Wavelet Analysis Data Set

Documentation and Final Paper
by William Chen
23 August 2013

This program calculates the mean kinetic energy (a close analogue to turbulent kinetic energy) of each single point along a grid of stream velocity data collected at the Oregon Hatchery Research Center, and appends it and all its wavelet scales and coefficients to a CSV file.

This program is highly specialized for the aforementioned Fish-ELJ data set, but with effort this may be generalized if needed.

Overview

The goal of the Fish-ELJ project is to determine the behavior of fish near log jams; one aspect of this is figuring out where fish like to reside near a log jam. Wavelet analysis combined with visualization can help us determine the distribution of energy around a log jam. The Matlab programming environment is capable of doing wavelet analysis, and taking its output data and storing it in a portable format will allow researchers to easily access wavelet analysis data in the future and do further studies.

Background

Generally, we consider wavelets to be oscillating patterns with zero mean (Torrence et al). In the case of this project, we are looking at oscillating patterns in a time series of velocities per single locations. In these patterns from our data, metrics such as velocity and energy often fluctuate rapidly over time. Using wavelet analysis can help provide us with coherent quantitative information from our noisy data.

This program extracts two fundamental pieces of wavelet information: scales and scale coefficients.

Scales are scalar quantities that represent factors of stretching and compression on a mother wavelet. They are related to the frequency property of sinusoidal waves ("Continuous Wavelet Transform").

Multiplying the scales with scale coefficients generates higher levels of detail, and composing the scales and scale coefficients onto the mother wavelet results in components of the original signal ("Continuous Wavelet Transform"). Thus, this program generates information consistent with the decomposition of its data inputs; in this case, the data is used to calculate energy, and the noisy signal of the function of energy over time is decomposed into wavelet scales and coefficients.

The mother wavelet used in this program is the Mexican Hat wavelet; this is an arbitrary choice.

Data Information/Input

There are two input files: velocity data from WinADV semicolon delimited files, and an Excel file containing position data. The Excel files are the summary statistics files. The data may be found at `\\pandora.bioe.orst.edu\Rivers\Projects\Fish_ELJ\Data\OHRC\ADV`.

Velocity data

The headers for each column in WinADV data correspond to the following:

```
"Time, seconds";"Position";"Flag";"Vx_0";"Vy_0";"Vz_0";"COR0_0" [...]
"COR1_0";"COR2_0";"SNR0_0";"SNR1_0";"SNR2_0";"AMP0_0";"AMP1_0"; [...]
"AMP2_0";"CORAvg_0";"SNRAvg_0";"AmpAvg_0"
```

The information extracted from each data file will be the file name and the three velocity components (`[...];"Vx_0";"Vy_0";"Vz_0";[...]`).

These files come from two folders, "Channel C" and "Channel D", representing different channels from which the data was collected. The data for each channel was collected at different dates.

Position data

Position data is extracted from the Excel summary statistics. For Channel C, I used `ADV_RootWad_filtered_summaries_20120319-20130425_final_edited.xlsx`, and for Channel D, I used `ADV_fulljam_filtered_summaries_20121218-20130226_final_edited.xlsx`, both of which may be found on the aforementioned server.

Running Instructions

1. Execute the program, named "mkeCalc". A file explorer dialog will appear.

2. Navigate to and select the folder containing the WinADV velocity data.
3. A matrix containing the wavelet analysis information will be available in memory. The matrix can also be found as a CSV file.

Method

Data management in code

Using Matlab's `textscan()` function and a loop, the velocity data is loaded into memory as a nested cell array. Each file in the folder specified by the user is opened, read into memory, and then closed.
To get the necessary position data, an Excel file is opened and its relevant columns are read using `xlsread()`.
We then want to create an output matrix containing all of the information we wish to output. We create an empty matrix with a variable number of columns. The columns correspond to the following information, with angled braces (`<` `>`) indicating alternation:

```
PID pos_X pos_Y pos_Z avg_MKE <wavlt_scale wavlt_coef>
```

Each file corresponds to one point in space, and thus one row in the matrix.

To fill in gaps in the WinADV velocity data, we use the `inpaint_nans()` function, which uses "sparse linear algebra and PDE discretizations" ("inpaint_nans"). See the source for details.

Calculations

For each row (where each each row corresponds to one point in space):

`PID pos_X pos_Y pos_Z`
The PID and position data columns are extracted from the Excel data loaded above. The `PID`refers to the "position identifier", `pos_x` refers to the x-coordinate of the point, `pos_y` refers to the x-coordinate, and `pos_z` refers to the x-coordinate of the point.

`avg_MKE`

The mean kinetic energy equation is $K = \frac{1}{2}\left(\overline{x^2} + \overline{y^2} + \overline{z^2}\right)$, where x, y, and z are components of velocity. The average mean kinetic energy for each point is generated from the WinADV velocity data using the mean kinetic energy equation, calculating the MKE for every time in the time series, and then further calculating the average of all of these MKE results across its respective time series. This is a just a summary value and is not used in the wavelet analysis. Rather, the unaveraged mean kinetic energy, a vector of the MKE of every point in time in each times series, is used in the wavelet analysis.

`<wavlt_scale wavlt_coef>`

For every point, the mean kinetic energy is calculated and a vector is generated; that vector is pushed through the `cwtft()` function. From the function's output, the scales can be obtained from the `scales` field, and the coefficients can be obtained from the `cfs` field ("cwtft()"). Since there are multiple coefficients for each scale, a extremely numerous space-separated tuple of the coefficients corresponding to each scale will be taken and converted into a string before being put into the output matrix. Across each row, the scales and coefficients will alternate in the matrix (see above).

Since each point will have a variable number of scales, the maximum horizontal size of the matrix may dynamically change. Since this is computationally inefficient, the horizontal size of the matrix is preset using a constant that is larger than the highest number of scales that can be expected to come out of our data set(s).

Outputs

The output matrix is written to a CSV file using the `cell2csv()` function, in the following format:

```
PID pos_X pos_Y pos_Z avg_MKE <wavlt_scale wavlt_coef>
```

Note that `wavlt_coef` columns will have a very large number of space-separated values in them. Opening up the file in Excel or other non-text editor viewer will result in these cells overflowing across the screen. For this reason, I recommend that the output format be changed in the future. I recommend exporting the matrix used to generate the CSV to a Matlab data file, or placing the data in the matrix in a relational database. This would be a good future project for computer science REU students.

Discussion

Processing of the generated data can result in the generation of a visualization of the energy spectrum of the data. Noting that energy spectrum is the distribution of energy scales over time or space (Gibson), we can potentially generate a graph of the spectrum by processing the scale data. The graph can be further extended to include distributions of fish at each scale etc.

The program is fairly slow running and may benefit from some changes. I recommend further optimization of the code to reduce running time. Additionally, future development may focus on moving the output data to a more efficient system, as CSV files may be slow and difficult to read and manipulate.

Additionally, I recommend a deeper study of wavelet analysis techniques. The process outlined above applies high level assumptions about wavelets and their properties. A less shallow look into wavelets and as Matlab's wavelet analysis functions may be fruitful in gaining more insight into the meaning of its output. It may also reveal high-level issues with the program that is the subject of this paper, whose resolution will result in a more fully informed and highly applicable data set of energy spectrum information.

Conclusion

I sincerely hope that the data set that my program generates will ultimately result in fruitful analysis of turbulence and energy spectra around log jams. I have learned very many new skills and concepts this summer and I hope to continue my studies in this area in the future.

Sources

cell2csv. (n.d.). *MathWorks - MATLAB and Simulink for Technical Computing*. Retrieved August 19, 2013, from http://www.mathworks.com/matlabcentral/fileexchange/7601-cell2csv.

Continuous Wavelet Transform - MATLAB & Simulink. *MathWorks*. Retrieved from http://www.mathworks.com/help/wavelet/gs/continuous-wavelet-transform.html

cwtft(). (n.d.). *MathWorks - MATLAB and Simulink for Technical Computing*. Retrieved August 13, 2013, from http://www.mathworks.com/help/wavelet/ref/cwtft.html.

inpaint_nans. (n.d.). *MathWorks - MATLAB and Simulink for Technical Computing*. Retrieved August 21, 2013, from http://www.mathworks.com/matlabcentral/fileexchange/4551-inpaintnans

Gibson, M. (2011). Turbulence. *Thermopedia*, doi: 10.1615/AtoZ.t.turbulence

Šavli, M. (2012). "Turbulence kinetic energy – TKE." http://mafija.fmf.uni-lj.si/seminar/files/20112012/MaticSavli2.pdf

Torrence, C., Gilbert P. Compo (1998): A Practical Guide to Wavelet Analysis. *Bull. Amer. Meteor. Soc.*, 79, 61–78.