The **Misc** pop-up menu

Jacques Lévy Véhel

22 May 2000

This text presents a brief explanation of the functionalities the **Misc** pop-up menu.

Contents

1 Overview

This menu gathers basic utilities allowing to perform simple structures manipulations. All of them usually require only one or two commands in the matlab window, so most people having basic knowledge of matlab will find it quicker to just ignore this menu.

The structures that you may manipulate here include continuous and discrete wavelet transforms, graphs and simple matrices or vectors. A common principle in all sub-menus is that each time you need to give a name to an output structure or enter a name for an input structure, you may get it from the **Variables** list of the main window, by first selecting the structure with the appropriate name in the list, and then pressing the **Get** button in the sub-menu.

2 Create CWT

This allows to build a structure of type CWT. Enter a **Name** or **Get** it from the **Variables** list. Of course, if you get a name from the **Variables** list, the corresponding data will be replaced by the ones you are just defining. Provide the wavelet coefficients on the line **Coeff**, and give the vector of **Scale**s, as well as the one of **Frequency**.

A possible use of this sub-menu is to first compute the CWT of a given signal, and then to press the **Get all** button at the bottom of the sub-menu window. Assuming your original CWT was called *cwt_sig*, you should see on the lines **Name**, **Coeff**, **Scale**, and **Frequency** the following elements: *cwt_sig.coeff*, *cwt_sig.scale* and *cwt_sig.frequency*. You can then edit individually each of the components of the CWT structure, and then create a new CWT by pressing **Create**.

3 Create DWT

This allows to build a structure of type DWT. Enter a **Name** for your DWT structure or **Get** it from the **Variables** list, and give the name of the vector that contains the wavelet coefficients on the line **wt**. The **Index** is a vector that gives the position of the first element of each scale level in the DWT. Finally, the **Length** is a vector that contains the number of coefficients at each scale. Of course, this is just 2^j at scale j.

Let us take an example. Starting from the signal *sig* with 2^n points, compute its DWT, called *dwt_sig*. This structure contains three vectors, *dwt_sig.wt*, *dwt_sig.index* and *dwt_sig.length*. Note that the coefficients are stored scale by scale, starting from the finest one. Also, the first values in *dwt_sig.wt* are not the wavelet coefficients, but values giving, in this order, the length of the original signal, the number of octaves, and finally the values identifying the filter that defines the wavelet. The number of these values depends on the particular filter. Thus, for instance, with a Daubechies-2 wavelet, and a 6-octaves transform, you'll get that the first 6 values of *dwt_sig.wt* are 2^n, 6, 0, 1.0000, 0.7071, and 0.7071. The first value that does correspond to a wavelet coefficient of *sig* is the one with index *dwt_fBm00.index(1)*, which, in this case, is 7. *dwt_fBm00.index(2)* will be equal to 2^(j-1) + *dwt_fBm00.index(1)*, etc...

A possible use of sub-menu is to first compute the DWT of a given signal, and then to press the **Get all** button at the bottom of the sub-menu window. You can then edit individually each of the components of the DWT structure, and then create a new DWT by pressing **Create**. For instance, you could take *dwt_sig.wt* and put a threshold so that all "small" coefficients become zero (this is the principle of the wavelet denoising method).

4 Create graph

This allows to build a structure of type graph. Enter a **Name** for your graph or **Get** it from the **Variables** list (recall however that this is an output name, thus if you **get** a name, your original data with this name will be erased). Enter the name of the vector containing the abscissa on the line **Data1**, and the name of the vector containing the ordinates on the line **Data2** (or **get** them). You may also give a **Title**, **X label** and **Y label**. When you're done, hit **Create**.

5 Create matrix

This is the same as above, only since a matrix is a single element structure, you just have to give the **Name** of the structure and the one of the **Matrix** before clicking on **Create**.

6 Extract matrix

This menu allows you to extract specified lines and columns from a matrix. Give first the name of the **Input** matrix, or get it from the **Variables** list, and that of the **Output**. Enter the **first** and **last lines** and **rows**, and hit **Create**.

7 Matrix Computation

This allows to make basic operations on a single or a couple of matrices or vectors. Choose your **Input 1** matrix. You can multiply this signal by the **scale** factor, and add an **offset**. Once you have specified a first signal, you can enter a second matrix as **Input 2** (this line and the two lines below are grayed out if no **Input 1** has been entered). Again, specify **scale** and **offset** factors. Choose the name of your **Output** signal. If you specified an second input, the **operation** line becomes active, and you may choose to **add**, **sub**stract, **mul**tiply or **div**ide the two inputs. Once you're all set, hit **Apply**.