An Implicit Expectation Conditional Maximization Algorithm for Non-homogeneous Poisson Process Software Reliability Models

Vidhyashree Nagaraju* Lance Fiondella

Abstract
Software reliability growth models (SRGM) based on the non-homogeneous Poisson process (NHPP) are a popular approach to estimate useful metrics such as the number of faults remaining, failure rate, and reliability, which is defined as the probability of failure free operation in a specified environment for a specified period of time. However, it is often difficult to apply SRGM in practice because even relatively simple models can require numerical solution of complex systems of equations. To overcome this limitation, we propose expectation conditional maximization (ECM) algorithms for NHPP SRGM. In contrast to the expectation maximization (EM) algorithm, the ECM algorithm reduces the maximum likelihood estimation process to multiple simpler conditional maximization (CM)-steps. The advantage of these CM-steps is that they only need to consider one variable at a time, enabling implicit solutions to update rules when a close form equation is not available for a model parameter. Two variants are proposed. The first obtains CM-steps for all p parameters of an NHPP SRGM, while an alternative reduced log-likelihood approach eliminates one parameter from the maximum likelihood estimation process. We compare the performance of these two ECM variants on several data sets from the research literature. Our results indicate that the reduced log-likelihood ECM algorithm may be appropriate when the SRGM possesses a relatively simple form, but that the log-likelihood approach can outperform the reduced log-likelihood ECM algorithm when the form of the SRGM is more complex.

Key Words: Software reliability, adaptive expectation-maximization algorithm, implicit expectation maximization algorithm, non-homogeneous Poisson process, software reliability growth model.

Acronyms

EM	Expectation-maximization
ECM	Expectation conditional maximization
GO	Goel-Okumoto model
LL	Log-likelihood function
MLE	Maximum likelihood estimation
MVF	Mean value function
NHPP	Non-homogeneous Poisson process
RLL	Reduced log-likelihood function
SRGM	Software reliability growth models

Notations

$m(t)$	MVF of NHPP
$F(t)$	Cumulative distribution function of software fault detection process
$\lambda(t)$	Instantaneous failure rate
a	Number of faults to be detected given infinite testing time
b	Scale parameter of Weibull SRGM testing
c	Shape parameter of Weibull SRGM testing
$\mathbf{T}$	Vector of failure times
t_i	Time of the i^{th} failure
t_{obs}	Time at which testing stopped
n	Observed number of failures

*V. Nagaraju and L. Fiondella are with the Department of Electrical and Computer Engineering, University of Massachusetts, Dartmouth, MA, 02747 USA e-mail: {vnagaraju,lfiondella}@umassd.edu.

m	Unobserved number of failures
N	Total number of faults
Θ	Vector of model parameters
G	Sequence of CM-steps
t_n	n^{th} Observed failure
ε	Convergence constant
p	Number of model parameters

1. Introduction

Software reliability [1], commonly defined as the probability of failure free operation in a specified environment for a specified period of time, can be estimated with the assistance of software reliability growth models. SRGMs [2] are a fundamental and well-established methodology, many of which are based on the non-homogeneous Poisson process [3, 4]. These NHPP SRGM also enable the estimation of useful metrics such as prediction of the number of faults remaining, failure intensity, etc. SRGM are also used in optimization problems to determine the amount of testing required to achieve a desired level of reliability [5] and to minimize testing costs, while considering the risk of post release failures [6].

A significant challenge associated with SRGM is the complexity of estimating the parameters of a model [7] with traditional fitting procedures that perform maximum likelihood estimation (MLE) [8]. This difficulty arises because traditional numerical procedures to find the maximum likelihood estimates of a software failure data set such as the Newton-Raphson method [9] are sensitive to initial parameter estimates and can fail to converge to the MLE if the initial parameter estimates are not sufficiently close to the MLE. This sensitivity of existing model fitting procedures requires a relatively high level of experience, which can deter potential users from applying NHPP-based SRGM to quantitatively assess the reliability of their software. Given the increasing demand for reliable software, a model fitting procedure that is less sensitive to initial parameter estimates is needed so that software reliability growth models can be fit to data with relatively little or no effort. Such a procedure will simplify the application of NHPP-based SRGM and encourage their widespread use.

This paper presents Expectation Conditional Maximization algorithms [10] to identify the maximum likelihood estimates of the parameters of a NHPP SRGM. The ECM algorithm overcomes limitations of the expectation maximization algorithm [11], which often involves a complicated M-step. The ECM simplifies the maximum likelihood estimation process by reducing a p dimensional problem to p one-dimensional problems, drastically simplifying the computational burden. Two variants of the ECM are proposed. The first derives CM-steps for all parameters of an SRGM, whereas the reduced log-likelihood approach eliminates one parameter from the estimation process, reducing the number of model parameters by one. Log-likelihood and reduced log-likelihood ECM algorithms are for the Goel-Okumoto and Weibull SRGM are presented. To assess the performance of these alternative approaches, the log-likelihood and reduced log-likelihood ECM algorithms are applied to these two models. Our results indicate that the reduced log-likelihood ECM algorithm may be appropriate when the SRGM possess a relatively simple form, but that the log-likelihood approach can outperform the reduced log-likelihood ECM algorithm when the SRGM is more complex.

The remainder of the paper is organized as follows: Section 2 provides an overview of software reliability growth models. Section 3 reviews methods to estimate the parameters of an SRGM, including maximum likelihood estimation with Newton's Method, identifying initial estimates with the EM algorithm and, the ECM algorithms. Section 4 presents

ECM algorithms for NHPP SRGM. Section 5 illustrates the performance of these algorithms through numerical examples. Section 6 offers conclusions and directions for future research.

2. Software Reliability Growth Modeling

This section provides an overview of NHPP software reliability growth models, presenting some of the general theory and two popular models including the Goel-Okumoto (GO) [12] and Weibull [2] SRGM.

2.1 NHPP software reliability growth models

The nonhomogeneous Poisson process is a stochastic process [3] that counts the number of events that occur by time t. The expected value of a NHPP is characterized by the mean value function (MVF), denoted $m(t)$. The MVF can take many functional forms. In the context of software reliability, the NHPP counts the number of faults detected after the software has been tested for a given period of time. Okamura *et al.* [13] note that the MVF of several SRGM can be written as

$$m(t) = a \times F(t), \tag{1}$$

where a denotes the number of faults that would be detected if the software were tested for an arbitrarily long period of time and $F(t)$ is the CDF of a continuous probability distribution characterizing the software fault detection process.

The rate of occurrence of failures is time varying with instantaneous failure rate

$$\lambda(t) = \frac{\mathrm{d}m(t)}{\mathrm{d}t}. \tag{2}$$

2.1.1 Goel-Okumoto SRGM

The Goel-Okumoto model was originally proposed by Goel and Okumoto [14]. The MVF is

$$m(t) = a(1 - e^{-bt}), \tag{3}$$

where b is the fault detection rate.

2.1.2 Weibull SRGM

The MVF of the Weibull model [8] is

$$m(t) = a\left(1 - e^{-bt^c}\right). \tag{4}$$

Here, b and c are the scale and shape parameters respectively. Setting $c = 1$ in Equation (4) simplifies to the exponential distribution, also known as the Goel-Okumoto model [12] introduced in section (2.1.1).

3. Parameter Estimation Methods

This section describes various methods to determine the maximum likelihood estimates of the parameters of an SRGM, including Newton's method, a systematic method to identify initial parameter estimates with the expectation maximization algorithm, and the expectation conditional maximization algorithm. These methods are explained in increasing order of complexity.

3.1 Maximum likelihood estimation and Newton's method

Maximum likelihood estimation methods maximize the likelihood function, also known as the joint distribution of the failure data. Commonly, the logarithm of the likelihood function is maximized because the monotonicity of the logarithm ensures that the maximum of the log-likelihood function is equivalent to maximizing the likelihood function. Failure time data consist of a vector of individual failure times $\mathbf{T} = \langle t_1, t_2, \ldots, t_n \rangle$. The log-likelihood function of a failure times data set is

$$LL(t_i; \Theta) = -m(t_n) + \sum_{i=1}^{n} \log\left[\lambda(t_i)\right], \tag{5}$$

where Θ is the vector of model parameters and $\lambda(t_i)$ in Equation (2) is the instantaneous failure rate at time t_i. Traditionally, the MLE is found by numerically solving the following system of simultaneous equations with the Newton-Raphson method [9].

$$\frac{\partial}{\partial \Theta} LL(\Theta) = \mathbf{0}, \tag{6}$$

where **0** is a vector of zeros of length p corresponding to the number of model parameters.

The Newton-Raphson method is a numerical algorithm to identify the roots of an equation. The iterative update rule for simple models where the system of equations can be reduced to one is

$$x_{m+1} = x_m - \frac{f(x_m)}{f'(x_m)} \tag{7}$$

where $f(x_m)$ is the partial derivative of the log-likelihood function with respect to the parameter and $f'(x_m)$ is the second derivative of the log-likelihood function with respect to this parameter. However, the Newton-Raphson method may not converge when the initial estimates chosen as input are not close to the maximum.

3.2 Initial parameter estimation

Unlike the Newton-Raphson method, the EM algorithm provides a systematic method to identify initial parameter estimates for some or all parameters of a model. Let $t_1 < t_2 < \cdots < t_N$ be the fault detection times, where N is the total number of faults in the software, which is characterized by the Poisson distributed random variable with parameter $a > 0$. The log-likelihood function for SRGM of the form given in Equation (1) is

$$LL(a, \Theta) = N \log\left[a\right] - a + \sum_{i=1}^{N} \log\left[f(t_i; \Theta)\right]. \tag{8}$$

Okamura *et al.* [13] showed that for a mean value function of the form specified in Equation (1), an initial estimate of the number of faults (a) is simply the observed number of faults (n), while the remaining initial parameter estimates can be determined by maximizing the log-likelihood function of the probability density function $f(\cdot; \Theta) = 0$ and solving to obtain closed-form expressions for the parameters contained in the vector Θ.

By the first-order optimality condition, initial estimates of parameter a and the additional parameters contained in the probability distribution function $F(t; \Theta)$ are given by

$$a^{(0)} = N \tag{9}$$

and

$$\Theta^{(0)} := \sum_{i=1}^{N} \frac{\partial}{\partial \Theta} \log\left[f(t_i; \Theta)\right] = \mathbf{0}. \tag{10}$$

For example, the initial estimate of the scale parameter of the GO SRGM is obtained from Equations (3) and (6)

$$b^{(0)} = \frac{n}{\sum_{i=1}^{n} t_i} \tag{11}$$

While the initial estimate of the scale parameter of the Weibull SRGM is obtained from Equations (4) and (6)

$$b^{(0)} = \frac{n}{\sum_{i=1}^{n} t_i^c} \tag{12}$$

However, parameter c lacks a closed-form solution. Thus, no analytical expression for the initial estimate of this shape parameter exists. In the absence of a closed-form solution, one feasible alternative is to set $c = 1$, which reduces to the exponential model. Thus, Equation (12) reduces to Equation (11) but provides a feasible initial estimate.

3.3 Expectation conditional maximization algorithm

Unlike the EM algorithm which commonly requires the solution of computationally intensive expressions for complex SRGMs, the ECM algorithm [15, 11] simplifies computation by dividing a single M-step into p conditional-maximization steps, where p denotes the number of model parameters. Instead of solving a system of simultaneous equations as a single p-dimensional M-step, the ECM algorithm updates only one parameter at a time holding all others constant and thus reduces the maximum likelihood estimation process to p one-dimensional problems.

In each CM-step of the ECM algorithm, a single dimension of the parameter space is searched. This is implemented by partitioning the vector of model parameters Θ into subvectors $\langle\Theta_1, \ldots, \Theta_p\rangle$. Successive CM-steps determine $\Theta_i^{(j)}$, which is the updated value of the i^{th} parameter in the j^{th} iteration. Let

$$G = \{g_j(\Theta); j = 1, \ldots, p\} \tag{13}$$

Without loss of generality, the CM-step which updates the i^{th} parameter in the j^{th} iteration takes $\Theta^{(jp+i)} = \langle\Theta_1^{(j+1)}, \Theta_2^{(j+1)}, \ldots, \Theta_{i-1}^{(j+1)}, \Theta_i^{(j)}, \ldots, \Theta_p^{(j)}\rangle$ as input, holds all values but $\Theta_i^{(j)}$ constant, and maximizes the partial derivative of the log-likelihood function with respect to Θ_i to produce $\Theta^{jp+(i+1)}$ containing $\Theta_i^{(j+1)}$. Each CM-step improves the log-likelihood function monotonically. Thus, the ECM algorithm preserves the monotonicity property of the EM algorithm [11].

3.3.1 Pseudo code of ECM algorithm

This section provides the pseudo code steps of the Expectation Conditional Maximization Algorithm in the context of a probability distribution.

- (S.1) Specify the log-likelihood of the given distribution from the density of the distribution, Given the likelihood function [1],

$$Lik(t_i; \Theta) = \prod_{i=1}^{n} f(t_i; \Theta), \tag{14}$$

the log-likelihood function (LL) of a failure times data set is

$$LL(t_i; \Theta) = \sum_{i=1}^{n} \log\left[f(t_i; \Theta)\right]. \tag{15}$$

- (S.2) Take the partial derivative of the log-likelihood function in Equation (15) with respect to each model parameter to obtain CM-steps, equating to zero, and solve for a closed-form expression whenever possible.
- (S.3) If closed-form expressions are obtainable, then the ECM algorithm can simply cycle through p CM-steps by holding $(p-1)$ parameters constant until the improvement in the log-likelihood is small. Otherwise, when CM steps for a subset of parameters contains no closed-form, it may be necessary to solve a system simultaneous equations for this subset of CM-steps.
- (S.4) Repeat these CM-steps until convergence in the log-likelihood function is achieved and return the parameter estimates $\hat{\Theta}^{(t)}$.

To illustrate these steps, consider the Weibull distribution with pdf

$$f(t) = bct^{c-1}e^{-bt^c} \tag{16}$$

Substituting Equation (16) into Equation (15) and simplifying, the log-likelihood function obtained from step S.1 is

$$LL(c, b|Y) = n\log b + n\log c - b\sum_{i=1}^{n} t_i{}^c + (c-1)\log\left(\sum_{i=1}^{n} t_i\right) \tag{17}$$

The CM-step for b is

$$b^{''} = \frac{n}{\sum_{i=1}^{n} t_i{}^{c^{'}}} \tag{18}$$

where $b^{''}$ is the updated value of b and $c^{'}$ is the present value of c. The CM-step for c is

$$c^{''} = \frac{n}{b^{''}\sum_{i=1}^{n}\left(t_i{}^{c^{'}}\log[t_i]\right) - \sum_{i=1}^{n}\log[t_i]} \tag{19}$$

Alternate execution of Equations (18) and (19) converges to the MLE.

4. Expectation Conditional Maximization Algorithm for Nonhomogeneous Poisson Process SRGM

This section presents the steps of an ECM algorithm for NHPP SRGM, which seeks to avoid the complexity of the traditional ECM algorithm by solving the conditional maximization steps of the individual parameters numerically.

4.1 Log-likelihood ECM

This section presents the steps of an ECM algorithm for NHPP SRGM.

- (S.1) Step one uses Equation (5) to specify the log-likelihood function of a failure times NHPP SRGM.
- (S.2) Step two derives the conditional maximization step for the p parameters by computing partial derivatives

$$\frac{\partial LL}{\partial \Theta_i} = 0 \tag{20}$$

for $(1 \leq i \leq p)$.

- (S.3) Step three cycles through the p CM-steps holding the other $(p-1)$ parameters constant and then applying a numerical root finding algorithm. This cycle repeats until a convergence criterion such as

$$|LL_j - LL_{j-1}| < \varepsilon \tag{21}$$

is satisfied, where $\varepsilon > 0$ is an arbitrarily small constant. This identifies the maximum likelihood estimates $\widehat{\Theta}$.

4.2 Reduced log-likelihood ECM

This section presents the steps of an ECM algorithm for NHPP SRGM by simplifying the log-likelihood to eliminate parameter a from the CM required steps.

- (S.1) Step uses Equation (5) to specify the log-likelihood function of a failure times NHPP SRGM.

- (S.2) Step two reduces the log-likelihood function from p to $(p-1)$ parameters by differentiating the log-likelihood function with respect to a, equating the result to zero, and solving for a.

$$\frac{\partial LL}{\partial a} = 0 \tag{22}$$

When the mean value function possesses the form $a \times F(t)$

$$\hat{a} = \frac{n}{F(t_n)} \tag{23}$$

Substituting the solution of Equation (22) or Equation (23) into the log-likelihood function produces a reduced log-likelihood (RLL) function with $(p-1)$ model parameters.

- (S.3) Step three derives the conditional maximization steps for the remaining $(p-1)$ parameters by computing partial derivatives

$$\frac{\partial RLL}{\partial \Theta_i} = 0 \tag{24}$$

for $(1 \leq i \leq p-1)$.

- (S.4) Step four cycles through the $(p-1)$ CM-steps holding the other $(p-2)$ parameters constant and then applying a numerical root finding algorithm. This cycle repeats until a convergence criterion such as

$$|RLL_j - RLL_{j-1}| < \varepsilon \tag{25}$$

is satisfied, where $\varepsilon > 0$ is an arbitrarily small constant. This identifies the maximum likelihood estimates $\widehat{\Theta}/a$.

- (S.5) Step five computes the MLE of a by substituting $\widehat{\Theta}/a$ into Equation (22) or Equation (23), producing $\widehat{\Theta}$, the MLE for all p parameters of the model.

The following sections derive the equations needed to apply the log-likelihood and reduced log-likelihood ECM algorithms to the models discussed in Section 2.

4.3 Goel-Okumoto SRGM

4.3.1 Log-likelihood ECM

Applying Equation (2) to Equation (3) for the MVF of the GO SRGM provides the instantaneous failure rate

$$\lambda(t) = abe^{-bt} \tag{26}$$

The log-likelihood function of the GO SRGM is therefore

$$LL(a, b|\mathbf{T}) = -a(1 - e^{-bt_n}) + \sum_{i=1}^{n} \log(abe^{-bt_i}) \tag{27}$$

The CM-steps for $a^{''}$ and $b^{''}$ are obtained by differentiating Equation (27) with respect to model parameters a and b to produce,

$$a^{''} = \frac{n}{1 - e^{-b^{'}t_n}} \tag{28}$$

$$b^{''} = \frac{n}{a^{'}t_n e^{-b^{''}t_n} + \sum_{i=1}^{n} t_i} \tag{29}$$

Equation (28) obtains the updated parameter $a^{''}$ by holding $b^{'}$ constant. However, Equation (29) must be solved numerically to update $b^{''}$ because a closed-form solution is not available.

4.3.2 Reduced log-likelihood ECM

From the log-likelihood function given in Equation (27), the maximum likelihood estimate of parameter a is

$$\hat{a} = \frac{n}{1 - e^{-bt_n}} \tag{30}$$

Substituting Equation (30) into Equation (27) produces the reduced log-likelihood function

$$RLL(b|\mathbf{T}) = -n + \sum_{i=1}^{n} \log\left[\frac{nbe^{-bt_i}}{1 - e^{-bt_n}}\right] \tag{31}$$

Since the RLL contains only one unknown parameter, namely b, the parameter can be estimated with a single application of a numerical root finding algorithm. Thus, in cases where the RLL contains only one parameter, the ECM algorithm reduces to a simple root finding problem.

4.4 Weibull SRGM

4.4.1 Log-likelihood ECM

Applying Equation (2) to Equation (4) for the MVF of the Weibull SRGM provides the instantaneous failure rate

$$\lambda(t) = abct^{c-1}e^{-bt^c} \tag{32}$$

The log-likelihood function is

$$LL(a, b, c|\mathbf{T}) \;=\; -a\left(1 - e^{-bt_n^c}\right) + \sum_{i=1}^{n} \log\left[abct_i^{c-1}e^{-bt_i^c}\right] \tag{33}$$

The CM-steps for $a^{''}$, $b^{''}$, and $c^{''}$ are obtained by differentiating Equation (33) with respect to model parameters a, b, and c

$$a^{''} = \frac{n}{1 - e^{-b^{'} t_n^{c^{'}}}} \tag{34}$$

$$b^{''} = \frac{n}{a^{'} t_n^{c^{'}} e^{-b^{''} t_n^{c^{'}}} + \sum_{i=1}^{n} t_i^{c^{'}}} \tag{35}$$

and,

$$c^{''} = \frac{n}{a^{'} b^{'} e^{-b^{'} t_n^{c^{''}}} \log t_n t_n^{c^{''}} - \sum_{i=1}^{n} \log t_i - b^{'} \sum_{i=1}^{n} t_i^{c^{''}} \log t_i} \tag{36}$$

Parameters b and c lack closed-form solutions. Thus, numerical root finding is required to obtain $b^{''}$ and $c^{''}$

4.4.2 Reduced log-likelihood ECM

From the log-likelihood function given in Equation (33), the maximum likelihood estimate of parameter a is

$$\hat{a} = \frac{n}{1 - e^{-b t_n^c}} \tag{37}$$

Substituting Equation (37) into Equation (33) produces the reduced log-likelihood function

$$RLL(b, c|\mathbf{T}) \quad = \quad -n + \sum_{i=1}^{n} \log \left[\frac{nbct_i^{(c-1)}}{e^{bt_i^c - 1}} \right] \tag{38}$$

Differentiating Equation (38) according to Equation (24), the ECM update rules for parameters b and c are

$$b^{''} \quad = \quad \left(-n t_n^{c^{'}} e^{-b^{''} t_n^{c^{'}}} + \left(\frac{n}{b^{''}} - \sum_{i=1}^{n} t_i^{c^{'}} \right) \left(1 - e^{-b^{''} t_n^{c^{'}}} \right) \right) \Big/ \left(1 - e^{-b^{''} t_n^{c^{'}}} \right) \tag{39}$$

and

$$c^{''} \quad = \quad \left(\frac{-nb^{'} t_n^{c^{''}} \log\left[t_n\right] e^{-b^{'} t_n^{c^{''}}}}{1 - e^{-b^{'} t_n^{c^{''}}}} \right) + \frac{n}{c^{''}} + \sum_{i=1}^{n} \log\left[t_i\right] - b^{'} \sum_{i=1}^{n} t_i^{c^{''}} \log\left[t_i\right] \tag{40}$$

Note that the CM expressions given in Equations (39) and (40) can be applied in any order. Thus, it is possible to update parameter b in odd iterations and parameter c in even iterations or reverse the order of their application so that parameters c and b are updated in odd and even iterations, respectively.

5. Illustrations

This section illustrates the ECM algorithm through a series of examples. The first example applies the ECM algorithm to fit the Weibull SRGM to a historical data set. The second example compares the performance of log-likelihood and reduced log-likelihood ECM algorithms.

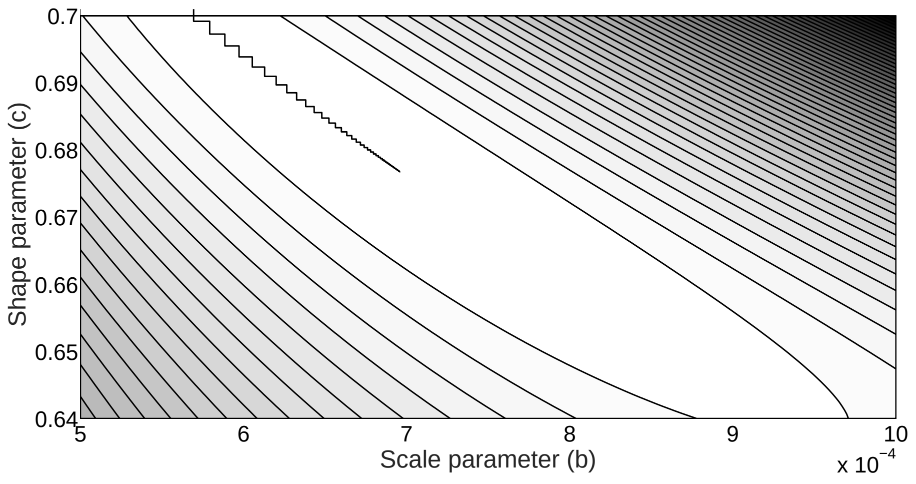

Figure 1: Iterations of ECM superimposed on contour plot of log-likelihood function

5.1 Application of ECM to NHPP SRGM

This example illustrates the steps of the ECM algorithm when the Weibull SRGM is applied to the SYS1 data set [2], which consists of $n = 136$ failure times. Since The EM algorithm lacks a closed-form expression for the initial value of c, we obtain an initial estimate based on the simpler exponential model according to Equation (11) by setting $c^{(0)} = 1.0$, providing the initial value of $b^{(0)} = 0.0000404$. Since the exponential model is a special case of the Weibull, this strategy of starting from $c^{(0)} = 1.0$ should perform well in practice because the EM algorithm for this simpler model often converges despite perturbations to the initial estimates [13]. The value of the likelihood function at these initial estimates is -975.899. The first iteration applies Equation (39), holding c constant at 1.0 and solving for $b^{(1)} = 0.000034$, increasing the log-likelihood value to -974.597. Similarly, the second iteration applies Equation (40), holding b constant at 0.000034 and solves to identify $c^{(1)} = 0.9917$, increasing the log-likelihood value to -974.172. Successive odd and even iterations update b and c, respectively. For example, iterations three and four update the parameters to $b^{(2)} = 0.0000371694$ and $c^{(2)} = 0.983453$, achieving a log-likelihood of -973.358. Thus, like the EM, the ECM improves the log-likelihood monotonically in each iteration. The implicit approach continues until the error between two successive values of the log-likelihood given in Equation (25) is less than the convergence constant $\varepsilon = 10^{-15}$. This occurs after 173 iterations, including the calculation of the initial estimate of $b^{(0)}$. The resulting maximum likelihood estimates are $\{\hat{b} = 0.000696057, \hat{c} = 0.676739\}$ and the corresponding value of the likelihood function evaluated at these estimates is -966.08033. Substituting the estimates for parameters b and c into Equation (37) produces the maximum likelihood estimate for the expected number of faults as $\hat{a} = 172.526$.

Figure 1 shows final iterations of the ECM algorithm superimposed on the contour plot of the log-likelihood function. The 90° angle movements illustrate how only one parameter is updated at a time. It can also be observed that the algorithm takes smaller and smaller steps as the parameter estimates converge to the MLE.

Figure 2 shows the monotonic improvements made by the ECM in each of the 172 iterations.

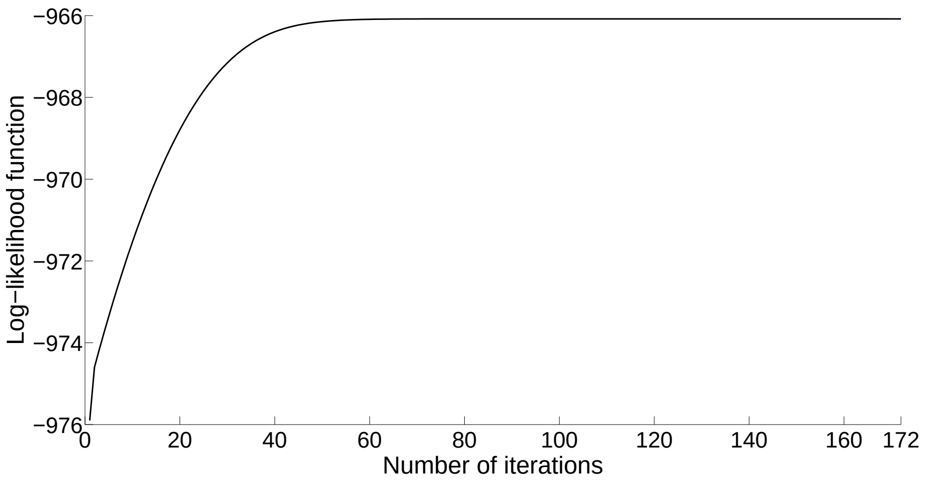

Figure 2: Monotonic improvement of log-likelihood function in iterations of ECM

5.2 Performance analysis of ECM algorithms

This example compares the performance of the log-likelihood and reduced log-likelihood ECM algorithm given in Sections 4.1 and 4.2, respectively. ECM algorithms with convergence error $\varepsilon < 10^{-10}$ for the GO and Weibull SRGM were applied to 10 failure times data sets taken from the Handbook of Software Reliability [2].

Table 1 reports the runtimes (in seconds) of the log-likelihood ECM (LL-ECM) and reduced log-likelihood ECM (RLL-ECM) algorithms given in Section 4.3 for each of the 10 failure time data sets [2] showing an average runtime of five runs. The ECM-LL runtimes were obtained with the Equations given in Section 4.3.1, while the ECM-RLL runtimes were computed using the Equations in Section 4.3.2. The LL and RLL results are shown in the second and third column respectively. The fourth column provides the ratio of LL and RLL, while the fifth column reports the p-value of a two means test for equivalence in the runtime of the two approaches.

Table 1: Comparison LL and RLL ECM algorithms for GO SRGM

Dataset	LL-ECM	RLL-ECM	LL/RLL	p-value
SYS1	0.1310	0.0624	2.1000	4.12×10^{-05}
SYS2	0.2309	0.0374	6.1667	5.25×10^{-11}
SYS3	0.5522	0.0998	5.5313	8.63×10^{-09}
S2	0.0530	0.0281	1.8889	0.002235093
S27	0.0406	0.0218	1.8571	0.000421916
SS3	1.6911	0.1217	13.8974	1.39×10^{-07}
SS4	8.1214	0.0842	96.4074	1.22×10^{-08}
CSR1	0.2683	0.1778	1.5088	3.15×10^{-06}
CSR2	0.1061	0.0624	1.7000	2.03×10^{-05}
CSR3	0.1903	0.0468	4.0667	2.24×10^{-10}

Because the ratio LL/RLL is greater than one Table 1 indicates that the runtime of the ECM algorithm with reduced log-likelihood function is that the ECM algorithm with log-likelihood for all 10 data sets considered. The p-values provided in the last column of Table 1 strongly prefer the RLL approach. Since it was noted in Section 4.3 that the reduced log-likelihood ECM algorithm for the GO SRGM reduces to a single application of a numerical root finding algorithm for parameter b in Equation (31). This indicates that Newton's method with the initial parameter estimates obtained from the EM algorithm is adequate to achieve convergence for each of the data sets considered. Thus, when the model is sufficiently simple it may be preferable to employ Newton's method with initial estimates determined by the EM algorithm.

Table 2 lists the run times (in seconds) of the log-likelihood and reduced log-likelihood ECM algorithms for the Weibull SRGM in Section 4.4. The ECM-LL runtimes were obtained with the Equations given in Section (4.4.1), while the ECM-RLL runtimes were computed using the Equations in Section (4.4.2). These algorithms were run five times with initial estimates determined from Equation (11) by setting $c^{(0)} = 1.0$ and the average runtime computed.

Table 2: Comparison LL and RLL ECM algorithms for Weibull SRGM

Dataset	LL-ECM	RLL-ECM	LL/RLL	p-value
SYS1	1.6536	1.1326	1.4601	1.06×10^{-09}
SYS2	1.9781	0.9391	2.1063	1.48×10^{-12}
SYS3	3.2916	1.3198	2.4941	1.56×10^{-07}
S2	0.8051	0.9422	0.8543	1.45×10^{-07}
S27	0.5897	0.8642	0.6823	1.32×10^{-11}
SS3	7.0856	0.5678	12.4780	8.21×10^{-17}
SS4	6.5271	1.2418	5.2563	3.16×10^{-15}
CSR1	4.9671	5.4507	0.9113	1.95×10^{-07}
CSR2	2.1185	2.6583	0.7969	1.98×10^{-06}
CSR3	0.9173	0.1903	4.8197	1.31×10^{-11}

Table 2 indicates that the reduced log-likelihood ECM algorithm outperformed the log-likelihood ECM algorithm on six out of 10 data sets where the ratio LL/RLL was greater than 1.0. However, the log-likelihood ECM algorithm outperformed the reduced log-likelihood ECM algorithm on the S2, S27, CSR1, and CSR2 data sets. The statistical significance for or against the log-likelihood or reduced log-likelihood ECM algorithm was very high. It is important to note that the reduced log-likelihood ECM algorithm outperformed the log-likelihood ECM algorithm by as much as 12.47 times, but that the reduced log-likelihood ECM algorithm never required more than 150% of the time taken by the log-likelihood ECM algorithm, since $1/0.6823 = 1.4656$. These observations suggest that the reduced log-likelihood ECM algorithm possesses value as model complexity increases because it can simplify the computation by reducing the number of model parameters to $(p - 1)$.

6. Conclusions and Future Research

This paper presents expectation conditional maximization algorithms to find maximum likelihood estimates of the parameters of a nonhomogeneous Poisson process software reliability growth model. Two variants of the ECM were proposed. The first derived CM-steps for all parameters of an SRGM, whereas the reduced log-likelihood approach eliminated the fault count parameter a, reducing the number of model parameters by one. Log-likelihood and reduced log-likelihood ECM algorithms were derived for the Goel-Okumoto and Weibull SRGM. These two variants of the algorithm were applied to both models. Ten failure times data sets from the Handbook of Software Reliability Engineering were considered. The results indicated that the reduced log-likelihood ECM algorithm performed better on the Goel-Okumoto model. For the Weibull SRGM, however, CM-steps obtained from the log-likelihood approach performed as much as 12 times faster than the reduced log-likelihood approach, but the reduced log-likelihood approach never out performed the log-likelihood approach by a factor of more than 1.5. These results suggest that the log-likelihood approach may exhibit better performance as model complexity increase.

Future research will apply the ECM algorithm to more complex models and compare the performance of the log-likelihood ECM and reduced log-likelihood ECM algorithms. This method will also be extended to failure rate and failure count models.

Acknowledgment

This work was supported by (i) the Naval Air Systems Command (NAVAIR) through the Systems Engineering Research Center (SERC), a Department of Defense (DoD) University Affiliated Research Center (UARC) under Research Task 139 : Software Reliability Modeling and (ii) the National Science Foundation (NSF) (#1526128).

References

[1] L. Leemis, *Reliability: Probabilistic Models and Statistical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[2] M. Lyu, Ed., *Handbook of Software Reliability Engineering*. New York, NY: McGraw-Hill, 1996.

[3] S. Ross, *Introduction to Probability Models*, 8th ed. New York, NY: Academic Press, 2003.

[4] M. Zhao and M. Xie, "On maximum likelihood estimation for a general non-homogeneous Poisson process," *Scandinavian Journal of Statistics*, pp. 597–607, 1996.

[5] S. Yamada, H. Ohtera, and H. Narihisa, "Software reliability growth models with testing-effort," *IEEE Transactions on Reliability*, vol. R-35, no. 1, pp. 19–23, apr 1986.

[6] K. Okumoto and A. Goel, "Optimum release time for software systems based on reliability and cost criteria," *Journal of Systems and Software*, vol. 1, pp. 315–318, 1980.

[7] S. Hossain and R. Dahiya, "Estimating the parameters of a non-homogeneous Poisson-process model for software reliability," *IEEE Transactions on Reliability*, vol. 42, no. 4, pp. 605–612, dec 1993.

[8] E. Elsayed, *Reliability Engineering*, 2nd ed. Hoboken, NJ: Wiley, 2012.

[9] R. Burden and J. Faires, *Numerical Analysis*, 8th ed. Belmont, CA: Brooks/Cole, 2004.

[10] X. Meng and D. Rubin, "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.

[11] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, no. 1, pp. 1–38, jan 1977.

[12] A. Goel, “Software reliability models: Assumptions, limitations, and applicability,” *IEEE Transactions on Software Engineering*, no. 12, pp. 1411–1423, 1985.

[13] H. Okamura, Y. Watanabe, and T. Dohi, “An iterative scheme for maximum likelihood estimation in software reliability modeling,” in *International Symposium on Software Reliability Engineering*, Denver, CO, nov 2003, pp. 246–256.

[14] A. Goel and K. Okumoto, “Time-dependent error-detection rate model for software reliability and other performance measures,” *IEEE Transactions on Reliability*, vol. 28, no. 3, pp. 206–211, 1979.

[15] D. Rubin and R. Little, “Statistical analysis with missing data,” *Hoboken, NJ: J Wiley & Sons*, 2002.