ISSN 2229-5518

Sample Based Curve Fitting Computation on the Performance of Quicksort in Personal Computer

Dipankar Das, Arijit Chakraborty, Avik Mitra

Abstract— In this paper we have used curve fitting technique for analyzing the classical Quicksort algorithm and its performance in worst case on personal computer. The proposed generic model can be viewed as: Time ~ f (Number of Elements). We fit the data points (Time vs Number of elements) in twenty one different models from various types of fit such as Polynomial, Exponential, Power, Gaussian and Fourier. This analysis leads us to identify the best model amongst these models. We found that a model of Fourier series is the best fit.

Index Terms— Curve Fitting, Experimental Algorithmics, Fourier Fit, Graphical Residual Analysis, Performance Analysis, Quicksort, Worst Case

1 Introduction

COMPUTATIONAL experiments on the performance of algorithms give us further insight into their behaviors in real world scenarios and supplement their theoretical analysis. Algorithmic work and experimentation are combined in Experimental algorithmics [1].

There is a trend in research community to measure performance of complex algorithms by employing various models. The situation is not different for common sorting algorithms as well; curve fitting gives us visual information to look upon the situation in a subtle way, based on the motivation of easily available computing devices like personal computers, laptops etc and softwares like GCC, MATLAB etc.

Performance analysis of Quicksort which is one of the popular sorting algorithms [2] in a simulated environment will give us a fair idea about various dimensions of the algorithm.

2 Related Work

There are two ways of analyzing Quicksort algorithm [34] by C.A.R. Hoare.

- Theoretical [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]
- Empirical [14], [15], [16]

Theoretical analyses are done with respect to the following *parameters*:

- Run-time [3], [4], [5]
- Probabilistic distribution on the number of comparisons required during sorting [7], [9]
- The limit theorem on the distribution [8], [10], and deviations from that limit for finite number of elements [11], [12].

Empirical analyses are based on the response time of Quicksort to patterns (i.e. already arranged, partially arranged, reverse arranged) in the list of elements that is to be sorted *i.e.* adaptive ability [14] of Quicksort.

Analyses of variations [15] of Quicksort like Qsorte [25], qsort7 [26] etc are based on:

- Number of comparisons and swaps required during running of Quick sort [14], [15].
- Comparative analysis of pivoting schemes [14], [15].
- Affect of computer configuration like type of microprocessor, L1 cache miss events, and *branch miss-predictions by the microprocessors etc* [14].

Several developments of Quicksort have been performed. These developments include how well the Quicksort be tuned for parallel processing environment like distributed memory system [19], graphics processing unit [18], or online version of Quicksort [20]; in the online version of Quicksort, called ***Quick sort on the fly***, the first element in the sorted arrangement is found and put in the output before the rest of the elements are sorted, thus, the output appears as a stream. A generalization of Quicksort has also been explored where respective the authors proposed a solution of the problem of sorting by considering the first k largest elements in a set of n elements [24].

Other types of developments include proposals for new single pivot selection schemes [21], [22], and multiple pivot selection schemes [21], [23]; in [23] three pivots are considered for reducing the run-time of Quicksort in average case.

3 Objectves Of The Study

We tried to identify the best curve that can be fitted to the data points (Time vs Number of elements) generated by simulating Quicksort algorithm for the worst case.

The objective of this study is to propose a generic mathematical model using large dataset that can explain the behavior of Quicksort for worst case in a personal computer.

- *Dipankar Das is currently working as a Faculty member in The Heritage Academy, Kolkata, India, PH-919831592063. E-mail: dipankr.das@gmail.com*
- *Arijit Chakraborty is currently working as a Faculty member in The Heritage Academy, Kolkata, India, PH-919830722218. E-mail: arijitchakraborty.besu@gmail.com*
- *Avik Mitra is currently working as a Faculty member in The Heritage Academy, Kolkata, India, PH-918902727029. E-mail: avik.mitra2@gmail.com*

International Journal of Scientific & Engineering Research, Volume 5, Issue 2, February-2014 886
ISSN 2229-5518

4 Research Methodology

4.1 Sample Data Set

Open source platform and ANSI C have been used to generate the experimental data. The sample data set is given in the following table (TABLE 1):

TABLE 1
SAMPLE DATA POINTS

Number of elements	Time (Seconds)
1000	0
5000	0.034
10000	0.136
15000	0.305
20000	0.54
25000	0.843
30000	1.219
35000	1.659
40000	2.166
45000	2.741
50000	3.517
55000	4.176
60000	4.875
65000	5.727
70000	6.64
75000	7.623
80000	8.677
85000	9.823
90000	10.982
95000	12.24
100000	13.565

4.2 Model Fitting

In this case, the researchers have chosen 'Time' as the response (dependent) variable and 'Number of elements' as the predictor (independent) variable for analyzing the data.

Thus the proposed generic model can be written as:

$$Time \sim f(Number\ of\ elements) \quad (1)$$

The data points have been fitted with twenty one different types of models from different types of fit (Polynomial, Exponential, Gaussian, Power and Fourier) and the goodness of fit statistics of these models are calculated.

The model expressions of these models have been tabulated below in the following tables (TABLE 2 – TABLE 6):

TABLE 2
POLYNOMIAL MODEL EXPRESSIONS IN MATLAB

Model Name	Model Expression
Linear	y = p1*x + p2
Quadratic	y = p1*x^2 + p2*x + p3
Cubic	y = p1*x^3 + p2*x^2 + p3*x + p4
Polynomial of degree 4	y = p1*x^4 + p2*x^3 + p3*x^2 + p4*x + p5
Polynomial of degree 5	y = p1*x^5 + p2*x^4 + p3*x^3 + p4*x^2 + p5*x + p6

TABLE 3
EXPONENTIAL MODEL EXPRESSIONS IN MATLAB

Model Name	Model Expression
Exponential1	f(x) = a*exp(b*x)
Exponential2	f(x) = a*exp(b*x) + c*exp(d*x)

TABLE 4
POWER MODEL EXPRESSIONS IN MATLAB

Model Name	Model Expression
Power1	f(x) = a*x^b
Power2	f(x) = a*x^b+c

TABLE 5
GAUSSIAN MODEL EXPRESSIONS IN MATLAB

Model Name	Model Expression
Gaussian 1	y = a1*exp(-((x-b1)/c1)^2)
Gaussian 2	y = a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2)
Gaussian 3	y = a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2) + a3*exp(-((x-b3)/c3)^2)
Gaussian 4	y = a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2) + a3*exp(-((x-b3)/c3)^2) + a4*exp(-((x-b4)/c4)^2)
Gaussian 5	y = a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2) + a3*exp(-((x-b3)/c3)^2) + a4*exp(-((x-b4)/c4)^2) + a5*exp(-((x-b5)/c5)^2)

TABLE 6
FOURIER MODEL EXPRESSIONS IN MATLAB

Model Name	Model Expression
Fourier1	f(x) = a0 + a1*cos(x*w) + b1*sin(x*w)
Fourier2	f(x) = a0 + a1*cos(x*w) + b1*sin(x*w) + a2*cos(2*x*w) + b2*sin(2*x*w)
Fourier3	f(x) = a0 + a1*cos(x*w) + b1*sin(x*w) + a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w)
Fourier4	f(x) = a0 + a1*cos(x*w) + b1*sin(x*w) + a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) + a4*cos(4*x*w) + b4*sin(4*x*w)
Fourier5	f(x) = a0 + a1*cos(x*w) + b1*sin(x*w) + a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) + a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w)
Fourier6	f(x) = a0 + a1*cos(x*w) + b1*sin(x*w) + a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) + a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) + a6*cos(6*x*w) + b6*sin(6*x*w)
Fourier7	f(x) = a0 + a1*cos(x*w) + b1*sin(x*w) + a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) + a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) + a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w)

In this study all the curve fitting has been performed using 'Non linear least square' [35] method, keeping 'Robust' [36] off. The researchers have used 'Trust-Region' [37] algorithm for curve fitting. The entire analysis was performed at 95% confidence level.

4.3 Model Selection Based on Goodness of Fit Statistics

For the purpose of this study the following goodness of fit statistics have been considered:

a. R^2
b. Adjusted R^2
c. Sum of squares due to error (SSE)
d. Root mean squared error (RMSE)

A better fit is indicated by high value of R^2 and Adjusted R^2 (value close to 1). Similarly, low value of SSE and RMSE (value close to 0) also indicate a better fit [27].

Decision rule:

The best model is chosen based on highest R^2 and Adjusted R^2 value (close to 1) and lowest values of SSE and RMSE (close to 0).

4.4 Diagnostic Procedure

In statistical modeling residual analysis plays an important role. The quality of a regression can be assessed by using residual plots [28].

Residual is defined as:
Residual = observed value – predicted value

In general residuals are a form of errors and the same general assumptions of errors can be applied to the residuals too. It is expected that the residuals should be (roughly) normal and (approximately) independently distributed with a mean of 0 and some constant variance [29].

In this study, the following graphical residual analysis methods have been used to assess the quality of regression:

a. Residual vs predictor plot
b. Residual plot
c. Residual lag plot
d. Histogram of the residuals
e. Q-Q plot of the residuals

a. Residual vs predictor plot –

To assess sufficiency of the functional part of the model this plot is used. If the residuals in the plot do not exhibit any systematic structure then it indicates that the model fits the data well [30].

b. Residual plot –

This plot is used to check the error variance. If the residual plot has an increasing/ decreasing trend then it indicates that the error variance increases/ decreases with the independent variable. A horizontal-band pattern indicates that the variance of the residuals is constant [28].

c. Residual lag plot –

This plot is used for checking the independence of the error term. It is a scatter plot of residuals (i) on the y axis and the residual (i-1) values on the x axis. Any random pattern in a lag plot suggests that the variance is random [28]. If there is no pattern or structure in the plot i.e. the points appear to be randomly scattered then it suggests that the errors are independent [31].

d. Histogram of the residuals –

This is test for residual normality. A symmetric bell shaped histogram which is evenly distributed around zero suggests that the residuals are normally distributed [32].

e. Q-Q plot of the residuals –

A Q-Q plot is a probability plot. It is a graphical method for comparing two probability distributions by plotting their quantiles against each other. If the points on the plot are linear then it suggests that the randomly generated data are normally distributed [33].

4.5 Software Used

We have used Fedora 14 (Kernel Version 2.6), ANSI C for generating experimental data and for the purpose of the data analysis of this paper MATLAB 7.7.0, SPSS 17.0 and MS Excel have been used.

ISSN 2229-5518

4.6 Hardware Platform

The Quicksort algorithm has been executed and simulated in a hardware platform which is as follows:

- Processor: Intel Core 2 Duo CPU T6570 @2.10 GHz
- RAM: 3GB; DDR-3; frequency-399.0 MHz
- L1 Data Cache Size: 32 KB (8-way set associative)
- L1 Instruction Cache Size: 32 KB (8-way set associative)
- L2 Cache size: 2048 KB (8-way set associative)

5 Data Analysis & Findings

The twenty one fitted models are first evaluated using the goodness of fit statistics and the best model amongst the fitted models has been identified based on the decision rule as discussed in sub section 4.3.

A summary of goodness of fit statistics is given in TABLE 7.

TABLE 7
Goodness Of Fit Statistics

Model Name	SSE	R^2	Adjusted R^2	RMSE
Linear	24.8430	0.934585	0.931142	1.1435
Quadratic#	0.0190	0.999950	0.999944	0.0325
Cubic#	0.0190	0.999950	0.999941	0.0334
Polynomial of degree 4#	0.0171	0.999955	0.999944	0.0327
Polynomial of degree 5#	0.0171	0.999955	0.999940	0.0338
Exponential 1	8.0181	0.978887	0.977776	0.6496
Exponential 2	10.0154	0.973628	0.968974	0.7676
Power1	0.0190	0.999950	0.999947	0.0317
Power2	0.0189	0.999950	0.999945	0.0324
Gauss1	0.6341	0.998330	0.998145	0.1877
Gauss2	0.0427	0.999888	0.999850	0.0534
Gauss3	0.2277	0.999400	0.999001	0.1377
Gauss4	0.0365	0.999904	0.999786	0.0637
Gauss5§	0.0281	0.999926	0.999754	0.0684
Fourier1	0.0190	0.999950	0.999941	0.0335
Fourier2	0.0171	0.999955	0.999940	0.0337
Fourier3	0.0141	0.999963	0.999943	0.0329
Fourier4	0.0108	0.999972	0.999948	0.0313
Fourier5	0.0077	0.999980	0.999955	0.0292
Fourier6	0.0061	0.999984	0.999954	0.0296
Fourier7Φ	0.0022	0.999994	0.999976	0.0212

Φ Best model.

 # Equation is badly conditioned.

§ Fit computation did not converge.

5.2 Diagnostic Procedure

In this subsection the graphical residual analysis of the best model (identified in sub section 5.1) has been performed.

a. Residual vs predictor plot –

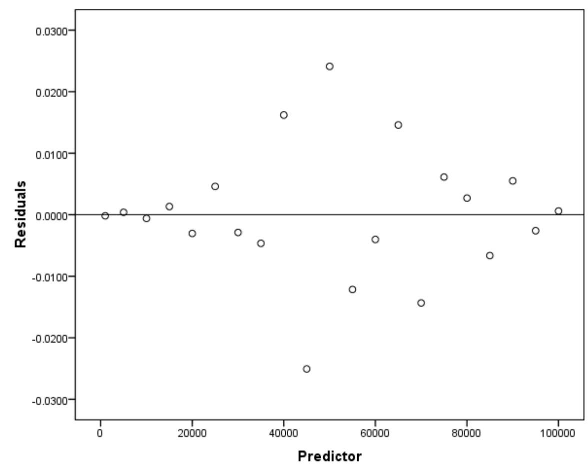

Fig. 1. Residual vs predictor plot of Fourier7 model.

From the above figure (Fig. 1.) it is evident that the residuals appear to behave randomly. Therefore, it suggests that the model fits the data well.

b. Residual plot –

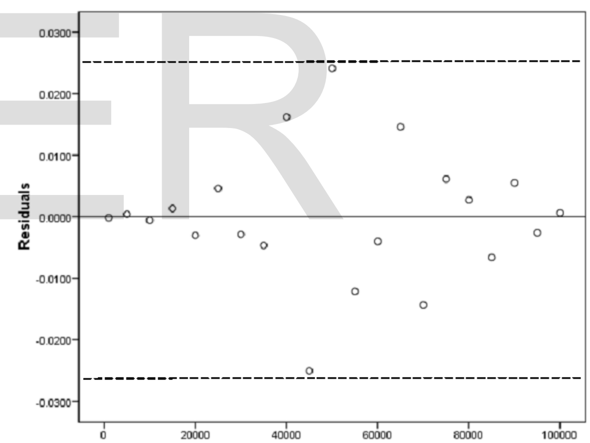

Fig. 2. Residual plot of Fourier7 model.

We can find a horizontal band pattern from the above figure (Fig. 2.) which suggests that the variance of the residuals is constant and the regression is a good one.

c. Residual lag plot –

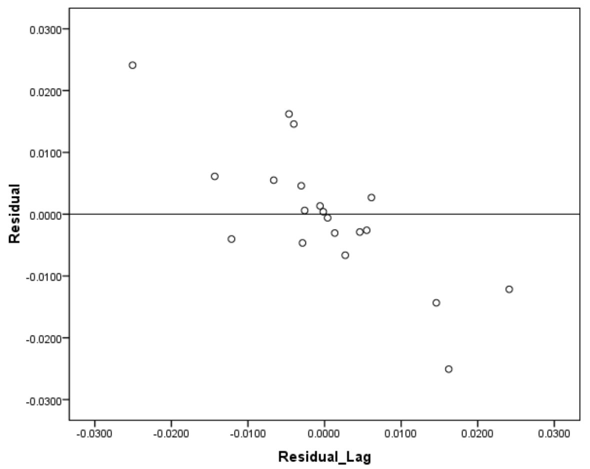

Fig. 3. Residual lag plot of Fourier7 model.

The random pattern in the lag plot (Fig. 3.) suggests that the variance is random and it suggests that the errors are independent.

d. Histogram of the residuals –

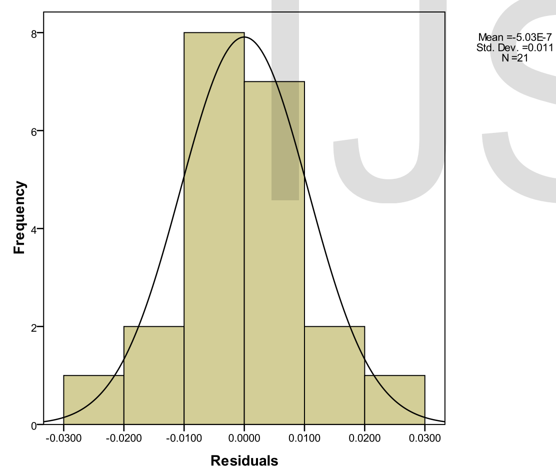

Fig. 4. Histogram of the residuals of Fourier7 model.

The above figure (Fig. 4.) exhibits a symmetric bell shaped histogram which is evenly distributed around zero. Therefore it suggests that the residuals are normally distributed.

e. Q-Q plot of the residuals –

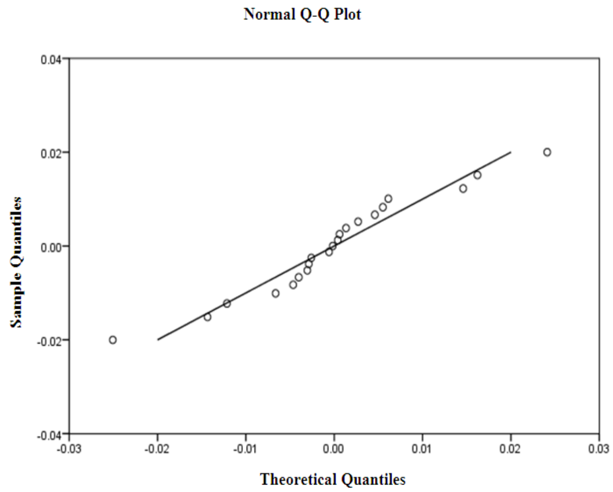

Fig. 5. Q-Q plot of the residuals of Fourier7 model.

It may be easily observed from the figure (Fig. 5.) that the residuals are approximately linear. Therefore it suggests that the residuals follow approximately normal distribution.

5.3 Result of the Diagnostic Procedure and the Proposed Model

From the subsection 5.2 the residuals appear to be randomly distributed around zero. Hence, it can be suggested that the model appear to fit the data well and it can be concluded that 'Fourier7' model is the best fit in this case.

The proposed mathematical model is given below:

$$
\begin{aligned}
f(x) = {} & 5.154 \\
& - 1.964 * \cos(x * w) - 5.382 * \sin(x * w) \\
& - 2.283 * \cos(2 * x * w) - 0.6931 * \sin(2 * x * w) \\
& - 1.089 * \cos(3 * x * w) + 0.6535 * \sin(3 * x * w) \\
& - 0.1839 * \cos(4 * x * w) + 0.6587 * \sin(4 * x * w) \\
& + 0.1954 * \cos(5 * x * w) + 0.3057 * \sin(5 * x * w) \\
& + 0.1457 * \cos(6 * x * w) + 0.04046 * \sin(6 * x * w) \\
& + 0.04612 * \cos(7 * x * w) - 0.04671 * \sin(7 * x * w)
\end{aligned}
$$

Here, the value of 'w' is equal to 0.0000486. The plot of the proposed model is shown in Fig. 6. The circles indicate data points and the solid line represents the curve of the proposed model.

ISSN 2229-5518

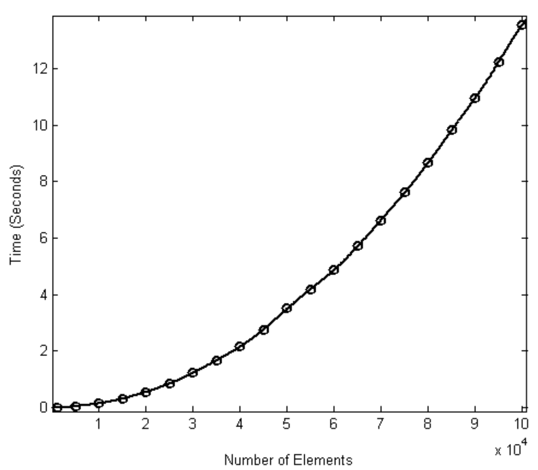

Fig. 6. Proposed model (Fourier7).

6 Limitations And Future Scope

We ran classical Quicksort algorithm on a particular machine configuration *i.e.* hardware and a state of the operating system. We are yet to explore the effects of different hardwares and operating system configurations and its states on the performance of Quicksort in worst case.

We have also not considered factors pertained to operating system (Fedora 14 in this case) i.e. context switch time, cache hit, cache miss, buffering etc.

Various pivoting schemes also are not explored in this paper.

It will certainly be our future endeavor to consider these factors as we believe that these factors can contribute to actual execution time of Quicksort.

This article is a simplistic representation of the said algorithm, hence we assumed the net iteration time to be our execution time, but in reality the total execution time of the algorithm is higher though in negligible proportion.

7 Conclusion

In this study we have used curve fitting techniques to examine the experimentally simulated dataset. After plotting the dataset given in TABLE 1 (Number of elements in X axis and Time in Y axis), it is evident that the dataset has a definite pattern and takes a shape of a curve i.e. it is following curvilinearity. The data have been analyzed by using different type of fits namely Polynomial, Exponential, Power, Gaussian and Fourier. The analysis shows that the Fourier type of fit outperformed all the other types of fit in terms of goodness of fit statistics. In this case, '***Fourier7***' is the best fit with highest R^2 (0.999994), highest Adjusted R^2 (0.999976), lowest SSE (0.0022) and lowest RMSE (0.0212). Graphical residual analysis of ***Fourier7*** model exhibits that the model fits the data well, variance of the residuals is constant, errors are independent, residuals are normally distributed and approximately linear. This indicates that the performance of Quicksort in worst case follows Fourier type of fit. The study shows that among the various available models, the Fourier type of models may be explored in depth while analyzing the performance of Quicksort in worst case.

References

[1] R. Fleischer, B. Moret and E.M. Schmidt, eds., *Experimental Algorithmics From Algorithm Design to Robust and Efficient Software*, Berlin, Heidelberg: Springer-Verlag, pp. V, 2002.

[2] T. Niemann, "Sorting and Searching Algorithms," http://epaperpress.com/sortsearch/download/sortsearch.pdf. (Accessed November, 2013).

[3] I. Parberry, "Analysis of Quicksort", Technical Report, Department of Computer Science, University of North Texas, 1997.

[4] M. Fouz, M. Kufleitner, B. Manthey, N.Z. Jahromi, "On Smoothed Analysis of Quicksort and Hoare's Find", *Algorithmica*, Vol. 62, Issue3-4, pp 879-905, April 2012.

[5] R. Sedgewick, "Implementing Quicksort Programs", *Communications of the ACM*, Vol. 21, Issue 10, October 1978.

[6] U. Rosler, "A Limit Theorem for Quicksort", *Theoretical Informatics and Applications*, Vol. 25, Issue 1, pp 85-100, 1991.

[7] C.J.H. McDiarmid, R.B. Hayward, "Large Deviations for Quicksort", *Journal of Algorithms*, Vol. 21, Issue 3, pp 476-507, November 1996.

[8] J.A.Fill, S. Janson, "Approximating The Limiting Quicksort Distribution", *Random Structures and Algorithms*, Vol. 19, Issue 3-4, pp 376-406, December 2001.

[9] M. Fuchs, "A Note on the Quicksort Asymptotics", *Random Structures and Algorithms*, November 2013, doi: 10.1002/rsa.20524.

[10] R. Neininger, "Refined Quicksort Asymptotics", *Random Structures and Algorithms*, April 2013, doi: 10.1002/rsa.20497.

[11] J.A. Fill, "Distributional Convergence for the Number of Symbol Comparisons Used by Quicksort", *The Annals of Applied Probability*, Vol. 23, No. 3, pp 859-1289, 2013.

[12] S. Wild, M.E. Nebel, "Average Case Analysis of Java 7's Dual Pivot Quicksort", *Proceeding of the 20th Annual European Conference on Algorithms*, pp 825-836, 2012.

[13] J.A. Fill, S. Janson, "Smoothness and Decay Properties of the Limiting Quicksort Density Function", *Trends in Mathematics*, pp 53-64, 2000.

[14] G.S. Brodal, R. Fagerberg, G. Moruz, "On the Adaptiveness of Quicksort", *ACM Journal of Experimental Algorithmics*, Vol. 12, Article No. 3.2, 2008.

[15] L. Khreisat, "Quicksort: A Historical Perspective and Empirical Study", *IJCNS*, Vol. 12, No. 12, December 2007.

[16] S.F. Solehria, S. Jadoon, "Multiple Pivot Sort Algorithm is Faster than Quicksort Algorithm: An Empirical Study", *International Journal of Electrical and Computer Sciences*, Vol. 11, No. 3, June 2011.

[17] H. Erkio, "The Worst Case Permutation for Median-of-Three Quicksort", *The Computer Journal*, Vol. 27, No. 3, 1984.

[18] D. Cederman, P. Tsigas, "GPU-Quicksort: A Practical Quicksort Algorithm for Graphics Processors", *ACM Journal of Experimental Algorithmics*, Vol. 14, Article No. 4, 2009.

[19] P. Sanders, S. Peter, T. Hansch, "On the Efficient Implementation of Massively Parallel Quicksort", *Solving Irregularly Structured Problems in Parallel*, pp13-24, 1997.

[20] M. Ragab, U. Roeslar, "The Quicksort Process", *Stochastic Processes and Their Applications*, Vol 124, Issue 2, pp 1036-1054, February 2014.

[21] K.C. Kiwiel, "Partitioning Schemes for Quicksort and Quickselect", *The Computing Research Repository*, Vol. 312054, 2003.

[22] A. Mohammed, M. Othman, “A New Pivot Selection Scheme for Quicksort Algorithm”, *Journal of Science and Technology*, Vol. 11, pp 211-215, 2004.

[23] S. Kushagra, A.L. Ortiz, A. Qiao, J. Ian Munro, “Multi-Pivot Quicksort: Theory and Experiments”, *Proceedings of the 16th Workshop on Algorithm Engineering and Experiments* (ALENEX), pp 47-60, 2014.

[24] S.H. Shiau, C.B. Yang, “The Generalization of Quicksort”, *Proceedings of Workshop on Internet and Distributed Systems* (WIND), 2000.

[25] R.L. Wainright, “Quicksort Algorithms with an Early Exit for Sorted Subfiles”, *Proceedings of the 15th Annual Conference on Computer Science*, pp 183-190, 1987.

[26] J.L. Bentley, M.D. Mcllroy, “Engineering a Sort Function”, *Software: Practice and Experience*, Vol. 23 (11), pp 249-265, November 1993.

[27] *Evaluating Goodness of Fit – MATLAB & Simulink – MathWorks India*, MathWorks, [online], http://www.mathworks.in/help/curvefit/evaluating-goodness-of-fit.html (Accessed: November, 2013).

[28] *Graphic Residual Analysis*, OriginLab, [online], http://www.originlab.com/www/helponline/origin/en/UserGuide/Graphic_Residual_Analysis.html (Accessed: November, 2013).

[29] *5.2.4. Are the model residuals well-behaved?*, Engineering Statistics Handbook, [online], http://www.itl.nist.gov/div898/handbook/pri/section2/pri24.htm (Accessed: November, 2013).

[30] *4.4.4.1. How can I assess the sufficiency of the functional part of the model?*, Engineering Statistics Handbook, [online], http://www.itl.nist.gov/div898/handbook/pmd/section4/pmd441.htm (Accessed: November, 2013).

[31] *4.4.4.4. How can I assess whether the random errors are independent from one to the next?,* Engineering Statistics Handbook, [online], http://www.itl.nist.gov/div898/handbook/pmd/section4/pmd444.htm (Accessed: November, 2013).

[32] *Normal Probability Plot of Residuals | STAT 501 – Regression Methods*, PENNSTATE, [online], https://onlinecourses.science.psu.edu/stat501/node/40 (Accessed: November, 2013).

[33] G. Bandyopadhyay, “Modeling NPA time series data in selected public sector banks in India with semi parametric approach,” *International Journal of Scientific & Engineering Research*, Vol. 4, Issue 12, pp 1876 – 1889, December 2013.

[34] C.A.R. Hoare, "Quicksort", *The Computer Journal*, vol 5, no.1, pp 10-16, 1962.

[35] *Nonlinear Least Squares (Curve Fitting) – MATLAB & Simulink – MathWorks India*, MathWorks, [online], http://www.mathworks.in/help/optim/nonlinear-least-squares-curve-fitting.html (Accessed: November, 2013).

[36] *fitoptions - MATLAB & Simulink – MathWorks India*, MathWorks, [online], http://www.mathworks.in/help/curvefit/fitoptions.html (Accessed: November, 2013).

[37] S.D. Sewell, “Overview of Matlab Curve Fitting Toolbox”, 2002. Avaiable from: http://dspace.mit.edu/bitstream/handle/1721.1/36390/8-13Fall-2002/NR/rdonlyres/Physics/8-13Experimental-Physics-I---II--Junior-Lab-Fall2002/031FB55D-EEFC-4B2C-9049-B7C4FD8BA3E0/0/matlabcurvefittingtoolbox.pdf (Accessed: December, 2013).