Lecture 6
Multi-Resolution Analysis and the Wavelet Basis

Jason J. Bramburger

In the previous lecture we introduced the idea of a discrete wavelet transform. Throughout we will take $a_0 = 2$ and $b = 1$, using the notation from the end of the previous lecture. This gives the set of wavelets

$$\psi_{m,n}(t) = 2^{m/2}\psi(2^m(t - n2^{-m})) = 2^{m/2}\psi(2^m t - n)),$$

corresponding to dilations of 2^m and translations of $n2^{-m}$, for integers $(m, n) \in \mathbb{Z}^2$. The general idea of the CWT with each of the $\psi_{m,n}(t)$ functions is the following. You can start by using a wavelet that is as wide as the entire signal. This captures the low frequencies with no time localization. Then you can take two wavelets that are half the size of the previous one. These capture slightly higher frequencies and have more time localization. Then you can scale down by a factor of two again to capture higher frequencies and increase the time localization. You can continue this until some desired level (or until you run out of resolution in the time domain). This is called **multi-resolution analysis** because you are resolving the signal at different levels.

As we saw at the end of the previous lecture, these resolutions can be achieved using the discrete wavelet transform. The most important issue to resolve is whether or not a given wavelet can be used to represent a given signal or function. With $\psi_{m,n}(t)$ as written above, this amounts to checking the following two conditions:

1. *orthogonality condition*: $\langle \psi_{m,n}, \psi_{k,l}\rangle := \int_{-\infty}^{\infty} \psi_{m,n}(t)\psi_{k,l}(t)\mathrm{d}t = \delta_{m,k}\delta_{n,l}$,
2. *basis condition*: we can write functions $f(t)$ as $f(t) = \sum_{n,m=-\infty}^{\infty} \mathcal{W}_\psi[f](m, n)\psi_{m,n}(t)$,

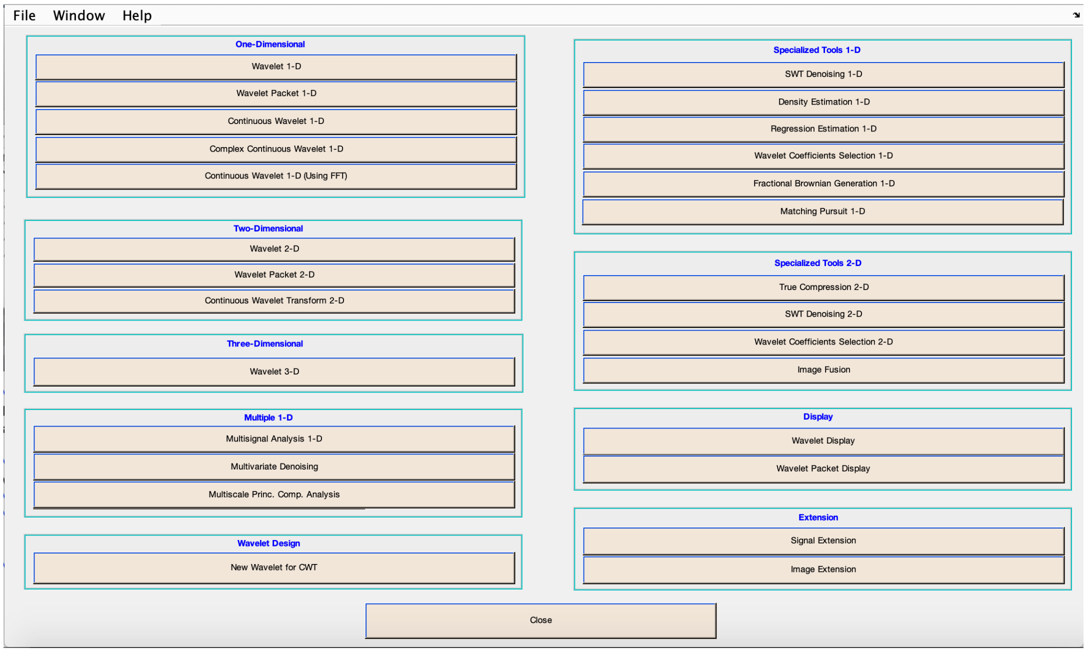
where $\delta_{i,j}$ is the Dirac delta function which returns the value 1 if $i = j$ and 0 otherwise. We won't go into the mathematical details about the above conditions, although they can be found in Section 13.6 of the suggested textbook. There are many known mother wavelets that lead to a wavelet basis $\{\psi_{m,n}\}_{(m,n)\in\mathbb{Z}^2}$ that satisfy the above conditions, most notably for us the Haar wavelet.

The one thing that you should notice about the above conditions is that they are similar to what we took advantage of with Fourier series. The difference is now that it is the multi-resolution structure of wavelets that makes them such a powerful tool. For example, imagine you have a signal that has some features that are really localized in time. Since sines and cosines are not local in time, it wouldn't make sense to try to expand this signal as a Fourier series. Instead, we can use wavelets which are localized. In particular, we can use a few low-frequency wavelets to capture the large scale structure and then some high-frequency wavelets in the ares where the high-frequency content is contained. That means that when you perform the wavelet transform, a bunch of terms are going to be zero! This makes wavelets a powerful tool for compression algorithms (wavelets are the basis for JPEG). In general, wavelets are really useful for data that takes on multiple scales. For example, if you have geological data that describes both large rock formations and small cracks and pores. The other advantage of wavelets is that there are infinitely many choices for the mother wavelet. This means that you can pick different wavelets that have ideal properties for different situations.

MATLAB's Wavelet Toolbox

MATLAB has a built-in wavelet toolbox that is an immensely powerful toolbox for signal processing, image processing, multi-resolution analysis and any manner of time-frequency analysis. To access the toolbox, simply type the following into the command window:

```
waveletAnalyzer
```



As you can see, it is almost overwhelming how many options you have here. Notice that in addition to such applications as signal processing and time-frequency analysis of 1D signals, you can also work with 2D data too.

As a specific application, click on the *wavelet 1-D* button. Under the **file** button at the top left of the GUI you can either import your own signal to analyze or load one of the examples provided to you. This example will use the noisy blocks signal under the **noisy signals - constant noise variance** subheading. In this example, a signal is provided with the full wavelet decomposition at different levels of resolution. Go to the right and change the wavelet to **haar** since this is the one we are most familiar with. Click **Analyze** to re-generate the wavelet decomposition. Descending vertically one sees that the signal is decomposed into its *big* features followed by its *finer* features. There are a number of other families of wavelets that you can use to complement your understanding of the given signal.