The Multiwavelet Assistant

A convenient interface to some commonly requested algorithms regarding scaling vectors and multiwavelets on $\mathbb{R}^d$ for $d = 1, 2$

Bastiaan Zapf

May 2, 2006

1 Introduction

This document explains the inner workings and user interface of `MWA`. MWA allows you to

- Edit the mask of a multiple scaling function, save or load it as a matlab file.
- Edit a corresponding multiwavelet mask, or generate a default for some scaling functions.
- Plot both the scaling function and the wavelet function.
- Estimate the regularity (smoothness) of the scaling function

All these functions are available for scaling functions on $\mathbb{R}$ and $\mathbb{R}^2$, and for arbitary scaling matrices.

2 Background

This section will give an overview over the mathematical and technical background of the program.

2.1 Mathematical Conventions

This software is based on a refinement equation of the form

$$\Phi(x) = \sum_{k \in \mathbb{Z}^d} A_k \Phi(Mx - k). \tag{1}$$

Thus, the mask has to be normalized such that the matrix

$$P := \sum_{k \in \mathbb{Z}^d} A_k \tag{2}$$

has an eigenvalue $m := |\det M|$. Moreover, we assume that P satisfies *Condition E*, i.e., P has a simple eigenvalue m and all other eigenvalues are smaller than m in modulus. If the mask is finite, this condition ensures the existence of a unique (up to multiplication with a constant) solution Φ of the refinement equation (1), see, e.g., [**?**] for details.

For regularity estimation, the scaling matrix M has to be *isotropic*, i.e., all eigenvalues $\lambda_1, \ldots, \lambda_d$ of M have to satisfy $|\lambda_1| = \ldots = |\lambda_d| = m^{1/d}$. Then, the

routine `smoothest.m` computes a lower bound s_0 such that the scaling vector Φ is contained in the Sobolev space $H^s(\mathbb{R}^d)$ for $s < s_0$. If Φ is stable, then the estimate is sharp and s_0 is the *critical Sobolev exponent* of Φ. The estimation of the regularity is based on the theory derived in [**?**], see also [**?**]. Actually, `smoothest.m` is an implementation of Theorem 7.1 in [**?**].

Furthermore, the mask must at least satisfy the *sum rules* of order 1, see [**?**, **?**] concerning the what, how, and why of sum rules. The routine `sumrules.m` can be used to estimate a lower bound of the sum rule order of a given mask. There, we use the notion of [**?**].

WARNING: Although the theory guarantees exact results, we are bound to numerical algorithms and therefore numerical inaccuracy. Since the computation of spectra of large matrices is involved in the regularity estimation, we have to choose tolerances that are far from machine accuracy (in fact, about the square root of machine accuracy). Thus, if our routines produce some dubious results, try to change these tolerances.

2.2 Plotting Scaling Vectors

To plot a scaling vector Φ corresponding to a mask $(A_k)_{k\in\mathbb{Z}^d}$, Φ has to be continuous. Using the Sobolev embedding theorem, this can be checked via our regularity estimation routine. Furthermore, we assume that the matrix P in (2) satisfies Condition E. Then, a plot of Φ can be computed via the well-known "eigenvector/eigenvalue trick", which works as follows:

Choosing $x := i \in \mathbb{Z}^d$, Equation 1 yields

$$\Phi(i) = \sum_{k\in\mathbb{Z}^d} A_{Mi-k}\Phi(k).$$

This is an eigenvector/eigenvalue problem, e.g., for the case $d = 1$ and $M = 2$ and assuming that $\operatorname{supp}\Phi \subset [0, n]$ we have

$$\begin{pmatrix} A_0 & 0 & 0 & \dots & & \\ A_2 & A_1 & A_0 & 0 & \dots & \\ A_4 & A_3 & A_2 & A_1 & \dots & \\ \vdots & & & & & \\ \dots & 0 & 0 & 0 & A_n & A_{n-1} \end{pmatrix} \begin{pmatrix} \Phi(0) \\ \Phi(1) \\ \Phi(2) \\ \vdots \\ \Phi(n-1) \end{pmatrix} = \begin{pmatrix} \Phi(0) \\ \Phi(1) \\ \Phi(2) \\ \vdots \\ \Phi(n-1) \end{pmatrix}.$$

The eigenvector of the left hand side matrix corresponding to the simple eigenvalue 1 is thus identical to $(\Phi(0), \Phi(1), \dots, \Phi(n))^T$ up to multiplication with a constant.

The next steps are easy then. Equation 1 can be used to *refine* Φ, since the values at x only depend on the values at Mx and the mask. After the desired number of iterations has been evaluated, the result is plotted.

2.3 File format

The files are supposed to be a matlab "saved struct" containing:

member name	contents
A	Coefficients as single matrices (arrays) in a cell array
r	multiplicity of the mask (dimension of a single matrix of A)
N	(maximum) dimension of the mask
d	number of non singleton dimensions (currently only 1 and 2 dimensional-masks are supported)

3 Obtaining and Installing the Program

The multiwavelet assistant is shipped as a compressed and archived directory, which contains

1. This documentation, `documentation.pdf`
2. A matlab script `MWA.m`, which serves as a wrapper to the java interface, making it possible to interact with other matlab scripts, and to load and save `.mat` files.
3. A java archive `MWA.jar`, which contains jvm code to display the interface.
4. Matlab scripts `plotWavelet.m`, `plotScalingvector.m`, `sumrules.m`, and `smoothest.m`, which perform various, not UI-related functions.

After unpacking, depending on your Matlab version, you might have to configure Matlab to use the interface.

1. Matlab 7 or higher: no configuration required
2. previous versions (with Java support): Include the MWA directory in the Matlab Java classpath. Alternatively, it is possible to move `MWA.jar` to a place that is already included in the classpath. Additionally, two lines in `MWA.m`, which only work with newer versions, have to be removed. You might have to restart matlab for these operations to take effect.
3. Versions without Java support: The assistant is written in Java. It will not work without a java-capable Matlab.

4 Running the Program

Change to the directory where `MWA.m` resides. Execute by typing `MWA` at the matlab prompt. A window should show up soon. The matlab prompt will be blocked. Should you desire to use the assistant while working with matlab otherwise at the same time, you might start another instance of Matlab and use one for the Program, and the other one for other calculations.

4.1 Hints

- There are no keys to be pressed to acknowledge inputs. What is visible in the interface is what is visible to matlab.
- By using "Mnemonics", using the interface can be faciliated and accellerated. Whenever an underlined letter appears, pressing `Alt+<letter>` activates the according function.

5 The main window

After the splash screen, you will see the main window (see Figure 1 for reference). It contains the following elements:

1. "Load Scaling Mask" – allows you to load the mask of a multigenerator which has to be stored in the `.mat` mask format. See Section 2.3 for details.
2. "Edit Scaling Mask" – pops up the editor window, containing the loaded mask (if any). See Section 7 for details.

3. dimension/scaling tabs – select the desired dimension via the "tabs", then enter the scaling matrix inside the tab. Default is 2 for the one-dimensional case and the quincunx matrix for the two-dimensional case.

   Loading a wavelet or generator mask will make the interface choose the according dimension.

4. sum rule order – input the sum rule order for regularity estimation here.

5. "Estimate" – press this button to estimate the sum rule order. Depends on the mask and the scaling Matrix.

6. output/error pane – various messages and errors will be displayed here.

7. "Plot Generator" – pops up the plot window for generators. See section 6 for details. Depends on generator mask and scaling matrix.

8. "Plot Wavelet" – pops up the plot window for wavelets. Depends on wavelet mask and scaling matrix. See section 6 for details on wavelet masks and the plot window.

9. "Estimate Regularity" – runs the regularity estimator. Depends on mask, dilation matrix and sum rule order.

6 The plot windows

The plot window appears different, depending on the circumstances. The full layout is shown in Figure 1. Some elements do not appear when the respective settings are not applicable.

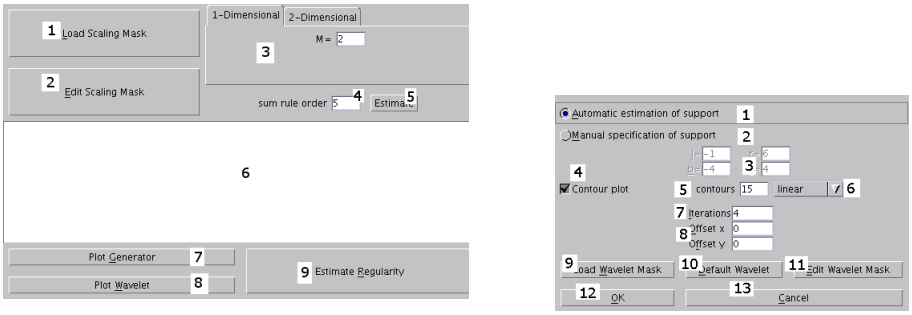
1. Automatic estimation of support produces a very coarse[1] support estimate. This is valid for the case of idempotent scaling matrices only, i.e., $M^\ell = |\det M|^{\frac{\ell}{d}} I_d$ holds for an $\ell \in \mathbb{N}$. Then, the support of Φ can be estimated via

$$\operatorname{supp}(\Phi) \subset \sum_{k=0}^{\ell-1} M^{-k} \operatorname{conv}(\operatorname{supp}(A_\beta)),$$

   where $\operatorname{conv}(S)$ denotes the convex hull of a set S, cf. [**?**].

[1] The automatic estimation usually overestimates the support by far. The best method is to use it with a small number of iterations, then to meter the support from the plot and enter it manually to plot with a better resolution.

Figure 1: The main and plot windows



2. Manual specification of support should be used if the scaling matrix is not idempotent or a large (6 or more) number of iterations is chosen. However, the plot may not be correct if the actual support is not included in the chosen interval/rectangle.

3. Support – specify the left, right, bottom and top borders of the support. To modify these entrys, select "manual specification of support".

4. Contour Plot – checking this box will render the plot as contours (`contour3`).

5. Contour Count specifies the number of contour lines.

6. Contour Zoom – modifies the spread of contour lines. Higher orders will make contours more dense near zero.

7. Iterations – specifies the number of refinings.

8. Offset entrys – specifies an offset to account for negative mask coordinates.

9. Load wavelet mask button – loads a wavelet mask

10. Default wavelet button – generates a wavelet mask from the generator mask, by switching the signs of $A_{1,2}$ and $A_{2,2}$. For interpolating wavelets, this yields a valid wavelet mask. For other wavelets, a wavelet mask has to be loaded or entered by hand.

11. Edit wavelet mask button – pops up the wavelet editor window.

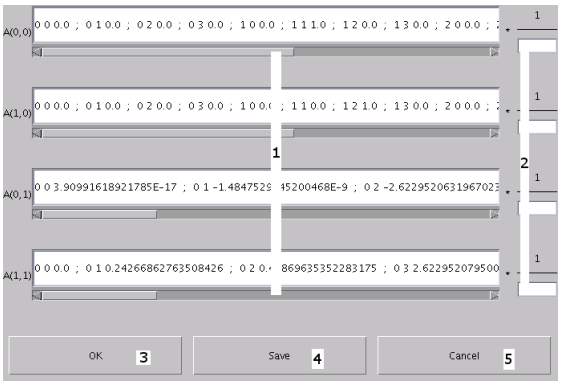
12. OK – leave and plot with the selected options.

13. Cancel – leave to the main menu, don't plot.

If the 1-d tab is selected in the main window, options 2 to 6 don't appear (automatic estimation is always optimal, contour plot is not applicable), and option 8 degenerates to a single offset parameter.

If a scaling function is to be plotted, options 9, 10 and 11 don't appear.

Figure 2: The editor window



7 The editor window

This window (cf. Figure 2) will appear to edit a mask, either the generator mask (via main window "Edit Mask") or the wavelet mask (via plot wavelet window "Edit Wavelet Mask").

1. mask values – put the mask values and their coordinates here, in the format:

   `<first coordinate> <second coordinate> <value>`

   or `<coordinate> <value>`, seperating two entries by `;`. You may or may not use scientfic notation for the values. Empty entries are silently ignored. Coordinates are shifted to non-negative values. To account for that in plotting, use the "Offset" option in the plot windows. The second coordinate is optional to facilitate entering univariate masks and defaults to 0 if missing. It is required that the input is consistently 1d or 2d.

2. mask values denominators – You can specify a denominator for each submask. If you specifiy denominators in *every* component, these will be applied to the entered values before using them. A message is displayed that indicates if the denominators were applied or not.

   This is intended to facilitate entering masks where all entries have a common denominator.

3. ok – Leave to main menu, with the displayed mask now being the active mask.

4. save – Save the displayed mask. This function will *not* set the active mask. To do so, follow "Save" with "Ok".

5. cancel – Leave to the main menu, do not touch the active mask.

References

[1] C. Cabrelli, C. Heil, U. Molter, in *Memoirs of the AMS*, vol. 170 (2004).

[2] R. Q. Jia, Q. T. Jiang, *SIAM J. Matrix Anal. Appl.* **24**(4), 1071 (2003).

[3] B. Han, *J. Approximation Theory* **124**(1), 44 (2003).

[4] Q. T. Jiang, *Trans. Amer. Math. Soc.* **351**, 2407 (1999).

[5] B. Han, *J. Approximation Theory* **110**(1), 18 (2001).

[6] S. Dahlke, W. Dahmen, V. Latour, *Appl. Comput. Harmon. Anal.* **2**(1), 68 (1995).