Nested Archimedean Copulas Meet R— The nacopula Package

Marius Hofert[*]
ETH Zurich

Martin Mächler
ETH Zurich

Abstract

The package **nacopula** provides procedures for constructing nested Archimedean copulas in any dimensions and with any kind of nesting structure, generating vectors of random variates from the constructed objects, computing function values and probabilities of falling into hypercubes, as well as evaluation of characteristics such as Kendall's tau and the tail-dependence coefficients. As a by-product, algorithms for various distributions, including exponentially tilted stable distributions, are implemented. Detailed examples are given.

Keywords: Archimedean copulas, nested Archimedean copulas, sampling algorithms, Kendall's tau, tail-dependence coefficients, exponentially tilted stable distribution, R.

1. Introduction

A *copula* is a multivariate distribution function with standard uniform univariate margins. Standard references for an introduction are Joe (1997) or Nelsen (2007).

Sklar (1959) shows that for any multivariate distribution function H with margins F_j, $j \in \{1, \dots, d\}$, there exists a copula C such that

$$H(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)), \ \boldsymbol{x} \in \mathbb{R}^d. \tag{1}$$

Conversely, given a copula C and arbitrary univariate distribution functions F_j, $j \in \{1, \dots, d\}$, H defined by (1) is a distribution function with marginals F_j, $j \in \{1, \dots, d\}$. On one hand, Sklar's Theorem tells us that we can decompose any given multivariate distribution function into its margins and a copula. By this decomposition, copulas allow us to study multivariate distributions functions independently of the margins. This is of particular interest in statistics. On the other hand, Sklar's Theorem provides a tool for constructing large classes of multivariate distributions and is therefore often used for sampling multivariate distributions via copulas. This is indispensable for many applications in the areas of statistics and finance. For sampling the multivariate distribution H it suffices to sample the common dependence structure, given by the copula C, and to transform the obtained variates to the correct margins F_j, $j \in \{1, \dots, d\}$. Since this transformation is usually easy to achieve (simply apply the *generalized inverse* $F_j^-(y) := \inf\{x \in \mathbb{R} : F_j(x) \geq y\}$ corresponding to F_j, $j \in \{1, \dots, d\}$, with

[*]The first author (Willis Research Fellow) thanks Willis Re for financial support while this paper was completed.

the convention that $\inf \emptyset := \infty$), sampling from H usually boils down to sampling the copula C under consideration.

Besides elliptical copulas, Archimedean copulas play an important role in practical applications. In contrast to elliptical ones, Archimedean copulas are given explicitly in terms of a generator and they are able to capture different kinds of tail dependencies, e.g., upper, but not lower tail dependence. With the algorithm of Marshall and Olkin (1988), they are also usually easy to sample. Their functional symmetry (in u_j, $j = 1, \dots, d$), also referred to as *exchangeability*, however, is often considered to be a drawback, e.g., in risk-management applications where the considered portfolios are typically high-dimensional. To circumvent exchangeability, Archimedean copulas can be nested into each other under certain conditions. The resulting copulas are referred to as *"nested Archimedean copulas"* and allow to model hierarchical dependence structures.

2. Archimedean copulas

2.1. Archimedean copulas and their properties

An *Archimedean generator*, or simply *generator*, is a continuous, decreasing function $\psi : [0, \infty] \to [0, 1]$ which satisfies $\psi(0) = 1$, $\psi(\infty) := \lim_{t\to\infty} \psi(t) = 0$, and which is strictly decreasing on $[0, \inf\{t : \psi(t) = 0\}]$. A d-dimensional copula is called *Archimedean* if it is of the form

$$C(\boldsymbol{u}; \psi) := \psi(\psi^{-1}(u_1) + \dots + \psi^{-1}(u_d)),\ \boldsymbol{u} \in [0, 1]^d, \tag{2}$$

for some generator ψ with inverse $\psi^{-1} : [0, 1] \to [0, \infty]$, where $\psi^{-1}(0) := \inf\{t : \psi(t) = 0\}$. A necessary and sufficient condition for an Archimedean generator ψ to generate a proper copula in all dimensions d is that ψ is *completely monotone*, i.e., $(-1)^k\psi^{(k)}(t) \geq 0$ for all $t \in (0, \infty)$ and $k \in \mathbb{N}_0$, see Kimberling (1974) in the context of t-norms or Hofert (2010b, p. 54) for a rework in terms of copulas. Recall that the most simple dependence model, namely, independence, is provided by $\psi(t) = \exp(-t)$, with $\psi^{-1}(t) = \log(t)$, and corresponding *independence* copula $C(\boldsymbol{u}) = \prod_{j=1}^{d} u_j$.

The class of all completely monotone Archimedean generators is denoted by Ψ_∞ in what follows. By Bernstein's Theorem, see, e.g., Feller (1971, p. 439), this class coincides with the class of Laplace-Stieltjes transforms of distribution functions F on the positive real line, where the Laplace-Stieltjes transform of F, also known as the *Laplace transform of the distribution*, is defined as

$$\mathcal{LS}[F](t) := \int_0^\infty \exp(-tx)\, dF(x),\ t \in [0, \infty).$$

For a $\psi \in \Psi_\infty$, we hence have the relation

$$\psi = \mathcal{LS}[F],\ \text{or, equivalently,}\ F = \mathcal{LS}^{-1}[\psi].$$

for a distribution function F on the positive real line.

Note that the distribution function F is known for virtually all commonly used Archimedean generators, see, e.g., Hofert (2010b, p. 62). The package **nacopula** currently provides the most widely used families of Ali-Mikhail-Haq, Clayton, Frank, Gumbel, and Joe, see Table 1 for the generators and their corresponding distribution functions. Except for Clayton's family, where we use a slightly simpler generator, these generators are the ones given in Nelsen (2007, pp. 116).

These Archimedean copula families are provided as `"acopula"` R objects, containing as slots the corresponding generator ψ, `psi`, its inverse ψ^{-1}, `psiInv`, and the "sampler", i.e., random number generator for $V \sim F$, as `V0`:

```
> require(nacopula)
> ls("package:nacopula", pattern = "^cop[A-Z]")

[1] "copAMH"     "copClayton" "copFrank"   "copGumbel"  "copJoe"

> copClayton

Archimedean copula ("acopula"), family "Clayton"
 It contains further slots, named
  "psi", "psiInv", "paraConstr", "paraInterval", "V0", "tau",
  "tauInv", "lambdaL", "lambdaLInv", "lambdaU", "lambdaUInv",
  "nestConstr", "V01"

> copClayton@psi

function (t, theta)
{
    (1 + t)^(-1/theta)
}
<environment: namespace:nacopula>

> copClayton@psiInv # the inverse of psi(), psi^{-1}

function (t, theta)
{
    t^(-theta) - 1
}
<environment: namespace:nacopula>

> copClayton@V0     # "sampler" for  V ~ F()

function (n, theta)
{
    rgamma(n, shape = 1/theta)
}
<environment: namespace:nacopula>
```

The majority of slots of such copula objects are functions, encoding properties of that copula family. In what follows, many of these functions are presented.

Sampling Archimedean copulas

From a mixture representation with respect to F, the following algorithm may be derived for sampling Archimedean copulas, see Marshall and Olkin (1988).

Algorithm 2.1 (Marshall and Olkin (1988))

(1) **sample** $V \sim F := \mathcal{LS}^{-1}[\psi]$

(2) **sample** $R_j \sim \mathrm{Exp}(1)$, $j \in \{1, \dots, d\}$

(3) **set** $U_j := \psi(R_j/V)$, $j \in \{1, \dots, d\}$

(4) **return** $\boldsymbol{U} = (U_1, \dots, U_d)^{\mathsf{T}}$

In order for this algorithm to be easily applied, we need to know how to sample the distribution functions $F = \mathcal{LS}^{-1}[\psi]$. For the families of Ali-Mikhail-Haq, Clayton, Frank, Gumbel, and Joe, see Table 1.

Family	ϑ	$\psi(t)$	$V \sim F = \mathcal{LS}^{-1}[\psi]$
Ali-Mikhail-Haq	$[0,1)$	$(1-\vartheta)/(\exp(t)-\vartheta)$	$\mathrm{Geo}(1-\vartheta)$
Clayton	$(0,\infty)$	$(1+t)^{-1/\vartheta}$	$\Gamma(1/\vartheta, 1)$
Frank	$(0,\infty)$	$-\log(1-(1-e^{-\vartheta})\exp(-t))/\vartheta$	$\mathrm{Log}(1-e^{-\vartheta})$
Gumbel	$[1,\infty)$	$\exp(-t^{1/\vartheta})$	$\mathrm{S}(1/\vartheta, 1, \cos^{\vartheta}(\pi/(2\vartheta)), \mathbf{1}_{\{\vartheta=1\}}; 1)$
Joe	$[1,\infty)$	$1-(1-\exp(-t))^{1/\vartheta}$	$\binom{1/\vartheta}{k}(-1)^{k-1},\ k \in \mathbb{N}$

Table 1 Commonly used one-parameter Archimedean generators.

For the family of Ali-Mikhail-Haq, $\mathrm{Geo}(p)$ denotes a *geometric distribution* with success probability $p \in (0,1]$ and mass function $p_k = p(1-p)^{k-1}$ at $k \in \mathbb{N}$ (which in R is `dgeom(`k`,`p`)`). For Clayton's family, $\Gamma(\alpha,\beta)$ denotes the *gamma distribution* with shape $\alpha \in (0,\infty)$, rate $\beta \in (0,\infty)$, and density $f(x) = \beta^{\alpha} x^{\alpha-1} \exp(-\beta x)/\Gamma(\alpha)$, $x \in (0,\infty)$ (provided in R by `[dpqr]gamma(., shape=`α`, rate=`β`)`). For the family of Frank, $\mathrm{Log}(p)$ denotes a *logarithmic distribution* with parameter $p \in (0,1)$ and mass function $p_k = p^k/(-k\log(1-p))$ at $k \in \mathbb{N}$. For sampling this distribution, we provide `rlog(., `p`)`, using the algorithm "LK" of Kemp (1981). For Gumbel's family, F corresponds to a $S(\alpha,\beta,\gamma,\delta;1)$, i.e., a *stable*, distribution with characteristic function

$$\phi(t) = \exp\bigl(i\delta t - \gamma^{\alpha}|t|^{\alpha}(1 - i\beta\,\mathrm{sgn}(t) w(t,\alpha))\bigr),\ t \in \mathbb{R}, \tag{3}$$

where

$$w(t,\alpha) = \begin{cases} \tan(\alpha\pi/2), & \alpha \neq 1, \\ -2\log(|t|)/\pi, & \alpha = 1, \end{cases}$$

see, e.g., Nolan (2009, p. 8) for this "1-parameterization". For sampling from $S(.)$, we provide `rstable1(., `α`,`β`,`γ`, `δ`, 1)`, having implemented an algorithm for sampling stable distributions according to the ideas presented in Chambers, Mallows, and Stuck (1976), and improving on previous implementations in R. For the family of Joe, the mass function of V is of the form $p_k = \binom{\alpha}{k}(-1)^{k-1}$ at $k \in \mathbb{N}$ (take $\alpha := 1/\vartheta$), with $\alpha \in (0,1]$. This distribution can be sampled via the R function `rFJoe(., `α`)` which we implemented based on an algorithm presented in Hofert (2010a).

The rank-correlation coefficient Kendall's tau

It is often desirable to measure the degree of dependence between random variables by a real number, a generalized correlation. Such measures are referred to as *measures of association* and are usually studied for pairs of random variables. One such measure is *Kendall's tau*, defined by

$$\tau := \mathbb{E}[\operatorname{sign}((X_1 - X_1')(X_2 - X_2'))],$$

where $(X_1, X_2)^\mathsf{T}$ is a vector of two continuously distributed random variables, $(X_1', X_2')^\mathsf{T}$ is an independent and identically distributed copy of $(X_1, X_2)^\mathsf{T}$, and $\operatorname{sign}(x) := \mathbf{1}_{(0,\infty)}(x) - \mathbf{1}_{(-\infty,0)}(x)$ denotes the signum function (as R's `sign(x)`). Kendall's tau is a measure of concordance, see Scarsini (1984), and therefore, informally, measures the likelihood, as a number in $[-1, 1]$, with which large values of one variable are associated with large values of the other. If C is a bivariate Archimedean copula generated by a twice continuously differentiable generator ψ with $\psi(t) > 0$, $t \in [0, \infty)$, Kendall's tau can be represented in semi-closed form as

$$\tau = 1 + 4 \int_0^1 \frac{\psi^{-1}(t)}{(\psi^{-1}(t))'}\, dt = 1 - 4 \int_0^\infty t(\psi'(t))^2\, dt,$$

see Joe (1997, p. 91). For the Archimedean families of Ali-Mikhail-Haq, Clayton, and Gumbel, this integral can be evaluated explicitly, for Frank's family it involves the Debye function of order one, i.e., $D_1(\vartheta) := \frac{1}{\vartheta} \int_0^\vartheta t/(e^t - 1)\, dt$, and for Joe's family, it is given as a series, see Table 2.

Tail-dependence coefficients

Another notion of association is tail dependence. Tail dependence measures the probability that one random variable takes on values in its tail, given the other one takes on values in its tail. To be more precise, if $X_j \sim F_j$, $j \in \{1, 2\}$, are continuously distributed random variables, the *lower tail-dependence coefficient*, respectively the *upper tail-dependence coefficient*, of X_1 and X_2 are defined as

$$\lambda_l := \lim_{t \downarrow 0} \mathbb{P}(X_2 \le F_2^-(t) \mid X_1 \le F_1^-(t)), \quad \lambda_u := \lim_{t \uparrow 1} \mathbb{P}(X_2 > F_2^-(t) \mid X_1 > F_1^-(t)),$$

provided that the limits exist. These measures of association can be expressed in terms of the copula C corresponding to $(X_1, X_2)^\mathsf{T}$. If C is a bivariate Archimedean copula generated by ψ with $\psi(t) > 0$, $t \in [0, \infty)$, then

$$\lambda_l = \lim_{t \to \infty} \frac{\psi(2t)}{\psi(t)} = 2 \lim_{t \to \infty} \frac{\psi'(2t)}{\psi'(t)}, \quad \lambda_u = 2 - \lim_{t \downarrow 0} \frac{1 - \psi(2t)}{1 - \psi(t)} = 2 - 2 \lim_{t \downarrow 0} \frac{\psi'(2t)}{\psi'(t)},$$

where the equalities involving derivatives are obtained by l'Hôpital's rule and therefore the corresponding assumptions are required to hold. For the implemented Archimedean families, these limits can easily be found and are given in Table 2.

2.2. A three-dimensional Joe copula

As an example, we define a three-dimensional Joe copula with parameter chosen such that Kendall's tau (for the corresponding bivariate marginal copula of the Archimedean type) equals 0.5.

```
> (theta <- copJoe@tauInv(0.5))
```

Family	ϑ	τ	λ_l	λ_u
Ali-Mikhail-Haq	$[0,1)$	$1-2(\vartheta+(1-\vartheta)^2\log(1-\vartheta))/(3\vartheta^2)$	0	0
Clayton	$(0,\infty)$	$\vartheta/(\vartheta+2)$	$2^{-1/\vartheta}$	0
Frank	$(0,\infty)$	$1+4(D_1(\vartheta)-1)/\vartheta$	0	0
Gumbel	$[1,\infty)$	$(\vartheta-1)/\vartheta$	0	$2-2^{1/\vartheta}$
Joe	$[1,\infty)$	$1-4\sum_{k=1}^{\infty}1/(k(\vartheta k+2)(\vartheta(k-1)+2))$	0	$2-2^{1/\vartheta}$

Table 2 Kendall's tau and tail-dependence coefficients for commonly used one-parameter Archimedean generators.

```
[1] 2.856234
```

```
> C3joe.5 <- onacopula("Joe", C(theta, 1:3))
```

The internal structure of this object[1] is

```
> str(C3joe.5) # str[ucture] of object
```

```
Formal class 'outer_nacopula' [package "nacopula"] with 3 slots
  ..@ copula   :Formal class 'acopula' [package "nacopula"] with 15 slots
  .. .. ..@ name        : chr "Joe"
  .. .. ..@ psi         :function (t, theta)
  .. .. ..@ psiInv      :function (t, theta)
  .. .. ..@ theta       : num 2.86
  .. .. ..@ paraConstr  :function (theta)
  .. .. ..@ paraInterval:Formal class 'interval' [package "nacopula"] with 2 slots
  .. .. .. .. ..@ .Data: num [1:2] 1 Inf
  .. .. .. .. ..@ open : logi [1:2] FALSE TRUE
  .. .. ..@ V0          :function (n, theta)
  .. .. ..@ tau         :function (theta, noTerms = 446)
  .. .. ..@ tauInv      :function (tau, tol = .Machine$double.eps^0.25, ...)
  .. .. ..@ lambdaL     :function (theta)
  .. .. ..@ lambdaLInv  :function (lambda)
  .. .. ..@ lambdaU     :function (theta)
  .. .. ..@ lambdaUInv  :function (lambda)
  .. .. ..@ nestConstr  :function (theta0, theta1)
  .. .. ..@ V01         :function (V0, theta0, theta1, approx = 1e+05)
  ..@ comp     : int [1:3] 1 2 3
  ..@ childCops: list()
```

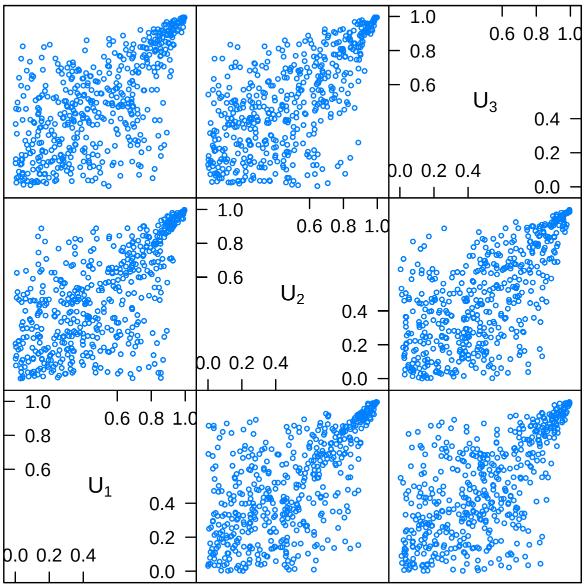
Let us sample 500 random variates (each in $[0,1]^3$) from this copula and visualize the generated data with a scatter-plot matrix.

```
> require(lattice)
> set.seed(1)
> dim(U3 <- rnacopula(500, C3joe.5))
```

[1]Note that we use a parametric *nested* Archimedean copula (see below), without any nesting, and corresponding `*nacopula()` functions for all this, since they generalize the present non-nested case.

```
[1] 500   3
```

```
> print(splom2(U3, cex = 0.4))
```



Scatter Plot Matrix

Let us compare the population and sample versions of Kendall's tau for the generated data.

```
> copJoe @ tau(theta) # ~= 0.5 because theta was chosen as such!

[1] 0.4999982

> round(cor(U3, method="kendall"), 3)

      [,1]  [,2]  [,3]
[1,] 1.000 0.467 0.476
[2,] 0.467 1.000 0.483
[3,] 0.476 0.483 1.000
```

Next, let us evaluate this Joe copula at $(0.5, 0.5, 0.5)^\mathsf{T}$ and $(0.99, 0.99, 0.99)^\mathsf{T}$.

```
> c(pnacopula(C3joe.5, c(.5, .5, .5)),
+   pnacopula(C3joe.5, c(.99,.99,.99)))

[1] 0.3009054 0.9853092
```

Now let us answer the question what the probability is for $\boldsymbol{U}$ to fall in the cube $(0.8, 1]^3$.

```
> prob(C3joe.5, c(.8, .8, .8), c(1, 1, 1))

[1] 0.1293357
```

Finally, the lower and upper tail-dependence coefficients for this copula can be obtained as follows.

```
> c(copJoe @ lambdaL(theta),
+   copJoe @ lambdaU(theta))

[1] 0.000000 0.725341
```

3. Nested Archimedean copulas

3.1. Construction

In contrast to elliptical copulas, Archimedean copulas are not restricted to radial symmetry, which implies that they can capture different tail dependencies, i.e., $\lambda_l \neq \lambda_u$. Further, they are given explicitly, which facilitates computing probabilities for such a dependence model. However, the exchangeability inherent in Archimedean copulas implies that all margins of the same dimension are equal. For modeling purposes, this becomes an increasingly strong assumption in the dimension. Asymmetries, i.e., more realistic dependencies, can be modeled by a hierarchical structure of Archimedean copulas, obtained by plugging in Archimedean copulas into each other. A d-dimensional copula C is called *nested Archimedean* if it is an Archimedean copula with arguments possibly replaced by other nested Archimedean copulas. If C is given recursively by (2) for $d = 2$ and, up to permutation of the arguments, by

$$C(\boldsymbol{u}; \psi_0, \dots, \psi_{d-2}) := \psi_0\big(\psi_0^{-1}(u_1) + \psi_0^{-1}(C(u_2, \dots, u_d; \psi_1, \dots, \psi_{d-2}))\big), \ \boldsymbol{u} \in [0,1]^d, \tag{4}$$

for $d \geq 3$, C is called *fully nested Archimedean copula* with $d-1$ *nesting levels* or *hierarchies*. Otherwise, C is called *partially nested Archimedean copula*. Fully and partially nested Archimedean copulas are summarized as *nested* (or *hierarchical*) *Archimedean copulas*.

Note that the structure of a nested Archimedean copula can be depicted by a tree, see Figure 1 for the three-dimensional case of Type (4) involving the generators ψ_0 and ψ_1. Due to this

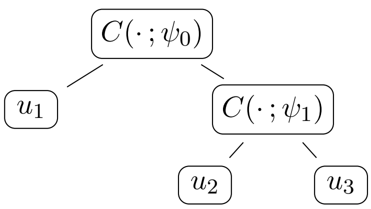


Figure 1 Tree structure of a three-dimensional fully nested Archimedean copula.

tree representation, we refer to the outermost Archimedean copula generated by ψ_0 as *root copula*. We refer to Archimedean copulas on an innermost nesting level, i.e., Archimedean copulas that contain at least one argument which is not a nested Archimedean copula, as *leaf copulas*. Further, we call an Archimedean copula appearing in the tree structure of a nested Archimedean copula *parent copula* if it is a nested Archimedean copula with at least one nested Archimedean copula as one of its components. We refer to these components as *child copulas*.

In R, because of the recursive tree structure, a powerful approach is to use a *recursive* class definition: In the **nacopula** package, we define the `nacopula` (**n**ested **a**rchimedean **copula**) class, with three components (`slots`),

```
setClass("nacopula",
         representation(copula    = "acopula",
                        comp      = "integer",  # from 1:d -- of length in [0,d]
                        childCops = "list" # of nacopulas, possibly empty
                       ),
         validity = function(object) {
            ......
             if(!all("nacopula" == sapply(object@childCops, class)))
                 return("All 'childCops' elements must be 'nacopula' objects")
            ......
         })
```

i.e., by its root copula (slot `@ copula`, a `"acopula"` object), a vector of indices of its "direct components" (slot `@comp = 1` for u_1 in the example), and a list of child copulas (slot `@ childCops`). The `outer_nacopula` is a version of the *same* class (i.e., it "contains" `nacopula` without any further slots), just with stricter validity checking, namely requiring that all components from all child copulas are exactly the set $\{1, 2, \ldots, d\}$.

```
setClass("outer_nacopula", contains = "nacopula",
         validity = function(object) {
             ## *Extra* checks in addition to those of "nacopula"
             .......
        })
```

For example, (one parametrization of) the three-dimensional example from Figure 1 is

```
> C3 <- onacopula("A", C(0.2, 1, C(0.8, 2:3)))
> C3

Nested Archimedean copula ("outer_nacopula"), with slot
'comp'   = (1)  and root
'copula' = Archimedean copula ("acopula"), family "AMH", theta= (0.2)
and 1 child copula
   Nested Archimedean copula ("nacopula"), with slot
   'comp'   = (2, 3)  and root
   'copula' = Archimedean copula ("acopula"), family "AMH", theta= (0.8)
   and *no* child copulas

> ## and the above is shortened from
> stopifnot(identical(C3,
+       onacopula("A", C(0.2, 1, list(C(0.8, 2:3, list())))))
+ ))
```

The recursive definition (4) of nested Archimedean copulas not only leads to recursive class definitions in R, but also to recursive functions for computing on such "nacopulas". All the following functions and methods (from our package `nacopula`) are defined *recursively*, typically using `lapply(x @ childCops, <fun>)`: The utilities `dim()`, `allComp()`, `printNacopula()`

(which is the hidden `show()` method), and the principal functions `pnacopula()` and `rncopula()` (via recursive utility `rnchild()`. As a simple example of these, `pnacopula(x, u)` simply evaluates the (recursive) formula (4), recursively applying itself to its child copulas:

```
pnacopula <- function(x,u) {
    stopifnot(is.numeric(u), 0 <= u, u <= 1, length(u) >= dim(x)) # can be larger
    C <- x@copula
    th <- C@theta
    ## use u[j] for the direct components 'comp':
    C@psi(sum(unlist(lapply(u[x@comp], C@psiInv, theta=th)),
              C@psiInv(unlist(lapply(x@childCops, pnacopula, u = u)),
                       theta=th)),
          theta=th)
}
```

In order for (4) being indeed a proper copula, Joe (1997, p. 88) and McNeil (2008) present the sufficient *nesting condition* that $\psi_i^{-1} \circ \psi_j$ is completely monotone for all nodes (with parent i, child j) appearing in a nested Archimedean copula. This condition can be derived from a mixture representation of C based on the distribution functions $F_0 := \mathcal{LS}^{-1}[\psi_0]$ and $F_{ij} := \mathcal{LS}^{-1}[\psi_{ij}(\cdot\,; V_0)]$ for

$$\psi_{ij}(t;x) := \exp\bigl(-x\psi_i^{-1}(\psi_j(t))\bigr),$$

$x \in (0,\infty)$.

3.2. Sampling

If the nesting condition is fulfilled for all nodes in the nested Archimedean structure, the following algorithm may be derived for sampling nested Archimedean copulas.

Algorithm 3.1
Let C be a nested Archimedean copula with root copula C_0 generated by ψ_0. Let $\boldsymbol{U}$ be a vector of the same dimension as C.

(1) **sample** $V_0 \sim F_0 = \mathcal{LS}^{-1}[\psi_0]$
(2) **for** all components u of C_0 that are nested Archimedean copulas **do** {
(3) **set** C_1 with generator ψ_1 to be the nested Archimedean copula u
(4) **sample** $V_{01} \sim F_{01} = \mathcal{LS}^{-1}[\psi_{01}(\cdot\,; V_0)]$
(5) **set** $C_0 := C_1$, $\psi_0 := \psi_1$, and $V_0 := V_{01}$ and **continue** with (2)
(6) }
(7) **for** all other components u of C_0 **do** {
(8) **sample** $R \sim \mathrm{Exp}(1)$
(9) **set** the component of $\boldsymbol{U}$ corresponding to u to $\psi_0(R/V_0)$
(10) }
(11) **return** $\boldsymbol{U}$

Note that for sampling nested Archimedean copulas when all generators involved belong to the same parametric family, it suffices to know how to sample

$$V_0 \sim F_0 = \mathcal{LS}^{-1}[\psi_0], \quad F_{01} = \mathcal{LS}^{-1}[\psi_{01}(\cdot\,; V_0)]$$

as all distribution functions F_{ij} take the same form as F_{01}, only the parameters may differ. In our R package **nacopula**, the supported Archimedean family objects therefore provide the three slots `V0`, `nestConstr` and `V01`, all functions. `nestConstr`, is a `function(`θ_0`, `θ_1`)` returning `TRUE` when the nesting condition is fulfilled, and `V0` and `V01` are random number generating functions, generating V from Table 1 and V_{01} from Table 3, respectively, e.g., for the most simple case, Ali-Mikhail-Haq,

```
> copAMH @ nestConstr

function (theta0, theta1)
{
    copAMH@paraConstr(theta0) && copAMH@paraConstr(theta1) &&
        theta1 >= theta0
}
<environment: namespace:nacopula>

> copAMH @ V01

function (V0, theta0, theta1)
{
    rnbinom(length(V0), V0, (1 - theta1)/(1 - theta0)) + V0
}
<environment: namespace:nacopula>
```

Sampling strategies for F_0 and F_{01} for many known Archimedean generators are presented in Hofert (2008), Hofert (2010a), and Hofert (2010b). For the families of Ali-Mikhail-Haq, Clayton, Frank, Gumbel, and Joe, Table 3 summarizes stochastic representations for V_0 and V_{01}.

Family	nesting condition	$V_{01} \sim F_{01} = \mathcal{LS}^{-1}[\psi_{01}(\cdot\,; V_0)]$
Ali-Mikhail-Haq	$\vartheta_0 \le \vartheta_1$	$V_0 + X,\ X \sim \mathrm{NB}(V_0, (1-\vartheta_1)/(1-\vartheta_0))$
Clayton	$\vartheta_0 \le \vartheta_1$	$\tilde{\mathrm{S}}(\vartheta_0/\vartheta_1, 1, (\cos(\pi\vartheta_0/(2\vartheta_1))V_0)^{\vartheta_1/\vartheta_0}, V_0\mathbf{1}_{\{\vartheta_0=\vartheta_1\}}, \mathbf{1}_{\{\vartheta_0\neq\vartheta_1\}}; 1)$
Frank	$\vartheta_0 \le \vartheta_1$	$\sum_{j=1}^{V_0} V_j$; for V_j, see Hofert (2010a)
Gumbel	$\vartheta_0 \le \vartheta_1$	$\mathrm{S}(\vartheta_0/\vartheta_1, 1, (\cos(\pi\vartheta_0/(2\vartheta_1))V_0)^{\vartheta_1/\vartheta_0}, \mathbf{1}_{\{\vartheta_0=\vartheta_1\}}; 1)$
Joe	$\vartheta_0 \le \vartheta_1$	$\sum_{j=1}^{V_0} V_j$; $V_j \sim \binom{\alpha}{k}(-1)^{k-1},\ k \in \mathbb{N}$

Table 3 Nesting conditions and stochastic representations for V_{01} (see also comments below).

First note that for nested Archimedean copulas based on generators belonging to the same Archimedean family, all implemented families indeed lead to proper copulas if the generators on a more nested (inner) level have larger parameter values than the ones on a lower (outer) level. This is equivalent to saying that Kendall's tau for a pair of random variables having an Archimedean copula which resides on a deeper nesting level as common margin has to be larger than or equal to the one for a pair of random variables having an Archimedean marginal copula residing on a lower nesting level. Slightly more informally, the inner (or lower) nested components u_j are *more* correlated than the outer ("higher up") ones.

Some comments on the distributions F_{01} of V_{01} for the different implemented families: For the family of Ali-Mikhail-Haq, V_{01} admits the stochastic representation $V_0 + X$, where X follows a negative binomial distribution with parameters as given in Table 3. The parameterization is $\mathrm{NB}(r,p)$, $r \in (0,\infty)$, $p \in (0,1)$, with mass function $p_k = \binom{k+r-1}{r-1} p^r (1-p)^k$, $k \in \mathbb{N}_0$, which in R is `dnbinom(`k`, size=`r`, prob=`p`)`.

For Clayton's family, F_{01} can be interpreted as a special case (take $h := 1$, $\alpha := \vartheta_0/\vartheta_1$) of the *exponentially tilted stable distribution*

$$\tilde{\mathrm{S}}(\alpha, 1, (\cos(\alpha\pi/2)V_0)^{1/\alpha}, V_0\mathbf{1}_{\{\alpha=1\}}, h\mathbf{1}_{\{\alpha\neq 1\}}; 1) \tag{5}$$

with $\alpha \in (0,1]$, $h \in [0,\infty)$, and $V_0 \in (0,\infty)$, and corresponding Laplace-Stieltjes transform

$$\psi(t) = \exp\bigl(-V_0((h+t)^\alpha - h^\alpha)\bigr),\ t \in [0,\infty].$$

Hofert (2010a) suggested a fast rejection algorithm for sampling this distribution. Devroye (2009) suggested an algorithm for sampling the exponentially tilted stable distribution

$$\tilde{\mathrm{S}}(\alpha, 1, \cos(\alpha\pi/2)^{1/\alpha}, \mathbf{1}_{\{\alpha=1\}}, \lambda\mathbf{1}_{\{\alpha\neq 1\}}; 1)$$

with $\alpha \in (0,1]$ and $\lambda \in [0,\infty)$, and corresponding Laplace-Stieltjes transform

$$\psi(t) = \exp\bigl(-((\lambda+t)^\alpha - \lambda^\alpha)\bigr),\ t \in [0,\infty].$$

One can easily check that by setting $\lambda := hV_0^{1/\alpha}$ and generating V_λ with the algorithm of Devroye (2009), the random variable V_{01} from F_{01} as given by (5) can be obtained via $V_{01} = V_0^{1/\alpha} V_\lambda$. Therefore, the distribution as given in (5) may be sampled by either the algorithm of Devroye (2009) or the one of Hofert (2010a). The former author reports that the complexity of his algorithm is bounded, the latter author shows that the complexity of his algorithm is $\mathcal{O}(V_0 h^\alpha)$. We implemented both algorithms for sampling (5) in the package **nacopula** and decide for each drawn V_0 which method is to be applied. As a simple rule, investigated by several parameter combinations, the default chooses the method of Hofert (2010a) if $V_0 h^\alpha < 4$ and the one of Devroye (2009) otherwise. As mentioned before, note that for sampling nested Clayton copulas, we have $h = 1$. Further, $\mathbb{E}[V_0] = 1/\vartheta_0$. Hence, in the mean, the algorithm of Hofert (2010a) is more often applied if $\vartheta_0 > 1/4$, equivalently, if Kendall's tau for the bivariate Archimedean copula generated by ψ_0 is greater than 1/9.

For Frank's family, Hofert (2010a) presented a rejection algorithm. For the family of Gumbel, $\psi_{01}(t; V_0) = \exp(-V_0 t^\alpha)$, $\alpha := \vartheta_0/\vartheta_1$, hence, see Table 1, F_{01} corresponds to the stable distribution as given in Table 3. Finally, for Joe's Archimedean family, V_{01} can be represented as a V_0-fold sum of independent and identically distributed random variables V_k, $k \in \{1, \dots, V_0\}$, from the same distributional class as V_0, see Table 1. For more information on the distributions of V_{01}, see Hofert (2010a) and Hofert (2010b), or the source code of the package **nacopula**.

3.3. A nine-dimensional nested Clayton copula

In this example, we consider a nine-dimensional partially nested Archimedean copula C of the form

$$C(\boldsymbol{u}) = C(u_3, u_6, u_1, C(u_9, u_2, u_7, u_5, C(u_8, u_4; \psi_2); \psi_1); \psi_0)$$

with tree structure depicted in Figure 2. Such a copula can be defined as follows, where we choose the parameters of ψ_0, ψ_1, and ψ_2 such that the corresponding Kendall's taus are 0.2, 0.5, and 0.8, respectively.

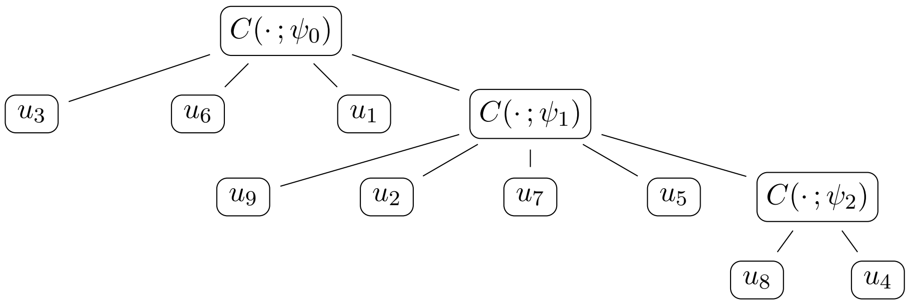


Figure 2 Tree structure of the nine-dimensional partially nested Archimedean copula C.

```
> theta0 <- copClayton@tauInv(0.2)
> theta1 <- copClayton@tauInv(0.5)
> theta2 <- copClayton@tauInv(0.8)
> c(theta0, theta1, theta2)

[1] 0.5 2.0 8.0

> C_9_clayton <- onacopula("Clayton",
+                          C(theta0, c(3,6,1),
+                            C(theta1, c(9,2,7,5),
+                              C(theta2, c(8,4)))))
> C_9_clayton # show(.) it

Nested Archimedean copula ("outer_nacopula"), with slot
'comp'   = (3, 6, 1)  and root
'copula' = Archimedean copula ("acopula"), family "Clayton", theta= (0.5)
and 1 child copula
   Nested Archimedean copula ("nacopula"), with slot
   'comp'   = (9, 2, 7, 5)  and root
   'copula' = Archimedean copula ("acopula"), family "Clayton", theta= (2)
   and 1 child copula
      Nested Archimedean copula ("nacopula"), with slot
      'comp'   = (8, 4)  and root
      'copula' = Archimedean copula ("acopula"), family "Clayton", theta= (8)
      and *no* child copulas
```

Let us sample 500 random variates (each in $[0,1]^9$) from this copula (this involves our efficient procedure for exponentially tilted stable distributions) and visualize the generated data with a scatter-plot matrix.

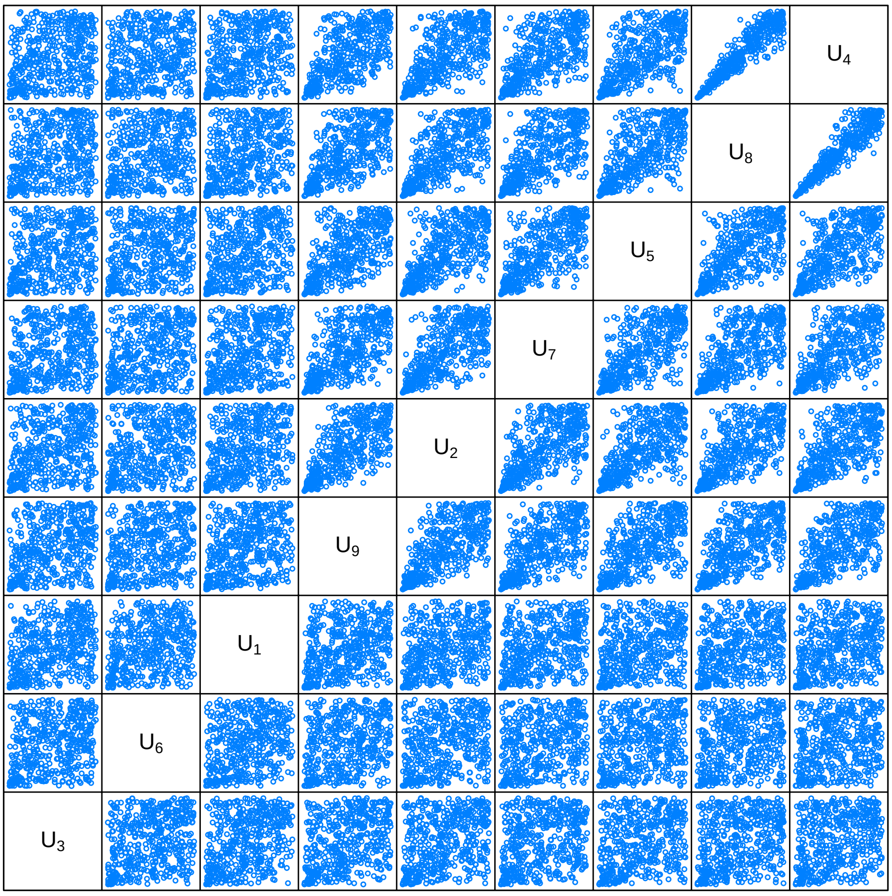
```
> set.seed(1)
> dim(U9 <- rnacopula(500, C_9_clayton))

[1] 500   9

> ## For plotting, re-order the columns according to the same strength
> ## of dependence:
> j <- allComp(C_9_clayton)
> (vnames <- do.call(expression,
+                    lapply(j, function(i) substitute( U[I], list(I=0+i)))))
```

```
expression(U[3], U[6], U[1], U[9], U[2], U[7], U[5], U[8], U[4])
```

```
> print(splom2(U9[, j], varnames= vnames,
+              cex = 0.4, pscales = 0))
```



Scatter Plot Matrix

The population version of Kendall's tau for $(U_4, U_5)^\mathsf{T}$ is 0.5. Let us check if the sample version of Kendall's tau is close to this value.

```
> round(cor(U9[,9],U9[,7], method="kendall"), 3)
```

```
[1] 0.5
```

Evaluating this nine-dimensional nested Clayton copula at $(0.5, \dots, 0.5)^\mathsf{T}$ and near the upper corner $(0.99, \dots, 0.99)^\mathsf{T}$ leads to the following results.

```
> c(pnacopula(C_9_clayton, rep(.5,9)),
+   pnacopula(C_9_clayton, rep(.99,9)))

[1] 0.09375995 0.91747302
```

The probability mass in the cube $(0.8, 1]^9$ can be determined as follows.

```
> prob(C_9_clayton, rep(.8,9), rep(1,9))

[1] 0.001061674
```

Finally, let us find the different lower and upper tail-dependence coefficients appearing on different levels for this nested Archimedean copula.

```
> c(copClayton @ lambdaL(theta0),
+   copClayton @ lambdaU(theta0))

[1] 0.25 0.00

> c(copClayton @ lambdaL(theta1),
+   copClayton @ lambdaU(theta1))

[1] 0.7071068 0.0000000

> c(copClayton @ lambdaL(theta2),
+   copClayton @ lambdaU(theta2))

[1] 0.917004 0.000000
```

4. Outer power Archimedean copulas

For an Archimedean generator $\psi \in \Psi_\infty$, $\psi(t^{1/\vartheta})$ is also a valid generator in Ψ_∞ for all $\vartheta \in [1, \infty)$, see, e.g., Feller (1971, p. 441). The resulting copulas are referred to as *outer power Archimedean copulas*. Note that two parametric Archimedean generators $\psi(t^{1/\vartheta_0})$, $\vartheta_0 \in [1, \infty)$, and $\psi(t^{1/\vartheta_1})$, $\vartheta_1 \in [1, \infty)$, of this type, constructed with the same "base" generator $\psi(t)$, $t \in [0, \infty)$, fulfill the nesting condition if $\vartheta_0 \leq \vartheta_1$. Therefore, one can build so-called *nested outer power Archimedean copulas*. Hofert (2010a) derives some results for these copulas, including instructions for sampling the corresponding random variables V_0 and V_{01}, as well as an explicit formula for Kendall's tau in terms of the Kendall's tau of the copula generated by the base generator ψ. Further, note that if the tail-dependence coefficients exist, they are greater than or equal to the ones corresponding to the base generator. For the Archimedean families implemented in the package **nacopula**, these can all be computed explicitly.

The goal of this section is to show how one might work with outer power Archimedean copulas with the package **nacopula**. For now, we use the outer power transformation `opower()` which generates an outer power family based on the provided copula family `copbase` with corresponding parameter `thetabase`.

```
> str(opower)
```

```
function (copbase, thetabase)
```

We believe that it is both more natural and flexible to work with copula families that are *generalizations* of our current families, each including the power as an extra parameter (such that `theta`, i.e., ϑ, will become 2-dimensional), rather than using the `opower()` construction below. This section therefore should be considered mainly as an outlook to further features of the package **nacopula**, where this transformation is not required anymore for working with outer power Archimedean copulas.

Using this transformation, we define a valid outer power Clayton copula with base generator of Clayton's type and with base parameter such that Kendall's tau equals 0.5.

```
> thetabase <- copClayton@tauInv(.5)
> (opow.Clayton <- opower(copClayton, thetabase))
```

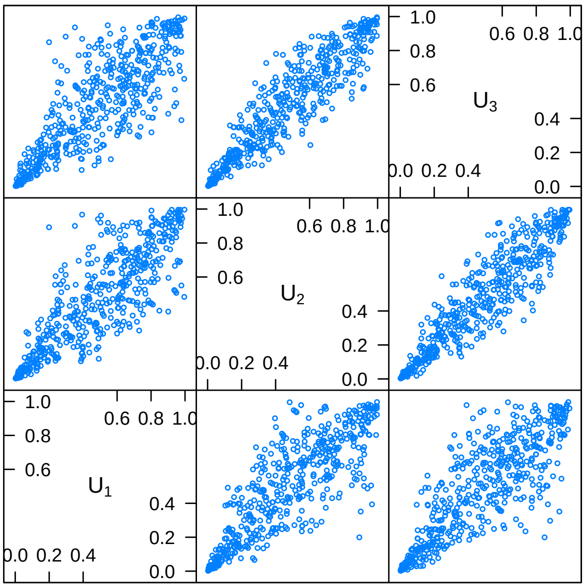
```
Archimedean copula ("acopula"), family "opower:Clayton"
 It contains further slots, named
  "psi", "psiInv", "paraConstr", "paraInterval", "V0", "tau",
  "tauInv", "lambdaL", "lambdaLInv", "lambdaU", "lambdaUInv",
  "nestConstr", "V01"
```

Based on this copula generator, we would like to define and sample a three-dimensional fully nested outer power Clayton copula with parameters such that Kendall's tau are 2/3 and 0.75.

```
> theta0 <- opow.Clayton@tauInv(2/3) # should be 1.5
> theta1 <- opow.Clayton@tauInv(.75) # should be 2
> ## Define a 3d fully nested Archimedean copula based on  opow.Clayton
> opC3 <- onacopula(opow.Clayton, C(theta0, 1, C(theta1, c(2,3))))
```

Now sample 500 random variates from this copula and visualize the generated data with a scatter-plot matrix. In contrast to Clayton copulas, note that this outer power Clayton copula has both lower and upper tail dependence.

```
> U3 <- rnacopula(500, opC3) ; stopifnot(dim(U3) == c(500,3))
> print( splom2(U3, cex = 0.4) )
```



Scatter Plot Matrix

Further, we can compare the population and sample versions of Kendall's tau for the generated data. The $(1, 2)$- and $(1, 3)$-entry of the matrix of pairwise sample versions of Kendall's tau should be close to $2/3$, the $(2, 3)$-entry should be close to 0.75.

```
> round(cor(U3, method="kendall"), 3)
```

```
      [,1]  [,2]  [,3]
[1,] 1.000 0.663 0.644
[2,] 0.663 1.000 0.764
[3,] 0.644 0.764 1.000
```

The different lower and upper tail-dependence coefficients for this copula can be obtained as follows.

```
> rbind(th.0 =
+       c(L = opow.Clayton @ lambdaL(theta0),
+         U = opow.Clayton @ lambdaU(theta0)),
+       th.1 =
+       c(L = opow.Clayton @ lambdaL(theta1),
+         U = opow.Clayton @ lambdaU(theta1)))
```

```
             L         U
th.0 0.7937005 0.4125989
th.1 0.8408964 0.5857864
```

5. Session Info

```
> toLatex(sessionInfo())
```

- R version 2.11.1 Patched (2010-07-01 r52421), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=de_CH.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=de_CH.UTF-8`, `LC_MONETARY=C`, `LC_MESSAGES=de_CH.UTF-8`, `LC_PAPER=de_CH.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=de_CH.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tools, utils
- Other packages: lattice 0.18-8, nacopula 0.4-2
- Loaded via a namespace (and not attached): grid 2.11.1, gsl 1.9-3

```
> my.strsplit(  packageDescription("nacopula")[["LastChanged"]]  )

$LastChangedRevision: 226 $
$LastChangedDate: 2010-07-02 09:41:53 +0200 (Fri, 02. Jul 2010) $
```

6. Conclusion

The package **nacopula** allows to easily construct and work with nested Archimedean copulas. First and foremost, fast sampling algorithms for these copulas are implemented. As a by-product, the package also provides related mathematical and random number generating functions, e.g., an efficient sampling algorithm for exponentially tilted stable distributions. Further features include the evaluation of nested Archimedean copulas, as well as computing probabilities of a random vector falling into a given hypercube. Concerning measures of association, Kendall's tau and the tail-dependence coefficients are implemented. Currently supported Archimedean families include the well-known families of Ali-Mikhail-Haq, Clayton, Frank, Gumbel, and Joe.

References

Chambers JM, Mallows CL, Stuck BW (1976). "A Method for Simulating Stable Random Variables." *Journal of the American Statistical Association*, **71**(354), 340–344.

Devroye L (2009). "Random variate generation for exponentially and polynomially tilted stable distributions." *ACM Transactions on Modeling and Computer Simulation*, **19**(18).

Feller W (1971). *An Introduction to Probability Theory and Its Applications*, volume 2. 2nd edition. Wiley.

Hofert M (2008). "Sampling Archimedean copulas." *Computational Statistics & Data Analysis*, **52**(12), 5163–5174.

Hofert M (2010a). "Efficiently sampling nested Archimedean copulas." *Computational Statistics & Data Analysis*. In press.

Hofert M (2010b). *Sampling Nested Archimedean Copulas with Applications to CDO Pricing*. Südwestdeutscher Verlag für Hochschulschriften AG & Co. KG.

Joe H (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall/CRC.

Kemp AW (1981). "Efficient Generation of Logarithmically Distributed Pseudo-Random Variables." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **30**(3), 249–253.

Kimberling CH (1974). "A probabilistic interpretation of complete monotonicity." *Aequationes Mathematicae*, **10**, 152–164.

Marshall AW, Olkin I (1988). "Families of Multivariate Distributions." *Journal of the American Statistical Association*, **83**, 834–841.

McNeil AJ (2008). "Sampling nested Archimedean copulas." *Journal of Statistical Computation and Simulation*, **78**, 567–581.

Nelsen RB (2007). *An Introduction to Copulas*. Springer.

Nolan JP (2009). "Stable Distributions—Models for Heavy Tailed Data." URL http://academic2.american.edu/~jpnolan/stable/chap1.pdf.

Scarsini M (1984). "On measures of concordance." *Stochastica*, **8**(3), 201–218.

Sklar A (1959). "Fonctions de répartition à n dimensions et leurs marges." *Publications de L'Institut de Statistique de L'Université de Paris*, **8**, 229–231.

Affiliation:

Marius Hofert
Department of Mathematics
ETH Zurich
8092 Zurich, Switzerland
E-mail: marius.hofert@math.ethz.ch
URL: http://www.math.ethz.ch/~hofertj/

Martin Mächler
Seminar für Statistik, HG G 16
ETH Zurich
8092 Zurich, Switzerland
E-mail: maechler@stat.math.ethz.ch
URL: http://stat.ethz.ch/people/maechler