Non-recombining trees for pricing of multi-variate options

Michal Kaut
michal.kaut@iot.ntnu.no*

Stein W. Wallace
stein.w.wallace@himolde.no†

March 24, 2003

Abstract

This paper presents a method for pricing multi-variate options using non-recombining trees. The trees are generated using a moment-based approach, so no distributional assumptions are needed. To apply the method, the user has to know the properties (moments and correlations) of the risk-neutral distributions of returns of all the assets the options depend on.

European options are priced on single-period trees, the path-dependent options on multi-period trees. We provide exact formulas for generation of trees without any inter-period dependency of the return distributions. With inter-period dependency, we provide only an ex-post correction procedure.

Keywords: non-recombining trees, option pricing, multivariate options

1 Introduction

Currently, the two most common approaches for option pricing are the Black & Scholes continuous-time framework, and Cox, Ross, Rubinstein discrete-time approach using binomial (recombining) trees. While the former is an exact formula for European options, the latter is an approximate approach used mainly for path-dependent options (for example American and Bermudian options). Both of the approaches have, at least in their "classical" versions, the disadvantage of assuming that the asset returns are normally distributed. Recently, there have appeared attempts to drop the normality assumption, both in the theoretical B&S-like context (see for example [12, 24]), and in the context of binomial or trinomial recombining trees ([21, 5, 6, 22]).

Yet, most of the methods can price only options depending on one asset. In the recent years, however, several types of multi-variate options (also called basket options) have appeared. An example is the option to buy the better of two indices. Methods for pricing such options were also presented, yet most of them are based on the normality assumption. Beyond normality, the pricing tools are still scarce: [3] provides a theoretical derivation for pricing of bivariate options, while [18, 19, 20] price bivariate options using binomial trees. All of these methods are based on copulas as a description of the multi-variate distribution.

In this paper, we provide a method for pricing general multi-variate options, using non-recombining multi-period trees. We do not make any distributional assumptions, and use a

*Dept. of Industrial Economics and Technology Management, NTNU, N-7491 Trondheim, Norway
†Molde University College, Postboks 2110, N-6402 Molde, Norway

moment-matching approach for construction of the trees. Hence, the method may be useful in contexts where a description of the risk-neutral distribution by moments and correlations is more natural than using copulas – for example because we do not have enough information/data to estimate the copula, or we want to avoid the various assumptions (on type of copula, or on the density function of the marginals), needed for the copula estimation.

In addition to the multi-period algorithm, we show how the moment-based approach provides an easy way to price multi-variate European options, using large one-period trees.

Since we use non-recombining trees, we face the usual "curse of dimensionality", i.e. the exponential grow of the tree with the number of periods. However, with the presented algorithm we can process a tree with a quarter of billion terminal nodes in approximately half a minute on a 1GHz PC. This number of nodes corresponds to a 12-period tree with five branches per node—if there is a need for a significantly finer time-discretization, the method is thus not applicable.

In the binomial trees, the risk-neutral distribution (probabilities) are computed directly from the prices and the risk-free rate. For non-recombining trees with more than two branches, however, this is no longer possible and we need to know the risk-neutral distribution in advance. Typically, the risk-neutral distribution is estimated from historical option prices. In this paper, we assume that we have already made the estimation, and thus know the risk-neutral distribution—or, more precisely, its moments and correlations. For information about recovering the risk-neutral distribution from historical option prices, see for example [11, 7, 19, 20].

The moment-based approach is simpler than the copula-based approaches mentioned above. However, our results show that there is a price for the simplicity: For some of the basket options, the quality of the multi-variate structure of the return distribution becomes very important. In such cases, correlation coefficients may not be enough to describe the dependencies, and the resulting prices become only approximations. The method is thus best suited for cases where either the copula approach is not applicable, or only an approximation of the option prices is needed. As an example of the latter, consider the case of real options, where the risk-neutral distribution often is not known perfectly, so only an approximation of the price is obtained regardless of the method used.

The rest of the paper is organised as follows: in Section 2 we show how to price multi-variate European options. The rest of the paper concerns multi-period trees for pricing of path-dependent options: Section 3 presents an efficient generating/pricing procedure for multi-period trees, including a numerical results. In Section 4, we discuss ways of controlling the distribution in the final period of a multi-period tree by adjusting the distributions of the one-period subtrees it consists of. Section 5 presents and discusses stability of the obtained prices with respect to the choice of the tree, and, finally, Section 5 concludes the paper.

2 European options

European options can be exercised only at a one given time, so their prices are determined by the distribution of prices at the exercise time, together with the initial parameters. Hence, when we price an European option using a tree, only the root and the distribution of the last period affect the price. It is thus enough to use a single-period tree.

As we will see later, it is not trivial to generate a multi-period tree with a given distribution of the last period, especially in the multi-variate case. On the other hand, there are several methods for generating a single-period trees with a specified distribution, see for example [23, 1, 13, 14, 10, 15, 8]. We use a moment-matching method from [10], which gives us control over the first four moments and the correlations of the random variables. Once we have the tree for the risk-neutral distribution, the price of the option is simply the expected value of the distribution, i.e. weighted sum of the prices in the tree.

2.1 Example

The following example comes from [20]: We consider two different one-month options, both based on S&P 500 and DAX 30 indices. The risk-neutral distribution of the returns (defined as $r_t = {}^{V_t}/_{V_{t-1}} - 1$) was estimated, using data from December 1999. The properties of the distribution are in Table 1.

Table 1: Properties of the risk-neutral distribution of returns from [20]

	S&P 500	DAX 30
mean (annualised)	7.30%	4.28%
standard deviation (annualised)	22.34%	29.82%
skewness	-0.58	-0.36
kurtosis	3.80	3.38
correlation	0.5192	

The two options considered are an underperformance option, which gives the buyer a right to buy the worse of the two indices, and an outperformance option that gives a right to buy the better of the two indices. In both cases, the strike price is equal to the spot price, and we buy indices for \$100.

We use a moment-matching scenario generation method from [10] to generate the trees. To achieve reasonable stability, we have used trees with 10.000 scenarios—it still takes less than a second to both generate the tree and price the options. Using 10 runs, we have estimated the price of the underperformance option as $\$1.72 \pm \$.025$, and the outperformance option as $\$4.74 \pm \$.025$.

In [20], the prices are \$1.52 and \$4.86, respectively, making our prices 13% and 2% different. The difference comes from differences in the bi-variate distributions: we control only the correlation coefficient, which is not enough to get the same structure of the bi-variate distribution as in [20]. It is also confirmed by the fact that the error is much bigger for the underperformance option, which is more sensitive to the bi-variate dependency structure – see [20] for a detailed discussion.

The difference between our prices and the prices from [20] show that the correlation alone is not enough to describe the bi-variate dependency structure. Hence, when we need a more accurate estimation of the price of an multi-variate option, we have to use another method. One possibility is using a copula, provided we have enough data/information to estimate it. See [17, 4] for an introduction to copulas, or [18, 19, 20, 3] for copula-based option-pricing techniques.

3 Generating/pricing algorithm for a multi-period tree

In this section we describe the algorithm for generating a multi-period tree and pricing an option based on this tree. We take the distributions for given—the problem of finding the right distributions is discussed in the next section.

When describing a multi-period tree, we use the following conventions: We call the points in time represented by nodes *stages*, and the time intervals between them *periods*. Hence, a tree with p periods have $p+1$ stages. The root of the tree is indexed as stage zero, so period t is an interval between stages $t-1$ and t. A distribution of random variables with outcomes at stage t can thus be referred to as a distribution *in* period t, or as a distribution *at* stage t.

In a multi-period tree we have to distinguish between conditional and unconditional distributions. For example, with using the tree in Figure 1, nodes $\{3, 4, 5\}$ represent a distribution of the second period (or a distribution at stage 2), conditional on the values at node 1. The *conditional probabilities* of node $\{3, 4, 5\}$ sum up to one. On the other hand, the *unconditional distribution* of the second period is represented by nodes $\{3, 4, 5, 15, 16, 17\}$. Specifically, the unconditional distribution of the last period of a tree is referred to as the *final-stage distribution*.

In addition, by generating a subtree of node n we understand generating a one-period tree rooted at node n, i.e. finding the outcomes at the direct successors (children) of node n, and the probabilities thereof. For example, generating a subtree of node 1 from Figure 1 means finding the values of the random variables at nodes 3, 4 and 5, as well as the probabilities of these nodes.

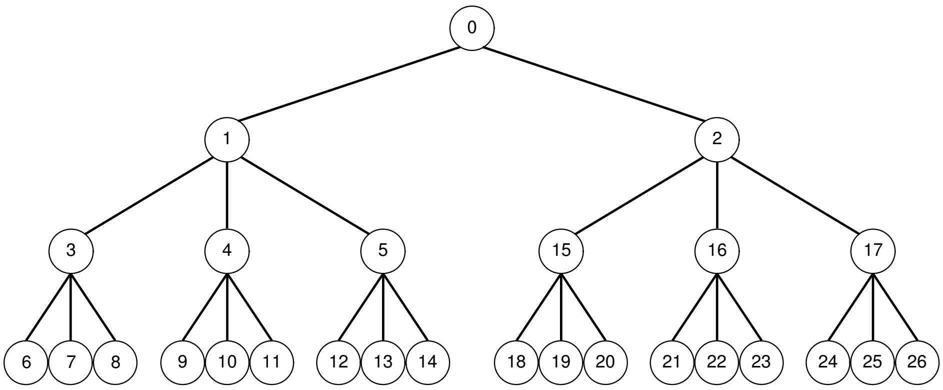

Figure 1: Example of a multi-period tree

The multi-period tree is constructed by one-period subtrees, starting in the root. The subtrees are generated using a moment-matching approach from [9], so every subtree is constructed to have specified first four moments (mean, variance, skewness, kurtosis) and correlations. Generating the multi-period tree by subtrees allows us to specify an inter-period dependency: we create a subtree, move to one of the nodes, update the distribution of the consecutive subtree (based on the original distribution and the outcomes at the node and its predecessors), and generate the subtree with the updated distribution. See Appendix B for an example of the update formulas, in the case of first order autocorrelation.

There is an important difference between generating a scenario tree for an optimization model and generating a tree for the option pricing: while for the optimization model we need to store the whole tree prior to the optimization, the option pricing can be done simultaneously with the tree-generation. In the following text we explain the pricing procedure and show that it decreases the storage requirements enormously.

The important property is that the price at every node depends only on the outcomes at the node and the prices of its direct successors (children). Hence, once we have computed the price at a given node, we can discard all its successors. When we use a post-order traversal of the tree, we need to store at most one path from the root to the leafs. The number of trees stored at any moment is thus at most equal to the number of periods p.

As an example, consider generating/pricing the tree from Figure 1. Table 2 describes the procedure step-by-step, showing for every step nodes that are generated, nodes in which we calculate the price, and nodes that can be discarded. Figure 2 represents the same procedure graphically.

Table 2: Step-by-step generation and pricing of the tree from Figure 1

	new values at nodes		dropped nodes	nodes stored after the step	
step	outcomes	prices	outcomes & prices	outcomes	prices
1	1–2			0,1–2	
2	3–5			0,1–2,3–5	
3	6–8	6–8,3	6–8	0,1–2,3–5	3
4	9–11	9–11,4	9–11	0,1–2,3–5	3–4
5	12–14	12–14,5,1	12–14,3–5	0,1–2	1
6	15–17			0,1–2,15–17	1
7	18–20	18–20,15	18–20	0,1–2,15–17	1,15
8	21–23	21–23,16	21–23	0,1–2,15–17	1,15–16
9	24–26	24–26,17,2,0	24–26,15–17,1–2	0	0

A complete generating/pricing algorithm is presented in Figure 3. Note that the algorithm is written in a recursive form only for the sake of simplicity: since recursive algorithms are usually slower than their non-recursive versions, the actual implementation is not recursive (and therefore significantly more complicated than the presented version). The formulation also assumes that all the structures (outcomes and prices) are local to the function $value(n)$, and are therefore disposed of automatically by the system at the moment of the departure from the function.

3.1 Pre-generation of subtrees

Since the size of the multi-period tree grows exponentially with the number of periods p, the efficiency of the whole procedure becomes crucial. The most time-consuming task of the algorithm presented in Figure 3 is the generation of subtrees, because the rest consists of simple algebraic operations. In some cases we can, however, avoid generating all the subtrees, speeding thus the procedure substantially.

The easiest case is when all the subtrees have the same number of branches and the same distributions, i.e. the case with no inter-period dependency. In this case a single subtree can be used throughout the whole multi-period tree. This subtree can be generated prior (or at the start of) the generating/pricing algorithm, so there would be no scenario generation during the rest of the algorithm.

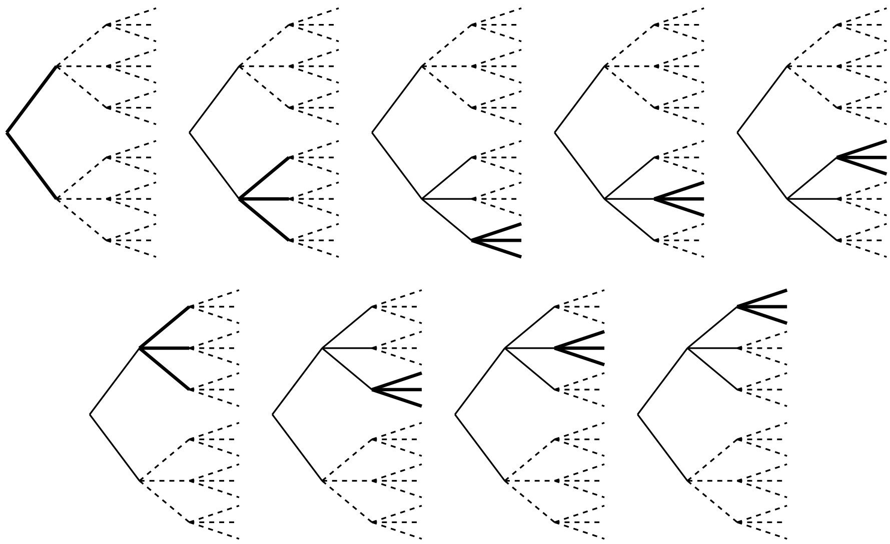

Figure 2: Example of step-by-step generation of a multi-period tree. The bold arcs represent the currently generated subtree, the other subtrees kept in the memory have solid arcs. Dashed arcs represents part of the multi-period tree that is not being processed at the moment.

If we want the number of branches of the subtrees to differ for every period (typically starting with more branches and decrease the number for later periods), we still only need one subtree for every desired size, as long as the distributions are kept constant, i.e. as long as we do not have inter-period dependencies.

When we introduce inter-period dependencies (for example mean-reversion or autocorrelation), the situation becomes more complicated. However, we may still avoid "on the fly" generation of the subtrees in the case when the dependency rules change only the means and variances of the distributions: for example mean-reversion or volatility clumping. In this case we can use a linear transformation to update the distribution: if we have a pre-generated subtree $\tilde{\boldsymbol{X}}$ with zero means and variances of one, then $\boldsymbol{\sigma}\tilde{\boldsymbol{X}} + \boldsymbol{\mu}$ gives a subtree with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$, while the higher moments and the correlations remain unchanged. The linear transformation is much cheaper than generating a new tree, and the total running time will be of the same magnitude as the case without inter-period dependencies.

The most difficult case is when we introduce inter-period rules that update also the higher moments and/or the correlations of the subtree distribution. If we want to avoid generating all the subtrees even in this case, we have to use an approximate approach: we pre-generate a set of subtrees with different combinations of the properties that are updated, and during the generating/pricing procedure choose the one closest to the desired properties. For example, discretizing the interval of possible values of the correlation, the skewness, and the kurtosis to 10 points each gives $10 \times 10 \times 10 = 1000$ subtrees. These have to be generated in advance. Since a typical multi-period tree will have millions of subtrees, the speed-up of the whole generating/pricing process is substantial.

```
function value(n) {
  if n is not a leaf {
    generate subtree of n = {x(m), p(m) | m ∈ C(n)};
    for all children m ∈ C(n) do
      v(m) = value(m);
    value = max{ f_e(x(n)), Σ_{m∈C} v(m)p(m) };
  }
  else
    value = f_e(x(n));
}
```

Figure 3: The generating/pricing algorithm, using a recursive post-order traversal. For any node n of the tree, $\mathbf{C}(n)$ is a set of all children of n, $\boldsymbol{x}(n)$ are prices of the assets at n, $p(n)$ is a probability of the node n, and $f_e(\boldsymbol{x}(n))$ is the profit of immediate exercise of the option at node n. The price of the option is then found as $price = value(root)$.

3.2 Speed test

The tests were run on an `Intel Pentium III` machine with 996.76 MHz processor and 256 MB RAM, running `Windows 2000`™. The algorithm was implemented using the `C` programming language and compiled with `GNU C++` compiler.[1]

In the tests, we created 12-period trees and priced the two options from Section 2.1 on them. Each option was priced both as an European and an American call. In these tests we assume that all the subtrees have the same distributions and sizes, so we need to generate only one subtree and use it throughout the whole tree. The time for generating is not included in the reported time, but it typically is only a fraction of a second. The results are presented in Table 3.

Table 3: Time to generate a 12-period tree and price two options on the tree

# branches per subtree	subtrees	terminal nodes	time
5	61.035.156	244.110.625	32 sec
6	435.356.467	2.176.782.336	5 min

In these tests, we have used the same definition of returns as in Section 2.1: $r_t = {}^{V_t}/_{V_{t-1}} - 1$, so $V_t = (1+r_t)V_{t-1}$. If we use an alternative definitions $r_t = \ln(V_t) - \ln(V_{t-1})$ and $V_t = e^{r_t} V_{t-1}$, the code runs about three times slower, because the ln() and exp() functions take more CPU time than division and multiplication.

4 Controlling the final-stage distribution

There are (at least) two possible approaches in generating multi-period scenario trees: we may be concerned with the distributions of the one-period subtrees and with the inter-period dependencies, or we may focus on the final-stage distribution. Since we are applying the scenario trees for option pricing, we take the latter approach—having a correct final-stage

[1] The code was also compiled with `Visual C++`, in which case it runs about 5% slower.

distribution is crucial for comparing with other models, where the final-stage distribution is often the only specified property.

Like in the previous section, we describe the final-stage distribution by its moments and correlations, and control the subtrees by the same means. Hence, the problem is to find the right moments for the one-period subtrees, given the final-stage distribution, the number of periods and possibly the inter-period dependency rules and the outcomes of the predecessors of the subtree.

First, we focus on the case with no inter-period dependency, i.e. the case where the distribution of a subtree of a given node does not depend on the outcomes at the predecessors of this node. This corresponds to the "traditional" binary trees that have independent periods as well. In this case, we provide formulas to find moments and correlations of the subtrees that result in the final-stage distribution with given properties.

The assumption of independent periods is, unfortunately, not very realistic for financial applications, where effects like autocorrelation and mean-reversion are commonplace. In the second part of this section we thus look at trees with inter-period dependencies. In this case, however, we did not manage to find corresponding formulas for controlling the subtree distributions. Instead, we present a procedure that generates a tree and thereafter corrects the means and variances to obtain the required unconditional distribution (distribution over all nodes in the same stage). Used in a loop, the procedure corrects the means and variances in the whole tree.

4.1 Independent periods

In this section we present formulas for the moments and the correlations of distribution of the subtrees that will results in a multi-period tree with the correct moments and correlations at the final stage. The formulas use an assumption that all the subtrees have the same distributions, i.e. that there is no iter-period dependency. A detailed derivation of the formulas is in Appendix A.

Throughout the section we use the following notation: in a multi-period tree with p periods we denote by $\mu, \sigma^2, \gamma, \delta$ the first four moments[2] of the final-stage distribution, and $\mu_p, \sigma_p^2, \gamma_p, \delta_p$ the first four moments of the corresponding subtrees.[3] For two random variables $\tilde{X}$ and $\tilde{Y}$, we extend the notation by ρ for their final-stage correlation, and ρ_p for the correlation in the subtrees. In addition, we add upper indices X and Y to the moments defined above, in order to distinguished between moments of $\tilde{X}$ and $\tilde{Y}$ where needed.

The formulas depend on the nature of the underlying stochastic process: It may be either discrete, updating values only at the stages of the tree, or it may be a process with continuous compound.

Formulas for discrete processes

By a discrete process we understand a process $\{\tilde{\boldsymbol{V}}_t\}$ evolving as

$$\tilde{V}_t = (1 + \tilde{X}_t)\tilde{V}_{t-1}.$$

[2] The standard notation is γ_1 for skewness and γ_2 for normalised kurtosis. Since we need indices for the moments, we have introduced a new notation $\gamma = \gamma_1$ and $\delta = \gamma_2 + 3$.

[3] The index p thus means that these properties are implied by their respective final-stage values, and the number of periods p.

The final-stage return at the tree is then given as

$$\prod_{k=1}^{p}(1+\tilde{X}_k)-1.$$

Then, to get a tree with μ, σ^2, γ, δ, and ρ, we have to use subtrees with:

$$\mu_p = (1+\mu)^{1/p}-1 \tag{1a}$$

$$\sigma_p^2 = \left((1+\mu)^2+\sigma^2\right)^{1/p}-(1+\mu_p)^2 \tag{1b}$$

$$\gamma_p = \frac{1}{\sigma_p^3}\left[\left((1+\mu)^3+3(1+\mu)\sigma^2+\sigma^3\gamma\right)^{1/p}-(1+\mu_p)^3-3(1+\mu_p)\sigma_p^2\right] \tag{1c}$$

$$\begin{aligned}\delta_p = \frac{1}{\sigma_p^4}\Big[&\left((1+\mu)^4+6(1+\mu)^2\sigma^2+4(1+\mu)\sigma^3\gamma+\sigma^4\delta\right)^{1/p}\\ &-(1+\mu_p)^4-6(1+\mu_p)^2\sigma_p^2-4(1+\mu_p)\sigma_p^3\gamma_p\Big]\end{aligned} \tag{1d}$$

$$\rho_p = \frac{1}{\sigma_p^X\sigma_p^Y}\left[\left((1+\mu^X)(1+\mu^Y)+\sigma^X\sigma^Y\rho\right)^{1/p}-(1+\mu_p^X)(1+\mu_p^Y)\right] \tag{1e}$$

This, for example, means that in order to get the standard normal distribution in the final stage of a p-periodic tree ($\mu=0,\ \sigma^2=1,\ \gamma=0,\ \delta=3$), the moments of the subtrees have to be

$$\begin{aligned}\gamma_p &= \frac{2^{1/p}-2}{\sqrt{2^{1/p}-1}} \quad \xrightarrow[p\to\infty]{} -\infty\\ \delta_p &= \frac{1}{\sigma_p^4}\left(10^{1/p}-4\cdot 4^{1/p}+6\cdot 2^{1/p}-3\right) \quad \xrightarrow[p\to\infty]{} \infty.\end{aligned} \tag{2}$$

If we use subtrees with the correct mean and variance, but keep them normal (i.e. $\gamma=0$ and $\delta=3$), the resulting p-periodic tree would have skewness and kurtosis:

$$\begin{aligned}\gamma &= \left(3\cdot 2^{1/p}-2\right)^p-4 \quad \xrightarrow[p\to\infty]{} 4\\ \delta &= \left(3\cdot 4^{1/p}-2\right)^p-4\left(3\cdot 2^{1/p}-2\right)^p+9 \quad \xrightarrow[p\to\infty]{} 41\end{aligned} \tag{3}$$

Note also that zero correlation in the subtrees leads to a zero correlation in the final-stage distribution. Generally, the correlation is very stable, i.e. the difference between ρ and ρ_p is typically small.

Formulas for continuous processes

By a continuous process we understand a process $\{\tilde{\boldsymbol{V}}_t\}$ evolving as

$$\tilde{V}_t = e^{\tilde{X}_t}\tilde{V}_{t-1}.$$

The final-stage return is then given as

$$\ln\left(\frac{\tilde{V}_p}{V_0}\right) = \ln(\tilde{V}_p)-\ln(V_0) = \ln\left(e^{\sum_{k=1}^{p}\tilde{X}_k}V_0\right)-\ln(V_0) = \sum_{k=1}^{p}\tilde{X}_k.$$

Then, to get a tree with μ, σ^2, γ, δ, and ρ, we have to use subtrees with:

$$\mu_p = p^{-1}\mu \qquad\qquad \gamma_p = \sqrt{p}\,\gamma \tag{4a}$$

$$\sigma_p^2 = p^{-1}\sigma^2 \qquad\qquad \delta_p = p(\delta - 3) + 3 \tag{4b}$$

$$\rho_p = \rho \tag{4c}$$

Unlike the discrete case, here we know that a sum of independent normal distributions is again normally distributed, so $(\gamma_p, \delta_p) = (0, 3)$ implies $(\gamma, \delta) = (0, 3)$, and vice versa. In addition, for any fixed γ_p and δ_p, we know that the final-stage distribution converges to the normal distribution as p goes to infinity, hence $\gamma \xrightarrow[p\to\infty]{} 0$ and $\delta \xrightarrow[p\to\infty]{} 3$.

Example

As an example, consider the distributions presented in Section 2.1. Properties of subtrees resulting in multi-period trees with this bi-variate distribution are presented in Table 4. We see that there is a substantial difference in behaviour of the discrete and the continuous trees: in the discrete case, the subtrees are much more non-normal. As a result, the minimum number of branches of the subtrees, needed to obtain the desired properties, is usually higher in the discrete case, since more extreme distributions typically need more scenarios.

Table 4: Moments of subtrees to get a given final-stage distribution

	# periods	1	2	4	6	8	10	12
discrete processes	mean	0.0061	0.0030	0.0015	0.0010	0.0008	0.0006	0.0005
	std.dev.	0.0645	0.0454	0.0321	0.0262	0.0227	0.0203	0.0185
	skew	-0.580	-0.952	-1.438	-1.800	-2.100	-2.362	-2.598
	kurt	3.800	5.061	7.627	10.203	12.781	15.361	17.940
	mean	0.0036	0.0018	0.0009	0.0006	0.0004	0.0004	0.0003
	std.dev.	0.0861	0.0607	0.0429	0.0350	0.0303	0.0271	0.0247
	skew	-0.360	-0.685	-1.092	-1.388	-1.632	-1.844	-2.035
	kurt	3.380	4.183	5.867	7.568	9.273	10.980	12.688
	corr	0.5192	0.5196	0.5198	0.5198	0.5199	0.5199	0.5199
continuous processes	mean	0.0061	0.0030	0.0015	0.0010	0.0008	0.0006	0.0005
	std.dev.	0.0645	0.0456	0.0322	0.0263	0.0228	0.0204	0.0186
	skew	-0.580	-0.820	-1.160	-1.421	-1.640	-1.834	-2.009
	kurt	3.800	4.600	6.200	7.800	9.400	11.000	12.600
	mean	0.0036	0.0018	0.0009	0.0006	0.0004	0.0004	0.0003
	std.dev.	0.0861	0.0609	0.0430	0.0351	0.0304	0.0272	0.0248
	skew	-0.360	-0.509	-0.720	-0.882	-1.018	-1.138	-1.247
	kurt	3.380	3.760	4.520	5.280	6.040	6.800	7.560
	corr	0.5192	0.5192	0.5192	0.5192	0.5192	0.5192	0.5192

4.2 Dependent periods

In many cases, the assumption of independent periods is too restrictive. Also in finance, effects like mean-reversion or autocorrelation are very common. There are many models of inter-period dependency, see for example [16, 23, 2].

Instead on focusing on modelling of a special type of dependency, we discuss what can be done in the case when we need a scenario tree with dependent periods, but also with some degree of control over the unconditional distributions (distributions over all nodes in the same stage). As an example, Appendix B presents a way to implement first order autocorrelation, in a way that allows, at the same time, to control the unconditional means and variances.

In the case of a general inter-period dependency, we did not manage to repeat the results from the previous section, and produce formulas for controlling the final-stage distribution. The best we can do is to control the unconditional means and variances by a procedure that first generates the tree, and then corrects the unconditional distribution. This procedure is independent of the type of the inter-period dependency used.

If we stored the whole tree, we could correct the means and variances in all the periods at once. However, when we use the constructing/pricing algorithm (Fig. 3), we do not store the tree,[4] so we have to proceed stepwise, correcting one period at a time. As a result, we have to know the (unconditional) means and variances at every stage in order to do the corrections. If we know the unconditional means and variances only at the final stage, we have to interpolate the values for the rest of the stages.

Correcting means and variances for one period

Here we show how to correct the unconditional means and variances for a given period t, i.e. the period between stages $t-1$ and t. We want to change the update rules for the distribution at period t, so that the unconditional distribution of period t will have desired means $\boldsymbol{\mu}_t$ and variances $\boldsymbol{\sigma}_t^2$.

First, we have to compute the current properties $\bar{\boldsymbol{\mu}}_t$ and $\bar{\boldsymbol{\sigma}}_t^2$, so we have to generate a tree with at least t periods.[5] Once we know $\bar{\boldsymbol{\mu}}_t$ and $\bar{\boldsymbol{\sigma}}_t^2$, we can update the generating/pricing algorithm from Figure 3: after the generation of the subtree $\{\boldsymbol{x}(m), p(m) \,|\, m \in \mathbf{C}(n)\}$, we add the lines

$$\texttt{if } n \texttt{ is} \text{ in stage } t-1$$

$$\boldsymbol{x}(m) \leftarrow \frac{\boldsymbol{x}(m) - \bar{\boldsymbol{\mu}}_t}{\bar{\boldsymbol{\sigma}}_t}\boldsymbol{\sigma}_t + \boldsymbol{\mu}_t = \frac{\boldsymbol{\sigma}_t}{\bar{\boldsymbol{\sigma}}_t}\boldsymbol{x}(m) + \boldsymbol{\mu}_t - \frac{\boldsymbol{\sigma}_t}{\bar{\boldsymbol{\sigma}}_t}\bar{\boldsymbol{\mu}}_t \quad \text{for all } m \in \mathbf{C}(n).$$

Instead of transforming the outcomes $\boldsymbol{x}(m)$, it is more efficient to transform the means and variances before generating the subtree: if the means and variances for a subtree of node n are $\mathbb{E}\big[\tilde{\boldsymbol{X}}_t^n\big]$ and $\mathrm{Var}\big[\tilde{\boldsymbol{X}}_t^n\big]$, we change them to:

$$\mathbb{E}\big[\tilde{\boldsymbol{X}}_t^n\big] \leftarrow \frac{\boldsymbol{\sigma}_t}{\bar{\boldsymbol{\sigma}}_t}\mathbb{E}\big[\tilde{\boldsymbol{X}}_t^n\big] + \boldsymbol{\mu}_t - \frac{\boldsymbol{\sigma}_t}{\bar{\boldsymbol{\sigma}}_t}\bar{\boldsymbol{\mu}}_t \tag{5a}$$

$$\mathrm{Var}\big[\tilde{\boldsymbol{X}}_t^n\big] \leftarrow \frac{\boldsymbol{\sigma}_t}{\bar{\boldsymbol{\sigma}}_t}\mathrm{Var}\big[\tilde{\boldsymbol{X}}_t^n\big] \tag{5b}$$

[4] The fact that the algorithm does not need to store the whole tree is its main advantage, since it allows for processing of very large trees.

[5] Even if we do not store the tree, it is easy to compute the unconditional moments: During the execution of the algorithm, we accumulate the values of the non-central moments $\mathbb{E}\big[\tilde{X}^k\big]$. After the run, these are transformed to the central moments $\mathbb{E}\big[\big(\tilde{X} - \mathbb{E}\big[\tilde{X}\big]\big)^k\big]$, using the formulas from Appendix A.1.

Controlling unconditional means and variances at every period

To control the means and variances of the whole tree we apply the formulas from the previous section iteratively:

> `for` t `in` 2 `to` p {
>   generate tree with t periods, using corrections already found for $t' < t$
>   compute the unconditional means and variances in period t
>   compute the corrections of means and variances in period t
> }
> generate tree with p periods, using corrections for all $t' \in \{2 \dots p\}$.

Note that we have to generate $p - 1$ trees (growing in size from 2 to p periods), before we can generate the whole p-period tree with the right properties. Since the size of the trees grows exponentially with the number of periods, the whole procedure can be expected to take 2–3 times the time needed for generation of one p-period tree.

5 Stability issues

In sections 3 and 4.1 we explained that, in the case of independent periods with equal distributions, it is possible to generate only one subtree and reuse it throughout the whole multi-period tree.

While this resembles the "classical" binomial trees, there is an important difference: in a binomial tree the subtree is unique, in our trees there are infinitely many trees with the given properties (assuming that the subtrees have enough branches). Since we randomly pick only one, we have to test how much it influences the resulting option prices.

We tested prices of the two options defined in Section 2.1, both priced as American-style[6] options. To be able to test also the influence of the size of the subtrees, the test was done for 6-period trees. The size of the subtrees varied from 5 to 40. For every size, we generated 25 trees, priced the options on them, and computed the means and variances of the prices. Results are reported in Table 5.

Table 5: Stability of option prices with respect to the size of the subtrees

	# branches	5	10	15	18	20	40
underperf.	mean	1.76	1.84	1.89	1.93	1.83	1.77
	std. dev.	0.41	0.23	0.20	0.18	0.16	0.16
outperf.	mean	4.63	4.60	4.56	4.54	4.65	4.70
	std. dev.	0.45	0.22	0.21	0.19	0.16	0.17

We see that the prices are not stable, i.e. they depend on the choice of the constituting subtree. The question is, where does this instability come form? Since all the subtrees are

[6] Both options are call options. For the common one-asset call options, the price of an American call is equal to the price of an European call. In our case it turns out that the outperformance call is also never exercised before the final stage. For the underperformance call, the American call has higher price than the European, yet the difference is only about 3%.

constructed in such a way that the unconditional final-stage distributions have the right first four moments and correlations, the variance must come from some other properties.

One possible source is the variance of the higher moments of the marginals, which we do not control. However, our previous experience shows that the first four moments give so much stability that the variance from higher moments can not explain the observed variance of prices. To test it, we have used the same trees to price common one-asset options, and in this case the prices were stable.

Since we price American options, another possible source of the observed variance is an instability of the paths. However, when we price the options as European, the variance of the prices does not decrease – one of the reasons is the fact that the difference between the prices of American and European options is very small.[6]

Hence, the variance of the prices must come from differences in the structure of the multi-variate dependency. As we did in Section 2.1, we again see that the correlation matrix is not enough to describe the dependency. We may also document this graphically: Figure 4 shows density maps of two distributions with the same four moments and correlations – they correspond to trees with 40 branches per subtree, taking the cases with the highest and the lowest price for the underperformance option.

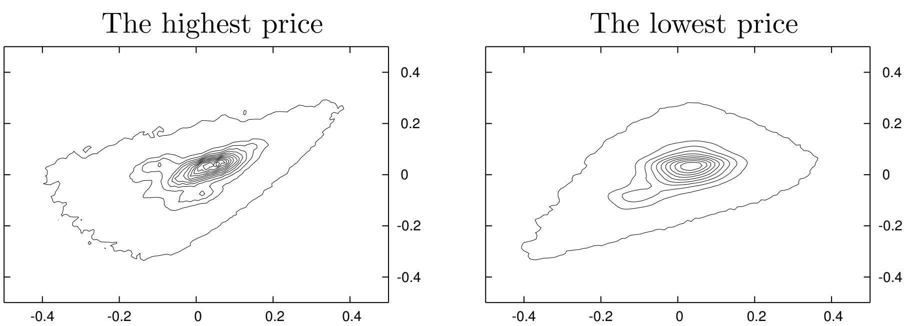

Figure 4: Two distributions with equal first four moments and correlations

This means that if we know only the moments and correlations of the return distribution, the prices of multi-variate options are not uniquely determined. Hence, the interval of prices from multiple runs can be interpreted as an interval of possible option prices, given this limited information. If we, on the other hand, know more about the multi-variate distribution, other approaches than the moment-matching presented in this paper should be appropriate. This is left for a future research.

There is, however, an easy way to reduce the variance of the option prices: instead of re-using the same one-period subtree throughout the whole multi-period tree, we may pre-generate many subtrees with the same properties, and sample from them during the generating/pricing algorithm (Fig. 3). The problem with this procedure is that we do not know anything about the distribution of the subtrees, so there is no guarantee that the sampling will be unbiased.

Acknowledgments

We would like to thank Ser-Huang Poon from the University of Strathclyde, Glasgow, Scotland, for suggesting the topic of option pricing and an initial help with the finance part of

the problem.

Conclusions

In this paper, we have presented a method to price multi-variate options. The method is based on non-recombining trees, which are created by a moment-matching approach. We have shown that even with the exponential growth of the size of the trees, it is possible to price options on 12-period trees in approximately half a minute. In addition, we have shown how to price European options using one-period trees.

For multi-period trees without inter-period dependency of the return distributions, we have shown formulas for moments and correlations of the constituting one-period subtrees, resulting in a desired final-stage distribution. In the case with inter-period dependency, we could not replicate the result and present only an ex-post procedure for correcting means and variances of the unconditional distributions.

We have also shown that the moment-based approach, with correlations as the only description of the multi-variate dependency, has its limits: if we have several distribution of prices with the same moments and correlations, the price of some multi-variate options may vary significantly, because these options can be very sensitive to the multi-variate structure of the distribution. Hence, better methods for description of the multi-variate structure should be used, if we have enough data/information. This is left for a future research.

References

[1] M. C. Cario and B.L. Nelson. Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois, 1997.

[2] Marne C. Cario and Barry L. Nelson. Numerical methods for fitting and simulating autoregressive-to-anything processes. *INFORMS Journal on Computing*, 10:72–81, 1997.

[3] Umberto Cherubini and Elisa Luciano. Multivariate option pricing with copulas. Working paper 05, International Centre For Economic Research, 2002.

[4] Robert T. Clemen and Terence Reilly. Correlations and copulas for decision and risk analysis. *Management Science*, 45(2):208–224, February 1999.

[5] Emanuel Derman and Iraj Kani. Riding with a smile. *Risk*, 7(2):32–39, 1994. Previously appeared as "The Volatility Smile and Its Implied Tree".

[6] Emanuel Derman, Iraj Kani, and Neil Chriss. Implied trinomial trees of the volatility smile. *Journal of Derivatives*, 3(4):7–22, 1996.

[7] F. Fornari and R. Violi. The probability density function of interest rates implied in the price of options. Banca d'Italia, Temi di Discussione del Servizio Studi 339, Banca d'Italia, October 1998.

[8] N. Gülpınar, B. Rustem, and R. Settergren. Multistage stochastic programming in computational finance. *Annals of Operations Research*, to appear?, 2001.

[9] K. Høyland and S. W. Wallace. Generating scenario trees for multistage decision problems. *Management Science*, 47(2):295–307, 2001.

[10] Kjetil Høyland, Michal Kaut, and Stein W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2-3):169–185, 2003.

[11] J. C. Jackwerth and M. Rubinstein. Recovering probability distributions from option prices. *Journal of Finance*, 51(5):1611–1631, 1996.

[12] Emmanuel Jurczenko, Bertrand Maillet, and Bogdan Negrea. Multi-moment approximate option pricing models: A general comparison (part 1). In *JMA 2002: 19èmes Journées de Micro-économie Appliquée*, Rennes and Saint-Malo, June 2002.

[13] R. R. P. Kouwenberg. Scenario generation and stochastic programming models for asset liability management. *European Journal of Operational Research*, 134(2):51–64, 2001.

[14] P. M. Lurie and M. S. Goldberg. An approximate method for sampling correlated random variables from partially-specified distributions. *Management Science*, 44(2):203–218, 1998.

[15] Johan Lyhagen. A method to generate multivariate data with moments arbitrary close to the desired moments. Working paper 481, Stockholm School of Economics, dec 2001.

[16] Benjamin Melamed. TES: A class of methods for generating autocorrelated uniform variables. *ORSA Journal on Computing*, 3(4):317–329, 1991.

[17] Roger B. Nelsen. *An Introduction to Copulas*. Springer-Verlag, New York, 1998.

[18] Joshua V. Rosenberg. Pricing multivariate contingent claims using estimated risk-neutral density functions. *Journal of International Money and Finance*, 17(2):229–247, April 1998.

[19] Joshua V. Rosenberg. Semiparametric pricing of multivariate contingent claims. Available at `http://pages.stern.nyu.edu/~jrosenb0/`, August 1999.

[20] Joshua V. Rosenberg. Nonparametric pricing of multivariate contingent claims. Available at `http://pages.stern.nyu.edu/~jrosenb0/`, April 2001.

[21] Mark Rubinstein. Implied binomial trees. *Journal of Finance*, 49(3):771–818, July 1994.

[22] Mark Rubinstein. Edgeworth binomial trees. *Journal of Derivatives*, 5(3):20–27, 1998.

[23] Wheyming Tina Song, Li-Ching Hsiao, and Yun-Ju Chen. Generating pseudo-random time series with specified marginal distributions. *European Journal of Operational Research*, 94:194–202, 1996.

[24] Luiz Vitiello, Antonio Camara, and Ser-Huang Poon. Pricing options with high moment distributions. Working paper at Strathclyde University, Glasgow, Scotland, November 2002.

A Moments of trees with independent stages

In this section we develop the formulas for the moments of the final-stage distribution as functions of the moments of the one-period subtrees, as well as the inverse formulas. In the formulas we assume that all the one-period subtrees have the same distributions and that the periods are independent. (A typical example is repeating the same subtree over the whole multi-period tree.)

In a multi-period tree with p periods we denote $\mu, \sigma^2, \gamma, \delta$ the first four moments of the final-stage distribution, and $\mu_p, \sigma_p^2, \gamma_p, \delta_p$ the first four moments of the corresponding subtrees.

A.1 General equalities

$$
\begin{aligned}
\mu &= \mathbb{E}\left[\tilde{X}\right] \\
\sigma^2 &= \mathrm{Var}\left[\tilde{X}\right] = \mathbb{E}\left[(\tilde{X} - \mathbb{E}\left[\tilde{X}\right])^2\right] = \mathbb{E}\left[\tilde{X}^2\right] - \mathbb{E}\left[\tilde{X}\right]^2 \\
\sigma^3\gamma &= \sigma^3 \,\mathrm{skew}\left[\tilde{X}\right] = \mathbb{E}\left[(\tilde{X} - \mathbb{E}\left[\tilde{X}\right])^3\right] = \mathbb{E}\left[\tilde{X}^3\right] - 3\,\mathbb{E}\left[\tilde{X}\right]\mathbb{E}\left[\tilde{X}^2\right] + 2\,\mathbb{E}\left[\tilde{X}\right]^3 \\
\sigma^4\delta &= \sigma^4 \,\mathrm{kurt}\left[\tilde{X}\right] = \mathbb{E}\left[(\tilde{X} - \mathbb{E}\left[\tilde{X}\right])^4\right] \\
&= \mathbb{E}\left[\tilde{X}^4\right] - 4\,\mathbb{E}\left[\tilde{X}\right]\mathbb{E}\left[\tilde{X}^3\right] + 6\,\mathbb{E}\left[\tilde{X}\right]^2\mathbb{E}\left[\tilde{X}^2\right] - 3\,\mathbb{E}\left[\tilde{X}\right]^4
\end{aligned}
$$

and the inverse relations:

$$
\begin{aligned}
\mathbb{E}\left[\tilde{X}^2\right] &= \mathrm{Var}\left[\tilde{X}\right] + \mathbb{E}\left[\tilde{X}\right]^2 = \sigma^2 + \mu^2 \\
\mathbb{E}\left[\tilde{X}^3\right] &= \sigma^3\mathrm{skew}\left[\tilde{X}\right] + 3\,\mathbb{E}\left[\tilde{X}\right]\mathbb{E}\left[\tilde{X}^2\right] - 2\,\mathbb{E}\left[\tilde{X}\right]^3 = \sigma^3\gamma + 3\mu(\sigma^2 + \mu^2) - 2\mu^3 \\
&= \sigma^3\gamma + 3\mu\sigma^2 + \mu^3 \\
\mathbb{E}\left[\tilde{X}^4\right] &= \sigma^4\mathrm{kurt}\left[\tilde{X}\right] + 4\,\mathbb{E}\left[\tilde{X}\right]\mathbb{E}\left[\tilde{X}^3\right] - 6\,\mathbb{E}\left[\tilde{X}\right]^2\mathbb{E}\left[\tilde{X}^2\right] + 3\,\mathbb{E}\left[\tilde{X}\right]^4 \\
&= \sigma^4\delta + 4\mu\left(\sigma^3\gamma + 3\mu\sigma^2 + \mu^3\right) - 6\mu^2\left(\sigma^2 + \mu^2\right) + 3\mu^4 \\
&= \sigma^4\delta + 4\mu\sigma^3\gamma + 6\mu^2\sigma^2 + \mu^4
\end{aligned}
$$

A.2 Formulas for discrete processes

In this section we present formulas for processes where the value is changed only at the stages of the tree, so the process V_t evolves as

$$\tilde{V}_t = (1 + \tilde{X}_t)\tilde{V}_{t-1}.$$

The final-stage return is then given as

$$\prod_{k=1}^{p}(1 + \tilde{X}_k) - 1.$$

For computing the moments of the distribution of final-stage returns, the following equalities are useful. Note that we use the independence assumption here.

$$
\begin{aligned}
\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right] &= \prod_{k=1}^{p}\mathbb{E}\left[1+\tilde{X}_k\right] = \prod_{k=1}^{p}\left(1+\mathbb{E}\left[\tilde{X}_k\right]\right) = \prod_{k=1}^{p}(1+\mu_p) \\
&= (1+\mu_p)^p \\
\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)^2\right] &= \prod_{k=1}^{p}\mathbb{E}\left[(1+\tilde{X}_k)^2\right] = \prod_{k=1}^{p}\mathbb{E}\left[1+2\tilde{X}_k+\tilde{X}_k^2\right] \\
&= \prod_{k=1}^{p}\left(1+2\mu_p+\sigma_p^2+\mu_p^2\right) = \left(1+2\mu_p+\sigma_p^2+\mu_p^2\right)^p \\
&= \left((1+\mu_p)^2+\sigma_p^2\right)^p \\
\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)^3\right] &= \prod_{k=1}^{p}\mathbb{E}\left[(1+\tilde{X}_k)^3\right] = \prod_{k=1}^{p}\mathbb{E}\left[1+3\tilde{X}_k+3\tilde{X}_k^2+\tilde{X}_k^3\right] \\
&= \prod_{k=1}^{p}\left(1+3\mu_p+3(\sigma_p^2+\mu_p^2)+(\sigma_p^3\gamma_p+3\mu_p\sigma_p^2+\mu_p^3)\right) \\
&= \left(1+3\mu_p+3\mu_p^2+\mu_p^3+3\sigma_p^2+3\mu_p\sigma_p^2+\sigma_p^3\gamma_p\right)^p \\
&= \left((1+\mu_p)^3+3(1+\mu_p)\sigma_p^2+\sigma_p^3\gamma_p\right)^p \\
\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)^4\right] &= \prod_{k=1}^{p}\mathbb{E}\left[(1+\tilde{X}_k)^4\right] = \prod_{k=1}^{p}\mathbb{E}\left[1+4\tilde{X}_k+6\tilde{X}_k^2+4\tilde{X}_k^3+\tilde{X}_k^4\right] \\
&= \prod_{k=1}^{p}\left(1+4\mu_p+6(\sigma_p^2+\mu_p^2)+4(\sigma_p^3\gamma_p+3\mu_p\sigma_p^2+\mu_p^3)\right. \\
&\quad \left.+(\sigma_p^4\delta_p+4\mu_p\sigma_p^3\gamma_p+6\mu_p^2\sigma_p^2+\mu_p^4)\right) \\
&= \left(1+4\mu_p+6\mu_p^2+4\mu_p^3+\mu_p^4+6\sigma_p^2(1+2\mu_p+\mu_p^2)\right. \\
&\quad \left.+4\sigma_p^3\gamma_p(1+\mu_p)+\sigma_p^4\delta_p\right)^p \\
&= \left((1+\mu_p)^4+6(1+\mu_p)^2\sigma_p^2+4(1+\mu_p)\sigma_p^3\gamma_p+\sigma_p^4\delta_p\right)^p
\end{aligned}
$$

The formulas for moments

$$
\mu = \mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)-1\right] = \prod_{k=1}^{p}\mathbb{E}\left[(1+\tilde{X}_k)\right]-1 = (1+\mu_p)^p-1
$$

$$
\mu_p = (1+\mu)^{1/p}-1 \tag{6}
$$

$$
\begin{aligned}
\sigma^2 &= \mathrm{Var}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)-1\right] = \mathrm{Var}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right] \\
&= \mathbb{E}\left[\left(\prod_{k=1}^{p}(1+\tilde{X}_k)\right)^2\right] - \left(\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right]\right)^2 \\
&= \big((1+\mu_p)^2+\sigma_p^2\big)^p - \big((1+\mu_p)^p\big)^2 = \big((1+\mu_p)^2+\sigma_p^2\big)^p - (1+\mu)^2 \\
\sigma_p^2 &= \big((1+\mu)^2+\sigma^2\big)^{1/p} - (1+\mu_p)^2 \qquad (7)
\end{aligned}
$$

$$
\begin{aligned}
\sigma^3\gamma &= \mathrm{skew}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)-1\right] = \mathrm{skew}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right] \\
&= \mathbb{E}\left[\left(\prod_{k=1}^{p}(1+\tilde{X}_k)\right)^3\right] - 3\,\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right]\mathbb{E}\left[\left(\prod_{k=1}^{p}(1+\tilde{X}_k)\right)^2\right] \\
&\quad + 2\left(\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right]\right)^3 \\
&= \big((1+\mu_p)^3+3(1+\mu_p)\sigma_p^2+\sigma_p^3\gamma_p\big)^p - 3(1+\mu)\big((1+\mu_p)^2+\sigma_p^2\big)^p + 2(1+\mu)^3 \\
&= \big((1+\mu_p)^3+3(1+\mu_p)\sigma_p^2+\sigma_p^3\gamma_p\big)^p - (1+\mu)\left[3\big((1+\mu_p)^2+\sigma_p^2\big)^p - 2(1+\mu)^2\right] \\
&= \big((1+\mu_p)^3+3(1+\mu_p)\sigma_p^2+\sigma_p^2\gamma_p\big)^p - (1+\mu)\big(3\sigma^2+(1+\mu)^2\big) \\
&= \big((1+\mu_p)^3+3(1+\mu_p)\sigma_p^2+\sigma_p^3\gamma_p\big)^p - (1+\mu)^3 - 3(1+\mu)\sigma^2 \\
\gamma_p &= \frac{1}{\sigma_p^3}\left[\big((1+\mu)^3+3(1+\mu)\sigma^2+\sigma^3\gamma\big)^{1/p} - (1+\mu_p)^3 - 3(1+\mu_p)\sigma_p^2\right] \qquad (8)
\end{aligned}
$$

$$
\begin{aligned}
\sigma^4\delta &= \operatorname{kurt}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)-1\right] = \operatorname{kurt}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right] \\
&= \mathbb{E}\left[\left(\prod_{k=1}^{p}(1+\tilde{X}_k)\right)^4\right] - 4\,\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right]\mathbb{E}\left[\left(\prod_{k=1}^{p}(1+\tilde{X}_k)\right)^3\right] \\
&\quad + 6\left(\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right]\right)^2 \mathbb{E}\left[\left(\prod_{k=1}^{p}(1+\tilde{X}_k)\right)^2\right] - 3\left(\mathbb{E}\left[\prod_{k=1}^{p}(1+\tilde{X}_k)\right]\right)^4 \\
&= \big((1+\mu_p)^4 + 6(1+\mu_p)^2\sigma_p^2 + 4(1+\mu_p)\sigma_p^3\gamma_p + \sigma_p^4\delta_p\big)^p \\
&\quad - 4(1+\mu)\big((1+\mu_p)^3 + 3(1+\mu_p)\sigma_p^2 + \sigma_p^3\gamma_p\big)^p \\
&\quad + 6(1+\mu)^2\left((1+\mu_p)^2+\sigma_p^2\right)^p - 3(1+\mu)^4 \\
&= \big((1+\mu_p)^4 + 6(1+\mu_p)^2\sigma_p^2 + 4(1+\mu_p)\sigma_p^3\gamma_p + \sigma_p^4\delta_p\big)^p \\
&\quad (1+\mu)\left[-4\big(\sigma^3\gamma + 3(1+\mu)\sigma^2 + (1+\mu)^3\big) + 6(1+\mu)\left(\sigma^2+(1+\mu)^2\right) - 3(1+\mu)^3\right] \\
&= \big((1+\mu_p)^4 + 6(1+\mu_p)^2\sigma_p^2 + 4(1+\mu_p)\sigma_p^3\gamma_p + \sigma_p^4\delta_p\big)^p \\
&\quad - (1+\mu)^4 - 6(1+\mu)^2\sigma^2 - 4(1+\mu)\sigma^3\gamma
\end{aligned}
$$

$$
\begin{aligned}
\delta_p = \frac{1}{\sigma_p^4}\Big[&\big((1+\mu)^4 + 6(1+\mu)^2\sigma^2 + 4(1+\mu)\sigma^3\gamma + \sigma^4\delta\big)^{1/p} \\
&-(1+\mu_p)^4 - 6(1+\mu_p)^2\sigma_p^2 - 4(1+\mu_p)\sigma_p^3\gamma_p\Big]
\end{aligned}
\tag{9}
$$

The formulas for correlations

Finally, for two random variables $\tilde{X}$ and $\tilde{Y}$, we can also measure the correlation ρ. For this formulas, we add upper indices X and Y to the notation for mean and variance.

$$
\begin{aligned}
\sigma^X\sigma^Y\rho &= \operatorname{Cov}\left[\prod_{k=1}^{p}\left(1+\tilde{X}_k\right)-1, \prod_{k=1}^{p}\left(1+\tilde{Y}_k\right)-1\right] = \operatorname{Cov}\left[\prod_{k=1}^{p}\left(1+\tilde{X}_k\right), \prod_{k=1}^{p}\left(1+\tilde{Y}_k\right)\right] \\
&= \mathbb{E}\left[\prod_{k=1}^{p}\left(1+\tilde{X}_k\right)\prod_{k=1}^{p}\left(1+\tilde{Y}_k\right)\right] - \mathbb{E}\left[\prod_{k=1}^{p}\left(1+\tilde{X}_k\right)\right]\mathbb{E}\left[\prod_{k=1}^{p}\left(1+\tilde{Y}_k\right)\right] \\
&= \mathbb{E}\left[\prod_{k=1}^{p}\left(1+\tilde{X}_k\right)\left(1+\tilde{Y}_k\right)\right] - (1+\mu^X)(1+\mu^Y) \\
&= \prod_{k=1}^{p}\mathbb{E}\big[\left(1+\tilde{X}_k\right)\left(1+\tilde{Y}_k\right)\big] - (1+\mu^X)(1+\mu^Y) \\
&= \prod_{k=1}^{p}\left(\mathbb{E}\big[\left(1+\tilde{X}_k\right)\big]\mathbb{E}\big[\left(1+\tilde{Y}_k\right)\big] + \mathtt{Cov}(\tilde{X},\tilde{Y})\right) - (1+\mu^X)(1+\mu^Y) \\
&= \big((1+\mu_p^X)(1+\mu_p^Y) + \sigma_p^X\sigma_p^Y\rho_p\big)^p - (1+\mu^X)(1+\mu^Y)
\end{aligned}
$$

$$
\rho_p = \frac{1}{\sigma_p^X\sigma_p^Y}\left[\big(\sigma^X\sigma^Y\rho + (1+\mu^X)(1+\mu^Y)\big)^{1/p} - (1+\mu_p^X)(1+\mu_p^Y)\right] \tag{10}
$$

A.3 Formulas for continuous processes

In this section we present formulas for continuous processes, i.e. processes whose values V_t are compounded at continuous rate

$$V_t = e^{\tilde{X}_t} V_{t-1}.$$

The final-stage return is then defined as

$$\tilde{X} = \ln\left(\frac{V_p}{V_0}\right) = \ln(V_p) - \ln(V_0) = \ln\left(e^{\sum_{k=1}^{p} \tilde{X}_k} V_0\right) - \ln(V_0) = \sum_{k=1}^{p} \tilde{X}_k.$$

Using the formulas from Section A.1, we get the following relations between the subtree properties and the final-stage properties:

$$\begin{aligned} \mu = \mathbb{E}\left[\sum_{k=1}^{p} (\tilde{X}_k)\right] = \sum_{k=1}^{p} \mathbb{E}\left[\tilde{X}_k)\right] = p\,\mu_p \\ \mu_p = p^{-1}\mu \end{aligned} \tag{11}$$

$$\begin{aligned} \mathbb{E}\left[\tilde{X}^2\right] &= \mathbb{E}\left[\left(\sum_{k=1}^{p} \tilde{X}_k\right)^2\right] = \mathbb{E}\left[\sum_{k=1}^{p}\sum_{l=1}^{p} \tilde{X}_k \tilde{X}_l\right] = \sum_{k=1}^{p}\sum_{l=1}^{p} \mathbb{E}\left[\tilde{X}_k \tilde{X}_l\right] \\ &= \sum_{k=1}^{p}\left(\mathbb{E}\left[\tilde{X}_k\right]\sum_{l\neq k}\mathbb{E}\left[\tilde{X}_l\right] + \mathbb{E}\left[\tilde{X}_k^2\right]\right) \\ &= \sum_{k=1}^{p}\left(\mathbb{E}\left[\tilde{X}_k\right]\left(\sum_{l=1}^{p}\mathbb{E}\left[\tilde{X}_l\right] - \mathbb{E}\left[\tilde{X}_k\right]\right) + \mathbb{E}\left[\tilde{X}_k^2\right]\right) \\ &= \sum_{k=1}^{p}\left(\mu\mathbb{E}\left[\tilde{X}_k\right] + \mathbb{E}\left[\tilde{X}_k^2\right] - \left(\mathbb{E}\left[\tilde{X}_k\right]\right)^2\right) = \mu^2 + \sum_{k=1}^{p}\sigma_k^2 = \mu^2 + p\,\sigma_p^2 \\ \sigma^2 = \mathrm{Var}\left[\sum_{k=1}^{p}(\tilde{X}_k)\right] &= \mathbb{E}\left[\tilde{X}^2\right] - \mathbb{E}\left[\tilde{X}\right]^2 = \mu^2 + p\,\sigma_p^2 - \mu^2 = p\,\sigma_p^2 \\ \sigma_p^2 &= p^{-1}\sigma_p^2 \end{aligned} \tag{12}$$

$$\begin{aligned} \mathbb{E}\left[\tilde{X}^3\right] &= \mathbb{E}\left[\left(\sum_{k=1}^{p} \tilde{X}_k\right)^3\right] = \mathbb{E}\left[\sum_{k,l,m=1}^{p} \tilde{X}_k \tilde{X}_l \tilde{X}_m\right] = \sum_{k,l,m=1}^{p} \mathbb{E}\left[\tilde{X}_k \tilde{X}_l \tilde{X}_m\right] \\ &= \sum_{k=1}^{p}\left(\sum_{l,m\neq k}\mathbb{E}\left[\tilde{X}_k\right]\mathbb{E}\left[\tilde{X}_l\tilde{X}_m\right] + 2\sum_{l\neq k}\mathbb{E}\left[\tilde{X}_k^2\tilde{X}_l\right] + \mathbb{E}\left[\tilde{X}_k^3\right]\right) \\ &= \sum_{k=1}^{p}\mathbb{E}\left[\tilde{X}_k\right]\sum_{l,m\neq k}\mathbb{E}\left[\tilde{X}_l\tilde{X}_m\right] + 2\sum_{k=1}^{p}\mathbb{E}\left[\tilde{X}_k^2\right]\sum_{l\neq k}\mathbb{E}\left[\tilde{X}_l\right] + \sum_{k=1}^{p}\mathbb{E}\left[\tilde{X}_k^3\right] \end{aligned}$$

$$
\begin{aligned}
\sum_{l,m\neq k}\mathbb{E}\left[\tilde{X}_l\tilde{X}_m\right] &= \sum_{l\neq k}\left(\sum_{m=1}^{p}\mathbb{E}\left[\tilde{X}_l\tilde{X}_m\right]-\mathbb{E}\left[\tilde{X}_l\tilde{X}_k\right]\right)\\
&= \sum_{l=1}^{p}\left(\sum_{m=1}^{p}\mathbb{E}\left[\tilde{X}_l\tilde{X}_m\right]-\mathbb{E}\left[\tilde{X}_l\tilde{X}_k\right]\right)-\left(\sum_{m=1}^{p}\mathbb{E}\left[\tilde{X}_k\tilde{X}_m\right]-\mathbb{E}\left[\tilde{X}_k^2\right]\right)\\
&= \mu+\sigma^2-2\sum_{l=1}^{p}\mathbb{E}\left[\tilde{X}_l\tilde{X}_k\right]+\mathbb{E}\left[\tilde{X}_k^2\right]\\
&= \mu+\sigma^2-2\left(\mu\mathbb{E}\left[\tilde{X}_k\right]+\mathbb{E}\left[\tilde{X}_k^2\right]-\left(\mathbb{E}\left[\tilde{X}_k\right]\right)^2\right)+\mathbb{E}\left[\tilde{X}_k^2\right]\\
&= \mu+\sigma^2-2\mu\mathbb{E}\left[\tilde{X}_k\right]-\mathbb{E}\left[\tilde{X}_k^2\right]+2\left(\mathbb{E}\left[\tilde{X}_k\right]\right)^2
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}\left[\tilde{X}^3\right] &= \sum_{k=1}^{p}\mathbb{E}\left[\tilde{X}_k\right]\left(\mu+\sigma^2-2\mu\mathbb{E}\left[\tilde{X}_k\right]-\mathbb{E}\left[\tilde{X}_k^2\right]+2\left(\mathbb{E}\left[\tilde{X}_k\right]\right)^2\right)\\
&\quad+2\sum_{k=1}^{p}\mathbb{E}\left[\tilde{X}_k^2\right]\left(\mu-\mathbb{E}\left[\tilde{X}_k\right]\right)+\sum_{k=1}^{p}\mathbb{E}\left[\tilde{X}_k^3\right]\\
&= \mu\left(\mu+\sigma^2\right)+2\mu\sigma^2\\
&\quad+\sum_{k=1}^{p}\left(-\mathbb{E}\left[\tilde{X}_k\right]\mathbb{E}\left[\tilde{X}_k^2\right]+2\left(\mathbb{E}\left[\tilde{X}_k\right]\right)^3-2\mathbb{E}\left[\tilde{X}_k^2\right]\mathbb{E}\left[\tilde{X}_k\right]+\mathbb{E}\left[\tilde{X}_k^3\right]\right)\\
&= \mu^3+3\mu\sigma^2+\sum_{k=1}^{p}\left(\mathbb{E}\left[\tilde{X}_k^3\right]-3\mathbb{E}\left[\tilde{X}_k\right]\mathbb{E}\left[\tilde{X}_k^2\right]+2\left(\mathbb{E}\left[\tilde{X}_k\right]\right)^3\right)
\end{aligned}
$$

$$
\sigma^3\gamma = \mathbb{E}\left[\tilde{X}^3\right]-\mu^3-3\mu\sigma^2 = \sum_{k=1}^{p}\sigma_k^3\gamma_k = p\,\sigma_p^3\gamma_p
$$

$$
\sigma_p = \frac{\sigma^3}{p\,\sigma_p^3}\gamma = \frac{p^{3/2}\sigma_p^3}{p\,\sigma_p^3}\gamma = \sqrt{p}\,\gamma \tag{13}
$$

Similar, but more complicated, calculations show that for kurtosis we have

$$
\begin{aligned}
\delta-3 &= p^{-1}\left(\delta_p-3\right)\\
\delta_p &= p\left(\delta-3\right)+3
\end{aligned} \tag{14}
$$

Finally, for correlations we get:

$$
\begin{aligned}
\mathbb{E}\left[\tilde{X}\tilde{Y}\right] &= \mathbb{E}\left[\left(\sum_{k=1}^{p}\tilde{X}_k\right)\left(\sum_{l=1}^{p}\tilde{Y}_l\right)\right] = \sum_{k,l=1}^{p}\mathbb{E}\left[\tilde{X}_k\tilde{Y}_l\right] \\
&= \sum_{k=1}^{p}\left(\sum_{l\neq k}\mathbb{E}\left[\tilde{X}_k\right]\mathbb{E}\left[\tilde{Y}_l\right] + \mathbb{E}\left[\tilde{X}_k\tilde{Y}_k\right]\right) \\
&= \sum_{k=1}^{p}\left(\mathbb{E}\left[\tilde{X}_k\right]\left(\sum_{l=1}^{p}\mathbb{E}\left[\tilde{Y}_l\right] - \mathbb{E}\left[\tilde{Y}_k\right]\right) + \mathbb{E}\left[\tilde{X}_k\tilde{Y}_k\right]\right) \\
&= \sum_{k=1}^{p}\mathbb{E}\left[\tilde{X}_k\right]\sum_{l=1}^{p}\mathbb{E}\left[\tilde{Y}_l\right] + \sum_{k=1}^{p}\Big(\mathbb{E}\left[\tilde{X}_k\tilde{Y}_k\right] - \mathbb{E}\left[\tilde{X}_k\right]\mathbb{E}\left[\tilde{Y}_k\right]\Big) \\
&= \mathbb{E}\left[\tilde{X}\right]\mathbb{E}\left[\tilde{Y}\right] + \sum_{k=1}^{p}\mathrm{Cov}\left[\tilde{X}_k,\tilde{Y}_k\right] \\
\sigma^X\sigma^Y\rho = \mathrm{Cov}\left[\tilde{X},\tilde{Y}\right] &= \mathbb{E}\left[\tilde{X}\tilde{Y}\right] - \mathbb{E}\left[\tilde{X}\right]\mathbb{E}\left[\tilde{Y}\right] = \sum_{k=1}^{p}\mathrm{Cov}\left[\tilde{X}_k,\tilde{Y}_k\right] \\
&= \sum_{k=1}^{p}\sigma_k^X\sigma_k^Y\rho_k = p\,\sigma_p^X\sigma_p^Y\rho_p \\
\rho_p &= \frac{\sigma^X\sigma^Y}{p\,\sigma_p^X\sigma_p^Y}\rho = \frac{\sqrt{p}\,\sigma_p^X\sqrt{p}\,\sigma_p^Y}{p\,\sigma_p^X\sigma_p^Y}\rho = \rho
\end{aligned}
\tag{15}
$$

B Example of dependency: first order autocorrelation

In this section we show how to add a first order autocorrelation to the scenario-tree generating procedure, in such a way that we do not change the means and variances of the unconditional distributions. We describe only the simplest case when the distribution of a marginal $\tilde{X}_{t,i}$ of the random vector $\tilde{\boldsymbol{X}}_t$ depends explicitly[7] only on the previous value $\boldsymbol{x}_{t-1,i}$. For more general information about modelling the autocorrelation, see for example [16, 23, 2]. For the rest of the section, we drop the index i to simplify the formulas.

In the context of scenario trees, the first order correlation can be added during the generation of the tree, using the algorithm described in Section 3. Hence, we need formulas for adding the autocorrelation to a subtree of a given node n at stage t of the tree. Since the tree generation proceeds from the root to the leaves, we already know the outcome $\boldsymbol{x}_t$ of the random vector $\tilde{\boldsymbol{X}}_t$ at node n. We also know the unconditional distribution of the successive subtree, $\tilde{\boldsymbol{X}}_{t+1}$. We want to update the distribution in such a way that

$$\mathrm{corr}\left(\tilde{\boldsymbol{X}}_{t+1},\tilde{\boldsymbol{X}}_t\right) = \rho_1.$$

To distinguish between the (unconditional) moments of $\tilde{\boldsymbol{X}}_t$ and $\tilde{\boldsymbol{X}}_{t+1}$, we add a time index to the symbols for moments μ and σ^2.

[7] We do not say that $\tilde{\boldsymbol{X}}_{t,i}$ does not depend on $\{\boldsymbol{x}_{\tau,i},\ \tau < t-1\}$, we just do not control these dependencies.

To introduce the correlation, we use the Cholesky decomposition of the correlation matrix

$$R = \begin{bmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \rho_1 & \sqrt{1-\rho_1^2} \end{bmatrix} \cdot \begin{bmatrix} 1 & \rho_1 \\ 0 & \sqrt{1-\rho_1^2} \end{bmatrix} = LL^T.$$

Since $\tilde{X}_{t+1}$ is independent of $\tilde{X}_t$, we know that, with the random vector $\tilde{\boldsymbol{Y}} = L\left[\tilde{X}_t, \tilde{X}_{t+1}\right]^T$, we have

$$\begin{aligned} \tilde{Y}_1 &= \tilde{X}_t \\ \tilde{Y}_2 &= \rho_1 \tilde{X}_t + \sqrt{1-\rho_1^2}\tilde{X}_{t+1} \\ \operatorname{corr}\left(\tilde{\boldsymbol{Y}}_{t+1}, \tilde{\boldsymbol{Y}}_t\right) &= \rho_1 \end{aligned} \tag{16}$$

We want $\tilde{Y}_2$ to become the updated (autocorrelated) version of the random variable $\tilde{X}_{t+1}$, so their unconditional distributions should be equal. We do not know how to guarantee equal distributions, yet we can at least control the means and variances: The transformation $\tilde{\boldsymbol{Y}} = L\tilde{\boldsymbol{X}}$ does not change the means and variances, if $\tilde{\boldsymbol{X}}$ has zero means and variances of one. We can therefore first normalise $\tilde{\boldsymbol{X}}$, then do the transformation $\tilde{\boldsymbol{Y}} = L\tilde{\boldsymbol{X}}$, and finally transform $\tilde{\boldsymbol{Y}}$ to the moments μ_{t+1}, σ_{t+1}^2. Since all of these are linear transformations, they will not change the correlation ρ_1. The formula then becomes

$$\begin{aligned} \tilde{Y}_2 &= \left(\rho_1 \frac{x_t - \mu_t}{\sigma_t} + \sqrt{1-\rho_1^2}\frac{\tilde{X}_{t+1} - \mu_{t+1}}{\sigma_{t+1}}\right)\sigma_{t+1} + \mu_{t+1} \\ &= \mu_{t+1} + \sigma_{t+1}\rho_1 \frac{x_t - \mu_t}{\sigma_t} + \sqrt{1-\rho_1^2}\left(\tilde{X}_{t+1} - \mu_{t+1}\right). \end{aligned} \tag{17}$$

From this formula, we get the following properties:

$$\text{conditional:} \quad \begin{cases} \mathbb{E}\left[\tilde{Y}_2 | \tilde{X}_t = x_t\right] = \mu_{t+1} + \sigma_{t+1}\rho_1 \dfrac{x_t - \mu_t}{\sigma_t} \\ \operatorname{Var}\left[\tilde{Y}_2 | \tilde{X}_t = x_t\right] = \left(1-\rho_1^2\right)\sigma_{t+1}^2 \end{cases} \tag{18}$$

$$\text{unconditional:} \quad \begin{cases} \mathbb{E}\left[\tilde{Y}_2\right] = \mu_{t+1} \\ \operatorname{Var}\left[\tilde{Y}_2\right] = \sigma_{t+1}^2 \end{cases} \tag{19}$$

We see that the unconditional distribution of $\tilde{Y}_2$ has the same unconditional mean and variance as the distribution of $\tilde{X}_{t+1}$. Unfortunately, this is not case with the higher moments. For example, even in the simple case skew $\left[\tilde{X}_t\right]$ = skew $\left[\tilde{X}_{t+1}\right]$, we get

$$\operatorname{skew}\left[\tilde{Y}_2\right] = \left(\rho_1^3 + \left(1-\rho_1^2\right)^{3/2}\right)\operatorname{skew}\left[X_t\right].$$

Note also that there are two equivalent ways of implementing the transformation: we may either transform the vector of outcomes $\left\{x_{t+1}^m \mid m \in \mathbf{C}(n)\right\}$, or we may transform its mean and variance before we generate the outcomes.