A Nonparametric Software Reliability Growth Model

Douglas R. Miller*
The George Washington University

Ariela Sofer
George Mason University

August 1988

Abstract

Recently Miller and Sofer have presented a new nonparametric method for estimating the failure rate of a software program. The method is based on the complete monotonicity property of the failure rate function, and uses a regression approach to obtain estimates of the current software failure rate. This paper extends this completely monotone software model and demonstrates how it can also provide long-range predictions of future reliability growth. Preliminary testing indicates that the method is competitive with parametric approaches, while being more robust.

1 Introduction

Suppose a program is executed for a length of time T. During this time, n bugs are detected and removed when they manifest themselves as failures. The successive failures occur at times

$$0 < t_1 < t_2 < \ldots < t_n < T. \tag{1}$$

When bugs are corrected without introducing new faults, the program evolves into a more reliable program, hence the term "reliability growth". Given the past software data (1) we would like to make various statistical inferences concerning the current and future reliability of the software. In particular we are interested in the number of failures expected over some future horizon, the present failure rate, and the future failure rate after an additional specified time of debugging.

Over the years, many competing models for software reliability growth have been developed. These models include those by Duane(1964), Jelinski and Moranda(1972), Goel and Okumoto(1979), Littlewood(1981) and Musa and Okumoto(1984). These are all parametric models. It is interesting to note that all the above models have a common property: complete monotonicity of the failure rate function. Let $N(t)$ be the (random) number of failures observed in $[0, t]$, and let $M(t) = E(N(t))$ be the expected number of failures. $M(t)$ is known as the mean function. The intensity function of the point process $\{N(t), 0 \leq t\}$ is $r(t) = dM(t)/dt, 0 < t$. The function $r(t)$ is also referred to as the failure rate of the process. A function $r(\cdot)$ is completely monotone if and only if it has derivatives of all orders, and they alternate in sign as follows:

$$(-1)^q \frac{d^q r(t)}{dt^q} \geq 0, \ t \geq 0, \ q = 0, 1, 2, \ldots. \tag{2}$$

Miller(1986) has shown that virtually all completely monotone functions can occur as intensity functions of reliability growth point processes. Thus, a general approach to software reliability growth modelling should include the entire class of completely monotone intenisities.

Miller and Sofer(1985) have previously introduced a nonparametric model for software reliability growth which is based on complete monotonicity of the failure rate function. The method uses a regression approach

to obtain estimates of the current software failure rate. Miller and Sofer(1986a) show that this method often gives estimates which have a lower bias than those of certain (widely-used) nonparametric methods; using Monte Carlo simulated failure data, these "completely monotone regression" estimates of current failure rate are also shown to be more robust than the estimates based on parametric models. Chan(1986) has estimated the distribution of time until next failure for real data using completely monotone regression estimates of current reliability: he starts with a raw estimate which is an exponential distribution with the estimated current failure rate and then "adapts" it to a more general distribution using the procedure of Littlewood and Keiller(1984). Chan then evaluates these estimates using criteria of Abdel-Ghaly, Chan and Littlewood (1986). This study shows that the completely monotone regression approach gives good estimates which are more robust than estimates from parametric models.

This paper extends the completely monotone software model by developing a method for providing long-range predictions of future reliability growth, based on the model. The paper derives upper and lower bounds on extrapolations of the failure rate function and the mean function into the future. These are then utilized to obtain estimates for the future software failure rate and the expected future number of failures.

2   Problem Formulation

Consider the failure data as in (1). Our goal is to find a completely monotone rate function and/or the associated mean function which best fits the data. (We note here that the mean function does not strictly satisfy the complete monotonicity property; rather, $M(t)$ is a nonnegative function whose derivative $dM(t)/dt$ is a completely monotone function). Our approach will be to obtain an initial raw estimate for the required function from the data, and then to smooth it by fitting a completely monotonic function which is closest to it in the least squares sense.

A natural raw estimate $\hat{M}(t)$ for the mean function is a piecewise linear function with breakpoints at t_i, $i = 1, \ldots, n$, such that $\hat{M}(t_i) = i$. Analytically, this function is given by

$$\hat{M}(t) = \begin{cases} i + (t - t_i)/(t_{i+1} - t_i) & t_i \le t \le t_{i+1}; \ i = 0, \ldots, n-1 \\ n + \delta(t - t_n)/(T - t_n) & t_n \le t \le T. \end{cases} \tag{3}$$

The second term in the final interval reflects the absence of a failure in the period $(t_n, T]$. The choice of δ is somewhat arbitrary, with higher values tending tending to give more conservative estimates. In this work we consider values of 0.0, 0.5 and 1.0 for δ, however one can argue for and against any particular value.

In practice, it is necessary to discretize the problem of finding a completely monotone function to the mathematically more tractable problem of finding a finite set of points along that function. The most plausible and straightforward approach is to consider discrete time points which are equally spaced. We thus divide the time interval $[0, T]$ to k intervals of equal length $\theta = T/k$, and define $s_i = i\theta$, $i = 0, \ldots, k$. Thus the sequence $\hat{m}_i = \hat{M}(s_i)$ will be an initial estimator for the values of the mean function at the fixed intervals s_i. In general, however, this sequence will not satisfy the complete monotonicity assumptions of the model, and will thus need some modification.

For the problem of estimating the rate function, one obtains an initial estimator from the slope of the mean function estimator $\hat{M}(t)$. Specifically, the sequence

$$\hat{r}_i = (\hat{m}_i - \hat{m}_{i-1})/\theta; \ i = 1, \ldots, k$$

will constitute a raw estimate of the failure rate function at the points s_i.

When working with discrete, equally spaced time points, the analogue of a completely monotone function is a completely monotone sequence. The sequence $(r_i, i = 1, 2, \ldots)$ is completely monotone if

$$(-1)^j \Delta^j r_i \ge 0, \ j + 1 \le i; \ j = 0, 1, \ldots \tag{4}$$

where Δ^j is the j-th order backward difference operator

$$\Delta^0 r_i = r_i, \ \Delta^1 r_i = r_i - r_{i-1}, \ \Delta^j r_i = \Delta^{j-1} r_i - \Delta^{j-1} r_{i-1}, \ j > 1 \tag{5}$$

In general, the initial estimate $(\hat{r}_1, \ldots, \hat{r}_k)$ will not have the complete monotonicity property. Our goal will be to find the "closest" completely monotone sequence $(r_1, \ldots, r_k)$, and use it as an estimate of the sequence

of failure rates at times s_i. Using the criterion of weighted least squares, the problem is to find a vector r which minimizes

$$D(r, \hat{r}) = \sum_{i=1}^{k} w_i(r_i - \hat{r}_i)^2 \tag{6}$$

subject to the complete monotonicity constraints of (4), where w_i is a set of prespecified weights.

In practice, numerical experience indicates that the effect of the very high order difference constraints on the optimal solution is at most marginal; moreover, their presence leads to ill-conditioning of the optimization problem. Consequently we relax the constraints in (4) and consider differences of at most d (not ∞), with d being typically 3 or 4. Similarly, it is unnecessary to constrain the sequence infinitely far into the future; we shall restrict the number of future intervals to l, rather than ∞. Finally, it should be noted that many of the constraints in (4) are redundant, e.g., $\Delta r_{i-1} \leq 0$ and $\Delta^2 r_i \geq 0$ imply that $\Delta r_i \geq 0$. Eliminating those redundant constraints, we finally obtain the reduced system of equations

$$
\begin{aligned}
(-1)^d \Delta^d r_i &\geq 0, & d+1 \leq i \leq k+l \\
(-1)^j \Delta^j r_{k+l} &\geq 0, & 0 \leq j \leq d-1,
\end{aligned} \tag{7}
$$

and our problem is to minimize (6) subject to (7).

Notice that for the case $d = 1$, the problem is the well known "isotone regression" problem described by Barlow et al. (1972) and addressed in the reliability growth context by Campbell and Ott (1979) , and Nagel et al. (1984). If the last interfailure happens to come from the right tail of the interfailure time distribution, $\hat{r}_k$ will underestimate $r(T)$, and the monotone constraint on r will have no effect, thereby leading to a negative bias. Imposing the additional constraint of convexity tends to pull this estimate up. In most software reliability applications, a positively biased estimate of the failure rate is safer than a negatively biased estimate; thus, higher order constraints seem to be desirable, and the generalization of isotone regression to completely monotone regression an improvement.

Let us return to the problem of estimating the mean function. Recalling that its first order derivative is a completely monotone function, and using the above, our problem is

$$
\begin{aligned}
\min \quad & D(m, \hat{m}) = \sum_{i=1}^{k} w_i(m_i - \hat{m}_i)^2 \\
\text{subject to} \quad & (-1)^{d+1} \Delta^d m_i \geq 0, & d \leq i \leq k+l \\
& (-1)^j \Delta^j m_{k+l} \geq 0, & 0 \leq j \leq d-1 \\
& m_k = n + \delta,
\end{aligned} \tag{8}
$$

where m_0 is defined as zero. If testing stopped at a failure, (i.e., $t_n = T$), the value $\delta = 0$ should be used. In the case of truncated testing however, the value $\delta = 0.5$ is a more plausible choice. Using an argument based on the assumption of a Poisson Process, a value $\delta = 1$ is also a plausible choice.

The optimization problems presented above are linearly constrained quadratic programming problems, and algorithms for their solution are readily available in the literature. However, it can be shown (see Miller and Sofer(1986b)) that our particular problem of least squares regression under higher order difference constraints becomes increasingly ill-conditioned as the problem size grows. Thus, a numerically stable algorithm should be employed for its solution. For a detailed description of a viable solution approach, see Miller and Sofer (1986b).

An additional difficulty which arises, when attempting to include monotonicity requirements into the future, is that the Hessian matrix, i.e., the matrix of the second order derivatives of the objective function is singular, since the future predictions r_i and m_i (where $i = k+1, \ldots, k+l$) do not appear in the objective. Moreover, the optimal future rate or mean estimators obtained by the least squares objective are not unique. In section 3, we show how to overcome the problem of singularity, by reformulating the constraints on the future rates (or mean function estimates) in terms of those of the past. Surprisingly, this approach also provides bounds – lower and upper envelopes for these future estimates.

3 Predictions into the Future

As mentioned previously, formulation (8) gives rise to some computational problems, when future predictions are requested, i.e., when $l > 0$. Algorithms for solving quadratic programming problems (see McCormick(1983)) usually require that the Hessian matrix of the objecitve function be positive-definite. However, the Hessian matrix of the objective for (6), (i.e., diag $(w_1, \ldots, w_k)$) is only positive semi-definite, and does in fact have singularities. As a result, not only will we encounter numerical difficulties when trying to solve the problem directly, but the optimal solution will not be unique. Indeed, any two solution vectors where the first k components are equal, will yield exactly the same objective value. In other words, if the completely monotone sequence $(r_1, \ldots, r_k)$ can be extrapolated l time intervals into the future, in a way that the resulting sequence $(r_1, \ldots, r_{k+l})$ is completely monotone, then *all* such possible extrapolations will have the same least squares objective. In the following we show, that among all such extrapolations, there exist a globally highest and a globally lowest extrapolation, and all other completely monotone extrapolations into the future must lie in between the highest and lowest bounds. We thus have an envelope in which all completely monotone extrapolations are restricted. In addition, we derive the conditions under which the sequence $(r_1, \ldots, r_k)$ can be extrapolated as a completely monotone sequence into the future.

Consider the completely monotone sequence of order d: $R = (r_1, \ldots, r_k)$. The sequence $(r_{k+1}, \ldots, r_{k+l})$ is defined to be a feasible completely monotone extrapolation of order d for R, if the sequence $(r_1, \ldots, r_{k+l})$ is completely monotone up to order d, i.e., it satisfies (7). In addition, this extrapolation constitutes an upper bound for all feasible extrapolations of order d, if any other such extrapolation, $(\bar{r}_{k+1}, \ldots, \bar{r}_{k+l})$ satisfies $\bar{r}_{k+i} \leq r_{k+i}$ for $i = 1, \ldots, l$. Similarly it constitutes a lower envelope if $\bar{r}_{k+i} \geq r_{k+i}$ for all i. In the following, we derive conditions for the existence for such higher and lower envelopes for the completely monotone extrapolations.

For the case $d = 1$ and $d = 2$, the sequence $(r_1, \ldots, r_k)$ can be extrapolated into the future by letting $r_{k+i} = r_k$, $i = 1, \ldots, l$. This extrapolation is clearly the upper envelope for all completely monotone extrapolations of order 1 and 2 , and is always feasible. Also for $d = 1$, the extrapolation $r_{k+i} = 0$ is clearly a lower envelope for all isotone extrapolations. The next proposition shows, that the lower envelope for feasible extrapolations of order $d = 2$ is along a piecewise linear function which has slope $\Delta^1 r_k$, until it reaches zero, after which it continues as a constant function zero. We define

$$p = \begin{cases} [-r_k/\Delta^1 r_k]) & \text{if } \Delta^1 r_k > 0 \\ l & \text{if } \Delta^1 r_k = 0 \end{cases} \tag{9}$$

where $[n]$ is the greatest integer not larger than n.

Proposition 1 *Consider the constraints (7) with $d = 2$ and fixed $l > 0$, and let $(r_1, \ldots, r_k)$ be a feasible solution to (7) with $l = 0$. Then the extrapolation*

$$r_{k+i} = \begin{cases} r_k + i\Delta^1 r_k & i = 1, \ldots, p \\ 0 & i = p+1, \ldots, l \end{cases} \tag{10}$$

is a lower envelope for all feasible extrapolations of order 2 to $(r_1, \ldots, r_k)$.

Proof: The solution above is clearly monotone, and $\Delta^2 r_{k+i} = 0$ for $i = 1, \ldots, p$ and $i = p+3, \ldots, l$. In addition, $\Delta^2 r_{k+p+1} = -(r_k + (p+1)\Delta^1 r_k)$ and $\Delta^2 r_{k+p+2} = r_k + p\Delta^1 r_k$, which, by definition of p are both nonnegative. Thus the constraints of (7) for $d = 2$ are satisfied. Note also, that for any other feasible extrapolation $(\bar{r}_{k+1}, \ldots, \bar{r}_{k+l})$ we have

$$\Delta^1 \bar{r}_{k+j} \geq \Delta^1 r_k.$$

Thus, if $i \leq p$ then

$$\bar{r}_{k+i} = r_k + \sum_{j=1}^{i} \Delta^1 \bar{r}_{k+j} \geq r_k + i\Delta^1 r_k = r_{k+i}.$$

It follows that (10) is a lower envelope as proposed.

4

Proposition 2 *Consider the constraints (7) with $d = 3$ and fixed $l > 0$. A solution $(r_1, \ldots, r_k)$ which satisfies (7) with $l = 0$ can be extrapolated to a vector $(r_1, \ldots, r_{k+l})$ which satisfies (7) with $l > 0$ if and only if*

$$r_k + j\Delta^1 r_k + \frac{1}{2}j(j+1)\Delta^2 r_k \geq 0; \quad j = 1, \ldots, l. \tag{11}$$

In addition, let

$$q = \begin{cases} [-\Delta^1 r_k / \Delta^2 r_k] & \text{if } \Delta^2 r_k > 0 \\ l & \text{if } \Delta^2 r_k = 0 \end{cases}$$

Then the upper envelope of all feasible extrapolations for $d = 3$ is given by

$$r_{k+i} = \begin{cases} r_k + i\Delta^1 r_k + \frac{1}{2}i(i+1)\Delta^2 r_k & i = 1, \ldots, q \\ r_{k+q} & i = q+1, \ldots, l. \end{cases}$$

Proof: First we note, that any feasible extrapolation satisfies

$$
\begin{aligned}
r_{k+i} &= r_k + \sum_{j=1}^{i} \Delta^1 r_{k+j} \\
&= r_k + \sum_{j=1}^{i}\left(\Delta^1 r_k + \sum_{h=1}^{j} \Delta^2 r_{k+h}\right) \\
&= r_k + i\Delta^1 r_k + \sum_{j=1}^{i}\sum_{h=1}^{j} \Delta^2 r_{k+h} \\
&\leq r_k + i\Delta^1 r_k + \frac{1}{2}i(i+1)\Delta^2 r_k
\end{aligned}
$$

and the nonnegativity of r_{k+i} implies that (11) must hold. Conversely, assume that (11) holds. Now since $\{r_i\}$ is completely monotone of order 3, the sequence $\{-\Delta^1 r_i\}$ is completely monotone with order $d = 2$. Using Proposition 1 for the lowest feasible convex extrapolation for $\{-\Delta^1 r_i\}$, we obtain the upper envelope for completely monotone extrapolations of $(r_1, \ldots, r_k)$ of order 3.

Proposition 3 *Consider the constraints (7) with $d = 3$ and fixed $l > 0$, and let $(r_1, \ldots, r_k)$ be a solution to (7) with $l = 0$ satisfying (11). Let p be defined as in (9).*
(a). If $p \geq l$, then the extrapolation

$$r_{k+i} = r_k + i\Delta^1 r_k, \quad i = 1, \ldots, p$$

is a lower envelope for all feasible extrapolations of order 3 to $(r_1, \ldots, r_k)$.
(b). If $p \leq l$, let

$$u = \min(l, 1 + [-2r_k/\Delta^1 r_k]).$$

Then the extrapolation

$$r_{k+i} = \begin{cases} r_k + i\Delta^1 r_k + \frac{1}{2}i(i+1)\left(\frac{-2(r_k + u\Delta^1 r_k)}{u(u+1)}\right) & i = 1, \ldots, u \\ 0 & i = u+1, \ldots, l \end{cases} \tag{12}$$

is a lower envelope for all feasible extrapolations of order 3 to $(r_1, \ldots, r_k)$.

The proposition states that the lowest envelope is a linear function with slope $\Delta^1 r_k$, provided that such a linear function is feasible (nonnegative); otherwise it starts as a quadratic function with constant second order difference

$$a = -\left(\frac{2(r_k + u\Delta^1 r_k)}{u(u+1)}\right)$$

which flattens to zero at r_{k+u}, and from there continues as zero.

Proof: If $p \geq l$ then (10) is a feasible extrapolation of order 3, thus (a) follows from Proposition 1.
If $p \leq l$ then (10) will not satisfy the third order difference constraints. We now show that for this case, the function of (12) is indeed a lower envelope for any feasible extrapolation. First assume that there exists a feasible extrapolation $\bar{r}_{k+1} \ldots \bar{r}_{k+l}$ for which $\Delta^2 \bar{r}_{k+1} < a$. Then

$$
\begin{aligned}
\bar{r}_{k+1} + (u-1)\Delta^1 \bar{r}_{k+1} + \tfrac{1}{2}(u-1)u\Delta^2 \bar{r}_{k+1} &\leq r_k + u\Delta^1 r_k + \tfrac{1}{2}u(u+1)\Delta^2 \bar{r}_{k+1} \\
&= r_k + u\Delta^1 r_k - (r_k + u\Delta^1 r_k) = 0,
\end{aligned}
$$

in contradiction to the conditions given by Proposition 2, for a feasible extrapolation for $r_1, \ldots, r_k, \bar{r}_{k+1}$. We therefore conclude that any feasible extrapolation has a second order difference of at least a. If, on the other hand, $\Delta^2 \bar{r}_{k+1} > a$, then $\bar{r}_{k+1} > r_{k+1}$. An inductive argument starting from r_{k+1} completes the proof.

Proposition 4 *Consider the constraints (7) with $d = 4$ and fixed $l > 0$. A solution $(r_1, \ldots, r_k)$ which satisfies (7) with $l = 0$ can be extrapolated to a vector $(r_1, \ldots, r_{k+l})$ which satisfies (7) with $l > 0$ if and only if*

$$
\Delta^1 r_k + j\Delta^2 r_k + \frac{1}{2}j(j+1)\Delta^3 r_k \leq 0, \quad j = 1, \ldots, l \tag{13}
$$

$$
r_k + l\Delta^1 r_k + \frac{1}{2}l(l+1)\Delta^2 r_k \geq 0, \tag{14}
$$

$$
r_k + \frac{2}{3}(j-1)\Delta^1 r_k + \frac{1}{6}j(j-1)\Delta^2 r_k \geq 0, \quad j = 1, \ldots, l. \tag{15}
$$

If $\Delta^1 r_k + l\Delta^2 r_k \leq 0$ then the upper envelope of all such extrapolations is given by

$$
r_{k+i} = r_k + i\Delta^1 r_k + \frac{1}{2}i(i+1)\Delta^2 r_k, \quad i = 1, \ldots, p. \tag{16}
$$

Otherwise, let

$$
v = \min(l, 1 + [-2\Delta^1 r_k / \Delta^2 r_k]).
$$

Then the upper envelope of all such extrapolations is given by

$$
r_{k+i} = \begin{cases} r_k + i\Delta^1 r_k + \frac{1}{2}i(i+1)\Delta^2 r_k + \frac{1}{6}i(i+1)(i+2)\left(\frac{-2(\Delta^1 r_k + v\Delta^2 r_k)}{v(v+1)}\right) & i = 1, \ldots, v \\ r_{k+v} & i = v+1, \ldots, l. \end{cases} \tag{17}
$$

Proof: If the sequence $\{r_{k+i}\}$ is a feasible extrapolation of order $d = 4$ then the sequence $\{-\Delta^1 r_{k+i}\}$ is a feasible extrapolation of order $d = 3$. By Proposition 2, the conditions for existence of the latter are given by (13). In addition, the upper envelope of all extrapolations for $d = 4$ is the sequence $\{r_{k+i}\}$ for which $\{-\Delta^1 r_{k+i}\}$ constitutes the lower envelope of all extrapolations of order $d = 3$. Applying Proposition 3 with respect to the sequence $\{-\Delta^1 r_{k+i}\}$ and integrating over this lower envelope yields the sequence of (16) and (17). Note that by construction, the resulting sequence is nonincreasing, convex with nonpositive third order difference. It remains to determine the conditions under which this sequence is nonnegative. First, we note that condition (14) guarantees that (16) will be nonnegative. From Proposition 2 this is also a necessary condition. Also conditions (15) guarantee that r_{k+v} is nonnegative for any possible value of v between 1 and l. Since (17) represents a decreasing function which becomes constant for $i \geq v$, this guarantees that r_{k+i} is also nonnegative for any i. To show that conditions (15) are also necessary, define

$$
P(j) = r_k + \frac{2}{3}(j-1)\Delta^1 r_k + \frac{1}{6}j(j-1)\Delta^2 r_k.
$$

It is easy to see that $P(j)$ decreases for $j = 1, \ldots, v$ and increases for $j = v, \ldots, l$. Suppose that (15) is violated for some j. Let $\bar{j}$ be the smallest index to violate this condition. It follows that $\bar{j} \leq v$ and that $P(v) \leq 0$. This in turns implies that $r_{k+v} \leq 0$, and thus no feasible extrapolation with $d = 4$ is feasible, hence a contradiction. This completes the proof.

We shall now derive the envelopes for prediction for the mean function. Consider a sequence of order d: $M = (m_1, \ldots, m_k)$ which satisfies (8). The sequence $(m_{k+1}, \ldots, m_{k+l})$ is defined to be a feasible extrapolation of order d for M, if the sequence $(m_1, \ldots, m_{k+l})$ satisfies (8). In addition, this extrapolation constitutes

an upper bound for all feasible extrapolations of order d, if any other such extrapolation $(\bar{m}_{k+1}, \ldots, \bar{m}_{k+l})$ satisfies $\bar{m}_{k+i} \leq m_{k+i}$ for $i = 1, \ldots, l$. Similarly it constitutes a lower envelope if $\bar{m}_{k+i} \geq m_{k+i}$ for $i = 1 \ldots, l$.

In the following, we derive conditions for the existence for such higher and lower envelopes for the feasible extrapolations for M. We note that the derivative of the mean function is a completely monotone function. Therefore, the lower and upper bounds for all feasible extrapolations of order d to $m_1, \ldots, m_k$ are obtained by integrating respectively over the lower and upper bounds for all feasible extrapolations of order $d - 1$ to $\Delta m_1, \ldots, \Delta m_k$.

Consequently, for the case $d = 1$, $d = 2$ and $d = 3$, the sequence $(m_1, \ldots, m_k)$ can always be extrapolated into the future. The upper envelope for all feasible extrapolations of order up to 3 is the linear function

$$m_{k+i} = m_k + i\Delta^1 m_k.$$

For $d = 1$ and $d = 2$ the extrapolation $m_{k+i} = m_k$ is clearly a lower envelope for all feasible extrapolations. The next proposition shows, that the lower envelope for feasible extrapolations of order $d = 3$ is along a quadratic which tapers off to a constant function.

Proposition 5 *Consider the constraints (8) with $d = 3$ and fixed $l > 0$, and let $(r_1, \ldots, r_k)$ be a feasible solution to (8) with $l = 0$. Let*

$$p = \begin{cases} [-\Delta^1 m_k / \Delta^2 m_k] & \text{if } \Delta^2 m_k > 0 \\ l & \text{if } \Delta^2 m_k = 0 \end{cases} \tag{18}$$

Then the extrapolation

$$m_{k+1} = \begin{cases} m_k + i\Delta^1 m_k + \frac{1}{2}i(i+1)\Delta^2 m_k & i = 1, \ldots, p \\ m_{k+p} & i = p+1, \ldots, l \end{cases}$$

is a lower envelope for all feasible extrapolations of order 3 to $(m_1, \ldots, m_k)$.

Proof: Follows from Proposition 1.

Proposition 6 *Consider the constraints (8) with $d = 4$ and fixed $l > 0$. A solution $(m_1, \ldots, m_k)$ which satisfies (8) with $l = 0$ can be extrapolated to a vector $(m_1, \ldots, m_{k+l})$ which satisfies (8) with $l > 0$ if and only if*

$$\Delta^1 m_k + j\Delta^2 m_k + \frac{1}{2}j(j+1)\Delta^3 m_k \geq 0, \quad j = 1, \ldots, l. \tag{19}$$

Let

$$q = \begin{cases} [-\Delta^2 m_k / \Delta^3 m_k] & \text{if } \Delta^3 m_k > 0 \\ l & \text{if } \Delta^3 m_k = 0 \end{cases}$$

and let

$$\alpha = \Delta^1 m_k + q\Delta^2 m_k + \frac{1}{2}q(q+1)\Delta^3 m_k$$

Then the upper envelope of all such extrapolations is given by

$$m_{k+i} = \begin{cases} m_k + i\Delta^1 m_k + \frac{1}{2}i(i+1)\Delta^2 m_k + \frac{1}{6}i(i+1)(i+2)\Delta^3 m_k & i = 1, \ldots, q \\ m_{k+q} + (i-q)\alpha & i = q+1, \ldots, l. \end{cases}$$

Proof: Follows from proposition 2.

Proposition 7 *Consider the constraints (8) with $d = 4$ and fixed $l > 0$, and let $(m_1, \ldots, m_k)$ be a solution to (8) with $l = 0$ satisfying (19). Let p be defined as in (18).*
(a). If $p \geq l$, then the extrapolation

$$m_{k+i} = m_k + i\Delta^1 m_k + \frac{1}{2}i(i+1)\Delta^2 m_k, \quad i = 1, \ldots, p$$

is a lower envelope for all feasible extrapolations of order 4 to $(m_1, \ldots, m_k)$.
(b). If $p \leq l$, let

$$u = \min(l, 1 + [-2\Delta^1 m_k / \Delta^2 m_k]).$$

Then the extrapolation

$$m_{k+i} = \begin{cases} m_k + i\Delta^1 m_k + \frac{1}{2}i(i+1)\Delta^2 m_k + \frac{1}{6}i(i+1)(i+2)\left(\frac{-2(\Delta^1 m_k + u\Delta^2 m_k)}{u(u+1)}\right), & i = 1, \dots, u \\ m_{k+u} & i = u+1, \dots, l \end{cases}$$

is a lower envelope for all feasible extrapolations of order 4 to $(m_1, \dots, m_k)$.

The proposition states that the lowest envelope is either along a quadratic function, or it starts as a cubic function which tapers off to a constant function. Its proof follows from Proposition 3.

4 Monte Carlo Study of Performance

To get an idea of how well future prediction envelopes estimate future behavior, we conducted a small Monte Carlo simulation experiment. Our goal is to estimate the number of events over some finite horizon. As in Miller and Sofer (1986a), we compare the completely monotone approach to some of the more popular parametric models. A value of $d = 4$ is used for the completely monotone estimation (δ is taken as 1). Thus the least squares problem (8) is solved for $d = 4$, with the constraints of (19) replacing the constraints of (8) for $i = k+1, \dots, k+l$. Propositions 6 and 7 are applied to the resulting solution to obtain the upper and lower envelopes for the future mean function. Finally, we need a point estimate of the expected number of failures. We have arbitrarily decided to use the midpoint of the envelope.

Our choice of parameter models consists of three families of nonhomogeneous Poisson processes (NHPP). The mean functions of the NHPP's may have exponential, power or logarithmic form:

$$\begin{aligned} M_{exp}(t) &= \gamma(1 - e^{-\eta t}), \\ M_{pow}(t) &= \gamma t^\alpha, \\ M_{log}(t) &= \gamma \log(\beta t + 1). \end{aligned}$$

Those models are fit to data by using the method of maximum likelihood as described by Musa and Okumoto(1984). Furthermore, we define a fourth model which is a mixture of the above three. It is fit by selecting the best fitting (i.e., maximum likelihood) of the three models. This is the "best" parametric model, among the three possibilities.

We draw our data from 16 different Poisson processes. Each process is observed over the interval [0,100] and the future interval is [100,125], i.e., 25% into the future. We used $k = 20$ and $l = 5$. The 16 cases provide a variety of different growth patterns. Each case is replicated 400 times. The cases are summarized in Table 1.

The performance of the parametric models and the completely monotone approach are summarized in Tables 2, 3 and 4. Table 2 shows the average prediction made by each model for the 400 replicates of each case. Table 3 shows the average percentage error, or bias. Table 4 shows the root-mean-square percentage error for the 400 estimates made by each model for the 16 test cases. We note that when the data comes from a certain model, then that particular model gives the best predictions. However in most cases, the completely monotone comes in as "second best", i.e., it gives better predictions than those given by using the incorrect parametric model. In practice, of course, it is highly unlikely that a parametric model used for prediction will indeed be the "correct" model from which the failure data was generated. In Table 5 the performance of the prediction envelopes is summarized. Note that the majority of the envelopes have zero width, i.e., the upper envelope is identical to the lower envelope.

In conclusion, we stress that some components in the formulation of the completely monotone model were chosen arbitrarily. Other definitions of the raw estimates and other objective functions will give different, and possibly better estimates. Nevertheless, the completely monotone approach shows a robustness not exhibited by the individual parametric models. The procedure has quite low bias, which is less than that caused by using the incorrect parametric models for prediction. Comparisons to the "best" parametric model are unfair because the Monte Carlo data is, in effect, drawn from that model. We could use other models to generate data for which this "best" parametric model is inferior to the more robust completely monotone approach.

Model Number	Type of NHPP	Parameter	M(125) - M(100)
1	Homogeneous		10.00
2	Power	$\alpha = .749$	7.28
3	Power	$\alpha = .557$	5.29
4	Power	$\alpha = .410$	3.83
5	Power	$\alpha = .296$	2.73
6	Power	$\alpha = .208$	1.90
7	Logarithmic	$\beta = .0124$	6.42
8	Logarithmic	$\beta = .0429$	4.43
9	Logarithmic	$\beta = .131$	3.16
10	Logarithmic	$\beta = .461$	2.27
11	Logarithmic	$\beta = 2.43$	1.62
12	Exponential	$\eta = .00808$	5.88
13	Exponential	$\eta = .0167$	3.17
14	Exponential	$\eta = .0265$	1.47
15	Exponential	$\eta = .0385$	0.54
16	Exponential	$\eta = .0550$	0.12

Table 1: Data Models (Poisson Processes). All models are scaled so that $E(N(100)) = M(100) = 40$.

Model Number	True Mean	EXP	LOG	POW	BEST	CM. Mdpt.
1	10.00	8.67	8.86	9.33	8.73	9.52
2	7.28	5.62	6.11	7.34	6.38	7.72
3	5.29	2.97	3.73	5.36	4.70	5.88
4	3.83	1.36	2.23	3.88	3.62	4.50
5	2.73	0.51	1.31	2.76	2.65	3.40
6	1.90	0.15	0.75	1.92	1.87	2.52
7	6.42	6.10	6.76	8.20	6.48	7.44
8	4.43	3.41	4.68	6.63	4.19	5.38
9	3.16	1.64	3.29	5.27	2.89	4.06
10	2.27	0.64	2.35	4.11	2.28	3.08
11	1.62	0.18	1.66	3.10	1.71	2.30
12	5.88	5.86	6.57	8.09	6.26	7.31
13	3.17	3.21	4.70	6.72	3.67	4.73
14	1.47	1.51	3.62	5.63	1.91	2.84
15	0.54	0.57	2.95	4.77	0.75	1.60
16	0.12	0.14	2.48	4.07	0.21	0.88

Table 2: Average Predictions of Mean Number over Future Horizon

Model	Fitted Model				
Number	EXP	LOG	POW	BEST	CM
1	-13.	-11.	-7.	-13.	-5.
2	-23.	-16.	+1.	-12.	+6.
3	-44.	-30.	+1.	-11.	+11.
4	-65.	-42.	+1.	-6.	+17.
5	-81.	-52.	+1.	-3.	+24.
6	-92.	-61.	+1.	-2.	+33.
7	-5.	+5.	+28.	+1.	+16.
8	-23.	+5.	+50.	-6.	+21.
9	-48.	+4.	+67.	-8.	+29.
10	-72.	+3.	+81.	0.	+36.
11	-89.	+3.	+98.	+6.	+42.
12	0.	+12.	+32.	+6.	+24.
13	+1.	+48.	+112.	+16.	+49.
14	+3.	+146.	+282.	+30.	+93.
15	+6.	+448.	+788.	+40.	+199.
16	+14.	+1921.	+3216.	+74.	+615.

Table 3: % Prediction Error (Bias) for Mean Future Number

Model	Fitted Model				
Number	EXP	LOG	POW	BEST	CM
1	26.	24.	29.	24.	29.
2	39.	32.	23.	31.	30.
3	53.	39.	23.	32.	39.
4	69.	46.	23.	29.	48.
5	83.	55.	23.	26.	60.
6	93.	62.	23.	26.	73.
7	37.	32.	39.	36.	38.
8	44.	31.	59.	43.	50.
9	57.	26.	75.	47.	62.
10	75.	23.	89.	40.	74.
11	90.	21.	99.	27.	85.
12	38.	34.	47.	37.	45.
13	45.	61.	120.	53.	80.
14	54.	155.	292.	82.	142.
15	67.	461.	805.	134.	278.
16	95.	1956.	3268.	336.	776.

Table 4: % Root Mean Square Error for Mean Future Number Prediction

| Data | Fraction | Fraction | Av.width | True Mean Coverage | | |
| Model | Zero | Non-zero | Non-zero | Fraction | Fraction | Fraction |
Number	Width	Width	Envelope	Overestimate	Correct	Underestimate
1	.715	.285	0.348	.408	.067	.525
2	.515	.485	0.816	.575	.105	.320
3	.503	.497	0.966	.542	.182	.275
4	.548	.452	0.759	.550	.167	.283
5	.570	.430	0.588	.585	.160	.255
6	.600	.400	0.414	.595	.155	.250
7	.420	.580	1.116	.610	.193	.197
8	.417	.583	1.334	.515	.283	.202
9	.505	.495	0.897	.560	.243	.197
10	.573	.427	0.616	.600	.193	.208
11	.573	.427	0.441	.605	.182	.213
12	.363	.637	1.228	.648	.190	.162
13	.305	.695	1.828	.520	.400	.080
14	.321	.679	1.300	.574	.333	.093
15	.503	.497	0.656	.652	.243	.105
16	.698	.302	0.209	.925	.063	.013

Table 5:   Performance of Completely Monotone Prediction Windows

5 Bibliography

1. Abdel-Ghaly, A.A., Chan, P.Y. and Littlewood B. (1986). Evaluation of competing software reliability predictions. *IEEE Trans. Software Eng.*, SE-12, 950-967.

2. Barlow, R.E., Bartholomew, D.J., Bremner, J.M. and Brunk, H.D. (1972). *Statistical Inference Under Order Restrictions*. Wiley, New York.

3. Campbell, G. and Ott, K.O. (1979). Statistical evaluation of major human errors during the development of new technological systems. *Nuclear Science and Engineering*, 71, 267-279.

4. Chan, P.Y. (1986). Software Reliability Prediction. Ph.D. Thesis, Department of Mathematics, The City University, London.

5. Crow, L.H. (1974). Reliability analyses for complex repairable systems. *Reliability and Biometry* (F. Proschan and R.J. Serfling, eds.), SIAM, Philadelphia, 379-410.

6. Duane, J. T. (1964). Learning curve approach to reliability monitoring. *IEEE Trans. Aerospace*, 2, 563-566.

7. Goel, A.K. and Okumoto, K. (1979). Time independent error detection rate model for software reliability and other performance measures. *IEEE Trans. Rel.*, R-28, 206-211.

8. Jelinski, Z. and Moranda P. (1972). Software reliability research. *Statistical Computer Performance Evaluation* (W. Ferberger, ed.) Academic Press, New York, 465-484.

9. Littlewood, B. (1981). Software reliability growth: A model for fault removal in computer-programs and hardware-design. *IEEE Trans. Rel.*, R-30, 313-320.

10. Littlewood, B. and Keiller, P.A. (1984). Adaptive software reliability modelling. *Proceedings of the 14th International Conference on Fault-Tolerant Computing*. IEEE Computer Society Press, Washington D.C., 108-113.

11. McCormick, G.P. (1983). *Nonlinear Programming*. John Wiley and Sons.

12. Miller, D.R. (1986). Exponential order statistics models for software reliability growth. *IEEE Trans. Software Eng.*, SE-12, 12-24.

13. Miller, D.R. and Sofer, A. (1985). Completely monotone regression estimates of software failure rates. *Proceedings of Eighth International Conference on Software Engineering.* IEEE Computer Society Press, Washington, D.C., 343-348.

14. Miller, D.R. and Sofer, A. (1986a). A nonparametric approach to software reliability, using complete monotonicity. *Software Reliability: A State of the Art Report*, (A. Bendall and P. Mellor eds.), Pergammon, London, 183-195

15. Miller, D.R. and Sofer, A. (1986b). Least squares regression under convexity and higher order difference constraints with application to software reliability. *Advances in Order Restricted Inference* (R. Dykstra, T. Robertson and F.T. Wright, eds.). Springer Verlag, 91-124.

16. Musa, J.D. and Okumoto, K. (1984). A logarithmic Poisson execution time model for software reliability measurement. *Proceedings of Seventh International Conference on Software Engineering.* IEEE, New York, 230-238.

17. Nagel, P.M., F.W. Scholz and J. A. Skrivan. Software reliability: Additional investigations into modeling with replicated experiments, CR-172378, NASA, 1984.