SIMULATION FROM ENDPOINT-CONDITIONED, CONTINUOUS-TIME MARKOV CHAINS ON A FINITE STATE SPACE, WITH APPLICATIONS TO MOLECULAR EVOLUTION

Yifan Li, Jia Liu, Yiyang Yuan

April 30, 2017

1 Introduction

Continuous Markov chain has a wide range of applications, from computational finance to human genetics and genomics. In terms of Discrete time Markov chain(DTMC), it is very easy to get a complete path. All you need to do is checking the state at every time point. Then, you can get a path ,for example $\{A, B, A, B, B \dots\}$. But in Continuous time Markov chain(CTMC), the situation is different. Usually, we can't check the state continuously. Under normal circumstances, we can only obtain some data sampled at discrete time points of the continuous-time Markov chain instead of getting the continuously-observed sample paths. We don't know what happened between two time points. However, most of the application areas mentioned above need statistical inference which is based on continuous sample path. This requires us to simulate the continuous time Markov chain path by observing and analyzing discrete samples and this paper presents four general solutions to this path simulation problem when the state space of the CTMC is a finite and discrete space. Our work is to simulate continuous sample paths from the CTMC conditional on $X(T_0), X(T_1) \dots, X(T_N)$. $T_0, T_1 \dots, T_n$ are the time points we sample the path. As a consequence of the Markov assumption, knowledge of the data $X(T_0), ..., X(T_N)$effectively partitions the process into independent components $\{X(t) : T_k \leq t \leq T_{k+1}\}$ whose endpoints $X(T_k)$ and $X(T_{k+1})$ are known. Thus, we can sample from N independent CTMCs whose endpoints are known instead of sampling from the CTMC conditional on $X(T_0), X(T_1) \dots, X(T_N)$. Specifically, we only consider the CTMC whose beginning and ending states are known across a time interval of certain and try to simulate its sample path which include intermediate states as well as times of transition. This paper will first discuss the four main methods: (1) forward sampling, (2) modified rejection sampling, (3) direct sampling, and (4) uniformization. In addition, although the current path simulation is done by the computer, this process could be very time-consuming unless implemented by method which is sufficient enough. So after determining the feasible simulation method, it is also crucial to screen out a high efficiency strategy. So that in the second part, I'll analyze the expectation of computation cost of generating one sample path for each. In the final part, I will show some special examples to give you an intuitive understanding of the efficiency of each method.

2 Basic Knowledge of CTMC

Before we start simulating path, the first question we need to figure out is: how long will this process remain in a given state, say $x \in S$? Suppose $X(0) = T$ and let T_x denote the time we transition away from state x. From the Markov property, we have:

$$P(T_x > s + t | T_x > s) \quad = P(T_x > t)$$

Therefore, T_x satisfies the loss of memory property and as we know, exponential random variable has this property among continuous random variables . We denote Q_x as the parameter of the exponential holding time for state x, so that we have:

$$E[T_x] = \frac{1}{Q_x}$$

Thus, the higher the rate Q_x, representing the rate out of state x, the smaller the expected time for the transition to occur, which is intuitively pleasing. However, even if we already know Q_x, we still can not figure out what state it transition to? So we denote

$$p_{xy} = P(X(T_x) = y | X(0) = x)$$

as the probability that the process transitions to state y after leaving state x. Then we define:

$$Q_{xy} = p_{xy} Q_x$$

So we have:

$$\sum_{y \neq x} Q_{xy} = \sum_{y \neq x} p_{xy} Q_x = Q_x$$

Finally, we define:

$$Q_{xx} = -Q_x$$

After doing these preparations, we have the instantaneous rate matrix Q of CTMC with off-diagonal entries $Q_{xy} \geq 0$ and diagonal entries $Q_{xx} = -\sum_{y \neq x} Q_{xy} = Q_x < 0$.

$$Q = \begin{bmatrix} Q_{xx} & Q_{xy} & Q_{xz} \\ Q_{yx} & Q_{yy} & Q_{yz} \\ Q_{zx} & Q_{zy} & Q_{zz} \end{bmatrix}$$

and

$$f(X(t) = y|X(0) = x) = Q_{xy}e^{Q_{xx}t}$$

In DTMC, if the transition matrix P has good property, we have stationary distribution $\pi = (\pi_x, \pi_y, \dots)$ by solving the equation:

$$\pi P = \pi$$

Similarly, in CTMC, if the instantaneous rate matrix Q has good property, we can also get stationary distribution $\pi = (\pi_x, \pi_y, \dots)$. However, the stationary distribution π satisfy:

$$\pi Q = 0$$

3 Assumptions

In the section above, I have introduced some basic definitions of CTMC. Now, I will introdece some assumptions of my model

(1) The stochastic process$\{X(t) : 0 \leq t \leq T\}$ is irreducible and positive recurrent so that a stationary distribution π exists.

(2) State space is $\{x, y, z\}$

(3) The instantaneous rate matrix Q of CTMC is

$$Q = \frac{1}{s}\begin{bmatrix} Q_{xx} & Q_{xy} & Q_{xz} \\ Q_{yx} & Q_{yy} & Q_{yz} \\ Q_{zx} & Q_{zy} & Q_{zz} \end{bmatrix}$$

The scaling parameter s is chosen such that $\sum_{x \in S} Q_x \pi_x = 1$, implying that t substitutions are expected in t time units. By doing this, we can compare the efficiency of each method in different situations.(different instantaneous rate matrix Q)

(4) the process transitions from state x to state y before time t follows this distribution:

$$f(X(t) = y|X(0) = x) = Q_{xy}e^{Q_{xx}t}$$

4 Simulation Methods

4.1 Method 1: Forward sampling

Suppose we simulate a realization of a finite-state CTMC $\{X(t) : 0 \leq t \leq T\}$ conditional on its beginning state $X(0) = a$ and ending state $X(T) = b(a, b \in \{x, y, z\}$ and a and b can be the same). We denote τ as the time we transition away from state x. From assumptions above, we have $\tau \sim \exp(Q_x)$.

If $\tau > T$, this means that there is no state change in the interval $[0, T]$, and the corresponding sample path is constant. Otherwise, we need to choose a new state c drawn from the discrete probability distribution with probability masses $p_{ac} = Q_{ac}/Q_a$. By doing this, we get the first part of the path:

$$X(t) = \begin{cases} a & ,0 \leq t < \tau \\ c & .t = \tau \end{cases}$$

Now, the starting state is replaced by $X(\tau) = c$ and the time interval is replaced by $[\tau, T]$. This is equivalent to simulating a new realization conditional on its beginning state $X(0) = c$ and ending state $X(T - \tau) = b$. We then repeat the steps above until $\tau > T$.

Algorithm 1 (Forward sampling).

(1) Sample $\tau \sim \exp(Q_a)$. If $\tau \geq T$, we are done: $X(t) = a$ for all $t \in [0, T]$.

(2) If $\tau < T$, choose a new state $c \neq a$ from a discrete probability distribution with probability masses Q_{ac}/Q_a. Repeat the procedure with new beginning state c and new time interval $[\tau, T]$.

Under the assumption that the ending state $X(T) = b$ is observed, conditioning excludes all paths sampled from the preceding algorithm that fail to end in state b.

4.2 Method 2: Rejection sampling

In Forward sampling, the acceptable candidates are those for which the simulated ending state and the observed ending state are the same, so that the acceptance probability of this method is $P_{ab}(T) = \exp(Q_t)_{ab}$. Thus, if T is large, $\lim_{T\to\infty} \exp(Q_t)_{ab} = \pi_b$. Conversely, if T is small and $a \neq b$, the probability is approximately $Q_{ab}T$. Because of this, in the following two cases, the the acceptance probability of forward sampling is small.

(1) T is large and π_b is small

(2) T is small

This means that almost every path sampled from forward sampling fails to end in state b and will be excluded. We need to find a more efficient method.

By a conditioning argument, the time τ to the first state change given at least one state change occurs before T and X(0) = a has cumulative distribution function:

$$\begin{aligned} & P(\tau < t,\ t < T | \tau < T) \\ &= \frac{P(\tau < t,\ \tau < T, t < T)}{P(\tau < T)} \\ &= \frac{P(\tau < t)}{P(\tau < T)} \\ &= \frac{1 - e^{-tQ_a}}{1 - e^{TQ_a}} \end{aligned}$$

The corresponding density function is:

$$f(\tau) = \frac{Q_a e^{-tQ_a}}{1 - e^{-TQ_a}},\ 0 \leq \tau \leq T \tag{4.1}$$

Here is a trick. In programs like R or Matlab, it is easy to sample a random variable from common distributions, for example, normal distribution and chi square distribution. We just need to use command 'rnorm' or 'rchisq'. Is there a general way to sample a random variable from any density function like (4.1). The following is a solution to this problem.

First, we need to calculate its cumulative distribution function from its density function as we have done before. Then, we calculate the inverse function of its cumulative distribution function. Now, we have:

$$F^{-1}(u) = -log[1 - u(1 - e^{-TQ_a})]/Q_a.$$

Denote $u = F(x)$ and with a little calculation we can get:

$$\begin{aligned}
&P(u < y)\\
&= P(F(X) < y)\\
&= P(X < F^{-1}(y))\\
&= F(F^{-1}(y))\\
&= y
\end{aligned}$$

so that u follows uniform distribution. Thus, upon sampling u from a Uniform(0,1) distribution, transformation yields the sample waiting time $F^{-1}(u)$ to the first state change of the CTMC.

Algorithm 2 (Modified rejection sampling).

If a=b:

(1) Simulate from $\{X(t) : 0 \leq t \leq T\}$ using the forward sampling algorithm.

(2) Accept the simulated path if X (T) = a; otherwise, return to step 1 and begin anew.

If $a \neq b$:

(1) Sample τ from the density (4.1) using the inverse transformation method, and choose a new state $c \neq a$ from a discrete probability distribution with probability masses Q_{ac}/Q_a

(2) Simulate the remainder $\{X(t) : \tau \leq t \leq T\}$ using the forward sampling algorithm from the beginning state $X(\tau) = c$.

(3) Accept the simulated path if X (T) = b; otherwise, return to step 1 and begin anew.

In short, modified rejection sampling explicitly avoids simulating constant sample paths when it is known that at least one state change must take place. For example, the starting state and the ending state are different, we know that there must be at least one state change. But in forward sampling, we don't make use of this knowledge. If T is small,(which means most of paths we get from forward sampling are constant) we will save much time.

Nevertheless, if Q_{ab}/Q_a is small, that is to say, the transition from a to b is unlikely, rejection sampling is still not efficient enough.

4.3 Method 3: Direct sampling

Before I introduce direct sampling, there are some properties of matrix we need to know. If the matrix Q admits an eigenvalue decomposition, let U be an orthogonal matrix with eigenvectors as columns and let D_λ be the diagonal matrix of corresponding eigenvalues, then we have $Q = UD_\lambda U^1$.

Now we require the instantaneous rate matrix Q have this property. Then, for any time t, the transition probability matrix of CTMC can be calculated as:

$$P(t) = e^{Qt} = Ue^{tD_\lambda}U^{-1} \text{ and } P_{ab}(t) = \sum_j U_{aj}U_{jb}^{-1}e^{t\lambda_j} \tag{4.2}$$

First, we consider the situation that the starting state and the ending state are the same. That is, $X(0) = X(T) = a$. The probability that there are no state changes in the time interval [0, T] conditional on X(0) = a and X(T) = a is given by:

$$\begin{aligned} &P(\tau > T|X(t) = a,\ \forall t \in [0,T]) \\ &= \frac{P(\tau > T,\ X(t) = a,\ \forall t \in [0,T])}{P(X(t) = a,\ \forall t \in [0,T])} \\ &= \frac{P(\tau > T)}{P_{aa}(T)} \\ &= \frac{e^{-TQ_a}}{P_{aa}(T)} \end{aligned} \tag{4.3}$$

In the other word, the probability of a sample path from the CTMC being the constant $X(t) = a,\ \forall t \in [0,T]$ is p_a, so that the probability of at least one state change happening is $1 - p_a$.

Next, we consider the situation that the starting state and the ending state are different. That is, $X(0) = a,\ X(T) = b,\ a \neq b$. Let τ denote the waiting time until the first state change. The conditional probability that the first state change is to i at a time smaller than t is:

$$
\begin{aligned}
&P(\tau \leq t, X(\tau) = i | X(0) = a, X(T) = b) \\
&\quad = P(\tau \leq t, X(\tau) = i, X(0) = a | X(T) = b) / P(X(0) = a | X(T) = b) \\
&\quad = \int_0^t Q_a e^{-Q_a z} \frac{Q_{ai}}{Q_a} \frac{P_{ib}(T-z)}{P_{ab}(T)} \mathrm{d}z \\
&\quad = \int_0^t f_i(z) \mathrm{d}z
\end{aligned}
$$

Here we use $f_i(z)$ to replace the integrand. From the calculation above, we can obtain p_i, probability that we first change to state i conditional on the endpoints X(0) = a and X(0) = b:

$$
p_i = \int_0^T f_i(t) \mathrm{d}t, \ i \neq a, \ a \neq b \tag{4.4}
$$

Using (4.2), we can rewrite the integrand as:

$$
f_i(t) = Q_{ai} e^{-Q_a t} \frac{P_{ib}(T-t)}{P_{ab}(T)} = \frac{Q_{ai}}{P_{ab}(T)} \sum_j U_{aj} U_{jb}^1 e^{T\lambda_j} e^{-t(\lambda_j + Q_a)} \tag{4.5}
$$

If we put (4.5) into (4.4), we have:

$$
\begin{aligned}
p_i &= \int_0^T \frac{Q_{ai}}{P_{ab}(T)} \sum_j U_{aj} U_{jb}^1 e^{T\lambda_j} e^{-t(\lambda_j + Q_a)} \mathrm{d}t, \ i \neq a, \ a \neq b \\
&= \frac{Q_{ai}}{P_{ab}(T)} \sum_j U_{aj} U_{jb}^1 e^{T\lambda_j} \int_0^T e^{-t(\lambda_j + Q_a)} \mathrm{d}t \\
&= \frac{Q_{ai}}{P_{ab}(T)} \sum_j U_{aj} U_{jb}^1 J_{ai}
\end{aligned} \tag{4.6}
$$

where

$$
J_{ai} = \begin{cases} T e^{T\lambda_j} & , \lambda_j + Q_a = 0 \\ \frac{e^{T\lambda_j} - e^{-Q_a T}}{\lambda_j + Q_a} & . \lambda_j + Q_a \neq 0 \end{cases}
$$

Finally we get p_i, the probability that the first state change is to i in the time interval [0, T] conditional on X(0) = a and X(T) = a.

Algorithm 3(Direct sampling)

(1) If a = b, sample $Z \sim Bernoulli(p_a)$, where p_a is given by (4.3). If Z = 1, we are done: $X(t) = a, 0 \leq t \leq T$.

(2) If $a \neq b$ or Z = 0, then at least one state change occurs. Calculate p_i for all $i \neq a$ from (4.6). Sample $i \neq a$ from the discrete probability distribution with probability masses $p_i / p_{-a}, i \neq a$, where $p_{-a} = \sum_{j \neq a} p_j$.

(3) Sample the waiting time τ in state a according to the continuous density $f_i(t)/p_i, 0 \leq t \leq T$, where $f_i(t)$ is given by (4.5). Set $X(t) = a, 0 \leq t \leq \tau$.

(4) Repeat procedure with new starting value i and new time interval of length $T - \tau$.

4.4 Method 4: Uniformization

In the last method, we construct an auxiliary stochastic process Y(t) which is a discrete-time Markov process. Denote $\mu = \max_c Q_c$ and the transition matrix of DTMC Y(t) is given by:

$$R = I + \frac{1}{\mu} Q \tag{4.7}$$

Here are something we should pay attention. In DTMC, if $X(t) = X(t+1)$(for example, they are all in state a). we still take it as a transition. However, in CTMC, we should not take it into count. We call it virtual state changes and they occor when $R_{aa} > 0$.

The reason we do this transformation is to get the number of state changes N (including the virtual) for the conditional process that starts in X(0) = a and ends in X(T) = b. After we sampling N, let $t_1, \ldots, t_N$ denoted as the time state changes(including the virtual) occur and they are uniformly distributed in the time interval [0, T]. Then, we use the transition matrix R of the DTMC conditional on the beginning state X(0) = a and ending state $X(t_n) = b$ to sample $X(t_1), \ldots, X(t_{N-1})$. Then, we get the whole path.

Now, let's calculate $P(N = n|X(0) = a, X(T) = b)$. By doing some simple calculations, we can get:

$$P(t) = e^{Qt} = e^{\mu(R-I)t} = e^{-\mu t} \sum_{n=0}^{\infty} \frac{(\mu t R)^n}{n!} = \sum_{n=0}^{\infty} e^{-\mu t} \frac{(\mu t)^n}{n!} R^n \tag{4.8}$$

and

$$P_{ab}(t) = P(X(t) = b|X(0) = a) = e^{-\mu t} 1_{(a=b)} + \sum_{n=1}^{\infty} e^{-\mu t} \frac{(\mu t)^n}{n!} R_{ab}^n$$

Then, we can say that the number of state changes N (including the virtual) for the conditional process that starts in X(0) = a and ends in X(T) = b is given by:

$$P(N = n|X(0) = a, X(T) = b) = e^{-\mu T} \frac{(\mu T)^n}{n!} R_{ab}^n / P_{ab}(T) \tag{4.9}$$

Now, I can introduce the algorithm.

Algorithm 4(Uniformization)

(1) Simulate the number of state changes n from the distribution (4.9).

(2) If the number of state changes is 0, it means there is no state changes in time interval [0,T]. So we are done: $X(t) = a, 0 \leq t \leq T$.

(3) If the number of state changes is 1 and a=b, it means there is a virtual state changes in time interval [0,T]. So we are done: $X(t) = a, 0 \leq t \leq T$.

(4) If the number of state changes is 1 and $a \neq b$, it means there is a virtual state changes in time interval [0,T] and it changes from state a to state b. Then we simulate t_1 uniformly random in [0,T], we are done: $X(t) = a, t < t_1$, and $X(t) = b,\ t_1 \leq t \leq T$.

(5) When the number of state changes n is at least 2, simulate n independent uniform random numbers in [0, T] and sort the numbers in increasing order to obtain the times of state changes $0 < t_1 < \cdots < t_n < T$. Simulate $X(t_1), \ldots, X(t_{n1})$ from a discrete-time Markov chain with transition matrix R and conditional on starting state X(0) = a and ending state $X(t_n) = b$. Determine which state changes are virtual and return the remaining changes and corresponding times of change.

Now, I have introduced all four methods. In the next section, I will compare the efficiency of these four methods in different situation.

5 Efficiency

Before we doing comparison, here are three things we need to pay attention.

1. Since when starting state and ending state are the same, Rejection sampling is the same as Forward sampling and when starting state and ending state are different, Rejection sampling is better than Forward sampling. Thus, Rejection sampling is strictly better than Forward sampling. Because of this, we only need to compare Rejection sampling, Direct sampling and Uniformization sampling.

2. It is obvious that the efficiency of different methods depend on the instantaneous rate matrix Q. Thus, I'll do the comparisons for different the instantaneous rate matrixes.

3. In Rejection sampling, some path we simulate will fail to end in the state we want and tend to be rejected. This makes the probability of success will greatly affect the efficiency of the algorithm. Let p_{acc} be the acceptance probability for the rejection sampling algorithm. If the time interval T is large and the ending state is X(T)=b, then $p_{acc} \approx \pi_b$. Since we have introduced scaling parameter s to make the expected substitution is once in one time units, from figure 1(in appendix) we can see that the probability of success tends to be constant when the time interval T is larger than 4. In the following comparisons, I'll do the comparison in case of T=3 and T=30.

5.1 Situation 1

Now, it's time for us to think of strange matrixes. First, I would like to test a common situation:

$$Q = \frac{1}{s}\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

where s=2.

By some simple calculation, its stationary distribution is:

$$\pi = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$$

From figure 2 and figure 3(in appendix), we can draw conclusions:

(1) When T is small, Rejection sampling is the best method.

(2) When T is large, Uniformization sampling is better.

5.2 Situation 2

Second, I tested the situation that the probability of transition to one state is very large:

$$Q = \frac{1}{s}\begin{bmatrix} -11 & 10 & 1 \\ 1 & -2 & 1 \\ 1 & 10 & -11 \end{bmatrix}$$

where s=3.5.

Its stationary distribution is:

$$\pi = (\frac{1}{12}, \frac{10}{12}, \frac{1}{12})$$

From the instantaneous rate matrix Q, we can find in this stochastic process, the probability of transition to state b is extremely larger than the probability of transition to other states. In the stationary distribution, the probability of ending in state b is ten times of the probability of ending in other states.

From figure 4 and figure 5(in appendix), we can draw conclusions:

(1) When the ending state is not state b(the probability of transition to state b is extremely large), Uniformization sampling is the best method.

(2) When the ending state is state b, Rejection sampling is better.

(3) The length of T doesn't affect conclusions above.

5.3 Situation 3

Third, I tested the situation that the probability of transition from one state is very large:

$$Q = \frac{1}{s}\begin{bmatrix} -20 & 10 & 10 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

where s=20/7.

Its stationary distribution is:

$$\pi = (\frac{1}{21}, \frac{10}{21}, \frac{10}{21})$$

From the instantaneous rate matrix Q, we can find in this stochastic process, the probability of transition from state a is extremely larger than the probability of transition from other states. In the stationary distribution, the probability of ending in state b and is ten times of the probability of ending in state a.

From figure 6 and figure 7(in appendix), we can draw conclusions:

(1) When the ending state is not state a(the probability of transition from state a is extremely large), Rejection sampling is the best method.

(2) When the ending state is state a, Uniformization sampling is better.

(3) The length of T doesn't affect conclusions above.

5.4 Situation 4

Forth, I tested the situation that the probability of transition between a pair of states is very large:

$$Q = \frac{1}{s}\begin{bmatrix} -11 & 10 & 1 \\ 10 & -11 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

where s=8.

Its stationary distribution is:

$$\pi = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$$

From the instantaneous rate matrix Q, we can find in this stochastic process, the probability of transition between state a and state b is extremely larger than the probability of transition between other states. However the stationary distribution is the same as that in first state.

From figure 8 and figure 9(in appendix), we can draw conclusions:

(1) When T is small and the ending state is not state c(the probability of transition between state a and state b is extremely large), Rejection sampling is the best method.

(2) When T is small and the ending state is state c, it is hard to tell whether Rejection sampling or Uniformization sampling is better. But they are both better than Direct sampling.

(3) When T is large, Uniformization sampling is the best way

5.5 Situation 5

I tested some combination of the situations listed above. First is the situation that the probability of transition from and to one state is very large:

$$Q = \frac{1}{s}\begin{bmatrix} -20 & 10 & 10 \\ 10 & -11 & 1 \\ 10 & 1 & -11 \end{bmatrix}$$

where s=14.

Its stationary distribution is:

$$\pi = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$$

From the instantaneous rate matrix Q, we can find in this stochastic process, the probability of transition from state a and transition to state a is extremely larger than other probability.

From figure 10 and figure 11(in appendix), we can draw conclusions:

(1) When T is small, Rejection sampling is the best method.

(2) When T is large, Uniformization sampling is better.

5.6 Situation 6

This situation is that the probability of transition from one state and transition to another state is very large:

$$Q = \frac{1}{s}\begin{bmatrix} -20 & 10 & 10 \\ 1 & -2 & 1 \\ 1 & 10 & -11 \end{bmatrix}$$

Its stationary distribution is:

$$\pi = (\frac{2}{42}, \frac{35}{42}, \frac{5}{42})$$

From the instantaneous rate matrix Q, we can find in this stochastic process, the probability of transition from state a and transition to state b is extremely larger than other probability.

From figure 12 and figure 13(in appendix), we can draw conclusions:

(1) When the ending state is a(the probability of transition from state a is large), Uniformization sampling is the best method.

(2) When the ending state is b(the probability of transition to state a is large), Rejection sampling is better.

(3) The length of T doesn't affect conclusions above.

(4) When the ending state is state c, it is hard to tell whether Rejection sampling or Uniformization sampling is better. But they are both better than Direct sampling.

5.7 Situation 7

This situation is that the probability of transition between a pair of states and is the probability of transition from one of them is very large:

$$Q = \frac{1}{s} \begin{bmatrix} -20 & 10 & 10 \\ 10 & -11 & 1 \\ 1 & 1 & -2 \end{bmatrix}$$

Its stationary distribution is:

$$\pi = (\frac{2}{42}, \frac{5}{42}, \frac{35}{42})$$

From the instantaneous rate matrix Q, we can find in this stochastic process, the probability of transition between state a and state b is large while the probability of transition from state a is large, too.

From figure 14 and figure 15(in appendix), we can draw conclusions:

(1) When T is small, Rejection sampling is the best method.

(2) When T is large and the ending state c, Rejection sampling is still the best method.

(3) When T is large, Uniformization sampling is better.

6 Conclusion

Firstly, based on the basic knowledge back ground of CTMC, we distinguished CTMC from DTMC, especially the definition of transition probability matrix P which we call instantaneous rate matrix Q in CTMC. Then, we focused on simulating continuous sample path of a certain kind of CTMC whose endpoints, time interval T and instantaneous rate matrix Q is given. To achieve this goal, we mainly discussed three existing method:

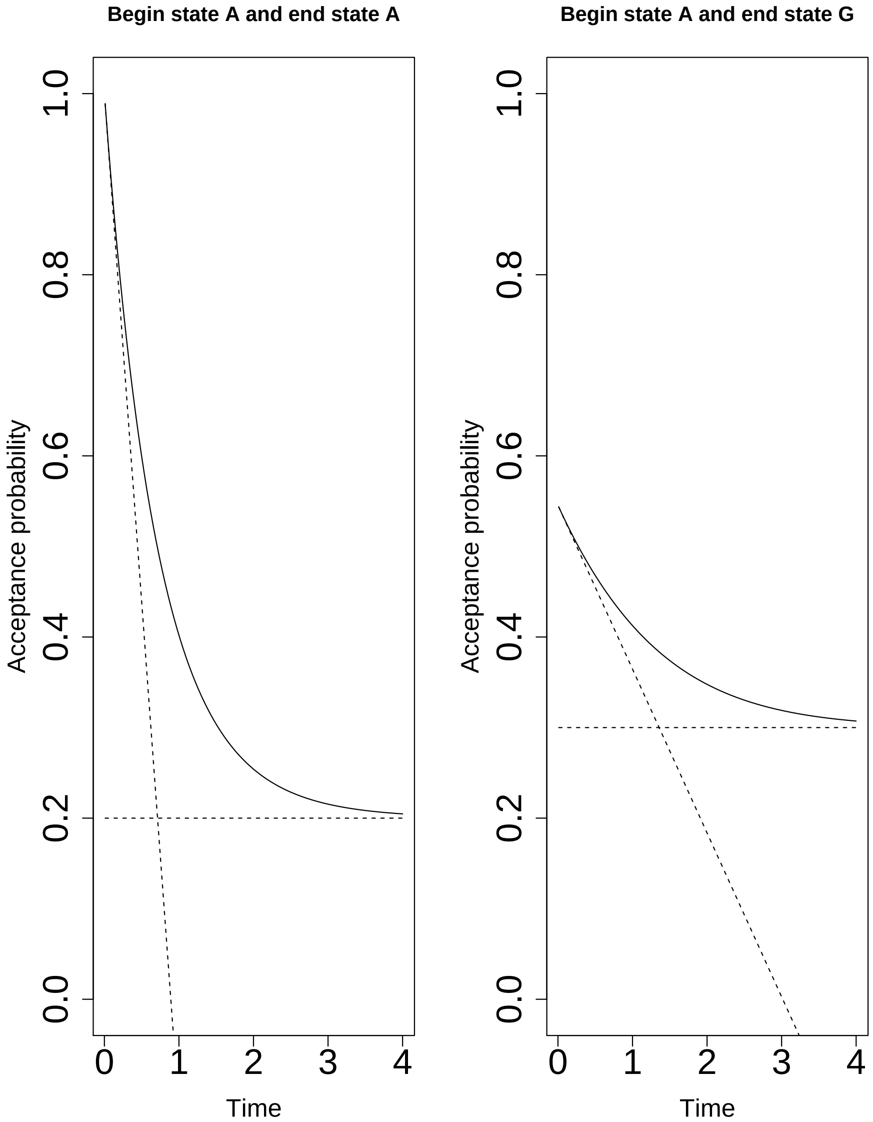
(1) Modified rejection sampling: but when transition from a to b is unlikely, rejection sampling is still not efficient enough.

(2) Direct sampling: require the matrix Q admits an eigenvalue decomposition.

(3) Uniformization.

Considering that different method could be suit for different length of time interval T, endpoints and matrix Q, the work of our project mainly processed those methods under different situations. Thus, we constructed seven different instantaneous rate matrix Q which stand for seven different situations, and most of them are strange which means extremity. Those situations can be classified to common situation, the performance of one situation is strange, and the combination of strange situations. After weve got the simulation results, we found out that when the length of time interval T matter, rejection sampling is always the best method for small T and uniformization suits for large T. If the length of T doesn't affect conclusions, the ending state will tell which method seems to be the best. Based on our programming result, we can say the efficiency is a relative measure that depends heavily on the properties of the conditioned stochastic, and no one algorithm performs best under all condition. Thus, the proper choice of sampling strategy is crucial in practice. Fortunately, since weve got seven basic situations results, its easy to choose a proper method by comparing with those seven situations.

A Appendix



Begin state A and end state A
Begin state A and end state G
Acceptance probability
Acceptance probability
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
0
1
2
3
4
Time
Time

Figure 1

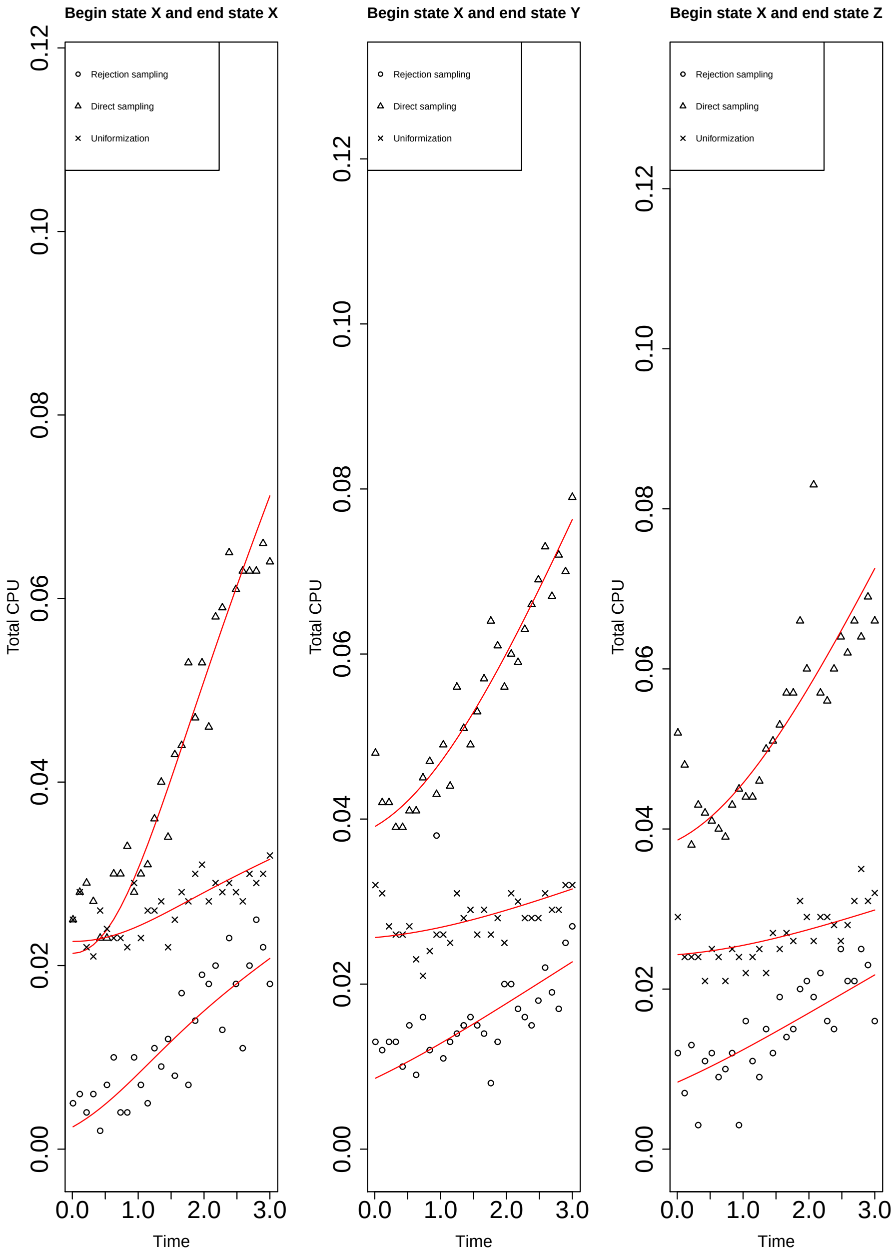
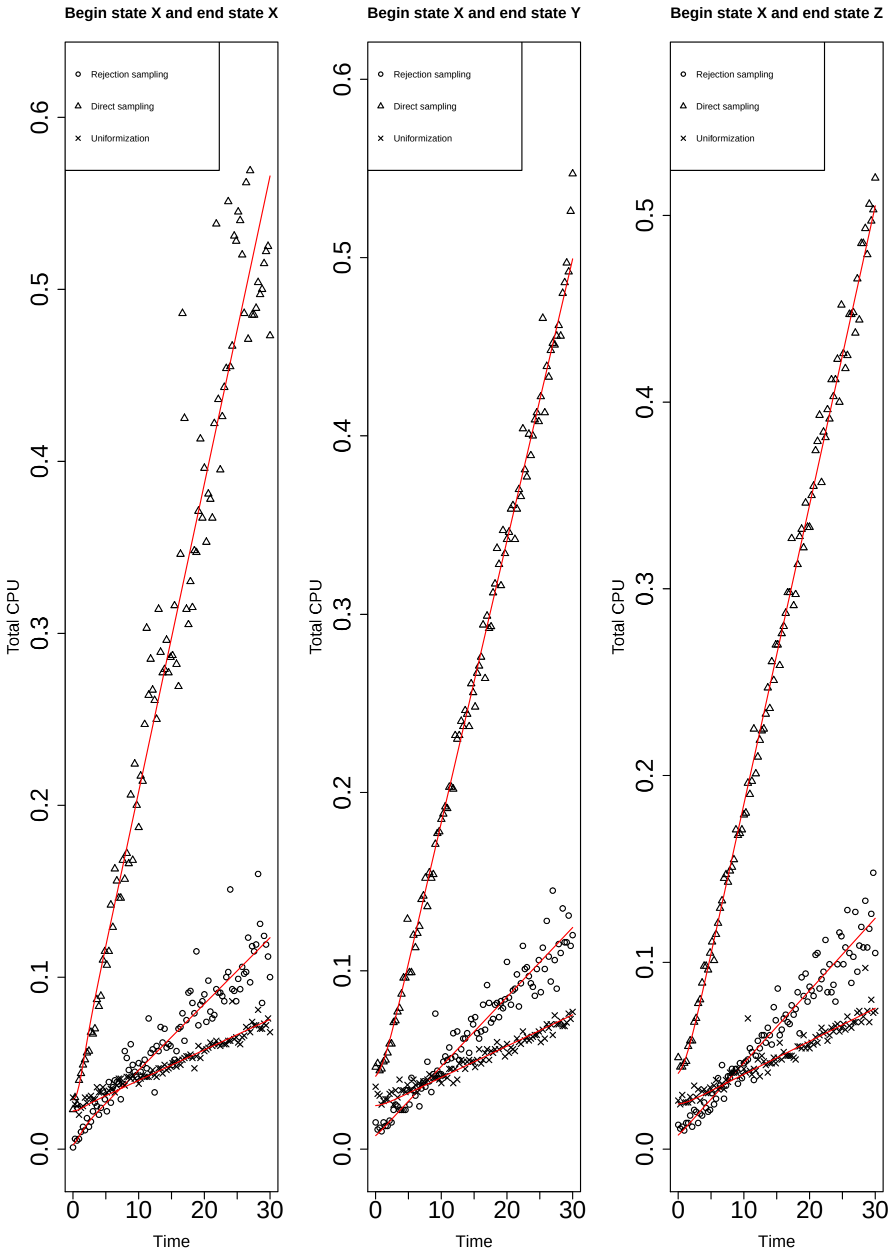

Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.0
1.0
2.0
3.0

Figure 2



Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30

Figure 3

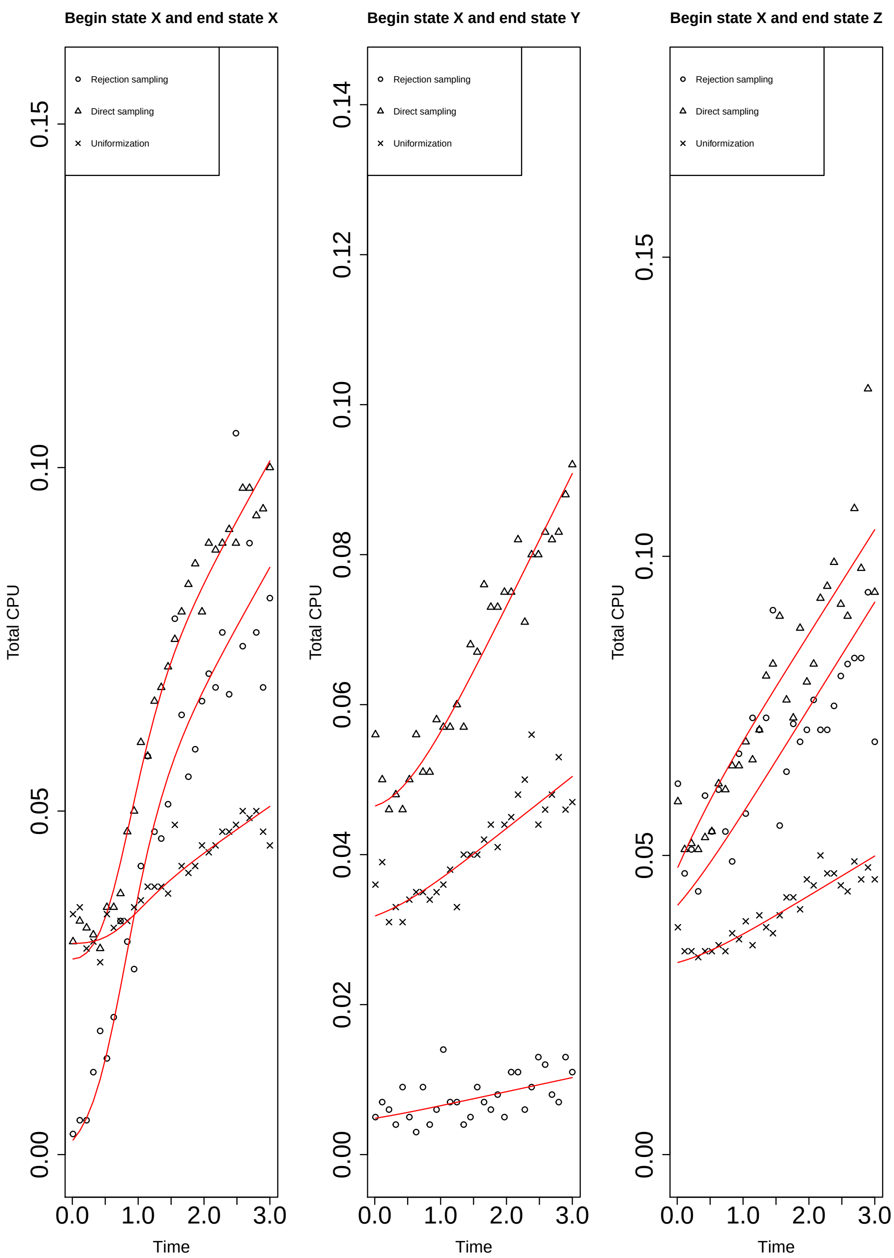

Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.00
0.05
0.10
0.15
0.02
0.04
0.06
0.08
0.12
0.14
0.0
1.0
2.0
3.0

Figure 4

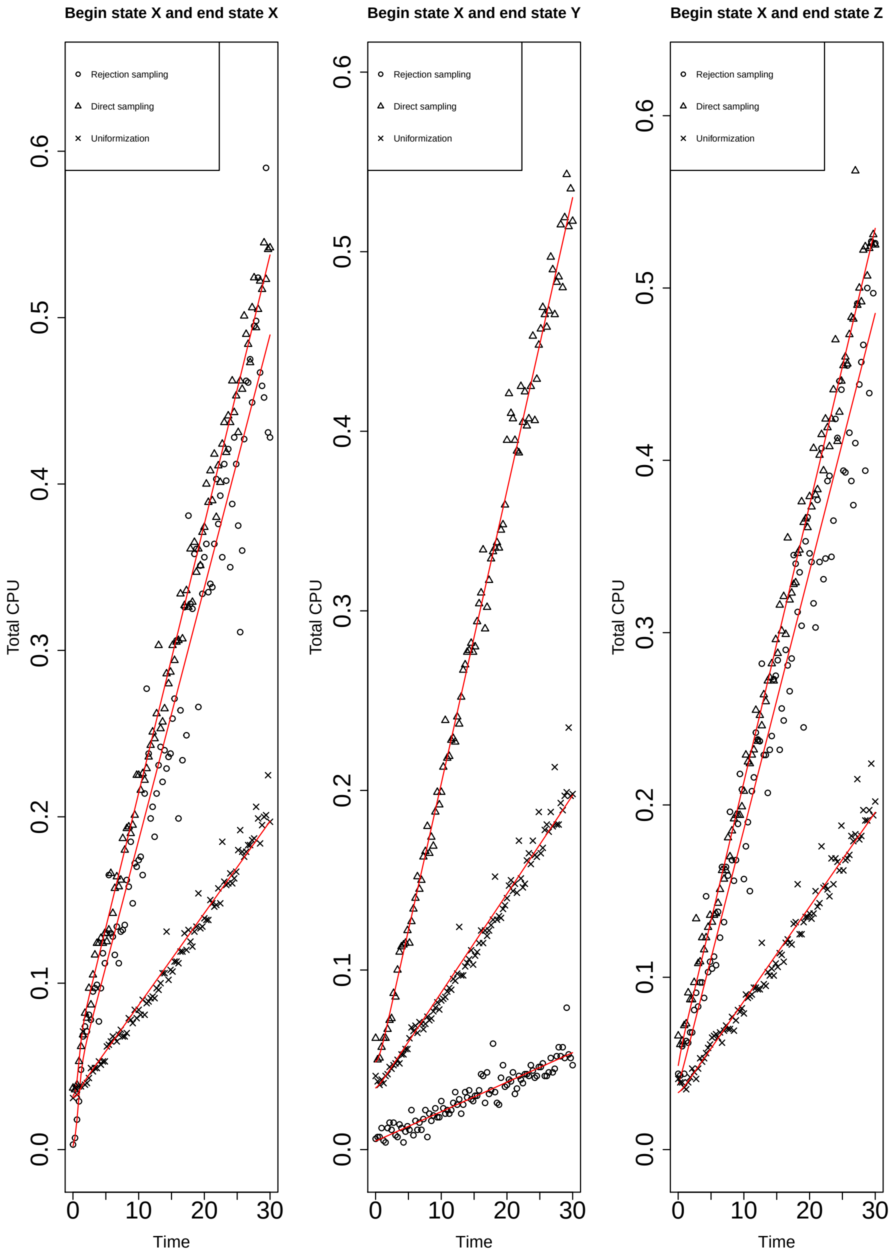
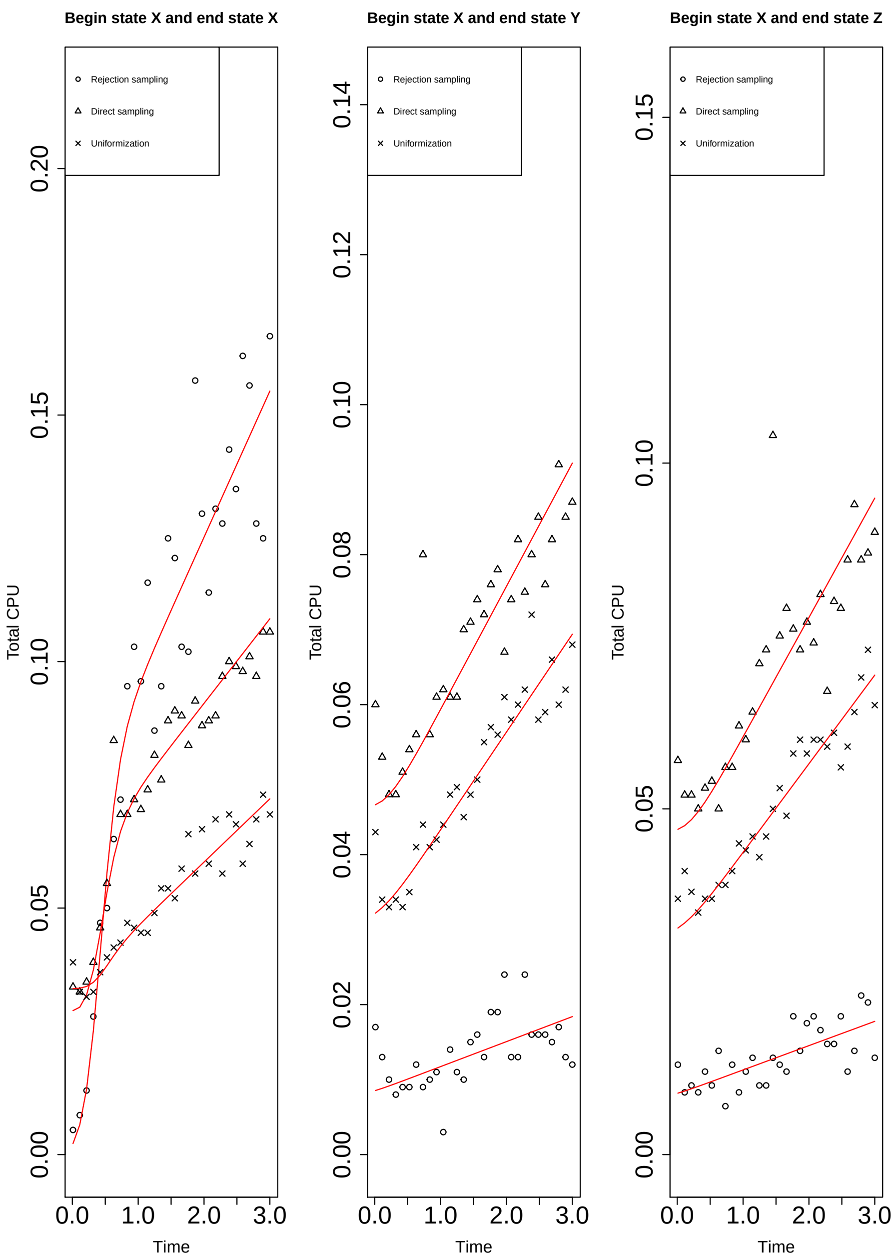
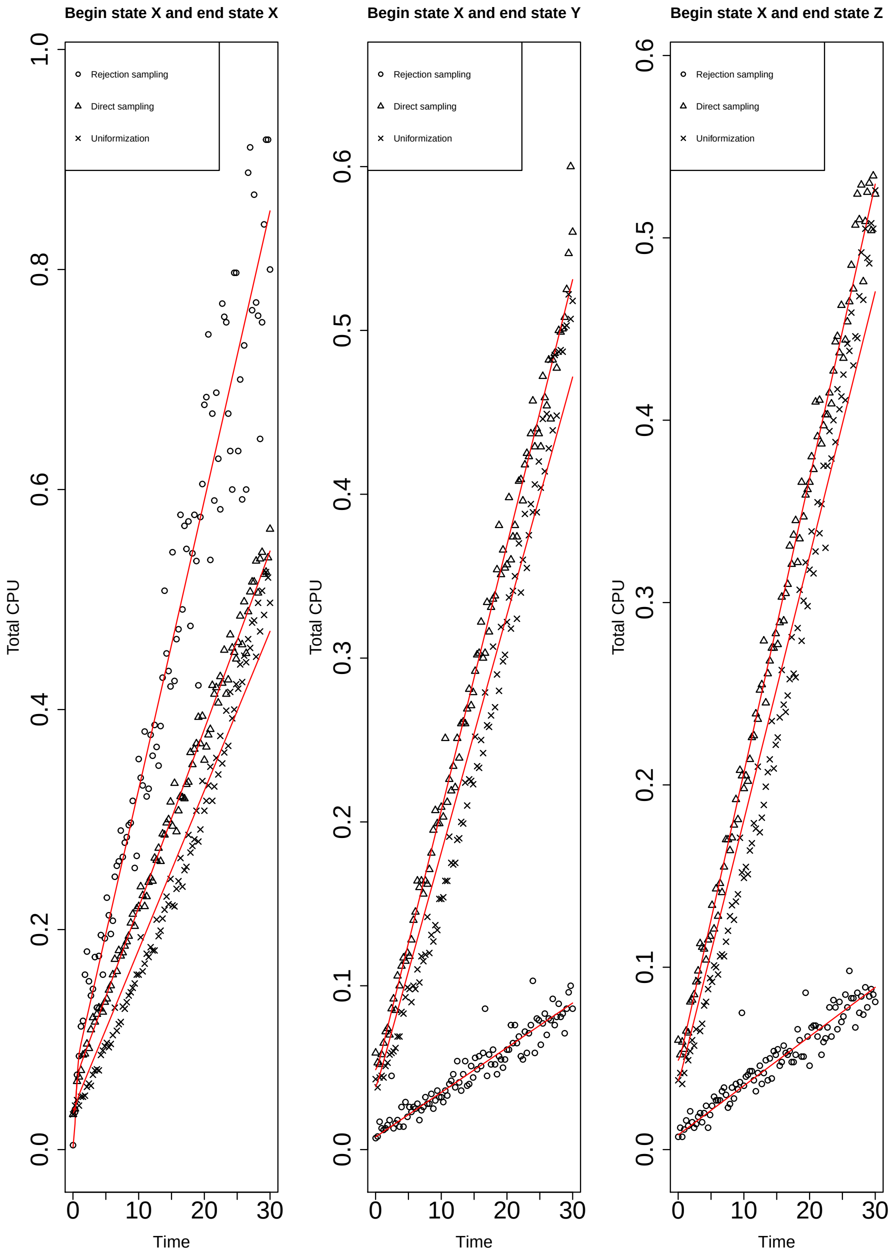

Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30

Figure 5



Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.00
0.05
0.10
0.15
0.20
0.02
0.04
0.06
0.08
0.12
0.14
0.0
1.0
2.0
3.0

Figure 6



Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time

Figure 7

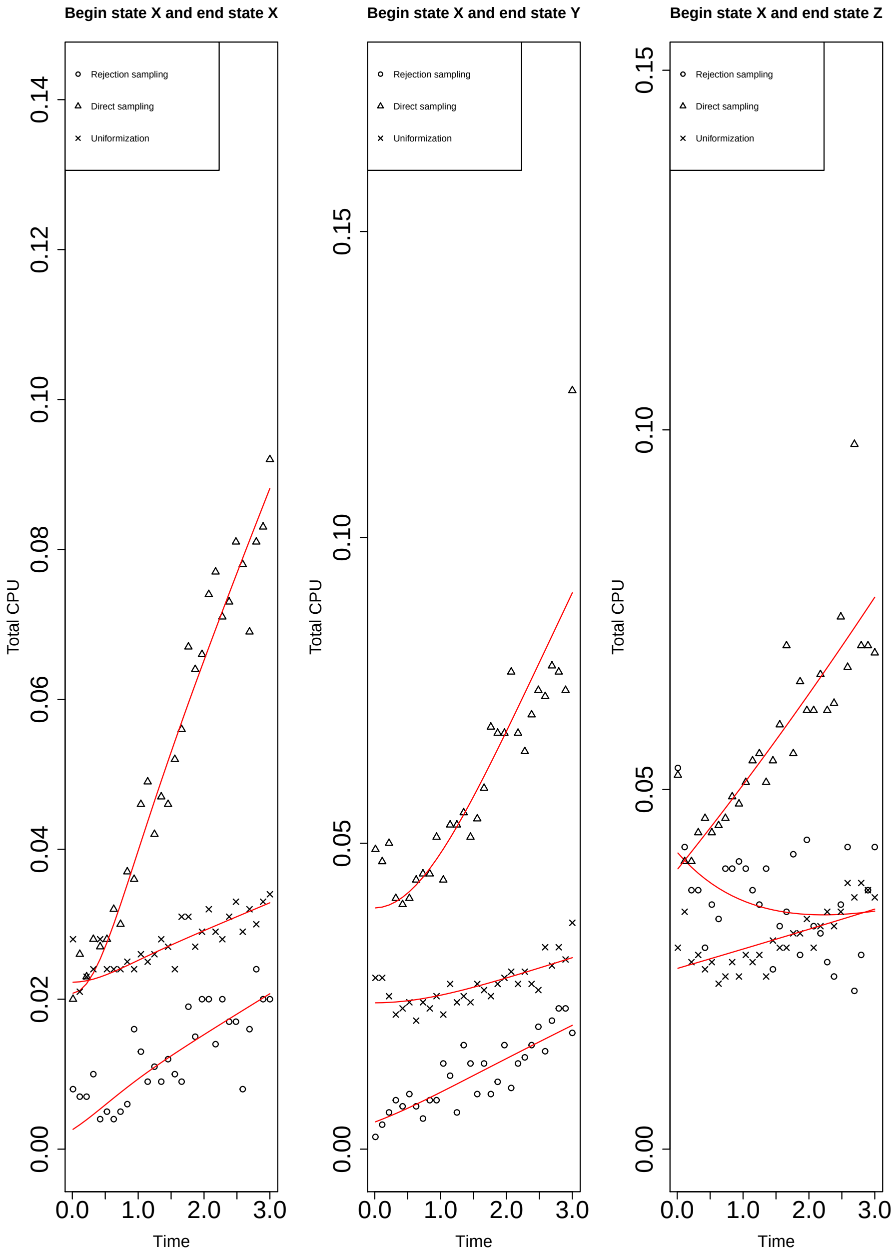
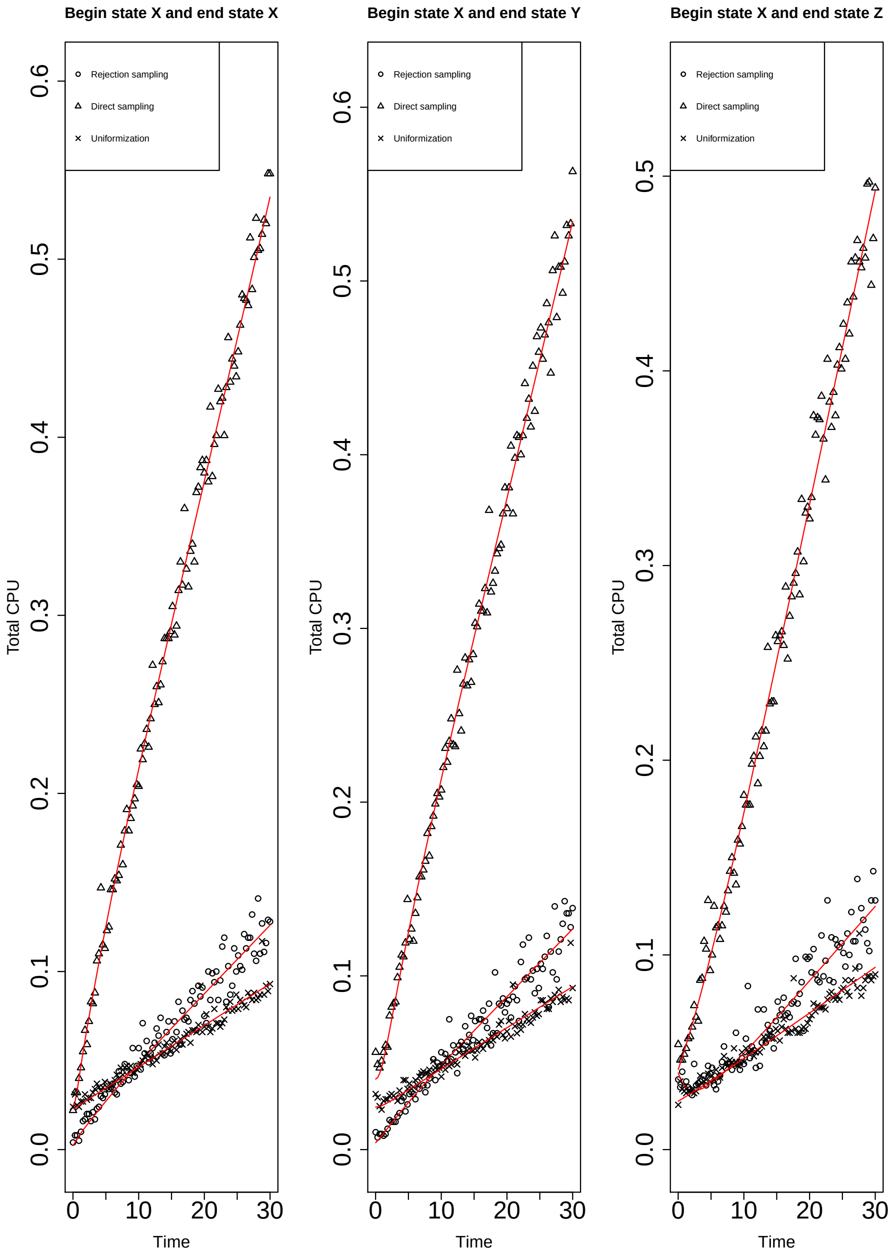

Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.05
0.15
0.0
1.0
2.0
3.0

Figure 8



Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30

Figure 9

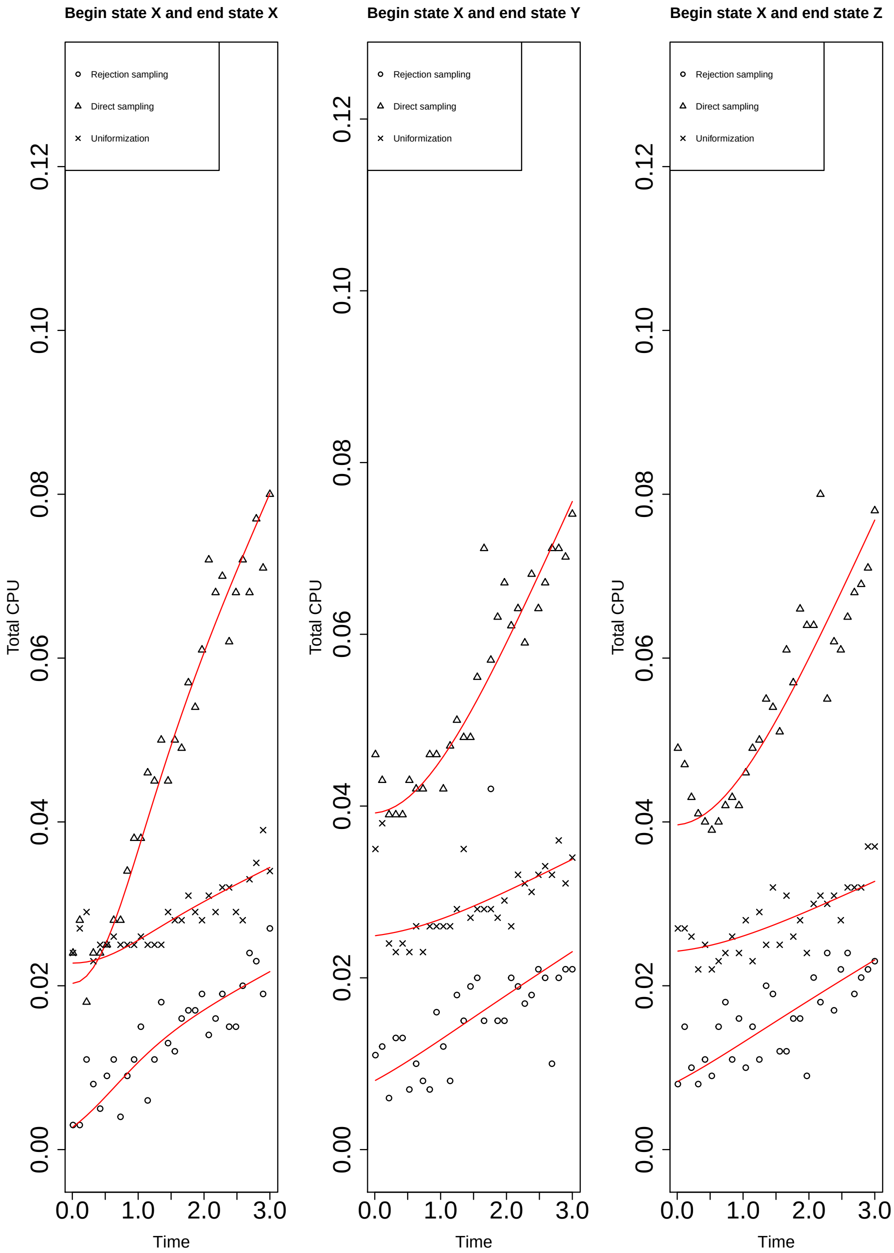
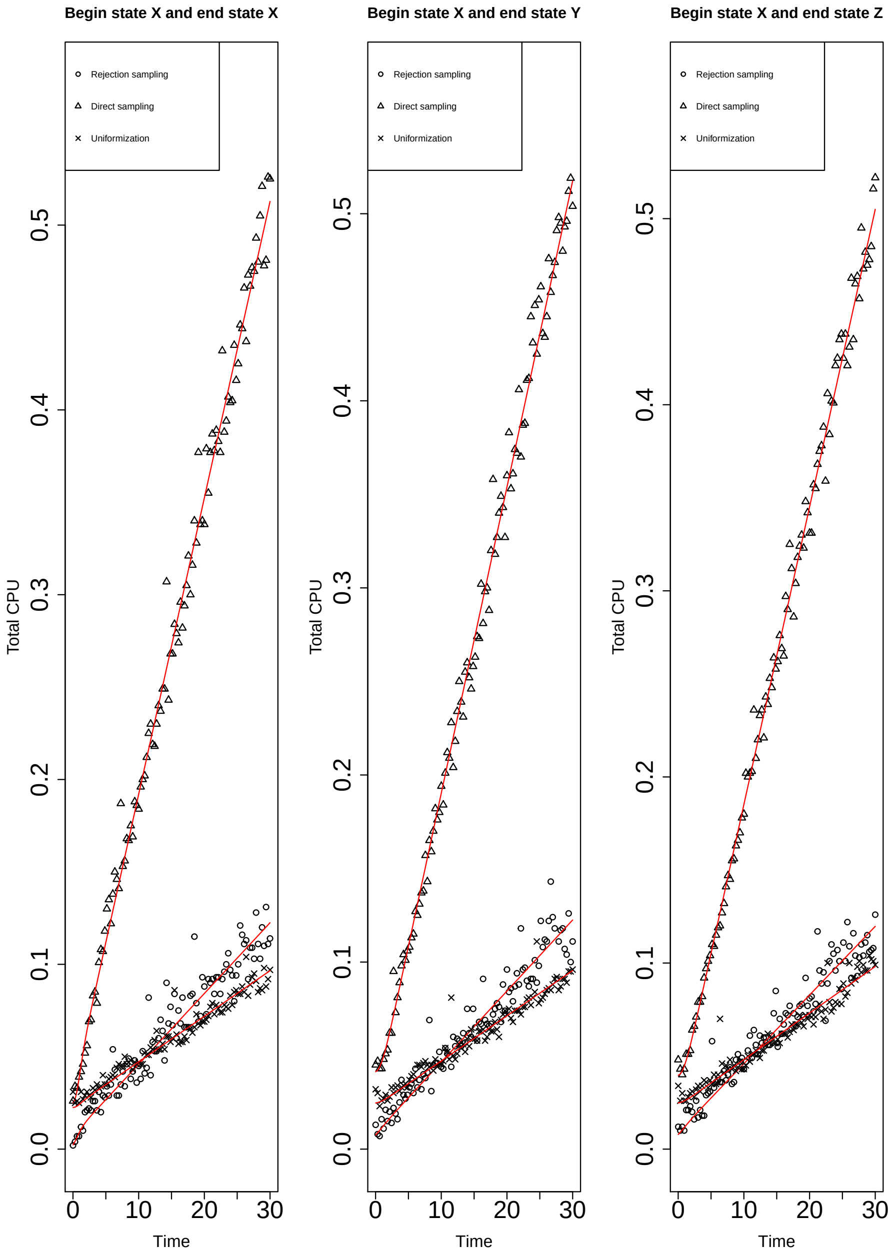
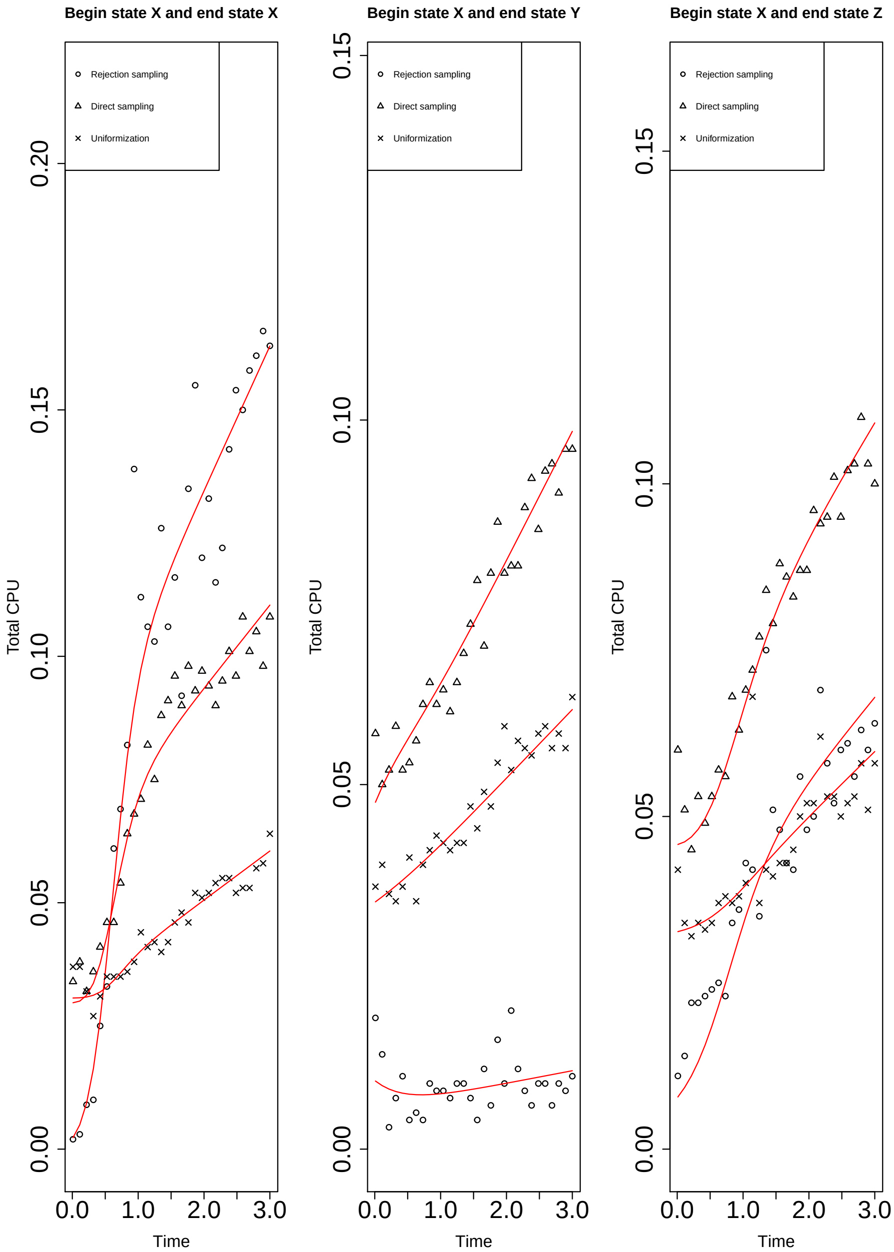

Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.0
1.0
2.0
3.0
Time

Figure 10



Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.0
0.1
0.2
0.3
0.4
0.5
0
10
20
30

Figure 11



Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.00
0.05
0.10
0.15
0.20
0.0
1.0
2.0
3.0

Figure 12

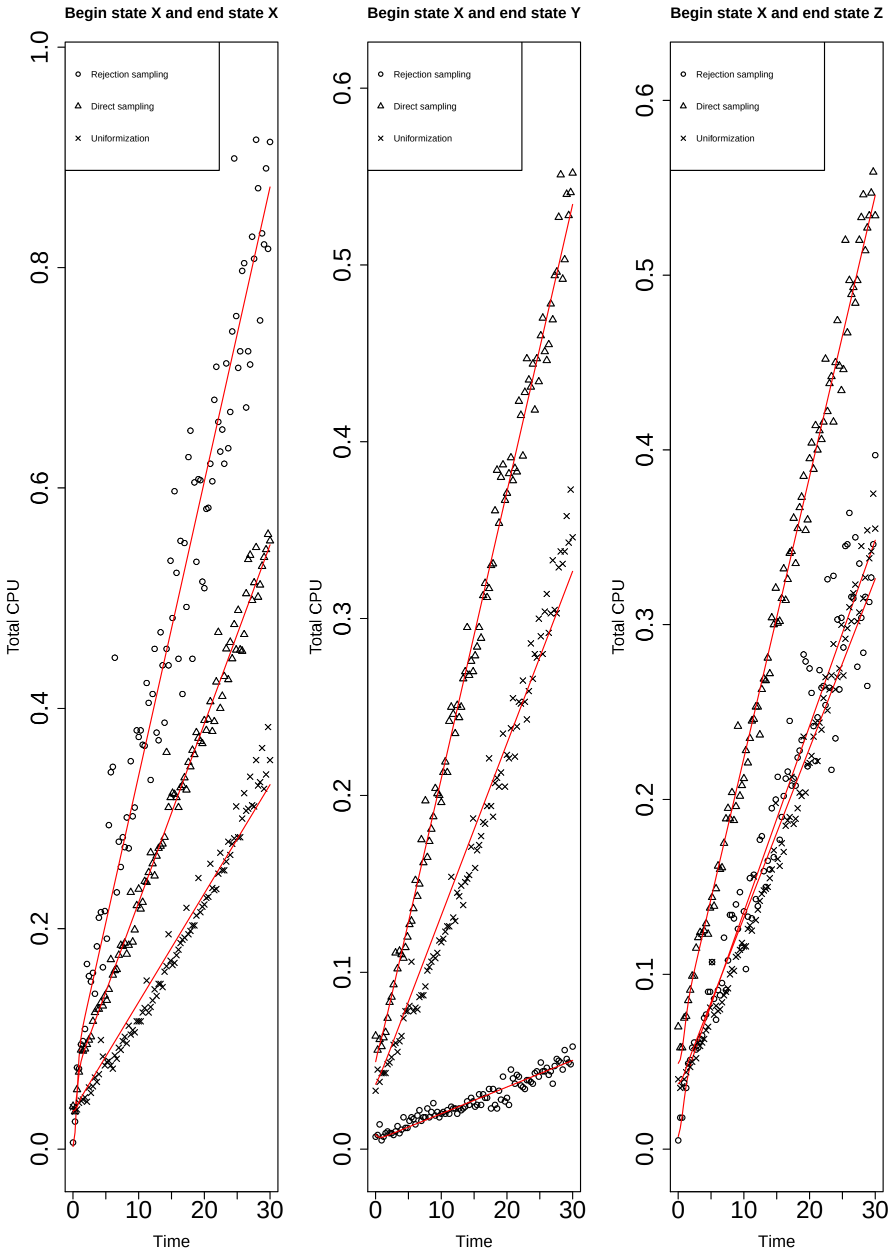
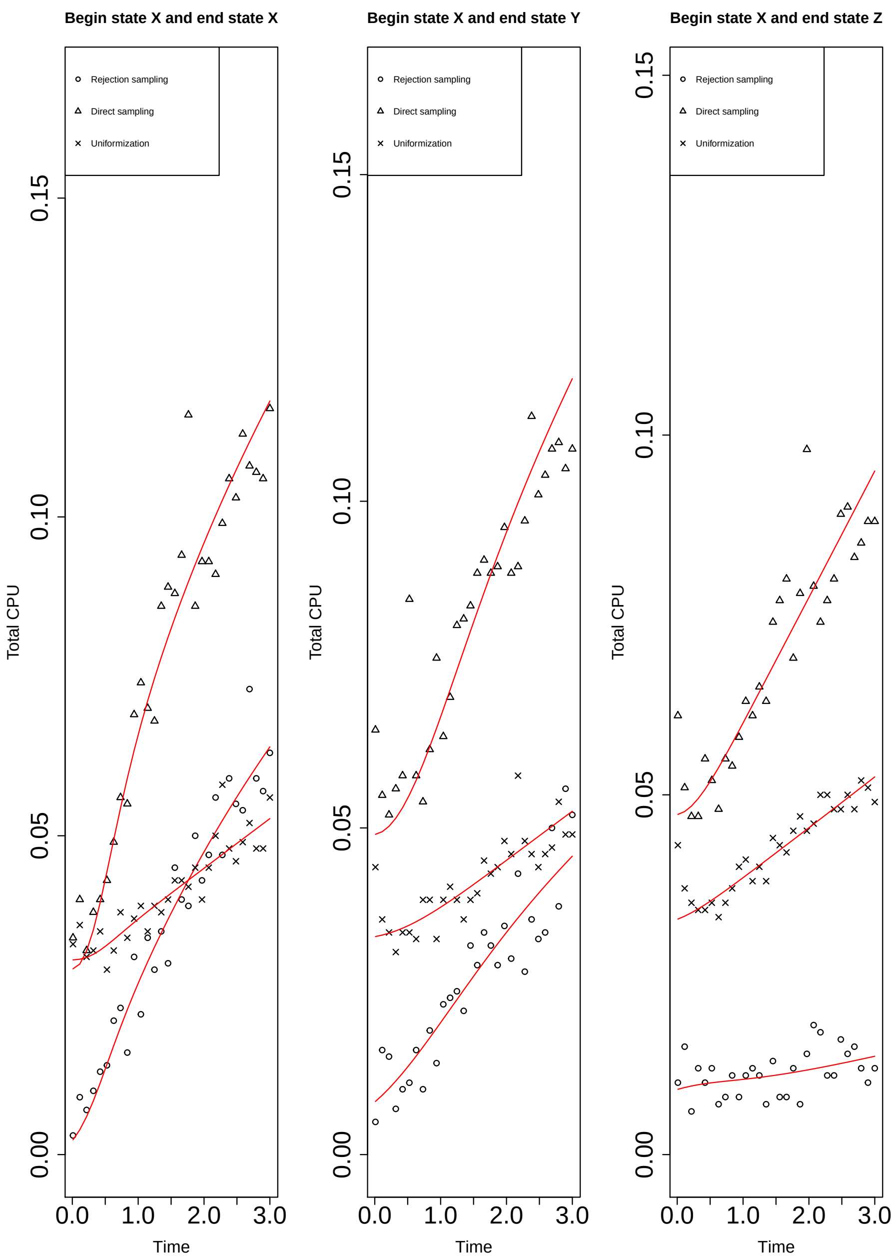

Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time

Figure 13



Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.00
0.05
0.10
0.15
0.0
1.0
2.0
3.0

Figure 14

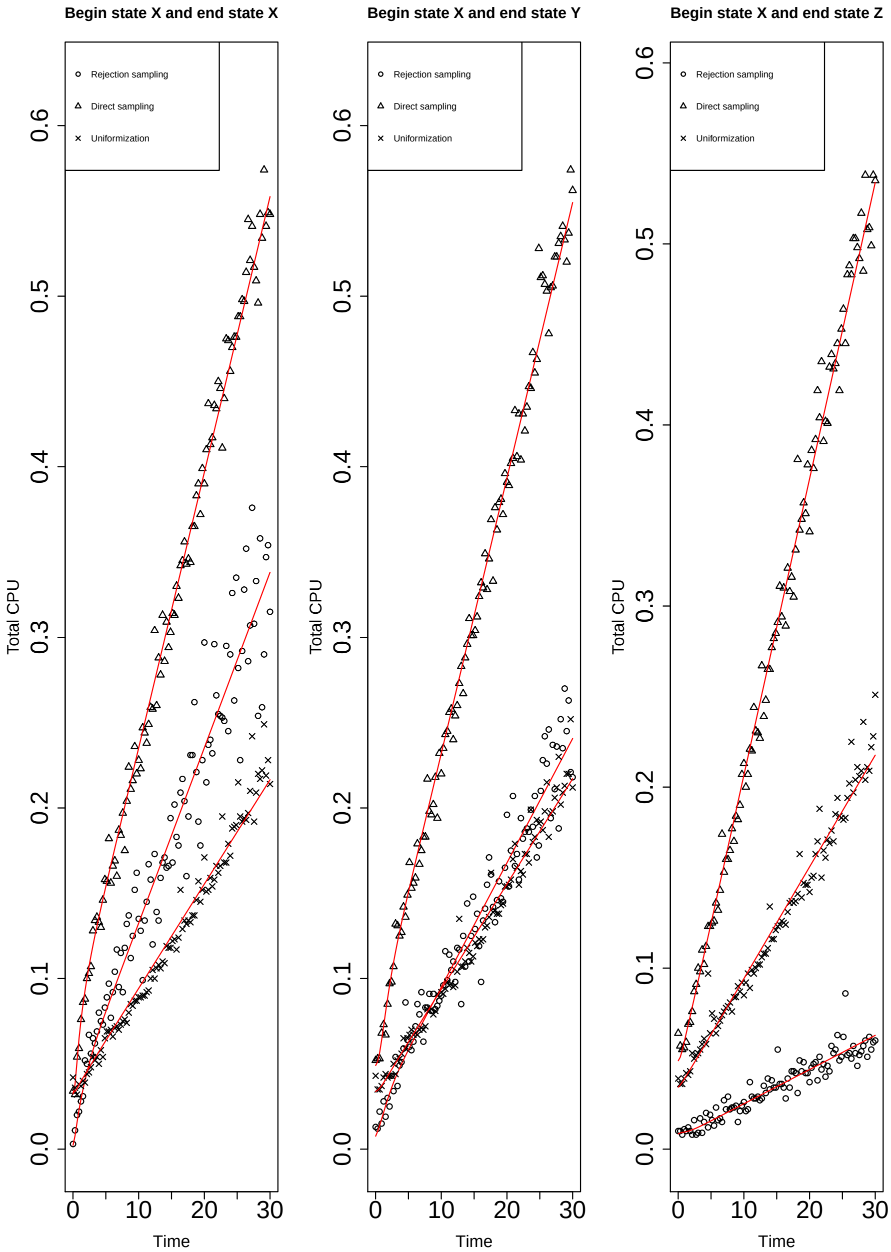

Begin state X and end state X
Begin state X and end state Y
Begin state X and end state Z
Rejection sampling
Direct sampling
Uniformization
Total CPU
Time
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
10
20
30

Figure 15