The *wtmm1d* package

Managing singularity spectrum computation of 1d data-series

Benjamin Audit
Laboratoire de Physique de l'Ecole normale supérieure de Lyon, France.
email : benjamin.audit@ens-lyon.fr

Contents

Chapter 1

Package wtmm1d 2.0

Package allowing to compute the different quantities involved in the Wavelet Transform Modulus Maxima method for singular signals
*** Authors and Copyright : B.Audit*

1.1 Defined types

1.1.1 Type &PF

This type is the basic type for partition function handling for the multifractal formalism and in particular the WTMM method.

- `&PF.method [= <method>]`
  Sets/Gets the method of a partition function i.e. the type of wavelet transform that was originally performed.
  (This field is used to avoid adding two pfs computed using different wavelets.)
- `&PF.amin [= <amin>]`
  Sets/Gets the smallest scale of the partition function.
- `&PF.noct`
  Gets the number of octaves of a partition function.
  (Your are not allowed to modify the octave number.)
- `&PF.nvoice`
  Gets the number of voices per octave of a partition function.
  (Your are not allowed to modify the voice number.)

- `&PF.nscale`
  Gets the number of computed scales of the partition function.
  (Your are not allowed to modify this field.)

- `&PF.sigsize [= <signal size>]`
  Sets/Gets the size of the original signals on which were computed the partition functions.
  (In the case you used the non standard addition, it is the sum of these fields. So, if you use the non standard addition on pfs that were the result of standard addition this field is not correct).

- `&PF.signum`
  Gets the number of signals that were used to compute the partition functions.
  (Your are not allowed to modify this field.)

- `&PF.qnumber`
  Gets the number of q on which were computed the partition functions.
  (Your are not allowed to modify this field.)

- `&PF.qlist [= <listv of q values>]`
  Sets/Gets the qList for which the partition functions have been computed.
  WARNING: when you set a new qList it erases all the previous results.

- `&PF.dim [= <dimension>]`
  Sets/Gets the dimension of the original data (1 for signal 2 for images) on which were computed the partition functions.

1.2 Commands to deal with the partition functions used in the Wavelet Transform Modulus Maxima method for signals

- **pf**

  - **pf** `add <pf1> <pf2> [-n]`

    Returns the sum of <pf1> and <pf2>. <pf1> and <pf2> have to be compatible for addition: they have to be the result of the same type of analysis (same WT on the same range of scale on the same number of signals of the same size). If the qlist are strickly different then it just appends the values of q, if the qlist are strickly equal it computes

the means of the partiton functions else the pf"s are not compatible.

- **pf** `cont [<pf>] <wtrans> <q-listv|q-signal> [-c]`

  Computes the continuous partition functions. It returns a new partition function if none were specified on the command line or <pf> itself.

- **pf** `copy <pfin> <pfout>`

  Old Lastwave 1.7 command.
  Not to be used, Use the generic copy function instead.

- **pf** `do1Scale <pf> <signal> [<oct> <voi>] [-c]`

  Computes the partition function for the ((oct-1)*nVoice+voi)th scale. It has to be the next uncomputed scale. The default computes the right scale.

- **pf** `get|getq|geti <type>=t|h|d|st|sh|sd <pf> <q>|<{q1 q2 ...} [-i]`

  Gets the partition functions T(q), H(q), D(q), stddev[H(q)], stddev[T(q)] or stddev[D(q)] according to <type> and q-values and returns them as one <&signal> or a <&listv> of signals. (stddev[T|H|D(q)] are the standard deviation in intensive mode, the extensive mode is not available.)
  -i : when <pf> is the result of the averaging of different partition functions it asks for the mean in intensive mode (it is the arithmetical mean of the partition functions). The default is to ask the mean in extensive mode (it is as if the partition functions had been computed on one very long signal).

- **pf** `init <pf> <method> <aMin> <nOct> <nVoice> <signalSize> <qlist>`

  Not documented.

- **pf** `read [<pf>] [<filename> | <stream>=stdin]`

  Reads a partition function from a file or a stream. (it can"t be a stream associated to string neither to the terminal). It returns <pf> or a new partition function if none were specified on the command line.

The partition function may be in ascii or binary format. It knows how to deal with big and little endian.

- **pf** `reset <pf>`

  sets all the partition functions to 0, <pf> is ready for a new computation using do1Scale.

- **pf** `write <pf> [<filename>| <stream> = stdout]`

  Writes the partition function in a file or a stream using an ascii format. (it can"t be a stream associated to string).

- **pf** `writebin <pf> <filename>`

  Writes the partition function in a file using a binary format.

- **pf** `wtmm [<pf>] <extrep> <q-listv|q-signal>`

  Computes the partition functions using the extrema representation. It is the so-called WTMM method. It returns a new partition function if none were specified on the command line or <pf> itself.

- **setpf**

  - **setpf** `amin <pf> [<amin>]`

    Sets/Gets the smallest scale of the partition functions.

  - **setpf** `method <pf> [<method name>]`

    Sets/Gets the type of wavelet transform that was performed. For now, the method is just referred to by the wavelet name. This is used in order to avoid adding two pf"s computed using different wavelets.

  - **setpf** `noct <pf>`

    Gets the number of octave of the partition functions.

  - **setpf** `nscale <pf>`

    Gets the number of computed scales of the partition functions.

  - **setpf** `nvoice <pf>`

    Gets the number of voice of the partition functions.

- **setpf** `qlist <pf> [<qlist>]`

  Sets/Gets the qList for which the partition functions have been computed.

- **setpf** `qnumber <pf>`

  Gets the number of q on which were computed the partition functions.

- **setpf** `signumber <pf>`

  Gets the number of signals that were used to compute the partition functions.

- **setpf** `sigsize <pf> [<signal size>]`

  Sets/Gets the size of the original signals on which were computed the partition functions. (In the case you used the non standard addition, it is the sum of these fields. So, if the pf"s were the result of standard addition this field is not correct).

1.3 Script Commands

• **singSpectrum** (in file `scripts/wtmm1d/wtmm1d.pkg`) `<pf> <log2ScaleMin> <log2ScaleMax>`
Computes the singularity spectrum D(h) from the partition function structure <pf>. It reads the partition function values for both h(q) and D(q) for all the available q's then performs automatic fits of the partition functions between the log scale values <log2ScaleMin> and <log2ScaleMax> in order to obtain the h(q) and D(q) values. The D(h) function is stored in the (xy) signal <signalOut>. WARNING : the fits are done automatically between <logScaleMin> and <logScaleMax>, you should first check by hand that they are not crazy fits !!

• **tauqSpectrum** (in file `scripts/wtmm1d/wtmm1d.pkg`) `<pf> <log2ScaleMin> <log2ScaleMax>`
Computes the tau(q) spectrum from the partition function structure <pf>. It reads the partition function values for tau(q) for all the available q's then performs automatic fits of the partition functions between the log scale values <log2ScaleMin> and <log2ScaleMax> in order to obtain the tau(q) values which are stored in <signalOut>. WARNING : the fits are done automatically between <log2ScaleMin> and <log2ScaleMax>, you should first check by hand that they are not crazy fits !!

1.4 Demos

Here is a list of all the Demo files and for each of them all the corresponding Demo commands. To try a Demo command, you should first source the corresponding Demo file then run the command. (When sourcing the Demo file, LastWave tells you about all the commands included in this file).
The Demo files corresponding to this package are :

Demo file **DemoWTMM1d**

• **Demo** (in file `scripts/misc/miscScripts`)

Command that explains how to run the demos of the different numerical packages

• **DemoWTMM1DBrownian** (in file `scripts/wtmm1d/DemoWTMM1d`)

Demo of the WTMM method on brownian walk (mono fractal example), averaging the result on 50 trials

• **DemoWTMM1DCantor** (in file `scripts/wtmm1d/DemoWTMM1d`)

Demonstration of the WTMM method on a non uniform triadic cantor set.

• **DemoWTMM1DDevilStair** (in file `scripts/wtmm1d/DemoWTMM1d`)

Demo of the WTMM method on a generalized devil staircase (multi fractal example), averaging the result on 50 trials

Index