Dynamic Copula Toolbox.

Manthos Vogiatzoglou
University of Macedonia, Egnatias 156, Thessaloniki, Greece
vogia@yahoo.com

Version: November 07, 2010

Copula functions have become the standard tool in modeling multivariate dependence over the last decade hence there are toolboxes available for simulating and estimating copulas in the major statistical software such as R/S+, SAS and MATLAB. However recent developments in copulas like copula – GARCH models (Jondeau and Rockinger, 2006) and copula vines (Aas et al 2009) have not been incorporated so far to any statistical language/software. The dynamic copula toolbox we present here is a list of MATLAB functions specifically designed to estimate the two aforementioned classes of copulas and it is particularly oriented towards cases met in finance, although scientists from other fields can also use the toolbox without any major modifications. The toolbox is publicly available over the internet from the Mathworks site.

Key Words: Copula - GARCH, Copula -Vines, MATLAB

1. INTRODUCTION

In many financial theories like asset allocation, derivatives pricing and risk management, the dependence among the risk factors is of crucial importance. Copula functions describe the dependence structure of a set of variables and thus have become the standard tool in modeling dependence among financial time series especially when the researcher does not want to resort to the highly criticized assumption of multivariate normality. Examples of applications of copulas in finance can be found at Mendez et al (2004), Dias and Embrechts (2007), Van de Goorbergh et al (2005), Rodriguez (2007) and Hurlimann (2004) to name a few. Patton (2007) provides a review of the use of copulas in finance while Genest et al (2009) depicts the increasing use of copulas in the statistical literature.

Sklar's theorem (Sklar, 1959) provides the link between a joint distribution and the corresponding copula. According to it, for every p – dimensional distribution $\boldsymbol{F}$, with marginal distributions F_i, $i = 1, \ldots, p$ there exists a copula $\boldsymbol{C}$, such that:

$$\boldsymbol{F}\left(x_1,...,x_p\right)=\boldsymbol{C}(F_1\left(x_1\right),...,F_p\left(x_p\right)). \tag{1}$$

The copula defined in (1) is unique if all marginal distributions are continuous. From (1) one can obtain the copula according to the following formula:

$$\boldsymbol{C}\left(u_1,...,u_p\right)=\boldsymbol{F}\left(F_1^{-1}\left(u_1\right),...,F_p^{-1}\left(u_p\right)\right), \tag{2}$$

where

$$u_i=F_i\left(x_i\right),i=1,...,p.$$

If $\boldsymbol{F}$ is p – times differentiable, the joint density can be obtained:

$$\begin{aligned}\boldsymbol{f}\left(\boldsymbol{x}\right) &= \frac{\partial^p}{\partial x_1\partial x_2...\partial x_p}\boldsymbol{F}(\boldsymbol{x})= \\ &= \prod_{i=1}^{p}f_i\left(x_i\right)\frac{\partial^p}{\partial u_1\partial u_2...\partial u_p}\boldsymbol{C}(F_1\left(x_1\right),...,F_p\left(x_p\right)) \\ \boldsymbol{f}\left(\boldsymbol{x}\right) &= \prod_{i=1}^{p}f_i\left(x_i\right)\boldsymbol{c}(F_1\left(x_1\right),...,F_p\left(x_p\right))\end{aligned}$$

and the corresponding copula density is:

$$\boldsymbol{c}\left(u_1,...,u_p\right)=\frac{\boldsymbol{f}\left(F_1^{-1}\left(u_1\right),...,F_p^{-1}\left(u_p\right)\right)}{\prod_{i=1}^{p}f_i\left(F_i^{-1}\left(u_i\right)\right)}, \tag{3}$$

where $\boldsymbol{x}=\left(x_1,...,x_p\right)$ and $\boldsymbol{c}$ is the copula density.

Copula parameters are usually estimated by optimizing the log – likelihood function:

$$\mathcal{L}\left(\boldsymbol{\xi};\boldsymbol{x}\right)=\sum_{j=1}^{T}\left(\sum_{i=1}^{p}\log\left(f_i\left(x_{i,t};\boldsymbol{\phi}_i\right)\right)+\log\left(\boldsymbol{c}(F_1\left(x_{1,t}\right),...,F_p\left(x_{p,t}\right);\boldsymbol{\theta})\right)\right), \tag{4}$$

where $\boldsymbol{\xi}=(\boldsymbol{\phi},\boldsymbol{\theta})$ is the vector that contains the marginal parameters $\boldsymbol{\phi}=\left(\boldsymbol{\phi}_1,...,\boldsymbol{\phi}_p\right)$ and the copula parameters $\boldsymbol{\theta}$. Equation 4 can be decomposed to two parts, the marginal log likelihoods:

$$m\mathcal{L}\left(\boldsymbol{\phi};\boldsymbol{x}\right)=\sum_{i=1}^{p}mll_{i}=\sum_{i=1}^{p}\sum_{j=1}^{T}\log\left(f_{i}\left(x_{i,t};\boldsymbol{\phi}_{i}\right)\right) \tag{5}$$

and the copula log likelihood:

$$c\mathcal{L}\left(\boldsymbol{\theta};\boldsymbol{u},\boldsymbol{\phi}\right)=\log\left(\boldsymbol{c}(F_{1}\left(x_{1,t}\right),...,F_{p}\left(x_{p,t}\right);\boldsymbol{\theta})\right), \tag{6}$$

where $\boldsymbol{u}=\left(F_{1}\left(x_{1}\right),...,F_{p}\left(x_{p}\right)\right)$. Since the number of parameters can be large even in moderate dimensions, two step methods are usually employed. The marginal parameters are estimated at the first step by optimizing the marginal log likelihoods mll_{i}, independently of each other and the copula parameters are estimated at the second step, by optimizing the corresponding copula log likelihood $c\mathcal{L}\left(\boldsymbol{\theta};\boldsymbol{u},\boldsymbol{\phi}\right)$, conditional upon the results from the first step. The properties of the two stage estimators are studied in Joe (1997) when the margins are estimated parametrically and in Chen and Fan (2004) when a semiparametric method for the margins is employed. A comparison of the two methods can be found at Kim et al (2007). In general, two stage estimators provide computational tractability in the expense of loss of full efficiency. The Dynamic copula toolbox supports the following general classes of models:

- Copula - GARCH models
- Copula Vines

Copula – GARCH models is the class of models where some of the parameters are potentially time varying, in an autoregressive manner, conditional on the set of past information. Patton (2006) extended Sklar's theorem to the conditional case and studied the attributes of this new class of models. Applications of copula - GARCH models in finance can be found for example in Panchenko(2006), Serban et al (2007), Huang et al (2009).and Wang et al (2009). The toolbox supports four time varying copulas, namely the Gaussian copula, t copula, Clayton copula and Symmetrized Joe - Clayton (SJC) copula. The latter two are supporter only when $p=2$, while the first two have no dimensional constraint. The time varying parameter is the correlation among the risk factors for the first two, Kendall's tau for the third and upper and lower tail dependence for the last.

Copula vines, known also as pair copula constructions are the class of models produced by the decomposition of a multivariate (p>2) copula to a cascade of bivariate copulas. A formal introduction to copula vines is beyond the scope of this paper and the interested reader is referred to Aas et al (2009) or Bredford and Cooke

(2001). In general, a p – dimensional copula can be decomposed into $\frac{p(p-1)}{2}$ bivariate copulas. Copula vines provide greater flexibility than the traditional multivariate copulas because some of the restrictions of multivariate copulas do not hold in the copula vine case. For example in a multivariate t copula all pairs share the same degree of freedom parameter, unlike a copula vine decomposed in t - copulas, where each pair is allowed to have different degrees of freedom. The toolbox supports two copula vine decompositions, the canonical vine and the d - vine, assuming that each bivariate copula is a t copula or a Clayton copula or an SJC copula. Applications of copula vines in finance can be found in Min et al (2009) and Chollete et al (2010).

The main functions of the toolbox are *modelspec* and *fitModel*. The former is used to define the model specifications and the latter is used to estimate the parameters of the model defined by *modelspec*. For the one step models, the user should call *modelspec* to define the model and then call *fitModel* to estimated the parameters. The same is true for the two step models also, only now the user has to repeat the procedure two times. First the marginal models are defined with *modelspec* and estimated with *fitModel*. Then the copula model is defined by *modelspec* and estimated with *fitModel*. These two functions will be presented in detail in the following two sections. For all the other functions in the toolbox, the interested user can find comments in the corresponding m - files. Function names or function inputs/outputs are emphasized, to avoid confusion. The toolbox is tested on MATLAB R2008B and it uses the optimization toolbox and statistics toolbox from Mathworks. The command:

$$>> \textit{CopulaToolboxTutorial}$$

provides a brief tutorial and a sort description of some of the Dynamic Copula Toolbox functions.

2. SPECIFICATION OF THE SUPPORTED MODELS

The function that estimates all supported models is *fitModel* however prior to the estimation of the model parameters, the model specifications should be defined. This is achieved with the command:

$$>> \textit{spec} = \textit{modelspec(data)}$$

The function *modelspec* creates a structured array called *spec* that contains all specifications of the model that the user wants to estimate. The function input,

data, is the data set. This routine utilizes the *menu* function of MATLAB to create a sequence of model specification options, conditioned upon the user's previous choices. Each option is presented with all possible values it can take and the user is called to choose one of these values. First, the user is called to define what type of model he wants to estimate. A snapshot of the first pop up menu item from *modelspec* is presented in figure 1

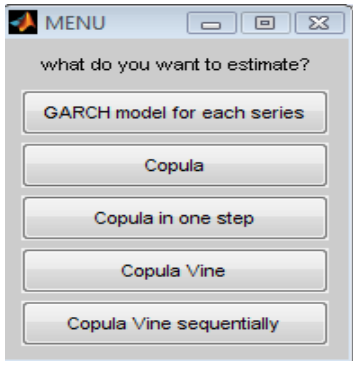

Figure 1: Snapshot of the first pop up menu item from the modelspec function

Choice number one: "GARCH model for each series" should be used to define the first step specifications when the user wants to employ a two step procedure for the estimation of the model parameters and when each series in the data set is heteroscedastic and possibly autocorellated, as in the case of financial data. The supported univariate GARCH models are:

- AR(q) - GARCH(1,1)
- AR(q) - GJR(1,1)

With the Gaussian, Student t or Skewed Student t (Hansen, 1994) distributions for the residuals. Having made this choice the user is called to define the order, q, of the autoregressive terms in the mean equation (non - negative integer), the variance equation ('GARCH(1,1)' or 'GJR(1,1)'), the distribution of the residuals ('Gaussian', 't' or 'SkewT') that defines the log likelihood function of the margins[1] and the method that transforms the standardized, *iid* residuals from the filtration, to uniform ('IFM' or 'CML'). This transformation is achieved by the *probability integral transform*: Let $\varepsilon_t, t = 1, ...T$ be a time series of *iid* variables, where we assume that: $\varepsilon_i \sim F$, $i = 1, ..., T$. Then the series:

[1] The expressions for the three supported marginal log likelihoods are given in the appendix.

$$u_t = F(\varepsilon_t),$$

is the probability integral transform of $\boldsymbol{\varepsilon}_t$ and it holds that $u_i \sim U[0,1]$, $i = 1, ..., T$. When the user chooses 'CML' the transformation is being made by the *empiricalCDF* function [2] :

$$F_i(x) = \frac{1}{T+1} \sum_{j=1}^{T} \mathbf{1}_{(X_{i,j} \leq x)},$$

where $\mathbf{1}$ denotes the indicator function: $\mathbf{1}(\text{expression}) = \begin{cases} 1, \text{ If expression is true} \\ 0, \text{ If expression is false} \end{cases}$ and T is the number of observations. On the other hand, if the user chooses 'IFM', the transformation to $U[0,1]$ is being made parametrically, by using the distribution assumed for the residuals. When the input argument *data* is multivariate the same specification is applied to all series. If the user wants to apply different specifications to each series he should input a single series to *modelspec*, estimate the parameters with the *fitModel* function and repeat this procedure for all series in the data set.

Choice number two: "Copula" should be used to define the second step specifications when the user wants to employ a two step procedure for the estimation of the model parameters. The available options for the copula models depend on p and they are presented in table 1 along with their possible values

user defined options	**possible values**			
copula family	Gaussian	t	Clayton	SJC
	$p \geq 2$	$p \geq 2$	$p = 2$	$p = 2$
dependence parameter evolution	DCC	DCC	Patton	Patton
		Static	Static	Static

Table 1: Supported copula options, based on the size p, of the data set.

The input *Copula family* defines the log – likelihood function (equation 6) of the dependence structure. The log likelihoods of the two elliptical copulas, the Gaussian and t copula are given by:

[2] The ecdf.m function of MATLAB is almost the same as the empiricalCDF.m, the only difference is that the denomenator of the former equals T while the denomenator of the latter is T+1. However ecdf.m should not be used because it can produce infitite values to some steps of the procedure, that cause the estimation routine to terminate. The *empiricalCDF.m* function was taken from A.J. Patton's home page: http://econ.duke.edu/~ap172/

$$\mathcal{L}_{Gaussian}\left(R;\boldsymbol{u}_t\right)=-\frac{1}{2}\sum_{t=1}^{T}\left(\log|R|+\boldsymbol{\epsilon}_t'\left(R^{-1}-I\right)\boldsymbol{\epsilon}_t\right). \tag{7}$$

$$\begin{aligned}\mathcal{L}_{St}\left(R,d,\boldsymbol{u}_t\right) &= -T\log\frac{\Gamma\left(\frac{d+p}{2}\right)}{\Gamma\left(\frac{d}{2}\right)}-pT\log\frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)}-\frac{d+p}{2}\sum_{t=1}^{T}\log\left(1+\frac{\boldsymbol{\epsilon}_t'R^{-1}\boldsymbol{\epsilon}_t}{d}\right)\\ &\quad -\sum_{t=1}^{T}\log|R|+\frac{d+1}{2}\sum_{t=1}^{T}\sum_{i=1}^{p}\log\left(1+\frac{\epsilon_{it}^2}{d}\right).\end{aligned} \tag{8}$$

The vector $\boldsymbol{\epsilon}_t$ is the vector of the *transformed standardized residuals* which depends on the copula specification. For the Normal copula it holds that: $\boldsymbol{\epsilon}_t = (\Phi^{-1}(u_{1,t}),\ldots,\Phi^{-1}(u_{p,t}))$, whereas for the t copula, the vector $\boldsymbol{\epsilon}_t$ is defined analogously, as: $\boldsymbol{\epsilon}_t = (t_d^{-1}(u_{1,t}),...,t_d^{-1}(u_{p,t}))$. Φ^{-1} denotes the inverse univariate standard normal distribution and t_d^{-1} is the inverse student's t distribution with d degrees of freedom. In both likelihoods R denotes the correlation matrix of $\boldsymbol{\epsilon}_t$. The log likelihoods of the two supported archimedean copulas, the Clayton and SJC copula, are presented in equations 9 and 10, respectively:

$$\mathcal{L}_{Clayton}\left(d;\boldsymbol{u}_t\right)=\sum_{t=1}^{T}\log\left(\left(1+d\right)\left(u_{1t}\cdot u_{2t}\right)^{-1-d}\left(u_{1t}^{-d}+u_{2t}^{-d}-1\right)^{-2-\frac{1}{d}}\right), \tag{9}$$

$$\begin{aligned}&\mathcal{L}_{SJC}\left(\tau^U,\tau^L;\boldsymbol{u}_t\right)=\\ &=\sum_{t=1}^{T}\log\left(\frac{\partial^2}{\partial u_1\partial u_2}\left(\frac{1}{2}\left(C_{JC}\left(\boldsymbol{u}_t|\tau^U,\tau^L\right)+C_{JC}\left(1-\boldsymbol{u}_t|\tau^L,\tau^U\right)+u_{1t}+u_{2t}-1\right)\right)\right),\end{aligned} \tag{10}$$

with $d=\frac{2\tau}{1-\tau}$, where τ is Kendall's tau and $\boldsymbol{u}_t=(u_{1t},u_{2t})$. C_{JC} is the Joe – Clayton copula with upper and lower tail dependence coefficients τ^U,τ^L respectively. The distribution of the JC copula is defined as:

$$C_{JC}\left(u,v|\tau^U,\tau^L\right)=1-\left(1-\frac{1}{\left(\frac{1}{\left(1-(1-u)^k\right)^\gamma}+\frac{1}{\left(1-(1-v)^k\right)^\gamma}-1\right)^{1/\gamma}}\right)^{1/k}, \tag{11}$$

where $k = \frac{1}{\log_2(2-\tau^U)}$ and $\gamma = -\frac{1}{\log_2 \tau^L}$. Both τ^U and τ^L belong to $(0,1)$. The analytic expressions of the SJC copula density and of the function $\frac{\partial C_{SJC}(u,v,\vartheta)}{\partial v}$ are given at the appendix.

The second option defines how copula parameters evolve through time. The choice 'static' defines a copula whose parameters are assumed constant over time. More specifically if a static t copula is assumed, a method similar to Marshal and Zeevi (2002) is employed, according to which the correlation matrix of the t copula is taken to be equal to the sample correlation of the transformed standardized residuals and only the degree of freedom parameter is estimated, therefore the number of copula parameters is one, independent of p. The 'DCC' choice defines that the degree of freedom parameter is static (for the t copula) and the correlation R_t evolves through time as in the DCC(1,1) model of Engle (2002):

$$\begin{aligned} Q_t &= (1-\alpha-\beta)\cdot\overline{Q} + \alpha\boldsymbol{\epsilon}_{t-1}\cdot\boldsymbol{\epsilon}'_{t-1} + \beta\cdot Q_{t-1} \\ R_t &= \widetilde{Q}_t^{-1} Q_t \widetilde{Q}_t^{-1}, \end{aligned} \tag{12}$$

where $\overline{Q}$ is sample covariance of $\boldsymbol{\epsilon}_t$, $\widetilde{Q}_t$ is a square $p \times p$ matrix with zeros as off - diagonal elements and diagonal elements the square root of those of Q_t. The parameter constraints for the DCC are the same as for the univariate GARCH(1,1) models:

$$\alpha + \beta < 1,\ a, \beta \in (0,1)\,.$$

The choice, 'Patton', available for the Clayton and SJC copulas, defines the evolution of the dependency parameter (Kendall's tau for the Clayton copula, upper and lower tail dependence coefficients for the SJC copula), according to the equations, defined in Patton (2006):

$$\tau_t = \Lambda\left(\omega + \beta\tau_{t-1} + \alpha\cdot\frac{1}{10}\sum_{i=1}^{10}|u_{1,t-i} - u_{2,t-i}|\right), \tag{13}$$

for the SJC copula, and

$$\Lambda\left(\omega + \beta\tau_{t-1} + \alpha\cdot|u_{1,t-i} - u_{2,t-i}|\right),$$

for the Clayton copula. Λ denotes the logistic transformation: $\Lambda(x) = (1 + e^{-x})^{-1}$ in order to keep the parameters of both Clayton and SJC copulas in $(0, 1)$.

Choice number three: "Copula in one step" should be used to define the model specifications when the user wants to employ the one step procedure for the estimation of copula parameters. This choice defines the options for both the marginal GARCH models and the copula. The available options for both the margins and the copula are identical to those of the two step procedures, therefore all models that can be estimated with the two step procedures can be estimated in one step, as well, with one exception. The one step is a fully parametric method, therefore the standardized residuals from the GARCH processes are transform to uniform parametrically, by automatically setting the PIT method to 'IFM'. Thus the user is not called to choose the PIT method, it is set to 'IFM' by default.

Choice number four:"Copula Vine" should be used to define the model specifications when the user wants to fit a copula vine to a multivariate ($p > 2$) data set that consists of *iid* uniform $U[0, 1]$ margins therefore, a filtration of the data set, usually through a GARCH model, that will extract the *iid* residuals from the raw data and transform them to uniform, is usually necessary. A copula vine is a decomposition of a multivariate copula to a product of bivariate copulas, in a form of a nested set of $p - 1$ trees. The first tree consists of $p - 1$ bivariate copulas, the second tree consists of $p - 2$ copulas and so on. The toolbox supports estimation of d - vines and canonical vines, assuming that all the bivariate copulas that the multivariate copula is decomposed to, are t copulas or Clayton copulas or SJC copulas. Table 2 illustrates the canonical vine and d - vine decomposition of a five dimensional copula. The corresponding copula vine log likelihood is the sum of these $(p(p-1))/2$ bivariate log likelihoods. The only difference between choices four: "Copula Vine" and five: "Copula Vine sequentially" is on how *fitModel* estimates the copula parameters. "Copula Vine" choice, forces the fitting function to estimate all the parameters of the vine at the same time, in one step. However this one step optimization scheme can be significantly time consuming therefore the use of good starting values is essential. The choice "Copula Vine sequentially" forces the fitting function to estimate the vine parameters in multiple steps, in each step a bivariate copula likelihood found in the corresponding decomposition, is optimized. The procedure ends when the parameters of all the bivariate copulas have been estimated. This sequential procedure is extremely faster when compared to one step method: it can save up to 90% of the time but it may result to a significant loss of efficiency. Aas et al (2009) suggests

that one should estimate the copula parameters with the multi step method first and then use these estimates as starting values for the one step method. In practice however the results under both methods are very close thus a user that favors speed over accuracy might consider using only the multi step estimators.

$c(F(x_1),...,F(x_5)) =$	$c(F(x_1),...,F(x_5)) =$	
$c_{12}\cdot c_{13}\cdot c_{14}\cdot c_{15}$	$c_{12}\cdot c_{23}\cdot c_{34}\cdot c_{45}$	Tree 1
$c_{23\|1}\cdot c_{24\|1}\cdot c_{25\|1}$	$c_{13\|2}\cdot c_{24\|3}\cdot c_{35\|4}F_{3\|4}, F_{5\|4}$	Tree 2
$c_{35\|12}\cdot c_{34\|12}$	$c_{14\|23}\cdot c_{25\|34}$	Tree 3
$c_{45\|123}$	$c_{15\|234}$	Tree 4

Table 2: The decomposition of a five dimensional copula, implied by the canonical vine (left panel) and the d - vine decomposition. $c_{ij|kl}$ stands for $c\left(F\left(x_{i|kl}\right), G\left(x_{j|kl}\right)\right)$, where F and G denote the distribution functions of $x_{i|kl}$ and $x_{j|kl}$, respectively.

3. ESTIMATION OF THE SUPPORTED MODELS

Having created the structure array that contains the model specifications the model parameters are estimated with the *fitModel* function:

>>[parameters, LogL, evalmodel, GradHess, udata] = fitModel(spec, data, solver)

The inputs of this function are *spec*, the structure that contains the model specifications, created with *modelspec*, the data array *data* and *solver*, a string that takes values 'fminunc' or 'fmincon', depending on which MATLAB function the user wants to use in order to optimize the corresponding log likelihood function. Since most of the supported models are constrained by nature, a reparametrization of the parameters is needed, in order to estimate them with fminunc. The increasing functions that are used to reparametrise the problem depend on the domain of the real parameter and are illustrated in table 3.

Real Parameter Space	reparametrization function
$y \in (a, +\infty)$	$y = a + \exp(x), \;\; x \in \mathbb{R}$
$y \in (a, a+b)$	$y = a + \frac{b\exp(x)}{1+\exp(x)}, \;\; x \in \mathbb{R}$

Table 3: Reparametrizations supported by the toolbox. x is an unconstrained parameter and y is the corresponding constrained one.

The output arguments of the function are the column vector *parameters* that contains the model parameters, *LogL* which is the log likelihood value at the optimum,

evalmodel which is a structure array that contains secondary outputs from the optimization procedure, like the exitflag and the number of iterations, a structure array *GradHess* that contains both the standard errors based on the Godambe information matrix and all elements used to calculate the standard errors and *udata* which contains the filtered data, transformed to uniform, if the model defined by *modelspec* is "GARCH model for each series". In any other case *udata* is an empty matrix. When the estimation is finished the optimum parameters along with their standard errors, the values of the log likelihood, the Akaike (AIC) and Schwarz (BIC) are printed on the computer screen. Figure 2 illustrates a typical output of the *fitModel* function, printed on the screen.

```
Computing finite-difference Hessian using user-supplied objective function.
Numerical hessian is not PSD, standard errors will be calculated with the Hessian from the solver.

Standard errors are calculated by the chain rule.

 Estimation output
parameter   St. Error    t-stats
---------------------------------
13.2175      5.449       2.4257
0.0160       0.006       2.7994
0.9797       0.007       142.4596
---------------------------------
Akaike: -577.1679
BIC: -561.7608
Log Likelihood: 291.584
---------------------------------
Estimation time is 24.48 seconds
```

Figure 2: FitModel output, printed on the computer screen.

Care should be taken when the model of choice defined with *modelspec* is "Copula" or "Copula Vine" or "Copula Vine sequentially". In all these cases the second input argument, *data,* should consist of margins that are *iid* uniform, therefore prior to estimating such models the user should transform his data to uniform as follows: First the user defines the desired model as "GARCH model for each series" and fits a GARCH model to each series with *fitModel,* in order to extract *udata*. Then he redefines the model as "Copula" or "Copula Vine" or "Copula Vine sequentially" and estimates the parameters by calling *fitModel*, with *udata* as the second input

argument.

3.1. Standard errors of the parameters

The variance covariance (VCV) matrix of model parameters is the Godambe information matrix:

$$VCV = H^{-1} M H^{-1}, \tag{14}$$

where H^{-1} is the expectation of the log likelihood Hessian and M is the covariance of the log likelihood scores. Both Hessian and scores of the log likelihood function are calculated numerically, at the optimum, with the functions *Hessian_2sided* and *MyFuncScores*[3], respectively. However the numerical Hessian calculated with *hessian_2sided* is not always positive definite at the optimum, unlike the Hessians provided by both fminunc or fmincon which are always positive definite. When the numerical Hessian is not positive definite and the *solver* input is 'fmincon', H in equation 14 is replaced by the Hessian provided by the solver, which is not a good approximation of the real hessian, especially when some of the parameters are close to the boundaries and thus standard errors might be inaccurate. When the *solver* is 'fminunc' H^{-1} is calculated from the fminunc Hessian H_u, with the chain rule, since H_u corresponds to the hessian of the unconstrained parameters. Let $\boldsymbol{c} = (c_1, ..., c_n)$ and $\boldsymbol{u} = (u_1, ..., u_n)$ be the vectors of constrained and unconstrained parameters, respectively and $h : \mathbb{R}^n \to \mathbb{R}^n$ the reparametrization function: $\boldsymbol{u} = h(\boldsymbol{c})$. The Hessian of the log likelihood function f, with respect to $\boldsymbol{c}$ is given by

$$H_c = J \cdot H_u \cdot J' + V, \tag{15}$$

where J is the Jacobian matrix of the reparametrization: $J = \begin{bmatrix} \frac{\partial u_1}{\partial c_1} & \cdots & \frac{\partial u_n}{\partial c_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial u_1}{\partial c_n} & \cdots & \frac{\partial u_n}{\partial c_n} \end{bmatrix}$ and V is an $n \times n$ matrix whose generic element V_{ij} is defined as: $V_{ij} = \nabla_u f' \cdot \mathcal{H}_{ij}$, with $\mathcal{H}_{ij} = \begin{bmatrix} \frac{\partial^2 u_1}{\partial c_i \partial c_j} & \cdots & \frac{\partial^2 u_n}{\partial c_i \partial c_j} \end{bmatrix}'$ and $\nabla_u f = \begin{bmatrix} \frac{\partial f}{\partial u_1} & \cdots & \frac{\partial f}{\partial u_n} \end{bmatrix}'$. In the toolbox case, instead of $\boldsymbol{u} = h(\boldsymbol{c})$, an one to one reparametrization was used: $u_i = h_i(c_i)$[4], $i =$

[3] Both functions are minor modifications of two similar functions found in the MFE GARCH toolbox of Kevin Sheppard, available at: http://www.kevinsheppard.com/wiki/MFE_MATLAB_Introduction

[4] For simplicity to the notations h_i corresponds to the inverse of the reparametrization functions g_i defined in table 3. That is $h_i \equiv g_i^{-1}$.

$1, ..., n$ therefore equation 15 is simplified to

$$H_{\boldsymbol{c}} = J^* \cdot H_{\boldsymbol{u}} \cdot J^* + \nabla^2_{\boldsymbol{c}} h \cdot \nabla_{\boldsymbol{u}} f',$$

where $J^* = \begin{bmatrix} \frac{du_1}{dc_1} & ... & 0 \\ \vdots & \ddots & \vdots \\ 0 & ... & \frac{du_n}{dc_n} \end{bmatrix}$ and $\nabla^2_{\boldsymbol{c}} h = \left(\frac{d^2 h_1}{dc_1^2}, ..., \frac{d^2 h_n}{dc_n^2} \right)'$.

4. EMPIRICAL APPLICATION OF THE TOOLBOX

The dependence of the daily log – returns of four stocks listed in the S&P500 index, with tickers AXP, BA, LMT and MS[5] is studied. The data was collected from yahoo! finance (http://finance.yahoo.com/q/cp?s=^GSPC+Components) covering a period from January 04, 2002 to December 29, 2006, total 1256 returns observations. From the stock prices, $p_{i,t}$, the geometric returns of the i - th stock, for the day - t, $r_{i,t}$, are calculated:

$$r_{i,t} = \log \left(\frac{p_{i,t}}{p_{i,t-1}} \right)$$

For the copula vine models the four variate data set was. For the copula models the bivariate set consisting of AXP and MS was used. The descriptive statistics of the returns are calculated with the *DescriptiveStatistics* function of the toolbox and are presented in table 4. We observe that all variables exhibit the characteristic features of financial time series like excess kurtosis and non positive skewness. The normality hypothesis is rejected for all cases, at every confidence level, according to the Jarque Bera test.

[5] AXP is American Express, BA is Boeing, LMT is Lockheed Martin and MS is Morgan Stanley

Statistic	Stock			
	AXP	BA	LMT	MS
Mean	0.0004	0.0006	0.0005	0.0002
Medean	-0.0005	0.0006	0.0002	0.0003
IQR	0.0150	0.0207	0.0151	0.0210
St. Deviation	0.0166	0.0172	0.0148	0.0194
Skewness	-0.0476	0.0056	-0.2920	-0.1068
Kurtosis	9.8807	4.7032	8.2831	5.6921
JB test p value	0	0	0	0

Table 4: Descriptive statistics of the four stock returns. JB test, p - val is the probability that the data comes from the Normal distribution, according to Jarque - Berra normality test

4.1. Two step estimation of Copula GARCH models

In the two step procedure the marginal parameters are estimated at the first step. An AR(1) - GJR(1,1)[6] with Skew-T residuals was fitted to each series. The choice of the Skew-T distribution is justified from the fact that three out of the four series in the data set exhibit negative skewness and the only distribution supported by the toolbox that can account for the presence of skewness in data is the Skew-T distribution. The AR(1) - GJR(1,1) model can be described as follows: Let $r_{i,t}$ denote the returns of the i-th asset at time t. Then:

$$r_{i,t} = c_0 + c_1 r_{i,t-1} + e_{i,t} \tag{16}$$

$$e_{i,t} = h_{i,t}\varepsilon_{i,t}, \quad \varepsilon_{i,t} \sim SkT(v,\lambda) \tag{17}$$

$$h_{i,t} = \omega_{i,t} + \alpha e_{i,t-1}^2 + \beta h_{i,t-1} + \gamma e_{i,t-1}^2 \mathbf{1}\left(e_{i,t-1} < 0\right) \tag{18}$$

Thus, each marginal model has eight parameters, two parameters (c_0, c_1) in the mean equation, four parameters $(\omega, \alpha, \beta, \gamma)$ in the variance equation and two distributional parameters (v, λ). $\mathbf{1}$ is the indicator function. To fit the GARCH model to each of the series, the *modelspec* function is called first, to create the structure array spec, that contains the model specifications. and then the *fitModel* function is called. All models were estimated with *'fminunc'* as the solver and *'IFM'* as the PIT method.

[6] For the GJR(1,1) model, the constraints applied to equation 18 are: $a+\gamma+2\beta < 2, a > -\gamma, \beta \in (0,1)$

The estimated parameters along with their standard errors, the values of the log likelihood and the Akaike (AIC) and Schwarz (BIC) criteria are illustrated in table 5.

	AXP	BA	LMT	MS
c_0	0.00042	0.00097	0.00068	0.0004
	(0.0003)	(0.0004)	(0.0003)	(0.0003)
c_1	-0.0403	-0.1234	-0.0426	-0.0281
	(0.0263)	(0.0315)	(0.0309)	(0.0262)
ω	$2.06{\cdot}10^{-6}$	$2.88 \cdot 10^{-6}$	$3.32 \cdot 10^{-6}$	$1.98\ 10^{-6}$
	$(9.1\ 10^{-7})$	$(1.60\ 10^{-6})$	$(2.47\ 10^{-6})$	$(9\ 10^{-7})$
α	0.0232	0.0065	0.0557	0.0042
	(0.0126)	(0.0117)	(0.0221)	(0.013)
β	0.9158	0.9537	0.9036	0.9565
	(0.0187)	(0.0185)	(0.0428)	(0.0187)
γ	0.1166	0.0586	0.0479	0.0635
	(0.0424)	(0.0198)	(0.0399)	(0.0418)
v	5.9828	12.736	9.114	10.481
	(1.4675)	(6.5802)	(2.988)	(1.4663)
λ	0.1009	0.0637	0.0205	-0.043
	(0.0359)	(0.0421)	(0.0385)	(0.0359)
$\mathcal{LL}$	3683	3428.8	3694.5	3339.02
AIC	-7350	-6841.6	-7373	-6662
BIC	-7308.9	-6800.5	-7332	-6621

Table 5: The estimated parameters correspond to equations 16 to 18. $\mathcal{LL}$ is the value of the log - likelihood function at the optimum. AIC and BIC correspond to the Akaike and Swcharz criteria. Standard errors are in parentheses.

4.2. Copula Results

Having estimated the marginal parameters, *modelspec* is called again to define the specifications of the copula. Then the copula is fitted by calling the *fitModel* function. Tables 6 and 7 illustrate the estimated copula parameters for the bivariate data set. Table 6 contains the estimated parameters from three elliptical copulas, namely the static t copula (t), the time varying t (tDCC) and Gaussian (GDCC) copulas. In both time varying copulas the time varying parameters is the correlation that follows the DCC model of Engle. Table 7 contains the estimated parameters

from three Archimedean copulas, the static SJC copula (SJC) along with the time varying SJC (tvSJC) and Clayton (tvC) copulas

	t	tDCC	GDCC
υ	10.0734	13.2175	-
	(0.011)	(5.449)	
α	-	0.016	0.0186
		(0.006)	(0.007)
β	-	0.9797	0.9747
		(0.007)	(0.009)
AIC	-542.087	-576.903	-566.751
BIC	-536.951	-561.496	-556.480
$\mathcal{LL}$	272.043	291.451	285.376

Table 6: Estimated parameters of three elliptical copula models. The parameters α and β correspond to the parameters of equation 12 and υ is the degree of freedom parameter of the t copula. Asymptotic standard errors are in parentheses.

	tvC	SJC		tvSJC	
τ^U - τ^L		0.4079	0.3461		
		(0.0325)	(0.0428)		
ω	-0.6943			1.6658	0.0877
	(0.3181)			(0.944)	(0.051)
α	-0.908			-7.032	-0.4709
	(0.5061)			(4.466)	(0.277)
β	-0.5007			-0.986	0.9804
	(0.3547)			(0.070)	(0.011)
AIC	-382.057	-507.504		-528.669	
BIC	-366.45	-497.232		-497.854	
$\mathcal{LL}$	194.028	255.752		270.334	

Table 7: Estimated parameters of three Archimedean copula models. The parameters ω, α and β correspond to the parameters of equations 13 and 14. For the tvSJC copula the left column corresponds to the upper tail equation parameters. Asymptotic standard errors are in parentheses.

4.3. One step versus two step results

To illustrate the differences between the two estimation methods, the tDCC model of section 4.1 was also estimated with the one step procedure. The estimated para-

meters from the two models are presented in table 8.

two step results			one step results		
	marginal parameters			marginal parameters	
	AXP	MS		AXP	MS
c_0	0.00043	0.0004	c_0	0.00054	0.00055
	(0.0003)	(0.0003)		(0.0003)	(0.00042)
c_1	-0.0402	-0.0281	c_1	-0.0431	-0.0175
	(0.0263)	(0.0262)		(0.0247)	(0.0244)
ω	2.03 10^{-6}	1.98 10^{-6}	ω	$3.06 \cdot 10^{-6}$	$2.8 \cdot 10^{-6}$
	(9.1 10^{-7})	(9 10^{-7})		(1.05 10^{-6})	(1.26 10^{-6})
α	0.0245	0.0042	α	0.0396	0.0135
	(0.0126)	(0.013)		(0.0143)	(0.0117)
β	0.9158	0.9565	β	0.9103	0.9561
	(0.0187)	(0.0187)		(0.0196)	(0.013)
γ	0.1138	0.0635	γ	0.0761	0.0382
	(0.0424)	(0.0418)		(0.0366)	(0.0192)
v	5.9807	10.481	v	5.4616	8.4805
	(1.4675)	(1.4663)		(1.1301)	(2.3537)
λ	0.1013	-0.043	λ	0.0989	-0.0411
	(0.0359)	(0.0359)		(0.032)	(0.0363)
copula parameters			copula parameters		
v	13.2175		v	12.0681	
	(5.449)			(4.8903)	
α	0.016		α	0.0143	
	(0.006)			(0.0054)	
β	0.9797		β	0.9804	
	(0.007)			(0.0066)	
$\mathcal{LL}$	7313.67		$\mathcal{LL}$	7318.39	

Table 8: Comparison of the estimated parameters of the tDCC copula, from the one and two step methods.

4.4. Copula Vine results

The toolbox supports six different copula vine models, the canonical vine and d - vine, assuming that each bivariate copula in the cascade is t, Clayton or SJC copula. The log likelihood function of the canonical vine t copula model (tCVine) did not

converge to a solution with fminunc and the numerical hessian was not positive definite, when fmincon was the solver, consequently the results are not reported here because they might be inaccurate. The results from the five copula vines are illustrated in table 9.

	canonical vine				d - vine			
	Clayton	SJC			Clayton	t	SJC	
a_{12}	0.2020	0.1564	0.2673	a_{12}	0.1895	12.934	0.1235	0.2537
	(0.021)	(0.0501)	(0.0427)		(0.019)	(0.029)	(0.0538)	(0.0421)
a_{13}	0.1201	0.0520	0.1324	a_{23}	0.2259	43.898	0.1838	0.2883
	(0.0212)	(0.032)	(0.0410)		(0.0162)	(0.006)	(0.0435)	(0.0388)
a_{14}	0.3008	0.4087	0.3464	a_{34}	0.0874	13.22	0.0608	0.0857
	(0.0191)	(0.0337)	(0.0464)		(0.0167)	(0.0144)	(0.0325)	(0.0387)
$a_{23\|1}$	0.1781	0.1657	0.1845	$a_{13\|2}$	0.0136	23.75	0.0000	0.0027
	(0.0176)	(0.0422)	(0.0403)		(0.017)	(0.0107)	(0.003)	(0.0124)
$a_{24\|1}$	0.0895	0.0311	0.0635	$a_{24\|3}$	0.1441	60.809	0.0079	0.1861
	(0.0139)	(0.0287)	(0.0328)		(0.016)	(65.048)	(0.0442)	(0.044)
$a_{34\|12}$	0.0090	0.0018	0.000	$a_{14\|23}$	0.2409	14.305	0.3498	0.2082
	(0.0124)	(0.0087)	(0.000)		(0.0214)	(6.800)	(0.0422)	(0.0605)
AIC	-836.241	-1050.473		AIC	-823.335	-1144.1	-1031.771	
BIC	-805.427	-988.845		BIC	-792.521	-1113.2	-970.143	
$\mathcal{LL}$	424.120	537.236		$\mathcal{LL}$	417.667	578.031	527.887	

Table 9: Estimated parameters from the two canonical vine (left panel) and three D vine models. $a_{ij|kl}$ denotes the parameter(s) of the copula $c_{ij|kl}$ in the decomposition. Zero value in a parameter indicates that the estimated parameter was less that 0.0001. Standard errors are in parentheses.

5. NUMERICAL ISSUES

During the development of the toolbox I encountered several numerical issues, that forced the optimization to crash, or provided `NaN`'s or `inf` for some of the copula parameters. This section deals with issues concerning the optimization of the log likelihood functions, like the starting values and the choice of the solver. For general numerical issues concerning copulas the interested reader is referred to Yan (2007).

When the user calls the *fitModel* function, he is asked to input the starting values for the optimization procedure. A good choice of the starting values is essential, since

if the starting values are away from the actual ones the routine might take a lot of time to converge to the true parameters vector, get stucked to a local minimum or even not converge at all. The user has three choices for the starting values. He can use a naive guess, stored inside the structured array *spec* that contains the models specifications in the *spec.theta0* array, that is created automaticaly when *modelspec* is called, he can type a vector of starting values in the Command Window or he can choose any vector located in the Workspace, to be used as starting values. In the latter two cases, the provided vector should have the correct size, otherwise an error message is printed and the procedure ends. The starting values should always be a column vector with structure that is identical to the structure of the estimated parameters vector. For example, in one step copula models the correct structure is: $\left[\phi_1, ..., \phi_p, \vartheta\right]'$ where ϕ_i and ϑ represent the marginal and copula parameters, respectively. Empirical implementation of the toolbox has revealed that two step copula models can be efficiently estimated with naive starting values, unlike one step copula models or copula vines where the choice of good starting values is critical. For one step copula models, the user is advised to first estimate the model in two steps and use the two step results as starting values for the one step model. For copula vines the user should estimate the bivariate copula parameters found in the cascade sequentially and use these estimates as starting values for the vine estimation.

Another critical choice is that of the solver. The constrained nonlinear minimization solver (fmincon) is more robust than the unconstrained one, in the sense that if good starting values are used, the solver always converges to a solution. The drawback of fmincon is that when some parameters are near the boundary, the solver Hessian is a rather poor approximation of the actual Hessian and hence the variance covariance matrix of the estimated parameters might be inaccurate. On the other hand fminunc provides a much better approximation of the real Hessian therefore fminunc is a better choice than fmincon when asymptotic standard errors are to be computed. The drawback of fminunc is that it doesn't always converge to a solution, in highly complex models. Furthermore there is an issue concerning linear inequality constraints. During the first iterations of the estimation procedure, fmincon may choose some extreme values for the parameters that may cause the optimization to crash, therefore the parametric space used by the toolbox is wide enough to cover almost all real data set cases, but not the same as the true parametric space. Table 12 summarizes the parametric spaces used by fmincon. In cases where the user believes that the true value of the parameter is outside the parametric space he can change

the bounds of the parameters, defined and created by the *CreateFminconConstraints* function.

parameter	true parametric space	used parametric space
degree of freedom parameter v	$(2, +\infty)$	$[2.01, 200]$
asymmetry parameter λ	$(-1, 1)$	$[-0.5, 0.5]$
GARCH parameter ω	$(0, +\infty)$	$\left[10^{-5}s, s\right]$
GARCH or DCC parameter a	$(0, 1)$	$\left[10^{-9}, 0.5\right]$
GARCH or DCC parameter b	$(0, 1)$	$\left[10^{-9}, 0.9995\right]$

Table 12: True and used parametric space. s denotes the sample variance of the corresponding series

A similar problem might appear when the fminunc solver is used. In order to keep the calculation of the Hessian, calculated by the fminunc Hessian, simple and feasible, each reparametrization function depends on only one parameter, thus the GARCH or DCC constraint $a + b < 1$, is not directly incorporated to the model. Instead, at each iteration the sum $a + b$ is calculated, if $a + b > 1$, a is set equal to $0.99999 - b$. In case this constraint is activated by the optimum values the standard error of a is NaN or Inf. In empirical applications however the constraint $a + b > 1$ is activated only for the first few iterations, if it is activated at all and the optimum values of the parameters, have sum $a^{opt} + b^{opt}$, that is close to unity but extremely rarely above unity. If such a problem does come up the user is advised to experiment with the reparametrization function *RescaleParameters*, in order to narrow the parametric space, or change the starting values.

6. CONCLUSION

A MATLAB toolbox for the estimation of the parameters of various copula models was presented and although it is designed to tackle problems usually met in finance scientists from other fields can also use it without any modifications. The toolbox is constantly expanding, following the research on the field, to cover more cases. Possible future extensions are:

1. More copulas, like the skewed t – copula (De Marta and McNeil, 2002)
2. Copula mixture models.
3. Functions to simulate copulas.
4. Extensions of copula vines like the CAVA model (Heinen and Valdesogo, 2009).
5. Analytical derivatives of the supported models.

6. Bayesian estimation of copula models.

The author welcomes collaborations with users that will lead to extensions of the toolbox.

7. APPENDIX

7.1. The log likelihood functions of the supported GARCH models

The log likelihoods of the two GARCH type models for the marginal time series $r_{i,t}$, are:

The Gaussian log likelihood

$$\mathcal{LL}_t\left(\phi_i;\varepsilon_{i,t}\right)=-\frac{1}{2}\left(\log\left(2\pi\right)-\log\left(h_{i,t}\right)-\varepsilon_{i,t}^2\right).$$

The student t log likelihood

$$\mathcal{LL}_t\left(\phi_i;\varepsilon_{i,t}\right)=\log\left(\frac{\Gamma\left(\frac{v+1}{2}\right)}{\Gamma\left(\frac{v}{2}\right)}\right)-\frac{1}{2}\left(\log\left(\pi\left(v-2\right)\right)+\log\left(h_{i,t}\right)+\left(v+1\right)\log\left(1+\frac{\varepsilon_{i,t}^2}{v-2}\right)\right).$$

The skewed student t log likelihood

$$\mathcal{LL}_t\left(\phi_i;\varepsilon_{i,t}\right)=-\frac{1}{2}\log\left(h_{i,t}\right)+p_t\left(v,\lambda;\varepsilon_{i,t}\right)$$

where p_t denotes the natural logarithm of the corresponding density:

$$d_t\left(\varepsilon_{i,t},v,\lambda\right)=\begin{cases} bc\left(1+\frac{1}{v-2}\left(\frac{b\varepsilon_{i,t}+a}{1-\lambda}\right)^2\right)^{-\frac{v+1}{2}} & \varepsilon_{i,t}<-a/b \\ bc\left(1+\frac{1}{v-2}\left(\frac{b\varepsilon_{i,t}+a}{1+\lambda}\right)^2\right)^{-\frac{v+1}{2}} & \varepsilon_{i,t}\geq -a/b, \end{cases}$$

with $a=4\lambda c\frac{v-2}{v-1}$, $b^2=1+3\lambda^2-a^2$ and $c=\frac{\Gamma\left(\frac{v+1}{2}\right)}{\Gamma\left(\frac{v}{2}\right)\sqrt{\pi(v-2)}}$.

7.2. The density and h function of the Symmetrized Joe - Clayton copula

The SJC copula was defined in Patton (2006), however the author does not provide an explicit formula of the density of the copula. Furthermore the h – function of the SJC copula is not available, since, according to my knowledge, this is the

first attempt to estimate a vine assuming all bivariate copulas are SJC copulas. The distribution of the the SJC copula is derived form the Joe - Clayton copula (eq. 8):

$$C_{SJC}\left(u,v|\tau^U,\tau^L\right)=\frac{1}{2}\left(C_{JC}\left(u,v|\tau^U,\tau^L\right)+C_{JC}\left(1-u,1-v|\tau^L,\tau^U\right)+u+v-1\right)$$

Therefore, the corresponding h - function is:

$$h_{SJC}\left(u,v|\tau^U,\tau^L\right)=\frac{\partial C_{SJC}}{\partial v}=\frac{1}{2}\left(\frac{\partial C_{JC}\left(u,v|\tau^U,\tau^L\right)}{\partial v}-\frac{\partial C_{JC}\left(1-u,1-v|\tau^L,\tau^U\right)}{\partial v}+1\right)$$

Where:

$$\begin{aligned}\frac{C_{JC}\left(u,v|\tau^U,\tau^L\right)}{\partial v} &= \frac{\left(1-\frac{1}{Z^{1/\gamma}}\right)^{-\frac{1}{k}+1}}{\left(1-(1-v)^k\right)^{\gamma+1}\cdot Z^{\frac{1}{\gamma}+1}}\\ Z &= \frac{1}{\left(1-(1-u)^k\right)^{\gamma}}+\frac{1}{\left(1-(1-v)^k\right)^{\gamma}}-1\end{aligned}$$

The density of the SJC copula is produced in a similar manner:

$$\begin{aligned}c_{SJC}\left(u,v|\tau^U,\tau^L\right) &= \frac{\partial^2 C_{SJC}}{\partial u\partial v}=\\ &\frac{1}{2}\left(\frac{\partial^2 C_{JC}\left(u,v|\tau^U,\tau^L\right)}{\partial u\partial v}-\frac{\partial^2 C_{JC}\left(1-u,1-v|\tau^L,\tau^U\right)}{\partial u\partial v}\right)\end{aligned}$$

Where:

$$
\begin{aligned}
\frac{\partial^2 C_{JC}\left(u,v|\tau^U,\tau^L\right)}{\partial u \partial v} &= A - B \\
A &= \frac{\gamma \cdot k\left(1-\frac{1}{Z^{1/\gamma}}\right)^{-2+1/k} \cdot \left(\frac{1}{\gamma}-1\right) \cdot (1-u)^{k-1} \cdot (1-v)^{k-1}}{\left(1-(1-u)^k\right)^{1+\gamma} \cdot \left(1-(1-v)^k\right)^{\gamma+1} \cdot Z^{2+1/\gamma}} \\
B &= \frac{k\left(1-\frac{1}{Z^{1/\gamma}}\right)^{-2+1/k} \cdot \left(\frac{1}{k}-1\right) \cdot (1-u)^{k-1} \cdot (1-v)^{k-1}}{\left(1-(1-u)^k\right)^{1+\gamma} \cdot \left(1-(1-v)^k\right)^{\gamma+1} \cdot Z^{2+2/\gamma}}
\end{aligned}
$$

8. REFERENCES

Aas K, Czado C, Frigessi A, Bakken H (2009). "Pair – copula constructions of multiple dependence." *Insurance: Mathematics and Economics,* **44**(2), 182 - 198.

Ausin M C, Lopez H F (2010). "Time varying joint distributions through copulas." *Computational Statistics and Data Analysis*, **54**(11), 2383 - 2399.

Bollerslev T, Wooldridge J M (1992). "Quasi – maximum likelihood and inference in dynamic models with time – varying covariances." *Econometric Reviews,* **11**(2), 143 – 172.

Bedford T., Cooke R M (2001). "Probability density decomposition for conditionally dependent random variables modeled by vines." *Annals of Mathematics and Artificial Intelligence* **32**, 245 – 268.

Chen X, Yanqin Fan, Victor Tsyrennifov (2004). "Efficient Estimation of Semiparametric Multivariate Copula Models." *Working Paper 0420,* Department of Economics, Vanderbilt University

Chollete L, Victor de la Pe[redacted]a, Ching-Chih Lu (2010) "International diversification: A copula approach." *Journal of Banking & Finance* (article in press).

De Marta S, McNeil A (2005). "The t - copula and related copulas." *International Statistical Review* **73** (1), 111 - 129.

Dias A, Embrechts P, (2004). "Dynamic copula models for multivariate high frequency data in finance." *ETH working paper.*

Embrechts P, McNeil AJ, Straumann D, (2002). "Correlation and dependence in risk management: Properties and pitfalls." *Risk management: Value at risk and beyond,* 176 – 223. Cambridge university press, Cambridge.

Engle R F (2002). "Dynamic conditional correlation – a simple class of multivariate GARCH models." *Journal of Business and Economic Statistics* **20**(3), 339 – 350.

Genest C, Ghoudi K, Rivest L P (1995). "A semiparametric estimation procedure of dependence parameters in multivariate families of distributions." *Biometrica*, **82**(3), 543 – 552.

Genest C, Gendron M, Bourdeau - Brien M (2009). "The Advent of Copulas in Finance." *The European Journal of Finance,* 15: 7 609 - 618.

Huang J J, Lee L J, Liang H, Lin W F (2009). "Estimating value at risk of portfolio by conditional copula-GARCH method." *Insurance:Mathematics and economics,* **45**(3), 315 - 324

Hurlimann V (2004). "Fitting bivariate cumulative returns with copulas." *Computational Statistics and Data Analysis* **45**(2), 355 - 372.

Heinen A, Robles A (2009). "Asymmetric CAPM dependence for large dimensions: The Canonical Vine Autoregressive Copula model." *Available at SSRN: http://ssrn.com/abstract*

Joe H, (1997). *Multivariate models and dependence concepts.* Chapman and Hall, London.

Jondeau E, Rockinger M (2006). "The copula - GARCH model of conditional dependencies: An international stock market application." *Journal of International Money and Finance* **25**(5), 827 - 853.

Kim G, Silvapulle M J, Silvapulle P (2007). "Comparison of semiparametric and parametric methods for estimating copulas." *Computational Statistics and Data Analysis* **51**(6), 2386 - 2850.

Min A, Czado C, Baumann T, Dakovic R (2009). "Pair Copula Constructions for modeling exchange rate dependence." *Unpublished manuscript*

Mashal R, Zeevi A (2002). "Beyond correlation: Extreme co - movements between financial assets." *Working paper, Columbia graduate school of business.*

Mendez B, Souza R (2004). "Measuring financial risks with copulas." *International Review of Financial Analysis* **13**(1), 27 - 45.

Panchenko V (2006). "Estimating and evaluating the predictive abilities of semiparametric multivariate models with application to risk management." *Society for Computational Economics,* 382.

Patton A J (2007). "Copula-Based Models for Financial Time Series" in T.G. Andersen, R.A. Davis, J.-P. Kreiss and T. Mikosch (eds.) *Handbook of Financial Time Series*, Springer Verlag.

Patton A J (2006). "Modelling Asymmetric Exchange Rate Dependence." *International Economic Review*, **47**(2), 527 – 556.

Rodriguez J C, (2007). "Measuring financial contagion: A copula approach." *Journal of Empirical Finance* **14**(3), 401 - 423.

Sklar A (1959). "Fonctions de repatitions a n dimensions et leur marges." *Publication de l'Institut de statistique de l'Universite de Paris* **8**, 229 - 231

Serban M, Brockwell A, Lehoczky J, Srivastava S (2007). "Modelling the dynamic dependence structure in multivariate financial time series." *Journal of time series analysis* **28**(5), 763 - 782.

Van den Goorbergh R W J, Genest C, Werker B.J.W (2005). "Bivariate option pricing using dynamic copula models." *Insurance: Mathematics and Economics* **37**(1), 101 – 114.

Wang Z R, Chen X H, Jin Y B, Zhou Y J (2009). "Estimating risk of foreign exchange portfolio: Using VaR and CVaR based on GARCH-EVT-Copula model " *Physica A: Statistical Mechanics and its Applications* 389(21), 4918 - 4928

Yan J (2007). "Enjoy the Joy of Copulas: With a Package copula." *Journal of Statistical Software* **21**(4).