Generating the Underlying Process of the Forward Kolmogorov Equations

Abdul – Kareem I. Sheet
Dep. of Mathematics , College of Basic Education , Mosul University, Mosul, Iraq
(Received / / 2007, Accepted / / 2008)

Abstract:
This paper deals with the numerical solution of the forward kolmogorov equations problem, in addition to, the use of the Markov Chain Monte Carlo simulation technique for generating the underlying process of these equations and to predict the sample path behavior of this process. As an illustration of the use of this approach, numerical and numerical simulation results are obtained when the underlying process is of Poisson type and Continuous Time Markov Chain type. We obvious from the results we obtained the efficiency of Monte Carlo Markov chain simulation technique to predict sample path of these processes through out extended time periods.

Introduction:
In the recent years the use of the Markov renewal equation as a nonlinear model for birth dynamics has been expanded dramatically. In mathematical demography, the Markov renewal equation has been used extensively as the vehicle for determining the future female birth trajectory in a closed population [10]. As these equations become more detailed, analytical results become very difficult to obtain, for this reason a *Markov Chain Monte Carlo* simulation for conducting experiments on a model of a real life system has become an increasingly attractive way for the study of this model equation. The goal of the *simulation* is the estimation of quantities of the dynamical system under study. In a Markov renewal equation of a pure birth process there are two important quantities:

1. The distribution of the number of births in a time interval.
2. The distribution of time until the population reaches a given size.

However, determining these quantities when the underlying stochastic process is a Markov renewal process is an unsolved problem. To solve this problem we developed the *Markov Chain Monte Carlo* (*MCMC*) method to generate the sample paths of the Markov renewal process as our first objective [8]. Section 2 gives a short outline of a Markov renewal process. Section 3 formulates a class of Markov renewal equations, known as the *forward kolmogorov equations*. The main results of this paper are presented in sections 4 and 5. Section 4 discusses the numerical solution of these equations when the underlying process is a *Poisson process* and *a Continuous time Markov chain*. Section 5 simulates the underlying stochastic process of these equations by providing methods of simulating a *Poisson process* and *a Continuous time Markov chain*.

A short outline of Markov Renewal Process:
The Poisson process has the property that the times between transitions are identically independent distributed with an exponential distribution function [11]:

$$F(t) = 1 - e^{-\lambda t} \qquad (1)$$

Continuous time Markov chain (*CTMC*) is a process that goes from state to state according to a Markov chain, and each time a state is visited the process stays a random time that is independent of the past behavior of the process and has an exponential distribution [5].

The Markov renewal process generalizes *CTMC* by allowing the time between transitions to be arbitrarily distributed nonnegative random variable which may depend on the current state and the next state during the time interval (0, t] [4].

Markov Renewal Equations:
The Markov renewal equation is the generalization of a renewal equation where the distribution function in stead of being numerical is matrix valued based on the fact that the theory of Markov renewal process generalizes renewal process and Markov chain, and is a blend of the two. Recall that the renewal process generalizes the Poisson process by allowing the distribution function *F* (*t*) to be any distribution function corresponding to a nonnegative random variable [9].

Birth Equations:
The transition function

$$P_{ij}(t) = \Pr\{X(t+s) = j : X(s) = i\} \qquad (2)$$

from state ***i*** to state ***j*** which describes the stochastic evolution of a birth process {X (t): t ≥ 0} with birth rate $\lambda \geq 0$ satisfies a system of differential equations known as the forward kolmogorov equations given by [11]:

$$P_j'(t) = \lambda P_{j-1}(t) - \lambda P_j(t) \quad ; \quad j \geq 1 \qquad (3)$$

$$P_0'(t) = -\lambda P_0(t) \qquad (4)$$

with initial conditions

$P_0(0) = 0 \quad ; \quad P_j(0) = 1 \quad for \quad j \geq 1$

where $P_j(t) = \Pr\{X(t) = j\}$ **(5)**

Continuous Time Markov Chain Equations:
Let {X(t): t≥0} be a Continuous time Markov chain with state space $E \subset \{0, 1, 2, ...\}$, recall that the transition function $P_{ij}(t)$ define by equation (2) represent the probability that a Continuous time Markov chain presently in state *i* at time *t* will be in state *j* at time (*t* + *s*). Define

$$q_j = \lim_{t \to 0} \frac{1 - P_{jj}}{t}(t), \qquad (6)$$

$$q_{kj} = \lim_{t \to 0} \frac{P_{kj}}{t}(t); \quad k \neq j \qquad (7)$$

According to [9], we have for all states *i, j* and time t ≥ 0:

$$P_{ij}'(t) = \sum_{k \neq j} q_{kj} . P_{ik}(t) - q_j . P_{ij}(t) \qquad (8)$$

This system of differential equations for $P_{ij}(t)$ is known as the forward kolmogorov equations for the Continuous time Markov chain {X (t): t≥ 0}.

A complete discussion of uniqueness and limiting behavior solution of this system is given in [1].

Numerical Approximation of Markov Renewal Equations:

The systems of differential equations for $P_{ij}(t)$ discussed in sections 3.1and 3.2 represented by equations (3,4) and (8) is difficult to work with analytically, and is not easy to use for computational purpose. For this reason we have to search for approximation methods to find the numerical approximation for these systems of differential equations for $P_{ij}(t)$.

Numerical Approximation of Birth Equations:

Equations (3) and (4) discussed in section 3.1 can be solved numerically by using the deterministic Monte Carlo method. One application of this method is the numerical differentiation. The Runge-Kutta methods are commonly utilized for approximation of the difference - differential equations (3, 4) and (8). These methods were developed to solve a single equation involving a single independent and a single dependent variable [2]. In order to solve equations (3) and (4) we rewrite them into the following finite system of n first order differential equations:

$$\frac{dy_n}{dt} = y'_n = -\lambda . y_n + \lambda . y_{n-1} \quad ; n \geq 1, \lambda \geq 0 \qquad (9)$$

with initial conditions $y_0 = 1, y_i = 0 \quad for \quad i = 1, 2, \ldots, n$

Figure 1 plotes the results of solving equation (9) with 6 equations (i.e. n = 6), using step size h = 0.1, $\lambda = 0.5$ and time t = 0, 0.5, 1.0, …, 10, against the solution of the first two equations (red curve for first equation, violet curve for the 2nd) of system (9).

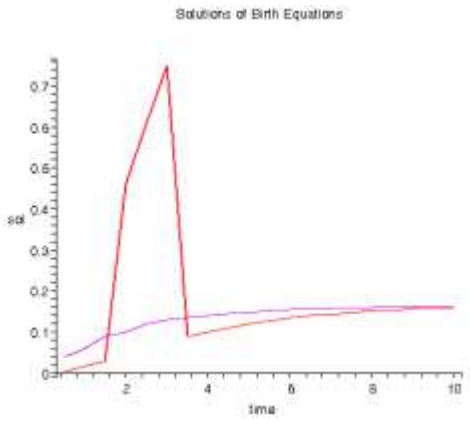

Figure 1: Solutions of Equation (9) using *n* = 2

Numerical Approximation of *CTMC* Equations:

In order to solve equation (8) by using the deterministic Monte Carlo method mentioned in section 4.1,we rewrite them into the following finite system of n first order differential equations:

$$y'_n = -q_n . y_n + q_{n+1} . y_{n+1} + q_{n+2} . y_{n+2} + q_{n+3} . y_{n+3} \quad ; n \geq 1, q_n > 0 \qquad (10)$$

$y_0 = 1, y_i = 0 \quad for \quad i = 1, 2, \ldots, n$ with initial conditions

Figure 2 plotes the results of solving equation (10) with 4 equations (i.e. n = 4), using time t = 0, 0.5, 1.0, …, 10 against the solution of the first two equations (red curve for first equation, violet curve for the 2nd) of system (10), with appropriate *relative error* = *0.001* and *absolute error* = *0.001*. where the values of the constants q's are:

$q_1 = 0.01$, $q_2 = 0.02$, $q_3 = 0.03$, $q_4 = 0.09$, $q_5 = 0.01$
$q_6 = 0.20$, $q_7 = 0.0023$, $q_8 = 0.15$, $q_9 = 0.02$, $q_{10} = 0.10$

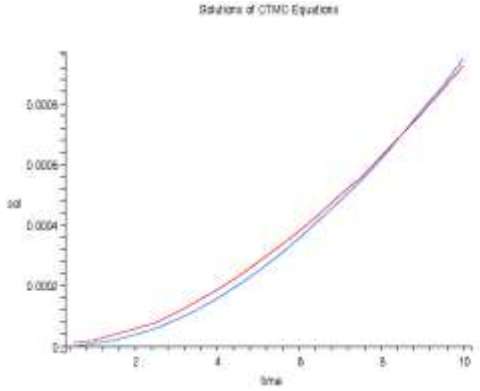

Figure 2: Solutions of Equation (10) using *n* = 2.

Numerical Simulation of Markov Renewal Equations:

In this section we discus the numerical simulation of Birth equations (3, 4) and *CTMC* equation (8) by providing a *Markov Chain Monte Carlo* (*MCMC*) these method for generating the underlying process of equations. *MCMC* draws samples by running a Markov chain that is constructed so that its limiting distribution is the joint distribution of interest [8].

Numerical Simulation of Birth Equations:

Since a Poisson process is a Birth process with rates $\lambda_i \equiv \lambda$, it is then easy to solve equations (3) and (4) by simulating the underlying Poisson process of these equations. To generate the arrival times that follow a Poisson process by using *MCMC* simulation [3], we use the model in figure (3).

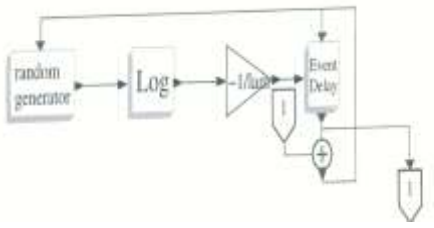

Fig 3: Super block generating arrival events based on Poisson's law.

This model can be formulated in to the following suggested algorithm.

Poisson process Monte Carlo (*PPMC*) algorithm:

1- Set the expected number of arrivals $\lambda = E[N(1)]$ in any unit interval.

2- Set the initial time $t_0 = 0$.

3- Generate a uniformly distributed random number U between 0 and 1.

4- Taking the logarithm of U and multiplying by $-1/\lambda$.

5- Return $t_i = t_{i-1} - (1/\lambda).\ln U$

6- Repeat steps 3 - 5 n times; stop

Table 1 summarizes the outputs of *PPMC* algorithm.

Rate process	**Arrival times t_i**									
λ	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
$\lambda = 0.1$	8.05	8.24	17.08	17.15	35.65	36.91	47.71	74.91	96.11	101.7
$\lambda = 0.2$	2.96	15.36	20.46	22.38	24.09	31.44	33.26	41.61	64.21	78.81
$\lambda = 0.3$	7.46	8.71	20.91	27.94	29.30	37.06	37.89	37.93	39.25	45.94
$\lambda = 0.4$	4.90	7.55	9.64	9.91	11.70	15.12	15.24	16.35	21.00	21.39
$\lambda = 0.5$	0.08	0.35	0.54	1.74	6.70	12.10	13.05	16.05	18.63	20.44
$\lambda = 0.6$	0.20	2.62	3.46	7.89	9.78	10.60	11.20	11.60	12.50	12.50
$\lambda = 0.7$	0.61	2.75	6.78	6.95	9.20	9.41	12.30	12.50	12.80	12.90
$\lambda = 0.8$	1.07	1.14	2.45	2.62	7.22	9.26	10.00	10.70	12.10	13.80
$\lambda = 0.9$	1.24	1.58	2.30	2.50	3.59	4.84	5.54	6.76	7.04	8.38
$\lambda = 1.0$	3.43	6.62	7.39	7.80	10.70	11.7	11.90	12.60	14.10	15.90

Table 1: Poisson arrival times $\boldsymbol{t_i}$.

Figure 4 plots the arrival times $\{t_i\}_{i=1}^{10}$ along with $\lambda_* = \min\{\lambda\} = 0.1$ and $\lambda^* = \max\{\lambda\} = 1.0$ as indicated. Values of arrival times $\{t_i\}_{i=1}^{10}$ from stationary Poisson process are marked by the crosses on the λ^* (red upper crosses) , λ_* (violate down crosses) lines. and do appear to be fairly uniformly spread.

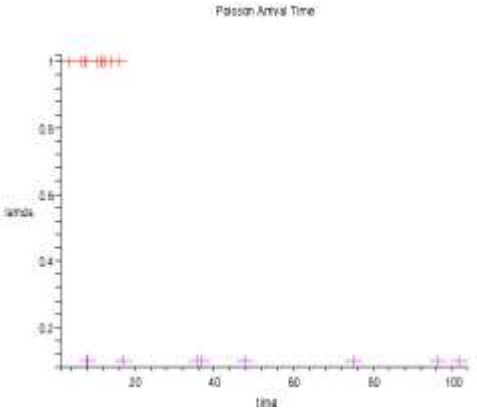

Figure 4: Poisson arrival times.

We generate a sequence of events where the time between events is independent random variable with exponential law. The result can be seen clearly in figure 5, where we plot the arrival times $\boldsymbol{t_1}$ (red upper curve), $\boldsymbol{t_2}$ (green down curve) against the parameter $\lambda = 0.1, 0.2, \ldots, 1.0$.

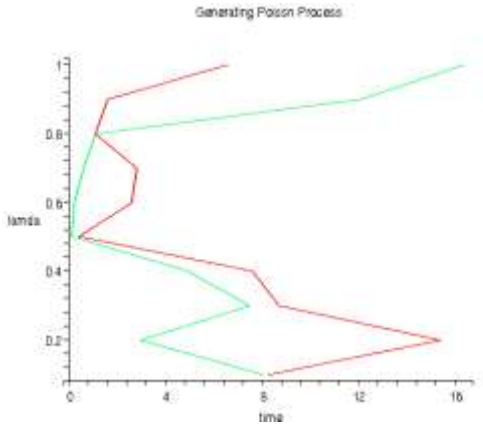

Figure 5: Generating Poisson process.

We can also generate a Poisson process with parameter λ by average the arrival times $\{t_i\}_{i=1}^{10}$ of the 10 runs for each value of λ . The result can be shown in figure 6, where we plot the average arrival times $\{\bar{t}_i\}_{i=1}^{10}$ against the parameter λ .

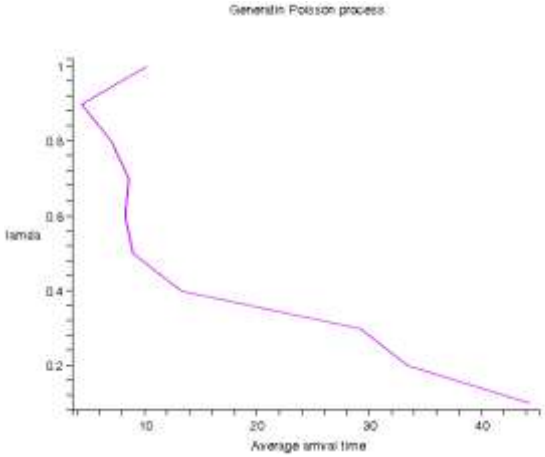

Figure 6: Generating Poisson process.

Generating a nonstationary Poisson process by thinning:

To do this we first discuss how to estimate the rate function $\lambda(t)$ of a nonstationary Poisson process using the following suggested method.

Piecewise constant method:

1- Divide the time interval I = [a, b] in to n subinterval $I_1, I_2, \ldots, I_n$.
2- For each day, determine the number of arrivals in each of these subintervals.
3- For each subinterval, compute the average number of arrivals over the m days, these n averages are estimates of expected number of arrival $E[N(s)]$ in corresponding subintervals.
4- To obtain an estimates of the arrival rate $\hat{\lambda}(t)$ for that subinterval divide the average number of arrivals in that subinterval by the subinterval length.

Table 2 summarizes the number of arrival customers to a mobile repairing shop between 10 A.M. and 2.0 P.M. for seven different days. To obtain an estimate of $\lambda(t)$, the 4 hour interval was divided into 24 subintervals each of length 10 sec.

Interval	Sat.	Sun.	Mon.	Tue.	Wed.	Thu.	Fri	**Average**
10.0,10.1	1	1	1	0	0	1	0	0.571
10.1,10.2	1	0	1	2	0	2	0	0.86
10.2,10.3	1	1	1	0	2	0	0	0.71
10.3,10.4	0	1	0	1	0	0	1	0.43
10.4,10.5	1	0	2	1	0	1	1	0.86
10.5,11.0	0	1	1	0	0	1	0	0.43
11.0,11.1	0	2	1	0	0	1	0	0.29
11.1,11.2	0	1	1	3	1	0	0	0.86
11.2,11.3	1	0	1	1	2	2	0	1.0
11.3,11.4	2	1	2	0	0	0	1	0.86
11.4,11.5	1	1	0	0	2	1	2	1.0
11.5,12.0	0	0	2	1	0	0	1	0.57
12.0,12.1	1	1	2	0	0	0	1	0.71
12.1,12.2	0	1	3	2	0	1	2	1.0
12.2,12.3	1	0	0	2	1	1	1	0.86
12.3,12.4	3	0	0	1	١	١	0	0.86
12.4,12.5	0	0	1	0	1	1	2	0.71
12.5,1.0	0	0	1	1	1	0	3	0.86
1.0,1.10	1	1	1	0	0	2	0	0.71
1.10,1.20	0	1	2	1	0	2	0	0.86
1.20,1.30	1	0	0	2	1	0	1	0.71
1.30,1.40	1	0	0	1	0	0	2	0.57
1.40,1.50	2	1	0	2	0	2	0	1.0
1.50,2.0	1	0	1	2	1	0	0	0.71

Table 2: Customers arrival data.

Applying step 4 of the Piecewise constant method of specifying $\lambda(t)$ we get the estimation of $\lambda(t)$ for each subintervals mentioned in table 4.

$\hat{\lambda}(t)$ = 5.71, 8.6, 7.1, 4.3, 8.6, 4.3, 2.9, 8.6, 1.0, 8.6, 1.0, 5.7, 7.1, 1.0, 8.6, 8.6 , 7.1, 8.6, 7.1, 8.6, 7.1, 5.7, 1.0, 7.1.

Figure 7 plots the estimated rate function in customers per minutes for the arrival process between 10 A.M. and 2 P.M.

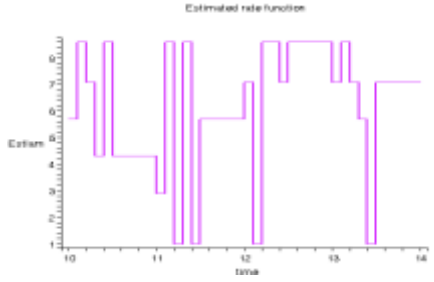

Figure 7: Estimated rate function $\lambda(t)$.

We now discuss how to generate arrival times that follow a nonstationary Poisson process with rate function $\lambda(t) = E[N(t)]$ by using *MCMC* method (*thinning algorithm*). We present the following suggested *thinning algorithms:*

Thinning algorithm :

This algorithm is the analog to the acceptance rejection method for variates generation [6]. Define the expectation function $\wedge(t)$ by :

$$\wedge(t) = \int_0^t \lambda(y)\, dy \qquad (11)$$

which is the expected number of arrivals between time 0 and time t. A nonstationary Poisson process with expectation function $\wedge(t)$ can be generated as follows:

1. Generate Poisson arrival times $\{t_i'\}$ at rate 1..

2. Set $t_i = \wedge^{-1}(t'_{i-1})$ to be a nonstationary Poisson arrival times.

Figure (8) plots the expectation function $\wedge(t)$;obtained from equation 10; corresponding to the rate function $\lambda(t)$ obtained by applying step 4 of the Piecewise constant method.

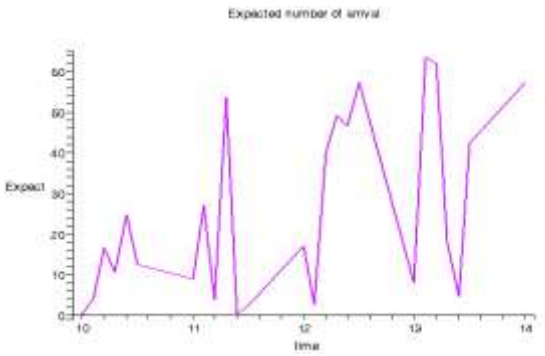

Figure 8: Generating a nonstationary Poisson process by thinning.

From figure 7 and 8 we note that $\wedge(t)$ is piecewise linear, since $\lambda(t)$ was specified to be piecewise constant. Also, $\lambda(t)$ rises most steeply for those values of t where $\lambda(t)$ is highest.

Numerical Simulation of *CTMC* Equations:

To numerically simulate the forward kolmogorov equations (8), we only need to simulate the associated underlying *CTMC* process using the following procedure:

Recall that a Markov chain is determined by a set of states, a transition matrix P, and an initial state vector X_0. The relation between successive state vector is given by [7]:

$$X_{n+1} = X_n \,.\, P \;;\; n \geq 0 \qquad (12)$$

and the state vector at the nth observation is given by $X_n = X_0 \,.\, P^n$. We can estimate the coordinates of the state vector at the nth observation using a simulation as follows:

1- Determining the state vector after *n* transitions for a specific initial state *i* using the following suggested algorithm:

- Generate *U* uniformly on [0, 1)
- Select a value u_j of *U*
- Find the correspondig value x_i of the random variable X using the relation:

$$\text{I f} \quad \sum_{k=0}^{j-1} p_k \leq u_j < \sum_{k=0}^{j} p_k \quad \text{then set } v_j = X_n \qquad (13)$$

where the sequence $\{v_j\}$ is distributed as the random variable X.

2- To determining the probability that the system will be in state j on observation *n*, we begin in state i and track the system through *n* transitions using data from the transition matrix. We repeat the process k times and use the fraction:

(number of times the system is in state *j* on observation *n*) / K (14)

as an estimate of the *n*-step transition probability $P_{ij}(n)$.

Application:

Consider a four states Markov chain with initial state vector $\mathbf{X_0}$ = [0.2, 0.4, 0.1, 0.3] and a transition matrix:

$$P = \begin{bmatrix} .4 & .3 & 0 & .3 \\ .5 & .1 & .4 & 0 \\ .1 & .2 & .1 & 6 \\ .2 & .1 & .7 & 0 \end{bmatrix}$$

we shall use simulation to estimate:

1- the five step transition matrix P^5
2- The state vector after five transitions, as follows:

The exact value of the five step transition matrix is given by:

$$P^5 = \begin{bmatrix} .2820 & .1922 & .2443 & .2815 \\ .2961 & .1752 & .3267 & .2021 \\ .2712 & .1956 & .2235 & .3097 \\ .2934 & .1713 & .3421 & .1932 \end{bmatrix}$$

To estimate the five step transition matrix using simulation, we use relation (13) for j = 1, 2, 3, 4 ; n = 5; and k = 10000, as an estimate for the probability that the system makes a transition from state 1 to state 1, 2, 3, 4 in five steps. The result of aspecific simulation consisting of 10000 runs is:

$$P^{\ddot{5}} = \begin{bmatrix} .2739 & .1895 & .2412 & .2801 \\ .2869 & .1745 & .3197 & .1979 \\ .2519 & .1936 & .2205 & .3067 \\ .2749 & .1691 & .3317 & .1892 \end{bmatrix}$$

Next to estimate the state vector after five transitions we use relation (14). Suppose we have an initial state vector $\mathbf{X_0}$ = [0.2, 0.4, 0.1, 0.3]. The first step of simulation is to generate values distributed as a random variable $\mathbf{X_0}$ with the probability density function shown in table 3:

State	1	2	3	4
propability	0.4	0.2	0.1	0.3

Table 3.

To do so, we select a random number $\boldsymbol{u}$ uniformly distributed on (0, 1]. If u < 0.4 then the initial state is 1; if $0.4 \leq$ u < 0.6 then the initial state is 2, if $0.6 \leq$ u < 0.7 then the initial state is 3, if 0.7< u then the initial state is 4. A simulation of 10000 runs gives the vector [0.2249, 0.2449, 0.4583, 0.2863] as an estimate for the given state vector after five transitions.

Discussion:

This paper deals wih the numerical solution of the forward kolmogorov equations of type Birth equations and Continuous Time Markov Chain equations. This research also treats Markov Chain Monte Carlo simulation problem for simulating the underlying process of these equations, we discuss three methods for generating stationary and a non-stationary Poisson process, in addition to generating the Continuous-time Markov Chain with many applications, we can make a

general comparison between numerical solution and numerical simulation solution of these equations as fllows:

As the forward kolmokrov equations become more detailed, a nalytical and numerical results become very difficult to obtain, for this reason Markov Chain Monte Carlo simulation technique become an increasingly attractive way for generating the underlying process of thes equations, and has been found to be a useful and simple powerful test.

References:

1- Asmussen S., Applied Probability and Queues , John Wiley and Sons, Inc., 1987.

2- Burden R., Faires, J. and Reynolds, Albert G., Numerical Analysis , Prindle, Weber and Schmidt, 7th Edition, 2001.

3- Campbell S., Chancelier J. and Nikoukhah R., Modeling and Simulation in Scilab / Scicos , Springer Science + Business, Inc., 2006.

4- Cinlar E., Introduction to Stochastic Process, Prentice - Hall, Inc., 1975.

5- Heyman D. and Sobel M. ,Stochastic Models in Operations Research (vol. I), McGraw-Hill Book Company, 1982.

6- Law A. and Kelton W., Simulation Modeling and Analysis, Mc Graw-Hill Higher Education, 3rd Edition, 2000 .

7- Maki D. and Thompson M., Mathematical Modeling and Computer Simulation , Thomson Brooks/Cole, 2006 .

8- Mackay D. J., Information Theory, Inference and Learning Algorithms, Draft Copy 2.4.1, 2002.

9- Ross, Sheldon M., Simulation , Academic Press, 2nd Edition, 1997.

10- Swick, K. E., A Nonlinear model for human population dynamics , SIAM J. APPL. MATH., Vol. 40, No. 2, 1981.

11- Taylor H. and Karlin S., An introduction to Stochastic Modeling , Academic Press Inc., 1984.

توليد العملية التحتية لمعادلات كولموكروف التقدمية

عبد الكريم إبراهيم شيت

قسم الرياضيات ، كلية التربية الأساسية ، جامعة الموصل ، الموصل ، العراق

(تاريخ الاستلام: / /٢٠٠٧ ، تاريخ القبول: / / ٢٠٠٧)

الملخص:

تناول هذا البحث مسألة الحل العددي لمعادلات ماركوف التقدمية، إضافة إلى استخدام أسلوب محاكاة المونت كارلو لسلاسل ماركوف لتوليد العملية التحتية لهذه المعادلات من اجل توقع مسارالعينة لهذه العمليات. لتوضيح كيفية استخدام هذا الأسلوب تم إجراء المحاكاة العددية وإيجاد الحل العددي عندما تكون العملية التحتية من نوع بواسون ومن نوع سلاسل ماركوف المستمرة الزمن. وتبين من خلال تفحص النتائج كفاءة أسلوب محاكاة المونت كارلو لسلاسل ماركوف وسهولته في توقع سلوك مسار العينة لهذه العمليات على امتداد الزمن.