International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

Calculation of the Multivariate Probability Distribution Function Values and their Gradient Vectors

Szantai, T.

IIASA Working Paper

WP-87-082

September 1987

CALCULATION OF THE MULTIVARIATE PROBABILITY DISTRIBUTION FUNCTION VALUES AND THEIR GRADIENT VECTORS

T. Szántai

September 1987
WP-87-82

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

FOREWORD

The described collection of subroutines developed for calculation of values of multivariate normal, Dirichlet and gamma distribution functions and their gradient vectors is an unique tool that can be used e.g. to compute the Loss-of-Load Probability of electric networks and to solve optimization problems with a reliability constraint.

Alexander B. Kurzhanski
Chairman
System and Decision Sciences Program

CONTENTS

CALCULATION OF THE MULTIVARIATE PROBABILITY DISTRIBUTION FUNCTION VALUES AND THEIR GRADIENT VECTORS

T. Szántai

Eötvös Loránd University, Budapest

1. INTRODUCTION

This paper describes a subroutine package on the calculation of some multivariate probability distribution function values and their gradient vectors. These calculations are very important in stochastic programming, reliability theory, statistics and practically all sciences that concern stochastic systems. The subroutine package has been developed in FORTRAN-77 language and makes the above described calculations possible in the case of the normal, gamma and Dirichlet distributions. Here the normal and Dirichlet distributions are well known and the multivariate gamma distribution is a new one developed by A. Prékopa and T. Szántai [5].

The main calculation procedure is based on the determination of all possible one- and two dimensional marginal probability distribution function values. By the aid of these we can give sharp lower and upper bounds on the multivariate probability distribution function. In many cases these bounds are close enough and their mean value can be regarded as the exact value of the multivariate probability distribution function. In other cases we use a special Monte Carlo simulation procedure for a more accurate estimation of the distribution function. This is a variance reduction technique as described by T. Szántai in [7] and [8].

The subroutine package has been developed in IMSL form. This means that the codes are supplied with such headings and comments as the usual IMSL subroutines are. In addition our subroutines use the standard IMSL subroutines whenever it is possible and they do not need any more user written supplementary code. So our package can be used on any computer supplied with the IMSL library.

The main subroutines of the package are named MDMNOR, MDMGAM and MDMDIR. They provide the calculation of the multivariate normal, gamma and Dirichlet distribution function values. The further main subroutines named MGMNOR, MGMGAM and MGMDIR calculate the gradient vector of the corresponding multivariate probability distribution functions. For the calculation of the one- and two dimensional normal probability distribution function values we use the MDNOR resp. MDBNOR standard IMSL subroutines. In the case of the gamma and Dirichlet distribution the IMSL library has subroutines only for the one dimensional probability distribution function value calculations. These are named MDGAM and MDBETA. For the calculation of the two dimensional probability distribution functions subroutines MDBGAM and MDBDIR have been developed. As the Monte Carlo simulation procedure requires the generation of the multivariate normal, gamma and Dirichlet distributed random vectors the package must contain subroutines for this purpose too. In the case of the normal distribution the standard IMSL subroutine GGNSM can be used. In the case of the Dirichlet distribution the problem is trivial and a separate subroutine is not needed. However in the case of the multivariate gamma distribution we have to solve not only the problem of the random vector generation but also the problem of the fitting the multivariate gamma probability distribution to the empirical data that is to an empirical covariance matrix. This problem is solved by the new subroutine named GGGML.

The next section of the paper contains a brief description of the algorithms used in the different subroutines. In the third section the usage of the individual subroutines is described in that form as they are contained in the headings of the codes. Finally in the fourth section we demonstrate the application of the subroutines by the solution of some test problems.

2. THE MAIN ALGORITHMS USED IN THE SUBROUTINE PACKAGE

The Monte Carlo simulation procedure for the calculation of the multivariate probability distribution function values was published in [8] for the case of the multivariate gamma distribution. This algorithm has been improved significantly in [7]. In the following we describe this latest version of the algorithm.

For any multivariate probability distribution function we have

$$F(z_1,\dots,z_n) = P(x_1 < z_1,\dots,x_n < z_n) =$$

$$= 1 - P(\bar{A}_1 + \cdots + \bar{A}_n) = 1 - \bar{S}_1 + \bar{S}_2 - \bar{S}_3 + \cdots + (-1)^n \bar{S}_n \ ,$$

where $x_1, \ldots, x_n$ are the components of the random vector x and

$$\bar{A}_i = P(x_i \geq z_i), \quad i = 1, \ldots, n \ ,$$

$$\bar{S}_k = \sum_{1 \leq i_1 < \cdots < i_k \leq n} P(\bar{A}_{i_1} \cdots \bar{A}_{i_k}), \quad k = 1, \ldots, n \ .$$

From the so called Bonferroni inequalities (see [6]) one easily can get the following lower and upper bounds

$$1 - \bar{S}_1 + \frac{2}{n} \bar{S}_2 \leq F(z_1, \ldots, z_n) \leq 1 - \frac{2}{k^* + 1} \bar{S}_1 + \frac{2}{k^*(k^* + 1)} \bar{S}_2 \ ,$$

where k^* is the greatest integer smaller than or equal to $2\bar{S}_2 / \bar{S}_1 + 1$.

As $\bar{S}_1$ and $\bar{S}_2$ can be expressed in terms of values of the one- and two dimensional marginal probability distribution functions these bounds easily can be calculated. The main idea of the algorithm is that three different estimates of the distribution function can be produced in the same Monte Carlo simulation procedure. The first one is the direct relative frequency corresponding to the probability $P(x_1 < z_1, \ldots, x_n < z_n)$. The second one is the relative frequency corresponding to the difference between the upper bound and the distribution function. The third one is the relative frequency corresponding to the difference between the distribution function and its lower bound.

As the above mentioned differences are equal to

$$- \frac{2}{k^* + 1} \bar{S}_1 + \frac{2}{k^*(k^* + 1)} \bar{S}_2 + \bar{S}_1 - \bar{S}_2 + \cdots + (-1)^{n+1} \bar{S}_n$$

and

$$- \frac{2}{n} \bar{S}_2 + \bar{S}_2 - \bar{S}_3 + \cdots + (-1)^n \bar{S}_n$$

if κ denotes the number of the inequalities $x_1 < z_1, \ldots, x_n < z_n$ which does not fulfill, the random variables

$$\nu_0 = \begin{cases} 1 \ , & \text{if } \kappa = 0 \ , \\ 0 \ , & \text{if } \kappa \geq 1 \ , \end{cases}$$

$$\nu_1 = \begin{cases} 0 \,, & \text{if } \kappa = 0 \,, \\ -\dfrac{2}{k^* + 1}\dbinom{\kappa}{1} + \dbinom{\kappa}{1} \,, & \text{if } \kappa = 1 \,, \\ -\dfrac{2}{k^* + 1}\dbinom{\kappa}{1} + \dfrac{2}{k^*(k^* + 1)}\dbinom{\kappa}{2} + \sum\limits_{j=1}^{\kappa} (-1)^{j-1}\dbinom{\kappa}{j} \,, & \text{if } \kappa \geq 2 \,, \end{cases}$$

$$\nu_2 = \begin{cases} 0 \,, & \text{if } \kappa \leq 1 \,, \\ -\dfrac{2}{n}\dbinom{\kappa}{2} + \sum\limits_{j=2}^{\kappa} (-1)^{j}\dbinom{\kappa}{j} \,, & \text{if } \kappa \geq 2 \,, \end{cases}$$

have the required expected value. After some elementary calculations one get

$$\nu_1 = \begin{cases} 0, & \text{if } \kappa = 0 \,, \\ \dfrac{1}{k^*(k^* + 1)}(\kappa - k^*)(\kappa - k^* - 1) \,, & \text{if } \kappa \geq 1 \,, \end{cases}$$

$$\nu_2 = \begin{cases} 0, & \text{if } \kappa = 0 \,, \\ \dfrac{1}{n}(\kappa - 1)(n - \kappa) \,, & \text{if } \kappa \geq 1 \,. \end{cases}$$

These formulas are more comfortable for the simulation procedure and it is also evident that instead of ν_1 and ν_2 one could simulate the $k^*(k^* + 1)\nu_1$ and $n\nu_2$ random variables.

When simulating the random variables ν_0, $k^*(k^* + 1)\nu_1$ and $n\nu_2$ their covariance matrix can be estimated, too. By the aid of these estimates a final and more efficient estimation can be constructed for the multivariate probability distribution function.

In the following we give a step by step description of our algorithm. In this description $\bar{S}_1$ and $\bar{S}_2$ have been eliminated and $F_i(x_i)$ resp. $F_{ij}(x_i, x_j)$ denote the one- resp. two dimensional marginal distribution functions.

Algorithm for the calculation of the multivariate probability distribution function

Step 1 Initialization

Let $N_0 = 0$, $N_1 = 0$, $N_2 = 0$; $c_{11} = 0$, $c_{22} = 0$, $c_{12} = 0$; $s = 0$.

Let further k be the largest integer smaller than or equal to

$$\frac{n^2 + (1 - 2n) \sum_{i=1}^{n} F_i(x_i) + 2 \sum_{1 \le i < j \le n} F_{ij}(x_i, x_j)}{n - \sum_{i=1}^{n} F_i(x_i)} .$$

Let

$$P_l = \left(\frac{2}{n} - 1\right) \sum_{i=1}^{n} F_i(x_i) + \frac{2}{n} \sum_{1 \le i < j \le n} F_{ij}(x_i, x_j) \ ,$$

$$P_u = \frac{k^*(k^* + 1) - 2nk^* + n(n-1)}{k^*(k^* + 1)} + \frac{2k^* - 2n + 2}{k^*(k^* + 1)} \sum_{i=1}^{n} F_i(x_i) +$$

$$+ \frac{2}{k^*(k^* + 1)} \sum_{1 < i < j < n} F_{ij}(x_i, x_j) \ .$$

If $| P_u - P_l | < 0.0005$ then let the estimation of the distribution function value equal to $P = (P_l + P_u)/2$ with variance zero and Stop.

Step 2 Generation of a new random vector

Let $s = s + 1$, if $s > S$ then go to Step 6. Generate the random numbers $x_1^{(s)}, \ldots, x_n^{(s)}$.

Step 3 Initialization of the cycle for checking the inequalities

Let $k^{(s)} = 0$, $i = 0$.

Step 4 The cycle for testing the inequalities

Let $i = i + 1$, if $i > n$ then go to Step 5. If $x_i^{(s)} < z_i$ then repeat Step 4 else let $k^{(s)} = k^{(s)} + 1$ and also repeat Step 4.

Step 5 Update the frequency values and the cross products

If $k^{(s)} = 0$ then $\quad N_0 = N_0 + 1$ $\quad$ and go to Step 2.

If $k^{(s)} = 1$ then $\quad i_1^{(s)} = (k^{s)} - k^*)(k^{(s)} - k^* - 1)$,

$N_1 = N_1 + i_1^{(s)}$,

$$c_{11} = c_{11} + i_1^{(s)} i_1^{(s)} \qquad \text{and go to Step 2.}$$

If $k^{(s)} \geq 2$ then

$$\begin{aligned}
i_1^{(s)} &= (k^{(s)} - k^*)(k^{(s)} - k^* - 1), \\
i_2^{(s)} &= (k^{(s)} - 1)(n - k^{(s)}), \\
N_1 &= N_1 + i_1^{(s)}, \\
N_2 &= N_2 + i_2^{(s)}, \\
c_{11} &= c_{11} + i_1^{(s)} i_1^{(s)}, \\
c_{22} &= c_{22} + i_2^{(s)} i_2^{(s)}, \\
c_{12} &= c_{12} - i_1^{(s)} i_2^{(s)} \qquad \text{and go to Step 2.}
\end{aligned}$$

Step 6 Calculation of the relative frequencies and their covariance matrix

Let

$$\hat{N}_0 = N_0/S, \quad \hat{N}_1 = \frac{1}{k^*(k^*+1)} N_1/S, \quad \hat{N}_2 = \frac{1}{n} N_2/S \ ,$$

$$\hat{c}_{00} = \hat{N}_0(1 - \hat{N}_0), \ \hat{c}_{11} = \left[\frac{1}{k^*(k^*+1)}\right]^2 c_{11}/S - \hat{N}_1^2, \ \hat{c}_{22} = \left[\frac{1}{n}\right]^2 c_{22}/S - \hat{N}_2^2 \ ,$$

$$\hat{c}_{01} = \hat{N}_0 \hat{N}_1, \quad \hat{c}_{02} = - \hat{N}_0 \hat{N}_2, \quad \hat{c}_{12} = \frac{1}{nk^*(k^*+1)} c_{12}/S + \hat{N}_1 \hat{N}_2 \ .$$

Step 7 Calculation of the final estimation

Let

$$\hat{P}_0 = \hat{N}_0, \quad \hat{P}_1 = P_u - \hat{N}_1, \quad \hat{P}_2 = P_l + \hat{N}_2 \ ,$$

$$\lambda_0 = \hat{c}_{01}(\hat{c}_{22} - \hat{c}_{12}) + \hat{c}_{11}(\hat{c}_{02} - \hat{c}_{22}) + \hat{c}_{12}(\hat{c}_{12} - \hat{c}_{02}) \ ,$$

$$\lambda_1 = \hat{c}_{00}(\hat{c}_{12} - \hat{c}_{22}) + \hat{c}_{01}(\hat{c}_{22} - \hat{c}_{02}) + \hat{c}_{02}(\hat{c}_{02} - \hat{c}_{12}) \ ,$$

$$\lambda_2 = \hat{c}_{00}(\hat{c}_{12} - \hat{c}_{11}) + \hat{c}_{01}(\hat{c}_{01} - \hat{c}_{12}) + \hat{c}_{02}(\hat{c}_{11} - \hat{c}_{01}) \ ,$$

$$\lambda = \lambda_0 + \lambda_1 + \lambda_2 \ ,$$

$$w_0 = \lambda_0/\lambda, \quad w_1 = \lambda_1/\lambda, \quad w_2 = \lambda_2/\lambda \ ,$$

$$\hat{p} = w_0 \hat{p}_0 + w_1 \hat{p}_1 + w_2 \hat{p}_2 \ .$$

Let the empirical variance of the final estimation equal to

$$\frac{1}{S}(w_0^2 \hat{c}_{00} + w_1^2 \hat{c}_{11} + w_2^2 \hat{c}_{22} + 2 w_0 w_1 \hat{c}_{01} + 2 w_0 w_2 \hat{c}_{02} + 2 w_1 w_2 \hat{c}_{12}) \ .$$

Stop.

We remark that in the above described algorithm it can occur that one or more of the values of N_0, N_1 and N_2 remain zero at the beginning of Step 6. In order to avoid a final estimation with zero variance we make some additional investigations at the beginning of Step 6.

Additional investigations at the beginning of Step 6.

If $N_0 = 0$ then let the final estimation equal to zero with the empirical variance

$$\frac{1}{S}\left[\frac{1}{S}\left(1-\frac{1}{S}\right)\right] .$$

If $N_1 = 0$ then let the final estimation equal to P_u with the empirical variance

$$\frac{1}{S}\left[\frac{1}{S}\left(\frac{1}{k^*(k^*+1)}\right)^2-\frac{1}{S^2}\left(\frac{1}{k^*(k^*+1)}\right)^2\right] .$$

If $N_2 = 0$ then let the final estimation equal to $\max(P_l, 0)$ with the empirical variance

$$\frac{1}{S}\left[\frac{1}{S}\frac{1}{n^2}-\frac{1}{S^2}\frac{1}{n^2}\right] .$$

For the calculation of the gradient vector of the multivariate probability distribution functions we apply the formula

$$\frac{\partial\, F(z_1,\ldots,z_n)}{\partial\, z_l} = F(z_1,\ldots,z_{l-1},\, z_{l+1},\ldots,z_n \,|\, z_l) f(z_l) \quad ,$$

where $F(z_1,\ldots,z_{l-1},\, z_{l+1},\ldots,z_n \,|\, z_l)$ is the conditional probability distribution function of the random variables $x_1,\ldots,x_{l-1},\, x_{l+1},\ldots,x_n$ according to the condition $x_l = z_l$ and $f(z_l)$ is the marginal probability density function of the random variable x_l. The application of this formula was first proposed by A. Prékopa in [4]. In the case of the normal and Dirichlet distribution the conditional probability distributions are also normal resp. Dirichlet distributions. The conditional probability distributions of the multivariate gamma distribution have been determined in the original paper by A. Prékopa and T. Szántai [5]. So for the calculation of the gradient vector components we can use the same Monte Carlo simulation procedure that has been developed for the calculation of the multivariate probability distribution function values. But in the case of the multivariate gamma distribution the calculation of the one- and two dimensional marginal probability distribution functions of the conditional probability distribution requires one and two dimensional numerical integration which could be a time consuming job. So in the case of the gamma distribution we use the crude Monte Carlo simulation procedure. The necessary random vector generation is based on the construction described in [5].

For the calculation of the two dimensional gamma probability distribution function we gave a series expansion involving Laguerre polynomials (see [8]). The Laguerre polynomials can be calculated by well known recursive formulae.

For the calculation of the two dimensional Dirichlet probability distribution function H. Exton gave a formula by the aid of the Lauricella functions (see [1]). In our subroutine we use a direct series expansion which is numerically more stable.

Here we remark that in [7] the following theorem concerning the Dirichlet distribution has been proved.

THEOREM

(i) *If the sum of the two smallest argumentum values of the multivariate Dirichlet distribution function is greater than one then*

$$F(z_1, \ldots, z_n) = 1 - n + \sum_{i=1}^{n} F_i(z_i) \ .$$

(ii) *If the sum of the three smallest argumentum values of the multivariate Dirichlet distribution function is greater than one then*

$$F(z_1, \ldots, z_n) = \frac{1}{2}(n-1)(n-2) - (n-2)\sum_{i=1}^{n} F_i(z_i) + \sum_{1 \le i < j \le n} F_{ij}(z_i, z_j) \ .$$

This theorem gives a good chance to calculate the multivariate Dirichlet probability distribution function without any Monte Carlo simulation technique. The results of the above theorem are incorporated in our subroutine.

In the case of the multivariate gamma distribution we apply a fast heuristic algorithm for the solution of the fitting problem. This algorithm was published in [7]. If the heuristic algorithm fails then we use a dual type LP algorithm to find the best possible multivariate gamma distribution. In this algorithm we simply can get an initial dual feasible basis and its inverse. The application of the LP techniques for the solution of the fitting problem was described in [5]. We remark that the GGGML subroutine can take as input a given 0-1 construction matrix as well. In this case the fitting procedure becomes unnecessary.

3. DESCRIPTION OF THE SUBROUTINES

In this section a list of the subroutines is given describing their purpose, usage, arguments and the list of the required IMSL subroutines.

3.1. Subroutine MDMNOR

Purpose		–	to calculate the multivariate normal probability distribution function.
Usage		–	call mdmnor(x, r, n, nrnd, dseed, plo, p, pup, pvar, ier)
Arguments	x	–	input. Argument vector.
	r	–	input vector of length $n(n+1)/2$. It contains the correlation matrix elements. r is a positive definite matrix stored in symmetric storage mode.
	n	–	input. Size of vector x.
	nrnd	–	input. Number of trials. $= 0$, only the bounds will be calculated.
	dseed	–	input. Seed of the random number generation.
	plo	–	output. Lower bound of distribution function.
	p	–	output. Value of distribution function.
	pup	–	output. Upper bound of distribution function.
	pvar	–	output. Variance of the estimated value.
	ier	–	output. Error parameter.
Reqd. IMSL routines		–	mdnor, mdbnor, ggnsm, ggnml, ggubs, mdnris, merfi, uertst, ugetio

3.2. Subroutine MGMNOR

Purpose		–	to calculate the gradient vector of the multivariate normal probability distribution function.
Usage		–	call mgmnor(x, r, n, nrnd, dseed, g, gvar, ier)
Arguments	x	–	input. Argument vector.
	r	–	input vector of length $n(n+1)/2$. It contains the correlation matrix elements. r is a positive definite matrix stored in symmetric storage mode.
	n	–	input. Size of vector x, g and gvar.
	nrnd	–	input. Number of trials. = 0, only the bounds will be calculated.
	dseed	–	input. Seed of the random number generation.
	g	–	output. Gradient vector.
	gvar	–	output. Variance of the gradient vector.
	ier	–	output. Error parameter.
Reqd. IMSL routines		–	mdnor, mdbnor, mdmnor, ggnsm, ggnml, ggubs, mdnris, merfi, uertst, ugetio

Remarks If the user wishes to continue generating multivariate gamma deviate vectors distributed with the same sigma, then multiple calls may be made to gggml with iw nonzero on input. If iw is set to 0 on input the calculation of the mg transformation matrix will be carried out.

3.4. Subroutine MDBGAM

Purpose		–	to calculate the bivariate gamma probability distribution function.
Usage		–	call mdbgam (x, y, a, b, c, p, ier)
Arguments	x	–	input. Upper limit of integration for the first variable.
	y	–	input. Upper limit of integration for the second variable.
	a	–	input. Parameter of the bivariate gamma distribution according to the first component only.
	b	–	input. Parameter of the bivariate gamma distribution according to both of the components.
	c	–	input. Parameter of the bivariate gamma distribution according to the second component only.
	p	–	output. Value of bivariate gamma distribution function.
	ier	–	output. Error parameter.
Reqd. IMSL routines		–	gamma, mdgam, uertst, ugetio.

3.5. Subroutine MDMGAM

Purpose		– to calculate the multivariate gamma probability distribution function.
Usage		– call mdmgam(x, sigma, iw, imat, mg, teta, n, nrnd, dseed, plo, p, pup, pvar, ier)
Arguments	x	– input. Argument vector.
	sigma	– input vector of length $n(n+1)/2$. Sigma contains the variance-covariance values. Sigma is a positive definite matrix stored in symmetric storage mode.
	iw	– input. Integer value. If it has zero value the mg transformation matrix will be calculated from the covariance matrix.
	imat	– input. Row dimension of matrices mg and rvec exactly as specified in the dimension statement in the calling program.
	mg	– output/input n by $n(n+1)/2$ matrix of 0 and 1 elements. On the first call it is an output matrix. After the first call it is an input matrix containing the transformation matrix required for the construction of the multivariate gamma random deviates.
	teta	– output/input vector of length $n(n+1)/2$. On the first call it is an output vector. After the first call it is an input vector containing the parameter values of the standard gamma distributed components in the construction of the multivariate gamma random deviates.
	n	– input. Size of vector x.
	nrnd	– input. Number of trials. $= 0$, only the bounds will be calculated.
	dseed	– input. Seed of the random number generation.
	plo	– output. Lower bound of distribution function.
	p	– output. Value of distribution function.
	pup	– output. Upper bound of distribution function.
	pvar	– output. Variance of the estimated value.
	ier	– output. Error parameter.
Reqd. IMSL routines		– dlgama, mdbgam, gggml, ggamr, ggubs, uertst, ugetio

3.6. Subroutine MGMGAM

Purpose	–	to calculate the gradient vector of the multivariate gamma probability distribution function.
Usage	–	call mgmgam(x, sigma, iw, imat, mg, teta, n, nrnd, dseed, g, gvar, ier)
Arguments x	–	input. Argument vector.
sigma	–	input vector of length $n(n+1)/2$. Sigma contains the variance-covariance values. Sigma is a positive definite matrix stored in symmetric storage mode.
iw	–	input. Integer value. If it has zero value the mg transformation matrix will be calculated from the covariance matrix.
imat	–	input. Row dimension of matrices mg and rvec exactly as specified in the dimension statement in the calling program.
mg	–	output/input n by $n(n+1)/2$ matrix of 0 and 1 elements. On the first call it is an output matrix. After the first call it is an input matrix containing the transformation matrix required for the construction of the multivariate gamma random deviates.
teta	–	output/input vector of length $n(n+1)/2$. On the first call it is an output vector. After the first call it is an input vector containing the parameter values of the standard gamma distributed components in the construction of the multivariate gamma random deviates.
n	–	input. Size of vector x, g and gvar.
nrnd	–	input. Number of trials. = 0, only the bounds will be calculated.
dseed	–	input. Seed of the random number generation.
g	–	output. Gradient vector.
gvar	–	output. Variance of the gradient vector.
ier	–	output. Error parameter.
Reqd. IMSL routines	–	dlgama, mdbgam, mdmgam, gggml, ggamr, ggubs, uertst, ugetio.

3.7. Subroutine MDBDIR

Purpose		–	to calculate the bivariate Dirichlet probability distribution function.
Usage		–	call mdbdir (x, y, a, b, c, p, ier)
Arguments	x	–	input. Upper limit of integration for the first variable.
	y	–	input. Upper limit of integration for the second variable.
	a	–	input. First parameter of the bivariate Dirichlet distribution.
	b	–	input. Second parameter of the bivariate Dirichlet distribution.
	c	–	input. Third parameter of the bivariate Dirichlet distribution.
	p	–	output. Value of bivariate Dirichlet distribution function.
	ier	–	output. Error parameter.
Reqd. IMSL routines		–	dlgama, uertst, ugetio.

3.8. Subroutine MDMDIR

Purpose		–	to calculate the Dirichlet probability distribution function.
Usage		–	call mdmdir (x, a, b, n, nrnd, dseed, plo, p, pup, pvar, ier)
Arguments	x	–	input. Argument vector.
	a	–	input. Parameter vector.
	b	–	input. Parameter value.
	n	–	input. Size of vector x and a.
	nrnd	–	input. Number of trials.
			= 0, only the bounds will be calculated.
	dseed	–	input. Seed of the random number generation.
	plo	–	output. Lower bound of distribution function.
	p	–	output. Value of distribution function.
	pup	–	output. Upper bound of distribution function.
	pvar	–	output. Variance of the estimated value.
	ier	–	output. Error parameter.
Reqd. IMSL routines		–	mdbeta, mdbdir, dlgama, ggamr, ggbtr, ggubs, ggubfs, uertst,ugetio.

3.9. Subroutine MGMDIR

Purpose		–	to calculate the gradient vector of the Dirichlet probability distribution function.
Usage		–	call mgmdir(x, a, b, n, nrnd, dseed, g, gvar, ier)
Arguments	x	–	input. Argument vector.
	a	–	input. Parameter vector.
	b	–	input. Parameter value.
	n	–	input. Size of vector x and a.
	nrnd	–	input. Number of trials. = 0, only the bounds will be calculated.
	dseed	–	input. Seed of the random number generation.
	g	–	output. Gradient vector.
	gvar	–	output. Variance of the gradient vector.
	ier	–	output. Error parameter.
Reqd. IMSL routines		–	mdbeta, mdbdir, mdmdir, dlgama, ggams, ggbtr, ggubs, ggubfs, uertst, ugetio.

4. SOME TEST RESULTS

This section contains some test results. These results have been produced by demo programs NORDEMO, GAMDEMO and DIRDEMO written in FORTRAN-77. They ask the input data from a user named data file and write the results to a user named output file. The codes and the data files are very similar so here only the program NORDEMO.F, the input data file N.DAT and the corresponding output data file N.RES are listed.

```
c This is a demo program for the calculation of the multivariate
c normal probability distribution function and its gradient vector
c
      program NORDEMO
      character*80 filnam,problem
      integer n,nrnd,ier
      real time(2)
      real x(50),r(2450),plo,p,pup,pvar
      real g(50),gvar(50),gstd(50)
      double precision dseed
      write(*,'(/a,\*(Do)') 'Enter name of  input data file : '
      read(*,*) filnam
      nin=7
      open(nin,file=filnam)
      write(*,'(a,\*(Do)')  'Enter name of output data file : '
      read(*,*) filnam
      nout=8
      open(nout,file=filnam,status='new')
      dseed=31925.0d0
   55 read(nin,*) problem
      read(nin,*) n
      if (n.eq.0) stop
      read(nin,*) (x(i),i=1,n)
      do 60 j=1,n
      kb=(j-1)*j/2+1
      ke=kb+j-1
   60 read(nin,*) (r(k),k=kb,ke)
      read(nin,*) nrnd
      dtim=dtime(time)
      call mdmnor (x,r,n,nrnd,dseed,plo,p,pup,pvar,ier)
      dtim=dtime(time)
      std=sqrt(pvar)
      write(nout,'(///a)')  problem
      write(nout,'(/a)')       '   Results for the distribution function'
      write(nout,'(a)')        '   -------------------------------------'
      write(nout,'(/a,i12)')   '   Error code      = ',ier
      write(nout,'(a,f12.6)')  '   Lower bound     = ',plo
      write(nout,'(a,f12.6)')  '   Estimated value = ',p
      write(nout,'(a,f12.6)')  '   Variance        = ',pvar
      write(nout,'(a,f12.6)')  '   Std. deviation  = ',std
      write(nout,'(a,f12.6)')  '   Upper bound     = ',pup
      write(nout,'(a,f12.6)')  '   User time       = ',time(1)
      write(nout,'(a,f12.6)')  '   System time     = ',time(2)
      write(nout,'(a,f12.6)')  '   Total time      = ',dtim
      write(nout,'(a)')        '   -------------------------------------'
      dtim=dtime(time)
      call mgmnor (x,r,n,nrnd,dseed,g,gvar,ier)
      dtim=dtime(time)
```

```
      do 70 i=1,n
   70 gstd(i)=sqrt(gvar(i))
      write(nout,'(/a)')        '   Results for the gradient vector'
      write(nout,'(a)')         '   --------------------------------'
      write(nout,'(/a,i12)')    '   Error code        = ',ier
      write(nout,'(a)')         '   Gradient vector = '
      write(nout,'(1x,5f12.6)') (g(i),i=1,n)
      write(nout,'(a)')         '   Variances         = '
      write(nout,'(1x,5f12.6)') (gvar(i),i=1,n)
      write(nout,'(a)')         '   Std. deviations = '
      write(nout,'(1x,5f12.6)') (gstd(i),i=1,n)
      write(nout,'(a,f12.6)')   '   User time         = ',time(1)
      write(nout,'(a,f12.6)')   '   System time       = ',time(2)
      write(nout,'(a,f12.6)')   '   Total time        = ',dtim
      write(nout,'(a)')         '   --------------------------------'
      go to 55
      end
```

The sample data file named N.DAT

```
   Problem-1
3
2.95029,3.934273,1.949334
1
0.360,1
0.125,0.571,1
10000
   Problem-2
3
2.662253,2.210704,6.5975
1
0.360,1
0.125,0.571,1
10000
   No more problems
0
```

The corresponding output file named N.RES

```
Problem-1

   Results for the distribution function
   -------------------------------------

   Error code        =           0
   Lower bound       =     0.972828
   Estimated value =     0.972849
   Variance          =     0.
   Std. deviation    =     0.
   Upper bound       =     0.972870
   User time         =     0.050000
   System time       =     0.016667
   Total time        =     0.066667
   -------------------------------------
```

```
   Results for the gradient vector
   -------------------------------

   Error code       =          0
   Gradient vector =
     0.004851     0.000058     0.059466
   Variances        =
     0.           0.           0.
   Std. deviations =
     0.           0.           0.
   User time        =     0.033333
   System time      =     0.016667
   Total time       =     0.050000
   -------------------------------

Problem-2

   Results for the distribution function
   -------------------------------------

   Error code       =          0
   Lower bound      =     0.982881
   Estimated value =     0.982954
   Variance         =     0.
   Std. deviation   =     0.
   Upper bound      =     0.983026
   User time        =     0.033333
   System time      =     0.016667
   Total time       =     0.050000
   -------------------------------------

   Results for the gradient vector
   -------------------------------

   Error code       =          0
   Gradient vector =
     0.010496     0.033860     0.000000
   Variances        =
     0.           0.           0.
   Std. deviations =
     0.           0.           0.
   User time        =     0.050000
   System time      =     0.
   Total time       =     0.050000
   -------------------------------
```

We remark that the above results correspond to the first two probability value published in Table 5 of [1]. One can see that in these examples the lower and upper bounds were close enough so the simulation was unnecessary. We were now able to calculate all of the probabilities contained in Table 5 in the case of the multivariate gamma probability distribution. The results are the following

Normal probability values	Gamma probability values
0.973	0.945
0.983	0.960
0.984	0.964
0.997	0.979
0.999	0.987

By the aid of the new subroutines for the calculation of the multivariate gamma probability distribution function and its gradient vector one can give the CALCON subroutine necessary to the nonlinear version of the MINOS system. So we were able to solve the problems of paper [1] also in that case when the random variables have multivariate gamma probability distribution. The results according to Table 6 of [1] are the following

x_0	x_1	x_2	x_3	x_4	Prob. lev.	CPU time	No. of major iterations
494.886	46.780	63.068	77.377	38.072	0.973	845.57	12
494.886	43.996	63.071	77.380	40.850	0.983	923.55	14
494.886	43.480	63.071	77.381	41.365	0.984	1020.17	15
494.886	38.100	60.597	84.334	42.266	0.997	1886.77	30
494.886	38.100	59.886	78.591	48.720	0.999	2502.17	40

Finally the three dimensional normal probability distribution function and its gradient vector have been calculated for different correlation matrices. Let us regard the correlation matrices

$$R_i = \begin{pmatrix} 1 & \rho_i & \rho_i \\ \rho_i & 1 & \rho_i \\ \rho_i & \rho_i & 1 \end{pmatrix}, \ i = 1,\ldots,6$$

and

$$R_i = \begin{pmatrix} 1 & -\rho_i & 0 \\ -\rho_i & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ i = 7,\ldots,11 \ ,$$

where $\rho_1 = 0.98$, $\rho_2 = 0.95$, $\rho_3 = 0.90$, $\rho_4 = 0.50$, $\rho_5 = 0.10$, $\rho_6 = 0.00$, $\rho_7 = 0.10$, $\rho_8 = 0.50$, $\rho_9 = 0.90$, $\rho_{10} = 0.95$, $\rho_{11} = 0.98$.

In the following tables of the distribution function estimations, their standard deviations and the gradient vector components are given for four different argumentum vectors. The gradient vectors have been normalized, i.e. they are given as unit vectors.

TABLE 1 The three dimensional normal probability distribution function and its gradient unit vector for the arguments $x_1 = 3.5$, $x_2 = 3.0$, $x_3 = 4.0$.

No. of the correlation matrix	Distribution function	Standard deviation	Gradient unit vector components		
1	0.998632	0.000000	0.003031	0.999995	0.000000
2	0.998618	0.000000	0.029867	0.999554	0.000111
3	0.998582	0.000000	0.074010	0.997255	0.002001
4	0.998411	0.000000	0.180770	0.983195	0.025477
5	0.998387	0.000000	0.192515	0.980851	0.029481
6	0.998386	0.000000	0.192909	0.980771	0.029578
7	0.998386	0.000000	0.193059	0.980741	0.029572
8	0.998386	0.000000	0.193117	0.980730	0.029570
9	0.998386	0.000000	0.193117	0.980730	0.029570
10	0.998386	0.000000	0.193117	0.980730	0.029570
11	0.998386	0.000000	0.193117	0.980730	0.029570

TABLE 2 The three dimensional normal probability distribution function and its gradient unit vector for the arguments $x_1 = 1.5$, $x_2 = 1.0$, $x_3 = 2.0$.

No. of the correlation matrix	Distribution function	Standard deviation	Gradient unit vector components		
1	0.835772	0.001482	0.004889	0.999988	0.000000
2	0.834285	0.001461	0.048220	0.998837	0.000283
3	0.833279	0.001556	0.121725	0.992550	0.005310
4	0.799433	0.001374	0.345737	0.933055	0.099366
5	0.773130	0.000500	0.417191	0.894594	0.160163
6	0.767516	0.000500	0.428256	0.887432	0.170475
7	0.764617	0.000500	0.437890	0.883424	0.166774
8	0.757781	0.000500	0.462005	0.873015	0.156195
9	0.756917	0.000500	0.466278	0.871121	0.154054
10	0.756917	0.000500	0.466278	0.871121	0.154054
11	0.756917	0.000500	0.466278	0.871121	0.154054

TABLE 3 The three dimensional normal probability distribution function and its gradient unit vector for the arguments $x_1 = 0.0$, $x_2 = 0.0$, $x_3 = 0.0$.

No. of the correlation matrix	Distribution function	Standard deviation	Gradient unit vector components		
1	0.454415	0.001600	0.577350	0.577350	0.577350
2	0.428312	0.001950	0.577350	0.577350	0.577350
3	0.396443	0.002376	0.577350	0.577350	0.577350
4	0.248065	0.003595	0.577350	0.577350	0.577350
5	0.146996	0.003932	0.577350	0.577350	0.577350
6	0.122978	0.003952	0.577350	0.577350	0.577350
7	0.115750	0.003950	0.589611	0.589611	0.552012
8	0.086492	0.003828	0.639602	0.639602	0.426402
9	0.041293	0.003119	0.692968	0.692968	0.198974
10	0.031258	0.002656	0.699991	0.699991	0.141514
11	0.021514	0.002179	0.704249	0.704249	0.089818

TABLE 4 The three dimensional normal probability distribution function and its gradient unit vector for the arguments $x_1 = - 0.5$, $x_2 = 0.0$, $x_3 = 0.5$.

No. of the correlation matrix	Distribution function	Standard deviation	Gradient unit vector components		
1	0.303910	0.003111	0.999977	0.006837	0.000000
2	0.302801	0.003152	0.997823	0.065943	0.000536
3	0.293886	0.003232	0.986416	0.163973	0.009839
4	0.199886	0.003747	0.876324	0.445402	0.183508
5	0.118743	0.003943	0.790285	0.526019	0.314251
6	0.100608	0.003931	0.769678	0.538188	0.343438
7	0.092843	0.003917	0.765605	0.556089	0.323442
8	0.056331	0.003684	0.750261	0.620106	0.229297
9	0.005333	0.001807	0.725009	0.683985	0.080786
10	0.001308	0.000973	0.718314	0.694022	0.048565
11	0.000000	0.000999	0.712645	0.701140	0.023240

REFERENCES

[1] Dupačová, J., A. Gaivoronski, Z. Kos and T. Szántai: Stochastic programming in water resources system planning: A case study and a comparison of solution techniques, WP-86-40, IIASA, Laxenburg.

[2] Exton, H.: Multiple Hypergeometric Functions and Applications (Ellis Horwood Limited, Chichester, 1976).

[3] IMSL Library Reference Manual (International Mathematical and Statistical Libraries, Houston, Texas, 1977).

[4] Prékopa, A.: On probabilistic constrained programming, in: Proceedings of the Princeton Symposium on Mathematical Programming (Princeton University Press, Princeton, N.J., 1970) 113–138.

[5] Prékopa, A. and T. Szántai: A new multivariate gamma distribution and its fitting to empirical streamflow data, Water Resources Research, 14 (1978) 19–24.

[6] Sathe, Y.S., M. Pradhan and S.P. Shah: Inequalities for the probability of occurrence of at least m out of n events, Journal of Applied Probability 17 (1980) 1127–1132.

[7] Szántai, T.: Numerical calculation of the probability values concerning to multivariate probability distributions, Dissertation in Hungarian, Budapest, 1986.

[8] Szántai, T.: Evaluation of a special multivariate gamma distribution function, Mathematical Programming Study 27 (1986) 1–16.