ソフトウェア工学 86－9
（1992．7．21）

# 平均ソフトウェア故障発生時間に基づく信頼性予測

木村光宏，山田茂，尾崎俊治

広島大学工学部

ソフトウェア開発の最終段階であるテスト工程において，開発管理者が最も興味をもつ事柄の一つとして，「現段階においてどのくらいの信頼性が達成されているか？」ということが挙げられよう．本研究では，テスト工程において観測されるソフトウェア故障発生時間データに着目し，非同次ポアソン過程によるソフトウェア信頼度成長モデルを適用する．このモデルに基づく定量的信頼性評価尺度として，次のソフトウェア故障が発生するまでの条件付き平均時間間隔およびソフトウェア故障時間間隔の条件付きメディアンを導出し，テスト工程におけるソフトウェア信頼性予測に関する考察を実測データを用いて行う．また，両評価尺度による予測値と実測値との適合性に関する比較評価を行う．

# SOFTWARE RELIABILITY PREDICTION BASED ON MEAN TIME BETWEEN SOFTWARE FAILURES

Mitsuhiro Kimura, Shigeru Yamada, and Shunji Osaki

Department of Industrial and Systems Engineering

Faculty of Engineering, Hiroshima University

Higashi-Hiroshima, 724 Japan

One of the most important issues for a development manager may be how to predict the reliability of a software system at an arbitrary testing time. In this paper, Using the software failure occurrence time data, we discuss a method of software reliability prediction based on software reliability growth models described by an NHPP (nonhomogeneous Poisson process). From the applied software reliability growth models, the conditional probability distribution of the time between software failures is derived, and its mean and median are obtained as software reliability assessment. Finally, using several numerical examples, we compare the performance between these measures from the view point of software reliability prediction in the testing phase.

## 1．まえがき

従来より，ソフトウェアの品質特性のうち，その計量化が急務であったソフトウェア信頼性を，実際の開発環境，特にテスト工程に即して定量的に評価するために，様々なソフトウェア信頼度成長モデル（software reliability growth model）が提案されてきた[1-6]．ソフトウェアの代表的な信頼性評価尺度の一つとしては，ソフトウェア故障が発生する（またはソフトウェアエラーが発見される）までの平均時間が挙げられる[2-4]．ここで，ソフトウェア故障とは，ソフトウェア内に潜在するソフトウェアフォールトあるいはソフトウェアエラーと呼ばれる人為的誤りや欠陥により，期待どおりにソフトウェアが動作しないことと定義する．

本研究では，ソフトウェア故障発生時間間隔に対する確率分布を用いて条件付き期待値およびメディアンを導出し，これらの評価尺度を用いてテスト工程におけるソフトウェアの信頼性予測に関する考察を行う．なお，これらの評価尺度の導出にあたり，非同次ポアソン過程（nonhomogeneous Poisson process，以下ＮＨＰＰと略す）に基づくソフトウェア信頼度成長モデルを適用する[4-7]．また，実測データを用いて適用例を示し，これらの評価尺度による予測値と実測値との適合性評価を行う．

## 2．ソフトウェア信頼度成長モデル

### 2．1　仮定とモデルの記述

テスト時刻 $t$ $(t \geq 0)$ までに発生した累積ソフトウェア故障数を $N(t)$ で表すと，ソフトウェア故障発生過程のもつ確率的性質により，以下の仮定の下で計数過程 $\{N(t),\ t \geq 0\}$ をＮＨＰＰとして取り扱うことができる．

(A1) 1つのソフトウェア故障は1つのソフトウェアエラーにより引き起こされる．

(A2) ソフトウェアエラーの修正時に新たなエラーは作り込まれない．

(A3) $N(0) = 0$．

(A4) 確率過程 $\{N(t),\ t \geq 0\}$ は独立増分をもつ．

(A5) $\Pr\{N(t+\Delta t) - N(t) = 1\} = h(t)\Delta t + o(t)$．

(A6) $\Pr[N(t+\Delta t) - N(t) \geq 2] = o(t)$．

上記の仮定のうち，仮定(A3)-(A6)は，計数過程 $\{N(t),\ t \geq 0\}$ に対するＮＨＰＰの性質を表している．

これらの仮定の下で $\{N(t),\ t \geq 0\}$ は次のように記述される．

$$\Pr[N(t) = n] = \frac{\{H(t)\}^n}{n!} \exp[-H(t)] \qquad (n = 0, 1, 2, \ldots), \tag{1}$$

$$H(t) = \int_0^t h(x)dx, \tag{2}$$

を得る．ここで，式（1）の $H(t)$ はＮＨＰＰの平均値関数（mean value function）と呼ばれ，テスト時刻 $t$ までに発生した累積ソフトウェア故障数の期待値を表す．また式（2）の $h(t)$ は，ＮＨＰＰの強度関数（intensity function）と呼ばれ，時刻 $t$ におけるソフトウェア故障発生率またはエラー発見率である．このＮＨＰＰに基づくソフトウェア信頼度成長モデルは，式（2）の平均値関数に物理的な意味をもった適当な関数形を指定することにより特徴づけられる．現在までに，多くの研究者によって様々な平均値関数が提案されている．本研究では，最も基本的なモデルとして，指数形ソフトウェア信頼度成長モデル（exponential software reliability growth model）[8]および遅延Ｓ字形ソフトウェア信頼度成長モデル（delayed S-shaped software reliability growth model）[4-6]を用いることにする．

#### 指数形ソフトウェア信頼度成長モデル

いま，$a$ をテスト開始前にソフトウェア内に潜在しているエラー数の平均値，$b$ をエラー1個当たりの発見率またはソフトウェア故障発生率とすると，テスト時刻 $t$ までに発生する総期待ソフトウェア故障数 $H(t)$ に関して次の関係式を得る．

$$\frac{dH(t)}{dt} = b[a - H(t)] \ (a > 0,\ b > 0)\ . \tag{3}$$

この関係式は，テスト時刻 $t$ におけるソフトウェア故障の起こり易さがその時点でソフトウェア内に潜在しているエラー数に比例するという，最も単純な仮定に基づいたものである．式（3）を初

期条件 $H(0)=0$ の下で解くと，平均値関数および強度関数

$$H(t)=a(1-e^{-bt}) \quad h(t)=abe^{-bt}, \tag{4}$$

を得ることができる．

遅延S字形ソフトウェア信頼度成長モデル

このモデルは，ソフトウェア故障の発生後その原因となったエラーの発見が無視できるほど短い時間間隔で行われるとする指数形ソフトウェア信頼度成長モデルを発展させたものである．すなわち，比較的複雑なソフトウェアをテストする場合，ソフトウェア故障を観測した後，その原因を十分に解析しなければ最終的にエラーの認知・修正が行えないことが多く，これを記述しようとするモデルである．ここで，ソフトウェア故障発生現象を観測する過程をソフトウェア故障発見（failure detection）過程と呼び，その原因解析を行ってエラーの発見・修正に至るまでの過程はエラー認知（error isolation）過程と呼んでいる．まず，ソフトウェア故障発見過程において，テスト時刻 $t$ までに発生する総期待ソフトウェア故障数を $H_f(t)$ とすると次の関係式，

$$\frac{dH_f(t)}{dt}=b_f[a-H_f(t)] \quad (a>0,\ b_f>0), \tag{5}$$

が成立する．ここで，$a$ はテスト開始前にソフトウェア内に潜在しているエラー数の平均値，$b_f$ はエラー1個当たりのソフトウェア故障発見率である．また，これに引き続くエラー認知過程において，テスト時刻 $t$ までに発見されるエラーの総期待数を $H(t)$ とすると，

$$\frac{dH(t)}{dt}=b_e[H_f(t)-H(t)] \quad (b_e>0), \tag{6}$$

が成立する．したがって，$b=b_e=b_f$ が近似的に成り立つとして，式（5）および式（6）を初期条件 $H(0)=0$ のもとで $H(t)$ に関して解くと，平均値関数および強度関数

$$\left.\begin{aligned} H(t)&=a[1-(1+bt)\exp[-bt]] \\ h(t)&=ab^2te^{-bt} \end{aligned}\right\}, \tag{7}$$

を得る．

## 2．2　モデルパラメータの推定

ここでは，テスト工程において得られたソフトウェア故障発生時刻あるいはエラー発見時刻に関する観測データに対して，前述のNHPPに基づくソフトウェア信頼度成長モデルを適用する場合の最尤法による未知パラメータの推定方法について述べる．

いま，テスト時刻 $s_k$ のとき $k$ 番目のソフトウェア故障が発生し，その時点までのソフトウェア故障発生時刻に関するデータ $(s_1,s_2,\ldots,s_k)(0\le s_1\le s_2\le\cdots\le s_k)$ を利用できるものとする．NHPPモデルのモデルパラメータに関する尤度関数 $L$ は，

$$\begin{aligned} L&=\prod_{i=1}^{k}h(s_i)\cdot\exp[-\int_{s_{i-1}}^{s_i}h(x)dx] \\ &=\exp[-H(s_k)]\prod_{i=1}^{k}h(s_i), \end{aligned} \tag{8}$$

により与えられる[4]．ここで，$s_0\equiv 0$ とする．式（8）の両辺の自然対数をとると，

$$\ln L=\sum_{i=1}^{k}\ln h(s_i)-H(s_k), \tag{9}$$

となる．

具体的に，式（4）の平均値関数をもつ指数形ソフトウェア信頼度成長モデルの場合，推定すべき未知パラメータは $a$ および $b$ である．したがって，パラメータの最尤推定値 $\hat{a}$ および $\hat{b}$ を求めるために，$a$ および $b$ について式（9）を偏微分して

$$\frac{\partial\ln L}{\partial a}=\frac{\partial\ln L}{\partial b}=0, \tag{10}$$

と置いて整理すれば，

$$a=\frac{k}{(1-e^{-bs_k})}, \tag{11}$$

$$\frac{k}{b}=\sum_{i=1}^{k}s_i+\frac{ks_ke^{-bs_k}}{(1-e^{-bs_k})}, \tag{12}$$

となる．これらの同時尤度方程式を数値的に解くことにより $a$ および $b$ の最尤推定値 $\hat{a}$ および $\hat{b}$ を得る．同様にして，式（7）の平均値関数をもつ遅延S字形ソフトウェア信頼度成長モデルの未知パラメータ $a$ および $b$ の最尤推定値 $a$ および $b$ は，同時尤度方程式

$$a=\frac{k}{[1-(1+bs_k)e^{-bs_k}]}, \tag{13}$$

$$\frac{2k}{b} = \sum_{i=1}^{k} s_i + \frac{kbs_k{}^2 e^{-bs_k}}{[1-(1+bs_k)e^{-bs_k}]}, \qquad (14)$$

を数値的に解いて得られる．

以下では，上記の両モデルにおける未知パラメータの最尤推定値を $\hat{a}$ および $\hat{b}$ を，$k$ 番目までのソフトウェア故障発生時刻のデータを使った最尤推定値という意味で $\hat{a}_k$ および $\hat{b}_k$ と表す．

## 3．テスト工程における信頼性予測

テスト工程におけるソフトウェア開発管理者の関心事の1つとして，「現段階においてどれぐらいの信頼性が達成されているか」ということが挙げられる．本研究では，ＮＨＰＰに基づくソフトウェア信頼度成長モデルにおける定量的な信頼性評価尺度として，次のソフトウェア故障が発生するまでの時間間隔の期待値およびメディアン（median，中央値）を取り上げる．

いま，$k$ 番目のソフトウェア故障発生時刻が $s_k$ であるとき，次の $(k+1)$ 番目のソフトウェア故障が発生するまでの時間間隔 $X_{k+1}$ の確率分布は

$$\Pr[X_{k+1} \le x \mid s_k] \equiv F_{X_{k+1}|s_k}(x)$$

$$= 1 - \exp[-\{H(s_k+x) - H(s_k)\}], \quad (15)$$

となる．この分布は，式（4）および式（7）の平均値関数 $H(t)$ が $t$ に関して有界であることから，有限な $s_k$ に対して $F_{X_{k+1}|s_k}(\infty) < 1$ となり，確率分布としての期待値などを求めることができない [9]．したがって，本研究では $s_k < +\infty$ なる条件の下での分布関数 $G_{X_{k+1}|s_k}(x)$ を考える．すなわち，

$$\Pr[X_{k+1} \le x \mid s_k < +\infty] \equiv G_{X_{k+1}|s_k}(x)$$

$$= \frac{1 - \exp[-\{H(s_k+x) - H(s_k)\}]}{1 - \exp[-n(s_k)]}, \quad (16)$$

である．ここで，$n(t)$ はテスト時刻 $t$ における期待残存エラー数であり，$H(\infty) = a$ のとき $n(t) = a - H(t)$ の関係がある．式（16）を用いて，条件付き期待値 $MTTF_{k+1}$ およびメディアン $MED_{k+1}$ は，それぞれ

$$MTTF_{k+1} = \frac{1}{(e^{n(s_k)} - 1)} \int_{s_k}^{\infty} (e^{n(x)} - 1)dx, \quad (17)$$

$$MED_{k+1} = n^{-1}[\ln(\frac{1 + e^{n(s_k)}}{2})] - s_k, \quad (18)$$

により導出できる．これらはともに，テスト時刻 $s_k$ での期待残存エラー数の関数となっている．

## 4．適用例と考察

文献 [10] に示された3種類のソフトウェア故障発生時間データ（それぞれ DS-1, DS-2, および DS-3 と呼ぶ）を用いて，信頼性予測に関する適用例を示す．これらのデータは各ソフトウェア故障の発生時刻からなり，テスト終了時点での総ソフトウェア故障数は，それぞれ 54 個，101 個，および 136 個である．

まず，DS-1 についてのデータ解析について述べる．30 番目のソフトウェア故障が発生した時点（すなわちテスト時刻 $s_{30}$）において，それまでに発生した累積ソフトウェア故障数の時間変化の様子から，ＮＨＰＰモデルとして前述の指数形ソフトウェア信頼度成長モデルを適用することが妥当であると判断した．そこで，モデルパラメータ $\hat{a}_{30}$ および $\hat{b}_{30}$ を推定すると，式（11）および式（12）からそれぞれ

$$\hat{a}_{30} = 33.172, \qquad \hat{b}_{30} = 0.000090211,$$

となった．これにより，式（17）および式（18）を用いて，31 番目のソフトウェア故障が発生するまでの時間間隔に対する予測値，すなわち $MTTF_{31}$ および $MED_{31}$ を求めると，それぞれ

$$MTTF_{31} = 4570.14, \qquad MED_{31} = 2550.96,$$

と予測される．このような推定を，逐次 $k = 54$ まで繰り返し行った場合の各評価尺度の予測値と実測値を図1に示す．また，DS-2 および DS-3 についても同様に，指数形ソフトウェア信頼度成長モデルを適用して逐次推定を行った結果を図2および図3に示した．ただし，DS-2 については $k = 70$ から評価を行い，DS-3 については $k = 35$ から行った．$X_{k+1}$ の確率密度 $g_{X_{k+1}|s_k}(x) = dG_{X_{k+1}|s_k}(x)/dx$ が右すその長い単峰形の（unimodal）分布であることから，$MTTF_{k+1}$ は常に $MED_{k+1}$ よりも大きい値をとることが読み取れる．また，実際のソフトウェア故障発生時間間隔の増減により評価尺度の予測値のばらつきは大きくなるが，メディア

ンに比べて期待値の方がばらつく度合いが大きいことも分かる．

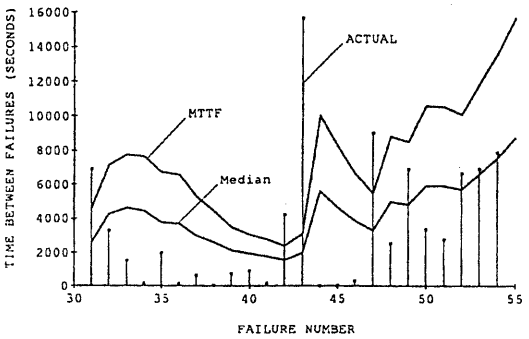


図１　ソフトウェア故障時間間隔に関する
予測値と実測値の比較（DS-1）

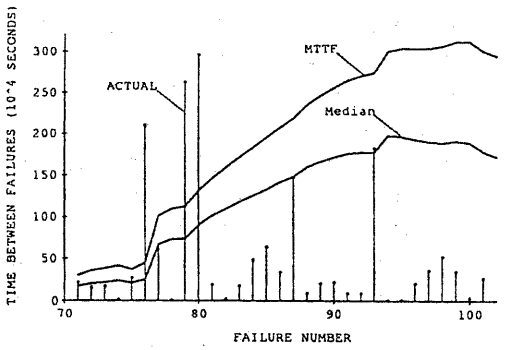


図２　ソフトウェア故障時間間隔に関する
予測値と実測値の比較（DS-2）

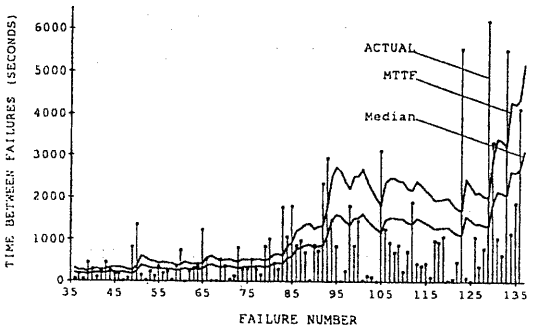


図３　ソフトウェア故障時間間隔に関する
予測値と実測値の比較（DS-3）

これらの評価尺度の推定値と実際のソフトウェア故障発生時間間隔との適合性を，偏差２乗和を用いて比較した結果を表１に示す．これにより，信頼性予測のための評価尺度としては，よく用いられる期待値よりもメディアンの方が，実測データでの適合性が高い尺度であるといえる．

一方，図２は他の２つの図とは傾向が異なっており，$k=81$ から $k=94$ までの予測値は実測値の傾向とは異なり常に増加している．このことは，図４に示す DS-2 の累積ソフトウェア故障数の挙動，すなわち信頼度成長曲線の時間的傾向から説明することができる．すなわち，81番目のソフトウェア故障が発見されたとき，推定された平均値関数はテスト時間に対してほぼ平坦となるので，すなわち推定された $H(t)$ の値がパラメータ $a$ の推定値に接近しており，その時点での期待残存エラー数は０に近い値となる．よって，式（１７）および式（１８）から明らかなように，それぞれの予測値は大きくなる．さらに，新たなソフトウェア故障が短い時間間隔で数個発生しても推定された平均値関数の挙動はほとんど変わらない（すなわち平坦である）ため，

表１　偏差２乗和による適合度比較

| | MTTF | Median |
|---|---|---|
| DS-1 | $8.4345\times10^{8}$ | $4.0015\times10^{8}$ |
| DS-2 | $1.1661\times10^{14}$ | $5.1894\times10^{13}$ |
| DS-3 | $1.7710\times10^{8}$ | $1.1324\times10^{8}$ |

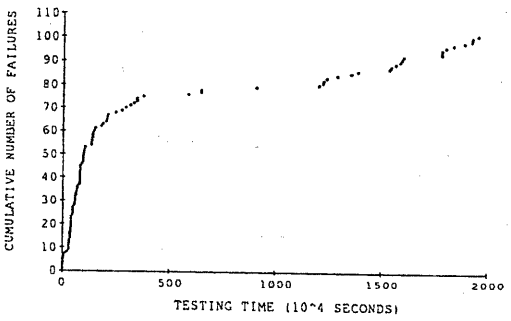


図４　累積ソフトウェア故障数のテスト時間の
経過に伴う変化（DS-2）

期待残存エラー数はさらに減少し，2つの予測値は共に増加することになる．このことから，DS-2のような傾向をもつデータの場合，本研究で述べた信頼性の予測手法はかなり楽観的な評価結果となることがわかる．

さらに，図5，図6および図7には，式（7）の平均値関数をもつ遅延S字形ソフトウェア信頼度成長モデルに対して本予測手法を適用した結果を示す．ただし，解析に用いたデータはシミュレーションにより発生させたものである（付録参照）．図5より，指数形ソフトウェア信頼度成長モデルの適用例と同様な結果が得られることがわかる．

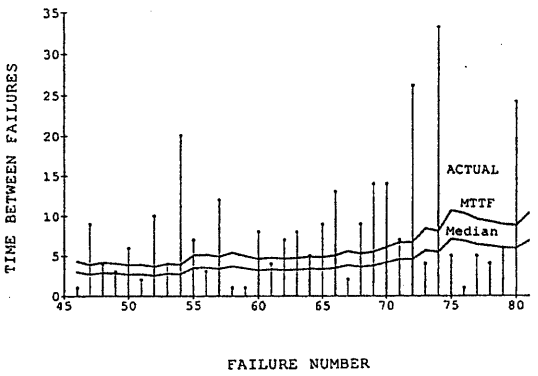


図5　ソフトウェア故障時間間隔に関する予測値と実測値の比較（シミュレーションによるデータ）

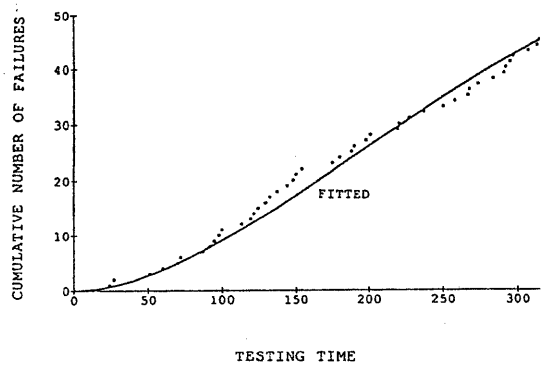


図6　$k=45$ における累積ソフトウェア故障数データとその推定結果

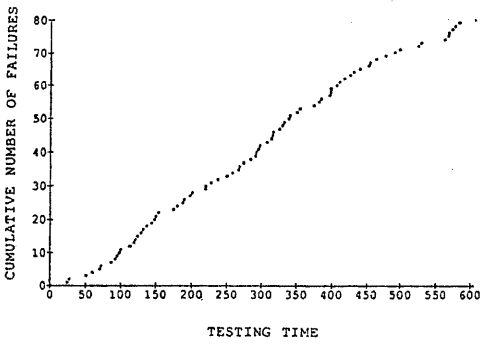


図7　シミュレーションにより作成した累積ソフトウェア故障数データ

## 5．むすび

本研究では，テスト工程においてソフトウェア故障発生時間間隔を逐次的に予測する場合の評価尺度として，その条件つき期待値およびメディアンを導出し，実際のソフトウェア故障時間データに対して適用例を示した．これにより，メディアンによる予測は期待値による場合に比べて実測値に対する適合性が良く，ソフトウェア信頼性を評価する上で悲観的な予測結果を与えることがわかった．また，観測データによる信頼度成長曲線の挙動によっては予測値が大きくずれることを示し，このようなデータを用いて信頼性評価を行う場合，評価尺度として条件付き期待値およびメディアンはかなり楽観的な評価結果を与えることがわかった．

**謝辞**

本研究の一部は，文部省科学研究費（一般研究(c) 課題番号 04650316 ）の補助を受けたことを付記する．

### 参考文献

[1] B. Littlewood and D. Miller (eds.): *Software Reliability and Safety*, Elsevier Applied Science, London (1991).

[2] Y. K. Malaiya and P. K. Srimani (eds.): *Software Reliability Models: Theoretical Development, Evaluation and Application*,

IEEE Computer Society Press, Los Alamitos (1990).

[3] J. D. Musa, A. Iannino and K. Okumoto: *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York (1987).

[4] 山田茂: "ソフトウェア信頼性評価技術", HBJ出版局, 東京 (1989).

[5] S. Yamada: "Software quality/reliability measurement and assessment:Software reliability growth models and data analysis",*J. Information Processing*, Vol. 14, No. 3, pp. 254-266 (1991).

[6] 山田茂, 大寺浩志: " ソフトウェアの信頼性〜理論と実践的応用〜", ソフト・リサーチ・センター, 東京 (1990).

[7] H. E. Asher and H. Feingold: *Repairable Systems Reliability: Modeling, Inference, Misconceptions and Their Causes*, Marcel Dekker, New York (1984).

[8] A. L. Goel and K. Okumoto: "Time-dependent error-detection rate model for software reliability and other performance measures", *IEEE Trans. Reliability*, Vol. R-28, No. 3, pp. 206-211 (1979).

[9] J. Hishitani, S. Yamada and S. Osaki: "Reliability Assessment Measures Based on Software Reliability Growth Model with Normalized Method", *J. Information Processing*, Vol. 14, No. 2, pp. 178-183 (1991).

[10] J. D. Musa: *Software Reliability Data*, Rome Air Development Center, NewYork (1979).

付録：シミュレーションデータの作成法

本研究で用いたシミュレーションデータの作成方法について述べる．

計数過程 $\{N(t),\ t \geq 0\}$ がＮＨＰＰに従っているとき，ＮＨＰＰの仮定から，

$$\Pr[N(t+\Delta t) = m+1 \mid N(t) = m]$$
$$= h(t)\Delta t \quad (m = 0,1,2,\ldots)\,, \quad (A-1)$$

である．ここで $\Delta t$ は十分に微小な時間長を表す．また，

$$t_j = j\Delta t \quad (j = 0,1,2,\ldots)$$

$$N_j = N(t_j) \quad (j = 0,1,2,\ldots)$$

と書くことにすると，微小時間区間 $\Delta t$ を適切に選んで次の手順を実行することにより，ＮＨＰＰの強度関数 $h(t)$ の特性に従うシミュレーションデータを作成することができる．

Step 1

$j = 0$, $N_0 = 0$ とする．

Step 2

$t = t_{j+1}$で $[0,1]$ の一様乱数 $\xi$ を１個発生させる．

もし，$0 \leq \xi \leq h(t)\Delta t$ ならば $N_{j+1} = N_j + 1$ とする．

もし，$h(t)\Delta t < \xi \leq 1$ ならば $N_{j+1} = N_j$ とする．

Step 3

$j = j + 1$ とする．

Step 4

$N_j$が必要な個数になるまで Step 2 に戻る．