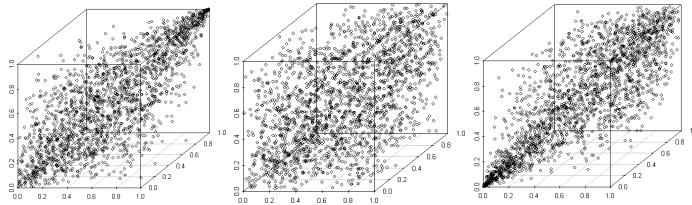


YieldCurve.com



Tools for sampling Multivariate Archimedean Copulas

Mario R. Melchiori CPA[†]
Universidad Nacional del Litoral
Santa Fe - Argentina
March 2006

Abstract

A hurdle for practical implementation of any multivariate Archimedean copula was the absence of an efficient method for generating them. The most frequently used approach named conditional distribution one, involves differentiation step for each dimension of the problem. For this reason, it is not feasible in higher dimension. Marshall and Olkin proposed an alternative method, which is computationally more straightforward than the conditional distribution approach. We present the tools necessary for understand it and use it. We introduce the Laplace Transform and its role in the generation of multivariate Archimedean copulas. In order to cover the gap between the theory and its practical implementation VBA code and R one are provided.[‡]

**The prices do not follow models.
We invent models to describe prices.**

Glyn Holton - www.contingencyanalysis.com

[†] BICA Coop. E.M.Ltda. 25 de Mayo 1774 - Santo tomé - SANTA FE- Argentina - E-mail: Mario.Melchiori@gmail.com
The opinions expressed in this paper are those of the author and do not necessarily reflect views shared by BICA Coop. E.M.Ltda. or its staff.

[‡] The codes are available from the author on request.

Introduction

A hurdle for practical implementation of any multivariate Archimedean copula was the absence of an efficient method for generating them. The most frequently used approach named conditional distribution one, involves differentiation step for each dimension of the problem. For this reason, it is not feasible in higher dimension. Marshall and Olkin proposed an alternative method, which is computationally more straightforward than the conditional distribution approach. A disadvantage is that it requires the generation of an additional variable. For bivariate applications, this means generating 50% more uniform random variables, but for higher dimension that drawback is negligible.

We describe now the algorithm to generate multivariate Archimedean copula. The d -dimensional Archimedean copulas may be written as:

$$C(u_1, \dots, u_d) = \phi^{-1}(\phi(u_1) + \dots + \phi(u_d)),$$

Where ϕ is a decreasing function known as the *generator* of the copula and ϕ^{-1} denotes the inverse of the generator (Frees and Valdez; McNeil et al.). In addition, ϕ^{-1} is equal to the inverse of the **Laplace transform** of a distribution function G on \mathbb{R}^+ satisfying $G(0) = 0$, the following algorithm can be used for simulating from the copula:

Algorithm 1:

1. Simulate d independent uniform variable u_i for $i = 1, \dots, d$
2. Simulate a variable Y with distribution function G such that the **Laplace transform** of G is the inverse of the generator.
3. Define $s_i = \frac{-\ln(u_i)}{Y}$ for $i = 1, \dots, d$
4. Define $X_i = \phi^{-1}(s_i)$ for $i = 1, \dots, d$

Then X_1, \dots, X_d have Archimedean copula dependence structure.

Name	Clayton	Gumbel	Frank
Generator	$\phi(t) = (t^{-\theta} - 1)$	$\phi(t) = (-\ln t)^\theta$	$\phi(t) = -\ln t \frac{e^{-\theta t} - 1}{e^{-\theta} - 1}$
Inverse Generator	$\phi^{-1}(s) = (1 + s)^{-\frac{1}{\theta}}$	$\phi^{-1}(s) = e^{\left(-s^{\frac{1}{\theta}}\right)}$	$\phi^{-1}(s) = -\frac{1}{\theta} \ln[1 - e^{-s}(1 - e^{-\theta})]$
Parameter	$\theta > 0$	$\theta \geq 1$	$\theta \in \mathbb{R} / \{0\}$
Y-Distribution	$Gamma\left(\frac{1}{\theta}, 1\right)$	$Stable(\alpha, \beta, \gamma, \delta)$ $\alpha = \frac{1}{\theta}, \beta = 1, \gamma = \left(\cos\left(\frac{\pi}{2\theta}\right)\right)^\theta, \delta = 0$	Logarithmic series on \mathbb{N}^+ with $\alpha = (1 - e^{-\theta})$
Density of Y	$\frac{1}{\Gamma\left(\frac{1}{\theta}\right)} e^{-y} y^{\frac{(1-\theta)}{\theta}}$	(No closed form is known)	$P[Y = k] = \frac{-1}{\ln(1 - \alpha)} \frac{\alpha^k}{k}$

Table I: Some generators for Archimedean copulas, their inverses and their Laplace transforms. Source: Marshall and Olkin (1988).

Though the density of a $\alpha - stable$ distribution's closed form is not known Nolan proposed the following simulation algorithm for generating random variables $\alpha - stable$ distributed:

Algorithm 2:

1. Simulate an uniform variable $\Theta \sim U\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$
2. Simulate an exponentially distributed variable W with mean 1 independently of Θ .
3. Set $\theta_0 = \arctan(\beta \tan(\pi\alpha/2))/\alpha$.
4. Compute $Z \sim St(\alpha, \beta, 1, 0)$

$$Z = \frac{\sin \alpha(\theta_0 + \Theta)}{(\cos \alpha\theta_0 \cos \Theta)^{1/\alpha}} \left[\frac{\cos(\alpha\theta_0 + (\alpha - 1)\Theta)}{W} \right]^{\left(\frac{1 - \alpha}{\alpha}\right)} \quad \alpha \neq 1$$

$$Z = \frac{2}{\pi} \left[\left(\frac{2}{\pi} + \beta\Theta \right) \tan \Theta - \beta \ln \left[\frac{\frac{\pi}{2} W \cos \Theta}{\frac{\pi}{2} + \beta\Theta} \right] \right] \quad \alpha = 1$$

5. Compute $X \sim St(\alpha, \beta, \gamma, \delta)$:

$$\begin{aligned} X &= \gamma Z + \delta & \alpha \neq 1 \\ X &= \gamma Z + \left(\delta + \beta \frac{2}{\pi} \gamma \ln(\gamma) \right) & \alpha = 1 \end{aligned}$$

Below, we use Kemp's second accelerated generator of Logarithmic Distribution random variables from Luc Devroye's book titled "Non-Uniform Random Variate Generation" chapter 10, page 548, freely available on <http://cgm.cs.mcgill.ca/~luc/rnbookindex.html>.

Algorithm 3:

Set $c = \ln(1 - \alpha)$

Simulate an uniform variable $V[0,1]$.

IF $V \geq \alpha$ then set $X = 1$

ELSE

Simulate an uniform variable $U[0,1]$.

Set $q = 1 - e^{cU}$

CASE

$V \leq q^2$: set $X = \text{int} \left[1 + \frac{\ln(V)}{\ln(q)} \right]$.

$q^2 < V \leq q$: set $X = 1$.

$V > q$: set $X = 2$.

X is Logarithmic Series distributed.

No other algorithm is necessary because the most of software have a built-in function that generates random variables Gamma distributed, such as **GammaInv** function on Excel and **rgamma** on R.

Laplace Transform

The **Laplace Transform** of a function $f(t)$ is denoted by $\mathcal{L}[f(t)]$ and defined by:

$$\mathcal{L}[f(t)] = \int_0^{\infty} f(t) e^{-st} dt$$

The **Laplace transform** $\mathcal{L}[f(t)]$ of $f(t)$ is, therefore, a function of the variable s and is commonly denoted by $f(s)$. The inverse Laplace transform is written as:

$$\mathcal{L}^{-1}[f(s)] = f(t).$$

Table of **Laplace Transform** pairs, which are of interest for our work:

Function $f(t)$	Laplace Transform $f(s) = \int_0^{\infty} f(t)e^{-st} dt$
1 for $t > 0$	$\frac{1}{s}$
e^{gt}	$\frac{1}{s-g}$ for $s > g$
$\frac{1}{\Gamma\left(\frac{1}{\theta}\right)} e^{-t} t^{\frac{(1-\theta)}{\theta}}$	$(1+s)^{-\frac{1}{\theta}}$ for $\theta > 0$
$P[T = t] = \frac{-1}{\ln(1-\alpha)} \frac{\alpha^t}{t}$	$-\frac{1}{\theta} \ln[1 - e^{-s}(1 - e^{-\theta})]$ for $\alpha = (1 - e^{-\theta})$
(No closed form is known)	$e^{\left(-s^{\frac{1}{\theta}}\right)}$ for $\theta \geq 1$

Table II - Laplace Transform pairs

From an economic point of view we immediately recognize the **Laplace transform** as the *present value* of a stream of returns $f(t)$ at the interest rate s . The present value of cash flows of \$1, $f(t) = 1$, paid annually and perpetually at the continuous interest rate s is $\mathcal{L}[f(s)] = \int_0^{\infty} 1e^{-st} dt = \frac{1}{s}$. On the other hand, the present discounted value at the

continuous interest rate s of a cash flow growing at the rate g , $f(t) = e^{gt}$, is $\mathcal{L}[f(s)] = \int_0^{\infty} e^{gt} e^{-st} dt = \frac{1}{s-g}$. If

$f(t)$ is continuous and differentiable at all $t \geq 0$, then it is possible to recover $f(t)$ from $\mathcal{L}[f(s)]$ through the inverse transformation:

$$\mathcal{L}^{-1}[f(s)] = f(t) = \lim_{b \rightarrow \infty} \frac{1}{2\pi i} \int_{a-ib}^{a+ib} e^{st} \mathcal{L}[f(s)] ds, a \geq \alpha.$$

This has the economic meaning that if we know the present discounted value of a stream of returns at every interest rate, we can recover the whole pattern of the stream of returns.

Experiment I

We conduct an experiment for seeing practically in what manner, by knowing the Laplace transform of a function, we can recover it. We will do it here, but you can reproduce it on your own computer. For doing this, we need an approach that implements numerically the inverse transformation $\mathcal{L}^{-1}[f(s)]$. We will use the Euler¹ method. Readers interested in deepening this subject should consult the following link:

<http://www.columbia.edu/~ww2040/LaplaceInversionJoC95.pdf>

¹ In **Appendix A**, we provide a VBA code that implements numerically the inverse **Laplace transform** by employing the Euler method. The **Laplace transform** is specified by the variable **Fs** in the function **Rf**. Inserting different transforms here can solve different problems.

The experiment consists of computing the flow of returns by knowing its net present value using the inverse **Laplace transform**. As we already saw, the net present value of a stream of returns that grows to the continuous rate g , represented by the function $f(t) = e^{gt}$, discounted to the continuous interest rate s to be $\frac{1}{s-g}$. Also $\frac{1}{s-g}$ is the Laplace Transform of the function $f(t) = e^{gt}$ as well. So that, we can calculate the stream of returns that grows at rate g^2 discounted to the continuous interest rate s using e^{gt} or employing the Inverse **Laplace Transform** of $\frac{1}{s-g}$.

In the VBA alluded to the **Laplace transform** $\frac{1}{s-g}$, is specified by the variable **Fs** in the function **Rf**, in complex notation. The VBA should look as follow:

```
Function Rf(ByVal X, ByVal Y) As Double
s = imsum(X, improduct("i", Y))
g=0.02
Fs = imdiv(1, (imsum(s, - g)))
Rfs = imreal(Fs)
Rf = Rfs
End Function
```

So in Excel's language e^{gt} is represented as **=Exp(0.02*A1)** for $t = 1$, **=Exp(0.02*A2)** for $t = 2$ and so on. On the other hand, the Inverse **Laplace Transform** of $\frac{1}{s-g}$ as **=InverseLaplace(A1)** for $t = 1$. **=InverseLaplace(A2)** for $t = 2$ and so on.

t	A	B	C
1	1	1.020201	1.020201
2	2	1.040811	1.040811
3	3	1.061837	1.061837
4	4	1.083287	1.083287
5	5	1.105171	1.105171
6	6	1.127497	1.127497
7	7	1.150274	1.150274
8	8	1.173511	1.173511
9	9	1.197217	1.197217
10	10	1.221403	1.221403
11	11	1.246077	1.246077
12	12	1.271249	1.271249
13	13	1.296930	1.296930
14	14	1.323130	1.323130
15	15	1.349859	1.349859
16	16	1.377128	1.377128
17	17	1.404948	1.404948
18	18	1.433329	1.433329
19	19	1.462285	1.462285
20	20	1.491825	1.491825
21	21	1.521962	1.521962
22	22	1.552707	1.552707
23	23	1.584074	1.584074
24	24	1.616074	1.616074
25	25	1.648721	1.648721

² In this experiment the stream of returns grows to the continuous rate of the 0.02 ($g = 0.02$)

Experiment II

The previous experiment was a warm-up for this one, which has more relation with our research. The point 2 of the [Algorithm 1](#) requires simulating a variable Y with distribution function G such that the **Laplace transform** of G is the inverse of the generator. In Table I, we can see in the Clayton case that the probability density function G is $Gamma\left(\frac{1}{\theta}, 1\right)$. So that, the inverse **Laplace transform** of the inverse generator $(1+s)^{-\frac{1}{\theta}}$ for $\theta > 0^3$ equals to the Probability Density Function of a $Gamma\left(\frac{1}{\theta}, 1\right)$ distribution. For doing this in the VBA alluded to, the **Laplace transform** $(1+s)^{-\frac{1}{\theta}}$, is specified by the variable **Fs** in the function **Rf**, in complex notation. The VBA should look as follow:

```
Function Rf(ByVal X, ByVal Y) As Double
    s = imsum(X, improduct("i", Y))
    theta= 1.84
    Fs = impower(imsum(1, s), -1 / theta)
    Rfs = imreal(Fs)
    Rf = Rfs
End Function
```

So that the **Laplace transform** $(1+s)^{-\frac{1}{\theta}}$ in Excel's language is represented by the function **=InverseLaplace(A1)** for $t = 1$, **=InverseLaplace(A2)** for $t = 2$ and so on. Also, the Probability Density Function of a

$Gamma\left(\frac{1}{\theta}, 1\right)$ distribution equals to $\frac{1}{\Gamma\left(\frac{1}{\theta}\right)} e^{-t} t^{\frac{(1-\theta)}{\theta}}$ or in Excel's language **=1/(Exp(Gamma.Ln(1/1.84)))*Exp(-**

A1)*A1^ ((1-1.84)/1.84) or =GammaDist(A1,1/1.84,1,0) for $t = 1$ and, **=1/(Exp(Gamma.Ln(1/1.84)))*Exp(-**
A2)*A2^ ((1-1.84)/1.84) or =GammaDist(A2,1/1.84,1,0) for $t = 2$ and so on.

t	B	C	D
1	0.225048	0.225048	0.225048
2	0.060333	0.060333	0.060333
3	0.018445	0.018445	0.018445
4	0.005950	0.005950	0.005950
5	0.001977	0.001977	0.001977
6	0.000669	0.000669	0.000669
7	0.000229	0.000229	0.000229
8	0.000079	0.000079	0.000079
9	0.000028	0.000028	0.000028
10	0.000010	0.000010	0.000010
11	0.000003	0.000003	0.000003
12	0.000001	0.000001	0.000001
13	0.000000	0.000000	0.000000
14	0.000000	0.000000	0.000000
15	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000
17	0.000000	0.000000	0.000000
18	0.000000	0.000000	0.000000
19	0.000000	0.000000	0.000000
20	0.000000	0.000000	0.000000

³ In this example the variable θ equal to 1.84.

We invite to readers to conduct a third experiment that computes the probability density function in the Frank case

using both the function $P[Y = k] = \frac{-1}{\ln(1 - \alpha)} \frac{\alpha^k}{k}$ and the inverse **Laplace Transform** of

$-\frac{1}{\theta} \ln[1 - e^{-s} (1 - e^{-\theta})]$ for $\alpha = (1 - e^{-\theta})$.

Conclusions

There is clear evidence that equity returns have unconditional fat tails, to wit, the extreme events are more probable than anticipated by normal distribution, not only in marginal but also in higher dimensions. This is important both for market risk models as credit risk one, where equity returns are used as a proxy for asset returns that follow a multivariate normal distribution, and, therefore, default times have a multivariate normal dependence structure as well. Other than normal distribution should be used both in marginal as joint distributions. To overcome these pitfalls, the concept of copula emerges. A hurdle for practical implementation of any multivariate Archimedean copula was the absence of an efficient method for generating them. The most frequently used approach named conditional distribution one, involves differentiation step for each dimension of the problem. For this reason, it is not feasible in higher dimension. Marshall and Olkin proposed an alternative method, which is computationally more straightforward than the conditional distribution approach. We present the tools necessary for understand it and use it. We introduce the Laplace Transform and its role in the generation of multivariate Archimedean copulas. In order to cover the gap between the theory and its practical implementation VBA code and R one are provided.

References

Abate, Joseph and Whitt Ward 1995 *Numerical Inversion of Laplace Transforms of Probability Distributions*. ORSA Journal on Computing, vol. 7, 1995, pp. 36.

(<http://www.columbia.edu/~ww2040/LaplaceInversionJoC95.pdf>)

Devroye, Luc (1986). *Non-uniform random variate generation*. Springer, Berlin, Heidelberg, New York. (<http://cgm.cs.mcgill.ca/~luc/rnbookindex.html>)

Duncan K. Foley (2005) *Laplace Transform* – Class Notes
(<http://homepage.newschool.edu/~foleyd/GECO6289/laplace.pdf>)

Embrechts, P., A. J. McNeil and D. Straumann (1999) *Correlation and Dependence in Risk Management: Properties and Pitfalls*. (<http://www.math.ethz.ch/~strauman/preprints/pitfalls.pdf>)

Frees, E. W. and Valdez, E. A. (1998): *Understanding relationships using copulas*, North American Actuarial Journal, 2, pp. 1-25.

Kjersti Aas (December 2004) *Modelling the dependence structure of financial assets: A survey of four copulas* . Norwegian Computing Center (<http://www.nr.no/files/samba/bff/SAMBA2204b.pdf>)

Lindskog, F. (2000) *Modeling Dependence with Copulas* ETH Zurich
(<http://www.math.ethz.ch/~mcneil/ftp/DependenceWithCopulas.pdf>)

Marshall, Albert W. and Ingram Olkin (1988) *Families of multivariate distributions*. Journal of the American Statistical Association, 83, 834-841.

Melchiori, Mario R. (2003) *Which Archimedean Copula is the right one?* Universidad Nacional del Litoral Argentina (www.riskglossary.com/papers/Copula_carta.PDF)

Nolan, J.P. (Forthcoming) *Stable Distributions: Models for Heavy Tailed Data*.
(<http://academic2.american.edu/~jpnolan/stable/chap1.pdf>)

Schönbucher, Philipp J. (August 2002) *Taken to the Limit: Simple and Not-so-simple Loan Loss Distributions* – Bonn University (http://www.defaultrisk.com/pdf_files/Taken_To_the_Limit-Smpl_nNt-s-mpl_Ln_Lss_Dstrbtms.pdf)

Appendix A

Abate, Joseph and Whitt Ward 1995 *Numerical Inversion of Laplace Transforms of Probability Distributions*. ORSA Journal on Computing, vol. 7, 1995, pp. 36.
 (<http://www.columbia.edu/~ww2040/LaplaceInversionJoC95.pdf>)

Function InverseLaplace(t As Double) As Double

Const Pi = 3.14159265358979

Dim SU(13), C(12)

m = 11

For n = 0 To m

C(n + 1) = Application.WorksheetFunction.Combin(m, n)

Next

A = 18.4

Ntr = 15

u = Exp(A / 2) / t

X = A / (2 * t)

h = Pi / t

Sum = Rf(X, 0) / 2

For n = 1 To Ntr

Y = n * h

Sum = Sum + (-1) ^ n * Rf(X, Y)

Next

SU(1) = Sum

For k = 1 To 12

n = Ntr + k

Y = n * h

SU(k + 1) = SU(k) + (-1) ^ n * Rf(X, Y)

Next

Avgsu = 0

Avgsul = 0

For j = 1 To 12

Avgsu = Avgsu + C(j) * SU(j)

Avgsul = Avgsul + C(j) * SU(j + 1)

Next

Fun = u * Avgsu / 2048

Fun1 = u * Avgsul / 2048

InverseLaplace = Fun1

errt = Abs(Fun - Fun1) / 2

End Function

Function Rf(ByVal X, ByVal Y) As Double

s = imsum(X, improduct("i", Y))

Fs = imdiv(1, (imsum(s, -0.02)))

Rfs = imreal(Fs)

Rf = Rfs

End Function
