2. Problems with Standard Models exposed or remedied with *Mathematica*

Part 2 of
"Modelling Financial Derivatives with *Mathematica*"

Dr William T. Shaw
Quantitative Analysis Group, Nomura International plc

Präsentiert von:
ADDITIVE Soft- und Hardware für Technik und Wissenschaft GmbH
Rohrwiesenstraße 2
61381 Friedrichsdorf
Telefon: 06172-5905-0 Fax: 06172-77613
mathematica@additive-net.de http://www.additive-net.de

1. Why Implied Volatility is a Potentially Undefined Concept

There are numerous reasons to take issue with the concept of implied volatility. We could argue all day about
pricing mechamism backwards through a simple model - this has always seemed to me like pointless curve fit
highlight the exceptionally bad behaviour of implied volatility when we consider the simplest non-vanilla opti

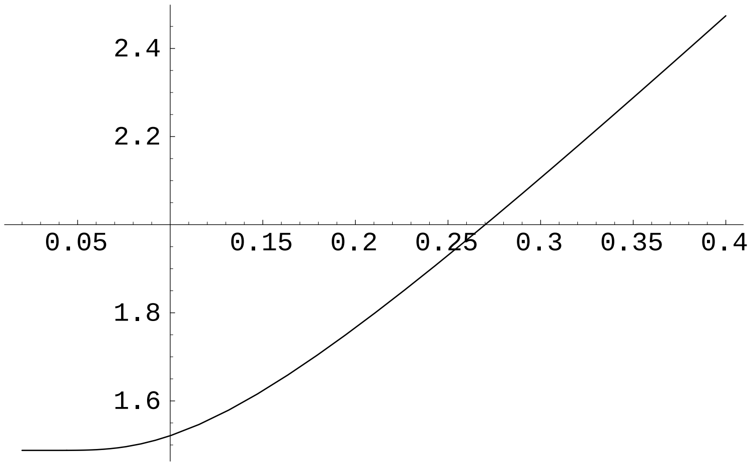
When Things are OK

When the valuation is a monotonic function of volatility (strictly increasing or decreasing), there is usually no
falls into this category. Let's build a symbolic model of this option and plot the value as a function of volatility

```
Norm@z_?NumberQD := NA0.5 ErfA z/√2 E + 0.5E; Norm@x_D := 1/2 i1 + ErfA x/√2 E
done@s_, s_, k_, t_, r_, q_D := (Hr - qL t + LogA s/k E)/(s √t) + s √t/2;
dtwo@s_, s_, k_, t_, r_, q_D := (Hr - qL t + LogA s/k E)/(s √t) - s √t/2; BlackScholes
 s Exp@-q tD Norm@done@s, v, k, t, r, qDD - k Exp@-r tD Norm@dtwo@s, v, k,
```

We plot the option value as a function of the volatility, for an option where the strike is at 10, the underlying a
free rate (cc) and zero dividends:

```
Plot@BlackScholesCall@11, 10, vol, 0.05, 0, 1D, 8vol, 0.02, 0.40<,
```



Things are sometimes OK, sometimes highly unstable

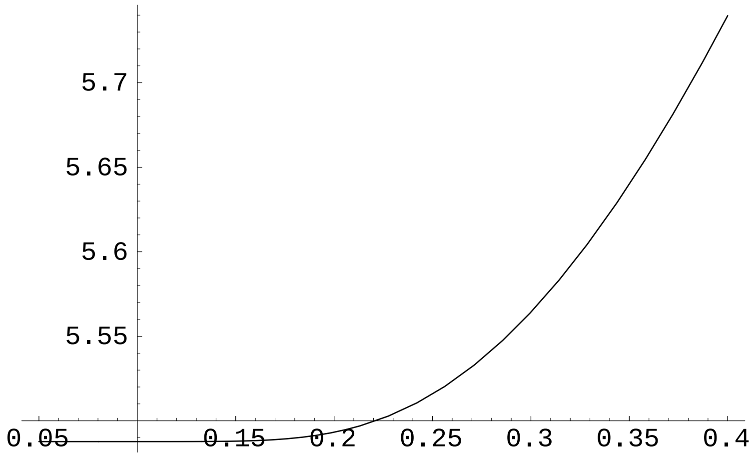
It does not take much to mess up the calculation of implied volatility. Let's add a dilution effect to the Black-S
This is one of the simpler models discussed by:

Lauterbach, B., & Schultz, P., Pricing Warrants: An Empirical Study of the Black-Scholes Model and its Alte

```
WarrantEqn@p_, k_, sd_, r_, t_, warprice_, shares_, warrants_,
shperwar_, q_D := Hshares * shperwar  Hshares + shperwar * warrantsLL *
BlackScholesCall@p * Exp@-q * tD + warrants * warprice  shares, k, sd, r,

WarrantValue@p_, k_, sd_, r_, t_, shares_, warrants_, shperwar_, q_D
warprice  . FindRoot@
warprice == WarrantEqn@p, k, sd, r, t, warprice, shares, warrants, shp
```

```
Plot@WarrantValue@15, 10, vol, 0.05, 1, 1000, 100, 1, 0D, 8vol, 0.0
```



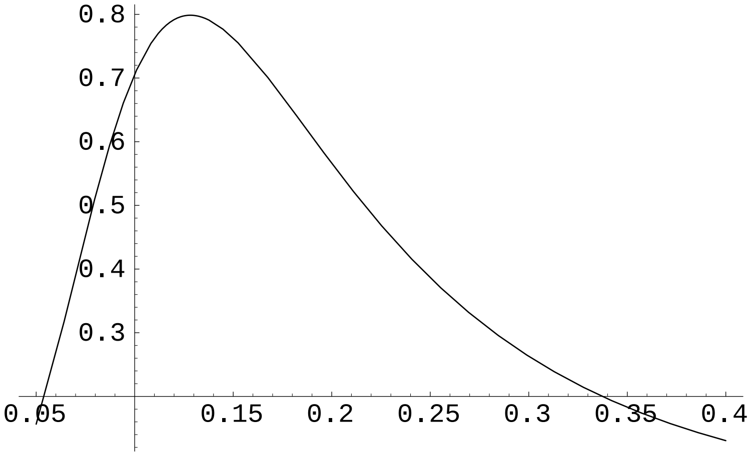
For lower prices the implied vol is highly unstable or non-existent.

Things are definitely not OK - there are two or even three implied volatilities

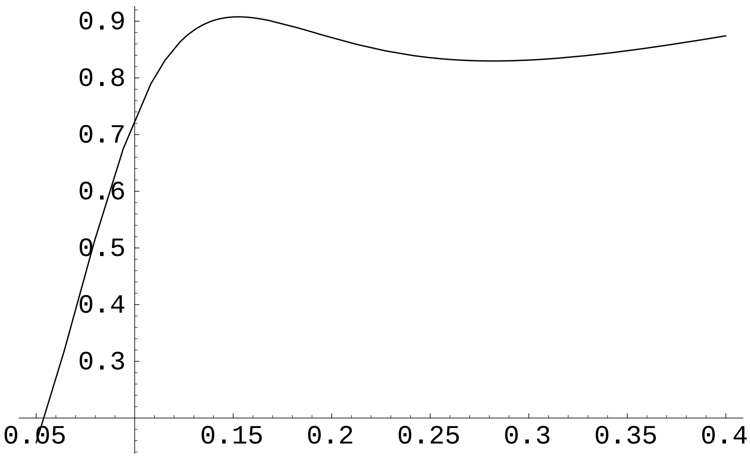
A simple barrier option will exhibit the phenomenon of there being two volatilities consistent with a given val underlying is at 45, the strike at 50 and the knockout barrier at 60. As we first increase the volatility from a ve volatility increses further the probability of knockout increases, lowering the value of the option. By tweaking for a small range of market option prices. The model of barriers we use a *Mathematica* implementation of the

Rubenstein, M. & Reiner, E., Breaking Down the Barriers, RISK Magazine, September 1991.

```
Plot@UpAndOutCall@0, 45, 50, 60, vol, 0.05, 0.0, 1D,
8vol, 0.05, 0.4<, PlotRange -> AllD;
```

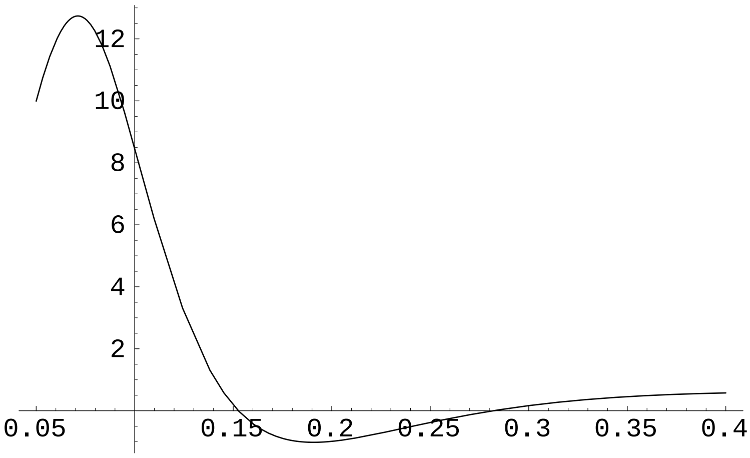



```
Plot@UpAndOutCall@1.7, 45, 50, 60, vol, 0.05, 0.0, 1D,
8vol, 0.05, 0.4<, PlotRange -> AllD;
```



In other words, vega is positive apart from an interval in which it is negative:

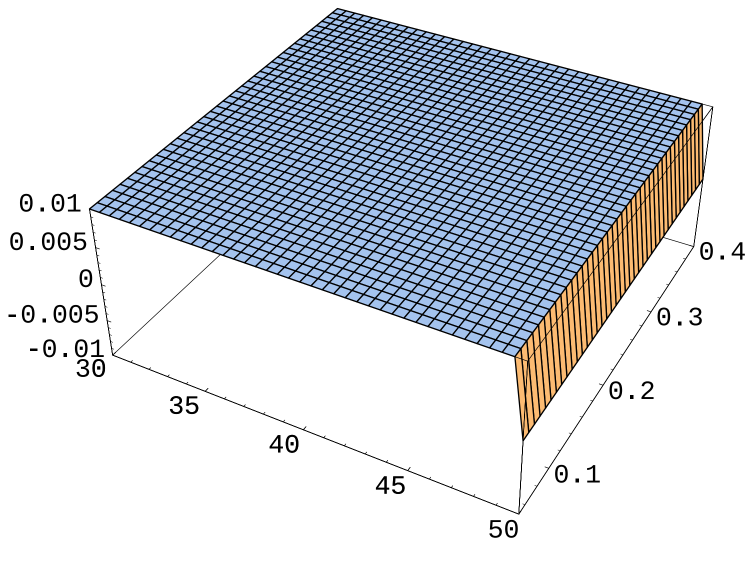
```
Plot@UpAndOutCallVega@1.7, 45, 50, 60, vol, 0.05, 0.0, 1D,
8vol, 0.05, 0.4<, PlotRange -> AllD;
```



Things are impossible - there are infinitely many answers

If you want to see just how bad it can get, a barrier option can do still more interesting things. If we consider a barrier, and arrange for the risk-free rate and the dividend yield to concide, we can get a dead zero vega (and z coincide with the computed value, you can have any implied volatility you want. Otherwise there is no implied part 1.

```
Plot3D[UpAndOutPut[0, S, 50, 50, vol,0.1, 0.1, 1],
{S, 30, 50}, {vol, 0.05, 0.4 }, PlotRange -> {-0.01, 0.01}, PlotPoi
```



A safe rule is to avoid reporting implied vols for anything but vanilla calls and puts. You need to check any ex reporting implied vol. Probably a good idea to persuade traders to stop using it - not easy.

2.. Why are Derivatives Tricky from a Math View?

Let's explain why derivatives are a lot trickier than most people might think they are. There are many issues. T

The computation of "a derivative" is the calculation of the value of a function **and** several of its first and secor **sensitivity parameters.**

Why does this makes things (i.e. numerical computation) so nasty?

It is useful to go back to basic mathematical analysis, to remind ourselves of some points.

Non-Uniform Convergence 101

Let's think about errors in calculations as a function f. Suppose it is parametrized by a variable n such as tree (

$$f_n(x) \to 0$$

as

$$n \to \infty$$

In this case a (mathematical!) analyst would say that f *converges pointwise to zero*. The question arises as to v Naively one might expect that also becomes small as the function becomes small. Indeed, many functions satis

$$\frac{x^m}{n};\ \frac{\sin(x)}{n}$$

Unfortunately, it is not always true. The following classic example makes this clear. We consider the function:

$$f(x, n) = \frac{\sin(n x)}{n}$$

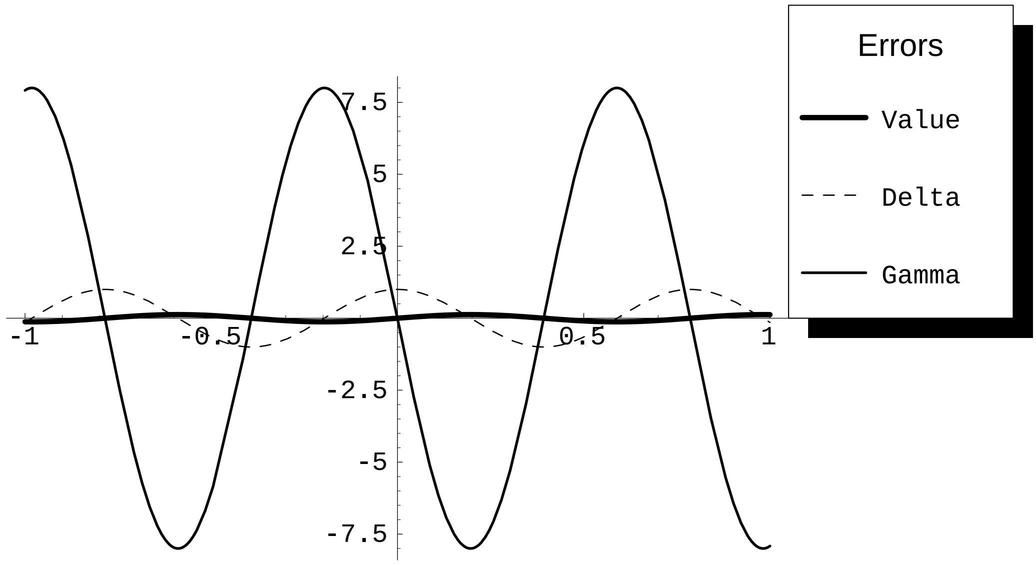
It's first derivative, or "delta", is then

$$\frac{\partial f(x, n)}{\partial x} = \cos(n\, x)$$

It's second derivative, or "gamma", is then

$$\frac{\partial^2 f(x, n)}{\partial x^2} = -n \sin(n\, x)$$

Suppose such a function is now thought of as the error in some numerical simulation. We must appreciate that possible moderate error in first derivative δ -> possible large error in second derivatives Γ.



The lesson of this is we can be deluded about the quality of our analysis by just looking at the accuracy of the how this can happen with a good algorithm is given below. Another simple visual example is to consider a seq goes to zero the slopes remain infinite at the edges.

3. Finite-Difference Methods - avoid Crank-Nicholson Schemes with bigger time steps

The use of implicit finite-difference schemes is increasingly popular. This is in part due to the fact that they ar schemes. You need to know that some of the popular ones are only stable, "sort of" - high frequency modes m havoc with the Greeks.

This is in fact not a new story. It has been known for about 40 years in the academic numerical analysis comm practitioners.

Background Refs

Mitchell, A.R. & Griffiths, D.F., 1980, The Finite Difference Method in Partial Differential Equations, John W **good example of how Douglas gives better results than CN**

Richtmeyer, R.D. & Morton, K.W., 1957, Difference Methods for Initial Value Problems, Krieger repri (Read this if you think I am crazy to recommend against CN!)

Smith, G.D., 1985, Numerical Solution of Partial Differential Equations: Finite Difference Methods, Oxford U

Wilmott, P., Dewynne, J. & Howison, S., 1993, Option Pricing - Mathematical models and computation,Oxfor

Wilmott, P., Dewynne, J. & Howison, S., 1995, The Mathematics of Financial Derivatives, Cambridge Univer

To develop this topic, I assume that one way or the other, our option pricing problem has been reduced to the c

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

I realise this is a major limitation - but the fact that interesting problems arise in this simplest of cases means o Black-Scholes, discretized IR PDEs). We introduce a discrete grid with steps Δt, Δx, where Δx is the grid ste time, and set

$$u_n^m = u(m\,\Delta t, n\,\Delta x)$$

All the difference schemes involve a parameter α that is given by

$$\alpha = \frac{\Delta t}{\Delta x^2}$$

and the second-order difference operator

$$\delta_x^2 u_n^m = u_{n+1}^m + u_{n-1}^m - 2 u_n^m$$

also plays a role. A well-known scheme for the solution of the diffusion equation is the Crank-Nicholson sche

$$u_n^{m+1} - \frac{1}{2} \alpha \left(u_{n-1}^{m+1} + u_{n+1}^{m+1} - 2 u_n^{m+1}\right) = u_n^m + \frac{1}{2} \alpha \left(u_{n-1}^m + u_{n+1}^m - 2 u_n^m\right)$$

Another scheme given by

$$(1 - 6 \alpha)\left(u_{n-1}^{m+1} + u_{n+1}^{m+1}\right) + (10 + 12 \alpha) u_n^{m+1} = (1 + 6 \alpha)\left(u_{n-1}^m + u_{n+1}^m\right) + (10 - 12 \alpha) u_n^m$$

is called the *Douglas* scheme. It is very important due to the fact that it is exact to order δ_x^4, even though it con
intial conditions (here payoffs), the Douglas scheme gives much better results than Crank-Nicholson.

Another view of the Dougals scheme is that it is the natural implicit generalization of the trinomial model, to v

However, neither of these implicit schemes are suitable for option-pricing problems, where the payoff is typic
strike), or is even discontinuous. This has in fact been known since at least the work of R & M.

3 Time-Level Schemes

There are a variety of ways of curing the oscillation problem, most of which involve 3 time-levels if accuracy
There is a 3 time-level extension of the Douglas scheme which takes the form.

$$\left(\frac{1}{8} - \alpha\right)\left(u_{n-1}^{m+1} + u_{n+1}^{m+1}\right) + \left(\frac{5}{4} + 2 \alpha\right) u_n^{m+1} = \frac{1}{6}\left(u_{n-1}^m + u_{n+1}^m + 10 u_n^m\right) - \frac{1}{24}\left(u_{n-1}^{m-1} + u_{n+1}^{m-1} + 10 u_n^{m-1}\right)$$

This type of process requires a kick-off procedure, since initially we only know u^1. We use the ordinary Doug
with $\alpha \to \alpha / 2$, to get the scheme going.

Crank-Nicholson vs 3 Time-Level Douglas

It is a simple matter to implement these schemes on a computer. I did so in *Mathematica* 3.0, for a vanilla Eur
solution for a European Put, inspecting both the valuation and the Greeks. Standard parameters were:

Underlying = 10;

Risk-free Rate = 5%;

Dividend Yield = 0;

Expiry = 5 Years;

Volatility = 20%

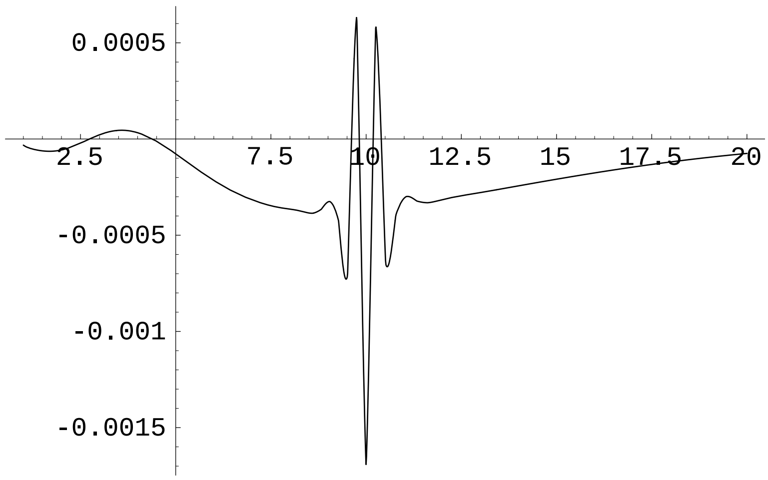
We were interested in using moderately large time-steps, so set $\alpha = 8$. Graphical results follows. In all cases th
using the *Mathematica* 3.0 compiler to create solvers for tridiagonal systems, and SOR, PSOR analogues

Error in Valuation - Crank-Nicholson

A cursory inspection of the numerical answers vs the exact solution suggests that all is well.

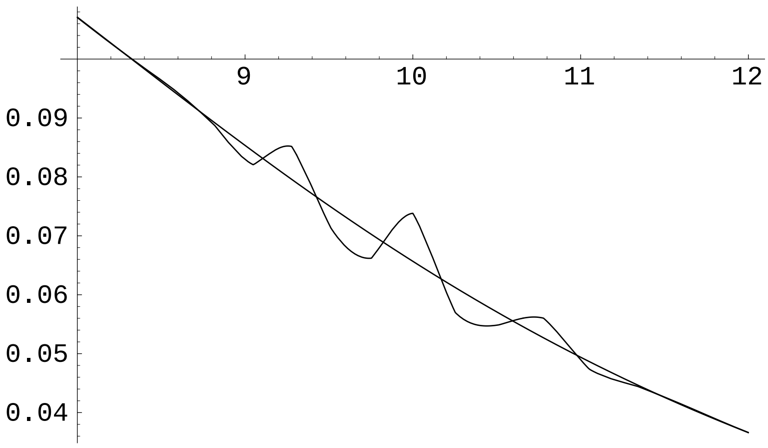
S	Exact	Crank-Nich	Error
2.00000	5.78850	5.78860	-0.00006
3.00000	4.80050	4.80050	0.00002
4.00000	3.86160	3.86150	0.00003
5.00000	3.02080	3.02090	-0.00008
6.00000	2.31060	2.31080	-0.00022
7.00000	1.73830	1.73870	-0.00032
8.00000	1.29290	1.29320	-0.00036
9.00000	0.95445	0.95478	-0.00033
10.00000	0.70016	0.70187	-0.00171
11.00000	0.51461	0.51492	-0.00031
12.00000	0.37735	0.37766	-0.00032
13.00000	0.27698	0.27726	-0.00028
14.00000	0.20370	0.20394	-0.00024
15.00000	0.15019	0.15040	-0.00021
16.00000	0.11108	0.11125	-0.00018

But nothing could be further form the truth, as a plot of the error in the numerically-computed solution reveals



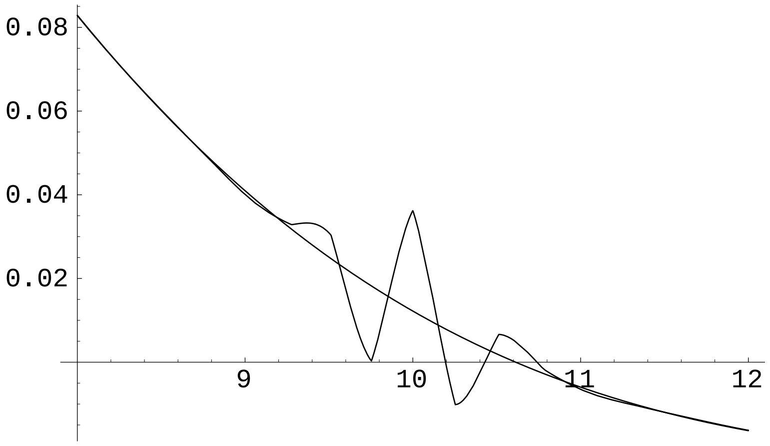
Numerical vs Exact Gamma - Crank-Nicholson

By the time we have differentiated to get gamma the error has become significant, as an overlay of the exact a



Numerical vs Exact Theta - Crank-Nicholson

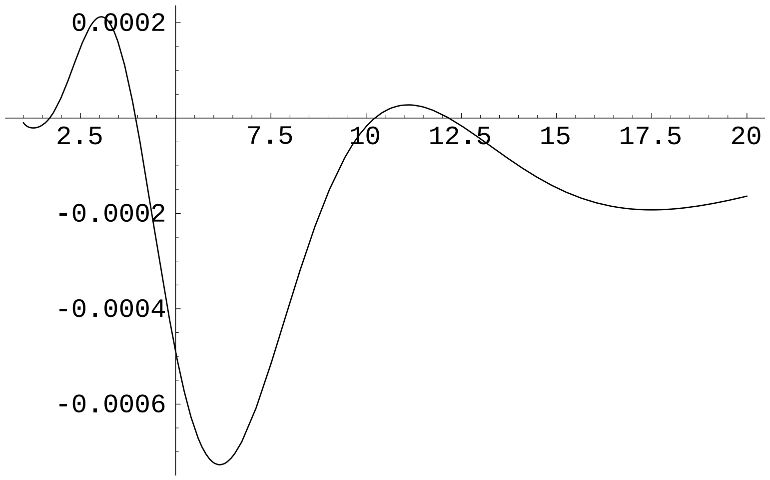
teh parameter theta is coupled to gamma through the Black-Scholes equation, resulting in even nastier problen



The error in theta is about 200% of its exact value.

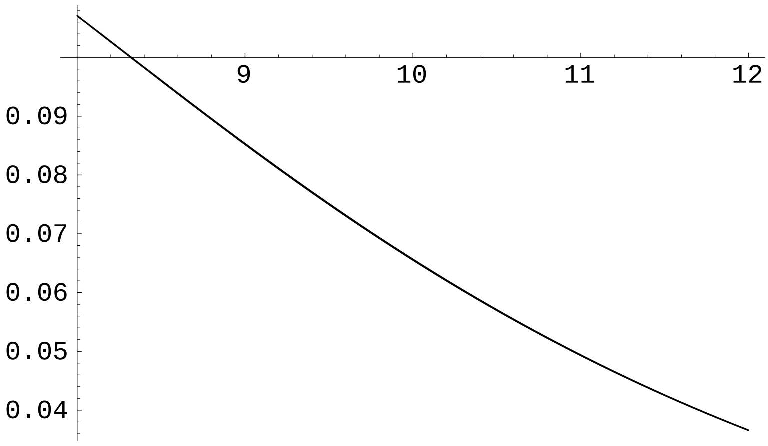
Error in Valuation - 3 Time-Level Douglas

This time the errors in the valuation are much smoother and slightly smaller in scale.



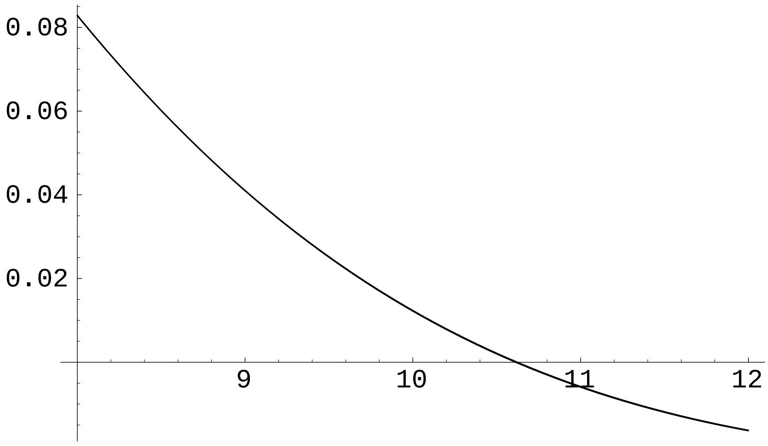
Numerical vs Exact Gamma - 3 Time-Level Douglas

This time the differentiation to get the Greeks gives us no problems - when we overlay the exact and computed



Numerical vs Exact Theta - 3 Time-Level Douglas

The 200% error in theta has gone away:



The 3 time-level scheme offers a dramatic increase in the accuracy of the Greeks, for very little extra work. **<u>Tl compromising accuracy in both the valuation and the Greeks - we have found the 3 time-level Douglas t 10 years to expiry with only 40 time steps.</u>**

4. Conceptual Issues with Trees

Probability vs Symmetry

It is well known that standard treee models suffer from the potential for negative probability or negative asset completely and work from the point of view of symmetry relations. This will allow us to see in a natural way l

Again - we pose and investigate the problem within the equity-like Black-Scholes world. Corresponding issue a backwards evolution scheme of the form

$$V(S, t) = e^{-r \Delta t} \sum_{i=1}^{n} w_i V(S u_i, t + \Delta t)$$

The variables w_i are called the "weights", and the points $S * u_i$ are called the "abscissas". If $n = 2$ we have a ge general trinomial scheme. Explicitly, for a binomial scheme

$$V(S, t) = e^{-r \Delta t} [w_1 V(S u_1, t + \Delta t) + w_2 V(S u_2, t + \Delta t)]$$

Basic Solutions of the Black-Scholes equation

What we shall do now is to write down some basic solutions of the Black-Scholes equation that we wish to be somewhat arbitrary. One question is how many solutions to use? For a binomial scheme, we have two weights For a trinomial scheme, there are 3 weights, 3 abscissas and the time-step, totalling 7. It is traditional to allow unknowns for a binomial scheme, and 6 for trinomial.

What we shall demand is that the tree rule is such that it is consistent with certain power solutions of the Black

$$V = Z \left(\frac{S}{K}\right)^n e^{-r(T-t)} e^{\left\{(r-q)n + \frac{n(n-1)\sigma^2}{2}\right\}(T-t)}$$

The special cases for $n = 0, 1, 2$ are:

$$V = Z\, e^{-r(T-t)}$$

$$V = Z \left(\frac{S}{K}\right) e^{-r(T-t)} e^{(r-q)(T-t)}$$

$$V = Z \left(\frac{S}{K}\right)^2 e^{-r(T-t)} e^{(2(r-q)+\sigma^2)(T-t)}$$

Considering the first negative powers, for $n = -1, -2$ we have:

$$V = Z \left(\frac{K}{S}\right) e^{-r(T-t)} e^{(-(r-q)+\sigma^2)(T-t)}$$

$$V = Z \left(\frac{K}{S}\right)^2 e^{-r(T-t)} e^{(-2(r-q)+3\sigma^2)(T-t)}$$

Let's demand that the binomial rule be consistent with some of the power solutions. **We call a tree satisfying "symmetric tree", because the mapping defined by the tree rule preserves, or leaves invariant, the solut** this gives us:

$$w_1 u_1^n + w_2 u_2^n = e^{\left(\frac{1}{2}\sigma^2 n(n-1) + n(r-q)\right)\Delta t}$$

and for $n = 0, 1, 2$ this gives

$$w_1 + w_2 = 1$$

$$w_1 u_1 + w_2 u_2 = e^{(r-q)\Delta t}$$

$$w_1 u_1^2 + w_2 u_2^2 = e^{(2(r-q)+\sigma^2)\Delta t}$$

The first pair of negative powers give us the equations, for -1, -2 respectively,

$$\frac{w_1}{u_1} + \frac{w_2}{u_2} = e^{(-(r-q)+\sigma^2)\Delta t}$$

$$\frac{w_1}{u_1^2} + \frac{w_2}{u_2^2} = e^{(-2(r-q)+3\sigma^2)\Delta t}$$

The condition that the weights sum to unity is traditionally regarded as arising from the fact that these weights view is unnecessary - it suffices to demand that "cash is conserved", i.e., that the tree preserves the simple cash these symmetry conditions can indeed be derived from a probabilistic view - you consider matching various m

It is clear that we can generate weights and abscissas consistent with up to 4 simple solutions of the Black-Sch solutions will be called "supersymmetric". First, however, we need to see how some standard models arise, an

Schemes of Jarrow-Rudd Type

We impose the conditions

$$w_1 = w_2 = \frac{1}{2}$$

which solves the cash conservation condition, leaving us with two unknowns and two conditions based on the

$$u_1 + u_2 = 2\, e^{(r-q)\Delta t}$$

$$u_1{}^2 + u_2{}^2 = 2\, e^{(2(r-q)+\sigma^2)(T-t)}$$

These can be solved with pen and paper, by solving a quadratic sequation, but we can be lazy and get *Mathem*

```
soln = Solve[{u1 + u2 == 2 Exp[(r - q) Δt], u1^2 + u2^2 == 2 Exp[(2 (r - q
```

The larger of the two solutions (here u_2) is now called u, as is conventional

```
u = Simplify[PowerExpand[Simplify[u2 /. soln[[1]]]]]
```

$$e^{(r-q)\Delta t}\left(1 + \sqrt{e^{\Delta t\, \sigma^2} - 1}\right)$$

```
d = Simplify[PowerExpand[Simplify[u1 /. soln[[1]]]]]
```

$$e^{(r-q)\Delta t}\left(1 - \sqrt{e^{\Delta t\, \sigma^2} - 1}\right)$$

These give us the "exact" form of the up and down states in the Jarrow-Rudd model. Note that the d variable c

$$\sigma^2 \Delta t \ll \log(2)$$

for this particular scheme to be sensible. This is the version of JR described by Wilmott et al. However, there i described by Hull (1996), and appears to be the form originally given by Jarrow and Rudd. In the approximate down scale factors are written as:

$$u = e^{\left(r-q-\sigma^2/2\right)\Delta t + \sigma\sqrt{\Delta t}}; \quad d = e^{\left(r-q-\sigma^2/2\right)\Delta t - \sigma\sqrt{\Delta t}}$$

This then conveniently avoids the down scale factor becoming negative, but this approximation violates the m

Schemes of Cox-Ross-Rubenstein Type

To get schemes similar to those of CRR, we impose the conditions

$$u_1 = 1/u_2$$

leaving us with two unknowns and conditions:

$$w_1\, u_1 + (1 - w_1)\, 1/u_1 = e^{(r-q)\Delta t}$$

$$w_1\, u_1^2 + (1 - w_1)\, 1/u_1^2 = e^{(2(r-q)+\sigma^2)\Delta t}$$

This leads to a very well known scheme with potentially negative up and down "probabilities". There are "exa

Supersymmetric Schemes?

The question arises as to whether we can build other schemes that are "supersymmetric". We can pick one oth abscissas so that this solution is preserved. The question arises as to which one to take. We can only rely on co different possibilities is helpful. We note that in some situations, for example Put options, we wish to model ca of the underlying. This suggests that it might be useful to consider setting the tree parameters so that the soluti use the first negative power, this creates a supersymmetric tree rule, and will turn out to have an interesting im

With this choice, we have the four unknows satisfying the four equations (the last one is the new constraint tha

$$w_1 + w_2 = 1$$

$$w_1 u_1 + w_2 u_2 = e^{(r-q)\Delta t}$$

$$w_1 u_1^2 + w_2 u_2^2 = e^{(2(r-q)+\sigma^2)\Delta t}$$

$$\frac{w_1}{u_1} + \frac{w_2}{u_2} = e^{(-(r-q)+\sigma^2)\Delta t}$$

In the binomial case, we wish to solve four of the relations:

$$w_1 u_1^n + w_2 u_2^n = e^{(\frac{1}{2}\sigma^2 n(n-1)+n(r-q))\Delta t} = A^n B^{n(n-1)/2};\quad A = e^{(r-q)\Delta t};\ B = e^{\sigma^2 \Delta t}$$

which serve to define A, B also. We give the set with $n = -1, 0, 1, 2$ to *Mathematica* as the system of equatio

```
symmetries = {w1 + w2 == 1, w1 u1 + w2 u2 == A, w1 u1^2 + w2 u2^2 == A^2 B, w1/u1 + w2/u2
```

When *Mathematica* is asked to solve this system, two solutions are obtained - we just extract the first, and sim

```
soln = Simplify[Solve[symmetries, {w1, w2, u1, u2}][[1]]]
```

```
{w1 -> (3 + B - Sqrt[-3 + 2 B + B^2])/(6 + 2 B), w2 -> (3 + B + Sqrt[-3 + 2 B + B^2])/(6 + 2 B), u2 -> 1/2 A (1 + B - Sqrt[-3
 u1 -> 1/2 A (1 + B + Sqrt[-3 + 2 B + B^2])}
```

One can also check the extent to which neighbouring symmetries apply:

```
CheckSymmetry[u_, d_, p_, q_, n_, a_, b_, solution_] :=
 Factor[Simplify[p u^n + q d^n - a^n b^((n - 1) n/2) /. solution]]
```

They are approximately satisfied, all having a factor $(B-1)^2$, which is $O(\Delta t^2)$. We can look at a whole range o form $B = 1 + x$. You should think of x as $\sigma^2 \Delta t$.

```
MatrixForm@
 Table@Series@HCheckSymmetry@u1, u2, w1, w2, n, A, B, solnD  . B ->
```

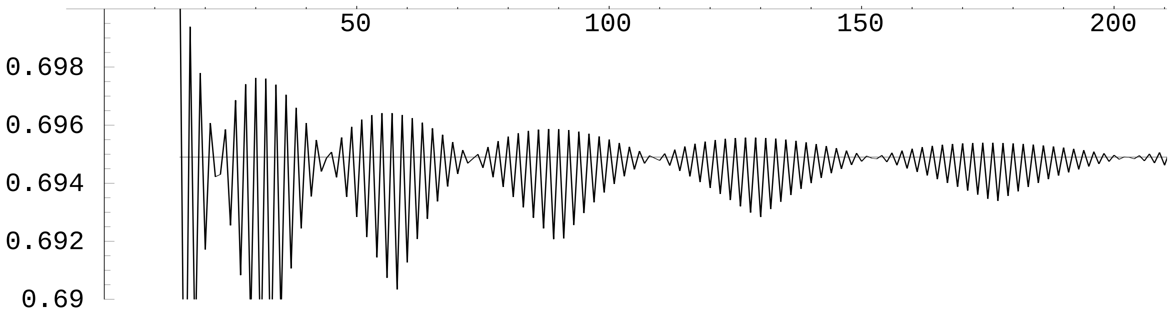
$$\begin{pmatrix} -\frac{140\,x^2}{A^6} + O@xD^3 \\ -\frac{70\,x^2}{A^5} + O@xD^3 \\ -\frac{30\,x^2}{A^4} + O@xD^3 \\ -\frac{10\,x^2}{A^3} + O@xD^3 \\ -\frac{2\,x^2}{A^2} + O@xD^3 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2\,A^3\,x^2 + O@xD^3 \\ -10\,A^4\,x^2 + O@xD^3 \\ -30\,A^5\,x^2 + O@xD^3 \\ -70\,A^6\,x^2 + O@xD^3 \end{pmatrix}$$

We see that all the positive and negative solutions are preserved with an error that is of order Dt^2, though the s
from zero. I conjecture that this is the best that can be done with a binomial scheme. With a trinomial scheme,
way, with *Mathematica*.

5. Numerical Issues with Trees

Convergence is appalling:

```
Show@explot, CRRPlotCApp, PlotRange -> 80.69, 0.70<, DisplayFunctio
```



The scale of the error here might be acceptable, but it is easy to get more spectacular problems:

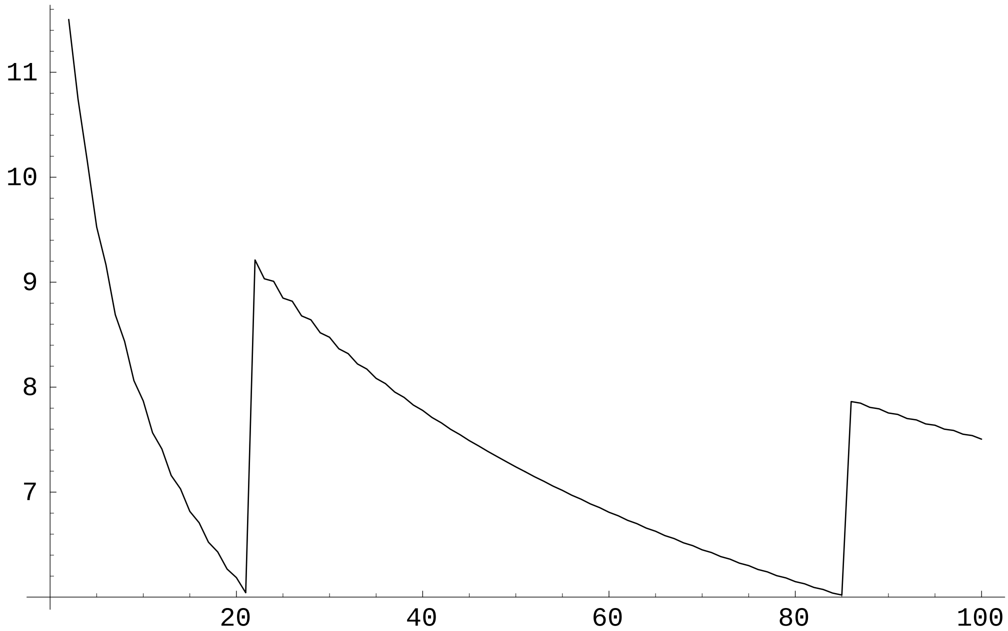
Boyle-Lau Effects with Barriers

Boyle, P.P. and Lau, S.H., - "Bumping Up Against the Barrier with the Binomial Method", Journal of Deriva

```
sawtooth = Table@8k, CRRDownAndOutCallApp@95, 100, 90, k, 0.1, 0, 0
```

We plot it, bearing in mind that the exact solution is almost exactly 6, which is given by the lower bound of th

```
ListPlot@sawtooth, PlotJoined -> TrueD;
```



The main point is to appreciate that with "standard" trees, there may be a severe mismatch between wh the option contract. If this is ignored, the values are not trustworthy. In simple vanilla cases, the best co be arranged to straddle the strike, using the methods we have described. With barriers, the Boyle-Lau r

Remarks on Greeks on Trees

There is a piece of folklore that says that Greeks are more difficult to define with the JR form than with the CR for the drift and get good values for delta, gamma, theta.

Rather more interesting is the observation, which seems to have been made recently by several people indepen based on probability - I derived them from the PDE view), that for an interesting class of European options, th other Greeks. That is:

$$\Lambda = S^2 \sigma \Gamma (T - t)$$

$$\rho = -(T - t)(V - S \Delta)$$

holds for large class of European style options. These formulae may be useful in situations where direct compu and FD (where you do it all again). I do not know

a) for precisely what class of options these constraints are satisfied;

b) how to generalize e.g., to Americans

I have checked, using *Mathematica*'s symbolic Greeks, that this allows you to get reasonable estimates of all G get usable estimates for Americans if you are well away from early exercise boundary, but it would be good to

6. Extending the Domain of Analytical Solutions with Mathematica

What we can do is to build implementations of:

a) Cunning exact solutions of difficult problems, but which may require advanced calculus tools;

b) Interesting analytical approximations.

Once we have these we can start to investigate other numerical algorithms, such as Monte Carlo methods. A si Viswanathan-Goldman-Sosin-Gatto models can be used to investigate the convergence of sampling models. T getting the variance down and getting the right answer in high-frequency sampling of the maxima/minima. Th to talk about a formula for the arithmetically averaged Asian developed by Geman, Yor and Eydeland. (G&Y, March 1995).

Suppose that the current time is t, and that the option matures at a time $T > t$. The averaging is arithmetic, con known average value of the underlying over the time interval $[t_0, t]$ is ES. GY define the following changes o

$$\tau = \frac{1}{4}\sigma^2 (T - t);$$

$$\nu = \frac{2(r - q)}{\sigma^2} - 1$$

$$\alpha = \frac{\sigma^2 (K(T - t_0) - (t - t_0)ES)}{4S}$$

and the function of a (transform) variable p as:

$$\mu(p) = \sqrt{\nu^2 + 2p}$$

The value of the average price option is then given by

$$\frac{e^{-r(T-t)} 4 S C(\tau, \nu, \alpha)}{(T - t_0)\sigma^2}$$

The remaining function $C[\nu, \tau, \alpha]$ is not given explicitly, but GY give its Laplace Transform,

$$U(p, \nu, \alpha) = \int_0^\infty C(\tau, \nu, \alpha) e^{-p\tau} \, d\tau$$

as an integral:

$$U(p, \nu, \alpha) = \frac{\int_0^{\frac{1}{2\alpha}} x^{\frac{\mu-\nu}{2}-2} (1 - 2\alpha x)^{\frac{\mu+\nu}{2}+1} e^{-x} \, dx}{p(p - 2\nu - 2)\Gamma\left(\frac{\mu-\nu}{2} - 1\right)}$$

where μ is as given above as a function of p and ν. GY develop a series description of the Transform and shov can be managed and simplified in *Mathematica*.

Mathematica Implementation of Exact Arithmetic Asian

The first part of the translation to software is obvious - we first enter the definitions of the various basic functi

```
τ[T_, t_, σ_] := σ^2 (T - t)/4;
ν[r_, q_, σ_] := 2 (r - q)/σ^2 - 1;
α[S_, ES_, K_, σ_, T_, t_, to_] := σ^2/(4 * S) (K * (T - to) - (t - to) *
μ[ν_, p_] := Sqrt[ν^2 + 2 * p]
```

Now we enter the definition of the integral that is part of the Transform, and request immediate evaluation:

```
F@p_, m_, n_, a_D =  ∫_0^{1/(2a)} x^((m-n)/2 - 2) H1 - 2 a xL^((m+n)/2 + 1) Exp@-xD  x
```

```
IfAa > 0 && ReA m/2 - n/2 E > 1 && ReA m/2 + n/2 E > -2 && Re@m + nD > -4,  1/Gamma@1 + mD
   J2^(1/2 H2-m+nL) a^(1/2 H2-m+nL) GammaA 1/2 H-2 + m - nLE GammaA 1/2 H4 + m + nLE Hypergeom
   ∫_0^{1/(2a)} E^-x x^(-2+ (m-n)/2) H1 - 2 x aL^(1+ (m+n)/2)  xE
```

We see that *Mathematica* can actually evaluate the expression in "closed form", albeit in terms of a special fur have received a result dependent on certain conditions - we can extract the answer (which is valid in our case)

```
G@p_, m_,  n_, a_D = F@p, m,  n, aD@@2DD
```

```
1/Gamma@1 + mD
 J2^(1/2 H2-m+nL) a^(1/2 H2-m+nL) GammaA 1/2 H-2 + m - nLE GammaA 1/2 H4 + m + nLE Hypergeome
```

There are further cancellations when we insert the other terms that make up the transform:

```
U@p_, m_, n_, a_D = SimplifyA G@p, m,  n, aD / (p Hp - 2 n - 2L GammaA (m-n)/2 - 1E) E
```

```
J2^(1/2 H2-m+nL) a^(1/2 H2-m+nL) GammaA 1/2 H4 + m + nLE Hypergeometric1F1A 1/2 H-2 + m - nL,
 Hp H-2 + p - 2 nL Gamma@1 + mDL
```

In standard mathematical notation, the Transform is just:

```
TraditionalForm@U@p, m, n, aDD
```

$$J2^{\frac{1}{2}(-m+n+2)} a^{\frac{1}{2}(-m+n+2)} \Gamma\left(\frac{1}{2}(m+n+4)\right) {}_1F_1\left(\frac{1}{2}(m-n-2); m+1; -\frac{1}{2a}\right)NN (p(p-2n-2)\Gamma(m+1))$$

We now have the ingredients to build the *Mathematica* model of the arithmetic average price Asian Call.

```
Off@NIntegrate::slwconD
```

```
AriAsianPriceCall@S_, ES_, K_, r_, q_, s_, T_, t_, to_D :=
 Module@8ti = t@T, t, sD, n = n@r, q, sD, a = a@S, ES, K, s, T, t, toD,
 contour = 2 n + 3;
 ac = Re@1  H2 PiL * NIntegrate@ U@Hcontour + I pL, m@n, Hcontour + I pLD,
       Exp@Hcontour + I pL * tiD, 8p, -500, 500<, MaxRecursion -> 11DD,
Exp@-r * HT - tLD * 4 * S  HHT - toL * s^2L * acD
```

```
AriAsianPriceCall@1.9, 0, 2, 0.05, 0, 0.5, 1, 0, 0D
```

```
0.193174
```

```
AriAsianPriceCall@2.0, 0, 2, 0.05, 0, 0.5, 1, 0, 0D
```

```
0.246417
```

```
AriAsianPriceCall@2.1, 0, 2, 0.05, 0, 0.5, 1, 0, 0D
```

```
0.306223
```

Remarks

This type of analytical result can be made the basis of a great deal of further study. They have been used by Fu College preprint, March 1997) to analyse in detail the efficiency of Monte Carlo simulation. This result can be used to test and expose the difficulties in a simulation world.

A Challenge! In my book I have issued a foolish challenge. To anyone who can get this working to 5dp in a sp can get it working in C++ within 40 programming hours, two bottles of vintage champagne. It took me a total Also contrast the effort required here with that required to solve the equivalent 3D PDE! This solution is some too quickly to a numerical method. We should spend more time on analytics and less time building huge simul cases need to be checked against the analytics.

7. The Normal Distribution

There are a number of rational approximations kicking around, with reasonable if not outstanding accuracy. A them. This may seem irrelevant, as we can differentiate the cumulative normal distribution analytically. Howe algorithm to get the Greeks, then you will see the problems.

What else can you do? It turns out that the power series and asymptotic series are not great, but that there is an is not new - but it is buried deep inside the algorithms section of the wonderful book "Numerical Recipes" (Pr (1972, Handbook of Mathematical Functions, Dover). Once you see it you realise it is easy to code in any lang small to moderate. Here is the latter:

```
ContinuedFractionApproxTwo@x_, n_D :=
Module@8u = Range@nD, v, w<,
v = Reverse@uD;
Fold@H2 * #2 - 1L + x^2 H-1L^#2 #2  #1 &, 1, vDD
```

```
ContinuedFractionApproxTwo@x, 20D
```

$$1-\frac{x^2}{3+\frac{2x^2}{5-\frac{3x^2}{7+\frac{4x^2}{9-\frac{5x^2}{11+\frac{6x^2}{13-\frac{7x^2}{15+\frac{8x^2}{17-\frac{9x^2}{19+\frac{10x^2}{21-\frac{11x^2}{23+\frac{12x^2}{25-\frac{13x^2}{27+\frac{14x^2}{29-\frac{15x^2}{31+\frac{16x^2}{33-\frac{17x^2}{35+\frac{18x^2}{37-\frac{19x^2}{20x^2+39}}}}}}}}}}}}}}}}}}}}$$

The approximation to the Normal distribution is given by:

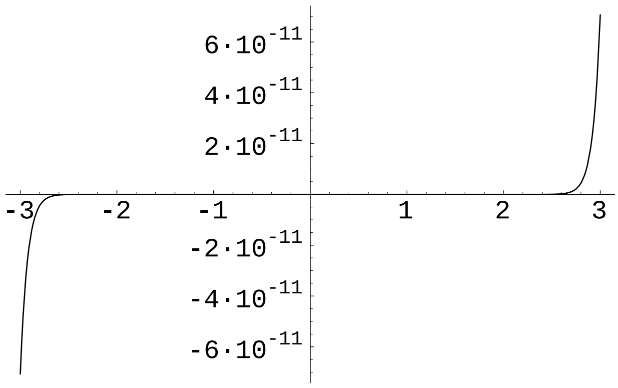
```
CFATwo@x_D = 1  2 + Exp@-x^2  2D  Sqrt@2 PiD x  ContinuedFractionApprox
```

```
Needs@"Derivatives`BlackScholes`"D
```

```
diffCFTwo@x_D = CFATwo@xD - Norm@xD;
```

Let's look at the error for $-3 < x < 3$:

```
Plot@diffCFTwo@xD, 8x, -3, 3<, PlotRange -> AllD;
```



This and the other continued fraction expansions are what I recommend for accurate computation of the cumu
Mathematica. They are given by equations (26.2.14) and (26.2.15) of Abramowitz and Stegun (1972). They ar
significantly smaller (here it is 5 orders of magnitude) than the best of the popular rational approximations.

Summary: Warnings and Suggestions

Do Not Dismiss Analytics - although they do not apply universally they are the only good tests;

Use a Symbolic Algebra System for Model Building **-** there are substantial gains when dealing with Greeks;

Often all is not what it seems:

Greeks may be way off even when fair value is accurate;

Implied Volatility may be useless;

Well respected algorithms (e.g Crank-Nicholson) may give wrong answers even when it looks good superficia

Models should be built adapted to boundary conditions - strength of Finite-Difference Models vs Binomial/Tri

Trees can be built so as to avoid negative probabilities or asset prices. However they are built their convergenc

Make more use of exact solutions - whether or not they are useful for a particular contract is irrelevant - they c
as control variates.

All comments and criticisms very welcome. Perhaps it is time for an open Derivatives Code Intercomparison S