Kussainov A.S., Kussainov S.G.

Hurst exponent estimation, verification, portability and parallelization

We present multiple software programs for the Hurst exponent calculations for a sample time series collected by a neutron monitor detectors array. The first application is carried out by the finite differences approach, using a spreadsheet-type application for a single one hour long data series; the second is a complete, one and a half week long, mathematical and graphical analysis of six acquisition channels in Matlab; the third and the fourth are the data file parser and analyzer in C/C++ compiler on Windows platform, and its modified Linux version for simultaneous, parallel computing on a virtual cluster of three machines. All applications produce the same results proving the codes' validity and portability across the operational systems and software packages.

Key words: Hurst exponent, rescaled range, parallel computing, time series, message passing interface,neutron monitor.

Құсайынов А.С., Құсайынов С.Г.

Херст экспонентасын бағалау, есептеу алгоритмінің тестілеуі, төзімділігі мен параллелденуі

Бізнейтрондықмонитормәндеріменкөрсетілгенуақыттыққатарға қолданылатын Херст экспонентасы мәнін есептеуші программалар қатарын жаздық. Соның ішінде, шеткі айырмашылықтарының реттілік аумағының ұзындығы бір сағат болатын форматтағы есептеу мысалы көрсетіліп, кейін мәліметтердің біржарым аптадағы алты канал бойынша толық графикалық анализіде көрсетілген, сонымен қатар Windows ортасында C/C++-те мәліметтер парсері жазылды, және параллель программалау үшін мұнан былай Linux арқылы басқарылатын 3 түйінді кластер ортасына ауыстырылды. Алынған нәтижелер алынған алгоритмнің дұрыстығы мен түрлі операциялық жүйелер мен программалау тілдеріне төзімділігін растай отырып, бір-бірімен өте жақсы үйлеседі.

Түйін сөздер: Херст экспонентасы, масштабталған диапазон, параллель есептеулер, уақыттық қатар, хабар беру интерфейсі, нейтрон монитор.

Кусаинов А.С., Кусаинов С.Г.

Оценка экспоненты Херста, тестирование, переносимость и распараллеливание алгоритма вычисления

Намибылнаписанрядпрограммвычисляющихзначениеэкспоненты Херстаприменительно к временному ряду представленному данными нейтронного монитора. В частности,был приведен пример вычисления в формате конечных разностей для участка последовательности длинной в один час, затем полный графический анализ данных за полторы недели по шести каналам, а также написан парсер данных на C/C++ в среде Windowsв дальнейшем перенесённый в среду дляпараллельного программирования на кластере из трех узлов под управлением Linux. Полученные результаты хорошо согласуются друг с другом, подтверждая правильность реализованного алгоритма и его переносимость на различные операционные системы и языки программирования.

Ключевые слова: экспонента Херста,масштабированный диапазон,параллельные вычисления, временной ряд, интерфейс передачи сообщений,нейтронный монитор.

UDC 53.088

[1,2]Kussainov A.S., [3*]Kussainov S.G.

[1]Physics and Technology Department,
al-Farabi Kazakh National University, Kazakhstan, Almaty
[2]National Nanotechnology Laboratory Open Type,
al-Farabi Kazakh National University, Kazakhstan, Almaty
[3]K.I. Satpaev Kazakh National Technical University, Kazakhstan, Almaty
[*]E-mail: arman.kussainov@gmail.com

HURST EXPONENT ESTIMATION, VERIFICATION, PORTABILITY AND PARALLELIZATION

Introduction

The applications of the Hurst exponent are ranging from stock market analysis [1] to electron gas modeling[2] and addressing data-statistics and system's fractalproperties. Originally, it was introduced in hydrology [3] with the purpose to construct an optimal irrigation system. Since then, multiple studies have been done including the studies of cosmic rays variations.Hurst exponent estimates are strongly dependent on the length of data sample.For example Sankar N.P. et al [4]analyzed36 years long data series on cosmic rays density covering almost three solar cycles and came to conclusion that "the present data is anti-persistent in behavior and the process is a short memory process" with the*H* value of 0.15. Flynn M.N. and Pereira W. on the contrary,studied extra short, hundred points and less, data sequences [5] and extracted vital information from a data sample on population dynamics.

Our primary goalin this workis effective parsing of raw data, reliability of results of the Hurst exponent calculations and cross platform compatibility software.

Methods

Conventional algorithmforthe Hurst exponent calculation is as follows:

Original time series of length N is divided into the sets of shorter serieswith length $n = N, N/2, N/4, \ldots, 4, 3,$ and 2 points. The upper,$n=N$,and lower,$n=2$, cutoff limits are different from study to study and depend ondata availability and the phenomena, targeted for analysis.

For each set with particular n value, and for every partial series $\{X_i\}$within this set, the following intermediate values have been calculated:

The mean value of each partial series

$$m = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad (1)$$

The mean-adjusted series derived from each $\{X_i\}$

$$Y_t = X_t - m \text{ for } t = 1, 2, \dots, n. \quad (2)$$

The standard deviation S

$$S(n) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - m)^2} \quad (3)$$

and the rescaled range$R$

$$R(n) = \max(Z_1, Z_1, \dots Z_n) - \min(Z_1, Z_1, \dots Z_n) \quad (4)$$

where the cumulative deviate series Z_n are given by the following expression

$$Z_i = \sum_{1=1}^{t} Y_i \text{ for } t = 1, 2, \dots, n. \quad (5)$$

The procedure is repeated for all possible values of *n*. Based on these *E(n)* and *n*values we have tabulated the following function

$$E\left[\frac{R(n)}{S(n)}\right] = Cn^H \quad (6)$$

The value of *H*then could be calculated from fitting the tabular data into a polynomial (see Matlab's*polyfit*data on Fig.3), or calculating the slope of the straight line *log(E)=log(C)+H*log(n)* (see Matlab's*lsqcurvefit* model plotted on Fig.2).For more elaborate and mathematically soundcalculations one may choose to work withthe generalized Hurst exponent which is directly related to fractal dimension [6].

Fig.1 shows our *H* estimates for an hour-long observation,calculated with the algorithm above. The complete 6 channels, 10 days long data analysis is shown on Fig.2-3. The results of the code adaptation to a C/C++ programming environment andparallel computation code are listed on the last-Fig.4.

Discussion

The original data wereretrieved from the NikolayPushkov'sInstitute of Earth Magnetism, Ionosphere and RadiowavesPropagation of the Russian Academy of Sciences (IZMIRAN) mobile 6NM64 supermonitor database [7].Neutron counts wereacquired at one minute interval from June the 31st, 2014 till August the 8th, 2014 in the vicinity of Moscow city.

All the steps described above in Eq.1-6 are shown in our first example in Fig.1. Here, the initial 64 data points (see column *B*) where divided into *n*=8 series each 8 points long. The following parameters have been calculated for each series: the mean values (see column *C*), cumulative deviate series Z_i(see column *D*), minimum and maximum values of this deviate series, and the range*R*(see column *F*). The last three columns*G-I* are the standard deviations*S(n)*, *R/S* for each subseries and a single value of *E* for *n*=8 which is an average over all values of *R(n)/S(n)*.

Next, using the range of the matrix tools and loop structures available in Matlab, we have calculated multiple values of *E*as a function of *n*for all 6 channels in the data file and fit them with the straight line for *log(n)*vs*log(E)* representation (see Fig.2),and with polynomial function similar to Eq.6 (see Fig.4).

Both linear and polynomial fitting models closely follow the original data points in the selected range of *n*.

Coefficients*C*, *log(C)* and *H* values are shown above each subplot, with 5 significant figures after the decimal point, though we use such precision mainly to control the algorithm performance across the different channels.No more than 2 significant figures are usually taken into considerationfor data analysis.

Microsoft Excel was used for the spreadsheet calculations, and Matlab 8.2.0.701 (R2013b) for Fig.2-3 results.

Next, we implemented our algorithm on C/C++ language with Bloodshed Dev-C++ compiler (the data is not shown for the brevity sake). Then, to address a persistent need for the effectiveparallel algorithms in data processing we designed the basic adaptation of C/C++ code to the Message Passing

Interface (MPI, and OpenMPI inour case) parallel computations environment. We have used channel-by-channel workload distribution between the threads, as shown in Fig.4. The coefficients *log(C)* and *H* with corresponding thread (process) for each individual channel, are also shown.

	A	B	C	D	E	F	G	H	I	J	K
1	n	Xi	mean mi	Yi=Xi-mi	Zt=Sum(Yi)1_t	R(n)	S(n)	R/S	E		
2	8,00	489,00	508,625	-19,63	-19,63	34,25					
3		507,00		-1,63	-21,25	-32,88					
4		497,00		-11,63	-32,88	67,13	19,49	3,44	2,6638		
5		524,00		15,38	-17,50						
6		547,00		38,38	20,88						
7		522,00		13,38	34,25						
8		487,00		-21,63	12,63						
9		496,00		-12,63	0,00						
10		531,00	530,000	1,00	1,00	1,00					
11		449,00		-81,00	-80,00	-122,00					
12		505,00		-25,00	-105,00	123,00	38,78	3,17			
13		513,00		-17,00	-122,00						
14		537,00		7,00	-115,00						
15		561,00		31,00	-84,00						
16		575,00		45,00	-39,00						
17		569,00		39,00	0,00						
18		471,00	503,000	-32,00	-32,00	11,00					
19		533,00		30,00	-2,00	-46,00					
20		471,00		-32,00	-34,00	57,00	30,31	1,88			
21		517,00		14,00	-20,00						
22		477,00		-26,00	-46,00						
23		560,00		57,00	11,00						
24		487,00		-16,00	-5,00						
25		508,00		5,00	0,00						
26		483,00	502,875	-19,88	-19,88	7,25					
27		530,00		27,13	7,25	-28,25					

Figure 1. Microsoft Excel spreadsheet calculation of a single E value for n=8 for the first acquisition channel in the data file.

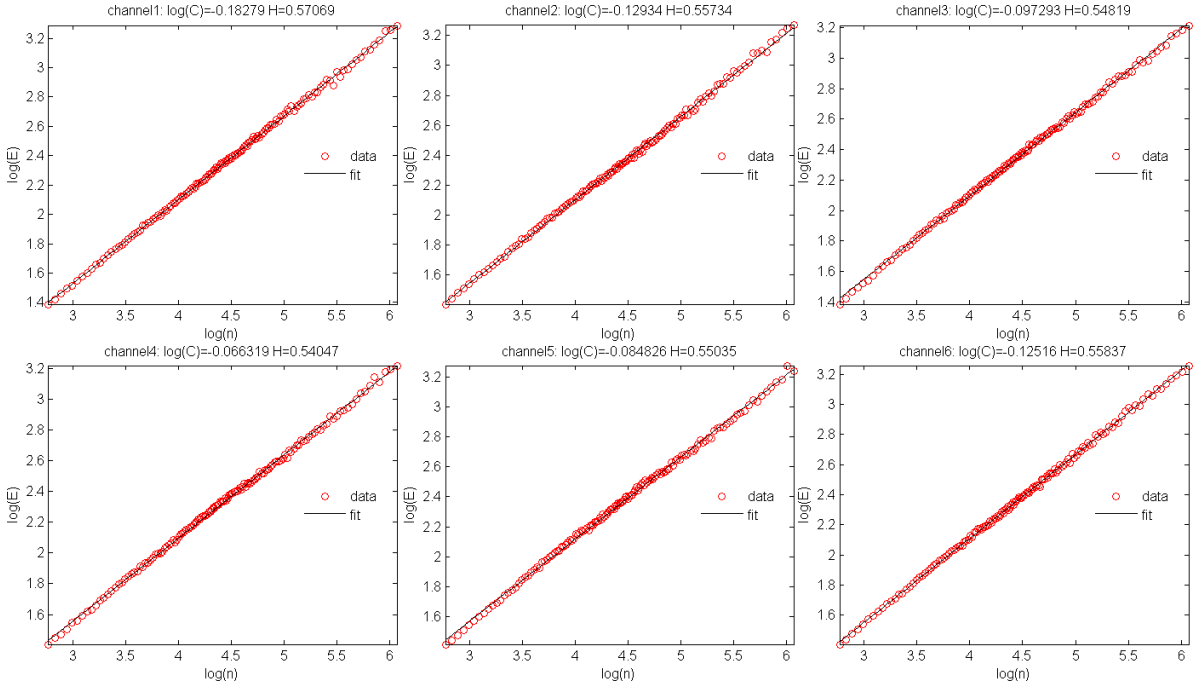

Figure 2. Matlab implementation of the 6 channels data analysis in logvslog representation and data fit by a straight line using polyfit function. Channel's number, values of log(C) and Hare printed above each subplot.

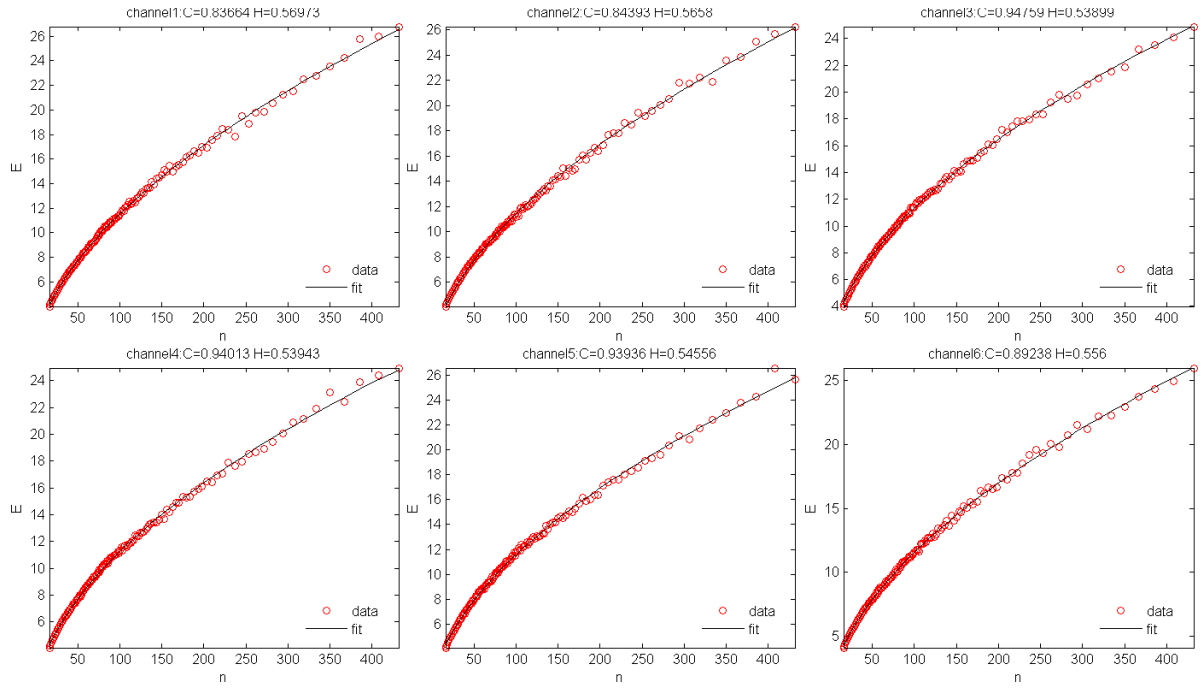

Figure 3. Matlab implementation of the 6 channels data analysis fitting with lsqcurvefit function. Channel's number, values of log(C) and Hare printed above each subplot.

```
mpiuser@debian-64bit-server: ~ (as superuser)
File  Edit  View  Search  Terminal  Help
Good morning!
changing home directory to
/home/mpiuser
compiling Aug18_2018_hurst_mpi
copying data to 10.0.2.15
Aug20_2014_hurst_mpi                                  100%  116KB 116.0KB/s   00:00
izmi!borons!31.7.2014!10.8.2014.txt                   100%  646KB 646.1KB/s   00:00
copying data to 10.0.2.16
Aug20_2014_hurst_mpi                                  100%  116KB 116.0KB/s   00:00
izmi!borons!31.7.2014!10.8.2014.txt                   100%  646KB 646.1KB/s   00:00
running 6 jobs on 3 nodes

 log(C)          H       id      hostname            wall time, sec      CPU time, sec
-0.196         0.574     0       debian-64bit-server      5.085           3.400

 log(C)          H       id      hostname            wall time, sec      CPU time, sec
-0.166         0.566     1       debian-64bit-1           5.179           3.920

 log(C)          H       id      hostname            wall time, sec      CPU time, sec
-0.080         0.549     4       debian-64bit-1           5.415           3.700

 log(C)          H       id      hostname            wall time, sec      CPU time, sec
-0.089         0.546     2       debian-64bit-2           5.560           3.920

 log(C)          H       id      hostname            wall time, sec      CPU time, sec
-0.112         0.550     3       debian-64bit-server      5.865           4.110

 log(C)          H       id      hostname            wall time, sec      CPU time, sec
-0.139         0.562     5       debian-64bit-2           6.066           4.170
Ready
mpiuser@debian-64bit-server:~$
```

Figure 4. Calculated data displayed in the terminal window of the master process in the virtual cluster. Calculated values of log(C) and H, process id, hostname and execution times for each process are given.

For the purpose of parallel computing we have configured a virtual cluster on the Oracle VM VirtualBox. The host is64 bitWindows 8.1operating system running on Intel Core i3-3220 CPU with 4 Gb of RAM. The guest operating systems are the three Linux machines with 64 bit Debian GNU/ Linux 7.4 (wheezy)with 512 Mb of RAM per each server and two nodes. Message Passing Interface is provided by OpenMPI v.1.4.5 bundled with Debian distribution.

The Hurst exponent estimates in our study are matching the majority of the previously obtained resultsfor geomagnetic indices [8] where*H*value isabove *0.5*. Variations in our computed values of the *C* and *log(C)* (seethe Fig.2-3 and Fig.4) are caused by the slightly different fitting models used for these estimates.

In addition, we have tested the code with a generated *sine* wave of the same durationas the longest neutron data sequence and with close to diurnal variations frequency. As we have anticipated, the obtained results of*H*~0.15 reflectno short memory in the series, supporting the validity of the coding.

To estimate the coding performance the "wall time" and "cpu time" have been streamed to the screen and file output for each process. To avoid possible interference between the processes the individual copies of the data files were supplied to each process/node at the beginning of the execution time by physically copying the data to the specified location, as shown in Fig.4.

Conclusion

We have demonstrated various methods of the Hurst exponent calculation on different OS and software. Basic parallel algorithm allocating the data asone channel per one thread fashion has been demonstrated as well. Further studies involve parallel algorithm optimization and interpretationin terms ofquantumalgorithms.

Our values for *H*range from 0.55 to 0.58 across all 6 channels, suggesting that at the chosen series duration, without prior noise filtering, we are pretty close to a stochastic signal. However, possibility of a long-term positive autocorrelation requires further studies.

The code is compiled to run independently on different computers and could be used as a tool to study time series of different nature and origin. Timing and optimization in this study are the subjects of further studies.

Acknowledgements

This research was in part supported by grant №2532/ГФ3 provided by the Science Committee at the Ministry of Science and Education of Republic of Kazakhstan to the principal investigator at the National Nanotechnology Open Laboratory.

We also would like to thank Dr. Yelena White, from East Georgia State College, Swainsboro, GA, USA for reading this text and helping with editing.

References

1. Zhou J., Gu G.-F., Jiang Z.-Q., et al.Microscopic determinants of the weak-form efficiency of an artificial order-driven stock market // arXiv:1404.1051.-2014.

2. Hurst J., Morandi O., Manfredi G., Hervieux P.-A. SemiclassicalVlasov and fluid models for an electron gas with spin effects // The European Physical Journal D.-2014.-Vol.68.-Issue 6.-P. 11.

3. Hurst H.E. Long term storage capacity of reservoirs // Trans. Am. Soc. Civ. Eng.-1951.-Vol. 116.-P. 770-779.

4. Sankar N.P. et al. Scaling and Fractal Dimension Analysis of DailyForbush Decrease Data // International Journal of Electronic Engineering Research.-2011.- Vol. 3.-Num. 2.-P. 237-246.

5. Flynn M, Pereira W. Ecological studies from biotic data by Hurst exponent and the R/S analysis adaptation to short time series // Biomatematica.-2013.-Vol.23.-P.1-14.

6. GorskiA.Z. et al. Financial multifractality and its subtleties: an example of DAX // Physica.-2002.-Vol. 316 .-P.496 –510.

7. The Shepetov's database in IZMIRAN at http://cr29.izmiran.ru/vardbaccess/frames-vari.htmlaccessed on August 10, 2014.

8. Pesnell W. D. Solar Cycle Predictions (Invited Review) // Sol. Phys.-2012.-Vol.281.-P.507–532.