Neural, Parallel, and Scientific Computations, **27**, No. 2 (2019), 115-129 ISSN: 1061-5369

A NOTE ON THE SONG–CHANG–PHAM'S SOFTWARE RELIABILITY MODEL. SOME APPLICATIONS. I

NIKOLAY PAVLOV[1], GEORGI SPASOV[2],
ANTON ILIEV[3], ASEN RAHNEV[4], AND NIKOLAY KYURKCHIEV[5]
[1,2,3,4,5]Faculty of Mathematics and Informatics
University of Plovdiv Paisii Hilendarski
24, Tzar Asen Str., 4000 Plovdiv, BULGARIA

ABSTRACT: The Hausdorff approximation of the shifted Heaviside function $h_{t_0}(t)$ by cumulative function based on the Song–Chang–Pham's model [1] (see, also [2]) is investigated and an expression for the error of the best approximation is obtained in this paper.

The results of numerical examples confirm theoretical conclusions and they are obtained using programming environment Mathematica.

We give real examples with dataset, was proposed by Musa in [3] using the new Song–Chang–Pham's software reliability model.

AMS Subject Classification: 41A46
Key Words: four parameters Song–Chang–Pham's cumulative function (4SHPcdf), Heaviside step–function $h_{t_0}(t)$, Hausdorff distance, upper and lower bounds

Received: May 27, 2019; **Accepted:** August 18, 2019;
Published: August 31, 2019. **doi:** 10.12732/npsc.v27i2.6
Dynamic Publishers, Inc., Acad. Publishers, Ltd. https://acadsol.eu/npsc

1. INTRODUCTION AND PRELIMINARIES

An important role within the hierarchical models in the procedure for quantifying the quality of software products is played by the so-called computational method based on the theoretical and empirical dependencies (usually at an early stage in their development), statistical data accumulated during tests, exploitation and the accompaniment of the program product.

Detailed description of all elements in the area of debugging theory may be found in the following books [4]–[6].

In the books [7]–[8], we pay particular attention to both deterministic approaches and probability models for debugging theories. A Hausdorff metric was chosen to evaluate the test data which are fitted to the sigmoid models proposed in these book. Some of the existing cumulative distributions (Gompertz–Makeham, Yamada-exponential, Yamada–Rayleigh, Yamada–Weibull, transmuted inverse exponential, transmuted Log-Logistic, Kumaraswamy–Dagum and Kumaraswamy Quasi Lindley) are considered in the light of modern debugging and test theories.

Some software reliability models, can be found in [9]–[34].

In this note we study the Hausdorff approximation of the Heaviside function $h_{t_0}(t)$ by function based on the Song–Chang–Pham's [1] cumulative function.

We propose a software modules (intellectual properties) within the programming environment CAS Mathematica for the analysis.

The models have been tested with real-world data.

A general mean value function $m(t)$ of software reliability models is given by solution of the equation

$$\frac{d\,m(t)}{dt} = \eta h(t)(a(t) - m(t)),$$

where

- $m(t)$ is the mean value function of faults detected up to time t;

- $a(t)$ is the total number of faults in the Software at time t;

- $h(t)$ represents the fault detection rate function dependent on time t

- (assume η's probability density function is $g(\eta)$)

and $m(t_0) = m_0$ is the marginal condition of

$$m(t) = \int_0^{+\infty} e^{-xH(t)} \left(m_0 + \int_{t_0}^{t} x a(\tau) h(\tau) e^{xH(\tau)} d\tau \right) g(x) dx,$$

$$H(t) = \int_0^t h(u) du$$

For example, for

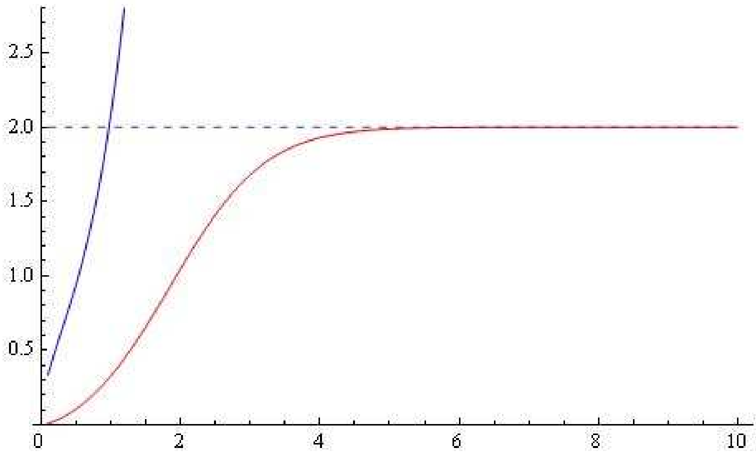

Figure 1: The functions $m(t)$–red; $h(t)$–blue; $a(t) = a$–dashed.

$$\begin{aligned} a(t) &= a, \\ h(t) &= b \ln a \times t^{b-1} a^{t^b} \\ g(x) &= \tfrac{\beta^\alpha x^{\alpha-1} e^{\beta x}}{\Gamma(\alpha)} \end{aligned}$$

we have the Pham's software reliability model with Vtub–Shaped fault detection rate and the uncertainty of operating environment:

$$m(t) = a\left(1 - \left(\frac{\beta}{\beta - 1 + a^{t^b}}\right)^\alpha\right)$$

(see, Fig. 1).

For some details, see [1]).

For a special choice of the functions $h(t)$ and $a(t)$, in [1] the authors considered a class of growth functions.

First, we study the Hausdorff approximation of the shifted Heaviside function $h_{t_0}(t)$ by sigmoidal function based on the Song, Chang and Pham [1] cumulative function (1).

Definition 1. Song, Chang and Pham [1] developed the following software reliability growth model:

$$M(t) = N\left(1 - \left(\frac{\beta}{\beta + \ln\frac{a+e^{bt}}{a+1}}\right)^{\alpha}\right). \tag{1}$$

where $a, b, \alpha, \beta > 0,\ t > 0$.

Definition 2. The shifted Heaviside step function is defined by

$$h_{t_0}(t) = \begin{cases} 0, & \text{if} \quad t < t_0, \\ [0,1], & \text{if} \quad t = t_0, \\ 1, & \text{if} \quad t > t_0 \end{cases} \tag{2}$$

Definition 3. [35] The Hausdorff distance (the H–distance) $\rho(f,g)$ between two interval functions f, g on $\Omega \subseteq \mathbb{R}$, is the distance between their completed graphs $F(f)$ and $F(g)$ considered as closed subsets of $\Omega \times \mathbb{R}$. More precisely,

$$\rho(f,g) = \max\{ \sup_{A \in F(f)} \inf_{B \in F(g)} ||A - B||, \sup_{B \in F(g)} \inf_{A \in F(f)} ||A - B||\},$$

wherein $||.||$ is any norm in $\mathbb{R}^2$, e. g. the maximum norm $||(t,x)|| = \max\{|t|,|x|\}$; hence the distance between the points $A = (t_A, x_A)$, $B = (t_B, x_B)$ in $\mathbb{R}^2$ is $||A - B|| = \max(|t_A - t_B|, |x_A - x_B|)$.

2. MAIN RESULTS

2.1. A NOTE ON THE SOFTWARE RELIABILITY GROWTH MODEL (1)

Without loosing of generality we will look at the following "cumulative sigmoid":

$$M^*(t) = 1 - \left(\frac{\beta}{\beta + \ln\frac{a+e^{bt}}{a+1}}\right)^{\alpha}, \tag{3}$$

with $N = 1$ (see (1)), and

$$t_0 = \frac{1}{b} \ln \left((1+a) e^{\frac{\beta \left(1-(1/2)^{\frac{1}{\alpha}}\right)}{(1/2)^{\frac{1}{\alpha}}}} - a \right). \tag{4}$$

Evidently, for the "median" we have

$$M^*(t_0) = \frac{1}{2}.$$

The one–sided Hausdorff distance d between the function $h_{t_0}(t)$ and the cumulative function (3) satisfies the relation

$$M^*(t_0 + d) = 1 - d. \tag{5}$$

2.2. APPROXIMATION RESULT

With some constraints imposed on the parameters a, b, α and β (which we will not stop here, for some details, see [1]), it can be shown that the following theorem gives upper and lower bounds for d

Theorem. Let

$$p = -\frac{1}{2},$$

$$q = 1 + \frac{b\alpha(1/2)^{\frac{1}{\alpha}} e^{bt_0}}{2\beta(a + e^{bt_0})}.$$

For the one–sided Hausdorff distance d between $h_{t_0}(t)$ and the function (3) the following inequalities hold for:

$$2.1q > e^{1.05}$$

$$d_l = \frac{1}{2.1q} < d < \frac{\ln(2.1q)}{2.1q} = d_r. \tag{6}$$

Proof. Let us examine the function:

$$F(d) = M^*(t_0 + d) - 1 + d. \tag{7}$$

From $F'(d) > 0$ we conclude that function F is increasing.

Consider the function

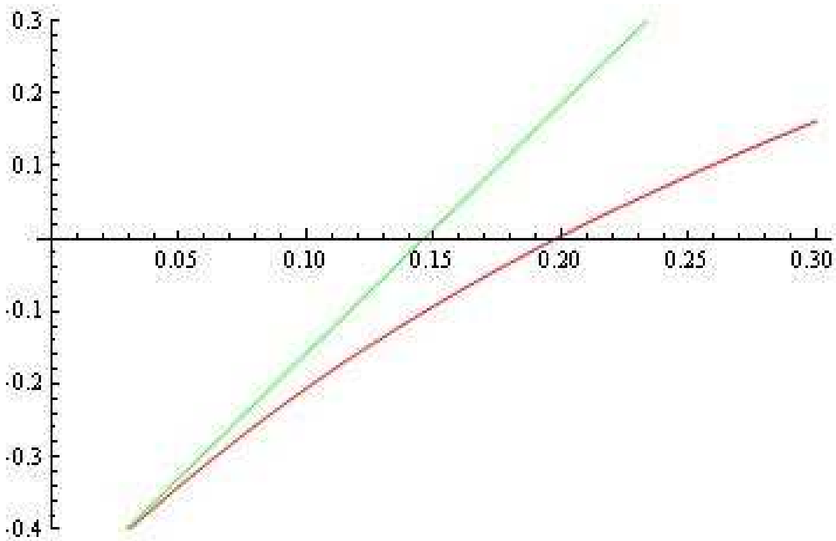

Figure 2: The functions $F(d)$ and $G(d)$.

$$G(d) = p + qd. \tag{8}$$

From Taylor expansion we obtain $G(d) - F(d) = O(d^2)$.

Hence $G(d)$ approximates $F(d)$ with $d \to 0$ as $O(d^2)$ (see Fig. 2).

In addition $G'(d) > 0$.

Further, for $2.1q > e^{1.05}$ we have $G(d_l) < 0$ and $G(d_r) > 0$.

This completes the proof of the theorem.

The model (3) for $\beta = 0.2$, $\alpha = 0.8$, $a = 16$, $b = 10.1$, $t_0 = 0.183735$ is visualized on Fig. 3.

From the nonlinear equation (5) and inequalities (6.6) we have: $d = 0.198019$, $d_l = 0.13901$ and $d_r = 0.274296$.

The model (3) for $\beta = 0.1$, $\alpha = 0.9$, $a = 20$, $b = 15.1$, $t_0 = 0.0850837$, $d = 0.136023$, $d_l = 0.0828472$, $d_r = 0.206352$ is visualized on Fig. 4.

The model (3) for $\beta = 0.05$, $\alpha = 0.99$, $a = 25$, $b = 20$ is visualized on Fig. 5.

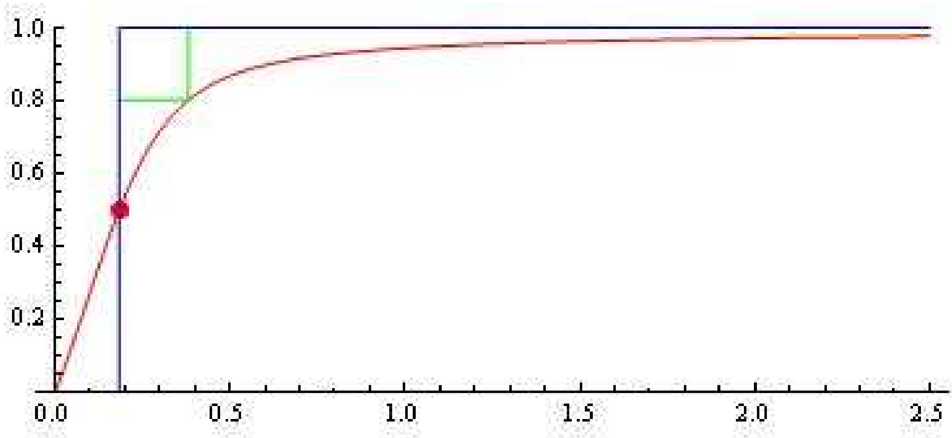

Figure 3: The model (3) for $\beta = 0.2$, $\alpha = 0.8$, $a = 16$, $b = 10.1$, $t_0 = 0.183735$; H–distance $d = 0.198019$, $d_l = 0.13901$ and $d_r = 0.274296$.

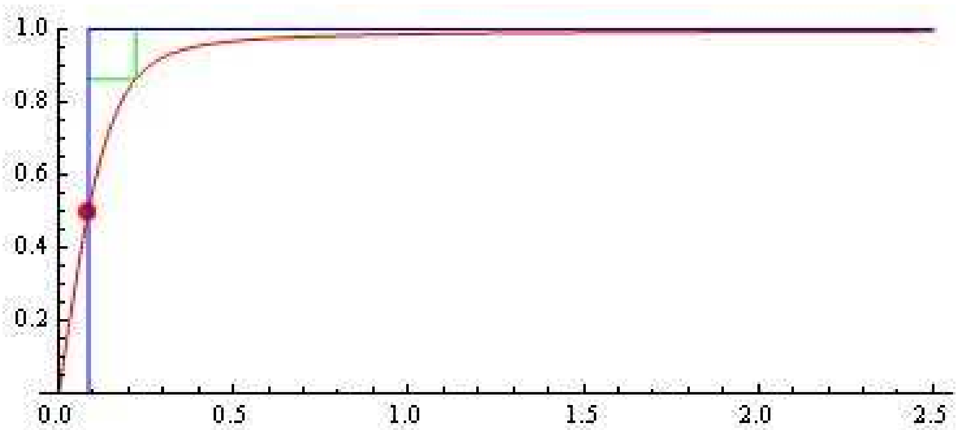

Figure 4: The model (3) for $\beta = 0.1$, $\alpha = 0.9$, $a = 20$, $b = 15.1$, $t_0 = 0.0850837$; H–distance $d = 0.136023$, $d_l = 0.0828472$, $d_r = 0.206352$.

3. NUMERICAL EXAMPLES.

Example 1. We examine the dataset, was proposed by Musa in [3].

For the first 12 hours of testing, the number of failures each hour is given in Fig. 6

Below, we will illustrate the fitting of this data, for example, with the $M_1(t)$ model.

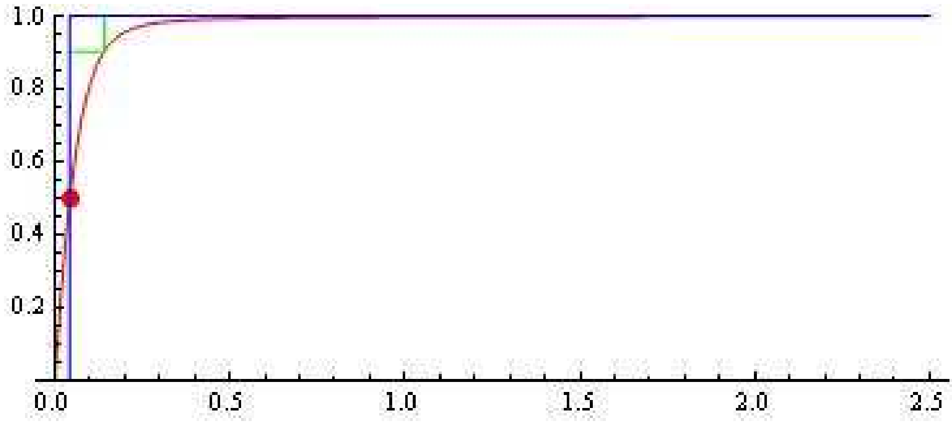

Figure 5: The model (3) for $\beta = 0.05$, $\alpha = 0.99$, $a = 25$, $b = 20$, $t_0 = 0.0427688$; H–distance $d = 0.0979944$, $d_l = 0.0503667$, $d_r = 0.150517$.

Hour	*Number of Failures*	*Cumulative failures*
1	27	27
2	16	43
3	11	54
4	10	64
5	11	75
6	7	82
7	2	84
8	5	89
9	3	92
10	1	93
11	4	97
12	7	104

Figure 6: Dataset [3].

The fitted model

$$M_1^*(t) = N\left(1 - \left(\frac{\beta}{\beta + \ln\frac{a+e^{bt}}{a+1}}\right)^{\alpha}\right)$$

based on the dataset for the estimated parameters:

$$N = 104; \quad a = 2.65; \quad \beta = 0.73006; \quad \alpha = 1.7; \quad b = 0.35$$

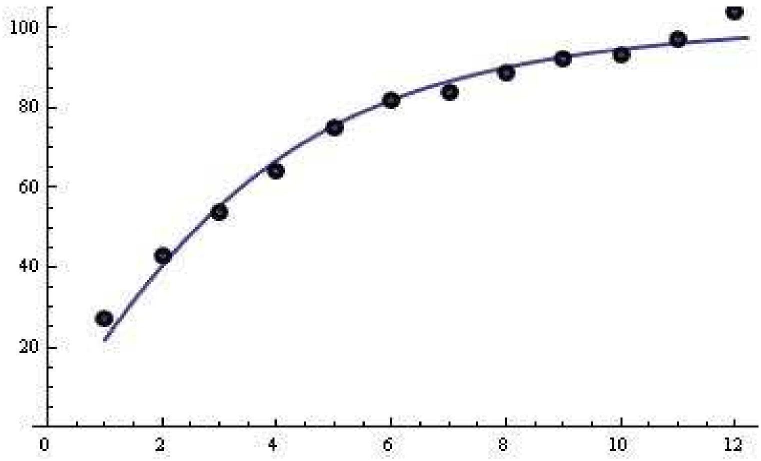

Figure 7: The model $M_1^*(t)$ with $N = 104$; $a = 2.65$; $\beta = 0.73006$; $\alpha = 1.7$; $b = 0.35$.

Age	*Length(mm)*
1	16.1
2	33.9
3	54.3
4	76.2
5	97.8
6	117.1
7	133.3
8	146.5
9	157.2
10	166
11	173.3
12	179.6
13	185
14	189.9
15	194

Figure 8: The extended data for modeling the growth of red abalone Haliotis Rufescens in Northern California.

is plotted on Fig. 7.

Example 2. We examine the following data for the growth of red abalone Haliotis Rufescens in Northern California.

The extended data for modeling the growth of red abalone is shown on Fig. 8.

For more details see [37].

For this data the fitted model for estimated parameters:

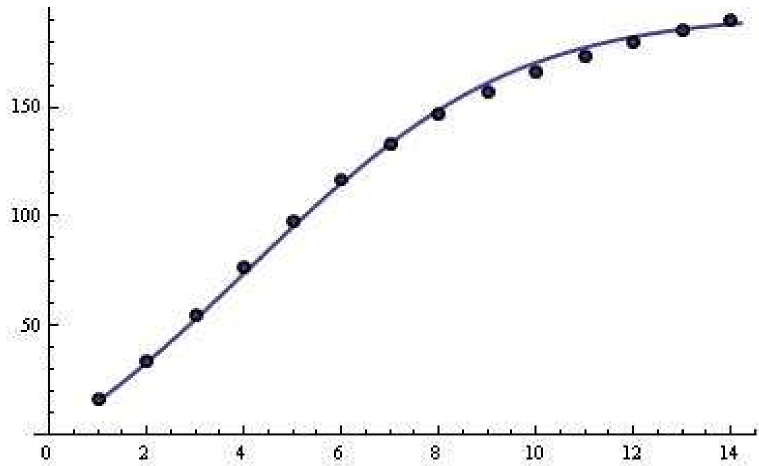

Figure 9: The fitted model $M_1^*(t)$ with $N = 194; a = 18.65; \beta = 0.589515; \alpha = 2.9;\ b = 0.29$.

$$N = 194;\ a = 18.65;\ \beta = 0.589515;\ \alpha = 2.9;\ b = 0.29$$

is plotted on Fig. 9.

Example 3. Analysis of MyDoom worm propagation.

MyDoom spreads by email [38], [39].

MyDoom searches local hard drive for addresses. MyDoom.O uses Web search engines.

On July 26, 2004 (see Fig. 10) a variant of Mydoom attacks Google, AltaVista and Lycos, completely stopping the function of the popular Google search engine for the larger portion of the workday, and creating noticeable slow-downs in the AltaVista and Lycos engines for hours.

Queries are distributed between Google (45%), Lycos (22.5%), Yahoo (20%) and Altavista (12.5%).

As usually we photographed the data from Fig. 10.

$$\mathit{data\_MyDoom} := \{\{0,0\},\{1,800\},\{2,3000\},\{3,9610\},\{4,23270\},$$
$$\{5,38846\},\{6,50000\},\{7,53846\},\{8,57300\}\}$$

Approximation is obtained by us using these data and the model (3), see Fig. 11.

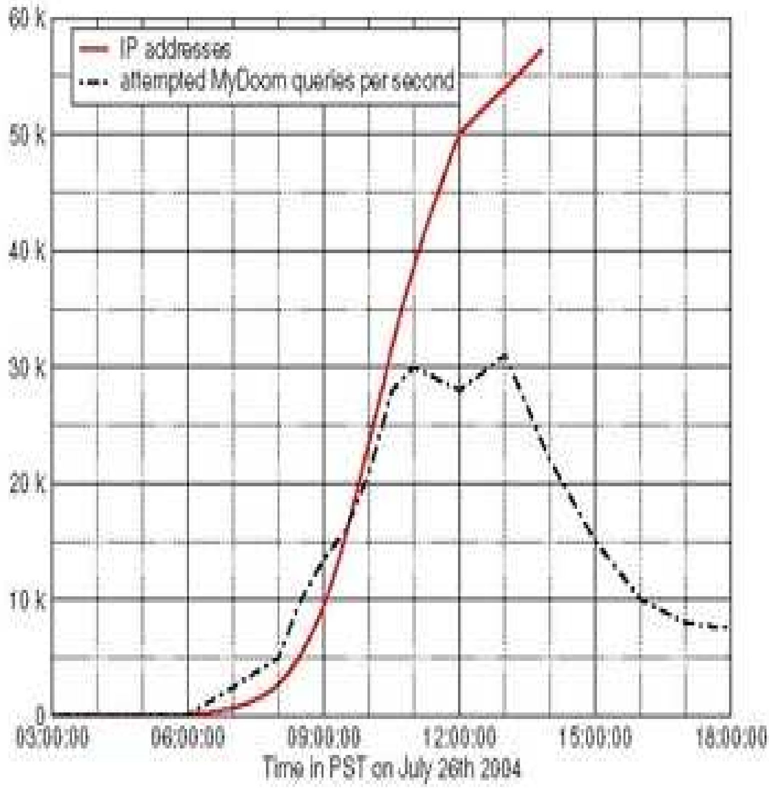

Figure 10: Google's view of MyDoom [38].

4. CONCLUDING REMARKS.

The Song–Chang–Pham's model can be applied to the debugging theory (Example 1), population dynamics (Example 2) and computer viruses propagation (Example 3).

In conclusion, we will note that the determination of compulsory in area of the Software Reliability Theory components, such as confidence intervals and confidence bounds, should also be accompanied by a serious analysis of the value of the best Hausdorff approximation - the subject of study in the present paper.

We hope that the results will be useful for specialists in this scientific area.

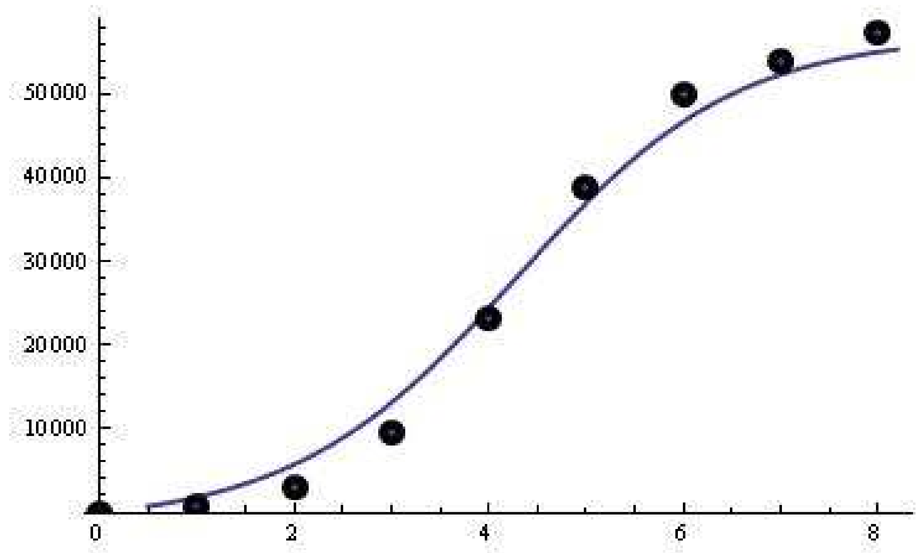

Figure 11: The model $M_1^*(t)$ with $N = 57300$; $\alpha = 1.102$; $a = 2160$; $b = 0.8585$; $\beta = 0.021036$.

ACKNOWLEDGMENTS

This paper is supported by the Project FP19-FMI-002 "Innovative ICT for Digital Research Area in Mathematics, Informatics and Pedagogy of Education" of the Scientific Fund of the University of Plovdiv Paisii Hilendarski, Bulgaria.

REFERENCES

[1] K. Song, I. Chang, H. Pham, NHPP software reliability model with inflection factor of the fault detection rate considering the uncertainty of software operating environments and predictive analysis, *Symmetry*, **11** (2019), 521.

[2] Q. Li, H. Pham, A generalized software reliability growth model with consideration of the uncertainty of operating environment, *IEEE Access*, **XX**, (2017).

[3] J. D. Musa, A. Ianino, K. Okumoto, *Software Reliability: Measurement*, Prediction, Applications, McGraw–Hill (1987).

[4] S. Yamada, *Software Reliability Modeling: Fundamentals and Applications*, Springer (2014).

[5] S. Yamada, Y. Tamura, *OSS Reliability Measurement and Assessment*, In: Springer

Series in Reliability Engineering (H. Pham, Ed.), Springer International Publishing Switzerland (2016).

[6] H. Pham, *System Software Reliability*, In: Springer Series in Reliability Engineering, Springer–Verlag London Limited (2006).

[7] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, *Some software reliability models: Approximation and modeling aspects*, LAP LAMBERT Academic Publishing (2018), ISBN: 978-613-9-82805-0.

[8] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, *Nontrivial Models in Debugging Theory (Part 2)*, LAP LAMBERT Academic Publishing (2018), ISBN: 978-613-9-87794-2.

[9] K. Ohishi, H. Okamura, T. Dohi, Gompertz software reliability model: Estimation algorithm and empirical validation, *J. of Systems and Software*, **82**, No. 3 (2009), 535–543.

[10] D. Satoh, S. Yamada, Discrete equations and software reliability growth models, in: *Proc. 12th Int. Symp. on Software Reliab. and Eng.*, (2001), 176–184.

[11] S. Yamada, A stochastic software reliability growth model with Gompertz curve, *Trans. IPSJ*, **33** (1992), 964–969. (in Japanese)

[12] E. P. Virene, Reliability growth and its upper limit, in: Proc. 1968, *Annual Symp. on Realib.*, (1968), 265–270.

[13] S. Rafi, S. Akthar, Software Reliability Growth Model with Gompertz TEF and Optimal Release Time Determination by Improving the Test Efficiency, *Int. J. of Comput. Applications*, **7**, No. 11 (2010), 34–43.

[14] S. Yamada, M. Ohba, S. Osaki, S–shaped reliability growth modeling for software error detection, *IEEE Trans. Reliab.*, **R–32** (1983), 475–478.

[15] S. Yamada, S. Osaki, Software reliability growth modeling: Models and Applications, IEEE Transaction on Software Engineering, **vol. SE–11**, (1985), 1431–1437.

[16] N. Pavlov, G. Spasov, A. Rahnev, N. Kyurkchiev, Some deterministic reliability growth curves for software error detection: Approximation and modeling aspects, *International Journal of Pure and Applied Mathematics*, **118**, No. 3 (2018), 599–611.

[17] N. Pavlov, A. Golev, A. Rahnev, N. Kyurkchiev, A note on the Yamada–exponential software reliability model, *International Journal of Pure and Applied Mathematics*, **118**, No. 4 (2018), 871–882.

[18] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, A Note on The "Mean Value" Software Reliability Model, *International Journal of Pure and Applied Mathematics*, **118**, No. 4 (2018), 949–956.

[19] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, Analysis of the Chen's and Pham's Software Reliability Models, *Cybernetics and Information Technologies*, **18**, No. 3 (2018), 37–47.

[20] N. Pavlov, A. Golev, A. Rahnev, N. Kyurkchiev, A note on the generalized inverted exponential software reliability model, *International Journal of Advanced Research in Computer and Communication Engineering*, **7**, No. 3 (2018), 484–487.

[21] A. L. Goel, Software reliability models: Assumptions, limitations and applicability, *IEEE Trans. Software Eng.*, **SE–11** (1985), 1411–1423.

[22] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, Transmuted inverse exponential software reliability model, *Int. J. of Latest Research in Engineering and Technology*, **4**, No. 5 (2018), 1–6.

[23] A. Pandey, N. Goyal, Early Software Reliability Prediction. A Fuzzy Logic Approach, In: Studies in Fuzziness and Soft Computing (J. Kacprzyk, Ed.), **303**, Springer, London (2013).

[24] N. Pavlov, G. Spasov, A. Rahnev, N. Kyurkchiev, A new class of Gompertz–type software reliability models, *International Electronic Journal of Pure and Applied Mathematics*, **12**, No. 1 (2018), 43–57.

[25] O. Rahneva, H. Kiskinov, A. Malinova, G. Spasov, A Note on the Lee-Chang-Pham-Song Software Reliability Model, *Neural, Parallel, and Scientific Computations*, **26**, No. 3 (2018), 297–310.

[26] A. Wood, Predicting software reliability, *IEEE Computer*, **11** (1996), 69–77.

[27] H. Pham, L. Nordmann, X. Zhang, A General Imperfect-Software-Debugging Model with S-Shaped Fault-Detection Rate, *IEEE Trans. Rel.*, **48**, No. 2 (1999), 169–175.

[28] P. K. Kapur, H. Pham, A. Gupta, P. C. Jha, *Software Reliability Assessment with OR Applications*, In: Springer Series in Reliability Engineering, Springer-Verlag, London (2011).

[29] P. Karup, R. Garg, S. Kumar, *Contributions to Hardware and Software Reliability*, World Scientific, London (1999).

[30] M. Lyu (Ed. in Chief), *Handbook of Software Reliability Engineering*, IEEE Computer Society Press, The McGraw-Hill Companies, Los Alamitos (1996).

[31] M. Ohba, Software reliability analysis models, *IBM J. Research and Development*, **21** (1984).

[32] D.R. Jeske, X. Zhang, Some successful approaches to software reliability modeling in industry, *J. Syst. Softw.*, **74** (2005), 85–99.

[33] K. Song, H. Pham, A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments, *Appl. Sci.*, **7** (2017), 983, doi:10.3390/app7100983, 16 pp.

[34] J. D. Musa, *Software Reliability Data*, DACS, RADC, New York (1980).

[35] B. Sendov, *Hausdorff Approximations*, Kluwer, Boston (1990).

[36] N. Kyurkchiev, S. Markov, On the Hausdorff distance between the Heaviside step function and Verhulst logistic function, *J. Math. Chem.*, **54** (2016), 109–119.

[37] L. RogersBennett, D. W. Rogers, S. A. Schultz, Modeling growth and mortality of red abalone Haliotis Rufescens in Northern California, J. of Shellfish Research, **26**, No. 3 (2007), 719-727.

[38] C. Zou, *Worms, School Of Electrical Engineering & Computer Science*, Spring (2012).

[39] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, N. Modadugu, The ghost in the browser: Analysis of web-based malware, *In: Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots)*, (2007).