

ADAPTING BASS-NIU MODEL FOR PRODUCT DIFFUSION TO SOFTWARE RELIABILITY

Sumantra Chakravarty

e-mail: sumontro@hotmail.com

INTRODUCTION

Software reliability growth is a well studied subject, perhaps starting with the classic work by Jelinski and Moranda [Jelinski 1972]. Applicability of reliability growth models is well established in where large (100 KLOC or more) software is written and maintained. Examples of such enterprises are application software, space exploration, telecommunication, etc. One can obtain the definition of standard terms (e.g., fault, failure) and operational summary of most widely used software reliability models from a document maintained by NASA Software Assurance Technology Center [Wallace].

Software reliability growth models address two important, but related, questions faced by the software industry: 1) How many remaining bugs are likely to be present in newly developed software and how much resources are needed for debugging to accepted level, 2) given that it is more expensive to fix a bug after software is released to the users, when it is economically prudent to release the software.

SOFTWARE RELIABILITY GROWTH MODELS

Software reliability growth models (SGRM) generally assume a finite but random number of initial faults that are revealed as failures according to a Non Homogeneous Poisson Process (NHPP). Time dependence of NHPP (see [Ross 2003] for an introduction to stochastic processes) rate is supposed to capture all underlying effects including learning effects by the debug team [Goel 1979, Ohba 1984, Pham 1999]. Some models also claim to incorporate effect of introducing new bugs during the actual debug process. Operationally, most SGRMs resort to estimating the mean value function of the NHPP using observed failure data and maximum likelihood estimation procedure (see [Hoel 1996] for an introduction to maximum likelihood estimation).

Observed software failure data tends to be S-shaped. Thus, SGRM applications of NHPP choose an S-shaped mean value function depending on two or more parameters. Homogeneous Poisson Process (HPP) is also used as SGRM [Ohba 1984] because of parsimony. However, exponential distribution arising from HPP does not fit all software failure data. When a software suite is composed of dissimilar modules, hyperexponential distribution has been used to model software reliability growth [Ohba 1984].

NHPP is not the only family of stochastic models used to model software reliability growth. Weibull PDF has also been used to model time rate of bug appearance [Kenny 1993]. Weibull model is justified when debug effort increases with calendar time as a power law and this is the dominant effect in determining the number of remaining bugs. If the debug effort grows linearly with time, an exponential model is obtained.

A common theme in SGRM is to a process or, a distribution and fit some observed time sequence of failure times to the model. One model may successfully summarize some datasets and fail to summarize some other datasets. Thus, statistical goodness of fit measure is the only criteria that can be used to evaluate any given SGRM. Sum of Squared Errors (SSE), Akaike Information Criterion (AIC), and Chi-square tests have been used in the literature [Pham 1999].

MODELS WITH RANDOM EXECUTION MEDIA

Software reliability growth models, except for the Jelinski-Moranda model [Jelinski 1972], described above assumes the number of bug to be a random quantity initially. We take the view that initial number of bugs in a code is fixed by unknown quantity. Execution media for the code provides a random environment. If different debug teams were to debug the same code, different realizations of fault detection would be achieved.

Ross has a model for software reliability growth starting with fixed number of bugs [Ross 2003]. However, practical use of the Ross model may be limited because it requires classification of faults according to the number of failures it causes. Lee et al. has recently attempted to model software reliability growth by a stochastic equation where random execution environment has been modeled by ad hoc Gaussian noise source [Lee 2004]. Lee model seems to fit much of the published data on software failures.

We posit another model for software reliability growth based on analogy with product diffusion of consumer durable goods. Bass-Niu (BN) formulation attempts to provide a proper stochastic formulation starting with fixed number of bugs in the code in the spirit of Jelinski et al., Ross and Lee at al. Bass model also provides interpretation of the parameters (p and q) based on intrinsic complexity of the code and ability of a debug team. Bass-Niu model also attempts to solve a problem with NHPP based SGRM. Because NHPP assumes independent increments in disjoint time intervals, it is difficult to model introduction of new bugs within NHPP setting.

BASS MODEL FOR PRODUCT DIFFUSION

Bass model for was introduced in the marketing research community to model production diffusion (adoption) of a durable (e.g., VCR) good [Bass 1969]. Let M be the size of the potential market and $N(t)$ be the number of adoptions by time t . It is assumed that M is large and continuous approximation of $N(\cdot)$ is justified. Product adoption is parametrized by to real positive constants p (coefficient of innovation) and q (coefficient of imitation). Bass postulates that time evolution of N , starting with no adoption at $t=0$, is described by the following (non-linear) differential equation.

$$\frac{dN(t)}{dt} = [M - N(t)] \left[p + q \frac{N(t)}{M} \right]; t \geq 0$$

Let $F(t)$ be the fraction of adoptions by time t . As $M \rightarrow \infty$, we have the Bass equation and the Bass formula (CDF) as its solution. Bass CDF is derived from the Bass equation in Appendix-A.

$$\frac{dF(t)}{dt} = [1 - F(t)] \cdot [p + qF(t)]; t \geq 0, F(0) = 0$$

$$\Leftrightarrow F(t) = \frac{1 - e^{-(p+q)t}}{1 + \left(\frac{q}{p}\right) e^{-(p+q)t}}; t \geq 0, p > 0, q \geq 0 .$$

Classic Bass solution offers some desirable features observed in production diffusion. $F(t)$ is S-shaped and has an inflection point. It also fits observed market data for many durable goods. However, transition from $N(\cdot)$ to $F(\cdot)$ assumes $M \rightarrow \infty$. This is difficult to justify in software reliability setting because it is not possible have infinite number of faults in a software with finite number of LOC (lines of code).

Notwithstanding this apparent difficulty, we will explore the similarities between the process of production and software reliability growth after introducing stochastic version of the Bass model.

STOCHASTIC BASS-NIU MODEL

Niu has provided a stochastic framework of the Bass model [Niu 2002]. Bass-Niu formulation of product diffusion is a pure-birth model (see [Ross 2003] for an introduction to pure-birth models) with birth rates:

$$\lambda_n = (M - n) \left[p + q \frac{n}{M - 1} \right]; n = 0, \dots, M - 1 .$$

$$\lambda_M = 0$$

Coefficients p and q have same interpretation as in the classic Bass Model. Note that deterministic Bass equation involved M is the denominator of its RHS. However, difference of M and $(M-1)$ is negligible as $M \rightarrow \infty$. Additionally, Niu has shown that

$$\lim_{M \rightarrow \infty} F_M(t) = \lim_{M \rightarrow \infty} E \left[\frac{n}{M} \right] = F(t) .$$

$$F(0) = 0$$

Time dependence of RHS arises from taking expectation with respect to time dependent probability. Thus, the BN model converges in the mean to the Bass formula and this equation exhibits the connection between the two models. Everyone adopts the product by $t \rightarrow \infty$ and $F(\infty) = 1$. This justifies calling the Bass fraction as a Cumulative Distribution Function (CDF). Niu also provides a differential equation for $F_M(t)$, $F(t)$ with finite M . We will not quote Niu's derivation here for brevity. Instead, we will provide an elementary derivation for $F_M(t)$ in Appendix-B.

Niu has found exact expressions for the asymptotic ($M \rightarrow \infty$) of variance M [Niu 2005].

$$\psi(t) = \lim_{M \rightarrow \infty} M \cdot \text{Var}\left(\frac{n}{M}\right) = F(t)[1 - F(t)] + C(t)$$

$$C(t) = \frac{(1 + q/p)(q/p)(1 - e^{-(p+q)t})}{[1 + (q/p)e^{-(p+q)t}]^4 e^{2(p+q)t}} \left\{ 2 \left[\frac{(p+q)t}{1 - e^{-(p+q)t}} \right] + \left(\frac{q}{p} \right) (1 - e^{-(p+q)t}) \right\}$$

Expression for asymptotic variance involves the Bass CDF and complement. It also involves contribution from the covariance of indicator function (denoting j^{th} adoption by time t ; see [Niu 2005] for details). Expressions of mean and variance of adoptions can be used to adapt the BN model product diffusion to software reliability growth.

BASS-NIU MODEL FOR SOFTWARE RELIABILITY GROWTH

Bass-Niu model for software reliability growth adapts the BN model for product diffusion with a different interpretation of parameters n , M , p and q . We will rewrite the BN birth rates for ease of reading.

$$\lambda_n = (M - n) \left[p + q \frac{n}{M - 1} \right]; n = 0, \dots, M - 1$$

$$\lambda_M = 0$$

M is interpreted to be the number of bugs initially present in the software. Here M is assumed to be fixed (not random) but unknown. λ denotes the time rate of bug appearance and n denotes the number of bugs found (assuming no new introduction).

We would like to solve the Bass-Niu equation for finite (but moderately large) M without further assumption on p or, q . An approximate solution is presented in Appendix-C. Another scheme to find a better solution, based on backward Kolmogorov equations, is presented in the section describing future work.

INTERPRETATION OF PARAMETERS P AND Q

Parameter p is a measure of intrinsic complexity of the code (for given M , because number of initial bugs and effective software complexity are likely to correlate with LOC). Effect of p can be understood by assuming $q=0$. If $q=0$, there is no debugging, and each bug reveals itself independently with mean time $1/p$. Bugs in a more complex software take longer to reveal themselves. (Exponential duration between two successive revelations is an approximation of the execution environment.)

Parameter q is a measure of learning by a particular debug team. Let's note that time of bug appearance increases initially (small value of n). Starting with no knowledge about the software, the debug team learns more about the software as another bug is isolated and eliminated. Let us also note that learning contribution to bug appearance rate enters a region of diminishing returns after enough (about \sqrt{M}) bugs have been fixed. This implies onset of learning saturation.

INTERPRETATION OF TIME DEPENDENT VARIANCE

Let $\lambda(t)$ be the intensity (rate) of a Non Homogeneous Poisson Process. It is well known [Ross 2003] that number of events in time interval $(0, t)$ is given by a Poisson distribution with mean

$$\int_0^t \lambda(u).du .$$

Moreover, the variance of a Poisson distribution is equal to the mean. This presents a problem for NHPP based SGRM. Let us assume no new bugs are introduced while debugging. Then, we expect all the bugs to be revealed in the time interval $(0, \infty)$. If software reliability is modeled as a Poisson process, variance of the number of bugs must be interpreted as the variance of initial number of bugs. Second, why should the variance grow as more bugs are observed? These present conceptual difficulties.

In contrast, Bass-Niu model assumes fixed number of initial bugs and physically plausible behavior for time dependent variance. Niu has calculated the expression for variance as $M \rightarrow \infty$ [Niu 2005]. Asymptotic variance of number of observed bugs is zero at $t=0$ and $t=\infty$. No bugs have been observed at $t=0$, with certainty; no bugs remain at $t=\infty$, also with certainty.

FITTING PUBLISHED SOFTWARE RELIABILITY DATA TO BASS-NIU MODEL

NHPP based models have been reasonably successful in modeling software reliability growth [Goel 1979, Ohba 1984, Pham 1999]. Further evidence comes from adoption of these models by NASA [Wallace]. Primary focus of this work is to address some theoretical questions presented by NHPP based SGRM, while not compromising on practical applicability.

It is easy to reinterpret asymptotic ($M \rightarrow \infty$) Bass CDF as the "inflection S-shaped growth function $h(t)$ of NHPP" (compare with Eq. 8 of [Ohba 1984] with proper scaling by M).

$$h(t) = M \frac{1 - e^{-\phi t}}{1 + \psi . e^{-\phi t}} ; t \geq 0, \phi > 0, \psi \geq 0, M > 0$$

$$F(t) = \frac{1 - e^{-(p+q)t}}{1 + \left(\frac{q}{p}\right) e^{-(p+q)t}} ; t \geq 0, p > 0, q \geq 0$$

We expect asymptotic solution of Bass-Niu model to be numerically close to $F_M(t)$ when M is moderately large. Thus, Bass-Niu model automatically fits observed software failure data fitted by inflection S-shaped NHPP-SGRM. As an example, Project 2 data of Rome Air Development Project [Brooks 1980] is fitted by $p = 0.0084$ and $q = 0.213$ or, $(p+q) = 0.221$ and $(q/p) = 25.3$. Fitted value of M is 1315.9 in this example. We note, magnitude of q is much bigger than that of p for this dataset.

FUTURE WORK BASED COAGULATION ANALOGY

We have found it difficult to solve the Bass-Niu equation by quadrature for large values of the learning parameter q because of non-linearity in the BN differential equation. This mathematical issue limits the applicability of the BN model to software reliability. There is another promising technique based on solving the backward Kolmogorov equation (see [Ross 2003] for an introduction to Kolmogorov equations) for the BN pure-birth model.

The backward Kolmogorov equation was solved exactly for a mathematically related problem of stochastic death model of droplet coagulation [Arciapiani 1980]. Arciapiani's method involves solving the backward Kolmogorov equations in matrix form by finding the eigenvalues and eigenvectors of the matrix of "death coefficients." Coefficient matrices for Arciapiani as well as BN models are bi-diagonal with quadratic terms (in number of droplets or, software bugs detected). Thus, eigenvalues can be found almost by inspection. BN birth rates turn out to be the eigen-values of the coefficient matrix for backward Kolmogorov equations.

Identification of eigenvalues leads to valuable insight into the system. As the Kolmogorov equations from a system of linear differential equations, the solution (state probabilities) is a linear combination of $\exp\{-\lambda_n t\}$, $n = 0, 1, \dots, (M-1)$. Let us observe that birth-rates of the Bass-Niu model may be rewritten as

$$\begin{aligned}\lambda_0 &= Mp \\ \lambda_1 &= (M-1)p + q \\ &\dots \\ \lambda_{M-1} &= 2(p+q) + \frac{2q}{M-1} \\ \lambda_M &= p + q\end{aligned}$$

Thus, birth rates for higher states (λ with large values of sub-script) tend to independent of M whereas, birth rates for low states tend to grow with M . Thus, only higher states contribute when M becomes large (with p , q and t fixed). However, finding the eigenvectors requires significant effort and this may be a topic for future work.

CONCLUSION

We have introduced NHPP based software reliability growth model. These models suffer from some theoretical problems even though they are able to fit observed failure data reasonably well. We have also introduced the Bass-Niu model for market diffusion of durable goods. We have mapped a particular (inflection S-shaped) NHPP based SGRM to the asymptotic solution Bass-Niu model. This provides empirical validation for Bass-Niu model in software reliability. We have provided approximate solution for the Bass-Niu model for large (but finite) M in Appendix-C. Finding an exact solution to the Bass-Niu model may be a project for the future.

ACKNOWLEDGEMENTS

We would like thank Prof. Shun-Chen Niu for helpful discussions.

REFERENCES

- Arcipiani, B. (1980), "The Backward Kolmogorov equation for statistical distribution of coagulating droplets," *J. Phys. A: Math. Gen.* Vol. 13, pp. 3367-3372.
- Bass, F.M. (1969), "A new product growth model for consumer durables," *Management Sciences*, Vol. 15, pp. 215-227.
- Brooks, W.D. and R.W. Motley (1980), "Analysis of Discrete Software Reliability Models," *Technical Report RADC-TR-80-84*, Rome Air Development Center, New York.
- Goel, A.K. and K. Okumoto (1979), "Time Dependent Error Detection Rate Model for Software Reliability and Other Performance Measures," *IEEE Transactions on Reliability*, R-28, pp. 206-211.
- Hoel, P.G., S.C. Port and C.J. Stone (1996), *Introduction to Statistical Theory*, Houghton Mifflin Company.
- Jelinski, Z. and P. Moranda (1972), "Software Reliability Research," *Statistical Computer Performance Evaluation*, W. Freidberger (ed.), Academic Press, pp. 465-484.
- Kenney, G.Q. (1993), "Estimating Defects in Commercial Software During Operational Use," *IEEE Transactions on Reliability*, V. 48, No. 1, pp. 107-115.
- Lee, C.H., Y.T. Kim and D.H. Park (2004), "S-shaped software reliability growth models derived from stochastic differential equations," *IEEE Transactions*, V. 36, pp. 1193-1199.
- Niu, S.-C. (2002), "A Stochastic Formulation of the Bass Model New-Product Diffusion," *Mathematical Problems in Engineering*, Vol. 8(3) pp. 249-263.
- Niu, S.-C. (2005), "A Piecewise-Diffusion model of New Product Diffusion," The University of Texas at Dallas School of Management Working paper, to appear in *Operations Research*.
- Ohba, M. (1984), "Software Reliability Analysis Models," *IBM Journal on Research and Development*, V. 28, No. 4.
- Pham, H., L.N. Rutgers and X. Zhang (1999), "A General Imperfect-Software-Debugging Model with S-Shaped Fault-Detection Rate," *IEEE Transactions on Reliability*, Vol 48, No. 2, pp. 169-175.
- Ross, S.M. (2003), *Introduction to Probability Models*, 8th ed., Academic Press.
- Wallace, D. and C. Coleman, "Hardware and Software Reliability (323-08)," NASA Software Assurance Technology Center.

APPENDIX-A: Derivation of Bass Formula

We start with the Bass differential equation

$$\begin{aligned} \frac{dF(t)}{dt} &= [1 - F(t)][p + qF(t)]; t \geq 0, F(0) = 0 \\ \Leftrightarrow \frac{1}{F'(t)} &= \frac{1}{1 - F(t)} \cdot \frac{1}{p + qF(t)} = \frac{1}{p + q} \cdot \frac{1}{1 - F(t)} + \frac{q}{p + q} \cdot \frac{1}{p + qF(t)}. \\ \Leftrightarrow (p + q)t + \text{const1} &= -\ln(1 - F) + \ln(p/q + F) \\ \Leftrightarrow \frac{1 - F(t)}{p/q + F(t)} &= \text{const2} \cdot e^{-(p+q)t} \end{aligned}$$

Initial condition $F(0)=0$ implies $\text{const2}=(q/p)$.

$$\begin{aligned} 1 - F(t) &= e^{-(p+q)t} + \frac{q}{p} e^{-(p+q)t} F(t) \\ \Leftrightarrow 1 - e^{-(p+q)t} &= F(t) \cdot \left[1 + \frac{q}{p} e^{-(p+q)t} \right] \end{aligned}$$

Finally we get,

$$F(t) = \frac{1 - e^{-(p+q)t}}{1 + (q/p) \cdot e^{-(p+q)t}}$$

Inflection point of Bass CDF can be found computing its second derivative. We start with the Bass differential equation.

$$\begin{aligned} \frac{dF(t)}{dt} &= [1 - F(t)][p + qF(t)]; t \geq 0, F(0) = 0 \\ \Leftrightarrow \frac{1}{F'(t)} &= \frac{1}{1 - F(t)} \cdot \frac{1}{p + qF(t)} = \frac{1}{p + q} \cdot \frac{1}{1 - F(t)} + \frac{q}{p + q} \cdot \frac{1}{p + qF(t)} \\ \Leftrightarrow \frac{-F''}{(F')^2} \cdot (p + q) &= \frac{F'}{(1 - F)^2} - \frac{F'}{(q/p + F)^2} \end{aligned}$$

Location of the inflection point is found by setting its second derivative to zero.

$$\begin{aligned} F''(t_*) = 0 &\Rightarrow (1 - F) = (p/q + F) \\ \Rightarrow F(t_*) &= \frac{1}{2} \left(1 - \frac{p}{q} \right) \\ \Rightarrow t_* &= \frac{-\ln\left(\frac{p+q}{4q}\right)}{p+q} \end{aligned}$$

We note that inflection point of Bass CDF exists only if $q > p$.

APPENDIX-B: Derivation of Bass-Niu equation

Finite state, pure-birth BN model is defined by the birth rates

$$\lambda_n = (M - n) \left[p + q \frac{n}{M - 1} \right]; n = 0, \dots, M - 1$$

$$\lambda_M = 0$$

Forward Kolmogorov equations for this model (dash denotes time derivative), together with probability normalization and initial conditions are

$$p_0' = -\lambda_0 p_0$$

$$p_n' = \lambda_{n-1} p_{n-1} - \lambda_n p_n; n = 1, \dots, (M - 1)$$

$$p_M' = +\lambda_{M-1} p_{M-1}$$

$$1 = \sum_{n=0}^M p_n(t)$$

$$p_0(0) = 1; p_n(0) = 0; n = 1, \dots, M$$

We can see that state- M is an absorbing state from the evolution equation for p_M . Fractional penetration (average number of bugs revealed) by time t is

$$F_M(t) = \frac{1}{M} \sum_{n=0}^M n p_n(t) = \sum_{n=0}^M p_n(t) \frac{n}{M}$$

Using the "tail formula" for expectation we get

$$F_M(t) = \frac{1}{M} \sum_{n=0}^M n p_n(t) = \frac{1}{M} \sum_{n=0}^M \sum_{k=n+1}^M p_k(t) = \frac{1}{M} \sum_{n=0}^M \left(1 - \sum_{k=0}^n p_k(t) \right)$$

$$\Leftrightarrow F_M(t) = \frac{M + 1}{M} - \frac{1}{M} \sum_{n=0}^M \sum_{k=0}^n p_k(t)$$

We may use the forward Kolmogorov equations to derive a system of differential equations for the cumulative probability.

$$\pi_n(t) = \sum_{k=0}^n p_k(t);$$

$$\pi_0(t) = p_0(t); \pi_M(t) = 1;$$

$$\pi_n' = -\lambda_n p_n(t); n = 0, \dots, M - 1$$

$$\pi_n(0) = 1$$

Differentiating the expression for $F_M(t)$ on both sides with respect to t , we get a (formal) differential equation get after some algebraic manipulations using the forward equations for cumulative probability.

$$\frac{dF_M(t)}{dt} = \sum_{n=0}^M p_n(t) \frac{\lambda_n}{M}.$$

Using the definition of birth rates, we finally have

$$\frac{dF_M(t)}{dt} = [1 - F_M(t)][p + qF_M(t)] + \frac{q}{M-1} \left[F_M(t) \cdot \{1 - F_M(t)\} - M \cdot \text{Var} \left(\frac{n}{M} \right) \right].$$

Asymptotic expression for the variance can be obtained from [Niu 2005].

APPENDIX-C: Reducing asymptotic Bass-Niu equation to quadrature

Bass-Niu differential involves the variance. Exact expression for variance for finite M is unknown at time. We will use asymptotic expression for variance found by Niu to find an approximate differential equation for large M . This approximation is justified when $q/(M-1) \ll 1$. Faced with a practical software reliability growth problem of complex software, we expect large number of undiscovered bugs initially (moderate to large M).

Even after the asymptotic expression for variance is substituted, we are unable to solve the resulting nonlinear differential equation. Hence, a further linearizing approximation becomes necessary. In the following, we will use work with the complimentary CDF

$$G_M(t) = 1 - F_M(t)$$

And we will linearize the asymptotic differential equation by approximating $G_M(t) = G(t) + \epsilon(t)$. From the Bass-Niu differential equations we have

$$-G'(t) = G(t) \cdot [p + q - q \cdot G(t)]$$

$$-G'_M(t) = G_M(t) \cdot [p + q - q \cdot G_M(t)] + \frac{q}{M-1} [F(t) \cdot G(t) - \psi(t)]$$

Bass CDF and CCDF are linearized in the following way

$$G_M(t) = G(t) + \epsilon(t); F_M(t) = F(t) - \epsilon(t)$$

$$F_M(t) \cdot G_M(t) = F(t) \cdot G(t) + [1 - 2 \cdot G(t)] \cdot \epsilon(t) - \epsilon(t)^2$$

$$G_M(t)^2 = G(t)^2 + 2 \cdot G(t) \cdot \epsilon(t) + \epsilon(t)^2$$

This give to the following linearized differential equation for the correction term

$$\frac{d\epsilon(t)}{dt} = - \left[p + q - 2G(t) + \frac{q}{M-1} (1 - 2 \cdot G(t)) \right] \epsilon(t) + \frac{q}{M-1} C(t)$$

Let us change the independent (t) variable to

$$\tau = e^{-(p+q)t}; \frac{d}{dt} = -(p+q)\tau \frac{d}{d\tau}$$

$$t = 0 \Rightarrow \tau = 1, t = \infty \Rightarrow \tau = 0$$

When $M \cdot q \cdot \epsilon(\tau) / (M-1) \ll 1$, the differential equation reduces to

$$\frac{d\varepsilon_M(\tau)}{d\tau} - g(\tau) = \frac{1}{M-1} \cdot \frac{q/p}{1+q/p} \cdot \frac{C(\tau)}{\tau}$$

$$g(\tau) = \frac{K_1}{\tau} - \frac{K_2'}{1+(q/p)\tau}$$

$$K_1 = 1 + \frac{q/p}{(1+q/p).(M-1)}; K_2' = \frac{2M(q/p)}{(1+q/p).(M-1)}$$

Integrating factor for this first order differential equation is

$$I(\tau) = \exp\left[-\int g(u).du\right] = -\ln\left[\frac{(1+q/p\tau)^{K_2}}{\tau^{K_1}}\right]$$

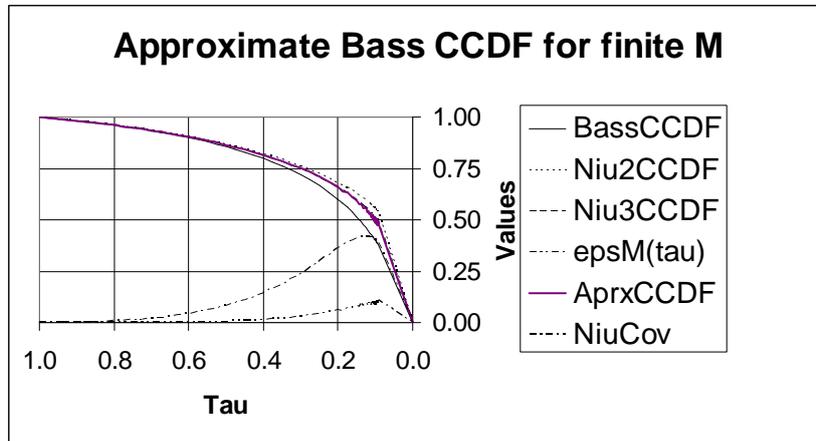
$$K_2 = \frac{p}{q} K_2' = \frac{2M}{(1+q/p).(M-1)}$$

Finally, approximate solution for the Bass-Niu equation is given by

$$\varepsilon_M(\tau) = \frac{1}{M-1} \cdot \frac{q/p}{1+q/p} \cdot \frac{1}{I(\tau)} \int_1^\tau I(u) \frac{C(u)}{u} du$$

$$F_M(\tau) \approx F(\tau) - \varepsilon_M(\tau); G_M(\tau) \approx G(\tau) + \varepsilon_M(\tau)$$

We illustrate the usefulness of this approximation with an example.



We have chosen $M=3$, $p=1$ and $q=5$ in this example. The plot shows exact expressions for the CCDF for $M=2$ and 3 [Niu 2002]. It also shows the Bass formula corresponding to $M=\infty$. Source term in the linearized differential equation, $C(\tau)$ [Niu 2005], is also plotted.

Integration of the linearized differential equation was performed in MS Excel with variable step Simpson method. This led to numerical evaluation of $\varepsilon_M(\tau)$. Distortion around $\tau=0$ ($t=\infty$) is an artifact of this Excel based integration scheme. Let's note that approximate expression almost coincides with the exact expression for this illustrative example.