International Journal of Statistical Sciences *ISSN 1683-5603*
Vol. 20(2), 2020, pp 5-36

Rejection Sampling Scheme for Simulating from Multivariate Normal Density

Syeda Fateha Akter[1] and Anamul Haque Sajib[2*]

Department of Statistics, University of Dhaka, Dhaka-1000, Bangladesh
[*]Correspondence should be addressed to Anamul Haque Sajib
(Email: sajibstat@du.ac.bd)

[Received June 15, 2020; Accepted August 24, 2020]

Abstract

The multivariate normal (MVN) distribution is often considered to be the underlying distribution of many observed samples for modelling purposes, and hence simulation from this distribution is required to verify the fitted model. The decomposition based approach is currently being used to simulate sample from MVN distribution whose building block is Cholesky or Eigen decomposition. Although the decomposition approach is routinely used to generate MVN in almost all statistical packages *(R, SAS, Stata)*, but this approach may have a numerical issue (Ripley, 1987). Unfortunately, there is no other alternative of this approach to generate sample from MVN density. Motivated by this problem, we develop an alternative method to generate sample from MVN density whose building block is rejection sampling. Through simulation study, we demonstrate the validity and efficiency of the proposed method.

Keywords: Gibbs sampling, Ratio-of-Uniforms, Rejection sampling.

AMS Classification: 62D05.

1. Introduction

MVN density is one of the most widely used distributions in Statistics, Computer science and other discipline as an underlying distribution of many observed samples for modelling purpose. Therefore, there is a need to simulate sample from MVN density to justify their consideration as an underlying distribution of observed sample. For example, suppose we have observed sample $\boldsymbol{X} = (x_{i1}, x_{i2}, \ldots, x_{id}), i = 1, 2, \ldots, n$, from d dimensional normal density with mean vector,

$\boldsymbol{\mu} = \boldsymbol{\mu_0}$ and variance covariance matrix, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma_0}$, and we wish to estimate $\boldsymbol{\mu_0}$ and $\boldsymbol{\Sigma_0}$ in Bayesian approach. For simplicity, we consider, $\boldsymbol{\Sigma_0}$ is known here and we need to estimate $\boldsymbol{\mu_0}$ only. Choosing d dimensional normal density with mean vector a_0 and variance covariance matrix B_0 (which are known) as a prior density for unknown mean vector $\boldsymbol{\mu}$ yields a posterior distribution of $\boldsymbol{\mu}$ (likelihood × prior distribution of), which is also d dimensional normal density. The parameters of the posterior density, d dimensional normal density, are the function of $\boldsymbol{X}, a_0$ and B_0. Point estimate of $\boldsymbol{\mu_0}$ can be obtained by taking mean of the generated sample (N draws) from the posterior density (d dimensional normal density). Currently, there is only one approach available in the literature for generating sample from multivariate normal density, which is decomposition based approach: Eigen and Cholesky. De- composition based approach is currently being widely used in almost all statistical packages (*R*, *SAS*, *Stata*) to generate sample from *d*-dimensional normal density.

However, Eigen and Cholesky decomposition methods may get stuck (numerically instable) for a particular covariance matrix (Ripley, 1987). There is, therefore, a need for an alternative method which does not require any decomposition.

Motivated by the above problems, we develop an alternative method based on rejection sampling for generating sample from multivariate normal density. Our proposed method does not require any decomposition to simulate sample from MVN density.

We organize the rest of the paper as follows: Section 2 introduces some important terminologies used in this paper. Our proposed method for MVN generation is discussed in Section 3. Section 4 presents the simulation setting required to check the performance of our proposed method. The results and discussions of our simulation study are presented in Section 5. In the penultimate section, we present an application of MVN density which is followed by conclusion and future work presented in Section 7.

2. MVN and Related Terminologies

In this section, we introduce multivariate normal density and related terminologies required to generate from MVN. We here used the text book written by Johnson and Wichern, Robert and Casella and the paper written by Metropolis et al. to prepare the following overview of related terminologies necessary to generate sample from MVN (Johnson & Wichern, 2002; Casella & Burger, 2001 and Metropolis et al., 1953).

2.1. Multivariate Normal Distribution

The multivariate normal distribution is a generalization of univariate normal distribution to two or more variables. If the d-dimensional random vector, $\boldsymbol{X} = [X_1, X_2, X_3, \ldots, X_d]$ has mean vector, $\boldsymbol{\mu}$, and a symmetric positive definite covariance matrix, $\boldsymbol{\Sigma}$, then $\boldsymbol{X}$ has the following MVN density

$$f(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\},$$

where $-\infty < X_i < \infty, i = 1, 2\ \ldots, d.$

2.2. Markov Chain

Suppose we have two countable sets: one for state space $S = 1, 2, \ldots, r$ and other for time space $T = 1, 2, \ldots, n$. Then a discrete time and discrete state space Markov chain is a sequence $X_1,\ X_2, \cdots, X_{n-1},\ X_n$ of a random variable taking values in S at time T which follows the following Markov property $\Pr(X_n \in S|X_1 = x_1, \cdots, X_{n-1} = x_{n-1}) = \Pr(X_n \in S|X_{n-1} = x_{n-1})$ i.e. the probability of moving to the next state depends on its immediate state not the other previous states. If the state space S is continuous then we call the Markov chain discrete time continuous state space Markov chain. The state space S can be discrete or continuous or mix.

2.3. Ergodic Theorem

An aperiodic, irreducible Markov chain with transitional kernel Q and stationary distribution π is ergodic, so that the ergodic average $\bar{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(X_t) \to E_\pi[h(X)]$ as $n \to \infty$ where $E_\pi[h(X)] = \int h(x)\,\pi(x)\,dx$.

2.4. MCMC

MCMC methods provide a way of simulating random variables from an arbitrary density $\pi(\theta)$, where $\pi(\theta)$ needs only be known up to a normalizing constant, and θ can be a high dimensional. The basis for MCMC methods is the combination of convergence and ergodic properties of a Markov chain. The basic idea is: (i) to sample from distribution $\pi(\theta)$ simulate a Markov chain with stationary distribution π (ii) to estimate any function of density π use the ergodic average of the chain. Two most commonly used MCMC techniques are Metropolis-Hastings (MH) and Gibbs sampling algorithms, and both of these techniques can be used to simulate random variables from an arbitrary density (possibly multivariate) known up to normalizing constant. In this paper, we have considered only Gibbs sampling algorithm to simulate random sample from multivariate normal density.

2.5. Gibbs Sampling

Gibbs sampling method is particularly used to generate samples from high dimensional distributions which are mathematically intractable (known up to normality constant). Geman and Geman (1984) introduced Gibbs sampling to simulate sample from high dimensional distributions (Geman & Geman, 1984). Gibbs sampling works by deriving d number of full conditional distributions to simulate sample from a d-dimensional posterior density (Geman & Geman, 1984). Simulating from d number of full conditional distributions is same as simulating from a d-dimensional distribution. It is a special type of sampling where each proposal is accepted. A general Gibbs sampling method is discussed in Algorithm 1, where $X_1, X_2, X_3, \ldots, X_d$ be the random variates and their initial values are set to $x_1^{(0)}, x_2^{(0)}, \ldots, x_d^{(0)}$. However, the observations generated through Gibbs sampling are correlated with each other. Therefore, for independent sample special care must be taken, which is essential for valid inference.

Algorithm 2: Gibbs Sampling

Input: Initial values of the variates.

Output: Sample from the target density.

Begin

Set $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_d^{(0)})$

For $i = 1, 2, \cdots\cdots, n$ **do**

1. $X_1^{(i)} \sim p(X_1 | X_2 = x_2^{(i-1)},\ X_3 = x_3^{(i-1)}, \ldots, X_d = x_d^{(i-1)})$,
2. $X_2^{(i)} \sim p(X_2 | X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \ldots, X_d = x_d^{(i-1)})$,

$\vdots$

3. $X_d^{(i)} \sim p(X_d | X_1 = x_1^{(i)},, X_2 = x_2^{(i)},, \ldots, X_{d-1} = x_{d-1}^{(i)})$.

End For loop

End Begin

2.6. Ratio-of-Uniforms Method

Ratio-of-Uniforms (RoU) method invented by Wakefield et al. (1991) is a random variates generation technique to generate samples from an arbitrary probability density (Wakefield, Gelfand, & Smith, 1991). This method is based on acceptance-rejection framework though it does not require any proposal density to sample from an arbitrary probability density like the conventional acceptance-rejection method. Suppose our aim is to simulate from a d-dimensional density, $f(\boldsymbol{X}) = \frac{f_1(\boldsymbol{X})}{\int f_1(\boldsymbol{X})d\boldsymbol{X}} = \frac{f_1(\boldsymbol{X})}{C} \propto f_1(\boldsymbol{X})$, where $C = f_1(\boldsymbol{X})d\boldsymbol{X}$ is the normalizing constant. If $f_1(\boldsymbol{X})$ is a positive integrable function over χ, a subset of $\mathbb{R}^d$ and the variables $(U, V_1, \ldots, V_d)$ are uniformly distributed on $R = \left\{(U, V_1, \ldots, V_d): 0 < U \leq \left[f_1\left(\frac{V_1}{U}, \ldots, \frac{V_d}{U}\right)\right]^{\frac{1}{d+1}}\right\}$, then $\boldsymbol{X} = (X_1, X_2, \ldots, X_d)$, where $X_i = \frac{V_i}{U}$, has density $\frac{f_1(\boldsymbol{X})}{\int f_1(\boldsymbol{X})d\boldsymbol{X}} = f(\boldsymbol{X})$ (Wakefield, Gelfand, & Smith, 1991). The region R will be enclosed within a bounding d-dimensional rectangle subject to the condition that, $f_1(\boldsymbol{X})$ and ${x_i}^{d+1} f_1(\boldsymbol{X})$ are bounded over χ. For, $i = 1, 2\ \ldots, d$, Kinderman and Monahan (1977) used the rectangle $0 < u \leq a,\ b_i^- \leq v_i \leq b_i^+$, where (i) $a = sup_\chi [f_1(\boldsymbol{X})]^{\frac{1}{d+1}}$, (ii) $b_i^- = inf_{\chi_i^-} x_i [f_1(\boldsymbol{X})]^{\frac{1}{d+1}}$, $i = 1, 2\ \ldots, d$ and (iii) $b_i^+ = sup_{\chi_i^+} x_i [f_1(\boldsymbol{X})]^{\frac{1}{d+1}}$, $i = 1, 2\ \ldots, d$ with $\chi_i^- = \{\boldsymbol{x} \in \chi: x_i \leq 0\}$, $\chi_i^+ = \{\boldsymbol{x} \in \chi: x_i \geq 0\}$. The theoretical probability of acceptance will be the volume of R relative to that of the enclosing d-dimensional rectangle and it is given by, $P_a = \frac{\int f_1(\boldsymbol{X})d\boldsymbol{X}}{a(d+1)\prod_{i=1}^{d}(b_i^- - b_i^+)}$.

However, the target density can be symmetric or asymmetric. For symmetric unimodal densities Kinderman and Monahan showed that the probability of acceptance is maximized when mode of these densities is relocated to zero which is stated below in Theorem 1 (Kinderman & Monahan, 1977).

Theorem 1: Without loss of generality mode ($\boldsymbol{X} = \boldsymbol{\mu}$) of a positive symmetric function $f_1(\boldsymbol{X})$ defined on $\mathbb{R}$ can be rescaled to $\boldsymbol{X} = \vec{\boldsymbol{0}}$. Furthermore, provided that $sup_\chi [f_1(\boldsymbol{X})]^{\frac{1}{d+1}} < \infty$, then sampling from $f_1(\boldsymbol{X})$ is equivalent to sampling from $f_1(\boldsymbol{X} - \boldsymbol{\mu})$. Under these conditions, P_a is maximized when $\boldsymbol{\mu} = \vec{\boldsymbol{0}}$. The proof of the above theorem is not considered here but available in their paper. The procedure to simulate a sample of size n from d-dimensional density via RoU method is discussed in Algorithm 2.

Algorithm 2: Algorithm of RoU

Input: Constraints $a,\ b_i^-\ ,\ b_i^+$, where $i = 1, 2\ \ldots, d$.

Output: Sample from the target density.

Begin

  For $i = 1, 2, \cdots\cdots, n$ **do**

  1. Generate $U_1, U_2, \ldots, U_{d+1} \sim Uniform(0,1)$
  2. Calculate $U = a \times U_1, V_i = \ b_i^- + (\ b_i^+ -\ b_i^-) \times U_{i+1}$
  3. $\boldsymbol{If}\ \ U \leq \left[f_1\left(\frac{V_1}{U}, \ldots, \frac{V_d}{U}\right)\right]^{\frac{1}{d+1}}$ **then**
     - $X = \left(\frac{V_1}{U}, \ldots, \frac{V_d}{U}\right)$

  Else

  - Go back to step 1

  End If

  End For loop

End Begin

2.7. Modified Ratio-of-Uniforms Method

Wakefield et al. (1991) modified the basic version of ratio-of-uniforms method for the sake of increasing the efficiency in terms of acceptance rate of a point generated in the bounding rectangle (Wakefield, Gelfand, & Smith, 1991). In the modified ratio-of-uniforms (MRoU) method, a more general version of basic ratio-of-uniforms was proposed by introducing a new function g, which is strictly increasing differentiable function on $\mathbb{R}^+$ such that $g\ (0)\ =\ 0$. The more general version of basic ratio-of-uniforms method was proposed by Wakefield et al. (1991) which is stated in Theorem 2 (Wakefield, Gelfand, & Smith, 1991).

Theorem 2: For a strictly increasing differentiable function g defined on $\mathbb{R}^+$ such that g (0) = 0, if the joint density of (d+1) uniform random variables uniformly distributed on $R = \left\{(U, V_1, \ldots, V_d): 0 < U \leq g^{-1}\left[kf_1\left(\frac{V_1}{g'(U)}, \ldots, \frac{V_d}{g'(U)}\right)\right]\right\}$, where k > 0 is a constant while $g'$and g^{-1} are the first derivative of the function g and its inverse function respectively, then $\boldsymbol{X} = (X_1, X_2, \dots, X_d)$, where $X_i = \frac{V_i}{g'(U)}$ has density $f(\boldsymbol{X}) = \frac{f_1(\boldsymbol{X})}{C} \propto f_1(\boldsymbol{X})$. The region R will be enclosed within a bounding d-dimensional rectangle subject to the condition that, $f_1(\boldsymbol{X})$ and ${x_i}^{rd+1} f_1(\boldsymbol{X})^r$ are bounded over χ. From the proof, it is observed that the MRoU method reduces to basic form of RoU method when $r = 1$. For $i = 1, 2\ \dots, d$, Kinderman and Monahan used the rectangle $0 < u \leq a$, $b_i^- \leq v_i \leq b_i^+$, where (i) $a(r) = sup_{\chi}[f_1(\boldsymbol{X})]^{\frac{1}{rd+1}}$, (ii) $b_i^-(r) = inf_{\chi_i^-} x_i [f_1(\boldsymbol{X})]^{\frac{r}{rd+1}}$, and (iii) $b_i^+(r) = sup_{\chi_i^+} x_i [f_1(\boldsymbol{X})]^{\frac{r}{rd+1}}$ with $\chi_i^- = \{\boldsymbol{x} \in \chi : x_i \leq 0\}$, $\chi_i^+ = \{\boldsymbol{x} \in \chi : x_i \geq 0\}$, $i = 1, 2\ \dots, d$ (Kinderman & Monahan, 1977).

Algorithm 3: Algorithm of MRoU

Input: Constraints $a(r)$, $b_i^-(r)$, $b_i^+(r)$ where $i = 1, 2\ \dots, d$.

Output: Sample from the target density.

Begin

For $i = 1, 2, \cdots \cdots, n$ **do**

1. Generate $U_1, U_2, \dots, U_{d+1} \sim Uniform(0,1)$
2. Calculate $U = a(r) \times U_1, V_i = b_i^-(r) + (b_i^+(r) - b_i^-(r)) \times U_{i+1}$
3. **If** $U \leq \left[f_1\left(\frac{V_1}{U^r}, \dots, \frac{V_d}{U^r}\right)\right]^{\frac{1}{rd+1}}$ **then**
    - $X = \left(\frac{V_1}{U^r}, \dots, \frac{V_d}{U^r}\right)$

   Else
    - Go back to step 1

   End If

End For loop

End Begin

The theoretical probability of acceptance will be the volume of R relative to that of the enclosing d-dimensional rectangle and it is given by,

$$P_a = \frac{\int f_1(\boldsymbol{X}) d\boldsymbol{X}}{(rd + 1) a(r) \prod_{i=1}^{d} ((b_i^-(r) - b_i^+(r))}.$$

This acceptance probability is a function of r as all the quantities $a(r)$, $b_i^-(r)$ and $b_i^+(r)$ depend on r. Thus, P_a needs to be maximized with respect to r to get a high acceptance rate, which is the measure of efficiency of an acceptance-rejection algo- rithm. Finally, relocation of the distribution by the mode is suggested by Wakefield et al. before optimizing P_a over r in the MRoU method, which yields higher accep- tance rate (Wakefield, Gelfand, & Smith, 1991). The procedure to simulate a sample of size n from d-dimensional density via MRoU method is discussed in Algorithm 3.

2.8. Mardia's Test

Mardia's test proposed by Mardia (1970) can be used to check whether a sample data come from a multivariate normal distribution or not (Mardia, 1970). This test can be seen as a multivariate extension of skewness and kurtosis measures. The algorithm of this test is: (i) H_0 : the data come from multivariate normal against H_a : the data come from different distribution. (ii) The following quantities denote the test statistics of Mardia's test, $A = \frac{1}{6n}\sum_{i-1}^{n}\sum_{j-1}^{n}\left[(x_i-\bar{x})^T\hat{\Sigma}(x_i-\bar{x})\right]^3$, $B = \sqrt{\frac{n}{8d(d+2)}\left[\frac{1}{n}\sum_{i=1}^{n}\left\{(x_i-\bar{x})^T\hat{\Sigma}(x_i-\bar{x})\right\}^2 - d(d+2)\right]}$, where $\hat{\Sigma} = \frac{1}{n}\sum_{j=1}^{n}\hat{\Sigma}\left(x_j-\bar{x}\right)\left(x_j-\bar{x}\right)^T$. (iii) $A \sim \chi^2_{(h)}, h = \frac{1}{6}d(d+1)(d+2)$ and $B \sim N(0,1)$ approximately under H_0 and reject the null hypothesis if $A > \chi^2_{(1-\propto,h)}$, where $\chi^2_{(1-\propto,h)}$ is the $(1-\propto)^{th}$ upper quantile of the χ^2 distribution with h degrees of freedom and if $B > B_{\frac{\propto}{2}}$, where $B_{\frac{\propto}{2}}$ is the $(1-\frac{\propto}{2})^{th}$ upper quantile of the standard normal distribution.

3. Proposed Method for Multivariate Normal Generation

We have proposed a new approach in this section to generate random sample from MVN density. The building block of our proposed method is rejection sampling scheme: (i) Gibbs Sampling, (ii) Ratio-of-Uniforms (RoU) and (iii) Modified Ratio-of-Uniforms (MRoU). We call our method RSSMVN as rejection sampling scheme is used to generate sample from MVN. Furthermore, the RSSMVN is denoted as RSSMVNG, RSSMVNR and RSSMVNR* when it uses Gibbs, RoU and MRoU as its building block respectively. In this section we have discussed the methodology of our proposed RSSMVN and its implementation technique for $d = 2$ and $d = 3$.

3.1. BVN Generation via RSSMVN

This section discusses how RSSMVN generates sample from multivariate normal density when $d = 2$. The density function of BVN distribution is $f(\boldsymbol{X}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ = $\frac{1}{(2\pi)|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\}$, where $-\infty < \mu, X < \infty$ and the mean vector and covariance matrix are $\boldsymbol{\mu} = (\mu_1 \quad \mu_2)^T$ and $\boldsymbol{\Sigma} = \begin{pmatrix}\sigma_{11} & \sigma_{12}\\ \sigma_{12} & \sigma_{22}\end{pmatrix}$ respectively.

3.1.1. RSSMVNG Method

To generate sample from BVN density via RSSMVNG, it derives the conditional distributions of $X_1|X_2 = x_2$ and $X_2|X_1 = x_1$. All required calculations to derive the conditional distributions of $X_1|X_2 = x_2$ and $X_2|X_1 = x_1$ are shown in detail. Deriving full conditional distributions require the simplification of exponent part of BVN density which involves $\Sigma^{-1} = D^{-1}(\sigma_{22}\sigma_{11} - \sigma_{12}^2)$. After simplifying the exponent part, the density function $f(\boldsymbol{X}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ becomes

$$\frac{1}{(2\pi)|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\{-0.5D^{-1}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\}.$$

Considering $X_2 = x_2$ in $f(\boldsymbol{X}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ yields the full conditional distribution of X_1,

$$f(X_1|X_2 = x_2) = \frac{1}{(2\pi)|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\{-0.5D^{-1}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(x_2-\mu_2) + \sigma_{11}(x_2-\mu_2)^2]\} = \frac{1}{(2\pi)|\boldsymbol{\Sigma}|^{\frac{1}{2}}}K\exp\left\{\frac{\sigma_{22}}{2D}\left[X_1-\mu_1-\frac{\sigma_{12}(x_2-\mu_2)}{\sigma_{22}}\right]^2\right\},$$

where K is a constant involving x_2. After simplification, the full conditional distribution of $X_1|X_2 = x_2$ can be written as

$$X_1|X_2 = x_2 \sim N\left(\mu_1 + \frac{\sigma_{12}(x_2-\mu_2)}{\sigma_{22}}, \frac{D}{\sigma_{22}}\right).$$

Similarly, considering $X_1 = x_1$ yields the following full conditional distribution of $X_2|X_1 = x_1$,

$$X_2|X_1 = x_1 \sim N\left(\mu_2 + \frac{\sigma_{12}(x_1-\mu_1)}{\sigma_{11}}, \frac{D}{\sigma_{11}}\right).$$

The above procedures for generating sample from BVN density via RSSMVNG is summarized in Algorithm 4. Algorithm 4 needs to be run n times to get a sample of size n.

Algorithm 4: RSSMVNG for $d = 2$

Input: Initial values of X_1 and X_2.
Output: Sample from the BVN.

Begin

Set $x^{(0)} = (x_1^{(0)}, x_2^{(0)})$.

For $i = 1, 2, \cdots\cdots, n$ **do**

1. $X_1^{(i)} \sim N\left(\mu_1 + \frac{\sigma_{12}(x_2{}^{(i-1)} - \mu_2)}{\sigma_{22}}, \frac{D}{\sigma_{22}}\right)$
2. $X_2^{(i)} \sim N\left(\mu_2 + \frac{\sigma_{12}(x_1{}^{(i)} - \mu_1)}{\sigma_{11}}, \frac{D}{\sigma_{11}}\right)$

End For loop

End Begin

3.1.2. RSSMVNR Method

This section shows how the RSSMVNR generates sample from BVN density. The BVN probability density defined in Section 3.1. can be written as $f(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{f_1(\boldsymbol{X})}{C} \propto f_1(\boldsymbol{X})$, where $C = (2\pi)|\boldsymbol{\Sigma}|^{\frac{1}{2}}$ and $f_1(\boldsymbol{X}) = \exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1 - \mu_1)^2 - 2\sigma_{12}(X_1 - \mu_1)(X_2 - \mu_2) + \sigma_{11}(X_2 - \mu_2)^2]\right\}$. The following quantities need to be calculated to simulate sample from BVN via RSSMVNR:

(i) $a = sup_{\chi}[f_1(\boldsymbol{X})]^{\frac{1}{3}} = sup_{\chi}\left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1 - \mu_1)^2 - 2\sigma_{12}(X_1 - \mu_1)(X_2 - \mu_2) + \sigma_{11}(X_2 - \mu_2)^2]\right\}\right]^{\frac{1}{3}}$,

(ii) $b_1^- = inf_{\chi_1^-} X_1[f_1(\boldsymbol{X})]^{\frac{1}{3}} = inf_{\chi_1^-} X_1\left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1 - \mu_1)^2 - 2\sigma_{12}(X_1 - \mu_1)(X_2 - \mu_2) + \sigma_{11}(X_2 - \mu_2)^2]\right\}\right]^{\frac{1}{3}}$,

(iii) $b_1^+ = sup_{\chi_1^+} X_1 [f_1(\boldsymbol{X})]^{\frac{1}{3}} = sup_{\chi_1^+} X_1 \left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}\right]^{\frac{1}{3}}$,

(iv) $b_2^- = inf_{\chi_2^-} X_2 [f_1(\boldsymbol{X})]^{\frac{1}{3}} = inf_{\chi_2^-} X_2 \left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}\right]^{\frac{1}{3}}$ and

(v) $b_2^+ = sup_{\chi_2^+} X_2 [f_1(\boldsymbol{X})]^{\frac{1}{3}} = sup_{\chi_2^+} X_2 \left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}\right]^{\frac{1}{3}}$,

where $\chi_i^- = \{\boldsymbol{X} \in \chi : X_i \le 0\}$, $\chi_i^+ = \{\boldsymbol{X} \in \chi : X_i \ge 0\}$, $i = 1, 2$. Analytical calculation of the values of $a, b_1^-, b_1^+, b_2^-, b_2^+$ are difficult. Therefore, we have used 'genoud' function in R under 'rgenoud' package to calculate these values. The detail procedure of RSSMVNR to simulate sample from BVN is summarised in algorithm 5. Finally, we calculate the probability of acceptance (P_a) by plugging in the values of $a, b_1^-, b_1^+, b_2^-, b_2^+$ in $P_a = \frac{C}{3a\prod_{i=1}^{2}(b_i^- - b_i^+)}$.

Algorithm 5: RSSMVNR for $d = 2$

Input: Constraints $a, b_1^-, b_1^+, b_2^-, b_2^+$.
Output: Sample from the BVN.

Begin
For $i = 1, 2, \cdots\cdots, n$ **do**
1. Generate $U_1, U_2, U_3 \sim Uniform(0,1)$
2. Calculate $U = a \times U_1$, $V_1 = b_1^- + (b_1^+ - b_1^-) \times U_2$ and $V_2 = b_2^- + (b_2^+ - b_2^-) \times U_3$.
3. **If** $U \le \left[f_1\left(\frac{V_1}{U}, \frac{V_2}{U}\right)\right]^{\frac{1}{3}}$ **then**
   - $\boldsymbol{X} = \left(\frac{V_1}{U}, \frac{V_2}{U}\right)$

   Else
   - Go back to step 1

   End If

End For loop
End Begin

3.1.3. RSSMVNR* Method

This section shows how the RSSMVNR* generates sample from BVN density. The BVN probability density defined in Section 3.1. can be written as $f(\boldsymbol{X}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{f_1(X)}{C} \propto f_1(\boldsymbol{X})$, where $C = (2\pi)|\boldsymbol{\Sigma}|^{\frac{1}{2}}$ and $f_1(\boldsymbol{X}) = \exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}$. The following quantities need to be calculated to simulate sample from BVN via RSSMVNR*:

(i) $a(r) = sup_{\chi}[f_1(\boldsymbol{X})]^{\frac{1}{2r+1}} = sup_{\chi}\left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}\right]^{\frac{1}{2r+1}}$,

(ii) $b_1^-(r) = inf_{\chi_1^-}X_1[f_1(\boldsymbol{X})]^{\frac{r}{2r+1}} = inf_{\chi_1^-}X_1\left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}\right]^{\frac{r}{2r+1}}$,

(iii) $b_1^+(r) = sup_{\chi_1^+}X_1[f_1(\boldsymbol{X})]^{\frac{r}{2r+1}} = sup_{\chi_1^+}X_1\left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}\right]^{\frac{r}{2r+1}}$,

(iv) $b_2^-(r) = inf_{\chi_2^-}X_2[f_1(\boldsymbol{X})]^{\frac{r}{2r+1}} = inf_{\chi_2^-}X_2\left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}\right]^{\frac{r}{2r+1}}$ and

(v) $b_2^+(r) = sup_{\chi_2^+}X_2[f_1(\boldsymbol{X})]^{\frac{r}{2r+1}} = sup_{\chi_2^+}X_2\left[\exp\left\{-\frac{1}{2D}[\sigma_{22}(X_1-\mu_1)^2 - 2\sigma_{12}(X_1-\mu_1)(X_2-\mu_2) + \sigma_{11}(X_2-\mu_2)^2]\right\}\right]^{\frac{r}{2r+1}}$,

Where $\chi_i^- = \{\boldsymbol{X} \in \chi: X_i \leq 0\}$, $\chi_i^+ = \{\boldsymbol{X} \in \chi: X_i \geq 0\}$, $i = 1, 2$. Analytical calculation of the value of $a(r), b_1^-(r), b_1^+(r), b_2^-(r), b_2^+(r)$ is difficult. Therefore, we have used 'genoud' function in R under 'rgenoud' package to calculate these values. The detail procedure of RSSMVNR* to simulate sample from BVN is summarised in algorithm 6. Finally, we calculate the probability of acceptance(P_a) by plugging in the values of $a(r), b_1^-(r), b_1^+(r), b_2^-(r), b_2^+(r)$ in $P_a = \frac{C}{(2r+1)a(r)\prod_{i=1}^{2}(b_i^-(r) - b_i^+(r))}$. The value of r that maximizes the expression of P_a is the optimal value of r. Mathematical analysis reveals that $r = 0.5$ is optimal for multivariate normal case, regardless of dimension and covariance structure (Wakefield, Gelfand & Smith, 1991).

Algorithm 6: RSSMVNR* for $d = 2$

Input: Constraints $a(r), b_1^-(r), b_1^+(r), b_2^-(r), b_2^+(r)$.

Output: Sample from the BVN.

Begin

For $i = 1, 2, \cdots\cdots, n$ **do**

1. Generate $U_1, U_2, U_3 \sim Uniform(0,1)$
2. Calculate $U = a(r) \times U_1$, $V_1 = b_1^-(r) + (b_1^+(r) - b_1^-(r)) \times U_2$ and $V_2 = b_2^-(r) + (b_2^+(r) - b_2^-(r)) \times U_3$.
3. $\boldsymbol{If}$ $U \leq \left[f_1\left(\frac{V_1}{U^r}, \frac{V_2}{U^r}\right)\right]^{\frac{1}{2r+1}}$ **then**
   - $\boldsymbol{X} = \left(\frac{V_1}{U^r}, \frac{V_2}{U^r}\right)$

   Else

   - Go back to step 1

   End If

End For loop

End Begin

3.2. TVN Generation via RSSMVN

This section discusses how RSSMVN generates sample from multivariate normal density when $d = 3$. The density function of TVN distribution is $f(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$= $\frac{1}{(2\pi)^{3/2}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}$ exp$\{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\}$, where $-\infty < \mu, X < \infty$ and the mean vector and covariance matrix are $\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{pmatrix}$ respectively.

3.2.1. RSSMVNG Method

To generate sample from TVN density via RSSMVNG, it derives the conditional distributions of $(X_1|X_2 = x_2, X_3 = x_3)$, $(X_2|X_1 = x_{1,}X_3 = x_3)$ and $(X_3|X_1 = x_{1,}X_2 = x_2)$. All required calculations to derive the conditional distributions of $(X_1|X_2 = x_2, X_3 = x_3)$, $(X_2|X_1 = x_{1,}X_3 = x_3)$ and $(X_3|X_1 = x_{1,}X_2 = x_2)$ are shown in detail. Deriving full conditional distributions require the simplification of exponent part of TVN density which involves $D = \sigma_{11}(\sigma_{22}\sigma_{33} - {\sigma_{23}}^2) - \sigma_{12}(\sigma_{12}\sigma_{33} - \sigma_{13}\sigma_{23}) + \sigma_{13}(\sigma_{12}\sigma_{23} - \sigma_{13}\sigma_{22})$ and

$\Sigma^{-1} = \frac{1}{D}\begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{pmatrix}$ (say), where $p_{11} = \sigma_{22}\sigma_{33} - {\sigma_{23}}^2$, $p_{12} = \sigma_{13}\sigma_{23} - \sigma_{12}\sigma_{33}$, $p_{13} = \sigma_{12}\sigma_{23} - \sigma_{13}\sigma_{22}$, $p_{22} = \sigma_{11}\sigma_{33} - {\sigma_{13}}^2$, $p_{23} = \sigma_{12}\sigma_{13} - \sigma_{11}\sigma_{23}$ and $p_{33} = \sigma_{11}\sigma_{22} - {\sigma_{12}}^2$.

After simplifying the exponent part, the density function $f(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ becomes

$\frac{1}{(2\pi)^{3/2}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\{-0.5D^{-1}[p_{11}a_1^2 + p_{22}a_2^2 + p_{33}a_3^2 + 2p_{12}a_1a_2 + 2p_{13}a_1a_3 + 2p_{23}a_2a_3]\}$.

Here, $a_1 = X_1 - \mu_1, a_2 = X_2 - \mu_2$ and $a_3 = X_3 - \mu_3$. Considering $X_2 = x_2$ and $X_3 = x_3$ in $f(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ yields the full conditional distribution of X_1,

$$\begin{aligned} f(X_1|X_2 = x_2, X_3 = x_3) &= C \exp\left\{-\frac{1}{2D}[p_{11}a_1^2 + 2p_{12}a_1a_2 + 2p_{13}a_1a_3]\right\} \\ &= Cexp\left\{-\frac{p_{11}}{2D}\left[a_1^2 - 2a_1\left(\frac{-p_{12}}{p_{11}}a_2 + \frac{-p_{13}}{p_{11}}a_3\right)\right]\right\} \\ &= CK \exp\left\{-\frac{p_{11}}{2D}\left[a_1 - \left(\frac{-p_{12}}{p_{11}}a_2 + \frac{-p_{13}}{p_{11}}a_3\right)\right]^2\right\} \\ &= CK \exp\left\{-\frac{p_{11}}{2D}\left[X_1 - \mu_1 \right.\right. \\ &\qquad \left.\left. - \left(\frac{-p_{12}}{p_{11}}(x_2 - \mu_2) + \frac{-p_{13}}{p_{11}}(x_3 - \mu_3)\right)\right]^2\right\} \end{aligned}$$

where, K is a constant involving x_2 and x_3. After simplification, the full conditional distribution of $(X_1|X_2 = x_2, X_3 = x_3)$ can be written as

$$X_1|X_2 = x_2, X_3 = x_3 \sim N\left(\mu_1 - \frac{p_{12}(x_2-\mu_2)}{p_{11}} - \frac{p_{13}(x_3-\mu_3)}{p_{11}}, \frac{D}{p_{11}}\right).$$

Similarly, considering $X_1 = x_1$, $X_3 = x_3$ and $X_1 = x_1$, $X_2 = x_2$ yield the following full conditional distributions of $(X_2|X_1 = x_{1,}X_3 = x_3)$ and $(X_3|X_1 = x_{1,}X_2 = x_2)$ respectively,

$$X_2|X_1 = x_1, X_3 = x_3 \sim N\left(\mu_2 - \frac{p_{12}(x_1-\mu_1)}{p_{22}} - \frac{p_{23}(x_3-\mu_3)}{p_{22}}, \frac{D}{p_{22}}\right),$$

$$X_3|X_1 = x_{1,}X_2 = x_2 \sim N\left(\mu_3 - \frac{p_{13}(x_1-\mu_1)}{p_{33}} - \frac{p_{23}(x_2-\mu_2)}{p_{33}}, \frac{D}{p_{33}}\right).$$

All the steps discussed above is summarized in Algorithm 7 which needs to be run n times to get a sample of size n.

Algorithm 7: RSSMVNG for $d = 3$

Input: Initial values of X_1, X_2 and X_3.
Output: Sample from the TVN.

Begin

Set $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)})$.

For $i = 1, 2, \cdots\cdots, n$ **do**

1. $X_1^{(i)} \sim N\left(\mu_1 - \frac{p_{12}(x_2{}^{(i-1)}-\mu_2)}{p_{11}} - \frac{p_{13}(x_3{}^{(i-1)}-\mu_3)}{p_{11}}, \frac{D}{p_{11}}\right)$,
2. $X_2^{(i)} \sim N\left(\mu_2 - \frac{p_{12}(x_1{}^{(i)}-\mu_1)}{p_{22}} - \frac{p_{23}(x_3{}^{(i-1)}-\mu_3)}{p_{22}}, \frac{D}{p_{22}}\right)$,
3. $X_3^{(i)} \sim N\left(\mu_3 - \frac{p_{13}(x_1{}^{(i)}-\mu_1)}{p_{33}} - \frac{p_{23}(x_2{}^{(i)}-\mu_2)}{p_{33}}, \frac{D}{p_{33}}\right)$.

End For loop

End Begin

3.2.2. RSSMVNR Method

This section shows how the RSSMVNR generates sample from TVN density. The TVN probability density defined in Section 3.2. can be written as $f(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{f_1(\boldsymbol{X})}{C} \propto f_1(\boldsymbol{X})$, where $C = (2\pi)|\boldsymbol{\Sigma}|^{\frac{3}{2}}$ and $f_1(\boldsymbol{X}) = \exp\{-0.5D^{-1}[p_{11}a_1^2 + p_{22}a_2^2 + p_{33}a_3^2 + 2p_{12}a_1a_2 + 2p_{13}a_1a_3 + 2p_{23}a_2a_3]\}$.

Here, $a_1 = X_1 - \mu_1, a_2 = X_2 - \mu_2$, $a_3 = X_3 - \mu_3$, $p_{11} = \sigma_{22}\sigma_{33} - {\sigma_{23}}^2$, $p_{12} = \sigma_{13}\sigma_{23} - \sigma_{12}\sigma_{33}$, $p_{13} = \sigma_{12}\sigma_{23} - \sigma_{13}\sigma_{22}$, $p_{22} = \sigma_{11}\sigma_{33} - {\sigma_{13}}^2$, $p_{23} = \sigma_{12}\sigma_{13} - \sigma_{11}\sigma_{23}$, $p_{33} = \sigma_{11}\sigma_{22} - {\sigma_{12}}^2$ and $D = \sigma_{11}(\sigma_{22}\sigma_{33} - {\sigma_{23}}^2) - \sigma_{12}(\sigma_{12}\sigma_{33} - \sigma_{13}\sigma_{23}) + \sigma_{13}(\sigma_{12}\sigma_{23} - \sigma_{13}\sigma_{22})$.

Algorithm 8: RSSMVNR for $d = 3$

Input: Constraints $a, b_1^-, b_1^+, b_2^-, b_2^+, b_3^-, b_3^+$.
Output: Sample from the TVN.

Begin
For $i = 1, 2, \cdots\cdots, n$ **do**

1. Generate $U_1, U_2, U_3, U_4 \sim Uniform(0,1)$
2. Calculate $U = a \times U_1$, $V_1 = b_1^- + (b_1^+ - b_1^-) \times U_2$, $V_2 = b_2^- + (b_2^+ - b_2^-) \times U_3$ and $V_3 = b_3^- + (b_3^+ - b_3^-) \times U_4$
3. **If** $U \le \left[f_1\left(\frac{V_1}{U}, \frac{V_2}{U}, \frac{V_3}{U}\right)\right]^{\frac{1}{4}}$ **then**
   - $\boldsymbol{X} = \left(\frac{V_1}{U}, \frac{V_2}{U}, \frac{V_3}{U}\right)$

   Else
   - Go back to step 1

   End If

End For loop
End Begin

The following quantities need to be calculated to simulate sample from TVN via RSSMVNR:

(i) $a = sup_{\chi}[f_1(\boldsymbol{X})]^{\frac{1}{4}}$,

(ii) $b_1^- = inf_{\chi_1^-} X_1 [f_1(\boldsymbol{X})]^{\frac{1}{4}}$

(iii) $b_1^+ = sup_{\chi_1^+} X_1 [f_1(\boldsymbol{X})]^{\frac{1}{4}}$,

(iv) $b_2^- = inf_{\chi_2^-} X_2 [f_1(\boldsymbol{X})]^{\frac{1}{4}}$,

(v) $b_2^+ = sup_{\chi_2^+} X_2 [f_1(\boldsymbol{X})]^{\frac{1}{4}}$,

(vi) $b_3^- = inf_{\chi_3^-} X_3 [f_1(\boldsymbol{X})]^{\frac{1}{4}}$ and

(vii) $b_3^+ = sup_{\chi_3^+} X_3[f_1(\boldsymbol{X})]^{\frac{1}{4}}$,

where $\chi_i^- = \{X \in \chi : X_i \leq 0\}$, $\chi_i^+ = \{X \in \chi : X_i \geq 0\}$, $i = 1, 2, 3$. The detail procedure of RSSMVNR for simulating from TVN is summarised in algorithm 8.

Finally, we calculate the probability of acceptance(P_a) by plugging in the values of $a, b_1^-, b_1^+, b_2^-, b_2^+, b_3^-, b_3^+$ in $P_a = \frac{C}{4a\prod_{i=1}^{3}(b_i^- - b_i^+)}$.

3.2.3. RSSMVNR* Method

This section shows how the RSSMVNR* generates sample from TVN density. The TVN probability density defined in Section 3.2. can be written as $f(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{f_1(\boldsymbol{X})}{C} \propto f_1(\boldsymbol{X})$, where $C = (2\pi)|\boldsymbol{\Sigma}|^{\frac{3}{2}}$ and $f_1(\boldsymbol{X}) = exp\{-0.5D^{-1}[p_{11}a_1^2 + p_{22}a_2^2 + p_{33}a_3^2 + 2p_{12}a_1a_2 + 2p_{13}a_1a_3 + 2p_{23}a_2a_3]\}$.

Here, $a_1 = X_1 - \mu_1, a_2 = X_2 - \mu_2$, $a_3 = X_3 - \mu_3$, $p_{11} = \sigma_{22}\sigma_{33} - {\sigma_{23}}^2$, $p_{12} = \sigma_{13}\sigma_{23} - \sigma_{12}\sigma_{33}$, $p_{13} = \sigma_{12}\sigma_{23} - \sigma_{13}\sigma_{22}$, $p_{22} = \sigma_{11}\sigma_{33} - {\sigma_{13}}^2$, $p_{23} = \sigma_{12}\sigma_{13} - \sigma_{11}\sigma_{23}$, $p_{33} = \sigma_{11}\sigma_{22} - {\sigma_{12}}^2$ *and* $D = \sigma_{11}(\sigma_{22}\sigma_{33} - {\sigma_{23}}^2) - \sigma_{12}(\sigma_{12}\sigma_{33} - \sigma_{13}\sigma_{23}) + \sigma_{13}(\sigma_{12}\sigma_{23} - \sigma_{13}\sigma_{22})$.

The following quantities need to be calculated to simulate sample from TVN via RSSMVNR*:

(i) $a(r) = sup_{\chi}[f_1(\boldsymbol{X})]^{\frac{1}{3r+1}}$,

(ii) $b_1^-(r) = inf_{\chi_1^-} X_1[f_1(\boldsymbol{X})]^{\frac{r}{3r+1}}$,

(iii) $b_1^+(r) = sup_{\chi_1^+} X_1[f_1(\boldsymbol{X})]^{\frac{r}{3r+1}}$,

(iv) $b_2^-(r) = inf_{\chi_2^-} X_2[f_1(\boldsymbol{X})]^{\frac{r}{3r+1}}$,

(v) $b_2^+(r) = sup_{\chi_2^+} X_2[f_1(\boldsymbol{X})]^{\frac{r}{3r+1}}$,

(vi) $b_3^-(r) = inf_{\chi_3^-} X_3[f_1(\boldsymbol{X})]^{\frac{r}{3r+1}}$ and

(vii) $b_3^+(r) = sup_{\chi_3^+} X_3[f_1(\boldsymbol{X})]^{\frac{r}{3r+1}}$,

where $\chi_i^- = \{\boldsymbol{X} \in \chi : X_i \leq 0\}$, $\chi_i^+ = \{\boldsymbol{X} \in \chi : X_i \geq 0\}$, $i = 1, 2, 3$. Analytically determining the values of $a(r)$, $b_1^-(r)$, $b_1^+(r)$, $b_2^-(r)$, $b_2^+(r)$, $b_3^-(r)$, $b_3^+(r)$ is difficult. Therefore, we have used 'genoud' function in R under 'rgenoud' package to calculate these values. The detail procedure of RSSMVNR* to simulate sample from TVN is summarised in algorithm 9.

Algorithm 9: RSSMVNR* for $d = 3$

Input: Constraints $a(r), b_1^-(r), b_1^+(r), b_2^-(r), b_2^+(r), b_3^-(r), b_3^+(r)$.

Output: Sample from the TVN.

Begin

For $i = 1, 2, \cdots\cdots, n$ **do**

1. Generate $U_1, U_2, U_3, U_4 \sim Uniform(0,1)$
2. Calculate $U = a(r) \times U_1$, $V_1 = b_1^-(r) + \left(b_1^+(r) - b_1^-(r)\right) \times U_2$, $V_2 = b_2^-(r) + \left(b_2^+(r) - b_2^-(r)\right) \times U_3$ and $V_3 = b_3^-(r) + \left(b_3^+(r) - b_3^-(r)\right) \times U_4$
3. **If** $U \leq \left[f_1\left(\frac{V_1}{U^r}, \frac{V_2}{U^r}, \frac{V_3}{U^r}\right)\right]^{\frac{1}{3r+1}}$ **then**
   - $\boldsymbol{X} = \left(\frac{V_1}{U^r}, \frac{V_2}{U^r}, \frac{V_3}{U^r}\right)$

   Else
   - Go back to step 1

   End If

End For loop

End Begin

Finally, we calculate the probability of acceptance(P_a) by plugging in the values of $a(r), b_1^-(r), b_1^+(r), b_2^-(r), b_2^+(r), b_3^-(r), b_3^+(r)$ in

$$P_a = \frac{C}{(3r+1)a(r)\prod_{i=1}^{3}\left(b_i^-(r) - b_i^+(r)\right)}.$$

The value of r that maximizes the expression of P_a is the optimal value of r. Mathematical analysis reveals that $r = 0.5$ is optimal for multivariate normal case, regardless of dimension and covariance structure (Wakefield, Gelfand & Smith, 1991).

4. Simulation Setting

To see the performance of proposed RSSMVN, we conducted an extensive simulation study, and result of our simulation study are presented in this section. More specifically, simulation is conducted to check the normality and randomness properties of generated sample along with performance comparison between existing method and RSSMVN. Furthermore, acceptance rate of RSSMVNR and RSSMVNR* are also calculated to determine their efficiency. Here all the numerical computations are computed in *R* on a MacBook Air with an Intel (R) Core (TM) i7 processor running at 1.80 GHz.

The proposed RSSMVN should be able to simulate for all valid $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values. We choose mean vector, $\boldsymbol{\mu}$, arbitrarily (without help of any statistical packages) as choosing $\boldsymbol{\mu}$ is straight forward. However, choosing $\boldsymbol{\Sigma}$ arbitrarily is not straight forward like choosing mean vector as it needs to be a positive definite and invertible matrix. Therefore, we used a R function '`genPositiveDefMat`' in R under '`clusterGeneration`' package to generate a positive definite matrix (Joe, 2006).

In our simulation study, we have considered different combinations of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values, but we presented here only the following combinations.

- BVN case $(d = 2)$
  1. $\boldsymbol{\mu}_1 = (0 \quad 0)^T$ and $\boldsymbol{\mu}_2 = (6 \quad 5)^T$.
  2. $\boldsymbol{\Sigma}_1 = \begin{pmatrix} 8.31 & 0.38 \\ 0.38 & 7.70 \end{pmatrix}$ and $\boldsymbol{\Sigma}_2 = \begin{pmatrix} 4.74 & 0.30 \\ 0.30 & 3.73 \end{pmatrix}$.
- TVN case $(d = 3)$
  1. $\boldsymbol{\mu}_1 = (0 \quad 0 \quad 0)^T$ and $\boldsymbol{\mu}_2 = (6 \quad 5 \quad 8)^T$.
  2. $\boldsymbol{\Sigma}_1 = \begin{pmatrix} 5.65 & -3.03 & -1.19 \\ -3.03 & 7.21 & -1.93 \\ -1.19 & -1.93 & 4.98 \end{pmatrix}$ and

     $\boldsymbol{\Sigma}_2 = \begin{pmatrix} 9.26 & -0.57 & -0.03 \\ -0.57 & 6.361 & -0.61 \\ -0.03 & -0.61 & 8.82 \end{pmatrix}$.

Here, we have considered 4 combinations of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for each dimension.

5. Results and Discussion

In this section, the results of simulation study of our proposed method, RSSMVN, are presented along with discussions. All the results presented here are produced using a random seed number and we have found that using different seed numbers produce similar kind of results. In RSSMVNR and RSSMVNR*, all optimization is done by using '`genoud`' function in *R* under '`rgenoud`' package as optimization of d-dimensional ($d \geq 2$) density is difficult. Furthermore, in RSSMVNR* method, we are to determine the optimal value of r as acceptance rate entirely depends on the value of r. Mathematical analysis reveals that $r = 0.5$ is optimal for multivariate normal case, regardless of dimension and covariance structure (Wakefield, Gelfand, & Smith, 1991). Therefore, in this paper, while generating multivariate normal variates we have used $r = 0.5$ in RSSMVNR* method.

5.1. Bivariate Normal Generation

To test the normality of the generated observations Mardia's test for multivariate normality is used. Table 1 shows the results of Mardia's test for generated samples for different combinations of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. From Table 1, it is observed that the P - Values of Mardia's both test statistics are greater than $\alpha = 0.05$ for all the three methods under RSSMVN, which support the null hypothesis of normality of generated BVN samples.

To test the randomness of generated samples obtained under all methods, we used the graphical technique (ACF plot) and statistical test (Ljung- Box test). Usually, sample observations generated through any method whose building block is Gibbs sampling are correlated as it uses full conditional distributions iteratively. This is also obvious in RSSMVNG as well. The simulated observations obtained under RSSMVNG method are correlated with each other which is shown in the first two plots of Fig 1. From Figure 1, it is observed that some of the peaks at lag around 1, 30, 39, 60, 70, 79 and 98 for X_1 and at lag around 1, 3, 9, 12, 70 and 100 for X_2 are beyond the significance confidence bands (95%).

A possible remedy of successive correlation in the sample is that the sample are thinned so that resulting sample is close to independent. For our simulation study, we have retained every 100^{th} observation from each sample of 50000 observations and thinned samples have been named as modified samples. ACF plots for thinned sample are shown in the second row of Figure 1. Ljung-Box test is used to test the randomness of X_1 but result of this test is not presented here. From the Ljung-Box test, we have found that for thinned sample lag 1 is insignificant. Similarly, randomness of X_2 is checked through Ljung-Box test, and we have found the same conclusion like X_1. Finally, sample generated via RSSMVNG needs to be carefully used (thinned) before drawing any inference regarding unknown parameter.

Table 1: Mardia's test results (n = 500, d = 2)

Method	$\boldsymbol{\Sigma}$	$\boldsymbol{\mu}$	Measure	P - value
RSSMVNG	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_1$	Skewness	0.412
			Kurtosis	0.366
		$\boldsymbol{\mu}_2$	Skewness	0.549
			Kurtosis	0.723
	$\boldsymbol{\Sigma}_2$	$\boldsymbol{\mu}_1$	Skewness	0.552
			Kurtosis	0.096
		$\boldsymbol{\mu}_2$	Skewness	0.621
			Kurtosis	0.482
RSSMVNR	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_1$	Skewness	0.785
			Kurtosis	0.644
		$\boldsymbol{\mu}_2$	Skewness	0.641
			Kurtosis	0.122
	$\boldsymbol{\Sigma}_2$	$\boldsymbol{\mu}_1$	Skewness	0.641
			Kurtosis	0.448
		$\boldsymbol{\mu}_2$	Skewness	0.608
			Kurtosis	0.546
RSSMVNR*	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_1$	Skewness	0.641
			Kurtosis	0.725
		$\boldsymbol{\mu}_2$	Skewness	0.343
			Kurtosis	0.390
	$\boldsymbol{\Sigma}_2$	$\boldsymbol{\mu}_1$	Skewness	0.420
			Kurtosis	0.729
		$\boldsymbol{\mu}_2$	Skewness	0.188
			Kurtosis	0.968

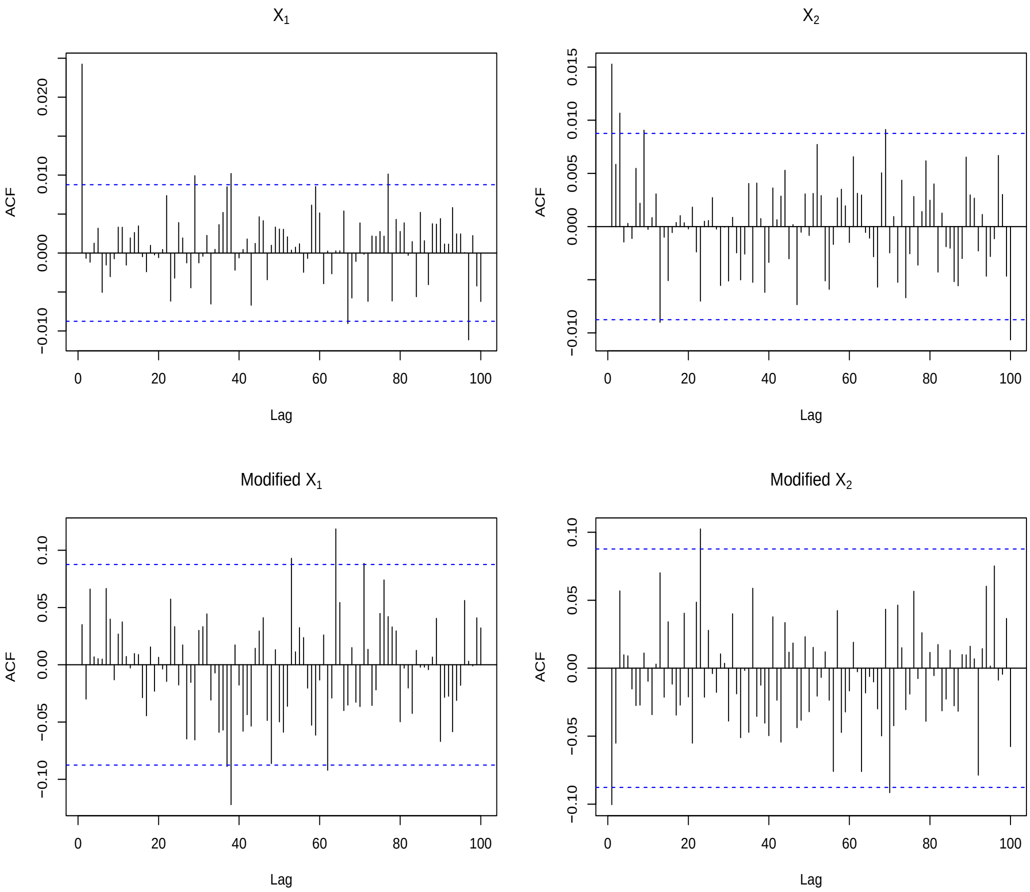

Figure 1: ACF plot of observations (d = 2) generated via RSSMVNG

As uniform random numbers are used (play the role like the proposal density in accept reject algorithm) to generate sample from MVN in both RSSMVNR and RSSMVNR*, the samples are expected to be independent. Figure 2 presents the ACF plots of observations simulated via RSSMVNR and RSSMVNR* methods. It is observed that some of the peaks at lag around 4, 19, 62, 67 and 91 of X_1, and 30, 43, 74 and 82 of X_2 for RSSMVNR method and 23, 79, 86 and 98 of X_1 and 17, 54, 73 and 99 of X_2 for RSSMVNR* method are beyond the significance confidence bands (95%). However, it does not guarantee the presence of autocorrelation, and may happen because of sampling error. Ljung-Box test is carried out to confirm this, and we have found insignificant lag at 1.

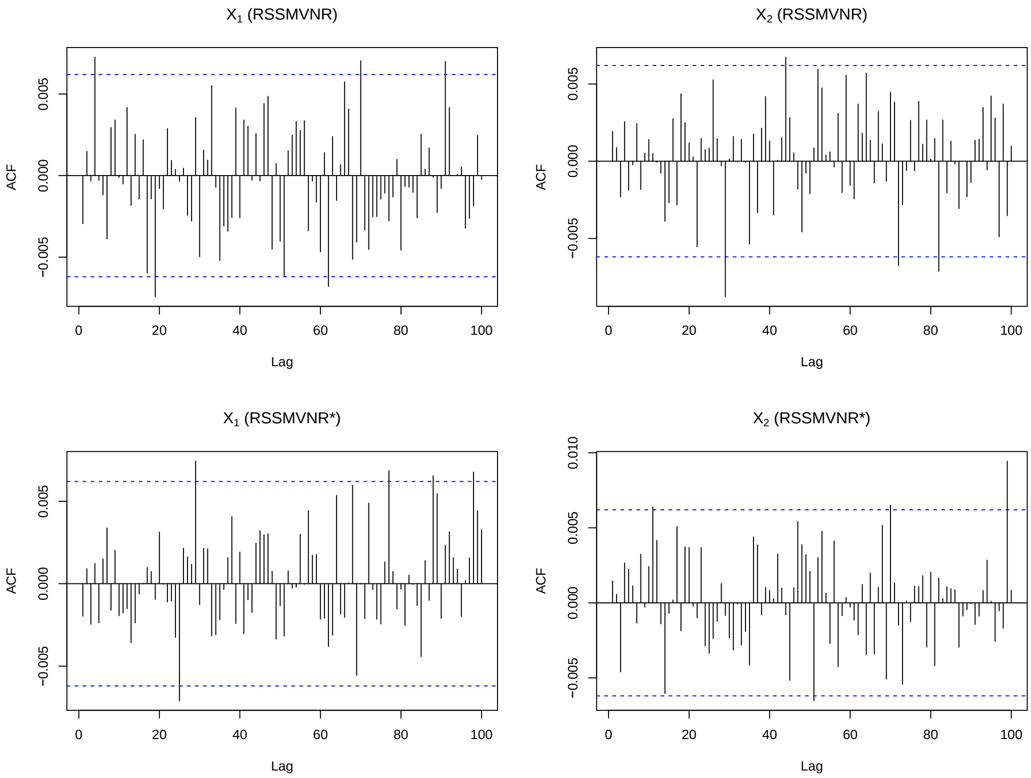

Figure 2: ACF plot of observations (d = 2) generated via RSSMVNR and RSSMVNR*

Table 2 presents the simulated acceptance rate against the theoretical acceptance rate for each combination. From Table 2, it is observed that when mode is shifted from zero vector then acceptance rate decreases from 47% to 28%. From Table 2,

Table 2: Theoretical and simulated acceptance rates of RSSMVNR for different $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values.

Covariance	Mean	P_{accep}	$\hat{P}_{accep}$
$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_1$	0.473	0.476
	$\boldsymbol{\mu}_2$	0.281	0.268
$\boldsymbol{\Sigma}_2$	$\boldsymbol{\mu}_1$	0.473	0.477
	$\boldsymbol{\mu}_2$	0.200	0.193

it is also observed that simulated acceptance rate, calculated based on a sample of size 5000, is very close to theoretical acceptance rate. Therefore, before generating $\boldsymbol{X} \sim N_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the density is relocated to $\boldsymbol{X} \sim N_d(\boldsymbol{0}, \boldsymbol{\Sigma})$ first to achieve maximum acceptance rate. The acceptance rate for RSSMVNR* is shown against its counterpart RSSMVNR in Table 3. We have presented only first combination where zero mean vector is considered. For other combinations where mode is away

Table 3: Theoretical and simulated acceptance rates for RSSMVNR and RSSMVNR*

Method	$\boldsymbol{P_{accep}}$	$\boldsymbol{\widehat{P}_{accep}}$
RSSMVNR	0.473	0.476
RSSMVNR*	0.533	0.535

from zero acceptance rate decreases in both cases. However, RSSMVNR* has a higher acceptance rate than RSSMVNR irrespective of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values. Here, we can also rescale BVN density to the density where mode is zero to get maximum acceptance rate.

Table 4: Average bias of μ and Σ calculated based on 50 data sets (d = 2)

Parameter	**Method**	**Sample Size ($\boldsymbol{n}$)**				
		100	**250**	**500**	**1000**	**10000**
μ	Existing	0.378	0.222	0.178	0.107	0.035
	RSSMVNG	0.121	0.096	0.079	0.036	0.011
	RSSMVNR*	0.321	0.244	0.162	0.124	0.033
Σ	Existing	4.324	1.735	0.953	0.309	0.036
	RSSMVNG	5.035	1.337	0.148	0.024	0.084
	RSSMVNR*	4.907	1.565	0.882	0.345	0.039

* $\|A-B\|_F$ denotes the distance between two matrices A and B where $\|X\|_F = tr(X^T X)$ is the Frobenius norm.

Table 4 presents the average bias of mean and covariance which are calculated based on 50 BVN samples simulated from $N_2(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_1})$ through existing (decomposition based) and proposed methods respectively. These results are reproducible as specific seed number is used to simulate each of the 50 date sets under both methods. We have used different size of samples to compare the performance of these methods. In RSSMVNG method, bias of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are calculated based on the thinned samples. The Frobenius norm is used to calculate the bias of covariance matrix while simple Euclidean distance is used to calculate the bias of mean vector (Habeck, 2009). From Table 4, it is evident that the average bias of both mean and covariance across all the methods are generally close to each other. Furthermore, as sample size increases the values of average bias of mean and covariance decrease for all the methods.

Table 5: Average computing time in seconds calculated based on 50 data sets (*d* =2)

Method	**Sample size**				
	100	**250**	**500**	**1000**	**10000**
Existing	0.0001	0.0004	0.0005	0.006	0.0020
RSSMVNG	0.1732	0.4573	0.9141	1.8146	19.3400
RSSMVNR*	1.4739	1.2320	0.1240	1.2538	1.7694

The average computing time required to simulate sample from $N_2(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_1})$ under existing and proposed methods for different n values are presented in Table 5. From Table 5, it is clear that our proposed method requires higher computing time compared to the existing method irrespective of all sample sizes, and this is because of thinning and numerical solutions required in RSSMVNG and RSSMVNR* respectively.

5.2. Trivariate Normal Generation

To test the normality of the generated observations Mardia's test for multivariate normality is used. Table 6 shows the results of Mardia's test for generated samples for different combinations of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. From Table 6, it is observed that the P - Values of Mardia's both test statistics are greater than $\alpha = 0.05$ for all the three methods under RSSMVN, which support the null hypothesis of normality of generated TVN samples.

To test the randomness of generated samples obtained under all methods, we used the graphical technique (ACF plot) and statistical test (Ljung- Box test). Usually, sample observations generated through any method whose building block is Gibbs sampling are correlated as it uses full conditional distributions iteratively. This is also obvious in RSSMVNG as well. The simulated observations obtained under RSSMVNG method are correlated with each other which is shown in the first three plots of Figure 3. A possible remedy of successive correlation in the sample is that the sample are thinned so that resulting sample is close to independent. For our simulation study, we have retained only every 100^{th} observation from each sample of 50000 observations and thinned samples have been named as modified samples. ACF plots for thinned sample are shown in the second row of Fig 3. Ljung-Box test is used to test the randomness of X_1 but result of this test is not presented here. From the Ljung-Box test, we have found that for thinned sample lag 1 is insignificant. Similarly, randomness of X_2 and X_3 are checked through Ljung-Box test, and we have found the same conclusion like X_1. Finally, sample generated via RSSMVNG needs to be carefully used (thinned) before drawing any inference regarding unknown parameter. As uniform random numbers are used (play the role like the proposal density in accept reject algorithm) to generate sample from MVN in both RSSMVNR and RSSMVNR*, the samples are expected to be independent.

Table 6: Mardia's test results (n = 500, d = 3)

Method	$\boldsymbol{\Sigma}$	$\boldsymbol{\mu}$	**Measure**	$\boldsymbol{P}$**-value**
RSSMVNG	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_1$	Skewness	0.245
			Kurtosis	0.161
		$\boldsymbol{\mu}_2$	Skewness	0.897
			Kurtosis	0.645
	$\boldsymbol{\Sigma}_2$	$\boldsymbol{\mu}_1$	Skewness	0.352
			Kurtosis	0.866
		$\boldsymbol{\mu}_2$	Skewness	0.185
			Kurtosis	0.744
RSSMVNR	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_1$	Skewness	0.436
			Kurtosis	0.521
		$\boldsymbol{\mu}_2$	Skewness	0.192
			Kurtosis	0.378
	$\boldsymbol{\Sigma}_2$	$\boldsymbol{\mu}_1$	Skewness	0.780
			Kurtosis	0.637
		$\boldsymbol{\mu}_2$	Skewness	0.122
			Kurtosis	0.274
RSSMVNR*	$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_1$	Skewness	0.233
			Kurtosis	0.296
		$\boldsymbol{\mu}_2$	Skewness	0.183
			Kurtosis	0.818
	$\boldsymbol{\Sigma}_2$	$\boldsymbol{\mu}_1$	Skewness	0.999
			Kurtosis	0.700
		$\boldsymbol{\mu}_2$	Skewness	0.163
			Kurtosis	0.793

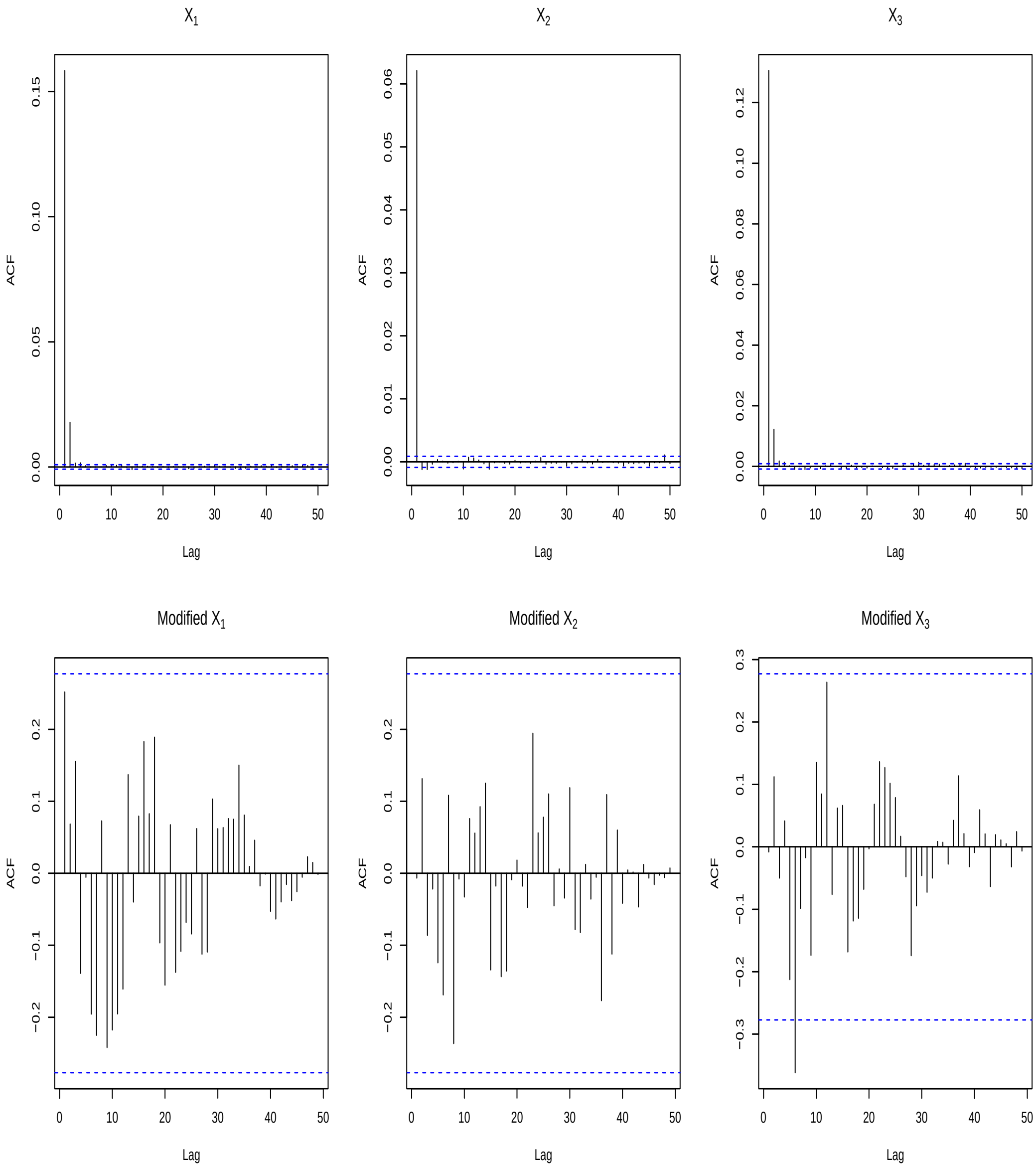

Figure 3: ACF plot of observations (d = 3) generated via RSSMVNG

Table 7 presents the simulated acceptance rate against the theoretical acceptance rate for each combination. From Table 7, it is observed that when mode is shifted from zero vector then acceptance rate decreases from 22% to 8%. From Table 7, it is also observed that simulated acceptance rate, calculated based on a sample of size 5000, is very close to theoretical acceptance rate. Therefore, before generating $\boldsymbol{X} \sim N_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ the density is relocated to $\boldsymbol{X} \sim N_d(\boldsymbol{0}, \boldsymbol{\Sigma})$ first to achieve maximum acceptance rate.

The acceptance rate for RSSMVNR* is shown against its counterpart RSSMVNR in Table 8. In this Table, we have shown only two combinations where mean vector is zero vector. For other combinations in which mode is away from zero acceptance rate decreases in both cases. However, RSSMVNR* has a higher acceptance rate than RSSMVNR irrespective of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values. Here, we can also rescale TVN density to the density where mode is zero to get maximum acceptance rate.

Table 7: Theoretical and simulated acceptance rates of RSSMVNR for different $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ values.

Covariance	Mean	$\boldsymbol{P_{accep}}$	$\widehat{\boldsymbol{P}}_{\boldsymbol{accep}}$
$\boldsymbol{\Sigma}_1$	$\boldsymbol{\mu}_1$	0.229	0.227
	$\boldsymbol{\mu}_2$	0.081	0.080
$\boldsymbol{\Sigma}_2$	$\boldsymbol{\mu}_1$	0.274	0.272
	$\boldsymbol{\mu}_2$	0.127	0.125

Table 8: Theoretical and simulated acceptance rates for RSSMVNR and RSSMVNR*

Covariance	Method	RSSMVNR	RSSMVNR*
$\boldsymbol{\Sigma}_1$	$\boldsymbol{P_{accep}}$	0.229	0.262
	$\widehat{\boldsymbol{P}}_{\boldsymbol{accep}}$	0.227	0.252
$\boldsymbol{\Sigma}_2$	$\boldsymbol{P_{accep}}$	0.274	0.314
	$\widehat{\boldsymbol{P}}_{\boldsymbol{accep}}$	0.272	0.319

Table 9 presents the average bias of mean and covariance calculated based on 50 samples simulated from $N_3(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_1})$ through existing and proposed methods. Like the case $d = 2$ shown in earlier, different size of samples are considered to compare the performance of these methods along with considered a specific seed number to generate each of the 50 data sets so that all the results are reproducible. From Table 9, it is evident that the average bias of both mean and covariance parameters under all methods are close to each other. Again, as sample size increases the values of average bias of mean and covariance decrease for all the methods.

Table 9: Average bias of μ and Σ values calculated based on 50 data sets (d = 3)

Parameter	**Method**	**Sample Size ($\boldsymbol{n}$)**				
		100	**250**	**500**	**1000**	**10000**
μ	Existing	0.404	0.229	0.174	0.111	0.038
	RSSMVNG	0.400	0.242	0.188	0.131	0.037
	RSSMVNR*	0.379	0.233	0.179	0.122	0.037
Σ	Existing	4.291	2.058	0.953	0.450	0.042
	RSSMVNG	4.431	1.914	0.979	0.460	0.049
	RSSMVNR*	5.145	1.870	0.983	0.434	0.041

The average computing time required to simulate sample from $N_3(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_1})$ under existing and proposed methods for different n values are presented in Table 10. From Table 10, it is clear that proposed method requires higher computing time compared to the existing method irrespective of all sample sizes, and this is because of thinning and numerical solutions required in RSSMVNG and RSSMVNR* respectively.

Table 10: Average computing time in seconds calculated based on 50 data sets (d = 3)

Method	**Sample size**				
	100	**250**	**500**	**1000**	**10000**
Existing	0.0002	0.0005	0.0006	0.0009	0.0047
RSSMVNG	0.1640	0.4798	0.9259	1.8776	20.334
RSSMVNR*	3.3589	3.4595	3.5066	3.6092	4.8426

From the above results and discussion, it is clear that sample obtained through proposed method satisfies both the normality and the randomness properties. Furthermore, the performance of the proposed method is very similar to that of performance of the existing method in terms of bias of the parameters although the proposed method requires higher computing time compared to the existing method irrespective of all the sample sizes. However, one can use the proposed method as an alternative method to simulate sample from MVN as required computing time in proposed method is not too big to think about in a modern day. Although any statistical efficiency (consistently getting less bias) of the proposed method is not found over existing method in our simulation studies, the proposed method could be really handy in computational aspects (where existing method may get stuck, Ripley 1987). The proposed decomposition free method discussed in this paper can be used as an alternative method where existing method fails (due to numerical instability because of decomposition of a specific covariance matrix). Exploring such numerical issues of existing method could be really an interesting computational work for future researchers.

6. Application

Suppose we want to calculate $\Pr(X_1^2 + 2X_2^3 > 3)$ where $\boldsymbol{X} = (X_1, X_2)$ follows bivariate standard normal distribution with $cov(X_1, X_2) = 0.25$. Mathematically, $J = \iint I_{[X_1^2+2X_2^3>3]} f(x_1 x_2) dx_1 dx_2$, where $-\infty < X_1, X_2 < \infty$ and $I_{[X_1^2+2X_2^3>3]}$ is an indicator function. Analytical solution of J is not straightforward. However, estimating the integral using Monte- Carlo approach is very easy.

Monte- Carlo Approach:

- Generate $\boldsymbol{X} = (X_1, X_2) \sim BVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
- $\hat{J} = n^{-1} \sum_{i=1}^{n} (x_{1i}^2 + x_{2i}^3 > 3)$.

Considering $n = 1000$, using any of the methods of our proposed RSSMVN, we have $\hat{J} = 0.207$.

7. Conclusion and Future Works

In this paper, we have proposed a new technique RSSMVN for generating sample from multivariate normal density. The building blocks of the proposed method are Gibbs, Ratio-of-Uniforms and Modified Ratio-of-Uniforms method. The validity and efficiency of the RSSMVN is investigated through an extensive simulation study. From the simulation study, it is observed that RSSMVN can be used as an alternative method to generate sample from MVN. From our simulation study it is observed that RSSMVN has a high efficiency (53% acceptance rate for d = 2 while it is 31% for d = 3) when its building block is MRoU method. However, this paper does not cover the numerical issues arose due to decomposition of specific $\mathbf{\Sigma}$ in decomposition method. Finding those $\mathbf{\Sigma}$, for which decomposition based approach gets stuck, and investigating the robustness of RSSMVN for them could be a very interesting computational research work in future.

Reference

[1] Casella, G. and Berger, R. L. (2002). Statistical inference. Pacific Grove, CA: Duxbury, (2), 337-472.

[2] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on pattern analysis and machine intelligence, (6), 721-741.

[3] Habeck, M. (2009). Generation of three-dimensional random rotations in fitting and matching problems. Computational Statistics, 24(4), 720-721.

[4] Joe, H. (2006). Generating random correlation matrices based on partial correlations. Journal of Multivariate Analysis, 97(10), 2177-2189.

[5] Johnson, R. A. and Wichern, D. W. (2002). Applied multivariate statistical analysis. Upper Saddle River, NJ: Prentice hall, 5(8).

[6] Kinderman, A. J. and Monahan, J. F. (1977). Computer generation of random variables using the ratio of uniform deviates. ACM Transactions on Mathematical Software (TOMS), 3(3), 257-260.

[7] Ljung, G. M. and Box, G. E. (1978). On a measure of lack of fit in time series models. Biometrika, 65(2), 297-303.

[8] Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. Biometrika, 57(3), 519-530.

[9] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6), 1087-1092.

[10] Ripley, B. D. (1987). Stochastic Simulation, John Willey & Sons. New York, 98.

[11] Wakefield, J. C., Gelfand, A. E. and Smith, A. F. M. (1991). Efficient generation of random variates via the ratio-of-uniforms method. Statistics and Computing, 1(2), 129-133.