The *mrw* package

Simulation/Estimation of Log-Normal Multifractal Random Walks

Emmanuel Bacry
CMAP, Ecole polytechnique, 91128 Palaiseau Cedex, France
email : lastwave@cmap.polytechnique.fr
web : http://www.cmap.polytechnique.fr/˜ bacry/LastWave

The *mrw* package was co-written by

- **Emmanuel Bacry** *CMAP, Ecole Polytechnique, 91128 Palaiseau Cedex France.*
- **Jean François Muzy**, *CNRS, Université de Corse, Quartier Grossetti, 20250, Corte, France.*

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

Version 2, June 1991

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The ”Program”, below, refers to any such program or work, and a ”work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term ”modification”.) Each licensee is addressed as ”you”.
Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.
You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.
Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.
In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.
3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail

to address new problems or concerns.
Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Contents

Part I

Documentation

Chapter 1

Using the Multifractal Random Walk (`mrw`) package

1.1 What is the mrw package

This package includes Simulation/Estimation procedures on log-normal MRW processes. Information about MRW processes can be found on *http://www.cmap.polytechnique.fr/ bacry/biblio.html*. I recommend the paper *Log-Normal continuous cascades: aggregation properties and estimation. Application to financial time-series* for a detailed review or the paper *Multifractal Random Walks* for an introduction.

1.2 Loading the package

Again, the first thing you need to do is to use this package is to load it. This is done automatically by the original `startup` file or directly by typing `package load mrw` in the terminal.

1.3 Introduction to Log-Normal MRW

The log-normal MRW is a continuous time process which has stationnary increments and exact scale invariance. The process $X(t)$ can be written as

$$X(t) = B(M(t))$$

where $B(t)$ is a Brownian motion and $M(t)$ the corresponding Multifractal Random Measure (a non-decreasing process) referred to as the MRM. The MRM $M(t)$ is obtained as the limit of a log-normal random measure :

$$M(t) = \lim_{l \to 0^+} M_l(t).$$

It has two parameters :

- λ^2 the intermittency coefficient
- T the integral scale (or the autocorrelation scale)

For the MRW, there is one more parameter

- σ the standard deviation of $B(t)$.

To build an approximated discretized version of $M(t)$ at scale 1, we build a discretized version at scale l of $M_l(t)$ for $l << 1$ and we subsample it. Same procedure is used for building the approximated discretized MRW. l is referred to as the oversampling factor.

1.4 A simple example

Let us generate a realization of an mrw process.

```
a> x = [mrw 16384 4 200 0.02 2]
```

This generates an MRW with 16384 samples, $\sigma = 2$, $\lambda^2 = 0.02$ and $T = 200$. Using the previous section notation, we chose $l = 2^4$ (oversampling factor). (Thus in order to generate such an MRW, an approximated MRM of length $16384 * 2^4$ is first build.) The option `-m` can be used to generate the MRM and the option `-M` to return a `listv` made of the MRW and the corresponding MRM. You can display the MRW :

```
a> disp MRW x
```

Now let us run the estimation procedure. It is based on GMM. The GMM method is based on the use of MRW increment. For best result we will use increment on the minimum available scale, i.e., $tau = 1$. Moreover GMM uses correlation function of the logarithm of these increments. We need to specify the lags that will be used in this correlation function (lag 0 is forbidden). We will use all the lags from 1 to 40. So in order to run the GMM algorithm we must type

```
a> res = [mrwgmm x 1:40 0 1]
```

Let us note that the procedure to find the minimum is very simple (in a future version, something more elaborate will be done). It just loops on different possible values for each parameter, except σ^2 of course. So that it loops on all the possible values for λ^2 and T. The default range are

- for λ^2 : `0.01:#30:0.04`
- for $ln(T)$: `ln(100):#30:ln(1000)`

These ranges can be changed since they are the two last optional parameters of the `mrwgmm` command. All the ranges are studied and the parameters with the minimum value is stored. Then some dichotomic iterations are performed in order to obtain more precision. The maximum number of such iterations is fixed by the third parameters pf the `mrwgmm` command. In our case we specified 0, which means that there is no maximum iteration (it automatically stops when precision is enough).

When you run the `mrwgmm` command, the intermediate estimation are displayed. The last line displayed (when the command is over) is the 5% confidence interval (assuming gaussian asymptotics is reached... which is generally not likely!). If you want to get the parameters you should type

```
a> sigma = exp(res[0][0])
a> lambda2 = res[0][1]
a> T = exp(res[0][2])
```

The covariance matrix is stored in `res[1]`.

Part II

Reference

Chapter 2

Package mrw 1.0

Package allowing simulation/estimation of log-normal MRW processes.
*** Authors and Copyright : E.Bacry and J.F.Muzy*

2.1 Commands related to the mrw package

- **gmm**

  - **gmm** `cut <param1> ... <paramN>`

    Computes a cut in the function to minimize. One of the <param...> must be a range/signal.

  - **gmm** `dcut <parameter_index> <param1> ... <paramN>`

    Computes a cut in the gradient of the function to minimize (in the direction of <parameter_index>). One of the <param...> must be a range/signal.

  - **gmm** `matrix <parameters>`

    Gets the GMM weight matrix corresponding to parameters <parameters>

  - **gmm** `moment <parameters>`

    Gets the GMM vector corresponding to parameters <parameters>

  - **gmm** `start {listv of <initParams>} [<nGMMIteration>=1] [<GMMmatrix>]`

    Starts estimation with the best parameters in the listv of <initParams> (expressed as a signal). In the case of MRW, the parameters are <sigma^ 2,lambda2,ln(T)>. This command returns the estimated parameters.

- **mrwinitgmm** `<mrw> <signal_lags> <tau> [-s]`
Initializes GMM with the mrw <mrw> using the ln of the increments on time scale <tau>. The moments that are fitted are, the variance of the mrw increments and the variance and the covariance corresponding to lags <signal_lags> of the ln of the mrw increments. (Let us note that the <signal_lags> should be an increasing signal and should not include lag 0). It returns the historical sqrt(variance). If -s then no init is made, the command only returns the sqrt(variance).

2.2 Script Commands

• **gaussprocess** (in file `LastWave_3_0/scripts/mrw/simulation`) `<auto-correlation> [<size>=4096] [<flagZero>]`
Generates a realization of size <size> of the stationnary gaussian random process (of mean 0) defined by the autocorrelation function <auto-correlation>. The closest power of 2 greater than <size> must be smaller than the size of the <auto-correlation> signal UNLESS <flagZero> is 1 in which case the autocorrelation signal is padded with 0's. The <auto-correlation> signal must code only the positive lags (starting from lag 0, i.e., coding the variance). The algorithm is optimized for <size> which are power of 2. The returned realization has the same dx as the <auto-correlation>

• **mrw** (in file `LastWave_3_0/scripts/mrw/simulation`) `[<size>=4096] [<subsample>=4] [<T>=200] [<lambda2>=0.02] [<sigma>=1] [{-m | -M}]`
Generates realization of size <size> (optimized for size which are power of 2) of a MRW process (or the correponding MRMmeasure if -m is set or both in a listv if -M) of parameter <T>, <lambda2> and <sigma>. The realization is computed using a time step of 1. The (going to 0) parameter l is 1/subsample.

• **mrwgmm** (in file `LastWave_3_0/scripts/mrw/mrwgmm`) `<mrw> <lags> [<iterMax>=0] [<tau>=1] [<lambda2Range>] [<lnTRange>]`
Perform GMM parameter estimation of the <mrw>. It is based on the increments of the mrw at scale <tau>. It uses correlaton function of the logarithm of these increments at different lags given by the signal <lags> (should not contain 0). The GMM function is evaluated for all the possible parameters lambda2 and ln(T) given by a specified range (default is 0.01:#30:0.04 for lambda2 and ln(100):#30:ln(1000) for ln(T)) only the parameters with the minimum value is stored. Then dichotomic iterations are performed (maximum number is <iterMax>). It returns a listv of the form {{ln(sigma) lambdaˆ 2 T} covariance_matrix}

2.3 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 **DemoMRW**

• **Demo** (in file `LastWave_3_0/scripts/misc/miscScripts`)

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

• **DemoMRW** (in file `LastWave_3_0/scripts/mrw/DemoMRW`)

Demonstration of the MRW package : simulation/estimation