Bloomberg

Speeding up VAR with Smart Monte Carlo

Jan W. Dash, PhD
Xinchong Zhang

February 2018

Table of Contents

Abstract

We show that VAR calculation speedup of an order of magnitude can be obtained using Smart Monte Carlo with a sophisticated interpolator. As a byproduct, we give some encouraging numerical results for evaluating N-dimensional Gaussian integrals without doing any integrals at all.

Executive Summary

We present "Smart Monte Carlo" or SMC, previously introduced to finance by Dash and Yang[i], aimed here at speeding up Value at Risk (VAR) calculations. Preliminary results with Historical VAR (HVAR) give a speedup factor of 10-20, with good accuracy (a few % or better for 99% CL 1-year HVARs: asset class, total, marginal). The speed-up factor is bigger for longer-term HVAR (e.g. 10 year HVAR). SMC is an adaptive method applicable over many time steps or over only one time step, as is the case with VAR. Deals are evaluated one by one. The basic idea uses one-step real-world scenarios or simulation to generate VAR "paths", depending on the deal. Risk-neutral pricers (that must be provided) are called only at a limited number of points, "from the outside"[1]. A critical ingredient is a refined n-dimensional Cauchy interpolator that can handle non-linearities in order to get results at points where the pricers are not called. The interpolation is much faster than calling the pricer. A simplification involves only using the interpolator in one dimension that is "most important", and using Greeks for the "less important" dimensions. Difficulties include the suppression of oscillations and dealing with discontinuities. Our tests use a portfolio consisting of an auto-callable option, a down-out option with rebate, and liquid options chosen to make the portfolio delta and gamma neutral. We also present some ideas aimed at complex basket knock-out options.

We also present some numerical results that explore approximations to N-dimensional Gaussian integrals that do not involve doing ANY integrals, previously suggested by Dash[ii]. This potential enables further applications of speed-up for both risk calculations (i.e. outside the pricers) and pricing (i.e. inside pricers).

A number of appendices (see table of contents) give details.

Introduction to Smart Monte Carlo - SMC

SMC is an adaptive method that can be used to perform pricing and risk, based on combining ordinary scenarios and Monte Carlo MC with path integral techniques[iii]. It produces speed-up in documented cases, with similar accuracy to ordinary MC. Here, an application of SMC to calculate VAR for a portfolio is given.

Here is a summary recipe format for one deal.

1. "*Scout Path or Scout Point SP*" generation [scenarios and/or standard MC with limited number of paths giving deal prices for a restricted number of values in the underlying variable(s)]

2. "*Brownian Bridge BB*" generation [uses path integrals based on the SP results, providing deal prices in "interesting" regions where the deal lives; this can be skipped in the simplest

---

[1] Smart MC can also be employed inside pricers, although that is not the focus here. See the third paper in ref. i for example.

implementations, and is skipped in this paper]

3. *Interpolation at fixed time* [other deal prices based on 1 and 2 without calling the pricing function]. We use a sophisticated n-dimensional interpolator with a "Cauchy" form. These generate "look-up tables" of results.

The SMC speed-up is due to not using the pricers for most of the points, instead using the deal prices from SMC first stage look-up tables. Statistics are carried out by combining results for deals in portfolios. The Appendices give some details.

Connecting Value at Risk VAR and SMC

VAR by its nature can be viewed as a one-step real-world simulator with frozen deal life-cycle kinematics. For example, Historical VAR (HVAR) uses historical one-day differences applied to the current market, in order to get one-day scenarios from "today" to one day ahead. If we use 10-day historical differences we logically would get 10-day scenarios from today to 10 days ahead. The portfolio value changes under these scenarios are used to calculate HVAR (see e.g. iii, Ch. 26)[2].

Pictorially the SMC technique applied to VAR uses that uses one time step is shown in Figure 1 below (for more information see Appendix 4).
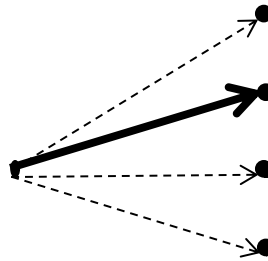
Figure 1. SMC for VAR viewed as a single time-step real-world simulator. The bold-face arrow indicates a particular scenario

---

[2] The life-cycle events of the deals in the portfolio are frozen at today's configuration and no cash-flow events are included (e.g. time to exercise, fixings, etc. are frozen). However if for example the deal changes character when a scenario is applied (e.g. knockout), the pricing function will take this "spatial" transition into account.

SMC VAR Testing

Down & Out Option with rebate

To test the efficiency and feasibility of the SMC techniques, we use a European down-out (DO) option with a rebate[3]. The pricer (a closed form solution) generates the points in red taken as equally spaced (these are the "Scout Paths"). The interpolator is used to get the rest of the points. The most difficult feature to describe involves the discontinuity from the DO phase above the barrier to the constant rebate below the barrier. The description is quite accurate. The steps in getting the interpolator are described in Appendix 3.
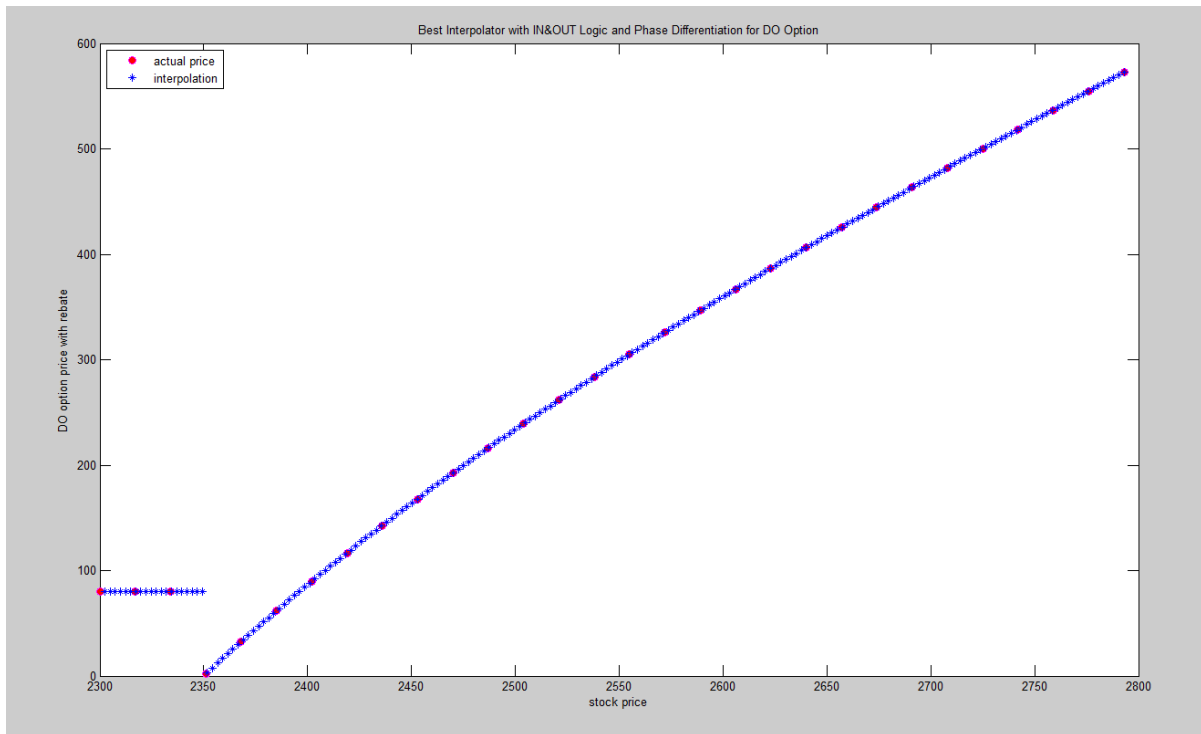
Figure 2: Down-out option evaluated at a subset of points in red and interpolated with the interpolation function described in Appendix 3.

---

[3] **Deal Details**
  - Down and Out European Call Option with rebate on SPX
  - Strike: 2,450
  - Barrier: 2,350
  - Maturity: 5 years
  - Rebate: 80
  - Other parameters
    - Initial stock price: 2,400
    - Annualized interest rate: 2% (real-world parameter)
    - Annualized stock volatility: 8.14% (real-world parameter)
    - Time interval: 10 days

Autocallable Option

Autocallable options are both popular and difficult and expensive to price. We use a hypothetical deal described in the box below:

Deal Parameters		Float Funding Parameters	
Mode	Swap	Notional	5,000,000.00
Direction	Rec Funding Pay Option	Index	US0003M Index
Notional	5,000,000.00	Currency	USD
Funding Type	Float	Tenor	3 MO
Currency	USD	Leverage	1.00
Initial Fixing Type	Fixed	Spread	100.00 bp
Effective Date	01/02/2018	Frequency	Quarterly
Expiry Date	06/28/2019	Daycount	ACT/360
Maturity Date	07/02/2019	Final Redemption	
Standard Payoff Parameters		Performance Type	Worst
Final Payment	100.00%	Redemption Payoff	Standard Payoff
Participation	100.00%	Final Barrier Type	Down & In
Floor	0.00%	Final Barrier	Discrete
Coupon		Low Barrier	80.00%
Coupon Type	High/Low	Early Redemption Parameters	
High Coupon Amount	3.00%	Call Barrier	120.00%
Low Coupon Amount	0.50%	Call Amount	0.00%
Low Coupon Amount	0.50%	Call Amount	0.00%
Fixed Periods	1	Call Amount Increment	0.00%
Coupon Barrier	80.00%	Frequency	Quarterly
Frequency	Quarterly	Observation	
☐ Coupon Per Annum		Frequency	Daily
▬ Basket			
Ticker	Initial Fixing		
⊙ SPX Index	2,673.61011		
Add entry			

Figure 3: The Autocallable definition

Speed-up Factors

Here are the results for the Autocallable using SMC vs. using full revaluation (reval). One year of data was used to calculate the P&L results and the VAR at 99% and 95% CLs. We look at two cases: 20 revals (scout paths) and 10 revals (scout paths) for 1 year VAR with 1-day differences (264 days or 263 differences). Hence the speed-up factors are $263/20 = 13$ and $263/10 = 26$ in the two cases, ignoring the interpolation which is much faster than the pricer. If we were to consider 10 year VAR, the expected speedups would be an order of magnitude greater. We have not, however, examined this case in detail.

(1) Autocallable Case 1: 20 scout points evenly spaced
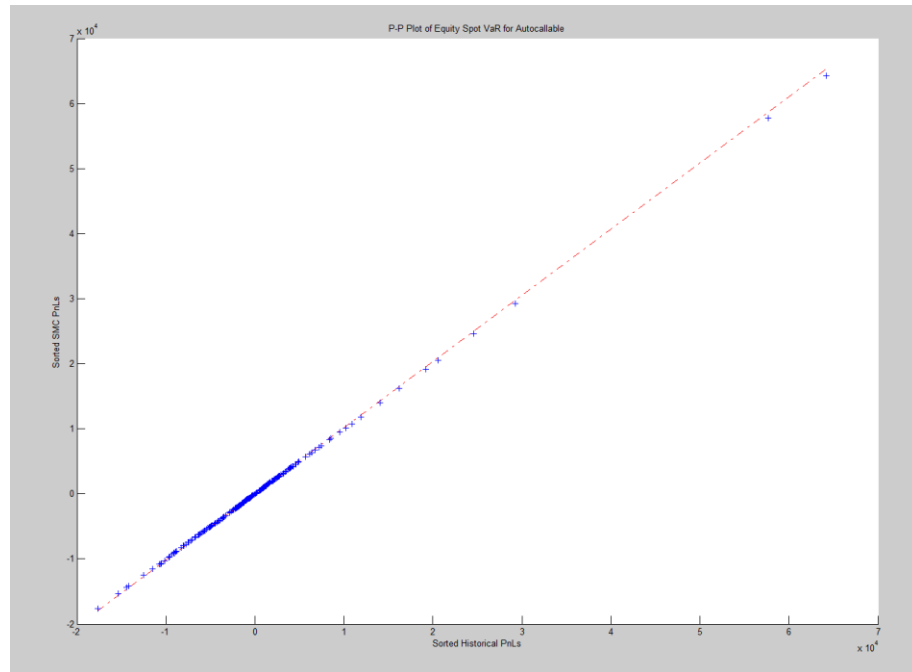
Figure 4: The pp plot for stock price change P&L – SMC vs. Full Reval
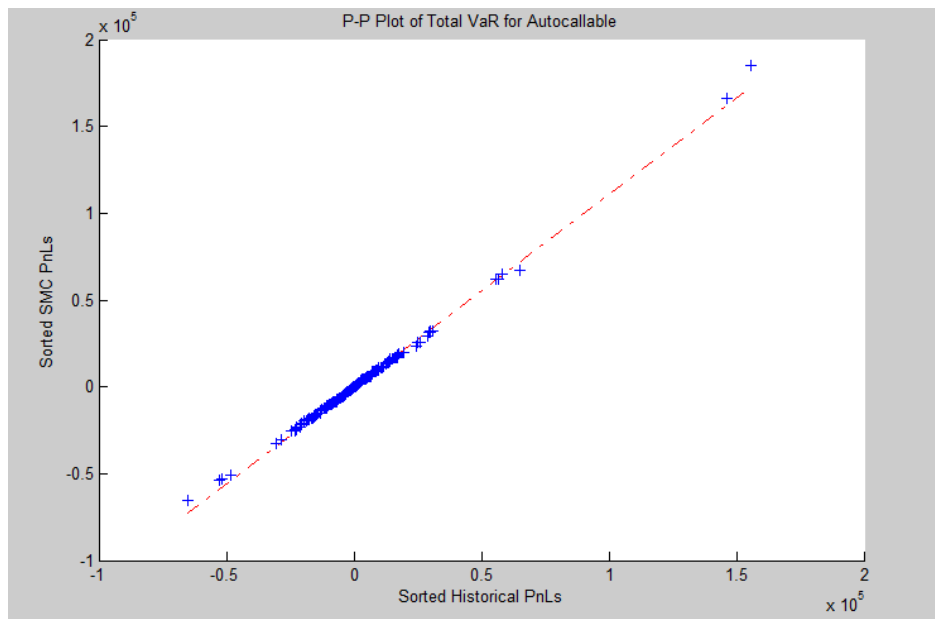
Figure 5: The pp plot for Total P&L – SMC (using Greeks for vega and rho) vs. Full Reval

(2) Autocallable Case 2: 10 scout points evenly spaced
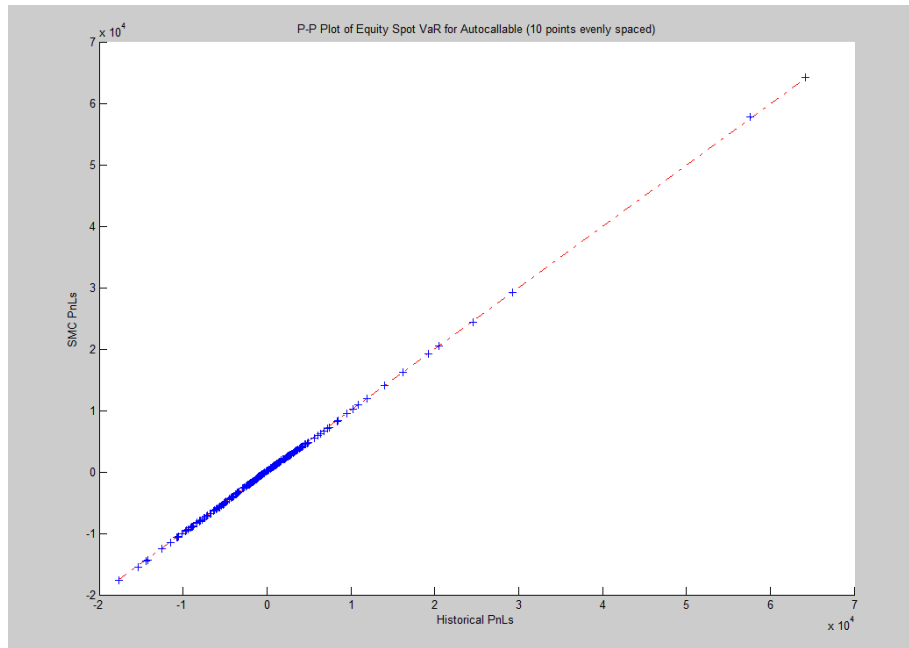
Figure 6: The pp plot for stock price change P&L (SMC vs. Full Reval)
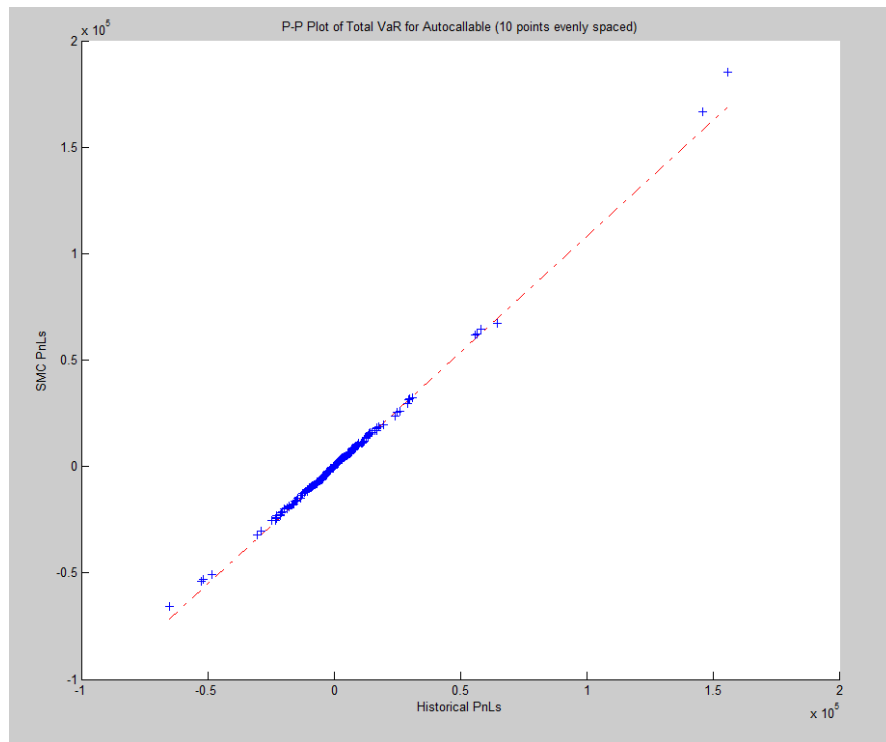
Figure 7: The pp plot for Total P&L – SMC (using Greeks for vega and rho) vs. Full Reval

Numerical results for one-year HVAR for the autocallable

The table below gives the results for one-year HVAR for the Autocallable, at 99% and 95% confidence levels. We show the full reval (exact) result and the SMC model results with 10 and 20 scout path (points). The results are generally quite close. The SMC model results for total VAR are obtained by adding the Greek contributions for vega and rho to the interpolated equity result. Thus the interpolation is done in one dimension.

	Full Reval	SMC (10)	SMC (20)
99% Equity Spot VaR	14,403	14,570	14,375
Diff	--	1.16%	0.19%
95% Equity Spot VaR	9,570	9,487	9,591
Diff	--	0.87%	0.22%
99% Total VaR	51,821	53,115	52,962
Diff	--	2.5%	2.20%
95% Total VaR	21,027	22,935	22,899
Diff	--	9.07%	8.90%

Table 1: Comparison of results for SMC with 10 and 20 points vs. full reval

Marginal HVaR

We created a portfolio with the autocallable, the down-and-out option with rebate, and two vanilla options on SPX Index tuned to make the portfolio basically delta and gamma neutral. Here are the details of the portfolio.

Security Description	Amount	Base $Value (no scenario)	Equity Delta	Equity Gamma
AutoCallable	1	(204,863)	(11,097)	1,943
Down&Out with rebate	1000	136,881	13,012	1,071
Liquid option 1	42	(482,502)	(61,722)	(4,157)
Liquid option 2	38	1,269,507	60,759	1,335
PORTFOLIO		719,022	951	192

Table 2: Portfolio used for Marginal VAR

Marginal HVaR results (SMC 20 points evenly spaced):

Table 3 below gives Marginal VAR results at 99% confidence level for the portfolio above when the Autocallable is present/absent. The accuracies are reasonable.

	99% Equity Spot VaR (Full Reval)	99% Total VaR (Full Reval)	99% Equity Spot VaR (SMC)	99% Total VaR (SMC)
With Autocallable	1,955	29,928	2,007	30,329
Without Autocallable	31,232	64,679	31,232	64,679
Marginal	-29,277	-34,751	-29,225	-34,350
Diff	--	--	0.2%	1.2%

Table 3: Marginal VAR results for the portfolio above when the Autocallable is present/absent

Approximate Analytic Evaluation of Multivariate Gaussian Integrals

Approximate analytic evaluation of multivariate Gaussian integrals has myriad potential applications in both risk and pricing. The original theoretical work was due to Dash [ii]. See Appendix 8 for some details. Here we present some numerical results. We have not done exhaustive testing. Results were obtained using equal upper limits for all variables. Other regions were not examined. We also only use the first term in the perturbative expansion[4].

We use data for five US equities: AAPL US Equity, F US Equity, GE US Equity, IBM US Equity and TSLA US Equity, (thus this is a 5-dimension integral). The original 5x5 correlation matrix ρ is:

$$\rho = \begin{pmatrix} 1 & 0.0667 & 0.0757 & 0.1289 & 0.2170 \\ 0.0667 & 1 & 0.4315 & 0.3101 & 0.1693 \\ 0.0757 & 0.4315 & 1 & 0.3297 & 0.0719 \\ 0.1289 & 0.3101 & 0.3297 & 1 & 0.0904 \\ 0.2170 & 0.1693 & 0.0719 & 0.0904 & 1 \end{pmatrix}$$

The approximate matrix ρ_f is generated by the vector c_i (see Appendix 8)

C_1	C_2	C_3	C_4	C_5
0.3494	0.4944	0.4767	0.4634	0.3703

We get

$$\rho_f = \begin{pmatrix} 1 & 0.1727 & 0.1665 & 0.1619 & 0.1294 \\ 0.1727 & 1 & 0.2357 & 0.2291 & 0.1831 \\ 0.1665 & 0.2357 & 1 & 0.2209 & 0.1765 \\ 0.1619 & 0.2291 & 0.2209 & 1 & 0.1716 \\ 0.1294 & 0.1831 & 0.1765 & 0.1716 & 1 \end{pmatrix}$$

The eigenvalues of the matrix ρ_f are:

$$\alpha_1 = 0.7626,\ \alpha_2 = 0.7797,\ \alpha_3 = 0.8400,\ \alpha_4 = 0.8721, \alpha_5 = 1.7457$$

Numerically ς^* is:

$$\varsigma^* = -0.4433$$

Formula 1~4 are calculated numerically with $x_i^{max} = 1, i = 1,2,3,4,5$. The limits are thus -infinity to

---

[4] We also tried adding the next (beta=1) term in the expansion. and using a Pade approximation including the beta=1 term. But the results here were not improved over just the beta=0 term. However the Beta=1 term may help when different variables have different upper limits. We leave this for future work.

one, the same for all five integrals. The other column headings mean:

1. Exact - with original correlations, numerical integration
2. Correlation Approx - with modified correlations, numerical integration
3. Decrease dimension - modified correlations, 1D integral with integrand product of cumulative normal distributions
4. Saddle point - modified correlations, saddle point approx to 1D integral

The results for this 5x5 case are presented in Table 4 below:

	1 – Exact	2 – Correlation Matrix Approx	3 – Decrease Dimension	4 – Saddle Point Approx
Numerical Results	0.48839	0.48383	0.48382	0.48383
Errors compared with exact	--	0.929%	0.930%	0.929%

Table 4: Results for the 5x5 integral approximations defined above

20-Dimensional Tests

We tested different (but equal) limits for 20 variables[5], including equities, indices, commodities and bonds. The results are below and are all reasonably close. The errors are mostly due to the original correlation matrix approximation. Little difference exists between different numerical procedures given the correlation matrix approximation.

Limits	Exact	Correlation Matrix Approx	Decrease Dimension	Saddle Point Approx
Limit = -1	2.04 e-09	2.16 e-09	2.20 e-09	2.20 e-09
Limit = -0.5	1.54 e-06	1.41 e-06	1.41 e-06	1.41 e-06
Limit = 0	0.000236	0.000199	0.000200	0.000200
Limit = 0.5	0.008457	0.007219	0.007216	0.007208
Limit = 1	0.08825	0.07863	0.07863	0.07846
Limit = 2	0.6802	0.6639	0.6640	0.6665
Limit = 3	0.9749	0.9738	0.9738	0.9758
Limit = 4	0.99938	0.99938	0.99937	0.99945
Limit = 5	0.999994	0.999994	0.999994	0.999947

Table 5: Results for 20 dimensional integrals with the approximations defined above

---

[5] 20 variables is near the size limit for Matlab.

Appendix 1: Smart MC vs Standard MC

SMC Meta Code

1. For a given deal, generate Scout Paths SPs (deterministic and random at VAR time step)
2. Call the pricer for each SP
3. Interpolate the results for all SPs
4. Analyze the results for "interesting features" (e.g. see in-out procedure)
5. Generate more points around regions of interest
6. Determine if the probabilities are large enough to warrant additional pricing at each region of interest; this uses path integral Brownian Bridge concept
7. Call the pricer at the above-determined interesting points
8. Interpolate the new results
9. Determine if accuracy sufficient
10. If accuracy not sufficient, iterate above procedure
11. If accuracy sufficient, stop

Actually in this paper we only used the first three of these points for the prototype results.

SMC procedure - details

There is a well-known problem with Monte Carlo simulations. MC simulations start with stochastic equation(s) modeling the dynamics of underlying variables (stock prices, rates etc). Here is the problem. The instruments to be priced and/or risked have discounted cash flows CFs depending on these underlying variables, but whose profiles do not line up with the MC paths very well. An example is an out-of-the-money OTM option for which the CFs are mainly in regions with a relatively small proportion of MC paths. To get accurate results (given a stochastic model), a large number of MC paths must be run, and the option pricing function is called for every path. This is inefficient, because the option price on many (most) paths is simply negligible. This inefficiency is the problem we address with SMC. Results equivalent to MC simulation accuracy are obtained with speedup of a factor of at least 2.

Here is a list of relevant points in applying SMC. We have included a large number of ideas; only the most simple of which were implemented in this paper.
1. Work at deal level
2. **SCOUT PATHS OR POINTS**: Scout paths (points) for SMC are generated up to a pre-determined scout path budget. Pricing is done for each scout path. The pricing is done for market variables at the VAR time step but with frozen deal lifecycle parameters. Here, scout paths are generated in two ways: random and deterministic scenarios, This procedure generalizes the original procedure using only random scout paths (refs). The deterministic scenarios are described below and are chosen judiciously. We also generate additional random scout paths as in the original SMC procedure, using a real-world simulator with empirical parameters.
3. **DETERMINATION OF "IMPORTANT" AND "UNIMPORTANT" VARIABLES:** The idea here is that for e.g. equity deals, the equity variation is the most important for changes in the security. The important variable is carefully modeled including the non-linearities described by the sophisticated interpolator. Other variables (e.g. volatility and

interest rates) are less important. The variations in the security from these variables is simply approximated by Greeks (e.g. vega and rho). We use this idea in the paper.

4. **PRICINGS AT UNIFORMLY SPACED POINTS BETWEEN MIN, MAX (1 DIMENSION)**: This is the simplest implementation of scout paths (points). We use this implementation in this paper.

5. **PRICING AT POINTS THAT PRODUCED YESTERDAY'S VAR:** The idea here is that, since we know the state that produced yesterday's VAR, we can use that state today (or the closest state to it) to define a scout path point to perform one of the revaluations. In that case we may need to eliminate another point to avoid unwanted oscillations.

6. **PRICINGS AT STRESS TEST POINTS[6]**: In general stress tests have to be run, in addition to VAR. These can be run before the VAR, essentially constituting a subset of deterministic scout paths. Because they are stress tests, these points will also be relatively extreme, and so can help the stability of the interpolation.

7. **DEFINE Asset Class Tail Greek ACTG scenarios**: For given deal, generate "Asset Class Tail Greeks" or ACTGs. These are defined using a relatively large move (e.g. 2 stdevs) for all variables within an asset class (e.g. stocks or FX etc). Both positive and negative moves should be taken. Previous experience running VAR suggests that for a diversified portfolio, VAR even at a very high (99.97%) CL generally involves individual variable returns not further out on their tails than 2 stdevs. Therefore if a deal is in a diversified portfolio, and if the deal is an important component of VAR we still do not expect that the portfolio VAR will entail individual return moves much over 2 stdevs.

8. **DEFINE Individual Tail Greek ITG scenarios**: If an ACTG is large, individual variable moves within that asset class can be moved +- 2 stdevs for refinement, defining "Individual Tail Greeks" or ITGs, moved in the same direction as the large ACTG. If the ACTG is small, no further refinement should be necessary.

9. **Maxima scenarios:** For insurance, we can take the day's return with the maximum value of a variable return if the ITG is large for that variable. It is not clear that this will be necessary however.

10. **Yield Curve YC scenarios**: Besides parallel shifts, we can perform stylized tilts (e.g. +- k tilt stdevs around the 5-year point) for the forward rates. This avoids complications of principal components PCs and also avoids complexities due to the time dependence of PCs. Insight can be gained by analyzing the statistics for YC moves in the period of time required for VAR.

11. **Deal-dependent scenarios**: These can be formulated as needed. Insight can be obtained by running backtests on deals with historical market changes[iv]. Changes in the deal are regressed against historical changes of a given risk factor, essentially giving the sensitivity or delta to that risk factor empirically. In this way, those risk factors important to a given deal can be obtained, and insight gained. Scenarios for specific types of deals are in Appendix VI.

12. **Random Scout Paths:** We also generate additional random scout paths as in the original SMC procedure, using a real-world simulator with empirical parameters. The idea is to take advantage of known systematics of real-world risk in scout paths for efficiency in HVAR calculations where the HVAR scenarios are taken from the real world. By using a real world simulator we improve in a statistical sense the sampling of real-world risk variables closer to those we want for VAR.

13. **ADDITIONAL PRICINGS AT "INTERESTING" POINTS FOR THE DEAL**: We call "phase space" the n-dimensional space of returns. The deal may have "interesting" points or regions in phase space. These interesting points can be barriers, places where the deal varies rapidly, etc. Interesting points are detected in first approximation by the pricings from the scout paths. Decisions about whether or not to perform additional deal pricings can be automated by iterative routines (TBD) and considerations described in Appendix V, along

---

[6] I thank Adam Litke for this suggestion

with satisfying Brownian Bridge (BB) VAR constraints described below.

14. **The "In-Out" procedure**[6] – All points at which the deal has already been priced are interpolated using the interpolation function[7]. This interpolation is compared with the interpolations obtained by leaving out each point successively. Large differences can be associated with large changes, thus with interesting points to examine with further pricings. In this procedure, points that give near-zero prices for the deal are uninteresting and so should be skipped over in this procedure.

15. **BROWNIAN BRIDGE PORTFOLIO VAR CONSTRAINTS:** The BB constraints have to do with the portfolio VAR in which the deal exists[8], as follows. Generate Brownian Bridges in that (those) region (regions) of phase space where "interesting" aspects of the deal are found, ending after 1 day (or 10 days) at a putative pricing point with some predetermined phase space volume (bin size). Calculate the probabilities of these BBs, P(BB). The probabilities will give insight as to whether or not we expect to find real-world historical differences in the risk factors in that region. For example if a P(BB) is quite small (e.g. a region of phase space "on Mars"), we do not expect to find historical returns in the BB bin, so details of the deal – even if interesting – are unlikely to be sampled in the VAR calculation from historical returns, so in this case the deal is not repriced[9]. Thus, interesting regions for the deal but that are irrelevant for VAR are avoided for repricing the deal, saving time. This logic is empirical and needs to be tested and evaluated on sample portfolios. It will work better for diversified portfolios and less well for non-diversified portfolios.

16. **Additional Pricings using Brownian Bridges BB:** This is the second part of SMC, after the Scout Paths are determined (random and deterministic, described above). BBs are generated in neighborhoods of "interesting points", and if the BB VAR constraint described above is satisfied, the deal is repriced at that BB.

17. **Iterate** – The above procedure can be iterated, if consistent with the time budget.

18. **Final Interpolation:** Having finished pricings with the deal pricer, we obtain the values of each deal at all points in phase space of the VAR (e.g. HVAR) returns using the interpolation function.

19. **Repeat for other deals**

20. **Calculate VARs** for the various portfolios required.

---

[7] Thus we interpolate at intermediate steps to get a better idea of feature recognition for the deal in order to help to decide where to put more points.

[8] There may be several portfolios in which a deal exists, and the BB constraints are portfolio dependent.

[9] Some previous experience with BBs in SMC indicated that placing BBs in asymptotic regions helped the stability of interpolation.

Appendix 2: Smart Monte Carlo – SMC Details

Scout Paths (SP) - First Step in SMC

The first step is to get an idea of what the deal "looks like" by generating some scenarios and standard real-world MC paths, which we call "scout paths" or SPs. The pricing function is called for the SPs and the information is returned. It is important to note that for VAR application of SMC, we are not inside the deal pricing function, but rather call the pricing function from the outside.

Asset-Level Greeks as scenario scout paths for high dimensional problems

Asset-level Greeks can be useful for initial information for scout paths for deals that depend on a large number of variables, so the dimension of the problem is high. We are supposed to have a list of all variables that enter in the deal pricing, although we do not have the deal term sheet so we do not know the nature of the deal.

The idea is straightforward. We want to know, for example, if interest rates or stock prices or FX play a substantial role in the pricing. There may be many stock prices, interest rates, and/or FX rates on which the deal depends. However one asset class may dominate. We calculate Greeks for parallel movements of each asset class separately (stock prices, interest rates, FX). In this way we can get an overall idea of the importance of each asset class. Not many calculations are needed, far fewer than calculating all the Greeks for all the variables.

We can then refine this initial analysis by calculating the Greeks for all the members of the asset class on which the deal depends.

For example, stock price movements may dominate the risk. We find this out using asset level Greeks, and then calculate the Greeks for all the stocks. We do not calculate Greeks for any of the individual members of the other asset classes.

We can organize the calculation using principal components, although this is not necessary.

Following this initial calculation we can generate scout paths along the directions indicated by the Greeks of highest importance. For example say two stocks and one vol dominate the deal risk. We generate scout paths along these three dimensions. In this way we achieve dimensional reduction. For the other dimensions we can use the present values, a scenario of forward values, parallel shifts etc.

Path Integrals/Brownian Bridges (BB) – Second Step in SMC

The second step is to use the scout paths as information to generate more efficient sampling of the deal value. This is the core idea of our resolution to the MC inefficiency problem. Consider an OTM[10] option. We would like to send paths into the OTM region without wasting time generating paths that are in regions where the deal value is negligible for the purposes or pricing/risk. We can achieve this goal using path integrals and Brownian Bridges.

This is necessary for SMC applied to VAR because we interpret VAR as a one-step simulator.

Interpolated Brownian Bridges (IBBs) – Third Step in SMC

The third step is that we carry out interpolation for points sampled neither by SPs nor BBs. We term a

---

[10] OTM means out of the money with respect to today's price.

path for which the interpolation takes place as an "interpolated Brownian Bridge" or IBB. This enables an effective "pricing" for all values of the underlying variables without calling the pricing function "too many times".

Call x* a point at which we want to interpolate the deal with price C. We get C(x*) from the SP and BB values: $\left\{C(x_b^{(SP)})\right\}$, $\left\{C(x_b^{(BB)})\right\}$. To do this we need a distance measure and an interpolation function, discussed below.

SMC Stage 2 – Running the Portfolio

Setup for Portfolio run

The second stage for SMC for VAR (in the HVAR framework) uses observed historical changes. However, this second SMC stage never calls the pricing function, but rather uses the values of deals using the output of the three steps of the first SMC stage, namely price "look-up" tables that result from the prices from the SPs, BBs, and interpolation. Both stages constitute "Smart Monte Carlo".

Full MC simulation of the SMC Stage 2

Consider a portfolio that consists of deals $\{C_1\}$. In stage 1, we create the look-up tables for individual deals. Again, a look-up table lists the deal values produced from the interpolation for all points on a regular grid created in the space of the risk factors. Because the interpolation is conducted in a stepwise fashion, the number of look-up tables for the prices of a given deal equals the number of time steps in the simulation. VAR only needs one time step. The collection of the look-up tables provides the deal values on any path that has evolved arbitrarily. It is this feature that leads to substantial time saving in Stage 2 that only looks up deal values using the tables without either calling the pricer or doing the on-the-fly interpolations. Note that Stage 2 analysis does not need to compute the SMC probabilities. In a standard Monte Carlo simulation, all paths are equally weighted with a total weight of 1.

The general procedure for the Stage 2 analysis consists of the following steps:
1. Run SMC Stage 1 analysis for each deal C_1. The SMC is customized for each type of deal using the strategies described above.
2. Create the SMC Stage 1 look-up tables for each deal $\{C_1\}$.
3. Generate standard paths for Stage 2.
4. Look up values for each deal along each path, which involves looking up through a table at each time in the time partition. For VAR there is only one time step.

Deal/Risk aggregation (Stage 2)

In Stage 2, the desired portfolio risks are calculated by aggregation - summing up the risks of individual deals for all MC paths. For each time step, a MC path provides all the necessary information in terms of the values of the underlying risk factors that are used to determine the deal values from the SMC look-up tables. Theoretically, there is no limit on the number of Stage 2 paths except for the possible constraint concerning the time for aggregating the deals. Deals simple structures and complicated structures are treated in the same way. More importantly, the aggregation has no offset effect on the efficiency gained in the SMC Stage 1, in which SMC paths are customized

to line up with the risk profiles of individual deals.

In summary the procedure is:

1. Aggregate the values/risks of all deals (portfolio) for all MC paths.
2. Calculate the portfolio desired statistic, as in a conventional MC simulation.

SMC VAR convergence

SMC convergence to the exact infinite sampling result is guaranteed theoretically (whether for a deal or portfolio). This is because ordinary MC converges for an infinite number of paths, and this is just what SMC uses for the scout paths. The Brownian bridges in the limit as the bin sizes go to zero is equivalent to an infinite MC simulation, and is in fact the path integral. Mixing the two together is a convenience for faster calculations with a finite number of random numbers, but does not affect the convergence.

Appendix 3: Distance measure and interpolation

Denote the square of the distance from vector point $\mathbf{x}_b$ to vector point $\mathbf{x}^*$ as $d^2(x^*, x_b) = \| x^* - x_b \|^2$ for a certain number N_{nbs} of nearest SP, BB neighbors $\mathbf{x}_b$ with respect to $\mathbf{x}^*$. Based on analysis in the references, we choose $N_{nbs} = 5$ nearest neighbors as the default.

As the distance from x^* to one of the x_b gets small, the number of nearest neighbors needs to decrease eventually to 1, i.e. $N_{nbs} \to 1$, in order to reproduce the deal value at x_b. This is a requirement for the distance function.

We choose a nearest-neighbor cutoff "Cauchy" interpolation weight using the squared distance function $d^2(x^*, x_b)$. The Cauchy distance bandwidth cutoff parameter κ is used to control smoothness vs. variation[11].

$$w(x^*, x_b) = \frac{\left[d^2(x^*, x_b) + \kappa^2 \right]^{-1}}{\displaystyle\sum_{x_{b'} = N_{nbs} \text{ nearest nbs to x*}} \left[d^2(x^*, x_{b'}) + \kappa^2 \right]^{-1}} \tag{A3.1}$$

Note that for small $\dfrac{d^2(x^*, x_b)}{\kappa^2} < 1$ the form of $w(x^*, x_b)$ is approximately Gaussian since $\dfrac{1}{1+y^2} \approx e^{-y^2}$. For larger values of $\dfrac{d^2(x^*, x_b)}{\kappa^2}$ the Cauchy weight decreases more slowly than exponential and so more contribution comes from further away neighbors with the Cauchy weight than with the Gaussian weight.

Interpolated value of deal

The final interpolated value of the deal at x* called $C_{\text{interp}}(x^*)$ is then

$$C_{\text{interp}}(x^*) = \sum_{\mathbf{x}_b = N_{nbs} \text{ nearest nbs to x*}} C(x_b) w(x^*, x_b) \tag{A3.2}$$

As $x^* \to x_b$ we get the required result $C_{\text{interp}}(x^*) \to C(x_b)$ because we decrease the

---

[11] There is also a path-dependent Cauchy-form interpolator. See references.

number of nearest neighbors $N_{nbs} \rightarrow 1$ and so $w(x^*, x_b) \rightarrow 1$ as $x^* \rightarrow x_b$.

This can be done with a functional form

$$k^2 = k_0^2 f(x^*, \{x_{b'}\}) \tag{A3.3}$$

Here we define

$$f(x^*, \{x_{b'}\}) = \prod_{N_{nbs} \text{ nearest nbs to } x^*} \frac{d^2(x^*, x_b)}{1 + d^2(x^*, x_b)} \tag{A3.4}$$

With this definition, as the interpolated point goes to one of the reval points, the weight becomes one near that reval point and is zero for the other reval points in the sum in eq. 3.2.

We emphasize that the number of dimensions and the number of nearest neighbors are not the same. Actually, to formulate the criteria for nearest neighbors in more than one dimension is complicated and requires cluster analysis. In this paper, even though the interpolator is valid in multiple dimensions, we only use it in one dimension.

Interpolator improvements

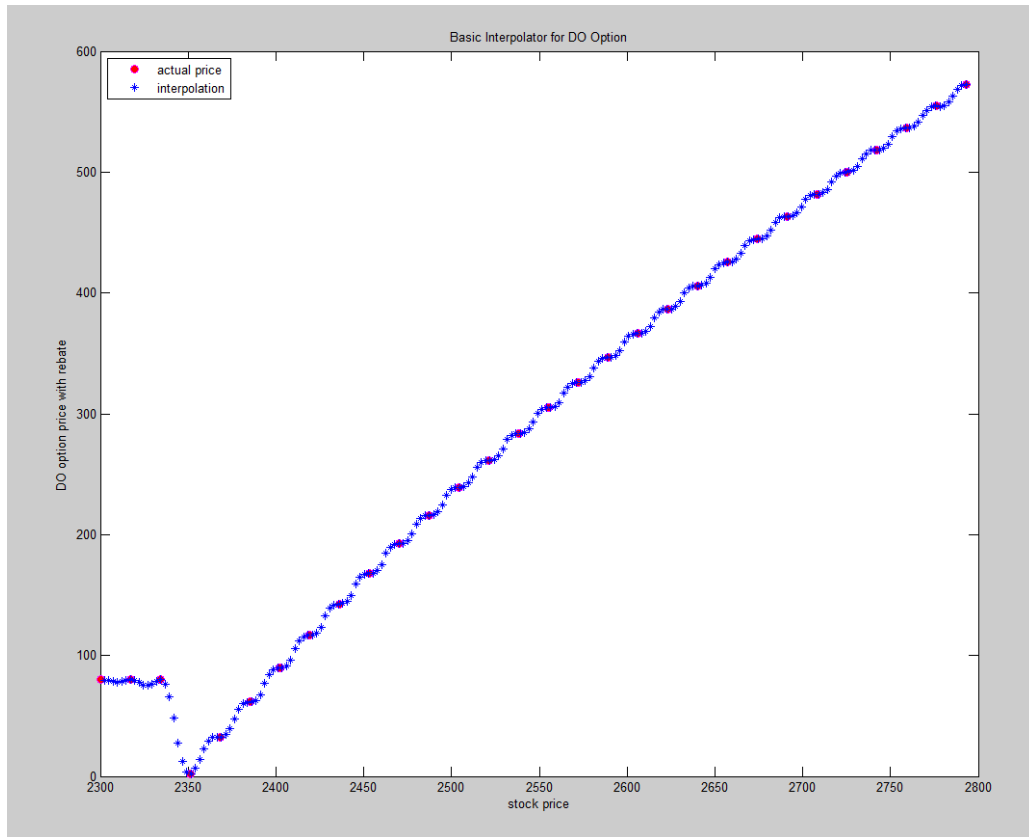## 1. Interpolator without improvements exhibiting oscillations

Figure A3.1: The down-out option with the basic interpolator exhibits oscillations. This will be improved below.

2. Techniques to suppress interpolator-caused oscillations

Here are the techniques we used to help suppress oscillations.

(1) Ensure at least one point is on either side

After finding the N nearest neighbors, we check whether they are on the same side. If so, add one point from the other side; if not, use these N points.

(2) Add first derivative term and second derivative term

To reduce oscillations, we add a "momentum" term. The momentum approximation is to place the first derivative "delta" term into the interpolation pricing formula. This is done without spoiling the agreement between the interpolation and the reval at a revaluation point.

The basic interpolator is

$$C_{interp}(x^*) = \sum_{N_{nbs} \, nearest \, nbs \, to \, x^*} C(x_b) w(x^*, x_b) \tag{A3.5}$$

After adding a first derivative delta term, the improved interpolator is

$$C_{interp}(x^*) = \sum_{N_{nbs} \, nearest \, nbs \, to \, x^*} (C(x_b) + \Delta_b C(x^* - x_b)) w(x^*, x_b) \tag{A3.6}$$

Here ((the index i refers to different dimensions in the variables)[12]

$$\Delta_b C(x^* - x_b) = \sum_{i=1, x_{b'}=NN(x_b)}^{D} \frac{C(x_b) - C(x_{b'})}{x_{bi} - x_{b'i}} (x_i^* - x_b) \tag{A3.7}$$

In practice, the first-derivative delta momentum term reduces most of the oscillation problem[13].

To further reduce oscillations, we add the second derivative "gamma" term into the interpolation pricing formula.

The second derivative term is (again the index i refers to different dimensions in the variables)[14]

$$\nabla_b^2 C(x^* - x_b)^2 = \sum_{i=1}^{N} \left\{ \left[\frac{C(x_{b'}) - C(x_b)}{x_{b'_i} - x_{b_i}} - \frac{C(x_{b''}) - C(x_b)}{x_{b''_i} - x_{b_i}} \right] * \frac{1}{x_{b'_i} - x_{b''_i}} * (x_i^* - x_{bi})^2 \right\} \tag{A3.8}$$

Therefore, the final interpolated price is

---

[12] The first derivative delta term is calculated using its nearest left side point.

[13] We use a threshold = 3 standard deviation criterion with respect to the standard deviation through time of these delta and gamma terms. If an individual delta (gamma) is larger than 3sd, we set it to zero in the interpolation formula to avoid jumps in the interpolation.

[14] We ignore cross-gamma terms between different dimensions if they exist.

$$C_{interp}(x^*) = \sum_{N_{nbs}\ nearest\ nbs\ to\ x^*}[C(x_b) + \Delta_b C(x^* - x_b) + \frac{1}{2}\nabla_b^2 C(x^* -$$
$$x_b)^2]\ w(x^*, x_b) \tag{A3.9}$$

Oscillations can also be caused by density fluctuations in the points where revaluation is done. For that reason, in practice some points may need to be removed.

(3) Do averaging for jumpy derivative terms

If successive values of delta jump more than 3 standard deviations of changes between deltas of nearest neighbors, the deltas on either side of the jump are replaced by the average of 5 nearest neighbor deltas. A similar replacement is made if there is a jump in gamma. This removes unwanted effects due to jumpy derivative terms. Consistency is maintained with the interpolated value being the reval value at a scout point.
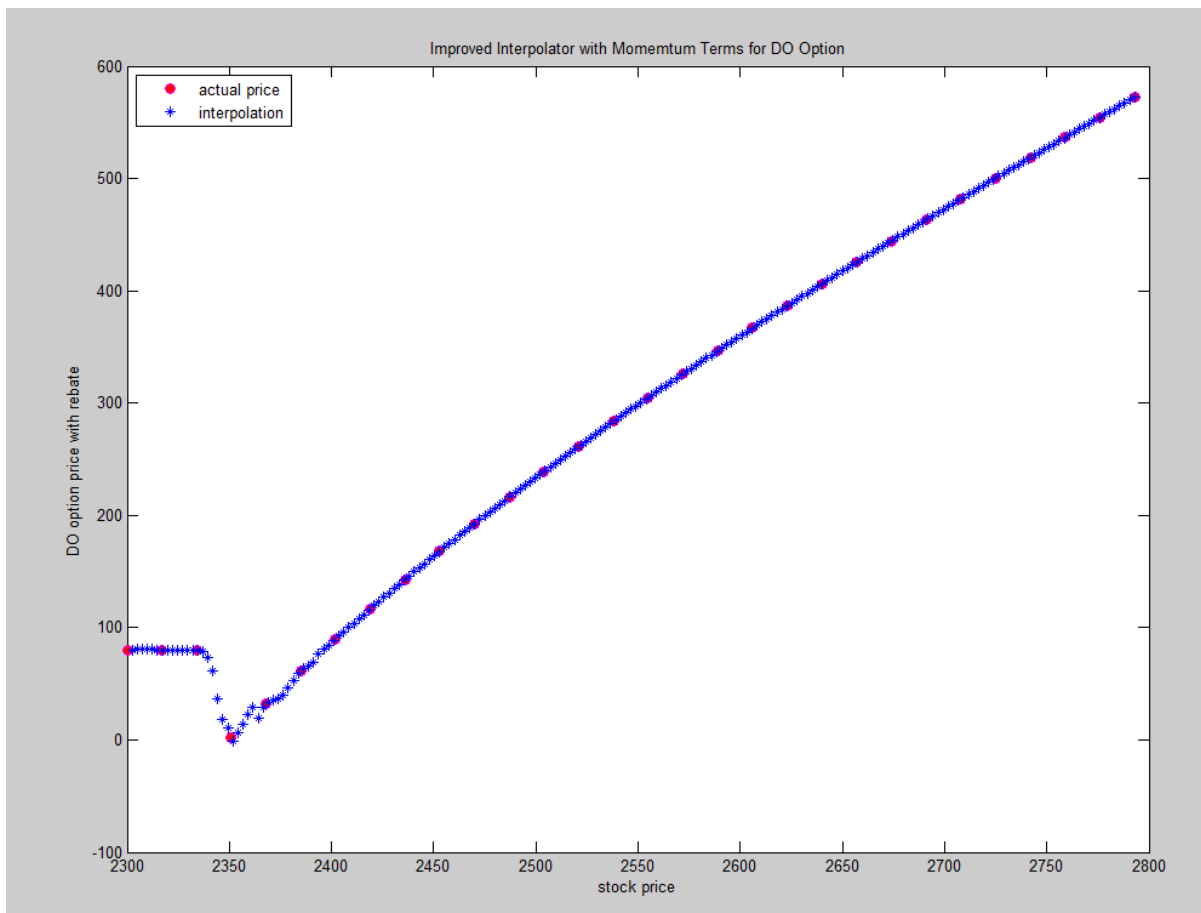

Figure A3.2: Result for down-out option with improvements described above.

3. Dealing with jumps

(1) Phase differentiation

For a Down & Out Option with rebate, two phases can be defined - the rebate phase and the option phase - which should have interpolations that are independent from each other. But in our intermediate interpolator above, information from both sides are mixed – when we do interpolation for points in rebate phase, we actually use information from the option phase. We need to prevent this situation. Therefore, we implement phase differentiation – when interpolating for points in rebate phase, se only use information in this phase; the same for option phase.

In general we need to define phases. Here we just use the option phase as starting at the first point where the value is not the rebate value. This is a generally difficult problem in pattern recognition. We can looks for jumps in the value of the security and/or in its derivatives.

If the deal terms are known in advance, we know the positions of the jumps (e.g. the position of a barrier below which the deal has a constant rebate value). If the deal terms are not known in advance, the problem becomes much harder and we have no general solution.

We leave a more general solution to future work.

(2) Removing nearest-neighbor jump caused by one scout point coming in and one dropping out using a linear combination

There is a problem as nearest neighbors change in the sum over nearest neighbors. Instead of a sudden change of scout points, we smoothly change scout points by assigning weights on the ending points of a window. At the beginning of the window, we have nearest neighbor #1, and at the end of the window we have nearest neighbor #6, which becomes nearest neighbor #5 when the previous nearest neighbor #1 drops out. For intermediate points in the window, we linearly interpolate the contributions between these two nearest neighbors. Thus we have effectively 5 nearest neighbors (part of #1, #2, #3, #4, #5, part of #6).

The final version of the down-out option with interpolation using all the modifications is shown in the graph below, which is the same as the figure in the introduction.
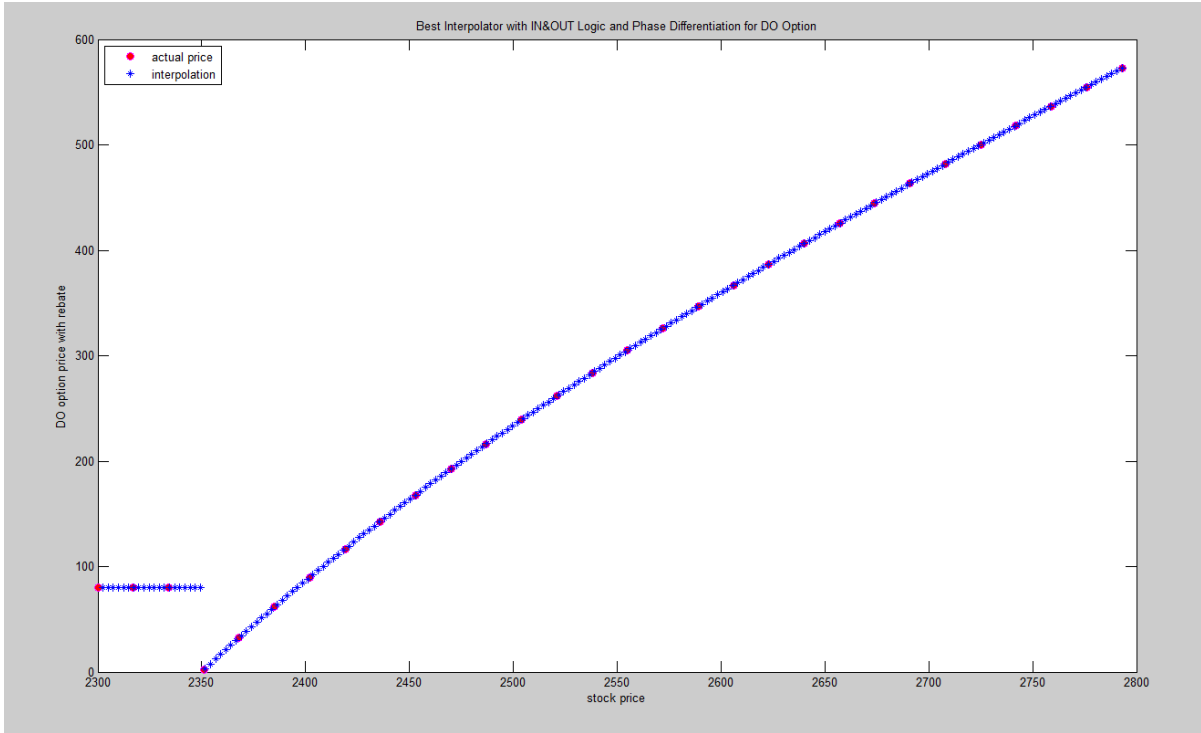
Figure A3.3: Result for down-out option with all refinements

Appendix 4: Path Integrals and Brownian Bridges

An introduction to path integrals applied to finance is in Ref. iii.

Here, the reader only has to know that a definite probability (consistent with the stochastic equation) is obtained for all paths that start from a given point at a given time, traverse all allowed points at all intermediate times, and wind up in a given "bin" region of the variable(s), at a later time. Such an (infinite) set of paths is called a "Brownian Bridge" or BB. For example for an OTM option, we want send BBs into the OTM region (with respect to today's price) to sample the option in a more detailed way than ordinary MC for the same number of calls to the pricer.

Figure 4.1 shows a picture of Brownian Bridges in time, with a specific path of BBs in dark font. Again, VAR only has one time step, as illustrated in Figure 4.2.
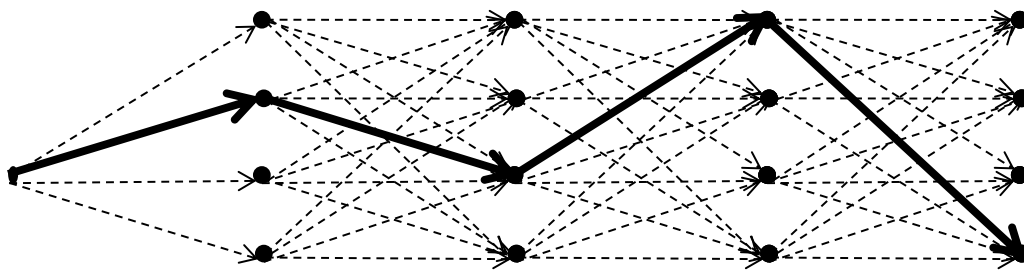
Figure 4.1: SMC illustration of BBs over multiple time steps, with one path of BBs in dark font. VAR uses only one time step
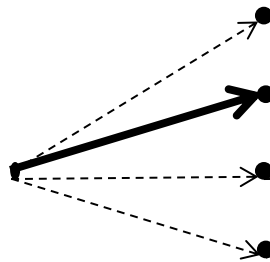
Figure 4. 2: SMC BBs for VAR – one time step

How do we know a-priori where to send the BBs, e.g. where the OTM region is? Again recall the first step that runs a limited, although not **too** limited, scenario and real-world MC simulation of scout paths SPs for which the pricing function is called. The SPs come back with enough information about "interesting" regions where substantial deal value exists. For step 2, we send BBs to those "interesting regions"[15,v] . More information is in Appendix 6 below.

---

[15] The BB construction in SMC is conceptually similar to some Metropolis-like filter that selects MC paths according to some criteria, with the criteria here depending on the financial instrument. Note that randomness still exists with a simple filter of MC paths; the method here is not a simple filter.

Brownian Bridge Generation Details

The material for this section is from Appendix 3 in the second paper of Ref. i to which we refer the reader. It is included for completeness.

General aspects - Where should we put Brownian Bridges?

The issue in this section concerns where to position the Brownian Bridges for complex situations, to formulate rules for reasonable, if not optimal, BB end positions. The effectiveness depends on the metric that quantifies the "closeness" between a BB and a SP. Appropriate closeness is needed to characterize the local risk profile in a profitable fashion. The algorithm may need to be customized to individual deal features that vary significantly.

The algorithm described below was used in the first paper of Ref. i for the caps portfolio, the Matlab logo and the down and out option.

The BB generation algorithm starts from the SPs with positive payoffs. SP payoffs are sorted beforehand and the process starts with the SP with the largest payoff.

The algorithm specifies that each SP produces 5 BBs in the direction towards larger cash flows within a range of 3 standard deviations of underlying variable volatility, away from the current SP.

We repeat the generation of BBs for SPs until the budget for the number of BBs calling the pricing function is reached. In case of an unfilled BB budget, new BBs are generated from existing BBs with positive payoffs following the algorithm in the paragraph above. BBs are forced to be different from the "parent" SP.

If no BB meets the increasing payoff criteria, shrink the search range to be $(3 - \delta)$ standard deviations and start the process over. This trial process stops until δ hits an upper bound.

In case of multi-dimensional risk factors, the BB generation process is carried out in the following manner:

- Generate a co-ordinate system
- On the sphere defined by e.g. 3 stdev, generate e.g. 5 points at random.
- Test to see if these points are reasonable for BB positions via some criteria, if not generate other points at random.

This is necessary since otherwise the budget would be rapidly exhausted[16].

---

[16] For two dimensions it may be possible to fill a small area. This was the procedure for the Matlab logo description above, except if a BB was not associated with a positive payoff it was not used.

Appendix 5: Deal-dependent Scenarios

The following metrics are suggested to discern specific types of deals.

- Yield-Curve spreads (information is given that rates r_2, r_{10} exist; test for spread dependence $s_{2,10} = r_{10} - r_2$). Use results of parallel shift and tilt to discern.

- Long-short index vs. basket (information is given that index s_{index} and stocks $\{s_\alpha\}$ exist). Use scenario for $\delta S_{index,basket} = s_{index} - \dfrac{1}{N}\sum_{\alpha=1}^{N} s_\alpha$ with equal weights to discern[17].

- Outperformance (information given that stocks $\{s_\alpha\}$ exist): Use scenario for $S_{\text{Out Performance}} = \sum_{\alpha < \beta} |s_\alpha - s_\beta|$ to discern[17].

---

[17] Stocks are normalized s(t)/s(0)

Appendix 6: Equity basket with knock-outs

In this appendix we present a conjecture that we believe will be useful in dealing with equity baskets with complicated knockouts. For example, we can have a deal that has zero value if any stock goes below a pre-defined level.

The idea is to use a simple approximate model to reduce the variance. We choose to look at the disjoint cases of no stock going below its level, and successively each stock (but not the others) going below its level. This misses the cases when two stocks etc. go below their levels. We further assume zero correlation between the stock going below its level and the sub-basket composed of the other stocks. However, full correlations can be used for the construction of the sub-basket. We then perform all permutations.

It turns out that a solution of this problem exists for constant parameters (volatilities etc). The solution for each stock knocking out is four image solutions. Each image solution is a 2 dimensional construction combining a one-dimensional knock-in Green function and a one-dimensional knock-out Green function (using terminology of Ref. iii). Integrals then produce the probabilities of the occurrence of each of these separate cases. With these probabilities we can get an approximate analytic solution.

We then look at the ratio between the analytic solution and the reval solution, at appropriate scout points. The ratios are then interpolated with the interpolation functions described in the text in order to obtain an approximate result.

We have not had time to carry out numerical analysis of this idea.

Specifically, the equations in the notation of ref iii (Ch. 17, 19) for the proposed simple approximate model are below for the simple case of a European option.

Let S_1 denote the first stock, with the remaining stocks 2…n treated as a basket or index (like SPX) with price called $\mathrm{Basket}_{2...n}$ treated as a single variable, for example, $\mathrm{Basket}_{2...n} = \dfrac{1}{n-1}\sum_{\alpha=2}^{n} S_\alpha$. It is assumed that S_1 crosses the barrier H_x sometime before expiration, but none of the other stocks cross their barriers, or equivalently that the basket does not cross an effective barrier $H_{\mathrm{Basket}(2...n)}$ that has to be calculated. We write

$$
\begin{aligned}
x_1 &= \ln(S_1) \\
y_1 &= \ln(\mathrm{Basket}_{2...n})
\end{aligned}
\qquad (A6.1)
$$

We assume lognormal dynamics for x_1, y_1 and further assume the correlation between x_1, y_1 returns is zero, i.e. $\rho_{x_1,y_1} = 0$.

We denote x_1^*, y_1^* as x_1, y_1 at expiration. The 2 dimensional Green function $G_1^{(2D\_tot)}$ is written as the difference

$$\mathcal{B}_1^{(2D\_tot)} = \mathcal{B}_1^{(2D)} - \tilde{\mathcal{B}}_1^{(2D)} \qquad \text{(A6.2)}$$

Here the model specifies $\mathcal{B}_1^{(2D)}$ and $\tilde{\mathcal{B}}_1^{(2D)}$ as (using notation of ref. iii, Ch 17, 19 to which we refer the reader)

$$\mathcal{B}_1^{(2D)} = \begin{cases} K(x_1)G^{(2D)}(x_1^{\text{image}}, y_1; x_1^*, y_1^*) \text{ if } S_1^* > H_x \text{ (case A)} \\ G^{(2D)}(x_1, y_1; x_1^*, y_1^*) \text{ if } S_1^* < H_x \text{ (case B)} \end{cases} \qquad \text{(A6.3)}$$

$$\tilde{\mathcal{B}}_1^{(2D)} = \begin{cases} K(x_1)K(y_1)G^{(2D)}(x_1^{\text{image}}, y_1^{\text{image}}; x_1^*, y_1^*) \text{ if } S_1^* > H_x \text{ (case A)} \\ K(y_1)G^{(2D)}(x_1, y_1^{\text{image}}; x_1^*, y_1^*) \text{ if } S_1^* < H_x \text{ (case B)} \end{cases} \qquad \text{(A6.4)}$$

We then obtain the probability of one stock crossing as the integral

$$\text{Prob}^{\text{Model}}(\text{one stock crossing}) = \int_{-\infty}^{\infty} dx_1^* \int_{\ln H_{Basket(2...n)}}^{\infty} dy_1^* \mathcal{B}_1^{(2D\_tot)} + \text{Permutations} \qquad \text{(A6.5)}$$

The probability of payoff in the model is then

$$\text{Prob}^{\text{Model}}(\text{payoff}) = 1 - \text{Prob}^{\text{Model}}(\text{one stock crossing}) \qquad \text{(A6.6)}$$

The value of an option is then approximated by

$$C^{\text{Model}}(\text{with KO}) \approx \text{Prob}^{\text{Model}}(\text{payoff}) * C(\text{no KO}) \qquad \text{(A6.7)}$$

The suggested procedure is then to interpolate the ratios of model values with exact revaluation, viz

$$\text{Ratio (Reval point)} = \frac{C^{\text{Model}}(\text{with KO})}{C^{\text{Exact}}(\text{with KO})}(\text{Reval point}) \qquad \text{(A6.8)}$$

Pictorially the solution for the first stock crossing can be viewed as contributions from four regions in the plane:

1st quadrant: x_1, y_1 and also x_1^*, y_1^*

2nd quadrant: x_1^{image}, y_1 and also x_1^*, y_1^*

3rd quadrant: $x_1^{\text{image}}, y_1^{\text{image}}$

4th quadrant: x_1, y_1^{image}

Appendix 7: SMC VAR calculated for a portfolio (using Greeks)

It turns out that there is a way of calculating VAR for a portfolio directly. For a portfolio, the space of underlying returns ("phase space") is high dimension and therefore very difficult to sample. This idea uses SMC with full reval applied starting in a region in phase space determined by an approximation. The starting region for SMC is obtained using a linear approximation (Greeks, exposures). The VAR for the linear approximation is determined along with the component VARs (CVARs) and the component VAR volatilities (CVAR vols) and the correlations between component VARS (CVAR correlations). See Ref. iii, Ch. 28 for details.

This procedure determines an ellipsoid in the multidimensional space of the underlyings. A neighborhood of this ellipsoid gives the various configurations that can exist for a fixed VAR[18]. This ellipsoid is the desired starting point in the huge phase space for SMC. Thus, this ellipsoid (including neighborhoods) is then probed by SMC using full pricing reval.

The maximum value of the portfolio loss obtained by full-reval SMC is then the candidate for the non-linear VAR for the portfolio, and the point of this maximum is saved.

The shortest distance (using all variables) between the historical differences and the above maximal point then determines the distinguished HVAR date, and the full reval of the loss in the portfolio at the HVAR date is the historical VAR.

Contributions of any sub-portfolio or deal at the HVAR date to the HVAR are known since these must all be evaluated to get the HVAR for the portfolio.

---

[18] A picture is on page 416 of ref. iii for the case of two variables.

Appendix 8: Approximate Analytic Evaluation of Multivariate Gaussian Integrals - Details

See ref.ii for the original paper. Here we only use the zero-th order term in the perturbative expansion in the index "beta"[19].

We need to determine the regions of interest and whether the probabilities are large enough to warrant additional pricing at each region of interest. For deals with multiple parameters, we need to determine the probability of prices moving into a hypercube, which requires the calculation of a multivariate integral:

$$I_N[\{x_i^{max}\}; \rho] = [\Delta(\rho)]^{-\frac{1}{2}} \int_{-\infty}^{x_1^{max}} \cdots \int_{-\infty}^{x_N^{max}} \left(\frac{1}{\sqrt{2\pi}}\right)^N \exp[-\frac{1}{2}x^T\rho^{-1}x] \prod_{i=1}^{N} dx_i \qquad (A8.1)$$

We first do correlation matrix factorization and replace correlation matrix ρ with a one-factor approximation matrix ρ_f.

Define following Dash[20]

$$c_i = \left[\frac{1}{N-1} abs(\sum_{l \neq i} \rho_{il})\right]^{\frac{1}{2}} sgn(\sum_{l \neq i} \rho_{il}) \qquad (A8.2)$$

Where the Curnow-Dunnett form is assumed to construct the approximation matrix

$$\rho_{fij} = \begin{cases} 1 & i = j \\ c_i c_j & i \neq j \end{cases}$$

We have in this approximation

$$I_N[\{x_i^{max}\}; \rho] \approx I_N[\{x_i^{max}\}; \rho_f]$$

$$= [\Delta(\rho_f)]^{-\frac{1}{2}} \int_{-\infty}^{x_1^{max}} \cdots \int_{-\infty}^{x_N^{max}} \left(\frac{1}{\sqrt{2\pi}}\right)^N \exp[-\frac{1}{2}x^T\rho_f^{-1}x] \prod_{i=1}^{N} dx_i \qquad (A8.3)$$

We then decrease the dimension from N to 1 by changing integral variables.

$$I_N\left[\{x_i^{max}\}; \rho_f\right] = \int_{-\infty}^{\infty} \frac{d\varsigma}{\sqrt{2\pi}} \exp(-\frac{1}{2}\varsigma^2) \prod_{i=1}^{N} R\left[\xi_i^{max}(\varsigma)\right] \qquad (A8.4)$$

where $\xi_i^{max}(\varsigma) = (x_i^{max} - c_i\varsigma)/s_i$ and $s_i = \sqrt{1-c_i^2}$.

---

[19] For those reading ref. ii, we need in addition to expand the determinants in the variable epsilon, in the notation of that paper. We also need to normalize the probabilities to one over all space by dividing by the integrals all taken from minus infinity to infinity.

[20] If all c_i have magnitude less than one, it can be shown that the matrix r_f is positive definite.

We rewrite the integral as

$$I_N\left[\left\{x_i^{max}\right\};\rho_f\right] = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(\Phi(\varsigma))d\varsigma \qquad \text{(A8.5)}$$

Here, $\Phi(\varsigma) = -\frac{1}{2}\varsigma^2 + \sum_{i=1}^{N} \ln\left[R\left(\frac{x_i^{max} - c_i\varsigma}{s_i}\right)\right]$.
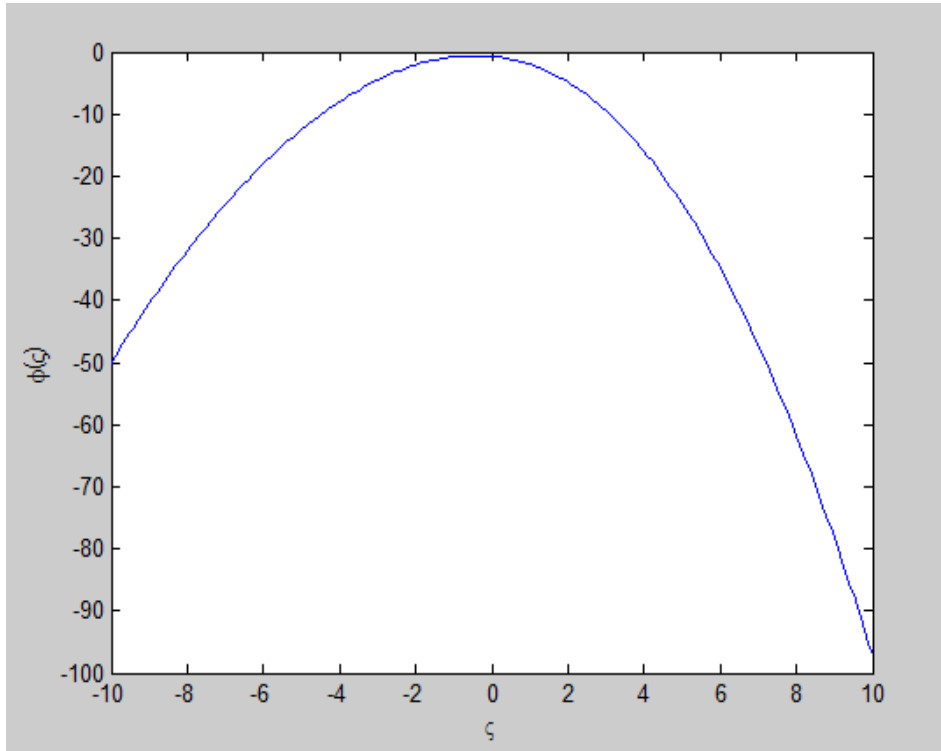
The plot of $\Phi(\varsigma)$ is shown below:

Figure A8.1

We use the saddle point approximation. Expand the function $\Phi(\varsigma)$ around its maximum value:

$$\Phi(\varsigma) = \Phi\left(\varsigma^*\right) + \frac{1}{2}\Phi''\left(\varsigma^*\right)\left(\varsigma - \varsigma^*\right)^2 \quad \text{(A8.6)}$$

Plugging in the cumulative normal functions, we obtain an expression for the second derivative $\Phi''\left(\varsigma^*\right)$.

The final approximation is done without any integral since the cumulative normal has an analytic approximation, cf. [vi]. We get

$$I_N\left[\left\{x_i^{max}\right\};\rho_f\right] = \exp\left(\Phi\left(\varsigma^*\right)\right)\cdot\left(-\Phi''\left(\varsigma^*\right)\right)^{-\frac{1}{2}} \qquad \text{(A8.7)}$$

References

[i] Dash, J. W. and Yang, X., *PATH INTEGRALS AND SMART MONTE CARLO - I* (SMC I), Bloomberg L.P. working paper 2016, http://papers.ssrn.com/abstract=2808177 ; Dash, J. W. and Yang, X., *PATH INTEGRALS AND SMART MONTE CARLO – II* (SMC II), Bloomberg L.P. working paper 2016, http://papers.ssrn.com/abstract=2808179 ; Dash, J. W. and Yang, X., *SMART MONTE CARLO, PATH INTEGRALS, AND AMERICAN OPTIONS* (SMC III), Bloomberg L.P. working paper 2016, http://papers.ssrn.com/abstract=2808186

[ii] Dash, J. W., *Multivariate Integral Perturbation Techniques. I: Theory,* International Journal of Theoretical and Applied Finance, Vol. 10, No. 8 (2007) 1287-1304

[iii] Dash, J. W. *Quantitative Finance and Risk Management, A Physicist's Approach*, 2nd Edition (World Scientific, 2016). ISBN 978-981-4571-23-4. See chapters 41-45 and chapters 17,19.

[iv] M. Bondioli, private communication

[v] Atchade', Y. F. and Rosenthal, J. S., *On adaptive Markov chain Monte Carlo algorithms*, Bernoulli 11 815-828 (2005).

[vi] Abramowitz, M., Stegun, I., *Handbook of Mathematical Functions.* National Bureau of Standards, Applied Mathematics Series #55, 1964. See Sect. 26.2.17

File = SMC_VAR_Speedup_v6.docx; 2/22/2018 12:00:00 PM