SHIPLEY CENTER FOR INNOVATION
CLARKSON UNIVERSITY

Uncertainty Arithmetics Applied to the Black-Scholes Model

Applications of CertainError Geometric Arithmetic

Ronald LaFleur – rlafleur@clarkson.edu

8/17/2015

EXECUTIVE SUMMARY

Background: Uncertainty in Financial Systems

Numbers play a central role in domestic and world financial systems. Computers are used to track accounts and transfers with numbers. However, in many cases there are uncertainties about the values used because the source of information may be imperfect or fluctuate with time. Knowing how to control errors and uncertainty is an advantage worth pursuing.

This report demonstrates the superiority of Duals Arithmetic on the Black-Scholes Call Option model making further financial applications feasible. In short, this application enhances financial decision making. Implementation of Duals Arithmetic is recommended to increase the confidence level of financial decisions and minimize the need for uncertainty buffers.

Better Uncertainty Calculation: Duals Arithmetic

A new idea is to format *all* numbers to have two (dual) parts: a value and a multi-dimensional error vector. The new basis for calculating with these dual numbers, Duals Arithmetic, has been developed by Professor Ronald LaFleur of Clarkson University and *CertainError*. This report demonstrates the effectiveness of this new method compared to four other uncertainty calculation methods: Intervals, Monte-Carlo, Differentials and Chordals and two baseline-validation methods that do not calculate uncertainty: Traditional Arithmetic and Exact Arithmetic.

A pilot example is the Black-Scholes model of Call Value vs. Time-to-Expiry (0 to 2 years) and Spot Price ($80 to $120). Three parameters are the Strike Price ($100), Risk-Free Rate (4%/year) and Volatility (2 √%/√year). Uncertainty is applied as 5% relative error (the balance of 95% confidence) on each of the five inputs and these propagate to the Call Value and Error of Call Value.

Results

Key performances reported are informational content, computation time cost, memory requirements, and where some methods fail. Duals Arithmetic possesses the following advantages:

1. **Effective** – Duals Arithmetic provides six times the information of Traditional Arithmetic requiring only 23% longer computational time. Duals Arithmetic provides three times the information of the Monte-Carlo Arithmetic in at least 41% shorter computational time. *Duals Arithmetic is better and faster*
2. **Robust** – Duals Arithmetic tolerates operations that crash the other methods. The divide-by-zero and square-root-of-negative problems occur near zero Time-to-Expiry and when the Spot Price is near the Strike Price. *Duals Arithmetic doesn't fail*
3. **Better uncertainty information** – Duals Arithmetic has the lowest Error of Call Value, solves the dependency problem and creates an advantageous error-manipulation surface. *Gain knowledge*
4. **Strategy** – Duals Arithmetic automatically provides a multi-dimensional compass that directs tailoring of the error budget. *Duals Arithmetic increases your advantage*

Table of Contents

1. INTRODUCTION

A well-known formula for calculating the value of a call option is the Black-Scholes model [1]. This model expresses the Call Value as the difference between the value of the option-to-buy and the value of the underlying stock. In the original notation (Equation 13 from [1])

$$w(x,t) = xN(d_1) - ce^{r(t-t^*)}N(d_2)$$
$$d_1 = \frac{\ln(x/c) + (r + \frac{1}{2}v^2)(t^* - t)}{v\sqrt{t^* - t}}$$
$$d_2 = \frac{\ln(x/c) + (r - \frac{1}{2}v^2)(t^* - t)}{v\sqrt{t^* - t}}$$

Here, x is the stock price, t is time, c is the strike price, t^* is the maturity time, r is the risk free interest rate, v is the volatility, and N(d) is the cumulative normal distribution.

Since the log function is used in the 'd' expressions, the stock price follows a log-normal distribution. There are alternative, but equivalent ways to write the Black-Scholes model to either enhance understanding or to provide easier computational implementation. Black&Scholes (1973) [1] cite Sprenkle (1961) who published a similar formula earlier. Therefore the main novelty of Black&Scholes [1] was to provide a theoretical basis for persistent unknowns in earlier valuation formulas.

The Black-Scholes model sets a range of variability using dynamic-like parameters such as Time-to-Expiry, Risk-free Rate and the Volatility of the stock. While the stock is thought to fluctuate randomly, it is expected to follow a log-normal distribution which means the potential change in price over a small time interval depends on the current price. All of this is contained in the model yet there is always some doubt about its validity. While it may be clear at the expiry date to exercise the option (or not), this definite decision is offset by the indefinite information considered at the earlier time when the option is written.

A source of doubt or uncertainty is from the fact that the model does not capture everything that could occur once the clock starts ticking. For example, the Black-Scholes model relies on a set of assumptions about the underlying stock's behavior

1. Risk-free rate – this is assumed constant over the life of the option
2. No dividend – this adds to the value of the stock but is not used to make option decisions
3. Random stock fluctuations follow a log-normal distribution or, equivalently, the log follows a normal distribution. This ignores large changes due to rogue events.

This is supplemented by assumptions about environmental mechanics such as freedom to buy and sell any amount of the stock, the option can only be exercised at maturity (European) and no-transaction fees or tax implications are considered when calculating the option value. In any practical situation, it is not certain that the burden of each assumption is met. But does this destroy the information yielded?

This report presents a study of different uncertainty arithmetics as they are applied to the same Black-Scholes calculation of Call Value. This includes popular methods such as Interval Arithmetic, Monte-Carlo Arithmetic and Differential Arithmetic. Two new methods, based on geometric arithmetic, Chordal Arithmetic (CertainError class 1) and Duals Arithmetic (CertainError class 2) are proposed to replace the popular methods and this forms a contest to determine which method is best. These two methods are Patent Pending with future licensing from *Clarkson University* and applications development by *CertainError*.

Each type of uncertainty arithmetic used calculates the same Black-Scholes cases and its performance is judged according to uncertainty information provided, memory requirements, computational speed and robustness. To validate each uncertainty arithmetic as computer code, two benchmark methods, Traditional Arithmetic and Exact Arithmetic, are included. Neither of these provide uncertainty information but give a baseline performance for memory requirements and computational speed.

The second section poses the Black-Scholes model as a procedure of unary and binary operation. This is done to provide common steps for calculating and allow easier conversion of the traditional operations to uncertainty arithmetics. Section three then proposes the versions of number formats and arithmetic that define each method in the contest. A novel approach is to interpret these methods not as just numerical in nature but also geometrical.

Sections four through six provide the resulting performance measures of each uncertainty arithmetic. Section seven provides the details of the Black-Scholes surfaces of Call Value and Error of Call Value to assess the robustness of each calculation and to illuminate where results are similar to the past or novel. Due to the added information from the Duals Arithmetic, section eight is devoted solely to detail these results.

The conclusion is that Duals Arithmetic is superior in many respects and should be implemented as a replacement for older uncertainty arithmetics or adopted for new applications.

2. CALCULATION PROCEDURE

Modelling attempts to represent the stock and financing with a mathematical formula or set of instructions (a program) such that 'what-if' scenarios can be studied prior to committing to an actual option. Calculations and numbers form a definite basis for decision making yet some degree of uncertainty must be considered. The computational implementation relies on the following:

1. Defining common terms that can be calculated once and used later
2. Converting the formula to a series of binary operations

Using these ideas, the following top-to-bottom procedure for a Call Value is used, as five specifications (in bold) and sixteen calculation steps that each have intermediate results leading to one final result.

1. **Specify Time-to-Expiry** **τ**
2. Calculate square root of Time-to-Expiry $\theta=\sqrt{\tau}$
3. **Specify Risk Free Rate** **r**
4. Calculate rate for Time-to-Expiry $\rho=r\times\tau$
5. Calculate discount factor $D=\exp(-\rho)$
6. **Specify Strike Price** **K**
7. Calculate discounted strike price $P2=D\times K$
8. **Specify Spot Price** **P1**
9. Ratio Spot Price to Strike Price $G=P1\div P2$
10. Calculate natural logarithm of the ratio $H=\ln(G)$
11. **Specify Volatility** **σ**
12. Calculate volatility for Time-to-Expiry $\eta=\sigma\times\theta$
13. Calculate drift effect $a=H\div\eta$
14. Calculate scatter effect $b=\eta\div 2$
15. Calculate high $d1=a+b$
16. Calculate low $d2=a-b$
17. Calculate probability of high $N1=N(d1)$
18. Calculate probability of low $N2=N(d2)$
19. Calculate high return $C1=N1\times P1$
20. Calculate low cost $C2=N2\times P2$
21. Calculate value of option $C=C1-C2$

A majority of these calculation steps are basic arithmetic (addition, subtraction, multiplication and division). These are binary operations (two inputs). The unary functions (exponential, square root, natural logarithm, cumulative normal distribution) can be implemented as series of basic arithmetic operations, using built-in functions or novel approaches such as *trans-imaginary numbers* in the Duals Arithmetic [2].

The above calculation procedure works using single numbers and Traditional Arithmetic. It is not difficult to imagine that a unique Call Value, C, is calculated from a single case of five inputs.

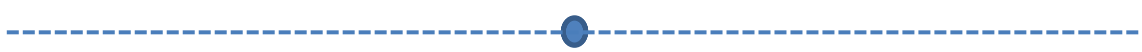
3. INFORMATIONAL CONTENT – alternate numbers and arithmetic

A deviation from an assumed-constant value for one of five specified inputs will change the resulting Call Value, C. It is also possible that changing two inputs could either offset each other, keeping C constant, or compound to change C dramatically. How these changes in the inputs affect the single Call Value illustrates how information propagates from input to output in the Black-Scholes model.

Instead of applying a change to one input as a set of repeated calculations of C (using the 21 step procedure over-and-over again), another approach is to represent each input as a domain of values. This represents each input as two numbers, one being the baseline value (the center) and the other being the amount of potential variation in either direction (+/-) away from the baseline (the error). This idea formats *every input* as a 'dual number' where the center and error are independently specified. The intermediate numbers and final answer, C, are also dual numbers.

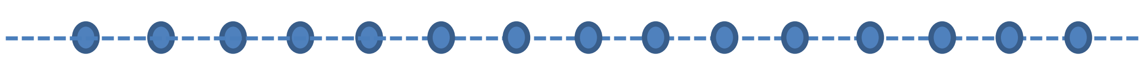
Every step of arithmetic in the 21 step procedure must be changed to respect both the center and the error [3]. A 'method' is defined as three items: the numbers represented in the data type, the arithmetic used to calculate with these numbers and the rendering of the answers for display. The data type also formats geometric information as shown by the accompanying sketches. There are a variety of approaches to uncertainty methods as follows:

1. **Traditional** – Uncertainty is not calculated because each number remains a single number. The idea of error is not addressed. Traditional Arithmetic of addition, subtraction, multiplication and division is used along with built-in functions such as natural logarithm and exponential.



2. **Exact** – The dual number is formatted for each input, as a center and error. New arithmetic is also formatted as one of the methods listed below. However, as a validation, the input errors are all zeroed such that the inputs are exact. The label 'exact' is synonymous with 'zero error.' Note that the error has to be formatted, and this is not the same thing as Traditional Arithmetic, yet the answers have to be the same to demonstrate validity.

3. **Intervals** [4,5] – The error domain about the center is populated with a uniform grid of points and calculations are performed by evaluating all combinations of point calculations and then finding the maximum result and the minimum result. The mid-point between the maximum and minimum is the result center and the result error is +/- distance between center and the maximum and minimum. The number of grid points chosen for each input is important. For example if each input error is grided with 100 points, a calculation with two inputs evaluates 10,000 points and calculation with three inputs evaluates 1,000,000 points.



4. **Monte-Carlo** [6] – The error domain about a center is independently populated with a randomly spaced sample of points. Then each input is a data set of the same size and calculations are done point-by-point. The *result data set* is of the same size as the inputs. The result center is the mean value of the result data set. The result error is the standard deviation of the data set scaled to a confidence interval using the cumulative student-t distribution (for example 95%) as a coverage factor. The size of the random sample for each input is important.

5. **Differential** [7] – The center is evaluated first from the error-free inputs (only centers). Knowing the arithmetic operation, a partial differential is formulated for each input and evaluated using the centers. Each of these centered-partial differentials is multiplied by its corresponding error input to get an error contribution. The independent error contributions are combined using a Pythagorean sum, that is the contributions are squared, the squares are added over inputs and the square root of the total is the resulting error (root-sum-squared, RSS). Basic arithmetic steps can be completed easily and form blocks for larger algorithms but this can introduce something called 'the dependency problem.' This problem can be reduced by substitution but the resulting larger formula can be difficult to treat with partial differentiation.

6. **Chordal** [8] – An upper point for each input is obtained by adding the error to each center. Similarly, a lower point is obtained by subtracting the error from each center. The input to each calculation step is coordinated to how it determines the upper or lower points for the result. This shadows the sensitivity of the output to each input. For example, if subtraction is used, C=C1-C2, the upper point of the result C is determined by the upper point of C1 and the lower point of C2. This typically gives the worst case error as the calculation is based on the bounds of error domains and the arithmetic is performed on a 'flat' geometry of scaled points. Implementation is made difficult when inputs appear more than once in a calculation step and sensitivity of output to inputs is cloudy.

7. **Duals** [9] – The error for each input is formatted as a vector with dimension equal to the number of inputs. For example, the Black-Scholes model has five inputs, therefore each input has a five-dimensional (5D) error vector. Along with the center, each input of the Black-Scholes model is formatted as six numbers. Therefore, while the Traditional Arithmetic has five numbers input and any one of the five methods described above (Exact, Interval, Monte-Carlo, Differential, Chordal) have ten numbers input (center and error for each input), the duals-

arithmetic has thirty numbers input (five inputs with six numbers each). This number format is used consistently for every operation and through the entire calculation procedure. At the end, the Call Value has all five components of the error vector. When reporting is needed, either all components are reported or a rendering process flattens this to an equivalent dual number that has one center and one error.

In summary there are seven methods being considered as shown in Table 1. These are distinguished by their numerical and geometrical content.

Table 1 – Uncertainty Methods for the Black-Scholes Call Option Model

Method	Numbers for input	Numbers Formatted	Geometry	Practical Scope
Traditional Arithmetic	1	5	NA	Widely used, certain
Exact Arithmetic	2	10	variety	Validation
Interval Arithmetic	2	10	Uniform grid of points	Not dominant
MonteCarlo Arithmetic	2	10	Random grid of points	Dominant for difficult
Differential Arithmetic	2	10	Local linearity	Dominant for simple
Chordal Arithmetic	2	10	Points	CertainError class 1
Duals Arithmetic	6	30	Points & error vectors	CertainError class 2

4. MEMORY REQUIREMENTS

While the Black-Scholes model is a good small example, larger applications require we know how the memory requirements scale-up. The required memory for calculations can be categorized into what supports the number data type and the overhead that is necessary to support the arithmetic. The Black-Scholes calculation is performed on a looped 2D grid from a Time-to-Expiry axis and a Spot Price axis. This means the required memory has three parts: overhead, out-of-loop numbers and within loop numbers. The memory required increases with the density of the grid points on each axis as this determines the number of loop steps as shown in Table 2. The general program is similar among all methods and provides memory management that frees memory when localized sub-calculations are completed. This correctly assesses the memory requirements of each method.

Table 2 – Memory Use for Uncertainty Arithmetics (in bytes)

n, Number of Grid Points on Each Axis	1	11	21	31	41	51
Traditional Arithmetic	264	1464	4264	8664	14664	22264
Exact Arithmetic	680	2760	8040	16520	28200	43080
Interval Arithmetic	680	2760	8040	16520	28200	43080
MonteCarlo Arithmetic	680	2760	8040	16520	28200	43080
Differential Arithmetic	680	2760	8040	16520	28200	43080
Chordal Arithmetic	680	2760	8040	16520	28200	43080
Duals Arithmetic (5D error vectors)	1712	8592	26672	55952	96432	148112

Traditional Arithmetic requires a number format of just one field as it provides no uncertainty information. Adding the error field to the number format doubles the memory requirement for the five inputs and any numbers calculated from them. The Exact Arithmetic formats the error field but then the number is set to zero in the memory to validate the general program as any Exact results should match with the Traditional results. The Duals Arithmetic for the Black-Scholes model requires that each variable have a format of six fields.

The memory requirements are scalable according to formula that fits the table exactly (R^2=1). There are three formula, one for each group of Table 2

$$\begin{array}{cc} Traditional & M = 8n^2 + 24n + 232 \\ Dual\ Numbers & M = 16n^2 + 16n + 648 \\ 5D\ Duals & M = 56n^2 + 16n + 1640 \end{array}$$

The constant number on the far right is the overhead memory that stays the same regardless of loop size. The central term, proportional to grid density, is for out-of-loop number formatting. Some numbers that feed into and out of a loop are indexed and sized proportionally to the number of loop steps. The quadratic term is for in-loop number formatting. The term is quadratic as there is one nested loop and both loops run n steps. This term is dominant and is often the only consideration for memory demand. In that regard, we see that general uncertainty arithmetic requires twice as much memory as Traditional Arithmetic. This is expected as each number was doubled in memory to simultaneously track error.

The Duals Arithmetic requires seven times as much memory as Traditional Arithmetic. This is more than expected for duals that retain just six number fields. *Why would this be?* The reason is that, like the other methods, the nested loops store a rendered dual number (two fields as a center and error). But unlike the other methods, the nested loops for duals arithmetic also store the five independent error vector components. The 5D error vector carries all the information that is needed and the magnitude is redundant information. Therefore the in-loop memory for the Duals Arithmetic has a coefficient of 56 instead of 48 so that the magnitude of the error vector is immediately available.

These formulae determine required memory when scaling-up the uncertainty arithmetic. For example, the American style and other option style operate on similar inputs with some additions but require more frequent calculations to update the numerical basis for decisions. For any type of option that involves calculations with numbers, errors propagate to uncertainty information. An outstanding question is how the memory resource, as a capital cost for calculating, provides beneficial information. The answer is especially important for Duals Arithmetic that provides much more information.

One benefit is that the error is an indicator of the effectiveness of an algorithm. This means, with constant input centers and errors, alternative algorithms can be judged according to the reported error components. The 21 step procedure used to calculate the Call Value could be changed to another procedure with less number of steps and larger formula on some steps (substitution). If the reported Call Value changes, then the uncertainty arithmetic being used has a ‘dependency problem.’

The dependency problem is due to branching in the algorithm (the 21 step procedure has 6 branch points) and the use of arithmetic steps that render the output as a dual number of one center and one error. An alternative algorithm can be defined to remove a branch but this usually results in larger formula with more inputs and more complicated form. This becomes a challenge for the Differential Arithmetic that requires formula for sensitivity derivatives (differential calculus) derived prior to programming error propagation. The Duals Arithmetic does not have the ‘dependency problem’ because errors are calculated algebraically (no calculus!) in five-dimensions and this is carried through every calculation step to every output.

5. COMPUTATIONAL TIME COST

Every program uses computer time. In the world of finance it is best to have information before your competition and this advantage motivates faster computer software and hardware technology. This is especially true of international finance where timing and transfer occurs between decentralized systems.

An easy way to compare methods is to have a general program that has constant format and simply changes the number format and arithmetic that is used. All cases would be run on the same computer hardware to even the playing field and lead to meaningful comparisons.

For example, a line of computer code that subtracts two numbers could have the following text-based options, respecting three number formats (traditional, dual number and duals) and the chosen arithmetic. The result is stored in the last of the four fields listed while the operation is specified by the first field and the order of the second and third field respects non-commutative operations such as a subtraction or a division

1. TraditionalArithmetic [‘subtract’,C1,C2,C]
2. ExactArithmetic [‘subtract’,C1Dual,C2Dual,CDual]
3. IntervalArithmetic [‘subtract’,C1Dual,C2IDual,CDual]
4. MonteCarloArithmetic [‘subtract’,C1Dual,C2Dual,CDual]
5. DifferentialArithmetic [‘subtract’,C1Dual,C2Dual,CDual]
6. ChordalArithmetic [‘subtract’,C1Dual,C2Dual,CDual]
7. DualsArithmetic [‘subtract’,C1Duals,C2Duals,CDuals]

It is easy to see the commonality of code format and that the distinct parts are easily changed by editing. This is a shell code and the actual arithmetic is specialized within the shell.

Due to the variability of computer clock rates, like a numerical experiment, the different arithmetic versions were run 100 times and the maximum, minimum, mean and standard deviation of run times were calculated. A single run performs the Black-Scholes calculation on a 21x21 grid (441 cases) of Time-to-Expiry and Spot Price. Table 3 shows the results of the run time tests with 3 significant figures.

Run times for the Interval Arithmetic depend on the number of grid points used to populate the error range of each number. The fastest time for interval arithmetic is the coarsest grid of just 3 points and this was used for runtime comparisons.

Run times for the Monte-Carlo Arithmetic should depend on the number of random points used to populate the error range of each number. The runtimes did not vary much between small cases of 2, 10 and 100 random points. The fastest case of 2 random points was used for runtime comparisons.

Table 3 Comparison of Run times for a Variety of Uncertainty Arithmetic

RunTimes of Uncertainty Arithmetic *100 runs of a 21x21 Black-Scholes Surface*	MAX [sec]	MIN [sec]	MEAN [sec]	STDev [sec]
Traditional Arithmetic (only centers)	5.22	4.38	4.47	0.142
Exact Arithmetic (zero errors)	5.54	4.89	4.99	0.125
Interval Arithmetic (3 points per axis)	6.93	6.17	6.28	0.154
MonteCarlo Arithmetic (2 instances per sample)	10.8	9.98	10.2	0.197
Differential Arithmetic	5.95	4.89	4.98	0.158
Chordal Arithmetic	5.98	5.08	5.17	0.154
Duals Arithmetic (5D error vectors)	6.42	5.72	5.82	0.149

One observation is that the mean runtimes are much closer to the minimum runtimes suggesting that the maximum runtimes are outliers or there is a skew in the distribution of runtime results. However, in any one calculation it is not known which runtime will result and the maximum run time represents a conservative result for scaling-up uncertainty arithmetic for larger applications.

The standard deviation shows how much the runtimes varied over the 100 run experiment. Among cases that report uncertainty, the worst variation is the Monte-Carlo Arithmetic and the lowest variation is the Duals Arithmetic. This is an indirect measure of consistency and utilization.

The Traditional Arithmetic is the fastest as it is not burdened with computations of error numbers. This can be used as a baseline to assess time-cost of calculating uncertainty. The second fastest is the Exact Arithmetic that has error formatted and calculated but it is consistently held to zero error throughout. Any of the other methods must match the Exact Arithmetic when the input errors are set to zero.

The Differential and Chordal Arithmetic take about the same amount of time (within one standard deviation) and are third fastest. The next fastest is the duals-arithmetic. Considering the memory requirements, the Duals Arithmetic was expected to be slower as more numbers have to be crunched. However, the error vector processing is accommodated without loops. Loops increase the effective number of executed lines of code.

The slowest methods are those using Interval and Monte-Carlo Arithmetic. Comparisons were made by making these two methods the fastest they could be. The lowest grid density on the Interval and the lowest number of random instances in a sample were chosen. Generally these choices are not practical. However, increasing the number of random instances in the Monte-Carlo comes with a low increase in runtime. But it is, by far the slowest method with the largest variation in runtimes. The Interval arithmetic cannot be helped as better error representation requires a higher density grid to populate the error range of each number and this slows the computation greatly.

6. BENEFIT-TO-COST RATIO

Considering runtime as a cost-of-computing and cost for the amount of time waiting-for-answers, the benefit gained is the information the calculation provides. The Traditional Arithmetic provides a unit of 1 piece of information. Extending to dual numbers, the Interval, Monte-Carlo, Differential and Chordal Arithmetics each provide 2 pieces of information. Finally, the Duals Arithmetic, provides 6 pieces of information.

An ad-hoc measure of Benefit-to-Cost (BCR) and computational effectiveness is the ratio of information to runtimes. The runtimes are first benchmarked to the Traditional Arithmetic runtimes. These are relative runtimes as costs shown in Table 4.

Table 4 Comparison of Benefit-to-Cost Ratio for Uncertainty Arithmetics

Benefit-to-Cost Ratio (BCR) *from Table 2 Max Run time*	Relative Run time	Information	BCR
Traditional Arithmetic (only centers)	1.00	1	1.0
Exact Arithmetic (zero errors)	1.06	1	0.9
Interval Arithmetic (3 points per axis)	1.33	2	1.5
MonteCarlo Arithmetic (2 instances per sample)	2.07	2	1.0
Differential Arithmetic	1.14	2	1.8
Chordal Arithmetic	1.15	2	1.7
Duals Arithmetic (5D error vectors)	1.23	6	4.9

One run of a Traditional Arithmetic calculation provides 1 piece of information at the cost of 1 unit of relative run time. If uncertainty was calculated by repeated use of the Traditional Arithmetic with variation of the inputs, each time the calculation is performed, 1 additional piece of information is gained at the cost of 1 additional time unit. In this scenario, the BCR would remain constant as 1 and further scaling up of the code by reuse will not change the BCR. There is no change in computational effectiveness.

However, if we can develop arithmetic that provides more information in the same amount of time, or provide the same information in less time, then the BCR is greater than 1 and this is a measure of goodness. In method shown in Table 4, the BCR is a combination of the cost of runtime and the benefit of information, both being different than the Traditional Arithmetic.

The BCR for the Exact Arithmetic is less than 1 showing that it has lower performance than the Traditional Arithmetic. The Exact Arithmetic does not provide uncertainty information as input errors are set to zero but since it is formatted, it is carried in the calculations. This makes the run time slightly higher than Traditional Arithmetic but with no benefit. The Exact Arithmetic is not practical and is only used for validation studies.

The BCR=1 of the Monte-Carlo Arithmetic verifies that it is equivalent to using Traditional Arithmetic over and over again. The Monte-Carlo method is well-known and simple to use as a computer code of Traditional Arithmetic. The code can be kept fixed inside a 'black-box' whereby inputs are varied randomly and outputs are processed using statistics. This 'black-box' approach is simple to implement but does not offer any advantages on computational effectiveness.

The most notable case is the Duals Arithmetic. At a cost of about 23% on runtime, the method provides a five-fold gain in information compared to Traditional Arithmetic. This fundamentally shows that the resulting gain in performance (BCR) is due to the sophistication of the mathematics used in the code, results that cannot be obtained by scaling or reusing any of the 'lower' arithmetic. *CertainError* has developed tools specific to implementation of the Duals Arithmetic to address this software challenge.

Since the BCR improvement from the Duals Arithmetic is primarily from the gain in information, a decision has to be made on how to use this information and take practical advantage of the benefit.

7. RESULTS FOR VARIOUS UNCERTAINTY ARITHMETICS

To judge the added information that dual numbers and duals provide over traditional numbers, we need to look at sample results from the Black-Scholes model. A set of representative inputs was used to investigate the different uncertainty arithmetic without being burdened by all possible cases. Table 5 gives the chosen sample values with prices around $100 and moderate rates.

Table 5 – Example Input Parameters for Calculating with the Black-Scholes Model

Inputs	Symbol	Units	Grid Number	High Center	Low Center	Relative Error	High Error
Time-to-Expiry	τ	years	21	2	0	5%	0.1
Spot Price	S	dollars	21	120	80	5%	6
Risk Free Rate	r	percent/year	1	4	4	5%	0.2
Strike Price	K	dollars	1	100	100	5%	5
Volatility Rate	σ	√percent/√year	1	2	2	5%	0.1

To show the effect of uncertainty, fixed errors correspond to 5% of the center value. The grided parameters of Time-to-Expiry and Spot Price have errors based on the 5% applied to the maximums . In many studies, 95% is a common confidence level, meaning for future cases, we can expect 19 out of 20 trials to be predicted. The balance of this is 5%, meaning we expect predictions to fail in 1 of 20 trials. Considering error to be a failure-of-prediction, then a 5% relative error is suitable. The alternative is to have enough data of past behavior to assess the error level. However, the application of statistics to past data has limitations. For example, the description of rogue events or outliers that occur and do not fit the usual pattern of behavior is an issue. This remains a failure of statistics and a challenge to any new methodology.

Validation is by multiplying the input errors by zero, such that uncertainty arithmetic reduces to the Exact Arithmetic and this should correspond to the Traditional Arithmetic. This was completed successfully by all methods, verifying one aspect of proper uncertainty programming.

7.1. Traditional Arithmetic Results

The first calculation is the Black-Scholes model with Traditional Arithmetic. With no uncertainty information, there is only one graph, Figure 1 below, and this is the Call Value vs. Time-to-Expiry and Spot Price.

There is a missing part of the surface at zero Time-to-Expiry caused by using straight arithmetic to calculate ‘a’ in Step 13 of the calculation procedure. This shows one weakness in the Traditional Arithmetic – the divide-by-zero problem. One way to fix this, is to not use an ‘exact zero’ as the lower bound for Time-to-Expiry and instead use a small positive number near zero such as +0.01. This is essentially introducing a degree of error to fix a problem.

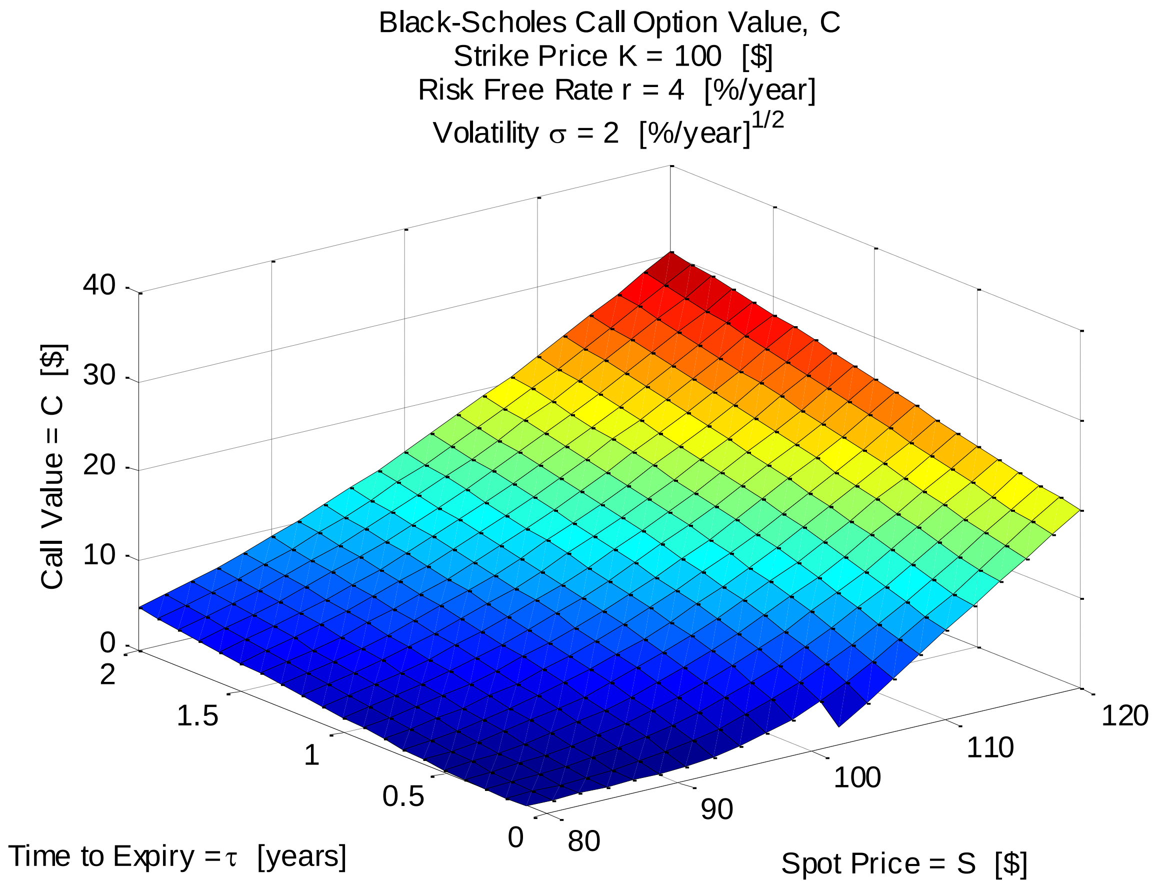


Figure 1 – Black-Scholes Model using Traditional Arithmetic

From this baseline case, there are common results and interpretations (refer to Figure 1 above) that occur in most of the arithmetic cases.

1. Call Value calculated for different grid points of Spot Price and Time-to-Expiry is plotted as a surface.
2. Due to the variable 'a', calculated from an inverse of Time-to-Expiry, there is a divide-by-zero challenge at zero Time-to-Expiry and this is called the 'critical edge'.
3. With feature 2 above, the case where Spot Price equals the Strike Price is a challenge and this is called the 'critical point'.
4. At low Time-to-Expiry and low Spot Price, the Call Value surface touches or crosses the zero Call Value reference plane. Cases above this plane define a region of profitability for the call option.
5. Considering a premium to buy the call option, this shifts the reference plane (of feature 4 above) upward and reduces the cases that yield profit.
6. The Call Value surface is not linear and has curvature in both directions. The Call Value increases with increasing Spot Price but at different rates due to the surface curvature (hedge).

The next set of graphs are for methods using dual numbers and a variety of arithmetic types. This extends each traditional number by attaching an error number to represent uncertainty.

7.2. Exact Arithmetic Results

Exact Arithmetic operates on a dual number that has zero error. The error field is formatted but remains zero. Figure 2 shows the resulting Black-Scholes surface for Call Value.

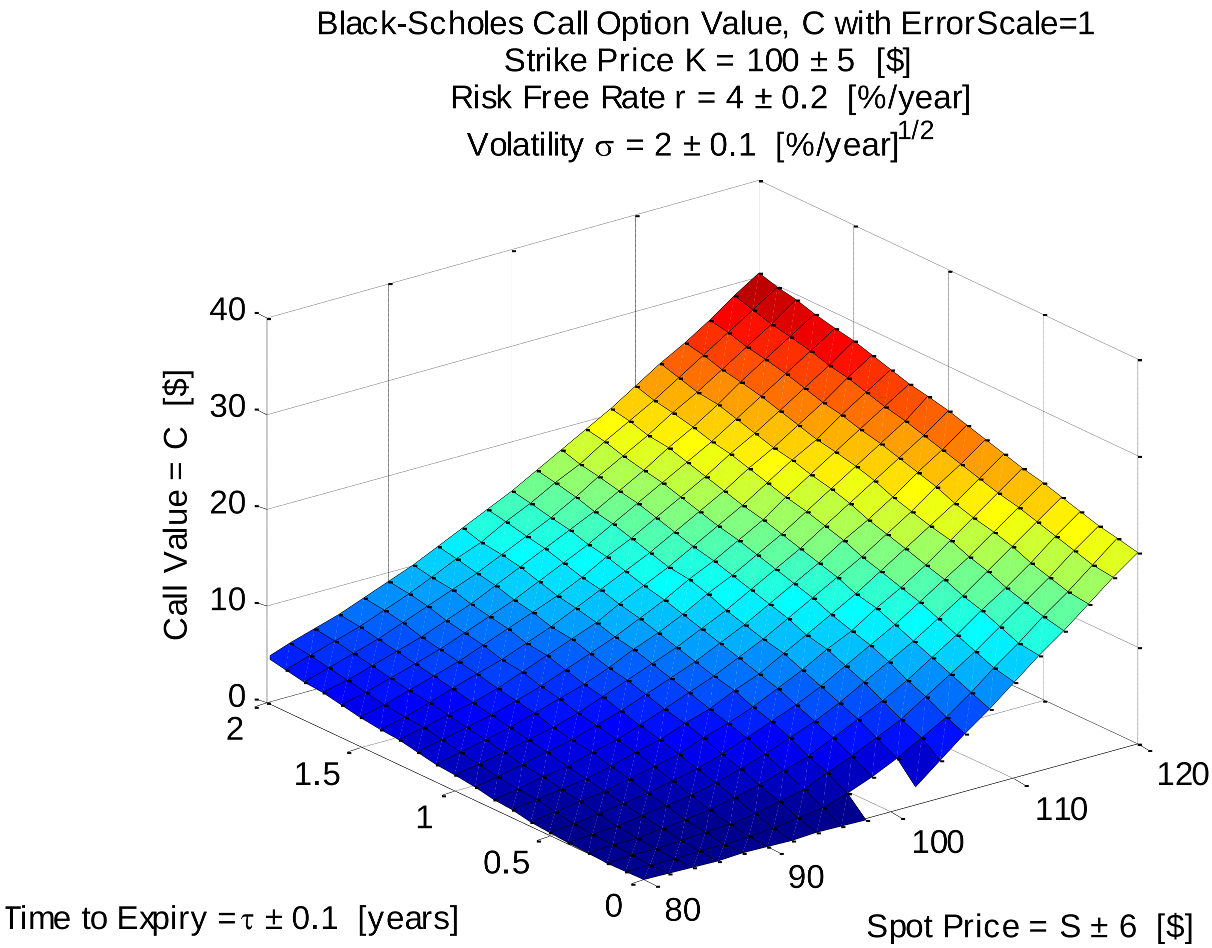


Figure 2 – Black-Scholes Model using Exact Arithmetic

This is identical to Figure 1 except now, by including an error number in the calculation, the divide by zero problem is reduced such that the surface panels at low spot price are complete and touch zero Call Value. However, there is still a cut-out region around the remaining divide-by-zero problem when the Spot Price is equal to the Strike Price. Theoretically this would be zero Call Value corresponding to the Figure 1 given by Black&Scholes [1] and the limit of zero Time-to-Expiry.

A second graph for the Error of Call Value surface is shown in Figure 3. It is non-interesting as a flat plane of zero error for the Exact Arithmetic. This confirms the calculation produces no error by itself and only propagates error from the uncertainty in the five input sources to the Call Value.

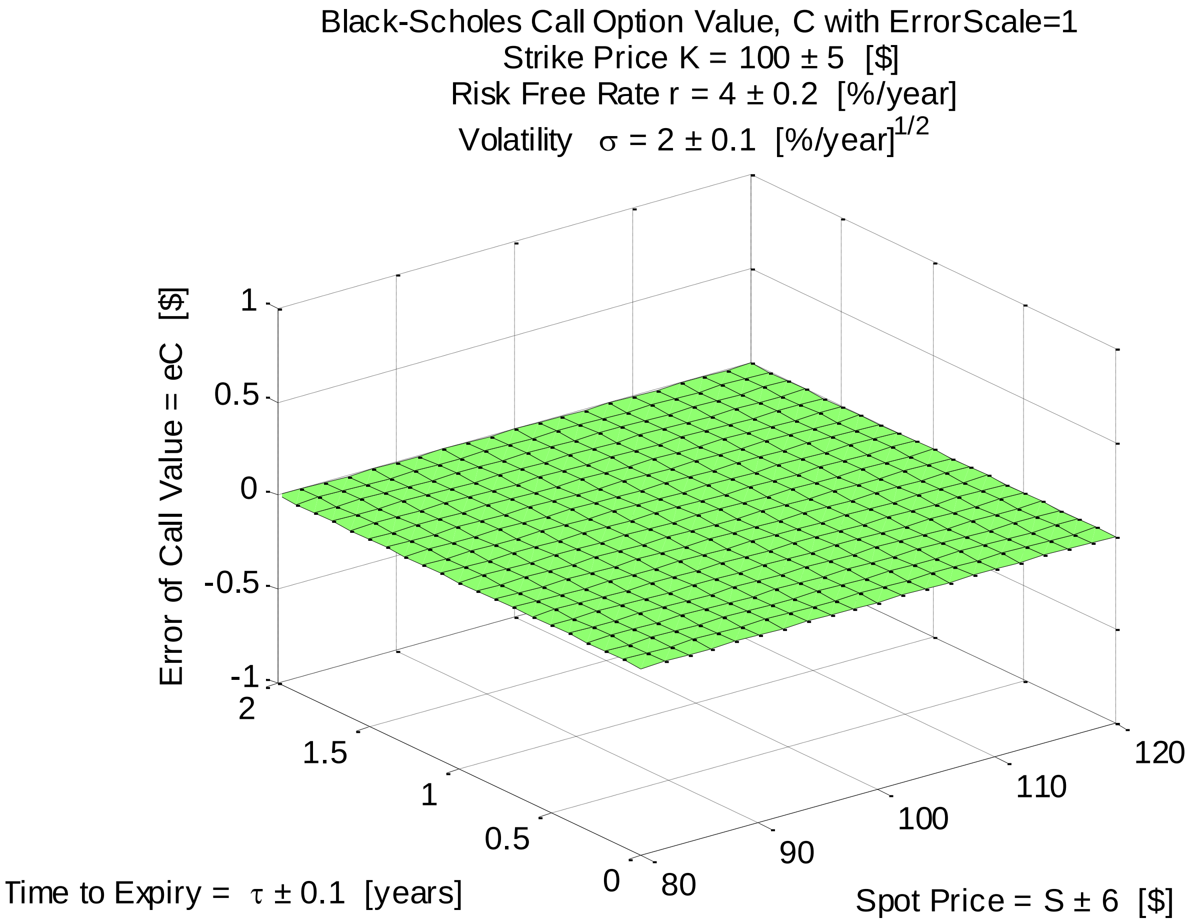


Figure 3 – Black-Scholes Model Error using Exact Arithmetic

7.3. Interval Arithmetic Results

Interval Arithmetic has difficulty with the Black-Scholes model at the critical edge of near zero Time-to-Expiry. The reason is the square-root-of-negative problem produced from step 2 of the procedure that propagates to the 'd' calculations and causes failure of the normal distribution function. To obtain practical results, the square-root-of-intervals function is equipped with a switch to not-report cases when the square-root-of-negative problem occurs. However this is not an arithmetic operation but instead is a relational operation.

The time-performance tests used the lowest grid density (3 points) for the intervals. For the best detailed results, the interval grid density is increased to (100 points). This means each binary operation has 10,000 calculated points to feed into the maximum and minimum and resolve the result dual number.

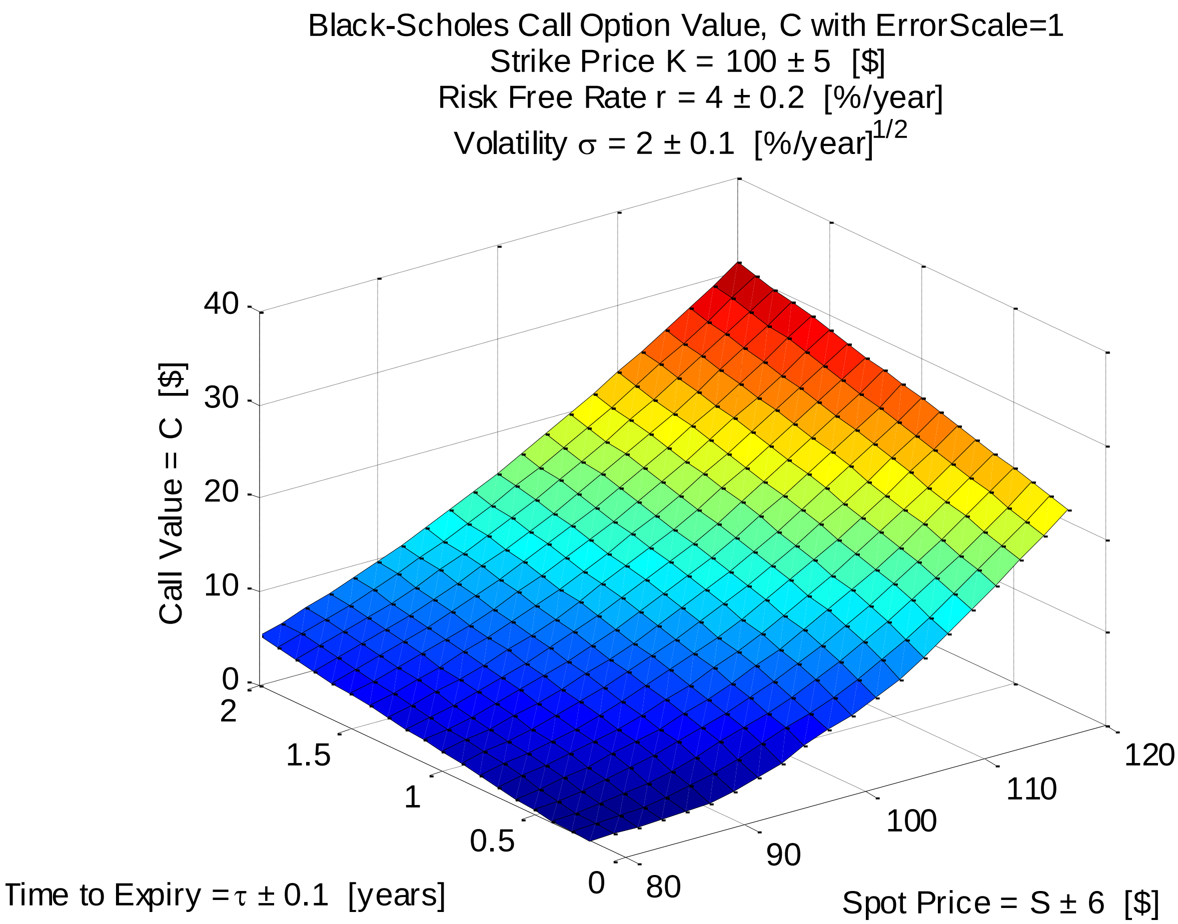


Figure 4 – Black-Scholes Model using Interval Arithmetic

Figure 4 shows the Interval Arithmetic produces nearly the same Black-Scholes surface as Traditional Arithmetic. However, the calculation fails not only at zero Time-to-Expiry, but also for the next time near zero. This is worse than the Traditional Arithmetic as the Interval Arithmetic superimposed the error in such a way that near-zero positive numbers can change to negative or very near zero.

There is also a slight ripple in the surface at low Time-to-Expiry and where Spot Prices are nearly equal to the Strike Price. This ripple is caused by the error affecting the center calculations. This means that uncertainty calculation is not a separate procedure that can be 'done later.' The error participates simultaneously with the center calculations and there is an interaction. Furthermore, all of the uncertainty calculating methods except Differential Arithmetic have error that interacts with the center, just as we would expect the center value to influence the error through surface slope.

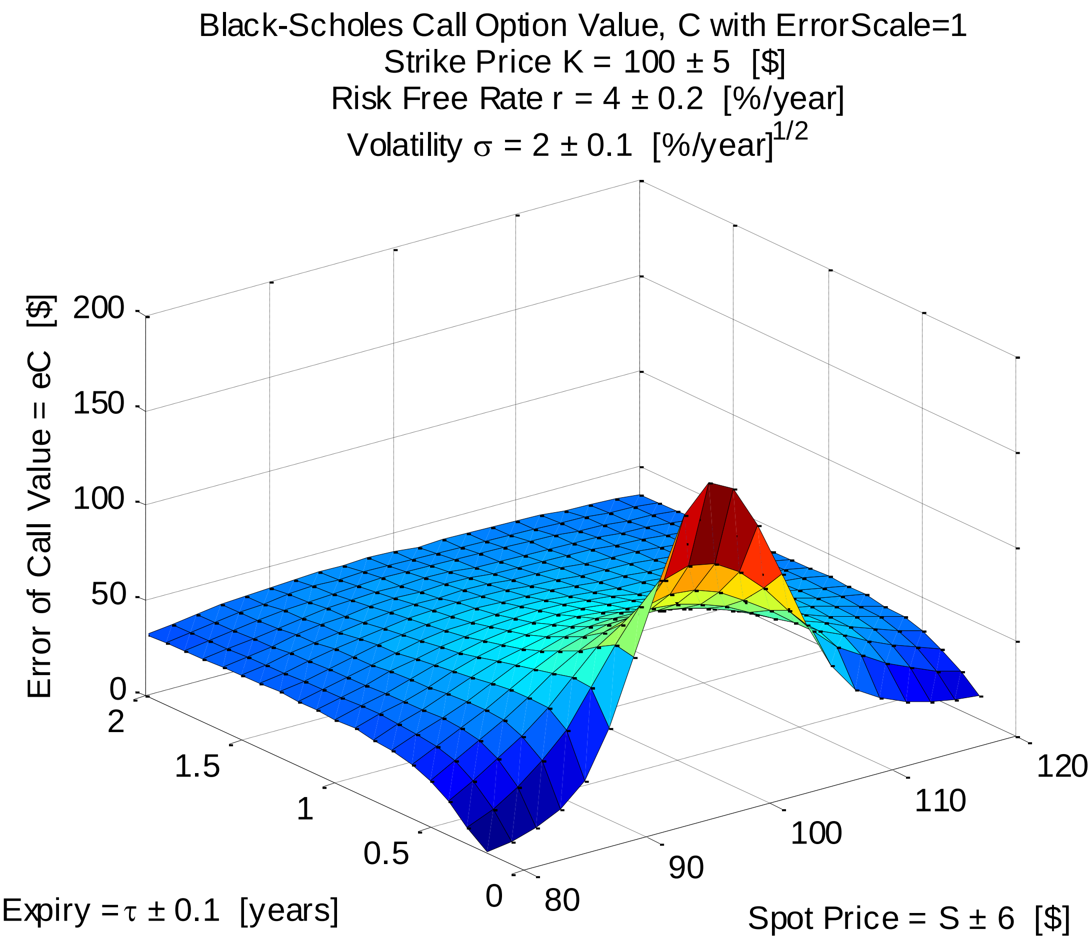


Figure 5 – Black-Scholes Model Error using Interval Arithmetic

Figure 5 shows the Black-Scholes surface for the Error of Call Value from the Interval Arithmetic. This shows a large amount of error, about $32, at high Time-to-Expiry. As the chosen Time-to-Expiry is reduced, the surface becomes a wave with a peak error where the Spot Price is equal to the Strike Price.

This bump is evidence of why there is a ripple on Figure 4 near the same case. Interestingly, as the Spot Price deviates from the Strike Price (gain or loss), there is a rapid reduction in the error. Due to the failure at the critical edge near zero Time-to-Expiry, we cannot tell if the surface touches zero error and the calculated Call Value becomes certain. Overall, the errors are too high to consider the Interval Arithmetic acceptable.

Summarizing Error of Call Value, common results and interpretations can be listed for Error of Call Value surfaces to be shown later.

1. The Error of Call Value for different grid points of Spot Price and Time-to-Expiry is plotted as a surface.
2. All methods except the Duals Arithmetic show a rapid growth of Error of Call Value as the Time-to-Expiry shortens.
3. All methods except the Duals Arithmetic fail the challenge of the critical edge near zero Time-to-Expiry and the challenge critical point at zero Time-to-Expiry and Spot Price equal to Strike Price. This means those methods do not support the theory for very short Time-to-Expiry. This may have an impact on using Black-Scholes European Style results to develop other 'quicker-update' option styles such as the American Style option.
4. Some methods have regions of very large Error of Call Value and this indicates those methods are not practical for calculating the uncertainty of the Black-Scholes model.

7.4. Monte-Carlo Arithmetic Results

The behavior of the Monte-Carlo Arithmetic follows a random process; therefore the results are not repeatable. To obtain understanding, the number of samples to populate each input number's error was increased to 100,000. Both the Call Value and Error of Call Value surfaces are not smooth. In addition, the errors are very large near the critical edge and critical point. Therefore, the error plot is limited by cutting-off error values above twice the Strike Price – *an outlandish amount of error*.

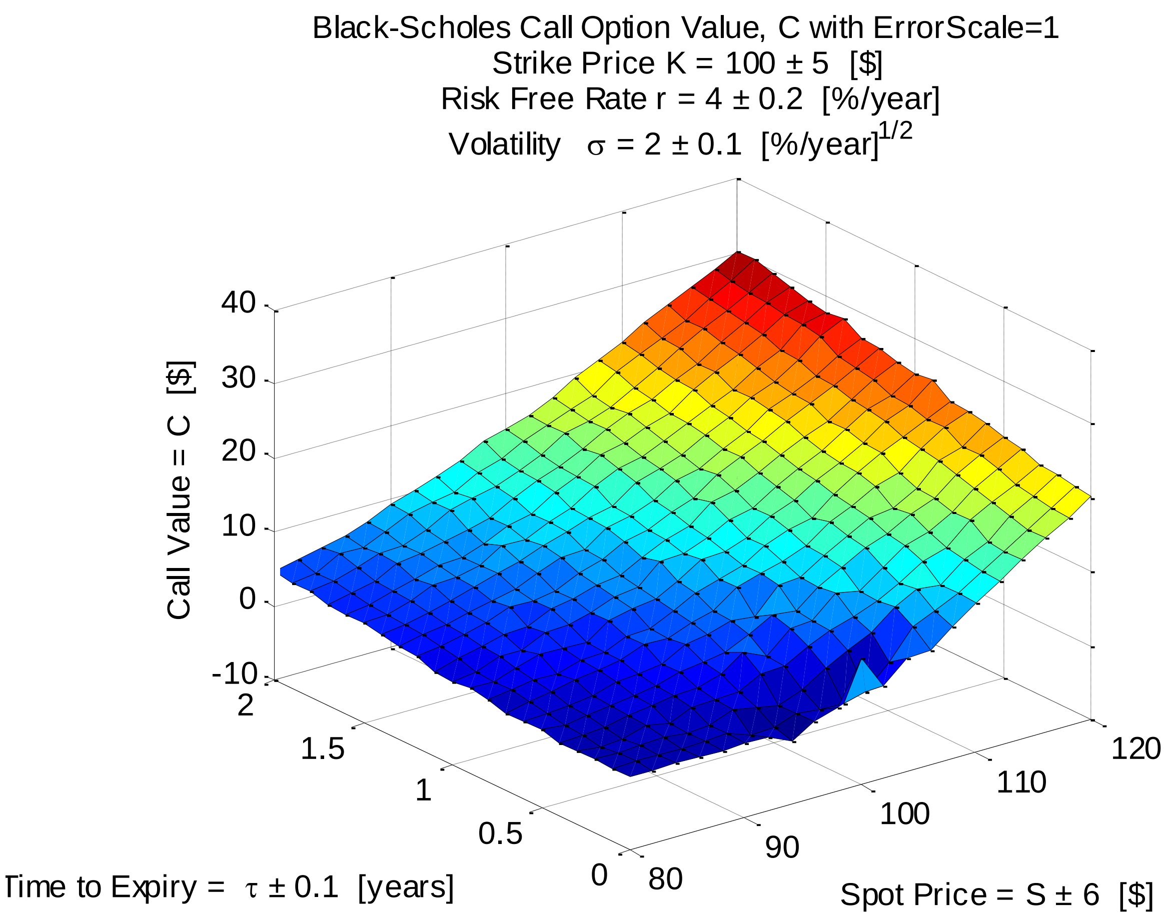


Figure 6 – Black-Scholes Model using Monte-Carlo Arithmetic

Monte-Carlo Arithmetic relies on random number generated population and the mean value of this population is the center value. Figure 6 shows the Call Value surface for Monte-Carlo Arithmetic. The surface texture is dictated by the variation present in every random population.

The randomness induces local changes in slope and curvature rapidly changing convex to concave. This change in curvature would alter the strategy of using the Black-Scholes surface as the comparison of +/- deviations away from any point would depend highly on local conditions of the point (see the description of curvature starting on page 638 and the demonstration of hedged position starting on page 641 of Black&Scholes [1]).

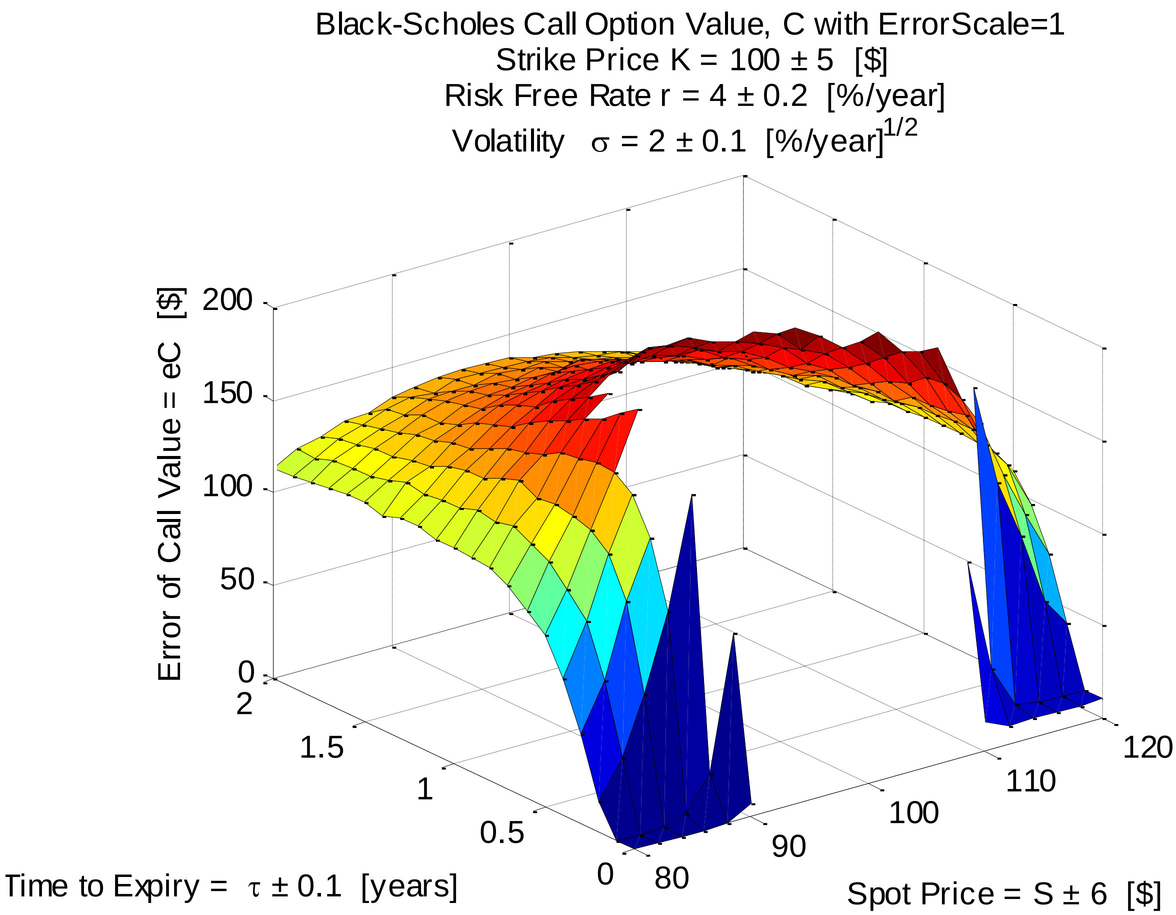


Figure 7 – Black-Scholes Model Error using Monte-Carlo Arithmetic

Figure 7 shows the Error of Call Value surface for Monte-Carlo Arithmetic. The texture is due to the randomness of statistics on 100,000 error instances. This calculation employed a logic switch to flip the sign of the superimposed random error and guard against square-root-of-negative. The critical edge benefits from this logical guard. The surface shows this as it rapidly descends to zero error at zero Time-to-Expiry. However, there is still a large problem region around the critical point where the Monte-Carlo Arithmetic fails and the errors are very large. These large errors are cut-off beyond two times the baseline Strike Price.

7.5. Differential Arithmetic Results

The Differential Arithmetic is sequential as it calculates the Call Value center without error first and follows-up with error propagation. Figure 8 shows that the Call Value surface is similar to the Exact Arithmetic (error is formatted but is set to zero).

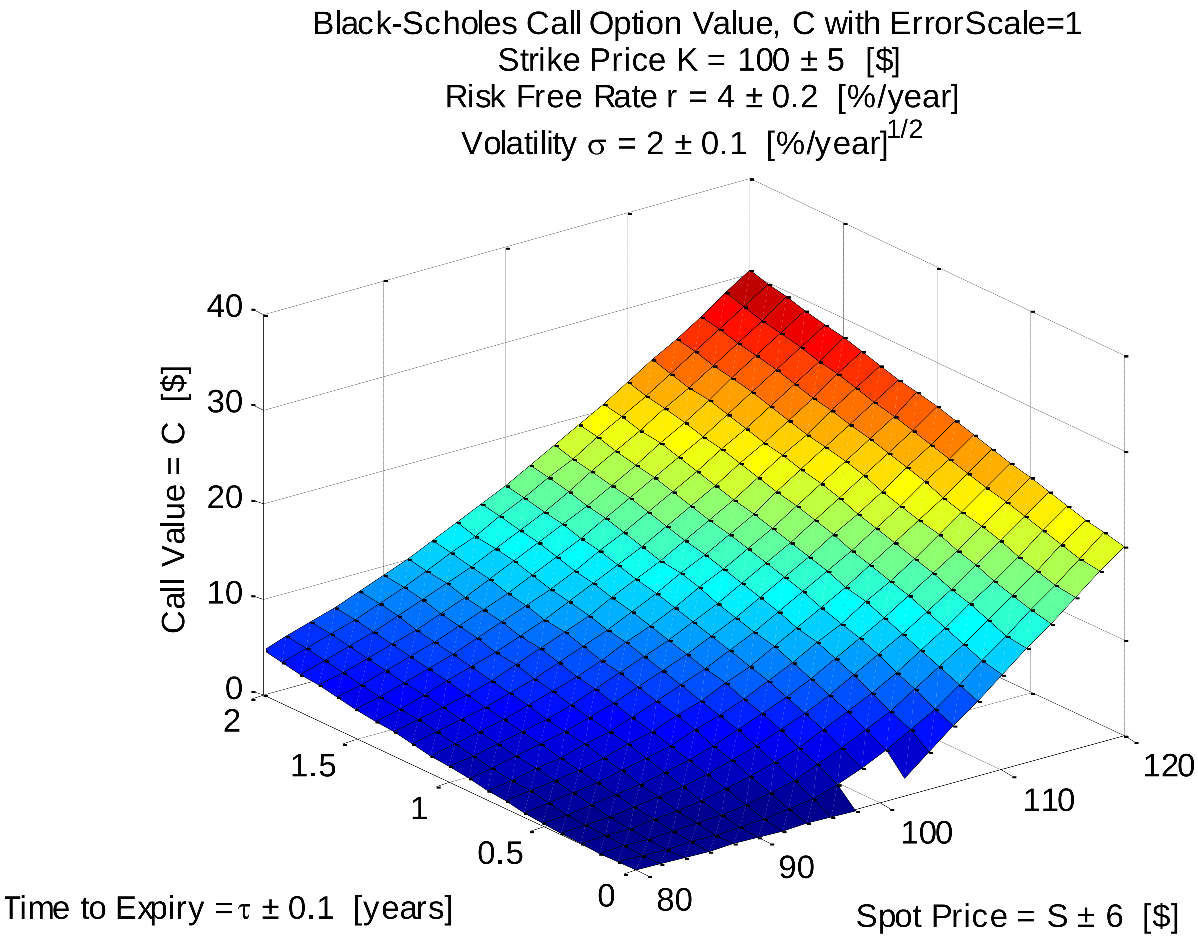


Figure 8 – Black-Scholes Model using Differential Arithmetic

This surface has a problem with the critical point. This problem is not influenced by error calculations because the Differential Arithmetic does not calculate uncertainty simultaneously. The surface's missing information is due to the 'divide-by-zero' problem in the 'a' calculation mentioned earlier.

After the center of Call Value is calculated, the Differential Arithmetic performs a secondary calculation to determine the Error of Call Value. This sequence makes the center independent of the error, but with center results established, the error calculation uses this information. Figure 9 shows the Error of Call Value follows the common surface pattern.

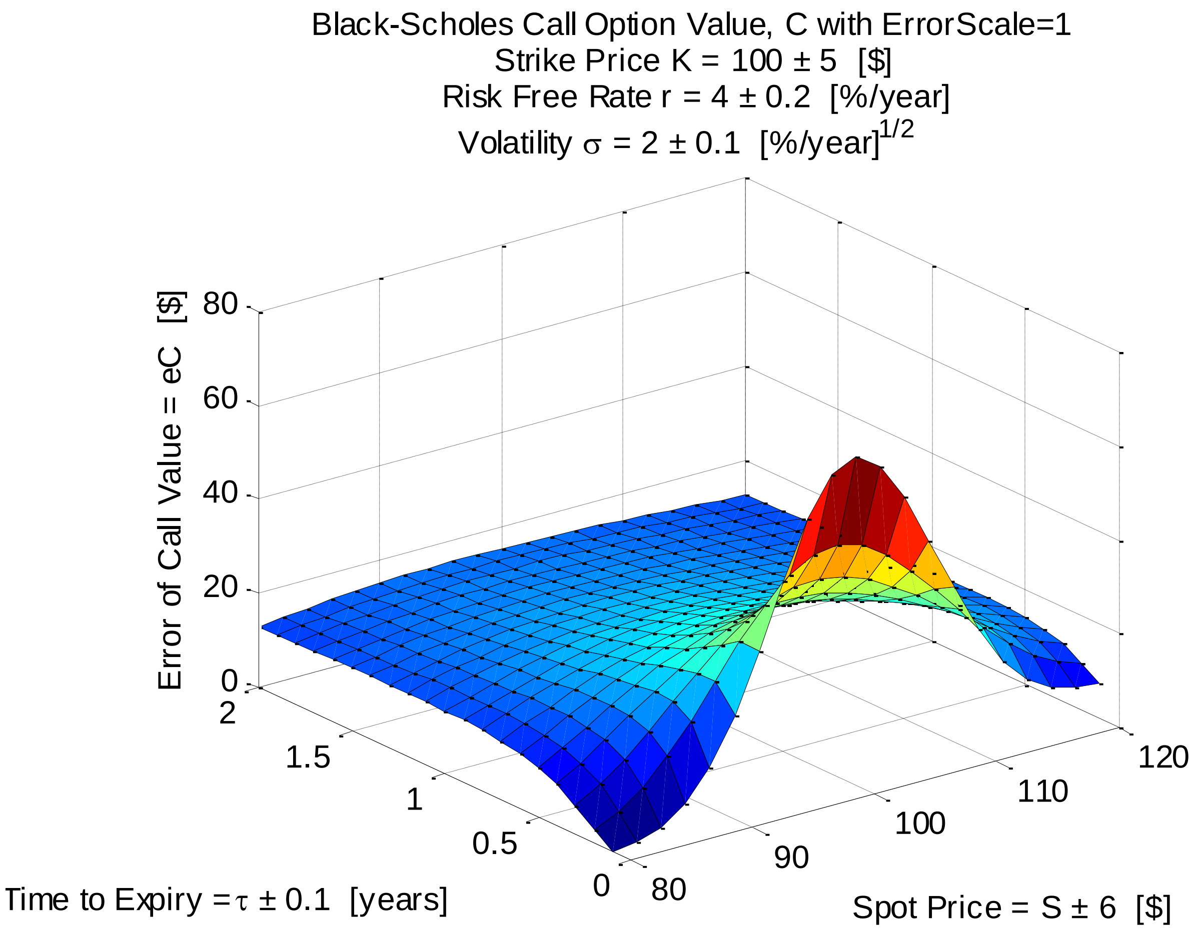


Figure 9 – Black-Scholes Model Error using Differential Arithmetic

The critical edge at zero Time-to-Expiry is missing as the Differential Arithmetic cannot divide-by-zero. The scale of the Error of Call Value is similar to Monte-Carlo and less than the Interval Arithmetic.

7.6. Chordal Arithmetic Results

The Chordal Arithmetic is based on the geometry of two points. These two points typically span the range of errors for each number and create a two-track calculation. This is similar to the Interval Arithmetic with two points on the grid, except now the maximum and minimum operations are not used. Instead inputs for the chordals have to be coordinated to the operation being performed.

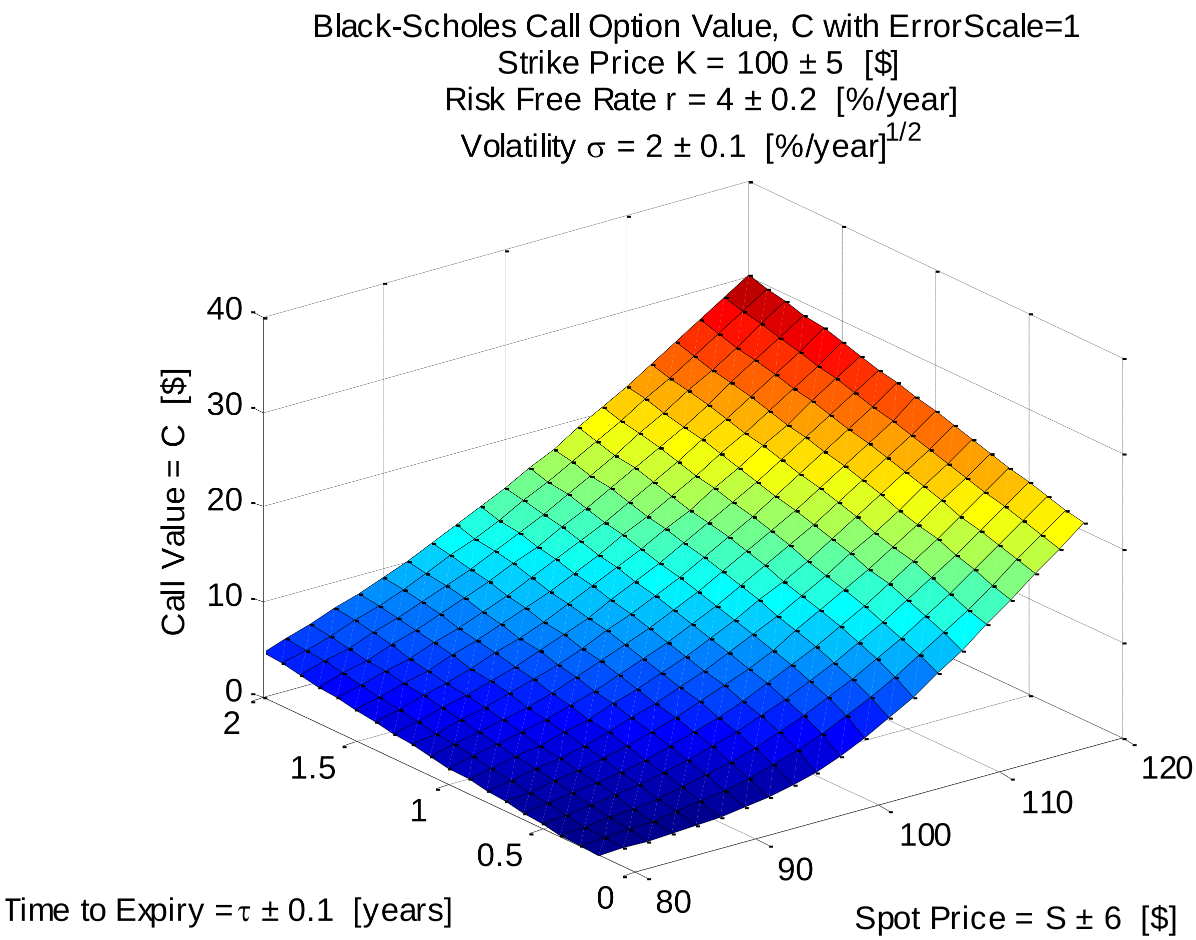


Figure 10 – Black-Scholes Model using Chordal Arithmetic

Figure 10 shows the chordal results are similar to the interval results. This is due to the fact that most of the geometric operations in the procedure satisfy a monotonic condition and have the condition of staying either positive or negative to exclude the possibility of zero. Under those conditions the chordal arithmetic's two point grid is similar to the interval method with a three point grid. With low points, the coordination of the chordal method matches the maximum and minimum operations. Like the Interval

Arithmetic, the Chordal Arithmetic fails at and near the critical edge. This is because a negative error case is superimposed on a small positive center and the net chordal number is negative.

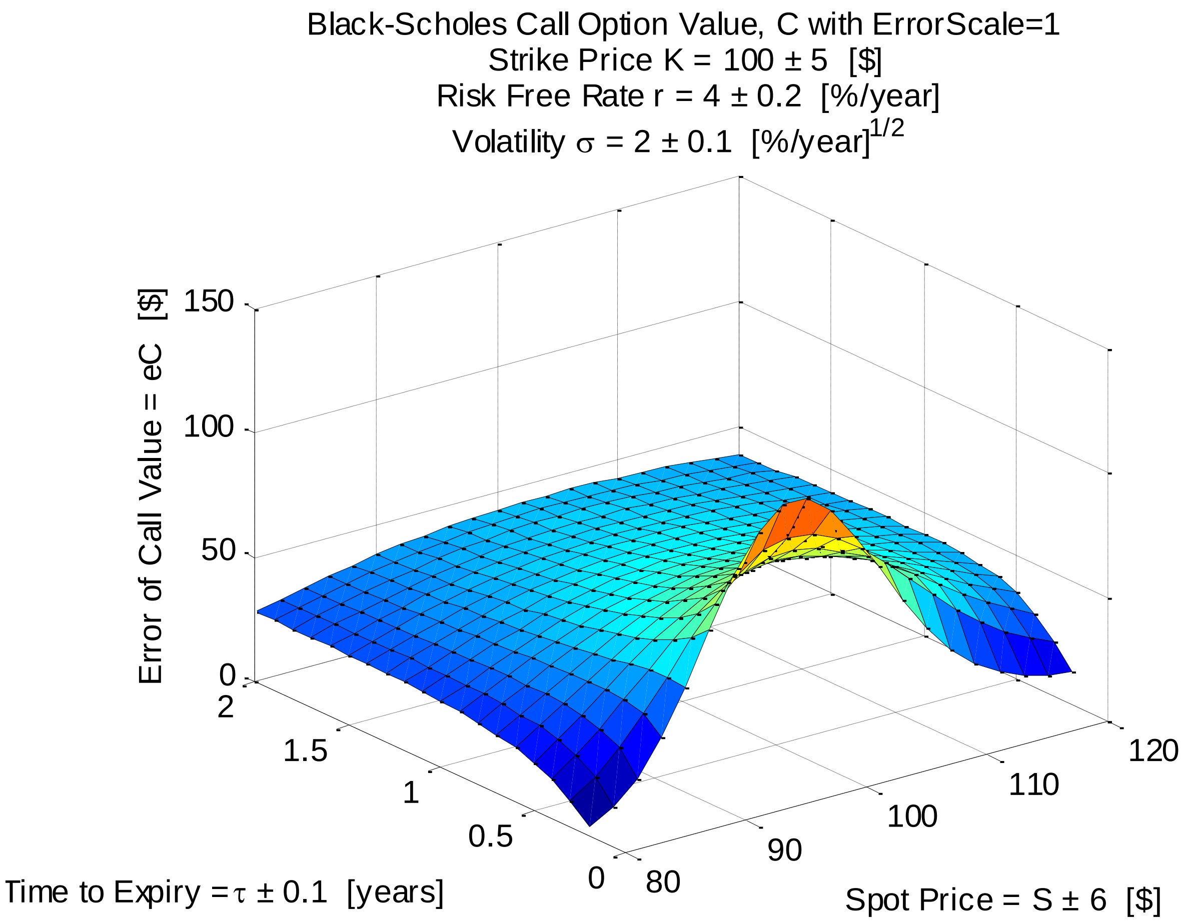


Figure 11 – Black-Scholes Model Error using Chordal Arithmetic

Similar to the Call Value, the results for Chordal Arithmetic for the Error of Call Value are similar to the Interval Arithmetic results. Again the surface is truncated at and near the critical edge but the error scale is slightly lower than the N=3 interval results.

8. RESULTS FOR DUALS ARITHMETIC

The Duals Arithmetic represents the center value as a scalar for a point and the error as an error vector of as many components (or dimensions) as there are inputs to a calculation. This yields a sophisticated way to calculate error and provides results that are amenable to manipulation of error.

8.1. Call Value and Error Magnitude of Call Value

The first graphs reported are the rendered dual number as a center and an error. Figure 12 shows the Call Value surface for Duals Arithmetic.

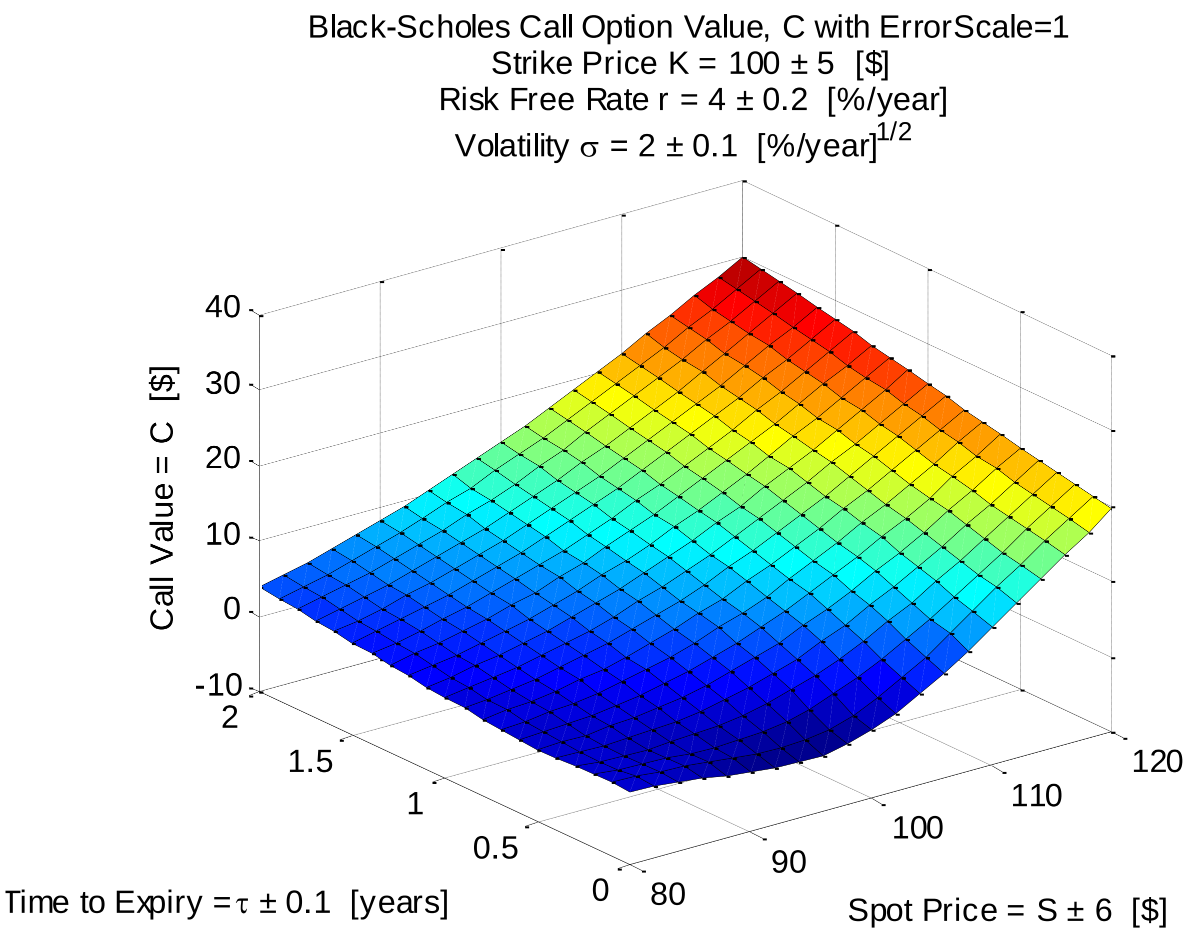


Figure 12 – Black-Scholes Model using Duals Arithmetic

At first glance, this surface follows the common pattern of the Call Value using the Black-Scholes model. However, the Duals Arithmetic is distinct from all other methods because does not have any calculation problems at the critical edge and critical point. This is because the Duals Arithmetic is robust and can tolerate divide-by-zero (zero with error) and square-root-of-negative problems that normally cause very bad results or halt a calculation. A distinct feature is the surface dips below zero in a region near the critical point. Considering the premium, this will limit the profitability of the call option and, compared to Traditional Arithmetic, reduce the number of situations where one would decide to purchase a call option. This is caused by the simultaneous influence of uncertainty on the center value.

Figure 13 shows the Error of Call Value as the rendered magnitude of the 5D error vector. The rendering is accomplished using a Pythagorean sum.

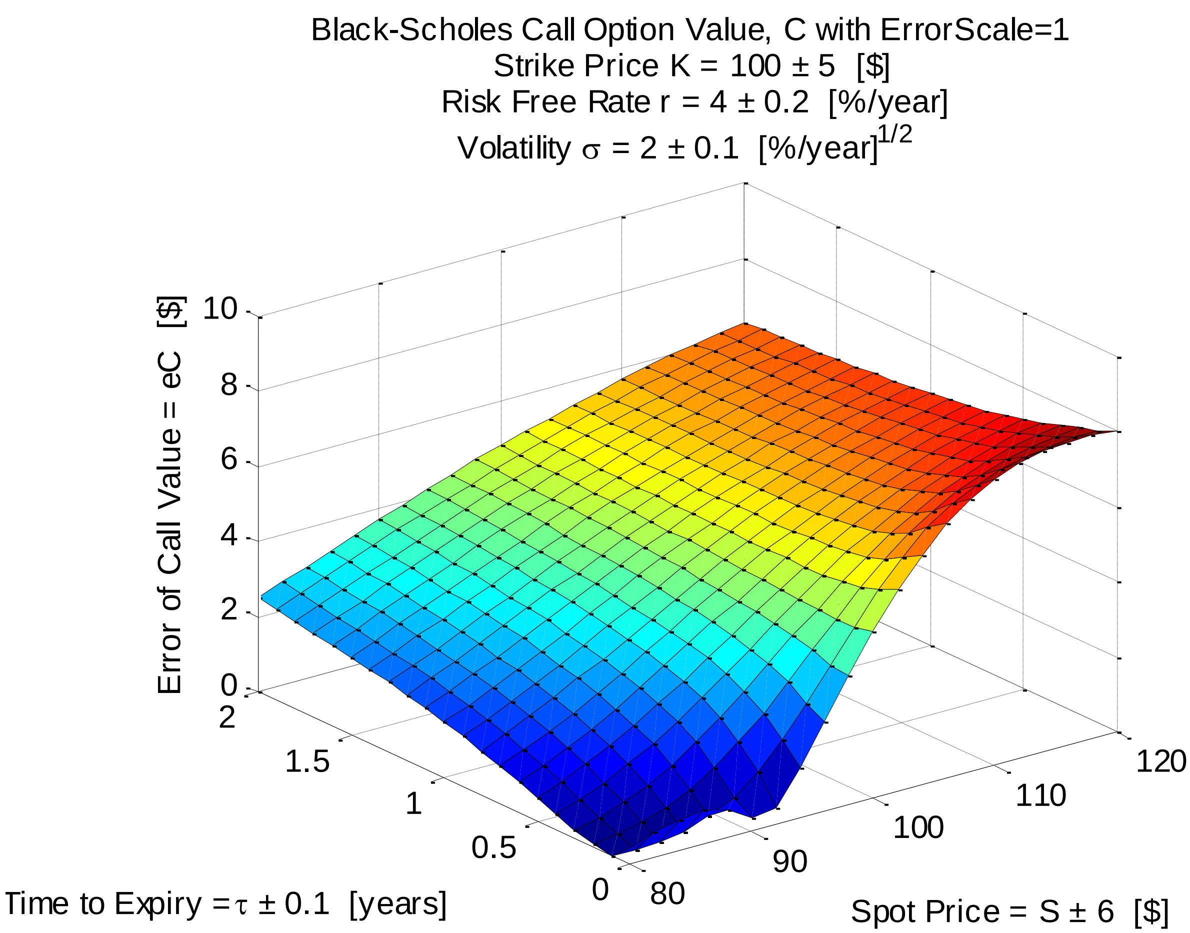


Figure 13 – Black-Scholes Model Error Magnitude using Duals Arithmetic

This surface shape is distinct from the other methods that have problems at the critical edge and critical point. There is no evidence of problems except in the small region of zero Time-to-Expiry and low Spot Price where the surface has a small curl upward and defines a small, minimum error valley. The key feature is, instead of a ‘zero slope’ surface emanating from the critical point to longer Time-to-Expiry, the Duals Arithmetic provides a surface with Error of Call Value increasing with increasing Spot Price. The slope of this surface is a maximum at the critical point rather than the ‘zero slope’ from other methods. However, this slope decreases away from the critical point and ultimately at high Spot Price, the uncertainty reaches a ceiling of about $8. Compared to the other methods, this is a much lower uncertainty ceiling.

There is also no rapid growth of error as Time-to-Expiry is shortened; the surface is relatively flat in the time direction. However, at the extremes of Spot Price there are opposite trends of the Error of Call Value vs. Time-to-Expiry; one going up and the other going down over time. This indicates an organized twist to the surface. If the error magnitude is to be reduced, it appears that a Spot-Price below the Strike Price is favorable. But Call Value is higher with Spot-Price above Strike Price. In the context of the premium, there may be situations where profit is accomplished with lower error and higher certainty. There is a trade-off in the simultaneous use of both the Call Value and Error of Call Value surfaces.

If this error ceiling and therefore, the overall scale of the Error of Call Value can be manipulated, then overall higher certainty can be provided. The key is ‘how to manipulate the Error of Call Value.’ To do this, we have to look at how the five inputs each contribute to the overall Error of Call Value. The only method that provides this information is the Duals Arithmetic. To get this information from other methods would require a complete re-work or re-use of the calculation steps, increasing the runtime proportionally to obtain the sensitivity-to-input information.

8.2. Error Contributions by Duals Arithmetic

The Duals Arithmetic represents every number as a center and a 5D error vector. Since the center is included, there is a mechanism for simultaneous error calculation and center values are influenced by all five error vectors. This provides a direct method for investigating sensitivity of Error of Call Value by using components of the final error vector.

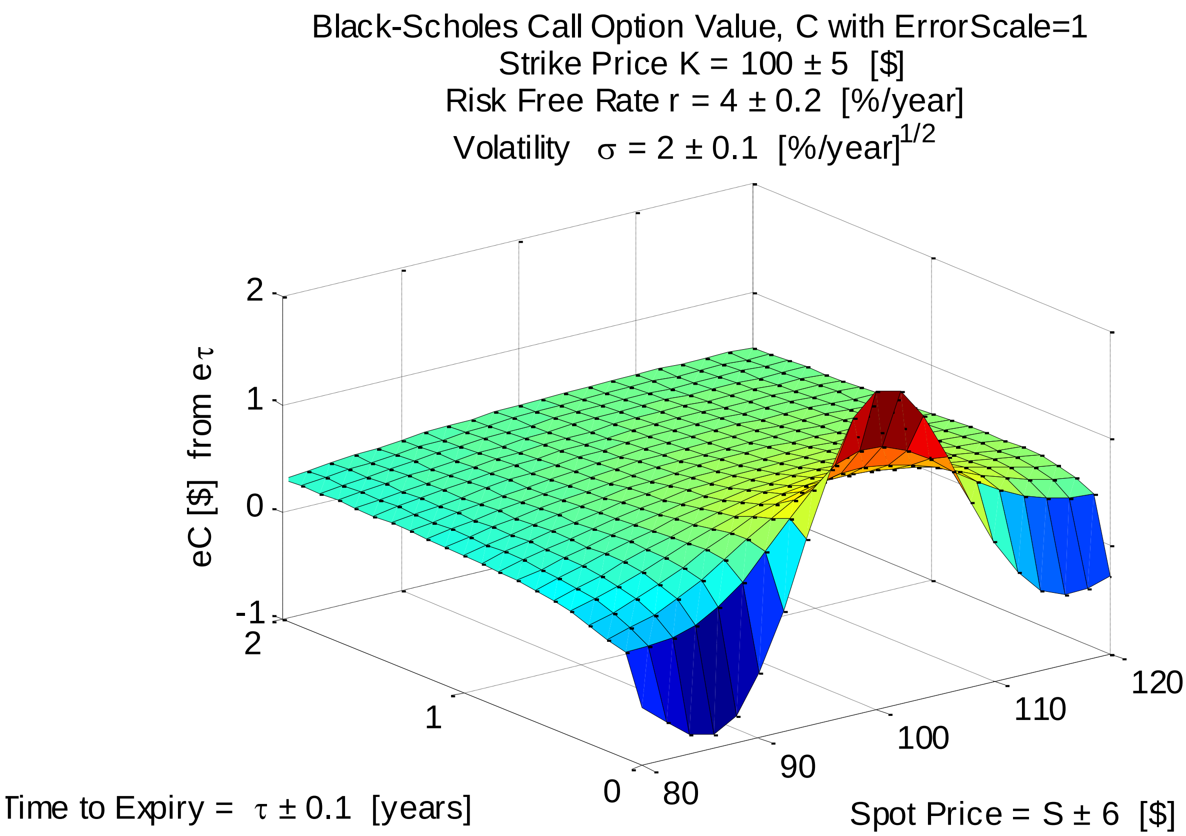


Figure 14 – Error of Call Value Due to Error of Time-to-Expiry using Duals Arithmetic

Figure 14 shows the Time-to-Expiry component of the Error of Call Value follows a similar surface shape when compared to the Error of Call Value surfaces shown earlier for other methods. However, the error scale is much smaller. The surface shape suggests that the contribution from time errors is significant in the other methods. But this information is not available in the other methods. As Time-to-Expiry nears zero there are regions where this error vector component 'does a 180' changing from a positive direction to a negative direction as the Spot Price varies. However, its contribution to the overall Error of Call Value will still be positive as the independent vector component is squared in the Pythagorean sum.

This surface is similar in shape to 'Greek theta' related to negative partial derivative of C with respect to time [10]. However the units of the Error of Call Value are in [dollars] while 'Greek theta' has the units of [dollars/year]. This connection suggests that the Duals Arithmetic is a way to automatically calculate 'Greeks' and provides strategic information.

Some similarities of components of the Error of Call Value vector can be noticed when compared to 'Greeks.' Once a formula is known, such as the Black-Scholes model, one can proceed on a path of generating first, second and higher derivatives. This is a straightforward use of differential calculus and well-known mathematics such as Series Expansion. However, differential calculus relies on small differentials (theoretically zero errors applied to an algebraic expansion of differences) and this essentially assumes a local linear behavior. On some level this ignores curvature (important for hedging) and motivates the next round of differential calculus. On the other hand, Duals Arithmetic is not limited to small errors and uses geometric arithmetic rather than differential calculus.

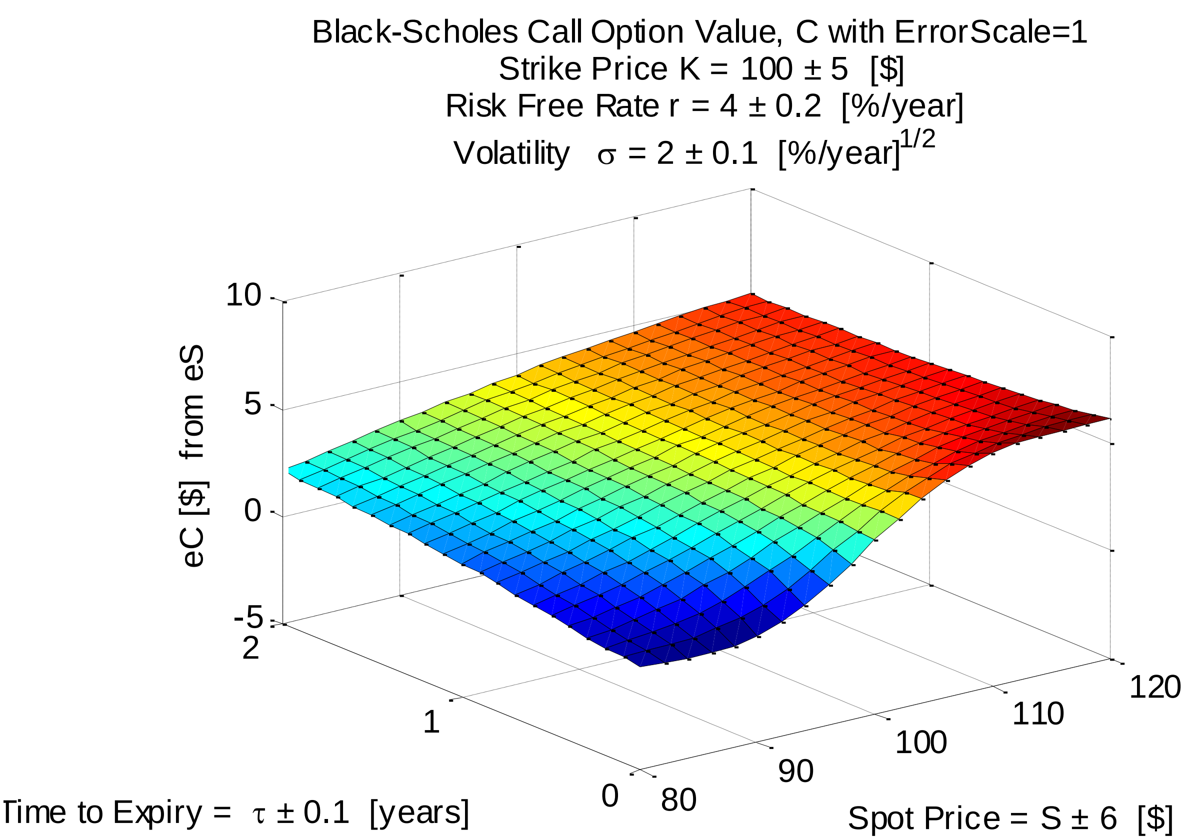


Figure 15 – Error of Call Value Due to Error of Spot Price using Duals Arithmetic

Compared to Figure 13, the Error of the Call Value due to the Error of the Spot Price is not only a large contribution but also dictates the shape of the magnitude of Error of Call Value surface. This has the same twist and trends at extreme Spot Prices. Considering that error transmission of the Error of Spot

Price to the Error of Call Value is due in part to the slope of the Black-Scholes surface of Figure 13, slope of the surface in Figure 15 gives some indication of curvature. Similar to the 'Greek theta,' this surface is similar in shape to the 'Greek Delta' related to a partial derivative of C with respect to S. However, Delta is unitless, representing a potential, while the Error of Call Value has the units of [dollars], representing the error that would be realized in money units.

The curvature (related to the 'Greek Gamma') establishes a hedging strategy by generating a difference when +/- deviations in Spot Price are compared (see the demonstration of hedged position starting on page 641 of Black&Scholes, [1]). However, there is more information in Figure 15 than just slope and curvature due to Spot Price. Each vector component of error has the ability to communicate many effects that accumulate through the calculation.

The Spot Price appears in three places in the procedure and each of these affects the results differently. Since the Duals Arithmetic solves the 'dependency problem', it is the only method discussed in this report that correctly accounts for error from a 'multi-pronged' input. This is enabled by representing error as a multi-dimensional vector and this suspends the rendering process normally used in other methods.

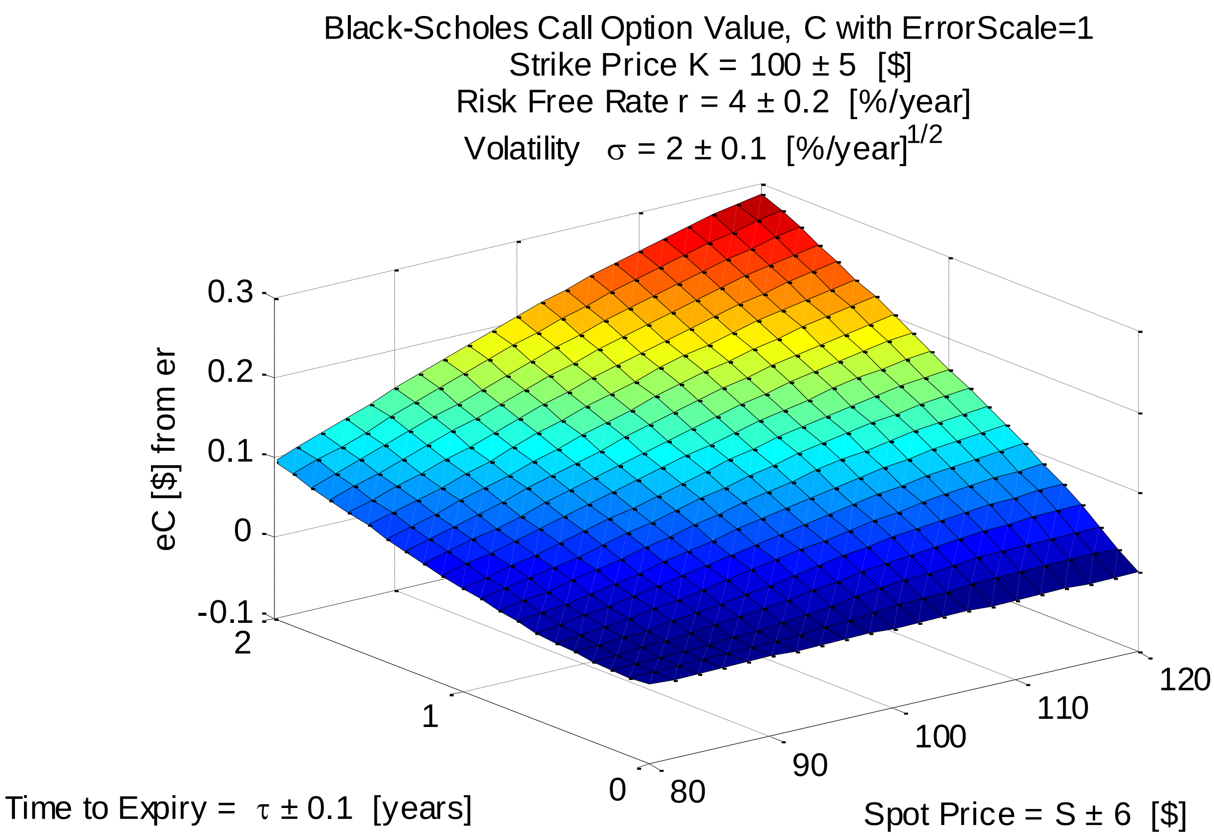


Figure 16 – Error of Call Value Due to Error of Risk-Free Rate using Duals Arithmetic

Figure 16 shows the contribution of the Error of Risk-free Rate on the Error of Call Value. The amount of error from the rate-error is very small (cents on the $100). It also is highest at the beginning time period of the call option and diminishes, becoming zero at the exercise time. This means a small uncertainty in the risk-free rate does not have much effect on the Call Value. However, the Call Value does change significantly if the Risk-free Rate (the center, not error) changes and all else is held fixed. This is built into the Black-Scholes model as shown by discount factor for the Strike Price and influence on the probability-drift of the fluctuating prices. This surface is similar in shape to the 'Greek rho' related to a partial derivative of C with respect to r. However, the units do not match and the Error of Call Value is in [dollars]. This verifies the relative unimportance of variation of Risk-free rate on Call Value pricing.

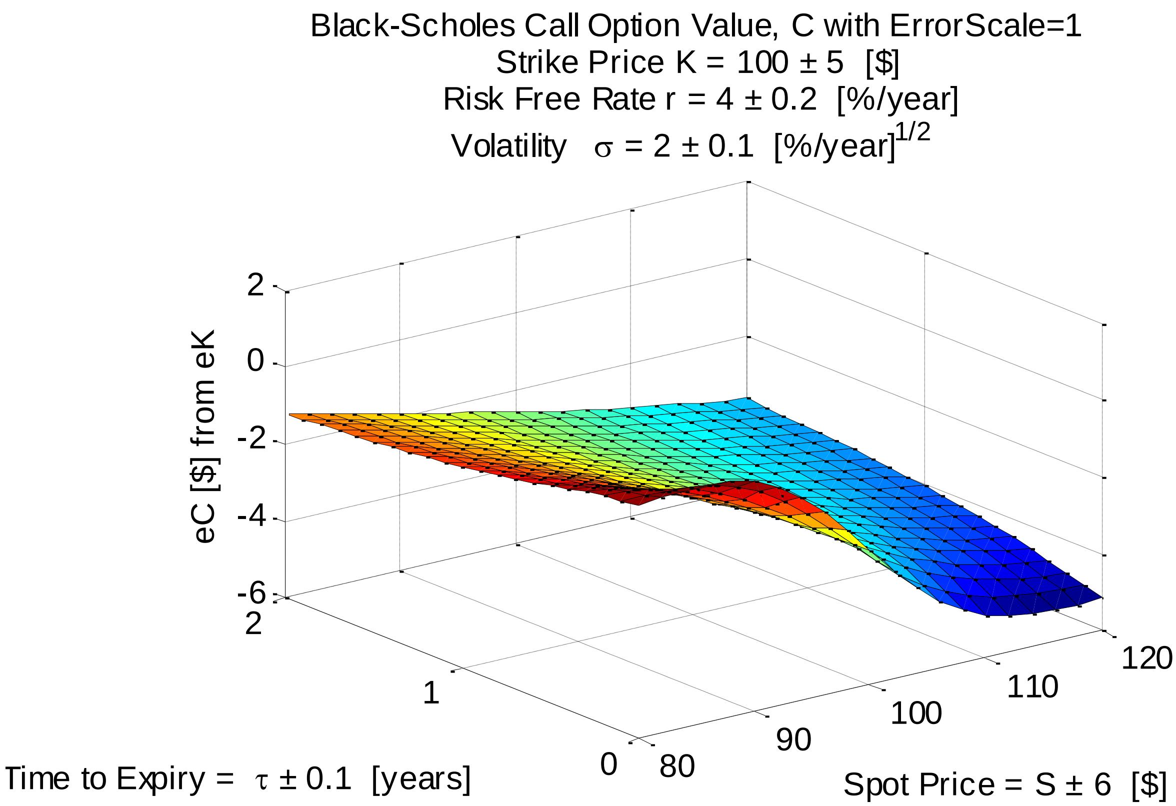


Figure 17 – Error of Call Value Due to Error of Strike Price using Duals Arithmetic

The Call Value plotted against Spot Price because that is watched and reported over the life of the option. The Black-Scholes model provides information when the purchase of a call option is being considered and the Strike Price is the key target to gauge the future stock price. Discounting is used to roll back the Strike Price to a time when the option is purchased. In the Duals Arithmetic, this discounting calculation is counted on at least three dimensions of the error vector.

Uncertainty in the Strike Price is calculated when the option is being considered and is also subject to the discount factor. Figure 17 shows a negative error vector contribution to Error of Call Value. Its largest contribution is at higher Spot Prices. This means the higher the expected Spot Price the more uncertain it becomes due to Strike Price uncertainty. The time-dependency of this surface is not great but has an interesting curl-toward-zero at low Spot Price. This may be the reason for the $1 curl observed on the Error of Call Value surface shown in Figure 13 that occurs when other error vector components become small at near zero Time-to-Expiry and low Spot Price.

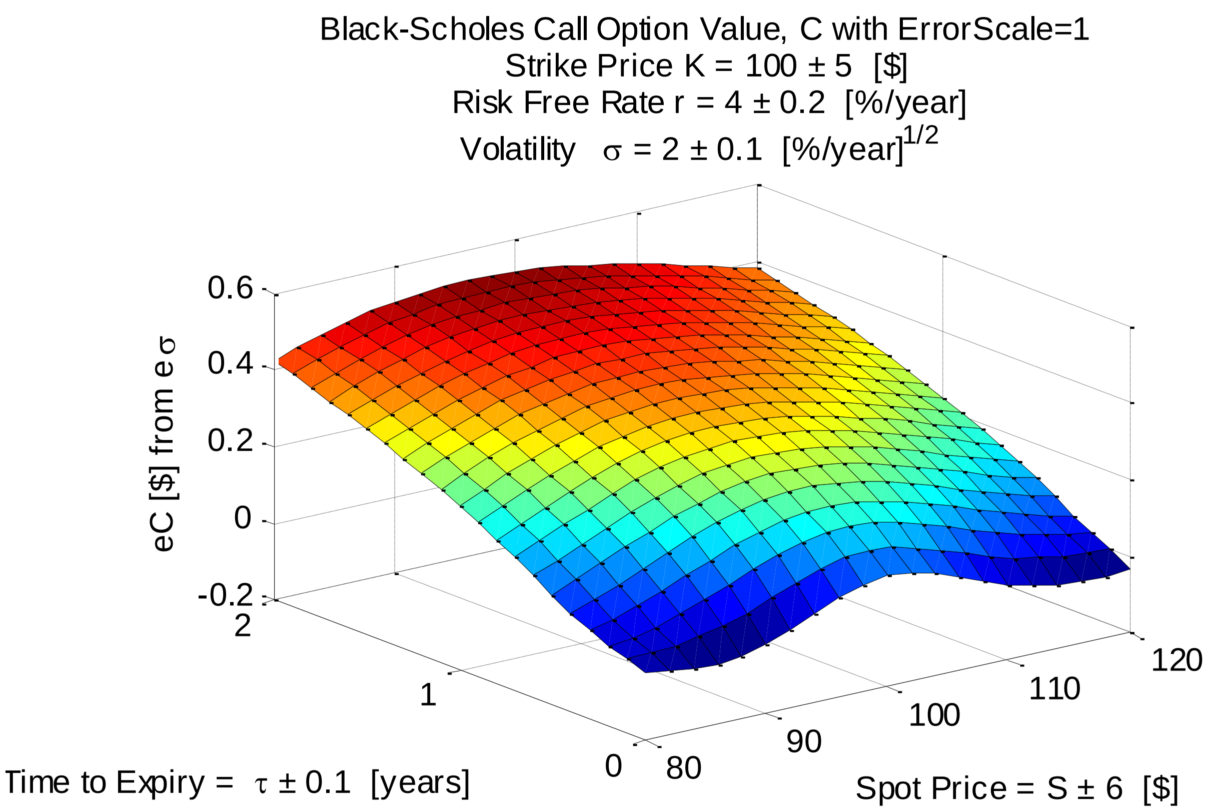


Figure 18 – Error of Call Value Due to Error of Volatility using Duals Arithmetic

Volatility acts through the log-normal distribution probabilities as if it alters the Risk-free rate. Its square is the variance of the price fluctuation process (assumes a historic trend is amenable to statistics with no great outlier events). However, the Time-to-Expiry (or the direction of time), affects how the contribution of volatility (or variance), alters the effective rate. Figure 18 shows, like the Risk-free rate, the Error on Volatility is a small contribution to Error of Call Value (cents on the $100). However, since the variance is analogous to a rate adjustment, the error due to volatility is about twice that of the error due to risk-free rate. Similar to Figure 16, the surface slopes downward with lowering Time-to-Expiry. This means the longer time the volatility is allowed to act on the projection, the more contribution its error has to Error of Call Value. This surface is similar in shape to the 'Greek Vega' related to a partial derivative of C with respect to sigma. However, the units do not match and the Error of Call Value is in [dollars]. This verifies the relative unimportance of variation of Volatility on Call Value pricing.

8.3. Error Budget by Duals Arithmetic

The rendered error for the Duals Arithmetic is the magnitude of the 5D error vector. Similar to other methods, this yields a dual number. However, the magnitude does not capture the contributions from each input. Examining the five contributions provides a basis for understanding major and minor contributions. This understanding can be turned around to manipulate the contributions and reduce the uncertainty in the Black-Scholes model.

Normally the contributions to an arithmetic sum can be judged according to the total.

$$T = c_1 + c_2 + c_3 + c_4 + c_5$$

For example, each contribution can be divided by the total to get a relative contribution indicated by percent. This makes it easy to judge the contributions as the total of the relative contributions has to be 100% and these can be plotted as a pie chart with five slices adding to the whole

$$1 = \frac{c_1}{T} + \frac{c_2}{T} + \frac{c_3}{T} + \frac{c_4}{T} + \frac{c_5}{T}$$

The problem with contributions to the error vector is that this equation <u>does not apply</u> directly. Instead, the total is calculated using a Pythagorean sum such as (the point equation with uniform signatures)

$$V^2 = v_1^2 + v_2^2 + v_3^2 + v_4^2 + v_5^2$$

With the contributions, c, and total, T, defined as squared values, then the relative contributions add to 100% and a pie chart can be plotted to judge major and minor contributions.

$$1 = \frac{v_1^2}{V^2} + \frac{v_2^2}{V^2} + \frac{v_3^2}{V^2} + \frac{v_4^2}{V^2} + \frac{v_5^2}{V^2}$$

One problem is that, when squaring, small components become relatively smaller and larger components become larger. This distorts the 'linear' contributions represented in the multi-dimensions.

This approach forms an 'error budget' showing the five contributions. It does not matter if an error component is +/- as the square is always positive. This error budget and a corresponding pie chart exist at every case of Spot Price and Time-to-Expiry. Instead of looking at 441 pie charts, a survey of just four charts, one from each of the four corners of the grid, are shown. These are labelled by the paired points (Time-to-Expiry, Spot Price) = (0, 80), (0, 120), (2, 80) and (2, 120) and the relative contributions are rounded to whole numbers.

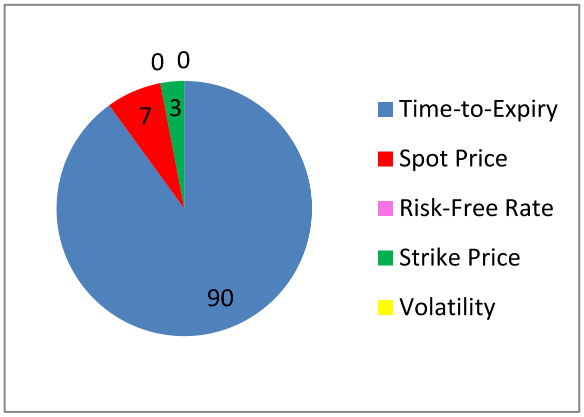


Figure 19 – Error Budget for Corner (0, 80)

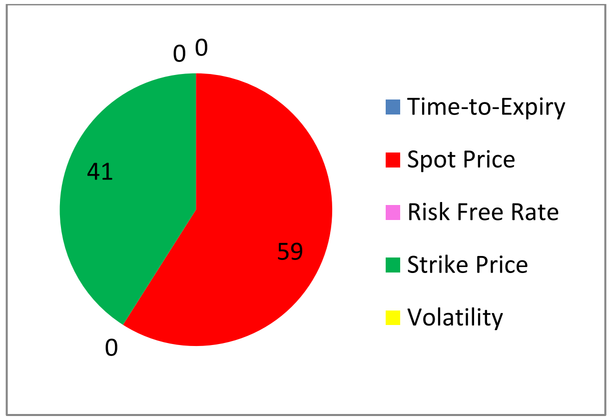


Figure 20 – Error Budget for Corner (0, 120)

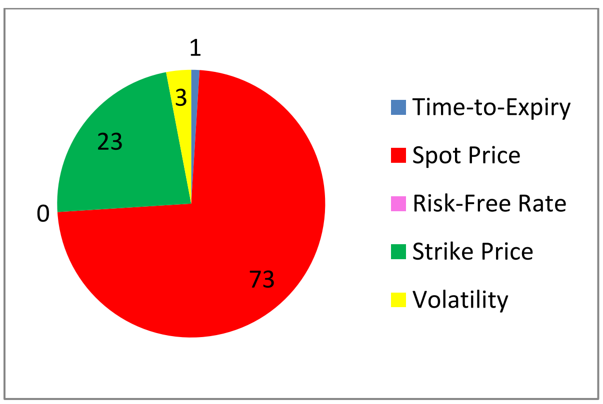


Figure 21 – Error Budget for Corner (2, 80)

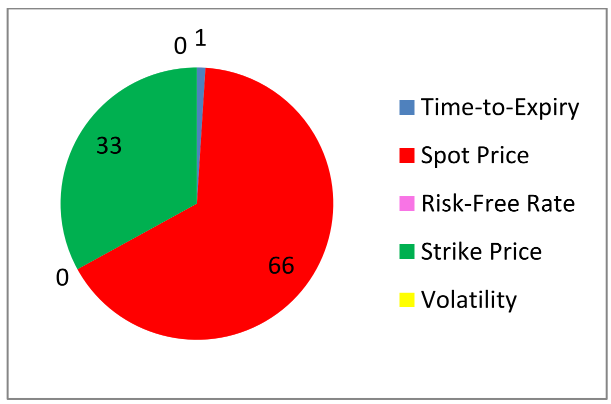


Figure 22 – Error Budget for Corner (2, 120)

The Spot Price is by far the largest contribution. This means to improve the certainty of the Call Value, lower error or better information on the Spot Price should be pursued. For example, the calculation shown used a 5% error on the Spot Price and better knowledge, such as a 4% or 3% error would improve the certainty of the Call Value. This seems like an obvious benefit, but the Duals Arithmetic provides a quantitative way to assess what is needed and its potential impact. Once the call option is written and purchased, the stock price variation becomes 'natural', subject to free and controlled market forces. However, the Black-Scholes model is intended to simulate these future changes using the risk free rate, volatility and log-normal distribution from a random process. There are always effects that do not follow the Black-Scholes theory and the uncertainty may bookend the range of expectations.

The Strike Price is the second largest contribution. Similar to the Spot Price, higher certainty of the Strike Price directly benefits the Call Value. The Duals Arithmetic shows that Strike Price error is important to reduce, but not as important as Spot Price error.

The Error of Risk Free Rate has very little contribution to the Error of Call Value and rounding essentially zeros its contribution compared to the Spot Price and Strike Price. The Duals Arithmetic provides the knowledge that exact information on the Risk-Free Rate is not necessary and the Black-Scholes calculation could be simplified by holding the Risk-Free Rate as an error-free number and using a 4D error vector in the Duals Arithmetic.

The Error of Time-to-Expiry generally has a very small contribution to the Error of Call Value. However, near zero Time-to-Expiry and at low Spot Price its contribution becomes dominant over all other contributions. This rapid change in the contribution is due to Spot Price and Strike Price errors becoming zero for those cases, allowing the Error of Time-to-Expiry to emerge.

The Error of Volatility is a minor contribution to the Error of Call Value. At long Time-to-Expiry and low Spot Price, there is a small contribution. However, the only significant contributions for those cases are the Error of Spot Price and Error of Strike Price and those are nearly the whole of the Error of Call Value. Some reduction in the Error of Volatility is beneficial but has a small impact. Note that Volatility and Error of Volatility are distinct quantities. Reducing Volatility means the Stock Price is more stable over time. This means the stock is less risky but also has less opportunity for gains. Reducing the Error of Volatility means there is more certainty in the value being used to specify Volatility. The error is a measure of the quality of the information about Volatility. Any use of statistics on past data may have to be re-derived using the uncertainty number formats. For example, the formulas for mean and standard deviation have to be updated to handle dual numbers and duals.

8.4. Utilization of Error Calculated with Duals Arithmetic

Overall, there is a multi-dimensional recipe that can be derived based on the ranking of error contributions in the order of importance:

1. Error of Spot Price
2. Error of Strike Price
3. Error of Time-to-Expiry
4. Error of Volatility
5. Error of Risk-free Rate

Changes on the above listed errors have to be taken in context with the centers as this determines the location on the Black-Scholes surface graphs. With this extra information, a utilization process would include algebraic combinations of the Call Value (center) and Error of Call Value components. Such algebraic formulas would connect surface graphs that have been shown. For example, the Call Value (Figure 12) can be connected to the large contributor, Error of Call Value due to Spot Price (Figure 15), that parallels the surface shape of 'Greek Delta'. This would illuminate trade-offs for profitability of the call option and the components of risk from the uncertainty or error propagations.

One example is to interpret the contributions to Error of Call Value as a *thickening* of the Call Value surface shown on Figure 12. This idea is based on the superposition principle that applies the +/- as local up and down deviations of the exactly-thin Black-Scholes surface of Figure 12. Then the Error of Call Value on Figure 13 is superimposed on the Figure 12 surface to obtain a new graph that is a *Thick Black-Scholes surface*. This is one way to interpret the error. To use the five-dimensional error vector requires that we have an idea about what 'thickness' means for a five-dimensional object. For example, if someone wants to measure the thickness of a cube-like three-dimensional object, there are three possible answers. The ability to flatten or render the error vector to a single magnitude helps view it but also does not contain every bit of information the Duals Arithmetic provides.

Since the Duals Arithmetic does not use statistical concepts such as mean, standard deviation or error distributions, there is no preference for the Call Value location within the thickness of this new surface. There is no peak on a distribution or tails of the distribution because there is no distribution. The center is not a magnet that dictates any peak in a distribution; *when it comes to error, one has to discard the idea of the distribution*.

Why is discarding the distribution a good idea? The idea of descriptive statistics is to generate single numbers (moments) to represent a data set. For example, the mean value is determined by a centrality condition that solves the most-central-datum as indicated by the minimum variance relative to a uniform datum. The problem with this is that many data sets contain outliers or rogue events and distributions are meant to show centrality or modality in the data set. Clinging to the 'statistical idea' means we either have to ignore these rogue events or try to comfort ourselves into believing that they are rare and our model formulas deal only with a trend that fits the theory. The problem is these events do happen and are not rare [11]. After an event happens, the traditional statistics are not equipped to describe what happened. As mentioned earlier, instead of being a source for uncertainty analysis, statistics is an application for Duals Arithmetic and formulas have to be updated to handle Duals number format and Duals Arithmetic.

The Duals Arithmetic does not use statistical concepts but relies instead on error sourcing for every number and conversion of the currently-used algorithm. The Duals Arithmetic uses algebra and not differential or integral calculus that depend on the assumption of small error. Consequently, the Duals Arithmetic can represent a wide range of events. Other methods can do this too but large error sources can grow and overwhelm the primary calculation. For example, the Monte-Carlo Arithmetic is not well behaved and requires careful implementation, such as using a very large sample size. A system that is in transition due to a rogue event is similar to this error-growth. In the transition, past behavior becomes less important until an entirely new regime of behavior unfolds. The Duals Arithmetic reports the smallest Error of Call Value and therefore any growth of error is due to the modelled behavior rather than a by-product of poor uncertainty arithmetic.

9. CONCLUSIONS

Conclusions are drawn according to the implementation and performance of five uncertainty arithmetics: Interval, Monte-Carlo, Differential, Chordal and Duals. These are validated using Exact Arithmetic and Traditional Arithmetic.

The use of memory, although a capital cost, provides the benefit of information. All uncertainty arithmetics using dual numbers have the same memory requirements-double that required for Traditional Arithmetic. A more advanced method, Duals Arithmetic, requires six times the memory requirements as Traditional Arithmetic. But this increased information provides details on the contributions to Error of Call Value and a basis for manipulating the error budget. All of the uncertainty arithmetics except Differential provide error calculations in parallel with the center (primary) calculation and this means they are simultaneous. Only the Duals Arithmetic reports all components of Error of Call Value simultaneously.

The Black-Scholes model is a small pilot example and, based on this, formula were shown to calculate the memory requirements for scaling up to larger calculations.

The fastest calculation is the Traditional Arithmetic but this provides no uncertainty information. The runtimes of other methods were benchmarked to the Traditional Arithmetic to assess the cost of adding uncertainty calculation capability. Relative runtime was defined as the runtime divided by the Traditional Arithmetic runtime. The multi-point methods of Intervals and Monte-Carlo were the slowest to run. The Duals Arithmetic was third slowest.

A benefit-to-cost ratio (BCR) was defined as information divided by relative runtime. As a reference case, running Traditional Arithmetic a second time with perturbed inputs is a way to obtain an uncertainty calculation. While this doubles the information, the runtime is doubled and the BCR is constant. The Duals Arithmetic has the highest BCR mainly due to the greater amount of information provided in a slightly slower speed. But Duals Arithmetic also had a smaller variance in runtime making a better case for scale-up to larger problems. Monte-Carlo Arithmetic was the worst BCR as it is slower than all other arithmetics and provides the same information as the other methods (except Duals) and also had a great amount of variance in the runtimes making scale-up more difficult to project.

Many uncertainty calculation methods are not applicable over the defined range of Spot Price and Time-to-Expiry. In particular the critical edge near zero Time-to-Expiry causes failures in the calculations due to either a divide-by-zero problem or a square-root-of-negative problem. Only the Duals Arithmetic is fully robust, allowing calculations at the zero Time-to-Expiry edge and Spot Price-equal-to-Strike Price point on this edge. The reason is Duals Arithmetic allows a divide-by-zero-dual, as the defined 'zero dual' has a zero center (hence the name 'zero') but a non-zero error vector. The square-root-of-negative that occurs in the presence of Error of Time-to-Expiry near zero Time-to-Expiry is also solved using the closure-signatures feature of the Duals Arithmetic.

All uncertainty arithmetics except the Duals Arithmetic calculate errors that are too large and invalidate use of the Black-Scholes calculations. However, since the fundamental model equations are common among all methods and the Duals Arithmetic results are acceptable, this shows that it is the poor uncertainty arithmetic in other methods that is to blame. The Black-Scholes model is always in doubt due to adherence to theoretical assumptions. The specification of numbers for input is a form of assumption, that is, we assume we know the number being input. The use of dual numbers allows the possibility that we do not know the inputs exactly at every moment and permits a range of possible values for inputs. With dual numbers and appropriate arithmetic, the calculation is run once and uncertainty is calculated. This is a better alternative to re-running Traditional Arithmetic in a 'black-box' with varied inputs.

Once Error of Call Value is known, it can be used in decision making to obtain higher certainty. However, some methods are not cast-in-stone and the error information yielded can be manipulated. For example, the Interval Arithmetic and Monte-Carlo Arithmetics can be improved by chosing a larger number of points that superimpose error instances on inputs. Other methods such as Differential Arithmetic and Duals Arithmetic have no parameters and give one consistent answer. However, the error answers for the Differential Arithmetic deteriorate as error grows and locally linear regions cannot fit curvature. The Duals Arithmetic is algebraic and stays consistently good as error grows.

A strategy of error reduction and lowering uncertainty requires knowledge about the sources input and how they contribute to the Error of Call Value. Methods using dual numbers force all error inputs into one output error for each operation. Unless the operation is an unary function, this stamps out the identity of error that was input and it is difficult to discover the contribution of each input. Additional calculations would need to be run, changing the role of each input error. Duals Arithmetic formats a multi-dimensional error vector that is maintained at every step of calculation, evolving the contributions as the calculation proceeds. The added memory is a benefit when contributions are easily identified from the error vector. Knowing each contribution means a strategy can be employed to reduce overall error by watching its components.

The Error of Spot Price has the largest contribution and this would be an effective place to start. The surface shape is important as it dictates the potential decrease of error. The Error of Risk-free rate has a very small contribution and is waste of effort to reduce. The same can be said of Error of Volatility. Although there are cases where these weights change, the low contributions remain true over the majority of the Black-Sholes surfaces.

Based on this report, the new frontier of uncertainty calculations is based, not on numeric arithmetic but on geometric arithmetic. The dual numbers are cast as scalars for geometrical objects and the arithmetic of addition, subtraction, multiplication and division has to apply to geometry. Two geometric arithmetic methods, chordals (CertainError class 1) and duals (CertainError class 2) were demonstrated in this report and the Duals Arithmetic was found to be advantageous and superior on many fronts.

Two additional methods, using multi-duals (CertainError class 3) and geoms (CertainError class 4) number formats utilize even more sophisticated arithmetic but provide higher fidelity in the error calculations and are suitable for applications where high certainty is critical. These two methods are beyond the scope of this report.

REFERENCES

1. 'The Pricing of Options and Corporate Liabilities,' Black, F.; Scholes, M., Journal of Political Economy, Vol. 81, No. 3, May-June, 1973, pp. 637-654.
2. 'Lecture 15.2 – Unary Functions,' ES581 Fall 2014 Course Notes, Prof. R.S. LaFleur
3. 'A New Approach to Error Arithmetic,' Olver, F.W.J., SIAM Journal on Numerical Analysis, Vol. 15, No. 2, April, 1978, pp. 368-393.
4. 'A Lucid Interval,' Hayes, B., American Scientist, Vol. 91, No. 6, November-December, 2003, pp. 484-488.
5. 'Interval Arithmetic and Automatic Error Analysis in Digital Computing,' Moore, R.E., Applied Mathematics and Statistics Laboratories, Stanford University, Technical Report No. 25, Nov. 15, 1962, pp. 1-9.
6. 'Evaluation of measurement data – Supplement 1 to the "Guide to the expression of uncertainty in measurement" Propagation of distributions using Monte Carlo method,' JCGM 101:2008. pp. 13-16.
7. 'Describing Uncertainties in Single-Sample Experiments,' Kline, S.J.; McClintock, F.A., Mechanical Engineering, January 1953, p. 3-8.
8. 'Lecture 9 – Chordals,' ES581 Fall 2014 Course Notes, Prof. R.S. LaFleur
9. 'Lecture 16 – 2D Applications,' ES581 Fall 2014 Course Notes, Prof. R.S. LaFleur
10. 'Deriving derivatives of derivative securities,' Carr, P., J. Computational Finance, Vol. 4, No. 2, Winter 2000/2001.
11. 'Option Traders Use (very) Sophisticated Heuristics, Never the Black-Scholes-Merton Formula,' Haug, E.G.; Taleb, N.N., J. Economic Behavior and Organization, Vol. 77, No. 2, 2011.