Communications in Applied Analysis, **24**, No. 1 (2020), 47-59 ISSN: 1083-2564

ON A CUMULATIVE FUNCTION WITH "POLYNOMIAL VARIABLE TRANSFER". SOME APPLICATIONS

NIKOLAY KYURKCHIEV[1], ANTON ILIEV[2], AND ASEN RAHNEV[3]

[1,2,3]Faculty of Mathematics and Informatics
University of Plovdiv Paisii Hilendarski
24, Tzar Asen Str., 4000 Plovdiv, BULGARIA

ABSTRACT: In this paper we consider a new mean value function with "polynomial variable transfer". The use of the new model with many free parameters makes it attractive for analysis and approximation of specific data from Software Reliability Growth Analysis and Debugging and Test Theory.

Some approximation problems related to the "saturation" in Hausdorff sense are given. In Section 3 some applications are also given.

AMS Subject Classification: 41A46
Key Words: mean value function with "polynomial variable transfer", "saturation" in Hausdorff sense

Received: June 20, 2020; **Accepted:** November 7, 2020;
Published: November 14, 2020. **doi:** 10.12732/caa.v24i1.4
Dynamic Publishers, Inc., Acad. Publishers, Ltd. http://www.acadsol.eu/caa

1. INTRODUCTION

In order to assess the reliability, many time–dependent Software Reliability Growth Models (SRGM) [1] have been developed. In [2] the authors propose the mean value function for fault removed during $(0,t)$:

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

where $\alpha>0$, $\beta>0$, $p>0$, $a>0$, $b>0$. The function $m(t)$ is the solution of differential equation $\frac{d\,m(t)}{dt}=b(t)\left(a(t)-pm(t)\right)$ for some initial conditions and plays an important role in reliability assessment of multi-release software system under imperfect fault removal phenomenon. The basic class of the family (1) is the following:

$$M(t)=A\left(1-(1+bt)e^{-bt}\right). \quad (2)$$

When studying the intrinsic properties of these families, in addition to the analysis of the important characteristic "confidential bounds", it is appropriate to study the "saturation" to the horizontal asymptote in the Hausdorff sense [3]. In Section 2 we prove upper and lower estimates for the "saturation" - d in the Hausdorff sense by means of standard function $M(t)$. We also study some new "mean value functions" with "polynomial variable transfer" for fault detection in software systems. In Section 3 some applications are given.

2. MAIN RESULTS

First, in this Section we give estimates for the "saturation" - d in the Hausdorff sense by means of function $M(t)$. The behavior of this function, and more precisely, " saturation to the horizontal asymptote $A=1$, in the Hausdorff sense " is important for professionals working in the field of debugging and test theory. We have

$$M(d)=1-d \quad (3)$$

or, equivalently,

$$H(d):=e^{bd}-b-\frac{1}{d}=0. \quad (4)$$

The following theorem gives upper and lower bounds for d.

Theorem 1. The "saturation"–d satisfies the following inequality

$$d_l:=\frac{1}{b+1}<d. \quad (5)$$

Proof. We consider the interval $[0, +\infty)$.

Clearly, $H'(t) > 0$ and $H(t)$ is increasing function of $t \in [0, +\infty)$.

Hence, if (4) has a root, then it is unique.

Evidently

$$H(d_l) = H\left(\frac{1}{1+b}\right) = e^{\frac{b}{1+b}} - (2b+1) = (2b+1)\left(\frac{e^{\frac{b}{1+b}}}{2b+1} - 1\right).$$

For the derivation of

$$\eta(b) = \frac{e^{\frac{b}{1+b}}}{2b+1} - 1$$

we have

$$\eta'(b) = -\frac{e^{\frac{b}{1+b}}}{(2b+1)^2}(2b^2 + 2b + 1) < 0.$$

Therefore η is a decreasing function of b.

Using $b > 0$ we have

$$H(d_l) = (2b+1)\eta(b) < (2b+1)\eta(0) = 0. \tag{6}$$

This completes the proof of the theorem.

For example, for $b = 36$ from the nonlinear equation (3) we have $d = 0.106007$ and for $b = 66$ we find $d = 0.066586$ (see, Fig. 1).

Following the ideas given in [4]–[5] we consider new nontrivial function with "polynomial variable transfer", i.e. $t \mapsto G(t)$, where $G(t)$ is a polynomial of degree n.

For example, we find the new **"mean value function" with "polynomial variable transfer"**:

$$\begin{aligned} M^*(t) &= A\left(1 - (1 + G(t))e^{-G(t)}\right), \\ G(t) &= \sum_{i=0}^{n} a_i t^i; \;\; a_0 = 0. \end{aligned} \tag{7}$$

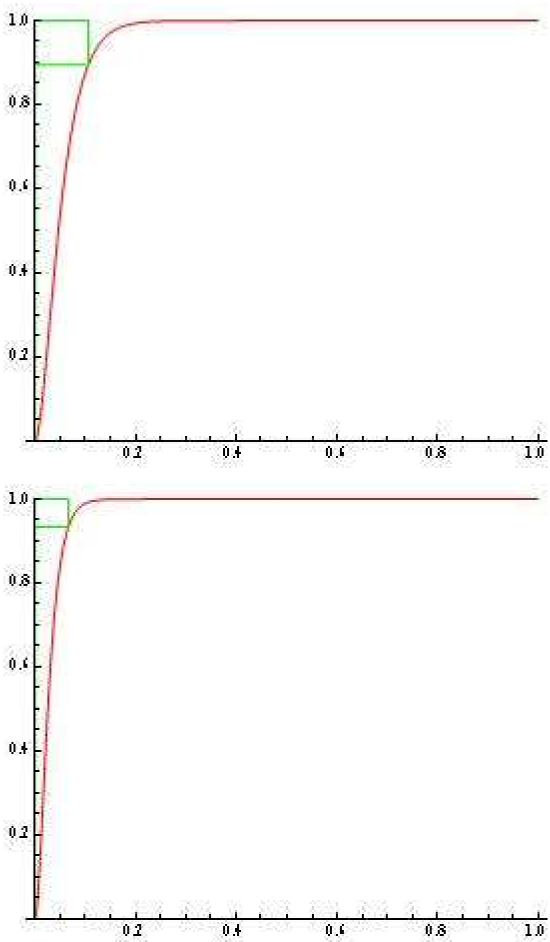

Figure 1: The function $M(t)$ for a) $b = 36$; Saturation in Hausdorff sense $d = 0.106007$; b) $b = 66$; Saturation in Hausdorff sense $d = 0.066586$.

3. SOME APPLICATIONS

Example 1. The following data is used in the modelling process [2] (see, Fig. 2 for Release 4).

For the actual values in the specified period the model - (7) for

$$n = 7,\ A = 43,\ a_0 = 0,\ a_1 = 3.00704,\ a_2 = -1.81421,\ a_3 = -9.59678,$$

$$a_4 = 35.2115,\ a_5 = -41.5765,\ a_6 = 21.0359,\ a_7 = -3.8811$$

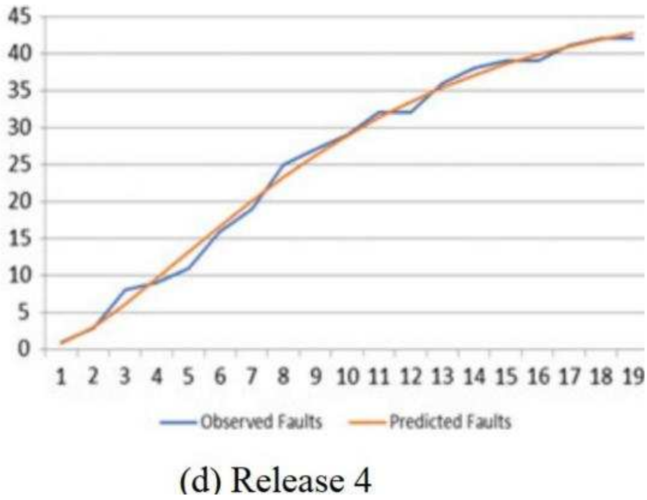

Figure 2: Goodness of fit curve for release 4 [2].

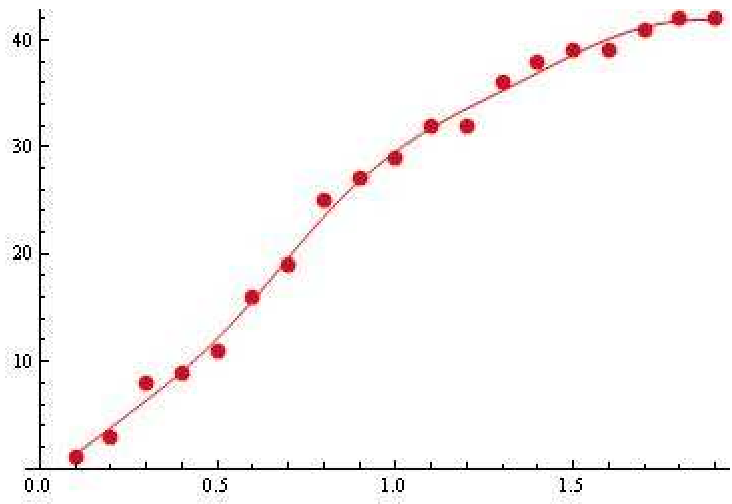

Figure 3: The model (7) ($n = 7$, $A = 43$, $a_0 = 0$, $a_1 = 3.00704$, $a_2 = -1.81421$, $a_3 = -9.59678$, $a_4 = 35.2115$, $a_5 = -41.5765$, $a_6 = 21.0359$, $a_7 = -3.8811$) for the "actual data" (Release 4).

is depicted on Fig. 3.

(We have adopted a scale on the horizontal axis: 0.1 division corresponds to 1 time interval).

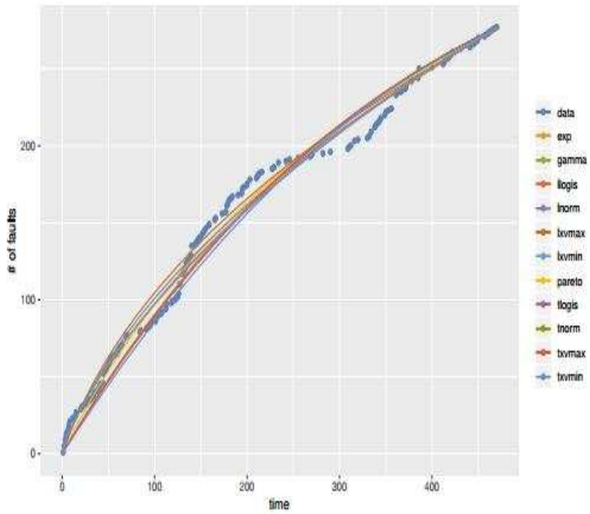

Figure 4: The cumulative number of faults used and the estimated mean value functions (Exponential, Gamma, Log-Logistic, Truncated Gompertz and others) [8].

In [8] Okamura and Dohi revise the Kolmogorov–Smirnov test for Software reliability models (SRM) in the case where the model parameters of (SRM) are estimated from grouped data of the number of detected faults.

Example 2. For the data collected by Musa [7], Fig. 4 shows the cumulative number of faults and the estimated mean value functions for 11 models including Exponential, Gamma, Log-Logistic, Truncated Gompertz and others.

The grouped data consists of the number of faults detected for 427 days, and the number of total faults is 279 [7].

(Here, appropriate scaling has been selected for this dataset).

For the real data [7] the new model (7) for

$$n = 5, \quad A = 289, \;\; a_0 = 0, \quad a_1 = 0.479715, \quad a_2 = -0.0893574,$$

$$a_3 = 0.0110773, \;\; a_4 = -0.000623966, \;\; a_5 = 0.0000127272$$

is depicted on Fig. 5.

Example 3. We examine the following data (see, Table 1). The data were reported by Musa [7] and represent the failures observed during system

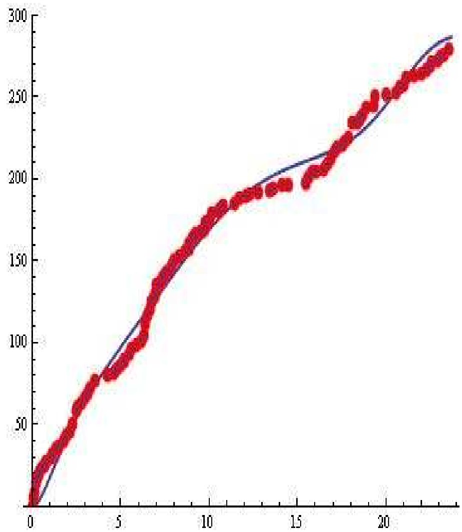

Figure 5: The model (7) ($n = 5$, $A = 289$, $a_0 = 0$, $a_1 = 0.479715$, $a_2 = -0.0893574$, $a_3 = 0.0110773$, $a_4 = -0.000623966$, $a_5 = 0.0000127272$) for the data [7].

testing for 25 hours of CPU time.

Here we will approximate the data from [7] with model (7).

For the actual data in the specified period the model - (7) for

$$n = 5,\ A = 138,\ a_0 = 0,\ a_1 = -7.6955,\ a_2 = 17.948,\ a_3 = -13.0274,$$

$$a_4 = 4.04697,\ a_5 = -0.45042$$

is depicted on Fig. 6.

(We have adopted a scale on the horizontal axis: 0.1 division corresponds to 1 time interval).

4. CONCLUSION

From the analysis of the conducted experiments it is clear that the proposed new model behaves extremely adequately in approximating data in the field of Software Reliable Analysis.

$Hour$	$Number\,of\,failures$	$Cumulative\,failures$
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
13	2	106
14	5	111
15	5	116
16	6	122
17	0	122
18	5	127
19	1	128
20	1	129
21	2	131
22	1	132
23	2	134
24	1	135
25	1	136

Table 1: Failures in 1 Hour (execution time) intervals and cumulative failures [7]

The models can be successfully expanded in the light of our considerations - by inserting corrective corrections of the type ”polynomial variable transfer” and here we will skip their analysis.

Appropriate parameters - a_i; $i = 1, 2, \ldots,$ for the various models of type (8) have been obtained using an optimization method for minimizing the sum of the squared differences between the data ”actual values” and the computed theoretical solution.

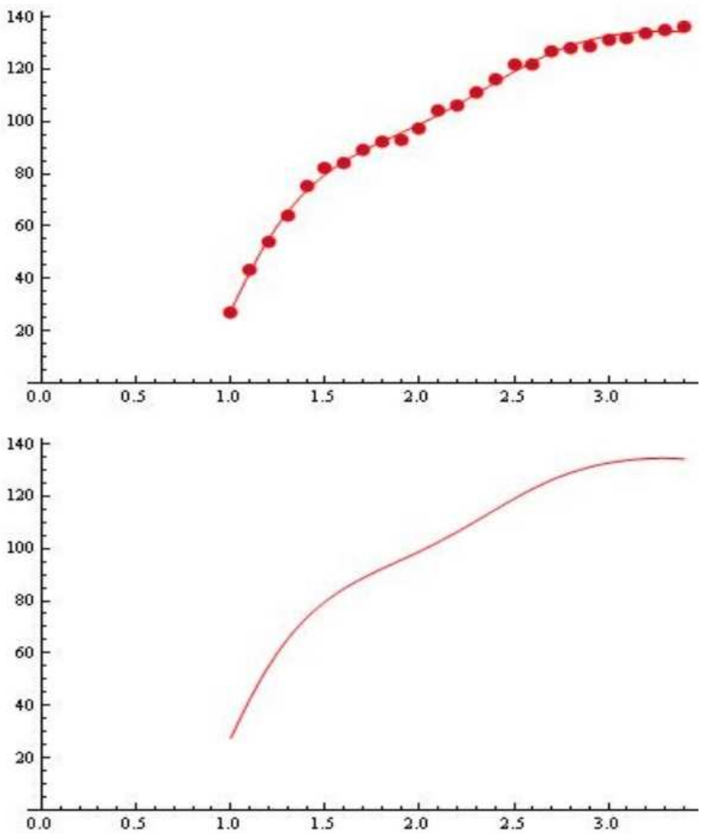

Figure 6: a) The "actual data"; b) The model (7) ($n = 5$, $A = 138$, $a_0 = 0$, $a_1 = -7.6955$, $a_2 = 17.948$, $a_3 = -13.0274$, $a_4 = 4.04697$, $a_5 = -0.45042$) for the data [7].

The only drawback is the fact that the model is very sensitive to the number and distribution of zeros of polynomial f [6].

We recommend the use of this model at a degree of polynomial $n \leq 10$.

In many cases, due to the large number of free parameters - a_i, standard program operators, implemented in different programming environments and designed to minimize heavy functionality (of many variables) - do not work or do not give satisfactory results.

This necessitated, when conducting the numerical examples contained in this article, to write a specialized program module, which is in a sense - overriding (for example, for the software environment *CAS Mathematica*).

For other approximation and modelling results, see [9]–[13].

ACKNOWLEDGMENTS

This paper is supported by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)", financed by the Ministry of Education and Science.

REFERENCES

[1] S. Yamada, M. Ohba, S. Osaki, S–shaped software reliability growth models and their applications, *IEEE Trans. Reliab.*, **33** No. 4 (1984), 289–292.

[2] V. Verma, S. Anand, A. Gupta Aggarwal, Reliability assessment of multi–release software system under imperfect fault removal phenomenon, *In: Decision analytics applications in industry*, Asset analytics (Eds.: P. Kapur et al.), Springer Nature, Singapore Pte Ltd. (2020), 367–380.

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

[4] N. Kyurkchiev, A. Iliev, A. Rahnev, On the Half-Logistic Model with "polynomial variable transfer". Application to approximate the specific "data CORONA VIRUS", *Int. J. of Differential Equations and Applications*, **19** No. 1 (2020), 45–61.

[5] N. Kyurkchiev, A. Iliev, A. Rahnev, *A Look at the New Logistic Models with "Polynomial Variable Transfer"*, LAP LAMBERT Academic Publishing (2020), ISBN: 978-620-2-56595-0.

[6] N. Kyurkchiev, *Initial approximation and root finding methods*, WILEY-VCH Verlag Berlin GmbH, **104** (1998).

[7] J. Musa, *Software reliability data*, Data and Analysis Center for Software, Rome Air Development Center, (1979).

[8] H. Okamura, T. Dohi, On Kolmogorov–Smirnov test for software reliability models with grouped data, *Proc. IEEE 19th International Conference on Software Quality, Reliability and Security*, (2019), 77–82.

[9] H. Pham, *System software reliability*, London, Springer (2006).

[10] 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.

[11] 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.

[12] O. Rahneva, A. Golev, G. Spasov, *Investigations on Some New Models in Debugging and "Growth" Theory (Part 3)*, LAP Lambert Academic Publishing (2020), ISBN: 978-620-2-66655-8.

[13] N. Kyurkchiev, A. Iliev, A. Golev, A, Rahnev, *Some Non-standard Models in "Debugging and Test Theory" (Part 4)*, Plovdiv, Plovdiv University Press (2020), ISBN: 978-619-2-02584-7.

[14] D. Costarelli, R. Spigler, Constructive Approximation by Superposition of Sigmoidal Functions, *Anal. Theory Appl.*, **29** (2013), 169–196.

[15] N. Pavlov, G. Spasov, M. Stieger, A. Golev, A Note on the Extended Song-Chang-Pham's Software Reliability Model. II, *International Journal of Differential Equations and Applications*, **18** No. 1 (2019), 87–98.

[16] 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.

[17] V. Kyurkchiev, A. Malinova, O. Rahneva, P. Kyurkchiev, On the Burr XII-Weibull Software Reliability Model, *International Journal of Pure and Applied Mathematics*, **119** No. 4 (2018), 639–650.

[18] V. Kyurkchiev, A. Malinova, O. Rahneva, P. Kyurkchiev, Some Notes on the Extended Burr XII Software Reliability Model, *International Journal of Pure and Applied Mathematics*, **120** No. 1 (2018), 127–136.

[19] V. Kyurkchiev, H. Kiskinov, O. Rahneva, G. Spasov, A Note on the Exponentiated Exponential-Poisson Software Reliability Model, *Neural, Parallel, and Scientific Computations*, **26** No. 3 (2018), 257–267.

[20] G. Spasov, O. Rahneva, A. Golev, Comments on a New Software Reliability Model, *Neural, Parallel, and Scientific Computations*, **27** No. 3&4 (2019), 185–200.

[21] G. Spasov, O. Rahneva, A. Golev, Investigations on a New Software Reliability Model. II, *Neural, Parallel, and Scientific Computations*, **27** No. 3&4 (2019), 203–213.

[22] A. Golev, G. Spasov, M. Stieger, A Note on the New Pham's Software Reliability Model, *Neural, Parallel, and Scientific Computations*, **27** No. 3&4 (2019), 151–164.

[23] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, Some Transmuted Software Reliability Models, *Journal of Mathematical Sciences and Modelling*, **2** No. 1 (2019), 64–70.

[24] 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.

[25] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, Transmuted Inverse Exponential Software Reliability Model, *International Journal of Latest Research in Engineering and Technology*, **4** No. 5 (2018), 1–6.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, A Power yet Under-Researched Software Reliability Model by Li and Pham, *Communications in Applied Analysis*, **23** No. 3 (2019), 453–464.

[31] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, Investigations of the K-stage Erlangian Software Reliability Growth Model, *International Journal of Pure and Applied Mathematics*, **119** No. 3 (2018), 441–449.

[32] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, On the Extended Chen's and Pham's Software Reliability Models. Some Applications, *International Journal of Pure and Applied Mathematics*, **118** No. 4 (2018), 1053–1067.

[33] N. Pavlov, A. Iliev, A. Rahnev, N. Kyurkchiev, Some Deterministic Growth Curves with Applications to Software Reliability Analysis, *International Journal of Pure and Applied Mathematics*, **119** No. 2 (2018), 357–368.

[34] N. Pavlov, G. Spasov, A. Iliev, A. Rahnev, N. Kyurkchiev, A Note on the Song-Chang-Pham's Software Reliability Model. Some Applications. I. *Neural, Parallel, and Scientific Computations*, **27** No. 2 (2019), 115–129.