Pricing Constant Maturity Floaters with Embedded Options Using Monte Carlo Simulation *

Engelbert J. Dockner **Hans Moritsch**

Department of Business
University of Vienna, Brünner Strasse 72, A-1210 Vienna, Austria
E-Mail: {dockner,moritsch}@finance2.bwl.univie.ac.at

Abstract

A popular interest rate security used by Austrian banks are so called secondary market yield floaters with caps or floors or other types of path dependent embedded options. Since there are no analytical pricing formulas for these constant maturity instruments, numerical techniques have to be employed. In this paper we present Monte Carlo Simulation techniques to price these products. Because these techniques are computationally intensive, emphasis is put on a parallel implementation of these products. We find that a parallel implementation enhances the performance of the numerical analysis considerably and makes an accurate pricing of complex interest rate sensitive products possible.

1 Introduction and Motivation

The pricing of interest rate dependent financial instruments is one of the most important areas in asset pricing theory. In case of simple instruments with deterministic (nondefaultable) cash flows pricing is based on an arbitrage argument on the basis of pure discount bond prices. Hence, if we have given a term structure of spot rates it is easy to derive the price of a fixed coupon bond.

*This research was a part of the Special Research Program SFB F011 AURORA supported by the Austrian *Fonds zur Foerderung der wissenschaftlichen Forschung.*

The theoretical price then simply is the present value of future cash flows. Many interest rate dependent products are not based on a fixed interest rate. In case of floating rate securities we have to distinguish two different classes of instruments depending on whether the maturity of the reference interest rate is smaller or equal to the period until the next interest rate adjustment takes place. In case the maturity of the reference interest rate is smaller than the adjustment period it can be shown that the price of the floater at the time of the next adjustment is equal to its face value so that the pricing of the floater is identical to that of a fixed income (deterministic cash flow) instrument. In case the maturity of the reference interest rate exceeds the period of adjustment we are faced with a so called constant maturity instrument and the pricing becomes much more involved. In particular one needs an interest rate model that allows for projections of future interest rates that can be used to forecast uncertain interest payments. Hull and White (see [10]) developed a single factor model that we will make use of in this paper to price constant maturity instruments.

The use of constant maturity instruments is very popular among Austrian banks (see [12]). In particular instruments based on the so called secondary market yield (SMY) are frequently used. Examples include loans, credits as well as bonds that use the SMY as a reference interest rate. The SMY is an index of Austrian government bond yields currently traded in the secondary market and can be interpreted as an average yield to maturity. Since the maturity spectrum of Austrian government bonds ranges from less than one year to thirty years the time bucket to which the SMY corresponds is in the range of five to seven years. Hence, all instruments that make use of the SMY fit into the class of constant maturity floaters.

The popularity of the SMY stems from its properties given a normal shaped yield curve. If bank deposits (liabilities) have on average a short maturity and loans with the SMY as interest rate (assets) have a longer time to maturity the normal shape of the yield curve guarantees a remarkable profit for the bank. Things are, however, quite different during a period of an inverse yield curve.

Despite these arguments the pricing of constant maturity instruments is an interesting theoretical issue and becomes even more involved if the interest rate product is characterized by option features.

In this paper we take up the issue of pricing CMI's with embedded options on the basis of numerical techniques. As far as the embedded options are concerned we look at the following products. As mentioned above the use of SMY floating instruments are very popular in Austria. Moreover some of these products have the follwing type of interest rate rate adjustment. There is an initial reference rate for that applies as long as the variable interest rate does not hit a lower or an upper bound. If the variable rate (the SMY) passes the bounds an interest rate adjustment is made that depends on a factor of adjustment as well as the previous adjustment levels. Hence these products are characterized by caps and floors but the cap/floor rate depends on past interest rate realizations. Therefore the products are path dependent. To capture these characteristics we make use of the Hull and White interest rate tree model and use Monte Carlo techniques to price the instruments. Using the Hull and White trinomial tree implies that we generate an entire term structure on the basis of a single factor (risk factor) which is the short term interest rate. The flexibility of the Hull and White model, however, guarantees that the term structures generated by the tree are consistent with todays observed one. Moreover the model can be calibrated to fit the current volatility structure as well.

Making use of Monte Carlo simulation techniques together with the Hull and White interest rate tree gives us enough flexibility to price a very large spectrum of interest rate products. There is one disadvantage to this approach, however. A Monte Carlo simulation is computationally very intensive. Therefore we present two possible implementations of our model. One is a sequential version and the other one makes use of data parallel structures. We present performance evaluations based on this two implementations and sensitivity analysis as the accuracy of the pricing tool is concerned. The pricing module presented in this paper is part of a larger model that has been developed within the AURORA research program. This model fits into the category of a financial planning tool that uses stochastic optimization techniques to derive optimal financial allocation decisions (see [6]).

Our paper is organized as follows. In the next section we present the Hull and White interest rate tree model and discuss its applications to the pricing of our products. In section 3 we present a description of the SMY floaters with caps and floors. Section 4 discusses the numerical implementation as well as the numerical results and section 5 concludes the paper.

2 The Hull and White Interest Rate Model

As outlined in the introduction our aim is to derive a pricing tool for constant maturity instruments based on the interest rate model developed by Hull and White (see [10]). The Hull and White model fits into the class of one factor arbitrage free models and is based on the following dynamic specification for the instantaneous short rate:

$$dr(t) = (\theta(t) - ar(t))dt + \sigma dZ \tag{1}$$

where $r(t)$ is the short rate at time t, σ is the volatility and a and $\theta(t)$ are related to the average short rate and the speed of adjustment. dZ are the increments of a standard Wiener process. In (1) the dynamics is characterized by a mean reverting process. The important difference between the Hull and White model and other one factor models like the classical Vasicek model stems from the modelling of $\theta(t)$ as a function of time. Through this approach it becomes possible that the model is exactly calibrated to the existing term structure in the market so that the model is consistent with the current spot rate curve observed in the market. From the specification of the short rate $r(t)$ it is possible to derive the prices of zero bonds and hence the term structure. In particular it can be shown that the bond prices implied by the Hull and White model are given by

$$P(r,t,T) = A(t,T)exp(-B(t,T)r) \tag{2}$$

where $P(r,t,T)$ is the price of a zero bond at time t with maturity $T > t$ when the short rate at time t is r. $A(t,T)$ and $B(t,T)$ are defined implicitely by

$$\begin{aligned} A_t - \theta(t)AB + \frac{\sigma^2 AB^2}{2} &= 0 \\ B_t - aB + 1 &= 0 \end{aligned}$$

whith the terminal conditions $A(T,T) = 1$ and $B(T,T) = 0$.

In this paper we will not make use of the continuous time version of the model but instead use the trinomial specification. In order for the tree to be recombining we assume that it is evenly spaced in the r space. Moreover we assume that it is evenly spaced over time as well (see Figure 1). The value of r at different nodes has the form $r_0 + k\Delta r$ where k is some positive integer. According to Hull and White the time steps Δt is set in such a way that

$$\Delta t = (\Delta r)^2/3\sigma^2 \tag{3}$$

holds. A node (i,j) in the tree will denote the value of $r = r_0 + j\Delta r$ and $t = i\Delta t$.

In the tree there are three branches at each node and the probabilities on these branches vary in the tree. There are now several methods for specifying these probabilities. Here we use the method that the probabilities are consistent with the probability properties of the model. This requires to specify the three probabilities according to the following three rules

1. $p_u + p_m + p_d = 1$
2. The expected change in r over the next interval Δt must be matched.
3. The variance of the change in r over the next interval Δt must be matched.

From this three conditions the probabilities can be calculated. In a next step the mean reverting property must be incorporated in the model. In the Hull and White model this is done by making use of nonstandard branching which we will not describe any further in this paper (see [10]). The probabilities need to be adjusted for the nonstandard branching. It can now be shown that if (3) holds and at all the nodes of the tree an inequality is satisfied all the transition probabilities are positive.

Based on the branching method of the three a forward induction method can be applied (see [11]) so that the tree model fits the initial term structure and hence the model can be used as an arbitrage free pricing model for interest rate sensitive products. For that matter we define Q_{ij} as the present value of a security that pays \$1 if node (i,j) is reached and zero otherwise. The Q_{ij} are also called state prices. The values of Q_{ij} can be computed inductively with the initial condition $Q_{00} = 1$. The following result specifies the approach (see [16]).

Lemma 1:

Let $q(k,j)$ be the probabilities of moving from node (i,k) to node $(i+1,j)$, then

$$Q_{i+1,j} = \sum_k Q_{i,k} q(k,j) \exp\left(-r(i,k)\Delta t\right)$$

where the summation is over all k for which this is not zero.

■

On the basis of this result the price of a discount bond that matures at time $(i+1)\Delta t$ can be expressed as

$$P_{i+1} = \sum_j Q_{i,j} \exp\left(-r(i,j)\Delta t\right)$$

which provides us with all the details that are needed to make use of this model for pricing.

Since the Hull and White model allows for calibration to the initial term structure the function $\theta(t)$ needs to be specified so that the current spot rates are consistent with the model. To do this, let $\theta(t)$ be specified as

$$\theta(i\Delta t) = \frac{1}{\Delta}(i+1)R(i+2) + \frac{\sigma\Delta t}{2} + \frac{1}{\Delta t^2} \ln \sum_j Q_{i,j} \exp(-r_{i,j}\Delta t + \alpha r_{i,j}\Delta t^2)$$

where $R(i+2)$ is the current spot rate for maturity $i+2$. On the basis of the function $\theta(t)$ the drift parameter of the short rate can be determined. The drift parameter $\mu_{i,j}$ becomes

$$\mu_{i,j} = [\theta(i\Delta t) + \alpha r_{i,j}]\Delta t.$$

Hence with these specifications all the necessary information for the interest rate tree is given so that all interest rate sensitive products can be priced. The building of the tree involves several steps. In a first step the evolution of the short rates is determined. Next the drift rate of the short rate is calculated because it is needed to derive the probabilities. (The drift rate is given by $\theta(t) - ar(t)$). If we are equipped with the probabilities the state prices can be calculated and the prices of zero bonds along the tree.

3 Pricing CMI's With Embedded Options

The types of product we are interested in this paper are floaters with embedded options where the reference interest rate is the secondary market yield of Austrian government bonds. The SMY of Austrian government bonds is a weighted yield to maturity of all government bonds currently traded in the market where the weights of this index are based on the volume of the bonds issued. Hence by issueing a new bond the government can indirectly influence the level of the SMY. This, however, poses a problem for the modeler of floaters that use the SMY as reference rate. In particular the future development of the SMY cannot be directly related to the interest rate tree introduced in the preceeding section, because the future values of SMY not only depend on the future development of the interest rates but also on the characteristics of the bonds issued by the government. In order to overcome this problem we make the assumption that the SMY can be specified as a linear combination of spot rates. In particular we assume that (see [12]):

$$\Delta SMY_t = \gamma_1 + \gamma_2 \Delta R(t+j) + \gamma_3 \Delta R(t+k)$$

where j and k specify the maturities of the spot rates that are used to replicate the SMY and γ_i are constants. The implication of this assumption is that now the interest rate tree can be used to price the floaters since we relate the variable SMY to two (or more) spot rates. Pricing floaters on the basis of the above specification is not very difficult and the use of numerical techniques is not necessary. Things, however, become much more complicated when we have floaters with embedded options. This are the types of products we are interested in this paper. We study two different types of embedded options. In the first case we assume that the product we are locking at is a bond with a coupon rate equal to the SMY that includes a call option by the issuer. In the second case we are looking at SMY floaters with variable caps and floors.

Some of these products have the following characteristics. There is an initial interest rate r_0 that applies as long as the SMY_t (as reference interest rate) does not hit a cap or a floor rate that are specified symetrically around a basis rate r_B:

$$| SMY_t - r_B | \leq k$$

If on the contrary

$$| SMY_t - r_B | > k$$

holds, the interest rate that applies changes according to the updating rule

$$r_{t+1} = r_t + f(SMY_t - r_B)$$

where $0 < f < 1$ is a factor of adjustment and the new level of the variable interest rate r_{t+1} depends on the previous level corrected for an adjustment term, and the basis rate is updated to the value of the reference rate: $r_B = SMY_t$.

The numerical treatment of those two products is quite different. While in case of the call option the product is not path dependent but needs to be calculated recursively since at each node we need to evaluate the whether or not it is optimal to call the bond, we have a strong path dependency in the case of the variable caps and floors, but we do not need to solve the tree backwards. The path dependency implies that analytical pricing is impossible so that we have to rely on numerical techniques. Given the large number of scenarios in the tree we therefore have to rely on Monte Carlo techniques. We use the Hull and White tree to run these Monte Carlo techniques.

We implement two different types of pricing procedures based on MC-techniques. One that involves solving the tree backwards. This is applied with embedded options. If we are at a node in the tree and we want to decide whether or not to exercise the option we need to calculate the price of the floater at the node and compare it to the exercise price. For calculating the price of the floater we use, however, MC-techniques for all the paths emanating from this node, thus establishing a second level of simulations. Figure 3 shows a sampled scenario, defined by the path emphasized. In order to compute the decision e.g. at node (2,1), in a a nested simulation, paths are sampled from the dotted area.

The other pricing technique involves sampling a path along the tree and checking if the floors or the caps are met, given the history of the interest rate. This sampling is repeated many times and the price of the product calculated on the basis of the mean of all sample paths.

In the next section we describe how these numerical techniques have been implemented and we present numerical results for products without and with embedded options.

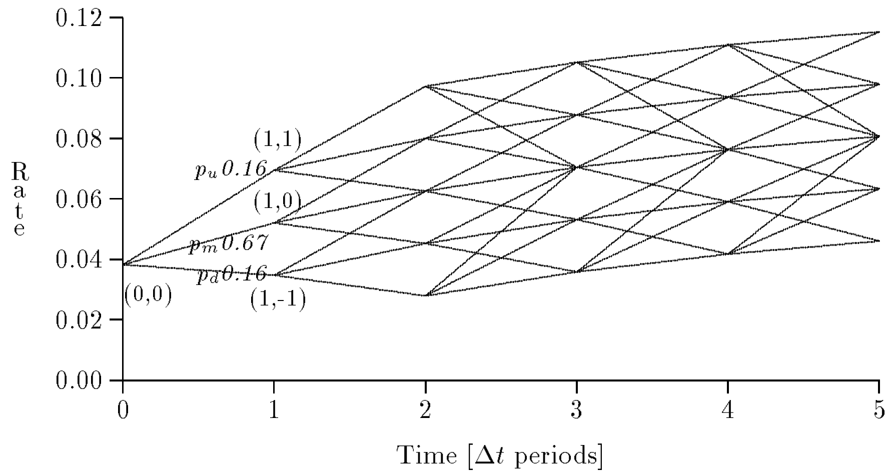

Figure 1: Hull and White tree for the Δt spotrate

4 Implementation and Results

4.1 Implementation

We start the implementation of the pricing tool using the Fortran 90 language. Fortran 90 supports a modular programming style with data abstraction while providing a vector notation for arrays. The parallel program version has been developed by annotating the Fortran 90 version with HPF and HPF+ directives. High Performance Fortran (HPF) is a Fortran 90 language extension which allows the programmer to explicitly express parallelism (see [8]). HPF+ provides further extensions of specific HPF features (see [2]).

The pricing tool consists of several modules which encapsulate the Hull and White tree, the financial instruments (e.g. bonds), the financial operations (e.g. discounting), and the Monte Carlo simulation, as outlined in Figure 2.

The *Hull and White* module contains the representation of the Hull-White interest rate tree and the tree building procedure. The *Financial Instruments* module specifies the instruments handled by the tool through functions describing the cash flows of these instruments. The *Financial Operations* module provides functions to compute the present value of a financial contract, based on given interest rates and future cashflows. The *Monte Carlo Simulation* module contains the driver routines for the Monte Carlo simulation.

The Monte Carlo Simulation module is the core of the pricing tool. It computes the price of a financial instrument as the present value at the root node of the Hull and White tree which contains the interest rates for the discounting

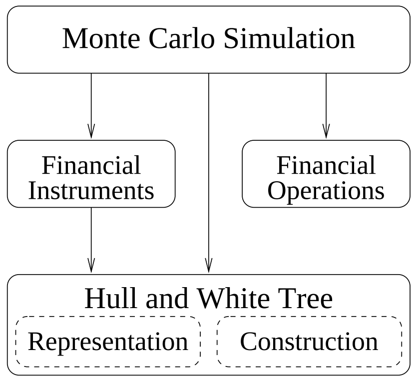

Figure 2: Pogram Modules

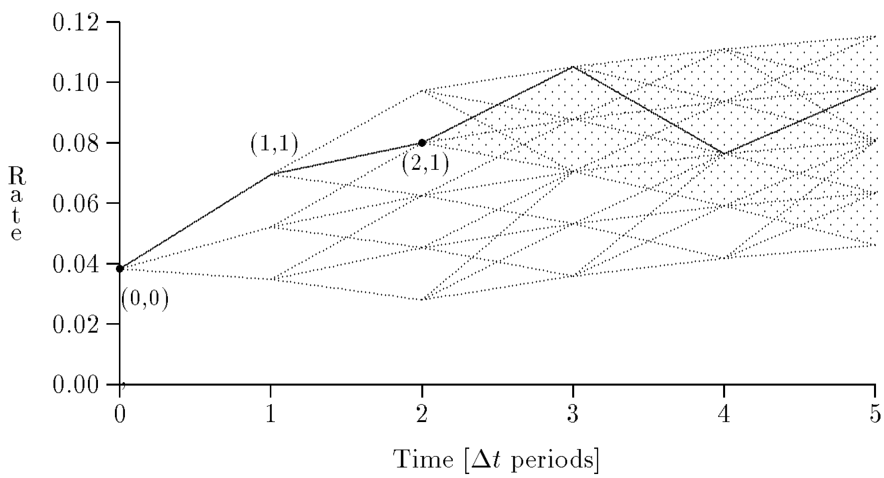

Figure 3: Sampled Path with Nested Simulation

process. The algorithm selects a number of N paths in the Hull and White tree from the root node to some final node (see Figure 3). Along each path, it iteratively discounts, backwards from the final node to the root node, the cash flow generated by the instrument along this path [1]. Discounting is performed using the discount factors along this path. The resulting price of the instrument is the mean value over all selected paths.

4.2 Parallelization

During the simulation, the information at the nodes of the tree is potentially used the by the computation of every path. This motivates a replication of the whole tree over all processors. The storage requirements for these structures are comparatively small and not critical in terms of local memory size.

According to the owner-computes rule for data parallel computation, the replication of the tree implies also a non-parallel execution of the tree construction. Measurements have shown, that this is a very small percentage of the total amount of computation, including the Monte Carlo simulation. Besides, there is communication overhead during a parallel tree construction. As described in section 2, the computation of the $Q_{m,j}$ contain dependencies, which limits the degree of parallelism achievable and induces communication. Then, after the tree construction, the partial results have to be communicated to all other processors, requiring many-to-many communication. For this reasons, the tree data structures as well as the tree construction procedure are replicated.

For the Monte Carlo Simulation sampling the process as well discounting along the paths can be done in parallel. Because all the path computations are independent from each other, they can be performed without communication. Every path computation has access to the whole tree. After processing the individual paths, the final price is computed through a summation of the path results over all processors. A reduction operation is used, which first computes partial sums on each processor simultanously, and then sends the partial results to a selected processor which computes the final sum. This is the only operation which requires communication.

4.3 Numerical results

Our numerical results are devided into two groups. In the first group we present the case of a coupon bond with a fixed or a variable interest rate and a call option. The variable interest rate is specified as the SMY where we make use of the assumption stated in Section 3 that the SMY is duplicated by a linear combination of spot rates.

[1] For variable coupon bonds, the cash flows are path dependent.

In the second group of results we derive the value of SMY floaters with variable caps and floors where the adjustment is given by the process outlined in Section 3. As already pointed out above these products are path dependent and hence do require Monte Carlo Simultion. For the products of the first group applying MC-techniques is - strictly speaking - not necessary. We include the simulation results, however, to demonstrate the differences between the simulated and the exact values of the products.

Tables 1, 2, and 3 show the prices for products in the first group. These products include fixed and variable coupon bonds, with and without a call option, and with maturities of 6, 8 and 10 years. The Hull and White tree is calibrated on the basis of the initial term structure given by $r_{Init}(t) = 0.08 - 0.05e^{-0.18t}$, and the are specified as $a = 0.1$, $\sigma = 0.25$, $\Delta t = 1$ year. For the SMY at time $m\Delta t$ we make use of the specification $SMY(m\Delta t) = 0.95r^{5\Delta t}(m-1,j) + 0.05r^{7\Delta t}(m-1,j)$, where j is the interest rate index of a predecessor node and $r^{5\Delta t}(m-1,j)$ is the five year spot rate.

The theoretical price in case of a fixed coupon bond is the result of dicsounting along the initial term structure, and exists for bonds without options only. As indicated above for the examples in the first group, it is possible to compute all the prices with backward induction techniques (see [9]), because only immediate predecessor nodes are needed for computations. For fixed coupon bonds with embedded call options, we calculate the price on the basis of the Black-Scholes formula, knowing that this is not correct. But we report this price as reference case.

In a next step we calculated the prices by making use of the full infomwtion given by the tree, i.e. we present results of traversing all paths, at the first as well as at the second level (in case of call options).

Additionally to the prices calculated by fully solving the tree once we report the results for different Monte Carlo Simultion runs. The sampling rates for these simulations are given as the percentage of the total number of paths [2]. For the nested version in case of options, the same percentage has been choosen for the first and the second level. For all simulation runs we report the follwing relative differences (i.e. errors): ρ (difference relative to the theoretical price), ρ' and ρ'' (differences to the price computed by traversing all paths at the first, and at the second levels).

Because of limited path dependency for products in the first group, it is possible to use the result of a nested simulation at a node, when this node is encountered later on, during the following simulations. Thus, at every node, a nested simulation is performed only once. The resulting prices are shown in the corresponding entries.

[2] The total number of paths is 3^n.

Fix coupon	*without option*	*with option*
theoretical	9332.61	
backward	9332.61	9131.09
Black-Scholes		9323.32
all paths	9332.61 ρ= $1.95{\cdot}10^{-14}$	9081.20
10%	9365.94 ρ'= 0.3571	9131.38 ρ'= 0.5526
1%	9548.40 ρ'= 2.3122	9285.55 ρ'= 2.2503
nested, all paths		9107.67
nested, 10%		9097.06 ρ''=-0.1165
nested, 1%		9171.73 ρ''= 0.7034
nested, 10% once		9047.72 ρ''=-0.6582
nested, 1% once		9302.12 ρ''= 2.1350
Variable coupon	*without option*	*with option*
backward	10537.83	10186.02
all paths	10537.83	10177.30
10%	10548.37 ρ'= 0.1000	10183.93 ρ'= 0.0651
1%	10597.04 ρ'= 0.5619	10193.15 ρ'= 0.1557
nested, all paths		10185.27
nested, 10%		10183.45 ρ''=-0.0179
nested, 1%		10191.01 ρ''= 0.0564
nested, 10% once		10180.59 ρ''=-0.0459
nested, 1% once		10192.16 ρ''= 0.0676

Table 1: Prices of 6 year bonds

The second group of products consists of floaters with varying caps and floors. For those product it is not possible to solve for the prices analytically. This would require in case of 20 time steps to solve for 3^{20} scenarios which is virtually impossible. Hence in that case a MC-simulation is the only feasible approach. Therefore in table 4 we present our result for this class of products. We specify the base rate as $r_B = 0.035$ the maximum level of deviation from the base rate without adjustment as $k = 0.03$, the adjustment factor as $f = 0.08$ and the initial rate as $r_0 = 0.05$. The results in case of the MC-simulation, on the basis of $\Delta t = 1$ month, vary substantially depending on how many samples we choose as a percentage of all possible scenarios. Although on the other hand quite substantial time reductions are possible if we look at a smaller number of sample paths (see table 5).

Fix coupon	*without option*	*with option*
theoretical	8883.49	
backward	8883.49	8703.52
Black-Scholes		8873.31
all paths	8883.49 ρ=-8.19$\cdot 10^{-14}$	8592.34
10%	8895.03 ρ'= 0.1299	8597.73 ρ'= 0.0627
1%	8882.95 ρ'=-0.0061	8622.23 ρ'= 0.3479
nested, all paths		8626.26
nested, 10%		8625.38 ρ''=-0.0102
nested, 1%		8602.90 ρ''=-0.2708
nested, 10% once		8622.27 ρ''=-0.0463
nested, 1% once		8557.21 ρ''=-0.8005
Variable coupon	*without option*	*with option*
backward	10540.12	10174.84
all paths	10540.13	10148.45
10%	10542.87 ρ'= 0.0260	10148.00 ρ'=-0.0044
1%	10542.28 ρ'= 0.0204	10152.76 ρ'= 0.0425
nested, all paths		10171.85
nested, 10%		10172.13 ρ''= 0.0028
nested, 1%		10170.63 ρ''=-0.0120
nested, 10% once		10172.46 ρ''= 0.0060
nested, 1% once		10172.16 ρ''= 0.0030

Table 2: Prices of 8 year bonds

Fix coupon	*without option*	*with option*
theoretical	8476.90	
backward	8476.90	8249.79
Black-Scholes		8466.05
all paths	8476.90 ρ=-4.29$\cdot 10^{-14}$	8150.92
10%	8475.38 ρ'=-0.0179	8147.16 ρ'=-0.0461
1%	8512.37 ρ'= 0.4184	8168.58 ρ'= 0.2167
nested, all paths		8179.09
nested, 10%		8178.26 ρ''=-0.0101
nested, 1%		8189.47 ρ''= 0.1269
nested, 10% once		8153.83 ρ''=-0.3088
nested, 1% once		8233.72 ρ''= 0.6679
Variable coupon	*without option*	*with option*
backward	10525.22	10164.51
all paths	10525.22	10116.92
10%	10524.61 ρ'=-0.0058	10116.56 ρ'=-0.0036
1%	10534.62 ρ'= 0.0893	10117.71 ρ'= 0.0078
nested, all paths		10160.34
nested, 10%		10160.27 ρ''=-0.0007
nested, 1%		10160.96 ρ''= 0.0061
nested, 10% once		10160.39 ρ''= 0.0005
nested, 1% once		10160.74 ρ''= 0.0039

Table 3: Prices of 10 year bonds

maturity	*12 months*	*18 months*
10%	15030.79	
1%	15025.08	
0.1%	15021.14	20531.19
0.01%		20518.46
0.001%		20514.67

Table 4: Prices of bonds with variable cap/floor

maturity	*12 months*	*18 months*
10%	15.0415	
1%	1.52592	
0.1%	0.195376	155.977
0.01%		15.5015
0.001%		1.60011
total number of paths	531441	387420489

Table 5: Execution times for bonds with variable cap/floor (in seconds)

4.4 Performance Results

We ran the parallel program on the Meiko CS-2 HA, a distributed memory system with up to 128 SPARC computing nodes, installed at the Vienna Center for Parallel Computation (VCPC) at the University of Vienna. The program has been compiled with the VFC compiler which makes use of the PARTI runtime routines for parallel loops (see [1], [5]).

Figure 4 shows the execution times of parallel MC-simulations for the products in the first group with 10% of the total 59049 paths sampled, running on a different number of processors. The parallelization is very efficient, because the communication overhead, induced only by the computation of the total sum, is very small. The computationally intensive part, namely the processing of the paths, is performed in parallel without communication. This results in a nearly linear speedup.

The comparison of sequential execution times for the different methods, with different parameters, in table 6 shows, that the added effort for the nested simulation is very high. Significant improvements can be gained through attempts to reuse simulation results once computed. It also shows, that the nested simulation behaves very sensitive with respect to an increasing number of timesteps.

5 Conclusions and Future Research

Investigations in the behaviour of the simulation algorithm have exposed sources of optimizations. They are based on the observation, that partial results can be reused later on. For example, some paths are choosen much more than once, depending on their probabilities. Storing path results and reusing them later on can save up to 80 percent of computation time.

Another approach exploits the fact, that a value at a particular node (m, j)

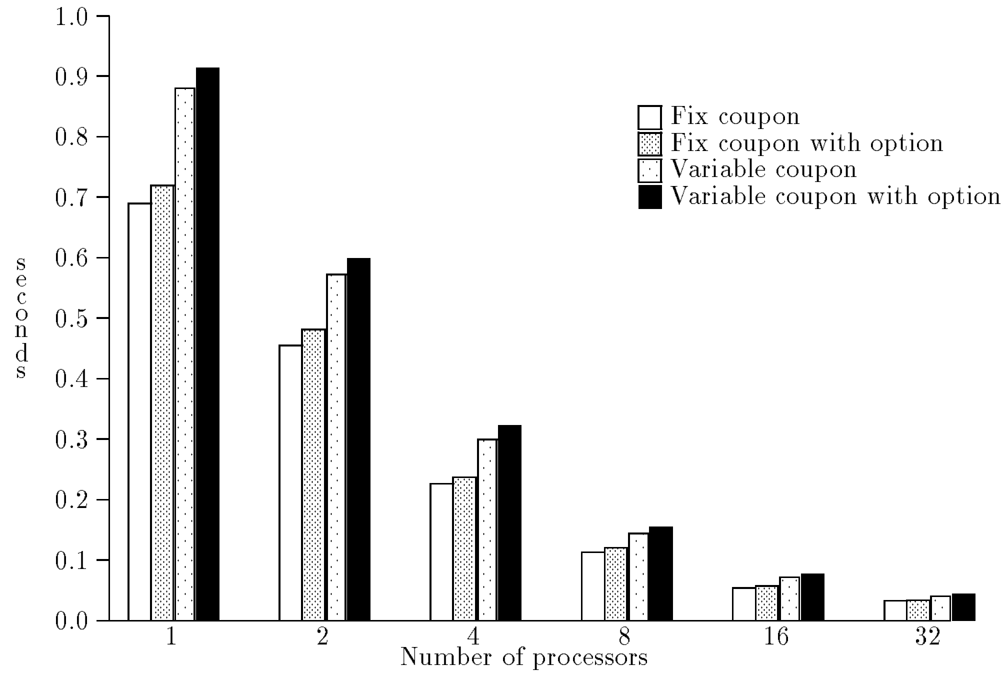

Figure 4: Execution times for $10\Delta t$ - Period, 10 % sample

Variable Coupon *with Option*	$10\Delta t$ - *period*	$8\Delta t$ - *period*	$6\Delta t$ - *period*
backward	0.108		
all paths	9.694		
10%	0.914		
1%	0.106		
nested, all paths	27614.018	2579.550	255.928
nested, 10%	4712.586	6.723	1.435
nested, 1%	110.993	2.543	0.428
nested, 10% once	14.655		
nested, 1% once	2.090		

Table 6: Execution times for different methods

of a path, computed once via a nested Monte Carlo simulation, can be reused also during the processing of another path containing (m, j), thus saving an additional nested simulation at this node. This is valid only if the cash flows at the paths emanating from node (m, j) are independent of its predecessors. Some numerical results and sequential computation times have been presented in the previous sections.

The parallel implementation of these optimizations raises a number of synchronization, communication, and load balancing problems. For example, results must be sent as soon as possible to all the processors potentially using it. This can lead to costly many-to-many communication. Various techniques can be applied to overcome this bottleneck, including the assignment of disjoint set of paths to individual processors in a separate phase, prior to the path computations. Future work will concentrate on finding an efficient algorithm which computes these sets without requiring much communication.

References

[1] S. Benkner, K. Sanjari, V. Sipkova. *VFC - A Compiler for HPF+.* Esprit Project HPF+, Deliverable D2.2c. University of Vienna, March 1998.

[2] S. Benkner, E. Laure, H. Zima. D2.1c: *HPF+: An extension of HPF for advanced industrial applications.* Esprit Project HPF+, Deliverable D2.1c. University of Vienna, February 1998.

[3] P. Boyle, M. Broadie, P. Glasserman. *Monte Carlo methods for security pricing.* Journal of Economic Dynamics and Control, 21: 1267 - 1321 1997.

[4] L. Clelow, C. Strickland. *Implementing derivative Models.* John Wiley & Sons, 1998.

[5] R. Das, J.Saltz. *A manual for PARTI Runtime Primitives - Revision 2.* Internal Research Report, University of Maryland, Dec. 1992.

[6] E. Dockner, H. Moritsch, G.Ch. Pflug and A. Swietanowski. *The AURORA financial management system: From Model Design to Implementation.* Technical report AURORA TR1998-08. University of Vienna, June 1998.

[7] F.J. Fabozzi, T.D. Fabozzi. *The Handbook of Fixed Income Securities. Fourth Edition.* Irwin Professional Publishing, 1995.

[8] High Performance Fortran Forum. *High Performance Fortran Language Specification. Version 1.0.* May 3, 1993.

[9] J. C. Hull. *Options, Futures, and Other Derivatives.* Prentice Hall, April 1997.

[10] J. C. Hull, A. White. *One Factor Interest Rate Models and the Valuation of Interest Rate Derivative Securities.* Journal of Financial and Quantitative Analysis, 28: 235-254. 1993.

[11] F. Jamshidian. *Forward Induction and Construction of Yield Curve Diffusion Models.* Journal of Fixed Income, 1: 62-74. 1991.

[12] S. Pichler. *Bewertung von Zahlungsströmen mit Variabler Verzinsung.* Habilitationsschrift. Vienna, September 1998.

[13] L.C.G. Rogers, E.J. Stapleton. *Fast accurate binomial pricing.* Finance and Stochastics, Volume II (1998), issue 1. Springer, 1998.

[14] S. A. Zenios. *Parallel Monte Carlo simulation of mortgage-backed securities.* In S. A. Zenios, editor, *Financial Optimization.* Cambridge University Press, 1993.

[15] Y. Censor, S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications.* Numerical Mathematics and Scientific Computation. Oxford University Press, 1997.

[16] Zhong Ge. *A Numerical Study of One-factor Interest Rate Models.* Graduate Department of Computer Science. University of Toronto, 1998.