Performance of Shell Sort in Personal Computer Using 3D Surface Analysis

Conference Paper · August 2014

DOI: 10.13140/2.1.2963.1042

3 authors:

Dipankar Das

13 PUBLICATIONS **1** CITATION

SEE PROFILE

Avik Mitra

The Heritage Academy, Kolkata

11 PUBLICATIONS **7** CITATIONS

SEE PROFILE

Arijit Chakraborty

Indian Institute of Engineering Science and T…

14 PUBLICATIONS **15** CITATIONS

SEE PROFILE

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 00 (2012) 000–000

Procedia Technology

www.elsevier.com/locate/procedia

IEMCON2014 Conference on Electronics Engineering and Computer Science

Performance of Shell Sort in Personal Computer Using 3D Surface Analysis

Dipankar Das[a,*], Avik Mitra[b], Arijit Chakraborty[c]

[a]*The Heritage Academy, Chowbaga Road, Anandapur, Kolkata, Pin: 700 107, India*
[b]*The Heritage Academy, Chowbaga Road, Anandapur, Kolkata, Pin: 700 107, India*
[c]*The Heritage Academy, Chowbaga Road, Anandapur, Kolkata, Pin: 700 107, India*

Abstract

In this research work we have analyzed the performance of shell sort taking original starting skip length policy and random starting skip length policy and have used 3D surface fitting techniques for doing the analysis. We fit the data points (Execution time versus Data size and Starting skip length) in different surfaces and observed that a 3D linear plane ($R^2 \approx 0.99$) fits the experimentally simulated dataset (in our study) for random starting skip length as opposed to a 3D full quadratic plane ($R^2 \approx 0.8$) for the experimentally simulated dataset (in our study) in case of original starting skip length.

Keywords: Surface fitting; Skip length; Gap; Residual Analysis; 3D linear surface; 3D full quadratic surface.

1. Introduction

Shell sort, is also termed as Shell's method, it is considered as an in-place sorting algorithm in a sense that it sort a list of items by comparisons, thereby can be viewed as a generalized form of bubble sort [1] and

* Corresponding author. Mobile.: +919831592063.
E-mail address: dipankr.das@gmail.com.

insertion sort [2]. The main idea behind this algorithm is to start sorting of unevenly dispersed elements and progressively reducing the distance or gap between them. Starting with farthest elements and successively moving some appropriate out-of-place elements into position where exchange with its nearest neighbor can be made quicker than an orthodox nearest neighbor exchange algorithm.

Shell sort is named after Donald Shell who published the first version Shell Sort in 1959 (Shell, 1959). The run time of Shell sort is very much dependent upon the distance vector it uses.

In this study we altered one of the general skip length policy i.e. $\lfloor n/2 \rfloor$, $n \in N$ to randomized skip length policy in an half open interval$(1, n]$, to observe the worst case run time behaviour of shell sort in personal computer using 3D surface analysis.

It is well known that algorithmic analysis of shell sort yields $O(n^2)$ in worst case and $O(n \log n)$ in best case whereas average case performance somewhat depends upon the gap sequence and this inspires us to construct this study where we tried to simulate the execution time behaviour of shell sort in worst case at randomly generated skip length. However to maintain simplicity we only considered simulated run time data while ignoring the other operating system parameters.

2. Related Work

The run-time of the Shellsort algorithm is dependent upon two input parameters: the nature and size of list of elements to be sorted; and the nature and size of list of skip-lengths, called skip length sequence. The list of elements could be sorted, unsorted, or sorted in reverse order, respectively resulting best-case, average-case and worst-case performances of the algorithm. Worst-case performance of the algorithm finds the upper bound on the execution time of the algorithm. The nature and size of skip length sequence determines the skip-length policy, where each entry in the list determines the skip-length to be used at the present iteration. It is the skip-length policy that results variations of the Shellsort algorithm.

Various skip-length policies have been proposed and the upper bound of run-time orders has been derived (Shell, 1959; Pratt, 1971; Papernov and Stasevich, 1965; Sedgewick, 1996; Incerpi and Sedgewick, 1985; Selmer, 1989; Plaxton, Poonen and Suel, 1992; Vitanyi, 2007). All these policies are geometric in nature (Plaxton, Poonen and Suel, 1992). Apart from finding run-time order in the worst-case, Weiss (1991) investigates execution-time performance of various Shellsort algorithms in average-case. A random variation of skip-lengths has been investigated by Goodrich (2011), where it is claimed that the policy can sort in O(nlog n) time with high probability, this means that it is possible that the list containing the elements is not sorted after execution of the algorithm.

Though various skip-length policies have been proposed, none of them are optimal skip-length policy, and the problem is still an open one. Moreover, the above works, mostly find the run-time order, that is, the expression for run-time is not provided. In other words, the execution-time variation for skip-length is not reported. Apart of these, the random variation of skip-length is not investigated for worst-case scenario; random variation of skip-length may yield a sequence of skip-length that can give an optimal execution and run-time performance. In the next section we address this issue and propose a skip-length policy.

3. Objectives of the Study

The objectives of this study is as follows –

(i) To find out the best surface that can be fitted to the data points (Execution Time versus Data Size, Random Starting Skip length) in case of random skip length policy in worst case

(ii) To find out the best surface that can be fitted to the data points (Execution Time versus Data Size, Original Starting Skip length) in case of original skip length policy in worst case

4. Research Methodology

4.1. Data generation

Our aim to compare the proposed skip-length policy with the original skip-length policy proposed by Shell (1959). The methodology for data generation for testing the proposed policy involved three programs: first, is a Shellsort program implemented in C programming language, named shellsort.c, that takes the number of elements to be sorted and the starting random skip-length as two command line arguments; second, is a Java program, called GenerateRandomInitialSkipLength.java, for generating random numbers which are put into a file; third, a C program, called runShellSorts.c, that automates the data generation procedure. The runShellSorts.c works as follows:

(i) It compiles and runs GenerateRandomInitialSkipLength.java so that the random numbers are put in a file called initial_r_skip_len.txt.
(ii) The random starting skip-lengths are read from initial_r_skip_len.txt and stored in a dynamic array.
(iii) The Shellsort program (compiled output of shellsort.c) is run for each of the starting skip-length obtained in step 2 and number of elements as the other command line argument. During running of the Shellsort program for each initial random skip-length, the skip-length is mapped in the interval (1, number of elements -1], since setting skip-length beyond the maximum index of the array containing the elements to be sorted can result garbage value as part of output. Moreover, each combination of skip-length and number of elements is run 100 times and the execution time is noted, because it may be possible that one run of the combination results zero execution time in case of higher machines configuration.
(iv) The number of elements is varied from 1000 to 101000 with the interval of 5000.

Data generation for the original version of Shellsort involved the same C and Java programs, with one exception: the second argument of shellsort.c is not used but present in the code, and the initial skip-length is set to the half of the first argument as described by Shell (1959), that is, the number of elements to be sorted. This approach ensures that both the programs results in nearly same machine code after compilation.

4.2. Sample dataset of random skip length policy

The experimentally obtained data set is given in the following table (Table 1).

Table 1. Sample data set of random skip length

Data Size	Random Starting Skip Length	Execution Time (Sec)	Data Size	Random Starting Skip Length	Execution Time (Sec)	Data Size	Random Starting Skip Length	Execution Time (Sec)
1000	191	3.156	36000	31079	3.984	71000	25242	4.891
6000	5617	3.438	41000	18808	4.094	76000	4351	4.953
11000	6530	3.344	46000	22818	4.156	81000	25491	5.203
16000	1081	3.5	51000	9350	4.454	86000	19321	5.172
21000	5810	3.593	56000	1747	4.39	91000	14543	5.359
26000	8320	3.75	61000	28357	4.625	96000	12052	5.438
31000	25669	3.844	66000	5134	4.797	101000	1332	5.547

4.3. Sample dataset of original skip length policy

The experimentally obtained data set is given in the following table (Table 2).

Table 2. Sample data set of original skip length

Data Size	Original Starting Skip Length	Execution Time (Sec)	Data Size	Original Starting Skip Length	Execution Time (Sec)	Data Size	Original Starting Skip Length	Execution Time (Sec)
1000	500	3.313	36000	18000	3.937	71000	35500	4.89
6000	3000	3.234	41000	20500	4.031	76000	38000	4.891
11000	5500	3.609	46000	23000	4.141	81000	40500	4.984
16000	8000	4.813	51000	25500	4.625	86000	43000	5.204
21000	10500	3.562	56000	28000	4.375	91000	45500	5.375
26000	13000	3.703	61000	30500	4.594	96000	48000	7.562
31000	15500	3.782	66000	33000	4.75	101000	50500	7.234

4.4. Model fitting

Here in the case of random skip length policy the researchers have chosen ‘Data Size’ and ‘Random Starting Skip Length’ as predictor (independent) variables and ‘Execution Time’ as the response (dependent) variable. The proposed generic model can be viewed as:

Execution Time ~ f(Data Size, Random Starting Skip Length) *(1)*

We have fitted the data points with ‘3D Linear’ model from 3D polynomial type of fit and goodness of fit statistics of this model is obtained.

In the case of original skip length policy the researchers have chosen ‘Data Size’ and ‘Original Starting Skip Length’ as predictor (independent) variables and ‘Execution Time’ as the response (dependent) variable. The proposed generic model can be viewed as:

Execution Time ~ f(Data Size, Original Starting Skip Length) *(2)*

In this case we have fitted the data points with three (3) different models from 3D polynomial type of fit and goodness of fit statistics of these models is obtained.

The model expressions of these models are tabulated below (Table 3):

Table 3. Model expressions

Model name	Model expression
3D Linear	f(x,y) = a + bx + cy
3D Simple Quadratic	f(x,y) = a + bx + cy + dx2 + fy2
3D Full Quadratic	f(x,y) = a + bx + cy + dx2 + fy2 + gxy

In this study we have used surface fitting (Surface Fitting – MATLAB & Simulink – MathWorks India, n.d.) technique for analyzing the data. In all the cases, the researchers have done the surface fitting at 95%

confidence level using 'Non Linear Least Square' method (Nonlinear Least Squares (Curve Fitting) – MATLAB & Simulink – MathWorks India, n.d.) and keeping 'Robust' (fitoptions – MATLAB & Simulink – MathWorks India, n.d.) off.

4.5. Goodness of fit statistics of the model

In this study the researchers have considered R^2, Adjusted R^2, Sum of squares due to error (SSE) and Root mean squared error (RMSE) as goodness of fit statistics. Any model which has very high R^2 and Adjusted R^2 (value close to 1) along with low SSE and RMSE (value close to 0) is considered to be a better fit (Evaluating Goodness of Fit – MATLAB & Simulink – MathWorks India, n.d.). The best model is selected on the basis of highest R^2 and Adjusted R^2 and lowest SSE and RMSE (Das, Chakraborty and Mitra, 2014).

4.6. Diagnostic procedure

In this paper for residual analysis (Graphic Residual Analysis, n.d.) the researchers have used Residual plot (Graphic Residual Analysis, n.d.; Das, Chakraborty and Mitra, 2014), Residual lag plot (Graphic Residual Analysis, n.d.; 4.4.4.4. How can I assess whether the random errors are independent from one to the next?, n.d.; Das, Chakraborty and Mitra, 2014), Histogram of the residual (Normal Probability Plot of Residuals | STAT 501 – Regression Methods, n.d.; Das, Chakraborty and Mitra, 2014) and Q-Q plot of the residuals (Bandyopadhyay, 2013; Das, Chakraborty and Mitra, 2014).

4.7. Software used

The software used for data generation and data analysis is given in the following table (Table 4).

Table 4. Software used

Component Name	**Component Details**
Operating System	Windows XP Professional SP2
Compiler	Dev-C++ 4.9.9.2; Java 6.0 (for generating the random number)
Software for execution-time analysis	MS Excel, MATLAB, SPSS

4.8. Hardware platform

The hardware platform used for generating the data points is given in the following table (Table 5).

Table 5. Hardware platform

Component Name	**Configuration**
CPU	Intel Mobile Core 2 Duo T6570
Frequency	2.10 GHz and 2.09 GHz
L1 Cache (Data)	2 X 32 KB 8-way set associative
L1 Cache (Instruction)	2 X 32 KB 8-way set associative
L2	2048KB 8-way
RAM	3GB (DDR 3)

5. Data Analysis & Findings

5.1. Goodness of fit statistics of the surface using random skip length policy

The summary of goodness of fit statistics is given in the following table (Table 6).

Table 6. Goodness of fit statistics of the surface using random skip length

Model	SSE	R^2	Adjusted R^2	RMSE
3D Linear	0.08181	0.9928	0.992	0.06742

5.2. Goodness of fit statistics of the surface using original skip length policy

The summary of goodness of fit statistics is given in the following table (Table 7).

Table 7. Goodness of fit statistics of the surface using original skip length

Model	SSE	R^2	Adjusted R^2	RMSE
3D Linear	7.1414	0.7160	0.7295	0.5832
3D Simple Quadratic	4.6772	0.8140	0.7675	0.5407
3D Full Quadratic#	4.6772	0.8140	0.7520	0.5584

Best model.

5.3. Diagnostic procedure of the surface using random skip length policy

In this subsection we have performed the graphical residual analysis of '3D Linear' model which is illustrated below.

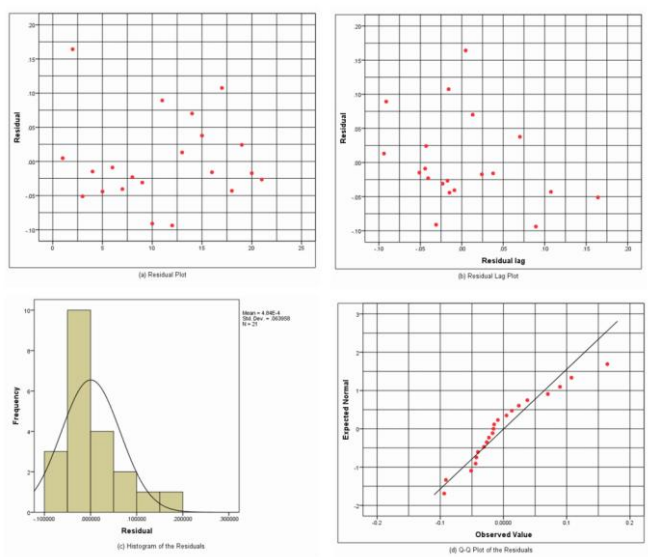

Fig. 1. (a) Residual Plot; (b) Residual Lag Plot; (c) Histogram of the Residuals; (d) Q-Q Plot of the Residuals

We observe a horizontal band pattern in the Residual Plot (Fig. 1. a) which indicates that the variance of residuals is constant. There is no pattern or structure in the Residual Lag Plot (Fig. 1. b) from which we may conclude that the errors are independent. We obtain a symmetric bell shaped histogram of the residuals (Fig. 1. c) which is evenly distributed around zero suggesting that the residuals are normally distributed. The points on the Q-Q plot (Fig. 1. d) are approximately linear from which we may conclude that the residuals follow approximately random distribution.

5.4. Diagnostic procedure of the surface using original skip length policy

In this subsection first we have performed the graphical residual analysis of the best model (identified in the subsection 5.2 i.e. '3D Full Quadratic').

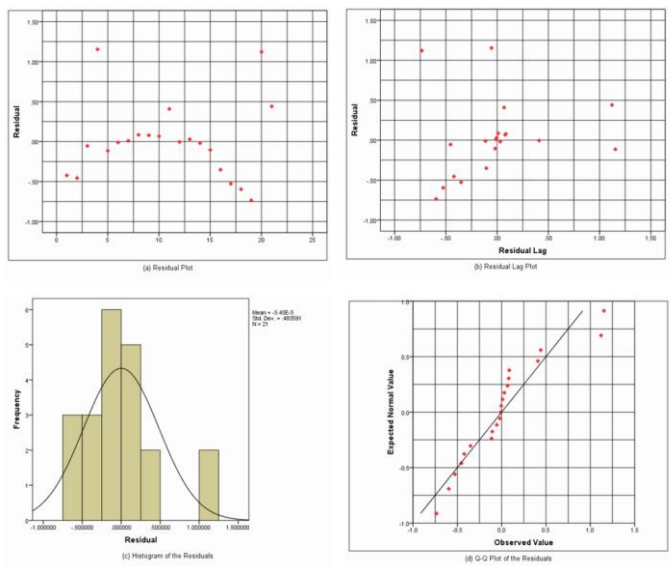

Fig. 2. (a) Residual Plot; (b) Residual Lag Plot; (c) Histogram of the Residuals; (d) Q-Q Plot of the Residuals

We observe a horizontal band pattern in the Residual Plot (Fig. 2. a) which indicates that the variance of residuals is constant. There is no pattern or structure in the Residual Lag Plot (Fig. 2. b) from which we may conclude that the errors are independent. We obtain a symmetric bell shaped histogram of the residuals (Fig. 2. c) which is evenly distributed around zero suggesting that the residuals are normally distributed and the points on the Q-Q plot (Fig. 2. d) are approximately linear from which we may conclude that the residuals follow approximately random distribution.

5.5. Proposed mathematical model and surface using random skip length policy

From the goodness of fit statistics given in subsection 5.1 and residual analysis performed in the subsection 5.3 we can conclude that the model '3D Linear' fits the data well. The proposed mathematical model representing the '3D Linear' model is as follows –

$$f(x,y) = 3.127 + 2.422e\text{-}005*x + 2.61e\text{-}007*y \qquad (3)$$

The surface of the proposed model is shown in Fig. 3.

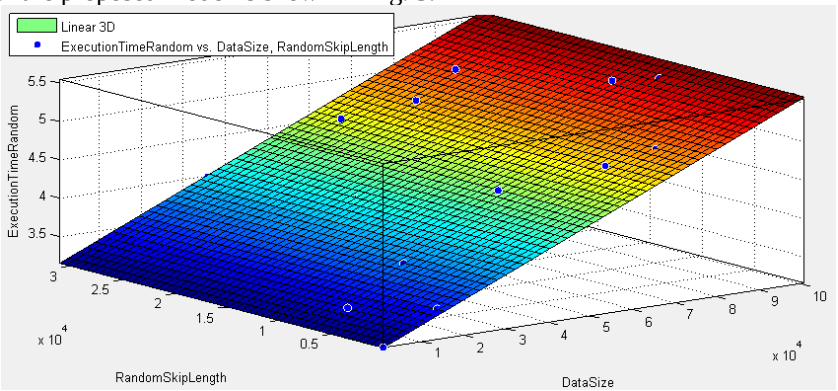

Fig. 3. Surface of 3D Linear model using random skip length policy

5.6. Proposed mathematical model and surface using original skip length policy

From the goodness of fit statistics given in subsection 5.2 and residual analysis performed in the subsection 5.4 we can conclude that the model '3D Full Quadratic' reasonably fits the data. The proposed mathematical model representing the '3D Full Quadratic' model is as follows –

$$f(x,y) = 3.747 - 1.222e\text{-}005*x + 8.052e\text{-}008*y + 3.194e\text{-}010*x\text{^}2 + 7.985e\text{-}011*y\text{^}2 + 1.597e\text{-}010*x*y \quad (4)$$

The surface of the proposed model is shown in Fig. 4.

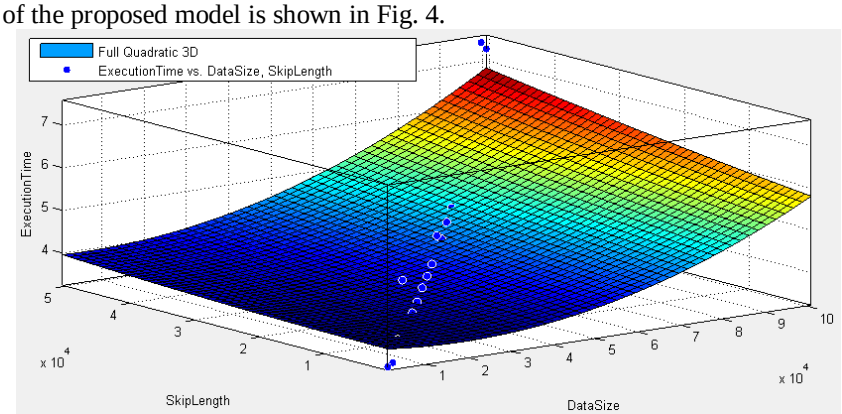

Fig. 4. Surface of 3D Full Quadratic model using original skip length policy

6. Limitations & Future Scope

Generating data in order to perform 3D surface analysis plays a pivotal role and we did the same by simulating these algorithms, however various factors that affect execution time of any sorting algorithm (in this case shell short) are many i.e. context switch time, L1,L2 cache size, family of processors etc. Considering these contributory factors at the maxim is beyond the scope of this study and in order to maintain simplicity we ignored these factors and it will be one such interesting scope in future.

We are very much keen on running this algorithm on various hardware and software platforms and it shall be certainly our future endeavor in this direction to obtain and analysis data obtained from these heterogeneous platforms in order to observe such possible interesting result.

7. Conclusion

We have proposed a random skip-length policy for Shellsort where the initial value of the skip-length is a random integer greater than 1 and less than or equal to the maximum index of the array containing the elements to be sorted, and then halving the skip-length value at each iteration, until it becomes 1. We have employed 3D surface fitting technique and investigate the best surface that can be fitted to the experimentally simulated data set for both the variants of Shellsort in their worst cases i.e. random and original skip length policy proposed by D. Shell. We have observed that both the datasets follows definite patterns. For the proposed random skip-length policy, we observed that 3D linear plane fits the data well, whereas, in case of original skip-length policy, 3D full quadratic plane fits the data well. This suggests that the proposed skip-length policy performs better than the original skip-length policy. The linear fit for the proposed skip-length policy only suggests that the rate of increase of execution time in the range of input data is very less, and does not disprove that the execution time is less than O(n log n), since Shellsort and its variants are comparison sorts.

As a part of our future work, we shall investigate the question: "Does all the random starting skip length policies will result a 3D linear fit in all the platforms for this range of input, if it is not, then is there any specific patterns of skip-lengths that may give such results?"

References

Shell, D.L., 1959. A high-speed sorting procedure, Communications of ACM, Volume 2, Issue 7, pp. 30 – 32.

Pratt, V., 1971. Shellsort and Sorting Networks, PhD Thesis Stanford University.

Papernov, A.A., Stasevich, G., 1965. A method of information sorting in computer memories, Problems of Information Transmission, Volume 1, Issue 3, pp. 81 – 98.

Sedgewick, R., 1996. "Analysis of Shellsort and Related Algorithms," Fourth Annual European Symposium on Algorithms, pp. 1 – 11.

Incerpi, J., Sedgewick, R., 1985. Improved Upper Bounds on Shellsort, Journal of Computer System Sciences, Volume 31, Issue 2, pp. 210 – 224.

Selmer, E.S., 1989. On Shellsort and the Frobenius problem, BIT Numerical Mathematics, Volume 29, Issue 1, pp. 37 – 40.

Plaxton, C.G., Poonen, B., Suel, T., 1992. "Improved Lower Bounds for Shellsort," Proceedings of 33rd Annual Symposium on Foundations of Computer Science, pp. 226 – 235.

Vitanyi, P., 2007. Analysis of Sorting Algorithms by Kolmogorov Complexity (A Survey), Entropy, Search, Complexity, Bolyai Society Mathematical Studies, Volume 16, pp. 209 – 232.

Weiss, M.A., 1991. Empirical Study of the expected running time of Shellsort, The Computer Journal, Volume 34, Issue 1, pp. 88 – 91.

Goodrich, M.T., 2011. Randomized Shellsort: A Simple Data-Oblivious Sorting Algorithm, Journal of ACM, Volume 58, Issue 6, Article no. 27.

Surface Fitting – MATLAB & Simulink – MathWorks India, MathWorks, (website): http://www.mathworks.in/help/curvefit/surface-fitting.html. Accessed May 10, 2014.

Nonlinear Least Squares (Curve Fitting) – MATLAB & Simulink – Math-Works India, MathWorks, (website): http://www.mathworks.in/help/optim/nonlinear-least-squares-curve-fitting.html. Accessed May 10, 2014.

fitoptions - MATLAB & Simulink – MathWorks India, MathWorks, (website): http://www.mathworks.in/help/curvefit/fitoptions.html. Accessed May 10, 2014.

Evaluating Goodness of Fit – MATLAB & Simulink – MathWorks India, MathWorks, (website): http://www.mathworks.in/help/curvefit/evaluating-goodness-of-fit.html. Accessed May 10, 2014.

Das, D., Chakraborty, A., Mitra, A., 2014. Sample Based Curve Fitting Computation on the Performance of Quicksort in Personal Computer, International Journal of Scientific & Engineering Research, Volume 5, Issue 2, pp. 885 – 891.

Graphic Residual Analysis, OriginLab, (website): http://www.originlab.com/www/helponline/origin/en/UserGuide/Graphic_Residual_Analysis.html. Accessed May 10, 2014.

4.4.4.4. How can I assess whether the random errors are independent from one to the next?, Engineering Statistics Handbook, (website): http://www.itl.nist.gov/div898/handbook/pmd/section4/pmd444.htm. Accessed May 10, 2014.

Normal Probability Plot of Residuals | STAT 501 – Regression Methods, PENNSTATE, (website): https://onlinecourses.science.psu.edu/stat501/node/40. Accessed May 10, 2014.

Bandyopadhyay, G., 2013, Modeling NPA Time Series Data in Selected Public Sector Banks in India with Semi Parametric Approach, International Journal of Scientific & Engineering Research, Volume 4, Issue 12, pp. 1876 – 1889.