

Efficient Risk Simulations for Linear Asset Portfolios

Sak, Halis; Hörmann, Wolfgang; Leydold, Josef

Published: 01/12/2008

Document Version

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Sak, H., Hörmann, W., & Leydold, J. (2008). *Efficient Risk Simulations for Linear Asset Portfolios*. Research Report Series / Department of Statistics and Mathematics No. 80

Efficient Risk Simulations for Linear Asset Portfolios



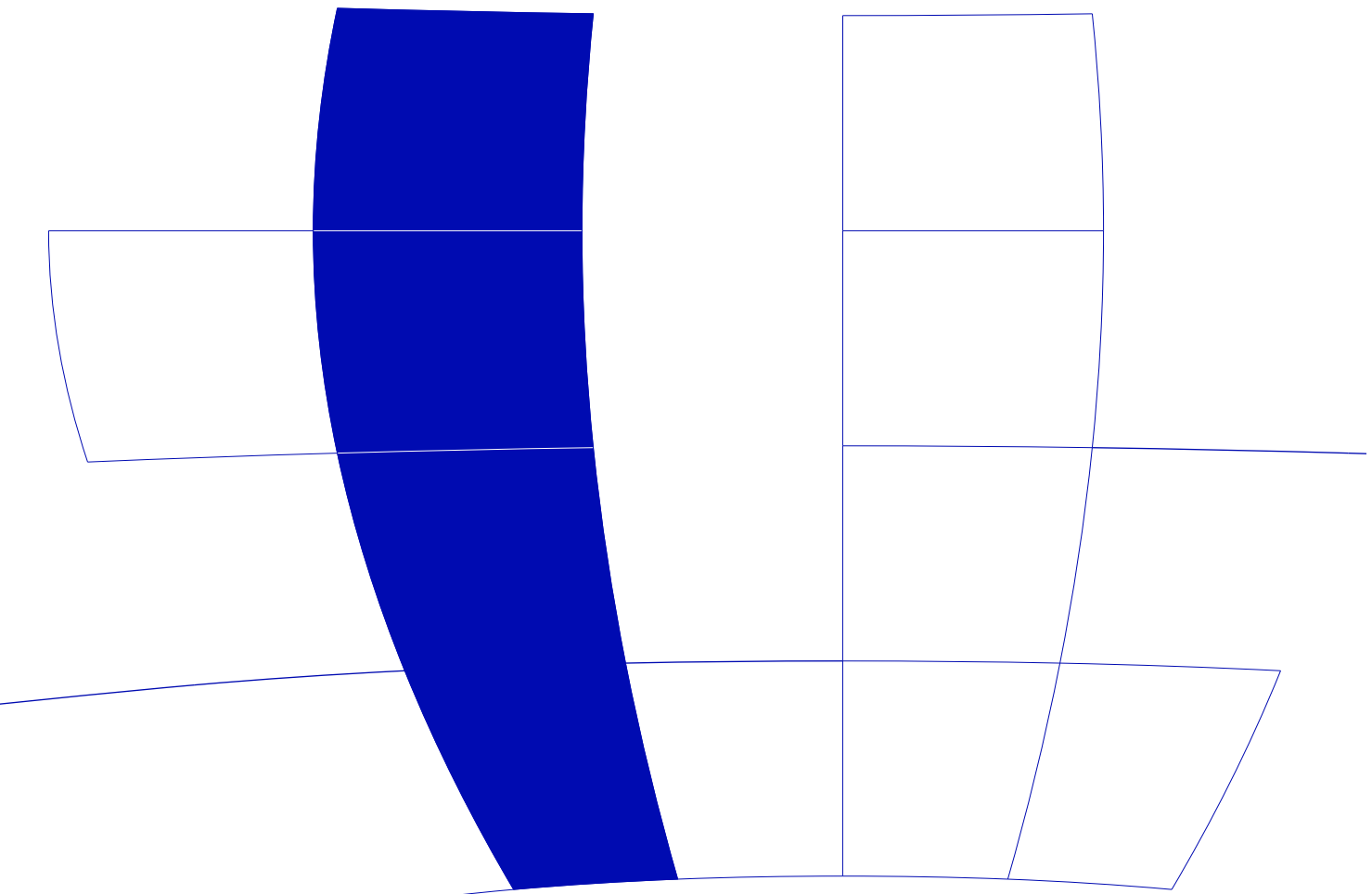
Halis Sak, Wolfgang Hörmann, Josef Leydold

Department of Statistics and Mathematics
Wirtschaftsuniversität Wien

Research Report Series

Report 80
December 2008

<http://statmath.wu-wien.ac.at/>



Efficient Risk Simulations for Linear Asset Portfolios

Halis Sak^{*,a}, Wolfgang Hörmann^b, Josef Leydold^a

^a*Department of Statistics and Mathematics, Vienna University of Economics and Business Administration, Augasse 2-6, A-1090 Wien, Austria*

^b*Department of Industrial Engineering, Boğaziçi University, 34342 Bebek-İstanbul, Turkey*

Abstract

We consider the problem of calculating tail probabilities of the returns of linear asset portfolios. As flexible and accurate model for the logarithmic returns we use the t -copula dependence structure and marginals following the generalized hyperbolic distribution. Exact calculation of the tail-loss probabilities is not possible and even simulation leads to challenging numerical problems. Applying a new numerical inversion method for the generation of the marginals and importance sampling with carefully selected mean shift we develop an efficient simulation algorithm. Numerical results for a variety of realistic portfolio examples show an impressive performance gain.

Key words: Risk management; importance sampling; linear asset portfolio; t -copula; generalized hyperbolic distribution

1. Introduction

Estimation of the profit/loss or of the return distribution lies at the heart of risk management since the computation of risk measures assumes the availability of either of these distributions especially of their tail behavior. It is a well-known fact that the returns of financial assets over short time horizons can not be described by the normal distribution due to semi-heavy tails and non-zero kurtosis. To remedy this modeling problem, several families of distributions have been successfully fitted to financial data; the most

*Corresponding Author: Department of Statistics and Mathematics, Vienna University of Economics and Business Administration, Augasse 2-6, A-1090 Wien, Austria. Phone: +43.1.31336-4180, Fax: +43.1.31336-774, Email: halis.sak@wu-wien.ac.at

Email addresses: halis.sak@wu-wien.ac.at (Halis Sak), hormannw@boun.edu.tr (Wolfgang Hörmann), josef.leydold@wu-wien.ac.at (Josef Leydold)

flexible and best fitting seems to be the generalized hyperbolic distribution (see Aas and Haff, 2006; Behr and Pötter, 2009; Prause, 1997). However, it has been seldom used in risk management and pricing implementations probably due to its unfriendly density function that makes the evaluation of the distribution function very expensive. Also random variate generation for Monte Carlo simulation is claimed to be slow (Behr and Pötter, 2009). This is certainly true if the inversion method has to be applied which is the case for copula generation.

A similar situation is true for the multivariate return distributions where, among others, Embrechts et al. (2002) and Frey and McNeil (2001) stress the necessity of using copulas to model the dependence structure of multivariate variables in finance. Furthermore, Mashal et al. (2003) conclude that the t -copula fits empirically better than the Gaussian copula without assuming specific marginal distributions for asset and equity returns. In a more recent work, Kole et al. (2007) apply goodness-of-fit tests to the t , Gaussian and Gumbel copula for the risk management of linear asset portfolios. The t -copula is preferred to Gaussian and Gumbel copulas because of capturing the dependence better in the non-extremes and extremes (tails).

Combining the conclusions of these recent papers the t -copula with generalized hyperbolic marginals seems to be currently the most flexible and realistic model for the joint distribution of logarithmic stock returns. According to many authors the main drawback of that model is the numerical difficulty it poses for probability calculations and simulation. It seems therefore worth-while to develop a simulation methodology that allows efficient simulation of tail-loss probabilities for the above model and that is the aim of our paper. To our knowledge we are the first to consider efficient risk simulations for stock portfolios when the logarithmic returns follow the t -copula with generalized hyperbolic marginal distributions.

There are recent papers proposing efficient simulation methods for market and credit portfolio risk computations using the t -copula dependence structure. Glasserman et al. (2002) develop an efficient method for simulating the value-at-risk of option portfolios using delta-gamma approximations; Kang and Shahabuddin (2005) and Bassamboo et al. (2008) use importance sampling techniques for the estimation of multi-factor portfolio credit risk. Our method is related to the methods explained there as we are also considering importance sampling for a t -copula model. On the other hand those references discuss different financial instruments and different models. Glasserman et al. (2002) use the t -copula to model the price change (not the logarithmic return) and the other two papers consider credit portfolios

instead of stock portfolios. This implied that we had to develop all details of the selection of the importance sampling distribution from scratch.

The paper is organized as follows: Section 2 gives the formal definition of the model we consider, Section 3 shortly describes fast numerical inversion necessary to generate random vectors from a t -copula with generalized hyperbolic marginals. In Section 4, we present the details of the importance sampling technique which is used to decrease the variance of the simulation of the tail-loss probabilities. In Section 5, we present our numerical results for a variety of realistic asset portfolios whereas Section 6 provides final comments.

2. The Model

We assume that the log-returns of d stocks over a short time horizon (typically a day) follow a t -copula with ν degrees of freedom and its dependence structure is described by the positive definite matrix Σ ; L denotes the (lower triangular) Cholesky factor of Σ satisfying $LL' = \Sigma$. To generate the random return vector from the t -copula it is well known that we start with a vector Z of d iid. standard normal variates that is then transformed into the correlated normal vector $\tilde{Z} = LZ$. We obtain the vector T from the multivariate t -distribution by generating a random variate Y from the chi-square distribution with ν degrees of freedom and calculating $T = \tilde{Z}/\sqrt{Y/\nu}$. The log-return vector $S = (S_1, S_2, \dots, S_d)'$ is then the result of the component-wise transform

$$S_j = G_j^{-1}(F_\nu(T_j))$$

where $F_\nu(\cdot)$ denotes the CDF of the t -distribution with ν degrees of freedom and G_j the CDF of the marginal distribution of the return of the j -th stock. Consider now a portfolio of these linear assets (stocks, equities, etc.) with weight-vector w . Then the portfolio return is a function of the random input vector Z and the random variate Y which also depends on the fixed parameters ν and Σ and on the CDFs G_j of the marginal distributions. We write for that return function

$$R(Z, Y) = \sum_{j=1}^d w_j e^{c_j G_j^{-1}(F_\nu(T_j))} \quad \text{with} \quad T = LZ/\sqrt{Y/\nu}, \quad (1)$$

where c_j denotes a scaling factor related to the yearly volatility σ_j and the variance var_j of the j -th marginal distribution by the formula

$$c_j = \sqrt{\frac{\sigma_j^2}{252} \frac{1}{var_j}}. \quad (2)$$

Finally we have to specify the marginal distribution. The logic of the simulation algorithm and the variance reduction described in the next section work for arbitrary fixed marginal distributions. The only requirement is that the calculation of the inverse of the CDF is feasible and not too slow. (Actually that is a quite stringent requirement as most inverse CDFs have no closed form and are numerically difficult to handle.) In our numerical experiments we first used marginals following the t -distribution but with different degrees of freedom for each stock. Note that even this setting is much more general than assuming a multi- t distribution where all marginals and the t -copula itself must have the same number of degrees of freedom.

As second family of marginals we considered the generalized hyperbolic distribution as it is reported to lead to very satisfactory fits for empirical stock data. The generalized hyperbolic distribution (see Barndorff-Nielsen and Blæsild, 1983; Prause, 1999) is described by the density function:

$$f_{GH}(x; \lambda, \alpha, \beta, \delta, \mu) = \kappa (\delta^2 + (x - \mu)^2)^{\frac{1}{2}(\lambda - \frac{1}{2})} K_{\lambda - \frac{1}{2}} \left(\alpha \sqrt{\delta^2 + (x - \mu)^2} \right) e^{\beta(x - \mu)} \quad (3)$$

where

$$\kappa = \frac{(\alpha^2 - \beta^2)^{\frac{\lambda}{2}}}{\sqrt{2\pi} \alpha^{\lambda - \frac{1}{2}} \delta^\lambda K_\lambda \left(\delta \sqrt{\alpha^2 - \beta^2} \right)}.$$

$K_\lambda(t)$ is the modified Bessel function of the third kind with index λ and is defined as

$$K_\lambda(t) = \frac{1}{2} \int_0^\infty \mu^{\lambda-1} e^{-\frac{1}{2}t(u+u^{-1})} du, \quad t > 0.$$

The domains of the parameters are $x, \mu \in \mathbb{R}$ and

$$\begin{aligned} \delta &\geq 0, |\beta| < \alpha, \text{ if } \lambda > 0 \\ \delta &> 0, |\beta| < \alpha, \text{ if } \lambda = 0 \\ \delta &> 0, |\beta| \leq \alpha, \text{ if } \lambda < 0. \end{aligned}$$

3. Fast inversion for the marginals

As it was already mentioned above the evaluation of the inverse functions of the CDFs $G^{-1}(\cdot)$ of the marginals is a difficult numerical task. Even for t -marginals where a specialized quantile algorithm is available, the evaluation of the inverse CDFs of the marginals requires about 80 percent of the total execution time for our C-implementation of the naive tail-loss probability simulation Algorithm 1. This comes from the fact that the quantile evaluation for the t -distribution is about 20 times slower than the inversion method for the exponential distribution.

But there exist very fast numerical inversion algorithms described in the literature (see Hörmann and Leydold, 2003). The algorithm HINV (Hermite inversion) requires the CDF of the distribution and a slow set-up to build a large table that holds for many subintervals the polynomial approximation of the inverse CDF constructed by Hermite interpolation; for a maximal error smaller than 10^{-10} typically much less than 1000 sub-intervals are required. The main advantage is that the sampling part of that inversion algorithm is faster than inversion for the exponential distribution. Using the R-implementation of that algorithm available in package Runuran (Leydold and Hörmann, 2008), leads for the inversion algorithm for the t -distribution to a speed up of more than 40 times. Of course such speed up factors are only reached for large sample sizes as the setup with the table calculations is slow. Note that for precise simulation results large samples are required anyway.

For the generalized hyperbolic distribution the situation is even more difficult. The R-package ghyp (Breymann and Luethi, 2008) includes a quite fast random variate generation algorithm but unfortunately it is not useful for copula generation as it is based on rejection. The quantile function available in package ghyp is really slow (about 5000 times slower than inversion for the exponential distribution) but also the implementation of the CDF is very slow; only the PDF of the generalized hyperbolic distribution is not too slow. This is a problem as many CDF evaluations are required in the set-up of HINV which makes the set-up for the generalized hyperbolic distribution very slow. A solution for that problem is the inversion algorithm PINV (see Derflinger et al., 2008) that requires the PDF instead of the CDF. In the set-up numeric integration and Newton-interpolation are combined to construct again a polynomial approximation of the inverse CDF for many sub-intervals. As PINV is included in Runuran as well we were able to use it for our simulations. Compared to the quantile function of the R-package ghyp, using PINV is about 10,000 times faster.

4. Simulating Tail loss probabilities with Importance Sampling (IS)

We use simulation to evaluate the tail-loss probability $P(R < x)$ (a simple risk measure) for example for $x = 0.98$. The result is the probability that the portfolio loses more than two percent of its value in a single day. If we estimate tail-loss probabilities for several values of x we can interpolate the results and invert them to calculate value-at-risk. In addition it is possible to use the IS-density presented in this paper for direct value-at-risk or expected short-fall (also called conditional value-at-risk) simulations. To be able to focus on the main point we only consider tail-loss probability simulation in this paper.

We start presenting the naive simulation algorithm to estimate $P(R < x)$ as Algorithm 1. We used R (R Development Core Team, 2008) as a convenient working environment and therefore mention some of the important R-functions below. For speed comparisons we have implemented our algorithms also in C.

Algorithm 1 Computation of $P(R < x)$ using naive simulation.

Initialization. Compute the lower triangular matrix L with $LL' = \Sigma$ using Cholesky decomposition; calculate the $c_j, j = 1, \dots, d$ using (2).

1. Repeat for replications $k = 1, \dots, n$:
 - (a) Generate independent standard normal variates Z_j .
 - (b) Generate Y from χ_ν^2 distribution.
 - (c) Calculate the t -distributed vector $T = LZ/\sqrt{Y/\nu}$.
 - (d) Calculate the total return of this repetition, $R^k = \sum_{j=1}^d w_j e^{c_j G_j^{-1}(F_\nu(T_j))}$
 2. Return $\frac{1}{n} \sum_{k=1}^n \mathbf{1}\{R^{(k)} < x\}$, where $\mathbf{1}\{\cdot\}$ denotes the indicator function.
-

A vector Z of iid. normal variates and a chi square random variate Y are generated in the beginning of the loop of Algorithm 1. An importance sampling strategy is best applied directly to these variates. We change their distribution in order to increase the frequency of very low return values R . To obtain easy IS-densities and simple likelihood ratios we only change the mean values of the normal variates by adding a mean shift vector with negative entries to the normal vector Z . As the chi-square distribution is a special case of the gamma distribution with shape parameter $\frac{\nu}{2}$ and scale parameter 2, a natural choice for the IS density for Y is the gamma distribu-

tion. As we divide all components of the vector LZ by $\sqrt{Y/\nu}$ (see formula (1) above) a decrease of Y will result in an increase in the dependence of asset returns. Thus as IS density we use a gamma distribution with the same shape parameter but a smaller scale parameter as this increases the probability of very low returns.

It is well known that a main practical problem in the application of importance sampling is the choice of the parameters of the IS-distribution. We utilize the general idea (see eg. Glasserman et al., 1999) to select the parameters such that the mode of the resulting IS density is equal to the mode of the zero-variance IS function which is for our problem defined by

$$f_0(z, y) = \mathbf{1}\{R(z, y) < x\} f_N(z) f_Y(y), \quad (4)$$

where $f_N(z)$ denotes the density of iid. standard normal variates and $f_Y(y)$ the density of the chi-square distribution with ν degrees of freedom. We therefore want to use the mode μ of $f_0(z, y)$ as mean shift vector (ie. mean vector of the IS distribution for the normal vector Z). Finding the mode μ of $f_0(z, y)$ requires the solution of the multidimensional optimization problem;

$$\max_{z, y} \left(\mathbf{1}\{R(z, y) < x\} e^{-z^T z/2} y^{\nu/2-1} e^{-y/2} \right). \quad (5)$$

Figure 1 shows the zero-variance IS function for the normal copula with t -marginals with 5 and 12 degrees of freedom. As we can see the presence of the indicator function implies that the zero-variance IS function is not continuous. Instead it is zero in a large part of the plane and proportional to the two-dimensional density in the region satisfying $R(Z, Y) \leq x$. Due to that discontinuous objective function our optimization problem is not solvable directly with standard “in the box” algorithms as we experienced even with the derivative free Nelder-Mead method. Therefore it is necessary to reformulate the optimization problem by the following considerations:

It is obvious that for the case $Y = \nu$ the vectors T and Z coincide. So for the moment we consider $Y = \nu$ fixed and only discuss the optimization with respect to the Z -vector which consists of iid. normal random variates and has a unimodal density radially symmetric around the origin. Thus for any line through the origin in the direction z_d the objective function is maximal for the unique point satisfying $R(\alpha z_d, \nu) = x$ and the value of α can be easily found using a one-dimensional numerical root finding algorithm. Therefore the objective function only depends on the direction of z_d . We define:

$$z_d^1 = z_d / \|z_d\| \quad \text{and} \quad r_0 \text{ unique length that satisfies } R(r_0 z_d^1, \nu) = x.$$

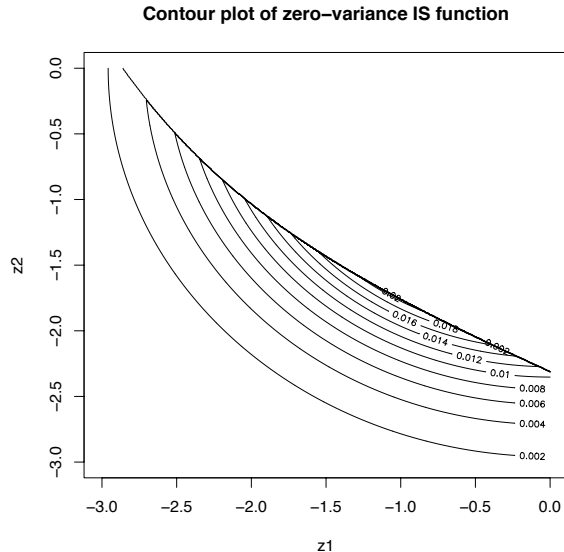


Figure 1: The zero-variance IS function for $d = 2$, normal copula with t -marginals with 5 and 12 degrees of freedom; equal weights, volatilities 0.1 and 0.25 and $x = 0.98$.

Note that for any given direction vector z_d , $r_0(z_d)$ is the distance from the origin to the unique z -value where the objective function is discontinuous. To calculate $r_0(z_d)$ we have to use a derivative-free numerical root finding procedure. (We use the R-function `uniroot()` that implements Brent's algorithm. To be sure that r_0 is finally on the correct side of the discontinuity it is necessary to replace x by $x - \Delta$ for a positive Δ much smaller than x .) That r_0 is the distance to the maximum for the given direction is clear for any fixed value of Y as $f_N(z)$ is radially symmetric.

So far we have assumed $Y = \nu$. To find the optimal value for Y we consider the vector $T = Z/\sqrt{Y/\nu}$. It is easy to see that to reach the optimal value in the direction z_d it is necessary that we have

$$T = Z/\sqrt{y/\nu} = r_0 z_d^1 \quad \text{and therefore} \quad \|z\| = r_0 \sqrt{y/\nu}.$$

Plugging the formula of $\|z\|$ into the log-density formula of the normal distribution and adding the log-density of the chi-square distribution we get the log-density of the joint distribution of (z, y) :

$$\log f(z, y) = C - 0.5(r_0^2 y/\nu) + (\nu/2 - 1) \log(y) - y/2. \quad (6)$$

The one-dimensional optimization with respect to y given r_0 is easily done in closed form. So for the direction z_d we obtain the maximum

$$\mu(z_d) = (z_0, y_0) \quad \text{with} \quad y_0 = \frac{\nu - 2}{1 + r_0^2/\nu} \quad \text{and} \quad z_0 = r_0 \sqrt{y_0/\nu} z_d^1.$$

To obtain the simplified objective function of the optimization problem we plug that result into the log-density (6) with constant $C = 0$ and get

$$o_f(z_d) = (\nu/2 - 1) \left(\log \left(\frac{\nu - 2}{1 + r_0^2/\nu} \right) - 1 \right).$$

We summarize the details of the function that returns the value of the objective function for a given z -direction as Algorithm 2.

Algorithm 2 Computing the z and y values and the objective function for a given direction z_d :

1. Set $z_d^1 = z_d/||z_d||$. Calculate r_0 by solving $(R(z = r_0 z_d^1, y = \nu) - (x - \Delta) = 0)$ numerically. (Use eg. $\Delta = 10^{-5}$).
 2. Return the starting mean shift vector $\mu_0 = (z_0 = r_0 \sqrt{y_0/\nu} z_d^1, y_0 = \frac{\nu-2}{1+r_0^2/\nu})$ and the objective function value: $(\nu/2 - 1)(\log(y_0) - 1)$.
-

Before we can start with the optimization we need a sensible starting direction z_d . To calculate it we consider the linear approximation of the return function for the case of the multinormal distribution.

$$\sum_{j=1}^d w_j e^{c_j(LZ)_j} \approx \sum_{j=1}^d w_j (1 + c_j(LZ)_j) = 1 + \sum_{j=1}^d v_j(LZ)_j = 1 + v' LZ = R_a(Z), \quad (7)$$

where we use the notation $v_j = c_j w_j$. Maximizing the objective function (5) with R replaced by R_a is very easy: We just have to find the point of the hyperplane $R_a(Z) = x$ that is closest to the origin. Clearly the vector from the origin to that point is perpendicular to the hyperplane, and has the direction

$$z_d = L'v.$$

z_d is useful as simple starting direction for the optimization algorithm.

The objective function described in Algorithm 2 requires as input only the direction, thus our maximization is just $d - 1$ dimensional. To reach that dimension reduction in practice we can therefore fix, without loss of generality, the first component of the vector z_d to one; (to avoid numerical problems

we could sort the assets according to the vector of the $c_j \cdot w_j$ in descending order as this guarantees together with the Cholesky factor that the first component of the z -vector cannot be close to zero. Our experiments show that in practice the sorting is not necessary.) As the objective function includes a numeric root finding algorithm it is not possible to find a closed form expression for its derivatives. Using a derivative-free optimization method like the Nelder-Mead method (the default choice of the R-function `optim()`) is possible but slow in higher dimensions. Using a quasi-Newton method with the gradients approximated by a finite difference method leads to much faster and closer to optimal results in higher dimensions. As the global quasi-Newton method leads to problems if one of the suggested direction vectors has negative components we used the constrained version suggested in (Byrd et al., 1995) (called method “L-BFGS-B” in the R-function `optim()`). As constraints we just used the non-negativity of all components of the direction vector. Algorithm 3 collects all the details of the optimization algorithm that calculates the mean shift vector μ and the mode of Y . Note that in Algorithm 3 we have non-negativity constraints for all components of z_d . However, the final mean shift vector μ for Z is negative because of getting a minus value for r_0 in Algorithm 2. In our experiments (described in Section 5 below) Algorithm 3 required for $d = 20$ between 4500 and 12,000 evaluations of the return function $R(Z, Y)$.

Algorithm 3 Computation of the mean shift vector μ and y .

1. Compute the (lower triangular) Cholesky factor L of the correlation matrix. Compute the $c_j, j = 1, \dots, d$ as in (2).
 2. Compute $v = c * w$ (component-wise product) and the direction vector $z_d = L'v$.
 3. Call an optimization algorithm (we used a quasi-Newton method with constraints) with starting direction z_d , with objective function as given in Algorithm 2 and with non-negativity constraints for all components of z_d . It returns the optimal direction $optz_d$.
 4. Call Algorithm 2 a last time with direction $optz_d$ to get the optimal vector (z_0, y_0) . Return the optimal mean shift $\mu = z_0$ and y_0 , the optimal mode for Y .
-

Remark: We have discussed the objective function without considering the marginals. If x is small and (some) marginal distributions are heavy tailed it is possible that the the zero-variance IS function $f_0(z, y)$ has local maxima. As all derivative based algorithms find the nearest hill from the starting values, we are aware of the possibility of not finding the global

maximum. However, having more than one local maxima is unlikely for real stock data and our experiments show that our Algorithm 3 always leads to acceptable and in most cases to very good results.

Algorithm 3 returns the optimal mean-shift μ and the optimal mode y_0 of the IS-density for Y . To calculate the optimal scale parameter θ of the gamma IS-density for Y is easy now. For $\theta = 1$, Y has the mode $\nu/2 - 1$. Thus we have to use the scale parameter

$$\theta = \frac{y_0}{\nu/2 - 1} \quad (8)$$

to obtain the required mode. Then the likelihood ratio is

$$w_{\mu,\theta}(Z, Y) = \exp(-\mu'Z + \mu'\mu/2 - Y/2 + Y/\theta + \log(\theta/2)\nu/2), \quad (9)$$

where $\exp(-\mu'Z + \mu'\mu/2)$ accounts for the mean shift we have added to the normal vector and the term $\exp(-Y/2 + Y/\theta + \log(\theta/2)\nu/2)$ relates the density of the χ_ν^2 distribution to that of the gamma distribution with shape parameter $\nu/2$ and scale parameter θ . The final IS algorithm is presented as Algorithm 4.

Algorithm 4 Computation of $P(R < x)$ using importance sampling

Initialization. Compute the lower triangular matrix L where $LL' = \Sigma$, $c_j, j = 1, \dots, d$ as in (2) and μ and y_0 using Algorithm 3. Then, compute θ using (8).

1. Repeat for replications $k = 1, \dots, n$:
 - (a) generate $Z_j \sim N(\mu_j, 1), j = 1, \dots, d$, independently;
 - (b) generate y from the gamma distribution with shape parameter $\nu/2$ and scale parameter θ ;
 - (c) calculate $w_{\mu,\theta}^{(k)}$ as in (9);
 - (d) Calculate the t -distributed vector $T = LZ/\sqrt{Y/\nu}$.
 - (e) Calculate the total return of this repetition, $R^{(k)} = \sum_{j=1}^d w_j e^{c_j G_j^{-1}(F_\nu(T_j))}$
 2. Return $\frac{1}{n} \sum_{k=1}^n w_{\mu,\theta}^{(k)} \mathbf{1}\{R^{(k)} < x\}$.
-

5. Numerical Experience

We started our experiments for log-returns following the t -copula with t -marginals using portfolio sizes $d = 2, 5, 10, 15, 20$ and degrees of freedom

of the t -copula equal to 5, 10, 15, 20. For the marginals we used the t -distribution and selected random degrees-of-freedom between 5 and 50, different for each stock. The off-diagonal elements of Σ were chosen randomly between 0 and 0.3 and volatilities randomly between 0.05 and 0.2. To choose the portfolio weights for the assets, we first generated independent variates uniformly distributed in $(1/1000, 1)$ and then normalized their sum to 1. The lower bound on the weights is necessary as we do not want to decrease the dimension of the problem. The range of the parameter values for the marginals was selected according to the values obtained in (Karadağ, 2008) when fitting that model to NYSE data. We are therefore confident that our variance reduction results are similar to what will be obtained for real world stock portfolios.

To compare the performance of our IS strategy given in Algorithm 4 with naive simulation, we did 10 simulations for the same portfolio size d and the same degrees of freedom ν of the t -copula but with different randomly selected values for the degrees of freedom of the marginals and for the volatilities. For each of the ten randomly selected parameter sets we selected one x -value such that $P(R < x) \approx 0.05$ and the second such that $P(R < x) \approx 0.001$ in Table 1. As the variance reduction factor of IS strongly depends on the probability of the rare event that selection is necessary to make the results within a column comparable.

We report the average and the worst result of the ten different randomly selected problems by reporting the average and minimal variance reduction factors (VR) in Table 1. As the execution time may be longer for the IS algorithm we also calculated the improvement factor (I) which is the variance reduction factor multiplied by the execution time of naive over the execution time of the IS simulation. We included the time necessary to find the mean shift in our improvement factors. The sample size for the simulations is 10^5 . Due to the setup smaller (larger) sample sizes would lead to smaller (larger) improvement factors.

The aim of this paper was to develop an efficient method for simulating the risk of medium-sized stock portfolios. The assumptions of log-returns following the t -copula with marginals following the generalized hyperbolic distribution are of special interest as many recent paper suggest that these distributions have a very good fit when used for real stock-data. Therefore our second numerical example includes portfolios having log-returns with generalized hyperbolic marginals. We use the parameters estimated for German stocks from January 1988 to May 1994 given in (Prause, 1997). Out of these stocks we randomly sample with replacement d parameter values for the marginal distributions of our portfolio. The random selection of

Table 1: Average and minimum values of the variance reduction factors (AVR and MVR) and of the improvement factors (AI and MI) of importance sampling for a t -copula model of dimension d and degrees of freedom ν with t -marginals. The average and minimum are taken over ten randomly selected sets of values for Σ and for the degrees of freedoms of the marginals. $n = 100,000$ is used for both naive and the IS simulation.

d	ν	$P(R < x \approx 0.05)$				$P(R < x \approx 0.001)$			
		AVR	MVR	AI	MI	AVR	MVR	AI	MI
2	5	7	7	6	5	247	200	223	170
	10	9	8	8	7	280	217	243	176
	15	9	8	8	7	273	225	239	189
	20	9	8	8	7	278	232	251	214
5	5	7	7	6	5	205	166	183	153
	10	9	8	8	6	234	201	204	158
	15	9	8	8	7	285	222	240	184
	20	9	8	8	8	268	208	227	167
10	5	7	7	6	5	190	141	166	126
	10	9	8	7	7	254	219	212	190
	15	9	9	8	7	258	216	212	191
	20	9	9	8	7	278	239	238	202
15	5	7	7	6	5	190	151	156	123
	10	9	8	7	6	262	220	218	184
	15	9	8	8	7	269	235	218	184
	20	9	8	8	7	267	223	214	177
20	5	7	7	6	5	189	153	154	123
	10	9	8	7	6	257	212	208	164
	15	9	8	7	6	266	234	210	187
	20	9	9	8	7	268	206	208	170

Σ , the volatilities and the weights and the two values selected for x were done as for the first example.

We use the R-packages Runuran (Leydold and Hörmann, 2008) and ghyp (Breymann and Luethi, 2008) to generate random variates from the t -copula with generalized hyperbolic marginal distributions.

Average and minimum values of the variance reduction and the improvement factors for 10 simulations for the d and ν pairs are given in Table 2. The required time for finding the optimal mean shift corresponds to 700 – 12,000 replications of the naive simulation.

5.1. Interpreting the results

Looking at the results of Tables 1 and 2 we probably first notice the clear difference in the variance reduction between small (0.001) and large (0.05) tail probabilities. This is a well known situation for importance sampling: Its efficiency rapidly increases with the rarity of the event. But can we be satisfied with the reached variance reduction results? As easy benchmark for a comparison we can consider the simulation problem $P(Z < x)$ for a standard normal variate. Here importance sampling with a simple mean shift leads to a variance reduction-factor of about 9.4 for probability 0.05 and to a variance reduction of about 290 for probability 0.001. So we may say that our results for the variance reduction are close to that benchmark. For the 0.05 probability case our results are very close to the bench mark but of course improvement factors around 7 or 8 are considerable but not really great. The improvements are very consistent and obviously little influenced by the special parameters of the stocks; there is practically no difference in the results for t and generalized hyperbolic marginals. Even the size of the portfolio d does not influence the variance reduction. This is really a good result as in many applications of IS it is known that the improvement deteriorates with increasing dimension. The robustness of the variance reduction in the 0.05 probability case may be explained by the fact that we are not far in the tails; thus the marginals and the t -copula are not too different from normal and also the return function is approximated quite well by a linear function.

For tail probability 0.001 the situation is different. Generally speaking the improvement factors are really impressive, all of their averages are above 90, most of them even above 200 which simply means that the precision of the result reached by the naive algorithm in three minutes is delivered on average in one or at most 2 seconds when using IS. Again the variance reduction is not reduced by the portfolio size d . It becomes clear from the

minimum variance reduction factor that the special parameters of the portfolio have an influence, in some unlucky parameter situations the variance reduction is clearly smaller than the average. 17 and 25 are the two smallest values for the variance reduction occurring in Table 2 for $P(R < x) \approx 0.001$. (Note that for that small tail-probability case a repetition of the simulation experiment may lead to quite different results for the two minimum columns MVR and MVI on the right hand side of Table 2). Here we also observe a difference in the results of the t -marginals and the generalized hyperbolic marginals. The latter show a much stronger fluctuation of the variance reduction factors. The minimum variance reduction is especially high for the case that the t -copula has $\nu = 20$ degrees of freedom, this is a t -copula close to the normal copula. For smaller values of ν the minimum variance reduction is larger which is of practical importance as smaller ν implies that we have stronger tail dependence. Mashal et al. (2003) estimate the number of degrees-of-freedom of the t -copula and obtain $\nu = 12$ for asset returns and $\nu = 13$ for equity returns. We can say that our IS approach is successful for these degrees of freedom which are of practical relevance for portfolios.

When we consider the difference between the variance reduction and the improvement factors we can see that the latter are between 10 and 20 percent smaller. This indicates that the extra computing afford necessary for the set-up and execution of IS is never more than 20 percent.

5.2. A practical example

We can run the tail-loss probability simulation algorithms for several values of x to obtain a figure that shows the general risk situation and can be also used to find the VaR for several α values. Therefore we simulated the tail-loss probabilities for generalized hyperbolic marginals with $d = 20$ and $\nu = 10$ for 16 x -values between 0.91 and 0.99. Figure 2 shows the results of these simulations based on 10,000 replications. The three curves show the sample mean (the center line) and the corresponding 95% confidence interval borders computed for each of the 16 x -values. The solid lines show the IS-results whereas the results of the naive algorithm are shown as dashed lines. That figure clearly shows that naive simulation should not be used for $\alpha < 0.01$ and must not be used for $\alpha < 0.001$.

5.3. Execution Times

We coded our algorithm in R and also in C. The timing results are of course more interesting for the C-implementation. As mentioned in Section 3 using fast table inversion speeds up the inversion method of the t -distribution

Table 2: Average and minimum values of the variance reduction factors (AVR and MVR) and of the improvement factors (AI and MI) of importance sampling for a t -copula model of dimension d and degrees of freedom ν with generalized hyperbolic marginals. The average and minimum are taken over ten randomly selected sets of values for Σ and the marginals from randomly selected German stocks. $n = 100,000$ is used for both naive and the IS simulation.

d	ν	$P(R < x \approx 0.05)$				$P(R < x \approx 0.001)$			
		AVR	MVR	AI	MI	AVR	MVR	AI	MI
2	5	8	8	8	7	248	195	238	187
	10	10	9	8	8	218	46	207	42
	15	9	8	9	7	196	34	187	33
	20	9	8	8	8	173	41	165	40
5	5	8	8	8	8	281	234	267	219
	10	9	9	9	8	215	133	201	127
	15	9	8	9	8	114	25	109	24
	20	8	8	8	8	90	27	84	26
10	5	9	8	8	7	290	242	271	224
	10	9	9	9	8	217	167	203	154
	15	9	9	8	8	161	77	150	72
	20	9	8	8	8	104	42	98	39
15	5	8	8	8	7	269	215	249	198
	10	9	9	8	8	270	231	246	211
	15	9	9	9	8	185	150	168	137
	20	9	8	8	7	129	17	118	15
20	5	8	8	8	7	267	197	245	185
	10	10	9	9	8	263	221	238	199
	15	9	9	8	8	199	141	178	126
	20	9	9	9	8	163	116	147	104

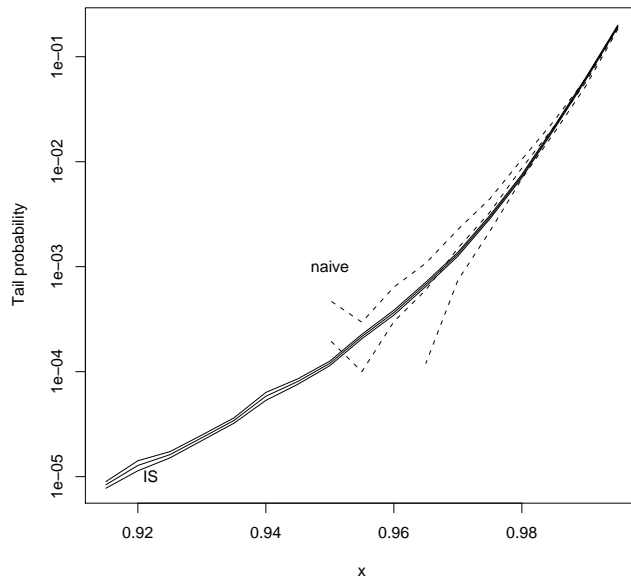


Figure 2: Comparison of the IS with naive simulation in estimating tail-loss probabilities for our second example for $d = 20$ and $\nu = 10$ using 10,000 replications.

by a factor of about 40. For the t -copula with t -marginals, $d = 20$ stocks and 100,000 repetitions our C-implementation of the naive algorithm takes about 10 seconds on our Linux PC when using the fast function `qt()` (of the C-library `Rmath` from the R-project) for calculating the t -quantiles. When using fast table inversion the execution time is reduced to three seconds which is not bad. Note that after the speed-up more than 70 percent of the remaining computing time is required for evaluating the CDF of the t -distribution. So here an improvement seems still possible.

One of the big advantages of fast table inversion is that its execution time is practically not influenced by the distribution that is generated. So for generalized hyperbolic marginals we get the same execution time of about 3 seconds as for t -marginals. Using the quantile function of the R-package `ghyp` instead leads to an execution time of more than 1000 seconds. So here the speed-up is really substantial.

As mentioned when discussing the improvement factors the IS algorithm is less than 20 percent slower than naive simulation as the weight calculations

are very simple.

6. Conclusions

The aim of this paper was to develop efficient simulation methods for estimating tail-loss probabilities of the returns of linear asset portfolios for realistic stock-return models. We therefore developed an IS strategy and solved all the non-trivial technical problems for multivariate log-returns following the t -copula and having t or generalized hyperbolic marginals. For the generation of the log-returns a new numerical inversion method has to be utilized to obtain good speed. We have demonstrated with several examples that the resulting algorithm is fast and reliable. Thus we have shown that for the t -copula model with generalized hyperbolic marginals precise risk estimates can be obtained in short time by combining clever importance sampling with fast table inversion methods.

References

- Aas, K., Haff, I. H., 2006. The generalized hyperbolic skew student's t -distribution. *Journal of Financial Econometrics* 4 (2), 275–309.
- Barndorff-Nielsen, O., Blæsild, P., 1983. Hyperbolic distributions. In: Johnson, N. L., Kotz, S., Read, C. B. (Eds.), *Encyclopedia of Statistical Sciences*. Vol. 3. Wiley, New York, pp. 700–707.
- Bassamboo, A., Juneja, S., Zeevi, A., 2008. Portfolio credit risk with extremal dependence: Asymptotic analysis and efficient simulation. *Operations Research* 56 (3), 593–606.
- Behr, A., Pötter, U., 2009. Alternatives to the normal model of stock returns: Gaussian mixture, generalised logf and generalised hyperbolic models. *Annals of Finance* 5 (1), 49–68.
- Breymann, W., Luethi, D., 2008. ghyp: A package on the generalized hyperbolic distribution and its special cases. R package version 1.3.0, <http://www.idp.zhaw.ch>.
- Byrd, R. H., Lu, P., Nocedal, J., Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing* 16, 1190–1208.

- Derflinger, G., Hörmann, W., Leydold, J., 2008. Random variate generation by numerical inversion when only the density function is known. Research Report Series of the Department of Statistics and Mathematics 78, Wirtschaftsuniversität Wien, Augasse 2–6, A-1090 Wien, Austria, <http://epub.wu-wien.ac.at/english/>.
- Embrechts, P., McNeil, A., Straumann, D., 2002. Correlation and Dependence in Risk Management: Properties and Pitfalls. ed. M.A.H. Dempster, Cambridge University Press, Cambridge, pp. 176–223.
- Frey, R., McNeil, A., 2001. Modelling dependent defaults. ETH, Eidgenössische Technische Hochschule Zürich, Department of Mathematics, <http://e-collection.ethbib.ethz.ch/view/eth:26296>.
- Glasserman, P., Heidelberger, P., Shahabuddin, P., 1999. Asymptotically optimal importance sampling and stratification for pricing path dependent options. *Mathematical Finance* 9 (2), 117–152.
- Glasserman, P., Heidelberger, P., Shahabuddin, P., 2002. Portfolio value-at-risk with heavy-tailed risk factors. *Mathematical Finance* 12 (3), 239–269.
- Hörmann, W., Leydold, J., 2003. Continuous random variate generation by fast numerical inversion. *ACM Trans. Model. Comput. Simul.* 13 (4), 347–362.
- Kang, W., Shahabuddin, P., 2005. Fast simulation for multifactor portfolio credit risk in the t-copula model. In: WSC '05: Proceedings of the 37th conference on Winter simulation M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds. Winter Simulation Conference, pp. 1859–1868.
- Karadağ, D. T., 2008. Portfolio risk calculation and stochastic portfolio optimization by a copula based approach. Master's thesis, Boğaziçi University Istanbul.
- Kole, E., Koedijk, K., Verbeek, M., 2007. Selecting copulas for risk management. *Journal of Banking and Finance* 31 (8), 2405–2423.
- Leydold, J., Hörmann, W., 2008. Runuran – R interface to the UNU.RAN random variate generators, Version 0.8. Department of Statistics and Mathematics, WU Wien, A-1090 Wien, Austria, <http://cran.r-project.org/>.

- Mashal, R., Naldi, M., Zeevi, A., 2003. Comparing the dependence structure of equity and asset returns. *Risk* 16, 82—87.
- Prause, K., 1997. Modelling financial data using generalized hyperbolic distributions. FDM preprint 48, University of Freiburg.
- Prause, K., 1999. The generalized hyperbolic model: Estimation, financial derivatives, and risk measures. Ph.D. thesis, University of Freiburg.
- R Development Core Team, 2008. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, <http://www.R-project.org>.