The IGPM Villemoes Machine

Arne Barinka[*], Stephan Dahlke[†], and Nicole Mulders

February 10, 2000

Abstract

In this note, we describe a program which can be used to estimate the Hölder regularity of refinable functions. The regularity estimates are carried out by means of the refinement mask. The theoretical background is briefly explained and a detailed description how to install and to use the program is given.

Key words: Refinable functions, Hölder regularity, wavelets.
AMS subject classification: 26B35, 39B52, 42C15, 46E35

1 Introduction

We shall be concerned with the estimation of the Hölder regularity of refinable functions. These functions play an important role in wavelet analysis and in CAGD. In many cases, they are not known analytically so that the smoothness analysis has to be carried out by means of the refinement mask. This important problem has attracted a lot of scientists in the last few years. The first results in this direction were given by I. Daubechies [2, 3] by deriving pointwise decay estimates for the Fourier transform of the refinable function under consideration. To this end, certain infinite products derived from the refinement mask have to be studied. Sharper results are available by using Littlewood–Paley techniques. This approach has been carried out

[*] The work of this author has been supported by the Volkswagen Stiftung.

[†] The work of this author has been supported by Deutsche Forschungsgemeinschaft, Grant Da 117/13–1.

by T. Eirola [4] for the univariate case and by L. Villemoes [8] for the general case, respectively. It turns out that the spectral radius of a certain matrix associated with the given refinement mask has to be computed, see Section 2 for details. Since then, many other contributions to this problem have been given, see, e.g., [1, 6, 7]. We want to point out that this list is by no means complete.

The objective of this paper is to explain and to discuss a program which can be used to treat the regularity problem. This program can be viewed as a realization of a condensed version of the approach given by Villemoes [8]. Our aim was neither to provide new theoretical insights nor to develop a commercial code. Nevertheless we wanted to write a program which is easy to install and to handle and is hopefully usefull for people working in any field related to wavelet analysis. Therefore we did focus on easy handling by e.g. including a graphical front end, then on speed.

This paper is organized as follows. In Section 2, we briefly recall the theoretical background as far as it is needed for our purposes. Then, in Section 3, we give a description of our program. First of all, in Subsection 3.1, we explain how to install it. Then, in Subsection 3.2, we illustrate how to use the program. A detailed description of the necessary steps is given. We finish we some remarks on the data organization, on trouble shooting and modification of the code in the Subsections 3.3, 3.4, and 3.5 respectively.

The program can obtained from the IGPM–homepage. Go to `http://www.igpm.rwth-aachen.de/barinka/mattoys/soft.html` and follow the instructions there. If you have any further problems or suggestions, please contact A. Barinka, `barinka@igpm.rwth-aachen.de`.

2 Theoretical Background

We want to estimate the Hölder regularity of refinable functions. In general, a function ϕ is called a *refinable* function or a *scaling* function if it satisfies a *two–scale-relation* with *mask* $\mathbf{a} := \{a_k\}_{k \in \mathbf{Z}^d} \in \ell_2$,

$$\phi(x) = \sum_{k \in \mathbf{Z}^d} a_k \phi(Mx - k), \tag{1}$$

where M is an *expanding* integer scaling matrix, i.e., all its eigenvalues have modulus larger than one. We shall always assume that supp $\mathbf{a} := \{k \in$

$\mathbf{Z}^d \mid a_k \neq 0\}$ is finite.The *symbol* of ϕ is defined by

$$m(\omega) := \frac{1}{q} \sum_{k \in \mathbf{Z}^d} a_k e^{-2\pi i \langle k, \omega \rangle}, \quad q := |\det M|. \tag{2}$$

Let us assume that the *Strang–Fix–conditions* of order L are satisfied, i.e.,

$$\left(\frac{\partial}{\partial \omega}\right)^l m(M^{-T}\rho) = 0 \quad \text{for all} \quad |l| \leq L \quad \text{and all} \quad \rho \in R^T \backslash \{0\}, \tag{3}$$

where R^T denotes a complete set of representatives of $\mathbf{Z}^d / M^T \mathbf{Z}^d$. We want to determine the Hölder regularity of ϕ,

$$\alpha^* := \sup\{\alpha : \ \phi \in C^\alpha\}.$$

It is well–known that $\alpha^* \geq \kappa_{\sup}$, where $\kappa_{\sup}$ is defined by

$$\kappa_{\sup} := \sup\{\kappa : \ \int_{\mathbf{R}^d} (1 + |\omega|)^\kappa |\hat{\phi}(\omega)| d\omega < \infty\}. \tag{4}$$

Our program can be used to estimate $\kappa_{\sup}$ from below. We employ the following result which was developed by L. Villemoes [8], see also [1, 4, 6].

Theorem 2.1 *For an integer L, let*

$$V_L := \{v \in \ell_0(\mathbf{Z}^d) \ : \ \sum_{k \in \mathbf{Z}^d} p(k) v_k = 0, \quad \textit{for all } p \in \Pi_L\},$$

where Π_L denotes the polynomials of total degree L. Assume that M is a dilation matrix with a complete set of orthonormal eigenvectors. If the symbol $m(\omega)$ according to (2) *is nonnegative and satisfies Strang–Fix–conditions* (3) *of order L, then for a suitable choice of a set Ω with supp $\mathbf{a} \subseteq \Omega$, V_L is invariant under the matrix*

$$\mathcal{H} := [q a_{Ak-l}]_{k,l \in \Omega} \, . \tag{5}$$

Let ϱ be the spectral radius of $\mathcal{H}|_{V_L}$. Then the exponent $\kappa_{\sup}$ satisfies

$$\kappa_{\sup} \geq -\frac{\log(\varrho)}{\log(|\lambda_{\max}|)}, \tag{6}$$

where $|\lambda_{\max}|$ denotes the maximum modulus of the eigenvalues of M^T.

The matrix $\mathcal{H}$ can furthermore be used to check if the refinable function ϕ under consideration is an *interpolating* function in the sense that

$$\phi(k) = \delta_{0,k}, \qquad k \in \mathbf{Z}^d. \tag{7}$$

In fact, we may use the following theorem which goes back to Lawton, Lee, and Shen [5].

Theorem 2.2 *Let $m(\omega)$ be a trigonometric polynomial which satisfies the condition*

$$m(0) = 1. \tag{8}$$

A necessary and sufficient condition for an associated continuous refinable function to be interpolatory is that the sequence δ is the unique eigenvector of the matrix $\mathcal{H}$ defined in (5) *corresponding to a simple eigenvalue 1.*

3 Description of the Program

The IGPM Villemoes Machine is a graphical user interface (GUI) written in MATLAB 5.3.0.10183 that can be used to estimate the Hölder regularity of refinable functions in one and two spatial dimensions. The IGPM Villemoes Machine will also automatically check if the resulting refinable function is interpolating by inspecting the conditions of Theorem 2.2. In the following, we want to explain how to install and use the program. Note that underlined expressions refer to elements you can see in the graphical user interfaces such as buttons.

3.1 Installation of the IGPM Villemoese Machine

After downloading `villemach.tar.gz` containing the Villmeos machine from `http://www.igpm.rwth-aachen.de/barinka/mattoys/soft.html`, the installation of the Villemoes Machine can be done in two steps:

1 Extraction
Extract `villemach.tar.gz`, e.g., by typing

```
gtar -zxvf villemach.tar.gz .
```

Thereafter the program files should all be found in the directory you extracted the files to, say ⟨*homedir*⟩ (see Section 3.3 for a complete list of included files).

Please note that `villemach.tar.gz` also contains some typical examples of masks you might want to test the Villemoes Machine with. Therefore, after extraction, there should be a subdirectory *Data* of ⟨*homedir*⟩. For further information on the data organization and the data format, please see Subsection 3.3.

2 Edit initialization–file

Use any editor to edit the file `villmach_init.m`.

- Specify the ⟨*homedir*⟩ directory containing the program files and the ⟨*datadir*⟩ directory containing the data you want to work with by doing the following: Line 14 and 15 of `villmach_init.m` look like
  ```
  homedir=(' /home/igpm/barinka/matlab/villemoes');
  datadir=(' /home/igpm/barinka/matlab/villemoes/Data');
  ```
  Edit these lines and make sure that ⟨*homedir*⟩ and ⟨*datadir*⟩ are set to the corresponding directories on your computer. The directory ⟨*datadir*⟩ does not necessarily have to be a subdirectory of ⟨*homedir*⟩. Note: Both strings should start with a blank. Please give the full path.

- In line 20 of `villmach_init.m`, you should specify the command for the postscript viewer you want to use. Default is `PSviewer='gv';`.

- Optionally you can set startup position of the lower left corner of the Villemoes Machine in the line `p_position=[` ⟨x⟩, ⟨y⟩ `];`.

This completes the installation. Now call MATLAB and run the IGPM Villemoes Machine by typing `vmm` in a MATLAB shell.

3.2 How to use the IGPM Villemoes Machine

3.2.1 The Main Window

After starting the IGPM Villemoes Machine, the main window will appear. This window is to work with already existing masks. For the organization of data, we refer again to Subsection 3.3. For the creation of new masks, or the deletion/modification of existing masks, the Villemoes Machine provides a mask editor. You can enter this tool by pressing the Create/Delete mask button at the lower right of the main window. For further details, see Subsection 3.2.2. To work with an existing mask, please switch to the main window and proceed as follows:

- First decide if you want to work in one or two dimensions by pressing the button 1D or 2D, respectively. The choosen button will appear highlighted. Default is 2D. Notice that the list Current mask shows a list of currently existing masks of the choosen dimension, i.e., all one or two dimensional masks in the data directory $\langle$*datadir*$\rangle$ (see Subsection 3.3. From this list, you can choose the mask you want to work with by clicking on it. Once selected, the name of the file is highlighted and the mask and a corresponding comment (if available) are displayed in the text window.

- Next, enter the parameters *N*1 and *N*2. They have to be chosen in such a way that the corresponding rectangle $[N_1, N_2] \times [N_1, N_2]$ contains the invariant set Ω.

- Choose the order L of Strang–Fix–conditions according to (3) in the field Strang–Fix condition:order.

- Now enter the dilation matrix *M*. By default, empty fields have the value 0.

- Then hit the Go! button. The Villemoes Machine will now estimate the Hölder regularity of the current refinable function defined by the selected mask and parameters. To this end, the matrix $\mathcal{H}$ according to (5) is assembled and the eigenvectors and eigenvalues are computed. Then, for the given value of L, it is checked which eigenvectors are contained in the space V_L. The corresponding eigenvalues are finally used to compute the spectral radius ϱ. The program also takes care of the case of multiple eigenvalues. There, special attention is necessary, because it may happen that none of the computed eigenvectors lies in the invariant space V_L, cf. (5), but a suitable linear combination does. The Villemoes Machine will also check if the refinable function is interpolating. The result is displayed in the text window. Eventually also warnings will appear there.

- In the upper right corner of the IGPM Villemoes Machine window you will find button mode that allows you to optionally select the amount of output, displayed in the MATLAB shell. There are the following selections which you can choose by click/holding on mode:

quiet	No output in the MATLAB shell (Default).
loud	Almost every step is commented, intermediate results are given.
algebra	Only the eigenvectors and eigenvalues of $\mathcal{H}$ are printed.

To avoid difficulties, please be aware of the following facts.

- Choose the parameters N_1 and N_2 as small as possible. These parameters determine the size of the matrix $\mathcal{H}$. Therefore, if you choose N_1 and N_2 to be very large, the computations may take quite a while. However, do not choose these parameters too small! For then, $[N1, N2] \times [N1, N2]$ may not contain the invariant set Ω and you will get a wrong result!

- The symbol m has to be *positive*. If this is not the case, you may use $|m(\omega)|^2$ instead of $m(\omega)$. This is the symbol corresponding to the *autocorrelation function* $\phi(\cdot) * \phi(-\cdot)$. Then the program provides you with the L_2–Sobolev exponent

$$\kappa_2 := \sup\{\kappa : \int_{\mathbf{R}^d} (1 + |\omega|^2)^\kappa |\hat{\phi}(\omega)|^2 d\omega < \infty\}. \tag{9}$$

  From this exponent, you may compute the Hölder regularity by using the Sobolev embeding theorem.

3.2.2 Create New Masks, Delete Old Ones: The Mask Editor

From the main window of the IGPM Villemoes machine, you can open the Mask Editor by pressing Create/Delete mask. This will bring you to the Mask Editor window. There you can create new masks and inspect, modify and delete old ones.

Again you first have to choose in which dimension you want to work by pressing 1D or 2D. The corresponding already existing mask will be displayed in the list on the right. When you open the mask editor, its default dimension will be the same as the chosen dimension of the Villemoes machine main window.

Creating a New Mask

To create a new mask, the Mask Editor offers you two possibilities. As an example, let us assume that you want to enter the mask of the centralized B–spline of order 2 (1D example) and the mask of the Courant finite element (2D example), say.
1D example: $a_{-1} = 1/2,\ a_0 = 1,\ a_1 = 1/2.$
2D example: $a_{0,2} = a_{0,1} = a_{1,0} = a_{1,2} = a_{2,0} = a_{2,1} = 1/2,\ a_{1,1} = 1.$

The first possible way, which is also default, is to enter the matrix of mask coefficients without specifying the coordinates in detail:

- Click/hold on the button Edit mode: and select *Edit mode: Borders and mask matrix* .
- Enter the borders of the support of your mask in the field a1, b1 and (for 2D) a2, b2
- Enter the mask in the field Enter the mask.... You should do this in standard MATLAB format. For our 1D examples, this reads as follows

  $$\begin{aligned} a1 &= -1,\ b1 = 1, \\ \mathbf{a} &= [1/2\ \ 1\ \ 1/2]; \longrightarrow (1/2\ \ 1\ \ 1/2). \end{aligned}$$

  For the 2D example, enter

  $$\begin{aligned} a1 &= 0,\ b1 = 2,\ a2 = 0,\ b2 = 2 \\ \mathbf{a} &= [1/2\ \ 1/2\ \ 0\ \ ;1/2\ \ 1\ \ 1/2\ \ ;0\ \ 1/2\ \ 1/2]; \longrightarrow \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix}. \end{aligned}$$

  Note that the column index referes to the x–direction, i.e., the columns correspond to $[a1, b1]$. To avoid difficulties, give the coefficients of the mask as precise as possible. Otherwise you may get wrong results due to round–off errors in the Villemoes algorithm.
- You may optionally enter four lines of comment that will be displayed in the Villemoes Machine. To do this, just fill out the corresponding fields.

- After you have completed your editing, specify a file name in the field Save mask as:. You can choose any valid file name $\langle name \rangle$. Finally press Save. The mask will be saved as $\langle name \rangle.m$ in the current directory, corresponding to the data directory and the dimension you have chosen. If $\langle name \rangle$ already ends on $.m$, no extra $.m$ will be appended.

The second way is to enter the mask coefficients including the coordinates.

- Click/hold on the button Edit mode: and select *Edit mode: Lattice points and mask coefficients.*
- In the field Enter..., please enter now a matrix in standard MATLAB format containing the lattice points and the corresponding mask coefficients. For our 1D examples, this reads as follows

$$\mathbf{a} \; = \; [-1 \;\; 1/2 \;\; ;0 \;\; 1 \;\; ;1 \;\; 1/2]; \longrightarrow \begin{pmatrix} -1 & 1/2 \\ 0 & 1 \\ 1 & 1/2 \end{pmatrix}.$$

For the 2D example, enter

$$\begin{aligned} \mathbf{a} \;\; = \;\; & [0 \;\; 2 \;\; 1/2 \;\; ;1 \;\; 2 \;\; 1/2 \;\; ;0 \;\; 1 \;\; 1/2 \;\; ;1 \;\; 1 \;\; 1 \;\; ;2 \;\; 1 \;\; 1/2 \;\; ;1 \;\; 0 \;\; 1/2 \;\; ; \\ & 2 \;\; 0 \;\; 1/2 \;\;]; \\ \longrightarrow \; & \begin{pmatrix} 0 & 2 & 1/2 \\ 1 & 2 & 1/2 \\ 0 & 1 & 1/2 \\ 1 & 1 & 1 \\ 2 & 1 & 1/2 \\ 1 & 0 & 1/2 \\ 2 & 0 & 1/2 \end{pmatrix}. \end{aligned}$$

- Finish the saving process as before.

Which mode you like to choose depends on the mask you want to create. The format of the file actually written by the mask editor is described in Subsection 3.3.

Inspect and Modify a Mask

- Choose the editing mode by click/holding on Edit mode:.
- Select a mask from the list of existing masks by clicking on it. The mask and the current comment as well as the filename will appear in the corresponding fields of the editor. Furthermore, the mask will be displayed in the text window. There you can check how it will appear in the Villemoes Machine main window.
- Change any of the fields until you are satisfied with your changes.
- Press Save.

Delete a Mask

- Select a mask from the list of existing masks by clicking on it.
- Press Delete.

3.3 Organization of Data

The data of the Villemoes Machine will be taken from the subdirectory ⟨*datadir*⟩ as specified in the initialization file, see Section 3.1. The one dimensional masks are stored in the subdirectory ⟨*oneD*⟩ of ⟨*datadir*⟩. Likewise ⟨*twoD*⟩ containes the two dimensional data. The data files usually start with the lines

```
global m_mask;
global m_mask_comment;
m_mask_comment={'⟨comment line 1⟩' '⟨comment line 2⟩...'};
```

The lines concerning `m_mask_comment` are optional. Then the mask is specified in the following format

dimension	format
1D	`m_mask=[`x_1 $coeff_1$; x_2 $coeff_2$; ... `];`
2D	`m_mask=[`x_1 y_1 $coeff_1$; x_2 y_2 $coeff_2$; ... `];`

As an example, here is the file for the courant finite element (`courant.m`).

```
global m_mask;
global m_mask_comment;
```

```
m_mask_comment={'This is the mask for the Courant finite element' };
m_mask=[0 0 1/2; 1 0  1/2; 0 1 1/2; 1 1 1; 2 1 1/2; 1 2 1/2; 2 2 1/2];
```

3.4 Trouble Shooting

If any error occurs like *??? Undefined function or variable...*, the Villemoes Machine is very likely confused about the directories it is working on. Make sure that the line `homedir=` in the `villmach_init.m` file is correct. It is supposed to contain the directory the program files are located in. Once the Villemoes Machine got confused, change to the Villemoes Machine home directory in the MATLAB shell. Check the error messages and make sure that no file is missing. If no data is found or wrong data appear, check the line `datadir=` in the `villmach_init.m` file. Note: Both strings should begin with a blank!

3.5 Modification of Code

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public Licence for further details.
If you don't have a copy of the GNU General Public Licence write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111, U.S.A.

The following is a list of files belonging to the IGPM Villemoes Machine.

file name	purpose
`villemach_gui.m, .mat`	graphical information for the Villemoes Machine GUI
`villmach.m`	commands (callbacks) for the Villemoes Machine GUI
`doOne.m`, `doTwo.m`	main programs
`villmach_init.m`	initialization file
`vmm.m`	startup batch
`masked_gui.m, .mat`	graphical information for the Villemoes Machine mask editor GUI
`masked.m`	commands (callbacks) for the Villemoes Machine mask editor GUI

References

[1] A. Cohen, K. Gröchenig, and L. Villemoes, Regularity of multivariate refinable functions, *Constr. Approx.* **15** (1999), 241–255.

[2] I. Daubechies, Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.* **41** (1988), 909–996.

[3] I. Daubechies, *Ten Lectures on Wavelets*, CBMS–NSF Regional Conference Series in Applied Math. **61**, SIAM, Philadelphia, 1992.

[4] T. Eirola, Sobolev characterization of solutions of dilation equations, *SIAM J. Math. Anal.* **23** (1992), 1015–1030.

[5] W. Lawton, S.L. Lee, and Z. Shen, Stability and orthonormality of multivariate refinable functions, Preprint.

[6] S.D. Riemenschneider and Z. Shen, Multidimensional interpolatory subdivision schemes, *SIAM J. Numer. Anal.* **34** (1997), 2357–2381.

[7] A. Ron and Z. Shen, The Sobolev regularity of refinable functions, Preprint, 1999, submitted.

[8] L. Villemoes, Wavelet analysis of refinement equations, *SIAM J. Math. Anal.* **25** (1994), 1433–1460.

Arne Barinka, Stephan Dahlke, Nicole Mulders
RWTH Aachen
Institut für Geometrie und Praktische Mathematik
Templergraben 55
52056 Aachen
Germany
{barinka,dahlke,nicole}@igpm.rwth-aachen.de