COMPUTER GENERATION OF DIRECTIONAL DATA

by

Carl Ka-Fai WONG

A

Thesis

Submitted to

(Division of Statistics)

The Graduate School

of

The Chinese University of Hong Kong

In Partial Fulfillment

of the Requirements of the Degree of

Master of Philosophy

(M. Phil.)

May, 1991

THE CHINESE UNIVERSITY OF HONG KONG

GRADUATE SCHOOL

The undersigned certify that we have read a thesis, entitled "Computer Generation of Directional Data" and submitted to the Graduate School by Wong Ka Fai Carl (王家輝) in partial fulfillment of the requirements of the degree of Master of Philosophy in Statistics. We recommend that it be accepted.

Dr. K.H. Li,
Supervisor

N.N. Chan

Dr. N.N. Chan

Leung Pui Lam

Dr. P.L. Leung

Prof. W.H. Wong
External Examiner

ACKNOWLEDGEMENT

I would like to express my gratitude here to my supervisor, Dr. Kim-hung Li for his earnest support and guidance that leads to the completion of this research project. In addition, it is my great pleasure to give thanks to my Father who always loves me and encourages me and it is that very reason which makes everything a pleasure.

DECLARATION

No portion of the work referred to in this thesis has been submitted in support of an application for another degree of qualification of this or any other university or other institute of learning.

ABSTRACT

Various methods are discussed for the generation of directional data on the circle, sphere and higher dimensional sphere. The distribution for directional data may be uniform or non-uniform. Along with the existing methods, efforts have been made to develop some competitive or more effective methods. Theoretical and empirical comparisons of the efficiencies of various generating methods are conducted. Advantages and disadvantages of these methods are then discussed. An envelope which is proved to be more efficient and compact than other competing envelopes is proposed for the Dimroth-Watson distribution on the sphere and its usefulness in the generation of variates from other distributions like Bingham and Bingham-Mardia distributions are also explored.

Keywords: Directional data, n-dimensional spherical distributions, envelope-rejection method, sampling efficiency, inversion method, composition method, marginal generation time.

CONTENTS

Chapter One

INTRODUCTION

§1.1 Directional Data and Computer Simulation

Directional data arise in many areas of observation and scientific experimentation. Examples include the homing preference of migrating birds, the optical orientation of ants, arrival directions of showers of cosmic rays and palaeomagnetic directions in rocks. Moreover, observations that are by no means having orientations in nature can sometimes be usefully expressed in the form of directions and analyzed as directional data. For example in social science, it has been the practice to analyze data on occupational judgements by individuals as unit vectors. Extensive examples can be found in Batschelet (1981) and Fisher et al (1987).

In principle, directions in the n-dimensional space can be represented by the vector $X = (X_1,\ldots,X_n)^T$, where $\sum_{i=1}^{n} X_i^2 = 1$ and A^T stands for the transpose of matrix/vector A. The set of all possible X defines an n-dimensional unit hypersphere (n-sphere). A point X on the n-sphere can be uniquely represented by n - 1 angles $\Theta_1, \Theta_2, \ldots, \Theta_{n-1}$ and the following polar transformation:

$$
\begin{aligned}
X_1 &= \sin\Theta_1 \sin\Theta_2 \ldots\ldots \sin\Theta_{n-3} \sin\Theta_{n-2} \sin\Theta_{n-1} \\
X_2 &= \sin\Theta_1 \sin\Theta_2 \ldots\ldots \sin\Theta_{n-3} \sin\Theta_{n-2} \cos\Theta_{n-1} \\
X_3 &= \sin\Theta_1 \sin\Theta_2 \ldots\ldots \sin\Theta_{n-3} \cos\Theta_{n-2} \\
&\vdots \\
X_{n-2} &= \sin\Theta_1 \sin\Theta_2 \cos\Theta_3 \\
X_{n-1} &= \sin\Theta_1 \cos\Theta_2 \\
X_n &= \cos\Theta_1,
\end{aligned}
$$

where $\Theta_j \in [0,\pi]$, $j = 1, 2, \ldots, n-2$; $\Theta_{n-1} \in [0,2\pi)$.

The probability density functions (p.d.f.s) or densities of directional data models have their entire support on the surface of the n-sphere. Much work has been done in the estimation and hypothesis testing of parameters in n-dimensional spherical distributions; see e.g. Watson and Williams (1956), Mardia (1972, 1975) and Watson (1983). Investigation of power of certain statistics, robustness and distributional properties is often analytically impossible and simulation therefore becomes an important alternative when computer algorithms are available for generating random directions from these distributions. Some obvious applications of computer simulation are briefly mentioned here: to examine the performance of new statistical procedures; to access the small-sample properties of procedures based on asymptotic results; and to investigate the properties of standard procedures when the underlying assumptions are violated. These applications are as relevant to directional statistics as to other areas of the subject.

§1.2 Computer Simulation Techniques

Computer simulation techniques useful for generating directional data are no different from the techniques for generating random variates from univariate distributions. The following techniques are oftenly employed in this paper. We briefly discuss them here.

(1) Inversion method

Denote the uniform distribution on an interval (a,b) by U(a,b). The inversion method is stated simply as:

1. Generate $U \sim U(0,1)$.
2. Set $X = \inf\{ x : F(x) \geq U \}$.

The resulting variate X from step 2 has distribution function (d.f.) F. In our applications, F is strictly increasing, so the inverse function F^{-1} is well-defined and step 2 becomes $X = F^{-1}(U)$. This method is very efficient as long as F^{-1} is easily computable.

(2) Envelope-rejection method

Let $f(x)$ be the p.d.f. of a random variable which is to be sampled. Represent $f(x)$ as

$$f(x) = Cg(x)\varphi(x), \qquad C \geq 1,$$

where $g(x)$ is also a p.d.f. and $0 < \varphi(x) \leq 1$. The envelope-rejection method can be summarized as follows:

1. Generate $U \sim U(0,1)$ and $Y \sim g(y)$ independently.
2. If $U > \varphi(Y)$, then go to step 1; otherwise accept $X = Y$.

The accepted variate X in step 2 has p.d.f. $f(x)$. We call $Cg(x)$ the envelope for $f(x)$. The sampling efficiency which is in fact the probability that a Y in step 1 can be accepted in step 2 is defined as

$$\Pr(U \leq \varphi(Y)) = 1/C.$$

There are three prerequisites for the method to be useful. They are: (a) it is easy to generate Y from g, (b) the sampling efficiency is not close to zero, and (c) it is easy to compute $\varphi(x)$. The importance of (c) can be lessened if a good "squeezing function" (function that is close to and less than φ) is available.

The same idea is applicable to the multivariate case.

(3) Composition method

In certain cases the p.d.f. of interest can be expressed as a probability mixture of other p.d.f.s. Let $f(x)$ be the p.d.f. of interest. Suppose

$$f(x) = \sum_{j=1}^{\infty} p_j f_j(x),$$

where $\{f_j\}$ are p.d.f.s and $\{p_j\}$ are non-negative constants with $\sum_{j=1}^{\infty} p_j = 1$. The first stage of the composition method consists of generating a discrete random variable Y with $Pr(Y=j) = p_j$. Suppose Y take the particular value J. Then the second stage consists of generating X from f_J. The variate X so generated has p.d.f. f(x).

§1.3 Implementation and Preliminaries

In later chapters we will discuss various generating methods for random vectors from distributions on the n-sphere. Unless the generating methods are very simple and straightforward (e.g. by inversion method), we access the performance of each method by considering either its sampling efficiency (if envelope-rejection method is used) or its mean requirement of random numbers used to generate one random vector, or both. The demand for random numbers is an important measure as their generation is usually more time-consuming than the usual arithmetic operations such as additions and multiplications.

In addition to the theoretical considerations, empirical comparisons are necessary. One such comparison is to consider the marginal generation time which is the average CPU time to generate one random vector. The marginal generation time should not include any computer processing time for the set-up calculations for constants required before the first vector can be generated as these constants are unaltered once they are computed. In some applications, these constants need to be re-set (re-calculated) and the mean generation time for each vector is therefore longer. Throughout this paper, we will mainly consider marginal generation time unless otherwise stated.

All algorithms are programmed as Fortran subroutines and executed on the IBM4381 computer of the Chinese University of Hong Kong for comparisons. Random numbers are obtained by RNUN (subroutine) or RNUNF (function) from the IMSL Stat/Library (1987). Both are based on a generator of the multiplicative congruential type. RNUN is used in chapter two only and the function version RNUNF is used throughout the remaining chapters. The Fortran codings of selected algorithms are listed in Appendix 1.

CHAPTER TWO

GENERATING RANDOM POINTS ON THE N-SPHERE

§2.1 Methods

Random points on the n-sphere means points which are independently and uniformly distributed on the n-sphere. They correspond to equally likely directions.

The generation of random points on the n-sphere is important in the simulation of n-dimensional spherical processes on computer (details will be given in Chapter 5). Applications of spherical processes have been considered by Muller (1956) and Motoo (1959). Stephens (1964), Watson and Williams (1956) considered the statistical problems associated with random points on 3-sphere.

The methods for generating random points on the n-sphere may depend on the dimension n. Clearly with $n = 2$, the random point, $X = (\cos\Theta, \sin\Theta)$ where Θ is uniform on $(0,2\pi)$. For $n = 3$, $X = (\sin\Theta\cos\Phi, \sin\Theta\sin\Phi, \cos\Theta)$ where Φ is uniform on $(0,2\pi)$ and $\cos\Theta$ is uniform on $(-1, 1)$, with $\sin\Theta = (1 - \cos^2\Theta)^{1/2}$ can be used. In principle, this polar method can be extended to higher dimensional cases but it has not been used because far more efficient methods are available.

One approach to generate random points on the n-sphere is to use envelope-rejection method. The idea is to simulate a point uniformly distributed in the n-dimensional hypercube $\{(X_1,\ldots X_n): -1\le X_i \le 1 \text{ for all } i\}$ and accept the direction of $(X_1,\ldots,X_n)$ if $(X_1,\ldots,X_n)$ falls inside the n-sphere. The main drawback of this method is that it becomes extremely inefficient when n is large. Therefore we shift our focus to some direct methods.

Muller (1959) suggested the use of standard normal variates normalized by the root sum of squares as the Cartesian coordinates of a random point on the unit hypersphere. Suppose $Z_1,\ldots,Z_n$ are independent standard normal variates, the vector determined by

$$X_i = \frac{Z_i}{(Z_1^2+\ldots+Z_n^2)^{1/2}}, \qquad i = 1, 2,\ldots, n \qquad (2.1)$$

will be distributed uniformly on the n-sphere. The advantages of (2.1) are its simplicity, applicability virtually to all dimensions, and the availability of efficient normal generator.

One efficient normal generator is by Box and Muller (1958): given two independent U_1, $U_2 \sim U(0,1)$, a pair of independent standard normal variates can be obtained via

$$\begin{cases} X_1 = (-2\ln U_1)^{1/2}\cos 2\pi U_2 \\ X_2 = (-2\ln U_1)^{1/2}\sin 2\pi U_2 \end{cases}$$

and $-2\ln U_1$ is distributed as χ_2^2. Based on (2.1) and Box-Muller's method for normal random variables, Sibuya (1962) took a step further by considering the generation of an ordered sample of variates from U(0,1) rather than directly generating the standard normal variates. Suppose $n = 2m$. Let $U_{(i)}$ be the ith smallest uniform random number in a sample of size $m - 1$. Thus

$$0 \equiv U_{(0)} \le U_{(1)} \le \ldots \le U_{(m-1)} \le U_{(m)} \equiv 1.$$

Set

$$Y_i = U_{(i)} - U_{(i-1)}, \qquad i = 1, 2,\ldots, m$$

and let $R_1,\ldots,R_m$ be another stream of U(0,1) variates. The vector $\mathbf{X} = (X_1,\ldots,X_{2m})$ with

$$\begin{cases} X_{2i-1} = \sqrt{Y_i}\cos 2\pi R_i, & i = 1, 2,\ldots, m \\ X_{2i} = \sqrt{Y_i}\sin 2\pi R_i, & i = 1, 2,\ldots, m \end{cases} \qquad (2.2)$$

has uniform distribution on the n-sphere.

It is noted that the joint distribution of m - 1 ordered U(0,1) is equivalent to that of the m sub-intervals by random partitioning the unit interval, i.e. the joint p.d.f. of $U_{(1)},\ldots,U_{(m-1)}$ is equivalent to the joint p.d.f. of $Y_1,\ldots,Y_{m-1}$. Since

$$T_1/(\sum_{i=1}^{m} T_i),\ldots\ldots,\ T_m/(\sum_{i=1}^{m} T_i) \tag{2.3}$$

are random partitions of a unit interval, where T_i's are independent χ^2_2 and that $(\sqrt{T_i}\cos 2\pi R_i,\ \sqrt{T_i}\sin 2\pi R_i)$ are independent standard normal variates, (2.2) is actually standard normal variates normalized by root sum of squares, a variation of Muller's method.

When n = 2m+1, we take the first 2m+1 components $X_1,\ldots,X_{2m+1}$ from a random point $(X_1,\ldots,X_{2m+2})$ generated by the preceding procedure and then normalize. In another method for odd dimensions, Sibuya (1962) transformed a random point $(X_1,\ldots X_{2m})$ on the 2m-sphere into a point $(X^*_1,\ldots X^*_{2m+1})$ on the (2m+1)-sphere with

$$X^*_i = SX_i, \qquad i = 1, 2,\ldots, 2m$$
$$X^*_{2m+1} = \pm(1-S^2)^{1/2},$$

where S^2 is a Beta(m,1/2) random variate and $\pm$'s are independent random signs.

The speed of Sibuya's method depends on how fast the sequence of ordered uniforms is produced. One way to enhance the speed is to use an efficient sorting algorithm to sort an existing sequence of uniforms into order. Another way to obtain an ordered sequence of uniforms is to make use of (2.3). It is known as the exponential-spacings method. Gerontidis and Smith (1982) compared other methods also.

From a different viewpoint, Tashiro (1976) used a more direct method to generate random points on the n-sphere. Like Sibuya's method, his algorithm also depends on whether n is even or odd. When n is even, it is the same as Sibuya's method but with the ordered uniforms generated by sequential method. Tashiro made the analysis simpler, even in odd dimensions.

Let $y_1,\ldots,y_n$ be the Cartesian coordinates of a random point in the n-sphere. We shall denote by X_i's and R_i's independent uniform random numbers. Tashiro's algorithm is as follows:

When n = 2m:

1. Set $Y_m = 1$, $Y_0 = 0$. Define recursively

$$Y_i = Y_{i+1} X_i^{1/i}, \qquad i = 1, 2, \ldots, m-1$$

2. Evaluate

$$\begin{aligned} y_{2i-1} &= \sqrt{Y_i - Y_{i-1}}\cos 2\pi R_i, \qquad i = 1, 2, \ldots, m \\ y_{2i} &= \sqrt{Y_i - Y_{i-1}}\sin 2\pi R_i, \qquad i = 1, 2, \ldots, m \end{aligned} \tag{2.4}$$

When n = 2m+1:

1. Set $Z_{m+1} = 1$. Define recursively

$$Z_i = Z_{i+1} X_i^{2/(2i-1)}, \qquad i = 1, 2, \ldots, m$$

2. Evaluate

$$\begin{aligned} y_1 &= \pm\sqrt{Z_1}, \\ y_{2i} &= \sqrt{Z_{i+1} - Z_i}\cos 2\pi R_i, \qquad i = 1, 2, \ldots, m \\ y_{2i+1} &= \sqrt{Z_{i+1} - Z_i}\sin 2\pi R_i, \qquad i = 1, 2, \ldots, m \end{aligned} \tag{2.5}$$

where ±'s are random signs.

Apart from the general methods which are applicable to all dimensions, some tailor-made algorithms have been designed for the 3-dimensional case. For example, Cook (1957) devised a method based on elegant theory and avoided the use of square root. However his method is too slow to be useful.

Making use of the following two facts:

(1) If (Z_1, Z_2, Z_3) is uniform on the 3-sphere, each Z_i is uniform on $(-1,1)$ and given Z_3 the pair (Z_1, Z_2) is uniformly distributed on the circle with radius $(1 - Z_3^2)^{1/2}$;

(2) If (V_1, V_2) is a random point inside the unit circle, then $S = V_1^2 + V_2^2$ is distributed as U(0,1) and is independent of $(V_1/S^{1/2}, \ V_2/S^{1/2})$,

Marsaglia (1972) proposed a faster method. His algorithm is

1. Generate U_1, U_2 independent U(0,1).
2. If $S = U_1^2 + U_2^2 > 1$, go to step 1; otherwise form

$$(2U_1\sqrt{1-S}\,,\ 2U_2\sqrt{1-S}\,,\ 1-2S) \tag{2.6}$$

as a random point on the 3-dimensional sphere.

Marsaglia (1972) also suggested a similar method for the 4-dimensional case. If U_1, U_2 are independent U(-1,1) such that $S_1 = U_1^2 + U_2^2 < 1$ and U_3, U_4 are independent U(-1,1) such that $S_2 = U_3^2 + U_4^2 < 1$, then the point

$$(U_1,\ U_2,\ U_3\sqrt{(1-S_1)/S_2},\ U_4\sqrt{(1-S_1)/S_2}) \tag{2.7}$$

is uniform on the 4-sphere.

§2.2 Comparison of Methods

In previous section, we have discussed various methods in simulating random points on the surface of an n-dimensional unit sphere. Now we compare the performance of these methods. To be specific, the following algorithms are compared (Refer to Appendix 1 for the Fortran codings of each algorithm):

1. **Algorithm RPN1** - Muller's (1959) method on the n-sphere (see (2.1)).

A normal generator RNNOF from the IMSL library is used for the generation of variates from the standard normal distribution N(0,1). The routine, based on Kinderman and Ramage (1976), has been tested to be very efficient among other commonly used generators.

2. **Algorithm RPN2** - Sibuya's (1962) method on the n-sphere.

Random numbers are generated first and then sorted in ascending order by an IMSL routine called SVRGN. This sorting routine uses a combination of quicksort and shellsort described by Singleton (1969) and is proved to be very efficient.

For odd dimensions, say 2m+1, the algorithm extracts the first 2m+1 components of $(X_1,...,X_{2m+2})$ and then normalize by their root sum of squares. (The other method for odd dimensions suggested by Sibuya is not recommended because the generation of the square root of the Beta(m,1/2) variates requires rejection method whose rejection rate is quite high in high dimensions (see Sibuya (1962)), and hence is not efficient.)

3. **Algorithm RPN3** - A modification of Sibuya's method.

It is basically the same as RPN2 except that ordered uniforms are generated by exponential-spacings method.

4. **Algorithm RPN4** - Tashiro's (1976) method on the n-sphere (see (2.4) and (2.5)).

5. **Algorithm RP3** - Marsaglia's (1972) method on the 3-sphere (see (2.6)).

6. **Algorithm RP4** - Marsaglia's (1972) method on the 4-sphere (see (2.7)).

To compare the performance we mean to compare the marginal generation time to generate one random point on the n-sphere. The marginal generation time for each algorithm is obtained by simulating a sample of 1000 random points. A number of dimensions have been selected for comparison. The result is listed in table 2.1.

Table 2.1 Marginal generation time (ms) for a random point on the n-sphere.

	n					
Algorithm	3	4	8	10	21	100
RPN1	.23	.27	.55	.70	1.41	6.73
RPN2	.25	.24	.45	.48	1.04	4.31
RPN3	.22	.21	.40	.49	1.09	4.63
RPN4	.17	.20	.42	.52	1.09	5.19
RP3	.07	-	-	-	-	-
RP4	-	.10	-	-	-	-

Of the four RPN's algorithms, which are applicable to nearly all dimensions ($n \geq 3$), there is none which are uniformly faster than the others. In higher dimensions ($n \geq 10$), RPN2 seems to be the best. The result may well illustrate that a very efficient sorting algorithm such as the one RPN2 used will enhance the speed greatly. It has been tried that if a less efficient sorting algorithm such as shellsort or heapsort (see for example, Press et al (1986)) is used, RPN2 is approximately 20% slower when n is 100. But the reduction in speed becomes negligible when n becomes smaller, say less than 21.

To the other end where the dimensions is small ($n < 8$), all the RPN's algorithms are very similar in speed. When the dimension is 3 or 4, however, the two algorithms RP3 and RP4 for n equals 3 and 4 respectively, win the race. Both are at least twice as fast as their all-round counterparts.

Apart from speed, portability is also important in determining the usefulness of an algorithm. Since all six algorithms can be easily transferred from one machine to another with a minimal amount of change, they are all portable.

Perhaps only in some rare ocassions where speed is of considerable importance such as when we need to generate a large sample in very high dimensions, Sibuya's method in the light of a very efficient sorting routine would be the choice. If such sorting routine is unavailable, then the modified Sibuya's method using exponential-spacings method to generate ordered U(0,1) is reasonably fast enough to meet the need. In general, Muller's simple (it consists of only a few lines of instruction codes) but efficient (depends on the normal generator used) method is good enough for most applications.

CHAPTER THREE

GENERATING VARIATES FROM NON-UNIFORM DISTRIBUTIONS ON THE CIRCLE

§3.1 Introduction

In this and the coming chapters, we will consider the generation of variates from probability distributions other than uniform distribution on the n-sphere. This chapter will be devoted to the case n is two: simulating variates from circular distributions.

Most of the basic distributions on the circle have been derived either from the transformation of univariate random variables (e.g. the wrapped distributions) or as circular analogies of important univariate characterizations. (e.g. the von Mises distribution). Mardia (1972) gives detailed account of the properties of some of these important circular distributions.

Computer generation of variates from circular distributions is essentially the same as that from univariate distributions. Basic variates generation techniques such as the envelope-rejection method is very useful in the circular case as well as in the univariate case. Others include the transformation of univariate random variables generated by some standard methods.

Let $f(\theta)$, $\theta \in \Theta \equiv (-\pi, \pi]$ be the p.d.f. of a circular distribution where θ is the angle subtended at the origin of the unit circle. Our aim is to devise procedures to generate realizations of Θ from f.

One important family of circular distributions is the symmetric unimodal distribution. It serves as a basic probability model for circular data which exhibits preference in certain direction in the plane. An important member from this family is the von Mises distribution. A distribution is symmetric unimodal with mode at $\theta = \theta_o$ if and only if (i) $f(\theta_o) > f(\theta)$ for $\theta \neq \theta_o$ and (ii) $f(\theta-\theta_o) = f(\theta_o-\theta)$ with the convention that $f(\theta_1) = f(\theta)$ if $\theta_1 \equiv \theta \pmod{2\pi}$. For each θ define $\theta' \in (-\pi,\pi]$ such that $\theta' \equiv (\theta-\theta_o) \pmod{2\pi}$. With the help of such transformation, we can shift the modal direction to zero. Therefore, without loss of generality, we shall assume that for any symmetric unimodal distribution, $\theta_o = 0$.

§3.2 Methods for Circular Distributions

In this section, we will discuss procedures for the generation of variates from some basic circular distributions like the lattice distributions, the wrapped normal distribution, the wrapped Cauchy distribution, the wrapped Poisson distributions, the triangular distribution, the cardioid distribution, the angular Gaussian distribution and the von Mises distribution. Of all these distributions, only the wrapped Poisson and the lattice distributions are discrete, others are continuous. And all but the wrapped Poisson, the angular Gaussian and the lattice distributions are symmetric unimodal.

With the exception of the von Mises distribution, the generation of variates from all these practically encountered distributions is very straightforward and their algorithms which we are going to discuss are also efficient enough. On the other hand, generation from the von Mises distribution is not that straightforward. Many attempts have been tried to seek an efficient algorithm. Some have been proposed and proved to work well. We will discuss these methods later and suggest another algorithm which is also simple to use and fast enough.

3.2.1 Lattice Distributions

Mardia (1972) defined the lattice distribution as a discrete distribution with

$$\Pr(\theta = -\pi + 2\pi r/m) = p_r, \qquad r = 1, 2, \ldots, m$$

where

$$p_r \geq 0 \text{ for all } r \quad \text{and} \quad \sum_{r=1}^{m} p_r = 1.$$

Note that the mass points $-\pi + 2\pi r/m$ are located at equal distance on the unit circle.

Since there are finite number (m) of mass points, an efficient procedure known as the alias rejection method devised by Walker (1977) and modified by Kronmal and Peterson (1979) can be used. The method is useful for general distributions p_r. In particular, if $p_r = 1/m$, the algorithm is simply

$$\Theta = -\pi + 2\pi([mU] + 1)/m,$$

where $U \sim U(0,1)$ and $[x]$ denotes the integral part of x.

3.2.2 Triangular Distributions

It is a symmetric unimodal distribution having p.d.f.

$$f(\theta) = (4 + \pi^2\rho - 2\pi\rho|\theta|)/8\pi, \qquad \theta \in (-\pi,\pi], \qquad \rho \in [0,4/\pi^2],$$

with modal direction at $\theta = 0$. Note that f can be written as the following mixture

$$f(\theta) = (1 - \rho\pi^2/4)\,(1/2\pi) + (\rho\pi^2/4)\,((\pi - |\theta|)/\pi^2)$$

$$= p\, f_1(\theta) + (1-p)\, f_2(\theta)$$

where

$$p = 1 - \rho\pi^2/4,$$

$$f_1(\theta) = 1/2\pi,$$

$$f_2(\theta) = (\pi - |\theta|)/\pi^2,$$

with $f_1(\theta)$ the p.d.f. of $U(0,2\pi)$ and $f_2(\theta)$ the p.d.f. of the difference of two independent $U(0,\pi)$. Thus, the generating algorithm is based on composition method and it may be stated as follows:

With probability p, generate $U_1 \sim U(0,1)$ and set $\Theta = \pi(2U_1-1)$. Similarly, with probability 1-p, generate U_1, U_2 independent $U(0,1)$ and set $\Theta = \pi(U_1-U_2)$.

A yet faster algorithm which uses two uniform random numbers is suggested here.

Algorithm TG

1. Generate U_1, U_2 independent $U(0,1)$.

2. If $U_2 < 0.5 + \frac{\pi^2\rho}{4}(0.5-U_1)$, then accept $\Theta = \pi U_1$;

   otherwise accept $\Theta = \pi(U_1-1)$.

The method used here is in part an envelope-rejection procedure, with first-stage sampling from a uniform distribution defined on $[0,\pi]$. However, if the acceptance test is not passed, the 'rejected' variate undergoes a transformation, the transformed variate being accepted. The above algorithm is explained as follows: a point $(X,Y) = (\pi U_1, U_2/\pi)$, where U_1 and U_2 are independent from $U(0,1)$, is generated uniformly in the square $\{(x,y): x \in [0,\pi], y \in [0,1/\pi]\}$. The ordinate X is accepted as Θ if $Y < f(X)$, otherwise $\Theta = X - \pi$ is accepted.

3.2.3 The Cardioid Distribution

Cardioid distribution has p.d.f.

$$f(\theta) = (1 + 2\rho\cos\theta) / 2\pi, \qquad \theta \in (-\pi,\pi], \qquad |\rho| < 1/2.$$

An obvious way to generate variates from f is to use envelope-rejection method with $(1 + 2|\rho|)/2\pi$ as envelope.

Algorithm CD

1. Generate U_1, U_2 independent $U(0,1)$.
2. $\Theta = \pi(2U_1 - 1)$.
3. If $U_2 \le (1 + 2\rho\cos\Theta)/(1 + 2|\rho|)$, then accept Θ;
   otherwise, go to step 1.

This method has unity sampling efficiency when ρ is zero and it decreases to the minimum 0.5 when ρ is 0.5. In general, the efficiency is $(1 + 2|\rho|)^{-1}$.

When $\rho \in (0,1/2)$, the linear envelope which uses the inequality $\cos\theta < a - b\theta$ for $a = 1.28$, $b = 0.73$, $\theta \in [0,\pi]$, improves the sampling efficiency. However, computer simulation shows that this need not do faster (indeed, for $\rho = 0.3$, it is approximately 20% slower on IBM4381) than the uniform envelope. The reason is due to more arithmetic operations involved in generating variate from the linear envelope.

3.2.4 The Angular Gaussian Distribution

The distribution is also known as the offset normal distribution and its p.d.f. can be found in Mardia (1972, p.52).

Even though its p.d.f. is complicated, generating variates from this distribution is a simple matter. Suppose $X_1 \sim N(\mu_1,\sigma_1^2)$ and $X_2 \sim N(\mu_2,\sigma_2^2)$ and the correlation between the two random variables is ρ. If (R,Θ) is the polar coordinate of (X_1,X_2), then Θ is an angular Gaussian variable with parameters μ_1, μ_2, σ_1^2, σ_2^2 and ρ. Thus the problem becomes the generation of variates from bivariate normal distribution.

Algorithm AG

1. Generate Z_1, Z_2 independent $N(0,1)$.

2. $X_1 = \mu_1 + \sigma_1 Z_1$, $X_2 = \mu_2 + \sigma_2[\rho Z_1 + (1-\rho^2)^{1/2} Z_2]$,

$T = X_1 / (X_1^2 + X_2^2)^{1/2}$.

3. When $X \geq 0$, accept $\Theta = \cos^{-1}T$; otherwise accept $\Theta = -\cos^{-1}T$.

3.2.5 Wrapped Distributions

Suppose X is a random variable defined on the real line with d.f. F(x). For each X, define $\Theta \in (-\pi,\pi]$ such that

$$\Theta \equiv X \pmod{2\pi}. \tag{3.1}$$

The transformation (3.1) defines a wrapped random variable Θ having d.f.

$$F_w(\theta) = \sum_{k=-\infty}^{\infty} [F(\theta+2\pi k) - F(2\pi k)], \qquad \theta \in (-\pi,\pi].$$

Therefore, generation of variates from wrapped distributions can be a direct application of (3.1). Let us consider some important wrapped distributions.

(w1) The Wrapped Poisson Distribution

Define the domain of Θ be $[0,2\pi)$. Like the lattice distributions, the wrapped Poisson distribution is also discrete with probability mass function (p.m.f.)

$$\Pr(\theta = 2\pi r/m) = \sum_{k=0}^{\infty} p(r+km;\, \lambda), \qquad r = 0,\ 1,\ 2, \ldots,\ m-1$$

where $p(x;\lambda) = e^{-\lambda}\lambda^x/x!$, $\lambda>0$, $x = 0, 1, 2, \ldots$.

Therefore if X is a Poisson random variable with parameter λ, then $\Theta = 2\pi X/m \pmod{2\pi}$ is distributed as wrapped Poisson distribution with parameter λ. An efficient (univariate) Poisson generator may be based on alias rejection method. Atkinson (1979a,b) compared other Poisson generators as well.

(w2) The Wrapped Normal Distribution

The wrapped normal distribution is obtained by wrapping the univariate normal distribution of zero mean. Let $X \sim N(0,\sigma^2)$. By (3.1), the p.d.f. of Θ is

$$f(\theta;\ \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \exp\{-(\theta+2k\pi)^2/2\sigma^2\}, \qquad \theta \in (-\pi,\pi]$$

The generation procedure for Θ is:

Algorithm WN

1. Generate $Z \sim N(0,1)$.
2. Find $\Theta \in (-\pi,\pi]$ that satisfies $\Theta \equiv (\sigma Z) \pmod{2\pi}$.

(w3) The Wrapped Cauchy Distribution

The p.d.f. of Cauchy distribution on the real line is

$$c(x;\ a) = a/[\pi(a^2 + x^2)], \qquad x \in (-\infty,\infty), \qquad a > 0.$$

Under the transformation (3.1), the d.f. of Θ is

$$F(\theta;\ \rho) = (2\pi)^{-1}\cos^{-1}\left[\frac{(1+\rho^2)\cos\theta - 2\rho}{1+\rho^2 - 2\rho\cos\theta}\right], \qquad \theta \in (-\pi,\pi] \qquad (3.2)$$

where $0 \le \rho = e^{-a} < 1$ (see Mardia (1972), p.56). Since $X = a\tan[\pi(U - 1/2)]$, where $U \sim U(0,1)$, is distributed as $c(x;\ a)$, (3.1) gives the desired variate.

Another approach is based on the inversion of (3.2). Best and Fisher (1979) used this approach to simulate the wrapped Cauchy distribution for the generation of variates from the von Mises distribution.

Algorithm WC

S. Set $r = (1+\rho^2)/2\rho$

1. Generate U_1, U_2 independent $U(0,1)$.
2. $z = \cos(\pi U_1)$, $\qquad f = (1+rz)/(r+z)$.
3. $\phi = \cos^{-1} f$.

4. If $U_2 < 0.5$, accept $\Theta = \phi$;   otherwise accept $\Theta = -\phi$.

Step S is a set-up step for evaluating some constants that are unaltered throughout the generation process. Instead of using (3.2) exactly, the above algorithm computes the inverse of the d.f. of the folded wrapped Cauchy distribution defined on $[0,\pi]$,

$$F^*(\theta;\ \rho) = \pi^{-1}\cos^{-1}\left[\frac{(1+\rho^2)\cos\theta - 2\rho}{1+\rho^2 - 2\rho\cos\theta}\right], \qquad \theta \in [0,\pi] \qquad (3.3)$$

Obviously, $\cos^{-1}f$ in step 3 has d.f. (3.3). Step 4 is to re-define θ on $(-\pi,\pi]$ by utilizing the fact that (3.2) is symmetric about zero.

3.2.6 The von Mises Distribution

The distribution was introduced by von Mises as early as 1918 to study the deviations of measured atomic weights from integral values. It is perhaps the most important of all circular distributions. It possesses some of the desirable properties of the univariate normal distribution (see Mardia (1972)).

The p.d.f. of the standardized (after shifting the modal direction to zero) distribution of Θ is

$$v(\theta;\ k) = [2\pi I_o(k)]^{-1}\exp(k\cos\theta), \qquad \theta \in (-\pi,\pi], \quad k > 0, \qquad (3.4)$$

where $I_o(k)$ is the modified Bessel function of the first kind of order zero. The parameter k is known as the concentration parameter. When k is large, most of the density will concentrate towards the direction $\theta = 0$.

Inversion method is unsuitable here because the inverse of the d.f. of Θ is not easily computable. Various attempts have been made by using envelope-rejection method. Seigerstetter (1974) used a uniform envelope. However, his envelope is too crude to be effective when k is large. Some better envelopes like wrapped normal, Cardioid, polynomial or piecewise linear envelope have been tried, but none is effective enough. In view of these, Best and Fisher (1979) proposed another envelope which is proportional to the wrapped Cauchy distribution and they had shown that this envelope works very well for all feasible values of k. Ulrich (1984) suggested a similar envelope which is a special case of a more general class of envelopes for distributions on the n-sphere ($n \geq 2$). We shall discuss his work later in chapter five. Here let us consider Best and Fisher's work.

Instead of dealing with (3.4) directly, Best and Fisher generate the von Mises variates from its folded distribution,

$$f(\theta;\ k) = [\pi I_o(k)]^{-1}\exp(k\cos\theta), \qquad \theta \in [0,\pi]. \tag{3.5}$$

Inversion method, as described earlier (see algorithm WC), is used to generate from the folded wrapped Cauchy distribution whose p.d.f. is

$$g(\theta;\ \rho) = \frac{1-\rho^2}{\pi(1+\rho^2-2\rho\cos\theta)}, \qquad \theta \in [0,\pi].$$

Note that the d.f. is identical to (3.3). In order to maximize the sampling efficiency, the optimal value ρ^* is determined by

$$\min_{\rho\in[0,1)}\left(\max_{\theta\in[0,\pi]} \frac{f(\theta;\ k)}{g(\theta;\ \rho)}\right) = \frac{f(\theta^*;\ k)}{g(\theta^*;\ \rho^*)},$$

with the pair (θ^*,ρ^*) given by

$$\cos\theta^* = (1+\rho^{*2}-2\rho^*/k)\ /\ 2\rho^*$$

and

$$\rho^* = [\tau - \sqrt{2\tau}]\ /\ 2k,$$

where $\tau = 1 + (1+4k^2)^{1/2}$. The envelope is thus found to be

$$[k(1-\rho^{*2})\ I_o(k)\ /\ 2\rho^*]^{-1}\ \exp(k\cos\theta^*)\ g(\theta;\ \rho^*).$$

The complete algorithm is as follows:

Algorithm VMBF

S. Set $\tau = 1+(1+4k^2)^{1/2}$, $\rho^* = [\tau-\sqrt{2\tau}] / 2k$, $r = (1+\rho^{*2}) / 2\rho^*$.

1. Generate U_1, U_2 independent $U(0,1)$.
2. $z = \cos(\pi U_1)$, $f = (1+rz)/(r+z)$, $c = k(r-f)$.
3. If $U_2 < c(2-c)$, then go to step 5.
4. If $\ln(c/U_2) + 1 - c < 0$, then go to step 1.
5. $\Theta = \cos^{-1} f$.
6. Generate $U_3 \sim U(0,1)$.
7. If $U_3 < 0.5$, then accept $\Theta = -\Theta$.

Step 3 is a pre-test (known as the 'squeeze' test) of step 4. It uses the fact that $e^x \geq 1+x$ to avoid, at least some of the time, the use of logarithm in step 4. The sampling efficiency is

$$(1-\rho^{*2}) I_o(k) \Big/ \{(2\rho^*/k) \exp[k(1 + \rho^{*2})/2\rho^* - 1]\},$$

which tends to unity as $k \to 0$, and tends to the minimum $(2\pi/e)^{-1/2} = 0.658$ as $k \to \infty$. Therefore, VMBF is efficient for all values of k.

Dagpunar (1983, 1990) proposed another algorithm for the von Mises distribution using Forsythe's (1972) rejection method. In his experiment, the proposed algorithm is uniformly faster than Best and Fisher's method for all $k > 0$ when k is fixed between calls. When k is re-set between calls, Dagpunar's method is fastest when $k \leq 0.5$. When $k > 0.5$, numerical integrations are necessary, making the algorithm more difficult to implement and thus less compact than Best and Fisher's method. Dagpunar's method is as follows:

The p.d.f. of the von Mises distribution $f(\theta;\, k)$ can be represented by the probability mixture

$$f(\theta;\ k) = \sum_{j=1}^{n_k} p_j \psi_j(\theta),$$

where

$$n_k = \begin{cases} 2k, & \text{when } 2k \text{ is integer} \\ [2k+1], & \text{otherwise} \end{cases}$$

$$\psi_j(\theta) = \begin{cases} \exp(k\cos\theta+j-k-1) \Big/ \int_{\theta_{j-1}}^{\theta_j} \exp(k\cos\theta+j-k-1)d\theta, & \theta \in [\theta_{j-1},\theta_j] \\ 0, & \text{elsewhere} \end{cases}$$

$$\theta_j = \cos^{-1}(1 - (j/k)), \qquad j = 0,\ 1, \ldots,\ n_k-1$$

$$\theta_{n_k} = \pi,$$

$$p_j = \int_{\theta_{j-1}}^{\theta_j} \exp(k\cos\theta)d\theta \Big/ \int_0^{\pi} \exp(k\cos\theta)d\theta, \qquad j = 1,\ 2, \ldots,\ n_k$$

and [x] denotes the integer part of x.

With probability p_j, algorithm VMD selects an interval (θ_{j-1},θ_j) and generate a von Mises variate within this interval using Forsythe's method.

Algorithm VMD (let $\{U_i\}$ be a sequence of independent U(0,1) random numbers)

S. Compute n_k, $\{\theta_j\}$, $\{p_j\}$.

1. Generate $U \sim U(0,1)$. Find smallest j such that $\sum_{i=1}^{j} p_i > U$.

2. $U = (\sum_{i=1}^{j} p_i - U) / p_j$.

3. $\Theta = \theta_{j-1} + U(\theta_j - \theta_{j-1})$.

4. $\lambda = k - j + 1 - k\cos\Theta$.

5. Set $N = 1$ if $\lambda < U_1$, otherwise

   $N = n$ if $\lambda \geq U_1 \geq U_2 \ldots \geq U_{n-1} < U_n$.

6. If N is even, generate $U \sim U(0,1)$ and then go to step 3.

7. $\Theta = \begin{cases} \mathrm{sign}((U_N - \lambda)/(1 - \lambda) - 0.5)\ \Theta, & N = 1 \\ \mathrm{sign}((U_N - U_{N-1}) / (1-U_{N-1}) - 0.5)\ \Theta, & N > 1 \end{cases}$

where sign(x) is the sign of x.

Step 2 and step 7 re-use random numbers by the conditional uniformity of U (step 2) and U_N (step 7).

Apart from VMBF and VMD, we hereby suggest a method which is also based on envelope-rejection technique and is shown to be efficient for all values of k.

A new method

It is simple to show that $\cos x \le 1 - x^2/2v$, $x \in (-\pi,\pi]$, where $v = \pi^2/4$. Using this inequality and (3.4), the p.d.f. of the von Mises distribution is

$$v(\theta;\ k) \le [2\pi I_o(k)]^{-1}\exp[k(1 - \theta^2/2v)], \qquad \theta \in (-\pi,\pi]. \tag{3.6}$$

Inequality (3.6) suggests to use an envelope which is proportional to a truncated normal density defined on $(-\pi,\pi)$ of mean zero and variance $v / k = \pi^2/ 4k$, denoted by $N_t(0,\frac{v}{k};\ -\pi,\pi)$. A way to generate variate from $N_t(0,\frac{v}{k};\ -\pi,\pi)$ is by accepting the prospective variate obtained from the corresponding normal variate defined on the whole real line if it falls in $(-\pi,\pi)$. It is simple but is ineffective when k is small, say < 0.5. An alternative way follows the idea of Box-Muller (1958) for standard normal variate. Algorithm TNORM below is the modified Box-Muller method for $N_t(0,\frac{v}{k};\ -\pi,\pi)$. It is effective even when k approaches zero.

Algorithm TNORM To generate X_1, X_2 independent $N_t(0, \frac{v}{k};\ -\pi,\pi)$.

S. Set $\sigma^2 = v/k$, $a = 4k$, $b = 1-\exp(-a)$.

1. Generate U_1, U_2 independent U(0,1).

2. $R = -2\ln(1-U_1 b)$, $\alpha = 2\pi U_2$, $c = \cos\alpha$.

3. $\begin{cases} S_1 = Rc^2. \\ S_2 = R - S_1. \end{cases}$

4. If $S_1 > a$ or $S_2 > a$, then go to step 1.

5. $\begin{cases} X_1 = (\sigma^2 R)^{1/2}\ c. \\ X_2 = (\sigma^2 R)^{1/2}\ \sin\alpha. \end{cases}$

Step 2 generates R, a truncated χ_2^2 on (0,2a) by inversion method. Note that $\sqrt{R}c$ and $\sqrt{R}\sin\alpha$ in step 5 are independent $N_t(0,1; -\sqrt{a},\sqrt{a})$. In practice, TNORM will return one variate each time the algorithm is called and store up the other for the next call. The complete algorithm for von Mises using (3.6) is therefore:

Algorithm VMTN

S. Set $v = \pi^2/4$.

1. Generate $U \sim U(0,1)$.

2. Generate $X \sim N_t(0,\frac{v}{k}; -\pi,\pi)$ by TNORM.

3. $T = 1 + k(\cos X - 1 + X^2/2v)$.

4. If $U \le T$, then go to step 6.

5. If $U > \exp(T-1)$, then go to step 1.

6. Accept $\Theta = X$.

Step 4 is the pre-test of step 5 using $e^x \ge 1+x$. Table 3.1 gives the mean requirements of random numbers per generated von Mises variate in VMBF and VMTN and table 3.2 shows the timings for simulating a sample of size 10000 for selected values of k (fixed and re-set between calls).

Table 3.1 The mean requirement of random numbers to generate one von Mises variate

	k							
	0	0.1	0.5	1	2	5	10	50
VMBF	3	3.01	3.11	3.30	3.61	3.88	3.96	4.03
VMTN	2.57	2.56	2.57	2.67	2.87	3.05	3.10	3.13

Table 3.2 The mean time (μs) to generate a von Mises variate when k is fixed / re-set (bracketed numbers) between calls

	k							
	0	0.1	0.5	1	2	5	10	50
VMBF	134	135	141	152	170	187	189	192
	(172)	(170)	(176)	(189)	(207)	(222)	(226)	(228)
VMTN	141	143	149	158	172	185	192	190
	(160)	(161)	(170)	(186)	(202)	(216)	(219)	(203)

The timings demonstrate that VMTN and VMBF are very similar in speed when either k is fixed or re-set between calls. VMBF is slightly faster when k is fixed whereas the reverse is true when k is re-set between calls. The change can be explained by the longer set-up time for the evaluation of two square roots in VMBF, compare to one exponentiation in VMTN. Note also from table 3.1 that VMTN requires fewer random numbers than VMBF. Therefore slower random number generator will increase the speed of VMTN relative to VMBF.

Chapter Four

GENERATING VARIATES FROM NON-UNIFORM DISTRIBUTIONS ON THE SPHERE

§4.1 Introduction

Let $\mathbf{X} = (X_1,X_2,X_3)^T$, for which $X_1^2 + X_2^2 + X_3^2 = 1$ be the Cartesian coordinates of an observation on the surface of a 3-sphere (or simply sphere). The direction of $\mathbf{X}$ can be specified by the polar coordinates (Θ,Φ) with the following relations

$$X_1 = \sin\Theta\cos\Phi, \qquad X_2 = \sin\Theta\sin\Phi, \qquad X_3 = \cos\Theta,$$

where $\Theta \in (0,\pi)$, $\Phi \in (0,2\pi)$. Let the joint p.d.f. of (Θ,Φ) be $f(\theta,\phi)$. Computer simulation of spherical data aims at producing realizations of (Θ,Φ) from f. In practice, there are four main types of spherical distributions, namely, the uniform distribution, the unimodal distributions, the bimodal distributions and the girdle distributions. For a thorough discussions of some of these distributions, one is recommended to refer to Mardia's (1972) and Watson's (1983) texts. Both give detailed mathematical treatments on the various distributions on the sphere. Fisher et al (1987) also offered an overview and statistical methods to most popular spherical distributions.

Owing to an extra dimension, variates generation in the spherical case is usually not as easy as in the circular case. For some distributions like the Fisher distribution and the Dimroth-Watson distribution where the random vector (Θ,Φ) are independent, each component can be generated separately. However, for distributions like Bingham distribution where dependence exists between Θ and Φ, the generation procedure may be more complicated.

In addition, the complexity of a distribution can make variate generation from that distribution difficult to achieve. Therefore a distribution involving many parameters is always very difficult to simulate. Even if a procedure exists for that distribution, it may work well only for some limited parameter values. Sometimes only when the parameter values are restricted or when there are constraints on the parameters will allow an algorithm to be useful.

§4.2 Methods for Spherical Distributions

In this section, we will devise algorithms for the generation of commonly encountered spherical distributions like the Fisher distribution, Arnold distribution, Selby distribution, Dimroth-Watson distribution, Bingham distribution and Bingham-Mardia distribution. With the exception of Fisher distribution, new algorithms are developed and will be compared with existing algorithms, if any, to demonstrate the advantages of the new methods.

4.2.1 Fisher Distribution

It was for the investigation of certain statistical properties of palaeomagnetism that Fisher distribution was studied by Fisher (1953). This important distribution is unimodal with rotational symmetry and serves as an all-purpose probability model for directions in $\mathbb{R}^3$, much as the von Mises distribution in $\mathbb{R}^2$ and the normal distribution in $\mathbb{R}$. In the standardized form in which the modal direction is at (0,0), the p.d.f. of (Θ,Φ) is

$$f(\theta,\phi;\ k) = C_F \exp(k\cos\theta)\sin\theta, \qquad \theta \in [0,\pi], \qquad \phi \in [0,2\pi),$$

where

$$C_F = \frac{k}{4\pi\sinh k}$$

is the normalizing constant. The modal direction is at $\theta = 0$. The distribution has one parameter $k > 0$ which is known as the concentration parameter because the larger the value of k, the higher the probability density around the direction $\theta = 0$.

Since the density does not involve Φ, Φ has a uniform distribution on $[0,2\pi)$. For large k, Mardia (1972) has shown that $k\Theta^2$ is approximately distributed as χ^2_2. An approximate method for generating Θ which simply involves the generation of a χ^2 random variable is thus obtained when k is large. However as approximate method is usually not desirable, a demand for exact method is expected. An exact method can be obtained by inversion method.

Note that the marginal density of Θ is

$$f_\Theta(\theta;\, k) = \frac{k}{2\sinh k} \exp(k\cos\theta)\sin\theta, \qquad \theta \in (0,\pi)$$

Let $X = \cos\Theta$. Then the p.d.f. of X is

$$f_X(x;\, k) = \frac{k}{2\sinh k} \exp(kx), \qquad x \in (-1,1)$$

and the corresponding d.f. is

$$F_X(x;\, k) = (e^{kx} - e^{-k})/(e^{k} - e^{-k}).$$

Upon the inversion, $X = F_X^{-1}(U)$ where $U \sim U(0,1)$, we obtain the following procedure for generating (Θ,Φ):

Algorithm FSH

S. Set $\lambda = e^{2k}-1$.

1. Generate U_1, U_2 independent $U(0,1)$.

2. $X = -1 + k^{-1}\ln(\lambda U_1+1)$.

3. $\Theta = \cos^{-1}X$.

4. $\Phi = 2\pi U_2$.

Fisher et al (1981), on the other hand, considered the transformation $Y = \exp[-2k\sin^2(\Theta/2)]$. Thus, the density of Y is

$$f_Y(y;\ k) = f_\Theta(\theta;\ k)\ \left|\frac{d\theta}{dx}\right|$$

$$= (1-e^{-2k})^{-1}, \qquad\qquad y \in (e^{-2k},1)$$

i.e. Y is distributed uniformly on $(e^{-2k},1)$. The algorithm is

Algorithm FSHR

S. Set $\lambda = e^{-2k}$.

1. Generate U_1, U_2 independent U(0,1).

2. $Y = \lambda + (1-\lambda)U_1$.

3. $\Theta = 2\sin^{-1}\left(\sqrt{\frac{-1}{2k}\ \ln Y}\ \right)$.

4. $\Phi = 2\pi U_2$.

Algorithm FSHR is a more accurate algorithm than FSH as the latter induces appreciable rounding error when k is large.

4.2.2 Arnold Distribution

Arnold distribution is a rotational symmetric distribution which was introduced by Arnold (1941) and discussed by Selby (1964). It is a girdle distribution which has its density concentrated around the great circle (or the equator), $\theta = \pi/2$. Its standardized p.d.f. is

$$f(\theta,\phi;\ k) = C_A\exp(-k|\cos\theta|)\sin\theta, \qquad k > 0, \quad \theta \in [0,\pi], \quad \phi \in [0,2\pi),$$

where

$$C_A = \frac{k}{4\pi(1-e^{-k})}$$

is the normalizing constant. The parameter k is the concentration parameter.

When k is large, the distribution becomes more concentrated around the equator.

Since Φ is independent of Θ, it can be generated as $2\pi U$, where $U \sim U(0,1)$. Inversion method can be applied for Θ. As the marginal d.f. of Θ is

$$F_\Theta(\theta;\ k) = \begin{cases} (e^{-k\cos\theta} - e^{-k}) \,/\, 2(1 - e^{-k}), & \theta \in [0,\pi/2] \\ 1/2 + (1 - e^{k\cos\theta}) \,/\, 2(1 - e^{-k}), & \theta \in (\pi/2,\pi] \end{cases}$$

its inverse can be readily obtained.

Algorithm ARND

S. Set $B = 1 - e^{-k}$.

1. Generate U_1, U_2 independent $U(0,1)$.

2. If $U_1 > 0.5$, then accept $\Theta = \cos^{-1}\{\ k^{-1}\ln[1+(1-2U_1)B]\ \}$;

   otherwise accept $\Theta = \cos^{-1}\{\ -k^{-1}\ln[1-(1-2U_1)B]\ \}$.

3. $\Phi = 2\pi U_2$.

4.2.3 Selby Distribution

Like the Arnold distribution, it is a girdle distribution with rotational symmetry. The distribution was introduced by Selby (1964) to model axial data. Its standardized p.d.f. is

$$f(\theta,\phi;\ k) \propto \exp(k\sin\theta)\sin\theta,$$

where $k > 0$, $\theta \in [0,\pi]$, $\phi \in [0,2\pi)$. The marginal density of Θ is

$$f_\Theta(\theta;\ k) \propto \exp(k\sin\theta)\sin\theta.$$

Since the inverse of the d.f. of Θ is not easily computable, we resort to envelope-rejection method to generate Θ. Two envelopes are proposed here, each based on a simple inequality:

(S1) Use of the inequality: $\cos\theta + \sin\theta \le \sqrt{2}$, $\theta \in [0,\pi/2]$.

Since Θ is symmetric about $\pi/2$, we restrict Θ to $[0,\pi/2]$. We have

$$\exp(k\sin\theta)\sin\theta \le \exp(k(\sqrt{2}-\cos\theta))\sin\theta, \qquad \theta \in [0,\pi/2].$$

Therefore the envelope for f_Θ is proportional to $\exp(-k\cos\theta)\sin\theta$ which is essentially the folded Fisher's density and hence variate from this envelope can be simulated easily. Unfortunately, the envelope based on the inequality (S1) is not good enough as its sampling efficiency

$$\frac{ke^{-\sqrt{2}k}}{1-e^{-k}}\int_0^1 \exp(k(1-x^2)^{1/2})dx$$

tends to zero as k approaches ∞.

(S2) Use of the inequality: $(1-x^2)^{1/2} \le 1 - \frac{1}{2}x^2$ $\qquad x \in [-1,1]$

Consider the transformation $X = \cos\Theta$. The density of X is proportional to

$$\exp(k(1-x^2)^{1/2}), \qquad x \in [-1,1]. \tag{4.1}$$

By inequality (S2), we have

$$\exp(k(1-x^2)^{1/2}) \le \exp(k(1 - \tfrac{1}{2}x^2)), \qquad x \in [-1,1].$$

Since $k > 0$, the envelope for X having density (4.1) is proportional to $N_t(0,\frac{1}{k};\ -1,1)$, the truncated normal distribution defined on $(-1,1)$ of mean zero and variance $1/k$. Generating variates from $N_t(0,\frac{1}{k};\ -1,1)$ is simple. Recall in chapter three that an efficient algorithm TNORM is proposed for $N_t(0,\frac{\pi^2}{4k};\ \pi,\pi)$ for simulating von Mises variates. A slight modification of TNORM will make it suitable for $N_t(0,\frac{1}{k};\ -1,1)$.

Algorithm TNRML To generate X_1, X_2 independent $N_t(0,a^{-2};\ -1,1)$.

S. Set $a^2 = k$, $\qquad b = 1 - \exp(-a^2)$.

1. Generate U_1, U_2 independent $U(0,1)$.

2. $R = -2\ln(1-U_1 b)$, $\alpha = 2\pi U_2$, $c = \cos\alpha$.

3. $\begin{cases} S_1 = Rc^2. \\ S_2 = R-S_1. \end{cases}$

4. If $S_1 > a^2$ or $S_2 > a^2$ go to step 1.

5. $\begin{cases} X_1 = (R/a^2)^{1/2}c \\ X_2 = (R/a^2)^{1/2}\sin\alpha. \end{cases}$

Therefore, if X^* is distributed as $N_t(0,\frac{1}{k}; -1,1)$, the conditional p.d.f. of X^* given

$$U \le \frac{\exp(k(1-X^{*2})^{1/2})}{\exp(k(1-\frac{1}{2}X^{*2}))},$$

where $U \sim U(0,1)$ is independent of X^*, is proportional to (4.1). An algorithm suggested for (Θ,Φ) is

Algorithm SLBY

1. Generate $U_1 \sim U(0,1)$.

2. Generate $X \sim N_t(0,\frac{1}{k}; -1,1)$ by TNRML.

3. $T = 1 + k[(1-X^2)^{1/2}-(1 - \frac{1}{2}X^2)]$.

4. If $U \le T$, then go to step 6.

5. If $U > \exp(T-1)$, then go to step 1.

6. $\Theta = \cos^{-1}X$.

7. Generate $U_2 \sim U(0,1)$ and form $\Phi = 2\pi U_2$.

The mean requirement of random numbers used to generate one vector (Θ,Φ) by SLBY is obtained by the formula

$$N_e = 1 + (1 + \frac{1}{p})r^{-1},$$

where

$$p = \Pr(S_1 < k \quad \text{and} \quad S_2 < k \quad \text{in step 4 of TNRML})$$

$$= \frac{2}{\pi(1-e^{-k})} \left(\int_0^{\sqrt{k}} \exp(-\tfrac{1}{2}z^2)dz \right)^2$$

and

$$r = \Pr(X \text{ is accepted in algorithm SLBY})$$

$$= \int_0^1 \exp(k(1-x^2)^{1/2})dx \Big/ [e^k \int_0^1 \exp(-\tfrac{k}{2}x^2)dx].$$

Table 4.1 lists N_e for some values of k.

Table 4.1 The mean requirement of random numbers to generate one vector from Selby distribution using SLBY

	k						
	0	0.5	1	2	5	10	∞
N_e	3.57	3.5	3.44	3.34	3.17	3.08	3

Table 4.1 shows that N_e is quite stable to the value of k. An average of slightly more than three random numbers used to generate (Θ,Φ) make SLBY an efficient algorithm for Selby distribution.

4.2.4 Dimroth-Watson Distribution

The distribution is sometimes called the Scheidegger-Watson distribution or simply Watson distribution. We follow Mardia's usage in his influential book (1972) and call it Dimroth-Watson distribution.

This distribution is used as an important model for axial data distributed with rotational symmetry in either bipolar or girdle form. Its standardized p.d.f. is

$$f(\theta,\phi;\ k) \propto \exp(k\cos^2\theta)\sin\theta, \qquad \theta \in [0,\pi], \qquad \phi \in [0,2\pi) \qquad (4.2)$$

where k is the scale parameter which may either be positive or negative; when $k > 0$, the distribution is bipolar such that the distribution has highest density at its two poles $\theta = 0$ and $\theta = \pi$, and when $k < 0$, it is girdle with density concentrated around the equator $\theta = \pi/2$.

Since the d.f. of Dimroth-Watson distribution does not have a closed form and there are no convenient transformations of other distributions, Best and Fisher (1986) devised algorithms based on envelope-rejection method to overcome these difficulties. Their algorithm considers two cases: (i) the bipolar case, $k > 0$ and (ii) the girdle case, $k < 0$. Their methods are discussed here.

(i) Bipolar case, $k > 0$

As Θ is symmetric about $\pi/2$, consider $\Theta \in [0,\pi/2]$ and put $X = \cos\Theta$. The envelope used is proportional to the density Ce^{kx}, $x \in [0,1]$ where $C = k(e^k-1)^{-1}$, which is easily generated by inversion method.

Algorithm DWBF ($k > 0$)

S. Set $C = 1/(e^k-1)$.

1. Generate U_1, U_2 independent $U(0,1)$.
2. $X = k^{-1}\ln(1 + U_1/C)$.
3. If $U_2 \le 1 + k(X^2-X)$, then go to step 5.
4. If $U_2 > \exp(k(X^2-X))$, then go to step 1.
5. $\Theta = \cos^{-1}X$.
6. Generate $U_3 \sim U(0,1)$.
7. If $U_3 < 0.5$, then $\Theta = \pi - \Theta$ and $\Phi = 4\pi U_3$;

   otherwise $\Phi = 2\pi(2U_3-1)$.

(ii) Girdle case, $k < 0$

Again put $X = \cos\Theta$ for $\Theta \in [0,\pi/2]$ and take the envelope proportional to the density $C(1+\xi^2x^2)^{-1}$, $x \in [0,1]$ where $\xi = \sqrt{-k}$ and $C = \xi(\tan^{-1}\xi)^{-1}$, which can be generated conveniently by inversion method.

Algorithm DWBF $(k < 0)$

S. Set $\xi = \sqrt{-k}$, $\quad B = \tan^{-1}\xi$.

1. Generate U_1, U_2 independent $U(0,1)$.
2. $X = \tan(BU_1)$.
3. If $U_2 \le 1 - X^4$ then go to step 5.
4. If $U_2 > (1+X^2)\exp(-X^2)$, then go to step 1.
5. $\Theta = \cos^{-1}X$.
6. Generate $U_3 \sim U(0,1)$.
7. If $U_3 < 0.5$, then $\Theta = \pi - \Theta$ and $\Phi = 4\pi U_3$;

   otherwise $\Phi = 2\pi(2U_3-1)$.

In both cases, 'squeeze' method, depends on the bound $e^x \ge 1+x$, have been used in step 3 to avoid, at least some of the time, the use of the exponential functions at step 4. Step 7 aims to re-define Θ on $[0,\pi]$ and re-uses U_3 to generate Φ by the conditional uniformity of U_3.

The sampling efficiencies for the two cases are

$$R^+(k) = k(e^k-1)^{-1} \int_0^1 \exp(ky^2)dy, \qquad k > 0$$

and

$$R^-(k) = \sqrt{\pi}(\tan^{-1}\sqrt{-k})^{-1} \, [\phi(\sqrt{-2k}) - \tfrac{1}{2}], \qquad k < 0$$

respectively, where $\phi(x)$ is the d.f. of the standard normal distribution. Both $R^+(k)$ and $R^-(k)$ are decreasing functions in k with $R^+(0) = R^-(0) = 1$, $R^+(\infty) = 0.5$ and $R^-(-\infty) = 0.56$. The mean requirements for random numbers in each case are $1 + 2/R^+$ $(k > 0)$ and $1 + 2/R^-$ $(k < 0)$ respectively.

A new algorithm

Here we propose an envelope which can handle both the bipolar case and the girdle case. It is very easy to generate variates from this envelope and the new algorithm based on this envelope is proved to be more efficient than Best and Fisher's and it is also more compact. The envelope we use can also be applied to other distributions which have similar forms as the Dimroth-Watson distribution. The results are encouraging and we will investigate the applicability of the envelope to those distributions later.

Put $X = \cos\Theta$ and consider $\Theta \in [0,\frac{\pi}{2}]$. By (4.2), the p.d.f. of X is

$$f(x;\ k) = C_f \exp(kx^2), \qquad x \in [0,1], \qquad k \in (-\infty,\infty)$$

where $C_f = [\ \int_0^1 \exp(ku^2)du\ ]^{-1}$. Define

$$\gamma(x) = C_f \beta (1-\rho x^2)^{-3/2}, \qquad x \in [0,1]$$

where

$$\beta = \alpha^{3/2}\exp(-\tfrac{3}{2}(1 - \tfrac{1}{\alpha})), \qquad \alpha = 3\rho/2k,$$

$$\rho = [\lambda - (\lambda^2-16k)^{1/2}]/4, \qquad \lambda = 2k+3.$$

In Appendix 2 it is shown that $\gamma(x)$ is an envelope for $f(x;\ k)$ and that the choice of ρ maximizes the sampling efficiency. Note that $\gamma(x)$ is proportional to the p.d.f.

$$g(x;\ \rho) = C_g (1-\rho x^2)^{-3/2}, \qquad x \in [0,1]$$

where $C_g = \sqrt{1-\rho}$. Sample from $g(x;\ \rho)$ is obtainable by inversion method. If a random variable Y is defined by

$$Y = \frac{U}{[1 - \rho(1-U^2)]^{1/2}}, \qquad \text{where } U \sim U(0,1),$$

then Y is distributed as $g(x;\ \rho)$. Suppose $X^* \sim g(x;\ \rho)$. Then if

$$U \le f(X^*;\ k) \,/\, \gamma(X^*)$$

$$= \beta^{-1}(1-\rho X^{*2})^{3/2}\exp(kX^{*2}), \tag{4.3}$$

where $U \sim U(0,1)$ independent of X^*, the conditional p.d.f. of X^* given that (4.3) holds is $f(x;\ k)$. Below is the algorithm for Dimroth-Watson distribution using envelope $\gamma(x)$.

Algorithm DWAG

S. Set $\lambda = 2k+3$, $\rho = [\lambda - (\lambda^2-16k)^{1/2}]/4$,

$\alpha = 3\rho/2k$, $q = \alpha^3\exp(-3 + 3/\alpha)$.

1. Generate U_1, U_2 independent $U(0,1)$.

2. $S = U_1^2 \,/\, [1-\rho(1-U_1^2)]$.

3. $W = 1 + kS$.

4. If $W > 0$ and $U_2^2 \le (1-\rho S)^3 W^2/q$, then go to step 6.

5. If $U_2^2 > (1-\rho S)^3\exp(2(W-1))/q$, then go to step 1.

6. $\Theta = \cos^{-1}\sqrt{S}$.

7. Generate $U_3 \sim U(0,1)$.

8. If $U_3 < 0.5$, then $\Theta = \pi - \Theta$ and $\Phi = 4\pi U_3$;

   otherwise $\Phi = 2\pi(2U_3-1)$.

Instead of performing the acceptance test directly as in (4.3), we square both sides of (4.3), as shown in step 5, so as to avoid performing fractional exponentiations. Step 8 re-defines Θ on $[0,\pi]$ and re-uses U_3 to generate Φ by the conditional uniformity of U_3. The sampling efficiency is

$$R = \left[\int_0^1 \gamma(x)dx \right]^{-1}$$

$$= \left(\frac{2k}{3\rho}\right)^{3/2} \sqrt{1-\rho} \exp\left(\frac{3}{2} - \frac{k}{\rho}\right) \int_0^1 \exp(ku^2)du$$

and, on the average, DWAG requires 1 + 2/R random numbers to generate a vector (Θ,Φ) from Dimroth-Watson distribution. Table 4.2a and 4.2b show the sampling efficiencies and mean requirements of random numbers for Best and Fisher's algorithms DWBF and DWAG. Before looking at the tables and considering the empirical comparisons between these algorithms, let us investigate another algorithm for the girdle Dimroth-Watson distribution.

Table 4.2a The mean requirement of random numbers to generate one vector from Dimroth-Watson distribution for the bipolar case using DWBF and DWAG (Sampling efficiencies in parentheses)

	k					
	0	0.5	2	5	10	50
DWBF	3 (1)	3.17 (0.92)	3.70 (0.74)	4.43 (0.58)	4.77 (0.53)	4.96 (0.51)
DWAG	3 (1)	3.02 (0.99)	3.28 (0.88)	3.98 (0.67)	4.42 (0.58)	4.75 (0.53)

Table 4.2b The mean requirement of random numbers to generate one vector from Dimroth-Watson distribution for the girdle case using DWBF, DWAG and DWTN (Sampling efficiencies in parentheses).

	k					
	0	-0.5	-2	-5	-10	-50
DWBF	3 (1)	3.04 (0.98)	3.26 (0.89)	3.60 (0.77)	3.85 (0.70)	4.23 (0.62)
DWAG	3 (1)	3.01 (0.99)	3.12 (0.94)	3.29 (0.87)	3.40 (0.84)	3.49 (0.80)
DWTN	2.57	2.36	2.08	2	2	2

When $k < 0$, the p.d.f. of $X = \cos\Theta$ for $\Theta \in [0,\pi]$ is in fact the truncated normal distribution defined on $[-1,1]$, of mean zero and variance $-1/2k$, that is, $X \sim N_t(0,\frac{-1}{2k}; -1,1)$. We have discussed the generation of $N_t(0,\frac{1}{k}; -1,1)$ before (see Selby distribution). Thus, using this result, the algorithm TNRML can be used with only the change $a^2 = -2k$ in the set-up step. An algorithm for the girdle case is

Algorithm DWTN

1. Generate $U \sim U(0,1)$
2. Generate $X \sim N_t(0,\frac{-1}{2k}; -1,1)$ by TNRML.
3. $\Theta = \cos^{-1}X$, $\qquad \Phi = 2\pi U$.

The average number of random numbers used to generate a vector (Θ,Φ) is given by $1+1/p$ (see table 4.2b for some selected values of k) where

$$p = \Pr(S_1 < k \text{ and } S_2 < k \text{ in step 4 of TNRML} \mid a^2 = -2k)$$

$$= \frac{2}{\pi(1-e^{2k})}\left(\int_0^{\sqrt{-2k}} \exp(-\tfrac{1}{2}z^2)dz\right)^2.$$

Now it is time to compare the performances of the various algorithms that we have already discussed for the generation of variates from the Dimroth-Watson distribution. It is seen from table 4.2a and b that the proposed algorithm DWTN uses the fewest random numbers on the average to simulate a realization from the girdle Dimroth-Watson distribution while the proposed algorithm DWAG, in both the bipolar and girdle situations, uses fewer random numbers uniformly than DWBF for all values of k. Since the complexity of all algorithms are very similar, it is expected that the execution speed of DWAG will be faster than DWBF while DWTN is the fastest for k < 0. As the magnitude of k increases, the demand for random numbers increases gradually for DWBF and DWAG but not for DWTN which shows a gentle decrease instead. This is explained by the fact that the sampling efficiencies of DWBF and DWAG are decreasing functions in |k| and that for large |k|, DWTN becomes more efficient as generation from $N_t(0,\frac{-1}{2k}; -1,1)$ by TNRML is most effective when |k| is large. When k > 0, both the efficiencies of DWBF and DWAG converge to 0.5 (empirically for DWAG) as $k \to \infty$. When k < 0, the efficiency for DWBF converges to 0.56 and that for DWAG approximately to 0.78 as $k \to -\infty$. Note that the discrepancy between the two efficiencies is negligible when |k| is small (say < 2). Table 4.3a & b below show the marginal generation time for the three algorithms by simulating a sample of size 10000.

Table 4.3a Marginal generation time (μs) for the bipolar Dimroth-Watson distribution

	k						
	0.1	0.5	1	2	5	10	50
DWBF	144	149	162	181	224	242	256
DWAG	147	146	155	167	210	237	250

Table 4.3b Marginal generation time (μs) for the girdle Dimroth-Watson distribution

	k						
	-0.1	-0.5	-1	-2	-5	-10	-50
DWBF	137	138	140	148	162	174	193
DWAG	143	143	146	150	159	162	164
DWTN	145	136	131	122	115	114	114

As expected, DWTN is the fastest in the girdle case and its execution time decreases as |k| increases owing to the fewer random numbers used on the average when $|k|$ is large. It is also observed that when $|k| > 2$, DWAG is uniformly faster than DWTN. In fact the speed between DWBF and DWAG do not differ very much for all values of k. In conclusion, DWAG is a more versatile and efficient (slightly) algorithm than DWBF. In the girdle case, even more efficient algorithm DWTN is available.

4.2.5 Bingham Distribution

This is an antipodally symmetric distribution introduced by Bingham (1964, 1974). It serves as a multipurpose model for axial data. Its standardized p.d.f. is

$$\begin{aligned} f(\theta,\phi;\ k_1,k_2) &= C(k_1,k_2)\ \exp[(k_1\cos^2\phi+k_2\sin^2\phi)\sin^2\theta]\sin\theta \\ &\equiv C(k_1,k_2)\ e(\theta,\phi) \end{aligned} \tag{4.4}$$

where $\theta \in [0,\pi]$, $\phi \in [0,2\pi)$ and $C(k_1,k_2)$ is the normalizing constant. The standardized distribution has two parameters k_1 and k_2. When $k_1 = k_2$, the p.d.f. (4.4) reduces to the Dimroth-Watson distribution. Other values of k_1 and k_2 may give uniform distribution ($k_1 = k_2 = 0$), symmetric and asymmetric girdle distributions and bimodal distributions. In general, the random angles Θ and Φ are dependent. This makes sampling from the distribution more difficult.

Wood (1987) considered the simulation of a special case ($k_1 = -k_2$) of Bingham distribution. Here, we make use of his idea to consider the general case.

In Cartesian coordinates, the Bingham p.d.f. can be written as

$$f(\mathbf{x};\ k_1,k_2) = C(k_1,k_2)\ \exp(k_1x_1^2 + k_2x_2^2)$$

where $\mathbf{X} = (X_1,X_2,X_3)^T$, $X_1^2 + X_2^2 + X_3^2 = 1$. Let $T = X_3$ and put $X_1 = (1-T^2)^{1/2}\cos\Phi$, $X_2 = (1-T^2)^{1/2}\sin\Phi$. The marginal p.d.f. of T is

$$g(t;\ k_1,k_2) = \int_0^{\pi} C(k_1,k_2)\ \exp[k_1(1-t^2)\cos^2\phi + k_2(1-t^2)\sin^2\phi]\ d\phi$$

$$= 2\pi I_o[\tfrac{1}{2}(k_1-k_2)(1-t^2)]\ C(k_1,k_2)\exp[\tfrac{1}{2}(k_1+k_2)(1-t^2)],$$

where $t \in [-1,1]$ and $I_o(x)$ is the modified Bessel function of the first kind of order zero. We call T the mixture variable and $g(t;\ k_1,k_2)$ the mixture density. It follows that the conditional density of 2Φ given $T = t$ is

$$\left(2\pi I_o[\tfrac{1}{2}\ (k_1-k_2)(1-t^2)] \right)^{-1} \exp[\tfrac{1}{2}(k_1-k_2)(1-t^2)\cos 2\phi],$$

i.e, $2\Phi|T = t$ has a von Mises distribution with concentration parameter $\frac{1}{2}(k_1-k_2)(1-t^2)$. Thus an algorithm for generating variates from (4.4) is developed:

Algorithm BH

1. Generate T from the mixture density $g(t;\ k_1,k_2)$.
2. $\Theta = \cos^{-1}T$.
3. Generate $\delta \sim$ Bernoulli(0.5).
4. Generate $\psi \in (-\pi,\pi)$ from the von Mises distribution with concentration parameter $\frac{1}{2}(k_1-k_2)(1-T^2)$.
5. If $\psi < 0$, then accept $\Phi = (1+\delta)\pi + \psi/2$; otherwise accept $\Phi = \delta\pi + \psi/2$.

Efficient methods have been discussed for the von Mises distribution in chapter three. The problem remains the generation of the mixture variable T.

By the fact that $I_o(x) \le \cosh x$ for all x, we obtain

$$g(t;\ k_1,k_2) \ \le\ a^{-1}[p\ g(t;\ k_1,k_1) + (1-p)\ g(t;\ k_2,k_2)] \qquad (4.5)$$

where $p = C(k_2,k_2)/[C(k_1,k_1) + C(k_2,k_2)]$ and $a = 2pC(k_1,k_1)/C(k_1,k_2)$.

Note that:

(i) $g(t;\ k,k)$ is the marginal Dimroth-Watson p.d.f. with concentration parameter $-k$.

$$\text{(ii) } C(k,k) = \begin{cases} [\ 4\pi\, D(\sqrt{-k})/\sqrt{-k}\ ]^{-1}, & k < 0 \\ [\ 2\pi^{3/2}e^{k}\ \mathrm{erf}(\sqrt{k})/\sqrt{k}\ ]^{-1}, & k > 0 \end{cases}$$

where $D(x) = \exp(-x^2)\int_0^x \exp(u^2)du$ is the Dawson's integral and $\mathrm{erf}(x) = (2/\sqrt{\pi}) \int_0^x \exp(u^2)du$ is the error function. Routines for calculating $D(x)$ and $\mathrm{erf}(x)$ are available in the IMSL Sfun/Library (1987). Then an algorithm to generate T from the mixture density based on (4.5) is:

1. Generate $U \sim U(0,1)$.
2. With probability p (or (1-p)), generate T from $g(t;\ k_1,k_1)$ (or $g(t;\ k_2,k_2)$) and accept the prospective variate if $U \le I_o(v)/\cosh v$, where $v = \frac{1}{2}(k_1-k_2)(1-T^2)$.

Efficient algorithms like DWAG discussed earlier can be employed to generate variate from $g(t;\ k,k)$. The modified Bessel function can be evaluated by routine in the IMSL Sfun/Library.

Johnson (1987) described another method for Bingham distribution which is based on Atkinson's (1982) bipartite rejection scheme. We briefly describe Johnson's method here.

In his method, by restricting Θ initially to $[0,\pi/2]$, two envelopes are used for (4.4) depending on the value of Θ:

$$\begin{aligned} e(\theta,\phi) &\le \exp(k_m \sin^2\theta)\sin 2\theta \equiv d_1(\theta,\phi) && \text{for } \theta \in [0,\pi/3] \\ &\le \exp(k_m)\sin\theta \equiv d_2(\theta,\phi) && \text{for } \theta \in (\pi/3,\pi/2] \end{aligned}$$

where $k_m = \max(k_1,k_2)$. Variates from both envelopes $d_1(\theta,\phi)$ and $d_2(\theta,\phi)$ can be easily generated since $\Phi \sim U(0,2\pi)$ is independent of Θ which can be generated readily by inversion method. Let

$$\Delta_1 = \int_0^{2\pi}\int_0^{\pi/3} d_1(\theta,\phi)\, d\theta d\phi = 2\pi[\exp(3k_m/4)-1]/k_m,$$

$$\Delta_2 = \int_0^{2\pi}\int_{\pi/3}^{\pi/2} d_2(\theta,\phi)\, d\theta d\phi = \exp(k_m)\pi,$$

$$S_1 = \sup_{\theta,\phi} [e(\theta,\phi)/d_1(\theta,\phi)] = 1,$$

$$S_2 = \sup_{\theta,\phi} [e(\theta,\phi)/d_2(\theta,\phi)] = 1,$$

and

$$p = \Delta_1 S_1/(\Delta_1 S_1+\Delta_2 S_2) = \Delta_1/(\Delta_1+\Delta_2).$$

Then, with probability p, (Θ^*,Φ^*) is sampled from $d_1(\theta,\phi)$ and is accepted with probability $e(\Theta^*,\Phi^*)/d_1(\Theta^*,\Phi^*)$. Similarly, with probability 1 - p, (Θ^*,Φ^*) is sampled from $d_2(\theta,\phi)$ and is accepted with probability $e(\Theta^*,\Phi^*)/d_2(\Theta^*,\Phi^*)$. The accepted (Θ^*,Φ^*) will have Bingham distribution having p.d.f. (4.4). Denote this algorithm as BHM. The sampling efficiency of BHM is

$$R_{BHM} = \int_0^{2\pi}\int_0^{\pi/2} \exp[(k_1\cos^2\phi+k_2\sin^2\phi)\sin^2\theta]\sin\theta\, d\theta d\phi \;/\; (\Delta_1+\Delta_2).$$

We have shown in the discussion of Dimroth-Watson distribution that an envelope proportional to $(1-\rho x^2)^{-3/2}$ is very efficient for the marginal distribution. By noting the similarity between Dimroth-Watson distribution and Bingham distribution, in fact the former is a special case of the latter, we may apply a similar envelope to aid sampling from the Bingham distribution. We discuss the new method here.

Let $k(\phi) = k_1\cos^2\phi + k_2\sin^2\phi$, $k_m = \max(k_1,k_2)$ and consider $\theta \in [0,\frac{\pi}{2}]$. Then

$$e(\theta,\phi) = \exp[k(\phi)\sin^2\theta]\ \sin\theta$$
$$= \exp[k(\phi)]\ \exp[-k(\phi)\cos^2\theta]\ \sin\theta. \tag{4.6}$$

With a change of variable $X = \cos\Theta$, (4.6) becomes

$$\exp[k(\phi)]\exp[-k(\phi)x^2] \le \exp(k_m)\exp(-k_m x^2), \qquad x \in [0,1]. \tag{4.7}$$

Note that $\exp(-k_m x^2)$ is proportional to the marginal Dimroth-Watson p.d.f. This suggests $C(1-\rho x^2)^{-3/2}$ as an envelope for $X = \cos\Theta$ where C is some constant. By using the result in Appendix 2, we have

$$\exp(-k_m x^2) \ \le\ \beta(1-\rho x^2)^{-3/2} \tag{4.8}$$

where $\beta = \alpha^{3/2}\exp(-\ \frac{3}{2}(1\ -\ \frac{1}{\alpha}))$, $\alpha = -3\rho/2k_m$, $\rho = [\lambda\ -\ (\lambda^2\ +\ 16k_m)^{1/2}]/4$ and $\lambda = -2k + 3$. Hence, by (4.7) and (4.8), we have

$$\exp[k(\phi)(1-x^2)] \le \exp(k_m)\ \beta(1-\rho x^2)^{-3/2} \tag{4.9}$$

Since it has been shown that the envelope determined by (4.8) is a good envelope for the marginal Dimroth-Watson p.d.f (the sampling efficiency in the worst case is 0.5 when $k \to \infty$), the success of our method depends almost solely on the inequality (4.7). Thus, when k_1 and k_2 are close in values, our envelope due to (4.9) will be good. Otherwise, when the two parameters differ much apart, the sampling efficiency

$$R_{BGM} = (1-\rho)^{1/2}\int_0^{2\pi}\int_0^{\pi/2}\exp[(k_1\cos^2\phi+k_2\sin^2\phi)\sin^2\theta]\sin\theta\ d\theta d\phi\ /\ 2\pi\beta\exp(k_m)$$

will be too small. Denoted by BGM, our algorithm based on (4.9) is

Algorithm BGM

S. Set $k_m = \max(k_1,k_2)$,

$\lambda = -2k_m+3,\qquad \rho = [\lambda\ -\ (\lambda^2+\ 16k_m)^{1/2}]/4,$

$\alpha = -3\rho/2k_m,\qquad q = \alpha^3\exp(-3\ +\ 3/\alpha).$

1. Generate U_1, U_2, U_3 independent U(0,1).

2. $\Phi = 2\pi U_1$.

3. $k_p = k_1 \cos^2\phi + k_2 \sin^2\phi$,

4. $S = U_2^2 / [1-\rho(1-U_2^2)]$.

5. $W = 1 + k_p(1-S) - k_m$.

6. If $W > 0$ and $U_3^2 \le (1-\rho S)^3 W^2/q$, then go to step 8.

7. If $U_3^2 > (1-\rho S)^3 \exp(2(W-1))/q$, then go to step 1.

8. $\Theta = \cos^{-1}\sqrt{S}$.

9. Generate $U_4 \sim U(0,1)$.

10. If $U_4 < 0.5$, then $\Theta = \pi - \Theta$.

To compare our algorithm BGM with Johnson's BHM. Let's look at the ratio of their sampling efficiencies, R_{BHM}/R_{BGM} first:

k_m	-80	-10	-2	-0.5	0	0.5	2	10	100
Ratio	0.95	0.91	0.8	0.79	0.8	0.82	0.86	0.66	0.22

The ratio is smaller than one for all k, implying that BGM may be more efficient than BHM. Furthermore, the mean requirement of random numbers for BGM is $1 + 3/R_{BGM}$ which is always smaller than that, $1 + 4/R_{BHM}$, for BHM. This gives more evidence that the proposed algorithm is better than Johnson's algorithm.

An empirical comparison is conducted to compare the marginal generation time for BH, BHM and BGM by generating a sample of size 10000 using the three algorithms respectively. For algorithm BH, the mixture variable is generated by using the same envelope as the one we use in DWAG for Dimroth-Watson distribution and variates from the von Mises distribution are generated by algorithm VMBF. By experience, it is advisable to approximate the von Mises distribution by the uniform distribution when the concentration parameter is too small, say < 0.001, to avoid tremendous rounding error in VMBF. The result is shown in table 4.4.

Table 4.4 Marginal generation time (μs) for Bingham distribution

	(k_1,k_2)				
	(.5,.1)	(2,.1)	(6,5)	(20,5)	(-.1,-.5)
BH	649	708	670	1948	633
BHM	293	429	411	2709	299
BGM	205	316	288	1146	203

Table 4.4 (cont'd)

	(k_1,k_2)				
	(-.1,-2)	(-5,-6)	(-5,-20)	(1,-1)	(10,-10)
BH	684	703	873	695	2113
BHM	414	406	734	427	2196
BGM	273	294	492	300	1267

For all the selected values of (k_1,k_2), BGM is uniformly fastest and this well conforms to our expectation. BH is inferior to the other algorithms with the exceptional case that it is faster than BHM when $|k_1-k_2|$ is large. It is shown that the marginal generation time for all algorithms are sensitive to the absolute difference between k_1 and k_2. The larger the $|k_1-k_2|$, the longer the generation time. In fact, the generation time tends to infinity when $|k_1-k_2| \to \infty$. This follows because all the sampling efficiencies tend to zero in this case.

4.2.6 Bingham-Mardia Distribution

This distribution, introduced by Bingham and Mardia (1978), is usually used as a model for data concentrated towards a small circle on the sphere. So it belongs to a family known as the small-circle distributions. The standardized p.d.f. is

$$f(\theta,\phi;\ k,\ell) \propto \exp[k(\cos\theta-\ell)^2]\sin\theta, \tag{4.10}$$

where $\theta \in [0,\pi]$, $\phi \in [0,2\pi)$ and $\ell \in [0,1)$, with rotational symmetry about the direction $\theta = 0$. The small-circle girdle form corresponds to $k < 0$ with maximum concentration around the small circle specified by $\theta = \cos^{-1}\ell$. When $k > 0$, we have a 'squeezed belt' distribution with minimum concentration around the circle $\theta = \cos^{-1}\ell$.

The form of (4.10) is very similar to the Dimroth-Watson distribution. In fact, it reduces to the Dimroth-Watson density when $\ell = 0$. A generation procedure similar to DWAG is thus suggested for (4.10) here.

With a change of variable $Y = \cos\Theta - \ell$, the p.d.f. of Y is

$$f(y) = C_f\exp(ky^2), \qquad y \in [-1-\ell,1-\ell].$$

Split the interval $y \in [-1-\ell,1-\ell]$ into the two intervals $[-1-\ell,0)$ and $[0,1-\ell]$ and use envelopes proportional to the following p.d.f.s correspondingly:

$$\begin{cases} g_1(y) = C_{g1}(\rho_1)(1-\rho_1y^2)^{-3/2}, & y \in [-1-\ell,0), \\ g_2(y) = C_{g2}(\rho_2)(1-\rho_2y^2)^{-3/2}, & y \in [0,1-\ell], \end{cases}$$

where

$$\begin{cases} C_{g1}(\rho_1) = [1-\rho_1(1+\ell)^2]^{1/2}/(1+\ell). \\ C_{g2}(\rho_2) = [1-\rho_2(1-\ell)^2]^{1/2}/(1-\ell). \end{cases}$$

Define

$$\begin{cases} \rho_1 = [\lambda_1 - (\lambda_1^2 - 16k(1+\ell)^2)^{1/2}]/4(1+\ell)^2, & \lambda_1 = 3 + 2k(1+\ell)^2, \\ \rho_2 = [\lambda_2 - (\lambda_2^2 - 16k(1-\ell)^2)^{1/2}]/4(1-\ell)^2, & \lambda_2 = 3 + 2k(1-\ell)^2, \end{cases}$$

and

$$\beta(\rho_i) = \alpha_i^{3/2} \exp(-\tfrac{3}{2}(1 - \tfrac{1}{\alpha_i})), \qquad \alpha_i = 3\rho_i/2k,$$

where i = 1, 2. Following the spirit of the proof in Appendix 2, it can be shown that

$$\begin{cases} \exp(ky^2) \le \beta(\rho_1)(1-\rho_1 y^2)^{-3/2}, & y \in [-1-\ell, 0). \\ \exp(ky^2) \le \beta(\rho_2)(1-\rho_2 y^2)^{-3/2}, & y \in [0, 1-\ell]. \end{cases}$$

and that the choice of ρ_1, ρ_2 is optimal so as to maximize the sampling efficiency over each interval of y, that is, for i = 1 and 2,

$$\max_{\rho} \left(C_{gi}^{-1}(\rho)\ \beta(\rho) \right)^{-1} = \left(C_{gi}^{-1}(\rho_i)\ \beta(\rho_i) \right)^{-1}.$$

Let

$$\begin{cases} \Delta_1 = \int_{-1-\ell}^{0} (1-\rho_1 y^2)^{-3/2}\, dy = C_{g1}^{-1} \\ \Delta_2 = \int_{0}^{1-\ell} (1-\rho_2 y^2)^{-3/2}\, dy = C_{g2}^{-1} \end{cases}$$

and U_1, U_2, U_3 be independent U(0,1). Then, the steps in generating (Θ,Φ) from Bingham-Mardia distribution are:

Procedure BM

1. With probability $\beta(\rho_1)\Delta_1/[\beta(\rho_1)\Delta_1 + \beta(\rho_2)\Delta_2]$, generate Y from $g_1(y)$ and accept X = Y if $U_1 < [\beta(\rho_1)]^{-1}\exp(kY^2)\ (1-\rho_1 Y^2)^{3/2}$. Similarly, with probability $\beta(\rho_2)\Delta_2/[\beta(\rho_1)\Delta_1 + \beta(\rho_2)\Delta_2]$, generate Y from $g_2(y)$ and accept X = Y if $U_2 < [\beta(\rho_2)]^{-1}\exp(kY^2)\ (1-\rho_2 Y^2)^{3/2}$.

2. Return $\Theta = \cos^{-1}(X+\ell)$ and $\Phi = 2\pi U_3$.

It is simple to generate variate from $g_i(y)$, $i = 1, 2$:

(i) To generate Y from $g_1(y)$, $y \in [-1-\ell, 0)$:

Set $Y = -(1+\ell)U \,/\, [1-\rho_1(1+\ell)^2(1-U^2)]^{1/2}$, $U \sim U(0,1)$.

(ii) To generate Y from $g_2(y)$, $y \in [0, 1-\ell]$:

Set $Y = (1-\ell)U \,/\, [1-\rho_2(1-\ell)^2(1-U^2)]^{1/2}$, $U \sim U(0,1)$.

Special case: When $\ell = 0$, ρ_1 equals ρ_2 and we choose to generate variate from $g_1(y)$, $y \in [-1,0)$ or $g_2(y)$, $y \in [0,1]$ with equal probability. This is exactly the same as the case for Dimroth-Watson distribution using DWAG.

Note that the mean requirement of random numbers to generate one vector (Θ,Φ) using BM is $N_e = 1 + 3/R$ where

$$R = [\beta(\rho_1)\Delta_1 + \beta(\rho_2)\Delta_2]^{-1} \int_{-1-\ell}^{1-\ell} \exp(kx^2)\, dx$$

is the sampling efficiency. It is because, on the average, 1/R random numbers is used to select which envelope to generate, 1/R to generate from the chosen envelope, 1/R to check the acceptance of Y while the remaining one is to generate Φ. Table 4.5 below lists N_e and R for some chosen values of k and ℓ.

Table 4.5 The mean requirement of random numbers to generate one vector from Bingham-Mardia distribution using BH (Sampling efficiencies in parentheses).

	k							
ℓ	-20	-5	-1	-0.1	0.5	1	5	20
0.2	4.7 (.82)	4.4 (.88)	4.1 (.98)	4 (1)	4 (.99)	4.2 (.95)	5.9 (.62)	6.5 (.54)
0.5	4.6 (.84)	4.4 (.89)	4.1 (.96)	4 (1)	4.1 (.96)	4.5 (.87)	6.2 (.58)	6.6 (.54)
0.8	4.4 (.87)	4.5 (.87)	4.3 (.92)	4 (1)	4.3 (.92)	4.9 (.77)	6.4 (.56)	6.7 (.53)

The mean requirement of random numbers N_e ranges from the minimum 4 when $k = 0$, to below 7. The effect of ℓ is only slight on the value of N_e. Therefore, BM is considered an efficient procedure for Bingham-Mardia distribution. Note also that the sampling efficiency appears to converge to 0.5 when $k \to \infty$ regardless of the value of ℓ.

4.2.7 Other distributions

Wood (1987) develops envelope-rejection procedures for generating a sub-family of the general Fisher-Bingham distributions. This sub-family contains 6 parameters and is called FB_6 distribution. With an appropriate choice of coordinate axes, the FB_6 density can be written as

$$f(\mathbf{x};\ k,\beta,\gamma) = [2\pi C(k,\beta,\gamma)]^{-1}\exp[kx_3+\gamma x_3^2+\beta(x_1^2-x_2^2)],$$

where $k \geq 0$, $\beta \geq 0$ and $\gamma \in (-\infty,\infty)$, $\mathbf{x} = (x_1,x_2,x_3)^T$, for $x_1^2 + x_2^2 + x_3^2 = 1$.

Some special cases included are:

(1) uniform ($k=\beta=\gamma=0$); (2) Fisher ($\beta=\gamma=0$); (3) Dimroth-Watson ($k=\beta=0$); (4) Bingham, a special case ($k=0$) (4) FB_4 ($\beta=0$); (5) Kent distribution, (or FB_5) ($\gamma=0$).

By letting

$$X_3 = Z, \quad X_1 = (1-Z^2)^{1/2}\cos\Phi \quad \text{and} \quad X_2 = (1-Z^2)^{1/2}\sin\Phi, \tag{4.11}$$

Wood shows that $2\Phi \mid Z=z$ has von Mises distribution with mode 0 and concentration parameter $\beta(1-z^2)$ and the marginal p.d.f. of Z is

$$g(z;\ k,\beta,\gamma) = C(k,\beta,\gamma)^{-1}I_o[\beta(1-z^2)]\ \exp(kz^2+\gamma z^2), \qquad z \in [-1,1].$$

Since efficient generators for von Mises distribution already exist, the problem is to generate variate from $g(z;\ k,\beta,\gamma)$. Wood proposed four envelopes for g according to the values of k, β and γ (In each cases, the sampling efficiency is denoted by a_*):

(i) $\beta = 0,\ \gamma \le 0$ $\qquad g(z;\ k,0,\gamma) \le a_*^{-1}(k,\gamma)\ g(z;\ k,0,0)$

where $a_*(k,\gamma) = C(k,0,\gamma)\ /\ C(k,0,0)$.

(ii) $\beta = 0,\ \gamma \le 0$ $\qquad g(z;\ k,0,\gamma) \le a_*^{-1}(k,\gamma)\ g(z;\ k+2\gamma,0,0)$

with $a_*(k,\gamma) = e^{\gamma}C(k,0,\gamma)\ /\ C(k+2\gamma,0,0)$.

(iii) $\beta = 0,\ \gamma \ge 0$

$$g(z;\ k,0,\gamma) \le a_*^{-1}(k,\gamma)\ [pg(z;\ k+\gamma,0,0)+(1-p)g(z;\ k-\gamma,0,0)]$$

where $p = C(k+\gamma,0,0)\ /\ [C(k+\gamma,0,0) + C(k-\gamma,0,0)]$ and

$a_*(k,\gamma) = p(1+e^{-2\gamma})C(k,0,\gamma)\ /\ C(k+\gamma,0,0)$.

(iv) For all possible k, β and γ

$$g(z;\ k,\beta,\gamma) \le a_*^{-1}(k,\beta,\gamma)\ [pg(z;\ k,0,\gamma-\beta)+(1-p)g(z;\ k-\gamma,0,\gamma-\beta)]$$

where $p = C(k,0,\gamma-\beta)\ /\ [C(k,0,\gamma-\beta) + e^{-2\beta}C(k,0,\gamma+\beta)]$ and

$a_*(k,\beta,\gamma) = 2pe^{-\beta}C(k,\beta,\gamma)\ /\ C(k,0,\gamma-\beta)$.

The envelope for the marginal FB_4 density with negative quadratic term by (i) and (ii), which are based on the inequalities $\gamma z^2 \le 0$ and $\gamma z^2 \le \gamma(2z-1)$ respectively and valid for $\gamma \le 0$, are proportional to marginal Fisher densities.

In (iii), the marginal FB_4 density with positive quadratic term is surrounded by an envelope proportional to a mixture of two marginal Fisher densities based on the inequality $e^{\gamma z}+e^{-\gamma z} \ge e^{\gamma z^2}(1+e^{-2\gamma})$, $z \in [-1,1]$, $\gamma \ge 0$.

The general envelope by (iv) is proportional to a mixture of two marginal FB_4 densities based on the inequality $I_o(x) \le \cosh x$.

The generation procedure for FB_6 is thus

Procedure FB6

1. Generate Z from $g(z;\ k,\beta,\gamma)$ using one of the four envelopes suggested.
2. Generate $\delta \sim$ Bernoulli(0.5).

3. Generate $\psi \in (-\pi,\pi)$ from von Mises distribution with concentration parameter $\beta(1-Z^2)$.

4. If $\psi < 0$, then accept $\Phi = (1+\delta)\pi + \psi/2$; otherwise accept $\Phi = \delta\pi + \psi/2$.

5. Obtain $X = (X_1,X_2,X_3)^T$ from Z and Φ via (4.11).

In step 1, how to decide which envelopes to use depends on the values of k, β and γ. For FB_4 distributions ($\beta = 0$), Wood discusses situations in which a particular envelope is best for $g(z; k,0,\gamma)$. All candidate envelopes are proportional to marginal Fisher densities.

For the FB_5 distributions ($\gamma = 0$), the envelope chosen is proportional to the marginal Fisher density with parameter k. The procedure is adequate when $k \geq 2\beta$ where FB_5 is unimodal. The performance is less satisfactory for bimodal FB_5 when $k < 2\beta$.

In the general case, the FB_6, envelope by (iv) is used. The sampling efficiency depends on the (lower) bound $I_o(\beta)/\cosh \beta$ which, as shown by Wood, is a decreasing function of β. Thus, the procedure FB6 will be inefficient for large β.

Wood's procedure is also suitable for the Dimroth-Watson distribution ($k = \beta = 0$). The envelopes he uses are either uniform (see (i)) or proportional to the marginal Fisher densities (see (ii) and (iii)). In terms of sampling efficiency, however, these envelopes are inferior to those envelopes we discussed earlier.

CHAPTER FIVE

GENERATING VARIATES FROM NON-UNIFORM DISTRIBUTIONS ON THE N-SPHERE

§5.1 Introduction

In this chapter we will discuss simulation of variates from distributions on the higher dimensional (> 3) sphere. Though applications of directional data models have been much concentrated to the two and three dimensions, one important application of higher dimensional distributions is given by Stephens (1982) in the analysis of continuous proportions.

Important distributions on the n-sphere include the von Mises-Fisher distribution and the Bingham distribution (throughout this chapter, we will unambiguously use the name Bingham distribution to stand for the general n-dimensional Bingham distribution) which frequently appear in literature. Ulrich (1984) developed a procedure for constructing algorithms to generate variates from the Saw distribution. The method is applied to the von Mises-Fisher distribution, a special case of the Saw distribution, and is found to be very efficient. In the next section we will see how Ulrich's method work and investigate the effectiveness of his new method in a small simulation study. In addition, a procedure for the Bingham distribution is proposed.

§5.2 Methods for Higher Dimensional Spherical Distributions

Before discussing Ulrich's work, let us consider the n-dimensional Saw distribution first.

Let $\mathbf{X}$ be an $n \times 1$ random vector with unit length (that is $X^TX = 1$). Saw (1978) defined a p.d.f. of $\mathbf{X}$ as

$$f(\mathbf{x};\ k,\xi) = \frac{g(k\mathbf{x}^T\xi)}{\alpha_n C_k}, \tag{5.1}$$

where

$g(u) \geq 0$, $g'(u) > 0$, $g(u)$ is the kernel function;

$k \geq 0$ is the concentration parameter;

ξ is an $n \times 1$ vector called the modal vector with $\xi^T\xi = 1$;

$\alpha_n = 2\pi^{n/2} / \Gamma(n/2)$;

$C_k = \int_{-1}^{1} g(kt) \dfrac{(1-t^2)^{(n-3)/2}}{B(1/2,(n-1)/2)}\, dt$, is the normalizing constant and

$B(\alpha,\beta)$ is the beta function.

The density defined by (5.1) is known as the Saw distribution. The following theorem is the key to the problem of generating random vector $\mathbf{X}$ from (5.1).

<u>Theorem</u> Let W be a random variable with p.d.f.

$$\frac{g(kw)(1-w^2)^{(n-3)/2}}{B(1/2,(n-1)/2)}, \qquad w \in [-1,1], \qquad n \geq 2, \tag{5.2}$$

and let $\mathbf{V} \sim U_{n-1}$, the uniform distribution on the $(n-1)$-sphere, be independent of W. Then the vector $\mathbf{X}$, where

$$\mathbf{X} = ((1-W^2)^{1/2}V^T,\ W)^T,$$

has p.d.f. $f(\mathbf{x};\ k,\xi)$ as (5.1) with modal vector $\xi^T = (0,0,\ldots,0,1)$.

The theorem characterizes a property of the Saw distribution that it has constant density on all $(n-1)$-dimensional subspheres. Since there are many efficient methods to generate $\mathbf{V}$ from U_{n-1} (see chapter 2), the problem remains generating the mixture variable W from (5.2).

5.2.1 The von Mises-Fisher Distribution

When the kernel function $g(\mathbf{u}) = \exp(\mathbf{u})$, the p.d.f. defined by (5.1) becomes the von Mises-Fisher distribution, denoted as F_n. The p.d.f is

$$f(\mathbf{x};\ k,\xi) = \frac{\exp(k\mathbf{x}^T\xi)}{\alpha_n C_k}, \tag{5.3}$$

which further reduces to U_n when $k = 0$.

Some particular cases of F_n include: (i) the von Mises distribution ($n = 2$) and (ii) the Fisher distribution ($n = 3$), which have been discussed in chapter three and chapter four respectively.

Instead of using Cartesian coordinates $\mathbf{X}$, it is sometimes convenient to consider the distribution F_n in terms of spherical polar coordinates $\Theta = (\Theta_1,\ldots,\Theta_{n-1})^T$. Without loss of generality, let the modal vector ξ be $(0,\ldots,1)^T$. The p.d.f. of Θ is

$$f(\Theta;\ k) = C_n(k)\exp(k\cos\theta_1)\sin^{n-2}\theta_1 \ldots\ldots \sin\theta_{n-2}, \tag{5.4}$$

where $C_n(k)$ is the normalizing constant and $\theta_j \in [0,\pi]$, $j = 1, 2,\ldots, n-2$; $\theta_{n-1} \in [0,2\pi)$, $k > 0$.

To generate the mixture variable from (5.2) when the kernel function is the exponential function, Ulrich suggested to use envelope-rejection method with an envelope proportional to the p.d.f.

$$e(x;\ b) = \frac{2b^{(n-1)/2}}{B((n-1)/2,(n-1)/2)}\ \frac{(1-x^2)^{(n-3)/2}}{[(1+b)-(1-b)x]^{n-1}}, \qquad x \in [-1,1], \tag{5.5}$$

where

$$b = \frac{-2k + [(n-1)^2 + 4k^2]^{1/2}}{n-1}.$$

It is easy to generate variate from $e(x;\ b)$ since $X \sim e(x;\ b)$ if

$$X = [1 - (1+b)Z]\ /\ [1 - (1-b)Z],$$

where $Z \sim \mathrm{Beta}((n-1)/2,(n-1)/2)$.

In order to generate $X \sim F_n$, we have the following algorithm:[1]

Algorithm VMFU

S. Set $b = \{-2k + [(n-1)^2 + 4k^2]^{1/2}\} / (n-1)$,

$$d = (n-1)\,[1 - \ln(\tfrac{n-1}{2})].$$

1. Generate $U \sim U(0,1)$ and $Z \sim \mathrm{Beta}((n-1)/2,(n-1)/2)$.

2. $T = (n-1)(1+b) / 2[1-(1-b)Z]$.

3. If $(n-1)\ln T - T + d < \ln U$, then go to step 1.

4. Generate $V \sim U_{n-1}$.

5. $W = [1 - (1+b)Z] / [1 - (1-b)Z]$.

6. $\mathbf{X} = ((1-W^2)^{1/2}V^T,\ W)^T$.

Ulrich used a preliminary step

(2') If $n + \ln(\phi^{(n-1)}\theta) + d - (n-1)\phi/T - T \geq \theta U$, then go to step 4.

before step 3 to increase execution speed. He suggested to use $\theta = 1.25$ and $\phi = n-1$. An algorithm based on envelope-rejection technique is suggested to generate Z from the symmetric beta distribution in step 1 when $n \geq 3$:

Algorithm SBETA To generate $Z \sim \mathrm{Beta}(\alpha,\alpha)$.

1. Generate U_1, U_2 independent $U(0,1)$.

2. $S = (2U_1-1)^2 + U_2^2$.

3. If $S > 1$, then go to step 1.

4. $Z = 0.5 + U_1U_2S^{-1}[1 - S^{1/(\alpha-0.5)}]^{1/2}$.

When $n = 2$ (that is, the von Mises distribution), the symmetric beta distribution can be generated efficiently (the sampling efficiency is $\pi/4$)

---

[1]Some corrections and simplification have been made to Ulrich's (1984) original algorithm due to some mistakes found in his paper.

by Jöhnk's (1964) envelope-rejection method:

Algorithm SHBETA To generate $Z \sim Beta(1/2,1/2)$.

1. Generate U_1, U_2 independent U(0,1).
2. $S = U_1^2 + U_2^2$.
3. If $S > 1$, then go to step 1.
4. $Z = U_1^{\,2}/S$.

To access the performance of algorithm VMFU (with pre-test 2' and step 4 by Muller's (1959) method), we compare the timings for generating 10000 samples by VMFU with that of another algorithm known as VMFPS. A normal generator RNNOF from IMSL is employed to generate standard normal variates for Muller's method. Results are listed in table 5.1. Now let us see what VMFPS is and how it works.

VMFPS initially generates a random vector Θ from (5.4) and then obtains **X** by polar transformation. By (5.4), it is obvious that $\Theta_1,\ldots,\Theta_{n-1}$ are all independent with $\Theta_{n-1} \sim U(0,2\pi)$ and Θ_j, $j = 2, 3,\ldots, n-2$ having power sine densities (since the marginal p.d.f. of Θ_j is proportional to $\sin^{n-j-1}\theta_j$ for $j = 2, 3,\ldots, n-2$) and $f(\theta_1) \propto \exp(k\cos\theta_1)\sin^{n-2}\theta_1$. Generation from the power sine densities can be achieved, for example by envelope-rejection method suggested by Johnson (1987) (except for θ_{n-2} which can be generated directly by inversion method). His method can be summarized as follows ($j = 2, 3,\ldots, n-3$):

Algorithm PWS

1. Generate U_1, U_2 independent U(0,1).
2. Set $\Theta_j = \pi U_1$.
3. If $\sin^{n-j-1}\theta_j < U_2 < 1 - |\cos^{n-j-1}\theta_j|$, then go to step 1.
4. If $U_2 \le \sin^{n-j-1}\theta_j$, then accept θ_j.
5. Otherwise, $U_2 \ge 1 - |\cos^{n-j-1}\theta_j|$. If $\Theta_j \le \pi/2$, accept $\Theta_j = \Theta_j + \pi/2$.

If $\Theta_j > \pi/2$, accept $\Theta_j = \Theta_j - \pi/2$.

The generation of Θ_1 can be achieved by noting that

$$\exp(k\cos\theta_1)\ \sin^{n-2}\theta_1 \ \le\ \exp(k\cos\theta_1)\ \sin\theta_1, \qquad \theta_1 \in [0,\pi]. \tag{5.6}$$

That is, the marginal p.d.f. of Θ_1 is surrounded by an envelope which is proportional to the marginal Fisher density. Algorithm FSH or FSHR discussed in chapter four can be used for the marginal Fisher distribution. Here, we use FSH. The complete algorithm of VMFPS is stated as follows:

Algorithm VMFPS

S. Set $B = e^{2k} - 1$.

1. Generate U_1, U_2 independent U(0,1).

2. Set $\Theta_1 = \cos^{-1}[-1 + k^{-1}\ln(BU_1 + 1)]$.

   Accept Θ_1 if $U_2 \le \sin^{n-3}\Theta_1$; otherwise go to step 1.

3. Generate $\Theta_2, \Theta_3, \ldots, \Theta_{n-3}$ using PWS.

4. Generate U_3, U_4 independent U(0,1).

5. Set $\Theta_{n-2} = \cos^{-1}(1-2U_3)$.

6. Set $\Theta_{n-1} = 2\pi U_4$.

7. Obtain $\mathbf{X} = (X_1,\ldots,X_n)$ by $\Theta = (\Theta_1,\ldots,\Theta_{n-1})$ through polar transformation.

Table 5.1 Timings (s) for generating a sample of size 10000 from von Mises-Fisher distribution using VMFU (timings for VMFPS are in blankets).

	n			
k	4	5	7	10
0.5	5.72(3.25)	6.67(4.75)	8.72(8.18)	11.7(14.2)
1	5.72(3.28)	6.71(4.85)	8.76(8.29)	11.7(14.5)
3	5.75(3.61)	6.67(5.55)	8.73(9.90)	11.7(17.3)
20	5.95(5.85)	6.80(14.9)	8.76(72.2)	11.7 (477)

As pointed out by Ulrich that the execution times of the algorithm VMFU will not depend appreciably upon the value of k. From table 5.1 we see that the execution times of VMFU are almost the same for all selected values of k at each chosen dimension n. It is observed that the execution time increases linearly with the increase in the dimension n. In fact, based on the above data, the following relation can be formulated (assume k has no effect on execution time and treat the execution times at each dimension as repeated observations):

$$\hat{T} = 1.78 + 0.99\, n, \qquad \text{with } R^2 \cong 0.999, \qquad (5.7)$$

where $\hat{T}$ is the estimated execution time. Thus, empirically, VMFU is an $O(n)$ routine. The linear relation (5.7) demonstrates that the envelope which is proportional to the transformed symmetric beta density (5.5) is a very efficient envelope for the mixture density defined by (5.2) when the kernel function is the exponential function.

In comparison with VMFU, algorithm VMFPS is only faster when n and k are small (e.g. $k \le 3$ when $n = 5$). Its execution time depends heavily on the parameter k: if k increases, then the execution time increases dramatically. From the empirical observations on table 5.1, it can be shown that VMFPS is of order n^2 when k is small ($k \le 3$). The situation gets much worse when k is large. The problem is mainly due to the poor envelope used in generating Θ_1. If we examine step 4 of VMFPS, we can find that when k is large, there is only a small probability that Θ_1 be accepted.

Sometimes we are more interested in the generation of the polar angles Θ than the Cartesian coordinates **X**. One advantage of VMFPS is that it can generate Θ directly. Since VMFU in its creation is designed for generating **X** directly, inverse polar transformation is necessary to obtain Θ. In such case, it would be time-consuming to evaluate the numerous arc-sine and arc-cosine functions. Thus only in the situation when both n and k are small and polar angles are much desirable is VMFPS a preferable choice. In general, VMFU is the fastest algorithm for von Mises-Fisher distribution at all situations.

5.2.2 Bingham Distribution

This is a generalization of the 3-dimensional Bingham distribution for axial directional data in the n-dimensional space. A random vector X = $(X_1,\dots,X_n)$, with $\sum_{i=1}^{n} X_i^2 = 1$, is said to have Bingham distribution if its p.d.f.

$$f(\mathbf{x};\ \mu,K) \propto \operatorname{etr}(K\mu^T\mathbf{x}\mathbf{x}^T\mu)$$

where $\operatorname{etr}(.) = \exp(\operatorname{tr}(.))$, $\mu = (\mu_1,\dots,\mu_n)^T$ is an orthogonal matrix and $K = \operatorname{diag}(k_1,\dots,k_n)$ is a diagonal matrix of parameters. Without loss of generality, we assume $\mu = \mathbf{I}$, the identity matrix. The p.d.f. becomes

$$f(\mathbf{x};\ \mathbf{I},K) \propto \exp\left(\sum_{i=1}^{n} k_i x_i^2\right).$$

Since the parameters k_i are unique up to an additive constant, we may assume $k_n = 0$ for uniqueness. In terms of polar coordinates $\Theta = (\Theta_1,\dots,\Theta_{n-1})^T$, the p.d.f. of Θ is proportional to

$$b(\theta;\ \mathbf{K}) = \exp(k_o(\theta)) \prod_{i=2}^{n-1} \sin^{i-1}\theta_{n-i}\ , \tag{5.8}$$

where $k_o(\theta) = \sum_{i=1}^{n-1} k_i (\cos\theta_{n-i+1} \prod_{j=0}^{n-i} \sin\theta_j)^2$ and $\sin\theta_o = \cos\theta_n = 1$, $\theta_j \in [0,\pi]$, $j = 1, 2,\dots, n-2$, $\theta_{n-1} \in [0,2\pi)$.

An envelope-rejection procedure is proposed to generate Θ from (5.8) here. To start, we seek an envelope such that all components of Θ are independent. This has the advantage that the proposed envelope will be easily generated. An envelope is derived below:

Let $k_m = \max(k_1,\ldots,k_{n-1})$. Then

$$b(\theta;\ K) \le \exp(k_m \sin^2\theta_1) \prod_{i=2}^{n-1} \sin^{i-1}\theta_{n-i}$$

$$= \exp(k_m \sin^2\theta_1)\sin^{n-2}\theta_1\ \sin^{n-3}\theta_2\ \ldots.\ \sin\theta_{n-2}$$

$$\equiv e_1(\theta_1;\ k_m)\ e_2(\theta_2)\ \ldots.\ e_{n-2}(\theta_{n-2})$$

$$\equiv e(\theta;\ k_m). \tag{5.9}$$

where

$$e_1(\theta_1;\ k_m) \equiv \exp(k_m \sin^2\theta_1)\sin^{n-2}\theta_1,$$

$$e_j(\theta_j) \equiv \sin^{n-j-1}\theta_j, \qquad \text{for } j = 2,\ 3,\ldots,\ n-2.$$

The inequality (5.9) defines an envelope $e(\theta;\ k_m)$ for $b(\theta;\ K)$ and it is obvious that this envelope has independent components and hence each of these components can be generated separately.

Based on the envelope $e(\theta;\ k_m)$ we can derive an envelope-rejection procedure for the Bingham distribution. Suppose $\Theta^* = (\Theta_1^*,\ldots,\Theta_{n-1}^*)^T$ is a random vector of polar coordinates distributed as $e(\theta;\ k_m)$. Then under the condition that

$$U \le \frac{b(\Theta^*;\ \mathbf{k})}{e(\Theta^*;\ k_m)}$$

$$= \exp[k_o(\Theta^*) - k_m \sin^2\Theta_1], \tag{5.10}$$

where $U \sim U(0,1)$ is independent of Θ^*, the vector Θ^* will have a distribution as $b(\theta;\ K)$.

Our problem is how to generate variates from $e(\theta;\, k_m)$.

Note that the marginal distributions of Θ_j, $j = 2, 3, \ldots, n-2$ are power sine distributions and these variates can thus be generated by the same methods we have discussed in the generation of the von Mises-Fisher distribution (algorithm VMFPS). Namely, $e_{n-2}(\theta_{n-2})$ is generated by inversion method and $e_j(\theta_j)$, $j = 2, 3, \ldots, n-3$ by Johnson's envelope-rejection method (algorithm PWS).

To generate Θ_1 from $e_1(\theta_1;\, k_m)$, the result in Appendix 2 is helpful to show that

$$\begin{aligned} e_1(\theta_1;\, k_m) &\equiv \exp(k_m \sin^2\theta_1)\sin^{n-2}\theta_1 \\ &\le \exp(k_m \sin^2\theta_1)\sin\theta_1 \\ &\le \exp(k_m)\, \beta(1 - \rho\cos^2\theta_1)^{-3/2}\sin\theta_1, \end{aligned} \tag{5.11}$$

where

$$\beta = \alpha^{3/2}\exp(-\tfrac{3}{2}(1 - \tfrac{1}{\alpha})), \qquad \alpha = -3\rho/2k_m,$$

$$\rho = [\lambda - (\lambda^2 + 16k_m)^{1/2}]/4, \qquad \lambda = -2k_m + 3.$$

Let $X = \cos\Theta_1$. By (5.11), we obtain

$$\exp[k_m(1-X^2)]\ (1-X^2)^{(n-3)/2} \ \le\ \exp(k_m)\ \beta(1 - \rho X^2)^{-3/2}$$

or

$$\exp[-k_m X^2)]\ (1-X^2)^{(n-3)/2} \ \le\ \beta(1 - \rho X^2)^{-3/2}.$$

Generating variate from $\beta(1 - \rho X^2)^{-3/2}$ is easily achieved by the method of inversion. Therefore, if X^* , generated from $\beta(1 - \rho x^2)^{-3/2}$, satisfies the condition

$$U \ \le\ \frac{1}{\beta}\exp[-k_m X^{*2})]\ (1-X^{*2})^{(n-3)/2}\ (1 - \rho X^{*2})^{3/2}, \tag{5.12}$$

where $U \sim U(0,1)$ is independent of X^*, it will be distributed as $e_1(\theta_1;\, k_m)$.

Algorithm NBG below shows clearly the steps in generating Bingham variates using envelope $e(\theta;\ k_m)$.

Algorithm NBG

S. Set $k_m = \max(k_1,\ldots,k_{n-1})$,

$$\lambda = -2k_m+3, \qquad \rho = [\lambda - (\lambda^2+16k_m)^{1/2}]/4,$$

$$\alpha = -3\rho/2k_m, \qquad q = \alpha^3\exp(-3 + 3/\alpha),$$

1. Generate U_1, U_2 independent U(0,1).

2. $S = U_1^2 \,/\, [1-\rho(1-U_1^2)]$.

3. $W = 1 - k_m S$.

4. If $W > 0$ and $U_2^2 \le (1-\rho S)^3(1-S)^{n-3}\, W^2/q$, then go to step 6.

5. If $U_2^2 > (1-\rho S)^3(1-S)^{n-3}\exp[2(W-1)]/q$, then go to step 1.

6. Set $\Theta_1 = \pm\cos^{-1}\sqrt{S}$, where $\pm$ is a random sign.

7. Generate $\Theta_2, \Theta_3,\ldots, \Theta_{n-3}$ using algorithm PWS.

8. Generate U_3, U_4, U_5 independent U(0,1).

9. Set $\Theta_{n-2} = \cos^{-1}(1-2U_3)$.

10. Set $\Theta_{n-1} = 2\pi U_4$.

11. Compute $k_o = \displaystyle\sum_{i=1}^{n-1} k_i\,(\cos\Theta_{n-i+1}\prod_{j=0}^{n-i}\sin\Theta_j)^2$. $\qquad (\sin\Theta_o = \cos\Theta_n = 1)$

12. $T = 1 + k_o - k_m\sin^2\Theta_1$.

13. If $U_5 \le T$, then go to step 15.

14. If $U_5 > \exp(T-1)$, then go to step 1.

15. Accept $\Theta = (\Theta_1,\ldots,\Theta_{n-1})^T$.

The algorithm is self-explainable. Step 2 generates the envelope for $e_1(\theta_1;\ k_m)$. Step 5 is based on the acceptance test (5.12) while step 14 perform the test (5.10).

To access the performance of the proposed algorithm NBG, we programmed the algorithm in Fortran and ran it on an IBM4381 computer. Based on a selection of the dimensionality n and the parameters $k_1, \ldots, k_{n-1}$, a sample of size 1000 was generated and the time requirements (in seconds) are listed as follows:

Table 5.2 Timings (s) for generating a sample of size 10000 from the Bingham distribution using NBG

n		
4	6	10
(-10,-9,-6) **1.16**	(-10,-9,-9,-8,-6) **4.88**	(-10,-9,-9,-8.5,-8,-8,-7,-7,-6) **23.9**
(1,2,5) **1.91**	(1,1,2,3,5) **5.27**	(1,1,1.5,2,2,2.5,3,3,5) **14.2**
(1,2,8) **4.17**	(1,1,2,3,8) **21.2**	(1,1,1.5,2,2,2.5,3,3,8) **133**
(11,12,15) **2.09**	(11,12,13,15) **5.68**	(11,11,11.5,12,12,12.5,13,13,15) **14.8**

Note: (1) Timings are highlighted;
(2) Blanketed values represent parameters k_1, k_2,... and so on.

To summarize, the efficiency of the procedure NBG depends on these factors:

(1) Dimensionality n: The larger the value n, the lower the efficiency.

It is because most of the components of the envelope determined by $e(\theta;\ k_m)$ are generated by various envelope-rejection methods whose sampling efficiencies decline as n increases.

(2) Relative differences between the k's: The larger the differences, the lower the efficiency.

As the values of the parameters differ much apart, the envelope $e(\theta;\ k_m)$ will not tightly cover the target function $b(\theta;\ \mathbf{K})$. Therefore the probability of accepting the vector Θ generated from $e(\theta;\ k_m)$ by (5.10) decreases.

The above two factors are deterministic to the efficiency of NBG. The effect of other factors such as the magnitude of k's, on the other hand, seems to be less dramatic.

CHAPTER SIX

SUMMARY AND DISCUSSION

The generation of random points on the n-sphere has been discussed by many people and some efficient generators have been developed. Of all the efficient generators we have discussed in chapter two, Muller's (1959) method is perhaps the most important. Theoretically, his method is simple and elegant. As an O(n) routine, its applicability covers almost all dimensions and simulation result (see table 2.1) ensures that the marginal generation time increases only linearly with the dimensionality. On the other hand, the other procedures developed by Sibuya (1962) and Tashiro (1976) are essentially based on the theory of Muller and hence they are also O(n) routines and significant improvement in speed should not be expected.

Methods proposed for the generation of distributions on the circle are basically very efficient and all enjoys simple algorithms. For the lattice distributions and wrapped Poisson distribution which are both discrete, variate generation are nothing new compares to the univariate cases. Other distributions like the wrapped normal, wrapped Cauchy and angular Gaussian distributions can be obtained readily by transforming the appropriate univariate/bivariate random variables. Simple algorithms based on envelope-rejection technique are proved to be useful for the Cardioid, triangular as well as the von Mises distributions. For the von Mises distribution, it is worthwhile to mention that the envelope used by Best and Fisher (1979) which is based on the wrapped Cauchy distribution is essentially the same as that used by Ulrich (1984) which is proportional to the transformed symmetric beta distribution (5.5) in the sense that one envelope can be converted to the other through a transformation of its parameter. Another envelope which uses the simple inequality $\cos x \le 1 - x^2/\sqrt{\pi}$, $x \in (-\pi,\pi]$ is also proved to be as efficient as that by Best and Fisher (1979) (or Ulrich (1984)). Dagpunar (1983, 1990) introduced a still faster method when k is fixed between calls, however, price has to be paid. The cost for enhancement in speed is higher program complexity as numerical integration becomes necessary in calculating some constants in the set-up step.

Variates generation in the 3-dimensional case is usually a tougher job than in the two dimensions. With the exception of the Fisher and Arnold distributions which can be generated simply by inversion method, most spherical distributions cannot be generated this way because they do not have closed forms and there are also no simple transformations of other univariate distributions which are easily generated as in the circular case. Envelope-rejection methods seems to work well in this situation. In this paper we have proposed a very useful envelope which is of form $C(1-\rho x^2)^{-3/2}$, where $x \in [-1,1]$ and C is some constant. According to its form, the envelope is expected to work for spherical distributions which are axial in nature. In fact, it is applied to spherical distributions like the Dimroth-Watson, Bingham (also in higher dimensions) and Bingham-Mardia distributions (for this small-circle distribution, slightly modified envelopes are used) and the results are quite satisfactory. For the Dimroth-Watson and Bingham-Mardia distributions, the envelopes work very well for all parameters of interest. While for the Bingham distribution, the envelope is efficient when the two parameters are rather close in values.

On the higher dimensional sphere (n > 3), variates generation is even more difficult. The polar method suggested for the Bingham distribution is only effective when the values of $k_1, \ldots, k_{n-1}$ are all close to one another while the other polar method for the von Mises-Fisher distribution is useful only when k is small. Ulrich's (1984) algorithm has been shown to be very efficient for the latter case. It is empirically an O(n) routine and thus should be applicable virtually to all dimensions.

It is quite common that the usual variate generation technique employed when the dimension is greater than two is the envelope-rejection method. One obvious drawback of envelope-rejection method is that an envelope which is itself easily generated is usually too 'coarse' to cover the target p.d.f. tightly especially when the p.d.f. contains many parameters and the dimension is high. Thus efficient envelopes may only be found suitable for certain parameters values and in low dimensions. This greatly limits the strength of the generators developed by this method. Unfortunately, there seems to be no better way out.

It can be shown that the envelope $C(1-\rho x^2)^{-3/2}$ is proportional to a particular member of the 3-dimensional angular Gaussian distributions. Since the family of n-dimensional angular Gaussians can be easily generated (by normalizing a multivariate normal vector by its root sum of squares) and with suitable choice of parameters it covers many n-dimensional spherical distributions such as the unimodal distributions, symmetric girdle and bipolar distributions, there is a conjecture that this family of distributions may be useful to construct envelopes for spherical distributions of interest on the n-sphere. Such extension requires some work.

In this paper, variates generating algorithms of quite a lot of useful n-dimensional spherical distributions ($n \geq 2$) have been discussed. However, there are still some other distributions missed. Computer simulation of random variates from such distributions are yet under exploration.

References

Arnold, K.J. (1941) On spherical probability distributions. Ph.D. Thesis, Massachusetts Institute of Technology.

Atkinson, A.C. (1979a) The computer generation of Poisson random variables. *Appl. Statist.* **28**, 29-35.

Atkinson, A.C. (1979b) Recent developments in the computer generation of Poisson random variables. *Appl. Statist.* **28**, 260-263.

Atkinson, A.C. (1982) The simulation of generalized inverse Gaussian and hyperbolic random variables. *SIAM J. Scient. Statist. Comput.* 3, 502-517.

Batschelet, E. (1981) *Circular statistics in biology.* Academic Press, London.

Best, D.J. and Fisher, N.I. (1979) Efficient simulation of the von Mises distribution. *Appl. Statist.* **28**, 153-157.

Best, D.J. and Fisher, N.I. (1986) Goodness-of-fit and discordancy tests for samples from the Watson distribution on the sphere. *Austral. J. Statist.* **28**, 13-31.

Bingham, C. (1964) Distributions on the sphere and on the projective plane. Ph.D. Thesis, Yale University.

Bingham, C. (1974) An antipodally symmetric distribution on the sphere. *Ann. Statist.* **2**, 1201-1225.

Bingham, C. and Mardia, K.V. (1978) A small circle distribution on the sphere. *Biometrika* **65**, 379-389.

Box, G.E.P. and Muller, M.E. (1958) A note on the generation of random normal deviates. *Ann. Math. Statist.* 29, 610-611.

Cook, J.M. (1957) Rational formula for the production of a spherically symmetric probability distribution. *Math. Comput.* 11, 81-82.

Dagpunar, J.S. (1983) Methods for generating variates from probability distributions. Ph.D. Thesis, Brunel University.

Dagpunar, J.S. (1990) Sampling from the von Mises distribution via a comparison of random numbers. *J. Appl. Statist.* 17, 165-168.

Fisher, R.A. (1953) Dispersion on a sphere. *Proc. Roy. Soc. Lond.* A217, 295-305.

Fisher, N.I., Lewis, T. and Embleton, B.J.J. (1987) *Statistical analysis of spherical data.* Cambridge University Press, London.

Fisher, N.I., Lewis, T. and Willcox, M.E. (1981) Tests of discordancy for samples from Fisher's distribution on the sphere. *Appl. Statist.* 30, 230-237.

Forsythe, G.E. (1972) Von Neumann's comparison method for random sampling from the normal and other distributions. *Math. Comput.* 26, 817-826.

Gerontidis, I. and Smith, R.L. (1982) Monte Carlo generation of order statistics from general distributions. *Appl. Statist.* 31, 238-243.

IMSL Sfun/Library (1987) User's Manual: Fortran subroutines for evaluating special functions. Version 2.0

IMSL Stat/Library (1987) User's Manual: Fortran subroutines for statistical analysis. Version 1.0

Jöhnk, M.D. (1964) Erzeugung von Betaverteilten und Gammaverteilten Zufallszahlen. *Metrika* **8**, 5-15.

Johnson, M.E. (1987) *Multivariate Statistical Simulation.* John Wiley, New York.

Kendall, D.G. (1974) Pole-seeking Brownian motion and bird navigation. *J. R. Statist. Soc.* **B36**, 365-417.

Kinderman, A.J. and Ramage, J.G. (1976) Computer generation of normal random variables. *J. Am. Statist. Assoc.* **71**, 893-896.

Kronmal, R.A. and Peterson, A.V. (1979) On the alias method for generating random variables from a discrete distribution. *Am. Statist.* 33, 214-218.

Mardia, K.V. (1972) *Statistics of directional data.* Academic Press, London.

Mardia, K.V. (1975) Distribution theory for the von Mises-Fisher distribution and its application. *Statistical Distributions in Scientific Work.* **1**, 113-130. Reidel, Dordrecht (Holland).

Marsaglia, G. (1972) Choosing a point from the surface of a sphere. *Ann. Math. Statist.* **43**, 645-646.

Motoo, M. (1959) Some evaluations for continuous Monte Carlo method by using Brownian hitting process. *Ann. Inst. Statist. Math.* **11**, 49-54.

Muller, M.E. (1956) Some continuous Monte Carlo methods for the Dirichlet problem. *Ann. Math. Statist.* **27**, 569-589.

Muller, M.E. (1959) A note on a method for generating points uniformly on N-dimensional spheres. *Commun. Assoc. Comput. Mach.* **2**, 19-20.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1986) *Numerical recipes - the art of scientific computing*. Academic Press, London.

Saw, J.G. (1978) A family of distributions on the m-sphere and some hypothesis tests. *Biometrika* **65**, 69-73.

Seigerstetter, J. (1974) Discussion of Kendall, D.G. (1974). Pole-seeking Brownian motion and bird navigation. *J. R. Statist. Soc.* B36, 411-412.

Selby, B. (1964) Girdle distributions on the sphere. *Biometrika* **51**, 381-392.

Sibuya, M. (1962) A method for generating uniformly distributed points on N-dimensional spheres. *Ann. Inst. Statist. Math.* **14**, 81-85.

Singleton, R.C. (1969) Algorithm 347: An efficient algorithm for sorting with minimal storage. *Commun. ACM.* **12**, 185-187.

Stephens, M.A. (1964) The testing of unit vectors for randomness. *J. Am. Statist. Assoc.* **59**, 160-167.

Stephens, M.A. (1982) Use of the von Mises distribution to analyse continuous proportions. *Biometrika* **69**, 197-203.

Tashiro, Y. (1976) On methods for generating uniform random points on the surface of a sphere. *Ann. Inst. Statist. Math.* **29**, 295-300.

Ulrich, G. (1984) Computer generation of distributions on the m-sphere. *Appl. Statist.* **33**, 158-163.

Walker, A.J. (1977) An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Software* 3, 253-256.

Watson, G.S. (1983) *Statistics on Spheres*. University of Arkansas Lecture Notes in the Mathematical Sciences, Volume 6. John Wiley, New York.

Watson, G.S. and Williams, E.J. (1956) On the construction of significance tests on the circle and the sphere. *Biometrika* **43**, 344-352.

Wood, A. (1987) The simulation of spherical distributions in the Fisher-Bingham family. *Commun. Statist. -Simula.* **16**, 885-898.

Appendix 1

SELECTED FORTRAN SUBROUTINES

A1.1 TNRML(AS,X), truncated normal

```
      SUBROUTINE TNRML(AS,X)
C
C  SUBROUTINE GENERATES A TRUNCATED NORMAL VARIATE X
C    DEFINED ON (-1,1) OF MEAN ZERO AND VARIANCE 1/AS.
C
      DATA PI2/6.283185307/
      DATA IN,INIT/0,0/
      IF (INIT.EQ.0) THEN
        B = 1.0 - EXP(-AS)
        INIT = 1
      ENDIF
      IF (IN.EQ.1) THEN
        X = Q*SIN(ETA)
        IN = 0
        RETURN
      ENDIF
2     RS = -2*ALOG(1 - RNUNF()*B)
      ETA = PI2*RNUNF()
      C = COS(ETA)
      Z1S = RS*C*C
      Z2S = RS - Z1S
      IF (Z1S.GT.AS .OR. Z2S.GT.AS) GOTO 2
      Q = SQRT(RS/AS)
      X = Q*C
      IN = 1
      RETURN
      END
```

A1.2 RPN1(N,X), random point

```
      SUBROUTINE RPN1(N,X)
C
C  SUBROUTINE GENERATES A RANDOM POINT ON THE N-SPHERE,
C    USING MULLER'S (1959) METHOD.
C  OUTPUT:  X(1), X(2),..., X(N)
C    ARE THE CARTESIAN COORDINATES OF THE RANDOM POINT.
C
C  RNNOF IS A STANDARD NORMAL GENERATOR (FUNCTION) FROM IMSL.
C
      REAL X(N)
      S = 0.0
      DO 20 J = 1,N
        X(J) = RNNOF()
        S = S + X(J)*X(J)
```

```
20    CONTINUE
      SRI = 1.0/SQRT(S)
      DO 30 J = 1,N
30    X(J) = X(J)*SRI
      RETURN
      END
```

A1.3 RPN2(N,X), random point

```
      SUBROUTINE RPN2(N,X)
C
C  SUBROUTINE GENERATES A RANDOM POINT ON THE N-SPHERE,
C    USING SIBUYA'S (1962) METHOD.
C  OUTPUT:  X(1), X(2),..., X(N)
C    ARE THE CARTESIAN COORDINATES OF THE RANDOM POINT.
C
C  SVRGN IS A SORTING SUBROUTINE FROM IMSL.
C
      PARAMETER (LM = 500)
      REAL X(N),U(LM)
      DATA PI2/6.283185307/
      IF (N.LT.2 .OR. N.GE.2*LM) THEN
        WRITE(6,100) 2*LM-1
100     FORMAT(' DIMENSIONS SMALLER THAN 2 OR GREATER THAN',I5)
        STOP
      ENDIF
      L = N/2
      HN = .5*FLOAT(N)
      IF (L.NE.HN) L = L + 1
      LL = L - 1
      U(L) = 1.0
      IF (N.GT.2) THEN
        CALL RNUN(LL,U)
        CALL SVRGN(LL,U,U)
      ENDIF
      DO 30 I = 1,L
        IF (I.EQ.1) THEN
          DIFFR = SQRT(U(1))
        ELSE
          DIFFR = SQRT(U(I)-U(I-1))
        ENDIF
        CALL RNUN(1,R)
        ETA = PI2*R
        X(2*I-1) = DIFFR*COS(ETA)
        IF (I.EQ.L .AND. L.NE.HN) THEN
          XN = DIFFR*SIN(ETA)
        ELSE
          X(2*I) = DIFFR*SIN(ETA)
        ENDIF
30    CONTINUE
      IF (L.NE.HN) THEN
        SRI = 1.0/SQRT(1.0 - XN*XN)
        DO 40 I = 1,N
40      X(I) = X(I)*SRI
```

```
      ENDIF
      RETURN
      END
```

A1.4 RPN3(N,X), random point

```
      SUBROUTINE RPN3(N,X)
C
C  SUBROUTINE GENERATES A RANDOM POINT ON THE N-SPHERE,
C    USING A MODIFIED SIBUYA'S (1962) METHOD.
C  OUTPUT:  X(1), X(2),..., X(N)
C    ARE THE CARTESIAN COORDINATES OF THE RANDOM POINT.
C
C  ORDERED UNIFORMS ON (0,1) ARE GENERATED BY
C    EXPONENTIAL-SPACINGS METHOD.
C
      PARAMETER (LM = 500)
      REAL X(N),U(LM)
      DATA PI2/6.283185307/
      IF (N.LT.2 .OR. N.GE.2*LM) THEN
        WRITE(6,100) 2*LM-1
100     FORMAT(' DIMENSION SMALLER THAN 2 OR GREATER THAN',I5)
        STOP
      ENDIF
      L = N/2
      HN = 0.5*FLOAT(N)
      IF (L.NE.HN) L = L+1
      CALL RNUN(L,U)
      S = 0.0
      DO 20 I = 1,L
        U(I) = -ALOG(U(I))
        S = S + U(I)
20    CONTINUE
      DO 25 I = 1,L
25    U(I) = U(I)/S
      DO 30 I = 1,L
        DIFFR = SQRT(U(I))
        CALL RNUN(1,R)
        ETA = PI2*R
        X(2*I-1) = DIFFR*COS(ETA)
        IF (I.EQ.L.AND.L.NE.HN) THEN
          XN = DIFFR*SIN(ETA)
        ELSE
          X(2*I) = DIFFR*SIN(ETA)
        ENDIF
30    CONTINUE
      IF (L.NE.HN) THEN
        SRI = 1.0/SQRT(1.0 - XN*XN)
        DO 40 I = 1,N
40      X(I) = X(I)*SRI
      ENDIF
      RETURN
      END
```

A1.5 RPN4(N,X), random point

```
      SUBROUTINE RPN4(N,X)
C
C  SUBROUTINE GENERATES A RANDOM POINT ON THE N-SPHERE,
C    USING TASHIRO'S (1976) METHOD.
C  OUTPUT:  X(1), X(2),..., X(N)
C    ARE THE CARTESIAN COORDINATES OF THE RANDOM POINT.
C
      PARAMETER (LM = 500)
      REAL X(N),U(LM)
      DATA PI2/6.283185307/
      IF (N.LT.2.OR.N.GE.2*LM) THEN
        WRITE(6,100) 2*LM-1
100     FORMAT(' DIMENSION SMALLER THAN 2 OR GREATER THAN',I5)
        STOP
      ENDIF
      L = N/2
      HN = 0.5*FLOAT(N)
      IF (L.EQ.HN) THEN
        LL = L - 1
        U(L) = 1.0
        IF (N.GT.2) THEN
          CALL RNUN(LL,U)
          DO 20 I = LL,1,-1
20        U(I) = U(I+1)*U(I)**(1.0/FLOAT(I))
          ENDIF
        DO 30 I = 1,L
          IF (I.EQ.1) THEN
            DIFFR = SQRT(U(1))
          ELSE
            DIFFR = SQRT(U(I) - U(I-1))
          ENDIF
          CALL RNUN(1,R)
          ETA = PI2*R
          X(2*I-1) = DIFFR*COS(ETA)
          X(2*I) = DIFFR*SIN(ETA)
30      CONTINUE
      ELSE
        U(L+1) = 1.0
        CALL RNUN(L,U)
        DO 70 I = L,1,-1
70      U(I) = U(I+1)*U(I)**(2.0/FLOAT(2*I - 1))
        CALL RNUN(1,R)
        X(1) = SQRT(U(1))
        IF (R.LT.0.5)  X(1) = -X(1)
        DO 80 I = 1,L
          DIFFR = SQRT(U(I+1) - U(I))
          CALL RNUN(1,R)
          ETA = PI2*R
          X(2*I) = DIFFR*COS(ETA)
          X(2*I+1) = DIFFR*SIN(ETA)
80      CONTINUE
      ENDIF
```

```
      RETURN
      END
```

A1.6 RP3(X), random point

```
      SUBROUTINE RP3(X)
C
C  SUBROUTINE GENERATES A RANDOM POINT ON THE 3-SPHERE,
C    USING MASAGLIA'S (1972) METHOD.
C  OUTPUT:  X(1), X(2), X(3)
C    ARE THE CARTESIAN COORDINATES OF THE RANDOM POINT.
C
      REAL X(3)
1     CALL RNUN(2,X)
      X(1) = 2.0*X(1) - 1.0
      X(2) = 2.0*X(2) - 1.0
      S = X(1)*X(1) + X(2)*X(2)
      IF (S.GT.1.0) GOTO 1
      SS = 2.0*SQRT(1.0 - S)
      X(1) = X(1)*SS
      X(2) = X(2)*SS
      X(3) = 1.0 - 2.0*S
      RETURN
      END
```

A1.7 RP4(X), random point

```
      SUBROUTINE RP4(X)
C
C  SUBROUTINE GENERATES A RANDOM POINT ON THE 4-SPHERE,
C    USING MASAGLIA'S (1972) METHOD.
C  OUTPUT:  X(1), X(2), X(3), X(4)
C    ARE THE CARTESIAN COORDINATES OF THE RANDOM POINT.
C
      REAL X(4),U(2)
1     CALL RNUN(2,U)
      X(1) = 2.0*U(1) - 1.0
      X(2) = 2.0*U(2) - 1.0
      S1 = X(1)*X(1) + X(2)*X(2)
      IF (S1.GT.1.0) GOTO 1
2     CALL RNUN(2,U)
      X(3) = 2.0*U(1) - 1.0
      X(4) = 2.0*U(2) - 1.0
      S2 = X(3)*X(3) + X(4)*X(4)
      IF (S2.GT.1.0) GOTO 2
      SS = SQRT((1.0 - S1)/S2)
      X(3) = X(3)*SS
      X(4) = X(4)*SS
      RETURN
      END
```

A1.8 **VMBF(M,K,THETA), von Mises**

```
      SUBROUTINE VMBF(M,K,THETA)
C
C  SUBROUTINE GENERATES M RANDOM POINTS (THETA) FROM THE VON MISES
C     DISTRIBUTION ON THE CIRCLE WITH DENSITY PROPORTIONAL TO
C     EXP(K*COS(THETA)),  WHERE THETA IN [0,PI],
C     USING BEST AND FISHER'S (1979) METHOD.
C
C  PARAMETER: K > 0
C
      REAL K,THETA(M)
      DATA PI/3.141592654/
      IF (K.LE.0.0) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS NOT GREATER THAN ZERO')
        STOP
      ENDIF
      TAU = 1.0 + SQRT(1.0 + 4.0*K*K)
      RHO = (TAU - SQRT(2.0*TAU))*0.5/K
      R = 0.5/RHO + 0.5*RHO
C
      DO 10 I = 1,M
2       Z = COS(PI*RNUNF())
        F = (1.0 + R*Z)/(R + Z)
        C = K*(R - F)
        T1 = C*(2.0 - C)
        U = RNUNF()
        IF (U.GE.T1) THEN
          T2 = ALOG(C/U) + 1.0 - C
          IF (T2.LT.0.0) GOTO 2
        ENDIF
        THETA(I) = ACOS(F)
        IF (RNUNF().LT.0.5)  THETA(I) = -THETA(I)
10    CONTINUE
      RETURN
      END
```

A1.9 **VMTN(M,K,THETA), von Mises**

```
      SUBROUTINE VMTN(M,K,THETA)
C
C  SUBROUTINE GENERATES M RANDOM POINTS (THETA) FROM THE VON MISES
C     DISTRIBUTION ON THE CIRCLE WITH DENSITY PROPORTIONAL TO
C     EXP(K*COS(THETA)),  WHERE THETA IN [0,PI],
C     USING AN ENVELOPE-REJECTION METHOD.
C
C  PARAMETER: K > 0
C
      REAL K,THETA(M)
      DATA V/2.4674011/
C         V = PI*PI/4
      IF (K.LE.0.0) THEN
```

```
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS NOT GREATER THAN ZERO')
        STOP
      ENDIF
      DO 10 I = 1,M
1       CALL TNORM(K,V,THETA(I))
        T = 1.0 + K*(COS(THETA(I)) + 0.5*THETA(I)*THETA(I)/V - 1.0)
        U = RNUNF()
        IF (U.GT.T) THEN
          T = EXP(T - 1.0)
          IF (U.GT.T) GOTO 1
        ENDIF
10    CONTINUE
      RETURN
      END

      SUBROUTINE TNORM(K,V,X)
C
C  SUBROUTINE GENERATES A TRUNCATED NORMAL VARIATE X
C    DEFINED ON (-PI,PI) WITH MEAN ZERO AND VARIANCE V/K.
C
      REAL K
      DATA PI2/6.283185307/
      DATA IN,INIT/0,0/
      IF (INIT.EQ.0) THEN
        SIGMAS = V/K
        AS = 4.0*K
        B = 1.0 - EXP(-AS)
        INIT=1
      ENDIF
      IF (IN.EQ.1) THEN
        X = W*SIN(ETA)
        IN = 0
        RETURN
      ENDIF
2     RS = -2.0*ALOG(1.0 - RNUNF()*B)
      ETA = PI2*RNUNF()
      C = COS(ETA)
      Z1S = RS*C*C
      Z2S = RS - Z1S
      IF (Z1S.GT.AS.OR.Z2S.GT.AS) GOTO 2
      W = SQRT(RS*SIGMAS)
      X = W*C
      IN = 1
      RETURN
      END
```

A1.10 FSHR(M,K,THETA,PHI), Fisher

```
      SUBROUTINE FSHR(M,K,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE FISHER
C    DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C    EXP(K*COS(THETA))*SIN(THETA),
```

```
C     WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C     USING FISHER ET AL'S (1981) METHOD.
C
C  PARAMETER:  K > 0
C
      REAL K,THETA(M),PHI(M)
      DATA PI2/6.283185307/
      IF (K.LE.0.0) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS NOT GREATER THAN ZERO')
        STOP
      ENDIF
      B = EXP(-2.0*K)
C
      DO 10 I = 1,M
        Y = B + (1.0-B)*RNUNF()
        THETA(I) = 2.0*ASIN( SQRT(-0.5*ALOG(Y)/K) )
        PHI(I) = PI2*RNUNF()
10    CONTINUE
      RETURN
      END
```

A1.11 ARND(M,K,THETA,PHI), Arnold

```
      SUBROUTINE ARND(M,K,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE ARNOLD
C     DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C     EXP(-K*ABS(COS(THETA)))*SIN(THETA),
C     WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C     USING AN INVERSION METHOD.
C
C  PARAMETER:  K > 0
C
      REAL K,THETA(M),PHI(M)
      DATA PI2/6.283185307/
      IF (K.LE.0.0) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS NOT GREATER THAN ZERO')
        STOP
      ENDIF
      B = 1.0 - EXP(-K)
C
      DO 10 I = 1,M
        U = RNUNF()
        IF (U.GT.0.5) THEN
          THETA(I) = ACOS( ALOG(1.0 + (1.0-2.0*U)*B)/K )
        ELSE
          THETA(I) = ACOS(-ALOG(1.0 - (1.0-2.0*U)*B)/K )
        ENDIF
        PHI(I) = PI2*RNUNF()
10    CONTINUE
      RETURN
      END
```

A1.12 SLBY(M,K,THETA,PHI), Selby

```
      SUBROUTINE SLBY(M,K,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE SELBY
C    DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C    EXP(K*SIN(THETA))*SIN(THETA),
C    WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C    USING AN ENVELOPE-REJECTION METHOD.
C
C  PARAMETER:  K > 0
C
      REAL K,THETA(M),PHI(M)
      DATA PI2/6.283185307/
      IF (K.LE.0.0) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS NOT GREATER THAN ZERO')
        STOP
      ENDIF
C
      DO 10 I = 1,M
1       CALL TNRML(K,X)
        T = 1.0 + K*(SQRT(1.0-X*X) + 0.5*X*X -1.0)
        V = RNUNF()
        IF (V.GT.T) THEN
          T = EXP(T - 1.0)
          IF (V.GT.T) GOTO 1
        ENDIF
        THETA(I) = ACOS(X)
        PHI(I) = PI2*RNUNF()
10    CONTINUE
      RETURN
      END
```

A1.13 DWBF(M,K,THETA,PHI), Dimroth-Watson

```
      SUBROUTINE DWBF(M,K,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE DIMROTH-
C    WATSON DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C    EXP(K*COS(THETA)**2)*SIN(THETA),
C    WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C    USING BEST AND FISHER'S (1986) METHOD.
C
C  PARAMETER:  K ANY REAL NUMBER EXCEPT ZERO
C
      REAL K,THETA(M),PHI(M)
      DATA PI,PI2,PI4/3.141592654,6.283185307,12.56637061/
C         PI4 = 4*PI
      IF (K.EQ.0.0) THEN
        WRITE(6,100)
```

```
100     FORMAT(' PARAMETER K IS ZERO')
        STOP
      ENDIF
C
      IF (K.GT.0.0) THEN
        C = 1.0/(EXP(K) - 1.0)
        DO 10 I = 1,M
1         X = ALOG(RNUNF()/C + 1.0)/K
          T = 1.0 + K*X*(X - 1.0)
          U = RNUNF()
          IF (U.GT.T) THEN
            T = EXP(T - 1.0)
            IF (U.GT.T) GOTO 1
          ENDIF
          THETA(I) = ACOS(Y)
          V = RNUNF()
          IF (V.LT.0.5) THEN
            THETA(I) = PI - THETA(I)
            PHI(I) = PI4*V
          ELSE
            PHI(I) = PI4*V - PI2
          ENDIF
10      CONTINUE
      ELSE
        Z = SQRT(-K)
        B = ATAN(Z)
        DO 20 I = 1,M
2         X = TAN(B*RNUNF())
          W = X*X
          T = 1.0 - W*W
          U = RNUNF()
          IF (U.GT.T) THEN
            T = (1.0 + W) * EXP(-W)
            IF (U.GT.T) GOTO 2
          ENDIF
          THETA(I) = ACOS(X/Z)
          V = RNUNF()
          IF (V.LT.0.5) THEN
            THETA(I) = PI - THETA(I)
            PHI(I) = PI4*V
          ELSE
            PHI(I) = PI4*V - PI2
          ENDIF
20      CONTINUE
      ENDIF
      RETURN
      END
```

A1.14 DWAG(M,K,THETA,PHI), Dimroth-Watson

```
      SUBROUTINE DWAG(M,K,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE DIMROTH-
C    WATSON DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
```

```
C     EXP(K*COS(THETA)**2)*SIN(THETA),
C     WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C     USING AN ENVELOPE-REJECTION METHOD.
C
C   PARAMETER:  K ANY REAL NUMBER EXCEPT ZERO
C
      REAL K,THETA(M),PHI(M)
      DATA PI,PI2,PI4/3.141592654,6.283185307,12.566370610/
C         PI4 = 4*PI
      IF (K.EQ.0.0) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS ZERO')
        STOP
      ENDIF
      H = 2.0*K + 3.0
      P = (H - SQRT(H*H - 16.0*K))*.25
      G = 1.5*P/K
      Q = EXP(-3.0 + 3.0/G)*G*G*G
C
      DO 10 I = 1,M
1       U1 = RNUNF()
        U2 = RNUNF()
        U1S = U1*U1
        U2S = U2*U2
        S = U1S/(1.0 - P*(1.0 - U1S))
        W = 1.0 + K*S
        V = 1.0 - P*S
        R = V*V*V/Q
        T = W*W*R
        IF (W.LT.0.0 .OR. U2S.GT.T) THEN
          T = EXP(2.0*K*S)*R
          IF (U2S.GT.T) GOTO 1
        ENDIF
        THETA(I) = ACOS(SQRT(S))
        V = RNUNF()
        IF (V.LE.0.5) THEN
          THETA(I) = PI - THETA(I)
          PHI(I) = PI4*V
        ELSE
          PHI(I) = PI4*V - PI2
        ENDIF
10    CONTINUE
      RETURN
      END
```

A1.15 DWTN(M,K,THETA,PHI), Dimroth-Watson

```
      SUBROUTINE DWTN(M,K,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE DIMROTH-
C    WATSON DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C    EXP(K*COS(THETA)**2)*SIN(THETA),
C    WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C    USING TRUNCATED NORMAL DISTRIBUTION.
```

```
C
C  PARAMETER:  K < 0
C
      REAL K,THETA(M),PHI(M)
      DATA PI2/6.283185307/
      IF (K.GE.0.0) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS NOT SMALLER THAN ZERO')
        STOP
      ENDIF
C
      DO 10 I = 1,M
        CALL TNRML(-2.0*K,X)
        THETA(I) = ACOS(X)
        PHI(I) = PI2*RNUNF()
10    CONTINUE
      RETURN
      END
```

A1.16 BH(M,K1,K2,THETA,PHI), Bingham (3-dim)

```
      SUBROUTINE BH(M,K1,K2,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE BINGHAM
C    DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C    EXP((K1*COS(THETA)**2 + K2*SIN(THETA)**2)*SIN(THETA)**2)*SIN(THETA),
C    WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C    USING A METHOD DERIVED FROM WOOD (1987).
C
C  PARAMETER:  K1 AND K2 ARE ANY REAL NUMBERS EXCEPT ZEROES
C
C  DAWS AND ERF ARE IMSL SUBOUTINES FOR EVALUATING THE DAWSON'S INTEGRAL
C    AND THE ERROR FUNCTION RESPECTIVELY.
C
      REAL K1,K2,KA,K1A,K2A,THETA(M),PHI(M)
      DATA PI/3.141592654/
      DATA V/1.128379167/
C         V = 2/SQRT(PI)
      IF (K1.EQ.0.0 .OR. K2.EQ.0.0) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETERS K1 OR K2 OR BOTH IS ZERO')
        STOP
      ENDIF
      K1A = SQRT(ABS(K1))
      K2A = SQRT(ABS(K2))
      KA = K1A/K2A
      IF (K1.LT.0.0 .AND. K2.LT.0.0) THEN
        D1 = DAWS(K1A)
        D2 = DAWS(K2A)
        P = D1/(D1 + KA*D2)
      ELSEIF (K1.GT.0.0 .AND. K2.GT.0.0) THEN
        E1 = ERF(K1A)
        E2 = ERF(K2A)
        P = E1/(E1 + EXP(K2 - K1)*KA*E2)
```

```
      ELSEIF (K1.GT.0.0 .AND. K2.LT.0.0) THEN
        E1 = ERF(K1A)
        D2 = DAWS(K2A)
        P = E1/(E1 + V*EXP(-K1)*KA*D2)
      ELSE
        D1 = DAWS(K1A)
        E2 = ERF(K2A)
        P = D1/(D1 + EXP(K2)*KA*E2/V)
      ENDIF
      DO 10 I = 1,M
1       IF (RNUNF().LT.P) THEN
          CALL DW(-K1,TS)
        ELSE
          CALL DW(-K2,TS)
        ENDIF
        R = 0.5*(K1 - K2)*(1.0 - TS)
        T = BSIO(R)/COSH(R)
        IF (RNUNF().GT.T) GOTO 1
        THETA(I) = ACOS(SQRT(TS))
        IF (RNUNF().LT.0.5)  THETA(I) = PI - THETA(I)
        CALL VMBF(1,R,PSI)
        PHI(I) = 0.5*PSI
        IF (PSI.LT.0.0)  PHI(I) = PHI(I) + PI
        IF (RNUNF().LT.0.5)  PHI(I) = PHI(I) + PI
10    CONTINUE
      RETURN
      END

      SUBROUTINE DW(K,TS)
C
C  SUBROUTINE DW GENERATES THE MIXTURE VARIABLE T (IN FACT, TS = T**2  IS
C    GENERATED) USING AN ENVELOPE PROPORTIONAL TO (1 - P*X**2)**(-3/2).
C
      REAL K
      DATA PI/3.141592654/
      H = 2.0*K + 3.0
      P = (H - SQRT(H*H - 16.0*K))*.25
      G = 1.5*P/K
      Q = EXP(-3.0 + 3.0/G)*G*G*G
2     U1 = RNUNF()
      U2 = RNUNF()
      U1S = U1*U1
      U2S = U2*U2
      TS = U1S/(1.0 - P*(1.0 - U1S))
      W = 1.0 + K*TS
      V = 1.0 - P*TS
      R = V*V*V / Q
      T = W*W*R
      IF (W.LT.0.0 .OR. U2S.GT.T) THEN
        T = EXP(2.0*K*TS)*R
        IF (U2S.GT.T) GOTO 2
      ENDIF
      RETURN
      END
```

A1.17 BHM(M,K1,K2,THETA,PHI), Bingham (3-dim)

```
      SUBROUTINE BHM(M,K1,K2,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE BINGHAM
C    DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C    EXP((K1*COS(THETA)**2 + K2*SIN(THETA)**2)*SIN(THETA)**2)*SIN(THETA),
C    WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C    USING JOHNSON'S (1987) METHOD.
C
C  PARAMETER:  K1 AND K2 ARE ANY REAL NUMBERS WITH THE LARGEST NOT
C              EQUAL TO ZERO.
C
      REAL K,K1,K2,KA,THETA(M),PHI(M)
      DATA PI,PI2/3.141592654,6.283185307/
      K = AMAX1(K1,K2)
      IF (K.EQ.0.0) THEN
        WRITE(6,100)
100     FORMAT(' THE LARGEST OF K1 AND K2 IS ZERO')
        STOP
      ENDIF
      DK = K1 - K2
      D = EXP(.75*K) - 1.0
      DELTA = PI2*D/K
      P = DELTA/(DELTA + PI*EXP(K))
C
      DO 10 I = 1,M
1       PHI(I) = PI2*RNUNF()
        CP = COS(PHI(I))
        KA=DK*CP*CP + K2
        IF (RNUNF().LT.P) THEN
          STS = ALOG(1.0 + RNUNF()*D)/K
          CT = SQRT(1.0 - STS)
          W = 1.0 + (KA - K)*STS
          U = RNUNF()
          IF (U.GT.W*0.5/CT) THEN
            T = EXP(W - 1.0)*0.5/CT
            IF (U.GT.T) GOTO 1
          ENDIF
          THETA(I) = ACOS(CT)
        ELSE
          U = RNUNF()
          STS = 1.0 - 0.25*U*U
          W = 1.0 + KA*STS - K
          U = RNUNF()
          IF (U.GT.W) THEN
            T = EXP(W - 1.0)
            IF (U.GT.T) GOTO 1
          ENDIF
          THETA(I) = ASIN(SQRT(STS))
        ENDIF
        IF (RNUNF().LT.0.5)  THETA(I) = PI - THETA(I)
10    CONTINUE
      RETURN
```

```
      END
```

A1.18 BGM(M,K1,K2,THETA,PHI), Bingham (3-dim)

```
      SUBROUTINE BGM(M,K1,K2,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE BINGHAM
C     DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C     EXP((K1*COS(THETA)**2 + K2*SIN(THETA)**2)*SIN(THETA)**2)*SIN(THETA),
C     WHERE THETA IN [0,PI] AND PHI IN [0,PI2=2*PI),
C     USING AN ENVELOPE-REJECTION METHOD.
C
C  PARAMETER:  K1 AND K2 ARE ANY REAL NUMBERS WITH THE LARGEST NOT
C              EQUAL TO ZERO.
C
      REAL K,K1,K2,KO,KA,THETA(M),PHI(M)
      DATA PI,PI2/3.141592654,6.283185307/
      K = AMAX1(K1,K2)
      IF (K.EQ.0.0) THEN
        WRITE(6,100)
100     FORMAT(' THE LARGEST OF K1 AND K2 IS ZERO')
        STOP
      ENDIF
      DK = K1 - K2
      H = -2.0*K + 3.0
      P = (H - SQRT(H*H + 16.0*K))*.25
      G = -1.5*P/K
      Q = EXP(-3.0 + 3.0/G)*G*G*G
C
      DO 10 I = 1,M
1       PHI(I) = PI2*RNUNF()
        CP = COS(PHI(I))
        KA = DK*CP*CP + K2
        U1 = RNUNF()
        U2 = RNUNF()
        U1S = U1*U1
        U2S = U2*U2
        S = U1S/(1.0 - P*(1.0-U1S))
        W = 1.0 + KA*(1.0 - S) - K
        V = 1.0 - P*S
        R = V*V*V/Q
        T = W*W*R
        IF (W.LT.0.0 .OR. U2S.GT.T) THEN
          T = EXP(2.0*W - 2.0)*R
          IF (U2S.GT.T) GOTO 1
        ENDIF
        THETA(I) = ACOS(SQRT(S))
        IF (RNUNF().LT.0.5)  THETA(I) = PI - THETA(I)
10    CONTINUE
      RETURN
      END
```

A1.19 BM(M,K,L,THETA,PHI), Bingham-Mardia

```
      SUBROUTINE BM(M,K,L,THETA,PHI)
C
C  SUBROUTINE GENERATES M RANDOM VECTORS (THETA, PHI) FROM THE BINGHAM-
C     MARDIA DISTRIBUTION ON THE SPHERE WITH DENSITY PROPORTIONAL TO
C     EXP(K*(COS(THETA) - L)**2)*SIN(THETA),
C     WHERE THETA IN [O,PI] AND PHI IN [0,PI2=2*PI),
C     USING AN ENVELOPE-REJECTION METHOD.
C
C  PARAMETER:  K IS ANY REAL NUMBER EXCEPT ZERO;
C              L IS ANY REAL NUMBER IN [0,1]
C
      REAL K,L,THETA(M),PHI(M)
      REAL LS(2),H(2),P(2),G(2),Q(2),D(2)
      DATA PI2/6.283185307/
      IF (K.EQ.0.0 .OR. L.LT.0.0 .OR. L.GT.1.0) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS ZERO OR L IS OUT OF THE RANGE [0,1]')
        STOP
      ENDIF
      LS(1) = (1.0 + L)*(1.0 + L)
      LS(2) = (1.0 - L)*(1.0 - L)
      N = 2
      IF (L.EQ.1.0) N = 1
      DO 6 J = 1,N
        H(J) = 3.0 + 2.0*K*LS(J)
        P(J) = (H(J) - SQRT(H(J)*H(J) - 16.0*K*LS(J)))*0.25/LS(J)
        G(J) = 1.5*P(J)/K
        Q(J) = EXP(-3.0 + 3.0/G(J))*G(J)*G(J)*G(J)
        D(J) = SQRT(LS(J)/(1.0 - P(J)*LS(J)))
6     CONTINUE
      IF (N.EQ.1) THEN
        SPR1 = 1.0
      ELSE
        SPR1 = D(1)*SQRT(Q(1))/(D(1)*SQRT(Q(1)) + D(2)*SQRT(Q(2)))
      ENDIF
C
      DO 10 I = 1,M
1       IF (RNUNF().LE.SPR1) THEN
          J = 1
        ELSE
          J = 2
        ENDIF
        U1 = RNUNF()
        U2 = RNUNF()
        U1S = U1*U1
        U2S = U2*U2
        YS = LS(J)*U1S/(1.0 - P(J)*LS(J)*(1.0 - U1S))
        W = 1.0 + K*YS
        V = 1.0 - P(J)*YS
        R = V*V*V/Q(J)
        T = W*W*R
        IF (W.LT.0.0 .OR. U2S.GT.T) THEN
          T = EXP(2.0*W - 2.0)*R
```

```
          IF (U2S.GT.T) GOTO 1
        ENDIF
        IF (J.EQ.1) THEN
          X = -SQRT(YS)
        ELSE
          X = SQRT(YS)
        ENDIF
        THETA(I) = ACOS(X + L)
        PHI(I) = PI2*RNUNF()
10    CONTINUE
      RETURN
      END
```

A1.20 VMFPS(N,K,X), von Mises-Fisher

```
      SUBROUTINE VMFPS(N,K,X)
C
C  SUBROUTINE GENERATES A RANDOM POINT FROM THE VON MISES-FISHER
C     DISTRIBUTION ON THE N-SPHERE USING POWER METHOD.
C  OUTPUT:  X(1), X(2),..., X(N)
C     ARE THE CARTESIAN COORDINATES OF THE RANDOM POINT.
C
C  PARAMETER:  K > 0
C  DIMENSION:  N > 2
C
      REAL K,X(N),THETA(100),S(100)
      DATA PI2/6.283185307/
      IF (K.LE.0.0 .OR. N.LT.3) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS NOT GREATER THAN ZERO OR DIMENSION IS',
     &         'SMALLER THAN 3')
        STOP
      ENDIF
      NN = N - 3
1     Y = -1.0 + ALOG((EXP(2.0*K) - 1.0)*RNUNF() + 1.0)/K
      T = SQRT(1.0 - Y*Y)**NN
      IF (RNUNF().GT.T) GOTO 1
      THETA(1) = ACOS(Y)
      DO 20 J = 2,NN
20    CALL PWS(J,THETA(NN-J+2))
      IF (N.GT.3)  THETA(N-2) = ACOS(1.0 - 2.0*RNUNF())
      THETA(N-1) = PI2*RNUNF()
C
C        TRANSFORM THETA TO X THROUGH POLAR TRANSFORMATION
C
      S(1) = SIN(THETA(1))
      DO 30 I = 2,N-1
        S(I) = S(I-1)*SIN(THETA(I))
        X(N-I+1) = S(I-1)*COS(THETA(I))
30    CONTINUE
      X(1) = S(N-1)
      X(N) = COS(THETA(1))
      RETURN
      END
```

```
      SUBROUTINE PWS(J,X)
C
C  SUBROUTINE GENERATES A VARIATE X FROM THE POWER SINE
C    DISTRIBUTION WITH DENSITY PROPORTIONAL TO
C    SIN(X)**J,  WHERE  J > 1  AND  0 < X < PI,
C    USING JOHNSON'S (1987) METHOD.
C
      DATA PI,PIH/3.141592654,1.570796327/
C         PIH = PI/2
2     X = PI*RNUNF()
      BL = SIN(X)**J
      BU = 1.0 - ABS(COS(X))**J
      V = RNUNF()
      IF (V.GT.BL .AND. V.LT.BU) GOTO 2
      IF (V.LE.BL) THEN
        RETURN
      ELSE
        IF (X.LE.PIH) THEN
          X = X + PIH
        ELSE
          X = X - PIH
        ENDIF
      ENDIF
      RETURN
      END
```

A1.21 VMFU(N,K,X), von Mises-Fisher

```
      SUBROUTINE VMFU(N,K,X)
C
C  SUBROUTINE GENERATES A RANDOM POINT FROM THE VON MISES-FISHER
C    DISTRIBUTION ON THE N-SPHERE USING ULRICH'S (1984) METHOD.
C  OUTPUT:  X(1), X(2),..., X(N)
C    ARE THE CARTESIAN COORDINATES OF THE RANDOM POINT.
C
C  PARAMETER:  K > 0
C  DIMENSION:  N > 2
C
      REAL K,N1,X(N)
      DATA E/0.223143551/
C         E = LN(1.25)
      IF (K.LE.0.0 .OR. N.LT.3) THEN
        WRITE(6,100)
100     FORMAT(' PARAMETER K IS NOT GREATER THAN ZERO OR DIMENSION IS',
     &         'SMALLER THAN 3')
        STOP
      ENDIF
      N1 = FLOAT(N-1)
      B = (-2.*K + SQRT(4.*K*K + N1*N1))/N1
      D = N1*(1.0 - ALOG(0.5*N1))
1     CALL SBETA(0.5*N1,Z)
      U = RNUNF()
      T = 0.5*N1*(1.0 + B) / (1.0 - (1.0 - B)*Z)
```

```
      Q = N + E + N1*ALOG(N1) + D - N1*N1/T - T - 1.25*U
      IF (Q.LT.0.0) THEN
        Q = N1*ALOG(T) - T + D - ALOG(U)
        IF (Q.LT.0.0) GOTO 1
      ENDIF
      NN = N - 1
      CALL RPN1(NN,X)
      X(N) = (1.0 - (1.0 + B)*Z) / (1.0 - (1.0 - B)*Z)
      DO 20 J = 1,NN
      X(J) = X(J)*SQRT(1.0 - X(N)*X(N))
20    CONTINUE
      RETURN
      END

      SUBROUTINE SBETA(A,Z)
C
C  SUBROUTINE GENERATES A SYMMETRIC BETA VARIATE Z WITH PARAMETER A
C
      REAL A,Z
 2    U = 2.0*RNUNF() - 1.0
      V = RNUNF()
      R = U*U + V*V
      IF (R.GT.1.0) GOTO 2
      Z = 0.5 + U*V*SQRT(1.0 - R**(1.0/(A - 0.5)))/R
      RETURN
      END
```

A1.22 NBG(N,K,X), Bingham (n-dim)

```
      SUBROUTINE NBG(N,K,THETA)
C
C  SUBROUTINE GENERATES A RANDOM POINT FROM THE BINGHAM DISTRIBUTION
C    ON THE N-SPHERE USING POWER METHOD.
C  OUTPUT:  THETA(1), THETA(2),..., THETA(N-1)
C    ARE THE POLAR COORDINATES OF THE RANDOM POINT.
C
C  PARAMETER:  K(1), K(2),..., K(N-1) WITH THE LARGEST NOT EQUAL TO ZERO
C  DIMENSION:  N > 2
C
C  REMARK:  S(I) = SIN(THETA(I)), C(I) = COS(THETA(I))
C
      PARAMETER (NMAX = 99)
      REAL KO,KM,K(N-1),THETA(N-1),S(NMAX),C(NMAX)
      DATA PI,PI2/3.141592654,6.283185307/
      IF (N.LT.3 .OR. N.GT.NMAX+1) THEN
        WRITE(6,100) NMAX+1
100     FORMAT(' DIMENSION SMALLER THAN 3 OR GREATER THAN',I4)
        STOP
      ENDIF
C
C       FIND KM = MAX( K(1), K(2), ..., K(N-1) )
C
      KM = K(1)
      DO 10 I = 2,N-1
```

```
10    IF (KM.LT.K(I)) KM = K(I)
      IF (KM.EQ.0.0) THEN
        WRITE(6,105)
105     FORMAT(' THE LARGEST OF K(1),..., K(N-1) IS ZERO')
        STOP
      ENDIF
C
C       GENERATE THETA(1) WHOSE DENSITY IS PROPORTIONAL TO
C         EXP(KM*Y**2) * Y**(N-2),
C       WHERE Y = S(1) IS ALSO CALCULATED.
C
      H = -2.0*KM + 3.0
      P = (H - SQRT(H*H + 16.0*KM))*0.25
      G = -1.5*P/KM
      Q = EXP(-3.0 + 3.0/G)*G*G*G
1     U1 = RNUNF()
      U2 = RNUNF()
      U1S = U1*U1
      U2S = U2*U2
      XS = U1S/(1.0 - P*(1.0 - U1S))
      W = 1.0 - KM*XS
      V = 1.0 - P*XS
      R = V*V*V*(1.0 - XS)**(N - 3)/Q
      T = W*W*R
      IF (W.LT.0.0 .OR. U2S.GT.T) THEN
        T = EXP(2.0*W - 2.0)*R
        IF (U2S.GT.T) GOTO 1
      ENDIF
      THETA(1) = ACOS(SQRT(XS))
      IF (RNUNF().LT.0.5)  THETA(1) = PI - THETA(1)
      S(1) = SIN(THETA(1))
C
C       GENERATE THETA(2),..., THETA(N-2) FROM THE POWER SINE
C       DISTRIBUTIONS AND THETA(N-1) FROM U(0,PI2=2*PI).
C       C(2),..,C(N-1) AND S(2),..,S(N-1) ARE ALSO CALCULATED.
C
      DO 20 J = 2,N-3
      CALL PWSA(J,THETA(N-J-1),C(N-J-1),S(N-J-1))
20    CONTINUE
      IF (N.GT.3) THEN
        C(N-2) = 1.0 - 2.0*RNUNF()
        THETA(N-2) = ACOS(C(N-2))
        S(N-2) = SIN(THETA(N-2))
      ENDIF
      THETA(N-1) = PI2*RNUNF()
      C(N-1) = COS(THETA(N-1))
      S(N-1) = SIN(THETA(N-1))
C
C       CALCULATE THE SUM (OVER I=1, 2,..., N-1) OF
C         K(I) * (C(N-I+1)*S(1)*S(2)*...*S(N-I))**2
C       WHERE C(N) = 1.
C
      KO = 0.
      DO 50 I = 1,N-1
      PROD = 1.0
```

```
      DO 60 J = 1,N-I
      PROD = PROD*S(J)
60    CONTINUE
      IF (I.GT.1) PROD = PROD*C(N-I+1)
      PROD = K(I)*PROD*PROD
      KO = KO + PROD
50    CONTINUE
C
C       ACCEPTANCE TEST
C
      V = RNUNF()
      T = 1.0 + KO - KM*S(1)*S(1)
      IF (V.GT.T) THEN
        T = EXP(T - 1.0)
        IF (V.GT.T) GOTO 1
      ENDIF
      RETURN
      END

      SUBROUTINE PWSA(J,X,C,S)
C
C  IT IS IDENTICAL TO SUBROUTINE PWS (SEE A1.20) EXCEPT THAT
C    S = SIN(X) AND C = COS(X) ARE RETURNED AS WELL.
C
      DATA PI,PIH/3.141592654,1.570796327/
C         PIH = PI/2
2     X = PI*RNUNF()
      S = SIN(X)
      C = COS(X)
      BL = S**J
      BU = 1.0 - ABS(C)**J
      V = RNUNF()
      IF (V.GT.BL .AND. V.LT.BU) GOTO 2
      IF (V.LE.BL) THEN
        RETURN
      ELSE
        IF (X.LE.PIH) THEN
          X = X + PIH
          TEMP = C
          C = -S
          S = TEMP
        ELSE
          X = X - PIH
          TEMP = C
          C = S
          S = -TEMP
        ENDIF
      ENDIF
      RETURN
      END
```

APPENDIX 2

<u>Theorem</u> The best upper envelope for the p.d.f.

$$f(x;\ k) = C_f \exp(kx^2), \qquad x \in [0,1], \qquad k \in (-\infty,\infty),$$

with normalizing constant $C_f = \left(\int_0^1 \exp(ku^2)du \right)^{-1}$, which is proportional to the p.d.f.

$$g(x;\ \rho) = C_g\ (1-\rho x^2)^{-3/2}, \qquad x \in [0,1], \qquad \rho < 1,$$

where $C_g = \sqrt{1-\rho}$, is

$$\gamma(x) = C_f\ \beta(1-\rho_* x^2)^{-3/2}, \qquad x \in [0,1],$$

with

$$\rho_* = [\lambda - (\lambda^2-16k)^{1/2}]/4, \qquad \lambda = 2k+3;$$

$$\beta = \alpha^{3/2}\exp(-\tfrac{3}{2}(1 - \tfrac{1}{\alpha})), \qquad \alpha = 3\rho/2k,$$

and that the choice of ρ $(= \rho_*)$ maximizes the sampling efficiency.

<u>Proof</u>: Let

$$\varphi(x;\ k,\rho) = \frac{f(x;\ k)}{g(x;\ \rho)}$$

$$= C_f\ \exp(kx^2)\ (1 - \rho x^2)^{3/2} \Big/ \sqrt{1-\rho}, \qquad \rho < 1.$$

The sampling efficiency of generating a random variable X having p.d.f. $f(x;\ k)$ using an envelope proportional to $g(x;\ \rho)$ is

$$R(k,\rho) = \left(\max_{x\in[0,1]} \varphi(x;\ k,\rho) \right)^{-1}. \qquad \text{(A2.1)}$$

The first step is to find $x_* \in [0,1]$ that maximizes $\varphi(x;\ k,\rho)$. Now

$$\frac{\partial}{\partial x} \ln\varphi(x;\ k,\rho) = 2kx - 3\rho x/(1-\rho x^2)$$

and

$$\frac{\partial^2}{\partial x^2} \ln\varphi(x;\ k,\rho) = 2k - 3\rho(1+\rho x^2)/(1-\rho x^2)^2.$$

The first derivative is zero when $x = 0$ or $x = \sqrt{-3/2k + 1/\rho}$. In order that $x_o \equiv \sqrt{-3/2k + 1/\rho}$ be in $[0,1]$, we require

$$\begin{cases} 2k/(3+2k) \le \rho \le 2k/3, & |k| < 3/2; \\ \rho \le 2k/3, & k \le -3/2; \\ 2k/(3+2k) \le \rho < 1, & k \ge 3/2. \end{cases} \tag{A2.2}$$

Furthermore, by examining the second derivative, it is noted that

$$\max_{x\in[0,1]} \varphi(x;\, k,\rho) = \begin{cases} \varphi(x_o;\, k,\rho), & \text{when (A2.2) holds;} \\ \varphi(1;\, k,\rho), & \text{when } \rho < 2k/(3+2k),\ \ k > -3/2; \\ \varphi(0;\, k,\rho), & \text{when } \rho > 2k/3,\ \ k < 3/2. \end{cases}$$

So to maximize the sampling efficiency (A2.1), it is required to determine the value of ρ such that $\max_{x\in[0,1]} \varphi(x;\, k,\rho)$ is minimized. Let $\rho_1 = 2k/(3+2k)$ and $\rho_2 = 2k/3$. It is obvious that $\varphi(1;\, k,\rho) > \varphi(1;\, k,\rho_1)$ when $\rho < \rho_1$, $k > -3/2$ and $\varphi(0;\, k,\rho) > \varphi(0;\, k,\rho_2)$ when $\rho > \rho_2$, $k < 3/2$. As both ρ_1 and ρ_2 satisfy (A2.2), there exists some ρ satisfying (A2.2) such that

$$\varphi(x_o;\, k,\rho) \le \varphi(1;\, k,\rho_1), \qquad \text{when } k > -3/2$$

and

$$\varphi(x_o;\, k,\rho) \le \varphi(0;\, k,\rho_2), \qquad \text{when } k < 3/2.$$

Therefore, it needs only to find ρ that satisfies (A2.2) and that minimizes

$$\varphi(x_o;\, k,\rho) = C_f \exp(-\frac{3}{2} + \frac{k}{\rho}) \left(\frac{3\rho}{2k} \right)^{3/2} \Big/ \sqrt{1-\rho}. \tag{A2.3}$$

Taking logarithm on both sides of (A2.3), we obtain

$$\begin{aligned} \ln \varphi(x_o;\, k,\rho) &= \text{const.} - \tfrac{1}{2}\ln(1-\rho) + \frac{k}{\rho} + \tfrac{3}{2}\ln(\tfrac{3\rho}{2k}) \\ &\equiv L(\rho), \end{aligned}$$

and so

$$L'(\rho) = [-2k + (2k+3)\rho - 2\rho^2] \,/\, [2(1-\rho)\rho^2]$$

which is equal to zero when

$$\rho = [\lambda \pm (\lambda^2-16k)^{1/2}]/4, \qquad \lambda = 2k+3.$$

Denote $\rho_a = [\lambda + (\lambda^2-16k)^{1/2}]/4$ and $\rho_b = [\lambda - (\lambda^2-16k)^{1/2}]/4$. It can be shown that only ρ_b satisfies (A2.2) and that $L''(\rho_b) > 0$. Therefore, the optimal ρ that maximizes the sampling efficiency is ρ_b and the envelope so obtained is

$$\varphi(x_o)\ g(x;\ \rho_b) \ = \ C_f \exp(-3/2 + k/\rho_b)\ (3\rho_b/2k)^{3/2}\ (1 - \rho_b x^2)^{-3/2}. \qquad \blacksquare$$