral methods. If we desire a frequency resolution of 0.02 Hz (often a default choice), then we require at least T > 3/0.02 = 150 s record length to use the cross-spectral methods in V2.0.

The absence of cross-spectral methods for short records has several consequences. The most important are the inability to a) choose the dominant frequencies from tile coherences (derived from the cross-spectral matrix, or CSM) and b) to use eigenvectors of the CSM as tile phase maps for finding k-. In addition, version 2.0 analyzes user-specified frequency bands that represent the expected wave climate in the region, knowing that Fourier estimates are dense between these band centers. This is not true for short records – available Fourier frequencies will be sparse and may be poorly aligned with a user selection. Thus, standard Fourier frequencies are used instead of user suggestions.

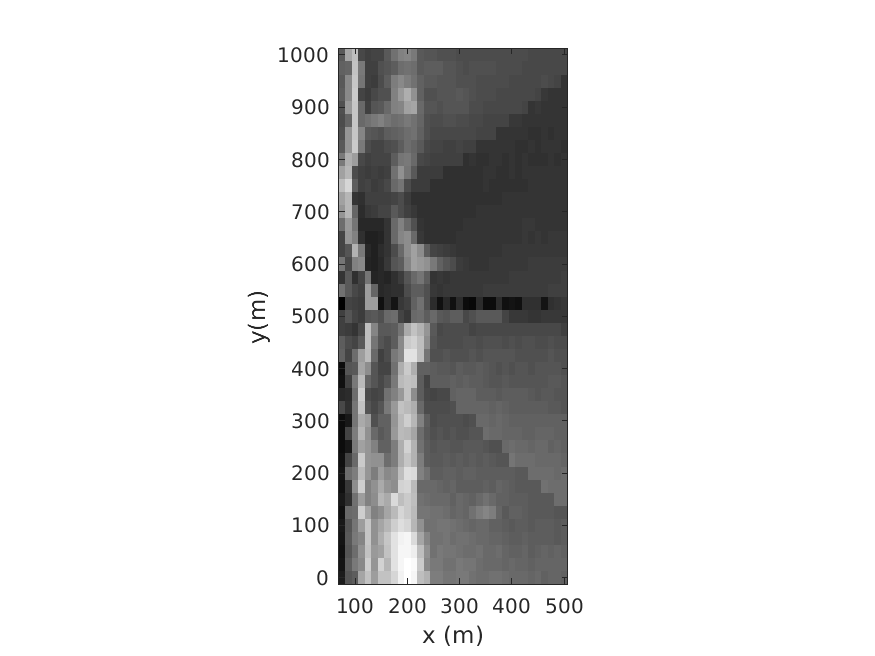
Algorithm changes to handle these issues are now described.

*Selection of Dominant Frequencies*

cBathy was designed to handle the full range of wave conditions that might be encountered without user intervention. The main unknown for any data collection is the frequencies of wave energy for any day or site. Thus, cBathy must assess from among a suite of possible choices, those frequencies that are present in a dataset. For long data runs, cBathy does a tile-by-tile assessment prioritized by the total coherence in the CSM, so frequencies that appear most coherent across the tile are selected.

In the absence of a CSM, an alternate method is needed. Outside of the surf zone, it is reasonable to simply find peaks in spectra, i.e. choose the frequencies associated with the N most energetic spectral values. However breaking waves completely dominant spectral energy so surf zone waves will often have spectral peaks that are associated with groupiness, or fluctuations in wave breaking, and are not representative of the ocean wave frequencies. As a further complication, it is not usually known a priori how wide the surf zone will be.

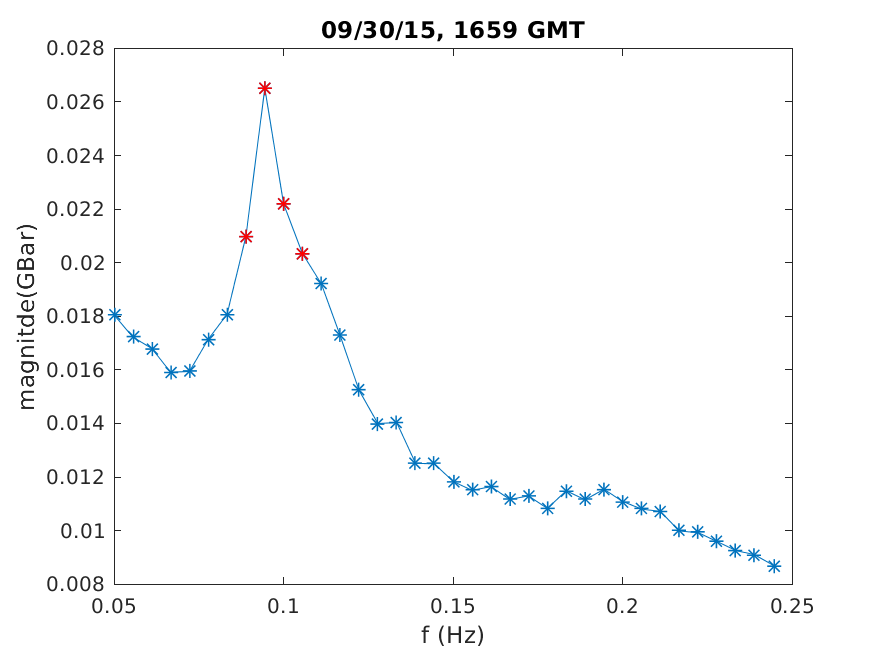
A pragmatic solution was to find an average cross-shore transect of pixel intensity, IBar(x), by averaging in time and in the longshore direction, computed simply by alongshore-averaging the time exposure image that is found as part of cBathy processing and is stored in the bathy structure (Figure 1). Using this, a weighted mean pixel intensity spectrum, GBar(f), can be found as the sum over all pixel samples of G(x, y, f)\*w(x) divided by the sum(w(x)), where the weighting, w(x), was taken as 1/IBar(x) (Figure 2). The greatest nKeep frequencies were then used for cBathy processing for ALL locations.



*Figure 1. Example coarse time exposure for 09/30/15; 1659 GMT, at Duck, NC. Estimates are made at all xm, ym locations to match the cBathy output array. The FRF pier is visible at y = 500;*

*Phase Map Tiles*

The heart of cBathy is the fitting of phase map data over a suite of tiles to accurately determine the wavenumber for each frequency. For longer records, the phase maps were found as the first eigenvector from the cross-spectral matrix and thus were a robust representation of the wave field in that frequency band that was not diluted by wave trains from multiple directions.



*Figure 2. Weighted mean spectrum (technically just the magnitude of the Fourier components rather than the square) found for the data shown in Figure 1. The asterisks show all of the Fourier frequencies in the wave band while the red asterisks are those chosen as being dominant. These were used for all spatial locations.*

For short records, the CSM cannot be computed. Instead it was decided to directly use the maps of the complex Fourier coefficients for each selected frequency. Since these maps will only be based on coefficients from a single frequency, they will be less robust and will be susceptible to error when waves are present from multiple directions. Nevertheless, the analysis seems to still be successful.

There are several other required changes with short records. Fourier phases are poorly defined when the magnitude of the Fourier coefficients is small or close to noise levels. Thus, the magnitudes of the Fourier coefficients were used a weighting for the nonlinear phase map fitting, replacing the magnitude of the magnitudes of the EOF. Similarly, there is no longer an eigenvalue to measure the relative dominance of the first EOF over noise, so this value is dropped from weighting calculations for both phase I and phase II of cBathy.

*Processing Time Issues:*

The requirement for real time output during drone flights forced an emphasis on understanding processing time for both versions 2.0 and 3.0 of cBathy. The following results are just examples for a case study at Duck, NC, on 09/30/15; 1659 GMT, chosen because there was significant wave breaking over the sand bar. The standard analysis region was for xm = [75: 12.5: 500] and ym = [0: 25: 1000]. By default, four frequencies were used (nKeep = 4). The same pixel array was used as input, consisting of 29,143 pixels spanning from y = -500 to 1500 and x = 80 to 800 (more pixels than were used) and sampled at 2 Hz for 2048 points. For short run tests, the first 300 s of these data were used. Tests were all run on a standard desktop Linux machine running parallel processing using four workers (for phase I calculations) and Matlab version 2016a.

The standard cBathy analysis for 300 s record length took 62 s to complete phases I and II (to provide depths and estimated errors for the collect). Of course, cBathy Version 3.0 for short records will still work for long records. The same analysis for the full 1024 s data set took 89 s, only 43% longer despite handling 5.7 times as much input data. This is not surprising since most of the compute time is in analyzing four phase maps for each tile.

Several tricks can be used for faster computation while a drone might still be in flight, in order to meet the requirement of real time results. The simplest is to decrease the resolution of the spatial maps from x = 12.5 m to 25 m, and y = 25 m to 50 m. This reduced processing time to 20.5 s. We can also reduce the number of frequencies analyzed from the standard of four to two (or any other number). Keeping the full resolution and only two frequencies led to a compute time of 38.8 s, while also reducing the resolution took a compute time of 13.9 s. Table I summarizes these results, but it is clear that credible products can be produced in near real time. The effect of these analysis reductions on cBathy product quality has not been systematically explored but looks quite acceptable.

|  |  |
| --- | --- |
| **Characteristics** | **Compute Time (seconds)** |
| Standard resolution, 4 frequencies | 64.2 |
| Standard resolution, 2 frequencies | 38.8 |
| Half resolution, 4 frequencies | 20.5 |
| Half resolution, 2 frequencies | 13.9 |

*Table I, computation timing for various configurations of a 300 s cBathy data set using cBathy version 3.0.*

Other points:

* In using nlinfit, we use a number of non-standard settings including the function tolerance, variable tolerance, and maximum number of iterations. In previous versions, we set these repetitively for each tile, a waste of approximately 10 s of compute tile. This has now been moved outside csmInvertKAlpha.
* Failing to change these nlinfit parameters away from default values causes a significant increase in compute time.
* Running using the parallel toolbox is very helpful. Note that matlab 2013 does not support this.
* Version 3.0 does NOT normalize the Fourier coefficient magnitudes in preBathyInput.
* Griddata was taking a long time in findKAlphaSeeds. I reduced the grid resolution from 2 m to 4 m.